Available online at:https://ijact.in

Date of Submission	20/01/2019
Date of Acceptance	19/02/2019
Date of Publication	28/02/2019
Page numbers	3053-3058 (6 Pages)

ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

USER TRAINABLE SIGN LANGUAGE TO SPEECH GLOVE USING KNN CLASSIFIER

Shwetha V, Dr.Vijayalaxmi, Dhanin Anoop Asarpota, Himanshu Verma
Dept. of Electrical & Electronics Engineering, Manipal Institute of Technology,
Manipal Academy of Higher Education, Manipal, Karnataka, India-576104
Shwetha.V@manipal.edu

Abstract:A sizable population around the world has some form of hearing or speaking disability. This creates a communication barrier among them and the rest of the world. Sign language was introduced to bridge this gap. The objective is to design a glove that can help translate sign language to text and that can be trained by the user itself if required. To achieve this a glove was designed using five flex sensors, three contact sensors and an accelerometer. Flex sensors were chosen as they are resistive devices that change resistance when bent. Due to their compactness, they can also be easily put on to a glove along with contact sensors and an accelerometer. The data from these sensors is then fed to an Arduino where it is read and processed before being sent to MATLAB via Bluetooth. After getting the values, the smart gesture detection algorithm must be designed so as to improve accuracy. To do this the data from the Arduino is first used to train a KNN model for classification. The model created after training is then used for classification of the gestures. A GUI was designed with control signals that allows the user to make a word from these gestures and then the word is converted to speech. The glove accurately gives us data points that can be used to classify various gestures of the American Sign Language. The interactive GUI developed in MATLAB enables a user to easily use the glove to make and/or edit a word created by the user and then recite it out on a speaker.

Keywords: Sign Language, Smart Glove, K Nearest Neighbour (KNN) Graphical User interface (GUI).

I. INTRODUCTION

Around 70 million people worldwide are deaf.They communicate with the outside world by using sign language.This poses a barrier for them as they must have either an interpreter with them at all times or must hope the person they communicate with knows sign language.With advancements in technology, we can think of making a smart glove that overcomes the problem a user faces. To achieve this we must design this glove to be compact, comfortable and have a high accuracy for detecting gestures made.

We used the American Sign Language as a reference for the gestures to be converted to speech because of its large user base. American Sign Language includes facial expressions, two-handed gestures and the orientation of hand with respect to the torso for the complete sign language. However, we only look at the alphabetical part of the American Sign Language that is made using one hand scientific disciplines[1].

The basic idea of the designing the glove based system for data collection and gesture recognition using different sensors and measuring the characteristics are discussed in the work[2,3] and also having the consideration with people of different hand sizes and shapes and a factors such as finger angular bend with respect to resistance change using different sensors are compared which helped in the selection of required sensors in this work [4] which is alsoa

low power design in compare with the earlier work which are cost effective but under developed [5][6]
Using image recognition system and Kinetic sensors are used in the work [7]which includes heavy data processing for the greater accuracy and lack of portability and restricted particular area the for the communication and image recognition system presented In this work [8] which was able to recognize the 15 letters of Arabic manual alphabets with the 83.7% of accuracy
Capacitive touch sensor and Raspberry pi embedded system is used to translate the hand gestures presented [9] and Piezo sensor are attached with the glove when the sensor is bended to certain degree its resistance starts to decrease intern decreases voltage associated with the sensors problem occurred with respect to speed and stability and response is very slow with unfolded gesture [10] System designed using flexi sensor and PIC microcontroller implemented in [11][12] which was able to recognize 20 out of 26 letters of American Sign Language and give us an idea of using flexi sensor based glove system for our prototype
Neural network interface used to recognize and classify the basic four hand glove gestures [13] give an idea of soft based classification is better and more reliable and adaptive once the system designed and trained KNN algorithm is simple among all other classification algorithm [14] is implemented.
Our design involves the use of flex sensors. These sensors are conductive polymer based sensors. They are made using resistive polymer on a side. The length of these polymers increases when the sensor is bent thus increasing the resistance of the sensor. This change in resistance gives us the bend angle of the finger. Also, American sign language uses motion to classify some gestures, hence to detect these motion we use an accelerometer which gives us acceleration in the X, Y and Z plane.
Software for the accurately detects these gestures. We used classification algorithm of machine learning since their libraries are easily available and have tutorials on how to use them.

