Human Activity Recognition using CNN-LSTM

By Rajesh Raj Tudu
dept. name:Electrical Engineering
Roll No. 120EE0521

Topic:Imageprocessing & Computer Vision
Supervisor:Prof. Dipti Patra
email address :120ee0521@nitrkl.ac.in

***Abstract*— Human Activity Recognition (HAR) is a challenging task due to the complexity of human motions and the variability of datasets. It gained significant attention in recent years due to its wide range of applications in various fields, including healthcare, sports analytics, and human-computer interaction. Deep learning techniques, particularly Long Short-Term Memory (LSTM) networks, have shown remarkable performance in accurately classifying and predicting human activities based on time-series data. This report provides an in-depth analysis of the application of deep learning, specifically LSTM, in the context of human activity recognition.**

Index term— Human Activity Recognition using CNN-LSTM

I. Introduction

Human Activity Recognition (HAR) has gained significant attention in recent years due to its wide range of applications in various fields, including healthcare, sports analytics, and human-computer interaction. HAR involves the identification and classification of human activities based on data collected from various sensors, such as accelerometers and gyroscopes. Deep learning methods have demonstrated superior performance in handling complex temporal data, making them well-suited for HAR tasks.

The traditional machine learning method involves extracting statistical features from raw sensor data, a task that requires specialized knowledge and can be time-consuming. It also poses a risk of losing valuable information, such as temporal relationships between actions, during feature extraction.

New deep learning models have demonstrated superior performance in HAR without handcrafted feature extraction, thanks to their stacking structure that allows them to learn representative features.

Deep learning has introduced a CNN-LSTM branch network model to identify human activities using time-series data from inertial sensors.

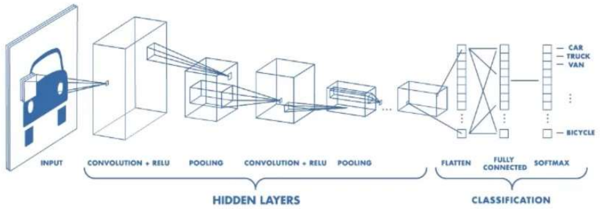
- The CNN-LSTM branch model is designed to automatically extract features while maintaining time dependencies for human activity classification.
- The process involves employing a CNN model to extract characteristics from raw data frames, which are then interpreted by an LSTM model.
- Experiments on the proposed model show superior performance compared to conventional machine learning methods and deep learning models like CNN, LSTM, or a combined CNN-LSTM architecture.

II. DEEP LEARNING ARCHITECHTURE

A. *Convolutional Neural Networks (CNNs):*

1) Convolutional neural networks (CNNs) are specialized neural networks designed for processing data with a known grid-like topology. A digital image is a binary representation of visual data consisting of a grid-like series of pixels with pixel values indicating brightness and color.

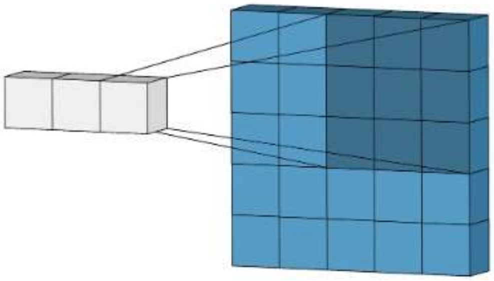
The human brain processes a significant amount of information when we see an image. Each neuron works in its own receptive field and is connected to other neurons covering the entire visual field. In a CNN, each neuron processes data only in its receptive field, detecting simpler patterns like lines and curves and more complex ones like faces and objects.



A CNN typically has three layers:

i. ***Convolution layer***

The convolution layer is the fundamental component of a CNN, responsible for distributing the majority of the network's computational workload.



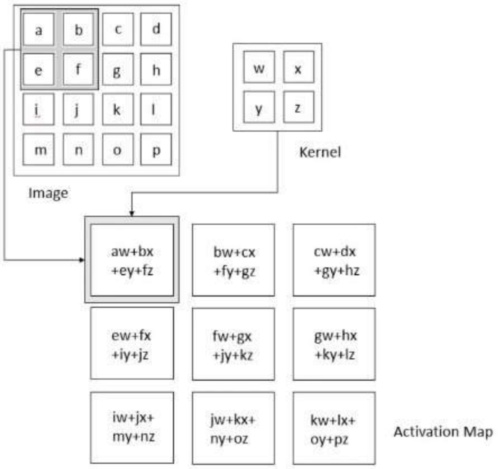
(Illustration of Convolution Operation)

The layer performs a dot product between two matrices, one representing the set of learnable parameters and the other the restricted portion of the receptive field. The kernel is spatially smaller than an image but more in-depth, meaning that in an RGB image with three channels, the kernel height and width are spatially small.

