Human Activity Recognition

Sanket Vikenbhai Shah
Post-graduate Student
Department of Computer Science
Dalhousie University
Halifax, NS, Canada B3H 4R2
sn488207@dal.ca

Abstract

Human Activity Recognition conceptualized on wearable sensors is one of the most agile and rapidly growing areas of research in computer vision for various contexts like security surveillance, healthcare, and human computer interaction. In the following project, the assembled signal sequence of accelerometers coupled with gyroscopes will enable Convolutional Neural Networks to automatically learn the optimal features for the activity recognition task. For the same, I have used the ConvLSTM approach for activity recognition. The CNN will extract the features from each block and the LSTM will interpret the features extracted from each block. I have tested this model using UCI Human Activity Recognition Dataset, which is captured using 30 subjects for 6 different activities. For this, Data is captured using accelerometer and gyroscope sensors.

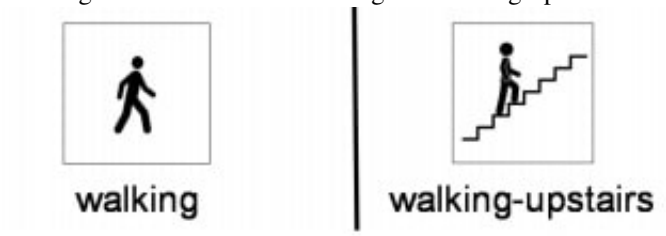
1 Introduction

Human Activity Recognition is a captivating topic for vast research and is full of arduous challenges. With improved technology, we now have precisely more accurate sensors and faster processors with lesser power consumption. Very Recently, Human Activity Recognition Dataset has attracted rapt attention both from the industry as well as academia. Due to rapidly increasing amount of video records, based on automatic video analysis such as visual surveillance, human-machine interfaces, sports video analysis, and video retrieval it is important to make the detection more accurate and robust.

Human Activity Recognition system is classified into two types based on sensing method: vision-based methods and acceleration-based methods. While the vision-based method generally uses one or more cameras to collect data, the acceleration-based method asks the users to enable several accelerometers for collecting data by wearing them. The advantage of a vision-based system compared to acceleration-based system is that it works without placing any sensors with users, although its recognition performance depends on the light condition primarily, as well as visual angles, and other parameters. On the contrary, an acceleration-based system does require the users to wear a device, but tries to eliminate all the external interferences.

Preprint. Under review.

Figure 1: Activities of walking and walking-upstairs



With the aim of developing a Human Activity Recognition system with high accuracy, good robustness, and quick response, I successfully built a deep architecture for an acceleration-based Human Activity Recognition system. The model has been evaluated and thoroughly checked on a large dataset (with 30 subjects from 6 typical activities). The result has yielded to be promising, and reaches a higher accuracy than previously implemented methods.

2 Related work

There are so many architectures and algorithms that can be used for Human Activity Recognition. Evaluating an activity in most cases do requires Gyroscope and Accelerometer, which can be expensive and not feasible to everyone on a large scale.

Method for detecting these activities is using these data gathered from these devices which will be later evaluating by the model being prepared for the same. Therefore, one of the most basic ways to recognize the Human Activity is using Convolution Neural Networks (CNNs).

The CNN-based method consists of different stages like Data Visualisation, Scaling, Data Sampling as well as Image labelling and Image resizing. The first stage is the classification of images using Data Visualisation which includes removing any outliner. Another stage includes Scaling which is basically scaling of the data as of considering average data. Third stage of this is Data Sampling which is the part of the data classification.

Earlier, some of the Object Detection fields started with Recurrent Neural Networks (RNN) and moved to faster and more advanced algorithms like YOLO (You Only Look Once) and SSD (Single-Shot Detection). The RNNs use selective search algorithms to extract 2000 bounding boxes from the input image in the first step itself to stick with processing only the most essential features in an input based on colour, pattern, shape, and size [11].

This RNNs use memory to record past happenings, also it produces output based on it. RNNs is useful because of its feature that it can feed back the output back into the network by copying the output.

Also, LSTM can be the approach for Human Activity Recognition as this is the extension of recurrent neural networks in terms of features. LSTM is extended memory which solves the issue of vanishing gradients.

The ConvLSTM approach is useful to produce good results with the usage of fewer inputs. This approach has convolutional structures which in input-to-state and state-to-state transitions. The ConvLSTM network can capture spatiotemporal correlations in good and regular manner which overcome the features of FC-LSTM [9].

