**DMGS- 14, Harvard University Summer School 2020.**

Handwriting Recognition Project With Motion Gesture

**Josh Ostrower, Philip Eisner, Nhan Nguyen, Ander Peterson**

1. Introduction

1.1. **Motivation:**

The motivation behind our project is to assemble a wearable device that can collect gesture data, with a purpose for use with surgical patients post-surgery who are being asked to communicate their pain level on a typical 1-10 scale. This character recognition wearable device could potentially serve this post-surgical population when speaking aloud is painful or the patient is simply fatigued. One of our group members volunteers at a hospital surgery-center and has observed this problem repeatedly. Ideally the accelerometer could be incorporated into the standard fingertip Pulse-Oximeter already on board each patient for pulse and blood oxygen levels. (At that point, the accelerometer would capture gesture data from that fingertip) Because most patients have difficulty speaking during the phase of post-surgical recovery, this device has the possibility of both relieving the discomfort of speaking and aiding the patient’s communication with the staff. This device could serve to benefit the nexus of the post-surgical environment: patient, staff, family and doctors, eliminating the frustrating bottleneck of information, adding efficiency and potentially speeding-up the recovery process. Future adaptations to this design include recognition of letters and other characters to widen the communication range between the patient and caregiver by more accurately addressing their needs and pain level. This may include classifications that widen the scope of communication.

**1.2. System architecture:**

The System Architecture for our wearable device was designed to ensure data capture from the Sensortile’s accelerometer gyroscope and magnetometer to accurately record movements of the data-collector’s hand while writing a corresponding number. This accelerometer sensor system detects movement of the shoulder, elbow and wrist joints, culminating in the ultimate gesture of standard hand motion writing numbers 0 through 9 on a piece of standard 8.5 x 11” paper.

The conceptual model includes a design strategy that was devised for capturing multiple iterations of team member’s handwriting. An additional strategy to recognize gesture motion for data collection was to utilize code from the first-half of Tutorial 8. We were able to unlock the Sensortile’s Bluetooth capability which allowed for pairing between the ST BLE Sensor app and a Bluetooth enabled phone. With the ST BLE Sensor app running we could accurately record our data via CSV files. For our purposes, we used the Movement in 3-D Space recording feature within the app, capturing data from the accelerometer and gyroscope magnetometer. This sensor data was compiled into CSV files:each trial set of CSV files are then organized, separated and labeled into folders describing their trial number and corresponding digit for classification.

Once captured, this robust data collection would be processed by a separate Github repository. This particular Github repository provided the necessary code to process our data to enable the system’s machine learning capabilities.

**1.3 Characteristic of our system:**

Handwriting motion profiles for each digit were recorded with Accelerometer, Gyroscope and Magnetometer across three dimensional coordinates within the sensortile.

**1.4 Data collection system:**

A SensorTile Board powered by a computer via USB-C cable was attached to the back of the writer's hand. Data collected by the sensortile was transmitted and saved as a CSV file via STE BLE mobile app. Each trial of a digit produced one accelerometer CSV file, one magnetometer CSV file, and one gyroscope CSV file. A total of 1200 CSV files were generated by writing each digit 40 times.

**A picture containing table, holding, using, computer

Description automatically generated**

**Figure 1: SensorTile set up for data acquisition**

1. **Data Extraction and Visualization:**

In order to understand visually the differences between digits we plot their accelerometer, gyroscope, magnetometer profiles individually as well as plot all of them together. These plots help us understand the differences in the motion of the writer's hands while writing them, and let us create a better strategy to classify them. They are shown in the Figures below.

