Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Research on optimizing vocal singing posture based on image recognition technology

Ting Huang[1], Yurun Li[2,✉]

[1] *Music College, Sichuan University of Science & Engineering, Zigong, Sichuan, 643000, China*

[2] *Music and Dance School, West China Normal University, Nanchong, Sichuan, 637000, China*

ABSTRACT

Vocal singing is a key art form of many stage singing arts, specifically including acting and singing. The study firstly is to introduce the detection principle of YOLOv5 target detection algorithm, on the basis of which the original YOLOv5 algorithm is improved by reconstructing the backbone network with the use of SENet and GhostNet, then the original YOLOv5 algorithm and the improved YOLOv5 algorithm are tested for comparison, and the test results show that on the target detection dataset Precision, Recall and mAP values reach 85.75%, 72.34% and 78.48% respectively, which are all improved compared with the original algorithm. Secondly, a high-resolution human posture estimation network incorporating multiple attention mechanisms is proposed to further extract multi-scale feature information and global feature information, and validated on publicly available datasets, CDLNet has an AP value of 0.662 and an AR value of 0.731 on the vocal singing posture estimation dataset, comparing with similar methods, the method has an MPJPE in Human3.6M The lowest is 44.6, which is suitable for use in vocal singing posture estimation in vocal singing scenarios. Finally, an action recognition model based on multi-granularity spatio-temporal graph convolutional neural network designed in this paper is used to analyze the singing gesture action recognition for singing action categories, and experiments show that the average recognition rate of MGstgcn can reach 86.5% on the HSiPu2 dataset, which meets the demand of vocal singing gesture action recognition analysis.

Keywords: target detection, gesture estimation, spatio-temporal graph convolution, action recognition, attention mechanism

1. Introduction

In the process of learning vocal music, it is perceived that good posture not only affects the state of breathing, but also coordinates the interactions between the vocal organs [1-3]. Only by ensuring correct singing posture can the vocal organs function in an overall coordinated and orderly manner.

✉ Corresponding author.
E-mail address: lyrpig@126.com (Y. Li).
Received 15 January 2024; Revised 10 March 2024; Accepted 30 November 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-230

On the contrary, wrong singing posture will bring serious adverse effects on vocal singing and directly affect the overall singing effect of the singer.

At the same time, singing posture is also the external manifestation of the performer's image, temperament and mental state [4-5]. The singer's posture during vocalization is closely related to the stage image to be presented. If the singer adopts the wrong posture to sing on stage, it will hinder or destroy the integrity of the whole singing process [6-8]. Especially for beginners with little singing experience, if they do not develop good singing posture habits at the beginning of the learning process, it will be extremely unfavorable to their future vocal studies. Correct posture in singing is natural and smooth and conforms to the overall singing state [9]. In the whole process of singing, the singer needs to maintain a positive and relaxed state, the singing state can not be overly tense, but not too loose, to be in the "loose but unremitting, tight but not rigid" in the best state. Correct singing posture is the first prerequisite for singing, can create favorable conditions for singing, the whole performance and singing has a positive role in promoting [10-11]. Establishing a scientific and correct singing posture will make the organs of vocalization in normal working condition, the breathing airflow will become smooth, and the breath will become even and powerful. On the contrary, wrong singing posture will lead to the whole singing can not be carried out smoothly. For example: in the standing center of gravity is not stable, lift the shoulders back to the neck, tilt the head, uneven pelvic force, etc., the limbs will show obvious incoordination and sense of tension, and will be directly in the singing voice response, which is manifested in the vocal powerlessness, high respiratory position, the phenomenon of shouting is more obvious, etc. [12]. In addition to the problem of the center of gravity, head movements also affect the musical quality of singing [13].

Therefore, in vocal music teaching, we need to fully recognize the importance of singing posture, pay attention to singing posture, develop good singing habits, help regulate the voice, better and faster mastery of more scientific singing skills [14-15].

For the vocal singing target recognition part, the backbone network of YOLOv5 target detection algorithm is reconstructed, and GhostNet and SENet are fused in the network to optimize its performance in terms of model volume and algorithmic accuracy, and experimental validation is carried out to achieve efficient and accurate image recognition of people's poses in vocal singing videos. A high-resolution singing pose estimation network incorporating multiple attention mechanisms is also proposed, using coordinate attention to construct a new module for substitution to enhance the feature extraction capability, and adding a multiscale attention module and a nonlocal attention module to further extract multiscale feature information and global feature information. Subsequently, a vocal singing gesture recognition model based on multi-granularity spatio-temporal graph convolutional neural network is designed, which contains three parts: encoder, decoder and classifier. Among them, the encoder helps the network to learn more stable motion features between coarse and fine granularity, and the decoder uses self-supervised training based on motion prediction head to help the network to extract motion features using unlabeled data for semi-supervised learning. Not only that, the network also achieves fully supervised learning with the encoder, decoder and classifier. Finally, the estimation and recognition algorithms for vocal singing gestures are tested on the self-constructed dataset separately.

2. Based on the improved YOLOv5 target detection algorithm design

2.1. Principle of YOLOv5 target detection algorithm

YOLOv5 is divided into four models: YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5x. In this paper, considering the size of the network model and the time cost of detection, YOLOv5s is chosen to construct the target detection model.The structure of YOLOv5 algorithm can be divided into four parts: Input, Backbone, Neck, and Prediction.Here, the network composition is introduced with the structure of YOLOv5s which is used in this system [16].

1) Input layer. YOLOv5 in the Input layer is utilizing Mosaic data enhancement, Adaptive Anchor Frame Calculation and Adaptive Image Scaling as three ways to preprocess the input image, which greatly enriches the dataset as well as reduces the performance required in the training phase.
2) Backbone layer. YOLOv5 is mainly composed of three modules, Focus, CSP and SPP, in the Backbone layer, which effectively extends the acquisition range of the feature extraction network and clearly distinguishes the most important contextual features.
3) Neck layer. YOLOv5 adopts the structure of FPN+PAN in the Neck layer. Where FPN passes the strong semantic features of the input feature map from top to bottom. But the structure only passes the strong semantic features and ignores the localization information. Therefore, PAN is to add a bottom-up channel behind the FPN to convey strong localization features.
4) Prediction layer. YOLOv5 in Prediction layer is mainly reflected in the two aspects of loss function calculation and non-extremely large value suppression.The loss function of YOLOv5 is initially calculated by GIoU.GIoU is used by YOLOv5 to calculate the loss of the bounding box during the training, which is in turn calculated by IoU, as shown in Equation (1).

$$GIoU = IoU - \frac{|A_C - U|}{A_C} \tag{1}$$

where A_c is the minimum closure area between the two frames, the predicted frame and the real frame, and U is the area of the region outside the two frames in A_c. However, GIoU is inaccurate in regression when the predicted box overlaps with the target box and cannot distinguish the relative positions of the two boxes when their center points coincide. Therefore, later YOLOv5 algorithm uses the more effective CIoU, which takes into account the overlapping area between the predicted and target boxes, the distance between the center points, and the corresponding aspect ratios of the two boxes, as shown in Equation (2).

$$CIoU = IoU - \frac{Dis_2^2}{Dis_c^2} - \alpha v \tag{2}$$

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{3}$$

Where, v is the aspect ratio consistency parameter, Dis_2 is the Euclidean distance between the target frame and the center of the prediction frame, Dis_c is the diagonal distance between the target frame and the smallest outer rectangle of the prediction frame, and α is the equilibrium parameter as shown in equation (3).

2.2. Improved YOLOv5 algorithm structure design

2.2.1. SENet. The three operations performed by the SE module for processing are as follows:
First, the SE module compresses the high-dimensional input features υ to one dimension by Squeeze (squeezing) method using global average pooling technique. The whole space can be encoded after utilizing the Squeeze method to obtain the global compressed feature volume of the current feature.The specific operation of Squeeze is shown in equation (4).

