EOG Movement Detection using Wavelet Transform

Ricardo Brioso, *Master's Student, FEUP* and Christophe Hurter, *Professor, ENAC*

***Abstract*—Write abstract.**

***Index Terms*—eog, electrooculography, saccade, eye movement, blink removal, pre-processing, wavelet transform, peak detection, machine learning**

I. Introduction

AIrplane pilots and professions alike require a tremendous amount of attention in order to maintain safety in their environment. Therefore, it is interesting to study what could cause or stimulate distraction in these situations.

The human brain, as it goes through the process of recalling a memory, usually takes away attention from the external environment [ref1]. This decoupling from the external environment can be observed in eye movements, for example gazing at the ceiling or the floor while trying to remember something. It is believed that looking at a neutral background such as the floor or the ceiling can help the brain abstain from the outside world and focus into its internal world in order to retrieve a memory. [ref2].

The purpose of this project is to analyse electrooculogram (EOG) data from several recordings for detecting and classifying different movements of the eye, focusing mostly on saccades.

The EOG is analysed with state of the art signal processing techniques after pre-processing the datasets and the algorithm's performance is evaluated using ground truth labels for each eye movement.

Lastly, this algorithm is aimed to be used for analysis of neuropsychology studies that have EOG data. To determine some of the statistical characteristics of these eye movements in the different parts of the experiments.

The code for this project is available at https://github.com/RicardoCQB/enac-eog-analysis.

II. Background

Here is presented some quick basic definitions of certain eye movements and other concepts:

The **EOG** is the signal obtained from the potential difference between the cornea and the retina. As the eye moves around, the distance between the cornea and retina relative to the electrodes changes.

For example, if a signal is measured with two electrodes, one electrode above the eye and the other electrode bellow the same eye. As the eye goes up, the potential difference between the up electrode and the down electrode would be higher.

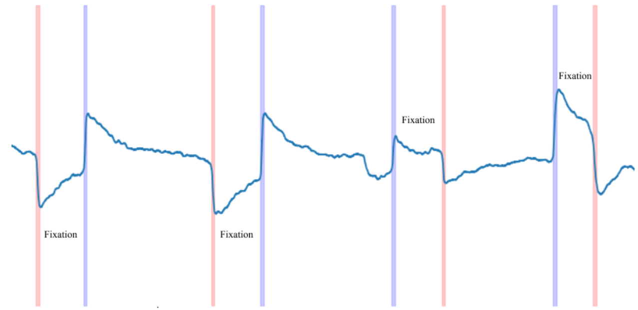

Fig. 1. EOG signal with saccades and fixations. Blue and red labels correspond to upward and downward saccades, respectively. The interval between the blue and red labels corresponds to the eye fixation.

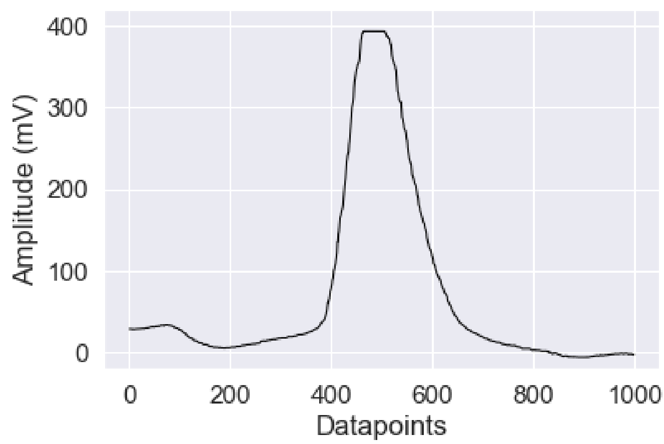

Fig. 2. Blink present in the vertical EOG signal.

A **saccade** is a type of eye movement that occurs when both eyes move rapidly in the same direction. This movement can be reflexive or voluntary, usually happens before and after a fixation on an object, or can be happen during visual search or memory retrieval. In the EOG signal, a plateau with high amplitude corresponds to a saccade followed by a **fixation** and then another saccade, as it is shown in Fig. 1.

A **blink** is a well known movement that corresponds to the closure of the eyelids. When the subject blinks, the EOG vertical signal shows a very high amplitude and low duration peak (Fig. 2).

There is also an eye movement called **smooth pursuit** that in contrast to the saccade (fast and abrupt movement) is slow and usually happens when following an object that is slowly moving relatively to the observer's field of view.

III. Similar work on the subject

Several articles that mentioned EOG and its applications were analysed. The purpose of this work is classify eye movements using the EOG signal, therefore the articles that mentioned some kind of classification of eye movements were gathered for keyword, method and accuracy extraction.

Since different articles had different methods and datasets it is not very interesting to compare them between each other. Nonetheless, it's useful to know what different methods have been used and their success rate.

Fig. 3. WordCloud made with keywords of the articles in Table I

TABLE I
ARTICLES THAT MENTION CLASSIFICATION OF EYE MOVEMENTS OR ACTIVITY RECOGNITION BASED ON EOG SIGNAL.

Author	Year	Method	Accuracy
Bulling [19]	2008	Wavelet Transform	87%
Bulling [20]	2009	-	76%
Usakli [65]	2010	Nearest Neighbor Relation	95%
Vidal [24]	2011	K-Nearest Neighbor (K-NN)	-
Bulling [1]	2011	Wavelet Transform	94%
Banarjee [42]	2012	K-NN	69%
Hossain [81]	2014	Velocity, Differentiation	95%
OuYang [84]	2015	Wavelet Transform	90%
Hossain [87]	2015	Feature Extraction (FT)	100%
Barbara [97]	2016	Threshold Algorithm	73%
Lee [101]	2017	Wavelet Transform	87%
Thakur [107]	2017	Wavelet Transform and FT	99%
Heo [110]	2017	-	95%
Zheng [61]	2017	Wavelet Transform	-
[142]Merino	2010	Derivative and amplitude	94%
[140]SAMANN	2017	-	95%
[141]Reda	2019	K-NN, LDA, SVM	94%
[144]Cafasso	2017	Wavelet Transform and FT	93%

In Table 1, some of the articles have the objective of developing a way for the algorithm to recognize eye movements or the activity that the subject is doing. All of these analysis are done using the EOG signal. For example, in Bulling (2008), the Continuous Wavelet Transform was used successfully to classify and recognize gestures executed by the subjects using the EOG signal.

Electrocardiogram signal (ECG) has several public classification challenges and public datasets. So EOG datasets are scarcer and therefore it is more difficult to compare different methods of classification on EOG.

EOG signal eye movement classification is very useful for neuropsychology, activity recognition and Human Computer Interfaces (HCI). So improving these techniques and organising public datasets could be promising for the study of the EOG signal.

IV. Signal Pre-Processing

In signal processing, before analysing the signal it is strongly recommended to apply some pre-processing to the data. The EOG signal usually has high frequency noise present in the signal and the presence of baseline wandering (very low frequency waveform across the signal).

A. Bipolar Transformation

This step can be considered part of the pre-processing. The typical layout of electrodes of the data used, consists in using two electrodes to obtain the vertical EOG signal and two other electrodes for the horizontal EOG signal (as it can be seen in Figure 4).

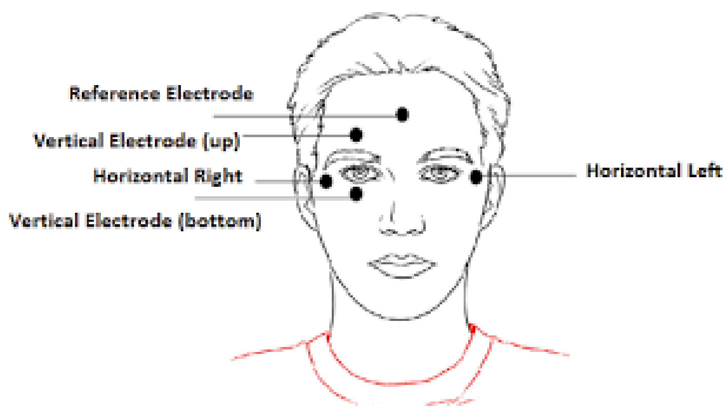

Fig. 4. EOG electrode configuration used.

The bipolar transformation turns 4 signals that origin from the 4 basic electrodes (up, down, left and right) into the two relevant EOG signals, vertical and horizontal signal. This transformation is showed in the following figures, 5 and Figure 6

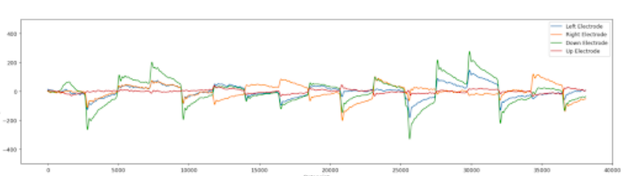

Fig. 5. EOG electrodes individual signals

B. Signal Noise Reduction

The most common ways of reducing the noise of EOG signal is to use a median filter or a low-pass filter, as it is done in [insert references]. The low-pass filter removes the noise by eliminating higher frequencies in the signal and the

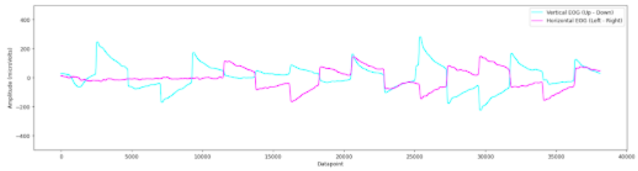

Fig. 6. EOG vertical (cyan) and horizontal (magenta) signal after bipolar transformation.

median filter removes the noise by averaging the signal in a small filter window.

The method in this approach is the median filter with a 200 samples window. The difference between the raw signal and the denoised signal are shown in Figure 7.

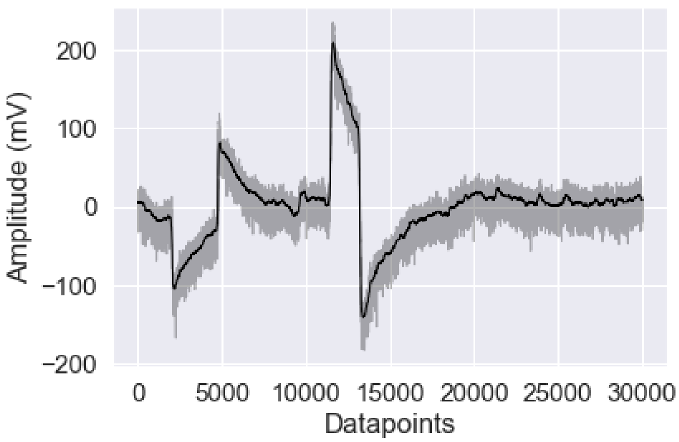

Fig. 7. EOG signal after the application of median filter with 200 samples of sliding window (the darkest like corresponds to the denoised signal).

C. Baseline Wandering Reduction

Baseline signal corresponds to a very low frequency artefact that the EOG might have when recorded. In current literature [reference to review], several methods for baseline wandering reduction were tested and in this work, it was tested two of those methods. But the resulting signal after the baseline wandering reduction was very similar to the initial signal. Therefore, this part of the pre-processing was not included in the general pipeline for the signal analysis.

In Figure 8. a Butterworth high-pass filter with a $0.04\,\mathrm{Hz}$ cut-off frequency was applied to the initial signal and the baseline in the entire signal does not change.

V. Signal Analysis using Continuous Wavelet Transform

In recent literature [list article refs], **Continuous Wavelet Transform (CWT)**was widely use for detection of eye movements. In this project it was also applied the Continuous Wavelet Transform to the EOG signal, this CWT was used with a **Mexican Hat wavelet** (also called Ricker Wavelet) of scale of 30 (Figure 9). Various scales and wavelets were testes using the PyWavelet Python's library.

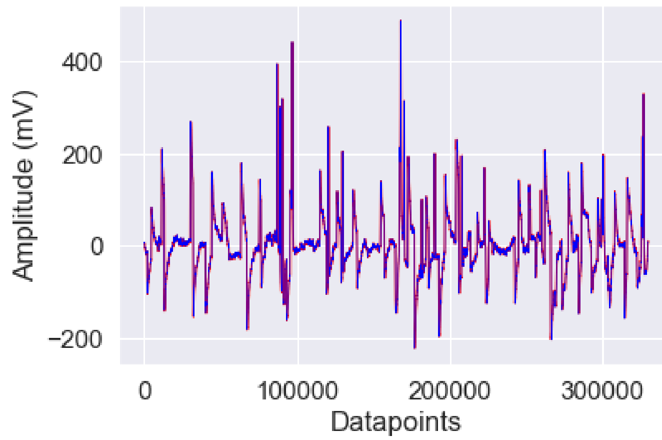

Fig. 8. EOG signal before and after applying a high-pass filter with 0.04 cutoff frequency.

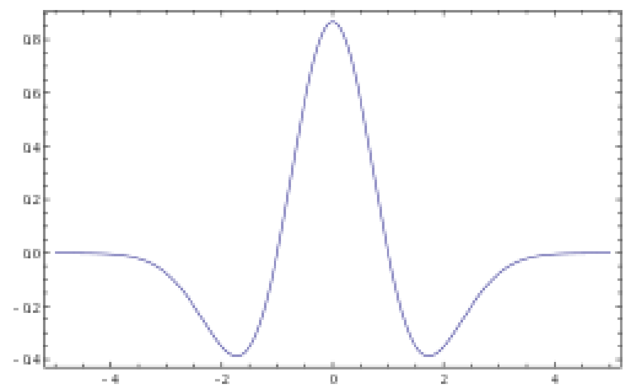

Fig. 9. Mexican Hat or Ricker Wavelet.

In the following Figure 10 we can the initial EOG horizontal signal in both graphs (in magenta and in black) but in the second graph we can see a dashed line that represents the coefficients of the CWT of the signal. As it is observed, the coefficient data has very sharp impulses that are present when the original EOG horizontal signal has a fast rise or fast downfall.

This could be interpreted as the correlation between the Ricker Wavelet 9 and the EOG horizontal signal, as the signal becomes more similar to the Mexican Hat wavelet, the stronger the correlation and therefore the bigger is the peak in the CWT coefficients.

Following the CWT and after obtaining the coefficients, it is used a peak detector function from the Scipy library, after tweaking the peak detection parameters manually to find the best peak detection, the peaks are translated into a sequence of 0's (peaks that are positive or pointing up) and 1's (peaks that are negative or pointing down).

The saccades were then classified as a 'positive' saccade or as a 'negative' saccade, that translates to 'up' and 'down' for vertical EOG and 'left' and 'right' for horizontal EOG, respectively. A 'positive' saccade is identified by having one

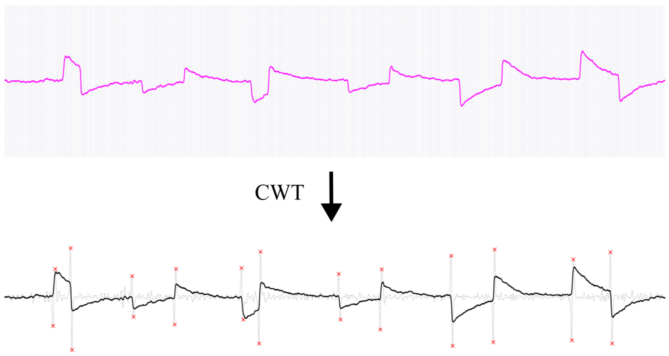

Fig. 10. EOG horizontal signal (magenta and black lines) and EOG horizontal signal CWT coefficients (dashed grey line).

negative peak followed by one positive peak in a very fast succession. And the 'negative' saccade is identified by having a positive peak followed by a negative peak in a very fast succession, as it is visually explained in Figure 11.

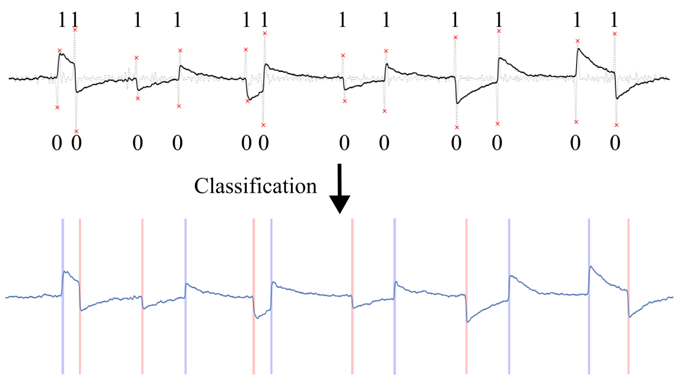

Fig. 11. Classification of 'Positive' and 'Negative' Saccade Example ('Positive' Saccades in blue and 'Negative' Saccades in red).

VI. Algorithm Evaluation

The evaluation of this algorithm was done by using **groundtruth** annotations of the eye movements present in the signal, such as saccades and their directions and the blinks.

The groundtruth annotations were made by using a custom tool that could display the signal and register the interval of the signal that corresponded to an eye movement.

In Figure 12, two graphs can be observed, the upper graph corresponds to part of a horizontal EOG signal whose eye movements were classified. The second graph shows the groundtruth annotations made with the custom tool. It is obvious that the groundtruth saccade intervals are greater then the intervals corresponding to the detection.

Therefore, the evaluation counted as a correct classification a saccade interval that was **contained** in a groundtruth interval.

True Positive (TP) - Detected movement that was present in the groundtruth
True Negative (TN) - Detected movement that was misclassified.
False Positive (FP) - Detected movement that was not present in the groundtruth

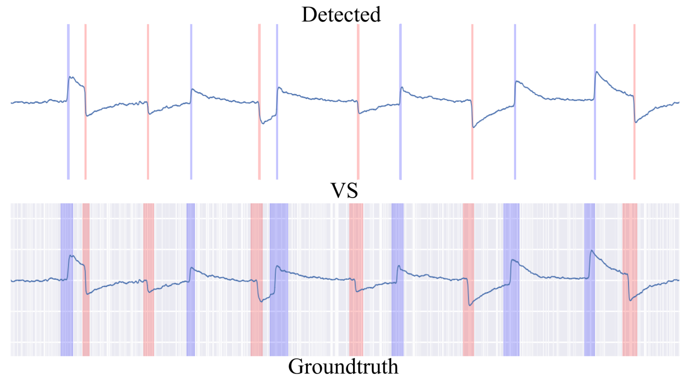

Fig. 12. Detected Saccades vs Groundtruth Saccade Annotations.

The algorithm was tested in several datasets and in the following Table II we can observe the performance metrics for each dataset:

TABLE II
Performance of the tested datasets

Dataset	Precision	Recall	F1-Score
ricardo1	0,85	0,95	0,9
ricardo2	0,86	0,99	0,92
ricardo3	0,88	0,98	0,93
guillaume1	0,49	0,92	0,64
guillaume2	0,41	0,96	0,57
wassim1	0,76	0,91	0,83
wassim2	0,84	0,89	0,86

The average F1-Score is 80%, it is lower than the performance of other works on the subject, as shown in Table I. But in each article we have a different data set and different evaluation method. In order to compare results, it would be necessary to test it in the same conditions.

In these datasets, the **precision is always lower than the recall**. The mean precision is **73%** and the mean recall is **94%**. This means that the algorithm often classifies parts of the signal that do not correspond to eye movements in the ground truth. But on the other hand, it usually classifies correctly the eye movements when they exist in the ground truth.

There exists a confusion matrix for every dataset, that informs how many movements were correctly classified and how many were miss classified as the opposite movement. These confusion matrices can be observed in the following Figure **??**.

The count and visualization of the **false positives** (Table III) is important to know how many parts of the signal are being mistaken with saccades. In the dataset **guillaume1**, for example, we can see a lot more false positives for the vertical EOG saccade detection (up and down saccades) than for the horizontal EOG detection.

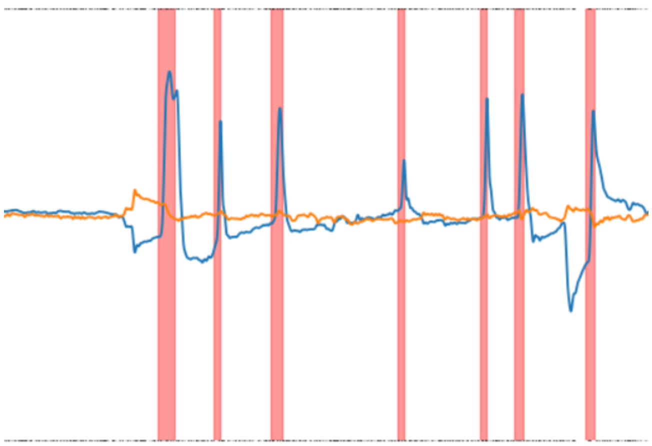

Fig. 13. Detected Blinks using EOGERT algorithm.

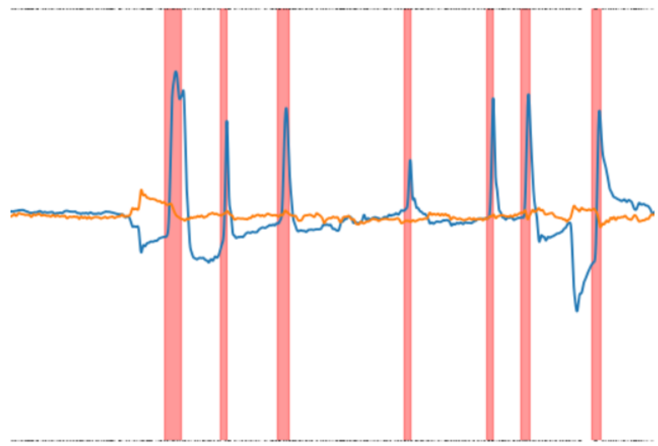

Fig. 14. Detected Blinks using EOGERT algorithm.

This can be explained by the presence of blinks in the vertical EOG, because of the rise speed of blinks, they can also be detected using the CWT. Although before counting the saccades, the blinks are detected and signaled so they aren't counted as saccades. So the explanation for this difference in the false positives may lay on noise of the acquisition of this signal in specific or miss classification of the blinks.

In Table IV we can see the difference in performance in three of the signals were the blink intervals were not ignored for the saccade detection.

TABLE III
FALSE POSITIVES OF THE TESTED DATASETS

Dataset	Up FP	Down FP	Left FP	Right FP
ricardo1	21	10	4	6
ricardo2	21	15	11	9
ricardo3	14	12	17	15
guillaume1	142	145	42	41
guillaume2	91	84	25	21
wassim1	69	36	14	17
wassim2	8	10	7	10

VII. Blink Detection Methods

Blink detection and removal methods are more common than saccade detection methods because of the blink's presence in some EEG signals as artifacts.

In this project, the neurokit2 library [ref neuroki2] was used to test blink detection, it compiled different state of the art blink detection methods from various authors.

Although, after some testing on the datasets, the blink detection algorithm used was EOGERT [ref]. EOGERT had a mean of 94% precision and 93% recall in detecting blinks, see Figure 14.

TABLE IV
PERFORMANCE FALSE POSITIVES OF THREE DATASETS WITHOUT BLINK REMOVAL

Dataset	Precision	Recall	F1-Score	Up Down Left Right FP
ricardo1	0,72	0,95	0,82	45 32 4 6
ricardo2	0,73	0,98	0,84	64 45 11 9
ricardo3	0,8	0,98	0,88	43 33 17 15
Averages	0,75	0,97	0,85	

VIII. Averaging the eye movements in the datasets

Using the ground truth of all the datasets, all the movements were collected and averaged so it could be observed the general shape of each movement in each dataset.

For the **saccade averaging** , the ground truth interval was used and centered using the peaks of the CWT coefficients in the signal.

In the **blink averaging**, the intervals were centered by assuming the peak of the blink would be maximum value of the signal in that small ground truth interval. In Figure 15, it is shown the code block responsible for centering and normalizing the blinks in order to make a graph like in Figure 16.

```
for blink in blinkGroundTruth:
    blinkDataPoints = verticalEOG[blink[0]:blink[1]]

    blinkPeak = max(blinkDataPoints)

    blinkDataPoints = blinkDataPoints.tolist()
    blinkPeakIndex = blinkDataPoints.index(blinkPeak)

    centerIndex = blink[0] + blinkPeakIndex

    blink[0] = centerIndex - int(blinkIntervalSize/2)
    blink[1] = centerIndex + int(blinkIntervalSize/2)
    blink[2] = blinkPeakIndex
    newGroundTruth.append(blink)
```

In the following images, we can see the different averaged eye movements for each dataset.

IX. Conclusion

In summary, this paper addresses

Acknowledgment

References

```python
for blink in blinkGroundTruth:
    blinkDataPoints = verticalEOG[blink[0]:blink[1]]

    blinkPeak = max(blinkDataPoints)

    blinkDataPoints = blinkDataPoints.tolist()
    blinkPeakIndex = blinkDataPoints.index(blinkPeak)

    centerIndex = blink[0] + blinkPeakIndex

    blink[0] = centerIndex - int(blinkIntervalSize/2)
    blink[1] = centerIndex + int(blinkIntervalSize/2)
    blink[2] = blinkPeakIndex
    newGroundTruth.append(blink)
```

