International Journal of Computational Intelligence and Applications

THE HUMAN ACTION RECOGNITION RESEARCH BASED ON DEEP NEURAL NETWORK

JUAN CHEN[M1] *

SHU-UTS SILC Business School, Shanghai University,

Shanghai 201899, China;

chenjuan82@shu.edu.cn; Tel.: 86-10-69980028-55051

ZHIJUN ZHONG

SHU-UTS SILC Business School, Shanghai University,

Shanghai 201899, China;

18817200584@163.com

As one of the hottest research fields in recent years, human action recognition can be widely applied in various fields in real life. Based on the deep neural network, a series of studies on HAR is developed. Firstly, the relevant theory of HAR is briefly introduced from the two aspects of feature extraction algorithm and classifier. Secondly, the two novel networks on the fundament of 2D deep neural networks and 3D convolutional neural networks are proposed. As for 2D, aiming at the poor performance with the long-term recurrent convolutional network algorithm, an improved network, which replaces the AlexNet network used to extract spatial features with a deeper ResNet-34 network, is presented. With respect to 3D, the article also put forward an ungraded action recognition algorithm based on 3D CNN. The algorithm uses the architecture with eight layers of 3D convolution and five layers of pooling. Furthermore, the experimental results on the UCF101 human behavior dataset show that the proposed improved methods outperform original methods on the HAR tasks.

Keywords: Human Action Recognition; Deep Neural Network; LRCN Algorithm; Features Extraction; 3D CNN.

1. Introduction

Human action recognition (HAR) has been widely used in diverse fields since its advent, such as games, animation, video surveillance, human-computer interaction, robotics, intelligent housing systems and so forth.[1] In recent years, artificial intelligence has developed rapidly and computer performance has gradually enhanced. Meanwhile, deep learning algorithms have made great progress in the realms of image recognition, natural language processing, object detection, and so on, which contributes to increasing researches on the video-based deep learning algorithm. However, as the different background and light conditions, the variability of perspectives, the complexity of human behavior, and the large gap between similar behaviors, the research is extremely challenging.

Currently, HAR research has been undergoing consistent development, whereas this topic still faces some knotty challenges. The challenges are as follows:

(1) Since it is difficult to capture the deep features of motion, deep neural network (DNN) only relying on the apparent features of human behavior has trouble in obtaining accurate and effective motion information. Despite the network can identify training samples more accurately, the recognition effect on the new dataset is still not ideal, making the research hardly applicable;

(2) Video-based human behavior is continuous processes. The same action clips may exist in different videos, resulting in classification confusion. For instance, in the two actions of sitting down and standing up, both having a sitting state, which requires considering the state before and after the sitting for classifying the actions. However, the video clips with a few frames are only available as the input of the spatial-temporal convolutional neural network (CNN), which means unable to capture the connection between adjacent video frames, so that cannot reach a superior recognition effect;

(3) For the sakes of attaining better training effects and avoiding overfitting, DNN tends to have tremendous parameters and the model requires ample samples. Therefore, if the training data for experiments is limited, optimizing the network structure would be a crucial work.

The human motion in the video contains two dimensions of features, one is the spatial feature, and the other is the time-sequential feature, and the recognition effect is depending on the validity and utilization of the action information extracted from the video. With regard to the video-based HAR research, the improved Long-term recurrent convolutional network (LRCN) algorithm is proposed, which extracts features of human action with ResNet-34 network instead of AlexNet in LRCN from the space and time, and experiments verify proposed approach is capable of capturing more abundant behavioral features and attaining higher classification accuracy. In addition, a novel 3D CNN with eight convolutional layers and five pooling layers is presented. The network is used to extract the spatial features of action, and then obtain the classification result through two fully connected layers and a softmax layer. Moreover, both proposed HAR algorithms are verified on the UCF101 human action dataset and outperform the preceding methods.

The rest of this paper is organized as follows. Section 2 publishes a review of related work. An overview of HAR tasks is introduced in Section 3. Section 4 and 5 describe the 2D and 3D methods, respectively. The experimental results and the performance analysis are presented in section 6, followed by the conclusion in section 7.

2. Related works

Video-based HAR research can realize practical application and considerable economic benefits in numerous fields. Therefore, scores of researchers have attached great importance on this topic in recent years. Currently, substantial organizations have made a great contribution to this research. For example, the Visual Surveillance and Monitoring project in the United States is capable of automatically monitoring the military by reviewing and analyzing video on the battlefield and public.[3]

In Ref.4, a dual-stream CNN structure with spatial-temporal networks for video motion recognition is proposed. Despite the limited training data, it proved that the CNN training multi-frame dense optical flow could effectively obtain human motion information. However, this method still has few shortcomings: First, the experiment needs more training samples, which is obviously tough; second, the current architecture lacks the employment of the shallow feature representation approaches in state-of-the-art, such as the local feature pooling in the spatial-temporal dimension; third, the display processing of camera motion should be refined by superseding the average displacement subtraction method. In Ref.5, Krizhevsky trained a large-scale deep CNN and used it to classify 1.2 million high-resolution sample images. In the test data, the classification error rates overwhelmingly outperform state-of-the-art methods. However, that research still needs enhancement. For example, exerting the unsupervised pre-training to simplify the experiment, which will enlarge the scale of the network without increasing the amount of label data if the computing power satisfies certain conditions.

To take advantage of temporal-spatial features of the image sequence data, two action recognition algorithms are presented in Ref.6. Firstly, a Fourier Time Pyramid model based on residual features. Secondly, the feature recalibration strategy is applied to the LRCN for features extraction. However, some points should be considered. For example, adding the optical flow feature of the individual motion based on the RGB features to improve the recognition effect. Additionally, making efforts on the algorithm stability aiming at the complexity of the background of the individuals in the video and the variability of their postures.

In Ref.7, Dong replaced the large convolutional kernels in the original 3D CNN model with multiple stacked small convolutional kernels, and expanded the feature map into feature vectors for optimizing the network structure of deep learning. Nevertheless, this study has a lot of improvements can be attempted in the recognition accuracy like extending the fashionable network structure to 3D CNN, or using the methods applied in target detection to segment characters from complex backgrounds ahead of recognize. To address the obstacles of the lengthy training process and redundant feature in the LRCN algorithm, Zhang[3] made the improvement through adopting threshold cycle unit instead of the long-short

term memory units in the LRCN. The experimental results show that the adjustment is capable of boosting the recognition accuracy and shortening the needed time. Nonetheless, it is still questionable whether the algorithm is practicable in the real surveillance video. In Ref.8, Ji et al. constructed a novel 3D CNN action recognition model, which extracts features from spatial-temporal dimensions and captures motions encoded in multiple near frames utilizing three-dimensional convolution. Despite the proposed model outperforms the basic line method, it still needs ampler label samples, unless the model has been pre-trained using the unsupervised algorithm in advance.

