Human Activity Recognition

Ashna Jain(201501008)
ashna.j.btechi15@ahduni.edu.in

Dhruti Chandarana (201501015)
dhruti.c.btechi15@ahduni.edu.in

Janvi Patel(201501072)
janvi.p.btechi15@ahduni.edu.in

Nishi Patel(201501076)
nishi.p.btechi15@ahduni.edu.in

Charmi Chokshi (201501021)
charmi.c.btechi15@ahduni.edu.in

Hena Ghonia (201501032)
hena.g.btechi15@ahduni.edu.in

Abstract—Human Activity Recognition based on wearable sensors is one of the active areas of research in computer vision for various different contexts like security surveillance, healthcare and human computer interaction. Assembled signal sequence of accelerometers and gyroscopes will enables Convolutional Neural Networks to automatically learn the optimal features for the activity recognition task. We have used ConvLSTM approach for activity recognition. CNN will extract the features from the each block and LSTM will interpret the features extracted from each blocks. We have tested our model using UCI HAR Dataset, which is captured using 30 subjects for 6 different activities. Data is captured using accelerometer and gyroscope sensors.

Keywords- tri-axial acceleration signal; human activity recognition; deep architecture; convolution kernel; LSTM

I. INTRODUCTION

Human activity recognition is a significant research topic and it is full of challenges.With improved technology , we now have more accurate sensors and faster processors with lower power consumption.Recently HAR Dataset has attracted more and more attention both from industry as well as from academia.

A HAR system can be divided into several modules , sensing , segmentation , feature extraction , classification and post pre-processing.Generally, according to sensing method, we can classify HAR system into two types: vision-based and acceleration-based methods. Vision-based method usually uses one or more cameras to collect data, while acceleration-based method asks the users to wear several accelerometers for data collecting. The advantage of vision-based system is that it works without placing any sensors with users, but its recognition performance highly depends on light condition, visual angle and other outer factors. On the contrary, acceleration-based system requires users to wear a device, but almost eliminates all those outer interferences.

In the recent years deep architectures due to their dramatically encouraging performance, have been involved in solving series of problems. Some of the applications based on deep architecture have already been brought into service. With the aim of developing a HAR system with high accuracy, good robustness and quick response, we successfully built up a deep architecture for acceleration-based HAR system. The model has been evaluated on a large dataset (with 30 subjects from 6 typical activities). The result is promising, which reaches a higher accuracy than former methods. And more notably, the deep model directly operates on raw data. In other words, the model doesn't need an extra procedure of feature extraction, which matches our goal of quick response. In this paper, we briefly describe our used CNN approach.

II. Related work and Proposed Approach

A. UCI HAR dataset Preprocessing

The pre-processing steps included:
Pre-processing accelerometer and gyroscope using noise filters. Sensor Data is captured at frequency of 50 Hz.
Splitting data into fixed windows of 2.56 seconds (128 data points) with 50% overlap.Splitting of accelerometer data into gravitational (total) and body motion components.
A number of time and frequency features commonly used in the field of human activity recognition were extracted from each window. The result was a 561 element vector of features.
The dataset was split into train (70%) and test (30%) sets based on data for subjects, e.g. 21 subjects for train and nine for test.
Own dataset - Matlab android application

B. Locate the best epochs

Once the structure of the CNN is determined, we have to train the network, tuning the best parameters for activity recognition. Generally, the error rate of training set will gradually reduce as the training proceeds. At the beginning, the error rate on the test set will decrease, after a specific epoch, the error rate will stop decreasing, sometimes even increase. This phenomenon is called over fitting, it’s caused by over training the network.

C. Architecture of CNN
About the inputs

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

Convolution + pooling + convolution + pooling + dense + dense + dense

learning_rate = 0.001

dropout = 0.8

training_epoch = 20

kernel_size = 64 (total 32)

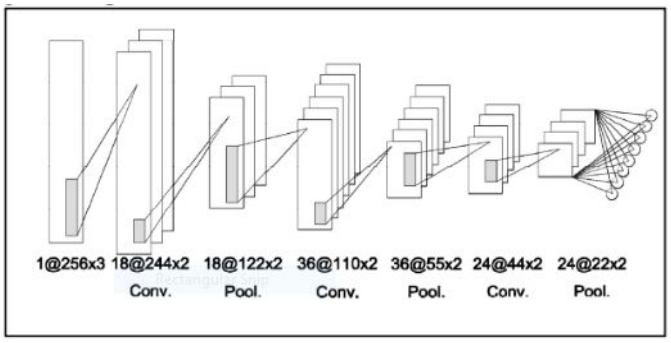
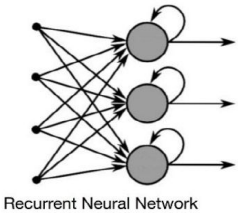
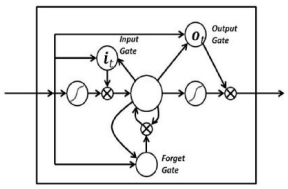


