International Journal of Advances in Electrical Engineering

E-ISSN: 2708-4582
P-ISSN: 2708-4574
IJAEE 2024; 5(1): 01-09

www.electricaltechjournal.com
Received: 01-11-2023
Accepted: 05-12-2023

Harish Chandra Sati
CSE-AIT, Chandigarh University, Punjab, India

Tanish
CSE-AIT, Chandigarh University, Punjab, India

Correspondence
Harish Chandra Sati
CSE-AIT, Chandigarh University, Punjab, India

Human activity detection using deep learning approaches

Harish Chandra Sati and Tanish

DOI: https://doi.org/10.22271/27084574.2024.v5.i1a.46

Abstract
One of the most crucial computer vision jobs is the recognition of human activity, which has proven beneficial in a variety of industries like security, healthcare, and athletic training. Several methods have been investigated to do this task, some of which make use of sensor data and others use video data. In this paper, we explore two deep learning-based methods to recognize human activities in videos: convolutional long short-term memory and single frame convolutional neural networks (CNNs). It is preferable to use a convolutional neural network-based approach since CNNs can automatically extract features and long short-term memory networks are excellent for working with sequence data, like video. The two models were trained and assessed using an action recognition benchmark. On the UCF50 benchmark action recognition dataset as well as a second dataset generated specifically for the experiment, the two models were trained and assessed. Despite the good accuracy of both models, the single frame.

Keywords: Human activity recognition, convolutional neural network (CNN), convolutional LSTM, Open CV

Introduction
Recognizing human activity is important for interpersonal interactions and human-to-human communication. It is challenging to extract since it contains details about a person's identity, personality, and psychological condition. One of the key research topics in the fields of computer vision and machine learning is the human capacity for activity recognition. This research has led to the need for a multiple activity detection system in numerous applications, such as video surveillance systems, human-computer interaction, and robotics for characterizing human behavior. [1]

The dynamics of family life are changing as contemporary society advances, with an increasing number of parents having to care for their children and elderly relatives while they are away at work. This raises serious issues regarding the protection, general welfare, and supervision of weaker family members who could need aid or monitoring while the parents are away. [2]A promising technology called human activity recognition (HAR) has evolved that can help with these issues by remotely monitoring and giving information on the activities of young children and elderly family members at home. In critical locations like airports, banks, or military bases, HAR can be utilized to identify suspect activity or intrusions. Overall, HAR is an area of research and development that is fast expanding and has a wide range of applications in numerous disciplines and businesses. [3]

In an effort to address the issue of human activity recognition, various strategies have been researched in the past. A few of them are based on sensor data, while others are based on video data [1]. A wearable sensor of some kind must typically be worn on the body of the subject being photographed in order to use sensor data. Wearing a sensor may not always be possible in situations like the home setting, and this means that this strategy may not always be simple to put into practice. Another method involves using a camera to record a video and then identifying the activity from the video. In situations where it would be challenging, this strategy would be more appropriate. [4]

A few studies have investigated approaches to the problem of recognizing human activity in videos. A hierarchical codebook model of local spatio-temporal video volumes serves as the foundation for one such technique [5]. This approach is founded on a utilizing a bag of video words (BOV) representation and not requiring prior action knowledge motion estimation or background subtraction. But a significant flaw with this approach is that such A visual depiction of events in a setting cannot be used to predict behavior over the long run. understanding, for instance, actions that include a lot of events that happen in order. Other methods like Hidden Markov Models (HMM) [6] and Support Vector Machines (SVM) [7] have also undergone testing to determine how well they recognize human movement.

In this research, we provide two deep learning-based methods for anticipating human behavior in videos.

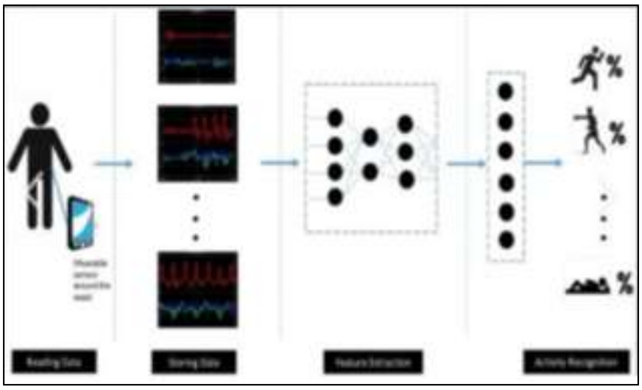


Fig 1

Convolutional LSTM is the second model, whereas single frame CNN is the first. We want to look into how well these models perform on the datasets. Benchmark action recognition dataset UCF50 is the first dataset we are using to train and test the model. [8]

The model we develop ought to be able to anticipate behaviors in a movie where a variety of diverse activities take place one after another. In order to evaluate which model works better, we also compare the results of the two models based on their testing accuracies.

Algorithms

Long Short-Term Memory

A Recurrent Neural Network (RNN) variant known as Long Short-Term Memory (LSTM) has been successfully applied to a number of sequential data-related applications, including Human Activity Recognition (HAR). Like other RNNs, LSTM models are made to assess data sequences and store internal memories of earlier inputs. This allows them to maintain the temporal relationships between various sequence segments. [11] The main advantage of LSTMs above all other RNNs is their ability to consciously forget or keep data from earlier time steps. In ordinary RNNs, vanishing gradients are a problem that this helps to resolve. In the input sequence, LSTMs can successfully replicate long-term dependence. They work well for challenging HAR tasks like detecting anomalies and distinguishing intricate human activities. [12]

In numerous benchmark datasets, LSTM-based models showed significant improvements in HAR tasks, achieving state-of-the-art performance. They have additionally demonstrated tenacity in spotting intricate activities and coping with input sequences of varying duration. [13] However, just like other deep learning models, LSTMs have a number of disadvantages for HAR, including the need for enormous amounts of labeled data, computational expense, and model interpretability.

