http://dx.doi.org/10.21172/ijiet.261.03

# A Unified Approach to Human Activity Recognition: Leveraging CNN and LSTM Networks

Bidesh Chakraborty
*Department of Computer Science and Engineering*
*Haldia Institute of Technology, Haldia, West Bengal, India*

**Abstract- This study introduces a novel approach for human activity recognition (HAR), crucial across various domains like human-computer interaction, robotics, surveillance systems, and more. It focuses on integrating both temporal and spatial features for effective representation. To achieve this, Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) Networks are utilized to extract spatial and temporal features from video datasets. A deep learning model incorporating both CNN and LSTM was developed and trained on the UCF50-Action Recognition dataset. The experimental results demonstrate the effectiveness and superiority of this approach compared to existing methods on the UCF-50 dataset, achieving an accuracy of 94.14%.**

Keywords – **HAR, Deep Learning, CNN, LSTM**

## I. INTRODUCTION

HAR stands as a prominent and captivating area of research within computer vision and deep learning (DL). It serves as a foundational technology with diverse applications spanning intelligent monitoring, human-computer interaction, robotics, digital entertainment, and healthcare. Despite its significance, accurately recognizing human activities poses considerable challenges due to factors like varying lighting conditions, complex backgrounds, and diverse perspectives. Existing approaches often struggle to achieve high accuracy, primarily due to the extensive range of activities and the highly unconstrained nature of the video datasets.

The review of HAR literature can be categorized into two main streams: traditional, feature-based techniques and DL-based techniques. Traditional methods typically involve a sequence of steps, including preprocessing to remove noise and outliers, extracting low-level features from the preprocessed data, and classification to map these features to specific activity classes. However, this review paper specifically focuses on DL-based approaches, as the presented model relies on deep features. Nonetheless, readers interested in exploring further can refer to comprehensive assessments of baseline HAR methods[8].

In this study, DL serves as the foundation for Activity Recognition. Various image classification models, such as AlexNet [1], GoogleNet [2], and VGGNet [3], have emerged from the ImageNet Large Visual Recognition Challenge (ILSVRC). These models, primarily based on CNNs, have proven effective not only in classifying images but also in excelling at diverse tasks like object detection [4], scene labeling [5], and activity recognition [6].

Activity recognition can utilize data from various sources such as accelerometers, sensors, images, or video frames. Collecting data from sensors often requires individuals to wear multiple sensors at different body locations. The collected data undergoes various processing steps, including segmentation and feature extraction, which can be particularly challenging for sensor data.

The success of neural networks heavily relies on the quality of the dataset used for training. Datasets designed to train neural networks for activity classification play a crucial role in the performance of these networks. To develop an architecture that yields accurate results, it is essential to have an appropriate dataset tailored to the specific problem. One such dataset is UCF50 [7], which was created at the University of Central Florida by expanding the UCF11 Action dataset to include activities like basketball and other sports.

This paper proposes a fusion network, utilizing CNN and LSTM, to extract spatial and temporal information from real-life action recognition datasets like UCF50. The standard computer vision pipeline consists of two main steps: activity feature representation and activity recognition. Activity feature representation involves extracting essential details from the video as features, a critical step in the recognition process as the quality of the features directly impacts the recognition outcome. The feature vector obtained in this stage serves as input for the activity recognition phase, where an algorithm learns parameters and classifies activities based on the learned features.

An analysis of the UCF-50 dataset was conducted, yielding impressive results. The significant contributions of this paper are as follows:

1. Unlike current HAR approaches, this paper proposes a unique mechanism that uses a CNN approach to capture action videos, followed by a sequential learning method, achieving new state-of-the-art accuracy.
2. CNN is employed to detect spatial features, while LSTM is used to identify temporal correlations among these features, enhancing the accuracy of HAR. Effective HAR relies heavily on both spatial and temporal features.
3. The proposed framework's effectiveness is evaluated on the challenging UCF-50 dataset, achieving an experimental state-of-the-art accuracy of 94.14%.