In Ref.9, a model named TC3D based on motion course is proposed. The results manifest that TC3D with the fusion of time series features and 3D CNN features can enhance the recognition accuracy and has superb robustness. However, the flaw with this study is that did not extract the motion track of all 3D convolutional layers, leading to the lower recognition rate. Yang et al.[10] stated that comparing with two-dimensional convolution, three-dimensional convolution involves more parameters, which result in computational complexity and enormous memory consumption. Consequently, a viable asymmetric unidirectional three-dimensional convolution approximating the traditional three-dimensional convolution is proposed. For magnifying the feature learning capability of asymmetric three-dimensional convolution, the local three-dimensional convolutional networks with multi-scale three-dimensional convolution branches are proposed simultaneously. Based on both aspects, an asymmetric deep 3D CNN model is constructed. Experiments show that the asymmetric model is superior to the existing 3D CNN model in both effectiveness and efficiency. In the following research, the author can take an asymmetric residual 3D convolution network into account for increasing the model's depth, thereby further improving the motion recognition performance.

In order to raise the accuracy of behavior recognition, analysis of deep learning autonomously extracting image features and deep maps in RGB-D information can provide 3D scene structure feature information, Liu et al.[37] came up with a novel algorithm based on RGB-D behavior recognition and deep learning. Specifically, the deep information is combined with the random forest to generate skeleton data, and the spatial information of the skeleton image is added to the color image to generate a picture with ROI. The experimental results indicate that the algorithm effectively uplifts the accuracy of behavior recognition. However, if the author considers the richer information in the depth picture and adds time information to the merged picture, the accuracy will be further improved. For the sake of enhancing the generalization ability of DNN for skeleton-based HAR tasks, data augmentation is a widely used technique, whereas conventional methods often augment data to generate a new sample by hand transformation, due to the lack of learning parameters, these methods usually cannot be trained and just discarded after the test. In Ref.38, a novel data-augment network SFN that generates new samples over LSTM automatic encoder network is proposed. Hence, SFN and human behavior recognition networks can be cascaded together to form a combined network that trains in an end-to-end approach. Nonetheless, the authors can consider the other network architectures (such as DenseNet architecture) incorporated into SFN for improving the model validity, rippled SFN method outward other modalities, such as RGB and depth, and further enhance its performance by using the automatic encoder deeper network.

3. Human action recognition overview

3.1. Research framework

The essence of HAR is to obtain the temporal and spatial feature information of the individual motion by learning video or image data, subsequently distinguish the action's category. Accordingly, it can be simply regarded as a combination of two steps of feature extraction and classifier recognition, the typical identification process is shown in Fig.1.

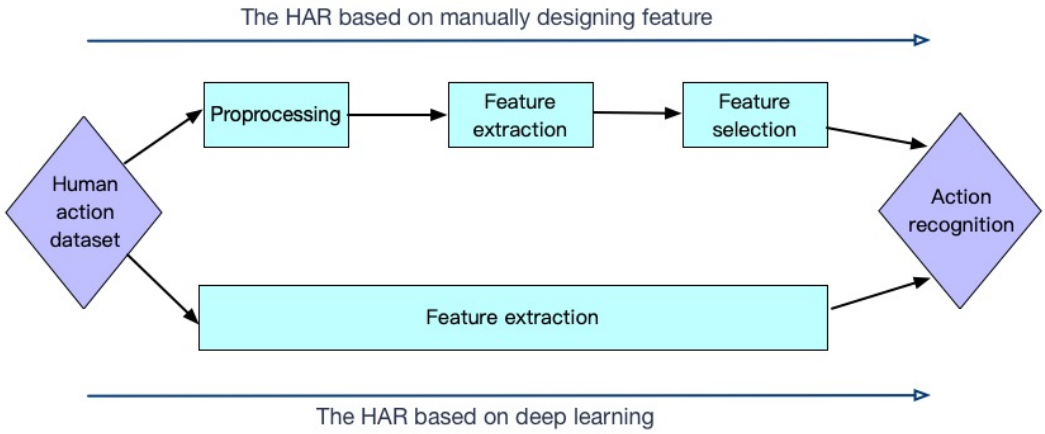

Fig.1. The basic framework of HAR [6]

According to the above figure, the HAR process first extracts deep features that can represent human behavior. This process is analogous to the attention mechanism by assigning a probability of 0 to 1 between different types of data to represent the importance of information, focusing on the necessary information.[2] The following step is to pass the extracted features to the classifier for recognition and classification by preprocessing and specifying vector representation. The first step in the entire process is crucial and also takes the longest time in all work, the accuracy of feature extraction often directly affects the performance of the algorithm and the accuracy of the classification. Hence effective methods must be used for improving the ability of features extraction and express.

3.2. Feature extraction

Existing handcrafted feature extraction includes local feature extraction and global feature extraction. The global feature is obtained by segmenting characters and backgrounds from the video data, subsequently sent it to the classifier for classification. Additionally, the local feature is obtained by detecting spatial-temporal interest points in the video, alternatively sampling the interested region and then convey to the classifier.[9] The existing feature extraction method has certain drawbacks. It is readily susceptible to noise, occlusion and light conditions, and also has poor stability.

Recently, with the gradual development and advancement of technology, deep learning algorithms have been employed in the feature extraction and recognition of human motion. At present, besides the common CNN and recurrent neural network (RNN), there are deep network models such as Restricted Boltzmann Machine (RBM) and Automatic-Encoder (AE), which can also be used to address the relevant issues related HAR.

Since CNN has certain advantages in spatial feature extraction, it is growing applied to video-based feature extraction. For instance, for each frame image of video, Ji et al. firstly extracted feature from the five channels of the gradation, the gradient in the x-direction, the gradient in the y-direction, the optical flow in the x-direction, and the optical flow in the y-direction as the input of the model, then implemented the operations of 3D convolution and pooling for each channel, and finally merged the information obtained so that generated the final feature description, which also contains spatial and timing information.[8] Karpathy et al. combined the information contained in all frames in a video by the correlation of adjacent image frames, so that the character action is capable of expressing more effectively.[22] Simonyan simultaneously added spatial and temporal flows to CNN.[4] The former is used to obtain static spatial and morphological features, and the other is exerted to obtain temporal and motion features between successive frames. After the above two features pass the softmax layer, acquiring the eventual recognition result by the average weight or SVM algorithm.[23-24] In Ref.25, Taylor et al. harnessed the restricted Boltzmann algorithm to capture the implicit feature representations in adjacent frames by convolving, normalizing, and pooling the image frames.

3.3. Classification method

The classification techniques for action recognition includes the classification method based on the generation model and the method based on the discrimination model.[26]

Commonly used generation model classifiers are: Hidden Markov Model (HMM), Dynamic Bayesian Network (DBN), Latent Semantic Analysis (LSA) and so forth; regular discrimination model classifiers involve: conditional random field (CRF), k nearest neighbor (KNN), Adaboost, support vector machine (SVM), decision tree (DT), dynamic time warping (DTW), random forest (RF), and softmax classifier.

