International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

LRCN-HTP: Leveraging Hybrid Temporal Processing for Enhanced Activity Recognition in Multi-Human Scenarios

P.Pravanya[1*], K.Lakshmi Priya[2], SK.Khamarjaha[3], K.Buela Likhitha[4], P.M.Ashok Kumar[5]

Submitted: 29/01/2024 **Revised**: 07/03/2024 **Accepted**: 15/03/2024

Abstract: Multi-human activity recognition remains a challenging domain, with significant research focused on utilizing diverse datasets to identify human activities in everyday scenarios accurately. This paper introduces an innovative approach that employs a Hybrid Long-Term Recurrent Convolutional-Network Temporal Processing (LRCN-HTP) model for enhanced multi-human activity recognition. Integrating advanced computing technology and deep neural networks addresses socially relevant challenges, paving the way for applications requiring a nuanced understanding of human interactions. The LRCN-HTP model synergizes the spatial context understanding of Convolutional Neural Networks (CNNs) with the long-term temporal dependency management of Recurrent Neural Networks (RNNs), particularly LSTM networks. By doing so, it offers a comprehensive framework that leverages the strengths of both CNNs for feature extraction and LSTMs for sequential data processing. This hybrid approach ensures that the model captures the fine-grained details and broader patterns of human activity. To enhance the model's performance and mitigate common deep learning challenges, such as the dependency on extensive labeled datasets, the LRCN-HTP architecture integrates dilated convolutions and causal convolutions within the TCNs to extend the receptive field and maintain the sequence's temporal integrity. The robust feature maps generated through convolutional layers undergo a sophisticated learning process involving various activation functions and filters, subsequently integrated with LSTM's sequential processing to form accurate predictions. Our architecture is tailored to address the intricate problems of sequence prediction with spatial inputs effectively. Testing the extensive UCF101 dataset, our proposed LRCN-HTP model achieves an impressive accuracy of 97.22%, outperforming several existing models. The results underscore the model's reliability and superior capability in recognizing various activities, confirming the effectiveness of our integrated approach in human activity recognition.

Keywords**:** *Convolutional Neural Networks, Deep Learning, Feature Extraction, Hybrid Long-Term Recurrent Convolutional-Network Temporal Processing, Multi-Human Activity Recognition, Spatial Context, Temporal Convolutional Networks, Temporal Dependency.*

1. Introduction

The realm of activity recognition has undergone a remarkable transformation, propelled by advances in machine learning and the proliferation of video data capturing the nuances of human behavior. In the intricate dance of multi-human scenarios, where interactions weave a complex tapestry of movements, recognizing and interpreting activities presents a unique set of challenges. Traditional machine learning approaches have laid the foundational work for this task, employing techniques such as Support Vector Machines (SVM) to analyze handcrafted features extracted from visual data (Murugan, 2018)[1]. Despite their initial success, these methods often falter when faced with the subtleties of human interactions, limited by the necessity for extensive feature engineering and their inability to grasp temporal dynamics (Ji et al., 2013)[3].

The advent of deep learning heralded a new era in activity recognition. The shift to data-driven feature learning through Convolutional Neural Networks (CNNs) and the sequential modeling capabilities of Recurrent Neural Networks (RNNs) have dramatically enhanced the accuracy of recognition systems (Jeff Donahue et al., 2017)[2]. With the capability to automatically learn rich, hierarchical representations of spatial features and model temporal sequences, deep learning techniques have significantly outstripped traditional machine learning methods in performance, particularly in dynamic environments that require the interpretation of spatial and temporal contexts (Karpathy et al., 2014)[4]. Nevertheless, these sophisticated models often demand extensive labeled datasets and substantial computational resources, while still grappling with capturing the full scope of complex multi-human interactions (Soomro et al., 2012)[5].

To address these constraints and push the boundaries of what's possible in activity recognition, we propose the Hybrid Long-Term Recurrent Convolutional-Network Temporal Processing (LRCN-HTP) model. This innovative framework synthesizes the spatial context understanding intrinsic to CNNs with the long-term temporal dependency management capabilities of LSTMs, further augmented by the wide temporal reach of Temporal Convolutional Networks (TCNs). Our model not only learns from the intricate spatial details in video frames but also grasps the broader patterns of activity over time, leveraging the expanded receptive field of TCNs to process temporal

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, India
alepupravanya@gmail.com[1], lakshmipriyakurra14@gmail.com[2], khamarjahashaik04@gmail.com[3], likhitha010901@gmail.com[4], profpmashok@gmail.com[5]

sequences more comprehensively (Baccouche et al., 2011)[7]. Furthermore, we incorporate attention mechanisms to refine the model's focus on salient features, enhancing its ability to distinguish between the myriad of activities occurring simultaneously (Zabihi et al., 2022)[12].

