[1]* **Merit Khaled**

[2] **Beladgham Mohammed**

# Improving Human Action Recognition in Videos with CNN–sLSTM and Soft Attention Mechanism

***Abstract:*** - Action recognition in videos has become crucial in computer vision because of its diverse applications, such as multimedia indexing and surveillance in public environments. The incorporation of attention mechanisms into deep learning has gained considerable attention. This approach aims to emulate the human visual processing system by enabling models to focus on pertinent aspects of a scene and derive significant insights. This study introduces an advanced soft attention mechanism designed to enhance the CNN-sLSTM architecture for recognizing human actions in videos. We used the VGG19 convolutional neural network to extract spatial features from the video frames, whereas the sLSTM network models the temporal relationships between frames. The performance of our model was assessed using two widely used datasets, HMDB-51 and UCF-101, with precision as the key evaluation metric. Our results indicate substantial improvements, achieving accuracy scores of 53.12% (base approach) and 67.18% (with attention) for HMDB-51 and 83.98% (base approach) and 94.15% (with attention) for UCF-101. These results underscore the effectiveness of the proposed soft attention mechanism in improving the performance of video action recognition models.

***Keywords:*** Action recognition, Soft Attention Mechanism (SAM), Convolutional Neural Network (CNN), Scalar Long Short-Term Memory Neural Networks (sLSTM), VGG19.

## I. INTRODUCTION

In human action recognition (HAR) systems, the a posteriori probability $P(Y|X)$ is usually modelled to build a relationship between a sequence of observations X and a specific Y, therefore tying the inputs X to their corresponding class labels Y. The objective was to identify the action being performed from a set of potential actions depicted in the video of an individual. While humans can effortlessly recognize specific actions using visual cues alone, developing an automated solution poses significant challenges owing to various factors. These factors include substantial variability among individuals regarding physical appearance, such as body structure and clothing, and differences in how actions are performed. Furthermore, the environment where the action occurs often involves complex conditions, including crowded settings, shadows, lighting changes, occlusions, and other variables, such as the angle of view and the subject's distance from the camera. Human actions encompass both spatial and temporal dimensions that are highly variable, resulting in the same action never being performed identically. The work of Dutta et al. is highly recommended for a comprehensive examination of the current challenges in this domain [1].

Human Activity Recognition (HAR) is a subject of significant importance in the fields of computer vision and pattern recognition, as the ability to automatically detect actions performed in a recording can serve as a valuable resource for various applications:

- Evaluating surveillance video footage [2] is a common scenario. Numerous security systems rely on the data collected from an extensive array of cameras. When the number of cameras is substantial, manual detection of critical events within the videos becomes challenging or even impractical.
- A practical application discussed in the previous section is the deployment of video comprehension techniques to supervise and care for elderly individuals and children in confined settings such as private residences and intelligent healthcare facilities.
- The monitoring and automatic recognition of daily activities can significantly aid residents and facilitate the generation of reports regarding their functional capacities and health status [3].

[1] * Laboratory of TIT, Department of Electrical Engineering, Tahri Mohammed University of Bechar, Algeria; merit.khaled@univ-bechar.dz

[2] Laboratory of TIT, Department of Electrical Engineering, Tahri Mohammed University of Bechar, Algeria; beladgham.mohammed@univ-bechar.dz

- Another application generates concise video summaries by selecting significant scenes from original content. This process entails a content-based search within the video databases. The ability to automatically generate textual descriptions of a given video obviates the need for manual annotations, and is crucial for developing more valuable and informative databases.

- Human-computer interaction [4] represents another domain that significantly benefits from advancements in action recognition techniques. These techniques can be employed to develop interfaces for individuals with reduced mobility, enhance their interactions with computers, and facilitate communication with others.

- Another example is the progression of video games [5], which permit users to engage with a console or computer without needing a physical device.

- Behavior-based biometrics has recently garnered significant attention. In contrast to traditional biometric methods, such as fingerprint analysis, behavior-based techniques collect identification data without disrupting an individual's activity. A notable example of this approach is the identification of individuals based on their gait.

- A related application is the development of tools that automatically guide patients in rehabilitation with motor problems.

In summary, emerging advancements in video action recognition techniques are expected to capture significant interest in various applications.

This study aimed to establish a system for identifying actions in video sequences. To accomplish this objective, the following approach is recommended:

(1) The CNN–sLSTM model uses a convolutional neural network (CNN) to extract video features, whereas an sLSTM neural network categorizes videos into specific classes. (2) An attention mechanism was included in this study. The remainder of this paper is organized as follows: Section II presents a summary of the state of the art of the problem in question; Section III describes the overall basic architecture of the model; and Section IV presents the attention mechanism of the model. Section V explains the datasets used, the evaluation approach, the experiments conducted, and the results obtained. Section VI concludes the study by summarizing the results and proposing directions for future research.

## II. Literature Review

The approaches applied in the literature to address the challenge of recognizing human actions can be classified into three groups:

### *A. Classical Approaches*

Traditional methodologies focus on extracting descriptors that encapsulate video frames' visual features and motion.

- Methods utilizing spatio-temporal points of interest (STIP) effectively capture video points within the spatio-temporal domain. A point of interest can be reliably identified by an STIP-based detector, such as a corner point or an isolated point where the intensity reaches a maximum or minimum, the endpoint of a line, or the point on a curve where the curvature is at its peak. Liu et al. [6] extend the Harris edge detector [7] to develop a 3D-Harris detector. STIPs are invariant to translation and scale but not to rotation.

- Trajectory-based methods utilize the paths of tracked feature points to represent the actions. Labana et al. [8] introduced a dense trajectory approach. Initially, clouds of characteristic points are sampled from each video frame, and the movement data is determined by tracking these features across frames applying an optical flow method. The resulting trajectories were used to represent the videos. In this research domain, compensating for camera movement by aligning characteristic points between frames using accelerated robust function descriptors (SURF [9]) and integrating them with other local descriptors, such as the histogram of oriented gradients (HOG [10]), histogram of optical flow (HOF [11]), and motion boundary histogram (MBH [12]), has yielded highly favorable results [13] in controlled environments.

### *B. Deep Approaches*

Applying deep learning to computer vision challenges has yielded remarkable outcomes in recent years. The primary achievement of neural networks (ANNs) is their capacity to approximate any continuous function, given a sufficient number of neurons [14]. Multilayered neural networks must integrate an adequate number of layers and neurons to learn meaningful input-output mappings efficiently. They obtain hierarchical features from unprocessed information with ascending complexity levels before the classification stage.

Convolutional Neural Networks (CNNs) were created to handle spatial data. Although they succeeded in image classification, they overlooked the temporal aspects of spatiotemporal data, such as videos. In particular, 3D CNNs are an extension of CNNs in the time domain that can capture features in a frame and the temporal evolution between consecutive frames. Vrskova et al. [15] propose a 3D-CNN approach, extracting data characteristics in spatial and temporal dimensions, thus capturing motion information in video transmissions. This architecture generates different feature maps from the features obtained through convolution and downsampling from each channel of successive video frames independently, with the final feature being formed from all channels. Experimental results show a considerable increase in performance from the modified models over the 2D-CNN architecture and other classic methods. However, it's important to acknowledge that the expectation of a universal feature map is problematic since each convolution is dependent upon a predetermined number of successive frames captured.

