Computer & Robotics

Human Action Recognition using Convolutional LSTM with Three-Time Variables

Arash Asefnejad [a], Javad Mohammadzadeh [b],* , Mitra Mirzarezaee [a]

[a] *Department of Computer Engineering, Science and Research branch, Islamic Azad University, Tehran, Iran*

[b] *Department of Computer Engineering, Karaj Branch, Islamic Azad University, Karaj, Iran*

Received 11 October 2023; Accepted 02 May 2024

Abstract

With the appearance of deep neural networks, and at the head of it, convolutional neural networks, a great revolution in machine vision was created. Also, the growth of video data and the need for automated processing of this data type have made deep neural network usage increasingly important. There are several methods to recognize the type of movement in the videos. One of the methods is using LSTM and a convolutional neural network in order to extract the time dependencies from video images more accurately. In this study, we present an extended version of the LSTM that can learn longer temporal dependencies. Besides the convolutional neural network, our extended version of the LSTM forms a strong structure to recognize human activity. The results of this study on data set UCF 101 and HMDB51 show that the presented architecture, with a performance accuracy of 96.28 on data set UCF101 and 78.02 on data set HMDB51, performs better than the most similar methods.

Keywords: Action recognition, deep neural networks, LSTM, CNN

1.Introduction

With the increase in the number and volume of digital images and also the increase in the number of cameras, video image processing has become one of the principal challenges in the field of machine vision. In this regard, automatic action detection and recognition in video images, considering their comprehensive application, has become one of the principal areas of research in this field [1]. Action data has two features: temporal and spatial. If we investigate the action in the video, we will find that the action has long-term dependencies that increase with the complexity of the type of action [2]. According to what has been said, one of the challenges in this field is the complexity of the presented neural network, which can hardly solve the problems related to the temporal dependence of movement [3].

Regarding the recognition of the type of actions in the video, 3 methods are possible: The first one is the method that tries to recognize the type of movement after extracting spatial features from video images and adding movement features [4]. For example, the 3D convolutional neural network is a type of network that tries to extract temporal features by adding a third dimension [5]. The mentioned method has many problems the most important of which are:

1- The number of learning parameters and also determining the number of input parameters that the training must be done based on them [6].

2- The number of frames that must be selected from each video to recognize the action according to them [7].

The second method is those that use two video streams for action recognition [8]. These streams are as below:

1- Video frames that are used to recognize the spatial situation.

2-Optical Flow

* Corresponding Author E-mail: j.mohammadzadeh@kiau.ac.ir

It should be noted that despite the second method being more suitable for action recognition, they are so time-consuming. Third methods: in these methods, long-term temporal dependencies are used for action recognition. LSTM neural networks are used in this field [9]. These types of networks are able to process temporal data and they can provide temporal data processing by using their recurrent connection. The architecture of these types of networks contains several layers of convolutional networks that are connected to each other and finally, their output is given to an LSTM network [10].

After the revision of the past method, it can be said that it is very complex to design a system for recognizing the action in video and it has many challenges [11]. These challenges are:

1-Various types of activities: For example, the size of a person's body and even age can affect the action's performance type. So that method is suitable which ignores the differences between various instances of a category[12].

2-Environmental settings and data recording: The method of environmental setting is another challenge to recognize the type of human activity. For example, if we consider dynamic environments and try to recognize human activity in such environments, challenges like locating humans in dynamic and mixed environments, covering some parts of the human body, lighting conditions, observing an action from different angles of view, Background changes, movement, and camera changes will make it difficult to human action recognition [34].

3-Variety of speed when performing different actions: Because many activities are carried out at different speeds [35].

4-activities overlap: Some activities overlap with each other in terms of the stages of performing the movement, which makes it difficult to identify the movement [16]. This problem exists especially in action recognition methods where shorter motion sequences are considered for recognition [43].

5-Lack of Sufficient Data: Deep learning models require large amounts of labeled data for training. Acquiring a comprehensive dataset of human movements can be challenging, particularly when considering diverse poses, lighting conditions, and backgrounds.

6-Spatial and Temporal Dependencies: Understanding the temporal dependencies between different frames in a video sequence, along with spatial relationships between body parts, is crucial for accurate movement recognition. CNNs might struggle with capturing long-term temporal dependencies, while LSTMs may face challenges in modeling complex spatial relationships.

7-Overfitting and Generalization: Deep learning models, especially when dealing with complex datasets, may suffer from overfitting, where the model becomes too specific to the training data and fails to generalize to unseen data.

8- Computational Complexity: Implementing CNN-LSTM models for human movement recognition requires significant computational resources. Training and inference times can be substantial, particularly when dealing with high-resolution video data.

9- Data Augmentation and Preprocessing: Preprocessing video data to ensure consistency, removing noise, and augmenting data to account for various environmental conditions and variations in human movement can be a challenging task, requiring careful consideration and domain expertise.

10- Model Interpretability: CNN-LSTM models are often considered black boxes, making it difficult to interpret how and why the model makes specific predictions. Interpreting these models is critical for ensuring their reliability and trustworthiness in practical applications.

hence, the presented method must overcome all the mentioned challenges and recognize the type of action, correctly. It should be noted that there are three different forms of information in video data [36-38], which are as below:

1- spatial information

2-Action Information

3-Temporal dependence of this information on each other

The remainder of this paper is organized as follows: in section 2, the background and related work of deep learning methods that were used for human action recognition are reviewed. In section 3, our extended version of the LSTM (convolutional LSTM with three-time variables) is described. In section 4, our network architecture is presented. we present

experimental results in section 5. Finally, the paper is concluded in section 6.

2. Background and Related Work

Due to the widespread use of human activity recognition in video, including robotics, emotion analysis, and video control, this field of machine vision has received much attention today.

As initial research to recognize human activity in which CNNs were used, it could be mentioned to [40]. In this article, the purpose is to build a model in which some of the capabilities known in the human visual system are used to solve the problem of human activity recognition.

For this purpose, the human visual system is completely described and finally, the proposed method was a kind of CNN. In [41], to recognize the human activity, a modified model of the CNN is used and by using 3D input, transmission-independent features were extracted. Also, the weighted network with minimum/maximum fuzzy was used for classification. In [27] a convolutional Restricted Boltzmann machine network was used to learn unsupervised spatial-temporal features. The obtained features were given to the CNN to recognize the activity. The first problem of this method is the large number of parameters of this network. Also, this method does not consider the input image forms. In [29], in order to detect human activity, 3D CNN was used to extract the spatial and temporal features of the video simultaneously and to consider motion information. In [28,42], two separate streams of information are used to train CNNs. In these architectures, spatial and temporal information was extracted respectively by using RGB images and Optical Flow data and used to identify human activity. One of these methods is [43], which has a high accuracy of recognition by using two-stream ConvNets. The features obtained from different layers of the deep neural network were used to identify the type of action. Finally, the features obtained make a final features vector. Although using Optical Flow in action recognition is very time-consuming and increases the computational load, it can be said that the results obtained from these methods are very acceptable. The following, methods [41,45,46] used CNN and LSTM networks to recognize human action. In these methods, spatial features are extracted by using CNN, and temporal features are extracted by using LSTM networks. In [15] the product operator was replaced by the convolutional operator to cover the spatial information of the films and created LSTM Based on attention(ALSTM). The network was called videoLSTM. In [47] Conv-LSTM was integrated with spatial attention to encoding the sequential convolutional features with the spatial layout. The main defect of these methods is neglecting critical temporal cues. In [48]. critical frames along the timeline were highlighted by temporal attentive LSTM units. In [18] spatial and temporal attention was created by using 3D CNN and bidirectional LSTM. In [19] it was attempted to integrate spatial and temporal attention and LSTM was trained. [1]Proposes an end-to-end learning framework for action detection, leveraging both spatial and temporal information through CNN and LSTM. [2] Introduces a Temporal Segment Network (TSN) that uses both CNN and LSTM to model long-range temporal dependencies for action recognition. [3] Introduces a Spatial-Temporal Graph Convolutional Network (ST-GCN) to model spatial dependencies and temporal dynamics in human motion, achieving state-of-the-art results in skeleton-based action recognition. [4] explores the use of 3D convolutional neural networks (C3D) for video classification tasks, demonstrating the effectiveness of combining CNNs with temporal information. [5] Introduces the Two-Stream Inflated 3D ConvNet (I3D) architecture, which inflates 2D CNNs to 3D for spatiotemporal action recognition. It also introduces the Kinetics dataset. [6] Proposes SlowFast networks, a two-stream network with different frame rates for spatial and temporal processing, achieving state-of-the-art performance in video recognition. [7] Introduces a Dynamic Refinement Network (DRN) for object detection in videos, employing both temporal and spatial information for improved accuracy. [8] Proposes Non-Local Neural Networks, addressing long-range

