Hybrid Deep Learning Techniques for Video Processing using attention-based LSTM and 3D CNN

P. Nagaraju, Research Scholar, Department of Computer Science & Engineering, Kakatiya University, Warangal, Telangana, India, nagarajupampati1234@gmail.com

Sadanandam Manchala, Professor & Dean, Department of Computer Science & Engineering, Kakatiya University, Warangal, Telangana, India, msadanandam@kakatiya.ac.in

Abstract

Large-scale video classification is a crucial task in computer vision, with applications ranging from autonomous systems to surveillance and content retrieval. Videos are complex as they involve both spatial and temporal dynamics, requiring models that can process this vast amount of information efficiently. Traditional approaches struggle with the high dimensionality and temporal continuity of video data. By combining a Long Short-Term Memory network, a tweaked and optimized 3D Convolutional Neural Network, and attention processes, it provides a unique method for video action identification. When it comes to handling the complexities of real-world situations, this synergy improves overall performance and gives an edge over current approaches. This method can extract both temporal and spatial information from video sequences, combined with the addition of an attention mechanism that highlights specific regions in the sequences to enhance identification accuracy. This model is well-suited for handling complex situations, including interactions among multiple actors, dynamic objects, and occlusion. This approach effectively addresses the subjectivity and variability inherent in action annotations within datasets. To improve model performance, it incorporates diverse preprocessing techniques. Comprehensive experiments conducted on the UCF101 and HMDB51 benchmark datasets demonstrate that the proposed method significantly outperforms existing state-of-the-art approaches in action recognition. These findings emphasize the potential of this framework to advance future research in video action detection.

Keywords— Video Action Recognition, 3D Convolutional Neural Network, Long Short-Term Memory, Attention Mechanism, Deep Learning, Human Activity Recognition, UCF101 Dataset, HMDB51 Dataset.

1. INTRODUCTION

One important area of artificial intelligence research is Human Behavior Analysis (HBA). Its applications are many and include smart retail settings, environment-assisted living, and video surveillance, among others. Thanks to the efforts of industry leaders, the amount of human video footage available is rapidly growing. This research work addresses HBA from a deep learning perspective. Deep Learning approaches have made significant advances in the categorization context over the last several years, due to an increase in processing capability. RNNs are utilized for short-term understanding, such as text or video, while Convolutional

Neural Networks (CNNs) are used for picture comprehension.

In computer vision and artificial intelligence, action recognition is a rapidly expanding discipline that seeks to recognize and categorize human activities in video clips. Although the field has made significant strides, several issues remain, including variations in perspective, size, and occlusion. The performance of an Action recognition algorithm may be significantly impacted by these differences, which also make the job more difficult. Moreover, action detection may be challenging when dealing with significant changes in camera angles, lighting, and background information. To properly identify activities, it is also essential to consider both temporal and geographical information [1]. Action recognition has been approached in several ways over the years, including trajectory-based approaches [2], Histograms of Oriented Gradients (HOG), Space-Time Interest Points (STIP), a Bag of Words (BoW), and more sophisticated approaches like 3D Convolutional Neural Networks (3D CNNs) and Recurrent Neural Networks (RNNs). However, the adoption of deep learning-based techniques in this sector has increased, as they have been more successful in identifying complex activities.

In recent years, Human Action Recognition (HAR) has increasingly adopted deep learning because of its ability to extract intricate patterns and representations from large amounts of data. Utilizing Convolutional Neural Networks (CNNs) is one of the most used deep learning methods in HAR. In addition to learning spatial representations of the activities, CNNs may extract characteristics from the input data. Convolutional filters are applied to the input data to extract progressively complicated features, with layers of filters stacked [3]. Another popular deep learning method for HAR is the use of Recurrent Neural Networks (RNNs), including Long Short-Term Memory (LSTM) networks and various variations. By using recurrent connections between the network's hidden states, RNNs can learn temporal representations of the activities and model temporal dependencies in the input data, enabling information to flow from one step to the next [4]. Nevertheless, these deep learning methods have some drawbacks, even though it is successful in HAR. For example, CNNs are excellent at processing picture frames to extract spatial information, but they struggle to capture long-term temporal connections. CNNs analyze frames separately, ignoring context over time, which leads to this restriction and may make it difficult to capture the temporal dynamics of activities. By analyzing frame sequences, RNNs, on the other hand, are designed to capture temporal information. However, they may struggle to utilize spatial information. RNNs may not use the rich spatial information in frames, as their primary concentration is on simulating temporal connections between frames. Learning long-term dependencies is also difficult with RNNs due to the vanishing gradient issue. Additionally, complicated situations where actions may be obscured or entangled, such as scenes involving numerous actors or objects, are difficult for most existing approaches to manage. To develop more reliable and accurate action detection techniques that can handle the complexity of real-world situations, further research is required, as these difficulties highlight.

Deep learning is a subfield of machine learning that can identify characteristics from large amounts of unlabeled training data by using sophisticated, multi-level "deep" neural networks. The use of deep learning has gained popularity over the past decade due to its superior performance compared to traditional techniques, such as decision trees, support vector

machines, and Bayesian networks. Advances in processing power have enabled the examination of biologically inspired algorithms that were designed many years ago.

A Convolutional Neural Network, a type of artificial neural network, is designed to process a large volume of input data, including audio, video, and pictures. It would be very inefficient to eliminate the function by using a fully connected (FC) network because of the volume of incoming data. Broadly speaking, CNNs extract certain properties by focusing on discrete parts of the data, hence minimizing the details. CNNs are built using filters, or kernels, which perform similar tasks to the Fully Connected ANN's weights. One convolutional filter is applied to each input area to generate a single output; this is the only way the weights of the FC vary from one another. The term for this is Local Receptive Fields, and it is very advantageous to reduce the number of weights CNN can recognize. Sliding the filter across the input allows for the measurement of the output. At each point, the product between each kernel element and the overlapped input element is calculated. The production at the present location is then calculated by adding up all the items.

Increasing the size of feature maps is a popular method for cutting down the computational load and parameters in a CNN. The pooling layer employs the maximum or average pooling in the feature map to minimize the areas while operating independently (often after a convolutional layer). Combining many convolutional and pooling layers in parallel is a highly effective way to discover features. The reason for this is that multiple kernel sizes may be implemented in parallel, enabling the identification of both basic and complicated functionality at various network levels. AlexNet (2012) was the first network to use parallelism.

While the methods discussed above focus on organizing independent data, handling time-dependent information requires a different approach. To meet this requirement, a specialized neural architecture was introduced to capture temporal sequence patterns. These systems, known as Recurrent Neural Networks (RNNs), enable information to persist across iterations. At each time step, a streaming input (x_t) generates a streaming output (o_t), which in turn serves as an input to the next step. Although simple RNNs can model short temporal dependencies, practical applications involving long information sequences typically require Long Short-Term Memory (LSTM) networks, a more robust form of RNN.

