Deep Learning-Based Involution Feature Extraction for Human Posture Recognition in Martial Arts

Desheng Chen[1*], Sifang Zhang[2*]
[1]School of Physical Education, Anyang Preschool Education College, Anyang 456150, China
[2]Department of Physical Education, Wuhan Sports University, Wuhan 430205, China
Email: ayysbgscds@126.com, z1234567sf@sina.com
[*]Corresponding author

Keywords: human action recognition, deep learning, long- and short-term memory neural networks, lightweight networks, feature extraction

Received: August 30, 2024

With the development of computers in recent years, human body recognition technology has been vigorously developed and is widely used in motion analysis, video surveillance and other fields. This study is based on deep learning to improve human pose estimation. Firstly, Involution's feature extraction network was proposed for lightweight human pose estimation, and this feature extraction network was combined with existing human pose estimation models to recognize human pose. Label and classify each joint point of the human body separately, add weights to each different part, extract feature between joint points at different times, and then input the extracted feature into long short-term memory neural networks for recognition. The experimental results show that the improved human pose estimation model reduces the parameter and computational complexity by about 40% compared to the original model, while also slightly improving accuracy. Comparing the performance of models under various algorithms with the proposed model in this study, the accuracy under the Eigen method is 81.3%, the accuracy under the STOP method is 82.5%, the accuracy under the DMM&HOG method is 85.3%, the accuracy under the Actionlet method is 87.6%, and the accuracy under the JAS&HOG2 method is 83.5%. The accuracy of the InNet LSTM method is 90.6%. The results indicate that the proposed model has good performance and can recognize different martial arts movements.

Povzetek: Za prepoznavanje človeške drže v borilnih veščinah so porabljene involucijske ekstrakcije značilk za globoko učenje.

1 Introduction

With the development of computers, artificial intelligence has become increasingly relevant to people's lives. The advent of computer vision enables computers to automatically recognize human actions and classify them [1]. Initially human movement recognition relied on decomposing video frame by frame and then acquiring information from it, and then recognizing human movements through image processing. This approach requires manually designing motion feature to represent the human body and then modelling the motion feature to achieve the recognition effect, but manually acquiring the feature requires a lot of time and effort [2]. In this study, a human skeleton network was created by using Deep Residual Networks (ResNet) combined with Involution's improved algorithm for feature extraction. The human posture at each moment is represented by the human skeleton, so that the human posture feature can be quantified by the human skeleton network. The extracted feature is then fed into the Long Short-Term Memory (LSTM) neural network for processing and recognition. This model is designed to efficiently extract and accurately recognize and classify human feature. The research is divided into four main sections, the first of which is a brief review of other research topics on human recognition. The second part is a review of the main methods used in this research, and the third part is the results of the model obtained by applying the methods to the research and analysing the results. The fourth part is a summary of all the above studies and an outlook for future research.

2 Literature review

With the development of computers, human body recognition technology has been vigorously developed and is widely used in motion analysis, video surveillance, etc. Liu et al. proposed a method for estimating the 3D pose of a single person in two views without camera parameters in order to cope with the problem of needing to know the camera parameters to obtain coordinate accuracy in the camera's two views. It extracts the joint points from two different views through 2D estimation and inputs them into a 3D regression network to generate 3D joint point coordinates. The coordinates are then combined with a 3D human pose recognition model to identify the human pose. The results of the study

indicated that this method extracted a high accuracy rate for human pose action recognition [3]. Ferreira et al. proposed a skeleton structure and deep semantic feature based on human pose estimation network to train a repetition counting and validation system, which is able to make detection of human activities and quickly identify the function of invalid repetition information. The results show that the system is able to accurately identify human movements and remove invalid repetitive information from them [4]. Liu et al. propose a new elliptical distribution coding method in order to help computers to accurately identify human movements. The method first describes the human skeleton by elliptical Gaussian coordinate coding, then measures the difference between the predicted heat map and the ground truth heat map, and finally the human pose images for recognition. The results of the study show that the method has a good performance in both datasets of the experiment and can provide high recognition accuracy [5]. Vishwakarma proposes a method for recognizing human actions in videos that can be identified by deterministic actions, which uses a double transform of wavelets to perform feature extraction of human actions. The extracted feature is then recognized. The results show that the method has high recognition accuracy in different datasets [6]. Tian et al. argue that the key points of the human body under many images in the video may produce unreasonable prediction results from the human pose estimation method due to issues such as illumination, occlusion, etc. To address this problem, the team designed a new generative adversarial network to address the situation where some keypoints are not visible, but the model still has high recognition accuracy. The model consists of two components, namely a cascaded feature network and a graph structure network. The results show that the model has excellent recognition accuracy [7]. Zhang et al. found that existing 3D human pose estimation methods focus on overall joint error reduction, which leads to large errors in endpoint and bone length. To address this problem, the group proposed a human structure-aware network that can extract feature data from existing 2D joints to repair the positions of 3D joint points. The results show that this method can effectively reduce the error between endpoint and bone length, resulting in a high improvement in recognition accuracy [8].

Ht et al. found that traditional human action recognition uses manual feature from traditional classifiers and is unable to make recognition of complex human actions using advanced spatio-temporal feature. To address this problem, the research team proposed a coding technique that converts poses into feature images, extracts high-level feature from the feature images and feeds them into a feature recognition system for recognition. The results show that the method is able to recognize human actions with high recognition accuracy [9]. Silva and Marana argue that existing human pose extraction uses straight lines to represent body parts in a two-dimensional human model. The team proposes an improved method based on existing human pose extraction, which maps each segment of a 2D pose to a point to extract spatial feature. The results of the study indicate that the method is effective in improving the recognition rate [10].

