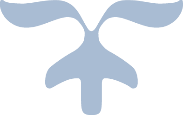


A PROTOTYPE DEVICE FOR FACIAL ELECTROMYOGRAPHY SIGNAL COLLECTION

Final Report



**EMG Device Design Team**

MAY 13, 2021

SENIOR DESIGN TEAM 18

CHUONG NGUYEN (TEAM LEADER) | TOAN VU | JOHNATHAN NGUYEN

UNIVERSITY OF HOUSTON

SPONSOR: DR. ROSE T. FAGHIH

ECE DEPARTMENT OF CULLEN COLLEGE OF ENGINEERING

FACILITATOR: DR. STEVEN PEI

Chuong Nguyen | Toan Vu | Johnathan Nguyen

May 13, 2021

Dr. Leonard P. Trombetta

University of Houston

4800 Calhoun Rd, Houston,

TX 77004, United States

Dear Dr. Leonard P. Trombetta,

We are the senior engineering team that was assigned to your inquiry to design a device that can help improve the current statistics of mental illnesses. Despite numerous constraints we encountered during the last two semesters, we were able to accomplish the prototype device that was assigned. However, we were not able to finish the 3D-printed prototype headband due to the winter storm. The major deliverables which are the EMG signal collection and visualization, the machine learning model and the mobile application were successfully completed. The machine learning model has a prediction accuracy up to 90%, however when integrated with the mobile application, our accuracy percentage drops to about 70%. Possible justifications for this significant drop are discussed in *Section 10* of the report. Some recommendations for future work are also suggested in *Section 11* in case this project is to be continued by another group.

If you have any questions, please do not hesitate to contact us.

Sincerely,

EMG Device Design Team

CC: Dr. Steven Pei

1. **Background**

Mental disorders are becoming one of the most common illnesses in the United States. In a recent research, approximately 70 percent of American adults have experienced a traumatic event at least once in their lives. An estimated 20 percent of them will then later develop post-traumatic stress disorder (PTSD) [1]. In addition to this, about 7.1 percent of the U.S population aged over 18 suffered from major depressive disorders in 2017, and that is approximately 17.3 million American adults [2].

Unlike other diseases, mental disorders are harder to diagnose since their symptoms are not as visibly clear and treatments for such illnesses take a lot of time, effort and money. To address these issues, we proposed to design a device that can collect physiological data and send it to a mobile application to analyze and predict the patient’s emotions.

Compared to psychotherapy (counselling), which explores the patients’ thoughts, feelings and behaviors through talk sessions, our proposed system can provide medical professionals with real and live data from the mobile application. This would help psychiatrists make more accurate diagnosis and come up with better treatment courses for the patients.

1. **Purpose**

The purpose of the project is to develop a prototype for a wearable device that can detect users’ emotions. The implementation of the device will be composed of three major parts: collecting electromyography (EMG) signals from three sensors, storing and analyzing the collected data utilizing machine learning, and displaying the prediction on a smartphone application.

1. **Patents:**
   1. **US 8214214 B2:**

This project aims to detect emotions by analyzing the user’s prosody (rhythm and intonation of voice). The voice data is then extracted and analyzed using a machine learning algorithm. The main difference between this project and ours is the input data. Our project uses EMG signals of the facial area to indicate the user’s emotion whereas the patent’s project uses prosody signals.

* 1. **US 9600715 B2:**

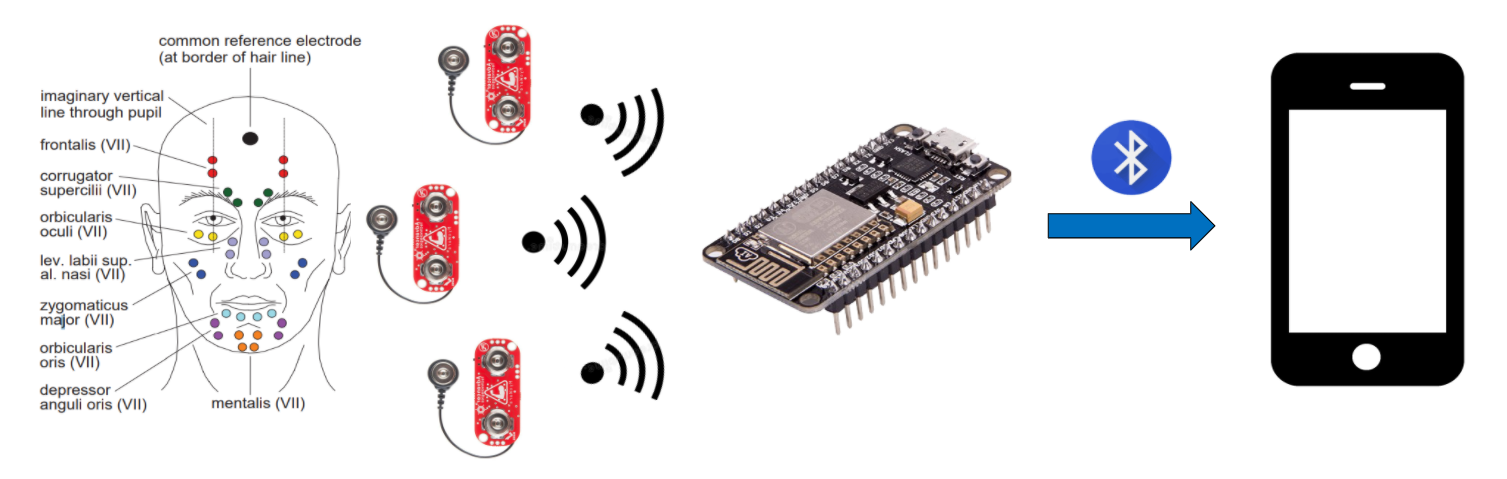
This project aims to detect emotions by analyzing raw image data (pixels) of a user’s facial expression. The data is then extracted into 78 landmark points to reduce the data volume and protect the privacy of the subjects. The difference between this project and ours is the input data. This project uses image pixels to analyze and predict emotions whereas our project uses EMG signals.