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1} \sum_{j=1} u_c(i, j) \tag{4}$$

The result of the Squeeze operation is then subjected to the Excitation operation, which has two fully connected layers. The first fully-connected layer in the figure is to reduce the dimensionality of the input, and then use a nonlinear activation function for the output. The second fully connected layer restores the dimensionality to the initial state. The two operations improve the generalization ability of the model and reduce the complexity of the model.The Excitation operation is shown in equation

(5).

$$s = F_{ex}(z,W) = \sigma(g(z,W)) = \sigma(W_2\delta(W_1 z)) \tag{5}$$

Where, σ denotes the Sigmoid function, δ denotes the ReLU function, $W_1 \in \mathbb{R}^{\frac{C}{r}\times C}, W_2 \in \mathbb{R}^{C\times\frac{C}{r}}$ and here r are parameters introduced to reduce the dimensionality of the fully connected layer.

Finally, the values obtained from each channel after Excitation and the input features U are subjected to the Scale (scaling) operation to obtain new features. F_{scale} The meaning of the operation is to act the weight values of the importance of each channel obtained from the previous operation on the corresponding channel respectively, so as to achieve the role of feature rescaling.The formula for the Scale operation is shown in equation (6).

$$x_c = F_{scale}(u_c, s_c) = s_c \cdot u_c \tag{6}$$

2.2.2. GhostNet. The Ghost module adds linear operations to the traditional convolutional operations to merge the generated Ghost feature maps with the eigenfeature maps and finally get the output results.

For a given input $X \in \mathbb{R}^{c\times h\times w}$, where c is the number of input channels, h is the height, and w is the width of the input, the operation procedure for generating the eigenfeature map by the conventional convolution operation is shown in equation (7).

$$Y = X * f + b \tag{7}$$

where $f \in \mathbb{R}^{c\times k\times k\times n}$ is the convolution kernel of this layer, $*$ is the convolution operation, b is the deviation term, and $Y \in \mathbb{R}^{h\times w\times n}$ refers to the eigenfeature maps of the n channels. In addition, h' and w' denote the height and width of the output data, respectively, and $k\times k$ is the kernel size of the convolution kernel. The computational effort (FLOPs) of this convolution operation is $n\cdot h'\cdot w'\cdot c\cdot k\cdot k$.

However, this convolution operation is performed with a large number n of convolution kernels and a large number of channels c, which results in a lot of redundancy in the output feature map. In order to reduce the generation of these redundant features, a single convolution can be used to generate an eigenfeature map of smaller size as shown in equation (8).

$$Y' = X * f' \tag{8}$$

where $f' \in \mathbb{R}^{c\times k\times k\times m}$ is the convolutional kernel used, $Y' \in \mathbb{R}^{h\times w'\times m}$ is the eigenfeature map of the m channels, $m \le n$ and the bias term is omitted for this operation.

Ideally, the Ghost module contains 1 constant transform and $n/s(s-1)$ linear operations, and the kernel size of the operation for each linear operation is $d\times d$. Therefore, the speedup example of replacing the normal convolution operation using the Ghost module is shown in Eq. (9).

$$\begin{aligned} r_s &= \frac{n\cdot h'\cdot w'\cdot c\cdot k\cdot k}{\frac{n}{s}\cdot h'\cdot w'\cdot c\cdot k\cdot k + (s-1)\cdot\frac{n}{s}\cdot h'\cdot w'\cdot d\cdot d} \\ &= \frac{c\cdot k\cdot k}{\frac{1}{s}\cdot c\cdot k\cdot k + \frac{s-1}{s}\cdot d\cdot d} \\ &\approx \frac{s\cdot c}{s+c-1} \\ &\approx s \end{aligned} \tag{9}$$

where $d\times d$ and $k\times k$ have similar magnitudes and $s \ll c$, then the theoretical parameter compression example is shown in Eq. (10).

$$
\begin{aligned}
r_c &= \frac{n \cdot c \cdot k \cdot k}{\frac{n}{s} \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot d \cdot d} \\
&\approx \frac{s \cdot c}{s + c - 1} \\
&\approx s
\end{aligned}
\tag{10}
$$

So the theoretical parameter compression ratio obtained by replacing the ordinary convolution with the Ghost module is approximately equal to the acceleration ratio, but it is much less computationally intensive than the ordinary convolution before replacement.

2.2.3. Structural design. The SENet and GhostNet described above are incorporated into the YOLOv5 backbone network for the purpose of improving the YOLOv5 algorithm, as shown in Fig. 1.

The improved model utilizes the Ghost Bottleneck module with a step size of 1 instead of the original feature extraction layer, and the feature extraction is not affected in any way. The ordinary convolutional layer in the original network model is replaced by the Ghost Bottleneck module with step size 2. The GhostNet network composed of two Ghost Bottleneck modules utilizes linear transformations in the model to reduce the volume size and reduce the computational cost of the network. On the other hand, the dependency between feature channels is considered, and the SE module is utilized for the fusion of feature channels, which completes the validation of the channel dimensions of the Ghost Bottleneck module against the original features, enhances the expression of features, improves the robustness and accuracy of the extracted features of the Ghost Bottleneck module, and accelerates the convergence of the network model.

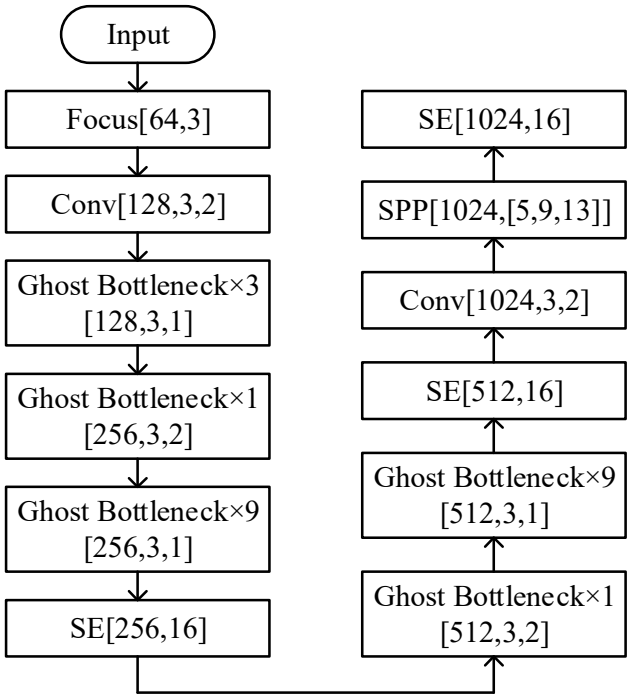

Fig. 1. Reconstructed YOLOv5 main dry network

2.3. Experimentation and Analysis

2.3.1. Data set selection. The goal of this chapter is to train a highly accurate human target detection model that facilitates the task of pose estimation for vocal singing.In order to have a more critical judgment on improving the performance of the model, the official dataset MS COCO dataset was chosen

as the dataset for the human detection experiments.The MS COCO dataset is a large dataset of target detection, segmentation, and keypoint detection that contains multiple types of image annotations. Since the research objective of this paper is the category of "person", the "person" class in the MS COCO dataset is selected as the dataset of the experiment in this chapter, which contains 45174 pictures of the human body, including 185316 labeling frames of the "person" class, and 5000 pictures are selected as the object detection dataset of this paper due to the limitation of the experimental platform and experimental time, and the training set is divided according to the ratio of 8:1:1. Validation set and test set, that is, 4000 sheets are randomly selected as the training set, 500 sheets as the validation set, and 500 sheets as the test set.

