Filipino Sign Language to Text Converter using K-Nearest Neighbor Algorithm

John Vincent G. Calites[1], Jean Tristan L. Reyes[2], Glenn Christian D. Sioson[3]

School of Electrical, Electronics, and Computer Engineering, Mapúa University

Muralla St., Intramuros, Manila 1002, Philippines

[1]jvgcalites@mymail.mapua.edu.ph

[2]jtlreyes@mymail.mapua.edu.ph

[3]gcdsioson@mymail.mapua.edu.ph

***Abstract*— The Philippines has recognized Filipino Sign Language as the national sign language of the Filipino deaf and the official sign language of the government (RA 11106), this involves all the transactions involving the deaf, its usage in schools, broadcast media, and workplaces. This project develops a glove that will translate Filipino Sign Language (FSL) to text using the K-Nearest Neighbor (KNN) algorithm. The device will only translate the alphabet instead of words. The device, while worn by the user, will do a real-time language translation through the means of capturing hand movements and gestures. The device is portable in the sense that it will only need an android phone in viewing the translated gesture while connected to the gloves through Bluetooth. After conducting several tests, it shows an accuracy of 89.63% when translating FSL to text.**

Keywords: Android, Bluetooth, FSL, Gestures, Hand movements

I. Introduction

Hearing impairment is the decreased ability to hear a sound which can occur at any age, this may be due to some genetic factors which can make a newly born child deaf, or people at any age become deaf due to complications, diseases, nerve damages, or injuries [1]. Communication plays a vital role in society; it employs the transfer of information that is essential in developing relations, although several forms of communication exist for those with hearing impairment such as sign language, it is not understood by many [2]. Thus, the communication for those who have little to no knowledge of sign language and people with hearing disabilities must be addressed. On October 30, 2018, the Philippines has recognized Filipino Sign Language as the national sign language of the Filipino deaf and the official sign language of the government (RA 11106), this involves all the transactions involving the deaf, its usage in schools, broadcast media, and workplaces. Since most of the community uses American Sign Language (ASL), all government offices must strictly use Filipino Sign Language (FSL) as the medium of instruction for education [3].

Several kinds of research have been made to bridge the problem in communicating with the hearing impaired, a research done by Piscasio implemented an FSL translator using accelerometers and Bluetooth to connect to the PC as the processing and output device [4]. Another research from Mapua University developed a system that converts hand gestures into Filipino words using a webcam to capture hand gestures which will then be processed into Filipino words [5]. Another research uses a web camera that captures hand gestures, this will then serve as an input to their python code in OpenCV to determine the gesture done using various techniques [6].

All these researches pointed out a problem to solve, which is to create a device capable of bridging the problem between signers and non-signers, but not all of the time there is a capable machine of using real-time image processing and machine learning simultaneously to classify a gesture. That is why the researchers intend to create a portable device that will help by providing a means of training people in Filipino Sign Language. The proposed solution is a glove that will translate Filipino Sign Language (FSL) to text, which can be viewed on an android phone. The glove, while worn by the user, will do a real-time language translation through the means of capturing hand movements and gestures [7].

In this design project, the group intends to develop a real-time language translation of Filipino Sign Language to text. Specifically, the group wants to achieve the following: (a) to use a flex sensor, gyroscope, and accelerometer to capture hand movements and gestures, (b) to create a portable device utilizing an Android application capable of translating FSL to text, (c) to use the K-Nearest Neighbor Algorithm for classifying gestures.

The study is made specifically to benefit those who have little knowledge about FSL and have a hard time understanding it. This device will also benefit the deaf since they can now train by themselves and their knowledge will be reinforced by showing them if their gesture is correct. The device can help those who are beginners in terms of FSL for them to learn the basics of sign language which is fingerspelling.

The system is limited to the following; Only the right-hand glove will be designed and built for gesture recognition. The microcontroller will only send the data coming from the sensor to the android phone once the force sensitive resistor is pressed, this is done so that it will only send data to the android phone when the user wants to fingerspell a letter and to avoid transmitting data when it is idle. Only one android phone will be able to connect to the device, this will be used to process the data and convert the signals to text because KNN uses large amounts of resources.