The kernel slides across the image's height and width, creating an activation map that represents the kernel's response at each spatial position. The size of the kernel's sliding is called a stride, ensuring accurate representation of the image.

If we have an input of size W x W x D and D_{out} number of kernels with a spatial size of F with stride S and amount of padding P, then the size of output volume can be determined by the following formula:

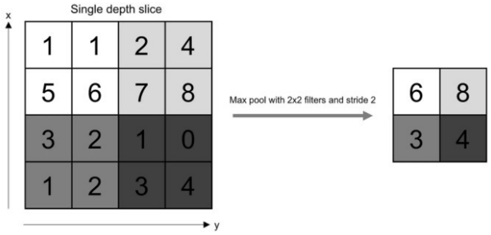
$$W_{out} = \frac{W - F + 2P}{S} + 1$$



ii. *Pooling layer*

The pooling layer replaces network output at specific locations by generating a summary statistic of nearby outputs, reducing the representation's spatial size and computation and weights required, as the operation is processed individually on each slice of the representation.

Pooling functions include the average of the rectangular neighborhood, L2 norm, and weighted average based on distance from the central pixel, but max pooling is the most popular method, reporting the maximum output from the neighborhood.



If we have an activation map of size W x W x D, a pooling kernel of spatial size F, and stride S, then the size of output volume can be determined by the following formula:

$$W_{out} = \frac{W - F}{S} + 1$$

(Formula for Padding Layer)

Output volume of size Wout x Wout x D

In all cases, pooling provides some translation invariance which means that an object would be recognizable regardless of where it appears on the frame.

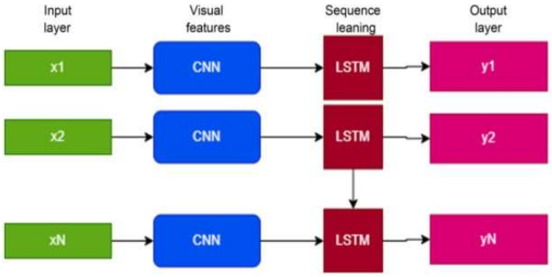
iii. *Fully connecting layer*

This layer's neurons exhibit full connectivity with all neurons in the preceding and succeeding layers, as seen in a regular FCNN, allowing for computation through matrix multiplication and bias effect.
The FC layer is the mapping the representation between the input and the output.

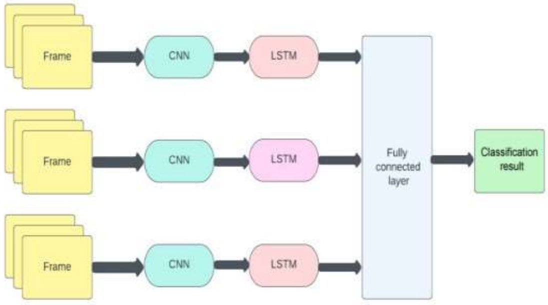
$$a \tag{1}$$

B. *Long short-term memory(LSTM):*

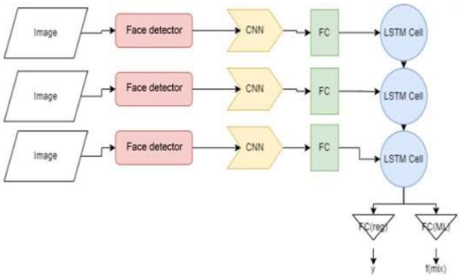
1) The Long Short-Term Memory (LSTM) model is a subtype of Recurrent Neural Networks (RNN) used to identify patterns in data sequences like sensor data, stock prices, or natural language. RNNs recognize patterns by considering the actual value and its position in the sequence in prediction.



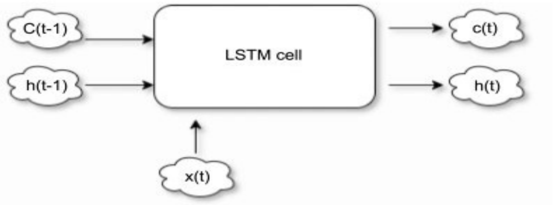
A CNN-based LSTM network at the grassroots level is shown in the flow chart below,



When an image is put in a CNN-based LSTM network the image below shows us a basic structure in which Feature Extraction happens .



