VISION BASED COGNITIVE FATIGUE DETECTION

BY NEELESH KUMAR

A thesis submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Computer Science

Written under the direction of

Prof. Konstantinos Michmizos

and approved by

_____

_____

_____

New Brunswick, New Jersey

May, 2017

ABSTRACT OF THE THESIS

Vision Based Cognitive Fatigue Detection

by Neelesh Kumar

Thesis Director: Prof. Konstantinos Michmizos

Analyzing human activity is a basic component of any system, be it biological or artificial, that aims to predict future behavior. Tracking and recognizing voluntary and involuntary action traits are basic endeavors for artificial vision systems that aim to predict cognitive fatigue, a major cause for road and workplace accidents. In this work, we developed a vision system to detect the early onset of fatigue. In collaboration with the Magnetoencephalography Lab at MIT, we collected synchronous brain (Magnetoencephalography - MEG) and behavioral (high-speed camera) data from 13 subjects in a 3-hour task that was designed to induce cognitive exhaustion. We derived a set of 8 eye-movement and 6 head-movement features and trained classifiers(Random Forest, K-Nearest Neighbor, and Support Vector Machine) for two classes (fatigue, non-fatigue) and three classes (fatigue, transition stage, non-fatigue). The models achieved average test accuracies of 98%, 97%, 92% (two classes) and 92%, 90%, 87% (three classes) respectively, for combined features. To further validate our models, we used the alpha band power in the MEG data as the neural indicator of fatigue. A regression analysis between the camera-based features and the alpha band power revealed an average ρ^2 = 0.59 coefficient. Here, we also propose a new method to detect the early stages of fatigue by using the classification error as our behavioral marker. Specifically, we found

that the accuracy of the classifiers was higher when the distance between the time intervals of labels for non-fatigue and fatigue was larger; We estimated the total number of the mis-classified fatigue and non-fatigue data points in a sliding window: the fatigue (non-fatigue) number was high (low) in the beginning -non-fatigue stage- and became lower (higher) with time, signifying clear periods of fatigue and non-fatigue. We also observed a sharp change in the labels from non-fatigue to fatigue in the interval of 40-50 minutes, when using a sliding window, which signifies early stages of fatigue. Our results are promising in terms of designing a fully automated system that can predict ones effective operation range, based on behavioral and neurophysiological cues.

Acknowledgements

I would like to thank my advisor Prof. Konstantinos Michmizos for his continous guidance and support throughout my thesis. His sincerity, thoroughness and perseverance have been a constant source of inspiration for me. It is only through his cognizant efforts that this work has seen the light of the day. A special thanks to Prof. Dimitris Metaxas for his guidance at various stages of this work. I would also like to thank Prof. Ahmed Elgammal for being a part of my defense committee and showing interest in my work.

I am grateful to Sepehr Janghorbani for helping us conduct EEG experiments for my thesis. Special thanks also to the Magnetoencaphalography (MEG) Lab at Massachusetts Institute of Technology (MIT), and its co-founder Prof. Dimitri Pantazis. The work is in joint collaboration with them.

Many thanks to the faculty members of the Department of Computer Science at Rutgers. I am lucky to have been taught by them. I would also like to acknowledge the administrative staff at the department who have made life easy for me during my master's program. Finally, I would like to thank Chintan Trivedi, with whom this work has been done.

Table of Contents

Abstract ii

Acknowledgements iv

List of Figures vii

List of Tables x

1. Introduction 1

1.1. Motivation 1

1.2. Prior Work 1

1.3. Flow of Process 3

1.4. Outline of the report 4

2. Theory 5

2.1. Machine Learning Design Cycle 5

2.1.1. Data Collection 5

2.1.2. Feature Extraction 6

2.1.3. Model Selection 7

2.1.4. Training the classifier 7

2.1.5. Testing and Evaluation 7

2.2. Classifiers in Machine Learning 8

2.2.1. Random Forest 8

2.2.2. K- Nearest Neighbour 8

2.2.3. Support Vector Machine 8

2.3. Magnetoencaphalography (MEG) 9

2.3.1. Working principle of MEG 9

2.3.2. Components of MEG 9
2.3.3. Features of MEG 10
2.3.4. Fatigue Detection using MEG 10

3. Data collection and Feature extraction 11
3.1. Data collection 11
3.2. Feature Extraction 11
Eye Related Features 14
Head Movement Related Features 17

4. Model Fitting and Integration with MEG 28
4.1. Creating labeled data 28
Two Class Classification 28
Three Class Classification 29
4.2. Preparing data for training and testing 29
4.3. Separability of Features 30
4.4. Training and Testing 31
4.5. Out of Bag 32
4.6. Early stages of Fatigue 33
4.7. MEG Data Analysis 36

5. Discussion of Results and Conclusion 38
5.1. Model Fitting 38
5.2. Integration with MEG 40
5.3. Conclusion 40

List of Figures

2.1. Machine Learning Design Cycle 6
2.2. Alpha band in MEG signal 10
3.1. Face tracking points for test subject 1 12
3.2. Sliding window technique for feature extraction. We slide the blue window of 1 minute starting at 0, and derive 14 features per window. For a 3 hour video per subject, we have 180 feature vectors of 14 dimensions. 13
3.3. Plot of eye blink rate vs. sliding window for test subject 1. Yellow curve is the raw feature value with lots of peaks and valleys. Blue curve shows the median filtered values which is much smoother. Red curve shows the increasing trend of the feature. 14
3.4. Plot of average eyelid distance vs. sliding window for test subject 2. Blue curve shows the median filtered values. Red curve shows the trend of the feature. 15
3.5. Plot of eye closure time vs. sliding window for test subject 2. Blue curve shows the median filtered values. Red curve shows the trend of the feature. 16
3.6. Plot of eye blink rate vs. sliding window for test subject 2. Blue curve shows the median filtered values. Red curve shows the trend of the feature. 17
3.7. Plot of eye surface area vs. sliding window for test subject 2. Blue curve shows the median filtered values. Red curve shows the trend of the feature. 18
3.8. Plot of eye circularity vs. sliding window for test subject 2. Blue curve shows the median filtered values. Red curve shows the trend of the feature. 19
3.9. Plot of distance from fixed point vs. sliding window for test subject 2. Blue curve shows the median filtered values. Red curve shows the trend of the feature. 20

3.10. Plot of eye opening time vs. sliding window for test subject 2. Blue curve shows the median filtered values. Red curve shows the trend of the feature. 20
3.11. Plot of eye closing time vs. sliding window for test subject 2. Blue curve shows the median filtered values. Red curve shows the trend of the feature. 21
3.12. Plot of average eyelid distance for all test subjects. Blue curve shows the median filtered values. Red curve shows the trend of the feature. 21
3.13. Plot of eye blink rate for all test subjects. Blue curve shows the median filtered values. Red curve shows the trend of the feature. 22
3.14. Plot of eye closure time for all test subjects. Blue curve shows the median filtered values. Red curve shows the trend of the feature. 22
3.15. Plot of circularity of eye for all test subjects. Blue curve shows the median filtered values. Red curve shows the trend of the feature. 23
3.16. Plot of surface area of eye for all test subjects. Blue curve shows the median filtered values. Red curve shows the trend of the feature. 23
3.17. Plot of distance of eye from external point for all test subjects. Blue curve shows the median filtered values. Red curve shows the trend of the feature. 24
3.18. Plot of eye opening time for all test subjects. Blue curve shows the median filtered values. Red curve shows the trend of the feature. 24
3.19. Plot of eye closing time for all test subjects. Blue curve shows the median filtered values. Red curve shows the trend of the feature. 25
3.20. Plot of pitch speed against sliding window for all test subjects. 25
3.21. Plot of pitch acceleration against sliding window for all test subjects. . . 25
3.22. Plot of yaw speed against sliding window for all test subjects. 26
3.23. Plot of yaw acceleration against sliding window for all test subjects. . . 26
3.24. Plot of pitch+yaw speed against sliding window for all test subjects. . . 26
3.25. Plot of pitch+yaw acceleration against sliding window for all test subjects. 27
4.1. Designating labels for two class classification 29

4.2. Designating labels for three class classification 30
4.3. Understanding separability of the points in the feature matrix. The red mode corresponds to the class of the data point that was selected (in this case Fatigue), and the blue mode corresponds to the other class. 31
4.4. Out of bag Analysis 33
4.5. Test 1: Checking classification results on a sliding window. We have fixed windows for fatigue and non fatigue, and we train classifiers using these class labels. 34
4.6. Result of the analysis averaged over all test subjects. 34
4.7. Result of the analysis for each test subject. 35
4.8. Assignment of labels for this analysis. In case 1, we kept the non fatigue class fixed, and had a sliding window for fatigue class. In case 2, we had a sliding window for non fatigue class and kept the fatigue class fixed. . 36
4.9. Result of the analysis for both the cases. 36

List of Tables

4.1. Test Accuracy for 2 class classification (in %) 32

4.2. Test Accuracy for 3 class classification (in %) 32

4.3. Test Accuracy for out of bag (in %) 33

4.4. Regression ρ^2 values 37

Chapter 1

Introduction

The phenomenon of cognitive fatigue is defined as the gradual decrease in the optimal cognitive performance, as a result of performing an activity which requires use of cognitive resources, for a long period of time. The symptoms include low attention levels, tiredness, low energy levels, decreased motivation, or even depression. This is different from physical fatigue, where tiredness is a result of physical exhaustion.

1.1 Motivation

According to The National Highway Traffic Safety Administration, drowsy driving resulted in 72,000 crashes, 44,000 injuries and 800 deaths in 2013. These figures are an underestimate. The presence of a reliable and robust system that is capable of detecting early signs of fatigue can alert the drivers when they are drowsy and vastly reduce the number of accidents. The system also has many applications elsewhere. In the industries, it can be used to determine the number of hours that a particular work requires before the worker needs to be given a break. Other applications include alerting workers operating heavy and dangerous machinery, and mining sites.

1.2 Prior Work

Much work has been done in the area of fatigue detection. These studies can broadly be divided into two categories: Vision based techniques and Brain Data based techniques. Some of these works are listed below.

- **Vision based fatigue detection:** Kong et al. [1] use machine vision and adaboost algorithm to train a classifier to distinguish between open and close state

of the eye. They use PERCLOS measure to detect fatigue in real time based on whether the eye is closed or open. However speed and accuracy remains an issue. Singh and Kaur [2] also give a machine vision based approach in which localization and segmentation algorithms are applied to extract eye region from the image of the face, and then it is determined in the processed image whether the eye is open or close.

Sacco and Farrugia [3] use Viola-Jones classifier to detect driver's facial features, and integrate with it a support vector machine to determine whether the facial appearance is fatigued or not. However the system is not robust to poor illumination.

Zhang et al. [4] use Haar algorithm for localization of face. Using eye tracking and vertical projection matching, they detect presence or absence of fatigue. Their technique is robust to poor lighting, and head movements, but lack speed.

Peopie et al. [5] present yet another vision based fatigue detection algorithm which detects eyelid movements and uses normalized cross correlation function as a classifier to determine if the eyes are closed or open. The system raises an alarm if they eyes are closed for long time.

Most of these works take into account only one or two eye related features (generally eye closing time). More emphasis has been on localization and segmentation algorithms to localize eye region, rather than deriving features that may help in classification.

- **Brain data based fatigue detection:** Other approaches have used brain data (Electroencaphalography or Magnetoencaphalography) to detect fatigue. These approaches are based on the fact that the power of the brain signals in the frequency range 8-12 Hz (also known as alpha signal) increases when the subjects become more tired. Wang et al. [6] extract eigen values by percentage power spectral density of the filtered EEG signals. They then use ratio of eigen values, and a trained back propagation neural network as the two methods to detect fatigue based on EEG data. However the accuracy that they achieve isn't too high.

Shen et al. [7] extract four features from four spectrum (alpha, beta, theta and delta) of EEG signal from 19 channels to construct a concatenated feature matrix, and use this to train a probabilistic support vector machine. The accuracy that they achieve is lower than that achieved when visual feature are used.
Gharagozlou et al. [8] extract alpha band power of EEG signal ,and perform a P-value test to determine if the increase in alpha band power when the subject is fatigued is significant enough or not. Their results show that the increase in alpha band power is indeed significant.

Our work integrates the two different approaches to fatigue detection- vision based and brain data based. We used visual features to train a classifier to detect fatigue, and then validated those features using Magnetoencaphalography data. We also propose a novel method where we use classification error as a means to detect early stages of fatigue

1.3 Flow of Process

1. The subjects were made to play a game which requires cognitive resources for a period of three hours. The game is adaptive in nature which means that if the subjects were performing well, the difficulty of the game would increase. A high speed camera was used to keep track of the subjects faces over the entire duration in which they played the game. Using face recognition, 66 face landmark points were identified and the coordinates stored.

2. We extract features based on these face points recorded over the duration of the game. We use these features to inspect visual symptoms of fatigue.

