TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

Kathmandu Engineering College

Department of Electronics and Communication Engineering

Smart Gloves

[Code No: EX 755]

PROJECT REPORT SUBMITTED TO

THE DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

IN PARTIAL FULLFILLMENT OF THE REQUIREMENT FOR THE BACHELOR'S

DEGREE IN ELECTRONICS & COMMUNICATION ENGINEERING

By

Aakriti Basnet	/071BEX001
Adarsh Ghimire	/071BEX005
Anushma Shrestha	/071BEX010
Bhupendra Kadayat	/071BEX014

Kathmandu, Nepal

2075

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

Kathmandu Engineering College

Department of Electronics and Communication Engineering

CERTIFICATE

The undersigned certify that they have read and recommended to the Department of Electronics and Communication Engineering for acceptance, of a final year project report entitled **"SMART GLOVES"** submitted by Aakriti Basnet (71001), Adarsh Ghimire (71005), Anushma Shrestha (71010) and Bhupendra Kadayat (71014) in partial fulfilment of the requirements for the Bachelor's degree in Electronics & Communication Engineering.

Er. Suramya Sharma Dahal
Project Supervisor
Lecturer
Department of Electronics and
Communication Engineering
Kathmandu Engineering College

Prof. Dr. Ram Krishna Maharjan
External Examiner
Department of Electronics and
Communication Engineering
Pulchowk Campus
Institute of Engineering (IOE)

Er. Surendra Khatri
Project Coordinator
Department of Electronics and
Communication Engineering
Kathmandu Engineering College

Er. Sagun Manandhar
Head of Department
Department of Electronics and
Communication Engineering
Kathmandu Engineering College

COPYRIGHT

ACKNOWLEDGEMENT

We would like to show our gratitude to **Er. SURAMYA SHARMA DAHAL, MAJOR PROJECT SUPERVISOR, KATHMANDU ENGINEERING COLLEGE, TRIBHUVAN UNIVERSITY** for giving us a good guidelines for project throughout numerous consultations.

We appreciate the help received from the Department of Electronics and Communication Engineering of Kathmandu Engineering College. Similarly, we are thankful to, Er. Surendra Khatri and Er. Sanjeevan Satyal for providing us idea regarding the project and ensuring that we meet the deadlines on time.

Lastly, we would also like to express our gratitude to all the faculty members of Electronics and Communication Engineering for their valuable comments and suggestions.

ABSTRACT

Everyday communication with the hearing population poses a major challenge to those with hearing loss. For this purpose, an automatic Sign Language recognition system (model) is developed using Random Forest Classifier and to translate the sign language alphabets and common words into text and sound. A glove circuit is designed with flex sensors, 3- axis accelerometer and gyroscope to capture the gestures. The finger bending data is obtained from the flex sensors on each finger whereas the accelerometer provides the trajectories of the hand motion. Some local features are extracted from the sign language alphabets and common words which are then classified using machine learning. The main purpose of this Smart Gloves is to provide an ease of sharing basic ideas, minimized communication gap and an easier collaboration for the hard of hearing people.

TABLE OF CONTENT

Title Page No.

COPYRIGHT iii

ACKNOWLEDGEMENT iv

ABSTRACT v

LIST OF FIGURES viii

LIST OF ABBREVIATION ix

CHAPTER 1: INTRODUCTION 1

1.1 Background 1

1.2 Problem Statement 2

1.3 Objective 2

1.4 Scope or Application 2

CHAPTER 2: LITERATURE REVIEW 4

CHAPTER 3: THEORITICAL BACKGROUND 9

3.1 Hardware 9

3.1.1 Arduino Mega 9

3.1.2 Flex Sensor 10

3.1.3 MPU-6050 11

3.1.3.1 Accelerometer 13

3.1.3.2 Gyroscope 14

3.2 Software 15

3.2.1 Machine Learning 15

3.2.1.1 Supervised Learning 16

3.2.1.2 Unsupervised Learning 17

3.2.2 Random Forest Classification 17

CHAPTER 4: METHODOLOGY 20
4.1. Hardware assembling 20
4.2. Dataset Preparation 20
4.2.1. Dataset Collection 20
4.2.2. Data Preprocessing 20
4.2.3 Dataset description 21
4.3 System Block Diagram 21
4.4 Explanation of Block Diagram 23
4.5 Algorithm 24
4.5.1 Algorithm for dataset preparation at Arduino Mega 2560 24
4.5.2 Algorithm for data set preparation at python end 24
4.5.3 Algorithm for real time application 25
4.6 Flowchart 26
4.6.1 Flowchart for Dataset preparation using Arduino Mega 2560 26
4.6.2 Flowchart for data set preparation using python 27
4.6.3 Flowchart of real time application 28
CHAPTER 5: RESULT AND ANALYSIS 29
5.1 Result 29
5.2 Analysis 29
CHAPTER 6: EPILOGUE 36
6.1 Conclusion 36
6.2 Future enhancement 36
References 37

LIST OF FIGURES

3.1 Pin out diagram of Arduino Mega 2560……………………………………….9

3.2 Flex Sensor………………..……………………………………….………….11

3.3 MPU 6050 Module…………..……………….………………………………...12

3.4 Tilt or inclination of MPU 6050 accelerometer……..……………………………..13

3.5 Orientation and Polarity of MPU 6050…………………………………………....14

3.6 Machine learning techniques..……………………….…………………………16

4.1 Feature extraction Diagram..…………………………………………………….21

4.2 System Block Diagram……………………………………...………………….22

4.3 Flowchart for dataset preparation using Arduino…………………………………….26

4.4 Flowchart for dataset preparation using python.…………………………………….27

4.5 Flowchart of real time application…………….…………………………………..28

5.1 Demonstration…………………………….…………………………………….30

5.2 Correlation plot for alphabet ‘a’……………….…………………………………….34

5.3 Correlation plot for alphabet ‘b’……………….…………………………………….34

5.4 Correlation plot for alphabet ‘e’……………….…………………………………….35

5.5 Correlation plot for alphabet ‘s’……………….…………………………………….35

LIST OF ABBREVIATION

AAC: Augmentative and alternative communication

AC: Alternate current

ADC: Analog to Digital Conversion

ANN: Artificial Neural Network

ARQ: Automatic Repeat Request

BP: Back Propagation

CSV: Comma separated values

DC: Direct current

DOF: Depth of field

DTMF: Dual Tone Multi Frequency

EMG: Electromyography

FPGA: Field Programmable Gate Array

HMM: Hidden Markov Model

ICSP: In circuit serial programming

ISL: Indian Sign Language

LED: Light Emitting Diode

MEMS: Microelectromechanical system

MSA: Mathematical sound architecture

PC: Personal Computer

PIC: Peripheral Interface Controller

PWM: Pulse width modulation

RGB: Red, Green and Blue

SVM: Support vector machine

UART: Universal asynchronous reception and transmission

CHAPTER 1: INTRODUCTION

1.1 Background

Human beings have a natural ability to see, listen and interact with their external environment. Unfortunately, there are some people who are differently abled and do not have the ability to use their senses to the best extent possible. Deaf and Dumb population is a result of the physical disability of hearing for deaf people and disability of speaking for dumb people. In the recent years, there has been a rapid increase in the number of hearing impaired and speech disabled victims due to birth defects, oral diseases and accidents. When a speech impaired person speaks to a normal person, the normal person finds it difficult to understand and asks the deaf-dumb person to show gestures for his/her needs. Dumb persons have their own language to communicate with us; the only thing is that we need to understand their language.

Sign language is used by deaf and mute people and it is a communication skill that uses gestures instead of sound to convey meaning simultaneously combining hand shapes, orientation and movement of the hands, arms or body and facial expressions to express fluidly a speaker's thoughts. But most of the time normal people find it difficult to understand this sign language. This presents a major roadblock for people in the deaf and dumb communities when they try to engage in interaction with others, especially in their educational, social and professional environments. Therefore, it is necessary to have an advance gesture recognition or sign language detection system to bridge this communication gap.

The people who cannot speak or have lost their ability to speak in some accident, it becomes difficult for them to convey their message within the society. To overcome this, a project called 'SMART GLOVE' has been designed. Giving a voice to the voiceless has been a cause that many have championed throughout history, but it's safe to say that none of those efforts involved packing a bunch of sensors into a glove. The main objective of this project is to help deaf and dumb people by removing communication barrier so they are not restricted in a small social circle and are able to convey their feelings and emotions whenever they want.

Smart glove is based on the wearable technology. It is basically a device which has some specific wearable sensors with phenomenal temperature stability. All the sensors are fitted on a glove which measures the different analog parameters associated with the movement of fingers and orientation of the hand during any particular gesture. These sensors read those particular analog values and coding is done in the microcontroller according to these values to recognize the corresponding sign language. The goal of this project is to develop a portable communication system having multiple sensors for Sign Language Recognition and to translate these gestures into text and sound.

1.2 Problem Statement

Deaf and normal person communication is as same as two different persons from different countries using two different languages for communication without any common language which leads to problem in communication. Sign language is the only communication tool used by deaf people to communicate to each other. However, normal people do not understand sign language and this creates a large communication barrier between deaf people and normal people. In addition, the sign language is also not easy to learn due to its natural differences in sentence structure and grammar. Therefore, there is a need to develop a system which can help in translating the sign language into text and voice in order to ensure the effective communication can easily take place in the community.

1.3 Objective

The objectives of our project are

- To build a glove embedded with sensors to read the sign language and convert it into text and speech
- Help to deaf and dumb people to communicate with normal people especially during emergency situation.

1.4 Scope or Application

The scope or application of the project are

- For all deaf and dumb people
- Institution for deaf and dumb people

1.6 Organization of Report

The report is organized in three parts: the introduction with acknowledgement, abstract, list of figures and abbreviation, the main body with information on the project, and a final part that describes the result and analysis of the project, identifies some future research needs and conclusion. To assist the reader, each part of this report, including the sections within each part, are organized in a similar manner and follow the same format and order of presentation. The format of this report is explained below.

Part I is the introduction. It consists of cover page, acknowledgement, abstract of the project, list of figures and abbreviations used throughout the report.

Part II is the main body of the report and is divided into several chapters.

Chapter 1: Introduction, explains the background and importance of the topic, problem statement, objective, scope or application.

Chapter 2: Literature Review, explains the past findings, research, case studies, projects and experiments done in the past.

Chapter 3: Related Theory, explains the theoretical background of the project and briefly explains the extraction of alphabets and words through gesture recognition using machine learning algorithm.

Chapter 4: Methodology, explains the system block diagram, algorithm and flowchart. It gives detailed explanation of the methods and steps used in making our project both in terms of hardware and software.

Part III focuses on result, analysis, conclusion and future research. It is divided into two chapters.

Chapter 5: Result and Analysis, explains the result of our project and it is analyzed whether our analysis meets our scope and objective.

Chapter 6: Epilogue, discusses the conclusion and future enhancements of the project.

At last references which have been used for certain inputs are listed after the key words. Wherever these references have been quoted / data or technical specifications taken in the text, these have been cross-referred by their serial number in the list of References.

