Two-Stream Convolutional Neural Network for Video Action Recognition

Han Qiao[1], Shuang Liu[1], Qingzhen Xu[1], Shouqiang Liu[2,*], and Wanggan Yang[3]
[1]School of Computer, South China Normal University,
Guangzhou 510631, China
[e-mail: qiaohan@m.scnu.edu.cn]
[2]School of Artifificial Intelligence, Faculty of Engineering,
South China Normal University, Nanhai 528225, China
[e-mail: liusq@m.scnu.edu.cn]
[3]Nelson Mandela College of Government and Social Sciences,
Southern University and Agricultural & Mechanical College, Baton Rouge 70813, USA
[e-mail: wgyang3@gmail.com]
*Corresponding author: Shouqiang Liu

Received July 11, 2021; revised August 21, 2021; accepted Septebmer 1, 2021; published October 31, 2021

Abstract

Video action recognition is widely used in video surveillance, behavior detection, human-computer interaction, medically assisted diagnosis and motion analysis. However, video action recognition can be disturbed by many factors, such as background, illumination and so on. Two-stream convolutional neural network uses the video spatial and temporal models to train separately, and performs fusion at the output end. The multi segment Two-Stream convolutional neural network model trains temporal and spatial information from the video to extract their feature and fuse them, then determine the category of video action. Google Xception model and the transfer learning is adopted in this paper, and the Xception model which trained on ImageNet is used as the initial weight. It greatly overcomes the problem of model underfitting caused by insufficient video behavior dataset, and it can effectively reduce the influence of various factors in the video. This way also greatly improves the accuracy and reduces the training time. What's more, to make up for the shortage of dataset, the kinetics400 dataset was used for pre-training, which greatly improved the accuracy of the model. In this applied research, through continuous efforts, the expected goal is basically achieved, and according to the study and research, the design of the original dual-flow model is improved.

Keywords: video action recognition, multi segment, two-stream convolutional neural network, transfer learning, pre-training

The Project was supported by Guangzhou Science and Technology Plan Project (No.201903010103), the "13th Five-Year Plan" for the development of Philosophy and Social Sciences in Guangzhou (No.2018GZYB36), Science Foundation of Guangdong Provincial Communications Department, China (No.N2015-02-064), and the Ministry of Education's 2018 first batch of Industry-University Cooperation Collaborative Education Information Security curriculum system construction projects (201801087012).

http://doi.org/10.3837/tiis.2021.10.011 **ISSN : 1976-7277**

1. Introduction

Today, video behavior recognition technology has become very popular in computer vision research, its purpose is to want from an access to video or image sequence analysis among people in various behavior [1], be able to identify what are the people in the video at any time, doing things, this is what we call "w4 model", which includes who, when, where, and what [2].

The video behavior recognition task involves identifying different actions from 2D frame sequences video clips, which may or may not be coherent throughout the video [3]. Video behavior recognition has a wide range of applications, from helping police search for prisoners to acting as an auditor for video websites to automatically identify and filter out bad videos [4]. It can also help people to better understand image features and video features, which have great economic and social value [5].

According to the further research and optimization improvement of video behavior recognition based on Two-Stream neural network [6], the main research content is to use the Xception model as the basis of the Two-Stream network model, at the same time will be segmented with convenient access to video motion characteristics of each point in time. Finally, the spatial latitude and temporal latitude of the video are extracted for the fusion of behavior characteristics, and the behavior types are obtained through comprehensive evaluation [7]. In this application, we need to think about how to realize these four aspects as follows.

(1) How to convert the video into an appropriate type and send it to the neural network as input.

(2) How to alleviate the problem of model underfitting in a realistic environment with relatively few video datasets.

(3) How to fuse two convolutional neural networks.

(4) How to optimize and improve the Two-Stream neural network model to improve its accuracy.

2. Related work

At present, there are four types of spatio-temporal convolutional networks for computer vision behavior recognition methods at home and abroad [8].

2.1 Spatio-temporal Convolutional Neural Network Based on LSTM

Lstm-based spatio-temporal convolutional neural network is an early attempt to extend 2D network to be able to process spatio-temporal data [9]. The operation can be described as 3 steps. Every frame is processed by a 2D network, and then the last layer of the convolutional neural network extracted the feature vectors [10]. The input of LSTM adopts these characteristics from different time steps to obtain the effect on the time dimension. Finally, the results are averaged or combined linearly, and then passed to a Softmax activation function to output the predicted results [11].

2.2 C3D Convolutional Neural Network

C3D convolutional neural network is an end-to-end behavioral recognition network proposed by Facebook in 2019. It can process video directly [12]. In order to effectively combine

motion information in video analysis, 3D convolution is applied to the convolution layer of CNN, so that identifiable features in both time and space dimensions can be captured. Multiple types of features can be extracted by using multiple different convolution operations at the same position of the input. Depending on 3D convolution, different 3DCNN architectures can be used to design and analyze video data. The final feature representation is obtained by combining multi-channel information.

2.3 Two-Stream Convolutional Neural Network

The Two-Stream method uses the RGB frame image (space) of the video and the dense optical flow field (time) extracted from the RGB image. The Two models are trained separately. After the model produces output, a fusion is performed on the network. These two models represent static information and short-order information respectively. It is very effective at identifying the type of behavior in a diagram [13], This is shown in **Fig. 1**.