3. On the basis of these symptoms, we decide whether fatigue has started to set in for the subject or not.

1.4 Outline of the report

- Chapter 2 is divided into two sections. The first section gives an overview of the Machine Learning and Computer Vision theories that we have used. The second section discusses Cognitive Fatigue from the view of Magnetoencaphalography(MEG).

- In chapter 3, we discuss how the data was collected, and how and what features were extracted. Here we make a distinction between eye related features and head movement related features. We also discuss how the features were pre-processed.

- Chapter 4 gives an in-depth explanation of how the classifiers were trained and tested, and the different kinds of classification tasks that were performed. We also discuss the correlation between the results provided by MEG data and the classifiers.

- Chapter 5 is a discussion of the results we obtain.

Chapter 2

Theory

In this chapter, we give a brief overview of the various machine learning and computer vision principles that we adopted for our research. We also explain the principle of Magnetoencaphalography (MEG) and how it relates to fatigue.

2.1 Machine Learning Design Cycle

Figure 1.1 shows the design cycle of a traditional machine learning system.The main building blocks are:

2.1.1 Data Collection

Collecting data is the first and one of the most critical steps in machine learning. In fact, many experts believe that data plays a greater role in determining the success of one's application, than the machine learning algorithms themselves. Typically, machine learning applications require high volume of data which may not always be available. But the goal is to collect enough data representing each class (in case of supervised learning), and cluster (in unsupervised learning). For this to happen, data must be uniformly sampled from the population. This ensures that data is unbiased. For supervised learning, we must also collect labels along with data. Generally, we don't have enough labelled data. So after the data is collected, we label it manually based on domain knowledge. We then apply a series of pre-processing steps on the collected data to remove noise, make it homogeneous if collected from multiple sources, and normalize it. The collected data is then partitioned into training and test data.

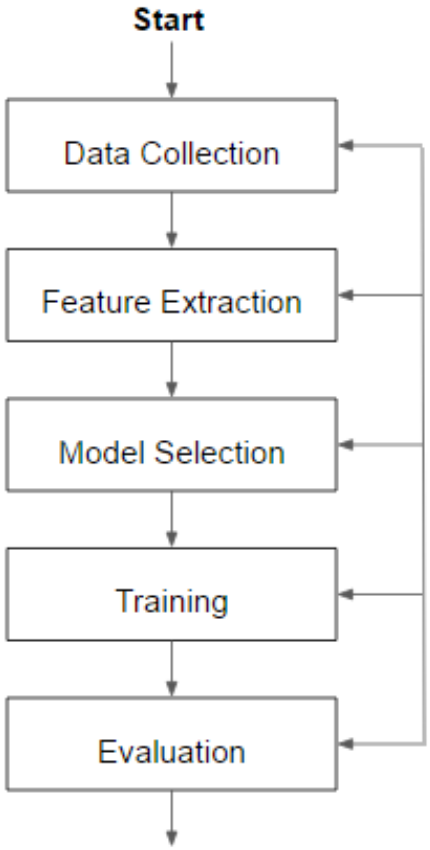

Figure 2.1: Machine Learning Design Cycle

2.1.2 Feature Extraction

Features are quantifiable characteristics of data. For the purpose of machine learning, they are those measurements that give similar values for data belonging to same category. Feature extraction is usually done heuristically, using domain knowledge and prior information. Care should be taken while choosing features. For the machine learning algorithms to be effective, the chosen features should be as discriminative as possible; this implies that they should have same values for data belonging to same category, but sufficiently different values for different categories. They should also be invariant and robust with respect to the variations and noise in the environment. Lastly, they should be simple to extract in order to save on computational costs.

While in most of the cases, the features are determined using expert knowledge, in some other cases where expert knowledge is unavailable, we use raw data itself

as features. This is common in Vision applications, where the pixel intensity values themselves become features. An example of such kind of algorithms is deep learning.

2.1.3 Model Selection

The general objective of machine learning algorithms is to understand the underlying distribution that generates the data. Model selection is the problem of deciding what statistical model to fit to the data, in order to best explain the underlying distribution. Expert knowledge plays an important role in model selection. It allows us to make reasonable assumptions about the underlying distribution, which narrows down the possible models to choose from. One can, for example assume that the data is linearly separable, and we can select from a class of linear classifiers to fit to the data. Here, a distinction is made between parametric and non parametric models. While parametric models make strong assumptions about distribution of data, non parametric models make weak or no assumptions about the distribution. A lot of factors go into deciding what model to select. Computational costs, performance, availability of expert knowledge are some of them.

2.1.4 Training the classifier

Once we have selected the model (also known as classifier in the case of supervised learning), we train our dataset on that classifier. Essentially, we are trying to find the function that maps the input space, which is the matrix of features, to the output space, which is the class labels. We only use the training partition of our data for this purpose.

2.1.5 Testing and Evaluation

After we find the functional mapping from input space to output space from the previous step, we use it to test our classifier. We evaluate the function at the data points that are part of the test partition of our data, and obtain the output. We then use a variety of evaluation measures to check if the output returned is to our satisfaction or not. Some measures are: average error, mean squared error, loss, etc. The model should be

able to generalize well, which means that if it performs well on already seen data, it should perform well on new data points as well.

2.2 Classifiers in Machine Learning

In the context of supervised machine learning, a classifier is an algorithm that returns a function that maps input data to one of the categories. Depending upon the data itself, various kinds of classifiers can be used. We describe below the three classifiers that we have used in this work:

2.2.1 Random Forest

Random forest is an extension of decision trees. During training, the algorithm constructs many decision trees by randomly sampling (with replacement) from the training set and fitting trees to these samples. When a new data point arrives, we evaluate it using all the decision trees that are constructed during training. The class of the new data points is the mode of the class returned by all decision trees.

2.2.2 K- Nearest Neighbour

K- nearest neighbour (KNN) is a non parametric classifier, i.e. it doesn't make any assumption about the underlying distribution of the data, and rather tries to estimate it based on the data points observed. In order to classify a new data point, the algorithm finds out the k points that are "closest" to it, based on distance measures such as euclidian distance, manhattan distance, etc. The class of the new data point is the mode of the class of the k neighbors. Choosing the optimal value of k is critical. Usually, cross validation is done in order to determine k.

2.2.3 Support Vector Machine

Support Vector Machine (SVM) is a margin based classifier. Margin is the minimum distance of the data points from the hyperplane that separates the data points. Usually, points which are closer to the hyperplane are harder to categorize, because any slight

shift in the hyperplane can change the class of those data points. Amongst all the hyperplanes that separate the data, SVM returns one that has the maximum margin, in order to better classify "hard points". In other words, SVM tries to find the hyperplane that best separates the classes.

2.3 Magnetoencaphalography (MEG)

Magnetoencaphalography (MEG) is a technique that records the magnetic fields that are generated by the neuron activity inside the brain. MEG allows to measure the brain activity on a millisecond by millisecond basis. Because of the large number of sensors that MEG uses, it is possible to localize brain activity with reasonable accuracy.

2.3.1 Working principle of MEG

Individual neurons in the brain produce post synaptic current on application of a synaptic input. This current has a very weak magnetic field associated with it. While the magnetic field produced by a single neuron is negligible, the effect produced by tens of thousands of neurons that are simultaneously activated in a region is significant enough. This magnetic field can then be detected outside the head, and is of the magnitude of order 10-15 Tesla.

2.3.2 Components of MEG

The basic component of MEG scanners is SQUID (superconducting quantum interference device). The SQUID sensors detect the magnetic flux around brain. The MEG system contains around 100-300 SQUID sensors. The SQUID sensors are bathed in a large liquid helium cooling unit at approximately -269 degrees C. Due to low impedance at this temperature, the SQUID device can detect and amplify magnetic fields generated by neurons a few centimeters away from the sensors. The MEG system needs to be operated in a magnetic shielded room, in order to avoid the effect of external magnetic fields.

2.3.3 Features of MEG

- Unlike fMRI which is a secondary measure of brain activity, MEG is a direct measure. MEG shows absolute brain activity while fMRI shows relative brain activity (by comparing to a reference neuronal activity).
- The technique is completely non invasive, and hence safe.
- MEG has excellent spatial and temporal resolution. It can record brain activity on millisecond by millisecond basis, and the source of the signals can be localized with reasonable accuracy.
- MEG does not make noise during operation, and allows some degree of movement of head.

2.3.4 Fatigue Detection using MEG

Previous works[7-8] have shown that fatigue is directly related to the oscillations in the frequency range of 8-12 Hz of brain signals (EEG or MEG), also known as alpha band. In particular, when the person becomes tired, or loses concentration, or there is decrease in mental alertness as a result of fatigue, the alpha band power decreases. This fact can be exploited to determine the early stages of fatigue in a subject when he is performing an activity that requires cognitive resources.

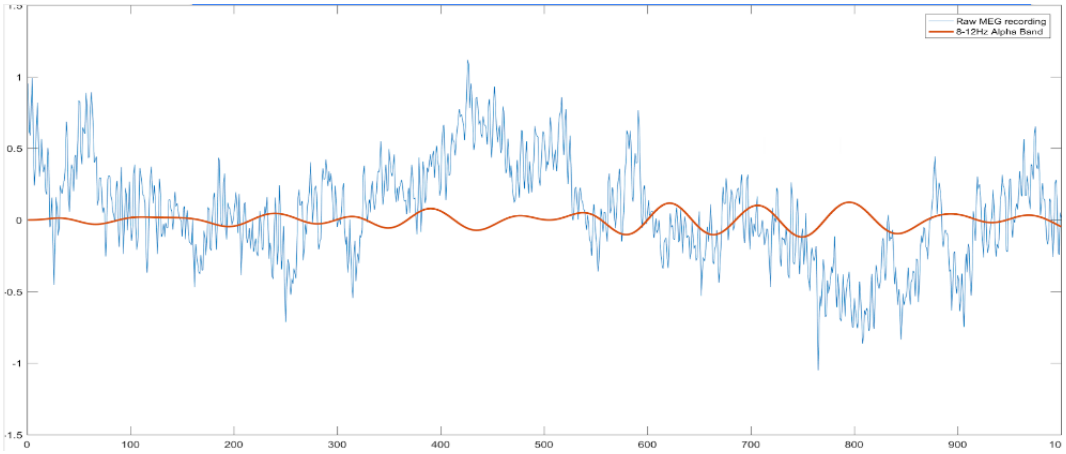

Figure 2.2: Alpha band in MEG signal

Chapter 3

Data collection and Feature extraction

In this chapter, we explain in detail the data acquisition process, and how we represented and stored the data. We then discuss the features that we extracted, along with the pre-processing steps that were applied.

3.1 Data collection

13 test subjects were made to play a game that required cognitive efforts for a duration of three hours. The game is adaptive in nature, which means that its difficulty increases if the subjects are playing it very well. The adaptive nature of the game ensured that the subjects did not get bored, and concentrated throughout the duration of the game. After a point of time, it was expected that the subjects will get tired.

Using a high speed camera, the subjects were recorded for the entire duration of the game. A MEG system was also installed on the head of the subjects, in order to record MEG data.

We obtained 66 face landmark points for each test subject, as illustrated in figure 3.1 for each frame in the video recording. The face tracking points were expressed in terms of x and y coordinates. The raw data is hence a collection of face landmark points for all test subjects, through the entire duration of the video.

3.2 Feature Extraction

Features were extracted from the face tracking points that we obtained in the above step. In order to extract meaningful features, we used a sliding window approach. We kept the length of the window as 60 seconds with no overlap, and calculated a set of 8

Figure 3.1: Face tracking points for test subject 1

eye related and 6 head movement features for each frame in that window. Depending upon the nature of the feature, we took the mean, median or mode of the feature values in that window. We then slid the window through the duration of the video, and generated more features per window. Figure 3.2 illustrates how features were extracted We applied the following pre-processing steps to the features:

1. **Median Filtering**: For each feature, we applied a one dimensional median filter across the vector containing feature values for all windows. This was done to remove noisy instances in the feature vector which can occur due to reasons beyond our control. It also interpolates missing data where face tracking was lost, or the subject was taking a break. The idea is to run through the vector element by

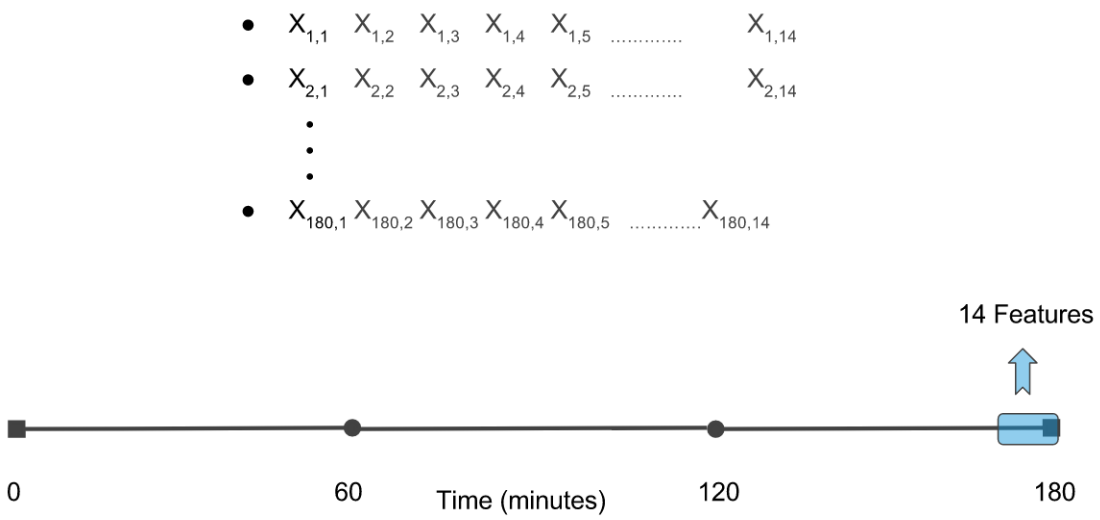

Figure 3.2: Sliding window technique for feature extraction. We slide the blue window of 1 minute starting at 0, and derive 14 features per window. For a 3 hour video per subject, we have 180 feature vectors of 14 dimensions.

element, and replace each element with the median of the neighbouring elements. The number of neighbours was specified as a parameter to the algorithm, and was determined by cross validation.