2.3.2. Experimental setting and evaluation indicators. The model uses precision, recall, weighted score, average precision, mean average precision, accuracy, and frames per second as evaluation metrics to examine the model performance. The calculation formula is as follows:

$$Precision = \frac{TP}{TP + FP} \times 100\% \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \tag{12}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\% \tag{13}$$

$$AP = \int_0^1 P(R)\, dR \tag{14}$$

$$mAP = \frac{\Sigma AP}{N(Class)} \tag{15}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{16}$$

The AP value is the value of the area of the PR curve for a particular category, and the vertical coordinate in the Cartesian coordinate system where the PR curve is located is the precision rate P, and the horizontal coordinate is the recall rate R. The precision rate is the proportion of the number of positive samples correctly predicted by the model to the number of all samples predicted to be positive, and the recall rate is the proportion of the number of positive samples correctly predicted by the model to the number of all samples that were actually positive, and at different confidence thresholds, there are different precision rates and recall rates. Plot the PR curve based on the (R, P) values and calculate the area under that curve, which is the AP value for a category. Calculate the AP value for all categories and find the mean value which is the mAP value.

2.3.3. Experimentation and analysis. The experiments in this chapter use Adam optimizer to train the model with momentum of 0.937 and a total of 400 Epochs, 100 Epochs are trained in the freezing phase with batch_size set to 8 and a learning rate of 0.001, and after thawing, another 300 Epochs are trained with batch_size set to 4 and a learning rate of 0.0001. The Warm up method was used to speed up the model convergence during the training process.

The training and validation loss curves of the improved YOLOv5 algorithm model are shown in Fig. 2. During the training process, the training loss and validation loss are smoother, and the size of the loss value drops below 0.4 in the first 20 epochs, and the model is able to quickly converge to a lower loss value in the early stage of training, which indicates that the model exhibits a good performance on the learning task. The P-R curve of the detection results on the test set is shown in Figure 3, and the AP value of “person” is 78.11% by calculating the area of the lower part of the P-R curve in the coordinate system.

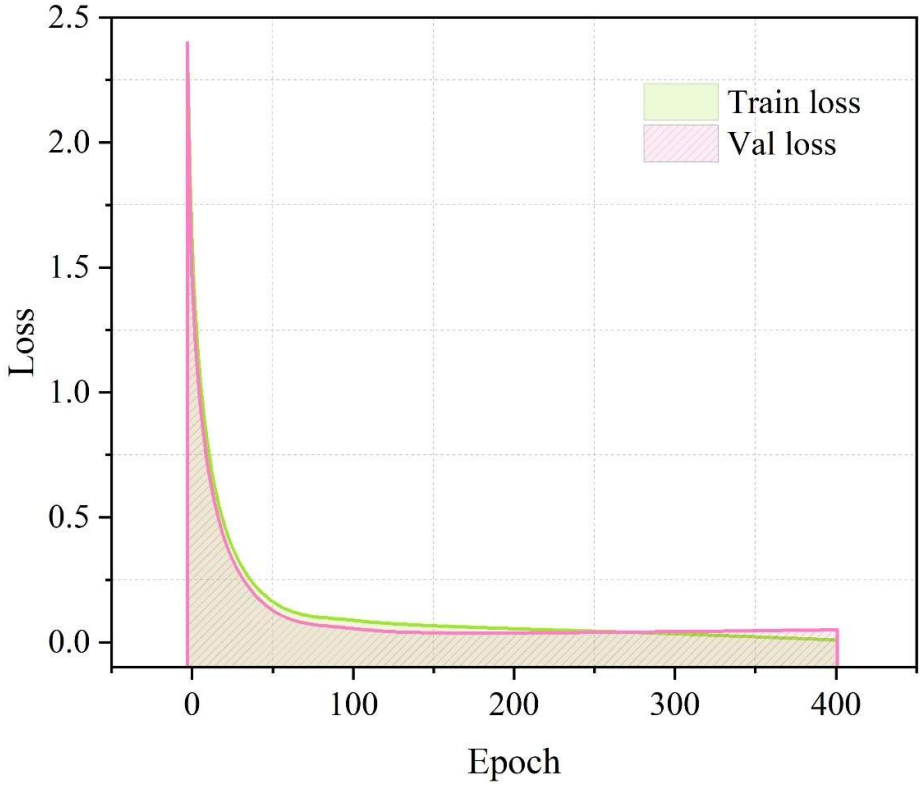

Fig. 2. Improved YOLOv5 algorithm loss curve

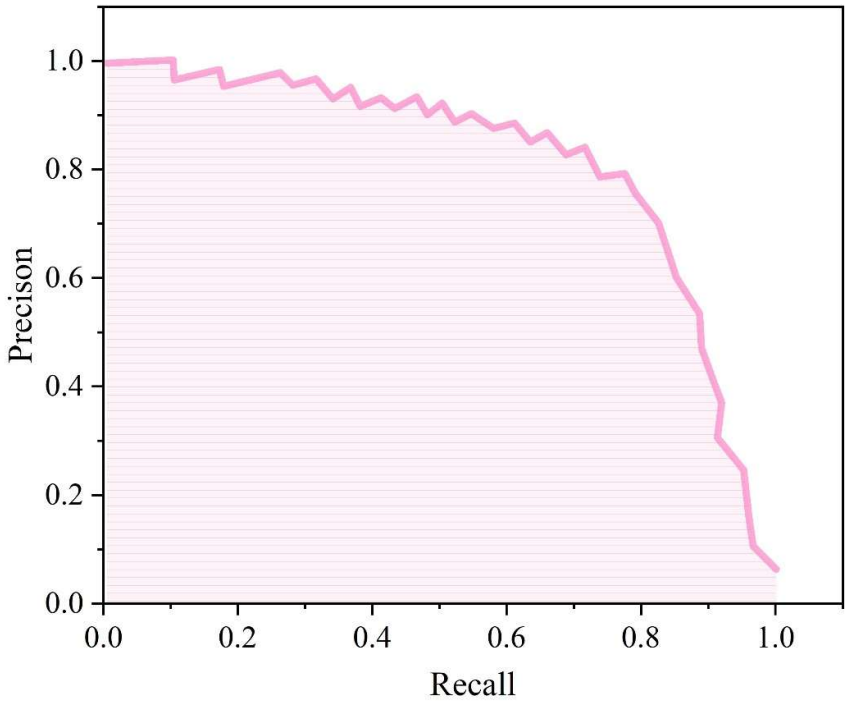

Fig. 3. Improved YOLOv5's p-r curve

In order to better verify the effectiveness of the algorithm in this paper, comparison experiments before and after the algorithm improvement are conducted. A total of four groups of networks are verified, respectively, YOLOv5s, C2f-YOLOv5s, C2f-CC3-YOLOv5s, and the improved YOLOv5 algorithm, all of which use the dataset produced in this paper, all of which are trained in the same experimental platform, and after obtaining the optimal model, the test is conducted on a test set, and the results of

the objective evaluation of the experimental models are compared and contrasted as shown in Table 1. Where the complexity of the network is measured by the number of parameters and floating point operations (FLOPs), the smaller the values of the number of parameters and FLOPs, the lower the network complexity. Compared with the original algorithm, the improved YOLOv5 algorithm has improved in the number of parameters and floating point operations, but the network complexity of this algorithm is still at a very low level compared to other deep learning based target detection algorithms. The improved YOLOv5 algorithm achieves a Precision value of 84.46%, a Recall value of 72.34%, a mAP value of 78.14%, a Parameters value of 8.716 × 106, and an FPS of 51.04 on the dataset of this chapter.The number of parameter of the model of this algorithm has slightly increased, and the speed of detection has slightly decreased, but the Accuracy, the Recall, and the average precision are greatly improved.

Table 1. The YOLOv5 algorithm improves the performance contrast of the test

Algorithm	P (%)	R (%)	F1 (%)	mAP (%)	Parameters (106)	FLOPs (106)	FPS
YOLOv5	79.15	66.29	74.72	74.32	8.023	17.267	53.51
C2f-YOLOv5	83.24	70.15	76.48	75.37	8.541	21.052	52.16
C2f-CC3-YOLOv5	85.28	72.06	78.11	77.23	8.716	21.064	51.22
Improved YOLOv5	85.75	72.34	78.48	78.14	8.716	21.064	51.04

3. Vocal singing posture and movement recognition

3.1. Singing posture estimation incorporating multiple attention mechanisms

3.1.1. High-resolution networks. The feature extraction process is shown in Fig. 4, where the high-resolution thermogram is first down-sampled to reduce the resolution of the feature map, and then returned to the original resolution by up-sampling. However, the serial method is difficult to effectively integrate the information from different resolution feature maps, and the repeated up and down sampling process may result in the loss of feature information, thus affecting the accuracy of attitude estimation.