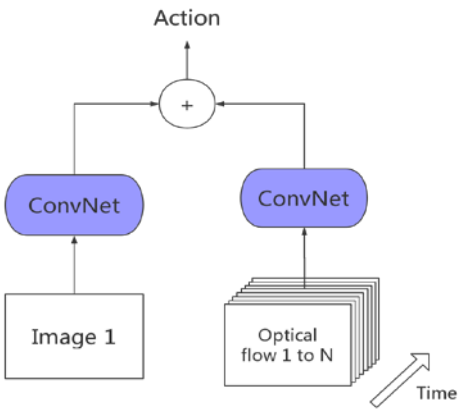

Fig. 1. Reductive model of double-flow convolutional neural network

2.4 Graph Convolutional Neural Network

Graph Convolutional Neural Network is an emerging method in recent years. Compared with traditional convolutional neural network, which can only process Euclidean spatial data, this network method based on graph structure can not only process non-Euclidean spatial data commonly used in daily life, but also enhance the diversity and scale of image recognition. At present, the main fields of Graphic Convolutional Neural Network are convolutional operator construction, complex information modeling on the graph and training process optimization. Graph convolutional neural network mainly includes pooling operator and convolutional operator construction.

3. Detailed Design and Code Implementation of Video Behavior Recognition Model

This application is developed using CoLab, Kaggle and Geek Cloud platform, using Python 3.6, using TensorFlow 2.0 Keras, and using Xception model for neural network model. This application uses UCF101 dataset and Kinetics400 dataset.

UCF-101 is a very classic dataset in video behavior recognition. Many articles on video recognition classification are based on the training in UCF101, which was published by the University of Florida in the United States in 2008. Collected from YouTube and BBC/ESPN,

among others, it contains 101 categories of daily behaviors, such as eyeliner, squatting and yo-yo, with a total of 13,320 videos.

The neural network model used in this experiment is a multi-segmental Two-Stream Convolutional neural network model [14]. In the design process, the Convolutional Two-Stream network model proposed in the literature is referred to and the content of this study is adjusted. The model consists of five parts which are the transformation layer, input layer, convolutional neural network layer, segment consensus layer and Two-Stream fusion layer [15-16], This is shown in **Fig. 2**.

3.1 Transformation Layer

The work of this layer is divided into two main steps. The first step is to convert datasets from AVI video format to framework-jpg images saved as input to the spatial stream. The second step, after converting the JPG image, calculates the resulting dense optical flow field using successive frames of the same video as input to the time flow [17]. The flow chart of the transformation layer is shown in **Fig. 3**.

Convert Video to JPG Image. At the beginning, we first get the file path of all the video data, and then OpenCV was used to read the video to load every frame one by one.

Continuous Frame Calculation of Dense Optical Flow Field. Continuous frames of each video were obtained in the first step of AVI conversion into JPG, and then the Optical Flow motion between successive frames was calculated. In this experiment, TV-L1 was used to calculate Dense Optical Flow. The full name of TV-L1 is Total Variation. It is used to describe the cumulative Variation of function values. The method of L1 can increase its robustness. The OpenCV library is used to calculate TV-L1 optical flow field [18]. The steps are as follows.

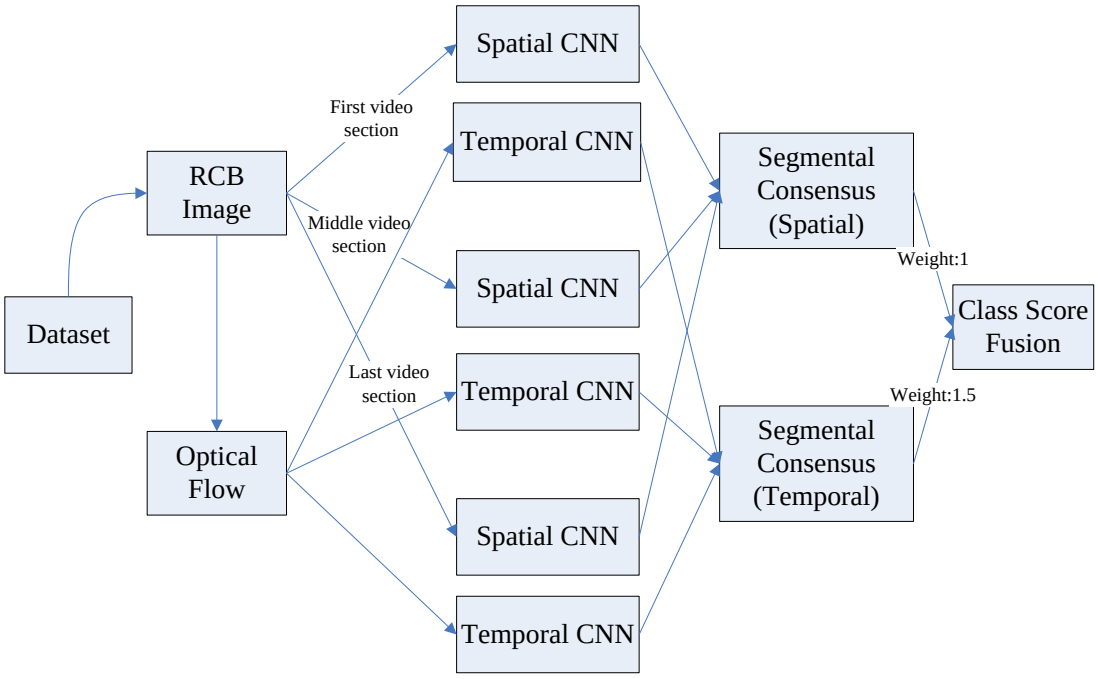

Fig. 2. The proposed model structure diagram

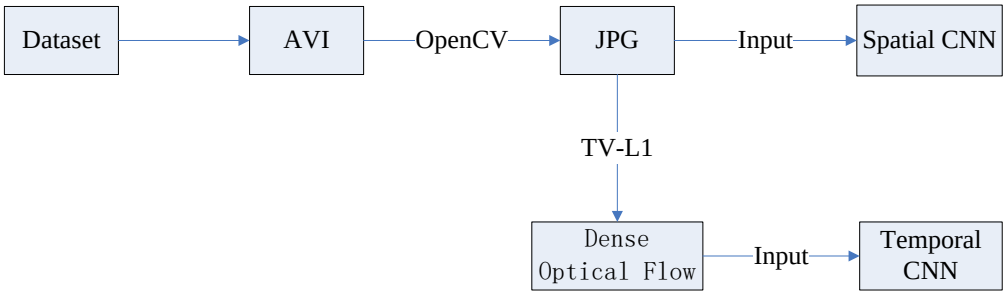

Fig. 3. The flow chart of the transformation layer

(1) Traversed the path of continuous frames converted from all video data, and performed the optical flow function extraction.

