*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 10 Issue: 2s*
*DOI: https://doi.org/10.17762/ijritcc.v10i2s.5932*
***Article Received**: 14 October 2022 **Revised**: 21 November 2022 **Accepted**: 19 December 2022*

# Spatio-Temporal Human Action Recognition Model using Deep Learning Techniques

**[1*]Ashok Sarabu, [2*]Ajit Kumar Santra**
[1]School of Information Technology and Engineering
Vellore Institute of Technology
Vellore – 632014, Tamilnadu, India
e-mail: sarabu.ashok@gmail.com
[2]School of Information Technology and Engineering
Vellore Institute of Technology
Vellore – 632014, Tamilnadu, India
e-mail: ajitkumar@vit.ac.in
* Correspondence: sarabu.ashok@gmail.com; ajitkumar@vit.ac.in

**Abstract**— Two-stream human recognition achieved great success in the development of video action recognition using deep learning. Recently many studies have shown that two-stream action recognition is a powerful feature extractor. The main contribution in this work is to develop a two-stream model based on spatial and temporal networks using convolutional neural networks with a convolution long-short term memory. The two-stream model with ImageNet pre-trained weights is used to retrieve spatial and temporal features. Output feature maps of the two-stream model are fused using sum fusion and fed as input to convolutional long-short-term memory. SoftMax function is used to get the final classification score. To avoid overfitting, we have adopted the data augmentation techniques. Finally, we demonstrated that the proposed model performs well in comparison to state-of-the-art two-stream models with an accuracy of 96.1% on UCF 101 dataset and 70.9% accuracy on the HMDB dataset.

**Keywords**- Convolution LSTM, Two-Stream Networks, Action Recognition, Human activity.

## I. INTRODUCTION

Action recognition aims to identify objects and their motion. Human action recognition is one of the real-world active research problems in computer vision. It has an incredibly enormous number of applications like behavior analysis, video retrieval, sport and health surveillance, human-computer interaction, video indexing, smart home, video surveillance, etc.

Convolutional neural networks (CNN) for action classification in images with end-to-end learning cannot improve significantly over hand-crafted video action classification techniques. There are two major problems with using CNN for video action classification. First, structuring long-range temporal data plays a major role in understanding underlying video features [1]–[3]. Moreover, CNN architectures mainly focus on short-term features, which lack in learning long-term temporal features [4], [5]. Recently, several attempts have been made by the research community to address this issue [6], [7]. These methods mainly depend on dense temporal samples with fixed sampling intervals. When these methods are applied to lengthy videos, it will lead to high computation costs, leading to missing information in real-world applications. Second, training deep CNNs demand a tremendous amount of training data to gain optimal performance. Another reason is that the publicly available video data with annotation remains limited in diversity and size (e.g., HMDB51, UCF101).

Recently, deep CNNs attained significant improvement in image classification. Using deep CNN for image classification may overfit the video action data. These Challenges bring us to two research issues. One issue is to design a practical framework for training video representation that can preserve the spatial information in the entire model. Second, training a CNN model with limited data will not give optimal accuracy. To solve the problem mentioned earlier, we build the model based on the basic two-stream model [4].

Moreover, consecutive sets of frames in videos consist of many redundant features, which is unnecessary for temporal modeling. To overcome this, a temporal segment network is adopted [8]. This framework divides video into snippets; only one frame is used from the snippet to avoid redundant features. In this way, snippet-based temporal modeling can model long-range temporal video data.

In this work, we present a two-stream convolutional network as an improvement over the baseline two-stream model. The human visual system processes the perceived data in two different actions/processes. Sarabu et al. [9], the authors presented a two-stream model motivated by the human visual system. The proposed two-stream model is developed based on [4], [9]. Many researchers have investigated various two-stream models and demonstrated their model with high accuracy. However, the existing architectures fall short in their ability to use spatial and temporal information. To solve these issues, RNN was implemented at the end of the two-stream [4], [8], [10]. The output feature maps from both CNN models are fed into an RNN model as the input in the latest two-stream models with CNN and RNN. The RNN takes the output of the CNN as its input, transforming the resulting one-dimensional feature maps from three-dimensional feature maps [11]. Compared with the earlier work, this procedure will reduce the number of parameters and also results in a loss of spatial information. Xingjian et al. [12], presented a convolutional long-short-term memory (C − LSTM) network that uses an extended-LSTM to a third-dimension, which has shown better performance. We further enhance this technique by employing different architectures; we trained the proposed two streams model end-to-end, fused the CNNs output, and fed as input to C – LSTM. Finally, a SoftMax is applied at the end of convolution long-short-term memory.