Recurrent Neural Networks (RNN) are typically trained to understand complex temporal dynamics, meaning RNN architectures excel at tasks involving sequential data, such as word generation, speech recognition, and human action recognition. Human actions are a series of complex movements and motor acts which, at their core, can be considered temporal dynamics. Thus, it's logical to develop a way through RNN architectures capable of understanding such sequential data. Moreover, specific scalar versions can prove even more valuable. For example, the Scalar Long Short-Term Memory Neural Networks (sLSTM) [16], [17] was designed to combat some inherent problems with basic RNNs, such as the vanishing gradient. The sLSTM is a memory cell that learns internal states through the storage, adjustment, and retrieval of information over time; thus, it excels at retaining and predicting information over long time dependencies. Therefore, using the sLSTM would be beneficial in any scenario where long-term temporal dynamics need to be understood and predicted, like with human action recognition.

Ye et al. [18] implement a hybrid model combining 3D convolutional networks to extract spatiotemporal characteristics from recording content with an Long short-term memory network that simplifies the temporal sequence into the video's ultimate feature vector.

Zhang et al. [19] use the recording sequence's velocity vector rather than the optical flow stream for online human action recognition to reduce processing time for faster, real-time results.

Sharma et al. [20] propose an LSTM neural network with an attention mechanism that allows each video frame to focus on a region most distinctive for the task. Learning such weights is a part of model training.

### *C. Dual-Flow Approaches*

Ibrayev et al. [21] propose the dual-stream hypothesis, which posits that the human visual cortex comprises two distinct pathways: the ventral stream, responsible for object recognition, and the dorsal stream, which is involved in the perception of movement.

Simonyan et al. [22] present a network of two flows containing a spatial and temporal network, exploiting the ImageNet dataset for pre-training and optical flow calculation to capture motion information explicitly.

Feichtenhofer et al. [23] implement a two-stream network with ResNet architecture [24] and additional connections between streams [25]. The additional two-stream approaches include Time Segment Networks [26], Action Transformations [27], and Convolutional Fusion [28].

## III. CNN - sLSTM Base Approach

Consider $v = \{x_1, x_2, \dots, x_n\}$ represent a video consisting of a sequence of frames $x_i$ with $i = 1, \dots, n,$. Fig. 1 illustrates a foundational Human Activity Recognition (HAR) system, which begins with the input phase, during which the video is normalized to 40 frames. In the Convolutional Neural Network (CNN) phase, a pre-trained VGG19 model was employed to extract the video features, resulting in feature dimensions of $40 \times 25088$.

Subsequently, the Scalar Long Short-Term Memory (sLSTM) phase is implemented, accompanied by a dense layer containing one node per class for the final classification.

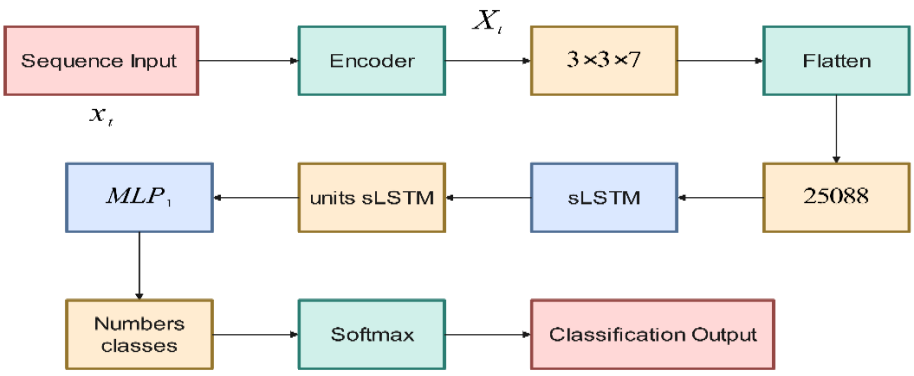



**Figure 1. Proposed CNN-sLSTM Base Architecture.**

The color blocks that make up the CNN-sLSTM base architecture, shown in Fig. 1, are as follows:

Encoder: We used the VGG19 convolutional architecture proposed by [22]. For each $x_t \in v$, we encode the frame in a cuboid $X_t$ of size $7 \times 7 \times 512$, resulting from the subsampling layer of VGG19, as shown in Fig. 2.

sLSTM: As proposed by [16], sLSTM is the natural behavior of remembering information over long periods. The inputs for a specific time $t$ are frame $x_t$, previous state $h_{t-1}$, and prior memory $c_{t-1}$. The outputs are the present state $h_t$ and memory $c_t$.

MLP1: The multilayer perception comprises three or more layers: an input layer, an output layer, and the remaining intermediate layers, called hidden layers. The details of the classification stage are presented in Table 2 (first row of the table).

: Indicates output dimension.

It should be noted that the weights associated with the sLSTM, such as the MLP, are part of the architecture training.

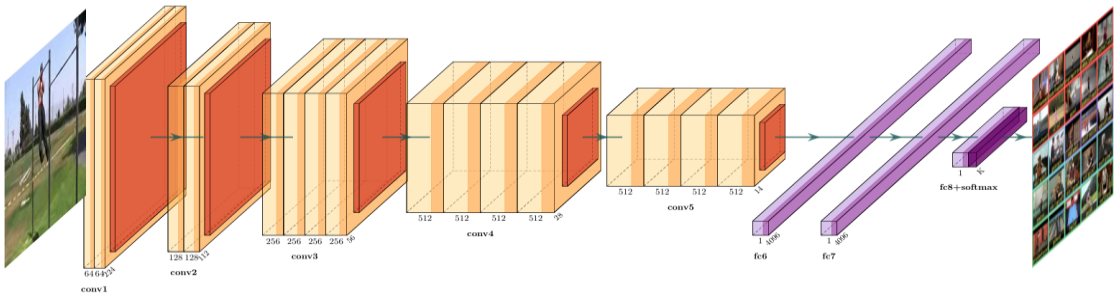



**Figure 2. Illustration of the VGG19 Architecture.**

### *A. Extraction of Features*

CNNs are constructed from a sequential set of layers that process input data. Each comprises computational modules that function based on the results of the previous layer. The most commonly applied layers are as follows: (a) convolutional layers, which employ k filters (or kernels) to generate k activation maps. (b) Subsampling layer: In most cases, a max-pooling operation is applied to each feature map, which systematically decreases the spatial

dimensions of the representation and the number of weights that require training. (c) Dense layer: This layer consists of fully connected neurons. The features derived from the training dataset were used to classify the input image into one of the predefined categories. The VGG19 convolutional architecture, as introduced by [22], achieved outstanding performance in classification and localization tasks during the ImageNet Large Scale Visual Recognition Challenge (ILSVRC-2014). Table 1 lists the layers comprising this architecture. The first column specifies the layer number and type of operation, such as 2 × Conv for Convolution, Max Pool for Max Pooling, and FC for Fully Connected. The second column denotes the number of the feature maps. The third column presents the sizes of the output features of each layer. The fourth column lists the kernel and stride architecture parameters.

