Human Gesture Recognition
Phase II Report
Data Mining
CSE 572
Fall 2018

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

Submitted By: Group 20
Sam Sunder Golla (sgolla2@asu.edu)
Aravind Manoharan (amanoh10@asu.edu)
Anto Oswin Nihal (anihal@asu.edu)
Pravin Kumar Ravi (parvi8@asu.edu)
Madhu Venkatesh (mvnka24@asu.edu)

Table of Contents

1. Introduction

This academic project is a part of the fall 2018 course - Data Mining (CSE 572) at the IRA Fulton School of Engineering at ASU. In this project, we attempt to develop an intelligent system that would be able to accurately recognize human gestures, upon its completion. We used two Myo Gesture Control Armbands (one for each hand), present at the Impact Lab at ASU, to gather gesture data for each hand, and later analyzed and processed the data to obtain logical patterns within. This report describes our progress while working on the second phase of the project.

2. The Team

The team consists of the following members:

- Sam Sunder Golla
- Aravind Manoharan
- Anto Oswin Nihal
- Pravin Kumar Ravi
- Madhu Venkatesh

3. Phase 1: Data Collection

In the first phase of the project, we recorded the Myo sensor readings for each hand. One of the team members, Aravind, was chosen to perform the 10 gestures (and, about, can, cop, deaf, decide, father, find, go out, hearing), and performed each gesture 20 times (i.e 20 actions for each gesture) wearing one Myo sensor on each hand. The Myo sensor is equipped with the following sensors to record the hand movements: Accelerometer sensor, Electromyography sensor, Gyroscope sensor, and Orientation sensor. The sensor readings for each action of a gesture is stored in a file (20 actions x 10 gestures x 37 groups = 7400 total files) to serve as input for phase 2.

ALX	ALY	ALZ	ARX	ARY	ARZ	EMG0L	EMG1L	EMG2L	EMG3L	EMG4L	EMG5L	EMG6L	EMG7L	EMG0R	EMG1R
0.90625	-0.3994141	-0.0898438	0.8657227	0.4584961	-0.0112305	-2	-1	0	-2	-1	-1	-1	-1	-2	1
0.9008789	-0.3994141	-0.0820313	0.862793	0.4521484	-0.0195313	-2	-2	0	-1	1	0	-3	-1	-2	-1
0.9145508	-0.3989258	-0.0859375	0.8598633	0.4560547	-0.0205078	0	-3	0	-2	-2	-1	-1	-2	1	0
0.9111328	-0.4047852	-0.0888672	0.8642578	0.4506836	-0.0253906	0	-2	-2	-2	-1	-1	1	0	-1	-1
0.9072266	-0.394043	-0.0908203	0.871582	0.4487305	-0.03125	-2	-1	-1	-2	0	-3	-1	-1	0	2
0.9106445	-0.3969727	-0.0917969	0.878418	0.4609375	-0.019043	0	-1	-1	-1	-1	-3	-4	-2	0	-1
0.9873047	-0.4921875	-0.0986328	0.9023438	0.4736328	-0.0351563	5	1	-3	-1	-2	-1	-2	-1	0	0
0.9711914	-0.4492188	0.02050781	0.871582	0.4765625	0.02197266	1	1	-1	1	5	1	1	-2	-2	-1
0.8862305	-0.6074219	0.02685547	0.859375	0.496582	0.03759766	-1	0	-1	1	-3	-4	-1	-2	-2	1
0.9228516	-0.6616211	0.1074219	0.8339844	0.4829102	0.06591797	1	0	0	-2	0	1	-1	0	-1	0
0.5668945	-0.7431641	0.1953125	0.7822266	0.5092773	0.08203125	-2	1	-1	0	2	0	-1	0	0	-2
0.3950195	-0.4155273	0.05810547	0.7822266	0.4829102	0.01708984	0	0	-3	0	0	-2	-1	-1	0	0
0.5141602	-0.5463867	0.09423828	0.8632813	0.5678711	-0.0341797	-1	-2	-1	-1	-1	-3	0	-1	-1	0
0.8173828	-0.7539063	0.1108398	1.01709	0.6083984	0.06738281	-4	-1	-2	-3	-3	-3	0	-1	-1	-3
1.033691	-1.038086	0.3256836	0.9882813	0.6567383	0.1777344	0	-2	-2	-2	-3	-1	-3	-1	1	4
0.7749023	-1.138184	0.2827148	0.8452148	0.7387695	0.2226563	-5	-2	-6	-11	-6	-8	-4	1	-1	-5
0.440918	-1.011719	0.2973633	0.5605469	0.6489258	0.3442383	0	-3	-2	-3	-2	16	11	3	0	-10
0.1982422	-0.6298828	0.2626953	0.5917969	0.7041016	0.4116211	-3	-4	2	0	-6	-15	-9	-4	-5	8
0.09765625	-0.4799805	0.1547852	0.4204102	0.7836914	0.5180664	-4	-4	-7	-2	-5	-10	3	4	-2	2
0.05371094	-0.8149414	0.2802734	-0.0024414	0.6308594	0.2558594	6	5	1	-1	-7	-15	-3	13	3	5
0.05810547	-0.8237305	0.2807617	0.1230469	0.7041016	0.3217773	-3	-2	0	-1	1	9	-17	-6	3	3
0.1015625	-1.047852	0.3178711	0.03808594	1.300781	0.644043	0	-3	-2	-2	-3	-5	17	5	10	1
0.1000977	-1.055176	0.3491211	-0.4399414	0.7421875	0.4916992	-2	2	1	-2	4	6	7	2	2	4

Fig. 1: The sensor readings for one of the gestures

4. Phase 2 : Feature Recognition

Phase 2 was divided into 3 subtasks data synchronization, feature extraction and feature selection. The raw sensor data from phase 1 was processed and reformatted so that the feature extraction and feature selection tasks can be performed on it. We now describe each of the subtasks in more detail:

4.1 Task 1 : Data Synchronization

The objective of task 1 is to reshape the raw sensor data files from phase 1, and reformat it to a format that can be easily read by different feature extraction algorithms. To do so, the sensor data of all the actions of a gesture were grouped together into one file and the resulting data was transposed so that the columns indicate the time series data and each row corresponds to a different sensor of the Myo sensor. Each action was recorded for 3 seconds at 15 Hz frequency due to which there should be a maximum of 45 columns. If an action has more than 45 columns, we chose to not add it to the consolidated file because we consider that action as erroneous. If an action had less than 45 columns, we zero padded the missing columns and added it to the consolidated file. We don't consider such actions as erroneous because in all such actions, only one or two columns were missing, zero padding which, does not cause the data to become inaccurate.

Action1ALX	0.943848	0.95166	0.95459	0.95459	0.949707	0.956543	0.955078	0.948731	0.961914	1.070313	1.193848	1.049805	0.644531	0.322754	0.329102
Action1ALY	-0.27979	-0.25244	-0.2627	-0.26465	-0.2627	-0.23633	-0.25146	-0.25684	-0.25684	-0.22607	-0.2583	-0.48242	-0.68359	-0.51709	-0.5708
Action1ALZ	0.12207	0.12793	0.12207	0.127441	0.116211	0.122559	0.121582	0.109375	0.111328	0.006836	0.103516	0.444824	0.470703	0.455566	0.44043
Action1ARX	0.918457	0.918457	0.918457	0.918457	0.918457	0.918457	0.918457	0.918457	0.918457	0.918457	0.918457	0.918457	0.918457	0.918457	0.918457
Action1ARY	0.310059	0.310059	0.310059	0.310059	0.310059	0.310059	0.310059	0.310059	0.310059	0.310059	0.310059	0.310059	0.310059	0.310059	0.310059
Action1ARZ	0.130371	0.130371	0.130371	0.130371	0.130371	0.130371	0.130371	0.130371	0.130371	0.130371	0.130371	0.130371	0.130371	0.130371	0.130371
Action1EMG0L	0	-1	-2	-1	-3	1	-2	-1	-1	-8	-2	5	1	-6	-3
Action1EMG1L	5	2	0	-3	-1	2	-1	-1	0	0	1	-1	3	-2	-3
Action1EMG2L	1	-4	1	-2	0	1	-2	-2	0	0	0	-15	2	-1	-13
Action1EMG3L	-3	-1	-2	0	0	-1	0	-1	-4	-3	-10	3	-1	-5	-6
Action1EMG4L	0	1	-2	-1	-2	-1	-1	-2	-1	13	-7	1	-4	5	-3
Action1EMG5L	-2	-2	1	-1	-2	-1	-1	0	0	-3	-3	1	-4	5	2
Action1EMG6L	-6	1	0	-1	-1	-1	3	0	-4	-6	-4	-6	0	-3	2
Action1EMG7L	-5	2	-3	-3	3	-3	-1	4	-1	-3	-3	-4	0	3	2
Action1EMG0R	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
Action1EMG1R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 2: Snapshot of the consolidated file for 'about' gesture showing the first 15 Hz of the sensor recording

The acronyms indicate the following sensor names:

Sensor	Full Name
ALX	Accel X axis of Left hand
ALY	Accel Y axis of Left hand
ALZ	Accel Z axis of Left hand

ARX	Accel X axis of Right hand
ARY	Accel Y axis of Right hand
ARZ:	Accel Z axis of Right hand
EMG0L	EMG 0 pod of Left hand
EMG1L	EMG 1 pod of Left hand
EMG2L	EMG 2 pod of Left hand
EMG3L	EMG 3 pod of Left hand
EMG4L	EMG 4 pod of Left hand
EMG5L	EMG 5 pod of Left hand
EMG6L	EMG 6 pod of Left hand
EMG7L	EMG 7 pod of Left hand
EMG0R	EMG 0 pod of Right hand
EMG1R	EMG 1 pod of Right hand
EMG2R	EMG 2 pod of Right hand
EMG3R	EMG 3 pod of Right hand
EMG4R	EMG 4 pod of Right hand
EMG5R	EMG 5 pod of Right hand
EMG6R	EMG 6 pod of Right hand
EMG7R	EMG 7 pod of Right hand
GLX	Gyroscope X axis of Left hand
GLY	Gyroscope Y axis of Left hand
GLZ	Gyroscope Z axis of Left hand
GRX	Gyroscope X axis of Right hand
GRY	Gyroscope Y axis of Right hand
GRZ	Gyroscope Z axis of Right hand
ORL	Orientation roll of Left hand
OPL	Orientation Pitch of Left hand
OYL	Orientation Yaw of Left hand
ORR	Orientation roll of Right hand
OPR	Orientation Pitch of Right hand
OYR	Orientation Yaw of Right hand

4.2 Task 2: Feature Extraction

The objective of Task 2 is to represent actions performed in the feature domain. Features are attributes which helps us to identify and distinguish different actions/observations. For example we can use height, weight, body-to-mass ratio, skin color, nationality, hair/eye color as features to identify people. To extract features we use feature extraction methods such as FFT, DWT, PWD and statistical features like RMS and STD. we used inbuilt MATLAB libraries to perform the various feature extraction methods.

To implement this task, the following feature extraction methods were chosen.

4.2.1) Fast Fourier Transform (FFT)
4.2.2) Power Spectral Density (PSD)
4.2.3) Discrete Wavelet Transform (DWT)
4.2.4) Root Mean Square (RMS)
4.2.5) Standard Deviation (SD)

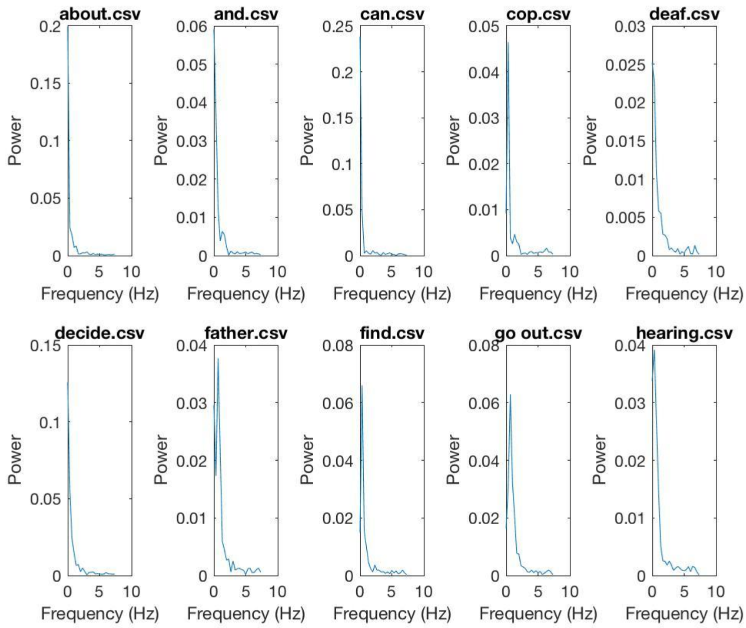
4.2.1 Fast Fourier Transform (FFT)

a) FFT helps us to convert time domain data into frequency domain data. It portrays peaks which are proportional to the power of the signal at that frequency. We can use these properties to extract the maximum power from different sensor's data signal in frequency domain. We use MATLAB '*FFT*' function to transform several sensors temporal data into the frequency domain. The objective of features is to identify each action by its own unique characteristics. Instead of handcrafting each feature, we wrote a script to identify signals that are most distinguishable than others. It means that we need to somehow identify sensors which are unique from others among all the gestures. We used correlation coefficient to achieve this. A signal which has highest negative correlation coefficient will be more unique than others. We chose 5 best attributes and extract features from them. By default, *FFT* function in MATLAB will perform discrete Fourier transform and outputs double sided power spectrum. Since input is not exactly periodic, output comprises of multiple peaks. We considered only single sided power spectrum and chose the 3 peak values as feature values. For each chosen sensor, we selected 3 maximum values. So we will have 5x3 = 15 feature values from FFT method for each action.

