Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 5, No. 1, February, 2021

Human Action Recognition in Videos using Convolution Long Short-Term Memory Network with Spatio-Temporal Networks

Ashok Sarabu [1*], Ajit Kumar Santra [1*]

[1] *SITE School, VIT University, Vellore -632014, TamilNadu, India*

Abstract

Two-stream convolutional networks plays an essential role as a powerful feature extractor in human action recognition in videos. Recent studies have shown the importance of two-stream Convolutional Neural Networks (CNN) to recognize human action recognition. Recurrent Neural Networks (RNN) has achieved the best performance in video activity recognition combining CNN. Encouraged by CNN's results with RNN, we present a two-stream network with two CNNs and Convolution Long-Short Term Memory (CLSTM). First, we extricate Spatio-temporal features using two CNNs using pre-trained ImageNet models. Second, the results of two CNNs from step one are combined and fed as input to the CLSTM to get the overall classification score. We also explored the various fusion function performance that combines two CNNs and the effects of feature mapping at different layers. And, conclude the best fusion function along with layer number. To avoid the problem of overfitting, we adopt the data augmentation techniques. Our proposed model demonstrates a substantial improvement compared to the current two-stream methods on the benchmark datasets with 70.4% on HMDB-51 and 95.4% on UCF-101 using the pre-trained ImageNet model.

Keywords:

Convolution LSTM;
Action Recognition;
Human Activity;
Two-Stream Networks.

Article History:

Received:	28	November	2020
Revised:	16	January	2021
Accepted:	25	January	2021
Published:	01	February	2021

1- Introduction

Human Action Recognition (HAR) in videos has received tremendous attention in the realm of pattern recognition and computer vision academic and research community because of its broad spectrum of applications like video monitoring, video retrieving, human-computer interaction, medical applications, etc. Compared to still image recognition, video action recognition is difficult. Because videos contain temporal correlation between frames, these temporal data is additional information that needs to be analyzed to find the action in a video. At the same time, this task demands more computations, because each video contains hundreds of frames. In recent times, deeper CNN applications have shown a steep performance increase in video activity recognition. Driven by the rapid growth in the performance of deep CNN models, the computer vision academic and research community started to expand the application of CNNs to human action recognition [1, 2].

Action classification in videos is comparatively slow when compared to action classification in still images. There are two factors for comparatively slow; existing video activity recognition datasets are small in size and diversity compared to the still image recognition datasets. Therefore, datasets that are small in size will overfit the model and cannot generate the generalized solution for action classification. It is also hard and challenging to build bigger-sized video datasets and train them on deep networks. Second, the video classification task requires complex data analysis because it consists of additional information called temporal data. Recently, many researchers have addressed the

* **CONTACT**: Sarabu.ashok@gmail.com; Ajitkumar@vit.ac.in

DOI: http://dx.doi.org/10.28991/esj-2021-01254

solution to these complex action classification problems in videos. Karpathy et al. [1] proposed different performance analysis solutions to the video activity classification on SPORTS-1M dataset and showed the several CNN models outcomes.

Simonyan et al. [3], first developed two-stream CNN architecture, which used two CNNs for recognizing human activities in videos.. Even though the authors did not achieve great performance compared to the hand-crafted solution, but laid a path in which further research showed a considerable performance increase. In a two-stream methodology, feature maps extracted from the RGB image and optical flow images contain spatial information and additional cue called temporal information. The final prediction is calculated by fusing the results of two CNNs. Moreover, many researchers have explored the two-stream network architectures and proven with good performance. However, these architectures lack in exploiting Spatio-temporal dynamics. To solve this, researchers further extended and proposed the architecture with the combination of CNN and Recurrent Neural Network [3–5]. In recent two-stream architectures with CNN and RNN models, CNN resulting vectors is fed as an input to the RNN. Since the input of the RNN is the output of CNN, it converts the three-dimensional feature maps to one-dimensional feature vectors [6]. Doing this process will decrease the number of parameters compared to its previous work, and this process will diminish the spatial information. Xingjian et al. [7] extended Long Short Term Memory (LSTM) to three-dimensional and proposed CLSTM, proved with better performance. We further extend this method with different architecture; that is, we trained two streams architecture/model end-to-end and fuse the output and feed it as input to CLSTM.

We propose a two-stream architecture for action recognition with a combination of CNNs and RNN, as shown in Figure 1. First, we train and fine-tune the spatial stream and temporal stream networks with inputs as RGB image and optical flow frames with a pre-trained ImageNet model. Second, we fuse two stream's outputs with their respective dimensions (7*7*2048). Finally, to train the long-term temporal dependencies, the resulting fully connected layers output features are given as input to the CLSTM. Furthermore, Article is organized as 2. Related work, discussion of a related literature survey, 3. Technical approach, 4. Experimental section discussion of implementation details and comparison with State-of-art results.