## II. Related works

Recognition of human activity in videos has drawn a lot of interest from computer vision researchers recently. There are two broad classifications of human action recognition tasks: deep learning-based approaches and hand-crafted methods. The reader can refer to the recent research and survey papers published [13]–[15]. In this section, we discuss the most recent developments in the study of video-based human-activity recognition. Besides, we classify human action recognition techniques into one of three broad classes. 1. Methods based on 3-D convolutional networks, 2. Methods based on 2-D convolutional networks, and 3. Multiple stream methods.

### A. *Multiple stream methods*

In light of the success of LSTM in modelling sequence data, scientists are now attempting to apply it to human action recognition in videos. [18,19]. Donahue et al.[8], presented the Long-Term Recurrent Convolutional Network (LRCN) which employs a Convolutional Neural Network (CNN) to draw out the relevant spatial features, and then feeds those features into a Long Short-Term Memory network to draw out the relevant temporal information. Zhu et al. [20], implemented cost function with mixed-norm regularization of the LSTM model to use the similar features of the joints in action in the LSTM model. Veeriah et al. [21], developed a new model called differential LSTM - to discover features within temporal features by adding a new gate into LSTM. In some research papers, authors employed the LSTM to analyze the motion utilising human skeletal data. Liu et al. [22], authors used the skeleton data with adjacent joints, where body parts are divided into smaller parts. And the long short-term model is extended with the tree-based traversal approach to the Spatio-temporal streams. LSTM has not shown promising performance as it has additional parameters and variations in speech and video data [12].

### B. *Three-dimensional Convolutional Neural Network*

Qui et al. [23], presented Pseudo three-dimensional ResNet (P3D ResNet) by modifying three-dimensional convolution. They used 1*3*3 and 3*1*1 convolutional filters instead of a 3*3*3 convolution filter. Compared to the traditional three-dimensional convolutional network, this performs better for video action recognition. The benefit of P3D ResNet is that it captures the characteristics of the temporal information. Tran et al.[6], built a model with three-dimensional convolutions and three-dimensional pooling to generalize the convolution kernel into time-series domain. Both spatial and temporal convolution operations are carried out simultaneously. Even though space and time domains are considered by this method, memory and computational cost are too high. Diba et al. [24], presented a three-dimensional DenseNet-based architecture and a new temporal layer called Temporal Transition Layer (TTL) to simulate time-varying convolution kernel depth, called Temporal 3-D ConvNet. In addition, the temporal information that is crucial in video classification is downplayed by a 3D convolutional neural network, and considers only RGB images for classification.

### C. *Two-Stream Network Model*

Simonyan et al. [4], developed a two-stream model for video-based activity recognition. The two-stream model consists of two streams; one stream is used to process the spatial data, and another is used to process the temporal data. Video is decomposed into two types of data, RGB and Optical flow frames, which are feed into spatial and temporal streams. A convolutional neural network model with SoftMax scores is used for each stream. In the end, the final classification score is derived by combining the results of two streams. Wang et al. [16], introduced Trajectory-Pooled-Deep-Convolutional Descriptors, which fuse trajectory features and two stream models. This model is the best example of combining shallow local features and a convolutional neural network. Feichtenhofer et al. [17], explored better methods for combining the spatial and temporal models for two stream networks. Their findings revealed that instead of averaging the SoftMax outputs, fusing at the convolutional layer can more accurately simulate the correlation of both streams. Wang et al. [8], presented an improved Temporal Segmentation Network (TSN) performance using multi-modality input. Feichtenhofer et al. [18], fused the spatial and temporal streams with ResNets from 2-dimension to 3-dimension. To further enhance the interaction between spatial and temporal models, the residual connection is implemented in the appearance/motion stream. Wang et al. [19], introduced the spatiotemporal bilinear operation and hierarchical fusion techniques to combine spatial and temporal features and named the model a Spatio-temporal pyramid network. Ji et al. [20], presented an end-to-end learning architecture to classify and segment activities at the pixel level. Authors addressed the issue of video activity recognition using a two-stream CNN architecture and temporal information aggregation.

