# Cooking Activity Recognition using Deep Convolutional Bidirectional LSTM from Acceleration data


**Submitted By**

Rakib Rafshanjani

ID: 171-35-236

&

Bayazid Hasan Tamim

ID: 171-35-212


Submission Date: 17th June, 2021


Department Of Software Engineering
Daffodil International University

# APPROVAL

This thesis titled "**Cooking Activity Recognition using Deep Convolutional Bidirectional LSTM from Acceleration data**", submitted by **Rakib Rafshanjani, ID: 171-35-236 and Bayazid Hasan Tamim, ID: 171-35-212** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc in Software Engineering (SWE) and approved as to its style and contents.

## BOARD OF EXAMENERS

-------------------------------------------------

**Dr. Imran Mahmud**
**Associate Professor and Head**
Department of Software Engineering — **Chairman**
Faculty of Science and Information Technology
Daffodil International University

-------------------------------------------------

**Mr. SK. Fazlee Rabby**
**Lecturer**
Department of Software Engineering — **Internal Examiner 1**
Faculty of Science and Information Technology
Daffodil International University

-------------------------------------------------

**Nayeem Hasan**
**Lecturer (Senior Scale)**
Department of Software Engineering — **Internal Examiner 2**
Faculty of Science and Information Technology
Daffodil International University

-------------------------------------------------

**Md. Fazle Munim**
**Technology Lead, a2i**
Bangladesh Government — **External**

# THESIS DECLARATION

We hereby declare that we have taken this thesis under the supervision of **Shariful Islam, Lecturer, Department of Software Engineering, Daffodil International University.** We also declare that neither this thesis nor any part of this has been submitted elsewhere for award of any degree.


Rakib

……………………………………

**Rakib Rafshanjani**
**ID: 171-35-236**
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

Tamim

……………………………………

**Bayazid Hasan Tamim**
**ID: 171-35-212**
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University


Supervised by:


Shariful

………………………………………

**Mr. Shariful Islam**
**Lecturer**
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

# ACKNOWLEDGEMENT

The key focus of this thesis was on developing models that could be utilized to detect cooking activity. Without the aid and support of a large number of individuals, it would not have been achievable. We are very grateful to Daffodil International University for its guidance and constant monitoring by Shariful Islam sir, such as for providing us with the knowledge and help we necessary to finish this study. For additional guidance, I would like to express my appreciation to our honorable teachers Bkash Kumar Paul sir and SAM Matiur Rahman sir. We like to offer our respect to our parents, batch mates, and DIU members for their helpful cooperation and sympathy.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

CARC = Cooking Activity Recognition Challenge;

CNN = Convolutional Neural Network;

AR = Activity recognition;

ML = Machine learning;

DL = Deep learning;

AI = Artificial intelligence;

RNNs = Recurrent Neural Networks;

LSTM = Long short-term memory;

DCBL = Deep Convolutional Bidirectional LSTM;

Mo-cap = Motion capture;

Conv2D = Two-dimensional convolution;

ReLU = Rectified Linear Unit;

# ABSTRACT

The difficulty of identifying a body's behavior based on sensor data, such as an accelerometer in a smartphone, is known as activity recognition. It's among the most widely studied topics in the field of machine learning-based classification. Cooking Activity Recognition Challenge (CARC) asked participants to recognize food preparation using motion capture and acceleration sensors. Two smartphones, two wristbands, and motion-capturing equipment were used to collect three-axis (x, y, z) acceleration data and motion data for the CARC dataset. One of the most challenging difficulties to solve in this investigation was identifying complicated tasks as smaller activities that are part of larger activities. Using a Convolutional Neural Network (CNN) and a Bidirectional LSTM, we've built a deep learning approach that extracts dynamical data for macro and micro activity identification. The model we proposed for that kind of dataset has a classification accuracy of 83% for macro activity and 85.3% for micro activity, respectively.

# CHAPTER 1: INTRODUCTION

Human Activity Recognition is a popular study area right now. Because of the minimal cost of sensors and accelerometers, as well as improvements in machine learning, deep learning, and IoT [1].

One approach for obtaining activity recognition (AR) is to use machine learning (ML) techniques. However, machine learning may necessitate property extraction of features, which takes a long time and has low accuracy. The Deep Learning [2] (DL) paradigm, often known as deep neural networks, has recently become one of the most significant algorithms in machine learning. Picture recognition [3], object classification [4], image partitioning [5] are only a few examples. The ability for DL to automatically re-integrate a given feature into a fresh presentation gives it an edge. However, in order to scale efficiently with the model, DLs require a large amount of data with a lot of features [6, 7].

IoT and AI are combining to provide services and applications for individualized health diagnosis and treatment. Monitoring at home is one of these services that has gotten a lot of attention because it influenced aged care. These services keep track of what the individual is doing and send out alerts in the event of an emergency.

This investigation has been carried out to determine the cooking activities. But what's the point to identify kitchen works? Consider a situation where a cooking robot prepares your food at home. You have been able to observe what the robot was preparing at home. Smart devices will transmit alarms to an emergency station in the event of an accident while cooking. In addition, cooking is a multi-step process that includes steps such as "getting from the refrigerator," "cleaning the food," "mixing throughout the dish," and so on. Identifying the stages might be utilized to remind someone of a step that they have forgotten. It assists in making decisions such as what task a given individual should complete inside the kitchen to finish a meal preparation process. In the case of nursing, verifying that safety standards, such as hand washing at the appropriate times have been implemented.

