Traitement du Signal
Vol. 38, No. 2, April, 2021, pp. 529-538

Journal homepage: http://iieta.org/journals/ts

Dance Action Recognition and Pose Estimation Based on Deep Convolutional Neural Network

Fengling Zhu[1*], Ruichao Zhu[2]

[1] College of Music, Langfang Normal University, Langfang 065000, China
[2] College of Underwater Acoustic Engineering, Harbin Engineering University, Harbin 150001, China

Corresponding Author Email: zhufengling@lfnu.edu.cn

https://doi.org/10.18280/ts.380233

Received: 25 October 2020
Accepted: 6 February 2021

Keywords:
deep convolutional neural network (DCNN), dance action, pose estimation, DeepPose

ABSTRACT

Sports action recognition helps athletes correct their action range and standardize their poses. But it is not an easy task to recognize sports actions, due to the individual difference in action execution. Besides, the difficulty of action recognition increases with the diversity of actions and the complexity of background. The previous studies have not fully considered temporal changes, and failed to determine the exact staring point of actions. To solve the problem, this paper proposes a new method to recognize dance actions and estimate poses based on deep convolutional neural network (DCNN). Firstly, the authors presented full-effect expression of global and local features of dance actions, and derived an optimal model based on DeepPose. Next, a dance pose evaluation model was established based on time sequence segmentation network, and the sparse time sampling strategy was introduced to realize efficient and effective learning of the frame sequence of the whole video. Experimental results confirm the superiority of the full-effect expression of global and local features, and the effectiveness of the proposed model. The research results provide a reference for the application of deep learning (DL) in other scenarios of action recognition and pose estimation.

1. INTRODUCTION

Human action recognition involves multiple disciplines, such as image processing, machine vision, and artificial intelligence (AI). It is widely used in behavior capture and analysis, video surveillance, security control, and environmental prediction. Concerning the standardization of sports actions, human action recognition also has a huge application potential, and enables athletes to correct their action range and standardize their poses [1, 2].

In recent years, great functional progress has been made in visual object recognition and object behavior detection, thanks to deep learning (DL) algorithms like pattern recognition and machine vision [3-5]. Notably, deep convolutional neural network (DCNN) brings breakthroughs in image and video processing [6, 7]. Therefore, it is both theoretical and practical significant to recognize dance actions and estimate poses from the respective of DL.

From simple to complex, the contents of human action recognition can be divided into three levels, namely, mobile vision, action vision, and behavior vision [8-12]. However, weakly correlated action frames are often poorly processed, and the current method for representation and information fusion cannot select desirable features. To overcome these defects, Ozcan and Basturk [13] designed an action sequence segmentation algorithm for aerobics: the authors provided an approach to eliminate the continuous extreme values of the pose variation curve in continuous frames, extracted the three-dimensional scale-invariant feature transform (3D-SIFT) features and optical flow features of aerobic actions, calculated the similarity between action sequences, and imported the eigenvectors and calculated similarity into the classifier, which realizes the recognition of aerobic actions.

Good pose control can improve the power chain transmission efficiency and sports performance in all body parts of the athlete [14-19]. Nguyen et al. [20] performed action recognition and pose analysis on the screenshots of athlete videos, determined high-quality metrics of the qualified lifting action of wrestlers based on the results of expert interviews, set up a test group and a control group to compare the action qualities before and after intervention in pose control, and effectively improved the strength, speed, and torso stability of athlete actions.

Traditional motion parameter capture systems mostly use sensors to collect data [21-24]. Herath et al. [25] processed static and dynamic sports training images with OpenPose algorithm and the behavior sequence segmentation tool called derivative dynamic time warping (DDTW) algorithm, respectively, and identified the bone joint points based on optical flow and human motion continuity. Using OpenPose algorithm, Iosifidis [26] carried out body pose estimation and body tracking through measuring inter-frame pose distance, and greatly reduced the false positive, miss rate, and fallout ratio. Using the DDTW algorithm for behavior sequence segmentation, Faraki et al. [27] segmented the entry action sequence of divers, extracted the key frames of entry actions based on its relationship with the vertical motion trajectory of the athletes, and verified the feasibility and applicability of the algorithm by evaluating the extracted frames. Perera et al. [28] shot the golf swing actions of excellent athletes and professional students with fixed point, focus, and distance, extracted the features of rotation action from the three phases of the action executed by excellent athletes, compared the rotation features and kinematic indices between excellent athletes and professional students under the 3D image analysis system of Ariel, and offered professional students suggestions on improving the rotation angles of hip and shoulder, the coordinates of upper limbs, the gravity center of the body, and the exercise time.

The domestic and foreign studies on the detection and recognition of human actions have achieved fruitful results. However, the recognition of dance actions is challenged by the diversity of such actions. From the angle of standardization, there is a certain degree of individual difference in the execution of the same action. Besides, the difficulty of action recognition increases with the diversity of actions and the complexity of background. The effect of action recognition is greatly affected by the identification of starting point of each action. It is of great necessity to take account of temporal change while identifying the starting point.

For the above reason, this paper put forward a novel method for dance action recognition and pose estimation based on DCNN. Section 2 explains the full-effect expression of global and local features of dance actions, constructed an optimal model based on DeepPose, and realized the fusion between the results detected by local convolution model and the output of the global model. Section 3 develops a dance pose evaluation model based on time sequence segmentation network, and ensures effective and efficient learning of the frame sequence of the whole video by the sparse time sampling strategy. Finally, experiments were conducted to verify the superiority of the full-effect expression of global and local features, and to confirm the effectiveness of the proposed model.

2. FULL-EFFECT EXPRESSION OF GLOBAL AND LOCAL FEATURES OF DANCE ACTIONS

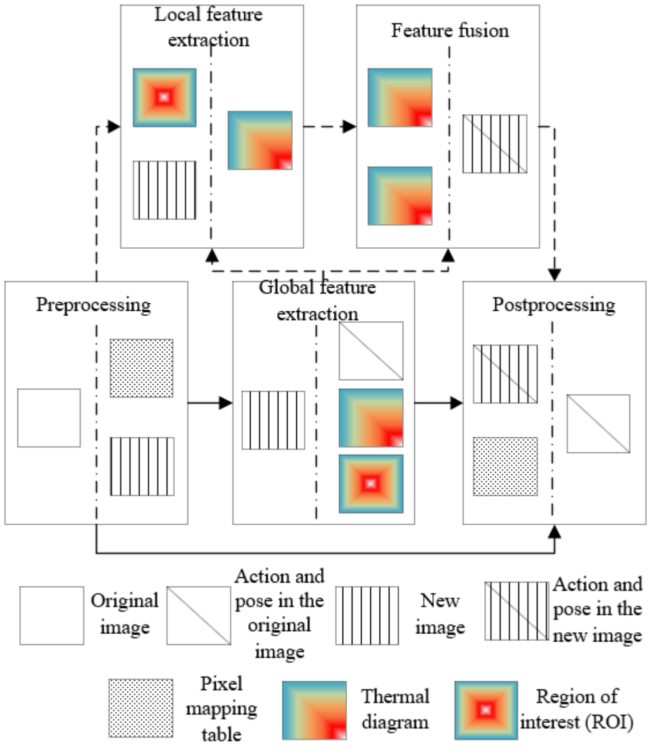


Figure 1. Process of full-effect expression of dance actions

Figure 1 explains the process of full-effect expression of the global and local features for dance actions. It is clear that our models are premised on the fusion between the local and global features of the frames in the dance action video. The dance actions involving *L* joints can be expressed as a vector $H=(\dots, h_i, \dots)^T$, $i\in\{1, \dots, L\}$, where the coordinates of joint *i* are $h_i=(a_i, b_i)$. Let *P* be the data of a dance action image; *H* and H^* are the actual pose and estimated pose, respectively. Then, a labeled image sample of dance actions can be described as (*P*, *H*).