The video action recognition methods mentioned in this section use the two-stream model utilizing both spatial and temporal information, which is applicable to the limited training data. Researchers proposed another type of method apart from the three types mentioned above of methods. Sengupta et al. [21], presented a pillar network by fusing two ResNet and two inception networks with multiple kernel support vector machines. Zhu et al . [22],, mined the key volume in the video to improve the action recognition accuracy. Sengupta et al. [23], presented pillar networks++, which employs a Gaussian Process classifier to enhance action recognition precision. Sharma et al. [24], presented an attention model to introduce the attention function into action recognition.

## III. TECHNICAL APPROACH

Our proposed architecture using baseline two streams is shown in Figure 1: a combination of the original two-stream architecture and a recurrent neural network. First, baseline two-stream model is trained with two different CNN for spatial and temporal streams. RGB images and optical flow frames are fed into the spatial and temporal stream network. Both streams are trained on pre-trained ImageNet model. Second, both streams are combined at fully connected layers. Finally, combined feature maps of two stream are given as input to C-LSTM to retain long-term temporal dependencies.

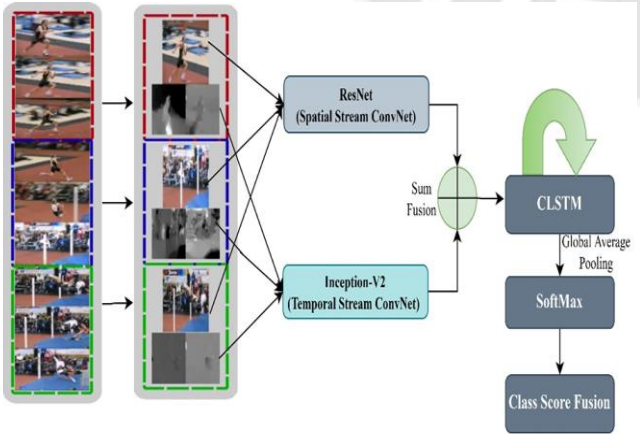

Figure 1. Overview of proposed two-stream model using CNN and C-LSTM.

### A. *Two-stream convolution networks*

A video is collection of a consecutive set of still images. The consecutive set of frames in videos consists of Spatio-temporal information. Humans' visual cortex has two distinct processes for processing incoming data: namely spatial and temporal [25]. In the spatial stream, the perceived information is used to recognize the static image and depict objects and scenes. In the temporal stream, perceived information is used to identify the mobility of objects between successive frames, which the camera observes by observing the change in object orientation.

Simonyan et al. [4], inspired by the human visual cortex, the authors developed a two-stream action recognition model for videos. Authors have developed a model that consists of two convolutional networks for processing the information called spatial and temporal. The spatial stream convolutional model only processes the single RGB frames, which are extracted from videos to identify an object. The optical flow frames are taken from the videos are processed by the temporal stream convolutional network to identify the object's motion. Consecutive Optical flow frames consist of horizontal and vertical gradient frames that feed into the temporal stream. The number of input frames for the temporal stream is fixed to L=10 [4]. The number of images is fixed as 20, because the optical flow frame is made up of both horizontal and vertical gradient images (2L=20). Then, two streams are trained individually, and their scores are combined to produce the final result. Support vector machines (SVM) and averaging are two popular fusion strategies utilized to obtain the final classification score as specified in [4].. The authors used both convolutional networks that are identical and trained independently.

This section presents a two-stream recurrent model based on the two-stream architecture [4]. In the proposed model, the two streams are trained with two deep convolutional networks and a convolutional LSTM, as shown in Figure 2. The reason for proposing the model with different CNNs is that training two streams with similar CNN models will generate redundant features. Training with similar CNNs for both streams, will result in redundant training features because optical flow images consist of horizontal and vertical gradient images generated from RGB images. Another reason is that, in HAR, motion and object recognition are two different tasks; in the same way, the proposed model is trained with two different CNNs. The reason for using C-LSTM is, to maintain spatial information in the entire model to acquire maximum accuracy. Second, we want to demonstrate that RNN outperforms the conventional two-stream model. Compared to previous experiments, we find that the proposed model provides more accurate predictions.

