Hindawi
Journal of Mathematics
Volume 2023, Article ID 9784351, 1 page
https://doi.org/10.1155/2023/9784351

Retraction

Retracted: Dance Evaluation Based on Movement and Neural Network

Journal of Mathematics

Received 19 December 2023; Accepted 19 December 2023; Published 20 December 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] Y. Lei, X. Li, and Y. J. Chen, "Dance Evaluation Based on Movement and Neural Network," *Journal of Mathematics*, vol. 2022, Article ID 6968852, 7 pages, 2022.

Hindawi
Journal of Mathematics
Volume 2022, Article ID 6968852, 7 pages
https://doi.org/10.1155/2022/6968852

Research Article

Dance Evaluation Based on Movement and Neural Network

Yan Lei, Xin Li, and Yi Jiao Chen

Arts College of Sichuan University, Chengdu, Sichuan 610000, China

Correspondence should be addressed to Yan Lei; robert_dials_00@subr.edu

Received 23 November 2021; Revised 8 December 2021; Accepted 13 December 2021; Published 1 February 2022

Academic Editor: Naeem Jan

In terms of music-driven dance movement generation, the music movement matching model and the statistical mapping model have poor fit between the dance generated by the model and the music self. The generated dance movement is incomplete, and the smoothness and rationality of long-term dance sequences are low. The new dance moves and other related issues cannot be generated by the traditional model. In order to address these issues, we design a dance generation algorithm based on movements and neural networks that will extract the mapping between voice and movement features. In the first stage, where the prosody features and audio beat features extracted from music are used as music features, and the coordinates of key points of the human body extracted from dance videos are used as motion features for training. In the second stage, the basic mapping of music and dance movements are realized through the generator module of the model to generate a smooth dance posture; the consistency of dance and music is realized through the discriminator module; the audio characteristics are more possessed through the Autoencoder module representative. In the third and final stage, the modified version of the model transforms the dance posture sequence into a realistic version of the dance. Finally, a realistic version of the dance that fits the music is obtained. The experimental data is obtained from dance videos on the Internet, and the experimental results are analyzed from five aspects: loss function value, comparison of different baselines, evaluation of sequence generation effect, user research, and quality evaluation of real-life dance videos. The results show that the proposed dance generation model has a good effect in transforming into realistic dance videos.

1. Introduction

The two modalities of vision and hearing are strictly related. As long as the object moves, visual changes will inevitably lead to the production of auditory sound. At present, most machine learning still stays in the learning of information in a single mode. In recent years, with the vigorous development of artificial intelligence technology, the transition from single-modal learning to multimodal learning has become the key to a better understanding of machine perception. More and more researchers have begun to pay attention to the learning of multi-modal information, including cross-modal retrieval, multi-modal information joint make a decision, and cross-modal generation. Cross-modal generation aims to synthesize one modal or several modal data based on the information of different modalities. One-way or two-way generation of text to image, text to video, audio to image, and audio to video are all examples of cross-modal generation.

With the development and popularization of deep learning in recent years, artificial neural networks have been successfully applied to the generation of dance movements. The significant advantage of using deep learning for dance generation is that they can directly extract advanced features from raw data. In addition, deep neural networks can create new dance moves. However, the dance generation algorithm based on deep learning also has some problems. For example, due to the end-to-end model, the generated dance may not be smooth before and after the frames, which will make the visualization effect of the generated dance worse; on the other hand, dances directly generated by algorithms are often difficult to match with music. In addition, for the visualization of dance, people often draw the skeleton of the human body or perform animation processing directly according to the coordinates of the key points of the human body, and there is room for further improvement in the visualization effect.

Dance data often comes from the real world. It is necessary to extract continuous dance pose data from a specific dance video using human pose estimation technology, and design a specific audio feature encoder to extract audio features from the music matched by the dance. The dance data reflects the changes in the coordinates of the key points of the human body in different times. It is a typical time series data, so it has the characteristics of multiscale, multidimensional, and dynamic correlation.