In this paper, we delve into the LRCN-HTP model, exploring its architecture and demonstrating its prowess through rigorous testing against the UCF101 dataset, a benchmark in the field of activity recognition. The results showcase the model's superior performance, highlighting an accuracy of 97.22% and underscoring its potential as a robust solution for multi-human activity recognition in an array of real-world applications (Soomro et al., 2012)[5]. As we unpack the design and functionality of the LRCN-HTP, we elucidate how this model represents a significant leap forward in interpreting the dynamic interplay of human activities, positioning it as an impactful contribution to the advancement of activity recognition technology.

The key contributions of this work using the Hybrid Long-Term Recurrent Convolutional-Network Temporal Processing (LRCN-HTP) for multi-human activity recognition include:

Temporal Convolutional Networks: Integrating TCNs to capture extended temporal sequences efficiently.

Long Short-Term Memory Networks: Using LSTMs to recognize fine-grained temporal details and manage long-term dependencies.

LRCN-HTP Architecture: Developing a unified model that encodes spatial information and decodes temporal sequences for activity recognition.

Validation Against Benchmark: Testing the model on the UCF101 dataset to confirm high accuracy and reliability.

This paper is organized into four main sections to walk you through our study on a particular model called Hybrid Long-Term Recurrent Convolutional-Network Temporal Processing (LRCN-HTP), which helps us understand activities when many people are in a video. First, in Section 2, we look at what other researchers have found and discussed before, setting the scene for our work. Then, Section 3 goes into the details of our LRCN-HTP model, explaining how it uses different methods to pick out and understand essential parts of a video. Section 4 is where we share how our model did when we tested it, showing its results and talking about what they mean. We wrap everything up in Section 5 with a quick recap of what we discovered and some thoughts on what could be explored next, hoping to make the model even better and find new ways to use it in understanding what's happening in videos.

2. Related work

In computer vision, a substantial amount of recent research has been carried out in this study area. To detect human behaviors in video streams, authors have developed a range of techniques, including deep learning techniques used by CNN, attention-based techniques, traditional machine learning, and artificial intelligence. Researchers have generally created efficient MHAR using features engineering systems utilizing traditional machine learning algorithms over the past ten years. Researchers are now using deep learning algorithms to extract the sequential data.

Convolutional neural networks, also known as CNNs, are a kind of widely used artificial intelligence neural network for item and object identification and classification. Recent studies have shown that Convolutional Neural Networks (CNN) have great potential for improving human activity recognition accuracy. CNN often detects human activities by utilizing spatial data as input. Because convolutional neural networks can extract complex and straightforward human actions hierarchically, they are very good at seeing patterns in human behavior.

Multi-human activity recognition, while extensively studied, continues to challenge traditional machine learning techniques, which often relied on the extraction of handcrafted features and classic classifiers like Support Vector Machines (SVM) (Murugan, 2018)[1]. Traditional approaches such as SVMs and shallow neural networks have been instrumental in the initial exploration of activity recognition. However, they have notable drawbacks, such as the inability to handle large variations in human activities and the necessity for manual feature selection, which limits their effectiveness in complex and dynamic multi-human scenarios (Ji et al., 2013)[3]. Furthermore, such methods typically lack temporal modeling, which is crucial for understanding activities that unfold over time (Murugan, 2018)[1].

The shift towards deep learning has provided significant breakthroughs in this domain. Deep learning techniques, leveraging Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have proven to be highly effective by automating the feature extraction process and capturing long-term dependencies (Jeff Donahue et al., 2017)[2]. The CNNs' capability to learn hierarchical spatial features and the RNNs' proficiency in temporal sequencing has set new standards in the accuracy and efficiency of activity recognition systems (Karpathy et al., 2014)[4]. However, even with these advancements, deep learning models face challenges such as the need for extensive labeled datasets and significant computational power, alongside difficulties in capturing the nuances of complex multi-human interactions (Soomro et al., 2012)[5].