dependencies in video understanding by capturing non-local interactions between different spatial and temporal positions. [9] Introduces a collaborative temporal and spatial modeling approach for video object segmentation, combining LSTM and CNN to enhance spatiotemporal reasoning.[10] Proposes a multi-stage CNN architecture for temporal action localization in untrimmed videos, utilizing LSTM for handling temporal dependencies. [11] Introduces TALL, a method for temporal activity localization in videos using natural language queries, incorporating LSTM for understanding temporal relationships. [12] Proposes Temporal Segment Networks (TSN) for action recognition, combining spatial and temporal information through the integration of CNN and LSTM. [13] Introduces Structured Segment Networks for temporal action detection, utilizing LSTM for structured temporal modeling. [14] Proposes Aggregated Residual Transformations for deep neural networks, enhancing the performance of CNNs in capturing spatial features for human motion. [15] Proposes Graph Convolutional Networks (GCNs) for temporal action localization, leveraging both spatial and temporal relationships between video frames. [16] Introduces the Tube Convolutional Neural Network (T-CNN) for action detection in videos, combining spatial and temporal features through the integration of CNN and LSTM. [17] Proposes a method for learning temporal action proposals with fewer labels, incorporating LSTM for improved temporal understanding. [18] Introduces a Semantic Proposal approach for activity localization in videos via sentence queries, leveraging LSTM for understanding the temporal context of sentences. [51]Proposes an end-to-end learning framework for action detection, leveraging both spatial and temporal information through CNN and LSTM. [52] Introduces a Temporal Segment Network (TSN) that uses both CNN and LSTM to model long-range temporal dependencies for action recognition. [11] Introduces a Spatial-Temporal Graph Convolutional Network (ST-GCN) to model spatial dependencies and temporal dynamics in human motion, achieving state-of-the-art results in skeleton-based action recognition. [53] explores the use of 3D convolutional neural networks (C3D) for video classification tasks, demonstrating the effectiveness of combining CNNs with temporal information. [54] Introduces the Two-Stream Inflated 3D ConvNet (I3D) architecture, which inflates 2D CNNs to 3D for spatiotemporal action recognition. It also introduces the Kinetics dataset. [57] Proposes SlowFast networks, a two-stream network with different frame rates for spatial and temporal processing, achieving state-of-the-art performance in video recognition. [55] Introduces a Dynamic Refinement Network (DRN) for object detection in videos, employing both temporal and spatial information for improved accuracy. [56] Proposes Non-Local Neural Networks, addressing long-range dependencies in video understanding by capturing non-local interactions between different spatial and temporal positions. [58] Introduces a collaborative temporal and spatial modeling approach for video object segmentation, combining LSTM and CNN to enhance spatiotemporal reasoning.[59] Proposes a multi-stage CNN architecture for temporal action localization in untrimmed videos, utilizing LSTM for handling temporal dependencies. [60] Introduces TALL, a method for temporal activity localization in videos using natural language queries, incorporating LSTM for understanding temporal relationships. [61] Proposes Temporal Segment Networks (TSN) for action recognition, combining spatial and temporal information through the integration of CNN and LSTM. [62] Introduces Structured Segment Networks for temporal action detection, utilizing LSTM for structured temporal modeling. [63] Proposes Aggregated Residual Transformations for deep neural networks, enhancing the performance of CNNs in capturing spatial features for human motion. [64] Proposes Graph Convolutional Networks (GCNs) for temporal action localization, leveraging both spatial and temporal relationships between video frames. [65] Introduces the Tube Convolutional Neural Network (T-CNN) for action detection in videos, combining spatial and temporal features through

the integration of CNN and LSTM. [66] Proposes a method for learning temporal action proposals with fewer labels, incorporating LSTM for improved temporal understanding. [67] Introduces a Semantic Proposal approach for activity localization in videos via sentence queries, leveraging LSTM for understanding the temporal context of sentences.

The main problem of all the mentioned methods is the failure to recognize action in complicated movements and long sequences. The proposed method, which will be studied below, tries to solve this problem.

3. Convolutional LSTM with Three-Time Variables

Typically, most of the methods used to recognize the action use these three different forms of information [39]. In the proposed architecture, we have serialized the different parts of the network so that they can extract the information mentioned above. In this architecture, convolutional networks extract spatial features of the image, and LSTM networks extract action features and long-term temporal dependencies in the video. The proposed architecture is shown in Figure 1.

In this architecture, firstly, video frames are given to 3 towers of convolutional networks as input. The mentioned networks work with each other in parallel form and share their weights with each other, as well. Convolutional networks extract spatial features by convolving different filters in convolutional layers and crossing them through pooling layers. These features finally will concatenate in the last layer and make a 2D matrix to be proposed to a layer of the LSTM units as input. In other words, the different video images that are considered as input include different frames that are given separately to the convolution towers. Finally, their outputs concatenate each other and make a feature vector. In this article, we modify the LSTM units in a way that against the previous methods that extracted action information based on optical stream data, our method extracts action information and time dependence in video frames together. Finally, the proposed LSTM output is given to a Softmax and the type of action is recognized based on the labels assigned.

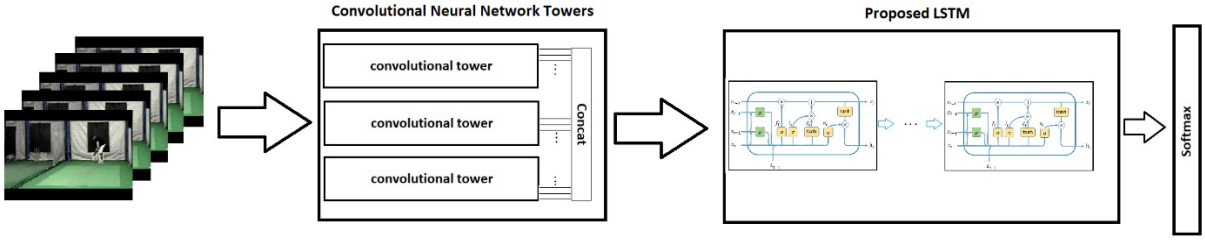

Fig.1 Proposed architecture to recognize the action

Briefly, it can be said that network inputs in the proposed architecture (which are the same as network frames) will be given to parallel convolutional networks so that the outputs of these networks will be integrated with each other. In this step, the information of the spatial features of each frame is extracted. It can be noted that the outputs of convolutional networks also contain temporal information. In the next step, the features extracted from the
previous step are given as the input of the proposed LSTM network, in order to extracts temporal dependencies and the action information together. The proposed LSTM units have been modified in a way that extract motion information in addition to temporal information.

