**TRIBHUWAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

Kathmandu Engineering College

Kalimati, Kathmandu

**Major Project Report on:**

**Sign Language Recognition System using Flex Sensors**



Submitted By:

Aarati Shrestha (76002)

Ira Shrestha (76010)

Reet Aryal (76018)

Roshika Gurung (76021)

To:

DEPARTMENT OF ELECTRONICS, COMMUNICATION

&

INFORMATION ENGINEERING

Kathmandu, Nepal

FALGUN, 2080

**TRIBHUVAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

Kathmandu Engineering College

Department of Computer Engineering

CERTIFICATE

The undersigned certify that they have read and recommended to the Department of Electronics, Communication and Information Engineering a major project work entitled “Sign Language Recognition System Using Flex Sensor” submitted by Aarati Shrestha-76002, Ira Shrestha-76010, Reet Aryal-76018 and Roshika Gurung-76021 in partial fulfillment of the requirements for the degree of Bachelor of Engineering.

Prof. Ram Krishna Maharjan

(External Examiner)

Department of Electronics and Computer Engineering

Pulchowk Campus

Institute of Engineering (IOE)

Asst.Prof. Suramya Sharma Dahal

(Project Supervisor)

Department of Electronics, Communication and Information Engineering

(Kathmandu Engineering College)

Asst.Prof. Rajan Lama

(Head of Department)

Department of Electronics, Communication and Information Engineering

(Kathmandu Engineering College)

Asst.Prof. Rajan Lama

(Project Supervisor)

Department of Electronics, Communication and Information Engineering

(Kathmandu Engineering College)

# ACKNOWLEDGEMENT

First and foremost, we offer our sincerest gratitude towards the **Department of Electronics Communication, and Information Engineering, Kathmandu Engineering College, Kalimati** and Institute of Engineering (IOE) for providing us the wonderful opportunity to undertake this project. We would like to thank our project supervisor **Er. Suramya Sharma Dahal** and **Er.** **Rajan Lama** for giving us the necessary guidance to approach this project and review of this report. We would also like to sincerely thank **Er. Dipen Manandhar** sir, Project Coordinator and Supervisorfor his generous and continuous support for helping us in the different aspects of the way of performing this project. We would like to extend our sincere gratitude to **Er. Rajan Lama** Sir**,** Head of the Department of Electronics and Communications Engineering for providing us with this opportunity to perform our Major project.

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

# ABSTRACT

As a human being, we 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. Such people depend on other means of communication like sign language. According to World Health Organization (WHO), about 5%, approximately 70 million, of the world’s population are mute and deaf and only a faction from this number know how to communicate well using correct sign language. Sign language is a method of non-verbal communication that is used by deaf and mute people. 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 which is implemented in the gloves with sensors. The glove is embedded with sensors to read the sign language and send data as text as well as speech. This project calibrates the information received from the sensor and displays the appropriate output reducing the communication gap between normal people and differently abled people. Depending on different hand signs the resistance value throughout the flex sensor changes and certain messages are shown. We have also compared various machine learning algorithms and compared it. Neural Network showed the better result so we used it.

**Keywords**: *Arduino, Arduino Mega, Flex Sensor, MS-Excel, Python, Panda, Training, Testing, Machine Learning, Neural Network*

**TABLE OF CONTENTS**

[ACKNOWLEDGEMENT i](#_Toc160393113)

[ABSTRACT ii](#_Toc160393114)

[LIST OF FIGURES vi](#_Toc160393115)

[LIST OF ABBREVIATION vii](#_Toc160393116)

[LIST OF TABLES viii](#_Toc160393117)

[CHAPTER 1: INTRODUCTION 1](#_Toc160393118)

[1.1 Background theory 1](#_Toc160393119)

[1.1.1 Nepali Sign Language 2](#_Toc160393120)

[1.2 Problem Statement 3](#_Toc160393121)

[1.3 Objectives 4](#_Toc160393122)

[1.4 Scope or Application 4](#_Toc160393123)

[1.5 Organization of Report 4](#_Toc160393124)

[CHAPTER 2: LITERATURE REVIEW 5](#_Toc160393125)

[CHAPTER 3: RELATED THEORY 9](#_Toc160393126)

[3.1 Machine Learning 9](#_Toc160393127)

[3.2 Neural Network 10](#_Toc160393128)

[3.3 Android Studio 12](#_Toc160393129)

[3.4 Arduino Mega 12](#_Toc160393130)

[3.5 Flex sensor 14](#_Toc160393131)

[3.6 MPU-6050 Module 15](#_Toc160393132)

[3.6.1 Accelerometer 16](#_Toc160393133)

[3.6.2 Gyroscope 17](#_Toc160393134)

[3.7 Bluetooth module 17](#_Toc160393135)

[3.8 Tools Used 18](#_Toc160393136)

[3.8.1 Python 18](#_Toc160393137)

[3.8.2 Scikit-learn 18](#_Toc160393138)

[3.8.3 Pandas 18](#_Toc160393139)

[3.8.4 Jupyter Notebook 18](#_Toc160393140)

[3.9 Verification and Validation 18](#_Toc160393141)

[3.9.1 Accuracy 18](#_Toc160393142)

[3.9.2 Precision 18](#_Toc160393143)

[3.9.3 Recall 19](#_Toc160393144)

[3.9.4 F1 Score 19](#_Toc160393145)

[3.9.5 Confusion Matrix 19](#_Toc160393146)

[CHAPTER 4: METHODOLOGY 20](#_Toc160393147)

[4.1 Hardware Assembling 20](#_Toc160393148)

[4.2 Dataset Preparation 21](#_Toc160393149)

[4.2.1 Dataset Collection 21](#_Toc160393150)

[4.2.2 Data Preprocessing 21](#_Toc160393151)

[4.2.3 Dataset Description 21](#_Toc160393152)

[4.3 Block Diagram 22](#_Toc160393153)

[4.4 Algorithm 24](#_Toc160393154)

[4.4.1 Algorithm for training and validation of model 24](#_Toc160393155)

[4.4.2 Algorithm for real time application 24](#_Toc160393156)

[4.5 Flowchart 25](#_Toc160393157)

[4.5.1 Flowchart for Dataset Preparation 25](#_Toc160393158)

[4.5.2 Flowchart for real time application 26](#_Toc160393159)

[CHAPTER 5: RESULT AND ANALYSIS 27](#_Toc160393160)

[5.1 Completed Results 27](#_Toc160393161)

[5.1.1 Research Works 27](#_Toc160393162)

[5.1.2 Data Collection 27](#_Toc160393163)

[5.1.3 Writing data into Excel file and Labeling 28](#_Toc160393164)

[5.1.4 Model Performance 29](#_Toc160393165)

[5.1.5 Comparison among models 30](#_Toc160393166)

[5.1.6 Translation 31](#_Toc160393167)

[5.1.7 Display 31](#_Toc160393168)

[5.1.8 Commands and Outputs 31](#_Toc160393169)

[5.1.9 Display in Application 33](#_Toc160393170)

[5.1.10 Problems Encountered 35](#_Toc160393171)

[5.1.11 Future enhancement 36](#_Toc160393172)

[REFERENCES 37](#_Toc160393173)

# LIST OF FIGURES

Figure 1: Hand Gesture using Flex Sensor 3

Figure 2:Simple Neural Network 11

Figure 3: Flex Sensor 14

Figure 4: MPU-6050 Module 15

Figure 5: Bluetooth Module 17

Figure 6: Hardware Assembling 20

Figure 7: Block Diagram 22

Figure 8: Flowchart for dataset preparation 25

Figure 9: Flowchart for real time application 26

Figure 10: Confusion Matrix for Testing Data 29

Figure 11: Learning Curve 29

Figure 12: Command 1 31

Figure 13: Command 2 32

Figure 14: Command 3 32

Figure 15: Command 4 32

Figure 16: Command 5 33

Figure 17: Command 6 33

Figure 18: Display in application in the form of text 34

# LIST OF ABBREVIATION

AC : Alternate Current

ADC : Analog to Digital Conversion

AVT : Artificial Vocal Track

CSV : Comma Separated Values

DC : Direct Current

DOF : Depth of Field

GND : Ground

MEMS : Micro electromechanical System

PCA : Principal Component Analysis

DTMF : Dual Tone Multi Frequency

MSA : Mathematical Sound Architecture

PIC : Peripheral Interface Controller

TP : True Positive

TN : True Negative

FP : False Positive

FN : False Negative

IDE : Integrated Development Environment

HCI : Human Computer Interface

ANN : Artificial Neural Network

SNN : Sequential Neural Network

# **LIST OF TABLES**

[Table 1 : Tech Specification 13](#_Toc1963)

[Table 2 : Dataset 28](#_Toc15371)

[Table 3 : Comparison among Models 30](#_Toc28484)

[Table 4 : Performance Parameter 30](#_Toc24911)

# CHAPTER 1: INTRODUCTION

## Background theory

In a world where communication is essential for personal, social, and professional interactions, it is crucial to ensure that everyone has the ability to express themselves and be understood. Unfortunately, not everybody has the capability of speaking and hearing. For individuals who are differentiable or deaf, spoken language presents a significant barrier to effective communication. In 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-differentiable people to show gestures for his/her needs. Differentiable people have their own language to communicate with us; the only thing is that we need to understand their language.