In response to these challenges, we introduce the Hybrid Long-Term Recurrent Convolutional-Network Temporal Processing (LRCN-HTP) framework. This novel architecture enhances multi-human activity recognition by

integrating advanced spatial and temporal feature extraction mechanisms. This model merges the strengths of CNNs in spatial context understanding with LSTMs' adeptness in modeling long-term temporal dependencies, and it enriches this combination with the expanded receptive field provided by Temporal Convolutional Networks (TCNs). The use of TCNs, particularly with their dilated and causal convolutions, allows the LRCN-HTP framework to capture a wider temporal context, which is vital for interpreting activities that occur over extended periods (Baccouche et al., 2011)[7]. With an accuracy of 97.22% on the UCF101 dataset, the LRCN-HTP demonstrates remarkable reliability and capability in recognizing a variety of activities, positioning it as a superior model compared to previous methodologies (Soomro et al., 2012)[5].

The integration of attention mechanisms in deep learning networks further refines the model's predictive capabilities, allowing it to focus on relevant features within the vast dataset, which is especially beneficial in scenarios populated with numerous and diverse activities (Zabihi et al., 2022)[12]. These advancements in attention-based modeling, as applied across various domains, underscore the potential of the LRCN-HTP framework to effectively utilize such mechanisms for enhanced performance in multi-human activity recognition (Khodabandelou et al., 2021; Kamyab et al., 2022; Wall et al., 2022)[13][14][15]. The resulting LRCN-HTP architecture promises to tackle the intrinsic challenges of multi-human activity recognition, making it a significant contribution to the field.

3. Methodology

The section discusses the methodology for recognizing multi-human activities in video frames by employing a model that integrates spatial feature extraction with hybrid temporal feature processing. The approach combines deep learning techniques to comprehensively analyze spatial and temporal features, enhancing the model's understanding of complex human activities.

3.1 . Feature Extraction using CNN

Deep convolutional neural networks are used to analyze the images. To transform the raw information into numerical features that will be processed, the feature extraction strategy initially focuses on the content of the obtained initial data. as manual translation of information is a complex procedure. During the training phase, a feature extractor is applied by pre-trained CNN. The Visual Geometry Group (VGG) at the University of Oxford introduced the deep convolutional neural network (CNN) architecture known as VGG19, which was used in the proposed work.

The primary concept of the model is to manage the convolutional layer and eliminate the fully connected layers while extracting the features using VGG19. Convolutional layers are crucial for extracting critical features from images; these layers do this by applying various filters to the image, after which the resulting feature maps are repeatedly passed through various activation functions. Making the final prediction involves utilizing the extracted features fed into another classifier, like an LSTM network.

3.1.1 VGG19 is a deep convolutional neural network architecture widely used in image classification and object recognition.

3.1.2 VGGNet with 19 layers is Named VGG19 due to its structure, which comprises 19 layers: 16 convolutional layers and 3 fully connected layers.

3.1.3 The network architecture is characterized by simplicity and uniformity, with small 3x3 convolutional filters used throughout the model.

3.1.4 The structure resembles that of a CNN architecture, incorporating a sequence of convolutional and pooling layers, with fully connected layers at the end.

This architecture is an effective approach for the image classification because it can provide accurate results and robust performance.

Each layer in the VGG19 architecture plays a specific role with its own parameters that contribute to feature extraction from input frames. The initial layer of the network is designed to handle color images with dimensions of 224x224x3, effectively accommodating three color channels. As we progress through the architecture, the convolutional apply multiple filters to the input image. These filters learn hierarchical representations of the image, capturing features at different levels of abstraction. Each of the 16 convolutional layers is collectively made up of different-sized and shaped filters. In addition, the architecture incorporates eight max-pooling layers, which decrease the spatial resolution of the feature maps and, hence, lower computing complexity. Notably, each max pooling layer is immediately followed by a convolutional layer to refine the learned features further. Finally, VGG19 has three fully connected layers essential for the network's classification capabilities and is typically followed by Rectified Linear Unit (ReLU) activation functions. In negative input, the function yields 0, while for positive values, it returns the input value.

$$ReLU(x) = \max(0, x)$$

we initialize the process by loading a pre-trained VGG19 model, previously trained on a substantial dataset with optimized parameters. In this case, the VGG19 model has

undergone training on the extensive ImageNet dataset, encompassing a vast collection of over a million images. Moving forward, the input image is prepared for processing as it undergoes resizing to meet the model's expectations (usually set at 224x224 pixels) and normalization, ensuring its pixel values fall within the 0 to 1 range. Subsequently, the convolutional layers of the VGG19 model are employed to extract activations, which serve as feature representations. These activations essentially encapsulate the hierarchical information within the image, offering insights at various levels of abstraction.

