An Improved Human Action Recognition Method Based on 3D Convolutional Neural Network

Jingmei Li, Zhenxin Xu(✉), Jianli Li, and Jiaxiang Wang

College of Computer Science and Technology, Harbin Engineering University,
Harbin 150001, China
18845898726@163.com

Abstract. Aiming at the problems such as complex feature extraction, low recognition rate and low robustness in the traditional human action recognition algorithms, an improved 3D convolutional neural network method for human action recognition is proposed. The network only uses grayscale images and the number of image frames as input. At the same time, two layers of nonlinear convolutional layers are added to the problem of less convolution and convolution kernels in the original network, which not only increases the number of convolution kernels in the network. Quantity, and make the network have better abstraction ability, at the same time in order to prevent the network from appearing the phenomenon of overfitting, the dropout technology was added in the network to regularize. Experiments were performed on the UCF101 data set, achieving an accuracy of 96%. Experimental results show that the improved 3D convolutional neural network model has a higher recognition accuracy in human action recognition.

Keywords: Human body motion recognition · 3D convolutional neural network · Dropout

1 Introduction

As an important research direction in the field of computer vision, human action recognition has a wide range of application value and significant research significance in human-computer interaction, intelligent video surveillance, film animation, video retrieval, virtual reality and other fields. Although human action recognition is a research hotspot in the field of computer vision, there are still many challenges, such as the difference in the performance of the action, the complex background, and the differences in perspectives. At present, human action recognition can be divided into two categories: (1) traditional human action recognition methods; (2) human action recognition methods based on deep learning.

The traditional human action recognition mainly analyzes the video frame sequence, and it manually designs features to manually extract features to identify the action in the video. Bobick et al. [1] based on the assumption that the same actions have similar spatiotemporal data. They reconstruct the data by extracting the foreground parts of the video data, and then identify the actions by comparing the similarity

S. Liu and G. Yang (Eds.): ADHIP 2018, LNICST 279, pp. 37–46, 2019.
https://doi.org/10.1007/978-3-030-19086-6_5

of the foreground data in each video data. Sheikh et al. [2] used the motion trajectory of important human joints such as head, hand, and foot in the process of human motion to determine the similarity between motion samples based on similar invariance. Yamato et al. [3] proposed applying the Hidden Markov Model (HMM) to recognize human motion. Oliver et al. [4] generated a coupled hidden Markov model (CHMM) by applying multiple HMM to model the interactions between humans.

The introduction of deep learning brings new research directions to human motion recognition. The main goal is to construct an effective network recognition framework and to automatically learn features. It does not require manually designing and extracting features, so that the algorithm is not subject to artificial influence and thus has better robustness. Taylor et al. [5] proposed a model to learn the latent representation of image sequences from continuous images. The convolutional structure of the model can be extended to realistic image sizes by using a compact parametric quantization. The model extracts potential "flow file" that correspond to transitions between input frames, and it extracts low-level motion features in a multi-level framework of action recognition. It achieves competitive performance on both KTH and Hollywood2 datasets. Ji et al. [6] proposed a 3D convolutional neural network model for human action recognition. The hard link layer is used to extract pixel information such as grayscale values, horizontal gradients, vertical gradients, horizontal optical streams, and vertical optical streams in consecutive input video frames. And above information is alternately convolved and subsampled to combine information to get the final feature description. Experiments on the KTH and TRECVID 2008 datasets have shown more performance than some benchmarking methods. Experiments on KTH and TRECVID2008 datasets show that it is more effective than some traditional methods.

2 Network Model Overall Design

Ji et al. [7] proposed a 3D convolutional neural network for video in 2013. They extended the convolutional neural network to three dimensions in order to extract spatial and temporal information in the video. It is proposed that the gray image of the continuous seven frames of the video combined with the horizontal gradient, the optical flow, the vertical gradient, and the optical flow feature map have a total of 5 channels as the input of the 3D CNN. Due to the very small size of the 3D CNN network at the time, some problems will arise. For example, in each layer of 3D CNN, the number of kernels is very small, which leads to fewer types of features learned at each layer, and weaker network expression capabilities. Its first convolution layer has 10 kernels and the second convolution layer has only 30 kernels. In order to solve the above problems, the paper uses the method of increasing the number of convolution layers to extend the convolutional layer to four layers. It including two linear convolution layers and two MLP (Multi-Layer Perceptron) convolutional layers, and then increase the number of network layers. It increases the abstraction capability of the network and use dropout to regularize the network to prevent over-fitting of the network.

