Long -Term Activity Detection From The Wrist

Sourav Poudyal (1607167)
Shishir Pokhrel (1603242)
Thitaree Sunticheewawong (1603226)
Dipanjan Mukherjee (1638965)

Abstract

Deep learning models are suitable for identifying and classifying human activity recognition data. In this project, activity recognition of players playing basketball is performed using Deep convolution LSTM to classify the actions namely dribbling, layup, shot, pass and rebound in basketball class, and for Running, Walking, Standing and Sitting in the locomotion class.The data is collected from wearable sensors, for 12 subjects. Leave one subject out technique is implemented to cross validate the results with 11 subjects used in training and one for cross validation.

1 Introduction

Human activity detection has become an intriguing problem with the advancement of sensor technology and its application in small wearable gadgets. One body Inertial Sensor, senses hand gestures and movements to gather information in the form of acceleration signals in 3 coordinates. The goal is to provide information on a user's behavior, which permits computing systems to proactively assist users with their tasks in various sectors such as industry, health, sports and daily activities to name a few.

Research in this field poses a few challenges in the form of (a) inter class variability owing to the idea that activity are performed by individuals differently, (b) class similarity where different activities could share similar gestures, and finally Class problem due to the fact that few parts of the continuous data stream are relevant for such classification [2]

Furthermore, designing specific experiments to capture human activity data can be problematic as labeling or capturing the data can be difficult and time-consuming or inaccurate, largely due to the fact that many body parts move relative to each other during an activity, thus requiring more wearables and mobile video recording for the labeling process, as video recording from cameras for labeling purposes can be a constraint. [2] Moreover, as observation is objective, ground truth annotation is another issue that inevitably leads to an imbalance in class.In this article [6] proposes data augmentation as a technique to mitigate the effects of imbalance in class in time series data for classification problems.

Although there are some challenges in this field, progress in recognition and classification of human activities is an interesting topic. Models such as Long Term Short Term Memory (LSTM) and Recurrent Neural Network (RNN) have been known to tackle problems mentioned above, as they have feedback connections, allowing them to work on sequential data as well.

2 Related Work

Long term Short Memory Neural networks are memory based networks that perform well on sequential data due to their performance on extracting temporal dependencies. Six different human

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

activities have been identified using LSTM-RNN networks. [5] The choice of recurrent LSTM cells is justified by [4] with the proposition that LSTM cells are fundamental to allow classification of similar gestures and activities. Furthermore, [3] uses inputs from multiple sensors in time windows to feed into LSTM based RNN model. The inputs are raw signals obtained from multimodal-sensors, segmented into windows and fed into LSTM-based DRNN model improving performance of time series data compared to CNN based networks.

The approach for this project is inspired by [1], wherein single layer LSTM is shown to outperform double layers when it comes to human activity recognition.

3 Methodology

Complex sequence of movements are captured by wearable sensors which are later run through deep learning framework based on convolution and LSTM networks. Such models have proven to be suitable for multimodel sensors, whose data can be collected and processed easily. Deep learning uses neural network which exploit many layers of non linear information processing performing feature extraction and classification. Some methodology used in this project are described below.

RNN tackles the problem of data dependencies in series data by feeding back the activation of a neuron to itself with a weight and unit time delay, which provides it with a memory of the past activation, which allows it to learn the temporal dynamics of the sequential data.

LSTM are recurrent networks that include a memory to model temporal dependencies in time series problems.