Different from Logistic regression (namely binary logistic regression), the softmax classifier extends the logistic regression into the multiple logistic regression. The main distinction between the two is that the latter solves the multi-classification problem, while the former aims at the two-class problem. The softmax classifier uses the absolute value of the calculation result to indicate the possibility of action belongs to a certain category, and the larger the value, denoting the greater the probability. In addition, experiments show that softmax classifiers are prone to perceive changes in accuracy than SVM classifiers. For the calculated probability values for each category, it always makes it closer to the standard results and attains smaller loss function value.[26] Accordingly, the following experiment in this article will use the softmax classifier to tackle the problems related to HAR.

4. 2D methods

4.1. Long-term recurrent convolutional network

Excepting the requisite of the spatial information in the video, video-based HAR should also adequately utilize the timing relationship between video frames. Combining the CNN and the RNN constitutes a LRCN algorithm. The LRCN model proposed by Donahue et al. consisting of two parts[28-29]. The first part uses the AlexNet network presented by Krizhevsky et al. for extracting intra-frame space features.[5] The second part uses the LSTM invented by Hochreiter et al. for modeling the time series relationship.[30] It first uses the AlexNet network to extract the spatial features of each frame of the video, subsequently sends it to the LSTM network to obtain the timing information. Finally, the output of each LSTM unit is normalized by the softmax function and then averaged. The calculated value is used to decide the category of the entire video.

The basic idea of LRCN is to input the static feature of human motion captured by AlexNet into the LSTM network to obtain time information of consecutive frames. Specifically, all the frames initially intercepted by each video are extracted by the AlexNet network, and the output of the fc6 layer in the network is used as the feature vector, and denoting the feature vector with the dimension of 4096×1 obtained from the number t frame as f_t, later send it to the LSTM as the input of the softmax classification layer, and the classification result of the frame after normalization is expressed by the following Eq. (1) [31]:

$$P(y_t = c) = softmax(\hat{y}_t) = \frac{\exp(\hat{y}_t, c)}{\sum_{c' \in C} \exp(\hat{y}_t, c')}, c \in C \quad (1)$$

Where, *C* represents all the classification categories. Finally, the average value fusion method is used to obtain the average result of the entire video prediction result, and which category corresponds to the largest prediction value, that is the classification category result of the video.

4.2. The improved LRCN

4.2.1 The framework of improved LRCN

Due to the slow convergence rate and limited recognition effect of the existing LRCN network on the human behavior dataset, the deeper ResNet-34 network is applied to extract the spatial feature of the video in this paper, replacing the original AlexNet network.[32] The experimental results show that the improved LRCN network can enhance recognition accuracy. The structure of the network is exhibited in Fig. 2.

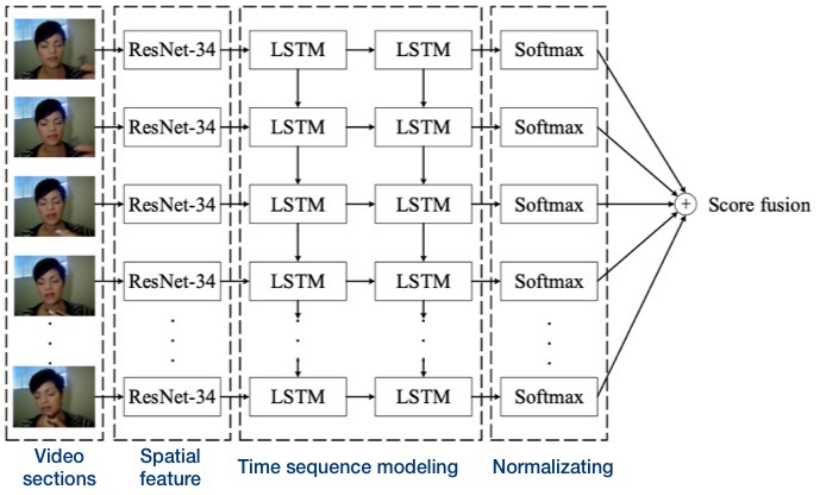

Fig.2. The algorithm framework of improved LRCN

After extracting the spatial features of each video frame with ResNet-34, the output feature vector with 512×7×7 from the fifth convolutional layer is employed as a spatial feature description of the individual action. Afterward, the above feature is globally averaged for pooling as a new feature vector. Therein, the output vector of t-frame of the image with the dimension of 512 × 1 is denoted as f_t, and then feed it into LSTM and receive the $\hat{y}_t$ value, which is trained as the input of the softmax layer, obtained the prediction results of t-frame picture after normalization. Finally, the prediction result of each frame image is averaged, and the prediction value corresponding to which category of action is the largest, and the corresponding C value is used as the prediction result of the entire video.

4.2.2 Feature extraction with ResNet-34

In Ref.32, He et al. designed the basic architecture of the ResNet network, which is made of 34 network layers, consisting of 33 convolutional layers and 1 fully connected layer. There is no positive correlation between the recognition effect of the algorithm and the number of network layers. Namely, the deeper network does not equal better classification effect of the model. Conversely, if the number of network layers is exceeding, the accuracy would probably drop. This phenomenon means that the accuracy initially increases as deepening network layers, and then reach the highest point, but when the network continues to deepen, the accuracy will begin to decrease, this happens on both training and testing.

As long as the deep network is connected to the identity mapping and reduced to a shallow network, the accuracy drop problem can be solved. Given that the network structure is designed to H (x) = F(x) + x, converting it into the learning problems of residual function F(x) = H(x) - x. It is able to construct the identical mapping with H(x) = x and easily calculate fitting residuals in the case of F(x) = 0.

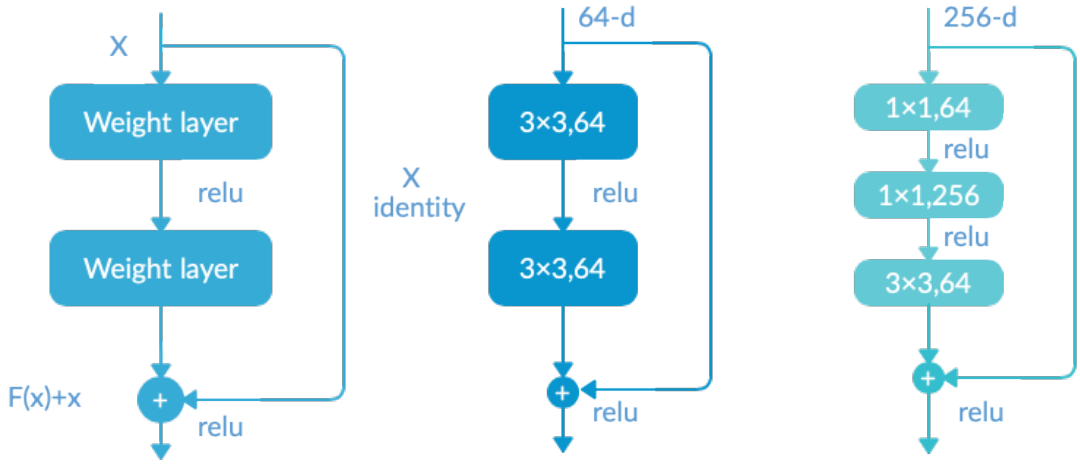

Fig.3. (a) Fig.3. (b) Fig.3. (c)

