No, that’s not what I mean
Recognition of Affective Prosody

Sumeet Bhalla, Aditya Chayapathy, Kevin Thomas, Siddhant Kanwar, Abhijith Shreesh

sbhalla5@asu.edu, achayapa@asu.edu, kthoma46@asu.edu, skanwar2@asu.edu, ashreesh@asu.edu

1212927233, 1213050538, 1213122571, 1210688216, 1213204276

Abstract— *In social neuroscience, it is essential to separate brain processes within human brain. Event-related potential is a crucial instrument that helps in evaluating cognitive functions. To observe attention, focus, or other cognitive processes, P300 is one of the most important components for assessment. We initially collected brain signal data of three subjects who were told that they will be showed a particular image but in actuality were showed a different image. Then we cleaned the data by using various filters. We used machine learning algorithms such as k-nearest neighbors algorithm and support vector machine to classify the brain signals into expected and unexpected classes. We later compared the accuracy of SVM and KNN algorithms.*

Keywords— Machine Learning, KNN, SVM, Deep Learning, Brain Waves, Neural Network, Prediction, MATLAB, Regression, Correlation, N300, P300, Low pass filter, High pass filter, Signal Processing

I. Introduction

It is extremely important to separate different brain processes in social neuroscience. In the current research, the goal is to test whether recent findings of negative event-related potential, a measured brain response that is the direct result of a specific sensory event, in response to the unexpected gaze are unique to eye gaze stimuli[1]. As humans, we tend to predict another person’s intentions based on some arbitrary event performed by the other person. Based on that event, we expect them to behave in a certain way.

In this project, we are trying to classify brain signals into expected and unexpected classes using machine learning algorithms such as KNN and SVM. The dataset which was provided to us includes three subjects and multiple trials for each subject. We utilized low pass filter and Discrete Wavelet Transformation (DWT) which decreases signals with frequencies higher than the cutoff frequency. In addition, we utilized signal processing methods such as P300 and N300 to represent the data points in a new vector space of peaks. Once we perform classification using P300 and N300 values using machine learning models, we compared the accuracy of SVM and KNN algorithms.

II. Project Setup

- Hardware Requirements:
  - Make and Model - MacBook Pro (2017 model)
  - OS - MacOS (Mojave)
  - Processor - 2.3 GHz Intel Core i5
  - RAM - 8 GB
- Software Requirements:
  - MATLAB
  - Anaconda Python 3.7

III. Implementation

The data provided consisted of brain readings of three subjects. Each of these subjects were told that they will be shown an image of an animal or an object and was shown one of the following:

- Same animal or object as mentioned. We call these cases the **Expected** class.

- Some other animal or object completely different from what was told before. We call these cases the **Unexpected** class.

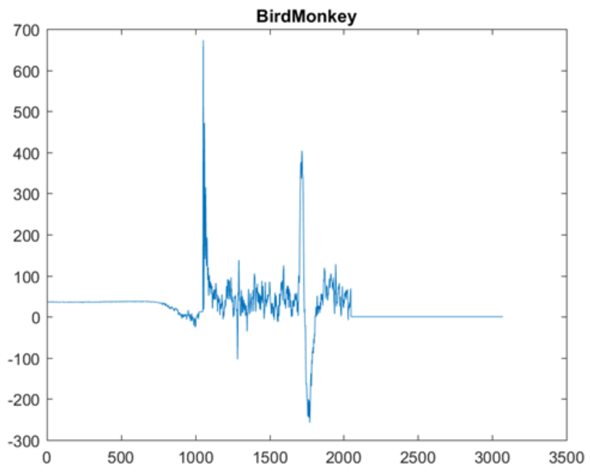
We grouped the expected class data of Subject 1 and Subject 2 and used it as training data for the Expected class of our classifier models. We grouped the unexpected class data of Subject 2 and Subject 3 and used it as training data for the Unexpected class of our classifier models. We took both the Expected and Unexpected class data of Subject 1 and used it as Testing data for our classifier models. We used three different classification techniques:

- Support Vector Machines (SVM) classification
- K-Nearest Neighbor (KNN) classification
- Decision Tree (DT) classification

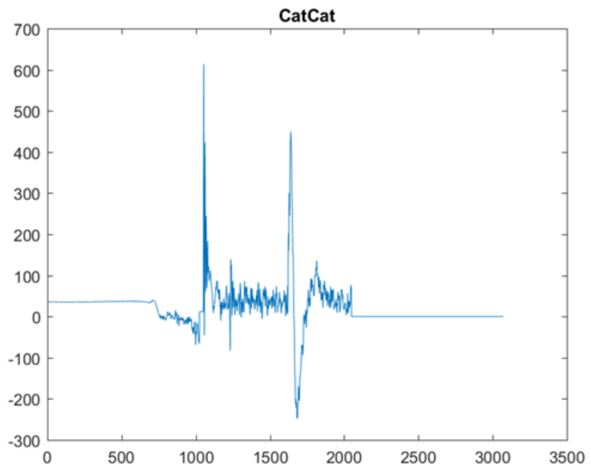
We coded the entire part on MATLAB. We have three different files which predicts the class labels of the testing data. The three files are segregated based on the type of input fed into the classification models. It is segregated as follows:

- ***ExpectedClassification_RawData***:
  This program feeds the raw data which was given to us into the three classification models for prediction.

