ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1195

Human Activity Recognition for Occupational Safety Management Using Deep Learning Models

Neetu Pal[1]

[1]Maya Devi University

Publication Date: 2026/06/30

Abstract: Recently, recognition of activities has gained popularity as a study topic in the ubiquitous subject of computer science. Recently, there has been a lot of interest in recognizing human actions in videos. Most PC vision functions, including human-computer interactions, suspicious activity detection, and medical activity detection, are predicated on the idea of activity detection. One of the best methods for working with activity data is to use deep learning technology and its various models. Activity detection is the process of automatically labelling activity from video frames. Deep networks have led to the separation of the activity detection process into two categories: the automated characteristic extraction approach and the classic characteristic-based approach. This research work, which is based on a conventional and automatic technique, focuses on several approaches employed in current literature. With the advent of repeating neural networks and convolutional neural networks, the entry modality's machine learning capability provides it the primary option to use for activity recognition. This paper examined several methods, considering their technique, datasets, accuracy, and classifiers. An outline of the HAR's importance, difficulties, and potential developments is provided in this study. Additionally, briefly discuss the HAR system using a variety of potential methods.

Keywords: *Deep Learning, HAR, LSTM, ConvLSTM, LRCN, Convolutional.*

How to Cite: Neetu Pal (2026) Human Activity Recognition for Occupational Safety Management Using Deep Learning Models. *International Journal of Innovative Science and Research Technology*, 11(6), 1714-1722. https://doi.org/10.38124/ijisrt/26jun1195

I. INTRODUCTION

To recognize, identify, and categorize human behavior, several programs with human-focused tracking were developed, and researchers have offered numerous methods. Popularity among people is a crucial indicator of a person's dynamism, and it may be measured with the use of deep learning algorithms. This processing can be done in real time using a new technique that combines time-delay embedding and supervised learning for feature extraction [1]. In the smart city age, video surveillance has become a vital necessity to improve the quality of life and establish the neighborhood as a safe place. By looking at how context-aware application areas for data distribution and activity restructuring might be found using new machine learning approaches [2]. To ensure strong insurance of a surrounding area, security cameras are typically positioned at a specific distance. More video analysis and understanding are therefore essential, with significant ramifications for the safety equipment [3]. Healthcare, transportation, manufacturing, schools, malls, and markets can all benefit from a visual information-pushed device. Here are a few domain name examples where popularity could be highly desired in human hobbies. An alert to the manipulation room is typically generated by abnormal activity in a system that is mostly based on HAR. Instead of sitting in front of the digital cam feed and watching what is happening, it is crucial in these situations to understand the good aspects of the stated matters.

The main goal of human hobby popularity is to characterize human interactions and behaviours mostly using previously unidentified information sequences. It is frequently challenging to identify human sports from video statistics due to issues with background conversion and unappealing movies. Two fundamental questions emerge among distinct human hobby identity systems: "Where precisely is it inside the video?" is the localization task, and "Which motion is performed?" is the motion popularity task. Image sequences are called frames. Therefore, processing incoming movies that enable you to realize the following human sports is the primary goal of a movement reputation project. The authors proposed an RBM-based activity that uses the Snapdragon to identify models using smartwatch classes. [3] [4] and wristwatches, instruments, or smartphones equipped with gyroscopes and accelerometers.

- *Brief Overview of Har*

Human pastime recognition, or HAR for short, is a broad field of study concerned with identifying a person's character or motion based solely on sensor data. In indoor sports, walking, talking, standing, and sitting are all commonplace. These could also be more specialized sports, such as those played on the floor of a manufacturing plant or

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1195

in a restaurant [4]. It is possible to remotely record sensor statistics using various Wi-Fi methods or video. Additionally, one can instantly gather information about the issue by donning customized clothing. by using several IoT platform levels according to various state-of-the-art technologies.[5] [6]

➢ *Types of HAR:*

The HAR is divided into two groups, mostly according to the equipment's.

- *Vision-Based HAR*

Numerous areas have static cameras installed for surveillance purposes, which record the videos and store them on servers. These captured videos or digital camera feeds are then used for tracking. For instance, the use of a single static digital camera in video surveillance programs has made human posture popular [7]. This type of HAR is utilized for crowd monitoring, site visitor control, public safety, and street safety, among other purposes. Figure 1 depicts the daily phases of a completely human pastime popularity system, primarily based on eyesight.

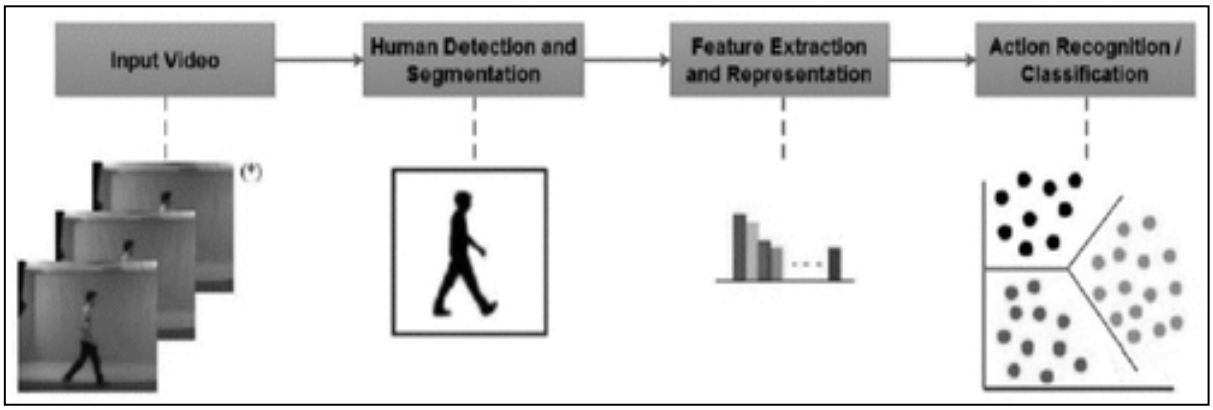

Fig 1 A Typical System for Recognizing Human Activities.