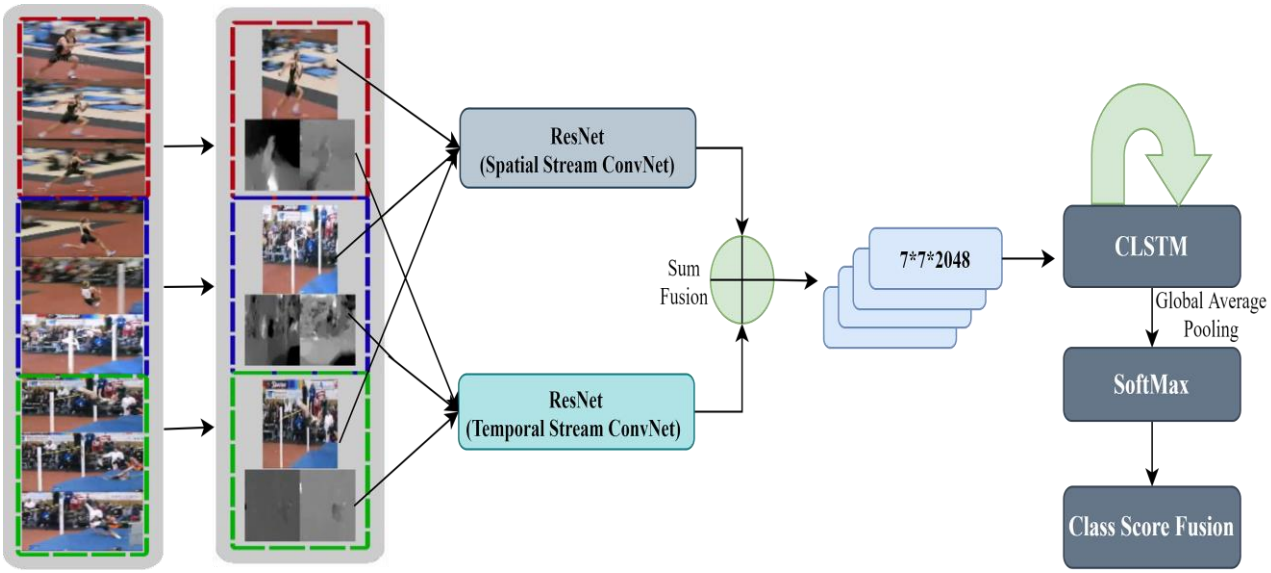

Figure 1. Proposed Spatio-Temporal CNN + RNN model.

2- Related Works

Hand-crafted methods for activity recognition in videos like space-time interest point [8], 3D-SIFT [9], dense trajectories [10] have shown good performance, but the solutions are not generalized. Whereas deep learning techniques for video activity recognition demand more computational cost than hand-crafted techniques, but solutions are more generalizable and shown to improve accuracy. But, with the decrease in the hardware cost, training deep networks became easy and produces better results. With the generalized solutions, deep CNN has achieved tremendous progress in the task of action recognition. Moreover, deep CNNs are occupying and replacing the hand-crafted methods because of the lesser the hardware cost. Along with this, deep CNNs also perform well in motion recognition tasks using RGB and optical flow frames extracted from videos. Optical flow frames are work as one of the hidden cues in gaining high-performance accuracy.

Two stream architecture utilizes two convolutional models, one CNN to extract spatial data and another CNN to extract temporal data in videos. Feichtenhofer et al. [11] introduced a new two-stream CNN model and investigated various methods to fuse two stream's results. They showed that fusing of the output of networks spatially at final

convolve layers will increase the accuracy. K. Simonyan et al. [3] introduced a two-stream architecture for video activity classification, where RGB images are given as input to the spatial stream network, and optical flow frames are given as input to the temporal stream network. Final action classification scores are calculated by combining the outputs of the spatial and temporal stream network. Wang et al. [12] introduced the Trajectory-pooled Deep-convolution Descriptor (TDD), in this method author's trained using deep CNNs and unified the trajectory features. This method shows significant improvement in performance by combining depth network features and shallow local features. Wang et al. [5] presented Temporal Segment Networks; author's showed an increase in accuracy by training complete video by introducing long-range temporal model. Feichtenhofer et al. [13] proposed ST-ResNet; authors trained the models by combining spatial and temporal network models, improving the correlation across two streams. Ji et al. [14] presented an end-to-end learning architecture; this method executes the pixel-level activity classification and segmenting. With this, the authors addressed the video activity recognition by two-stream CNN architecture and aggregating temporal information. Carriera et al. [15] proposed I3D networks, which achieves the best performance and accuracy using the three-dimensional convolutional neural networks, using the pre-trained models, Kinetics and ImageNet. Tran et al. [2] proposed a new three-dimensional CNN called C3D, which works using three-dimensional convolutional kernels.