2. **Trend Analysis**: We fit a polynomial of an appropriate order to the features in order to observe trend. This is an important step because it allowed us to see how feature values were changing with time over the course of the game. We expect the trend to be either increasing or decreasing, if the features that we extract are relevant.

3. **Z-score Normalization**: Z score normalization takes a vector as an input, and subtracts from each entry the mean of that vector, and divides it by the standard deviation of the vector.

$$z = \frac{x - \mu}{\sigma}$$

where x is an element in the vector, μ is the mean of that vector and σ is the standard deviation.

This was done to correct the following two types of variances:

- Across feature variance- Different features of the same subject can have different scale of values. Features of higher magnitudes may dominate other features, even if they are not relatively significant for classification. In order to correct this problem, we performed feature-wise normalization using z score, which scales up or scales down all the values to same scale.
- Across subject variance- Different subjects can exhibit vastly different values of the same feature. In order to compare different subjects, we performed subject wise normalization using z score.

An example of a raw feature for one test subject (Eye blink rate) and its preprocessed version is shown in figure 3.3. We extracted the following 8 eye related and 6 head movement related features:

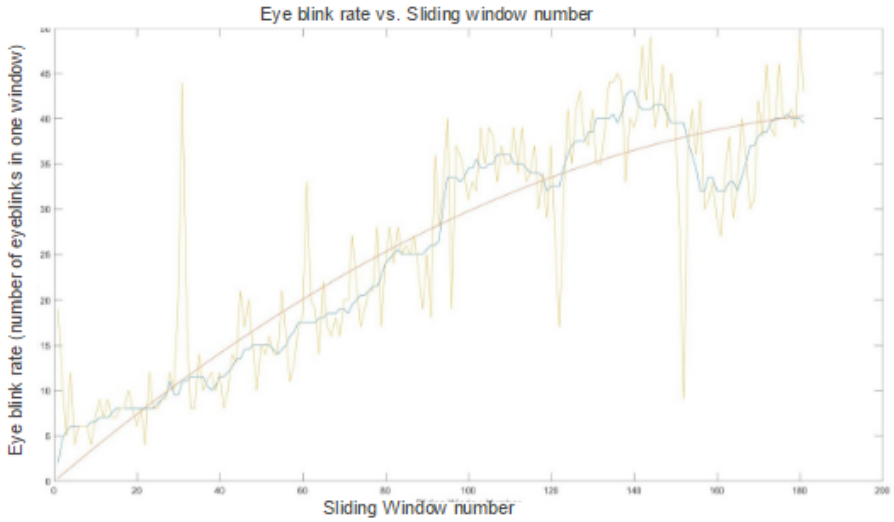

Figure 3.3: Plot of eye blink rate vs. sliding window for test subject 1. Yellow curve is the raw feature value with lots of peaks and valleys. Blue curve shows the median filtered values which is much smoother. Red curve shows the increasing trend of the feature.

Eye Related Features

The face tracking points in and around the eyes were used to compute the following eye related features:

- **Average eyelid distance:** We have four uniformly spread points each on upper and lower eyelid. We computed the euclidian distance between the corresponding

pair of points on the upper and lower eyelid, and took the average of the four distances for each frame. This is called the eyelid distance. Then we took the average of these eyelid distances for all frames in a window of 60 seconds to generate one single average eyelid distance for that window. Figure 3.4 shows the plot of the average eyelid distance against the sliding windows for test subject 2.

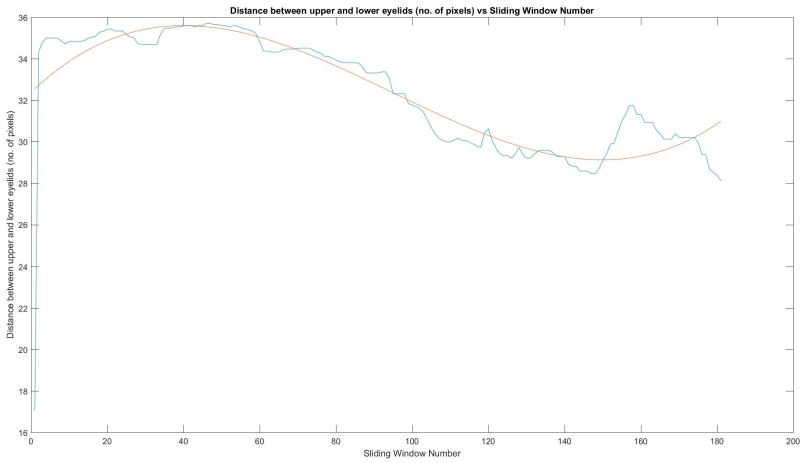

Figure 3.4: Plot of average eyelid distance vs. sliding window for test subject 2. Blue curve shows the median filtered values. Red curve shows the trend of the feature.

- **Eye closure time:** We considered the eyes to be closed if the eyelid distance is below a certain threshold. The threshold may be different for each subject and is determined from the histogram of the eyelid distances of each subject. Eye closure time is the total number of frames in the window of 60 seconds, for which the eyelid distance is lesser than the threshold. Figure 3.5 shows the plot of the eye closure time against the sliding windows for test subject 2.

- **Eye blink rate:** For each frame in a window of 60 seconds, we determined the state of the eye- open or closed (using the definition in the above feature). An eye blink is said to take place when the state of the eye changes from closed to open. We counted the number of times there was a state change from closed to open in the window which gave us the eye blink rate (blinks per minute). Figure 3.6 shows the plot of the eye blink rate against the sliding windows for test subject 2.

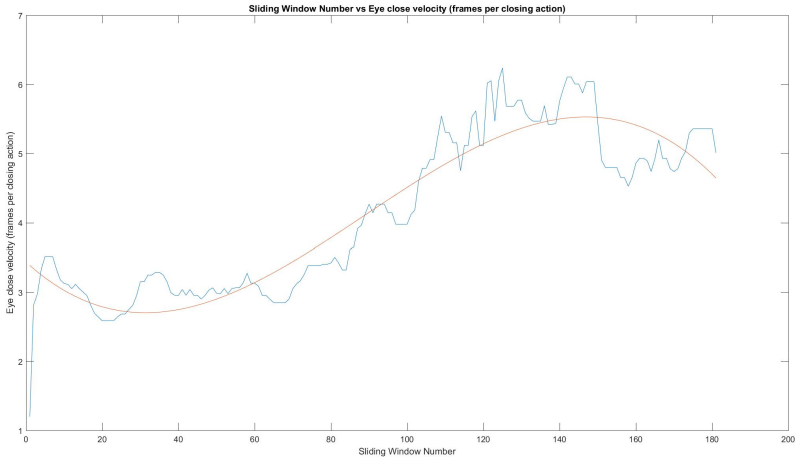

Figure 3.5: Plot of eye closure time vs. sliding window for test subject 2. Blue curve shows the median filtered values. Red curve shows the trend of the feature.

- **Surface area of the eye:** The surface area of an eye is the area of the polygon formed by vertices of the 4 corner points of the eye- minimum x coordinate (left), maximum x coordinate (right), minimum y coordinate (down) and maximum y coordinate (up). For each frame in the window of 60 seconds, we computed the surface area of left and right eye and took the average of it. Then we returned the mean of the surface area of all frames in a particular window as the surface area of the eye in that window. Figure 3.7 shows the plot of the surface area of the eye against the sliding windows for test subject 2.

- **Circularity of eye:** The circularity of eye was calculated using the following formula:

$$c = \frac{4 * \pi * A}{P^2}$$

  where A is the average surface area of left and right eye for a frame, and perimeter is the average perimeter of left and right eye for a frame.

  Perimeter for an eye was calculated as the sum of the euclidian distances of each consecutive pair of face tracking points around the eye. For a window, circularity of eye is the average of the circularity of all frames in the window. Figure 3.8 shows the plot of the circularity against the sliding windows for test subject 2.

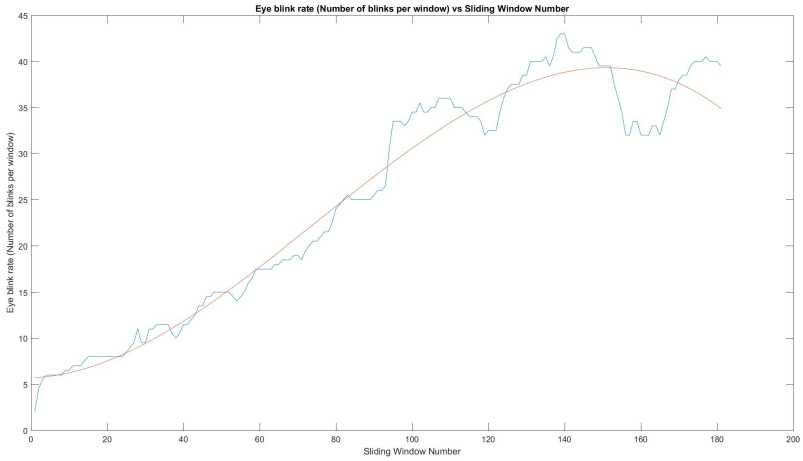

Figure 3.6: Plot of eye blink rate vs. sliding window for test subject 2. Blue curve shows the median filtered values. Red curve shows the trend of the feature.

- **Distance of eye corners from fixed point:** The feature computes the average distance of the four eye corner points from a fixed point for each frame, and then averages it over all the frames in a window. Figure 3.9 shows the plot of the feature against the sliding windows for test subject 2.

- **Eye opening time:** It is a measure of how quickly subjects open their eyes from closed state. Figure 3.10 shows the plot of the eye opening velocity against the sliding windows for all thirteen subjects.

- **Eye closing time:** It is a measure of how long subjects take to close their eyes when its open. Figure 3.11 shows the plot of the eye closing velocity against the sliding windows for all thirteen subjects.

Finally, figures 3.12-3.18 show the eye related features for all the other subjects:

Head Movement Related Features

The head movement related features are based on the three basic kinds of movements- pitch, yaw and roll. We extract the following 6 head movement related features:

- **Pitch velocity:** This is a measure of how quickly subjects move their head in a particular direction. It is calculated as the absolute difference of the mean of

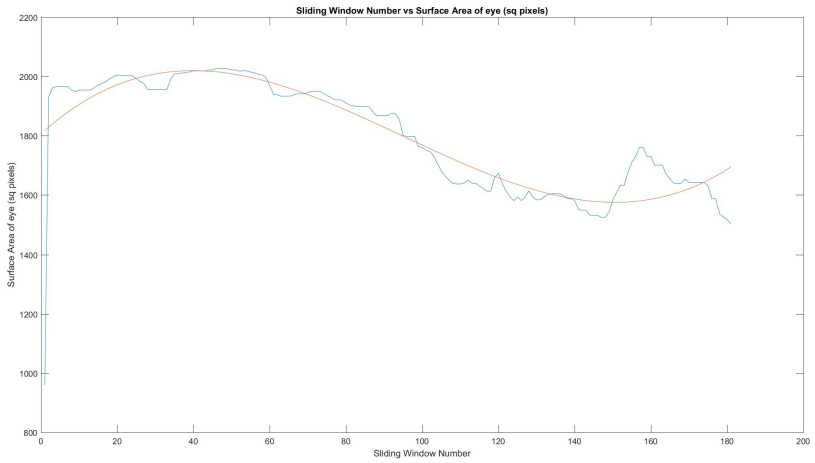

Figure 3.7: Plot of eye surface area vs. sliding window for test subject 2. Blue curve shows the median filtered values. Red curve shows the trend of the feature.

the pitch angles in the first five frames of a window and the mean of the pitch angles in the last five frames of a window. Figure 3.21 shows the plot of the pitch velocity against the sliding windows for all thirteen subjects.

