IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 4, 2018, accepted January 10, 2019, date of publication January 21, 2019, date of current version February 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2894025

Spatiotemporal Relation Networks for Video Action Recognition

ZHENG LIU AND HAIFENG HU, (Member, IEEE)
School of Electronic and Information Technology, Sun Yat-sen University, Guangzhou 510006, China

Corresponding author: Haifeng Hu (huhaif@mail.sysu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61673402, Grant 61273270, and Grant 60802069, in part by the Natural Science Foundation of Guangdong Province under Grant 2017A030311029, in part by the Science and Technology Program of Guangzhou of China under Grant 201704020180, and in part by the Fundamental Research Funds for the Central Universities of China.

ABSTRACT Two-stream convolutional networks have shown strong performance in a video action recognition task for its ability to capture spatial and temporal features simultaneously. However, the calculation of optical flow is time-consuming and it cannot be applied to the real-time processing of video. To address this problem, this paper proposes a new end-to-end architecture called SpatioTemporal Relation Networks (STRN) to extract spatial information and temporal information simultaneously from the video with the only RGB input. STRN consist of two branches, called appearance stream and motion stream, respectively. Appearance stream retains the structure of the original spatial stream in the two-stream architecture with the input of consecutive frames instead of a single frame. Motion stream, which takes relation information between the adjacent features in the appearance stream as an input, can effectively complement appearance stream. A relation block is an extractor which is used to extract relation information from the appearance stream. STRN can learn spatiotemporal information from the video with the only RGB input, which avoids the calculation of optical flow. We validate the STRN on UCF-101 and HMDB-51 and achieve better performance.

INDEX TERMS Action recognition, convolutional neural networks, two-stream networks.

I. INTRODUCTION

Video action recognition is of great significance for its wide range of practical applications in video surveillance, video retrieval, human computer interaction, etc.

Current methods are mainly based on deep learning, particularly Convolutional Neural Networks (CNNs) [1], since CNNs have turned out to be a sort of powerful model and achieved great success in image processing domain. However, early CNNs-based methods [2], [3] did not make satisfactory progress. The main reason may be that video is a 3D modality by nature and it contains both appearance information and motion information, while CNNs are designed for 2D images to capture appearance information. So early CNNs-based video action recognition methods failed to extract motion information from videos and were inferior to the best human-crafted method [11].

How to learn spatiotemporal information from videos is a fundamental problem in video action recognition. Both the early hand-crafted methods and current deep learning-based methods attempt to extract effective spatiotemporal features to represent video information. There are two kinds of successful architectures: two-stream networks [4] and 3D CNNs [3]. Two-stream networks divide video information into spatial and temporal information. Spatial information is extracted by a spatial stream with RGB input, while a temporal stream takes stacked optical flow which is the displacement vector of adjacent frames and calculated in advance as input, and it can be used as an extractor of temporal information. Introducing optical flow makes two-stream networks achieve the state-of-the-art performance and surpass the best hand-crafted methods at that time [11]. However, the calculation of optical flow is time-consuming and it is not suitable for real-time processing. Therefore, 3D convolution (Conv3D) is introduced to overcome this limitation. It can be used to extract spatiotemporal information with the input of video clips. However, it is hard to train a 3D CNN and the performance of 3D CNNs are inferior to two-stream architecture. So we still can't be sure whether the 3D model can efficiently learn spatiotemporal information from videos or not.