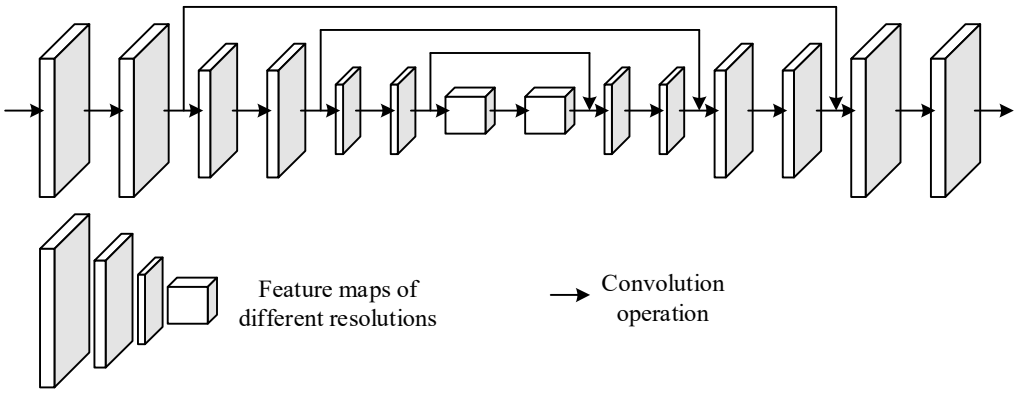

Fig. 4. Traditional network structure

3.1.2. Network architecture design. In this chapter, we mainly use HRNet as the backbone network, and design a high-resolution network based on multiple attention mechanisms by fusing the attention mechanisms to enhance its feature extraction ability in multi-person complex scenes.

The design process of CDLNet is as follows:

1) At the very beginning, a preprocessing operation needs to be performed on the input image, in which two convolutions with a step size of 3×3 will be used to reduce the resolution of the image.

2) In the first stage four Coorneck modules are utilized to replace the original Bottleneck modules for enhanced extraction of spatial coordinate feature information.
3) The subsequent three stages utilize Coorblck modules to replace the original Basicblock modules, also used to enhance the extraction of coordinate information. The resolution of each stage is gradually reduced to 1/4, 1/8, 1/16 and 1/32 of the original.
4) A 64×48×17 thermogram is finally obtained as output for performing human body attitude estimation.

3.1.3. Module construction based on coordinate attention. In the coordinate attention module, the global pooling in the channel attention is replaced by encoding two one-dimensional vectors, and then the feature X is encoded in the horizontal direction through the pooling kernel (H, 1) and in the vertical direction through the pooling kernel (1, W). Where the one-dimensional features in the horizontal direction vertically are shown in equations (17) and (18).

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i < W} x_c(h, i) \tag{17}$$

$$z_c^h(w) = \frac{1}{H} \sum_{0 \le j < H} x_c(j, w) \tag{18}$$

The above two expressions transform the features from different directions into feature maps corresponding to their directions, and then the far distance relationship is obtained by the attention module and the space is preserved, and then Eq. (17) and Eq. (18) are concatenated and passed through a 1×1 convolutional layer to obtain the attention maps by performing a target transformation on the embedded information. This transformation process can be represented by Eq. (19):

$$f = \delta(F_1([z^h, z^w])) \tag{19}$$

Where f denotes the intermediate feature mapping; δ denotes the activation function to be used; and F_1 denotes the splicing of the pooling results in different directions. $f \in K^{Clr \times (H+W)}$ Because it contains intermediate features in different directions, it is necessary to decompose f into two other features $f^h \in K^{Clr \times H}$ and $f^w \in K^{Clr \times W}$ using spatial dimensionality, and then use 1×1 convolution and sigmoid function to make their dimensions consistent with the inputs as shown in Eqs. (20) and (21):

$$g^h = \sigma(F_h(f^h)) \tag{20}$$

$$g^w = \sigma(F_w(f^w)) \tag{21}$$

Combining g^h and g^w into a weight matrix is used to calculate the output of the coordinate attention module, and finally, the output of the coordinate attention module is shown in equation (22):

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j) \tag{22}$$

Where, y_c denotes the output features; x_c denotes the input features; $g_ò^h$ denotes the spatial features in vertical dimension; g_c^w denotes the spatial features in horizontal dimension.

3.1.4. Multi-scale attention module. The multiscale attention module contains multiple parallel branches of attention, each responsible for processing features at different scales. Among them, different branches have different receptive field sizes to accommodate different levels of features. By adding this module to the high-resolution network structure, the network can be made to consider feature information from multiple scales and fuse them across different branches to extract feature information from each branch [17].

In the first branch, the input feature map A (C × H × W) needs to be subjected to three convolutional

layers to obtain three other feature maps B, C, and D, which are then reconstructed as C × N, where N = H × W. Next, the reconstructed B is transposed and multiplied by the elements of the reconstructed feature map C, and the result is passed through the SoftMax function to obtain the spatial attention map S (N × N). The attention map S is then multiplied by D and scaled by a scale factor α. The resulting tensor is reshaped back to its original shape and added to the input feature map A to obtain the final output E. This is shown in equation (23):

$$s_{ji} = \frac{\exp(B_i \cdot C_j)}{\sum_{i=1}^{N} \exp(B_i \cdot C_j)} \tag{23}$$

where S_{ji} denotes the effect of i^{th} -position on j^{th} -position. Then in order to obtain the final output feature map E, a new feature map D is obtained by first passing the feature A through the convolutional layer.Then D is shaped into a new representation $K^{C\times H}$ and a transformation tensor is obtained by computing the transpose between the shaped attention maps S and D by performing a matrix multiplication $K^{C\times H\times W}$. Finally, $K^{C\times H\times W}$ scaling is performed through the parameter α and the original feature A is added to the elements to obtain the final tensor result $E \in K^{C\times H\times W}$, the specific process is shown in equation (24):

$$E_j = \alpha \sum_{i=1}^{N} (s_{ji} D_i) + A_j \tag{24}$$

Where α denotes the weight factor and its initial value is 0. In the second branch, it is necessary to multiply the two extracted feature maps, and then in the feature transfer through the SoftMax function, so that a new channel attention map can be obtained $X(C\times C)$. The specific process is shown in Equation (25):

$$x_{ji} = \frac{\exp(A_i \cdot A_j)}{\sum_{i=1}^{c} \exp(A_i \cdot A_j)} \tag{25}$$

where x_{ij} denotes the effect of channel i^{th} on channel j^{th}. After the above operation, the reconstructed feature mapping $K^{\mathrm{C}\times H\times W}$ is obtained by calculating the transpose of X and performing a matrix multiplication with A. Subsequently, the obtained result is multiplied by the scale parameter β and an element-by-element summation operation is performed using A. This process produces the final output $E \in K^{C\times H\times W}$ as shown in Equation (26), where β learns the weights starting from zero:

$$E_j = \beta \sum_{i=1}^{C} (x_{ji} A_i) + A_j \tag{26}$$

3.1.5. Non-localized Attention Module. First, $X = T\times H\times W\times 256$ in the module needs to go through three 1 × 1 convolution operations to reduce its own dimension, which results in new three feature maps a, b, and c. This has the advantage of reducing the number of channels in the network and reducing the number of parameters. The three feature maps are then reshaped to a size of T × H × W × 128. The reshaped feature map b is transposed and multiplied with the matrix to obtain the product ab, which is then multiplied element-by-element with the feature map c after passing through the SoftMax function, and then the result of the multiplication is subjected to a 1×1 convolution process to restore the original number of dimensions so that it matches the input dimensions. In this case, the nonlocal attention mechanism is defined as shown in equation (27):

$$y_i = \frac{1}{C(x)} \sum_{\forall j} f(x_i, x_j) g(x_j) \tag{27}$$

where i denotes the output position index; j denotes the index of all possible positions; and $C(x)$ denotes the normalization function. Finally, combining Eq. (27), the nonlocal module definition is obtained as shown in Eq. (28):

$$z_i = W_z y_i + x_i \tag{28}$$

where w_z denotes a learnable matrix; y_i denotes an output; and x_i denotes an input.

