IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 28, 2019, accepted May 19, 2019, date of publication May 24, 2019, date of current version June 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2918808

Two-Stream Convolutional Network for Improving Activity Recognition Using Convolutional Long Short-Term Memory Networks

W. YE[1], J. CHENG[1], F. YANG[2], AND Y. XU[1]
[1]School of Physics and Technology, Nanjing Normal University, Nanjing 210023, China
[2]School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China

Corresponding author: Y. Xu (xuyinlin@njnu.edu.cn)

This work was supported by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under Grant 1812000006474.

ABSTRACT The two-stream convolutional network (ConvNet) plays a vital role in the development of the deep learning network for activity recognition. Recently, there are many studies about activity recognition using the two-stream network as a powerful feature extractor. The combination of two-stream ConvNet and fully connected long short-term memory (FC-LSTM) and the combination of two-stream ConvNet and temporal segment LSTM had achieved the best performance for activity recognition. In this paper, we are motivated to explore the performance's limit of networks that combine two-stream and recurrent neural network, so we highlight the necessity of maintaining spatial structure throughout the deep learning networks when the sequential data show correlations in space and stress the importance of appropriate fusion method when integrating feature maps and we demonstrate with experiments that these methods work well. Three main contributions can be concluded from our work. First, we propose to combine convolutional LSTM (ConvLSTM) networks with a two-stream ConvNet based on RGB streams and optical streams first. The spatiotemporal features are extracted by a two-stream ConvNet which is pre-trained on the dataset of ImageNet, and then the fused sequential three-dimensional feature maps are classified by the ConvLSTM. Second, we explored the effect of fusing the feature maps of the two-stream network at different layers with different fusing strategy and conclude that appropriate fusing location and fusing method can improve our model to the state-of-art performance. Third, we demonstrated that better overall performance can be achieved, given proper care to the ConvLSTM. Our analysis shows that our proposed network structure can achieve the state-of-art 69.4% accuracy on HMDB51 and 93.9 % accuracy on UCF101 among the methods composed by the ConvNets with the recurrent neural network without pre-training on Kinetics dataset.

INDEX TERMS Activity recognition, convolutional long short-term memory networks, convolutional neural network, two-stream.

I. INTRODUCTION

Activity recognition has been studied for many years as a challenging research task in the field of computer vision. Since two-stream ConvNet (convolutional network) was proposed by Donahue *et al.* [1], many studies have been carried on with this method. Simonyan and Zissermanused two independent convolutional networks to extract the feature maps of RGB images and multi-frame optical flow images. These feature maps not only contain spatial information but also temporal information. Their final prediction of their networks are the fusion of two-stream networks' output scores. Ma *et al.* [2] pointed out that two-stream ConvNets are unable to exploit the most critical component in action recognition because they ignore the intrinsic spatiotemporal links across spatial and temporal streams. For this reason, a series of network structures [2]–[4] that utilize two-stream ConvNet + RNN (recurrent neural network) have been prospered.

Most of the two-stream ConvNet+RNN structures [2], [5] fuse the output of the fully connected layer of ConvNets and then feed them into RNN part. However, the operation of feeding the RNN part with the output of the fully connected layer of ConvNets will turn the 3D (three-dimensional)

The associate editor coordinating the review of this manuscript and approving it for publication was Mu-Yen Chen.

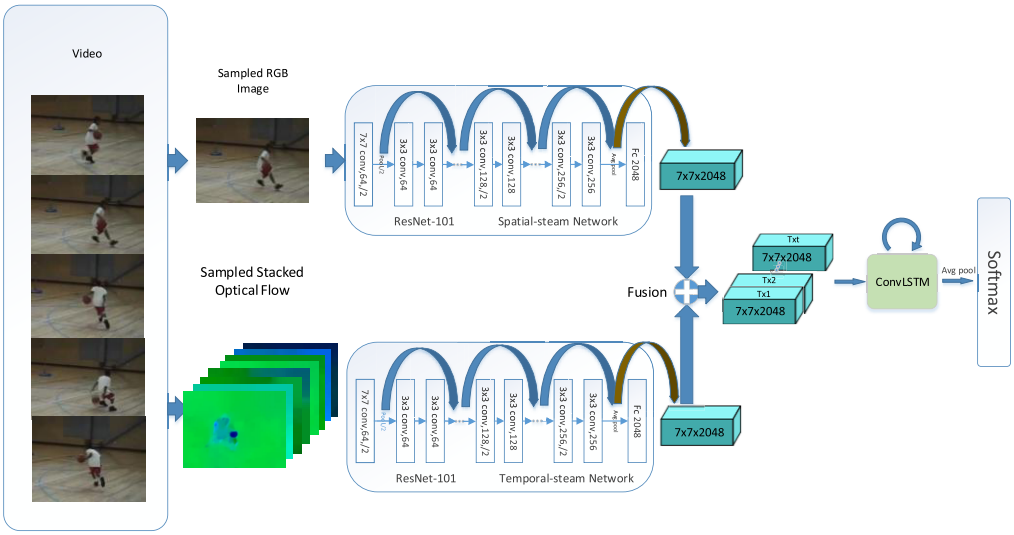


FIGURE 1. Overview of the proposed framework.

feature maps into one-dimensional vectors. Although the number of parameters of the feature map is largely diminished, the damaging of the spatial information is inevitably brought about due to this operation [6]. Besides, Xingjian *et al*. [7] extended LSTM to 3D and proposed ConvLSTM, a structure that has been proved to be better in the real-time precipitation prediction task than the FC-LSTM in their paper. Therefore, aiming to cope with this problem and to see how far two-stream network can go, we combine two-stream ConvNets and ConvLSTM and feed the fused output of the convolution layer ahead of the fully connected layer to ConvLSTM, which means when we extract features with two-stream ConvNet, the spatial structure of feature maps is kept as well. In this way, the spatial correlation information in RNN forward propagation is retained as much as possible.