Here is an example of a graph from the Unexpected class data:



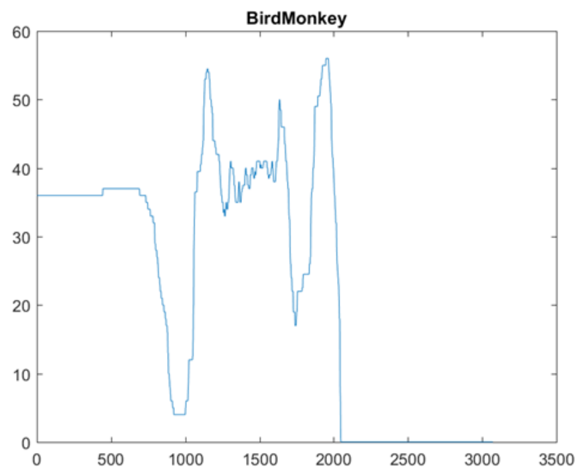
Here is an example of a graph from the Expected class data:



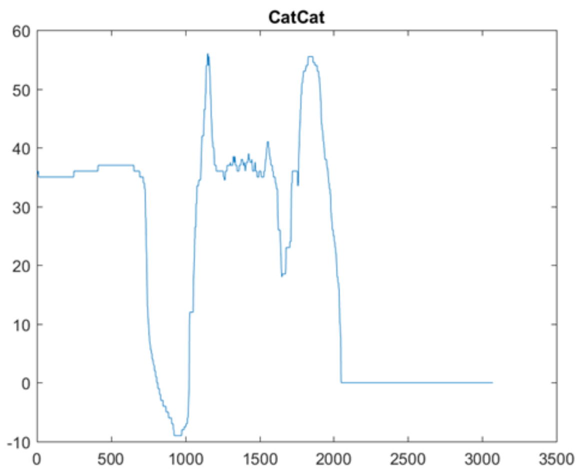
The point where the data starts varying is the point of stimulus. We trim the data before the 500th data point and data after 2250th data point.

- ***ExpectedClassification_MovMedian***:
  This program feeds the moving median over 200 values of the raw data which was given to us into the three classification models for prediction. This acts like a low pass filter thereby removing high magnitude noise from the data[2].

Here is an example of a graph from the Unexpected class data after applying MovMedian() on it:

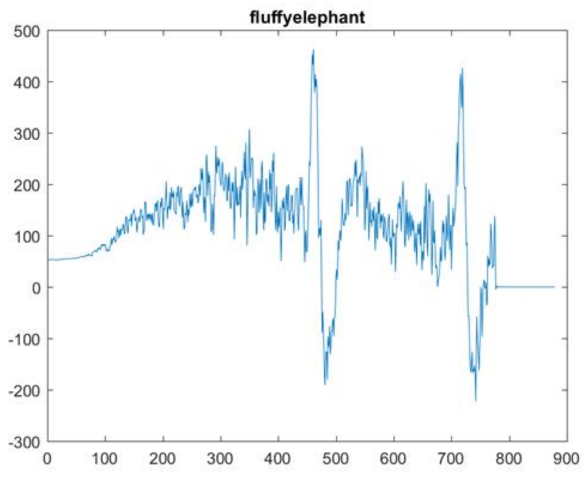


Here is an example of a graph from the Expected class data after applying MovMedian() on it:

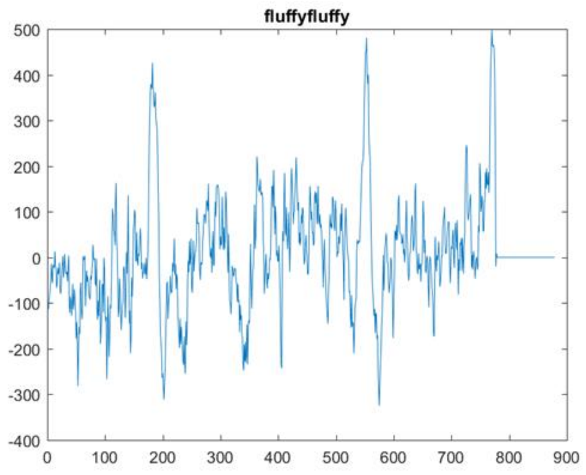


• ***ExpectedClassification_DWT***: this program feeds the DWT of the raw data which was given to us into the three classification models for prediction. This is done to provide better features which can be used for classification[3].

