RESEARCH PROJECT REPORT

Human Activity Recognition Using 1-D CNN in Tensorflow

Author:
Loubna FARES

Supervisor:
Prof. Abderrahman BOUHAMIDI

A report submitted in fulfillment of the Research Project

June 9, 2021

Declaration of Authorship

I, Loubna FARES, declare that this report titled, "Human Activity Recognition Using 1-D CNN in Tensorflow" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research project at this University.
- Where any part of this report has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the report is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: LOUBNA

Date: 09.06.2021

"Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism."

Dave Barry

Abstract

In the recent decades, traditional pattern recognition methods have shown a great progress. However, these methods rely heavily on manual feature extraction, which may hinder the generalization model performance. With the increasing popularity and success of deep learning methods, using these techniques to recognize human actions in mobile and wearable computing scenarios has attracted widespread attention. Application of deep learning in human activity recognition has played a vital role in healthcare and daily routine support. Hence in this project we design a convolutional neural network based deep learning model in Tensorflow for human activity recognition. We use the WISDM data set which contains six classes of human activity i.e., jogging, walking, upstairs, downstairs, standing and sitting to train and test the proposed model. Furthermore, we analyse the performance of proposed model using the state of art matrices: Recall, precision and F1-score. The proposed model provides an overall classification accuracy of 93%.

Acknowledgements

I would like to express my deepest gratitude and appreciation to all people who helped and encouraged me. I am thankful to my supervisor Dr. Abderrahman BOUHAMIDI for giving me the opportunity to work on this project. My appreciation and heartfelt thanks and gratitude also extends to my parents for supporting me. Lastly, not forgetting my dear friends for their insights which meant a lot in making assignment a success.
I am thankful and fortunate enough to get constant encouragement, support and guidance from all teaching staffs of Computer Science which helped us in successfully completing our project work.

Contents

List of Figures

List of Tables

List of Abbreviations

HAR	**H**uman **A**ctivity **R**ecognition
CNN	**C**onvolutional **N**eural **N**etwork
WISDM	**W**ireless **S**ensor **D**ata **M**ining
API	**A**pplication **P**rogramming **I**nterface
IoT	**I**nternet **o**f **T**hings
DL	**D**eep **L**earning
DBN	**D**eep **B**elief **N**etwork
MLP	**M**ulti-**L**ayers **P**erception
AI	**A**rtificial **I**ntelligence
ML	**M**achine **L**earning
TP	**T**rue **P**ositive
TN	**T**rue **N**egative
FP	**F**alse **P**ositive
FN	**F**alse **N**egative
ReLU	**Re**ctified **L**inear **U**nit

Chapter 1

Introduction

1.1 Introduction to HAR

Human activity recognition (HAR) has become an important research field over the two last decades, especially because of the spread of electronic devices like mobile phones, smart cell phones and video cameras in our daily lives [1]. Additionally, the progress of deep learning and other algorithms has made it possible for researchers to use HAR in many fields including sports, health and well being. HAR is, for example, one of the most promising resources for helping older people with the support of their cognitive and physical function through day-to-day activities. HAR is one of the most challenging research area, it's performance depends on various factors most of which focus on robustness, accuracy and real-time capability. Thanks to the advancement of microelectronics and computer systems, there has been a comprehensive development in high-capacity, low-power, low-cost and miniaturized sensors. Using these sensors in internet of things (IoT) architecture a lot of data has been collected. The data available today made it possible for us to train deep learning model to find the hidden patterns in it to help human kind. Especially, sensors embedded in mobile devices collect a lot of data which can be used to analyze human daily activities and it has become feasible in order to guarantee health surveillance, for physical training and for safety proposes of soldiers in strategic situations.

Particularly, the human activity recognition has become the key interest within the field, especially for medical, military, and security applications. For instance, patients with diabetes, obesity, or heart disease are often required to follow a well defined exercise routine as part of their treatments [2]. Hence, human activity recognition such as sitting, walking, running, upstairs, and downstairs becomes very useful to provide feedback to the healthcare practitioners about the patient's behavior. Similarly, patients with dementia and other mental pathologies could be observed and monitored to detect abnormal activities and thereby preventing undesirable events to occur [3]. Furthermore, in security and defence scenarios, precise and accurate information about the soldiers' activities along with their locations and health conditions, is highly beneficial for their performance, health and safety. Such type of information is also helpful to support quick and reliable decision making in both combat and training scenarios [4].

In the last decade, due to advances of the processing capabilities, a large amount of Deep Learning (DL) techniques have been developed and successfully have been applied in recognition tasks. These techniques allow an automatic extraction of features without any domain knowledge domains [5, 6]. Especially, Convolutional Neural Networks (CNN), which can capture local dependency and scale invariance of data from sensors has been widely used in human activity recognition and image recognition and they outperform the classical machine learning methods like random forest, and K-nearest neighbours, etc. Hence in this project we propose a human activity recognition deep learning model base of convolution neural networks. Further details of our contribution is provided in section 1.2.

1.2 Contributions

In this project we design a convolution neural network based classifier for the classification of human activity in Tensorflow using keras API. We use a public data to train and test the proposed model. The data contains samples of six different human activities collected using an accelerometer. Furthermore, we also analyse the proposed model using four measuring metrices: accuracy, precision, recall, and f1-score. The proposed classifier provides and overall accuracy of 93%.