In this paper, we propose a model based on two-stream ConvNet and ConvLSTM for activity recognition, as illustrated in Fig 1. First, we feed RGB images and stacked optical flow images into spatial-stream network and temporal-stream network respectively to fine-tune this two ConvNets that have been pre-trained on ImageNet. Subsequently, the feature maps output by the spatial-stream network and temporal-stream network are fused in their channel dimension. Finally, ConvLSTM is deployed to learn long-term spatiotemporal dependencies further.

The rest of the article is organized as follows: A brief review about the relevant work of activity recognition is provided in section 2. The proposed network architecture for dealing with the concerned problem is discussed in section 3. Section 4 presents the obtained experimental results and the conclusions and discussion are given in section 5.

II. RELATED WORK

Although the hand-crafted feature such as the IDT (Improved Dense Trajectories) [8] descriptor, SIFT-3D (three-dimensional scale-invariant feature transform) [9] *et al*. is elaborately constructed and can get good performance for activity recognition, but with the growing capacity of CNN to express general problems, deep learning networks for activity recognition are gradually occupying the dominant position in this field. The two-stream method based on RGB stream and optical stream performs very well in many motion recognition solutions. In recent years, many studies have introduced optical stream to be a complement of raw RGB frames and have achieved considerable improvement in performance.

The optical flow usually plays a ''black box'' role to help activity recognition methods get state-of-art performance, and we can conclude that Optical Flow can help deep learning architecture improve performance with many experimental instances such as TSN (Temporal Segment Networks) [4], Two-Stream I3D (Inflated 3D ConvNet) [10], Convolutional Two-stream + IDT [11], However, there is no clear answer to the question that why optical flow is so useful in these studies. Many researchers intuitively considered that the temporal information hidding in optical flow is responsible for the success of optical flow until Laura *et al*. [12] demonstrates with a large number of experiments that it is the invariance to appearance of the representation that entitles the optical flow such prevalence in the task of activity recognition.

In order to better capture the spatiotemporal features, Ji *et al*. [13] proposed 3D convolutional network to stack 2D convolutional feature maps from consecutive frames into 3D expecting to capture spatiotemporal information hiding in videos. Karpathy *et al*. [14] exploited a two-stream network base on Raw RGB stream and fovea stream which named as Multiresolution CNN architecture. Karpathy *et al*. also show that different kinds of ways to fuse features can affect the accuracy and the number of parameters to be learned. Instead of using 2D convolutional kernels and stacking feature maps as Karpathy *et al*. [14] and Ji *et al*. [13], Tran *et al*. [15] invited

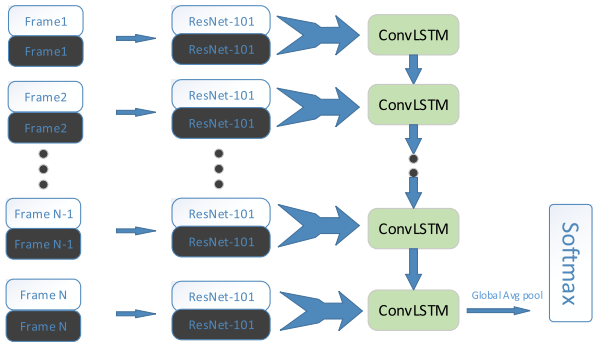


FIGURE 2. Basic block diagram showing the proposed method.

a new structure C3D (3D Convolutional Networks) which based on 3D convolutional kernels to help them boost the performance. Furthermore, Carreira and Zisserman [10] developed Two-Stream I3D, and achieve the best performance in the field of 3D ConvNets, using the ConvNets pre-trained on the dataset of Kinetics and ImageNet.

Another branch to handle this task is ConvNets with RNN. Considering the capability to encode state and capture temporal ordering, RNN can better satisfy the need of activity recognition, and especially the LSTM structure are able to capture long and short range dependencies from input data comparing 3D ConvNet which is considered more suitable for learning short-term spatiotemporal dependencies. Donahue *et al*. [16] and Srivastava *et al*. [17] use an architecture consisting of a single CNN and a FC-LSTM for activity recognition. Inspired by the research of Wang *et al*. [4], Ma *et al*. [2] adapt the temporal segment method and deploy it on LSTM layer, which achieved considerable improvement comparing traditional structure of ConvNet with LSTM.

As mentioned in Section 1, ConvLSTM can suit applications related to spatiotemporal data such as videos because the fully-connected gates in LSTM are replaced with convolutional gates in ConvLSTM. Xingjian *et al*. [7] proposed ConvLSTM to address the issue of precipitation nowcasting prediction from radar images. After that, many studies including video autoencoding [18] and anomaly detection [19]. Kim *et al.* [20] also demonstrate ConvLSTM is a good choice to deal with tasks that spatial and temporal information both matter.