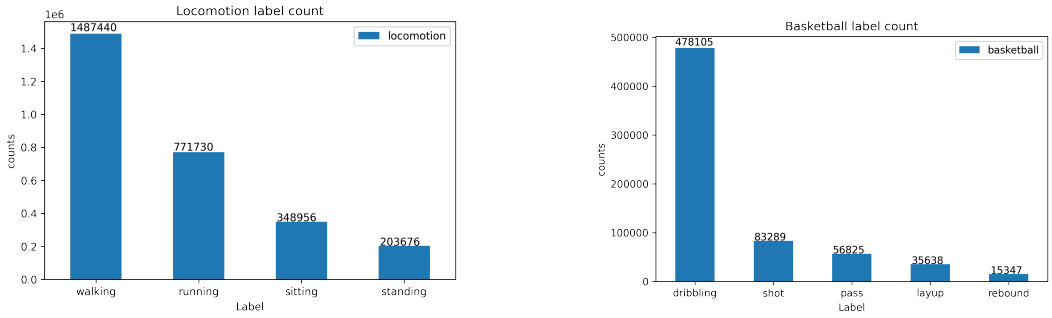
LOSO(Leave one subject out cross validation) Leave-One-Subject-Out can be described as k-fold Cross-Validation where a fold represents all data of one subject and thus k is equal to the number of subjects in the data. The validation loop is then k training cycles where each time a new subject becomes the validation data-set and all other subjects become the training set.

4 Experiment

4.1 Data Collection

Data is collected from wearable sensors and is analysed and labeled using an open source video analysis program, ELAN, over a video of the basketball session that the subjects had taken part in. This includes, game, basic locomotion and basketball specific activities, as displayed in figures below.

To illustrate the distribution of both locomotion, the highest number of all activities is the dribbling (the not labelled does not count), then the lower number is shot, pass, layup, and rebound subsequently (see figure 1). Appendix A.0.1 shows the visualisation of the raw sensor data collected from wearables.



(a) Locomotion

(b) Basketball

Figure 1: Label counts.

4.2 Prepossessing

Given that data labelling for this project is done manually, it is prone to errors during the labelling process. It is essential to clean, create and structure the data for it to be trainable. The steps in preprocessing include the following.

- Combine data: Regarding the separated sensor on each wrist has individual data set, the training network need all players data, to fit the requirement. The combination is the first step of pre-processing.
- Cleaning: The existance of NaN values, blank data and data without labels can undermine the performance during the training part of the project, hence they need to be cleaned. Some missing values are interpolated in the data frame.
- Scaling: The range of number that allow to go through training network is between -1 and 1. The data can be simply rated all data in every different label of activity. But the more efficient strategy is to split each activity out from main resource, after that perform the scaling for each activity.
- Splitting and applying sliding window: The data is divided into smaller windows in order to be able to feed it to our neural network. As a result, sliding window method is used to split data into manageable chunks for our input layer. The window size being considered is 50 with a sampling rate of 50 for each each activity

4.3 Network Architecture

The network and the training that this project applies, come from Marius Bock, PhD in Ubiquitous Computing and Computer Vision of University of Siegen. Basically of starting network must be activated function, relu, following with four convolution layers, one layer LSTM, before the last layer is dropout, and then linear classification layer. The network architecture is presented in figure 2.

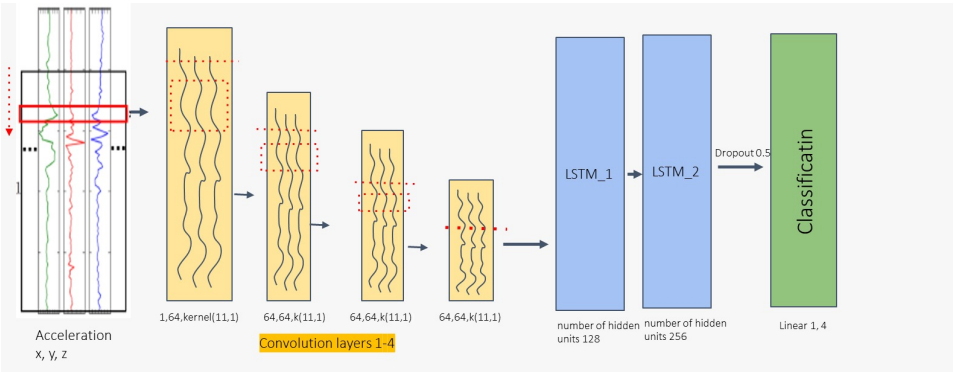


Figure 2: Network architecture[1]

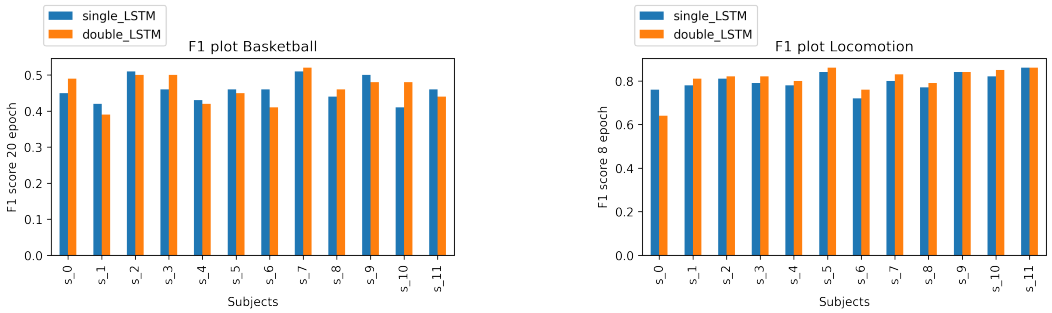
4.4 Training and Cross Validation

Training is terminated using Adams optimization algorithm, the completed loss function was Cross Entropy loss and therefore the Actual predictions were found using Soft Max probability. Furthermore the Leave One Subject Out Cross Validation technique was implemented. First, the data is divided into train set as the drill, game and/or warm up sessions of 11 subjects and validation set as drill, game and/or warm up session of the current subject. Second, sliding window is applied to this new training and validation data, then the network is trained for some epoch, furthermore repeating this procedure for all other subjects, and training the network each time for certain epoch the LOSO cross validation technique is achieved.

4.5 Results

4.5.1 F1 Score Per Subject

We did some experiment on our model by making changes to the network architecture, primarily we tried to implement double LSTM layer to make predictions and it seems that Single layer LSTM would perform slightly better as compared to double layer in case of basketball label, however in case of locomotion label both have similar score. Figure 3 presents the results obtained. The values are tabulated in section A.0.3.

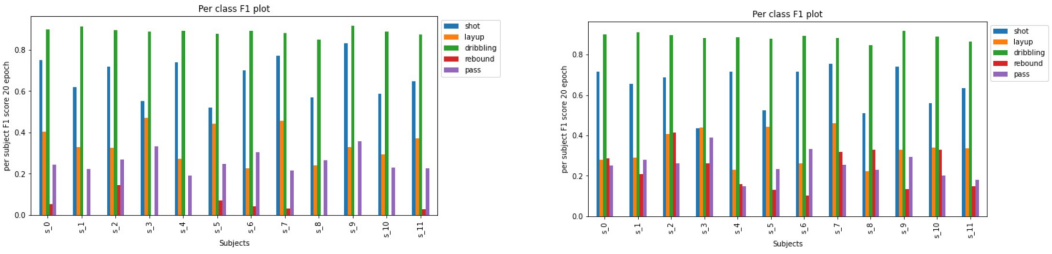


(a) Basketball (b) Locomotion

Figure 3: F1 scores for single and double LSTM models.

4.5.2 Augmented Basketball vs Non Augmented Basketball

By observing the label distribution of the basketball course it can be seen that the data for course Rebound is relatively less than other activity therefore because of this reason our network cannot predict this activity effectively we make data augmentation by doubling the data for rebound by adding some noise to the rebound data and creating augmented data which would be included in our dataset. By implementing this small experimental change we got better prediction for Rebound activity. Figure 4 presents the results obtained. The values are tabulated in section A.0.2.

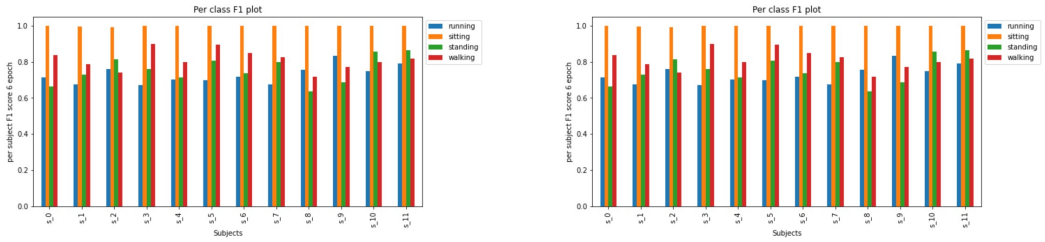


(a) Non augmented data (b) Augmented data

Figure 4: F1 scores for non augmented and augmented basketball data

4.5.3 F1 Score Per class

If we include Batch Normalisation layer between the Convolution layer then we get significant improvement in performance, for basketball data with data augmentation, however for locomotion data set there is not much difference in the result.

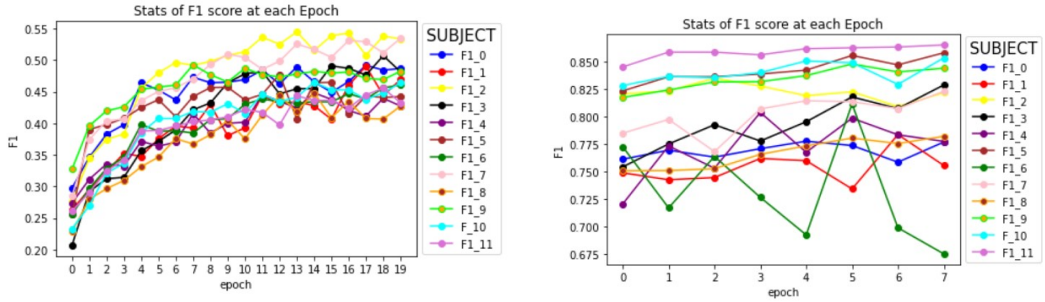


(a) With batch normalisation (b) Without batch normalisation

Figure 5: F1 scores for locomotion with and without batch normalisation

4.5.4 F1 score at each stage

One observation could be made while cross validating locomotion activity, despite it reaches convergence quite rapidly after few epoch the values starts fluctuating so we decided to run few epoch. However basketball data has steady rise in the stats and starts getting stagnant at about 20 epoch. Figure 6 presents the obtained results.



(a) Basketball (b) Locomotion

Figure 6: F1 scores at each training epoch

5 Conclusion

In this report, a collection of sensor data from a group of subjects playing basketball were labelled based on classes, and processed. The data was trained using Deep convoluted LSTM, and classes for dribbling, layup, passing, shot and rebounds were classified. The LOSO cross validation technique, was deployed. We propose to use single layer LSTM with one batch normalisation layer between the convolution layer and if the data set is less for certain class, for example in our case for rebound, data augmentation would rather generate better results. Furthermore while prepossessing activity wise scaling of the data has significantly improving effect on the final result. Kernel size of convolution layers induces higher training efficiency, but keep in mind that sliding window size need to correspond to kernel size. Further work that we propose is to perform more augmentation to the data to increase the training example and also including a pooling layer in between convolution layer.

References

[1] Marius Bock, Alexander Hölzemann, Michael Moeller, and Kristof Van Laerhoven. Improving deep learning for HAR with shallow lstms. *CoRR*, abs/2108.00702, 2021.

[2] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys*, 46, 01 2013.

[3] Abdulmajid Murad and Jae-Young Pyun. Deep recurrent neural networks for human activity recognition. *Sensors*, 17:2556, 11 2017.

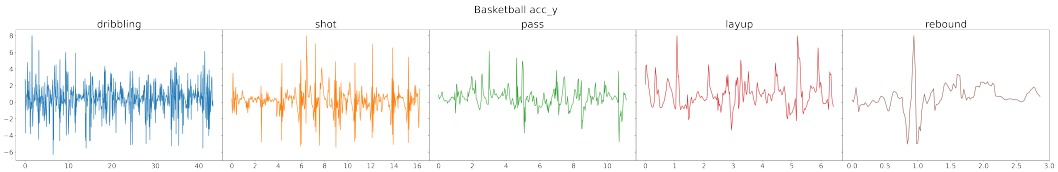
[4] Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1), 2016.