CHAPTER 2: LITERATURE REVIEW

Enable Talk is a student project, whose main idea is to translate sign language into speech. The project was presented at the Microsoft Imagine Cup competition in 2012 at Sydney, Australia and won the first prize for software design competition [1]**.** The team was from country Ukraine with city Donetsk and school Computer Academy Step. The concept of the project consisted of two sensor embedded gloves and a mobile device, which entailed the recognition process.

Glove- based system is composed of an array of sensor, electronics for data acquisition or processing, power supply & a support for sensors that can be worn on user's hand [2]. LED glove, data glove, Sayre glove, cyber glove are the different type of glove used here. Glove based system helps user for selecting a particular glove for a particular application.

Glove Talk II is a system which translates hand gestures to speech, which is based on the gesture to format model developed by Sidney Fels and Geoffrey Hinton, Department of Computer Science of University of Toronto [3]. Neural networks were used to implement an adaptive interface, called Glove Talk II, which contains hand gestures to control the parameters of a parallel formant speech synthesizer to allow a user to speak with his hands. It is used to implement an artificial vocal tract. Glove-Talk-II is a system which translates hand gestures to speech through an adaptive interface. Hand gestures are mapped continuously to 10 control parameters of a parallel formant speech synthesizer. The mapping allows the hand to act as an artificial vocal tract that produces speech in real time. This gives an unlimited vocabulary, multiple languages in addition to direct control of fundamental frequency and volume. Currently, the best version of Glove-Talk II uses several input devices (including a Cyberglove, a Contact Glove, a polhemus sensor, and a foot-pedal), a parallel formant speech synthesizer and 3 neural networks [4]. The gesture-to-speech task is divided into vowel and consonant production by using a gating network to weight the outputs of a vowel and a consonant neural network. The gating network and the consonant network are trained with examples from the user. The vowel network implements a fixed, user-defined relationship between hand-position and vowel sound and does not require any training examples from the user. Volume, fundamental frequency and stop consonants are produced with a fixed mapping from the input devices.

Bend sensor modeling is used for motion recognition. The model is used to track human joint movement and it recovers the original signal waveforms, which shows the joint rotation .also for the fastest human speed. Bend sensor modeling is demonstrated that bend sensor can be applied for human posture recognition.

Harmeet Kaur, et al. in their paper, presented a brief description about the past attempts that were made to convert sign language to understandable form. In their paper, they have thoroughly scrutinized the previous attempts over this technology and also suggested various possible ways to implement the design of a simple smart glove [5].

Speak jet is sound synthesizer which is used to convert text data into voice [6]. It uses mathematical Sound Architecture (MSA) technique to control five channel sound synthesizer to generate a speech signal. It is having 72 speech elements, 43 sound effects and 12 DTMF touch tones.by using MSA component and also pitch, rate, bend, and volume parameter user can generate various sound effects. They tried to develop Electronic Speaking Glove, designed to facilitate an easy communication through synthesized speech for the benefit of speechless patients. Generally, a speechless person communicates through sign language which is not understood by the majority of people. The proposed system is designed to solve this problem. Gestures of fingers of a user of this glove will be converted into synthesized speech to convey an audible message to others, for example in a critical communication with doctors. The glove is internally equipped with multiple flex sensors that are made up of "bend-sensitive resistance elements". For each specific gesture, internal flex sensors produce a proportional change in resistance of various elements. The processing of this information sends a unique set of signals to the PIC microcontroller and speaks jet IC which is pre-programmed to speak desired sentences.

In a P5 Glove from Essential reality was used. It is an inexpensive (∽ 50 Euro) glove with integrated 6 DOF tracking designed as a game controller [7]. 6 DOF means six degrees of freedom, in fact the ability to move forward/backward, up/down, left/right (translation in three perpendicular axes) combined with rotation about three perpendicular axes (pitch, yaw, roll). The glove consists of five bend sensors to track the flexion of the wearer's fingers. An infrared-based optical tracking system is used to compute the glove position and orientation without the need for additional hardware. The glove is connected with a cable to the base station.

Tushar Chouhan et al. implemented wired interactive glove, interfaced with a computer running MATLAB or Octave, with a high degree of accuracy for gesture recognition. The glove maps the orientation of the hand and fingers with the help of bend sensors, Hall Effect sensors and an accelerometer. The data is then transmitted to a computer using automatic repeat request (ARQ) as an error controlling scheme. The system is modelled for the differently abled section of the society to help convert sign language to a more human understandable form such as textual messages. The hardware section of their proposed design has its constituent electronic components as bend sensor, hall-effect sensor, accelerometer and Machine Learning Algorithms Used for Gesture Recognition. The bend sensor outputs are fed to the analog multiplexer (HEF4051B by NXP Semiconductors). The output of this multiplexer is given to a current to voltage converter circuit. Since the voltage output of the Hall sensor is low, an amplifier is needed. Sensor outputs obtained are given to the inbuilt ADC (analog to digital converter) of MSP430G2553(by Texas Instruments)for sampling the values given by the sensors, which is also used for interfacing the glove with a computer running the machine learning algorithms. The data acquisition process starts with the processor sending control signals to multiplexer for receiving values from the different sensors sequentially and temporarily storing it in an array. These stored values are transmitted to the computer using universal asynchronous reception and transmission (UART) connection for further processing and decoding of the received signals. While transmitting the values in UART automatic repeat request (ARQ) scheme has been incorporated for avoiding the information loss because of transmission errors [8].

Abhishek Tandon, et al. in their paper presented a brief introduction of their proposed design of 'Smart Glove' along with the previous attempts done in the area of augmentative and alternative communication (AAC). The proposed design of their glove converts the Indian Sign Language (ISL) into text and speech. Their proposed design consists of five flex sensors, one for each finger of the hand. These flex sensors are connected to five analog inputs of the microcontroller. They used microcontroller to process input voltage of the flex sensors and send the desired text output to the android device (smart phone) using Bluetooth module. Their android device has a software application which can convert the text into audible (speech) signals [9].

M. Mohandes et al. proposed an image based system for recognizing Arabic sign language. First, the system detects the signer‘s face by using the Gaussian skin model. Centroid of the detected face region is taken as the origin for each frame and then the hands movement is tracked by applying region growing technique. Hidden Markov Model (HMM) is used to perform the classification of the signs during the recognition stage, through some computation based on the features extracted from the input images [10].

M. P. Paulraj et al. presented a simple sign language recognition system that is capable of recognizing nine phonemes in English using a machine vision system. The system had been developed based on skin color segmentation and Artificial Neural Network (ANN). There are three processing stages in the system; preprocessing, feature extraction and gesture classification. Skin color detection and region segmentation are carried out during the preprocessing stage. Skin color of the hand is detected based on the RGB values in the image frame. The feature extraction stage extracts moment invariant features, obtained by calculating the blob in the set of image frames, from the right and left hand gesture images. The gesture classification stage then uses these features as its input to ANN to recognize the sign. It is reported that the average recognition rate for this system is 92.85% [11].

Wang et al. presented a sign language recognition system that uses tensor subspace analysis to model a multi-view hand gesture. The hand recognition process is achieved through color segmentation. Input image that is in RGB color space is converted to YCbCr color space to ease the process of detecting the skin that employs the Back Propagation (BP) networks model. The sign language recognition is modeled and recognized using tensor. Then, the matching process is carried out to identify the input hand gesture [12].

Glove based systems on the other hand employs sensors attached to the glove captures the movement of the hand and finger and also the rotation. Many efforts have been made to interpret hand gestures, particularly the signals which are changing over time Hidden Markov Model (HMM) is employed as an effective tool in most of the works. A system with two data gloves and three position trackers as input devices and a fuzzy decision tree as a classifier is used to recognize the Chinese Sign Language gestures [13]. With a 5113 sign vocabulary, it achieves 91.6% recognition accuracy. The combined accelerometer and the surface electromyographic sensor provides an alternative method of gesture sensing unlike the above two methods mentioned previously. The Kinematic information of the hand and the arm are provided by the accelerometer and also it is capable of distinguishing the hand orientations or movements with different trajectories [14]. EMG Sensor measures the electrical activity provided by the skeletal muscles and the signal contain rich information for the coactivation and coordination of multiple muscles associated with different sign gestures. Recent works on ACC and sEMG have demonstrated the improved recognition performances. For instance, it is shown that the sEMG combined with ACC can achieve accuracy in the range of 5–10% for recognition of various wrist and finger gestures. The complementary functionality of both sensors is examined for the recognition of seven isolated words in German sign language. The intrinsic mode entropy is successfully applied on ACC and sEMG data acquired from the dominant hand to recognize 60 isolated signs in Greek sign language [15]. Recently, a wearable system using body-worn ACC and sEMG sensors is designed to remotely monitor functional activity in stroke.

CHAPTER 3: THEORITICAL BACKGROUND

3.1 Hardware

The hardware components used in our project are

3.1.1 Arduino Mega

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller.

In order to get started with it we need to simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or a battery. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

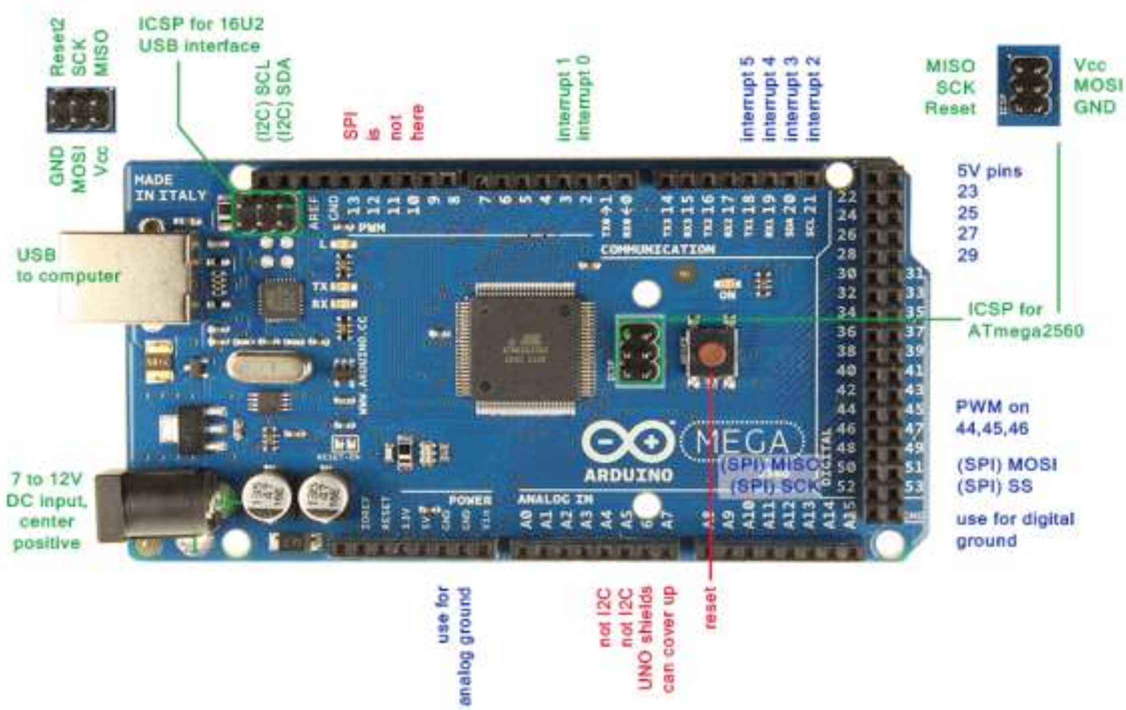


Figure 3.1 Pin out diagram of Arduino Mega 2560 [16]

3.1.2 Flex Sensor

Flex sensors are sensors that changes the resistance depending upon on the amount of bend on the sensor. They convert the change in bend to electrical resistance- the more the bend, the more the resistance value. They are usually in the form of a thin strip from 1” – 5” long that vary in resistance. They can also be made uni-directional or bi-directional. The flex sensors are made with the same principle as strain gauges (changing their resistance on the bending occasion), but they have large resistance differences. Inside the flex sensors are the carbon resistive elements within a thin flexible substrate when bent produces a resistance output relative to the bend. Flex sensors works in the principle of voltage divider form.