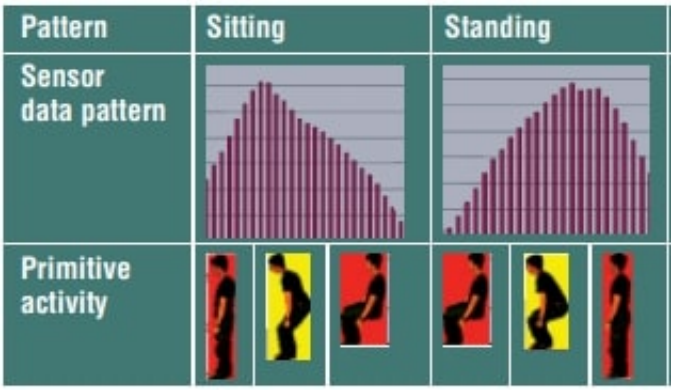
3 Data and Methodology

3.1 UCI Human Activity Recognition dataset pre-processing steps included:

Pre-processing accelerometer and gyroscope using noise filters are enabled. Sensor data is captured at a frequency measuring exactly 50 Hz. The Splitting of data takes place into fixed windows of 2.56 seconds (128 data points) with precisely 50% overlapping. The Splitting of accelerometer data takes place as composition of the gravitational (total) and body motion components.

Numerous frequency and time features, commonly used in the field of human activity recognition were extracted from each window. The result was a 561-elements' vector comprising the features. The dataset was split into the train (70%) and test (30%) sets based on data for subjects, e.g. 21 subjects for train and nine for the test.

Figure 2: Effect on sensor while doing the above activities[13]

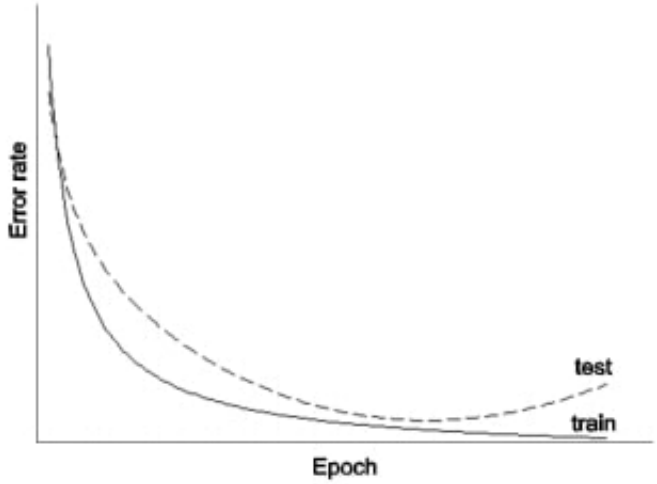


Own dataset - MATLAB android application

3.2 Locate the best epochs

Once the structure of the CNN is determined, the network has to be trained by tuning the best parameters for activity recognition. Generally, the error rate of the training set will gradually reduce as training proceeds. In the beginning, the error rate on the test set will decrease. After a specific epoch, the error rate will stop decreasing, and may increase sometimes. This phenomenon is called overfitting and is caused by over-training the network.

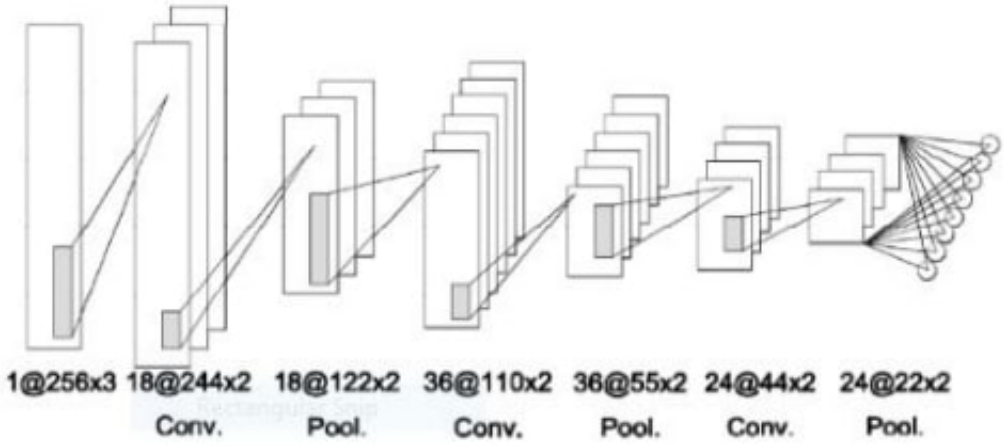
Figure 3: Test and Train error in training process[15]



3.3 Architecture of CNN About the inputs

That dataset contains 9 channels in total of the inputs: (acc_body, acc_total and acc_gyro) on x-y-z. Hence the input channel is 9. Towards the end, I reformatted the inputs from 9 inputs files to 1 file, the shape of that file is [n_sample,128,9], that is, every window has 9 channels with each channel having length 128.