Aiming at the characteristics of dance data, this paper constructs a custom music and dance dataset, which contains about 270,000 frames of music and dance movements. The human body pose estimation technology is used to extract the coordinates of the iconic human skeleton key points as the dance pose features, and the design is designed. The specific music feature encoder proposes a model for dance movement generation based on deep learning and performs end-to-end training on the extracted dance features and music features. The model is optimized through quantitative and qualitative experiments, and a dance generation model that best fits music is obtained. Finally, the generated dance is visualized using the improved Pix2Pix model, and a live-action dance video is obtained. Under the premise of no additional labeling of data, an end-to-end dance generation model is obtained through self-supervised learning, which is useful for intelligent dance teaching, game fields, cross-modal generation, and exploring the relationship between audiovisual information a certain value.

2. Related Work

Current Research Status. The cross-modal generation from audio to video can be divided into three categories: body motion generation, audio-driven image generation, and talking face video generation.

Synthesizing the corresponding face video through speech or music is a typical cross-modal generation task. Early research on the generation of talking faces was mainly to synthesize a specific identity from a dataset based on arbitrary speech and audio. Kumar et al. [1] tried to use delayed LSTM [2] to generate key points synchronized to the audio, and then another network generated video frames conditioned on the key points. This is the first network architecture that uses any text as input to generate the corresponding voice and lip-sync video that syncs photos to reality. Unlike other published methods, their method only consists of a fully trainable neural network and does not rely on any traditional computer graphics methods. The model uses three main modules: Char2 Wav-based text-speech network, delayed LSTM for generating voice points synchronized with audio, and Pix2-Pix-based network for generating videos based on these key points. Subsequently, Chung et al. [3] tried to use an Encoder-Decoder CNN model to learn the correspondence between the original audio and video, which used the joint embedding of face and audio to generate synthetic speaking facial video frames. The model inputs the still image and audio voice segment of the target face and outputs the lip-shaped video of the target face synchronized with the audio. Jalalifar et al. [4] combined RNN and GAN [5] to create a sequence of real faces synchronized with the input audio by two networks. One of them is the LSTM network, which is used to create lip landmarks based on audio input. The other is conditional GAN, which is used to generate facial images based on a given set of lip marks. Together, these two networks can generate a natural speaking face sequence synchronized with the input audio track. Borra et al. [6] further proposed a time consistency method for dynamic pixel loss. Compared with the direct audio-to-image method, this cascade method avoids fitting false correlations between audiovisual signals that are not related to speech content. In order to avoid these pixel jitter problems, the authors strengthened the network's attention to audio-visual related areas and proposed a new dynamic and adjustable pixel-level loss attention mechanism. In addition, in order to generate clearer images with well-synchronized facial motion, they proposed a new regression-based discriminator structure that takes into account sequence-level information and frame-level information.

Cross-modal conversion through audio and image is a kind of cross-modal generation problem. Chen et al. [7] first tried to use conditional generation confrontation network to solve this cross-modal generation problem, realized the mutual conversion of music sounds and corresponding playing instrument pictures, and also realized cross-modal audio-visual mutual generation. The researchers, respectively, defined a sound-image network and an image-sound network to generate images and sounds, respectively. Brahmaiah et al. [8] from the Institute of Automation of the Chinese Academy of Sciences and others considered a cross-modal cyclic generation confrontation network and combined different generation subnetworks into one network and proposed a cross-modal generation model based on the cyclic confrontation generation network. The mutual generation effect between music and pictures is enhanced. Recently, there have been some studies trying to reconstruct facial images from speech fragments. Duarte et al. [9] proposed a deep neural network, which is trained from scratch in an end-to-end manner and directly generates faces from the original speech waveform without any additional identity information. Their model is trained in a self-supervised manner by using naturally aligned audio and video features in the video. Another type of cross-modal generation task is to generate corresponding speech videos from voice or text end-to-end without the intervention of specified rules. Some researchers considered combining acoustic analysis with text [10], demonstrating a method of generating 3D virtual humans from audio signals by inferring the acoustic and semantic characteristics of speech. Through prosodic analysis of acoustic signals and linking them with the semantics of the words spoken, dynamic virtual facial expressions and behaviors are generated, including head movements, eye saccades, gesture, blinking, and staring. Research has shown that their technology is superior to the method of generating virtual humans using only phonetic prosody. Some other researchers have realized the speech of any given speaker through self-supervised training in speech video [11], generated the corresponding speech posture without adding any semantic information, and then synthesized realistic speech video.