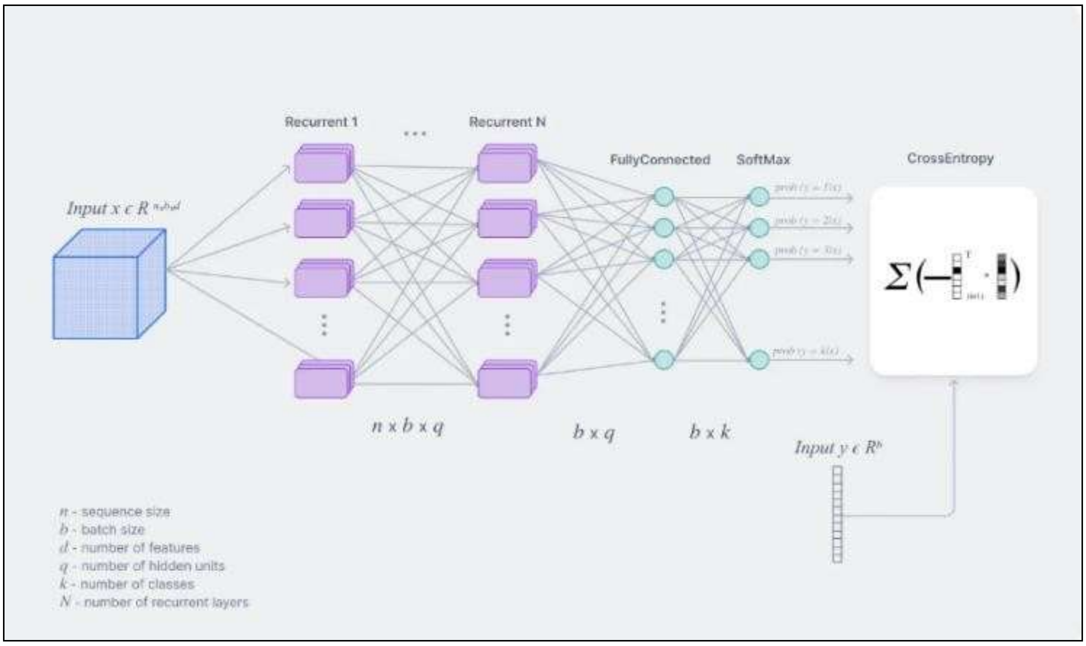


Fig 2: RNN-LSTM basic outltine

A Recurrent Neural Network (RNN) variant known as Long Short-Term Memory (LSTM) has been successfully applied to a number of sequential data-related applications, including Human Activity Recognition (HAR). [19] Like other RNNs, LSTM models are made to assess data sequences and store internal memories of earlier inputs. This allows them to maintain the temporal relationships between various sequence segments. [20]

Convutional Neural Networks

A deep learning architecture that excels in processing image and video data is convolutional neural networks (CNNs). In the context of Human Activity Recognition (HAR), CNNs have been used to automatically and accurately identify and categorize human actions from sensor data. [14]

Time-series data collected by sensors is frequently used as the input data for HAR using CNNs. Time is used as the x-axis and sensor data is used as the y-axis in a first transformation of the time-series data into a 2D image-like format. In order to extract and categorize characteristics, the resulting data matrix is subsequently fed into CNN. CNN's convolutional layers apply filters to the incoming data using a sliding window approach. Each filter extracts a certain feature, such as edges or corners, from the input data at different places. [15]

The pooling layers take the output of the convolutional layers and down sample the recovered features while preserving their vital spatial correlations. [16] The output of the pooling layers is then smoothed and sent to fully connected layers, which categorize the retrieved features into different human activities. [17] The softmax function then creates a probability distribution over the different activities using the output of the fully linked layers.

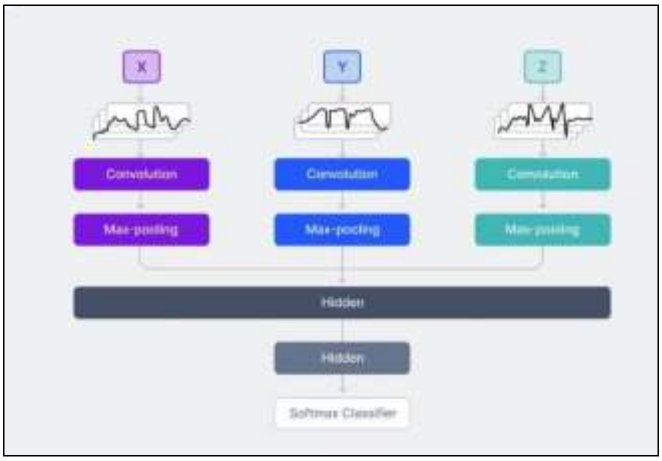


Fig 3: CNN basic outline

The ability of CNNs to process input data of diverse shapes and sizes enables them to analyze sensor data from a variety of devices. [18] Additionally, CNNs may learn hierarchical feature representations of the data input, enabling them to learn both the low-level and high-level components necessary for identifying human behavior.