II. METHODOLOGY

A. Conceptual Framework

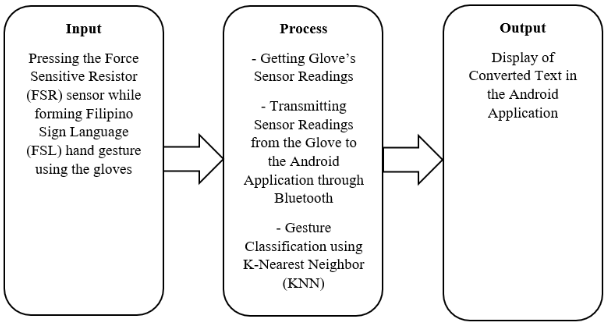

Figure 1 Conceptual Framework

Figure 1 shows the conceptual framework of the system. The input will come from the hand gestures of the user that corresponds to the Filipino Sign Language (FSL) once the Force Sensitive Resistor (FSR) is pressed. There will be sensors providing the data from the hand gestures. The flex sensor will provide the resistance following the flexion of the fingers while the MPU6050 will provide the gyroscope and accelerometer values of the device [8][9]. The data will be read by the microcontroller and will be sent to the android phone via Bluetooth connection. In the android phone, the data will be processed using the K-Nearest Neighbor (KNN) Algorithm which finds the most frequent label for a given value of K on the trained data [10]. Lastly, the label's equivalent text will be displayed on the android phone.

B. Hardware Development

1) Block Diagram

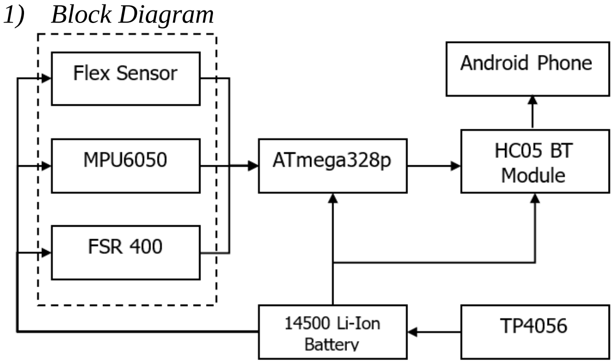

Figure 2 Block Diagram

Figure 2 illustrates the interaction among the components of the device. The microcontroller ATmega328p will serve as the controller for the system in which it controls and reads data from the different components. The ATmega328p reads the resistance coming from the flex sensor, the acceleration, and position from the MPU6050, and if the glove is ready to transit sensor values by pressing the FSR. The microcontroller then sends it to the android phone through the HC05 Bluetooth Module. Then, the android phone uses the data gathered by the sensors to classify the gesture using the KNN algorithm and translates it accordingly. The translated text will be displayed on the android phone. The power will come from the Li-Ion battery that is rechargeable through means of the TP4056.

2) Isometric Model

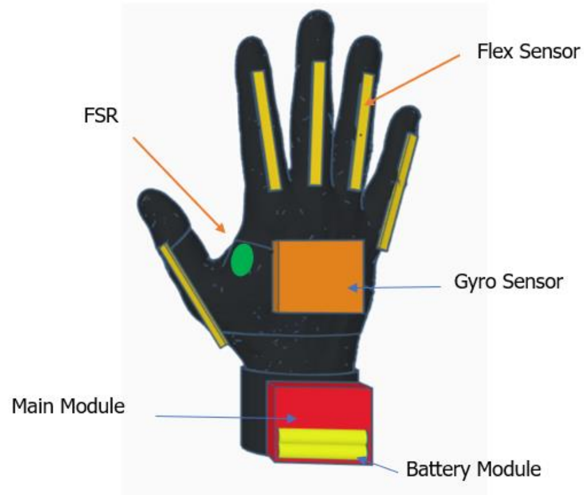

Figure 3 Isometric Model of the Prototype

Figure 3 shows the casing for the components. The flex sensors will be placed on top of the fingers. The force-sensitive resistor is placed on the thenar space while the MPU6050 Gyro and Accelerometer sensor is placed on the back of the hand. The circuits of the main module and the battery module will be placed in a case along the top of the wrist. All the sensors in the glove will be connected to the main module, along with the ATmega328p microcontroller and the Bluetooth Module. The two 3.7V Li-Ion battery will be connected to the battery module, along with the voltage regulator and TP4056 charging module. The battery module will be responsible for powering up the main module. The actual prototype created is shown in Figure 4.