- **Pitch acceleration:** This is the rate of change of pitch velocity during a window of 60 seconds. Figure 3.22 shows the plot of the pitch acceleration against the sliding windows for all thirteen subjects.

- **Yaw velocity:** Figure 3.23 shows the plot of the Yaw velocity against the sliding windows for all thirteen subjects.

- **Yaw acceleration:** Figure 3.24 shows the plot of the Yaw acceleration against the sliding windows for all thirteen subjects.

- **Pitch+Yaw velocity:** Figure 3.25 shows the plot of the Pitch+Yaw speed against the sliding windows for all thirteen subjects.

- **Pitch+Yaw acceleration:** Figure 3.26 shows the plot of the Pitch+Yaw acceleration against the sliding windows for all thirteen subjects.

  For every subject, we calculated these fourteen features for each window. Hence for one window, we have a 1 x 14 dimensional feature vector. The video recordings for each subject were 3 hours long. Hence the total number of windows for one

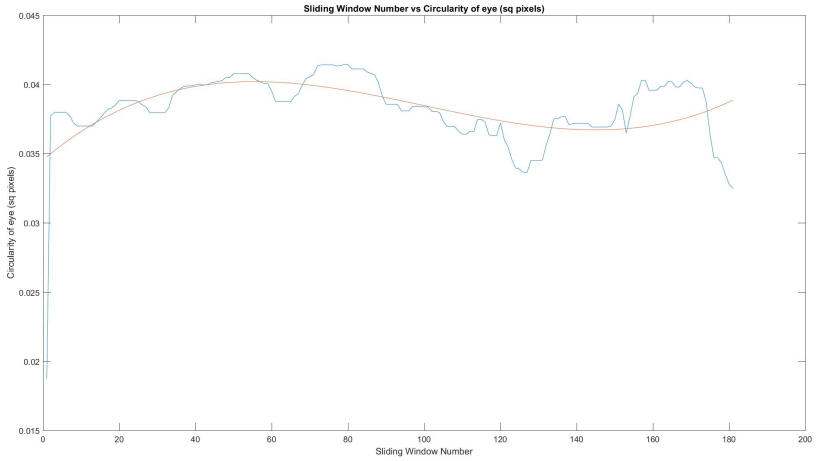

Figure 3.8: Plot of eye circularity vs. sliding window for test subject 2. Blue curve shows the median filtered values. Red curve shows the trend of the feature.

subject is: (60*180)/60= 180. Thus, the dimension of the feature matrix for one subject is 180 x 14.

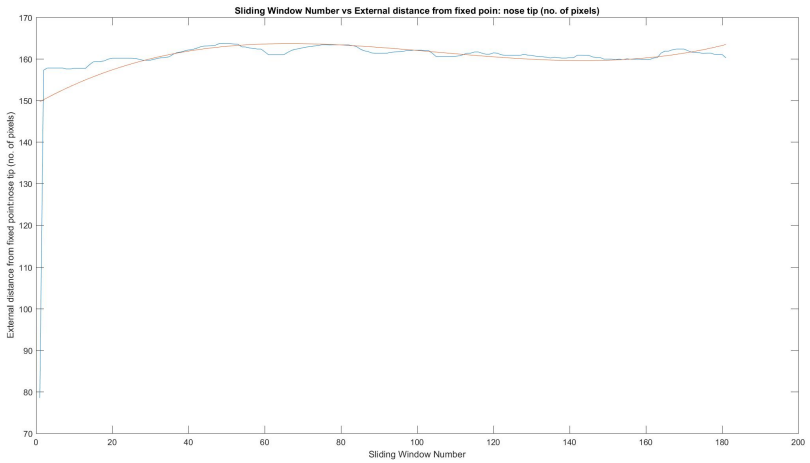

Figure 3.9: Plot of distance from fixed point vs. sliding window for test subject 2. Blue curve shows the median filtered values. Red curve shows the trend of the feature.

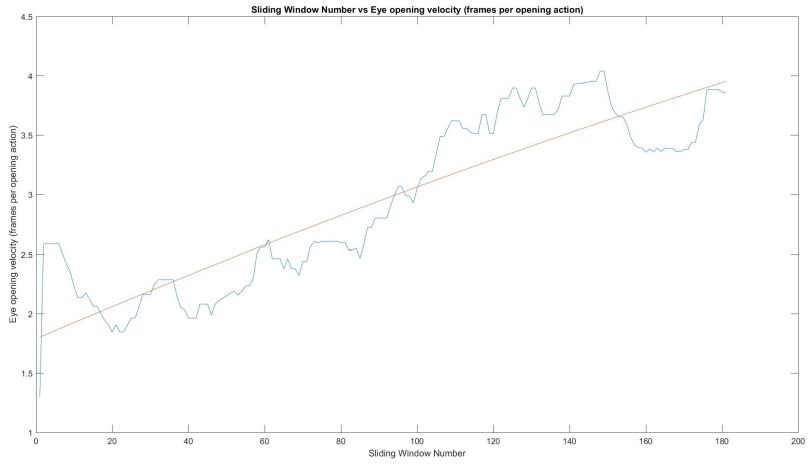

Figure 3.10: Plot of eye opening time vs. sliding window for test subject 2. Blue curve shows the median filtered values. Red curve shows the trend of the feature.

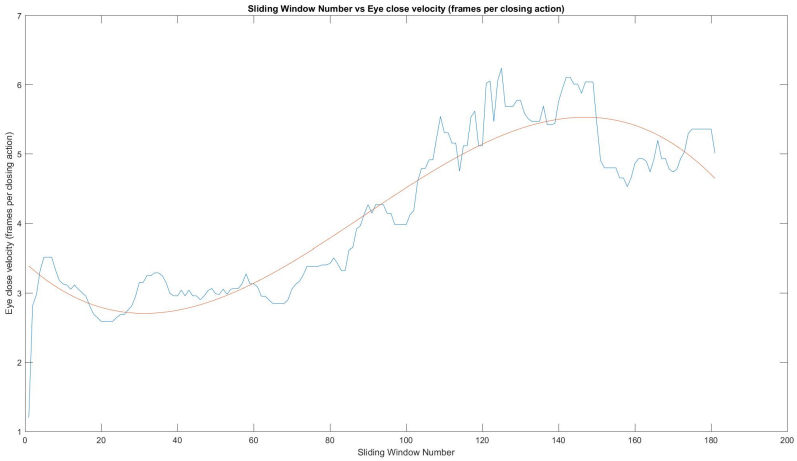

Figure 3.11: Plot of eye closing time vs. sliding window for test subject 2. Blue curve shows the median filtered values. Red curve shows the trend of the feature.

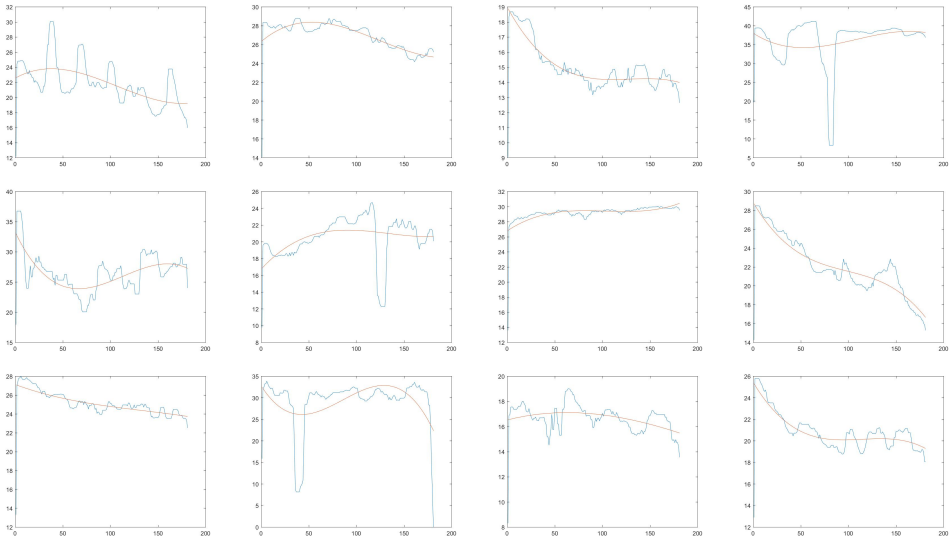

Figure 3.12: Plot of average eyelid distance for all test subjects. Blue curve shows the median filtered values. Red curve shows the trend of the feature.

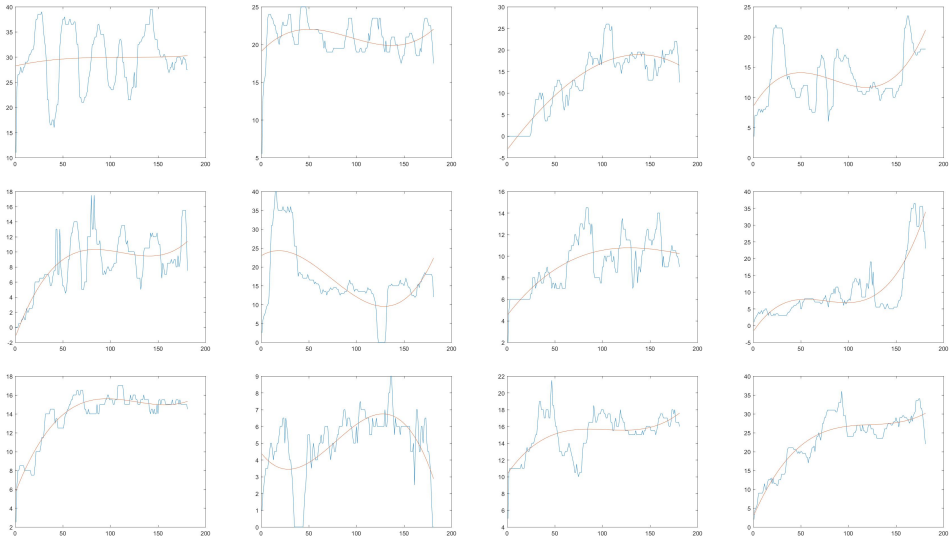

Figure 3.13: Plot of eye blink rate for all test subjects. Blue curve shows the median filtered values. Red curve shows the trend of the feature.

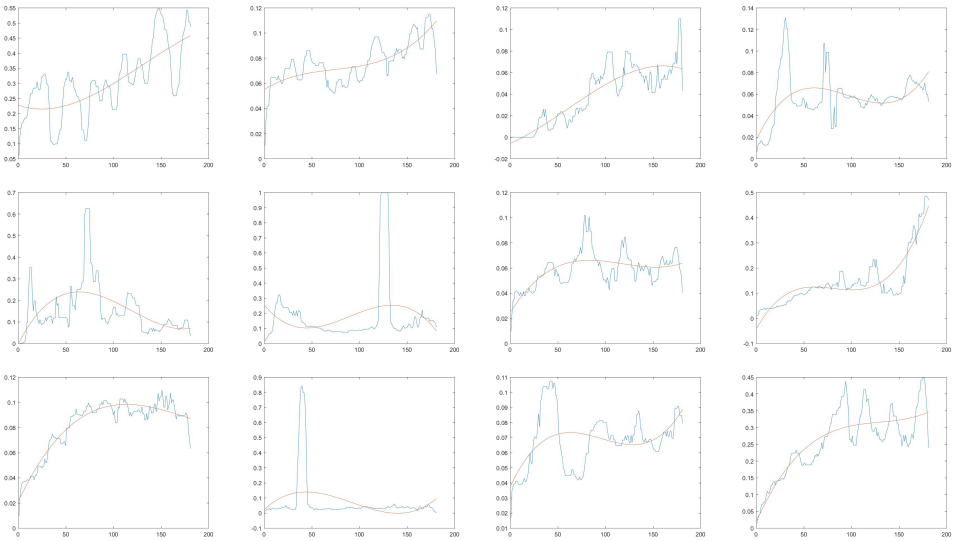

Figure 3.14: Plot of eye closure time for all test subjects. Blue curve shows the median filtered values. Red curve shows the trend of the feature.