Table 1: Literature review

Study	Method	Application	Key Findings	Performance Comparison	References
Liu L et al.	Dual-view 3D pose estimation without camera parameters	Human pose estimation in dual views	Extracts joint coordinates from dual 2D images, inputs to 3D regression network	High accuracy in human pose recognition	[3]
Ferreira B et al.	Skeleton and deep semantic feature training system	Human activity detection and filtering	Detects activities and removes redundant repetitions	Accurate recognition with effective redundancy filtering	[4]
Liu H et al.	Elliptical Gaussian coordinate encoding	Action recognition in skeletal models	Uses heatmap differences for precise pose identification	High recognition accuracy on various datasets	[5]
Vishwakarma	Dual-wavelet	Human action	Extracts motion	Consistently high	[6]

D K	transformation	recognition in videos	feature using wavelet transform	accuracy across datasets	
Tian L et al.	Generative Adversarial Network (GAN)	Pose estimation with occlusion	Cascade and graph-based networks handle lighting and occlusion	High accuracy even with occluded keypoints	[7]
Zhang X et al.	Structure-aware network	3D joint correction in skeletal models	Reduces endpoint and bone length errors	Enhanced accuracy with reduced joint errors	[8]
Ht A et al.	Pose encoding to feature images for high-level feature extraction	Complex human behavior recognition	Converts pose to feature images for advanced feature recognition	High accuracy in complex activity recognition	[9]
Silva V et al.	Spatial feature extraction from mapped pose segments	2D human pose representation improvement	Maps 2D segments to extract spatial feature	Improved recognition rates	[10]

In summary shown in Table 1, many scholars have conducted research in the field of human pose recognition and achieved significant results, but there are still some limitations. Firstly, many methods rely on multi view inputs or high-quality data, and the recognition accuracy may decrease in single view or complex backgrounds. Secondly, encoding methods based on skeleton or feature images have limited performance in dealing with large occlusions or complex non repetitive actions. Some methods have high computational complexity and are not user-friendly for real-time applications, and models such as generative adversarial networks rely heavily on training data, increasing the complexity of model construction and training. In addition, information loss during the encoding process may affect recognition performance, especially in situations where there are rich pose details or diverse pose changes, limiting the applicability and accuracy of these methods. The deep residual network combined with the improved algorithm of Involution is used for feature extraction, creating a human skeleton network to recognize and classify human actions.

3 Martial arts movement recognition based on human posture estimation

With the development of the Internet, human body recognition technology has been vigorously developed and is widely used in motion analysis, video surveillance and other fields. In this study, Involution's feature extraction network is first proposed for lightweight human pose estimation, which is combined with existing human pose estimation models to recognize human pose. The extracted feature is then fed into a longand short term memory neural network.

3.1 Involution feature extraction network based human pose recognition

In the field of computer imaging, the main indicator of the strength of a neural network's performance is the strength of its feature extraction performance. By analysing the existing convolutional kernels, two drawbacks are found, one is that the perceptual field is difficult to capture feature dependencies over long distances due to the limitation of the convolutional kernel size.

The other is that the information between channels is rather complex and redundant. To solve this problem, this research proposes a new neural network operator Involution to assist feature extraction [11]. Involution is spatially specific, spatially specific in that it increases the receptive field by increasing the size of the convolution kernel, and channel invariant in terms of channels. Channel invariance allows neural networks to share in terms of channel dimensions, thus solving the problem of complex redundancy of information between channels. The main function of Involution is the reallocation of arithmetic power, which allows the computer to perform optimally.

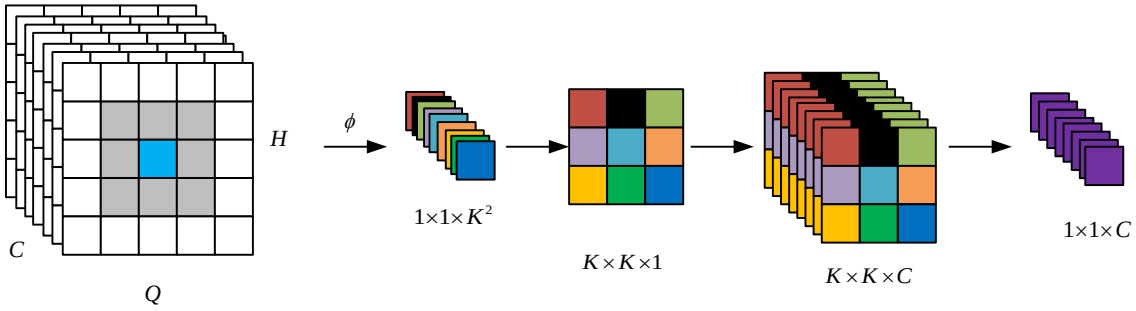


Figure 1: Generate convolutional kernel diagram

Figure 1 shows the process of generating a convolution kernel by Involution. Firstly, a multi-channel feature map is input and the feature vector of any point in the feature map is selected. Multiplying this kernel with the feature vector adjacent to the point gives the $K\times K\times C$ feature map, and finally the $K\times K\times C$ kernel is superimposed to obtain the final output feature map, with Involution generating different kernels for different locations and sharing a single kernel at the same location on the channel [12]. The traditional Convolution kernel counts and Involution counts are shown in Equation (1).

$$\begin{cases} 1\times1\times C_0\times C_i\times K\times K \\ H\times Q\times K\times K\times G \end{cases} \tag{1}$$

In Equation (1), 1×1 denotes the convolution kernel shared at $H\times Q$ pixel points, C_0 denotes the number of channels in the output, C_i denotes the number of channels in the input, K denotes the size of the convolution kernel, and G denotes the number of groupings. The number of channels is usually larger, the number of groups is usually much smaller than the number of channels, and the size of the Involution convolution kernel does not have a number of channels, so the ability to capture long distance feature can be enhanced by increasing the size of the convolution kernel. Involution is able to increase the accuracy of the model by this method while reducing the number of model parameters and the amount of computation [13].

This research uses a deep residual network combined with an Involution modified algorithm for feature extraction. The neuron learning feature maps of the general neural network and ResNet are shown in Figure 2.

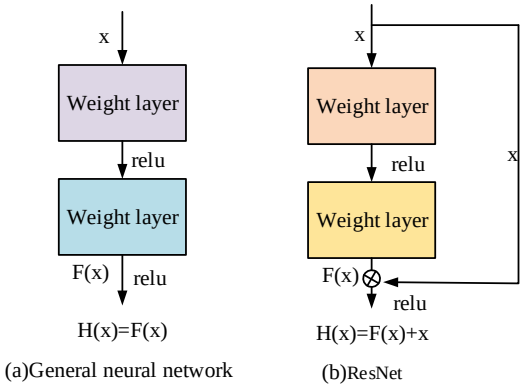


Figure 2: Neuron learning feature map

Figure 2(a) shows the process of learning feature in the fully-connected layer of a general neural network, which can be seen to be learning directly on the mapping between input and output. Figure 2(b) represents the process of learning feature in the fully-connected layer of ResNet, which can be seen to learn the residuals between the input and output. The InNet unit still has the same structure as the ResNet unit, with three convolutional layers in series, the first layer still reduces the dimension of the input channel, and the second layer uses the convolutional kernel generated by Involution to replace the original the second layer uses the convolution kernel generated by Involution to replace the original convolution kernel. The third layer is to expand the reduced-dimensional feature to the desired size. This improvement improves the feature extraction capability of InNet and also reduces the number of parameters and computational effort [14]. Convolution and Involution are shown in Equation (2).

