MIE324 – Final Report

Jay Jaewon Yoo – 1002939671

David Zhang – 1003230918

Word Count: 2000

Jay Jaewon Yoo – 1002939671
David Zhang – 1003260918

1. Goal and Motivation

The purpose of this project was to predict a user's hand movement based on electromyography (EMG) signals produced from the user and visualizes this projection onto a virtual reality (VR) environment. To do so, EMG signals were recorded via the Thalmic Labs Myo armband which is lined with eight EMG sensors. These signals were then passed into a convolutional neural network, classified, and displayed to the user using an animated hand. This enables a more seamless interaction between the user and VR as it removes the need for the user to use handheld controllers to interface with the VR environment; instead, the user only needs to move their hand as they would in real life and it would transfer the same gesture to VR.

This project was motivated by the prominent uses of VR in numerous fields such as medicine, psychology, therapy, rehabilitation, robotics, and entertainment. One example of the potential of VR in these industries is the potential of to create accurate surgery simulations to train future doctors. The world is headed towards an age of intuitive natural user interfaces this is one way to create more seamless interactions with technology.

2. Description of Overall Software Structure

The project is divided into three main components. These components are the data collection and processing, the convolutional neural network, and the VR environment.

First, the EMG signals are recorded by sending the EMG signals sensed by the Myo armband to the computer using Bluetooth. The armband senses electrical activity caused by the user's hand muscles and converts it into a time-independent array of size eight, each element representing one sensor output. Following this, the array is pushed into a queue of length 10. Each of the eight signals are then averaged over the entire queue, processed into a format that the model could accept, a Tensor of size 1x8x1, and then passed into the neural network. Due to the fact that each passed array is time independent, the data cannot be normalized.

The convolutional neural network then passes in the formatted array through the model and outputs the resultant predicted gesture corresponding to the signal which is encoded as an integer. This prediction is then pushed into another queue of length 50, where the mode of the queue is then passed as

the final prediction. Both queues serve to buffer the results by reducing the noise present in the raw inputs.

As the neural network was written in Python and the Unity VR environment requires C#, the final prediction was converted into a datagram using UTF-8 encoding that was then sent from the Python server to localhost using the User Datagram Protocol (UDP). The C# code included a UDP server which received and parsed the datagrams using a buffer of size 256. As each datagram was approximately 100 bytes large, this minimized possible delays due to datagrams clogging the buffer as only two datagrams at most may occupy the buffer.

The Unity VR environment memorises all rotations the hand joints can be in as well as the current hand joint rotation. These are represented by two arrays of size 3 which store the local Euler angles of all joints in the hand. When Unity receives an encoded gesture that differs from the current one, the array containing the current rotation is updated so that the hand moves to the new gesture's rotation over several frames. This allows for smooth animations between positions as well as a fast response time; if the "new" gesture oscillates between multiple integers, the algorithm is able to interpolate the required corrections in the rotation path such that the resulting animations is always smooth and leads to the most recent "new" gesture. These animations are then displayed on screen in real time using the Unity play feature.

3. Sources of Data

a. Describe where you found your data & how you made it usable

The convolutional neural network was trained on data we acquired from ourselves. This was done by equipping the Myo armband, holding the hand at a specific gesture for a minute, and collecting the EMG signals as a two-dimensional numpy array which is then exported as a comma-separated values (CSV) file. Knowing that all input signals during that minute period is the same gesture, they can all be labelled as the same gesture, removing the issue of labelling each data sample. As each instance is time independent, each row of the two-dimensional array is an independent valid data sample. This was performed multiple times per gesture while varying arm positions, such as holding arm up high, at level height, and down low, so that the dataset had some variation within each gesture. It should be noted that the data cannot be normalized as the values are time independent.

4. Describe the machine learning model

a. Its overall structure (# layers, types)
b. Insights as to why this structure is good

A convolutional neural network was used as the machine learning model. The input is an array of eight floating point values, which are then passed into four convolutional one-dimensional layers. Each of them uses a randomized leaky rectified linear unit (RReLU) as its activation function. RReLU follows leaky ReLU for values above 0 but instead of having a linearly decreasing component for values below 0, it applies a random activation between two bounds (by default, this is a uniform distribution between $1/8$ and $1/3$). RReLU has the benefit of reducing the likelihood of overfitting due to its randomized nature; it is similar to the data augmentation technique of adding Gaussian noise. Following this, the activation map of the fourth convolutional layer is then passed into three more linear layers, with RReLU as the activation functions. The result is then passed through SoftMax to acquire the probabilities for each class and the largest probability is used as the prediction. This model is paired with the Adam optimizer and the mean squared error loss function. RReLU, Adam, and mean squared loss were used because they yielded the highest accuracies. More specifically, RReLU decreased the likelihood for overfitting, enabling the model to train for a greater number of epochs. The Adam optimizer is a standard optimizer to begin with and happened to yield accuracies above 98% and mean squared error was used as the loss function because it produced the greatest accuracy amongst the loss functions available in PyTorch.