Fig. 15. Blink Averaging block of code

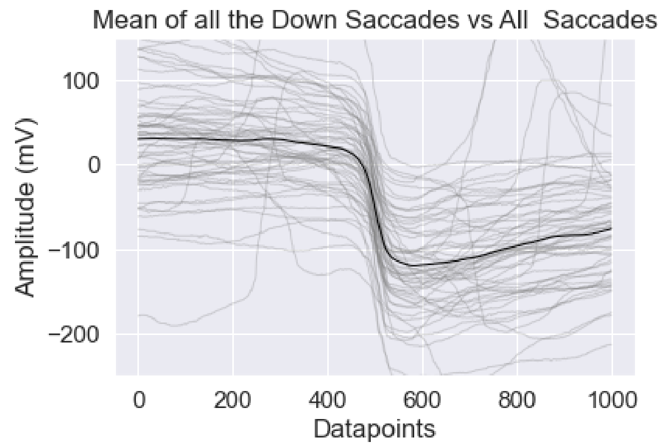

Fig. 18. Average of all the down saccades in dataset ricardo1

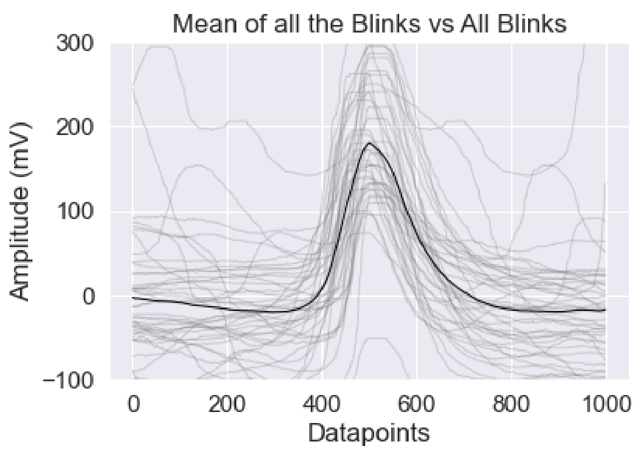

Fig. 16. Average of all the blinks in dataset ricardo1

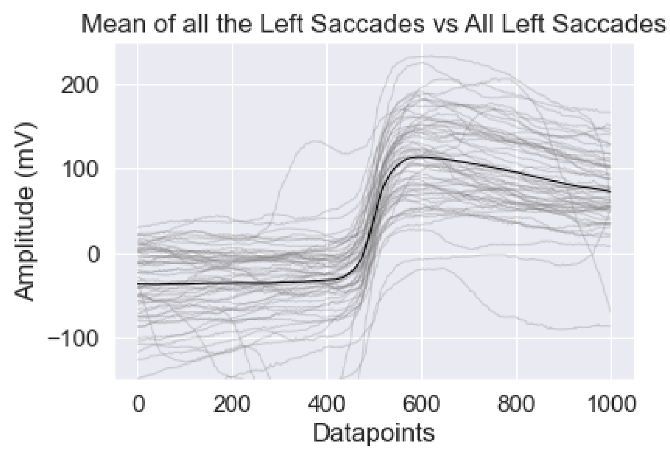

Fig. 19. Average of all the left saccades in dataset ricardo1

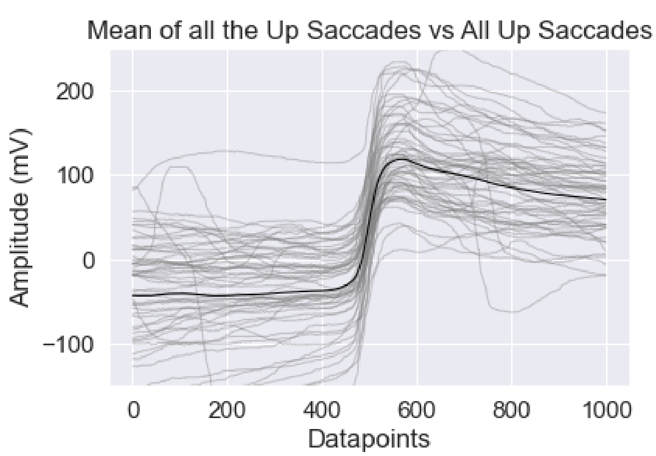

Fig. 17. Average of all the up saccades in dataset ricardo1

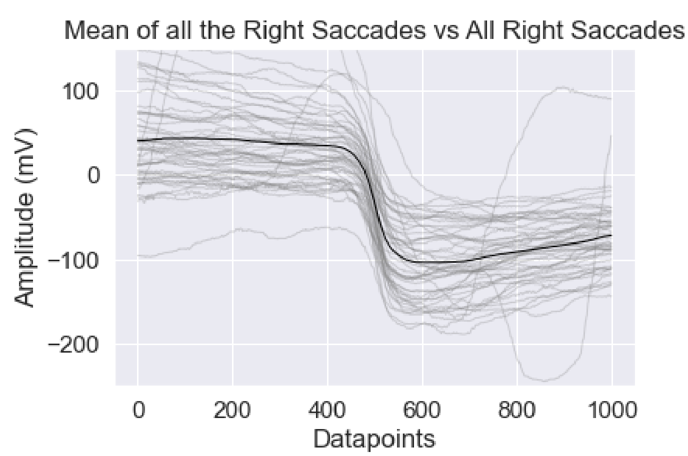

Fig. 20. Average of all the right saccades in dataset ricardo1