The entire paper is arranged in different chapters. In Chapter 2, the related works are described in order to extract the fundamental things observed in previous research as well as in our framework. In Chapter 3, the dataset used for training and evaluating the models are comprehensively discussed, including the dataset's flaws and pre-processing to overcome those issues. The proposed approach, as well as the characteristics of the models that we created, are described in Chapter 4. With the help of multiple graphs and tables, we clarified the results and ultimate findings in Chapter 5. All of the findings and model efficiency are detailed in this section. In Chapter 6, a summary of the entire study is included as a conclusion.

# CHAPTER 2: RELATED WORK

Numerous investigations have been conducted regarding the activities. Long short-term memory (LSTM) networks, which are widely utilized in time series data analysis and are commonly combined with recurrent neural networks (RNNs) [8]. Zhu et al. [9] introduced an RNN network with an LSTM model capable of learning feature presentation from joint co-presence. These models were shown to be effective in detecting simple terms activity. However, they were not appropriate for dealing with extended datasets containing repeated structures.

Hossain et al. [10] proposed a traditional machine learning method using handcrafted features from the dataset. However, handcrafted features may not perform well with time-series data [11]. In addition, traditional machine learning algorithms are not suitable for recognizing complex activities with missing data [12].

Yoon et al. [13] proposed a multi-directional RNN that has been reduced missing data errors and increase the performance of the model by 35%. Swapnil et al. [14] suggested a deep convolutional network pipeline for detecting complicated activities with missing data handling. While the pipeline improved at macro processes, it was not significantly improved the classification of micro tasks. In addition, training the pipeline necessitates a large amount of data.

Atsuhiro et al. [15] introduced a convolutional LSTM that calculates loss using several loss functions. The pipeline's ultimate categorization was determined by a majority vote. Even though the model used a qualified majority approach, loss functions occasionally reduced the true output. As a result, the voting mechanism forecasted incorrect activity. Picard et al. [16] devised a system that used just mo-cap data to identify cooking actions. With training data, the model scored 95%, however, the validation score was much lower than the training score.

Various study has been done concerning macro and micro activity with some technical glitches in the above-related works. So, there is an adequate opportunity to design a model to properly identify kitchen micro and macro activity.

In this investigation, connectivity has been created with a convolutional layer and a bidirectional LSTM (long short-term memory) layer to develop a model for sensors in each body part termed Deep Convolutional Bidirectional LSTM (DCBL). As an input to the model, features have been separated into a 30-second window. To polish missing data and handle varying sample rates, a controlled time shift has been utilized. Binary classification loss functions have been utilized as loss functions for multi-label data assessment in both macro and micro activities. In addition, compare and contrast DCBL with various deep learning models, as well as analyze the final results.

# CHAPTER 3: DATASET DESCRIPTION

To conduct this investigation, we accessed data from the ABC Conference's Cooking Activity Recognition Challenge-2020. The cooking dataset has a total of thirty activities, containing three macro and ten micro tasks. The tasks are as shown:

- Macro activities – sandwich, fruit salad & cereal
- Micro activities – add, cut, mix, open, peel, pour, put, take, wash & other

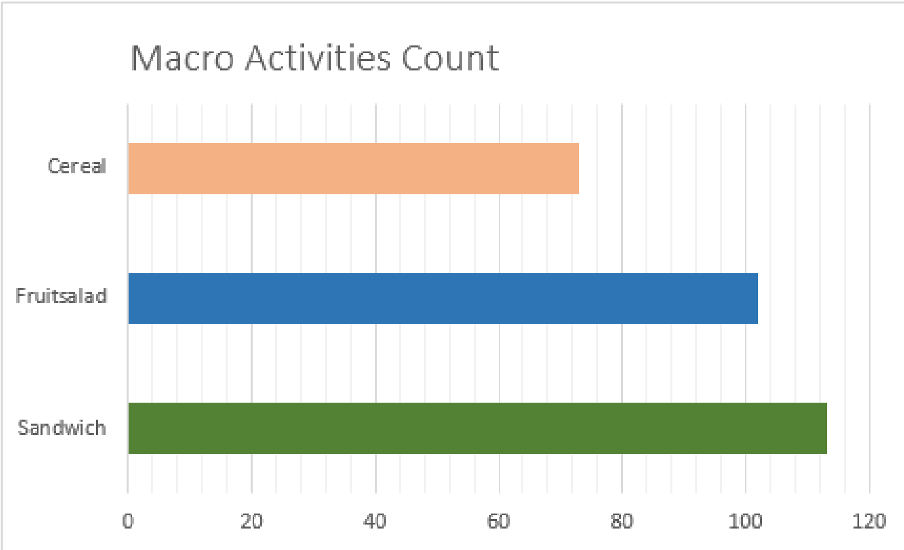


Figure 1: This statistic illustrates the number of files in the dataset that include each macro (cereal, sandwich, fruit salad) event. It indicates that there have been 73, 102, and 112 files for cereal, fruit salad, and sandwich, respectively, in the dataset.

The cooking dataset contains only numerical data. The accelerometer and motion capture sensors were used to acquire all the data. Four volunteers worked to record the sensor data, and they cooked three dishes (sandwich, cereal, fruit salad) five times each. Participants act as naturally as possible while preparing the dishes and they were given a script to follow in order to repeat all of the cooking steps as accurately as possible.