$$\begin{cases} K^2C^2 \\ \dfrac{C^2+K^2GC}{r} \\ HQ\times K^2C^2 \\ HQ\times\dfrac{C^2+K^2GC}{r}+HQ\times K^2C \end{cases} \tag{2}$$

Equation (2) shows the number of parameters for Convolution and Involution and the amount of calculation for Convolution and Involution. Where H is the height of the input feature map, Q is the width of the input feature map, C is the number of input feature map channels, and r is the channel reduction ratio. The Involution Pose Estimation Net (IPEN) is used as the basis for extracting feature for the convolution kernel by Involution as shown in Figure 3.

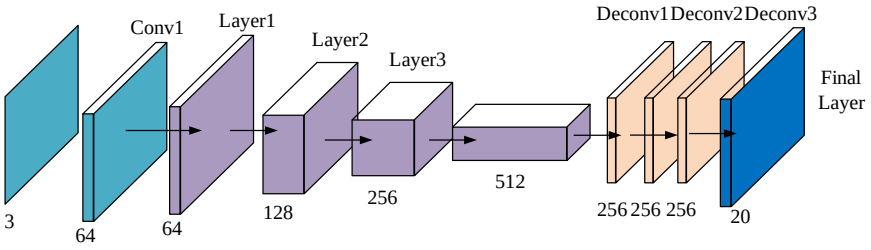


Figure 3: Convolutional kernel feature extraction graph

As shown in Figure 3, firstly, the input is a 3-channel image, and after passing through the first convolutional layer Conv1, the number of channels increases to 64. Next, after passing through three consecutive convolutional layers, Layer1, Layer2, and Layer3, the number of channels in the feature map increases to 128, 256, and 512, respectively. After completing the three convolutional layers, it enters the deconvolution stage (Deconv1, Deconv2, and Deconv3). In the deconvolution stage, each deconvolution layer gradually reduces the number of channels in the feature map from 512 to 256, resulting in a final output of 20 channels. The human pose recognition network uses InNet as the feature extraction network of the recognition network, and after expanding it by ordinary convolutional layers, Involution is used to extract feature information from the image, and the nodes are obtained by three convolutional layers that act as regressors. The metrics used to evaluate the model are Object Keypoint Similarity metrics, as shown in Equation (3).

$$Oks_p=\frac{\sum_l \exp\left\{-d_{pl}^2/2S_p^2\sigma_I^2\right\}\delta(v_{pl}=1)}{\sum_l \delta(v_{pi}=1)} \tag{3}$$

In Equation (3), p represents the person ID, l represents the number of keypoints S_p represents the current person's scale factor, v_{pl} represents whether the l th key point of the p th person is observable, d_{pl} represents the rated Euclidean distance between each person and each person's predicted joint point, σ_I represents the normalisation factor for the I th skeletal point, and δ represents the function that calculates the visible point [15].

3.2 Research on martial arts movement recognition based on human posture

Since traditional neural networks often fail to achieve the desired results when processing data with temporal information such as video and audio, Recurrent Neural Network (RNN) was introduced to process the data. Recurrent Neural Networks are capable of outputting information that is dependent on both present input and historical records. The structure of an RNN is shown in Figure 4.

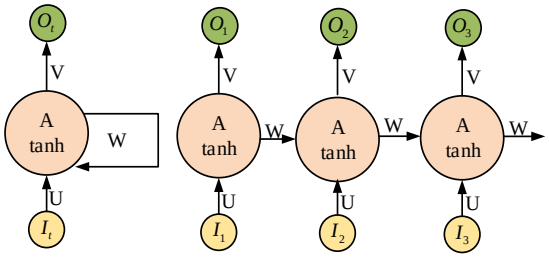


Figure 4: Structure diagram of recurrent neural network

Figure 4 represents the structure of an RNN, where A represents a single neural network unit, O_t represents the output at the time point, and I_t represents the input at the time point. U, V and W represent the different network weights respectively. The Long Short Term Memory Neural Network is an improvement on the RNN, which can process time series like the RNN and has a similar structure to the RNN, but the recurrent structure in the LSTM network is not the same as that of the RNN. The recurrent structure consists of three parts respectively three gate structures, one unit state and four neural network layers [16]. The structure of the LSTM neural network is shown in Figure 5.

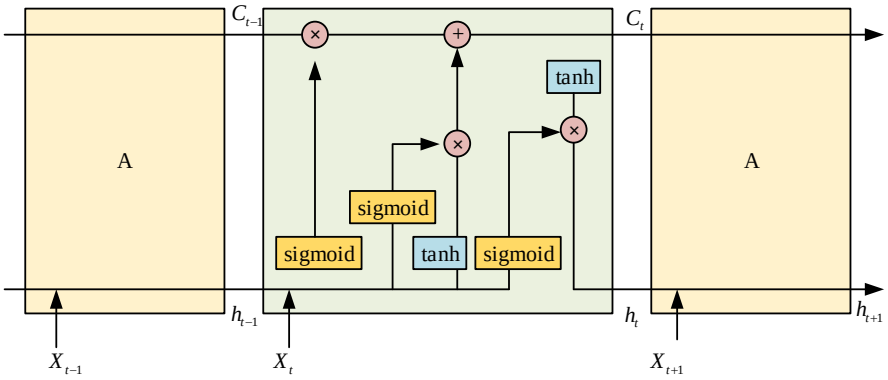


Figure 5: LSTM neural network structure diagram

As can be seen from Figure 5, the entire recurrent structure consists of a short-term memory module, a current memory module and a long-term memory module. In the current memory module, there are four neural network layers, three of which are single-layer sigmoid feed forward neural networks and one is a single-layer tanh feed forward neural network, and the LSTM neural network is mainly used to filter the feature information and determine the retention status through three gate structures: input gate, output gate and forgetting gate. Each gate structure is composed of a vector operation and a sigmoid neural network layer. Classify human joints using a human pose recognition network combined with human joint data, as shown in Figure 6.