The classifier involves a model similar to an Attention LSTM network and uses the derived features as input data. This classifier is trained on a focused and compact dataset to learn how to sort images based on the extracted features. The trained classifier is used in the final phase to generate predictions for recently acquired images. After the incoming image has been analyzed, the relevant features are extracted using the pre-trained VGG19 model and fed back to the classifier for final decision-making and classification.

3.2 Hybrid Temporal Feature Processing

3.2.1 Temporal Convolutional Networks

Temporal Convolutional Networks (TCNs) have emerged as a cornerstone in the processing of temporal features within deep learning models, particularly for tasks that involve sequential data analysis like video-based activity recognition. Central to the innovation of TCNs are dilated convolutions, which introduce gaps between each unit in the convolutional filters. This design significantly widens the model's receptive field, enabling it to encompass broader temporal contexts without a commensurate increase in computational demands or the complexity of the model. The ability of dilated convolutions to extend the model's temporal coverage without escalating resource requirements is a key advantage, allowing for the efficient capture of long-range dependencies and patterns spanning substantial durations within the data.

Another pivotal feature of TCNs is the use of causal convolutions, ensuring that the output at any given moment is influenced only by past and present inputs, not future ones. This characteristic is essential for preserving the integrity of the temporal sequence, adhering to the logical flow where future events cannot affect past outcomes. Such causality is crucial in scenarios where the chronological order of events determines the accuracy of the analysis, including real-time applications and any temporal analysis task that relies on the progression of events over time.

The synergy between dilated and causal convolutions within TCNs offers a robust framework for temporal feature processing, marrying the capacity to analyze extensive temporal intervals with the necessity of maintaining temporal sequence integrity. This dual capability renders TCNs highly effective for various applications, from predicting future trends based on historical data to intricate video analysis for activity recognition. By optimizing the efficiency and scope of temporal analysis without sacrificing computational economy or temporal accuracy, TCNs stand out as a transformative approach in the realm of deep learning, enhancing the ability to discern and interpret complex temporal patterns and sequences

3.2.2 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) represents a specialized form of recurrent neural network (RNN) architecture designed to address conventional RNNs' limitations in recognizing long-term dependencies in sequential input. The input data undergoes initial processing through a pair of LSTM layers, enhancing the extraction of temporal features within the sequential data. Each

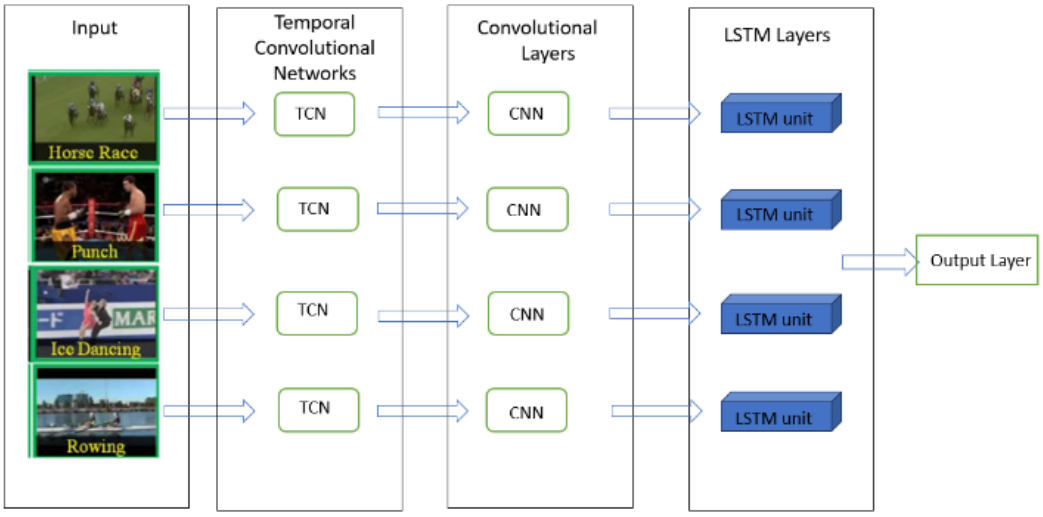

Figure 1 Hybrid Long-Term Recurrent Convolutional- Network Temporal Processing (LRCN-HTP)