The RNN-based LSTM networks [18], which were first presented by Hochreiter and Schmidhuber (1997), are models that can learn long-term dependencies. Vanilla RNNs are quite basic in construction, consisting of a single perception and a single tanh activation layer. On the other hand, the structure of LSTM cells is more intricate and consists of four layers (tanh) that interact uniquely, as opposed to only one. By adding a temporal dimension, video analysis provides additional information for the recognition problem, enabling the further utilization of motion and other data. Processing brief video snippets also presents a significant computing challenge since each video may include hundreds or thousands of frames, not all of which are useful. Seeing video frames as still pictures and using CNNs to recognize each frame and average video level forecasts would be a foolish approach. Nevertheless, such a method would be employing incomplete information and might easily mislead groups, especially if fine-grained differences or areas of the video are unrelated to the intriguing action. This is because each video frame only constitutes a small part of the movie's plot. A comprehensive understanding of a video’s temporal structure is essential for accurate data classification. This presents a modelling challenge, as it requires representing videos of varying durations using a fixed set of parameters.

The primary goal of this network is to develop and implement an effective deep learning system that utilizes a combination of CNN and RNN architectures to predict and categorize human behavior.

Applications

Video Processing play vital role in several applications including:

- **Action Recognition**: Identifying human actions in videos (e.g., walking, jumping).
- **Video Summarization**: Creating short summaries of videos by identifying key events.
- **Event Detection**: Detecting specific events in videos, such as in sports or surveillance footage.
- **Video Captioning**: Generating text descriptions of video content.
- **Autonomous Driving**: Understanding and classifying scenes in real-time from dashcam footage.

In video processing, existing popular methodologies related to Artificial Intelligence, Neural Networks, and Computer Vision play a vital role in improving performance and accuracy. In this paper, we described different methods used for video processing, and proposed 3D CNN and Attention-based LSTM hybrid framework to perform Video Processing. The popular methods used for video processing is summarized in figure 1.

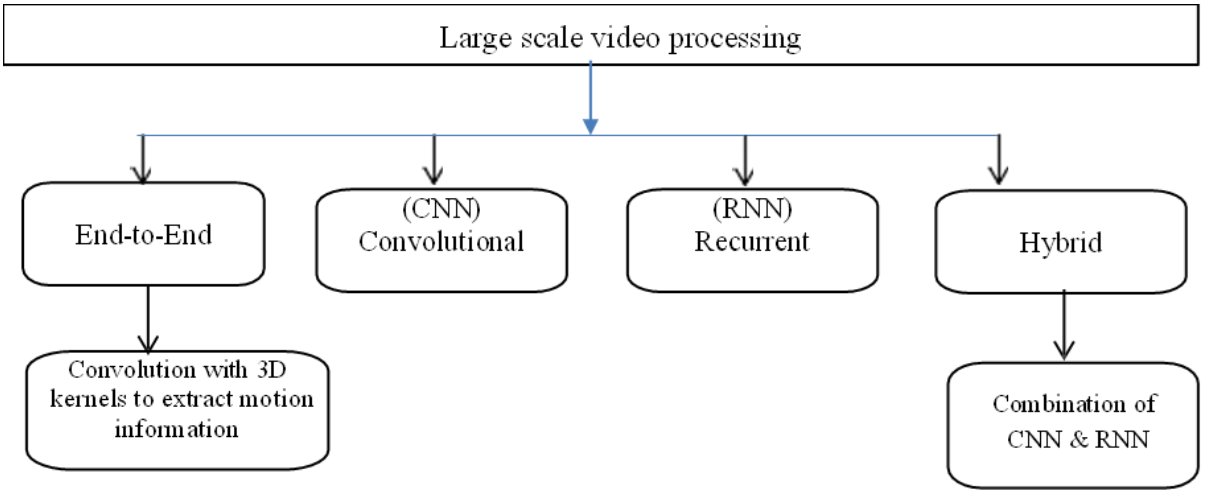


Fig. 1 Summary of methods of Video Processing

2. LITERATURE SURVEY

Over the past decade, researchers have proposed numerous handcrafted and deep learning-based systems for action recognition. Previous efforts were backed by characteristics created by hand for action videos that lack realism. Given that the suggested approach is based on a deep neural network (DNN), this paper discusses the evaluating relevant works that support DNN. Over the last several years, various deep learning model variations have been put out. for identifying human actions in films and have succeeded in outstanding results in computer vision challenges. Ji and others [4] used 3D convolutional kernels on video frames in a highly time-axis to record temporal and geographical information.

Karpathy et al. [5] applied CNNs to multiple frames within a sequence, employing single-frame, early, late, pooling, and gradual fusion strategies to capture temporal dependencies. That method, however, showed only modest improvements over the single-frame baseline.

Simonyan and Zisserman [6] introduced a two-stream CNN framework that combined spatial and motion information. One stream processed RGB frames, while the other used pre-computed stacked optical flow to capture short-term motion. Although optical flow alone cannot model long-term temporal dynamics, incorporating the second stream significantly enhanced action recognition performance, highlighting the importance of motion features.

Tran et al. [20] proposed the C3D architecture, which eliminates the need for pre-computed optical flow by enabling temporal feature learning directly within a 3D convolutional network. However, C3D captures only short-range temporal dependencies. To address the long-range temporal model, Wang et al. [7] developed the Temporal Segment Network (TSN), employing sparse temporal sampling to effectively capture extended temporal structures.

Feichtenhofer et al. [8] explored multiple strategies for fusing CNN streams to leverage both spatial and temporal information from appearance and optical flow networks. Despite these advancements, CNN-based approaches largely focus on appearance features and cannot model long-range temporal dynamics, failing to fully separate spatial and temporal domains.

Overall, while CNN-based methods have significantly advanced video action recognition, they are limited in capturing long-term temporal dependencies, motivating the exploration of alternative architectures such as RNNs, LSTMs, and Transformer-based models.

Several research methodologies have combined the advantages of both handcrafted and deep-learned features, as demonstrated in [9, 10], achieving promising results. These approaches integrate improved trajectories [1] and ConvNets [8]—two key components of effective video representations—into dual-stream architectures. Efforts have focused on determining optimal strategies to combine these feature types for robust video descriptions.

Depth imaging has also been employed in various studies, including the use of dynamic images and depth maps [3]. Bilen et al. [2] introduced the Dynamic Image Network, which generates dynamic images for action videos; however, it lacks certain discriminative capabilities, relying primarily on video frame ordering for supervision. Taylor et al. [11] proposed a Boltzmann restricted convolutional gated model that produces a flow field between adjacent frames rather than a single video-level map for action recognition. Similarly, Fisher Vector [13] and Rank Pooling [12] attempted to generate motion representations over time, but these methods fail to capture long-range temporal dependencies between video frames.

To model temporal dynamics effectively, Recurrent Neural Networks (RNNs) have been explored in video-based human action recognition due to their ability to learn and analyze hidden spatiotemporal patterns. Sequential data processing occurs by passing the hidden state st−1s_{t-1}st−1 from the previous time step, along with the current input $x_{t,}$ to the network. Numerous state-of-the-art approaches [14–19] have combined CNNs and RNNs for action recognition, demonstrating impressive results. However, standard RNNs often suffer from vanishing gradient problems and diminishing influence of the original input over multiple layers due to extensive parameter computations.