* 1. **US 8170656 B2:**

This project aims to detect gestures of parts of the body by analyzing EMG signals from sensors attached to those parts. The device can be in the form of a wristband or an armband with embedded EMG sensors. The process of this project is similar to ours in terms of analyzing EMG signals to predict an outcome. However, the purpose of our project is to predict emotions based on those signals, whereas the patent’s project aims to predict the gestures of parts of the body.

**Patentability:** Even though the purpose of the project is different (gestures vs emotion detection), the process of our project is quite similar to patent (c). Based on the non-obviousness criteria of patentability, there is nothing in our project that needs to be patented.

1. **Overview Diagram**



*Figure 1: Project Overview Diagram*

1. **Deliverables**

For this project, we have three deliverables. The first deliverable is microcontroller programing. In this deliverable, we programmed a microcontroller to uniformly collect the EMG signals from the muscle sensors and send the data to a smartphone through Bluetooth. The programming was done through the Arduino IDE.

The second deliverable is the machine learning model. We used machine learning to process the EMG data collected through the sensors and predict an emotion with this data. A training dataset was collected for this deliverable. Tensorflow was used, and the machine learning model was based on recurrent neural network and long-short term memory.

The last deliverable is the iOS mobile application. This app displays real-time EMG data from the sensors, the emotion prediction with an emoji representing the emotion, and percentages of the machine learning emotion prediction.

1. **Engineering Specifications**

*Table 1: Engineering Specifications*

|  |  |
| --- | --- |
| **Bluetooth Connectivity** | Bluetooth Low Energy (BLE)  Range: 2-3 meters |
| **Sample frequency** | 1 [KHz] |
| **Sensors** | 3 Myoware Muscle Sensors |
| **Power Consumption** | 127 [mAh] |
| **Machine Learning** | TensorFlow, Recurrent Neural Network, Long-Short Term Memory |
| **Machine Learning Model** | 90% Accuracy |
| **Emotion Prediction** | 70% Accuracy |

The ESP-WROOM-32 was chosen as the microcontroller to use for many reasons. One reason is the ADC (Analog to Digital Converter). The device needed 3 ADC channels for the 3 sensors and 1[KHz] sampling frequency. The ESP32 has 18 ADC channels and a sampling frequency of 6 [KHz], which fit both criteria. The other option was a Raspberry Pi Zero. This did not come with an ADC, so an extension is required for it, and the sampling frequency is 240 [Hz]. The difference between the two was obvious so the ESP32 was chosen. It also is compatible with the Arduino IDE, which our team was familiar with.

The ESP-WROOM-32 is also capable of utilizing Bluetooth Low Energy (BLE). This has a range of around 2-3 meters and will reduce power consumption of the device.

The MyoWare muscle sensor was chosen to measure the EMG signals due to its small size, wearable design, and being designed specifically for microcontrollers. As mentioned before, it is 52.3 x 20.7 [mm]. Three was the number chosen for the number of sensors because too few sensors would make the emotion prediction less accurate, and too many sensors would clutter the face.

The Myoware sensor consumes 9 [mA] and the ESP32 consumes 100[mA] when completely active. The total power consumption is 127 [mA]. With BLE, this power consumption will be lower.

The datasheet for the ESP32 and Myoware muscle sensor can be found here [3],[4].

For the machine learning, TensorFlow was used. This is a Python library specialized in machine learning. The machine learning model is based on recurrent neural network and long-short term memory structure, as they can process sequences of continuous data, which our EMG data is.

The machine learning model has around a 90% accuracy. More details into this performance will be discussed in *Section 10*.

The emotion prediction on the app was around 70%. The app was able to predict the emotion happy and sad consistently, but with the angry emotion, the results were varying. This is probably due to the EMG signal associated with the angry face having high intensity in the beginning, then falling off over time.

