TransLingual: Breaking Language Boundaries

Charles Mitchiner, Stan Wijnen, Daan Sieben, and Moos Nijssen
Group 5

ABSTRACT

This article presents the design of a communication tool that employs machine learning to facilitate seamless interaction between deaf individuals and those who do not understand sign language. The tool translates sign language gestures into written text and converts spoken language into text, effectively bridging the communication gap. By utilizing machine learning the system accurately interprets hand movements to translate sign language gestures. Automatic speech recognition (ASR) methods based on recurrent neural networks and long short-term memory models convert spoken language into text. The tool features a very simple user interface, visual feedback mechanisms, and adaptive learning algorithms, taking into account the specific needs of deaf users. Experimental evaluations demonstrate its effectiveness, making it a valuable aid for enhancing accessibility and inclusivity for the deaf community across various domains.

INTRODUCTION

A staggering 430 million individuals, accounting for more than 5% of the entire global population, find themselves in need of rehabilitation to effectively address their hearing impairment. This includes 432 million adults and 34 million children who face daily challenges in communication and engagement. Alarming projections indicate that the prevalence of disabling hearing loss will escalate in the coming decades. By the year 2050, an estimated 700 million individuals, equivalent to 1 in every 10 people worldwide, will experience disabling hearing loss [9].

TransLingual aims to address the communication obstacles faced by individuals who are deaf or hard of hearing by facilitating the translation of sign language into written text. Concurrently, it enables individuals who are not proficient in sign language to effectively engage in reciprocal communication by employing speech-to-text software. Through this approach, TransLingual effectively bridges the communication gap and allows seamless interaction between the two languages.

Related Work

There are multiple products with similar intentions as TransLingual. One example makes use of Convoluted Neural Networks (CNNs), which are capable of recognizing patterns in volumetric data such as videos. This specific implementation by Doctor Krishan Kumar and Shikhar Sharma, involved training a CNN to classify 100 ASL words using the Boston ASL (Lexicon Video Dataset) LVD dataset, which consists of over 3,300 English words signed by six different individuals. The dataset was divided into two parts: 70% was used for training the model, while the remaining 30% was used for testing its performance. Through experimental evaluation, the proposed approach surpassed the existing state-of-the-art models in terms of precision (3.7%), recall (4.3%), and f-measure (3.9%). These metrics indicate the model's effectiveness in accurately recognizing ASL signs [8].

Another example is the Heart-Speaker; by utilizing the Heart-Speaker device, doctors can easily communicate with deaf patients. The system automatically captures the movements of sign language when the doctor points the Heart-Speaker at the patient, and then translates the sign language into meaningful semantics. When a doctor provides a diagnosis or poses a question to the patient, the system displays a corresponding sign language video and subtitles, enabling effective two-way communication between doctors and patients. This feature ensures that both parties can understand each other's messages accurately. To achieve sign language recognition, the system employs the MobileNet-YOLOv3 model. This model has been specifically trained to recognize and interpret sign language gestures, allowing for accurate and reliable communication between healthcare professionals and deaf-mute patients [10].

CONCEPT

Intelligence

The learning problem addressed by TransLingual is the accurate recognition and translation of sign language gestures in real-time. Additionally, TransLingual addresses the problem of translating speech into text. However, the specifics of this issue will not be discussed as it was solved by implementing an open-source speech recognition library made by Florian Schulz [4].

The skill that TransLingual learns from the data is the ability to map patterns of accelerometer and gyroscope readings to specific sign language gestures. The data collected from the sensors, represented as X = [m_ax, sd_ax, m_ay, sd_ay, m_az, sd_az, m_gx, sd_gx, m_gy, sd_gy, m_gz, sd_gz], provides information about the movement characteristics of the user's hands and arms while performing sign language gestures. These features include mean and standard deviation values of the XYZ accelerometer and gyroscope readings, which capture the overall motion and variability of the gestures.

Using this data and a machine learning algorithm, TransLingual then predicts a corresponding class, denoted as Y = [Gesture 1, Gesture 2, Gesture 3, Gesture 4, Gesture 5, Gesture 6, Gesture 7, Gesture 8, Gesture 9]. Each class represents the written text translation for a recognized sign language gesture. Respectively these are: "Hello", "I am", "Deaf", "Nice", "To meet you", "Your", "Name", "What" and "How are you?".

The skill acquired by TransLingual through the learning process is not programmable but learned from the data. This means that the system gains the ability to recognize and translate sign language gestures by discovering patterns and relationships in the accelerometer and gyroscope data. This approach is preferred over traditional programming because sign language is a complex and nuanced form of communication, relying on subtle variations in movement, handshapes, and expressions. Moreover, everyone who speaks ASL gestures in their own unique way, like how those who speak have their own accent. Capturing and encoding all these intricate details and differences through programming rules or algorithms would be challenging and impractical.