Long Short-Term Memory (LSTM) networks [15, 17, 20, 21] address this challenge by preserving long-term dependencies and sequence information through memory units and multiplicative gating mechanisms. Initially introduced by [20], LSTMs have achieved significant success in various sequential modelling tasks, including machine translation, speech recognition, and video description. Most LSTM-based video recognition approaches leverage fully connected CNN layers to extract high-level features that are fed into the LSTM, with the controlling information flow and error propagation being handled efficiently.

Each consecutive point in a video sequence is temporally linked to its neighboring points, and the integration of CNNs with RNNs provides an effective representation for modelling long-term motion and temporal sequences. RNNs leverage CNN-extracted features to capture more durable, long-range dependencies, whereas CNNs alone are limited to encoding local temporal information within individual video frames. It incorporates Long Short-Term Memory (LSTM) networks to capture motion-based cues and global sequence dependencies across the input video. LSTMs process fused spatiotemporal features, modelling complex hidden sequential patterns between frames. This approach demonstrates strong performance across videos of varying durations, achieving significant improvements in action recognition, as confirmed through extensive experimental evaluation.

1. **Karpathy et al. (2014)** introduced one of the first approaches for large-scale video classification using deep networks. Their model used convolutional neural networks (CNNs) to capture spatial information and pooled over time to extract temporal features. The benchmark dataset used was **Sports-1M**, one of the first large-scale video datasets with over 1 million videos.

2. **Simonyan & Zisserman (2014)** proposed the **Two-Stream CNN** architecture, which explicitly decouples the spatial and temporal features into two networks. The spatial stream handles raw frames, while the temporal stream focuses on optical flow to capture motion. This approach demonstrated significant performance improvements on action recognition tasks, particularly on the **UCF-101** and **HMDB-51** datasets.

3. **Tran et al. (2015)** developed **3D CNNs** that extend the convolution operations into the temporal dimension, enabling the network to model both spatial and temporal information simultaneously. Their **C3D model** was tested on multiple datasets, including **Sports-1M** and **UCF-101**, achieving state-of-the-art performance.

4. **Wang et al. (2016)** proposed **Temporal Segment Networks (TSN)**, which improved upon two-stream networks by better capturing long-range temporal dependencies. TSN achieved top performance on several datasets, like **Kinetics-400** and **something-Something**.

5. **Attention-based approaches (2019-present)**: The introduction of **Transformer** models in video classification, such as **ViViT (Video Vision Transformer)**, marked a new era where long-range temporal modeling became possible. These methods outperform conventional architectures on complex datasets such as **Kinetics-600** and **Moments in Time**.

2.1 Challenges

Despite these advancements, several challenges remain in video processing as per the state of the art:

- **High dimensionality**: Videos contain sequences of frames, which makes the input size large and computationally expensive.
- **Spatiotemporal features**: Unlike images, videos require both spatial (frame-level) and temporal (sequence-level) analysis to accurately classify actions or events.
- **Large-scale datasets**: Handling millions of video samples and thousands of classes is complex and demands efficient model design and optimization.
- **Video streaming**: An imbalance between moving and stationary regions, an abundance of duplicate data between frames, and other issues are common challenges for watermarking.
- **Digital watermarking:** The current techniques for digital watermarking proved inadequate in thwarting collusion assaults of type-1, type-2, composite, and ambiguity.

These challenges highlight the need for a robust framework capable of capturing long-range temporal dependencies, effectively integrating spatial and temporal features, and focusing on the most informative frames. To address these issues, It is proposed to design hybrid model combining 3D CNNs for spatio-temporal feature extraction with an attention-based LSTM network for sequential modelling, offering improved action recognition performance across complex video scenarios.

3. NEURAL NETWORK MODEL FRAMEWORKS

Hybrid deep learning techniques are widely used for large-scale video classification and can be evaluated using benchmark datasets that enable fair comparisons of model performance. These techniques combine the strengths of multiple approaches—such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformers—to capture both spatial and temporal information in videos. Large-scale video classification remains challenging due to the high dimensionality of video data, temporal dependencies between frames, and the size of available datasets. This section discusses key hybrid architectures, their underlying mathematical formulations, and popular benchmark datasets used for evaluation.

Here are some key hybrid deep learning techniques commonly used for large-scale video classification.

- **CNN + RNN (LSTM/GRU) Hybrids**
- **Two-Stream CNNs**
- **3D CNNs**
- **CNN + Transformer Hybrids**
- **Spatiotemporal Graph Convolutional Networks (ST-GCN)**

These methods are evaluated on large-scale benchmark datasets to measure their accuracy and computational efficiency.

3.1 CNN + RNN (LSTM/GRU) Hybrid

Convolutional Neural Networks (CNNs) are well-established for their ability to learn hierarchical spatial representations from visual data. When applied to video frames, CNNs capture localized spatial features such as edges, textures, and shapes, which are essential for understanding the appearance and structure within each frame.

On the other hand, Recurrent Neural Networks (RNNs) are designed to process sequential data by maintaining a hidden state that captures information over time. However, standard RNNs often suffer from vanishing or exploding gradient problems when handling long sequences. To address this, variants such as LSTM and GRU have been developed, incorporating gating mechanisms that enable the networks to learn long-term temporal dependencies more effectively.

In the CNN-RNN hybrid architecture [5], **CNNs** extract spatial features from video frames, and **RNNs** capture the temporal dynamics of the sequence. (e.g., LSTMs or GRUs)

- **Convolutional Neural Networks (CNNs)** are effective for extracting spatial features from individual video frames. They can capture the appearance and structural information within each frame.
- **Recurrent Neural Networks (RNNs)** and their variants like **Long Short-Term Memory (LSTM)** and **Gated Recurrent Units (GRUs)** are adept at modeling the temporal relationships between frames in a sequence. By combining CNNs with RNNs, the spatial features extracted by the CNN can be processed by the RNN to capture temporal dynamics.

Example: CNN can be used to extract frame-wise features, which are then fed into an LSTM to model the sequential nature of the video. This architecture is often used for action recognition.

Mathematical Formulation:

- Let a video be represented as a sequence of frames:

$$X=\{x1, x2\ldots xT\},\ x_t \in R^{H\times W\times C}$$

Where T is the number of frames, H and W are the height and width of each frame,

and C is the number of channels (e.g., RGB).

- The CNN extracts spatial features f_t from each frame:

$$f_t = CNN(x_t), f_t \in R^d\_$$

Where d is the dimension of the CNN-extracted feature vector.

- These features $\{f1, f2,\ldots,f_T\}$ are then fed into an RNN (e.g., LSTM):

$$h_t = LSTM(f_t, h_{t-1}), h_t \in R\_$$

Where h_t is the hidden state at time t

- Finally, the classification output is obtained from the hidden state of the final frame:

$$y = \text{softmax}(Wh_T + b)$$

In this architecture, CNN extracts frame-level spatial features, which are fed into an LSTM network to capture temporal dynamics across frames.

Benchmark Datasets:

- **UCF-101**: Consists of 13,320 videos across 101 action classes.
- **HMDB-51**: Contains 7,000 videos across 51 action classes.

3.2 Two-Stream CNN Networks (Spatial + Temporal Streams)

In this network, the input video data is processed through two distinct streams-**the spatial stream and the temporal stream**. Each stream designed to capture different aspects of the video, i.e., appearance information and motion information. Spatial stream is used to capture the static appearance and temporal stream is used to capture motion dynamics among the consecutive frames.