2.1 Network Model Process Design

The network model process design is shown in Fig. 1. It mainly includes two parts: The first part mainly selects the parameters of the model. First, the data set is downloaded and pre-processed. And the pre-processed data is put into the model for training. Finally, the analysis of the output results to determine the impact of the input data on the performance and stability of the model. The second part is to load the trained model into the classification program. The classification algorithm used is the 3D CNN algorithm designed in this chapter.

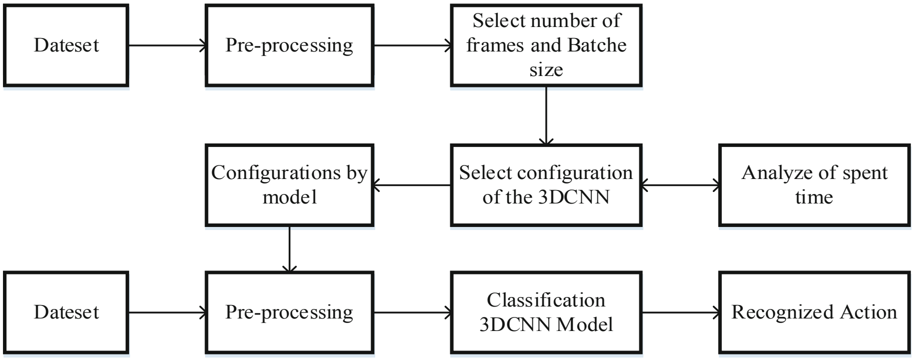


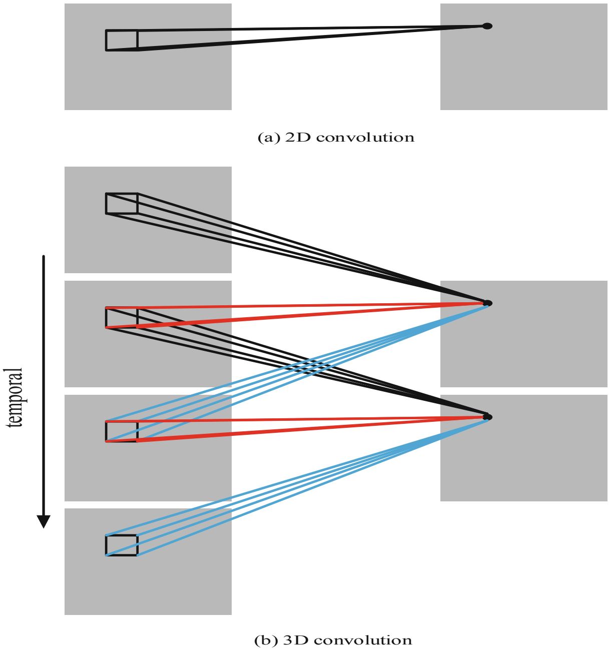
Fig. 1. Network model process design

2.2 3D Convolutional Neural Network

3D CNN can simultaneously compute two dimensional features in both time and space. It is achieved by convolving a 3D convolution kernel in a space consisting of several consecutive frames of image. Because of this structure, the feature map in the convolutional layer is connected to a plurality of frames in the next layer so as to capture motion information. The difference between 2D convolution and 3D convolution is clearly shown in Fig. 2. The 3D convolution formula is:

$$v_{ij}^{xyz} = \tanh(b_{ij} + \sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)}) \quad (1)$$

In Formula 1, $\tanh(\cdot)$ is a hyperbolic tangent function, b_i is the offset of the feature map, R_i is the size of the three-dimensional convolution kernel in the time dimension and w_{ijm}^{pqr} is the value of the convolution kernel connected to the m feature maps in the previous layer at the point (p, q, r).



