IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Analysis on Deep Learning approaches in Video classification of Human Activity Recognition

[1]S A Amutha Jeevakumari, [2]Koushik Dey,

[1]Assistant Professor (Sr), [2]Master's Student,
[1]School of Computer Science and Engineering,
[1]Vellore Institute Technology, Chennai, India

Abstract : This paper investigates two cutting-edge Deep Learning approaches: Long-term Recurrent Convolutional Networks (LRCN) and Convolutional Long Short-term Memory (ConvLSTM) networks in the video classification of Human Activity Recognition (HAR). ConvLSTM adds convolutional components to recurrent networks so that the model may efficiently collect spatial-temporal data. LRCN is a sequential neural network that confidently exhibits proficiency in processing sequential and spatial data, thereby addressing the challenge of video action classification effectively and confidently. The LRCN architecture employed in this model comprises an encoder and decoder. The encoder is composed of time-distributed convolutional layers, succeeded by an LSTM layer, which results in a reduction of spatial dimensions. The decoder encompasses a dense layer, an LSTM layer incorporating dropout, and a subsequent dense layer, designed for classification. We implement a variety of experimental configurations, encompassing distinct frame counts per video and employing learning strategies to enhance overall performance. Both models underwent training and assessment utilizing a standard action recognition dataset, UCF50. While both models demonstrate noteworthy accuracy, the LRCN model surpasses the ConvLSTM model by achieving a remarkable 96.67% accuracy with the UCF50 dataset.

IndexTerms **- Convolutional Neural Networks, Long Short-Term Memory, Human activity recognition, Spatial and temporal analysis.**

INTRODUCTION

As technology rapidly progresses, a myriad of notable innovations arises, each with immense potential to enhance our everyday lives. Recently, we've been using smart techniques like object recognition, feature extraction, and motion analysis. These have helped us spot human activities with amazing accuracy. The benefits of Human Activity Recognition are wide and varied, touching many different fields. These areas cover healthcare, training for sports, fun and entertainment, robotics, and even things like management. As societal norms continue to evolve, it's becoming increasingly apparent that family dynamics are also experiencing a significant shift. More and more, caregivers are finding themselves looking after their kids and older family members at home, all while fulfilling their professional obligations. This situation raises pertinent concerns regarding their safety, welfare, and efficient management, especially in the absence of their caregivers. A technology full of promise is on the horizon, ready to tackle these issues. It offers remote monitoring and crucial understanding of the daily routines of vulnerable family members living in the home. The incorporation of human activity recognition technology holds significant potential within the healthcare industry, particularly in the realm of monitoring the daily routines of patients enduring chronic illnesses, such as Parkinson's disease. The process of monitoring enables healthcare practitioners to identify any changes in an individual's behavior or mobility, and subsequently make necessary adjustments to their treatment regimen. In the realm of sports, the utilization of HAR can serve as a means to evaluate the performance and techniques of athletes, thereby identifying specific areas for improvement and optimizing training regimens. HAR can facilitate the automation of household appliances and devices in the realm of smart homes by leveraging the occupants' activities. For instance, the system can activate the lighting fixtures upon detecting the entry of an individual into a room. Within the field of robotics, the integration of HAR has the potential to augment the capabilities of robots by facilitating a deeper understanding of human behavior and promoting more efficient interactions with individuals. The utilization of HTTP Archive HAR in the domain of security can facilitate the detection of dubious conduct or unauthorized access in locations that are deemed to be of elevated risk, such as airports, financial institutions, or military compounds. To summarize, HAR is not limited to one area - it has a wide variety of uses across many industries. Plus, it's a field that's quickly growing and improving all the time.

The identification of human actions continues to be a complex problem that has captivated researchers for over two decades. This complexity mainly arises from the difficulties in precisely discerning actions within video sequences. Humans are equipped with a wealth of contextual information and advanced sensory input, which allows them to proficiently recognize actions in real-time. However, machines do not share this capability.

For a machine to accurately classify an action, it must first identify which data points within an image correspond to the object of interest. Additionally, it must examine the differences between frames that represent motion, trajectory, and interactions among various objects. Furthermore, video analysis for action recognition necessitates more discriminative features that span spatial and temporal dimensions [21], [22], as compared to image-based recognition. These features are essential for capturing motion, human subjects, and background variations to effectively classify distinct action categories.

Therefore, besides acquiring spatial features, it is imperative to integrate an extra dimension in video action recognition tasks that effectively encodes the temporal dependencies. The successful acquisition of spatial-temporal cues is crucial for action classification. In recent years, numerous innovative methods have emerged for extracting spatial-temporal features, with deep learning techniques being the most prevalent among them. These methodologies execute feature learning on a layer-by-layer basis and possess the ability to identify spatial and temporal disparities among image sequences. Consequently, deep neural networks exhibit exceptional efficacy in tackling object identification, pose approximation [23], and human body partitioning associated with action recognition tasks.

The fast progress in deep learning algorithms has made analyzing videos much more complex, especially in the realm of action classification. This demands considerable computational power and sophisticated pre-processing techniques for extracting optical flow with confidence. Addressing large-scale action recognition in resource-limited settings poses considerable obstacles. To mitigate computational expenses, researchers have concentrated on reducing the number of frames [24], [25], yielding encouraging outcomes. Nonetheless, this study primarily aims to determine the ideal equilibrium between performance and the extraction of diverse frame quantities for action recognition. This is achieved by employing a range of hybrid deep-learning networks. The goal is to maximize the accuracy of action recognition while minimizing computational resources and maintaining efficiency.

In this paper, two advanced approaches based on deep learning are presented. Their purpose is to predict human actions within videos, which simplifies the process of understanding and mastering video classification using neural networks. The first approach makes use of a framework known as Convolutional Long Short-Term Memory (LSTM), while the second approach employs a structure called Long-term Recurrent Convolutional Networks (LRCN). The performance of these models is evaluated using the UCF50 dataset, which is a widely accepted benchmark in the field of action recognition.

The dataset is used to train and test the model, which is designed to predict behaviors in films that show a variety of different activities. In order to determine which model is better, it is intended to compare the outcomes of the two models by evaluating the testing accuracy of each. This document has the following structure: HAR literature is reviewed and the datasets utilized in Section 2; Section 3 explores the proposed approach and analyses the results; and Section 4 concludes with recommendations for further study.

RELATED WORK

Before the emergence of deep learning, action recognition was tackled using a variety of methods. These methods relied on feature descriptors such as Histogram of Oriented Gradients (HOG), Scale Invariant Feature Transformation (SIFT), Local Binary Patterns (LBP), and Histogram of Optical Flow (HOF). The advancement of the field was achieved through techniques that incorporated both spatial and temporal information, with particular emphasis on optical flow and trajectories. The Motion Boundary Histograms (MBH) and Dense Trajectories were instrumental in this regard. After that, classifiers used the outcomes of these techniques to make predictions for each image frame using techniques like Support Vector Machines (SVM). Due to deep learning's success in picture classification, many people are now using deep networks for video analysis, which entails dissecting films into a series of images. The section that follows will examine various deep neural networks that have recently been created with action classification in mind.