Dataset

There are various kinds of datasets available for use in recognizing human behavior. Some of these datasets concentrate on information gathered by sensors like gyroscopes, accelerometers, and ECG sensors, while others concentrate on video data. Various video datasets are available, some of which contain actors staging acts in planned settings, while others contain videos that have been gathered from websites like YouTube. We have used the UCF50 benchmark action recognition video dataset as well as a dataset that we have built for our implementation. The UCF50 dataset consists of 50 action categories that each contain genuine YouTube videos [21]. Due to the large differences in camera movements, item look and posture, viewpoint, object scale, cluttered background, lighting condition, etc., the dataset is pretty difficult. [22]

The 50 action categories in the UCF50 data set that were gathered from YouTube are: Baseball Pitch, Basketball Shooting, Bench Press, Biking, Billiards Shot, Breaststroke, Clean and Jerk, Diving, Drumming, Fencing, Golf Swing, Playing Guitar, High Jump, Horse Race, Horse Riding, Hula Hoop, Javelin Throw, Juggling Balls, Jump Rope, Jumping Jack, Kayaking, Lunges, Military Parade, Mixing Batter Indoor rock climbing, rope climbing, rowing, salsa spinning, skate boarding, skiing, skijet, playing the tabla, Tai Chi, tennis swinging, trampoline jumping, playing the violin, volleyball spiking, walking a dog, and yo yo are some of the activities you can do.

The list of activities includes horseback riding, jumping rope, diving, and dog walking. There are typically 133 videos in each action group. [23]

Walking, sitting, and jumping are the three fundamental activities that make up the dataset we created. Some of the movies in our dataset depict members of our family or friends carrying out the tasks, while the others were pulled from YouTube and websites that offer free stock video footage. Because of how the data was gathered, the dataset mostly contains Indian faces. There are 21 videos in each area of action, with runs between 3 and 15 seconds.

Fig 4

Proposed Approach

In this research, we propose to recognize human behaviors from video data using deep learning algorithms.

To create the human activity recognition system, we employ two models: Single Frame Convolutional Neural Networks and ConvLSTM. We start by gathering video data, after which we take a step to preprocess the video files and extract the frames from them. After that, a dataset with a predetermined number of images for each class is created using the preprocessed frames. The model is then trained using this dataset. Figure 5 shows the suggested approach's basic architecture.

To begin the training, the labels (activities) are One Hot Encoded. Categorical variables can be expressed as numerical values using one hot encoding, where each distinct value is represented by a different binary feature. This method is particularly helpful for multi-classification problems like ours. We can guarantee that each class is handled individually and that the model does not assume an ordinal relationship between the classes by utilizing a single hot encoding. Following that, we can move on to training our models. In a later section of this article, we will go into more detail about the architecture of the two models we are employing. Early stopping callback is used to check the validation loss while the model is being trained. The epochs are set to 50 during model fitting, indicating the maximum number of times the training dataset can travel through the network. However, the early ending callback will end the training process early if the validation loss does not decrease for a predetermined number of epochs (in this case, we've set that predetermined amount to 15). When the training is over, the best weights are recovered. By stopping the epochs before they begin to over fit, early stopping reduces the model training time and prevents over fitting.

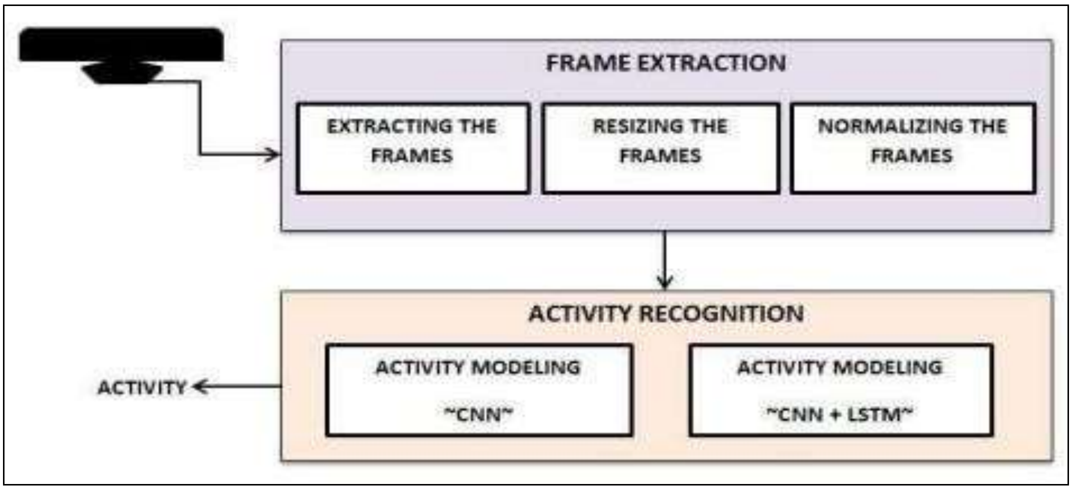


Fig 5: Architecture of the proposed approach

Data Preprocessing

For video-based applications, the frame extraction process is a crucial pre-processing step. Machine learning algorithms that use video must have access to each every frame in order to assess it. Iterating through the entire video is necessary for the video frame extraction process. [24] files in the class directories of the dataset, reading each video file with the Open CV Using the Video Capture technique, which extracts the video frames before iterating through each one, It should be resized to a fixed size of 64x64 pixels, and the pixel values should be normalized by 255. Then, a temporary list is created for each class with the preprocessed frames. Later, we execute a code to extract the frames from all of the action-related videos. The training process is simplified and helps to improve the model's performance and accuracy by resizing the frames to a fixed image height and width and normalizing the pixel values to a value between 0 and 1.

