Human Activity Recognition Using CNN & LSTM

Chamani Shiranthika
Department of Electrical Engineering
National Taipei University
New Taipei City, Taiwan
chamanijks2@gmail.com

Nilantha Premakumara
Department of Electrical Engineering
National Taipei University
New Taipei City, Taiwan
nilaprem108@gmail.com

Huei-Ling Chiu
School of Gerontology Health Management
Taipei Medical University
Taipei, Taiwan
reiko@tmu.edu.tw

Hooman Samani
Department of Electrical Engineering
National Taipei University
New Taipei City, Taiwan
hooman@mail.ntpu.edu.tw

Chathurangi Shyalika
Faculty of Information Technology
University of Moratuwa
Katubedda, Sri Lanka
chathurangijks@gmail.com

Chan-Yun Yang
Department of Electrical Engineering
National Taipei University
New Taipei City, Taiwan
cyyang@mail.ntpu.edu.tw

***Abstract* — In identifying objects, understanding the world, analyzing time series and predicting future sequences, the recent developments in Artificial Intelligence (AI) have made human beings more inclined towards novel research goals. There is a growing interest in Recurrent Neural Networks (RNN) by AI researchers today, which includes major applications in the fields of speech recognition, language modeling, video processing and time series analysis. Recognition of Human Behavior or the Human Activity Recognition (HAR) is one of the difficult issues in this wonderful AI field that seeks answers. As an assistive technology combined with innovations such as the Internet of Things (IoT), it can be primarily used for eldercare and childcare. HAR also covers a broad variety of real-life applications, ranging from healthcare to personal fitness, gaming, military applications, security fields, etc. HAR can be achieved with sensors, images, smartphones or videos where the advancement of Human Computer Interaction (HCI) technology has become more popular for capturing behaviors using sensors such as accelerometers and gyroscopes. This paper introduces an approach that uses CNN and Long Short-Term Memory (LSTM) to predict human behaviors on the basis of the WISDM dataset.**

Keywords—Human Activity Recognition, Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM)

I. INTRODUCTION

Human Activity Recognition is the process of defining, assessing and understanding what sort of acts and objectives one or more agents or individuals will perform. Decisions would be made on the basis of their past behavioral acts. In his or her day-to-day routine, a typical human may perform major activities such as walking, running, sitting, standing, laying, walking-upstairs, walking-downstairs, etc. If HAR could be combined with IoT technologies and means, defining and evaluating different human behaviors will bring out some smart solutions relevant to childcare and eldercare areas.[1]. For example, assuming a situation in which a child is held in a day care center, parents have gone to work and need to verify what their child is actually doing right now or is healthy at this time, this HAR may be used as a measure to predict the actions of the child. Even, in the case of elderly people watching guardians or caretakers, by avoiding certain acts elderly people prefer to do, this technology may be used to create a safer atmosphere for them. Thus, HAR could have enticing solutions for real life human problems , ranging from personal fitness to gaming, security fields, the healthcare industry and even more.

CNN and RNN architectures have become more predominant with the recent emerging trend of Deep Learning, and the application of Deep Learning models to train time series of inertial sensor data is still under investigation by researchers[2],[3]. Deep learning models such as CNN and RNN concentrate on a data-driven approach to sequential information to learn discriminatory characteristics from raw sensor data. Human activities normally are measured with sensors either be external or wearable such as accelerometers and gyroscopes. Accelerometer data measures people's speed of doing things and gyroscope data measures the angular velocity of the actions. Then, since these sensors provide a large dataset development, it will be an important task to process and analyze the entire dataset of correct automated systems. In this context, to avoid the data analysis problems associated with the system, HAR systems will have an important task. A feature vector will be extracted from this large raw data collected, and an activity recognition model based on the feature vector will be generated at the end of the learning algorithms[1]. Therefore, it is essential to select a well-trained, efficient model to grasp the maximum accuracy of the recognition process.