Two-stream CNN networks [6] are designed to process both spatial and temporal information separately. These two streams are then fused (often via late fusion techniques) to make predictions.

- **Spatial stream**: A CNN processes raw video frames (RGB images) to extract spatial features.
- **Temporal stream**: Another CNN processes optical flow or other motion-related information between frames to capture motion dynamics.

Example: The two-stream architecture has been widely used for action recognition tasks, where the spatial stream captures appearance information, and the temporal stream captures movement.

Mathematical Formulation:

Spatial stream for RGB frame extraction:

$$f_t^{spatial} = CNN^{spatial}(x_t)$$

Temporal stream for optical flow between consecutive frames:

$$f_t^{temporal} = CNN_{temporal}(\Delta x_t)$$

The two streams are fused using concatenation or element-wise summation:

$$f_t = \text{Fusion}(f_t^{spatial}, f_t^{temporal})$$

The fused features are passed through a fully connected layer for classification:

$$y = \text{softmax}(Wf_T + b)$$

Here, spatial CNN extracts features from RGB frames, while the temporal CNN captures motion via optical flow. The features are fused and fed into the classifier.

Benchmark Datasets:

- **Kinetics-400**: Contains 240k videos from 400 action categories.
- **Something-Something**: Includes human-object interaction videos for fine-grained action recognition

3.3 3D CNN for Spatiotemporal Feature Extraction

A 3D CNN is a type Neural Networks that performs 3 dimensional convolutions – operations that learn features across height, width, and depth. In spatiotemporal tasks, the third dimension is usually time or sequence.

In a **3D CNN** [4], the convolutional operations are extended into the temporal dimension, making it suitable for capturing spatiotemporal information across frames simultaneously. Unlike 2D CNNs, which process individual frames, **3D CNNs** extend convolutions to the time dimension, allowing them to capture both spatial and temporal features simultaneously.

Example: A 3D CNN can process a stack of frames as a single input and learn spatiotemporal features directly from the video data. This approach is effective for tasks such as video classification, human action recognition, and video understanding.

Mathematical Formulation:

Let the video sequence be X. A 3D convolution is applied with a kernel

$$K \in R^{kH \times kW \times kT}$$

where kH, kW, and kT are the kernel sizes for height, width, and time.

The 3D convolution at time t is computed as:

$$f_t = \sum K(i,j,k) \cdot X(i,j,k)$$

Pooling is performed to reduce the dimensionality:

$$p = \text{Pooling}(ft)$$

Finally, the pooled features are fed into a fully connected layer for classification.

$$y = \text{softmax}(Wp+b)$$

Here, 3D CNN processes multiple frames by applying spatiotemporal convolutions, followed by pooling and classification.

Benchmark Datasets:

- **Sports-1M**: Over 1 million videos from YouTube, categorized into 487 sports.
- **Kinetics-600**: An extended version of Kinetics-400 with 600 action classes.

3.4 CNN + Transformer Hybrid

CNNs are highly effective at modelling local spatial features within individual frames. They learn hierarchical features such as edges, textures, shapes, and object-level representations by applying convolutional filters in a localized manner. In video processing, CNNs are commonly used as feature extractors that process each frame to generate compact, spatially informative embeddings.

Transformers have performed well in computer vision tasks because they can capture long-range dependencies and overall context using self-attention mechanisms. In the context of video, Transformers can capture relationships across time (temporal modelling) and within spatial regions, enabling a more holistic understanding of scene dynamics.

Recent models combine the strengths of both CNNs and transformers for video classification. CNNs handle local spatial feature extraction, while transformers capture long-range dependencies across time. The extracted features are fed into a transformer encoder, which applies self-attention to model temporal dependencies.

Example: Models like **X3D** and **MViT (Multi-scale Vision Transformers)** use hierarchical architectures that apply CNN-based feature extraction followed by transformer blocks to model long-term temporal dependencies.

Mathematical Formulation:

CNN for feature extraction is computed as:

$$f_t = \text{CNN}(x)_t \; f_t \in R^d$$

Transformer encoder to model temporal dependencies:

$$Z_t = \text{Transformer Encoder}(\{f_1, f_2, \ldots, f_T\})$$

The self-attention mechanism is defined as:

$$\text{Attention}(Q,K,V) = \text{softmax}(KQ^T/d_k)V$$

Where Q, K, and V are query, key, and value matrices derived from input features, and d_k is the dimension of the key.

The final output is classified as:

$$y = \text{softmax}(Wz_T + b)$$

Benchmark Datasets:

- **Charades**: Contains 9,848 videos across 157 action classes, emphasizing activities in indoor environments.
- **Moments in Time**: A large-scale dataset with over 1 million videos across 339 action classes.

3.5 Spatiotemporal Graph Convolutional Networks (ST-GCN)

Spatiotemporal Graph Convolutional Networks are a class of neural networks designed to model data with inherent graph structures that evolve over time. They are particularly effective in tasks like skeleton-based action recognition, where human poses are represented as a graph of joints (nodes) connected by bones (edges).

ST-GCNs model [8] apply graph convolutions in both space and time to capture the relationships between key points (e.g., in a human skeleton) across multiple frames. By learning from both spatial structure and temporal progression, ST-GCNs are well-suited for understanding complex motion patterns and have demonstrated strong performance in human

action recognition tasks.

Example: This approach is used for tasks such as skeleton-based action recognition, where the spatial configuration and temporal dynamics of body joints are critical.

Mathematical Formulation:

Let the nodes of the graph G = (V, E), G = (V, E), G = (V, E) represent key points in a skeleton (e.g., human joints), and the adjacency matrix AAA defines the edges.

The node features are updated through a spatiotemporal graph convolution:

$$h^{(l)} = \sigma(k=\sum \alpha_k A_k h^{(l-1)} W^{(l)})$$

Where h(l) is the node representation at layer l,

A^k represents the adjacency matrix raised to power k,

and $W^{(l)}$ is a learnable weight matrix.

Finally, a classifier is applied:

$$Y = \text{softmax}(Wh_T + b)$$

Benchmark Datasets:

- **NTU RGB+D**: A large-scale dataset for 3D human activity recognition, containing 60 action classes.
- **Kinetics-Skeleton**: Derived from Kinetics by representing human poses using skeleton data.

3.6 Challenges and Solutions in Hybrid Approaches

- **Computational Complexity**: Video data is computationally expensive due to its high dimensionality. Hybrid models require significant computational resources, especially those involving transformers and 3D convolutions. Optimizations like model pruning, knowledge distillation, and model compression can help mitigate these challenges.

- **Overfitting on Large-Scale Datasets**: Due to the vast number of parameters in hybrid models, overfitting can be an issue, especially with limited training data. Regularization techniques, such as dropout, data augmentation, and pre-training on large-scale datasets, can help improve generalization.

4. PROPOSED METHODOLOGY

Video classification in video processing is remains a challenging task due to the intrinsic complexity of video data, which encompasses both spatial and temporal features. While spatial information can be extracted from individual frames, temporal information arises from the relationships between frames over time. Traditional approaches struggle to simultaneously capture both aspects. Hybrid deep learning techniques have therefore gained prominence, combining the strengths of CNNs—effective at spatial feature extraction—with RNNs or

Transformers, which excel at modelling temporal dependencies. The proposed model integrates the 3D CNN and attention-based LSTM architecture for video processing.

