Sports Pose Estimation Based on LSTM and Attention Mechanism

Chuanlei Zhang, Lixin Liu(✉), Qihuai Xiang, Jianrong Li, and Xuefei Ren

School of Computer Science and Information Engineering, Tianjin University of Science and Technology, Tianjin 300457, China
18232172760@163.com

Abstract. In our life, we often need to estimate the accuracy of sports pose, which usually costs a lot of time and human resources. To solve the problem, we propose a LSTM-Attention model. In spatial dimension, we use the two-branch multi-stage CNN to extract human joints as features, which not only guarantees the real-time performance, but also ensures the accuracy. For the time dimension, the extracted joint features sequence is input into the LSTM-Attention model for training. In order to verify the effectiveness of our proposed method, we collected data for processing and trained with the proposed model. The experimental results show that our method has a high performance.

Keywords: Sports pose estimation · Two-branch multi-stage CNN · LSTM-attention mechanism

1 Instruction

As one of the research hot spots in the field of computer vision, human action recognition based on video technology has high scientific research value and application value. It includes automatic human behavior detection, recognition and understanding of image sequence of in video. At present, there are a lot of researches on human behavior, but few in motion pose estimation. In the army, it is necessary to conduct physical training and assessment on soldiers regularly, including push-ups, pull-ups and sit-ups. If people supervise and assess them artificially, it will cause the waste of human resources. And the assessment result will have certain emotional color. However, if we record the motion through a camera during the assessment, and then identify and score them, it can save a lot of time and human resources. By analyzing the data, it can reach high accuracy and find many problems that cannot be found by human, so we can correct them in time. Therefore, the research of motion pose recognition has important application value.

Before the emergence of deep learning methods, most of the traditional behavior recognition methods are divided into three steps: (1) Behavior feature extraction. Spare spatial temporal interest points are extracted, for example, Harris corner detection [11] is applied to three-dimensional spatial temporal domain. In addition, there are methods to extract local dense visual features from video data. (2) Description of behavior features.

X. Wang et al. (Eds.): 6GN 2020, LNICST 337, pp. 538–550, 2020.
https://doi.org/10.1007/978-3-030-63941-9_45

The extracted behavior features need to be combined into a standard video description, in which the word bag model is a more commonly used feature description model. (3) The feature description is classified by Support Vector Machine (SVM).

Laptev [12] et al. proposed to extend 2D Harris corner detection operator to three-dimensional domain to extract spatial temporal interest points from video. However, the 3D Harris interest point detection operator is too sparse to describe the behavior accurately, and its robustness is poor, which cannot solve the common problems, such as occlusion, illumination and perspective change in video data. For this reason, Dollar [13] et al. put forward corresponding improved methods using Gabor filter and Gaussian filter to detect the spatial temporal position of interest points in the spatial temporal domain to improve the density of interest points. Although the corresponding problems have been improved, they are not solved perfectly. In 2013, Wang [14, 15] et al. proposed a dense trajectory algorithm and an improved dense trajectory algorithm (IDT) based on the original algorithm. The algorithm is based on the shape characteristics of the trajectory, and integrates the characteristics of hog, HoF and MBF. Although the improved dense trajectory algorithm is a very classical algorithm for manual feature extraction and has good recognition effect, there are problems in practical application: the algorithm speed is relatively slow due to the intensive calculation, and it is difficult to deal with large-scale data sets.

In recent years, researchers have been trying to apply Convolutional Neural Network to video behavior recognition. In 2014, Karpathy [16] et al. proposed to use the pretrained 2D Convolutional Neural Network to extract the spatial features of each frame, and in the final stage, the spatial features of continuous frames were fused to get the classification results, and several fusion methods were investigated. Although the method of deep learning is applied, the experimental results are significantly worse than the algorithm based on artificial design features. There are two main reasons for the failure: the lack of diverse data sets and the inability of network models to effectively extract dynamic features. Simmoyan and Zisserman [3], based on the previous experience of Karpathy and other, proposed a Two-Stream Convolutional Neural Network with spatial network and temporal network. The architecture is no longer a single network to extract spatial features, but has two independent networks, which has a profound impact on the follow-up research. The temporal network extracts the dynamic features of the behavior with the input of stacked dense optical flow vectors. The spatial network extracts the behavior static features from the single video frame as the input, and finally obtains the results through SVM classification. Although this kind of Two-Stream Convolutional Neural Network has good performance, the training process of the two networks is separate, not the end-to-end training process. Du Tran [6] et al. proposed a convolution network of C3D. The network structure no longer uses 2D convolution, but extends to 3D convolution which can deal with temporal information, and computes features simultaneously in the spatiotemporal dimension of video data. The C3D Convolutional Network is pre-trained on Sports-1 M, and then the pretrained model is used on other data sets and can achieve better results. And it was found in the experiments that if artificially designed features such as IDT were used, the model would perform better. It is worth noting that the main advantages of C3D are its operating speed and processing efficiency, which makes it a good application prospect. Feichtenhofer [8] et al. further