Feature Extraction Using Vision Transformer

The architecture of ViT-Base-16 is entirely based on the standard transformer architecture and achieved remarkable accuracy when compared to CNN-based models for image classification tasks. It uses self-attention mechanism to capture long-range relationship between input sequences. [25] ViT is actually an attempt to use transformer model for image classification. Basically, it divides the input image into a number of patches that are linearly projected with learnable positional embedding to learn the order of patches followed by transformer encoder with multilayer perceptron for final classification. [26]

In the first part, the input image is divided into non overlapping patches, because a standard transformer receives 1D sequence of token as an input. Usually, the image is in 2D format; therefore, to handle the 2D image, an image is reshaped into a sequence of flattened 2D patches. Herein, represents the height, width, and channels of the image, while is the resolution of each image patch, and is the total number of patches. [27] Typically, the patch size is chosen as 16×16 *or* 32×32, where the small *P* size is able to capture longer sequences and vice versa. In our case, we have used the for features extraction; in the subsequent section, these sub modules are discussed in detail.

Single Frame CNN

Convolutional Neural Networks, or CNN, is a deep learning method that has been widely applied throughout the years to address computer vision challenges including object identification and image classification. CNNs are created in such a way that they can automatically learn hierarchical representations of data, drawing inspiration from the structure and operation of the human visual system.

We're trying to find a technique that would take a video as input and output the activity that is being done in the video. We would run a model that would do image classification on each and every frame of the movie in this single frame CNN approach to identify the action being performed. For each frame of the input video, the model creates a probability vector that indicates the likelihood that various activities will be present in that frame. The final output probabilities vector would then be obtained by averaging all the individual probabilities. [28]

It would not be practical or essential to run the classification model on every frame of the videos because they can last for many seconds. Running the model on a few frames scattered over the entire movie might be adequate. [29]

Two convolutional layers, a batch normalization layer, a max-pooling layer, a global average pooling layer, and two fully connected (dense) layers make up the architecture of a convolutional neural network (CNN). [30] A blank model object is created and can be used to sequentially add layers to the model. A set of 64 filters (also known as kernels) with a combined size of 3x3 are applied to the input image by the model's first convolutional layer. Rectified Linear Unit (ReLU), which is employed as the activation function for this layer, is defined as

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

Additionally, the model's second convolutional layer applies a total of 64 filters to the input image, where each filter measures 3 by 3. ReLU is also the activation function employed for this layer. The second convolutional layer is followed by a batch normalizing layer, which is added to the model. This surface normalizes the output of the preceding layer, which helps to stabilize the learning process. A 2D max The output of the preceding layer is then subjected to maximum pooling by the pooling layer, with a pool size of (2, 2). The output's spatial dimensions are reduced as a result. then comes a 2D global average pooling layer. used to calculate the average of each feature map's values, producing a single value for each feature map. [31]

The output's dimensions are further diminished as a result. Afterwards, a completely connected (dense) The model gets a layer. This employs ReLU activation and has 256 units. After the dense layer, the model is given another batch normalizing layer. The second dense layer serves as the final layer of output. There are as many nodes in this layer as there are classes in the In our example, the categorization task is the quantity of human acts. The method of activation The softmax function for this layer transforms the output of the layer into a probability spread among the classrooms.

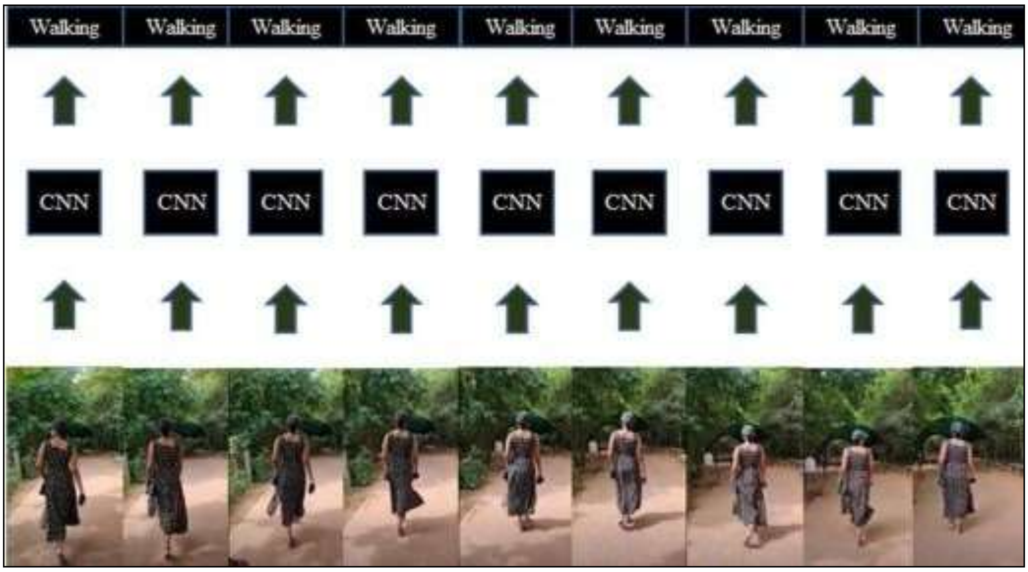


Fig 6: Single Frame CNN

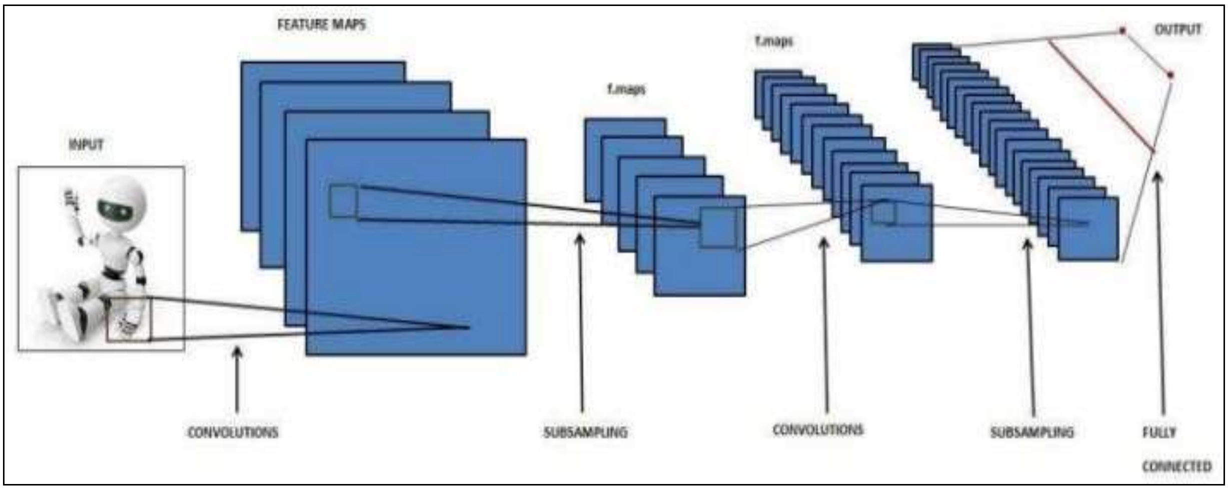


Fig 7: CNN architecture

Convolutional LSTM

A variation of the Long Short-Term Memory (LSTM) network architecture called Convolutional LSTM (ConvLSTM) is made to handle sequential data with spatial structures, such video data [18]. 2D For processing spatiotemporal data in the form of two-dimensional (2D) arrays, such as image sequences or video frames, ConvLSTM is a subclass of Convolutional LSTM. Traditional CNNs are effective in processing picture data; as a result, they may be used to extract spatial properties from individual frames. However, because LSTM networks excel at simulating temporal dependencies, they are well suited for using sequence data. ConvLSTM is a useful method for tackling computer vision issues like video categorization because it combines the strength of CNN and LSTM networks to efficiently capture both spatial and temporal aspects of the input.

Similar to conventional LSTMs, the 2D ConvLSTM adds a spatial convolution operation to the input and recurrent connections. A 2D ConvLSTM cell has a 2D array as its input and output and can be layered to create deeper networks that can capture more intricate spatiotemporal patterns.

The model's first layer is a ConvLSTM2D layer, which applies a 3x3 convolutional filter with four output filters and a tanh activation function on input image sequences having the form (sequence length, image image_height, image_width, 3). The initial ConvLSTM2D layer's output is routed through a 3D max pooling layer with a pool size of (1, 2, 2) to minimize the output's spatial dimensions while maintaining the same number of filters. A Time Distributed layer that applies dropout regularization at a rate of 0.2 to each time step of the sequence is applied after the output of the max pooling layer. Overfitting is prevented by the Dropout layer by setting a portion of the input units to zero at each update during training. The model is extended with two more ConvLSTM2D layers, each followed by a max pooling layer and a Time Distributed dropout layer and a progressively larger number of filters. In order to transform the output of the preceding layer into a 1D tensor, a Flatten layer is utilized last. Following this, a fully connected layer called the Dense layer with softmax activation is used to forecast the class probabilities for each input sequence.

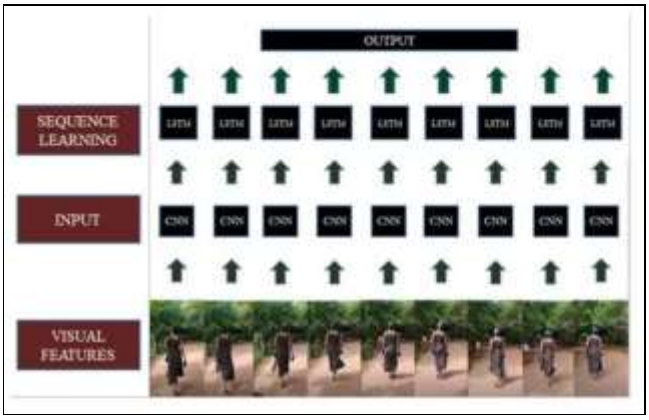
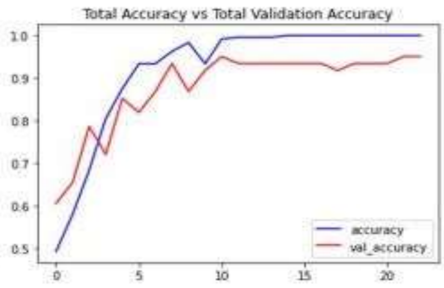


Fig 8: Convolutional LSTM

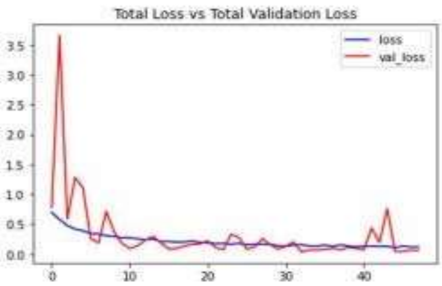
Results

During experimentation, we evaluated both the models on the two datasets that we described above: UCF50 and our own dataset. Both the datasets were used individually, and were not combined for experimentation. 80% of the dataset was used for the training stage and 20% of the dataset was used for testing, in both the cases. While fitting the model, the validation split parameter was set to 0.2, hence 20% of the training data was used for validation. The four cases that we have explored are Single Frame CNN with UCF50 dataset, Single Frame CNN with our own dataset, Convolutional LSTM with UCF50 dataset and Convolutional LSTM with our own dataset. For all the four cases, we have plotted the total loss vs. total validation loss graph and total accuracy vs. total validation accuracy graphs, in Fig.7 Both these plots can be used to assess the performance of the model during training. The x-axis shows the number of epochs, while the yaxis shows the value of the loss function or the value of the accuracy metric. For all the four cases, we have recorded the testing accuracy which is given in Table 1. Based on the results, we can see that the single frame CNN model outperforms that Convolutional LSTM model. We can also observe that the accuracy obtained is more with UCF50 dataset than with our own dataset. This is understandable since the dataset we created is small, hence the training size is less. The highest accuracy obtained is for the single frame CNN model with UCF50 dataset, which is 99.8%. We have used a heatmap to represent the confusion matrix for the four cases, for demonstrating the correspondence between the predicted labels, along the x-axis, and the true labels, along the y-axis, and to represent the recognition performance for each action class that was selected. A confusion matrix generally includes 4 groupings: True Positive, which denotes the instances that were correctly identified as positives, False Positive, which denotes the negative examples incorrectly identified as positives, True Negative, which denotes the negative examples that are correctly predicted as negatives and finally False Negative, which denote the positive instances incorrectly predicted as negative. In our case, the diagonal of the confusion matrix or heatmap represents the activities that are correctly recognized. The other cells represent activities that were predicted as some other activity, for example, if a video of ‘Jumping’ was predicted as ‘Walking’. It is to be noted here that, while we used our own dataset for training, we included the whole dataset during the training process since the size of the dataset is small and the dataset includes only 3 activities. However, while training with the UCF50 dataset, we selected 3 activities, namely ‘PullUps’, ‘WalkingWithDog’ and ‘PlayingGuitar’ from the 50 activities in the dataset for training. This was done due to the resource and time constraints, and to reduce the speed of training the model, since the overall size of UCF50 dataset is quite large. In the heatmaps representing either model’s performance on UCF50 dataset, 0 denotes the activity ‘PullUps’, 1 denotes ‘WalkingWithDog’, and 2 denotes ‘PlayingGuitar’ on the x-axis and y-axis. In the heatmaps representing either model’s performance on our own dataset, 0 denotes the activity ‘Jumping’, 1 denotes ‘Walking’, and 2 denotes ‘Sitting’ on the x-axis and y-axis. By analyzing the heatmaps, we can observe that except for the heatmap of the convolutional LSTM model on our own dataset, all others are showing a really good performance. The heatmap of convolutional LSTM on our own dataset shows that 83.33% of the time, the activity ‘Jumping’ is mistaken for ‘Walking’.

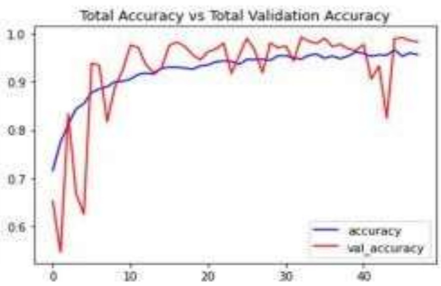
Total Loss vs. Total Validation Loss plot – Convolutional LSTM on UCF dataset



Total Accuracy vs. Total Accuracy Loss plot – Convolutional LSTM on UCF dataset

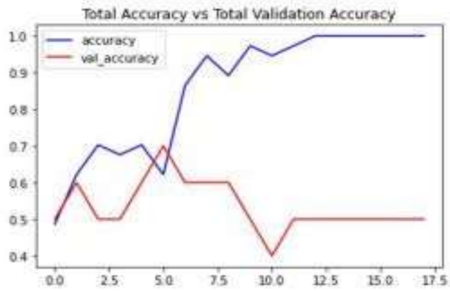


Total Loss vs. Total Validation Loss plot – Single Frame CNN on own dataset



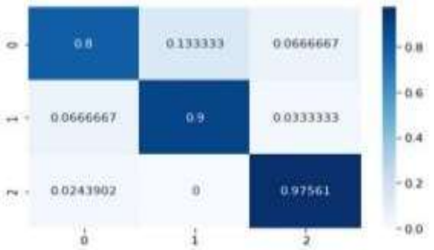
Total Accuracy vs. Total Accuracy Loss plot – Single Frame CNN on own dataset