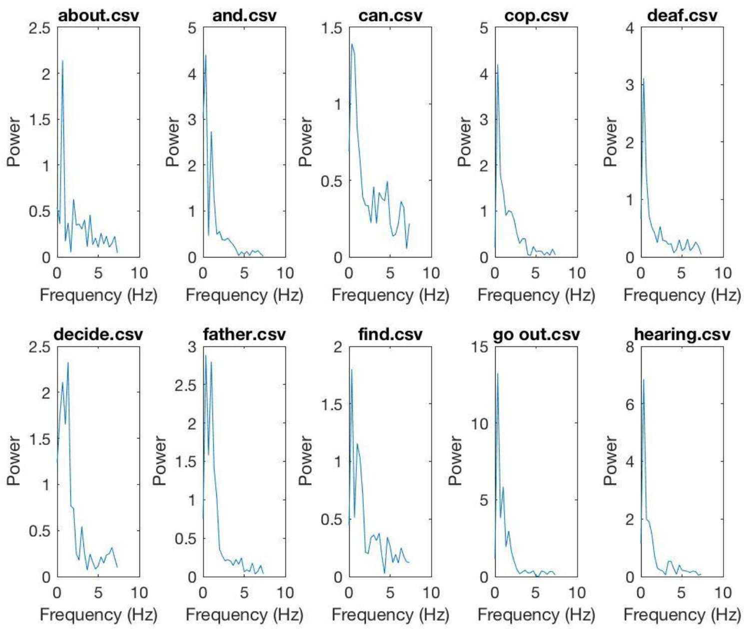
b) The intuition behind using FFT is to choose some interesting points in frequency domain which distinguish all the gestures. The Myo sensor which has different inbuilt sensors like Gyrometer, Accelerometer and EMG will collect different values for different gestures, each sensor will play a prominent role. It means, for '*about*' action orientation matters more than hand movement. Similarly, for '*cop*' action both angular velocity and EMG are dominant. We need to generalize in such a way that chosen sensor readings differ for each action. Our script identifies 5 sensor readings which helps us to distinguish various gestures. Hence, choosing 3 peak values in frequency domain for the chosen attributes supports the intuition.

c) ‘*FFT_Method.m*’ file has relevant code for feature extraction using FFT. The FFT transformation of the selected features can be seen in the plots below. We can observe that it is different for each gesture.

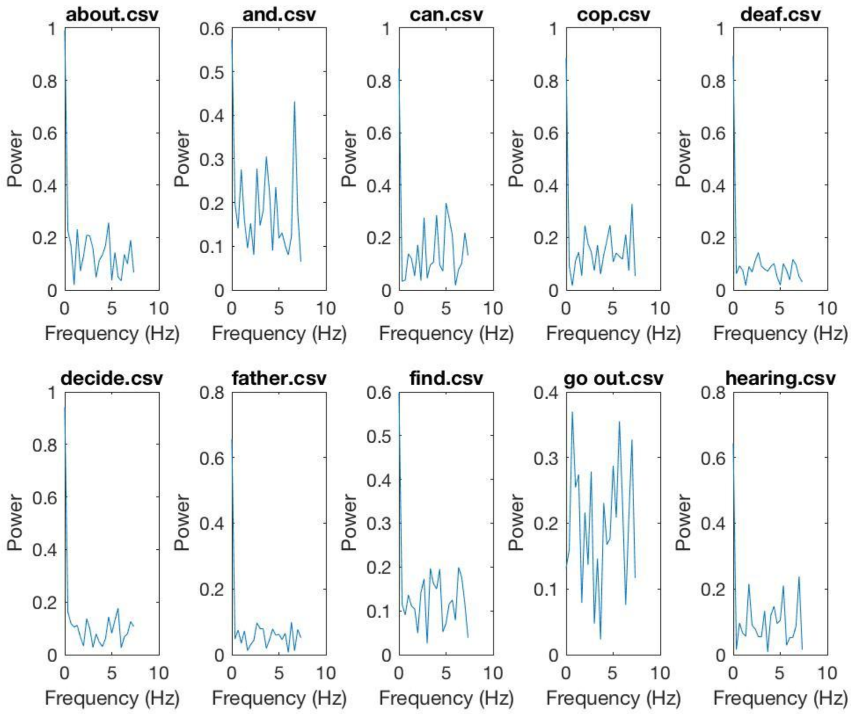
ALY



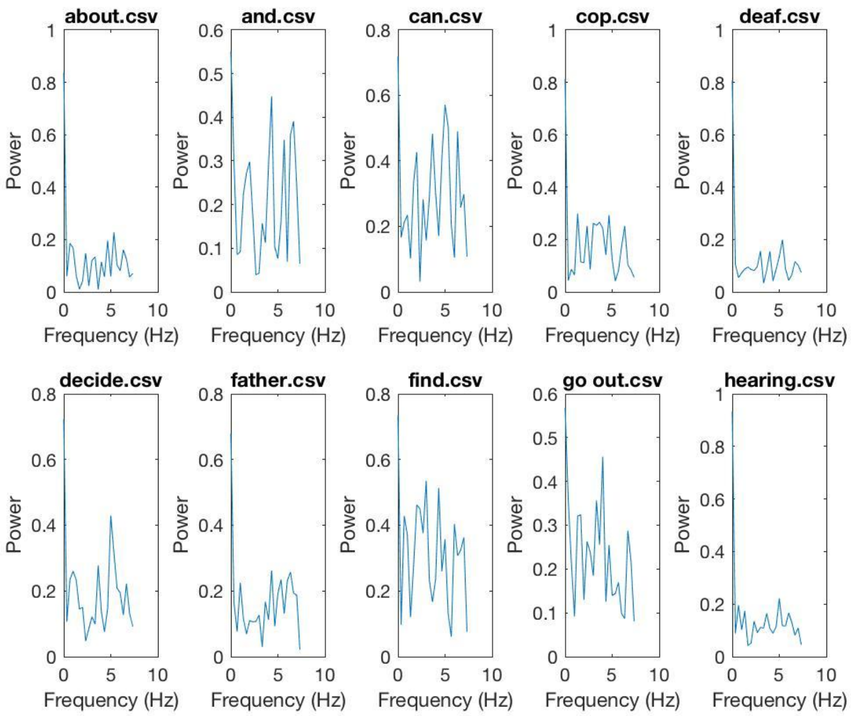
EMG0R



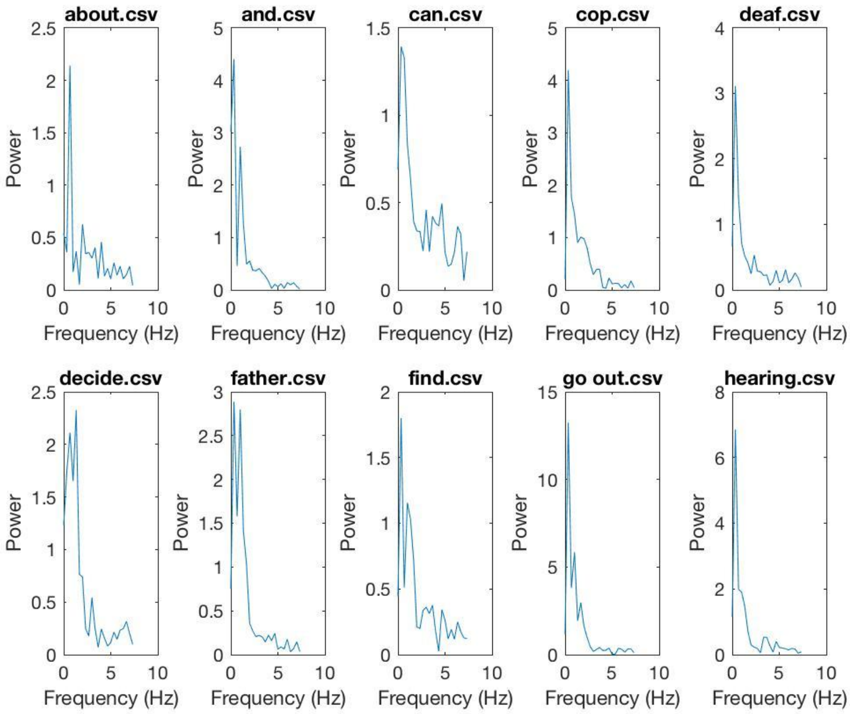
EMG4R


about.csv
and.csv
can.csv
cop.csv
deaf.csv
decide.csv
father.csv
find.csv
go out.csv
hearing.csv
Power
Frequency (Hz)

GLY


about.csv
and.csv
can.csv
cop.csv
deaf.csv
decide.csv
father.csv
find.csv
go out.csv
hearing.csv
Power
Frequency (Hz)

GLX



Analysis of Output

Using the script FFT_Method.m, in which we used the concept of correlation coefficient to get the most distinguishable sensors i.e. the script will show us which among the 34 attributes, helps us to identify the gesture (i.e. FFT of those attributes will be most distinguishable). We observed that ALY, EMG0R, EMG4R, GLY, GLZ are the most distinguishable features using FFT feature extraction method. The following plots verify our claim.

Please refer the output folder to see the plots of the selected attributes.

Conclusion

Since ALY, EMG0R, EMG4R, GLY, GLZ helps us to distinguish the gestures, we will use them as features for every action i.e. top 3 peak values of each selected attributes.

4.2.2 Power Spectral Density (PSD)

a) The power spectral density converts time domain sensor data into frequency domain and shows which frequencies have high and low variations. We used the dspdata.psd function in MATLAB to produce the power spectral density for each attributes. The function outputs the signal in the frequency domain and we take the first three peak values since it implies those frequencies which have high variations. Hence the resulting matrix is N x 5.

b) The intuition behind using power spectral density as one of our feature extraction method is that it distributes the power of the time series data into frequency components. In other words, PSD shows strong and weak variations for specific frequencies. So, after applying PSD to a signal, the peak values shows the strong variations of that signal and the least values shows the weak variations of that signal. Hence, we have decided to go with the first three peak values of the attributes which shows more distinction among all the gestures.

c) The relevant MATLAB code is *PSD_Method.m*

d) The attributes which are more distinct from each other are EMG4L, EMG0L, EMG1L, EMG3R and EMG5L. This implies that the EMG sensors' values are very distinct from each other.

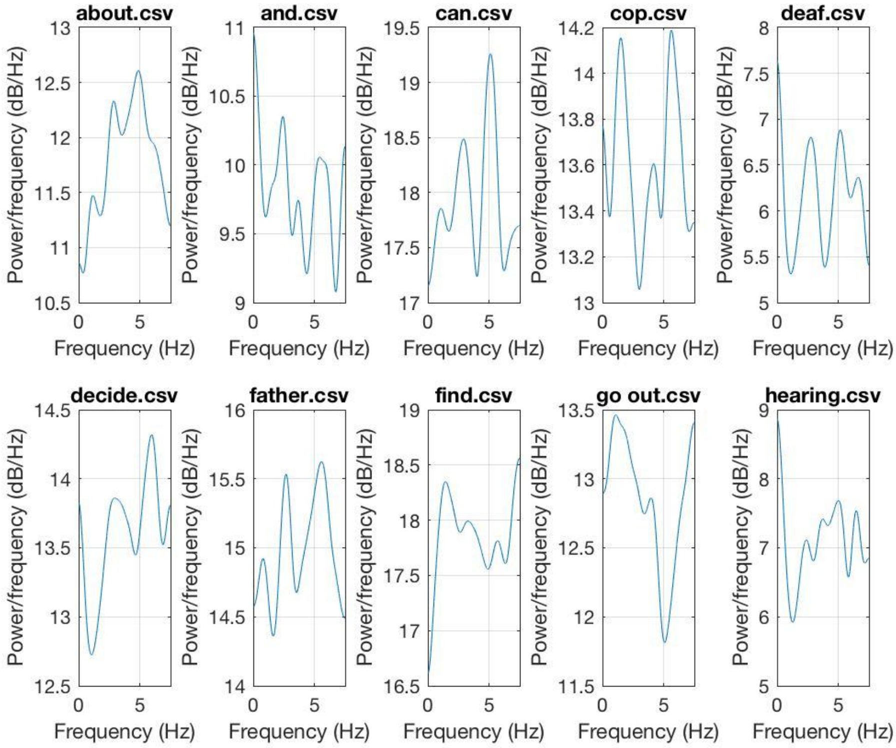
Analysis of Output

Using the script PSD_Method.m, in which we used the concept of correlation coefficient to get the most distinguishable sensors i.e. the script will show us which among the 34 attributes, helps us to identify the gesture (i.e. PSD of those attributes will be most distinguishable). We observed that EMG4L, EMG0L, EMG1L, EMG3R and EMG5L are the most distinguishable features using PSD feature extraction method. The following plots verify our claim.
Please refer the output folder to see the plots of the selected attributes.

