EASYTALK: A TRANSLATOR FOR SRI LANKAN SIGN LANGUAGE

Final (Draft) Report

K. Bavanraj
IT17032766

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

K. Bavanraj
IT17032766

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 articles or books).

Name	Student ID	Signature
K. Bavanraj	IT17032766	

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

……………………………………..   ……………………..

Signature of the Supervisor: Date:

ACKNOWLEDGEMENT

I 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, I would like to thank Dr Dharshana Kasthurirathna for his continuous encouragement & for providing useful information & tips regarding the research domain.

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

ABSTRACT

The hearing-impaired and the deaf community people use sign language as their basic communication method. These community people are often ignored in the social world because of the communication gap. For a professional world, understanding sign language is important as much as understanding the natural language. Only then, these people can get involved in the professional world. There are many translator applications available to do these translation jobs. But most of them are still in the research state and they take more time to process and translate. If a communication system like this lack in real-time translation, it means the participants are not having a healthy conversation. Our research is mainly focusing on this problem. We are trying to achieve a real-time translation system for Sri Lankan sign language (SSL).

When it comes to implementing a system to translate sign language to text, the process requires a series of images. Then the images should be converted into a meaningful word. The translation must happen in real-time. For that, each component of the translator must do their job in a small amount of time. Machine learning techniques have advanced in recent years. By using ML algorithms, we can shrink a big process into small. So, we are proposing a machine learning solution to recognize and translate signs.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

API – Application Programming Interface.

CNN – Convolutional Neural Network.

DTW – Dynamic Time Wrapping.

FPS – Frames Per Seconds.

GIF – Graphics Interchange Format.

HD – High Definition.

HMM – Hidden Markov Models.

HOG – Histogram of Oriented Gradients.

KNN – Kth Nearest Neighbor algorithm.

LMC – Leap Motion Controller.

MLP – Multilayer Perception algorithm.

SDLC – Software Development Life Cycle.

SVM – Support Vector Machine.

1. INTRODUCTION

1.1. Background

Communication is a basic need for all living creatures. Through communication only people can connect and express their needs and feelings. If people lack in communication skills, they will be ignored by other people. According to the World Health Organization's statistics, over 5% of the world's population has a hearing disability problem and 15% of the population lives with some sort of disability [1][2]. These people should interact with other people casually but in most cases, they have been neglected. They are ignored in schools, job interviews, social life, and in many more cases. It has been acknowledged that the hearing society has misunderstandings about the hearing disabled community [3].

People who have hearing or speaking problems use sign language to communicate with each other. As natural language sign language also has a lot of variations worldwide. Each natural language has a unique signing method, and that method differs from country to country. For example, the signs for the English alphabet in America are not the same as the signs in the United Kingdom. According to the "Language Blog," there are around 135 to 300 unique sign languages available. Below listed are the widely used sign languages,

- American Sign Language (ASL).
- Mexican Sign Language (MSL).
- Chinese Sign Language (CSL).
- French Sign Language (FSL).
- Arabic Sign Language (ArSL).

For each country, there is one or more sign language based on the languages spoken in those countries. However, in most scenarios, the interchangeability is comparatively lower in between sign languages. For instance, a person communicating in British Sign Language might not understand any communications done in Sri Lankan Sign Language. This is a great barrier for local and overseas communications which are based on sign languages. These signs can be represented using one hand or both hands and some use hands with facial expressions. Signs are not only limited to the alphabet but also available for a set of words.

In Srilanka, according to the department of census and statistics, there are almost 45 thousand people are affected by congenital deaf and dumbness [4]. Most of them are learning in special schools and they are teaching the local sign language. These schools teach a unique sign language system for the major 3 languages. Since it’s unique to Srilanka it is commonly called Sri Lankan sign language (SSL). For the Sinhala alphabet, they are using one hand signing system and for the English alphabet, they are using two hands [5]. Figure 1.1 shows the Sri Lankan sign representation for some of the local places.

Figure 1.1: Sri Lankan sign language.

The communication technology has gone so far for the hearing community, but it lacks in the hearing-impaired society. To connect a normal person with a disabled person technology is the only way. But there is no standard tool or technology to help the hearing disabled people. Some of the existing tools are limited to certain features. Some of them use sensors like Kinect which is costly in Sri Lankan context. For Srilanka, we need a translator system that is affordable and available. So, the more economical way is to develop a system using computer vision. By signing in front of a camera we can translate the signs into a natural language and make a normal person understand it. This system can solve the problems of hearing-impaired people in job interviews, presentations, examinations, and in many places. The existing systems like Nihanda, Kathana [6] are not a successful system because they lack real-time translation. Communication must happen fluently, then only it becomes a healthy conversation. By reducing the barriers of disabled people in Srilanka, we can get their help in a productive way to increase our national economy.

There are many researches going in sign language translators. 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 clearly. Commonly used sensors are Kinect and Leap motion controller. Kinect sensors are used to capture 3D depth images [7]. Leap motion controllers use near-infrared rays to track hands and take greyscale images [8]. These high-end sensors can help to reduce the time taken in translations. Figure 1.2 and 1.3 show above mentioned sensors.

