An Automatic System for Characterization and Detection of Ocular Noise

by

Alexander Melville

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Engineering
(Computer Engineering)
in the University of Michigan-Dearborn
2017

Master's Thesis Committee:

Assistant Professor Omid Dehzangi, Chair
Associate Professor Di Ma
Associate Professor Hafiz Malik
Associate Professor Bruce Maxim
Associate Professor Brahim Medjahed

Table of Contents

Abstract iv

List of Tables v

List of Figures vi

List of Abbreviations vii

Chapter I: Introduction 1

1.1: The Blink Detection Task 1

1.2: Current Methods in Blink Detection 1

1.3: Thesis and Contribution 3

1.4: Overview of Proposed System 3

Chapter 2: EOG Regression and Autoregressive Blink Detection 6

2.1: EOG Regression 6

2.3: Autoregressive Blink Detection 12

2.4: Conclusion 14

Chapter 3: Label Generation using Dynamic Time Warping 16

3.1: Template Selection 17

3.2: DTW Performance 18

3.3: Conclusion 22

Chapter 4: Time Space Blink Detection 23

4.1: Feature Selection and Extraction 23

4.2: Soft-Margin Support Vector Machine (SVM) Classification 27

4.3: Experimental Setup (Workload Assessment Task) 29

4.4: Detection and Feature Ranking Methods 30

4.5: Conclusion 36

Chapter 5: Independent Component Space Blink Detection 37

5.1: The runica Algorithm 38

5.2: The Effect of ICA on EEG Data 38

5.3: The Effect of ICA on Blink Detection 40

Conclusion 43

References 45

Abstract

Eye blinks cause high amplitude noise in electroencephalograms (EEGs), the noise from these blinks causes interference in several very important frequency bands. The method detailed in this paper uses independent component analysis and a diversified feature space to identify and filter out eye blink noise during wearable electroencephalographic tests. Prior work used autoregressive modeling in the time domain to identify blink segments in the recorded data. While the previous autoregressive method showed high accuracy in short trials, the goal of this work is to create a more advanced system capable of filtering blink noise in long, continuous trials. One of the major applications for this system is improving the quality of data collected during workload assessment tasks. Trials that consider the subject's workload over time involve sensitive calculations done over the long term, and blinking resides in frequency bands that are known to be useful in determining the subject's csurrent workload. A blink in one of these bands could give a false positive result for workload, or it could confuse an algorithm during training. In smaller studies subjects have been told not to blink, or were told to keep their eyes closed, but for workload assessment tasks it's usually not practical to tell the subject to not blink during a strenuous trial. Other methods have been introduced that involve electrooculogram (EOG) data; the proposed system only uses electrooculogram data for training purposes, after this channels can be removed, so that wearable systems can reduce the amount of data recorded per second.

List of Tables

Table 1: Features listing by index 30
Table 2: Blink Detection Accuracy by Channel 32
Table 3: Time Space: Top 20 Best Features 33
Table 4: Time Domain: Channel 1 Best Features 34
Table 5: Time Domain: Channel 2 Best Features 35
Table 6: Time Domain: Channel 3 Best Features 35
Table 7: Time Domain: Channel 4 Best Features 35
Table 8: Time Domain: Channel 5 Best Features 35
Table 9: IC Space: Best Features for each Independent Component 40
Table 10: IC Space: Best Features for IC 1 41
Table 11: IC Space: Best Features for IC 2 41
Table 12: IC Space: Top 20 Best Features 41
Table 13: Top 10 Time Space Features: p-Values 42
Table 14: Top 10 Independent Component Features: p-Values 43

List of Figures

Figure 1: Training Data Annotation Process 3
Figure 2: Training Data Feature Extraction Process 4
Figure 3: Time space blink detection model training process 5
Figure 4: ICA blink detection model training process 5
Figure 5: EOG Regression Constant k Equation 7
Figure 6: EOG Regression Constant C equation 7
Figure 7: 10 Seconds of Raw EEG Data 8
Figure 8: 10 Seconds of Filtered EEG Data using EOG Regression 8
Figure 9: Approximately 1 second of EOG regression results 9
Figure 10: Distortion caused by only using a single EOG channel 10
Figure 11: Raw VEOG and HEOG signal 11
Figure 12: VEOG and HEOG with regression coefficients applied 11
Figure 13: A Graphical representation of DTW finding the shortest distances between graphs 16
Figure 14: Blink Pattern 18
Figure 15: DTW distances of all four templates on a sample blink 19
Figure 16: DTW performance during saccade 20
Figure 17: Performance of DTW under noisy conditions 21
Figure 18: DTW performance on small blinks 22
Figure 19: Possible learners with a small amount of features 23
Figure 20: Possible learners when the number of features increases 24
Figure 21: Soft margin SVM hyperplane, margin, and distances to samples 28
Figure 22: A graphical comparison of soft and hard margin SVM 28
Figure 23: Workload Assessment Task Channel Locations 29
Figure 24: EOG Signal (HEOG on top, VEOG on bottom) 39
Figure 25: Independent Components 40

List of Abbreviations

- EEG: Electroencephalogram – An acquisition system for recording neural activity, consisting of a series of electrodes placed on the scalp plus recording equipment.
- EOG: Electrooculogram – A system used to measure electrical potentials due to eye movement, consisting of two or four electrodes placed nearby the eyes plus recording equipment. The EOG can be set up to be vertical, horizontal, or both (see below).
- VEOG: Vertical Electrooculogram: This is an electrooculogram system that records using electrodes above and below each eye.
- HEOG: Horizontal Electrooculogram: This is an electrooculogram system that records using electrodes to the left and right of each eye.
- ICA: Independent Component Analysis, an algorithm for finding statistically independent signals found in a multi-channel signal.
- IC: Independent Component: A single independent component signal obtained by running independent component analysis
- SVM: Support Vector Machine
- DTW: Dynamic Time Warping
- SFT: Short Fourier Transform

Chapter I: Introduction

The machine learning system for blink detection presented in this thesis has two major components: Automatic data annotation, and model training using features. In order to make the system as accurate as possible, the training data must consist of a large variety of features extracted from the input data. A theoretical ideal model for blink detection would contain only the relevant features that are needed to detect blinks, and would detect blinks with no error. The main goal in feature selection is to find features that can create a hypothesis class that is capable of generating this ideal model. The features that would create this ideal model are not known, so it is necessary to extract and test different categories of features from the input data, so that the feature space contains as relevant of information as possible to the generalization ability of the predictor.

1.1: The Blink Detection Task

The Electroencephalogram (EEG) is a recording of the electrical potentials recorded from several electrodes placed on the scalp. While the skull is a very good insulator, some brain activity makes it through the bones and up to the scalp to be measured. One of the challenges of using an EEG to analyze the mental state of a subject is that blinking and eye movement cause high amplitude noise in the frontal EEG channels. This noise occurs in a very similar frequency band to real neurological data, so it's not possible to use a band pass filter to remove it. More sophisticated filters are typically implemented in order to filter out this noise without severely affecting useful parts of the EEG signal.

1.2: Current Methods in Blink Detection

EOG regression is a common and lightweight solution for removing ocular noise. It has been mentioned in (Croft & Barry, 1998) and (Schlögl et al., 2007), this method has been used as a basis for comparison against newer methods, for example in (He, Kahle, Wilson, & Russell, 2005). While electrooculogram (EOG) regression is simple to implement and quick to process, this method has been known to remove neural data as well as ocular noise because some neural data propagates forward into the EOG channels. In addition to this, regressive methods have limitations on non-stationary signals like the data collected during EEG trials.

Chapter I: Introduction

Independent Component Analysis (or ICA) has been shown to be a competent method for separating out independent components from EEG signals in many papers. (Siew Cheok & Raveendran, 2008) demonstrated that it was possible to estimate EOG recordings using ICA and EEG data. (Makeig, J. Bell., Jung, & Sejnowski, 1996) is an early work in ICA and EEG research, their results showed examples of independent components separating out various neural and noise activity(Wang et al., 2014) proposed a method that used both ICA and autoregressive methods for ocular noise removal, in order to build a model for noise correction that did not remove as much neural activity as ICA by itself. (Mahajan & Morshed, 2015) designed a system for noise removal using ICA, including features such as kurtosis, modified multiscale sample entropy, and wavelet analysis.

(Winkler, Haufe, & Tangermann, 2011) is especially relevant to this work's subject. The work by Winkler, Haufe, and Tangermann involved collecting a set of features on independent components in order to detect many different kinds of noise, using a 121 channel EEG. Unlike Winkler, Haufe, and Tangermann, this study targets blink detection in wearable technology with a far lower number of channels (6 channels), and this study took a greater number of features under consideration.

(Kong et al., 2013)'s work in this subject showed that it's possible to create an accurate eye blink identification system using independent components and power distribution conditions. Kong, et al's work provides strong support for the idea that blinking is effectively separated out of the EEG signal using Independent Component Analysis.

(Ghandeharion & Erfanian, 2010) is another ICA and statistics based study that involved 6 channels, EOG, and feature extraction. Their system did not involve machine learning, instead it used voting between a variety of features. A window of independent component data was assessed based on its wavelet and kurtosis features, as well as its correlation to two chosen reference signals, the amplitude of the EOG, and mutual information between an independent component and the reference signals.

The works that use Independent Component Analysis get very high accuracy but the ICA algorithm has very long, hard to predict processing times that would be difficult to bring into real time. A high accuracy blink detection system without the need for ICA would be far more useful in the long term for real time systems. Some of the methods investigated in the literature survey for this paper compare their results to expert opinions on what a "clean" EEG is, and some other studies use simulated EEG data. The simulated EEG studies introduce known blinking noise into a signal thought to be clean (e.g., blinking noise could be introduced into EEG data recorded while the subject's eyes were closed). The clean EEG signal and the raw EEG signal are compared using various techniques such as mutual information and variance. The validation done in the methods presented in this thesis focuses on the

accuracy of the classifier itself. This is an indicator of how well the classifier would integrate into an existing system. This study also focuses on how relevant features in EEG signals are to analyzing whether a window of data contains a blink.

1.3: Thesis and Contribution

In this thesis, a feasible system for labeling EOG data and detecting blink activity in segments of wearable headset EEG data is proposed. The system consists of labels (blink or non-blink) obtained by running DTW on the VEOG data from a trial, features extracted from the EEG signal, and a model which is trained to recognize a window of time as either blink or non-blink. A comparison of the performance of ICA and time space feature extraction was performed as well. The results showed that ICA did improve detection accuracy, but the two methods were close in performance. With improvements to the automatic label generation algorithm and feature space, the accuracy of this system will improve significantly.

1.4: Overview of Proposed System

1.4.1: Training Data Annotation

This step in the process creates aligned labels out of the electrooculogram data. A set of known blink patterns were selected as templates for use in the dynamic time warping (DTW) algorithm.

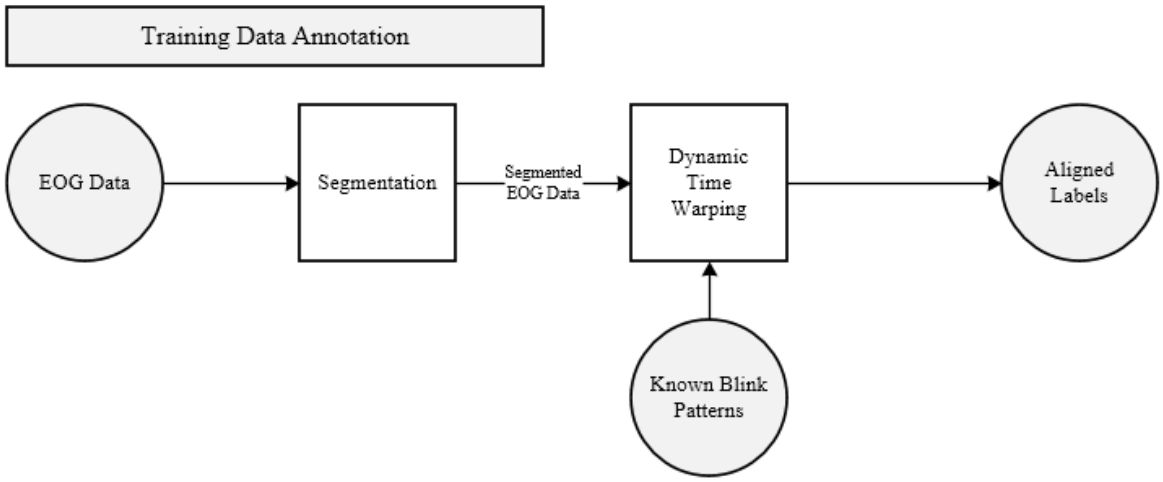

Figure 1: Training Data Annotation Process

As shown in Figure 1, the EOG data is segmented (into 250 sample windows with 90% overlap) before it is passed into the Dynamic Time Warping (DTW) algorithm. The window's distance from the four selected blink templates is stored for use as labels later on in the algorithm.

1.4.2: Training Data Feature Extraction

The features used for blink detection are extracted in this step. For comparison purposes, features are extracted from both the raw EEG signals and the independent components (ICs) from the EEG signals.

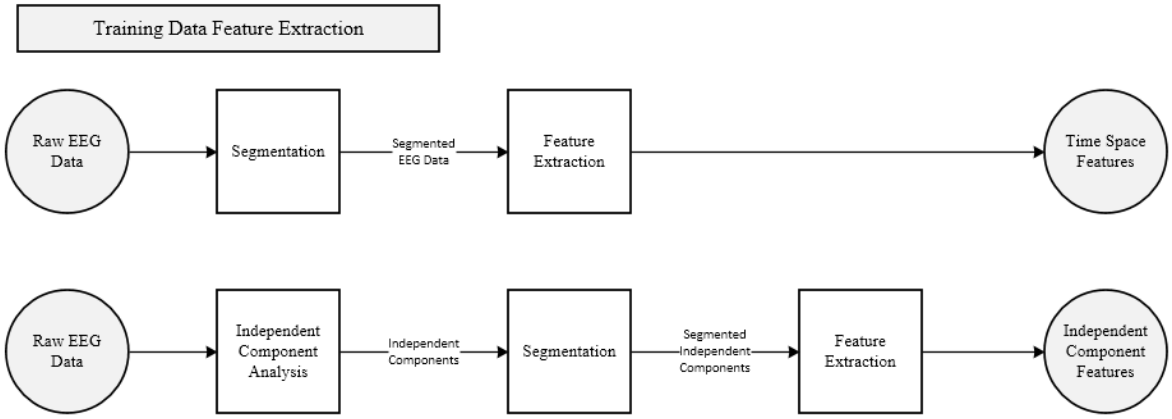

Figure 2: Training Data Feature Extraction Process

As shown in Figure 2, the raw EEG data is processed in two different ways in this step. For the time space features, the EEG data is segmented into 250 sample windows with 90% overlap, and then features are extracted from each window. For the second part of this step, Independent Component Analysis (ICA) is used to extract independent components from the EEG signals. The independent components are then segmented (also into 250 sample windows and 90% overlap) and features are extracted from each window in each independent component.

1.4.3: Blink Detection Model Training

Using the generated labels and extracted features (see above), the proposed system trains two sets of Support Vector Machine (SVM) models, in order to compare their results.

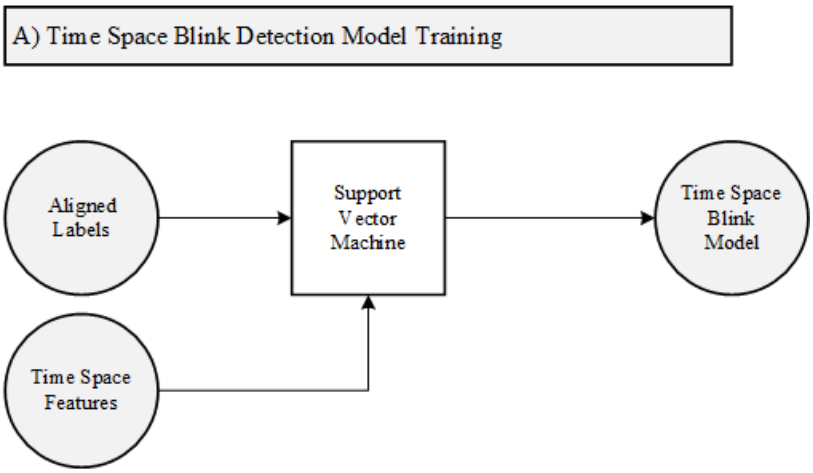

Figure 3: Time space blink detection model training process

As shown in Figure 3, the model training process uses a discriminative classifier (the Support Vector Machine algorithm) to identify the boundaries between blink and non blink windows in the selected feature space. It is not known which features (and which feature values) can be used to detect blinks, the SVM algorithm's purpose in this system is to discover these boundaries iteratively based on sample data.

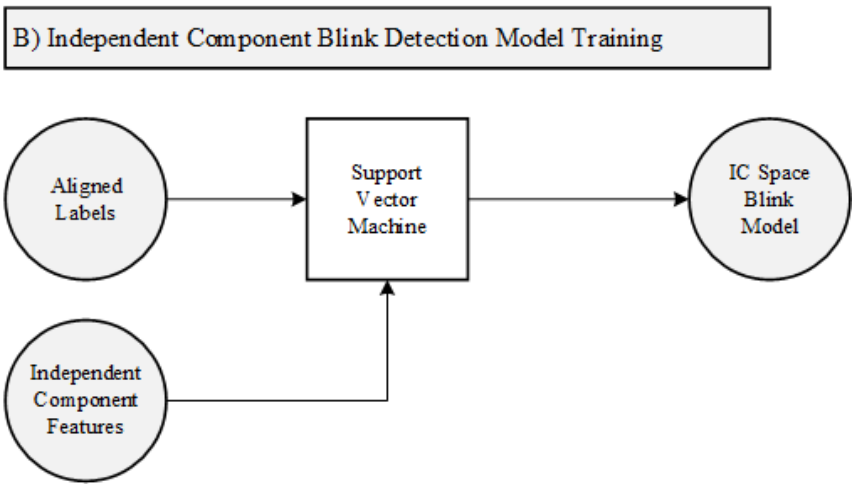

Figure 4: ICA blink detection model training process

The process for extracting features from independent components is the same as extracting features from the raw data (shown in Figure 4). The two cases are considered independently in order to evaluate the accuracy of blink detection on raw data and independent components.

Chapter 2: EOG Regression and Autoregressive Blink Detection

This chapter will cover methods for removing ocular noise that do not involve the large scale feature extraction that chapters 4 and 5 cover. To establish general concepts in ocular noise removal, this section will first cover EOG regression, which is an early method for removing ocular noise. Next, this chapter will cover prior work done with autoregressive blink detection.

2.1: EOG Regression

EOG regression is a method that calculates the correlation of each EEG channel to an EOG channel (VEOG, HEOG, or both). This correlation is used to determine how much the EEG channel is effected by ocular noise found in the EOG. This method was used in (He et al., 2005), it consists of modeling the noisy (raw) EEG signal using the following equation:

$$EEG_{Raw}(t) = EEG_{Pure}(t) + F(EOG(t)) \quad (2.1)$$

Where:

$EEG_{Raw}(t)$ is a single channel of recorded (raw) EEG

$EEG_{Pure}(t)$ is an ideal EEG signal free of ocular noise

EOG(t) is either a vertical or horizontal EOG signal

F(g(t)) is a transfer function that achieves this result.

EOG regression requires a transfer function that can relate the EOG signal to an EEG channel. The simplest class of transfer function for this task is time domain regression, which just requires some constant k_i for each EEG channel i, that satisfies the following equations:

$$k_i = \frac{\sum_t [EOG(t) - \overline{EOG}][EEG_{Raw}(t) - \overline{EEG_{Raw}}]}{\sum_t [EEG_{Raw}(t) - \overline{EEG_{Raw}}]^2} \quad (2.2)$$

Figure 5: EOG Regression Constant k Equation

$$C_i = \overline{EEG_{Raw}} - k_i \overline{EOG} \quad (2.3)$$

Figure 6: EOG Regression Constant C equation

$$EEG_{Pure} = EEG_{Raw}(t) - k_i EOG(t) - C_i \quad (2.4)$$

Where:

k_i , C_i are regression constants for EEG channel i

While EOG Regression is quick to process and easy to implement, there are some issues pointed out in (Croft & Barry, 1998), for example:

1. This method will remove some neural data as well as ocular artifacts, because some useful data will propagate forward towards the EOG; since this method just subtracts the (scaled) EOG signal from the EEG signal, the neural data that becomes present in the EOG data will be removed as well.
2. The calculated regression constants may be inflated because the EOG will contain some EEG signals.
3. The reference signal may also pick up some EOG, causing the values of the regression constants to be distorted as well.

2.1.2: EOG Regression Performance

This section will show the results of using EOG regression on a segment of EEG data.

Overview: 10 Seconds of Results

Chapter 2: EOG Regression and Autoregressive Blink Detection

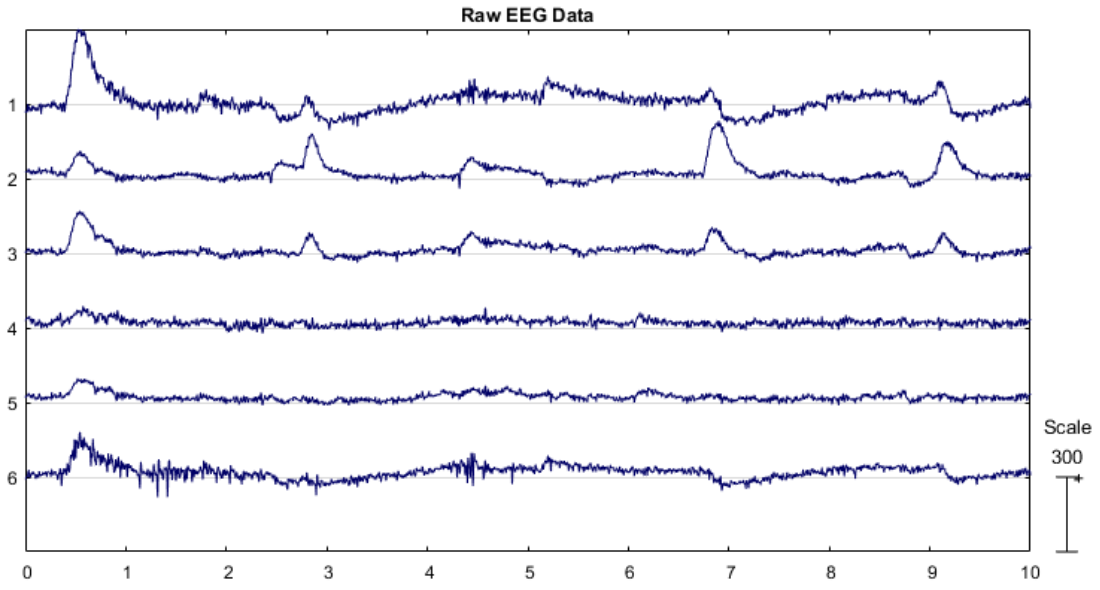

Figure 7: 10 Seconds of Raw EEG Data

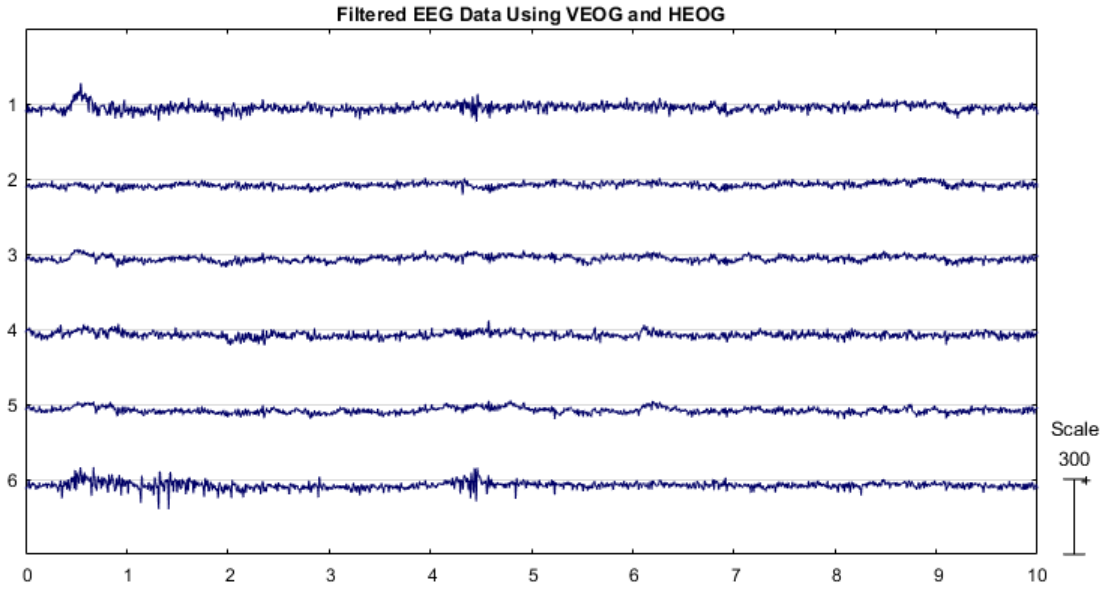

Figure 8: 10 Seconds of Filtered EEG Data using EOG Regression

As can be seen in figures 7 and 8, the EOG regression algorithm has removed lot of the high amplitude noise has been removed in the signal. For example, the blink near the 7 second mark has been reduced to the point where it's not even visible in the plot.

2.1.3: The effect of VEOG, HEOG, and VEOG + HEOG

This section compares the results of EOG regression using only a VEOG signal, only a HEOG, signal, and using both signals. Experiments done as part of this study have shown that using VEOG and HEOG to calculate the regression coefficients give the best results. Using only VEOG or only HEOG actually inserts a significant amount of distortion into the filtered signal.

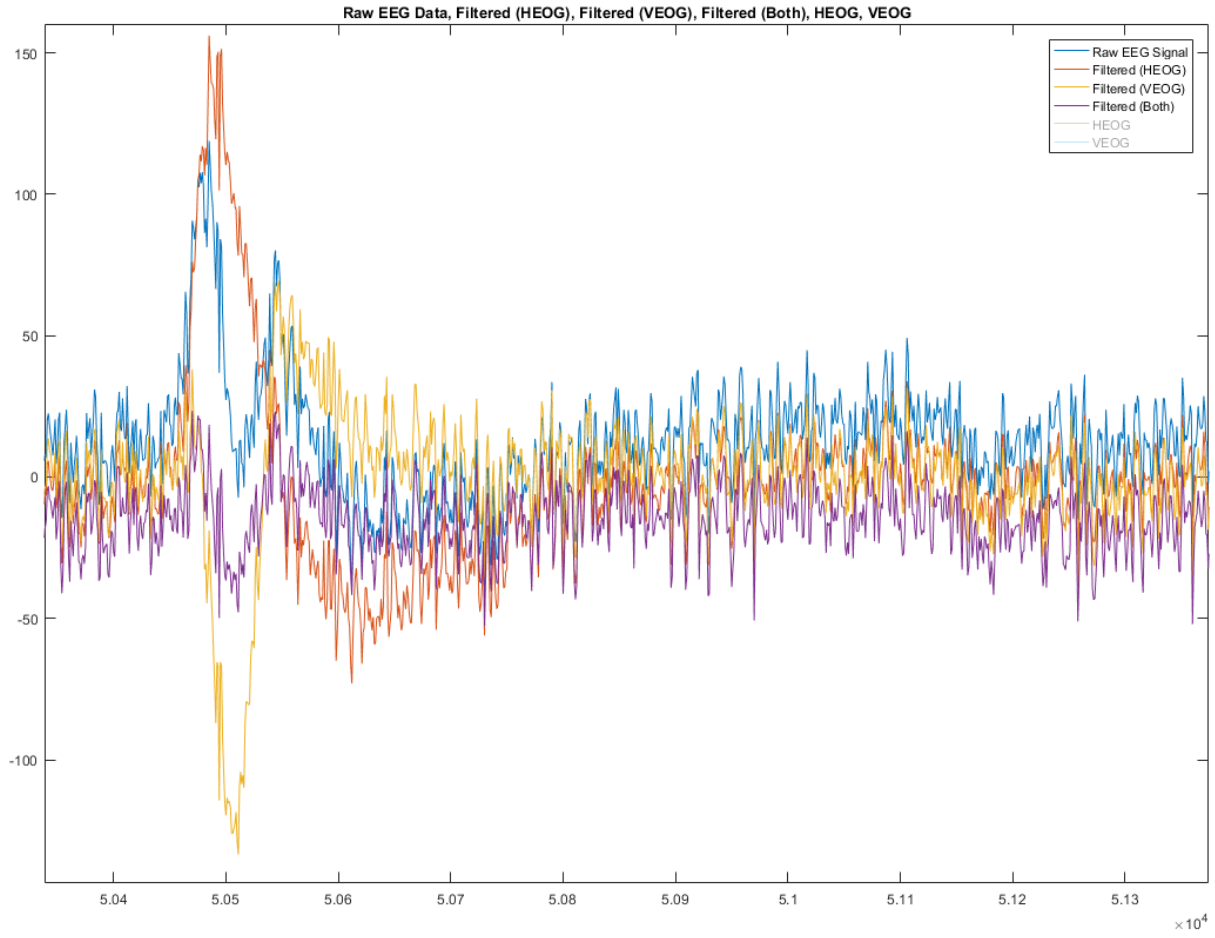

Figure 9: Approximately 1 second of EOG regression results

Figure 9 shows the effect of using various combinations of VEOG and HEOG when filtering using EOG regression. The most important observations are listed below:

- Note how the filter results that used both VEOG and HEOG (purple) have the least amount of high amplitude activity in this sample.
- The filters that used only HEOG (red) or only VEOG (yellow) actually introduced noise into the signal that was not present in the raw data (shown in blue).

- Accuracy was improved when both HEOG and VEOG were considered because the coefficient k for HEOG turned out to be negative.
- This means that a positive spike in VEOG tends to be balanced out by a negative spike in HEOG during a blink.

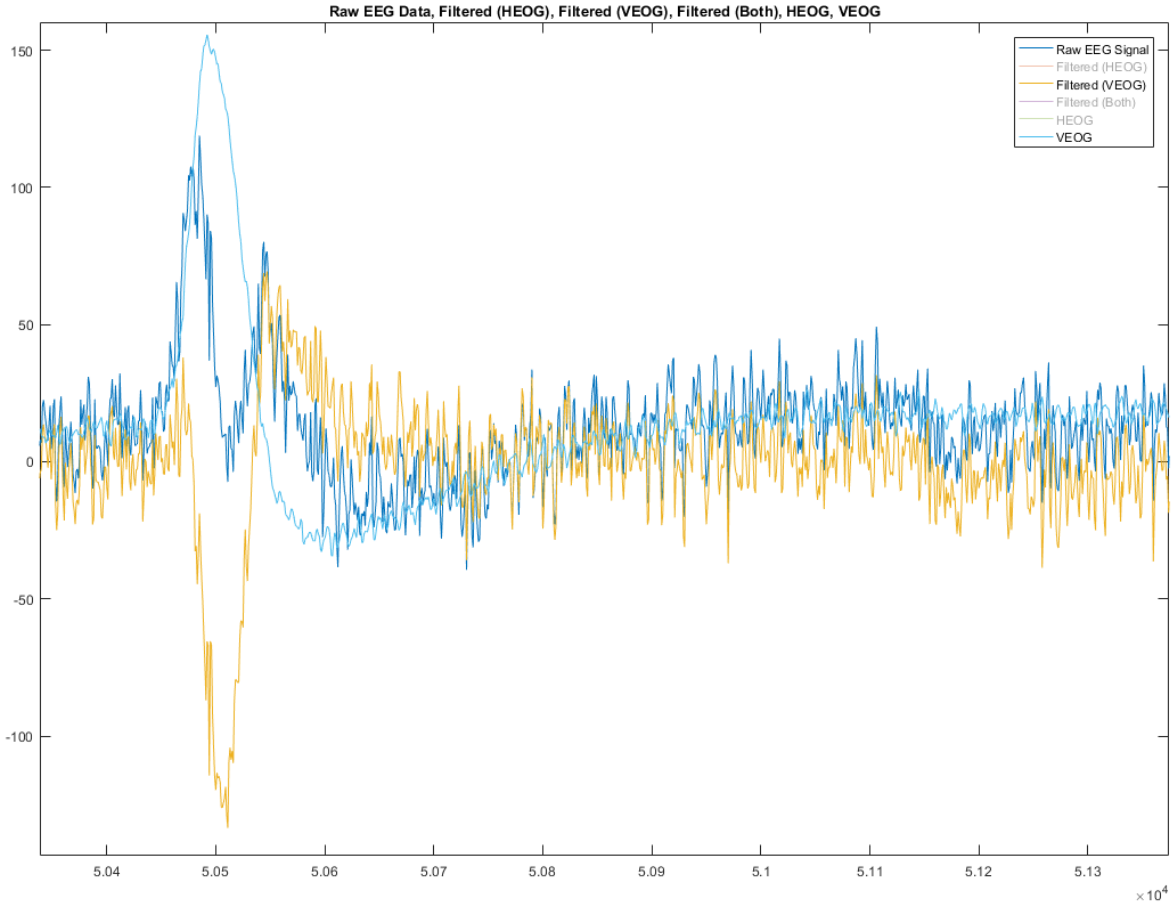

Figure 10: Distortion caused by only using a single EOG channel

Note in Figure 10, when the VEOG is used by itself (yellow curve), it actually introduces noise into the signal, the peak due to blinking is very high and the HEOG signal isn't there to balance it out. Figure 11 and Figure 12 show the VEOG and HEOG data by themselves to emphasize how these two signals balance each other out in the filtering process.

2.1.4: Regression Coefficients

This section shows how the regression coefficients are applied to the VEOG and HEOG signals to filter out ocular noise.

Chapter 2: EOG Regression and Autoregressive Blink Detection

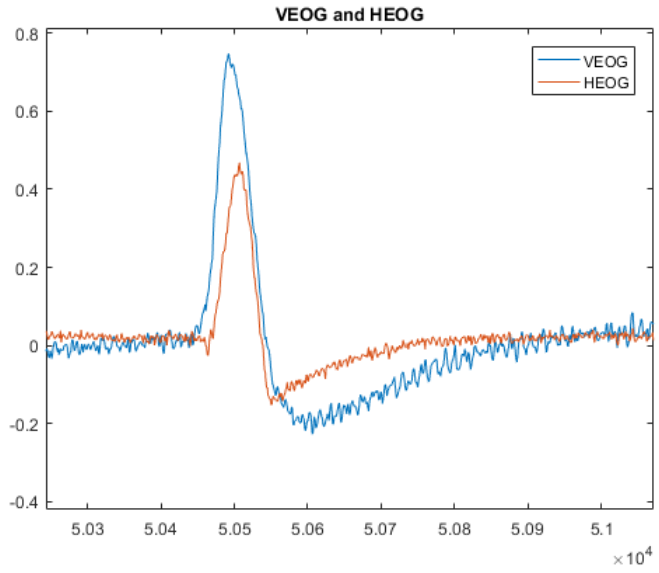

Figure 11: Raw VEOG and HEOG signal

Channel 1 Regression Coefficients:

- $VEOG_k$ =195
- $VEOG_C$ = 9.7
- $HEOG_k$ = -236
- $HEOG_C$ = 19.7

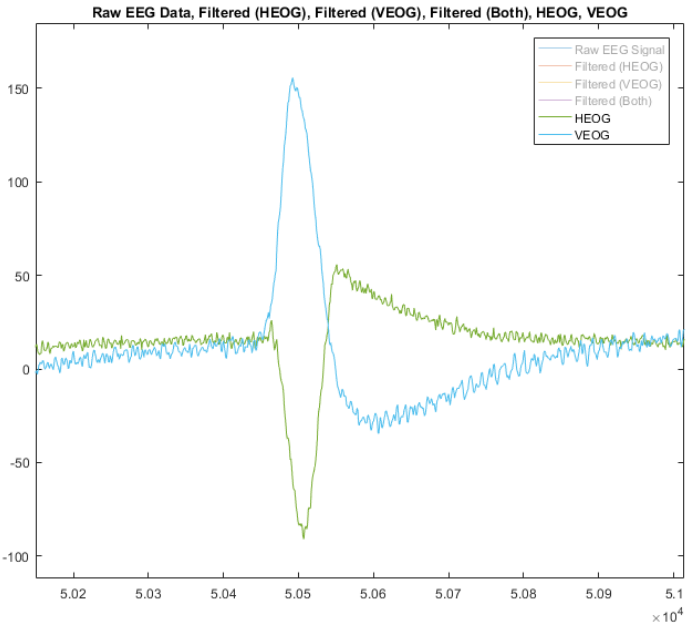

Figure 12: VEOG and HEOG with regression coefficients applied

2.3: Autoregressive Blink Detection

In (Lawhern, Hairston, & Robbins, 2013), an algorithm was developed using autoregressive modeling. Known noise profiles were used to train a SVM model. The training data and testing data consisted entirely of time domain signals, unlike methods discussed later in this section.

Using DETECT to analyze neural data is a two step process, first you provide a segment of data for training purposes with labels specifying what sort of noise is found in each segment. This provides training data for DETECT's internal SVM. The trained autoregressive model can then be used on data to be tested for noise. If DETECT finds a trained noise pattern in the testing data, it will return the time range where the noise occurred, and a certainty value.

2.3.1. Experiment Setup

- In order to test out DETECT's ability to differentiate one type of noise from another, short EEG trials were performed with as little interference as possible.
- The subject was asked to move in a way that has been known to cause significant noise when they heard a beep.
- These recordings were performed on wearable EEG hardware.
- 10 trials were performed for each category of noise.

2.3.2. Results

The results of this experiment varied depending on the complexity of the noise involved. Motions that produced rather indistinct noise such as jaw movement were hard to detect reliably. One out cross validation was used to validate the results. One trial was selected as testing data, then every other trial was used as training data; if the type of noise detected by the model on the testing data was correct, it was counted as a match.

Overall:

- The algorithm was correct for 78 out of 90 trials were correctly identified (86.7% accuracy)

Results by category:

Category 1: Baseline (Control):
9 out of 10 baseline trials were identified correctly

- One baseline trial was incorrectly identified as "jaw movement"

Category 2: Eye Blink:

10 out of 10 eye blink trials were identified correctly

Category 3: Eyes Closed:

8 out of 10 eyes closed trials were identified correctly

- One eyes closed trial was incorrectly identified as “head downward movement”
- One eyes closed trial was incorrectly identified as “jaw movement”

Category 4: Eyebrow Up:

9 out of 10 eyebrow up trials were identified correctly

- One eyebrow up trial was incorrectly identified as “eyes closed”

Category 5: Eyeball Movement:

9 out of 10 eyeball movement trials were identified correctly

- One eyeball movement trial was incorrectly identified as “baseline”

Category 6: Head Rotation:

9 out of 10 head rotation trials were identified correctly

- One head rotation trial was incorrectly identified as “head movement down”

Category 7: Head Movement Up:

8 out of 10 trials were identified correctly

- Two head movement up trials were incorrectly identified as “jaw movement”

Category 8: Head Movement Down:

8 out of 10 trials were identified correctly

- Two head movement down trials were incorrectly identified as “head movement up”

Category 9: Jaw Movement:

8 out of 10 trials were identified correctly

- One jaw movement trial was incorrectly identified as “eyes closed”
- One jaw movement trial was incorrectly identified as “head down”

2.3.3. Analysis of results

- The most common incorrect identification was jaw movement
    - This seems to indicate that the training samples for jaw movement were not distinct enough, or that some samples had some unintentional jaw movement included.
    - It's very likely that the head movement up trials included some unintentional jaw movement, just by the nature of how the movement was performed.

- The head movement trials had incorrect identifications, some of which were just movement in a different direction.
    - One head rotation trial was incorrectly identified as “Head Down”. It's possible that there was a small amount of rotation involved in the head down movement.
    - Two head down movements were incorrectly identified as “Head Movement Up”. Interestingly, no head movement up trials were identified as “Head movement down”

- DETECT is a useful algorithm when all noise patterns are known and distinct
    - The algorithm requires training data for each class, including a baseline signal. For best results, each type of noise should have a clear, consistent shape.
    - Time domain autoregression has been known to be sensitive to noise, a more robust feature set would improve the accuracy of the noise detection system.

2.4: Conclusion

Blink and saccade removal using EOG regression is a very fast operation to preform, and can be done in real time, however studies have found that propagation of neural activity into the EOG signal is a significant problem, especially for neural activity happening near the front of the brain. Also, since the regression coefficients are only calculated once, this setup is a rough estimation of the relationship of (non-stationary) EOG and EEG signals.

Chapter 2: EOG Regression and Autoregressive Blink Detection

Autoregressive blink detection is good for short, isolated trials where blinks are distinct and repeatable, and the noise patterns are known ahead of time and do not vary significantly in their rate. This method is also useful for other types of movement noise such as certain kinds of head movement, but as shown in the results, when movements become more complicated its accuracy is diminished.

Chapter 3: Label Generation using Dynamic Time Warping

Dynamic Time Warping (or DTW) is an established algorithm that measures the similarity between two sequences of sample values. A major advantage of DTW is that these sequences can vary in speed. This means that a signal that is "stretched" or "shrunk" in the time domain can still be found based on its similarity to the unstretched signal.

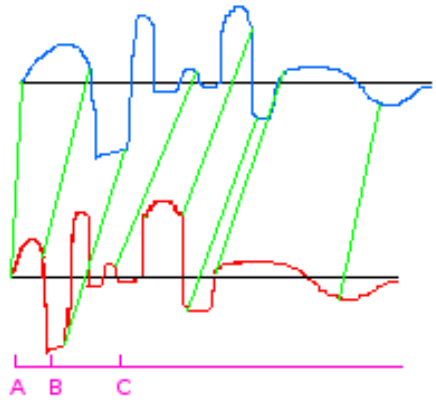

Figure 13: A Graphical representation of DTW finding the shortest distances between graphs

Figure 13 illustrates how DTW would be able to relate the signal in red to the signal in blue. Even though the signals progress at three different rates (A: Compressed, B: Less compressed, and C: Stretched), dynamic time warping can still calculate how similar these signals are.

The DTW algorithm has two parts to it, it is O(mn) in complexity:

$$C(x,y) = \sum_{k=1}^{K} c(M_{m_k}, N_{n_k}) \tag{3.1}$$

$$P_{k-1} = \begin{cases} C(1, \quad n-1) \; if \; m = 1 \\ C(m-1, \quad 1) \; if \; n = 1 \\ \quad otherwise \\ \min \begin{cases} C(m-1, n-1) \\ C(m, \quad n-1) \end{cases} \end{cases} \tag{3.2}$$

Where:

- M is a time series (e.g., a blink template).
- N is a time series (e.g., one 250 sample window of data).
- m is the length of time series M.
- n is the length of time series n.
- m_k is a point in time series M, path index k
- n_k is a point in time series N, path index k
- K is equal to n * m.
- C is a cost matrix, each entry in C(x,y) is the Euclidean distance between two points m_k and n_k
- P_k is the smallest distance required to connect point m_k to n_k.

This algorithm traverses the cost matrix C to find the minimum set of distances P between time series M and N.

3.1: Template Selection

To label windows of electrooculogram (EOG) activity as blink or non-blink, it is necessary to provide the dynamic time warping algorithm with some templates of known blink activity. For this study four windows containing blinks were selected as templates. Of the four templates selected, only one (the huge blink template) was used for labeling purposes. The reason why the other three were removed was because they had a high false positive rate, as will be discussed in the following paragraphs. The four blink samples used as templates for DTW are shown in Figure 14.

Chapter 3: Label Generation using Dynamic Time Warping

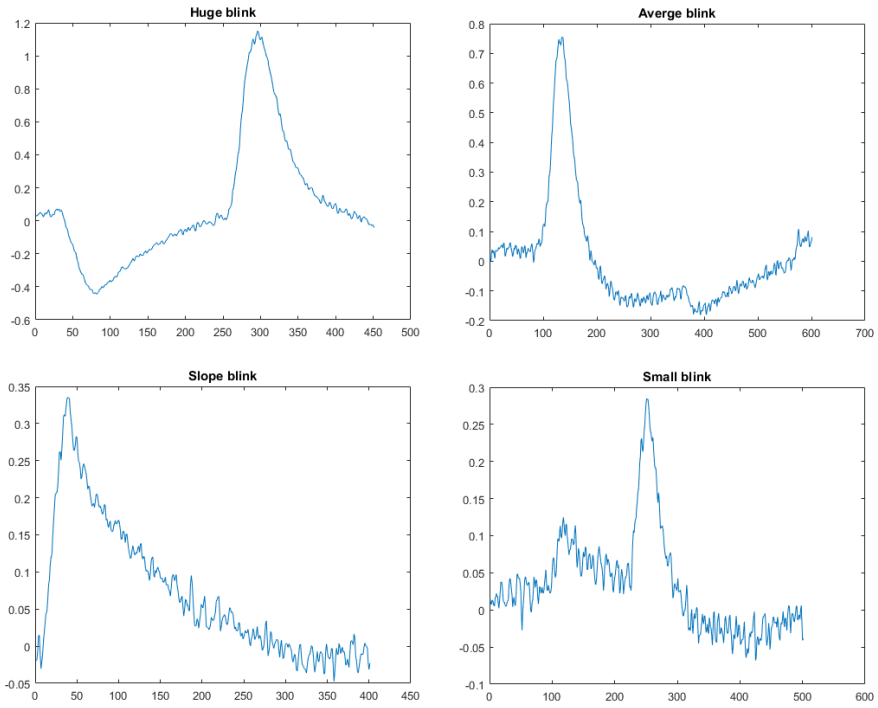

Figure 14: Blink Pattern

3.2: DTW Performance

This section will show the results of the DTW algorithm on 250 sample windows of VEOG (vertical electrooculogram) data. The DTW algorithm processes each window and gives a distance metric for how different a given window is from each blink template.

3.2.1: Performance on partial blinks

Figure 15 shows a portion of EOG data that demonstrates the performance of DTW on windows containing only part of a blink. Note how the “huge blink” template distance (in orange) is below the proposed threshold (in purple) for windows 22 through 28. The DTW algorithm has issues finding partial blinks, for example the distance in window 30 is very high. While this issue does make the labels less accurate for blink detection, it is worth noting that an expert manually labelling these segments of time would also need to make a judgement call on how much of a partial blink should cause a window to be labelled as a blink.

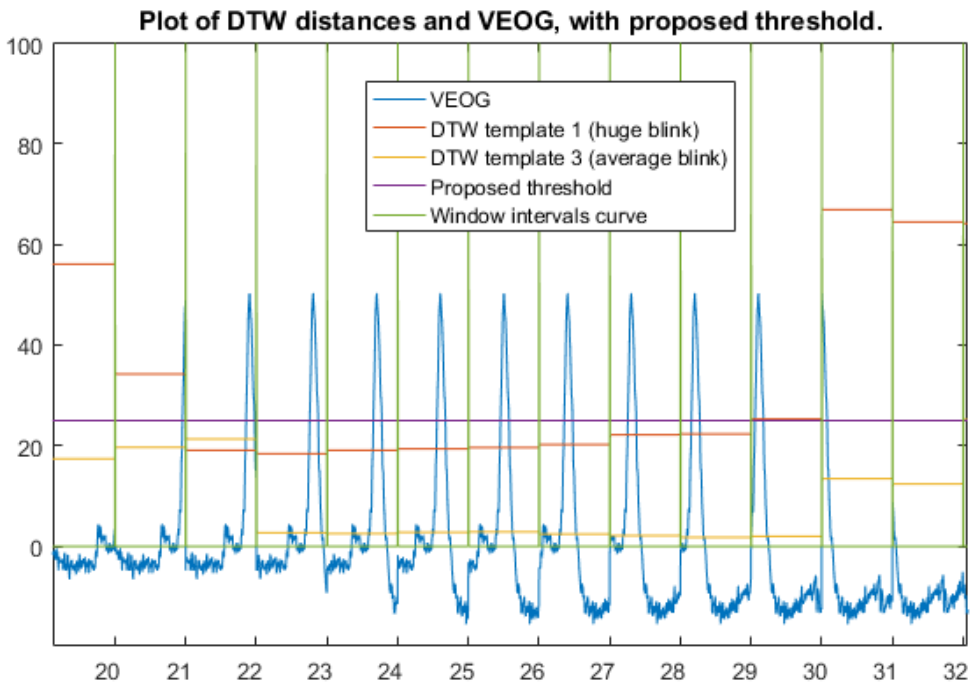

Figure 15: DTW distances of all four templates on a sample blink

3.2.2: Performance during vertical saccade

In addition to blinks, the VEOG signal also contains noise due to vertical saccade. This is typically caused by the subject moving their eyes up and down. This system's "huge blink" template contains a short saccade followed by a blink. This template is good for identifying many of the high amplitude blink/saccade combinations present in the data, but it is not an ideal sample for cases where the saccade lasts far longer than one or two windows. Figure 16 shows a sample of EOG data that demonstrates this.

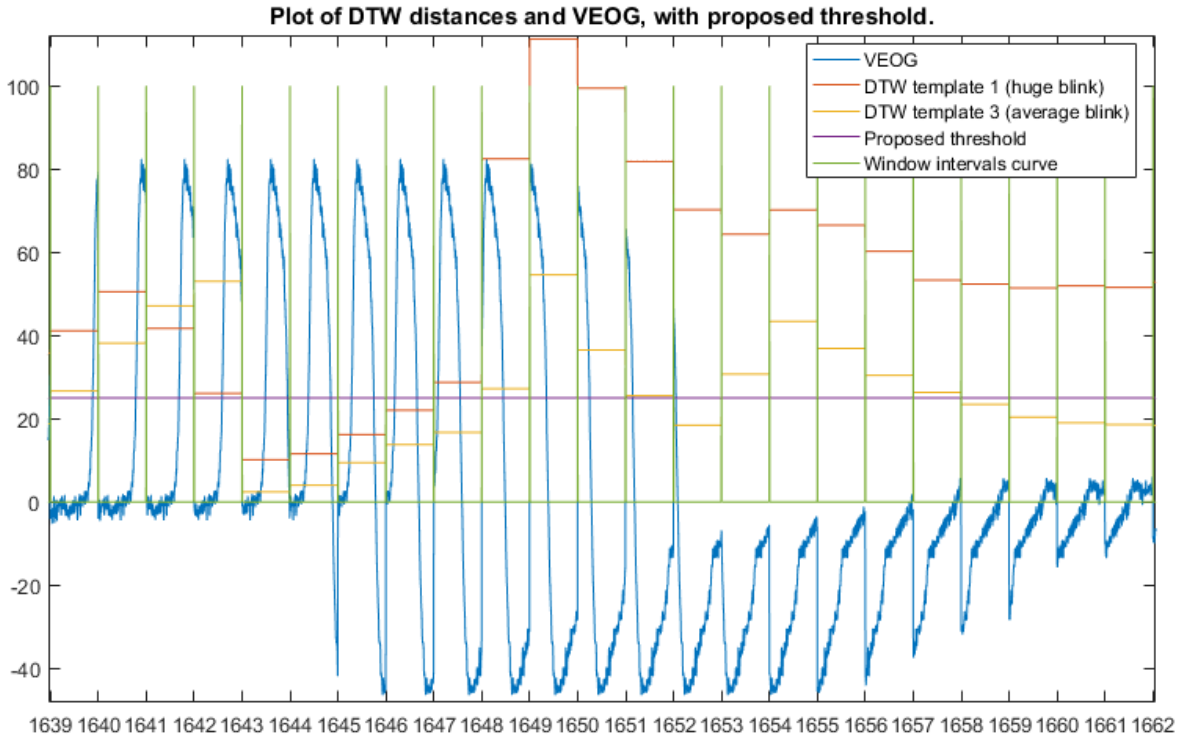

Figure 16: DTW performance during saccade

Note how the distance is very large in window 1649, and DTW also gives high distances for 1650-1659. In this sample only windows 1643-1646 would be labeled as blinks, the others would be considered non-blink activity. A DTW implementation capable of labeling saccade activity as well as blink activity is a task for future work. It would likely require larger window sizes because saccades have a large amount of variation in size and shape. A detailed study consisting of common saccade profiles would be necessary.

3.2.3: Performance during line noise

There are several portions of data in this study that contain severe line noise. The segment shown in Figure 17 below illustrates how well DTW performs under noisy conditions. The distance is much lower than normal for non-blink windows 300-309, nearing the threshold line. The distance goes down to expected levels during blink windows 310-318. As mentioned before, partial blink windows are difficult to label, and windows 319 and 320 are labeled as non-blink.

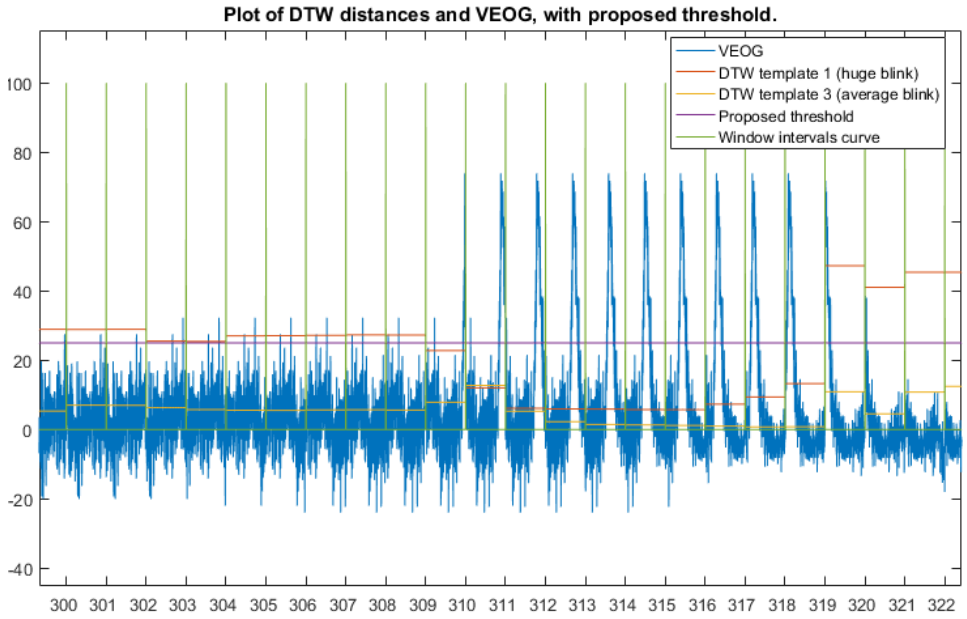

Figure 17: Performance of DTW under noisy conditions

3.2.4: Performance during small blinks

The set of windows shown in Figure 18 contains a relatively small blink that the DTW algorithm was not able to label properly, at no point during the blink's duration was the 'huge blink" template useful for identifying this blink. The "average blink" template shows decent contrast for this area, but the issue with using smaller blink templates for label generation is that these templates tend to get confused with noise. Finding the best set of templates to label a variety of blink shapes and sizes is a good topic for future work.

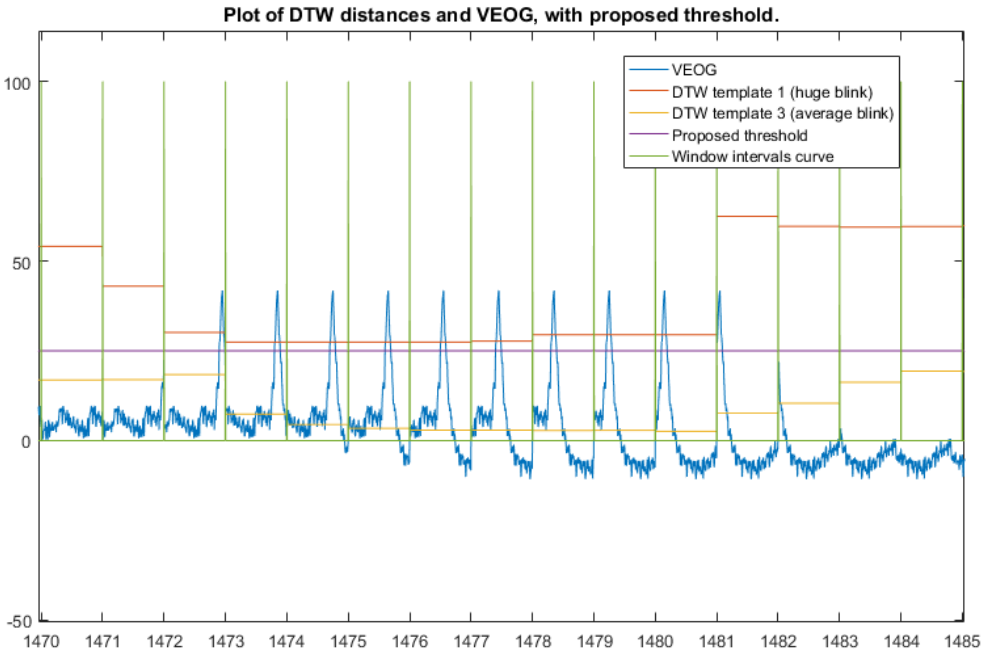

Figure 18: DTW performance on small blinks

3.3: Conclusion

DTW is an effective algorithm for providing labels for most kinds of blinks present in VEOG data. The current implementation struggles with cases where there is only a partial blink in the window, or where there is saccade and blink activity combined in a single window, but this is something that even a manual reviewer may have trouble with. The results could be improved by adding more templates for saccades and various shapes of blinks, this would allow the algorithm to more accurately label the VEOG signal which naturally includes several varied shapes of blinks, line noise, and saccade. Another good topic for future work is to integrate the HEOG into data annotation.

Chapter 4: Time Space Blink Detection

This section discusses the types of features that were extracted for this study, it then briefly covers how the detection algorithm works and its results, then it discusses the application of t-test to determine the most relevant features.

4.1: Feature Selection and Extraction

Having a large variety of features available for blink detection is a crucial part of this system. The effect adding features has on the performance of a blink detection system is depicted in Figure 19.

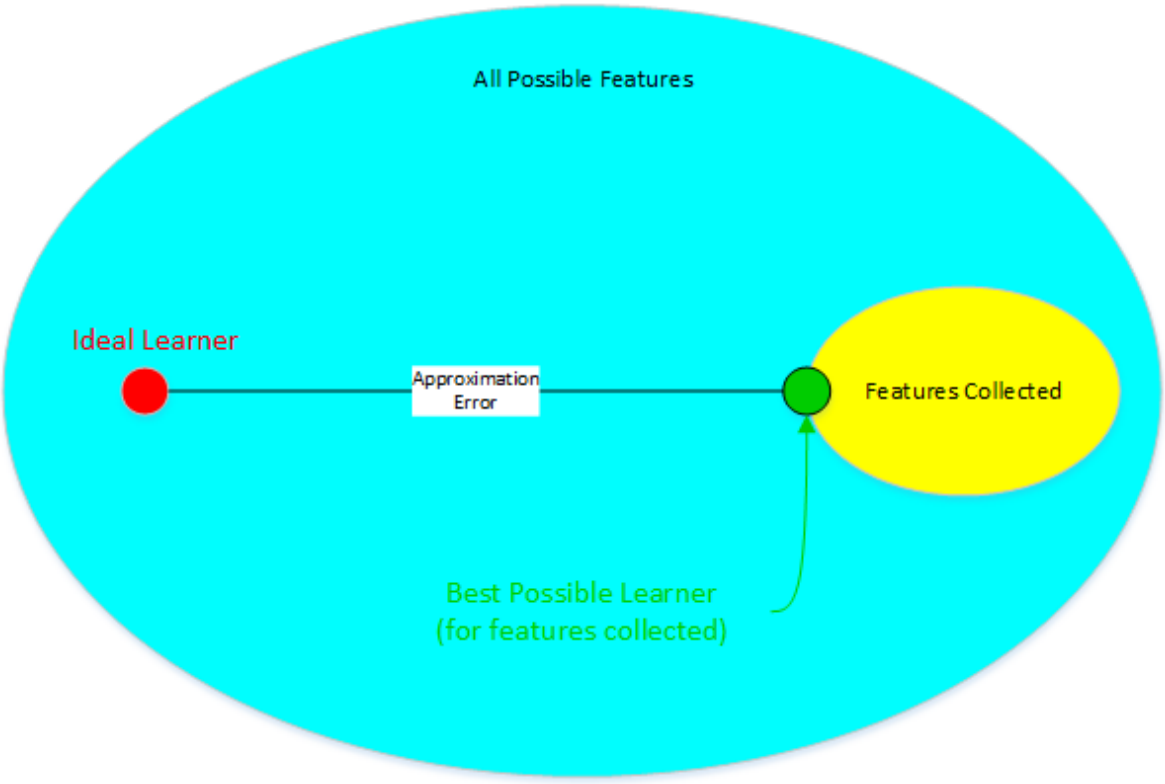

Figure 19: Possible learners with a small amount of features

With a small number of features, the dimensionality of the feature space is small and the best possible learner is far from the ideal learner. Approximation error with a small number of features is large because there are not enough dimensions to get close to the ideal learner.

As features are added, the best possible learner using those features gets closer to the ideal learner, as shown in Figure 20.

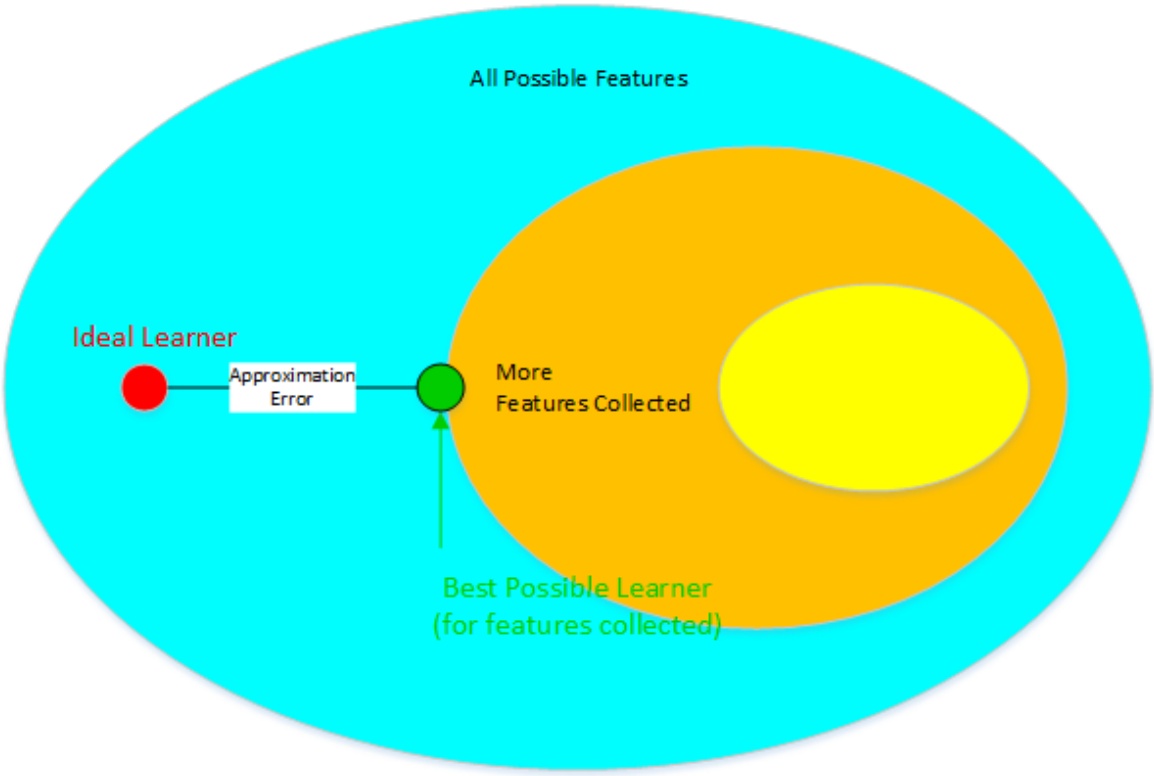

Figure 20: Possible learners when the number of features increases

MATS (Kugiumtzis & Tsimpiris, 2010) provided a proven set of features that are implemented specifically for use in EEG data mining operations. Using features proven for electroencephalogram research provides a strong baseline for what features are expected to be helpful for blink detection, however as the hypothesis class is enlarged, the problem of finding the best possible learner becomes more difficult to solve. Finding the ideal learner is an np-hard problem, because the number of possible learners is equal to the size of the universe of discourse (i.e., every possible combination of values for each feature). If the dimensionality of the features becomes too large to be covered by the sample data, overfitting will occur, and adding features after that will increase the error of the classifier rather than decreasing it.

Several techniques exist for reducing the dimensionality of a feature space. This study investigated the feasibility of LDA (Linear Discriminant Analysis) to project the collected feature set to a lower dimensional space while still maintaining separation between blink and non-blink classes. LDA was

found to not provide usable results on its own in this system. This is because the universe of discourse of the feature set is too large for the amount of sample data considered. Not enough sample points were provided to make the LDA matrix invertible. This situation should be improved by combining LDA with PCA, which is a good subject for future work.