For these reasons above, we propose to use ConvLSTM as one of the important blocks in the proposed model as shown in the Fig 1 for activity recognition.

III. METHODOLOGY

Fig.1 and Fig.2 illustrate the basic structure of our proposed approach. The proposed deep architecture is composed of four main steps: input preprocessing, two-stream CNN propagation, feature map fusion, and ConvLSTM propagation.

A. PREPROCESSING

Before being fed into the temporal network, the optical flow will be calculated from the raw RGB frames. After the calculation, the spatial stream network accepts raw video frames while the temporal stream network gets optical flow frames as input. There are two common kinds of algorithm for optical flow extracting—Brox [21] and TV-L1(total variation-L1) [22]. We follow the study of Ma *et al.* [2], which shows the TV-L1 algorithms is slightly better than Brox. We use the same method like [1], [2], [5], [10], and stack ten two-channels optical flow images into a new frame with 20 channels using the TV-L1 method. Besides, we use a linear transformation to rescale the horizontal and vertical components of the optical flow to the range [0, 255] which is the same as the value range of RGB data. The reason we do so is that the extracted feature maps from temporal and spatial networks will be fused. Without doing this step, severe overfitting would be introduced. In order to get data with timing information from optical flow frames and RGB frames, we sample each video at the same intervals, with sampling 25 frames.

B. TWO-STREAM CNN PROPAGATION

As mentioned above, two-stream CNN is composed of two individual spatial-stream and temporal-stream ConvNets. Therefore, we used ResNet-101 that has bigger model capacity than relatively shallow ConvNet such as VGG-16 [23] and GoogLeNet [24] to extract high-dimensional feature maps. The output feature maps at time step *t*from the spatial-stream and temporal-stream ConvNets can be represented as $f_t^s \in R^{w_S \times h_S \times c_S}$ and $f_t^T \in R^{w_T \times h_T \times c_T}$ respectively. Note that $w_S \times h_S \times c_S$ and $w_T \times h_T \times c_T$ are both dimension of $7 \times 7 \times 2048$. Using pre-trained models is an effective way to help ConvNet be equipped with the capability to learn and extract basic image features, which works well on dataset that does not have enough training samples. For the spatial-stream ConvNet, the ConvNet is pre-trained on ImageNet and fine-tuned on RGB images extracted from UCF101 dataset with classification loss for predicting activities. For the temporal-stream ConvNet, since we have discretized the optical flow fields into the interval from 0 to 255 by a linear transformation as mentioned in Section 1, it makes sense that we use optical flow frames to fine-tune the temporal-stream ConvNet whose initial parameters exactly the same as the spatial-stream ConvNet except the first convolutional layer.

It is worth mentioning that the main difference between the structure of temporal-stream ConvNet and the structure of spatial-stream ConvNet is in their first convolutional layer because for spatial-stream ConvNet, the input is RGB images which have 3 channels, but for temporal-stream ConvNet, the input is 10-stacked optical flow images. Based on this ground, we follow the procedure of Wang *et al*. [4] where they average the weights across the RGB channels and replicate this averaged weights to every channel of the first convolutional layer of temporal network.

C. FEATURE MAP FUSION

Feichtenhofer *et al*. [11] demonstrated that different methods to fuse the feature maps from two stream networks and the

TABLE 1. Performance comparison on the split 1 of HMDB51 for Fusion layers in the ResNet101.

Fusion layers	Best Accuracy(%) (method used)	#Layers in ResNet101	#Parameters
First Layer in the second block of Conv3_x	43.9(*)	15	2508800
Second Layer in the second block of Conv3_x	44.1(*)	16	2508800
First Layer in the third block of Conv3_x	44.0(*)	18	2508800
Second Layer in the third block of Conv3_x	47.4(*)	19	2508800
First Layer in the 4th block of Conv3_x	43.4(*)	21	2508800
Second Layer in the 4th block of Conv3_x	45.2(+)	22	2508800
First Layer in the 22th block of Conv4_x	60.9(+)	87	1254400
Second Layer in the 22th block of Conv4_x	63.3(*)	88	1254400
Third Layer in the 22th block of Conv4_x	65.3(+)	89	5017600
First Layer in the 23th block of Conv4_x	61.0(+)	90	1254400
Second Layer in the 23th block of Conv4_x	62.3(+)	91	1254400
Third Layer in the 23th block of Conv4_x	64.8(*)	92	5017600
First Layer in the first block of Conv5_x	61.9(+)	93	627200
Second Layer in the first block of Conv5_x	65.5(+)	94	627200
Third Layer in the first block of Conv5_x	67.0(+)	95	2508800
First Layer in the second block of Conv5_x	64.9(*)	96	627200
Second Layer in the second block of Conv5_x	67.2(*)	97	627200
Third Layer in the second block of Conv5_x	67.3(+)	98	2508800
First Layer in the third block of Conv5_x	65.0(*)	99	627200
Second Layer in the third block of Conv5_x	68.2(*)	100	627200
Third Layer in the third block of Conv5_x	**69.3(+)**	**101**	**2508800**
FC	68.4(+)	102	4096

place of the fusion layer can affect the accuracy of the prediction. There are 4 kinds of methods to fuse the feature maps, including Sum fusion, Max fusion, Concatenation fusion, and conv fusion, whose formulas are shown below.