1. **Constraints**

Though we expected to create a prototype that will work on every person, many constraints have emerged. The biggest constraint on this project was the Covid-19 pandemic. Covid-19 has made it impractical to gather a training dataset that includes multiple subjects. This has limited our project into only having one test subject to obtain a training dataset for. Our device will only be accurate for that test subject.

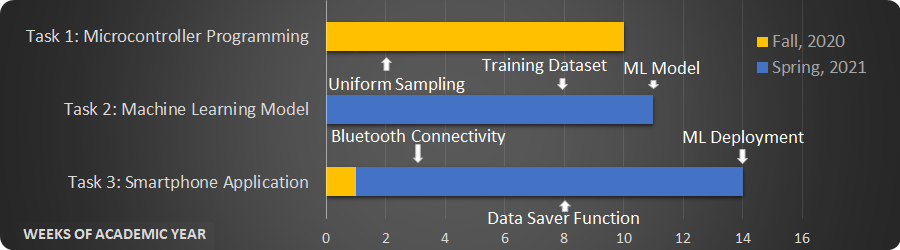
Other constraints that have impacted our project are locations of team members and budget. Due to Covid, a member in our team has decided it would be best to stay in Vietnam. This makes it impossible to all meet up to work on the project. Also, due to a limited budget, not all members will have the equipment to work on certain parts of the project. For example, due to the cost of the Myoware muscle sensor, it is not practical for the team member in Vietnam to have one.

This project was meant for a team of four, but there are only three members working on this project. The team has decided to drop one of the features for better labor force allocation.

Finally, there was a limitation in the hardware involved in the project as our team members’ personal computers did not have the required computing power. Even though the team was able to overcome this constraint by utilizing cloud computing for better computability, the limitation obviously caused some delays in the project.

1. **Schedule**

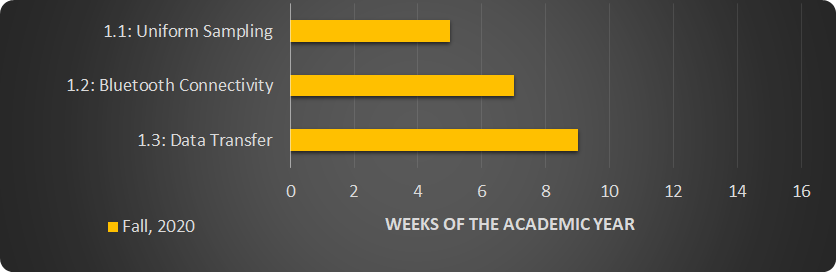
Figure 2 shows the schedule for the project, which is divided into 3 main tasks: microcontroller programming, machine learning model, and smartphone application. Each task has milestones to indicate major steps in the project. Each task will then be split into sub-task to ensure the requirements, functionality, and compatibility are met.

**

*Figure 2: Project Schedule*

* 1. **Task 1: Microcontroller Programming**

This task is divided into three subtasks, which are uniform sampling, bluetooth connectivity, and data transfer as shown in Figure 3.

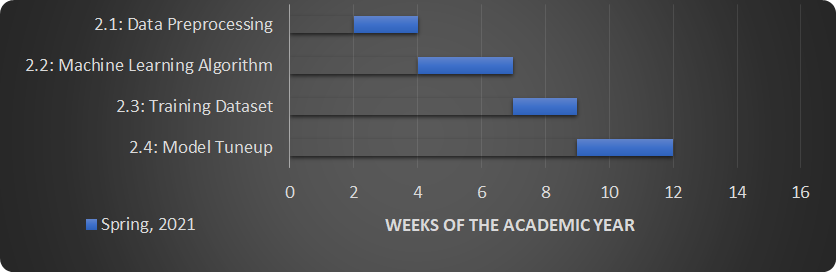


*Figure 3: Sub-task Schedule for Microcontroller Programming*

The microcontroller is programmed to create a connection from the three Myoware sensors, to the microcontroller itself, and to a smartphone. An interrupt is used in the microcontroller’s embedded system to uniformly sample the data collected from the sensors at the defined frequency of 1 [KHz]. Using Bluetooth Low Energy (BLE), the connection between the microcontroller and the smartphone is expected to work stably within a working range of 2 to 3 meters for the data to be uninterruptedly transferred to the smartphone without a loss.

* 1. **Task 2: Machine Learning Model**

As can be seen in figure 4, the subtasks for the machine learning model include data pre-processing, Machine Learning algorithm, training dataset, and model tune-up.



*Figure 4: Sub-task Schedule for Machine Learning Model*

The process of pre-processing data is based on the open-source EMG dataset for hand gesture recognition [5], due to the similarities between the two projects. The data is read from a “comma-separated values”, or .csv, file, which is split into columns of timestep, activity, and electrodes. A depiction of the data format can be found in the Appendix. The dataset is divided into three separate sub-datasets for the Machine Learning model: the train dataset, the validation dataset, and the test dataset. The Machine Learning algorithm will convert these sub-datasets into tensors (a Tensorflow data form), which will be run through an artificial recurrent neural network (RNN).