**Table 1. Implementation of VGG using the pre-trained model.**

| Layer | | Feature map | Size | Kernel | Stride | Activation |
|---|---|---|---|---|---|---|
| Input | Image | 1 | $224 \times 224 \times 3$ | - | - | - |
| 1 | 2 × Conv2D | 64 | $224 \times 224 \times 64$ | $3 \times 3$ | 1 | ReLU |
| | MaxPooling2D | 64 | $112 \times 112 \times 64$ | $2 \times 2$ | 2 | - |
| 2 | 2 × Conv2D | 128 | $112 \times 112 \times 128$ | $3 \times 3$ | 1 | ReLU |
| | MaxPooling2D | 128 | $56 \times 56 \times 128$ | $3 \times 3$ | 2 | - |
| 3 | 4 × Conv2D | 256 | $56 \times 56 \times 256$ | $3 \times 3$ | 1 | ReLU |
| | MaxPooling2D | 256 | $28 \times 28 \times 256$ | $3 \times 3$ | 2 | - |
| 4 | 4 × Conv2D | 512 | $28 \times 28 \times 512$ | $3 \times 3$ | 1 | ReLU |
| | MaxPooling2D | 512 | $14 \times 14 \times 512$ | $3 \times 3$ | 2 | - |
| 5 | 4 × Conv2D | 512 | $14 \times 14 \times 512$ | $3 \times 3$ | 1 | ReLU |
| | MaxPooling2D | 512 | $7 \times 7 \times 512$ | $3 \times 3$ | 2 | - |
| 6 | Flatten | - | 25088 | - | - | - |
| 7 | FC | - | 4096 | - | - | ReLU |
| 8 | FC | - | 4096 | - | - | ReLU |
| Output | FC | - | 1000 | - | - | Softmax |

For each $x_i \in v$, we encode the frame to a cuboid of shape $X_i$ of $7 \times 7 \times 512$, resulting in the subsampling layer of the VGG19.

*B. Classification*

The sLSTM networks introduced by [16], [17] are characterized by their ability to retain information over long periods. Fig. 3 shows the fundamental configuration of the sLSTM unit. At any given time $t$, the inputs consist of the frame $x_t$, the preceding output $h_{t-1}$, and the preceding memory $c_{t-1}$. The results produced are the present output $h_t$ and the memory $c_t$.

The Scalar Long Short-Term Memory (sLSTM) network can add or remove information from its memory cell using gates. These gates enable the system to selectively transmit information, update the memory cells, or release information.

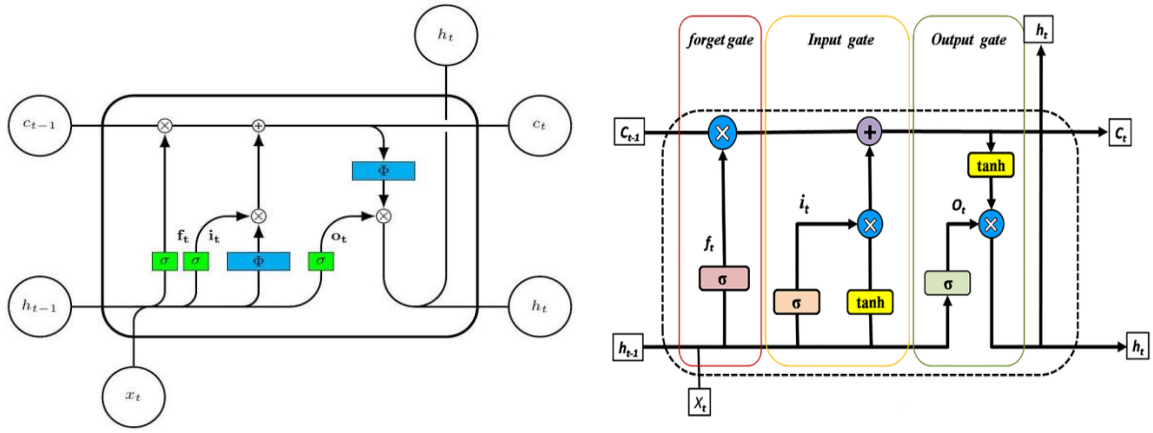
The first step in the sLSTM is to decide what information will be retained in the memory cell. This decision is made by the forgetting gate, which at time $t$ looks at the output of the memory block at time $t-1$, ht, in the input sequence at time $t$, $x_t$, and in the state of the memory cell, $c_{t-1}$. Equation (1) shows how the gate of oblivion calculates its value:

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right) \quad (1)$$

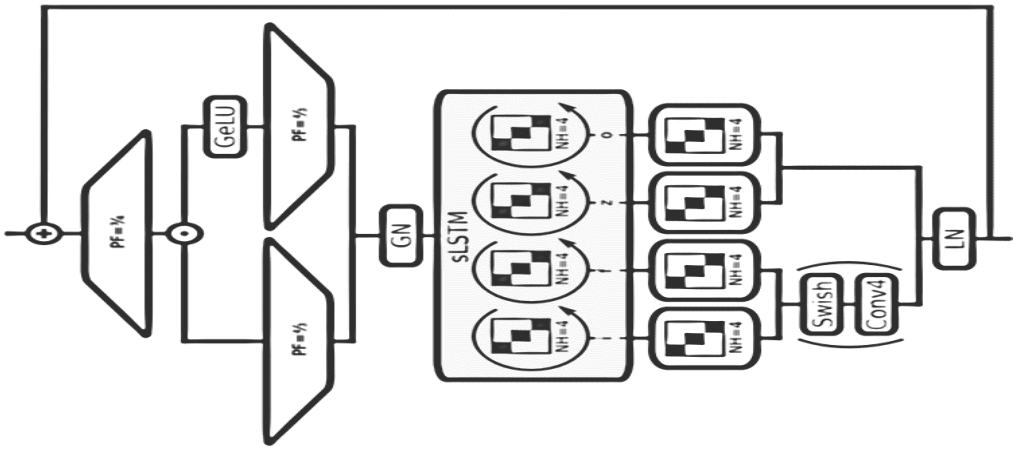
The following step involves selecting the information to be integrated into memory cells. This is accomplished through a two-step process: initially, the input to the neural network and the output from the sLSTM block at time $t-1$ are analyzed to determine the vector that refreshes the memory cell. Subsequently, the input gate is computed, which functions similarly to the forget gate; however, in this context, it regulates the volume of new information permitted to enter the memory cells. These computations are detailed in (2) and (3), respectively.

$$z_t = \phi(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \quad (2)$$

$$i_t = exp(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (3)$$




**(a)**




**(b)**

**Figure 3. The structure of the 2 deep learning models:**
**(a) The basic unit of LSTM. (b) The basic unit of sLSTM.**

After computing all the required values, the components necessary for updating the memory cell were ready, enabling the process to continue. Initially, the forget gate multiplies the existing value of the memory cell, effectively discarding information that the forget gate determines to be redundant. Subsequently, the integration of newly scaled information, as the input gate dictates, allows the memory cell to be updated. These processes were conducted simultaneously, as shown in (4).

$$c_t = f_t \otimes c_{t-1} \oplus i_t \otimes \phi(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{4}$$

It is essential to determine the specific information that must be generated. The output from the sLSTM block corresponds to the value of the memory cells, albeit with some modifications. Initially, a sigmoidal activation referred to as the exit gate was applied. This gate functions similarly to oblivion or entry gates, determining which memory cell components are generated. Subsequently, the memory cell values were processed using a hyperbolic tangent (tanh) function, which confines the output to a range between $-1$ and 1. This outcome is then multiplied by the output gate value that was previously computed, ensuring that only selected components are generated. These procedures are elaborated in (5) and (6), respectively.

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{5}$$

$$h_t = o_t \otimes \tilde{h}_t \tag{6}$$

$$\tilde{h}_t = \frac{c_t}{n_t} \tag{7}$$

$$n_t = \mathrm{f}_t n_{t-1} + \mathrm{i}_t \tag{8}$$

In this context, $\sigma$ represents the sigmoidal function, while the symbol $\otimes$ denotes the multiplication of gate values with the matrix weights, indicated by $W_{ij}$.

## IV. CNN-sLSTM Approach with Attention

Following the base architecture described in Section III, we include the attention mechanism proposed by [20]. Fig. 6 shows a general schematic of the architecture. To generate an attention map:

- The cuboid $x_t$ is transformed into a vector representation by averaging the feature map, which is input into an $mlp4$.
- The context vector $h_{t-1}$ is used as the input for $mlp2$.
- The weighting vector is formulated based on the output from mlp3, thereby resizing the vector to dimensions of $F \times F$, which encapsulates the probability distribution across all pixels within each feature map. In the attention map, a higher pixel value indicates a more critical image region influencing the decision-making process during the classification phase.

Table 2 presents the MLP configurations.

**Table 2. Configuration of the $MLP_s$.**

| MLP | Layer | Parameter |
|---|---|---|
| $\mathbf{MLP_1}$ | Fully Connected (FC) | #classes (neurons) |
| | Dropout | 0.5 |
| $\mathbf{MLP_2}$, $\mathbf{MLP_3}$ and $\mathbf{MLP_4}$ | Fully Connected (FC) | 128 units (neurons) |
| | Dropout | 0.5 |
| $\mathbf{MLP_h}$ and $\mathbf{MLP_c}$ | Fully Connected (FC) | 256 units (neurons) |
| | Dropout | 0.5 |

To initialize $h_0$ and $c_0$, Xu et al. [29] compressed all video details $v$ to achieve faster convergence, which is determined as (9) and (10):

$$h_0 = mlp_h\left(\frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{F^2}\sum_{i=1}^{F^2} X_{t,i}\right)\right) \tag{9}$$

$$c_0 = mlp_c\left(\frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{F^2}\sum_{i=1}^{F^2} X_{t,i}\right)\right) \tag{10}$$

Where $T = 40$ indicates the number of frames in the videos, and $F = 7$ specifies the dimension of the VGG19 feature map. All frames $x_i \in v$ through VGG19 produce $T$ cuboids. To compress this information, we first averaged the number of cuboids and the overall pixel values in each map of the characteristics. The resulting vector feeds one $mlp_h$ to obtain the initial state $h_0$ and one $mlp_c$ to obtain the initial memory $c_0$ . Table 2 shows the configuration of the $MLP_s$ for initialization.

## V. Experiments and Results

### *A. Evaluation Metrics*

To evaluate the performance of the system, we used the following evaluation methods compatible with multi-class classifications:

Accuracy is defined as the ratio of correctly identified samples to the overall number of samples and is expressed as follows:

$$\text{Accuracy } = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

Precision is the number of samples correctly classified as class $i$ divided by the total number of samples classified as class $i$.

$$\text{Precision }_i = \frac{TP}{TP + FP} \tag{12}$$

Recall is the portion of class $i$ samples that are correctly classified.

$$\text{Recall}_i = \frac{TP}{TP + FN} \tag{13}$$

Where, TP: True Positives, TN: True Negatives, FP: False Positives and FN: False Negatives.

### *B. Database*

- **HMDB-51 Human Motion dataset** [30] comprises a collection of action categories derived from videos sourced from various platforms, such as films, the Prelinger archive repository, YouTube, and Google. The actions were categorized into five distinct types: general facial actions, facial actions involving object handling, holistic corporeal activities, body movements involving object interaction, and body movements intended for human interaction. The dataset provides information for creating three splits, comprising 5,100 videos, with 3,570 allocated for training and 1,530 designated for testing. This allocation corresponded to a 70/30 split per class. Fig. 4 shows samples from the HMDB-51 dataset.

**Figure 4. Samples from the HMDB-51 Dataset.**

- **UCF-101 dataset** proposed by [31] contains 101 categories that can be classified into 5 types (human-object interaction, body movement only, human-human interaction, playing musical instruments, and sports). The total duration of these videos was over 27 hours. All videos were collected from YouTube with a frame rate of 25 FPS and a resolution of $320 \times 240$. The dataset also provides information for creating 3 splits, with the videos of a class divided into 25 groups. Seven clusters were designated for the test set, and the other 18 clusters were reserved for training. Fig. 5 shows the samples from the UFC-101 dataset.

**Figure 5. Samples from the UCF-101 Dataset.**

*C. Results*

Our architecture was developed in Python, employing the TensorFlow framework as the core library [32] on an Intel(R) Core (TM) $i7-12800H$ CPU @ $5.0\,GHz$ computer with $32GB$ of $DDR5$ memory and Windows 11 $Pro\ 64-bit\ (x64)$ System Software (OS). The experiments were conducted on an NVIDIA $Quadro\ RTX\ A2000\ GPU\ 8\,GB\ up\ to\ 24\,GB$.

The tuning of the network hyperparameters was achieved by reducing the cross-entropy loss function and employing stochastic gradient descent in combination with the $RMSProp$ optimization algorithm [33].

Table 3 shows the results achieved by our model applying $k-fold$ cross-validation with $k=4$, using 3 folds for training and 1 fold for testing. The algorithm's average accuracy ($\overline{Precision}$) and average recall ($\overline{Recall}$ ) are the averages of the k iterations.

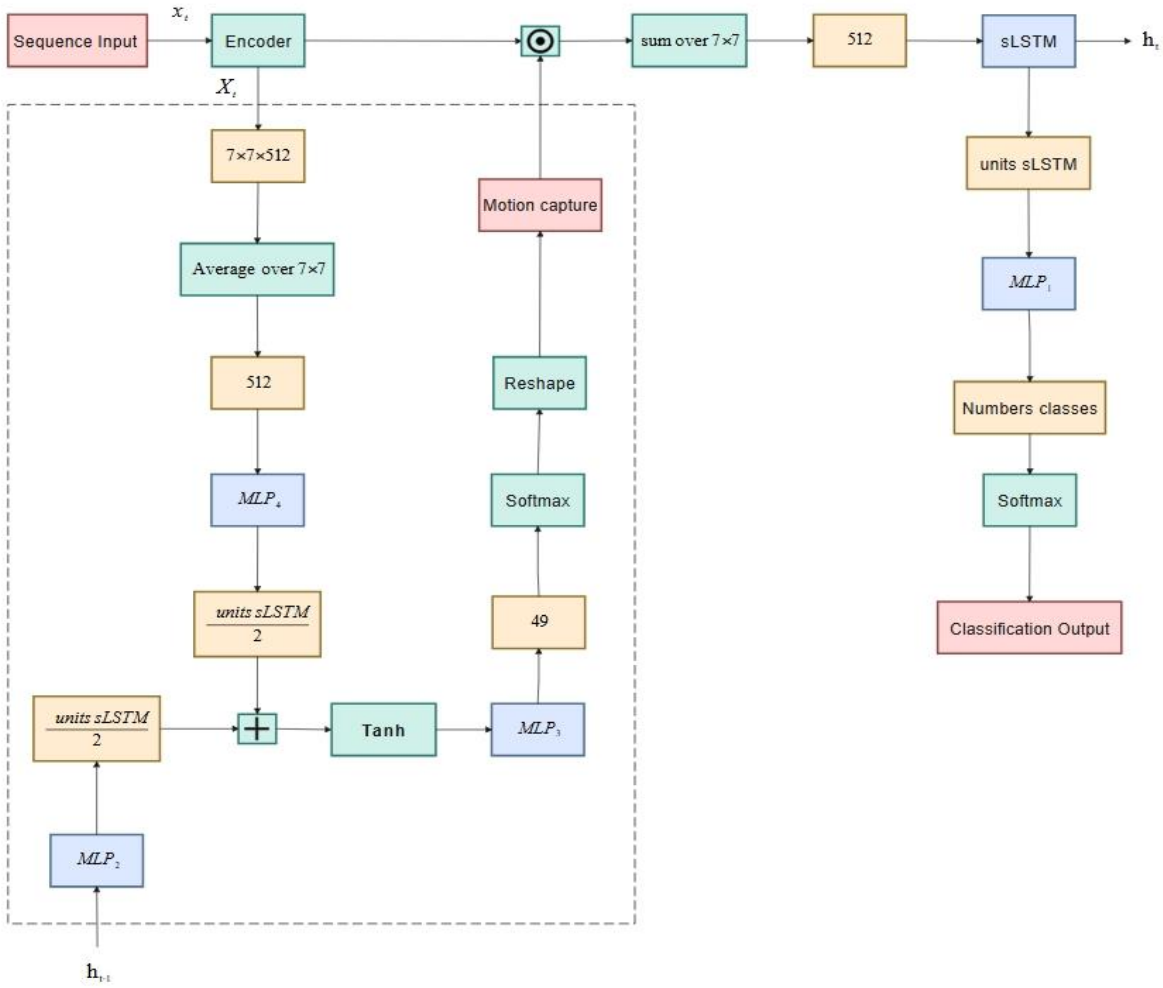



**Figure 6. Architecture with Attention Mechanism.**

**Table 3. Results for the $\overline{\text{Precision}}$ metrics (average precision) and $\overline{Recall}$ (average recall).**

| | | $\overline{\text{Precision}}$ | $\overline{\text{Recall}}$ |
|---|---|---|---|
| **HMDB-51** | Base Approach | 52,56 % | 53,67 % |
| | Approach with Attention | 66.14 % | 68.21 % |
| **UCF-101** | Base Approach | 83.94 % | 84.02 % |
| | Approach with Attention | 93.33 % | 94.97 % |

In Table 3, an increase in Precision and Recall can be observed when attention is applied to both databases. During the experiment, it was observed that discrimination became difficult for some pairs of classes consisting of similar actions or actions with similar backgrounds. For example, in the HMDB-51 dataset, the classes '$drink$' and '$eat$' tended to be confused more. The same occurred with the classes' '$smile$' and '$smoke$,' also from the HMDB-51 data set. In the UCF-101 data set, the classes '$brushing\ teeth$' and '$apply\ lipstick$' tended to be confused. The same happened with '$field\ hockey\ penalty$' and '$golf\ swing$'.

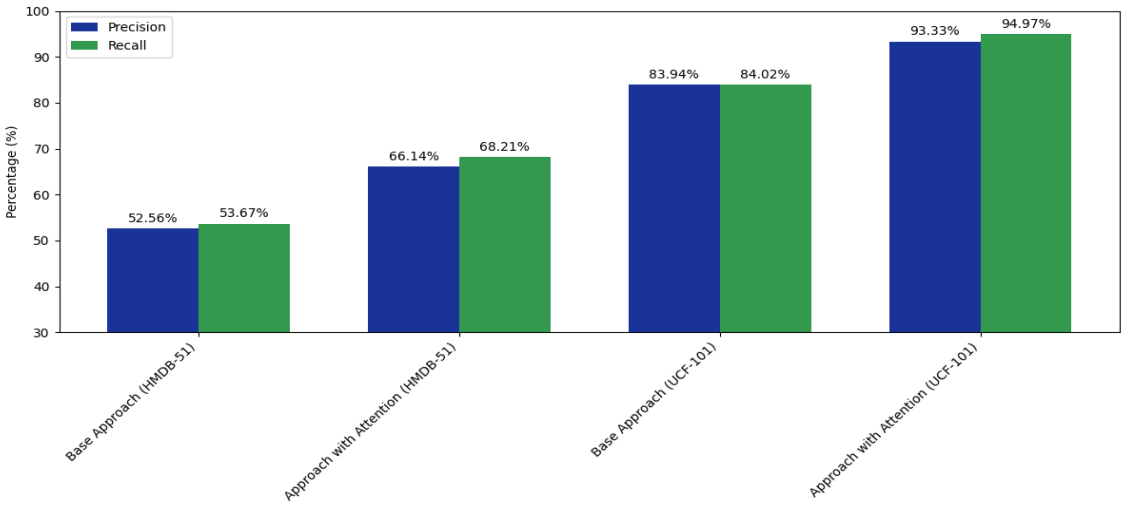



**Figure 7. Comparison of Precision and Recall for HMDB-51 and UCF-101 Datasets (Base vs. Attention) Approaches.**

Figure 7 compares the model performance using the "Base" and "Attention" approaches across the two datasets, HMDB-51 and UCF-101, regarding Precision and Recall. The data reveal that the Attention approach consistently outperforms the Base approach in both metrics for both datasets. Notably, the performance improvement is more substantial for the UCF-101 dataset, where the attention model achieves a precision of 87.33% and a recall of 89.97%, compared to the base model's precision of 71.94% and recall of 72.02%. In contrast, for HMDB-51, the Attention model achieved an accuracy of 47.14% and a recall of 48.21%, a minor improvement compared to the UCF-101 dataset.

Furthermore, UCF-101 outperformed HMDB-51 under all conditions, suggesting that it may be a more suitable dataset for this model or is better tuned to its characteristics. The improvements observed with the attention approach were apparent.

Table 4 summarizes our system's accuracy (ACC) after applying the original evaluation protocol for the HMDB-51 and UCF-101 datasets, alongside the results obtained with other approaches cited in the literature. The column '*Type*' describes the general approach following the classification described in Section II (where CA: Classical Approach, DL: Deep Learning, and DF: Double-Flow Networks).

**Table 4. Accuracy comparison of the proposed methods and state-of-the-art approaches on the UCF-101 and HMDB-51 databases.**

| References | Type | Model | Backbone | Modality | UCF-101 (%) | HMDB-51 (%) |
|---|---|---|---|---|---|---|
| Wang et al. [13] | CA | Dense Trajectories + Motion Boundary Descriptors | Dense Optical Flow | Optical Flow | - | 46.6 |
| Ye et al. [18] | DL | Spatiotemporal-LSTM | LSTM | RGB, Optical Flow | 85.4 | 55.2 |
| Zhang et al. [19] | DL | Two-Stream CNN | Motion Vector CNN | RGB, Motion Vector (MV) | 86.4 | - |
| Sharma et al. [20] | DL | Soft Attention Model | RNN, LSTM | RGB | - | 41.3 |
| Simoyan et al. [22] | DF | Two-stream ConvNet | ConvNet | RGB, Optical Flow | 88.0 | 59.4 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Feichtenhofer et al. [23] | DF | Two- Stream fusion | VGG-16 | RGB, Optical flow | 92.5 | 65.4 |
| Kuehne et al. [30] | CA | HOG/HOF | Harris3D detector | Optical Flow | - | 23.0 |
| Jiang et al. [34] | CA | Trajectory-Based Motion Modeling | Dense Trajectories | Optical Flow | - | 40.7 |
| Gaidon et al. [35] | CA | Hierarchical Motion Decomposition with BOF-Tree | Dense Tracklets | Optical Flow | - | 41.3 |
| Meng et al. [36] | DL | ConvLSTM-based attention | ResNet50/ResNet101 | RGB | 87.11 | 53.07 |
| Li et al. [37] | DF | DANet | ResNet-50 | RGB | 86.7 | 54.3 |
| Donahue et al. [38] | DL | LRCN | LSTM | RGB | 87.6 | - |
| Kay et al. [39] | DL | CNN-LSTM | ResNet-50, LSTM | RGB | 84.3 | 43.9 |
| Hara et al. [40] | DL | R3D | ResNet-101 | RGB | 88.9 | 61.7 |
| | DL | ResNeXt-101 | ResNet-101 | RGB | 90.7 | 63.8 |
| Zhao et al. [41] | DL | Bi-LSTM | LSTM | RGB | - | 50.1 |
| Hu et al. [42] | DL | ST-D LSTM | LSTM | RGB | 75.70 | 44.11 |
| Vrskova et al. [15] | DL | 3D-CNN | 3D-CNN | RGB | 79.9 | - |
| Yosry et al. [43] | DL | Video-based (R3D) | ResNet-101 | RGB | 77.0 | 50.0 |
| | DL | Image-based (R2D-LSTM) | 2D ResNet-101, LSTM | RGB | 93.0 | 65.0 |
| Zhou et al. [44] | DL | CoCo Framework | TSM, BERT | RGB | 57.6 | 34.6 |
| Proposed work (2025) | DL | CNN-sLSTM | VGG19 | RGB | **83.98** | **53.12** |
| | DL | CNN-sLSTM + Soft Attention | VGG19 | RGB | **94.15** | **67.18** |

As shown in Table 4 and Figure 8, our proposed CNN-sLSTM-based model demonstrated strong performance in action recognition on the UCF-101 and HMDB-51 datasets. The model leverages the VGG19 convolutional neural network (CNN) for feature extraction, which has been fine-tuned for a large-scale image classification task with 1000 classes. An attention mechanism was incorporated into the architecture to enhance its performance further, allowing the model better to capture the videos' relevant spatial and temporal dependencies. This is a lightweight architecture that achieves a tradeoff between accuracy and computational efficiency, making it optimal for extensive video action-recognition operations.

Our model demonstrates competitive and, in some cases, superior performance compared to state-of-the-art approaches. For example, Feichtenhofer et al. [23] developed a double-flow network (DF) with a VGG-16 backbone that achieved 92.5% on UCF-101 and 65.4% on HMDB-51. Our results exceed theirs on HMDB-51 at 67.18%, using VGG19 as the backbone and attention mechanism. Thus, this further recognition stems from VGG19's more intricate features for trained results and the attention mechanism's capability of honing in on critical aspects of short video frames.

In addition to outperforming DF approaches in certain areas, our proposed model also surpasses the performance of other deep learning (DL) methods, such as Sharma et al. [20], who used a GoogleNet feature extractor with an attention mechanism, achieving 41.3% accuracy on the HMDB-51. Our model achieved significantly better accuracy on HMDB-51 and UCF-101 datasets, with 67.18% and 94.15%, respectively. This highlights the strength of incorporating a CNN-sLSTM architecture combined with attention compared to previous attention-based models.

Another major advantage of our approach is the application of static CNN features from VGG19 rather than employing a specialized feature extraction process as in Wang et al. [13]. To illustrate, Wang et al. [13] implemented Dense Trajectories + Motion Boundary Descriptors for feature extraction—a much more processor-intensive approach that needs much more feature engineering. We, however, implemented VGG19, which is a model trained with image classification, to provide high-level representations and avoid any type of manual feature extraction.

Incorporating the attention mechanism within our model also played a pivotal role in improving its performance, especially when compared with previous models, such as Sharma et al. [20]. By using RNNs and LSTMs with GoogleNet, Sharma's method had limited accuracy on both datasets, surpassing our method's ability to dynamically focus on the most relevant portions of the input video frames. This focus on critical features enabled our model to achieve a substantial performance boost, particularly in UCF-101, where it outperformed several other models, including those of Ye et al. [18] and Meng et al. [36], whose results were also based on spatiotemporal and LSTM methods.

Our approach also offers a more efficient alternative to double-flow networks, such as those proposed by Zhang et al. [19], where two separate CNNs process RGB and optical flow data, increasing computational costs. Our model, with a single-stream architecture and an attention mechanism, offers competitive performance with 93.6% accuracy on UCF-101 and 67.18% on HMDB-51 while requiring less computational overhead and shorter training time. This balance between accuracy and efficiency underscores the versatility of the CNN-sLSTM model for real-time video-action recognition tasks.

In conclusion, the proposed CNN-sLSTM-based model achieved state-of-the-art performance on the UCF-101 and HMDB-51 datasets, surpassing several existing methods. Integrating a pre-trained VGG19 backbone, CNN-sLSTM architecture, and attention mechanism resulted in superior performance compared with previous models, such as those by Sharma et al. [20], Feichtenhofer et al. [23], and others. Furthermore, our model achieves these results with a lower computational cost than double-flow models, making it a high-performance and efficient solution for action recognition.

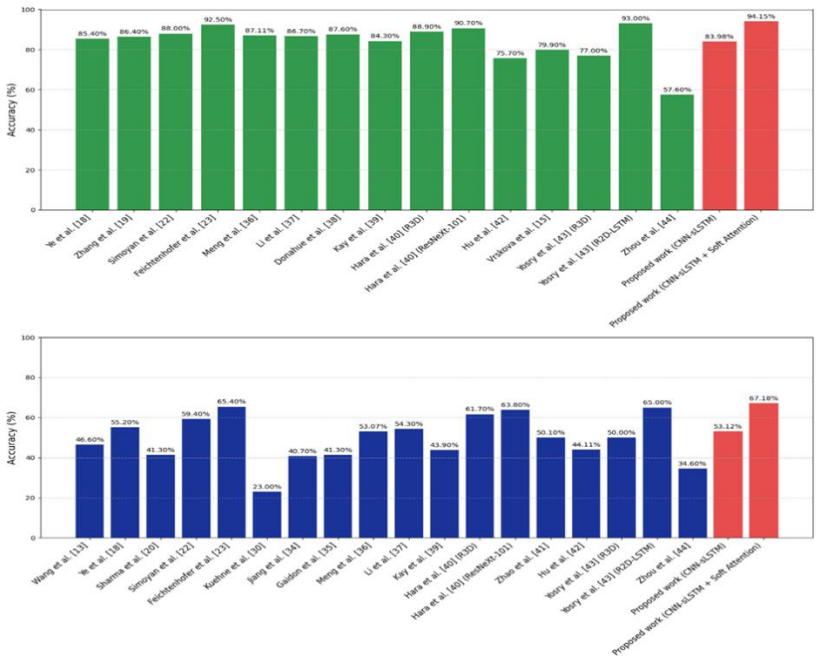



**Figure 8. Comparison of Accuracy for HMDB-51 (Top) and UCF-101 (Bottom) Datasets.**

## VI. Conclusions and Future Work

This study introduced an efficacious video action recognition framework based on a CNN-sLSTM neural network enhanced with an adapted attention mechanism. We studied improving and enhancing action recognition performance by leveraging spatial features extracted from video frames using the VGG19 model, followed by temporal feature modeling using an sLSTM network. The proposed framework incorporates an attention mechanism to focus on salient spatiotemporal features, thereby improving the model's ability to classify actions accurately. The model was developed in Python using the TensorFlow framework and evaluated on the HMDB-51 [30] and UCF-101 [31] datasets. The evaluations were performed using an NVIDIA Quadro RTX A2000 GPU to ensure computational efficiency.