For time t, we fuse two feature maps X_t^a, X_t^b to an output y_t, where $X_t^a, X_t^b \in R^{H\times W\times C}$ and $y_t \in R^{H'\times W'\times C'}$. Note that H, W, C represent the width, height and number of channels of the respective feature maps.

Sum fusion. $y^{sum} = f^{sum}\left(X^a, X^b\right)$ computes the sum of the value of two point at the same location i, j, and c in spatial and temporal feature map. The new value at point (i, j, c):

$$y_{i,j,d}^{sum} = X_{i,j,c}^a + X_{i,j,c}^b, \tag{1}$$

where $1 \le i \le H, 1 \le j \le W, 1 \le c \le C$ and $X_t^a, X_t^b, y_t \in R^{H\times W\times C}$.

Max fusion. $1 \le i \le H, 1 \le j \le W, 1 \le c \le C$ takes the maximum of the two feature map:

$$y_{i,j,c}^{\max} = \max\left\{X_{i,j,c}^a, X_{i,j,c}^b\right\}, \tag{2}$$

where $1 \le i \le H, 1 \le j \le W, 1 \le c \le C$ and $X_t^a, X_t^b, y_t \in R^{H\times W\times C}$.

Concatenation fusion. $y^{cat} = f^{cat}\left(X^a, X^b\right)$ stacks two feature maps across the feature channels c:

$$\begin{cases} y_{i,j,2c}^{cat} = X_{i,j,c}^a \\ y_{i,j,2c-1}^{cat} = X_{i,j,c}^b \end{cases} \tag{3}$$

or,

$$\begin{cases} y_{i,j,1:c}^{cat} = X_{i,j,1:c}^a \\ y_{i,j,c+1:2c}^{cat} = X_{i,j,1:c}^b, \end{cases} \tag{4}$$

where $1 \le i \le H, 1 \le j \le W, 1 \le c \le C$, $X_t^a, X_t^b \in R^{H\times W\times C}$ and $y_t \in R^{H\times W\times 2C}$.

Conv fusion. $y^{onv} = f^{covv}\left(X^a, X^b\right)$ first stacks two feature maps across the feature channels c as Concatenation fusion and then convolves the stacked data with a bank of filters $f \in \mathrm{R}^{1\times 1\times 2C\times C}$ and add biases $b \in \mathrm{R}^c$:

$$y^{conv} = y^{cat} * f + b, \tag{5}$$

where $y_t \in R^{H\times W\times C}$.

Feichtenhofer *et al.* [11] shows that Conv fusion has the best performance while the Max fusion has the worst performance on UCF101. We adopt the Sum fusion in that this strategy has less parameters to compute and the performance is nearly as good as the Conv fusion strategy in our experiment. We argue that the score fusion or the late fusion undermines the spatial information hiding in the 3D spatial structure. Therefore, we propose to fuse the feature maps at several Conv layers ahead of fully connected layer rather than fusing the output of fully connected layer. Experimentally, we compare the difference on accuracy results from fusing different layers with different fusion strategies in Table 1 in Section 4.

D. ConvLSTM PROPAGATION

Convolutional long short-term memory (ConvLSTM) is a variant of traditional long short-term memory (LSTM). Fully-connected gates of the LSTM module are replaced by convolutional gates, which means ConvLSTM replaces matrix multiplication with convolution operation at each gate.

The equations of gates in ConvLSTM are:

$$\begin{cases} i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + b_i) \\ f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + b_f) \\ o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + b_o) \\ C_t = f \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \\ H_t = o_t \circ \tanh(C_t) \end{cases} \quad (6)$$

where i_t, f_t, o_t are the input, forget, and output gate of ConvLSTM. σ is the sigmoid function, $X_1, \ldots, X_t$ are inputs, $C_1, \ldots, C_t$ are the cell states, $H_1, \ldots, H_t$ are the hidden states, and $W_{x\sim}$, $W_{h\sim}$ are the 2D convolutional kernels. Meanwhile, '$*$' represents the convolution operator and '5×10^{-6}' represents the Hadamard product. Note that the inputs, the cell states, the hidden states, and the gates of ConvLSTM are all 3D (5×10^{-6}) tensors thereby the spatiotemporal relationships will be mostly retained throughout our network. In practice, inspired by Xingjian *et al*. [7], we built both single layer ConvLSTM and 2-layers ConvLSTM to verify if it is useful that utilize deeper network. We also explore the strategy—using larger state-to-state kernels for capturing spatiotemporal correlations—in our experiment, which is demonstrated to be useful by Xingjian *et al*. [7]. We set the size of all 2D convolutional kernels (input-to-state and state-to-state) to 5×5 and 3×3 for comparison, and the size of all output states is $7 \times 7 \times 300$. Besides, we pad the boundary points of hidden states using zero padding. Once the feature maps finish to propagate in ConvLSTM, a global average pooling is performed on the output state of the ConvLSTM and is applied to the softmax layer.