4.1 Integrated 3D CNN and attention-based LSTM

The proposed hybrid architecture first employs a 3D Convolutional Neural Network (3D CNN) to extract rich spatiotemporal features from video frames, capturing both spatial patterns and short-term temporal dynamics. These feature maps are then fed into an attention-based Long Short-Term Memory (LSTM) network, which models long-range temporal dependencies while selectively focusing on the most informative frames and features within the sequence. Finally, the LSTM outputs are passed through fully connected (FC) layers, followed by a softmax layer, to perform classification or regression tasks. This design effectively integrates spatial, temporal, and attention-driven information, enabling robust video representation and improved action recognition performance. Figure 2 gives the overall workflow of the proposed model for video processing, and further sections will provide a breakdown of its key elements which is implemented in human action recognition system.

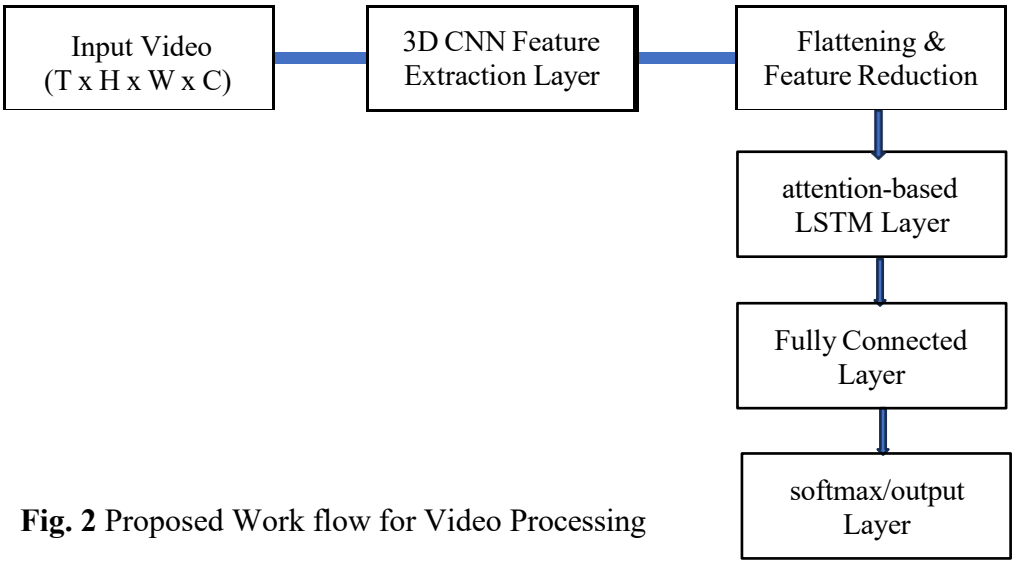


Fig. 2 Proposed Work flow for Video Processing

The workflow of the proposed model generally follows these steps:

- **Input Video**:

A sequence of video frames is represented as a 4D tensor (T×H×W×C), where

  - T = number of frames
  - H = height of the frame
  - W = width of the frame
  - C = number of color channels (e.g., RGB image have 3 channels)

- **3D CNN Layer**:

  - Extracts spatial and short-term temporal features from the video frames.
  - Outputs feature maps that retain spatial and temporal information.

- **Flattening & Feature Reduction** (optional):
  - Converts the 3D feature maps into a flattened sequence for input to the LSTM network.
- **Attention-based LSTM Layer**:
  - The LSTM processes the sequence of features extracted by the 3D CNN.
  - An attention mechanism is applied to help the model focus on the most informative parts of the sequence.
- **Fully Connected (FC) Layer**:
  - Maps the attention-weighted LSTM output to the final classification or regression output (e.g., action recognition, video classification).
- **Softmax/Output Layer**:
  - Outputs the predicted class labels or other relevant results.

4.2 Implementation of 3D CNN and attention-based LSTM Mechanism

The main technology for feature extraction from the videos is 3D Convolutional Neural Networks (3D CNNs). Because 3D CNNs can learn both spatial and temporal information from the videos—two crucial components of action recognition—they are especially well-suited for this job. Following a 3D CNN, an attention mechanism, and a Long Short-Term Memory (LSTM) network are used to analyze the feature extractions. Accurate action categorization is made possible by the combination of 3D CNNs and the LSTM-attention network, which enables to identify the underlying patterns in the movies.

The primary goal of this model is to design and develop an effective deep learning system that utilizes a combination of 3D CNN and an attention-based LSTM mechanism to predict and categorize human behavior or activity. The implementation of 3D CNN and attention-based LSTM mechanism is shown in Figure 3.

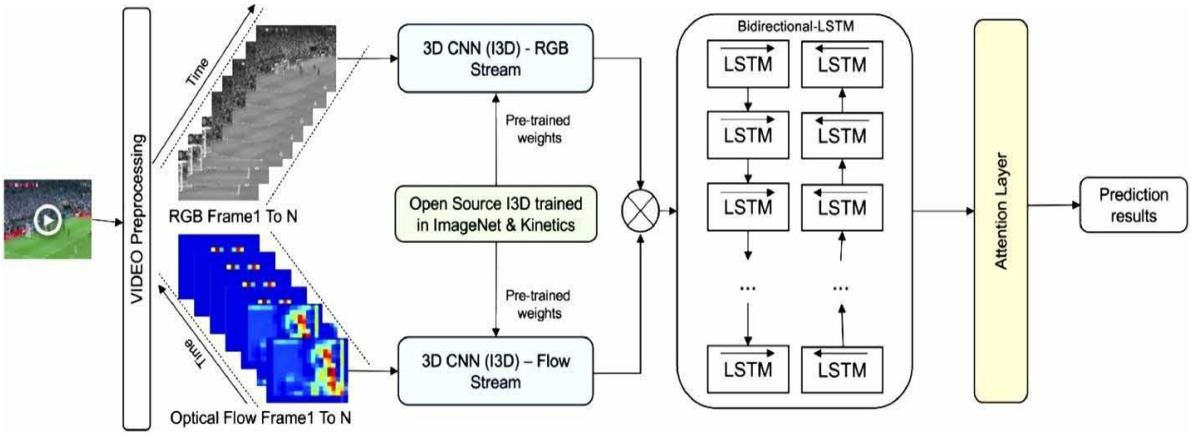


Fig. 3 Implementation of 3D CNN and attention-based LSTM Mechanism

The architecture consists of several key components designed to effectively capture both spatial and temporal features from video data:

- **Video Preprocessing:** Input video sequences are first pre-processed to extract individual RGB frames and optical flow frames. Optical flow captures short-term motion dynamics between consecutive frames, complementing spatial information from RGB frames.
- **3D Convolutional Neural Networks:** Two parallel 3D CNN streams process the video inputs: RGB Stream extracts spatiotemporal features from raw RGB frames and Flow Stream extracts motion-based features from optical flow frames. Both streams leverage pre-trained weights from large-scale datasets such as ImageNet and Kinetics, enabling transfer learning for robust feature extraction.
- **Feature Fusion:** Features from the RGB and flow streams are combined, creating a unified representation that encodes both appearance and motion information.
- **Bidirectional LSTM:** The fused features are fed into a stacked bidirectional LSTM network, which models long-range temporal dependencies in the video sequence. Bidirectional processing ensures that both past and future context within the sequence is captured.
- **Attention Layer:** An attention mechanism is applied to the LSTM outputs to emphasize the most informative frames and temporal regions, improving action recognition accuracy.
- **Prediction Layer:** The attention-weighted features are passed to the final prediction layer, producing class probabilities for action recognition tasks.

