Media Engineering and Technology Faculty

German University in Cairo

Sports And Activity Recognition

Bachelor Thesis

Author: Kerolos Romany Balamon

Supervisor: Dr. Amr Talaat

Submission Date: 12 June,2022

This is to certify that:

(i) *the thesis comprises only my original work toward the Bachelor degree*
(ii) *due acknowledgement has been made in the text to all other material used*

Kerolos Romany Balamon

12 June,2022

Acknowledgments

I would like to sincerely thank Dr. Amr for the help he has generously offered throughout this project. I appreciate his assistance, guidance and patience. I would also like to express my gratitude towards my beloved parents and family for all their love and support.

Abstract

Activity Recognition in sports is a key component in smart health in that it is valuable to solve many real-life, human-centric problems such as eldercare and healthcare. Activity Recognition aims to recognize common human activities in real life settings. Accurate activity recognition is challenging because human activity is difficult to model and highly diverse. Many modern devices can be used to collect the data of human daily activity such as a smartphone, computer vision, smart watch, etc. Activity Recognition has been investigated with different algorithms proposed. In our thesis, we focus on recognizing ambulation types of activities based on the data gathered from the accelerometer and gyroscope sensors. These ambulation types of activities include walking, running, sitting, standing, walking upstairs and walking downstairs. The research on Activity Recognition in this thesis include the activity classifications and the abnormal movements identification by employing machine learning algorithms such as Convolutional Neural Network algorithm (CNN).

The data that used in this thesis is from a previous project providing open access to the public. The data was collected with experiments carried out with a group of 30 volunteers. They have ages of 19-48 years. Each person performed these six activities. The obtained dataset was randomly partitioned into two sets, where 70% of the volunteers were selected for generating the training data and 30% the test data. This thesis design a machine learning solution to mine the data for activity recognition, analyse and evaluate the performance of the solution.

Table of Contents Page

CHAPTER ONE: INTRODUCTION

1.1 Motivation

Recognizing the activity of a person and his motive from sensor data is one of the major challenges in human-computer interaction and computer vision. Identifying the activities, processing them for classification and making decision on whether it is walk, sit, stand or fall is the prime functionality of Human Activity Recognition (HAR). Athletes face many problems including suddenly heart attacks, cardiac arrests, overtraining, chest pain, stomach ache etc. Therefore, athletes or patients could be definitely supported by HAR which would help to monitor their activity and any changes in their behaviours or any occurring critical activity. This review provides a comprehensive study on HAR approaches along with the datasets used. The performance metrics used for the experimental evaluations are also analyzed and highlighted. The motivation of the research on HAR and the directions for future research are also explored.[1]

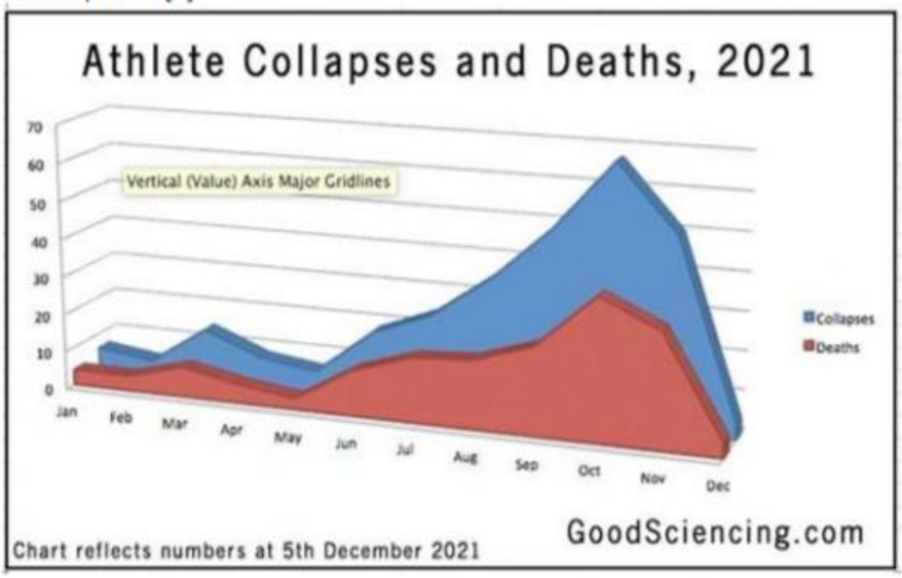


Figure 1.1: Athlete Collapses and Deaths [24]

1.2 Project Aim

In general, an activity recognition is used in different technologies to help people keep track of their daily activity movements. Some technologies are used to monitor the movement of users and to encourage them to move. The HAR system can keep a continuous observation on basic human activities of daily living.

For this thesis, we used open data sources to collect the dataset that has two kinds of sensors to recognize and predict human's daily activity by implementing the CNN algorithm. These sensors (Accelerometer and Gyroscope) are embedded to calculate the body activity for accurate results. These sensors gather signals endlessly and assign more emphasis on real-time signal achievement [2].

1.3 Methodology

The methods followed to fulfill the aim of this project briefly are as follows:

1. Choosing a suitable dataset using the good number of volunteers with the needed sensors.
2. Make a neural model using CNN on python.
   i) Train the model
   ii) Test the model
3. Using Gyroscope and accelerometer on ESP32 board.
4. Connect hardware components to Arduino to send the outputs.
5. Using the outputs with the neural model to detect the activity.

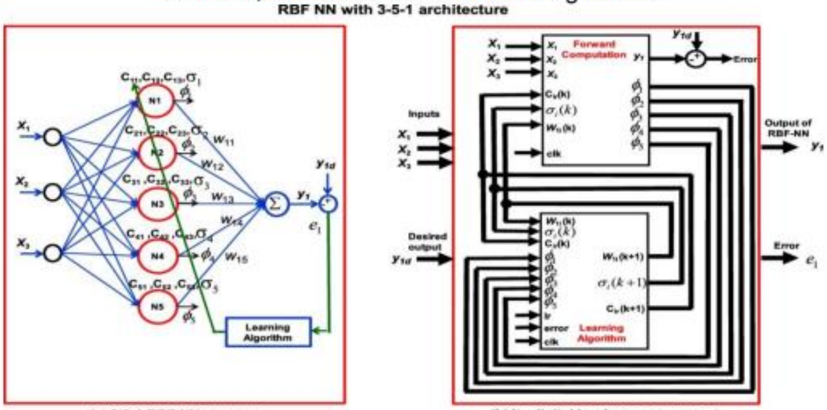
1.4 Thesis Organization

The thesis consists of 5 Chapters. Chapter 1 is the introduction, where an overview of the problem is discussed as well as the methodology proposed at solving it. Chapter 2 is the state of the art where a background about the issue is discussed and the different approaches found in the literature. Chapter 3 is the Approach Specifications, the proposed algorithm and methods used in this project are explained generically. In chapter 4, the implementation of the proposed algorithm and the tools used are presented and explained in details. Numerical examples are also given and the mathematical equations involved are introduced in Chapter 4. As for Chapter 5, it has the conclusion which summaries the work presented in this thesis and the future work and recommendations for the continuation of this work.

