EASYTALK: A TRANSLATOR FOR SRI LANKAN SIGN LANGUAGE

Project ID: 2020-077

Final (Draft) Report

D. Manoj Kumar IT17050272

G.M.A.S. Bastiansz IT17143950

K. Bavanraj IT17032766

S. Thavananthan IT17068192

B.Sc. (Hons) Degree in Information Technology
Specialized in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology
Sri Lanka

September 2020

EASYTALK: A TRANSLATOR FOR SRI LANKAN SIGN LANGUAGE

Project ID: 2020-077

Final (Draft) Report

The dissertation was submitted in partial fulfilment of the requirements for the BSc Special Honors degree in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology
Sri Lanka

September 2020

DECLARATION

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 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.

Also, I hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation in whole or part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as article or books).

Name	Student ID	Signature
D. Manoj Kumar	IT17050272	
G.M.A.S. Bastiansz	IT17143950	
K. Bavanraj	IT17032766	
S. Thavananthan	IT17068192	

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

…………………………………….. ……………………..

Signature of the Supervisor: Date:

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to the supervisor, Mr. S.M.B. Harshanath & the co-supervisor, Mr. J. Alosius for their continued support & advice for this research. They provided us with great insight & how to & what not to do when it comes to conducting professional research. Moreover, we would like to thank Dr Dharshana Kasthurirathna for his continuous encouragement & for providing useful information & tips regarding the research domain.

Above all, our extreme thanks go to our parents who actively encouraged us to excel in our academics & providing a great motivation to take this research as an opportunity to show our skills in academics.

ABSTRACT

A Language is a way of communicating thought between two parties. Effective communication exists when both parties actively engage in the session and respond. Just like speech and text that we use to communicate, sign language is also a method used by the deaf and mute community all over the world to communicate with each other and with other normal people. Not all of us are aware of the sign language and we do require a translation. As Sri Lanka is still known as a developing country, most of the people don’t have a clear understanding of the sign language system which is used locally. Therefore, many ordinary people are refusing to communicate with disabled people. Only the people who engage with these community are willing to learn sign language. There are some basic translator systems which are used to translate sign language into normal vocal language. But there are no robust interpreters available in Sri Lanka and most of them lack in real-time translation. Cost of the existing systems are high, and the quantity is not enough for a wide communication system.

Through this report, we are proposing a solution to develop a translation system with a low-resolution camera, which should be able to convert Sri Lankan sign language into English and vice-versa in real-time. The research is divided into four major components. The first component does the data acquisition and feature extraction part. Then it moves to the translation part where extracted images are changed into letters. The third component will do the text and voice assistant part and the final component will convert the texts into sign images. Our main goal is to achieve a system that can perform in real-time. With the advancement of image processing and machine learning, we think we can attain our goal.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

2D – 2 Dimensional

AI – Artificial Intelligence

ANN – Artificial Neural Network

API – Application Programming Interface

ASL – American Sign Language

CNN – Convolutional Neural Network.

COVID – Corona Virus Disease

CV – Computer Vision

DTW – Dynamic Time Wrapping.

FPS – Frames Per Seconds.

GIF – Graphics Interchange Format.

GoSL – Government of Sri Lanka

GUI – Graphical User Interface

HD – High Definition.

HMM – Hidden Markov Models.

HOG – Histogram of Oriented Gradients.

ICTA – Information Communication Technology Association

IDE – Integrated Development Environment

JSON – JavaScript Object Notation

KB – Kilo Bytes

KNN – Kth Nearest Neighbor algorithm.

LMC – Leap Motion Controller.

MLP – Multilayer Perception algorithm.

NLP - Natural Language Processing

OS – Operating System

PWD – Persons With Disabilities

RCNN – Region-based Convolutional Neural Network

ROI – Region of Interest

SDK – Software Development Kit

SDLC – Software Development Life Cycle

SSL – Sri Lankan Sign Language

TTS – Text To Speech

VS – Visual Studio

Wi-Fi – Wireless Fidelity

XML – eXtensible Markup Language

1. INTRODUCTION

1.1. Background

In Sri Lanka, 8.7% of the total population is considered as PWD [1]. They possess any form of disability from hearing impaired to physical disability. When we take the portion of the population who cannot speak or hear, they usually communicate using Sign Languages. Different types of Sign Languages are practiced around the world. In the Sri Lankan context, the deaf & mute of the country also follows a Sign Language called 'The Sri Lankan Sign Language' [2] which comprises of Sinhala, Tamil and English alphabets and a set of pre-defined words which are for common use.

The literacy rate of Sri Lanka is well above 96% [3] and growing. This is mainly due to the introduction of the free education system in the early 1950s. This action has led to creative and innovative solutions for different problems faced by different sectors of the community. However, the contribution of the deaf and mute community is much lesser due to communication issues, lack of financial and provisional donations etc. They're reluctant to actively participate in the socio-cultural activities in the background that they cannot speak or hear like other normal people.

According to basic human rights, the needs and rights of the deaf and mute should also be listened and fulfilled [4]. In Sri Lanka however, this is followed up to some extent. The country has special schools, meal programs, government aids etc. to keep them running daily. But their right to be a helpful citizen just like the others is not taken from them but rather ignored.

Table 1: Population of Deaf and Mute.

Types of difficulties	Number of Persons	% to total Cases	Number of persons not possible at all	% to total Case	Difficulty published in census	% to total Cases
Hearing	354,871	22.8	28,674	20.3	389,077	24.0
Communication	133,623	8.6	47,210	33.5	180,833	11.2

According to the above table, we can see that there is a considerable amount of deaf and mute. This is just a census report from 2012. At the current date (2020) there can be even more individuals who are deaf and mute.

1.2. Literature Survey

to the Census of Population & Housing – in 2012, there were 21 people older than 5 years out of a sample of 1000 people were suffering from difficulties in deaf [5]. The deafness can occur to someone at any stage & this may impact on his/her abilities to function well.

People around the world are using different types of sign languages such as Sri Lankan Sign Language [6], American Sign Language [7], British Sign Language [8], Indian Sign Language [9] & French Sign Language [10]. All these sign languages are unique to each country.

There are many types of research going in sign language translators all over the world. Most of them use sensors to detect the signer's actions. The main advantage of the sensors is they get a clear view of the signs and they detect edges. Commonly used sensors are Kinect and Leap motion controller. Kinect sensors are used to capture 3D depth images [14]. Leap Motion controllers use near-infrared rays to track hands and take greyscale images [15]. These high-end sensors can help to reduce the time taken in translations.

This research is mainly focusing on Sri Lankan Sign language which is used by hearing-impaired & inarticulate community in Sri Lankan society. There are some applications which were developed based on SSL too. “Nihanda Ridma”, “Ahanna” & “Katharina” are three examples for those applications based on SSL. Through this Literature Survey section, we try to provide a summary of each system we mentioned above & comparison with our proposed system.

1.2.1. Nihanda Ridma System

“Nihanda Ridma” is an application which can be used to convert dynamic gestures into text or voice & vice versa [11]. It is used as a game-based learning system for children with hearing-impaired. There are three components which are focused throughout the project. They are 3D model animating, Motion Tracking & Voice Recognition. 3D model animation used to output the result to a deaf person. Motion Tracking is used to detect & track the movement of the person & it acted as the key ingredient to recognize the sign language gesture in this application. “Nihanda Ridma” captures the text which entered by the user & finds out the relevant animation clip with the sign. Then the system displays the animation clip to the user in avatar mode. It is a web-based application.

1.2.2. Ahanna System

“Ahanna” system is mainly targeting teaching Sinhala Sign Language to the users who are willing to use the system. This is also a web-based application. The main purpose of this application is to spread Buddhism through the deaf community in Sri Lanka. It is also providing many other benefits such as giving more valuable activities, innovative products & new thoughts to improve the knowledge, education of the deaf community.

1.2.3. Kathana, Speech Recognition System

“Kathana” is a Sinhala Speech Recognition System [12]. It uses to convert speeches on the Sinhala language captured by the microphone, to a group of words. Then the recognized words are used as commands, data entries for application.

1.2.4. Sanwadha System

“Sanwadha” is an intelligent mobile assistant application which was build targeting hearing-impaired people in Sri Lanka [13]. The core of this project is Instant Messaging (IM). Here the system gets the Sinhala text from an ordinary person &

converts it into SSL. This output displays as a GIF. Also, through the system, a user can convert SSL into text or voice format as he/she prefers. The main objective of this application is to reduce the digital divide between enabled & hearing-impaired users.

1.2.5. Mexican sign language recognition using Kinect

In this research, they talk about the usage of the Kinect sensor. When a signer shows a sign, the system will store the colour, depth and the skeleton tracking information using the sensor. Then using the Dynamic Time Wrapping (DTW) algorithm the gathered information is interpreted. For testing, they have used the K-Fold Cross-validation method. This testing has shown a mean accuracy of 99.1%. From these results, we can understand how accurate the Kinect sensor in recognition is [16].

1.2.6. Real-time sign language recognition using the Leap Motion Controller

This system has used a leap motion controller to detect and track signers' hands and fingers. The extracted and normalized features will be sent to the classification part. In the classification part, they have used Multilayer Perception algorithm (MLP) which uses features as input and convert them into specific alphabet letter. For training, authors have implemented the Backpropagation algorithm. The total system has shown 96.15% of a recognition rate. This high percentage is achieved only because of the quality feature extraction through LMC [17].

1.2.7. Sign Language Recognition using Microsoft Kinect

This paper is focusing on identifying Sign Language using Microsoft Kinect [18] device. Using computer vision algorithms, they've developed a characteristic depth and motion profile for each sign language gesture. The feature matrix thus generated was trained using a multi-class SVM classifier and the results were compared with existing techniques [19]. The practical application of this system seems expensive since this system requires a Kinect device. The monetary provisions and technical expertise lack in this country. A mobile application or a web camera would be much more feasible since everyone has access to the internet and especially mobile phones.

1.3. Research Gap

There is a communication gap between people with difficulties in hearing -speaking & ordinary people. Most of the time people use interpreters while communicating with hearing-impaired & mute people. But when there are no interpreters (translators), people face a big problem because they cannot understand what each other says. To avoid this issue the countries like America, India found out various inventions what can easily convert sign language to textual format & vice-versa [20]. But those systems are not practical systems for Sri Lanka based on some reasons. The first reason is Sri Lanka still listed as a developing country. The tools & technologies which were used to build such systems by other countries are very expensive nowadays. Sri Lankan government cannot be able to spend that amount of money for only one main purpose. The second reason is that even we buy one of those systems, people cannot get the benefits of the system as each country has their sign language. Another reason is that the system which already implemented for a specific language such as Sinhala. Due to this reason, if any person who doesn't have a proper practice in that language won't understand the system. Therefore, by-today the ordinary people who are willing to communicate with hearing-impaired & mute people have challenged with a huge communicating gap in their daily lives.

Table 2: Comparison between the proposed system and existing systems

Features	Nihanda Ridma	Ahanna	Kathana	Sanwadha	Proposed System's Component
Text recognition system	✗	✗	✓	✓	✓
Convert English text to sign language	✗	✗	✗	✗	✓

Display the combination of signs as a GIF	✗	✗	✗	✓	✓
Low resolution web application	✗	✗	✗	✗	✓

1.4.Research Problem

Sri Lanka is very hospitality country from the ancient times. Almost all the people in the country don’t think twice to friend with another person who looks in pleasant manner. Therefore, in Sri Lanka “communication” acts a main role among the society.

As mentioned in Research Gap, there are different types of people in Sri Lanka. Among those categories, people with difficulties in speaking & hearing get a special attention. Those people use Sri Lankan Sign Language (SSL) for their communication purposes with people with same manner. But problems arise when there is no proper understanding about the SSL among the ordinary people. This issue may be occurred when having a poor guidance about the SSL. Even though many Sri Lankan universities, private institutions who are holding degrees & diplomas on SSL, people won’t show any interesting because of above mentioned reason. Even there is a bit of interesting, only limited number of students get the chance to study. Due to this reason most of the ordinary people lose the chance to learn SSL. Therefore, they are refusing to be friendly & communicate with hearing-impaired & mute people frightening both parties could not be able to understand what each other tries to say.

But it is important to build a communication bridge between ordinary people & people with difficulties with speaking & hearing. Hearing-impaired & mute people always communicate physically & visually. But those people too refuse to talk with ordinary people because ordinary people can’t recognize the language if there are no interpreters for translation purposes.

Even there are some applications in Sri Lanka which are capable of translating SSL into natural vocal language or translating natural local language into SSL, it is hard to find out a system which can do both translation modes using one interface. Also, most of the all the implemented system uses Sinhala language for translating purposes. Even Sinhalese is the main language in Sri Lanka, some people cannot even understand Sinhalese. In cases like that, ordinary people or deaf people or mute people who are not familiar with Sinhala face to many problems while using existing systems.

Another problem is that using only one type of data set for translating purposes in the system. In many research papers we can see that the researchers use only Sri Lankan Sign Language alphabet "or" the set of pre-defined words in their research projects. It is not a best practical to use only one type of data sets in a translating system due to many reasons.

If there is a proper converting application instead of interpreters which can be used as a self-guidance, ordinary people as well as people with difficulties in hearing & speaking will be able to use them whenever they want. Then, it may be help them to break the communication gap & be friendly with each & everyone in the society.

This is the problem address in our research component. Some researchers have put their effort to address this fact. Also, they have come up with applications. But unfortunately, none of those application can fulfill all the requirements, which are expecting from an application such as converting day-to-day words into SSL, presenting converted answer as a proper identifiable output & vice-versa.

1.5. Objectives

1.5.1. Main Objective

- The main idea of this research is to improve the communication bond between the ordinary people & people with difficulties in hearing & speaking.
- To successfully fulfil the above idea, this system should be able to convert SSL into text format & converting English text format into SSL using low-resolution web camera.

The sub-objectives which are specified to each component can be listed as below:

1.5.2. Specific Objectives

- To verify that the product is reliable for ordinary people who are willing to keep better communication bond with people with difficulties in hearing & speaking and vice-versa.
- To capture an image using low resolution camera and train models through those low-resolution images. This would facilitate the system being used in a web interface of a laptop camera.
- To use R-CNN, in order to identify series of hand gestures with the aid of pre trained models of low-resolution images.
- To prepare an optimal dataset to create the effective data model.
- To apply a proper machine learning algorithm for translation.
- To optimize the overall model to real time translating.
- To convert alphabet into a textual format.
- To investigate already-defined Sri Lankan signs for day-to-day words.
- To prepare the given text message to a GIF to convert the message to SSL with more accurately & eye-pleasant manner.
- To emerge the application in English language to reach the Sri Lankan society in effective way.
- To determine the usage of low-resolution web application for hearing-impaired people, mute people as well as ordinary people can be observed as an effort, which allows those communities to clarify any need of learning & communication at anytime, anywhere.
- All the above objectives should be accomplished in real time. This is because we are proposing to build a web application in which the application response speed is of utmost importance.

2. METHODOLOGY

2.1. System Overview

In this part, the main focus is on how the application is developed. The overall system is divided into four main components.

- Data Acquisition
- Sign Language Translation
- Text and Audio output generation
- Text to Sign Language conversion

This section shows how the all four components are implemented and integrated with each other. First, the hand is detected and send to a sign classifier. The classified sign is sent for output generation. On the other hand, the fourth component converts text into sign language.

The proposed system is to be developed as a real-time system that produce instant output without any additional clicks or taps. The goal is to minimize user clicks and increase the usability of the application. The application will be a web based responsive application. This is to make the application available and usable on the mobile and tablet browsers as well.