4.1.2: Features Listing by Category

Statistical:

- Pearson autocorrelation
- Cumulative Pearson autocorrelation
- Kendall autocorrelation
- Spearman autocorrelation
- Cumulative Spearman autocorrelation
- Partial autocorrelation
- Correlation dimension
- Arithmetic mean
- Median
- Variance
- Standard deviation
- Interquartile range
- Skewness

Temporal:

- Autoregressive fit: MAPE, NMSE, NRMSE, and CC error
- Autoregressive prediction: MAPE, NMSE, NRMSE, and CC error

Information Theory:

- Mutual information
- Cumulative mutual information
- Mutual information: Time to first minimum

Momentum:

- Kurtosis
- Hjorth mobility

- Hjorth complexity

Spatial / Geometric:

- Hurst exponent
- Detrended fluctuation analysis
- Mean of local maxima
- Median of local maxima
- Standard deviation of local maxima
- Interquartile range of local maxima
- Mean of local minima
- Median of local minima
- Standard deviation of local minima
- Interquartile range of local minima
- Mean time between local minimums
- Median time between local minimums
- Standard deviation of time between local minimums
- Interquartile range of time between local minimums
- Mean amplitude between local minimum and next local maximum
- Median amplitude between local minimum and next local maximum
- Standard deviation of amplitude between local minimum and next local maximum
- Interquartile range of amplitude between local minimum and next local maximum

Spectral:

- Alpha band power
- Beta band power
- Delta band power
- Theta band power
- Wavelet Aj mean
- Wavelet Dj mean
- Wavelet Dj1 mean
- Wavelet Dj2 mean
- Wavelet Aj band power
- Wavelet Dj band power

- Wavelet Dj1 band power
- Wavelet Dj2 band power
- Wavelet Aj standard deviation
- Wavelet Dj standard deviation
- Wavelet Dj1 standard deviation
- Wavelet Dj2 standard deviation
- SFT: Delta band
- SFT: Theta band
- SFT: Alpha band
- SFT: Beta band

4.2: Soft-Margin Support Vector Machine (SVM) Classification

Blink or non blink classification was conducted using a linear Support Vector Machine (SVM). The SVM approach has been used in many different applications (Cortes & Vapnik, 1995). Soft margin SVM is a commonly used method to solve 2 class discrimination problems, the details of this algorithm are shown below:

$$\min\left(\frac{1}{2}\right)||w||^2 + C\sum_i \beta_i \tag{4.1}$$

$$y_i(w \cdot x_i + b) \geq \left(1 - \beta_i\right), \forall x_i$$

$$\beta_i \geq 0$$

Soft margin SVM seeks to minimize β_i (the sum of distances from the hyperplane) while maximizing the margin between classes. Figure 21 shows how β_i is minimized and the effect of outliers on this type of classifier. The constant C can be set to adjust the tradeoff factor between margin width and misclassifications. A simple grid parameter search is performed to find the best values for C and β_i.

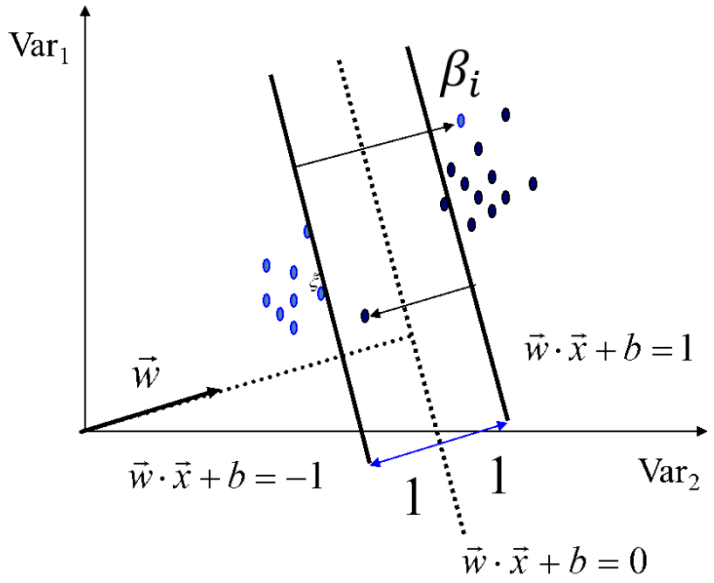

Figure 21: Soft margin SVM hyperplane, margin, and distances to samples

Soft-Margin SVM is always able to generate a hyperplane even if the problem is not linearly separable, it is also more resilient to outliners, because they are included in the objective function. Note how in Figure 22 the margin of the trained hyperplane is larger in the soft margin SVM, this improves the generalization ability of the hyperplane.

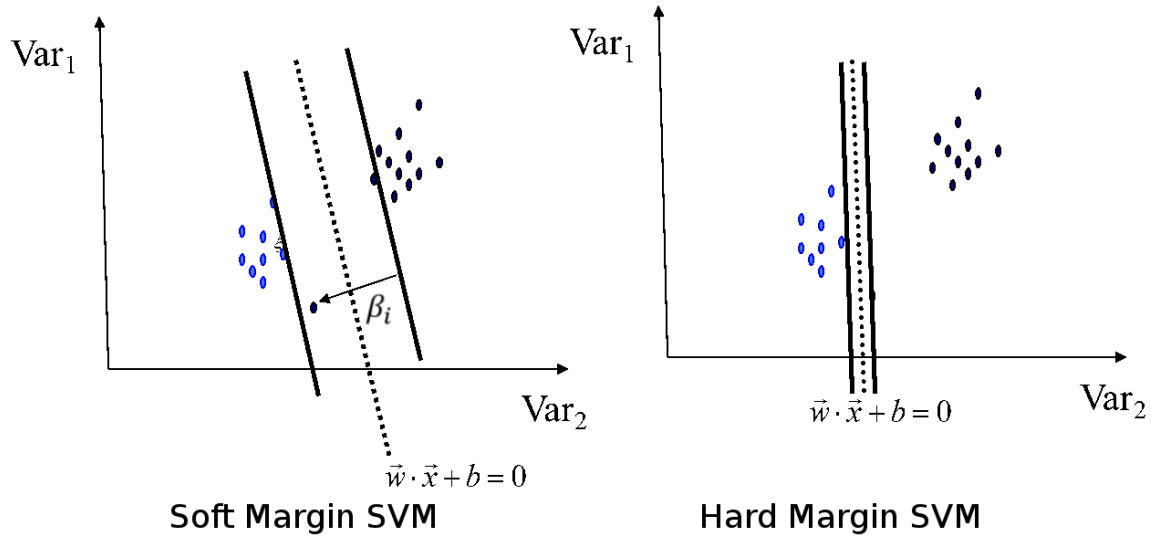

Figure 22: A graphical comparison of soft and hard margin SVM

4.3: Experimental Setup (Workload Assessment Task)

In the trials that make up the data for this study, the subject was instructed to track either one or two objects on a screen. After each experiment, the subject was asked to give their opinion on how strenuous the task was. These trials varied in length.

4.3.1: Recording Hardware

- Electroencephalogram and electrooculogram data was collected at 500Hz on a BioRadio wet EEG (Great Lakes NeuroTechnologies, Valley View, OH)
- The workload assessment task data collection consisted of seven EEG channels was well as a horizontal and vertical electrooculogram (HEOG and VEOG).
- Six out of seven EEG channels were used in the algorithms presented in this paper.

Channel Locations

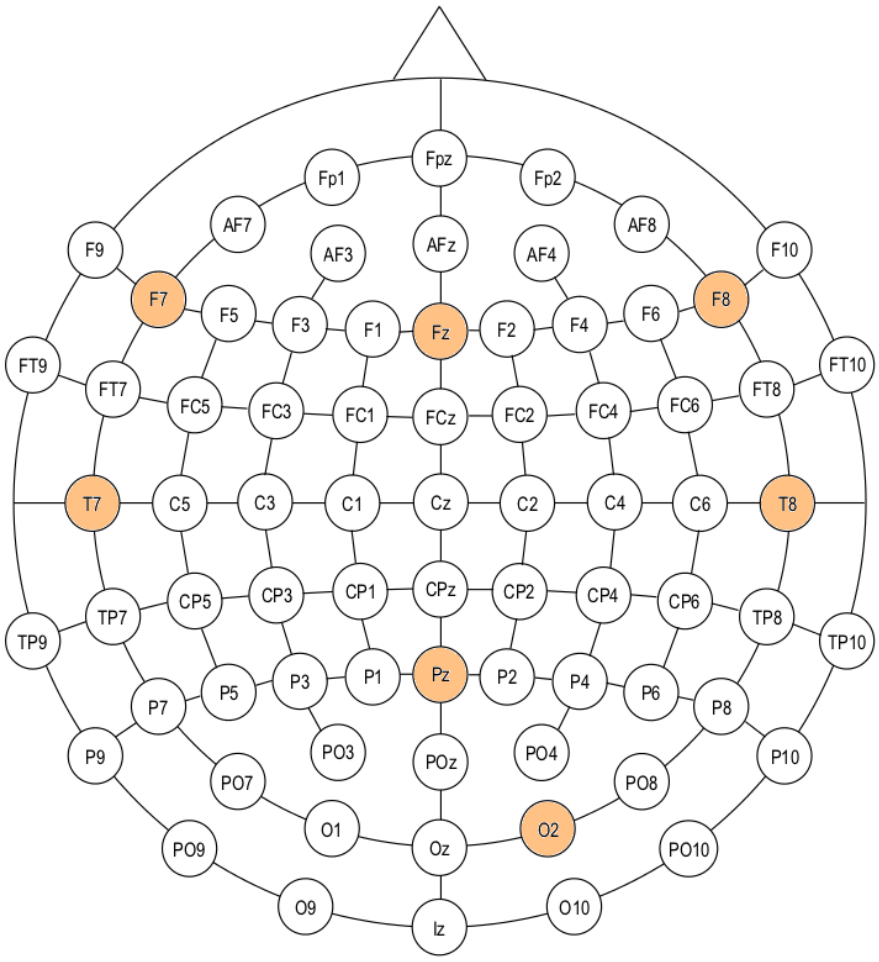

Figure 23: Workload Assessment Task Channel Locations

4.4: Detection and Feature Ranking Methods

Each trial's EEG and EOG data was split into 250 sample (.5 second) windows. The longest trial contained 8,235 windows, the shortest trial contained 5,896 windows. The EEG and EOG windows are synchronized and equal length in order to make the labelling and detection process as accurate as possible.

4.4.1: Time Space Blink Detection Process Overview

1. Divide up both the EEG and EOG data into synchronized 250 sample windows, with 90% overlap
2. Use Dynamic Time Warping (DTW) to obtain distances from each EOG window to four pre-defined blink windows (see Chapter 3)
3. Extract a diverse set of features from each EEG window
4. Train a Support Vector Machine, using the extracted feature set and the DTW distances as labels
5. Verify the detection results using 10cv cross validation.

10cv cross validation method was based on a grid optimization technique by (Chang, Chih-Chung, & Lin, 2011).

4.4.1: Features Listing (by index)

The following features were extracted from each window of EEG data.

Table 1: Features listing by index

1	'Pearson Autocorrelation'
2	'Cumulative Pearson Autocorrelation'
3	'Kendall Autocorrelation'
4	'Spearman Autocorrelation'
5	'Cumulative Spearman Autocorrelation'
6	'Partial AutoCorrelation'
7	'Autoregressive fit: MAPE Error'
8	'Autoregressive fit: NMSE Error'
9	'Autoregressive fit: NRMSE Error'
10	'Autoregressive fit: CC Error'
11	'Autoregressive Prediction: MAPE Error'
12	'Autoregressive Prediction: NMSE Error'
13	'Autoregressive Prediction: NRMSE Error'
14	'Autoregressive Prediction: CC Error'

15	'Mutual Information'
16	'Cumulative Mutual Information'
17	'Mutual Information: Time to first minimum'
18	'Correlation Dimension'
19	'Arithmetic Mean'
20	'Median'
21	'Variance'
22	'Standard Deviation'
23	'Interquartile Range'
24	'Skewness'
25	'Kurtosis'
26	'Hjorth Mobility'
27	'Hjorth Complexity'
28	'Hurst Exponent'
29	'Detrended Fluctuation Analysis'
30	'Mean of local maxima'
31	'Median of local maxima'
32	'Standard deviation of local maxima'
33	'Interquartile range of local maxima'
34	'Mean of local minima'
35	'Median of local minima'
36	'Standard deviation of local minima'
37	'Interquartile range of local minima'
38	'Mean time between local minimums'
39	'Median time between local minimums'
40	'Standard deviation of time between local minimums'
41	'Interquartile range of time between local minimums'
42	'Mean amplitude between local min and next local max'
43	'Median amplitude between local min and next local max'
44	'Standard deviation of amplitude between local min and next local max'
45	'Interquartile range of amplitude between local min and next local max'
46	'Alpha band power'
47	'Beta band power'

48	'Delta band power'
49	'Theta band power'
50	'Wavelet: Aj Mean'
51	'Wavelet: Dj Mean'
52	'Wavelet: Dj1 Mean'
53	'Wavelet: Dj2 Mean'
54	'Wavelet: Aj Band Power'
55	'Wavelet: Dj Band Power'
56	'Wavelet: Dj1 Band Power'
57	'Wavelet: Dj2 Band Power'
58	'Wavelet: Aj Standard Deviation'
59	'Wavelet: Dj Standard Deviation'
60	'Wavelet: Dj1 Standard Deviation'
61	'Wavelet: Dj2 Standard Deviation'
62	'SFT Feature: Delta'
63	'SFT Feature: Theta'
64	'SFT Feature: Alpha'
65	'SFT Feature: Beta'

4.4.3: SVM Performance

Table 2: Blink Detection Accuracy by Channel

Blink Detection Accuracy by Channel		
Ch#	Highest Accuracy	Lowest Accuracy
1: F7	96.1015	91.8965
2: F8	**98.2727**	**94.7177**
3: Fz	**98.2191**	**94.3048**
4: O2	92.6056	86.4558
5: Pz	93.5255	87.7223
6: T7	94.1982	90.0034

The table above shows each channel's independent ability to detect a blink. As expected the frontal channels F7, F8, and Fz show the highest accuracy, because blink activity is most noticeable near the front of the scalp. F8 (Right Front) and Fz (Center Front) have the highest accuracy (94-98%) of the frontal electrodes, interestingly F7 (Left Front) has slightly lower accuracy (91-96%) with this feature set.