Figure 4: CNN Architecture



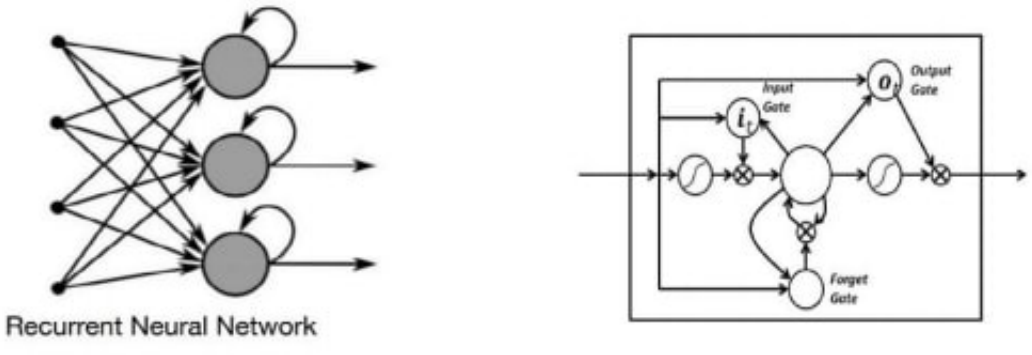
Convolution + pooling + convolution + pooling + dense + dense + dense learning_rate = 0.001 dropout = 0.8 training_epoch = 20 kernel_size = 64 (total 32)

3.4 LSTM vs RNN

A Recurrent Neural Network is able to remember its past happenings, because of its internal memory. It produces an output, copies that output and feeds it back into the network. There are two issues of standard RNN: Exploding and Vanishing Gradients.

Long Short-Term Memory (LSTM) networks are an extension for recurrent neural networks, which basically extend their memory and mainly solve the problem of exploding as well as vanishing gradients.

Figure 5: RNNs



3.5 ConvLSTM Approach

The following components are explained: CNN: to read sub sequences from the main sequences in block by extracting feature from each block

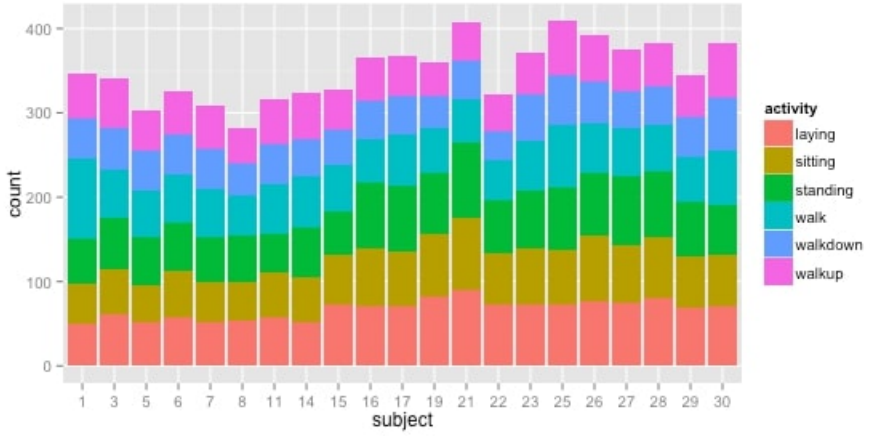
LSTM: to interpret the features extracted from each block as Input. Samples: n, for the number of windows in the dataset.

Time: 4, for the four sub sequences that a window of 128-time steps was split into. Rows: 1, for the one-dimensional shape of each subsequence. Columns: 32, for the 32-time steps in an input subsequence. Channels: 9, for the nine input variables.

3.6 Details about Datasets

The Dataset used in this paper is formed using a single accelerometer. It is different from UCI Human Activity Recognition dataset as UCI Human Activity Recognition uses multiple sensors to capture the data. The UCI Human Activity Recognition dataset uses many signal processing approaches to find out feature vectors. DCT (Discrete Fourier Transform), FFT (Fast Fourier Transform), PCA (Principal Component Analysis), AR (Autoregressive Model) and HAAR filters are used to find the feature vectors of constructed dataset. The Dataset that I have used, consists of 31688 samples and 8 different activities. Using this dataset, the researchers are able to achieve an accuracy of precisely 87.9%.

Figure 6: Plotting average acceleration for first subject[14]



The researchers have suggested using HCII - SCUT dataset [3] which is also constructed using tri-axial accelerometer for the implementation of same application. It consists of 1278 samples of 44

different subjects and 10 different activities. It has also used the models DCT, FFT and AR model to construct the feature vector.

3.7 Model Summary

Figure 7: Model Summary

```
Model: "sequential_61"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv_lst_m2d_61 (ConvLSTM2D) (None, 1, 30, 64)         56320
_________________________________________________________________
dropout_60 (Dropout)         (None, 1, 30, 64)         0
_________________________________________________________________
flatten_60 (Flatten)         (None, 1920)              0
_________________________________________________________________
dense_120 (Dense)            (None, 100)               192100
_________________________________________________________________
dense_121 (Dense)            (None, 6)                 606
=================================================================
Total params: 249,026
Trainable params: 249,026
Non-trainable params: 0
```

4 Experiments

Various experiments were conducted on the selected dataset. For the purpose of experiments, data had been collected through MATLAB mobile application which includes various activities for the recognition. Those activities are as described like Walking, Walking upstairs, Walking downstairs, Sitting, Standing and Laying. Below image describes Walking upstairs activity.

Figure 8: Sensors data while different activities

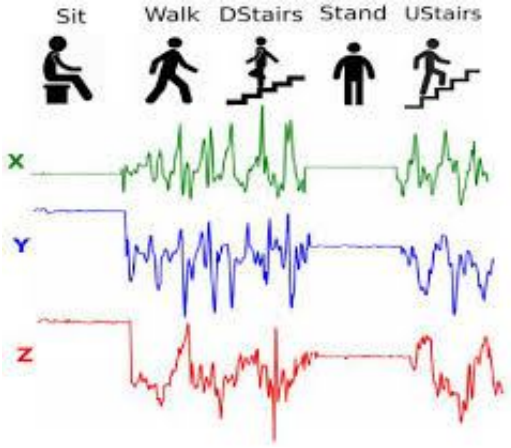
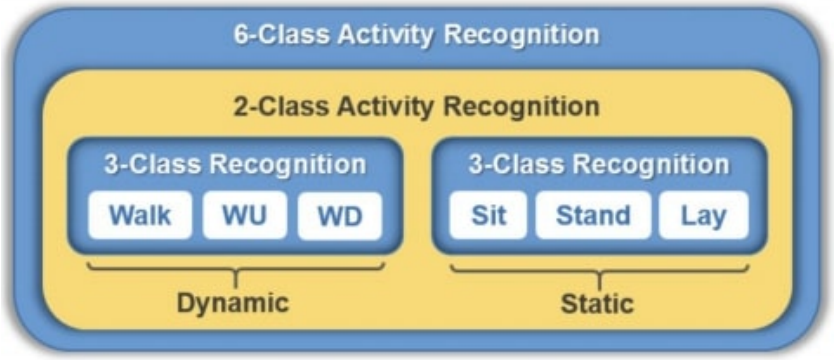


Figure 9: Walking Upstairs Example

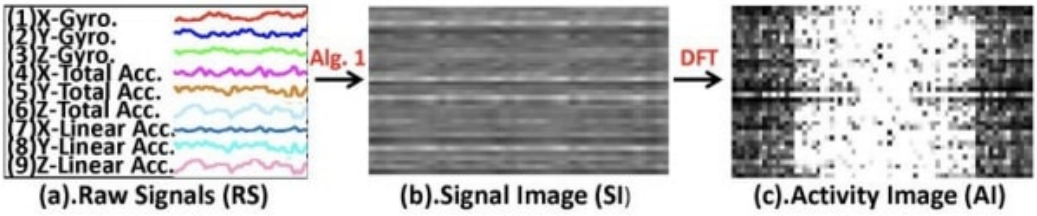
Experimented activities were divided into two parts like Dynamic and Static activities.