For instance, a study by [26] investigated various strategies for classifying actions in videos by integrating data from a series of frames. In order to accurately anticipate actions, early, late, and slow fusions were utilized in conjunction with Convolutional Neural Networks (CNNs) to extract temporal characteristics from individual frames. Multi-resolution networks that emphasized the computing constraints necessary to construct such fusion networks were also put forth.

Expanding the convolutional operations or augmenting the network's depth may result in a notably sluggish processing time since CNNs are demanding in terms of resources, requiring a considerable amount of both time and computing power to effectively train models. When examining high frame rate videos, this is not practical. There are many ways to lower the computational cost. The performance of the network is severely compromised by reducing the number of convolutional layers, which is one possible solution. The researchers chose to address this problem by lowering the picture resolution of the input stream rather than reducing the number of CNN layers in the study. One network was used to focus on low-resolution context, whereas the other network was used to focus on high-resolution foveate. This strategy quadrupled the training process' speed. A set of 20 frames underwent enhancement through diverse cropping and flipping techniques, followed by four iterations of network execution for the purpose of action classification within the refined frames. In order to streamline the procedure and shorten the time needed for video action recognition, predictions from each frame were then averaged.

HAR was investigated in depth by [29], who looked at its potential in a variety of settings, including video games, exergames, and aid for individuals with neurological impairments. Machines can carry out particular activities based on human body gestures thanks to HAR technology, which makes it possible to identify these gestures. The ability to participate with games and exergames with ease utilizing basic gestures makes it easier for elderly people and folks with neurological injuries. Additionally, HAR enables surgeons to operate intraoperative imaging monitors with predetermined free-hand motions. Human activity recognition, which interprets human motion using computer and machine vision technologies, has generated a lot of interest because of its potential, applicability, and versatility in a variety of sectors.

Human activity identification was broken down into two categories in research by Danaei Mehr and Polat [30]: vision-based and sensor-based. The field of sensor technology includes a wide range of devices, including wearable sensors, sensors fastened to objects, and high-density sensing techniques. Another study by Chen, Liu, Peng, and Wu entitled "Deep Learning Based Multimodal Complex Human Activity Recognition Using Wearable Devices" looked at the body of work that uses a vision-based strategy for recognizing human activity. Unimodal and multimodal research were the two main categories used to classify this study. Stochastic (random) methods, rule-based methods, space-time methods, and shape-based methods are a few different ways to solve problems. Contrarily, multimodal approaches make use of information from several sources and can be further broken down into approaches that concentrate on social networking, emotions, and behavior.

The purpose of the study [11] was to continuously track over a long length of time the activities of elderly people in their homes using data acquired from a CNN based model. 15 subjects, 12 males and 3 females, between the ages of 25 and 50, took the same test nine times during the data collection phase. After that, using CNN and LSTM algorithms, the sensor data was divided into nine different activities. According to the study's results, CNN-based structures are more accurate than LSTM-based ones. The CNN-based architecture demonstrated an overall classification accuracy of 97 percent for the nine separate tasks, which included the identification of unusual behaviors that might be suggestive of dangerous health issues or emergency situations. Additionally, the system produced promising results even with a modest training dataset.

Temporal features are typically restricted to small areas in the field of hybrid networks, which combine CNNs with Recurrent Neural Networks (RNN) to form CNN-RNN networks. A less than ideal representation of sequence information is the result of this. A thorough representation of the data is needed to address this, and time-based aggregation approaches like temporal pooling can help with this. According to study paper [31], to process the sequence data produced by CNNs, a variety of temporal pooling techniques were combined with LSTM networks. When evaluated using public action datasets, these approaches showed to be extremely successful and efficient in terms of processing resources, obtaining an accuracy rate of more than 80%. In addition, the LSTM variation, which included 30 frames of optical flow images in addition to the initial frames, achieved a remarkable accuracy level of 88.6\%. Due to these remarkable results, two new action categorization networks, LRCN and ConvLSTM, were developed. They both have distinctive network architectures and encoder-decoder frameworks.

Using linear dynamical systems (LDS) and deep features, the research [34] presents a novel approach to action recognition in video. Deep features can produce more accurate representations of activities, according to the researchers, who contend that LDS can capture the temporal dynamics of actions more well than standard techniques. To recognize actions in fresh videos, the suggested methodology comprises training an LDS model on the deep features retrieved from video frames. A pre-trained CNN is used to extract these deep features from video frames, and then Principal Component Analysis (PCA) is used to condense their dimensionality. The LDS model that captures the temporal dynamics of actions is then trained using these reduced characteristics. On multiple benchmark datasets, including UCF101, HMDB51, and Hollywood2, the efficacy of this strategy is assessed, showing greater accuracy compared to state-of-the-art techniques. The significance of various parts of their approach is further examined through ablation studies, demonstrating the importance of both the LDS model and the deep features for reaching high accuracy. In conclusion, this suggested method offers a promising strategy for action recognition in films, possibly opening the door for ground-breaking uses in industries including surveillance, robotics, and human-computer interaction.

An approach to human activity recognition in video recordings based on deep learning is presented in [35]. The researchers address issues such size variations, inadequate lighting, improper perspectives, and background clutter related to HAR. They contrast Machine Learning and Deep Learning approaches to solving the HAR problem. Due to the deterministic nature of machine learning, the retrieved characteristics and action characterization must be defined, managed, and improved by the user. On the other hand, Deep Learning uses Deep Neural Networks (DNN) that operate similarly to the human brain to independently resolve all attributes. To train deep learning models for HAR, the researchers advise employing transfer learning models. These models have already been pre-trained using large datasets like ImageNet, which has over a million images that are ideal for used in transfer learning model training. Using the UCF50 action dataset, the effectiveness of three pre-trained deep learning model DenseNet, VGG19, and EfficientNet is assessed. The researchers use data from sizable databases like ImageNet by employing pre-trained deep learning. The neural network being trained in a new domain receives data using the transfer learning approach from an already trained model. On this dataset, the effectiveness of several deep learning models is examined, and their accuracy is assessed in comparison to cutting-edge methods. In the beginning, frames were taken from each group of action videos and entered a deep learning model that had already been trained. The EfficientNet model has the best testing accuracy, scoring 94.25 percent. The researchers also provide confusion matrices for classifying 50 activities from the UCF50 dataset with the use of the VGG19 model, DenseNet 161, and EfficientNet b7.