Various methodologies and technologies have been used for the development of this application. The application is developed in a way that it will overcome most of the technical barriers such as:

- Slow internet
- Devices with less specs
- Web Cameras with low resolution.

The below diagram shows the overall system diagram.

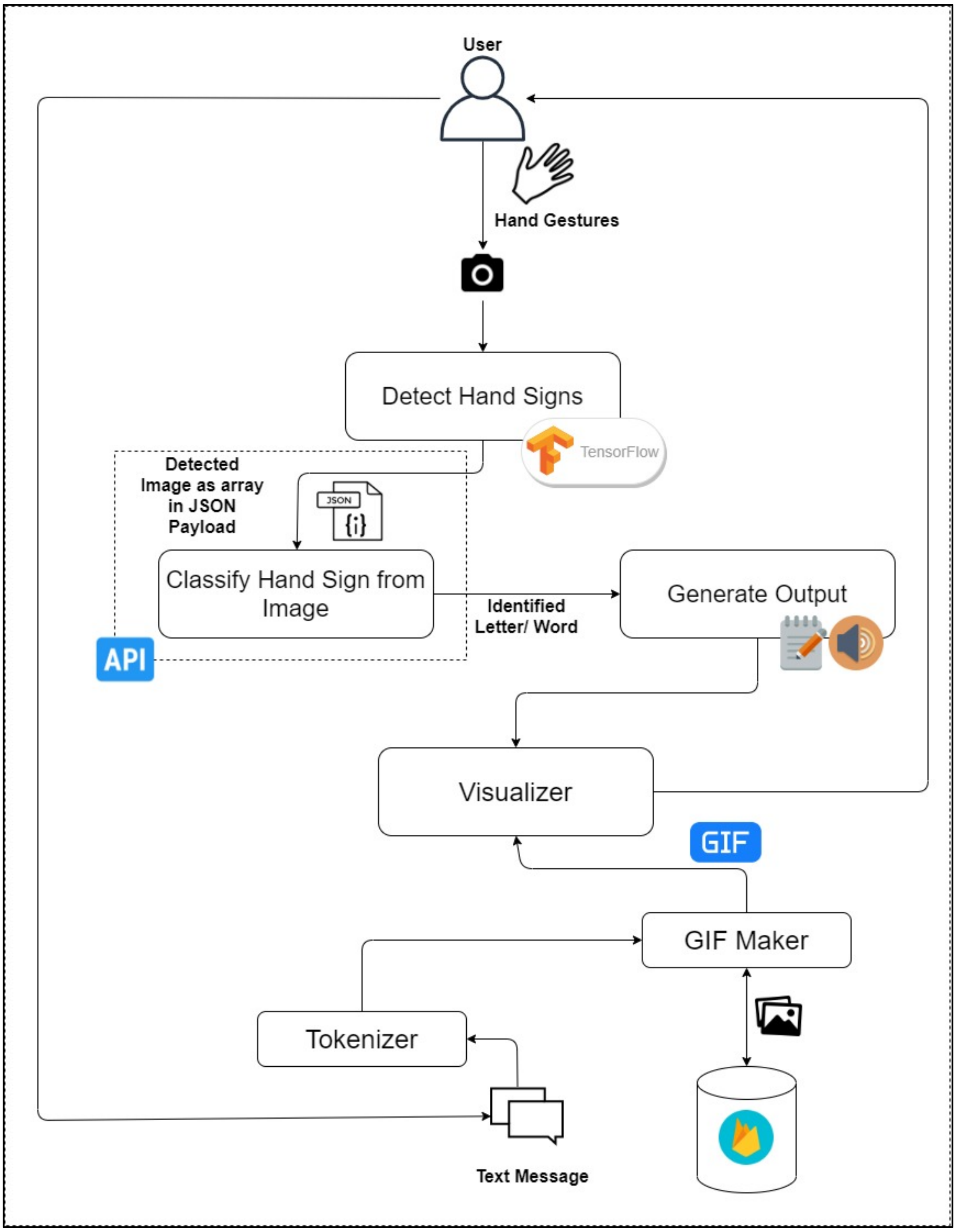

Figure 2.1: System diagram.

2.1.1. Data Acquisition

In this chapter, the focus is on how the research component – Hand Gesture Detection is implemented and tested. The implementations include how the data is a gathered, initial insights on how the methodology was thought to be implemented, how it was implemented and to a working component.

This part of the solution (research component) presents a hand sign detector to be used to classify different hand signs.

To implement the system using Faster RCNN, we must first understand the concept of CV. CV is used for many applications and most of them include image classification. Features such as edge detection, Object Detection, Object segmentation and Semantic Segmentation are some of the key features of CV. In this research component, the main objective is to identify hands (hand gestures) in order to take imagery and send it to the next component for classification.

In brief, the model is trained on top of TensorFlow models with Faster RCNN configuration. The generated model is an inference graph and will be stored locally in the running system. Once given a stream of images through web camera's live feed, the model will detect hands and if the hand matches one of the input, it will show a bounding box over the detected hand in real-time. The bounding box will show what the object is wrapped around and how accurately it has detected. Initially, the datasets are created using hand signs by the members.

The datasets are created in a way that for each English SSL, 05 images are taken manually. Those image sets were taken in the laptop camera which is of 0.3MP. Each of these images is of approximately 300KB – 600 KB depending upon the colour content of each image. After that, they're labelled in *PASCAL* format. After generating the labels for each image and the label are exported as .xml files. Using a small piece of code, the individual XML files are aggregated into two files both for training and testing respectively. Those training and testing label sets are then used by the TensorFlow trainer to train the model. It is a long process and requires time to train on average computers. It is possible to see how good or bad the model is being trained in tensorboard. Tensorboard is a web based GUI where it is possible to have all insights about the training process in real time.

Once training process is completed and the loss level goes below 0.2, it is recommended to terminate the trainer. It saves an inference graph (TensorFlow model) which it refers while detecting objects. Once it is done, the next step is to take the Faster RCNN configuration from the TensorFlow model garden and edit the label information according the project's needs. Here, it is only one class to be labelled at(hand). So, it is comparatively easier to train a model in this scenario rather than training a model for multi-class detection/classification purposes.

The below figure shows the system overview.

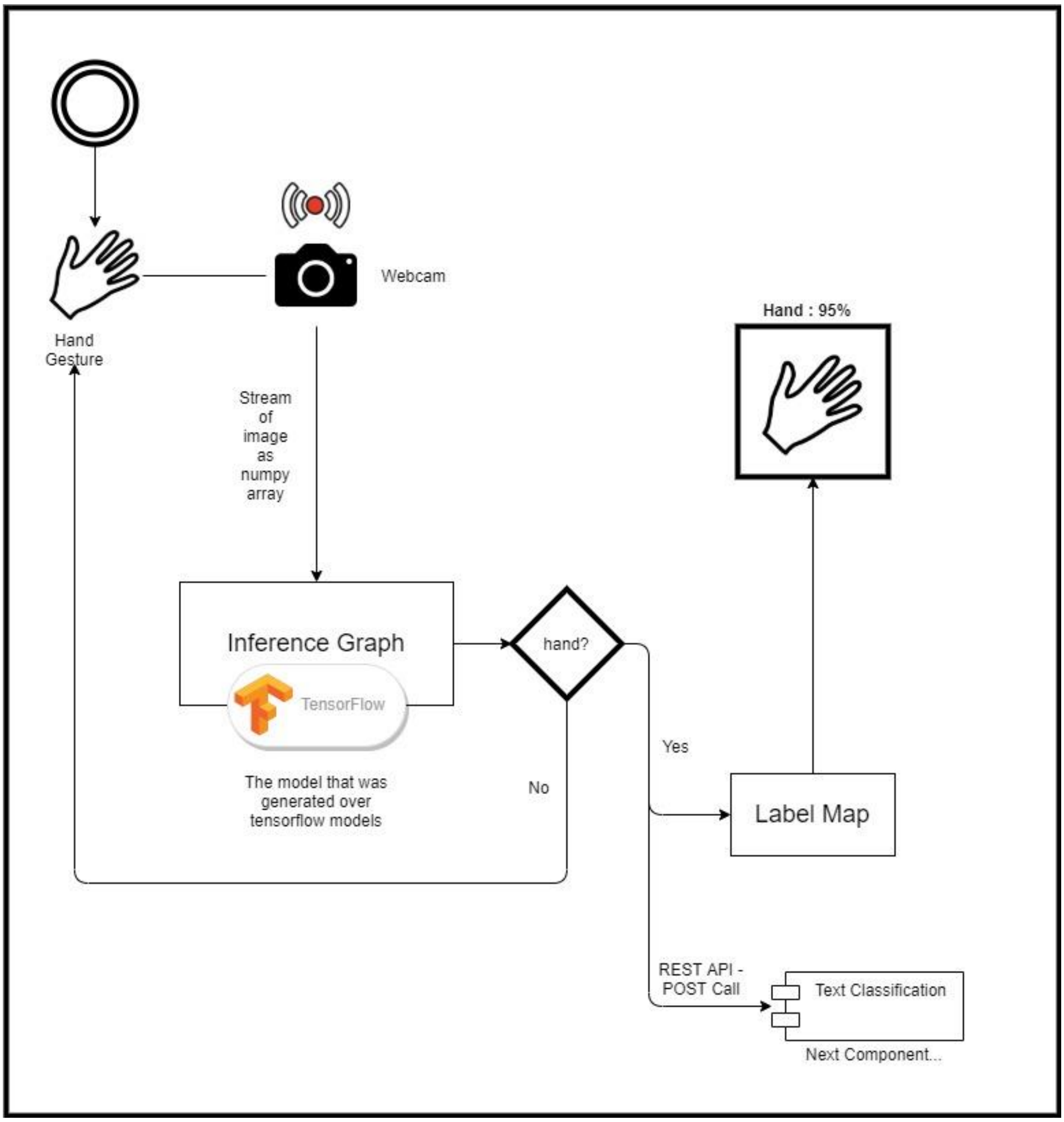

Figure 2.2: System Overview Diagram for Data Acquisition Model.

2.1.2. Sign Language Translation

The individual component being described here is “Sign Recognition and Translation” in real time. It is one of the major components in the web application “EasyTalk”. Basically, this component is built as an API which contains a machine learning model in it. This API fetches a sign image and identifies the English alphabet letter through the CNN machine learning model. Then the identified letter will be passed to the next component in “EasyTalk”. The main objective of this component is to translate it in real time.

Figure 2.3 shows the system diagram of the component. The extracted sign images are fetched from before component and will be send to the classification model. In the classification model the images are compared with the pretrained dataset and the letter according to the sign will be identified. The identified letters will be sent to the next component so that it will be made as a meaningful word. Optimizing the whole component is the final part. Translations has to happen in real time. For that, optimization must happen in the classification part. Time optimization will happen in two parts. The first part is measuring the total time that has been taken by the method to build the training model. Second part is measuring the total time that has been taken to find the result, based on the test data. By using these results, we can optimize the time in each part and achieve the "real time processing" goal.

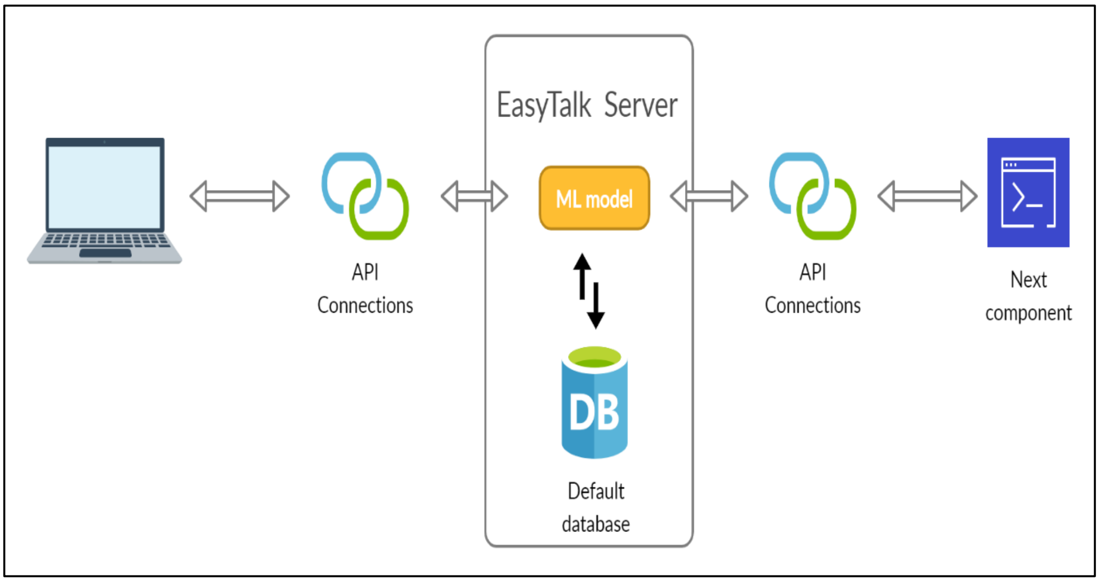

Figure 2.3: System Overview Diagram for Sign Translation Model.

2.1.2.1. Data Collection

To build this research component we have to go through several steps. The major part for this component is the dataset. As the system is going to develop through a machine learning algorithm, we must choose the optimized dataset for the process. At first, we were planning to take 500 images for a sign alphabet. As there are 26 letters in the English alphabet, we gathered 13000 images to train the module.

Figure 2.4: Collected sign images.

Multiple signers show the signs of English alphabet in Sri Lankan sign language method. These images were taken by the web camera in different lightning conditions and backgrounds. From these images we select the most optimal images to create the effective data model.

2.1.2.2. Machine Learning Model

As the next step the dataset has to be trained using a machine learning algorithm such as Convolutional Neural Network (CNN). CNN algorithm can take images as an input and can learn different objects in the images. The image will be recognized as an array of pixels and it depends on the image quality. At the end the algorithm will process the image and classify it under given categories. So, in our context we can categorize the set of images to the A - Z alphabet. Compared to other algorithms CNN doesn't need much pre-processing for the image. It increases the processing speed of the component, which is a main advantage of using this algorithm [23].

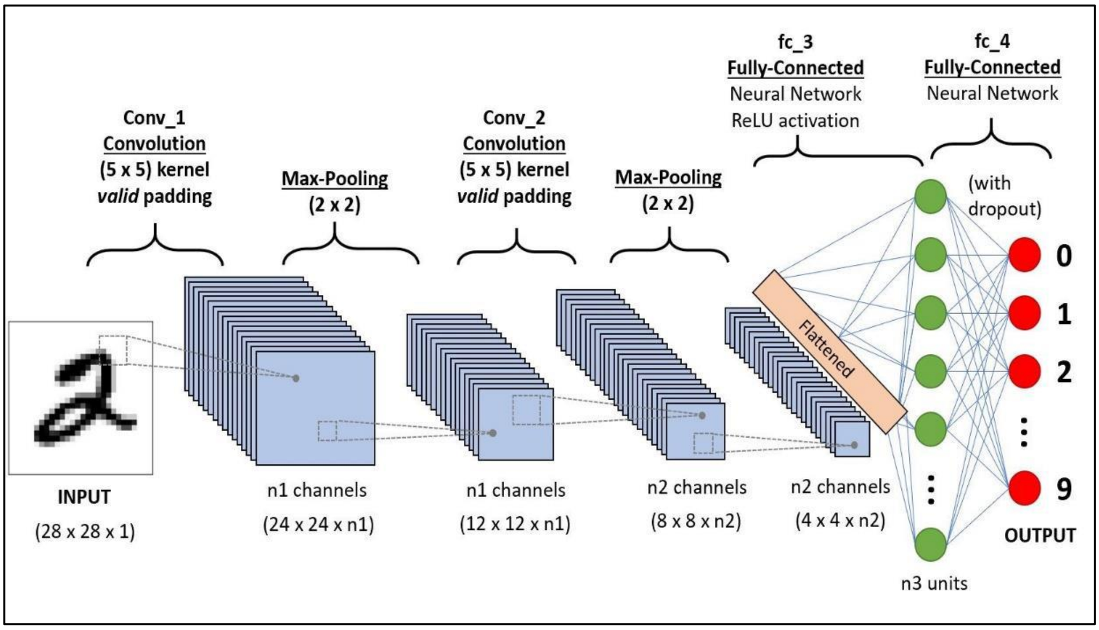

Figure 2.5: Process in the CNN algorithm.