### B. *Base networks*

The two-stream video action recognition model has been discussed in section 3.1. A great deep learning model is one that extract the more non-redundant spatial and temporal features. [26], [27],authors have shown that discrete features can be extracted from deep CNN models. [27], the mechanism of CNN model and hidden layer features are visualized. The deep learning model's ability to retrieve the discriminant features improves by adding more layers to it. Some studies have shown that the deeper the network, the more features can be learned in lengthy videos [28], [29]. The issue of network degradation for deeper CNN models is addressed using residual networks [26]. In our proposed model, the underlying networks are Inception-V2, ResNet, and C-LSTM. The spatial features are extracted using ResNet, while Inception-V2 is adopted to extract temporal features and improve performance, and ConvLSTM is used to retain spatial information. We introduce the Inception-V2, ResNet, and ConvLSTM in the following sub-section.

### C. *Residual networks*

ResNet is used to train spatial stream in the proposed two-stream model. The reason for using deep ResNet is that it learns the discriminant features from images. He et al. [26], deep ResNet is presented as a potential solution to the issue of network degradation. Author's used f (x) = H(x) – x as the fitness function instead of vanilla fit function. Residual unit as, xp+1 = σ(xp + F(xp, wp)) where σ is the ReLU function. F(xp, wp) is the non-linear residual mapping of the weight of CNN filters wl={wl,k |1<k<K} and xp, xp+1 are the input and outputs of pth layer of the network [30]. The benefit of the residual unit is it enables the signal to be sent directly from the first layer to any layer within the network. This omits the traditional method of moving the gradient from one layer to the next. The gradient skips the intermediate layer and can travel directly from the loss to the shallow layer, solving the gradient explosion issue. With this skip connection, there is no increase in computational complexity and parameters. After each convolution and prior to the activation layer, ResNet employs batch normalization. This step increases the network convergence fast and solves the co-variate shift problem. SoftMax layer is used in conjunction with global average pooling to produce the final result. This brings down the number of parameters when compared to using the fully connected layer with the SoftMax function.

### D. *Inception-V2*

Inception-V2 is the second generation of the inception network developed by Google. Inception-V2 consists of 164 layers and can classify 1000 categories. The input size of each image is 299*299. Thus, the network can learn a wide range of images with a lot of feature representations. The most significant change that was made to Google-LeNet was the addition of the inception module, which was used in the process of building the network through the superposition of other inception modules. The introduction of the inception module increases the network performance with the increase of the depth and breadth of the network. Serge et al. [31], improvised the Google LeNet architecture called Inception-V2. In Inception-V2, 3*3 convolution is replaced with 5*5 convolution. Thus, it learns more features with the increase of more non-linear transformations and a decrease in the number of parameters. Additionally, the batch normalization layer at the end of each layer is normalized to N(0,1) distribution, which reduces the internal co-variate shift problem. Inception-V2 and ResNet are taken as base networks. Because, Inception-V2 can improve the network performance with the increase in breadth and depth, and ResNet can learn more features with the increase in the number of layers [26], [31]. Experiments are conducted to determine the best network structure for both networks to maximize the efficiency of the proposed model. The initial two-stream model used VGG-M-2048 as the base network for both spatial and temporal streams. Later [17], improved the performance of the baseline two-stream network by replacing VGG-M-2048 with VGG-16. Compared to VGGNet, ResNet has a smaller number of filters and less computational complexity. Therefore, in the proposed two-stream architecture, inception-v2 and ResNet are used to extract more temporal and spatial information. The number of FLOPs utilized by the ResNet-152, VGG-16, and VGG-19 is 11.3, 15.3, and 19.6 billion FLOPs, respectively. ResNet-101 and ResNet-50, respectively, require 7.6 and 3.8 billion floating point operations per second (FLOPs) in order to train [26]. Finally, the proposed model has 182M number of parameters. This proposed model uses a ResNet with only two dimensions, but it achieves comparable performance to that of the more common three-dimensional residual network while using fewer parameters [18].