Figure 2 shows the proposed LSTM unit. In the designed unit, x_t, x_{t-1} and x_{t-2} are considered as LSTM inputs. According to formula 1, first the correlation between the two matrices x_t and x_{t-1}, then x_{t-1} and x_{t-2} are calculated so that these matrices are added together and finally are normalized.

Therefore, according to what has been said, we can say that the proposed LSTM unit is different

from the basic LSTM unit in two various aspects. The first difference between this unit and the basic LSTM unit is that instead of the multiplication operator, the convolution operator is used to extract the action information in addition to the temporal information. In this architecture, the input gates and weights are 2D arrays. Also, inputs x_{t-1} and x_{t-2}, which are related to times $t-1$ and $t-2$ are considered so that the network can learn longer temporal dependency.

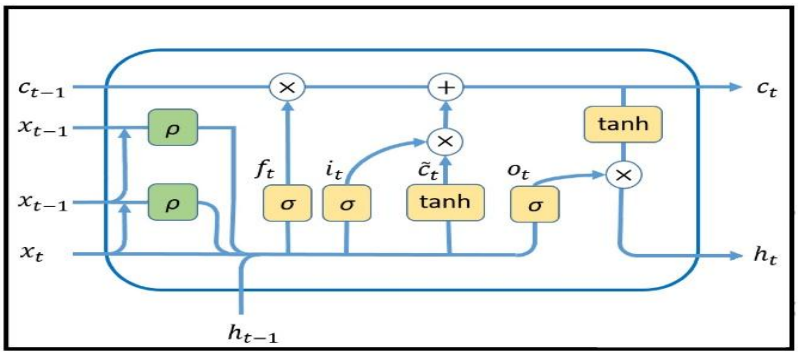

Fig.2 proposed LSTM unit

(1)

$$C_{R_t} = x_{t-1} \otimes x_t$$

$$C_{R_{t-1}} = x_{t-1} \otimes x_{t-2}$$

$$\bar{C}_{R_t} = (C_{R_t} + C_{R_{t-1}}) / 2$$

$$I_t = \delta(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} * \bar{C}_{R_t} + b_i)$$

$$F_t = \delta(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} * \bar{C}_{R_t} + b_f)$$

$$O_t = \delta(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} * \bar{C}_{R_t} + b_i)$$

$$\bar{C}_t = \delta(W_{xc} * X_t + W_{hc} * H_{t-1} + W_{cc} * \bar{C}_{R_t} + b_i)$$

$$C_t = \delta(F_t \odot C_{t-1} + I_t \odot \bar{C}_t)$$

In formula 1, operator $\otimes$ is batch-wise convolution. * is convolution operator. $\odot$ is dot operator. $\boldsymbol{C}_{\boldsymbol{R}_t}$ is Correlation matrix at time t, $\boldsymbol{C}_{\boldsymbol{R}_{t-1}}$ is a Correlation matrix at time $t-1$. $\bar{\boldsymbol{C}}_{\boldsymbol{R}_t}$ is The result of the average of the addition and normalization of correlation matrices at times t and $t-1$.

As we mentioned above, we use operator batch-wise convolution. Look at the formula number 2 carefully. In this formula, $\otimes$ is cross-correlation operator.

cross-correlation is an integer. This number can be used to compare two grayscale images. Assume we use a camera to record the information. Many images retrieved from the camera are duplicates, and although no changes have been made to the images taken from the scene, the images are different because of the changes in intensity. In other words, if we compare the images pixel by pixel, the images will be different from each other. On the other side, images may have different sizes and we do not intend to compare all the images with each other. Therefore, to compare two images, firstly the similar parts of them (spatially) can be considered and then calculate the cross-correlation of those parts. If we assume that we have two pieces $(m+1)\times(m+1)$ of two images with the centers (c_{11}, c_{12}) and (c_{21}, c_{22}) so that to calculate cross-correlation, formula 2 is used:

$$D(I_1, I_2, c_{11}, c_{12}, c_{21}, c_{22}) = \sum_{i=-n}^{n} \sum_{j=-n}^{n} (I_1(c_{11}+i, c_{12}+j) - I_2(c_{21}+i, c_{22}+j))^2$$

$$\overline{\mathrm{Im}\, g}(u,v,n) = \frac{1}{(2n+1)^2} \sum_{i=-n}^{n} \sum_{j=-n}^{n} (I\ (u+i, v+j))^2$$

$$\sigma(u,v,n) = \sqrt{\frac{1}{(2n+1)^2} \sum_{i=-n}^{n} \sum_{j=-n}^{n} (I\ (u+i, v+j) - \overline{\mathrm{Im}\, g}(u,v,n))^2}$$

$$ZND(I_1, I_2, c_{11}, c_{12}, c_{21}, c_{22}) = \frac{\frac{1}{(2n+1)^2} \sum_{i=-n}^{n} \sum_{j=-n}^{n} \prod_{p=1}^{2} (I\ (u_p+i, v_p+j) - \overline{\mathrm{Im}\, g}(u_p, v_p, n))^2}{\sigma_1(u_1, v_1, n) \times \sigma(u_2, v_2, n)} \quad (2)$$

The values obtained are in the range [0,1]. In this article, we use formula number 3 to calculate the 3D cross-correlation matrix:

$$C = A \otimes B$$

$$C(x, y, Z) = A_z \circ B(x, y) = \sum_{j=-N}^{N} \sum_{i=-N}^{N} \frac{1}{\sigma_{A_z}} \frac{1}{\sigma_B} (A_z(i, j) - \bar{A_z})(B(x+i, y+j) - \bar{B}) \quad (3)$$

So that:

$$A_z = \{A(i, j) \mid (i, j) \in \{0, ..., w\} \times \{0, ..., h\}, z = ([\frac{i}{p}] + [\frac{i}{p}], [\frac{w}{p}], p = 2N+1)\}$$

In Formula number 3, $\boldsymbol{A}$ is a matrix $L \times Q$. Here, the operator cross-correlation is used to obtain the output similarity of CNN network in two sequential times. In fact, cross-correlation is a method for tracking the data changes in two or several times series that compares the data relationship with each other and considers the compatibility between this data, and finds the best match point between time series. In action recognition, the existence of long temporal dependencies in sequential frames is extractable by the mentioned operator.

To calculate the 3D matrix $\boldsymbol{C}$, first matrix $\boldsymbol{A}$ is broken into smaller patches, and each patch is slid around the input to obtain a location that has high overlap. The patches are square with $2N+1$ size. For example, if we consider $N=1$, the size of the patches will be 3×3. To clarify the issue, assume that the size of $\boldsymbol{A}$ is 9×12 (means that $L=9$ and $Q=12$). $\boldsymbol{A_z}$ is each patch obtained from $\boldsymbol{A}$ Which z is a value from 0 to $Q-1$. If we consider the size $N=1$, then the size will be $P=3$. For patches with size 3 ($P_z=3$) the value of i will be {3,4,5} and the value of j will be {0,1,2}. In this formula, σ_{A_z} is a standard deviation of the patch $\boldsymbol{A_z}$ and $\bar{\boldsymbol{A}}_{\boldsymbol{z}}$ is the average of $\boldsymbol{A_z}$. Similarly, σ_B and $\bar{\boldsymbol{B}}$ are standard deviation and the average of $\boldsymbol{B}$ which are used to obtain and calculate cross-correlation. It can be said that the cross-correlation operator is used to calculate the temporal dependence and the convolution operator is used to obtain spatial features in the LSTM structure. It should be noted that the

proposed structure is very compatible with the variations of a particular type of action and long-term temporal dependencies in the video. Wherever the cross-correlation between the two inputs increases, it shows an increase in similarity between the inputs. Therefore, the mentioned method will be suitable for tracking a patch in sequential frames of a video, and its value can have a significant impact on the input gates of a basic LSTM such as T_l, F_t, O_t, C_t and H_t, which in order are: input gates, forgetting, outputs and memory unit.