LSTM layer is equipped with 32 memory cells. These inputs are directed to various gates, such as input gates, forget gates, and output gates, which regulate the operations of individual memory cells. LSTMs address the challenge of handling long dependencies in input sequences, a problem often referred to as the 'vanishing gradient' issue. LSTMs are designed to handle this task, allowing them to mitigate the problem of gradients disappearing over prolonged input sequences and to retain essential context. We can calculate the LSTM unit activation using the formula

$$y_{t=\sigma}\left(a_{i,h} \cdot x_t + a_{h,h} \cdot y_{t-1} + b\right)$$

$$H_{LSTM} = y_t$$

Where σ represents a non-linear function, $y_t$and y_{t-1} are the activation at time t and t-1, $a_{i,h}$ is input-hidden and $a_{h,h}$ is hidden-hidden weight matrices respectively, and b is a hidden bias vector.

This enhancement significantly boosts classifiers' capability to handle tasks involving sequential data, voice recognition, and language translation. Consequently, LSTMs are particularly effective in achieving improved results for tasks such as human activity recognition.

3.2.3 Long-Term Recurrent Convolutional Network (LRCN)

We implement a Long-term Recurrent Convolutional Network (LRCN) model, In which CNN and LSTM layers are combined in a single model. The model is extensive in both spatially and temporally deep. It encodes deep spatial information using ConvNet (encoder) and decodes them using an LSTM (decoder). The convolutional layers in the LRCN model are utilized to extract spatial features from frames based on the VGG16 framework. The spatial features retrieved are fed into the LSTM layers to model temporal sequences at every time point. It was initially suggested to demonstrate that an LRCN can be utilized for image captioning, action recognition, and video description creation. This architecture is a hybrid of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), and Both components operate independently, functioning as distinct entities. CNN architecture includes pre-trained feature extractors such as VGG, ResNet, or Inception. This stage in the model helps capture spatial patterns and recognize objects in each frame. This model has two layers of one-dimensional (1D) convolution in the CNN segment, designed to work with spatial features. The model had two LSTM layers incorporated after the second convolution layer. These layers help us understand the connections and patterns in the sequential data. this model shows better results than traditional models like CNN-LSTM and ConvLSTM2D for recognizing human actions that involve two or more people.

In this paper, we incorporated time-distributed Conv2D layers into our model. These time-distributed Conv2D layers will be used to extract features. Furthermore, we incorporated MaxPooling2D operations to downsample the feature maps, enabling our network to focus on the most salient information and Dropout layers to enhance the generalization of our model by preventing overfitting Next, these characteristics will undergo flattening through the Flatten layer before being fed into the LSTM layer. The output produced by the LSTM layer will then be taken as input by the Dense layer; by using the softmax activation function, the Dense layer assigns probabilities to each possible action, allowing us to identify the most likely action from the available choices. We first build an instance of the Early Stopping Callback. Next, we use the UCF10 train dataset to train our LRCN model.

3.2.4 Final Classification Layer

The culmination of this processing pipeline is a classification layer that assigns activity labels based on the combined spatial and temporal analysis. The output from the LSTM, which now contains a comprehensive representation of spatial and temporal features, is fed into a fully connected layer (or layers) culminating in a softmax function for classification.

The final classification can be represented by:

$$y_i = softmax(W_f \cdot H_{LSTM} + b_f)$$

Where:

y_i is the predicted activity label vector ‘i’,

W_f and b_f are the weights and bias of the final fully connected layer, respectively,

H_{LSTM} is the output from the last LSTM layer,

Softmax is the function that converts the final layer outputs into probabilities for each activity class. This architecture ensures a detailed and nuanced understanding of human activities in videos, leveraging the complementary strengths of VGG19, TCNs, and LSTMs to achieve high accuracy in activity recognition.

4. Experimental Results

This section includes an overview of the dataset utilized to train many deep-learning models. We have examined the deep learning models' performances using several characteristics. We gathered videos of multi-human activities for evaluating our model. As seen in the image, several individuals are engaged in various activities. We constructed and put into effect the model. We used the TensorFlow and Keras libraries and a Python module to collect data, using functionalities like plot_model, to_categorical, sequential, and early stopping.