The design incorporates a Long Short-Term Memory (LSTM) network with an attention mechanism for accurate action categorization after integrating a 3D Convolutional Neural Network (3D CNN) for feature extraction from video input.

5. RESULTS AND SYSTEM PERFORMANCE

5.1 Datasets

Experiments are performed by utilizing the University of Central Florida 101 (UCF101) [23] and Human Motion Database 51 (HMDB51) [22] datasets to assess the effectiveness of the suggested approach. A difficult test bed for the suggested system, both datasets are often used to assess video action recognition systems. 51 action types and 6766 movies make up the HMDB51 dataset. Movies and online videos were among the sources from which the videos in the dataset were gathered. The dataset contains a variety of human activities, including dancing, running, and walking. The HMDB51 Dataset provides instance of classes which are shown in Figure 4.

Fig. 4 HMDB51 dataset instance of classes.

The movies in the collection vary in length, frame rate, and resolution. About 131 videos are included in each lesson. 13,320 movies in 101 action classes make up the UCF101 dataset. The dataset consists of a variety of human activities, including martial arts, sports, and playing musical instruments, which were gathered from YouTube. The movies in the collection vary in length, frame rate, and resolution. The normal train-test splits supplied by the dataset developers were used for both datasets. 70% of the movies in each dataset are in the trained set, while the remaining 30% are in the test set. This split ratio is often used in the literature to assess action recognition models [22, 23] because it offers a balance between testing and training data, enabling sufficient model assessment, and training. Movies in the test set were used to assess the suggested system's performance, while movies in the train set were used to train the system. The UCF101 Dataset provides instance of classes which are shown in Figure 5.

Fig. 5 A UCF101 dataset instance of classes.

The collection is perfect for testing human activity recognition algorithms since it includes a range of human behaviors that were recorded from various perspectives and resolutions. Because it contains a variety of human actions that were captured from different angles and resolutions, the collection is ideal for testing human activity detection algorithms.

5.2 Experimental Setup

The suggested system was implemented in Python and imported a few popular libraries, including TensorFlow, Keras, and OpenCV. Gradient calculations, which are essential for deep learning models, are handled by TensorFlow, an open-source framework created by Google Brain. A high-level neural network API called Keras is used to develop the architecture of the deep learning models. Real-time computer vision activities like reading and converting movies are done using OpenCV, a library dedicated to this field. For video categorization, these libraries enable the effective use of the I3D + LSTM-Attention network.

The Adam optimization method, which has gained widespread recognition for its effectiveness in deep learning, is used in the suggested system. Adam is an abbreviation for adaptable Moment Estimation, which reflects the adaptable character of this optimization technique [24]. Using Adam's default settings, implemented with a learning rate of 0.001 and beta values of (beta1=0.9, beta2=0.999). These beta parameters allow the optimization method to continue to work over time by regulating the pace at which previous gradient estimations deteriorate. In deep learning training, the number of epochs and batch size must be carefully considered. 32 is the batch size chosen for this investigation because it offers a fair compromise between

computing efficiency and stability. Setting the number of epochs at 100 minimizes the chance of overfitting while enabling the model to converge.

To address intra-class variability in deep learning models, the center loss strategy is used. The learnt feature representations are made more intra-class compact and inter-class separable by this auxiliary loss function. To enhance the suggested model's performance on the video movement classification test by including the center loss in the objective function.

5.3 Evaluation Metrics

The results are usually quantified using the following metrics:

- **Accuracy:** Measures the percentage of correctly classified video sequences.

$$\text{Accuracy} = \frac{T_{pos} + T_{neg}}{T_{pos} + T_{neg} + F_{pos} + F_{neg}}$$

- **Precision, Recall, and F1-Score**: For assessing the performance in detecting specific classes, especially in imbalanced datasets.

$$\text{Precision} = \frac{T_{pos}}{T_{pos} + F_{pos}}$$

$$\text{Recall} = \frac{T_{pos}\, T_{pos}}{+ F_{neg}}$$

$$F1 - \text{Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Confusion Matrix**: Provides insights into which classes are being misclassified and helps identify problematic areas. The confusion matrix is a 2D matrix where each row represents the actual class, and each column represents the predicted class. Table 1 shows the actual classes and predicted classes of confusion matrix.

Table 1. Actual classes and Predicted classes of Confusion matrix.

Class	Predicted Positive	Predicted Negative
Actual Positive	Tpos	Fneg
Actual Negative	Fpos	Tneg

- **AUC-ROC Curve**: Evaluates the model's ability to distinguish between classes, especially in binary classification tasks. Receiver Operating Characteristic (ROC) curve plots True Positive Rate and False Positive Rate.

$$\text{True Positive Rate} = \frac{T_{pos}}{T_{pos} + F_{pos}}$$

$$\text{False Positive Rate} = \frac{F_{pos}}{T_{neg} + F_{pos}}$$

Area Under the Curve (AUC) represents the area under the ROC curve.

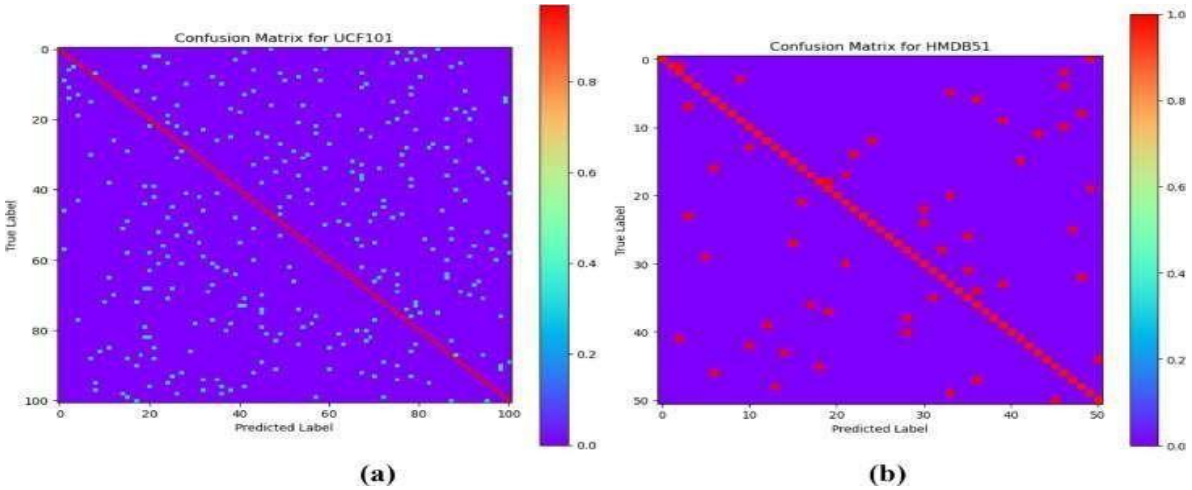
- AUC = 1: Perfect classifier
- AUC = 0.5: Random guess
- AUC < 0.5: Worse than random (likely mislabelled data)

Confusion Matrix Analysis

A confusion matrix can reveal which classes are frequently misclassified. For instance:

- **True Positive (T_{pos}):** Correctly identified actions.
- **False Positive (F_{pos}):** Misclassified non-action frames as actions.
- **False Negative (F_{neg}):** Failed to recognize an action frame.