Figure 10: 6-Class Activities



Processing of Raw Sensor data into an Image is being done as per below shown.

Figure 11: Raw Sensor data into an Image



4.1 Model Tuning

At the first, all the CNNs were tuned separately to improve overall performance. CNNs, RNNs, LSTM and ConvLSTM model were used for tuning. From all of these, ConvLSTM gave the most accuracy and relevant required result.

4.2 Model Results

4.2.1 Result of CNN based architecture

Figure 12: CNN Result

dropout	learning_rate	training_epoch	training_accuracy	testing_accuracy
1	0.001	100	0.9592	0.8782
0.9	0.001	100	0.9334	0.8724
0.85	0.001	100	0.945	0.868
0.8	0.001	50	0.936	0.866
0.8	0.001	100	0.925	0.865
1	0.001	30	0.9237	0.8629
1	0.001	100	0.9539	0.8622
0.75	0.001	50	0.935	0.861
0.9	0.001	100	0.9162	0.8588
1	0.005	30	0.9377	0.8558

4.2.2 Result of ConvLSTM based architecture

Figure 13: Result of implementation of ConvLSTM

```
Mounted at /content/drive
(7352, 128, 9) (7352, 1)
(7352, 128, 9) (7352, 6)
(2947, 128, 9) (2947, 1)
(2947, 128, 9) (2947, 6)
>#1: 90.702
>#2: 89.786
>#3: 92.535
>#4: 91.110
>#5: 90.227
>#6: 90.533
>#7: 90.261
>#8: 88.191
>#9: 90.940
>#10: 91.381
[90.7024085521698, 89.78622555732727, 92.53478050231934, 91.10960364341736, 90.22734761238098, 90.53274393081665,
Accuracy: 90.567% (+/-1.070)
```

5 Learnings

5.1 Activation function and Loss function

In a neural network, an activation function specifies how the input's weighted number is converted into an output from a node or nodes in a layer. The activation function chosen has a significant impact on the neural network's capability and efficiency.

Below are some images which shows the data for test training and Test Accuracy in respect of Loss function:

Figure 14: Test Accuracy and Loss

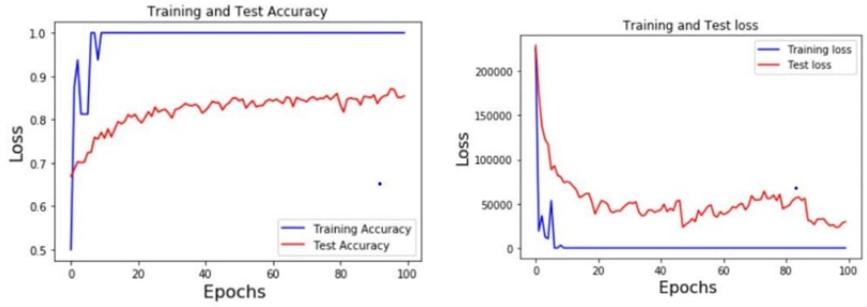
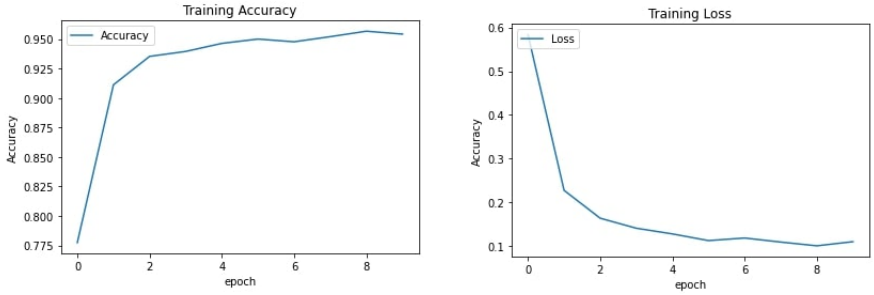


Figure 15: Training Accuracy and Loss



From above Fig 14, it can be seen that training accuracy reaches stagnancy after 40-60 epochs and rarely changes as epochs keep increasing. In the beginning, the validation preprint accuracy was linearly increasing with loss, but then it did not increase significantly.

Changes done while implementing the approach used by author: I implemented this model on UCI Human Activity Recognition dataset and will do it on my existing dataset as well. I have tried this with all of the existent 9 channels instead of only 3 channels. I decreased the input size to 128 instead of 256 and reduced to 2 convolutional layer and added one more fully connected layer to improve accuracy and decrease training and for specifically testing loss.