CNN with RNN is another approach to classify action recognition in videos. RNN is capable of encoding the present state and retrieving the temporal information. Among the different RNNs, LSTM architecture/model performs better to capture the long-range temporal information and differentiate videos with intra-class variations from the input video. Donahue et al. (2014) and Srivastava et al. (2015) the authors proposed a model with one convolutional neural network and a fully connected Long Short-Term Memory Network for video action recognition [16, 17]. Ma et al. [4] adopted the Temporal-Segment method [5] to extricate the long-range temporal dependency giving output to LSTM layers and achieved the performance improvement. Xingjian et al. [7] introduced convolutional LSTM, replacing the fully connect gates with convolutional gates to extract spatiotemporal information effectively. Xingjian et al. (2015) [7], applied the proposed convolutional LSTM method for radar images and achieved better results. Later, many researchers [18–20] applied convolutional LSTM and demonstrated it as a good choice. After the literature survey, convolutional LSTM is used as one of the models in our proposed approach.

3- Technical Approach

Figure 2 demonstrates the basic network framework of our proposed model. The proposed methodology consists of a pre-trained CNN model, data pre-processing for temporal stream network, CLSTM.

3-1- Two-Stream CNN

Video is a combination of frames. The sequence of frames in the video contains spatial information and motion information. Human visual sensory processes the perceived information with spatial and temporal streams. Spatial and temporal streams are two individual systems in terms of receiving and processing the data. Only static information is processed in the spatial stream process; that is, only static image appearance and object are identified. In the temporal stream process, the motion of the objects is identified using information across the frames; that is, only the object's motion is calculated. Simonyan et al. [3] presented two-stream CNN to process spatial and temporal data with two convolutional models; one for spatial stream and another for the temporal stream. In the spatial stream, RGB images are fed as input to the spatial stream to recognize still objects. To be specific, single RGB images are used as input to the spatial stream convolutional network. To recognize the motion across the sequence of frames, optical flow frames are given as input to the temporal network. The optical flow frame is the collection of horizontal and vertical convolved frames, and the total number for this is initialized to 2L=20. Lastly, the spatial stream and temporal stream are trained independently. The final output of two streams is fed as input to the convolutional long short term memory (CLSTM). The input of soft-max layers is the output of CLSTM to identify the final score for classification.

In this subsection, we present a human activity recognition in videos using CLSTM with Spatio-temporal networks. In the proposed network model, the spatial network and temporal network are trained with two CNNs and CLSTM, as shown in Figure 2. The purpose of modeling our network model is to retain the spatial information using RNN at the end of two streams. Second, to show that RNN (CLSTM) perform better along with the original two-stream architecture. With many experiments, we observed that the model accuracy of the proposed architecture is superior to the existing two-stream model.