- *Sensor-Based HAR*

Recently, smartphones have developed into a tool for international communication as well as a means of understanding people. Sensors on smartphones can continuously record information about human movements. Research determined that people were interested in using smartphone sensors. Using this technique, data is gathered from the phone's gyroscope and accelerometer sensors, then gadget analysis techniques are applied to distinguish between human hobbies [8]. This type of HAR works well for systems that track impacted individuals, track the interests of male and female athletes, and so on. However, it is not applicable to the broader application of human pastime reputation for security, tracking, and other purposes [9].

The studies in human sports popularity may be classified, in line with the character of the statistics entered, into subclasses:

According to the nature of the entered statistics, the research on human sports popularity can be divided into the following subclasses:

The reputation of human hobbies is mostly based on still images: the device can recognize actions from pictures, such as eating, walking, sitting, and so on. The features of the interest distinguish it from others [10]. Our paintings aim to enforce deep learning algorithms on real-world global datasets so that one may examine their accuracy and draw useful conclusions. Human hobby popularity, or HAR, is a challenging time collection task [11]. To properly create functions from raw data when building a deep mastery model, it usually requires deep area comprehension and sign processing approaches to predict a person's motion based only on sensor statistics. Convolutional neural networks and recurrent neural networks are two examples of deep learning techniques that have recently been used to verify their performance or even create cutting-edge effects by analysing characteristics from raw sensor data[12].

II. LITERATURE SURVEY

Using technique a deep learning, this section offers a concise synopsis of the body of existing HAR literature. Comprehensive supervised and unsupervised models are used to construct HAR. Fig. 2 displays the classifications. This figure has been depicting an altered version of the Inception Time network architecture. The created model was validated by using publicly available datasets, including StanWiFi, SignFi, and ARIL. The proposed CSI Time has achieved 98.20 percent, 98 percent, and 95.42 percent accuracy for WiFi-based activity recognition for the ARIL, Stan WiFi, and Sign Fi datasets, respectively [13]. They developed a edge devices using lightweight deep learning model for HAR activity to use less power. The suggested model's algorithm performance is evaluated using data from the participant's six daily activities. [14]

Fast Classification of EEG Artifacts (FCEA) is a deep learning model that is used to classify EEG artifacts based on an individual's physiologic activity. [15] The suggested approach classifies human behaviour by utilizing long-term

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1195

memory and the best characteristics of a network of convolutional neurons. Clenching of Jaw and head and eye movements are challenging to pick up on with sensory technology that is commonly employed to identify human activities. In terms of the F1 score, the suggested model performs best [16]. The Raspberry Pi3 may use the light deep learning model for human activity recognition, which requires fewer computing resources. Data from each of the participant's six daily activities is used to assess the performance of the proposed model. The suggested model outperforms many of the most advanced deep learning methods currently in use [17]. The creation of a thorough, in-depth learning system aims to identify human-to-human communication over Wi-Fi signals. A well-known CSI data set that was gathered from 40 different patient couples during 13 human-to-human contacts was used to evaluate this model. The average accuracy of the developed model was 86.3% overall [18]. An algorithm for HAR visualization. Binary movement images are used in the deep learning model that has been constructed. Both 2-D and 3-D datasets can be used with visual models. This model uses subsampling layers that incorporate translations, rotations, and minor movement changes [19]. To detect human activity, smartphone sensors and core kernel components (KPCA) are analyzed. The attributes are initially extracted by the proposed model, and then they are processed using KPCA and linear discriminant analysis. For efficient recognition, a Deep Belief Network (DBN) is used to create the recommended template. Artificial neural networks (ANTs) and traditional machine learning models are less accurate than the KPCA [20]. A deep learning model for the HAR that is unsupervised. The proposed model makes use of the encoding architecture to minimize reconstruction mistakes. The big dataset and real-time environment are not suitable for the model [21]. Use the LSTM template to classify movies using the first ResNet and transfer learning technique for HAR. UCI 101 and HMDB 51 are the two distinct datasets that were used to train the model. The highest accuracy scores, 92 percent and 91 percent, respectively, are offered by ResNet v2 [22]. The authors used deep learning to introduce multi-frequency in different domain ATMs based on progressive continuous wave radars. To extract features, it makes use of automated encoders and deep convolutional neural networks (DCNN). DCNN's primary purpose is to recover a spectrogram's micro-Doppler properties. On the other hand, learning range distribution features is the primary goal of auto-encoding. The recognition accuracy of the suggested in-depth learning model is 96.42% [23]. Currently, quality-oriented learning is being reinforced profoundly to recognize robot activities. To achieve machine learning through manipulation, it employs a policy research paradigm. For MDR, the recommended approach offers good precision [24]. It combined many convolutional neural networks to create MDT. Numerous CNN models and assemblies were created and evaluated. Compared to previous models, the overall model provided provides greater precision. This method's primary advantage is its ability to extract the traits required for the model. By avoiding the pretreatment phase, this model reduces the amount of time needed for testing and training [25]. The RSA's acquisition of occasional feature films. While maintaining the accuracy of the current classes, the deep learning model's insufficient achievement allows it to incorporate a larger number of non-rotating classes, resulting in a significant rise in model size. Because it makes use of the Fully Convolution Organization–Long Short-term Memory (FCN-LSTM), this model is also lighter than the best in its class [26]. When a patient with mental or solid skeletal issues needs assistance with self-testing, this motion recognition template can be used as a web motion observer The suggested approach offers a great location change in HAR; nevertheless, it necessitates a review of the changes between the final video highlights rather than the group characterisation like unchangeable. The edge is debatably altered for the success metric, meaning it responds to a different movement.