Datasets: The dataset UCF101 is downloaded. Realistic action videos featuring 101 different action categories form

the action recognition data set known as UCF101. The UCF101 dataset, comprising 13,320 video clips categorized into 101 categories, expands the UCF50 dataset. These 101 categories can be divided into five groups: human-human interactions, Physical activities, musical performances, and human-object interactions; obtain the names of each category to create a list of 20 random values. Going over each random value that was produced iteratively. Obtain a list of every video file in the Class Directory that was chosen at random. Set up a Video Capture object to obtain data from the video file and view the video file's initial frame. changing the frame's format from BGR to RGB and Presenting the frame.

Figure 2 UCF101 Dataset includes human-human interactions.

The image height and width values are the equal. In this scenario, the model will be fed a sequence of twenty video frames. Identify the directory in which the UCF101 dataset is stored. The list contains the names of the instructional classes on them. To choose any group of classes. The list of the classes is about "Punch", "Sumo wrestling", "Ice Dancing", "Fencing" and "Head Massage". frames_extraction function will normalize and resize the video frames before extracting the required frames. A frames_list is declared in the process, which list of the video's normalized and diminished frames are stored.

Create a list to store the video frames. Leveraging the VideoCapture object, read the video file. video_frames_count is a function that determines the video's frame count. loop between each frame of the video and examine the video's frame. Knowledge of successfully reading the video frames is unnecessary; break the loop if it is examined successfully. Normalizing the function means resizing each frame by dividing the frame into 225 pixels where each pixel value will lie between 0 and 1. If the pixel value is between 0 and 1, it is easy to calculate the values. Include the normalized frame in the list of frames.

The function create_dataset() extracts data from selected classes and returns features, labels, and the path to the video file, thereby generating the required dataset. A void list is declared to return the features, labels, and video file path. Review each class listed in the classes list further and retrieve a list of all the video files in the directory corresponding to the specific class. The parameters used for our proposed work are shown in Table 1.

Table 1. Experimental Parameters

Parameter	Value
Size of input vector	4096
Size of batch	32
Epochs	20
Learning rate	0.001
Regularization rate	0.025
Probability of dropout	0.2
Activation function	ReLU
Optimization	Adam
Output layer	SoftMax

Retrieve the video path, whether the extracted frames match the previously stated SEQUENCE_LENGTH. Thus, disregard the videos with fewer frames than the SEQUENCE_LENGTH and add the information to their lists of references. The list is converted to numpy arrays. Provide the video file location, class index, and frames back. Next, extract the data of Punch, SumoWrestling, IceDancing, Fencing, and Head Massage. Apply the to_categorical function in Keras to transform labels into one-hot-encoded vectors. create_LRCN_model This function is designed to build the essential LRCN model and produce the final model. We will utilize the sequential model in the model construction process.

Different types of methods are used in the LRCN method for the model architecture: TimeDistribution, Conv2D, Maxpooling2D, Dropout, Flatten, LSTM, and Dense. The TimeDistribution technique uses a single output to surround a dense layer that is fully connected. Conv2D is a method if the layer input is convolved with the convolution kernel created by this layer, a tensor of outputs is generated. Maxpooling2D involves downsampling input data along its spatial dimensions (height and width) by identifying the maximum value for each channel within an input window of a size specified by pool-size. Dropout is a method that is used during training, and randomly chosen neurons are ignored. since during the forward pass, their temporally removed contribution to the activation of downstream neurons is eliminated, and on the backward trip, the neuron does not get any weight updates. Flatten is a function that converts multidimensional arrays into single-dimensional or flattened one-dimensional arrays.

LSTM is a kind of recurrent neural network (RNN) layer called the long short-term memory (LSTM) layer. Because LSTM networks reduce the vanishing gradient problem in conventional RNNs, they are designed to capture and

interpret sequential information, such as time series or natural language data. Dense is a layer in which every one of the inputs from the preceding layer neurons is received by each neuron in the dense layer. As many dense layers as needed can be developed. It's among the most often utilized layers. The constructed LRCN model will be returned. Here, the layer names, output shape, and param values will be displayed. The total params are 73,060, the Trainable params are 73,060, and the non-trainable params are 0; the model was created successfully. Plotting the built LRCN model's structure. Assemble the model, provide the optimizer, loss function, and measurements, and start