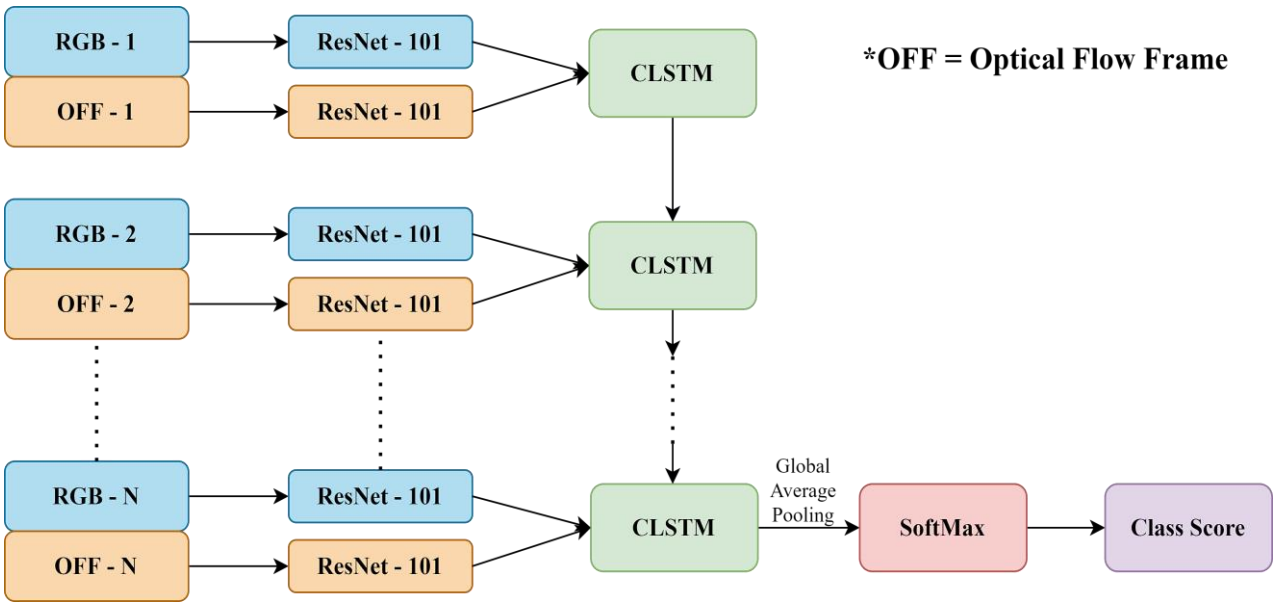

Figure 2. Basic proposed two-stream network architecture.

3-2- Residual Networks

Residual networks are adopted to train spatial and temporal stream networks in our proposed network. Since deep layer ResNet extricates discriminant features from frames, we use deep ResNets instead of shallow layered ResNet. As the number of layers increases, network degradation increases. To mitigate this problem, He et al. [21] introduced the deeper layered ResNet. Rather than the vanilla fit function, they trained the ResNets by mapping function $F(x) := H(x) - x$. Residual unit as $x_{k+1} = \sigma(x_k + F(x_k, W_k))$. Where x_k and x_{k+1} are the input and output of the k^{th} network layer. σ is the ReLU function [22], $F(x_i; W_k)$is non-linear residual mapping of weight of CNN filters. The implementation of the residual block is because, it works as the shortcut link that interconnects the first layer to any other layer in the network, which breaks the traditional method of connecting one layer to the preceding layer. This step will avoid the gradient explosion problem by bypassing some layer's loss and directly transferring the loss to any connected shallow layers in the network. This simplified solution will avoid the increase in the number of parameters and computational complexity. In Residual networks, batch normalization (BN) [23] is performed before every activation layer and after every convolution operation. With this step, the network's convergence will be fast; along with this, the problem of co-variate shift is solved [21]. Finally, instead of a fully connected layer and SoftMax layers, global average pooling and SoftMax operations are combined. With this step, the number of parameters is effectively reduced. In addition to this, the bottleneck structure will reduce the cost of complexity, and the overall network model's performance is guaranteed.

3-3- Data Pre-processing for Temporal Stream

Optical flow frames are the input to the temporal stream. Frames are given as input data so that the input to temporal stream is trainable on pre-trained networks. For this, data need to be pre-processed to get the optical flow frames from RGB images. Two optical flow frames are generated for every processed RGB image, one frame with vertical and another with horizontal edges. There are two popular methods to get optical flow frames. 1. Brox [24] and 2. Total Variation- L1 [25]. Ma et al. [4], the authors have demonstrated that Total Variation- L1 performs better than Brox. We establish the same network mentioned in [4,15,26,27] and stack the optical flow frames with input L=20 consisting of 10 vertical and ten horizontal edge frames. Along with this, we perform horizontal transformation, vertical transformation, re-scaling to keep the input frame in the range of [0,255]. The final frame values will be in [0,255] and able to use pre-trained networks.

3-4- Convolutional LSTM

Convolution LSTM is a variant of vanilla LSTM, in which fully connected gates are replaced with convolutional gates. Instead of matrix multiplications in fully connected gated, Convolution gate performs convolution operations at every gate. And, equations of the convolutional gates in Convolution LSTM are:

$$i_q = \sigma(W_{li} * X_q + W_{mi} * H_{q-1} + b_i) \quad (1)$$

$$f_q = \sigma(W_{lf} * X_q + W_{mf} * H_{q-1} + b_f) \quad (2)$$

$$o_q = \sigma(W_{lo} * X_q + W_{mo} * H_{q-1} + b_o) \quad (3)$$

$$C_q = f \circ C_{q-1} + i_q \circ \tanh\left(W_{xc} * X_q + W_{mc} * H_{q-1} + b_c\right) \quad (4)$$

$$H_q = o_q \circ \tanh\left(C_q\right) \quad (5)$$

In above equation, *i, f, o* are convolution LSTMs, input, forget, and output gates. *C* and *H* are cells and hidden states. Sigma is the sigmoid function. *W* are the convolutional kernels. * is the concatenation operation. All the inputs, convolutional gates, cells, and hidden states are the three-dimensional tensors of size $5/10^6$. With these operations, Spatio-temporal relations will be maintained throughout the network. We initialize the two-dimensional input and output convolutional kernels to 5*5 and 3*3. Hidden states are zero-padded, and dimensions of all output to 7∗7*300. Once the features are trained in convolutional LSTM, global average pooling is applied to it. The final classification is decided using the SoftMax layer. Inspired by the [7], we try to explore the results of the single-layer convolutional LSTM and two-layers convolutional LSTM.

