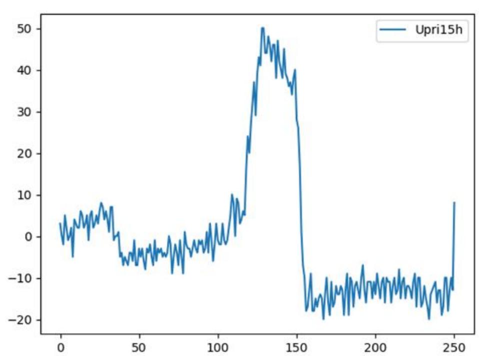
Ain Shams University
Faculty of Computer & Information Sciences
Scientific Computing Department

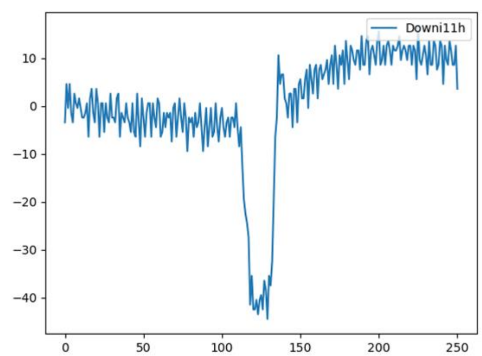
Electro-oculogram (EOG) UP and Down

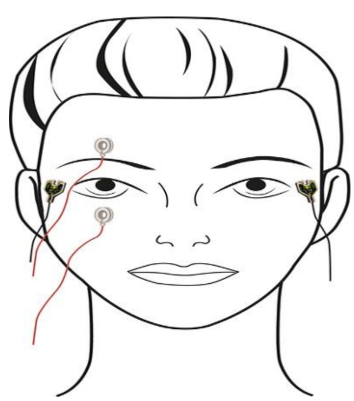
- Team members :
  - Mohamed Mostafa Soliman Mostafa
  - Mohamed Hussein El-Sayed Kamal
- Under Supervision of:
  - Dr. Manal Mohsen Tantawi
  - TA. Aya Nasser
  - TA. Hader Elsaadawy

➢ Problem Definition :

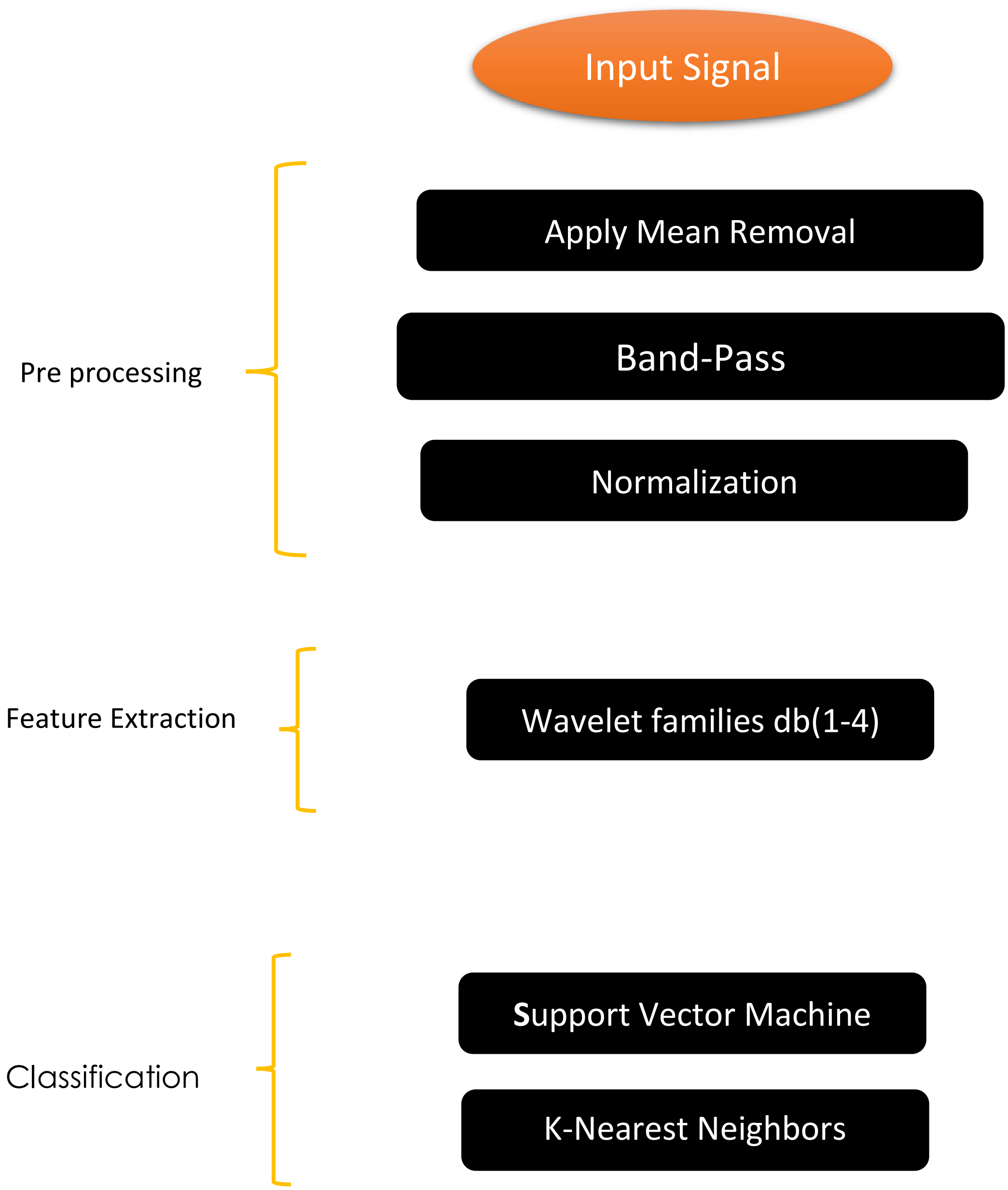
Amyotrophic lateral sclerosis (ALS) patients communicate with the world using eye movement, since their voluntary muscles are paralyzed. Many efforts are exerted to support this way of communication by tracking or detecting eye movement. Electro-oculogram (EOG) is an electro-physiological signal generated by eye movement and can be measured using electrodes placed around the eyes. The electrodes will be placed on the forehead. The two channels are arranged vertically and horizontally. Each eye movement displays a positive or negative peak in the corresponding EOG signal. Human-machine interface (HMI) can be developed based on eye movement recognition. These interfaces can help paralyzed people to interact with smart phones or laptops and also can help them to move wheelchair or play games without any help from others. In this project, we are interested to distinguish between up and down movements using vertical EOG signals.







- System Architecture :



Experiments

SVM

Exp No	SVM Kernel	C	Gamma	Bandpass Levels	Wavelet Names	Wavelet Levels	Train Accuracy	Test Accuracy	Comments
1	Linear	1-5000	0.1	1	Db1	1	0.966 - 1	0.6	
2	Linear	100	0.01	1	Db1	1	1	0.6	
3	Linear	100	0.001	1	Db1	1	1	0.6	
4	Linear	100	0.0001	1	Db1	1	1	0.6	
5	Linear	100	0.0001	2	Db1	1	1	0.6	
6	Linear	100	0.0001	3	Db1	1	1	0.7	More Accurate Coefficient for filter the signal due to the higher level
7	Linear	100	0.0001	6	Db1	1	1	0.9	More Accurate Coefficient for filter the signal due to the higher level
8	Linear	100	0.0001	7	Db1	1	0.73	0.6	As we increase the order of the level the model underfit t the train data due to the complexity of the function of the bandpass ,the signal may be damaged according to bad coefficient
9	Linear	100	0.0001	8	Db1	1	0.56	0.5	As we increase the order of the level the model underfit t the train data due to the complexity of the function of the bandpass ,the signal may be damaged according to bad coefficient
10	Linear	100	0.0001	9	Db1	1	0.50	0.70	As we increase the order of the level the model underfit t the train data due to the complexity of the function of the bandpass