Chapter 2

Background and literature Review

2.1 literature Review

A lot of research has been done in HAR systems [7, 8] where the researchers focused on a number of activities in different application domains [9, 10]. For instance, the activities can be including, walking, running, sitting, upstairs walking, and downstairs walking, etc. Regarding the duration and complexity of the activities; they can be categorized into three key groups: complex activities, short activities, and simple activities. The group of short activities consist of activities with very short duration such as a state transition from sit to stand. The second kind of activities are basic activities walking and reading and writing. The last on is basically combinations of progressions of basic activities with the interaction with other objects and individuals. Owing to the benefits of human activity recognition, a lot of research has been done so far. For example, [11] presented a smartphone inertial sensors-based approach for human activity recognition. Efficient features were first extracted from raw data. Finally, the features are trained with a Deep Belief Network (DBN) for successful activity recognition. In [12] authors presents a novel framework to classify and analyze human activities. A new convolutional neural network (CNN) strategy is applied to a single user movement recognition using a smartphone. Three parallel CNNs are used for local feature extraction, and latter they are fused in the classification task stage. The whole CNN scheme is based on a feature fusion of a fine-CNN, a medium-CNN, and a coarse-CNN. A tri-axial accelerometer and a tri-axial gyroscope sensor embedded in a smartphone are used to record the acceleration and angle signals. Six human activities successfully classified are walking, walking-upstairs, walking-downstairs, sitting, standing and laying. Performance evaluation is presented for the proposed CNN.

2.1.1 Background

In this section we provide the background information needed for this project, such as Convolutional neural networks, Tensorflow, etc.

2.1.2 Convolutional neural networks

Convolutional neural networks (CNN) emerged from a study of the brain's visual cortex and have been used in image recognition since the 1980s. CNN is an artificial neural network used for analysing images, but it can also be used for other data classification such as object detection for driver less cars, face recognition on social media or image analysis in healthcare. In this section, only image recognition will be studied.

Convolution layers

The convolution layer is the most important block of the CNN. In the first layer, the neurons are not connected to every pixel of the image but only to the pixels of their local receptive field, for example a rectangular field. In turn, the neurons in the second convolution layer are each connected only to the neurons inside a small rectangle in the first layer. This architecture allows the network to focus on low-level features in the first hidden layer, then assemble them into higher-level features in the next hidden layer, etc. This hierarchical structure is recurrent in real images.

Pooling layers

The purpose of pooling layers is to shrink the input image to reduce the computational load, memory usage and number of parameters. Its operation is similar to that of the convolutional layers.

Basically, a succession of filters is applied to an input image, the input data can be multi-dimensional. These filters will generate convolutional layers which will be processed with max-pooling layers in a recurrent way. The recovered elements became the input data of a Multilayer Perception (MLP), which has fully connected layers. The role of this part is to combine the characteristics of the CNN code to classify the input.

2.1.3 TensorFlow

TensorFlow [13] is widely used as a machine learning implementation library. It was created by Google as part of the Google Brain project and was later made available as an open source product, as there were multiple machine learning and deep learning frameworks that were capturing the attention of users. With open source availability, more and more people in the artificial intelligence (AI) and machine learning communities were able to adopt TensorFlow and build features and products on top of it. It not only helped users with implementation of standard machine learning and deep learning algorithms but also allowed them to implement customized and differentiated versions of algorithms for business applications and various research purposes. In fact, it soon became one of the most popular libraries in the machine learning and AI communities—so much so that people have been building a huge number of apps using TensorFlow under the hood. This is principally owing to the fact that Google itself uses TensorFlow in most of its products, whether Google Maps, Gmail, or other apps.

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications. TensorFlow offers multiple levels of abstraction so you can choose the right one for your needs. Build and train models by using the high-level Keras API, which makes getting started with TensorFlow and machine learning easy, as shown in Figure 2.1.

Simpler APIs

One of the most common criticisms of TensorFlow by users regarded its APIs, which were not user-friendly, thus a major focus of TensorFlow 2.0 has been on overhauling its APIs. Now, TensorFlow 2.0 provides two levels of APIs:

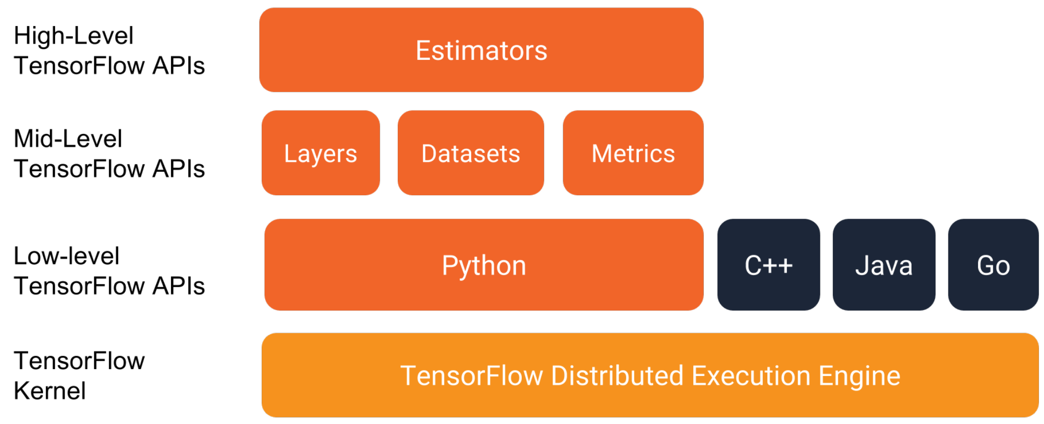


FIGURE 2.1: TensorFlow and other APIs

1. High-level APIs
2. Lower-level APIs

High-Level APIs

The high-level APIs make it easier to use TensorFlow for various applications, as these APIs are more intuitive in nature. These new high level APIs have made debugging relatively easier than in earlier versions. As TensorFlow 1.0 was graph control–based, users were not able to debug their programs easily. TensorFlow 2.0 has now introduced eager execution, which performs operations and returns output instantly

Lower-Level APIs

Another available set of APIs are lower level APIs which offer much more flexibility and configuration capability to the users in order to define and parameterise the models as per their specific requirements.

Eager Execution

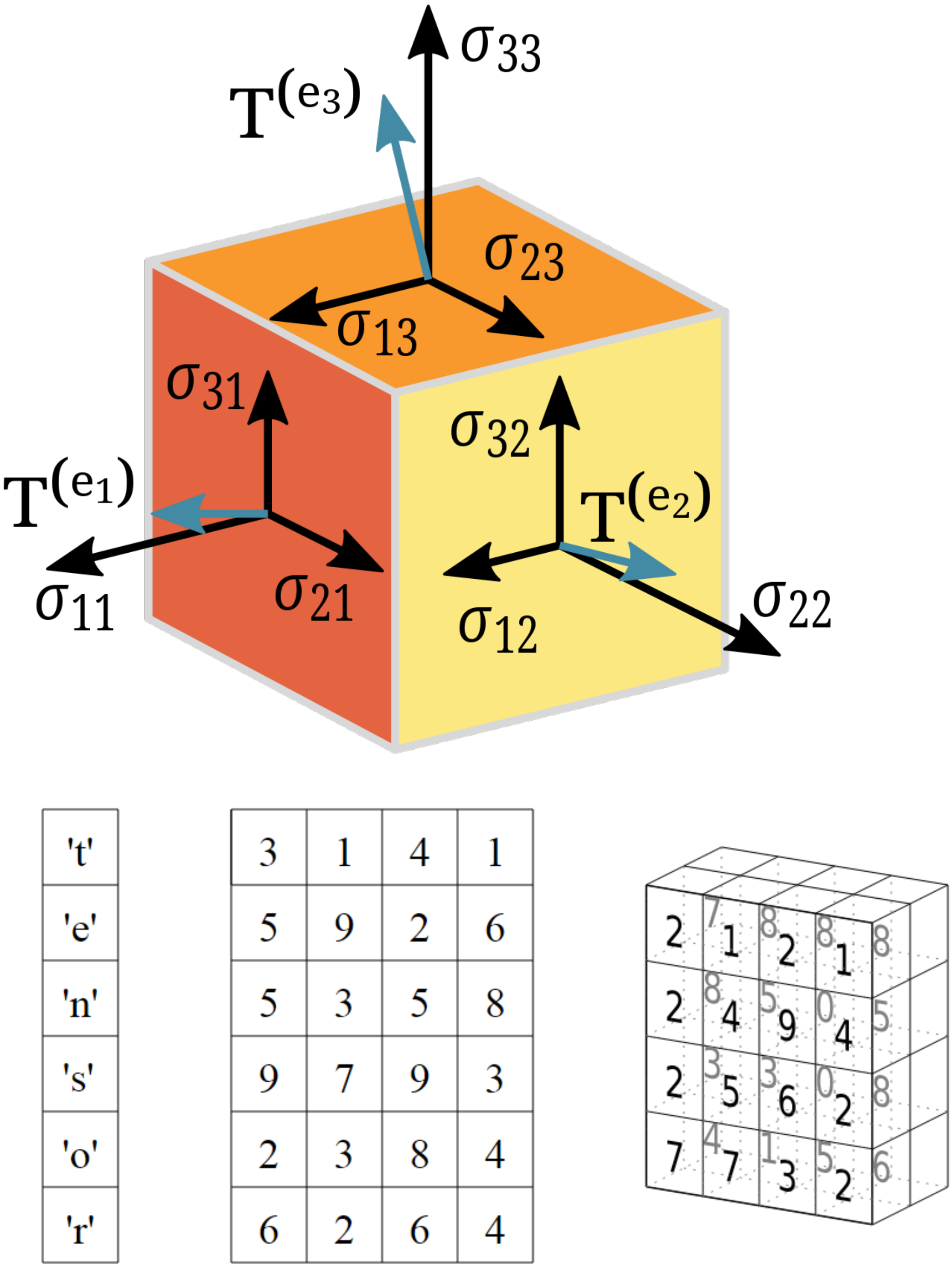
Keras has become the official high level API for TensorFlow. It is a Deep Learning API written in Python and it's user-friendliness. Keras offers abstractions that make it easy to develop deep learning model. However, it does not have it's own execution engine and is dependent on other frameworks like Theano or TensorFlow.