CHAPTER TWO: STATE OF THE ART

2.1 Introduction

In this chapter, we are going to show the previous works that tried to solve the Human Activity Recognition (HAR) problem that is related with the healthcare field. It will be split into three sections: the first section is human activity recognition software using CNN; the second section is sensors that we used to recognize human daily activities, and the last section HAR algorithms.



(a) 3-5-1 RBF NN structure (b) Its digital hardware component

Figure 2.1: digital hardware with neural network [25].

2.2 HAR

There are several papers that study Human Activity Recognition (HAR) from different views and using different algorithms. In [3], the researchers study human activity that depends on periodic System Designations from a single instance using CNN and the height of feet above the ground as the feature. The changing in feet above or below cause classification errors in the experiment results. Andres and Cesar in [4] discuss the importance of monitoring and assessing the physical

movements for people who should balance their movements because of obesity or metabolism syndrome.
An increasing interest in these types of systems causes them to be one of the keys of several applications such as visual surveillance [5], video retrieval [6] and human–computer interaction [7], among others.
Because activity recognition is a classification problem, there are several technologies for recognition of human daily activity, such as a smartphone, computer vision, etc. A smartphone [8], [9], [10] could shift in a person's pocket, which is not ideal for tracking hand based activities. The smartphone-based Human Activity Recognition (HAR) is especially limited for women and elderly people since they typically do not tend to keep a phone in their pockets for a long time. Thus, a smart band that is worn in a consistent position and ideally situated for tracking hand-based activities has recently become popular for monitoring human's daily activities. It can be utilized to decrease the death rate and predict early detection of a heart attack [11]. Recognition of human activities can be considered as the last step of a set of previous tasks, such as image capture, segmentation, tracking, identification, and classification. So, the HAR is an important area of healthcare research. There are many applications, including surveillance systems, patient monitoring systems, and a variety of systems that involve interactions between persons and electronic devices such as human-computer interfaces. Most of these applications require an automated recognition of high-level activities, composed of multiple simple (or atomic) actions of people.

Human body activity is the coordinated movement of different body parts and the connected joints. Researchers believe that knowledge of limb and joint angles is useful to detect the termination and commencement of different actions. Many studies have used information from the movement of body parts such as the trunk, arms and legs to analyse human motion in for healthcare purposes.

2.3 SENSORS

Recognizing activities for Human Activity Recognition systems have been challenging. Therefore, different types of sensors have been used to sense human's activities during daily lives. These sensors include state change sensors [7] attached to appliances and RFID tags and readers used with household items [12] to collect object usage data as an indirect way to infer user activity. In [13], the authors study HAR with a variety of sensors, including RFID sensors, switch sensors, and motion sensors, and offer an evaluation in real-world conditions. In addition, HAR becomes useful for elderly monitoring applications by using depth video sensors' technologies. Depth video sensors, which produce depth or distance information, have greatly improved HAR [14]. 5

Body Sensor Networks (BSNs) and heterogeneous sensors have been used for automatic and intelligent human daily activities to monitor elderly people [11], [14]. They aim to capture the state of a user and its environment by utilizing information from heterogeneous sensors, which allows for continuous monitoring of numerous physiological signals by attaching these sensors to the subject's body. This can be immensely useful in activity recognition for identity verification, health, ageing, sport and exercise monitoring applications. In [15] a high-accuracy HAR system based on single tri-axial accelerometer ADXL330 sensor for use in a naturalistic environment was developed. It is manufactured by Analog Devices, which is capable of sensing accelerations with tolerances. The output signal of this sensor is sampled at 100 Hz. Additionally, the data generated by it was transmitted to a personal computer using a Bluetooth device. Data from this sensor has the following attributes: time, acceleration along x-axis, acceleration along y-axis and acceleration along z-axis. In [16], a key challenge in inferring human activities from multiple sensors is the fusion of low-level streams of raw sensor data into higher-level assessments of activity. These sensors are Binaural microphones, a USB camera, and a keyboard and mouse to recognize the overall office situation, such as the presence of a phone conversation, a face to face conversation, an ongoing presentation, a distant conversation, nobody in the office, or a user is present and engaged in some other activity. [17] studied the state of the art in HAR based on wearable sensors. Because these sensors have high computational power, small size, and low cost, users can incorporate them into their daily lives

2.4 HAR ALGORITHMS

Human Activity Recognition (HAR) problem approach is broken into two stages. The first is Feature Generation, which is the process of taking raw unstructured data and defining it as a potential problem. The second stage is Feature Extraction, which tests transformations of the original features. It develops or enhances the movements into pure groupings [9]. Feature extractions are the different algorithms used for grouping the data. Recently, machine learning algorithms are generally used for recognition and classification, such as image recognition and face recognition. Oscar Lara and Miguel Lobrador in [17] present the Waikato Environment for Knowledge Analysis (WEKA), which is certainly the best-known tool in the machine learning research community. It contains implementations of several learning algorithms, and allows them to be easily evaluated for a dataset using cross validation and random split, among others. Thus, it helps to solve HAR. There are many papers that use different feature extractions other than what we used in this paper. Zhenyu and Lianwen [15] represent a high accuracy HAR system using discrete cosine transform (DCT), the Principal Component Analysis (PCA) and Support Vector Machine (SVM) for classification of human different activity. In [18], the researchers used Shift-invariant Sparse Coding algorithm for Learning Features for Activity Recognition. In [16], the authors train and test the performance of LHMMs and HMMs on recorded office activity data for (10 minutes per activity, 6 activities and 3 users). Additionally, the other papers use the WEKA Explorer mode to classify and categorize HAR by using some the classifier algorithms (such as Bayes, Functions, Lazy, Meta, Mi, Misc, Rules, and Trees) [19]. Finally, there are many different feature extractions, and the technology used for the problem will offer diverse results.

CHAPTER THREE: APPROACH SPECIFICATIONS

3.1 Introduction

In this chapter, we will present more details about the parts that were used to create the proposed system and more general explanations about them. This chapter will be separated into two sections. The first section is about the hardware that was used to build this system, such as the sensors that are used to collect data from people, and will explain the common use for each sensor. The second section explains the software that includes HAR as a real-life human-centric problem such as CNN.

3.2 HARDWARE

In this section, we explain the hardware devices that are used to build the proposed system, including the Body Sensors Network.

3.2.1 Development Board

ESP32 can perform as a complete standalone system or as a slave device to a host MCU, reducing communication stack overhead on the main application processor. ESP32 can interface with other systems to provide Wi-Fi and Bluetooth functionality through its SPI / SDIO interfaces [20].

