Computers, Materials & Continua CMC, vol.63, no.3, pp.1545-1561, 2020

Hidden Two-Stream Collaborative Learning Network for Action Recognition

Shuren Zhou[1, *], Le Chen[1] and Vijayan Sugumaran[2]

Abstract: The two-stream convolutional neural network exhibits excellent performance in the video action recognition. The crux of the matter is to use the frames already clipped by the videos and the optical flow images pre-extracted by the frames, to train a model each, and to finally integrate the outputs of the two models. Nevertheless, the reliance on the pre-extraction of the optical flow impedes the efficiency of action recognition, and the temporal and the spatial streams are just simply fused at the ends, with one stream failing and the other stream succeeding. We propose a novel hidden two-stream collaborative (HTSC) learning network that masks the steps of extracting the optical flow in the network and greatly speeds up the action recognition. Based on the two-stream method, the two-stream collaborative learning model captures the interaction of the temporal and spatial features to greatly enhance the accuracy of recognition. Our proposed method is highly capable of achieving the balance of efficiency and precision on large-scale video action recognition datasets.

Keywords: Action recognition, collaborative learning, optical flow.

1 Introduction

Understanding the content in the video is an important part of computer vision, such as action recognition. Comparing to the images with only static information, the videos have more temporal information. Karpathy et al. [Karpathy, Toderici, Shetty et al. (2014)] adopted superimposing video multi-frame input to the network for action recognition learning, but this method is worse than manually extracting features. Simonyan et al. [Simonyan and Zisserman (2014)] proposed two-stream convolutional networks, which means that deep learning has taken a major step in action recognition. The two-stream convolutional network is divided into two parts, one processes RGB images, the other

[1] School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, 410114, China.

[2] Department of Decision and Information Sciences, School of Business Administration, Oakland University, Rochester, 48309, USA.

[*] Corresponding Author: Shuren Zhou. Email: zsr@csust.edu.cn.

Received: 22 January 2020; Accepted: 28 February 2020.

processes optical flow images. Then the work jointly trains the model with the extracted features and finally classifies the actions with the trained model.

Although the two-stream method achieves good performance, it relies on extracting the optical stream from the video in advance, and then learning the optical flow characteristics for action recognition which results in a reduction in the efficiency of the entire network. In order to solve this problem, a variety of methods have been proposed to capture motion information in the videos other than optical flow methods, such as recurrent neural network (RNN) [Du, Wang and Wang (2015)] and 3D CNN [Qiu, Yao and Mei (2017); Tran, Wang, Torresani et al. (2018)]. Some new feature representations of motion information are also proposed, such as motion vectors [Zhang, Wang, Wang et al. (2016); Wang, Long, Wang et al. (2017)], RGB image differences and warped optical flow fields [Wang, Xiong, Wang et al. (2016)]. However, the traditional optical flow methods for human action recognition are more effective than these feature representations. Recently, Zhu et al. [Zhu, Lan, Newsam et al. (2018)] use the CNN method to learn optical flow, implicitly generating motion information for action recognition, effectively avoiding expensive calculations and massive storage and increasing the speed of the entire task. This method solved the problem more directly. However, since only weighted fusion is performed, the interaction between spatial features and temporal features cannot be captured, which results in the situation that there is one flow success and the other flow failure and affects the overall action recognition efficiency.

Therefore, the spatial and temporal integration strategy is particularly important, and many existing methods [Karpathy, Toderici, Shetty et al. (2014); Ji, Xu, Yang et al. (2012); Tran, Bourdev, Fergus et al. (2015)] build appropriate integration strategies by exploiting the advantages of convolution. Compared with the original methods using Fisher Vector [Perronnin, Sánchez and Mensink (2010)], HOF [Laptev, Marszałek, Schmid et al. (2008)], and dense trajectory features [Wang and Schmid (2013); Wang, Qiao and Tang (2015)], these methods which directly use CNN for video action recognition have no crushing advantage. Although CNN has achieved outstanding performance in image analysis tasks, CNNs cannot make full use of the spatial-temporal features in video understanding tasks. in order to make full use of spatial-temporal features, in addition to using standard CNN streams to capture appearance information, some recent methods [Simonyan and Zisserman (2014); Wang, Xiong, Wang et al. (2016)] have attempted to input video optical flow images into another CNN to extract features that contain video motion information. However, we have found that in these models, usually with one stream failing, while the other stream is still correctly misclassified. Therefore, the weighted average fusion strategy of the original two-stream method cannot fully utilize the apparent information and motion information of the video. On the contrary, we believe that the apparent information and motion information in the video should be mutually reinforcing. Recently, two-stream Collaborative Learning with spatial-temporal attention for video classification (TCLSTA) [Peng, Zhao and Zhang (2018)] paid spatial-temporal attention to video static and motion features so as to distinguish the different contributions of different regions in static frames to the final recognition result and discriminative frames in the frame sequence. Then TCLSTA uses the discriminative static and motion features extracted from the spatial-temporal attention model to mutually enhance representation learning and optimizes the combined weight of frames and the optical flow of video classification. But TCLSTA relies on extracting the optical flow from the video in advance, and then learning the optical flow features for action recognition, which greatly reduces the efficiency of the entire system.

In order to solve these two limitations and pursue the balance between speed and precision, our work proposes a hidden two-stream collaborative learning method for action recognition without storing pre-computed optical flow, which not only improves the efficiency of the whole network, but also captures spatial features and the interaction of temporal features and improves the accuracy of action recognition.

Overall, the paper has two contributions as follows:

1. We propose a novel framework, HTSC for action recognition (Section 3) without pre-computing optical flow, effectively avoiding expensive computational and massive storage, which improves the efficiency of the entire network.