Fig.3. Residual block and residual function F

Based on this principle, He et al. proposed the ResNet network by calculating the residual value F(x) between two adjacent layers and outputting the value of F(x)+x so that addressing the degradation problem. On the basis of the existing DNN, the ResNet adds an identical mapping. Table 3(a) shows the structure of the residual block of ResNet, and its expression is as follows[32]:

$$y = F(x, \{W_i\}) + x \tag{2}$$

For each residual block, X denotes the input vector, Y denotes the output vector, the function $(x, \{W_i\})$ denotes the residual mapping constructing with the multi convolutional layers. In Fig.3. (a), the linear rectifying function $F = W_2\sigma(W_1 x)$ is expressed by σ.

If the dimension of F is consistent with x, the shortcut can sequentially accumulate the elements in the feature map of each channel, preparing for the F+x operation. If the dimension of F is inconsistent with x, the Eq. (3) is employed to tackle the dimension of x.

$$y = F(x, \{W_i\}) + W_s x \tag{3}$$

The linear mapping is denoted by the matrix W_s, and projecting x by applying a 1×1 convolutional kernel so that W_s has the same dimension with $F(x, \{W_i\})$. Thereafter Relu the output vector again after adding.

If the output of each residual block has the same dimensions as the input, an identical mapping of the respective values would be executed by the Eq. (2). There are two ways to cope the situation of increasing dimensions: First, use the shortcut to zero-fill the input vector, so that the dimension of x is increased and no redundant parameters are added; second, according to Eq. (3), the shortcut projection principle applied in correlating the dimension. When the size of the two feature maps is different, the shortcut merge has a step size of 2.

Furthermore, the number of convolutional layers in the residual function is unfixed. Fig.3. (b), Fig.3. (c) show the structure of the residual block 2nd and 3rd convolutional layer, respectively. The residual block in the ResNet-34 network consists of two 3 × 3 convolutional kernels, as shown in Fig.3. (b). For ResNet-50 and other deeper networks, the residual block consists of three convolutional kernels with sizes of 1×1, 3×3, and 1×1, as shown in Fig.3. (c). In order to reduce the time complexity of the algorithm and keep the spatial complexity stability, first use the first and third 1×1 convolutional kernel to reduce and then increase the dimension. Subsequently, the second 3×3 convolutional kernel gets the low-dimension input and output. And adjacent layers' convolutional kernels all introduce ReLU nonlinear mapping.

As the following table 1 shown, Conv1 signifies the first convolutional layer, each convolutional layer of later five are indicated as Conv 2_X, Conv3_X, Conv4_ X and Conv5_X, the correspondingly residual block number is 3,4,6 and 3, and each residual block contains two convolutional layers. The two-dimensions maximum pooling process of Max Pool is executed after the first convolutional layer, in the fifth convolutional layer, performed two-dimensions pooling average processing of Ave pool, and the last layer *fc* is a fully connected layer, which is connected with a softmax layer. The *table 1.* below specifically shows the structure and parameter of each network layer.

Table 1. ResNet-34 model structure

Network layer	Input dimension	Output dimension	Kernels
Conv1	224×224	112×112	7×7，64，stride2
	112×112	56×56	3×3max pool，stride2

Conv2_x	56×56	56×56	$\begin{bmatrix} 3\times3 & 64 \\ 3\times3 & 64 \end{bmatrix}\times3$
Conv3_x	56×56	28×28	$\begin{bmatrix} 3\times3 & 128 \\ 3\times3 & 128 \end{bmatrix}\times4$
Conv4_x	28×28	14×14	$\begin{bmatrix} 3\times3 & 256 \\ 3\times3 & 256 \end{bmatrix}\times6$
Conv5_x	14×14 7×7	7×7 1×1	$\begin{bmatrix} 3\times3 & 512 \\ 3\times3 & 512 \end{bmatrix}\times3$
			Ave Pool，100-d fc，softmax

5. 3D methods

5.1. Comparison between 2D method and 3D method

The 2D DNN is a traditional method for solving the HAR tasks, namely, extracting the spatial-temporal features of human motion in the video sequence, subsequently sending it into the classifier to acquire the final classification result. Accordingly, the recognition effect depends on the process of feature extraction. However, in real life, the motion feature of the target human is rarely acquired in advance, and the behavior recognition algorithm is difficult to capture the most important features through detection. In addition, unlike the target detection, the clothing, posture and figure of the character all also have an impact on the recognition effect. Consequently, the recognition effect of this method is often limited.

CNN is widely used in the fields of target detection, pattern recognition, and computer vision. In terms of image target recognition, CNN has two unique advantages. Firstly, through the processing of the CNN, the external interference such as lighting conditions and complex background environment can be neglected, making the algorithm more consistent. Secondly, when processing image data, it is unnecessary to input the extracted features, but directly input the image matrix, which greatly reduces the computational and algorithmic complexity, and also strengthens practicability.[35]

5.2. 3D Convolutional Neural Network

5.2.1. 3D convolution

When using CNN to process the video sequences set, it usually directly processes the input of each captured video frame image, which only performs 2D convolution operations on multiple images without reflecting the temporal feature between successive video frame sequences.[36] Therefore, this paper considers the time dimension in CNN processing and expands the 2D convolution to 3D convolution to obtain both spatial and temporal information.

The input of the 3D CNN is composed of stacking consecutive frames, which is capable of extracting features in three dimensions at the same time. The feature cube can be connected with consecutive video frames of the previous layer through the three-dimension convolutional kernel, thereby realizing the feature extraction task of the continuous multi-frame, and a period individual motion information can be acquired for. This process can be expressed in Eq.4:

$$v_{ij}^{xyz} = \tanh\left(b_{ij} + \sum_m \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)}\right) \quad (4)$$

The output of pixel (*x, y, z*) of the *j*-feature of the *i*-layer is represented by the v_{ij}^{xyz}, and b_{ij} denotes the shared bias of the *j*-feature of the *i*-layer, and the number of feature maps of the *i*-layer is *m*. It is shown that P_i and Q_i signify the spatial dimension of the *i*-layer's 3D convolutional kernel, and R_i

represents the time dimension of the *i*-layer's 3D convolutional kernel. w_{ijm}^{pqr} is the convolutional kernel weight of the previous layer connecting to the *m*-feature. Compared to 2D convolution, 3D convolution adds the time dimension. Fig. 4 and 5 are schematic diagrams of 2D convolution and 3D convolution.[8]