Figure 1.2: Kinect sensor.

Figure 1.3: Leap motion controller connected to a laptop.

Let us see some of the past researches where these high-end devices are used.

1.2. Literature Review

- **Mexican sign language recognition using Kinect** [9]

  In this research, they talk about the usage of the Kinect sensor. When a signer shows a sign, the system will store the color, depth, and skeleton tracking information using the sensor. Then using the Dynamic Time Wrapping 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 that how accurate is the Kinect sensor in recognition.

- **Real time sign language recognition using the Leap Motion Controller** [10]

  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 classification part, they have used the Multilayer Perception algorithm which uses the gathered features as input and converts them into a specific alphabet letter. For training, authors have implemented the Backpropagation algorithm. The total system has shown a 96.15% of recognition rate. This high percentage is achieved only because of the quality feature extraction through LMC.

- **A Chinese Sign Language Recognition System Using Leap Motion** [11]
This research was conducted by four Chinese students who build the translator using the Leap motion controller and applying KNN algorithm for classification. They get the 3D sign gesture data using the sensor. The sensor was accurate enough to detect finger movements with millimeter-level precision. The device worked at 120 fps rate which provides the max frames for the feature extraction. For classification, the team has used 250 images for each sign. They get the highest accuracy of 96.8% in the end. However, for some sign letters they get the accuracy less than 80 percent because of the sign similarities.

- **Gesture Recognition Using Kinect for Sign Language Translation** [12]
Students from Jaypee Institute of Information Technology developed a motion detection system using Microsoft Kinect device. The system records the user's both hands and sends a series of frames until the user puts his hands lower than the hip. This sensor uses 3D coordinates of 6 body parts which are hands, two elbows and two shoulder joints. Using the vector mechanism, the features are classified into alphabet letters. The model is trained with ten different gestures and tested with 150 sign gestures. The number of tests passed was 136 and failed was 14. This system can work in different backgrounds and in various lighting conditions. Overall, this system produced a good result than others.

- **User-dependent Sign Language Recognition Using Motion Detection** [13]
In this paper, the authors have discussed two implementation methods for Arabic sign language. One is a sensor implemented gloves base system. For this, they have used DG5-VHand data gloves. The other one is the Polhemus G4 tracker. Two datasets were collected from each sensor. The first dataset

was collected using DG5-VHand gloves. It is implemented with 5 sensors. For each finger, there is a sensor for motion tracking.

Figure 1.4: DG5-VHand data gloves.

The second dataset was collected using the Polhemus G4 motion tracker. It provides measurements in 6 gesture aspects. They are the three Euler angles coordinate and three coordinates of the Cartesian position.

Figure 1.5: Polhemus G4 motion tracker.

Both datasets use 40 sentences from Arabic sign language, which are formed using 80 words. The model was trained and tested with the KNN classifier and with the HMM classifier. The higher recognition rate was achieved for the dataset which is collected using Polhemus G4 tacker with the combination of KNN classifier.

- **Research and Implementation of Sign Language Recognition Method Based on Kinect** [14]

  In this system, researchers have used a Kinect sensor with SVM and HOG algorithms. With Kinect, gestures of 72 words are gathered in a unique 3D sign gesture dataset and test trials are carried out to check the translation process. The trial outcomes show that the application of the HOG and SVM algorithms substantially improves the precision of the Kinect recognition. The model has an accuracy rate of 89.8% which proves that sensors can be useful in detection and classification.

Other than these devices, there are many research works conducted using web camera in different languages. The advancement in artificial intelligence has boost this researches in the last decade. They were supported by machine learning and deep learning algorithms. when it comes to machine learning there are several steps in image feature grouping. They are image preprocessing, extraction in the feature, and classification. Preprocessing is performed in the sense of eliminating an unnecessary factor from the scope. The picture attribute will be derived from the Region of Interest (ROI) and will be defined using a certain process. Let’s see some past studies which have used machine learning with the combination of a web camera in their works.

- **Machine Learning Model for Sign Language Interpretation using Webcam Images** [15]

  This research was done by two Indian students in the IT field. They have built a system for Indian sign language using web camera images. This system records video from the webcam and interprets it into text and voice format. The machine learning model which is used here has been trained with the Haar Cascade classifier. 500 positive and 500 negative image datasets have been used to train the model. Simple sentences like “Hello”, “excuse me” and “Bye” are tested with this model and they gave an average accuracy of 92.68%. Even though it is a good result set, the system only works for the pre-captured videos.

- **Real-Time American Sign Language Recognition Using Skin Segmentation and Image Category Classification with Convolutional Neural Network and Deep Learning** [16]

  This research and development were done by Shadman and the team. They developed the translator using skin segmentation and analysis. Based on color details, they have presented an automated human skin segmentation algorithm. The color space of YCbCr is used in this context because it is usually used in video processing and gives a strong use of RGB color space data to model the color of human skin. The algorithm's output is demonstrated by the calculations carried out on photographs, representing persons of various ethnicities. To remove features from the pictures, the convolutional neural network has been used.