Figure 4 Actual Prototype

C. Software Development

1) ATmega328p Microcontroller

The program must be able to let the ATmega328p microcontroller to receive data from the flex sensors, accelerometer, gyroscope, and force resistive resistor and send it to the Android phone via Bluetooth.

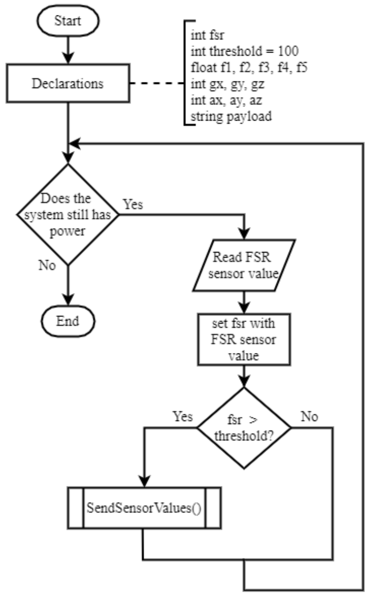

Figure 5 ATmega328p Program Flowchart

Figure 5 shows the flow of the program that is being implemented on the microcontroller. All the variables to be used are declared at the start of the program. From here, the program will loop until the system gets powered down. While the system has power, it will read the Force Sensitive Resistor (FSR) Sensor value. If the FSR value is greater than the given threshold, the module in Figure 6 will be used.

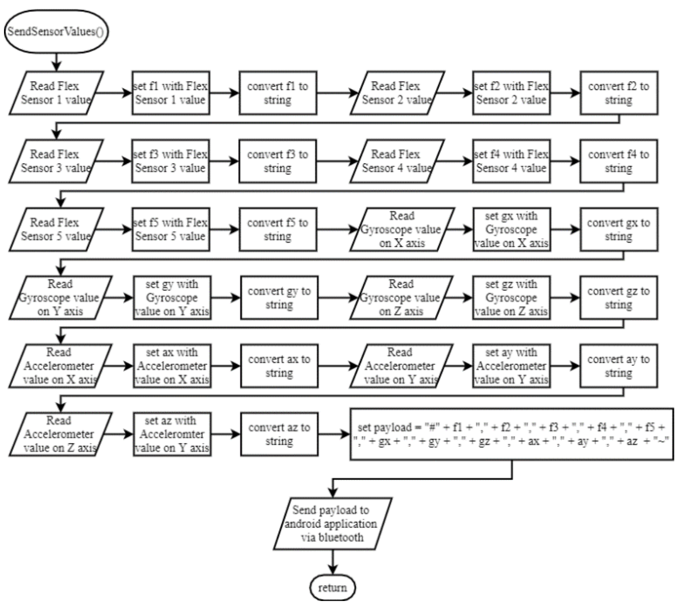

Figure 6 Send Sensor Value module flowchart

Figure 6 shows the module to be used when the FSR reached the threshold value. This module reads all the value of Flex Sensors, Gyroscope, and Accelerometer, assign each sensor readings to variables, and converts all the variables to a string. The variables are then concatenated with delimiters to produce a specific format that will be able to parse by the android application.

2) Training Application Development

A training application is needed to create a dataset that will be used in the android application. The training application should be able to receive sensor data from the microcontroller of the gloves and classify it to a specific label using the K-Nearest Neighbor (KNN).

Since KNN is a supervised learning algorithm, the training application should be able to ask the user if the classification is correct or not [11]. In this way, every time the classification is wrong, it could be corrected. Lastly, when the user responds if the classification is correct or not, the application should be able to collect the received sensor data and add the proper label with it. Figure 7 shows a screenshot of the training application that was created for the creation of the dataset that will be used to classify hand gestures using KNN.