2.1 Image preprocessing

Image enhancement generally includes four steps: rotation, translation, scaling, and flipping. Here, the origin of the coordinates is defined as the first pixel in the upper left corner of the dance action image. Taking the horizontal direction as the *a*-axis and the vertical direction as the *b*-axis, the original dance action image of the size (P_a, P_b) was rotated clockwise by an angle of $\varPsi$, $0\leqq\varPsi\leqq\pi/2$. Then, the coordinates of joint *i* under the rotated coordinate system can be represented by $h_i^s=(a_i^s, b_i^s)$:

$$\begin{cases} a_i^s=\left(a-\dfrac{P_a}{2}\right)cos\varPsi-\left(b-\dfrac{P_b}{2}\right)sin\varPsi+\dfrac{P_a^s}{2} \\ b_i^s=\left(b-\dfrac{P_b}{2}\right)cos\varPsi-\left(a-\dfrac{P_a}{2}\right)sin\varPsi+\dfrac{P_b^s}{2} \end{cases} \tag{1}$$

The size of the rotated dance action image P^s can be described by:

$$\begin{cases} P_a^s=\left(P_b-P_a tan\varPsi\right)sin\varPsi+P_a cos\varPsi \\ P_b^s=\left(P_a-P_b tan\varPsi\right)sin\varPsi+P_b cos\varPsi \end{cases} \tag{2}$$

Let p_1 and p_2 be the coordinates of the upper left and lower right corners in the cropped area of the image, respectively. To translate the rotated image, the first step is to determine the minimum cropped area based on the labels of the dance actions:

$$\begin{cases} p_1=\left(min\left(a_1^s,\dots,a_L^s\right),min\left(b_1^s,\dots,b_L^s\right)\right) \\ p_2=\left(max\left(a_1^s,\dots,a_L^s\right),min\left(b_1^s,\dots,b_L^s\right)\right) \end{cases} \tag{3}$$

By randomly enlarging the minimum cropping area (3), new images *P'* can be obtained for different translation actions. Let $h^p{}_i$ be the coordinates of joint *i* on image *P'*; $\varPhi$ be the displacement of p_1. Then, the labels of dance action image samples can be converted to the new coordinate system by:

$$h_i^p=h_i^s-\left(p_1-\varPhi\right) \tag{4}$$

Translation is followed by scaling. Let $(P^r{}_a, P^r{}_b)$ be the size of the scaled dance action image; K^a and K^b be the index matrices of a-axis and b-axis, respectively; ω^a and ω^b be the weight matrices of a-axis and b-axis, respectively; $s\times e$ be the size of convolution kernel. Then, the four matrices $K^a{}_{P^r{}_a\times s}$, $K^b{}_{P^r{}_a\times e}$, $\omega^a{}_{P^r{}_a\times s}$, and $\omega^b{}_{P^r{}_b\times e}$ can be solved through interpolation.

The pixels of the original image used to derive the l-th pixel on the a-axis of the scaled dance action image can be expressed as the l-th row of the index matrix K^a, whose size is $P^r{}_a\times s$. The pixels of the original image used to derive the l-th pixel on the b-axis of the scaled dance action image can be expressed as the l-th row of the index matrix K^b.

The weight matrices ω^a and ω^b have one-to-one correspondence with K^a and K^b, and characterize how much the relevant original image pixels contained in K^a and K^b contribute to the scaled images.

Based on $K^a{}_{P^r}{}_{a\times s}$, $K^b{}_{P^r}{}_{a\times e}$, $\omega^a{}_{P^r}{}_{a\times s}$, and $\omega^b{}_{P^r}{}_{b\times e}$, it is possible to update the scaled coordinates for the labels of the image samples under the old coordinate system.

The scaling is ensued by horizontal flipping. Let P_{FH} be the flipped image; $(a^r{}_i, b^r{}_i)$ be the coordinates of the left joint in the original image. Then, the flipped coordinates of the j-th right joint, which is symmetric to the left joint, can be expressed as:

$$\begin{cases} a_i^{FH} = P_a^r - a_i^r \\ b_i^{FH} = b_i^r \end{cases} \tag{5}$$

Formula (5) shows that the flipping can be done by calculating the value of a_i^{FH}, and swapping the coordinates of left joints with those of the right joints. Through rotation, translation, scaling, and flipping, the training samples for our dance action recognition model not only grew in number, but also became more diversified.

The bounding box of dance actions can be obtained through target detection on the preprocessed new image and its label (P^{FH}, H^{FH}). Let (p^a, p^b) be the coordinates of the upper left corner in the original image. Then, the are cropped from the original image based on the bounding box can be determined as:

$$\left(a^p, b^p\right) = \left(a - p_a, b - p_b\right) \tag{6}$$

Thus, as long as (p_a, p_b) are known, the coordinates of the cropped area can be mapped to the original image:

$$\left(a, b\right) = \left(a^p + p_a, b^p + p_b\right) \tag{7}$$

The cropped image P_B needs to be further scaled by the above steps. After preprocessing, the joint coordinates of each dance action image can be predicted by our dance action recognition model as $h_{FH}{}^*=(a_{FH}{}^*, b_{FH}{}^*)$. By looking up the pixel mapping table, the predicted values of the dance action recognition model on the original image can be obtained as:

$$\left(a^*, b^*\right) = \left(e^a\left[a_{FH}^*\right] + p_a, e^b\left[b_{FH}^*\right] + p_b\right) \tag{8}$$

2.2 Construction of global model

To realize the adaptive full-effect expression of human dance actions, richer local details must be included, without sacrificing the expression of global features of the actions. This can be achieved through optimization of rough action description. Based on DeepPose, this paper constructed an optimal model to describe dance action recognition and pose estimation.

The traditional DeepPose mainly faces four problems: (1) The dance action recognition and pose estimation are treated as a regression problem, adding difficulty to the training of the neural network model; (2) Some local information of the original image gets lost in the three pooling operations of the model; (3) Global reasoning makes the mapping between images and dance actions highly nonlinear; (4) Global reasoning and local optimization adopt the same neural network.

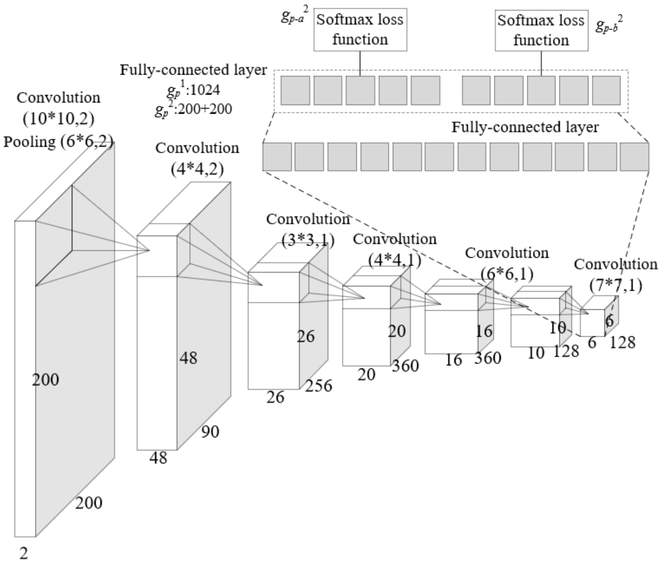


Figure 2. Structure of the global model

This paper improves the DeepPose to address the above four problems. The improved model consists of a global model, and a local convolution model. The structure of the global model is shown in Figure 2.