(a) 2D convolution

(b) 3D convolution

Fig. 2. The difference between 2D convolutional neural network and 3D convolutional neural network

2.3 NIN (Network in Network)

The forward neural network or multi-layer perceptron (MLP) is a non-linear model with strong abstraction ability. Therefore, adding MLP to the network will definitely increase the abstraction ability of the network. The linear convolutional layer and MLP convolution layer are shown in Fig. 3. The MLP convolutional layer uses a nonlinear activation function (ReLU in this paper) so that the feature map of the current layer is obtained. The MLP convolutional layer is calculated as shown in Formula 2:

$$\begin{aligned} f^{1}_{i,j,k_1} &= \max(w^{1^T}_{k_1} x_{i,j} + b_{k_1}, 0) \\ &\ldots.. \\ f^{n}_{i,j,k_n} &= \max(w^{n^T}_{k_n} f^{n-1}_{i,j} + b_{k_n}, 0) \end{aligned} \tag{2}$$

In formula 2, (i,j) is the pixel index of the feature map, $x_{i,j}$ is the input at the center (i,j), k_n is the index of the feature map, n is the number of layers of the MLP. Because the ReLU activation function is used, the maximum value compared to 0 is used in all formulas.

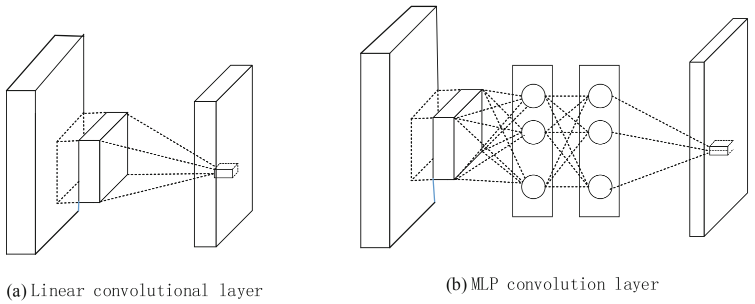


Fig. 3. Linear convolutional layers and MLP convolutional layers

2.4 Network Architecture Design

The network architecture used in the experiment is shown in Fig. 4. It contains two hidden layers and four convolutional layers (two linear convolutions and two nonlinear convolutions). All convolutional layers use a 3 × 3 × 3 convolution kernel size and set the number of MLP layers to 3. For the pooled layer, 2 × 2 × 2 non-overlapping sliding windows are used. At each MLP layer and pooling layer, a Dropout regularization layer with a probability of 0.5 is used to prevent over-fitting of the network. Then there are a fully connected layer with 512 output neurons and a Dropout hidden layer with a probability of 0.25. Finally, there is a fully connected layer with 50 output neural units. The network has about 2.3 million parameters to be trained. This article chooses the Xavier method to initialize the network, and the activation function selects ReLU.

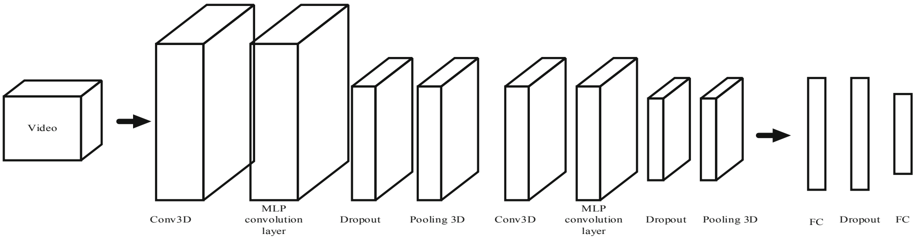


Fig. 4. Network architecture design

3 Experiment

3.1 Data Set Introduction

In 2012, scholars such as Soomro produced the UCF101 [8], the largest human action data set at that time, which included 101 types of actions and more than 1,300 video segments. And these actions can be roughly divided into five types, which are human and object interaction, human body movement, human-human interaction, playing musical instruments and sports. The video clips are all from the YouTube website, the frame rate is fixed at 25 fps, and the image frame size is 320 × 240. In addition, due to the complexity and diversity of videos on the YouTube site, the database is affected by factors such as insufficient light, background clutter, and intense camera shake. Figure 5 shows some of the sample frames in the UCF101 database.