4.Network Architecture

The output of the avg_pool layer of the ImageNet network is used as a convolution tower in implementation. As mentioned above, the duty of the mentioned network is to extract spatial features from the input image frames. The output from the avg_pool layer is used as the input of the proposed LSTM. From the input, 40-frame will be selected as the sample of the network inputs. Also, in order to avoid overfitting and to be able to consider different movement speeds and periods, the video starting points are considered random and the number of frames is obtained with different steps (between 40 and 50 frames). Also, videos with less than 40 frames and more than 360 frames are excluded. This is because no acceptable information can be extracted from the frames related to these videos. Also in the proposed LSTM structure, the size of the patches is selected 3×3 considering $N=1$, and to make the calculating easy, the size of the convolution filter is considered equal to the size of the patches. We used Data Augmentation and dropout techniques to reduce overfitting. Also, horizontal rotations are done in the frames to prevent overfitting, and some cuts have been made in some of the frames.

If we consider the number of video frames as n and assume that the size of each frame is $w\times h$, then n' is the number of frames that are selected from the frame of the original image. Also, the size of these selected frames changes with the size of $w'\times h'$ after normalization operation. So we have:

$$40 \le n' \le 50 \qquad AND \qquad n' \le n$$

On the other hand, if we consider f as a set of the selected frames from the original video, so that:

$$f = \{ f_i \mid i = r + s, r \in \{1,2,...,n+1\}, s = \left\lfloor \frac{n}{n'} \right\rfloor, 40 \le n' \le 50 \} \quad (4)$$

In Formula number 4, the number of selected frames from the original video is i and r is a random number to select the first selected frame (beginning frame). Also, n' indicates the number of selected frames of the original video. The values of n' and r are selected randomly.

Figure number 3 shows the zero normalized cross-correlation obtained from the cross-correlation between the two input images. The image on the right side shows the output of the cross-correlation between two similar images. As you can see, the intensity is higher in the center of the image.

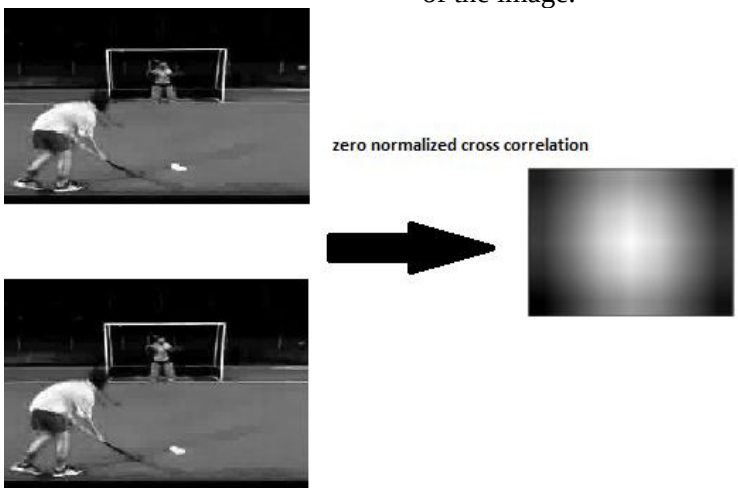

Fig.3 the result of applying zero normalized cross-correlation on a sample image where the correlation of two images is computed

4.1.Evaluation Metrics

In order to evaluate the proposed method, common and well-known benchmarks in this field have been used so that the proposed method can be examined in equal conditions. In this article, we use two well-known benchmarks in the field of video classification. The first benchmark is the UCF101 [49] dataset. This dataset contains 101 actual activities that videos are collected from YouTube. As of writing this article, the UCF101 dataset has the most diversity (in terms of performance) with 13,320 videos of 101 different activities. Also, camera movement, object shape, viewpoint, disordered backgrounds, intensity, and different brightness of this dataset are very challenging. The average clip length of this dataset is 7.21 sec. The length of the shortest clip is 10.6 sec and the length of the longest clip is 71.4 sec. Also, the framerate is equal to 25 fps and the resolution is equal to 320×240. This dataset is one of the challenging datasets.

The second benchmark is HMDB51 [50] dataset. This dataset contains 6766 videos with frame rate of 30 Fps. There are 51 different activities in this dataset. The accuracy of the proposed method is based on the averaging of the labeling accuracy on two datasets. The labeling is based on the highest score dedicated by Softmax to each clip. Figure 4 shows examples of HMDB51 and UCF101 dataset classes.

Fig.4. Examples of HMDB51 and UCF101 dataset classes

4.2.Implementation Details

In order to implement the proposed architecture, 3 ImageNet blocks have been used as convolutional towers. The features obtained from the avg_pool layer are used to extract spatial features. The features obtained in this step become a vector with 2048 properties. These features are given to the proposed LSTM unit in the next step. The size of the filters in the proposed LSTM units is considered 3×3. Also, there are pieces of input features with the size of 3×3 that are used to calculate the correlation. As mentioned earlier, horizontal and vertical rotations are performed on the input video frames to reduce overfitting. Also, the same cuts are made on the objects of the input images. In order to solve the problem of videos with different sizes (in terms of temporal) and different speeds on performing an action in them, the sampling of frames is done by starting from different points. If $V \in R^{n\times l\times h\times ch}$ is considered as a set of frames of a video, then $V' \in R^{n'\times l'\times h'\times ch}$ is the selected

frames` set from the original video which is given as input to the CNN network. Here $n' \leq n$ is the number of selected frames from the original video. Also, $l' \leq l$ and $h' \leq h$ are cropped size of the clip. ch is the number of video channels. According to the main video sequence, $v = (f_i)_{i=1}^{t}$ the number of selected frames from the main video ($v = (f_j)_{j=1}^{t'}$) is selected as follows:

$$v_{clip} = \{ f_{ij} \mid i_1 \in \{1, 2, ..., t - t' \times s\}, i_j = i_{j-1} + s\} \quad (5)$$

Where the i_1 is index of the first frame of video and s is the step between them. This parameter (means the first video frame) is selected randomly.

5.Results and Discussion

As mentioned before, we use data Augmentation techniques to reduce overfitting. For this purpose, temporally Augmentation can be used. Thus activities with different speeds can be created, which reduces overfitting. Temporally Augmentation has a great impact on the HMDB51 data set. It is expected that pre-training helps faster convergence and our experiments confirm. Data augmentation is done in the spatial dimension by making cuts on different parts of the image and also rotating in horizontal directions. The results are shown in Table 1. According to the mentioned Table 1, using temporally Augmentation data and pertained networks have better results than the original data and data on which only temporally Augmentation data occurred.

Table 1
Shows the results of the proposed method with different conditions

Different test conditions	UCF101	HMDB51
Original data (unchanged)	90.1	53.2
Temporally Augmented	93.2	61.2
Temporally augmented and pre trained	96.28	78.02

In the following in Table 2, we compare the proposed method of this article with ConvLSTM and ordinary LSTM to recognize action type in each of the two datasets, which it became clear that the proposed method has better results than the two other methods.