II. Glove Design and Gesture Recognition

Flex sensors are used to provide the bend level of the finger. We use these sensors to detect the finger position. Also, an accelerometer is mounted on the glove that gives us spatial coordinates for the hand. This is to identify some characters of AmericanSign Language where movement is involved of the finger. All the input data from the sensors is in an analog format so it has to be converted to the digital domain using an ADC. For example, to differentiate between D, G and Z the accelerometer values play a vital role. Contact sensors help to differentiate between C and O, U and V, U and R. All the input data from the sensors is in an analog format so it has to be converted to the digital domain using an ADC.After we have to get the values we must now use these values to either classify the gesture or store it for further mapping. If the training phase is going on we must store the value. If the detection phase is going on we must take the sensor inputs and then classify them using an algorithm. The proposed methodology is given in Figure 1.

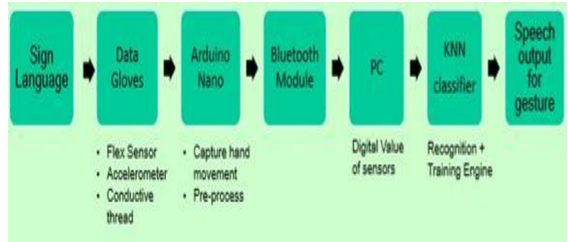

Figure 1: Proposed System using MATLAB

2.1 Hardware Implementation

For the hardware, we use five flex sensors, four conductive thread pieces stitched to make contact sensors and an accelerometer as the sensors that give us analog inputs for the orientation of the complete hand glove circuit diagram as shown in Figure 2.

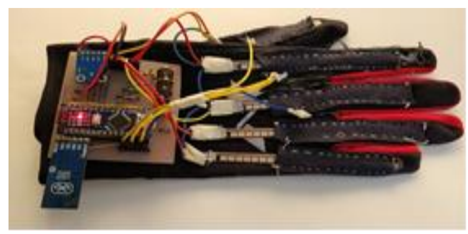

Figure 2: Final Glove

A major challenge while designing the hardware is the way the sensors should be mounted on the fingers. We tried sticking the sensors with double-sided tape, tape the sensor on starting and ending points and sticking it on all points except the joints but none of the designs gave us the right amount of the flexibility. Finally, the best solution was to stitch a sleeve on the top of the glove that would hold the flex sensor in place while allowing a user to freely bend their fingers. Circuit diagram as shown in figure 3.

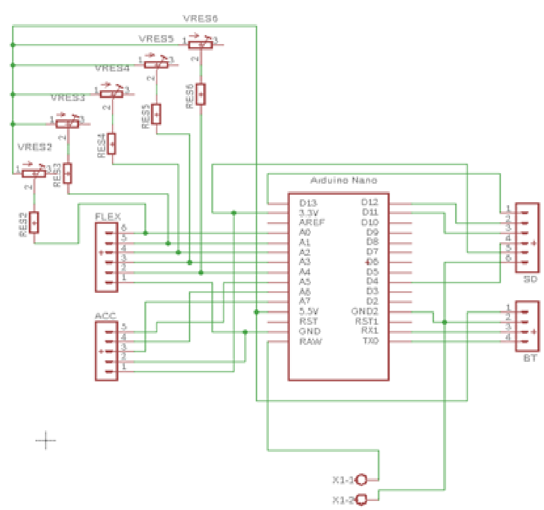

Figure 3: Circuit Schematic

The flex sensor values cannot read directly, hence we had to design a voltage divider circuit where voltage would be measured across the sensor. While testing the sensors we found that the bent and straight resistance of a sensor varies for the same sensor length. Hence to find optimal resistance for the sensors a MATLAB program developed. These resistances gave us the maximum range of values when used in the voltage divider circuit for each flex sensor.