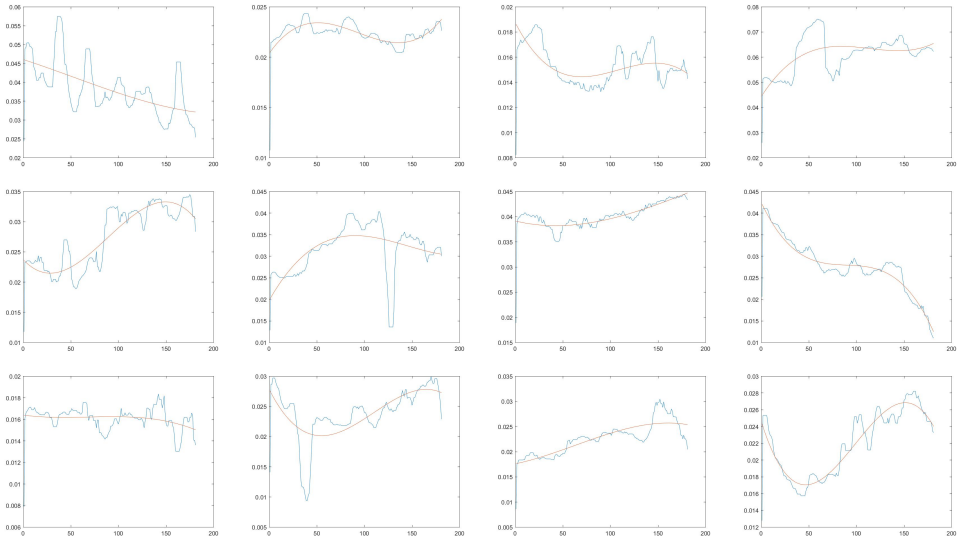

Figure 3.15: Plot of circularity of eye for all test subjects. Blue curve shows the median filtered values. Red curve shows the trend of the feature.

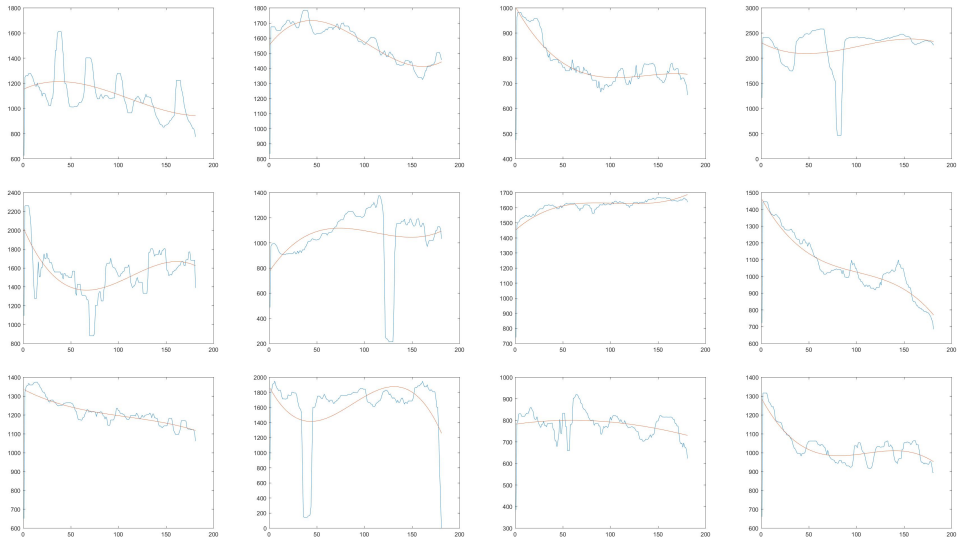

Figure 3.16: Plot of surface area of eye for all test subjects. Blue curve shows the median filtered values. Red curve shows the trend of the feature.

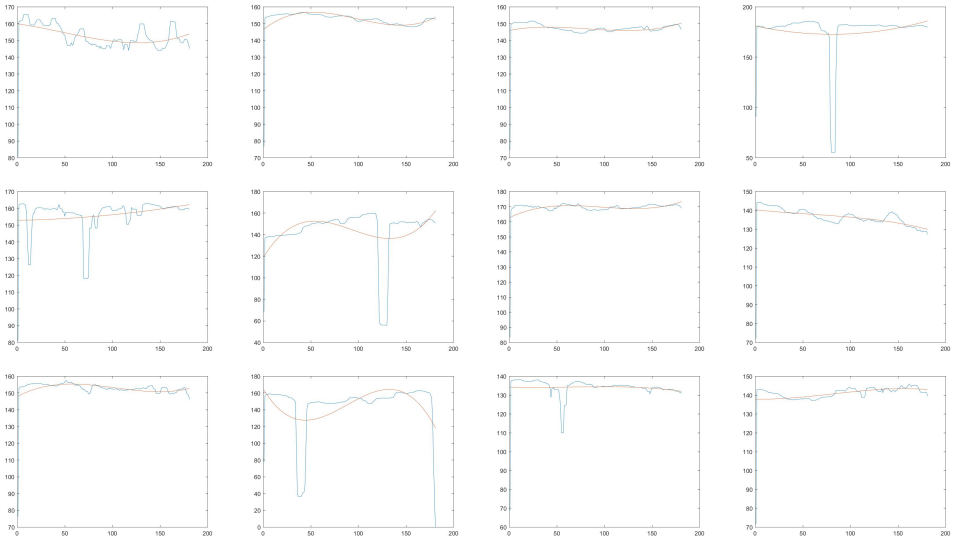

Figure 3.17: Plot of distance of eye from external point for all test subjects. Blue curve shows the median filtered values. Red curve shows the trend of the feature.

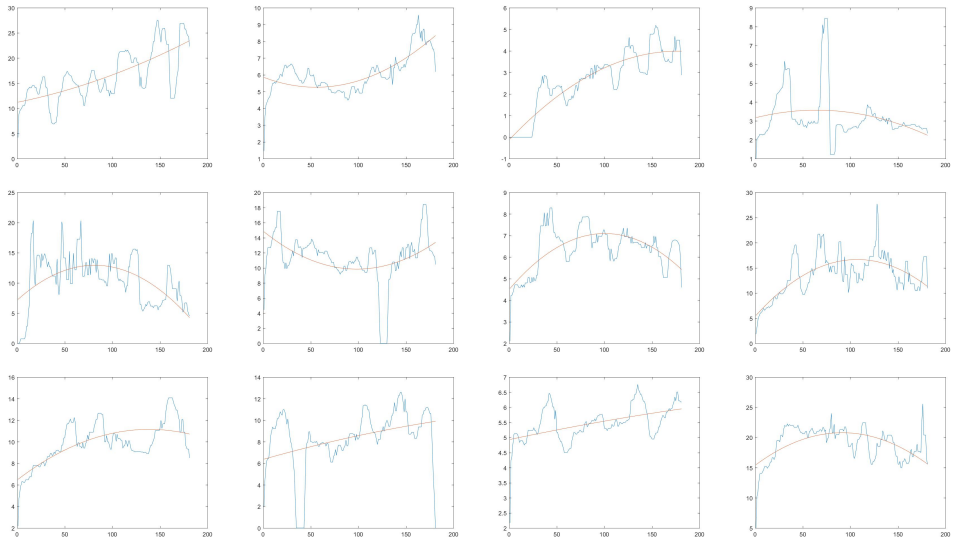

Figure 3.18: Plot of eye opening time for all test subjects. Blue curve shows the median filtered values. Red curve shows the trend of the feature.

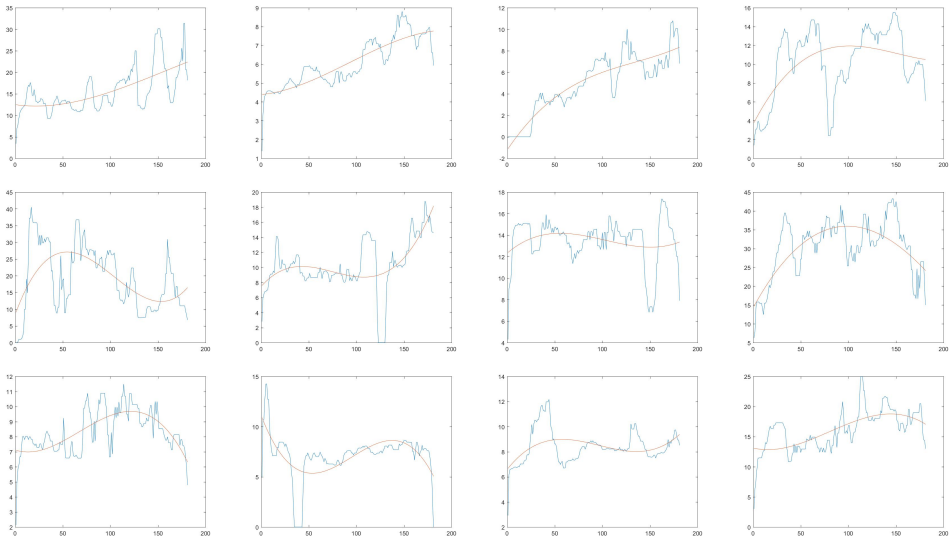

Figure 3.19: Plot of eye closing time for all test subjects. Blue curve shows the median filtered values. Red curve shows the trend of the feature.

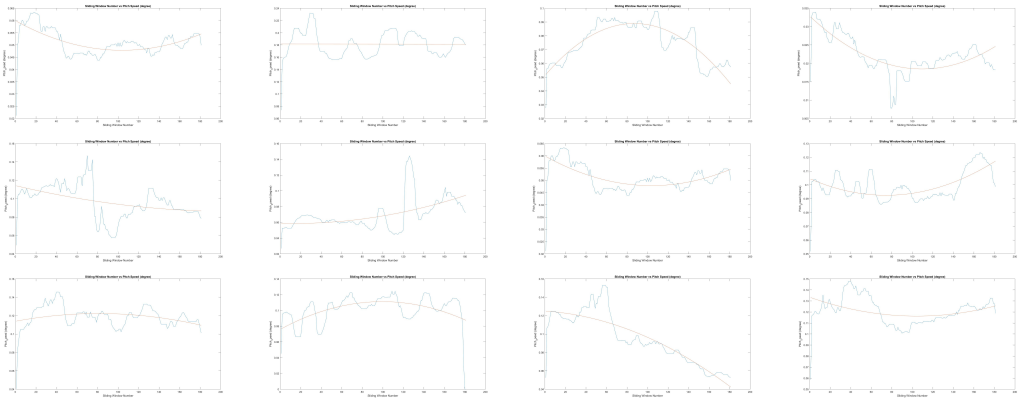

Figure 3.20: Plot of pitch speed against sliding window for all test subjects.

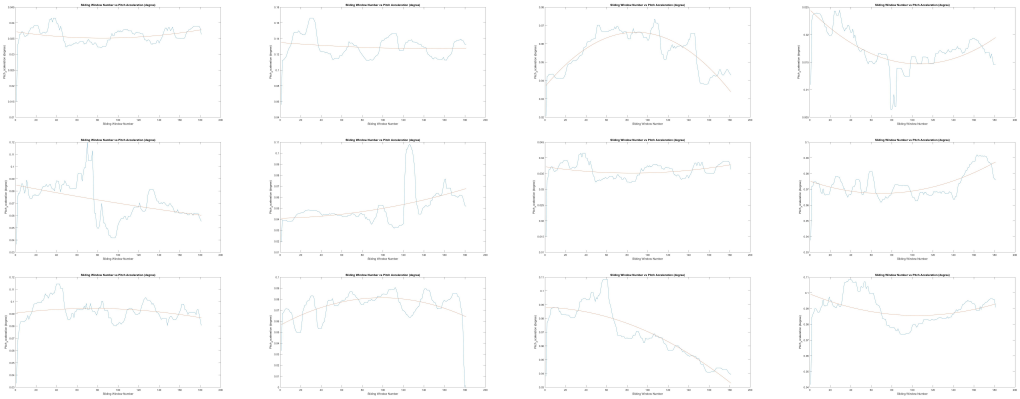

Figure 3.21: Plot of pitch acceleration against sliding window for all test subjects.

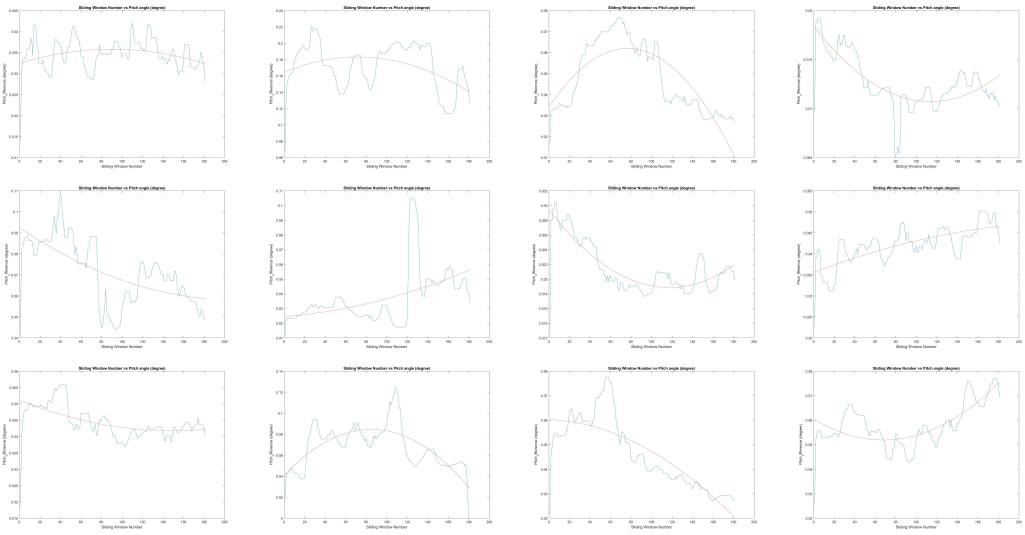

Figure 3.22: Plot of yaw speed against sliding window for all test subjects.