Figure 2: X-coordinate accelerometer data profiles of 0-9 digits

Figure 3: Y-coordinate accelerometer data profiles of 0-9 digits

Figure 4: Z-coordinate accelerometer data profiles of 0-9 digits

Figure 5: Magnitude of accelerometer data profiles of 0-9 digits

Figure 6: X-coordinate gyroscope data profiles of 0-9 digits

Figure 7: Y-coordinate gyroscope data profiles of 0-9 digits

Figure 8: Z-coordinate gyroscope data profiles of 0-9 digits

Figure 9: Magnitude of gyroscope data profiles of 0-9 digits

Figure 10: X-coordinate magnetometer data profiles of 0-9 digits

Figure 11: Y-coordinate magnetometer data profiles of 0-9 digits

Figure 12: Z-coordinate magnetometer data profiles of 0-9 digits

Figure 13: Magnitude of magnetometer data profiles of 0-9 digits

A close up of text on a white background

Description automatically generated

Figure 2: X-coordinate accelerometer data profiles of 0-9 digits

A close up of text on a white background

Description automatically generated

Figure 3: Y-coordinate accelerometer data profiles of 0-9 digits

A close up of text on a white background

Description automatically generated

Figure 4: Z-coordinate accelerometer data profiles of 0-9 digits

A screenshot of a cell phone

Description automatically generated

Figure 5: Magnitude of accelerometer data profiles of 0-9 digits

A screenshot of text

Description automatically generated

Figure 6: X-coordinate gyroscope data profiles of 0-9 digits

A screenshot of a cell phone

Description automatically generated

Figure 7: Y-coordinate gyroscope data profiles of 0-9 digits

A screenshot of a cell phone

Description automatically generated

Figure 8: Z-coordinate gyroscope data profiles of 0-9 digits

A screenshot of a cell phone

Description automatically generated

Figure 9: Magnitude of gyroscope data profiles of 0-9 digits

A close up of a map

Description automatically generated

Figure 10: X-coordinate magnetometer data profiles of 0-9 digits

A close up of a map

Description automatically generated

Figure 11: Y-coordinate magnetometer data profiles of 0-9 digits

A close up of a map

Description automatically generated

Figure 12: Z-coordinate magnetometer data profiles of 0-9 digits

A close up of text on a white background

Description automatically generated

Figure 13: Magnitude of magnetometer data profiles of 0-9 digits

1. **Machine learning and classifier design:**

There were many steps in getting the machine learning model working. The first step was actually getting the data into our model. This was done through a python script which looped through the data folder and checked all of the sub folders for .csv files. It then used the name of the folder the .csv files were in, as well as the name of the file itself to determine the digit recorded and the type of data(accelerometer, gyroscope or magnetometer). After that, the data was loaded into numpy arrays and parameterized into mean, min, max and standard deviation. We also normalized the data by dividing by the mean of the magnitude of each data type. It was then split into training and testing sets. We used 90% of the data for training and 10% for testing, the files used are randomly determined.

After this, the data is loaded into the machine learning model. For our classifier we chose a perceptron neural network, although we experimented with SVM as well. We used sklearn for this. The program then outputs the accuracy and the actual predictions on the test set compared to the real answers. After that it allows for input of a new data set for testing the model, not part of the original data collection, allowing for future real time implementations.

The classifier is around 90% accurate for any subset of the dataset we provide. When writing new numbers and adding the data it also makes predictions on that as well. See Figure 14 for results.

Source code of the machine learning model can be found here:

<https://github.com/Phileisner/Handwriting-Recognition/blob/master/machinelearning/final_workingML.py>

A close up of a logo

Description automatically generated

Figure 14: Prediction accuracy of machine learning model

**Issues and Improvements:**

Due to time constraints and hardware difficulties, we did not collect a lot of data. Only two people in our group collected data, and we only collected 40 trials of each digit. Each digit was only written by one person as well. Because of differing writing styles, the machine learning classifier cannot determine the correct digit when written by someone that it is not trained on, i.e. if Philip writes a digit Josh trained, the prediction is not correct. However, when written by the same person it is trained on, it is 90-100% accurate, which indicates the issue is lack of training rather than poor coding.

In the future for an expansion project, we would collect much more data with many more people in varying writing styles, hopefully to the point that we encompass the majority of writing styles and anyone can be recognized and predicted correctly. As it stands if the test data is within the same environment as the training data the model is very accurate, which is very promising for future expansion.

1. **Future work and scalability:**

Future work includes the following adaptations and enhancements of our device in order to expand its capacity, function, and end-use.

1. Expand character recognition to identify all 52 characters of the English alphabet via training recognition programs. (both upper and lowercase forms). To apply this feature, we would utilize the processes and methods described above in sections 1.2-1.4.
2. Program and implement gesture detection for intra-word spacing between characters. This would greatly enhance and expand the recognition of words and phrases with our device. This may be best accomplished by researching currently successful algorithms compatible with our device and processing system.
3. Spellchecker: develop working character models for word recognition that could potentially be used in tandem with a spellchecker to obtain a higher accuracy rating.
4. Continual Improvement: create a pipeline between the Sensortile data collection system and machine learning model in order to predict letters and digits in real-time. This would allow new characters to be implemented and learned faster, allowing the program to be more accurate as it updates to acclimate to a specific wearer’s movements.
5. Monitoring for performance success: design and implement end-to-end monitoring of character recognition success rates. This percentage would guide the machine learning processes to maximize character recognition success rates within the system.
6. Design and implement a hybrid character recognition model: fuse the character recognition application with a smart text app to create real-time texting with an embedded motion gesture app. This would expand the use of the device for multiple applications.
7. **Source code and GitHub:**

[**https://github.com/Phileisner/Handwriting-Recognition**](https://github.com/Phileisner/Handwriting-Recognition)

Team contribution

Nhan: I worked on data visualization as well as assisting in the creation of the final report and presentation.

Anders:  I contributed to the group by doing section 1.1, 1.2, and future work and scalability sections of the report.

Philip: I was the primary developer for the project. I wrote the code to import the .csv files into the program using the file data structure we had agreed upon before, which majorly streamlined the process of getting the data into the program. I additionally wrote the machine learning code that trained and tested the model, as well as allowing for the additional test data sets to be used after the model is trained. I also assisted in creating the slideshow and the final report.

Joshua: I took on the majority of the data collection for this project. About 70 to 80 percent of the data was recorded and subsequently organized to fit our data structure by me. I was also the primary motivator for the original idea and assisted in group organization and discussion. I also helped create the final report.