3.2. Singing action recognition based on spatio-temporal map convolutional neural network

3.2.1. Overall Architecture and Problem Definition for Action Recognition Modeling. The main framework based on the Multi-Granularity Spatio-Temporal Graph Convolutional Neural Network (MGstgcn) is shown in Fig. 5, and the model consists of three main modules: an encoder for extracting the motion information, a decoder with temporal level online difficult joint-point mining training, and an action classifier. The encoder module consists of a joint point coordinate subtraction block, a granularity information fusion module and a granularity transformation module. Its purpose is to extract richer semantic information of movements by mutual guidance of different granularity information, and output three feature granularities as inputs to decoder and classifier.

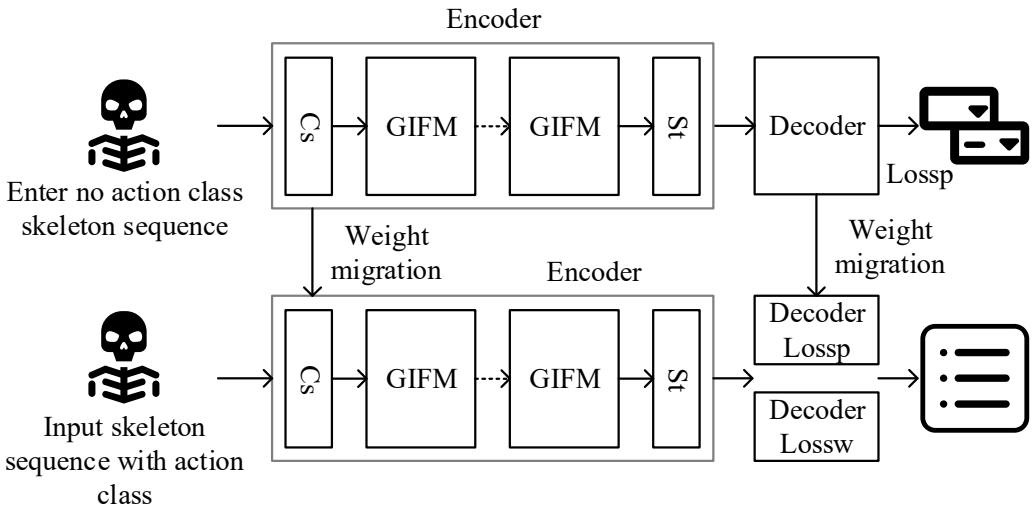

Fig. 5. Overall structure diagram of the model

In this paper, a skeleton graph is denoted using $G = (V, B, C)$, where V denotes a joint point, B denotes a body skeleton connected by a joint point, and C denotes a body part connected by a skeleton. The adjacency matrix is defined as $A \in \{0,1\}^{n \times n}$, where $A_{i,j} = 1$ when i is connected to j, and 0 otherwise.The adjacency matrices at all three granularities use the same partitioning strategy as ST-GCN.

3.2.2. Encoder module for extracting features. First, a new representation of the human body is proposed: the multi-granularity map, and unlike previous work that only divides the joints, this paper divides the human joints into granularity according to the human skeleton and body parts. The multi-granularity computation relies on the Cs module, which is not generated by erasing the extra joints on the bones or body parts as in previous work, but by subtracting the coordinates of the joints. This approach preserves more motion information without destroying the integrity of the information.

Secondly, this paper proposes a granularity information fusion module (GIFM), as shown in Fig. 6, which consists of a single-granularity feature extraction block ST-GCN and a multi-granularity fusion block (Msfb). After single-granularity motion features are extracted by ST-GCN, then the granularity interaction motion information is acquired by Msfb.

In this paper, a three-layer MLP as shown in Fig. 6 is designed to obtain the attention matrices of two neighboring granularities $A_{s_2s_1}$, which speeds up the computation and is easy to train. The process of generating As_is_j can be described as follows:

$$h_{s_1} = f_{MLP_1}(F^{s_1}) \tag{29}$$

$$h_{s_2} = f_{MLP_2}(F^{s_2}) \tag{30}$$

$$A_{s_2s_1} = soft\max(h_{s_1}^T h_{s_2}) \tag{31}$$

where f_{MLP_1} and f_{MLP_2} denote the MLP, and F^{s_1} and F^{s_2} denote the features at joints S_1 and S_2, respectively. $A_{s_2s_1}$ is the attention matrix from S_2 to S_1. After obtaining the attention matrix $A_{s_2s_1}$, cross-granularity human correlations are adaptively explored in different ways. And using $A_{s_2s_1}$ to introduce information from neighboring granularities S_2 to S_1 into S_1, the features at S_1 are updated as:

$$F^{s_1} \leftarrow A_{s_2s_1}F^{s_2} + F^{s_1} \tag{32}$$

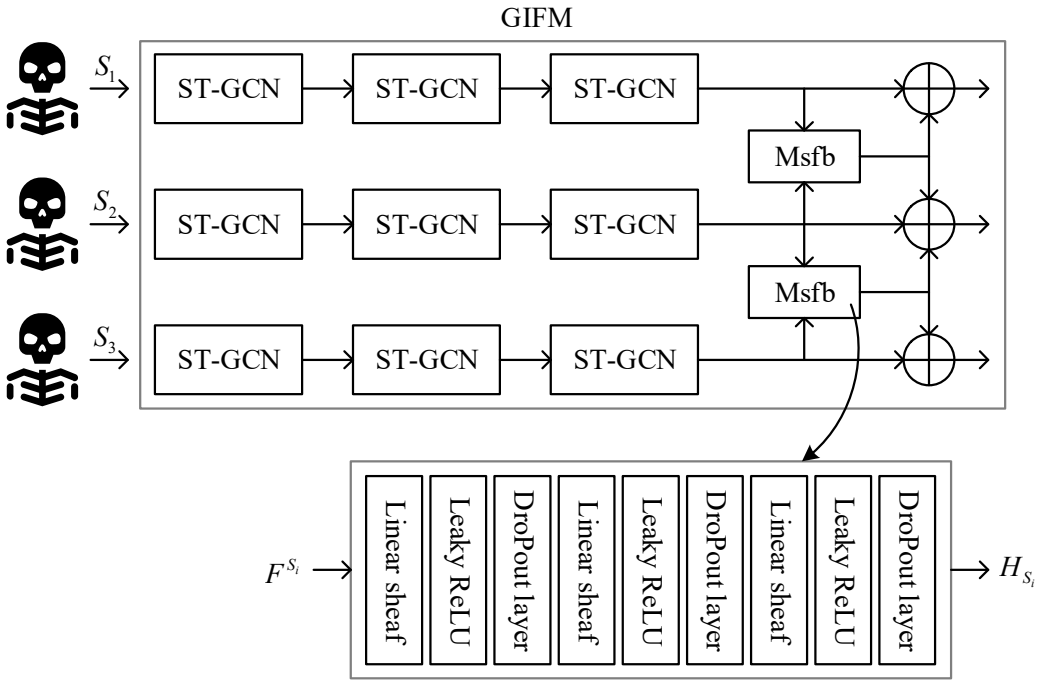

Fig. 6. Structure diagram of the granularity information fusion module

Then after GIFM training, the skeleton granularity and body part granularity are reduced to six-node granularity as input to the decoder or classifier. The skeleton granularity map or body part granularity map is converted to an articulation point granularity map. For example, one joint point of the left arm is converted to elbow, wrist and hand joint points. One MLP is applied to each granularity map part, and since there are 15 parts in total for S_2 and S_3, a total of 15 MLPs are applied to realize the granularity conversion of the graph. When converting S_2 back to S_1 (S_3 operates similarly), it can be described by Eq. (33):

$$F_{i,j}^{s_2} = f_2(F_k^{s_2}) \tag{33}$$

where $F_{i,j}^{s_2}$ is the feature at S_2, but with the same number of nodes as S_1. $F_k^{S_2}$ is the feature when the granularity is skeletal.