If the model struggles with certain actions, additional training data or fine-tuning of the attention mechanism may be needed. The performance of the proposed method was evaluated using two widely used datasets for video action recognition, UCF101 and HMDB51. Given that most of the projected labels match the actual labels of the data, it is evident from the confusion matrices that the suggested approach achieved high levels of accuracy. The high true label rate, which was above 97% for both datasets, demonstrates this. The graphics' x-axis displays the expected labels, while the y-axis displays the dataset's actual labels. The number of occurrences that were assigned the appropriate label is shown by the color intensity in each cell. Overall, the confusion matrix results support the class-wise performance histogram results, indicating the efficacy and resilience of the suggested method for video action identification. The model's excellent accuracy on both datasets demonstrates how well it can categorize activities in a variety of video footage. Figures 6a and 6b shows the confusion matrix graph for the UCF101 and HMBD51 datasets, respectively.



(a) **(b)**

Fig. 6. (a) Confusion Matrix graph for UCF101 **(b)** Confusion Matrix graph for HMDB51.

It shows how well the suggested method performs, with real labels on the y-axis and anticipated labels on the x-axis. The color intensity of each cell represents the number of occurrences assigned to the associated label. It gives a summary of the model's sensitivity and positive predictive value by displaying the precision, recall, and F1-score determined for each dataset. With precision scores of 97.77 and 99.05 for UCF101 and HMDB51, respectively, the model demonstrated a high degree of accuracy in identifying the right action in both datasets.

The model was able to recognize most of the activities in the movie, as seen by the recall scores of 95.55% and 99.50% for the two Datasets, respectively. Furthermore, when it considers both precision and recall, the F1-scores for UCF101 and HMDB51 are 94.76% and 99.25%, respectively. This is an excellent indicator of a model's accuracy since it is a harmonic mean of precision and recall. These results show the effectiveness of the suggested method for action recognition in films, and it is broadly applicable to other datasets with multiple classes. The performance indicators for the UCF101 and HMDB51 Datasets are shown in Table 2.

Table 2. Performance Indicators for the UCF101 and HMDB51 Datasets:

Dataset	Precision	Recall	F1-Score
UCF101	97.77	95.55%	94.76%
HMDB51	99.05	99.50%	99.25

The performance assessment per class for the UCF101 and HMDB51 datasets must be examined to provide more detail on the effectiveness of the suggested method for video action recognition. provide a visual representation of this information. The method regularly performs well across all classes with a high degree of accuracy, according to a thorough review of the numbers. Notably is the average class accuracy of more than 90% on difficult Datasets like UCF101 and HMDB51. It is crucial to note that the accuracy may be lower than 90% in certain courses. Numerous factors, including the intricacy of the action itself, the caliber of the films in the collection, and the difficulty of the acts within that class, may be to blame for this. Nonetheless, the suggested technique's total performance across all classes demonstrates how well it handles the complexity of these datasets. Figure 7 shows the performance assessment at the class level for the UCF101 Dataset. In this figure, the vertical axis shows the equivalent % accuracy of the suggested approach, while the horizontal axis indicates the various classes.

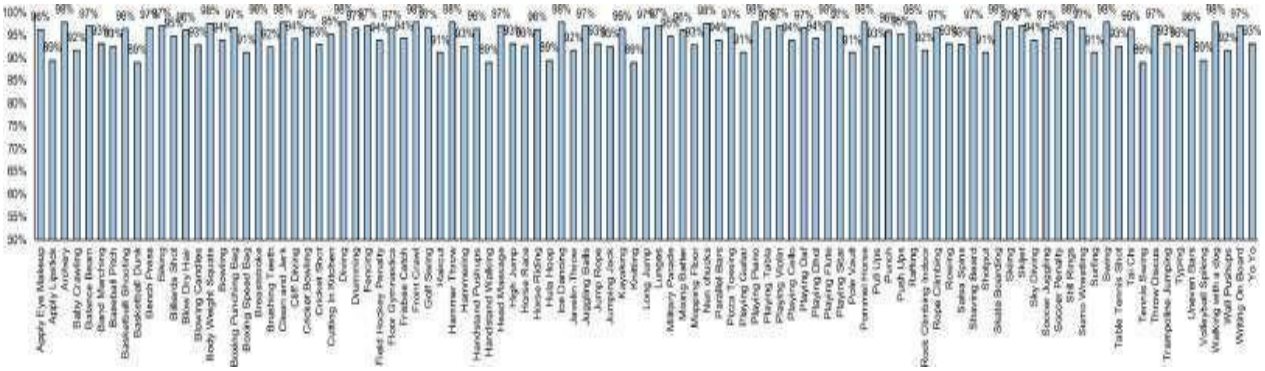


Fig. 7. Performance Assessment at the Class Level for the UCF101 Dataset.

The accuracy of the suggested method for every class in the HMDB51 Dataset is shown by the histogram. The vertical axis shows the relevant accuracy %, while the horizontal axis shows the various classifications. Figure 8 shows the performance assessment at the class level for the HMDB51 Dataset.

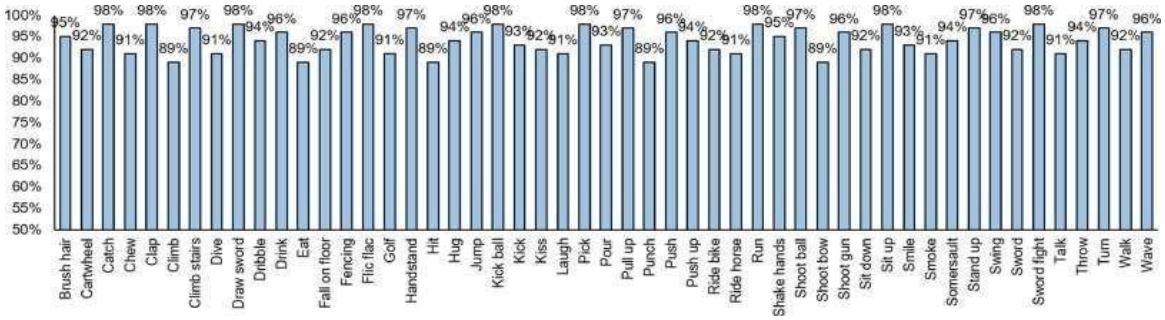


Fig. 8. Performance Assessment at the Class Level for the HMDB51 Dataset:

The ROC (Receiver Operating Characteristic) curve employed to evaluate the proficiency of the proposed method. With an AUC value of 0.97 for the HMDB51 Dataset, the classifier demonstrated a high degree of accuracy and a robust performance in terms of differentiating between true and false positive findings. The classifier's ability to correctly identify behaviors in the HMDB51 dataset is shown by this outcome. In a similar vein, the UCF101 dataset's AUC score of 0.95 showed that the classifier performed well there as well. These values shows that the classifier can effectively differentiate between true and false positive findings in the UCF101 dataset, even if it is not as high as the HMDB51 result. The ROC curve for the UCF101 and HMDB51 Datasets are depicted in figures 9a and 9b.