3.2.2 BODY SENSORS NETWORK (BSN)

The Body Sensors Network (BSN) connects the physical environment with electronic systems. It is also used to collect information from the physical environment. It has collections of sensors that are responsible for processing information by format conversion, logical computing, data storage, and transmission [21]. The proposed system uses two types of sensors (accelerometer and gyroscope sensors). These are a new type of inertial sensor with a small size, light weight, low cost, and low power consumption [22].

3.2.2.1 ACCELEROMETER SENSORS

The accelerometer sensor is a device that can measure acceleration (the rate of change in velocity). It is able to detect changes in orientation and tell the screen to rotate. Basically, it helps our device know up the directions . The main use for this sensor is to measures the linear acceleration of the device on the X- axis (lateral), Y-axis (longitudinal), and Z-axis (vertical) [23].

Figure 3.2.2.1: Accelerometer sensor [26].

It has been used heavily in activity recognition. For instance, if a user changes his/ her activity from walking to jogging, it will react on the signal shape of the acceleration reading: along the vertical axis there will be an abrupt change in the amplitude. Furthermore, the acceleration data could demonstrate the motion pattern within a given a period, which is helpful in the difficult HAR.

3.2.2.2 GYROSCOPE SENSORS

The gyroscope is a device that can provide orientation information as well, but with greater precision. The gyroscope also has been used to measure the device's rotation rate by detecting the roll, pitch, and yaw motions of them along the x, y, and z axis, respectively. It adds an additional dimension to the information supplied by the accelerometer by tracking rotation or twist. Also, it measures the angular rotational velocity and rate of change. The accelerometer sensor can give either a “noisy” information output that is responsive, or give a "clean" output that's sluggish. But when we combine the 3-axis accelerometer with a 3-axis gyroscope, we get an output that is both clean and responsive at the same time [27]. Table 1 shows descriptions of accelerometer and gyroscope sensors, as well as their functions [28].

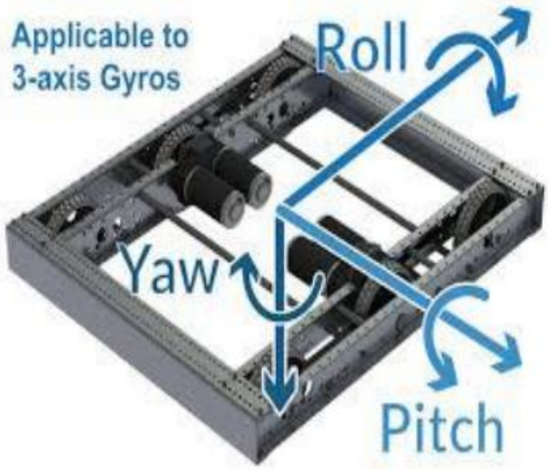


Figure 3.2.2.2: Gyroscope sensor [29].

Sensor	Description	Common use
Accelerometer	Measures the acceleration force in	Motion detection

	m/s2 on all three physical axes (x, y and z)	
Gyroscope	Measures a device's rate of rotation in rad/s on all three physical axes (x, y and z)	Rotation detection

Table 1 Accelerometer and gyroscope sensors, descriptions, and functions

3.3 SOFTWARE

Human Activity Recognition is an important technology in widespread computing because it can be applied to many real-life, human-centric problems, such as eldercare and healthcare. Activity recognition aims to recognize common human activities in real-life settings. Accurate activity recognition is challenging because human activity is difficult and highly diverse. Several probabilities based algorithms have been used to build activity. For example, the paper [8] explained several algorithms that are used to recognize human daily activity. HAR has been approached in two different ways, namely using external and wearable sensors. In the former, the devices are axed in predetermined points of interest, so the inference of activities entirely depends on the voluntary interaction of the users with the sensors. In the latter, these devices are attached to the user. An activity recognition requires two stages: training and testing. The training stage initially requires a time series dataset of measured attributes from individuals performing each activity. The time series are split into time windows to apply feature extraction, thereby altering relevant information in the raw signals. After that, learning methods are used to generate an AR model from the dataset of extracted features. In testing stage, data are collected during a time window.

3.3.1 CNN

The Convolution Neural Network (CNNet) is a feed-forward type of machine-learning algorithm based on the human nervous system. The typical architecture of a CNNet is composed of one or more stages containing convolution layers and pooling layers. These stages are followed by one or more fully-connected layers prior to a top-level. Some activation layers include Softmax, Rectified linear units (RelU), TanH etc. The CNNet has a great potential to determine the signals of HAR. Specifically, the processing units in the lower layers are used to characterize the nature of each basic movement in a human activity.

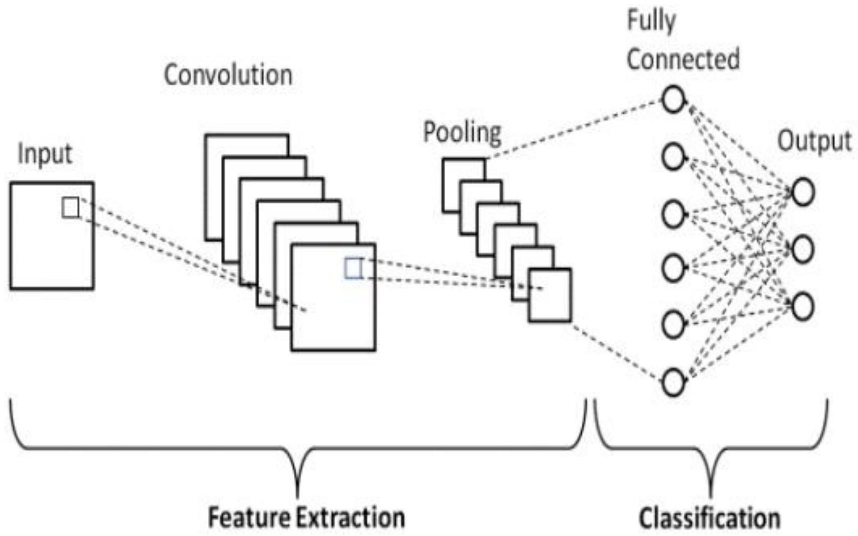


Figure 3.3.1.1: CNN [30].