E. IMPLEMENTATION DETAILS

Since the temporal-stream ConvNet is transformed by the spatial-stream ConvNet, we initially set the learning rate of the spatial-stream ConvNet to 5×10^{-6} and set the learning rate of the temporal -stream ConvNet to 5×10^{3} in that the distribution of optical flow frames is not close to RGB data and we found that if we set learning rate of the temporal -stream ConvNet same as spatial-stream ConvNet the tuning process would be quite slow. For two stream ConvNets, the learning rate will be divided by 10 when the accuracy is saturated. The weight decay in our training process for ConvNets is set to be 1×10^{-4}, and momentum is set to 0.9 to prevent overfitting. For ConvLSTM, we adopt ADAM optimizer to help train ConvLSTM. Meanwhile, we set the learning rate to 5×10^{-5} for ConvLSTM.

We use several data augmentation methods to prevent overfitting in that our training dataset is not large enough. In the proposed approach, first, we use Random Crop to crop sub-image with size 256×256. Second, randomly scale the cropped image to the size of 3/4 to 4/3 of its original size and then the randomly scaled images and cropped images are scaled to 224×224 again. Finally, color jittering is used for temporal-stream ConvNet.

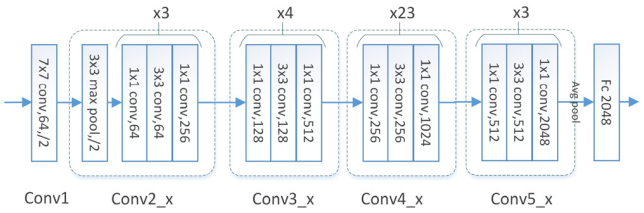


FIGURE 3. Illustration of the location of different fusion layers in ResNet101.

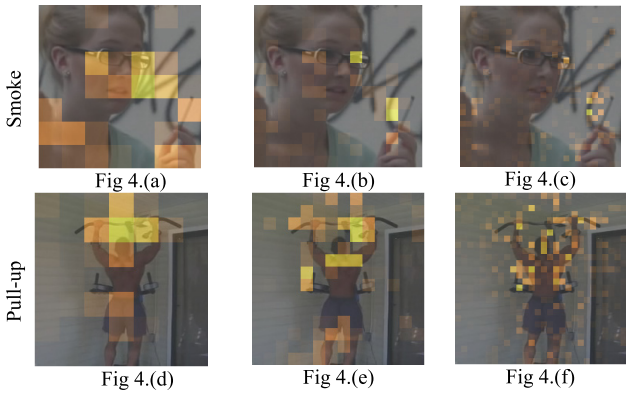


FIGURE 4. Saliency maps of feature maps of different convolutional layers: (a),(d):Conv5_x. (b),(c):Conv4_x. (c),(f): Conv3_x.

IV. EVALUATION

A. DATASETS

We evaluate our approach on two popular activity recognition benchmarks datasets: UCF101 and HMDB51, which consist of 13320 action videos in 101 categories and 6766 action videos in 51 categories respectively. We split the UCF101 and HMDB51 into three splits as the official instruction of these two datasets for training and validating our proposed models. And the results we calculated is the average results over three splits.

B. FUSION STRATEGY

The location of Fusion layers in ResNet101 in Table 1 can be illustrated in Fig3. ($*$), ($+$), (&): denote Conv Fusion, Sum Fusion, and Concatenation Fusion respectively. FC: denotes fully-connected layer and we use LSTM to learn the information in the feature map of FC. #Parameters: denotes the parameters numbers needed for representing a video.

The output of every convolutional layer is a feature map that maintains the spatial structure. To identify where is the most suitable location of the layer that can best express spatial information, we explored the impact on the accuracy when we choose different fusion layers. And the result in Table 1 indicates that (1) Even though the numbers of parameters of latter Conv layers and the former Conv layers are on the same order of magnitude, it is demonstrated that using feature maps in later location in a specific ConvNet instead of the former ones can help us achieve better accuracy.

This is because the spatial attribution of a feature map (i.e, how the neurons in feature maps affect the CovNet's output) tends to be more meaningful in later layers. To give

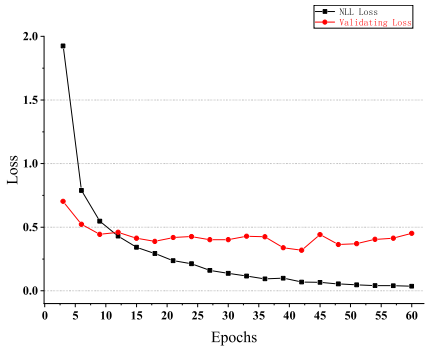


Fig 5.(a)

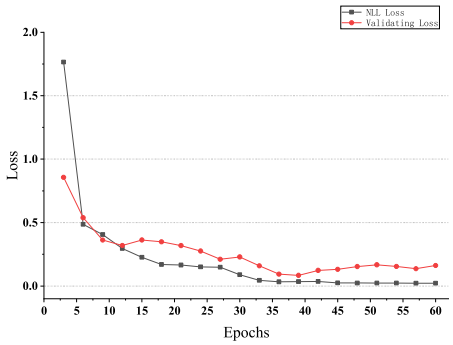


Fig 5.(b)

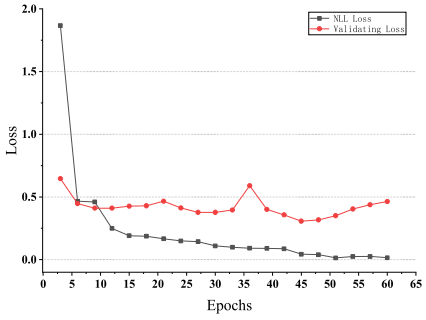


Fig 5.(c)