use 3D convolution kernel to fuse spatial and temporal networks on the basis of Two-Stream Convolution Neural Network. Limin Wang [17] et al. of the Chinese University of Hong Kong proposed a TSN network structure in 2016. This network structure can extract K short video fragments with the same time in a long video by sparse sampling method, and then randomly sample the fragments from the K fragments. The rest steps are similar to the Two-Stream Convolutional Neural Network, and have achieved better recognition results. In addition, Recurrent Neural Networks (RNN) are also attracting attention because of its ability to process temporal series data. For example, Ng [18, 19] et al. applied Long Short-Term Memory, (LSTM) to the fusion of temporal domain information in a Two-Stream Neural Network, but the effect is average. Long-term Recurrent Convolutional Network (LRCN) [9] extracts feature from single frame image information through convolutional network, and then outputs the features through LSTM in chronological order. The whole architecture is an end-to-end training process. The author also compares RGB and optical flow as input, and finds that the best recognition effect can be obtained by weighting the prediction based on the two inputs.

Motivated by these facts, we proposed an attention-based LSTM architecture for motion pose assessment in videos, which effectively determines the accuracy of motion posture. We take the sequence of the joint point features as input and input it into the LSTM-Attention model, and then take the output of LSTM-Attention model through Softmax as result. The advantage of this paper is that using the architecture of two-branch multi-stage CNN [1] which can accurately and effectively extract the features of human joint points in the video. Inputting the extracted features into the LSTM network can express the serialized features well. Adding the Attention model on the one hand improves the performance of the last model. On the other hand, using the attention mechanism can facilitate the observation of how the information in the input sequence affects the final output sequence, which helps to better understand the internal working mechanism of the model.

The remainder of this paper is organized as follows. Section 2 expounds the theory and design of LSTM-Attention network model. Section 3 designs experiment to verify the feasibility of the proposed method, and analyzes the experimental results. Finally, the conclusions based on this paper are given in Sect. 4.

2 Model Design

For spatial dimension, we use the two-branch multi-stage CNN to extract the joint points. For temporal dimension, we input the sequence of joint feature to obtain the temporal feature. The LSTM-Attention architecture is shown in Fig. 1. There are about four major modules, and we intend to discuss them in details in the following.

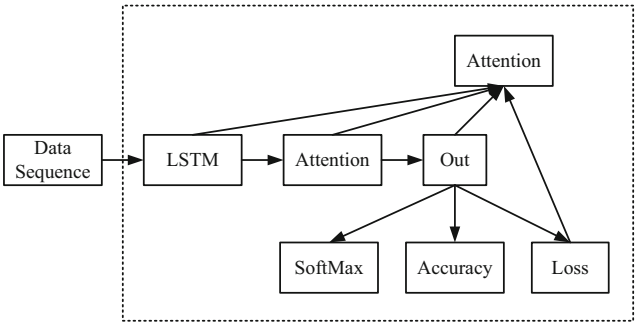

Fig. 1. The LSTM-attention architecture

2.1 The Two-Branch-Multi-stage CNN

The traditional method of pose estimation is top-down, which refers to detecting the human body area first, and then detecting the key points of the human body in the area. Because it is necessary to perform forward key point detection for each detected human body area, the speed is slow. Therefore, the Real-time Multi-Person 2D Pose Estimation [1] presents the first bottom-up representation of association scores via Part Affinity Fields (PAFs), a set of 2D vector fields that encode the location and orientation of limbs over the image domain. Based on the detected joint points and Part Affinity Fields, using the greedy inference algorithm, these joint points can be mapped to different individuals. The network structure [1] is shown in Fig. 2.