Sign language is a vital means of communication that transcends auditory limitations. Sign language is not merely a translation of spoken language into gestures, it is a gesture representation that involves simultaneously combining hand shapes, orientation and movement of the hands, arms or body, and facial expressions to express fluently a speaker’s thoughts. It is a language that embodies a rich cultural heritage, fostering a sense of identity and belonging within the deaf community. 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 differentiable communities when they try to engage in interaction with others, especially in their education, social and professional environment. Therefore, it is necessary to have an advanced gesture recognition or sign language detection system to bridge this communication gap. And even more for Nepali people as most of the sign language is in the English standard form. Another issue with Nepali Sign Language is that it does not have a single gesture for a phrase. There are only gestures for words and letters because of this, people have to give gesture for each word even if they want to say a long sentence.

To overcome this barrier, we have come up with a project called “Sign Language Recognition System using Flex Sensor”. By developing a system that converts NSL gestures into spoken Nepali language, we aim to empower deaf individuals to communicate more effectively and participate fully in society. Nepali Sign Language (NSL) is a rich and expressive visual language that allows deaf individuals in Nepal to communicate their thoughts, feelings, and ideas. However, the limited understanding of NSL within the hearing community often leads to communication barriers and social isolation for deaf individuals.

The development of a flex sensor-based system offers a promising solution, enabling real-time conversion of NSL gestures into spoken Nepali language. As this project is a wearable technology, all the sensors are fitted on a glove. Flex sensor plays a major role in the project. Flex sensors are bendable devices that change resistance based on the degree of flex or bend applied to them. They are typically integrated into the gloves on the fingers, to capture the movements and gestures made while signing. When a person wearing the gloves with sensors performs sign language gestures, the flex sensors detect the bending and flexing of the fingers and hand. This data is then transmitted to a processing unit or microcontroller embedded in the glove. The processing unit analyzes the sensor data using algorithms and machine learning techniques to identify the specific sign language gestures being performed. Once the sign language gestures are recognized, the system can then convert them into text and speech output. This advanced project holds tremendous promise in transforming the way deaf individuals communicate and interact with society. By leveraging technology to break down barriers, it paves the way for a more inclusive and accessible world, where effective communication knows no boundaries.

### 1.1.1 Nepali Sign Language

The primary sign language used in Nepal by specially-abled people is called Nepali Sign Language (NSL). It is a somewhat standardized language that is loosely based on the variation spoken in Kathmandu and combines aspects of the dialects used in Pokhara and other places. The Kathmandu Association of the Deaf (KAD) and American Patricia Ross collaborated to create the first NSL dictionary. This connection suggests that ASL may have played a role in the process. The deaf community in the Kathmandu valley created NSL as a natural language, but it has also been affected by other sign languages and sign systems like Total Communication and Simultaneous Communication. Although it was more pronounced during the early stages of creation, this effect from the outside resulting from contact still exists today in a variety of forms and to varying degrees. We decided to translate into Nepali gestures because in Nepali sign language there are no single gestures for whole phrase. In order to indicate a phrase a person has to enact entire words. To solve this problem we have created seven gestures to represent seven different mostly used gestures.



Figure 1: Hand Gesture using Flex Sensor

## Problem Statement

The lack of a standardized translation system for converting hand gesture sign language into the Nepali language poses significant communication barriers for individuals who are deaf or hard of hearing in Nepal. This results in their limited ability to express themselves, understand others, and fully participate in various aspects of life. The absence of a recognized framework also hinders educators, interpreters, and service providers in effectively assisting individuals with hearing disabilities. To address this problem, there is a need to develop a comprehensive and widely recognized translation system that promotes inclusivity, empowers the deaf community, and ensures equal access to communication and opportunities in Nepal.

## Objectives

The objective of our project is

* Enable individuals who are deaf or hard of hearing to communicate effectively with others.
* To build a glove embedded with sensors to read the sign language and convert it into text and speech.
* Support inclusive education by providing a reliable translation system in educational settings.

## Scope or Application

The scope or application of the project is

* For all deaf and differential people.
* Enhanced employment opportunities and workplace inclusivity.

## 1.5 Organization of Report

The first chapter of our report deals with the introductory part of the project. It mentions about the background of the project that we have done. Also, the objectives of our project, the problem that we faced while doing the project are talked about along with the scope. Chapter 2 deals with the literature review that describes the past works that were undertaken related to this project and also the components that were used in the past. Chapter 3 discusses about the hardware and software that are used in the project in detail. Chapter 4 explains the system block diagram, algorithm and flowchart along with the information about dataset, how it is created and how it works. It deals with conceptual design and outline of the project. Chapter 5 shows the work that is completed and various stages of completion. It also informs about the problems that were faced during the development of the project and how we tackled it and also about the future aspects and enhancement of this project.

# CHAPTER 2: LITERATURE REVIEW

In 2008, a survey was conducted on smart gloves [[1]](#one). This survey suggests that there was a significant gap in research and development in the field of smart gloves during that time. The survey refers to the system as “Glove-based systems” which are described as consisting of multiple components. These components include of sensors, electronics for data acquisition and processing, a power supply, and a support structure for the sensors. They had limitations in the form of portability, as they required wired physical connections, limited haptic sensing and naturalness of movement.

There is an absence of communication with deaf people in our society so to overcome this barrier the introduction of sign language took place. To convey message or meaning, sign language makes use of various patterns that are visually transmitted sign patterns. This system uses a camera which captures various gestures of the hand. First, pre-processing of the image takes place and then, determination of edges occurs by using an edge detection algorithm. A template matching algorithm identifies the sign and display the text. This curtails the difficulty to communicate with the deaf [[2]](#two). Many projects used glove-based systems for automatic understanding of gestural languages used by deaf community. The systems developed in these projects differed in characteristics such as number of classifiable signs, which could range from few dozen to several thousands, types of signs, which could either be static or dynamic, and percentage of signs correctly classified. The simplest systems were limited to understanding of finger spelling or manual alphabets (a series of hand and finger static configurations that indicate letters) [[3]](#three). Murakami and Taguchi used a Data Glove for recognition of the Japanese alphabets [[4]](#four).

Mandeep Kaur et al. presented a scheme using a database driven hand gesture recognition based upon skin color model approach and thresholding approach along with an effective template matching with can be effectively used for human robots applications and similar other application [[5]](#five). Initially, hand region is segmented by applying skin color model in YCbCr color space. In the next stage thresholding is applied to separate foreground and background. In this way, the team developed template based matching technique using Principal Component Analysis (PCA) for recognition.

Sidney Fels and Geoffrey Hinton developed a system called Glove-Talk (II) which translates hand gestures to speech through adaptive interface [[6]](#six). Many different possible schemes exist for converting hand gestures to speech. The choice of scheme depends on the granularity of the speech. Glove-Talk is a system that implements an AVT ( i.e. Artificial Vocal Tract). The AVT allows unlimited vocabulary, multiple languages in addition to direct control of fundamental frequency and volume, control of pitch and non-verbal sounds. Neural networks were used to implement an adaptive interface, called Glove Talk II, which contains hand gestures to control the parameters of a parallel format speech synthesizer to allow a user to speak with his hands. Hand gestures are mapped continuously to 10 control parameters of a parallel format speech synthesizer. Due to which, it allows system to act as AVT. Currently, the best version of Glove-Talk II uses a several input devices such as a Cyber glove, Contact glove, a parallel formant speech synthesizer and 3 neutral networks. The gesture to speech task is divided into vowel and a constant production by using a gating network to weight the outputs of a vowel and a consonants neutral 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 doesn’t require any training examples from the user.

In 1977, Daniel J. Sandin and Thomas DeFanti created the first wired glove which is based on an idea by Richard Sayre as a project for the National Endowment for the Arts at the Electronics Visualization Laboratory [[7]](#seven). This system is also known as the Sayre Glove or Data Glove. It used light based sensors with flexible tubes with light sources at one end and a photocell at the other end. As the fingers were bent, the amount of light that hit the photocells varied, thus providing a measure of finger flexion. It was lightweight, inexpensive effective method for multidimensional control, mainly used to manipulate sliders.

The CyberGlove, invented by Thomas Massie and Mike Daily in the early 1990s, is a notable haptic glove that has made significant contributions to the field of haptic technology [[8]](#eight). Designed to provide users with a sense of touch and tactile feedback in virtual environments, the CyberGlove incorporates flexible sensors embedded in a glove, enabling real-time tracking and interaction with virtual objects. These sensors detect the movement and the position of the user’s fingers and hand, while small vibrators or actuators generate haptic feedback, simulating the sense of touch on the user’s fingertips.