III. METHODOLOGY

➢ *Data Preparation*:

This section outlines the entire data preprocessing procedure in detail:

- *Use Labels to Visualize the Data:*

Show a few selected movies from each dataset class in this phase. This will help us get a clear picture of the dataset's appearance.

- *Read and Preprocess the Dataset:*

In this stage, write a function that will take frames out of every video and do pre-processing tasks like normalizing and resizing images. The input for this method is the path to a video file. After that, the software reads the video file frame by frame, resizes each frame, appends the normalized frame to a list, normalizes the resized frame, and then returns the list. Noise removal is the process of eliminating noise from a signal. Information will be lost when noise is added. Taking pictures in poor light is one of the many things that can produce noise. Different filters are needed for different kinds of noise. Therefore, the first step in denoising an image using conventional filters is to determine what kind of noise it contains. Once this has been established, apply the appropriate filter. Convolutional neural networks function well in this project to eliminate noise from photos.

- *Dataset Creation (Features and Labels):*

This stage makes use of the frame extraction() function to produce the final pre-processed dataset by creating a new function named construct dataset. Feature selection is used to select the most pertinent features from the data. By choosing only the pertinent elements of the data, the classification system can achieve higher predicted accuracy and a lower computing load. The way this function operates is as follows:

- ✓ Iterate through each of the six action classes found in the PSRA6 dataset, which are listed in the classes list.
- ✓ At this point, go through every video file in every class.
- ✓ Apply the frame extraction technique to every video file.
- ✓ Include the returned frames in a list named "temp features."
- ✓ After processing every video in a class, add video frames to the features list at random (equivalent to the maximum number of photos per class).

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1195

- ✓ After processing all films from all classes, return the features and labels as NumPy arrays.
- ✓ A list of feature vectors and the labels corresponding to them will be returned when this function is invoked.
- ✓ Next, create one-hot encoded vectors from class labels using the one-hot encoding technique.

- *Split the Dataset into Train and Test Sets:*

Two NumPy arrays are split in this phase, one with all features and the other with just labels. The ratio is 0.8 for model training and 0.2 for model evaluation. Data must first be jumbled, which has already been done, before it can be separated. The effectiveness of algorithms for machine learning, appropriate for prediction-based algorithms is estimated using the train-test split. This quick and easy process allows one to compare the outcomes of their own machine learning model with machine results from the dataset.

➢ *Construct the Model*

The entire model construction procedure is covered in this section. The authors of this work created a basic CNN classification model using two CNN layers.

- A feature for creating the model
- Using a sequential model to generate the model
- Definition of the model architecture: In Section 3, a prototype design is already explained.
- Use the model's Conclusion() function to print the models' summary. The function Summary(). Section 3 already provides an explanation of one model overview. To create the model, use the create_model() function.
- Make a model plot. The plot_model() function is used to plot the model, as explained in Section 3.2.2.

➢ *Training of Model*

The PSRA6 algorithm training is provided on the dataset for generate enough epochs after that the deep learning model (2-layer CNN) is chosen. Following training, a weight file (.h5 file) is used to classify six jobs from the PSRA6 dataset. The model is then evaluated using a camera clip as input, predicting activity in the video, and generating a labelled activity video and a text file for activity labels as outputs, as seen in Figure 1. Before the model is trained, the training parameters and hyperparameters, which are shown in Table 1 and Table 2 for the authors' scenario, must be set in accordance with the relevant framework, the configuration model's parameters are inherent to the model. Hyperparameters are specifically defined parameters used to regulate the training process. It consist of amount of neurons layer, the number of training iterations, etc., must be used to make predictions. Model optimization necessitates the use of hyperparameters such as learning rate, epoch count, train-test split, optimizer, activation function, and batch size.

Using the fit() method, we will train our model with the following parameters: the number of epochs, validation data, target data (train Y), and training data (train X). We will use the test set, which we have divided into X and Y tests, as validation data. It depends upon the number of run on the data is indicated by the number of epochs. The more epochs we run, the better the model gets, up to a point. The model will no longer get better with each new epoch after that. The number of epochs in our model will be set to 100.

- *Existing Models*

The following are some examples of classification algorithms that are used to group speech signals based on emotions:

✓ *ConvLSTM*

It is a recurrent layer, just like the LSTM, but instead of underlying matrix multiplications, convolution operations are employed. Because of this, the data that goes through the ConvLSTM cells keeps the dimension of input (3D in our example), instead of just being a 1D vector with features. An alternative to a ConvLSTM is a Convolutional-LSTM model, in which the picture is processed through the convolution layers and the acquired features are flattened into a 1D array. When this process is performed for each image in the time set, it produces a set of characteristics across time that serve as the input for the LSTM layer.