3. Method

In this section, the proposed method is described in detail. First of all, the long- and short-term memory (LSTM) network is reviewed and then its extensions, promotions, and improvements are discussed. The updated formula is also presented. Further, the deep learning is used as a base for the construction of the dance generation model. Then, the design of prosody feature extraction is presented. It is followed by the representation of training data. Finally, the generator design is given.

3.1. Long- and Short-Term Memory Network. Long- and short-term memory network is usually abbreviated as LSTM, which is a special type of RNN. It is designed to solve the long-term dependence of recurrent neural networks. It was introduced by Hochreiter and Schmidhuber [12]. It was refined and promoted by many people in subsequent work. LSTM has achieved good results on many time series problems and has been widely used.

The first step of LSTM is to decide what information to discard from the cell. This decision is controlled by a sigmoid layer called the "forgotten gate." For each element in the cell state C_{t-1}, the forgetting gate passes input h_{t-1} and x_t and then outputs a number between 0 and 1, which represents the percentage of information retained from the previous cell state C_{t-1} to the current cell f_t. 1 means "keep all this information," and 0 means "discard all this information." The updated formula of f_t is as follows:

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right). \tag{1}$$

The next step is to decide what new information the model will store in the cell state. This step is divided into two parts.

$$\begin{aligned} i_t &= \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right), \\ \widetilde{C}_t &= \tanh\left(W_c \cdot \left[h_{t-1}, x_t\right] + b_c\right). \end{aligned} \tag{2}$$

Then, update the old cell state to the new state.

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t. \tag{3}$$

Finally, after updating the cell state, the final output result needs to be determined according to the input h_t and x_t. The output will be based on the current cell state and some information will be filtered.

$$\begin{aligned} O_t &= \sigma\left(W_0 \cdot \left[h_{t-1}, x_t\right] + b_0\right), \\ h_t &= O_t \cdot \tanh\left(C_t\right). \end{aligned} \tag{4}$$

3.2. Model Overall Design. Figure 1 shows the overall design of a dance generation model based on deep learning. The gray box represents the processing module or the network module, and the red and blue boxes represent music characteristics and dance gesture characteristics, respectively. The light-orange box represents the loss function setting. As shown in the figure, first perform audio feature extraction and action feature extraction on the dance data, and then input the audio features into the dance generator to obtain the predicted dance posture, and make MSE Loss with the real dance posture; the audio features are obtained through the Autoencoder module. The audio features of the structure are constructed, and the loss of audio reconstruction is made; the predicted dance posture and the real dance posture are sent to the discriminator for discrimination, and the model is trained against loss.

3.3. Design of Prosody Feature Extraction. In the field of sound processing, Mel-Frequency cepstrum is a linear transformation of the logarithmic energy spectrum based on the nonlinear Mel scale of the sound frequency. Mel-Frequency Cepstral Coefficients (MFCC) are the coefficients that make up the Mel-frequency Cepstral spectrum. The frequency band division of the Mel-frequency cepstrum is equidistantly divided on the Mel scale, which is closer to the human auditory system than the linearly spaced frequency bands in the normal cepstrum. Such feature representation can provide better characterization of sound signals in many fields, such as audio compression and speech recognition. In summary, we choose the 24-dimensional Mel spectrum feature and the 8-dimensional tempogram feature as the vector representation of the audio melody, as shown in Table 1.

The design of rhythm feature extraction takes into account that all music has a fixed rhythm; that is, each piece of music has a fixed drum beat, so the rhythm feature can be further extracted from the audio feature. When the audio feature vector representing the melody and the rhythm feature vector representing the rhythm of the drums are used as neural network input together, the model is easier to understand the entire audio feature sequence. These beat characteristics are shown in Table 2 and can be used as beat control signals for the dance generation model. By constructing a feature matrix in the form of a three-dimensional arithmetic sequence, the model can add beat information on the basis of audio features, as shown in Table 2. The beat feature vector of the first dimension is the position of each audio frame in the whole piece of music; the beat feature vector of the second dimension is the position of the audio frame within each beat of the music; the beat feature vector of the third dimension is the position of each audio frame in the music.

3.4. Training Data Representation. In summary, the data extracted from the original dance video for training the dance generation model can be expressed as follows.