The resistance of the flex sensor increases as the body of the component bends. Sensors like these were used in the Nintendo Power Glove. They can also be used as door sensors, robot whisker sensors, or a primary component in creating sentient stuffed animals. Flex sensors are available in two sizes: one 2.2" (5.588cm) long and another 4.5" (11.43cm) long. When the flex sensors are left flat, these sensors will look like a 30kΩ resistor. As they bend, the resistance between the two terminals will increase to as much as 70kΩ at a 90° angle [17]. By combining the flex sensor with a static resistor to create a voltage divider, you can produce a variable voltage that can be read by a microcontroller’s analog-to-digital converter.

One side of the sensor is printed with a polymer ink that has conductive particles embedded in it. When the sensor is straight, the particles give the ink a resistance of about 30k Ohms. When the sensor is bent away from the ink, the conductive particles move further apart, increasing this resistance. When the sensor straightens out again, the resistance returns to the original value. By measuring the resistance, you can determine how much the sensor is being bent.

Figure 3.2: Flex sensor [18]

3.1.3 MPU-6050

The MPU-6050 is the world's first and only 6-axis Motion Tracking devices designed for the low power, low cost, and high performance requirements of smartphones, tablets and wearable sensors. The MPU6050 has an embedded 3-axis MEMS gyroscope, a 3-axis MEMS accelerometer. IMU sensors usually consist of two or more parts. The MPU 6050 is a 6 DOF (degrees of freedom) or a six-axis IMU sensor, which means that it gives six values as output: three values from the accelerometer and three from the gyroscope [19]. The MPU 6050 is a sensor based on MEMS (micro electro mechanical systems) technology. Both the accelerometer and the gyroscope are embedded inside a single chip. This chip uses I2C (inter-integrated circuit) protocol for communication. IMU sensors help us get the position of an object attached to the sensor in three-dimensional space. These values are usually in angles to help us to determine its position.

MPU6050 sensor module is complete 6-axis Motion Tracking Device. It combines 3-axis Gyroscope, 3-axis Accelerometer and Digital Motion Processor all in small package. Also, it has additional feature of on-chip Temperature sensor. It has I2C bus interface to communicate with the microcontrollers. It has Auxiliary I2C bus to communicate with other sensor devices like 3-axis Magnetometer, Pressure sensor etc. If 3-axis Magnetometer is connected to auxiliary I2C bus, then MPU6050 can provide complete 9-axis Motion Fusion output.

Figure 3.3 MPU 6050 Module [19]

INT: Interrupt digital output pin.

AD0: I2C Slave Address LSB pin. This is 0th bit in 7-bit slave address of device. If connected to VCC then it is read as logic one and slave address changes.

XCL: Auxiliary Serial Clock pin. This pin is used to connect other I2C interface enabled sensors SCL pin to MPU-6050.

XDA: Auxiliary Serial Data pin. This pin is used to connect other I2C interface enabled sensors SDA pin to MPU-6050.

SCL: Serial Clock pin. This pin is connected to microcontrollers SCL pin.

SDA: Serial Data pin. This pin is connected to microcontrollers SDA pin.

GND: Ground pin. This pin is connected to ground connection.

VCC: Power supply pin. This pin is connected to +5V DC supply.

MPU-6050 module has Slave address (When AD0 = 0, i.e. it is not connected to Vcc) as,

Slave Write address(SLA+W): 0xD0

Slave Read address(SLA+R): 0xD1

3.1.3.1 Accelerometer

The MPU6050 consist 3-axis Accelerometer with Micro Electro Mechanical (MEMs) technology. It used to detect angle of tilt or inclination along the X, Y and Z axes as shown in below figure.

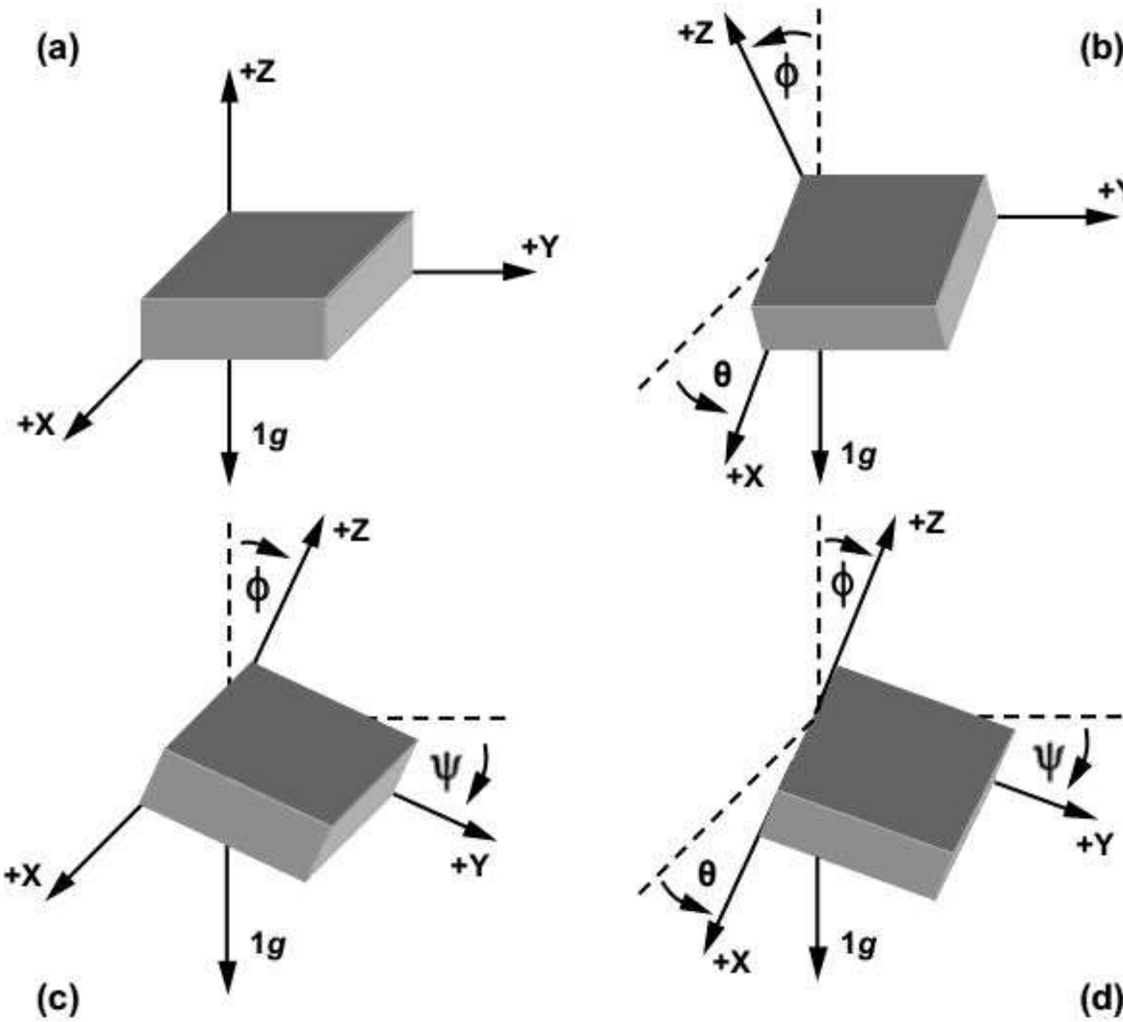


Figure 3.4: Tilt or inclination of MPU 6050 accelerometer [19]

An accelerometer is an electromechanical device that will measure acceleration forces. These forces may be static, like the constant force of gravity pulling at your feet, or they could be dynamic caused by moving or vibrating the accelerometer. By measuring the amount of static acceleration due to gravity, you can find out the angle the device is tilted at with respect to the earth. By sensing the amount of dynamic acceleration, you can analyze the way the device is moving. There are many different ways to make an accelerometer. Some accelerometers use the piezoelectric effect i.e they contain microscopic crystal structures that get stressed by accelerative forces, which causes a voltage to be generated. Another way to do it is by sensing changes in capacitance. If you have two microstructures next to each other, they have a certain capacitance between them. If an accelerative force moves one of the structures, then the capacitance will change. Adding some circuitry to convert from capacitance to voltage you will get an accelerometer.

Acceleration along the axes deflects the movable mass. This displacement of moving plate (mass) unbalances the differential capacitor which results in sensor output. Output amplitude is proportional to acceleration. 16-bit ADC is used to get digitized output. The full-scale range of acceleration are +/- 2g, +/- 4g, +/- 8g, +/- 16g. It measured in g (gravity force) unit. When device placed on flat surface it will measure 0g on X and Y axis and +1g on Z axis.

3.1.3.2 Gyroscope

The MPU6050 consist of 3-axis Gyroscope with Micro Electro Mechanical System (MEMS) technology. It is used to detect rotational velocity along the X, Y, Z axes as shown in below figure.

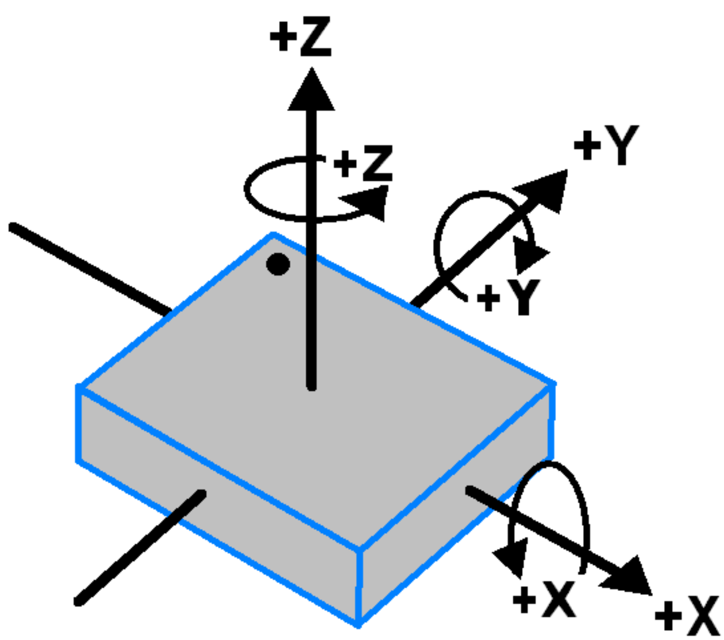


Figure 3.5: Orientation and Polarity of MPU 6050 Gyroscope [19]

A gyroscope is a device used for measuring or maintaining orientation and angular velocity [20]. It is a spinning wheel or disc in which the axis of rotation is free to assume any orientation by itself. When rotating, the orientation of this axis is unaffected by tilting or rotation of the mounting, according to the conservation of angular momentum. Gyroscopes work on the principle of Coriolis acceleration.