DATASET

Numerous techniques are available for recognizing human activities, which can leverage diverse sets of data. Certain datasets prioritize the collection of data from sensors such as accelerometers, ECG sensors, and gyroscopes, whereas others concentrate on video data. In the context of video datasets, certain compilations showcase videos portraying actors executing actions in pre- determined environments, whereas others comprise videos procured from platforms such as YouTube. The UCF50 database is employed in this specific implementation.

The UCF50 dataset, a creation of the University of Central Florida, is a collection of 50 distinct action categories featuring realistic videos sourced from YouTube. This dataset distinguishes itself from other action recognition datasets, which are often simulated, due to its inherent complexity. The UCF50 dataset poses a significant challenge due to its wide array of varying elements such as camera motion, object appearances, pose, perspective, size, background clutter, and lighting conditions. It encompasses a

diverse range of actions, from simple activities like walking a dog and diving to more complex tasks like horse riding and jumping rope. Each category of motion boasts an average of 133 unique video recordings, providing a substantial anthology for extensive analysis.

The videos categorized together could all have something in common, such focusing on the same person. There are fifty distinct groups that each video belongs to. The collection includes 6,618 films, demonstrating 50 different types of classes. In the training process using the UCF50 dataset, the primary aim is to classify the ten categories, specifically 'BaseballPitch', 'Basketball', 'BenchPress', 'Biking', 'Billiards', 'BreastStroke', 'CleanAndJerk', 'Diving', 'Drumming', and 'Fencing', out of the 50 activities present in the dataset. This approach was chosen due to constraints in both time and resources, along with the goal of speeding up the model training, considering the extensive size of the UCF50 dataset.

ConvLSTM and LRCN Approach

The primary aim of this study is to utilize deep learning methodologies for the purpose of recognizing human actions based on video data. In order to accomplish this objective, two distinct models, ConvLSTM and LRCN, have been employed to construct a system for recognizing human activities. The first stage of the process entailed the acquisition of video data, which was subsequently subjected to frame extraction and preprocessing. The pre-processed frames were employed to construct a dataset consisting of a predetermined number of images per category, which was subsequently utilized to train the model. Figure 1 depicts a streamlined framework of the suggested methodology.

Prior to the training phase, the categorical labels denoting various activities were transformed into numerical values through the utilization of One Hot Encoding. This method involves the representation of categorical variables as binary vectors. This methodology confers notable advantages in scenarios involving multi-classification quandaries, as it guarantees the independent treatment of each class, without presuming any ordinal association among them. The detailed explanation of the architecture of the two models employed is provided subsequently in this section. The number of epochs was specified as 400, representing the complete number of iterations that the training dataset was allowed to traverse through the network. The TensorFlow model training process is defined with the specification of two callback functions. The initial function generates a callback for TensorBoard, which generates a directory for logging purposes with a designated name and stores the training process log files. An early stopping callback is the model's second kind of callback. Its main job is to keep track of the validation loss and stop thetraining process when it doesn't get better after a certain number of epochs. Additionally restored by this callback are the model weights from the epoch with the best validation loss. If the validation loss does not decrease for a certain number of epochs, the third callback in the model is intended to lower the learning rate by a factor of 0.2. This strategy aims to reduce the danger of overfitting and enable quicker model convergence.

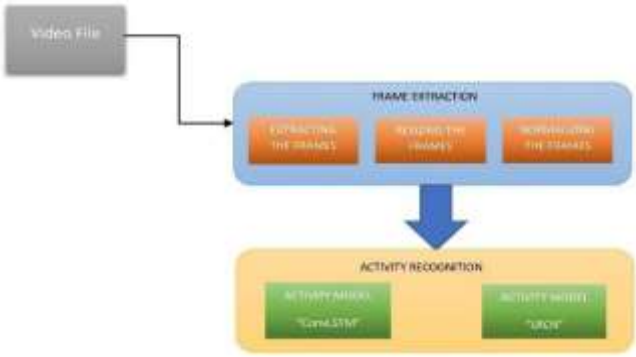

Figure 1: The architecture of the ConvLSTM and LRCN approach

4.1Data Preprocessing

The process of extracting frames is an essential preliminary measure for applications that are dependent on video analysis. The utilization of video data in machine learning algorithms necessitates the analysis of individual video frames. The process of extraction entails an iterative procedure whereby all video files contained within the directories of the dataset's respective classes are accessed. The OpenCV Video Capture method is utilized to read each file, after which the frames are extracted. The frames undergo processing and are resized to a standardized dimension of 64x64 pixels, following which their pixel values are normalized through division by 255, as per reference [36]. The frames that have undergone processing are provisionally retained in a roster for every category. Subsequently, a function is executed to extract frames from all videos encompassing all categories of actions. The optimization of the training process and enhancement of the model's accuracy can be achieved by ensuring uniformity in the dimensions of frames and standardizing pixel values within the range of 0 to 1.

4.2 The Proposed ConvLSTM Network for Action Recognition

The Convolutional Long Short-Term Memory network was initially examined in a scholarly investigation conducted by the author referenced as [36]. The study revealed that the preservation of spatial data is not achieved when a fully connected LSTM network is utilized to transform an image into a one-dimensional space. In order to address this issue, it is necessary to employ a CNN feature extractor that can effectively extract spatial features and subsequently transform them into a unidimensional vector space. The ConvLSTM network was devised as a solution for video classification tasks [3] and is capable of processing 2D convolutions as input. The ConvLSTM architecture utilizes convolutional operations on input images to extract spatial features, while incorporating LSTM layers to capture the temporal dynamics between frames. The ConvLSTM network has the ability to efficiently capture spatial and temporal signals, which is not possible with a fully connected LSTM in isolation.

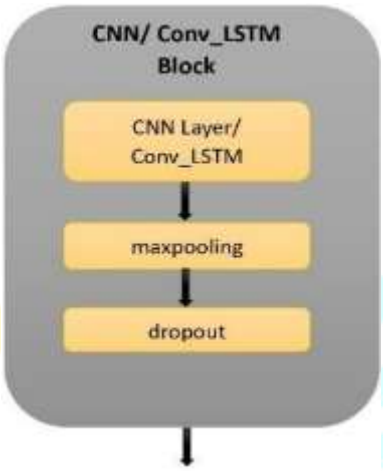

Figure 2: A Representative Convolutional LSTM Unit

The ConvLSTM network, applied in this research to expedite video categorization, is detailed as follows. A condensed version of the ConvLSTM network is suggested to optimize computational proficiency while maintaining high performance. This model includes four ConvLSTM2D layers and a final dense layer, as depicted in Figures 1 and 2. The action recognition model is constructed using TensorFlow's ConvLSTM2D implementation in Python. The construction process begins with the instantiation of the Sequential () class, which includes methods for creating a structurally feasible model blueprint.