Audio characteristics:

$$M_i = \langle m_i^1, m_i^2, \ldots, m_i^{32} \rangle. \tag{5}$$

Beat characteristics:

$$B_i = \langle b_i^1, b_i^2, b_i^3 \rangle. \tag{6}$$

Posture feature:

$$P_i = \langle p_i^1, p_i^2, \ldots, p_i^{36} \rangle. \tag{7}$$

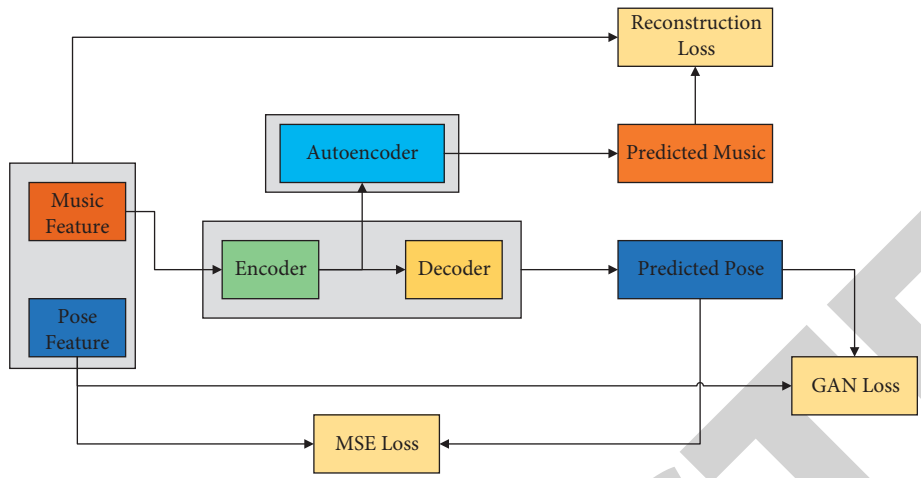

Figure 1: Schematic diagram of the network structure of the dance generative model.

Table 1: Table of music features.

Audio characteristics	Feature dimension
MFCC	$M_1, \ldots, M_{24}$
Tempogram	$M_{24}, \ldots, M_{32}$

Table 2: Table of beats feature.

Rhythmic characteristics	Feature dimension
The position of the entire music audio frame	b_1
The position of the audio frame within the beat	b_2
The relative value of the audio frame within the beat	b_3

Audio data:

$$M = \{M_1, M_2, \ldots, M_n\}. \tag{8}$$

Dance posture data:

$$P = \{P_1, P_2, \ldots, P_n\}. \tag{9}$$

The training data required by the model can be expressed as M and P. Among them, $M = \{M_1, M_2, \ldots, M_n\}$, that is, a total of N frames of audio data; each frame of audio data is represented by M_n; $P = \{P_1, P_2, \ldots, P_n\}$; that is, the dance pose data corresponds to the audio data and the same frame.

3.5. Generator Design. In order to describe the generator design, we classified it into three major phases. The first one is encoder design, which extracts the audio features from the input. Secondly, the process of attention calculation begins. For that reason, a module called attention weight calculation is designed. Lastly, the decoder is designed, which decodes the audio features.

3.5.1. Encoder Design

(1) Encoder Module. In order to extract long-term audio features, the encoder module is composed of multiple layers of LSTM and CNN. The input is the extracted audio feature vector and rhythm feature vector. The output is a music context vector. The specific representation is shown in formulas (10) and (11), where F_1, F_2, F_3 are three convolution kernels, Re LU is the nonlinear activation on each convolution layer, and Encoder Recurrency represents the bidirectional LSTM. The music feature sequence after feature extraction is first sent to a three-layer convolution layer to extract music context information, and then sent to a two-way LSTM to generate the hidden state H of the encoder. As shown in the figure, the input vector $X = \{x_1, x_2, \ldots, x_i\}$. The hidden layer state $H = \{h_1, h_2, \ldots, h_i\}$ in the green box is obtained after coding. After the hidden state of the encoder is generated, it will be sent to the attention network to generate an audio context vector.