The subsequent sections of the work are structured as follows: Section II covers the Literature Review. Sections III and IV detail the Proposed Methodology and Experimental Results, respectively. Section V offers the conclusion at the end.

## II. LITERATURE RIVIEW

In paper [9], the authors designed a deep neural network architecture capable of recognizing human activities in videos by utilizing action bank features from the UCF50 database. Meanwhile, the authors of paper [10] developed 3D CNN models for activity recognition. These models perform 3D convolutions, generating features that account for both spatial and temporal dimensions. To achieve this, a deep architecture is required to produce multiple channels of information from the surrounding input frames, allowing convolution and subsampling to be performed separately in each channel. The data from all channels is then combined to create the final feature representation.

In paper [11], the authors focused on behavior recognition by classifying behaviors based on spatiotemporal parameters. They introduced a new spatiotemporal interest point and analyzed various cuboid descriptors. These cuboid prototypes were used to develop a more effective and reliable behavior descriptor.

In paper [12], the authors proposed a model to evaluate the performance of CNNs in video classification. They found that the slow fusion model outperforms early and late fusion models, as its performance is not solely dependent on architectural characteristics. Given the dynamic nature of videos, which require complex processing due to their temporal extent, the model simplifies classification by treating each video as a collection of fixed-size clips. This approach allows the spatio-temporal features to be effectively understood by extending network connectivity in the time dimension. The study focuses on three primary connectivity classifications: Early Fusion, Late Fusion, and Slow Fusion. The model was tested using the UCF-101 and Sports1M datasets.

In paper [13], the authors proposed two video classification mechanisms that integrate frame-level CNN outputs into video-level predictions, allowing entire videos to be processed in one go. This video classification model effectively employs temporal feature pooling, representing a bag-of-words. Motion attribute-based images are computed at each time frame, quantized, and then pooled across time. Various pooling techniques are examined, with features from specific layers aggregated accordingly. However, a high number of gradients can cause fully connected layer pooling and average pooling to fail. The model utilizes datasets like Sports-1M and UCF-101 in conjunction with LSTM networks for video classification.

In paper [14], the authors introduced the LRCN model, which is a class of models that are deep both spatially and temporally. These models are flexible and can be applied to a wide range of vision tasks requiring sequential inputs and outputs. The evaluation of the LRCN model was conducted using the UCF101 dataset, which involves the classification and categorization of videos into various human action classes.

In paper [15], the authors proposed a method that involves the study of recurrent neural networks (RNNs), where links between nodes form a directed graph, making them suitable for time series applications. This approach trains a model to capture temporal dynamic behavior, incorporating a memory segment to handle input sequences of varying lengths. DL networks are used to transform and extract basic features from the input data. These features are further optimized and reduced using a sparse autoencoder (SAE) network. However, a drawback of SAE is that its sensing capability depends on input quality. To address this, LSTM-based RNNs are employed. At the end of the process, features can also be extracted using models built on the LSTM-RNN architecture. The Softmax Regression Algorithm is used as the classifier, with the trained network's coefficients serving as tools for further training.

In paper [16], the authors proposed a comprehensive DL-based architecture for activity recognition, specifically using a CNN-LSTM network. This architecture enables more accurate prediction of human activities from raw data while simplifying the model and eliminating the need for complex feature engineering. The CNN-LSTM network is deep both spatially and temporally. The model achieves 92% accuracy on the public UCI HAR dataset. It performs well compared to previously proposed deep neural network (DNN) architectures and machine learning (ML) models that rely on manually generated feature datasets.

A CNN-LSTM model for two classes was developed by the authors of study [17]. They used their own dataset to validate the model, and they evaluated its performance in terms of error and classification accuracy against that of existing DL and ML models, such as SVM and LSTM.