For better results we must optimize the selected machine learning algorithm. After training the dataset and optimizing the algorithm we can predict the signs using the image flow. Once the sign is identified, it will be passed to the next research component as an English alphabet letter.

Figure 2.6 illustrates the code segment which shows the RGB pixel intensities of the image which is captured through web camera.

```
[ ] import matplotlib.pyplot as plt

    def plotHistogram(a):

        plt.figure(figsize=(10,5))
        plt.subplot(1,2,1)
        plt.imshow(a)
        plt.axis('off')
        histo = plt.subplot(1,2,2)
        histo.set_ylabel('Count')
        histo.set_xlabel('Pixel Intensity')
        n_bins = 30
        plt.hist(a[:,:,0].flatten(), bins= n_bins, lw = 0, color='r', alpha=0.5);
        plt.hist(a[:,:,1].flatten(), bins= n_bins, lw = 0, color='g', alpha=0.5);
        plt.hist(a[:,:,2].flatten(), bins= n_bins, lw = 0, color='b', alpha=0.5);
    plotHistogram(X_train[1])
```

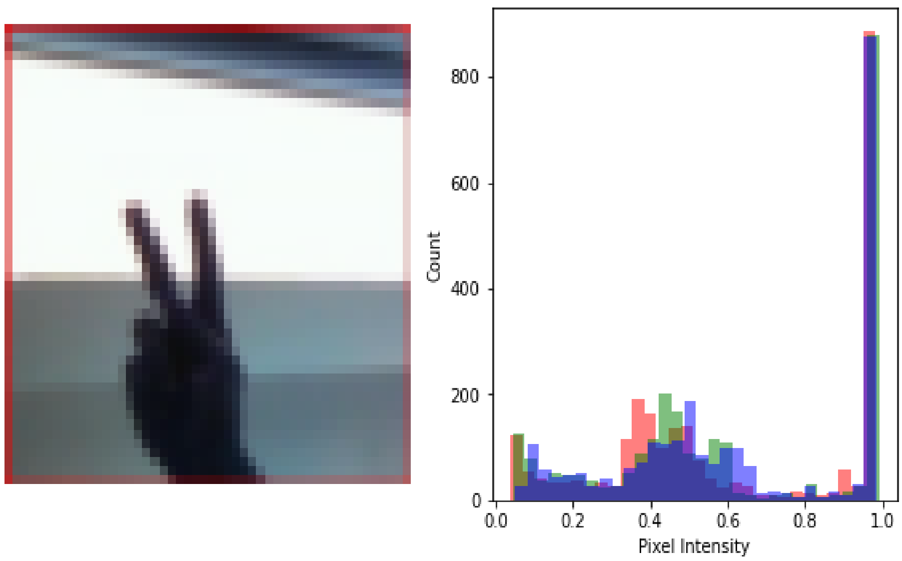

Figure 2.6: Code segment and output for pixel intensity.

2.1.2.3 Building API

After completing the machine learning model, we have to get the use out of it. So, we used Flask to create the API. Flask is a web services framework for Python. It comes with an embedded web server that requires minimal configuration and can be managed with our Python scripts. When a HTTP request sent to the API, it gets the sign image, and do the classification. Identified alphabet will be sent as response.

2.1.3. Text and Audio output generation

In this chapter, the focus is on how the research component – Text and Audio output generation. The implementations include how to convert alphabet into a textual format, check the spelling for each collection and convert text to speech. how it was implemented and to a working component.

The Predicted Stop Container is divided into the following essential components:

- Convert alphabet into a textual format
- Check the spelling for each collection.
- Convert text to speech local languages.

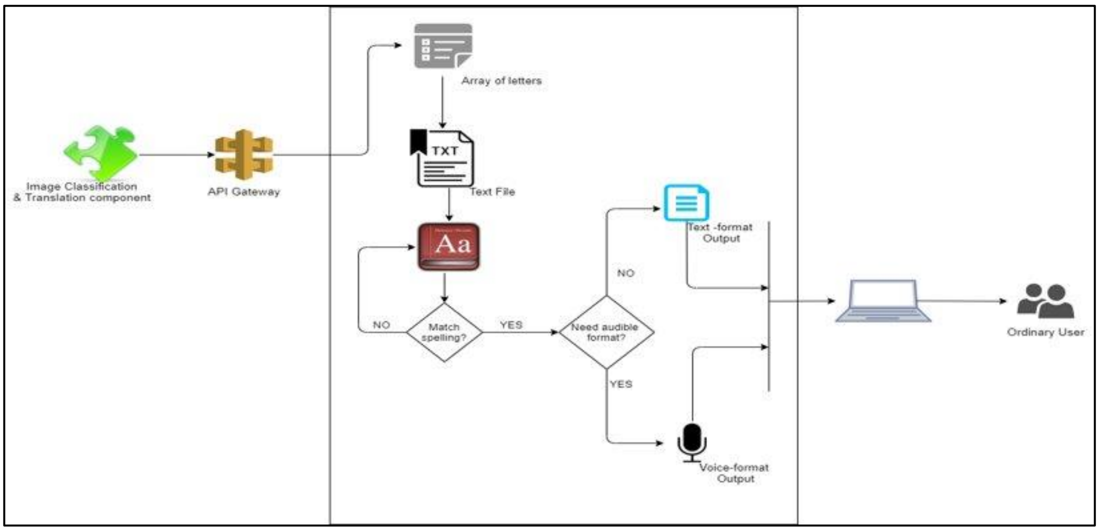

Figure 2.7: Component overview.

2.1.3.1 Convert alphabet into a textual format

The trillion-word data supported was republished in 2006 by Google's Thorsten Brands and Alex France, made possible by the Linguistic Data Federation. Information classification summarizes the primary texts by counting the number of word origins and the sequence of words two, three, four, and five. Nevertheless, the Turks are the most outraged organization, and the Republicans and liberals get angry from time to time. However, Democrats and Conservatives have not made a list. How? Would I say this data is beautiful and not a bit mediocre? The unique number is average. Numbers of billions — even billions — is gorgeous because it doesn't say much about the English language. It sounds too much, except about the world in which the speakers live. The data is attractive because it represents what it deserves to say. Before we look at everything we can do with data, we need to learn the j words we need to speak. Each choice of text called for a corpus. We negotiate the canon as a series of stories and punctuation of tokens. Many separate tickets are called one type, so there are four tickets in the text "Run, Lola Run" (individual comma number one) and only three types. All kinds of collections are called vocabulary. Google Corpus has a trillion features and 13 million classes. The 1-token progression continues to be a unigram, and the 2-token file is a pictogram. Also, an n-token array is an n-gram. Bis about probability, while against p (the) = .022, the possibility of the token "the" relative is .022 or 2.2%. If W is a series token, then W3 is the third token. Also, W1: 3 is the first upgrade by third tokens. P (Wi = the | Wi-1 = of) is the forbidden possibility of "of," which means "of" refers to the last receipt. Individual readers will make the right decision by pulling out the exaggerated many years of experience; it is impossible to encode that knowledge into a supercomputer algorithm. However, we want the container to be an alternative that works surprisingly well: see the Pigram table's phrase.

Suppose it is challenged, including the obligation to refer to the code word "sufficient numbers." Imagine for a second you were transposed into the karmic driven world of Earl. Meanwhile, the numbers: 20751 Adequate Numbers 32378 "Adequate Numbers" is 50% higher than "Adequate Numbers," but it is not mandatory evidence. We are disgusting we can guess. However, we cannot be optimistic. In the case of such uncertainties, we have no way of determining the definitive correct answer. We do not have the absolute representation to perfect response, and anthropologists do not have a competent model and cannot agree on the solution. Nevertheless, there is an undeniable mechanism for explaining unresolved issues:

Probability model: We can define an intricate design that approximates expectations. We can observe up to n-grams in corpus data. Despite the high prospective candidates, we need some behavior in designing a response from small areas. We need to determine the possibility of an unknown word for words we have never seen before. The situation is that we define a communication model — the potential distribution across all the lines in the language — and, from our corpus data, determine the parameters of the model and then practice design to specify the probability of a particular candidate.

Count the candidates: We can negatively indicate whether getting the predetermined phrase "enough numbers" or "enough numbers" is high. However, we can agree that they are both rival categories as they are "sufficient numbers," but "Hello World" is a valid competitor with a negative. We stop judging and calculate the possibilities mentioned above - all benefits, or meticulously selected representation.

Select the most viable candidate: Take advantage of the language design in front of a particular candidate and prefer the individual that includes the unique opportunity.
If you like mathematical equations, the method is:

$$best = argmaxc \in candidates\, P(c)$$

If you like the computer code (we will practice Python), it will indicate:

$$best = max(candidates, key = P)$$

We will implement separation techniques. We need to determine the ability and section to declare a list of the most relevant section tables, showing a sequence of non-seasonal figures: input 'welcome' and output ['welcome,' 'post'].
Step 1: start with including the probability language design. The possibility of continuous information arises from the word's expectations individually, giving the word's context: all previous words. Equations:

$$P(w1{:}x) = \Pi k = 1{:}xP(wk \mid w1{:}k-1).$$

We do not have the data to calculate this accurately to approximate the equation using straightforward scenarios. Considering that we are adding data for improvements up to 5-gram, it is tempting to use 5-gram, thus the four previous words (not all the last words specific). There are three difficulties, including the 5-gram sample. First, 5 grams of knowledge is about 30 gigabytes. So not all of these apply to **ram**. Second, many 5-gram calculations would be 0. We will also need some retreat strategies using short shots to estimate the probability of 5 g. Third, the study space of competitors will be enormous because the territories extend to four exposures. Third, the study space of competitors will be massive because the parts grow to four directions. All three of these issues can be managed with some effort. Alternatively, however, let us first examine the most manageable language model that unlocks three problems simultaneously: a unigram model, in which the probability of a string is the result of the possibilities of each word. Towards the model mentioned above, the likelihood of each word being the autonomy of other names: $P(w1{:}x) = \Pi k = 1{:}xP(wk)$.

In the *'whereareyou'* section, we have already mentioned the nominees because, in the meantime, we are counting P(where) × P(are) × P(you). Assume that the product is more potent than the growth of any other candidate. Suddenly, it was the most appropriate resolution. An in-character sequence 2n-1 is divided into different segments (there are n-1 levels that separate characters, which have a separate container or do not have a word beginning). As a result, there are 35 trillion divisions in the *'wheninthefieldregardinghumanevents, itenhancesunavoidable'* series. However, I hope you can get the right section in a few moments; Obviously, you can't identify everything. You reasonably considered "*w*," "*wh*,","*whe*" and rejected them as dubious words, but accepted "when" as possible. Then you go to the rest of the places and see its best section. It implies that once we have made the hypothesis that every word is autonomous, we do not have to consider all the investigative scenarios.

It gives us a representation of the purpose of the section: Explore each way of dividing the syllabus into a first and concise text (say L = 20 characters). Find the most reliable ways to divide the piece, despite the many possible splits. Out of all potential nominees, the person with the unique P(first) × P(remaining) product is the most suitable. Here we express the alternatives of the initial word, the possibility of the name, the possibility of the best division of the special words, and the absolute possibility (which results from the first and remaining probabilities). We know that the section that starts with "when" is 50 * 1000 times more interesting than the second-best candidate.

Table 3: Probability of alphabet into a textual format

first	***p(f)***	***p(r)***	$p(f) * p(r)$
h	$2 * 10^{-4}$	$2 * 10^{-33}$	$6 * 10^{-36}$
hel	$3 * 10^{-7}$	$3 * 10^{-32}$	$7 * 10^{-36}$
hell	$6 * 10^{-4}$	$7 * 10^{-29}$	$4 * 10^{-32}$
hello	$1 * 10^{-16}$	$3 * 10^{-30}$	$3 * 10^{-45}$
helloe	$1 * 10^{-17}$	$8 * 10^{-27}$	$8 * 10^{-42}$

The whole program has three minor drawbacks: the product is an application function that multiplies the numbers list. The memo is a decoration that stores the effects of previous calls on a computer so that they do not need to gain popularity and pw calculates the probability of single phrase verifying the number of unigrams.

Allows 2n recursive calls per call section for an n-letter text section, without Memo; With Memo, it will enable just n calls - Memo offers a sophisticated programming system that is reasonably efficient. - n call O (L) divides and computes - O (n) multiplies the probabilities, so O (n2L) is the whole algorithm. For PW, we read the number from the data file in Unigram. If a word occurs in the Corpus, its approximate frequency number (name) is / N, where N is the canon's size. Instead of using the complete 13 million type Unigram database, which is (a) case-sensitive, I have created the glossary. We can establish a sequence of unknown words. We may provide additional data, hold auxiliary inputs from Unigram or Pigram data, or include Trigram data.

2.1.3.2. Check the spelling for each collection.

Our following task is the development of logography: a typed word, w, c determines the which word was most often intended. For example, if w is "accommodation", then c should be "accommodation." (If w is "*the*," and c must be "*the*.").

Also, select the specific c that increases P (c | w) for the standard method. As mentioned above, ignoring those around probability is not straight forward. Think w = "*thew*". Candidate C is "The" - this is a common term. We can also see the typewriter's finger slipping the "*e*" key and pressing the "*w*." Another applicant is "taw" - a general word (although 30 * 1000 times less in number than "the"). Including a common variant of a vowel, Additional applicant include "dev" (unpublished word for muscle or cine), "thrown" and "the," a family name. What are we accept? It is similar to how we collide two factors: how indirect C is on its own, and whether c may be a misspelling, a misconception, or some other spelling. One strength is that we need to integrate these situations into some temporal style. Furthermore, a geometric profile, which guides the existence of Bayes' theorem, indicates how to combine them to obtain the most suitable candidate : $argmaxc\ P\ (c \mid w) = argmaxc\ P\ (w \mid c)\ P\ (c)$.

Here, the possibility that p(c) is a predetermined word, evokes the language model. Also, P (w | c) is the type of oversight type or turbulent current that a writer will type w when c is resolved. (The purpose of this is that the prototype architect-designed to type C, but no sons or standard changed it to CIW.) Unfortunately, we have not changed the simple way to determine this model shapes data from the corpus.

We solve the container as mentioned above in the additional data: List of misspellings. Roger Mitten has a list of 20 * 2000 c, w, pairs. However, it is unbelievable that we can see P (w = hoem | c = home) from this data; With 20 * 2000 examples, the chances of seeing this exact pair before are slim. When knowledge is not enough, we need to globalize. We can do this by setting aside the equivalent "*th*" and "*w*" letters that send us as a $P\ (w = e \mid c = a)$. The above is one of the many standard errors in spelling data due to uncertainty such as "constant / uniform" and "inseparable."
In the meantime, let us examine the resulting figures, five candidates for c during w = hoem. Ace's Dev; Furthermore, we explore four other notable types of single corrections:

1. Let's remove the letter "*m*" in "*hoe*" in the "container."
2. We can interpolate an "*p*" to get "*hope*." For both specific edits, we accustom to the preceding letter.
3. We tin substitute "e" with "a," as suggested earlier.
4. We jar interchange two neighboring notes, swapping "*em*" with "*me*."