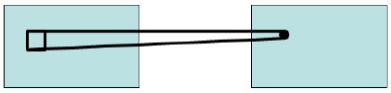

Fig.4. 2D convolution

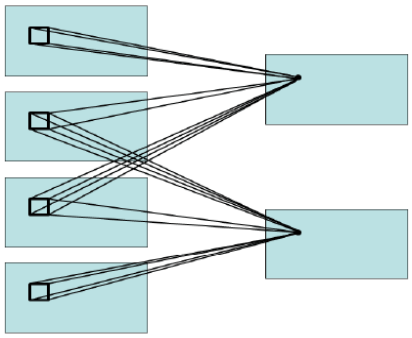

Fig.5. 3D convolution

5.2.2. Improved 3D CNN architecture

The 3D CNN model constructed by Ji et al. consists of a hardwired layer, three convolutional layers, two sampling layers, and a fully connected layer, due to the limited convolution layers and kernels, so that the network can only extract meager features.[8] Based on the above 3D CNN model, a novel network structure is proposed, which not only increases the convolutional layers but also the convolutional kernels, so that the subsequent convolutional layer can combine the features extracted from the previous convolutional layer and receive more abstract features.

The proposed 3D CNN includes eight convolutional layers, five pooling layers, two fully connected layers, and one softmax classification layer. According to the research of 2D CNN, the small receptive field of 3×3 convolutional kernel is capable of producing the best results in a deeper architecture. Therefore, in the architecture design, the size of the convolutional kernel is determined to be 3×3×3 in this paper. Where 3×3 represents the spatial feature dimension and 3 signifies the temporal feature dimension. The number of convolutional kernels in each convolutional layer is 64, 128, 128, 256, 512, and 512, and the stride of each movement of the convolutional kernel is 1. Besides, reducing the size of the feature map and the computational scale through max pooling. Excepting the Pool1 using a 1×2×2 window to downsample the spatial dimension, and the remaining layers use a 2×2×2 window to simultaneously downsample the time dimension and the spatial dimension. The number of neurons in both fully connected layers is 4096. After the fully connected layer, classifying the feature vectors and recognizing the action. The structure of 3D CNN is shown in *Table 2*.

Table 2. 3D CNN structure

layers	Kernels	kernel_size	subsample	pool_size	strides	output_dim
Conv1 (Conv3D) Pool1(MaxPool3D)	64	3×3×3	(1,1,1)	(1,2,2)	(1,2,2)	/
Conv2 (Conv3D) Pool2(MaxPool3D)	128	3×3×3	(1,1,1)	(2,2,2)	(2,2,2)	/

Conv3a (Conv3D)						
Conv3b (Conv3D)	256	3×3×3	(1,1,1)	(2,2,2)	(2,2,2)	/
Pool3(MaxPool3D)						
Conv4a (Conv3D)						
Conv4b (Conv3D)	512	3×3×3	(1,1,1)	(2,2,2)	(2,2,2)	/
Pool4(MaxPool3D)						
Conv5a (Conv3D)						
Conv5b (Conv3D)	512	3×3×3	(1,1,1)	(2,2,2)	(2,2,2)	/
Pool5(MaxPool3D)						
Fc6 (Dense)	/	/	/	/	/	4096
Fc7 (Dense)	/	/	/	/	/	4096

6. Experiments and analysis

6.1. Dataset

For HAR tasks, HMDB51 and UCF101 are typical and popular datasets[33-34]. Among the UCF101 is one of the datasets containing the most action categories and the largest number of video clips and the following experiment is based on it. The data source is https://www.crcv.ucf.edu/data/UCF101.php .

In Ref.34, Soomro et al. captured video clips of 13320 personal body behavior from YouTube, which cover 101 actions and 27 hours long. Each action has 25 different objects and these object actions are collected in 4-7 video clips. The 101 class are divided into five broad categories: human and object activities, human body movements, activities between different individuals, playing instruments, and sports. The specific video information of the UCF101 dataset is displayed in *Table 3*.

Table 3. UCF101 Dataset

Dataset	Categories	Training video clips	Testing video clips	Total video clips	Video length
UCF101	101	9537	3783	13320	5-10 秒

The video data of UCF101 dataset is collected in real-life scenarios with the different executor for each action. Meanwhile, the HAR algorithm effect can be persuasively verified due to the complexity of the motion background. The reason why the action categories with the same border color are put together is that they have similar background information, such as applying eye shadow and lip sticking, all in the indoor background. The UCF101 in each category averagely contains 130 video clips with a slightly quantitative difference between each action. Each video has complete human action progress with excellent quality.

6.2. Data Preprocessing

In the experiment, the data is divided by the division criteria of split1 in UCF101. The dataset is processed through the following steps:

(1) Intercepting video into a frame

Since the video sequence of the original data cannot be directly used as the network input, firstly converting the video data into image data for storage. In this way, not only can the data conform to the network requiring format, but also avoid the process of decoding the video into an image, so that network can directly read the continuous image sequence, which greatly improves the training speed. Therefore, prior to process the subsequent data, the original video sequence is parsed into image sequences one by one, and named according to the original order of the data, afterward the parsed image is stored with the mode of JPG image encoding.

(2) Coding action category

After getting the name of each action, read it into the list and sort all the category names by the first letter. For quantizing the character data, it is necessary to encode the category name with one-hot for obtaining a quantized list.

(3) Unifying video frame length

Posterior to convert the video data into image data, the four columns of training/test set, action category, video name, and video frame are written into a new file. Since each video length is inconsistent, in order to avoid a large gap in the number of video frames, the length of each video sequence is controlled to be in the range between 40 and 300 frames. *Table 4* below shows the first five lines of data for the file.

Table 4. data_file.csv file example

Train/Test set	Motion category	Video name	Frame
train	ApplyEyeMakeup	v_ApplyEyeMakeup_g08_c01	121
train	ApplyEyeMakeup	v_ApplyEyeMakeup_g08_c02	118
train	ApplyEyeMakeup	v_ApplyEyeMakeup_g08_c03	147
train	ApplyEyeMakeup	v_ApplyEyeMakeup_g08_c04	225
train	ApplyEyeMakeup	v_ApplyEyeMakeup_g08_c05	277

(4) Selecting randomly data

The implementation of the random selection of data is based on the random function of Python that disrupts the data sequence. For an array of file names and storage paths containing all images, select 16 video sequences at a time and get their category names, then scramble the order of the video sequences in the training and testing set, respectively. Last, within each video sequence, the decoded video frame image obtained is arbitrarily selected as an input.

(5) Data augmentation

In the UCF101 dataset, the average length of each video segment after decoding is about 200 frames, so that a complete video sequence extracted be unable to be sent to the network for training. Thereby, firstly selecting a video sequence and obtaining the number of images it contains, namely, the length of the video frame, and then transfers the original sequence into a shorter sequence in steps of 40 frames, which is conducive to generate different training samples and achieve data augmentation.