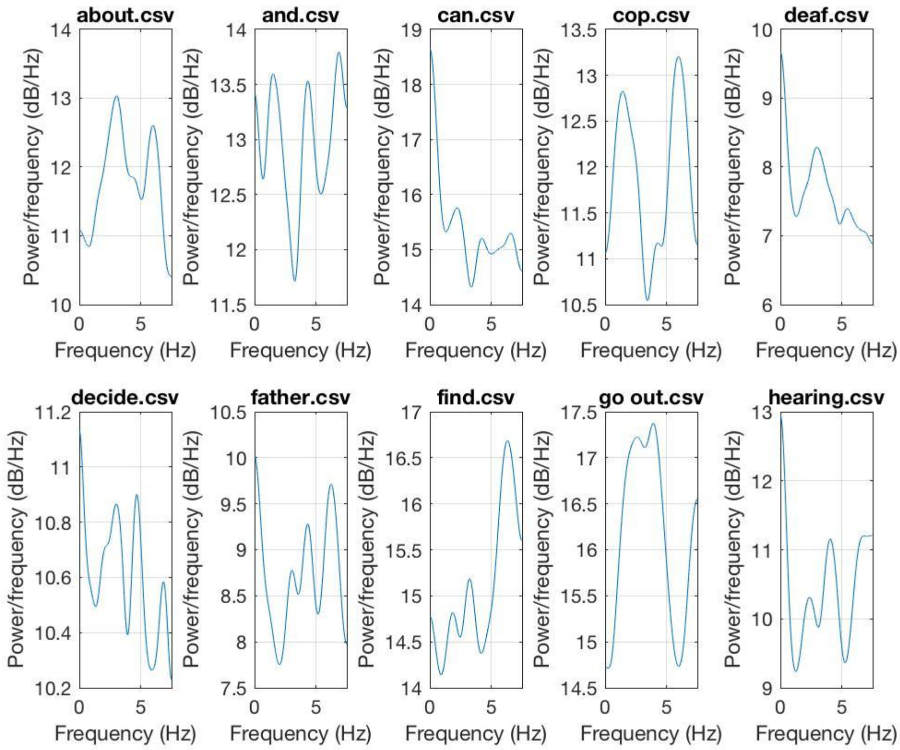
Conclusion

Since EMG4L, EMG0L, EMG1L, EMG3R and EMG5L helps us to distinguish the gestures, we will use them as features for every action i.e. top 3 peak values of each selected attributes.

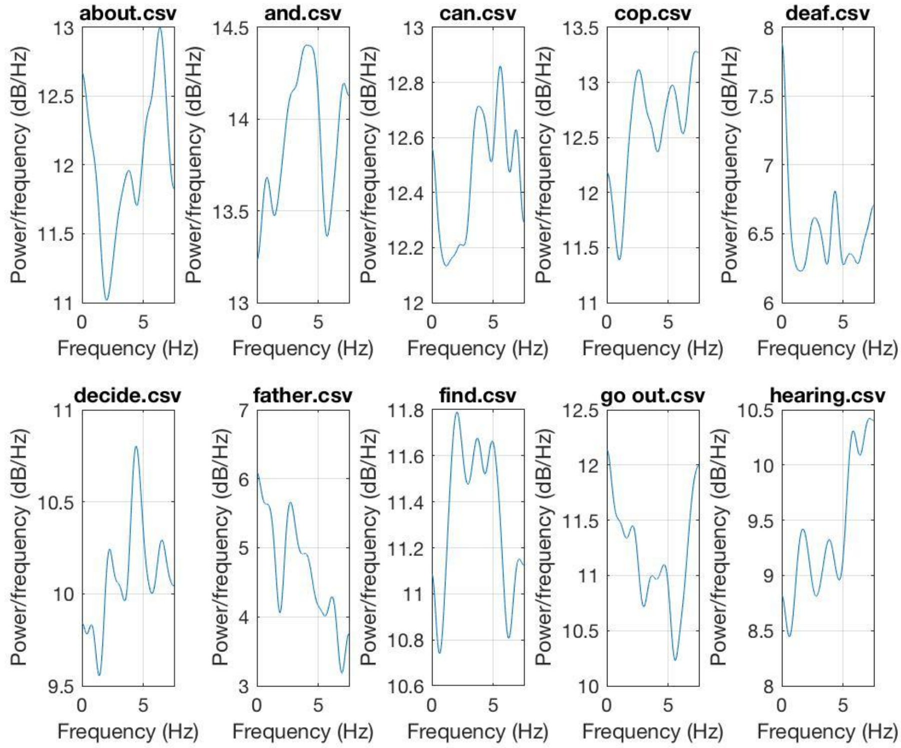
EMG0L


about.csv
and.csv
can.csv
cop.csv
deaf.csv
decide.csv
father.csv
find.csv
go out.csv
hearing.csv
Power/frequency (dB/Hz)
Frequency (Hz)

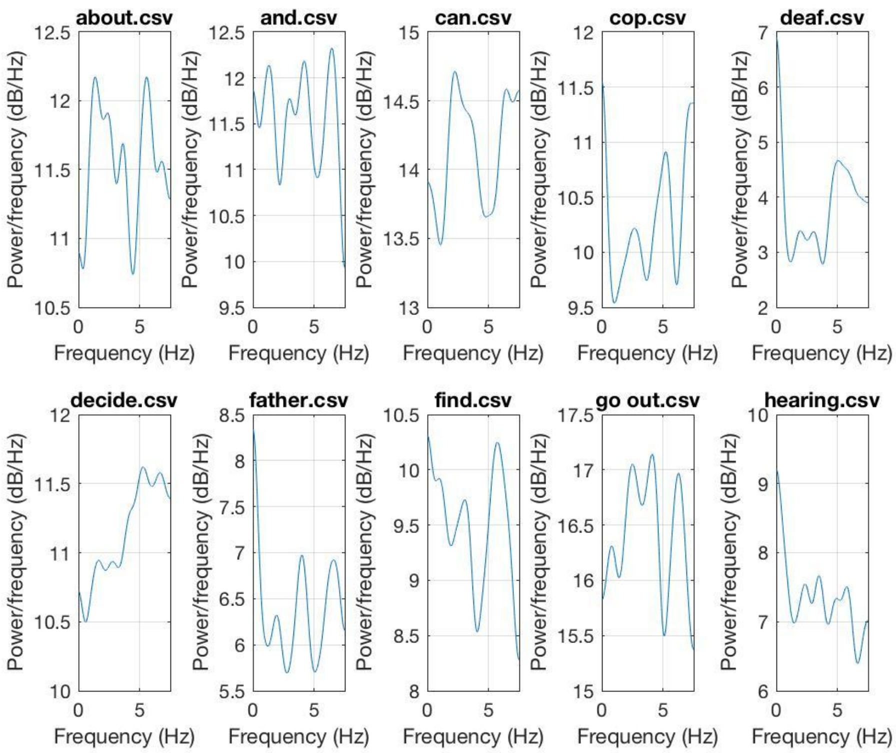
EMG1L


about.csv
and.csv
can.csv
cop.csv
deaf.csv
decide.csv
father.csv
find.csv
go out.csv
hearing.csv
Power/frequency (dB/Hz)
Frequency (Hz)

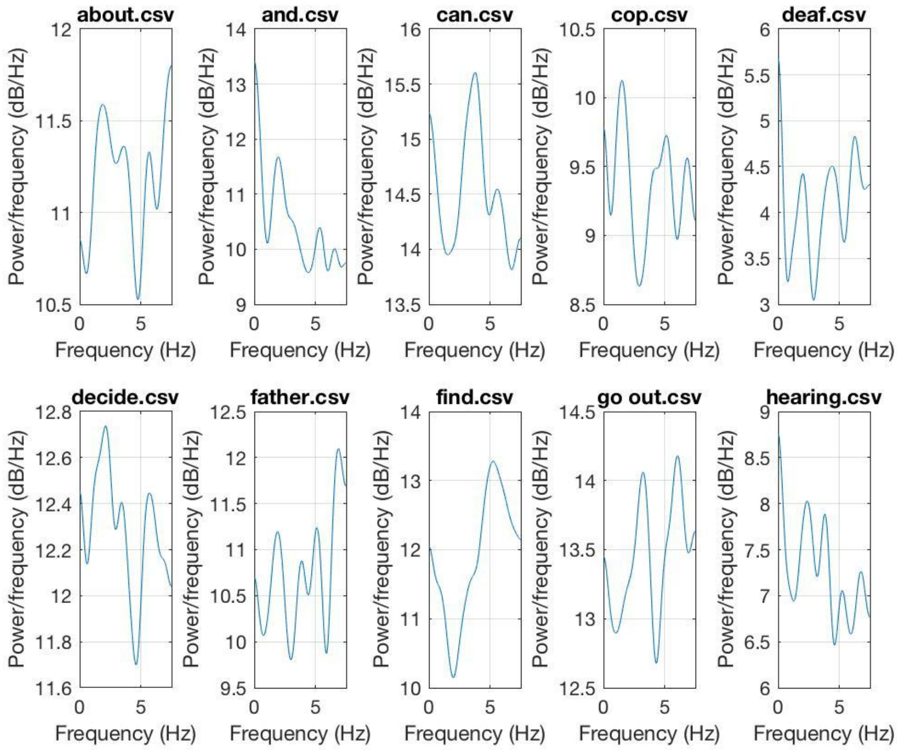
EMG3R


about.csv
and.csv
can.csv
cop.csv
deaf.csv
decide.csv
father.csv
find.csv
go out.csv
hearing.csv
Power/frequency (dB/Hz)
Frequency (Hz)

EMG4L


about.csv
and.csv
can.csv
cop.csv
deaf.csv
decide.csv
father.csv
find.csv
go out.csv
hearing.csv
Power/frequency (dB/Hz)
Frequency (Hz)

EMG5L



4.2.3 Discrete Wavelet Transform (DWT)

a) The wavelet transform can provide us with the frequency of the signals and the time associated to those frequencies. The data recorded by various sensors can be decomposed as wavelets which best represent the frequency and time information. DWT passes the signals through high pass and low pass filters, followed by down sampling to obtain the approximation and detail coefficients. We have used dwt function in MATLAB to obtain the approximation and detail coefficients. The detail coefficients which show less variations are ignored as they don't help much in distinguishing between the gestures. Graphs are plotted with the approximation coefficients. To find the sensors which can differentiate the gestures well, we found correlation between the approximation coefficients of every sensor. We chose 5 of these unique signals to identify all gestures. For each chosen sensor, we selected 3 peak values. So, we will have 5x3 = 15 feature values from DWT method for each action.

b) 'DWT_Method.m' file has relevant code for feature extraction using DWT. The DWT transformation of the selected features can be seen in the plots below. We can observe that it is different for each gesture.

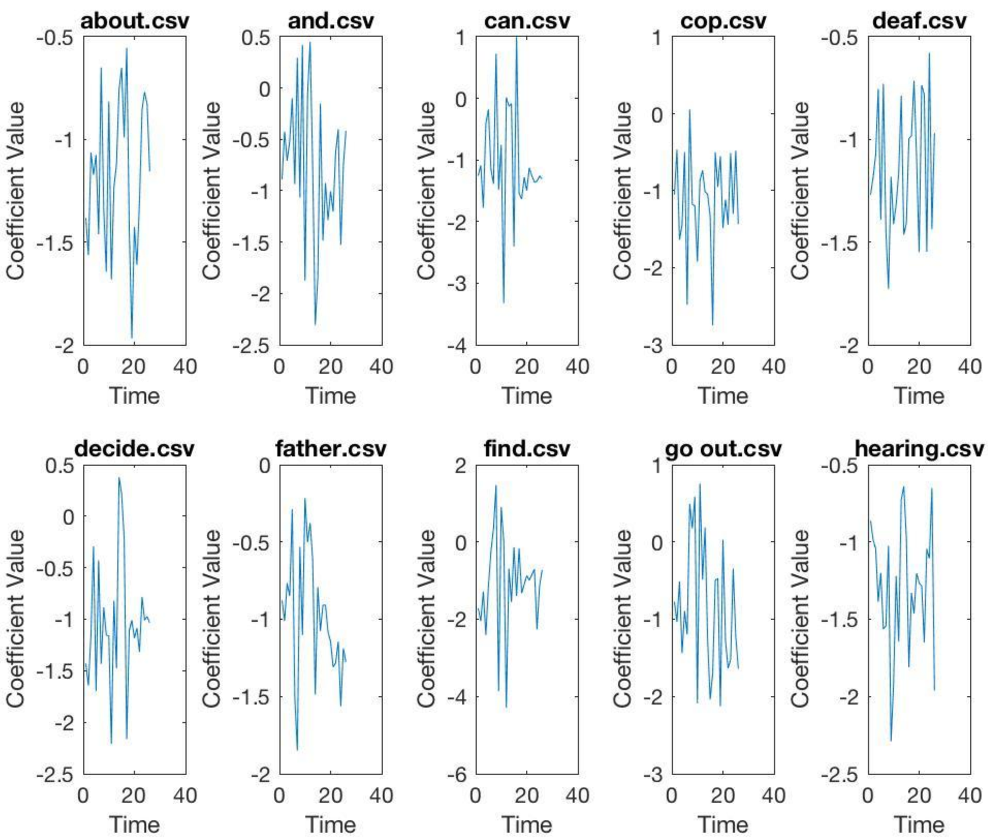
Analysis of Output

Using the script DWT_Method.m, in which we used the concept of correlation coefficient to get the most distinguishable sensors i.e. the script will show us which among the 34 sensors; helps us to identify the gesture (i.e. DWT of those sensors will be most distinguishable). We observed that EMG5L, EMG0R, EMG4R, OYR, EMG4R are the most distinguishable features identified using DWT feature extraction method. The following plot verifies our claim. Please refer the output folder to see the graphs of all sensors after transformation