(2) Read continuous frames of video and convert BGR tee to GRAY single channel.

(3) Calculate the optical flow field.

(4) The optical flow is linearly dispersed from 0 to 255, and the optical flow field is saved. u and v respectively represent the optical flow field moving along x and y.

3.2 Input Layer

Access Path. The glob method is used to get the file path of all videos, and then the prefix is modified according to this path and directed to the file of RGB image and optical flow field image respectively [19].

Create One-hot Coding Label. Extract the subdirectory under the video dataset file and make it into a dictionary with lower labels. Label each piece of data with a subscript dictionary and replace it with one-hot coding [20].

Image Enhancement. Due to the small amounts of datasets related to video behavior, it is easy to have the problem of overfitting in the actual training. However, image enhancement of video images makes the data become diverse, which can effectively alleviate the problem of overfitting [21].

The image enhancement methods used in this experiment include random size cropping, random left-right reversal, random up-down reversal, and random brightness adjustment. All these methods can improve the robustness and adaptability of the model.

Data Input. In this experiment, dataset was used to process the input data. The MAP method read the images and preprocessed them, and then divided the and training set according to the ratio of 8:2. Finally, we recommend setting the batch to 256. According to the official statement of TensorFlow, when repeating or shuffling a dataset, first writing repeat would blur the meaning of the epoch, but it could effectively improve the performance of the model [22].

3.3 Convolutional Neural Network Layer

The convolutional layer's function is to carry out feature extraction on the data of inputting, and it includes many convolutional cores. Every element of the convolutional core corresponds to a bias vector and a weight coefficient, which is similar to the neuron of a feedforward neural network [23].

$$Z^{l+1}(i,j)=[Z^{l}\otimes\omega^{l+1}](i,j)+b=\sum_{k=1}^{K_l}\sum_{x=1}^{f}\sum_{y=1}^{f}[Z_k^{l}(s_0 i+x,s_0 j+y)\omega_k^{l+1}(x,y)]+b \quad (1)$$

$$L_{l+1} = \frac{L_l + 2p - f}{s_0} + 1 \tag{2}$$

Where $(i, j) \in \{0,1,...L_{l+1}\}$. b is the deviation, Z^{l+1} and Z^l is the convolution output and input of the $l+1$ layer. L_{l+1} is the size of Z_{l+1}. $Z(i, j)$ corresponds to the feature graph's pixels; K is the feature graph's channels number; p, f and s_0 are convolution layer's parameters.

In particular, when the step size $s_0 = 1$, convolution kernel is size $f = 1$, and does not include a filled unit convolution kernel [24].

$$Z^{l+1} = \sum_{k=1}^{K_l}\sum_{i=1}^{L}\sum_{j=1}^{L}(Z_{i,j,k}^{l}\omega_k^{l+1}) + b = \omega_{l+1}^{T} Z_{l+1} + b \tag{3}$$

The convolutional layer includes excitation functions to assist in the expression of complex features, which are expressed as follows [23].

$$A_{i,j,k}^{l} = f(Z_{i,j,k}^{l}) \tag{4}$$

L_p pooling is inspired by the hierarchical structure in the visual cortex, which is a kind of pooling model and expressed as follows [24].

$$A_k^l(i, j) = [\sum_{x=1}^{f}\sum_{y=1}^{f} A_k^l(s_0 i + x, s_0 j + y)^p]^{\frac{1}{p}} \tag{5}$$

Mixed pooling can be described as a linear combination of mean pooling and maximum pooling [26]:

$$A_k^l = \lambda L_1(A_k^l) + L_\infty(A_k^l), \quad \lambda \in [0,1] \tag{6}$$

BP framework is used for learning in supervised learning by Convolutional neural network.

$$\left(\frac{\partial E}{\partial A}\right)_{i,j}^{l} = \sum_{k=1}^{K_l}\sum_{x=1}^{f}\sum_{y=1}^{f}\left[\omega_k^{l+1}(x, y)\left(\frac{\partial E}{\partial A}\right)_{s_0 i+x, s_0 j+y, k}^{l+1}\right] f'\left(A_{i,j}^{l}\right) \tag{7}$$

In the formula, E is the cost function's calculation error, and f' is the excitation function's derivative, and α is the learning rate. If the convolution kernel's forward propagation is calculated by convolution, then the back propagation reverses the convolution kernel to carry out the convolution operation too.

All kinds of regularization methods in neural network algorithm can be used for convolutional neural network to prevent over-fitting. Common regularization methods include L_p-norm regularization, spatial dropout, and random connection deactivation. Regularization adds hidden layer parameters to define the loss function to constrain the complexity of the neural network:

$$L(X, Y, \omega) = L(X, Y, \omega) + \lambda \sum \| \omega \|^p \tag{8}$$

Where $L(X, Y, \omega)$ is the loss function, and the summation term containing the Frobenius norm is called the regularization term, where is the regularization parameter to determine the constraint of the regularization term.

3.3.1 Network Architecture of Xception

Xception structure based on ResNet, but will be replaced by the Convolution layer Separable Convolution. Xception is another improvement to Inception-v3 proposed by Google. It uses Depthwise Separable Convolution to replace the convolution operation in Inception-v3. Xception is slightly more accurate in ImageNet image recognition than Inception-v3, and it has fewer parameters than Inception-v3. At the same time, the residual module of Xception significantly accelerates its convergence and improves its accuracy [27].