### E. *Convolution long short-term memory*

Convolution Long Short-Term Memory is used to process the sequential images/ videos and classify the action. It is a variant of the recurrent neural network, just like an LSTM, but internal matrix multiplication is exchanged with convolution operations. In another way, the convolutional gates of the LSTM module are replaced with fully convoluted gates. Therefore, the data flow in the Conv-lstm cell keeps the input dimension in three-dimension instead of one dimension. The Conv-LSTM's gate equations:

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

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

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

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

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

Where $i_q$, $O_t$, $f_t$ is the input, output, and forget gates, σ: sigmoid function $X_1$, $X_2$, $X_3$, $X_4$ …. $X_n$: inputs, $H_1$, $H_2$, $H_3$, $H_4$ …. $H_t$: are the hidden states, C1, C2, C3, C4 …. Ct: are cell states.

$W_{x\sim}$, $W_{h\sim}$ two-dimensional convolutional kernels. Where ∗ denotes the convolution operator. The size of state-to-state and input-to-input two-dimensional kernels are fixed to 3*3 and 5*5. The size of all output states is 7*7*300. Hidden states are padded with zero padding. After the output layer of C-LSTM, a global average pooling operation is carried out. Finally, a SoftMax layer is applied to get the final output.

### F. *Pre-processing for temporal stream*

Optical flow frames are fed to the temporal stream network as input in the two-stream model. The spatial stream accepts the raw frames in the RGB format, whereas the temporal stream is fed with the optical flow frames. These frames come from the RGB images by using different techniques. TV-L1 [32] and Brox [33] are two common types of methods used to get optical flow frames from a video. The research conducted by Ma et al. [10], demonstrates that the TV-L1 method is superior to the Brox method. We employ the same method as that stated in [6], [34], which entails stacking ten optical flow frames with two channels each using the TV-L1 method to create a new frame with twenty channels. Additionally, a linear transformation is used to re-size the image vertically and horizontally to keep the frames in the range of 0 and 255. This step is performed because two streams will be fused at the end.

### G. *Fragmenting of long-range temporal data*

Designing a long-range temporal data structure is difficult in a two-stream action recognition model. The baseline two-stream model utilizes only one frame for the spatial stream network and a stack of optical flow frames for the temporal stream model [4]. Because of this, it is difficult to extract long-range temporal data efficiently. Moreover, modeling long-range temporal information plays a major role for action recognition [2], [3]. For example, in intra-class videos running and mild-walking are similar kinds of action with differences in the speed of the object. Therefore, it is difficult to classify the object's action considering a short period of time, which will lead to performance degradation. Inspired by the [4], video fragmentation is implemented in conjunction with the proposed model to improve performance. In order to use long-range temporal data fragmentation, video is divided into f fragment namely f1, f2, f3. Then we randomly sample short videos v1, v2, v3 from the respective segments, and these segments are fed to proposed model.

$$\mathrm{F}_{\mathrm{TempData}}(V_1, V_2, \ldots, V_j) = \sigma(\mathrm{Avg}(\mathcal{F}(V_1; \mathbf{W}), \mathcal{F}(V_2; \mathbf{W}), \ldots\ldots, \mathcal{F}(V_j; \mathbf{W}))) \quad (6)$$

Where σ is SoftMax function, Avg represents an averaging function. F (T1; W) is a convolutional function with parameter W. These short videos are given as input to the proposed two-stream model to get action classification score. After, this value is fused with the average function to get the final decision among snippets. The final loss of the consensus category is

$$\mathcal{L}(y, Avg) = -\sum_{i=1}^{j} y_i \left( Avg_i - \log \sum_{j=1}^{j} \exp(Avg_j) \right) \quad (7)$$

Where yi is ground truth label of class i, j is number of action categories. All the fragments are used to optimize the network parameter W. In the back propagation w w.r.t loss value L is

$$\frac{\delta \mathcal{L}(y, Avg)}{\delta \boldsymbol{W}} = \frac{\delta \mathcal{L}}{\delta Avg} \sum_{k=1}^{3} \frac{\delta Avg}{\delta \mathcal{F}(V_k)} \frac{\delta \mathcal{F}(V_k)}{\delta \boldsymbol{W}} \quad (8)$$

Subsequently, to train the model stochastic gradient descent is used. As in Equation (3), fragmentation category Avg for three short video guarantees that the model parameters are updated. Thus, with this long-range fragmentation technique preserves the long-range temporal information and parameter are trained in entire video.