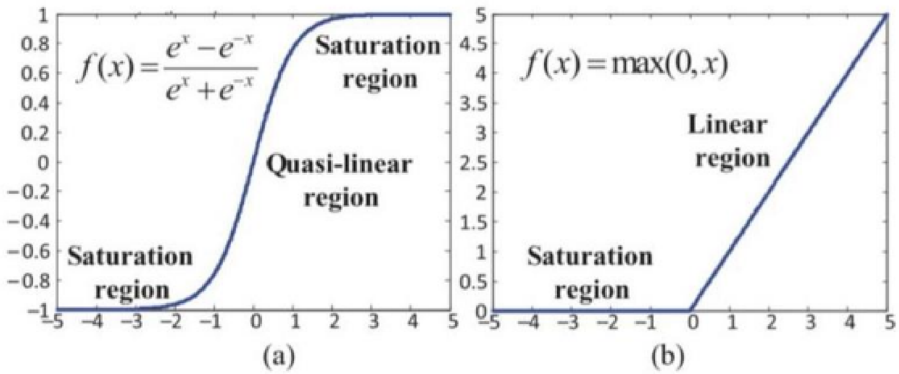


Figure 3.3.1.2: Two nonlinear activation functions. (a) TanH (b) ReLU [31]

In the higher layers, the processing units obtain the salient patterns of signals at high-level representation to distinguish the salience of a combination of several basic movements. The most important attribute of the CNNet is conducting different processing units, such as convolution, pooling, sigmoid/hyperbolic tangent squashing, rectifier and normalization alternatively [32]. The sliding window strategy is used with a length of values and with a certain percentage of overlap to extract input data for the CNNet and to segment the time series into a collection of short pieces of signals [32]. We used the L-layer CNN-based model with three kinds of layers: 1) An input layer whose values are fixed by the input data; 2) hidden layers whose values are derived from previous layers; and 3) an output layer whose values are derived from the last hidden layer [33]. We have a convolution layer that convolves the input with a set of kernels (filters) to be learned. The max pooling layer with a sub-sampling factor can preserve scale invariants. The normalization layer normalizes the values of different feature maps in the previous layer. The convolution layers are convolved with several convolutional kernels to be learned in the training process. The output of the convolutional operators is put through the activation function to form the feature map for the next layer. The value vij x,d is given by this equation as described in [32]

vij x,d = tanh (bij + ΣΣ$wijmppi−1p=0mv(i−1)mx+p,d$)

∀ d=1,….,D (1)

Where TanH represents the hyperbolic tangent function, bij represents the bias for this feature map, while m indexes over the set of feature map in the (i-1) the layer connected to the current feature map, and $wijmp$ represents the value at the position p of the convolutional kernel. Moving to the pooling layers, the resolution of the feature map is minimized to increase the invariance of features to distortions on the input. Feature maps in the previous layer are pooled by either max pooling function:

vij x,d = max ≤p≤ Qivij x,d ∀ d=1,….,D (2)

Or a sum pooling function:

vij x,d = 1\QiΣ1≤q≤Qi (vij x,d) ∀ d=1,….,D (3)

The number of layers of the CNNet classifier is varied as well to determine if machine learning will contribute to the performance of the system.

CHAPTER FOUR: IMPLEMENTATION

4.1 SOFTWARE

In this chapter, we describe a new solution for Human Activity Recognition (HAR) by using multiple sensors. First of all, we used open source dataset collection (UCI) that collected the data by using accelerometer and gyroscope. After that, we trained the data using Convolution Neural Networks algorithm. Also, we used this methodology to predict and evaluate the six activities that humans do every day.

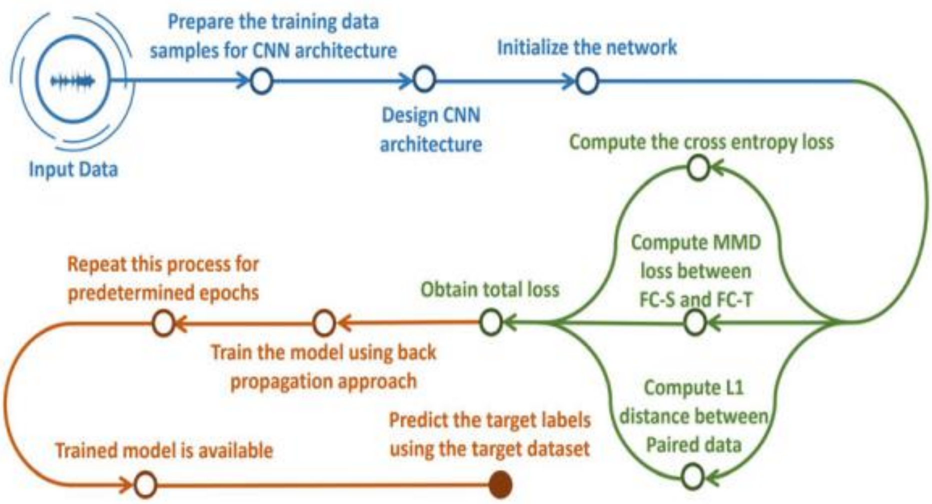


Figure 4.1: The flow diagram of CNN [34].

4.1.1 DATASET

The data that we used for detecting people was collected based on HAR by using accelerometer and gyroscope sensors that positioned in waist from the UCI Machine Learning Repository [35] . The experiments were carried out with a group of 30 volunteers between the ages of 19 and 48 years. Each person performed six activities: standing, sitting, lying, walking, walking downstairs and walking upstairs. Accelerometer and gyroscope sensors captured 3-axial linear acceleration and 3-axial angular velocity at the sampling rate of 50Hz. The dataset was randomly divided into two sets, Where 70% of the volunteers was selected for the training data and 30% the test data. The data were divided in two parts to be used separately. First part for Inertial sensor data that collect from accelerometer and gyroscope the tri-axle signals of all the volunteers trials.

Second part for records window that gather 561 feature vector connected with label that specify the activity done.

Figure 4.1.1 : HAR using gyroscope and accelerometer in waist [36].

The accelerometer and gyroscope signals were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of (2.56 sec.) with a 50% overlap (128 readings/window). The signal from an acceleration sensor has gravitational and body motion components. It also was separated using a Butterworth low-pass filter into body acceleration and gravity. The gravitational force is assumed to have only low frequency components, therefore a filter with (0.3 Hz) cutoff frequency was used. From each window, a vector of (561) features was obtained by calculating variables from the time and frequency domain.

4.1.2 MACHINE LEARNING

Machine learning is a sort of artificial intelligence that allows software applications to improve their prediction accuracy without being expressly designed to do so. In order to forecast new output values, machine learning algorithms use historical data as input [37].

4.1.2.1 NEURAL MODEL

Using CNN for doing neural model in this thesis are implemented in the Keras library, a lightweight library for building and training CNNs written in online Python compiler named Google Colabs. The model used in Laptop Hp Pavilion (Gaming) with Core i7 9th Generation, 8GB Ram,1TB HDD and 256GB SSD, NVIDIA GeForce GTX 1650. I used Laptop with a good GPU processors to make the implementation as fast as possible and it takes almost 15 minutes. We trained 70% of the data which consist of 7352 sample , and the 30% remaining were used at the same time for the testing which consist of 2947 sample, making our method very efficient trying to get the highest accuracy.