3.2.3. Decoder Module for Predicting Action Categories. In this paper, the features extracted by the encoder are decoded from larger to smaller granularity. The larger granularity provides information about the global motion evolution, which can indicate approximate motion direction, velocity, and class of action. The smaller granularity is then able to predict the exact joint position using the global information. By using Tohjm's training approach in the training phase, the network is able to focus its attention on joints where losses are difficult to reduce, and this strategy helps the model to focus more on hard samples and effectively reduce prediction errors.

3.2.4. Classifier Module for Recognizing Action Categories. The role of the classifier is to classify the human actions of the skeleton sequence based on the motion features extracted by the encoder. The classifier uses a spatial-temporal average pooling layer (STAVP) and a full connectivity layer (FC). The output scores of the joints from larger to smaller granularity are then summed as the final action classification score.

3.2.5. Loss function. Reconstruct the loss function of MGstgcn as follows:

$$Loss_w = Loss_c + \lambda Loss_p \tag{34}$$

$$Loss_c = -\sum_{c=1}^{M} p_c \log(q_c) \tag{35}$$

where $Loss_w$ is the loss function of the whole network, $Loss_c$ is the cross-entropy loss function, and $loss_p$ is the position loss error of the joints. $loss_p$ is shown in equation (36):

$$Loss_p = \frac{1}{MK}\sum_{j=1}^{K}\sum_{m=1}^{M}\left\|\hat{p}_{j,m} - p_{j,m}\right\|^2 = L_{t-ohjm} \tag{36}$$

where $\hat{p}_{j,m}$ is the predicted joint position and p_j, m is the true joint position.

4. Vocal singing posture recognition experiment and analysis

4.1. Analysis of vocal singing posture estimation

4.1.1. Data sets. For vocal singing gesture estimation, the output of vocal singing gesture estimation is used as the input of vocal singing gesture estimation, and the output is a sequence of vocal singing gesture points containing depth information. The vocal singing gesture estimation experiments are mainly trained and tested using the Human3.6M dataset. The action videos of the dataset were produced by four stationary RGB cameras shooting at 50 HZ. The dataset contains 15 different actions, including S1: standing, S2: jumping, S3: raising hands, etc. A total of 15 subsets are used in the experiments of vocal singing gesture estimation method in this chapter, the subsets (S1, S5, S6, S7, S8) in Human3.6M are used for the training of vocal singing estimation network, and the subsets (S9, S11) are used for the testing.

4.1.2. Evaluation indicators. For vocal singing posture estimation, the metric Mean Posture Point Position Error (MPJPE) is usually used to measure the model accuracy, and the MPJPE value is determined by the Euclidean distance between the predicted skeletal posture point coordinates and the corresponding actual labeled skeletal posture point coordinates, and the smaller the metric value is, the higher the model accuracy is represented, and the better the vocal singing posture estimation algorithm is. The MPJPE is computed as shown in Eq. (37).

$$MPJPE = \frac{\Sigma_{i=1,F=1}^{FJ}(\square P_i^{'3} - P_i^{gt_3}\ \square)}{FJ} \tag{37}$$

4.1.3. Related parameter settings. The experiments in this chapter are based on the high-resolution network mechanism with multiple attention mechanisms, mainly trained on the vocal singing posture estimation dataset, for Openpose, Pifpaf, HRNet and HigherHRNet using the trained models in the MMPose open source toolkit on the dataset in this paper. The experimental hardware and software environments used for 2D human posture estimation are shown in Table 2.

Table 2. Experimental hardware and software environment Settings

Hardware environment	CPU	GPU	Memory	Hard disk
	Intel® CoreTM i5-12400	NVIDIA 3080*1	DDR5- 4800MHz	Samsung 980 pro Nvme™ m.2
Software configuration	System	Python version	Anaconda version	CUDA version
	Windows11	3.8	2022.10 64bit	11.7

4.1.4. Analysis of experimental results. The vocal singing posture estimation experiments in this chapter are carried out using the same deep learning environment based on the vocal singing posture estimation dataset used in this paper under the same configuration conditions, comparing the bottom-up Openpose and Pifpaf methods as well as the top-down Mask-RCNN, HRNet, and HigherHRNet methods with the CDLNet method proposed in this paper in the performing vocal singing posture estimation.The specific training process loss curve of CDLNet method is shown in Fig. 7. After obtaining the training of the feature extraction network, the experiment counted the distribution of pose point confidence on the test set, and analyzed the correspondence between the erroneous poses in the visualization effect and the confidence of the pose points in the human body parts; the distribution of the pose points with confidence lower than 0.6 was particularly discrete, and the vast majority of the visually erroneous pose points were concentrated between 0.6 and 0.12.

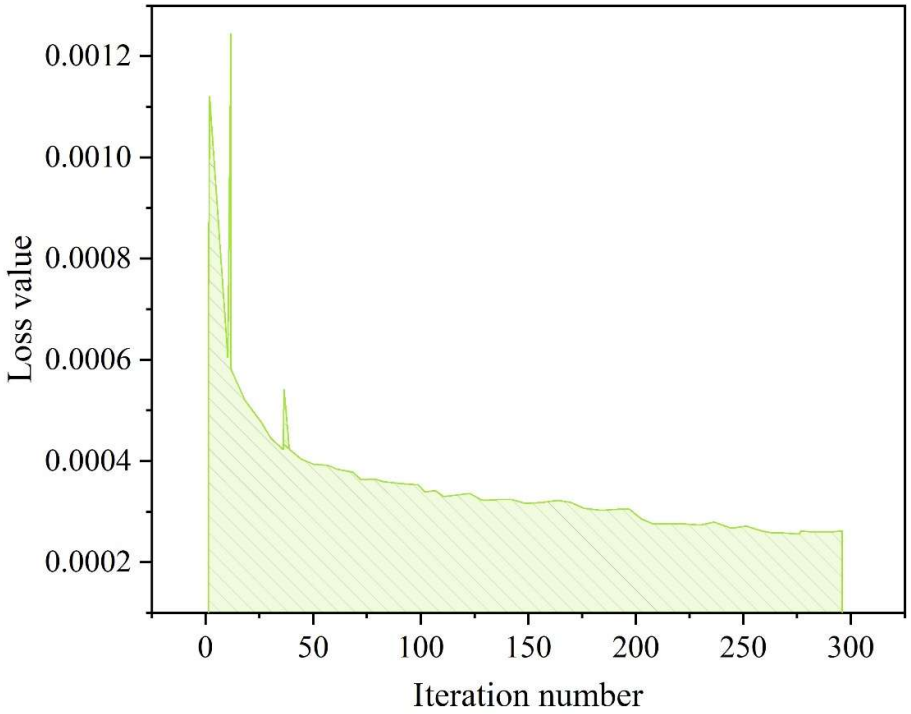

Fig. 7. CDLNet network training loss diagram

Based on the attitude point distribution, the starting value of the confidence threshold λ is incremented in steps of 0.01 from 0.60 to determine the confidence threshold λ in the attitude point correction module when making the highest model precision, and the results are shown in Fig. 8,

where the highest precision and recall of the CDLNet model is achieved at λ of 0.72.