The neural network consists of layers of Long-Short Term Memory architecture to learn information from the corresponding hand activity labels. The Machine Learning model will be trained with the train and validation datasets and tested with the test dataset.

The Machine Learning algorithm has adopted functions from the two open source Python libraries: Numpy and Pandas. Because these two libraries are specialized for data analysis, the upgraded algorithm has resulted in a more uniform and efficient dataset. Because of the nature of Neural Networks that models tend to memorize the training data set, which is defined as overfitting, rather than actually learn to predict, three regularization techniques are applied to the model to prevent this, which are time window, dropout, and data shuffling. While time-window is to make a dataset with overlapped time steps between two data batches; dropout, which is to randomly drop some training batches after a period of time, and data shuffling, which is to randomize the order of training batches, are used to increase generalizations.

The dataset is collected with 2,033,500 data points, which are categorized into three emotions: sadness, happiness, and anger. These data points are split into groups of 512 data points, called samples, with 462 overlapping data points. The whole dataset is divided into 3 sub-datasets with 70% for the train dataset, 15% for the validation dataset, and 15% for the test dataset.

The machine learning model will be constantly trained and tuned-up to achieve the expected accuracy of 90% at the training/validating stage and testing stage.

* 1. **Task 3: Mobile Application**

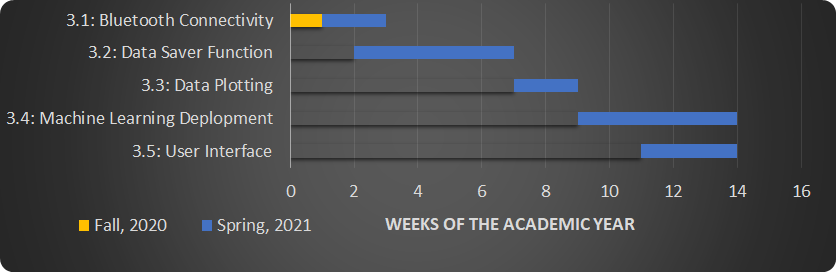
Figure 5 shows the subtasks of the mobile application objective, which include bluetooth connectivity, data saver function, data plotting, machine learning deployment, and user interface. The bluetooth connectivity on the smartphone application is configured to sync up with the Bluetooth Low Energy (BLE) connection from the microcontroller.

The data saver function is implemented to acquire data in two different ways. While making the dataset to train the machine learning model, the function randomly shows an illustration of one of the three basic emotions and writes accordingly into a comma-separated-value file. During the emotion prediction process, the function will write the signal data corresponding to the predicted emotions.

The collected data will be plotted in a graph in real time. It is noticed that, while the values read from the three sensors range from 0 to 4095, the written data is scaled to range from -1 to 1 for the purpose of collecting the training dataset for the machine learning model. An illustration of the plot can be found in the Appendix.

The machine learning model can be converted to a deployable module using Tensor Flow Lite. The data is arranged to match the model input, which contains 512 time-steps, and continuously gets fed into the machine learning model deployment for predictions.

The smartphone application user interface is designed to show the graph of data, an icon illustrating the predicted emotion, and the probabilities of predictions for all three emotions. An illustration of the user interface is included in the Appendix.

****

*Figure 5: Sub-task Schedule for Mobile Application*

1. **Test Plan**

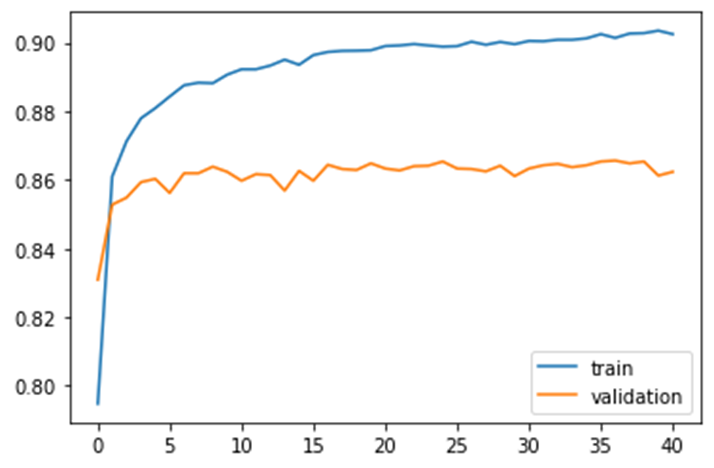
There are three main tasks that we are expecting to deliver for the product prototype: programming the microcontroller, building a Machine Learning model, and creating a smartphone application that runs on the iOS platform. Each task will be tested thoroughly to confirm its functionality.