The model uses ConvLSTM2D layers with a 3x3 kernel size and tanh activation function. The number of filters ranges from 8 to 64. It employs a synchronized many-to-many LSTM mode with return sequences set to true. A recurrent dropout rate of 0.2 is applied for efficiency, and the input shape is a 4D tensor. Each ConvLSTM2D layer is followed by a pooling layer that halves the resolution while preserving shift-invariance in the network.

Overfitting may be avoided by adding a time-distributed dropout layer after each pooling process, which discards 20% of the output frames. The ConvLSTM model is constructed by stacking several ConvLSTM blocks, each composed of ConvLSTM2D, pooling, and time-distributed dropout layers. Subsequently, a flatten layer is utilized to convert the 2D vector into a 1D vector. This is followed by the connection of the 1D vector to a dense layer that employs softmax activation. By selecting the class with the highest probability, this particular configuration allows the model to effectively predict the classification of the input video. The efficiency of the ConvLSTM model is evaluated using a large human action dataset, with varying numbers of frames extracted from the input video. The optimal number of frames is determined and then used in subsequent experiments to assess the model's performance.

The proposed ConvLSTM model is designed to effectively extract spatial-temporal dependencies. This model's performance is evaluated using a large dataset of human actions, with the number of frames from the input video varying. The optimal setup, determined by the number of frames, will be utilized in future experiments to assess the model's effectiveness.

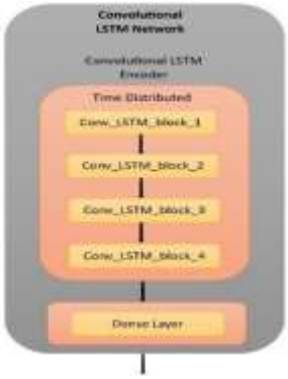

Figure 3: A ConvLSTM network sample comprising four ConvLSTM blocks and a singular dense layer.

4.3 Model size and Parameters of ConvLSTM

The ConvLSTM model in focus employs a sequence length of 20 input frames, an image resolution of 64 x 64 x 3, and a target output of 10 classes, leading to a total of 696,004 trainable parameters. In addition, the model, with its 10 output classes, comprises 369,098 trainable parameters. Table 1 illustrates the distribution of parameters across each layer. It's worth noting that max-pooling and dropout layers do not have trainable parameters, as they merely perform transformations.

The proposed model follows a sequential ConvLSTM architecture with an impressive 368,324 total parameters, adept at capturing spatiotemporal information with utmost efficiency. Comprising of four ConvLSTM2D layers, the model is fortified with time-distributed max- pooling and dropout layers, tailored to reducing spatial dimensions and preventing overfitting. Interestingly, the ConvLSTM2D layers undergo a gradual boost in intricacy, with the first layer adorned with 8 filters and 3,200 parameters, and succeeding layers exhibiting substantial growth in filters i.e., 16, 32, and 64, alongside a proportional increase in parameters, specifically 13,888, 55,424, and 221,440 parameters, correspondingly.

Layer	Output Shape	Parameters	Description
conv_lstm2d	(None, 20, 62, 62, 8)	3200	1st ConvLSTM layer with 20 filters (62x62), 8 channels
time_distributed	(None, 20, 31, 31, 8)	0	Apply same operation to each time step (no data modification)
time_distributed_1	(None, 20, 31, 31, 8)	0	Apply same operation to each time step (no data modification)
conv_lstm2d_1	(None, 20, 29, 29, 16)	13888	2nd ConvLSTM layer with 16 filters (29x29), 16 channels
time_distributed_2	(None, 20, 15, 15, 16)	0	Apply same operation to each time step (no data modification)
time_distributed_3	(None, 20, 15, 15, 16)	0	Apply same operation to each time step (no data modification)
conv_lstm2d_2	(None, 20, 13, 13, 32)	55424	3rd ConvLSTM layer with 32 filters (13x13), 32 channels
time_distributed_4	(None, 20, 7, 7, 32)	0	Apply same operation to each time step (no data modification)
time_distributed_5	(None, 20, 7, 7, 32)	0	Apply same operation to each time step (no data modification)
conv_lstm2d_3	(None, 5, 5, 64)	221440	4th ConvLSTM layer with 64 filters (5x5), 64 channels
max_pooling2d_3	(None, 3, 3, 64)	0	Reduce data dimensionality with max pooling (3x3 window)
dropout_3	(None, 3, 3, 64)	0	Randomly drop some data elements to prevent overfitting
Flatten	(None, 576)	0	Reshape data into a one-dimensional vector
Dense	(None, 128)	73856	Fully-connected layer with 128 neurons
dropout_4	(None, 128)	0	Randomly drop some data elements from the dense layer
dense_1	(None, 10)	516	Final output layer classifying data into 10 classes
Layer	**Output Shape**	**Parameters**	**Description**
			Classes: BaseballPitch, Basketball, BenchPress, Biking, Billiards, BreastStroke, CleanAndJerk, Diving, Drumming, Fencing
Total Parameters		368,324	
Trainable Parameters		368,324	All parameters can be adjusted during training

Table 1: ConvLSTM model parameter for 10 classes

After the ConvLSTM2D layer concludes, the model employs a sequence of additional layers to reshape the data and enable the fully connected (dense) layers. Starting with a max-pooling layer followed by a dropout layer, the data is then processed by a flattened layer to attain the desired shape. The first dense layer is then initialized with 128 neurons and a whopping 73,856 parameters, which is then teamed with, yet another dropout layer specifically designed to mitigate overfitting. Finally, the model culminates with an output layer that consists of a dense layer containing 4 neurons and 516 parameters. Notably, all the parameters in the model are trainable, as there are no non-trainable parameters, thereby ensuring that the model is highly adaptable during the training process.

4.4 The Proposed Long-term Recurrent Convolutional Network for Action Recognition

A sophisticated deep learning system, LRCN can extract sequential and visual features from a picture sequence. The versatility of the LRCN, which was first introduced by [30], is shown in applications including action recognition, picture captioning, and video description creation. This is possible because the LRCN makes use of several RNN variations.

The LRCN comprises two primary components, a CNN and an RNN. The CNN encoder's foremost purpose is to extract spatial characteristics and convert them into a one- dimensional vector. The resulting vector is then utilized as input for the RNN

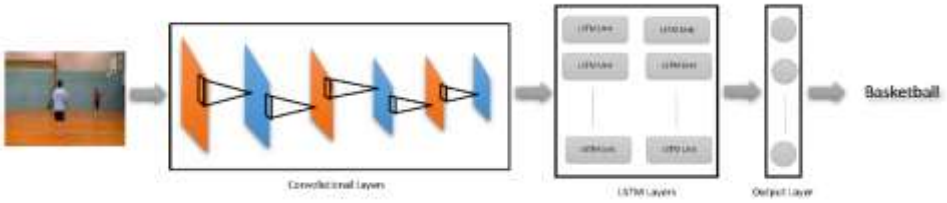

Figure 4: Combining Convolutional Layers, LSTM Layers, and an Output Layer for Advanced Feature Extraction and Prediction

encoder to deduce temporal dynamics. By employing shared weights, the network can simultaneously process multiple frames. The encoder-decoder design offers a substantial amount of flexibility in selecting architectures for both the CNN and RNN, as they function independently. Furthermore, since the recurrent neuron can receive variable-length input vectors, any flattened CNN architecture's final layer output can be employed as input, regardless of its dimensions.