The reviewed papers explore different DL techniques for human activity and behavior recognition. Approaches include deep neural networks using action bank features, 3D CNN models for spatial and temporal dimensions, and new spatiotemporal interest points for behavior recognition. Some models use slow fusion to handle video classification, while others integrate frame-level CNN outputs to generate video-level predictions with temporal feature pooling. The LRCN model addresses sequential vision tasks, and RNNs with LSTM are used for time series applications. CNN-LSTM architectures are highlighted for their high accuracy and simplified feature engineering, outperforming traditional models.

## III METHODOLOGY

### *A. Dataset Description –*

The UCF50 dataset, created by the University of Central Florida and released in 2011, is a widely-utilized video collection for human action recognition tasks. It comprises 6,618 videos across 50 diverse action categories, sourced primarily from YouTube, with most videos lasting from a few seconds to a couple of minutes and typically filmed at 25 frames per second. The action categories span a broad range, including sports, daily activities, interactions, exercise, and miscellaneous actions like applying makeup and yo-yoing. Videos feature significant intra-class variation, complex backgrounds, and camera movements, posing substantial challenges for recognition algorithms. With resolutions around 320x240 pixels and each video annotated with a single action label, UCF50 serves as a critical benchmark for developing and testing ML models in the action recognition domain. Researchers extensively use it to explore temporal and spatial features in video data and to evaluate the robustness of their models in real-world conditions. The dataset can be accessed and downloaded from the UCF50 homepage and other academic repositories. Figure 1 shown few sample images frames from the dataset.

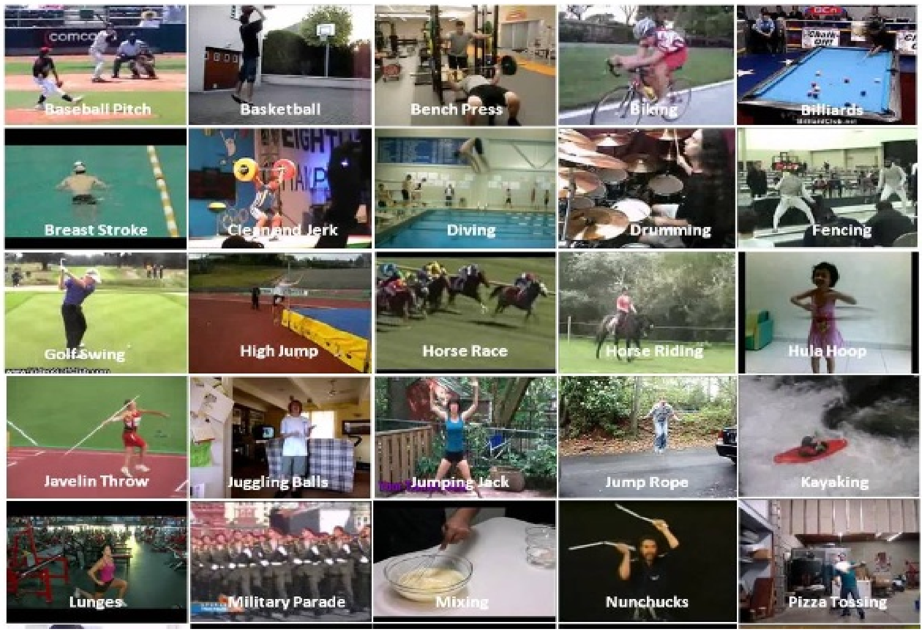

Figure 1. Sample image frames from the dataset UCF50

### *B. Proposed Methodology*

This paper proposes a hybrid approach for HAR utilizing both CNN and LSTM to capture spatial and temporal features from video data. Initially, video frames are loaded and preprocessed to ensure consistency in size and normalization of pixel values. A custom CNN model is developed from scratch to extract spatial features from these frames. This CNN model is then applied to each frame in the sequence using the TimeDistributed layer, allowing the extraction of features for each time step. To capture the temporal dependencies between frames, an LSTM layer is incorporated, followed by fully connected layers and a dropout layer to reduce overfitting. The model ends with a softmax activation function for classification.