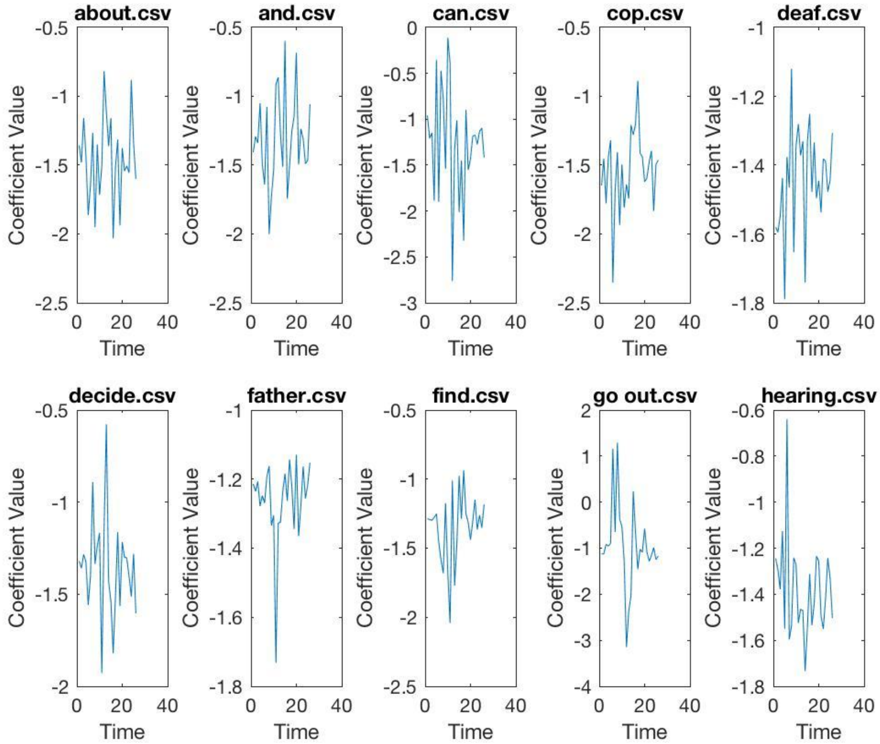
Conclusion

Since EMG5L, EMG0R, EMG4R, OYR, EMG4R helps us to distinguish the actions, we will use them as the features for every action.

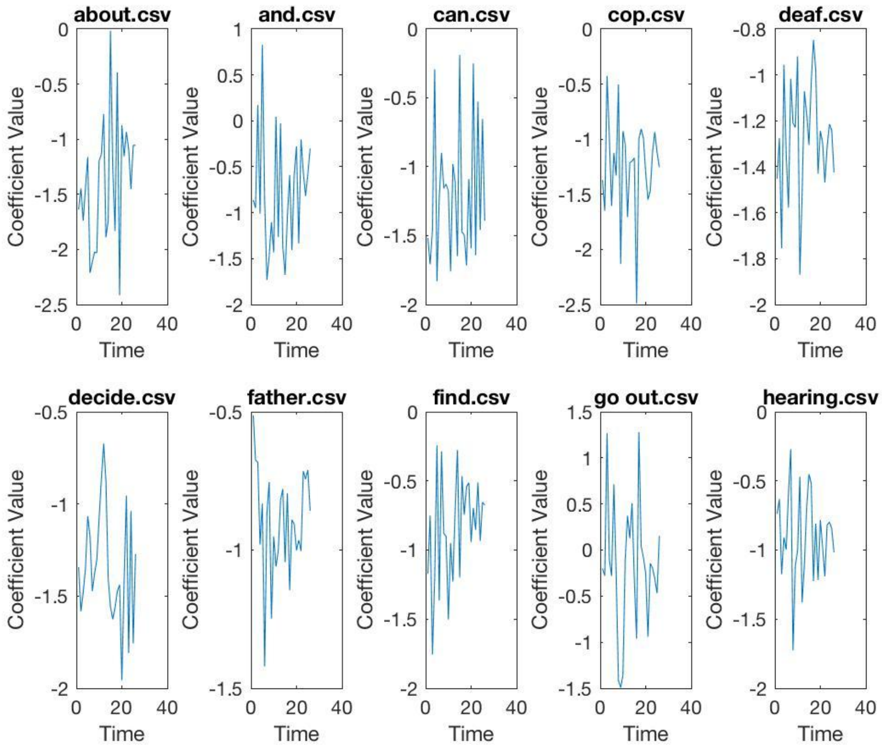
EMG0R



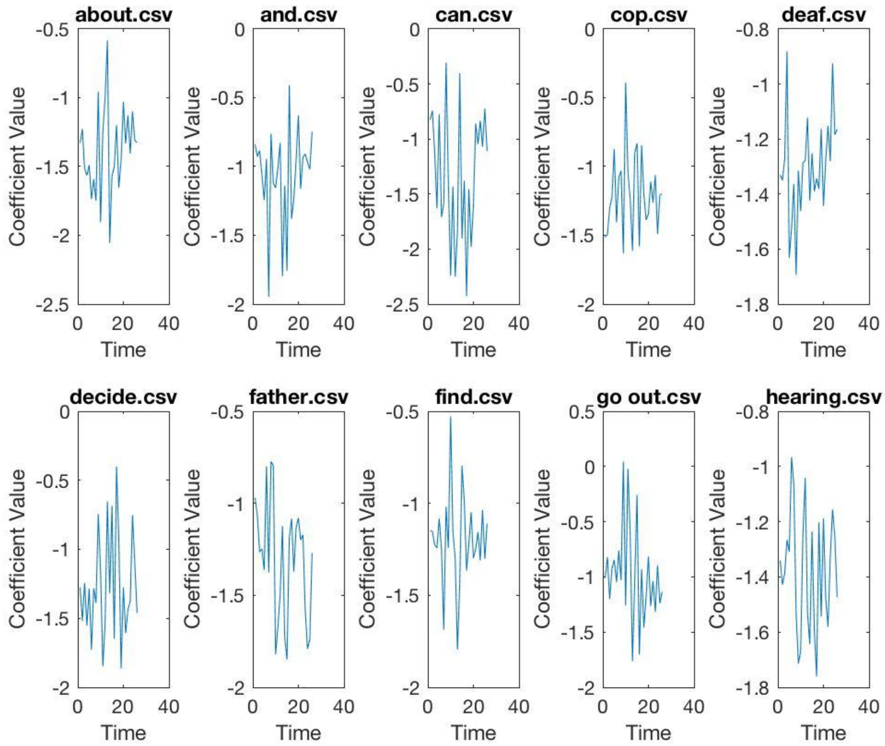
EMG4L


about.csv
and.csv
can.csv
cop.csv
deaf.csv
decide.csv
father.csv
find.csv
go out.csv
hearing.csv
Coefficient Value
Time

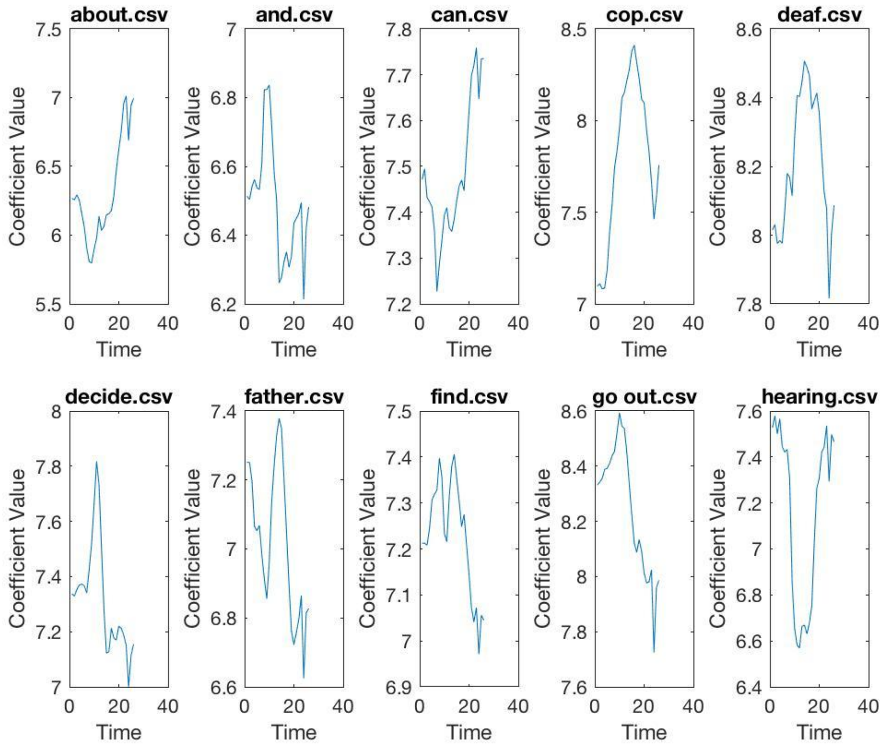
EMG4R


about.csv
and.csv
can.csv
cop.csv
deaf.csv
decide.csv
father.csv
find.csv
go out.csv
hearing.csv
Coefficient Value
Time

EMG5L


about.csv
and.csv
can.csv
cop.csv
deaf.csv
decide.csv
father.csv
find.csv
go out.csv
hearing.csv
Coefficient Value
Time

OYR


about.csv
and.csv
can.csv
cop.csv
deaf.csv
decide.csv
father.csv
find.csv
go out.csv
hearing.csv
Coefficient Value
Time

4.2.4 Root Mean Square (RMS)

a) The data matrix obtained from Task 1 is transformed into a column vector by applying the root mean square function (RMS) on it in MATLAB. Applying RMS would compress the deviating values of each sensor across the timestamps into a single value. The resultant column vector would have (number of actions) x 34 rows and 1 column i.e. one value for each sensor (rather than 45 time series values) for each action of that gesture.

b) The intuition behind using RMS is that it can be used as a measure of magnitude of a set of data. It gives a sense for the typical size of numbers. For a given set of sensor data, RMS basically derives a scalar value that lies in the positive quadrant and thus establishes similarity between all the values generated.

c) The column vector is then transposed and transformed in such a way that the columns indicate the different sensors of the Myo Sensor and the rows contain the different actions of the gesture. We would essentially have a (total number of actions) x 34 matrix at the end of this transformation.

d) The resultant matrix is then transformed into a row vector by applying MATLAB's *Trim-Mean* function – which finds the mean across columns while also eliminating outliers by assigning 0.5 weights to them. The intuition behind using mean to combine the Root Mean Square values of different actions is that mean serves as a statistical method to give equal weights to all the actions of a gesture and combine them into one which will most likely be a good representative of the gesture. We will end up with a 1 x 34 matrix for every gesture. We combine all these row vectors (10 in number) into one to create a single 10 x 34 matrix. Now we use MATLAB's Variance function to compute variances across rows for every sensor. Variance was used because it reflects how discriminating a sensor is. The resultant 1 x 34 row vector is sorted in descending order and the 4 sensors with the highest discriminating power are chosen.

e) The MATLAB code to perform all the functions described above can be found in the file *RMS_Method.m.*

Analysis of Output

Using the script *RMS_Method.m*, we found the 5 attributes that have the highest variance and hence would be the most discriminating.

Conclusion

Since these 5 attributes will help us to distinguish the gestures, we can use them as features for every action.

4.2.5 Standard Deviation (STD)

a) The data matrix obtained from Task 1 is transformed into a column vector by applying the standard deviation function (STD) on it in MATLAB. Applying STD would compress the deviating values of each sensor across the timestamps into a single value. The resultant column vector would have (number of actions) x 34 rows and 1 column i.e. one value for each sensor (rather than 45 time series values) for each action of that gesture.

b) The intuition behind using Standard Deviation is that Standard Deviation is a measure of the spread of the data around the mean value. A high value for Standard Deviation value indicates that that particular sensor has recorded a drastic change in hand movement and deems to be a good candidate to be highly discriminating. A low value for Standard Deviation indicates that the values of that particular sensor are more tightly packed around its mean and that the movement of the hand has been minimal with respect to what the sensor was recording.

c) The column vector is then transposed and transformed in such a way that the columns indicate the different sensors of the Myo Sensor and the rows contain the different actions of the gesture. We would essentially have a (total number of actions) x 34 matrix at the end of this transformation.

d) The resultant matrix is then transformed into a row vector by applying MATLAB's Trim-Mean function – which finds the mean across columns while also eliminating outliers by assigning 0.5 weights to them. The intuition behind using mean to combine the Standard Deviation values of different actions is that mean serves as a statistical method to give equal weights to all the actions of a gesture and combine them into one which will most likely be a good representative of the gesture. We will end up with a 1 x 34 matrix for every gesture. We combine all these row vectors (10 in number) into one to create a single 10 x 34 matrix. Now we use MATLAB's Variance function to compute variances across rows for every sensor. Variance was used because it reflects how discriminating a sensor is. The resultant 1 x 34 row vector is sorted in descending order and the 4 sensors with the highest discriminating power are chosen.

e) The MATLAB code to perform all the functions described above can be found in the file *STD_Method.m*.