First of all I tried to create a base model to my data

```
model = Sequential()
model.add(Dense(units=64,kernel_initializer='normal',activation='sigmoid',input_dim=x_train.shape[1]))
model.add(Dropout(0.2))
model.add(Dense(units=6,kernel_initializer='normal',activation='softmax'))
model.compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=['accuracy'])
history = model.fit(x_train, y_train, batch_size = 64, epochs= 10,validation_data = (x_test,y_test))

Epoch 1/10
115/115 [==============================] - 2s 6ms/step - loss: 1.3045 - accuracy: 0.4875 - val_loss: 1.0334 - val_accuracy: 0.5663
Epoch 2/10
115/115 [==============================] - 1s 5ms/step - loss: 0.8728 - accuracy: 0.7233 - val_loss: 0.7317 - val_accuracy: 0.7913
Epoch 3/10
115/115 [==============================] - 1s 4ms/step - loss: 0.6182 - accuracy: 0.8172 - val_loss: 0.5350 - val_accuracy: 0.8656
Epoch 4/10
115/115 [==============================] - 1s 8ms/step - loss: 0.4660 - accuracy: 0.8617 - val_loss: 0.4183 - val_accuracy: 0.8907
Epoch 5/10
115/115 [==============================] - 1s 8ms/step - loss: 0.3760 - accuracy: 0.8847 - val_loss: 0.3499 - val_accuracy: 0.9053
Epoch 6/10
115/115 [==============================] - 1s 8ms/step - loss: 0.3165 - accuracy: 0.8965 - val_loss: 0.2924 - val_accuracy: 0.9155
Epoch 7/10
115/115 [==============================] - 1s 9ms/step - loss: 0.2695 - accuracy: 0.9169 - val_loss: 0.2701 - val_accuracy: 0.9046
Epoch 8/10
115/115 [==============================] - 1s 7ms/step - loss: 0.2373 - accuracy: 0.9217 - val_loss: 0.2370 - val_accuracy: 0.9250
Epoch 9/10
115/115 [==============================] - 1s 7ms/step - loss: 0.2177 - accuracy: 0.9282 - val_loss: 0.2296 - val_accuracy: 0.9182
Epoch 10/10
115/115 [==============================] - 1s 6ms/step - loss: 0.1938 - accuracy: 0.9368 - val_loss: 0.2028 - val_accuracy: 0.9281
```

Figure 4.1.2.1.1: Base Model [Screenshot from code].

Then hypertuning that model

```
def build_model(hp):
    model = keras.Sequential()
    for i in range(hp.Int('num_layers', 2, 25)):
        model.add(layers.Dense(units = hp.Int('units' + str(i), min_value=32, max_value=512, step=32),
                               kernel_initializer= hp.Choice('initializer', ['uniform', 'normal']),
                               activation= hp.Choice('activation', ['relu', 'sigmoid', 'tanh'])))
    model.add(layers.Dense(6, kernel_initializer= hp.Choice('initializer', ['uniform', 'normal']), activation='softmax'))
    model.add(
          Dropout(0.2))
    model.compile(
        optimizer = 'adam',
        loss='sparse_categorical_crossentropy',
        metrics=['accuracy'])
    return model


tuner = RandomSearch(
    build_model,
    objective='val_accuracy',
    max_trials= 5,
    executions_per_trial=3,
    directory='project', project_name = 'Human_activity_recognition')

tuner.search_space_summary()
```

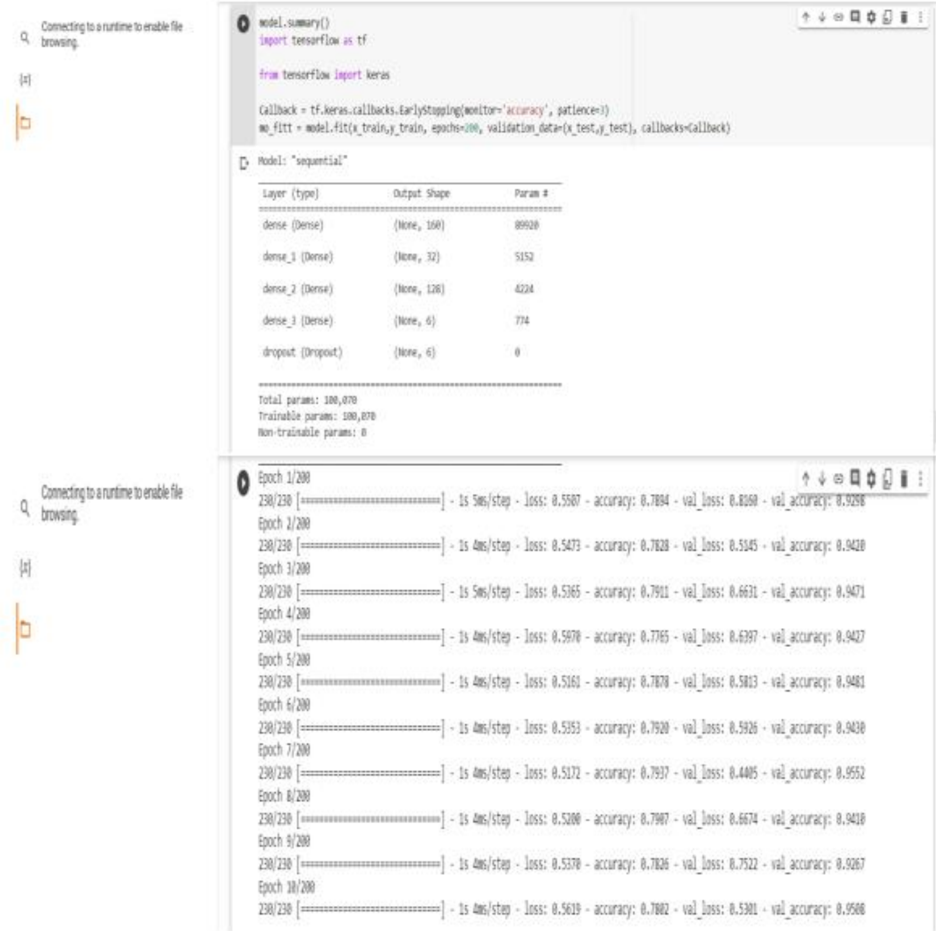

```
    executions_per_trial=3,
    directory='project', project_name = 'Human_activity_recognition')

tuner.search_space_summary()
```


```
Search space summary
Default search space size: 5
num_layers (Int)
{'default': None, 'conditions': [], 'min_value': 2, 'max_value': 25, 'step': 1, 'sampling': None}
units0 (Int)
{'default': None, 'conditions': [], 'min_value': 32, 'max_value': 512, 'step': 32, 'sampling': None}
initializer (Choice)
{'default': 'uniform', 'conditions': [], 'values': ['uniform', 'normal'], 'ordered': False}
activation (Choice)
{'default': 'relu', 'conditions': [], 'values': ['relu', 'sigmoid', 'tanh'], 'ordered': False}
units1 (Int)
{'default': None, 'conditions': [], 'min_value': 32, 'max_value': 512, 'step': 32, 'sampling': None}
```

Figure 4.1.2.1.2: Hypertuning Model [Screenshot from code].