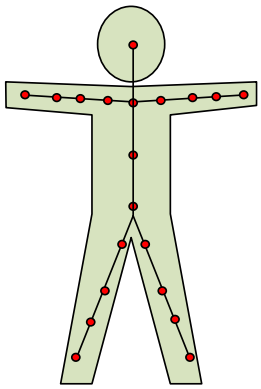


Figure 6: Skeleton division diagram of the human body

Figure 6 shows the division of the human skeleton. The importance of different bones to the human body varies. If bone joints are divided into 20 joints according to their importance, a sequence of human posture can be represented by equation (4) [17].

$$\begin{cases} S = \{K_1, K_2, \ldots, K_t\} \\ K_t^j = (x_j, y_j, z_j), 1 < j < M \end{cases} \tag{4}$$

In Equation (4), S represents the sequence of human skeletal articulation points, K_t represents the skeleton at t, M represents the number of articulation points, (x, y, z) represents the coordinates of the articulation points, and j represents the j th articulation point in the skeleton at t. The state of the human skeleton at each moment is coded into a network, and the skeleton joints at each moment change with time [18]. Define the interaction network of articulation points at different moments in time as shown in Equation (5).

$$SAN_t = (V_t, E_t) \tag{5}$$

In Equation (5), V_t denotes the set of vertices in the network at the moment of t and E_t denotes the set of edges in the network at the moment of t. For the skeleton state at the same moment, the joints are connected to each other and the relationship between the joints is expressed by calculating the Euclidean distance between each joint as shown in Equation (6).

$$d(i, j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{6}$$

In Equation (6), i is any one of the joints except j. Since the human body completes the action, it is not determined by individual joints, but by the overall coordination of the human body, just using the Euclidean distance cannot highlight the relationship between each joint well, so the human body is divided into different five parts, and different weight coefficients are set according to different parts as shown in Equation (7).

$$w(i, j)\begin{cases} d(i, j) \times a_1, 1 \le i, j \le 4 \\ d(i, j) \times a_2, 5 \le i, j \le 8 \\ d(i, j) \times a_3, 9 \le i, j \le 12 \\ d(i, j) \times a_4, 13 \le i, j \le 16 \\ d(i, j) \times a_5, 17 \le i, j \le 20 \end{cases} \tag{7}$$

In Equation (7), a_1 and a_2 represent the weight coefficients of the left and right arms, a_3 and a_4 represent the weight coefficients of the left and right leg parts, a_5 represents the weight coefficient of the torso part. After the skeleton nodes were constructed, the feature information of the image was extracted by CNN local convolution. The extracted feature data is then fed into the LSTM for processing, and the feature data is

filtered and judged by various gates. Each LSTM cell has an input gate, an output gate and an oblivion gate, and the input gate is calculated as shown in Equation (8).

$$i_t = g(W_{x^i} x_t + W_{h^i} h_{t-1} + b_i) \tag{8}$$

Equation (8) represents the input gate, x_t represents the input value of the network at the current time and h_{t-1} represents the output value of the network at the previous time. b_i denotes the input gate constant parameter. The output gate is calculated as shown in Equation (9).

$$f_t = g(W_{x^f} x_t + W_{h^f} h_{t-1} + b_f) \tag{9}$$

Equation (9) represents the output gate, x_t represents the input value of the network at the current moment, and b_i represents the output gate constant parameter. The formula for the forgetting gate is shown in Equation (10) [19].

$$o_t = g(W_{x^o} x_t + W_{h^o} h_{t-1} + b_o) \tag{10}$$

Equation (10) represents the forgetting gate and h_{t-1} represents the output value of the network at the previous moment. b_o denotes the constant parameter of the forgetting gate. In the IPN recognition technique for the skeleton, the human skeleton at each moment is encoded as a network, and the weights of the edges are calculated based on the distance between any two joints in the network as shown in Equation (11).

$$d(i,j) = 1 \Big/ \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{11}$$

In Equation (11), i and j denote the nodes in the network and (x, y, z) denotes the 3D coordinates of the node, it can be seen that the weights are expressed as the reciprocal of the Euclidean distance. In order to represent the transformation of the nodes in the network in time, metrics such as proximity centrality are introduced for evaluation, as shown in Equation (12).

$$CC_i = \frac{N-1}{\sum_{j \in U, j \neq i} d(i,j)} \tag{12}$$

In Equation (12), N is the number of nodes in the network and U is the set of all nodes in the network. Proximity centrality indicates how close the node is to each of the other nodes in the network; the closer the node is, the greater its closeness centrality, but the same node will change over time and its centrality will change as well. The eigencentricity vector of the network nodes is analysed as shown in Equation (13).

$$EC_i = c \sum_{j=1}^{N} A_{ij} EC_i \tag{13}$$

In Equation (13), EC_i represents the eigenvector centrality and sets its initial value to 1, represents the adjacency matrix in the network, A_{ij} represents the connection between nodes i and j, and the initial vector of EC_i is cyclically multiplied with A to obtain the value of EC_i. The stability of the network is usually assessed by the average degree as shown in Equation (14) [20-21].

$$K_i = \frac{\sum_{i=1}^{N} K_i}{N} \tag{14}$$

In Equation (14), K_i represents the weighted degree of the node i. The topological properties of the network nodes are combined with the topological properties of the network to represent the entire skeleton of the action network. A sample skeleton of all actions is shown in Equation (15).

$$Y_{input} = [\theta_1, \theta_2, \theta_3, \ldots, \theta_{u-1}, \theta_u] \tag{15}$$

In Equation (15), Y_{input} denotes the input to the LSTM and u denotes the number of samples. θ_u denotes the feature vector in u. The samples are classified by this method to identify human actions. The process of this model is as follows. Firstly, the Involution operation dynamically generates convolutional kernels that adapt to feature maps, enhancing the ability to capture long-distance feature and reducing information redundancy between channels. By combining this network structure with a deep residual network, an improved InNet was formed, which can efficiently extract feature and reduce the number of model parameters. Subsequently, LSTM was used to process time series data and analyze the dynamic changes of human joint points. Joint points are classified according to their importance, and Euclidean distance is calculated to describe the relationship between joints. The sensitivity of action recognition is improved by setting weights for different parts.