Two smartphones and two wristbands were used as sensors for collecting activities while cooking. Smartphones were worn on the right arm and left hip, while wristbands were worn on both wrists. Both the smartphone and the wristband used the same triaxial accelerometer for data collection. In addition, motion capture (mo-cap) sensors were implanted in 29 different parts of the body to collect data. Sensor data was segmented with a 30second window.

There were 288 files in total in the dataset. Each data file corresponds to a maximum of one macro activity and one or more micro activities. The accelerometer sample rate varied between 50 and 100 Hz, whereas the Mo-cap data were recorded at 100 Hz [14]. Form 288 files, 112 files pertain to sandwich activity, 102 files to fruit salad activity, and 73 files to cereal activity, as shown in Fig. 1.

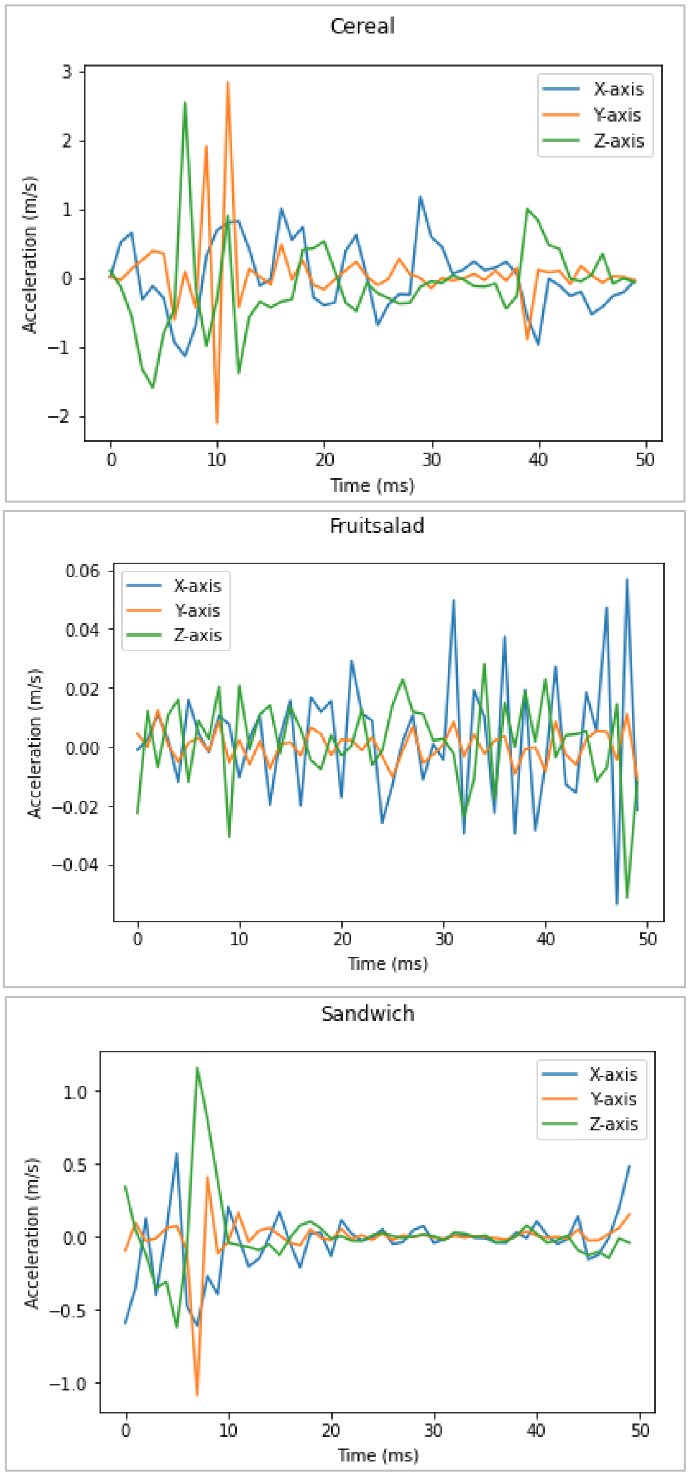


Figure 2: The graph represents the training data file for macro activities (cereal, fruit salad, and sandwich). This graph shows how actual accelerometer data was presented inside the dataset, showing all three axes (x, y, and z) concerning the sensor's time and acceleration.

The dataset we investigated has three major problems. The first was a data sequence that was out of sync, the second was missing values in data files, and the third was micro-actions themselves. Furthermore, some micro-actions appeared to be remarkably similar to others, making identification difficult.

## 3.1 Data Pre-processing

As previously stated, this dataset contains several significant flaws. In order to overcome those issues and get the maximum performance out of the model, some pre-processing has been conducted. The dataset had certain timestamps that were not in ascending order. As a solution, we added a time-shifter and rearranged the timestamps in ascending order to fix the timestamps problem. Furthermore, in various parts of the files, certain accelerometer numerical values were empty. Missing values were then filled using the random number assumption function.

A separate text file has been used to save labels such as sandwich, put, wash, and so on for the numerical input data files. To integrate the input files and labels, Python 3.9.0 was utilized. All micro labels (wash, put, cut, etc.) were encoded using one-hot encoding, whereas macro labels (cereal, sandwich, fruit salad) were encoded using Label encoding. The left wrist folder was dropped to improve training accuracy since it contained an average of 80% missing values. Furthermore, mo-cap data was ignored because it did not improve the model's performance. For train, test, and validation, the dataset was divided into three segments of 60:20:20 ratio.

# CHAPTER 4: METHODOLOGY

In this investigation, two deep neural architectures were virtually applied to the training process. Conv. Neural Networks and Deep Conv. Bidirectional LSTM.