In case where Keras, which is a high level API, is not sufficient for your needs, you'll still need to use the TensorFlow low-level API. In such cases, Eager execution mode in is activated by default.

Eager Execution is an imperative programming environment that evaluates operations immediately, without building graphs: operations return concrete values instead of constructing a computational graph to run later.

Tensor

A tensor is a mathematical entity which represents different properties, similar to a scalar, vector, or matrix. It is a generalization of a scalar or vector. In short, tensors are multidimensional arrays that have some dynamic properties [14] . A vector is a one-dimensional tensor, whereas two-dimensional tensors are matrices as shown in Figure 2.2.



tensor of dimensions [6]
(vector of dimension 6)

tensor of dimensions [6,4]
(matrix 6 by 4)

tensor of dimensions [4,4,2]

FIGURE 2.2: Tensor

Rank

The Rank (or order) of a tensor denotes the number of directions required to describe the properties of an object, meaning the dimensions of the array contained in the tensor itself. Breaking this down for different objects, a scalar doesn't have any direction and, hence, automatically becomes a rank 0 tensor, whereas a vector, which can be described using only one direction, becomes a first rank tensor. The next object, which is a matrix, requires two directions to describe it and becomes a second rank tensor.

Shape

The shape of a tensor represents the number of values in each dimension.

Scalar—45: The shape of the tensor would be [ ].

Vector—[1, 7, 3]: The shape of the first rank tensor would be [3].

Matrix = $\begin{bmatrix} 5 & 2 & 8 \\ 7 & 6 & 4 \\ 1 & 3 & 9 \end{bmatrix}$: The second rank tensor would have a shape of [3,3].

2.1.4 Evaluation Metrices

1. **Accuracy** is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same. Therefore, you have to look at other parameters to evaluate the performance of your model.

$$Accuracy = TP + TN/TP + FP + FN + TN \tag{2.1}$$

where TP, FN, FP and TN represent the number of true positives, false negatives, false positives and true negatives, respectively.

2. **Precision** is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$Precision = TP/TP + FP \tag{2.2}$$

3. **Recall (Sensitivity)** is the ratio of correctly predicted positive observations to the all observations in actual class - positive.

$$Recall = TP/TP + FN \tag{2.3}$$

4. **F1 score** is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

$$F1Score = \frac{2 \times (Recall \times Precision)}{(Recall + Precision)} \tag{2.4}$$

Chapter 3

The proposed Model

3.1 Methodology

In this section we describe the propose 1-D convolutional neural network that we developed in tensorflow using the keras API. It consist of a input layer, 4 hidden 1D convolutional layers and 3 1D max-pooling layers, a flatten layer, one dense layers and an output dense layer which also contains a softmax layer.The softmax serves as the classification function, and is given by the following equation:

$$y_i = \frac{\exp^{z_c}}{\sum_{c=1}^{C} \exp^{z_c}}, \tag{3.1}$$

where C is the total number of classes, z_c denotes the learned probability for a specific class c, and y_i is the final classification result for sample i.

We use rectified linear unit (Relu) function as activation function for all the hidden layer. Mathematically, it is by the following equation

$$y = max(0, x) \tag{3.2}$$

ReLU is the most commonly used activation function in neural networks, especially in CNNs. The summary of proposed model is given in Figure 3.1

3.1.1 Dataset

To test the proposed model in the report, we used a publicly available HAR dataset by WISDM group [15]. The dataset is contains data time-series data about six activities walking, jogging, walking upstairs, walking downstairs, sitting, and standing. While performing these activities, the sampling rate for accelerometer sensor was set to 20 Hz. Accelerometer provides data of three axis: x-axis, y-axis and z-axis. A sample for each axis is shown in Figure 3.2

Summary of the dataset is given in Table 3.1. It can see that the dataset is unbal-

TABLE 3.1: WISDM dataset Description

Class description	Number of Samples
Walking	424,400
Upstairs	122,869
Downstairs	100,427
Jogging	342,177
Sitting	59,939
Standing	48397

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 240, 1)]	0
input (Conv1D)	(None, 240, 64)	256
max_pooling1d (MaxPooling1D)	(None, 120, 64)	0
conv1d (Conv1D)	(None, 120, 32)	6176
max_pooling1d_1 (MaxPooling1	(None, 60, 32)	0
conv1d_1 (Conv1D)	(None, 60, 16)	1552
max_pooling1d_2 (MaxPooling1	(None, 30, 16)	0
conv1d_2 (Conv1D)	(None, 30, 8)	392
flatten (Flatten)	(None, 240)	0
dense (Dense)	(None, 118)	28438
dense_1 (Dense)	(None, 6)	714

Total params: 37,528
Trainable params: 37,528
Non-trainable params: 0

FIGURE 3.1: Proposed CNN classifier

anced and such type of data can cause performance degradation in the classification task. To address this issue we use upsampling to sample the classes which have fewer number of samples. The distribution of dataset before and after the upsampling is shown in Figure 3.3 and 3.4, respectively.