When the gyros are rotated about any of the sense axes, the Coriolis Effect causes a vibration that is detected by a MEM inside MPU6050. The resulting signal is amplified, demodulated, and filtered to produce a voltage that is proportional to the angular rate. This voltage is digitized using 16-bit ADC to sample each axis. The full-scale range of output are +/- 250, +/- 500, +/- 1000, +/- 2000. It measures the angular velocity along each axis in degree per second unit. Imagine that there is a fork-like structure that is in a constant back-and-forth motion. It is held in place using piezoelectric crystals. Whenever you try to tilt this arrangement, the crystals experience a force in the direction of inclination. This is caused as a result of the inertia of the moving fork. The crystals thus produce a current in consensus with the piezoelectric effect, and this current is amplified. The values are then refined by the host microcontroller.

3.2 Software

3.2.1 Machine Learning

Machine learning is a data analytical technique that teaches computers/machines to do what comes naturally to humans and animals: learn from experience. Machine learning algorithms can use computational methods to “learn” information directly from data without relying on a predetermined equation as a model. The algorithms adaptively improve their performance as the number of samples available for learning increases. It is a subset of artificial intelligence in the field of computer science that often uses statistical techniques to give computers the ability to "learn" (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed [21].

Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is sometimes conflated with data mining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning [22] [23]. Machine learning can also be unsupervised and be used to learn and establish baseline behavioral profiles for various entities and then used to find meaningful anomalies.

Machine learning uses two types of techniques: supervised learning, which trains a model on known input and output data so that it can predict future outputs, and unsupervised learning, which finds hidden patterns or intrinsic structures in input data.

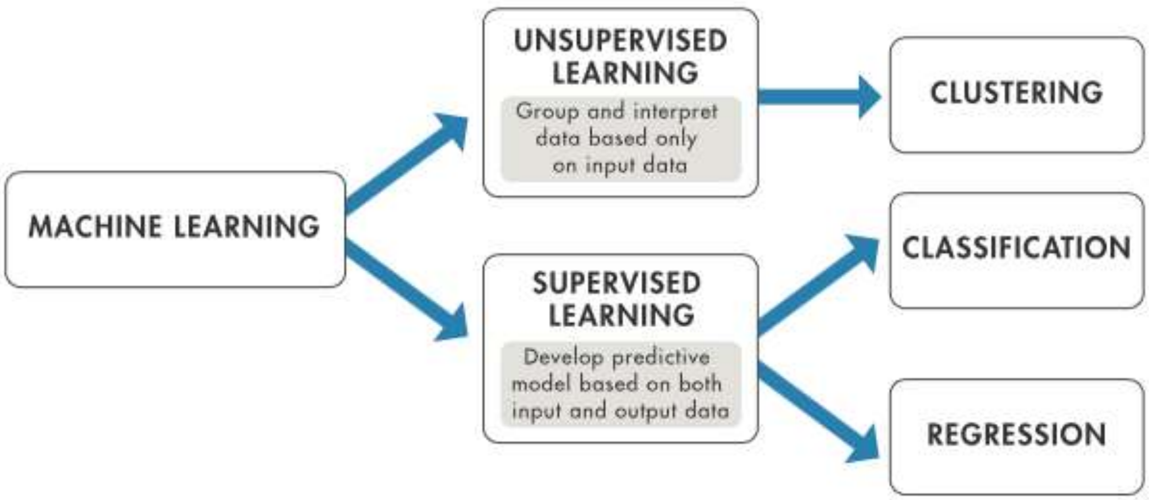


Figure 3.6: Machine learning techniques

3.2.1.1 Supervised Learning

Supervised machine learning builds a model that makes predictions based on evidence in the presence of uncertainty. A supervised learning algorithm takes a known set of input data and known responses to the data (output) and trains a model to generate reasonable predictions for the response to new data. Supervised learning uses classification and regression techniques to develop predictive models.

Classification techniques predict discrete responses—for example, whether an email is genuine or spam, or whether a tumor is cancerous or benign. Classification models classify input data into categories. Typical applications include medical imaging, speech recognition, and credit scoring.

Use classification if your data can be tagged, categorized, or separated into specific groups or classes. For example, applications for handwriting recognition use classification to recognize letters and numbers. In image processing and computer vision, unsupervised pattern recognition techniques are used for object detection and image segmentation.

Common algorithms for performing classification include support vector machine (SVM), boosted and bagged decision trees, *k*-nearest neighbor, Naïve Bayes, discriminant analysis, logistic regression, Random Forest Classification and neural networks.

Regression techniques predict continuous responses—for example, changes in temperature or fluctuations in power demand. Typical applications include electricity load forecasting and algorithmic trading.

Regression techniques if you are working with a data range or if the nature of your response is a real number, such as temperature or the time until failure for a piece of equipment. Common regression algorithms include linear model, nonlinear model, regularization, stepwise regression, boosted and bagged decision trees, neural networks, and adaptive neuro-fuzzy learning.

3.2.1.2 Unsupervised Learning

Unsupervised learning finds hidden patterns or intrinsic structures in data. It is used to draw inferences from data sets consisting of input data without labeled responses.

Clustering is the most common unsupervised learning technique. It is used for exploratory data analysis to find hidden patterns or groupings in data. Applications for cluster analysis include gene sequence analysis, market research, and object recognition.

For example, if a cell phone company wants optimize the locations where they build cell phone towers, they can use machine learning to estimate the number of clusters of people relying on their towers. A phone can only talk to one tower at a time, so the team uses clustering algorithms to design the best placement of cell towers to optimize signal reception for groups, or clusters, of their customers.

Common algorithms for performing clustering include k-means and k-medoids, hierarchical clustering, Gaussian mixture models, hidden Markov models, self-organizing maps, fuzzy c-means clustering, and subtractive clustering.

3.2.2 Random Forest Classification

Random forests or random decision forests are an ensemble learning method or supervised classification algorithm for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees [24]. As the name suggest, this algorithm creates the forest with a number of trees. In general, the more trees in the forest the more robust the forest looks like. In the same way in the random forest classifier, the higher the number of trees in the forest gives the high accuracy results. Random decision forests correct for decision trees' habit of over fitting to their training set. Ensembled algorithms are those which combines more than one algorithms of same or different kind for classifying objects [25]. For example, running prediction over Naive Bayes, SVM and Decision Tree and then taking vote for final consideration of class for test object.

Random forest classifier creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object.

Suppose training set is given as: [X1, X2, X3, X4] with corresponding labels as [L1, L2, L3, L4], random forest may create three decision trees taking input of subset for example,

1. [X1, X2, X3]
2. [X1, X2, X4]
3. [X2, X3, X4]

So finally, it predicts based on the majority of votes from each of the decision trees made. This works well because a single decision tree may be prone to a noise, but aggregate of many decision trees reduce the effect of noise giving more accurate results. Also, the random forest can apply weight concept for considering the impact of result from any decision tree. Tree with high error rate are given low weight value and vice versa. This would increase the decision impact of trees with low error rate.

There are two stages in Random Forest algorithm, one is random forest creation, and the other is to make a prediction from the random forest classifier created in the first stage. The whole process is explained below.

Random Forest creation pseudocode:

1. Randomly select “K” features from total “m” features where k << m
2. Among the “K” features, calculate the node “d” using the best split point
3. Split the node into daughter nodes using the best split
4. Repeat the a to c steps until “l” number of nodes has been reached

5. Build forest by repeating steps a to d for “n” number times to create “n” number of trees

In the next stage, with the random forest classifier created, we made the prediction. Random forest prediction pseudocode:

1. Takes the test features and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome (target)
2. Calculate the votes for each predicted target
3. Consider the high voted predicted target as the final prediction from the random forest algorithm

CHAPTER 4: METHODOLOGY

4.1. Hardware assembling

Six flex sensors were attached on the thumb, index, middle, ring, pinky fingers and the palm of a glove in order to measure the bent of the fingers and the clench of the hand, then the MPU6050 which comprised of both the accelerometer and gyroscope was placed on the back of the hand in order to determine the position and movement of the hand on the space. Flex sensors, and MPU6050 were interfaced with Arduino Mega 2560.

All the flex sensors were powered with a common voltage source of 5V and the voltage divider principle has been used with 10k Ohm resistor such that the varying voltage generated by the flex sensors due to bending are provided as input to the analog pins of the arduino.

MPU6050 has also been powered with 5V from the common power source, and for the communication between the MPU6050 and the Arduino Mega 2560 I^2C protocol has been used in which the SDA and SCL pin of the MPU6050 has been connected to the pin number 20 and 21 communication pin of the Arduino Mega 2560.

Arduino Mega has been programmed in such a way that it sent 12 datas from all the sensor which are 6 flex sensor data, 3 accelerometer data and 3 gyroscope data taken at duration of 500ms to its serial port. So, in 1 second duration there were 24 data at the serial port. Arduino was connected to the serial port of the computer and by the use of python programming language the data at the serial port was collected.

4.2. Dataset Preparation

4.2.1. Dataset Collection

Due to unavailability of the datasets for training the machine with our sensors data the dataset for training the model was made where data at the serial port for each gestures representing the alphabets and some frequently used words were collected and saved in comma separated values (csv) file format.

4.2.2. Data Preprocessing

All the data collected with various features(sensor value) as per the gesture were categorized with their respective alphabet or word as their target value, and the final collected data were randomized or shuffled in order to reduce the variance and to make sure that the model remain general and over fit less.

4.2.3 Dataset description

A total of 15050 data (after all preprocessing steps) taken at different time, condition, setting. Different time includes various timing for data set collection and condition and setting include different position of hand on space for the same category as well as various level of bent of the fingers showing the same category.