The next component is the contact sensor. This was added to increase accuracy for some gestures like U, R and S. To reduce the cost of the glove we used conductive thread to make our own sensor rather than buying a readymade one from the market. It is also ergonomic. The design involved stitching conductive thread on the thumb, index and middle finger. The index finger is grounded to a digital pin on the Arduino board whereas the thumb and middle finger is connected to a digital pull-up input. When the thumb or middle finger touches the index finger, the voltage across the thumb or middle drops to zero and a touch is registered. This is detected by the digital pins the thread is connected.
The last component is the accelerometer. It is connected directly to the PCB designed due to its compact nature. For processing this information from the sensors an Arduino board is used due its fast development and ease of use. This board will have the algorithm built into it to read the values, process these and pass it on further. A moving average filter is applied to flex sensor and accelerometer raw values. This acts as a lowpass filter.

Final Glove as shown in Figure 2 is implemented along with the Bluetooth module HC-05 is connected to the Arduino. This module helps transfer information from the Arduino to the MATLAB program. This data is then further processed on MATLAB and displayed to a user via the GUI developed.

2.2 Software Implementation

For the software, we explored two methods. The first involves implementing the algorithm on the Arduino itself. The second method involves the Arduino doing the Analog to data conversion and pre-processing the data before sending it to MATLAB as shown in Figure 4 and Figure 5.