Although Depthwise Separable Convolution can bring the ascension of accuracy or theoretical calculation has fallen dramatically, but as a result of the calculation process is relatively fragmented, existing Convolution neural network to realize its efficiency is not very high. Such as the amount of theoretical calculation of Xception in this paper is far less than that of Inception-v3, but the training iteration speed is slower [28].

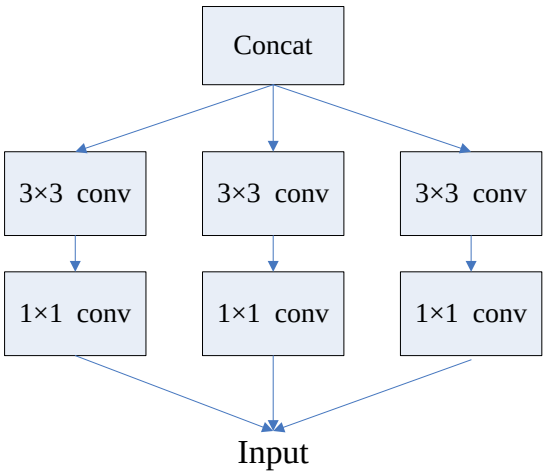

Fig. 4. Simply Inception network

Fig. 4 is a simple version of the Inception model [29]. For an input Feature Map, three groups of Feature maps are first obtained through three groups of 1×1 convolution. Assuming that the number of 1×1 convolution kernel in **Fig. 4** is k_1, the number of 3×3 convolution kernel is k_2, and the number of channels in the input Feature Map is m, then the number of parameters in this simple version is:

$$\mathrm{m}\times k_1+3\times3\times3\times\frac{k_1}{3}\times\frac{k_2}{3}=m\times k_1+3\times k_1\times k_2 \tag{9}$$

If Inception is to divide 3×3 convolution into 3 groups, then consider an extreme case. If we completely separate the Feature Map of k_1 channels obtained by 1×1 of Inception, that is, we use k_1 different convolution to convolve on each channel respectively, and the number of parameters is:

$$\mathrm{m}\times k_1+k_1\times3\times3 \tag{10}$$

More often, we want the output Feature Map of the two convolution groups to be the same. Here, we set the number of channels of Inception's 1×1 convolution as k_2, that is, the number of parameters is:

$$\mathrm{m}\times k_2+k_2\times3\times3 \tag{11}$$

It's number of parameters is $1/k$ of the ordinary convolution, and this form of Inception is called Extreme Inception.

3.3.2 Model Adjustment After Reference to Pre-training

In this experiment, the transfer learning method was used, and the weight parameters obtained from Xception model training on ImageNet were used as the initial weight of the model [30]. This is to extend the learning power of the Xception model through other datasets and apply this extended learning power to the identification of UCF-101 datasets. In this way, the efficiency of model learning can be accelerated, the overfitting problem caused by repeated training because the dataset is not large enough can be alleviated, and the accuracy of the model can be improved.

First, we introduce Keras Xcption model and download the weights referenced after ImageNet. In this step, we need to remove the output layer of the original model, and make the weights of the Xcpetion model to untrainable in preparation for later fine-tuning [31].

Set the new output layer. In this experiment, the target is categorized as class 101, so add the full connection layer of 101 units as the output, and the activation function is Softmax. During the experiment, it was found that it was easy to overfit in the training, so the Dropout layer was added before the output layer, and the rate was set to 0.8.

3.4 Segment Consensus Layer

The video is divided into the first, middle and last three segments, and the RGB video frame and optical flow field of these three segments are obtained randomly by using random Numbers. The model uses a multi-input model. The following model describes a spatial flow as an example.

3.4.1 Model and Code Implementation

Two-stream convolution network is unable to model the time structure of a long range, mainly because it operates only one frame of spatial network or a single heap frame (time network) in a broken segment, so access to the time context is limited. [32].

The model in this paper improves on this shortcoming of Two-Stream convolutional network. Like Two-Stream, it is also composed of space-stream convolutional network and time-stream convolutional network. However, unlike Two-Stream, which only operates on a single frame or a single dense optical Stream frame, the video is segmented several times in this model. The video was divided into the first, middle and last three sections. Each section was evaluated and recognized by the trained Xception network. After completion, the consensus was extracted from the three sections. The extraction methods are maximization, average and weighted average. After practical testing, it is found that the weighted average single flow results are the best, followed by average. However, the result of averaging is better than that of weighted averaging after two-flow fusion, so this model adopts the averaging method to extract segment consensus [33]. The model structure diagram of this part is shown in **Fig. 2**.

The segmented multi-processing method can fully and synthetically extract video image features and improve the robustness and accuracy of the model. The only disadvantage is that multiple stages may make the model larger, and the number of training parameters will lead to the increase of model training time and model space. To overcome this problem, the weight of the trained Xception model was Shared among the three segments when the multi segment model was established. That is to say, for the three-segment spatial flow and the three-segment

temporal flow, only two parameters of the Xception model are needed for the use of the time flow and the space flow respectively, and the number of parameters is almost the same as the original [34].

3.4.2 Model training

On the segment consensus layer, the training method of RGB model and optical flow model is basically the same, except that the RGB model uses the pre-training weight of ImagNet to initialize, while the optical flow model uses the weight of RGB model after training to initialize. This is the cross-input method mentioned by the author of the Two-Stream model. It can alleviate the problem that the optical flow model cannot find the appropriate pre-training data, which may easily lead to overfitting [35].

At the beginning, we refer to the training method of the author of the Two-Stream model. First, freeze the other layers and train only the output layer. Unfreeze part of the top layer and conduct training again [36]. As we can see from the spatial flow training, the training effect is greatly improved compared with the spatial flow result 72.8% of VGG16 training in the Two-Stream network before.

The optimizer used for spatial flow training was SGD (stochastic gradient descent) [37] and the loss function was categorical_crossentropy (multi-classification logarithmic loss function). The initial learning rate was 0.05, and every 10 Epoch decreased to 1/5 of its current level and finally to 0.0001.