Let D_P and E_P be the height and width of the pooling window of the global model, respectively; l_P be the step length of each sliding of the pooling window. Then, a convolution can be described as $CON(D_P * E_P, l_P)$, and a max pooling can be expressed as $POO(D_P * E_P, l_P)$. Before and after local response normalization, the value of point (a, b) in the feature map of the i-th dance action can be respectively denoted as $u^i{}_{a,b}$ and $v^i{}_{a,b}$. The number of feature maps in the current layer and that of feature maps participating in calculation can be respectively denoted as M_C and m_C. The hyperparameters can be denoted as γ, δ, and ξ. Then, the local response normalization of the m_C adjacent feature maps can be described as:

$$v_{a,b}^i = \frac{u_{a,b}^i}{\left(\gamma + \delta \sum_{j=max(0,i-m_C/2)}^{min(M_C-1,i+m_c/2)} \left(u_{a,b}^i\right)^2\right)^{\xi}} \tag{9}$$

The above normalization can enhance the generalization ability of dance action features.

The global model contains a fully-connected layer $g_p{}^1$ of 1,024 neurons, and the other fully-connected layer $g_p{}^2$, which serves as the output layer. The latter is equally divided into two parts $g_{p\text{-}a}{}^2$ and $g_{p\text{-}b}{}^2$, which presents the a-axis and the b-axis of the image coordinate system, respectively. To facilitate model training and learning, it is necessary to set independent loss functions for the two fully-connected layers.

Let $c_a=\{c^a{}_1, \dots, c^a{}_{ET}\}$ and $c_b=\{c^b{}_1, \dots, c^b{}_{DT}\}$ be the outputs of $g_p{}^1$ and $g_{p\text{-}b}{}^2$, respectively, with E_T and D_T being the width and height of the input image, respectively. Based on the independence between the outputs of $g_p{}^1$ and $g_{p\text{-}b}{}^2$, either c_a or c_b can be expressed as $c=\{c_1, \dots, c_{DT}\}$. In this way, the action features can be described more clearly. Let P_{mc} be the probability of joint coordinates falling on $a=m_c$ or $b=m_c$. Then, the probability distribution form of the outputs can be transformed by the maximum flexibility function:

$$\begin{cases} P(c)=\left[P_1(c)...,P_{m_c}(c)...,P_{D_T}(c)\right]^T \\ P_{m_c}(c)=\dfrac{e^{cm_c}}{\sum_{j=1}^{D_T}e^{cj}} \end{cases} \tag{10}$$

Let β be the true value; $\Gamma\{\cdot\}$ be the indicator function. If $\Gamma\{\cdot\}=1$, the parameter is true; if $\Gamma\{\cdot\}=0$, the parameter is false. The gap between the predicted and actual dance actions can be measured by cross entropy:

$$Loss(c,\beta)=-\sum_{m_c=1}^{D_T}||\{\beta=m_c\}\,logP_{m_c}(c) \tag{11}$$

The loss function (11) can be minimized through stochastic gradient descent:

$$\begin{aligned} Loss_{c_{m_c}}(c,\beta)&=\left(-\sum_{m_c=1}^{D_T}||\{\beta=m_c\}\log P_{m_c}(cm_c)\right)' \\ &=\left(\sum_{m_c=1}^{D_T}||\{\beta=m_c\}\left(log\sum_{i=1}^{D_T}e^{c_i}-c_{m_c}\right)\right)' \\ &=\sum_{m_c=1}^{D_T}||\{\beta=m_c\}\left(\frac{e^{c_\gamma}}{\sum_{i=1}^{D_T}e^{c_i}}-1\right) \\ &=\sum_{m_c=1}^{D_T}||\{\beta=m_c\}\left(P_\beta(c)-1\right) \end{aligned} \tag{12}$$

Judging by the independent observations of the backpropagation gradients of $g_{p\text{-}a}^2$ and $g_{p\text{-}b}^2$, the main difference between our model and traditional CNN lies in the calculation of both loss functions. Hence, two gradients generated by $g_{p\text{-}a}^2$ and $g_{p\text{-}b}^2$ will propagate reversely through the output layer g_p^2. Both will affect the adjustment and update of network parameters.

Let $g^1=\{g_1^1, \ldots, c_{1024}^1\}$ be the output of g_p^1. For simplicity, the output will be referred to as $g=\{g_1, \ldots, c_{1024}\}$. Besides, the connections between g_p^1 and g_p^2 can be described as $\omega=\{\omega_1, \ldots, \omega_m, \ldots, \omega_{DT}\}$, where ω_m stands for all the connections between the m-th neuron of ω_m and g_p^1. Let (β^a, β^b) and $(\omega_\beta{}^a, \omega_\beta{}^b)$ be the true coordinates of a joint and the weight of the corresponding neuron, respectively; $c^a{}_\beta=\omega^T{}_{\beta a}g$ and $c^a{}_\beta=\omega^T{}_{\beta b}g$ be the output of the β^a-th neuron in $g_{p\text{-}a}^1$ and that of the β^b-th neuron in $g_{p\text{-}a}^2$, respectively. Then, the partial derivative of ω_β can be calculated by:

$$\begin{aligned} L_{\omega_\beta}(c,\beta)&=\left(-\sum_{m=1}^{D_T}\Gamma\{\beta=m\}\log P_m(cm)\right)' \\ &=\left(\sum_{m=1}^{D_T}\Gamma\{\beta=m\}\left(log\sum_{j=1}^{D_T}e^{\omega_j^T g}-\omega_m^T g\right)\right)' \\ &=\sum_{m=1}^{D_T}\Gamma\{\beta=m\}\left(\frac{ge^{c_\beta}}{\sum_{j=1}^{D_T}e^{\omega_m^T}}-\Gamma\{\beta=m\}g\right) \\ &=g\left(P_\beta(c)-1\right) \end{aligned} \tag{13}$$

Substituting the β in formula (13) with β^a or β^b, it is possible to obtain the backpropagation gradient of $g_{p\text{-}a}^2$ or $g_{p\text{-}b}^2$.

2.3 Construction of local convolution model

The global model only outputs the ROI of dance actions. To pinpoint the joints of the dancer, local convolution is needed to prepare a thermal diagram. The global and local results can be combined to obtain a thermal diagram synthetizing visual information on both global and local scales. Figure 3 illustrates the structure of our local convolution model.