The other channels' accuracy is listed here for reference but based on t-test results (see below), channels 1-3 (especially channels 2 and 3) are far more important for blink classification.

4.4.4: Feature Ranking: t-test

This study also ranked the features collected to rate them based on their relevance to blink activity. The features in this study were ranked twice, one ranking found the best features for each channel, and one ranking found the best features using all channels. Each ranking was performed using a two sample t-test.

For each feature, with blink values vector x and non-blink values vector y:

$$t = \frac{\bar{x} - \bar{y}}{\sqrt{\frac{{s_x}^2}{n} + \frac{{s_y}^2}{m}}} \tag{4.2}$$

Where:

- $\bar{x}$ is the mean of this feature's values for non-blink windows.
- $\bar{y}$ is the mean of this feature's values for blink windows.
- n is the number of windows labeled as blinks by DTW.
- m is the number of windows labeled as non-blink.
- s_x is the standard deviation of this feature's values for non-blink windows.
- s_y is the standard deviation of this feature's values for blink windows.

This calculation was performed using MATLAB's ttest2 function.

This t value is used to calculate the p-value of each feature, which represents how relevant each feature is to determining whether a window is a blink or non-blink.

4.4.5: t-test Results: Overall Best Features

As mentioned in the previous section, the best features tend to come from channels 2 and 3. One feature from channel 1 made it onto this list.

Table 3: Time Space: Top 20 Best Features

All Channels: Best Features		
#	Ch	Feature Name
1	2	'SFT Feature: Theta'

2	3	'SFT Feature: Theta'
3	3	'Wavelet: Dj1 Band Power'
4	2	'Wavelet: Dj1 Standard Deviation'
5	2	'Wavelet: Dj1 Band Power'
6	2	'Wavelet: Dj1 Mean'
7	3	'Wavelet: Dj2 Band Power'
8	3	'Wavelet: Dj Band Power'
9	3	'SFT Feature: Delta'
10	3	'Wavelet: Dj Standard Deviation'
11	3	'Wavelet: Dj1 Standard Deviation'
12	2	'Wavelet: Dj2 Standard Deviation'
13	3	'Wavelet: Dj Mean'
14	3	'Wavelet: Dj1 Mean'
15	2	'Wavelet: Dj Standard Deviation'
16	3	'Wavelet: Aj Band Power'
17	2	'Wavelet: Dj2 Mean'
18	2	'Wavelet: Dj Mean'
19	1	'SFT Feature: Theta'
20	3	'Standard deviation of local maxima'

4.4.6: t-test Results: Best Features for each Channel (Independently)

For reference, the most useful features for blink detection on each channel are listed below.

Table 4: Time Domain: Channel 1 Best Features

Channel 1 Best Features	
#	Feature Name
1	'Wavelet: Dj1 Mean'
2	'Wavelet: Dj1 Standard Deviation'
3	Wavelet: Dj2 Mean'
4	'SFT Feature: Theta'
5	'Wavelet: Dj2 Standard Deviation'

Table 5: Time Domain: Channel 2 Best Features

Channel 2 Best Features	
#	Feature Name
1	'Wavelet: Dj1 Mean'
2	'Wavelet: Dj1 Standard Deviation'
3	'SFT Feature: Theta'
4	'Wavelet: Dj2 Standard Deviation'
5	'Hjorth Complexity'

Table 6: Time Domain: Channel 3 Best Features

Channel 3 Best Features	
#	Feature Name
1	'Hjorth Complexity'
2	'Wavelet: Dj Mean'
3	'Wavelet: Dj1 Mean'
4	'Wavelet: Dj2 Mean'
5	'Wavelet: Dj1 Band Power'

Table 7: Time Domain: Channel 4 Best Features

Channel 4 Best Features	
#	Feature Name
1	'Wavelet: Dj Band Power'
2	'Wavelet: Dj Mean'
3	'Autoregressive Prediction: NRMSE Error'
4	'Detrended Fluctuation Analysis'
5	'Autoregressive Prediction: NMSE Error'

Table 8: Time Domain: Channel 5 Best Features

Channel 5 Best Features	
#	Feature Name
1	'SFT Feature: Theta'

2	'Wavelet: Aj Band Power'
3	'Wavelet: Dj Mean'
4	'Wavelet: Dj Standard Deviation'
5	'Wavelet: Dj1 Standard Deviation'

4.5: Conclusion

Using the features extracted from this data, the system described in this section was able to train an accurate blink detection system. Even though the labels generated by the DTW algorithm in Chapter 3 could use some improvement when a window contains a partial blink, or when a blink is combined with a saccade, the system was still able to achieve very competitive results. T-test results showed that the most relevant channels for blink detection were F8 (Right Front) and Fz (Center Front). With either of these two channels, is possible to create a blink detection model that can identify blinks with 94 to 98% accuracy. Higher accuracy should be achievable when channels 1 through 3 are considered together, this is a good subject for future work.

Chapter 5: Independent Component Space Blink Detection

Independent Component Analysis (ICA) is an algorithm that tries to separate a signal with n channels into n statistically independent components (Hyvärinen & Oja, 2000).The process of separating out independent components is non-deterministic, and requires a significant amount of processing time, but it has been proven capable of splitting up even a very complex signal into accurate components. For example, if two radios were placed nearby each other, and the recording was done with two microphones, ICA could separate out two independent components with each IC being a clear signal from each radio, without any interference from the other.

There have been many projects using ICA for electroencephalogram research. Most of these involved rather short sessions with simple tasks, where the subject generally sat still and stared at a screen or responded to some other external stimuli. In (Makeig, J. Bell., Jung, & Sejnowski, 1996), the subject kept their eyes closed and were told to press a button when they heard a white noise signal increase in volume. The results of this simple test were promising, there was a clear separation of theta and alpha bands, which helped identify hit and miss responses when the subject responded or failed to respond to a volume change.

In (Naeem Mannan, Kim, Jeong, & Ahmad Kamran, 2016), an eye tracker was used as part of the noise removal process in order to better identify eye movement along with eye blinks. For blinking this study used composite multi scale entropy to identify the independent component that contained primarily blinks.

In (Siew Cheok & Raveendran, 2008), the variance of each signal was used to identify primarily artefactual independent components. High variance independent components were found to contain most of the noise due to ocular events. In addition to this, the study used the

position and propagation direction of the independent components; a propagation from left to right indicated that the independent component was related to horizontal eye movements, and a propagation from front to back indicates that the independent component is related to vertical eye movements and blinking.

5.1: The runica Algorithm

For this study, the runica algorithm in the eeglab (Delorme & Makeig, 2004) MATLAB toolbox was used. This algorithm uses Bell & Sejnowski's infomax algorithm in addition to the gradient feature of Amari, Cichocki and Yang (S. Amari, A. Cichocki, 1996).

Given a data vector x, the Infomax ICA algorithm treats this vector as a sum of independent components s. A weight matrix A can be used to transform this data vector x into a set of independent components **s**.

$$\boldsymbol{s} = \boldsymbol{A}\boldsymbol{x} \tag{5.1}$$

ICA is a blind source separation algorithm. The independent sources are not known ahead of time, the algorithm finds a mixing matrix **A** that is capable of separating these sources out of the original EEG channels. This is achieved by extracting a linear combination of signals out of the original signal, such that each signal's non-Gaussianity is maximized. Non-Gaussianity is a desirable trait for an independent component because a signal with high resemblance to the Gaussian distribution indicates that the signal is a mixture of several signals that need to be separated out. Each independent component's Gaussianity will be far lower than the raw signal (Hyvärinen, Karhunen, & Oja, 2001).

5.2: The Effect of ICA on EEG Data

In this study six channels of EEG data were used. The runica algorithm is able to extract six independent components from six channels of EEG data. Two of these independent components capture most of the high amplitude noise, and have been known to contain mostly ocular activity. There is a clear resemblance to the VEOG signal when their signals are compared. Figure 24 and Figure 25 show how ocular components are distributed into independent components.

5.2.1: Blink and Saccade Separation using ICA

Note in Figure 25 how the blink that occurs between 70-73 seconds on the EOG signal is placed into independent component 2, and the vertical saccade that occurs during that time is split into independent component 1.

Splitting up this section of time into a blink independent component and a saccade independent component makes the actual shape of the blink clear; the blink is not as distorted as it is in the EOG signal. The most major benefit of ICA in this case is that only IC1 and IC2 contain a significant amount of high amplitude noise. Independent components 3 through 6 are free from this, and their data is now significantly more representative of actual neural activity.

5.2.2: Processing Time

ICA is a very effective tool for separating out different sources in EEG signals, however it is a very computationally intensive. ICA is only able to find independent components in relatively long trials (even trials 40 seconds in length have been too short to find the mixing matrix), and longer trials take a considerable (and variable) amount of time to process, even using versions of ICA optimized for speed such as FastICA (Hyvarinen, 1999). Since FastICA was not deterministic enough for real time work in this case, and because runica has been known to be more accurate than FastICA, runica was chosen for processing in this study. Implementing real time processing involving ICA is a challenge, and is a good subject for future work.

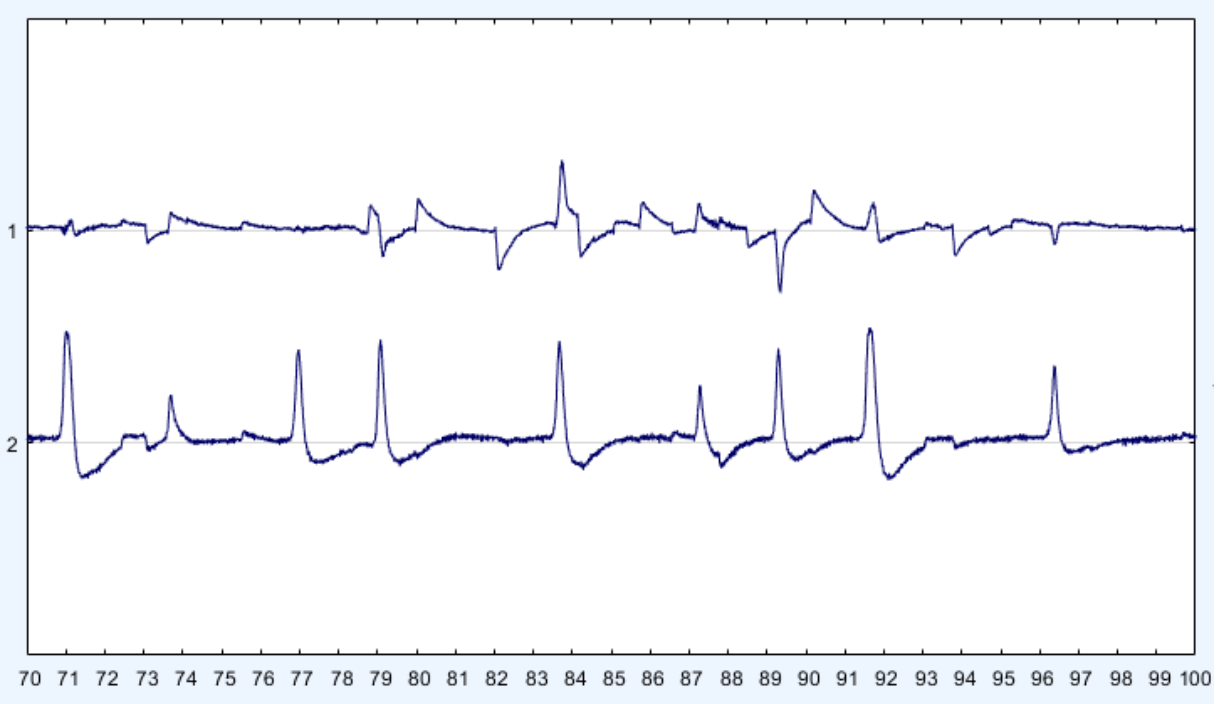

Figure 24: EOG Signal (HEOG on top, VEOG on bottom)