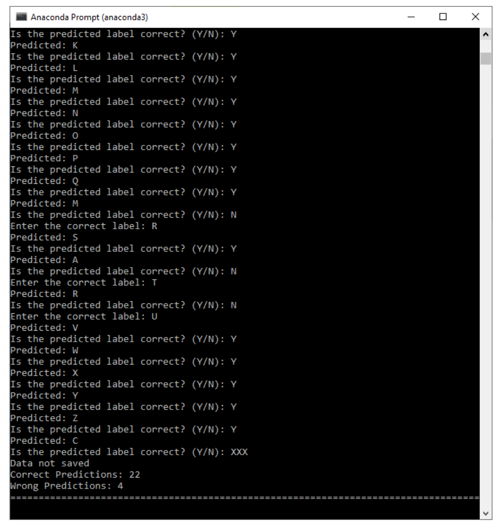

Figure 7 Training Application

With the use of the training application, the creation of the dataset was realized. Figure 8 shows a sample of a dataset created with the use of the training application. Each line is a sample, and in each sample, there are 11 features: 5 flex sensor values, 3 gyroscope sensor values, and 3 accelerometer values, and a label at the end of each line.

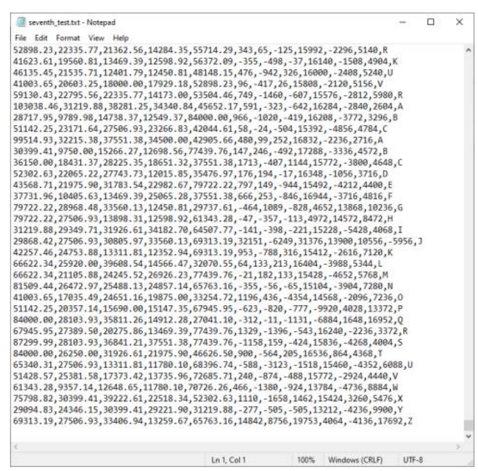

Figure 8 Sample Dataset

3) *Android Application*

The main purpose of developing an Android application is to display the equivalent text of a corresponding gesture. To implement this, the application should be able to let the Android phone to connect and receive sensor data from the microcontroller, classify the input data using K-Nearest Neighbor Algorithm, and display the output label as text on the screen of the android phone.