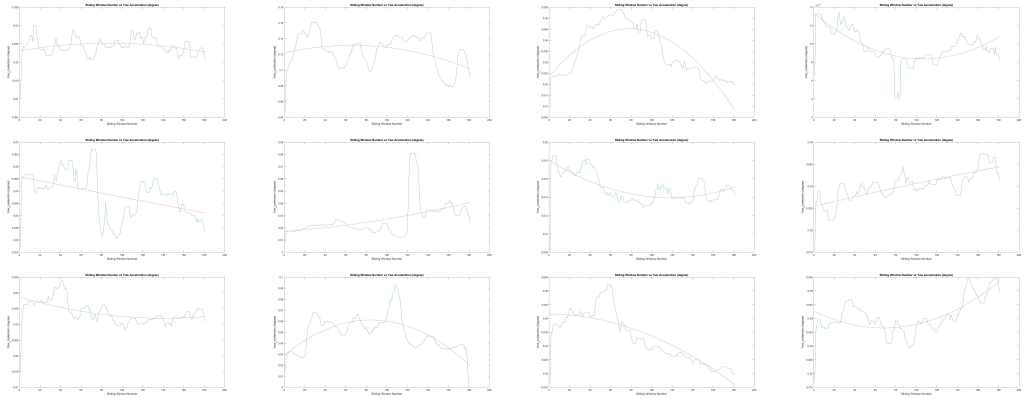

Figure 3.23: Plot of yaw acceleration against sliding window for all test subjects.

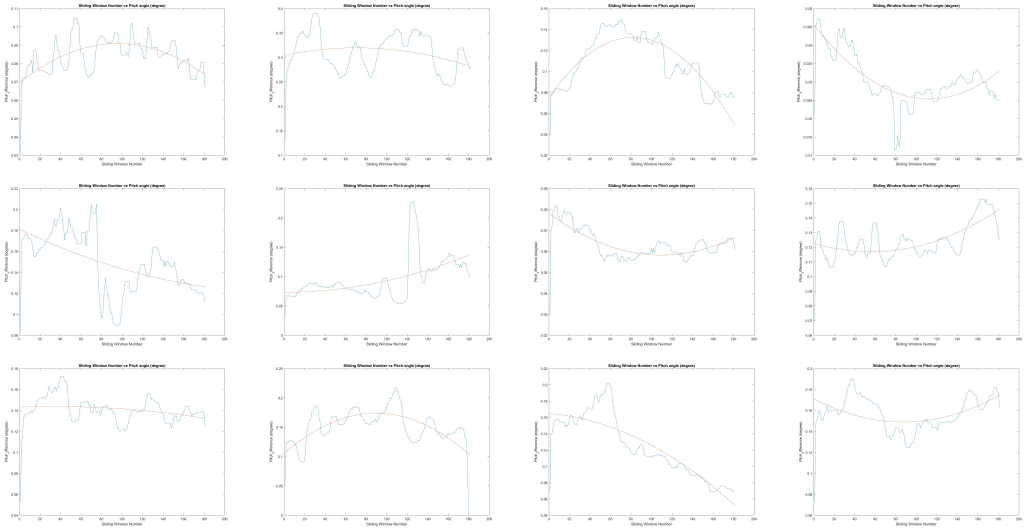

Figure 3.24: Plot of pitch+yaw speed against sliding window for all test subjects.

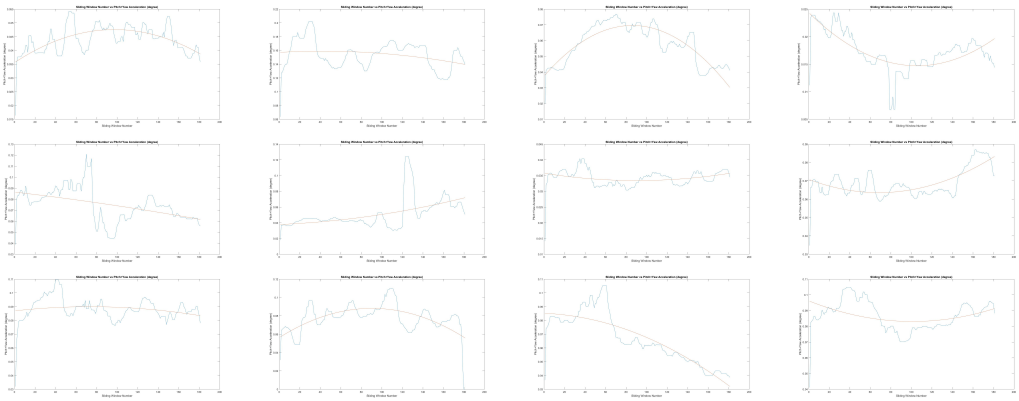

Figure 3.25: Plot of pitch+yaw acceleration against sliding window for all test subjects.

Chapter 4

Model Fitting and Integration with MEG

In this chapter, we analyze the feature matrix that we obtained from feature extraction step to understand their separability. We trained various classifiers using these features, and used the model obtained to test it on new data points. We then analyze the results obtained.

4.1 Creating labeled data

Each feature vector of the feature matrix for every subject obtained from feature extraction is a data point of dimension 14. We assigned labels to each data point for supervised learning. The number of inherent classes that exist in the data is determined heuristically or using domain knowledge. For our experiment, it is reasonable to assume the existence of two classes: one corresponding to the subject not being fatigued, and the other class corresponding to the subject being fatigued. We also included a third class which corresponds to the transition stage between fatigue and non fatigue. Hence we created the following two sets of labelled data:

Two Class Classification

The two classes correspond to the period of fatigue and non fatigue. As stated before, the subjects played the game for 180 minutes. We assigned the label to the interval between minutes 5 and 25 as "Non Fatigue", and between 145 and 165 minutes as "Fatigue". Since we chose the window to be of 60 seconds, we have 21 windows each for fatigue and non-fatigue. We ignored the beginning few minutes because we allowed the subjects to get used to the game. We also ignore the final few minutes because we expected the subjects to be eager to finish the experiment. Such factors may impact

their fatigue level. We did not take into consideration the data points in the interval between 26 and 144 minutes for two class classification. Figure 4.1 shows how we label data points for two class classification.

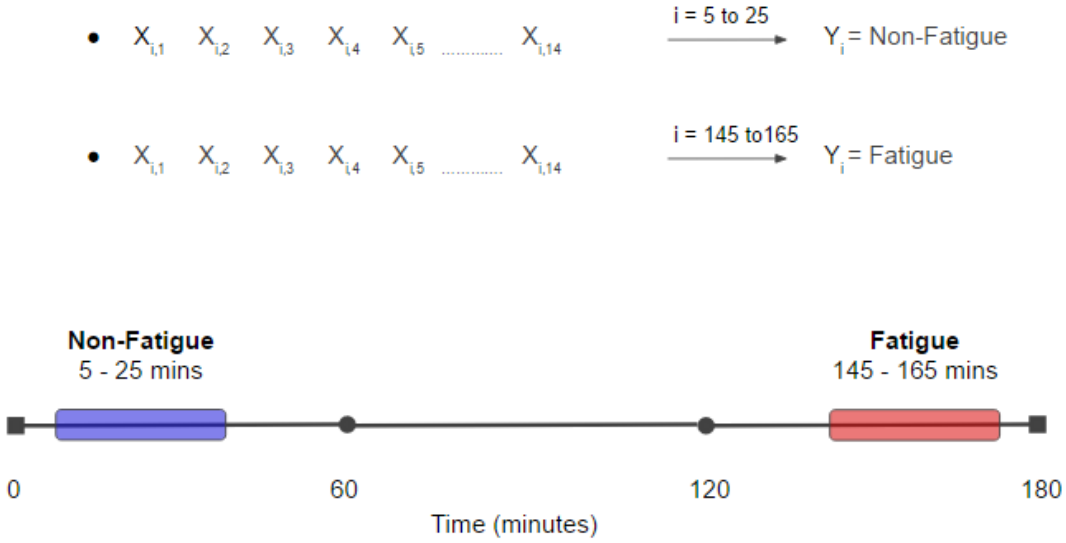

Figure 4.1: Designating labels for two class classification

Three Class Classification

Here, we included another class that corresponds to the transition stage between fatigue and non fatigue. We did this in order to understand how significantly different the fatigue and not fatigue intervals are from the middle interval. The intervals for the class labels corresponding to fatigue and non fatigue were same as that in two class classification (Non-fatigue for 5-25 minutes and Fatigue for 145-165 minutes). The interval for the class label corresponding to transition stage is a window of 21 minutes between 95 and 115 minutes. Figure 4.2 shows how we label data points for three class classification.

4.2 Preparing data for training and testing

We prepared the training data for two class and three class classifiers as follows:

- **2 class classifier:** We have 13 subjects. For each subject, we have 21 windows for non fatigue (a window is a period of 60 seconds), and 21 windows for fatigue.

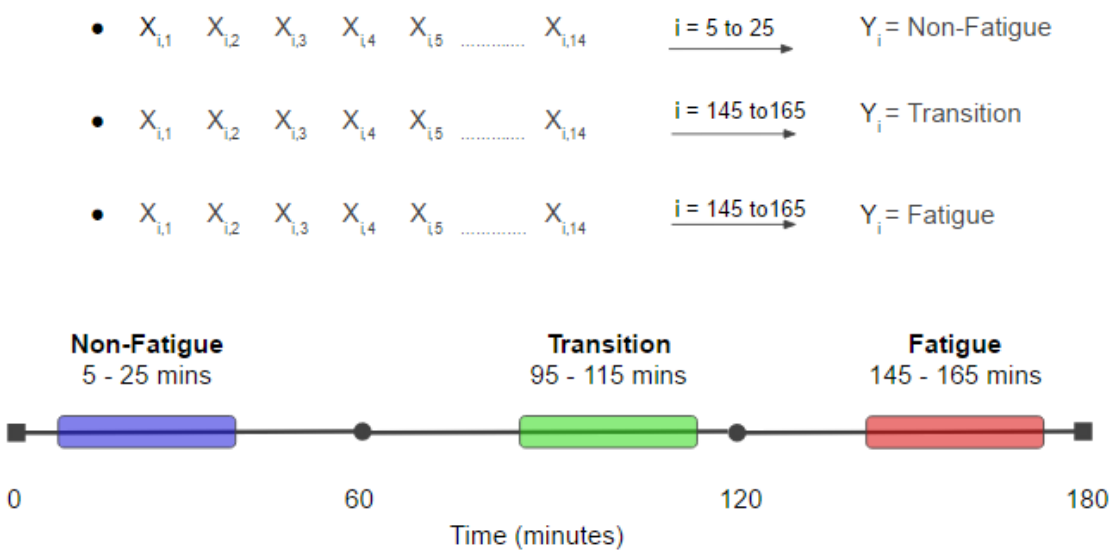

Figure 4.2: Designating labels for three class classification

We calculate 14 features for each subject. Hence for each subject, we have a 42 x 14 feature matrix. We concatenated the matrices for all subjects to obtain a single feature matrix of dimension 546 x 14.

- **3 class classifier:** For each subject, we have 21 windows each for non fatigue, fatigue and transition stage. The concatenated feature matrix in this case is of dimension 704 x 14.

We created random partitions of the dataset obtained from above in the ratio 70:30. 70% of the dataset was used for training and the remaining 30% was used for testing.

4.3 Separability of Features

Before attempting to fit a model through the data points, we tried to see if they are indeed separable or not. For each subject, we have 14 features. It is impossible to visualize 14 dimensional space. Hence, we perform the following analysis:
We took a 14 dimensional data point (which is not an outlier) from the 546 x 14 feature matrix. We computed its distance from all other points in the feature matrix. We obtained a bimodal distribution shown in the figure 4.3. The two modes correspond to the two classes, where one class is that of the data point that was selected, and the

other class is the class that is not of the data point selected. This implies existence of two classes in the data matrix.