Tangibility

To materialize the concept of TransLingual and provide a tangible representation of its functionality, a physical prototype was created (See Appendix X). The prototype features a sleek and elongated rectangular box constructed from foam core material, adorned with two circular buttons and an OLED screen positioned atop. These buttons serve the purpose of seamlessly switching between ASL interpretation and speech translation modes. The OLED screen displays pertinent information, including the current mode and the translated text of ASL gestures or speech. To track ASL gestures, an MPU6050 sensor is securely attached to the user's hand and wired into the device, ensuring accurate interpretation and interaction.

Interactivity

The interactive flow diagram is shown in Figure 1. For the prototype to work, it must first be connected to Processing and the MPU6050 must be attached to the hand of the deaf individual. Once these preliminary steps have been completed, the device is ready for use. When a user wants to interact with the device, they simply press the appropriate button and either sign in ASL or speak in English. The gesture or speech is then translated into written text and printed on the OLED display. Alternatively, the user can see the translation via the Processing GUI as shown in Appendix XI. This facilitates natural conversations where both users have the option to jump in to add something by pressing their button.

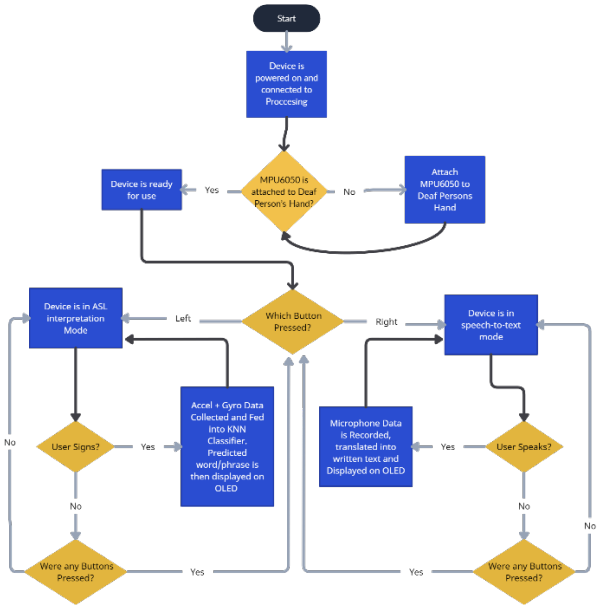


Figure 1 - Interactive Flow Diagram

REALIZATION

Hardware

TransLingual is comprised of several key hardware components that work together to facilitate gesture recognition and translation. These components include an ESP-32 microcontroller, MPU6050 accelerometer and gyroscope sensor, and a 128x32 pixel OLED display. Additionally, a microphone equipped computer capable of running the Processing IDE (Java) and Google Chrome (HTML & JavaScript) is required for the system to operate.

Data Acquisition

Gesture Recognition

For gesture recognition, the MPU6050 accelerometer and gyroscope data are collected. The ESP-32 communicates with the Processing IDE over a serial communication, transmitting the collected data. The Processing IDE utilizes a machine learning algorithm to predict the class based on the received data. This prediction is then relayed back to the Arduino over serial communication, enabling real-time updates and feedback on the OLED display.

Speech to Text

The implementation of speech-to-text functionality involves hosting an HTML webpage on Google Chrome that allows data collection from the computer's microphone. The microphone data is then passed to a JavaScript speech recognition library provided by Florian Schulz, which processes the audio and converts it into a textual representation [4].

The resulting text is printed into the website's console, where it can be accessed and relayed back to the Processing IDE using the WebSocket for Processing API [1]. Lastly, the

Processing IDE communicates the text back to the Arduino over serial communication to be displayed on the OLED screen. This allows for seamless communication and integration with the rest of TransLingual's functionality.

Feature extraction

The TransLingual project uses time-domain signal processing to obtain useful features from the raw MPU6050 data. This methodology utilizes an activation threshold, which acts as a trigger to initiate data collection when motion surpasses a predefined threshold value. Once activated, a fixed window of data is captured, enabling the system to capture a comprehensive representation of the gesture. Within this window, various statistical metrics such as the mean and standard deviation are computed for each relevant data type. This enables the collection of a substantial number of data points over a period to be represented as individual features, facilitating accurate and detailed analysis of the captured gestures.

Fine Tuning

By setting the threshold value to 20 units, undesired motions that fell below this threshold were effectively filtered out, ensuring that only motions intended for analysis were captured.

To capture all necessary information for accurate gesture recognition, a window size of 100 data points was utilized. This larger window size allowed for the detection of both short and long American Sign Language (ASL) gestures, ensuring that the system could accurately interpret a wide range of hand movements.