3-5- Fusing Techniques

In Feichtenhofer et al. (2016) [11] research, the authors demonstrated how to combine the feature extracts from the two-stream convolutional architecture. The authors also showed the different fusing methods (sum, max, concatenation, Conv.) and their accuracy. At temporal/time location L, we fuse the feature maps X_l^a , X_l^b of two CNNs to y_l Where X_l^a , $X_l^b \in \mathbb{R}^{V*H*C}$ and $y_l \in \mathbb{R}^{V'*H'*C'}$. V, H denotes height and width and C denotes number of channels of the feature maps.

3-5-1- Max Function

$y^{max} = fuse^{max}(X^a, X^b)$ computes the maximum of two CNN feature outputs at spatial location j, k, and at channel l.

$$y_{j,k,c}^{max} = maximum\{X_{j,k,c}^a + X_{j,k,c}^b\} \quad (6)$$

Where $1 \le j \le V, 1 \le k \le H, 1 \le c \le C$ and $x^a, x^b \in \mathbb{R}^{V*H*C}$

3-5-2- Concatenation Function

$y^{concat} = f^{concat}(X^a, X^b)$ stacks the output of the two CNN feature maps at the same spatial points j, k across the feature channels c

$$y_{j,k,2c}^{concat} = x_{j,k,c}^a \; and \; y_{j,k,2c-1}^{concat} = x_{j,k,c}^b \; where \; y \in \mathbb{R}^{V*H*2C} \quad (7)$$

3-5-3- Conv. Function

$y^{conv} = f^{conv}(X^a, X^b)$ stacks the same spatial locations j, k of the feature maps at feature channels c, similar to the above-mentioned concatenation function. And, convolve the features maps of stacked data with the filter size $f \in \mathbb{R}^{1*1*2c*c}$ and biases $b \in \mathbb{R}^c$.

$$y^{conv} = y^{concat} * f + b, \quad (8)$$

Where c in number of channel of output. 1*1*2c is the dimension of the filter.

3-5-4- Sum Function

$y^s = f^s(X^a, X^b)$ performs of the sum of the two feature maps of the CNN models at the spatial location j, k, and feature channel c.

$$y_{j,k,c}^{sum} = x_{j,k,c}^a + x_{j,k,c}^b \quad (9)$$

Where $1 \le j \le V$, $1 \le k \le H$, $1 \le c \le C$ and $x^a, x^b, y \in \mathbb{R}^{V*H*C}$.

Among all the fusion methods mentioned in Feichtenhofer et al. (2016) [11] research, Conv. fusion shows the best, and max fusion shows the worst performance. In our proposed method, we adopt the sum fusion method to aggregate the results of two-streams. Since sum fusion contains fewer computations than the other fusion functions and results are equivalent to the Conv. fusion function. Therefore, in the proposed model, we use the sum fusion function as a fusion function and compare the fusion results at the different CNN layers and tabulated in Table 1.

4- Experiments

4-1- Datasets and Implementation Details

We conducted the experiments and evaluated our proposed methodology on two large-scale video activity recognition datasets, HMDB-51 dataset [28] and UCF-101 dataset [29]. HMDB-51 activity recognition dataset consists of 51 activity classes and an overall of 6849 video clips. Each action category contains a minimum of 101 clips. HMDB-51 dataset is a collection of video clips from the sources of YouTube videos, Google videos. The UCF-101 consists of 101 activity

classes and an overall 13,3220 videos. On average, every video consists of 100 to 300 frames with 3 – 10 seconds of duration. Similarly, experiments are performed on these two datasets using a standard evaluation scheme with three train and test splits of the UCF-101 dataset. The findings of our proposed method are compared with the State-Of-The-Art methods. Analysis of the accuracies is evaluated on the three splits of both datasets.

We adopt different data augmentation methods to avoid CNN overfitting because of the smaller size of the datasets. In the proposed architecture, first, we apply random cropping with the size of 256*256. Second, random scaling is performed on the cropped image to 0.75 of its previous image size. Furthermore, we scale up the resulting image to 224*224.

The ADAM optimizer method is adopted to train the weight of the networks. We initially set the network weights with pre-training model weights from ImageNet [22]. We set the weight decay, batch size to 10^{-4}, 256 for both spatial and temporal stream CNNs. And, to prevent the over-fitting of both CNNs, momentum is set to $9*10^{-1}$.