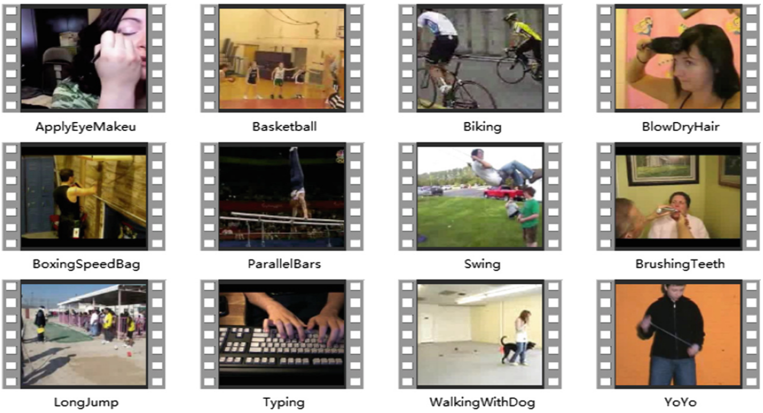


Fig. 5. Partial sample video image frames in UCF101 database

3.2 Experimental Analysis and Design

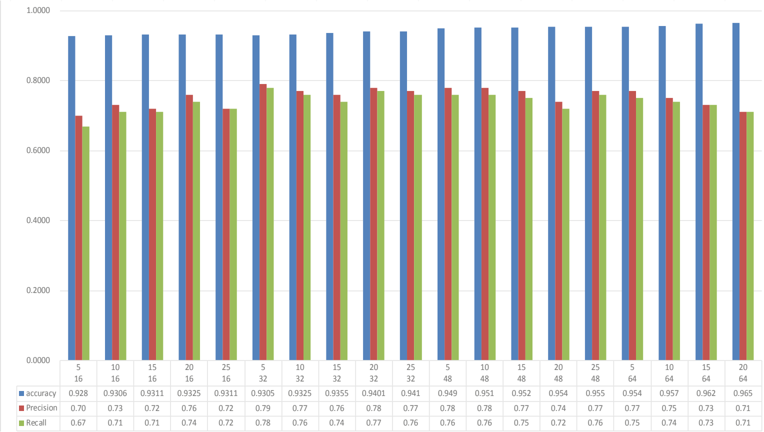
According to the flow chart shown in Fig. 1, this experiment is divided into two parts. The first part: pre-processing the downloaded data set and parameter selection of the model; the second part: for the first part of the parameter selection, the data Set to the model for training.

In the first part, the first step is to preprocess the UCF101 data set, and the second step is to set the parameters of the model. First, the images needed for the experiment are extracted from the video and the images are converted to grayscale images. Then the resolution of these grayscale images is adjusted to four resolutions (16, 32, 48, 64). The adjustment process is to take values for every row and column of the image. Finally, the number of consecutive image frames required to represent the video is set, and the number of consecutive frames is defined as (5, 10, 15, 20, 25) five consecutive image frames. After setting this parameter value, use 60% of the sample for training, 20% for cross-validation, and 20% for model testing to train the model. After 10 trainings of this model, the performance indexes of these models are respectively counted: accuracy rate, precision rate, and recall rate. Finally, these models were analyzed to determine which configured model was able to achieve the highest degree of match and the shortest processing time.

In the second part, the data is preprocessed using the same preprocessing method as in the first part, and then the model already defined is used, in accordance with the Leave-One-Group-Out (LOGO) proposed by Reddy and Shah [9] in 2013. The protocol trains the model. The model was trained 30 times, and then the test set was tested using the model, and the test results were compared with the test results obtained by other methods. Finally, according to the data partitioning criteria in the first section, the datasets were again randomly divided by the same segmentation ratio and the models were tested. The model's observations have the following three parts: accuracy, accuracy, and recall.