This approach uses a condensed deep learning model that encompasses four CNN encoder blocks and an LSTM decoder layer, enabling efficient feature extraction and sequence analysis. The CNN blocks comprise Conv2D, BatchNormalization, MaxPooling2D, and Dropout layers, encapsulated by a TimeDistributed layer. After the CNN blocks, the LSTM layer is followed by a Dropout layer and a Dense layer for classification.

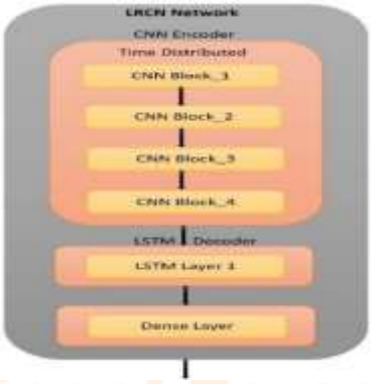

Figure 5: A LRCN network consisting of four convolutional neural network layers and a single long short-term memory layer.

4.5 Model size and Parameters of LRCN

The Long-term Recurrent Convolutional Network model, as presented here, is a fusion of CNNs and RNNs, with a particular emphasis on LSTM networks. The model begins with a series of time-distributed CNN layers, designed to process image data. In this case, they process video frames over a period of time. The CNN layers consist of four blocks, each containing a Conv2D layer, a BatchNormalization layer, a MaxPooling2D layer, and a Dropout layer. The Conv2D layers have 32, 64, 128, and 128 filters, respectively, and each uses a 3x3 kernel for convolution.

The MaxPooling2D layers are designed to reduce the spatial dimensions of the output volume. The Dropout layers help prevent overfitting by randomly setting a portion of input units to 0 during each update in the training phase. After the CNN layers, the model flattens the output and feeds it into an LSTM layer with 64 units. LSTM, a type of RNN, is adept at learning long-term dependencies, making it suitable for processing frame sequences over time.

The model concludes with a Dense layer that uses a softmax activation function to create a probability distribution across the classes for classification. This layer has the same number of units as the entire number of classes in the problem. This last layer is very important because it converts the high-level information that the network has learned into predictions for each class. It effectively acts as the model's decision-making element by deciding which class each input is most likely to belong to base on the characteristics that have been learned.

The total number of parameters in this model is a substantial 783,428. Of these, 782,724 are trainable parameters, while 704 are non-trainable parameters. The trainable parameters are those that the model can learn from the data during training, while the non-trainable parameters remain unchanged during training. This model is relatively large and may require significant

computational resources for training, especially with large datasets. However, the large number of parameters also gives the model a heightened ability to learn complex patterns within the data.

Layer	Output	Parameters	Description
time_distributed_6, 7	(20, 64, 64, 32)	1,024	Extract low-level features (edges, motion)
time_distributed_8, 9	(20, 32, 32, 32)	768	Combine features into complex patterns
time_distributed_10, 11	(20, 32, 32, 64)	19,296	Learn action-specific features (limbs, postures)
time_distributed_12, 13	(20, 16, 16, 64)	16,384	Analyze broader spatial context
time_distributed_14, 15	(20, 16, 16, 128)	78,408	Capture temporal relationships (action flow)
time_distributed_16	(20, 8, 8, 128)	57,344	Analyze overall structure and long-range dependencies
time_distributed_17, 18	(28, 8, 8, 128)	237,184	Refine and integrate features across time steps
time_distributed_19	(20, 8, 8, 128)	512	Prepare features for LSTM layer
time_distributed_20	(20, 4, 4, 2848)	20,480	Project features into high-dimensional space for LSTM
time_distributed_21	(None, 64)	0	Capture long-range temporal dependencies with bidirectional LSTM
Lstm	(None, 64)	540,928	
dropout_8	(None, 64)	0	Prevent overfitting
dense_2	(None, 10)	650	Classify action based on learned features
Total Params:		783,818	
Trainable Params:		783,114	
Non-trainable Params:		704	

Table 2: LRCN model parameter for 10 classes

Experimental Analysis

Throughout the research process, two separate models were evaluated on the previously mentioned dataset, UCF50. Each dataset was scrutinized independently, without any merging during the experimental process. The dataset was divided such that 80% was used for training and the remaining 20% was reserved for testing. During the model fitting phase, the validation split parameter was set to 0.2, indicating that 20% of the training data was utilized for validation purposes.

Two specific scenarios were investigated: the Convolutional LSTM with the UCF50 dataset and the LRCN with the UCF50 dataset. In both instances, the testing accuracy was documented and is presented in Table 3. The results indicate that the LRCN model outperforms the Convolutional LSTM model. The most impressive accuracy achieved is 97.01% for the LRCN model using the UCF50 dataset.

Model	ccuracy (%)	recision (%)
Convolutional LSTM	87.78%	87.69%
LRCN	96.67%	96.89%

Table 3: Accuracy and Precision of Models with UCF50 dataset

The performance of both models during their respective training phases was assessed by generating graphical representations of their overall loss versus validation loss, as well as their accuracy versus validation accuracy. These graphs can be observed in Fig. 6, Fig. 7, Fig. 8 and Fig. 9. The x-axis of these graphs denotes the number of epochs, while the y-axis displays either the rate of loss or the level of accuracy.

The "Training Loss" score reveals the degree of model error on the particular dataset it was trained on, whereas the "Validation Loss" score denotes the degree of error on unseen, novel data. The fundamental purpose of training a machine learning model is to minimize its loss, which entails reducing the amount of error committed by the model.

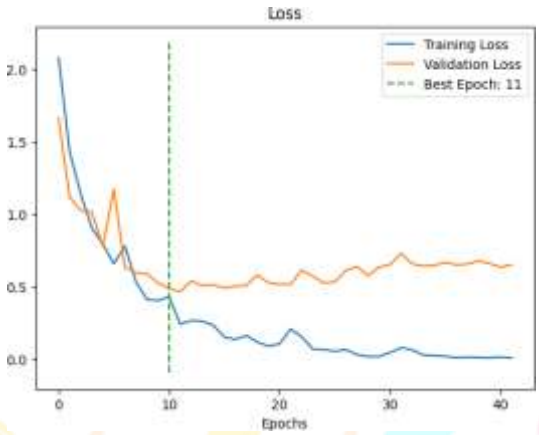

Figure 6: Total Loss vs. Total Validation Loss plot - Convolutional LSTM with the UCF50 dataset

It is vital to closely observe both the training and validation loss over time to ensure that the model continues to improve and that it is not overfitting, i.e., memorizing, the training dataset. The model's loss values are presented on the y-axis, while the x-axis portrays the number of epochs. The graphs depict two individual plotted lines: one for training loss, and the other for validation loss.