In the last layer of training, it is not necessary to train the accuracy to the highest level. After the learning rate reaches 0.005 and the training accuracy is stable, the next round of training can be started. In the experiment, after trying to train the accuracy of the output layer to the peak, when thawing the parameter training of the upper layer, the accuracy will have a big drop at the beginning. As the training gradually improved to the original level after the training, and the feeling did not help improve the final accuracy rate. Therefore, it is recommended to learn the parameters of the output layer to the stable accuracy of 0.005 learning rate during training.

After thawing the model, the learning rate must be adjusted to 0.005 in time, and then the training should continue.

Then we open all the layers after the last layer of training, and the result shows some improvement compared with the thawing of only 33 layers, with an accuracy rate of 82.55%. This result is later tested on the model.evalute function. In order to save time, the test set accuracy during training was only obtained on a set of random frame extraction test sets, with unstable and accurate results. Finally, the test results were obtained on three sets of random video frames.

The optical flow field convolutional neural network uses the crossing input method and starts training with the weight of the spatial flow as the initial weight [38]. At the same time, by referring to the superposition optical flow input proposed by the author of the Two-Stream network, the optical flow field of continuous L frame which contains X and Y directions is superposed together to form an input channel of 224*224*2L. According to the author's experiment, the best training result was obtained when L=10. In the experiment, it was found that when L=10, the hardware requirements of the machine were higher, and the size of Batch_size could only be about 64 on a machine with 4 graphics cards which have 48GB video memory. The results of training may not be up to the ideal level. Compared with L=5, the original author mentioned in the paper that the difference was between 1% and 1.5%. Therefore, when the machine hardware is insufficient, the recommended L value may be appropriately small. If the machine hardware is sufficient, write L=10 for the best effect. In

this experiment, the training situation of Xception flow in L=10, Batch_size=64, and section number 3 was studied.

The optical flow field needs to input from [224,224,3] rewrite as [224,224,20]. In the previous section, the Xception model and its pre-training parameters in ImageNet were referenced in Karas Application package. The reason for writing the detailed model code here is that the Xception contains some residual blocks that skip the parameters to the later layer, so it is not feasible to modify the input layer directly with the preceding convolution and then add the later layer layer by layer.

The way of cross input requires the weights trained by the RGB model, but the input is rewritten in the previous part, the convolution parameter dimension of the second layer also changes accordingly. Therefore, we also need to rewrite the second layer parameters of the RGB model. Here, we first average the parameters of the original three channels, and then duplicate 20 of them to generate the parameters of 20 channels.

3.4.3 Problems in Training

At the beginning of the training, the results were unsatisfactory and the over-fitting of data was serious. Later, a 4 * GTX2080Ti machine was rented on the geek cloud platform for training, and the satisfactory results were obtained. The accuracy of each stage is about 10% higher than before. Additionally, since each segment requires three images to be provided simultaneously, the 4-card machine can only set the Batch size to about 100 after opening all layers of Xception, and 128 up will OOM.

3.5 Two-Stream Fusion Layer

The weighted average fusion method is used in this paper. Spatial flow and time were respectively predicted to obtain the probability scores of each behavior, which were input into the fusion layer for weighted average fusion. The weight of spatial flow and time flow were set as 1 and 1.5 respectively. The model structure diagram of this part is shown in **Fig. 2**.

Firstly, the trained space flow model and optical flow field model are loaded. We note that the output of the last layer does not use the activation function Softmax, because what you need to output is to predict the scores for each category. If you use Softmax, you will ignore the output of all categories except for the highest score, and the fusion effect will not be achieved.

Finally, Two-Stream fusion is achieved by multiplying the output of the original Two-Stream by weight using Keras Lambda layer, and then adding it through the addition layer in the fusion layer to get the weighted average effect. Finally, Softmax activation function is used in the output layer to output the result.

4. Display of Experimental Results

4.1 ImageNet pre-training results

Table 1 shows the results of each stage of training under the Xception model of ImageNet pre-training. In the experiment, it was found that the Temporal Stream had a poor effect on training only the last layer, and the spatial Stream could be inferred that the full-layer open training was more effective than just a few Blocks after the training, so the Temporal Stream only trained the patterns of all the open layers. The final Two-Stream fusion is also derived from the full model training results of time flow and space flow.

Table 1. Results of ImageNet pre-training

Number of layers of training	Spatial Stream	Temporal Stream	Two-Stream
The Only Last Layers	68.2%	None	None
Fine-tuning block11-14	77.4%	None	None
All Layers	85.5%	88.6%	92.2%

4.2 Use Kinetics400 Pre-training to Improve the Model

The biggest problem of the above experiment is overfitting, because UCF101 dataset is not big enough for video behavior recognition training, and the performance of Xception is not fully played [39].

DEEPMIND has in recent years published a number of annotated video datasets such as Kinetics400, Kinetics600, Kinetics700 and so on. However, these datasets are also unfriendly to domestic users, as it only provides the tagged video table (CSV), and the specific video resources need to be crawled on YouTube according to the address and video name provided. The Kinetics400 dataset here is extracted from other crawled compressed shared resources and is incomplete without a small part. The total number of videos is 763017 [40]. The training set, test set, and verification set split ratio is 8:1:1.

Due to hardware problems, the time required to calculate kinetics400 dense optical flow is extremely long. Taking 13,320 videos of UCF101 as an example, it takes 30 hours to calculate the dense optical flow. Therefore, instead of time flow training, we only do space flow. the accuracy rate of space flow is about 63%. Because the batch size can only be set as 10 due to the limitation of hardware during training, the accuracy may be lost.

Due to the use of kineticS400 space flow weight initialization and UCF101 training, the training results were significantly improved, with full model training reaching 87.24%.

Table 2 compares the recognition accuracy with that of other classical algorithms on UCF101 dataset, and the data show that the two-stream Convolutional Neural Network we proposed can effectively improve the video recognition accuracy.