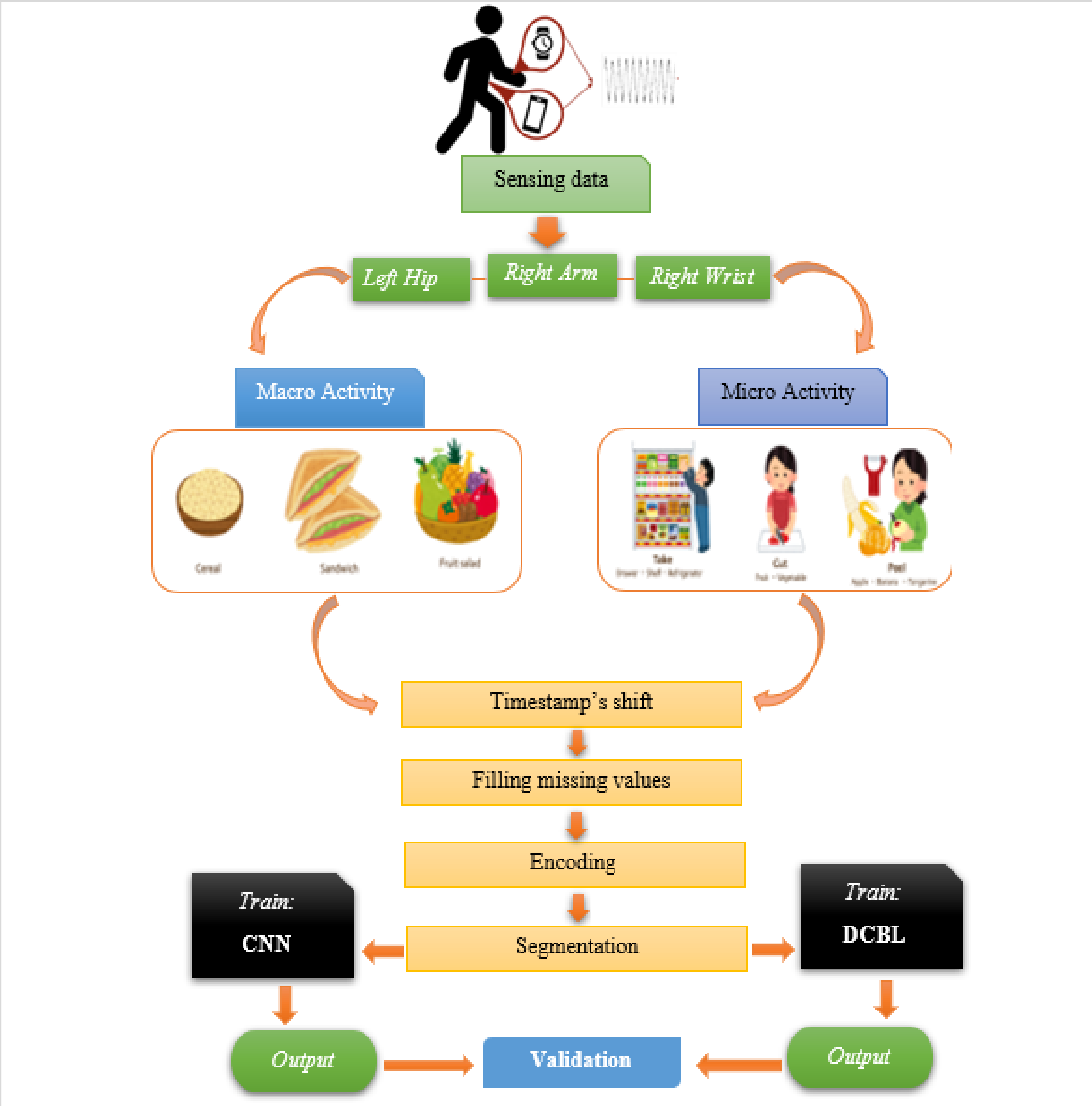


Figure 3: The proposed methodology's workflow has illustrated in this diagram.

The following methodology is based on two models (CNN and DCBL), as shown in Fig. 3. Although the design of these two models were relatively similar, the results were completely different. Sensor data from the left hip, right arm, and right wrist was utilized to test and evaluate the models.

The activities were divided into two segments. Preparing a meal was referred as macro activity (sandwich, fruit salad, and cereal), whereas minor actions for preparing those meals were referred to as micro activity (peel, wash, put etc.). For correcting timestamps in the dataset throughout the right sequence, the approach employed a time shift function. The function was implemented in the Python programming language. The random assumption function has been used to fill in the missing data. Because of the sensors malfunction while recording, the data folder on the left wrist does have a lot of missing values. As a result, data from the left wrist has been excluded.

Every data has to be in numerical format when feeding a model. However, the dataset's labels were in text format. Text labels were converted to numerical labels using encoding methods. For macro activity, label encoding was employed, while for micro activity, one-hot encoding was employed. The data has been segmented into 30 second windows and split into 60:20:20 ratios for train, test, and validation. The models that have been utilized for training and validating accuracies are discussed in detail in the section below.