The CNN extracts spatial features while the LSTM captures the temporal dynamics, enabling the model to effectively recognize activities from video data [19]. This combined CNN-LSTM model is trained on the UCF50 dataset, which provides a diverse range of action videos. The proposed methodology leverages the strengths of both CNN and LSTM, resulting in a robust and accurate HAR model that significantly improves performance on the UCF50 dataset. The experimental results demonstrate the model's capability to achieve new state-of-the-art accuracy, validating the effectiveness of the proposed approach.

Algorithm1 HAR Model

```
Step 1: Import necessary libraries
    - Import libraries for data manipulation (e.g., numpy, os)
    - Import libraries for video processing (e.g., cv2)
    - Import TensorFlow/Keras libraries for building models (e.g., Sequential, Conv2D, LSTM, Dense,
TimeDistributed, Adam)

Step 2: Load and preprocess video frames
    - Define a function to load video frames from a given path
    - Resize frames to a consistent height and width
    - Normalize pixel values
    - Ensure each video has a consistent number of frames by repeating the last frame if necessary

Step 3: Generate batches of data
```

```
    - Define a data generator function
    - Randomly select action classes and videos for each batch
    - Use the video loading function to preprocess the selected videos
    - Create batches of video data and corresponding one-hot encoded labels
    - Yield batches for training the model

Step 4: Define the custom CNN model
    - Initialize a sequential model
    - Add convolutional layers with activation functions and pooling layers
    - Flatten the output to prepare it for the LSTM layer

Step 5: Define the combined CNN-LSTM model
    - Initialize a sequential model
    - Use TimeDistributed to apply the CNN model to each frame in the sequence
    - Add an LSTM layer to capture temporal dependencies
    - Add fully connected layers and a dropout layer to reduce overfitting
    - Add an output layer with a softmax activation function for classification
    - Compile the model with a suitable loss function and optimizer

Step 6: Train the model
    - Initialize the data generator for training data
    - Define the number of epochs and steps per epoch for training
    - Use the fit method to train the CNN-LSTM model with the data generator
```

Algorithm 1 outlines the step-by-step procedure for constructing our model. Initially, necessary libraries for data manipulation, video processing, and model building are imported. Video frames are then loaded and preprocessed to ensure a consistent size and normalization of pixel values. A custom CNN model is developed from scratch to extract spatial features from these frames. This CNN model is applied to each frame in the sequence using a TimeDistributed layer, allowing feature extraction at each time step. To capture the temporal dependencies between frames, an LSTM layer is incorporated, followed by fully connected layers and a dropout layer to mitigate overfitting. The model concludes with a softmax activation function for classification.

The process begins with importing the necessary libraries and loading the video frames. The frames are resized and normalized, ensuring a uniform input for the model. A custom CNN is constructed to extract spatial features, which is then applied sequentially across all frames using TimeDistributed. An LSTM layer captures temporal relationships, and fully connected layers with dropout are added to enhance the model's robustness. Finally, the model is compiled and trained on the UCF50 dataset, demonstrating significant improvements in HAR accuracy.

## III. EXPERIMENTAL RESULTS

We have split the UCF50 dataset into an 80-20 ratio for training and testing. First, we ensured the dataset was organized with video files in separate folders for each activity class. We then created a list of all video file paths and their corresponding labels. Using the train_test_split function from the sklearn.model_selection module, we performed the split by specifying test_size=0.2 to allocate 20% of the data for testing and 80% for training. Additionally, we used the stratify parameter to ensure the split maintained the same proportion of each class in both sets, which is crucial for maintaining class balance in the training and testing subsets.

For training the model on the UCF50 dataset, several hyperparameters are set to control the learning process. These include the batch size, which decides how many samples are processed during each training cycle, typically set to 32. The learning rate governs the step size during optimization, often initialized at 0.001 for Adam optimizer. The number of epochs, typically set to 150, indicates how many times the entire dataset is passed through the model during training. Architectural hyperparameters such as the number of filters in convolutional layers (32), kernel size (3x3), and pool size (2x2) regulate the CNN's feature extraction capabilities. Additionally, the number of LSTM units (64) and dropout rate (0.5) control the complexity and regularization of the LSTM layer. These hyperparameters are crucial for fine-tuning the model's performance and generalization ability.

