PROJECT PHASE 2
CSE 572
DATA MINING
SPRING 2018

SUBMITTED TO:
Professor Ayan Banerjee
Ira A. Fulton School of Engineering
Arizona State University

A Report by:
Athithyaa Selvam (aselvam@asu.edu)
Hari Siddarth Velaudampalayam Kesavan (hvelauda@asu.edu)
Mohan Vasantrao Yadav (mvasantr@asu.edu)
Raam Prashanth Namakkal Sudhakar (rnamakka@asu.edu)
Sangeetha Swaminathan (sswami11@asu.edu)
(Team 18)

TABLE OF CONTENTS

1. INTRODUCTION

The goal of this project is to develop a system to identify human actions by tracking muscle and arm movement using Myo Gesture Control Armband and Kinect data. This project will be useful to help people who are differently abled (speech impaired). The sensors that are used are Accelerometer (X, Y, Z axis), Orientation (X, Y, Z axis), Gyroscope and EMG. The data is then used to analyze an action performed and classify it.

2. DATA COLLECTION - PROJECT PHASE 1

The first phase of the project was to collect data of the different sensors mentioned above from IMPACT lab. One team member had to wear the Myo armband and perform a set of twenty different actions (About, And, etc.) from the American Sign Language (https://www.signingsavvy.com/). Every action was repeated twenty times with an interval of three seconds in between and the data was stored in unique csv files with the action names as file names (For example, “About102303AM.csv”).

3. FEATURE EXTRACTION AND FEATURE SELECTION - PROJECT PHASE 2

In the next phase of the project each group was required to extract prominent features from the raw sensor data collected in the previous phase. The following tasks were performed in this phase. The Kinect data was ignored for this phase.

3.1. Preprocessing

In this step, we cleaned up the raw data to derive useful data for processing. The files that had at least 10 rows were only selected for the feature extraction because, files with less than 10 rows imply that the data was not properly collected from actions performed or the actions were performed improperly. We come to this conclusion as a proper data for an action will take around 40 rows of data. For uniformity purposes, we padded zeros to the files that contained less than 40 rows of data. In files having more than 40 rows of data we used only the first 40 rows for processing as the remaining rows are noise data.

3.2. Segmentation - Task 1

The raw data has now to be scaled down to just 10 actions namely, "*about*", "*and*", "*can*", "*cop*", "*deaf*", "*decide*", "*father*", "*find*", "*go out*", and "*hearing*". The data collected for 20 actions by 37 groups (37 directories) are used for this task. The data corresponding to each gesture in every directory was transformed and combined into a single csv file. The time series of each action was stored column-wise and every row has the data of the given sensors. Multiple actions were appended at the end of each action. At the end of this task we have 10 csv files corresponding to each action with 40 columns in each file.

The file *code/task1.m* generates the 10 csv files from the raw data discussed above.

3.3. Feature Extraction - Task 2

The behavior of each signal for different actions were analyzed from the plots drawn after performing certain existing feature extraction methods. We do this to find the features that show clear distinction between the actions.

a) The feature extraction methods that were used are:
   1. Statistical Feature - Minimum
   2. Statistical Feature - Maximum
   3. Statistical Feature - Root Mean Square
   4. Statistical Feature - Standard Deviation
   5. Fast Fourier Transform

Table 3.3.1. The selected features and the corresponding extraction methods

Feature	Extraction Method
Accelerometer X axis of left hand - ALX	Maximum
Accelerometer Y axis of right hand - ARY	Minimum
Orientation roll of left hand - ORL	Root Mean Square
EMG 0 pod of left hand - EMG0L	Fast Fourier Transform
Gyroscope X axis of left hand - GLX	Fast Fourier Transform

Minimum: This function returns the cell with the minimum value in a record.

Maximum: This function returns the cell with the maximum value in a record.

Root Mean Square: The function calculates the square root of the arithmetic mean of squares of a set of records.

$$RMS(ORL) = \sqrt{\frac{ORL_1{}^2 + ORL_2{}^2 + \cdots + ORL_n{}^2}{n}} \tag{3.3.1}$$

Standard Deviation: This function performs square root of variance where variance is determined by calculating variation between each data point relative to mean.

Fast Fourier Transform: This function is applied to calculate Discrete Fourier transform using Fast Fourier Transform algorithm.

b) Intuition to use these features:

We computed several features from the raw data and analyzed their efficiency in discriminating among all the ten classes. There are several ways to perform this kind of analysis. Some of the obvious ways include checking for a continuous increase or decrease in the pattern if the pattern is completely random or if it includes some cyclic movements. We have computed maximum, minimum, standard deviation, Root Mean Square and Fast Fourier Transform for all the sensors and picked the ones which showed a good discrimination.