### H. *Two-stream fusion*

In [19], authors employed different fusion strategies to combine the two feature maps of two CNN in a two-stream convolution neural network model. Fusion of layers can affect the accuracy of prediction. There are four kinds of fusion techniques Conv, concatenation, max, and sum fusion. On the UCF-101 dataset, the authors of the same paper also demonstrated that max fusion has the worst performance and conv fusion has the best performance. We use sum fusion to fuse two CNN, since the sum fusion requires fewer parameters to compute and is approximately as efficient as the Conv fusion. For time t, we combine two feature maps $X_t^p, X_t^q$ to an output $y_t$, where $x_t^p, x_t^q \varepsilon R^{Hi \times Wi \times Ch}$ and $y_t \varepsilon R^{Hi' \times Wi' \times Ch'}$. Where $C_h$, $H_i$, $W_i$ represent the number of channels, height and width of the respective feature maps.

Sum fusion: $y^{summation} = f^{summation}(x^p, x^q)$ calculates the sum of the value of two point at the same location i; j, and c in spatial and temporal feature maps. The new value at point (i, j, d) $y_{i,j,d}^{\mathrm{summation}} = X_{i,j,c}^a + X_{i,j,c}^b$.

***Article Received**: 14 October 2022 **Revised**: 21 November 2022 **Accepted**: 19 December 2022*

In the proposed two-stream model, sum fusion is employed to fuse the two CNN and combine them at the fully connected layer with the accuracy of 96.1% on UCF 101 and 70.9% on the HMDB dataset.

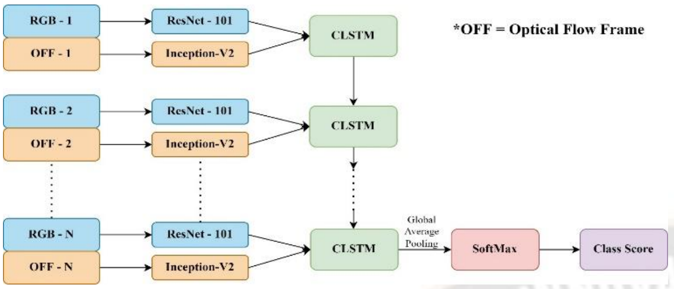

Figure 2. Spatio-Temporal Human Action Recognition using CNN and C-LSTM.

### I. *Data augmentation*

Data augmentation techniques are used when the dataset size is inadequate and to increase the diversity of the samples. Horizontal flipping, scale jittering, and corner cropping are used to increase the size of the dataset. The height and width of the cropped area were chosen randomly 168, 192, 224, 256. Scale jittering is used to set the size of the image to 256*340 (optical flow). In corner cropping, we extract the image's center or corner, thus to avoid the center of image as the default focus. Horizontal flipping is done randomly in all the training steps.

## IV. Implementation Details

### A. *Dataset*

The performance of the proposed model is evaluated using the two challenging human action recognition datasets: HMDB51 and UCF101.These two datasets include a number of complex actions and numerous uncontrolled scene changes. There are a total of 6849 videos in the HMDB51 dataset, which have been labelled with 51 different types of actions. HMDB51 is a curated dataset consisting of videos collected from YouTube and Google. The UCF101 contains 1,3320 action videos grouped into 101 action categories. Each clip comprises 100 to 300 frames with 3 to 10 seconds of video length.