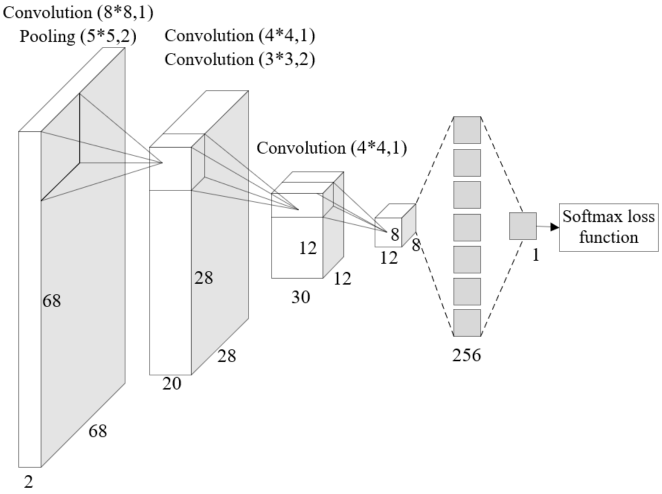


Figure 3. Structure of local convolution model

The outputs of $g_{p\text{-}a}^2$ and $g_{p\text{-}b}^2$ are respectively an N×1 vector c_a and an M×1 vector c_b. The serial number of the neuron that returns the maximum input vector can be denoted as $REmax(\cdot)$. Then, the coordinates of the target joint can be calculated by the serial number of the neuron making the largest response by:

$$(a_i',b_i')=\left(REmax(c_a),REmax(c_b)\right) \tag{14}$$

Let GTD be the global thermal diagram; $NO(c)$ be normalization; $SO(\cdot)$ be smoothing. Then, the two output vectors of the global model can be superimposed into a global thermal diagram:

$$GTD_{N\times M}=SO\left(\begin{bmatrix}NO(c_a)^T\\ \ldots \\ NO(c_b)^T\end{bmatrix}\times\left[NO(c_b),..,NO(c_b)\right]\right) \tag{15}$$

First, the $NO(c_a)^T$ was duplicated for m times, and sorted in rows, creating a matrix; meanwhile, the $NO(c_a)$ was duplicated for n times, and sorted in columns, creating another matrix. Then, the two matrices were multiplied, and the product was subject to Gaussian smoothing. Hence, the authors obtained the global thermal diagram about the distribution of human joints in the target image.

Then, a threshold was set up as $\mu MAX(GTD)$, $\mu\in[0, 1]$, where $MAX(GTD)$ is the largest element value in the global thermal diagram. Then, the pixels whose values are greater than the threshold were selected from the diagram, forming an ROI O. Let t_i be the value of the i-th element. Then, the ROI extraction process can be defined as:

$$O = \left\{u_1, \dots, u_j, \dots\right\}$$
$$u_j = i \times \| \left\{t_i \geq \mu \cdot MAX\left(GTD\right)\right\} \tag{16}$$

The thermal map helps to increase the accuracy of joint positioning. Let GTD_{LC} be the thermal diagram obtained by local convolution; $\xi \in [0, 1]$ be a hyperparameter of the confidence of the global thermal diagram. Then, the results of local convolution model can be fused with those of global model by:

$$\left(a_i', b_i'\right) = REmax\left(\xi \cdot GTD + \left(1 - \xi\right) \cdot GTD_{LC}\right) \tag{17}$$

3. DANCE POSE EVALUATION BASED ON TIME SEQUENCE SEGMENTATION NETWORK

Dance pose evaluation requires the analysis of the frame sequence in the entire video of dance actions. This section designs a dance pose evaluation model based on time sequence segmentation network, and adopts the sparse time sampling strategy to realize efficient and effective learning of the frame sequence in the entire video. Figure 4 explains the workflow of the proposed model. The time sequence segmentation network for video-level prediction consists of a spatial flow convolutional network and a time flow convolutional network.

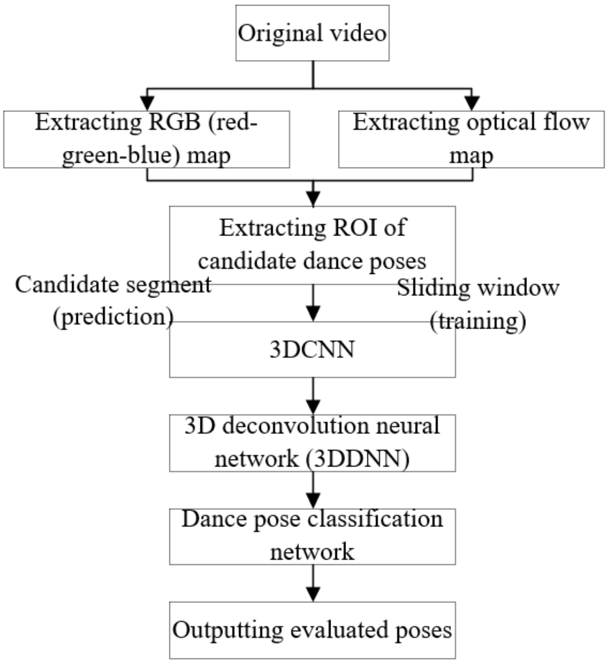

Original video
Extracting RGB (red-green-blue) map
Extracting optical flow map
Extracting ROI of candidate dance poses
Candidate segment (prediction)
Sliding window (training)
3DCNN
3D deconvolution neural network (3DDNN)
Dance pose classification network
Outputting evaluated poses

Figure 4. Flow chart of dance pose evaluation model based on time sequence segmentation network

The given dance action video *SP* was divided into *L* segments of equal length: {L_1, L_2, …, L_s}. Then, these segments form a sequence (R_1, R_2, …R_s). The processing of the CNN on segment R_s can be denoted as $G(R_s; Q)$, where *Q* is the neural network parameter. Let *PC* be the segment consensus function. The consensus between segments on the class of dance pose *s* can be derived from the estimated classes of dance poses based on different segments R_s. The temporal sequence segmentation network can model a series of segments by:

$$T\left(R_1, R_2, \dots R_s\right)$$
$$= F\left(PC\left(G\left(R_1, Q\right), G\left(R_1, Q\right), \dots, G\left(R_s, Q\right)\right)\right) \tag{18}$$

Let *TO* be the number of estimated classes of dance poses; b_i be the true value of class i. Then, the probability that the frame sequence of the entire dance action video belonging to each class can be calculated by the dance pose estimation function *F*. The cross entropy loss function for the standard classification of part of the consensus can be expressed as:

$$Loss\left(b, PC\right) = -\sum_{i=1}^{TO} b_i \left(PC_i - log \sum_{j=1}^{TO} e^{PC_j}\right) \tag{19}$$

The consensus function *PC* can be described as $PC_i = f(G_i(R_1), G_i(R_2), \dots G_i(R_s))$, where *f* is the aggregate function reflecting the final accuracy of pose evaluation by the averaging method. The score of a class PC_i can be obtained based on the averaging by *f*, from the segments belonging to the same class. Figure 5 shows the workflow of the time sequence segmentation network.