By comparing with the results of other algorithms, it can be seen that the two-stream Convolutional Neural Network proposed in this paper, on the basis of absorbing the excellent results of predecessors, exerts the characteristics of high accuracy of Xception Network. On this basis, we also introduce transfer learning into it.

The focus of this study is to use the spatio-temporal flow neural network model to determine the types of human behavior from the first, middle and last segments of the video. The possibility scores of each type were obtained by extracting the three result features, and the most likely result was obtained by synthesizing the spatio-temporal flow score, so as to achieve the purpose of behavior recognition. Moreover, a more effective Xception model was used this time, and the accuracy of the model was improved again through kineticS400 dataset pre-training at last.

Table 2. Comparison with classical algorithms on UCF101 dataset

Algorithms	**Accuracy**
iDT+FV [41]	85.9%
Motion Vector + FV [34]	78.5%
RGB + Enhanced Motion Vector [34]	86.4%
C3D + liner SVM [42]	82.3%
Ours	**87.2%**

5. Conclusion

5.1 Summarize

In this applied research, through continuous efforts, the expected goal is basically achieved. According to the study and research, the design of the original Two-Stream model is improved, and the accuracy of both Single-Stream and Two-Stream models is greatly improved. However, the model used in this article also has a number of areas that could be improved. First, although it is improved to divide the video into three segments, it also expands the calculation amount of space occupied by the model. In view of the phenomenon that multi-segmentation is more effective in extracting features from long-duration videos, the number of segments of the model can be designed to be more flexible according to the video length to be recognized by the actual task. The second is that the fusion method of segment consensus and Two-Stream part is relatively crude, so more effective fusion methods can be tried to be used instead.

5.2 Future prospects

In this experiment, it can be seen that in deep learning, the factors affecting the results mainly depend on three aspects: algorithm model, dataset and hardware equipment. Compared with the original Two-Stream author's model, this experiment used a more advanced Xception network and got better results than before. In the end, a larger dataset is used for pre-processing training to obtain a higher accuracy rate. With multiple graphics cards, the batch size is increased to make the model fit the data better.

Nowadays, with the continuous improvement of hardware performance, more and more efficient deep learning algorithms and models have been proposed. Among them, Transformer network model, which is the first to be applied in the field of natural language processing, has achieved the cutting-edge level in the field of natural language processing [43]. Google launched the Transformer model in 2017. The core part of the model is Attention, which enables pair comparison of all sequences and solves the problem of distance dependence that CNN and RNN cannot solve. Then, the Bert model [44] proposed by Google and the upgraded version of Bert Mask model [45] further improved the recognition rate compared with the Transformer model. The Transformer network model can use more computer memory more efficiently and is more powerful when it comes to complex tasks. The attention mechanism used by the Transformer structure allows computation to be done in parallel, extracting all the information we need from the input and its interrelationships simultaneously.

In recent years, Transformer model, which is very popular in natural language processing and has excellent performance, has been successfully applied in the field of computer vision, such as target detection and video understanding, and has achieved better experimental results than traditional convolutional neural network. At present, the applications of Transformer network model in the field of image are mainly divided into two kinds. One is to take the Attention mechanism as a layer and nested it into the convolutional neural network. The convolutional neural network output of Attention is made into a concat, which is connected and output to the next layer, so it is going to be faster. However, the efficiency of this method still needs to be improved, because this method still uses the operation of convolutional layer, so this method is usually suitable for pictures with relatively small images and not very high definition [46]. The second method is to remove Convolutional Neural Network and directly use Transformer network model in image processing. In 2021, Chen et al. [47] proposed a Transformer model suitable for vision, named Visformer. With the same computational complexity, its performance is even better than that of ordinary Transformer model.

Undoubtedly, an excellent model has been added to improve the operation. It is completely built by Transformer structure, and the image is split and processed, and then directly input Transformer structure to complete the image classification task. In addition, [48] proposed a multiscale vision Transformer (MVIT) for video and image recognition, linking the idea of multiscale feature levels with the Transformer model. Multiscale Transformer has several channel resolution scale levels. Starting with input resolution and a small channel dimension, the stage extends the channel capacity hierarchically while reducing the spatial resolution. This creates a multiscale feature pyramid, with the early layers modeling simple low-level vision information at high spatial resolution, and the deeper layers modeling spatially rough but complex higher-dimensional features. In addition, the TFPose model proposed by [49] transforms the pose estimation task into a sequence prediction problem, which avoids the disadvantage of pose estimation based on heat map. In addition, the TFPose model can adaptively focus on the features most relevant to the target key points, and internally utilizes the structural relations between the key points. This method overcomes the feature misalignment problem of previous methods based on regression to a large extent and improves the performance significantly. In addition, [50] proposes a new method of using Transformer to segment transparent objects in the field, which not only proposes a new fine-grained transparent object segmentation dataset, but also proposes a new segmented pipeline Trans2Seg based on Transformer. First, Trans2Seg's Transformer encoder provides a global acceptance domain instead of CNN's local acceptance domain. Second, the model defines semantic segmentation as a dictionary lookup problem, and a set of learnable prototypes are designed to serve as queries for the decoder of the Trans2Seg converter, with each prototype learning the statistics of a category in the entire dataset. The recognition rate of this method exceeds the resolution of the network model based on CNN structure.

In the next step, we will try to apply Transformer network model architecture to video content recognition according to the method proposed in the above paper, so as to further improve the accuracy and efficiency of recognition.

