SMART GLOVE FOR RECOGNITION OF SINHALA SIGN LANGUAGE

V.K.O.N THALPAWILA

(IA -178668G)

Degree of Master of Science in Industrial Automation

Department of Electrical Engineering

Faculty of Engineering

University of Moratuwa

Sri Lanka

June 2022

SMART GLOVE FOR RECOGNITION OF SINHALA SIGN LANGUAGE

V.K.O.N Thalpawila

(IA -178668G)

Dissertation submitted in partial fulfilment of the requirements for the degree Master of Science in Industrial Automation

Department of Electrical Engineering

Faculty of Engineering

University of Moratuwa

Sri Lanka

June 2022

DECLARATION OF THE CANDIDATE & SUPERVISOR

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date: 30/06/2022

The above candidate has carried out research for the Masters thesis under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of the supervisor:

Signature of the supervisor: Date: 30/06/2022

DEDICATION

To my mother and grandmother
who made all of this possible,
for making me who I am

ACKNOWLEDGEMENTS

I would like to express my utmost gratitude to my MSc research advisor Prof. D.P Chandima for the incessant support of my MSc study and related research, for his patience, motivation, and immense knowledge helped me to successfully complete the research and writing of this thesis.

Next, I would like to thank my friends and colleagues for their support, encouragement, and insightful comments. Their hard questions lead me to widen my research from various perspectives.

My sincere thank goes to my mother, husband, and aunt(mamma) who provided me support, and encouragement. It would not be possible to carry out this research successfully without their precious support and encouragement.

Finally, I should thank each and every friend and colleague whose names have not been mentioned here, for the support, encouragement and guidance provided to make the educational process a success.

ABSTRACT

Speech and hearing-impaired people used sign language to communicate with each other. Sign languages are made of gestures. The language consists of different gestures instead of letters or words.

The purpose of this research work is to reduce the communication gap between normal people and hearing and speech impaired people. The research incorporates a system comprising of a glove-based mechanism, consisting of sensors to recognize the hand gestures for Sinhala sign language (SSL) alphabet.

The solution combines electronics, sensors, embedded systems, machine learning algorithms, and natural language processing. The research based on a data glove with flex sensors that measure finger bending and an Inertial Measurement Unit (IMU) to recognise palm-turning gestures of the alphabet. Further, sample data with eleven independent variables and hundred data samples per gesture was used for the purpose. In the proposed system, data is trained and classified using Random Forest machine learning algorithm. And natural language processing (NLP) is completed using a newly developed Application Programming Interface (API) to make Sinhala words.

The results show that the proposed algorithm has a better recognition effect on gestures, and is capable of making words and sentences. The accuracy of the model on the prepared dataset was founded as 99% for the target user with regard to random forest classification.

Complete training for all possible combination of letters and preparation of words is necessary to continue NLP. Also, the system can customise as an education platform for sign language learners. Further, the developed smart glove can use separately for any other hand gesture base applications, the developed ML base system can use or customize separately for feature extraction of any smart wearable item, and finally, the newly developed Sinhala API can use separately for any Sinhala sign language base NLP research work.

Keywords: Sinhala sign language, Hand gesture, Machine learning, Data glove

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
AI	Artificial Intelligence
API	Application Programming Interface
ASL	American Sign Language
EGALE	Easily Applicable Graphical Layout Editor
EMG	Electromyography
FN	False Negative
FP	False Positive
HRI	Human-Robot Interaction
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit
ML	Machine Learning
NLP	Natural Language Processing
PCB	Printed Circuit Board
SSL	Sinhala Sign Language
TP	True Positive
TN	True Negative
WHO	World Health Organization

LIST OF APPENDICES

1 INTRODUCTION

1.1 Background

There is a communication gap between the community who are impaired in speech, hearing or both and the normal community. Sign language is the primary medium for their communication.

As a part of human-computer interaction research, gesture recognition has become a research hotspot in the past three decades. Gesture recognition is a field having a wide variety of applications. Most popular human-computer interactions are computer games, 3D mouse, virtual keyboard, and 3D animations. In addition, automation control, sign language recognition, and traffic signal are industry-related applications.
There are 5% of the world's population is impaired in speech and hearing [1]. Therefore, there is demand and requirement for technological advancement for research based on sign language and gestural recognition.

A wearable sensor-based recognition system generally uses a smart data glove to acquire gesture data. The data glove can collect data on the gesture using the hand orientation, and the bending angle of the finger. The advantage of the data glove is that the collected data is relatively stable and the signal processing is simple [2].

The proposed system provides the speech impaired people a method that can convert their gestures into letters, words, and sentences which can be read by both normal community and speech and hearing-impaired community. The data glove consists of five flex sensors and Inertia Measurement Unit (IMU), all together which can extract enough features on hand posture such as bending of fingers and palm orientation. These motion sensors used are based on acceleration, geomagnetism, and gyroscopes are not limited by external conditions, and can perform gesture recognition without limiting by environmental conditions.

All 56 letters of Sinhala sign language (SSL) were successfully extracted using the developed hand glove. The maximum accuracy was obtained using the Random Forest algorithm for 56 letters of SSL alphabet. As the second step, words and sentences spelled by the SSL were recognised and classified. Optimizing with further natural language processing (NLP) there are no limitations for fingerspelling and word or sentence creation.

1.2 The significance of the study

Every language in the world is unique in its own way and varies with the countries and regions. This norm equally applies to the language of signs. Sign language is made of gestures, like any other language is made of letters or words. This is the only communication medium of speech and hearing-impaired community in the world.

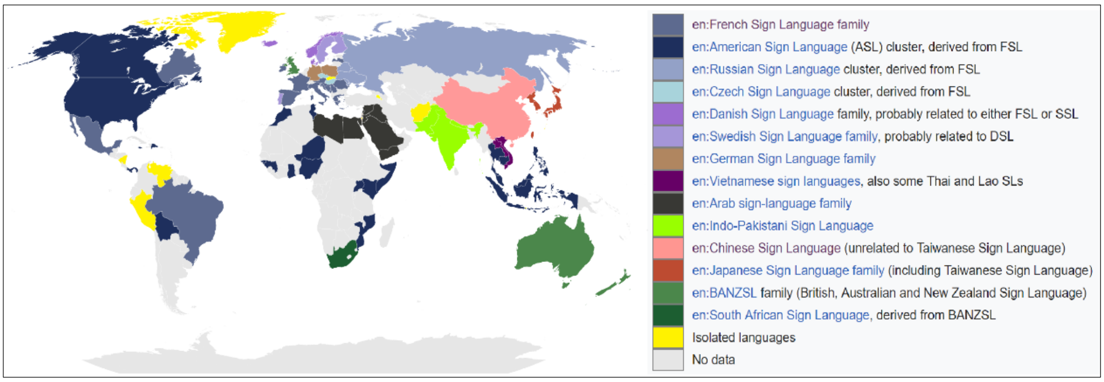


Fig. 1.1. Different sign language families. Adapted from [3]

Over the years, research work has been carried through various research aspects to reduce afore mention communication gap between speech and hearing-impaired communities and normal communities. These facilitated the learning, understanding, and most of all communication methods. American sign language (ASL) is used in America and other countries when using the English language. Therefore, most research works subjected to ASL. Besides ASL, there are a few languages such as British Sign Language (BSL), Chinees Sign Language (CSL), and Japanese Sign Language (JSL) which going through significant research works. But for today sign languages use in the developing world has very less research index [4].

While considering NLP, sign language also a natural language. As same as any other natural language it has its own syntax.

Recent developments in artificial intelligence (AI) and NLP fields bring highly advanced technological means and methods to human language-related research. Although modern AI technologies are developed and carrying NLP research, sign language-related NLP research is often ignored [5].

Also, there is no such system for Sinhala sign language recognition. The alphabet of Sinhala sign language consists of 56 alphabets.

Fig. 1.2. Alphabet of Sinhala sign language

1.3 Literature review

1.3.1 Background

Literature reveals that after decades all the hand gesture recognition-related research work and publications carried out can be categorized into three methods. Which are Vision based, EMG (Electromyography) based, and Glove based.

One of the oldest methods is the vision-based approach. For this purpose, camera captured images of gestures are processed by using different algorithms. The main drawback of the approach is that the camera-captured images or video are 2D and there is no impact on movement gestures. And background of the image and lightning conditions has considerable impact for input recognition. But as an advancement for the same approach, there are special hardware systems that output skeletal information. Microsoft Kinect and Leap motion sensor are better example of such systems.

Fig. 1.3. Leap motion controller. Adapted from [6]

Both EMG related and glove related systems are part of sensor-based sign language recognition approach. Inertial sensors (gyroscopes, accelerometers and magnetometers), pressure (tactile) sensors, EMG sensors (surface electromyography) and strain sensors are used to sense the movements of the hand gestures in sensor-base sign language approach. These sensor readings do not change due to environmental and background conditions when comparing to the camera-bases systems.

These sensors can be embedded or mounted on flexible joints or stretching muscles of moving part of the body which generate the gesture. But it may lead signer to movement discomfort. The main disadvantage of sensor-based approached is the possibility of discomfort or movement restriction due to sensor configuration to the user. The reduction of signers' ability to move their hand or body freely has huge impact on the accuracy of the data. Therefore, it is necessary to mount these sensors on wearables using proper methods. The research on sensor base wearable begins with this challenge. The most common approaches can be muscle guard, writs band, finger rings or hand glove with novel ideas.

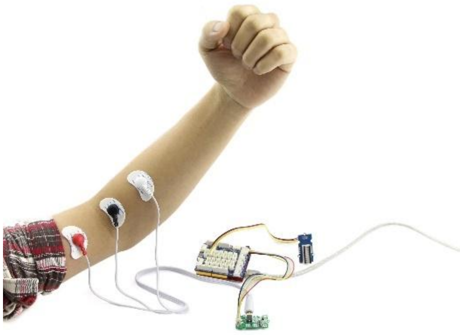


Fig. 1.4. Electromyography sensor. Adapted from [7]

Myo gesture control armband is EMG related product which has demonstrated ten hand gestures [8]. Compared to other wearable sensor-related accessories currently in use, the glove is most suitable for finger spelling applications, because glove covers most of the essential parts of the hands to identify the hand gestures. In addition to afore mentioned sensors there is wearable glove system developed using simple bimodal capacitive sensors to recognize complex gestures [12].

Literature revels there are gesture recognitions systems which involve both image processing and sensor base technique. The main disadvantage of the approach is different lighting conditions of image inputs. As a solution it is necessary to place signer at a defined background with defined lightning conditions. But this may restrict the system to develop as a portable system [9].

Gesture recognition using acoustic measurements nearby wrist is another significant approach revealed during the review session of the research. In the paper researchers have considered the movements of tendons that contribute to the finger motions. There is a relationship between the geometry structure of wrist and the generated sound due to this changing wrist and finger motions. Since this sound is difficult to hear and identify the difference by the human ear, sensitive microphones were integrated with a wearable wristband band. Background noise is a major obstacle which is difficult to overcome in general for these types of applications [10].

As a different approach there is ongoing research of a radar-based system to track the hand motions. The research involves with Google's Soli sensor and newly designed machine learning architecture suitable to measure radio frequency base motion recognition [11].

In general definition, sign language is made of gestures. These gestures can categorise according to features as illustrated in Fig. 1.5.

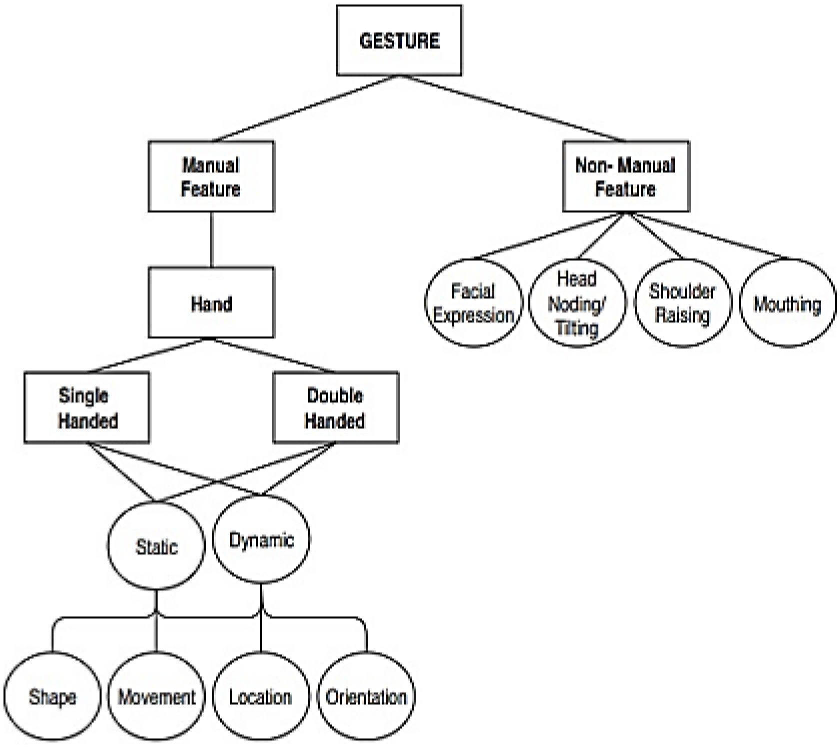


Fig. 1.5 Sign language gesture types. Adapted from [4]

The hand gestures categorise as manual features and other gestures that are non-involved hands categorise as non-manual features. These manual features again separable as dynamic and static featured gestures [4]. But the Sinhala sign language alphabet has both static and dynamic gestures, which only involves with one hand.

1.3.2 Machine learning (ML) methods

Machine Learning (ML) is a subsection of Artificial Intelligence (AI). That is different from formal software applications due to the explicit ability of data analysing and prediction. There are various ML algorithms, but many of them would use existing data as input to predict new output values. Machine learning uses various algorithms according to the data type that is to be analysed. Then build mathematical models and make predictions using historical data or information provided in the data training processes.

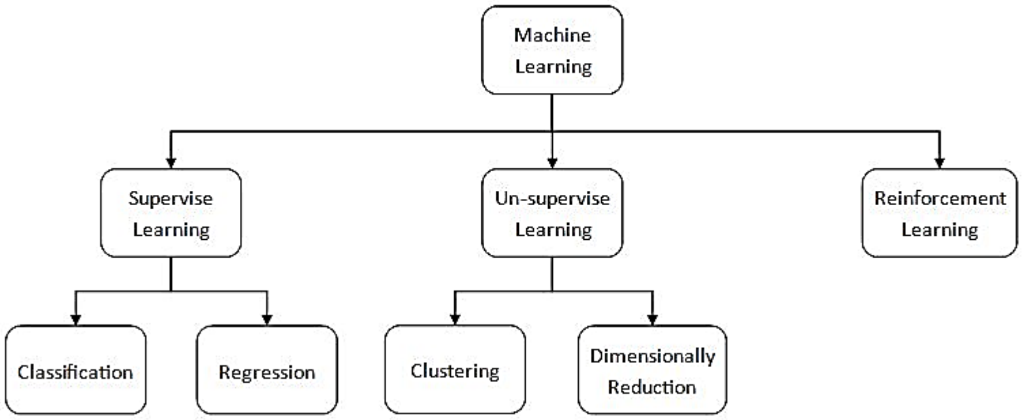


Fig. 1.6 Machine Learning Algorithm categorization

- Supervised ML method

In the Supervised ML method labelled data samples are provided to the machine learning system in order to train and predict the output. The system creates a model using these train-data. Then another data sample called test data used to evaluate the model on its prediction ability. The core process of supervised the ML method is to find relationship between input data and the output data using the labelled data provided and predict with an acceptable success rate. Spam filtering is one of the most common scenarios which uses supervised learning algorithm. This ML method is further categorised into classification and regression.

  - Classification

The ML method used to predict a discrete class label. The purpose may predict a continuous value. But that continuous value is the probability for the class label.

  - Regression

The ML method used to predict continuous quality. The purpose may predict a discreet value in the form of an integer.