(6) Data inputting

When dealing with minor datasets, numpy arrays are available to store and input the data. However, for a vast dataset such as UCF101, it is impossible to read all the data directly into the memory. Accordingly, reading directly maximum data into memory for improving the training speed, and the rest part is generated through the generator.

6.3. 2D experiment and analysis

6.3.1. Experimental procedure

The process of algorithm implementation includes training and testing. Firstly, the training sample of the known category is trained by the model; subsequently, the trained model is used to verify on the testing sample; and then output the action category of each video sequence most likely to belong to. The training and testing process of the entire model is illustrated in Fig. 6.

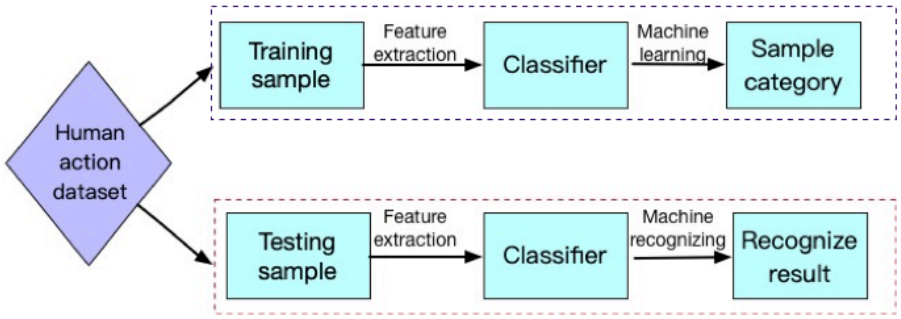

Fig.6. A flowchart of model training and testing

When setting up the network structure, set the activation function to Relu and increase the number of convolutional kernels in layers. The added Dropout with a ratio of 0.5 at the last layer of the AlexNet network is designed to prevent from overfitting. Applying Adam optimizer in the network training phase, Adam is different with the stochastic gradient descent (SGD), which uses the constant learning rate for updating weight values during training, and it updates the weight by matching adaptive learning rates to the different parameters. The configuration of the learning rate can affect the convergence speed of the model and the degree of overfitting. If the learning rate is small, the network will reach convergence for a long time, conversely, the network may not converge. In this paper, the initial learning rate is set to 1e-5, and the learning rate decays value to 1e-6, so that the value of the learning rate is automatically reduced by 1e-6 for each epoch. The model trains 500 epochs, each epoch iterates 100 times, namely, the number of iterations of the entire training process is 50,000 times. In the experiments, the batch_size is set to 16. The parameters involved in the model training and testing process are shown in *Table 5*.

Table 5. Parameter configuration

Name	batch_size	learning_rate	decay	nb_epoch	drop_out	input_shape
Parameter	16	1e-5	1e-6	500	0.5	(80,80,3)

6.3.2. Experimental and analysis

Fig.7. exhibits the accuracy curve of the training and testing when using the LRCN model for UCF101 dataset. Fig.8. displays the loss curve of the training and testing. The model performs 500 epoch iterations. Apparently, after about 200 epoch iterations, the accuracy of the training climbs to the peak and the loss value reaches a minimum, after about 225 epoch iterations, the accuracy and loss value of the testing begin to stabilize. However, there is still a certain degree of oscillation, and the oscillation amplitude on the testing is sizable. Additionally, although the training data can obtain a higher classification accuracy, the verification effect of the model on the testing data is suspicious. Finally, although the accuracy of the training set can reach in shortly 99.5%, the recognition rate on the testing set barely reached 54.3%. The poor recognition effect of the algorithm on testing set may be caused by the network structure of LRCN model. Since the number of LSTM network layers is restricted, plus the number of neurons in the hidden layer is only 256. Meanwhile, the extracted spatial feature is insufficient because of the shallow layers of the AlexNet network, so that the information of individual action is not effectively learned.

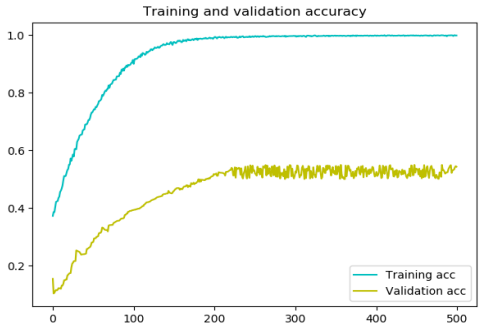

Fig.7. The variations of training and validation accuracy of LRCN

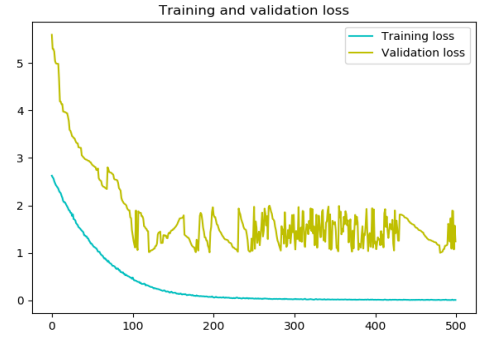

Fig.8. The variations of training and validation loss of LRCN

The UCF101 dataset was identified through the improved LRCN model in the same parameter and experimental configuration. Fig.9. displays the variation of the accuracy of the training and testing obtained by extracting spatial features with the ResNet-34 network, and Fig.10 exhibits the corresponding loss function. The model has reached convergence after about 50 epochs during training and reached convergence after about 150 epochs during testing. Compared with the existing LRCN model, the improved LRCN model network converges faster, and the recognition performance is greatly improved during testing. Although the accuracy and loss function of the network still oscillates after convergence, the amplitude of the oscillation is significantly smaller than the preceding LRCN model. Consequently, it is transparent that the ResNet-34 network can be applied in extracting spatial features for rich motion information so that attain more accurate and stable recognition performance. In the end, the classification accuracy of the model reached 99.7% during training, and the classification accuracy rate on the testing set reached 62.1%, which was 7.8% higher than the previous LRCN model. The experiment verified the feasibility and validity of the ResNet-34 network for spatial feature extraction.