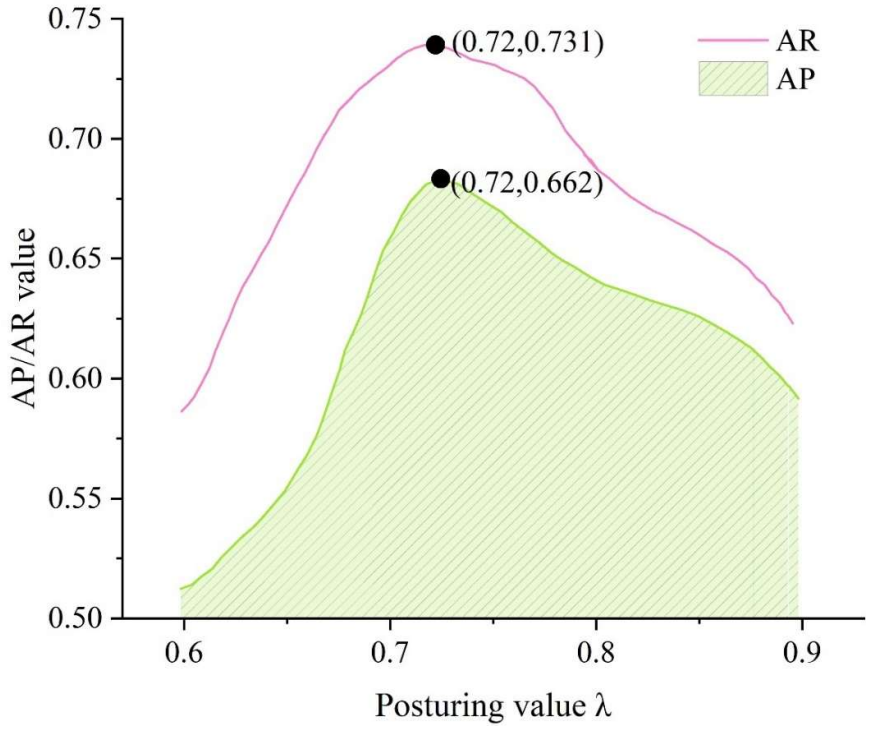

Fig. 8. The Ap/Ar value varies with the threshold

For the accuracy, Table 3 shows the comparison results between the vocal singing posture estimation method CDLNet proposed in this paper and the current mainstream methods in terms of accuracy (AP) and recall (AR) in the same experimental environment, from which it can be seen that the vocal singing posture estimation method CDLNet's accuracy (AP) is higher than that of all other methods, and the recall is 0.721.

Table 3. The vocal performance of the vocal performance was compared

Methods	Backbone network	AP	AP.5	AP.75	APM	APL	AR
Method 1	Vgg	0.571	0.802	0.645	0.543	0.657	0.636
Method 2	ResNet50	0.483	0.718	0.521	C0.358	0.591	0.545
Method 3	ResNet50-FPN	0.618	0.754	0.596	0.574	0.694	0.675
Method 4	HRNet-32	0.663	0.849	0.718	0.617	0.753	0.714
Method 5	ResNet50	0.641	0.836	0.664	0.584	0.736	0.704
CDLNet	-	0.653	0.844	0.707	0.608	0.762	0.721

For the vocal singing pose estimation method CDLNet used in this paper, the accuracy of its vocal singing pose estimation is evaluated after training, and the MPJPE metrics are compared with the current mainstream vocal singing pose estimation methods on the Human3.6M dataset, and the comparison results are shown in Table 4.

Method 13 in Table 4 is an end-to-end 3D human pose estimation method, all others are 2D pose mapping to 3D pose, sin denotes the interval step for video token selection, and sout denotes the interval step for outputting 3D pose frames of the video. Finally, in the complex motion action dataset, this chapter uses the vocal singing pose estimation method CDLNet to obtain the 3D human poses in the singing action frames, and Fig. 9 shows its visualization effect.

Table 4. The vocal performance of the vocal performance was compared

Method	sin	MPJPE	
		Key-frames	All frames
Method 1	4	47.7	46.7
Method 2	4	46.1	45
Method 3	4	45.8	44.8
CDLNet	4	45.9	44.6
Method 5	10	49.8	49.3
Method 6	10	47.2	46.6
Method 7	10	45.9	45.4
Method 8	10	45.9	45.2
Method 9	20	52.8	55
Method 10	20	47	49.3
Method 11	20	46.4	48.7
Method 12	20	46.3	46.9
Method 13	-	-	65.6

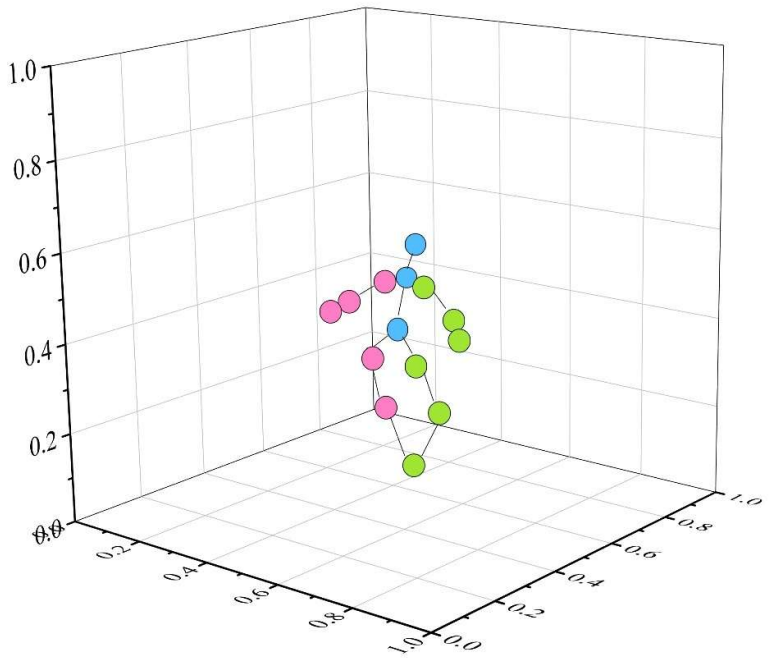

Fig. 9. Visual performance of vocal voice estimation model

4.2. Analysis of vocal singing movement recognition

4.2.1. Data set processing. In this paper, the pictures in the vocal singing posture section of HSiPu2 are reclassified into eight categories according to the correct or incorrect details of the movements, which contain upright posture, head and neck posture, swinging hand and loose shoulder posture, lifting muscle and shouting posture, smiling and frowning posture, and hand and arm posture. After data enhancement means such as left-right flip and Gaussian noise, a total of 9452 images were obtained.

4.2.2. Experimental Procedures. In order to verify the feasibility of the vocal singing posture recognition method based on convolutional neural network with multi-granularity spatio-temporal

map, this paper chooses the classical models such as VGG16, KNN, SVM, Decision Tree and Random Forest as the comparison models. Among them, VGG16 is not processed by human gesture estimation and gesture analysis, and uses the original pictures as input features, with the learning rate set to 0.00001 and epochs set to 100; KNN uses a 10-fold cross-validation method and takes the K value to be 4; SVM uses the lattice search method and takes 5 as the C value, and the kernel function is poly, and Decision Tree using grid search method, the maximum depth of the tree is taken as 6, the minimum number of samples required for leaf nodes is 2, the minimum number of samples required for node reclassification is 2, and the number of Random Forest classifiers is set to 200.Meanwhile, Accuracy is chosen as the evaluation criterion for each classification model. That is, for a given motion dataset, the ratio of the number of samples T correctly classified by the classifier to the total number of samples N is used as the evaluation index. Each experiment randomly divides the dataset proportionally and repeats five times, and the experimental results are taken as the mean value. The experimental environment of this paper is shown in Table 5.