Learning algorithm

In the development of TransLingual, we explored various learning algorithm implementations from the Weka4P library, including Linear Support Vector Classifiers (LSVC), Polynomial SVCs, Linear Regression, RBF Kernels, and K-Nearest Neighbors (KNN) [7]. Through experimentation and evaluation, we determined that KNN was the most suitable algorithm for our application, delivering the highest accuracy and precision in gesture recognition and translation.

This conclusion is also in line with the theory surrounding these algorithms. Sign language gestures often exhibit complex and non-linear patterns, making it challenging to accurately model them using linear algorithms like LSVC or Poly SVC. Moreover, algorithms like Linear Regression or RBF Kernel are primarily designed for regression tasks or binary classifications and are not well-suited for multi-valued nominal classification.

On the contrary, KNN operates based on the principle of proximity, where the class of a new data point is determined by the classes of its nearest neighbors. In the case of gesture recognition, this approach aligns well with the concept that similar gestures should have similar translations.

Optimization

In the process of optimizing the KNN model, we focused on refining the value of the 'k' parameter, which determines the number of nearest neighbors considered for classification. Through systematic experimentation, we tested different values of 'k' ranging from 1 to 9. Our model evaluation included many metrics such as classification accuracy, mean absolute error, false positive rate, and more. Allowing us to assess the performance of each 'k' value.

Upon analysis, we found that setting 'k' to 1 yielded the highest classification accuracy of 95.182% compared to k=2 through k=9. However, to get an idea about precision it was crucial to consider metrics such as average, or relative absolute error. Considering these, we found that k=3 had a slightly lower classification accuracy of 94.8% but 1% less relative absolute error and 3% less root relative squared error (See Appendix II through Appendix IX).

When deciding between k=1 and k=3, it was important to consider the trade-off between classification accuracy and error. For example, k=1 would provide slightly higher accuracy, but k=3 showcased lower error metrics, suggesting that it could yield more precise predictions on average. Considering the complexity and variability of ASL gestures, we believe it is beneficial to prioritize lower error metrics, and thus chose to use k=3.

Evaluation

The training data collection process involved multiple iterations of data collection. In the first iteration, one person performed each gesture 10 times, resulting in a training dataset of 90 ‘X’ inputs and their corresponding labels (outputs). In the second iteration, a different person collected their own set of 90 data points and combined it with the first data set. This process was repeated for a third and fourth iteration, resulting in a combined dataset of 360 data points from four different individuals. Or in other words, the fourth data set had 40 examples of each gesture. Tables 1 through 4 display the performance of all the collected datasets.

Upon analyzing the evaluation results of the model performance for different sets of gesture training data, the third dataset (Table 3) appears to be the most favorable choice. It offers the best balance between accuracy and error rate, offering reliable and accurate predictions.

	N	%
Correctly Classified	85	94.4444 %
Incorrectly Classified	5	5.5556 %
Mean Absolute Error	-	2.2290 %

Table 1 - Model Performance for 1 set of Gestures (k=3)

	N	%
Correctly Classified	170	94.4444 %
Incorrectly Classified	10	5.5556 %
Mean Absolute Error	-	1.7900 %

Table 2 - Model Performance for 2 sets of Gestures (k=3)

To validate the performance of the trained KNN model in real-world scenarios, a practical analysis was conducted using the third dataset (Table 3) for training. During the testing phase, all three individuals who had contributed to the data collection process were asked to interact with the device. Remarkably, the KNN model accurately predicted the correct gestures on either the first or second attempt for all nine gestures, demonstrating consistent and reliable performance.

Furthermore, to assess the model's generalizability, a new individual who had not participated in the data collection or training process was invited to test the device. Although the model's accuracy slightly diminished compared to the previous tests, it still exhibited satisfactory performance. In the worst-case scenario, the device required up to four attempts to accurately predict the intended gesture.

This testing in combination with the confusion matrix output, led to the discovery that a majority of incorrectly classified data was between Gesture D (“Nice”) and Gesture F (“Your”). We assume this is because both gestures use a hand motion that starts with the palm of the hand facing the ground and both finish by moving it across the horizontal plane. This will be further discussed in the following section.