Accelerometer X Axis on Left Hand - Maximum:

The graph in Figure 3.3.1 shows that the feature extracted for “can” increases and decreases significantly. Compared to all the other patterns found, this feature has significant peaks which is not found in the other features. The feature for “deaf” is mostly in between all the other features and it can also be significantly differentiated from the others. The feature for “decide” has an almost constant pattern throughout. Based on these, we chose this feature assuming that it will serve well for classification purpose.

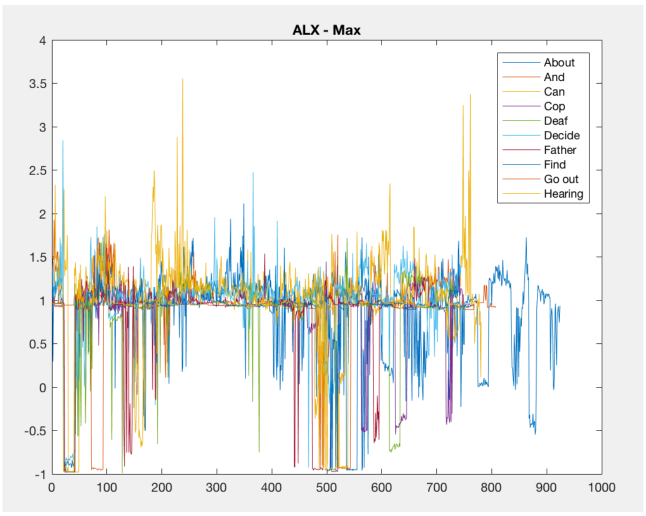

Figure 3.3.1. Plot of Maximum Feature Extraction Performed on ALX

Accelerometer Y Axis on Right Hand - Minimum:

For accelerometer Y axis of right hand, Min gave the best results. We can clearly see from Figure 3.3.2 that features for "can" and "find" actions decrease notably compared to the other feature patterns.

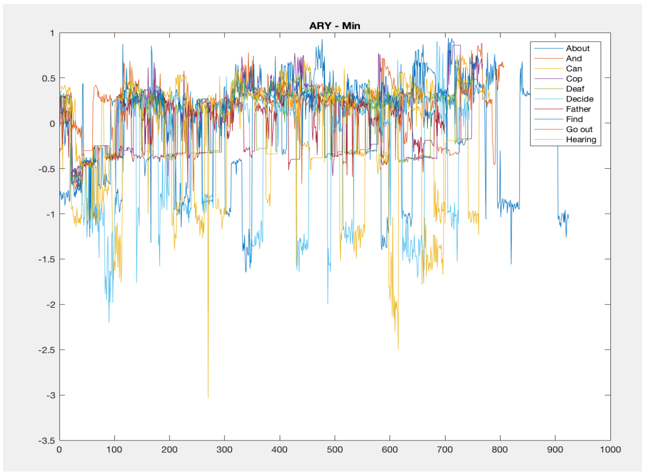

Figure 3.3.2. Plot of Minimum Feature Extraction Performed on ARY

Since it is very hard to visualize the difference when multiple actions are plotted on the same graph, we also plotted two actions at a time to better analyze the data. We can easily see the difference between father and find in Figure 3.3.3(a), deaf and hearing in the Figure 3.3.3(b) below.

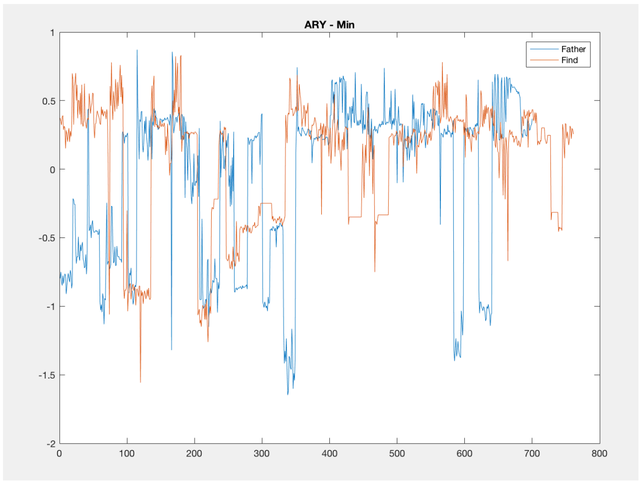

Figure 3.3.3(a). Plot of Minimum Feature Extraction Performed on ARY on gestures “father” and “find”