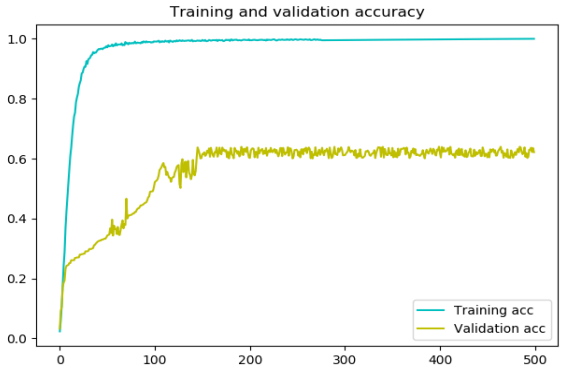

Fig.9. The variations of training and validation accuracy of the improved LRCN

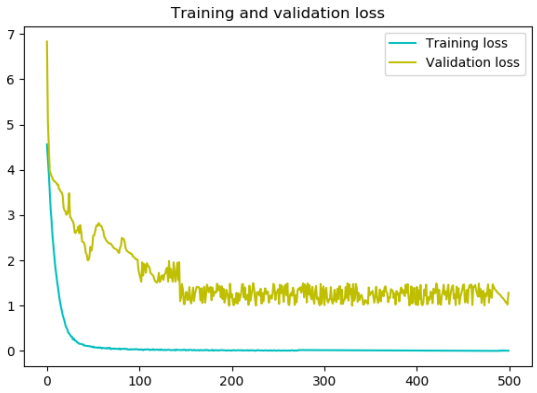

Fig.10. The variations of training and validation loss of the improved LRCN

Table 6 shows the duration of the UCF101 dataset classification, the number of training parameters, and the final classification accuracy information of training and testing after the 500 epochs training using the LRCN model and the proposed model, respectively. Latter model parameters required more time in training because of the increased amount of computation resulting from the deeper network of ResNet-34, but ultimately the higher recognition accuracy is achieved during both training and testing, which proves that proposed method is profound for the HAR research.

Table 6. Comparison with LRCN

Model	Epoch	Time(h)	Training parameters	Training accuracy (%)	Testing accuracy (%)
LRCN	500	108	5,529,477	99.5	54.3
This paper	500	120	8,714,359	99.7	62.1

6.4. 3D experiment and analysis

6.4.1. Experimental procedure

Based on 3D convolutional neural network, a novel algorithm is proposed, which directly uses the image as input without preprocessing, feature extraction, and data reconstruction after recognizing. The algorithm flow chart is illustrated in Fig.11.

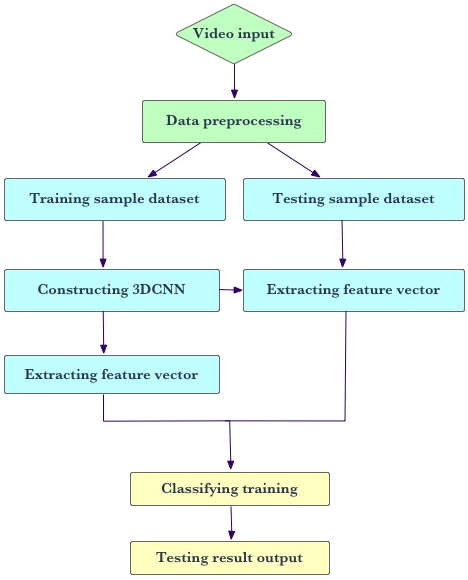

Fig.11. Algorithm flow chart [27]

(1) Video input
The video is intercepted into image data.
(2) Data preprocessing
It includes the steps of dividing the training and testing set, unifying video frame length, encoding behavior category name, and data augmentation according to the split1 standard.
(3) Constructing the 3D convolutional neural network
Building each convolutional and pooling layer in turn, and setting the relevant parameters based on the Keras deep learning framework.
(4) Extracting feature vectors
Each feature map obtained in the second sampling layer is drawn into a vector, then reduce the dimension with a common neural network, obtaining the feature vector of the image through the second fully connected layer.
(5) Classification training
Connecting the output layer with softmax classifier, and use it to classify the extracted feature vectors.
(6) Output prediction results
Using the trained model to test and output the predicted result.

6.4.2. Experimental and analysis

Fig.12. shows the curves of accuracy variations during training and testing using 3DCNN on UCF101 dataset and Fig.13. displays the loss function curve. Transparently, the accuracy and loss values reach maximum and minimum after approximately 50 iterations during training, respectively, and the testing reaches also after 180 iterations. Eventually, the designed HAR algorithm based on 3D CNN achieved 99.7% classification accuracy for the training and reach 85.9% accuracy during testing, which has a significant improvement in the recognition rate compared to the LRCN model and the improved LRCN model.

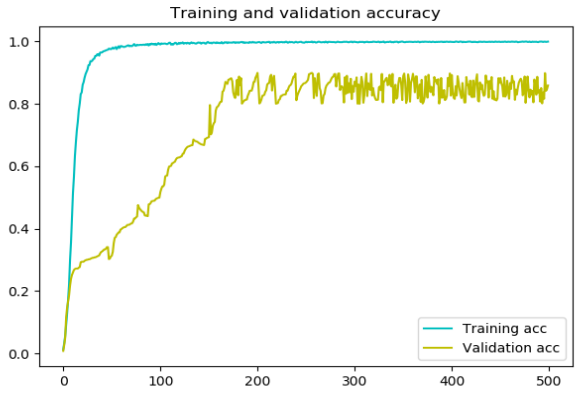

Fig.12. The variations of training and validation accuracy

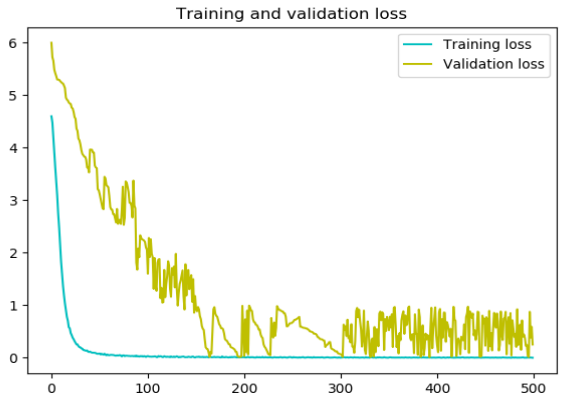

Fig.13. The variations of training and validation loss

Table 7 shows the experimental comparison between Ji et al. and presented method. Since the designed 3D CNN network embraces increased layers and greater parameters, eventually it takes more time to train. However, due to the increased convolution layers and convolutional kernels, the subsequent convolutional layer can combine the features extracted by the previous convolutional layer and obtain more abstract features, so that the model can exhibit stable recognition performance during training and testing. This experiment verifies that the improvement of network structure is significant for HAR research.

Table 7. Comparison between Ji's method and this paper

Model	Epoch	Time(h)	Training parameters	Training accuracy (%)	Testing accuracy (%)
Ji's method	500	144	34,120,165	90.5	74.3
This paper	500	168	82,603,877	99.7	85.9

In addition, comparing to the existing LRCN model, proposed 3D CNN model can achieve stabilization in fewer epoch, meanwhile the oscillation amplitude also decreased significantly, more importantly, because of the advantages of 3D convolution in feature extraction and the increasing number of layers, the classification effect of the 3D CNN model is significantly improved with more stable performance. Compared with the improved LRCN model, although the oscillation amplitude of the model after convergence is higher, the 3D CNN has outstanding recognition effect. The feature of the video is extracted by the 3D convolution, meaning the algorithm can avoid the problems of complicated

feature extraction and the sluggish training speed. Namely, the algorithm has the advantages that the extracted features have strong representation and fast extraction speed.