2. Our work can directly extract the motion information features from the frame sequence, and guide the spatial features and temporal features to each other, improving the accuracy of recognition.

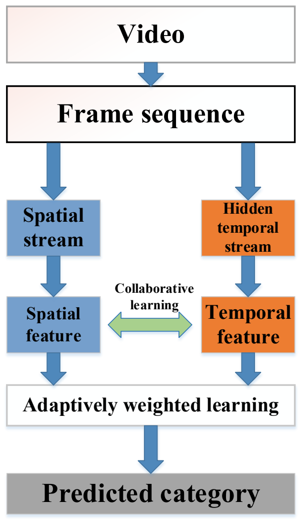

Figure 1: The steps of the proposed HTSC

2 Related work

Understanding what's in the video is an important part of computer vision. For example, video action recognition. In recent years, video human action recognition has made great achievements. At first, traditional hand-crafted extraction of frames, such as improved dense trajectory (IDT) [Wang and Schmid (2013)], is the method with the best effect, the best stability and the highest reliability before deep learning applied to this field, but the speed of this algorithm is slow. Convolutional neural network (CNN) [Karpathy, Toderici, Shetty et al. (2014); Zhu, Lan, Newsam et al. (2018)] is usually several orders of magnitude faster than IDT.

Deep CNN is gaining its popularity in recent years [Tang, Yang, Zhou et al. (2015); Wang, Gao, Yin et al. (2018); Xiang, Shen, Qin et al. (2019); Chen, Xia, Wang et al. (2019)], it has achieved the most advanced performance in image classification [Laptev, Marszałek, Schmid et al. (2008); Wang and Schmid (2013)]. Many works have designed deeper CNNs in order to apply CNN more effectively to action recognition tasks [Zhang, Wang, Wang et al. (2016); Ng, Hausknecht, Vijayanarasimhan et al. (2015); Qiu, Yao and Mei (2017); Wang, Qiao and Tang (2015); Peng, Zhao and Zhang (2018); Carreira and Zisserman (2017); Gui and Zeng (2019); Zhang, Jin, Sun et al. (2018)]. For example, several feature fusion strategies have been explored in the Sport1M dataset [Tran, Bourdev, Fergus et al. (2015)]. At the same time, the two-stream method proposes two CNNs for video action recognition, one of which is a static image stream and the other is an optical flow stream. Finally, the two streams are merged to capture static appearance information and motion information [Ng, Hausknecht, Vijayanarasimhan et al. (2015)] in video. Method [Tran, Bourdev, Fergus et al. (2015)] uses a 3D convolution kernel to extract features from a series of dense RGB frames. The temporal segment network (TSN) [Wang, Xiong, Wang et al. (2016)] first decomposes the video into static frames and optical flow images, then samples them and uses two CNNs to extract features, thereby extracting features containing video static information and video motion information. Method [Ng, Hausknecht, Vijayanarasimhan et al. (2015)] first use CNN to extract the features of static frames in order to better capture video motion information, and then use the long short-term memory (LSTM) model to explore the relationship between videos. Recently, I3D network [Carreira and Zisserman (2017)] used two streams CNN with expanded 3D convolution to achieve the most advanced performance on Kinetics data set on a dense RGB and optical flow sequence [Qiu, Yao and Mei (2017)].

A big disadvantage of the two-stream method is that it cannot model on a long-time video, and only extract temporal context for continuous video frames. In order to solve this problem, TSN network proposes to divide video into K segments, randomly select a snippet from each segment, and then apply two-stream method to these snippets, and finally integrate the features extracted from these snippets.

However, the original two-stream method has two major disadvantages: first, the optical flow must be extracted from video in advance, and then the optical flow features are learned for action recognition, which greatly reduces the efficiency of the entire system. Secondly, the spatial CNN and temporal CNN in the two-stream method are independent of each other. Only a simple weighted average fusion strategy is performed to obtain the final prediction. It cannot learn the subtle spatial-temporal relations. In order to explore better fusion strategies, Laptev et al. [Laptev, Marszałek, Schmid et al. (2008)] compared multiple CNN connection methods, but none of these methods can make good use of the static information and motion information of the video. Recently, TCLSTA designed a static-motion collaborative learning model, which enhanced the spatial and temporal features of each other, and optimized the combined weights of frames and optical flow. However, it relies on extracting optical flow from video in advance, and then learning optical flow features for action recognition, which greatly reduces the efficiency of the entire system. The hidden two-stream collaborative learning method proposed in this paper does not need to extract the optical flow in advance, which greatly improves the network efficiency, and at the same time can better capture the spatial-temporal interaction, achieving the balance of speed and precision.

3 Method

This section describes our proposed hidden two-stream collaborative learning method in detail, which includes two models: hidden two-stream model and collaborative learning model. Our idea is shown in Fig. 2.

In Section 3.1, hidden two-stream model was introduced. We first decompose the video into a sequence of frames, and then send them to the spatial stream CNN and hidden temporal CNN, respectively. The hidden temporal CNN obtains the motion features and spatial stream CNN obtains the spatial features.

In Sections 3.2 and 3.3, we introduced a collaborative learning model that performs a collaborative learning network to optimize the spatial and temporal features. Then adaptive weighted learning model learns the fusion weight of each video category adaptively and finally obtains the prediction result.

3.1 Hidden two-stream model

Given the frame sequence of the video, we hope to learn not only the static appearance features, but also the motion information from the frame sequence, which serves as the basis for judging the video action category. We can effectively realize the action recognition of the static image by two-stream network [Simonyan and Zisserman (2014)], so our spatial stream network adopts the same setting as the two-stream network, and is used to capture the static appearance information of the images. FlowNet [Dosovitskiy, Fischer, Ilg et al. (2015)] proves that optical flow can be estimated by CNN. We hope to use the CNN to learn the optical flow information of the frame sequence, which contributes to the human action recognition task.