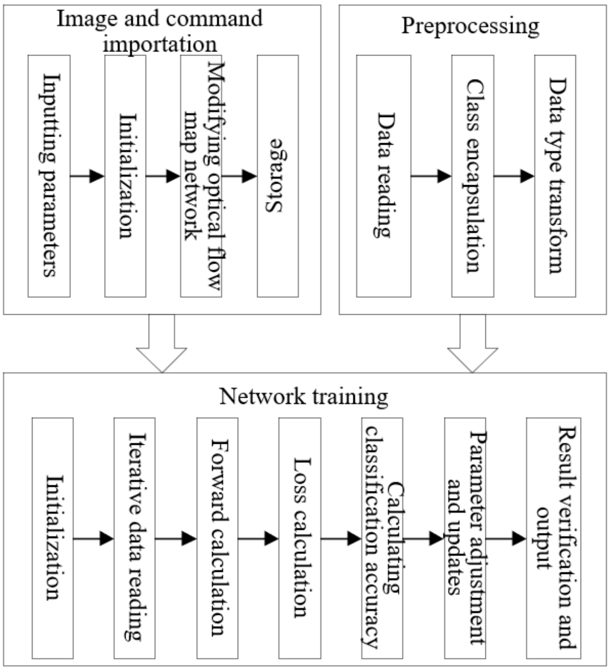


Figure 5. Training flow of the time sequence segmentation network

Let *L* be the number of video segments used by the time sequence segmentation network. The model parameter *Q* can be optimized through backpropagation based on multiple segments. The gradient of *Q* about the loss function can be obtained by:

$$\frac{\partial Loss\left(b, PC\right)}{\partial Q} = \frac{\partial Loss}{\partial PC} \sum_{l=1}^{Loss} \frac{\partial PC}{\partial G\left(R_k\right)} \frac{\partial G\left(R_k\right)}{\partial Q} \tag{20}$$

To position the dance poses based on temporal sequence, this section improves the framework of three-stage 3DCNN. As shown in Figure 6, the improved model consists of a proposal network, a classification network, and a positioning network.

First, the frame size was adjusted to a fixed size. Then, N_T time windows of different lengths yet with an overlap ratio of

3/4 were slid over the original video A. The set of candidate areas $\theta=\{(\delta_t, \theta_t, \Delta\theta_t)\}^{NT}_{t=1}$ generated by A were imported to the proposal network, where θ_t and $\Delta\theta_t$ are the starting time and ending time of the frames in the dance pose video, respectively. Figure 7 shows the flow chart of the precise positioning of starting and ending boundaries of dance actions.

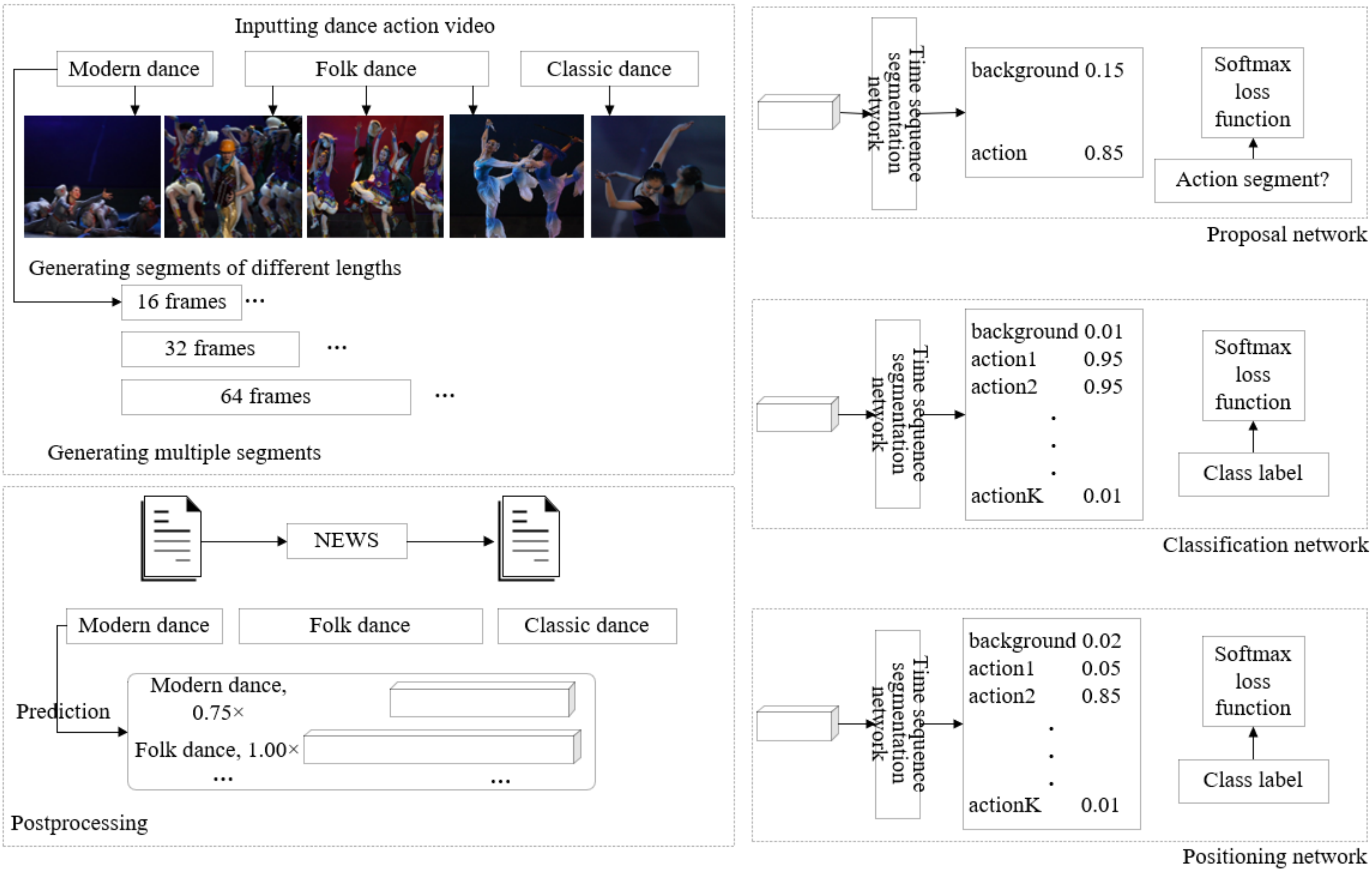


Figure 6. Framework of dance pose positioning model

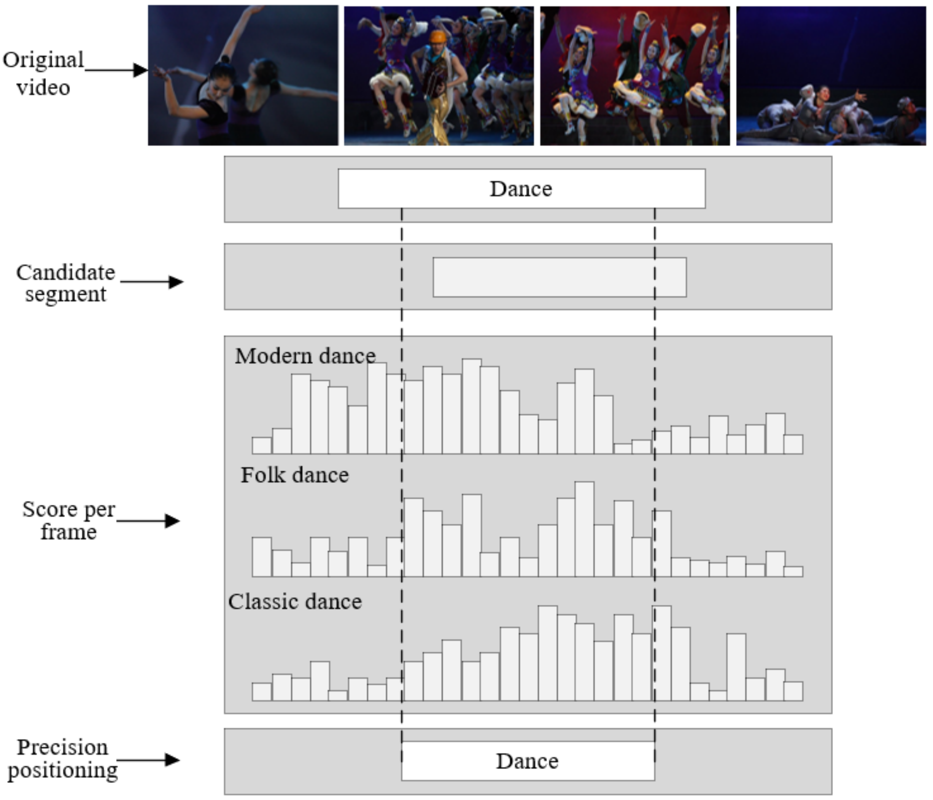


Figure 7. Precision positioning of starting and ending boundaries of dance actions