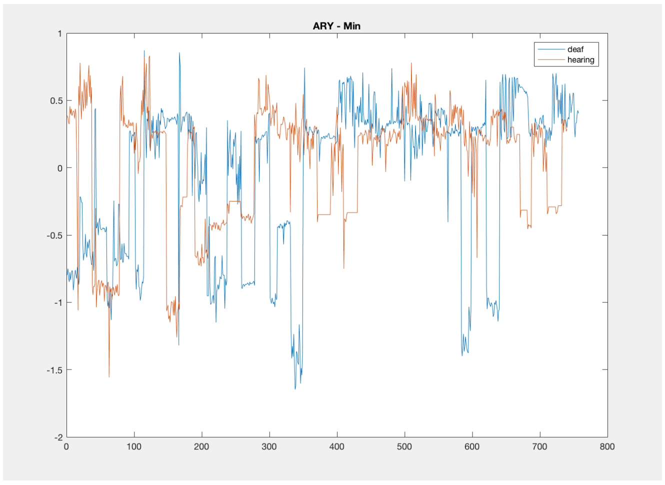

Figure 3.3.3(b). Plot of Minimum Feature Extraction Performed on ARY on gestures “deaf” and “hearing”

Orientation roll on Left Hand – Root Mean Square:

Figure 3.3.4 shows notable variance among the actions after performing RMS feature extraction on ORL sensor. By looking at the variance in the graph we can classify the different actions.

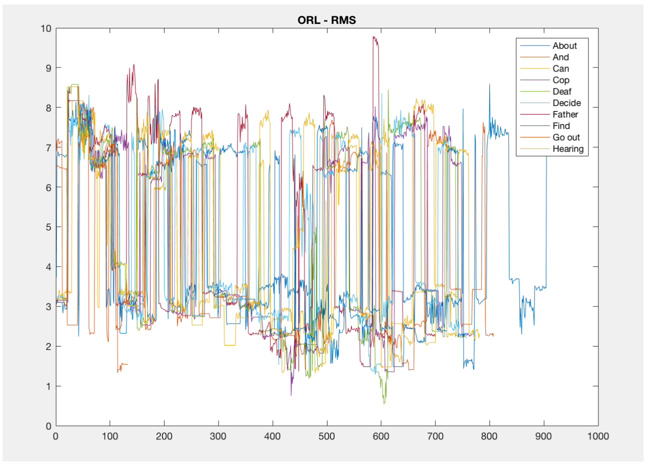

Figure 3.3.4. Plot of Root Mean Square Feature Extraction Performed on ORL

EMG0 pod on Left Hand – Fast Fourier Transform:

In the Figure 3.3.5 below, we can observe the pattern obtained by doing FFT of the EMG signal. The feature for hearing varies a lot in the below graph. By looking at the level of increase and decrease in the series, we will be able to differentiate between the several actions.

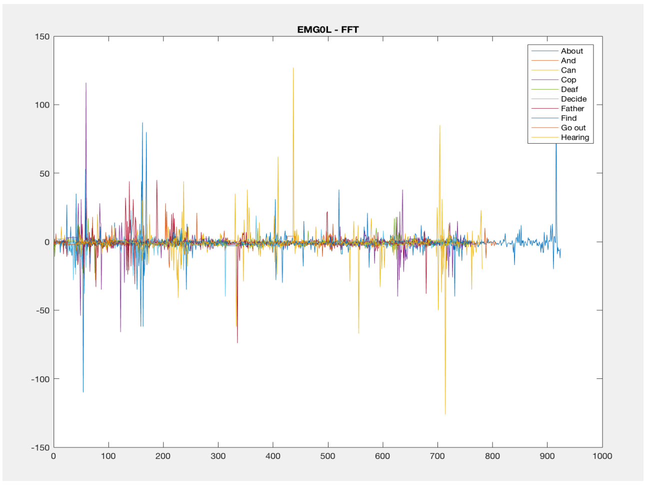

Figure 3.3.5. Plot of Fast Fourier Transform Feature Extraction Performed on EMG0L

When we extracted some features, we found that some didn't show any significant difference. Figure 3.3.6 shows an example of a feature extraction on EMG2R sensor which doesn't show any prominent changes and hence will not be of much use.

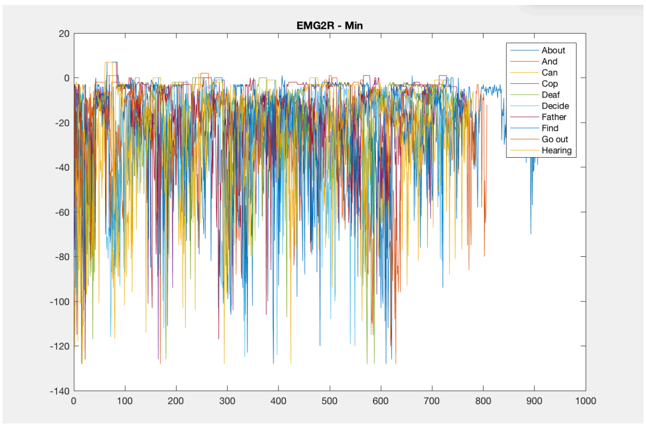

Figure 3.3.6. Plot of Minimum Feature Extraction Performed on EMG2R

c) A MATLAB code – *code/task2.m* is written to extract the features using the feature extraction techniques defined above. In built functions "*min*", "*max*", "*std*", "*rms*", and "*fft*" are used for the same.

d) Running *code/task2.m* generates 10 plots each corresponding to a gesture. These plots are saved in the folder *task2-D* with respective action names.

e) After analyzing the outputs and respective plots of each feature extraction, our intuition to select the said features holds true.