[5] Schalk Wilhelm Pienaar and Reza ÖMalekian. Human activity recognition using lstm-rnn deep neural network architecture. *CoRR*, 1905.00599, 2018.

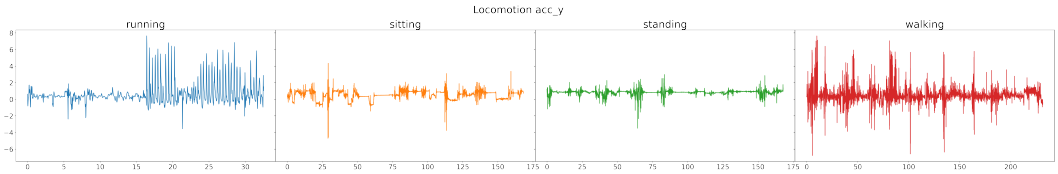
[6] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*, 2020.

A Appendix

A.0.1 Visualisation of raw sensor data



(a) Basketball



(b) Locomotion

Figure 7: Visualisation of sensor data for activity classes

A.0.2 Tabulated data of F1 scores for non augmented and augmented basketball data

Subjects	Shot	Layup	Dribbling	Rebound	Pass
S_0	0.749507	0.403101	0.897844	0.051282	0.244604.
S_1	0.619565	0.328205	0.913197	0.000000	0.222222
S_2	0.719057	0.325991	0.896958	0.145455	0.267516
S_3	0.551402	0.470588	0.889481	0.000000	0.332155
S_4	0.741507	0.270968	0.890602	0.000000	0.191358
S_5	0.519417	0.442105	0.876967	0.068966	0.245989
S_6	0.699809	0.224719	0.891892	0.043478	0.302839
S_7	0.772894.	0.457831	0.882483	0.030769	0.213836
S_8	0.570552	0.239521	0.849091	0.000000	0.266212
S_9	0.830769	0.329670	0.917347	0.000000	0.358209
S_10	0.587912	0.293706	0.887931	0.000000	0.228571
S_11	0.646967	0.371041	0.872968	0.027397	0.226562

Table 1: Data without augmentation

Subjects	Shot	Layup	Dribbling	Rebound	Pass
S_0	0.716469	0.280374	0.900452	0.285714	0.250000
S_1	0.654545	0.290698	0.911775	0.207407	0.280543
S_2	0.685714	0.406015	0.897184	0.414894	0.262626
S_3	0.435000	0.437878	0.883690	0.261538	0.388489
S_4	0.714487	0.229730	0.884831	0.160000	0.146789
S_5	0.524390	0.443350	0.877944	0.132075	0.232558
S_6	0.714012	0.262911	0.891188	0.103448	0.331361
S_7	0.754221	0.462006	0.881281	0.318182	0.256250
S_8	0.508961	0.223684	0.845070	0.328125	0.229730
S_9	0.738916	0.327434	0.917929	0.132353	0.294416
S_10	0.559194	0.339623	0.890562	0.327586	0.201923
S_11	0.635220	0.334928	0.864848	0.148571	0.178404

Table 2: Data with augmentation

A.0.3 Tabulated data of F1 scores for single and double layer LSTM implementation results

Subjects	Single_LSTM	Double_LSTM
S_0	0.48	0.49
S_1	0.46	0.39
S_2	0.53	0.50
S_3	0.48	0.50
S_4	0.42	0.42
S_5	0.44	0.45
S_6	0.46	0.41
S_7	0.53	0.52
S_8	0.42	0.46
S_9	0.48	0.48
S_10	0.46	0.48
S_11	0.43	0.44

Table 3: Comparision of results for Single v Double layer LSTM for Basketball

Subjects	Single_LSTM	Double_LSTM
S_0	0.76	0.64
S_1	0.78	0.81
S_2	0.81	0.82
S_3	0.79	0.82
S_4	0.78	0.80
S_5	0.84	0.86
S_6	0.72	0.76
S_7	0.80	0.83
S_8	0.77	0.79
S_9	0.84	0.84
S_10	0.82	0.85
S_11	0.86	0.86

Table 4: Comparision of results for Single v Double layer LSTM for Locomotion