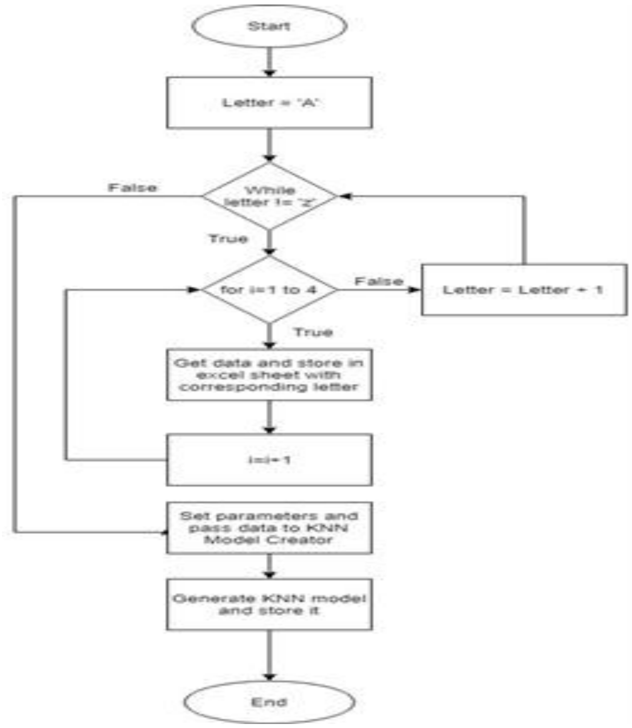

Figure 4: Flowchart for creating KNN model

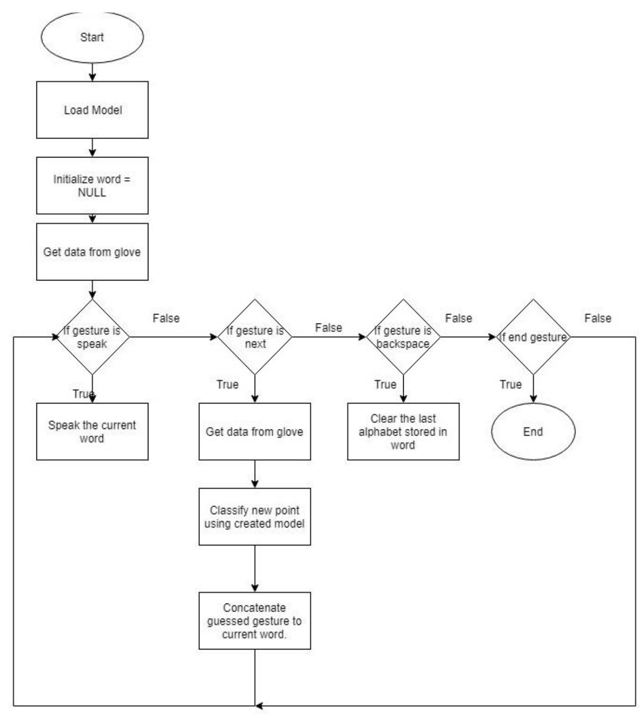

Figure 5: Flowchart for working of detection of gesture

We chose the former method due to Arduino having very limited onboard processing capability. In this, the Arduino takes data input from all the sensors and calculates an average of four values using a moving average filter. It then waits for a data request from the program and then sends the stored average values to MATLAB via Bluetooth.

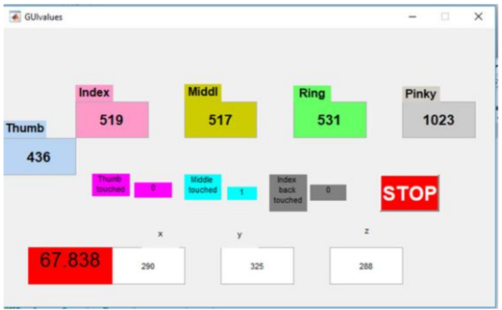

Figure 6: MATLAB GUI to display all values of sensor

To classify the gestures, we make use of K Nearest neighbor algorithm. First, we must make a model using this algorithm and use this model we can classify the data sent from the Arduino. To make a model we take four instances of all the data from the Arduino for each gesture of the American Sign Language. This is then stored in an Excel file. Using this value now a KNN model is made in MATLAB using k value as 3 and Euclidean distance as the distance type. Now, after the model is ready the second step is importing this model and using it to predict the gestures. For this, we developed a GUI with control gestures that guess each gesture after the "NEXT" control gesture is made. It then speaks the word out when the "SPEAK" control gesture is made. If there is an error, there is the "BACKSPACE" gesture. These three additional gestures were added in by us to allow more flexibility to the user in case any a gesture was unclassified, or an incorrect gesture was made by the user.

	Flex Sensor Value					Contact Sensor Value			Accelerometer Value		
Alphabet	Thumb	Index	Middle	Ring	Pinky	Thumb	Index	Middle	X	Y	Z
A	541	575	557	555	793	0	0	1	268	317	325
B	490	516	505	517	555	1	0	1	269	314	323
C	542	548	539	533	589	1	0	1	273	313	313
D	520	525	556	552	778	1	1	1	268	323	323
E	565	565	540	542	588	1	0	1	271	306	326
F	457	555	506	516	546	0	1	1	267	322	327
G	432	528	558	553	596	1	1	1	308	392	344
H	425	525	512	548	590	1	0	1	308	392	346
I	621	579	560	551	562	0	0	1	270	314	321
J	577	569	552	544	547	0	0	1	400	400	400
K	431	516	521	547	604	1	1	1	271	327	312
L	390	519	554	548	598	1	1	1	272	319	312
M	579	575	547	541	605	1	0	1	281	323	297
N	560	554	540	534	567	1	0	1	271	329	313
O	560	554	540	534	568	0	0	1	272	329	312
P	451	519	523	546	589	1	1	1	336	350	275
Q	445	545	551	547	592	1	1	1	339	346	275
R	475	524	519	546	586	1	1	0	267	324	325
S	614	582	562	552	607	0	0	1	269	312	352
T	645	557	550	547	585	1	1	1	272	333	312
U	530	518	515	547	586	1	0	1	269	328	317
V	516	518	513	545	580	1	1	1	268	327	320
W	536	521	506	526	569	1	1	1	278	319	301
X	561	551	546	544	584	1	1	1	274	321	306
Y	413	577	557	549	558	1	0	1	267	320	327
Z	530	528	552	547	605	1	1	1	400	400	400

Figure 7: A sample training Set

III. RESULTS AND DISCUSSIONS

3.1 Calculation of Optimal Resistance for Flex Sensor

During testing of the sensors, we found that the bent and straight resistances of each varies in Table 1. Hence to increase the accuracy of reading in the voltage divider circuit we had to ensure that an optimal resistance was chosen for the sensor. We used MATLAB to find optimal resistances for each sensor and then used those values for the voltage divider circuit.