$$f_c = \text{Re LU}\left(F_3 * \text{Re LU}\left(F_1 * E(X)\right)\right), \tag{10}$$

$$H = \text{Encoder Recurrency}\left(f_c\right). \tag{11}$$

3.5.2. Design of Attention Calculation Module

(1) Attention Weights Calculation Module. The hidden layer state $H = \{h_1, h_2, \ldots, h_i\}$ and the hidden layer state $S = \{s_1, s_2, \ldots, s_j\}$ can be calculated separately through the encoder module and the decoder module, where H and S are each the hidden state of the encoding layer and the hidden state of the decoding layer at a time step. Then, calculate the Attention Weights and assign the Attention Weights to the music context vector to obtain the audio feature vector after the weight is assigned. Attention calculation occurs at each decoder time step, and the target hidden state and each source state are calculated by a custom score function to generate Attention Weights. In order to reduce potential subsequence duplication or omissions in the decoding process, consider using the cumulative attention weight of the previous decoding process as an additional feature to keep the model consistent when moving forward along the input sequence. So, our model uses a position-sensitive attention mechanism, which is an extension of the previous

attention mechanism. As shown in formula (12), $f_{i,j}$ is the location feature obtained by convolution of the previous Attention Weights, and V_a^T, W, V, U, and B are the parameters to be trained. After the Attention Weights module, the Attention Weights between the hidden state h_j and s_i can be obtained.

$$e_{i,j} = \text{score}\left(s_i, ca_{i-1}, h_j\right) = v_a^T \tanh\left(Ws_i + Vh_j + Uf_{i,j} + b\right). \quad (12)$$

3.5.3. Decoder Design. According to the generated music context vector, the decoding tasks are executed sequentially, and each task focuses on one or several audio feature vectors; that is, different weights are assigned to the audio feature vectors. The encoder part adopts an autoregressive model and uses the predicted value of the dance pose at the previous time step as the input of the next time step to predict the dance pose at the next time step.

In summary, we describe the process of generator training as follows. The music training dataset is $M = \{M_1, M_2, \ldots, M_n\}$, where M_i is a sequence of audio feature vectors. The dance training dataset corresponding to music is $P = \{P_1, P_2, \ldots, P_n\}$, and P_i is the dance posture feature vector corresponding to M_i. The training data of a sample pair composed of $\{M_i, P_i\}$, M, and P are obtained from live dance videos through specific feature extraction schemes. The goal of the model is to train a dance generator G to realize the mapping relationship between M and P. As shown in formula (13), the model is first trained on $\{M_i, P_i\}$, and the MSE Loss is calculated between the dance $G(M_i)$ generated by the model and the real dance P_i. After the training, we input the given music into the training model to get the corresponding dance pose sequence.

$$\mathscr{L}_{\text{MSE}}(G) = \frac{1}{N}\sum_{i=1}^{N}\left\|P_i - G(M_i)\right\|^2. \quad (13)$$

4. Experiments and Discussion

4.1. User Research Results. We mainly conducted user research on the authenticity of the model-generated dance and the consistency between dance and music. First, we investigated whether the dance generated by the model is authentic and credible. We invited 20 observers to conduct a scoring experiment and showed each observer 15 dance fragments generated by five different models according to the dance category. Each observer scored according to the fidelity of the dance. The highest score is 10 points and the lowest score is 0 points. The score of each model is calculated based on the scores of 15 videos by the scorers, and finally the scores of all scorers are averaged to calculate the average value, and then the reality score of each model can be obtained.

According to the data shown in Figure 2, we set the model one to be the LSTM-PCA model, the second model is the LSTM PCA and Discriminator model, the third model is the Generator part of our dance generation model, and the fourth model is the Generator and Discriminator model. The fifth is the Generator and Discriminator and Autoencoder model. As can be seen from the table, in terms of authenticity, our Generator series models are better than other models. Specifically, model one scores 3.61 points, model two scores 5.43 points, model three scores 6.90 points, model four scores 7.85 points, and model five scores 8.52 points.

As can be seen from Figure 3, our Generator series model is better than other models in terms of music consistency. In terms of data, the music consistency of the Kpop dataset is higher than that of other types of dances. This may be due to the large differences in the internal data of the other two dance datasets. For example, the music in the same dance type is quite different. Training is more difficult, but Kpop dance data has no such problem. It also shows that the human body of Kpop dance makes more prominent emotional expressions, and the choreography is more in line with the music. Specifically, model 1 has a score of 4.54 on the Kpop dataset, a score of 2.87 on the Poppin dataset, and a score of 3.19 on the Hiphop dataset; model 2 has a score of 5.61 on the Kpop dataset and a score of 5.61 on the Poppin dataset. The score is 4.32, and the score on the Hiphop dataset is 4.21; the score of model three on the Kpop dataset is 6.54, the score on the Poppin dataset is 5.32, and the score on the Hiphop dataset is 5.39.