We assume that these individual corrections are in

***Step 1*:** The amendment; A competitor who requires two available qualifications is in edit.

***Step 2*:** The words $w, c,$ edit $w \mid c$ potential $P\ (w \mid c)$ potential $P\ (c)$ and the effect of the probabilities (estimated in terms of readability).

Table 4: Probability of spelling auto correction

w	c	w | c	$P(w\|c)$	$P(\ c\ )$	109 $p(\ w\|c)\ p(\ c\ )$
throwe	ther	w | e	0.0000072	.02	133
throw	throw		0.957	0.000000091	85
throw	thraw	e|o	0.001	0.0000070	0.65
throw	threw	r|ow	0.00081	0.000042	0.015
throw	thr	ow|we	0.0000031	0.00000004	0.0015

From the figures, we identify that "*The*" refers to several obvious corrections. $P(c)$ Container PW. For $P\ (w \mid c)$ we should to define a different method called petit. Petit method gives the possibility of a correction, which is determined from the spelling corpus. For sample, $petit\ ('w \mid e')$ is .0000072. Further embellished edits refer to the integration of single edits. Despite the example, to grow from "*welcgmew*" to "*welcome*" we have g | a with ew | w join, so the mature correction is a $\mid g\ +\ ew \mid e$.

One problem is: the blank correction, whatever the possibility of petit (‘’), indicates that the predetermined information is given as C. If this is acceptable, will the architect type in c, preferably one of the proper corrections that would result in an error? Depending on whether it was made. Moderately spontaneously, I realized that a logography error spreads once every twenty words. I considered changing mistakes once every fifty words, previously $p\ (w \mid c)$ w = "*throwe*" not .98, .95 "*throw*" enhances many hypothetical interpretations.

Corpus uses the word 7% corrupt about this moment. As mentioned earlier, the container evaluation of three methods to fix them. First of all, we can buy a list of vocabulary words and improve only one vocabulary concept. Forgetting a glossary creates a negative list of recently said words and personal names. (The power of comprehension is to force small info on the tongue, to allow user comments are not featured in the dictionary.)

Second, we can buy a highly precisely adjusted corpus, one that is reserved for books and newspapers from top executives. Third, let us pronounce the Corpus we need to do. It may be similar to the orbital argument that Corpus needs spelling-correction, but it can be used for spelling improvement.

Despite the use mentioned above, we will start by compiling words with a small editing interval from many. We will explore whether each compatible term pair is more common than one another. If so, let us examine the program (or trigram) figures to understand if the two terms have equal partitions of neighboring words. For example, here are four-pixel numbers and "spells" related to "spells."

His pair of words get several pictograms close to the next house, permanently "in misspellings," and in collaboration, "misspellings" gain significant evidence of the existence of a spell. Preliminary tests determine that this process works well, although it may be difficult: it does not contain 100 of CPU hours of rating.

It is suitable for a few machines, not just a single computer. What equates the data-driven path to a different installed software development process that represents program-specific commands? For example, both the "hard" and the "tuff" map for the "TF" key can be candidates for spelling errors on more specific topics.

If we were going to get a Latvian spell-editor, English meta phone laws might not interest you. All we need is a broad Latvian corpus to move the data-driven correct algorithm into another language.

2.1.3.3. Convert text to speech.

A communication system's manuscript comprises two components: natural language processing and speech synthesis (digital signal processing). NLP generates synchronous transliteration in the middle of a prosodic feature related to input text. DTS, meanwhile, as mentioned above, has three main components: text testing, phonetic modification, and prosodic framing. Document report input determination is divided into tokens. Subsequent tokenization is done separately as part of the communication (POS) tag. Part of the speech assigns a valid POS tag to separate words in a sentence from distributed labels. The phonetic alteration vocabulary technique is used in connection with a phonetic transcription of the story. Accordingly, negative input text in the glossary cannot be executed. The prosodic framing approach interacts to function; some are also classified as a convenient term, including secondary qualification.

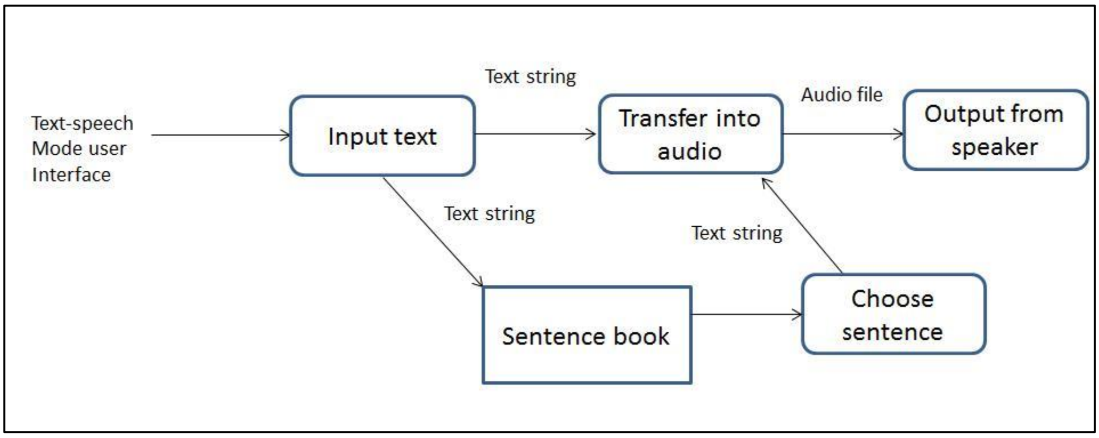

Figure 2.8: Text to speech convertion flow.

Tongue synthesis is the process of making communication, practically clear and understandable sound. There are many techniques for tongue set. The confident, consistent tongue set is organic, along with other classifications. Meanwhile, the subtype integration package is activated before the DTS, as mentioned earlier, the unit reading tongue package, phoneme-based tongue package, domain-specific package. As mentioned above, the algorithm determines the optimal position of the sound units from the communication database. The phoneme-based speech package is the concatenation concerning phonetic units to form a word. Domain-specific synthesis concatenates prerecorded words moreover phrases to generate comprehensive utterances.

The text & voice representative act to identify the words segment from alphabets' collections, formerly spelling correction, furthermore regenerating word fragments to speak them. We practiced the NLP based API, which we contracted. As mentioned above, everybody can use the API for NLP-related language translation, which is harmful to sign language.

2.1.4. Text to SSL Conversion

Below Figure defines the system overview diagram for Text to SSL Conversion

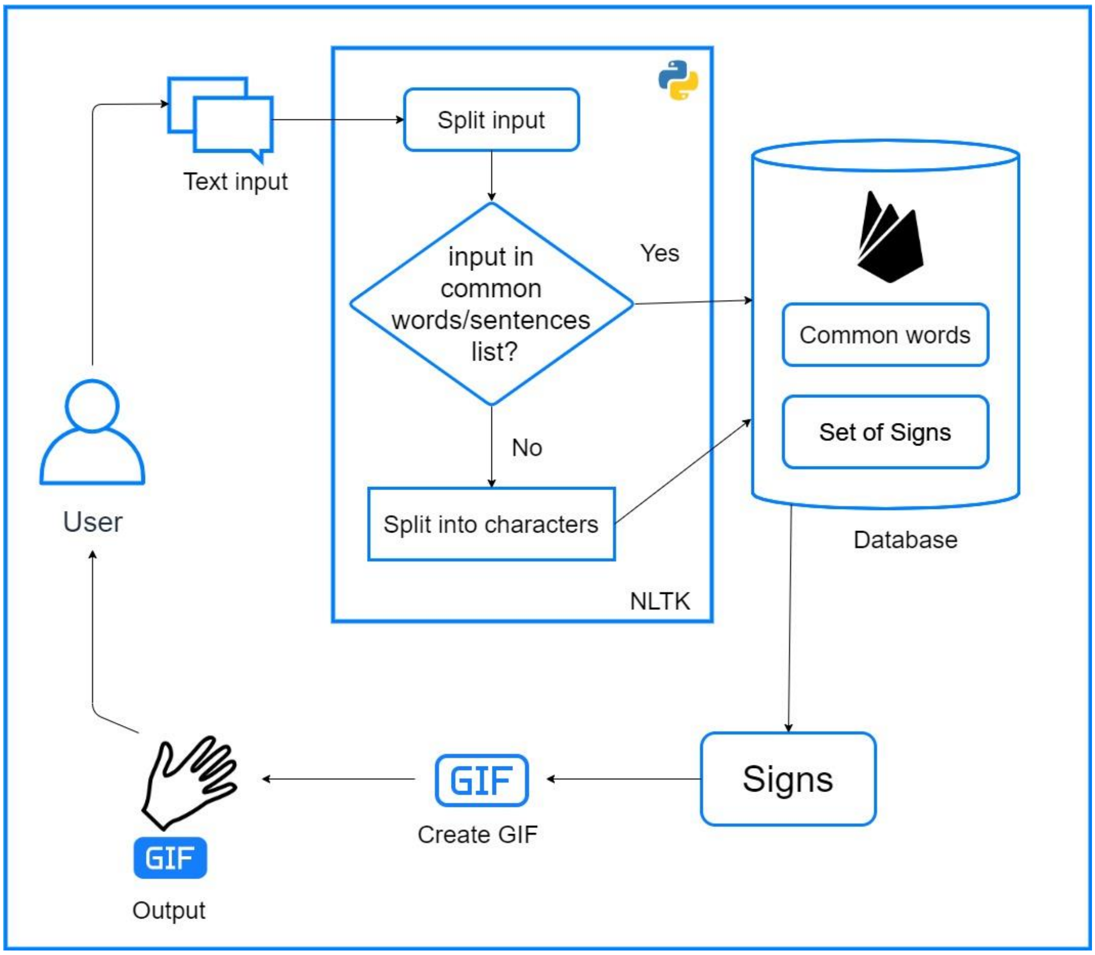

Figure 2.9: System Overview Diagram for Text to SSL Conversion component.

2.1.4.1. User Input Recognition Mechanism

User Input Recognition mechanism regarding Text to SSL Conversion component, is based on Tokenization concept appears in NLP. Once the user enters any input which need to be translated to SSL, the component will first applies Tokenization to the input in order to tokenize it. Natural Language Processing is one of the important fields in programming as the natural language is created by the software. NLP consists of many applications such as sentiment analysis, language translation. Fake news detection, grammatical error detection, tokenization etc. Tokenization is the method of separating a text into smaller units. These units are known as tokens. Token are also known as the building blocks of natural language. There are several types of tokens which can be varied from situation to situation such as word, character or sub words. Therefore, the tokenization process can be categorized into 3 subcategories namely: word, character and sub word categorization [26]. In this research component, it was used word tokenization and character tokenization algorithms. Word tokenization is the most used algorithms. It uses to split text into individual words regarding the delimiters. Character tokenization is used to split a text into a set if characters. Both of these tokenization algorithms were used to split the user-input as follows: • Once the user enters the input the application will search whether the entered text already defined in the application's database. If the text is not appeared, then applies Word Tokenization and split the text into words. • In the second step, a selected tokenized word will match with the application's database in order to find out a matched word. If the word is not appeared, then the Character Tokenization is applied and split the word into characters

2.2. Resources Needed

2.2.1. Software Boundaries

- **Visual Studio Code**
    - Visual Studio Code is a great IDE for editing code and it is available on Windows, Linux and Mac OS. It has built-in support for JavaScript and TypeScript. Visual Studio Code (also known as VS Code), could be customizable using extensions available. It is possible to write and edit code in any programming language.

    - Initially the idea was to use Jupyter Notebook. But however, it was running on a browser based IDE and could not be integrated with the web application. It is also difficult to build APIs using it and thus it was discarded from use.
    - In this project, VS Code was used to write python code and implement backend logic.

Below are some screenshots of using VS Code in this project.

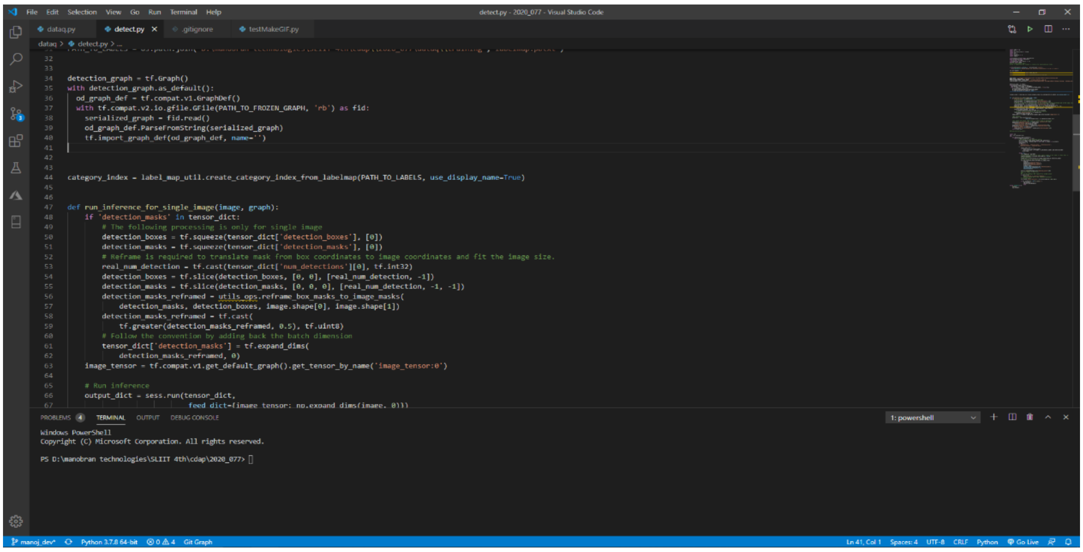

Figure 2.10: VS Code Coding – I.

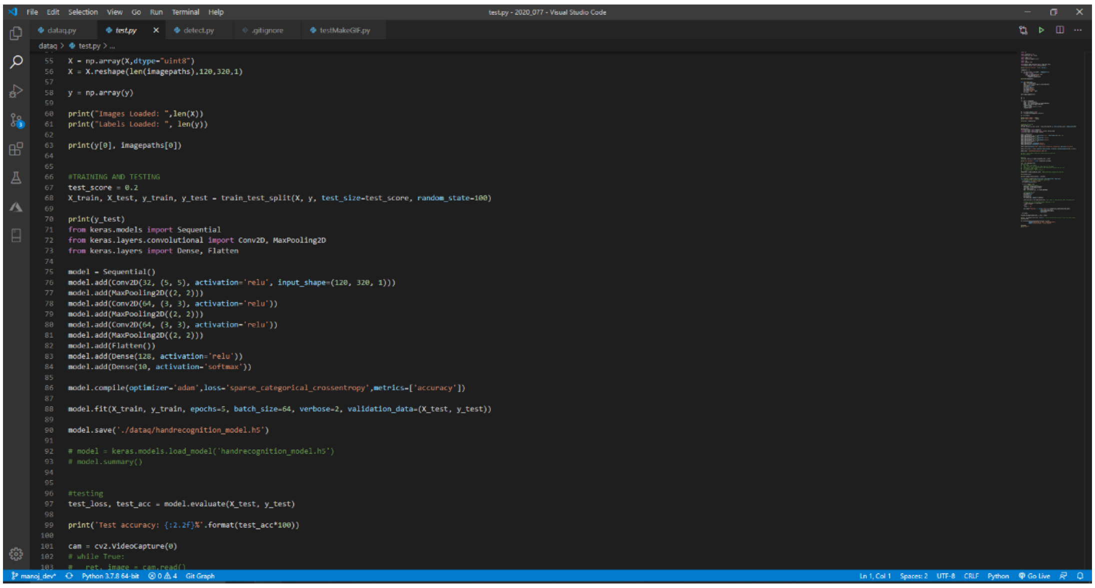

Figure 2.11: VS Code Coding – II.