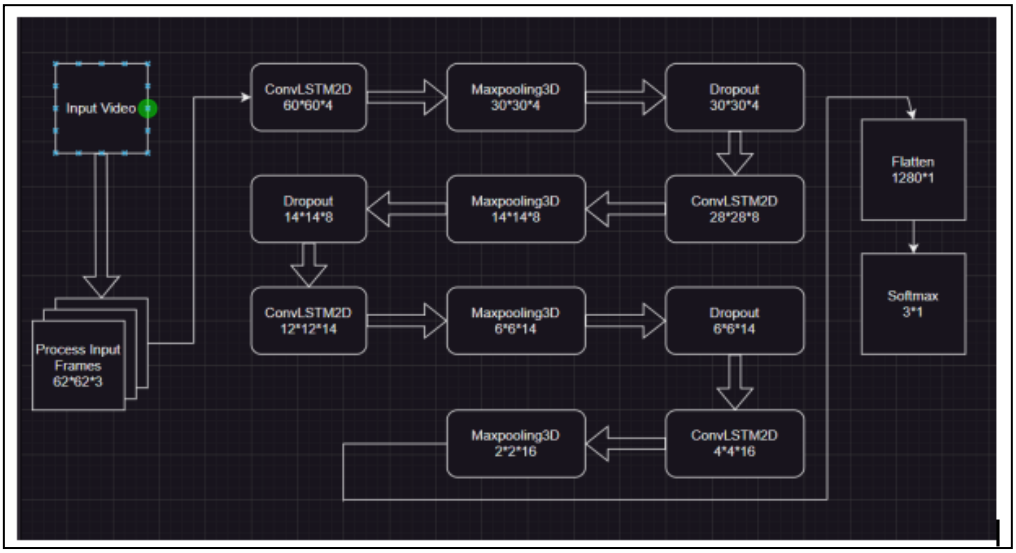

Fig 2 Architecture of ConvLSTM

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1195

✓ *CNN+LSTM (LRCN)*

CNN layers that extract features from input data and LSTM layers that provide sequence prediction are combined to create a CNN+LSTM model. Typically, the CNN+LSTM is employed for image labelling, video labelling, and activity identification. Creating textual annotations from image sequences and applying them to visual time series predictions are two of their commonalities.

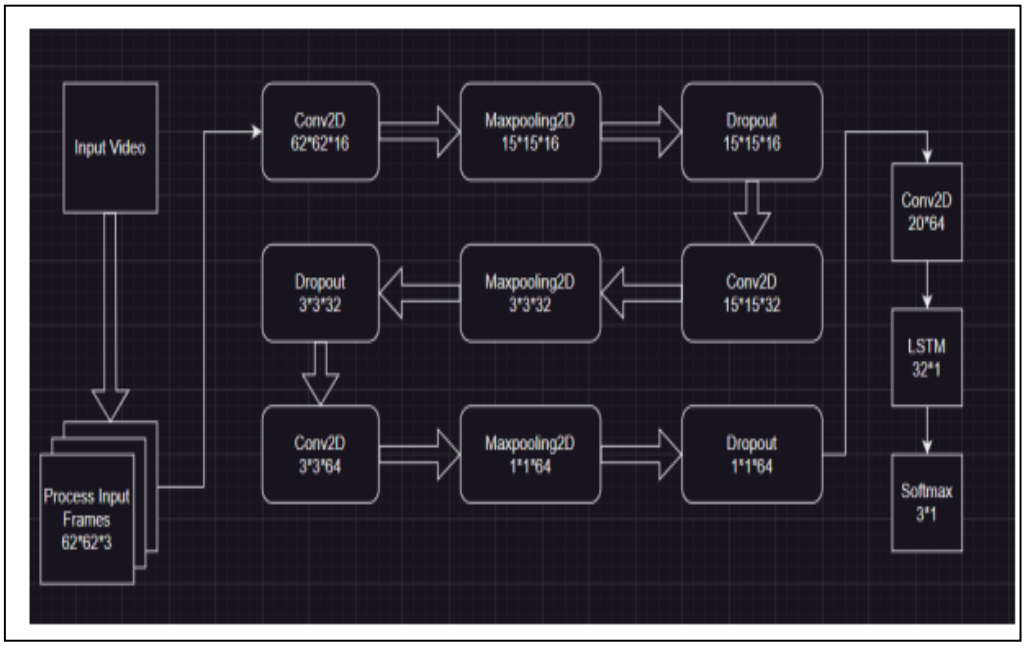

Fig 3 Architecture of LRCN

- *Proposed Model*

The suggested approach is a hybrid CNN+LSTM model that provides more accuracy than the current CNN, LSTM, and MLP models.

✓ *CNN+BiLSTM*

"Two-dimensional Convolutional Neural Network with Bidirectional Long Short-Term Memory (2D-CNN-BiLSTM)" is the name of the hybrid deep learning model that is proposed in this paper. CNNs are known to be the best models for a variety of computer vision problems. This is since CNNs can be built with many hidden layers and have many hyperparameter [7]. These characteristics allow CNNs to learn the internal representation of many signal dimensions, including images and videos, for feature learning. In situations like time series, the uniform method can also be used to utilize 2D signal data. However, even though 2D-CNN's performance in feature learning is already encouraging, its adaptability can be further investigated by incorporating other neural networks to achieve innovation in terms of speed and competence; this is where the Long Short-Term Memory (LSTM) is useful.