Fig1. CNN Architecture

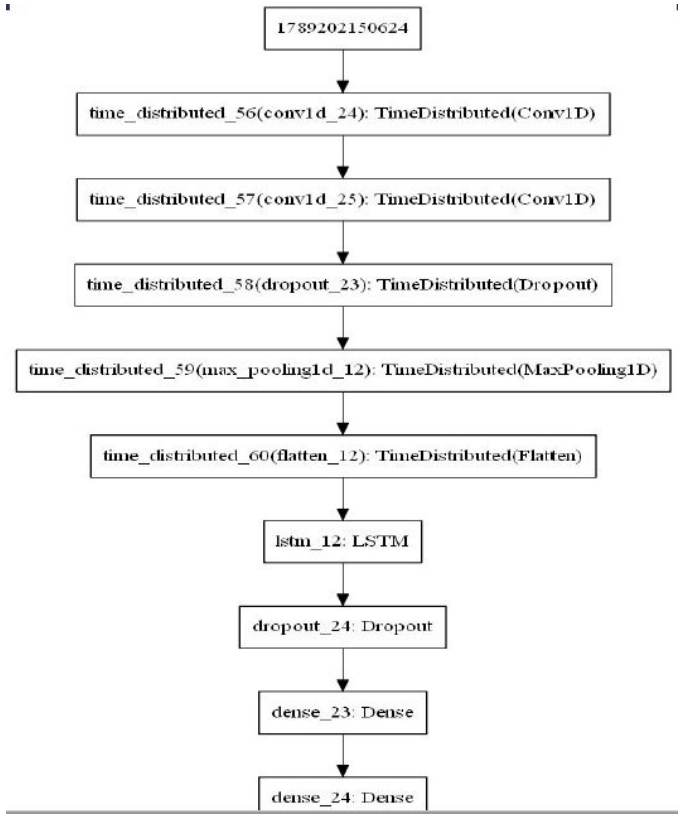
D. LSTM vs RNN
A Recurrent Neural Network is able to remember its past, because of it's internal memory. It produces output, copies that output and loops it back into the network.
There are two issues of standard RNN: Exploding and Vanishing Gradients.
Long Short-Term Memory (LSTM) networks are an extension for recurrent neural networks, which basically extends their memory and solves the problem of vanishing and exploding gradients.


Recurrent Neural Network



E. ConvLSTM Approach
CNN: read subsequences of the main sequences in block: extract feature from each block
LSTM: interpret the features extracted from each block.
Input:
Samples: n, for the number of windows in the dataset.
Time: 4, for the four subsequences that we split a window of 128 time steps 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.

F. Model Summary



Layer (type)	Output Shape	Param #
time_distributed_56 (TimeDis	(None, None, 30, 64)	1792
time_distributed_57 (TimeDis	(None, None, 28, 64)	12352
time_distributed_58 (TimeDis	(None, None, 28, 64)	0
time_distributed_59 (TimeDis	(None, None, 14, 64)	0
time_distributed_60 (TimeDis	(None, None, 896)	0
lstm_12 (LSTM)	(None, 100)	398800
dropout_24 (Dropout)	(None, 100)	0
dense_23 (Dense)	(None, 100)	10100
dense_24 (Dense)	(None, 6)	606

Total params: 423,650
Trainable params: 423,650
Non-trainable params: 0

None

G. Details about Datasets

Dataset used in this paper is constructed with the help of single accelerometer. It is different from UCI HAR dataset as it UCI HAR uses multiple sensors to capture the data. UCI HAR dataset uses many signal processing approaches to find out feature vector. DCT (Discrete Fourier Transform), FFT (Fast Fourier Transform), PCA (Principal Component Analysis), AR (Autoregressive Model) and Haar filters are used to find out feature vector of constructed dataset. Dataset consists of 31688 samples and 8 different activities. Using this dataset they are able to achieve accuracy of 87.9%.

They have also suggested 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 DCT, FFT and AR model to construct the feature vector.

III. RESULTS

Fig 1(I). Improved 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	0.8568
1	0.005	30	0.9377	0.8558

Fig1(II). Improved CNN result

```
Epoch:19,batch:1616,loss:3861.32617188,accuracy:0.81250000
Epoch:19,batch:2416,loss:3103.45263672,accuracy:0.93750000
Epoch:19,batch:3216,loss:0.00000000,accuracy:1.00000000
Epoch:19,batch:4016,loss:2461.26611328,accuracy:0.87500000
Epoch:19,batch:4816,loss:0.00000000,accuracy:1.00000000
Epoch:19,batch:5616,loss:10780.86328125,accuracy:0.87500000
Epoch:19,batch:6416,loss:6024.33496094,accuracy:0.93750000
Epoch:19,batch:7216,loss:0.00000000,accuracy:1.00000000
Epoch:20,batch:016,loss:0.00000000,accuracy:1.00000000
Epoch:20,batch:816,loss:0.00000000,accuracy:1.00000000
Epoch:20,batch:1616,loss:0.00000000,accuracy:1.00000000
Epoch:20,batch:2416,loss:9006.54785156,accuracy:0.68750000
Epoch:20,batch:3216,loss:0.00000000,accuracy:1.00000000
Epoch:20,batch:4016,loss:1838.11120605,accuracy:0.81250000
Epoch:20,batch:4816,loss:4942.12695312,accuracy:0.93750000
Epoch:20,batch:5616,loss:0.00000000,accuracy:1.00000000
Epoch:20,batch:6416,loss:0.00000000,accuracy:1.00000000
Epoch:20,batch:7216,loss:0.00000000,accuracy:1.00000000
Optimization finished!
Accuracy of testing:0.83338988
```