## 4.1 Convolutional Neural Networks (CNN)

Two-dimensional convolutional (Conv2D) layer, pooling layer, and fully connected layers that can operate on helpful feature extractor and classifier are the primary features of deep conv. neural networks [17]. Deep CNNs are capable of learning characteristics without prior experience [18]. In addition, deep CNN plays an essential role in removing an overfitting problem and reducing training time [19].

| Layer(type) | Output Shape | Param# |
|---|---|---|
| conv2d_1 (Conv2D) | (None, 9, 500, 128) | 1408 |
| conv2d_2 (Conv2D) | (None, 9, 500, 128) | 163968 |
| batch_normalization_1 | (None, 9, 500, 128) | 512 |
| conv2d_3 (Conv2D) | (None, 9, 500, 128) | 163968 |
| batch_normalization_2 | (None, 9, 500, 128) | 512 |
| max_pooling2d_1 | (None, 9, 166, 128) | 0 |
| conv2d_4 (Conv2D) | (None, 9, 166, 128) | 163968 |
| conv2d_5 (Conv2D) | (None, 9, 166, 128) | 163968 |
| batch_normalization_3 | (None, 9, 166, 128) | 512 |
| max_pooling2d_2 | (None, 9, 83, 128) | 0 |
| Dropout_1 | (None, 9, 83, 128) | 0 |
| conv2d_6 (Conv2D) | (None, 9, 83, 128) | 163968 |
| batch_normalization_4 | (None, 9, 83, 128) | 512 |
| max_pooling2d_3 | (None, 9, 41, 128) | 0 |
| dropout_2 | (None, 9, 41, 128) | 0 |
| Flatten | (None, 47232) | 0 |
| dense_2 | (None, 64) | 3022912 |

| | | |
|---|---|---|
| batch_normalization_5 | (None, 64) | 256 |
| dropout_3 | (None, 64) | 0 |
| dense_3 | (None, 3) | 195 |

Table 1: The architecture of the CNN model, which has been utilized to train macro activities.

The design of implementing CNN architecture is shown in Table 1. The architecture usually employs 128 hidden layers in Conv2D, with kernel dimensions of (1, 10), stride size of (1, 1), batch size of 64 in each iteration. The use of ReLU helps to keep the exponential improvement contained within the computing required to run the neural network.

## 4.2 Deep Convolutional Bidirectional LSTM (DCBL)

Bidirectional LSTMs are a kind of LSTM that improves model performance on sequence classification tasks. DCBL inputs flow in two directions: backward to forwards and forwards to backward [20]. The LSTM provides a critical memory cell state that can save previously useful timestamp data [21].

| Layer(type) | Output Shape | Param# |
|---|---|---|
| conv2d_1 (Conv2D) | (None, 9, 500, 128) | 1408 |
| conv2d_2 (Conv2D) | (None, 9, 500, 128) | 163968 |
| batch_normalization_1 | (None, 9, 500, 128) | 512 |
| conv2d_3 (Conv2D) | (None, 9, 500, 128) | 163968 |
| batch_normalization_2 | (None, 9, 500, 128) | 512 |
| max_pooling2d_1 | (None, 9, 166, 128) | |
| conv2d_4 (Conv2D) | (None, 9, 166, 128) | 163968 |
| conv2d_5 (Conv2D) | (None, 9, 166, 128) | 163968 |
| batch_normalization_3 | (None, 9, 166, 128) | 512 |
| max_pooling2d_2 | (None, 9, 83, 128) | 0 |
| dropout_1 (Dropout) | (None, 9, 83, 128) | 0 |
| conv2d_6 (Conv2D) | (None, 9, 83, 128) | 163968 |
| batch_normalization_4 | (None, 9, 83, 128) | 512 |
| max_pooling2d_3 | (None, 9, 41, 128) | 0 |
| dropout_2 (Dropout) | (None, 9, 41, 128) | 0 |
| permute_1 (Permute) | (None, 41, 9, 128) | 0 |
| reshape_1 (Reshape) | (None, 41, 1152) | 0 |

| | | |
|---|---|---|
| bidirectional_1 | (None, 256) | 1311744 |
| dense_1 (Dense) | (None, 3) | 771 |

Table 2: The architecture of the DCBL model, that has been utilized to train macro activities.

The design of the elevated DCBL is shown in Table 2. With the termination of the bidirectional LSTM layer, then added deep Conv. neural networks to increase model accuracy, validation accuracy, and test accuracy. The classification of macro and micro categories is linked to dense layers or completely connected layers.

# CHAPTER 5: RESULT ANALYSIS

Deep CNN and DCBL were used to compile only the raw accelerometer data from the original dataset. The findings were examined from three sensor data points: the left hip, right wrist, and right arm, all at the same time, to assist for macro and micro activity categorization. The files were broken into three portions for training both architectures (train: validation: test). 60% of the data was used for training, 20% for validation, and the remaining 20% was used for testing.

To see if the trained model was overfitted, the acceleration data was checked using the cross-validation (CV) method. The train data has been organized in a random format (RF) in this CV strategy. This RF approach has worked on the principle of separating particular samples before splitting them into frames for model training. As a result, when the model was trained, the accuracy of the original samples for prediction was maintained. It has been extremely evident how well the trained models could predict and evaluate the accuracy for both macro and micro activity recognition individually using this method of cross-validation.