11	Linear	100	0.0001	6	Db2-Db4	1	1	0.9	May be the wavelet db2 to db4 extract approximately the same
12	Linear	100	0.0001	6	Db1-Db4	2-15	1	0.9	May be the wavelet db1 to db4 extract approximately the same ,the increasing in the level not change the result the reason is more clear from the plotting
13	Rbf	100	0.0001	6	Db1-Db4	2	0.83	0.8	Rbf transform data into bad space such that the model can not fit the data well
14	Polynomial Degree = 4	100	0.0001	6	Db1-Db4	2	0.63	0.60	Polynomial transform data into bad space such that the model can not fit the data well

KNN

Exp No	Neighborhood	Wavelet	Wavelet level	Band level	Train Acc	Test Acc	Comments
1	1	Db1	1	1	1.0	0.7	
2	3	Db1	1	1	0.8334	0.6	The vote of the top 3 not represent the correct class according to the similarity between the two classes in the features
3	5	Db1	1	1	0.7667	0.6	As the number of neighborhood be 5 the train accuracy decrease which mean the model can not fit the data will according to its not discriminative features

4	7	Db1	1	1	0.8	0.6	The train accuracy increased but we loss the generalization
5	1	Db2	1	1	1.0	0.7	Db2 extract feature may be similar to the features extracted in exp 1
6	1	Db3-Db4	1	1	1.0	0.7	The features extracted by all Dbi may be approx similar (it is clear from the plotting of the features)
7	1	Db1	2	1	1.0	0.7	The addition wavelet level not extract useful information that can make the classification more easy
8	1	Db1	3	1	1.0	0.8	Wavelet level 3 extract some useful features that make the model more general to unseen data
9	1	Db1	4	1	1	0.8	Wavelet level 4 extract features may be similar to the features of exp 8
10	1	Db1	5	1	1	0.5	Wavelet level 5 extract feature that make the model overfit the train data and loss the generalization to unseen data
11	1	Db2	6	1	1	0.9	Db2 with wavelet level 6 extract more useful features that make the well fitting and generalization possible (approximately :D)

Conclusion :

Due to the size of the data is small (Very small than usual) the difference between the SVM and KNN not clear here as they give the same train and test accuracy in Exp(11) KNN and Exp(7) SVM , but the SVM according to its nature it generate the most discriminative line to separate the data into two classes and we know that it may be possible for svm to get 100% test accuracy when we spend more and more time in finetune its parameters like c , gamma and kernel type but according to the number of the parameters needed to finetune we prefer to stop as the test accuracy not bad because it is small 10 examples and the svm get 9 of them correct so we prefer it unlike KNN where when we change the number of neighborhood other than 1 we get bad result.

One more experiment is a reason for SVM to be the winner we train the svm with the same setup of Exp(7) on the original data after preprocessing without extract features and we get the same result 100% for train accuracy and 90% for test , where the KNN with the same setup of Exp(11) get bad result 100% for train and 70 % for test witch indicate that there is an overfitting of the training data and loss in the generalization for unseen data , so we conclude that for svm with setup of exp(7) to reduce the computation time we can train and test on the original data after preprocessing without extract features , but for KNN it is necessary for extract features after preprocessing on the data to get good result.

But why SVM is the winner ? the kernel type parameter witch act as a transformer transform the data from its original space to another space such that the data in the new space more suitable for training, for example assume we have not linearly separable data so the kernel can transform it to another space such that the data in the new space is approximately linearly separable, where the KNN not transform the data to another space instead it works in its space regardless the nature of the data.

Thanks