We initialize the value of momentum to 0.9. Initially, the spatial and temporal stream network's learning rate is initialized to $0.5 *10^{-7}$ and $0.5 *10^{-4}$. The learning rate of the spatial stream network is reduced to 0.1 after every 15000 iterations, and the complete training of the network stops at 36000 iterations. Similarly, the temporal stream network's learning rate is reduced to 0.1 at 20000 and 32000 iterations, and the complete training of the temporal stream network halt at 40000 iterations. For CLSTM, the initial learning rate is set to $0.5*10^{-6}$. We implement the random shuffle for every iteration for all 60 epochs in the CLSTM. TVL1 optical flow algorithm [25] is employed to generate optical flow frames from videos. We use data parallelization to accelerate the training process with multiple GPUs on the Pytorch platform, and associated code is posted on GitHub (https://github.com/ashoksarabu/SpatioTemporal_CLSTM).

4-2- Analysis of Feature Maps Fusion of Two CNNs

The fusing of features maps of the two CNN is an important process that will increase the model's final performance. We use methods described in Feichtenhofer et al. (2016) [11], max fusion, concatenation fusion, conv fusion, sum fusion, to fuse two feature maps. To learn the features (fully connected) after the fusion, we use the CLSTM. The feature map after every convolution layer will have the spatial structure of the frame. We find out the best layers where the fusion of two CNN will maintain the video's spatial structure using the methods described in section 3.5. And, the results are tabulated in Table 1.

From the Table results, we conclude that fusion of the last convolution layers will give the best accuracy compared to the fusion of the former convolution layer even though both convolution layers have the same magnitude. Similarly, the convolution layer that is more distant from the fully connected layer will have less impact on the accuracy. Using of conv and sum fusion methods will get indistinguishable performance, and using sum fusion will have less computation complexity than conv fusion.

Table 1. Comparison of Performance on Split 1 of UCF-101 using ResNet-101.

Layer Number	Block Number	Layer Name	Fusion Function	Accuracy (%)
1st	2nd	Conv3_x	Conv.	69.9
2nd	2nd	Conv3_x	Conv.	70.1
1st	3rd	Conv3_x	Conv.	70.0
2nd	3rd	Conv3_x	Conv.	73.4
1st	4th	Conv3_x	Conv.	69.4
2nd	4th	Conv3_x	Sum	71.2
1st	22	Conv4_x	Sum	86.9
2nd	22	Conv4_x	Conv.	83.3
3rd	22	Conv4_x	Sum	81.3
1st	23	Conv4_x	Sum	77.0
2nd	23	Conv4_x	Sum	78.3
3rd	23	Conv4_x	Conv.	80.3
1st	1st	Conv5_x	Sum	84.3
2nd	1st	Conv5_x	Sum	87.4
3rd	1st	Conv5_x	Sum	88.9
1st	2nd	Conv5_x	Conv.	91.0
2nd	2nd	Conv5_x	Conv.	93.3
3rd	2nd	Conv5_x	Sum	93.4
1st	3rd	Conv5_x	Conv.	91.1
2nd	3rd	Conv5_x	Conv.	94.3
3rd	3rd	Conv5_x	Sum	95.4
Fully Connected	–	–	Sum	94.5

4-3- Testing

We utilize the same network parameters of the original two-stream CNN to evaluate our proposed model [3]. We sample the number of input images/frames with an equal number of intervals for both spatial and temporal streams. The number of fixed input images/ frames is initialized to 25 (for RGB images and optical flow frames). To evaluate the CNNs, we perform some operations for every image/frame; they are: cropping four corners, one center, horizontal flips. Weighted averaging is used to fuse the outputs of spatial and temporal stream CNNs. The spatial and temporal stream CNN weights are initialized to 1 and 1.5 because, when training spatial stream and temporal stream, there is a small performance gap compared to the original two-stream CNN architecture [3].

4-4- Exploration Study and Comparison with State-Of-The-Art

The presented two-stream architecture is trained on PyTorch and, overall training is implemented end-to-end using the convolutional models that are pre-trained on ImageNet [22]. The proposed model performance has shown a significant performance improvement, as mentioned in section 4.1; the combination of the two-stream model with CLSTM. The I3D [15] achieved a significant performance improvement compared to the original two-stream convolutional network using a pre-trained model on Kinetics. We still utilized the pre-trained ImageNet models, and compared to the latest deep learning architecture; we achieved 95.4% and 70.8% on HMDB-51 and UCF-101 datasets. Moreover, our experiment on the proposed model achieved better performance than the other two stream models. We outperform on the TSN [5] by 2.3% on HMDB-51 and Spatio-Temporal ResNet [13] by 2.0% on UCF-101 dataset. The proposed method, two-stream CNN with RNN (CLSTM), demonstrates spatial information integrity and its correlation with temporal information. And, spatial correlation is maintained throughout the training process. Moreover, overall comparison with state-of-art results is shown in Table 2.