4 Performance analysis of martial arts movement recognition based on human posture estimation

The first section of this chapter analyses Involution's downsampling capability and then analyses the accuracy of the model under different dataset sizes to determine the best data size to calculate its feature extraction time. The second section provides an analysis of the introduction of

LSTM networks to compare the models under different algorithms.

4.1 Performance analysis of human pose recognition based on involution feature extraction network

To verify the performance of this feature extraction network using InNet as the recognition network, InNet was compared with ResNet. The CPU used in this experiment is Intel(R) Xeon® Gold6226@2.7GHz, the GPU used is NVIDIA GeForce Tesla V100S, and the memory is 32 GB. The learning rate of the model is set to 0.001 and decays by 0.1 every 10 epochs to gradually reduce the learning rate. The batch size is 32 to ensure efficient memory usage during the training process. The optimizer uses Adam because of its adaptive learning rate feature, which can handle sparse gradient problems. The loss function uses cross entropy loss, which is suitable for multi class classification tasks. Using L2 regularization, the weight decay parameter is set to 0.0001 to reduce the risk of overfitting. In terms of data augmentation, methods such as random cropping, rotation, and translation are applied during training to improve the model's generalization ability. In terms of feature extraction network, the number of layers in the Involution network is set to 5, and the number of channels is set to 128 to evaluate performance. In terms of LSTM configuration, the number of units is set to 256 to better capture time series feature. The training cycle is set to 100 epochs, using 20% of the data as the validation set to monitor model performance and prevent overfitting. When the number of layers in the network is small, Involution has less compression power, but the accuracy is improved. As the number of layers increases, Involution has a good improvement in compression, but with some loss of accuracy.

Table 2: Argument reduction capability of revolution

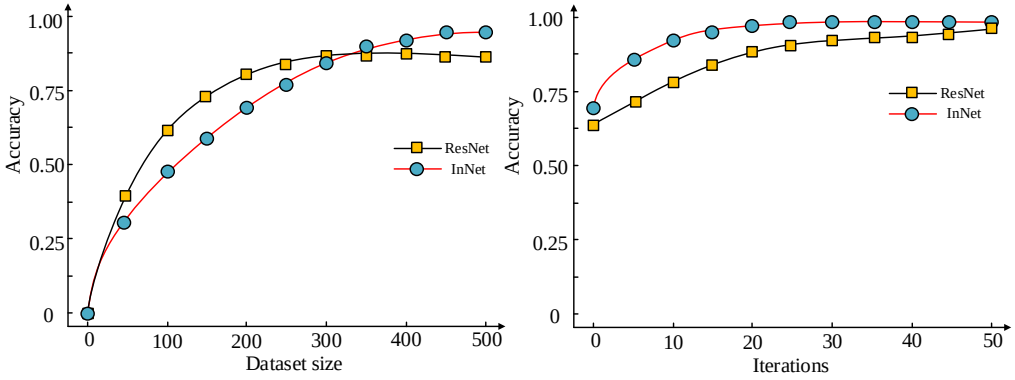
Method	Input size	Param	FLOPs
ResNet-Q32	256 x 192	28.4M	7.2G
	384 x 288	28.4M	16.5G
ResNet-Q48	256 x 192	63.9M	14.7G
	384 x 288	63.9M	32.5G
InNet-Q32	256 x 192	17.1M	4.7G
	384 x 288	17.1M	10.1G
InNet-Q48	256 x 192	38.7M	7.9G
	384 x 288	38.7M	20.4G

In Table 2, Q32 indicates that the number of channels for each convolutional layer is set to 32, and Q48 indicates that the number of channels for each convolutional layer is set to 48. Table 2shows the table of Involution's degree-reducing capacity, InNet for using Involution instead of Convolution, from the table it can be seen that ResNet's Param is 28.4M and 63.9M, InNet's Param is 17.1M and 38.7M, ResNet under different methods, different sizes of The FLOPs of different sizes for ResNet were 7.2G, 16.5G, 14.7G and 32.5G, respectively, and the FLOPs of different sizes for InNet were 4.7G, 10.1G, 7.9G and 20.4G, respectively, under different methods. The experimental results indicated that the InNet method using Involution instead of Convolution reduced the number of parameters and computation by about 40%, indicating that Involution has good capability of reducing parameters. Compare the computational complexity of different methods.

As shown in Table 3. InNet reduces its dependence on large convolution kernels through Involution, while ResNet relies on deep residual structures, and LSTM uses recursive structures to process time series. InNet has relatively low memory usage because it uses smaller feature maps, while ResNet requires more memory due to its deep structure. LSTM also increases memory requirements when processing long sequences. The latency of InNet is moderate, influenced by input size and sequence length. ResNet and LSTM can cause high latency when processing large inputs or long sequences.

Table 3: Comparison of computational complexity

Model	Processing Flow	Memory Usage	Latency
InNet	Utilizes Involution instead of convolution for feature extraction, followed by LSTM for sequence analysis	Low to moderate, depending on feature map size and number of channels	Moderate, influenced by input feature map size and time steps
ResNet	Employs multiple residual blocks for feature extraction, followed by fully connected layers for classification	High, especially in deeper networks	High, particularly when processing large input sizes
LSTM	Uses a recurrent structure to handle sequence data	High, due to the need to store hidden states and input sequences	High, especially with long sequences and multiple feature

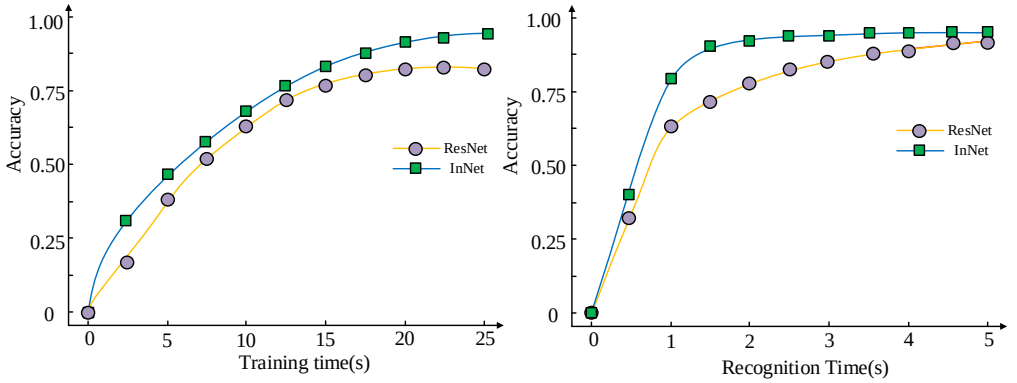


(a)The relationship between dataset size and accuracy (b)The relationship between the number of iterations and accuracy