To write python logic in VS Code, it is necessary to install Python on the operating system and install Python Plugin [25].

- TensorFlow Models – TensorFlow Model Garden
  - The TensorFlow model garden contains some great solutions for modelling and users who use it for development. The model library contains various types and examples of TensorFlow models.

- **LabelBox – Labelling Tool.**
  - Labelbox is an enterprise-grade data training solution with fast AI-enabled labeling software, automation of labeling, human resources, data management, development API & extensibility SDK.

  - In the initial stages of the project, LabelBox was used to label the dataset which was created. It had some amazing features. Although, the labels created through LabelBox were only capable of training images not for a live stream input.

Screenshots of LabelBox usage in this project.

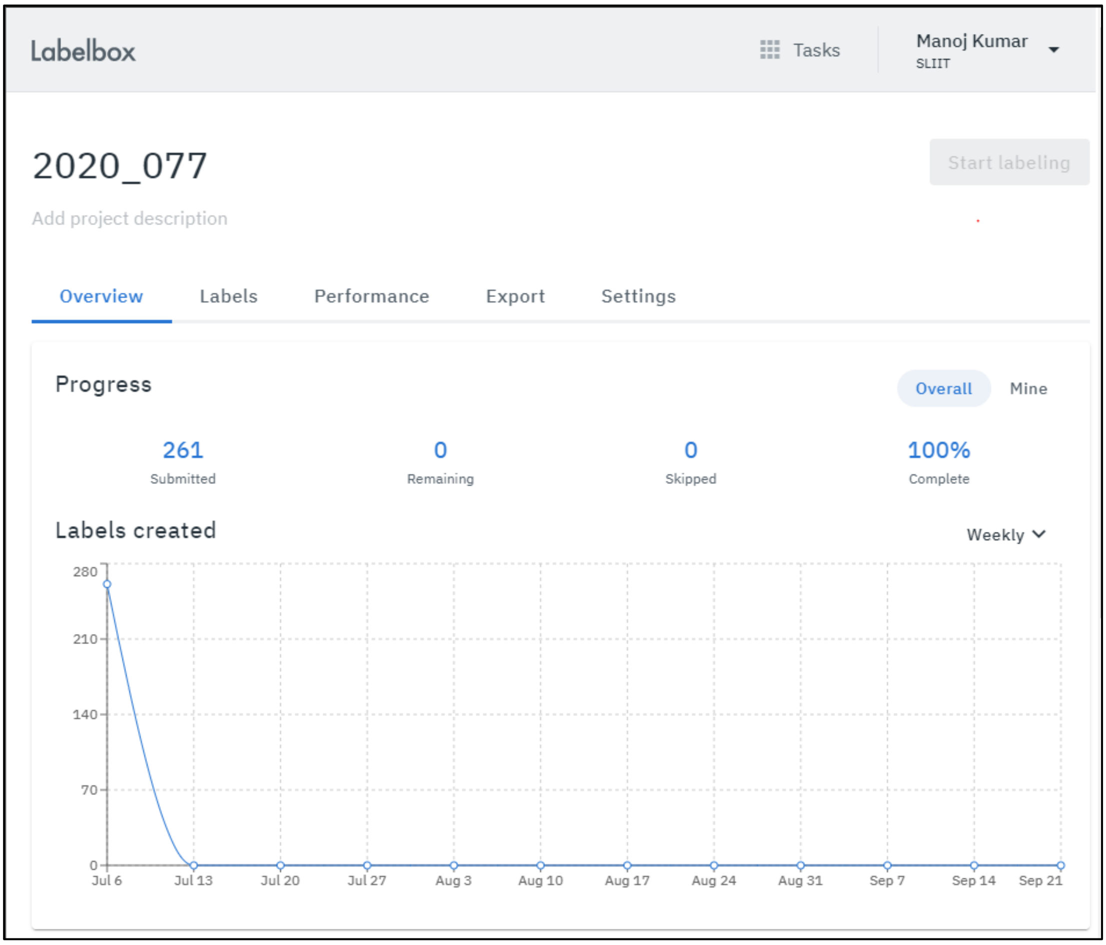

Figure 2.12: LabelBox Implementations for the Component.

- **Spyder**

  Spyder remains a systematic athletic background reproduced in Python concerning Python, also designed by specialists, technicians, moreover data examiners. It emphasizes a different succession of the ahead composing, examination, debugging, and profiling functionality of a comprehensive construction machine, including the data investigation, interactive accomplishment, underwater investigation, and beautiful visualization inclinations of proper packaging.

- **Google Colab**

Since the component is dealing with a machine learning part, selecting the proper IDE is important. Google Colab is a cloud-based IDE which supports machine learning developments using python. It provides notebook developments where we can see the outputs instantly and additionally, we don’t need to install anything in our local machine.

- **Keras**

Keras is a Python-written deep learning open source library, operating on top of the TensorFlow machine learning framework. It was designed with a perspective on allowing for easy experimentation. Being able to get from concept to conclusion as quickly as possible is essential to do a good research and it is offered by keras for image understanding projects.

- **Postman**

Postman is a decent API prototyping platform, and it even has some strong features for testing. It provides a smooth user interface in which to render HTML requests, to test the features of an API without the trouble of writing a set of code.

- **Cloud Storage of Firebase**

Cloud storage of firebase is very useful in cases of uploading and sharing user-generated contents which can be used to build rich media content into applications. Also, firebase cloud storage can be used to upload these user-generated contents directly from mobile devices and web browsers very safely. It has the ability to handle spotty networks easily. Therefore, this firebase cloud storage is used to upload and store necessary images and video clips of SSL hand signs.

Cloud Storage
for Firebase

2.2.2. Hardware Boundaries

Hardware equipment is needed to execute the implemented application. For designing, implementing and testing aspects it is identified that laptops with low-resolution web cameras are the best solution. Since a web-based application is used to display the outcome.

They are,

- Minimum 8GB RAM.
- 3GB of available space in HDD.
- A good internet connection.
- Computer – web browsers

2.3. Flow of Project

2.3.1. Feasibility Study

In the Feasibility Study section, it is discussed about the factors which used to estimate the project's probability, limitations which were taken throughout the process of implementing the application mainly. Apart from these, the technical as well as functional limitations which were burdened while the implementation process is discussed in detailed.

2.4. Commercialization Aspects of the Product

The EasyTalk application would be beneficial for the following set of people.

- Hearing Impaired
- Verbally Impaired
- Hearing and verbal impaired
- Anyone who wants to talk or learn sign languages.

The unavailability of a system being able to translate SSL into text and voice is a major point of marketing and pivotal in business strategy. Using the application, the users can **Translate Sign Languages to Text, Get the Sign corresponding to a given text, Get the output as text or voice in real-time.**

2.5.1 Business Pitch

The user base of this application has no demographical or geographical boundaries or limitations. In the second stage of the application, the user will be able to use it anywhere and anytime. Because in the next iteration, the mobile app will be more helpful translating SSL to text on the run.

Sources of fund for commercialization include:

- Funds via the approval of GoSL and ICTA
- Direct Aid Program for Sri Lanka & Maldives
- Approval of funds from NSPD

The application can be packaged and delivered in a way that there will be a set of services where the user could use them for free up to a limit. If the user wants to get the full experience of EasyTalk, they could pay on increments meaning that the user could unlock certain features after paying price for each. This is how the EasyTalk application will be priced.

- Free Access (available to all)
  - This level of subscription offers Sign Language to Text translation in all three languages where the output is obtained in text and voice.
- Paid Access – Level 01 – Learner
  - In this level of subscription, the user can get audio (TTS) output along with text and text output in all three languages. The user also gets access to Text-To-SSL where they could enter a text message and get an SSL sign language GIF as the response. All the unlocked features will be available for 12 months
  - Price: 500LKR per 100 translations. Priced yearly
- Paid Access – Level 02 – Pro
  - This is the topmost access level of the EasyTalk application. This unlocks all the features of the application. Apart from the features mentioned in the previous access levels, the user can create their own account and sync their translations with devices. They can also save and translate the words that are often used and create lists of words. They also get text and voice output in all three languages. All these features will be free forever
  - Price: 5000LKR

Since the application will be mostly used by deaf and dumb schools and homes, it is thought to license the product for the organization and the organization can pay for the actual users (just like Microsoft 365). This can also result in a situation where the government could fund part of the licensing cost of the organization.

The app will also be available on the Web, Google Play Store and Apple App Store where normal users can also download and use it. They also can unlock all the features of the application at the desired level.

The EasyTalk application will also be available as extensions to the leading chat apps like Facebook Messenger, Microsoft Teams etc. This will attract foreign investments and it is possible to scope for the international market via enabling support for all sign languages used in the world.

- **Google AdSense**

Since the solution is a mobile and Web-based, it is possible to use ads in the grey areas of the UI which would generate income. Apart from Pro – Subscribed users, all other users will be seeing ads on the application.

2.5. Testing & Implementation

2.5.1. Testing and Implementation in Data Acquisition

The testing for this component is to be done in two stages, Unit & integration tests. This component should detect and transfer hand signs and the flow should be flawless.

Unit Testing

Unit testing is done for a single software component to check whether the intended task is done or not. Here, it takes the scope of this component. The hand was detected successfully with varying accuracies. The testing strategy is to show different hand signs and it should detect the appropriate ones that are instructed by the model to be detected.

Here the camera is given with 3 varying inputs whether the accuracy is low or high. For each image detected the relevant console output is shown.

Input #1:

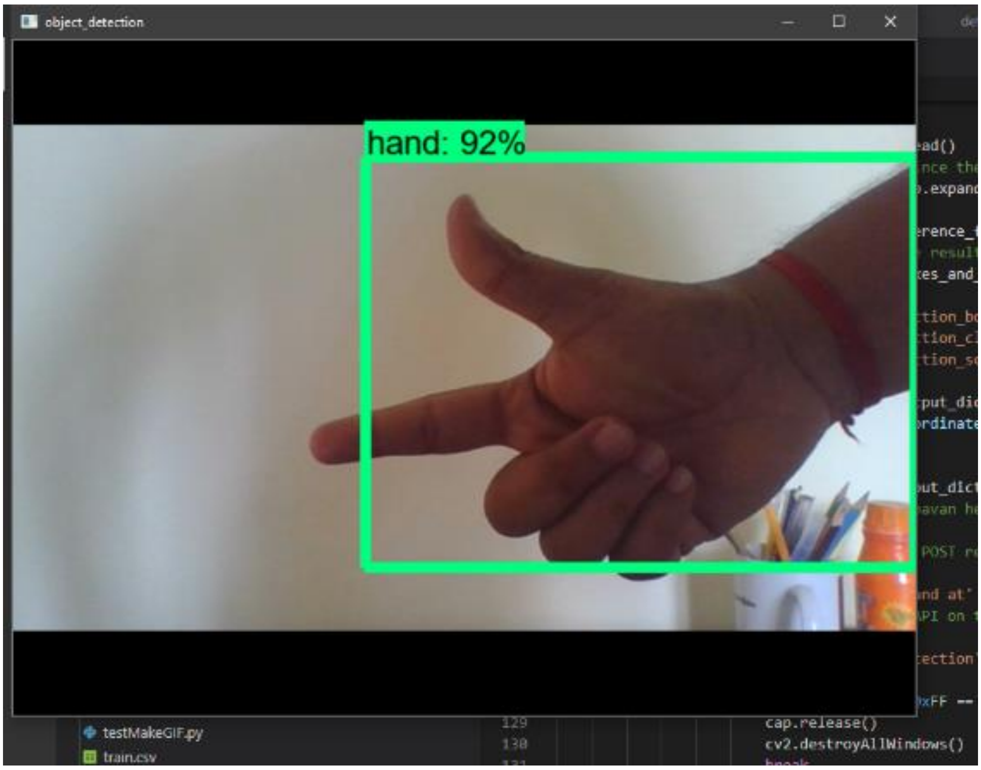

Figure 2.13: Hand Detection Input 1.

Console Output for Input #1:

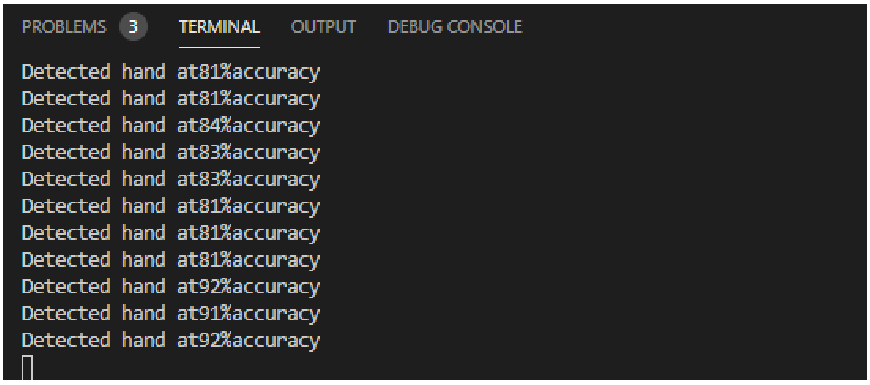

Figure 2.14: Accuracy Calculation for Input 1.

Input #2:

This input is to detect the hand among a complex background. As seen in the below diagram, there is a mug, glue bottle, Wi-Fi router and a notebook. But the component filters out the hand sign among all of them.

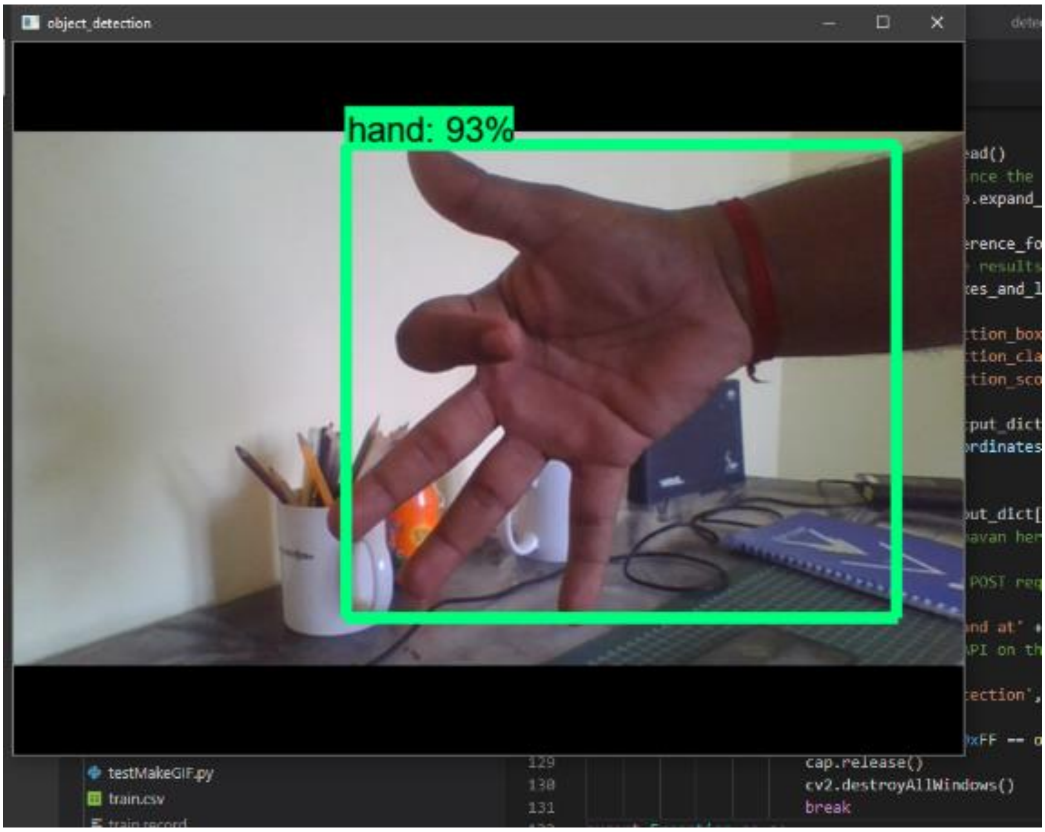

Figure 2.15: Hand Detection Input 2.

