ELECTROMYOGRAPHY (EMG) OF THE EXTRAOCULAR MUSCLES (EOM)

- Main Code: https://github.com/vasanza/EOG
- IEEEDataPort: https://dx.doi.org/10.21227/bhpj-mz94
- More Matlab Examples: https://github.com/Human-Machine-Interface
- Hardware: Instrumentation amplifier based on AD620
- Sampling Frequency = 120 Hz for approximately 2 seconds
- Electrical activity of EOG signals: Fmin=8 and Fmax=31
- Subjects: 10

Raw dataset preparation

```
clear;clc;%clear all
addpath(genpath('./src'))%functions folders
datapath = fullfile('./data/');%data folder
folders = FindFolders(datapath);
allData=[];
```

## Raw dataset preprocessing

% In this example no filtering was done, but it can be done

```
for i=1:length(folders)% Through all folders
    folderpath=fullfile(datapath,folders(i).name);%Select i folder
    filenames = FindCSV(folderpath);%List All CSV files
    for j=1:length(filenames)% Through all files
        data=readtable(fullfile(folderpath,filenames(j).name));%Select i CSV file
        dataNew=table2array(data);% Array Double
        DataNorm = fNormalization(dataNew);%Normalization
        Label = fLabelEOG(folders(i).name);%Name Folder to Label

        %max(EOG1,EOG2), min(EOG1,EOG2), mean(EOG1,EOG2), median(EOG1,EOG2),...
        %rms(EOG1,EOG2), std(EOG1,EOG2)
        DataFeatures = [max(DataNorm) min(DataNorm) mean(DataNorm)...
            median(DataNorm) rms(DataNorm) std(DataNorm) Label];%Feature extraction
        allData=[allData;DataFeatures];
    end
end
```
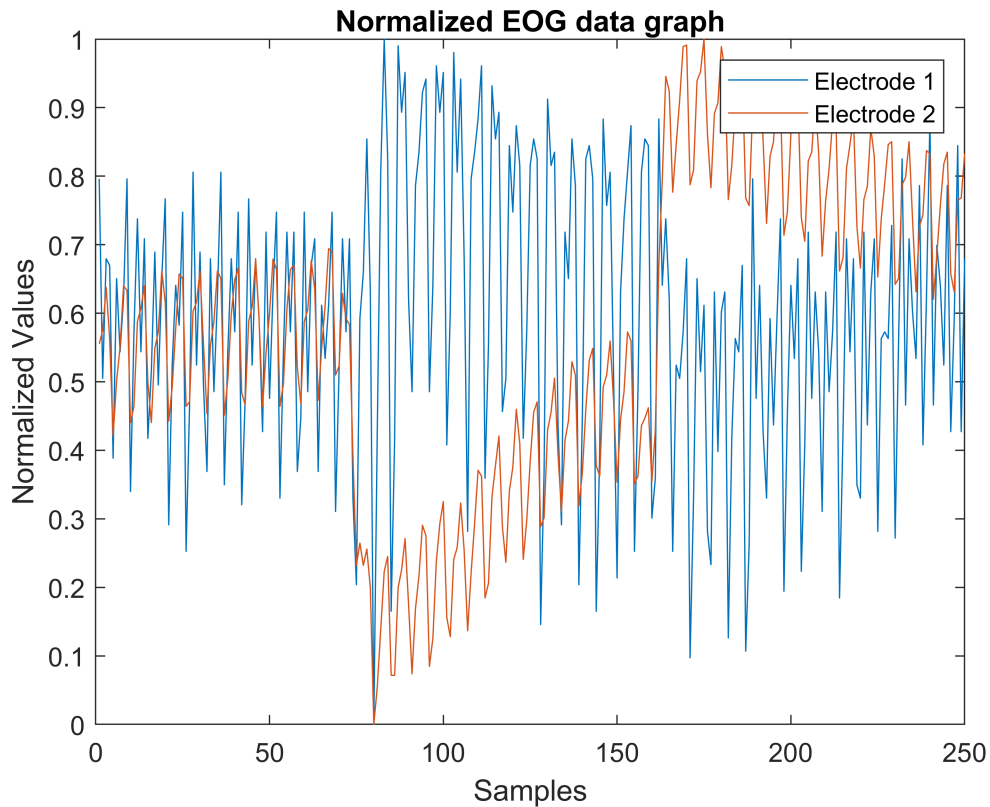
Plot Raw EOG dataset

```
figure
plot(dataNew);xlabel('Samples');ylabel('mVolts');
title('Raw EOG data graph');
legend('Electrode 1','Electrode 2');
```
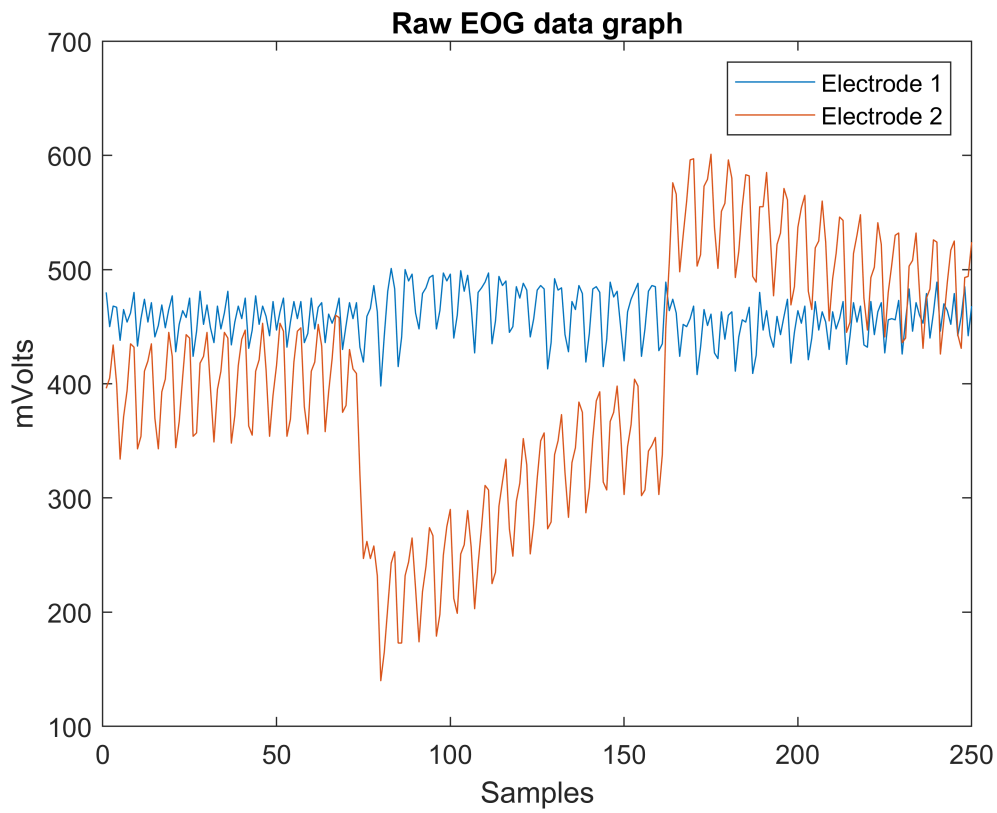
Raw EOG data graph

Plot Normalization EOG dataset

```
figure
plot(DataNorm);xlabel('Samples');ylabel('Normalized Values');
title('Normalized EOG data graph');
legend('Electrode 1','Electrode 2');
```

Normalized EOG data graph

Feature Selection

```
%max(EOG1,EOG2), min(EOG1,EOG2), mean(EOG1,EOG2), median(EOG1,EOG2),...
        %rms(EOG1,EOG2), std(EOG1,EOG2)
DataFeatureSelection=allData(1:end-1,1:end-1);%not including the label
R = corrcoef(DataFeatureSelection)
```

```
R = 12×12
     NaN      NaN      NaN      NaN      NaN      NaN      NaN      NaN ···
     NaN      NaN      NaN      NaN      NaN      NaN      NaN      NaN
     NaN      NaN      NaN      NaN      NaN      NaN      NaN      NaN
     NaN      NaN      NaN      NaN      NaN      NaN      NaN      NaN
     NaN      NaN      NaN      NaN   1.0000  -0.2582   0.9535  -0.3408
     NaN      NaN      NaN      NaN  -0.2582   1.0000  -0.3048   0.9415
     NaN      NaN      NaN      NaN   0.9535  -0.3048   1.0000  -0.4037
     NaN      NaN      NaN      NaN  -0.3408   0.9415  -0.4037   1.0000
     NaN      NaN      NaN      NaN   0.9597  -0.3195   0.9070  -0.3699
     NaN      NaN      NaN      NaN  -0.2929   0.9894  -0.3403   0.9546
     ⋮
     ⋮
```

```
%Remove NaN results
%mean(EOG1,EOG2), median(EOG1,EOG2), rms(EOG1,EOG2), std(EOG1,EOG2)
DataFeatureSelection=allData(5:end-1,5:end-1);%not including the label
R = corrcoef(DataFeatureSelection)
```

```
R = 8×8
    1.0000   -0.2672    0.9536   -0.3481    0.9591   -0.3017   -0.1405   -0.2543
   -0.2672    1.0000   -0.3114    0.9407   -0.3272    0.9892   -0.2539   -0.0500
    0.9536   -0.3114    1.0000   -0.4089    0.9062   -0.3468   -0.1358   -0.2704
```

```
   -0.3481     0.9407    -0.4089     1.0000    -0.3755     0.9540    -0.1586     0.1213
    0.9591    -0.3272     0.9062    -0.3755     1.0000    -0.3487     0.1387    -0.1745
   -0.3017     0.9892    -0.3468     0.9540    -0.3487     1.0000    -0.2106     0.0913
   -0.1405    -0.2539    -0.1358    -0.1586     0.1387    -0.2106     1.0000     0.2631
   -0.2543    -0.0500    -0.2704     0.1213    -0.1745     0.0913     0.2631     1.0000
```

```matlab
%Remove High correlation beatwen mean, meadian and rms
%rms(EOG1,EOG2), std(EOG1,EOG2)
DataFeatureSelection=allData(9:end-1,9:end-1);%not including the label
R = corrcoef(DataFeatureSelection)
```

```
R = 4×4
    1.0000    -0.3488     0.1381    -0.1756
   -0.3488     1.0000    -0.2031     0.0967
    0.1381    -0.2031     1.0000     0.2584
   -0.1756     0.0967     0.2584     1.0000
```

```matlab
%Save .CSV file with all EOG file features
csvwrite(strcat(datapath,'AllDataFeatures.csv'),allData(:,9:end));%Including the label
```

Motor Task Classification (Label + 8)

All Classifications Results:

- 1) CN, MR, ML, MU, MD, MP = "0", "1", "2", "3", "4", "5"
- 2) CN, MR, ML = "0", "1", "2"
- 3) CN, MP = "0", "5"
- 4) CN, MU, MD = "0", "3", "4"

```matlab
clear;clc;
addpath(genpath('./src'))%functions folders
path = fullfile('./data/');%data folder
% Upload .CSV file with the features of all EOG files
allData = fLoad_csv(path,'AllDataFeatures');

[CN, MR, ML, MU, MD, MP] = fIdxLabel_EOG(allData);%Index of rows by task type

num = input('Enter a number: ');
switch num
    case 1 %all task
        idx = [CN(1:end);MR(1:end);ML(1:end);MU(1:end);MD(1:end);MP(1:end)];
        disp('All task')
    case 2 %Horizontal Movements
        idx = [CN(1:end);MR(1:end);ML(1:end)];
        disp('Horizontal Movements')
    case 3 %Blinking
        idx = [CN(1:end);MP(1:end)];
        disp('Blinking')
    case 4 %Vertical Movement
        idx = [CN(1:end);MU(1:end);MD(1:end)];
        disp('Vertical Movement')
    otherwise
        disp('other value')
```

```
end
```

```
All task
```

```
TaskData=allData(idx,:);
```

Open the Classification Learner

```
%regressionLearner
classificationLearner
```

1) CN, MR, ML, MU, MD, MP = "0", "1", "2", "3", "4", "5"

☆ **1.1** Tree	Accuracy (Validation): 85.5%
Last change: Fine Tree	4/4 features

☆ **1.2** Tree	Accuracy (Validation): 84.7%
Last change: Medium Tree	4/4 features

☆ **1.3** Tree	Accuracy (Validation): 69.8%
Last change: Coarse Tree	4/4 features

☆ **1.4** Linear Discriminant	Accuracy (Validation): 82.8%
Last change: Linear Discriminant	4/4 features

☆ **1.5** Quadratic Discriminant	Accuracy (Validation): 93.3%
Last change: Quadratic Discriminant	4/4 features

☆ **1.6** Naive Bayes	Accuracy (Validation): 90.5%
Last change: Gaussian Naive Bayes	4/4 features

☆ **1.7** Naive Bayes	Accuracy (Validation): 92.3%
Last change: Kernel Naive Bayes	4/4 features

☆ **1.8** SVM	Accuracy (Validation): 88.5%
Last change: Linear SVM	4/4 features

☆ **1.9** SVM	Accuracy (Validation): 93.3%
Last change: Quadratic SVM	4/4 features

☆ **1.10** SVM	Accuracy (Validation): **93.5%**
Last change: Cubic SVM	4/4 features

☆	**1.11** SVM	Accuracy (Validation): 86.8%
	Last change: Fine Gaussian SVM	4/4 features
☆	**1.12** SVM	Accuracy (Validation): 93.0%
	Last change: Medium Gaussian SVM	4/4 features
☆	**1.13** SVM	Accuracy (Validation): 87.3%
	Last change: Coarse Gaussian SVM	4/4 features
☆	**1.14** KNN	Accuracy (Validation): 92.0%
	Last change: Fine KNN	4/4 features
☆	**1.15** KNN	Accuracy (Validation): 90.0%
	Last change: Medium KNN	4/4 features
☆	**1.16** KNN	Accuracy (Validation): 73.3%
	Last change: Coarse KNN	4/4 features
☆	**1.17** KNN	Accuracy (Validation): 86.2%
	Last change: Cosine KNN	4/4 features
☆	**1.18** KNN	Accuracy (Validation): 89.2%
	Last change: Cubic KNN	4/4 features
☆	**1.19** KNN	Accuracy (Validation): 91.0%
	Last change: Weighted KNN	4/4 features
☆	**1.20** Ensemble	Accuracy (Validation): 89.2%
	Last change: Boosted Trees	4/4 features

☆	**1.21** Ensemble	Accuracy (Validation): 90.8%
	Last change: Bagged Trees	4/4 features
☆	**1.22** Ensemble	Accuracy (Validation): 82.0%
	Last change: Subspace Discriminant	4/4 features
☆	**1.23** Ensemble	Accuracy (Validation): 85.5%
	Last change: Subspace KNN	4/4 features
☆	**1.24** Ensemble	Accuracy (Validation): 84.7%
	Last change: RUSBoosted Trees	4/4 features
☆	**1.25** Neural Network	Accuracy (Validation): 89.2%
	Last change: Narrow Neural Network	4/4 features
☆	**1.26** Neural Network	Accuracy (Validation): 91.2%
	Last change: Medium Neural Network	4/4 features
☆	**1.27** Neural Network	Accuracy (Validation): 91.3%
	Last change: Wide Neural Network	4/4 features
☆	**1.28** Neural Network	Accuracy (Validation): 88.5%
	Last change: Bilayered Neural Network	4/4 features
☆	**1.29** Neural Network	Accuracy (Validation): 91.3%
	Last change: Trilayered Neural Network	4/4 features

Best Classification algorithm

1.10 SVM
Last change: Cubic SVM

Accuracy (Validation): **93.5%**
4/4 features

▼ Current Model Summary

Model 1.10: Trained

Training Results
Accuracy (Validation)   93.5%
Total cost (Validation)   39
Prediction speed   ~4000 obs/sec
Training time   2.7749 sec

Model Type
Preset: Cubic SVM
Kernel function: Cubic
Kernel scale: Automatic
Box constraint level: 1
Multiclass method: One-vs-One
Standardize data: true

Optimizer Options
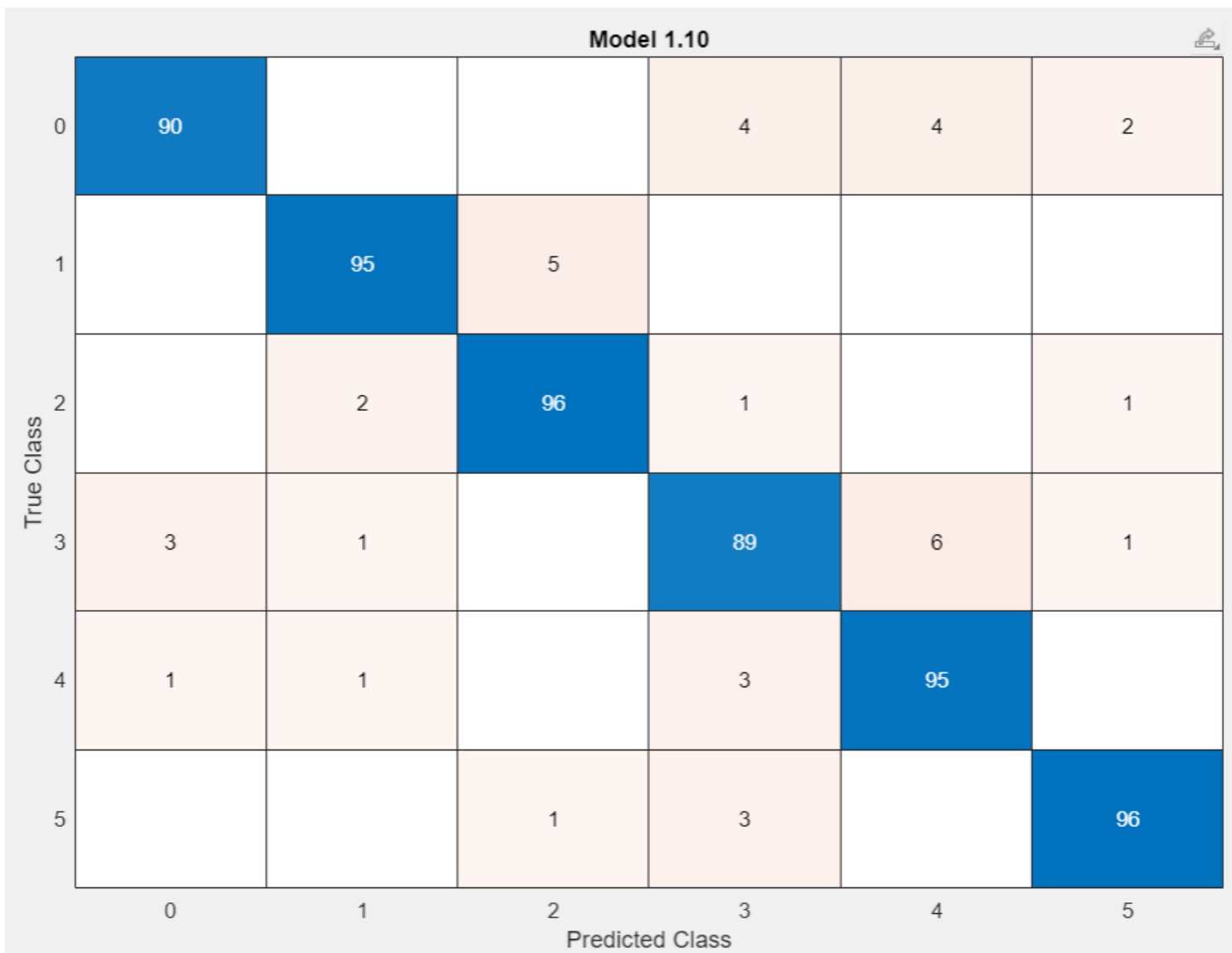Hyperparameter options disabled

Feature Selection
All features used in the model, before PCA

PCA
PCA disabled

Misclassification Costs
Cost matrix: default

Model 1.10

Model 1.10

AUC = 0.99

(0.01,0.90)

True positive rate

False positive rate

ROC curve
Area under curve (AUC)
Current classifier