Figure 7: Model accuracy of RseNet and InNet

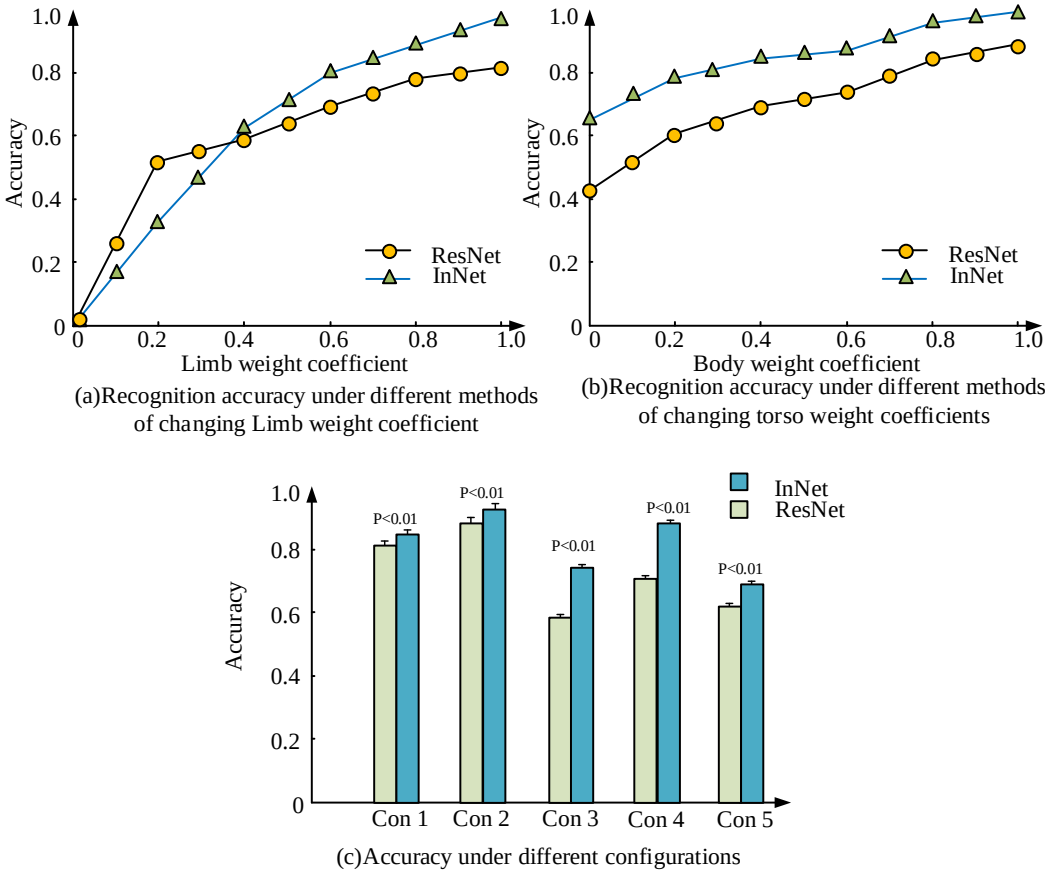
As can be seen from Figure 7(a), the extraction performance of both methods is better when the dataset is larger and contains more species. Since the number of Involution parameters and the amount of computation in InNet is less compared to that of the traditional Convolution in ResNet, the accuracy of InNet is still increasing when the size of the dataset reaches a certain amount, ResNet has levelled off. From Figure 7(b), it can be seen that with the selected dataset size, InNet has been able to achieve the best recognition performance with a small number of iterations, and ResNet has not yet achieved the best performance with the number of iterations where InNet's performance has reached its best, and reaches a point where when the performance no longer changes, it is still lower than InNet's performance. It can be seen that InNet has good performance in feature extraction. Judging the goodness of a model cannot only focus on its accuracy, its training time and recognition time is still an important indicator as shown in Figure 8.

Figure 8(a) shows the change in model performance for both methods as the training time increases. It can be seen that the training time for InNet is a little longer than that for ResNet, the situation is due to the fact that InNet uses a larger dataset during training and only a large enough dataset can satisfy InNet to allow it to train to achieve the best performance. Figure 8(b) shows the change in model accuracy as the recognition time increases for both methods. It can be seen that InNet is able to use a small amount of time to achieve the best recognition accuracy on images when recognizing. The results of the study indicate that the training time for InNet is slightly longer but within acceptable limits and that the overall performance of InNet is better than that of ResNet.



(a)The relationship between training time and accuracy (b)The relationship between recognition time and accuracy

Figure 8: Analysis of training time and recognition time for two models



(a)Recognition accuracy under different methods of changing Limb weight coefficient

(b)Recognition accuracy under different methods of changing torso weight coefficients

(c)Accuracy under different configurations

Figure 9: Model accuracy under different weight coefficients and configurations

4.2 Performance testing of a human posture-based martial arts movement recognition model

A selection of martial arts moves is identified, including the lunge punch, punch and pop kick, horse stance punch, horse stance frame punch and top stomp kick. The five movements are renamed as Movement 1, Movement 2, Movement 3, Movement 4 and Movement 5 respectively. The weighting coefficients for the left and right arms, the left and right legs and the torso were assigned to compare the influence of each part of the skeleton on the recognition system. This is shown in Figure 9.

Figure 9 (a) shows the recognition accuracy for different methods with a value of 1 for the torso weighting factor a_3 and changing the weighting factor for the extremities. Figure 9(b) shows the recognition accuracy for different methods of varying the torso weighting coefficients when the values of the limb weighting coefficients a_1 and a_2 are set to 1. It can be seen that when the limb weighting coefficients are changed, the accuracy rate increases significantly with the increase of limb weighting coefficients and stabilises when the limb weighting coefficients reach 0.8. When the weighting factor of the limbs was changed, the change in accuracy was minimal when the weighting factor of the torso was changed. The experimental results show that the influence of the limbs on the accuracy is greater than the influence of the torso on the accuracy. Figure 9 (c) Accuracy rates for five different weighting factors not chosen named configurations 1 to 5 respectively, at different weighting factors. Configuration 1 has a torso weight coefficient of 0.2 and limbs weight coefficient of 0.8, Configuration 2 has a torso weight coefficient of 0.6 and limbs weight coefficient of 0.8, Configuration 3 has a torso weight coefficient of 0.6 and limbs weight coefficient of 1.0. Configuration 4 has a torso weight coefficient of 0.4 and limbs weight coefficient of 0.8. Configuration 5 has a torso weight coefficient of 0.4 and limbs weight coefficient of 0.6. It can be seen that the accuracy of recognition is maximised when the torso weighting factor is 0.6 and the limb weighting factor is 0.8($P<0.01$). A comparison of recognition for different actions at different weighting factors is shown in Figure 10.