The rest of this paper is organized as follows. Section 2 gives an overview of the dataset used, LSTM architecture, CNN LSTM architecture and the HAR paradigm. Section 3 gives our implications and methodology used to implement the system. Experiment results are shown under the section 4. Finally, paper concludes giving the conclusion cited and expecting future works associated with the learnings and results of the overall research.

II. LITERATURE REVIEW

A. WISDM Dataset

The dataset used in the experiment is the standard WISDM dataset, which is also known as Smartphone and smartwatch activity, and Biometrics dataset. It contains accelerometer and gyroscope time series sensor data collected from a smartphone and smartwatch as 51 test subjects performing 18 activities for 3 minutes each with a 20Hz sampling rate. 36 users have been participated in the experiment. This is available and can be

978-1-6654-1475-3/20/$31.00 ©2020 IEEE

downloaded from the UCI machine-learning repository. The size of the dataset is 1,098,207 which contains data relate to 6 attributes as walking, jogging, upstairs, downstairs, sitting and standing. The columns are as user, activity, timestamp, x-acceleration, y-acceleration and z-acceleration. Originally this is an unbalanced dataset where walking contains 38.6% of data, Jogging contains 31.2% of data, upstairs include 11.2% data and Downstairs, Sitting and Standing contains 9.1%, 5.5% 4.4% of the data respectively.

B. CNN Architecture

The basic structure and the functionality of the visual cortex of the human brain have inspired CNN architecture. Instead of all the neurons in a fully connected layer, a neuron in a layer will only be connected to a tiny region of the layer before it. CNN is also referred to as ConvNet, too. CNN are not fully connected. The input layer, output layer and several hidden layers that can be the convolution layer, relu layer, pooling layer and finally the fully linked layer are included in the architecture of the traditional CNN. The final convolution often requires backpropagation in order to efficiently converge the measurement error and weight the end product correctly. The aim of the convolution operation would be to extract the high-level characteristics that seek to provide a more important and delicate link between the input and output of the classification.

C. LSTM Architecture

LSTM is the kind of RNN that is capable of learning and remembering very long-term dependencies over long sequences of input information. Thus, for time series analysis problems, LSTM is widely used. Among its recurrent modules, LSTM doesn't use activation functions. Values stored there are not updated. During training, LSTM does not have the vanishing gradient problem either. [5],[6].

Usually LSTM are implemented in 'blocks' or cell and blocks have 3 or 4 gates such as input gate, output gate or forget gate *etc* (Fig. 1). To deal with the vanishing gradient problem, LSTM uses the principle of gating[5]. Over various time periods, the cell is capable of recalling values.[5]. The cell is capable of remembering values over different time intervals.

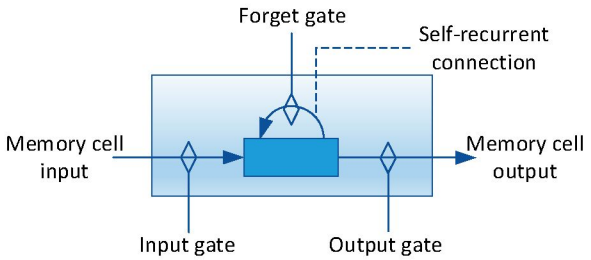


Fig. 1. Illustration of an LSTM memory cell [7].

Each of these cells are considered as an artificial neuron with an activation function of a weighted sum of the current data x_t, a hidden state h_{t-1} from the previous time step, and any bias b (Fig. 2).

The benefit of using LSTMs for sequence classification is that they can learn from the raw time series data directly, and in turn do not require domain expertise to manually engineer input features. The model can support multiple parallel sequences of input data, such as each axis of the accelerometer and gyroscope data. The model learns to extract features from sequences of observations and how to map the internal features to different activity types.