Notably, the "Best Epoch," which in this instance is identified as epoch 11, indicates the point in time at which the model attained its lowest validation loss. This is highly consequential as it represents the point in which the model performs most effectively on the validation dataset, and thus guards against overfitting.

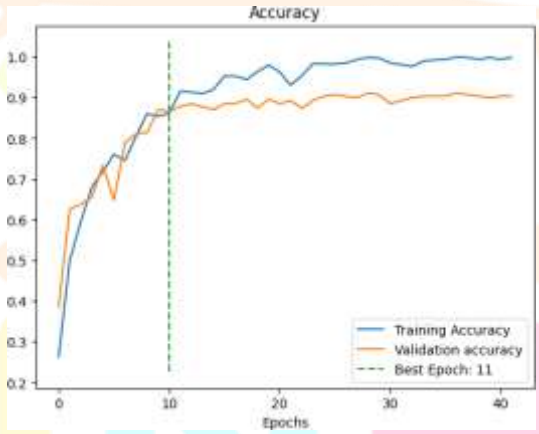

Figure 7: Total Accuracy vs. Total Accuracy Loss plot - Convolutional LSTM with the UCF50 dataset

The "Training Accuracy" metric illustrates the model's effectiveness in handling the data it was trained on. On the other hand, the "Validation Accuracy" score measures its effectiveness in handling new, unseen data. It's crucial to aim for a high validation accuracy to ensure the model is generalizing effectively, rather than just memorizing the training data. Furthermore, the

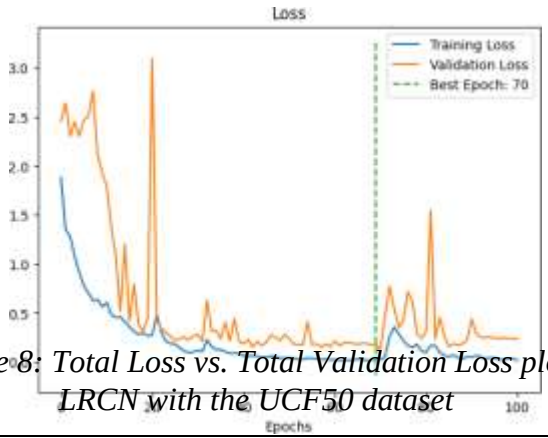

Figure 8: Total Loss vs. Total Validation Loss plot - LRCN with the UCF50 dataset

"BestEpoch" metric identifies the epoch at which the model reached the highest validation accuracy. Based on the available data, it appears that the model exhibited its best performance during epoch 11.

The term "Best Epoch" denotes the epoch with the minimum validation loss. Based on the current analysis, it appears that epoch 70 delivered the most optimal performance for the model.

The "Best Epoch" score indicates the epoch with the highest validation accuracy. In this case, the model appears to have performed best on epoch 70. It is important to balance the trade-off between maximizing accuracy and ensuring the model generalizes well to new data.

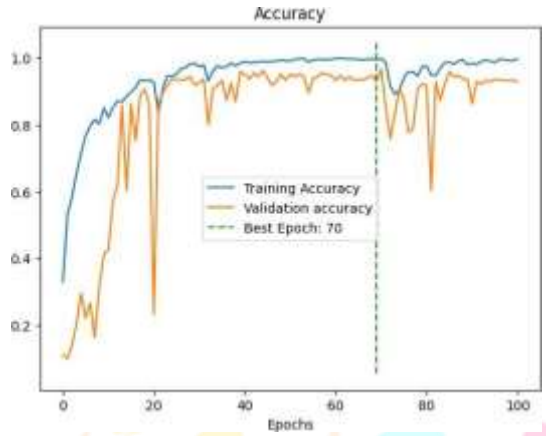

Figure 9: Total Accuracy vs. Total Accuracy Loss plot - LRCN with the UCF50 dataset

To demonstrate the relationship between predicted labels (x-axis) and true labels (y-axis), as well as to display the performance of each chosen action class, a heatmap is used to represent the confusion matrix for both scenarios.

A confusion matrix typically includes four categories: True Positive, False Positive, True Negative, and False Negative. In this study, the diagonal of the confusion matrix or heatmap represents the activities correctly recognized. The remaining cells indicate activities predicted as a different activity, such as a 'BaseballPitch' video being identified as 'Basketball'. In the heatmaps displaying each model's performance on the UCF50 dataset, the numbers 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9 represent 'BaseballPitch', 'Basketball', 'BenchPress', 'Biking', 'Billiards', 'BreastStroke', 'CleanAndJerk', 'Diving', 'Drumming', 'Fencing', respectively, on both the x-axis and y-axis.

It is imperative to arrive at a conclusive determination concerning the sufficiency of the outcomes and the extent of variations between them. In addition, it is vital to scrutinize any potential variables that could have influenced the final results. Once a precise comprehension of the discoveries is obtained, logical deductions can be made. To sum up, evaluating the effectiveness of the algorithm in accomplishing its intended purpose, and subsequently outlining forthcoming enhancements and objectives is of utmost importance.

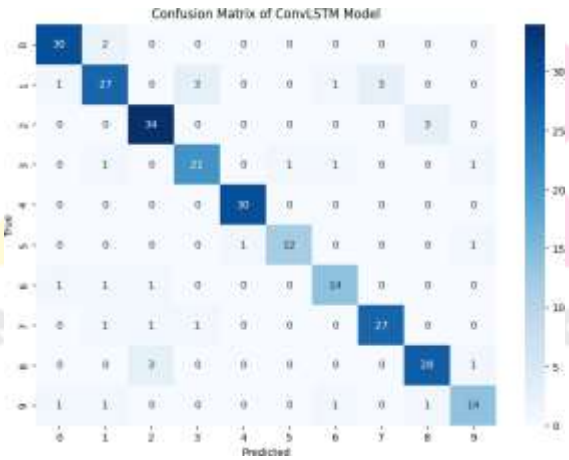

Figure 10: The result from classification by ConvLSTM

The comparative analysis between LRCN and ConvLSTM models reveals a distinct advantage of the former in terms of both precision and accuracy. This superior performance of the LRCN model can be primarily attributed to its inherent capability for superior temporal distribution and continuity processing. An in-depth examination of the LRCN's confusion matrix reveals that the network misclassified a single video from the 'Basketball' category as 'Biking', while it accurately classified 32 other types of videos. The network demonstrated exceptional performance across most other categories, with only one or three errors occurring infrequently. This indicates a high degree of accuracy in classifying categories without any mislabeling.