References

[1] C. Zhu, Y. K. Wang, D. B. Pu, M. Qi, H. Sun and L. Tan, "Multi-modality video representation for action recognition," *Journal on Big Data*, Vol. 2, No. 3, pp. 95-104, 2020. Article (CrossRef Link)

[2] J. Li, Y. Lv, B. Ma, M. Yang, C. Wang et al., "Video source identification algorithm based on 3d geometric transformation," *Computer Systems Science and Engineering*, vol. 35, no.6, pp. 513–521, 2020. Article (CrossRef Link)

[3] W. Fang, L. Pang and W. N. Yi, "Survey on the application of deep reinforcement learning in image processing," *Journal on Artificial Intelligence*, vol. 2, no. 1, pp. 39-58, 2020. Article (CrossRef Link)

[4] W Xing, Y Li, S Zhang, "View-invariant gait recognition method by three-dimensional convolutional neural network," *Journal of electronic imaging*, vol. 27, no. 1, pp. 248-258, 2018. Article (CrossRef Link)

[5] X. Shi, C. Ma, Y. Rao, X. Chen and J. Zhang, "Video preview generation for interactive educational digital resources based on the gui traversal," *Intelligent Automation & Soft Computing*, vol. 26, no.5, pp. 917–932, 2020. Article (CrossRef Link)

[6] M. Zhang, H. Xu, X. Wang, M. Zhou, S. Hong, "Application of Google TensorFlow machine learning framework," *Microcomputers and applications*, vol. 36, no.10, pp. 58–60, 2017. Article (CrossRef Link)

[7] W. Yin, S. Ebert, H. Schütze, "Attention-Based Convolutional Neural Network for Machine Comprehension," in *Proc. of the Workshop on Human-Computer Question Answering 2016*, San Diego, California, pp. 15-21, 2016. Article (CrossRef Link)

[8] D. Guan, W. Yuan, Z. Jin, et al., "Undiagnosed samples aided rough set feature selection for medical data," in *Proc. of Parallel Distributed and Grid Computing (PDGC), 2012 2nd IEEE International Conference on IEEE*, Solan, India, pp. 639-644, 2012. Article (CrossRef Link)
[9] J. Liu, D. He, "Research on The Comment Text Classification based on Transfer Learning," in *Proc. of 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC) IEEE*, Chongqing, China, pp. 191-195, 2020. Article (CrossRef Link)
[10] J. Lu, et al., "Application research of convolution neural network in image classification of icing monitoring in power grid," *EURASIP Journal on Image and Video Processing*, 2019. Article (CrossRef Link)
[11] W. Fang, F. Zhang, Y. Ding and J. Sheng, "A new sequential image prediction method based on lstm and dcgan," *Computers, Materials & Continua*, vol. 64, no. 1, pp. 217–231, 2020. Article (CrossRef Link)
[12] T. Kaur, T. K. Gandhi, "Automated Brain Image Classification Based on VGG-16 and Transfer Learning," in *Proc. of 2019 International Conference on Information Technology (ICIT) IEEE*, 2019. Article (CrossRef Link)
[13] S. Zhang, et al., "Fast Image Recognition Based on Independent Component Analysis and Extreme Learning Machine," *Cognitive Computation*, US, vol. 6, pp. 405-422, 2014. Article (CrossRef Link)
[14] P. Ma, Z. Tong, Y. Wang, "Research on Human Behavior Recognition Based on Convolutional Neural Network," in *Proc. of China Conference on Wireless Sensor Networks*, Springer, Singapore, pp. 131-144, 2019. Article (CrossRef Link)
[15] D. Wu, L. Shao, "Deep Dynamic Neural Networks for Gesture Segmentation and Recognition," in *Proc. of European Conference on Computer Vision Springer*, Cham, pp. 552-571, 2014. Article (CrossRef Link)
[16] T. Yang, Z. Chen, W. Yue, "Spatio-temporal double stream character action recognition model based on video deep learning," *Computer Applications*, vol.38, no. 3, pp. 895-899, 2018. Article (CrossRef Link)
[17] S. Nah, T. H. Kim, K. M. Lee, "Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring," in *Proc. of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 257-265, 2017. Article (CrossRef Link)
[18] W. Z. Shen, C. L. Zhang, Z. L. Chen, "Research on Automatic Counting Soybean Leaf Aphids System Based on Computer Vision Technology," *Journal of Agricultural Mechanization Research*, pp. 1635-1638, 2007. Article (CrossRef Link)
[19] N. Ahmed, J. I. Rafiq, M. R. Islam, "Enhanced Human Activity Recognition Based on Smartphone Sensor Data Using Hybrid Feature Selection Model," *Sensors (Basel, Switzerland)*, vol. 20. no. 1. 2020. Article (CrossRef Link)
[20] H. F. Sang, C. Xu, D. Y. Wu, J. Huang, "Research on the Real-Time Multiple Face Detection, Tracking and Recognition Based on Video," *Applied Mechanics & Materials*, vol. 373, pp. 442-446, 2013. Article (CrossRef Link)
[21] W. H. Wang, J. Y. Tu, "Research on License Plate Recognition Algorithms Based on Deep Learning in Complex Environment," *IEEE Access*, vol. 8, pp. 91661-91675, 2020. Article (CrossRef Link)
[22] R. L. Li, L. L. Wang, K. Wang, "A survey of human body action recognition," *Pattern Recognition and Artificial Intelligence*, vol.687, no. 12, pp. 3559-3569, 2014.
[23] I. Goodfellow, Y. Bengio, A. Courville, *Deep learning*, Cambridge: MIT press, vol. 1, 2016, pp. 326-366.
[24] J. Gu, et al., "Recent Advances in Convolutional Neural Networks," *Pattern Recognition*, vol. 77, pp. 354-377, 2018. Article (CrossRef Link)
[25] M. D. Zeiler, and R. Fergus, "Stochastic Pooling for Regularization of Deep Convolutional Neural Networks," *Eprint Arxiv*, 2013. Article (CrossRef Link)
[26] D. Yu, et al., "Mixed Pooling for Convolutional Neural Networks," in *Proc. of International Conference on Rough Sets & Knowledge Technology*, pp.364-375, 2014. Article (CrossRef Link)