* Microcontroller Programming:
  + Test to see if data is uniformly sampled at the serial monitor and transferred to and received at the smartphone application.
  + Test to see if the bluetooth connection is stable
* Machine Learning (ML) Model:
  + Test the model with different datasets
  + Expected accuracy of 90% during the training/validation stage and the testing stage
* Mobile Application:
  + Test for bluetooth connectivity’s range and stability, the expected range is 2-3 meters.
  + Test on a real subject, compare the subject’s emotional expressions and the predicted emotions to approximate the accuracy.
  + Check to see if data is properly and efficiently stored in local memory.

1. **Testing Result**

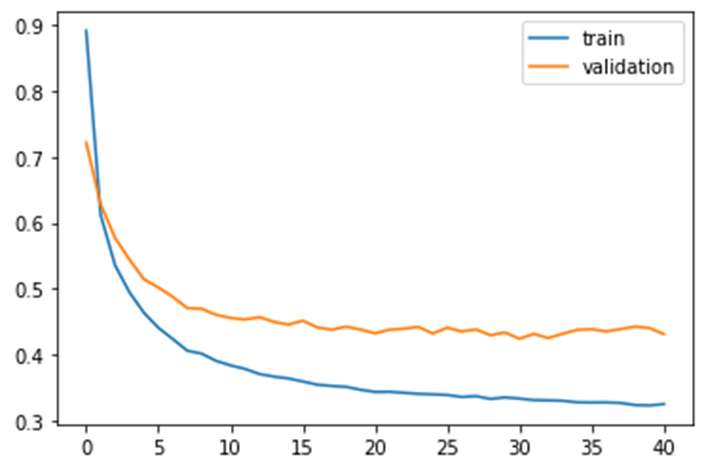
While the bluetooth connection has proved to be stable within the working range of 3 meters, the data collected at the microcontroller and transferred to the smartphone has been uniform. The data is physically counted to be 1,000 data points per second, which agrees with the frequency of 1 [KHz]. The same number of data points are collected at the receiving end.

The machine learning model has well performed as the accuracy has been increasing while the loss has been decreasing. Figure 6 shows that the accuracy of the model rises to be stable at approximately 90% for train and 86% for validation. Even though the nature of neural networks is stochastic, meaning that the model might have a different result even when training with the same dataset and the same parameters, the machine learning model that our team developed stays closed to 90% after trials of training.

****

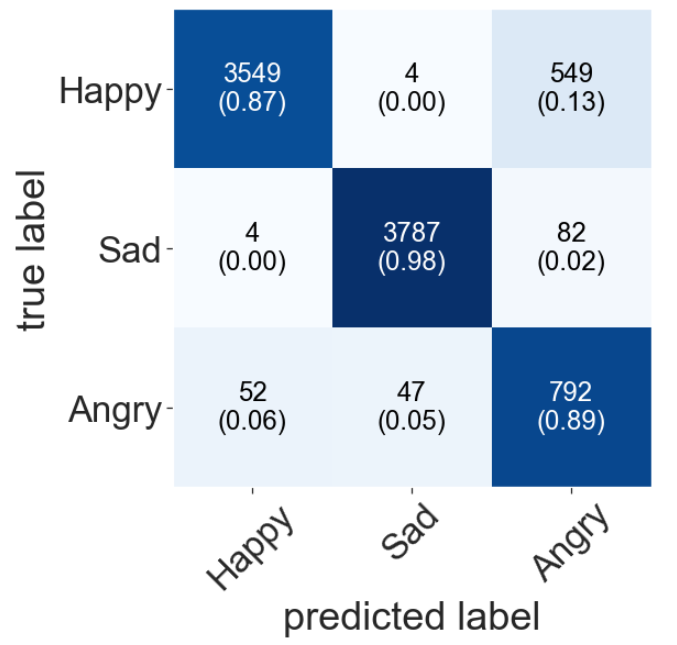
*Figure 6: Machine Learning Model’s Accuracy Approximation*

Furthermore, the loss function has resulted in a loss of about 0.35 in the train and about 0.45 in the validation as shown in figure 7. This means the model actually learns to predict rather than tries to memorize the dataset. We believe there is still room for improvement if we had a bigger dataset with more sensors.

****

*Figure 7: Machine Learning Model’s Loss Approximation*

At the testing phase, the model accuracy is 91.68%. It can be seen from figure 8 that the model seems to be confused when predicting if the emotion is angry or happy.

****

*Figure 8: Testing Phase’s Confusion Matrix*

Finally, we experienced a lower accuracy after we deployed the machine learning model on the smartphone application and tested on the subject. The model succeeds in detecting happy and sad emotions but struggles in identifying anger.