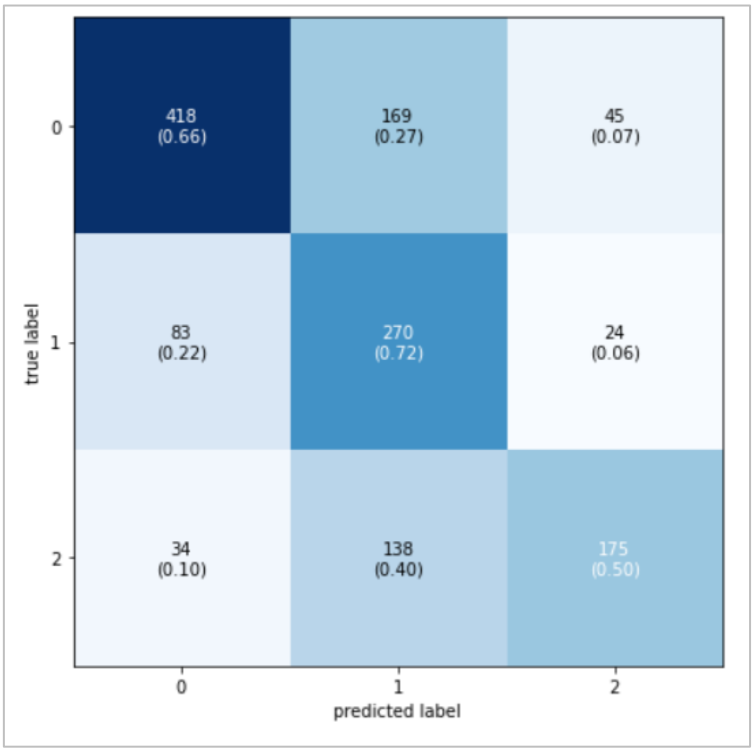


Figure 4: This is the CNN model's confusion matrix for macro activities. The actual labels are shown on the X-axis, while the anticipated labels are shown on the Y-axis. Furthermore, 0 indicates the recipe "sandwich," 1 indicates the recipe "fruit salad," and 2 indicates the recipe "cereal."

In table 3, a multi-label approach classifier was employed to detect and get the best accuracies for the macro activities "fruit-salad", "sandwich" and "cereal". To learn more, Table 4 uses a one-vs-all methodology to classify micro activities such as "add", "cut", "wash", "open", "put", "other", "pour", "peel", "mix", and "take". Categorical cross-entropy was applied as a loss function in compiling the model, and Adam was utilized as an optimizer because macro and micro activities were related with multiclass difficulties. Additionally, the Softmax activation function was implemented in the last layer to obtain multiple predictions.

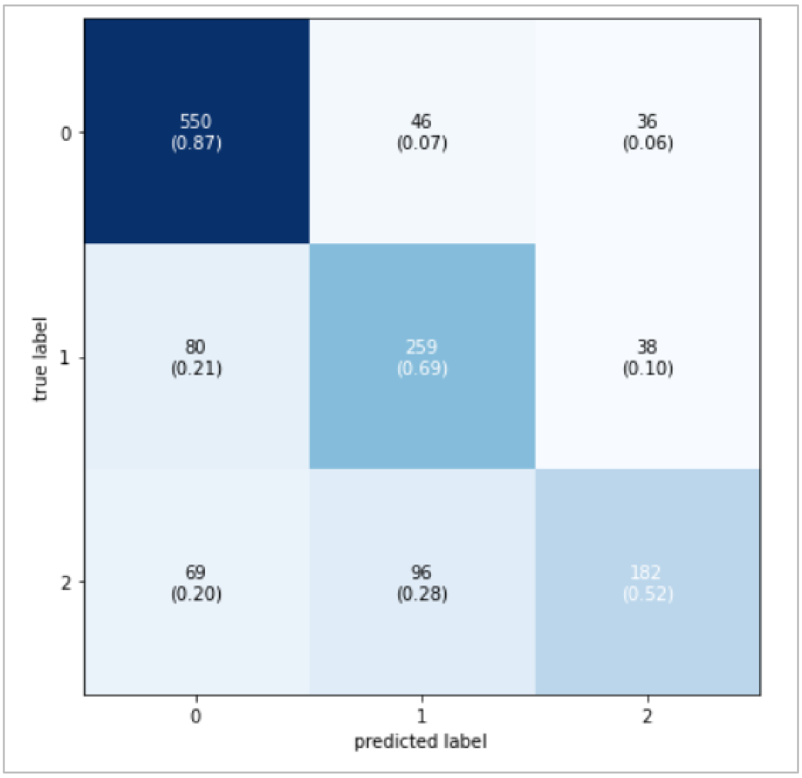


Figure 5: This is the DCBL model's confusion matrix for macro activities. The genuine labels are shown on the X-axis, while the anticipated labels are shown on the Y-axis. Furthermore, the numbers 0, 1, and 2 stands for a sandwich, fruit salad, and cereal, respectively.

The python language was utilized to train the two models. Keras, sklearn, and TensorFlow were some of the libraries included. The window was 10s in size, with a 2.40 GHz Intel 5-core CPU, 8GB of RAM, and a GPU of the NVIDIA GeForce 940MX.

### 5.1 Macro Activity Classification Accuracies

The result of the macro-activity classification is given in table 3. The trainable data reached the necessary macro activity categorization findings after thirty epochs. Table 1 demonstrates that the DCBL classifier model's macro activity classification result is substantially more accurate than the CNN classifier model.

| CNN | | | DCBL | | |
|---|---|---|---|---|---|
| Test | Train | Validation | Test | Train | Validation |
| 0.58 | 0.88 | 0.65 | 0.70 | 0.98 | 0.83 |

Table 3: This table illustrates the macro activities (cereal, fruit salad & sandwich) prediction accuracy for both the CNN and DCBL models.