Analysis of Output

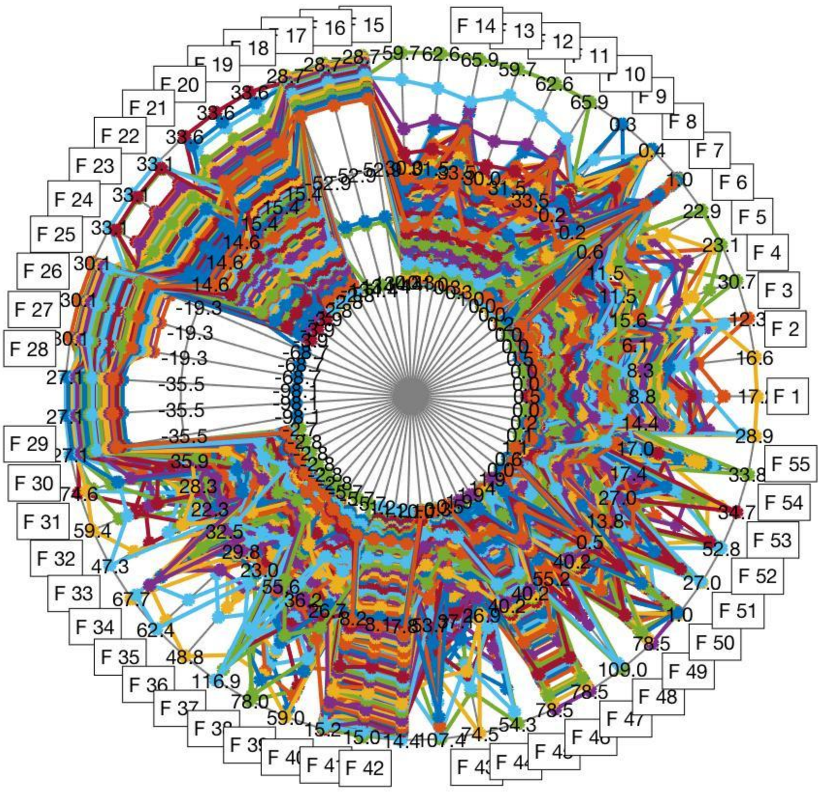
Using the script *STD_Method.m*, we found 5 attributes that have the highest variance and hence would be the most discriminating.

Conclusion

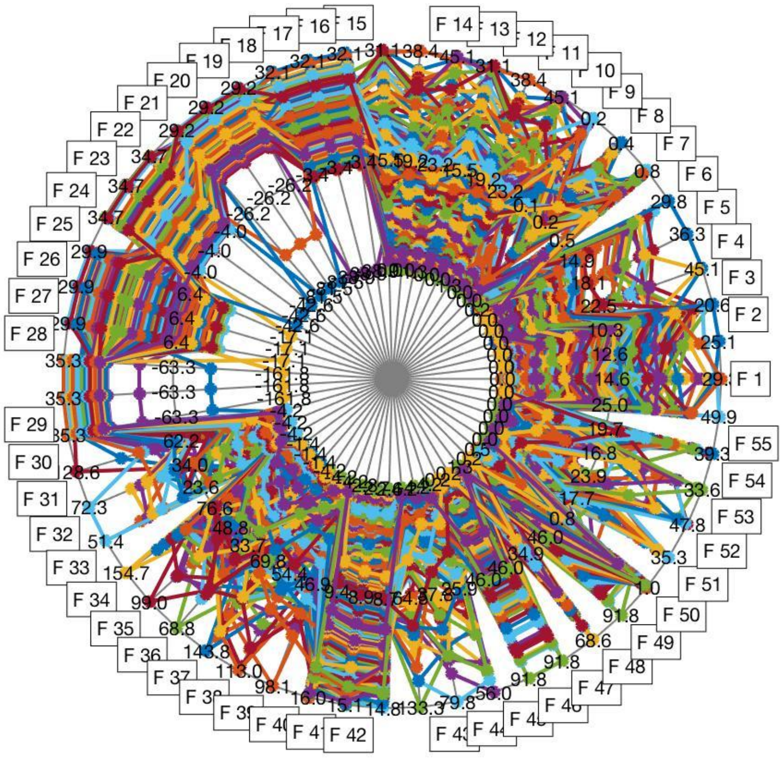
Since these 5 attributes will help us to distinguish the gestures, we can use them as features for every action.

Answer 2d) Since our N*55 (where N is the total number of actions performed for all gestures) matrix is a multi-dimensional matrix, we used Spider plots to analyze actions in feature domain. The output files for all the gesture are pasted below as spider plots.