The proposed framework demonstrates the effectiveness of combining spatial feature extraction, temporal modeling, and attention mechanisms for action recognition. The base architecture achieved competitive results, with accuracies of 53.12% on HMDB-51 and 83.98% on UCF-101. Including the attention mechanism significantly enhanced the performance, yielding accuracies of 67.18% on HMDB-51 and 94.15% on UCF-101 datasets. These results are comparable to state-of-the-art methods despite our approach's simplicity and resource efficiency. The attention mechanism was critical in improving the model's ability to capture discriminative spatiotemporal features, leading to superior classification performance.

The key contribution of this study is the presentation of a resource-efficient solution that achieves competitive results without the computational overhead associated with more complex architectures. By integrating an attention mechanism into the CNN-sLSTM framework, we demonstrated that even relatively simple models can achieve high performance when augmented with specific enhancements. This study highlights the potential of attention mechanisms for improving action recognition tasks, particularly with constrained computational resources.

In our future work, we will explore additional evaluation metrics to assess the proposed system's performance further and ensure a comprehensive understanding of its strengths and limitations. We will also consider using other datasets, such as Hollywood2 [45] and UCF-50 [46], to enhance the system's robustness and investigate techniques for mitigating overfitting. We propose experimenting with other convolutional neural networks for feature extraction, such as ResNet [24], to improve spatial feature representation further. Further exploration of advanced attention mechanisms [37], [47] will be conducted to refine the model's ability to focus on relevant spatiotemporal features. Another promising research direction is the application of transformer-based approaches [48] to video action recognition, which can potentially capture long-range dependencies more effectively than traditional recurrent architectures. These efforts contribute to developing more robust, scalable, and efficient action recognition systems suitable for real-world applications.

## References


[1] S. J. Dutta, T. Boongoen, et R. Zwiggelaar, « Human activity recognition: A review of deep learning-based methods », IET Computer Vision, vol. 19, no 1, p. e70003, janv. 2025. https://doi.org/10.1049/cvi2.70003

[2] H. Wu, X. Ma, et Y. Li, « Transformer-based multiview spatiotemporal feature interactive fusion for human action recognition in depth videos », Signal Processing: Image Communication, vol. 131, p. 117244, févr. 2025. https://doi.org/10.1016/j.image.2024.117244

[3] M. Abrar Ashraf et al., « A Novel Telerehabilitation System for Physical Exercise Monitoring in Elderly Healthcare », IEEE Access, vol. 13, p. 9120-9133, 2025. https://doi.org/10.1109/ACCESS.2025.3526710

[4] A. L. Kotian, R. Nandipi, U. M, U. R. S, Varshauk, et V. G. T, « A Systematic Review on Human and Computer Interaction », in 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India: IEEE, janv. 2024, p. 1214-1218. https://doi.org/10.1109/IDCIoT59759.2024.10467622

[5] B. Csontos et I. Heckl, « The evolution of video game accessibility on Xbox consoles in the Far Cry game series », Univ Access Inf Soc, mars 2025. https://doi.org/10.1007/s10209-025-01208-4

[6] H. Liu, Z. Ju, X. Ji, C. S. Chan, et M. Khoury, « Study of Human Action Recognition Based on Improved Spatio-Temporal Features », in Human Motion Sensing and Recognition, vol. 675, in Studies in Computational Intelligence, vol. 675. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2017. p. 233-250. https://doi.org/10.1007/978-3-662-53692-6_11

[7] Y. Ma, « Research on Sports Event Video Retrieval System Based on Harris Corner Detection Intelligent Algorithm », in 2024 4th Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS), Shenyang, China: IEEE, févr. 2024, p. 669-674. https://doi.org/10.1109/ACCTCS61748.2024.00124

[8] D. Labana et K. Modi, « Human Action Recognition Using Dense Trajectories », IJST, vol. 16, no 43, p. 3846-3853, nov. 2023. https://doi.org/10.17485/IJST/v16i43.2408

[9] D. Anandhasilambarasan, B. Bhushan, S. Ganga, R. Jain, P. Varma, et M. K. Mokashi, « SIFT and SURF: A Comparative Analysis of Feature Extraction Methods for Image Matching », in 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India: IEEE, juin 2024, p. 1-6. https://doi.org/10.1109/ICCCNT61001.2024.10726049

[10] L. Ke et Y. Luo, « A New Pedestrian Detection Method Based on Histogram of Oriented Gradients and Support Vector Data Description », in Frontiers in Artificial Intelligence and Applications, A. J. Tallón-Ballesteros, E. Cortés-Ancos, et D. A. López-García, Éd., IOS Press, 2024. https://doi.org/10.3233/FAIA231210

[11] J. D. Borneman, E. Malaia, et R. B. Wilbur, « Motion characterization using optical flow and fractal complexity », J. Electron. Imag., vol. 27, no 05, p. 1, juill. 2018. https://doi.org/10.1117/1.JEI.27.5.051229

[12] E. Gümüşkaynak et S. Eken, « The future of action recognition: are multi-modal visual language models the key? », SIViP, vol. 19, no 4, p. 345, avr. 2025. https://doi.org/10.1007/s11760-025-03951-w

[13] H. Wang, A. Kläser, C. Schmid, et C.-L. Liu, « Dense Trajectories and Motion Boundary Descriptors for Action Recognition », Int J Comput Vis, vol. 103, no 1, p. 60-79, mai 2013. https://doi.org/10.1007/s11263-012-0594-8

[14] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, et H. Arshad, « State-of-the-art in artificial neural network applications: A survey », Heliyon, vol. 4, no 11, p. e00938, nov. 2018. https://doi.org/10.1016/j.heliyon.2018.e00938

[15] R. Vrskova, R. Hudec, P. Kamencay, et P. Sykora, « Human Activity Classification Using the 3DCNN Architecture », Applied Sciences, vol. 12, no 2, p. 931, janv. 2022. https://doi.org/10.3390/app12020931

[16] M. Beck et al., « xLSTM: Extended Long Short-Term Memory », 2024, arXiv. https://doi.org/10.48550/ARXIV.2405.04517

[17] B. Alkin, M. Beck, K. Pöppel, S. Hochreiter, et J. Brandstetter, « Vision-LSTM: xLSTM as Generic Vision Backbone », 2024, arXiv. https://doi.org/10.48550/ARXIV.2406.04303

[18] Y. Ye et Y. Tian, « Embedding Sequential Information into Spatiotemporal Features for Action Recognition », in 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, USA: IEEE, juin 2016, p. 1110-1118. https://doi.org/10.1109/CVPRW.2016.142

[19] B. Zhang, L. Wang, Z. Wang, Y. Qiao, et H. Wang, « Real-Time Action Recognition With Deeply Transferred Motion Vector CNNs », IEEE Trans. on Image Process., vol. 27, no 5, p. 2326-2339, mai 2018. https://doi.org/10.1109/TIP.2018.2791180