Deep CNN and DCBL classifier comparison chart (Fig. 6) for macro activity clarification in a multi-label method. A deeper examination of Fig. 6 reveals that macro activity identification using DCBL outperformed Deep CNN.

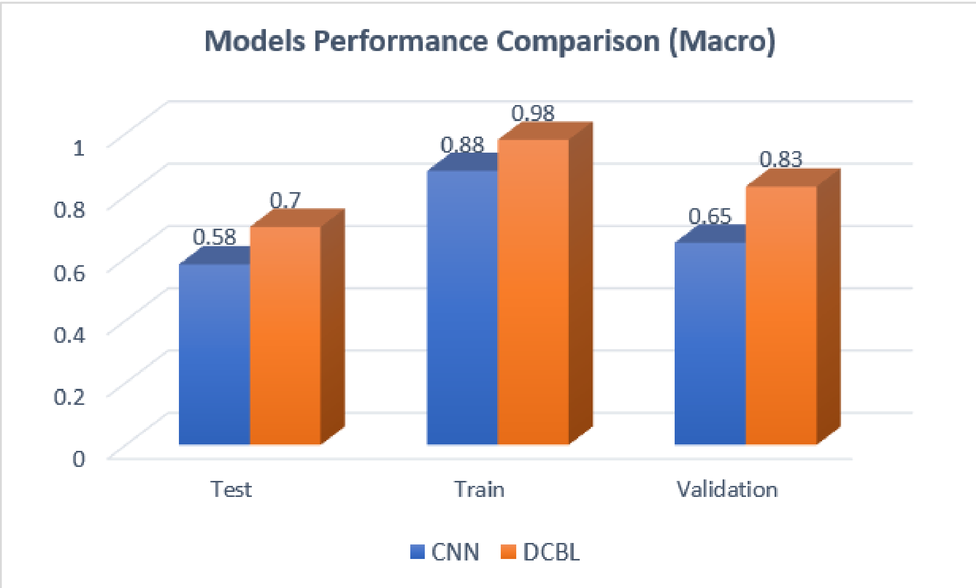


Figure 6: The accuracy of macro activities utilizing both CNN and DCBL has been compared in this graph. DCBL validation accuracy has 18% better than CNN validation, as shown in the graph. In addition, the colors blue and orange stand for CNN and DCBL, respectively.

## 5.2 Micro Activity Classification Accuracies

From table 2 after twenty epochs using the CNN method, micro activities classification was obtained average test accuracy of 0.81%, average train accuracy of 0.83%, average validation accuracy of 0.77%. The output shows an average test accuracy of 0.87 percent, an average train accuracy of 0.91, and an average validation accuracy of 0.85 percent using the DCBL approach. The DCBL's 0.06% mean test accuracy is greater than the CNN, 0.08% train, and 0.08% validation accuracy.

| Activity | CNN | | | DCBL | | |
|---|---|---|---|---|---|---|
| | Test | Train | Validation | Test | Train | Validation |
| Add | 0.94 | 0.93 | 0.94 | 0.94 | 0.95 | 0.96 |
| Cut | 0.74 | 0.68 | 0.70 | 0.90 | 1.00 | 0.86 |
| Mix | 0.92 | 0.94 | 0.92 | 0.92 | 0.95 | 0.94 |
| Open | 0.94 | 0.97 | 0.90 | 0.92 | 0.99 | 0.94 |
| Peel | 0.72 | 0.69 | 0.72 | 0.74 | 0.68 | 0.74 |
| Pour | 0.86 | 0.92 | 0.76 | 0.88 | 0.94 | 0.90 |
| Put | 0.62 | 0.66 | 0.48 | 0.86 | 0.92 | 0.75 |
| Take | 0.72 | 0.76 | 0.70 | 0.82 | 0.90 | 0.78 |
| Wash | 0.84 | 0.91 | 0.88 | 0.83 | 0.97 | 0.88 |
| Other | 0.82 | 0.85 | 0.74 | 0.90 | 0.84 | 0.78 |

Table 4: This table illustrates the micro activities prediction accuracy (train, test, and validation) for both the CNN and DCBL models.

Based on the differences in mean accuracy between the two models, conclude that the DCBL model provides more effective micro activity classification results than the CNN model.