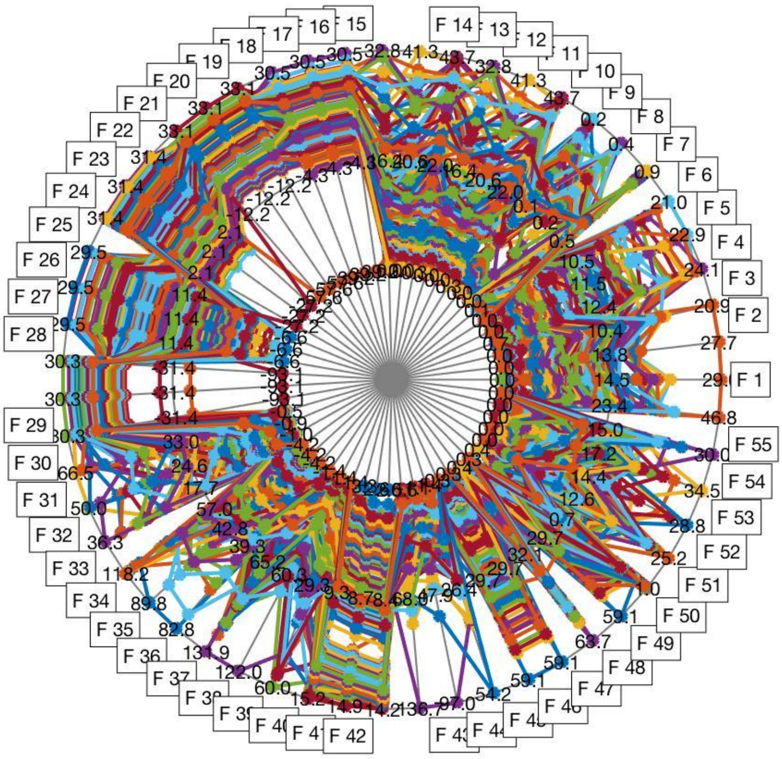
Spider Plot for 'about' gesture



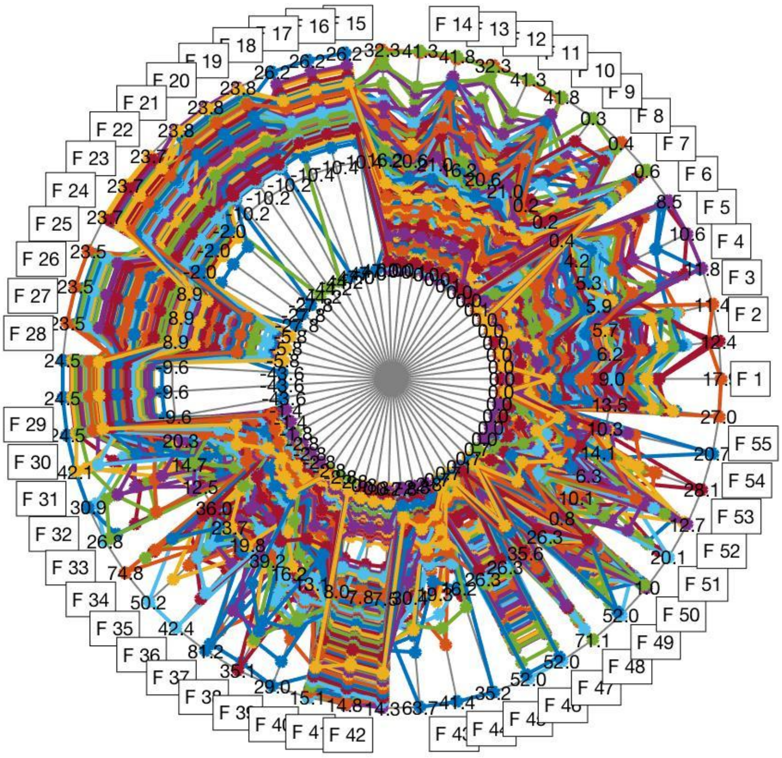
Spider Plot for 'and' gesture



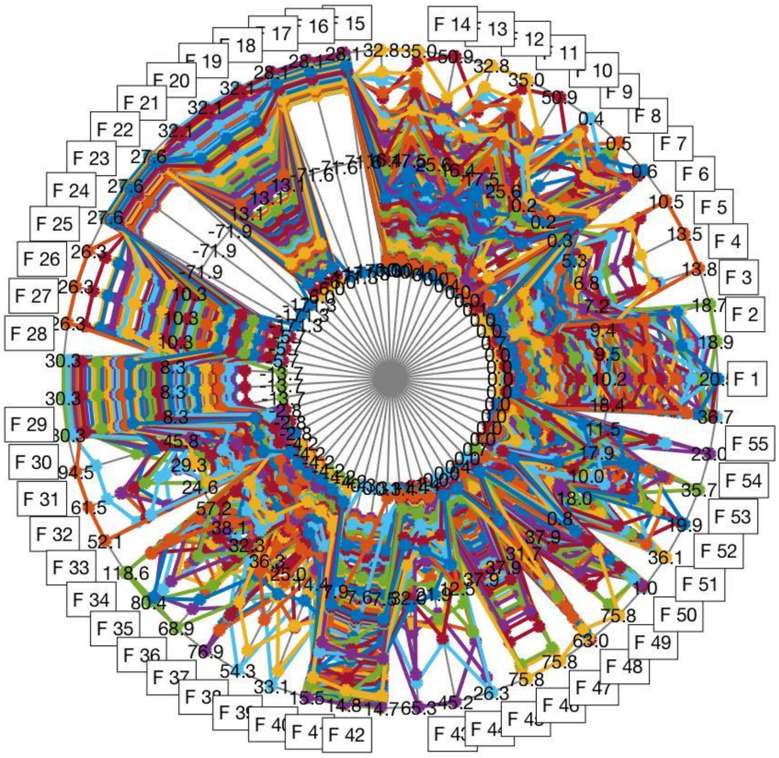
Spider Plot for 'cop' gesture



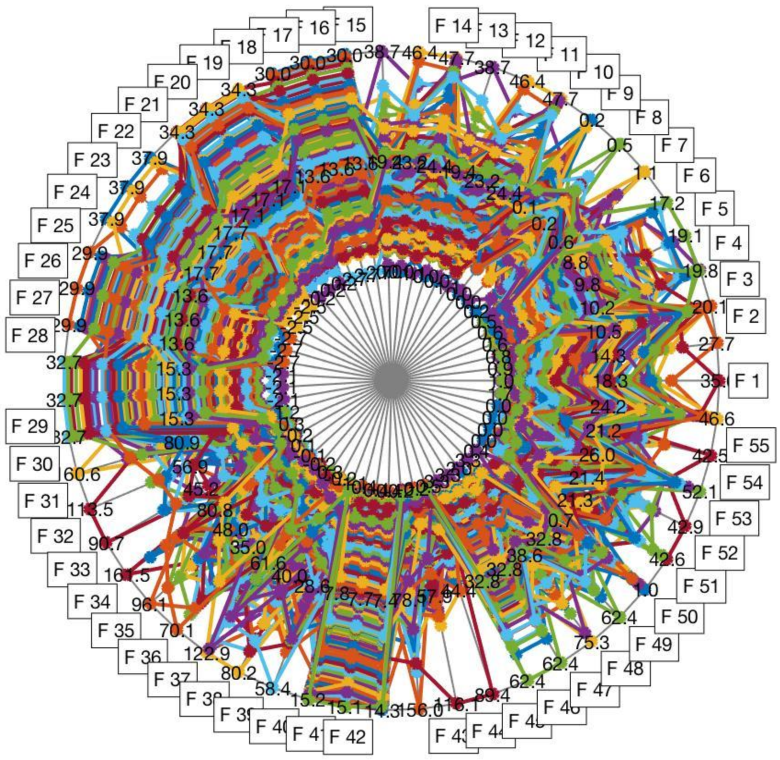
Spider Plot for 'deaf' gesture



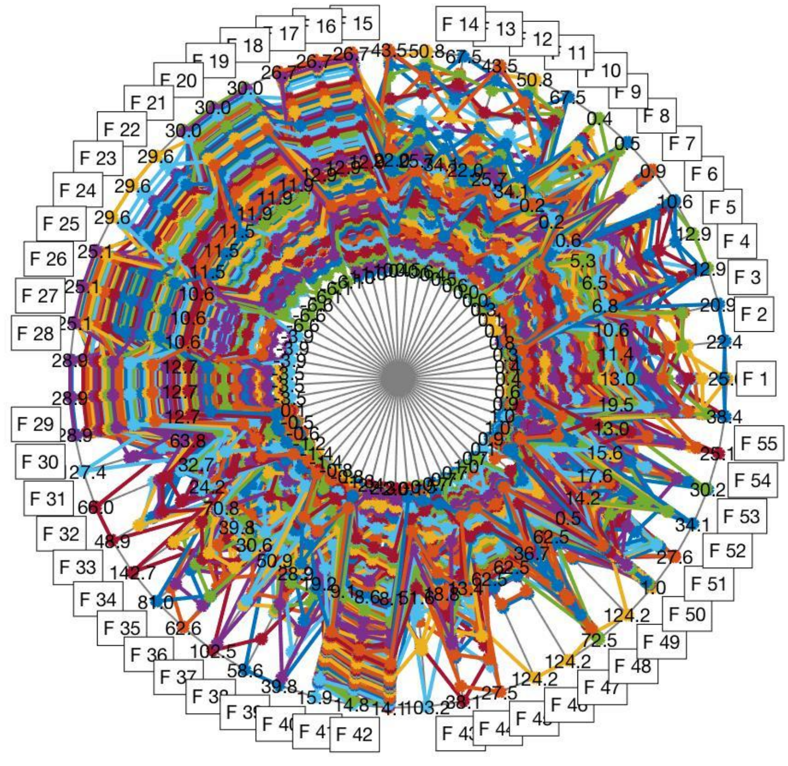
Spider Plot for 'father' gesture



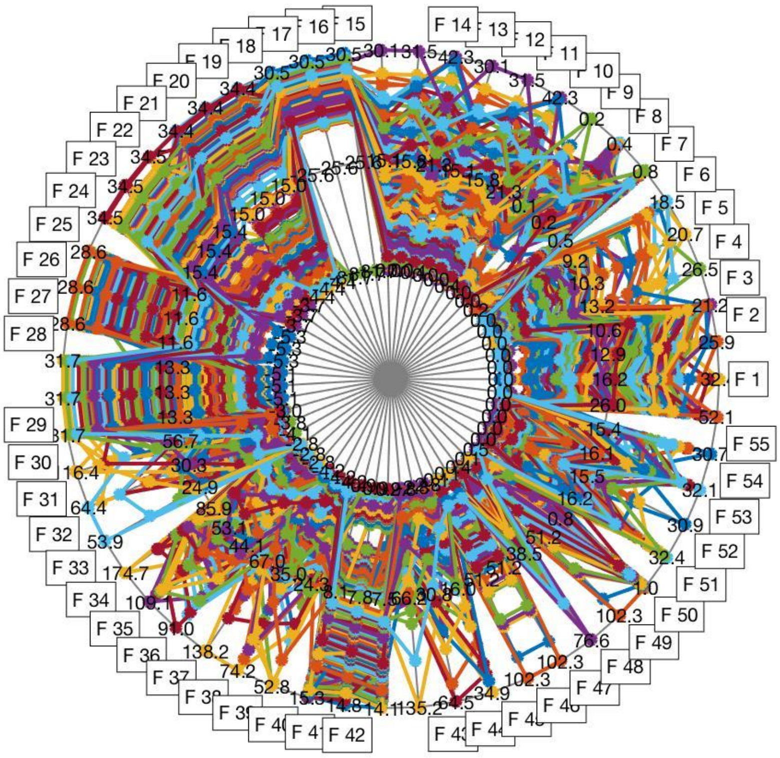
Spider Plot for 'can' gesture



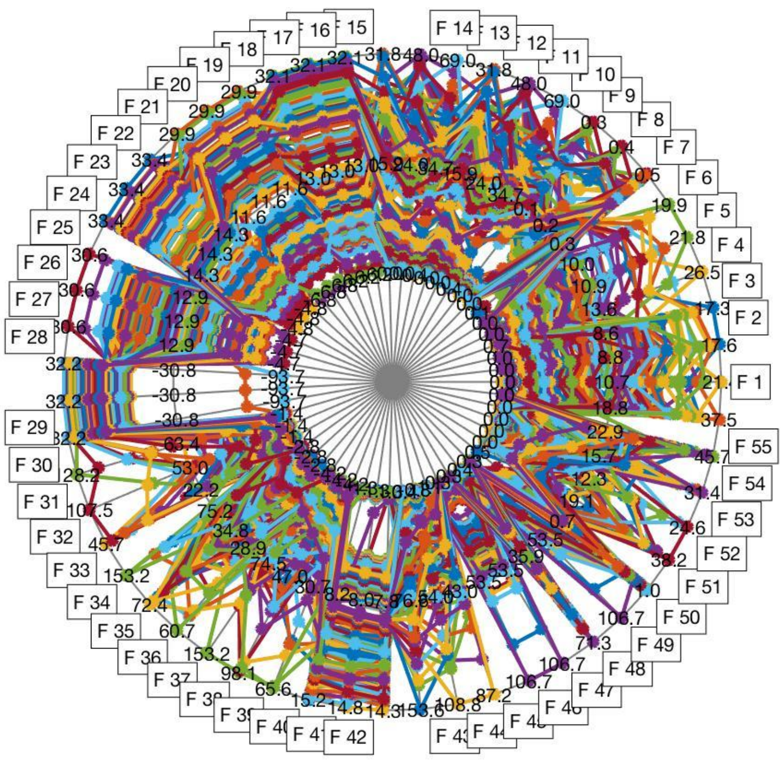
Spider Plot for 'decide' gesture



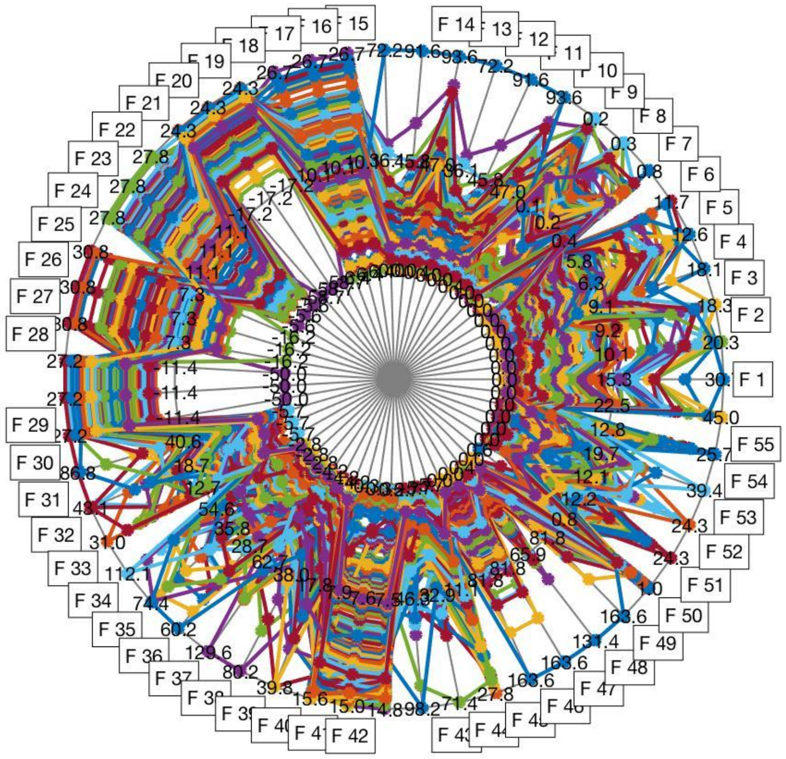
Spider Plot for 'find' gesture



Spider Plot for 'go out' gesture



Spider Plot for 'hearing' gesture



Answer 2e) From these Spider plot we can observe that, most of the features for a gesture are consistent (i.e. good features), and some are not (not so good features)

4.3 Task 3 : Feature Selection

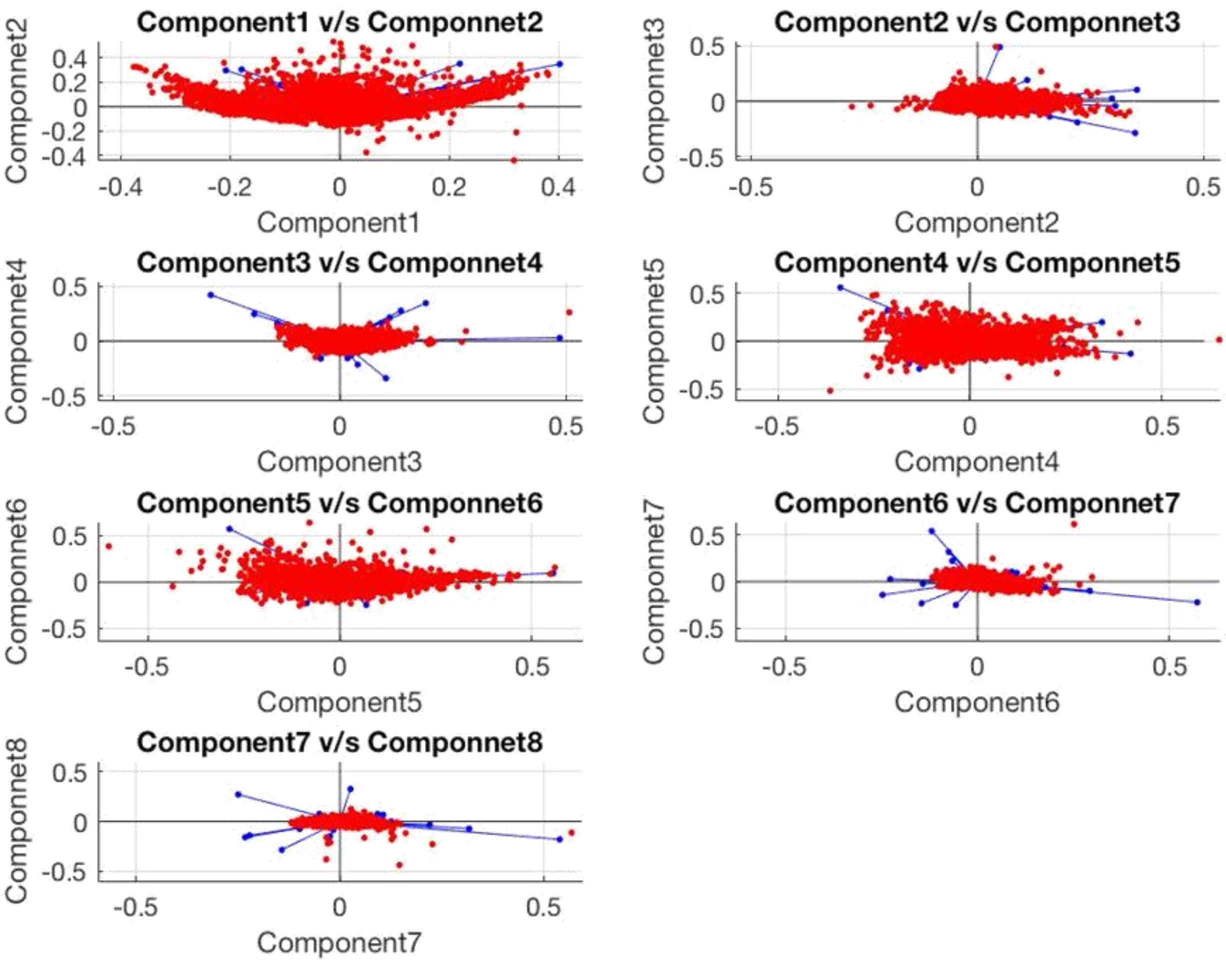
Feature selection is a very important task in data mining. It's a process of selecting the subset of relevant features which can be used for classification. In this project, we use Principle Component Analysis as the feature selection method. PCA will return the top features in the order of their discriminating power.

4.3.1 Arranging Feature Matrix

The feature matrices from task 2 are the input for PCA. The Feature matrix has instances along rows and feature values along columns. If we have N instances/actions, feature matrix of that gesture will be of dimension N*55.
The input to PCA should have all possible gestures along with their features. We combine all 10 feature matrix from task 2 to form the input matrix for PCA.

4.3.2 Execution of PCA

MATLAB's PCA( ) function is used to perform PCA on the feature matrix. From the latent output of PCA( ), we can observe that principle component has Eigen value (variance) in decreasing order. We also observed that we need only 5 eigenvectors to keep 80% and 8 eigenvectors to keep 90% of the variance. This is a drastic reduce in the dimensionality in feature space. Below are the biplots of first 5 eigenvectors which helps to keep 80% of the variance in data.