Conversely, the ConvLSTM model exhibited a higher rate of misclassification, as substantiated by its confusion matrix. Its overall accuracy, standing at 93.28%, was inferior to that of the LRCN. The ConvLSTM model encountered the most difficulty when classifying videos from the 'BenchPress' category, mislabeling three videos as 'Drumming', one as 'CleanAndJerk', and one as 'BreastStroke'. Additionally, in the 'Basketball' category, it misclassified two videos as 'BaseballPitch', one as 'Biking', one as 'CleanAndJerk', one as 'Diving', and one as 'Fencing'. Unlike the LRCN, the ConvLSTM model was unable to accurately identify all instances of any single video category without any errors.

It is important to note that the performance of these neural network structures could potentially be improved by adjusting the number of filters and their kernel dimensions. Additionally, incorporating more layers could also influence the results. However, within the context of the fundamental ConvLSTM and LRCN architectures, the accuracy outcomes were deemed satisfactory. In

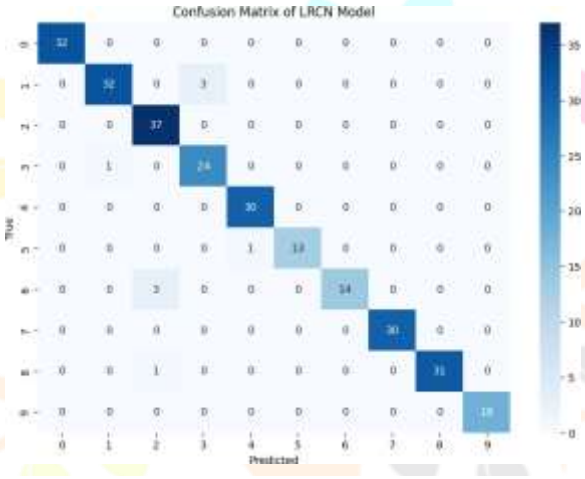

Figure 11: The result from classification by LRCN

conclusion, the LRCN architecture demonstrated superior performance, with significantly higher accuracy and precision values compared to the ConvLSTM approach. The enhanced classification accuracy of the LRCN model is clearly evident in the confusion matrix, which distinctly showcases an improved categorization of videos into their respective classes.

CONCLUSION

This research paper delves deeply into the function of artificial neural networks in relation to video classification, highlighting the rapidly increasing significance of these technologies in today's contemporary applications. Using the Anaconda development environment in conjunction with TensorFlow and Keras libraries is crucial in illustrating the vital role that neural networks play in video classification. Therefore, it is recommended that these technologies should be incorporated into educational curricula to promote a more widespread understanding of their potential.

The LRCN architecture, which integrates enhanced time continuity processing as well as a time distribution layer, proved to be remarkably accurate with an impressive accuracy rate of 96.67%. The ConvLSTM model, on the other hand, achieved an accuracy rate of 87.78% upon closer examination of its confusion matrix, it was clear that more frequent classification errors and a larger number of inaccuracies were present.

Looking forward, the advancements made in this paper illuminate a promising path for future research. The primary step on this path involves diversifying and enriching the UCF50 dataset. This can be achieved by adding training videos that showcase individuals from a variety of ethnic backgrounds, ensuring a comprehensive and inclusive representation within the data.

Simultaneously, the focus is on augmenting the capacity of the proposed CNN-LSTM models to effectively handle larger datasets, such as the extensive Kinetics 700. As these models adapt to such large datasets, their accuracy and predictive power are expected to significantly improve, propelling the pace of research in the field of human activity recognition technology.

Moreover, these models hold immense potential for practical applications. One such application could be their integration into home surveillance systems. By analyzing footage of everyday activities, these models could detect incidents that pose safety risks, such as falls. Coupled with an alert system that promptly notifies caregivers or family members via SMS, this technology could greatly enhance safety in home environments.

In conclusion, by expanding and improving the dataset, refining the models, and exploring practical applications, the aim is to continue making significant contributions to the advancements in human activity recognition technology.

REFERENCES

[1] Zhang, H., Xiao, Z., Wang, J., Li, F., & Szczerbicki, E. (2020b). A Novel IoT-Perceptive Human Activity Recognition (HAR) Approach Using Multihead Convolutional Attention. IEEE Internet of Things Journal, 7(2), 1072–1080. https://doi.org/10.1109/jiot.2019.2949715

[2] Ghate, V., & C, S. H. (2021b). Hybrid deep learning approaches for smartphone sensor-based human activity recognition. Multimedia Tools and Applications, 80(28–29), 35585–35604. https://doi.org/10.1007/s11042-020-10478-4

[3] Yen, C. T., Liao, J. X., & Huang, Y. K. (2020b). Human Daily Activity Recognition Performed Using Wearable Inertial Sensors Combined With Deep Learning Algorithms. IEEE Access, 8, 174105–174114. https://doi.org/10.1109/access.2020.3025938

[4] Jain, A., & Kanhangad, V. (2018). Human Activity Classification in Smartphones Using Accelerometer and Gyroscope Sensors. IEEE Sensors Journal, 18(3), 1169–1177. https://doi.org/10.1109/jsen.2017.2782492

[5] Huaijun Wang, Jing Zhao, Junhuai Li , Ling Tian, Pengjia Tu, Ting Cao, Yang An, Kan Wang, and Shancang Li, "Wearable Sensor-Based Human Activity Recognition Using Hybrid Deep Learning Techniques", 2020, https://doi.org/10.1155/2020/2132138

[6] Subasi, A., Dammas, D. H., Alghamdi, R. D., Makawi, R. A., Albiety, E. A., Brahimi, T., & Sarirete, A. (2018). Sensor Based Human Activity Recognition Using Adaboost Ensemble Classifier. Procedia Computer Science, 140, 104–111. https://doi.org/10.1016/j.procs.2018.10.298

[7] Fullerton, E., Heller, B., & Munoz-Organero, M. (2017b). Recognizing Human Activity in Free-Living Using Multiple Body-Worn Accelerometers. IEEE Sensors Journal, 17(16), 5290– 5297. https://doi.org/10.1109/jsen.2017.2722105

[8] Ihianle, I. K., Nwajana, A. O., Ebenuwa, S. H., Otuka, R. I., Owa, K., & Orisatoki, M. O. (2020b). A Deep Learning Approach for Human Activities Recognition From Multimodal Sensing Devices. IEEE Access, 8, 179028–179038. https://doi.org/10.1109/access.2020.3027979

[9] Wan, S., Qi, L., Xu, X., Tong, C., & Gu, Z. (2019b). Deep Learning Models for Real-time Human Activity Recognition with Smartphones. Mobile Networks and Applications, 25(2), 743– 755. https://doi.org/10.1007/s11036-019-01445-x