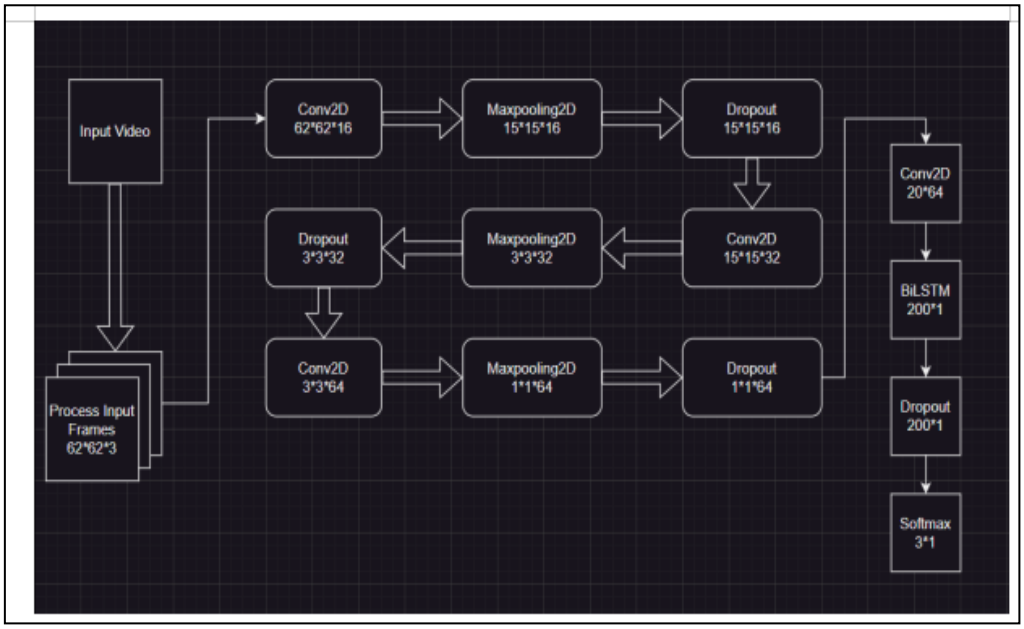

Fig 4 Architecture of CNN+BiLSTM

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1195

✓ *CNN+GRU*

The Gated Recurrent Unit (GRU) receives the feature vector from the convolutional layer. 2014 saw the introduction of a new gating mechanism called the Gated Recurrent Unit, which is a more recent RNN generation. Like LSTM, GRU has demonstrated superior performance on smaller datasets. Because GRU has a gated recurrent neural network structure, it is a variation of LSTM. In contrast to LSTM, which has three gates (forgetting, input, and output) and two gates (update and reset), GRU has fewer training parameters, which allows it to converge more quickly during training. GRUs have liberated themselves from the cell state and are now transferring information in the hidden state. Like the input and forget gates of an LSTM, the update gate determines what data should be added, retained, and released. The reset gate determines how much of the past data should be released (27)(28).

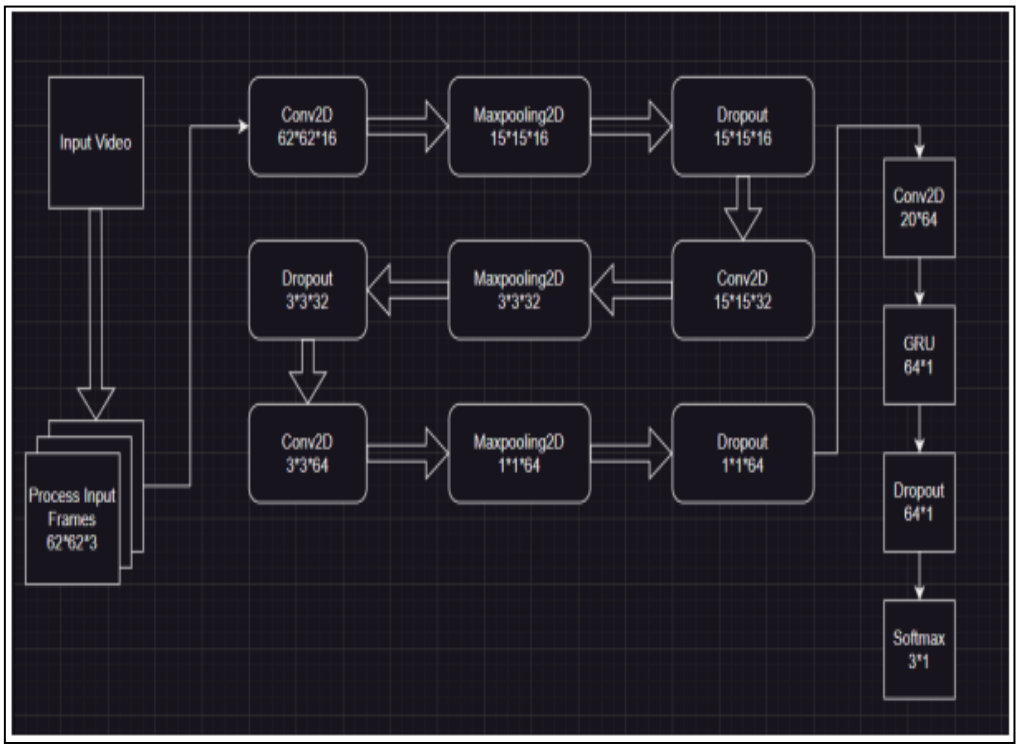

Fig 5 CNN+GRU Architecture

➢ *Evaluation Metrics*

After building and training our deep learning model, we need to assess its performance. The Table 3 shows comparative analysis the result of various algorithms. The SER assessment metrics are categorized using four distinct criteria: precision, recall, accuracy, and F1-score [5]. To achieve this, we assess our suggested model using these metrics. The definition of precision is.

$$Precision = \frac{TP}{(TP+FP)} \quad (1)$$

Additionally, the recall is described as:

$$Recall = \frac{TP}{(TP+FN)} \quad (2)$$

Where TP, FN, and FP represent the dataset's true positive, false negative, and false positive counts, respectively.

The F1-score is the harmonic mean of these two indicators (recall and precision), both of which must be high [7]. It is defined as.

$$F1\ Score = \frac{2*Recall*Precision}{Recall+Precision} \quad (3)$$

The accuracy is defined as the ratio of true outcomes (both true positive and true negative) to the total number of cases examined [9]. It is defined by:

$$Accuracy = \frac{TP+TN}{Total\ Population} \quad (4)$$

IV. EXPERIMENTAL RESULTS AND DISCUSSION

After building and training our deep learning model, we need to evaluate its performance. To classify the SER assessment metrics, four distinct criteria are used: precision, recall, accuracy, and F1-score [5]. To assess our suggested model, we employ these indicators. This is the definition of precision:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1195

➢ *Model Summary*

Table 1 Model Summary of ConvLSTM and LRCN.