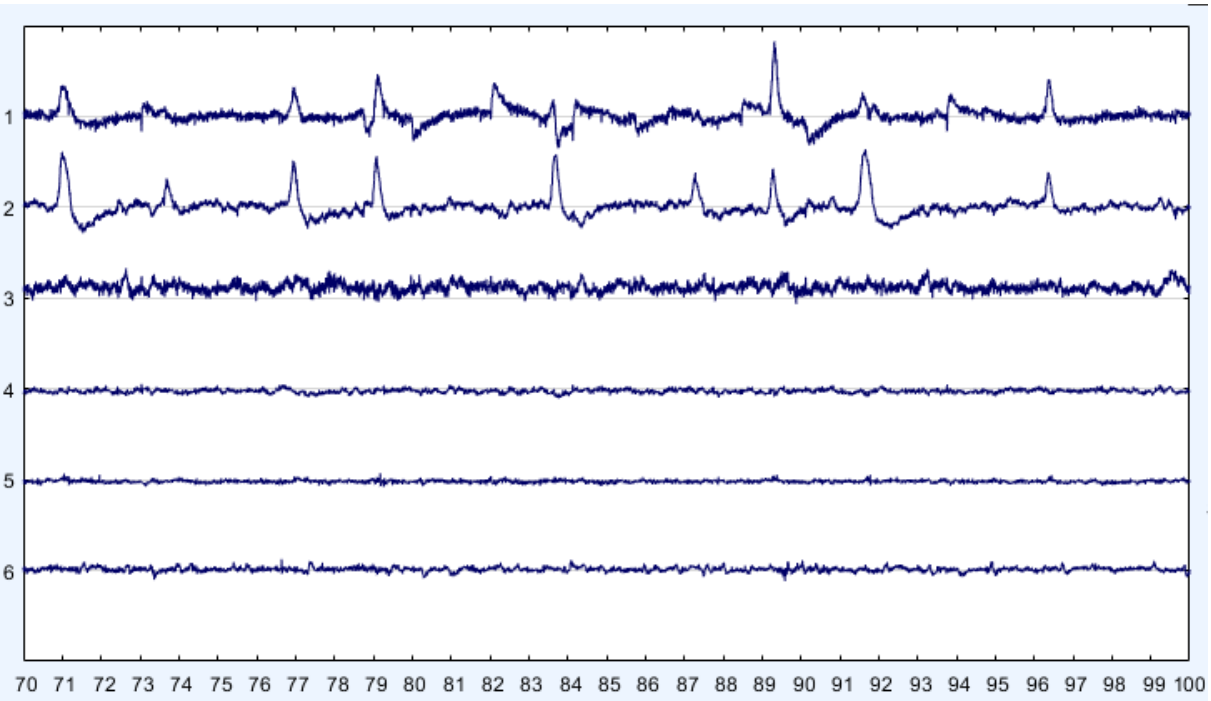

Figure 25: Independent Components

5.3: The Effect of ICA on Blink Detection

The same feature extraction and detection methods discussed in Chapter 4 were applied to the independent components extracted from each trial. Independent components 1 and 2, which contain the most blink and saccade activity consistently had high detection accuracies as blinks.

Table 9: IC Space: Best Features for each Independent Component

IC Index	**Highest Accuracy**	**Lowest Accuracy**
1	**98.7**	**91.2**
2	98.4	88.5
3	92.5	84.6
4	94.5	86.6
5	94	86
6	94.1	87.6

5.3.1: t-test Results: Feature Ranking (IC1 and IC2)

As mentioned previously, independent components 3 through 6 tend to contain low amounts of ocular activity, though they were still able to achieve a rather high accuracy on their own, due to the large variety of features extracted from the signal. Each independent component has its own set of best features. Listed below are the top five features for IC1 and IC2.

Table 10: IC Space: Best Features for IC 1

	IC1: Best Features
1	'Wavelet: Dj1 Standard Deviation'
2	'SFT Feature: Theta'
3	'Wavelet: Dj1 Mean'
4	'Wavelet: Dj2 Mean'
5	'Wavelet: Dj2 Standard Deviation'

Table 11: IC Space: Best Features for IC 2

	IC2: Best Features
1	'Wavelet: Dj2 Band Power'
2	'SFT Feature: Delta'
3	'SFT Feature: Theta'
4	'Wavelet: Dj2 Standard Deviation'
5	'Wavelet: Dj1 Band Power'

5.3.2: t-test Results: Feature Ranking (All ICs)

In addition to ranking the features independently, a t-test was also performed that considered all six independent components together to find the top 20 features using data from all ICs simultaneously.

Table 12: IC Space: Top 20 Best Features

	All ICs: Best Features
1	IC1 'SFT Feature: Theta'
2	IC1 'Wavelet: Dj1 Standard Deviation'
3	IC1 'SFT Feature: Delta'
4	IC1 'Wavelet: Dj1 Band Power'
5	IC1 'Wavelet: Dj2 Band Power'

6	IC1 'Wavelet: Dj1 Mean'
7	IC1 'Wavelet: Dj Standard Deviation'
8	IC1 'Wavelet: Dj Mean'
9	IC1 'Wavelet: Dj2 Standard Deviation'
10	IC1 'Wavelet: Dj Band Power'
11	IC1 'Variance'
12	IC1 'Wavelet: Dj2 Mean'
13	IC1 'Standard deviation of local maxima'
14	IC1 'Standard deviation of local minima'
15	IC1 'Standard Deviation'
16	IC1 'Interquartile Range'
17	IC1 'Wavelet: Aj Standard Deviation'
18	IC2 'Wavelet: Dj2 Band Power'
19	IC2 'SFT Feature: Theta'
20	IC1 'Interquartile range of local maxima'

Note how this table is primarily features from independent component 1. This table gives a good representation of how much more important independent component 1 is for detection than independent component 2.

5.3.3: The Effect of ICA on t-test Results

Table 13: Top 10 Time Space Features: p-Values

P Value Means: Subject 2 Time Space		
#	**p-Value**	**Description**
1	*Less than 2.2251e-308*	Ch 2 : 'SFT Feature: Theta'
2	*Less than 2.2251e-308*	Ch 3 : 'SFT Feature: Theta'
3	8.70488290E-269	Ch 3 : 'Wavelet: Dj1 Band Power'
4	4.54973536E-198	Ch 2 : 'Wavelet: Dj1 Standard Deviation'
5	2.69821111E-178	Ch 2 : 'Wavelet: Dj1 Band Power'
6	3.16146581E-175	Ch 2 : 'Wavelet: Dj1 Mean'
7	4.98444349E-168	Ch 3 : 'Wavelet: Dj2 Band Power'
8	5.94651367E-167	Ch 3 : 'Wavelet: Dj Band Power'

9	5.23889305E-164	Ch 3 : 'SFT Feature: Delta'
10	2.50498676E-162	Ch 3 : 'Wavelet: Dj Standard Deviation'

Table 14: Top 10 Independent Component Features: p-Values

P Value Means: Subject 2 ICA		
#	p-Value	Description
1	4.01825922E-62	IC 1 : 'SFT Feature: Theta'
2	5.54896226E-43	IC 1 : 'Wavelet: Dj1 Standard Deviation'
3	6.74126114E-42	IC 1 : 'SFT Feature: Delta'
4	3.09823368E-38	IC 1 : 'Wavelet: Dj1 Band Power'
5	1.04323902E-37	IC 1 : 'Wavelet: Dj2 Band Power'
6	1.00318338E-36	IC 1 : 'Wavelet: Dj1 Mean'
7	2.95876514E-33	IC 1 : 'Wavelet: Dj Standard Deviation'
8	4.49085258E-30	IC 1 : 'Wavelet: Dj Mean'
9	5.13018057E-27	IC 1 : 'Wavelet: Dj2 Standard Deviation'
10	1.87630819E-24	IC 1 : 'Wavelet: Dj Band Power'

As can be seen in the above tables, the top 10 feature values for the time space blink detection system has far lower p-values than the top 10 feature values for the independent component analysis system. Independent component analysis seems to have negatively affected the ability of these features in blink detection. This could explain the larger range of accuracies for the ICA trials, but note that despite the lower feature p-values, the maximum accuracy for any trial is still higher for ICA than it is for time space processing.

Conclusion

This thesis proposed a feasible wearable system for detecting blinks in both the independent component (IC) space and time space. The proposed system automatically annotates electooculogram data using dynamic time warping and known blink templates. DTW was found to be a capable method for blink detection for common types of blinks found in EOG signals. The labels from EOG are then used along with features extracted in the time and independent component space to train a model for blink detection. It was found that the frontal channels in the time domain are very accurate at detecting blinks. In the independent component space one independent component was found to be the most useful for blink detection. For this experiment, ICA separated the blink activity into a single IC (as shown by the high feature relevance and accuracy of IC1), but the time space classification had very high accuracy as well.

Based on this, it should be possible to refine the feature space and detection process further to make a system that can achieve high accuracy results without needing to use ICA.

References

Chang, Chih-Chung, & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 2:27:1–27:27.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Croft, R. J., & Barry, R. J. (1998). EOG correction: A new perspective. *Electroencephalography and Clinical Neurophysiology*, *107*(6), 387–394. https://doi.org/10.1016/S0013-4694(98)00086-8

Delorme, A., & Makeig, S. (2004). EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, *134*(1), 9–21. https://doi.org/10.1016/j.jneumeth.2003.10.009

Ghandeharion, H., & Erfanian, A. (2010). A fully automatic ocular artifact suppression from EEG data using higher order statistics: Improved performance by wavelet analysis. *Medical Engineering and Physics*, *32*(7), 720–729. https://doi.org/10.1016/j.medengphy.2010.04.010

He, P., Kahle, M., Wilson, G., & Russell, C. (2005). Removal of Ocular Artifacts from EEG: A Comparison of Adaptive Filtering Method and Regression Method Using Simulated Data. *Conference Proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, *2*(4), 1110–3. https://doi.org/10.1109/IEMBS.2005.1616614

Hyvarinen, A. (1999). Fast and robust fixed-point algorithm for independent component analysis. *IEEE Transactions on Neural Networks and Learning Systems*, *10*, 626–634. https://doi.org/10.1109/72.761722

References

Hyvärinen, A., Karhunen, J., & Oja, E. (2001). *Independent Component Analysis*. New York, NY: John Wiley & Sons.

Hyvärinen, A., & Oja, E. (2000). Independent component analysis: Algorithms and applications. *Neural Networks*. https://doi.org/10.1016/S0893-6080(00)00026-5

Kong, W., Zhou, Z., Hu, S., Zhang, J., Babiloni, F., & Dai, G. (2013). Automatic and direct identification of blink components from scalp EEG. *Sensors (Basel, Switzerland)*, *13*(8), 10783–10801. https://doi.org/10.3390/s130810783

Kugiumtzis, D., & Tsimpiris, A. (2010). Measures of Analysis of Time Series (MATS): A MATLAB Toolkit for Computation of Multiple Measures on Time Series Data Bases. *Journal Of Statistical Software*, *33*(5), 25. https://doi.org/http://dx.doi.org/10.18637/jss.v033.i05

Lawhern, V., Hairston, W. D., & Robbins, K. (2013). DETECT: A MATLAB Toolbox for Event Detection and Identification in Time Series, with Applications to Artifact Detection in EEG Signals. *PLoS ONE*, *8*(4). https://doi.org/10.1371/journal.pone.0062944

Mahajan, R., & Morshed, B. I. (2015). Unsupervised eye blink artifact denoising of EEG data with modified multiscale sample entropy, kurtosis, and wavelet-ICA. *IEEE Journal of Biomedical and Health Informatics*, *19*(1), 158–165. https://doi.org/10.1109/JBHI.2014.2333010

Makeig, S., J. Bell., A., Jung, T.-P., & Sejnowski, T. J. (1996). Independent Component Analysis of Electroencephalographic Data. *Advances in Neural Information Processing Systems*, *8*(3), 145–151. https://doi.org/10.1109/ICOSP.2002.1180091

Naeem Mannan, M. M., Kim, S., Jeong, M. Y., & Ahmad Kamran, M. (2016). Hybrid EEG???Eye tracker: Automatic identification and removal of eye movement and blink artifacts from electroencephalographic signal. *Sensors (Switzerland)*, *16*(2). https://doi.org/10.3390/s16020241

S. Amari, A. Cichocki, H. H. Y. (1996). A New Learning Algorithm for Blind Signal Separation. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.7.6283&rep=rep1&type=pdf

Schlögl, A., Keinrath, C., Zimmermann, D., Scherer, R., Leeb, R., & Pfurtscheller, G. (2007). A fully automated correction method of EOG artifacts in EEG recordings. *Clinical Neurophysiology*, *118*(1), 98–104. https://doi.org/10.1016/j.clinph.2006.09.003

Siew Cheok, N., & Raveendran, P. (2008). Removal of EOG artifacts using ICA regression method. In *IFMBE Proceedings* (Vol. 21 IFMBE, pp. 226–229). https://doi.org/10.1007/978-3-540-69139-6-59

Wang, Z., Xu, P., Liu, T., Tian, Y., Lei, X., & Yao, D. (2014). Robust removal of ocular artifacts by combining Independent Component Analysis and system identification. *Biomedical Signal Processing and Control*, *10*(1), 250–259. https://doi.org/10.1016/j.bspc.2013.10.006

Winkler, I., Haufe, S., & Tangermann, M. (2011). Automatic classification of artifactual ICA-components for artifact removal in EEG signals. *Behav Brain Funct*, *7*(1), 30. https://doi.org/10.1186/1744-9081-7-30