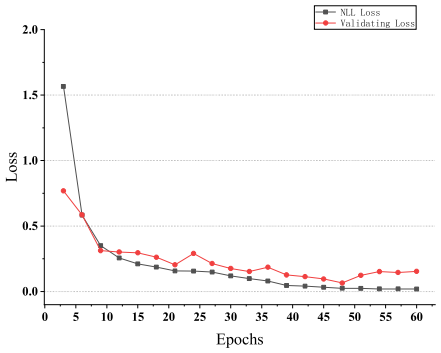


Fig 5.(d)

FIGURE 5. **(a)Learning curve of ConvLSTM of 1 layer on HMDB-51. (b) Learning curve of ConvLSTM of 1 layer on UCF-101. (c) Learning curve of ConvLSTM of 2 layers on HMDB-51. (d) Learning curve of ConvLSTM of 2 layer on UCF-101.Note that all the kernel size of figures above is 5 × 5 whose accuracy is higher than smaller 3 × 3 kernel by 0.5-1% depending on different datasets and different models.**

more visual explanation for this, we depict a series of saliency maps—simple heatmaps that highlights pixels of the input image that most caused the output classification— on the frame pictures of HMDB51 when we fine-tune the spatial-stream to fit the specified class of HMDB51. We follow the work of Zhou *et al*. [33] and Selvaraju*et al*. [34] and overlay these saliency maps on activation grids to provide information track such as pictures in Fig 4. The orange patches in Fig.4 represent how much impact this area have on classification results when the model do inference. The patches that are more orange represent greater contribution of this part will contribute to the classification result. The saliency maps of Fig.4 (a) and Fig.4 (d) are heatmaps of the last convolutional layers, and the orange patches in these maps are more integral and intensive. And we can find that feature maps that are more farther away from the fully connected layer have more disperse heatmaps, which means the contribution of these layers to the classification result will be much smaller. (2) Using feature maps that are outputs of inside layers in the ResNet blocks instead of feature maps that are output of last layer will lead to significantly worse performance since it attenuates the effectiveness of Shortcut Connection.(3) Sum Fusion and Conv Fusion can achieve similar performance on accuracy but less computational cost is needed when using Conv Fusion. These experiments lead us to conclude that the classification can be leveraged if the location and method of fusing can be carefully selected.

TABLE 2. **Complete comparison of each component in ConvLSTM onUCF101 and HMDB51 split 1.**

Layers of ConvLSTM	Kernel Size	Dropout	Acc on HMDB51	Acc on UCF101
1	3x3	1.0	66.3%	91.3%
1	3x3	0.8	66.5%	91.5%
1	5x5	1.0	66.7%.	92.3%
1	5x5	0.8	67.4%	92.5%
2	3x3	1.0	67.0%	92.4%
2	3x3	0.8	67.3%	92.9%
2	5x5	1.0	68.6%	93.3%
2	5x5	0.8	**69.3%**	**93.6%**
3	3x3	1.0	67.6%	93.2%
3	3x3	0.8	66.8%	93.0%
3	5x5	1.0	67.8%	92.9%
3	5x5	0.8	67.5%	93.1%

C. RESULTS

The proposed network are implemented based on the Torch7 [25] framework and we have released our code on Github[1]. We trained the proposed model with optimization

[1]https://github.com/yww211/two_stream_and _convLSTm#two_stream_and_convlstm

TABLE 3. Accuracy of action recognition techniques.

Method	Type	UCF-101	HMDB-51
Two-Stream [1]	D	88.0	59.4
Hybrid-BoW(Bag of Words)[26]	R	87.9	61.1
IDT[10]	R	61.7	86.4
Multi-Skip Feat. Stacking[27]	R	65.1	89.1
Dynamic Image Networks + IDT [28]	F	89.1	65.2
C3D[15]	D	85.2	-
C3D+IDT[15]	F	90.4	-
Convolutional Two-stream[11]	D	92.5	65.4
Convolutional Two-stream + IDT[11]	F	93.5	69.2
LRCN(Long-term Recurrent Convolutional Networks)[16]	R	82.9	-
SR-CNN(Semantic Region based CNN)[29]	D	92.6	-
ST-ResNet(Spatio-Temporal Residual Network)[30]	D	93.4	-
ST-ResNet + IDT [30]	F	94.6	70.3
Two-stream + LSTM[3]	D	88.6	-
TSN (2 modalities)[4]	D	94.0	68.5
Two-stream +TS-LSTM[2]	D	94.1	69.0
Two-stream +Temporal-Inception[2]	D	93.9	67.5
Two-Stream I3D(Kinetics pre-training)[10]	D	97.8	80.9
Ours(RGB+ConvLSTM)	D	89.7	64.8
Ours(Optical flow+ConvLSTM)	D	87.4	64.2
Ours(Two-stream+ConvLSTM)	D	**93.9**	**69.3**

means mentioned in the Section 3.5. We implemented random shuffling to every iteration among all 60 epochs in RNN part, and it takes about 30 hours to learn on every split of HMDB51 with a NVIDIA GTX1060(6G) GPU. As is mentioned in Section 3.4, we built three ConvLSTM whose number of staking ConvLSTM layers varies form single layer to three in order to verify if it is useful that utilize deeper ConvLSTM network and we also explore the strategy using different size of kernel in the ConvLSTM. Fig.5 shows the learning curves by two models that one is single layer and another is 2-layer on split 1 of HMDB51 and UCF101. The black curve denotes Negative Likelihood Loss (NLL) on training set and the red curve denotes validating error trend on testing set.