[10] Xia, K., Huang, J., & Wang, H. (2020b). LSTM-CNN Architecture for Human Activity Recognition. IEEE Access, 8, 56855–56866. https://doi.org/10.1109/access.2020.2982225

[11] Bianchi, V., Bassoli, M., Lombardo, G., Fornacciari, P., Mordonini, M., & De Munari, I. (2019b). IoT Wearable Sensor and Deep Learning: An Integrated Approach for Personalized Human Activity Recognition in a Smart Home Environment. IEEE Internet of Things Journal, 6(5), 8553–8562. https://doi.org/10.1109/jiot.2019.2920283

[12] Demrozi, F., Pravadelli, G., Bihorac, A., & Rashidi, P. (2020b). Human Activity Recognition Using Inertial, Physiological and Environmental Sensors: A Comprehensive Survey. IEEE Access, 8, 210816–210836. https://doi.org/10.1109/access.2020.3037715

[13] Almaslukh, B., Artoli, A., & Al-Muhtadi, J. (2018b). A Robust Deep Learning Approach for Position-Independent Smartphone-Based Human Activity Recognition. Sensors, 18(11), 3726. https://doi.org/10.3390/s18113726

[14] Bozkurt, F. (2021b). A Comparative Study on Classifying Human Activities Using Classical Machine and Deep Learning Methods. Arabian Journal for Science and Engineering, 47(2), 1507– 1521. https://doi.org/10.1007/s13369-021-06008-5

[15] Yang, Z., Qu, M., Pan, Y., & Huan, R. (2022c). Comparing Cross-Subject Performance on Human Activities Recognition Using Learning Models. IEEE Access, 10, 95179–95196. https://doi.org/10.1109/access.2022.3204739

[16] Nafea, O., Abdul, W., Muhammad, G., & Alsulaiman, M. (2021b). Sensor-Based Human Activity Recognition with Spatio-Temporal Deep Learning. Sensors, 21(6), 2141. https://doi.org/10.3390/s21062141

[17] Khan, I. U., Afzal, S., & Lee, J. W. (2022c). Human Activity Recognition via Hybrid Deep Learning Based Model. Sensors, 22(1), 323. https://doi.org/10.3390/s22010323

[18] Sri Harsha, N. C., Anudeep, Y. G. V. S., Vikash, K., & Ratnam, D. V. (2021b). Performance Analysis of Machine Learning Algorithms for Smartphone-Based Human Activity Recognition. Wireless Personal Communications, 121(1), 381–398. https://doi.org/10.1007/s11277-021- 08641-7

[19] Agarwal, P., & Alam, M. (2020). A Lightweight Deep Learning Model for Human Activity Recognition on Edge Devices. Procedia Computer Science, 167, 2364–2373. https://doi.org/10.1016/j.procs.2020.03.289

[20] Deep, S., & Zheng, X. (2019). Leveraging CNN and Transfer Learning for Vision-based Human Activity Recognition. 2019 29th International Telecommunication Networks and Applications Conference (ITNAC). https://doi.org/10.1109/itnac46935.2019.9078016

[21] Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., & Paluri, M. (2018). A Closer Look at Spatiotemporal Convolutions for Action Recognition. https://doi.org/10.1109/cvpr.2018.00675

[22] Wang, L., Xu, Y., Cheng, J., Xia, H., Yin, J., & Wu, J. (2018). Human Action Recognition by Learning Spatio-Temporal Features With Deep Neural Networks. IEEE Access, 6, 17913– 17922. https://doi.org/10.1109/access.2018.2817253

[23] Lee, I., Kim, D. Y., Kang, S., & Lee, S. (2017). Ensemble Deep Learning for Skeleton-Based Action Recognition Using Temporal Sliding LSTM Networks. https://doi.org/10.1109/iccv.2017.115

[24] Gowda, S. N., Rohrbach, M., & Sevilla-Lara, L. (2021). SMART Frame Selection for Action Recognition. Proceedings of the . . . AAAI Conference on Artificial Intelligence, 35(2), 1451– 1459. https://doi.org/10.1609/aaai.v35i2.16235

[25] Zheng, Y., Liu, Z., Lu, T., & Wang, L. (2020). Dynamic Sampling Networks for Efficient Action Recognition in Videos. IEEE Transactions on Image Processing, 29, 7970–7983. https://doi.org/10.1109/tip.2020.3007826

[26] Karpathy, A., Toderici, G., Shetty, S., Leung, T. W., Sukthankar, R., & Fei-Fei, L. (2014). Large-Scale Video Classification with Convolutional Neural Networks. https://doi.org/10.1109/cvpr.2014.223

[27] Feichtenhofer, C., Fan, H., Malik, J., & He, K. (2019). SlowFast Networks for Video Recognition. https://doi.org/10.1109/iccv.2019.00630

[28] Xiao, F., Lee, Y. R., Grauman, K., Malik, J., & Feichtenhofer, C. (2020). Audiovisual SlowFast Networks for Video Recognition. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2001.08740

[29] Beddiar, D. R., Nini, B., Sabokrou, M., & Hadid, A. (2020). Vision-based human activity recognition: a survey. Multimedia Tools and Applications, 79(41–42), 30509–30555. https://doi.org/10.1007/s11042-020-09004-3

[30] Mehr, H. D., & Polat, H. (2019). Human Activity Recognition in Smart Home With Deep Learning Approach. https://doi.org/10.1109/sgcf.2019.8782290

[31] Ng, J. Y., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G. (2015). Beyond Short Snippets: Deep Networks for Video Classification. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1503.08909

[32] Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning Spatiotemporal Features with 3D Convolutional Networks. https://doi.org/10.1109/iccv.2015.510

[33] Carreira, J., & Zisserman, A. (2017). Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. https://doi.org/10.1109/cvpr.2017.502

[34] Du, Z., Mukaidani, H., & Saravanakumar, R. (2020). Action Recognition Based on Linear Dynamical Systems with Deep Features in Videos. https://doi.org/10.1109/smc42975.2020.9283429

[35] Kumar, R., & Kumar, S. (2023). Light-Weight Deep Learning Model for Human Action Recognition in Videos. https://doi.org/10.1109/iscon57294.2023.10111975

[36] Megrhi, S., Jmal, M., Souidene, W., & Beghdadi, A. (2016). Spatio-temporal action localization and detection for human action recognition in big dataset. Journal of Visual Communication and Image Representation, 41, 375–390. https://doi.org/10.1016/j.jvcir.2016.10.016

[37] Shi, X., Chen, Z., Wang, H., Yeung, D., Wong, W., & Woo, W. (2015). Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1506.04214

[38] Luo, W., Liu, B., & Gao, S. (2017). Remembering history with convolutional LSTM for anomaly detection. https://doi.org/10.1109/icme.2017.8019325