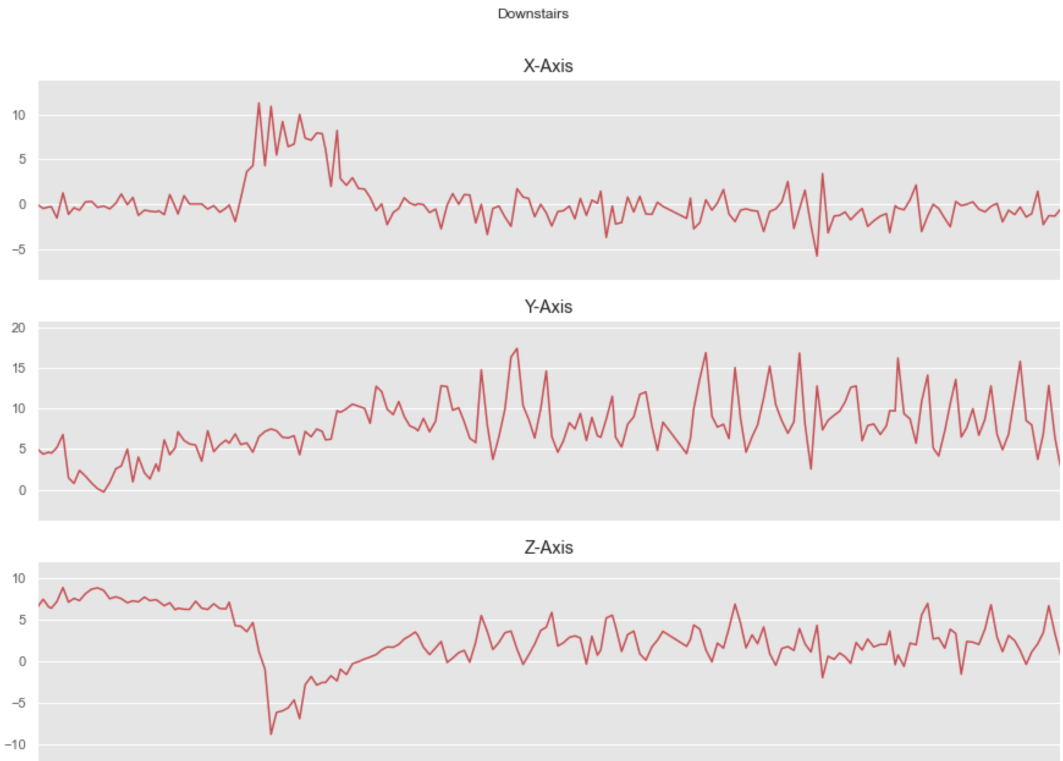


FIGURE 3.2: Sample Data

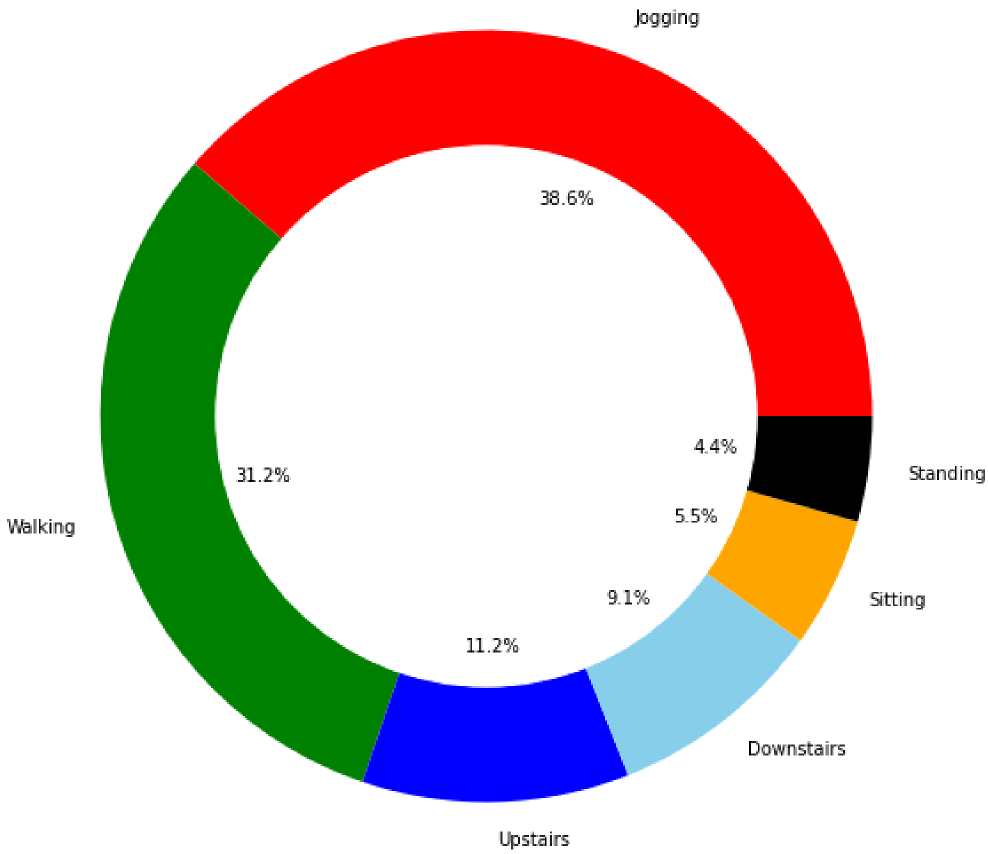


FIGURE 3.3: Dataset Distribution before Upsamling

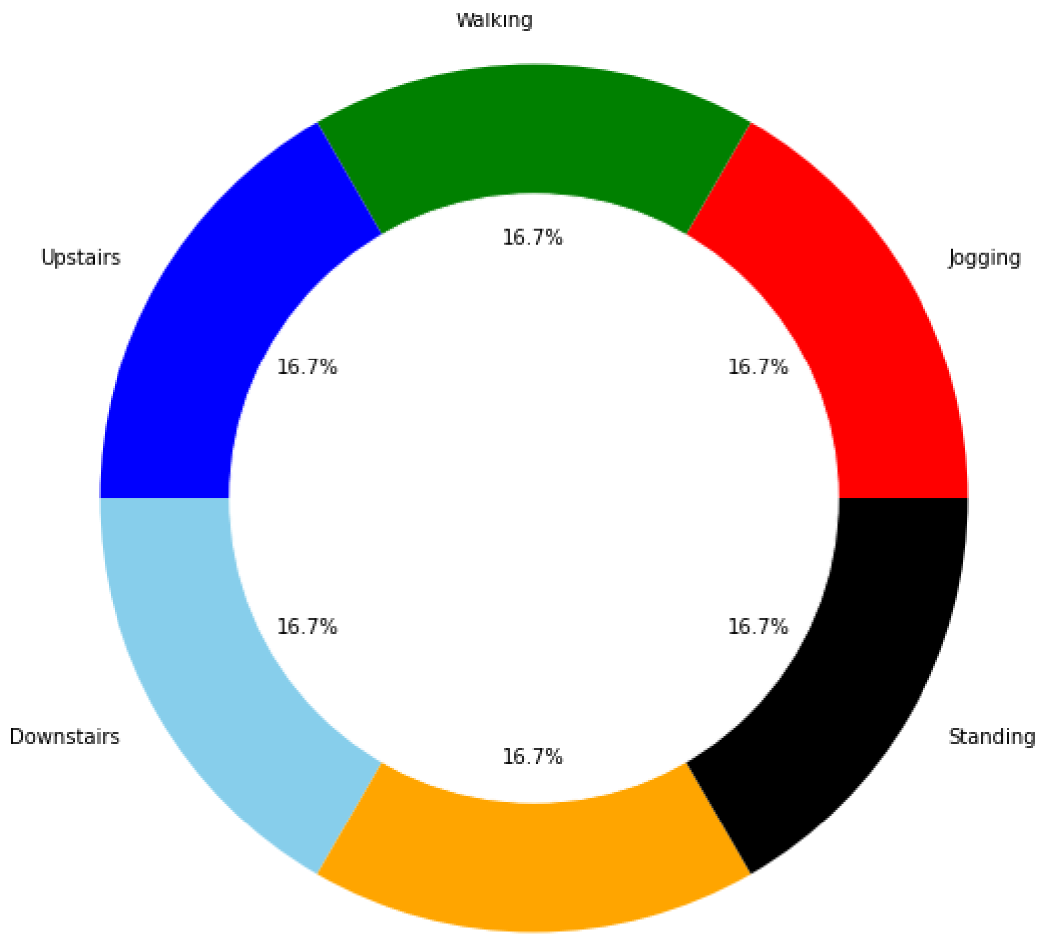


FIGURE 3.4: Dataset Distribution after Upsamling

Chapter 4

Experimental setup

4.1 Experimental setup

In this section we discuss the experimental setup. The training and testing of the proposed model is performed using a standard laptop with 2.4 Ghz CPU and 8GB RAM. Furthermore, we split the dataset by 80:20 ratio for training and testing, respectively. The classifier was trained for 50 training epochs using an *Adam* optimizer, with a learning rate of 0.01. Moreover, we use categorical crossentropy loss function to calculate the loss which is given a follow.

$$Loss = \sum_{i}^{Numberofsamples} y_i log(y_i^{'}) \tag{4.1}$$

Here y_i. is given label of a samplet input and $y_i^{'}$ is the predicted label of the given sample input. The training and validation accuracy of the proposed classifier is given in Figure 4.1. It can be observed that the validation accuracy increases until epoch 48 and starts decreasing after epoch 48, which means that the model can overfit if we train it for more epochs. Hence, it should not be trained for more than 50 epochs. Figure 4.2 shows the training and validation loss of the proposed classifier. It can be observed that the validation loss is no more decreasing after 48 epochs. Hence the training should be stopped at this point as further training can cause overfitting of the classifier.