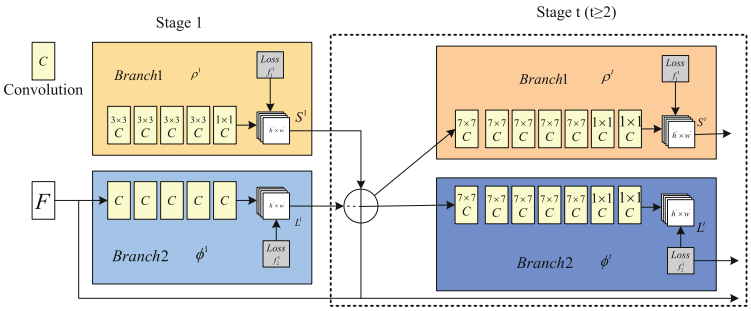

Fig. 2. The architecture of two-branch multi-stage CNN

The network is divided into two branches: the top branch predicts the confidence maps, and the bottom branch predicts the affinity fields. The image is first input to VGG-19, generating a set of feature maps F, which is input to the first stage of each branch. At the first stage, the network produces a set of detection confidence map $S^1 = \rho^1(F)$ and a set of part affinity fields $L^1 = \varnothing^1(F)$, where ρ^1 and $\varnothing^1$ are the CNNs for inference at Stage 1. In each subsequent stage, the predictions from both branches in the previous stage, along with the original image features F, are concatenated and used to produce

refined predictions [1],

$$S^t = P^t\left(F, S^{t-1}, L^{t-1}\right), \forall t \geq 2 \tag{1}$$

$$L^t = \varnothing^t\left(F, S^{t-1}, L^{t-1}\right), \forall t \geq 2 \tag{2}$$

Where ρ^1 and $\varnothing^1$ are the CNNs for inference at Stage t.

2.2 LSTM Neural Network

The recurrent neural network (RNN) is the network structure which can express the time sequence well in deep learning, and the best one is LSTM. Because LSTM operates on sequences, multi-layer LSTM stacking can increase the level of abstraction of the input. When time t increases, the stool can be observed in blocks, or the representation problem on different time scales can make the network extract more abstract features. Therefore, this paper uses multi-layer LSTM stacking to extract features in temporal domain. The motion posture evaluation problem we studied is a typical timing problem, that is, the value of a certain moment is affected by the previous moment or several moments, so we choose the LSTM model.

LSTM is a time-series convolutional neural network, which is derived from recurrent neural networks. By introducing structures called gates, it can mine the time series rules of relatively long intervals and delays in time series. The internal structure of LSTM [2] is shown in Fig. 3. Among them, x_t is he t-th input sequence element value. c is the cell state or memory cell, which controls the transmission of information, and is also the core of the network. i is input gate, which determines how much information is currently reserved for c_t by x_t. f is forget gate, which determines how many cell states c_{t-1} from the previous moment to the current c_t are saved. o is an output gate, which determines how much c_t is passed to the output h_t of the current state. h_{t-1} refers to the state of the hidden layer at time t − 1.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{3}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{4}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{5}$$

$$\overline{c_t} = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{6}$$

$$c_t = f_t \bullet c_{t-1} + i_t \bullet \overline{c_t} \tag{7}$$

$$h_t = o_t \bullet \tanh(c_t) \tag{8}$$

Among them, W_{xi}, W_{xf}, W_{xo} and W_{xc} are the weight vectors from the input layer to the input gate, the forget gate, the output gate and the cell state. W_{hi}, W_{hf}, W_{ho} and