	N	%
Correctly Classified	256	94.8148 %
Incorrectly Classified	14	5.1852 %
Mean Absolute Error	-	1.6800 %

Table 3 - Model Performance for 3 sets of Gestures (k=3)

	N	%
Correctly Classified	338	93.8889 %
Incorrectly Classified	22	6.1111 %
Mean Absolute Error	-	2.0000 %

Table 4 - Model Performance for 4 sets of Gestures (k=3)

DISCUSSION

Current Limitations

Unfortunately, the close resemblance between certain hand positions and motions in some gestures poses a challenge in collecting data with sufficient variation. As a result, the KNN classifier faces difficulties in accurately distinguishing these subtle differences. This highlights a notable limitation associated with the use of the MPU6050 sensor in our project. Moreover, this sensor only captures the motions of one hand and lacks finger tracking capabilities. This could prove problematic when training the model to classify other ASL gestures that make use of both hands and or finger movements.

Additionally, the reliance on a laptop and wired connections hampers its portability and convenience. To enhance user experience, the system should be designed to operate independently or with wireless connectivity.

Lastly, the prototype's construction material, foam core, may not provide sufficient durability for long-term usage. Exploring more resilient materials can ensure the device's longevity and withstand regular wear and tear.

Future Work

To enhance portability and eliminate the need for a laptop, wireless communication can be implemented. This can be achieved by leveraging the OOCSI library for Wi-Fi-based communication between the processing unit and Arduino [5]. Additionally, the text to speech functionality would need to be integrated into the Arduino. This could be done by adding a microphone to the circuit and making use of the Arduino Speech Recognition Engine to translate speech into text. This library also offers support for 40+ languages [2].

To improve data collection and classification accuracy, an alternative to the MPU6050 sensor could be explored. Utilizing an infrared motion controller camera, such as the Leap Motion camera, can help reduce noise in data collection and enhance gesture recognition accuracy [6]. This would mitigate issues like frequent misclassification of Gesture D as Gesture F and enable more precise classification of gestures involving both hands and or finger movements. Another approach would be to add an additional MPU6050 sensor to the other hand, which can provide better hand tracking and be a more cost-effective solution compared to using the Leap Motion camera.

By implementing these future improvements, the TransLingual system can overcome its current limitations, offering wireless functionality, enhanced gesture recognition accuracy, and improved usability for multilingual communication.

VIDEO PRESENTATION: HTTPS://YOUTU.BE/AX_3XGBPJHU

NEW DEMO WITH GUI SHOWCASE: HTTPS://YOUTU.BE/QSLKYRQN3NY

REFERENCES

[1] Alexandre Alapetite (n.d.). GitHub - alexandrainst/processing_websockets: A web socket library, including both server and client, for Processing. GitHub. https://github.com/alexandrainst/processing_websockets

[2] Arduino Speech Recognition Engine. (n.d.-b). Arduino Official Store. https://store.arduino.cc/products/speech-recognition-engine

[3] ChatGPT was used to summarize and shorten text, https://chat.openai.com/

[4] Floran Schulz. 2017. In *Speech Recognition for Java/Processing*. https://florianschulz.info/stt/

[5] Funk, M. F. (n.d.). OOCSI.Retrieved June 8, 2023, https://oocsi.id.tue.nl/

[6] Leap Motion Controller 2 | Smaller, Lighter, Better | Ultraleap. (n.d.) https://www.ultraleap.com/product/leap-2/

[7] Liang, R.-H. (2021, February 24). GitHub - howieliang/Weka4P: Weka for Processing. GitHub. Retrieved June 24, 2023, https://github.com/howieliang/Weka4P

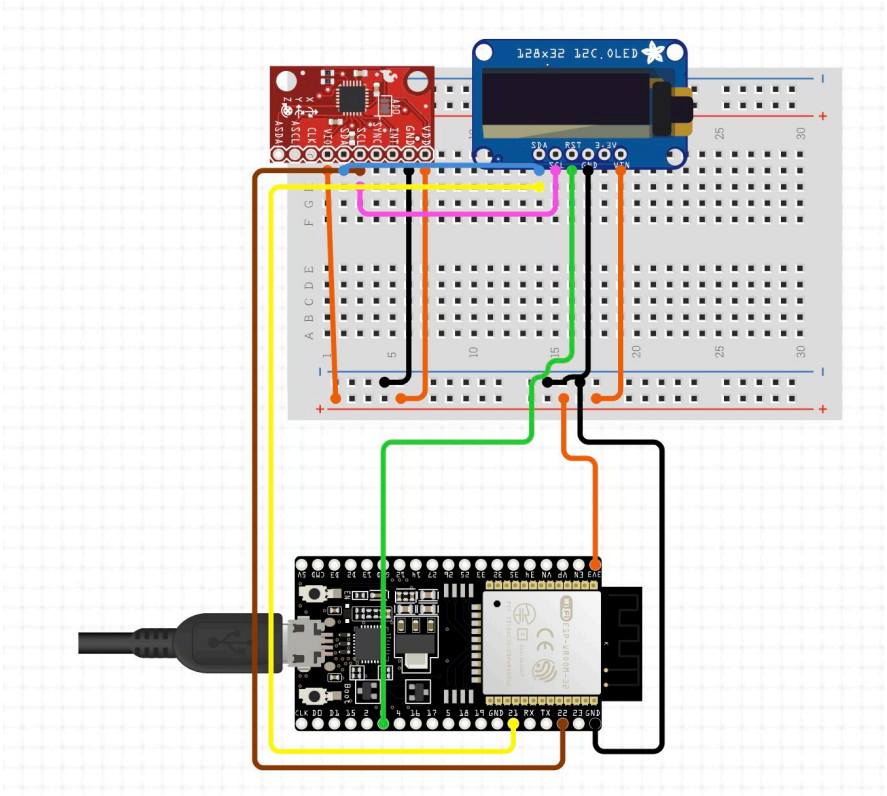
[8] Sharma, Shikhar & Kumar, Krishan. (2021). ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks. 80. 1-13. doi: 10.1007/s11042-021-10768-5.