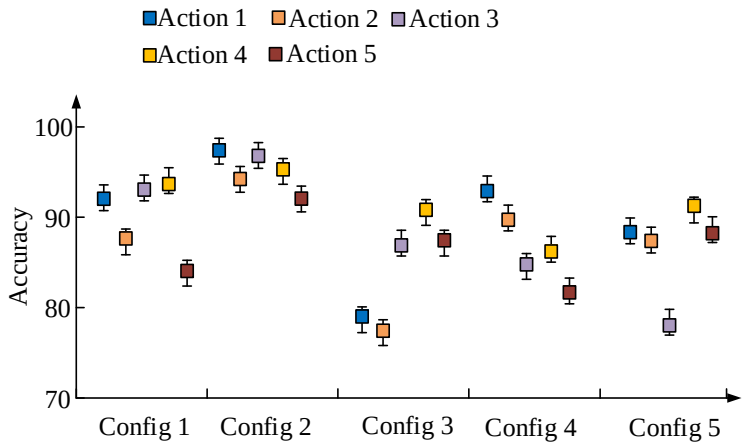


Figure 10: Accuracy of five different actions under five weight coefficients

Table 4: Recognition accuracy of different algorithms on datasets

Method	Eigen	STOP	DMM & HOG	Actionlet	JAS & HOG2	InNet-LSTM
Data 1 Accuracy (%)	81.3	82.5	85.3	87.6	83.5	90.6
Data2 Accuracy (%)	76.5	81.4	86.2	88.6	87.9	93.6

From Figure 10, the accuracy of five different actions with five weighting factors, it can be seen that the action with weighting configuration 2 has the highest average accuracy and has a more stable performance. The other weighting configurations all have large fluctuations in accuracy and show unstable performance. Considering both stability and accuracy, the weights used to construct the human skeleton were set to a torso weighting factor of 0.6 and an extremity weighting factor of 0.8. Different methods were introduced to compare with the method used in this study, and the MSR Action 3D dataset was chosen for this experiment. This dataset contains 6000 images specifically designed for human action recognition tasks, covering multiple explicit action categories including walking, running, jumping, and sitting. The image size of each sample is 640x480 pixels, ensuring clarity and detail. The sample distribution of these action categories is uneven, with more samples for walking and running, and relatively fewer samples for jumping and sitting, which may affect the training effectiveness and performance of the model. Each image is equipped with clear labels to indicate the corresponding action category, ensuring the accuracy of the training data. In addition, the dataset generates additional samples through data augmentation techniques, including random rotation, flipping, and scaling, to enhance the model's generalization ability. Different algorithms were used to divide the dataset into Dataset 1 and Dataset 2, and the recognition accuracy on different datasets is shown in Table 4.

According to Table 4, in dataset 1, the accuracy of Eigen method is 81.3%, STOP method is 82.5%, DMM&HOG method is 85.3%, Actionlet method is 87.6%, and JAS&HOG2 method is 83.5%. The accuracy of the InNet LSTM method is 90.6%. In dataset 2, the accuracy of each method is higher than in dataset 1. The accuracy of the proposed methods in this study was higher than the other methods. To further validate the accuracy of the InNet-LSTM method, the accuracy of the different methods was verified under different dataset sizes. This is shown in Figure (11).

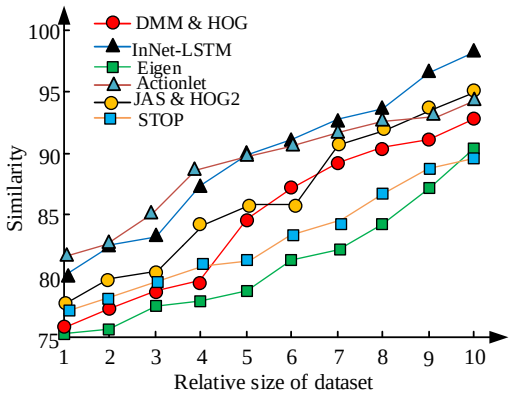


Figure 11 Accuracy of different methods under different dataset sizes

The accuracy of InNet-LSTM is lower than that of Actionlet method when the dataset is small, but when the dataset increases to a certain level, the accuracy of InNet-LSTM is greater than that of other methods.

5 Discussion

Human motion recognition relies on video frame by frame decomposition and manually designing motion feature to achieve recognition. The martial arts action recognition system based on Involution feature extraction network and LSTM proposed in the study optimizes recognition accuracy and efficiency by reducing the computational complexity of traditional convolutional networks. The experimental results show that compared with traditional convolutional networks such as ResNet, Involution significantly improves accuracy while reducing the number of parameters, especially on datasets of different sizes, with an average increase of 5% in object keypoint similarity and 8% in accuracy in the test set. This is due to the advantage of LSTM in time series modeling, which enables the system to better understand the dynamic changes in action sequences, especially achieving an accuracy gain of about 15% in complex martial arts action recognition. The innovation of InNet LSTM lies in using Involution instead of traditional convolution to achieve lightweight and efficient feature extraction, and combining LSTM for temporal modeling to capture motion dynamics. This method outperforms ResNet in accuracy, resource utilization, and computation time, and is suitable for martial arts action recognition and other dynamic scenarios. It has broad applicability and efficient real-time processing capabilities. However, there are still limitations when dealing with unstructured random actions. Due to the limitations of existing equipment, higher performance hardware can be introduced in the future to optimize training speed and expand the dataset size to enhance the system's generalization ability.