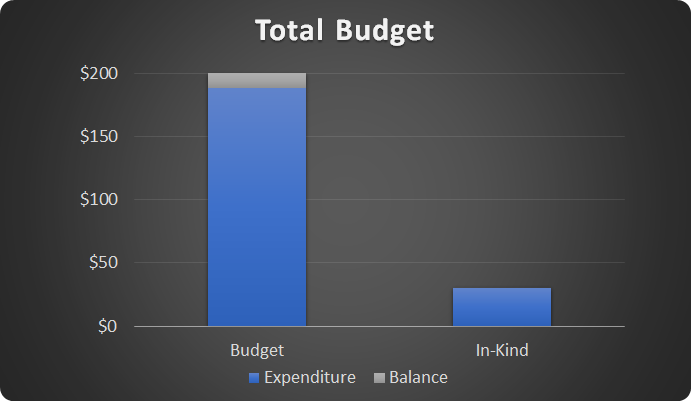
1. **Discussions and Suggestions**

It is not a coincidence that the machine learning model performs poorly in predicting if an emotion is angry or not, since angry emotion appears to be the hardest one to be detected in the three emotions. We observed, while collecting the signal for the training dataset, that the electrical signals for angry emotion are very strong at the beginning of facial expressions. However, the signals seem to get weaker quickly and eventually stay very close to -1. In contrast, the signals for happy and sad emotions are stable at their own ranges over time.

We suggest two solutions for this problem that we would have done: build a larger dataset with more testing objects and perform better data approximation at the deployment stage. The dataset used to train the machine learning model can be improved by obtaining more sensors to cover more facial muscles. The model can learn better about the differences between emotions and eventually will distinguish emotions. Furthermore, if the data is more appropriately approximated before feeding into the machine learning deployed model, the accuracy of the predictions can be higher. Due to the discussed constraints, our team did not acquire the required knowledge to upgrade the model deployment.

For future work on the project, we suggest using more compact sensors to get more EMG signals from the facial muscle and thus, get a better dataset for our model. Also, a 3D-printed prototype is needed to assemble all the hardware components into one single device.

1. **Budget and Spending**

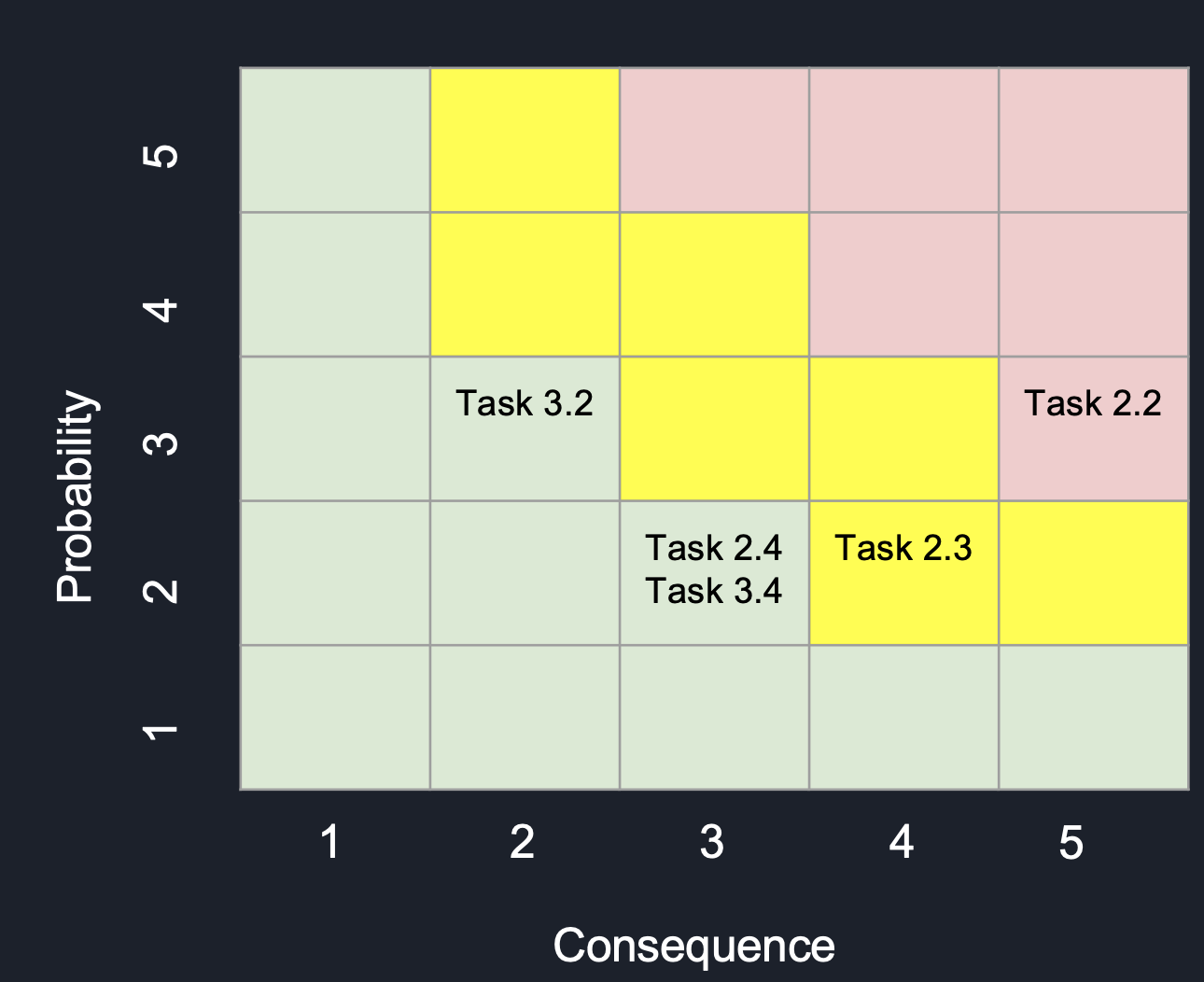
*Figure 9: Estimated Budget for the Project*

Our assigned budget was 200$ and our expenditure came close to about 190$. This includes the purchase of the ESP32 MCU, Myoware Sensors and the Electrodes. Our in-kind is around 30 dollars which is primarily the soldering station and the wires.

T*able 2: Project’s Detailed Expenditures*

|  |  |
| --- | --- |
| ***Part*** | ***Cost*** |
| Microcontroller ESP32 (x2) | *$23.80* |
| Myoware Muscle Sensor (x3) | *$123.24* |
| Non-Gelled Reusable Electrodes (x10) | *$16.24* |
| Gelled Electrodes (x100) | *$19.14* |
| Conductive Adhesive Gel (x1) | *$5.72* |
| Total | *$188.14* |

1. **Risk Management**

**

*Figure 10: Risk Matrix*

* High-risk Tasks:
  + Task 2.2 (ML Model Accuracy): the machine learning model is highly sensitive to changes. When we express an emotion, the intensity of the muscles won’t always stay the same. This will greatly affect the accuracy of the model while training and testing it.
* Medium-risk Tasks:
  + Task 2.3 (Training dataset): In the process of collecting the sample dataset, we couldn’t just collect the data all in one session since it is a very large dataset. Everytime we remove the sensors and reattach them, the locations of the sensors won’t always be exactly where they were last time. This will affect heavily on the credibility of the dataset.
* Low-risk Tasks:
  + Task 2.4 (ML Model Tune-up): Even though this task introduces low risk to the project, it has caused a delay in our progress because the GPU resource of our personal computer is not adequate to tune-up the model.
  + Task 3.4 (ML Model Deployment): This has a low probability rate since one of our members already has experience with developing an iOS application before.
  + Task 3.2 (Data Saver Function): does not have significant impact on the overall project.
* Mitigation Plans:
  + Task 2.2 (ML Model Accuracy): improve the accuracy by adding more samples to the training dataset, generalize the input sequence, and tuning up the training parameters of the model
  + Task 2.3 (Training dataset): record samples under various circumstances so that the Machine Learning Model can learn about the differences between the different locations and thus make a more accurate prediction.
  + Task 2.4 (ML Model Tune-up): instead of using our personal laptop, we use a cloud computing service to tune-up the model.
  + Task 3.4 (ML Model Deployment): use the ML Model directly on the ESP32 microcontroller and output the predicted emotions on the Arduino IDE.
  + Task 3.2 (Data Saver Function): store the data on a flash drive connected to the ESP32 while receiving data from the sensors.

1. **Summary**

Throughout two semesters of working through COVID19 and the winter-storm, we were able to accomplish the prototype for predicting emotions using EMG Signals. We completed the EMG Signal Collection and Visualization on the iOS Application. The Machine Learning Model was successfully created and trained for emotion prediction with an accuracy up to 90%. We were also able to deploy the model on the iOS Application and created a friendly user-interface for displaying the EMG Signals and the predicted emotions.

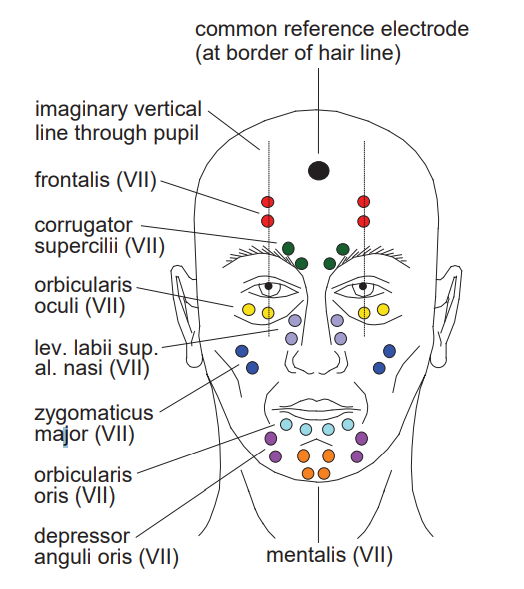
During our working process, we encountered numerous setbacks but were able to overcome them with proper project management and the hardworking spirit from the team members. Our biggest setback was still COVID-19, since we were not able to meet up face-to-face on a regular basis. Also, Chuong has been in Vietnam since last semester so the difference in time zone really makes it difficult for us to decide a time for a group meeting. Another big setback on our project was the computing limitations that our personal computers have introduced. We could not effectively train the machine learning model on a desktop as the training time was so long. We are changing our code to utilize the power of cloud computing. In the next period, we are expecting to finish the Machine Learning model tune-up as well as its deployment and the user interface on the smartphone.