3.3 Analysis of Results

Different resolutions are used as input to the model to evaluate the effect of different resolutions on the model. The results are shown in Fig. 6. In terms of image resolution, we can see from Fig. 6 that when the resolution of the input image is between 48 ∗ 48 and 64 ∗ 64, the model's accuracy index achieves better results. All values in this range are Must be higher than 94.9%. Especially when the resolution of the image is 64 ∗ 64, the average accuracy of the model is 95.6%. In addition, another result can be obtained from Fig. 6: When the input image resolution is increased, the accuracy of the model is also improved. However, due to the limitations of the computer's memory, it is not possible to use the 25-frame resolution 64 ∗ 64 image to train the model. In the input continuous image frame, it can be seen from Fig. 6 that when the input continuous image frame is 5 or 10, the model can obtain better accuracy. Despite this, the accuracy of the model and the average recall rate were 74.0% and 74%, respectively.



	5 16	10 16	15 16	20 16	25 16	5 32	10 32	15 32	20 32	25 32	5 48	10 48	15 48	20 48	25 48	5 64	10 64	15 64	20 64
🟦 accuracy	0.928	0.9306	0.9311	0.9325	0.9311	0.9305	0.9325	0.9355	0.9401	0.941	0.949	0.951	0.952	0.954	0.955	0.954	0.957	0.962	0.965
🟥 Precision	0.70	0.73	0.72	0.76	0.72	0.79	0.77	0.76	0.78	0.77	0.78	0.78	0.77	0.74	0.77	0.77	0.75	0.73	0.71
🟩 Recall	0.67	0.71	0.71	0.74	0.72	0.78	0.76	0.74	0.77	0.76	0.76	0.76	0.75	0.72	0.76	0.75	0.74	0.73	0.71

Fig. 6. Image resolution and the number of input frames

In addition to the image resolution and the number of consecutive frames, the execution time of the model is also considered. Figure 7 shows the time spent performing each model. From Fig. 7 the relationship between the input image resolution and the total number of images can be directly obtained. For example, when 16 ∗ 16 resolution is applied, the execution time of the model does not change substantially. However, when the resolution of the image is increased, the execution time of the model will increase substantially linearly.

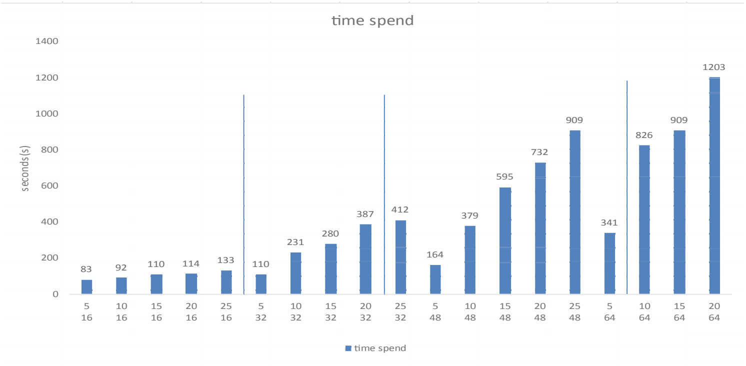


Fig. 7. Time spent running

A comprehensive comparison of each group of Figs. 6 and 7 is performed. When the input image resolution of the model is 64 ∗ 64 and it is a continuous 20-frame image, the execution time of the model is 1200 s. On the other hand, when the input image of the model is 32 ∗ 32 and is a continuous 5-frame image, the execution time of the model only takes 83 s. However, both are roughly the same in accuracy, recall and accuracy. Therefore, comparing the results in the two graphs, we can see that when the resolution of the input image is 32 ∗ 32 and the number of consecutive image frames is 5, the model not only has a shorter execution time, but also has a higher accuracy. Recall rate and accuracy. Therefore, the selection of an image with an image resolution of 32 ∗ 32 and a continuous five-frame image is selected as the input of the model to identify the action.