Table 2
Comparison of the proposed method with conventional ConvLSTM and LSTM methods

Method	UCF101	HMDB51
LSTM	80.5	49.8
ConvLSTM	85.2	54.0
Proposed method	96.28	78.02

In order to show the proposed method's performance, we compared the performance results of our method on two datasets UCF101 and HMDB51 with other methods. The methods presented in this table are classified into different types based on the model and type of input data.

1- Models that are in traditional form and are not called deep.

2- Models whose input is RGB data.

3- Models whose input is Optical Flow.

4- Models whose input is light stream Optical Flow and RGB images.

In Tables 3 to 6, we present 4 sections corresponding to 4 different types of methods and compare their results with two datasets. Our proposed method falls into the category of methods whose input is RGB data.

Table 3
Comparison of the performance of the method with models that are traditional and are not called deep

Methods		UCF101	HMDB51
Non-deep	iDT + FV [25]	85.9	57.2
Methods	iDT + HSV [26]	87.9	61.1
proposed method		96.28	78.02

Table 4
Comparison of the performance of the proposed method with models whose input is RGB data

	Methods	UCF101	HMDB51
Networks with RGB input	spatial stream network[27]	73.0	40.5
	Spatial net conv4 and conv5[28]	82.8	-
	scLSTM[30]	84.0	-
	C3D[29]	85.2	-
	MV-RAMDMM[23]	87.5	-
	Semi-CNN[22]	89.0	-
	MB-MHI[24]	89.3	61.2
	STDAN[16]	91.0	60.4
	proposed method	96.28	78.02

Table 5
Comparison of the performance of the proposed method with models whose input is Optical Flow

	Methods	UCF101	HMDB51
Networks with Optical-Flow input	Temporal net conv3 and conv4[31]	82.2	-
	Temporal network [27]	83.7	-
	Deeper Temporal Net[32]	84.9	-
	AytekNet[21]	93.9	-
	proposed method	96.28	78.02

Table 6
Comparison of the performance of the proposed method with models whose inputs are Optical Flow and RGB data

	Methods	UCF101	HMDB51
Networks with RGB and Optical-Flow input	RSTAN [19]	80.2	53.4
	TCLSTA (Frame + STA) [17]	85.9	54.8
	Two stream [27]	88.0	59.4
	end-to-end two-stream[14]	87.3	58.3
	JSTA [18]	88.6	59.8
	VideoLSTM [15]	89.2	56.4
	Two-stream model[13]	92.5	---
	DenseNet121[13]	92.5	59.3
	L2STM [33]	93.6	66.2
	proposed metho	96.28	78.02

The classification accuracy results of the proposed method on the UCF101 and HMDB51 datasets respectively are shown in Figures 5-a, 5-b, and 6.

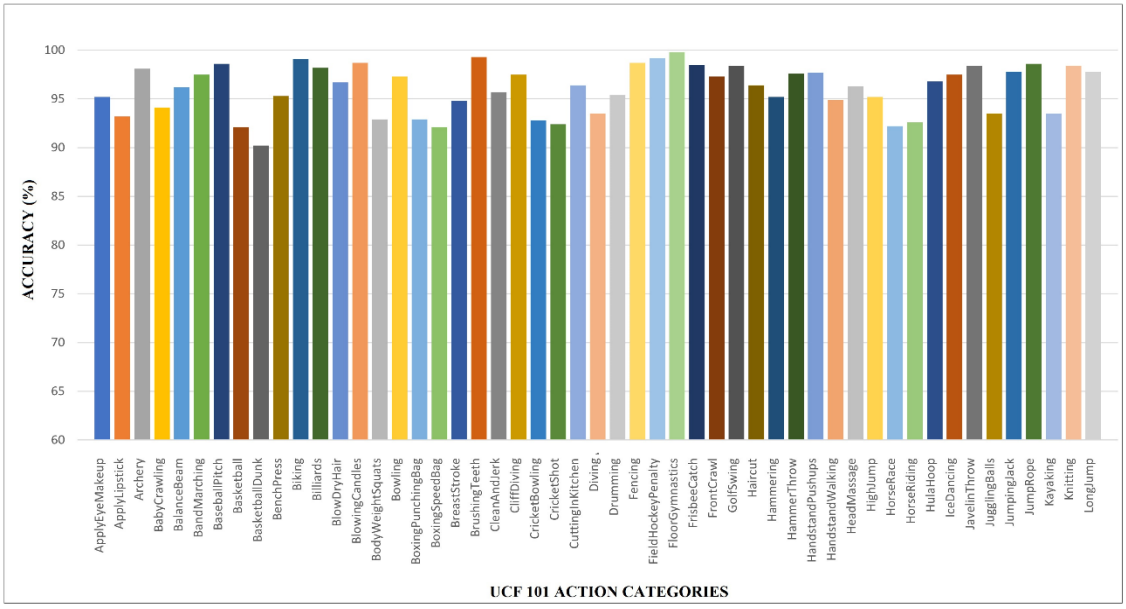

Fig.5-a Classification accuracy of the proposed method on the UCF101 data set (first part)

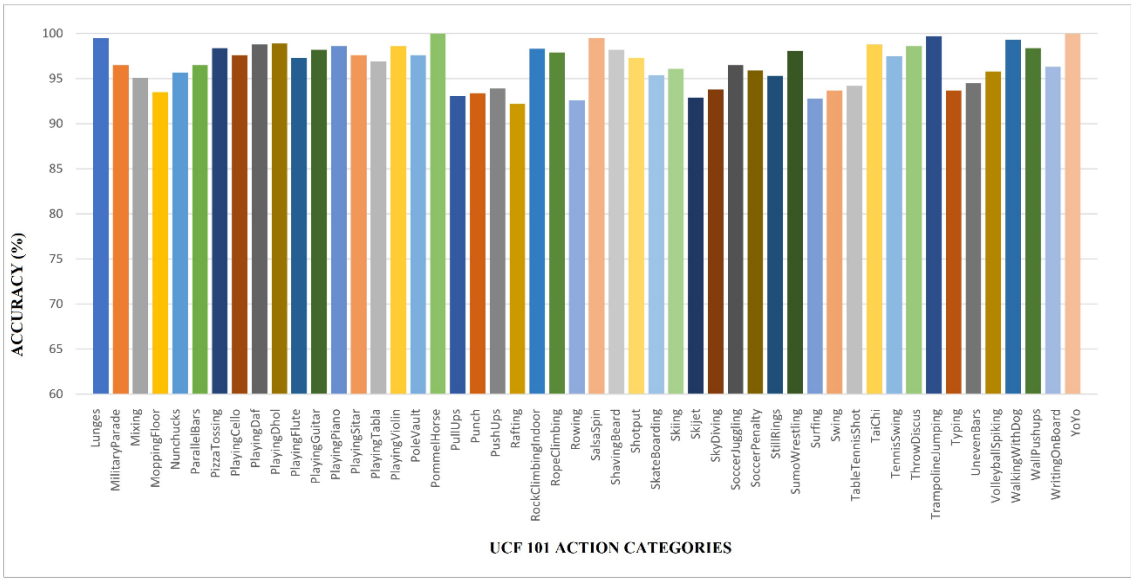

Fig.5-b Classification accuracy of the proposed method on the UCF101 data set (second part)

According to Figure 5, it can be seen that the "Basketball Dunk" and "Basketball" activities, as well as the "Horse Riding" and "Horse Race" activities, have the lowest recognition accuracy in the proposed method due to the high similarity of the activities and also the similarity of their combined movements. It can be said that the proposed method of this article recognizes "Pommel Horse" and "YoYo" activities 100% correctly.