Table 1. Comparison of accuracy with State-Of-The-Art Methods.

Methodology	HMDB-51	UCF-101
Two-Stream [3]	–	88.6%
C3D [2]	–	85.2%
Two-Stream + LSTM [26]	59.4%	88.0%
Spatio-Temporal ResNet [13]	–	93.4%
Two-Stream Network [11]	65.4%	92.5%
Temporal Segment Networks [5]	68.5%	94.0%
TS-LSTM [4]	69.0%	94.1%
Distinct Two-Stream [30]	67.9%	95.0%
TBRNET Encoder [31]	65.2%	92.0%
Proposed Method (CNN+ CLSTM)	70.8%	95.4%

5- Conclusion

Two-stream human action recognition for video generally uses two convolutional neural networks, one convolutional neural network for spatial stream and another convolutional neural network for the temporal stream. The Convolutional neural network with the recurrent neural network has proven a good performance for video action classification. However, these methods use a one-dimensional feature that contains damaged spatiotemporal features. The proposed architecture introduces convolution long-short term memory to the original two-stream CNN to overcome this problem, showing a significant performance improvement. Along with this, the addition of CLSTM preserves the spatial information. We explored various fusion functions to combine CNNs, and appropriate layer to integrate the spatial and temporal features. Moreover, there are some limitations to this proposed work; the solution may not work for the large videos, but there are still gaps to improve the work. For example, techniques like temporal segment networks can be implemented to preserve long-temporal dependencies for lengthy videos. In the future, we try to implement the two-stream network with TSN and use the latest pre-trained model like Kinetics to improve the accuracy furthermore.

6- Declarations

6-1- Data Availability Statement

The original contributions presented in the study, code, and datasets are provided with the GitHub link in the article; further enquiries can be directed to the corresponding author (Available online: https://github.com/ashoksarabu/SpatioTemporal_CLSTM).

6-2- Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

6-3- Conflicts of Interest

The author declares that there is no conflict of interests regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

7- References

[1] Karpathy, Andrej, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. "Large-Scale Video Classification with Convolutional Neural Networks." 2014 IEEE Conference on Computer Vision and Pattern Recognition (June 2014). doi:10.1109/cvpr.2014.223.

[2] Tran, Du, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. "Learning Spatiotemporal Features with 3D Convolutional Networks." 2015 IEEE International Conference on Computer Vision (ICCV) (December 2015). doi:10.1109/iccv.2015.510.

[3] Simonyan, Karen, and Andrew, Zisserman. "Two-Stream Convolutional Networks for Action Recognition in Videos.". In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1 (pp. 568–576). MIT Press, 2014. doi:10.5555/2968826.2968890.

[4] Ma, Chih-Yao, Min-Hung Chen, Zsolt Kira, and Ghassan AlRegib. "TS-LSTM and Temporal-Inception: Exploiting Spatiotemporal Dynamics for Activity Recognition." Signal Processing: Image Communication 71 (February 2019): 76–87. doi:10.1016/j.image.2018.09.003.

[5] Wang, Limin, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. "Temporal Segment Networks: Towards Good Practices for Deep Action Recognition." Lecture Notes in Computer Science (2016): 20–36. doi:10.1007/978-3-319-46484-8_2.

[6] Zhu, Guangming, Liang Zhang, Peiyi Shen, and Juan Song. "Multimodal Gesture Recognition Using 3-D Convolution and Convolutional LSTM." IEEE Access 5 (2017): 4517–4524. doi:10.1109/access.2017.2684186.

[7] Shi, Xingjian, Zhourong, Chen, Hao, Wang, Dit-Yan, Yeung, Wai-kin, Wong, and Wang-chun, WOO. "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting.". In Advances in Neural Information Processing Systems, Curran Associates, Inc., (2015): 802-810. doi:10.5555/2969239.2969329.

[8] Laptev, and Lindeberg. "Space-Time Interest Points." Proceedings Ninth IEEE International Conference on Computer Vision (2003). doi:10.1109/iccv.2003.1238378.

[9] Liu, Ping, Jin Wang, Mary She, and Honghai Liu. "Human Action Recognition Based on 3D SIFT and LDA Model." 2011 IEEE Workshop on Robotic Intelligence In Informationally Structured Space (April 2011). doi:10.1109/riiss.2011.5945790.

[10] Wang, Heng, and Cordelia Schmid. "Action Recognition with Improved Trajectories." 2013 IEEE International Conference on Computer Vision (December 2013). doi:10.1109/iccv.2013.441.