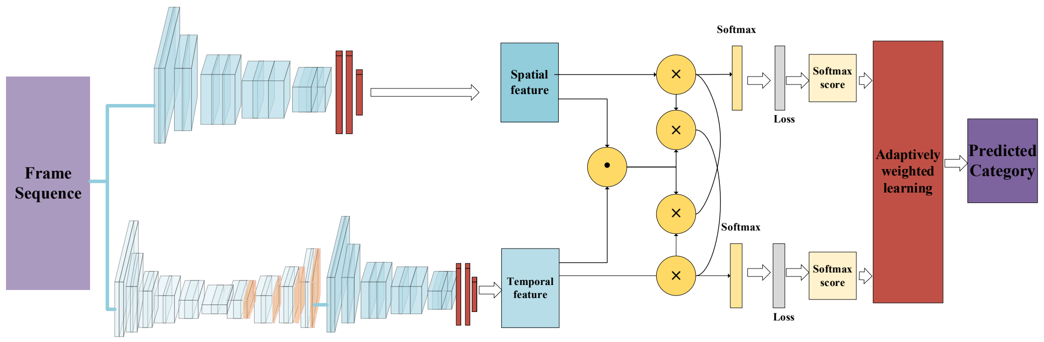

Figure 2: The proposed HTSC method consists of two parts: (1) hidden two-stream model learns from the frame sequence to spatial (static) features and temporal (motion) features. (2) collaborative learning model uses the complementarity of motion information and static information to optimize the motion features and static features, which improves the accuracy of action recognition

3.1.1 Spatial stream

Static appearance features (colors, lighting, textures, contour, etc.,) are a useful clue because some actions are closely related to specific objects and scenes. The input of our spatial stream Convnet is a static frame of video, which can effectively realize the action recognition of static images. In fact, the action classification of static frames (spatial streams) is inherently quite competitive. Due to the outstanding performance of CNN in image recognition tasks, we pre-train our spatial stream network on the basis of recent advance large-scale image recognition method [Perronnin, Sánchez and Mensink (2010)].

3.1.2 Temporal stream

Although some actions can be recognized using a single frame image, some actions are dependent on motion information. Therefore, the temporal stream of the original two-stream network takes the optical stream image as an input. The original two-stream network needs to obtain the optical flow images from the video in advance using methods such as TVL1. The information contained in the optical flow images is useful for the action recognition task [Simonyan and Zisserman (2014)]. The original method needs to extract the optical flow information in advance, but the extraction speed is slow. The storage of the optical flow images requires additional storage space. We consider optical flow prediction as an image reconstruction problem [Jason, Harley and Derpanis (2016); Zeng, Dai, Li et al. (2018)]. We use the hidden temporal stream to learn the optical flow information of the frame sequence that contributes to our task and generate an effective optical stream of adjacent frames. Taking adjacent frames f_1 and f_2 as inputs, if the predicted optical flow and f_2 can reconstruct f_1, the network learns the motion information.

Our temporal flow is divided into two parts: optical flow estimation and feature extraction. Our network details can be seen in Section 4.2.

We calculate losses on multiple scales in the network of optical flow. Three loss functions [Zhu, Lan, Newsam et al. (2018)] are adopted to generate optical flow of higher quality, which can be written as follows:

Standard pixel reconstruction error function:

$$L_P = \frac{1}{mn}\sum_g^m \sum_h^n F(f_1(g,h) - f_2(g + V_{g,h}^x, h + V_{g,h}^y)) \tag{1}$$

where L_P denotes the standard pixel reconstruction error function, where $V_{g,h}^x$ and $V_{g,h}^y$ are the estimated optical flow in the horizontal and vertical directions of the pixel (g, h), and m and n represent the height and width of frame f_1 and f_2. In order to reduce the effect of outliers, we adopt the equation $F(x) = (x^2 + \varepsilon^2)^\alpha$ [Lai, Huang, Ahuja et al. (2017)] (a variant of L1 loss, first used as a loss function in LapSRN)

$$L_{sm} = F(\nabla_a^x) + F(\nabla_b^x) + F(\nabla_a^y) + F(\nabla_b^y) \tag{2}$$

where L_{sm} is a smoothness loss function, which solves the aperture problem that leads to blurring when estimating motion in a non-textured region. ∇_a^x and ∇_b^x are gradients of the predicted optical flow field V^x in each direction. Analogously, ∇_a^y and ∇_b^y are the gradients of the optical flow field V^y, F (x) is the same as in Eq. (1)

$$\mathrm{SSMI}(f_{B_1}, f_{B_2}) = \frac{(2\mu_{B1}\mu_{B2}+k_1)(2\sigma_{B1B2}+k_2)}{(\mu_{B1}^2+\mu_{B2}^2+k_1)(\sigma_{B1}^2+\sigma_{B2}^2+k_2)} \tag{3}$$

Structural similarity (SSIM) loss function [Wang, Bovik, Sheikh et al. (2004)], which helps us learn the structure of frames, where f_{B_1} and $f_{B_2}$are local blocks of frames f_1 and f_2, respectively, and we set the size to 8×8. μ_{B1} and μ_{B2} are average values of the image blocks f_{B_1} and f_{B_2}, σ_{B1} and σ_{B2} are the variances of the two image blocks, σ_{B1B2} is the covariance, and k_1 and k_2 are two constants used to stabilize the division. In the experiments, we set it to 0.0001 and 0.001, respectively.

$$L_{ss} = \frac{1}{I}\sum_{i}^{I}(1 - SSIM(f_{1i}^{1}, f_{1i}^{\prime})) \quad (4)$$

In order to compare the similarity between two frames f_1 and $f_1^{\prime}$, we design a loss function L_{ss}, where I am the number of local blocks we can extract from the image, and i is the index of the local block.

$$L_s = \lambda_1 L_p + \lambda_2 L_{sm} + \lambda_3 L_{ss} \quad (5)$$

$$L_{all} = \sum_{s=1}^{5} \delta_s L_s \quad (6)$$

where δ_s is the parameter to regulate the losses of each scale [Zhang, Yin, Yang et al. (2017)]. The loss of each scale is the weighted sum of the previous three loss functions.

The feature extraction part is also similar to the CNN structure of spatial stream. Before sending the estimated optical flow to the CNN that extracts features, we normalize it to a range between 0 and 255. This normalization is important for good temporal stream performance [Simonyan and Zisserman (2014)]. Finally, the temporal stream extracts features containing optical flow information.

3.2 Collaborative learning model

For the two-stream method and its derivative methods [Simonyan and Zisserman (2014); Zhu, Lan, Newsam et al. (2018); Peng, Zhao and Zhang (2018); Diba, Pazandeh and Van Gool (2016); Feichtenhofer, Pinz and Zisserman (2016)], we carefully observe their recognition process and find that the spatial stream and the temporal stream are trained independently and tested, but only the final fusion of the scores of the two streams is finally performed. The disadvantage of this method is that one stream identification will fail, the other stream identification will succeed, and the overall network identification will fail. On the contrary, we hope that temporal stream (motion information) and spatial stream (static information) not only merge at the end, but promote each other in the process. In order to capture the interaction of spatial (static) information and temporal (motion) information, we hope that static and motion features interact, so our collaborative learning model uses motion and static information with symmetrical structural motion and static information to make static and motion features guide and optimize each other.

3.2.1 Algorithm formula

At time t, frame features are utilized to optimize optical flow features.

$$\mathrm{H} = \tan h\,(W^{o}F^{o} + (W_{V}^{s}V^{s})1^{T}) \quad (7)$$

where W^{o} and W_{V}^{s} are weight parameters. 1 is a vector (all values are 1). $F^{o} = [f_1^{o}, f_2^{o}, \dots, f_N^{o}]$ represents the temporal feature (optical flow feature), V^{s} is the video feature which is aggregated from the video frame feature at time t-1.

$$z^{o} = softmax\left(W_{h}^{oT}H\right) \quad (8)$$

$$V^{o} = \sum c_i^{o} f_i^{o} \quad (9)$$

We calculate the optical flow optimization coefficient c^o by Eq. (8), and we combine the optical flow features output by the previous model as the video feature V^o. W_h^o is also a weight parameter.

Next, we use the optical flow feature to optimize the frame features. The frame features are expressed as $F^s = [f_1^s, f_2^s, \dots, f_N^s]$. In general, the input of our module is: frames and optical flow features extracted from the previous model. The outputs are: optimized frame features F_f^s and optical flow features F_f^o.

3.2.2 Algorithm steps

Step 1. Define the optimization coefficient of the spatial (frame) feature as c^s.

Step 2. Initialize optimization coefficient.

Step 3. Using the spatial (frame) feature F^s to calculate the video feature $\sum c_i^s f_i^s$ by Eq. (9).

Step 4. Using V^s to calculate the optimization coefficient c^o of the temporal (optical flow) feature by Eqs. (7) and (8) to obtain the optimized temporal (optical flow) feature.

Step 5. Using the temporal (optical flow) feature F^o to calculate the video feature $\sum c_i^o f_i^o$ by Eq. (9).

Step 6. Using F^o to optimize the spatial (frame) features and obtain the optimization coefficient c^s on the spatial (frame) features

Step 7. Iteration, the convergence of the loss function stops.

Step 8. Stop and return the optimized frame feature $F_f^s = F^{sT} c^s$ and the optimized optical flow feature $F_f^o = F^{oT} c^o$.

3.3 Adaptive weighted learning

Since we have obtained the predicted scores (static and motion) for each stream, we can simply fuse the scores of the two streams as categories of video action. However, spatial (static) and temporal (motion) information contributes differently to different action categories. There are no obvious movements in some classes, such as “blow dry hair” and “pommel horse”. Therefore, these classes should be primarily recognized from static frames. Certain classes include obvious motion, however, motion information is significant for classifying categories, such as “diving” and “sky diving”. Finally, fusion weights of spatial and temporal streams of distinct classes are adaptively learned.

We express the network prediction score as $S_b^a = [{s_{b,1}^a}^T, {s_{b,2}^a}^T]^T \in \mathbb{R}^{2\times m}$, where a represents the a category in the dataset, and b represents the b category in the dataset. M is the number of action categories corresponding to the dataset. $s_{b,1}^a$ and $s_{b,2}^a$ represent the scores of the first stream and the second stream.

We represent the weight of the first stream of class a in the corresponding dataset as w_1^a and w_2^a as the second stream, $W_a = [w_1^a, w_2^a]$ is the two-stream fusion weight. The two-stream fusion weights for each category are learned separately by each category, we obtain the fusion weight W_a for each category by limiting $\sum_{b=1}^{2} w_{a,b} = 1, w_{a,b} > 0$.

$$\arg\max_{W_a} R_a - cN_a \tag{10}$$

Eq. (10) represents our objective function, and c is set to 5×10^{-3}, where R_a is defined as follows,

$$R_a = \sum_{b=1}^{n_a} W_a S_b^a A_a \tag{11}$$

$$N_a = \sum_{\{k=1,k\neq a\}}^{m} \sum_{b=1}^{n_k} W_a S_b^k A_a \tag{12}$$

where n_a denotes the number of all the data of the category a in the corresponding dataset, $\mathrm{A}_a = [0, \dots, 0, 1, 0, \dots, 0]^T \in \mathbb{R}^{m\times 1}$, only the a-th element is 1 and the other elements are 0 in this vector. The way to maximize R_a is to maximize the product of the column a vector of W_a and S_b^a. It also means to minimize the product of W_a and a-th column vector of S_b^k (k is not equal to a). R_a and N_a consider the relationship between the positive and negative samples of W_a, respectively, and are parameters that balance the weights of the positive and negative samples. Then, we can transform our objective function into:

$$arg \max_{W_a} W_a \left(\sum_{b=1}^{n_a} S_b^a A_a - c \sum_{\{k=1,k\neq a\}}^{m} \sum_{b=1}^{n_k} S_b^k A_a , s.t. \sum_{b=1}^{2} W_{a,b} = 1, W_{a,b} > 0\right) \tag{13}$$

Finally, our fusion weights are calculated by linear programming [Li, Liu, Wang et al. (2019); Reddy and Shah (2013)].

During the test, the SoftMax layer output of the two streams is expressed as Eq. (14).

$$S_i = [{s_{i,1}}^T, {s_{i,2}}^T]^T \in \mathbb{R}^{2\times m} \tag{14}$$

$$\arg\max_{b} W_b S_t A_a \tag{15}$$

The final classification result is determined by the highest fusion score.

4 Experiments

4.1 Datasets

We select 3 widely used action recognition datasets UCF101 [Soomro, Zamir and Shah (2012)], HMDB51 [Kuehne, Jhuang, Garrote et al. (2011)], and THUMOS14 [Idrees, Zamir, Jiang et al. (2017)] to validate our HTSC method. UCF101 is collected from real world, which are clipped from YouTube. UCF101 is a widely used action recognition dataset, including 101 different kinds of human action video. UCF101 consists of 13320 videos and 101 action categories. HMDB51 contains 6849 samples of 51 categories extracted from various resources (online videos and films). THUMOS14 is a large video dataset applied to action recognition and detection that includes long unclipped videos. THUMOS14 has 101 action classes. We adopt 13,320 videos for training and 1010 videos for verification, respectively. Then, we test the performance of our network on 1,574 videos.

(1) UCF 101 (2) HMDB51 (3) THUMOS14

Figure 3: We have selected 9 different categories of examples from each of the three datasets UCF101, HMDB51 and THUMOS14. The three datasets have 101, 51 and 101 categories respectively

4.2 Implementation details

4.2.1 Hidden two-stream model

With regard to two-stream method, our work is based on Vgg16 [Simonyan and Zisserman (2014)] model. The detailed network structure is shown in Fig. 2, and we utilize the spatial stream of the two-stream network to extract spatial feature. The convolution kernel in each convolution layer is represented as (W×H). The number K denotes the lineage of "Blocks" in Tab. 1. For spatial flow, the input size is 224×224×3 and its output feature map size is 1×1×4096. During the training process, we adopt the pre-trained Vgg16 model on ImageNet. Besides, we change the number of fully connected layer (classification) as the class number of the corresponding datasets.

Table 1: Architecture of the spatial stream network in the hidden two-stream model, and M is set to the number of corresponding dataset categories

Layers	Input size	Blocks	Output size
conv1_x	224×224×3 (224×224×20)	3×3,64 stride1	224×224×64
pool1	224×224×64	2×2 max pool stride2	112×112×64
conv2_x	112×112×64	3×3 ,128 stride 1	112×112×128
pool2	112×112×128	2×2 max pool stride 2	56×56×128
conv3_x	56×56×128	3×3,256 stride 1	56×56×256
pool3	56×56×256	2×2 max pool stride 2	28×28×256
conv4_x	28×28×256	3×3,512 stride 1	28×28×512
pool4	28×28×512	2×2 max pool stride 2	14×14×512
conv5_x	14×14×512	3×3,512 stride 1	14×14×512
pool5	14×14×512	2×2 max pool stride 2	7×7×512
fc6	7×7×512	3×3,4096 stride 1	7×7×4096
fc7	7×7×4096	3×3,4096 stride 1	1×1×4096
fc8	1×1×4096	3×3, M stride 1	1×1×M

For hidden temporal stream, the detailed structure of network is shown in Tab. 2, the input is frame sequence, and optical flow is estimated by the CNN. The optical flow is directly fed to the feature extraction network after normalization. Because the optical flow images are not stored, it is much faster than the two-stage methods. The two-stage method requires writing and reading the optical flow images and takes almost three times longer than all the other steps.

Table 2: Architecture of the hidden temporal stream network in the hidden two-stream model

Layers	Input	Blocks	Output size
conv1	224×224×33	3×3,64 str 1	224×224×64
conv1_1	224×224×64	3×3,64 str 1	224×224×64
conv2	224×224×64	3×3,128 str2	112×112×128
conv2_1	112×112×128	3×3,128 str1	112×112×128
conv3	112×112×128	3×3,256 str2	56×56×256
conv3_1	56×56×256	3×3,256 str1	56×56×256
conv4	56×56×256	3×3,512 str2	28×28×512
conv4_1	28×28×512	3×3,512 str1	28×28×512
conv5	28×28×512	3×3,512 str2	14×14 ×512
conv5_1	14×14×512	3×3,512 str1	14×14 ×512
conv6	14×14×512	3×3,1024 str2	7×7×1024
conv6_1	7×7×1024	3×3,1024 str1	7×7×1024
flow6(loss6)	7×7×1024	3×3,20 str1	7×7×20
deconv5	7×7×1024	4×4,512 str2	14×14 ×512
xconv5	14×14×1044 deconv5+flow6+conv5_1	3×3,512 str1	14×14 ×512
flow5(loss5)	14×14×512	3×3,20 str1	14×14 ×20
deconv4	14×14×512	4×4,256 str2	28×28×256
xconv4	28×28×788 deconv4+flow5+conv4_1	3×3,256 str1	28×28×256
flow4(loss4)	28×28×256	3×3,20 str1	28×28×20
deconv3	28×28×256	4×4,128 str2	56×56×128
xconv3	56×56×404 deconv3+flow4+conv3_1	3×3,128 str1	56×56×128
flow3(loss3)	56×56×128	3×3,20 str1	56×56×20
deconv2	56×56×128	4×4,64 str2	112×112×64
xconv2	112×112×212 deconv2+flow3+conv2_1	3×3,64 str1	112×112×64
flow2(loss2)	112×112×64	3×3,20 str1	112×112×20

4.2.2 Collaborative learning model

The detailed network structure of our collaborative learning module is shown in Fig. 3. The module consists of two parts: the first part is the collaborative learning layer, and the hidden two-stream model output features are used as its input. Static and motion features are optimized by collaborative learning layer. Like the hidden two-stream model, we design N hidden units in the two softmax layers, where N is the number of categories in the relative datasets. The second part is adaptive weighted learning model. The details are in Section 3. We select the cross-validation model and set the parameter c of Eq. (13) to 5×10^{-3}. And in order to optimize our collaborative learning network, we adopt the cross-entropy loss as our loss function. Finally, we predict the action category of the video by Eq. (15).

4.2.3 Comparison with the-state-of-art methods

Our HTSC method is tested on 2 trimmed video datasets and 1 untrimmed video dataset. Our experimental results are compared with latest competitive methods, and the results are shown in Tab. 3. For HMDB51 dataset, the early works [Diba, Pazandeh and Van Gool (2016); Kantorov and Laptev (2014); Gui and Zeng (2019)] selected the hand-crafted features as video feature representations, whose performance is limited and far worse than our proposed method. Some methods [Tran, Bourdev, Fergus et al. (2015); Diba, Pazandeh and Van Gool (2016)] utilize the features of 3D convolution as video representation, whose speed is fast. However, they need high computational cost and obtain lower accuracy than two-stream methods and their derivation methods. Other methods, for example, Karpathy et al. [Karpathy, Toderici, Shetty et al. (2014)] adopted two kinds of CNN to simulate static and motion information to obtain higher accuracy than conventional action recognition methods [Cai, Wang, Peng et al. (2014); Kantorov and Laptev (2014); Gui and Zeng (2019)]. But the improvements are limited because of the simple fusion strategy. Therefore, some researchers [Wang, Xiong, Wang et al. (2016); Feichtenhofer, Pinz and Zisserman (2016)] employ more complicated feature fusion strategies to combine static and motion information and obtain higher accuracy than [Karpathy, Toderici, Shetty et al. (2014)] method. But all these methods [Simonyan and Zisserman (2014); Wang, Xiong, Wang et al. (2016); Feichtenhofer, Pinz and Zisserman (2016)] need to extract optical flow in advance, which affects the efficiency of the whole network. Moreover, the spatial and temporal features of video extraction are not independent of each other. On the contrary, they have high complementarity. Our method achieved good results among the most advanced methods, with an increase of 1.3% over the highest results of the comparative method. This result occurs because our method allows the two types of information (static and motion information) of the video to learn and optimize each other. The accuracy of the method TCLSTA is slightly higher than our method because it not only utilizes the complementarity of spatial and temporal features, but also pays attention to spatial and temporal features before collaborative learning. However, the spatial-temporal attention model requires three-stage training and a deeper residual network in TCLSTA, which increases the cost of training. Compared with TCLSTA, our hidden two-stream method train the model end-to-end and implicitly extracts optical flow information, which saves storage space required by pre-extracted optical flow images. Therefore, our efficiency is higher than TCLSTA (Efficiency Evaluation section).

The comparison on UCF101 dataset is also shown in Tab. 3. The result trends for HMDB51 dataset and THUMOS14 dataset are similar to UCF101 dataset.

Table 3: Experimental results are compared with the latest technology on three datasets. The evaluation metric of the HMDB51 and UCF101 datasets is the accuracy, and THUMOS14 dataset is MAP

Methods	**UCF101**	**HMDB51**	**THUMOS14**
MVSV [Cai, Wang, Peng et al. (2014)]	0.835	0.559	--
MV+FV [Kantorov and Laptev (2014)]	0.785	0.467	--
EMV [Zhang, Wang, Wang et al. (2016)]	0.802	--	0.416
C3D (1 Net) [Tran, Bourdev, Fergus et al. (2015)]	0.823	0.497*	0.546
3DNet [Diba, Pazandeh and Van Gool (2016)]	0.902	--	--
Two-stream [Simonyan and Zisserman (2014)]	0.880	0.594	0.661
Two-Stream+Fusion [Feichtenhofer, Pinz and Zisserman (2016)]	0.92.5	0.654	--
TSN（RGB Diff）[Wang, Xiong, Wang et al. (2016)]	0.910	0.645*	0.719*
TCLSTA [Peng, Zhao and Zhang (2018)]	0.940	0.687	0.847
HTS [Zhu, Lan, Newsam et al. (2018)]	0.903	0.605	0.667
Ours	0.927	0.667	0.797

4.2.4 Ablation experiments

To prove the effectiveness of each component of the proposed method HTSC, we design the following ablation experiments, Tab. 4 shows the results of our ablation experiments.

Our method includes two streams: spatial stream and hidden temporal stream. Firstly, we use spatial stream and hidden temporal stream to predict the categories of videos, respectively. Then, we fuse spatial stream and hidden temporal stream. We find that the accuracy of two-stream fusion method is 6.2% higher than single-stream fusion method in UCF101 dataset, indicating that spatial stream and hidden temporal stream are complementary. In addition, we add collaborative learning network (CLN) on the basis of hidden two-stream network, we find that "hidden two-stream+CLN" achieves better classification accuracy than the results of hidden two-stream network without collaborative learning model. It is shown that the CLN can promote mutual learning of static and motion features and make use of their correlation to further improve the accuracy of action recognition. Finally, we add the adaptive weighted learning (AWL) model on the basis of "hidden two-stream+CLN". Compared with the network without adaptive weighted learning model, the accuracy is further improved, which proves the effectiveness of adaptive weighted learning. The accuracy of late fusion is lower than adaptive weighted learning model, the reason is that the late fusion cannot distinguish the importance of different categories of static and motion information. While the adaptive weight learning can distinguish the different significance of static and motion information of different semantic classes, which improves the accuracy of our entire network.

Table 4: Comparison of experimental results of each part of the model

Method	UCF101	HMDB51	THUMOS14
Spatial stream	0.809	0.514	0.721
Hidden temporal steam	0.841	0.545	0.673
HTS	0.903	0.605	0.746
HTS+CLN	0.919	0.645	0.783
HTS+CLN+AWL	0.927	0.667	0.797

4.2.5 Efficiency assessment

To evaluate the performance of the HTSC method, we calculate the test speed on HMDB51 dataset. The test process of the hand-crafted feature methods [Cai, Wang, Peng et al. (2014); Kantorov and Laptev (2014); Gui and Zeng (2019)] include local feature extraction, feature coding and classification. But the process of local feature extraction takes up lots of the computational cost, resulting in low efficiency. So, we do not compare our method with them. The results are compared with some existing deep learning methods in Tab. 5. The efficiency and accuracy of our method is better than Simonyan et al. [Simonyan and Zisserman (2014); Feichtenhofer, Pinz and Zisserman (2016)]. In addition, compared with the two-stream methods and their derivation methods, we do not need to extract optical flow images in advance. Our method is faster than TCLSTA, because our hidden two-stream model train the model end-to-end and implicitly extracts optical flow information, saving storage space required for pre-extracted optical flow images In general, although the efficiency is slightly lower than Zhu et al. [Zhu, Lan, Newsam et al. (2018)], our proposed method obviously achieve higher accuracy than other methods, and we show the results in Tab. 5.

Table 5: Efficiency assessment of the network in HMDB51 dataset

Method	Frames per second(fps)
Two-Stream+Fusion [Feichtenhofer, Pinz and Zisserman (2016)]	33.2
Two-stream [Simonyan and Zisserman (2014)]	99.7
TCLSTA [Peng, Zhao and Zhang (2018)]	89.5
HTS [Zhu, Lan, Newsam et al. (2018)]	120.4
Ours	115.1

5 Conclusions

This paper proposes a hidden two-stream collaborative learning network for human action recognition, which consists of a hidden two-stream model and a collaborative model. Conventional action recognition methods need to extract the optical flow of the video in advance to capture the motion information. Differently, the hidden two-stream model adopts CNN to capture the relationships between video frames, which improves the efficiency of the whole network and saves the storage space. The collaborative model adopts the hidden two-stream model to extract the spatial static frame features and the temporal flow motion features, which enhance the mutual representation to improve the accuracy of action recognition. Experiments of three widely used video classified datasets show the effectiveness of our method.

Acknowledgement: This work was supported by the Scientific Research Fund of Hunan Provincial Education Department of China (Project No. 17A007); and the Teaching Reform and Research Project of Hunan Province of China (Project No. JG1615).

Funding Statement: The author(s) received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References:

Cai, Z. W.; Wang, L. M.; Peng, X. J.; Qiao, Y. (2014): Multi-view super vector for action recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 596-603.

Carreira, J.; Zisserman, A. (2017): Quo vadis, action recognition? A new model and the kinetics dataset. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299-6308.

Chen, Y. T; Xia, R. L.; Wang, Z.; Zhang, J. M.; Yang, K. et al. (2019): The visual saliency detection algorithm research based on hierarchical principle component analysis method. *Multimed Tools Applications*, vol. 75, pp. 16943-16958.

Diba, A.; Pazandeh, A. M.; Van Gool, L. (2016): Efficient two-stream motion and appearance 3D CNNs for video classification. arXiv:1608.08851.

Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C. et al. (2015): Flownet: learning optical flow with convolutional networks. *IEEE International Conference on Computer Cision*, pp. 2758-2766.

Du, Y.; Wang, W.; Wang, L. (2015): Hierarchical recurrent neural network for skeleton based action recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1110-1118.

Feichtenhofer, C.; Pinz, A.; Zisserman, A. (2016): Convolutional two-stream network fusion for video action recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1933-1941.

Gui, Y.; Zeng, G. (2019): Joint learning of visual and spatial features for edit propagation from a single image. *Visual Computer*, pp. 1-14.

Idrees, H.; Zamir, A. R.; Jiang, Y. G.; Gorban, A.; Laptev, I. et al. (2017): The Thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, vol. 155, pp. 1-23.

Jason, J. Y.; Harley, A. W.; Derpanis, K. G. (2016): Back to basics: unsupervised learning of optical flow via brightness constancy and motion smoothness. *European Conference on Computer Vision*, pp. 3-10.

Ji, S. W; Xu, W.; Yang, M.; Yu, K. (2012): 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.

35, no. 1, pp. 221-231.

Kantorov, V.; Laptev, I. (2014): Efficient feature extraction, encoding and classification for action recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2593-2600.

Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R. et al. (2014): Large-scale video classification with convolutional neural networks. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1725-1732.

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; Serre T. (2011): Hmdb: a large video database for human motion recognition. *International Conference on Computer Vision*, pp. 2556-2563.

Lai, W. S.; Huang, J. B.; Ahuja, N.; Yang, M. H. (2017): Deep laplacian pyramid networks for fast and accurate super-resolution. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 624-632.

Laptev, I.; Marszałek, M.; Schmid, V.; Rozenfeld, B. (2008): Learning realistic human actions from movies. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8.

Li, W. J; Liu, H. Y.; Wang, J. X.; Xiang, L. Y.; Yang, Y. J. (2019): An improved linear kernel for complementary maximal strip recovery: simpler and smaller. *Theoretical Computer Science*, vol. 786, pp. 55-66.

Long, M.; Peng, F.; Li, H. Y. (2018): Separable reversible data hiding and encryption for HEVC video. *Journal of Real-Time Image Processing*, vol. 14, no. 1, pp. 171-182.

Ng, J. Y. H.; Choi, J.; Neumann, J.; Davis, L. S. (2018): Actionflownet: learning motion representation for action recognition. *IEEE Winter Conference on Applications of Computer Vision*, pp. 1616-1624.

Peng, Y. X.; Zhao, Y. Z.; Zhang, J. C. (2018): Two-stream collaborative learning with spatial-temporal attention for video classification. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 3, pp. 773-786.

Perronnin, F.; Sánchez, J.; Mensink, T. (2010): Improving the fisher kernel for large-scale image classification. *European Conference on Computer Vision*, pp. 143-156.

Qiu, Z. F.; Yao, T.; Mei, T. (2017): Learning spatio-temporal representation with pseudo-3D residual networks. *IEEE International Conference on Computer Vision*, pp. 5533-5541.

Reddy, K. K.; Shah, M. (2013): Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, vol. 24, no. 5, pp. 971-981.

Simonyan, K.; Zisserman, A. (2014): Two-stream convolutional networks for action recognition in videos. *Advances in Neural Information Processing Systems*, pp. 568-576.

Simonyan, K.; Zisserman, A. (2014): Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

Soomro, K.; Zamir, A. R.; Shah, M. (2012): Ucf101: a dataset of 101 human actions classes from videos in the wild. arXiv:1212.0402.

Tang, Q.; Yang, K.; Zhou, D. L.; Luo, Y. S. et al. (2015): A real-time dynamic pricing algorithm for smart grid with unstable energy providers and malicious users. *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 554-562.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. (2015): Learning spatiotemporal features with 3D convolutional networks. *IEEE International Conference on Computer Vision*, pp. 4489-4497.

Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y. et al. (2018): A closer look at spatiotemporal convolutions for action recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6450-6459.

Wang, H.; Schmid, C. (2013): Action recognition with improved trajectories. *IEEE International Conference on Computer Vision*, pp. 3551-3558.

Wang, J.; Gao, Y.; Yin, X.; Li, F.; Kim, H. J. (2018): An enhanced pegasis algorithm with mobile sink support for wireless sensor networks. *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1-9.

Wang, L. M.; Qiao, Y.; Tang, X. (2015): Action recognition with trajectory-pooled deep-convolutional descriptors. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4305-4314.

Wang, L. M.; Xiong, Y. J.; Wang, Z.; Qiao, Y.; Lin, D. H. et al. (2016): Temporal segment networks: towards good practices for deep action recognition. *European Conference on Computer Vision*, pp. 20-36.

Wang, Y. B.; Long, M. S.; Wang, J. M.; Yu, P. S. (2017): Spatiotemporal pyramid network for video action recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1529-1538.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; Simoncelli, E. P. (2004): Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612.

Xiang, L. Y.; Shen, X. B.; Qin, J. H.; Hao, W. (2019): Discrete multi-graph hashing for large-scale visual search. *Neural Processing Letters*, vol. 49, no. 3, pp. 1055-1069.

Ng, Y. H.; Hausknecht, M.; Vijayanarasimhan, S.; Vinyals, O.; Monga, R. et al. (2015): Beyond short snippets: deep networks for video classification. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4694-4702.

Zeng, D. J.; Dai, Y.; Li, F.; Sherratt, R. S.; Wang, J. (2018): Adversarial learning for distant supervised relation extraction. *Computers, Materials & Continua*, vol. 55, no. 1, pp. 121-136.

Zhang, B. W.; Wang, L. M; Wang, Z.; Qiao, Y.; Wang, H. L. (2016): Real-time action recognition with enhanced motion vector CNNs. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2718-2726.

Zhang, D. Y.; Yin, T.; Yang, G. B.; Xia, M.; Li, L. et al. (2017): Detecting image seam carving with low scaling ratio using multi-scale spatial and spectral entropies. *Journal of Visual Communication and Image Representation*, vol. 48, pp. 281-291.

Zhang, J. M.; Jin, X. K.; Sun, J.; Wang, J.; Sangaiah, A. K. (2018): Spatial and semantic convolutional features for robust visual object tracking. *Multimedia Tools and Applications*, pp. 1-21.

Zhu, Y.; Lan, Z. Z.; Newsam, S.; XHauptmann, S. (2018): Hidden two-stream convolutional networks for action recognition. *Asian Conference on Computer Vision*, pp. 363-378.