The mathematical calculations behind an LST M module :



LSTM Equations

$$i_t = \sigma\left(w_i\left[h_{t-1},\, x_t\right] + b_i\right)$$

$$f_t = \sigma\left(w_f\left[h_{t-1},\, x_t\right] + b_f\right)$$

$$o_t = \sigma\left(w_o\left[h_{t-1},\, x_t\right] + b_o\right)$$

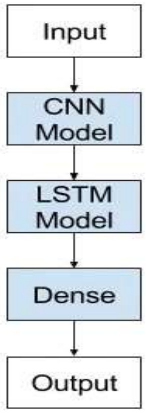
C. *CNN-LSTM Architectures:*

1) One of the most interesting and practically useful neural models come from the mixing of the different types of networks together into hybrid models.

Combining CNNs and LSTMs in hybrid architectures allows the model to simultaneously capture spatial and temporal features, enhancing the overall performance of HAR systems.

The deep CNN architecture is used to extract features from an image, which are then processed using the LSTM architecture to generate a caption.

2) The CNN-LSTM model is a deep, spatially and temporally deep architecture used for sequence prediction problems with spatial inputs like images or videos. It combines Convolutional Neural Network (CNN) layers for feature extraction and LSTMs for sequence prediction on feature vectors. This model is at the boundary of Computer Vision and Natural Language Processing, with immense potential for tasks like text classification and video conversion.



3) CNN-LSTM architecture is crucial in Computer Vision and Natural Language Processing, enabling the use of advanced neural models for NLP tasks like transformers for sequential image and video data, and powerful CNN networks for sequential data like natural language. This approach leverages the useful aspects of powerful models in tasks they have never been used for before. This post introduces hybrid neural models and encourages the use of different CNN-LSTM architectures.

III. METHODOLOGY

This study aims to identify human daily activities using camera sensor data, specifically accelerometer and gyroscope data, using a CNN-LSTM network architecture.

i. Processing

Human Activity Recognition uses sequential sensor data, which cannot be randomly split due to the possibility of data from the same participant appearing in both testing and training sets. To improve accuracy, the dataset needs to be split participant-wise, ensuring accurate representation of the model's performance.

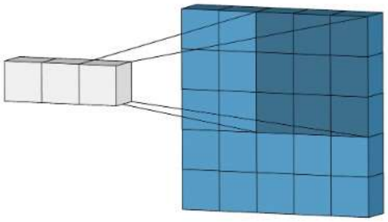
The proposed work uses publicly available datasets SHOAIB and UCI-HAR for experiments and evaluation of the approach's performance, but participant splits are required. **SHOAIB dataset:** Sensor data from five positions was collected from participants, with only left and right pocket data considered. Magnetometer sensor data was ignored. Frames were made on left and right pocket data separately with 50% overlap, then concatenated. Three participants were randomly chosen for test data, with 5,032 samples reserved for testing and 20,128 samples for training. This separation allows for a comprehensive evaluation of the model's performance on unseen or unobserved data, enabling a thorough evaluation of its efficacy.

UCI-HAR: The dataset is divided into two halves, with 30% for testing and 70% for training. Sensor data is divided into a window of size 128, 9 with 7,352 samples allocated for training and 2,947 for testing. The accelerometer signal is separated into gravitational force and body motion components using a low-pass filter with a 0.3 Hz cutoff frequency. Features are extracted from time and frequency domains to create feature vectors for each window.

ii. Features extraction

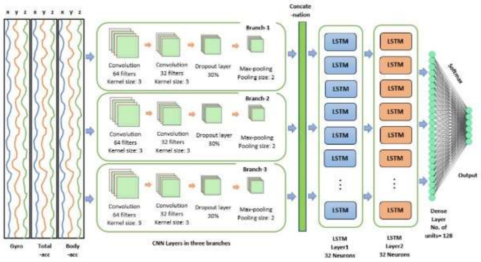
Camera sensors collect human activity data, forming a clear one-dimensional time series with strong correlation between closely related variables. This structure ensures accurate and reliable data collection.

The CNN algorithm is used to identify and isolate local features within data using local receptive fields, taking input from time-series data of
$(M \times N)$.