[9] World Health Organization: WHO. (2023). Deafness and hearing loss. https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing loss#:~:text=Overview,will%20have%20disabling%20hearing%20loss.

[10] Xia K, Lu W, Fan H, Zhao Q. A Sign Language Recognition System Applied to Deaf-Mute Medical Consultation. Sensors (Basel). 2022 Nov 24;22(23):9107. doi: 10.3390/s22239107.

APPENDIX

I – TransLingual Circuit, made on https://www.circuito.io/



II – Dataset 1 Evaluation (k=1)

```
Results
======

Correctly Classified Instances          86               95.5556 %
Incorrectly Classified Instances         4                4.4444 %
Kappa statistic                          0.95
Mean absolute error                      0.0307
Root mean squared error                  0.1
Relative absolute error                 15.5556 %
Root relative squared error             31.8174 %
Total Number of Instances               90

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i   <-- classified as
 10  0  0  0  0  0  0  0  0 |  a = A
  0 10  0  0  0  0  0  0  0 |  b = B
  0  0 10  0  0  0  0  0  0 |  c = C
  0  0  0 10  0  0  0  0  0 |  d = D
  0  0  0  0  9  0  1  0  0 |  e = E
  0  0  1  0  0  9  0  0  0 |  f = F
  0  0  0  0  0  0  9  0  1 |  g = G
  0  0  0  0  0  0  0 10  0 |  h = H
  0  0  0  0  0  0  1  0  9 |  i = I

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   MCC      ROC Area  PRC Area  Class
               1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     A
               1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     B
               1.000    0.013    0.909      1.000    0.952      0.947    0.994     0.909     C
               1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     D
               0.900    0.000    1.000      0.900    0.947      0.943    0.950     0.911     E
               0.900    0.000    1.000      0.900    0.947      0.943    0.950     0.911     F
               0.900    0.025    0.818      0.900    0.857      0.840    0.938     0.747     G
               1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     H
               0.900    0.013    0.900      0.900    0.900      0.888    0.944     0.821     I
Weighted Avg.  0.956    0.006    0.959      0.956    0.956      0.951    0.975     0.922
```

III – Dataset 2 Evaluation (k=1)

```
Results
======

Correctly Classified Instances         172               95.5556 %
Incorrectly Classified Instances         8                4.4444 %
Kappa statistic                          0.95
Mean absolute error                      0.0209
Root mean squared error                  0.0982
Relative absolute error                 10.5882 %
Root relative squared error             31.2375 %
Total Number of Instances              180

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i   <-- classified as
 20  0  0  0  0  0  0  0  0 |  a = A
  0 20  0  0  0  0  0  0  0 |  b = B
  0  0 20  0  0  0  0  0  0 |  c = C
  0  1  0 19  0  0  0  0  0 |  d = D
  0  0  0  0 19  0  1  0  0 |  e = E
  0  0  1  0  1 18  0  0  0 |  f = F
  0  0  0  0  1  0 18  0  1 |  g = G
  0  0  0  0  0  0  0 19  1 |  h = H
  0  0  0  0  0  0  1  0 19 |  i = I

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     A
                 1.000    0.006    0.952      1.000    0.976      0.973    0.997     0.952     B
                 1.000    0.006    0.952      1.000    0.976      0.973    0.997     0.952     C
                 0.950    0.000    1.000      0.950    0.974      0.972    0.975     0.956     D
                 0.950    0.013    0.905      0.950    0.927      0.918    0.969     0.865     E
                 0.900    0.000    1.000      0.900    0.947      0.943    0.950     0.911     F
                 0.900    0.013    0.900      0.900    0.900      0.888    0.944     0.821     G
                 0.950    0.000    1.000      0.950    0.974      0.972    0.975     0.956     H
                 0.950    0.013    0.905      0.950    0.927      0.918    0.969     0.865     I
Weighted Avg.    0.956    0.006    0.957      0.956    0.956      0.951    0.975     0.920
```