	R_Straight	R_Bent	R_Optimal	R_Used
Thumb	37.5	67.5	53.6	53.4
Index	8.9	10.6	9.7	9
Middle	8.6	10	9.3	9
Ring	12.2	13.9	13	13.5
Pinky	8.3	9.4	8.8	9

Table 1: Flex Sensor Optimal Resistance (K-Ohm)

Once we determined optimal resistance for each flex sensor we designed a PCB for the same. This PCB contained standard resistors values along with variable potentiometer for the fine-tuning of all the flex sensors. This gives better control over the resistance value. Another advantage is that if a sensor is changed there is no need to change the resistor mounted on PCB only the potentiometer value can be adjusted to achieve the new resistance.

Now for testing the working of the contact sensors, flex sensors and accelerometer we developed a GUI as shown in the Figure6 that read values at an interval of 0.5 seconds continuously and displayed them. This also allows us to debug any loose or broken connection since those sensors will give an unexpected output or no output at all.

The above methods allowed us to design the circuit for the glove along with providing us with a way to debug the hardware components on the glove quickly and efficiently. With the hardware complete we must now make the software that classifies these gestures. We chose the KNN algorithm for classification since it is fast and efficient. This algorithm is a lazy learner thatis there needs to be no training done for the algorithm to function beforehand. As a new point enters only then are the calculations done to classify it. K Nearest Neighbor algorithm works on the principle of a calculating distance of a new data point from the existing points stored. These existing points are clustered into groups according to their predictor labels. The least distance among all clusters is then found out and that is the classified cluster/gesture of the data point. For our glove, we have 11 dimensions of classification so we use a MATLAB machine learning library to make the K Nearest neighbor and algorithm.

We decided out parameters for the model we create. For the data set required for classification, we use loop. Inside the loop a user makes gestures for each alphabet. We take 4 readings of each of those alphabets and store them in an excel sheet. A sample average of each alphabet is shown in Figure 7.

Using four such training set we now have a total of 104 data sets for all the alphabets. Now we use the model created and the more expanded data set to classify every incoming gesture. To do this we created a GUI as shown in Figure 8. This GUI joins gestures made by a user to make a word and then converts the word to speech once a user puts the control signal of the same.

Figure 8:Final GUI for User

Apart from designing a GUI for a user to use now faced the problem of how to measure accuracy for the glove totally using this algorithm. To solve this problem, using the same program to get data we captured 2 more such training set i.e. 208 total data sets. We then check the accuracy of each data row with the created model. This allows us to fine-tune parameters in the model and if the machine learning algorithm is changed we can easily check accuracy. Using this method, we get an accuracy of 89 % in MATLAB as presented in Figure 9. And it falls only due to particular letters.