Lastly I go with the summary that have 100,070 params which all of them are trainable with high accuracy and 10 Epoch with accuracy rate of 94.2%.



```
model.summary()
import tensorflow as tf

from tensorflow import keras

Callback = tf.keras.callbacks.EarlyStopping(monitor='accuracy', patience=3)
mo_fitt = model.fit(x_train,y_train, epochs=200, validation_data=(x_test,y_test), callbacks=Callback)
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
dense (Dense)               (None, 160)               89920

dense_1 (Dense)             (None, 32)                5152

dense_2 (Dense)             (None, 128)               4224

dense_3 (Dense)             (None, 6)                 774

dropout (Dropout)           (None, 6)                 0

=================================================================
Total params: 100,070
Trainable params: 100,070
Non-trainable params: 0
_________________________________________________________________
```

```
Epoch 1/200
230/230 [==============================] - 1s 5ms/step - loss: 0.5507 - accuracy: 0.7894 - val_loss: 0.8160 - val_accuracy: 0.9298
Epoch 2/200
230/230 [==============================] - 1s 4ms/step - loss: 0.5473 - accuracy: 0.7828 - val_loss: 0.5145 - val_accuracy: 0.9420
Epoch 3/200
230/230 [==============================] - 1s 5ms/step - loss: 0.5365 - accuracy: 0.7911 - val_loss: 0.6631 - val_accuracy: 0.9471
Epoch 4/200
230/230 [==============================] - 1s 4ms/step - loss: 0.5970 - accuracy: 0.7765 - val_loss: 0.6397 - val_accuracy: 0.9427
Epoch 5/200
230/230 [==============================] - 1s 4ms/step - loss: 0.5161 - accuracy: 0.7878 - val_loss: 0.5813 - val_accuracy: 0.9481
Epoch 6/200
230/230 [==============================] - 1s 4ms/step - loss: 0.5353 - accuracy: 0.7920 - val_loss: 0.5926 - val_accuracy: 0.9430
Epoch 7/200
230/230 [==============================] - 1s 4ms/step - loss: 0.5172 - accuracy: 0.7937 - val_loss: 0.4405 - val_accuracy: 0.9552
Epoch 8/200
230/230 [==============================] - 1s 4ms/step - loss: 0.5200 - accuracy: 0.7987 - val_loss: 0.6674 - val_accuracy: 0.9418
Epoch 9/200
230/230 [==============================] - 1s 4ms/step - loss: 0.5370 - accuracy: 0.7826 - val_loss: 0.7522 - val_accuracy: 0.9267
Epoch 10/200
230/230 [==============================] - 1s 4ms/step - loss: 0.5619 - accuracy: 0.7802 - val_loss: 0.5301 - val_accuracy: 0.9508
```

Figure 4.1.2.1.3: Model Summary [Screenshot from code].

4.2 HARDWARE

As mentioned in previous chapters I used accelerometer sensor and gyroscope sensor on Esp32 board named (Esp32 devkit v1).
I linked both sensors with the board using female to female wires by the help of the data sheet of esp32 devkit v1

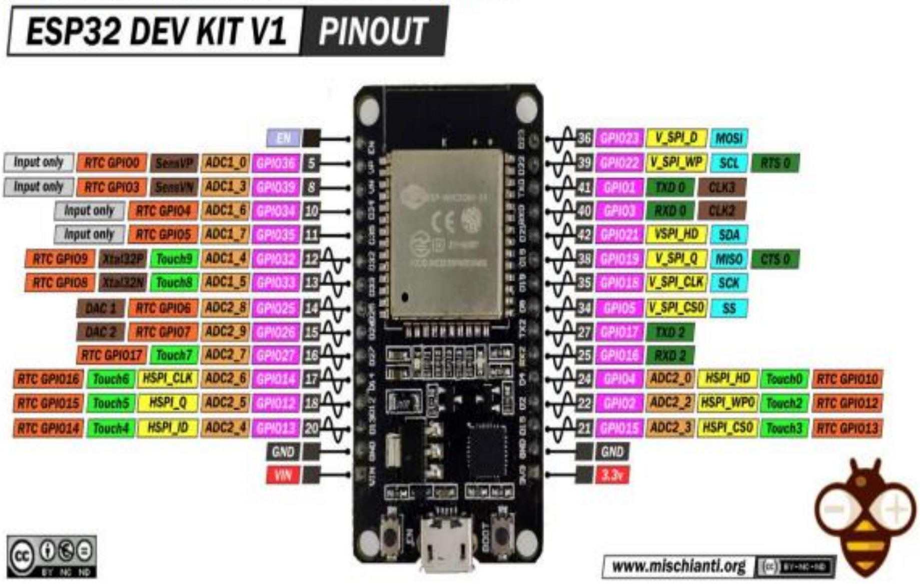


Figure 4.2: ESP32 Datasheet [38].

4.2.1 Arduino

Arduino is an open-source electronics platform that uses simple hardware and software to make it easy to use. Arduino boards can take inputs - such as light from a sensor, a finger on a button, or a Twitter message - and convert them to outputs - such as turning on an LED, triggering a motor, or publishing anything online. By providing a set of instructions to the board's microcontroller, you may tell it what to do. The Arduino programming language (based on Wiring) and the Arduino Software (IDE) (based on Processing) are used to do this [39].

After connecting esp32 to the accelerometer and gyroscope, I tried to translate our results into our software using Arduino.