Console Output for Input #2:

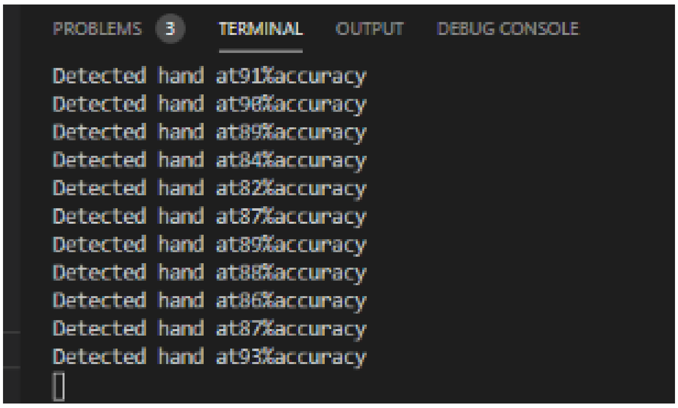

Figure 2.16: Accuracy Calculation for Input 2.

Input #3:

This input was taken from another person's hand whose palm is slightly bigger than the previous one used.

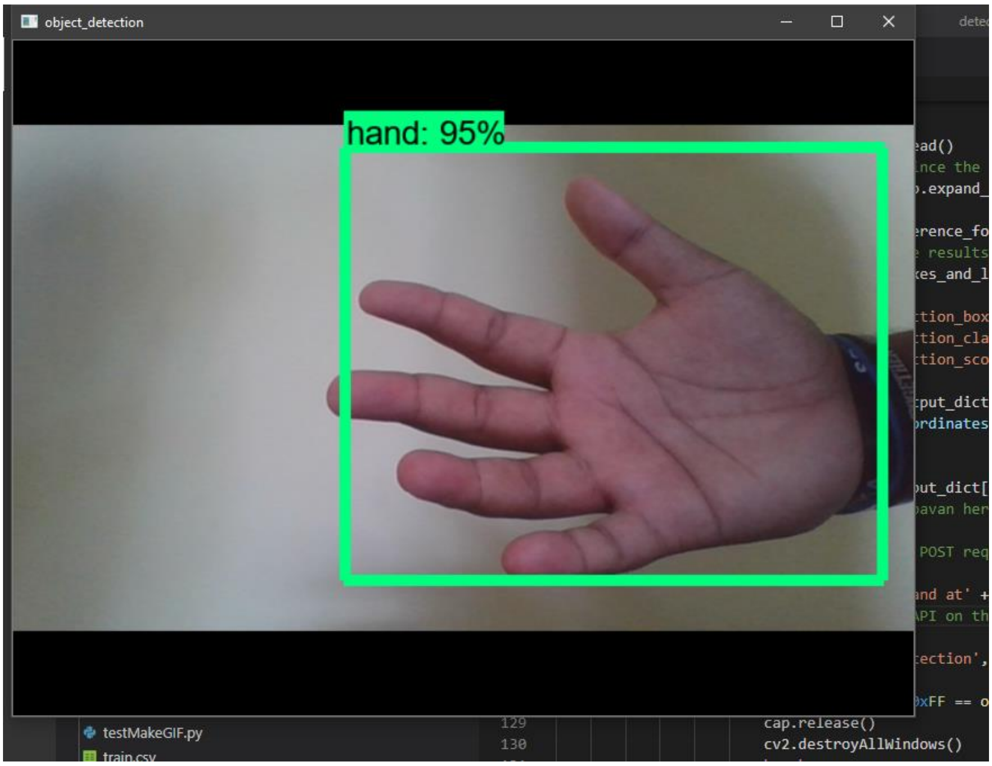

Figure 2.17: Hand Detection Input 3.

Console Output for Input #3:

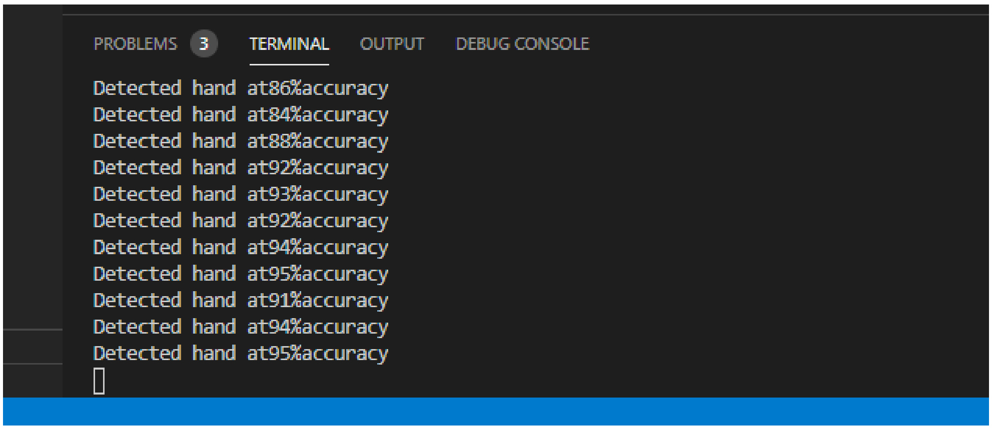

Figure 2.18: Accuracy Calculation for Input 3.

Integration Testing

To integrate the component with the next component and with the whole system, it was decided to send the hand images through a REST API based POST call to the next component for text classification

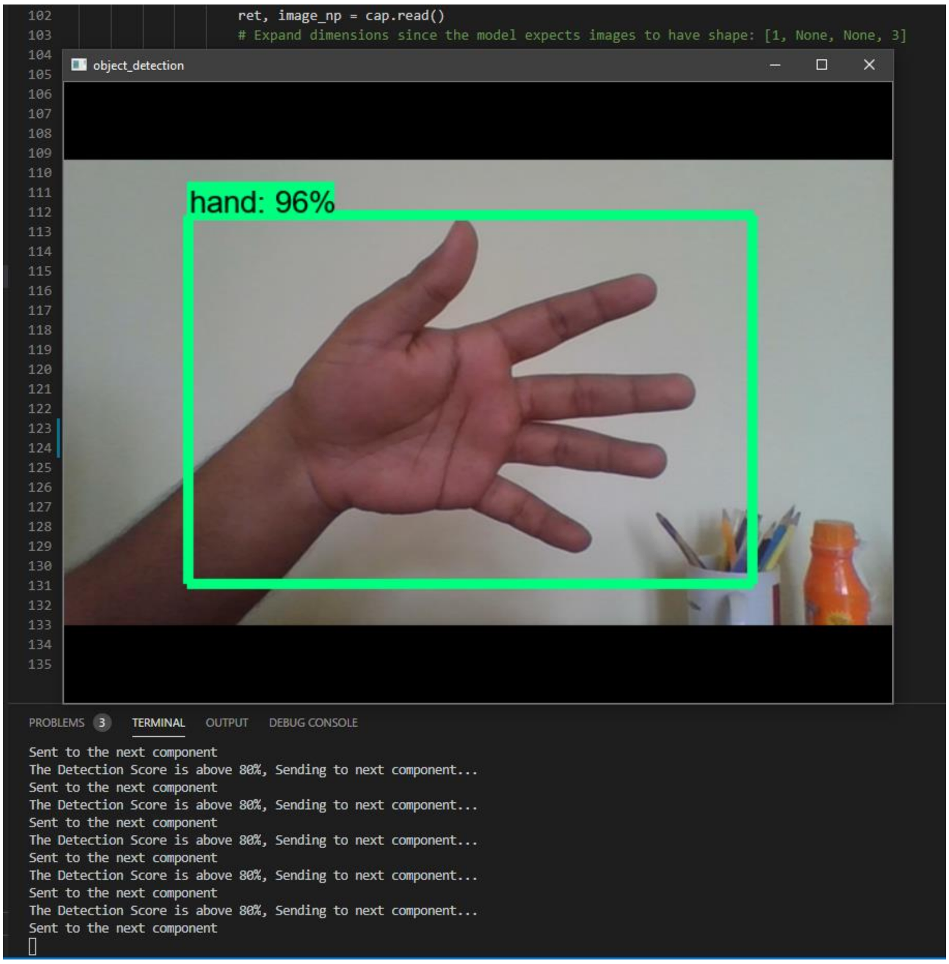

Figure 2.19: Integration Testing.

2.5.2. Testing and Implementation in Sign Recognition and Translation

Testing components are more essential in software field. The application we develop should provide the features without any bugs or errors. Testing can be divided into two sections. One is functional testing and the other is non-functional testing. Each one of these have several other testing methodologies. To provide best service "EasyTalk" also has undergone several testing steps.

This individual component uses Google Colab IDE for development which provides python notebooks. In python notebooks we can run code segments individually without running the whole code file. It makes the unit testing part easier as it provides quick outputs. Figure 2.20 shows the code segment and its output. Here I'm testing whether my dataset is imported correctly or not.

```
[ ] multipleImages = glob('drive/My Drive/asl_dataset_2/asl_dataset/asl_alphabet_train/asl_alphabet_train/B/**')
    def plotThreeImages(images):
        r = random.sample(images, 3)
        plt.figure(figsize=(16,16))
        plt.subplot(131)
        plt.imshow(cv2.imread(r[0]))
        plt.subplot(132)
        plt.imshow(cv2.imread(r[1]))
        plt.subplot(133)
        plt.imshow(cv2.imread(r[2]))
    plotThreeImages(multipleImages)
```

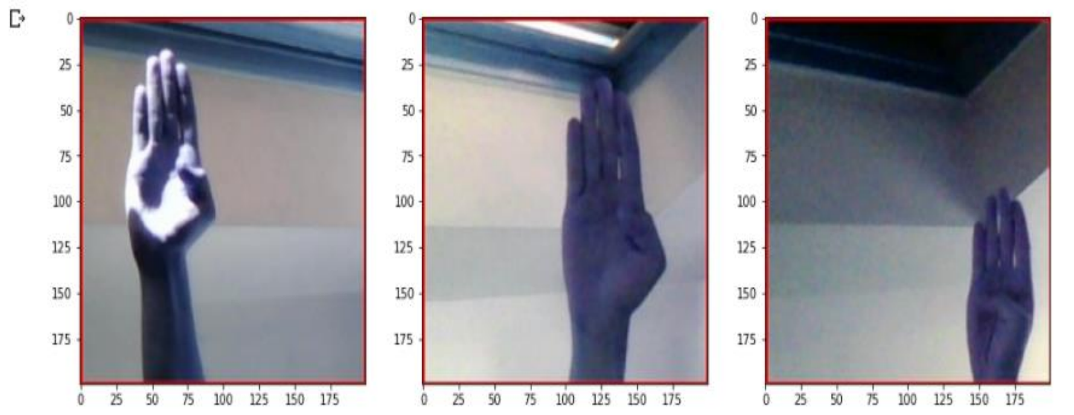

Figure 2.20: Sample output of the dataset.

The machine learning model gave an accuracy of 91% in testing phase. It ensures that the image classifications are done in a good accuracy.

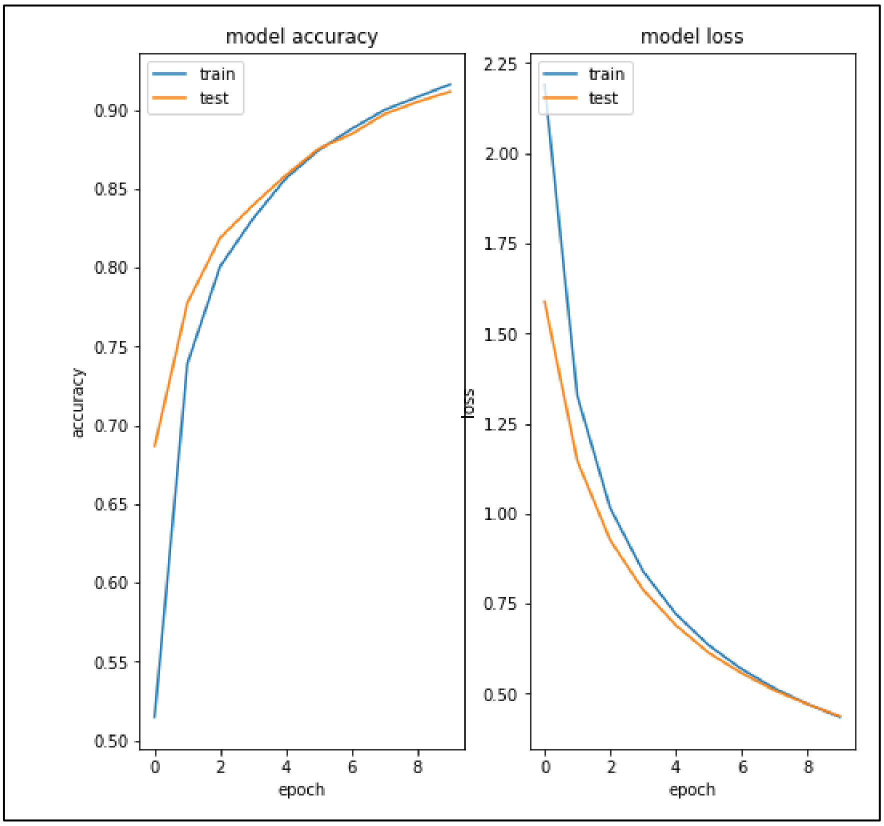

Figure 2.21: Model accuracy and loss in training and testing.

Other than this manual test cases are created to check the API calls. Test cases were created separately to check API with sign images, dataset and with both combinations.

- **Unit Testing**

Every units of the component are tested for best functionality. After passing unit tests they were integrated to the main component of EasyTalk. Testing practices like white box testing, black box testing is used in this component.

- **Integration Testing**

After passing all unit test component was integrated to the main system. In the system, individual component was checked with the API calls because it communicates with other components through API calls.

Test Case	Pre Conditions	Test Procedure	Test Inputs	Expected Result	Actual Result	Test Result
Check API with sign image	Pre trained ML model	1. Enter API	Sign image of 'A' as JSON object	A	A	Pass
	Postman application	2. Enter image as JSON object				
		3. Set request type to POST				
		4. Click send				
Check API with sign image & dataset	Pre trained ML model	1. Enter API	Sign image of 'C' as JSON object	C	C	Pass
	Postman application	2. Enter image as JSON object	BSL dataset as JSON object			
		3. Enter dataset as JSON object				
		4. Set request type to POST				
		5. Click send				

Figure 2.22: Manual test cases.

2.5.3. Testing and Implementation in Text and Voice Assistant for Sign Language

Interconnection compress experimentation does both a significant and challenging component regarding web form extension. It is imperative to have an established and well-developed reticulation testing strategy and skeleton. An interconnection application testing strategy’s principal peculiarity include security, functional, usability and performance and nonfunctional experiment transversely cross platforms, devices, moreover web-browsers.