Total Loss vs. Total Validation Loss plot – Convolutional LSTM on own dataset

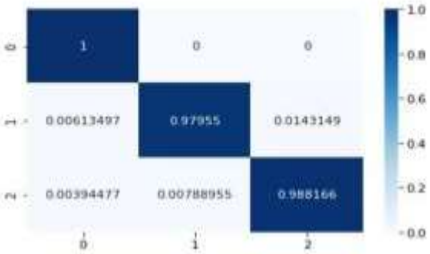


Total Accuracy vs. Total Accuracy Loss plot – Convolutional LSTM on own dataset

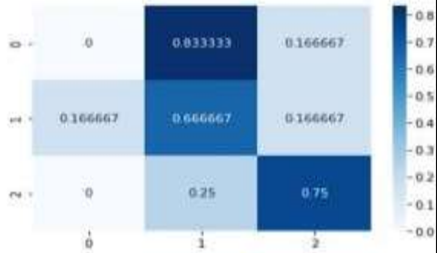
Heatmap of confusion matrix – Single Frame CNN with UCF50 dataset



Heatmap of confusion matrix – Convolutional LSTM with UCF50 dataset



Heatmap of confusion matrix – Single Frame CNN with own dataset



Heatmap of confusion matrix – Convolutional LSTM with own dataset

Heat Map

Conclusion

Human activity recognition using single frame CNN and Convolutional LSTM models present a promising solution for prediction of an action in a video the model has not seen before. Traditional CNNs and traditional LSTM networks both have proven to be efficient methods in solving various computer vision tasks. CNN effectively works on image data. LSTM networks on the other hand work well on sequence data. Combining the benefits of both the models makes Convolutional LSTM perfect for video classification, in our case, human activity recognition. Single Frame CNN as well works well for our use case, as CNN can extract features automatically from images. [32] In this study, the two models discussed were trained and tested on UCF50 dataset and a dataset that we created for the purpose of this experimentation. Both the models achieved good recognition performance, however, the single frame CNN model exhibited notably better accuracy than the convolutional LSTM model during testing. Also, the accuracy when testing on the UCF50 dataset was higher for both models when compared to testing on our own dataset. It is understandable as the size of our own dataset is quite small when compared to the size of the UCF50 dataset. For our future works, we would like to augment our newly created dataset to increase the training size and to include more videos of people from different ethnicities. Likewise we would like to expand the proposed models for a larger dataset such as Kinetics 700 because the models can be more effective when applied to a bigger dataset. Further we would like to explore the possibilities of implementing the models in a home monitoring system, by capturing video of activities performed by individuals alone at home in a camera and recognizing the activities done. If dangerous activities such as ‘falling’ are witnessed, an alert can be sent as an SMS to their caretakers’ phones.

References

1. Javan Roshtkhari M, Levine MD. Human activity recognition in videos using a single example. Image and Vision Computing. 2013 Nov;31(11):864-876. DOI: 10.1016/j.imavis.2013.08.005.
2. Marinho LB, de Souza Junior AH, Reboucas Filho PP. A New Approach to Human Activity Recognition Using Machine Learning Techniques. AISC. 2017 Feb;557.
3. Robertson N, Reid I. A general method for human activity recognition in video. Sep 2006.
4. Wang J, Zhang Z, Bin L, Lee S. An Enhanced Fall Detection System for Elderly Person Monitoring using Consumer Home Networks. IEEE Transactions on Consumer Electronics. 2014 Feb;60(1):23-29. DOI: 10.1109/TCE.2014.6780921.
5. Sharma, S., Tyagi, A., Kumar, S., & Kaushik, P. (2022). Additive manufacturing process based EOQ model under the effect of pandemic COVID-19 on non-instantaneous deteriorating items with price dependent demand. In A. Editor & B. Editor (Eds.), Additive Manufacturing in Industry 4.0 (1st ed.). CRC Press.
6. Balamurugan A, Krishna MV, Bhattacharya R, Mohammed S, Haralayya B, Kaushik P. Robotic Process Automation (RPA) in Accounting and Auditing of Business and Financial Information. The British Journal of Administrative Management. 2022;58(157):127-142.
7. Beddiar DR, Nini B, Sabokrou M, Hadid A. Vision-based human activity recognition: a survey. 2020 Aug;79:30509-30555.
8. Ke SR, Thuc HLU, Lee YJ, Hwang JN, Yoo JH, Choi KH. A Review on Video-Based Human Activity Recognition. Computers. 2013 Jun;2(2):88-131. DOI: 10.3390/computers2020088.
9. Mehr HD, Polat H. Human Activity Recognition in Smart Home with Deep Learning Approach. Aug 2019.
10. Chen L, Liu X, Peng L, Wu M. Deep learning based multimodal complex human activity recognition using wearable devices. 2020 Nov;51:4029-4042.
11. Ann OC, Lau BT. Human activity recognition: A review. Proceedings - 4th IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2014. Mar 2015. DOI: 10.1109/ICCSCE.2014.7072750.
12. Chopra Y, Kaushik P, Rathore SPS, Kaur P. Uncovering Semantic Inconsistencies and Deceptive Language in False News Using Deep Learning and NLP Techniques for Effective Management. International Journal on Recent and Innovation Trends in Computing and Communication. 2023;11(8s):681-692. DOI: 10.17762/ijritcc.v11i8s.7256.
13. Kaushik P. Role and Application of Artificial Intelligence in Business Analytics: A Critical Evaluation. International Journal for Global Academic & Scientific Research. 2022;1(3):01-11. DOI: 10.55938/ijgasr.v1i3.15.
14. Kaushik P. Deep Learning Unveils Hidden Insights: Advancing Brain Tumor Diagnosis. International Journal for Global Academic & Scientific Research. 2023;2(2):01-22. DOI: 10.55938/ijgasr.v2i2.45.
15. Kaushik P. Unleashing the Power of Multi-Agent Deep Learning: Cyber-Attack Detection in IoT. International Journal for Global Academic & Scientific Research. 2023;2(2):23-45. DOI: 10.55938/ijgasr.v2i2.46.
16. Kaushik P, Rathore SPS. Deep Learning Multi-Agent Model for Phishing Cyber-attack Detection. International Journal on Recent and Innovation Trends in Computing and Communication. 2023;11(9s):680-686. DOI: 10.17762/ijritcc.v11i9s.7674.
17. Kaushik P, Miglani S, Shandilya I, Singh A, Saini D, Singh A. HR Functions Productivity Boost by using AI. International Journal on Recent and Innovation Trends in Computing and Communication. 2023;11(8s):701-713. DOI: 10.17762/ijritcc.v11i8s.7672.
18. Kaushik P, Rathore SPS, Kaur P, Kumar H, Tyagi N. Leveraging Multiscale Adaptive Object Detection and Contrastive Feature Learning for Customer Behavior Analysis in Retail Settings. International Journal on Recent and Innovation Trends in Computing and Communication. 2023;11(6s):326-343. DOI: 10.17762/ijritcc.v11i6s.6938.
19. Kaushik P, Yadav R. Reliability design protocol and block chain locating technique for mobile agent. Journal of Advances in Science and Technology (JAST). 2017;14(1):136-141. DOI: 10.29070/JAST.
20. Kaushik P, Yadav R. Deployment of Location Management Protocol and Fault Tolerant Technique for Mobile Agents. Journal of Advances and Scholarly Researches in Allied Education [JASRAE]. 2018;15(6):590-595. DOI: 10.29070/JASRAE.
21. Kaushik P, Yadav R. Mobile Image Vision and Image Processing Reliability Design for Fault-Free Tolerance in Traffic Jam. Journal of Advances and Scholarly Researches in Allied Education (JASRAE). 2018;15(6):606-611. DOI: 10.29070/JASRAE.