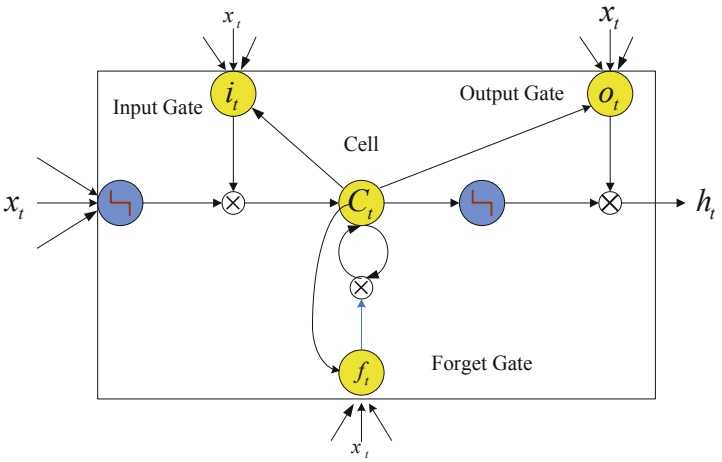

Fig. 3. LSTM internal structure

W_{hc} are the weight vectors from the hidden layer to the input gate, the forget gate, the output gate and the cell gate. b_i, b_f, b_o and b_c are the bias from the input gate, the forget gate, the output gate and the cell gate. $\sigma(\cdot)$ is Sigmoid activation function. tanh() means hyperbolic tangent activation function, which represents vector element multiplication.

Figure 4 shows the LSTM classification model, in which the input layer is $x_0, x_1, x_2, \cdots, x_t$ the corresponding video frame vector, and the upper layer of the input layer is the forward LSTM layer, which is composed of a series of LSTM units. The results of the addition and averaging of the LSTM outputs at all times are then used as the upper-layer representation. Finally, through the softmax layer, the full connection operation is carried out, then the predicted category y is obtained.

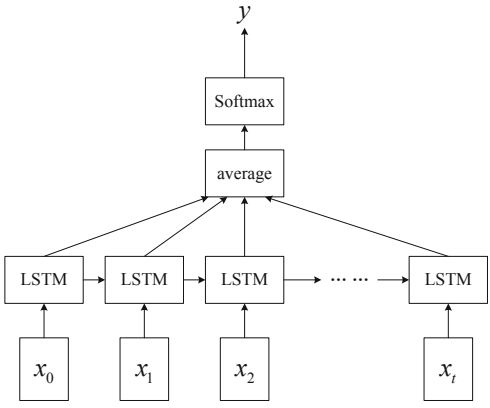

Fig. 4. LSTM classification model

2.3 Attention Mechanism

Attention mechanism is widely used in the field of image processing and natural language processing. Various attention mechanisms have been proposed by researchers, and the recognition effect is remarkable. We introduce the attention mechanism to LSTM, which can extract its own feature information from the input sequence and find the internal relationship between the feature information. It can output the recognition result by weighted average, which improves the recognition accuracy of the model. For a series of weight parameters, the main idea of attention mechanism is to learn the importance of each element from the sequence, and merge the elements according to their importance. On the one hand, adding the Attention mechanism can significantly improve the performance of the model. On the other hand, the attention mechanism can also be used to observe how the information in the input sequence affects the final output sequence, which helps to better understand the internal operation mechanism of the model and facilitate the parameter debugging of some specific input-output.

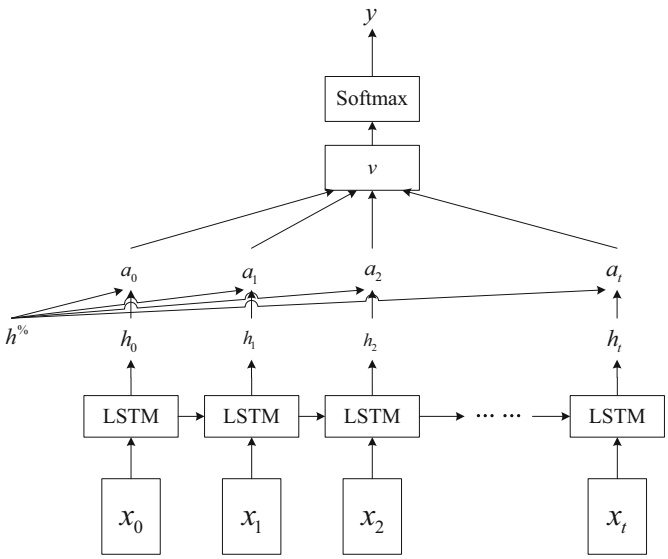

Fig. 5. LSTM-attention classification model

Therefore, in the model construction, we connect a layer of attention network after LSTM to extract temporal features. LSTM-Attention classification model is shown in Fig. 5. The input sequence $x_0, x_1, x_2, \cdots, x_t$ represents the joint point feature sequence of the video frame, which is sequentially input to the LSTM cell to obtain the output $h_0, h_1, h_2, \cdots, h_t$ of the corresponding hidden layer. $\alpha_0, \alpha_1, \alpha_2, \cdots, \alpha_t$ is the weight parameter generated by the attention model, which satisfies the constraint of $\sum_{t=1}^{T} a_t = 1$. h_i is the output state of the hide layer at the i-th time, and h is the feature representation vector one level higher than the video frame. h is initialized randomly as