7. Conclusion

In this paper, a novel 2D method is proposed for human action recognition that is based on LRCN with better expression ability on spatial features extraction, which can be more effectively applied in human action recognition. Additionally, an inventive 3D CNN architecture is also proposed. It has the benefits of faster convergence and remarkable recognition effect. Due to a relatively deep architecture. The proposed algorithm is time-consuming but efficiently applied to new dataset even real life. Evaluating the performance on one popular UCF101 HAR dataset, obtained results demonstrate that the proposed 2D and 3D models significantly outperform baseline approaches and previous models in both cases.

In future work, other feasible approaches such as manually extracting feature and the attention mechanism can be considered to improve the performance excepting the deep learning methods. Moreover, taking the various factors in the real scene into account is also a feasible way. Besides, the training speed can be still improved because of the tremendous model parameters and dataset. In the subsequent experiments, a more adaptive parameter adjustment method should be proposed to compare the recognition effects of the model with different parameter values.

Acknowledgments

This research was funded by National Natural Science Foundation of China, grant number 61104166. The authors would like to thank the National Natural Science Foundation of China (61104166). We declare there are no conflict of interest regarding the publication of this paper. We have no financial and personal relationships with other people or organizations that could inappropriately influence our work.

References

1. Yu Xing. Video action recognition technology research based on deep learning [D]. University of Electronic Science and Technology, 2018.
2. Xue Luqiang. Human action recognition research based on dual-stream fusion convolutional neural network [D]. Anhui University, 2018.
3. Zhang Rui. Human action recognition based on the deep convolutional neural network [D]. Nanchang Hangkong University, 2018.
4. Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos[J].2014,1(4):568-576.
5. Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[C]// International Conference on Neural Information Processing System, Curran Associate Inc,2012:1097-1105.
6. Zhou Wen. Space-time feature extraction and recognition algorithm for human action [D]. Beijing Jiaotong University, 2018.
7. Dong Guohao. Human action recognition research based on deep learning [D]. Xi'an University of Posts and Telecommunications, 2018.
8. Ji S, Xu W, Yang M, et al. 3D Convolutional Neural Networks for Human Action Recognition[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013,35(1):221-231.
9. Ji Lu. Action recognition methods research based on the 3D convolutional neural network [D]. Xi'an University of Technology, 2018.
10. Yang H, Yuan C, Li B, et al. Asymmetric 3d convolutional neural networks for action recognition[J]. Pattern Recognition, 2019, 85: 1-12.
11. Bobick A F, Davis J W. The Recognition of Human Movement Using Temporal Templates[J]. Pattern Analysis & Machine Intelligence IEEE Transactions on, 2001,23(3):257-267.
12. Derpanis K G, Sizintsev M, Cannons K, et al. Efficient action spotting based on a spacetime oriented structure representation[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE,2010:1990-1997.
13. Weinland D, Boyer E, Ronfard R, et al. Action Recognition from Arbitrary Views using 3D Exemplars[C]// IEEE International Conference on Computer Vision,2007:1-7.
14. Laptev, Lindeberg. Space-time interest points[C]// IEEE International Conference on Computer Vision,2003:432-439.

15. Willems G, Tuytelaars T, Gool L. An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector[C]// European Conference on Computer Vision. Springer-Verlag,2008:650-663.
16. Rapantzikos K, Avrithis Y, Kollias S. Spatiotemporal saliency for event detection and representation in the 3D wavelet domain: potential in human action recognition[C]// ACM International Conference on Image and Video Retrieval. ACM,2007:294-301.
17. Wang H, Klaser A, Schmid C, et al. Action recognition by dense trajectories[C]// IEEE Conference on Computer Vision and Pattern Recognition,2011:294-301.
18. Wang H, Schmid C. Action Recognition with Improved Trajectories[C]// IEEE Conference on Computer Vision and Pattern Recognition, 2011:3169-3176.
19. Laptev I, Marszalek M, Schmid C, et al. Learning realistic human actions from movies[C]// IEEE Conference on Computer Vision and Pattern Recognition,2008:1-8.
20. Wang Qiwei. Image histogram features and its application [D]. University of Science and Technology of China, 2014.
21. Bay H, Ess A, Tuytelaars T, et al. Speeded-Up Robust Features (SURF)[J]. Computer Vision and Image Understanding,2008,110(3):346-359.
22. Karpathy A, Toderici G, Shetty S, et al. Large-Scale Video Classification with Convolutional Neural Networks[C]// Computer Vision and Pattern Recognition. IEEE,2014:1725-1732.
23. Li Hang, Statistical Learning Methods [M]. Tsinghua University Press, 2012.
24. Zhou Zhihua, Machine Learning [M]. Tsinghua University Press, 2016.
25. Taylor G W, Fergus R, Lecun Y, et al. Convolutional learning of spatio-temporal features[C]// European Conference on Computer Vision. Springer-Verlag,2010:140-153.
26. Fu Mengyu. Human action recognition based on deep learning [D]. Harbin Institute of Technology, 2017.
27. Yexu Qing. Human action recognition based on 3D convolution neural network [D]. Xi'an University of Electronic Science and Technology, 2015.
28. Donahue J, Hendricks L A, Guadarrama S, et al. Long-term Recurrent Convolutional Networks for Visual Recognition and Description[J]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, 39(4):677-691.
29. Donahue J, Hendricks L A, Rohrbach M, et al. Long -term Recurrent Convolutional Networks for Visual Recognition and Description[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(4):677-691.
30. Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural Computation, 1997,9(8):1735-1780.
31. Tian Xinghui. Video-based research on human action analysis algorithm[D]. Southeast University, 2015.
32. He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[C]// Computer Vision and Pattern Recognition. IEEE,2016:770-778.
33. Kuchne H, Jhuang H, Sticfelhagen R, et al. HMDB51: A Large Video Database for Human Motion Recognition[M]// High Performance Computing in Science and Engineering l2. Springer Berlin Heidelberg, 2013:2556-2563.
34. Soomro K, Zamir A R, Shah M. UCF101: A Dataset of 101 Human Actions Classes from Videos in The Wild[J]. Computer Science,2012.
35. Ma Lijun. Action recognition algorithm analysis with 3D convolutional neural networks [D]. China University of Geosciences (Beijing), 2018.
36. Zhang Rui, Li Qishen, Chu Jun. A human action recognition algorithm based on 3D convolutional neural networks [J]. Computer Engineering, 2019,45 (01): 259-263.
37. Liu, Zhang, Wang Chuan Xu, Li. An action recognition algorithm based on RGB-D and deep learning [J]. Computer Engineering and Design, 2019, 40 (06): 1747-1750.
38. Meng F, Liu H, Member, Liang Y, et al. Sample Fusion Network: An End-to-End Data
39. Augmentation Network for Skeleton-based Human Action Recognition[C]// Transactions on Image Processing. IEEE,2019:1057-1072.