ConvLSTM

Layer (type)	Output Shape	Param #
conv_lstm2d_8 (ConvLSTM2D)	(None, 20, 60, 60, 4)	1024
max_pooling3d_8 (MaxPooling3D)	(None, 20, 30, 30, 4)	0
time_distributed_6 (TimeDistributed)	(None, 20, 30, 30, 4)	0
conv_lstm2d_9 (ConvLSTM2D)	(None, 20, 28, 28, 8)	3488
max_pooling3d_9 (MaxPooling3D)	(None, 20, 14, 14, 8)	0
time_distributed_7 (TimeDistributed)	(None, 20, 14, 14, 8)	0
conv_lstm2d_10 (ConvLSTM2D)	(None, 20, 12, 12, 14)	11144
max_pooling3d_10 (MaxPooling3D)	(None, 20, 6, 6, 14)	0
time_distributed_8 (TimeDistributed)	(None, 20, 6, 6, 14)	0
conv_lstm2d_11 (ConvLSTM2D)	(None, 20, 4, 4, 16)	17344
max_pooling3d_11 (MaxPooling3D)	(None, 20, 2, 2, 16)	0
flatten_2 (Flatten)	(None, 1280)	0
dense_2 (Dense)	(None, 3)	3843

Total params: 36,843
Trainable params: 36,843
Non-trainable params: 0

Model Created Successfully!

LRCN

Layer (type)	Output Shape	Param #
time_distributed_9 (TimeDistributed)	(None, 20, 62, 62, 16)	448
time_distributed_10 (TimeDistributed)	(None, 20, 15, 15, 16)	0
time_distributed_11 (TimeDistributed)	(None, 20, 15, 15, 16)	0
time_distributed_12 (TimeDistributed)	(None, 20, 15, 15, 32)	4640
time_distributed_13 (TimeDistributed)	(None, 20, 3, 3, 32)	0
time_distributed_14 (TimeDistributed)	(None, 20, 3, 3, 32)	0
time_distributed_15 (TimeDistributed)	(None, 20, 3, 3, 64)	18496
time_distributed_16 (TimeDistributed)	(None, 20, 1, 1, 64)	0
time_distributed_17 (TimeDistributed)	(None, 20, 1, 1, 64)	0
time_distributed_18 (TimeDistributed)	(None, 20, 1, 1, 64)	36928
time_distributed_19 (TimeDistributed)	(None, 20, 64)	0
lstm (LSTM)	(None, 32)	12416
dense_3 (Dense)	(None, 3)	99

Total params: 73,027
Trainable params: 73,027
Non-trainable params: 0

Model Created Successfully!

Table 2 Model Summary of Conv Bi-LSTM and Conv+GRU.

CNN + BiLSTM

Layer (type)	Output Shape	Param #
time_distributed_42 (TimeDistributed)	(None, 20, 62, 62, 16)	448
time_distributed_43 (TimeDistributed)	(None, 20, 15, 15, 16)	0
time_distributed_44 (TimeDistributed)	(None, 20, 15, 15, 16)	0
time_distributed_45 (TimeDistributed)	(None, 20, 15, 15, 32)	4640
time_distributed_46 (TimeDistributed)	(None, 20, 3, 3, 32)	0
time_distributed_47 (TimeDistributed)	(None, 20, 3, 3, 32)	0
time_distributed_48 (TimeDistributed)	(None, 20, 3, 3, 64)	18496
time_distributed_49 (TimeDistributed)	(None, 20, 1, 1, 64)	0
time_distributed_50 (TimeDistributed)	(None, 20, 1, 1, 64)	0
time_distributed_51 (TimeDistributed)	(None, 20, 1, 1, 64)	36928
time_distributed_52 (TimeDistributed)	(None, 20, 64)	0
bidirectional (Bidirectional)	(None, 200)	132000
dropout_22 (Dropout)	(None, 200)	0
dense_5 (Dense)	(None, 3)	603

Total params: 193,115
Trainable params: 193,115
Non-trainable params: 0

Model Created Successfully!

CNN + GRU

Layer (type)	Output Shape	Param #
time_distributed_20 (TimeDistributed)	(None, 20, 62, 62, 16)	448
time_distributed_21 (TimeDistributed)	(None, 20, 15, 15, 16)	0
time_distributed_22 (TimeDistributed)	(None, 20, 15, 15, 16)	0
time_distributed_23 (TimeDistributed)	(None, 20, 15, 15, 32)	4640
time_distributed_24 (TimeDistributed)	(None, 20, 3, 3, 32)	0
time_distributed_25 (TimeDistributed)	(None, 20, 3, 3, 32)	0
time_distributed_26 (TimeDistributed)	(None, 20, 3, 3, 64)	18496
time_distributed_27 (TimeDistributed)	(None, 20, 1, 1, 64)	0
time_distributed_28 (TimeDistributed)	(None, 20, 1, 1, 64)	0
time_distributed_29 (TimeDistributed)	(None, 20, 1, 1, 64)	36928
time_distributed_30 (TimeDistributed)	(None, 20, 64)	0
gru (GRU)	(None, 64)	24960
dropout_15 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 3)	195

Total params: 85,667
Trainable params: 85,667
Non-trainable params: 0

Model Created Successfully!