5.2 Confusion Matrix

Figure 16: Confusion Matrix

```
Confusion Matrix:
[[458   9  29   0   0   0]
 [  3 444  24   0   0   0]
 [  3   7 410   0   0   0]
 [  0  25   0 413  53   0]
 [  1   2   0  75 454   0]
 [  0  27   0   0   0 510]]
```

As shown in above 16, confusion matrix can be generated using the predicted and actual values. This matrix is used for evaluating the performance of a classification model.

5.3 Classification Report

Figure 17: Classification Report

Classification Report:

	precision	recall	f1-score	support
0	0.98	0.92	0.95	496
1	0.86	0.94	0.90	471
2	0.89	0.98	0.93	420
3	0.85	0.84	0.84	491
4	0.90	0.85	0.87	532
5	1.00	0.95	0.97	537
accuracy			0.91	2947
macro avg	0.91	0.91	0.91	2947
weighted avg	0.91	0.91	0.91	2947

As shown in above Fig 17, classification report is being generated using data of the respective model which represents the main classification metrics on a per-class basis.

6 Discussion and Conclusion

The System proposed here is an acceleration and gyroscope based Human Activity Recognition algorithm using CNN and ConvLSTM, a popular used deep architecture in image recognition. According to the characteristics of acceleration data, I modified the conventional CNN structure.

The experiments are executed on a large dataset of six kinds of typical activities from 30 subjects. The results show that the improved CNN works well, reaches an accuracy of precisely 83.33%. But the ConvLSTM works even better with accuracy of 90.567% as checked.

The proposed model is accurate and robust without any feature extractions and is is suitable for building a real-time Human Activity Recognition system on mobile platforms as well.

Acknowledgement

I am grateful to Dalhousie University for providing me the support required for the experiment. I want to thank professor Sageev Oore for enhancing my knowledge of Machine Learning.

References

[1] David G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints”: Computer Science Department, University of British Columbia, Vancouver, B.C., Canada, lowe@cs.ubc.ca; January 5, 2004

[2] Coşkun, Musab, et al. "Face recognition based on convolutional neural network." Modern Electrical and Energy Systems (MEES), 2017 International Conference on. IEEE, 2017.

[3] Yang Xue, and Lianwen Jin, "A naturalistic 3D acceleration-based activity dataset benchmark evaluations," Systems, Man and Cybernetics, pp. 4081-4085, 2010.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, pp. 1097-1105, 2012.

[5] Wang, Jindong, et al. "Deep learning for sensor-based activity recognition: A survey." Pattern Recognition Letters (2018).

[6] Ordóñez, Francisco Javier, and Daniel Roggen. "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition." Sensors 16.1 (2016): 115.

[7] Jiang W, Yin Z Proceedings of the 23rd ACM international conference on Multimedia. Human Activity Recognition using Wearable Sensors by Deep Convolutional Neural Networks[C] ACM, 2015:1307-1310.

[8] Yuwen Chen, Kunhua Zhong, Ju Zhang, Qilong Sun and Xueliang Zhao, “LSTM Networks for Mobile Human Activity Recognition” in International Conference on Artificial Intelligence: Technologies and Applications (ICAITA 2016).

[9] Hammerla, Nils Y., Shane Halloran, and Thomas Ploetz. "Deep, convolutional, and recurrent models for human activity recognition using wearables." arXiv preprint arXiv: 1604.08880(2016).

[10] Simonyan, Karen, and Andrew Zisserman. "Two-stream convolutional networks for action recognition in videos." Advances in neural information processing systems. 2014.

[11] Girshick, Ross, et al. "Region-based convolutional networks for accurate object detection and segmentation." IEEE transactions on pattern analysis and machine intelligence 38.1 (2015): 142-158.

[12] Dataset Index of /ml/machine-learning-databases/00240", Archive.ics.uci.edu, 2021. [Online]. Available: https://archive.ics.uci.edu/ml/machine-learning-databases/00240/. [Accessed: 25- March- 2021].

[13] https://ieeexplore.ieee.org/abstract/document/5370804

[14] http://rstudio-pubs-static.s3.amazonaws.com/19016_4609dff531fd43e1a87673795e2d2dcd.html

[15] Yuqing Chen, Yang Xue “A Deep Learning Approach to Human Activity Recognition Based on Single Accelerometer” 978-1-4799-8697-2/15 2015 IEEE.