(Feature extraction of CNN)

where M represents the total data length and N is the no. of features available in the data. For extracting features from time-series data convolution filters are used. The filter length of the proposed model in each branch is 3 and the depth is the same as the no. of features N. The no. of feature maps created by the convolution process depends on the no. of filters employed in the operation. The sliding window technique is utilized to segment the input data into frames. CNN treats each frame as a separate unit of data, ignoring any temporal context outside of frame borders. The temporal context between the data frames is also necessary to identify activities accurately. Therefore, to capture temporal features various techniques for HAR have utilized Recurrent Neural Networks (RNNs).


x y z x y z x y z
Gyro
Total -acc
Body -acc
Branch-1
Branch-2
Branch-3
Convolution 64 filters Kernel size: 3
Convolution 32 filters Kernel size: 3
Dropout layer 30%
Max-pooling Pooling size: 2
CNN Layers in three branches
Concate -nation
LSTM
LSTM Layer1 32 Neurons
LSTM Layer2 32 Neurons
Softmax
Output
Dense Layer No. of units= 128

(CNN-LSTM Architechrure)

RNN faces vanishing gradients problem, unable to capture long-term dependencies. LSTM, with its chain-like structure and numerous gates on the repeating module, is better at managing long-term dependencies, allowing the network to adapt its predictions more accurately to new contexts.

iii. Model constriction and validation

This proposed CNN-LSTM branch network uses convolution layers to extract human activity features in each branch, then passes these features through a LSTM layer, and a dense layer classifies the activity.

Above figure proposed CNN-LSTM branch approach classifies activities using the Keras API, which can be implemented quickly and efficiently. The model uses TensorFlow as the backend on NVIDIA GTX 1050TI(GPU) and takes 9 signals: acceleration(a_x, a_y, a_z), linear acceleration (la_x, la_y, la_z), and angular velocity(g_x, g_y, g_z). This makes it a valuable tool for conducting impactful research.

Sensor data is converted into a fixed window size and passed through three convolution layers with 64 filters, each with 32 filters. These layers extract essential features for human activity recognition. Rectified linear units (ReLU) are used to construct feature maps in both layers. A 30% dropout layer is added in each branch, and a pooling layer with pool size 2 is passed through. Max pooling is employed within the pooling layer to reduce computational load and improve translation invariance, reducing the number of factors needed.

The proposed model uses concatenated outputs from each branch and introduces two LSTM layers with 32 and 64 neurons to extract temporal dependencies for short-term human activities. The output is then passed to a Dense layer with 128 neurons, using the rectified linear units (ReLU) activation function. The 50% output of the Dense layer is normalized and passed to a fully linked output layer with Softmax activation for classifying human activity. The model was trained using an RMSprop optimizer with a 0.0001 learning rate and 32 batch sizes over 200 epochs with early stopping.

iv. **Matrices performance**

The suggested model's efficacy is evaluated using various performance metrics.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$f1 - score = \frac{2 \times precision \times recall}{precision + recall}$$

IV. RESULTS

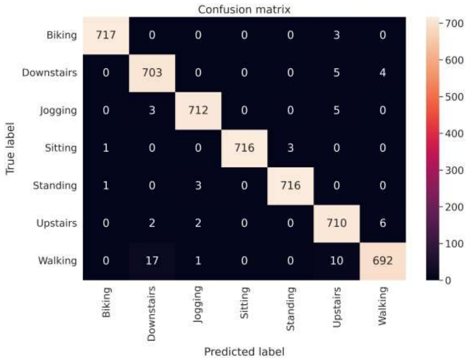
Algorithms	SHOAIB		UCI	
	Accuracy	Loss	Accuracy	Loss
CNN	96.7%	0.12	91.48%	0.27
LSTM	95.3%	0.17	89.03%	0.33
CNN-LSTM(without branch)	96%	0.14	89.21%	0.34
Proposed Method	**98%**	**0.07**	**93.72%**	**0.25**

TABLE I. PERFORMANCE MEASUREMENT OF HAR USING DIFFERENT ALGORITHMS

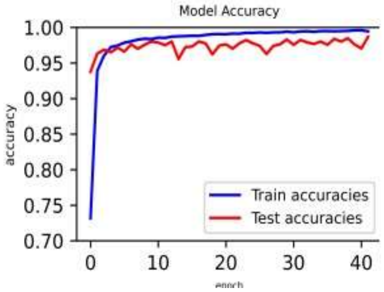
i. **SHOAIB Dataset:** Shoaib et al. conducted a study on ten male participants aged 25-30 who performed eight daily activities for 3-4 minutes. They wore five smartphones and collected data from sensors like gyroscope, accelerometer, linear accelerometer, and magnetometer. The data collection took place inside, except for biking. Fig. 3 depicts the accuracy and loss values across epochs, while Fig. illustrates the corresponding confusion matrix.