Sunitha K.A et al. in their paper aim to cover the various prevailing methods of deaf-mute communication interpreter system [[9]](#nine). The two broad classifications of the communication methodologies used by the deaf-mute people are – Wearable communication system and online learning system. Under wearable method, there are glove based system, keypad method and handicap Touch screen. All the above mentioned three sub-divided methods make use of various sensors, accelerometer, micro-controller, a text to speech conversion module, and touch-screen. The need for an external device to interpret the message between deaf-mute and non-deaf-mute people can overcome by the second method that means online learning system. The online learning system has different methods. The five sub-divided methods are SLIM module, TESSA, Wi-See Technology, SWI-PELE and Web-sign Technology.

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 [[10]](#ten). 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.

To convert data into voice, Speak Jet is used as sound synthesizer [[11]](#eleven). It uses mathematical Sound Architecture technique to control five channel sound synthesizers to generate a speech signal. It has 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 and 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.

A survey of hand posture and gesture recognition techniques and technology was developed by J. J. LaViola [[12]](#twelve) in Rice University. This paper surveys the use of hand postures and gestures as a mechanism for interaction with computers, describing both the various techniques for performing accurate recognition and the technological aspects inherent to posture- and gesture-based interaction.

R. H. Liang and M. Ouhyoung, “A real-time continuous gesture recognition system for sign language,”[13].In this paper, a large vocabulary sign language interpreter is presented with real-time continuous gesture recognition of sign language using a DataGloveTM. The most critical problem, end-point detection in a stream of gesture input is first solved and then statistical analysis is done according to 4 parameters in a gesture: posture, position, orientation, and motion. They have implemented a prototype system with a lexicon of 250 vocabularies in Taiwanese Sign Language (TWL). This system uses hidden Markov models (HMMs) for 51 fundamental postures, 6 orientations, and 8 motion primitives. In a signer dependent way, a sentence of gestures based on these vocabularies can be continuously recognized in real-time.

Klimis Symeonidis conducted research on Hand Gesture Recognition Using Neural Networks as part of his Master of Science degree in Multimedia Signal Processing.[[13]](#thirteen) The study aimed to recognize static hand gestures, specifically a subset of American Sign Language (ASL), using a pattern recognition system based on neural networks. Unlike previous systems that relied on datagloves or markers, this approach transformed hand gesture images into feature vectors, which were then compared with a training set of gesture vectors. The final system was implemented using a Perceptron network1. This project offers a comprehensive review of the literature on visual interpretation of hand gestures for HCI. Highlighting the appeal of hand gestures as a natural HCI interface, the research categorizes approaches into 3D hand models and appearance-based models, noting computational challenges with the former and limitations in generality with the latter. Despite progress, the review emphasizes the need for further theoretical and computational advancements before hand gestures become widely integrated into HCI, while also suggesting avenues for future research, including the integration of gesture recognition with other modes of interaction.

# CHAPTER 3: RELATED THEORY

## 3.1 Machine Learning

Machine learning is a field of artificial intelligence (AI) that focuses on the development of algorithms and statistical models that enable computers to perform tasks without explicit programming. The core idea is to allow computers to learn and improve from experience. Machine learning is a branch of computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. Machine learning is an important component of the growing field of data science.  Recently, generative [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network) have been able to surpass many previous approaches in performance. Machine learning approaches are traditionally divided into three broad categories, which correspond to learning paradigms, depending on the nature of the "signal" or "feedback" available to the learning system:

* [**Supervised learning**](https://en.wikipedia.org/wiki/Supervised_learning)**:** The computer is presented with example inputs and their desired outputs which is given, and the goal is to learn a general rule that [maps](https://en.wikipedia.org/wiki/Map_(mathematics)) inputs to outputs.
* [**Unsupervised learning**](https://en.wikipedia.org/wiki/Unsupervised_learning)**:** No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end ([feature learning](https://en.wikipedia.org/wiki/Feature_learning)).
* [**Reinforcement learning**](https://en.wikipedia.org/wiki/Reinforcement_learning)**:** A computer program interacts with a dynamic environment in which it must perform a certain goal (such as [driving a vehicle](https://en.wikipedia.org/wiki/Autonomous_car) or playing a game against an opponent). As it navigates its problem space, the program is provided feedback that's analogous to rewards, which it tries to maximize.

Training [machine learning algorithms](https://www.techtarget.com/whatis/definition/machine-learning-algorithm) often involves large amounts of good quality data to produce accurate results. The results themselves can be difficult to understand -- particularly the outcomes produced by complex algorithms, such as the deep learning [neural networks](https://www.techtarget.com/searchenterpriseai/definition/neural-network) patterned after the human brain. And [ML models](https://www.techtarget.com/searchenterpriseai/tip/What-are-machine-learning-models-Types-and-examples) can be costly to run and tune. Machine learning is a rapidly evolving field, and researchers and practitioners are continuously developing new algorithms, techniques, and best practices to improve model performance and address emerging challenges. Some of the most widely used machine learning algorithms are neural network, random forest, decision tree method, SVL, etc.

## 3.2 Neural Network

Neural network is a commonly used machine learning algorithm which works by using process that mimic the way biological neurons work together to identify the phenomena. It then weighs options and then come to a conclusion. Every neural network consists of layers of nodes, or artificial neurons—an input layer, one or more hidden layers, and an output layer. Each node connects to others, and has its own associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network. Neural networks rely on training data to learn and improve their accuracy over time. They are powerful tools in computer science and [artificial intelligence](https://www.ibm.com/topics/artificial-intelligence), allowing us to classify and cluster data at a high velocity. Tasks in speech recognition can take minutes that used to take hours when done manually. One of the best-known examples of a neural network is Google’s search algorithm. Neural networks are sometimes called  ANNs or  SNNs. They are a subset of machine learning. Most deep neural networks are feed forward, meaning they flow in one direction only, from input to output. However, you can also train your model through back propagation; that is, move in the opposite direction from output to input. Back propagation allows us to calculate and attribute the error associated with each neuron, allowing us to adjust and fit the parameters of the model(s) appropriately. There are two types of neural networks:

* Convolutional Neural Network (CNN): CNNs are similar to feed forward networks but they are usually utilized for image recognition, pattern recognition or computer vision. These networks harness principles from linear algebra, particularly matrix multiplication, to identify patterns with image.
* Recurrent Neural Network (RNN): RNNs are identified by their feedback loops. These learning algorithms are primarily leveraged when using time-series data to make predictions about future outcomes, such as stock market predictions or sales forecasting.

Here's a simple breakdown of how a neural network works:

* **Input Layer:** This layer receives the input features or initial data. Each node in the input layer represents a specific feature of the input data.
* **Hidden Layers:** Between the input and output layers, there can be one or more hidden layers. Neurons in these layers process the input data using weights that are adjusted during training. The hidden layers allow the network to learn complex patterns and relationships in the data.
* **Weights and Activation Function:** Each connection between nodes has a weight, representing the strength of the connection. The neural network learns by adjusting these weights during training based on the error in its predictions.

An activation function is applied to the weighted sum of inputs at each node. This introduces non-linearity to the model, enabling it to learn and represent complex relationships in the data.

* **Output Layer:** The final layer produces the network's output or prediction. The number of nodes in this layer depends on the nature of the problem (e.g., binary classification, multi-class classification, regression).
* **Training:** During training, the network is fed input data with known outcomes. The difference between the predicted output and the actual outcome is used to calculate an error. This error is then used to adjust the weights in the network through a process called back propagation.
* **Back propagation:** Back propagation is the core learning algorithm for neural networks. It involves propagating the error backward through the network, adjusting the weights to minimize the difference between the predicted and actual outcomes. This process is repeated over multiple iterations (epochs) until the network converges to an optimal set of weights.

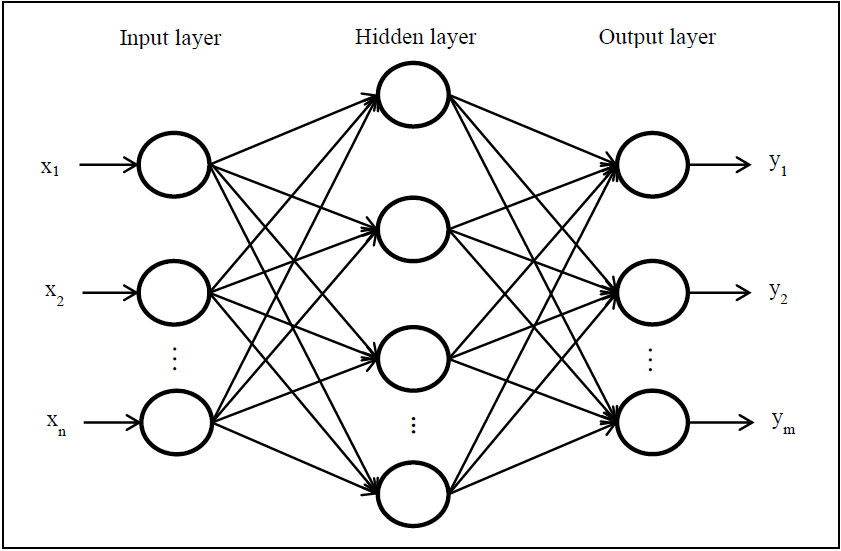


Figure 2: Simple Neural Network

## 3.3 Android Studio

Android Studio is the official integrated development environment (IDE) for Android application development. It is based on IntelliJ IDEA, a Java integrated development environment for software, and incorporates its code editing and developer tools. To support application development within the Android operating system, Android Studio uses a Gradle-based build system, Android Emulator, code templates and GitHub integration. Every project in Android Studio has one or more modalities with source code and resource files. These modalities include Android app modules, Library modules and Google App Engine modules. Android Studio is the official IDE for Android application development. It is based on IntelliJ IDEA, a Java integrated development environment for software, and incorporates its code editing and developer tools. Android Studio provides a complete IDE, including an advanced code editor and app templates. It also contains tools for development, debugging, testing, and performance that make it faster and easier to develop.