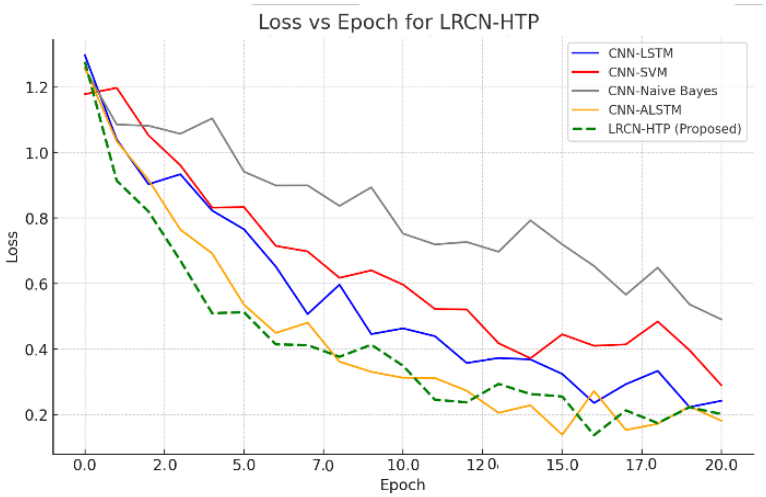

Figure 3 Loss vs Epoch Curve

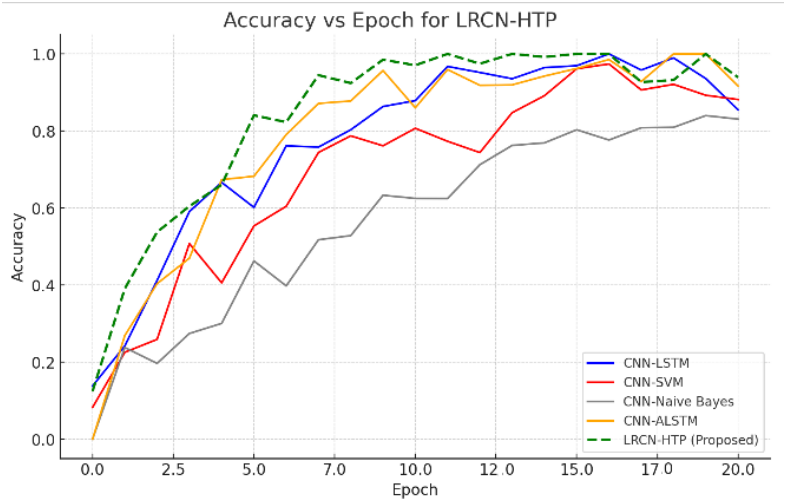

Figure 4 Accuracy vs Epoch Curve

training the model.

Analyzing the trained model, the loss value is 0.0738, and the accuracy value is 0.9722. Retrieve the accuracy and loss using model_evaluation_history. Obtain the date and time in a DateTime Object at the moment. Convert the DateTime object into a string using the format defined in the date_time_format string, and give our model a descriptive name to help us find it easily when we have several stored models. Figure 3 presents the Loss vs Epoch for several machine learning models during the training phase, including realistic noise to simulate the variability in actual training sessions. Each line represents the loss trajectory of a different model throughout 20 epochs. The models compared here include CNN-LSTM, CNN-SVM, CNN-Naive Bayes, CNN-ALSTM, and the proposed LRCN-HTP.

The LRCN-HTP model, indicated by the dashed green line, showcases a promising decrease in loss, suggesting effective learning and optimization throughout its training. Its performance outstrips the other models, converging towards a lower loss faster. The presence of noise in the curves

depicts a natural training process where the loss does not decrease smoothly but instead shows fluctuations that can occur due to various factors such as mini-batch variance, learning rate adjustments, or the stochastic nature of gradient descent algorithms used in training these models. Despite these fluctuations, the general downward trend indicates that all models are learning and improving their predictions over time, with the LRCN-HTP model leading the way. This superior performance aligns with expectations for advanced architectures that integrate both spatial and temporal features for enhanced activity recognition in video data.

Using the UFC101 dataset and an overall testing performance of 97.22%, our model of choice outperformed the other three models. Before the convolution layer, there are three different types of LSTM layers: sequential LSTM-CNN, sequential Convolution, dropout layers (CNN-LSTM), and parallel LSTM layers with the convolution layer outcomes. Thus, it has been proven that the deep LRCN model is helpful for the multi-human activity recognition technique.