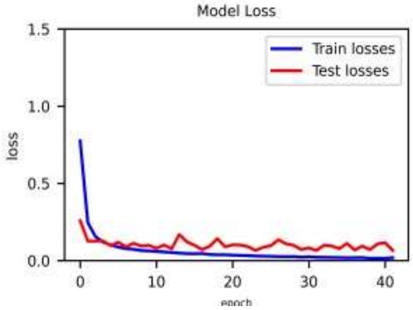
Activity	Specificity	Recall	Precision	F1-score
Biking	0.997217	0.995833	0.983539	0.989648
Walking	0.996058	0.938889	0.975469	0.956829
Jogging	0.998840	0.981944	0.992978	0.987430
Walking upstairs	0.994434	0.965278	0.966620	0.965949
Walking downstairs	0.990509	0.987360	0.944892	0.965659
Sitting	1.000000	0.997222	1.000000	0.998609
Standing	0.999768	0.994444	0.998605	0.996521

TABLE II. PERFORMANCE MEASUREMENT ON SHOAIB DATASET



Confusion matrix of SHOABI Dataset

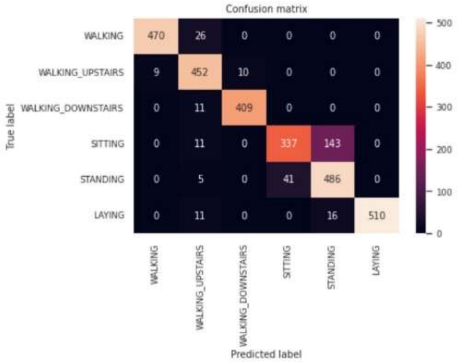




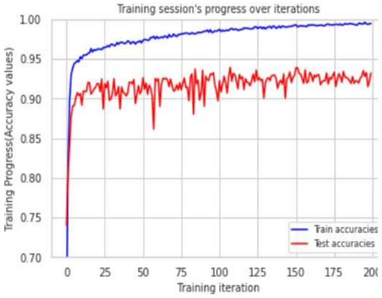
ii. **UCI Dataset**

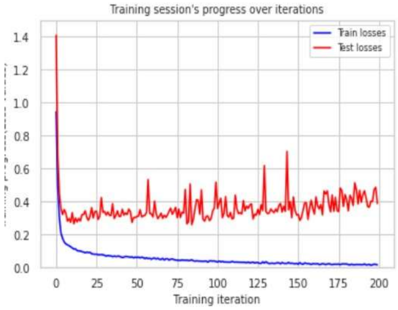
The study involved 30 volunteers aged 23 to 43 wearing a Dell laptop camera around their waists during six activities. Smartphone sensors recorded linear acceleration and angular velocity in three directions at a 50 Hz frequency. Data was manually labeled and randomly split into training and testing data, with a 70% training data and 30% testing data ratio.

Activity	Specificity	Recall	Precision	F1-score
Walking	0.990208	0.993952	0.953578	0.973346
Walking upstairs	0.983037	0.919321	0.911579	0.915433
Walking downstairs	0.989711	0.942857	0.938389	0.940618
Sitting	0.969055	0.818731	0.841004	0.829721
Standing	0.973085	0.851504	0.874517	0.862857
Laying	1.000000	1.000000	1.000000	1.000000



Confusion matrix of UCI Dataset





V. Conclusion

The proposed method accurately distinguishes various human activities using deep learning techniques, combining the advantages of LSTM networks and convolutional neural networks. This method eliminates the need for manually created features in conventional machine learning methods. The method for Human Activity Recognition (HAR) using camera sensors offers the highest accuracy and efficiency compared to traditional techniques. Future work aims to extend this method for camera position-independent activity recognition.

References

[1] Deep Learning with Python by François Chollet, 379–393, Jun. 2015,

[2] Machine Learning (NCIM) 16-17 June 2023, Gazipur-1707, Bangladesh.