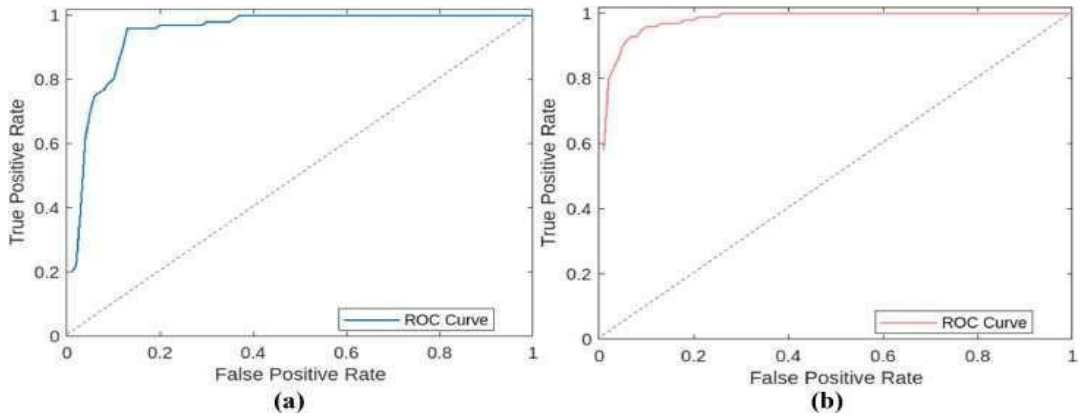


Fig. 9. (a) ROC Curve for the UCF101 (b) ROC Curve for the HMDB51 Datasets.

This classifier can differentiate between true and false positive findings for each dataset is shown by the ROC Curve for the UCF101 and HMDB51 Dataset. This paper also demonstrates the comparison of accuracies of different existing and proposed model on UCF101 and HMDB51 Datasets. Table 3 shows the action recognition accuracies against various models on UCF101 and HMDB51 Datasets.

Table 3. Action recognition accuracies against various models on UCF101 and HMDB51.

Method	**UCF101 accuracy (%)**	**HMDB51 accuracy (%)**
2D CNN + LSTM	99.12	96.23
3D CNN + LSTM	96.19	99.64
3D CNN + LSTM + Attention	91.80	93.01
3D CNN + LSTM + Attention + Center Loss (proposed)	**99.25**	**99.25**

6. CONCLUSION

Large-scale video classification is a challenging task due to the high-dimensional nature of video data, the temporal dynamics between frames, and the large-scale datasets involved. Hybrid deep learning techniques have gained popularity for tackling this challenge because they combine the strengths of multiple approaches, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformers, to process spatial and temporal information in video data.

This study concludes by demonstrating the efficacy of hybrid approach for video action recognition, which is achieved by combining 3D Convolutional Neural Networks (3D CNN), transfer learning with inflated 3D (I3D), and Bidirectional LSTM architecture in a harmonious manner, enhanced by an attention mechanism.

Experiments on the UCF101 and HMDB51 benchmark datasets yielded significant findings. Notably, this method surpassed established models, demonstrating exceptional performance on the HMDB51 dataset. These results underscore the importance of the attention mechanism and bidirectional LSTM architecture in capturing and interpreting complex temporal patterns in videos—an essential factor for accurate action recognition.

7. REFERENCES

[1] Wang, H.; Schmid, C. Action recognition with improved trajectories. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 1–8 December 2013; pp. 3551–3558.

[2] Bilen, H.; Fernando, B.; Gavves, E.; Vedaldi, A.; Gould, S. Dynamic image networks for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3034–3042.

[3] Chen, C.; Liu, K.; Kehtarnavaz, N. Real-time human action recognition based on depth motion maps. J. Real-Time Image Process. 2016, 12, 155–163. [CrossRef]

[4] Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. 2013, 35, 221–231. [CrossRef]

[5] Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Li, F.F. Large-scale video classification with convolutional neural networks. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.

[6] Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. arXiv 2014, arXiv:1406.2199

[7] Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Van Gool, L. Temporal segment networks: Towards good practices for deep action recognition. arXiv 2016, arXiv:1608.00859.

[8] Feichtenhofer, C.; Pinz, A.; Zisserman, A. Convolutional two-stream network fusion for video action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27– 30 June 2016; pp. 1933–1941.

[9] Wang, L.; Qiao, Y.; Tang, X. Action recognition with trajectory-pooled deep-convolutional descriptors. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.

[10] Lu, X.; Yao, H.; Zhao, S. Action recognition with multiscale trajectory-pooled 3D convolutional descriptors. Trans. Multimedia Tools Appl. 2017, 1–17. [CrossRef]

[11] Taylor, G.; Fergus, R.; LeCun, Y.; Bregler, C. Convolutional learning of spatiotemporal features. In Proceedings of the 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010; pp. 140–153.

[12] Fernando, B.; Gavves, E.; Oramas, J.M.; Ghodrati, A.; Tuytelaars, T. Modeling video evolution for action recognition. In Proceedings of the 28th IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5378–5387.

[13] Perronnin, F.; S´anchez, J.; Mensink, T. Improving the Fisher kernel for large-scale image classification. In Proceedings of the 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010; pp. 143–156.

[14] Zaremba, W.; Sutskever, I.; Vinyals, O. Recurrent neural network regularization. arXiv 2014, arXiv:1409.2329.

[15] Donahue, J.; Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; Darrell, T. Long-term recurrent convolutional networks for visual recognition and description. IEEE Trans. Pattern Anal. Mach. Intell. 2017, 39, 677–691. [CrossRef] [PubMed]

[16] Veeriah, V.; Zhuang, N.; Qi, G.J. Differential recurrent neural networks for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Santiago, Chile, 7–13 December 2015; pp. 4041–4049.

[17] Yue-Hei, J.; Hausknecht, M.; Vijayanarasimhan, S. Beyond short snippets: Deep networks for video classification. In Proceedings of the 28th IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4694–4702.

[18] Wu, Z.; Wang, X.; Jiang, Y. Modelling spatial-temporal clues in a hybrid deep learning framework for video classification. In Proceedings of the 23rd ACM International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 07 Issue: 05 | May 2020.

[19] Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. Neural Computation. 1997, 9, 1735–1780. [CrossRef] [PubMed]

[20] Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3D convolutional networks. arXiv 2015, arXiv:1412.0767.

[21] Srivastava, N.; Mansimov, E.; Salakhutdinov, R. Unsupervised Learning of Video Representations using LSTMs. arXiv 2015, arXiv:1502.04681.

[22] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A Large Video Database for Human Motion Recognition IEEE International Conference on Computer Vision (ICCV), 2011.

[23] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A Dataset of 101 Human Actions Classes from Videos in The Wild Report: CRCV-TR-12-01, University of Central Florida, 2012.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Proc. Int. Conf. Learn. Representations (ICLR), San Diego, CA, USA, 2015.