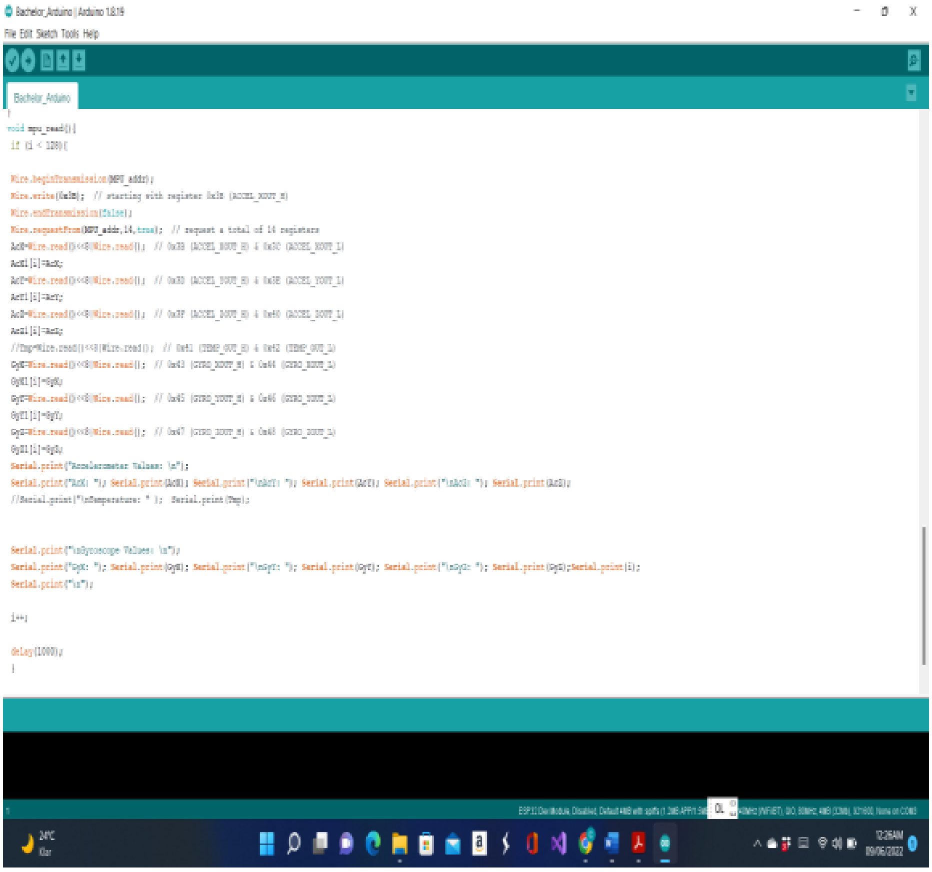


Figure 4.2.1.1: Reading Data [Screenshot from Arduino code].

As shown in figure 4.2.1.1, I wrote method called mpu_read that got called 128 time that read the data from sensors with delay of one second between every time and print in serial monitor the results.

While doing this function I made 6 arrays of size 128 that store every result corresponding to the specific place.

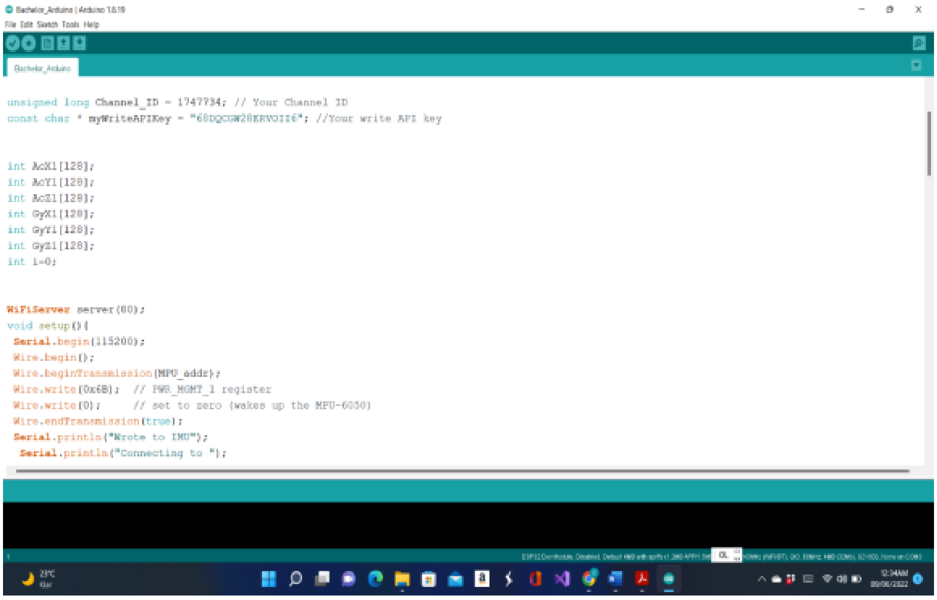


Figure 4.2.1.2: Reading Data [Screenshot from Arduino code].

CHAPTER FIVE: CONCLUSION

The goal of current technological advancements is to make every device in our environment smart as much as feasible. One of the goals of developing smart systems is to recognise daily human actions using a smart device worn around the waist to encourage people concerned about their health to live a safe and healthy life. In our thesis, we proposed employing Convolution Neural Networks as a machine learning approach to automate features from a dataset for the HAR task. To analyse multichannel time series data, the suggested method creates an unique machine learning machine architecture for CNNs. It makes use of convolution and pooling processes to capture the key patterns in sensor signals at various time scales. The data for our proposed software system was gathered utilising the HAR system and two sensors: an accelerometer and a gyroscope, which we obtained from the UCI Machine Learning Repository website. After that, we used a CNNs model to train the data, which means we used one algorithm for all of the data we had. On Google Colabs, we used the Python programming language to code this algorithm. The model training and classification are run on a Laptop Hp Pavilion (Gaming) with Core i7 9th Generation, 8GB Ram,1TB HDD and 256GB SSD, NVIDIA GeForce GTX 1650. Then, I used the same algorithm for the prediction and evaluation stages. The implementation of CNNs for this work is based on the knowledge that simulating the human nervous system to detect and identify objects will result in high accuracy over other machine learning techniques depending on all the experiments that we did on the dataset. In the other hand, Using Arduino to translate our sensors results into code to directs it to software helps us reach our goals.

5.1 FUTURE WORK

A proposed future work is to complete this investigation with CNNs and Arduino by linking it with each other using tensorflow lite library, but I faced alots of problems that based on the 561 features used in the neural model.

Making too much features makes high accuracy level but affects badly that making all these features wasn´t an easy task and if I had more time I will handle these features.

In addition, I could use other sensors and combine them with our current sensors to improve the healthcare monitoring applications for better performance and better results.

For examples using sensors that could calculate how many calories burnt depending on how many steps taken.

For instance, there are many applications, including surveillance systems, patient monitoring systems.

LIST OF FIGURES

LIST OF TABLES

TITLE **PAGE**

List of Terms

HAR	Human Activity Recognition
CNN	Convolutional Neural Network
WEKA	Waikato Environment for Knowledge Analysis
DCT	Discrete Cosine Transform
PCA	Principal Component Analysis
SVM	Support Vector Machine
LHMMs	Layer Hidden Markov Model
HMMs	Hidden Markov Model
BSN	BODY SENSORS NETWORK
MCU	Multipoint Control Unit
SPI	Serial Peripheral Interface
SDIO	Secure Digital Input Output
RelU	Rectified linear units

References

[1] By NFSC NEWS on December 7th, 2021, a report on 'Real Science' updated from December 5th to 6th: (Added Athlete Name) Adam El-Selhadar, Pompeo Tretola, Ben Madgen, Marvin M Organ, Ugur Tezel, Stevan Jelovac, Jake Lever. https://gnews.org/1741365/

[2] M. Zeng, L. Nguyen, B. Yu, O. Mengshoel, J. Zhu, P. Wu, and J. Zhang, Convolutional neural networks for human activity recognition using mobile sensors, in Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on, pp.197–205. IEEE, 2014.