- Unsupervised ML method

In the Unsupervised ML method, unlabelled data provide to the system. Simply in which a machine learns or maps inputs with output without any supervision or guidance. The goal of the algorithm is to extract correlation among features or attributes in provided data samples.
This can be further classifieds into two categories as clustering algorithms and reduction algorithms [13]:

  - Clustering

Algorithms find homogeneous subgroups within the data. These subgroups are generally non-overlapping and based on similarities among the data samples. Most probably these clustering algorithms find clusters even when non-exists in this case it is necessary to perform critical assessment on the result and validate using domain knowledge.

  - Dimensionally reduction

The algorithms reduced a number of dimensions and trained the model using a reduced number of variables [14].

- Reinforcement learning

The Reinforcement learning ML method is a feedback-based system that trains the system automatically using reward and penalty against to learning agent. A reward or penalty offering action and according to the feedback system improves the model's performance [13].

1.3.3 Supervised machine learning algorithms

- The k-Nearest Neighbours (KNN) Algorithm

This algorithm classifies the data using similarities between the new data and available data. During the training process training data is stored. When new data or test data feed the ML algorithm classifies data according to the similarity and categorises new data into the most similar class of the train data. The algorithm is used to solve both classification and regression problems.

In this algorithm, it is necessary to select the k value. It simply indicates how many nearest neighbours that going to consider in terms of distance (Euclidian or Manhatton) or weight. Among the considered nearest neighbours whichever turned to the highest category will take the new data point.

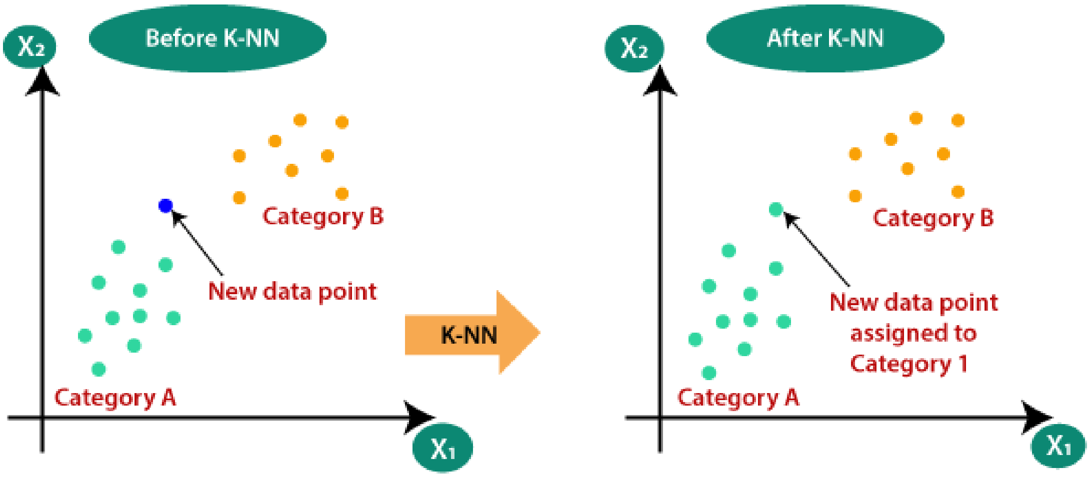


Fig. 1.7 KNN working scenario. Adapted from [10]

The advantages of the KNN Algorithm are simplicity, the robustness to noisy training data. But also, there are a few disadvantages of the KNN algorithm such as determining the number of neighbours to be considered (K) is a critical decision. Other than that, there may be a considerable computational cost with the highest number of data points [10].

- The Random Forest Algorithm

The algorithm can use for both classification and regression problems. The main feature of the algorithm is building decision trees on different samples and taking their majority vote for the classification. To gauge the performance of the model random forest in-built with concepts called ensemble learning. This is a process of combining multiple classifiers to solve a complex problem.

As mentioned early Random Forest classifier does not relay on one decision tree. It builds decision trees on different subsets of data set and takes the average to improve the predictive accuracy of the data set.

With a higher number of trees number of model, accuracy increases and reduces overfitting possibilities. In the first stage, Random Forest combines the decision trees and makes a forest. Then it makes predictions for created trees.

The main advantage of the Random Forest algorithm is data handling capability not only by quantity but also by data with high dimensionality. With a large, dataset Random Forest enhance the accuracy and this help to prevent overfitting issue.

1.4 Research gap and objectives

1.4.1 Problem statement

Sign languages are often ignored when it comes to NLP research. ASL which is the most developed and used sign language. Still ASL has taken for few numbers of NLP research work. Due to complexity and syntax issues, there is very less research work done on Asian sign languages with NLP research work. Therefore, there is no such system for Sinhala sign language (SSL). SSL alphabet contains 56 letters and each letter has bending features, few others have orientation and movement features.

The alphabet of SSL consists only single-handed gestures. These single-handed gestures are a mix of static & dynamic gestures. Basically, SSL letters separable using hand shapes & orientations. The gesture set used in SSL can be shown as a sub set of gestures features as separated in blue colour line of Fig. 1.8.

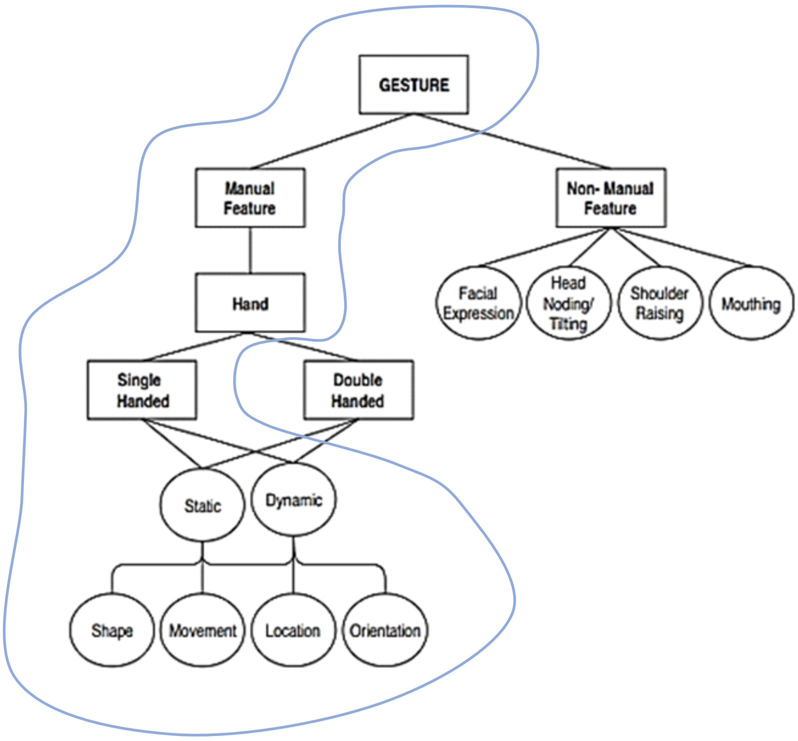


Fig. 1.8 Features of SSL

1.4.2 Objectives

The main objective of this research is to classify hand gestures related to Sinhala sign language (SSL) alphabet with significant accuracy using a smart glove. In addition, identification of words and sentences according to Sinhala sign language rule is also a requirement to complete the research work.

For the purpose following sub objectives were set;

1. Design and implement of data glove
2. Hand gesture classification and recognition
3. Preparation of words and sentences according to Sinhala language rules

2 SYSTEM DESIGN AND IMPLEMENTATION

The system design requirement was to implement a gesture recognition system based on the Sinhala sign language alphabet. As the very first step complete study was carried out on the complete SSL alphabet. And as a result of the study on SSL gestures, it was identified that, the necessity to find means and the method to extract features of hand shape and orientation. The selected suitable method based on the study was the data glove mechanism.

During the planning phase primary inputs, expected outcome and processes to be carried out identified and listed as shown in Fig.2.1

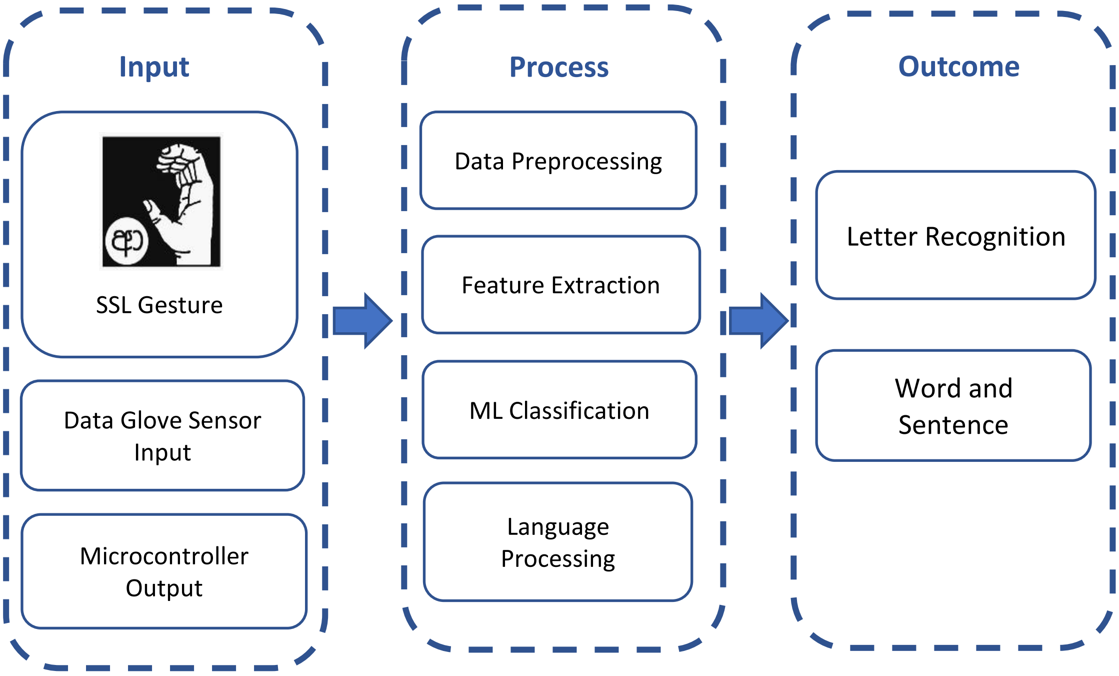


Fig. 2.1 Flow of design plan

The sensors were mounted in suitable positions to obtain data on hand posture. Sensors were interfaced with a printed circuit board (PCB). Data gathering, pre-processing, and applying a suitable machine learning algorithm was performed for the data classification.

An API was created according to Sinhala language rules, for language processing. The real-time data was acquired from the data glove for interpreting and feeding to the Sinhala sign language API. Real-time data prepossessing and classification was proceeded simultaneously.

The complete system comprises sensor base data glove and python scripts for classification algorithm and NLP API. Confirmed system design hardware, tools and methods illustrated as system overview in Fig. 2.2

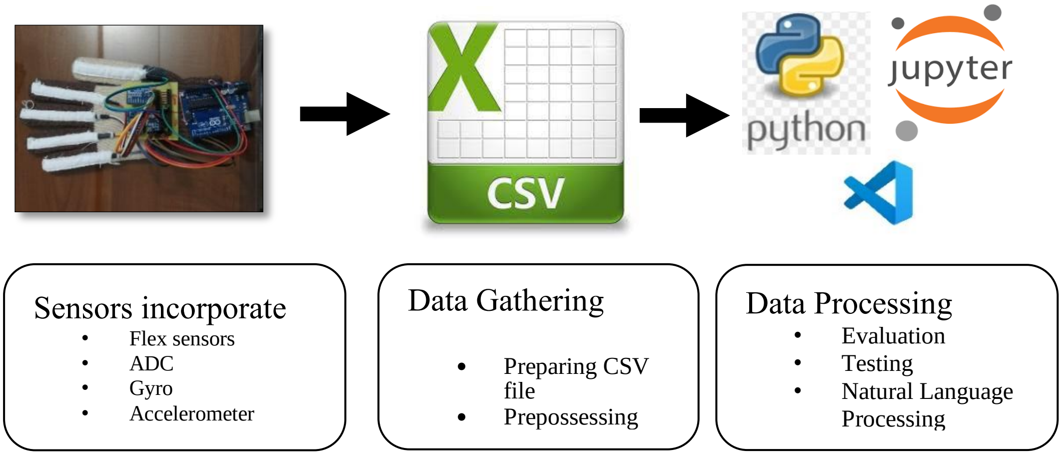


Fig. 2.2 System overview

As mentioned in previous chapter the hardware design consists of electronic sensor and microcontroller base wearable data glove. The incorporated sensors are flex sensors and six-axis IMU. One flex sensor per finger was stitched on the fabric material along with the fingers from the outer side of the data glove. One number of IMU was mounted on top of the outer palm. Flex sensors are analog sensors and all analog data is fetched to microcontrollers through the analog to digital converter (ADC).

All data was gathered in CSV format using excel sheets. Then the data training, testing evaluation, and NLP work were carried out using python scripts on both Jupyter notebook and Visual Studio Code IDEs.

Complete system design methodology is illustrated in the block diagram below in Fig.2.3

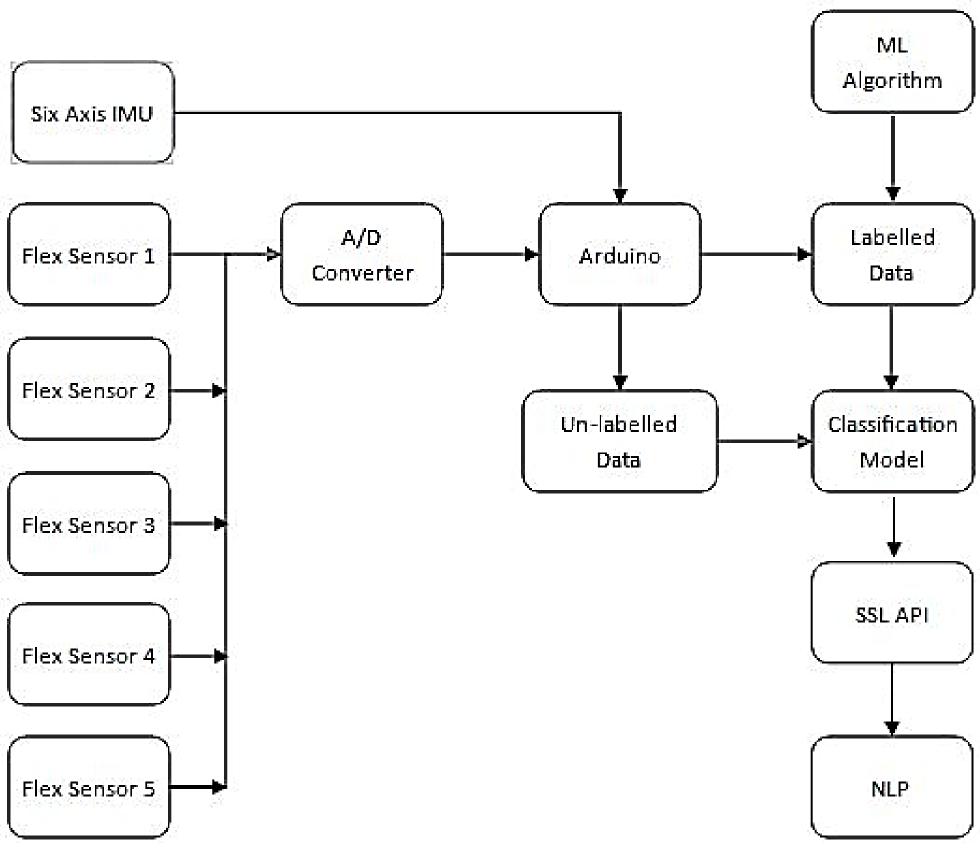


Fig. 2.3 Methodology of complete system design

2.1 Hardware implementation

2.1.1 Electronic system design

SSL alphabet contains 56 letters and each letter has bending features, and few of them have orientation and movement features. Finger bending feature was measured from flex sensors and hand orientation was measured using six axis IMU.

Flex sensor works as a variable resistor which response to finger bending. These analog sensor data is converted to digital data using ADC and prepared the data samples for different hand gestures with regard to each letter of the alphabet.