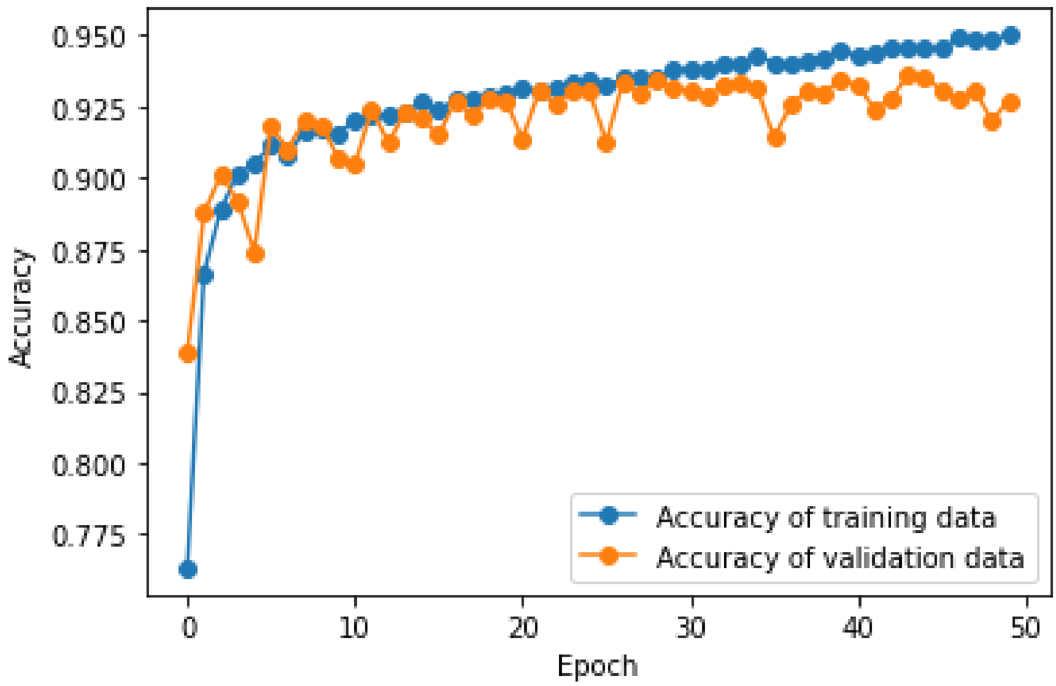


FIGURE 4.1: Training and Validation Accuracy of the Proposed Model

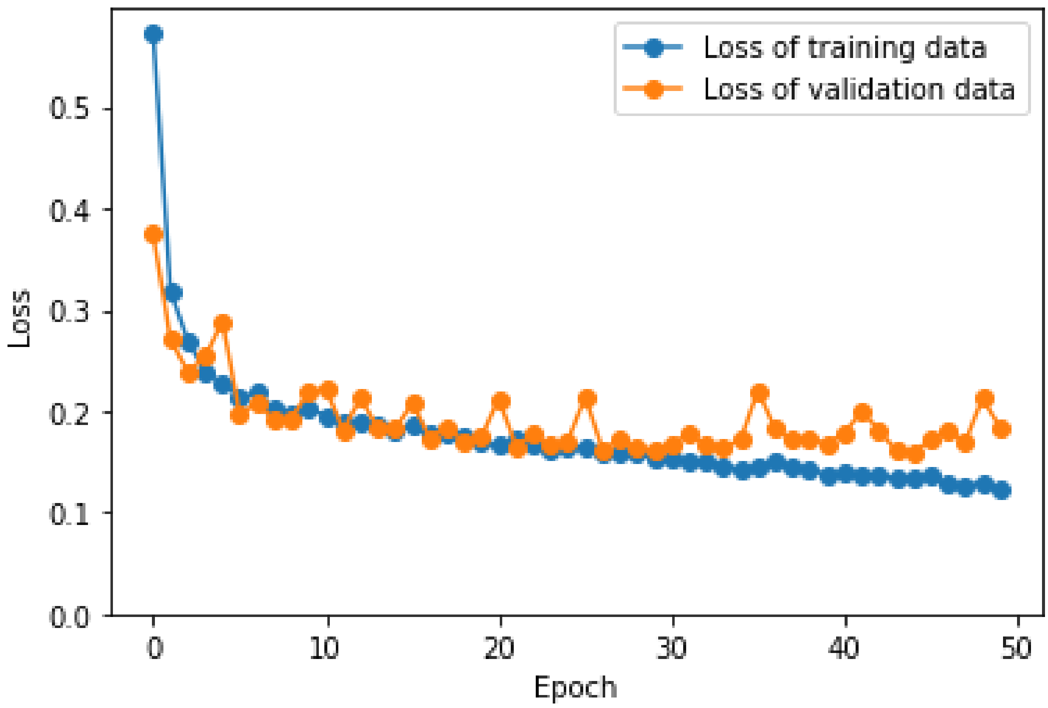


FIGURE 4.2: Training and Validation Loss of the Proposed Model

Chapter 5

Performance Analysis and Conclusions

5.1 Classification Performance Analysis

We measure the classification performance of the proposed classifier by using the state of art classification measuring metrices: Precision, recall, and F1-score, as shown in Figure 5.1. Class description and the respective Single-letter symbol is shown in Table 5.1. It can be seen the the proposed classifier provides 100% recall, precision and F1-score for class 1, 2, and 3. Moreover, recall, precision and F1-score for the remaining classes are acceptable. The overall accuracy of the classifier is 93%.