a parameter, which is updated gradually in the training process. The attention parameter can be computed with the following equation

$$\alpha_i = \frac{\exp(\beta_i)}{\sum_{j=1}^{n} \exp(\beta_j)} \tag{9}$$

where β_j represents the score of the i-th hidden layer output h_i in the video frame representation vector $\overline{h}$. The larger β_j is, the greater the attention of the input in the whole at this moment. It can be computed with the following equation

$$\beta_i = V^T \tanh(W\overline{h} + Uh_i + b) \tag{10}$$

The attention vectors can be obtained according to the outputs of the LSTM network and the temporal attention weight values at each running step with this equation.

$$v = \sum_{j=1}^{t} \alpha_j h_j \tag{11}$$

Finally, the prediction category y can be obtained after the softmax classification function, the formula is as follows:

$$y = soft\max(W_v v + b_v) \tag{12}$$

2.4 Loss Function

The loss function used we use is cross-entropy, which comes from information theory. In order to solve the problem of information measurement, we use the concept of "entropy" in physics to describe the average amount of information contained in the received message. In information theory, the larger the entropy of a message, the larger the information it carries. Simply speaking, in deep learning, cross entropy is to measure the similarity between two probability distributions p and q, which is more suitable to measure the distribution difference between two probabilities. The formula is as follows:

$$H_y \cdot (y) = \sum_i y_i' \log y_i \tag{13}$$

where y is the predicted probability distribution vector of the model output, and y′ is the true distribution. y_i is element 0 or 1 in vector y, which needs to be distinguished from the discrete value of sample i category, that is, y. In vector y, only the y-th element y_y is 1, and the rest are all 0 (one-hot coding). That is to say, the cross entropy only relates to the prediction probability of the correct result, as long as its value is large enough, it can also ensure that the classification result is correct.

3 Experiment and Analysis

To evaluate the effectiveness of LSTM-Attention model, we train the architecture and test the well-trained model on our dataset. We next describe the implementation details of our algorithms and discuss the experiment results.

3.1 Data Collection and Processing

The data of training set, test set and verification set used in our paper are collected by our team. The steps to obtain the data are as follows:

(1) Record all the actions into video. The collected video data is divided into 6 categories, which are push-up front, push-up side, sit-up front, sit-up side, pull-up front, pull-up side. The sample pictures of the dataset are shown in Fig. 6;
(2) Extract a frame every 6 frames of the video, that is, extract about 5 frames of images per second;
(3) Cut out the main part of the human behavior in the picture.

Fig. 6. The sample pictures of the dataset

After data collection, we need to process the collected data to facilitate the extraction of feature points. The steps to process the data are as follows:

(1) We need to classified the images. The captured images are manually marked, and the sequence of pictures marked as the same action is placed in the same folder.
(2) We mark each type of action, and divide it into standard action and non-standard action;

(3) We use the model of the two-branch multi-stage CNN to extract 17 joint features of each image into one-dimensional vector and store them in the CSV file. The extracted information includes joint points identification, relative coordinates, and confidence, so that 68 nodes of information stored into one-dimension vector can be extracted from a picture;
(4) we take 10 frames as a sequence to make the dataset of the model, That is, the joint points data of every ten frames is used as the input of the model.

Finally, the amount of each type of data sequence is shown in Table 1. We have cropped 8044 data sequences in total. We randomly divide the verification set and training set according to the ratio of 1:9. The joint information we extract is relative coordinates, that is, the coordinates relative to the length and width of the picture. In order to improve the generalization ability of the model, we cut the sequence pictures randomly during the process of extracting joint points to ensure that the relative positions of joint points change in turn, so as to expand the data set.