4.3 System Block Diagram

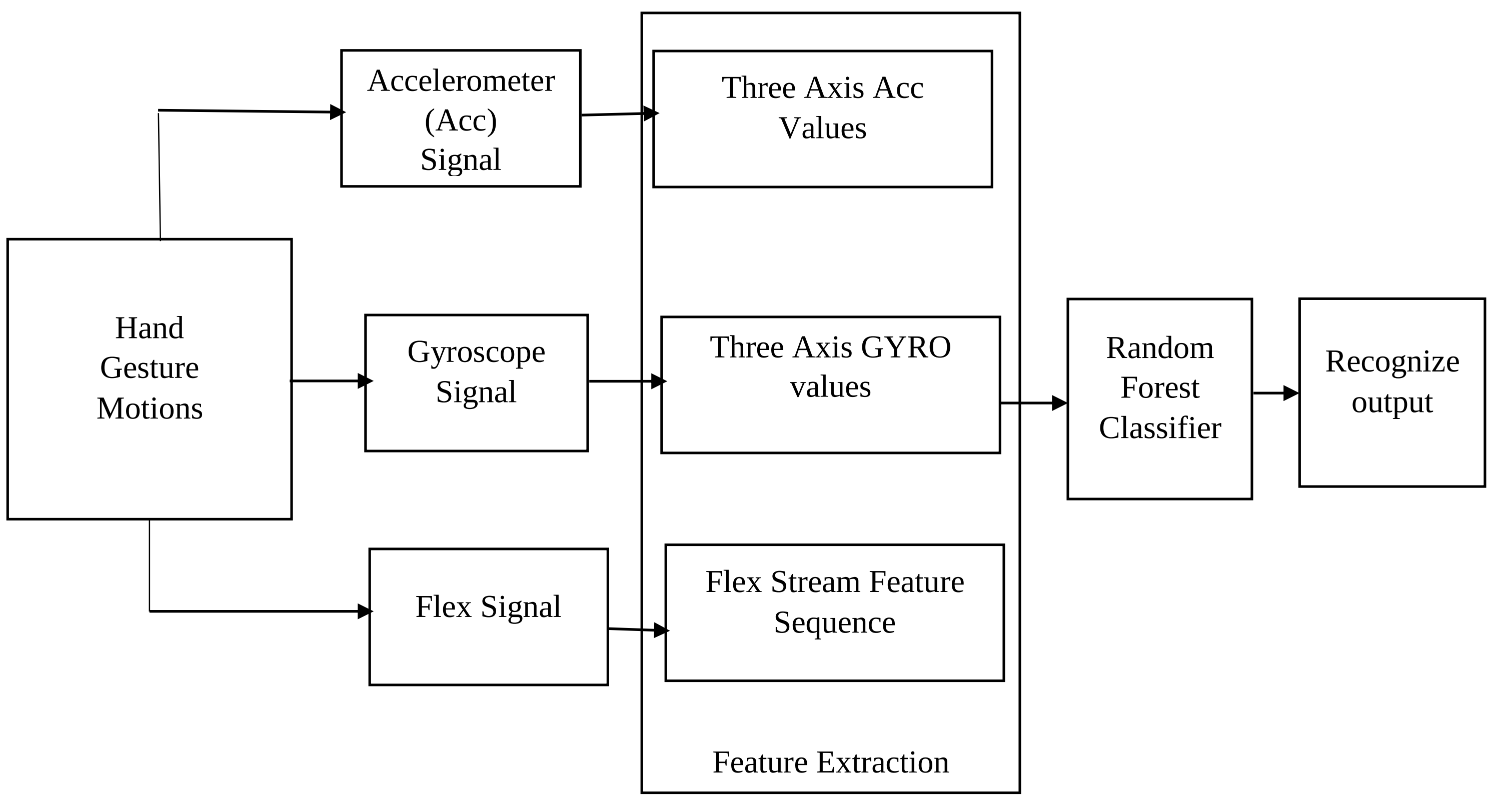


Figure 4.1: Feature Extraction Diagram

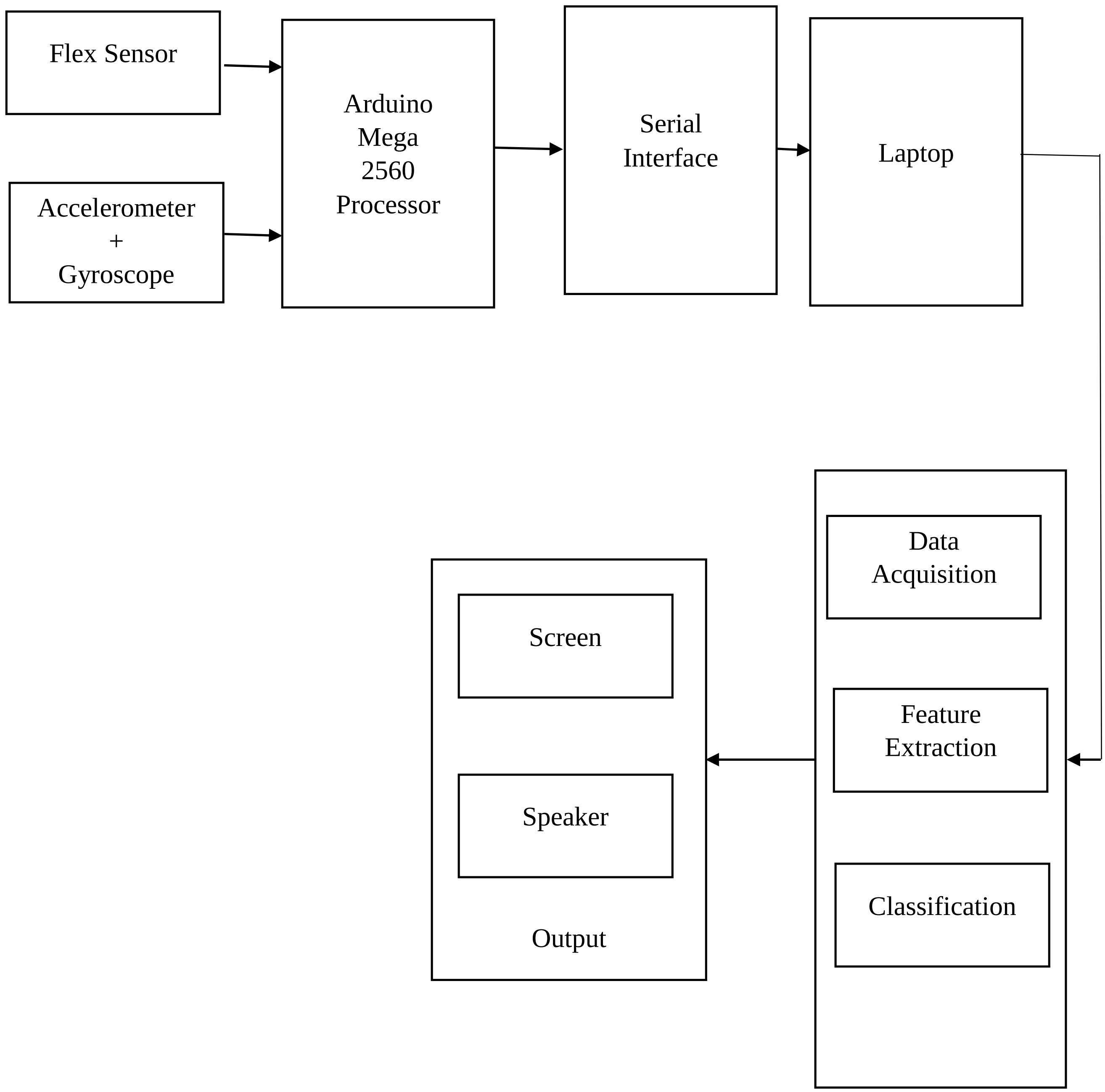


Figure 4.2: System Block Diagram

4.4 Explanation of Block Diagram

After preparing the trained dataset, the machine was trained with Random Forest Classifier. With the machine being trained with those dataset, then for the real time application of the trained machine, for the particular gesture the glove sensors data has been processed by the controller converting those raw data to meaningful data. Those data were sent from the controller to the serial port and was collected by the python through the serial interface and was saved as temporary dataset (CSV format) form, and finally the temporary dataset was passed through the machine such that the machine extracts the features of those dataset and predicts the appropriate output for that gesture with reference to the Random Forest Classifier model.

As the output has been recognized as per the model, laptop screen and speaker has been used. Output recognized was printed on the python output screen as well as the vocal of the particular gestures being recognized was played.

4.5 Algorithm

4.5.1 Algorithm for dataset preparation at Arduino Mega 2560

Step 1: Start

Step 2: Input: For a particular gesture

24 sensor data at 1 sec duration.

12 sensor data at 500 ms duration.

Step 3: Convert raw sensor data to meaningful data

Step 4: Set the baud rate

Step 5: Send data to the serial port

Step 6: Is arduino buffer flushed?

If yes, go to step 7

else , Wait for buffer to flush then continue

Step 7: Add delay of 1 second.

Step 8: Go to step 2.

4.5.2 Algorithm for data set preparation at python end

Step 1: Start

Step 2: Set the baud rate similar to arduino serial baud rate.

Set the count =0

Step 3: Is there data at serial port?

If no, wait till data available then continue.

Else go to step 4.

Step 4: Input: Collect data from the serial port.

Step 5: Open new file in write mode.

Step 6: Convert collected data to CSV format and write to the file.

Step 7: Flush the arduino buffer.

Step 8: count=count+1

Add delay of 1 second.

Step 9: Is count <=350?

If yes, go to Step 10.

Else go to Step 3.

Step 10: Close the file

Step 11: End

4.5.3 Algorithm for real time application

Step 1: Start

Step 2: Train the model using the training dataset.

Step 3: Set the baud rate similar to arduino serial baud rate.

Step 4: Is there data at serial port?

If no, wait till data available then continue.

Else go to step 5.

Step 5: Input: Collect data from the serial port at real time.

Step 6: Open new file in write mode.

Step 7: Convert the collected data to csv format and write to the file.

Step 8: Flush the arduino buffer.

Step 9: Pass the temporary dataset through the model.

Step 10: Trained machine predicts the output.

Step 11: Display the prediction on the screen and play the audio of the predicted word.

Step 12: Add 1 second delay

Step 13: Go to step 4.