Table 5. Experimental environment

Categories	Environmental details
Operating system	Ubuntu-LTS 20.04
Programming language	Python
RAM	64GB
CPU	Intel-12700KF
GPU	Nvidia-3090
Depth learning framework	torch= 1.12.1 torchvision= 0.13.1

4.2.3. Experimental results and analysis. The comparison of the final model's classification results on each posture category is shown in Table 6, and from the experimental results, it can be seen that MGstgcn outperforms the other comparison models in terms of accuracy, with an average accuracy of up to 86.5%. This is because the motion information can be better captured from the trajectory of the human joint points changing with the motion, which can be analyzed in combination with the rules of the gesture itself to better discriminate whether the movement is standard or not.

Table 6. Comparison of experimental results

	VGG16	KNN	SVM	Decision Tree	Random Forest	MGstgcn
Stand up	0.891	0.861	0.888	0.75	0.882	0.927
Neck position	0.717	0.677	0.764	0.365	0.594	0.823
Hand loose shoulder posture	0.828	0.817	0.849	0.831	0.856	0.877
The lifting muscle shouts	0.777	0.809	0.832	0.817	0.694	0.881
Smiling eyebrow	0.92	0.94	0.912	0.882	0.905	0.921
Palm arm posture	0.839	0.877	0.847	0.834	0.818	0.859
Average accuracy	0.829	0.83	0.849	0.747	0.792	0.865

In order to verify that there is variability in the performance of the algorithms, a Friedman test is used to make a judgment. The mean ordinal values of the algorithms are shown in Table 7, comparing k algorithms on the individual N datasets, such that r_i denotes the mean ordinal value of the ith algorithm. r_i obeys a chi-square distribution with mean $(k+1)/2$ and variance $(k^2+1)/2$. τ_x^2 obeys a chi-square distribution with degree of freedom $k-1$, and $\tau_F{}^{\tau}{}_F$ obeys an F-distribution with degrees of freedom $k-1$ and $k-1(N-1)$.

$$\tau_{x^2} = \frac{12N}{k(k+1)}\left(\sum_{i=1}^{k} r_i^2 - \frac{k(k+1)^2}{4}\right) \tag{38}$$

$$\tau_F = \frac{(N-1)\tau_x^2}{N(k-1)-\tau_x^2} \tag{39}$$

The statistic τ_F is calculated to be approximately equal to 7.673, which is greater than the corresponding critical test value in the F-distribution. Therefore, there is a significant difference between the performance results of multiple algorithms. In summary, the MGstgcn algorithm proposed in this paper is capable of intelligently analyzing and evaluating vocal singing posture in real time, providing a new means of recognition.

Table 7. Algorithm order value

	VGG16	KNN	SVM	Decision Tree	Random Forest	MGstgcn

Stand up	3	6	4	7	5	2
Neck position	4	5	3	7	6	2
Hand loose shoulder posture	4	7	5	6	3	2
The lifting muscle shouts	5	4	2	3	7	6
Smiling eyebrow	2	3	5	7	6	4
Palm arm posture	3	2	5	6	7	4
Average order value	3.5	4.5	4	6	5.7	3.3

5. Conclusion

The article investigates vocal singing posture based on image recognition technology, reconstructs the backbone network using SENet and GhostNet, and improves the original YOLOv5 algorithm, and finally conducts a comparison test between the original YOLOv5 algorithm and the improved YOLOv5 algorithm, and the improved YOLOv5 algorithm N has a precision value of 85.75% and a Recall value of 72.34%, and an FPS of 51.04. A proposed method based on the image recognition technology is based on the vocal singing posture, and the mAP value is 78.34%. 85.75%, Recall value of 72.34%, mAP value of 78.48%, Parameters value of 8.716×106, and FPS of 51.04. A high resolution network based on multi-attention mechanism (CDLNet) is proposed, which has the highest precision of gesture points in the singing action dataset, with an AP value of 0.662 and AR value of 0.731.According to the action recognition model based on multi-granularity spatio-temporal graph convolutional neural network proposed in this paper for gesture analysis of vocal singing gesture actions, experimental validation on the HSiPu2 dataset shows that this algorithm has a higher accuracy than the other commonly used algorithmic models in recognizing vocal singing gesture actions.

Funding

Project:
1) Research on the inheritance status and countermeasures of Zigong Salt Worker's Trumpet (MYYB2023-25).
2) Exploring the Teaching Mode of Ethnic Music Integrated into Physical Education Professional Practice Courses in Colleges and Universities (MYYB2023-20).

References

[1] Somme, L., Audouin, M., Chau, N., Beyaert, C., & Perrin, P. (2024). Associations of Postural Activities and Knowledge for Voice with Breathing Issues and Voice-Physical-Disorders Among Lyric Singers. Journal of Voice: Official Journal of the Voice Foundation, S0892-1997.

[2] McCarther, S. (2021). A review of the breathing mechanism for singing: part I: anatomy. Rider University, 1-9.

[3] Piatnytska-Pozdnyakova, I., Korshunova, A., & Galaidin, A. (2022). ROLE AND FUNCTIONS OF THE TORSO IN THE PROCESS OF VOCALIZATION. Scientific Journal of Polonia University, 51(2), 125-134.

[4] Cirilo, T. A. S., Oliveira, V. G. D., Ceniz, A. A., Tanigute, C. C., Fernandes, A. C. N., & Silva, E. M. D. (2021). Singers' postural alignment and vocal quality. Audiology-Communication Research, 26, e2499.

[5] Brown, K. S. (2021). My Body, My Instrument: How Body Image Influences Vocal Performance in Collegiate Women Singers. Teachers College, Columbia University.

[6] Nelson, S. H., & Blades, E. L. (2018). Singing with your whole self: A singer's guide to Feldenkrais awareness through movement. Rowman & Littlefield.

[7] Abdumutalibovich, M. A. (2022). The relevance of traditional singing and its place in higher education.

International Journal on Integrated Education, 5(2), 212-216.

[8] Peultier-Celli, L., Audouin, M., Beyaert, C., & Perrin, P. (2020). Postural Control in Lyric Singers. Journal of Voice: Official Journal of the Voice Foundation, 36(1), 141-e11.

[9] Gates, R. (2018). Free to Sing: Encouraging Postural Release for Vocal Efficiency. MTNA e-journal, 9(4), 8-23.

[10] Tarvainen, A. (2018). Singing, Listening, Proprioceiving: Some Reflections on Vocal Somaesthetics. Aesthetic Experience and Somaesthetics, 1, 120.

[11] Julia, C., Dominika, M., Katarzyna, H., Pawel, M., Grzegorz, F., Maciej, K., & Malgorzata, K. M. (2021). Investigation of the relationship between the diaphragm muscle relaxation therapy, voice emission and postural stability in amateur and professional singers of Academy of Music--preliminary study. Journal of Measurements in Engineering, 9(1), 13-23.

[12] Blanco-Piñeiro, P., Díaz-Pereira, M. P., & Martínez, A. (2017). Musicians, postural quality and musculoskeletal health: A literature's review. Journal of Bodywork and Movement Therapies, 1(21), 157-172.

[13] Knight, E. J., & Austin, S. F. (2020). The Effect of Head Flexion/Extension on Acoustic Measures of Singing Voice Quality. Journal of voice: official journal of the Voice Foundation, 34(6), 964-e11.

[14] Kilpatrick III, C. E. (2020). Movement, Gesture, and Singing: A Review of Literature. Update: Applications of Research in Music Education, 38(3), 29-37.

[15] Reed, B. S. (2021). Singing and the body: Body-focused and concept-focused vocal instruction. Linguistics Vanguard, 7(4), 20200071.

[16] Wu Yingjun,Shi Junfeng,Ma Wenxue & Liu Bin. (2024). Bridge Crack Recognition Method Based on Yolov5 Neural Network Fused With Attention Mechanism. International Journal of Intelligent Information Technologies (IJIIT)(1),1-25.

[17] Ke Tang,Xing Zhao,Min Qin,Zong Xu,Huojiao Sun & Yuebo Wu. (2024). Using convolutional neural network combined with multi-scale channel attention module to predict soil properties from visible and near-infrared spectral data. Microchemical Journal111815-111815.