5. Training, Validation and Test

a. A description of how your (models) were trained and how well they work

To train the model, all CSV files created during data collection were compiled into a single numpy array with 8 columns and n number of rows, where n is the number of available data samples. After expanding the dimensions of this numpy array to create the proper format, a training set and validation set was created using the train_test_split method from sklearn. Data loaders for the split datasets and an evaluate function were created to load the data and determine the accuracy of the model respectively. A batch size of 64 and learning rate of 0.001 increased the amount of computations required to train the model but also ensured the model would not overfit. Both the training and validation accuracy reached an accuracy of 96% within 10 epochs and gradually increased to 98%. The increase from 96% took approximately an extra 70 epochs. To check on the progress of the model during training, the evaluate method was called every 1024 batch iterations. After running the model for 100 total epochs, the model at the epoch with the highest validation accuracy was used as the final model. The final training

and validation accuracies were both close to 98% with a training accuracy of 98.3% and a validation accuracy of 98.0%. This suggests overfitting did not occur and training was a success.

Rather than using a test set from the existing dataset, we decided to create a demo that received EMG values from the Myo armband, generated predictions, and visualized those predictions onto a VR environment. This way of the testing of the model is as accurate as possible to how the project would be used in the real world. It also provided a limitless test set and also enabled us to easily test the performance of the entire project, not just the model itself. Using this approach to testing proved to be useful as this live testing showed that an earlier model with over 99% accuracy in fact failed to be accurate in the live demo. This implied that the model overfit to the dataset and the dataset reflected only a small subset of possible values. After altering the model and collecting new data, we reached the 98% training and validation accuracy mentioned before and found the live testing to be quite accurate. This accuracy unfortunately cannot be exactly quantified as live testing was used but it seemed to have over 90% accuracy.

6. Ethical Issues

a. Describe at least one ethical issue that arises in the project or the field that the project resides in.

The use of EMG-monitoring devices raises several ethical concerns over privacy, especially the tracking of the activities that people perform and the multitude of security problems that arise when dealing with delicate data. These security issues extend further as VR often has the user fully immersed in a virtual world, while leaving the user vulnerable. As this project is designed to remove the need to use handheld controllers, it will allow for higher degrees of immersion in the virtual world, amplifying this issue of security.

Another ethical concern is the possible psychological side effects that prolonged exposure to VR can cause. For instance, there has been a substantial rise in the prevalence of anti-socialness in East Asia which is closely correlated with the increased accessibility to online services such as the rise of online social media and entertainment[1]. By improving the responsiveness of VR, anti-socialness may increase as users would be able to experience more “realistic” virtual environments, thus reducing their motivation to

[1] "Most popular Asia-based mobile messenger apps as of 4th quarter 2017, based on number of monthly active users (in millions)." Internet: https://www.statista.com/statistics/250548/most-popular-asian-mobile-messenger-apps/, [Oct. 26, 2018].

experience the real world. This form of convenience isolates the users from the rest of society and may harm the local communities and culture overall.

7. Key Learnings

a. what would you do differently if you started all over again?

If we were to start again, we would first better scope the project. We entered the project with numerous goals that were pointed out during the first presentation to be unrealistic. The most notable goal that was unrealistic to complete was the inclusion of electroencephalography signals along with EMG signals. We spent some time attempting to incorporate both types of signals but failed, which could have been avoided with better scoping of the project.

We also strived to use frameworks that we were not familiar with such as Unity and socket programming. Learning and implementing these frameworks took nearly an entire week to accomplish and may have taken less time had we chosen to use frameworks we were much more comfortable with.

Another key learning was the importance of Git in group projects. We initially failed to use Git effectively and this hindered us due to the fact that distances forced us to work in separate location frequently. After learning to use Git more efficiently, we found the efficiency at which we completed our week drastically improved. The underestimation of Git was undoubtedly one of the greatest detriments to progress in our project.

One final note is that this project made us realize how difficult it is to convert a trained model into a practical application. This process took almost a third of the time available but we would still attempt to do so if we were to do the project again as this realization was very eye-opening.