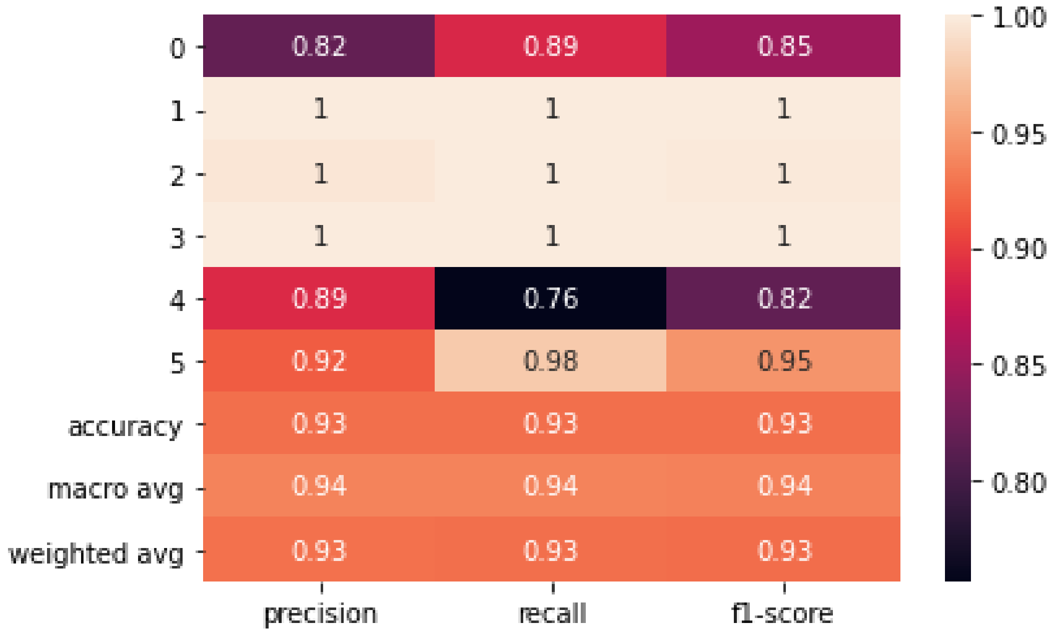


FIGURE 5.1: Classification Performance of the Proposed Model

5.2 Discussions and Conclusions

In this project we have proposed a convolutional neural network based classifier to classify the human activity into six classes. The proposed classifier provides an overall accuracy of 93%. Moreover, we also evaluated our proposed classifier using performance matrices like recall precision and F1-score, which showed that the proposed classifier also provides acceptable recall, precision and F1-score. Furthermore,

TABLE 5.1: The Six classes of Human Activity

Class description	Single-letter symbol
Walking	0
Upstairs	1
Downstairs	2
Running	3
Sitting	4
Standing	5

we also pointed out that, in future the algorithm could be improved by changing batch size or the learning rate. Also improving the model architecture by adding more hidden layers and empirically tuning the hyper-parameters can enhance the performance of model significantly.

Bibliography

[1] Yufei Chen and Chao Shen. “Performance analysis of smartphone-sensor behavior for human activity recognition”. In: *Ieee Access* 5 (2017), pp. 3095–3110.

[2] Yanjun Jia. “Diatetic and exercise therapy against diabetes mellitus”. In: *2009 Second International Conference on Intelligent Networks and Intelligent Systems*. IEEE. 2009, pp. 693–696.

[3] Jie Yin, Qiang Yang, and Jeffrey Junfeng Pan. “Sensor-based abnormal human-activity detection”. In: *IEEE Transactions on Knowledge and Data Engineering* 20.8 (2008), pp. 1082–1090.

[4] Oscar D. Lara and Miguel A. Labrador. “A Survey on Human Activity Recognition using Wearable Sensors”. In: *IEEE Communications Surveys Tutorials* 15.3 (2013), pp. 1192–1209. DOI: 10.1109/SURV.2012.110112.00192.

[5] Ulas Bagci and Li Bai. “A comparison of daubechies and gabor wavelets for classification of mr images”. In: *2007 IEEE International Conference on Signal Processing and Communications*. IEEE. 2007, pp. 676–679.

[6] Yichuan Tang, Ruslan Salakhutdinov, and Geoffrey Hinton. “Robust boltzmann machines for recognition and denoising”. In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 2264–2271.

[7] Oscar D Lara and Miguel A Labrador. “A survey on human activity recognition using wearable sensors”. In: *IEEE communications surveys & tutorials* 15.3 (2012), pp. 1192–1209.

[8] Andrea Mannini and Angelo Maria Sabatini. “Machine learning methods for classifying human physical activity from on-body accelerometers”. In: *Sensors* 10.2 (2010), pp. 1154–1175.

[9] Ben Nham, Kanya Siangliulue, and Serena Yeung. “Predicting mode of transport from iphone accelerometer data”. In: *Machine Learning Final Projects, Stanford University* (2008).

[10] Emmanuel Munguia Tapia, Stephen S Intille, and Kent Larson. “Activity recognition in the home using simple and ubiquitous sensors”. In: *International conference on pervasive computing*. Springer. 2004, pp. 158–175.

[11] Mohammed Mehedi Hassan et al. “A robust human activity recognition system using smartphone sensors and deep learning”. In: *Future Generation Computer Systems* 81 (2018), pp. 307–313.

[12] Carlos Avilés-Cruz et al. “Coarse-fine convolutional deep-learning strategy for human activity recognition”. In: *Sensors* 19.7 (2019), p. 1556.

[13] Google. URL: https://www.tensorflow.org/about.

[14] Pramod Singh and Avinash Manure. “Introduction to tensorflow 2.0”. In: *Learn TensorFlow 2.0*. Springer, 2020, pp. 1–24.

[15] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. "Activity recognition using cell phone accelerometers". In: *ACM SigKDD Explorations Newsletter* 12.2 (2011), pp. 74–82.