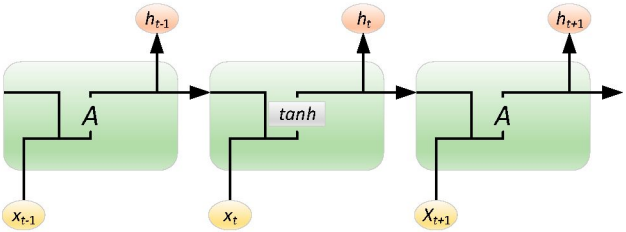


Fig. 2. The repeating module in a standard RNN which has a single layer.

D. Human Activity Recognition with CNN and LSTM

HAR has been extensively a wide research area in the past decade. HAR on smart phone data and various sensor data has been evolving rapidly around many multiple applications which benefit to human mankind. Advances in the area of mobile sensing enable users to quantify their sleep and exercise patters, monitor personal commute behaviors, track their emotional state or even any kind of human activity a person is involving in [5]. Statistical learning methods have also been used to tackle with the activity recognition problems. Some of them are Naïve Bayes, K-Nearest Neighbor (KNN) which has been used to recognize seven motions such as walking, running and jumping etc. However, they have required expert knowledge to design the features and systems have become more heuristic [10].

Literature proves many successful implementations on HAR using CNN and LSTM models. In [1] researchers describe a predictive model which use the UCI-HAR dataset for CNN layers for feature extraction on input data combined with LSTM to support sequence prediction. Also, they present an approach which uses convolutions directly as part of reading input into the LSTM units itself. CNN LSTM model have achieved an overall accuracy of 90.6% while Convolutional LSTM presented an accuracy of about 90%. [3] presents a methodology which uses CNN LSTM where the output of convolution is set as the input to LSTM. Convolution has been done on different time windows and via a "Time Distributed layer" that uses for time series analysis. Some probabilistic models for behavior prediction has also been developed applying LSTM network for behavior modeling [8].

In [4] researchers have implemented and improved a stacked LSTM architecture for the feature-free classification of activities using both accelerometer and gyroscope signals as the raw data input. There an approach using CNN and LSTM has been carried out for the human activity recognition from inertial sensor time series using batch normalized deep LSTM recurrent networks is presented. LSTM model is a multi-stacked architecture LSTM network for multi class HAR classification.

In literature, the use of basic classification techniques for HAR is also can be seen. Firstly, logic based classification algorithms such as decision trees have shown an accuracy of 98.7% in applications. Perceptron based algorithms show an accuracy of 89%. Alternatively, statistical algorithms such as

Naïve Bayes and Bayesian networks have shown an accuracy of 98% and 90.57% respectively. K-Nearest neighbor have had an average accuracy of 99.25% and 90.61%. Lastly SVM show an accuracy of 97.5% [1].

HAR systems also faces many challenges, such as large variability of a given action, similarity between classes, time consumption, and the high proportion of null values [9]. All of these challenges have led researchers to develop representation methods of systematic features and efficient recognition methods to effectively solve these problems. In [10] researchers proposed deep convolutional network with utilization of CNN and LSTM. This paper took advantage of LSTM to solve sequential human activity recognition problem and achieved a good precision. But the complex network framework suffered from low efficiency and can hardly meet real-time requirements in practice applications.

III. Methodology

With the literature review been conducted, it was revealed that the Deep Learning Models have been widely used resulting better scales of accuracy and to serve the Human Activity Recognition process. In addition, the neural network's robust nature, generalizable capability and scalability has inspired many researchers to apply deep learning in their machine learning models.

In our approach, the implementations were carried out under two main areas. First, we developed a basic CNN model for the activity recognition. Second, we developed a LSTM network model for our dataset. As the third step, we will be developing a CNN and LSTM hybrid model for the classification and the prediction of the six activities. Finally, we will propose a ConvLSTM model which is a further extension of the CNN LSTM model to perform the convolutions of the CNN as part of the LSTM. In this paper we basically focused on the CNN and LSTM models seperately. Prior to developing the models, we carried a comprehensive detailed exploratory data analysis of the WISDM dataset.

A. Exploratory Data Analysis of Dataset

Exploratory data analysis (EDA) is the approach of analyzing datasets to summarize the main characteristics presented in the data. Abbreviatedly, it is seeing what our data can tell us with discovering patterns with data. EDA is the backbone of any data science project which helps to understand the distribution of data which will be needed for better classification and prediction.

First, WISDM dataset was loaded in to Jupyter notebook environment. Here, several python open source libraries have been employed in the EDA analysis, including Pandas and Sklearn with various data processing functions [11]. With the help of the Pandas library, records with missing values were removed. Figure 3 illustrates the count distribution of recorded activities per person after the missing-value removal.

Next, the count of each activity with respect to the six different activities were plotted as shown in the following pie chart (Fig. 4). This pie chart clearly shows the unbalanced nature presented in the raw WISDM dataset which would potentially cause for irregularities during the data classification and the prediction process.

A balanced dataset would always be better in a perfect classification and prediction process because we can assure each of the representing classes would hold the same probability to occur without any biasness. Thus, WISDM dataset was balanced by selecting the same amount of data rows for each of the 6 activities which is graphically represented as the next pie chart (Fig. 5).

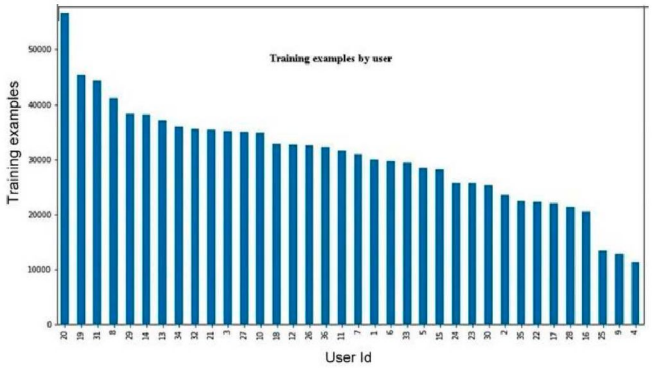


Fig. 3. Count of activities per person.

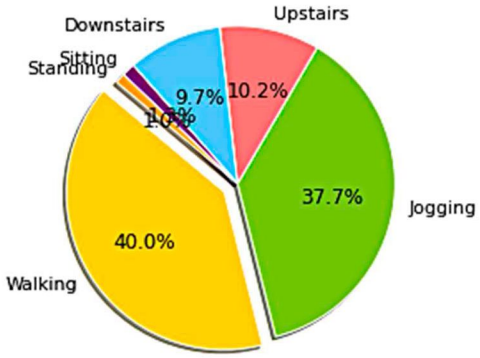


Fig. 4. Pie chart of record distribution by activity type with unbalanced data.

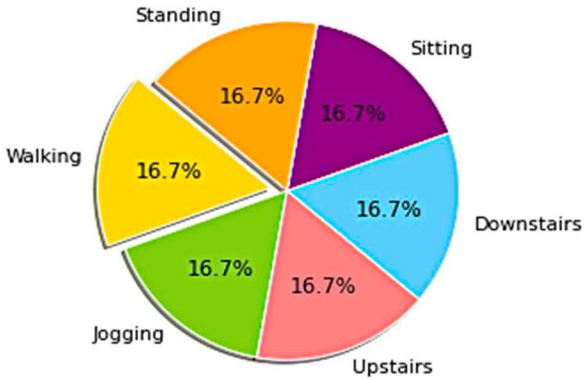
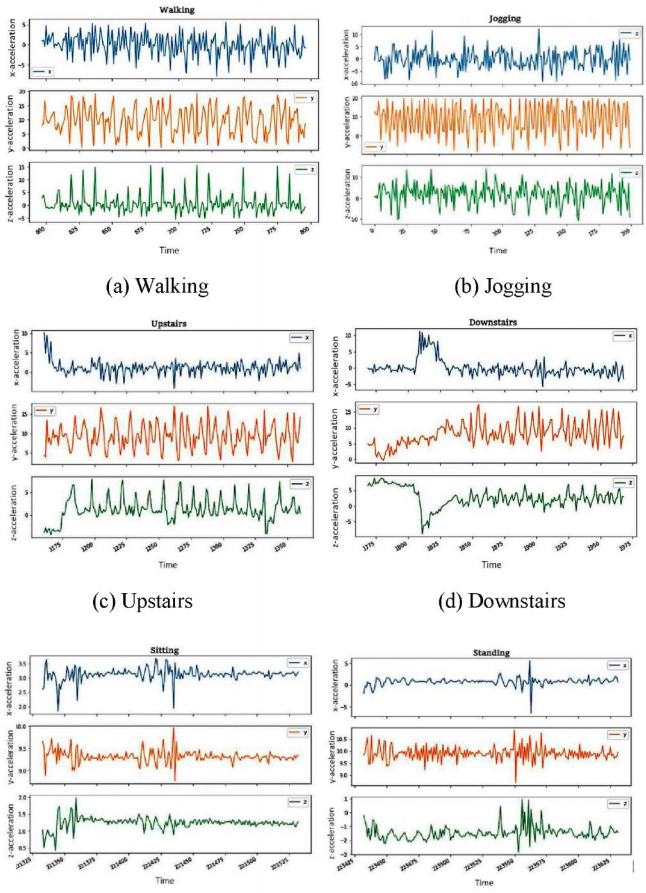


Fig. 5. Pie chart of record distribution by activity type with balanced data.

Next, in order to identify the nature of the signals relating to x, y and z axes, the accelerometer data was plotted with respect to the 6 activities separately for their first 200 entries as shown in the following plots. With a clear visual inspection of the following plots, the differences in each axis of the signals across each activity can be identified well. It is seen that the nonstationary activities like walking, jogging, upstairs

and downstairs have multiple variations with the signals while stationary activities, like sitting and standing, compasses only quite small amount of variations in their accelerometer signals, as those shown in Fig. 6.

The activity column which is a categorical variable in the dataset was then converted in to the numerical format. For this purpose, the LabelEncoder function from the Sklearn library was used for preprocessing. In the process of feature scaling, all the features were scaled to be within the same range, which would guarantee the value manipulations of every features equivalent and reweight naturally the prediction model by real dependency of the corresponding relevance of the features. Here, the Sklearn's StandardScaler function, which scale each feature by its maximum absolute value, was used for the scaling.



(a) Walking (b) Jogging

(c) Upstairs (d) Downstairs

(e) Sitting (f) Standing

Fig. 6. Variation nature of the signals for the six activities.

Finally, the data will need to be prepared in a format required by the designate models. For this purpose, fixed sized frame segments were created from the raw signals. The procedure would generate indexes as specified by a fixed size of steps moving over a thread of signal. The fixed step-size was parameterized with 20 for this study. The frame size used is 80 (step-size x 4), which equals to 4 seconds of data, *i.e.*, elementary sample are created in 4 seconds per segment. The label (activity) for each segment is selected by the most frequent class label or generally the mode presented in that window accordingly. The resulted dataset in the desired format is then split with an 8:2 ratio for training and testing datasets, respectively.

B. CNN Architecture

The CNN model was defined as having two CNN hidden layers. Each of them are followed by two dropout layers of 0.5 in order to reduce overfitting of the model to the training data. Then a dense fully connected layer is used to interpret the features extracted by the CNN hidden layers. Finally, a dense layer with the softmax activation function was added as the final layer to make predictions (Table I).

The sparse categorical cross entropy loss function will be used as the loss function and the efficient adam version of stochastic gradient descent was used to optimize the network with a learning rate of 0.001. CNN model was trained for 50 epochs and a batch size of 64 samples were used. After the model is fit, it was evaluated on the test dataset and the accuracy of the CNN model was obtained.

TABLE I. THE DIMENSIONAL SRUCTURE OF THE ADOPTED CNN MODEL.

Layer	Output Shape	Param #
Conv2D	None, 79, 2, 16	80
Dropout	None, 79, 2, 16	0
Conv2D	None, 78, 1, 32	2,080
Dropout	None, 78, 1, 32	0
Flatten	None, 2496	0
Dense	None, 64	159,808
Dropout	None, 64	0
Dense	None, 6	390
Total params: 162, 358 Trainable params: 162, 358 Non-trainable params: 0		

C. LSTM Architecture

The LSTM model was defined as having a single LSTM hidden layer. A dropout layer valuing 0.5 follows this. Then a dense fully connected layer is used to interpret the features extracted by the single LSTM hidden layer. Finally, a dense layer was added as the final layer to make predictions (Table II).

For the purpose of compiling and training the LSTM model, the same values for the loss function, optimizer, batch size and the number of epochs, which we used, in compiling and training the CNN model were used. After the model is fit, it was evaluated on the test dataset and the accuracy was obtained.

TABLE II. TABLE II. THE DIMENSIONAL SRUCTURE OF THE ADOPTED LSTM MODEL.

Layer	Output Shape	Param #
LSTM	None, 100	41600
Dropout	None, 100	0
Dense	None, 100	10100
Dense	None, 6	606
Total params: 52,306 Trainable params: 52,306 Non-trainable params: 0		

IV. Experimental Results

A. Results from CNN and LSTM Models

The implementation was realized under a Jupyter notebook environment of Google Colaboratory® by Python programming language. With the two model architectures described in the previous section, all the two models were compiled together with the sparse categorical cross entropy loss function and the Adam optimizer with a learning rate of 0.001. All the NN models was fitted for the training data and test data with a batch size of 64 and run for 50 epochs. The training accuracy was then plotted together with the validation accuracy varying the iterations for performance evaluation related to the two models (Fig. 7 and Fig. 8).

With respect to the CNN model, a training accuracy of 99.53% was achieved while the validation accuracy of 93.46% was simultaneously achieved as that shown in Fig. 7.

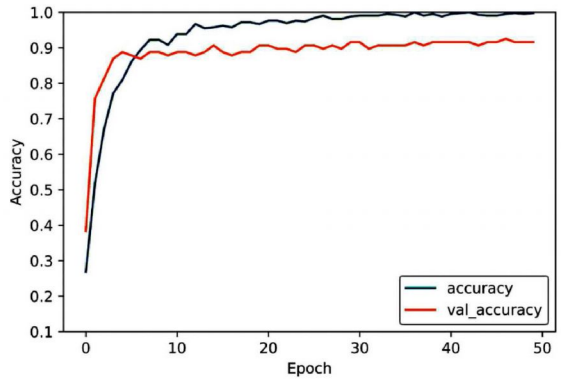


Fig. 7. Training and validation accuracies with the CNN model.

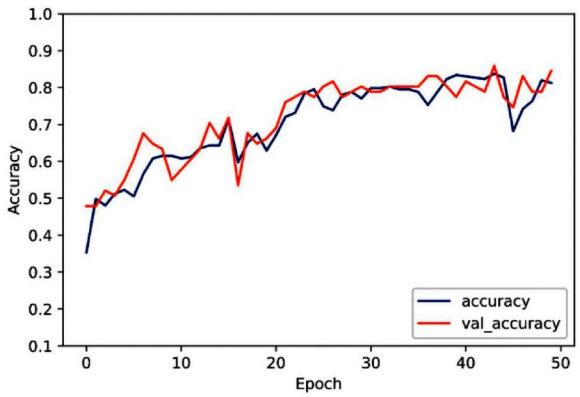


Fig. 8. Training and validation accuracies with the LSTM model.

Accompanying with the training and validation accuracies, the training and validation losses calculated during the procedure were also charted in a graph varying the number of iterations for all the two models. It is seen that both the training and validation losses are gradually decreasing with the iterations to converge to the approximation within respective precise ranges in both the models. The relative lower validation loss resulted in the two models guarantees that no overfitting happened to the converged models. Figures 9 and 10 show the training and validation loss resulted from the CNN model and LSTM model, respectively.

In addition to the accuracies, confusion matrices, which helps to graphically check the true label and the predicted label more comparatively, has also been constructed (Figs. 11 and 12). The confusion matrix contains information about the actual and predicted classifications done by a classification system and the performance of such systems is commonly evaluated using the data in the matrix. By checking on the test samples, the confusion matrices were charted as follows for the two models. Here the corresponding encoded values are 0 for walking, 1 for jogging, 2 for upstairs, 3 for downstairs, 4 for sitting and 5 for standing.

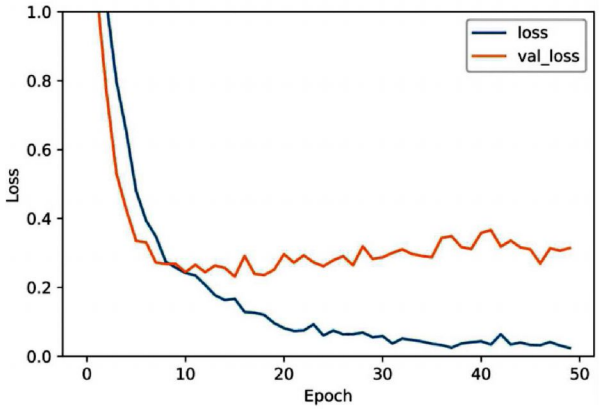


Fig. 9. Losses calculated during the iterative procedure for both training and validation with the CNN model.

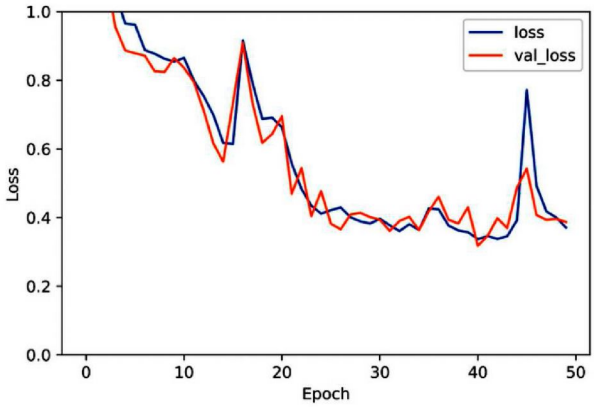


Fig. 10. Losses calculated during the iterative procedure for both training and validation with the LSTM model.

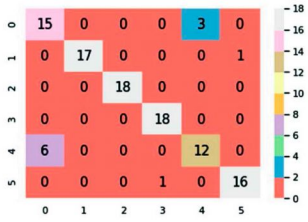


Fig. 11. Confusion matrix with CNN model.

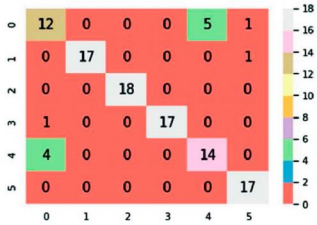


Fig. 12. Confusion matrix with LSTM model.

B. Comparison of Results from CNN and LSTM Models

We then constructed a classification report for the two models with the classification results achieved. There it was finally concluded that the CNN model shows far better accuracy terms compared with the LSTM model (Table III).

TABLE III. RESULTS COMPARISON BETWEEN CNN AND LSTM MODELS.

Model	CNN Model	LSTM Model
Accuracy	99.53%	84.71%
Average Precision	94%	77%
Average Recall	93%	79%
Average F1-score	93%	76%

V. CONCLUSION

In this paper, we have presented a CNN model and a LSTM model with 99.593% accuracy and 84.71% accuracy respectively for 6 daily life activities with the WISDM dataset. Use of Conv2D layers for CNN, Dropout regularization and using perfect model hyper parameters in the networks of the two models has made them fast and robust in terms of speed and accuracy. As further works, authors present an idea of using this presented Human Activity Recognition framework as a solution for a smart childcare or eldercare monitoring system based on IoT technologies. Also, it will be a perfect task if we can generate our own dataset with the use of appropriate sensors and applications for a defined number of frequent activities people are performing in day to day lives. This research area seems having multiple advanced applications with Deep Learning applications in near future. In addition, as future works authors suggest the application of reinforcement learning paradigm on the domain of activity recognition and classification.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support grants from Ministry of Science and Technology of Taiwan through its grant 108-2221-E-305-012, the National Taipei University through its grant 109-NTPU_ORDA-F-006 and the University System of Taipei Joint Research Program through its grant USTP-NTPU-TMU-109-01.

REFERENCES

[1] L. Alpoim, A. F. da Silva, and C. P. Santos, "Human Activity Recognition Systems: State of Art," in *2019 IEEE 6th Portuguese Meeting on Bioengineering (ENBENG)*, Lisbon, Portugal, Feb. 2019, pp. 1–4, doi: 10.1109/ENBENG.2019.8692468.

[2] S. Oniga and J. Suto, "Human activity recognition using neural networks," in Proceedings of the *2014 15th International Carpathian Control Conference (ICCC)*, Velke Karlovice, Czech Republic, May 2014, pp. 403–406, doi: 10.1109/CarpathianCC.2014.6843636.

[3] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," SIGKDD Explor. Newsl., vol. 12, no. 2, pp. 74–82, Mar. 2011, doi: 10.1145/1964897.1964918.

[4] A. Murad and J.-Y. Pyun, "Deep Recurrent Neural Networks for Human Activity Recognition," *Sensors*, vol. 17, no. 11, p. 2556, Nov. 2017, doi: 10.3390/s17112556.

[5] C. Jobanputra, J. Bavishi, and N. Doshi, "Human Activity Recognition: A Survey," *Procedia Computer Science*, vol. 155, pp. 698–703, 2019, doi: 10.1016/j.procs.2019.08.100.

[6] P. Kuppusamy and C. Harika, "Human Action Recognition using CNN and LSTM-RNN with Attention Model" *International Journal od Innovative Technology and Exploring Engineering(IJITEE),* vol.8, Issue 8, pp.1639-1643, 2019

[7] Y. Chen, K. Zhong, J. Zhang, Q. Sun, and X. Zhao, "LSTM Networks for Mobile Human Activity Recognition," presented at the *2016 International Conference on Artificial Intelligence: Technologies and Applications*, Bangkok, Thailand, 2016, doi: 10.2991/icaita-16.2016.13.

[8] C. Hofmann, C. Patschkowski, B. Haefner, and G. Lanza, "Machine Learning Based Activity Recognition To Identify Wasteful Activities In Production," *Procedia Manufacturing*, vol. 45, pp. 171–176, 2020, doi: 10.1016/j.promfg.2020.04.090.

[9] L. B. Marinho, A. H. de Souza Junior, and P. P. Rebouças Filho, "A New Approach to Human Activity Recognition Using Machine Learning Techniques," in *Intelligent Systems Design and Applications*, vol. 557, A. M. Madureira, A. Abraham, D. Gamboa, and P. Novais, Eds. Cham: Springer International Publishing, 2017, pp. 529–538.

[10] T. Zebin, M. Sperrin, N. Peek, and A. J. Casson, "Human activity recognition from inertial sensor time-series using batch normalized deep LSTM recurrent networks," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Honolulu, HI, Jul. 2018, pp. 1–4, doi: 10.1109/EMBC.2018.8513115.

[11] Wikipedia, "List of Python software," 2020. [Online]. Available: https://en.wikipedia.org/wiki/List_of_Python_software. [Accessed: 20- Sep- 2020].