4.6 Flowchart

4.6.1 Flowchart for Dataset preparation using Arduino Mega 2560

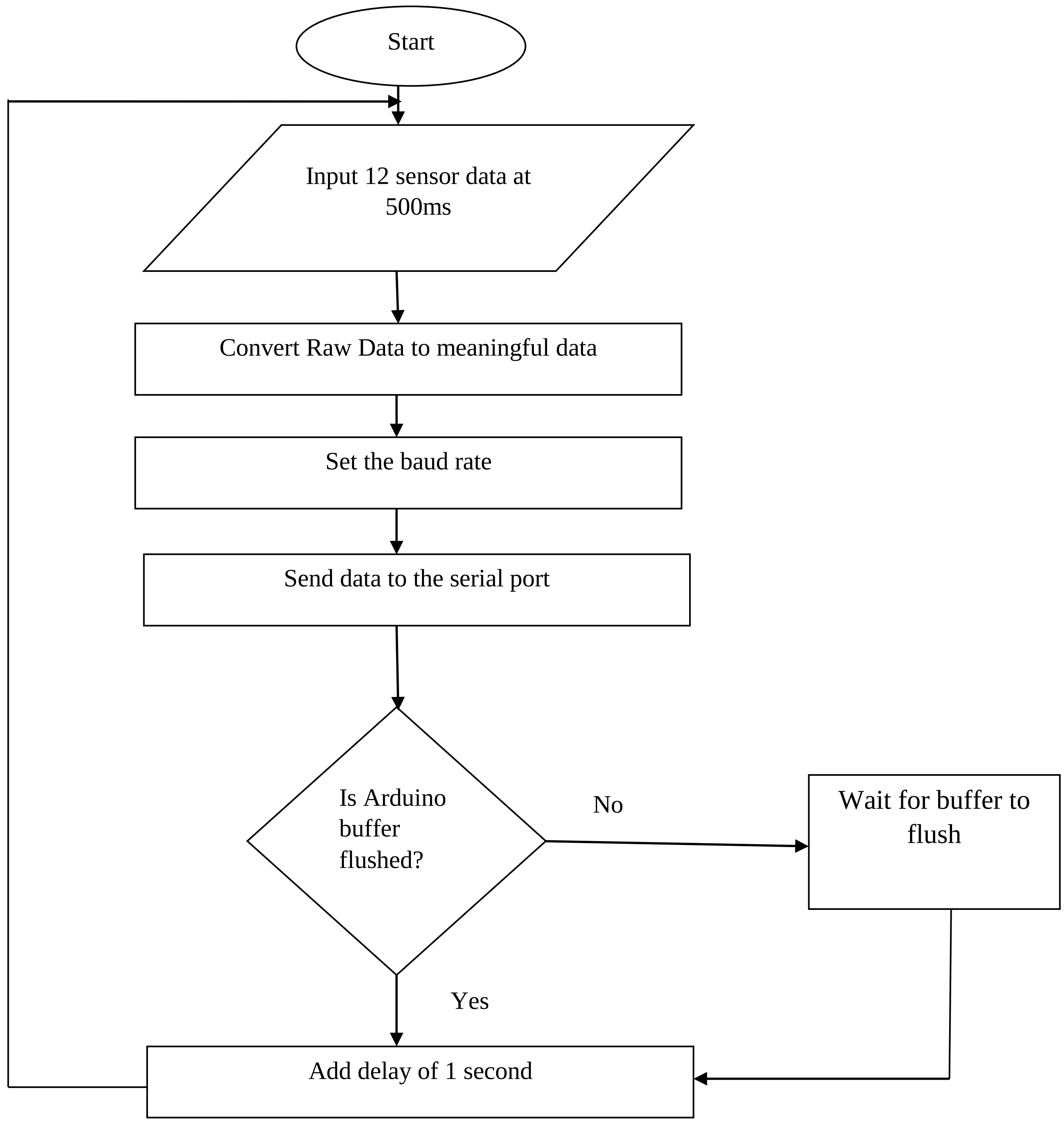


Fig 4.3: Flowchart for dataset preparation using Arduino Mega 2560

4.6.2 Flowchart for data set preparation using python

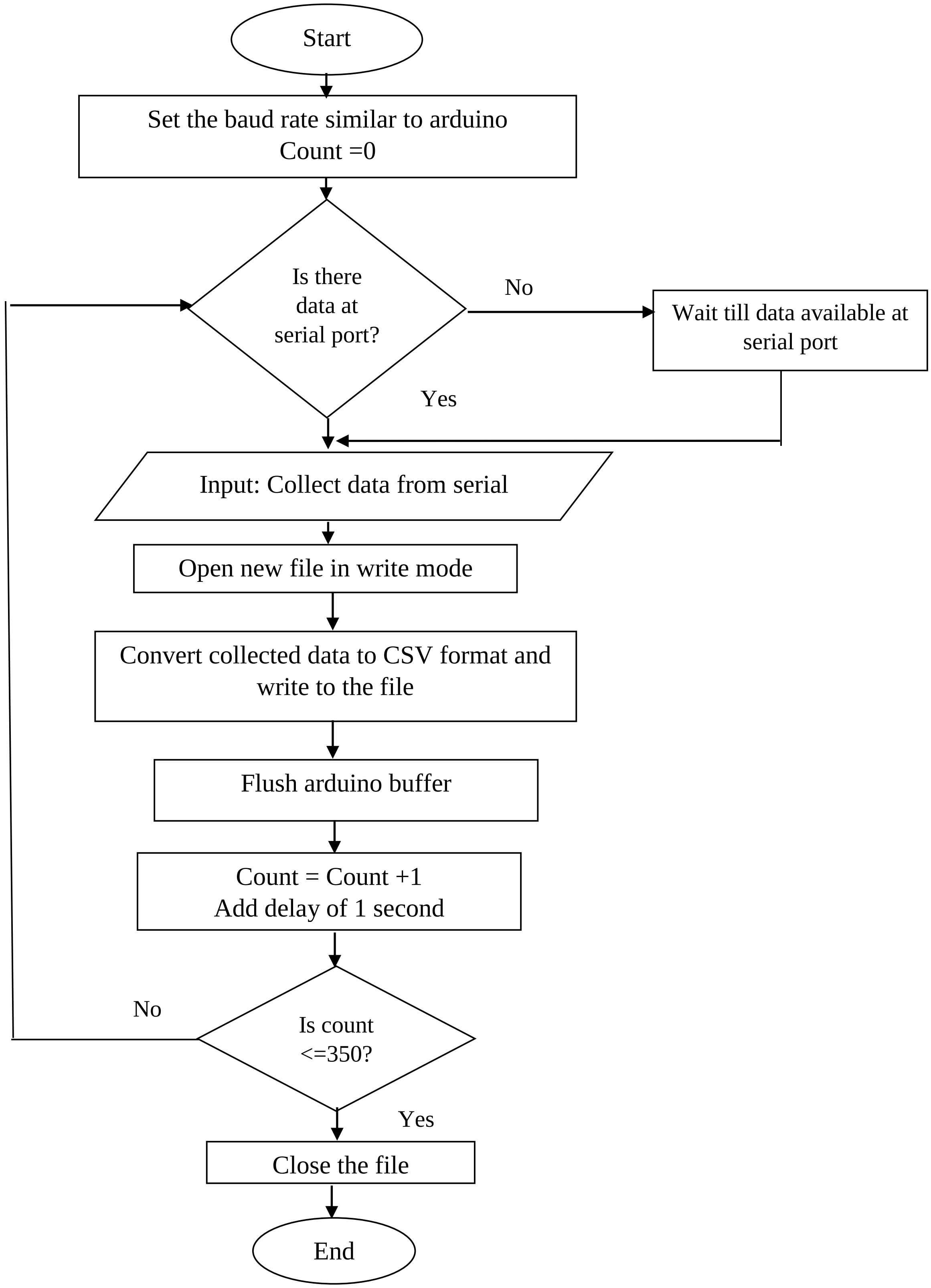


Figure 4.4: Flowchart of data set preparation using python

4.6.3 Flowchart of real time application

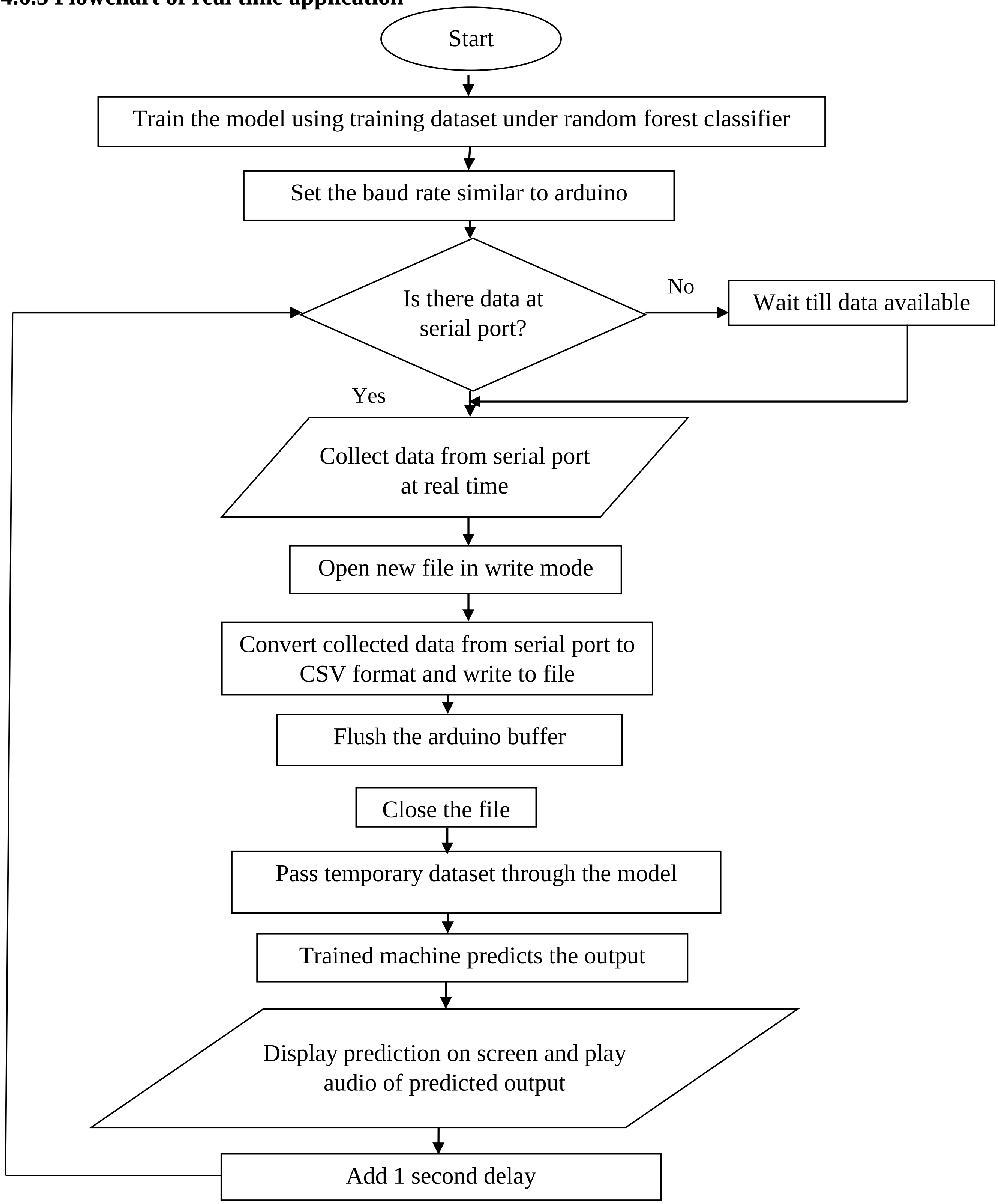


Figure 4.5: Flowchart of real time application

CHAPTER 5: RESULT AND ANALYSIS

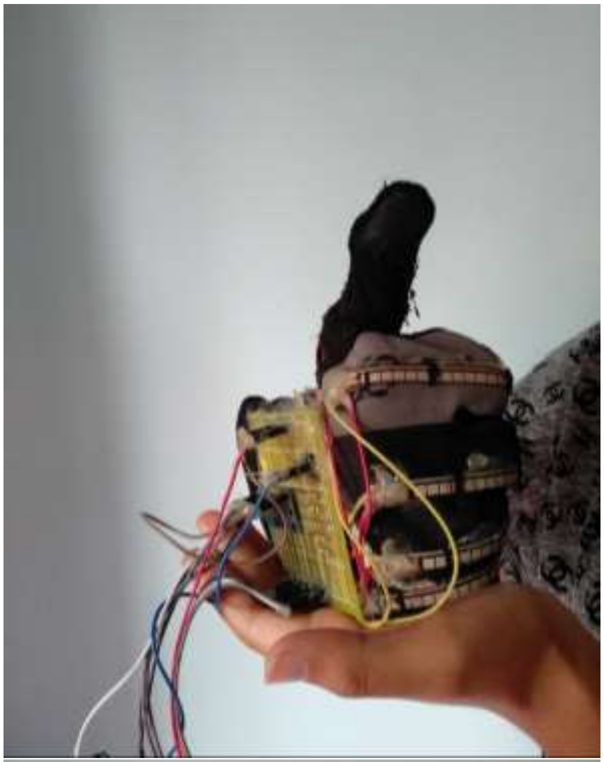
5.1 Result

The dataset for all the alphabets and frequently used words were created. Using those datasets the machine was trained and the model was created. Then dataset for all the alphabets were combined and shuffled in order to train the machine with reduced variance and make sure that model remains general and over fit less. When the input was taken for sign language of the alphabet or word, the input was passed through the trained machine and the closest value predicted was displayed and the audio was played as output. Then the correlation plot of each alphabet was compared with other alphabet to check the level of closeness of alphabets with each other and predict the accuracy of the model. The accuracy for our model was found to be 60% for alphabets and 80% for words.