Responsible for initializing the positioning network, the classification network only works in the training phase. This network outputs the probabilities of *L* classes of dance poses and one class of background. During the training, the samples with an intersect over union (IOU) greater than 0.7 and smaller than 0.3 were judged as positive samples and negative samples, respectively. In the meantime, the background class was sampled according to the average number of samples in each dance pose class.

The positioning network initializes and finetunes the parameters of the classification network. It also outputs the probabilities of dance pose classes and background class. Different from the classification network, the positioning network adopts a loss function related to the degree of overlap in time. Let μ_B be the scale factor. Then, the loss function of the positioning network can be defined as:

$$Loss = Loss_{SM} + \mu_B Loss_{OL} \tag{21}$$

Let RC_m and IOU_m be the true classes of dance poses and IOU in a video segment; OS_m be the output of the positioning network; υ be a hyperparameter. Then, we have:

$$Loss_{OL} = \frac{1}{M}\sum_m \left(\frac{1}{2}\left(\frac{\left(OS_m^{RC_m}\right)^2}{\left(IOU_m\right)^{\upsilon}}\right)-1\right), RC_m > 0 \tag{22}$$

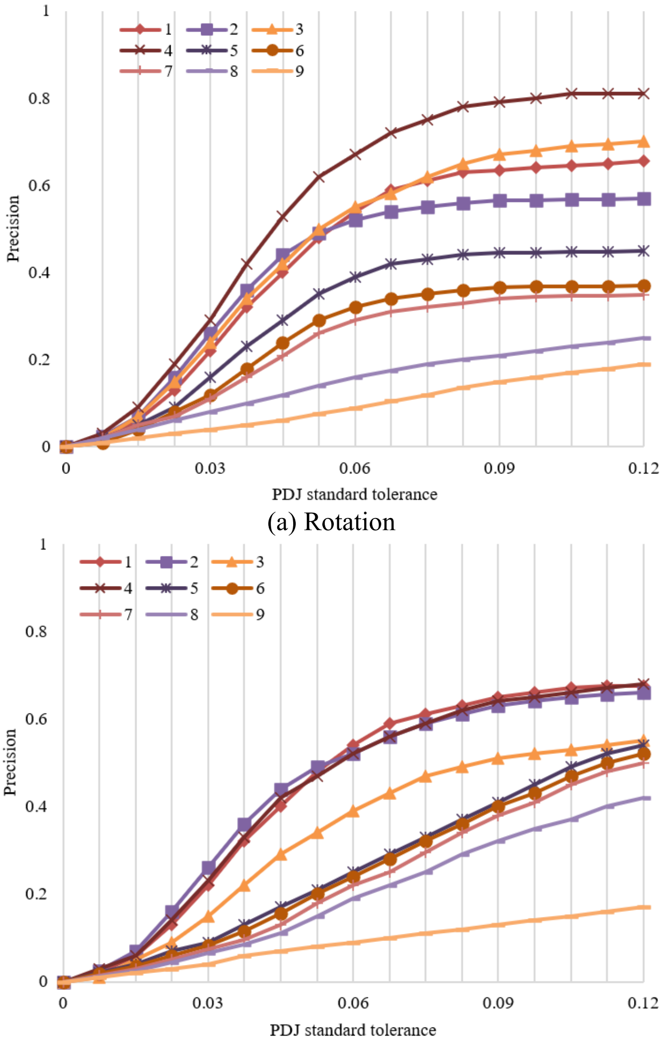
By suppressing non-maximal values and removing overlapped video segments, it is possible to remove the dance pose classes with relatively low scores, leaving only those with relatively high scores.

4. EXPERIMENTS AND RESULTS ANALYSIS

Table 1. Mean average precisions (mAPs) of different models in target detection

IOU threshold	0.1	0.2	0.3	0.4	0.5
Traditional CNN	1.4	0.8	0.6	0.2	0.1
Recurrent neural network (RNN)	18.2	16.5	13.7	11.9	8.7
3DCNN	37.9	37.1	27.5	21.4	15.2
Two-flow CNN	45.8	42.3	36.2	28.8	19.3
Our model	46.2	46.5	37.4	30.6	20.5

The dance pose evaluation model based on temporal sequence segmentation network was compared with three other models through experiments, including traditional CNN, RNN, and two-flow CNN. Table 1 compares the mAPs of different models in target detection. The traditional CNN had a very low recognition rate, and only recognized some simple actions. The RNN improved the recognition rates of the CNN by at least 10% at different IOU thresholds. The 3DCNN further improved the recognition rates by 7-21%, which significantly enhances the precision of dance action recognition. The two-flow CNN continued to improve the recognition rates by 4.5-8%, through full use of the spatiotemporal features of the dance action video, and demonstrated high innovativeness and good practical effect. Our model, that is, the 3DCNN with temporal sequence segmentation network and sparse sampling strategy, achieved better recognition rates than the other models on the ballet dance action image set at different IOU thresholds.



(a) Rotation

(b) Jump

Figure 8. Recognition rates of different classes of dance actions

Next, the full-effect expression of global and local features was compared with other action recognition methods, namely, principal component analysis (PCA), independent component analysis (ICA), Gaussian process latent variable model, random walk model, local linear coordination, Gaussian process dynamic model, scale-variable Gaussian process latent variable model, and hybrid expert model in terms of the precision on dance action image set. The recognition rates of rotation and jump actions are recorded in subgraphs (a) and (b) of Figure 8, respectively. The vertical axis “precision” was obtained through the weighted averaging of the recognition rates of the features for the left and right joint actions of the subjects; the horizontal axis is the PDJ standard tolerance.

Overall, Gaussian process latent variable model and our model achieved relatively high recognition rates on the dance actions. The former was more convenient and easy to operate than our model. Comparing the recognition rates of various classes of dance actions, our model, Gaussian process latent variable model, scale-variable Gaussian process latent variable model, and Gaussian process latent variable model had a large lead in recognition rate, which verifies the superiority of the proposed full-effect expression of global and local features.