After selecting the image resolution and continuous image frame parameters of the model, the training model of this parameter is then compared with other similarly used protocols proposed by Reddy et al. [10]. The comparison results are shown in Table 1. In the method proposed by Reddy et al. [10] in 2013, they combined the optical flow, 3D-SIFT and PCA methods to extract the features in the video, and then used the SVM classifier to classify them. Achieved 76.9% accuracy. In the methods of Wang and Schmid et al. [10] in 2014, they used optical flow and dense trajectories to kick off the features of the frames, and then used SVMs to classify them, and finally achieved an accuracy of 91.2%.

Table 1. This method is compared with the above method

Method	Accuracy
Reddy and Shah	76.9%
Liu et al.	87.16%
Wang and Schmid	91.2%
Peng et al.	92.3%
Method of this article	96%

In the method proposed by Liu et al. [11] in 2015, they used visual word bags to extract features for use in motion recognition. Later, they used multi-perspective ideas for features by acquiring data similarities. Finally, they applied the extracted features to a linear SVM, which ultimately achieved an accuracy of 87.9%. In the method proposed by Peng et al. [12] in 2016, they used descriptors such as HOG, HOF, and MBH to extract features, combined these features into the output of descriptors in the bag of words, and then used SVM to perform human motion recognition. Finally, 92.3% accuracy was obtained on the UCF101 dataset. The method proposed in this paper obtains a 96% specific accuracy rate on the UCF data set. Compared with the previous method, the accuracy rate is increased by 3.7%.

4 Conclusion

This article first describes the 3D convolutional neural network and its application in human actions. Then, in order to address the deficiencies in its network, this paper makes targeted improvements: First, to address the problem of a small number of convolution kernels in the network, Two layers of MLP nonlinear convolutional layers are added in the network layer, which not only increases the number of convolution kernels in the network, but also improves the abstraction capability of the network model. Second, in order to prevent over-fitting phenomenon in the network, A layer of Dropout with a probability of 0.5 was added before each pooling layer, and a layer of Dropout with a probability of 0.25 was added before the last fully connected layer. Experiments on the data set UCF101 show that the accuracy of the proposed network model is improved by 3.7% compared with the method in Table 1. Therefore, the human motion recognition method proposed in this paper has certain research significance and practical value.

Acknowledge. This work was supported by the National Key Research and Development Plan of China under Grant No. 2016YFB0801004.

References

1. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. IEEE Trans. Pattern Anal. Mach. Intell. **23**(3), 257–267 (2001)
2. Sheikh, Y., Sheikh, M., Shah, M.: Exploring the space of a human action. In: Tenth IEEE International Conference on Computer Vision, vol. 1, pp. 144–149. IEEE (2005)
3. Yamato, J., Ohya, J., Ishii, K.: Recognizing human action in time-sequential images using hidden Markov model. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 379–385. IEEE Computer Society (1992)
4. Brand, M., Oliver, N., Pentland, A.: Coupled hidden Markov models for complex action recognition. In: Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1997, p. 994. IEEE (2002)
5. Taylor, G.W., Fergus, R., LeCun, Y., Bregler, C.: Convolutional learning of spatio-temporal features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6316, pp. 140–153. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15567-3_11

6. Ji, S., Xu, W., Yang, M., et al.: 3D Convolutional neural networks for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 221–231 (2012)
7. Ji, S., Xu, W., Yang, M., et al.: 3D convolutional neural networks for automatic human action recognition. IEEE US8345984 (2013)
8. Soomro, K., Zamir, A.R., Shah, M.: UCF101: a dataset of 101 human actions classes from videos in the wild. Computer Science (2012)
9. Reddy, K.K., Shah, M.: Recognizing 50 human action categories of web videos. Mach. Vis. Appl. **24**(5), 971–981 (2013)
10. Wang, L., Qiao, Y., Tang, X.: Action recognition with trajectory-pooled deep-convolutional descriptors. In: Computer Vision and Pattern Recognition, pp. 4305–4314. IEEE (2015)
11. Liu, J., Huang, Y., Peng, X., et al.: Multi-view descriptor mining via codeword net for action recognition. In: IEEE International Conference on Image Processing, pp. 793–797. IEEE (2015)
12. Peng, X., Wang, L., Wang, X., et al.: Bag of visual words and fusion methods for action recognition. Comput. Vis. Image Underst. **150**(C), 109–125 (2016)