### B. *Training*

The model is trained using a mini-batch stochastic gradient descent algorithm. The ImageNet pre-trained weights are used to initialize the network parameters [30]. The momentum, weight decay, and batch size values are initialized to $9 * 10^{-1}$, $5 * 10^{-4}$, 256. Initially, spatial and temporal networks are initialized with a learning rate of 10−4. The spatial stream network's learning rate is reduced by ten times after every 15K iterations, and the complete training process ends after 36K iterations. The initial learning rate of temporal stream is set to $10^{-4}$. When training temporal stream, learning is reduced to ten times after 20k and 32k iterations. The maximum iterations are set to 40k. TV-L1 algorithm is employed to retrieve optical flow images from videos. An ADAM optimizer is used to train the model for convolution LSTM. And, the learning rate of convolution long short-term memory is set to 0.00005. On the Pytorch platform, we use data parallelization to speed-up the training process with multiple GPUs.

### C. *Testing*

To test the proposed two-stream model, we followed the original two-stream convolution network scheme. 25 frames (Optical, RGB) are stacked are sampled at equal intervals of time. Every sample consists of 10 input images obtained by horizontal flip, center cropping, and four corner cropping. The spatial and temporal streams of the proposed model are combined using the sum fusion function. Sum fusion is selected as the fusion method because it has fewer computer parameters and performance is good compared to the conv fusion method.

### D. *Exploration Study*

The proposed two-stream CNN with RNN model is trained on PyTorch. Pre-trained ImageNet weights are used to train the model end to end [30]. The proposed model shows a significant improvement in performance with the combination of two different CNN and convolutional long short-term. The experiments are carried-out on the proposed CNN model using similar and heterogenous CNN models. Sarabu et al. [9], authors showed significant performance improvement using ResNet-101 and Inception-V2. Based on the results in [9], ResNet-101 and inception-V2 are adopted as the base CNN for the proposed model. The proposed model performance is improved with the training strategy as described in section 4.2.

The I3D [35], showed an improvement in accuracy compared to the baseline two-stream CNN model on the pretrained Kinetics dataset. We still used the ImageNet pretrained model parameters in the proposed model achieved best performance, and compared them to the latest two-stream deep learning models. The proposed model achieved 96.1% and 70.9% on UCF101 and HMDB51 datasets. Additionally, the proposed model outperformed the other two-stream models in terms of performance. The proposed model outperforms the TSN [8] by 2.4% on HMDB51 and Spatio-Temporal ResNet [17] by the 2.7% percentage on UCF101 dataset. The proposed model demonstrates the integrity and correlation between spatial information and temporal information. Throughout the training process, spatial correlation is preserved. The accuracy

*DOI: https://doi.org/10.17762/ijritcc.v10i2s.5932*
***Article Received**: 14 October 2022 **Revised**: 21 November 2022 **Accepted**: 19 December 2022*

of the proposed model with state-of-art methods is shown in Table 1.

TABLE I. COMPARISON OF ACCURACY WITH STATE-OF-THE-ART METHODS

| **Methodology** | **HMDB51** | **UCF101** |
|---|---|---|
| Two-stream [4] | - | 88.6% |
| C3D [5] | - | 85.2% |
| Two-stream+ LSTM [6] | 59.4% | 88.0% |
| TSN [8] | 68.5% | 94.0% |
| Spatio-Temporal ResNet [18] | - | 93.4% |
| TS-LSTM [10] | 69.0% | 94.1% |
| Distinct two-stream [9] | 67.9% | 95.0% |
| Two-tream+Conv_LSTM [36] | 70.8% | 95.4% |
| TBRNet Encoder[37] | 65.2% | 92.0% |
| The proposed model (CNN+RNN) | 70.9% | 96.1% |

## V. CONCLUSIONS

Human action recognition model for video based on two-stream is proposed with two different CNNs, and C-LSTM. Humans' eye processes the visual data as two individual tasks (object detection and motion recognition). Inspired from this, we proposed a two-stream model with two different convolutional neural networks. However, two-stream models uses a one-dimensional feature that damages spatiotemporal features. RNN is used to overcome the damage to the spatial-temporal features. So, convolutional long-short term memory is used after two convolutional neural networks; as the recurrent neural network has proven with good performance. In our experiments, we found that ResNet-101 and Inception-V2 model with C-LSTM, when employed as network models for a two-stream network with fragmenting long-range temporal modeling, yield the best performance. To further enhance performance even with lengthy videos, we plan to implement the proposed model with the temporal segment network technique, using a pre-trained model such as Kinetics.