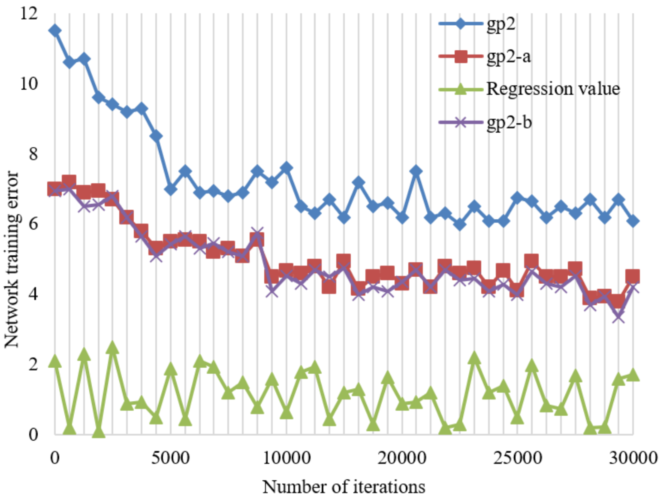
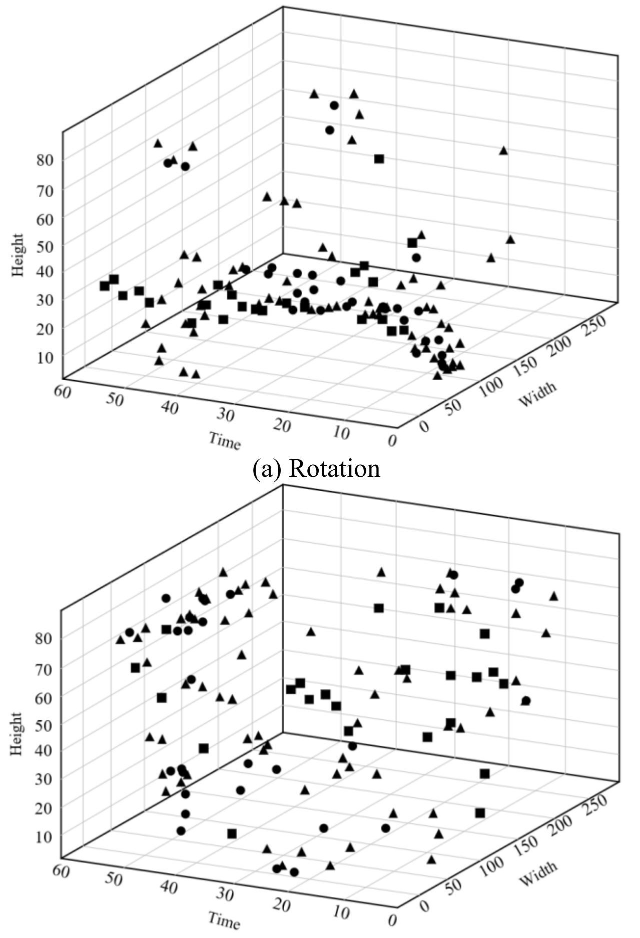


Figure 9. Training error curves of global models

The global network is the foundation of the full-effect expression of global and local features. This paper tests and analyzes the proposed global network by performance and efficiency. According to the training error curves of global models (Figure 9), from the 0th to 18,000th iteration, the loss function values of g_{p-a}^2 and g_{p-b}^2 declined with fluctuations, and tended to be stable at about the 24,000th iteration, while the loss of regression model did not decrease with oscillations. This means the global network can effectively train and learn the useful information for the determination of joint coordinates.

The spatiotemporal distribution of interest points in the ROI of dance action images reflects the evolution of interest points in time and space during the dance movement. Figure 10 presents the spatiotemporal distribution of interest points in different ROIs during rotation and jump actions. Different joint angles are represented by different shapes. Obviously, the interest points of the ROIs of the two actions had different spatiotemporal distributions. That is, the spatiotemporal distribution plays an important role in the recognition of dance actions and the estimation of poses.

Furthermore, 3D dance poses were estimated based on the dance action image set. The training results of four models are compared in Figure 11, including motion history map, space-time body, motion energy map, and our model. The four models were compared in terms of total training time, mean estimation error, the inclusion/exclusion of time factor, and the inclusion/exclusion of dimensionality reduction (Table 2). Figure 12 provides the mean error curves of dance pose estimation at different joint angles. It can be seen that our model consumed a short time in training, and minimized the training error and mean error at different joint angles, under the premise of considering the time factor and data dimensionality.



(a) Rotation

(b)Jump

Figure 10. Spatiotemporal distribution of interest points in different ROIs

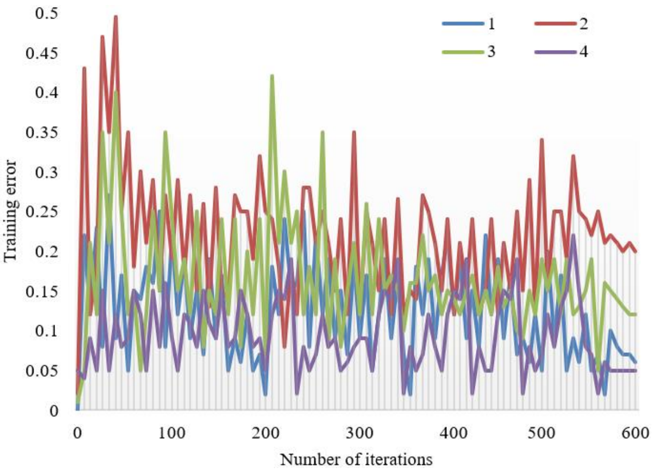


Figure 11. Training results of different pose estimation models

Table 2. Performance of different pose estimation models

Model	Total training time	Mean estimation error	Time factor considered?	Dimensionality reduction?
Motion history map	1 s	17.5	No	Yes
Space-time body	12 s	9.8	No	No
Motion energy map	2 s	8.6	Yes	Yes
Our model	1.5s	6.6	Yes	Yes

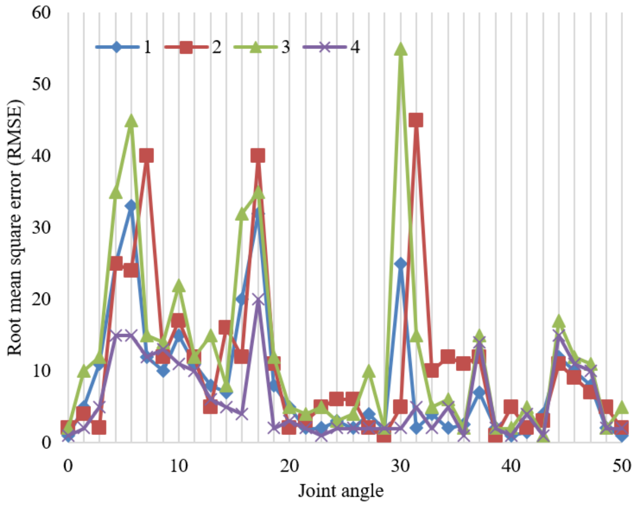


Figure 12. Pose estimation errors at different joint angles

5. CONCLUSIONS

This paper develops a novel DCNN-based strategy for dance action recognition and pose estimation. After detailing the workflow of full-effect expression of global and local features for dance actions, an optimal model was built up based on DeepPose. Next, a dance pose evaluation model was created on temporal sequence segmentation network, and the sparse time sampling strategy was adopted to realize effective and efficient learning of the frame sequence in the entire video. Through experiments, the authors compared the mAPs of different models in target detection, and the recognition rates of multiple methods in dance action recognition. The comparison confirms the superiority of the proposed full-effect expression of global and local features. In addition, experiments were conducted on the spatiotemporal distribution of interest points in different ROIs, and the estimated dance poses of 3D body. The results demonstrate the advantages of our model in total training time, mean estimation error, time factor consideration, and data dimensionality.