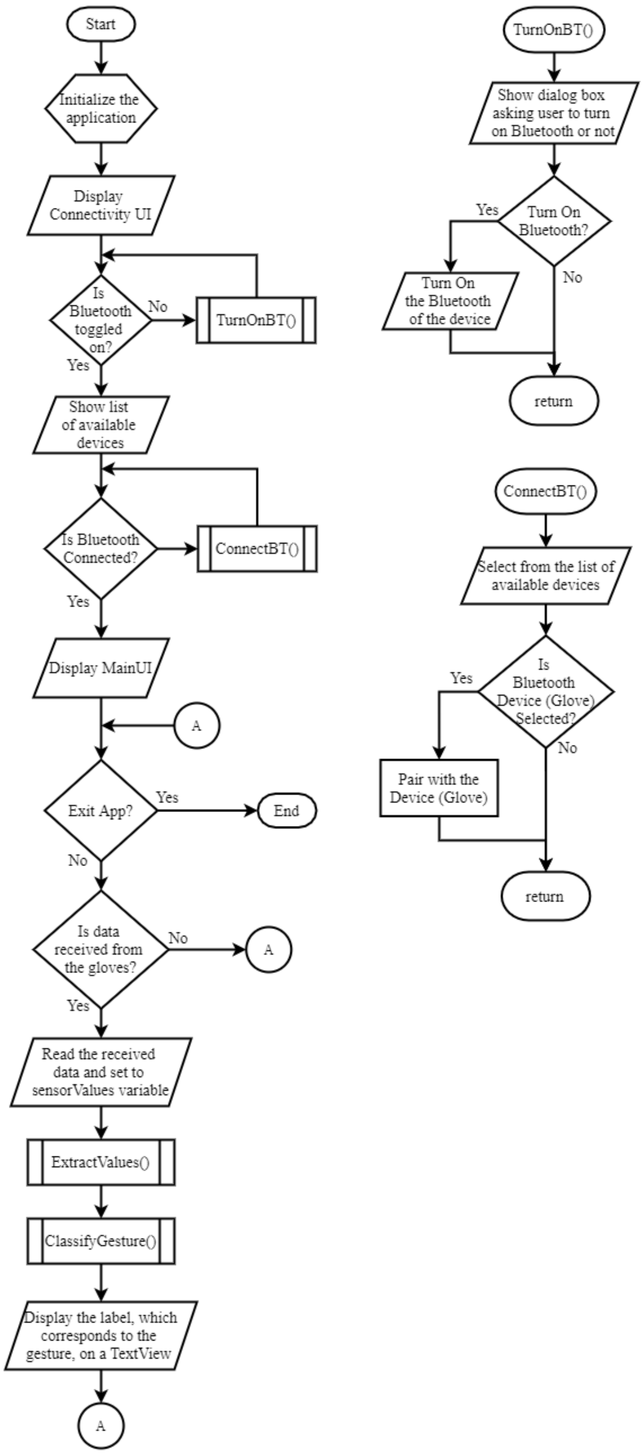

Figure 9 Android Application Flowchart

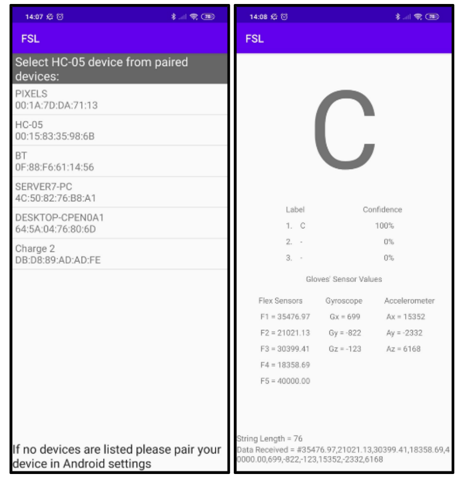

Figure 10 Connectivity UI and Main UI of the Android Application

Figure 9 illustrates the flow of the program of the android application. As the application starts, the application will initialize and ask the user to turn on the Bluetooth if it is turned off. Once the Bluetooth is turned on, the application will show a list of available devices to connect. The user must choose the glove as the device to connect to. Once connected, the main user interface will be shown and is now ready to receive data from gloves. Every time data is received from the gloves, the data will be extracted and will be used in classifying the hand gesture equivalent label using the K Nearest Neighbor algorithm. The resulting label will be displayed on the layout of the android application. Figure 10 shows the developed android application.

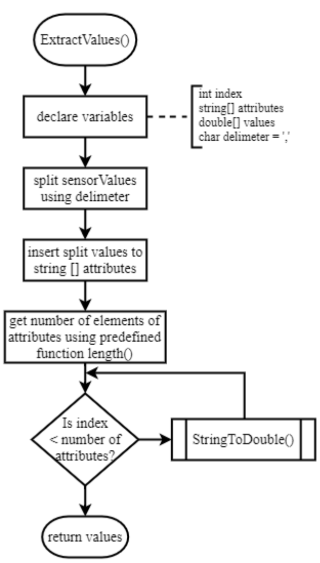

Figure 11 Extract Values Module

To extract the string received from the gloves, Figure 11 will be implemented. The string will be split using the assigned delimiter. The split parts will then be placed to an array named attributes, which will contain all the sensor values but with a string data type. To be able to process the elements in the array, each element needs to be converted to a double data type. Once converted, the array containing the elements can now be used as an input to the classifier.