[3] E. Ijjina and C. Mohan, One-shot periodic activity recognition using convolutional neural networks, in Machine Learning and Applications (ICMLA), 2014 13th International Conference on, pp. 388–391. IEEE, 2014.

[4] C. Torres-Huitzil and A. Alvarez-Landero, Accelerometer-based human activity recognition in smartphones for healthcare services, in Mobile Health, pp. 147–169, Springer, 2015.

[5] N. Haering, P. Venetianer, A. Lipton, the evolution of video surveillance: an overview, Machine Vision and Applications, 19 (2008) PP. 279–290.

[6] P. Geetha, V. Narayanan, A survey of content-based video retrieval, Computer Science 4 (6) (2008) 474–486.

[7] E. Tapia, S. Intille, and K. Larson, Activity recognition in the home setting using simple and ubiquitous sensors, in Proceedings of PERVASIVE, vol. 3001, 2004, pp. 158–175.

[8] G. Chetty and M. White, Body sensor networks for human activity recognition, in Signal Processing and Integrated Networks (SPIN), 2016 3rd International Conference on, pp. 660–665. IEEE, 2016.

[9] D. Anguita, A. Ghio, L. Oneto, X. Parra and J. Reyes-Ortiz, Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine, in International Workshop on Ambient Assisted Living, pp. 216–223. Springer, 2012.

[10] N. Lane and P. Georgiev, Can deep learning revolutionize mobile sensing?, in Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications, pp. 117–122. ACM, 2015.

[11] Y. Yadav and M. Gowda, Heart rate monitoring and heart attack detection using wearable device, international journal of technical research and applications e-ISSN: 2320-8163, volume 4, Issue 3 (May- June 2016), PP. 48-50.

[12] M. Philipose, K. Fishkin, M. Perkowitz, D. Patterson, D. Fox, H. Kautz, and D. Hahnel, Inferring activities from interactions with objects, IEEE Pervasive Computing, vol. 3, no. 4, pp. 50–57, 2004.

[13] B. Logan, J. Healey, M. Philipose, E. Tapia, and S. Intille, A long- term evaluation of sensing modalities for activity recognition, in Proc. of Ubicomp, 2007.

[14] J. Ahmad, K. Shaharyar, K. Daijin, A Depth Video Sensor-Based Life-Logging Human Activity Recognition System for Elderly Care in Smart Indoor Environments, Sensors 2014, 14, PP. 11735-11759; doi:10.3390/s140711735

[15] Z. He and L. Jin, Activity recognition from acceleration data based on discrete consine transform and svm, in Systems, Man and Cybernetics, 2009, SMC 2009, IEEE International Conference on, pp. 5041–5044. IEEE, 2009.

[16] N. Oliver and A. Garg, Layered representations for human activity recognition, proceedings of the Fourth IEEE International conference on multimodal interfaces (ICMI'02) 0-7695-1834-6/02 $17.00 © 2002 IEEE, pp. 1–6.

[17] Lara and M. Lobrador, A survey on human activity recognition using wearable sensors, IEEE communications surveys & tutorials, pp. 1192– 1209, VOL. 15, NO. 3, third quarter 2013.

[18] C. Vollmer, H. Gross, and J. Eggert, Learning features for activity recognition with shift-invariant sparse coding, in International Conference on Artificial Neural Networks, pp. 367–374, Springer, 2013.

[19] M. Ayu, S. Ismail, A. Abdul Matin and T. Mantoro, A comparison study of classifier algorithms for mobile-phone's accelerometer based activity recognition, international Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012), pp. 224 – 229.

[20] ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD. https://www.espressif.com/en/products/socs/esp32

[21] X. Lai, Q. Liu, X. Wei, W. Wang, G. Zhou, and G. Han, A survey of body sensor networks, Sensors, 13 (5):5406–5447, 2013.

[22] D. Yuan, X. Ma, Y. Liu, S. Yan, and C. Hao, Statistical modeling of additive noise and random drift for triaxial rate gyros and accelerometers, in Intelligent Control and Automation (WCICA), 2016 12th World Congress on, pages 308–313, IEEE, 2016.

[23] H. Cho, S. Kim, J. Baek and P. Fisher, Motion recognition with smart phone embedded 3-axis accelerometer sensor, 2012 IEEE International Conference on Systems, on pages 919– 924, Man, and Cybernetics October 14-17, 2012, COEX, Seoul, Korea.

[24] 292 athletes got cardiac arrest and 167 passed away after being vaccinated against the CCP virus https://gnews.org/1741365/

[25] Nguyen PhanThanh,Ying-ShiehKung,Seng-ChiChen,Hsin-HungChou, Digital hardware implementation of a radial basis function neural network

[26] https://opencircuit.shop/product/adxl345-3-axis-accelerometer-gy-291

[27] D. Johnson and M. Trivedi, driving style recognition using a smartphone as a sensor platform, 2011 14th International IEEE Conference on Intelligent Transportation Systems Washington, DC, USA. Pages 1609– 1615, October 5-7, 2011.

[28] W. Lee and R. Lee, Multi-sensor authentication to improve smartphone security, in: Conference on Information Systems Security and Privacy, 2015, pp. 1-11.

[29] https://docs.wpilib.org/en/stable/docs/hardware/sensors/gyros-hardware.html

[30] https://www.upgrad.com/blog/basic-cnn-architecture/

[31] G. Xu and H. Wu, Structural design of convolutional neural networks for steganalysis, IEEE SIGNAL PROCESSING LETTERS, on pagers 703– 712, VOL. 23, NO. 5, MAY 2016.

[32] J. Yang, M. Nguyen, P. San, X. Li Li, and S. Krishnaswamy, Deep convolutional neural networks on multichannel time series for human activity recognition, 2015.

[33] M. Zeng, L. Nguyen, B. Yu, O. Mengshoel, J. Zhu, P. Wu, and J. Zhang, Convolutional neural networks for human activity recognition using mobile sensors, in Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on, pp.197–205. IEEE, 2014.

[34] Siahpour, Shahin & Li, Xiang & Lee, Jay. (2020). Deep learning-based cross-sensor domain adaptation for fault diagnosis of electro-mechanical actuators. International Journal of Dynamics and Control. 8. 10.1007/s40435-020-00669-0.

[35] The UCI Machine Learning Repository, Smartphone-based recognition of human activities and postural transitions data set, 2015.

[36] Zhang, Mi & Sawchuk, Alexander. (2012). USC-HAD: a daily activity dataset for ubiquitous activity recognition using wearable sensors. 1036-1043. 10.1145/2370216.2370438.

[37] Ed Burns.(2020) In-Depth Guide to Machine Learning in the Enterprise.

[38] https://www.mischianti.org/2021/02/17/doit-esp32-dev-kit-v1-high-resolution-pinout-and-specs/

[39] https://www.arduino.cc/en/Guide/Introduction