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1195

➢ *Comparison of ConvLSTM, LRCN, CNN+BiLSTM, CNN+GRU*

Table 3 Comparison of ConvLSTM, LRCN, CNN+BiLSTM, CNN+GRU

Model	Epochs	Accuracy	Validation Accuracy	Loss	Precision	Recall	F1-Score
1. ConvLSTM	50	0.9907	0.8000	0.8838	0.8139	0.7777	0.7954
2. LRCN	50	09833	0.8889	0.3153	0.8888	0.8888	0.8888
3.CNN+BiLSTM	50	0.9987	0.9333	0.2747	0.9333	0.9333	0.9333
4. CNN+GRU	50	0.9952	0.8778	0.3101	0.8777	0.8777	0.8777

➢ *Confusion Matrix*

Table 4 Confusion Matrix of various deep learning models

ConvLSTM	LRCN	CNN + BiLSTM	CNN + GRU
Confusion Matrix [[18 7 7] [0 23 0] [4 0 31]]	Confusion Matrix [[25 5 2] [0 23 0] [3 0 32]]	Confusion Matrix [[27 2 3] [0 23 0] [1 0 34]]	Confusion Matrix [[27 2 3] [3 20 0] [3 0 32]]

➢ *Comparative Analysis Charts of Models Loss, Models Precisions, Models F1 Score, Models Accuracy*

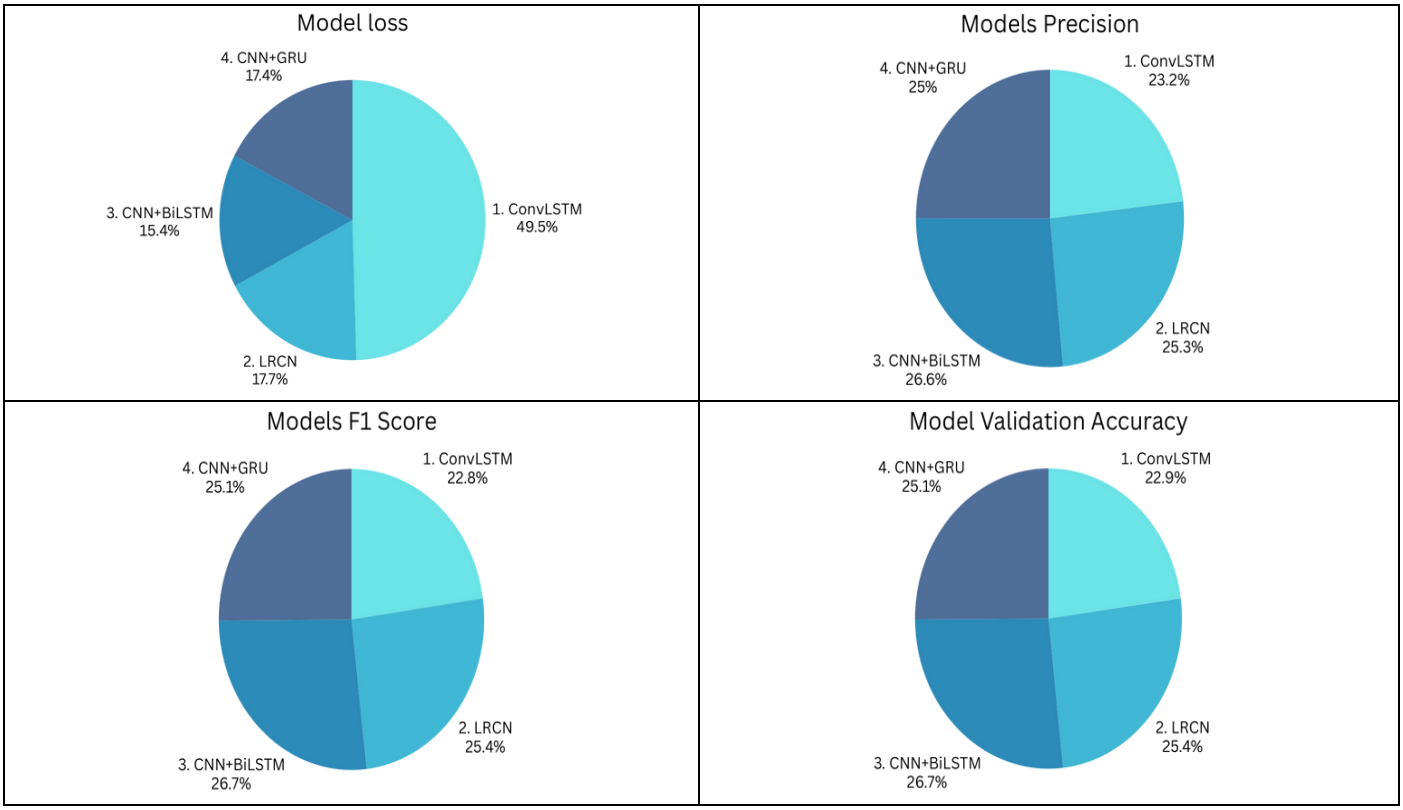

Fig 6 Comparative Analysis Charts of Models Loss, Models Precisions, Models F1 Score, Models Accuracy

V. CONCLUSION

The UCF50 dataset was used for this study, and the outcomes show how successful our suggested HAR approach is used. Specifically, our suggested hybrid model, CNN+BiLSTM, achieves an accuracy rate of 93.33%, while CNN+GRU achieves an accuracy rate of 87.78%. Additionally, the current model performs well, with an accuracy of 88.89% for LRCN and 80.00% for ConvLSTM. Lastly, with an accuracy of 93.33%, the CNN+BiLSTM model performs better than the other models. Our study's conclusions show that the suggested system can correctly classify human activities by utilizing characteristics and neural network models.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun1195

REFERENCES

[1]. Vrigkas, M., Nikou, C., & Kakadiaris, I. A. (2015). A review of human activity recognition methods. *Frontiers in Robotics and AI*, *2*, 28.