Table 1. The amount of data sequence

	Push-up front	Push-up side	Pull-up front	Pull-up side	Sit-up front	Site-up side
Standard	72	606	139	331	290	411
Non-standard	279	1896	450	1073	827	1670

3.2 Implement Details

We perform the experiment with the following implement details. First, we read the data from the Tfrecord file and put it into the memory buffer, then read a training group randomly with the batch size of 128. This training group is put into the LSTM attention model to perform training operations. Then we calculate the loss and use the optimizer back propagation to reduce the loss and adjust the network parameters of each layer. The optimizer we use is the Adam optimizer [20] provided by tensorflow. The initial value of learning rate is set to 0.0001, and we optimize the model by using learning rate exponential decay.

3.3 Experiment Results and Analysis

First, we train the model on the data set according to the above training process. The accuracy changes and loss changes are shown in Fig. 7 and Fig. 8 respectively. Among them, the blue line represents the training operation, and the orange line represents the test operation. As is shown in the Figures, we can find that after about 2000 batches the model begins to coverage. The accuracy gradually increases, and the loss gradually decrease.

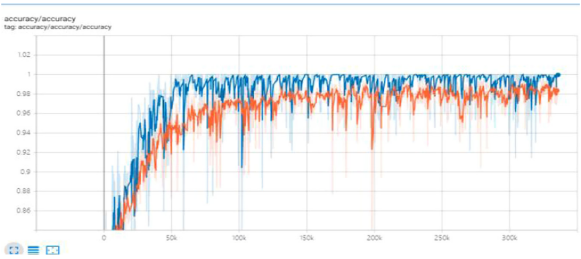

Fig. 7. Accuracy changes

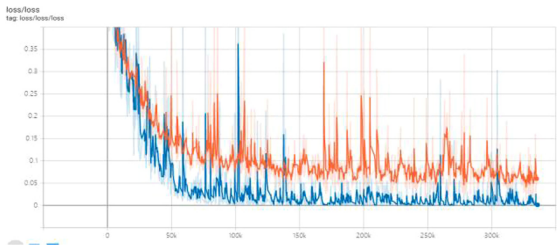

Fig. 8. Loss changes

After 300000 batches, the loss of training set converges to about 0.01, and the accuracy reaches about 0.99; the loss of verification set converges to about 0.09, and the accuracy reaches about 0.97. After training model, we test it in the video. The result is shown in Fig. 9 and Fig. 10.

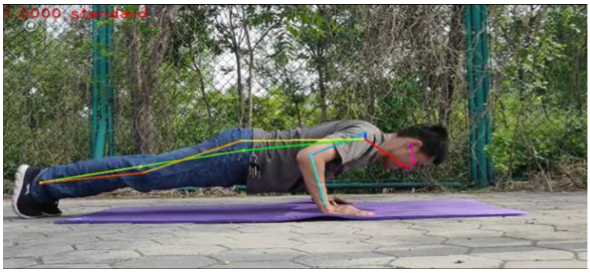

Fig. 9. Standard sports pose

The upper left corner of the picture shows the probability that this motion pose is standard. If the probability is greater than 0.5, it instructs that the motion pose is standard.

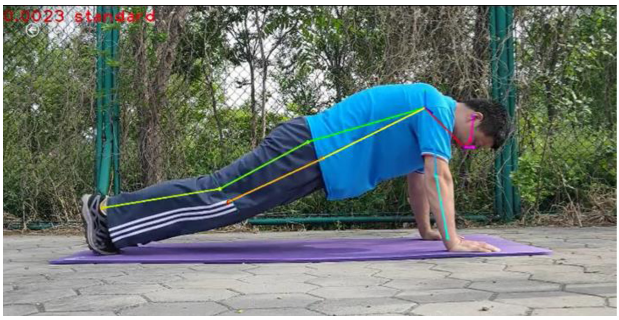

Fig. 10. Non-standard sports pose

4 Conclusion

We propose a sports pose estimation method based on LSTM-Attention network structure. Firstly, we use the two-branch multi-stage CNN to extract human joints as a spatial features. Secondly, the extracted joint features sequence is input to LSTM-Attention model to get the temporal features. The attention Mechanism which can adaptively learn detailed spatial-temporal attention feature to enhance the action recognition at each step of LSTM. Finally, we do some experiment to verify our proposal. The result proves that the recognition accuracy and loss of this method can reach a good state, which proves that the method proposed in this paper has certain significance and value. Later, we will further improve the performance of the method for video data in complex environment. We can expand the training set by collecting data sets in a variety of complex environments, and try to solve the problem of insufficient generalization ability by enhancing the pictures of the training set.