REFERENCES

[1] Sargano, A.B., Gu, X., Angelov, P., Habib, Z. (2020). Human action recognition using deep rule-based classifier. Multimedia Tools and Applications, 79(41): 30653-30667. https://doi.org/10.1007/s11042-020-09381-9

[2] Song, S., Lan, C., Xing, J., Zeng, W., Liu, J. (2018). Spatio-temporal attention-based LSTM networks for 3D action recognition and detection. IEEE Transactions on Image Processing, 27(7): 3459-3471. https://doi.org/10.1109/TIP.2018.2818328

[3] Kwon, H., Kim, Y., Lee, J.S., Cho, M. (2018). First person action recognition via two-stream convnet with long-term fusion pooling. Pattern Recognition Letters, 112: 161-167. https://doi.org/10.1016/j.patrec.2018.07.011

[4] Yousefi, B., Loo, C.K. (2018). A dual fast and slow feature interaction in biologically inspired visual recognition of human action. Applied Soft Computing, 62: 57-72. https://doi.org/10.1016/j.asoc.2017.10.021

[5] Seo, J.J., Kim, H.I., De Neve, W., Ro, Y.M. (2017). Effective and efficient human action recognition using dynamic frame skipping and trajectory rejection. Image and Vision Computing, 58: 76-85. https://doi.org/10.1016/j.imavis.2016.06.002

[6] Ullah, A., Ahmad, J., Muhammad, K., Sajjad, M., Baik, S.W. (2017). Action recognition in video sequences using deep bi-directional LSTM with CNN features. IEEE Access, 6: 1155-1166. https://doi.org/10.1109/ACCESS.2017.2778011

[7] Mellet, C.O., Nierengarten, J.F., Fernández, J.M.G. (2017). Multivalency as an action principle in multimodal lectin recognition and glycosidase inhibition: A paradigm shift driven by carbon-based glyconanomaterials. Journal of Materials Chemistry B, 5(32): 6428-6436. https://doi.org/10.1039/C7TB00860K

[8] Shao, Y.H., Guo, Y.C., Gao, C. (2015). Infrared human action recognition using dense trajectories-based feature. Journal of Optoelectronics Laster, 26(04): 758-763.

[9] Iosifidis, A., Tefas, A., Pitas, I. (2015). Distance-based Human Action Recognition using optimized class representations. Neurocomputing, 161: 47-55. https://doi.org/10.1016/j.neucom.2014.10.088

[10] Zhu, G., Zhang, L., Li, H., Shen, P., Shah, S.A.A., Bennamoun, M. (2020). Topology-learnable graph convolution for skeleton-based action recognition. Pattern Recognition Letters, 135: 286-292. https://doi.org/10.1016/j.patrec.2020.05.005

[11] Abdellaoui, M., Douik, A. (2020). Human action recognition in video sequences using deep belief networks. Traitement du Signal, 37(1): 37-44. https://doi.org/10.18280/ts.370105

[12] Khan, M.A., Akram, T., Sharif, M., Muhammad, N., Javed, M.Y., Naqvi, S.R. (2019). Improved strategy for human action recognition; experiencing a cascaded design. IET Image Processing, 14(5): 818-829. https://doi.org/10.1049/iet-ipr.2018.5769

[13] Ozcan, T., Basturk, A. (2020). Human action recognition with deep learning and structural optimization using a hybrid heuristic algorithm. Cluster Computing, 23: 2847-2860. https://doi.org/10.1007/s10586-020-03050-0

[14] Hoshino, S., Niimura, K. (2020). Robot vision system for human detection and action recognition. Journal of Advanced Computational Intelligence and Intelligent Informatics, 24(3): 346-356. https://doi.org/10.20965/jaciii.2020.p0346

[15] Ahmad, T., Mao, H., Lin, L., Tang, G. (2019). Action recognition using attention-joints graph convolutional neural networks. IEEE Access, 8: 305-313. https://doi.org/10.1109/ACCESS.2019.2961770

[16] Fernando, B., Gavves, E., Oramas, J., Ghodrati, A., & Tuytelaars, T. (2016). Rank pooling for action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4): 773-787. https://doi.org/10.1109/TPAMI.2016.2558148

[17] Luvizon, D.C., Tabia, H., Picard, D. (2017). Learning features combination for human action recognition from skeleton sequences. Pattern Recognition Letters, 99: 13-20. https://doi.org/10.1016/j.patrec.2017.02.001

[18] Asteriadis, S., Daras, P. (2017). Landmark-based multimodal human action recognition. Multimedia Tools and Applications, 76(3): 4505-4521. https://doi.org/10.1007/s11042-016-3945-6

[19] Martínez, F., Manzanera, A., Romero, E. (2017). Spatio-

temporal multi-scale motion descriptor from a spatially-constrained decomposition for online action recognition. IET Computer Vision, 11(7): 541-549. https://doi.org/10.1049/iet-cvi.2016.0055

[20] Nguyen, X.S., Nguyen, T.P., Charpillet, F., Vu, N.S. (2018). Local derivative pattern for action recognition in depth images. Multimedia Tools and Applications, 77(7): 8531-8549. https://doi.org/10.1007/s11042-017-4749-z

[21] Aihara, K., Aoki, T. (2015). Motion dense sampling and component clustering for action recognition. Multimedia Tools and Applications, 74(16): 6303-6321. https://doi.org/10.1007/s11042-014-2112-1

[22] Masudul, A.S.M., Kooi, T.J., Hyoungseop, K., Seiji, I. (2015). Human action representation and recognition: An approach to histogram od spatiotemporal templates. International Journal of Innovative Computing, Information and Control, 11(6): 1855-1867.

[23] Ren, H., Kanhabua, N., Møgelmose, A., Liu, W., Kulkarni, K., Escalera, S., Moeslund, T.B. (2018). Back-dropout transfer learning for action recognition. IET Computer Vision, 12(4): 484-491. https://doi.org/10.1049/iet-cvi.2016.0309

[24] Papadopoulos, G.T., Daras, P. (2016). Human action recognition using 3d reconstruction data. IEEE Transactions on Circuits and Systems for Video Technology, 28(8): 1807-1823. https://doi.org/10.1109/TCSVT.2016.2643161

[25] Herath, S., Harandi, M., Porikli, F. (2017). Going deeper into action recognition: A survey. Image and Vision Computing, 60: 4-21. https://doi.org/10.1016/j.imavis.2017.01.010

[26] Iosifidis, A. (2015). Recognition and action for scene understanding. Neurocomputing, 161: 1-2. https://doi.org/10.1016/j.neucom.2015.02.054

[27] Faraki, M., Palhang, M., Sanderson, C. (2015). Log-Euclidean bag of words for human action recognition. IET Computer Vision, 9(3): 331-339. https://doi.org/10.1049/iet-cvi.2014.0018

[28] Perera, A.G., Law, Y.W., Ogunwa, T.T., Chahl, J. (2020). A multiviewpoint outdoor dataset for human action recognition. IEEE Transactions on Human-Machine Systems, 50(5): 405-413. https://doi.org/10.1109/THMS.2020.2971958