The score of model four on the Kpop dataset is 8.01, the score on the Poppin dataset is 7.21, and the score on the Hiphop dataset is 7.45; the score of model five on the Kpop dataset is 9.01 and the score on the Poppin dataset is 7.98; the score in the Hiphop dataset is 7.32. In summary, our model has received the best user reviews compared to other models in terms of dance authenticity and music consistency.

4.2. Image Quality Evaluation Results. As shown in Table 3, using global content discriminator and local time discriminator, even a single frame result, its score is better. Due to the increased lack of attitude perception, the attitude becomes paradoxical, and then the different attitudes are transferred to the frame, which may cause the score to drop. In addition, more significant differences can be observed in our videos.

In order to evaluate the quality of the live-action dance video, BRISQUE [13] is used to evaluate the quality of the live-action dance video. Specifically, different models are used to generate dance poses for the same piece of music, and the same set of live-action generators are used to generate live-action videos. For each live-action video, 100 consecutive video frames are randomly sampled for quality evaluation. According to the quality evaluation results in the table, the effect of adding the Autoencoder module is slightly worse than not adding it. This may be due to the additional loss introduced by Autoencoder, which reduces the generated results.

4.3. Experimental Results. The experimental results of the dance generation model on user research show that for most users, the dance generated by our model exceeds other models in terms of authenticity and musical consistency. This reflects that our model's comprehensive dance generation effect is the best. The experimental results of the dance generation model on the quality of the live-action dance video show that the

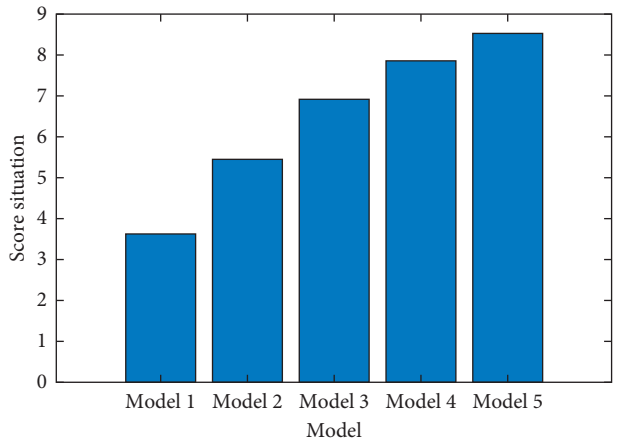

Figure 2: Dance authenticity rating chart.

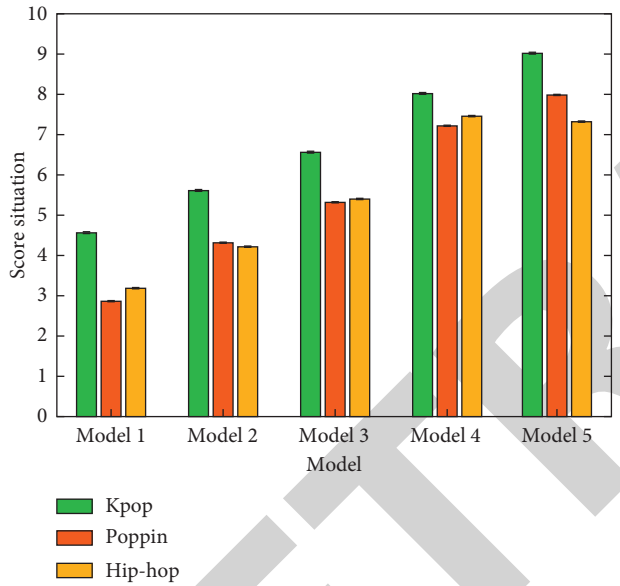

Figure 3: Music consistency rating chart.

Table 3: Image quality assessment score.

Method	BRISQUE
Generator	39.23
Generator-Discriminator	41.40
Generator-Discriminator-Autoencoder	40.37

posture sequence generated by our model is more reasonable, with fewer error or unreasonable posture frames, so the synthesized live-action effect is also the best. This shows that the humanized process we designed is reasonable, and it further reflects that our model has not only achieved good results in sequence generation, but also effective in real human transformation. In summary, our Generator and Discriminator and Autoencoder dance generation model program can effectively extract the characteristics of music, generate dance gesture sequences that fit the music, and transform them into realistic dance videos.