IV – Dataset 3 Evaluation (k=1)

```
Results
======

Correctly Classified Instances         257               95.1852 %
Incorrectly Classified Instances        13                4.8148 %
Kappa statistic                          0.9458
Mean absolute error                      0.0182
Root mean squared error                  0.1021
Relative absolute error                  9.2    %
Root relative squared error             32.4962 %
Total Number of Instances              270

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i   <-- classified as
 29  0  1  0  0  0  0  0  0 |  a = A
  0 29  0  1  0  0  0  0  0 |  b = B
  0  0 30  0  0  0  0  0  0 |  c = C
  0  3  0 27  0  0  0  0  0 |  d = D
  0  0  0  0 29  0  1  0  0 |  e = E
  0  0  2  0  1 27  0  0  0 |  f = F
  0  0  0  0  1  0 28  0  1 |  g = G
  0  0  0  0  0  0  0 29  1 |  h = H
  1  0  0  0  0  0  0  0 29 |  i = I

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.967    0.004    0.967      0.967    0.967      0.963    0.981     0.938     A
                 0.967    0.013    0.906      0.967    0.935      0.928    0.977     0.880     B
                 1.000    0.013    0.909      1.000    0.952      0.947    0.994     0.909     C
                 0.900    0.004    0.964      0.900    0.931      0.923    0.948     0.879     D
                 0.967    0.008    0.935      0.967    0.951      0.945    0.979     0.908     E
                 0.900    0.000    1.000      0.900    0.947      0.943    0.950     0.911     F
                 0.933    0.004    0.966      0.933    0.949      0.943    0.965     0.909     G
                 0.967    0.000    1.000      0.967    0.983      0.981    0.983     0.970     H
                 0.967    0.008    0.935      0.967    0.951      0.945    0.979     0.908     I
Weighted Avg.    0.952    0.006    0.954      0.952    0.952      0.946    0.973     0.912
```

V – Dataset 4 Evaluation (k=1)

```
Results
======

Correctly Classified Instances         332               92.2222 %
Incorrectly Classified Instances        28                7.7778 %
Kappa statistic                          0.9125
Mean absolute error                      0.0211
Root mean squared error                  0.123
Relative absolute error                 11.8491 %
Root relative squared error             41.2632 %
Total Number of Instances              360

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j   <-- classified as
 40  0  0  0  0  0  0  0  0  0 |  a = A
  0 33  0  7  0  0  0  0  0  0 |  b = B
  0  0 40  0  0  0  0  0  0  0 |  c = C
  0  9  0 31  0  0  0  0  0  0 |  d = D
  0  0  0  0 39  0  1  0  0  0 |  e = E
  1  0  3  0  0 36  0  0  0  0 |  f = F
  0  0  0  0  1  0 38  0  1  0 |  g = G
  0  0  0  0  0  0  0 39  1  0 |  h = H
  0  0  0  0  1  0  3  0 36  0 |  i = I

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.003    0.976      1.000    0.988      0.986    0.998     0.976     A
                 0.825    0.028    0.786      0.825    0.805      0.780    0.898     0.668     B
                 1.000    0.009    0.930      1.000    0.964      0.960    0.995     0.930     C
                 0.775    0.022    0.816      0.775    0.795      0.770    0.877     0.657     D
                 0.975    0.006    0.951      0.975    0.963      0.958    0.984     0.930     E
                 0.900    0.000    1.000      0.900    0.947      0.943    0.950     0.911     F
                 0.950    0.013    0.905      0.950    0.927      0.918    0.969     0.865     G
                 0.975    0.000    1.000      0.975    0.987      0.986    0.988     0.978     H
                 0.900    0.006    0.947      0.900    0.923      0.914    0.947     0.864     I
Weighted Avg.    0.922    0.010    0.923      0.922    0.922      0.913    0.956     0.864
```