In addition to the technical knowledge about EMG signals, Machine Learning Model and iOS Application Development obtained from working on this project, we also learned how to work together remotely in different time-zones. Our project management skills were also improved significantly. With the use of Gantt charts, we were able to keep track of our schedule and complete the deliverables on time. The risk matrix really helped us evaluate all the possible bad outcomes that could happen toward the end of the project so that we can come up with mitigation plans to deal with them.

**APPENDIX A**

**Sensor Placement:**

To design an unobtrusive wearable device that collects EMG signals, we must first look at the best places to put the sensors for facial EMG activity. Doing some research, we found that the best places to put the sensors are the frontalis, corrugator supercilii, and zygomaticus major [6].



*Figure 11: Electrode locations for measuring facial EMG activity [6]*

**Engineering Standards:**

1. **IEEE 11073:**

This standard comes from the IEEE (Institute of Electrical and Electronics Engineers), a professional society. The standard IEEE 11073 enables communication between health devices and external computer systems. It provides automatic and detailed electronic data capture of client-related information. Our device is a medical device for mental health, and it will send EMG signal data to a smartphone, so it should fall under this standard. The device will connect to the smartphone through BLE (Bluetooth Low Energy).

1. **IEC 60601-1:2005+A1:2012+A2:2020:**

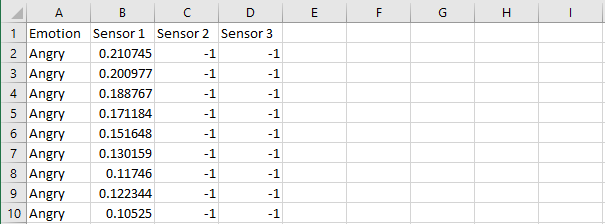
This standard comes from the IEC (International Electrotechnical Commission), an international standards organization. The standard IEC 60601 is a series of technical standards for the safety and essential performance of medical electrical equipment. In order to ensure our clients’ safety, our medical device should follow this standard. Our team will make sure that no components will be able to hurt the user. This might include picking out a safer battery or changing sensor locations.

1. **ISO 13485:**

This standard comes from the ISO (International Organization for Standardization). The ISO provides international industry standards. The standard ISO 13485 represents the requirements for a comprehensive quality management system for the design of medical devices. This standard focuses on the user's satisfaction. This standard might impact our project by having us make some changes to the headband design, so it fits better or making the app more user-friendly.

**Uniform Sampling Frequency:**

In order to create a 1 [KHz] uniform sampling frequency, we followed the tutorial from [7]. At high frequencies, the sampling rate of the loop function in the Arduino IDE is too slow and is not uniform, so at 1 [KHz], the sampling of the loop function becomes inaccurate. To solve this problem, we used timer interrupts. A timer on the ESP32 is set up to tick at 1 [MHz], and an alarm is set up to activate every 1000 ticks. This alarm triggers the ESP32 to read from the ADC. This will give us a sampling rate of 1 [KHz], but the code in the alarm cannot be too long as it only has 1000 ticks to finish its code. To solve this problem, ISR (Interrupt Service Routine) is used. The alarm will have a flag which will be checked by a non-interrupt code through ISR, and the more complex code will be done by this non-interrupt code.



*Figure 12: Dataset Format*



*Figure 13: Smartphone Application User Interface*

**References**

[1] Hull, M., 2020. PTSD Statistics. [online] The Recovery Village. Available at: <https://www.therecoveryvillage.com/mental-health/ptsd/related/ptsd-statistics/> [Accessed 16 September 2020].

[2] Depression and Bipolar Support Alliance. 2020. Depression Statistics - Depression And Bipolar Support Alliance. [online] Available at: <https://www.dbsalliance.org/education/depression/statistics/> [Accessed 16 September 2020].

[3] Shenzhen Anxinke Technology CO;LTD, “ESP-32S Datasheet”, ESP-32S Datasheet, Oct. 3rd, 2016

[4] Advancer Technologies, “3-lead Muscle / Electromyography Sensor for Microcontroller Applications”, MyoWare™ Muscle Sensor (AT-04-001), 2015

[5] Vector, “Electromyography dataset”. [online] Available at: https://www.kaggle.com/nccvector/electromyography-emg-dataset

[6] A. Van Boxtel, “Facial EMG as a Tool for Inferring Affective States,” vol. 2010, no. Figure 3, pp. 104–108, 2010.

[7] Toptal Engineering Blog. 2020. ESP32 ADC/Timing Tutorial. [online] Available at: <https://www.toptal.com/embedded/esp32-audio-sampling> [Accessed 20 November 2020].