4.3.3 Discussion of Eigen Vectors

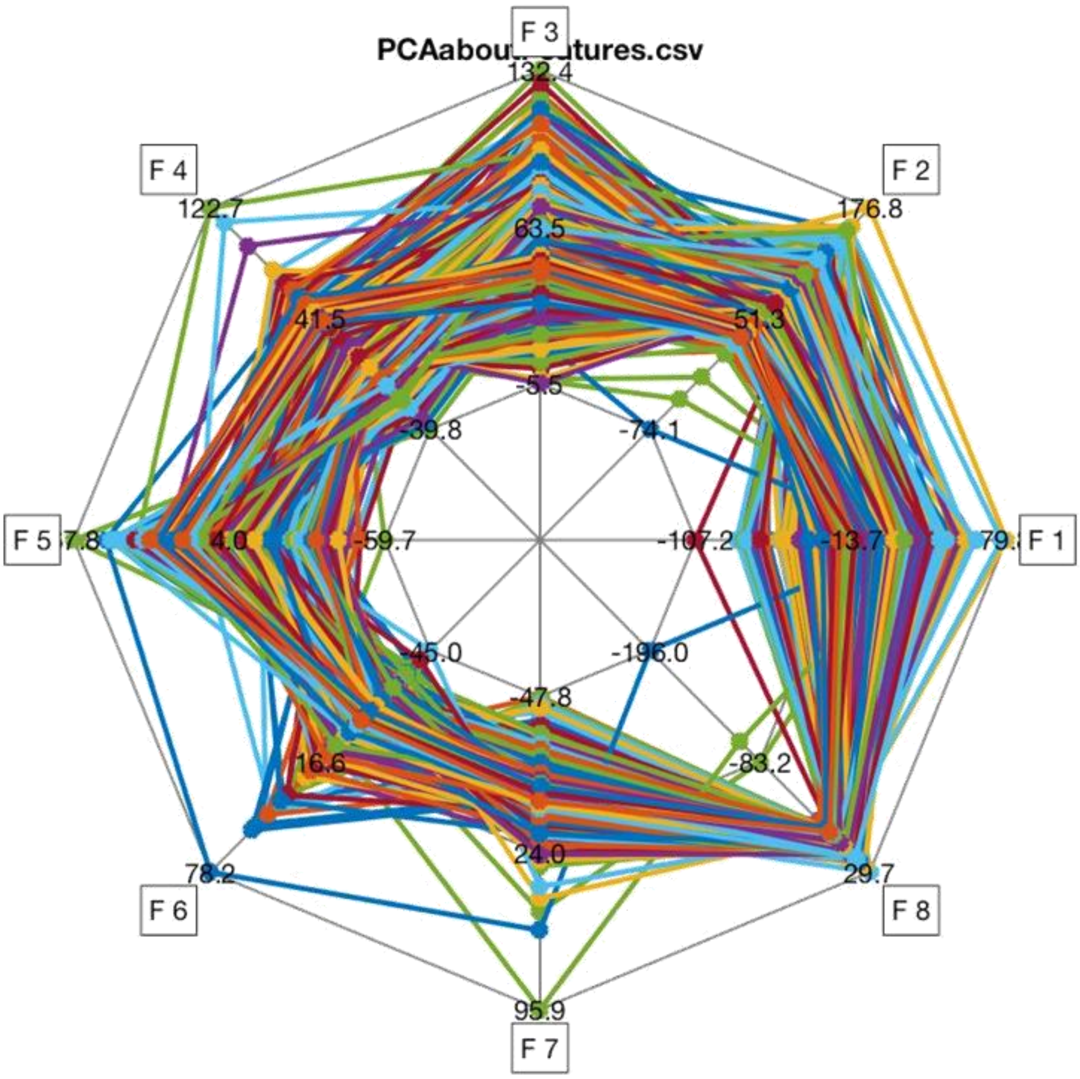
We can observe from the latent output of PCA( ) that the first component will reserve the highest variance 38.58% and the second PC will reserve around 20.11% of the variance. We can observe that in first PC, coefficient of feature 34, 35, 37, 47 46 (DWT_EMG0R_P1, DWT_EMG0R_P2, DWT_EMG4R_P1, RMS_GRY, RMS_GRX) has highest magnitude. This means that these features help us to preserve much of variance. This implies that features extracted from observing EMG and angular speed will help us to identify the gesture. Since our approach was to pick sensors which are more distinct from each other, we didn't know which among features are helpful in identifying gesture. But PC coefficients helps us to identify EMG and Gyrometer reading as the most helpful for the gesture recognition, which makes sense.

4.3.4 Results of PCA

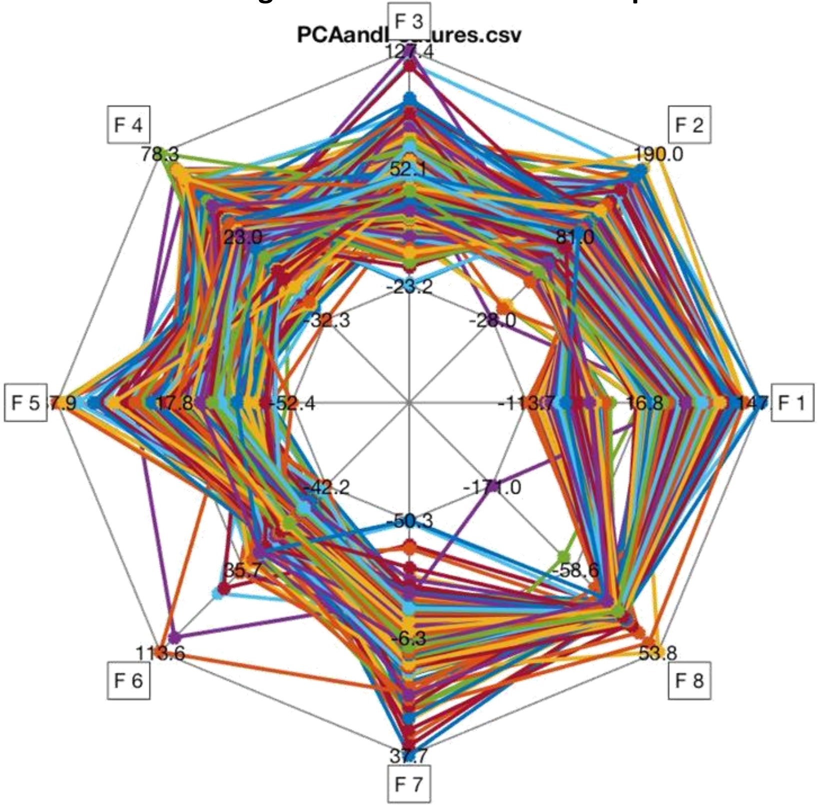
We can create new feature matrix my multiplying data matrix with eigenvectors. (Equivalent to *scores* output of MATLAB's *PCA()*). *new_feature_matrix* in *PCA.m* will have this new feature matrix. This matrix will have feature data for all the gestures. We divide this matrix to create 10 new feature matrix corresponds to each gesture.

Since these matrices are still multidimensional data, we can use spider plots to visualize them.

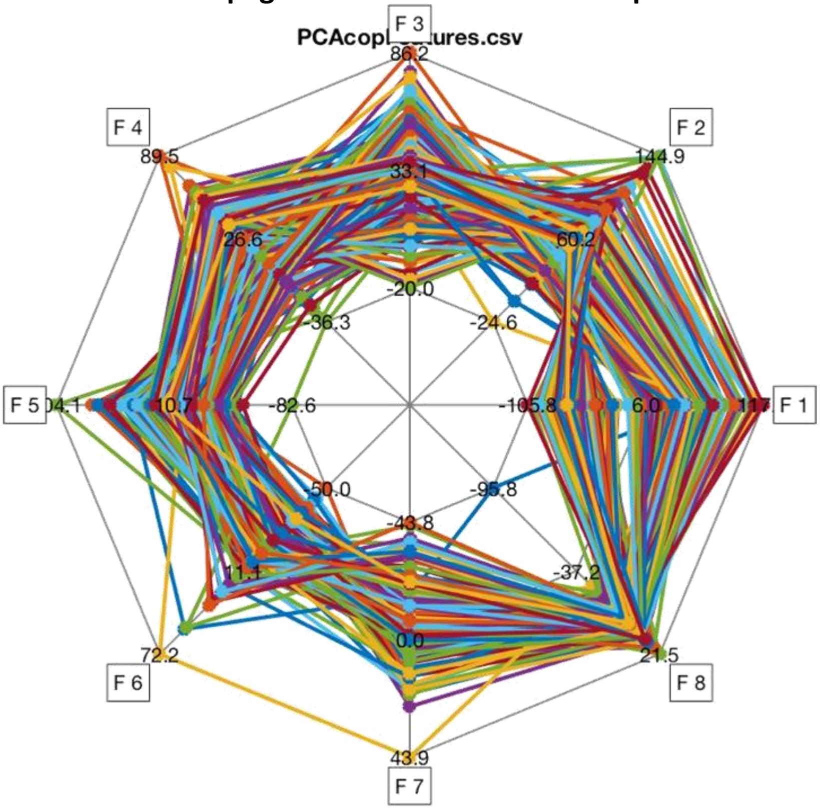
'About' gesture in reduced feature space