VI – Dataset 1 Evaluation (k=3)

```
Results
======

Correctly Classified Instances          85               94.4444 %
Incorrectly Classified Instances         5                5.5556 %
Kappa statistic                          0.9375
Mean absolute error                      0.0229
Root mean squared error                  0.1018
Relative absolute error                 11.6    %
Root relative squared error             32.3975 %
Total Number of Instances               90

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i   <-- classified as
 10  0  0  0  0  0  0  0  0 |  a = A
  0 10  0  0  0  0  0  0  0 |  b = B
  0  0 10  0  0  0  0  0  0 |  c = C
  0  0  0 10  0  0  0  0  0 |  d = D
  0  0  0  0  9  1  0  0  0 |  e = E
  0  0  2  0  0  8  0  0  0 |  f = F
  1  0  0  0  0  0  9  0  0 |  g = G
  0  0  0  0  0  0  0 10  0 |  h = H
  1  0  0  0  0  0  0  0  9 |  i = I

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.025    0.833      1.000    0.909      0.901    0.999     0.991     A
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     B
                 1.000    0.025    0.833      1.000    0.909      0.901    0.994     0.909     C
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     D
                 0.900    0.000    1.000      0.900    0.947      0.943    0.949     0.911     E
                 0.800    0.013    0.889      0.800    0.842      0.825    0.946     0.871     F
                 0.900    0.000    1.000      0.900    0.947      0.943    0.949     0.911     G
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     H
                 0.900    0.000    1.000      0.900    0.947      0.943    0.949     0.911     I
Weighted Avg.    0.944    0.007    0.951      0.944    0.945      0.940    0.976     0.945
```

VII – Dataset 2 Evaluation (k=3)

```
Results
======

Correctly Classified Instances         170               94.4444 %
Incorrectly Classified Instances        10                5.5556 %
Kappa statistic                          0.9375
Mean absolute error                      0.0179
Root mean squared error                  0.0965
Relative absolute error                  9.0769 %
Root relative squared error             30.6922 %
Total Number of Instances              180

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i   <-- classified as
 18  0  0  0  0  0  2  0  0 |  a = A
  0 20  0  0  0  0  0  0  0 |  b = B
  0  0 20  0  0  0  0  0  0 |  c = C
  0  1  0 19  0  0  0  0  0 |  d = D
  0  0  0  0 19  1  0  0  0 |  e = E
  0  0  3  0  1 16  0  0  0 |  f = F
  1  0  0  0  0  0 19  0  0 |  g = G
  0  0  0  0  0  0  0 20  0 |  h = H
  1  0  0  0  0  0  0  0 19 |  i = I

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0.900    0.013    0.900      0.900    0.900      0.888    0.999     0.991     A
                1.000    0.006    0.952      1.000    0.976      0.973    1.000     0.993     B
                1.000    0.019    0.870      1.000    0.930      0.924    0.996     0.944     C
                0.950    0.000    1.000      0.950    0.974      0.972    1.000     0.993     D
                0.950    0.006    0.950      0.950    0.950      0.944    0.971     0.908     E
                0.800    0.006    0.941      0.800    0.865      0.853    0.947     0.892     F
                0.950    0.013    0.905      0.950    0.927      0.918    0.974     0.951     G
                1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     H
                0.950    0.000    1.000      0.950    0.974      0.972    0.975     0.956     I
Weighted Avg.   0.944    0.007    0.946      0.944    0.944      0.938    0.985     0.959
```

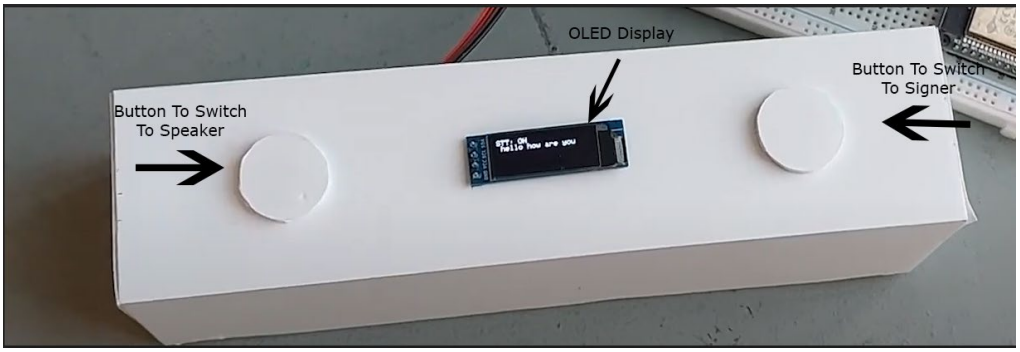
VIII – Dataset 3 Evaluation (k=3)

```
Results
======

Correctly Classified Instances         256               94.8148 %
Incorrectly Classified Instances        14                5.1852 %
Kappa statistic                          0.9417
Mean absolute error                      0.0168
Root mean squared error                  0.0911
Relative absolute error                  8.4932 %
Root relative squared error             29.001  %
Total Number of Instances              270

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i   <-- classified as
 28  0  2  0  0  0  0  0  0 |  a = A
  0 29  0  1  0  0  0  0  0 |  b = B
  0  0 30  0  0  0  0  0  0 |  c = C
  0  3  0 27  0  0  0  0  0 |  d = D
  0  0  0  0 29  1  0  0  0 |  e = E
  1  0  3  0  1 25  0  0  0 |  f = F
  0  0  0  0  1  0 29  0  0 |  g = G
  0  0  0  0  0  0  0 30  0 |  h = H
  1  0  0  0  0  0  0  0 29 |  i = I

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0.933    0.008    0.933      0.933    0.933      0.925    1.000     0.996     A
                0.967    0.013    0.906      0.967    0.935      0.928    0.999     0.982     B
                1.000    0.021    0.857      1.000    0.923      0.916    0.998     0.968     C
                0.900    0.004    0.964      0.900    0.931      0.923    0.999     0.982     D
                0.967    0.008    0.935      0.967    0.951      0.945    0.981     0.938     E
                0.833    0.004    0.962      0.833    0.893      0.883    0.966     0.930     F
                0.967    0.000    1.000      0.967    0.983      0.981    0.983     0.970     G
                1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     H
                0.967    0.000    1.000      0.967    0.983      0.981    0.983     0.970     I
Weighted Avg.   0.948    0.006    0.951      0.948    0.948      0.943    0.990     0.971
```