5.2 Analysis

Around 350 dataset for each alphabet and word was created, and for each reading 24 values from sensor were taken at the interval of 1 seconds. Those 24 values were divided into two parts, each of 12 values at 500 millisecond interval. The 12 values included reading of thumb, index finger, middle finger, ring finger, pinky finger, palm, 3 values from gyroscope and 3 values from accelerometer at x, y and z axis. Then the correlation plot of those 12 data were visualized which described the association between the data and indicated the presence of causal relationship between those data statistically.

From the correlation plot we analyzed that for alphabets having placement of fingers similar to each other, for example for alphabet ‘a’,’s’ and ‘e’ the placement of fingers are same except for the thumb. For these alphabet the machine gets confused and the prediction is not quite accurate as the alphabet is distinguished only on the basis of thumb.

	Help me
	Up

	A
	Yes

	Okay
	I need to go to washroom

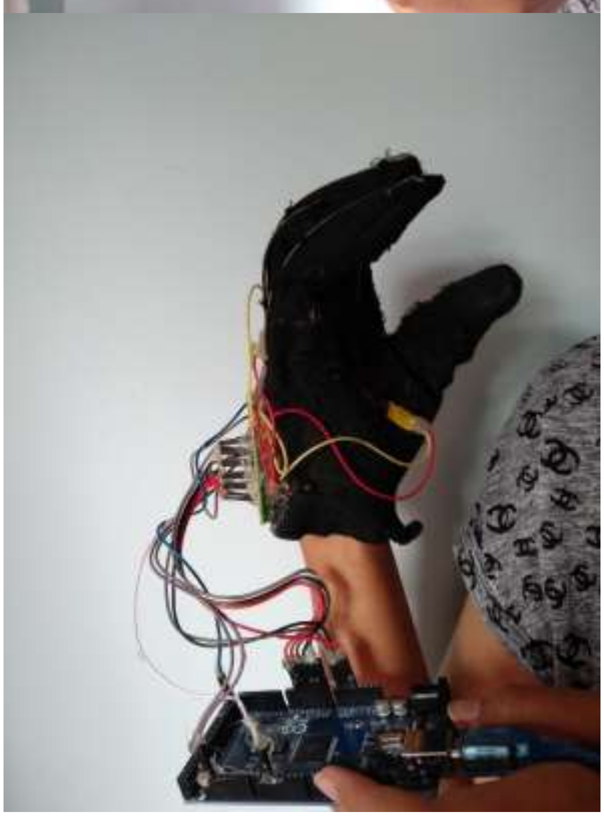
	Stop
	Be Careful

Figure 5.1: Demonstration

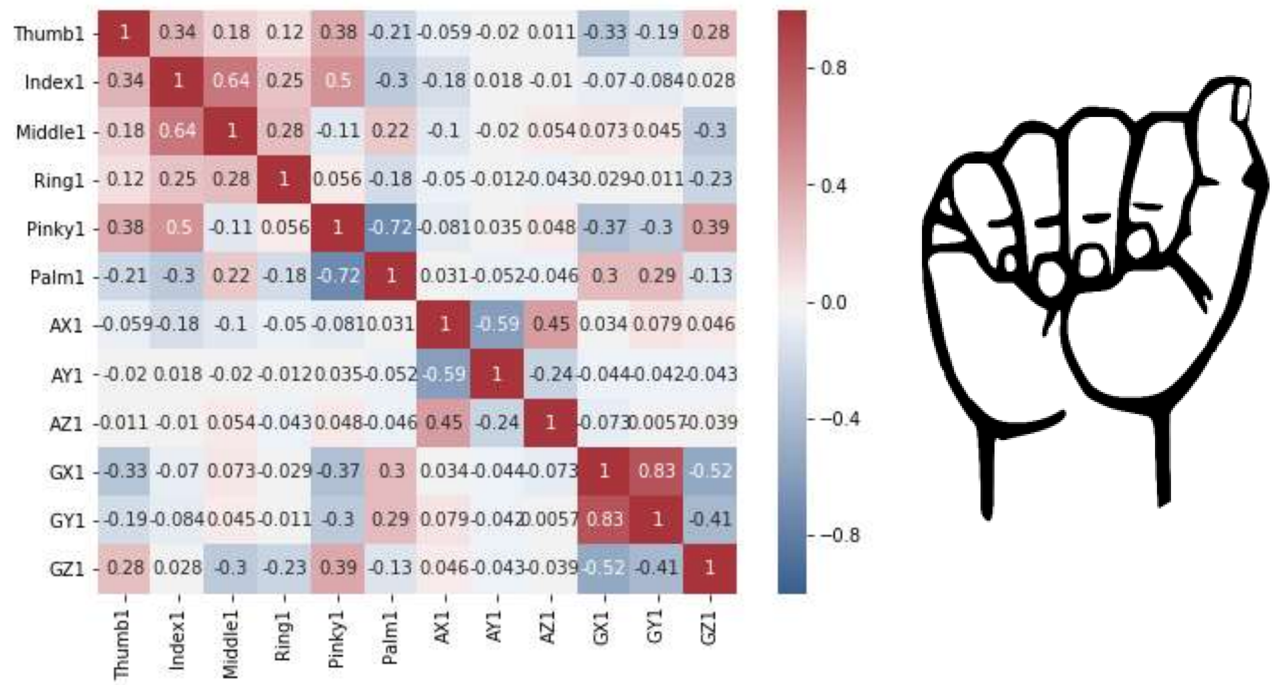


Figure 5.2: Correlation plot for alphabet ‘a’

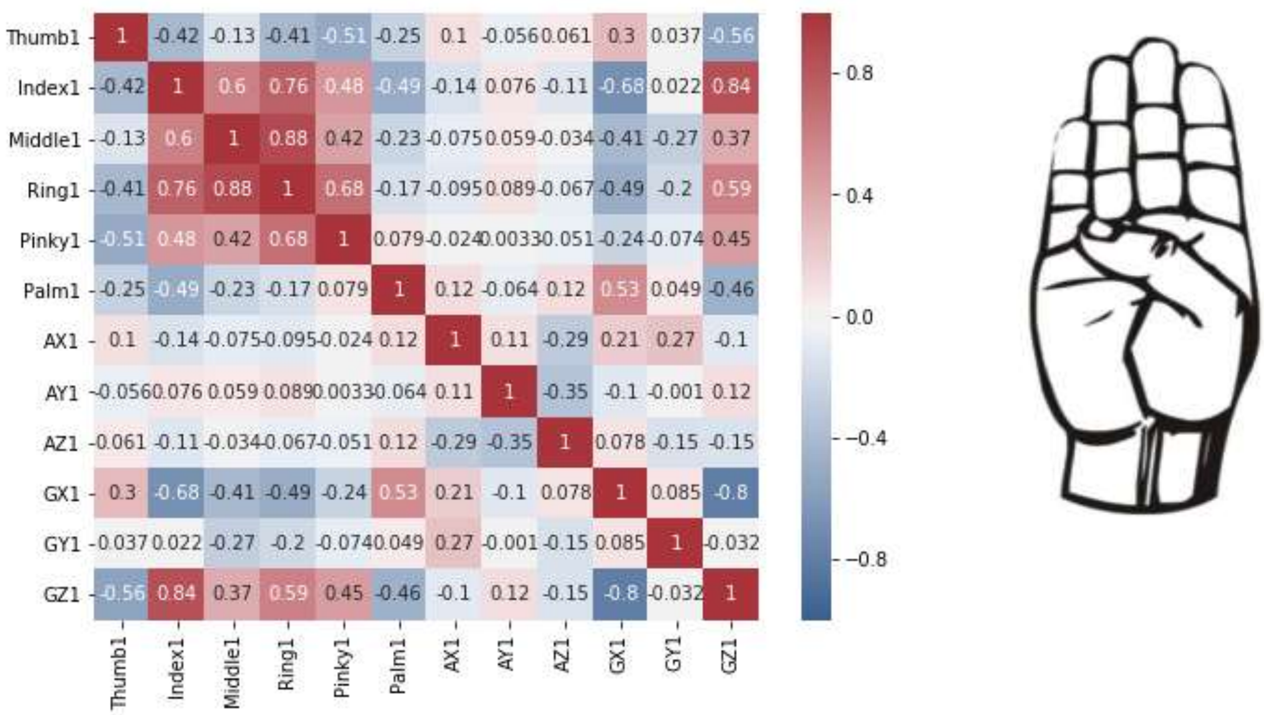


Figure 5.3: Correlation plot for alphabet ‘b’