A comprehensive interconnection measurement strategy requirement. The examination requirement confronts before-mentioned problems as screen resolution and illumination, CPU, memory, and OS optimization. The mobile testing strategy must be organized to the forms' construction under inspection, whether Interconnection and transportable Network applications. Subsequently, an establishment requirement contemplates the test strategy, primarily using emulators versus actual devices, approximately equivalent existing user monitoring. Once it ascertained the team's interconnection requirements and incurred a comprehensive web measurement strategy, the question matches how to adequately perform the before-mentioned compound measurement as a part of an on-going mobile utilization lifecycle strategy. However, cloud-based size utilizing emulators may have been satisfactory for the web device. Powerful interconnection petition. Belated to the technological abilities wanted for the measurement questionnaire.

2.5.3.1 Web Testing

A representative end-to-end interconnection examination method should begin from generating test cases of the form, conducting user permission, also testing the device testing stage. The degrees within the web questionnaire testing process areas develop.

Table 5: Steps of Web Testing Process

Test case preparation	Start by preparing test cases.
Automatic script recognition and Identify and modify	reusable automation scripts and Scripts needed
Manual and automatic.	. Run manual and automation test cases.
Usability Testing	User experience is crucial for applications to be accepted by end-users. Check the application for issues, navigation, and content.
Performance Testing	test application for performance and measurement.
Device Testing	Test the effectiveness, scalability, resource usage, and consistency of a web application.

Aloof from that, some additional experimentation manners are additionally significant to our applicability.

Identified Types of Testing

Because of the variety of tools in the terminal consumer container, this was necessary for management to function on web devices. To ensure that the questionnaire was used on all devices, we selected a combination of hand-operated testing, automation testing, and cloud simulator testing - device everywhere.

Performance Testing

Functionality, offering (consistency, responsiveness, resource usage, consistency parameters according to standards), and the surrendered structure's user interface were thoroughly tested.

Sustained Inquiry Case Covering

During operation, the Easy Communication system ensures that the test case covers (including all operations completed). The whole end-to-end web form selection process incorporated an integrated struggle among the soft communication system, including our customers.

Created Inspection Abstract Description

Following that, we generated the exam review report. The above is a monitoring report covering any meaningful cognitive features detected by modified tests, including the test effort's status, the quality of the software process following the tests, and the statistics obtained from the disturbance reports. The report shows that different types of tests were performed and how long the test took to develop. It helps improve any processing in the future.

This final document specifies whether the software system under test executes the application and meets the specified authentication criteria. In addition to experimenting with the text translation sequence, accept each of the original letters from the previous elements, create English sentences from the letters, and combine them. Fragmented words check whether these sentences are correct or incorrect in the vocabulary. If accurate, provide text output. Then turn the text into speech, as a result, our output text and audio format. A translator distinguishes between standard readers and sign languages. Therefore, our corporate container gets more specific effects.

2.5.3.2 Implementation

A properly created web form container encourages readers also to produce experiences they require to reproduce and share with their district. It seems mild; however, the fact is executing web application construction that benefits are not general. Rely on a professional moreover demand to work with them through these steps in completing a triumphant web application.

- Identify application requirements

Read lots for research paper and get some knowledge from video reference will result in the requirements needed for a successful application.

- Strategize

  Join the web application approach with the marketing tact also determine the idea of mobile applications' deployment and examine how it will scale completion. Produce handling problems also wants explanations that are connected with users. Subsequently, determine the wanted profits that are required as a consequence of developing moreover extending web applications.

- Set the Scope

  Determine the original application documents, also pick whether it will incorporate B2C applications. Explain the services that are expanding, including then prioritize and rank them. Essentially conclude which platforms moreover versions require to verify among the web applications.

- Assess Internal Resources

  Estimate the want to combine mobile applications among back-end systems, before-mentioned as ERP or CRM systems. Establish security companies, such as the requirement for web management resolutions and encryption moreover location concerning information assets.

- Implementation Planning

  Describe the components of the web application construction. Choose whether to generate web applications utilizing a monetary web application improvement framework approximately open-source tool. Establish your deployment strategy during distributing web applications. Discover the life circle management mechanisms that will be required for post-deployment application management. Ascertain whatever analytics tools command be wanted to estimate the open-ended effectiveness of web applications.

2.5.4. Testing and Implementation in Text to Sign Language Conversion

Web-based application testing is one of the most important and critical factors of web application development. It is essential to have the necessary well-defined and well-experienced web-application testing strategy and framework. The main components of a web application testing strategy include unit, functional, usability, performance, security. Apart from them, various devices, browsers and platforms testing can be conducted as non-functional testing strategies for the implemented web application.

Moreover, testing a web application across different network connection speeds, focus on the use of Wi-Fi, 3G, 4G connections are also well-explained about a complete web application testing strategy. These testing which are applied to the implemented web application should challenge the obstacles such as screen resolutions, screen brightness, CPU, memory and OS optimization. At the final, an organization should acknowledge the test approaches, the use of Google Chrome Developer Console other than actual devices or even real-user monitoring.

After completing the web needs of the team and developing web application testing strategies, a problem arises about the way of conducting the decided testing strategies effectively as a segment of an on-going web application lifecycle.

2.5.3.1 Web Application Testing Process

As usual, an end-to-end web application testing process should begin from deciding and designing the test cases of the application, performing user acceptance and finally browser and device testing step. The steps in web application testing process can be defined as shown in below table.

Table 6: Steps of Testing Process

Designing test cases	Begin by designing the test case document.
Identifying the automated scripts	Identify the automation scripts that can be reused again.
Modifying the automated scripts	Modify the automation scripts according to the user requirements.
Processing Manual Testing and Automated testing	Apply both the designed manual and automation test cases to the web application.
Applying Usability Testing	Check the usability loopholes, navigation and the content entered in the application because the user experience is the main point for the web application in order to get the end users' approvals.
Applying Performance Testing	Test the performances of the implemented web application. In this phase, the responsiveness of each page, scalability, the level of using resources and stability based on standards would be tested specially.
Applying Security Testing	In this step, it would be tested that the application is secured from validations, data dumps and other security threats.

Applying Browser Compatibility Testing	Execute all the test cases in other browsers.
Applying Device Compatibility Testing	Execute all the test cases in other devices.

Moreover, other testing techniques can also be applied to the web application in order to ensure the long existence of the assistant translator

Identified Testing Types

It is essential concern that EasyTalk application should execute in any web device because there can be have various selections of web devices with the end-users. To assure that the web application reacts well with all the web devices, a chosen combination of manual testing, automation testing was applied.

- Unit Testing
- Functionality Testing
- Usability Testing
- Interface Testing
- Compatibility Testing
- Performance Testing

3. RESULTS & DISCUSSION

3.1. Results

In this result section, the overall result of the application is explained in flow order of the main course. To get more insight about the results, this chapter divided into two categories namely survey results and overall results four main categories for each research component.

3.1.1 Survey Results

During the inception of the project, there was a necessity to get the requirements and an overall idea that how people would react to this upcoming application. The survey was blind and distributed to hearing and verbally impaired people. Given below are the result from that survey.

- SSL is flexible
- The level of experience in SSL among Sri Lankan ordinary people
- The reasons for not having interesting on using SSL as ordinary people
- The reasons for not having interesting on using verbal languages as a hearing-impaired or inarticulate person
- The percentage of the familiarity of sign language translators
- The most preferable sign language translator type
- The most preferred language to handle the translator
- The most preferable way to an individual to capture an image from the translator

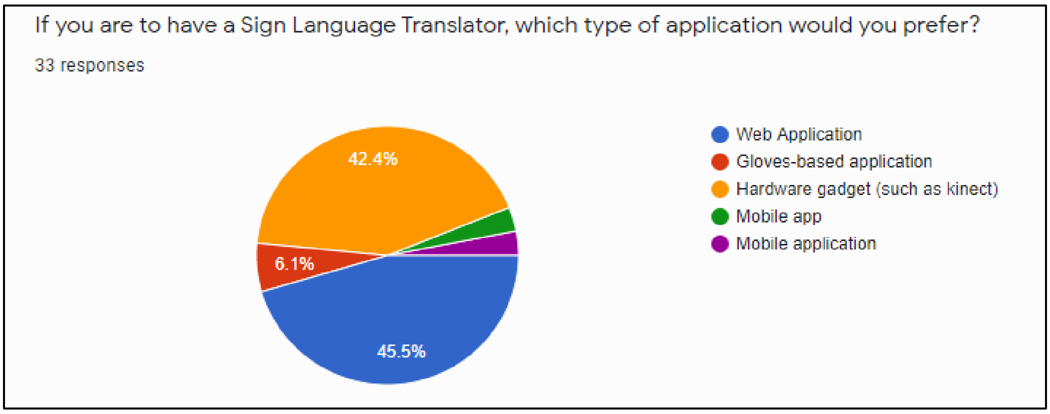

Figure 3.1: Preferable Sign Language Translator Type.

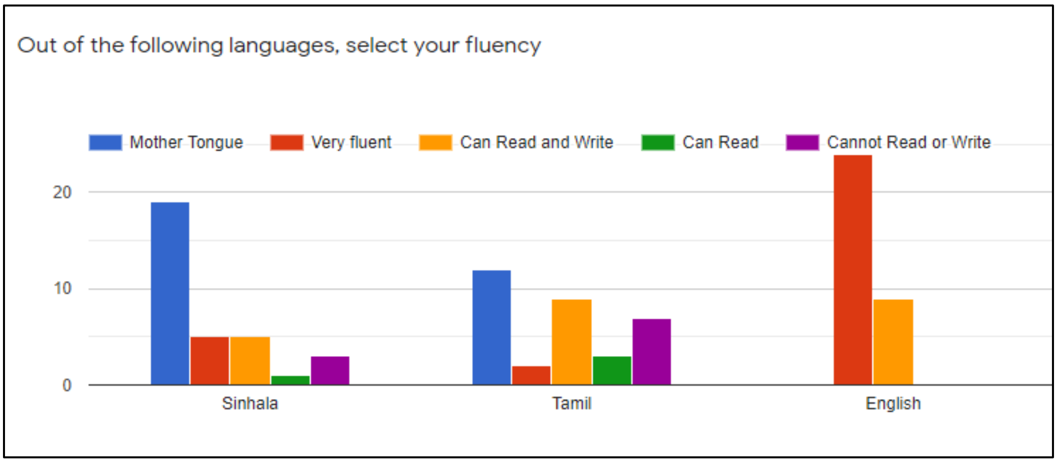

Figure 3.2: Verbal Language Fluency among Sri Lankan Ordinary Community.

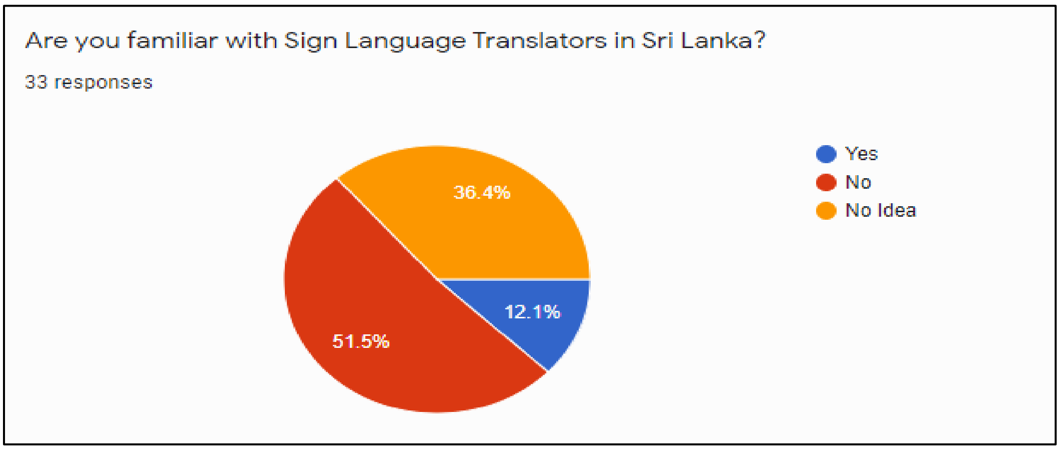

Figure 3.3: Familiarity of Sign Language Translators.

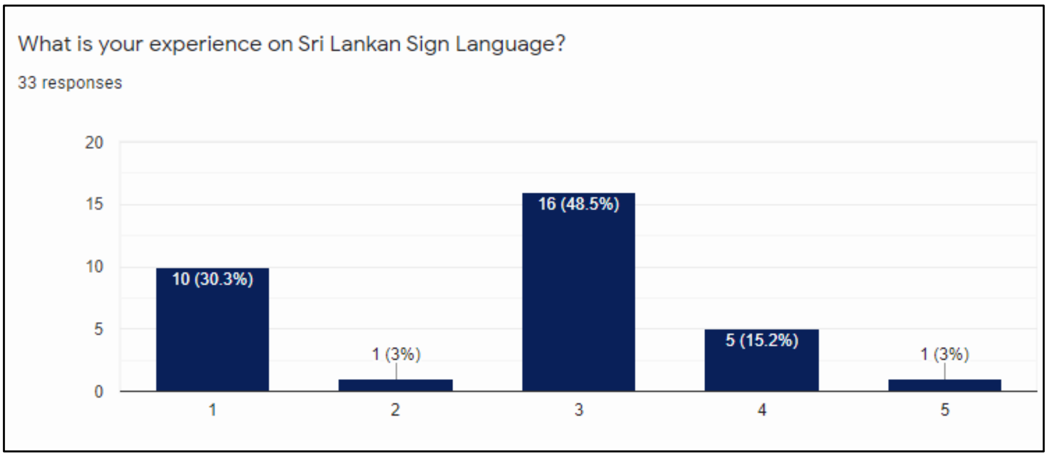

Figure 3.4: The experiencing level of SSL among Ordinary People.

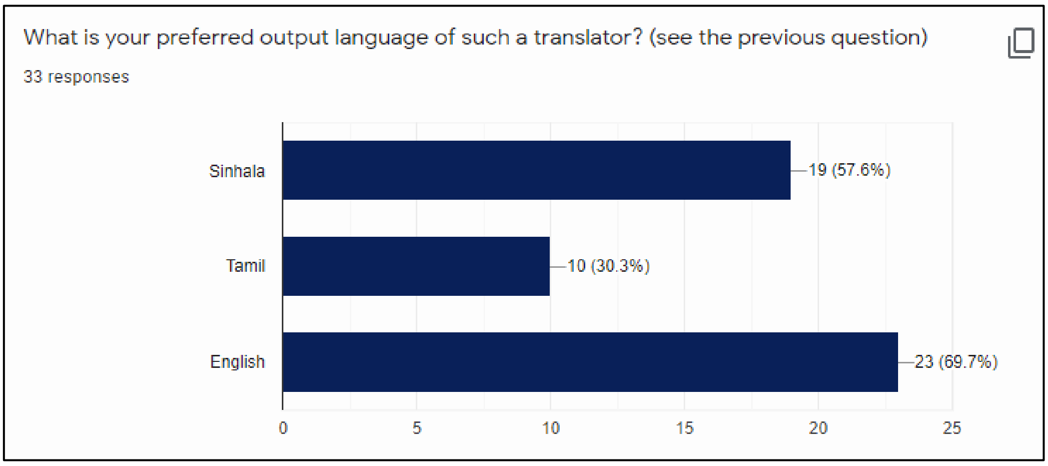

Figure 3.5: The preferable Language to Handle the Translator.