## 3.4 Arduino Mega

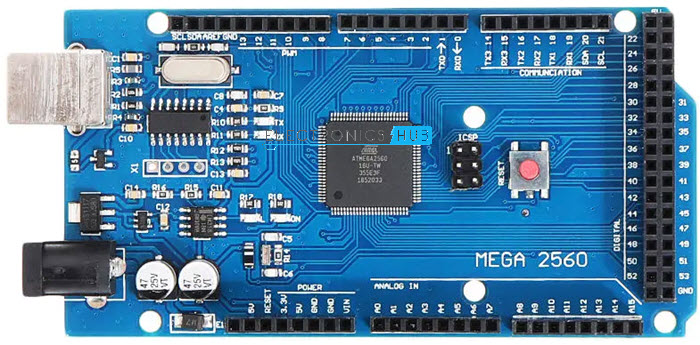


Figure 3.1:Arduino Mega [[14]](#fourteen)

The Arduino Mega 2560 is a microcontroller board based on the Atmega2560. It has 54 digital input/output pins of which 14 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. The Mega 2560 board is compatible with most shields designed for the Uno and the former boards Duemilanove or Diecimila. The Mega 2560 is an update to the Arduino Mega, which it replaces.

It simply needs to be connected to a computer with a USB cable or powered with an AC-to-DC adapter or battery to get started. The board can operate on an external supply of 6 -20V. If supplied less than 7V, however, the 5V pin may supply less than 5V and board may be unstable. The recommended voltage is 7-12V

|  |  |
| --- | --- |
| MICROCONTROLLER | [ATmega2560](http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf) |
| OPERATING VOLTAGE | 5V |
| INPUT VOLTAGE (RECOMMENDED) | 7-12V |
| INPUT VOLTAGE (LIMIT) | 6-20V |
| DIGITAL I/O PINS | 54 (of which 15 provide PWM output) |
| ANALOG INPUT PINS | 16 |
| DC CURRENT PER I/O PIN | 20 mA |
| DC CURRENT FOR 3.3V PIN | 50 mA |
| FLASH MEMORY | 256 KB of which 8 KB used by boot loader |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| CLOCK SPEED | 16 MHz |
| LED\_BUILTIN | 13 |
| LENGTH | 101.52 mm |
| WIDTH | 53.3 mm |
| WEIGHT | 37 g |

Table 1: Tech Specification

## 3.5 Flex sensor



Figure 3: Flex Sensor [[15]](#fifteen)

Flex sensors, also known as bend sensors or flexible sensors, are electronic devices that change their electrical resistance when they are bent or flexed. These sensors are classified into two types based on its size namely 2.2-inch flex sensor & 4.5-inch flex sensor. The size, as well as the resistance of these sensors, is dissimilar except the working principle. These sensors are typically composed of a thin strip or ribbon of a flexible material, such as plastic or silicone, which contains conductive elements.

The sensor’s resistance is lowest when it’s flat on the surface, increases when we bend it slowly and reaches its maximum when it’s at a 90-degree angle. As the sensor bends or flexes, the resistance changes, providing a corresponding change in the electrical output. The resistance of a flex sensor typically increases as it is bent, although some variations may exhibit a decrease in resistance. When the sensor is straight the resistance is about 10K, and when the sensor is bent the value is 22K.

The flex sensor has two pins, one is P1 and the other one is P2, these two pins can be used to retrieve data from the flex sensor. The sensor acts more like a variable resistor, whose resistance changes based on how much it is bent, hence just like a resistor, the pins on this sensor are also interchangeable. The required voltage of this sensor to activate the sensor ranges from 3.3V-5V DC. It can function on low voltage.

Flex sensors are commonly used in various applications where detecting and measuring bending or flexing is necessary. Some sensors are unidirectional, meaning they are sensitive to bending in one direction, while others are omnidirectional, capable of detecting bending in multiple directions. The applications of the flex-sensor include hinge actuators in machinery, robotics whisker sensors, door sensors, and even biotech wearable for position & mobility tracking in human joints, musical instruments, etc.

## 3.6 MPU-6050 Module

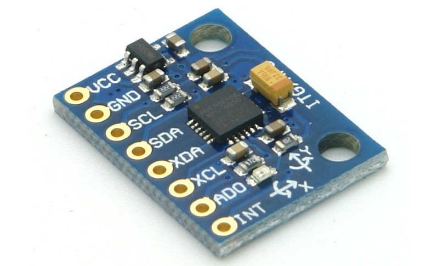


Figure 4: MPU-6050 Module [[16]](#sixteen)

The MPU-6050 module is a widely used motion sensor module that combines a 3-axis gyroscope and a 3-axis accelerometer in a single chip. It is based on the InvenSense MPU-6050 integrated circuit. It integrates both sensors along with an onboard digital motion processor.

It captures motion in X, Y and Z axis at the same time. It is used in different industrial projects and electronic devices to control and detect the 3-D motion of different objects. It helps us to measure velocity, orientation, acceleration, displacement and other motion like features. The MPU-6050 is a 6 DOF (degrees of freedom) or six-axis IMU sensor, which means it gives six values as output: three values from the accelerometer and three from the gyroscope. It is a sensor based on MEMS (micro electro mechanical systems) technology. Both the accelerometer and the gyroscope are embedded inside a single chip.

The MPU6050 module has a total of 8 pins. In which at least 4 pins are necessary for the interfacing. The pin out of a MPU6050 module is as follows:

VCC: Provides power for the module, Connect to the 5V pin of the Arduino.

GND: Ground Connected to Ground pin of the Arduino.

SCL: Serial Clock Used for providing clock pulse for I2C Communication.

SDA: Serial Data Used for transferring Data through I2C communication.

XDA: Auxiliary Serial Data – Can be used to interface other I2C modules with MPU6050.

XCL: Auxiliary Serial Clock – Can be used to interface other I2C modules with MPU6050.

ADD/ADO: Address select pin if multiple MPU6050 modules are used.

INT: Interrupt pin to indicate that data is available for MCU to read.

### 3.6.1 Accelerometer

Micro-electro-mechanical systems (MEMS) accelerometers are used wherever there is a need to measure linear motion, either movement, shock, or vibration but without a fixed reference. The accelerometer measures the acceleration forces acting on an object, in order to determine the object’s position in space and monitor the object’s movement. Acceleration, which is a vector quantity, is the rate of change of an object’s velocity (velocity being the displacement of the object divided by the change in time). All accelerometers work on the principle of a mass on a spring, when the thing they are attached to accelerates, then the mass wants to remain stationary due to its inertia and therefore the spring is stretched or compressed, creating a force which is detected and corresponds to the applied acceleration. Precise linear acceleration detection in two orthogonal axes is achieved by a pair of silicon MEMS detectors formed by spring ‘proof’ masses. When the sensor is subjected to a linear acceleration along its sensitive axis, the proof mass tends to resist motion due to its inertia, therefore the mass and its fingers become displaced concerning the fixed electrode fingers. The gas between the fingers provides a damping effect. This displacement induces a differential capacitance between the moving and fixed silicon fingers which is proportional to the applied acceleration. This change in capacitance is measured with a high-resolution ADC and then the acceleration is calculated from the rate of change in capacitance. In MPU6050 this is then converted into readable value and then it’s transferred to the I2C master device.

Accelerometers are widely used in smartphones, tablets, and gaming consoles for screen orientation changes, shake detection, and gesture-based controls. They enable features like auto-rotation, tilt-to-scroll, pedometers, and gaming motion controls. Accelerometers are used in conjunction with gyroscopes to track the movement of vehicles, aircraft, and drones. By integrating acceleration data over time, accelerometers help estimate position, velocity, and heading when GPS signals are unavailable or unreliable. Accelerometers are employed in automotive safety systems, such as airbag deployment, to detect sudden deceleration or impacts.

### 3.6.2 Gyroscope

The MEMS Gyroscope contains a set of four proof mass and is kept in a continuous oscillating movement. When an angular motion is applied, the Coriolis Effect causes a change in capacitance between the masses depending on the axis of the angular movement. This change in capacitance is sensed and then converted into a reading. Here is a small animation showing the movement of these proof masses on the application of an angular movement for different axis.