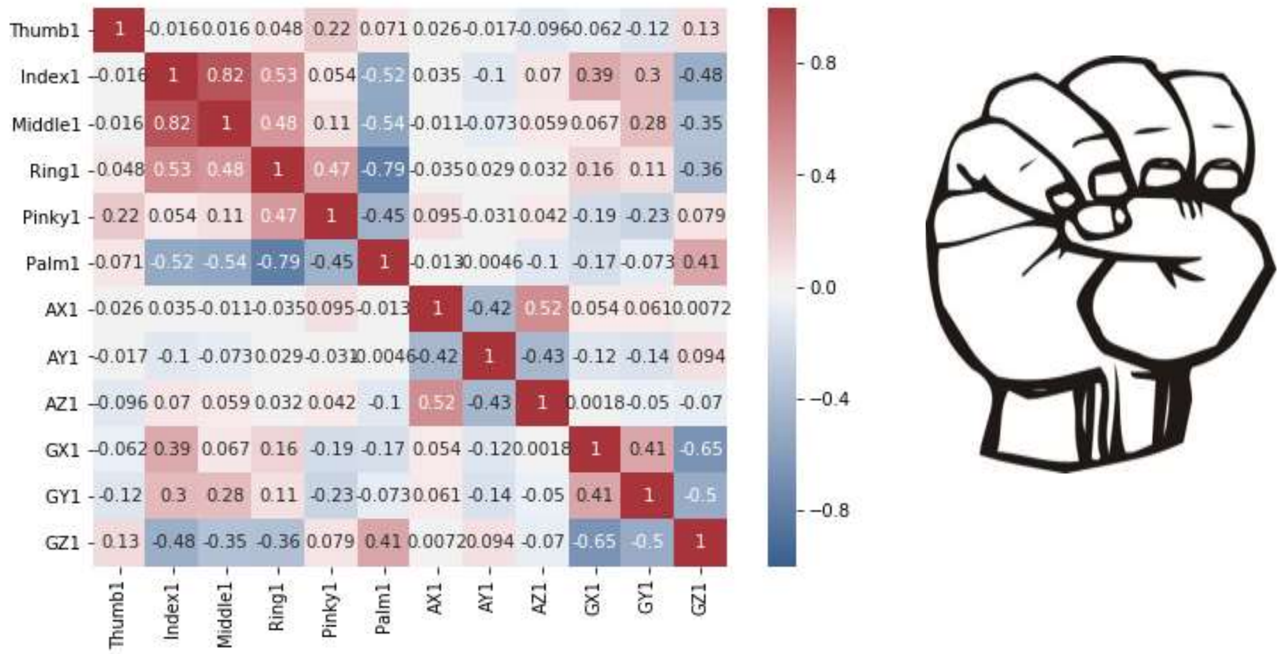


Figure 5.4: Correlation plot for alphabet ‘e’

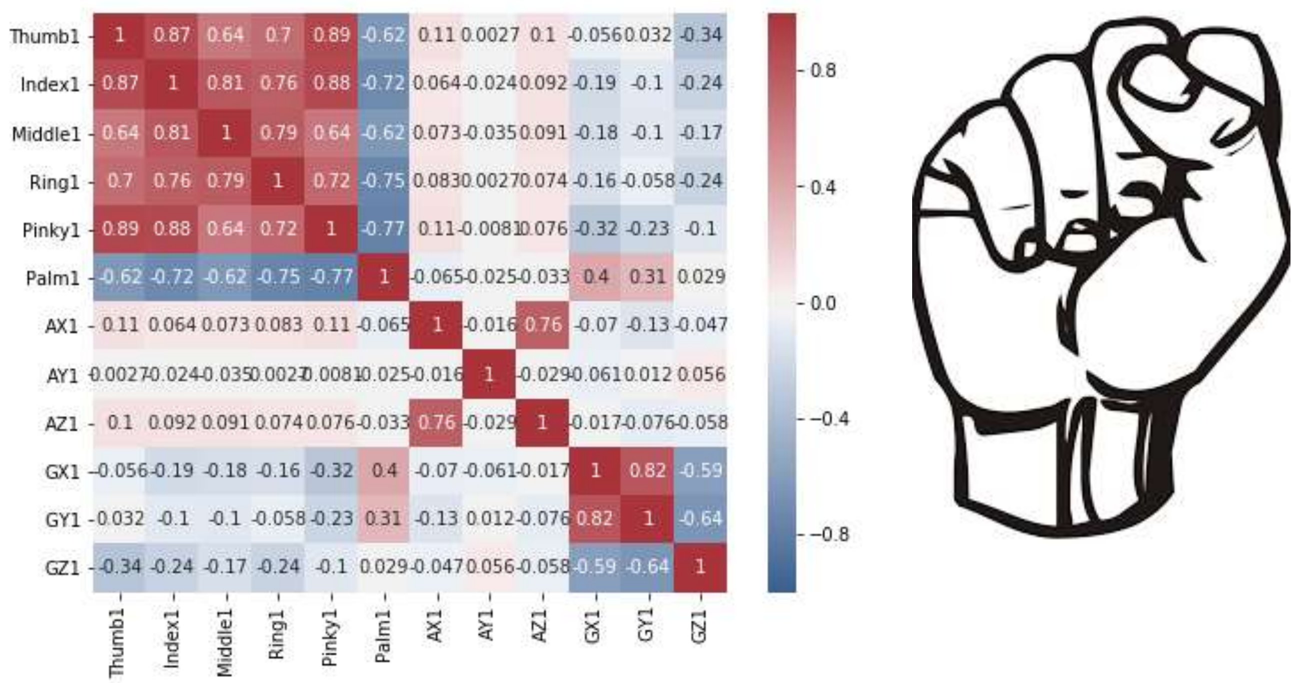


Figure 5.5: Correlation plot for alphabet ‘s’

CHAPTER 6: EPILOGUE

6.1 Conclusion

This report has discussed the project of smart gloves for deaf and dumb communities. The primary objective of this project was to make a device that could read the sign language to help deaf and dumb people communicate more efficiently with normal population. Machine learning was used to train the datasets for the sign language. Deaf people rely on sign language interpreters for communication. However, they cannot depend on interpreters in everyday life mainly due to high costs and difficulty in finding and scheduling qualified interpreters. This system will help them in improving their quality of life significantly.

A huge drawback of this model is the closeness of dataset of different alphabet and word with each other. Due to time limitation we only were able to take 350 dataset for each alphabet and word. As the number of dataset increases the accuracy also increases accordingly. Regardless of the issues mentioned, the designed glove could assist bridge communication gap between deaf dumb people and normal population to a certain level.

6.2 Future enhancement

Nothing can be ended in a single step. It is the fact that nothing is permanent in this world. So this project also has future enhancements in reading sign language and gestures with the highest efficiency possible. The system is a compatible one, so, addition of new modules can be done without much difficulty. Some of the enhancements that can be made to this project are listed as follows:

- Natural language processing can be implemented
- Mobile application can be used to make communication easy and interactive
- For military purpose to decode morse code
- Make wearable technology to bridge the communication gap with accurate and precise result

References

[1] D. Hubbell, "Blogs: Microsoft," 9 January 2014. [Online]. Available: https://blogs.msdn.microsoft.com/accessibility/2014/01/09/ukrainian-students-design-gloves-that-translate-sign-into-spoken-words/. [Accessed 11 December 2017].

[2] Laura Dipietro, Angelo M.Seior, member IEEE, "A Survey of Glove- Based System and their Applications," *IEEE transaction on System, Man and Cybernetics- Part C: Application and Review*, vol. 38, no. 4, 2008.

[3] Sidney Fels and Geoffrey Hinton, "Glove Talk II," Department of Computer Science of University of Toronto, Toronto, 6 November 1992.

[4] E.Hintom, S.SidenyFels and Geoffrey, "Glove Talk II- A neural network Interface which map gesture to parallel format speech synthesis controls," *IEEE Transaction on neural networks*, vol. 9, no. 1, 1998.

[5] Harmeet Kaur, Amit Saxena, Abhishek Tandon, Keshav Mehrotra and Khushboo Kashyap, *A Review Paper on Evolution of Smart Glove‖, International Journal of Scientific Research and 9Management Studies (IJSRMS)*, vol. 3, no. 3, pp. 124-128, 2016.

[6] "Speakjet," Magnevation Speak Jet Inc. USA, 4 June 2010. [Online]. Available: http://www.speakjet.com/. [Accessed 5 January 2017].

[7] M. Deller, A. Ebert, M. Bender and H. Hagen, "Flexible gesture recognition for immersive virtual evironments," in *Tenth International Conference on Information Visualization (IV 2006)*, IEEE, July 2006.

[8] Tushar Chouhan, Ankit Panse, Anvesh Kumar Voona and S. M. Sameer, "Smart Glove With Gesture Recognition Ability For The Hearing And Speech Impaired," in

IEEE Global Humanitarian Technology Conference- South Asia Satellite (GHTC-SAS) , September 26-27, 2014.

[9] Abhishek Tandon, Amit Saxena Keshav Mehrotra, Khushboo Kashyap, Harmeet Kaur, "A Review Paper on Smart Gloves- Converts Indian Sign Language(ISL) into Text and Speech," *International Journal for Scientific Research & Development (IJSRD)* , vol. 4, no. 8, pp. 269-272, 2016.

[10] M. Mohandes, S. I. Quadri, and M. D. King, "Arabic Sign Language Recognition an Image Based Approach," in *21st International Conference on Advanced Information Networking and Applications Workshops, 2007 (AINAW'07)* , 2007.

[11] M. P. Paulraj, S. Yaacob, H. Desa, and W. Majid, "Gesture Recognition System got KodTangan Bahasa Melayu (KTBM) using Neural Network," in *5th International Colloquium on Signal Processing and Its Applications (CSPA)*, 2009.

[12] S. J. Wang, D. C. Zhang, C. C. Jia, N. Zhang, C. G. Zhou, and L. B. Zhang, "A Sign Language Recognition Based on Tensor," in *Second International Conference on Multimedia and Information Technology (MMIT)*, Kaifeng, China, 2010.

[13] G. Fang, W. Gao, and D. Zhao, "Large vocabulary sign language recognition based on fuzzy decision trees," *IEEE Trans. Syst., Man, Cybern A, Syst, Humans,* vol. 34, no. 3, pp. 305-314, 2004.

[14] D. M. Sherrill, P. Bonato, and C. J. De Luca, "A neural network approach to monitor motor activities," in *Proc. 2nd Joint EMBS/ BMES Conference*, Houston, Texas, 2002.

[15] V. E. Kosmidou and L. J. Hadjileontiadis, "Sign language recognition using intrinsic mode sample entropy on sEMG and accelerometer data," *IEEE Trans. Biomed. Eng.,* vol. 56, no. 12, pp. 2876-2890, 2009.

[16] creative commons, "MegaQuickRef: arduino-info," wikispaces, [Online]. Available: https://arduino-info.wikispaces.com/MegaQuickRef. [Accessed 7 August 2018].

[17] Solanki Kumar, "Indian SIgn Language using Flex sensor Glove," *International Journal of Engineering Trends and Technology(IJETT)*, vol. 4, no. 6, 2013.

[18] creative commons, "Product: SparkFun Electronics," SparkFun Electronics, [Online]. Available: https://www.sparkfun.com/products/10264. [Accessed 7 August 2018].

[19] "electronicwings," ElectronicWings, [Online]. Available: http://www.electronicwings.com/sensors-modules/mpu6050-gyroscope-accelerometer-temperature-sensor-module. [Accessed 4 July 2018].

[20] Sandor Kabai, "Gyroscope," *Wolfram Demonstrations Project*, 30 April 2008.

[21] Samuel, Arthur, "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210-229, 1959.

[22] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[23] B. Dickson, "Exploiting machine learning in cybersecurity," 2 July 2016. [Online]. Available: https://techcrunch.com/2016/07/01/exploiting-machine-learning-in-cybersecurity/. [Accessed 20 June 2018].

[24] Ho, Tin Kam, "Random Decision Forests," in *3rd International Conference on Document Analysis and Recognition*, Montreal, QC, 1995.

[25] T.K. Ho , "The Random Subspace Method for Constructing Decision Forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, p. 832–844, 1998.