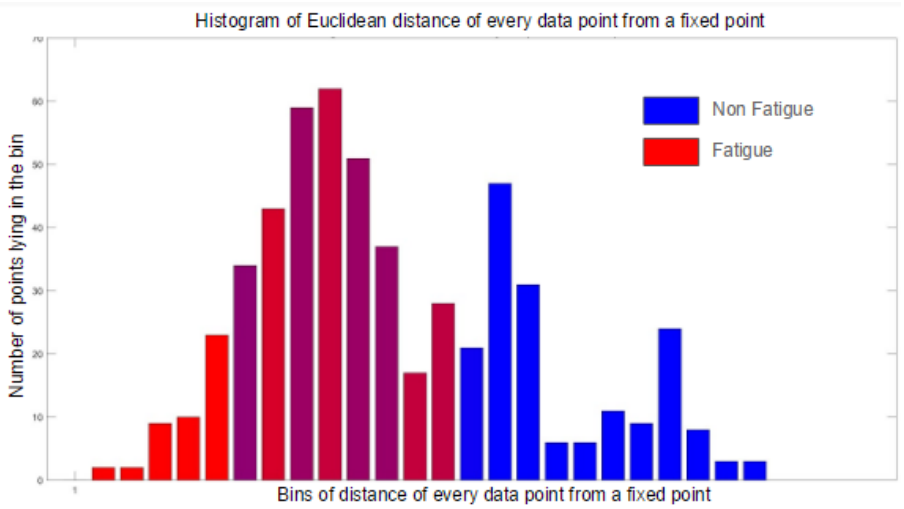

Figure 4.3: Understanding separability of the points in the feature matrix. The red mode corresponds to the class of the data point that was selected (in this case Fatigue), and the blue mode corresponds to the other class.

4.4 Training and Testing

We trained the dataset on the following classifiers:

1. Random Forest: The only hyper parameter that random forest requires is the number of decision trees to be sampled. The number of decision trees was set at 1000.

2. K- nearest neighbor: We used cross validation to determine the value of k, which is equal to 3.

3. Support Vector Machine: From the visualization, we see that the data points are sufficiently linearly separable. We used radial basis kernel function for support vector machine.

We trained and tested the classifiers for eye and head movement related features: i)individually and ii)combined.

We trained and tested on 1000 random partitions of the dataset, and obtained accuracy

for each run on test dataset. Accuracy is defined as the percentage of data points that are correctly labeled as fatigue or non-fatigue by the trained model. The average testing accuracies for 1000 random partitions of dataset for two class classifiers are shown in table 4.1, and that for three class classifiers are shown in table 4.2.

Table 4.1: Test Accuracy for 2 class classification (in %)

Features Used	**Random Forest**	**K- NN (k=3)**	**SVM**
Eye Related	95	90	89
Head Movement	78	78	72
Combined	98	97	92

Table 4.2: Test Accuracy for 3 class classification (in %)

Features Used	**Random Forest**	**K- NN (k=3)**	**SVM**
Eye Related	89	94	82
Head Movement	73	72	70
Combined	92	90	87

We observe that we obtained high accuracy for both 2 class classifier and 3 class classifier. The accuracies obtained when eye related features were used are much more than the those that were obtained when only head movement features were used. However, we see that accuracies are maximum when both features were used.

4.5 Out of Bag

In the traditional training and testing, it is possible that we may have tested on data points that were already seen during training. In the out of bag analysis, we trained on all the subjects except one, and tested on the subject that was left out of the training. This helps us in determining how the model performed on completely new data points. It also helps us in knowing which subjects conform to the trend (of not being tired initially and being tired at the end), and which subjects are exceptions.

We present in table 4.3 the accuracy obtained on each of the thirteen test subjects when they were left out of training. The analysis was done for two class classification. The parameters for each of the classifier were same as that in the previous analysis.

We observe that the average accuracy in this case is lower than what we obtained in

Table 4.3: Test Accuracy for out of bag (in %)

Test Subject	Random Forest	K- NN (k=3)	SVM
1	71.4	73.81	82
2	100	97.63	97.62
3	64.3	78.57	52.38
4	100	60	100
5	85.71	69.05	85.71
6	42.86	59.52	35.71
7	16.67	23.81	16.67
8	80.95	61.90	54.76
9	81	78.57	30.95
10	85.71	85.71	85.71
11	66.67	64.29	59.52
12	95.42	100	95.24
13	100	100	100

the analysis where training and testing was done on all subjects. We also see that the models performed very well for some of the subjects, and performed poorly on some others. We shall discuss these results in the next chapter. A plot of the results is shown in figure 4.4.

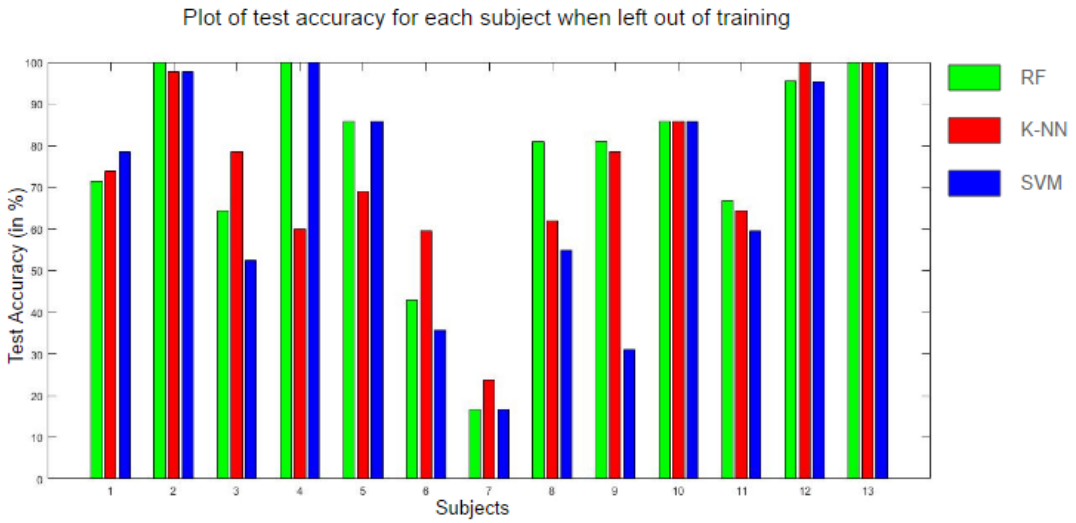

Figure 4.4: Out of bag Analysis

4.6 Early stages of Fatigue

Our aim is not only to identify intervals of fatigue and non fatigue, but also to detect early stages of fatigue, i.e. when is it that symptoms of fatigue start to kick in. To

understand that, we performed the following two experiments. In both the analysis, we used classification error as a means to detect early stages of fatigue. The classifier used was support vector machine (SVM).

1. **Testing on sliding window:** We labeled the interval between 5-25 minutes as "non-fatigue" and the interval between 145-165 minutes as "fatigue". We fixed one subject to validate the model on, and trained the model on 8 randomly selected subjects out of the remaining 12 subjects. For validation, we used a sliding window approach, where the validation data is a 21 minutes window starting at 5th minute. The window is then slid through the duration of the video. The trained model classified the points in this sliding window. Figure 4.5 shows an illustration of the how we partition the data into training and validation. The

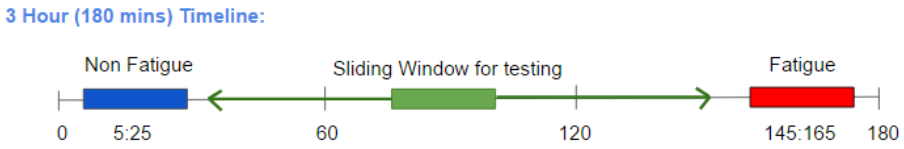

Figure 4.5: Test 1: Checking classification results on a sliding window. We have fixed windows for fatigue and non fatigue, and we train classifiers using these class labels.

results of this experiment averaged over all subjects are shown in figure 4.6. The

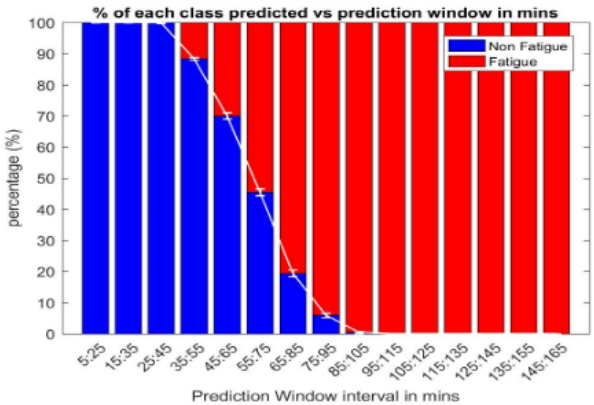

Figure 4.6: Result of the analysis averaged over all test subjects.

general trend is that the percentage of points classified as "Non-fatigue" is high

in the beginning and decreases, as we wove towards the end of the video. Likewise, the percentage of points classified as "Fatigue" is low in the beginning, and increases towards the end of the video. Figure 4.7 shows subject wise results. We observe that this trend is very clear in most of the subjects. However, some

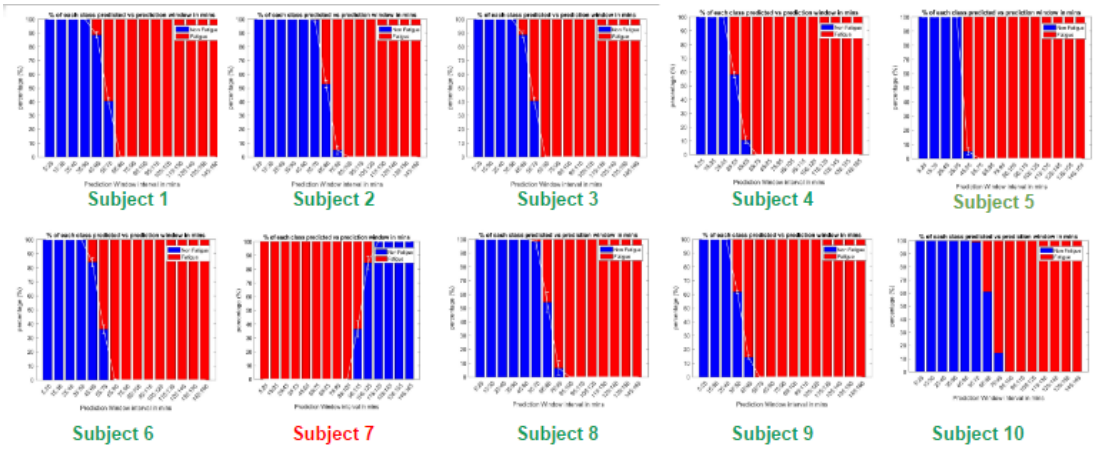

Figure 4.7: Result of the analysis for each test subject.

subjects were an exception to this trend.

2. **Sliding Labels:** In all our analysis so far, we have kept the class labels fixed. Here, we kept one of the class fixed, and had the other in a sliding window. First, we labeled the interval between 5-25 minutes as "non-fatigue". The interval for "fatigue" class was a sliding window of 21 minutes, that started at 5th minute and is slid through the duration of the video. Figure 4.8 is an illustration of such labeling. For each of the pair of fixed non-fatigue window and sliding fatigue window, we trained and tested in the same way as we did in section 4.3.

   Then, we fixed the "fatigue" class in the interval between 145-165 minutes, and used a sliding window of 21 minutes for "non-fatigue" class starting at 5th minute. We trained and tested the model for each pair of fixed fatigue window and sliding non fatigue window.

   The results for both the cases are shown in figure 4.9. We observe that the accuracy of the model is higher when the two classes are far away from each other. The accuracy keeps decreasing, when the classes move closer to each other.

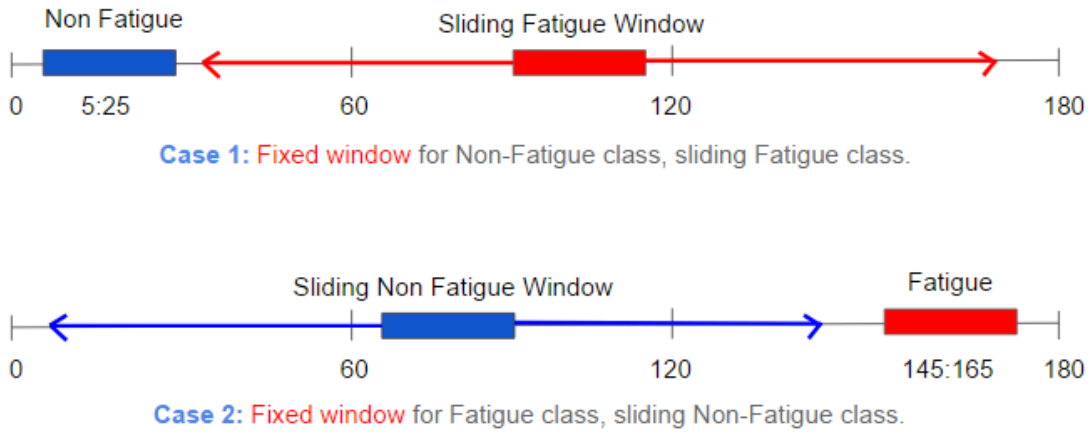

Figure 4.8: Assignment of labels for this analysis. In case 1, we kept the non fatigue class fixed, and had a sliding window for fatigue class. In case 2, we had a sliding window for non fatigue class and kept the fatigue class fixed.