Result from Fig 4 shows that the 5×5 kernel can help the lowest point of validating error hit lower place than 3×3 kernel no matter which dataset is used for testing or how deep the layer of ConvLSTM is. Besides, we argue that slightly deeper layers in ConvLSTM can help us achieve better performance but much deeper structure does not necessarily help in achieving better action recognition performance. Furthermore, the complete result of comparison of each component in ConvLSTM is shown in the Table 2 below.

D. FINAL PERFORMANCE

Finally, we compare against the state-of-the-art over all three splits of UCF101 and HMDB51 in Table 3.

The column Type indicates which method in purely Deep-net Based (D), Representation Based (R) or Fused Solution (F) is used.

While Two-Stream I3D [10] achieved significant breakthrough on activity recognition with pre-training on Kinetics dataset, we can still find our state of art position among the methods using Deep-net without pre-training on Kinetics dataset. To be specific, comparing other deep learning methods, our best result 93.9% accuracy on UCF-101 and 69.3% accuracy on HMDB-51, outperforms the model [2] using two-stream and RNN by 0.3% on HMDB51 dataset, TSN (2 modalities) [4] by 0.8% on HMDB51 and ST-ResNet [30] by 0.5% on UCF101 dataset. Besides, our experiment shows that our proposed model also achieves better performance than some top-tier fused solutions which can be termed as concrete manifestations of ensemble learning. The result outperforms Convolutional Two-stream +IDT [11] by 0.1% on HMDB51 and C3D+IDT [15] by 3.5% on UCF101. On the other hand, we also compare our proposed two-stream model with one-stream models in last three rows in Table 3, which indicates no matter spatial information or temporal information should not been neglected. The performance of our methods demonstrates the effectiveness of Two-stream +ConvLSTM and justifies the significance of integrality of spatial information when the sequential data show correlations in space. We can say the spatiotemporal correlation information is kept through the whole feature map learning process. Therefore, we argue that it is the introduction and

fine adjustment of ConvLSTM in two-stream network that bring about the better performance comparing those networks that did not consider the necessity of maintaining spatial structure.

V. CONCLUSION & DISCUSSION

Generally, ConvNets with Two-stream of the optical flow and original RGB have been widely used in activity recognition. The method of Two-stream ConvNets + RNN has been proved to be competitive in dealing with video understanding problems. However, these works use the one-dimensional hypercompressed vector as spatiotemporal feature, where the spatial structure of a video is largely damaged and the models are prone to be overfitting. In this paper, we explored the method of ConvLSTM to retain the spatial structure when samples passing through the ConvLSTM. First, we used TV-L1 algorithm to extract the optical flow and we rescaled the horizontal and vertical components' value range to make sure our Two-stream ConvNets works well. Second, we analyzed why ConvLSTM suit our task. Third, we studied the impact brought about by different fusion strategies and we demonstrated that, by appropriately integrating spatial and temporal feature maps and retaining the 3D structure in RNN part, the proposed method achieved state-of-the-art accuracy on both UCF101 and HMDB51 in terms of Deep-net without using Kinetics to pre-train our model. And we gained the result just by using common optimization method and manually searching the heperparameters for our ConvNets. On the other hand, as the significant improvement shown by Kinetics from the Two-Stream I3D method, the significance of the existence of extremely huge pre-training dataset is highlighted again in wrestling with the dilemma of improving generalization capability of deep learning models. Besides, the performance improvement brought by fused solutions should not be neglected as well, and representation based solutions is ought to be termed as complements of Deep-nets to achieve better overall performance. We also need to see the limitations in our work. The relatively large memory and GPU cost is needed in this work and the proposed network is still a two-stage network. And this network may not work well on those dataset whose length of video sequences largely varies. But there is still room for improvement in this work. For instance, method of sampling with variable intervals can be deployed like [31] to satisfy variable-length video sequences datasets, and method of reducing parameters number can be tried like [32] to speed up learning process. Besides, in the future, we also plan to pre-train our models on Kinetics with the capability of hardware enhanced, and try to use multiple models like [4] or fused solutions like [11], [30] to achieve better performance. Last but not least, in order to achieve better performance we can also deploy FPN(Feature Pyramid Networks)mentioned in the work of Lin *et al*. [35] in that the activities occupy different size of regions in the video dataset. In this way can we actually capture the activities' spatial information on different scales to improve model's performance.

REFERENCES

[1] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. CVPR*, Jun. 2015, pp. 2625–2634.

[2] C.-Y. Ma, M.-H. Chen, Z. Kira, and G. AlRegib, "TS-LSTM and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition," 2017, *arXiv.org/abs/1703.10677*. [Online]. Available: https://arxiv.org/abs/1703.10667

[3] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, vol. 1, no. 1, pp. 568–576.

[4] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. ECCV*, 2016, pp. 20–36.