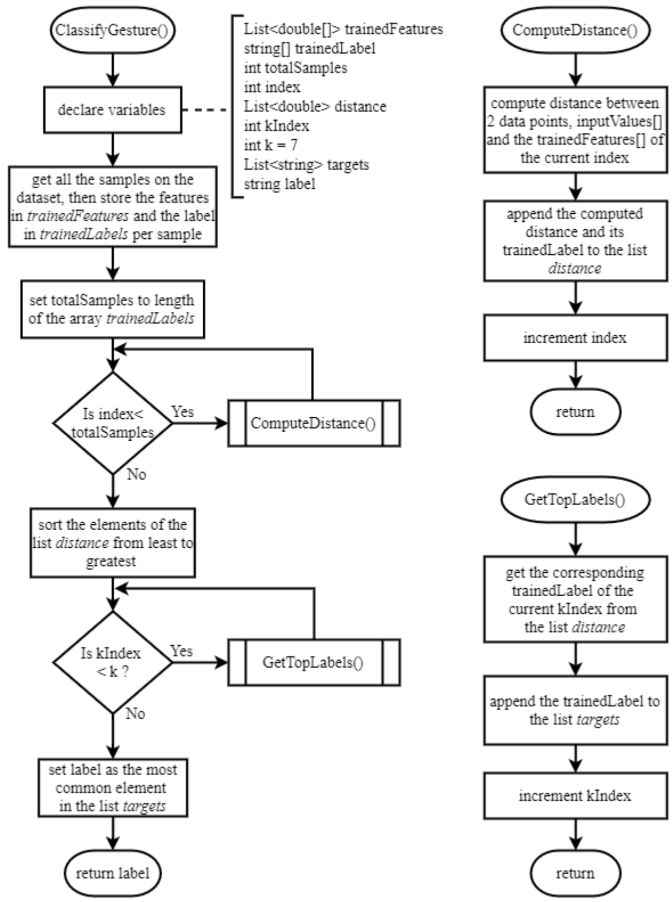

Figure 12 Classify Gesture Module

After preparing the input values to be used, classifying using KNN can now be performed. Figure 12 illustrates the simplified module that will be used to classify hand gestures to its equivalent label. The features and labels of each sample in the dataset will be prepared. Once the preparation is done, the distance between the features of the input value and each of the sample's features will be taken using the Euclidean distance formula. Each computed distance will be stored in a list, along with the assigned label. The list containing the distance will be sorted in ascending order. The top k least distance will be taken, and their assigned labels will be stored in a list. The most common element in the list containing the top k labels will be the equivalent label for the given input value.

III. Testing, Presentation, and Interpretation of Data

A. *Picking the K value*

Picking the value of K that will be used for the K-Nearest Neighbor algorithm is important as it affects the result of each classification. A small value of K will result in a noise having a large effect on the results, while a large value of K will make it computationally expensive for the device it will be applied to [12][13].

To pick the K value that will be used for the Android Application, the following steps are done.

1. A sample of sensor values from the gloves for each label was acquired through serial communication between the gloves and the computer using PuTTY.
2. The samples are prepared by placing it on a text file.
3. The text file containing the samples undergo K-Nearest Neighbor classification using the created dataset and using different values of K.
4. The result of the classification is recorded and the score for each value of K is counted.
5. There are 5 trials. Repeat steps 1 to 4 for each trial.

Table I
Accuracy of K

Trial	Value of K									
	1	3	5	7	9	11	13	15	17	19
1	23	24	23	23	21	23	22	22	21	20
2	24	24	22	23	23	23	23	23	23	23
3	23	26	24	26	24	22	24	24	22	23
4	24	23	23	23	22	23	22	23	21	21
5	24	23	21	21	22	21	21	21	21	20
Total	118	120	113	116	112	112	112	113	108	107

The table shows that the K value of 3 gives the highest total of correct classification. However, a small value of K means that noise will have a higher influence on the result. Thus, the K used for the KNN algorithm is 7, since it has the highest total among the value of k that is higher than 3.

B. *Testing the Android Application*

The application will be tested to measure its accuracy in translating FSL to text, this will be done by connecting the gloves to the created application to begin translating. The accuracy will be computed by dividing the number of successful attempts to the total attempts multiplied by 100.

This test will measure the accuracy of the system in translating Filipino sign language to text to determine how many times the gestures being translated is correct with the use of the created android application

To test the functionality of the system the following steps are done.

1. Install the application to the Android Phone.
2. Wear the device properly.
3. Pair the android device to the gloves.
4. Tap the translate button on the application to go to the translate page.
5. Tap the FSR for the gloves to be able to send data to the application.
6. The translated sign language will then be displayed to the application.
7. There are 10 attempts for each label, the number of successful attempts will be recorded, and the accuracy will be computed

TABLE II
TESTING THE ANDROID APPLICATION

Label	Number of Successful Attempts	Number of Attempts	Accuracy
	Using K = 7		
A	10	10	100
B	10	10	100
C	10	10	100
D	9	10	90
E	7	10	70
F	10	10	100
G	10	10	100
H	9	10	90
I	10	10	100
J	6	10	60
K	10	10	100
L	10	10	100
M	9	10	90
N	10	10	100
O	10	10	100
P	10	10	100
Q	10	10	100
R	4	10	40
S	9	10	90
T	6	10	60
U	9	10	90
V	6	10	60
W	10	10	100
X	8	10	80
Y	10	10	100
Z	10	10	100

Based on the collected data, the system shows an average of 89.63% accuracy for translating FSL to text using the KNN algorithm. Most of the letters in the alphabet have a 100% accuracy in translation except for D, E, H, J, M, R, S, T, U, and X, with R having the lowest accuracy of 40%.