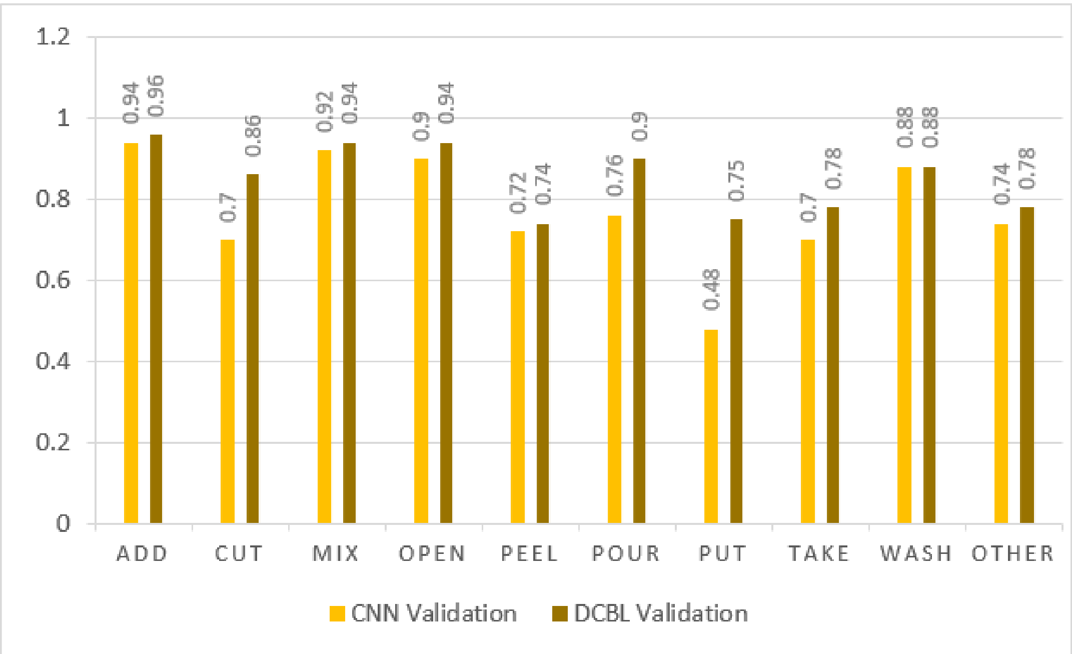


Figure 7: In this plot, the validation accuracy of micro activities using CNN and DCBL has been compared.

Using a CNN and DCBL classifier comparison graph (Fig. 7), micro activity clarification of the one-vs-all strategy is provided. Fig. 7 shows that one micro activity using DCBL classifier got the highest validation accuracy that is almost 96% as well as the lowest validation accuracy that is almost 74% which is ‘PEEL’ occupied. From all of the micro activities, the greatest validation accuracy 94% obtained ‘ADD' and the lowest validation accuracy 48% obtained ‘PUT' utilizing the deep CNN.

# CHAPTER 6: CONCLUSION

In this investigation, we showed how we used sophisticated deep learning approaches to classify macro and micro activities using accelerometer sensor data. We ran across a lot of issues at the time of training the models. In addition, the pre-processing stage for preparing training data was very difficult. They've previously been mentioned in the section on Dataset Descriptions. We overcome these challenges and attained an average accuracy of 85.3% in macro activity recognition and 83% in micro activity identification. In the future, these strategies will aid in the resolution of difficult activity difficulties. Additionally, these models will operate well with time-series datasets.

# REFERENCES


1. Jobanputra, C., Bavishi, J. and Doshi, N., 2019. Human activity recognition: A survey. *Procedia Computer Science*, *155*, pp.698-703.
2. LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. *nature*, *521*(7553), pp.436-444.
3. Tan, M. and Le, Q., 2019, May. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning* (pp. 6105-6114). PMLR.
4. Tan, M., Pang, R. and Le, Q.V., 2020. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10781-10790).
5. Liu, Z., Yang, C., Huang, J., Liu, S., Zhuo, Y. and Lu, X., 2021. Deep learning framework based on integration of S-Mask R-CNN and Inception-v3 for ultrasound image-aided diagnosis of prostate cancer. *Future Generation Computer Systems*, *114*, pp.358-367.
6. He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
7. Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*, pp.1097-1105.
8. Wang, J., Chen, Y., Hao, S., Peng, X. and Hu, L., 2019. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, *119*, pp.3-11.
9. Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L. and Xie, X., 2016, March. Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 30, No. 1).
10. Hossain, T. and Inoue, S., 2019, May. A Comparative study on missing data handling using machine learning for human activity recognition. In *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)* (pp. 124-129). IEEE.
11. Liu, R., Meng, G., Yang, B., Sun, C. and Chen, X., 2016. Dislocated time series convolutional neural architecture: An intelligent fault diagnosis approach for electric machine. *IEEE Transactions on Industrial Informatics*, *13*(3), pp.1310-1320.

12. Cook, D.J., Krishnan, N.C. and Rashidi, P., 2013. Activity discovery and activity recognition: A new partnership. *IEEE transactions on cybernetics*, *43*(3), pp.820-828.

13. Yoon, J., Zame, W.R. and van der Schaar, M., 2018. Estimating missing data in temporal data streams using multi-directional recurrent neural networks. *IEEE Transactions on Biomedical Engineering*, *66*(5), pp.1477-1490.

14. Saha, S.S., Sandha, S.S. and Srivastava, M., 2021. Deep Convolutional Bidirectional LSTM for Complex Activity Recognition with Missing Data. In *Human Activity Recognition Challenge* (pp. 39-53). Springer, Singapore.

15. Fujii, A., Kajiwara, D. and Murao, K., 2021. Cooking Activity Recognition with Convolutional LSTM Using Multi-label Loss Function and Majority Vote. In *Human Activity Recognition Challenge* (pp. 91-101). Springer, Singapore.

16. Ordóñez, F.J. and Roggen, D., 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, *16*(1), p.115.

17. Jogin, M., Madhulika, M.S., Divya, G.D., Meghana, R.K. and Apoorva, S., 2018, May. Feature extraction using convolution neural networks (CNN) and deep learning. In *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)* (pp. 2319-2323). IEEE.

18. Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*, pp.1097-1105.

19. Chen, Y., Jiang, H., Li, C., Jia, X. and Ghamisi, P., 2016. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, *54*(10), pp.6232-6251.

20. Jeyakumar, J.V., Lee, E.S., Xia, Z., Sandha, S.S., Tausik, N. and Srivastava, M., 2018, October. Deep convolutional bidirectional LSTM based transportation mode recognition. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers* (pp. 1606-1615).

21. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K. and Woo, W.C., 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *arXiv preprint arXiv:1506.04214*.