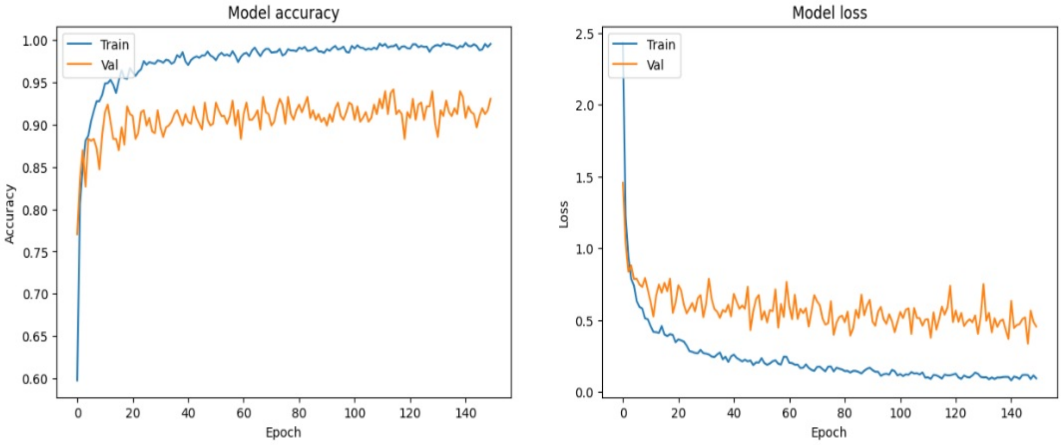

Figure 2. Model Accuracy and Model Loss

Figure 2 displays two graphs showing model accuracy and loss, both reflecting a consistent trend. As the number of epochs grows, the model's accuracy improves, while its loss decreases, indicating effective learning and performance enhancement on the training data. The model is trained over 150 epochs, achieving its highest validation accuracy of 94.14% at the 115th epoch. Towards the end of training, a slight gap between training and validation accuracy emerges, suggesting minor overfitting. This implies that the model might be overly customized to the training data, potentially affecting its generalization to new, unseen data.

Table 1. Performance comparison will all previous models

| Network Model | Accuracy(%) | Precision | Recall | F1 score |
|---|---|---|---|---|
| Lagrangian particle trajectories[21] | 80.97 | 0.83 | 0.80 | 0.81 |
| Relative motion descriptor (RMD) + Modes[22] | 82.03 | 0.82 | 0.79 | 0.80 |
| Dense cuboids + HOG + HOF + MBH +BoF[23] | 80.29 | 0.82 | 0.77 | 0.79 |
| Dense trajectory + HOG +HOF + MBH + BoF[23] | 84.57 | 0.86 | 0.84 | 0.84 |
| CNNs+LDS[20] | 82.76 | 0.83 | 0.81 | 0.81 |
| **CNN + LSTM (Proposed Method)** | **94.14** | **0.95** | **0.94** | **0.94** |

The proposed CNN + LSTM model stands out as the most effective approach among all previous models [20, 21, 22, 23] for HAR, shown in Table 1. While prior models achieved respectable performance, ranging from 80.20% to 84.50% validation accuracy, and F1 scores between 0.79 to 0.84, the CNN + LSTM model significantly surpassed them, achieving a remarkable validation accuracy of 94.14% and an F1 score of 0.94. Moreover, its precision and recall scores of 0.95 and 0.94, respectively, demonstrate its exceptional ability to accurately classify activities while minimizing both false positives and false negatives. This indicates that the proposed model not only achieves higher accuracy but also maintains a superior balance between precision and recall, making it a standout choice for HAR tasks.