[2]. Bux, A., Angelov, P., & Habib, Z. (2017). Vision-based human activity recognition: a review. *Advances in computational intelligence systems*, 341-371.

[3]. Ramasamy Ramamurthy, S., & Roy, N. (2018). Recent trends in machine learning for human activity recognition—A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4), e1254.

[4]. Jobanputra, C., Bavishi, J., & Doshi, N. (2019). Human activity recognition: A survey. *Procedia Computer Science*, *155*, 698-703.

[5]. Dhiman, C., & Vishwakarma, D. K. (2019). A review of state-of-the-art techniques for abnormal human activity recognition. *Engineering Applications of Artificial Intelligence*, *77*, 21-45.

[6]. Hussain, Z., Sheng, M., & Zhang, W. E. (2019). This survey examines different approaches for human activity recognition. *arXiv preprint arXiv:1906.05074*.

[7]. Zhang, S., Wei, Z., Nie, J., Huang, L., Wang, S., & Li, Z. (2017). A review on human activity recognition using vision-based method. *Journal of healthcare engineering*, *2017*(1), 3090343.

[8]. Gu, F., Chung, M. H., Chignell, M., Valaee, S., Zhou, B., & Liu, X. (2021). A survey on deep learning for human activity recognition. *ACM Computing Surveys (CSUR)*, *54*(8), 1-34.

[9]. Gupta, N., Gupta, S. K., Pathak, R. K., Jain, V., Rashidi, P., & Suri, J. S. (2022). Human activity recognition in artificial intelligence framework: a narrative review. *Artificial intelligence review*, *55*(6), 4755-4808.

[10]. Islam, M. M., Nooruddin, S., Karray, F., & Muhammad, G. (2022). Human activity recognition using tools of convolutional neural networks: A state of the art review, data sets, challenges, and future prospects. *Computers in biology and medicine*, *149*, 106060.

[11]. Arshad, M. H., Bilal, M., & Gani, A. (2022). Human activity recognition: Review, taxonomy and open challenges. *Sensors*, *22*(17), 6463.

[12]. Khan, I. U., Afzal, S., & Lee, J. W. (2022). Human activity recognition via hybrid deep learning based model. *Sensors*, *22*(1), 323.

[13]. Kulsoom, F., Narejo, S., Mehmood, Z., Chaudhry, H. N., Butt, A., & Bashir, A. K. (2022). A review of machine learning-based human activity recognition for diverse applications. *Neural Computing and Applications*, *34*(21), 18289-18324.

[14]. Ray, A., Kolekar, M. H., Balasubramanian, R., & Hafiane, A. (2023). Transfer learning enhanced vision-based human activity recognition: A decade-long analysis. *International Journal of Information Management Data Insights*, *3*(1), 100142.

[15]. Li, Y., Yang, G., Su, Z., Li, S., & Wang, Y. (2023). Human activity recognition based on multienvironment sensor data. *Information Fusion*, *91*, 47-63.

[16]. Raj, R., & Kos, A. (2023). An improved human activity recognition technique based on convolutional neural network. *Scientific Reports*, *13*(1), 22581.

[17]. Bibbo, L., & Vellasco, M. M. (2023). Human activity recognition (har) in healthcare. *Applied Sciences*, *13*(24), 13009.

[18]. Ray, A., Kolekar, M. H., Balasubramanian, R., & Hafiane, A. (2023). Transfer learning enhanced vision-based human activity recognition: A decade-long analysis. *International Journal of Information Management Data Insights*, *3*(1), 100142.

[19]. Saleem, G., Bajwa, U. I., & Raza, R. H. (2023). Toward human activity recognition: a survey. *Neural Computing and Applications*, *35*(5), 4145-4182.

[20]. Dentamaro, V., Gattulli, V., Impedovo, D., & Manca, F. (2024). This survey focuses on human activity recognition using smartphone-integrated sensors. *Expert Systems with Applications*, *246*, 123143.

[21]. Wei, X., & Wang, Z. (2024). TCN-attention-HAR: Human activity recognition based on attention mechanism time convolutional network. *Scientific Reports*, *14*(1), 7414.

[22]. Yang, Z., Li, K., & Huang, Z. (2024). MFCANN: A feature diversification framework based on local and global attention for human activity recognition. *Engineering Applications of Artificial Intelligence*, *133*, 108110.

[23]. Khan, I., Guerrieri, A., Serra, E., & Spezzano, G. (2025). A hybrid deep learning model for UWB radar-based human activity recognition. *Internet of Things*, *29*, 101458.

[24]. Thakur, D., Dangi, S., & Lalwani, P. (2025). A novel hybrid deep learning approach with GWO–WOA optimization technique for human activity recognition. *Biomedical Signal Processing and Control*, *99*, 106870.

[25]. Bukht, T. F. N., Rahman, H., Shaheen, M., Algarni, A., Almujally, N. A., & Jalal, A. (2025). A review of video-based human activity recognition: theory, methods and applications. *Multimedia Tools and Applications*, *84*(17), 18499-18545.

[26]. Xu, K., Wang, J., Zhu, H., & Zheng, D. (2025). Evaluating self-supervised learning for wifi csi-based human activity recognition. *ACM Transactions on Sensor Networks*, *21*(2), 1-38.

[27]. Zhu, B., & Wei, H. (2025, July). WiFi CSI Based Passive Human Activity Recognition Using BLSTM-TCN. In *2025 4th International Conference on Electronics, Integrated Circuits and Communication Technology (EICCT)* (pp. 226-229). IEEE.

[28]. Xu, K., Wang, J., Zhu, H., & Zheng, D. (2025). Evaluating self-supervised learning for wifi csi-based human activity recognition. *ACM Transactions on Sensor Networks*, *21*(2), 1-38.