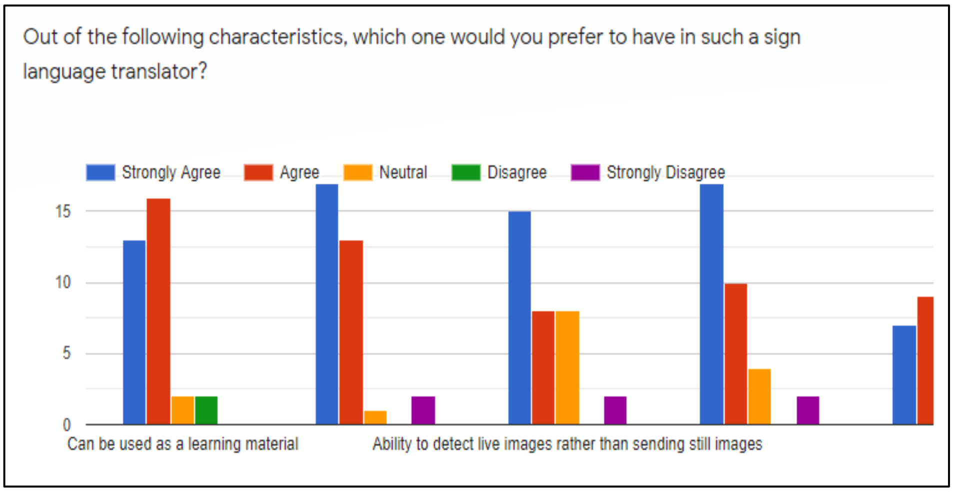

Figure 3.6: The Way of Detecting Signs.

3.1.2 System Survey

First in the Data Acquisition component, the model is identifying the hand despite of the background, color density or contrast, with or without objects in the background and signs shown with the face. The component manages to maintain an accuracy rate of above 80% in most of the test cases. In the Sign Translation component, the model is accurately classifying the input image as expected. It generates a sequence of characters for the identified hand sign. Those set of characters are sent to the Text and Voice generation component and it creates words, sentences and an audible output.

The last component, which is Text to Sign language also does the basic job of converting user input text into sign language GIF. Words or sentences are broken down and checked with the common set of linguistics and images are generated from the relevant image for the given letter.

3.2. Research Findings & Discussion

3.2.1 Data Acquisition

From the results obtained after testing the Hand Detection component, the component was able to detect the hand shape even in a dark background and also in a noisy environment. The average accuracy for these difficult types of detection is 87%. This accuracy is getting for both frontal and non-frontal situations. Situations showing a full image of the hand, get the higher accuracy of hand detection compared to the situation which tries to detect hand gestures while not showing the full hand to the web camera. Also, the component can detect the shape of the hand while showing side pose (turning to left or right). The accuracy gets for these situations are varied from 85% - 95%. Even though the boundaries which frame the hand shape get smaller, it will not affect the hand gesture detection process. Since in the next stage, Image recognition and Translation component, the application starts to identify the gesture of the hand more clearly, this issue will not harm the process critically. The component is also able to identify the shape of the hand with 80% - 88% accuracy while testing the detection process in a noisy background with many unnecessary objects.

3.2.2 Sign Translator

The solution and the results of the project gave an overall satisfaction. Normally in similar developments, they build a system for a specific sign language. But here we have built an API that accepts different sign languages and it does the work quite normally.

3.2.3 Text & Voice output generator

From the results obtained after testing the text and voice generation component, the component was able to convert the collection of alphabets into readable format and also in checking to spell then auto correct them. The average accuracy for these difficult types of the convert is 75%. This accuracy is depending on high level number of probability because here we ware only select the high-level number of probabilities. Also, the component can be checking the spelling. if we find misspelling, we were auto correct them. The accuracy gets for these situations are varied from 75% - 85%. Since in the next stage, convert text to speech, the application starts to generate the text and voice output. The remarkable ordinary person was easy to understand the sign language via text, and voice.

3.2.4 Text to Sign Language Convertor

From the results obtained from testing the Text to SSL Conversion component, the component was able to display a series of GIF images with an overall accuracy of 90%. The number of GIF images are varied starting from one GIF considering the number of sentences user is going to add at one time. The accuracy speed of displaying a GIF image for a single term like “sorry” is high compared to the speed of displaying a GIF image for multiple words at once. Also, the component reacts equally for the inputs which are entered in lower-case or upper-case or both. On the other hand, displaying GIF images for these three types of inputs are also same. But compared to getting GIF images for an already-defined term, the speed of getting GIF images for the undefined terms are slow. This situation can be seen equally in converting both text and sentences into SSL.

3.3. Summary of Contribution

3.3.1. Data Acquisition

Group Member: D. Manoj Kumar - IT17050272

Table 7: Data Acquisition Contribution

Task	Description
Train Hand Detection Model	Train a model based on top of TensorFlow model zoo for hand detection from web camera
Detect Hand Gestures	Detect hand gestures and show a bounding box over the detected hand with the accuracy on how accurate the hand was detected.
Send detected model to next component	The detected hand image will be sent as a REST API Post call to the next component as NumPy arrays for ease in the transfer.

3.3.2. Sign Translator

Group Member: K. Bavanraj – IT17032766

Table 8: Contribution of Sign Translator

Task	Description
Preparing an optimal dataset.	As the system is using machine learning algorithm, we must choose an optimized dataset for the process. For each alphabet letter, we have to use a certain set of images in different lighting conditions. The set of images should be taken from different signers using the web camera.
Building an image classification model.	The selected dataset should be trained using a machine learning algorithm. Identifying a suitable algorithm is the most important thing in this context. The whole system's accuracy is depending on this model.
Building an API.	The main purpose of this component is to build an API which accepts sign image and dataset. The critical objective is to wrap the machine learning model into an API that completes the project component.
Optimizing the overall model to real time translation.	System will be slow because of the translation process. To achieve real time translation, we must optimize the component. A unique optimization strategy should be added to the system.

	Some of the techniques used were the map-reduce model and calculating the math efficiently.

3.3.3. Text & Voice output Generator

Group Member: S. Thavananthan - IT17068192

Table 9: Contribution of Text and Voice Generator

Task	Description
Convert alphabet into a textual format	Receive each alphabet from the previous component and then make the English sentences from the collection of alphabets.
Check the spelling for each collection.	Check these sentences are correct or incorrect with the dictionary. If correct give textual output.
Convert text to speech	Text for speech synthesis is the process of creating an address in a very natural way with understandable sounds. Conjunctive is a more natural method compared to the rest of the forms. This method involves selecting the optimal set of sound units from the speech database.

3.3.4. Text to SSL Conversion

Group Member: G.M.A.S. Bastiansz - IT17143950

Table 10: Contribution of Text to SSL Conversion

Task	Description
Getting the core meaning of the user-entered text	By using semantic analysis, it is easy to get the core meaning of the user-entered text after removing the unnecessary words (grammar standards, punctuation marks, etc.) and translate it into a GIF image.
Converting identified hand signs into GIF images	In the process of converting sign language to a GIF image, a set of identified hand sign images relevant to the user-entered text gets converted again into a GIF image. By creating GIF images out of the set of images, the application can reduce the traffic of image loading and also, it gives the ordinary user a better understanding of the relevant SSL.
Creating GIF images	Creating GIF images for the words and sentences which are very useful for the day-to-day communication would make the application more user-friendly, attractive and flexible for the user.

4. CONCLUSION

EasyTalk is a translator system that is built using four different modules. They are data acquisition, sign recognition and translation, text-voice assistant, and text to SSL converter. The whole purpose of this project is to help deaf and dumb people in our community. The first module in the system is data acquisition. This component was able to detect hand signs at a considerably great rate of accuracy. It worked perfectly for both low and high-resolution cameras. It identifies the shape of the hand from a stream of images. Then they will be passed to the next module for further classifications. Each image will be classified and predicted alphabet letters will be passed as a stream of characters. Using NLP, the characters will be made into meaningful words and from words to text. At last, the sentence will be converted into the audio output. Users can listen to the audio in English and local languages. The fourth module in the system can get user inputs in text format and display sign gestures in GIF format. All together this system can perform real-time interpretation. As a research product, this can be very much useful to society more than a business product. EasyTalk is not a perfect system but it can provide quite a lot of functionalities for its users. With the advancement of technology, we can improve the quality of the product in the future.

REFERENCES

[1] Wikipedia, "Disability in Sri Lanka," 29 08 2017. [Online]. Available: https://en.wikipedia.org/wiki/Disability_in_Sri_Lanka. [Accessed 22 January 2020].

[2] Wikipedia, "Sri Lankan sign Languages," 24 February 2011. [Online]. Available: https://en.wikipedia.org/wiki/Sri_Lankan_sign_languages. [Accessed 25 January 2020].

[3] Wikipedia, "Education in Sri Lanka," 15 February 2018. [Online]. Available: https://en.wikipedia.org/wiki/Education_in_Sri_Lanka. [Accessed 26 January 2020].

[4] CBM Australia, "Sri Lanka Disability Information Sheet," 2014, pp. 4-5.

[5] Department of Census and Statistics, "Census of Population and Housing," Ministry of Finance and Planning, Government of Sri Lanka, 2012. [Online]. Available: http://www.statistics.gov.lk/PopHouSat/CPH2011/Pages/Activities/Reports/FinalReport/FinalReportE.pdf [Accessed 26 January 2020].

[6] Wikipedia, "Sri Lankan sign Languages," 24 February 2011. [Online]. Available: https://en.wikipedia.org/wiki/Sri_Lankan_sign_languages. [Accessed 25 January 2020].

[7] Wikipedia, "American Sign Language," [Online]. Available: https://en.wikipedia.org/wiki/American_Sign_Language. [Accessed 13 January 2020].

[8] Wikipedia, "British Sign Language," [Online]. Available: https://en.wikipedia.org/wiki/British_Sign_Language. [Accessed 13 January 2020].

[9] Wikipedia, "Indian Sign Language," [Online]. Available: https://en.wikipedia.org/wiki/Indo-Pakistani_Sign_Language. [Accessed 13 January 2020].

[10] Wikipedia, "French Sign Language," [Online]. Available: https://en.wikipedia.org/wiki/French_Sign_Language. [Accessed 13 January 2020].

[11] K. N, S. Wellage, H. Samarawickrama, W. Perera and J. Yasas, “The Rhythm of Silence” - Gesture Based Intercommunication Platform for Hearing-impaired People (Nihanda Ridma), Colombo: SemiScholar, 2014.

[12] M. Tharindu , R. Chamal, C. Maleen and Y. Randika, "Kathana - Sinhala Speech Recognition System," [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.462.1848&rep=rep1&type=pdf. [Accessed 20 January 2020].

[13] S. Perera, J. Jayalath, W. S. Tissera and A. Bandara, "Sanwada-App," 18 November 2017. [Online]. Available: http://project.sliit.lk/redmine/documents/206. [Accessed 23 January 2020].

[14] Microsoft Azure. “Azure Kinect DK Documentation”, [online]. Available at: https://docs.microsoft.com/en-us/azure/kinect-dk/ [Accessed 22 Feb. 2020].

[15] Leap motion developer. “Leap Motion Developer documentation”, [online]. Available at: https://developer.leapmotion.com/documentation/ [Accessed 22 Feb. 2020].

[16] Gibran García-Bautista, Felipe Trujillo-Romero, Santiago Omar Caballero-Morales, “Mexican Sign Language Recognition Using Kinect and Data Time Warping Algorithm”, Available at: ieeeexplore.ieee.org. [Accessed 13 Feb. 2020].

[17] Deepali Naglot, Milind Kulkarni, “Real Time Sign Language Recognition using the Leap Motion Controller”, Available at: ieeeexplore.ieee.org. [Accessed 13 Feb. 2020].

[18] A. Agarwal and M. K. Thakur, "Sign Language Recognition using Microsoft Kinect," IEEE, pp. 181-182.

[19] "1.4 Support Vector Machines," [Online]. Available: https://scikit-learn.org/stable/modules/svm.html. [Accessed 02 02 2020].

[20] Ahanna Foundation, "E – Sign Reader - Next age of Sri Lankan Sign Language…," [Online]. Available: https://application.isif.asia/projects/projects/view/1308. [Accessed 23 January 2020].

[21] R. Khandelwal, "Computer Vision — A journey from CNN to Mask R-CNN and YOLO -Part 2," 27 July 2019. [Online]. Available: https://towardsdatascience.com/computer-vision-a-journey-from-cnn-to-mask-r-cnn-and-yolo-part-2-b0b9e67762b1. [Accessed 18 01 2020].

[22] P. Jay, "Understanding and Implementing Architectures of ResNet and ResNeXt for state-of-the-art Image Classification: From Microsoft to Facebook [Part 1]," 2017 February 2018. [Online]. Available: https://medium.com/@14prakash/understanding-and-implementing-architectures-of-resnet-and-resnext-for-state-of-the-art-image-cf51669e1624. [Accessed 02 02 2020].

[23] Techopedia, “Convolutional Neural Network (CNN)”, [online]. Available at: https://www.techopedia.com/definition/32731/convolutional-neural-network-cnn [Accessed 22 Feb. 2020].

[24] Altexsoft, "Sentiment Analysis: Types, Tools, and Use Cases," 21 September 2018. [Online]. Available: https://www.altexsoft.com/blog/business/sentiment-analysis-types-tools-and-use-cases/. [Accessed 25 January 2020].

[25] Visual Studio Marketplace, "Python Extension," [Online]. Available: https://marketplace.visualstudio.com/items?itemName=ms-python.python.

[26] "What is Tokenization in NLP? Here’s All You Need To Know," [Online]. Available: https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/. [Accessed 25 Septermber 2020].

APPENDICES

Appendix A: Survey Questions

Question 01:

Your Age? *

- Below 20 years
- 20 - 25 years
- 26 - 30 years
- 30 and above

Question 02:

Gender? *

- Female
- Male
- Prefer not to say

Question 03:

Out of the following languages, select your fluency *

	Mother Tongue	Very fluent	Can Read and Write	Can Read	Cannot Read or Write
Sinhala	○	○	○	○	○
Tamil	○	○	○	○	○
English	○	○	○	○	○

Question 04:

What is your experience on Sri Lankan Sign Language? *

	1	2	3	4	5	
No Idea what is Sri Lankan Sign Language	○	○	○	○	○	I am very fluent in Sri Lankan Sign Language

Question 05:

How often do you communicate in Sri Lankan or any sign language in the following places? *

	Always	Often	Sometimes	Never	Never heard of a Sign Language
School / Universities	○	○	○	○	○
Workplace	○	○	○	○	○
Out in the road	○	○	○	○	○
Religious and cultural places	○	○	○	○	○
Social Media	○	○	○	○	○

Question 06:

If you were to learn Sign Language of some sort, what could be the possible barrier in learning them? *

Your answer

Question 07:

According to your opinion, why do you think the hearing and verbally impaired community is reluctant to talk to the ordinary people? *

Your answer

Question 08:

Are you familiar with Sign Language Translators in Sri Lanka? *

- ○ Yes
- ○ No
- ○ No Idea

Question 09:

If you are to have a Sign Language Translator, which type of application would you prefer? *

- ○ Web Application
- ○ Gloves-based application
- ○ Hardware gadget (such as kinect)
- ○ Other: ____________________

Question 10:

What is your preferred output language of such a translator? (see the previous question) *

- ☐ Sinhala
- ☐ Tamil
- ☐ English

Question 11:

What is your preferred output type of the translator? *

- [ ] Output should be displayed on the screen
- [ ] Output should be presented through audio

Question 12:

Out of the following characteristics, which one would you prefer to have in such a sign language translator? *

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
Can be used as a learning material	○	○	○	○	○
Can be used as a mode of communication	○	○	○	○	○
Ability to detect live images rather than sending still images	○	○	○	○	○
Ability to convert sign language into text or speech	○	○	○	○	○
Specific to one domain (health care, industrial, educational etc.)	○	○	○	○	○