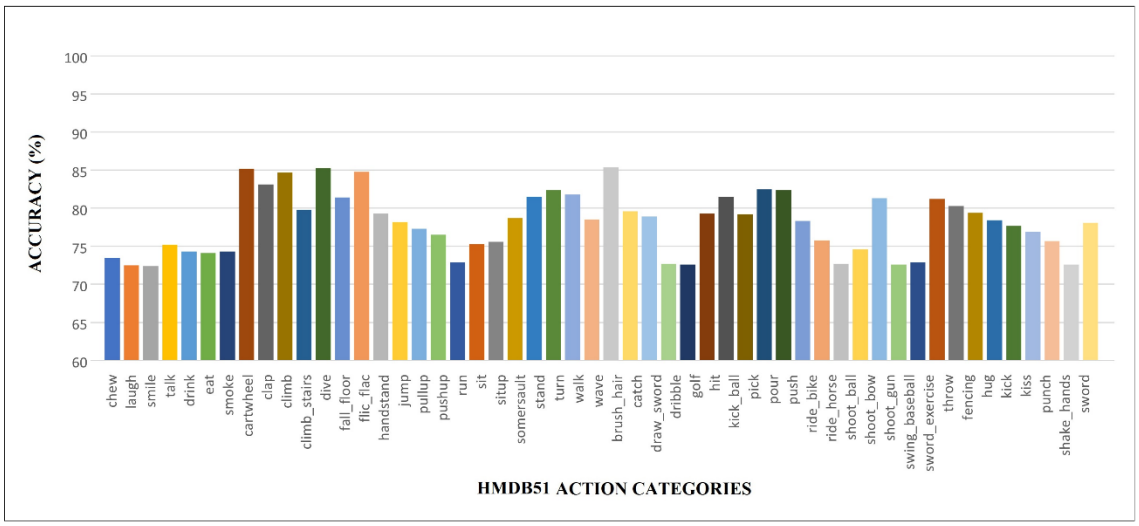

Fig-6 Classification accuracy of the proposed method on the UCF101 dataset

In Figure 6, it can be seen that “smile” and “laugh activities” have the lowest recognition accuracy. These activities have made the recognition operation difficult for our proposed system due to their similarity in performing activities. Also, some of these activities are recognized interchangeably due to their similarities. This issue can be seen in Figure 7 (Confusion matrix).

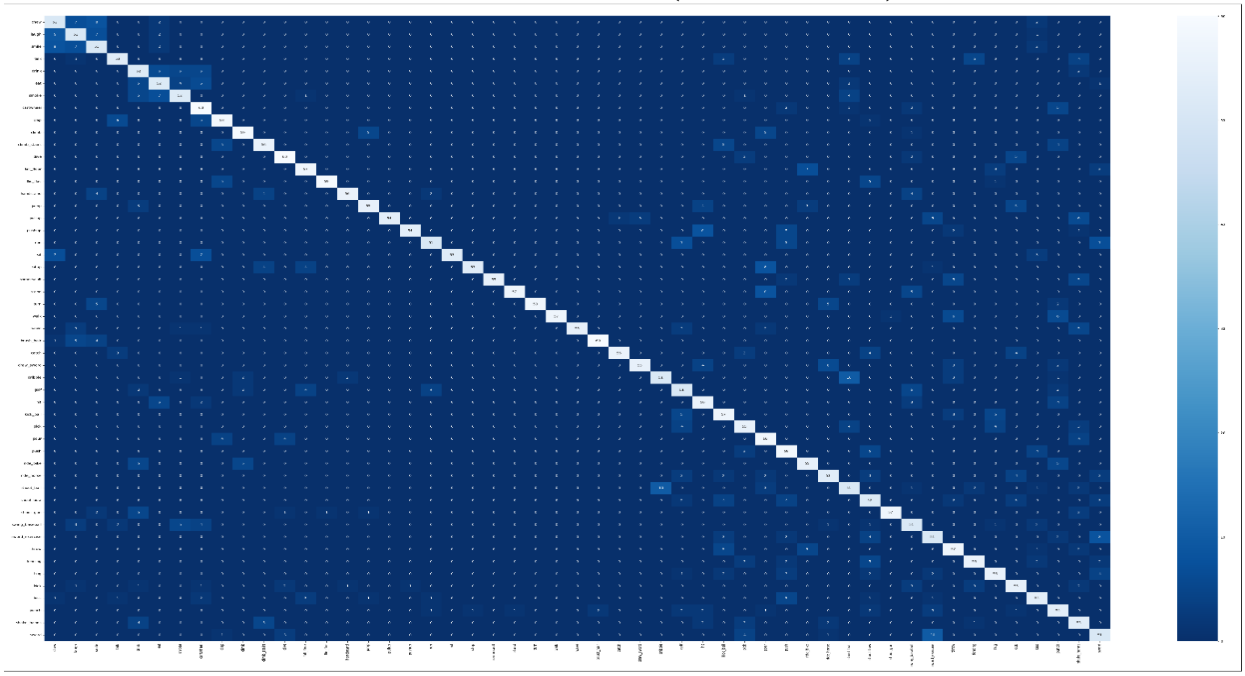

Fig-7 Confusion matrix obtained from the proposed method on the HMDB1 dataset

In the proposed method, “cartwheel”, “dive”, and “brush hair” activities from the HMDB1 dataset have the highest recognition accuracy.
Also, we selected different frames of the video as the starting frame (initial frame) and tested the recognition accuracy of the proposed method on them, the results of which can be displayed in Table no 7. The experience results show that the best performance is obtained when the starting frame is the sixth video frame.

Table 7
Proposed method experience results based on the selected starting frame

Experiences	Frame Selection	UCF101	HMDB1
No.1	Frame #2	94.21	74.91
No.2	Frame #4	95.28	76.36
No.3	Frame #6	96.28	78.02
No.4	Frame #8	94.87	76.04

According to the results shown in these tables, our proposed method is better than most existing methods because it considers longer-term temporal dependencies. According to Table 6, our proposed method had better performance than the other methods on the UCF101 dataset. It also had the best performance on the HMDB51 dataset compared to all other methods. You can also see a more complete comparison between our proposed method and other methods in Figure 8.