2.1.2 Bend sensor

The amount of deflection was measured using a flex sensor. There is a carbon surface arranged on a plastic strip and according to deflection sensor's resistance will be changed. This is a two-terminal device, and both of terminals of the flex sensor are not polarized terminals. Flat resistance of flex sensor is 25K ohms and resistance vairy from 45K -to 125K Ohms during deflection.

As the first approach, it was tried to measure the finger bending using conductive stretchable yarns with conductive fabric. But the durability of stretchable material, uncertainty, and calibration issues prevented the choice.

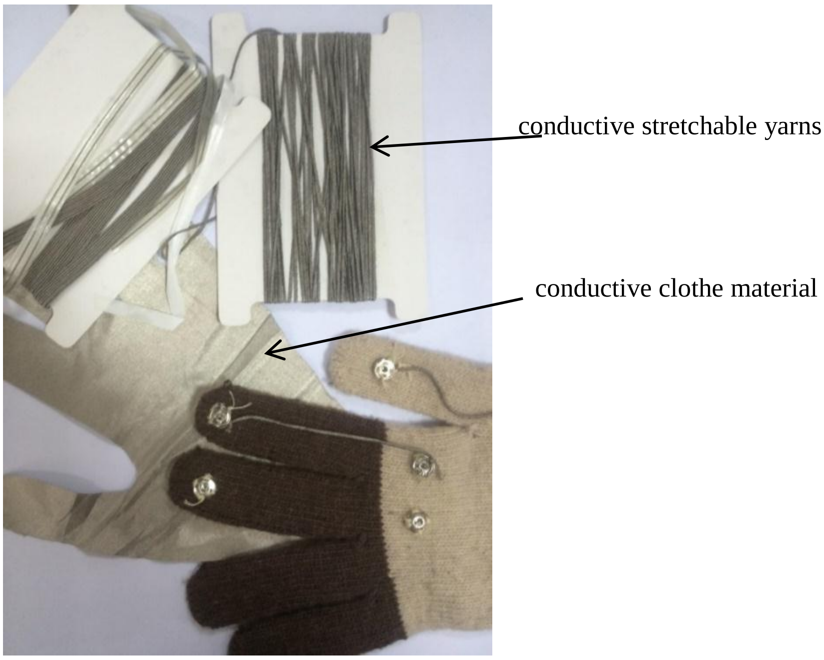


Fig. 2.4. Approach with conductive stretchable yarn

The next approach was to make flex sensor using Velostat and conductive thread. But again, due to uncertainty and calibration issues it was not successful.

Finally, it was concluded with the optimum option, which was to use commercially available flex sensors for this application.

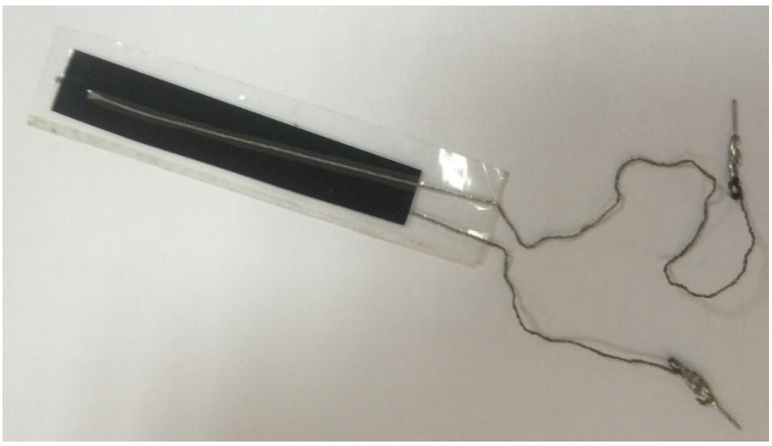


Fig. 2.5 Approach with Velostat and conductive yarn

2.2-Inch flex sensors were selected for the purpose and attached a one flex sensor alone with each finger and completed one voltage divider circuit per finger as shown in Fig.2.6. V_1 voltage variation between the terminal of flex sensor taken in to account through the ADC. Five flex sensor values were taken as five variables for different hand gesture shapes.

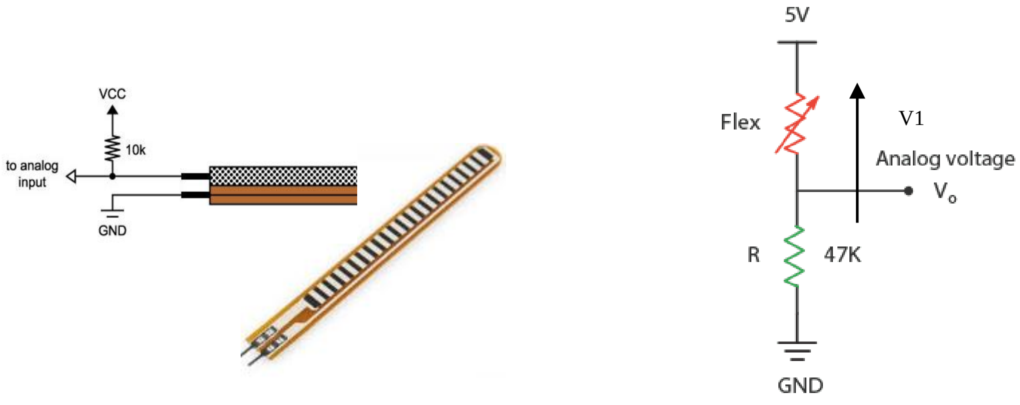


Fig. 2.6 Voltage divider circuit per finger

2.1.3 Analog to Digital Converter (ADC)

Analog-to-digital converter, converts an analog measure into a digital value. If the ADC is a 16-bit ADC, the ADC module has the ability to detect 2^{16} = 65,536 discrete levels. Similarly, there are 8-bit, ADCs and 10-bit ADCs.

ADS1115 is a precision analog-to-digital converter (ADC) module with 16 bits of resolution selected for the purpose. Two number of ADCs used on the circuit. Each one consists of four analog channels.

Fig. 2.7 ADS1115

ADC Calculator:

$$\text{Digital Output} = \frac{2^N \times \text{Analog Input Voltage}}{\text{Reference Voltage}} \qquad (1)$$

$$\text{N} = \text{Number of bits in ADC converter}$$

If operating the voltage is 5V in 10-bit ADC, Then the ratio-metric value of ADC can be simplified as mentioned in the following equation;

$$\frac{\text{Resolution of the ADC}}{\text{System Voltage}} = \frac{\text{ADC reading}}{\text{Analog Voltage measured}} \qquad (2)$$

$$\frac{1023}{5} = \frac{\text{ADC Reading}}{\text{Analog Voltage Measured}}$$

If the analog voltage is 3.12V the output would be 638.

$$\frac{1023}{5.00\text{V}} = \frac{x}{3.12\text{V}}$$

$$\frac{1023}{5.00\text{V}} \times 3.12\text{V} = x$$

$$x = 638$$

2.1.4 Six-axis Inertial Measurement Unit (IMU)

Six-axis IMU sensor module is MPU 6050 was selected for the purpose. This consists with gyroscope and accelerometer. These are used to measure orientation and acceleration. These primary measurements can be used to process velocity, displacement, and any other motion with considerable accuracy.

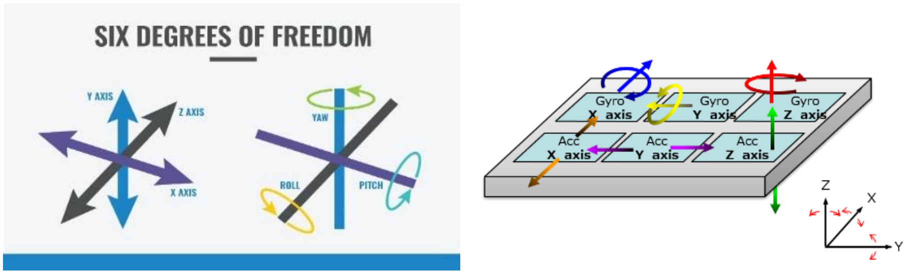


Fig. 2.8 Six-axis IMU. Adapted from [15 and 16]

The afore mentioned gyroscope and accelerometer are combined on the same silicon die. MPU 6050 has the ability to gather a full set of sensor data without intervention from the system processor due to the auxiliary master I^2C bus.

The main disadvantage of a using gyroscope is Gyro drift. The widely used method to overcome the matter is complementary function.

Accelerometer data is reliable only in the long term, as it measures every small force working on the object. This leads to measurement disturbance for this application. Therefore, accelerometer data sending through a low pass filter.

Gyroscope data is reliable in the short term, as it is precise and not susceptible to external forces. Therefore, it is integrated over timestep and feed through a high pass filter.

The accelerometer data through low pass filter and integrated gyroscope data through high pass filter combined with the above-mentioned accelerometer data. The simplest form of the filter is explained below briefly by Fig. 2.9

The function to mathematically calculate angle is;

$$\text{Angle} = 0.98 \times (\text{angle} + \text{gyro Data} \times dt) + 0.02\,(\text{accelerometer Data}) \qquad (3)$$

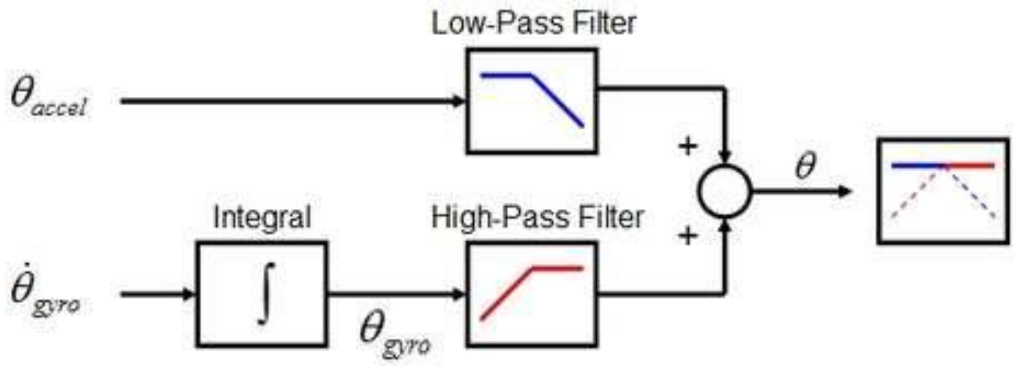


Fig. 2.9 Complementary filter process schematic. Adapted from [17]

The constants 0.98 and 0.02 can vary up to 1 to tune the filter. For this purpose, the complementary filter algorithm needs to work in an infinite loop.

After filtering out gyro drift error using above process, still the IMU data was unstable. This was due to Euler angle singular point (gimble lock).

Gimbal lock is a singularity. This occurs when the rotation around the y-axis approaches 90 or -90 deg. This proves that the orientation representation in Euler angles is not adequate to define complete rotation. Therefore, rotation around x-axis and rotation around the z-axis are not stable values when rotation around the y-axis approaches 90 degrees.

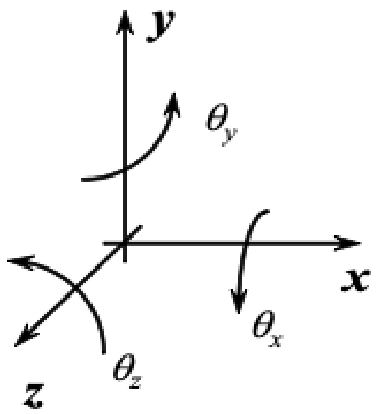


Fig. 2.10 Euler angle transformation

Fig. 2.10 illustrates the model of 3D rotations with Euler angles consists of a sequence of rotations around the Cartesian axis. The rotation sequence pattern starts with the x-axis and ends with the z-axis. Traking this order is important for the analysis of following matrix.

The matrix for a combined rotation around x then y then z is:

$$R(\theta_x,\theta_y,\theta_z)=R_zR_yR_x=\begin{bmatrix} C_yC_z & C_yS_z & -S_y & 0 \\ S_xS_yC_z\text{-}C_xS_z & S_xS_yS_z+C_xC_z & S_xC_y & 0 \\ C_xS_yC_z+S_xS_z & C_xS_yS_z\text{-}S_xC_z & C_xC_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Where S_x=Sinθ_x , C_x=Cosθ_x

The conversion of sine and cosine of θ used accordingly for the *y* and *z* axes.

This is a 3 degrees of freedom(DOF) transformation. The usual argument to present the Gimbal Lock problem is based on the analysis of the matrix of the transformation. If we fix one of the DOF, the remaining transformation should still have 2 DOF. That is not what happens if we choose the y-angle to be 90 degrees. Substituting the cosine and sine value of the y-angle by 0 and 1, respectively, the transformation matrix can be simplified to:

$$R(\theta_x,90°,\theta_z)=\begin{bmatrix} 0 & 0 & -1 & 0 \\ S_xC_z\text{-}C_xS_z & S_xS_z+C_xC_z & 0 & 0 \\ C_xC_z+S_xS_z & C_xS_z\text{-}S_xC_z & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

After simplification with trigonometry, this matrix is equivalent to:

$$R(\theta_x,90°,\theta_z)=\begin{bmatrix} 0 & 0 & -1 & 0 \\ \sin(\theta_x\text{-}\theta_z) & \cos(\theta_x\text{-}\theta_z) & 0 & 0 \\ \cos(\theta_x\text{-}\theta_z) & -\sin(\theta_x\text{-}\theta_z) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

This matrix discloses the fusion of the two remaining DOF into a single variable, one opposing the other [19].

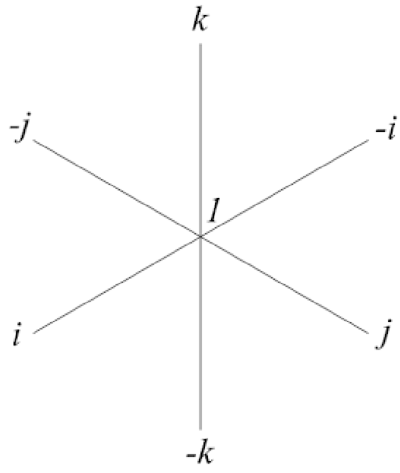


Fig. 2.11 Four- dimensional extension of the complex numbers

This needs to be prevented using a different orientation representation such as quaternions.

Quaternions are an extension of an imaginary number set. It is a 4- dimensional extension of the complex numbers, commonly referred to as a hyper-complex number.

Complex numbers are of the form:

$$a + bi$$

Where a, b $\in \mathbb{R}$ and *i* is the imaginary unit.

Quaternions are, on the other hand, are of the form:

$$a + bi + cj + dk$$

Where *a, b, c, d* $\in \mathbb{R}$ and *i, j,* and *k* are the fundamental quaternion units.

Quaternions are very efficient for analysing situations where rotations in R3 are involved. That they represent a point on the unit sphere. A quaternion is a 4-tuple, which is a more concise representation than a rotation matrix [20].

2.1.5 Circuit design

The circuit design was completed using Egal printed circuit board (PCB) design software. Easily Applicable Graphical Layout Editor (EAGLE) is a user-friendly PCB design software that has a considerably low learning curve. The free version consists of satisfactory features.

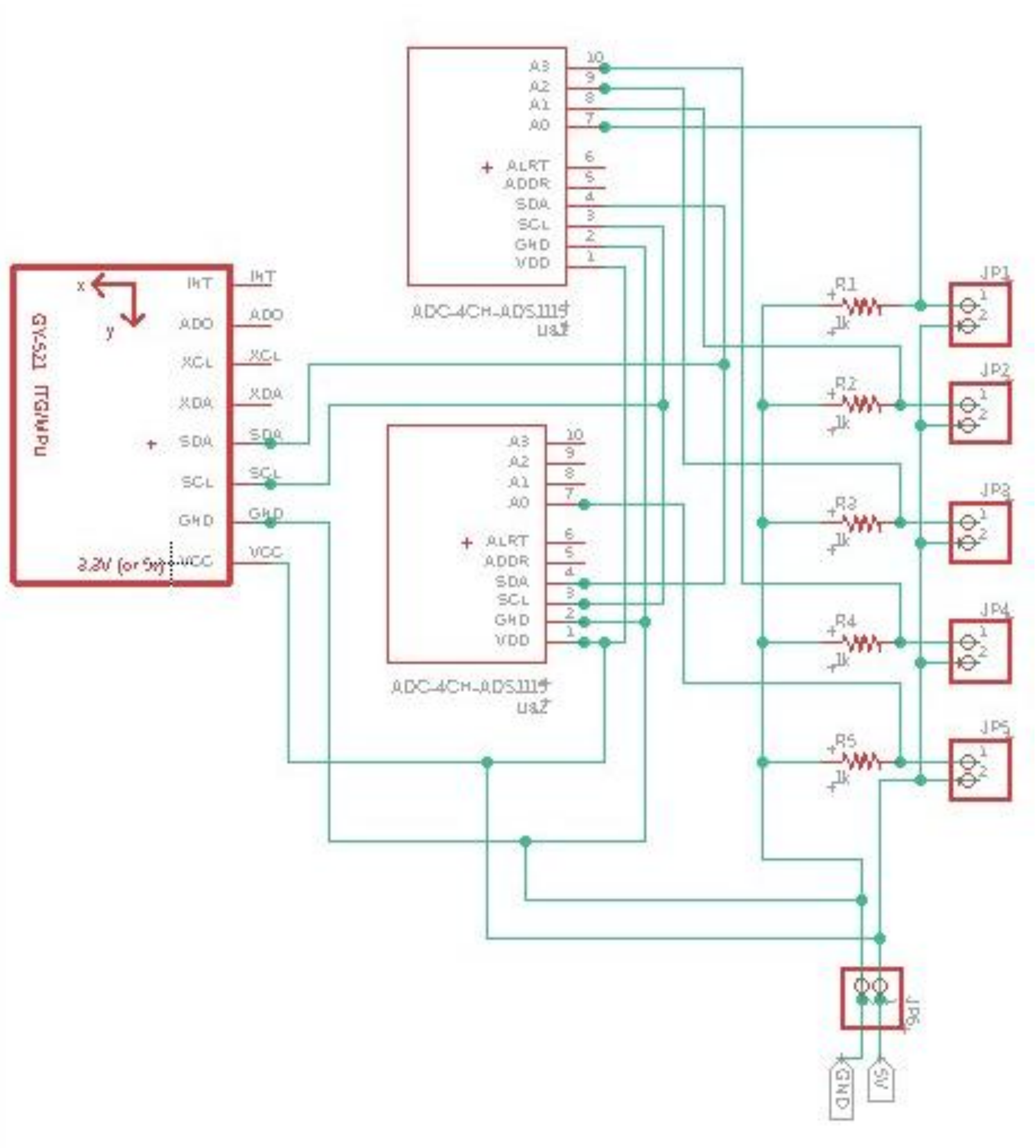


Fig. 2.12 Schematic diagram of the circuit

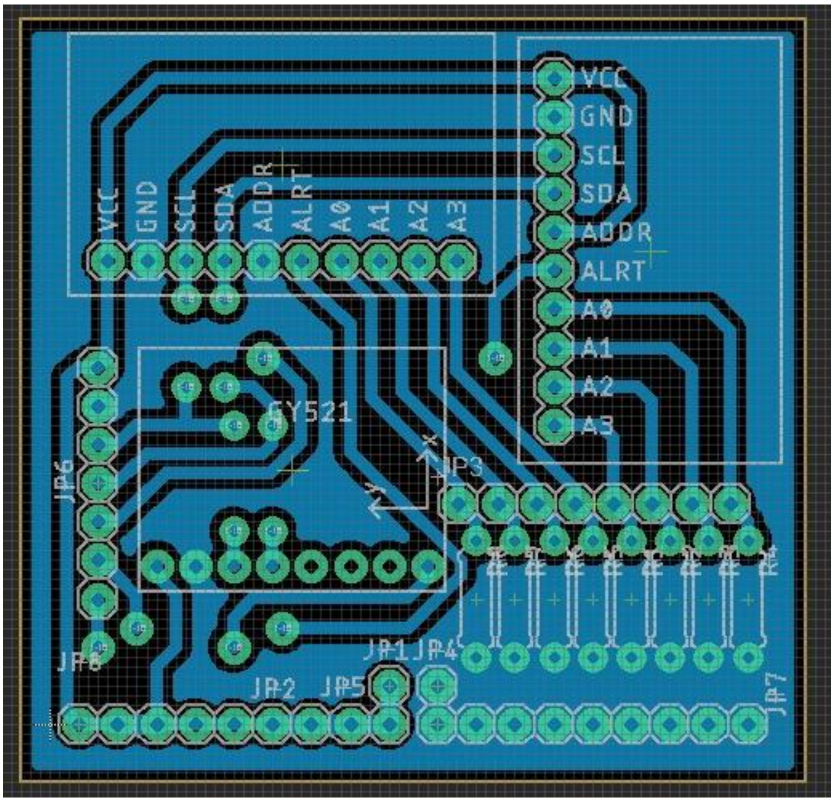


Fig. 2.13 PCB Layout of the circuit

2.1.6 Implementation of data glove

During the design and implementation of data glove, flex sensor positioning was a critical decision. Initially the flex sensor was positioned as shown in Figure 2.14 avoiding the Metacarpophalangeal joints (MCP) joint. The IMU unit was positioned at the middle of the outer palm.

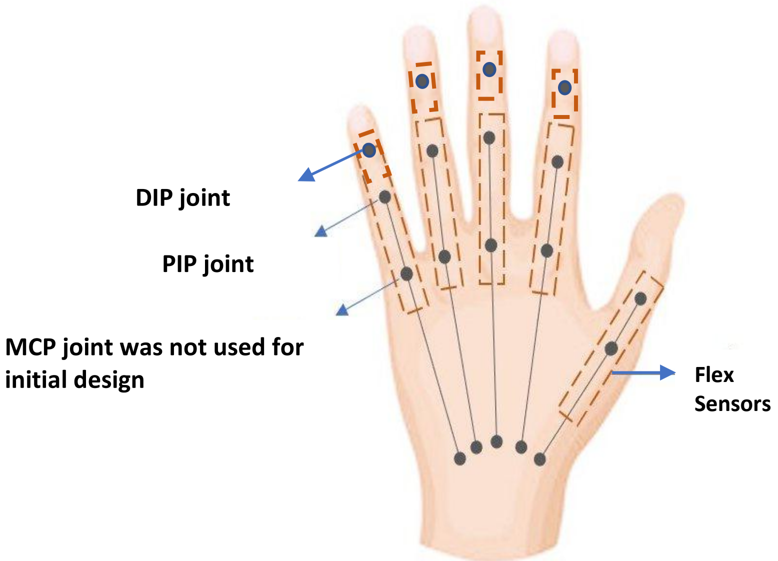


Fig. 2.14 Planned flex sensor arrangement

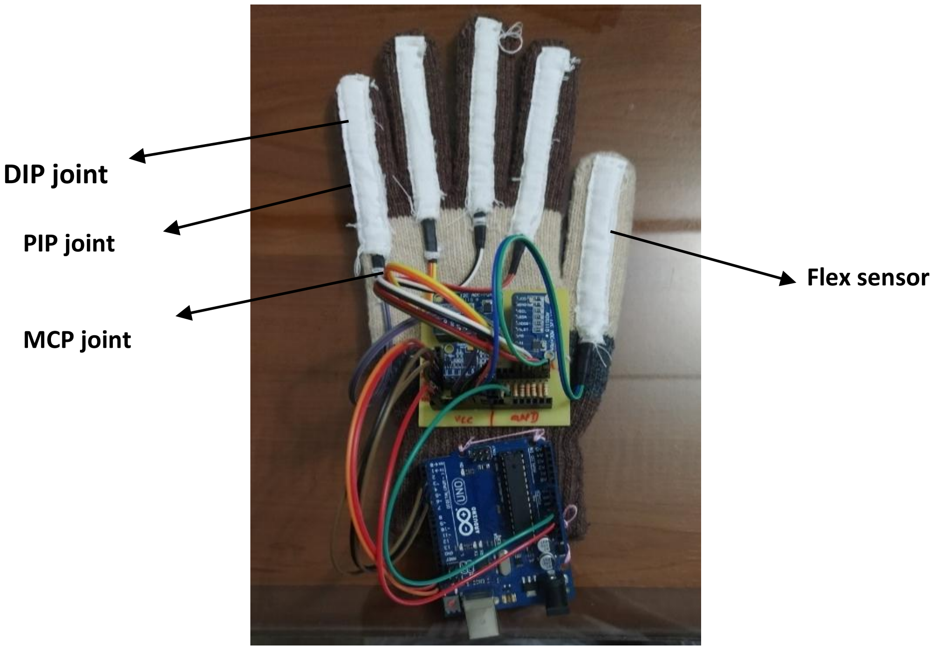


Fig. 2.15 Hand glove with initial sensor positions

After few gesture recognition sessions, it was identified that the bending measurement of Distal interphalangeal (DIP) joint do not make an effect on data. Since the DIP joint is not bending independently, the DIP joint movements can be omitted. The post literature review on hand anatomy revels clear disparity on bending of MCP joint with reference to DIP joint. Therefore, the hardware design revised to read only the bending measurements of MCP joint and PIP joint. The sensors were positioned in a way, where the bending of DIP joint does not make impact to the measurements. This was achieved by dragging down the flex sensors of the fingers other than the thumb finger on the outside of the palm along the same vertical line. The position shown in Fig. 2.16. Since the thumb finger has only two joints which shows clear bending, namely Interphalangeal (IP) joint and MCP joint, therefore the sensor of thumb finger was kept without any position change.

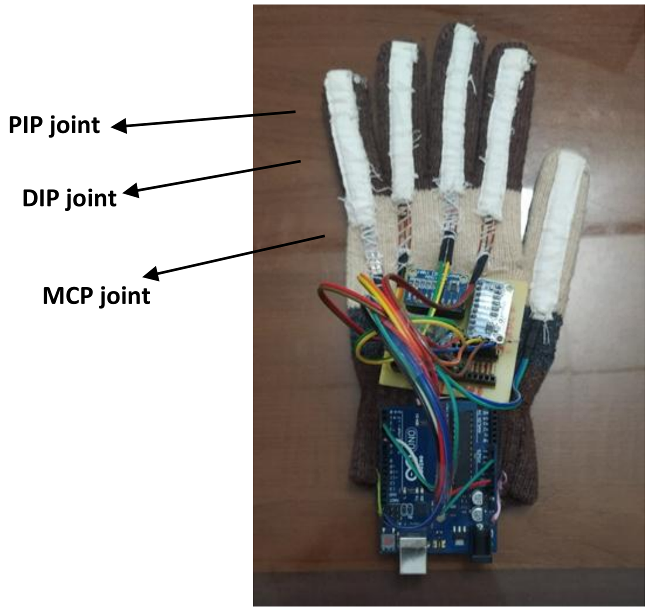


Fig. 2.16 Hand glove with final sensor positions

Few hand postures with designed data glove shown in Fig. 2.17

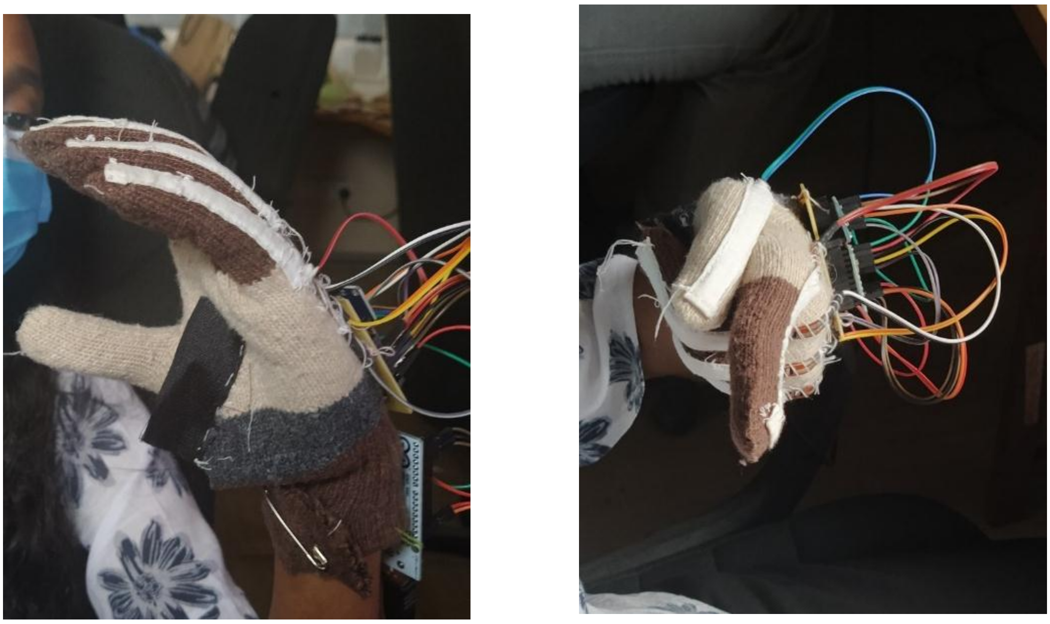


Fig. 2.17 Sample hand postures with the data glove

2.2 Data acquisition

The data samples were collected using volunteers. 100 data samples were collected for each gesture or sign language letter, of SSL alphabet. As there are 56 letters in SSL alphabet 5600 data samples were gathered. The labelled data sets were tabulated in the format shown in Table 2.1 saved as a comma separated value (CSV) file.

Each data sample of the above mentioned 5600 data consist of eleven attributes and one gesture label. Per sample gesture there are 11 attributes (values) collected from the data-glove circuit & values are stores from column 1 to 11 in the Table 2.1 These eleven attributes are made of XYZ values of accelerometer, XYZ values of gyroscope, and five-finger flex sensor values. The gesture label is named as the target group (TG) and included in the 12th column of the CSV file. TG values are varying from 1 to 56 the number represents the respective letter of the SSL alphabet which the data row belongs. Ex: If the TG value is 1 the respective data raw represent one record set out of the 100 samples with the eleven attributes of sensor readings belong to the letter ‘අ’.

Table 2.1 CSV file created using all attributes

Ax	Ay	Az	Gx	Gy	Gz	ADS1-A3	ADS2-A0	ADS2-A1	ADS2-A2	ADS2-A3	TG
10.09	0.35	0.62	0.06	-0.02	-0.03	1547.63	807.19	2815.69	1317.94	2823.94	1
10.12	0.38	0.75	0.05	-0.05	0	1566.56	808.13	2816.06	1317.75	2824.31	1
10.11	0.33	0.58	0.06	-0.05	0.03	1567.88	808.31	2816.62	1319.63	2828.06	1
10.13	0.26	0.6	0.1	-0.02	-0.04	1568.44	808.31	2816.62	1333.31	2829.94	1
10.2	0.28	0.65	0.06	-0.02	-0.01	1555.13	807.94	2813.62	1337.06	2816.81	1
10.16	0.16	0.62	0.04	-0.08	-0.01	1527.38	805.88	2813.25	1317.75	2822.06	1
10.17	0.3	0.64	0.08	-0.01	-0.01	1554.94	804	2815.31	1319.63	2820.19	1
10.16	0.34	0.53	0.05	-0.03	0.01	1557.38	803.63	2815.5	1320.19	2817	1
10.2	0.3	0.79	0.09	-0.03	0	1557.94	802.88	2814.37	1319.44	2814.94	1
10.16	0.18	0.52	0.06	-0.01	-0.02	1557.56	801.75	2812.69	1317.75	2813.62	1
10.1	0.25	0.71	0.07	-0.02	-0.01	1557.56	801.56	2812.12	1317	2812.87	1
10.08	0.39	0.64	0.08	-0.02	-0.02	1561.13	804.75	2821.5	1321.31	2824.5	1
10.15	0.52	0.85	0.1	-0.01	0.03	1610.25	776.81	2658.19	1445.44	2842.5	1
10.07	0.75	0.88	0.08	-0.03	0.03	1613.44	777.19	2662.31	1442.44	2841	1
9.97	0.66	0.78	0.08	-0.04	-0.04	1601.81	777.19	2662.12	1439.25	2839.31	1
10.13	0.57	0.86	0.04	-0.02	-0.01	1600.88	777	2657.25	1437.56	2833.69	1
10.07	0.54	0.84	0.08	-0.02	-0.02	1602	776.63	2657.44	1434.56	2830.69	1
10.06	0.61	0.78	0.07	-0.04	-0.01	1602.38	776.06	2657.25	1432.13	2828.81	1
10.05	0.69	0.82	0.08	-0.01	0.01	1602.38	776.06	2660.06	1428	2829.19	1
10.13	0.82	0.78	0.09	-0.03	-0.02	1593.94	775.88	2673.19	1416	2826.37	1
10.07	0.7	0.7	0.08	-0.02	0	1476.56	776.06	2679.19	1409.81	2805.75	1

2.3 Data analysis

The data analysis methodology is illustrated below in the Fig 2.18:

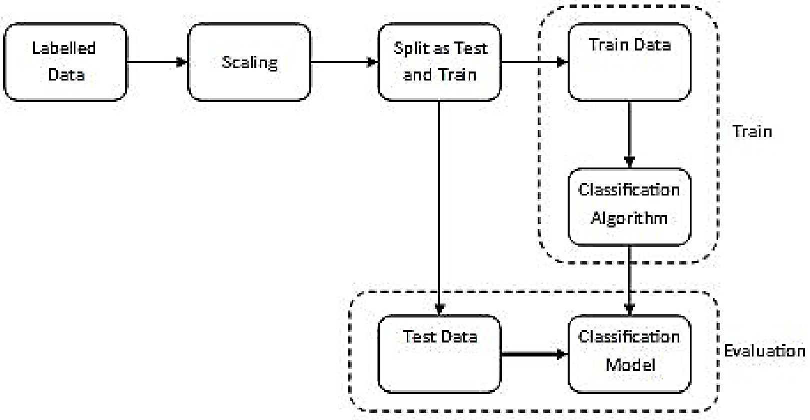


Fig. 2.18 Block diagram of data analysis and evaluation process

First labelled data was scaled with standard scaler and split into train and test sets. 70% of the data was taken in to train and the balance 30% of the data was taken in to test.

In standard scaling, all the features will be transformed in such a way that they will have the properties of standard normal distribution.

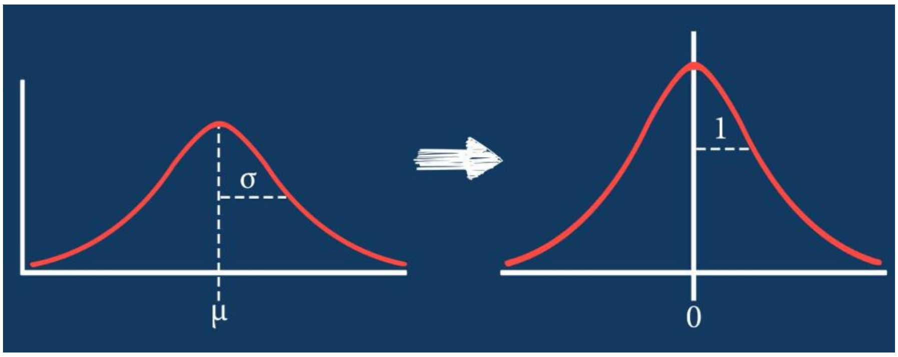


Fig. 2.19 Standard normal distribution- standard scaler. Adapted from [21]

$$z = \frac{x_i - \mu}{\sigma} \tag{7}$$

The train-test split procedure is used to test and estimate the performance of the machine learning model by feeding train data only for the training of the algorithm and use test data only for evaluate and test the training model. This procedure confirms the data used to make predictions on the data set is not used to train the model [16]. The train-test split of data is illustrated in Fig. 2.20

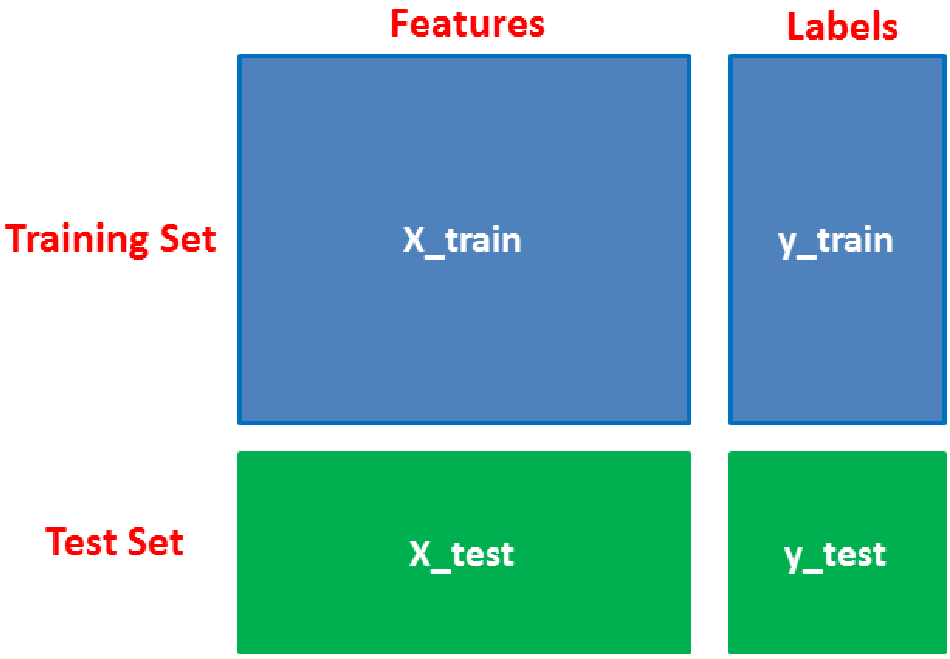


Fig. 2.20 Train - Test Split of data. Adapted from [22]

Since the problem was categorised as supervise classification problem, the supervise learning classification algorithms were applied. Supervise learning classification algorithms applied and evaluated using the above separated test data. The applied ML classification algorithms are K-Nearest Neighbour and Random Forest. It was observed the Random Forest classification algorithm gives better accuracy than KNN classification algorithm. Based on this observation the classification was proceeded with the Random Forest classification algorithm.

After classification letters were recognised, as the next step words and sentences were prepared using newly developed SSL API. Classification performed using Python script in Jupyter note book. Then the whole script was developed in Visual Studio Code integrated development environment (IDE). Python is an object-oriented high-level, programming language. Both the Jupyter notebook and Visual Studio Code can be defined as code editors, as well as IDEs.

2.4 Sinhala sign language API and alphabetic logic

Sinhala alphabet consists of vowels and consonants only. Modifiers are made from combinations of consonants and vowels. These different modifiers stored in the lookup table and fetched whenever necessary.

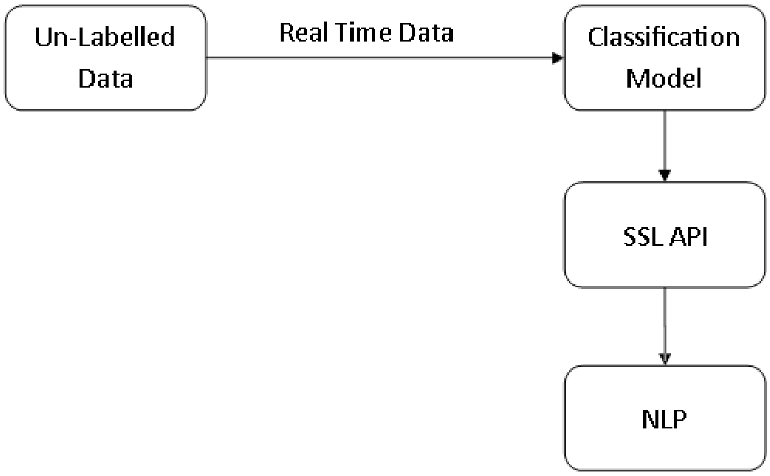


Fig. 2.21 Block diagram of API and NLP process

The lookup table 2.2 illustrates the moderators created by the combinations of consonants and vowels.

Table 2.2 Lookup table to create moderators

		Vowels											
		අ	ආ	ඇ	ඈ	ඉ	ඊ	උ	ඌ	එ	ඒ	ඔ	ඕ
Consonant	ක්	ක	කා	කැ	කෑ	කි	කී	කු	කූ	කෙ	කේ	කො	කෝ
	ග්	ග	ගා	ගැ	ගෑ	ගි	ගී	ගු	ගූ	ගෙ	ගේ	ගො	ගෝ
	ජ්	ජ	ජා	ජැ	ජෑ	ජි	ජී	ජු	ජූ	ජෙ	ජේ	ජො	ජෝ
	ට්	ට	ටා	ටැ	ටෑ	ටි	ටී	ටු	ටූ	ටෙ	ටේ	ටො	ටෝ
	ද්	ද	දා	දැ	දෑ	දි	දී	දු	දූ	දෙ	දේ	දො	දෝ
	ත්	ත	තා	තැ	තෑ	ති	තී	තු	තූ	තෙ	තේ	තො	තෝ
	ඩ්	ඩ	ඩා	ඩැ	ඩෑ	ඩි	ඩී	ඩු	ඩූ	ඩෙ	ඩේ	ඩො	ඩෝ
	න්	න	නා	නැ	නෑ	නි	නී	නු	නූ	නෙ	නේ	නො	නෝ
	ප්	ප	පා	පැ	පෑ	පි	පී	පු	පූ	පෙ	පේ	පො	පෝ
	බ්	බ	බා	බැ	බෑ	බි	බී	බු	බූ	බෙ	බේ	බො	බෝ
	ම්	ම	මා	මැ	මෑ	මි	මී	මු	මූ	මෙ	මේ	මො	මෝ
	ය්	ය	යා	යැ	යෑ	යි	යී	යු	යූ	යෙ	යේ	යො	යෝ
	ර්	ර	රා	රැ	රෑ	රි	රී	රු	රූ	රෙ	රේ	රො	රෝ
	ල්	ල	ලා	ලැ	ලෑ	ලි	ලී	ලු	ලූ	ලෙ	ලේ	ලො	ලෝ
	ව්	ව	වා	වැ	වෑ	වි	වී	වු	වූ	වෙ	වේ	වො	වෝ
	ස්	ස	සා	සැ	සෑ	සි	සී	සු	සූ	සෙ	සේ	සො	සෝ
	හ්	හ	හා	හැ	හෑ	හි	හී	හු	හූ	හෙ	හේ	හො	හෝ

moderators

Elaborating in technical terms, like any other language, in SSL consonents, vowels or moderators (combination of consonents and vowels) represents letters and combination of letters (one or more) creates word. To explain this further creation of Sinhala word “අම්මා” is considered.

The alphabetic logic for the Sinhala word “අම්මා” is illustrated below Fig.2.22;

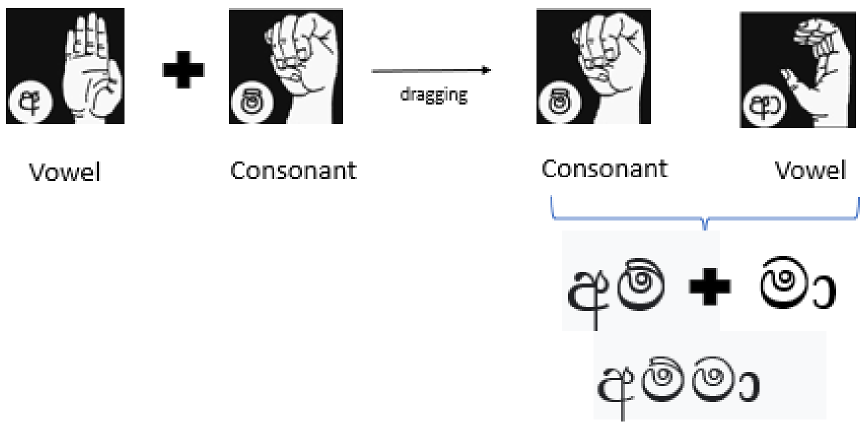


Fig. 2.22 Vowel and consonant conjunction

During the word identification process, there are quick transition among the letters. This make intermediate postures, which were taken into a separate target class called background class and ignored at the language API. Otherwise, these intermediate posture data will proceed through the classification with low predication accuracy.

The figure 2.23 illustrates the API data process flow chart.

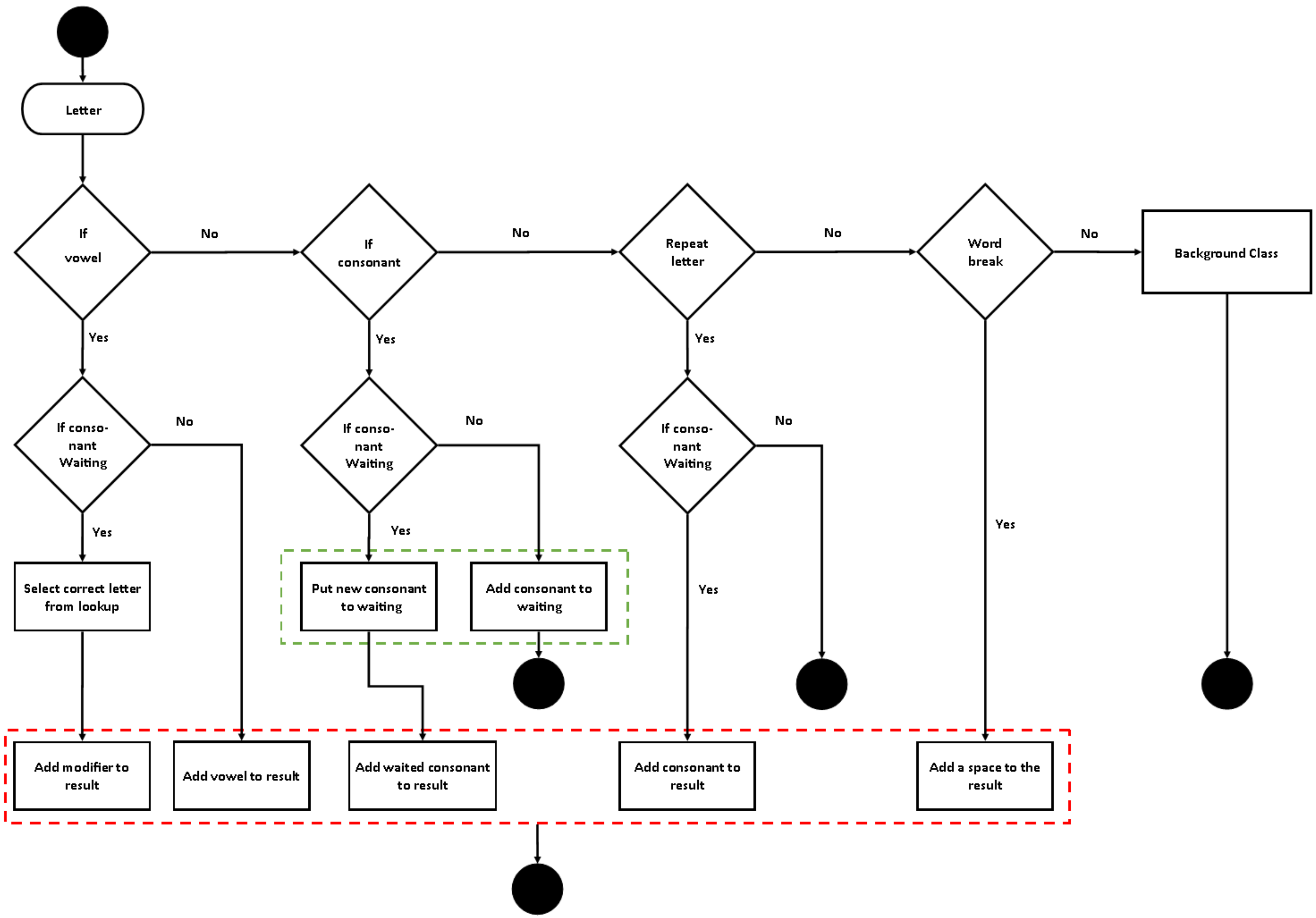


Fig. 2.23 API data flow chart

3 THE EXPERIMENT AND RESULTS

After the data set was prepared in CSV format as shown in Table 2.1, data training, testing evaluation was carried out. After obtain the classification reports, experiments carryout on letter identification, language processing using newly developed SSL API.

3.1 Classification results

After data gathering and prepossessing KNN and Random Forest classification algorithms were performed. Both ML models evaluated using classification reports and selected Random Forest classification algorithm due to high accuracy.

3.1.1 Classification reports

Classification reports consist of parameters which were used to define the accuracy and evaluate the ML model, which are precision, recall and F1 score for each class. Class column indicate 56 letters of the complete Sinhala sign language alphabet and support column indicate number of occurrences for each class. Further definitions on these parameters are as follow;

Precision – How many are correctly classified among that class
Recall – How many of the class find over the whole number of elements of that class
F1 score – The harmonic mean, in other word it is a type of numerical average
Support – Number of occurrences of the given class in given data set

$$\text{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

$$\text{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{Total}}$$

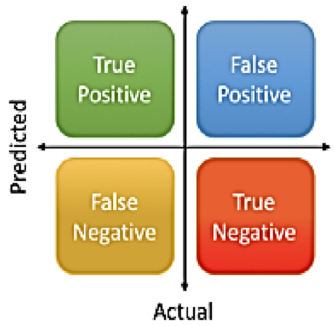


Fig. 3.1 Outcome of ML algorithm

True Positive (TP) is perfectly-identified correct letters**.**
True Negative (TN) is perfectly-identified but incorrect letters.
False Positive (FP) wrongly detected letters as positive which is actually not the correct letter.
False Negative (FN) is wrongly detected incorrect letters which is actually correct.

Classification report for complete alphabet of SLL with regard to KNN Classifier is as follow;

Table 3.1 Classification report for KNN algorithm

Class	Precision	Recall	F1-score	Support
1	1.00	1.00	1.00	36
2	0.97	1.00	0.99	36
3	0.95	0.97	0.96	37
4	1.00	0.83	0.91	30
5	1.00	1.00	1.00	25
6	0.87	0.79	0.83	34
7	1.00	1.00	1.00	34
8	1.00	1.00	1.00	30
9	0.94	1.00	0.97	30
10	1.00	1.00	1.00	37
11	0.94	0.94	0.94	32
12	1.00	0.71	0.83	35
13	0.97	0.94	0.96	36
14	0.87	0.93	0.90	28
15	1.00	1.00	1.00	31
16	1.00	1.00	1.00	29
17	1.00	1.00	1.00	30
18	1.00	1.00	1.00	29
19	1.00	1.00	1.00	21
20	0.94	1.00	0.97	32
21	0.94	1.00	0.97	31
22	1.00	1.00	1.00	30
23	1.00	1.00	1.00	35
24	0.96	1.00	0.98	22
25	1.00	1.00	1.00	32
26	0.94	1.00	0.97	32
27	0.93	1.00	0.96	25
28	0.86	1.00	0.93	31
29	0.90	1.00	0.95	36
30	0.93	1.00	0.96	26
31	0.83	0.45	0.59	33
32	0.83	1.00	0.91	25
33	0.93	1.00	0.96	26
34	0.79	1.00	0.88	23
35	1.00	0.92	0.96	25
36	1.00	0.27	0.43	33
37	1.00	0.95	0.97	20
38	0.88	1.00	0.94	29
39	1.00	0.93	0.97	30
40	1.00	1.00	1.00	33
41	1.00	1.00	1.00	27
42	1.00	1.00	1.00	29
43	0.85	1.00	0.92	33
44	0.97	1.00	0.98	29
45	1.00	1.00	1.00	24
46	0.88	1.00	0.94	29
47	1.00	1.00	1.00	40
48	0.96	0.93	0.94	27
49	1.00	1.00	1.00	35
50	0.93	1.00	0.96	27
51	0.93	1.00	0.96	26
52	0.92	1.00	0.96	24
53	0.97	0.92	0.95	39
54	0.68	0.87	0.76	30
55	0.95	0.88	0.91	24
56	1.00	1.00	1.00	28
accuracy			0.95	1680

Classification report for complete alphabet of SLL with regard to Random Forest classifier is as follow;

Table 3.2 Classification report for Random Forest algorithm

Class	Precision	Recall	F1-score	Support
1	1.00	1.00	1.00	31
2	1.00	1.00	1.00	28
3	0.90	1.00	0.95	28
4	0.94	0.86	0.90	35
5	1.00	1.00	1.00	24
6	0.88	1.00	0.93	28
7	1.00	1.00	1.00	25
8	1.00	1.00	1.00	30
9	1.00	1.00	1.00	42
10	1.00	0.96	0.98	27
11	1.00	1.00	1.00	25
12	1.00	1.00	1.00	31
13	1.00	1.00	1.00	24
14	0.96	1.00	0.98	27
15	1.00	1.00	1.00	32
16	1.00	1.00	1.00	32
17	1.00	1.00	1.00	37
18	1.00	1.00	1.00	30
19	1.00	1.00	1.00	29
20	1.00	1.00	1.00	44
21	1.00	1.00	1.00	35
22	1.00	1.00	1.00	38
23	1.00	1.00	1.00	26
24	1.00	1.00	1.00	33
25	0.97	0.97	0.97	33
26	1.00	1.00	1.00	28
27	1.00	1.00	1.00	37
28	0.97	1.00	0.99	33
29	1.00	1.00	1.00	27
30	1.00	1.00	1.00	23
31	0.97	0.97	0.97	34
32	1.00	0.97	0.98	30
33	0.93	1.00	0.96	27
34	1.00	1.00	1.00	32
35	1.00	1.00	1.00	31
36	1.00	0.83	0.91	35
37	1.00	1.00	1.00	38
38	1.00	0.96	0.98	27
39	1.00	1.00	1.00	26
40	1.00	1.00	1.00	27
41	1.00	1.00	1.00	31
42	1.00	1.00	1.00	27
43	1.00	1.00	1.00	31
44	1.00	1.00	1.00	32
45	0.96	0.96	0.96	27
46	1.00	1.00	1.00	20
47	1.00	1.00	1.00	34
48	1.00	1.00	1.00	27
49	1.00	1.00	1.00	24
50	1.00	1.00	1.00	18
51	1.00	1.00	1.00	32
52	1.00	1.00	1.00	28
53	0.96	0.96	0.96	27
54	0.96	0.96	0.96	28
55	1.00	1.00	1.00	28
56	0.97	1.00	0.99	37
accuracy			0.99	1680

3.2 Language processing results

The language processing experiment was carried out using Random Forest classification (due to high accuracy) with the newly developed SSL API mentioned in chapter 2. The real-time data fed via a serial port as shown in the Fig.3.2.

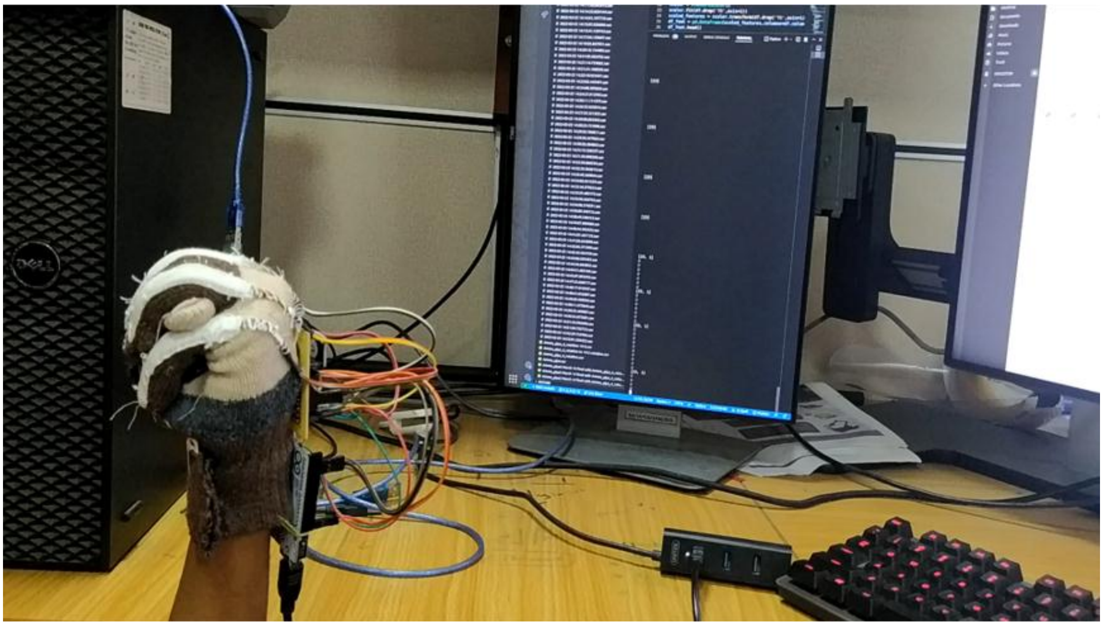


Fig. 3.2 Real-time data feeding via USB

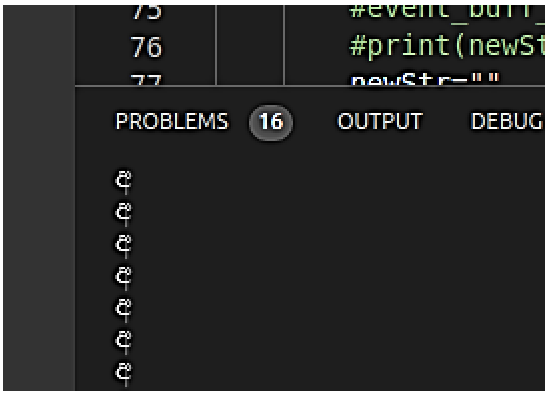
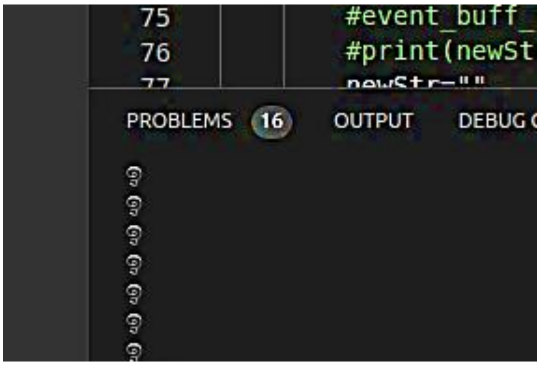
3.2.1 Preparation of letters

The shape of the hand and orientation positioned related to particular letter in SSL alphabet is immediately display as shown in Table 3.3

Table 3.3 illustrates experiment and resulted vowels ‘ඇ’ and ‘ඉ’, which can show by using one hand gesture.

But when it comes to moderators two gestures conjugate and the related letter fetch from lookup table mentioned in chapter 2, Table 2.2. The alphabetic logic described in chapter 2, Fig. 2.22.

Table 3.3 Experiment and results on letter recognition - Vowels

Letter	Gesture	Result
‘ඇ’		
‘ඉ’		

For an example let’s take the letter ‘ය’. The letter ‘ය’ is a moderator, which result from the conjunction of vowel ‘ඇ’ and consonant ‘ය්’. The result is displayed with moderate letter as in Table 3.4.

Table 3.4 Experiment and results on letter recognition - moderators

Letter	Gestures	Result
‘ය’	 	

To futher elaborate the work, Fig. 3.3 illustrate few random SSL alphabet letters fed to the system.

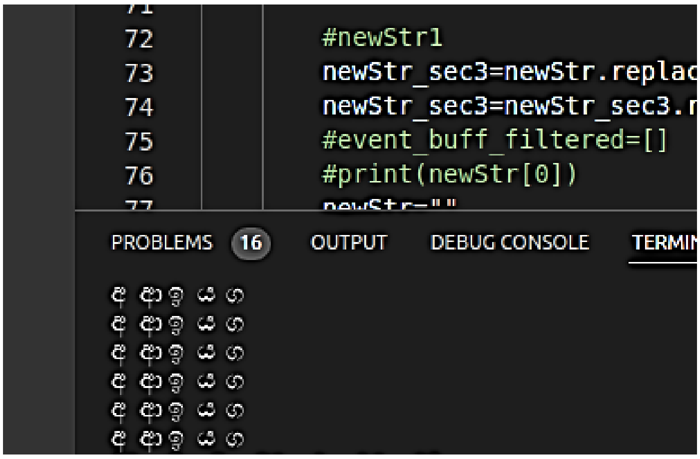


Fig. 3.3 Random letters

3.2.2 Preparation of words

The word-making process proceeds with adding and conjunction of vowels and consonants. The complete API flow chart developed for SSL alphabet is shown in Fig. 2.23.

During the transitions between letters, the intermediate postures are taken into a separate class called background class and rejected during the process. Therefore, the additional class call background class is used.

To elaborate the word preparation process let's consider the word "ගියා". The letter 'ගි' is a moderator created by consonant 'ග්' and vowel 'ඉ'. As same as moderator 'යා' created by consonant 'ය්' and vowel 'ආ'. This is illustrated in Fig. 3.4

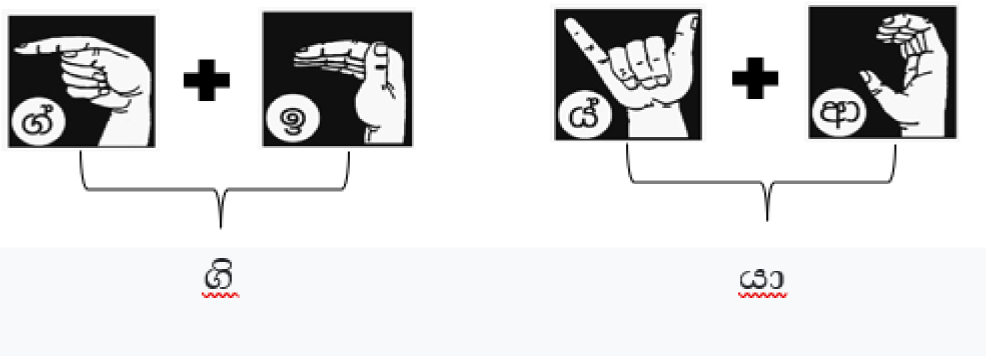


Fig. 3.4 Word preparation

During the transition from posture related to letter 'ග්' to posture related to letter 'ඉ' there are few rotations and hand shapes changes occurs. These postures have minor probability in classification. To make further discrimination we make separate class for these intermediate postures and remove these unnecessary data as background class. The result will appear as in Fig. 3.5

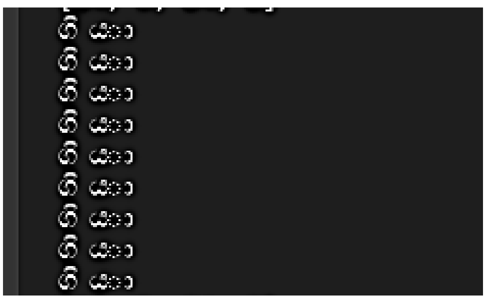
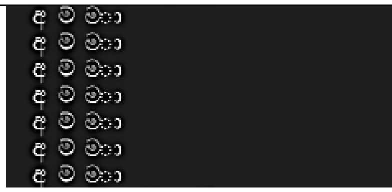
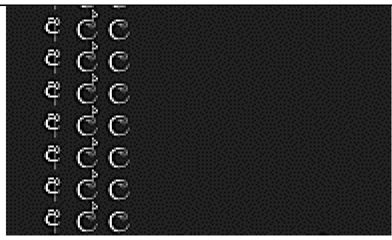
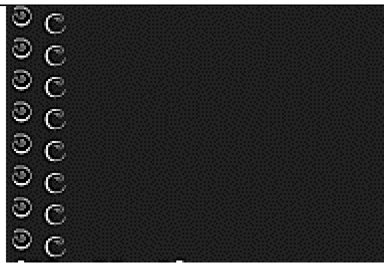
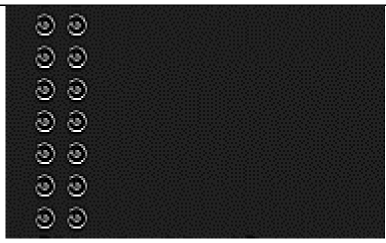
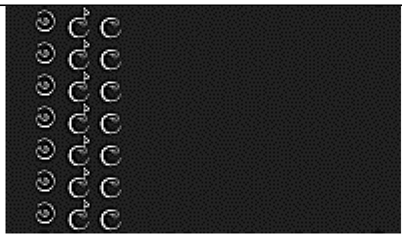
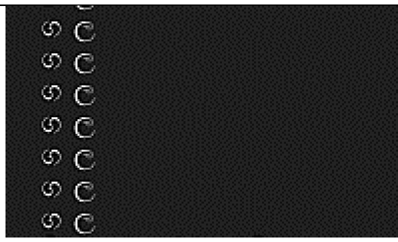


Fig. 3.5 Word "ගියා"

To further test and verify the designed API algorithm random words were fed to the program via data glove and the outputs were tested. Using the same methodology there is no limitation of preparation of any word in Sinhala language. Table 3.5 illustrate the output.

Table 3.5 Experiment and results on language processing – Words

	Word	Result
1	අම්මා	
2	ආවා	
3	ගම	
4	අල්ල	
5	මල	

6	මම	
7	අල	
8	අම්ා	
9	මල්ල	
10	ගල	

3.2.3 Preparation of sentence

Any sentence can display by adding words. If natural language processing was performed continuously during word preparation, making a sentence is just adding of words. To separate words space between word is added as separate gesture class call hand down posture. The sentences preparation experiment and result shown in Fig. 3.6, Fig. 3.7 and Fig. 3.8. Any sentence in Sinhala language can prepare using the proposed system.

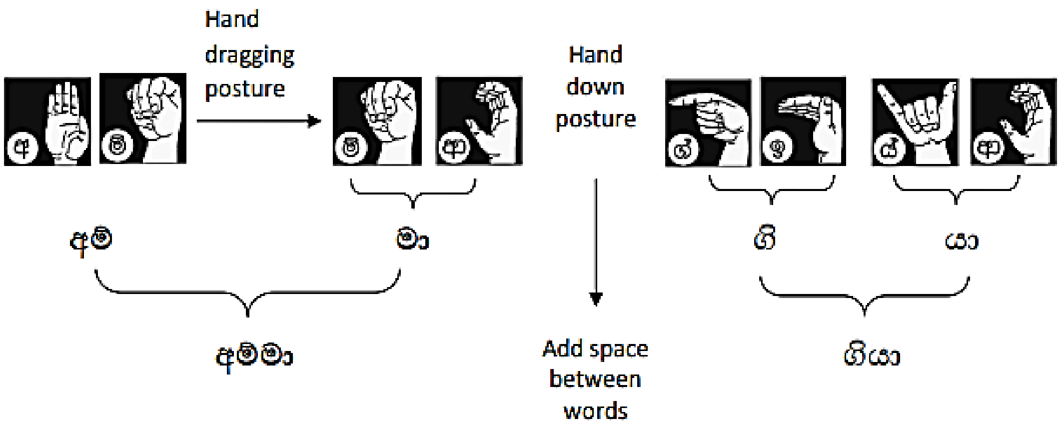


Fig. 3.6 Sentence preparation experiment

Fig. 3.7 Resulted sentence

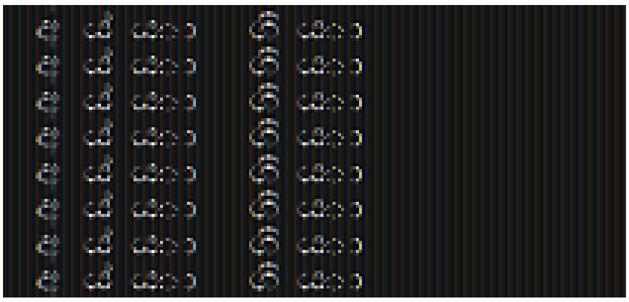


Fig. 3.8 Sentence preparation

4 CONCLUSIONS AND FUTURE WORK

4.1 Conclusions

Smart glove was implemented with five flex sensors and IMU. Which provide eleven number of independent variables per data sample. System designed for recognition of SSL alphabet.

Incorporating ML classification methods enabled to optimize electronic systems and increase their decision-making ability. In reference to the two classification algorithms used, Random Forest classification algorithm was the most adequate classifier for this application with 99% accuracy. Classification was completed for all letters of Sinhala sign language alphabet. An API was designed to identified the fed words and sentences according to the rules of the Sinhala language. Then the research moved towards the NLP research area due to some approaches which had to be taken to overcome the problems occurred during the research.

During the process of word preparation new class was created and added as background class. This is added externally to avoid misclassifying of intermediate transitions between letters. The class consist of gestures which occurs during transition between letters. When the data was trained the smart glove for more words, more and more letters were combined. With this continuous process more transition gestures need to add background class as there are a large number of words.

During the hardware design, flex sensor positions and IMU mount position was re-adjusted as shown in Fig.2.16. This change was done after halfway through the design. After the data was gathered it was identified that the DIP joint does not make significant impact to the data. That reason tempts us for carryout further study on hand anatomy with related to gestures of SSL alphabet. As a result, DIP joints of the fingers were ignored and PIP joint and MCP joint bend measured using flex sensor.

Finally, there is no limitation for convert SSL into words and sentences with further training. Any word and sentences can be created after proper training of data and identification of background class details.

4.2 Future work

Currently this research bridges SSL into a readable format which is common for both normal and speech and hearing-impaired community. But in addition, this can incorporate with text to a speech file and can make audio output to the normal community.

Complete training for all possible combination of letters and preparation of words is necessary to continue NLP. The data acquisition scenario can improve after a deep analysis of Sinhala language rules and letter arrangement. That would reduce the background class postures which need to recognise and eliminate during NLP.

In, addition instead of commercially available flex sensors we can follow the same research with stretchable conductive fabric materials. But before a headed it is recommended to carry out a complete analysis of fabric material properties and the variations with environmental conditions and time.

Also, the system can be customised as an education platform for sign language learners. With an additional display unit and database with pictures related to words, this system can improve as a student learning platform for both remote learners and self-learners. As further note though the system has been designed for fingerspelling, it can use for two-hand gesture signs with two wearable gloves.

Further, the developed smart glove can be use separately for any other hand gesture base applications and sign languages. The developed ML base system can use or customize separately for feature extraction of any smart wearable item, and finally, the newly developed Sinhala API can use separately for any Sinhala language base NLP research work.

REFERENCES

[1] *Deafness and hearing loss*. (2021, April 1). World Health Organization. Retrieved March1,2022, from https://www.who.int/en/news-room/fact-sheets/detail/deafness-and-hearing-loss

[2] Xia, P., & Du, Z. (2019, April). Research on Intelligent Recognition of Intelligent Gloves based on Acceleration Sensor. In *3rd International Conference on Mechatronics Engineering and Information Technology (ICMEIT 2019)* (pp. 550-555). Atlantis Press.

[3] Traynor, R. (2017b, June 9). *International Sign Languages*. Hearing International. Retrieved January 1, 2019, from https://hearinghealthmatters.org/hearinginternational/2017/international-sign-languages/

[4] Farooq, U., Asmat, A., Rahim, M. S. B. M., Khan, N. S., & Abid, A. (2019, November). A comparison of hardware based approaches for sign language gesture recognition systems. In *2019 International Conference on Innovative Computing (ICIC)* (pp. 1-6). IEEE.

[5] Yin, K., Moryossef, A., Hochgesang, J., Goldberg, Y., & Alikhani, M. (2021). Including signed languages in natural language processing. *arXiv preprint arXiv:2105.05222*.

[6] Su, H., Ovur, S. E., Zhou, X., Qi, W., Ferrigno, G., & De Momi, E. (2020). Depth vision guided hand gesture recognition using electromyographic signals. *Advanced Robotics*, *34*(15), 985-997.

[7] S. (2021, July 5). *What is EMG sensor, Myoware and How to use with Arduino?* Latest Open Tech From Seeed. Retrieved August 14, 2021, from https://www.seeedstudio.com/blog/2019/12/29/what-is-emg-sensor-myoware-and-how-to-use-with-arduino/

[8] Benalcázar, M. E., Motoche, C., Zea, J. A., Jaramillo, A. G., Anchundia, C. E., Zambrano, P., ... & Pérez, M. (2017, October). Real-time hand gesture recognition using the Myo armband and muscle activity detection. In *2017 IEEE second Ecuador technical chapters meeting (ETCM)* (pp. 1-6). IEEE.

[9] Kudrinko, K., Flavin, E., Zhu, X., & Li, Q. (2020). Wearable sensor-based sign language recognition: A comprehensive review. *IEEE Reviews in Biomedical Engineering*, *14*, 82-97.

[10] Siddiqui, N., & Chan, R. H. (2017, July). A wearable hand gesture recognition device based on acoustic measurements at wrist. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 4443-4446). IEEE.

[11] Wang, S., Song, J., Lien, J., Poupyrev, I., & Hilliges, O. (2016, October). Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (pp. 851-860).

[12] Pan, J., Li, Y., Luo, Y., Zhang, X., Wang, X., Wong, D. L. T., ... & Thean, A. V. Y. (2021). Hybrid-Flexible Bimodal Sensing Wearable Glove System for Complex Hand Gesture Recognition. *ACS sensors*, *6*(11), 4156-4166.

[13] S.J. (n.d.). *Machine Learning Tutorial | Machine Learning with Python - Javatpoint*. Www.Javatpoint.Com. Retrieved March 3, 2022, from https://www.javatpoint.com/machine-learning

[14] Gatto, L. (2021b, December 6). *Chapter 8 Unsupervised learning: dimensionality reduction | Bioinformatics*. Bioinformatics. Retrieved March 3, 2022, from https://uclouvain-cbio.github.io/WSBIM1322/sec-dimred.html

[15] *Six -axis IMU*. (n.d.). [Illustration]. Panasonic Industry. https://www.google.com/search?q=six+axis+IMU&sxsrf=ALiCzsZ4nDuXR_h3gOaWB8Db-wneSh8hjA:1651170727650&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjZor2lsrf3AhVmILcAHZFKBBAQ_AUoAXoECAEQAw&biw=1366&bih=657&dpr=1#imgrc=FcEIXAcpyvbRQM

[16] Pao, C. (2019b, September 13). *The importance of IMU Motion Sensors - CEVA’s Experts blog*. CEVAâ□™s Experts Blog. Retrieved March 28, 2022, from https://www.ceva-dsp.com/ourblog/what-is-an-imu-sensor/

[17] Rajesh, S. M., Bhargava, S., & Sivanathan, K. (2014). Mission planning and waypoint navigtion of a micro quadcopter by selectable GPS co-ordinates. *Int. J. of Advanced Research in Computer Science and Software Eng.*, *143*.

[18] Jouybari, A., Ardalan, A. A., & Rezvani, M. H. (2017). Experimental comparison between Mahoney and Complementary sensor fusion algorithm for attitude determination by raw sensor data of Xsens IMU on buoy. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *42*, 497-502.

[19] Gattass, M. (n.d.). *3D Rotations: Euler angles and the Gimbal Lock problem*. Prof. Marcelo Gattass Personal Home Page. Retrieved April 2, 2022, from http://webserver2.tecgraf.puc-rio.br/%7Emgattass/demo/EulerAnglesRotations-GimballLock/euler.html

[20] D. (2019, June 20). *How Quaternions Produce 3D Rotation*. Penguinmaths.Blogspot.Com. Retrieved April 2, 2022, from https://penguinmaths.blogspot.com/2019/06/how-quaternions-produce-3d-rotation.html

[21] Team, T. A. I. (2020, May 29). *How, When, and Why Should You Normalize / Standardize / Rescale. . .* Towards AI. Retrieved March 3, 2022, from https://towardsai.net/p/data-science/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38f

[22] Avinash Navlani. (2018, December 4). *Naive Bayes Classifier Tutorial: with Python Scikit-learn*. DataCamp Community. Retrieved January 14, 2021, from https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn

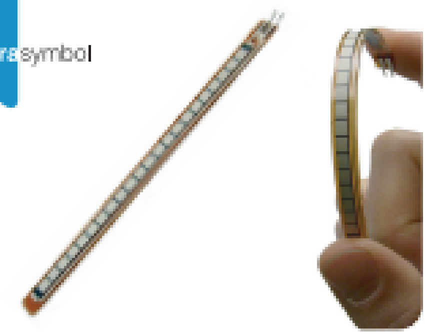
ANNEXURE A

Cost of Implemented unit

Component	Qty	Price (LKR)
Flex Sensor	5	2,500.00
MPU6050 Module	1	350.00
ADS115	2	300.00
33K 2W resistors	4	5.00
10K 2W resistors	4	5.00
Arduino Uno	1	500.00
Clothe glove	1	350.00
Total		**4,010.00**

ANNEXURE B

spectrasymbol



FLEX SENSOR FS

Special Edition Length

Features

- Angle Displacement Measurement
- Bends and Flexes physically with motion device
- Possible Uses
  - Robotics
  - Gaming (Virtual Motion)
  - Medical Devices
  - Computer Peripherals
  - Musical Instruments
  - Physical Therapy
- Simple Construction
- Low Profile

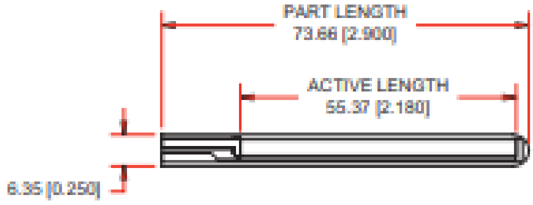
Mechanical Specifications

- Life Cycle: >1 million
- Height: ≤0.43mm (0.017")
- Temperature Range: -35°C to +80°C

Electrical Specifications

- Flat Resistance: 25K Ohms
- Resistance Tolerance: ±30%
- Bend Resistance Range: 45K to 125K Ohms (depending on bend radius)
- Power Rating : 0.50 Watts continuous. 1 Watt Peak

Dimensional Diagram - Stock Flex Sensor



How to Order - Stock Flex Sensor

FS	L	0055	253	ST
Series	Model	Active Length	Resistance	Connectors
FS = Flex Sensor	L = Linear	0055 = 55.37mm	253 = 25K Ohms	ST = Solder Tab

How It Works

Ultra-Small, Low-Power, 16-Bit Analog-to-Digital Converter with Internal Reference

Check for Samples: ADS1113 ADS1114 ADS1115

FEATURES

- ULTRA-SMALL QFN PACKAGE: 2mm × 1,5mm × 0,4mm
- WIDE SUPPLY RANGE: 2.0V to 5.5V
- LOW CURRENT CONSUMPTION: Continuous Mode: Only 150µA Single-Shot Mode: Auto Shut-Down
- PROGRAMMABLE DATA RATE: 8SPS to 860SPS
- INTERNAL LOW-DRIFT VOLTAGE REFERENCE
- INTERNAL OSCILLATOR
- INTERNAL PGA
- I^2C™ INTERFACE: Pin-Selectable Addresses
- FOUR SINGLE-ENDED OR TWO DIFFERENTIAL INPUTS (ADS1115)
- PROGRAMMABLE COMPARATOR (ADS1114 and ADS1115)

APPLICATIONS

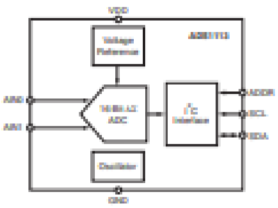
- PORTABLE INSTRUMENTATION
- CONSUMER GOODS
- BATTERY MONITORING
- TEMPERATURE MEASUREMENT
- FACTORY AUTOMATION AND PROCESS CONTROLS

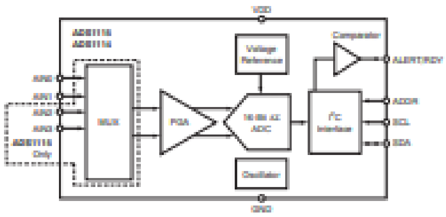
DESCRIPTION

The ADS1113, ADS1114, and ADS1115 are precision analog-to-digital converters (ADCs) with 16 bits of resolution offered in an ultra-small, leadless QFN-10 package or an MSOP-10 package. The ADS1113/4/5 are designed with precision, power, and ease of implementation in mind. The ADS1113/4/5 feature an onboard reference and oscillator. Data are transferred via an I^2C-compatible serial interface; four I^2C slave addresses can be selected. The ADS1113/4/5 operate from a single power supply ranging from 2.0V to 5.5V.

The ADS1113/4/5 can perform conversions at rates up to 860 samples per second (SPS). An onboard PGA is available on the ADS1114 and ADS1115 that offers input ranges from the supply to as low as ±256mV, allowing both large and small signals to be measured with high resolution. The ADS1115 also features an input multiplexer (MUX) that provides two differential or four single-ended inputs.

The ADS1113/4/5 operate either in continuous conversion mode or a single-shot mode that automatically powers down after a conversion and greatly reduces current consumption during idle periods. The ADS1113/4/5 are specified from –40°C to +125°C.





ABSOLUTE MAXIMUM RATINGS(1)

	ADS1113, ADS1114, ADS1115	UNIT
VDD to GND	–0.3 to +5.5	V
Analog input current	100, momentary	mA
Analog input current	10, continuous	mA
Analog input voltage to GND	–0.3 to VDD + 0.3	V
SDA, SCL, ADDR, ALERT/RDY voltage to GND	–0.5 to +5.5	V
Maximum junction temperature	+150	°C
Storage temperature range	–60 to +150	°C

(1) Stresses above those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. Exposure to absolute maximum conditions for extended periods may affect device reliability.

PRODUCT FAMILY

DEVICE	PACKAGE DESIGNATOR MSOP/QFN	RESOLUTION (Bits)	MAXIMUM SAMPLE RATE (SPS)	COMPARATOR	PGA	INPUT CHANNELS (Differential/ Single-Ended)
ADS1113	BROI/N6J	16	860	No	No	1/1
ADS1114	BRNI/N5J	16	860	Yes	Yes	1/1
ADS1115	BOGI/N4J	16	860	Yes	Yes	2/4
ADS1013	BRMI/N9J	12	3300	No	No	1/1
ADS1014	BRQI/N8J	12	3300	Yes	Yes	1/1
ADS1015	BRPI/N7J	12	3300	Yes	Yes	2/4

ELECTRICAL CHARACTERISTICS

All specifications at –40°C to +125°C, VDD = 3.3V, and Full-Scale (FS) = ±2.048V, unless otherwise noted.
Typical values are at +25°C.

PARAMETER	TEST CONDITIONS	ADS1113, ADS1114, ADS1115			UNIT
		MIN	TYP	MAX	
ANALOG INPUT					
Full-scale input voltage[1]	$V_{IN} = (AIN_P) - (AIN_N)$		±4.096/PGA		V
Analog input voltage	AIN_P or AIN_N to GND	GND		VDD	V
Differential input impedance			See Table 2		
Common-mode input impedance	FS = ±6.144V[1]		10		MΩ
	FS = ±4.096V[1], ±2.048V		6		MΩ
	FS = ±1.024V		3		MΩ
	FS = ±0.512V, ±0.256V		100		MΩ
SYSTEM PERFORMANCE					
Resolution	No missing codes	16			Bits
Data rate (DR)			8, 16, 32, 64, 128, 250, 475, 860		SPS
Data rate variation	All data rates	–10		10	%
Output noise		See Typical Characteristics			
Integral nonlinearity	DR = 8SPS, FS = ±2.048V, best fit[2]			1	LSB
Offset error	FS = ±2.048V, differential inputs		±1	±3	LSB
	FS = ±2.048V, single-ended inputs		±3		LSB
Offset drift	FS = ±2.048V		0.005		LSB/°C
Offset power-supply rejection	FS = ±2.048V		1		LSB/V
Gain error[3]	FS = ±2.048V at 25°C		0.01	0.15	%
Gain drift[3]	FS = ±0.256V		7		ppm/°C
	FS = ±2.048V		5	40	ppm/°C
	FS = ±6.144V[1]		5		ppm/°C
Gain power-supply rejection			80		ppm/V
PGA gain match[3]	Match between any two PGA gains		0.02	0.1	%
Gain match	Match between any two inputs		0.05	0.1	%
Offset match	Match between any two inputs		3		LSB
Common-mode rejection	At dc and FS = ±0.256V		105		dB
	At dc and FS = ±2.048V		100		dB
	At dc and FS = ±6.144V[1]		90		dB
	f_{CM} = 60Hz, DR = 8SPS		105		dB
	f_{CM} = 50Hz, DR = 8SPS		105		dB
DIGITAL INPUT/OUTPUT					
Logic level					
V_{IH}		0.7VDD		5.5	V
V_{IL}		GND – 0.5		0.3VDD	V
V_{OL}	I_{OL} = 3mA	GND	0.15	0.4	V
Input leakage					
I_H	V_{IH} = 5.5V			10	µA
I_L	V_{IL} = GND	10			µA

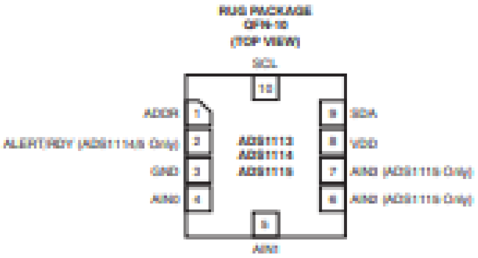
(1) This parameter expresses the full-scale range of the ADC scaling. In no event should more than VDD + 0.3V be applied to this device.
(2) 99% of full-scale.
(3) Includes all errors from onboard PGA and reference.

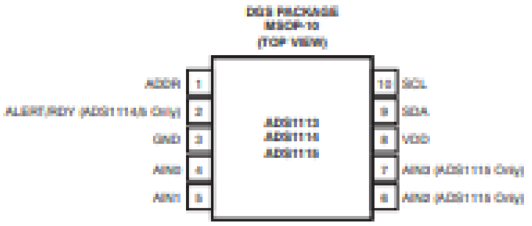
ELECTRICAL CHARACTERISTICS (continued)

All specifications at –40°C to +125°C, VDD = 3.3V, and Full-Scale (FS) = ±2.048V, unless otherwise noted.
Typical values are at +25°C.

PARAMETER	TEST CONDITIONS	ADS1113, ADS1114, ADS1115 MIN	TYP	MAX	UNIT
POWER-SUPPLY REQUIREMENTS					
Power-supply voltage		2		5.5	V
Supply current	Power-down current at 25°C		0.5	2	µA
	Power-down current up to 125°C			5	µA
	Operating current at 25°C		150	200	µA
	Operating current up to 125°C			300	µA
Power dissipation	VDD = 5.0V		0.9		mW
	VDD = 3.3V		0.5		mW
	VDD = 2.0V		0.3		mW
TEMPERATURE					
Storage temperature		–60		+150	°C
Specified temperature		–40		+125	°C

PIN CONFIGURATIONS





PIN DESCRIPTIONS

PIN #	DEVICE ADS1113	ADS1114	ADS1115	ANALOG/ DIGITAL INPUT/ OUTPUT	DESCRIPTION
1	ADDR	ADDR	ADDR	Digital Input	I²C slave address select
2	NC(1)	ALERT/RDY	ALERT/RDY	Digital Output	Digital comparator output or conversion ready (NC for ADS1113)
3	GND	GND	GND	Analog	Ground
4	AIN0	AIN0	AIN0	Analog Input	Differential channel 1: Positive input or single-ended channel 1 input
5	AIN1	AIN1	AIN1	Analog Input	Differential channel 1: Negative input or single-ended channel 2 input
6	NC	NC	AIN2	Analog Input	Differential channel 2: Positive input or single-ended channel 3 input (NC for ADS1113/4)
7	NC	NC	AIN3	Analog Input	Differential channel 2: Negative input or single-ended channel 4 input (NC for ADS1113/4)
8	VDD	VDD	VDD	Analog	Power supply: 2.0V to 5.5V
9	SDA	SDA	SDA	Digital I/O	Serial data: Transmits and receives data
10	SCL	SCL	SCL	Digital Input	Serial clock input: Clocks data on SDA

(1) NC pins may be left floating or tied to ground.

6 Electrical Characteristics

6.1 Gyroscope Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
GYROSCOPE SENSITIVITY						
Full-Scale Range	FS_SEL=0		±250		°/s	
	FS_SEL=1		±500		°/s	
	FS_SEL=2		±1000		°/s	
	FS_SEL=3		±2000		°/s	
Gyroscope ADC Word Length			16		bits	
Sensitivity Scale Factor	FS_SEL=0		131		LSB/(°/s)	
	FS_SEL=1		65.5		LSB/(°/s)	
	FS_SEL=2		32.8		LSB/(°/s)	
	FS_SEL=3		16.4		LSB/(°/s)	
Sensitivity Scale Factor Tolerance	25°C	-3		+3	%	
Sensitivity Scale Factor Variation Over Temperature			±2		%	
Nonlinearity	Best fit straight line; 25°C		0.2		%	
Cross-Axis Sensitivity			±2		%	
GYROSCOPE ZERO-RATE OUTPUT (ZRO)						
Initial ZRO Tolerance	25°C		±20		°/s	
ZRO Variation Over Temperature	-40°C to +85°C		±20		°/s	
Power-Supply Sensitivity (1-10Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		°/s	
Power-Supply Sensitivity (10 - 250Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		°/s	
Power-Supply Sensitivity (250Hz - 100kHz)	Sine wave, 100mVpp; VDD=2.5V		4		°/s	
Linear Acceleration Sensitivity	Static		0.1		°/s/g	
SELF-TEST RESPONSE						
Relative	Change from factory trim	-14		14	%	1
GYROSCOPE NOISE PERFORMANCE	**FS_SEL=0**					
Total RMS Noise	DLPFCFG=2 (100Hz)		0.05		°/s-rms	
Low-frequency RMS noise	Bandwidth 1Hz to10Hz		0.033		°/s-rms	
Rate Noise Spectral Density	At 10Hz		0.005		°/s/ √Hz	
GYROSCOPE MECHANICAL FREQUENCIES						
X-Axis		30	33	36	kHz	
Y-Axis		27	30	33	kHz	
Z-Axis		24	27	30	kHz	
LOW PASS FILTER RESPONSE						
	Programmable Range	5		256	Hz	
OUTPUT DATA RATE						
	Programmable	4		8,000	Hz	
GYROSCOPE START-UP TIME	**DLPFCFG=0**					
ZRO Settling (from power-on)	to ±1°/s of Final		30		ms	

1. Please refer to the following document for further information on Self-Test: *MPU-6000/MPU-6050 Register Map and Descriptions*

6.2 Accelerometer Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
ACCELEROMETER SENSITIVITY						
Full-Scale Range	AFS_SEL=0		±2		*g*	
	AFS_SEL=1		±4		*g*	
	AFS_SEL=2		±8		*g*	
	AFS_SEL=3		±16		*g*	
ADC Word Length	Output in two's complement format		16		bits	
Sensitivity Scale Factor	AFS_SEL=0		16,384		LSB/*g*	
	AFS_SEL=1		8,192		LSB/*g*	
	AFS_SEL=2		4,096		LSB/*g*	
	AFS_SEL=3		2,048		LSB/*g*	
Initial Calibration Tolerance			±3		%	
Sensitivity Change vs. Temperature	AFS_SEL=0, -40°C to +85°C		±0.02		%/°C	
Nonlinearity	Best Fit Straight Line		0.5		%	
Cross-Axis Sensitivity			±2		%	
ZERO-G OUTPUT						
Initial Calibration Tolerance	X and Y axes		±50		mg	1
	Z axis		±80		mg	
Zero-G Level Change vs. Temperature	X and Y axes, 0°C to +70°C		±35			
	Z axis, 0°C to +70°C		±60		mg	
SELF TEST RESPONSE						
Relative	Change from factory trim	-14		14	%	2
NOISE PERFORMANCE						
Power Spectral Density	@10Hz, AFS_SEL=0 & ODR=1kHz		400		µg/√Hz	
LOW PASS FILTER RESPONSE	Programmable Range	5		260	Hz	
OUTPUT DATA RATE	Programmable Range	4		1,000	Hz	
INTELLIGENCE FUNCTION INCREMENT			32		mg/LSB	

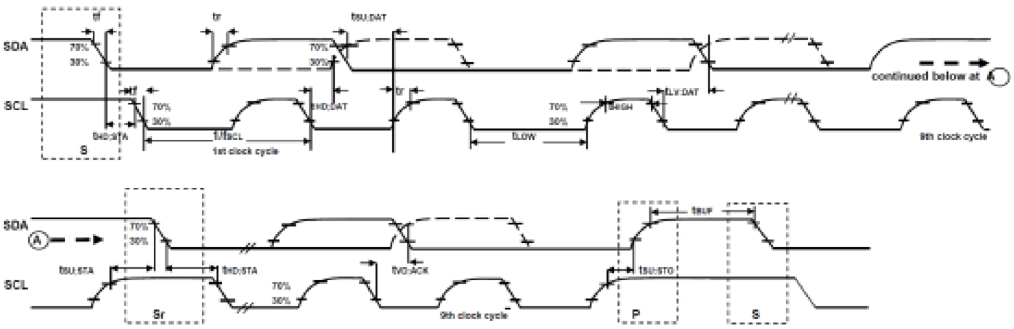
1. Typical zero-g initial calibration tolerance value after MSL3 preconditioning
2. Please refer to the following document for further information on Self-Test: *MPU-6000/MPU-6050 Register Map and Descriptions*

6.7 I²C Timing Characterization

Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

Parameters	Conditions	Min	Typical	Max	Units	Notes
I²C TIMING	**I²C FAST-MODE**					
f_{SCL}, SCL Clock Frequency				400	kHz	
$t_{HD.STA}$, (Repeated) START Condition Hold Time		0.6			μs	
t_{LOW}, SCL Low Period		1.3			μs	
t_{HIGH}, SCL High Period		0.6			μs	
$t_{SU.STA}$, Repeated START Condition Setup Time		0.6			μs	
$t_{HD.DAT}$, SDA Data Hold Time		0			μs	
$t_{SU.DAT}$, SDA Data Setup Time		100			ns	
t_r, SDA and SCL Rise Time	C_b bus cap. from 10 to 400pF	20+0.1C_b		300	ns	
t_f, SDA and SCL Fall Time	C_b bus cap. from 10 to 400pF	20+0.1C_b		300	ns	
$t_{SU.STO}$, STOP Condition Setup Time		0.6			μs	
t_{BUF}, Bus Free Time Between STOP and START Condition		1.3			μs	
C_b, Capacitive Load for each Bus Line			< 400		pF	
$t_{VD.DAT}$, Data Valid Time				0.9	μs	
$t_{VD.ACK}$, Data Valid Acknowledge Time				0.9	μs	

Note: Timing Characteristics apply to both Primary and Auxiliary I²C Bus



I²C Bus Timing Diagram

6.9 Absolute Maximum Ratings

Stress above those listed as "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these conditions is not implied. Exposure to the absolute maximum ratings conditions for extended periods may affect device reliability.

Parameter	Rating
Supply Voltage, VDD	-0.5V to +6V
VLOGIC Input Voltage Level (MPU-6050)	-0.5V to VDD + 0.5V
REGOUT	-0.5V to 2V
Input Voltage Level (CLKIN, AUX_DA, AD0, FSYNC, INT, SCL, SDA)	-0.5V to VDD + 0.5V
CPOUT (2.5V ≤ VDD ≤ 3.6V)	-0.5V to 30V
Acceleration (Any Axis, unpowered)	10,000g for 0.2ms
Operating Temperature Range	-40°C to +105°C
Storage Temperature Range	-40°C to +125°C
Electrostatic Discharge (ESD) Protection	2kV (HBM); 200V (MM)
Latch-up	JEDEC Class II (2),125°C ±100mA

5.1 JANALOG

Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	Analog input 1 /GPIO
11	A2	Analog/GPIO	Analog input 2 /GPIO
12	A3	Analog/GPIO	Analog input 3 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

5.2 JDIGITAL

Pin	Function	Type	Description
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select
12	MOSI	Digital	SPI1 Main Out Secondary In
13	MISO	Digital	SPI Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Power	Ground
16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)