Here is an example of a graph from the Unexpected class data after applying DWT on it:



Here is an example of a graph from the Expected class data after applying DWT on it:



In each of these files, the data is extracted from the .mat files provided, trimmed to start from the stimuli, converted into a form which is acceptable to the *fitcsvm*()[4], *fitcknn*()[5] and *fitcdt*()[6] methods. These are the methods provided by MATLAB which performs SVM, KNN and DT classification respectively. The training data is provided to these methods to train and the testing data is provided into the *predict*() method along with the classification model to predict the class labels. The results are provided in the next section.

Accuracy

The accuracy of the classification models using machine learning techniques are as follows:

Accuracy Table

	Raw Data.	*MovMedian Data*	*DWT Data*
SVM	47.06%	23.53%	76.47%
KNN	76.47%	58.82%	76.47%
DT	52.92%	35.29%	76.47%

We can see that applying DWT on the raw data improved the accuracy of the classification model.

IV. Data Preprocessing

The model is constructed with the goal of classifying inputs based on user responses/reflexes. The dataset for this project was data collected from the three subjects and stored in the form of '.mat' files. Since each '.mat' is a time series representation of a user's response to stimulus, all the time series data is stacked up in the form of a table to construct the dataset. The dataset is divided into two classes, 'expected' and 'unexpected'. For example, 'CatCat' would be in the 'expected' class and 'CatDog' would be in the 'unexpected' class. Data from Subjects 1 & 2 were used for training and data from Subject 3 was used for testing.

V. Signal Processing

For classification using signal processing, we tried a few methods for representing the data[7]:

A. *N300 peak average time window detection*

We use the training data and select a range of points from 350 to 2250, find the maximum negative peak (N300) in this range. We then identify the point at which the maximum negative peak as detected. This value is stored for all data points and the training points are separated based on class. Next, these values averaged for each class to get the average time (window) for negative peak detection for both the classes - 'expected' and 'unexpected'. For testing, we computed the maximum negative peak point for each sample and assigned it to the class which has the nearest average negative peak point.

B. *N300 peak detection using K-nearest neighbor*

For this task, we used a peak detection algorithm to get all the negative peaks of each signal (sample) and represented each sample in this new vector space of peak points. Both the training and testing samples were transformed into this vector space. We then classified the samples in this new vector using nearest neighbor algorithm. Multiple values of k were tested ranging from 2 to 5, but k= 3 produced the most accurate results.

C. *N300 peak detection using SVM:*

For this task, we used a peak detection algorithm to get all the negative peaks of each signal (sample) and represented each sample in this new vector space of peak points. Both the training and testing samples were transformed into this vector space. We then classified the samples in this new vector using Support Vector Machines. Different models were created using 'linear', 'poly' and 'rbf' kernels but the 'rbf' kernel was the most accurate one.

D. P300 peak detection using KNN:

We detect all the positive peaks in each data point and represent the data points as vectors of these peaks. The training and testing samples are represented in this vector space. We then classified the samples in this new vector using nearest neighbor algorithm. Multiple values of k were tested ranging from 2 to 5, but k= 3 produced the most accurate results.

E. P300 peak detection using SVM:

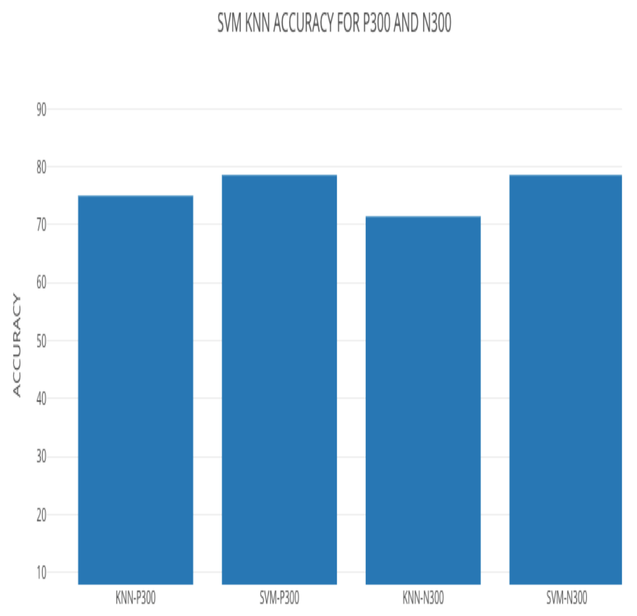
We detect all the positive peaks in each data point and represent the data points as vectors of these peaks. The training and testing samples are represented in this vector space. Support Vector Machines was used to classify these samples in the vector space of peaks. We tested the model with ‘linear’, ‘poly’ and ‘rbf’ kernels, with ‘rbf’ kernel producing most promising results