[5] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proc. CVPR*, Jun. 2015, pp. 4694–4702.

[6] G. Zhu, L. Zhang, P. Shen, and J. Song, "Multimodal gesture recognition using 3-D convolution and convolutional LSTM," *IEEE Access*, vol. 5, pp. 4517–4524, 2017.

[7] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. NIPS*, 2015, pp. 802–810.

[8] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. ICCV*, Dec. 2013, pp. 3551–3558.

[9] P. Liu, J. Wang, M. She, and H. Liu, "Human action recognition based on 3D SIFT and LDA model," in *Proc. RiiSS*, Apr. 2011, pp. 12–17.

[10] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the Kinetics dataset," in *Proc. CVPR*, Jul. 2017, pp. 4724–4733.

[11] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. CVPR*, Jun. 2016, pp. 1933–1941.

[12] L. Sevilla-Lara, Y. Liao, F. Guney, V. Jampani, A. Geiger, and M. J. Black, "On the integration of optical flow and action recognition," 2017, *arXiv:1703.10677*. [Online]. Available: https://arxiv.org/abs/1712.08416

[13] 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.

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

[15] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. ICCV*, Dec. 2015, pp. 4489–4497.

[16] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. ICML*, 2014, pp. 647–655.

[17] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Proc. NIPS*, 2015, pp. 2377–2385.

[18] V. Patraucean, A. Handa, and R. Cipolla, "Spatio-temporal video autoencoder with differentiable memory," 2015, *arXiv:1511.06309*. [Online]. Available: https://arxiv.org/abs/1511.06309

[19] J. R. Medel and A. Savakis, "Anomaly detection in video using predictive convolutional long short-term memory networks," 2016, *arXiv:1612.00390*. [Online]. Available: https://arxiv.org/abs/1612.00390

[20] Y. Kim, D. Lee, and S. Lee, "First-person activity recognition based on three-stream deep features," in *Proc. ICCAS*, Oct. 2018, pp. 297–299.

[21] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. ECCV*, 2004, pp. 25–36.

[22] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L^1 optical flow," in *Proc. JPRS*, 2007, pp. 214–223.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556v6*. [Online]. Available: https://arxiv.org/abs/1409.1556

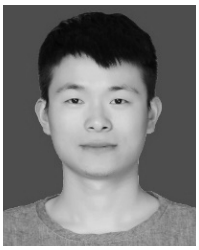
[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. CVPR*, Jun. 2015, pp. 1–9.

[25] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *Proc. NIPS*, 2011, pp. 1–6.

[26] X. Peng, L. Wang, X. Wang, and Y. Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice," *Comput. Vis. Image Understand.*, vol. 150, pp. 109–125, Sep. 2016.

[27] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj, "Beyond Gaussian pyramid: Multi-skip feature stacking for action recognition," in *Proc. CVPR*, Jun. 2015, pp. 204–212.
[28] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould, "Dynamic image networks for action recognition," in *Proc. CVPR*, Jun. 2016, pp. 3034–3042.
[29] Y. Wang, J. Song, L. Wang, L. Van Gool, and O. Hilliges, "Two-stream sr-cnns for action recognition in videos," in *Proc. BMVC*, 2016, pp. 108.1–108.2.
[30] C. Feichtenhofer, A. Pinz, and R. Wildes, "Spatiotemporal residual networks for video action recognition," in *Proc. NIPS*, 2016, pp. 3476–3484. [Online]. Available: http://dl.acm.org/citation.cfm?id=3157382.3157486
[31] K. Zhou, Y. Zhu, and Y. Zhao, "A spatio-temporal deep architecture for surveillance event detection based on ConvLSTM," in *Proc. VCIP*, Saint Petersburg, FL, USA, Dec. 2017, pp. 1–4.
[32] L. Zhang, G. Zhu, L. Mei, P. Shen, S. A. A. Shah, and M. Bennamoun, "Attention in convolutional LSTM for gesture recognition," in *Proc. NIPS*, 2018, pp. 1953–1962.
[33] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. CVPR*, 2016, pp. 2921–2929.
[34] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," 2016, *arXiv:1610.02391*. [Online]. Available: https://arxiv.org/abs/1610.02391
[35] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. CVPR*, Jul. 2017, pp. 936–944.

W. YE was born in Taixing, Jiangsu, China, in 1995. He is currently pursuing the master's degree majoring in electronics and communication with the School of Physics and Technology, Nanjing Normal University. His main research interests include activity recognition, face verification, and image process.



J. CHENG was born in Yangzhou, Jiangsu, China, in 1994. He is currently pursuing the master's degree majoring in image processing and optical character recognition with Nanjing Normal University.

F. YANG was born in Lianyungang, Jiangsu, China, in 1996. She is currently pursuing the master's degree majoring in instrumentation with Southeast University. Her main research interests include image process, visual simultaneous localization, and mapping.

Y. XU received the Ph.D. degree in acoustics from Nanjing University, in 2004. He is currently a Professor with the School of Physics and Technology, Nanjing Normal University. His research interests include biomedical electronics and precision intelligent instrument design. He has been engaged in the detection of electrophysiological signals (mainly in the design and manufacture of medical instruments), the analysis of nonlinear dynamics of electrophysiological signals (theoretical research), and the classification of high frequency electrocardiogram using neural networks.