The analysis revealed that activities such as PlayingGuitar, Fencing, MilitaryParade, and Drumming exhibited higher accuracy, possibly due to distinct characteristics that set them apart from other categories. However, the model exhibited lower performance for activities like GolfSwing, JavelinThrow, HighJump, and Nun Chucks, suggesting that the proposed technique may struggle when confronted with unclear backgrounds and high-speed movements.

## IV.CONCLUSION

This study introduces a novel DL approach tailored for HAR on real-world datasets. By leveraging the capabilities of both CNN and LSTM architectures, our method effectively extracts both temporal and spatial features from the data. Our experiments conducted on the UCF-50 dataset demonstrate higher quality output in comparison to most existing descriptors, achieving an accuracy of 94.14%, along with precision, recall, and F1 score of 0.95, 0.94, and 0.94, respectively.

## REFERENCES

[1] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems 25 (2012).*

[2] Wu, Zhirong, et al. "A gpu implementation of googlenet." *Tech. Rep., Technical report (2014): 6.*

[3] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556 (2014).*

[4] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.*

[5] Farabet, Clement, et al. "Learning hierarchical features for scene labeling." *IEEE transactions on pattern analysis and machine intelligence 35.8 (2012): 1915-1929.*

[6] Simonyan, Karen, and Andrew Zisserman. "Two-stream convolutional networks for action recognition in videos." *Advances in neural info rmation processing systems 27 (2014).*

[7] Reddy, Kishore K., and Mubarak Shah. "Recognizing 50 human action categories of web videos." *Machine vision and applications 24.5 (2013): 971-981.*

[8] Ramasamy Ramamurthy, Sreenivasan, and Nirmalya Roy. "Recent trends in machine learning for human activity recognition—A survey." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8.4 (2018): e1254.*

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

[10] Ji, Shuiwang, et al. "3D convolutional neural networks for human action recognition." *IEEE transactions on pattern analysis and machine intelligence 35.1 (2012): 221-231.*

[11] Dollár, Piotr, et al. "Behavior recognition via sparse spatio-temporal features." *2005 IEEE international workshop on visual surveillance and performance evaluation of tracking and surveillance. IEEE, 2005.*

[12] Karpathy, Andrej, et al. "Large-scale video classification with convolutional neural networks." *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2014.*

[13] Yue-Hei Ng, Joe, et al. "Beyond short snippets: Deep networks for video classification." *Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.*

[14] Donahue, Jeffrey, et al. "Long-term recurrent convolutional networks for visual recognition and description." *Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.*

[15] Shi, Zhenguo, et al. "Human activity recognition using deep learning networks with enhanced channel state information." *2018 IEEE Globecom Workshops (GC Wkshps). IEEE, 2018.*

[16] Mutegeki, Ronald, and Dong Seog Han. "A CNN-LSTM approach to human activity recognition." *2020 international conference on artificial intelligence in information and communication (ICAIIC). IEEE, 2020.*

[17] Kim, Kilho, et al. "A deep learning based approach to recognizing accompanying status of smartphone users using multimodal data." *Journal of Intelligence and Information Systems 25.1 (2019): 163-177.*

[18] Donahue, Jeffrey, et al. "Long-term recurrent convolutional networks for visual recognition and description." *Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.*

[19] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation 9.8 (1997): 1735-1780.*

[20] Zhang, Lei, et al. "Realistic human action recognition: When cnns meet lds." *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017.*

[21] Todorovic, Sinisa. "Human activities as stochastic kronecker graphs." *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II 12. Springer Berlin Heidelberg, 2012.*

[22] Oshin, Olusegun, Andrew Gilbert, and Richard Bowden. "Capturing relative motion and finding modes for action recognition in the wild." *Computer Vision and Image Understanding 125 (2014): 155-171.*

[23] Wang, Heng, et al. "Dense trajectories and motion boundary descriptors for action recognition." *International journal of computer vision 103 (2013): 60-79*