IV. CONCLUSION AND RECOMMENDATION

The design project of creating a Filipino Sign Language (FSL) to text converter was successfully developed. The system uses a flex sensor, gyroscope, and accelerometer which are used to capture hand movements and gestures that are used to interpret a sign language gesture. To be portable, the researchers utilized the portability of an android device and created an application that can translate FSL to text. The K-Nearest Neighbor (KNN) falls in the family of supervised learning, a labeled dataset consisting of the sensor values alongside with the corresponding label was created and used to classify an input from the user, the system shows an average of 89.63% accuracy for translating FSL to text using the KNN algorithm.

The prediction of the KNN algorithm is already acceptable; however, further improvements of the system could be done using the following.

(1) Use of flex sensors for each joint of the fingers to make each hand gesture more specific from one another.
(2) Placing the Force Sensitive Resistor to the wrist for easier hand gesture.
(3) The researchers recommend the use of a larger dataset for a better prediction.
(4) The researchers also recommend the collection of data with different hand sizes for much better accuracy.

REFERENCES

[1] World Health Organization (WHO). (2018, March 15). Deafness. Retrieved from World Health Organization: https://www.who.int/news-room/facts-in-pictures/detail/deafness#:~:text=There%20are%20466%20million%20people,rise%20to%20over%20900%20million

[2] Inquirer, P. D. (2016, November 6). In The Know: Sign languages. Retrieved February 27, 2019, from https://newsinfo.inquirer.net/302184/in-the-know-sign-languages

[3] Esposa, J. I., Jr. (2017, August 05). Difference between ASL and FSL. Retrieved February 27, 2019, from https://deafphilippines.wordpress.com/2017/08/03/difference-between-asl-and-fsl

[4] Piscassio (2014) A heuristic decision tree algorithm for Filipino Sign Language (FSL) translator implemented using accelerometers with Bluetooth connectivity to a host PC.

[5] Balbin, J. R., Padilla, D. A., Caluyo, F. S., Fausto, J. C., Hortinela, C. C., Manlises, C. O., . . . Ventura, L. T. (2016). Sign language word translator using Neural Networks for the Aurally Impaired as a tool for communication. 2016 6th IEEE International Conference on Control System, Computing, and Engineering (ICCSCE).

[6] Shrenika, S., & Bala, M. M. (2020). Sign Language Recognition Using Template Matching Technique. 2020 International Conference on Computer Science, Engineering, and Applications (ICCSEA).

[7] Cloutier, P., Helderman, A., & Mehilli, R. (2016). Prototyping a Portable, Affordable Sign Language Glove.

[8] Lom, J. B. (2016). Flex Sensor Hookup Guide. Retrieved March 3, 2019, from https://learn.sparkfun.com/tutorials/flex-sensor-hookup-guide/all

[9] Sadam (2018, March 9). MPU6050 Gyro Sensor Interfacing with Arduino. Retrieved March 2, 2018, from https://circuitdigest.com/microcontroller-projects/mpu6050-gyro-sensor-interfacing-with-arduinoMori, S. (2008). Economic Livelihood of the Deaf in the Philippines. Poverty Reduction for the Disabled in the Philippines, 159-164.

[10] Harrison, O. (2018, September 10). Machine Learning Basics with the K-Nearest Neighbors Algorithm. Retrieved March 5, 2019, from https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

[11] A Complete Guide to K-Nearest-Neighbors with Applications in Python and R. (2016, July 13). Retrieved September 06, 2020, from https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/

[12] Girish, M. (2018, March 23). Bluetooth Basics: How to Control an LED Using an Android phone and Arduino. Retrieved March 1, 2019, from https://maker.pro/arduino/tutorial/bluetooth-basics-how-to-control-led-using-android phone-Arduino

[13] Ada, L. (2015, November 19). Force Sensitive Resistor (FSR). Retrieved March 1, 2019, from https://learn.adafruit.com/force-sensitive-resistor-fsr/using-an-fsr