[20] S. Sharma, R. Kiros, et R. Salakhutdinov, « Action Recognition using Visual Attention », 2015, arXiv. https://doi.org/10.48550/ARXIV.1511.04119

[21] T. Ibrayev, A. Mukherjee, S. A. Aketi, et K. Roy, « Toward Two-Stream Foveation-Based Active Vision Learning », IEEE Trans. Cogn. Dev. Syst., vol. 16, no 5, p. 1843-1860, oct. 2024. https://doi.org/10.1109/TCDS.2024.3390597

[22] K. Simonyan et A. Zisserman, « Very Deep Convolutional Networks for Large-Scale Image Recognition », 2014, arXiv. https://doi.org/ 10.48550/ARXIV.1409.1556

[23] C. Feichtenhofer, A. Pinz, et A. Zisserman, « Convolutional Two-Stream Network Fusion for Video Action Recognition », in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA: IEEE, juin 2016, p. 1933-1941. https://doi.org/10.1109/CVPR.2016.213

[24] M. Shafiq et Z. Gu, « Deep Residual Learning for Image Recognition: A Survey », Applied Sciences, vol. 12, no 18, p. 8972, sept. 2022. https://doi.org/10.3390/app12188972

[25] Q. Liu, X. Che, et M. Bie, « R-STAN: Residual Spatial-Temporal Attention Network for Action Recognition », IEEE Access, vol. 7, p. 82246-82255, 2019. https://doi.org/10.1109/ACCESS.2019.2923651

[26] H. Gammulle, S. Denman, S. Sridharan, et C. Fookes, « Hierarchical Attention Network for Action Segmentation », Pattern Recognition Letters, vol. 131, p. 442-448, mars 2020. https://doi.org/10.1016/j.patrec.2020.01.023

[27] X. Wang, A. Farhadi, et A. Gupta, « Actions ~ Transformations », in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA: IEEE, juin 2016, p. 2658-2667. https://doi.org/10.1109/CVPR.2016.291

[28] H. Qiao, S. Liu, Q. Xu, S. Liu, et W. Yang, « Two-Stream Convolutional Neural Network for Video Action Recognition », KSII TIIS, vol. 15, no 10, oct. 2021. https://doi.org/10.3837/tiis.2021.10.011

[29] K. Xu et al., « Show, Attend and Tell: Neural Image CaptionGeneration with Visual Attention », Proceedings of the 32nd International Conference on Machine Learning, vol. 37, p. 2048-2057, 2015. https://doi.org/10.48550/arXiv.1502.03044

[30] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, et T. Serre, « HMDB: A large video database for human motion recognition », in 2011 International Conference on Computer Vision, Barcelona, Spain: IEEE, nov. 2011, p. 2556-2563. https://doi.org/10.1109/ICCV.2011.6126543

[31] K. Soomro, A. R. Zamir, et M. Shah, « UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild », 2012, arXiv. https://doi.org/10.48550/ARXIV.1212.0402

[32] M. Ramchandani et al., « Survey: Tensorflow in Machine Learning », J. Phys.: Conf. Ser., vol. 2273, no 1, p. 012008, mai 2022. https://doi.org/10.1088/1742-6596/2273/1/012008

[33] O. F. Razzouki, A. Charroud, Z. E. Allali, A. Chetouani, et N. Aslimani, « A Survey of Advanced Gradient Methods in Machine Learning », in 2024 7th International Conference on Advanced Communication Technologies and Networking (CommNet), Rabat, Morocco: IEEE, déc. 2024, p. 1-7. https://doi.org/10.1109/CommNet63022.2024.10793249

[34] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, et C.-W. Ngo, « Trajectory-Based Modeling of Human Actions with Motion Reference Points », in Computer Vision – ECCV 2012, vol. 7576, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, et C. Schmid, Éd., in Lecture Notes in Computer Science, vol. 7576. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, p. 425-438. https://doi.org/ 10.1007/978-3-642-33715-4_31

[35] A. Gaidon, Z. Harchaoui, et C. Schmid, « Activity representation with motion hierarchies », Int J Comput Vis, vol. 107, no 3, p. 219-238, mai 2014. https://doi.org/10.1007/s11263-013-0677-1

[36] L. Meng et al., « Interpretable Spatio-temporal Attention for Video Action Recognition », 2018, arXiv. https://doi.org/10.48550/ARXIV.1810.04511

[37] X. Li, M. Xie, Y. Zhang, G. Ding, et W. Tong, « Dual attention convolutional network for action recognition », IET Image Processing, vol. 14, no 6, p. 1059-1065, mai 2020. https://doi.org/10.1049/iet-ipr.2019.0963

[38] J. Donahue et al., « Long-Term Recurrent Convolutional Networks for Visual Recognition and Description », IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no 4, p. 677-691, avr. 2017. https://doi.org/10.1109/TPAMI.2016.2599174

[39] W. Kay et al., « The Kinetics Human Action Video Dataset », 2017, arXiv. https://doi.org/10.48550/ARXIV.1705.06950

[40] K. Hara, H. Kataoka, et Y. Satoh, « Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? », 2017, arXiv. https://doi.org/10.48550/ARXIV.1711.09577

[41] H. Zhao et X. Jin, « Human Action Recognition Based on Improved Fusion Attention CNN and RNN », in 2020 5th International Conference on Computational Intelligence and Applications (ICCIA), Beijing, China: IEEE, juin 2020, p. 108-112. https://doi.org/10.1109/ICCIA49625.2020.00028

[42] K. Hu, F. Zheng, L. Weng, Y. Ding, et J. Jin, « Action Recognition Algorithm of Spatio–Temporal Differential LSTM Based on Feature Enhancement », Applied Sciences, vol. 11, no 17, p. 7876, août 2021. https://doi.org/10.3390/app11177876

[43] S. Yosry, L. Elrefaei, R. ElKamaar, et R. R. Ziedan, « Various frameworks for integrating image and video streams for spatiotemporal information learning employing 2D–3D residual networks for human action recognition », Discov Appl Sci, vol. 6, no 4, p. 141, mars 2024. https://doi.org/10.1007/s42452-024-05774-9

[44] J. Zhou, J. Liang, K.-Y. Lin, J. Yang, et W.-S. Zheng, « ActionHub: A Large-scale Action Video Description Dataset for Zero-shot Action Recognition », 2024, arXiv. https://doi.org/10.48550/ARXIV.2401.11654

[45] M. Marszalek, I. Laptev, et C. Schmid, « Actions in context », in 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL: IEEE, juin 2009, p. 2929-2936. https://doi.org/10.1109/CVPR.2009.5206557

[46] K. K. Reddy et M. Shah, « Recognizing 50 human action categories of web videos », Machine Vision and Applications, vol. 24, no 5, p. 971-981, juill. 2013. https://doi.org/10.1007/s00138-012-0450-4

[47] H. Yang, C. Yuan, L. Zhang, Y. Sun, W. Hu, et S. J. Maybank, « STA-CNN: Convolutional Spatial-Temporal Attention Learning for Action Recognition », IEEE Trans. on Image Process., vol. 29, p. 5783-5793, 2020. https://doi.org/10.1109/TIP.2020.2984904

[48] Y. Ma, R. Wang, M. Zong, W. Ji, Y. Wang, et B. Ye, « Convolutional transformer network for fine-grained action recognition », Neurocomputing, vol. 569, p. 127027, févr. 2024. https://doi.org/10.1016/j.neucom.2023.127027