## theengineeringproject3.7 Bluetooth module

Figure 5: Bluetooth Module [[17]](#seventeen)

Bluetooth module (Bluetooth module) refers to the basic circuit set of the chip with integrated Bluetooth function, used for short-range 2.4G wireless communication module. For the end user, the Bluetooth module is a semi-finished product. Through the process of functional redevelopment and packaging of the shell based on the module, the final product capable of utilizing Bluetooth communication is realized. Generally, it refers to the module that supports the Bluetooth protocol below 4.0, which is generally used for relatively large data transmission, such as voice, music and other high data transmission. The device works on the frequency range from 2.402 GHz to 2.480GHz. HC-06 module has six pin . In them we only need to use four for successfully interfacing the module. The four pins are VCC, ground, transmitter and receiver. It operates on 4V to 6V.

## 3.8 Tools Used

### 3.8.1 Python

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

### 3.8.2 Scikit-learn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

### 3.8.3 Pandas

Pandas is the most popular software library for data manipulation and data analysis for the Python programming language. As an open-source software library built on top of Python specifically for data manipulation and analysis, Pandas offers data structure and operations for  powerful, flexible, and easy-to-use data analysis and manipulation.

### 3.8.4 Jupyter Notebook

This interactive coding environments allow us to develop and experiment with our code, making it easier to iterate and visualize result.

## 3.9 Verification and Validation

### 3.9.1 Accuracy

Classification accuracy is the accuracy we generally mean, whenever we use the term accuracy. We calculate this by calculating the ratio of correct predictions to the total number of input Samples.

3.1

### 3.9.2 Precision

Precision is a measure of a model’s performance that tells you how many of the positive predictions made by the model are actually correct. It is calculated as the number of true positive predictions divided by the number of true positive and false positive predictions.

### 3.2 3.9.3 Recall

Lower recall and higher precision give you great accuracy but then it misses a large number of instances. The more the F1 score better will be performance. It can be expressed mathematically in this way:

3.3

### 3.9.4 F1 Score

It is a harmonic mean between recall and precision. Its range is [0,1]. This metric usually tells us how precise (It correctly classifies how many instances) and robust (does not miss any significant number of instances) our classifier is.

3.4

### 3.9.5 Confusion Matrix

A confusion matrix is a performance evaluation tool in machine learning, representing the accuracy of a classification model. It displays the number of true positives, true negatives, false positives, and false negatives. This matrix aids in analyzing model performance, identifying misclassifications, and improving predictive accuracy.

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the total number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

# CHAPTER 4: METHODOLOGY

## 4.1 Hardware Assembling

We have used five flex sensors which goes in each of the finger of the palm i.e. thumb, index, middle, ring and little finger respectively. The flex sensor measures value with the bending of the finger. MPU6050 goes in the back of the palm. It includes two components namely accelerometer and gyroscope. MPU6050 is used to determine the position and bending of the palm. Flex sensors and MPU6050 is to be interfaced with Arduino Mega 2560.

Flex sensors are given 5V source voltage. They are connected in analog pin of Arduino. MPU6050 and Arduino Mega 2560 is to be interfaced with each other.

Arduino Mega is to be programmed to send data from five flex sensors. Data is received in laptop and processed by python programming language.



Figure 6: Hardware Assembling

## 4.2 Dataset Preparation

### 4.2.1 Dataset Collection

The dataset for training the machine is to be collected manually as the dataset was not available on online mediums. We have collected about 12000 data manually. There are six sign languages. So for one sign language there are exactly 2000 data points each.

### 4.2.2 Data Preprocessing

Data pre-processing is the process of cleaning and preparing the text for feature extraction and classification. The collected data is neatly arranged in MS Excel which creates csv file. This data is shuffled and randomized. The risk of over fitting is expected in training model in this process. Label Encoding is a technique used in machine learning to convert categorical data into numerical format. We have used Label Encoder to convert commands from categorical values to numerical values.

### 4.2.3 Dataset Description

The data is then categorized as per what the output should generate. This is the must process for supervised learning models. The data is categorized into six parts as per the movement of hand. These six categories represent six commands individually. This is given in the code so that the output is translated and displayed according to the movement of hand.

## 4.3 Block Diagram

Flex Sensors

Gyro-meter

Arduino Mega 2560

Accelerometer

Process

Output

Screen

Data Acquisition

Feature Extraction

Bluetooth module

Speaker

Classification

Figure 7: Block Diagram

The above figure shows the simple block diagram of the process of the project completed. We have used five flex sensors which goes in each of the fingers that is index finger, middle finger, ring finger, small finger and thumb. Flex sensor’s value gives resistance. The value of resistance changes according to the bending of finger. The value is maximum when the fingers are bent completely and reduces gradually as we open them. MPU-6050 is attached to the back of the hand. Its value changes according to the movement of arms. The axes in which MPU gives value is roll and pitch. The value from these flex sensors and MPU is taken and sent to arduino. The dataset that was created manually was already uploaded in arduino. The sent data points are compared to the existing data points and corresponding data is selected. The command is given is English language. This command is translated to Nepali language using Google Translator. This translated command is sent to mobile application which then displays that Nepali command in the form of text and speech.

## 4.4 Algorithm

### 4.4.1 Algorithm for training and validation of model

Step 1: Start

Step 2: Set the baud rate similar to Arduino and open Arduino IDE

Step 3: Collect enough data (>400) for a single command from serial port at real time in Arduino spreadsheet

Step 4: Close Arduino IDE

Step 5: Convert collected data to CSV file in exce.

Step 6: Split the data into testing and training data

Step 7: Train the model on the training data

Step 8: Test the model on training and testing data

Step 9: Collect the test metrics: confusion matrix, accuracy, precision, recall and f-score

Step 10: Evaluate the model using the test metrics and recollect data if the metrics are not satisfactory

Step 11: End

### 4.4.2 Algorithm for real time application

Step 1: Start

Step 2: Train the model using training dataset

Step 3: Set the baud rate similar to Arduino

Step 4: Is there data at serial port?

If no, wait till the data is available

Else go to step 5

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

Step 6: Pass data through the model

Step 7: Trained machine predicts the output

Step 8: Display the prediction on the screen and convert it to speech

Step 9: End

## 4.5 Flowchart

### 4.5.1 Flowchart for Dataset Preparation

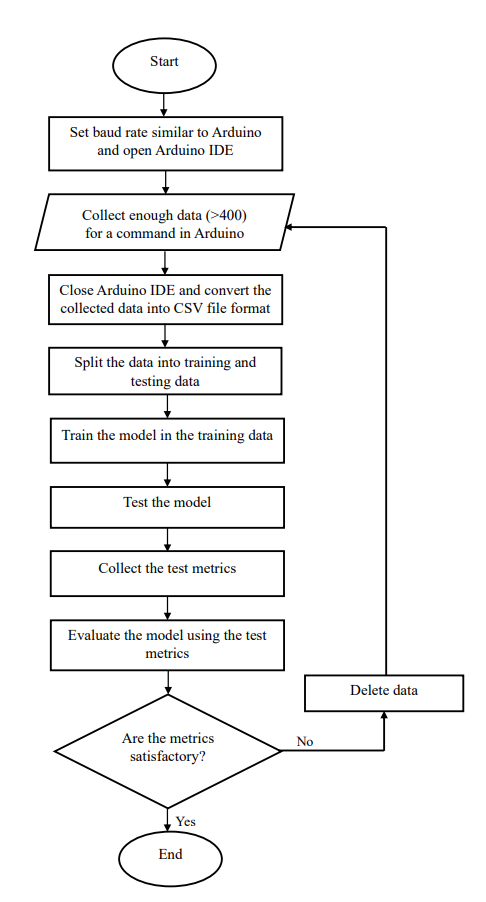


Figure 8: Flowchart for dataset preparation

### 4.5.2 Flowchart for real time application

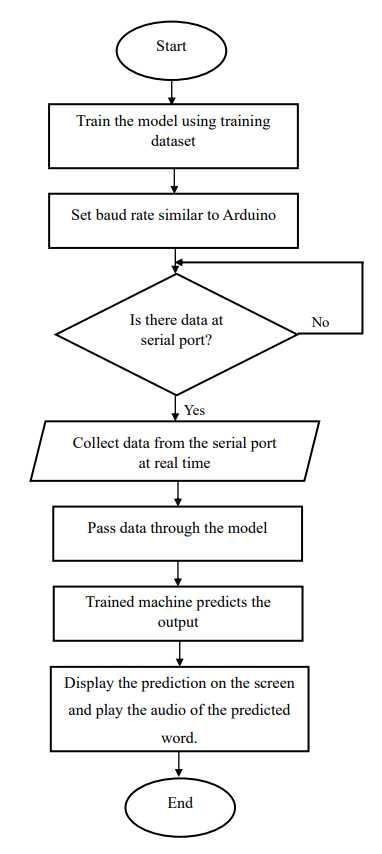


Figure 9: Flowchart for real time application

# CHAPTER 5: RESULT AND ANALYSIS

## 5.1 Completed Results

In this chapter we talk about our completed tasks and results produced. We have programmed this project to recognize and display upto six sign languages. The works that were performed are as follows:

### 5.1.1 Research Works

After analyzing the different supervised learning models for a classification problem involving data from flex sensors embedded in a glove, Neural Network has been selected as the most suitable algorithm. This is because the dataset contains continuous numerical input data that is not highly correlated, and Neural Network can handle numerical data well and capture complex non-linear relationships between input features and output classes.

In contrast, regression models were highly unsuitable for this dataset. Naïve Bayes assumes that probabilities are independent, which is not always the case for this dataset, and K-Nearest Neighbor would not perform well as the inputs are not highly correlated between various inputs. We applied multiple machine learning models, among which neural network and random forest performed the best. As random forest is the preferred algorithm for its ability to handle complex classification problems with numerical data that are not highly correlated between input features and we observed past projects done using random forest on similar context, we initially chose random forest algorithm. However, Random Forest did not give desired output for our dataset. Neural network is the one that seemed highly suitable for this one, the multilayer perceptron can theoretically be used for perfect classification. We collected enough data and can now use the multilayer perceptron and classify the data perfectly. Therefore, we chose neural network as it worked the best.

### 5.1.2 Data Collection

During this phase, a total of 12000 data points were collected using an Arduino device to facilitate data transmission. The values received from flex sensors and MPU was displayed in serial monitor which was then captured using the software called Cool Term. In order to generate the spreadsheet, the data needed to be properly parsed using the tab character (/t) as a column separator. The resulting data could then be easily transferred into a Microsoft Excel spreadsheet, which served as the foundation of the initial dataset.

### 5.1.3 Writing data into Excel file and Labeling

Following the data collection phase, it was necessary to label each row with the corresponding command being implemented. A total of six distinct commands were developed and applied, with each row meticulously labeled to ensure accurate representation of the associated command.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a\_x | a\_y | a\_z | g\_x | g\_y | g\_z | Thumb | Index | Middle | Ring | Small | Cx |
| 16340 | 836 | 3620 | -317 | 133 | -221 | 630 | 824 | 649 | 674 | 488 | C0 |
| 16308 | 868 | 3648 | -468 | 103 | -250 | 630 | 829 | 653 | 676 | 486 | C0 |
| 16144 | 964 | 3876 | -512 | 162 | -273 | 630 | 828 | 651 | 671 | 489 | C0 |
| 16584 | -896 | 3172 | -482 | 234 | -78 | 424 | 732 | 453 | 499 | 393 | C1 |
| 16432 | -780 | 3172 | -211 | 237 | 113 | 424 | 730 | 451 | 500 | 391 | C1 |
| 16476 | -688 | 3172 | -337 | 193 | -161 | 422 | 731 | 454 | 499 | 394 | C1 |
| 948 | 3832 | -16832 | -513 | 2252 | 340 | 424 | 819 | 550 | 688 | 472 | C2 |
| 608 | 4160 | -17204 | -627 | 2106 | 407 | 423 | 820 | 550 | 689 | 473 | C2 |
| 680 | 4724 | -19288 | -760 | 2260 | 492 | 426 | 820 | 550 | 688 | 473 | C2 |
| 976 | 16248 | -4740 | -527 | -251 | 682 | 399 | 828 | 592 | 722 | 374 | C3 |
| 852 | 15672 | -4688 | -646 | -223 | 749 | 400 | 818 | 596 | 720 | 374 | C3 |
| 492 | 16076 | -4048 | -457 | -222 | 726 | 400 | 807 | 591 | 719 | 374 | C3 |
| 14712 | 5296 | 5548 | -62 | 698 | 215 | 481 | 814 | 448 | 491 | 14712 | C4 |
| 14808 | 5460 | 5008 | -76 | 690 | 295 | 487 | 806 | 447 | 493 | 14808 | C4 |
| 14852 | 5064 | 5220 | -63 | 662 | 218 | 492 | 816 | 448 | 495 | 14852 | C4 |
| 15276 | -4832 | 3160 | 126 | 568 | -853 | 550 | 730 | 508 | 615 | 379 | C5 |
| 15596 | -4436 | 3716 | 183 | 518 | -824 | 544 | 708 | 507 | 614 | 379 | C5 |
| 15232 | -4624 | 3392 | 181 | 546 | -724 | 548 | 711 | 487 | 640 | 379 | C5 |
| 6228 | -15604 | -916 | -1400 | -1198 | 36 | 400 | 814 | 472 | 630 | 453 | C6 |
| 6692 | -15812 | -1992 | -1064 | -1216 | -215 | 400 | 809 | 473 | 636 | 446 | C6 |
| 6620 | -15264 | -1196 | -1282 | 813 | -1602 | 402 | 817 | 489 | 607 | 453 | C6 |

Table 2: Dataset

### 5.1.4 Model Performance

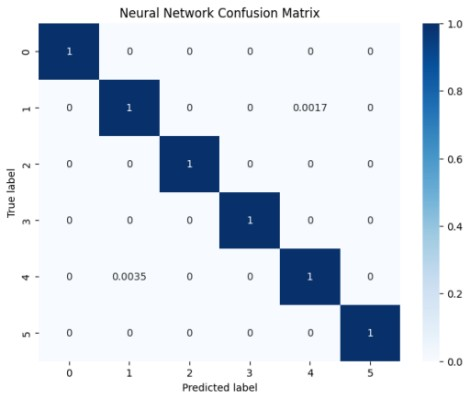


Figure 10: Confusion Matrix for Testing Data

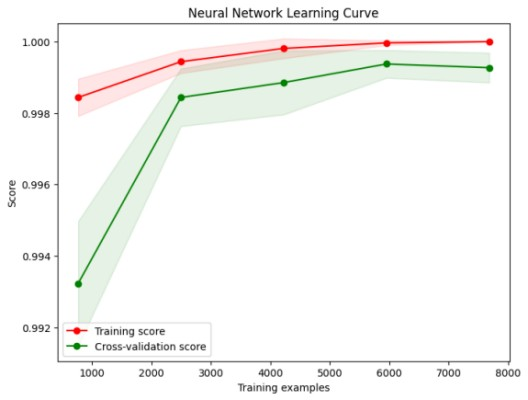


Figure 11: Learning Curve

The above two figure shows the confusion matrix and learning curve of neural network. We know, neural network works on layers such as input layer, hidden layer, output layer. Since we have taken 11 features as input which gives six outputs, the input layer has 11 nodes and output layer has 6 nodes. Our project has simple architecture with 12,000 data points and one hidden layer which is a moderate size for a neural network so it contributes to shorter training time of 12.8 seconds but training time can increase with larger dataset.

### 5.1.5 Comparison among models

|  |  |  |  |
| --- | --- | --- | --- |
| S.N. | Model | Training Accuracy | Validation Accuracy |
| 1 | Logistic Regression | 99.499404 | 99.249464 |
| 2 | Decision Tree | 99.223242 | 99.070765 |
| 3 | Random Forest | 99.490107 | 99.399643 |
| 4 | SVM | 99.380215 | 99.142244 |
| 5 | Neural Network | 99.514439 | 99.486620 |

Table 3: Comparison among Models

This table compares various machine learning algorithms. We have calculated training accuracy and validation accuracy of each of the four algorithms. We observe that the algorithm with highest training as well as validation accuracy is neural network and SVM has the lowest among all.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N. | Algorithm | Precision | Recall | F1 |
| 1 | Logistic Regression | 99.260206 | 99.249464 | 99.250396 |
| 2 | Decision Tree | 99.080303 | 99.070765 | 99.072326 |
| 3 | Random Forest | 99.508554 | 99.399643 | 99.300678 |
| 4 | SVM | 99.159099 | 99.142244 | 99.143154 |
| 5 | Neural Network | 99.6512207 | 99.830288 | 99.740674 |

Table 4: Performance Parameter

In this table we see that precision, recall and F1 are highest for neural network algorithm and lowest for decision tree. Precision is a measure of the accuracy of the positive predictions made by the model. Recall is the ratio of correctly predicted positive observations to the total actual positives. F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall and is particularly useful when there is an imbalance between the classes.

Taking all these measures into considerations, we have concluded that Neural network Algorithm is the best suited for our project since it has highest precision, recall and F1 score as well as the highest training accuracy and validation accuracy.

### 5.1.6 Translation

We have converted few of the most essential phrases that is needed to communicate for specially abled person. For this we used Google translator API.

### 5.1.7 Display

The output goes to app via Bluetooth module. The output is a text in Nepali language which is then converted to speech.

### **5.1.8 Commands and Outputs**

The following figures shows the six gestures that are predicted by the system and the displayed command which is translated to Nepali language is also mentioned.