22 Kaushik P, Yadav R. Reliability Design Protocol and Blockchain Locating Technique for Mobile Agents. Journal of Advances and Scholarly Researches in Allied Education [JASRAE]. 2018;15(6):590-595. DOI: 10.29070/JASRAE.
23 Kaushik P, Yadav R. Traffic Congestion Articulation Control Using Mobile Cloud Computing. Journal of Advances and Scholarly Researches in Allied Education (JASRAE). 2018;15(1):1439-1442. DOI: 10.29070/JASRAE.
24 Rathore SPS. Analysing the efficacy of training strategies in enhancing productivity and advancement in profession: theoretical analysis in Indian context. International Journal for Global Academic & Scientific Research. 2023;2(2):56-77. DOI: 10.55938/ijgasr.v2i2.49.
25 Rathore SPS. The Impact of AI on Recruitment and Selection Processes: Analysing the role of AI in automating and enhancing recruitment and selection procedures. International Journal for Global Academic & Scientific Research. 2023;2(2):78-93. DOI: 10.55938/ijgasr.v2i2.50.
26 Rathore R. Application of Assignment Problem and Traffic Intensity in Minimization of Traffic Congestion. IJRST. 2021 Jul-Sep;11(3):25-34. DOI: http://doi.org/10.37648/ijrst.v11i03.003.
27 Rathore R. A Review on Study of application of queueing models in Hospital sector. International Journal for Global Academic & Scientific Research. 2022;1(2):01-05. DOI: 10.55938/ijgasr.v1i2.11.
28 Rathore R. A Study on Application of Stochastic Queuing Models for Control of Congestion and Crowding. International Journal for Global Academic & Scientific Research. 2022;1(1):01-07. DOI: 10.55938/ijgasr.v1i1.6.
29 Rathore R. A Study of Bed Occupancy Management in the Healthcare System Using The M/M/C Queue And Probability. International Journal for Global Academic & Scientific Research. 2023;2(1):01-09. DOI: 10.55938/ijgasr.v2i1.36.
30 Sharma T, Kaushik P. Leveraging Sentiment Analysis for Twitter Data to Uncover User Opinions and Emotions. International Journal on Recent and Innovation Trends in Computing and Communication. 2023;11(8s):162-169. DOI: 10.17762/ijritcc.v11i8s.7186.
31 Sharma V. A Study on Data Scaling Methods for Machine Learning. International Journal for Global Academic & Scientific Research. 2022;1(1):23-33. DOI: 10.55938/ijgasr.v1i1.4.
32 Yadav M, Kakkar M, Kaushik P. Harnessing Artificial Intelligence to Empower HR Processes and Drive Enhanced Efficiency in the Workplace to Boost Productivity. International Journal on Recent and Innovation Trends in Computing and Communication. 2023;11(8s):381-390. DOI: 10.17762/ijritcc.v11i8s.7218.