Fig 2. ConvLSTM Result

```
Using TensorFlow backend.

(7352, 128, 9) (7352, 1)
(2947, 128, 9) (2947, 1)
(7352, 128, 9) (7352, 6) (2947, 128, 9) (2947, 6)
2947/2947 [==============================] - 4s 2ms/step
>#1: 88.768
2947/2947 [==============================] - 8s 3ms/step
>#2: 91.381
2947/2947 [==============================] - 8s 3ms/step
>#3: 90.567
2947/2947 [==============================] - 8s 3ms/step
>#4: 88.768
2947/2947 [==============================] - 8s 3ms/step
>#5: 90.872
2947/2947 [==============================] - 9s 3ms/step
>#6: 90.261
2947/2947 [==============================] - 8s 3ms/step
>#7: 89.413
2947/2947 [==============================] - 9s 3ms/step
>#8: 89.617
2947/2947 [==============================] - 9s 3ms/step
>#9: 91.279
2947/2947 [==============================] - 9s 3ms/step
>#10: 89.786
[88.76823888700373, 91.38106549032915, 90.56667797760434, 88.76823888700373, 90.87207329487615, 90.26128266033254, 89.41296233
457754, 89.61655921275874, 91.27926705123855, 89.78622327790974]
Accuracy: 90.071% (+/-0.906)
```

Fig 3. Result of implementation of CNN architecture

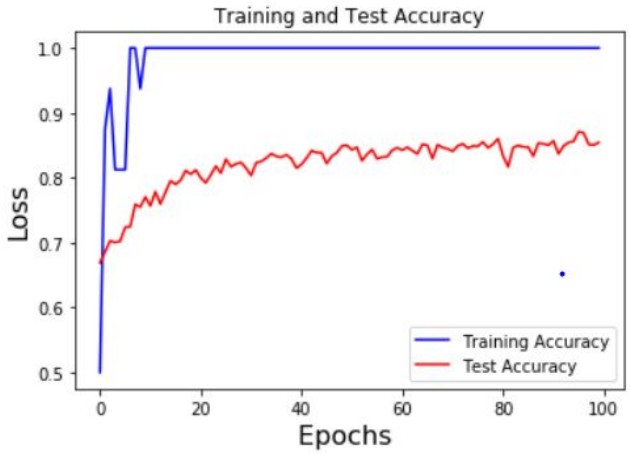


Fig 4. Training and test accuracy

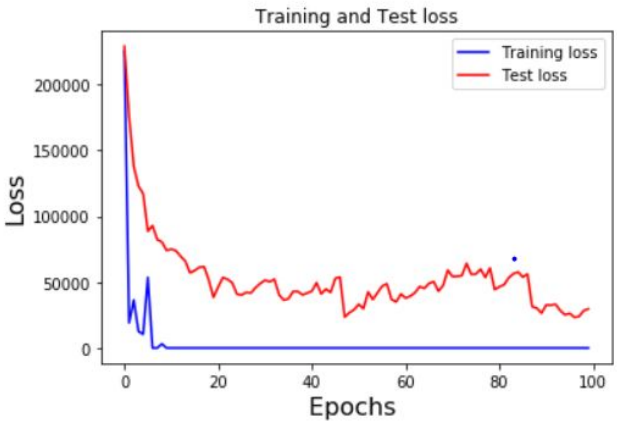


Fig 5. Training and test loss.

From above Fig 4. it can be seen that training accuracy becomes almost stagnant after 40-60 epochs and rarely changes as epochs increases. In the beginning, the validation

accuracy was linearly increasing with loss, but then it did not increase much.

Changes done while implementing the approach used by author: We implemented this model on UCI HAR dataset and will do it on our dataset as well. We tried with all 9 channels instead of 3 channels. We 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 testing loss.

IV. CONCLUSION

System proposed an acceleration and gyroscope based HAR algorithm using CNN and ConvLSTM, a popular used deep architecture in image recognition. According to the characteristic of acceleration data, we 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 83.33%. But the ConvLSTM works even better with accuracy of 90.071%. The proposed model is accurate and robust without any feature extraction, which is suitable for building a real-time HAR system on mobile platform.

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.