5. Conclusions

The deep leaning-based dance generation algorithm can input music with any type and style. Moreover, it can output the dance posture and real person that the music fits. Researching the related models, this article has completed the following tasks:

It finished reading a large number of domestic and foreign related documents. Furthermore, it understood the current situation and development trend of dance generation algorithms based on deep learning. The combination of the current popular dance generation algorithm and the traditional dance generation algorithm has a couple of problems. The first problem is the difficulty of generating smooth and graceful dance postures. The second problem is the difficulty of matching dance movements with music.

It completed the research on the methods of domestic and foreign dance generation algorithms, combined with the characteristics of music and dance data, and designed audio feature extraction and action feature extraction schemes. The dance generation model is constructed through the extracted audio feature vectors and action feature vectors: in order to achieve a smooth and complete dance sequence, a generator module is designed; in order to achieve a fit between dance and music, a discriminator module is designed; in order to have the extracted audio feature vector, the self-encoder module is designed for better characterization. In order to visualize the effect, the dance posture sequence generated by the dance generation model is transformed into a real person.

The experiments were performed for the purpose of verification in which the dance dataset downloaded from the internet was obtained. It analyzed the five aspects of model loss function value, loss comparison of different models, sequence generation effect, user research, and image generation effect of live-action dance. Experimental results showed that in the feature extraction stage, the use of prosody and rhythm features together as audio features is better than the use of prosody features alone. The use of error frames, missing value differences, and sequence smoothing for dance poses can make the action features smoother. The generation effect is better. In the model building stage, the Generator and Discriminator and Autoencoder model has the strongest generation effect, which can generate a dance posture sequence that is smooth and complete and fits the music. In the stage of live-action dance, the improved Pix2Pix model has also achieved good results in experiments.

The research results play an important reference role for dance generation algorithms. It solves the problem of failure to generate smooth and complete action sequences and dances that fit music in previous studies. It has certain value for intelligent dance teaching, game field, cross-modal generation, and exploring the relationship between audio-visual information. In the future, on the basis of this article, a larger dance dataset can be established to expand the training data to train a more representative and robust dance generation model.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

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

References

[1] R. Kumar, J. Sotelo, K. Kumar, A. de Brebisson, and Y. Bengio, "Obamanet: photo-realistic lip-sync from text," 2017, https://arxiv.org/abs/1801.01442.

[2] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architecture," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[3] J. S. Chung, A. Jamaludin, and A. Zisserman, "You said that?," 2017, https://arxiv.org/abs/1705.02966.

[4] S. A. Jalalifar, H. Hasani, and H. Aghajan, "Speech-driven facial reenactment using conditional generative adversarial net work," 2018, https://arxiv.org/abs/1803.07461.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pp. 2672–2680, Montreal Canada, December 2014.

[6] D. Borra, S. Fantozzi, and E. Magosso, "Interpretable and lightweight convolutional neural network for EEG decoding: application to movement execution and imagination," *Neural Networks*, vol. 129, pp. 55–74, 2020.

[7] L. Chen, S. Srivastava, Z. Duan, and C. Xu, "Deep cross-modal audio-visual generation," in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, pp. 349–357, Mountain View, CA, USA, October 2017.

[8] V. P. Brahmaiah, Y. P. Sai, and M. N. Giriprasad, "A new framework for recognizing normal and epileptic seizure from eye movement signals using genetic based convolutional neural network," *Traitement du Signal*, vol. 37, no. 3, pp. 493–501, 2020.

[9] A. Duarte, F. Roldan, M. Tubau et al., "Wav2Pix: speech-conditioned face generation using generative adversarial networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, May 2019.

[10] P. Koch, M. Dreier, A. Larsen et al., "Regression of hand movements from sEMG data with recurrent neural networks," in *Proceedings of the 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 3783–3787, Montreal, QC, Canada, July 2020.

[11] S. Ginosar, A. Bar, G. Kohavi, C. Chan, A. Owens, and J. Malik, "Learning individual styles of conversational gesture," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3497–3506, Long Beach, CA, USA, June 2019.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, 2012.