VI. RESULTS

The three models that we developed produced results with fair accuracy. The errors in the classification can be attributed to the dearth of training data. We got the following accuracies:

N300 accuracies:

- peak average time window detection = 78.51%
- peak detection using k-nearest neighbor = 71.42%
- peak detection using SVM = 78.51%
- P300 accuracies:
    - peak detection using k-nearest neighbor = 75%
    - peak detection using SVM = 78.51%



VII. COMPLETION OF TASKS

Serial	Task Name	Assignee
1	Obtaining data from IMPACT Lab	Aditya, Abhijith, Sumeet, Siddhant, Kevin
2	Understanding data format	Aditya, Abhijith, Sumeet, Siddhant, Kevin
3	Understanding stimuli	Aditya, Abhijith, Sumeet, Siddhant, Kevin
4	Segmenting data using stimuli	Kevin, Abhijith
5	Scientific Design - Finding stimulus (incongruity)	Kevin, Sumeet
6	Scientific Design - Low pass filter analysis	Kevin, Sumeet
7	Scientific Design - Separating the data into training and testing set	Kevin, Sumeet
8	Scientific Design - Machine learning analysis	Siddhant, Sumeet
9	Scientific Design - Accuracy analysis	Siddhant
10	Signal Processing - Data preprocessing	Kevin, Abhijith
11	Signal Processing - Designging algorithm to detect peaks	Sumeet, Aditya, Siddhant
12	Signal Processing - Implementing the algorithm to detect peaks	Siddhant, Abhijith
13	Signal Processing - Performance evaluation	Aditya, Siddhant
14	Peak Predicting - Feature Extraction	Abhijith, Aditya
15	Peak Predicting - Machine Learning Analysis	Abhijith, Aditya
16	Prediction - Performance evaluation	Abhijith, Aditya
17	Report and Documentation	Aditya, Abhijith, Sumeet, Siddhant, Kevin
18	Developing UI to collect data	N/A
19	Collecting user’s brain signal and collecting their feedback	N/A

20	Annotating brain data	N/A

VIII. LIMITATIONS

The following are the limitations of the application that we have developed:

- The machine learning analysis was performed on a limited dataset. This could have resulted in overfitting the model. In order to obtain high accuracies, we will need to train the machine learning models on a much larger dataset.
- In order to get detect the N300 and P300 peaks, we need to ensure that the data collected from the Neurosky sensor is noise free. Owing to a small fraction of inaccuracies from the Neurosky sensor, we are expecting few false positives in our results.
- For the application to detect the peaks in real-time, the subject is expected to wear the Neurosky sensor for continuous data collection. This is an infeasible ask as it is not very convenient to always wear the sensor. It is also an expensive device that must be handled with care.

IX. CONCLUSION

The raw data provided has a lot of noise in it so we had to figure out ways to remove this noise while doing our peak detections. We used a Low pass filter that passes signals with a frequency lower than a selected cutoff frequency and attenuates signals with frequencies higher than the cutoff frequency. We have also used Discrete Wavelet Transformation (DWT) for reducing noise as well. Wavelets are often used to denoise two dimensional signals. While classifying expected and unexpected data we figured out that the DWT's accuracy was better than the Low pass filter accuracy.

N300 is a recent finding in the context of semantic congruity and expectancy. The latency is usually interpreted as the speed of stimulus classification resulting from discrimination of one event from another. Shorter latencies indicate superior mental performance relative to longer latencies. P3 amplitude seems to reflect stimulus information such that greater attention produces larger P3 waves. The subject is instructed to respond to the infrequent or target stimulus and not to the frequently presented or standard stimulus. Reduced P300 amplitude is an indicator of the broad neurobiological vulnerability that underlies disorders within the externalizing spectrum. [8]. Once we performed classification using these P300 and N300 values using machine learning models we found that we got better accuracy using SVM over KNN.

X. REFERENCES

[1] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3682432/
[2] https://www.mathworks.com/help/matlab/ref/movmedian.html
[3] https://www.mathworks.com/help/wavelet/ref/dwt.html
[4] https://www.mathworks.com/help/stats/fitcsvm.html
[5] https://www.mathworks.com/help/stats/fitcknn.html
[6] https://www.mathworks.com/help/stats/fitcdt.html
[7] https://en.wikipedia.org/wiki/P300_(neuroscience)
[8] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3016705/