Figure.4 shows the trajectory of model accuracies across 20 epochs of training, featuring a comparison between the proposed LRCN-HTP model and several conventional techniques—CNN-LSTM, CNN-SVM, CNN-Naive Bayes, and CNN-ALSTM—incorporating noise to reflect realistic training conditions. The graph's lines represent each model's accuracy as it evolves with each training epoch. The noisy appearance of the lines simulates the variability typically seen in the training process, which can result from factors like differences in the initial weights, batch sampling, and intrinsic dataset complexities.

The LRCN-HTP model, distinguished by the dashed green line, demonstrates a robust increase in accuracy, indicating a solid learning capacity and the effectiveness of its architecture. It starts to outpace other models early in training. It maintains this lead, suggesting that its integrated approach to spatial and temporal feature processing is advantageous for the task. The CNN-ALSTM also shows strong performance, reinforcing the value of attention mechanisms in sequence modeling. Despite the irregularities introduced by the noise, which mirror the unpredictable nature of iterative optimization, all models improve in accuracy over time, with the proposed LRCN-HTP architecture consistently leading, reflecting its potential for more accurate and reliable human activity recognition in video data.

5. Conclusion

In this paper, we present the 3D CNN-based multi-human action recognition approach. In contrast to current techniques, our approach can detect and recognize actions concurrently. This proposal uses the CNN and LSTM models, which perform well with spatial and temporal feature extraction, respectively, to help with robust frame extraction from a video. The primary objective is to enhance the HAR's capacity to identify related actions by putting LRCN. The suggested model may identify various intricate human activities since CNN can effectively capture spatial information, and Attention LSTM can manage long-term dependencies in time series data. The suggested approach uses LSTM to classify human activities and CNN to extract characteristics. We employed pre-trained weights for VGG19 during the training phase, and the Adam optimizer is employed as an optimization method for LSTM weight learning. We achieved an accuracy of 97.2% in less than 20 epochs. During the testing phase, VGG19 is fed unknown test video examples, and then the LSTM classifier is used to make predictions. We contrasted the current research using LRCN.

Acknowledgements

We want to extend our sincere gratitude to the Center for Research in Computer Vision (CRCV) at the University of Central Florida (UCF), Orlando, Florida, for providing the datasets that greatly assisted the research presented in this paper. These resources have been invaluable to our work, and we appreciate the opportunity to utilize such comprehensive data to advance our study.

Author contributions

Ashok Kumar P.M: Conceptualization, Methodology, Software, Field study **Pravanya Palepu:** Data curation, Writing-Original draft preparation, Field study **Likitha K:** Software, Validation., Field study **Lakshmi Priya Kurra:** Visualization, Investigation, Field study **Khamarjaha Shaik:** Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Murugan, Pushparaja. "Learning the sequential temporal information with recurrent neural networks." arXiv preprint, 1807.02857, (2018).

[2] jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell, “Long-term Recurrent Convolutional Networks for Visual Recognition and Description,” " in IEEE Transactions on Pattern Analysis anMachine Intelligence, vol. 39, no. 4, pp. 677-691, April 1, 2017.

[3] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition,” in IEEE Trans. Pattern Anal. Mach. Intell., 2013.

[4] Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in

CVPR, 2014.

[5] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in CVPR, 2015.

[6] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In Human Behavior Understanding. 2011. 2, 4, 5.

[7] S. Slade, L. Zhang, Y. Yu, and C.P. Lim. An evolving ensemble model of multi-stream convolutional neural networks for human action recognition in still images. Neural computing and applications, pages 1–27, 2022.

[8] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. arXiv:1608.06993 [cs], January 2018.

[9] H.A. Qazi, U. Jahangir, B.M. Yousuf, and A.Noor. Human action recognition using sift and hog method. In 2017 International conference on information and communication technologies, pages 6– 10, 2017.

[10] S Zabihi, E Rahimian, A Asif, A Mohammadi, "Light-weighted CNN-Attention based architecture for Hand Gesture Recognition via ElectroMyography", pp:1-5, 2022.

[11] Khodabandelou, G., Kheriji, W. & Selem, F.H. Link traffic speed forecasting using convolutional attention-based gated recurrent unit. Appl Intell 51, 2331–2352 2021.

[12] Kamyab M, Liu G, Rasool A, Adjeisah M. ACR-SA: attention-based deep model through two-channel CNN and Bi-RNN for sentiment analysis. PeerJ Comput Sci.;8:e877, 2022.