Command: Help Me

Translated Command: मलाई सहयोग गर

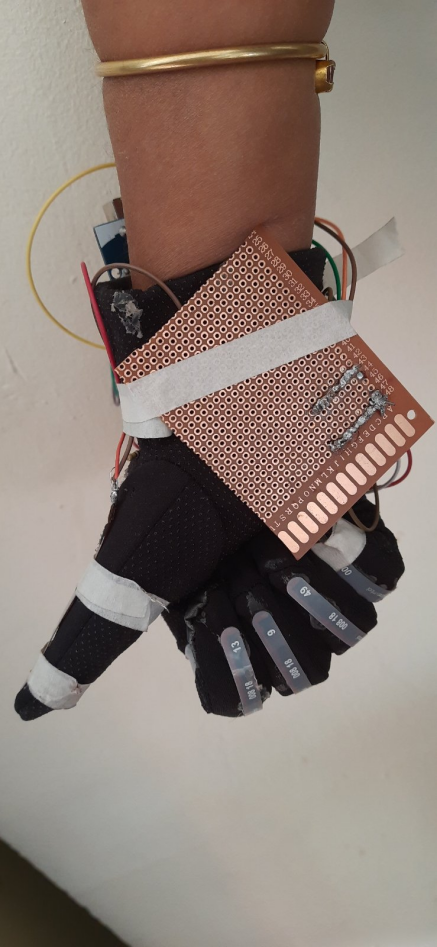
Figure 12: Command 1



Command: I am thirsty

Translated Command: मलाई तिर्खा लागेको छ

Figure 13: Command 2



Command: I am hungry

Translated Command: मलाई भोक लाग्यो

Figure 14: Command 3



Command: I want to call.

Translated Command: म कल गर्न चाहन्छु

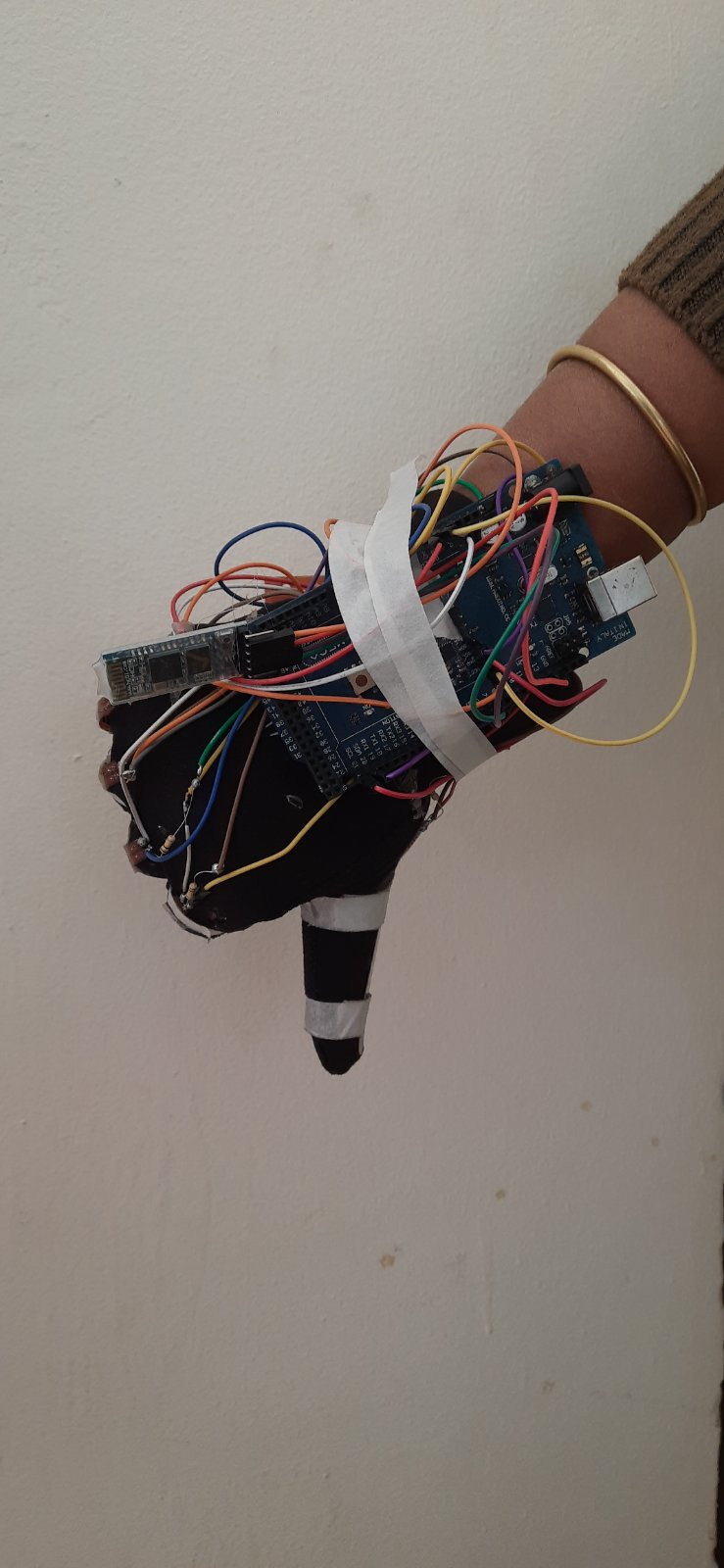
Figure 15: Command 4



Figure 16: Command 5

Command: Yes

Translated Command: हुन्छ



Command: No

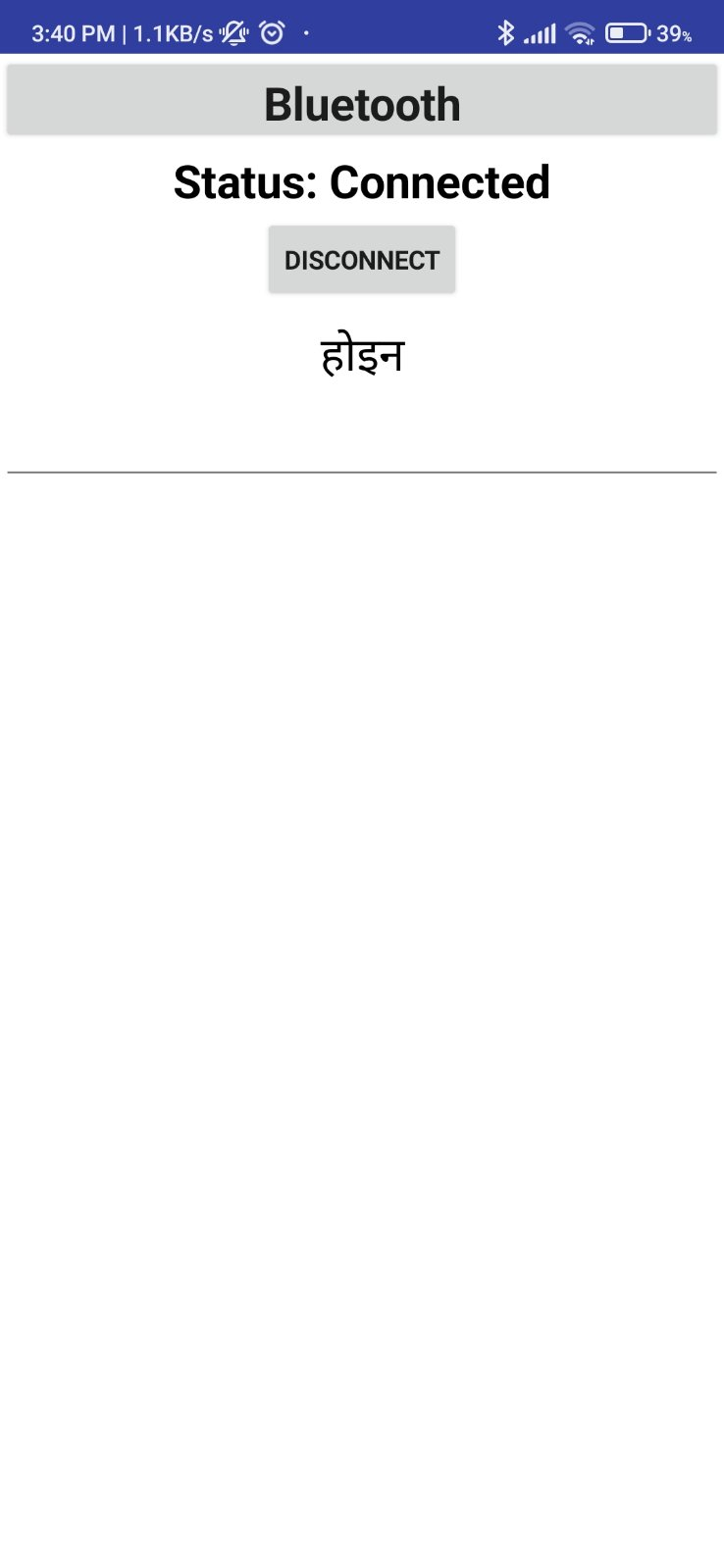
Translated Command: होइन

Figure 17: Command 6

### 5.1.9 Display in Application

The commands that were detected was transmitted through Bluetooth module to mobile application. These commands were displayed in Devnagari system. The commands were also spoken out loud through the mobile application. Total of six gestures were predicted. For the development of mobile application, we used Android Studio.