References

1. Cao, Z., Simon, T., Wei, S.-E.: Realtime multi-person 2D pose estimation using part affinity fields. In: CVPR (2017)
2. Dai, C., Liu, X., Lai, J.: Human action recognition using two-stream attention based LSTM. Appl. Soft Comput. J. **86**, 105820 (2019)
3. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos (2014)
4. Liu, J., Shahroudy, A., Xu, D., et al.: Spatio-temporal LSTM with trust gates for 3D human action recognition (2016)
5. Shi, X., Chen, Z., Hao, W., et al.: Convolutional LSTM Network: a machine learning approach for precipitation nowcasting. In: International Conference on Neural Information Processing Systems (2015)
6. Tran, D., Bourdev, L., Fergus, R., et al.: Learning spatiotemporal features with 3D convolutional networks (2014)
7. Xu, H., Das, A., Saenko, K.: R-C3D: region convolutional 3D network for temporal activity detection (2017)
8. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1933–1941 (2016)

9. Donahue, J., Anne Hendricks, L., Guadarrama, S., et al.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2625–2634 (2015)
10. Carreira, J., Zisserman, A.: Quo vadis, action recognition a new model and the kinetics dataset. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4724–4733. IEEE (2017)
11. Harris, C.G., Stephens, M.: A combined corner and edge detector. In: Alvey Vision Conference, vol. 15, no. 50, pp. 5234–5244 (1988)
12. Laptev, I., Marszalek, M., Schmid, C., et al.: Learning realistic human actions from movies. In: IEEE Conference on Computer and Pattern Recognition, CVPR 2008, pp. 1–8. IEEE (2008)
13. Dollar, P., Rabaud, V., Conttrell, G., et al.: Behavior recognition via sparse spatio-temporal feature. In: 2nd Joint International Workshop on Surveillance and Performance Evolution of Tracking and Surveillance, 2005, pp. 65–72. IEEE (2005)
14. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: IEEE International Conference on Computer Vision (ICCV), pp. 3551–3558 (2014)
15. Wang, H., Kläser, A., Schmid, C., et al.: Action recognition by dense trajectories. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3169–3176 (2011)
16. Karpathy, A., Toderici, G., Shetty, S., et al.: Large-scale video classification with convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1725–1732. IEEE Computer Society (2014)
17. Wang, L., et al.: Temporal segment networks: towards good practices for deep action recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 20–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_2
18. Ng, Y.H., Hausknecht, M., Vijayanarasimhan, S., et al.: Beyond short snippets: deep networks for video classification, vol. 16, no. 4, pp. 4694–4702 (2015)
19. Sharma, S., Kiros, R., Salakhutdinov, R.: Action recognition using visual attention. arXiv preprint arXiv:1511.04119 (2015)
20. Abadi, M.: TensorFlow: learning functions at scale. ACM SIGPLAN Not. **51**(9), 1 (2016)
21. Yi, Y., Lin, M.: Human action recognition with graph-based multiple-instance learning. Pattern Recognit. **53**, 143–162 (2016)
22. Pfifister, T., Charles, J., Zisserman, A.: Flowing ConvNets for human pose estimation in videos. In: ICCV (2015)
23. Papandreou, G., et al.: Towards accurate multi-person pose estimation in the wild. arXiv preprint arXiv:1701.01779 (2017)
24. Pishchulin, L., et al.: DeepCut: joint subset partition and labeling for multi person pose estimation. In: CVPR (2016)
25. Tang, P., Wang, H., Kwong, S.: Deep sequential fusion LSTM network for image description. Neurocomputing **312**, 154–164 (2018)
26. Liu, Z., Tian, Y., Wang, Z.: Improving human action recognition by temporal attention, In: 2017 IEEE International Conference on Image Processing, Beijing, China, September 2017, pp. 870–874 (2017)
27. Tompson, J.J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network and a graphical model for human pose estimation. In: NIPS (2014)
28. Ramakrishna, V., Munoz, D., Hebert, M., Andrew Bagnell, J., Sheikh, Y.: Pose machines: articulated pose estimation via inference machines. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8690, pp. 33–47. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10605-2_3