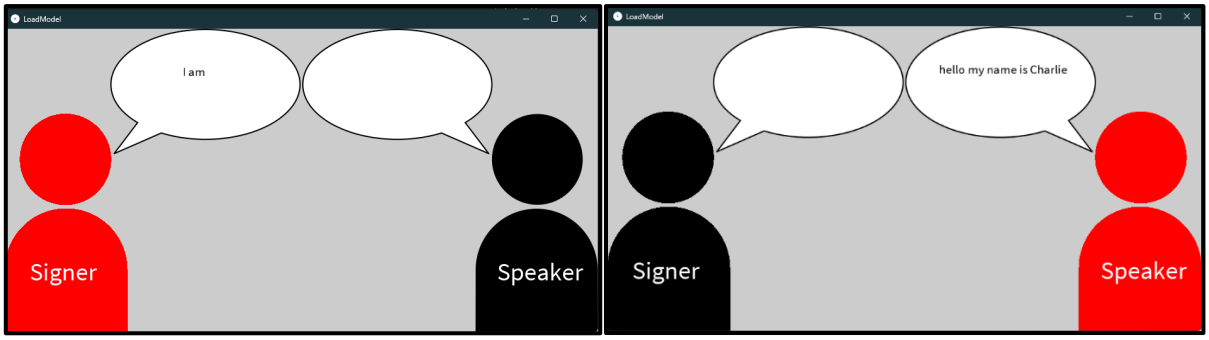
IX – Dataset 4 Evaluation (k=3)

```
Results
======

Correctly Classified Instances         338               93.8889 %
Incorrectly Classified Instances        22                6.1111 %
Kappa statistic                          0.9313
Mean absolute error                      0.02
Root mean squared error                  0.1053
Relative absolute error                 10.1031 %
Root relative squared error             33.4917 %
Total Number of Instances              360

=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i   <-- classified as
 39  0  1  0  0  0  0  0  0 |  a = A
  0 34  0  6  0  0  0  0  0 |  b = B
  0  0 40  0  0  0  0  0  0 |  c = C
  0  4  0 36  0  0  0  0  0 |  d = D
  0  1  0  0 39  0  0  0  0 |  e = E
  1  0  2  0  0 37  0  0  0 |  f = F
  1  0  0  0  1  0 37  0  1 |  g = G
  0  0  0  0  0  0  0 40  0 |  h = H
  2  0  0  0  1  0  1  0 36 |  i = I

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.975    0.013    0.907      0.975    0.940      0.933    1.000     0.993     A
               0.850    0.016    0.872      0.850    0.861      0.844    0.967     0.862     B
               1.000    0.009    0.930      1.000    0.964      0.960    0.998     0.972     C
               0.900    0.019    0.857      0.900    0.878      0.863    0.967     0.861     D
               0.975    0.006    0.951      0.975    0.963      0.958    0.986     0.953     E
               0.925    0.000    1.000      0.925    0.961      0.957    0.962     0.933     F
               0.925    0.003    0.974      0.925    0.949      0.943    0.974     0.951     G
               1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     H
               0.900    0.003    0.973      0.900    0.935      0.928    0.987     0.969     I
Weighted Avg.  0.939    0.008    0.940      0.939    0.939      0.932    0.982     0.944
```

X – Prototype Design



XI – GUI Design (Left: ASL Translation Mode | Right: Speech Translation Mode)



XII – Arduino & Processing Code & Training/Testing Data

https://drive.google.com/file/d/18s-63kZ7XBH2WMRCid5YEonU-xTnIo7D/view?usp=sharing