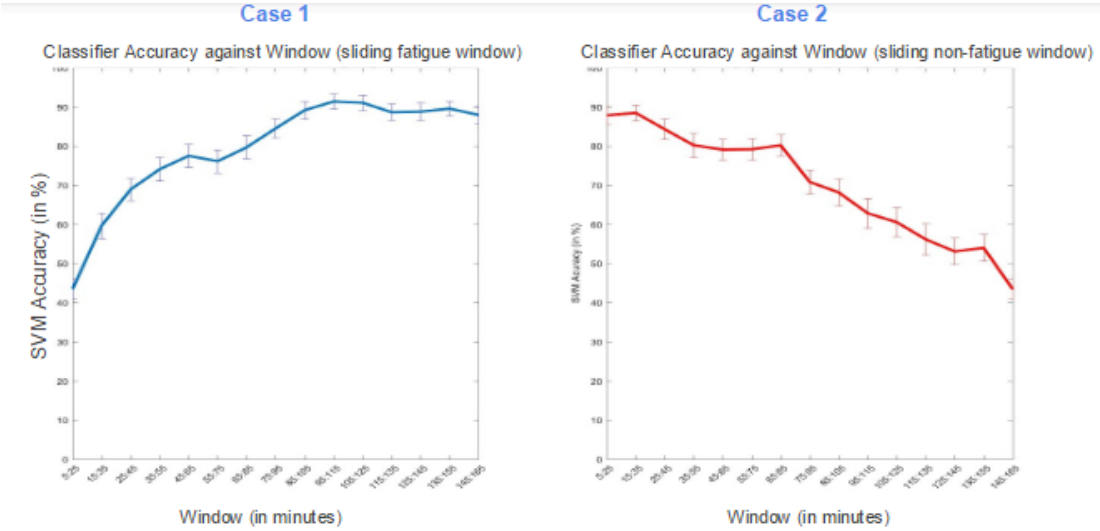

Figure 4.9: Result of the analysis for both the cases.

4.7 MEG Data Analysis

In order to correlate the camera data (visual features) with the brain data (MEG), we applied a bandpass filter to the MEG data to extract brain signals in the range 8-12 Hz, also known as alpha oscillations. We then applied the sliding window technique to compute the average power (root mean square) of the alpha band in a window of one minute. We performed a linear regression of the visual features against the alpha band power. We obtained an average ρ^2=0.59 coefficient. Subject wise ρ^2 values are shown in table 4.4.

Table 4.4: Regression ρ^2 values

Test Subject	ρ^2
1	0.669
2	0.554
3	0.603
4	0.514
5	0.397
6	0.548
7	0.59
8	0.667
9	0.567
10	0.573
11	0.443
12	0.618
13	0.678

Chapter 5

Discussion of Results and Conclusion

5.1 Model Fitting

The discussion of results can be divided into the following cases:

- **Training and Testing on all data points:** The training and test data sets were obtained by randomly partitioning the entire feature matrix. A high test accuracy was obtained, both for two class and three class classifiers for all models. The high accuracy can be explained on the basis that there is a high probability that the models saw data points from all subjects when it was trained. Hence, the test data points were not entirely new since a similar data point was already seen before. Also, the results emphasize the goodness of the features that we selected. A high accuracy can only be obtained if we have sufficient number of class discriminatory features. All of the features that we selected showed a good trend in most of the subjects. We also observed that the accuracy obtained when only eye related features were used was much more than that obtained when only head movement related features were used. This implies that eye related features are more discriminatory than the head movement features. However, using both the sets of features leads to a higher overall accuracy.

- **Out of Bag:** An out of bag analysis was done by excluding one subject from training, and testing on the subject that was left out of training. The analysis was done for two class classification only. We observed much lower accuracies, because the data points seen by the models in this case were completely new, as the subject hadn't been seen in training. We also observed that the models performed very well for some of the subjects and not so well for the others. This

is consistent with the trends that we observed in the features derived for each of the subjects, i.e. subjects which show high accuracy also show a good trend in their features, and vice versa.

- **Early stages of fatigue :** The idea was to use classification error as a means to detect early signs of fatigue. For this two analysis were performed. In this first analysis, where we test on a sliding window, we observed that the percentage of points classified as 'non-fatigue' is high in the beginning and decreases, as we move towards the end of the experiment. Simultaneously, the percentage of points classified as 'fatigue' which is low in the beginning increases towards the end of the experiment. This shows that there are clear periods of fatigue and non fatigue, with both having different values for visual features that were extracted. If it wasn't such a case, we would have expected the accuracy of classifier to stay constant throughout the duration of the video (signifying that there are no clear demarcations between fatigue and non fatigue). We also observed that there is an interval between 50-60 minutes, between which there is a sharp change in the labels from non fatigue to fatigue. This marks the early stages of fatigue. As in the previous analysis, the trend is absent in some of the subjects for which there were no trends in the features either.

  In the second analysis, we fixed the labels for one class, and the labels for the other class is slid through a sliding window of 21 minutes. We observed that the test accuracy of the classifier starts from 50% when both the labels are assigned to the same window. The accuracy kept increasing when the labels for one class were moved away from the labels of the fixed class. This analysis also signifies that there are clear periods of fatigue and non fatigue. Like in the previous analysis, if it wasn't the case, then changing the labels would have had no effect on the classifier accuracy.

5.2 Integration with MEG

We observed an average regression coefficient of 0.6 for alpha band power of MEG signal vs. the visual features. While the coefficient isn't too high, there is a significant correlation between the brain data (MEG) and visual features, thereby validating the features that we selected.

5.3 Conclusion

Our studies resulted in a set of 8 eye movement related and 6 head movement related features that we can use to determine intervals of fatigue or not. We also showed how we can use classification error as a means to detect early stages of fatigue, which is novel and hasn't been done before. Finally, we used the brain data (MEG) as a ground truth to validate our model. Although our results are promising, much work remains to be done in order to meaningfully use these studies. Specifically, improvements can be made in the following directions:

1. **Unsupervised Feature Learning:** Instead of heuristically determining features, like we did in our work, one can use deep learning to let the system decide on its own what features to use for classification. The challenges include getting enough labeled data for training (deep nets require enormous training data), and ensuring that there is a spatial or temporal relationship in the data.

2. **Online Fatigue Detection:** Our results hold promise in developing a fully automated system that detects fatigue in real time. It would be much more beneficial to alert the subject when he is starting to experience the symptoms of fatigue, rather than alerting him when he is already fatigued.

3. **Using Game Related Statistics:** Behavioral features such as reaction time can serve as another indicator for fatigue, and can be used to further validate our model.

4. **Source Localization:** One can try to find out alpha signal from which part of the brain contributes to fatigue.

Bibliography

[1] W. Kong, L. Zhou, Y. Wang, J. Zhang, J. Liu, and S. Gao, "A system of driving fatigue detection based on machine vision and its application on smart device," *Journal of Sensors*, vol. 2015, 2015.

[2] A. Singh and J. Kaur, "Driver fatigue detection using machine vision approach," in *Advance Computing Conference (IACC), 2013 IEEE 3rd International*. IEEE, 2013, pp. 645–650.

[3] M. Sacco and R. A. Farrugia, "Driver fatigue monitoring system using support vector machines," in *Communications Control and Signal Processing (ISCCSP), 2012 5th International Symposium on*. IEEE, 2012, pp. 1–5.

[4] Z. Zhang, S. Shu, S. Liu, Z. Guo, Y. Wu, X. Bao, J. Zheng, and H. Ma, "Activated brain areas during simple and complex mental calculation–a functional mri study," *Sheng li xue bao:[Acta physiologica Sinica]*, vol. 60, no. 4, pp. 504–510, 2008.

[5] P. S. Peopie, D. Dinges, G. Maislin *et al.*, "Evaluation of techniques for ocular measurement as an index of fatigue and the basis for alertness management," 1998.

[6] H. Wang, L. Zhou, and Y. Ying, "A novel approach for real time eye state detection in fatigue awareness system," in *Robotics Automation and Mechatronics (RAM), 2010 IEEE Conference on*. IEEE, 2010, pp. 528–532.

[7] K.-Q. Shen, X.-P. Li, C.-J. Ong, S.-Y. Shao, and E. P. Wilder-Smith, "Eeg-based mental fatigue measurement using multi-class support vector machines with confidence estimate," *Clinical Neurophysiology*, vol. 119, no. 7, pp. 1524–1533, 2008.

[8] F. Gharagozlou, G. N. Saraji, A. Mazloumi, A. Nahvi, A. M. Nasrabadi, A. R.

Foroushani, A. A. Kheradmand, M. Ashouri, and M. Samavati, "Detecting driver mental fatigue based on eeg alpha power changes during simulated driving," *Iranian journal of public health*, vol. 44, no. 12, p. 1693, 2015.

[9] X. Li and B. H. Duc, "Functional neuroimaging of circadian fatigue," *International Journal of Computer Applications in Technology*, vol. 45, no. 2-3, pp. 156–162, 2012.

[10] L. J. Trejo, K. Kubitz, R. Rosipal, R. L. Kochavi, and L. D. Montgomery, "Eeg-based estimation and classification of mental fatigue," *Psychology*, vol. 6, no. 05, p. 572, 2015.

[11] X. Yu, Z. Lin, J. Brandt, and D. N. Metaxas, "Consensus of regression for occlusion-robust facial feature localization," in *European Conference on Computer Vision*. Springer, 2014, pp. 105–118.

[12] M.-H. Sigari, M.-R. Pourshahabi, M. Soryani, and M. Fathy, "A review on driver face monitoring systems for fatigue and distraction detection," 2014.

[13] T.-P. Jung, S. Makeig, C. Humphries, T.-W. Lee, M. J. Mckeown, V. Iragui, and T. J. Sejnowski, "Removing electroencephalographic artifacts by blind source separation," *Psychophysiology*, vol. 37, no. 2, pp. 163–178, 2000.

[14] Q. Ji and X. Yang, "Real-time eye, gaze, and face pose tracking for monitoring driver vigilance," *Real-Time Imaging*, vol. 8, no. 5, pp. 357–377, 2002.

[15] Q. Wang, J. Yang, M. Ren, and Y. Zheng, "Driver fatigue detection: a survey," in *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, vol. 2. IEEE, 2006, pp. 8587–8591.

[16] D. F. Dinges, M. M. Mallis, G. Maislin, I. Powell *et al.*, "Evaluation of techniques for ocular measurement as an index of fatigue and the basis for alertness management," Tech. Rep., 1998.

[17] M. A. Schier, "Changes in eeg alpha power during simulated driving: a demonstration," *International Journal of Psychophysiology*, vol. 37, no. 2, pp. 155–162, 2000.

[18] W. Klimesch, M. Doppelmayr, H. Russegger, T. Pachinger, and J. Schwaiger, "Induced alpha band power changes in the human eeg and attention," *Neuroscience letters*, vol. 244, no. 2, pp. 73–76, 1998.

[19] N. Pattyn, X. Neyt, D. Henderickx, and E. Soetens, "Psychophysiological investigation of vigilance decrement: boredom or cognitive fatigue?" *Physiology & Behavior*, vol. 93, no. 1, pp. 369–378, 2008.

[20] L. J. Trejo, R. Kochavi, K. Kubitz, L. D. Montgomery, R. Rosipal, and B. Matthews, "Eeg-based estimation of cognitive fatigue," in *Proceedings of SPIE*, vol. 5797, 2005, pp. 105–115.

[21] R. Vigário, J. Sarela, V. Jousmiki, M. Hamalainen, and E. Oja, "Independent component approach to the analysis of eeg and meg recordings," *IEEE transactions on biomedical engineering*, vol. 47, no. 5, pp. 589–593, 2000.

[22] K. Glass, G. Frishkoff, R. Frank, C. Davey, J. Dien, A. Malony, and D. Tucker, "A framework for evaluating ica methods of artifact removal from multichannel eeg," *Independent component analysis and blind signal separation*, pp. 1033–1040, 2004.

[23] R. M. Chapman, R. Ilmoniemi, S. Barbanera, and G. Romani, "Selective localization of alpha brain activity with neuromagnetic measurements," *Electroencephalography and clinical neurophysiology*, vol. 58, no. 6, pp. 569–572, 1984.

[24] Z. Zhang and J. Zhang, "A new real-time eye tracking for driver fatigue detection," in *ITS Telecommunications Proceedings, 2006 6th International Conference on.* IEEE, 2006, pp. 8–11.

[25] Y. Wang, X. Liu, Y. Zhang, Z. Zhu, D. Liu, and J. Sun, "Driving fatigue detection

based on eeg signal," in *Instrumentation and Measurement, Computer, Communication and Control (IMCCC), 2015 Fifth International Conference on.* IEEE, 2015, pp. 715–718.