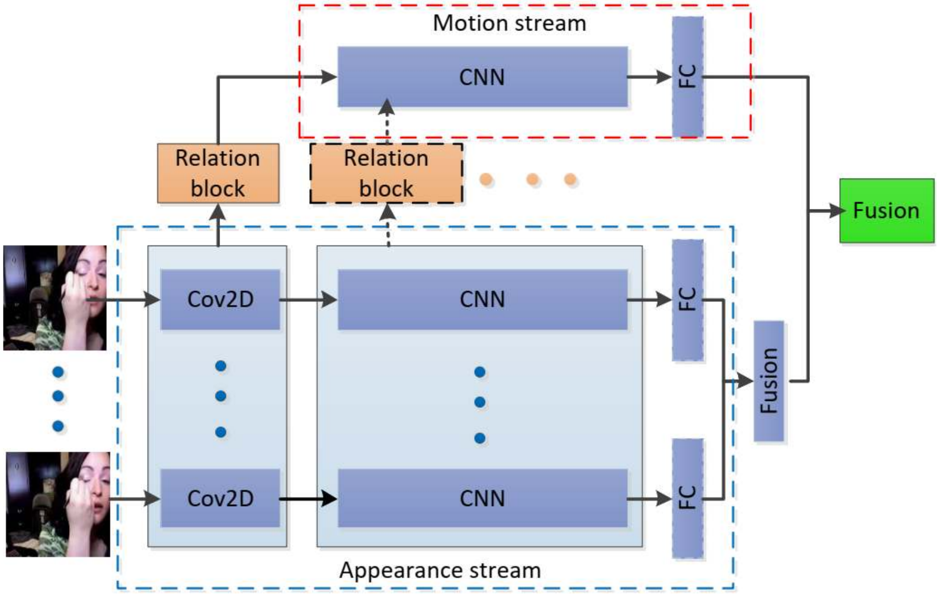

FIGURE 1. The overview of STRN. Our STRN consist of two branches, called appearance stream and motion stream. We train the whole architecture with the input of consecutive frames. A relation block is used to extract relation information from features in appearance stream and we train our motion stream with the extracted relation information. Extra relation blocks are added between appearance stream and motion stream to make a complement of relation information. We average scores of all the frames in appearance stream and finally fuse the two streams.

In this paper, a new architecture called SpatioTemporal Relation Networks (STRN) is proposed to make a better representation of spatiotemporal information. We hold the opinion that features extracted by basic 2D CNNs with the input of consecutive frames still retain the relation information among frames and can be used to represent motion information in video. So we attempt to extract relation information from appearance stream. We feed our motion networks with the input of extracted relation information. The whole structure can be seen in Fig. 1. The input of the whole architecture is consecutive frames. We extract features of each frame with a basic 2D CNN. The motion information is obtained with the input of relation information of consecutive frames which is extracted by a relation block after the first convolution layer in appearance stream. We also try to add extra relation blocks to make a complement of motion stream. Our structure retains the two branches architecture and avoid the calculation of optical flow. The relation block is designed to extract relation information between features in appearance stream, which can effectively represent the temporal information in videos but increase a little computing cost in STRN. On account that extracting relation information of adjacent frames is an important part in our networks, we explore different methods to extract relation information from features, including max pooling, average pooling and Conv3D.

Our contributions can be summarized as follows: (1) An end-to-end structure called STRN is proposed to extract appearance and motion information from videos with the input of consecutive frames only. Our networks can avoid the problem of extracting optical flow in two-stream networks and training huge parameters in 3D CNNs. (2) We find that features of adjacent frames still contain relation information, which can make a complement of appearance information and improve the performance of the whole architecture. (3) STRN are tested on two classical datasets UCF-101 [6] and HMDB-51 [7], and achieve the state-of-the-art performance.

II. RELATED WORKS

Methods for action recognition in videos can be briefly divided into two categories: hand-crafted methods and deep learning based methods.

Hand-crafted methods used spatiotemporal feature detectors for video representation. There were many different spatiotemporal feature detectors such as 3D-Harris [8], 3D-Hessian [9], Dense Trajectories [10] and Improved Dense Trajectories (IDT) [11]. They represented the video by computing a histogram descriptor such as Histogram of Gradient and Histogram of Flow (HOG/HOF) [12], Histogram of Motion Boundary (MBH) [10]. The most successful hand-crafted work was IDT, which extracted descriptors by tracing the trajectories of dense points. At that time, IDT with an SVM classifier could achieve the best result in video action recognition tasks.

Recently, with the proliferation of deep learning in computer vision, many CNNs-based action recognition methods have been proposed. Simonyan and Zisserman [4] proposed two-stream networks which were made up of a spatial stream for extracting appearance information and a temporal stream for learning motion information. Introducing optical flow greatly improved the performance of the whole architecture and made the two-stream networks achieve the state-of-the-art performance at that time. A lot of methods have been proposed based on two-stream architecture since then. Temporal Segment Networks (TSN) [13] were proposed to build a long-term model. TSN evenly divided videos into several segments and each segment adopted the structure of two-stream networks. Feichtenhofer *et al.* [14] proposed a variety of fusion methods after the convolution layer of the two-stream network to obtain spatiotemporal features. They also proposed Spatiotemporal Residual Networks [5] and Spatiotemporal Multiplier Networks [15]. It was a new way to extract spatiotemporal features by inserting residual connections between the spatial and temporal streams of two-stream architecture. There were many action recognition works on the basic of 3D CNNs which can extract spatiotemporal information from videos. Tran *et al.* [16] proposed 3D ConvNets (C3D) to extract spatiotemporal features directly. However, 3D networks were too numerous and hard to train. Qiu *et al.* [17] proposed Pseudo-3D Residual Networks. They simulated the $3 \times 3 \times 3$ convolutions with $1 \times 3 \times 3$ convolutional filters on spatial domain plus $3 \times 1 \times 1$ convolutions to construct temporal connections on adjacent feature maps in time. There were also other 3D-CNNs networks such as I3D [19] and T3D [18]. All of them attempted to directly extract spatiotemporal information from videos. Wang *et al.* [20] proposed a new architecture called Appearance-and-Relation Networks (ARTNet) which simultaneously extracted the appearance information and relation information from videos in a separate and explicit manner. Sun *et al.* [21] proposed Optical Flow guided Feature (OFF), which is derived from the definition of optical flow and used to extract temporal information from features.

The most closely related work to ours is two-stream networks, which separately extract spatial and temporal information with two branches. Our STRN retain the two branches, called appearance stream and motion stream separately. Compared with original two-stream architecture, we remove the first convolution layer in motion network and feed the motion stream with the input of relation information between adjacent frames instead of stacked optical flow. Our motion stream is designed to complement the appearance stream. Now that we can extract relation information from features in appearance streams, STRN contains spatial and temporal information simultaneously. Compared with two-stream architecture, STRN extract appearance and motion information from videos with the input of RGB only.

III. SPATIOTEMPORAL RELATION NETWORKS

In this section we will describe our STRN in detail. Firstly, we will describe the architecture of the STRN. Then, we will discuss several relation blocks for extracting relation information from consecutive frames. Finally, we will state the process of adding multiple relation blocks into the original networks to make a complement of relation information.

A. ARCHITECTURE

Original two-stream networks divide video information into spatial and temporal information. Optical flow is used as a symbol of the motion information in videos. Along with the improvement of performance, the computing time is also greatly increased. To address this problem, we propose a new two-stream networks called SpatioTemporal Relation Networks. As shown in Fig. 1, STRN consist of two branches, called appearance stream and motion stream respectively. Appearance stream is used to extract spatial information with the input of consecutive frames. Motion stream learns the motion information with the input of relation information extracted by a relation block. Relation information between adjacent frames is used as a representation of temporal information in videos. We attempt to learn spatial and temporal information simultaneously with the input of RGB only.

1) APPEARANCE STREAM

Appearance information is a vital cue for action recognition since it contains object and scene information. In our architecture, we use a 2D CNN to extract appearance information from videos. Instead of training the CNN with only a randomly selected frame, we feed our stream with several consecutive frames and average scores of all frames as the last result. Although the length of input frames increases, there is little gain in appearance stream since spatial information extracted from adjacent frames is similar to each other. However, consecutive frames can be used to extract motion information from videos which can complement appearance information.

2) MOTION STREAM

Our motion stream extracts motion information with the input of the relation of consecutive frames. Unlike two-stream architecture which uses optical flow as the representation of motion information, our motion stream extracts information from relation information of adjacent frames. We think that features of adjacent frames remain relation information between frames. So, several relation blocks are designed to extract relation information from adjacent frames, including max pooling, average pooling and Conv3D. A 2D CNN is used to extract motion information with the input of extracted relation information.

B. RELATION BLOCK

We train our motion stream with the relation of consecutive frames which contains motion cues. To extract efficient motion information, learning relation information is an important part in our networks. We design three different methods to extract relation information of adjacent frames, including average pooling, max pooling and Conv3D. All those methods are operated in time domain of appearance features. Suppose the length of input frames is T, our relation block can be represented as $f : x_a \rightarrow x_m$, where $x_m \in \mathbb{R}^{C \times W \times H}$ is the input of motion stream and $x_a \in \mathbb{R}^{T \times C \times W \times H}$

is the features of consecutive frames after the first convolution layer in appearance stream.

1) AVERAGE POOLING

We calculate the average of features of consecutive frames at the same spatial location and channels. Average pooling can be formulated as follows:

$$x_m = \frac{1}{T}\sum_{i=1}^{T} x_a^i \tag{1}$$

where $x_a^i \in \mathbb{R}^{C\times W\times H}$ stands for feature of the i-th frame. Since we should extract relation information from T frames, average pooling can simply define an arbitrary correspondence among all temporal channels and allow a flexible length of consecutive frames.

2) MAX POOLING

Like average pooling, we take the maximum of all features map at the same spatial location and channels. Our max pooling can be formulated as follows:

$$x_m^{(c,w,h)} = max\{x_{a_i}^{(c,w,h)} | i \in [1, T]\} \tag{2}$$

where $1 \leq c \leq C$, $1 \leq w \leq W$ and $1 \leq h \leq H$. Max pooling can also make a correspondence among all temporal channels.

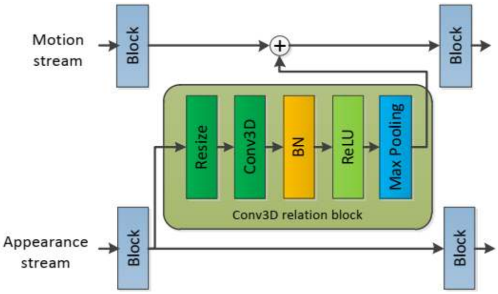

FIGURE 2. **The architecture of Conv3D relation block. We insert the relation block between two convolution block. Before Conv3D, we resize the features in appearance stream to $x_a \in \mathbb{R}^{C\times T\times W\times H}$. The kernel size of 3D max pooling is $T \times 1 \times 1$ and the kernel size of Conv3D is $3 \times 1 \times 1$ with the stride of 1.**

3) Conv3D

We apply a Conv3D to learn relation information between consecutive frames followed by a BN layer, a ReLU and 3D max pooling. Our Conv3D can be formulated as follows:

$$x_m = F(x_a, W) \tag{3}$$

W stands for the learnable parameters in the Conv3D. Before Conv3D, we should resize the x_a from $B \times C \times W \times H$ to $B' \times T \times C \times W \times H$ (B means the batch size). Since we should integrate all temporal channels, the kernel size of 3D max pooling layer should be appropriate and the length of frames is fixed to be consistent with kernel size. We described Conv3D relation block in detail in Fig. 2. Compared with average pooling and max pooling, Conv3D is learnable with a little cost, which may improve the performance of the relation block.

C. MULTIPLE RELATION BLOCKS

We have discussed several relation blocks for extracting relation information from features of consecutive frames after the first convolution layer in the above section. Extracted relation information is used to train our motion stream. Since features of each frame remain independent at every layer in appearance stream, relation information of appearance features can be extracted at every layer. So we apply our relation block at the remaining layers to make it complementary to motion stream. Relation blocks are added at different layers to see whether introducing extra connection will make an improvement in our networks or not. The additional relation block can be formulated as follows:

$$x_m^l = F(x_m^{l-1}, W) + f(x_a^l) \tag{4}$$

x_m^l is the input of the l-th layer in motion stream, x_a^l is the input of the l-th layer in appearance stream, f represents the mapping of relation block and F is a nonlinear residual mapping represented by convolutional filter weights W. Additional relation block is added as a residual part in the motion stream.

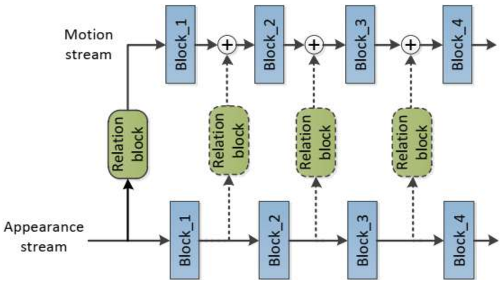

FIGURE 3. **The architecture of adding additional relation block on spatiotemporal relation networks. Our base network is a ResNet-50. We apply Conv3D to extract relation information from features on appearance stream. Our relation block can be inserted into every layer in appearance stream and we select to add the relation block after residual blocks.**

D. AN EXAMPLE OF STRN ON ResNet-50

To demonstrate STRN better, we implement our approach with the baseline network of ResNet-50, which is illustrated in Fig. 3. We insert additional relation block to extract relation information which is added as a residual into motion stream at the same location. Our relation block can be inserted at every layer in appearance stream to extract relation information. Relation blocks are inserted between residual blocks without changing the structure of each residual block. We do not try to insert our relation block after the residual block_4, since it is difficult to extract relation information from high-level semantic information.

E. DISCUSSION

In this paper, we propose a new end-to-end network for action recognition, called STRN. STRN consist of two branches, called appearance stream and motion stream respectively. Appearance stream is used to extract appearance information from videos with the input of consecutive frames. Motion stream are designed to extract temporal information with the input of relation information extracted from appearance stream. There are some works that are similar to ours such as Spatiotemporal Residual Networks [5] and Optical Flow guided Feature (OFF) [21]. Spatiotemporal Residual Networks insert residual connections between the spatial and temporal streams of two-stream architecture to extract spatiotemporal features. OFF calculate temporal information from two adjacent frames according to the calculation of optical flow. It should be noted that our STRN are quite different from them. Our STRN differ from Spatiotemporal Residual Networks mainly in the following two aspects. First, we extract relation information from appearance stream as a representation of temporal information instead of calculating optical flow. Second, we introduce extra connection between appearance stream and motion stream to complement relation information. We do not use residual connection like Spatiotemporal Residual Networks but just make direct additive fusion of the two streams. Compared with OFF, first, we use relation block to extract relation information while OFF calculate temporal information in terms of the calculation of calculating optical flow. Our relation block is more flexible and convenient. Second, we try to capture relation information from multiply frames while OFF extract temporal information only from two adjacent frames.

IV. EXPERIMENT

In this section experimental results are described. First, we introduce datasets and experiment settings. Then, we test the performance of several relation blocks to get the most effective ones to extract relation information between frames. Right after that, we study the influence of segment number and the input length of our networks. Then, we make some exploration about adding additional relation block after different layers in the appearance stream. We also test our networks with the RGB differences input. Finally, we test the efficiency of our networks and make a comparison between our method with the state-of-the-art approaches.

A. DETAILS

1) DATASETS

We validate our architecture in two standard video datasets: UCF-101 and HMDB-51. UCF-101 consists of 13,320 action videos in 101 categories. It is rich in the aspects of action types, variation of background, camera motion and viewpoint changing. The HMDB-51 dataset contains 6,766 videos in 51 action categories and it is more challenging than UCF-101 due to its complex environment. For both datasets, we report the average accuracy over the three splits of both datasets.

2) IMPLEMENTATION DETAILS

ResNet-50 is used as our underlying network. Firstly, we separately train spatial stream and temporal stream following with two-stream architecture. We use the pre-trained model to initialize our STRN during the testing phase of UCF-101. During the testing phase of HMDB-51, the former model pre-trained on UCF-101 is used to initialize our networks. The initialization is very important since two streams are jointly trained, during which one stream will dominate the whole architecture. When we initialize our two streams with the pre-trained parameters, it can address this problem to some extent.

We set the length of input frames to L. To learn a long-term motion information, each video is divided evenly into K segments similar to TSN [13]. For every segment, we train our model with L continuous frames. The input size of one frame is 224×224. Stochastic Gradient Descent (SGD) is used with a batch size of 64 samples and the procedure of batch normalization uses a smaller batch size of 4 to fit the GPU memory. Starting with a learning rate of 0.001 and it will reduce to 1/10 of its own when reaching the 40-th epoch and 60-th epoch while the total epoch is 80. During training phase, we just apply group random clipping and group random horizontal flipping for data augmentation. During testing phase, except for data augmentation used in training, we divide videos evenly into 25 segments to extract enough spatiotemporal information. For better convergence of the networks, gradient clipping is adopted during training phase and the threshold is set to 20. We average scores of appearance stream and then fuse the two branches.

TABLE 1. Classification accuracy of our STRN on different relation blocks. We set the segment number to 3 and RGB input length to 5. All models are trained and tested on split 1 of UCF-101.

Relation block	Appearance	Motion	Fusion
Average pooling	85.1%	86.7%	87.6%
Max pooling	85.5%	87.8%	88.0%
Conv3d	84.6%	87.7%	88.2%
TSN Spatial Network	85.0%		

B. EXPERIMENTAL RESULTS

1) EFFECT OF RELATION BLOCK

We have proposed three different relation blocks, including average pooling, max pooling and Conv3D. In this part, we will test the effect of different relation blocks in UCF-101 dataset. To show the performance of our network structure, a TSN spatial network is used as the baseline. We apply the same experimental setting in our spatiotemporal relation block and TSN spatial networks. During training phase, we set the segment number to 3 and the RGB input size to 5. The results shown in Table 1 reveal that three relation blocks have a significant improvement compared with TSN spatial networks. Conv3D improves the classification accuracy by 3.2%, followed by max pooling which improves 3%. Average pooling still outperforms TSN spatial networks by 2.6%. The results also reveal that motion stream can complement appearance stream, which proves our conjecture that features of adjacent frames still contain relation information and can complement appearance information. We also find that Conv3D can extract more effective relation information from consecutive frames compared with other two relation blocks. This could be that Conv3D is learnable. We can further find that the motion stream is always superior to appearance stream. It could be that our relation information is extracted from features among consecutive frames in appearance stream which contains spatial and temporal information simultaneously.

Our motion stream gains a little improvement after fused with appearance stream. That meas our relation information extracted from features in appearance stream can complement the appearance information. However, the temporal stream in two-stream architecture gains a significant improvement after fusion. We think there are two main reasons. First, our motion stream extract temporal information with the relation information, note that the relation information extracted from appearance stream contains a minimal amount of appearance information while there is great difference between optical flow and RGB. Second, we train appearance stream and motion stream jointly so that it will make one stream dominate the entire architecture causing an inferior result.

2) EFFECT OF SEGMENT NUMBER

We test the effect of segment number on our networks. Different segment numbers are tested to see the effect of long-term architecture. We set the segment number to 1 compared with that of 3. The results can be seen in Table 2. We find that both fusion methods gain some improvement with a higher segment number. As the number of segment increases, Cov3D has a 1.17% boost and average pooling improves by 2.65%. The results show that our model can benefit from high segment number. We think that with the segment number increasing, we can get a more efficient long-term model and extract more spatiotemporal information from videos.

TABLE 2. Classification accuracy of our STRN on different segment numbers and relation blocks. K means the segment number of our setting and we set the RGB input length to 5. All models are trained and tested on split 1 of UCF-101.

K	Relation block	Appearance	Motion	Fusion
K=1	Average pooling	81.95%	81.10%	84.93%
K=1	Conv3d	84.11 %	85.04%	87.07%
K=3	Average pooling	85.14%	86.65%	87.58%
K=3	Conv3d	84.59%	87.73%	88.24%

TABLE 3. Classification accuracy of our STRN on different data lengths. We set the segment number to 1. All models are trained and tested on split 1 of UCF-101.

L	Appearance	Motion	Fusion
3	83.74%	84.08%	85.91%
5	84.11%	85.04%	87.07%
7	83.90%	85.22%	87.31%
9	84.46%	85.46%	87.60%
11	84.32%	86.20%	87.60%

3) EFFECT OF INPUT DATA LENGTH

We also explore the influence of input data length in our networks. The results in Table 3 reveal that with data length increasing, the performance of our networks improves a lot. Compared with a data length of 3, there is a 1.7% increase when the data length is 9. When we further increase the data length, there is almost no increase in our networks while more GPU memory is required. So, we set the data length to 5 to coordinate the performance and GPU memory.

4) EFFECT OF MULTIPLE RELATION BLOCK

We have tested our three relation blocks in the above section and find that Conv3D is the best method to extract relation information from consecutive frames. Since our appearance stream always maintains the independence of each frame, we can extract relation information from every layer of appearance stream to complement relation information. Our experimental settings retain unchanged and ResNet-50 is used as the baseline CNN which contains one convolution layer, 4 residual blocks and one fully connected layer. To explore the effect of our relation block on different layers, we insert relation block after each residual block gradually.

The results can be seen in Table 4. Our STRN achieve the best result with the accuracy of 90.7% when additional relation blocks are added after the first residual block. However, when we choose to add relation block after the rest residual blocks, the performance of networks falls sharply. The results drop by 1.5% after adding relation blocks on second residual block and 3.2% on second and third residual block. We think that features at the top of the network are semantic information and similar to each other, so it is hard to extract relation information from those features. Meanwhile, the additional connection will make motion stream dominate the whole architecture, causing a worse result. This is also the reason that motion stream gets little gain after fusion. In conclusion, adding relation blocks at the bottom layer of the network can improve the performance.

TABLE 4. Classification accuracy of our STRN on UCF-101. We add the Conv3D relation block after location and gradually add the number of relation block as shown in the table. All models are trained and tested on split 1 of UCF-101.

Location	Appearance	Motion	Fusion
baseline	84.59%	87.73%	88.24%
block_1	84.70%	90.67%	90.71%
block_1, block_2	84.54%	89.06%	89.27%
block_1, block_2,block_3	84.56%	87.31%	87.51%

TABLE 5. Classification accuracy of our STRN with RGB differences input on the two datasets. We set the segment number to 3 and add extra relation block after block_1. All models are trained and tested on split 1.

datasets	RGB	RGB differences	Fusion
UCF-101	90.7%	91.5%	92.8%
HMDB-51	57.8%	64.1%	65.2%

5) RESULTS OF RGB DIFFERENCES INPUT

Even though our STRN have obtained a good performance with RGB input, we test our model with RGB differences input to make a complement of our networks. RGB differences, which contains some time information and is easy to calculate, is the stacked differences of RGB pixel intensities between consecutive frames. The results in Table 5 reveals that our networks perform well with the RGB differences input. After fusion with RGB differences input, our networks

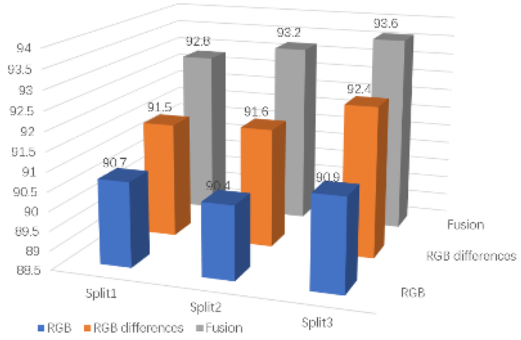

FIGURE 4. The results of our networks on the all splits of UCF-101. We set the segment numbers to 3 and insert extra relation block after block_1.

TABLE 6. Efficiency of STRN and other state-of-the-art models. All models are trained and tested on all splits of UCF-101.

Method	Speed (frames per second (fps))	Accuracy
TSN (RGB)	680	85.5%
TSN (Flow)	14	87.9%
TSN (RGB+Flow)	14	94.0%
I3D (RGB+Flow)	<14	93.4%
STRN (RGB)	410	90.7%
STRN (RGB difference)	410	91.8%

obtain an improvement by 2.1% on the UCF-101 datasets and 7.4% on the HMDB-51 datasets. The results of STRN on the all splits of UCF-101 can be seen in Fig. 4. We can find that fusing RGB differences input can achieve a better performance.

C. EFFICIENCY OF STRN

In order to evaluate the efficiency of our model, we make a comparison between STRN and other state-of-the-art methods. The results are shown in Table 6. TSN (RGB) means that the network structure used is TSN with the RGB input. The results in Table 6 reveals that compared with TSN (RGB), STRN (RGB) improve the accuracy by 5.2% at the expense of a small amount of computing speed. Besides, optical flow can seriously slow down the speed of network. When we introduce optical flow into TSN, the entire network runs at only 14 fps which is not acceptable for real-time video processing. Meanwhile, Our STRN (RGB differences) run over 410 fps with accuracy 91.8% which is efficient and effective.

D. COMPARISON WITH THE STATE-OF-THE-ART

In this part, we make a comparison between our spatiotemporal relation networks with the start-of-the-art approaches on the UCF-101 dataset and HMDB-51 dataset. We select the result of adding an additional relation block after the block_1 with the input of RGB and RGB differences.

The results can be seen in Table 7. First, we compare the performance of TSN spatial network (per-trained on ImageNet) and STRN. We can find that our STRN outperform TSN by 6.8% on the UCF-101 dataset and 11.2% on the HMDB-51 dataset. The excellent performance of our network structure proves our hypothesis that features of adjacent frames still contain relation information and can complement the appearance steam. Introducing relation information can improve the performance of the whole architecture. Then, we compare STRN with several 3D CNNs, including I3D, C3D and P3D. STRN outperform C3D (pre-trained on Sports-1M) by 7.4% on the UCF-101 dataset and 10% on the HMDB-51 dataset. The better performance reveals that our STRN are more prominent in the extraction of spatiotemporal information than C3D. STRN has a 8.7% improvement over I3D and 4.6% over P3D on the UCF-101 dataset. Our networks still outperform C3D (pre-trained on Kinetecs) by 3.4% and TSN spatial network (pre-trained on Kinetecs) by 2.1% on the UCF-101 datasets. Finally, we also compare our networks with many other networks, the details can be seen in Table 7. The results reveal that STRN can achieve comparable performance on the UCF-101 and HMDB-51 datasets. We can draw the conclusion that STRN can learn efficient spatiotemporal information with the only RGB input.

TABLE 7. Comparison with the state-of-the-art methods on the UCF-101 and HMDB-51 datasets. Our STRN are pre-trained on the ImageNet dataset and they achieve a comparable performance to other methods pre-trained on the Sport-1M and Kinetics datasets with only RGB input. Segment number is set to 3 and data length is set to 5. An additional relation block is added after block_1.

Architecture	UCF-101	HMDB-51
Two-Stream [4]	88.0%	59.4%
Spatial Stream ResNet [5]	82.3%	43.4%
Spatial TDD [22]	82.8%	50%
RGB-I3D [19]	84.5%	49.8%
TSN spatial network [13]	86.4%	53.7%
C3D [16]	85.8%	54.9%
LTC [23]	82.4%	48.7%
P3D [17]	88.6%	-
C3D [20] (with Kinetics)	89.8%	62.1%
TSN spatial network [13](with Kinetics)	91.1%	-
Our (RGB)	90.7%	57.8%
Our (RGB+RGB differences)	93.2%	64.9%

V. CONCLUSION

In this paper, we propose a new end-to-end two-stream networks, called STRN. The STRN consist of two branches, called appearance stream and motion stream respectively. We hold the opinion that features of adjacent frames also retain the relation information which can be the representation of motion information. We extract relation information between appearance features with relation block. Experimental results show that features at the bottom layer retain rich relation information. Our STRN can extract efficient spatiotemporal information from videos and achieves comparable performance on UCF-101 and HMDB-51.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[2] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.

[3] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[4] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[5] C. Feichtenhofer, A. Pinz, and R. Wildes, "Spatiotemporal residual networks for video action recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3468–3476.

[6] K. Soomro, A. R. Zamir, and M. Shah. (2012). "UCF101: A dataset of 101 human actions classes from videos in the wild." [Online]. Available: https://arxiv.org/abs/1212.0402

[7] H. Kuehne, H. Jhuang, and R. Stiefelhagen, T. Serre, "HMDB51: A large video database for human motion recognition," in *High Performance Computing in Science and Engineering*. Berlin, Germany: Springer, 2013, pp. 571–582.

[8] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, nos. 2–3, pp. 107–123, 2005.

[9] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2008, pp. 650–663.

[10] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 3169–3176.

[11] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3551–3558.

[12] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[13] L. Wang *et al.*, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 20–36.

[14] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1933–1941.

[15] C. Feichtenhofer, A. Pinz, and R. P. Wildes, "Spatiotemporal multiplier networks for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7445–7454.

[16] D. Tran, L. Bourdev, and R. Fergus, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4489–4497.

[17] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3D residual networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5534–5542.

[18] A. Diba, M. Fayyaz, and V. Sharma. (2017). "Temporal 3D ConvNets: New architecture and transfer learning for video classification." [Online]. Available: https://arxiv.org/abs/1711.08200

[19] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4724–4733.

[20] L. Wang, W. Li, W. Li, and L. Van Gool, "Appearance-and-relation networks for video classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1430–1439.

[21] S. Sun, Z. Kuang, L. Sheng, W. Ouyang, and W. Zhang, "Optical flow guided feature: A fast and robust motion representation for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1390–1399.

[22] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4305–4314.

[23] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1510–1517, Jun. 2018.

ZHENG LIU received the B.E. degree in communication engineering from Sun Yat-sen University, Guangzhou, China, in 2018, where he is currently pursuing the M.E. degree in electronics and communication engineering with the School of Electronics and Communication Engineering. His current research interests include computer vision and action recognition.

HAIFENG HU received the Ph.D. degree from Sun Yat-sen University, in 2004, where he has been an Associate Professor with the School of Information Science and Technology, since 2009. He has been publishing over 80 papers, since 2000. His research interests include computer vision, pattern recognition, image processing, and neural computation.