3.4. Feature Selection - Task 3

The aim of this task is to change the original feature space into a reduced feature space with just the features that show the maximum distance between different classes. To achieve this, we are using Principle Component Analysis that is discussed in the section below.

Principle Component Analysis:

It is the most commonly used dimensionality reduction algorithm. It is also called Eigen analysis or Eigen decomposition algorithm. The main goal of PCA is to preserve variance in the data even after reducing dimensions. It uses eigenvalues to determine which feature is more significant in determining the class. If the eigenvalue has a high value, it means that the corresponding eigenvector describes the data very well and the removal of that particular eigenvector will not help in preserving variance. If the variance is not preserved, we won't be able to classify the actions. We generally perform PCA to reduce the amount of data processing and to increase the speed of computation.

The input data to the PCA is the covariance of the feature matrix containing the features of all the actions. The output of PCA returns a set of eigenvalues and eigenvectors. The eigenvectors with high eigenvalues are retained and the ones with poor eigenvalues are thrown out. The resulting eigenvectors obtained is multiplied with the original data to get the resultant matrix which represents the transformation of the original space. This matrix can be further used for classification.

The final matrix obtained through PCA is split on the basis of actions and saved separately.

The file *code/task3.m* arranges the feature matrix and the pca function is run on this matrix. This gives the eigenvector and eigenvalues of the feature matrix and generates 10 new feature matrices for every action.

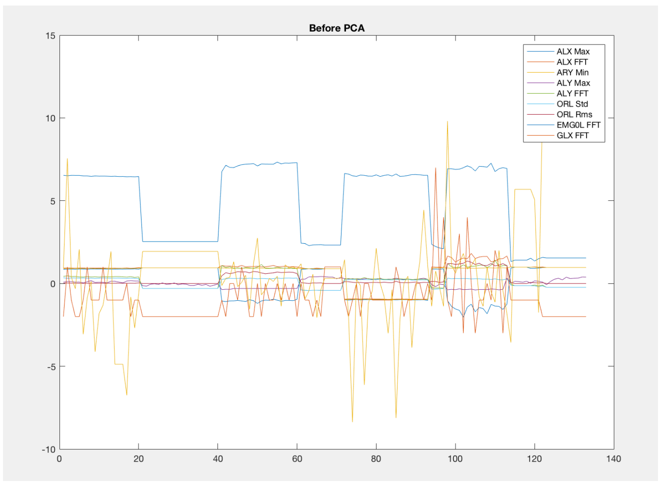

Figure 3.4.1(a). Plot of all features extracted for the action "go out" before applying PCA

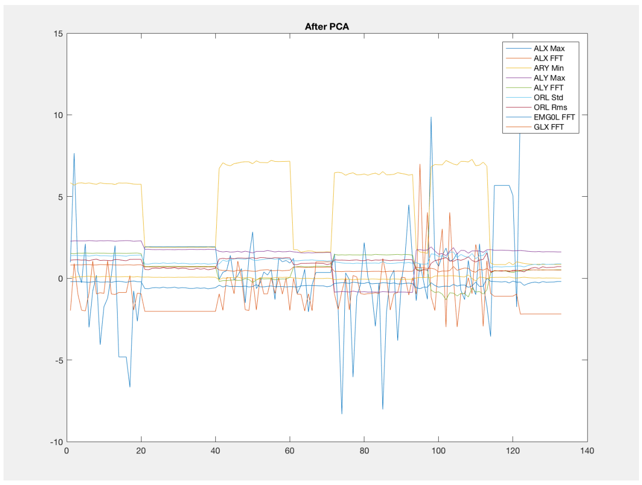

Figure 3.4.1(b). Plot of all features extracted for the action "go out" after applying PCA

Figure 3.4.1(a) shows the plot of all features extracted for the action "go out" before applying PCA and Figure 3.4.1(b) shows the plot of all features extracted for the action "go out" after applying PCA. On comparing these two plots we observe that the transformation of the original space looks very similar to the reduced space. We can conclude that the variance of the data is preserved even after applying dimensionality reduction and doing pca was helpful.