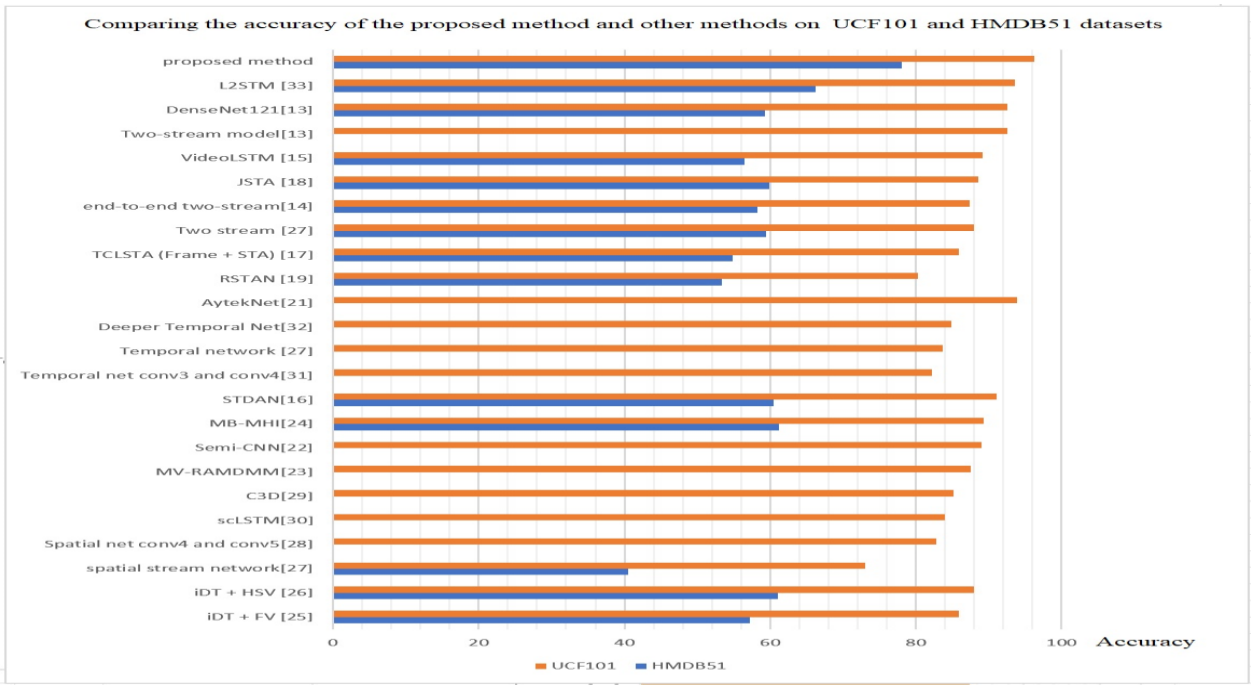

Fig.8 comparing the accuracy of the proposed method and the other methods on UCF101 and HMDB51 datasets

It should be noted that compared to the method [33] that uses a multi-modal to leverage both spatial and temporal information to recognize the action, our proposed method has less computational load and does not require preprocessed data. in fact, our proposed method can have efficient results in action modeling by backpropagating to the input image according to the class of action. This causes the area where the movement occurred to get prominent. Figure 9 shows two examples of videos in which backpropagation operations were performed. In this Figure, you can see that the proposed method can easily detect the moving points of the video. using action estimation based on correlation. It should be noted that according to [23] we can say that in the proposed method we used backpropagation through time in order to visualize the effect of kernelized cross-correlation in the modeling of action information. For this purpose, the neurons of the last layer are

Fig.9 detecting the moving points of the image

6.Conclusion

To action recognition in video images, we introduced a new architecture of a deep neural network with the design of a new LSTM unit that can recognize long-term time dependencies. Unlike conventional LSTM units, the proposed LSTM unit uses convolution operators to extract spatial properties along with temporal properties. In other words, the proposed unit is able to extract the 3 types of information from video images, which are:

1- temporal properties

2- spatial properties

3- temporal dependence in temporal and spatial properties

The results show the proposed method has better performance on larger and more complex datasets. Also, it shows that designed LSTM unit for longer video and where an action is done at different intervals has better performance than other methods.

References

[1] Uddin M, Joolee J, Alam A, Lee Y (2017) Human Action Recognition Using Adaptive Local Action Descriptor in Spark. IEEE Access 5: 21157-21167.

[2] Aurangzeb, Khursheed, Haider I, Attique Khan M, Saba T, Javed K, Iqbal T, Rehman A, Ali H, Shahzad Sarfraz M (2019) Human Behavior Analysis Based on Multi-Types Features Fusion and Von Nauman Entropy Based Features Reduction. J Medical Imaging and Health Informatics 9, no. 4: 662-669.

[3] Arshad, Habiba, Attique Khan M, Sharif M, Yasmin M, Younus Javed M (2019) Multi-level features fusion and selection for human gait recognition: an optimized framework of Bayesian model and binomial distribution. J Machine Learning and Cybernetics:1-18.

[4] Pham, Hieu H, Salmane H, Khoudour L, Crouzil A, Zegers P, Velastin S (2019) A Deep Learning Approach for Real-Time 3D Human Action Recognition from Skeletal Data. International Conference on Image Analysis and Recognition: 18-32.

[5] Zhang, Pengfei, Lan C, Xing J, Zeng W, Xue J, Zheng N (2019) View adaptive neural networks for high performance skeleton-based human action recognition. IEEE trans actions on pattern analysis and machine intelligence: 1963 – 1978.

[6] Sharif, Muhammad, Attique Khan M, Akram T, Younus Javed M, Saba T, Rehman A (2017) A framework of human detection and action recognition based on uniform segmentation and combination of Euclidean distance and joint

entropy-based features selection. EURASIP J Image and Video Processing 2017, no. 1: 89.

[7] Vishwakarma, Dinesh K, Kapoor R (2015) Hybrid classifier based human activity recognition using the silhouette and cells. Expert Systems with Applications 42, no. 20: 6957-6965.

[8] Sargano, Bux A, Wang X, Angelov P, Habib Z (2017) Human action recognition using transfer learning with deep representations. International Joint Conference on IEEE (IJCNN): pp. 463-469.

[9] Jaouedi N, Boujnah N, Bouhlel M.S (2020) New Hybrid Deep Learning Model for Human Action Recognition. J. King Saud Univ,Comput. Inf. Sci, 32: 447–453.

[10] Carrara F, Elias P, Sedmidubsky J, Zezula P (2019) LSTM-based real-time action detection and prediction in human action streams. Multimedia Tools and Applications; Springer, Berlin, Germany, Volume 78: 27309–27331.

[11] Yang K, Ding X, ChenW (2019) Multi-Scale Spatial Temporal Graph Convolutional LSTM Network for Skeleton-Based Human Action Recognition. International Conference on Video, Signal and Image Processing: 3–9.

[12] Rodríguez-Moreno, I., Martínez-Otzeta, J. M., Sierra, B., Rodriguez, I., & Jauregi, E. (2019). Video Activity Recognition: State-of-the-Art. Sensors: 19, 3160.

[13] Zhao Y, Man K, Smith J, Siddique K, Guan S (2020) Improved two-stream model for human action recognition. EURASIP J Image and Video Processing volume, Article number: 2-9.

[14] Dai C, Liu X, Lai J (2020) Human action recognition using two stream attention based LSTM networks. Applied Soft Computing, vol 86: 105820.

[15] Li Z, Gavrilyuk K, Gavves E, Jain M, Snoek C. G (2018) Video lstm convolves, attends and flows for action recognition. Computer Vision and Image Understanding 166: 41–50.

[16] Zhang Z, Lv Z, Gan C, Zhu Q (2020) Human action recognition using convolutional LSTM and fully-connected LSTM with different attentions. Neuro computing, vol 410: 304-316.

[17] Peng Y, Zhao Y, Zhang J (2019) Two-stream collaborative learning with spatial temporal attention for video classification. Circuits Syst, Video Technol. 29 (3): 773–786.

[18] Yu T, Guo C, Wang L, Gu H, Xiang S, Pan C (2018) spatial-temporal attention for action recognition. Pattern Recognition. Lett. 112 :226–233.

[19] Du W, Wang Y, Qia Y (2018) Recurrent spatial-temporal attention network for action recognition in videos. Image Process. 27: 1347–1360.

[20] Huang G, Liu Z, Van Der Maaten L, Weinberger K. Q (2017) Densely connected convolutional networks. Computer Vision and Pattern Recognition (CVPR): 2261–2269.

[21] Nebisoy A, Malekzadeh S (2021) Video Action Recognition Using spatial-temporal optical flow video frames: - arXiv preprint arXiv:2103.05101.

[22] Leong M.C, Prasad D.K, Lee Y.T, Lin F (2020) Semi-CNN Architecture for Effective Spatio-Temporal Learning in Action Recognition. Appl. Sci. 10, 557.

[23] Al-Faris M, Chiverton J, Yang Y, David N (2020) Multi-view region-adaptive multi-temporal DMM and RGB action recognition. Pattern Anal, Appl: 1587–1602.