## REFERENCES

[1]. A. Gaidon, Z. Harchaoui, and C. Schmid, "Temporal localization of actions with actoms," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2782–2795, 2013.

[2]. J. C. Niebles, C.-W. Chen, and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *European conference on computer vision*, 2010, pp. 392–405.

[3]. L. Wang, Y. Qiao, and X. Tang, "Latent hierarchical model of temporal structure for complex activity classification," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 810–822, 2013.

[4]. K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, Cambridge, MA, USA, 2014, pp. 568–576.

[5]. D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[6]. J. Donahue *et al.*, "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 677–691, Apr. 2017, doi: 10.1109/TPAMI.2016.2599174.

[7]. G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1510–1517, 2017.

[8]. L. Wang *et al.*, "Temporal segment networks: Towards good practices for deep action recognition," in *European conference on computer vision*, 2016, pp. 20–36.

[9]. A. Sarabu and A. K. Santra, "Distinct two-stream convolutional networks for human action recognition in videos using segment-based temporal modeling," *Data*, vol. 5, no. 4, p. 104, 2020.

[10]. C.-Y. Ma, M.-H. Chen, Z. Kira, and G. AlRegib, "TS-LSTM and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition," *Signal Processing: Image Communication*, vol. 71, pp. 76–87, 2019.

[11]. 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.

[12]. X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," *Advances in neural information processing systems*, vol. 28, 2015.

[13]. G. Cheng, Y. Wan, A. N. Saudagar, K. Namuduri, and B. P. Buckles, "Advances in human action recognition: A survey," *arXiv preprint arXiv:1501.05964*, 2015.

[14]. S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey," *Image and vision computing*, vol. 60, pp. 4–21, 2017.

[15]. H.-B. Zhang *et al.*, "A comprehensive survey of vision-based human action recognition methods," *Sensors*, vol. 19, no. 5, p. 1005, 2019.

[16]. L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4305–4314.

[17]. C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1933–1941.

[18]. R. Christoph and F. A. Pinz, "Spatiotemporal residual networks for video action recognition," *Advances in neural information processing systems*, pp. 3468–3476, 2016.

[19]. Y. Wang, M. Long, J. Wang, and P. S. Yu, "Spatiotemporal pyramid network for video action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 1529–1538.

[20]. J. Ji, S. Buch, A. Soto, and J. C. Niebles, "End-to-end joint semantic segmentation of actors and actions in video," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 702–717.

[21]. B. Sengupta and Y. Qian, "Pillar Networks for action recognition," in *IROS Workshop on Semantic Policy and Action Representations for Autonomous Robots*, 2017.

[22]. W. Zhu, J. Hu, G. Sun, X. Cao, and Y. Qiao, "A key volume mining deep framework for action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1991–1999.

[23]. B. Sengupta and Y. Qian, "Pillar networks++: Distributed non-parametric deep and wide networks," *arXiv preprint arXiv:1708.06250*, 2017.

[24]. S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," *arXiv preprint arXiv:1511.04119*, 2015.

[25]. M. A. Goodale and A. D. Milner, "Separate visual pathways for perception and action," *Trends in neurosciences*, vol. 15, no. 1, pp. 20–25, 1992.

[26]. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[27]. W. Yu, K. Yang, Y. Bai, T. Xiao, H. Yao, and Y. Rui, "Visualizing and comparing AlexNet and VGG using deconvolutional layers," in *Proceedings of the 33 rd International Conference on Machine Learning*, 2016.

[28]. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[29]. C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[30]. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[31]. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015, pp. 448–456.

[32]. C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime tv-l 1 optical flow," in *Joint pattern recognition symposium*, 2007, pp. 214–223.

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

[34]. J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4694–4702.

[35]. J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[36]. A. Sarabu and A. K. Santra, "Human action recognition in videos using convolution long short-term memory network with spatio-temporal networks," *Emerging Science Journal*, vol. 5, no. 1, pp. 25–33, 2021.

[37]. X. Wu and Q. Ji, "TBRNet: Two-stream BiLSTM residual network for video action recognition," *Algorithms*, vol. 13, no. 7, p. 169, 2020.