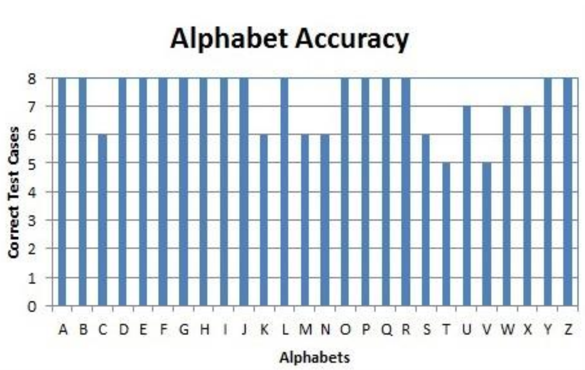

Figure 9: Result of classification of test set

IV. CONCLUSION

In this work, we designed a hand glove using flex sensor, contact sensors, and accelerometer as inputs to a KNN algorithm that would classify it. These sensors have very high accuracy and their outputs are accurately reproducible hence they were chosen. Another advantage of the glove is that if sensors are replaced the user can easily recalibrate the glove. We then design a classification algorithm using K Nearest Neighbours. We first use data set to make a model of the classifier and then using this model we classify the gesture.On checking accuracy with test sets we found out that our accuracy increase to 89 % for the glove. For the future, we can try to use different sensors to reduce cost. Also, we can implement a 2-hand glove implementation that makes words and sentence using advanced concepts of machine learning.

V. REFERENCES

[1] Anusha L, Usha Devi Y (2016) Implementation Of Gesture Based Voice And Language Translator For Dumb People. International Conference on Communication and Electronic Systems (ICCES) ISBN:978-1-5090- 1066-0.

[2] D. J. Sturman and D. Zeltzer. "A Survey of Glove-based Input", IEEE Comput. Graph. Appl., 14(1), pp. 30-39, 1994.,

[3] S. S. Fels and G. E. Hinton. "Glove-talk: A Neural Network Interface between a Data-glove and a Speech Synthesizer", IEEE Trans. Neural Networks, 4(1), pp. 2-8, 1993

[4] N.Tubaiz,T,Shanableh and K Assaleh,"Glove-Based Continous Arabic Sign Language Recognition in User-Dependent Mode,"IEEETranscation on Human-Machine Systems,Vol.45,no.4,pp.526-533 Aug.2015.

[5] L. K. Simone, D. G. Kamper, "Design considerations for a wearable monitor to measure finger posture", J. Neuro Eng. Rehabil., vol. 2, no. 1, pp. 5-5, 2005

[6] R. Gentner, J. Classen, "Development and evaluation of a low-cost sensor glove for assessment of human finger movements in neurophysiological settings", J. Neurosci. Methods, vol. 178, no. 1, pp. 138-147, 2009

[7] Zhou Ren ,Junsong Yuan ,Jingjing Meng , Zhengyou Zhang," Robust Part-Based Hand Gesture Recognition Using Kinect Sensor" IEEE Transactions on Multimedia ,Volume: 15 Issue: 5, pp:1520-9210, 2013.

[8] N. El-Bendary, H. Zawbaa, M. Daoud, A. ellaHassanien, K. Naka-matsu, "Arslat: Arabic sign language alphabets translator", Computer Information Systems and Industrial Management Applications (CISIM) 2010 , pp. 590-595, Oct 2010.

[9] Kalpattu S. Abhishek, Lee Chun Fai Qubeley and Derek. Ho,"Glove-based hand Gesture recognition sign language translator using capacitive touch sensor", IEEE International Conference on Electron Devices and Solid-State Circuits EDSSC), 2016

[10] EunseokJeong ,Jaehong Lee ,DaeEun Kim "Finger-gesture Recognition Glove using Velostat" 11th International Conference on Control, Automation and Systems, Oct. 2011

[11] M. Elmahgiubi ,MEnnajar ,N.Drawil and M.S Elbuni "Sign language translator and gesture recognition",Comuter& information technology (GSCIT),2015 Global Summit on Sousse,2015,PP.1-6

[12] Jamal Haydar, Bayan Dalal,, Shahed Hussainy,, Lina El Khansa, Walid Fahs," ASL Fingerspelling Translator Glove" IJCSI International Journal of Computer Science Issues, 2012,Vol. 9, Issue 6, No 1, PP-1694-0814

[13] S. Sidney and E. Geoffrey, "Glove talk—a neural-network interface between a data-glove and a speech synthesizer," IEEE Transactions on Neural Networks, vol. 4, no. 1, pp. 2–8, 1993

[14] Ahmed H. Aliwy1 and Esraa H. Abdul Ameer,Comparative Study of Five Text Classification Algorithms with their Improvements.International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 14 (2017) pp. 4309-4319.