[24] Naeem, H. B., Murtaza, F., Yousaf, M. H., & Velastin, S. A. (2020). Multiple Batches of Motion History Images (MB-MHIs) for Multi-view Human Action Recognition. Arabian Journal for Science and Engineering: 6109–6124.

[25] Wang H, Schmid C (2013) Action recognition with improved trajectories. International Conference of Computer Vision (ICCV): 3551–3558.

[26] Peng X, Wang L, Wang X, Qiao Y (2016) Bag of visual words and fusion methods for action recognition. Computer Vision and Image Understanding: 109–125.

[27] Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. Advances in neural information processing systems: 568–576.

[28] Wang L, Qiao Y, Tang X (2015) Action recognition with trajectory pooled deep convolutional descriptors: conference on computer vision and pattern recognition: 4305–4314.

[29] Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3d convolutional networks. International Conference on computer Vision: 4489-4497.

[30] Wang X, Gao L, Song J, Shen H (2017) Beyond frame-level CNN: Saliency-aware 3-d cnn with lstm for video action recognition. Signal Processing Letters: 510–514.

[31] Wang L, Qiao Y, Tang X (2015) Action recognition with trajectory-pooled deep convolutional descriptors. International Conference on computer vision and pattern recognition: 4305–4314.

[32] Han Y, Zhang P, Zhuo T, Huang W, Zhang Y (2018) Going deeper with two-stream convents for action recognition in video surveillance. Pattern Recognition Letters 107: 83–90.

[33] Sun, L., Jia, K., Chen, K., Yeung, D. Y., Shi, B. E., & Savarese, S. (2017). Lattice Long Short-Term Memory for Human Action Recognition (Version 1). arXiv preprint arXiv:1708.03958.

[34] Wang, Jun, Zhou S, Xia L (2018) Human interaction recognition based on sparse representation of feature covariance matrices. J Central South University 25, no. 2: 304-314.

[35] Meng, Bo, Liu X, Wang X (2018) Human action recognition based on quaternion spatial temporal convolutional neural network and LSTM in RGB videos. Multimedia Tools and Applications: 1-18.

[36] Baby, Anna S, Vinod B, Chinni C, Mitra K (2018) Dynamic Vision Sensors for Human Activity Recognition. arXiv preprint arXiv:1803.04667.

[37] Gao, Z, Li S, Zhang G, Zhu Y. J, Wang C, Zhang H (2017) Evaluation of regularized multi-task leaning algorithms for single/multi-view human action recognition. Multimedia Tools and Applications 587 76, no. 19: 20125-20148.

[38] Xiao, Q., & Song, R. (2017). Action recognition based on hierarchical dynamic Bayesian network. In Multimedia Tools and Applications: 6955–6968.

[39] Carrara F, Elias P, Sedmidubsky J, Zezula P (2019) LSTM-based: real-time action detection and prediction in human action streams. Multimedia Tools and Applications Volume 78: 27309–27331.

[40] Poppe R (2010) A survey on vision-based human action recognition. Image and vision computing: 976–990.

[41] Ji S, Xu W, Yang M, Yu K (2013) 3d convolutional neural networks for human action recognition. pattern analysis and machine intelligence: 221–231.

[42] Feichtenhofer C, Pinz A, Zisserman A (2016) Convolutional two-stream network fusion for video action recognition, in. Conference on Computer Vision and Pattern Recognition (CVPR): 1933–1941.

[43] Zhao S, Liu Y, Han Y, Hong R, Hu Q, Tian Q (2019) Pooling the convolutional layers in deep convNets for video action recognition. Circuits Syst, Video Techno: 1839–1849.

[44] Donahue J, Hendricks L.A, Guadarrama S, Rohrbach M, Venugopalan S, Darrell T, Saenko K (2015) Long-term recurrent convolutional networks for visual recognition and description. Computer Vision and Pattern Recognition (CVPR): 2625–2634.

[45] Xu Y, Han Y, Hong R, Tian Q (2018) Sequential video VLAD: training the aggregation locally and temporally. Image Process: 4933–4944.

[46] Javidani A, Aznaveh A.M (2018) Learning representative temporal features for action recognition. Multimedia Tools and Applications:1-19.

[47] Li D, Yao T, Duan L, Mei T, Rui Y (2018) Unified spatial-temporal attention networks for action recognition in videos, IEEE Trans. Image Process: 416–428.

[48] Liu Q, Che X, Bie M (2019) R-STAN: Residual spatial-temporal attention network for action recognition. IEEE Access 7: 82246–82255.

[49] Soomro, K., Zamir, A. R., & Shah, M. (2012). UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild (Version 1). arXiv preprint arXiv:1212.0402.

[50] Kuehne H, Jhuang H, Stiefelhagen R, Serre T (2013) Hmdb51: A large video database for human motion recognition, in.High Performance Computing in Science and Engineering 12: 571–582.

[51] Yeung, S., Russakovsky, O., Mori, G., & Fei-Fei, L (2015) End-to-end Learning of Action Detection from Frame Glimpses in Videos : - arXiv preprint arXiv: 1511.06984

[52] Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., & Van Gool, L (2016) Temporal Segment Networks: Towards Good Practices for Deep Action Recognition: - arXiv preprint arXiv: 1608.00859.

[53] Tran, D., Wang, H., Torresani, L., & Feiszli, M (2018) Video classification with convolutional neural networks. In Proceedings of the IEEE international conference on computer vision (ICCV): 4442-4450.

[54] Carreira, J., & Zisserman, A (2017) Quo vadis, action recognition? A new model and the kinetics dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR): 4724-4733.

[55] Zhang, Z., Qiao, Y., Xiong, Y., & Lin, D (2019) Dynamic refinement network for oriented and

detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR): 12240-12249.

[56] Wang, X., Girshick, R., Gupta, A., & He, K. (2018) Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR): 7794-7803.

[57] Feichtenhofer, C., Fan, H., Malik, J., & He, K (2019) Slowfast networks for video recognition. In Proceedings of the IEEE International Conference on Computer Vision (ICCV: 6202-6211.

[58] Zhang, F., Zhu, S., & Huang, Q (2019). Collaborative temporal and spatial modeling for video object segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR): 7152-7160.

[59] Shou, Z., Wang, D., & Chang, S. F (2017) Temporal action localization in untrimmed videos via multi-stage cnns. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR):1049-1058.

[60] Gao, J., Yang, Z., Sun, C., & Nevatia, R (2017) TALL: Temporal Activity Localization via Language Query. In Proceedings of the IEEE International Conference on Computer Vision (ICCV): 5277-5286.

[61] Xiong, Y., Chen, B., & Lv, W (2018) TSN: Temporal Segment Networks for Action Recognition. In Proceedings of the European Conference on Computer Vision (ECCV): 298-313.

[62] Zhu, X., & Yang, Y (2018) Online action detection and prediction with recurrent behavior modeling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR): 2697-2706.

[63] Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K (2018) Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) : 5987-5995.

[64] Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., & Lin, D (2018) Temporal action detection with structured segment networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR): 2914-2923.

[65] Hou, R., Chen, C., Shah, S. A. A., & Shah, S. M (2019) Tube Convolutional Neural Network (T-CNN) for Action Detection in Videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR): 7337-7346.

[66] Li, D., Chen, X., Zhang, Z., & Huang, K (2019) Learning Temporal Action Proposals with Fewer Labels. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR): 947-956.

[67] Deng, Z., Ding, C., Jiang, X., Qiao, Y., & Jia, A (2019) Semantic Proposal for Activity Localization in Videos via Sentence Query. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR): 2724-2733