6 Conclusion

In response to the problem of manually designing motion feature for recognition, which consumes energy and has very low recognition efficiency, research is conducted on improving human pose estimation based on deep learning. Firstly, Involution is proposed as a feature extraction network for light weighting of human pose estimation, and each joint point of the human body is labelled and classified separately. The experimental results show that the InNet method, which uses Involution instead of Convolution, decreases the number of parameters and the computational effort by about 40%. Comparing this method with other methods, the accuracy of the Eigen method is 81.3%, the STOP method is 82.5%, the DMM & HOG method is 85.3%, the Actionlet method is 87.6% and the JAS & HOG2 method is 83.5%. The accuracy of the InNet-LSTM method was 90.6%. It can be seen that the method proposed in this study has a high performance. However, there are still shortcomings in this study. When constructing a human skeleton model, the weights between the joints are determined by the distance between the joints, and the evaluation indicators are too single. And the research was conducted in a laboratory environment. Future research is considering using more indicators to construct human skeleton models and applying them to practical applications to test the performance of the models.

References

[1] S. Yan, Y. Xiong, D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. 2018, 32(1): 56-72. https://doi.org/10.1609/aaai.v32i1.12328.

[2] W. Luo, W. Liu, S. Gao. Normal graph: Spatial temporal graph convolutional networks-based prediction network for skeleton based video anomaly detection. Neurocomputing, 2021, 444(15): 332-337. https://doi.org/10.1016/j.neucom.2020.08.085.

[3] L. Liu, L. Yang, W. Chen, X Gao. Dual-View 3D human pose estimation without camera parameters for action recognition. IET Image Processing, 2021, 15(14): 3433-3440. https://doi.org/10.1049/ipr2.12277.

[4] B. Ferreira, P. M. Ferreira, G. Pinheiro, N. Figueiredo, F. Carvalho, P. Menezes, J. Batista. Deep learning approaches for workout repetition counting and validation. Pattern Recognition Letters, 2021, 151(12):259-266. https://doi.org/10.1016/j.patrec.2021.09.015

[5] H. Liu, Y. Chen, W. Zhao, S. Zhang, Z. Zhang. Human pose recognition via adaptive distribution encoding for action perception in the self-regulated learning process. Infrared Physics and Technology, 2021, 114(5): 1036-1045. https://doi.org/10.1016/j.infrared.2021.103660.

[6] D. K. Vishwakarma. A two-fold transformation model for human action recognition using decisive pose. Cognitive Systems Research, 2020, 61(6): 1-13. https://doi.org/10.1016/j.cogsys.2019.12.001.

[7] L. Tian, G. Liang, P. Wang, C. Shen. An adversarial human pose estimation network injected with graph structure. Pattern Recognition, 2021, 115(2):31-40. https://doi.org/10.1016/j.patcog.2021.107863.

[8] X. Zhang, Z. Tang, J. Hou, Y. Hao. 3D human pose estimation via human structure-aware fully connected network. Pattern Recognition Letters, 2019, 125(5): 404-410. https://doi.org/10.1016/j.patrec.2019.04.007.

[9] A. Ht, C. Chh, B. Ttn, B. Dska. Image representation of pose -transition feature for 3D skeleton-based action recognition. Information Sciences, 2020, 513(3): 112-126. https://doi.org/10.1016/j.ins.2019.12.063.

[10] V. Silva, N. Marana. Human action recognition in videos based on spatiotemporal features and bag-of-poses. Applied Soft Computing, 2020, 95(1) 84-93. https://doi.org/10.1016/j.asoc.2020.106513.

[11] B. Sun, D. Kong, S. Wang, L. Wang, B. Yin. Joint transferable dictionary learning and view adaptation for multi-view human action recognition, ACM Transactions on Knowledge Discovery from Data (TKDD), 2021, 2-55. ttps://doi.org/10.1145/3418897.

[12] L. Yu, L. Tian, Q. Du, J. Bhutto. Multi-stream adaptive spatial-temporal attention graph convolutional network for skeleton-based action recognition. IET Computer Vision, 2022, 162(2): 143-158. https://doi.org/10.1049/cvi2.12058.

[13] M. S. Alsawadi, M. Rio. Skeleton split strategies for spatial temporal graph convolution networks, Computers. Materials and Continuum, 2022, 1(6):4643-4658. https://doi.org/10.32604/cmc.2022.028266.

[14] Y. Hou, L. Wang, R. Sun, Y. Zhang, M. Gu, Y. Zhu, Y. Tong, X. Liu, X. Wang, J. Xia, Y. Hu, L. Wei, C. Yang, M. Chen. Crack-across-pore enabled high-performance flexible pressure sensors for deep neural network enhanced sensing and human action recognition. ACS NANO, 2022, 16(5): 8358-8369. https://doi.org/10.1021/acsnano.2c02609.

[15] A. Gharahdaghi, F. Razzazi, A. Amini. A non-linear mapping representing human action recognition under missing modality problem in video data. Measurement, 2021, 186(3): 1123-1133. https://doi.org/10.1016/j.measurement.2020.112123.

[16] W. Xu, M. Wu, J. Zhu, M. Zhou. Multi-scale skeleton adaptive weighted GCN for skeleton-based human action recognition in IoT. Applied Soft Computing, 2021, 104(3):1568-1579. https://doi.org/10.1016/j.asoc.2021.107596.

[17] H. B. Naeem, F. Murtaza, M. H. Yousaf, S. A. Velastin. T-VLAD: Temporal vector of locally aggregated descriptor for multiview human action recognition. Pattern Recognition Letters, 2021, 148(8): 22-28. https://doi.org/10.1016/j.patrec.2021.06.012.

[18] M. Yang. Research on vehicle automatic driving target perception technology based on improved MSRPN algorithm. Journal of Computational and Cognitive Engineering, 2022, 1(3): 147-151. https://doi.org/10.47852/bonviewJCCE20514

[19] Y. Lin, W. Chi, W. Sun, S. Liu, D. Fan. Human action recognition algorithm based on improved resnet and skeletal keypoints in single image. Mathematical Problems in Engineering, 2020, 2020(12): 1-12. https://doi.org/10.1155/2020/8827468.

[20] F. Daneshdoost, M. Hajiaghaei-Keshteli, R. Sahin. R. Tabu search based hybrid meta-heuristic approaches for schedule-based production cost minimization problem for the case of cable manufacturing systems. Informatica, 2022, 33(3): 499-522. https://doi.org/10.15388/21-INFOR471

[21] G. Dzemyda, M. Sabaliauskas, V. Medvedev. Geometric MDS performance for large data dimensionality reduction and visualization. Informatica, 2022, 33(2):299-320. https://doi.org/10.15388/22-infor491.