[27] J. Zhang, X. Li, "Handwritten character recognition based on TensorFlow platform," *Computer Knowledge and Technology*, vol. 12, no. 16, pp. 199-201, 2016. Article (CrossRef Link)
[28] C. Feichtenhofer, A. Pinz, A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1933-1941, 2016. Article (CrossRef Link)
[29] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *Proc. of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800-1807, 2017. Article (CrossRef Link)
[30] K. Simonyan, A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in Neural Information Processing Systems*, 2014. Article (CrossRef Link)
[31] L. Wang, Y. Xiong, Z. Wang, et al., "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. of European Conference on Computer Vision*, Springer, Cham, pp. 20-36, 2016. Article (CrossRef Link)
[32] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318-327, 2020. Article (CrossRef Link)
[33] Y. Zhao, Y. Xiong, L. Wang, et al., "Temporal action detection with structured segment networks," *The IEEE International Conference on Computer Vision*, pp. 2933-2942, 2017. Article (CrossRef Link)
[34] B. Zhang, L. Wang, Z. Wang, et al., "Real-time action recognition with enhanced motion vector CNNs," in *Proc. of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2718-2726, 2016. Article (CrossRef Link)
[35] L. Sun, K. Jia, D. Y. Yeung, et al., "Human action recognition using factorized spatio-temporal convolutional networks," in *Proc. of 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4597-4605, 2015. Article (CrossRef Link)
[36] G. W. Taylor, R. Fergus, Y. LeCun, et al., "Convolutional learning of spatio-temporal features," in *Proc. of European conference on computer vision*, Springer, Berlin, Heidelberg, pp. 140-153, 2010. Article (CrossRef Link)
[37] S. Yan, Y. Xiong, D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018. Article (CrossRef Link)
[38] S. Guan, et al., "SYSU iSEE submission to Moments in Time Challenge 2018," *Computer Vision and Pattern Recognition*, 2018. Article (CrossRef Link)
[39] H. Lee, S. Agethen, C. Lin, H. Hsu, P. Hsu, Z. Liu, H. Chu and W. Hsu, "Multi-Modal Fusion for Moment in Time Video Classification," *Computer Vision and Pattern Recognition*, 2018. Article (CrossRef Link)
[40] C. Li, Z. Hou, J. Chen, Y. Bu, J. Zhou, Q. Zhong, D. Xie and S. Pu, "Team DEEP-HRI Moments in Time Challenge 2018 Technical Report," *Computer Vision and Pattern Recognition*, 2018. Article (CrossRef Link)
[41] H. Wang, C. Schmid, "Action Recognition with Improved Trajectories," in *Proc. of 2013 IEEE International Conference on Computer Vision IEEE*, pp. 3551-3558, 2013. Article (CrossRef Link)
[42] Z. Qiu, T. Yao, T. Mei, "Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks," in *Proc. of 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5534-5542, 2017. Article (CrossRef Link)
[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., "Attention Is All You Need," *arXiv*, 2017. Article (CrossRef Link)
[44] J. Devlin, M. W. Chang, K. Lee, K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv. 1810.04805*, 2019. Article (CrossRef Link)
[45] Y. Cui, W. X. Che, T. Liu, B. Qin, Z. Q. Yang, S. J. Wang, G. et al., "Pre-Training with Whole Word Masking for Chinese BERT," *arXiv:1906.08101*, 2019. Article (CrossRef Link)
[46] I. Bello, B. Zoph, A. Vaswani, J. Shlens, Q. V. Le, "Attention Augmented Convolutional Networks," *arXiv:1904.09925v5*, 2019. Article (CrossRef Link)

[47] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. H. Zhai, T. Unterthiner, et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv. 2010.11929*, 2020. Article (CrossRef Link)
[48] H. Q. Fan, B. Xiong, K. Mangalam, Y. H. Li, Z. C. Yan, J. Malik, et al., "Multiscale Vision Transformers," *arXiv. 2104.11227*, 2021. Article (CrossRef Link)
[49] W. Mao, Y. Ge, C. Shen, Z. Tian, X. Wang, Z. Wang, "TFPose: Direct Human Pose Estimation with Transformers," *arXiv:2103.15320*, 2021. Article (CrossRef Link)
[50] E. Z. Xie, W. J. Wang, W. H. Wang, P. Sun, H. Xu, D. Liang, et al., "Segmenting Transparent Object in the Wild," in *Proc. of Computer Vision – ECCV 2020*, pp 696-711, 2020. Article (CrossRef Link)

Han Qiao received bachelor's degree from Qufu Normal University in the years 2015. He is studying for the master's degree in Electronic information at South China Normal University. His research interests are image processing and machine learning.

Shuang Liu received bachelor's degree in Electronic Information Science and Technology at Hubei University of Technology, Wuhan, China, in 2019. She is studying for the master's degree in Software Engineering at South China Normal University. Her research interests are machine learning and stereo vision.

Qingzhen Xu received PhD from Sun Yat-Sen University in 2006. He is now serving as a professor in South China Normal University. His main focus is on 3D clothing CAD, financial big data analysis, virtual surgery, intelligent surgery robot, image processing.

Shouqiang Liu received PhD in computer science from South China University of Technology University in 2012. He is now serving in School of Artifificial Intelligence, Faculty of Engineering at South China Normal University. His main focus is on information security, multimedia mining, Machine Vision, Natural Language Processing, Knowledge Graph. He has been studying for many years on gait recognition, behavior recognition and deepfake research and application.

Wanggan Yang is a Ph.D. candidate in Public Policy at Southern University and Agricultural & Mechanical College, Baton Rouge, USA. His research focuses on big data and international relations. He received his bachelor's degree in Environmental Engineering at Wuhan University, Wuhan, China, in 2003, master's degree in Environmental Engineering at Huazhong Agricultural University, Wuhan, China, in 2010, and master's degree in Civil Engineering and Master of Business Administration (MBA) at Louisiana State University, Baton Rouge, USA, in 2010 and 2015 respectively.