[11] Feichtenhofer, Christoph, Axel Pinz, and Andrew Zisserman. "Convolutional Two-Stream Network Fusion for Video Action Recognition." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016). doi:10.1109/cvpr.2016.213.

[12] Wang, Limin, Yu Qiao, and Xiaoou Tang. "Action Recognition with Trajectory-Pooled Deep-Convolutional Descriptors." 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2015). doi:10.1109/cvpr.2015.7299059.

[13] Feichtenhofer, Christoph, Axel, Pinz, and Richard P., Wildes. "Spatiotemporal Residual Networks for Video Action Recognition.". In Proceedings of the 30th International Conference on Neural Information Processing Systems (pp. 3476–3484). Curran Associates Inc., 2016.doi: 10.5555/3157382.3157486.

[14] Ji, Jingwei, Shyamal Buch, Alvaro Soto, and Juan Carlos Niebles. "End-to-End Joint Semantic Segmentation of Actors and Actions in Video." Lecture Notes in Computer Science (2018): 734–749. doi:10.1007/978-3-030-01225-0_43.

[15] Carreira, Joao, and Andrew Zisserman. "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset." 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017). doi:10.1109/cvpr.2017.502.

[16] Donahue, Jeff, Yangqing, Jia, Oriol, Vinyals, Judy, Hoffman, Ning, Zhang, Eric, Tzeng, and Trevor, Darrell. "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition.". In Proceedings of the 31st International Conference on International Conference on Machine Learning, jmlr.org - Volume 32, (2014):647-655. doi:10.5555/3044805.3044879.

[17] Srivastava, Rupesh Kumar, Klaus, Greff, and Jürgen, Schmidhuber. "Training Very Deep Networks.". In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, MIT Press, (2015): 2377-2385. doi:10.5555/2969442.2969505.

[18] Kim, Ye-Ji, Dong-Gyu Lee, and Seong-Whan Lee. "First-person activity recognition based on three-stream deep features." In 2018 18th International Conference on Control, Automation and Systems (ICCAS), pp. 297-299. IEEE, 2018.

[19] Patraucean, Viorica, Ankur Handa, and Roberto Cipolla. "Spatio-temporal video autoencoder with differentiable memory." arXiv preprint arXiv:1511.06309 (2015).

[20] Medel, Jefferson Ryan, and Andreas Savakis. "Anomaly detection in video using predictive convolutional long short-term memory networks." doi: arXiv:1612.00390 (2016).

[21] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016). doi:10.1109/cvpr.2016.90.

[22] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." Communications of the ACM 60, no. 6 (May 24, 2017): 84–90. doi:10.1145/3065386.

[23] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." In International conference on machine learning, pp. 448-456. PMLR, 2015. doi:10.5555/3045118.3045167.

[24] Brox T., Bruhn A., Papenberg N., Weickert J. (2004) High Accuracy Optical Flow Estimation Based on a Theory for Warping. In: Pajdla T., Matas J. (eds) Computer Vision - ECCV 2004. ECCV 2004. Lecture Notes in Computer Science, vol 3024. Springer, Berlin, Heidelberg. doi:10.1007/978-3-540-24673-2_3.

[25] Zach, C., T. Pock, and H. Bischof. "A Duality Based Approach for Realtime TV-L 1 Optical Flow." Pattern Recognition (n.d.): 214–223. doi:10.1007/978-3-540-74936-3_22.

[26] Donahue, Jeff, Lisa A. Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description" (November 17, 2014). doi:10.21236/ada623249.

[27] Zhenzhong Lan, Ming Lin, Xuanchong Li, Alexander G. Hauptmann, and Bhiksha Raj. "Beyond Gaussian Pyramid: Multi-Skip Feature Stacking for Action Recognition." 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2015). doi:10.1109/cvpr.2015.7298616.

[28] Kuehne, H., H. Jhuang, E. Garrote, T. Poggio, and T. Serre. "HMDB: A Large Video Database for Human Motion Recognition." 2011 International Conference on Computer Vision (November 2011). doi:10.1109/iccv.2011.6126543.

[29] Soomro, Khurram, Amir Roshan Zamir, and Mubarak Shah. "UCF101: A dataset of 101 human actions classes from videos in the wild." doi: arXiv:1212.0402 (2012).

[30] Sarabu, Ashok; Santra, Ajit K. 2020. "Distinct Two-Stream Convolutional Networks for Human Action Recognition in Videos Using Segment-Based Temporal Modeling" Data 5, no. 4: 104. doi:10.3390/data5040104.

[31] Wu, Xiao; Ji, Qingge. 2020. "TBRNet: Two-Stream BiLSTM Residual Network for Video Action Recognition" Algorithms 13, no. 7: 169. doi:10.3390/a13070169.