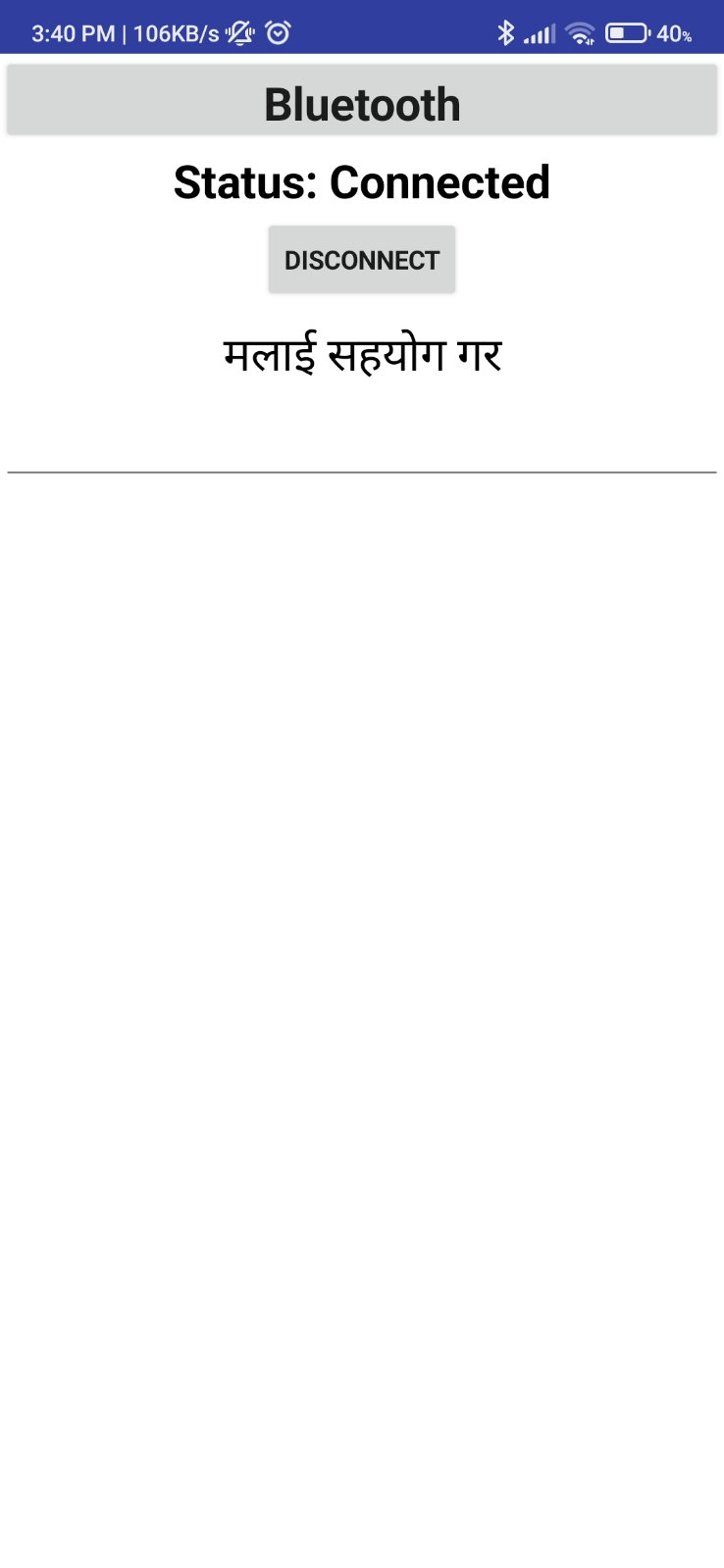


Figure 18: Display in application in the form of text

### 5.1.10 Problems Encountered

During the process of development of the project we encountered various problems and tackled them successfully. The problems encountered are discussed below:

**1. Flex Sensor Value Calibration Issue:**

**Problem Description**: During the initial phases of our project, we encountered difficulties in calibrating the values obtained from the flex sensors. Despite our efforts to ensure accuracy, the sensor readings exhibited inconsistencies, leading to inaccurate predictions by the model.

**Project Impact**: The calibration issue significantly hindered the reliability and effectiveness of our project. Inaccurate sensor readings directly impacted the training process, resulting in suboptimal model performance and unreliable gesture recognition.

**Resolution approach**: After looking upon it, we came to understand that it was because the flex sensors provided to us were not in the perfect working condition. To solve this problem, we were provided with new sensors.

**2. Inadequate Dataset Size (800 readings per command):**

**Problem Description**:Initially, the dataset size consisted of 800 readings for each sign language command. However, despite this seemingly substantial dataset size, the model struggled to accurately predict gestures during testing and real-time usage.

**Project Impact**: The insufficient dataset size directly impacted our model's ability to learn and generalize complex patterns inherent in sign language gestures. As a result, the model's predictive accuracy was compromised, leading to unreliable performance in recognizing and interpreting gestures.

**Resolution Approach**:To overcome this challenge, we expanded the dataset size by collecting 2000 data for each sign language command. By increasing the quantity and diversity of training examples, the model could better capture the underlying patterns and variations in gesture movements, leading to improved prediction accuracy and robustness.

**3. Bluetooth Connectivity Issue:**

**Problem Description**: During the project implementation, intermittent connectivity problems occurred between the Arduino and the Bluetooth module (HC-05). As a result, the mobile app connected via Bluetooth couldn't consistently display text and speech outputs.

**Project Impact**: These connectivity issues disrupted real-time interaction and usability of the sign language recognition system, compromising user experience.

**Resolution Approach**: Troubleshooting steps were taken to address potential causes, including power supply, and compatibility. Solutions ensuring adequate power supply, and optimizing Bluetooth connection setup and error handling in Arduino code. Thorough testing and refinement were conducted to ensure reliable communication between the Arduino and the mobile application.

### 5.1.11 Future enhancement

**1. Expand Gesture Vocabulary:**

Increase the number of recognized gestures to support a broader range of commands or actions, allowing users to perform more complex interactions with the system.

**2. User Interface Enhancements:**

Improve the user interface of the mobile app to enhance usability and accessibility, such as adding visual feedback for recognized gestures or integrating voice-controlled navigation options.

# REFERENCES

[1] DiPietro, L, Sabatini, A.M., & Dario, P. “A survey of glove-based systems and their applications.” In: IEEE Transactions on systems, Man, and Cybernetics, vol.38, 2008, pp.461-482.

[2] Nallusamy, C., Ariharan, A., Arun, P.K., & Sabith, K.N. “Sign Language Recognition.” In: International Journal of Health Science, vol.6, 2022, pp.13-22.

[3] Sturman, D,J., & Zeltzer, D. “A Survey of Glove-Based Input.” In: IEEE Computer Graphics and Applications, vol.14, January 1994, pp.30-39.

[4] Murakami, K., & Tguchi, H. “Gesture Recognition using Recurrent Neural Networks.” Conference on Human factors in Computing Systems, 1991, pp.237-242.

[5] Ahuja, M.K., Singh, A., & Kaur, A. “Hand Gesture Recognition using PCA.” In: International Journal of Computer Science Engineering and Technology (IJCSET), vol.5, July 2015, pp.267-272.

[6] Fels, Sydney, & Geoffrey, Hinton. “Glove Talk II.” Department of Computer Science, University of Toronto, Toronto, November 1992.

[7] Sandin, Daniel J., & Thomas Defanti. “The Sayre Glove” National Endowment for the Arts, Electronics Visualization Laboratory, Chicago, 1997.

[8] Massoe, T., & Daily, M. “The Glove: A Virtual Reality System for hand Tracking and Gesture Recognition.” Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, 1995, pp.177-185.

[9] Sunitha, K.A., Saraswathi, Anithi P., Aarthi, M., Jayapriya,K., & Lingam,Sunny. “Deaf Mute Communication Interpreter- a Review.” In: International Journal of Applied Engineering research, vol.11, 2016, pp.290-296.

[10] Ata U.R., Salman A., Muhammad A., & Raheel Y., "Microcontroller and Sensors Based Gesture Vocalizer," Rachna College of Engg. and tech., Air University Islamabad, Islamabad, Research Paper 2008.

[11] Magnevation Speak Jet Inc. USA “Speak Jet”, 4 June 2010.Available: <http://www.speakjet.com/>.

[12] LaViola, J.J. “A survey of hand posture and gesture recognition techniques and technology,” Brown University, Jun. 1999, pp. 11-99.

[13] Liang, R.H. & Ouhyoung, M. “A real-time continuous gesture recognition system for sign language,” In: Proc. IEEE Int. Conf. Autom. Face Gesture Recognit., 1998, pp. 558–567.

[14] <https://www.electroschematics.com/arduino-mega-2560-pinout/>

[15]<https://www.amazon.com/Taidacent-Sensor-Pressure-Electronic-Resistive/dp/B0821FNGBD>

[16] <https://www.electronicwings.com/sensors-modules/mpu6050-gyroscope-accelerometer-temperature-sensor-module>

[17]https://www.theengineeringprojects.com/2019/10/hc-05-bluetooth-module-pinout-datasheet-features-applications.html