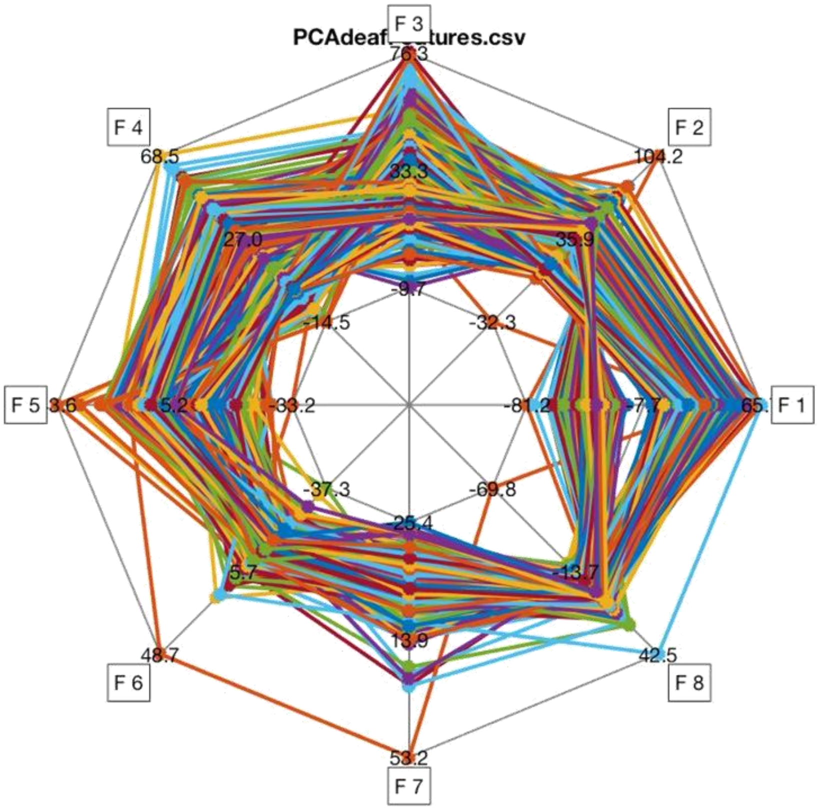
‘And’ gesture in reduced feature space



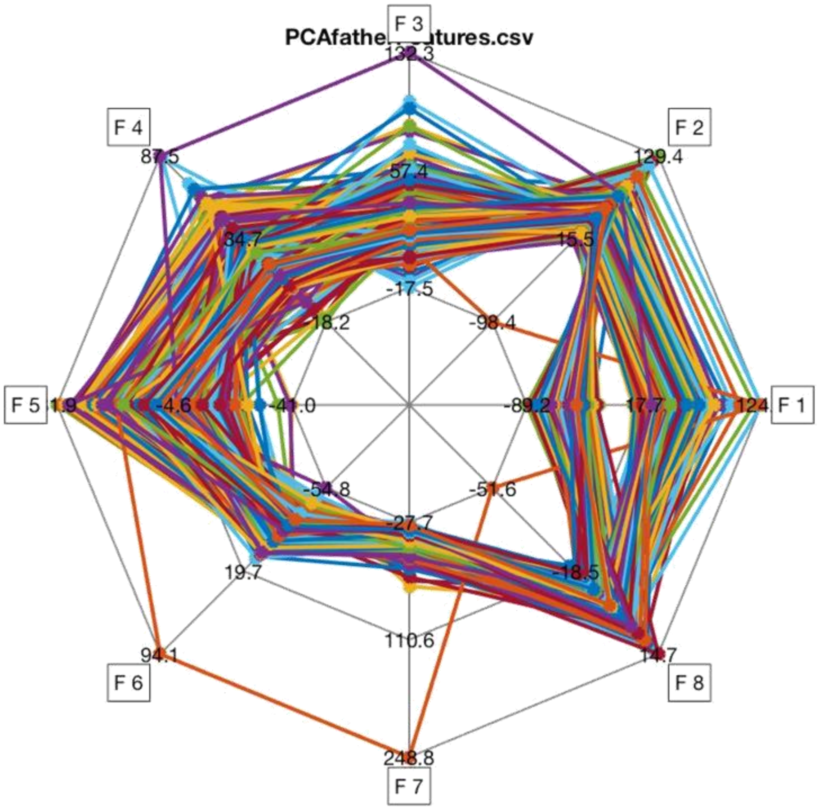
‘Cop’ gesture in reduced feature space



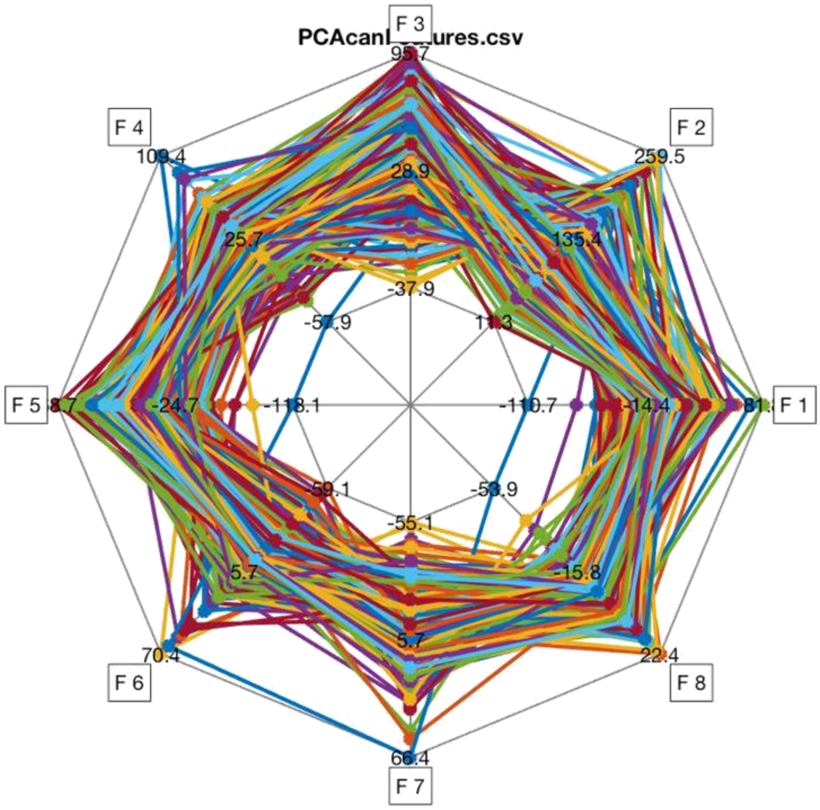
'Deaf' gesture in reduced feature space



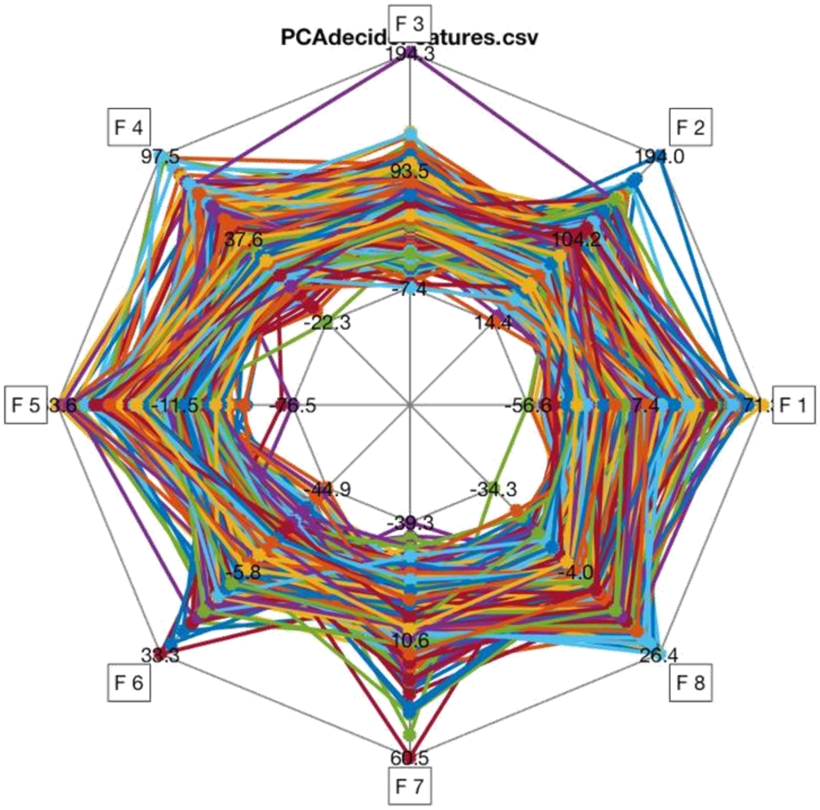
'Father' gesture in reduced feature space



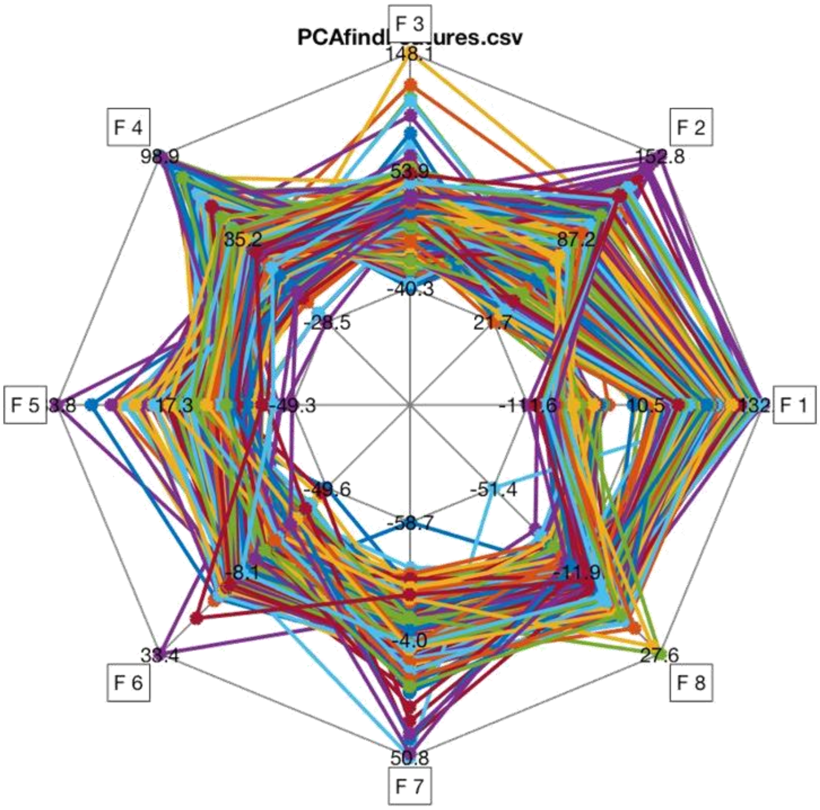
‘Can’ gesture in reduced feature space



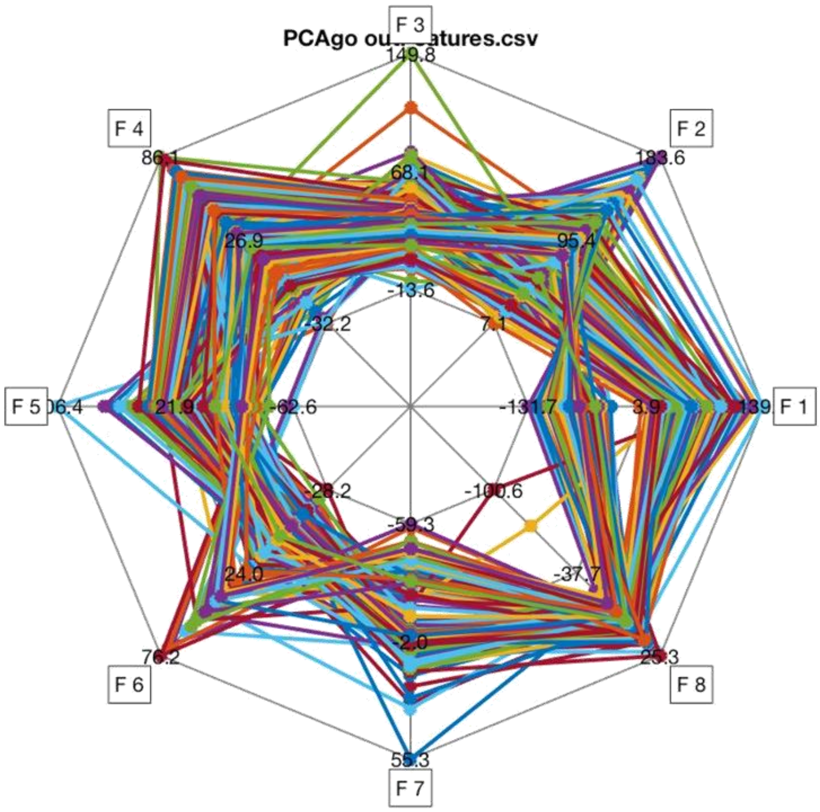
‘Decide’ gesture in reduced feature space



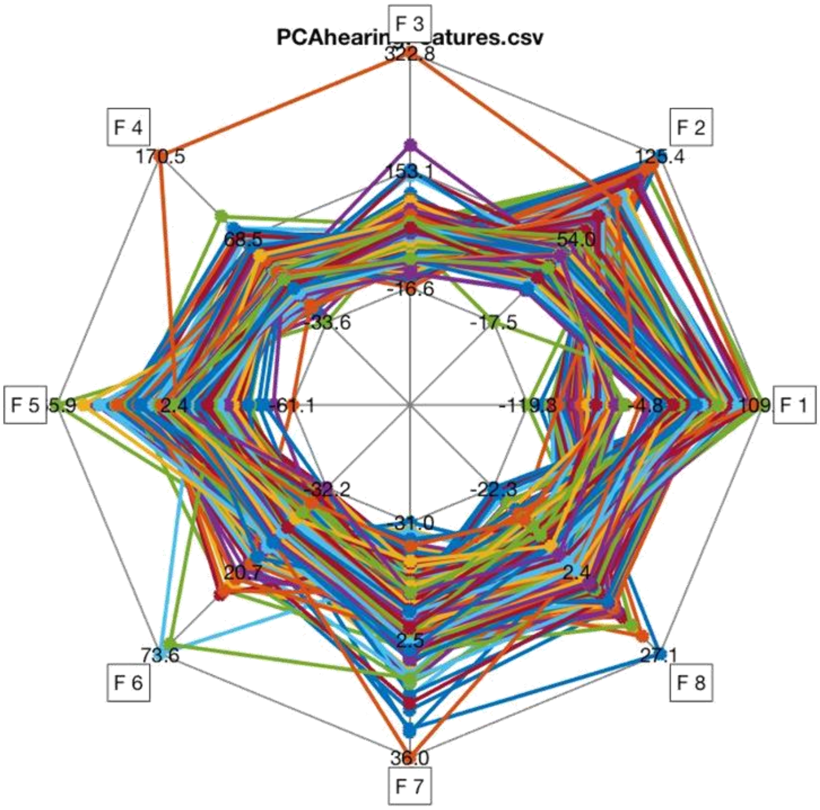
'Find' gesture in reduced feature space



'Go Out' gesture in reduced feature space



'Hearing' gesture in reduced feature space



5. Conclusion : Usefulness of PCA

Using PCA, we were able to reduce feature space from 55 dimensions to 5 dimensions keeping 80% of the variance. This is impressive because we need only 5 dimensions to represent each gesture. We can reserve 90% of data using only 8 features. This helps us to get rid of features which have less variance, which in turn are considered as bad features. It's also important to notice that a principle component tells which all important features are which helps us to keep maximum variance. It can be considered as good features.

For gesture recognition, we can see a clear distinct pattern in Spider plot for every gesture. We observed from task 2 that, among the chosen 55 features, '*good features*' had patterns whereas '*bad features*' did not. However, in the new PC's selected feature dimension, every feature has a good pattern.

6. Resources

- Myo Sensor : https://www.myo.com/
- Matlab Standard Deviation : https://www.mathworks.com/help/matlab/ref/std.html
- Matlab Root Mean Square: https://www.mathworks.com/help/signal/ref/rms.html
- Matlab Mean Excluding Outliers: https://www.mathworks.com/help/stats/trimmean.html

- Matlab Variance : https://www.mathworks.com/help/matlab/ref/var.html
- Matlab Colon : https://www.mathworks.com/help/matlab/ref/colon.html
- Matlab Wavelet Transform : https://www.mathworks.com/help/wavelet/ref/dwt.html
- Matlab Power Spectral Density : https://www.mathworks.com/help/signal/ref/dspdata.psd.html
- Matlab Principal Component Analysis : https://www.mathworks.com/help/stats/pca.html
- Matlab Transpose : https://www.mathworks.com/help/matlab/ref/transpose.html
- Eigen Vector Calculator : https://www.symbolab.com/solver/matrix-eigenvectors-calculator
- How PCA Works: https://georgemdallas.wordpress.com/2013/10/30/principal-component-analysis-4-dummies-eigenvectors-eigenvalues-and-dimension-reduction/
- Making Sense of PCA:
- https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues
- PCA: https://coolstatsblog.com/2015/03/21/principal-component-analysis-explained/
- Visual Explanation of PCA: http://setosa.io/ev/principal-component-analysis/
- Use of Eigen Vectors: https://math.stackexchange.com/questions/23312/what-is-the-importance-of-eigenvalues-eigenvectors/
- Practical Uses of Eigen values: https://www.quora.com/What-are-some-very-good-and-practical-uses-of-eigenvalues-of-a-matrix
- Visual Explanation of Eigen Vectors: http://setosa.io/ev/eigenvectors-and-eigenvalues/
- Application of Eigen Vectors: http://www.personal.soton.ac.uk/jav/soton/HELM/workbooks/workbook_22/22_2_apps_eigen_vals_vecs.pdf
- RMS: http://www.analytictech.com/mb313/rootmean.htm
- https://stats.stackexchange.com/questions/97203/question-about-pca-how-are-pc-scores-generated-using-pc-eigenvectors
- http://matlabdatamining.blogspot.it/2010/02/principal-components-analysis.html
- https://stackoverflow.com/questions/19052317/configuring-biplot-in-matlab-to-distinguish-in-scatter
- https://stats.stackexchange.com/questions/69157/why-do-we-need-to-normalize-data-before-principal-component-analysis-pca
- https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues