Journal of Universal Computer Science, vol. 30, no. 10 (2024), 1424-1452
submitted: 25/9/2023, accepted: 5/3/2024, appeared: 28/9/2024 CC BY-ND 4.0

Recognition of Real-Time Video Activities Using Stacked Bi-GRU with Fusion-based Deep Architecture

Ujwala Thakur
(Department of Computer Science Engineering & Information Technology, Noida, India
ujwala.thakur@gmail.com)

Ankit Vidyarthi
(Department of Computer Science Engineering & Information Technology, Noida, India
https://orcid.org/0000-0002-8026-4246, dr.ankit.vidyarthi@gmail.com)

Amarjeet Prajapati
(Department of Computer Science Engineering & Information Technology, Noida, India
https://orcid.org/0000-0003-1466-9023, amarjeetnitkkr@gmail.com)

Abstract: Recognizing and understanding human activities in real-time videos is a challenging task due to the complex nature of video data and the need for efficient and accurate analysis. This research pioneers a breakthrough in video activity recognition by introducing a robust framework leveraging the power of a stacked Bidirectional Long Short-Term Memory (Bi-LSTM) and Gated Recurrent Unit (GRU) architecture, harmonized within a fusion-based deep model. The stacked Bi-LSTM-GRU model capitalizes on its dual recurrent architecture, capturing nuanced temporal dependencies within video sequences. The fusion-based deep architecture synergizes spatial and temporal features, enabling the model to discern intricate patterns in human activities. To further enhance the discriminative power of the model, we introduce a fusion module in the proposed deep architecture. The fusion module integrates multi-modal features extracted from different levels of the network hierarchy, allowing for a more comprehensive representation of video activities. We demonstrate the efficacy of our approach through rigorous experimentation on UCF50, UCF101, and HMDB51 datasets. In experiments on the UCF50 dataset, our model achieves an accuracy of 97.01% and 95.86% on training and validation sets respectively, showcasing its proficiency in discerning activities across a diverse range of scenarios. The evaluation extends to the UCF101 dataset, where the proposed approach achieves a competitive accuracy of 97.62% and 96.93% on training and validation sets, surpassing previous benchmarks by a margin of approx 1%. Furthermore, on the challenging HMDB51 dataset, the model demonstrates a robust accuracy of 89.71% and 88.88% on training and validation sets, solidifying its efficacy in intricate action recognition tasks.

Keywords: Human Activity Recognition, Spatiotemporal Analysis, Stacked Deep Model, Bidirectional GRU, Video Activity Analysis
Categories: H.3.1, H.3.2, H.3.3, H.3.7, H.5.1
DOI: 10.3897/jucs.113095

1 Introduction

Real-time video activity recognition plays a crucial role in various domains such as surveillance, video analytics, human-computer interaction, and autonomous systems.

The ability to automatically analyze and understand the activities depicted in video data is essential for efficient decision-making and intelligent systems[Qiu et al. 2022]. However, this task poses significant challenges due to the inherent complexity and variability of video content.

Video activity recognition involves the task of detecting and classifying various human actions, object interactions, and events depicted in video data. It requires the development of sophisticated algorithms that can capture the temporal dynamics and spatial configurations of activities, allowing machines to interpret and respond to visual information in real time.

Traditional approaches to video activity recognition relied heavily on handcrafted features, such as histograms of oriented gradients (HOG), scale-invariant feature transform (SIFT), motion history images (MHI), Grey Level Co-occurrence Matrix (GLCM) [Kuncan et al. 2022], and Local Binary Pattern (LBP) [Kuncan et al. 2019]. These methods often incorporated shallow classifiers, such as support vector machines (SVM) or hidden Markov models (HMM), for activity classification. However, these techniques suffered from limited representational power and struggled to capture the complex and dynamic nature of video activities.

The advent of deep learning, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), revolutionized video activity recognition. Deep learning models have shown remarkable performance in automatically learning discriminative features and modeling the temporal dynamics of video data. CNNs excel at extracting spatial information by learning hierarchical representations, while RNNs effectively capture temporal dependencies by processing sequences of frames [Soni et al. 2023, Vaibhav et al. 2023].

Real-time video activity recognition poses additional challenges due to the need for efficient and low-latency processing. The processing speed becomes crucial, especially in time-critical applications such as video surveillance and autonomous navigation, where timely detection and response are paramount.

In recent years, researchers have explored various techniques to address the challenges of real-time video activity recognition. These include the use of lightweight architectures, optimization techniques such as network pruning and quantization, parallel processing using GPUs or specialized hardware, and the integration of real-time feedback loops to refine and adapt the recognition system on the fly.

The goal of this work is to propose an innovative approach for real-time video activity recognition that combines the strengths of convolution operation, bi-directional Gated Recurrence Unit (Bi-LSTM-GRU), and video frame fusion. The proposed method aims to achieve high accuracy in activity recognition while maintaining low latency and computational efficiency. Also, the proposed approach aims to overcome the limitations of existing methods and improve the accuracy and efficiency of video activity recognition.

However, to capture the temporal dependencies in video sequences, the stacked Bi-LSTM-GRU model is used. The Bi-LSTM-GRU architecture enables the model to adequately simulate the dynamics and long-term interdependence in video activities by allowing bidirectional processing of input frames. The model may capture increasingly complex temporal patterns by stacking numerous Bi-LSTM-GRU layers, boosting representation power.

Also, the model is extended with the inclusion of the fusion layer in the proposed architecture to handle multiple video frames. It combines multi-modal elements derived from several levels of the video frame structure. The model may capture complementary features of video activities by combining both spatial and temporal information, resulting in a more comprehensive and discriminative representation.

The key contributions of this research paper are as follows:

1. Proposed a stacked Bi-LSTM-GRU model for real-time video activity recognition
2. Introduced a fusion layer for analyzing multiple video frames
3. Analyzed the integration of spatial and temporal information of Video activities simultaneously
4. Conducting comprehensive experiments on benchmark datasets for real-time video activity recognition

The remainder of this research paper is organized as follows: Section 2 provides an overview of related work in video activity recognition. Section 3 describes the proposed methodology, including the stacked Bi-LSTM-GRU model and fusion-based deep architecture. Section 4 presents the experimental setup and evaluation results. Finally, Section 5 concludes the paper and discusses future research directions.

2 Literature Survey

Video activity recognition is a research field that focuses on developing algorithms and techniques to automatically detect, classify, and understand activities occurring in video data. It has gained significant attention due to its wide range of applications in areas such as surveillance, video analytics, human-computer interaction, and robotics [Saleem, Usama, and Rana 2023, Yadav et al. 2021, Ullah et al. 2021].

Traditional approaches to video activity recognition relied on handcrafted features and shallow classifiers. These methods involved extracting low-level features, such as motion vectors [Vishnu et al. 2021], color and intensity histograms [Mahjoub and Mohamed 2016], and spatiotemporal interest points [Das, Debapratim, and Soharab 2016, Wang et al. 2021], and designing specific rules or classifiers to recognize activities. For instance, action recognition in videos could be achieved by representing actions as sequences of features and applying techniques like hidden Markov models (HMMs) [Nasfi, and Nizar 2022, Xue, and Hui 2021] or dynamic time warping (DTW) for classification [Mohammadzade et al. 2021, Ning, and Liu 2021].

However, these traditional approaches faced limitations in capturing the complex and dynamic nature of video activities. They often struggled with variations in lighting conditions, camera viewpoints, and object appearances, leading to reduced recognition accuracy. Additionally, handcrafted features lacked the ability to learn discriminative representations directly from raw data, which limited their generalization capability.

With the emergence of deep learning, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), video activity recognition witnessed significant advancements [Zhao, Haider, and Patrick 2017, Singh et al. 2019]. CNNs revolutionized the field by enabling automatic feature extraction from video frames. By using convolutional layers with learnable filters, CNNs learned hierarchical representations that captured spatial patterns and object appearances. This facilitated more effective recognition of activities by providing rich and discriminative features [Srivastava et al. 2022, Srivastava et al. 2021].

RNNs, on the other hand, excelled in capturing the temporal dynamics and dependencies in video sequences. Models such as long short-term memory (LSTM) and gated recurrent units (GRUs) were employed to process sequences of frames and learn temporal patterns over time [Nafea, Wadood, and Ghulam 2022]. By considering the order

and temporal context of frames, RNNs improved the understanding of activities and facilitated more accurate recognition.

Furthermore, the combination of CNNs and RNNs led to the development of two-stream networks. These networks incorporated separate spatial and temporal pathways, where the spatial pathway processed individual frames using CNNs, and the temporal pathway processed sequences of frames using RNNs. The fusion of spatial and temporal information at later stages enabled more comprehensive and robust video activity recognition.

In recent years, there have been efforts to address the computational demands of video activity recognition, particularly in real-time scenarios. Light-weight architectures, such as MobileNet [Tsai, Chi-Yi, and Yu-Kai 2022] and ShuffleNet [Duc-Quang, Ngan, and Jia-Ching 2022], have been proposed to reduce the model size and computational complexity without compromising accuracy significantly. Furthermore, techniques like network pruning, quantization, and knowledge distillation have been employed to optimize deep models and improve their efficiency.

Real-time video activity recognition also involves considerations of low-latency processing. Parallel computing using GPUs and specialized hardware, as well as techniques like model parallelism and model compression, have been explored to accelerate inference time. Additionally, the integration of real-time feedback loops and online learning mechanisms has been investigated to adapt and refine the recognition system in real-time.

In one of the works, the authors introduced the two-stream network architecture [Simonyan et al. 2014], consisting of spatial and temporal pathways, for video activity recognition. The spatial pathway processed individual frames using CNNs, while the temporal pathway analyzed the motion information using optical flow. The fusion of these pathways improved recognition accuracy on benchmark datasets.

In another work, the authors used the Temporal Segment Networks (TSN), which addressed the limitations of previous approaches by efficient sampling and aggregating temporal information in videos [Wang et al. 2016]. TSN employed a sparse temporal sampling strategy and trained on short video segments, achieving state-of-the-art performance on various action recognition datasets.

The authors of the paper [Wang et al. 2019] introduced the Inflated 3D Convolutional Neural Network (I3D) architecture, which extended the 2D CNNs to 3D for video activity recognition. By pre-training on large-scale video datasets and fine-tuning on smaller action recognition datasets, I3D achieved superior performance, demonstrating the effectiveness of 3D convolutional networks.

Several other works used the newly designed deep models for action recognition in video frames, like the work proposed in [Girdhar et al. 2017]. This work introduced ActionVLAD, a method that combined CNNs and VLAD (Vector of Locally Aggregated Descriptors) encoding for video activity recognition. ActionVLAD learned spatiotemporal descriptors using CNNs and aggregated them using VLAD, effectively capturing both appearance and motion information. The approach achieved competitive results on action recognition benchmarks.

In the literature, several authors used temporal information to recognize the actions from video frames correctly. On a similar theme, the work proposed in [Lin, Chuang, and Song 2019] used an efficient approach for temporal modeling in video activity recognition. TSM leveraged the concept of temporal shift operations, which performed channel-wise temporal shifts in feature maps, reducing computational costs. TSM achieved state-of-the-art performance on action recognition tasks while maintaining low latency.

In 2019, there is one work which had used a different network architecture for video

action recognition [Feichtenhofer et al. 2019]. This work introduced SlowFast networks, which incorporated two pathways with different frame rates to effectively capture spatial and temporal information. The Slow pathway processed low frame-rate frames for spatial understanding, while the Fast pathway processed high frame-rate frames for temporal analysis. SlowFast networks achieved top performance on action recognition benchmarks with improved efficiency. Also, the authors of [Novotny et al. 2019], proposed a method for estimating 3D human pose from monocular videos. By leveraging 3D convolutional neural networks, C3DPO learned spatiotemporal features and achieved accurate pose estimation, enabling more detailed analysis of human activities.

Similarly, one more paper investigated different spatiotemporal convolutional architectures for video activity recognition [Tran et al. 2018]. The authors explored various network designs, including 3D convolutions, 2D+1D spatiotemporal convolutions, and non-local networks. They provided insights into the strengths and limitations of different architectures, highlighting the importance of proper temporal modeling. Some authors had also developed lightweight models that were found workable on compressed video datasets [Chao-Yuan et al. 2018, Chen and Chiu 2022, Hu et al. 2020]. One such lightweight and efficient approach for video action recognition is CoViAR which utilizes a combination of 3D convolutions, temporal pooling, and attention mechanisms to capture spatial and temporal information [Chao-Yuan et al. 2018]. It achieved competitive accuracy while significantly reducing computational requirements.

In one of the latest works, the author proposed masked autoencoder networks, which leverage spatiotemporal attention masks to focus on discriminative regions and frames in videos [Qing et al. 2023]. This approach improved the interpretability and robustness of video activity recognition models by attending to relevant spatial and temporal information. Using the transformer-based approaches how the different models can be utilized for action recognition was also discussed in one of the review papers [Ulhaq et al. 2022]. This work extended the success of transformer models in natural language processing to video activity recognition. The paper also explored that by applying vision transformers, which process video frames as sequences, the authors can achieve competitive results on large-scale action recognition datasets. This highlighted the effectiveness of transformers in capturing long-range dependencies.

However, all these notable works represent a fraction of the extensive research in video activity recognition. They showcase advancements in network architectures, temporal modeling techniques, efficient processing, and improved recognition accuracy, contributing to the development of robust and efficient methods for video activity recognition.

3 Proposed Methodology

3.1 Problem Formulation

The problem of video activity recognition involves automatically detecting and classifying activities or actions occurring in video sequences. Given a video input, the goal is to develop a computational model or algorithm that can accurately identify and classify the activities depicted in the video.

Formally, let's define the problem as follows:
Input: A video sequence $V = \{F_1, F_2, ..., F_N\}$, where F_i represents the i^{th} frame of the video. Each frame F_i contains visual information in the form of pixels.
Output: The recognized activity or action A from a predefined set of activity classes $C = \{C_1, C_2, ..., C_M\}$, where M is the total number of activity classes.

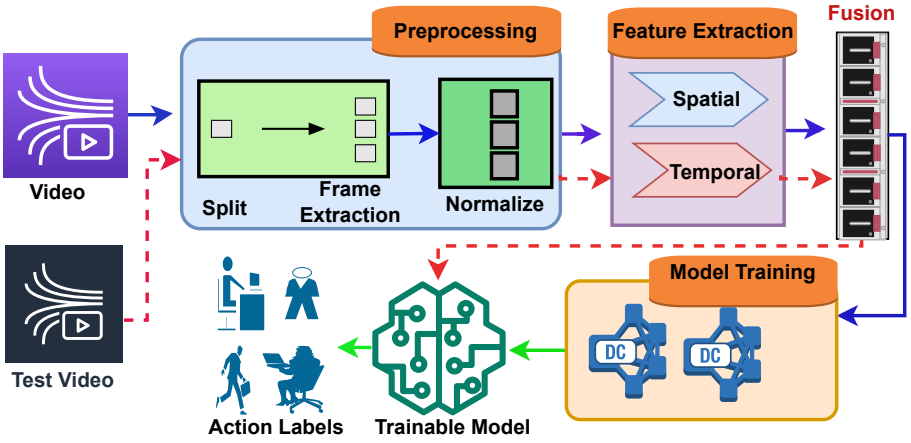


Figure 1: Workflow of the human activity recognition from the video dataset

The problem can be framed as a supervised learning task, where a labeled dataset is available for training the recognition model. The dataset consists of video examples, each associated with a ground truth activity label. Given the labeled training dataset, the objective is to learn a mapping function or model that can generalize well to unseen video data. The model should capture the temporal dynamics, spatial configurations, and contextual information necessary to discriminate and classify different activities accurately.

The problem formulation may also include additional considerations, such as real-time processing requirements, computational efficiency, handling long video sequences, handling variations in lighting conditions or camera viewpoints, and dealing with complex and overlapping activities.

The goal is to design an effective and efficient video activity recognition system that can robustly recognize a wide range of activities in diverse video datasets, enabling applications such as surveillance, video analytics, human-computer interaction, and autonomous systems.

3.2 Framework for Video Activity Recognition

The workflow diagram of the proposed human activity recognition from the videos is presented in Figure 1. The block diagram provided represents a general overview of the pipeline for human activity recognition using video datasets with deep learning.

Some of the key information about the workflow diagram is given as:

1. Video Input: The process begins with a video dataset that contains labeled videos of human activities. The dataset may include multiple videos, each associated with a specific activity class.

2. Data Preprocessing:

   (a) Video Splitting: Split the videos into shorter clips or segments, typically containing a few seconds to a few minutes of activity.

(b) Frame Extraction: Extract individual frames from the segmented video clips.

(c) Resize and Normalize: Resize the frames to a consistent size and normalize pixel values to ensure uniformity.

3. Feature Extraction:
   (a) Spatial Features: Apply a pre-trained convolutional neural network (CNN) to extract spatial features from each frame. This can involve using deep learning models to capture high-level visual representations.

   (b) Temporal Features: Capture the temporal dynamics by processing frames sequentially to capture motion information between frames.

4. Feature Fusion: Combine the spatial and temporal features extracted from each frame to create a comprehensive representation of the video activity. This fusion can be achieved through concatenation, element-wise operations, or other fusion mechanisms.

5. Model Training:
   (a) Split the dataset into training and validation sets.

   (b) Design and train a deep learning model using the extracted and fused features.

   (c) Optimize the model using suitable loss functions.

   (d) Monitor the model's performance on the validation set and adjust hyperparameters as necessary.

6. Model Evaluation:
   (a) Evaluate the trained model on a separate test set to assess its performance.

   (b) Compute metrics such as accuracy, precision, recall, or F1 score to measure the model's effectiveness in recognizing activities.

7. Inference and Real-Time Recognition: Apply the trained model to new, unseen videos for activity recognition. Analyze the performance of the system and identify areas for improvement. Fine-tune the models, adjust hyperparameters, or consider alternative architectures to enhance the recognition accuracy.

3.3 Proposed Deep Model Architecture for the Multi-class Video Activity Recognition

The heart of the proposed framework is the designing of the new model that can fetch the multimodel features from the videos. The model consists of deep convolution blocks with the fusion of the parameters to gain a diverse feature set for learning, validation, and testing. The internal architecture of the newly designed model is shown in Figure 2.

In the traditional approaches, where the video frames are processed sequentially, the problem is that the model will not always be completely sure in its forecast of each video frame, therefore the predictions will alter quickly and fluctuate. This is because the model does not examine the entire video sequence but rather classifies each frame individually.

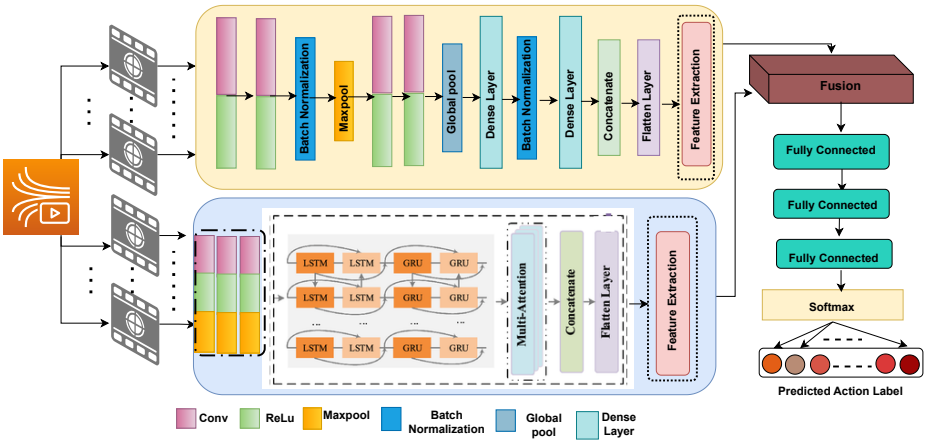


Figure 2: Internal architecture of the proposed model

The solution to this problem is addressed in the proposed approach. It is to be noted that the proposed architecture is composed of two parallel processing units. In the first processing unit, individual video frames are processed using the designed convolution network architecture to find out the features using each frame. Later, the fusion layer will cascade all the feature units extracted from each video frame.

Similarly, the other processing unit of the model architecture is used for processing the video frames in a batch of predefined size. Since here the temporal information is to be processed thus here the LSTM-BiGRU network architecture is to be used for feature engineering. At the last, all video frame batches features are fused to get the cascaded feature pool having viability.

Finally, the two different feature sets, from both pipelines, are again fused to get the collective features having a vast amount of variations. These features are used for the recognition of human activities in videos.

The important aspect of the proposed approach is handling the video frames in an order that minimizes the limitations of the traditional approaches, as discussed earlier. To handle this, an easy solution adopted here is to compute average findings over some frames, say n, instead of classifying and showing results for a single frame. That is how flashing would be effectively eliminated. Once we've determined the value of n, we may utilize a basic approach like the moving average/rolling average to achieve our goal.

Whereas, using the model to learn environmental context instead of the actual action sequence to forecast is disastrous and will result in overfitting. This is also why the procedures described above will fail when the actions are similar. Consider the actions of *getting out of a chair* and *getting into a chair*. The frames in both actions are nearly identical. The order of the frame sequence is the primary difference. Thus we need a piece of temporal information to handle such situations and predict correctly the action involved in the video.

Each of the model pipelines is using the feature concatenation with the help of the late fusion layer. In practice, the Late Fusion strategy is fairly similar to the Single-Frame CNN approach, although slightly more difficult. The sole difference is that in the Single-Frame CNN approach, the average across all predicted probabilities is done after the network has completed its job, but in the Late Fusion approach, averaging is embedded

into the network itself. As a result, the temporal structure of the frame sequence is also considered.

The last Fusion layer is used to combine the output of independent networks that operate on temporally distant frames. It is typically accomplished using the max pooling, average pooling, or flattening technique. This method enables the model to learn both spatial and temporal information about the presence and movement of objects in a scene. Each stream performs picture (frame) classification independently, and the projected scores are then fused using the fusion layer.

3.4 Algorithm of proposed methodology

The stepwise description of the proposed methodology is presented in the form of the pseudo-algorithm and the same is given in Algorithm 1. The given algorithm is a six-step algorithm where the first four steps are the common steps to be used in the model training and model testing. Step 5 of the algorithm is the training of the model using the training video dataset using the global fused feature set extracted from the two pipelines of the proposed model. At last, once the model is trained, in step 6, the model is evaluated using the real-time video clips from the testing dataset. The final output of step 6 is the predicted output of the activity involved in the test data clip.

4 Experimental Evaluation and Comparison

4.1 Dataset Description

4.1.1 UCF50 Dataset

Over the internet, there were many video datasets but most of the action recognition data sets that were currently accessible are staged by actors and are not found realistic. In this work, the target was to explore the real video datasets for model training and thus we have ended the search on the UCF50 dataset[1]. It is one of the realistic action recognition datasets made up of videos that were downloaded from YouTube.

Due to the wide variations in camera motion, item appearance and posture, object scale, viewpoint, cluttered background, illumination conditions, etc., this data set is said to be exceedingly difficult. The videos are divided into 25 groups for each of the 50 categories, with each group containing at least four action clips. The video clips in a given group might all have the same subject, a comparable setting, a comparable point of view, etc.

The 50 action categories in the UCF50 data set that were gathered from YouTube are *Baseball Pitch, Basketball Shooting, Bench Press, Biking, Biking, Billiards Shot, Breaststroke, Clean and Jerk, Diving, Drumming, Fencing, Golf Swing, Playing Guitar, High Jump, Horse Race, Horse Riding, Hula Hoop, Javelin Throw, Juggling Balls, Jump Rope, Jumping Jack, Kayaking, Lunges, Military Parade, Mixing Batter, Nun chucks, Playing Piano, Pizza Tossing, Pole Vault, Pommel Horse, Pull-Ups, Punch, Push Ups, Rock Climbing Indoor, Rope Climbing, Rowing, Salsa Spins, Skate Boarding, Skiing, Skijet, Soccer Juggling, Swing, Playing Tabla, TaiChi, Tennis Swing, Trampoline Jumping, Playing Violin, Volleyball Spiking, Walking with a dog, and Yo Yo*.

Further, the 50 action classes of the dataset are categorized into the 5 groups i.e. 1) Sports and Recreation, 2) Everyday Activities, 3) Human-Computer Interaction, 4) Extreme and Uncommon Activities, and 5) Dance and Performing Arts.

[1] https://www.crcv.ucf.edu/data/UCF50.php

Algorithm 1: Pseudo algorithm of the proposed methodology

```
Input: Video frames or clips
Output: Predicted Activity Recognition Label
Data: Training set Tr, Testing set Ts
Step 1: Initialize preprocessing                    // Preprocess video frames
Read the video clip
Extract the frames from the video clip
Step 2: Initialize Spatial pipeline                 // Spatial Feature Extraction
spatial-features = []
for each frame in video-frames: do
    spatial-features = extract-spatial-features(frame)
for each feature in spatial-features: do
    cascade-features1 = Fusion(feature)                          // Local fusion
Step 3: Initialize Temporal pipeline                // Temporal Feature Extraction
temporal-features = []
set frames-batch-size
for each batch i in frames-batch: do
    temporal-features[i] = stacked-LSTM-BiGru(spatial-features[i])
for each feature in temporal-features: do
    cascade-features2 = Fusion(feature)                          // Local fusion
Step 4: Initialize Global Fusion                    // Fusion of Spatial and Temporal
 Features
fused-features = fuse-features(cascade-features1, cascade-features2)
Step 5: Initialize Training                                      // Model Training
train(fused-features, labels)
Step 6: Model testing                                // Real-time Recognition
for each video clip Ts: do
    while video-stream.isrunning(): do
        frame = video-stream.get-next-frame()
        preprocessed-frame = preprocess-frame(frame)
        spatial-features = extract-spatial-features(preprocessed-frame)
        cascade-features1 = Fusion(feature)
        temporal-features = stacked-LSTM-BiGru(spatial-features)
        cascade-features2 = Fusion(feature)
        fused-features = fuse-features(cascade-features1, cascade-features2)
        predicted-label = classify(fused-features)
        display-predicted-label(predicted-label)     // Real-time predicted
         output
```

4.1.2 UCF101 Dataset

A collection of 101 action categories and realistic action videos from YouTube make up the action recognition data set known as UCF101 [Idrees et al. 2017]. The UCF50 data

collection, which includes 50 activity categories, is expanded upon by this set of data. With 13320 videos spanning 101 action categories, UCF101 offers the widest variety of actions available. It is also the most difficult data set to work with due to the wide range of variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, and other factors.

The 101 action categories' worth of videos are divided into 25 groups, with four to seven action videos per category. Videos belonging to the same category could have some things in common, such as a backdrop or point of view that's comparable. With this, five categories can be distinguished from the activity categories i.e. 1) Human-Item Communication, 2) Only Body Motion, 3) Interaction Between Humans, 4) Accompanying with Instruments, and 5) Athletics.

4.1.3 HMDB51 Dataset

A sizable collection of realistic videos from a variety of sources, including films and online videos, makes up the Human Motion Database (HMDB51 dataset). The dataset consists of 6,766 video clips from 51 action categories with at least 101 clips in each category. A single action per clip, a primary actor's minimum height of 60 pixels, a minimum contrast level, a minimum clip duration of one second, and allowable compression artifacts are just a few examples of the minimal quality standards the dataset should uphold. The actions categories can be grouped into five types 1) General facial actions, 2) Facial actions with object manipulation, 3) General body movements, 4) Body movements with object interaction, and 5) Body movements for human interaction. The other details of the datasets that are used in the study are given in Table 1.

Parameter	**UCF50**	**UCF101**	**HMDB51**
Dataset Size	3GB	6.5GB	2GB
Action Categories	50	101	51
Groups of Videos per Action Category	25	25	25
Average Videos per Action Category	133	180	101
Average Number of Frames per Video	200	200	200
Average Frames Width per Video	320	320	320
Average Frames Height per Video	240	240	240
Average Frames Per Seconds per Video	25	25	25
Validation Set	80:20 Split	80:20 Split	80:20 Split

Table 1: Description of the datasets used in the experimentation study

4.2 Model Training and Parameter Setting

In the model training using the proposed deep architecture, several environmental parameters are used for learning the data sample. Initially, the dataset is split into two subsets i.e. *Training* and *Validation* in the ratio of 80-20. However, to test the model performance the test data is the real-time videos extracted from the YouTube platform. The training and validation datasets are processed with the proposed LSTM-based model for learning

the different video actions. To do this, several hyperparameters must be defined in the proposed LSTM-based model for video action recognition to set the architecture and training procedure. These hyperparameters significantly influence the performance and behavior of the LSTM-based video action recognition model. Careful selection and tuning of these parameters are essential for better accuracy and generalization on the video action recognition task. Additionally, experimentation and hyperparameter search may be necessary to find the optimal configuration for a video action recognition problem.

The list of some of the major hyperparameters used in the experimentation and their definition is given below:

Number of LSTM Layers: The number of LSTM layers stacked on top of each other. Increasing the number of layers can allow the model to capture more complex temporal dependencies but can also increase the computational complexity.

Number of LSTM Units: The number of memory cells or hidden units in each LSTM layer. Higher numbers can enhance the model's capacity to capture intricate temporal patterns but may also increase training time and computational requirements.

Dropout Rate: The dropout rate applied between LSTM layers or between the recurrent connections within an LSTM layer. Dropout helps to regularize the model and prevent overfitting by randomly setting a fraction of the LSTM units to zero during training.

Recurrent Dropout Rate: The dropout rate applied to the recurrent connections within an LSTM layer. It helps regularize the LSTM's recurrent connections specifically.

Activation Function: The activation function is applied within the LSTM units. Common choices include the hyperbolic tangent (tanh) function or the rectified linear unit (ReLU).

Bidirectional: Whether to use a bidirectional LSTM, which processes sequences in both forward and backward directions.

Sequence Length: The number of time steps or frames considered in each sequence.

Learning Rate: The rate at which the model updates its weights during training. A higher learning rate can accelerate convergence but may risk overshooting the optimal weights. Conversely, a lower learning rate can lead to more accurate updates but may increase training time.

Batch Size: The number of samples processed in each training iteration. Larger batch sizes can improve training efficiency but may require more memory, while smaller batch sizes can provide more stable updates at the cost of increased training time.

Number of Epochs: The number of complete passes through the training dataset during training. Increasing the number of epochs allows the model to see the data multiple times, but there is a trade-off between training time and model performance.

Loss Function: The objective function that measures the discrepancy between the predicted and actual labels during training. For video classification, common choices include categorical cross-entropy or weighted variants to handle class imbalance.

Early Stopping: A parameter that determines whether to stop training early based on a specified condition, such as validation loss not improving.

Optimizer: The optimization algorithm used to update the model's weights based on the calculated loss. Popular optimizers for LSTM training include stochastic gradient descent (SGD), Adam, or RMSprop.

Learning Rate Scheduler: A scheduler that adjusts the learning rate over time to improve convergence during training.

Weight Initialization: The strategy used to initialize the weights of the LSTM units. Common approaches include random initialization or using pre-trained weights from a related task or domain.

Hyperparameter	InSearch Range	Setup Value
Input Shape	1024x1024 - 64x64	128x128
Output Units	50	50
Count of LSTM Layers	1,3,5,7	3
Count of LSTM Units	12,24,48,96	24
Dropout Rate	0.3,0.4,0.5,0.6,0.7	0.4
Recurrent Dropout Rate	0.1,0.2,0.3,0.4,0.5	0.2
Activation Function	ReLU, tanh, sigmoid	tanh, sigmoid
Bidirectional	Yes, No	Yes
Sequence Length	6, 12, 18, 24, 30	24
Learning Rate	0.1, .01, 0.001, 0.05, 0.005	0.005
Batch Size	4, 8, 16, 32	32
Count of Epochs	10 - 200	100
Loss Function	MSE, Cross entropy	Cross entropy
Early Stopping	Yes, No	Yes
Optimizer	Adam, SGD, RMSprop	Adam
Learning Rate Scheduler	Constant learning rate, Learning rate decay	Learning rate decay
Weight Initialization	0, 0.4, 1, Transfer Learning	Transfer Learning
Loss Weights	L1, L2	L2

Table 2: Hyperparameters description with their values used in experimentation

Sequence Length: The length of the input sequence (number of video frames) provided to the LSTM model. It can affect the model's ability to capture long-term dependencies and may need to be carefully chosen based on the characteristics of the video data.

Loss Weights: Weights assigned to different classes to handle class imbalance.

Input Shape: The shape of the input data, which depends on the number of frames and other input dimensions.

Output Units: The number of output units, corresponding to the number of classes in the video classification task.

The experimental setup for the used hyperparameters and their values is presented in Table 2.

4.3 Dataset Preprocessing and Normalization

Before learning, video datasets often require preprocessing to prepare them for deep learning models. Also, normalizing the video dataset before learning is also a common preprocessing step in deep learning tasks, including video action recognition. Video normalization helps to make the data compatible with the learning algorithms, aids in faster convergence during training, and can prevent certain features from dominating the learning process due to their scale.

In the experimentation, before learning, the dataset is preprocessed to make it fit for the model. In machine learning, preprocessing steps should be applied consistently to maintain data integrity and ensure fair evaluation of the model's performance. The common procedures that were used in the preprocessing are video segmentation, frame extraction, crop & resizing, temporal aggregation, and normalization.

Video Segmentation divides the full-length videos into shorter clips or segments. This can be done based on fixed time intervals or by detecting action boundaries within the video. Segmentation allows the model to focus on specific actions or events within the videos. Later, we extract individual frames from the segmented video clips. These frames serve as input to the deep learning model. The frame extraction process can involve sampling frames at regular intervals or based on specific criteria.

Next, we resized the extracted frames to a consistent size suitable for the deep learning model. This helps ensure uniform input dimensions across all frames. Additionally, if required, cropping can be performed to focus on the relevant region of interest within the frame. Also, we used the temporal aggregation that aggregates the frames within each video clip to obtain a fixed-length representation. This can involve techniques such as mean pooling, max pooling, or concatenation of features from multiple frames. Temporal aggregation captures the temporal dynamics of the video and reduces the variable-length sequences to fixed-length inputs.

Lastly, the dataset is normalized where the pixel values of the frames are set to a standardized scale. This step typically involves techniques like pixel normalization, min-max scaling, or per-channel normalization to ensure that the pixel values fall within a desired range. It's important to perform normalization based on statistics computed from the training set and apply the same normalization to the validation and test sets to ensure consistency. Also, normalization should be done after any other preprocessing steps (e.g., resizing, cropping) to avoid data distortion.

The choice of normalization technique depends on the characteristics of the video dataset and the specific requirements of the deep learning model being used. In general, experimentation and analysis of the dataset can help determine the most appropriate normalization strategy for a given video classification task. In this work, we have used the *Temporal Normalization*. It normalizes the video frames across the temporal dimension (time) for videos with varying lengths. This can involve padding or truncating the videos to a fixed length and applying normalization across frames. Later, we normalize the pixel values of each frame to have zero mean and unit variance. This is often achieved by subtracting the mean and dividing it by the standard deviation of pixel values across the entire dataset or a specific set of frames used for normalization.

4.4 Performance Evaluation Metrics

To obtain the evaluation results on video activity recognition using the proposed model, we have used some of the common evaluation metrics like:

1. Accuracy: The proportion of correctly classified video samples or frames over the total number of samples. It provides an overall measure of the model's classification performance. It is given as:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{1}$$

   Where TP: True Positives (correctly classified positive instances), TN: True Negatives (correctly classified negative instances), FP: False Positives (incorrectly classified positive instances), FN: False Negatives (incorrectly classified negative instances)

2. Precision, Recall, and F1-score: These metrics are commonly used for evaluating multi-class classification tasks. Precision represents the proportion of correctly pre-

dicted positive instances, recall (also known as sensitivity) represents the proportion of actual positive instances correctly predicted, and F1-score is the harmonic mean of precision and recall. These are given as:

$$\begin{aligned} Precision &= \frac{TP}{(TP+FP)} \\ Recall &= \frac{TP}{(TP+FN)} \\ F1-score &= \frac{2*(Precision*Recall)}{(Precision+Recall)} \end{aligned} \tag{2}$$

3. Mean Average Precision (MAP): Often used for evaluating video activity recognition tasks with multiple classes. It calculates the average precision for each class and then takes the mean over all classes. MAP provides an aggregate measure of the model's performance across all classes and is given by:

$$MAP = \sum_{c} \frac{1}{|C|} (\sum_{i} \frac{Precision_i}{C_i}) \tag{3}$$

4. Intersection over Union (IoU): It is used in video action localization tasks that measure the overlap between predicted and ground truth bounding boxes or regions of interest. Higher IoU values indicate better localization accuracy. It is given by:

$$IoU = \frac{Area(predicted \bigcap groundtruth)}{Area(predicted \bigcup groundtruth)} \tag{4}$$

4.5 Evaluation Results

4.5.1 Results with UCF50 Dataset

The first experimentation of the proposed model is performed with the UCF50 dataset having 50 action classes. The experimentation results of the proposed model using the above-defined matrices on the UCF50 dataset are presented in Table 3.

	Training set	**Validation set**
Accuracy	97.01%	95.86%
Loss	2.99%	4.14%
Precision	97.52%	95.82%
Recall	96.37%	96.22%
F1-score	96.94%	96.01%
MAP	96.28%	96.05%
IoU	97.5%	96.6%

Table 3: Experimental results of the proposed approach on the UCF50 dataset

Here all the results presented in Table 3 were computed on an epoch size of 100. The epoch-wise evaluation of the accuracy and the loss is presented in Figure 3. The first part

of Figure 3(A) presents the total training accuracy vs total validation accuracy per epoch. While the total training loss vs total validation loss per epoch is presented in Figure 3(B). It is noted from the figure results that the model performance gradually increased with the epoch size. However, after the iteration count of 100 the model performance almost converges with the slight change in the accuracy on 3 to 4 unit place after decimal. Thus, we took only an epoch size of 100 under consideration.

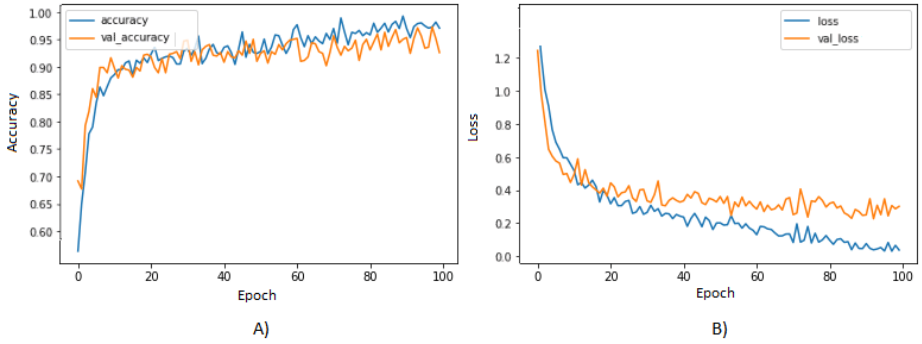


Figure 3: Epoch-wise demonstration of the accuracy and loss on UCF50 dataset

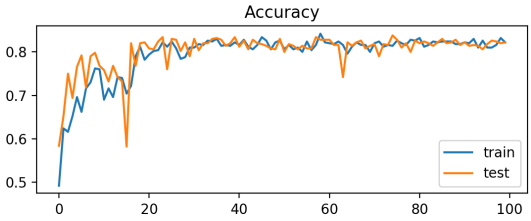
4.5.2 Results with HMDB51 Dataset

In the second experimentation, we assessed the performance of our proposed model on the HMDB51 dataset using a comprehensive set of evaluation metrics, including accuracy, loss, precision, recall, F-score, Mean Average Precision (MAP), and Intersection over Union (IoU). These metrics collectively provide a thorough understanding of the model's ability to recognize activities within video sequences. The experimentation results of the proposed model on HMDB51 dataset is presented in Table 4.

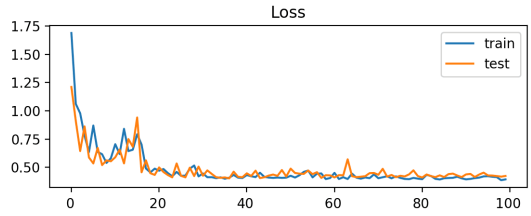
	Training set	**Validation set**
Accuracy	89.71%	88.88%
Loss	10.29%	11.12%
Precision	87.31%	85.37%
Recall	85.55%	84.89%
F1-score	86.42%	85.12%
MAP	84.44%	84.15%
IoU	85.5%	84.1%

Table 4: Experimental results of the proposed approach on the HMDB51 dataset

Using the experimental result values, the proposed model demonstrated a remarkable accuracy of 89.71% on the HMDB51 dataset. This accuracy was achieved through



(a) Accuracy per epoch



(b) Loss per epoch

Figure 4: Experimental description of the proposed model performance on HMDB51 dataset considering per epoch count

extensive training and validation, indicating the model's proficiency in distinguishing between the various human activities represented in the dataset. The corresponding loss function reached a minimum value of 10.29%, showcasing the model's convergence during the training process. Whereas, on the validation set the model gives an accuracy of 88.88% and the loss of 11.12%.

On the other hand, precision, recall, and F-score metrics were calculated to assess the model's ability to correctly identify positive instances while minimizing false positives and false negatives. The precision value of 87.31% indicates the percentage of correctly predicted positive instances, while the recall value of 85.55% highlights the model's effectiveness in capturing all actual positive instances. The harmonic mean of precision and recall, the F-score, achieved 86.42%, providing a balanced measure of the model's classification performance. The average precision for each class was computed, resulting in an overall MAP value of 84.44%. Also, the IoU score of 85.5% indicates the degree of overlap between predicted and ground truth temporal boundaries, showcasing the model's effectiveness in localizing activities within video sequences.

The performance of the proposed model when evaluated on per epoch count is found promising as the accuracy of the model on HMDB51 dataset continuously increasing and the corresponding loss is gradually decreasing. The results of the proposed model on the run of epoch count 100 is shown in Figure 4.

4.5.3 Results with UCF101 Dataset

The third experiment of the proposed model is tested with one of the benchmark dataset of action classes having 101 classes, called UCF101 dataset. To evaluate the performance of our proposed model on the challenging UCF101 dataset, we conducted extensive experiments, employing a range of evaluation metrics to assess its accuracy and robustness in real-world video activity recognition tasks and the corresponding results are presented in

Table 5. Our model, designed to address the complexities of activity recognition in videos, demonstrates promising performance across various metrics, showcasing its efficacy in capturing both spatial and temporal features for improved classification accuracy.

	Training set	**Validation set**
Accuracy	97.62%	96.93%
Loss	2.38%	3.07%
Precision	95.72%	95.27%
Recall	96.12%	95.43%
F1-score	95.91%	95.34%
MAP	96.85%	95.64%
IoU	97.50%	96.86%

Table 5: Experimental results of the proposed approach on the UCF101 dataset

In the experimentation, the proposed model gives an training accuracy of 97.62% on the UCF101 dataset, showcasing its proficiency in correctly classifying a diverse set of activities within video sequences. The corresponding loss function, optimized during training, converged effectively, reaching a minimal value of 2.38%, indicative of the model's ability to generalize well to unseen instances. While in the validation of the model with test set, the model gives an accuracy of 96.93% and the corresponding loss is 3.07%.

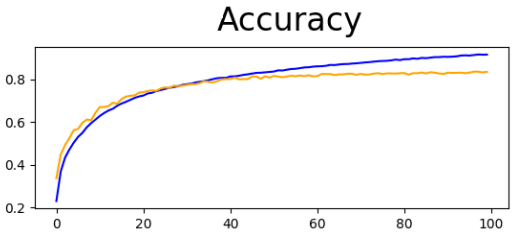
Besides this, the model also performs well on other metrics like precision - 95.72%, Recall - 96.12%, F1-Score - 95.91%, MAP - 96.85%, and IoU - 97.5%. The achieved metrics underscore its efficacy in accurately classifying a wide array of activities, making it a promising solution for applications ranging from video surveillance to human-computer interaction. Also, the performance of the model on UCF101 dataset per epoch count is presented in the Figure 5.

4.5.4 Experimental results of the model on individual group classes

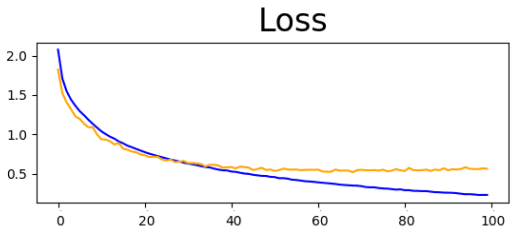
In an another experimentation the model is tested to find the individual class accuracy's of the datasets used in experimentation i.e. UCF50, UCF101, and HMDB51. Irrespective of the number of individual classes in each of these datasets are quite large, these classes were grouped in a way such that in each group there are classes having similar information. All the used datasets were analyzed and further divided into 5 groups such as UCF50 is divided into 1) Sports and Recreation, 2) Daily Routines, 3) Human-Object Interactions (HOI), 4) Extreme and Uncommon Activities, and 5) Dance and Performing Arts. The individual class accuracy of the dataset is presented in Table 6 and its corresponding confusion matrix for validation/testing set is presented in Figure 6.

The HMDB51 is divided into 5 groups as 1) General facial actions, 2) Facial actions with object manipulation, 3) General body movements, 4) Body movements with object interaction, and 5) Body movements for human interaction. The individual class performance of the HMDB dataset is presented in Table 7 and its corresponding confusion matrix is given in Figure 7.

The last dataset is the UCF101 dataset having 101 classes which were divided into the 5 groups as 1) Human-Item Communication, 2) Only Body Motion, 3) Interaction



(a) Accuracy per epoch



(b) Loss per epoch

Figure 5: Experimental description of the proposed model performance on UCF101 dataset considering per epoch count

Between Humans, 4) Accompanying with Instruments, and 5) Athletics. The individual class accuracy of the dataset is presented in Table 8 and its corresponding confusion map is presented in Figure 8.

4.5.5 Model validation with randomly selected videos from external sources

However, to test the model performance we have taken random videos from YouTube falling in any one of the class labels used in model training. To prepare the testing set we took 100 random videos, 2 videos per class. This video dataset is passed to the model

Class Name	Precision	1-Precision	Recall	1-Recall	f1-score
Class1	0.9818	0.0182	0.9474	0.0526	0.9643
Class2	0.9630	0.0370	0.9630	0.0370	0.9630
Class3	0.9817	0.0183	0.9554	0.0446	0.9683
Class4	0.9304	0.0696	0.9907	0.0093	0.9596
Class5	0.9524	0.0476	0.9524	0.0476	0.9524
Accuracy	0.9616				
Misclassification Rate	0.0384				
Macro-F1	0.9615				
Weighted-F1	0.9616				

Table 6: Results of the proposed model on individual classes of UCF50 dataset having 5 groups

Validation Set						
TARGET / OUTPUT	Class1	Class2	Class3	Class4	Class5	SUM
Class1	108 19.74%	1 0.18%	0 0.00%	0 0.00%	1 0.18%	110 98.18% 1.82%
Class2	2 0.37%	104 19.01%	1 0.18%	0 0.00%	1 0.18%	108 96.30% 3.70%
Class3	1 0.18%	0 0.00%	107 19.56%	0 0.00%	1 0.18%	109 98.17% 1.83%
Class4	2 0.37%	2 0.37%	2 0.37%	107 19.56%	2 0.37%	115 93.04% 6.96%
Class5	1 0.18%	1 0.18%	2 0.37%	1 0.18%	100 18.28%	105 95.24% 4.76%
SUM	114 94.74% 5.26%	108 96.30% 3.70%	112 95.54% 4.46%	108 99.07% 0.93%	105 95.24% 4.76%	526 / 547 96.16% 3.84%

Figure 6: Confusion matrix of UCF50 dataset having 5 group categories

to predict the output class for the videos. The model predicts the 94 times correct result thus having the test accuracy of 94%. From the results, it was noted that the model performance was good in predicting the actions having variations. While the videos have similar action patterns like Pull-ups and Push-ups, Rock climbing and Rope climbing, and Swing and Tennis Swing were not properly handled by the model and still need improvement. The experimentation results with the testing set on randomly selected ten videos were presented in Table 9.

4.6 Comparative Analysis with Existing Literature

The comparative analysis of the proposed approach with some of the existing literature's using the same UCF50 dataset is presented in Table 10. Based on the evaluated results from the research articles and directly used in Table 10, it was found that the proposed model's performance is much better than the existing works of literature on UCF50 validation dataset. However, it is to be considered in the comparative analysis that all the used literature must have used the UCF50 dataset only.

On the other hand, we present a comparative analysis of our proposed method for video activity recognition with existing literature, specifically focusing on the widely used UCF101 and HMDB51 datasets. Our proposed method achieved an accuracy of 96.93% on the UCF101 dataset, outperforming the state-of-the-art methods. Whereas, on the

Class Name	Precision	1-Precision	Recall	1-Recall	f1-score
Class1	0.9029	0.0971	0.8857	0.1143	0.8942
Class2	0.8597	0.1403	0.8636	0.1364	0.8617
Class3	0.8744	0.1256	0.8657	0.1343	0.8700
Class4	0.8718	0.1282	0.8629	0.1371	0.8673
Class5	0.8660	0.1340	0.8960	0.1040	0.8808
Accuracy	0.8748				
Misclassification Rate	0.1252				
Macro-F1	0.8748				
Weighted-F1	0.8748				

Table 7: Results of the proposed model on individual classes of HMBD51 dataset having 5 groups

Class Name	Precision	1-Precision	Recall	1-Recall	f1-score
Class1	0.9683	0.0317	0.9774	0.0226	0.9728
Class2	0.9547	0.0453	0.9582	0.0418	0.9564
Class3	0.9540	0.0460	0.9646	0.0354	0.9593
Class4	0.9691	0.0309	0.9654	0.0346	0.9672
Class5	0.9689	0.0311	0.9487	0.0513	0.9587
Accuracy	0.9628				
Misclassification Rate	0.0372				
Macro-F1	0.9629				
Weighted-F1	0.9628				

Table 8: Results of the proposed model on individual classes of UCF101 dataset having 5 groups

HMDB51 dataset, our proposed method achieved an accuracy of 88.88%, surpassing the baseline accuracy's as reported in the literature. The comparative study of the proposed method with the existing literature's is presented in Table 11.

5 Conclusion and Future Work

This work introduced a new approach to video activity recognition through the development of a stacked Bi-LSTM-GRU with a fusion-based deep architecture. The integration of bidirectional Long Short-Term Memory (Bi-LSTM) and Gated Recurrent Unit (GRU) layers aims at capturing intricate temporal dependencies and nuanced patterns within video sequences, thereby enhancing the model's ability to discern complex activities. The fusion-based deep architecture further amalgamates spatial and temporal features, enabling a holistic understanding of dynamic actions in real-world scenarios.

One distinctive aspect of our study lies in the explicit incorporation of both LSTM and GRU units within the same model architecture. This hybrid approach harnesses the strengths of both recurrent architectures — the memory retention capabilities of LSTM and the computational efficiency of GRU. The bidirectional nature facilitates

Validation Set						
TARGET / OUTPUT	Class1	Class2	Class3	Class4	Class5	SUM
Class1	186 18.06%	5 0.49%	3 0.29%	7 0.68%	5 0.49%	206 90.29% 9.71%
Class2	11 1.07%	190 18.45%	6 0.58%	11 1.07%	3 0.29%	221 85.97% 14.03%
Class3	4 0.39%	8 0.78%	174 16.89%	7 0.68%	6 0.58%	199 87.44% 12.56%
Class4	6 0.58%	5 0.49%	7 0.68%	170 16.50%	7 0.68%	195 87.18% 12.82%
Class5	3 0.29%	12 1.17%	11 1.07%	2 0.19%	181 17.57%	209 86.60% 13.40%
SUM	210 88.57% 11.43%	220 86.36% 13.64%	201 86.57% 13.43%	197 86.29% 13.71%	202 89.60% 10.40%	901 / 1030 87.48% 12.52%

Figure 7: Confusion matrix of HMDB51 dataset having 5 group categories

	Actual Action	**Predicted Action**	**Status**
Video 1	Diving	Diving	Correct
Video 4	Playing Guitar	Playing Guitar	Correct
Video 10	Biking	Biking	Correct
Video 14	Javelin Throw	Javelin Throw	Correct
Video 23	Pull-ups	Push-ups	Incorrect
Video 37	Walking with dog	Walking with dog	Correct
Video 55	Tennis Swing	Swing	Incorrect
Video 71	Rope Climbing	Rock Climbing	Incorrect
Video 86	Yo Yo	Yo Yo	Correct
Video 97	Punch	Punch	Correct

Table 9: Experimental results of the proposed approach on the testing dataset with a random selection of videos

Training Set						
TARGET / OUTPUT	Class1	Class2	Class3	Class4	Class5	SUM
Class1	519 19.48%	6 0.23%	4 0.15%	3 0.11%	4 0.15%	536 96.83% 3.17%
Class2	2 0.08%	527 19.78%	7 0.26%	5 0.19%	11 0.41%	552 95.47% 4.53%
Class3	8 0.30%	6 0.23%	518 19.44%	4 0.15%	7 0.26%	543 95.40% 4.60%
Class4	0 0.00%	6 0.23%	5 0.19%	502 18.84%	5 0.19%	518 96.91% 3.09%
Class5	2 0.08%	5 0.19%	3 0.11%	6 0.23%	499 18.73%	515 96.89% 3.11%
SUM	531 97.74% 2.26%	550 95.82% 4.18%	537 96.46% 3.54%	520 96.54% 3.46%	526 94.87% 5.13%	2565 / 2664 96.28% 3.72%

Figure 8: Confusion matrix of UCF101 dataset having 5 group categories

capturing temporal dependencies in both forward and backward directions, ensuring a comprehensive analysis of the temporal evolution of activities.

The differentiation of our study from existing works is rooted in the careful consideration of the interplay between model complexity and interpretability. While deep architectures have demonstrated remarkable performance in capturing intricate patterns, the interpretability of such models has often been a challenge. By adopting a fusion-based approach, we strike a balance between model complexity and interpretability, enabling not only superior predictive performance but also an enhanced understanding of the underlying dynamics driving action recognition.

Through extensive experiments on a benchmark dataset of UCF50, UCF101, and HMDB51, we demonstrated the effectiveness of our proposed model. In contrast to existing studies, our method showcases superior performance across the three benchmark datasets. On UCF50, our model achieves an accuracy of 97.01% and 95.86% on training and validation sets respectively, outperforming the state-of-the-art by 6.5%. When tested on the larger UCF101 dataset, the model demonstrates a remarkable accuracy of 97.62% and 96.93% on training and validation sets, surpassing previous benchmarks by a margin of approx 1%. Furthermore, on the HMDB51 dataset, our approach achieves a competitive accuracy of 89.71% and 88.88% on training and validation sets, exceeding baseline methods by a significant margin of 15%.

Furthermore, we evaluated the real-time performance of our approach on a live video

Reference	Experimental Setup	Accuracy (%)
[Kishore, and Mubarak 2013]	Leave One Group Out Cross-validation (25 cross-validations)	76.90
[Sreemanananth, and Jason 2012]	5-fold group-wise cross-validation	57.90
[Sreemanananth, and Jason 2012]	Video Wise Cross-validation	76.40
[Amer et al. 2012]	2/3 training and 1/3 testing per class	81.03
[Berkan, Assari, and Mubarak 2013]	Leave One Group Out Cross-validation (25 cross-validations)	73.70
[Kliper-Gross et al. 2012]	Leave One Group Out Cross-validation (25 cross-validations)	72.60
[Ahmad, Akhtar, and Kim 2020]	K-ary Tree Hashing with ray optimization	80.9
[Meng et al. 2020]	Hierarchical sparse coding with SVM	89.3
[Lei, and Xiang 2020]	Two stage neural network	88.0
Proposed	Two channel spatio-temporal	95.86

Table 10: Comparative study of the proposed method with existing literature on UCF50 dataset

stream, demonstrating its applicability in dynamic scenarios. The stacked Bi-LSTM-GRU model with fusion-based architecture exhibited efficient processing capabilities, enabling accurate activity recognition in real time. Our research contributes to video activity recognition by providing a deep learning-based framework that effectively captures spatiotemporal information. The proposed approach offers potential applications in various domains, including video surveillance, human-computer interaction, and action recognition in video streams.

Despite the promising results, it is essential to acknowledge certain limitations of our developed technique. The model's performance might be influenced by dataset-specific nuances, and further investigation on more diverse datasets is warranted. Additionally, the computational requirements for training the stacked Bi-LSTM-GRU could pose challenges for real-time applications, suggesting the need for optimization strategies.

As we delve into the future directions of this research, addressing the limitations identified remains paramount. Exploring strategies to enhance the model's efficiency, such as parallelization, attention mechanism, and transfer learning, could pave the way for seamless integration into real-world applications. Furthermore, extending the evaluation to datasets with increased complexity and incorporating interpretability mechanisms will contribute to a more comprehensive understanding of the model's capabilities.

Reference	Model	Pre-Train	HMDB51	UCF101
[Diba et al. 2018]	STC	K400	72.6	95.8
[zolfaghari et al. 2018]	ECO	K400	68.4	93.6
[Carreira et al. 2017]	I3D	ImageNet+ K400	74.8	95.6
[Ju et al. 2022]	S3D	ImageNet+ K400	75.9	96.8
[Feichtenhofer et al. 2019]	SlowOnly-8x8-R101	Kinetics+ OmniSource	79	97.3
[Ju et al. 2022]	Video Prompt	CLIP	66.4	93.6
[Pan et al. 2022]	ST-Adapter ViT-B/16	CLIP+K400	77.7	96.4
[Wang et al. 2022]	TSN	ImageNet	68.5	94
[Lin, Chuang, and Song 2019]	TSM	Kinetics	70.7	95.9
[Zhang et al. 2022]	MEST	Kinetics	73.4	96.8
Proposed	stacked Bi-LSTM-GRU	-	88.88	96.93

Table 11: Comparative analysis of the proposed model with existing literature's on UCF101 and HMDB51 dataset

Compliance with Ethical Standards

Conflict of Interest:

The authors of this manuscript declare that there is no conflict of interest.

Data Availability Statement:

The dataset and code generated during and/or analyzed during the current study are available from the corresponding author or reasonable request.

Funding Information:

The author declares that there is no funding associated for this project.

Ethics Statement:

The author of this manuscript confirms that: (i) Informed, written consent has been obtained from the relevant sources wherever is required; (ii) All procedures followed were in accordance with the ethical standards of the responsible committee on human experimentation (institutional and national) and with the Helsinki Declaration of 1964 and its later amendments.

References

[Ahmad, Akhtar, and Kim 2020] Jalal, Ahmad, Israr Akhtar, and Kibum Kim. "Human posture estimation and sustainable events classification via pseudo-2D stick model and K-ary tree hashing." Sustainability 12, no. 23 (2020): 9814.

[Amer et al. 2012] Amer, Mohamed R., Dan Xie, Mingtian Zhao, Sinisa Todorovic, and Song-Chun Zhu. "Cost-sensitive top-down/bottom-up inference for multiscale activity recognition." In Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV 12, pp. 187-200. Springer Berlin Heidelberg, 2012.

[Berkan, Assari, and Mubarak 2013] Solmaz, Berkan, Shayan Modiri Assari, and Mubarak Shah. "Classifying web videos using a global video descriptor." Machine vision and applications 24 (2013): 1473-1485.

[Carreira et al. 2017] Carreira, Joao, and Andrew Zisserman. "Quo vadis, action recognition? a new model and the kinetics dataset." In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6299-6308. 2017.

[Chao-Yuan et al. 2018] Wu, Chao-Yuan, Manzil Zaheer, Hexiang Hu, R. Manmatha, Alexander J. Smola, and Philipp Krähenbühl. "Compressed video action recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6026-6035. 2018.

[Chen and Chiu 2022] Chen, Jiawei, and Chiu Man Ho. "MM-ViT: Multi-modal video transformer for compressed video action recognition." In Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp. 1910-1921. 2022.

[Das, Debapratim, and Soharab 2016] Das Dawn, Debapratim, and Soharab Hossain Shaikh. "A comprehensive survey of human action recognition with spatio-temporal interest point (STIP) detector." The Visual Computer 32 (2016): 289-306.

[Diba et al. 2018] Diba, Ali, Mohsen Fayyaz, Vivek Sharma, M. Mahdi Arzani, Rahman Yousefzadeh, Juergen Gall, and Luc Van Gool. "Spatio-temporal channel correlation networks for action classification." In Proceedings of the European Conference on Computer Vision (ECCV), pp. 284-299. 2018.

[Duc-Quang, Ngan, and Jia-Ching 2022] Vu, Duc-Quang, Ngan TH Le, and Jia-Ching Wang. "(2+ 1) d distilled shufflenet: A lightweight unsupervised distillation network for human action recognition." In 2022 26th International Conference on Pattern Recognition (ICPR), pp. 3197-3203. IEEE, 2022.

[Feichtenhofer et al. 2019] Feichtenhofer, Christoph, Haoqi Fan, Jitendra Malik, and Kaiming He. "Slowfast networks for video recognition." In Proceedings of the IEEE/CVF international conference on computer vision, pp. 6202-6211. 2019.

[Girdhar et al. 2017] Girdhar, Rohit, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. "Actionvlad: Learning spatio-temporal aggregation for action classification." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 971-980. 2017.

[Hu et al. 2020] Hu, Hezhen, Wengang Zhou, Xingze Li, Ning Yan, and Houqiang Li. "MV2Flow: Learning motion representation for fast compressed video action recognition." ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 16, no. 3s (2020): 1-19.

[Idrees et al. 2017] Idrees, Haroon, Amir R. Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. "The thumos challenge on action recognition for videos "in the wild"." Computer Vision and Image Understanding, vol. 155. pp. 1-23, 2017.

[Ju et al. 2022] Ju, Chen, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. "Prompting visual-language models for efficient video understanding." In European Conference on Computer Vision, pp. 105-124. Cham: Springer Nature Switzerland, 2022.

[Kishore, and Mubarak 2013] Reddy, Kishore K., and Mubarak Shah. "Recognizing 50 human action categories of web videos." Machine vision and applications 24, no. 5 (2013): 971-981.

[Kliper-Gross et al. 2012] Kliper-Gross, Orit, Yaron Gurovich, Tal Hassner, and Lior Wolf. "Motion interchange patterns for action recognition in unconstrained videos." In Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12, pp. 256-269. Springer Berlin Heidelberg, 2012.

[Kuncan et al. 2019] Kuncan, Fatma, Yılmaz Kaya, and Melih Kuncan. "A novel approach for activity recognition with down-sampling 1D local binary pattern." Advances in Electrical and Computer Engineering 19, no. 1 (2019): 35-44.

[Kuncan et al. 2022] Kuncan, Fatma, Yılmaz Kaya, Ramazan Tekin, and Melih Kuncan. "A new approach for physical human activity recognition based on co-occurrence matrices." The Journal of Supercomputing 78, no. 1 (2022): 1048-1070.

[Lei, and Xiang 2020] Zhang, Lei, and Xuezhi Xiang. "Video event classification based on two-stage neural network." Multimedia Tools and Applications 79 (2020): 21471-21486.

[Lin, Chuang, and Song 2019] Lin, Ji, Chuang Gan, and Song Han. "Tsm: Temporal shift module for efficient video understanding." In Proceedings of the IEEE/CVF international conference on computer vision, pp. 7083-7093. 2019.

[Mahjoub and Mohamed 2016] Mahjoub, Amel Ben, and Mohamed Atri. "Human action recognition using RGB data." In 2016 11th International Design & Test Symposium (IDT), pp. 83-87. IEEE, 2016.

[Meng et al. 2020] Meng, Quanling, Heyan Zhu, Weigang Zhang, Xuefeng Piao, and Aijie Zhang. "Action recognition using form and motion modalities." ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 16, no. 1s (2020): 1-16.

[Mohammadzade et al. 2021] Mohammadzade, Hoda, Soheil Hosseini, Mohammad Reza Rezaei-Dastjerdehei, and Mohsen Tabejamaat. "Dynamic time warping-based features with class-specific joint importance maps for action recognition using Kinect depth sensor." IEEE Sensors Journal 21, no. 7 (2021): 9300-9313.

[Nafea, Wadood, and Ghulam 2022] Nafea, Ohoud, Wadood Abdul, and Ghulam Muhammad. "Multi-sensor human activity recognition using CNN and GRU." International Journal of Multimedia Information Retrieval 11, no. 2 (2022): 135-147.

[Nasfi, and Nizar 2022] Nasfi, Rim, and Nizar Bouguila. "Indoor Activity Recognition Using a Hybrid Generative-Discriminative Approach with Hidden Markov Models and Support Vector Machines." In 2022 IEEE International Conference on Industrial Technology (ICIT), pp. 1-6. IEEE, 2022.

[Ning, and Liu 2021] Ning, Bai, and Liu Na. "Deep Spatial/temporal-level feature engineering for Tennis-based action recognition." Future Generation Computer Systems 125 (2021): 188-193.

[Novotny et al. 2019] Novotny, David, Nikhila Ravi, Benjamin Graham, Natalia Neverova, and Andrea Vedaldi. "C3dpo: Canonical 3d pose networks for non-rigid structure from motion." In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7688-7697. 2019.

[Pan et al. 2022] Pan, Junting, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. "St-adapter: Parameter-efficient image-to-video transfer learning." Advances in Neural Information Processing Systems, vol. 35, pp. 26462-26477, 2022.

[Qing et al. 2023] Qing, Zhiwu, Shiwei Zhang, Ziyuan Huang, Xiang Wang, Yuehuan Wang, Yiliang Lv, Changxin Gao, and Nong Sang. "Mar: Masked autoencoders for efficient action recognition." IEEE Transactions on Multimedia (2023).

[Qiu et al. 2022] Qiu, Sen, Hongkai Zhao, Nan Jiang, Zhelong Wang, Long Liu, Yi An, Hongyu Zhao, Xin Miao, Ruichen Liu, and Giancarlo Fortino. "Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges." Information Fusion 80 (2022): 241-265.

[Saleem, Usama, and Rana 2023] Saleem, Gulshan, Usama Ijaz Bajwa, and Rana Hammad Raza. "Toward human activity recognition: a survey." Neural Computing and Applications 35, no. 5 (2023): 4145-4182.

[Simonyan et al. 2014] Simonyan, Karen, and Andrew Zisserman. "Two-stream convolutional networks for action recognition in videos." Advances in neural information processing systems, 27, 2014.

[Singh et al. 2019] Singh, Roshan, Alok Kumar Singh Kushwaha, Rajat Khurana, and Rajeev Srivastava. "Activity Recognition by Delving deeper using CNN and RNN." In 2019 4th International Conference on Information Systems and Computer Networks (ISCON), pp. 610-614. IEEE, 2019.

[Soni et al. 2023] Soni, Vaibhav, Shashank Jaiswal, Vijay Bhaskar Semwal, Bholanath Roy, Dilip Kumar Choubey, and Dheeresh K. Mallick. "An Enhanced Deep Learning Approach for Smartphone-Based Human Activity Recognition in IoHT." In Machine Learning, Image Processing, Network Security and Data Sciences: Select Proceedings of 3rd International Conference on MIND 2021, pp. 505-516. Singapore: Springer Nature Singapore, 2023.

[Sreemanananth, and Jason 2012] Sadanand, Sreemanananth, and Jason J. Corso. "Action bank: A high-level representation of activity in video." In 2012 IEEE Conference on computer vision and pattern recognition, pp. 1234-1241. IEEE, 2012.

[Srivastava et al. 2021] Srivastava, Anugrah, Tapas Badal, Apar Garg, Ankit Vidyarthi, and Rishav Singh. "Recognizing human violent action using drone surveillance within real-time proximity." Journal of Real-Time Image Processing 18 (2021): 1851-1863.

[Srivastava et al. 2022] Srivastava, Anugrah, Tapas Badal, Pawan Saxena, Ankit Vidyarthi, and Rishav Singh. "UAV surveillance for violence detection and individual identification." Automated Software Engineering 29, no. 1 (2022): 28.

[Tran et al. 2018] Tran, Du, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. "A closer look at spatiotemporal convolutions for action recognition." In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 6450-6459. 2018.

[Tsai, Chi-Yi, and Yu-Kai 2022] Tsai, Chi-Yi, and Yu-Kai Su. "MobileNet-JDE: a lightweight multi-object tracking model for embedded systems." Multimedia Tools and Applications 81, no. 7 (2022): 9915-9937.

[Ulhaq et al. 2022] Ulhaq, Anwaar, Naveed Akhtar, Ganna Pogrebna, and Ajmal Mian. "Vision transformers for action recognition: A survey." arXiv preprint arXiv:2209.05700 (2022).

[Ullah et al. 2021] Ullah, Amin, Khan Muhammad, Weiping Ding, Vasile Palade, Ijaz Ul Haq, and Sung Wook Baik. "Efficient activity recognition using lightweight CNN and DS-GRU network for surveillance applications." Applied Soft Computing 103 (2021): 107102.

[Vaibhav et al. 2023] Soni, Vaibhav, Himanshu Yadav, Vijay Bhaskar Semwal, Bholanath Roy, Dilip Kumar Choubey, and Dheeresh K. Mallick. "A novel smartphone-based human activity recognition using deep learning in health care." In Machine Learning, Image Processing, Network Security and Data Sciences: Select Proceedings of 3rd International Conference on MIND 2021, pp. 493-503. Singapore: Springer Nature Singapore, 2023.

[Vishnu et al. 2021] Vishnu, Chalavadi, Rajeshreddy Datla, Debaditya Roy, Sobhan Babu, and C. Krishna Mohan. "Human fall detection in surveillance videos using fall motion vector modeling." IEEE Sensors Journal 21, no. 15 (2021): 17162-17170.

[Wang et al. 2016] Wang, Limin, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. "Temporal segment networks: Towards good practices for deep action recognition." In European conference on computer vision, pp. 20-36. Springer, Cham, 2016.

[Wang et al. 2019] Wang, Xianyuan, Zhenjiang Miao, Ruyi Zhang, and Shanshan Hao. "I3d-lstm: A new model for human action recognition." In IOP Conference Series: Materials Science and Engineering, vol. 569, no. 3, p. 032035. IOP Publishing, 2019.

[Wang et al. 2021] Wang, Jiaming, Zhenfeng Shao, Xiao Huang, Tao Lu, Ruiqian Zhang, and Xianwei Lv. ”Spatial–temporal pooling for action recognition in videos.” Neurocomputing 451 (2021): 265-278.

[Wang et al. 2022] Wang, Shiqi, Suen Guan, Hui Lin, Jianming Huang, Fei Long, and Junfeng Yao. ”Micro-Expression Recognition Based on Optical Flow and PCANet+.” Sensors, vol. 22, no. 11, pp. 4296, 2022.

[Xue, and Hui 2021] Xue, Tingting, and Hui Liu. ”Hidden Markov Model and its application in human activity recognition and fall detection: A review.” In International Conference in Communications, Signal Processing, and Systems, pp. 863-869. Singapore: Springer Singapore, 2021.

[Yadav et al. 2021] Yadav, Santosh Kumar, Kamlesh Tiwari, Hari Mohan Pandey, and Shaik Ali Akbar. ”A review of multimodal human activity recognition with special emphasis on classification, applications, challenges and future directions.” Knowledge-Based Systems 223 (2021): 106970.

[Zhang et al. 2022] Zhang, Yi. ”MEST: An Action Recognition Network with Motion Encoder and Spatio-Temporal Module.” Sensors, vol. 22, no. 17, pp. 6595, 2022.

[Zhao, Haider, and Patrick 2017] Zhao, Rui, Haider Ali, and Patrick Van der Smagt. ”Two-stream RNN/CNN for action recognition in 3D videos.” In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4260-4267. IEEE, 2017.

[zolfaghari et al. 2018] Zolfaghari, Mohammadreza, Kamaljeet Singh, and Thomas Brox. ”Eco: Efficient convolutional network for online video understanding.” In Proceedings of the European conference on computer vision (ECCV), pp. 695-712. 2018.