1.3. Research Gap

Sign language-related researches are happening all around the world for many years. The main problem for a translator system is its cost and availability. In Srilanka, there are no proper real time translator applications for Sri Lankan sign language. There are some researches that happened in this field, but most of the outputs are not satisfied by the end-user. The main reason is that they lack real-time translation. When it's come to real time translation, each component should do their work in real time. From this research, we are trying to full fill the gaps in "sign recognition and translation" using a machine learning solution. In existing solutions image analysis and classification takes more time. By using a machine learning solution, we can optimize this in real time. Let's see some Sri Lankan sign language related researches and how they serve the deaf and mute people community.

- **Ahanna**

  This is a web-based application that is mainly focused on Sinhala sign language. Users can learn Sinhala sign language and also follow Buddhism related studies. The main purpose of this application is to help the dumb and deaf community educationally and spiritually [17].

- **Nihanda**

  This system was designed for children who are having vocal and hearing problems. This application has a game where individual signs and pronunciations can be described by videos and pictures. According to the voice inputs, system will demonstrate 2D images. It also detects signs using a leap motion controller and translates them into audio. For children, this game approach can be more effective to motivate themselves.

- **Kathana**

  This system was developed by the researchers at University of Moratuwa. It basically detects voices in Sinhala language and interprets it to a sentence. Interpreted words and sentences will be used as commands, data entries and other speech understanding archives [6].

- **Sanwadha**

  This project is a cross-platform mobile application for the hearing disabled people. It provides an instant chat service for the user. The application gets Sinhala text as input and changes it into Sinhala sign language. The hearing-impaired user on the other side will get it in GIF format. A normal person and a hearing disabled person can make meaningful communication through this application.

These developments have improved the quality and gave some ideas for new researches in this domain. However, these ideas are not in use because they are not providing all the facilities a mute or deaf person need, and they lack some practicality issues. Table 1.1 demonstrates the comparison of our product with the existing researches in this domain.

Table 1.1: Comparison with other researches.

	Application type	Data output	Data input device	Language	Users
Ahanna	Web-based	Visual	HD camera	Sinhala	Deaf community
Nihanda	Stand alone	Visual	Leap motion controller	English	Deaf children
Kathana	Stand alone	Audio	Microphone	Sinhala	Mute community
Sanwada	Mobile	Visual	Keypad	Sinhala	Mute & deaf community
EasyTalk	Web-based	Visual	Web camera	English	Mute & deaf community

2. RESEARCH PROBLEM

Communication technology has developed massively in the last two decades. The use of the internet and related ecosystems (IoT devices) have made day to day life easier and more convenient. But certain community people couldn't join with the flow because of their communication disability. For a human being, communication is the most essential thing to express his needs and feelings. Deaf and mute community people face a lot of problem in their day to day life. They find difficulties in getting vital services such as education, transportation, employment, housing, health and many more. For a normal person, there is a responsibility that resides in him to accept a disabled person's thoughts and concerns. But what happens if he couldn't understand him?

Our purpose of the research is to solve their communication problems and make them involve in the normal society. Deaf and mute people communicate physically and visually. People like this feel shy and awkward to communicate in public places and they simply get neglected. In short, the common problems they face are

- Cannot express their feeling to common people.
- Find difficulties or feel awkward to converse with their community in public.
- Discouraged to be social.
- Cannot communicate from long distance.
- Unable to get or give any support when needed.

To encourage them and to get them involved in the normal society we have developed an interpreter which can solve the communication problems in them. The stakeholders of this system can be both normal and disabled persons. This interpreter can capture the signs of a disabled person and translate it into a sentence. A normal person can learn signs by our GIF feature facility where signs will appear for certain sentences.

2.1. Problem Addressed by Component

Optimizing the recognition and translation process for a real time application is more challenging. The extracted images should be recognized and translated immediately into letters. This is the main problem we are trying to cover through this research component. Existing systems use sensors to detect signs which analyze and process in a small amount of time [9][10]. Other similar applications use external HD cameras which gives a clear image for the translation process. These systems cost more because of the additional sensors and cameras. But in our context, we are aiming to develop a system which is cheap in money and which is available for the consumer all the time. So, we are using the built-in low-resolution laptop cameras. Therefore, there is no need for additional devices, and it is always available. These cameras take low-resolution images which make the translation complicate. With low-resolution images we have to feed more data to the dataset to train machine learning algorithm and it takes more time to process. So, we must take additional steps to optimize these processes. Our solution tries to cover the time optimization part and gives a real time solution where other researches failed to meet.

3. RESEARCH OBJECTIVES

3.1. Main Objectives

The main objective of this component is to construct a translation API which can be used in sign language related software developments. The extracted gesture-images are gathered from the previous component and translated into English alphabet letters. Then it will be passed to the next component with the help of the API. To achieve this overall objective, we must complete some specific objectives. They are,

- Preparing an optimal dataset to create an effective data model.
- Applying a proper machine learning algorithm for translation.
- Optimizing the overall model to real time translating.
- Building an API using the machine learning model.

3.2. Specific Objectives

- **Preparing an optimal dataset to create an effective data model.**

  As the system is using a machine learning algorithm, we must choose an optimized dataset for the process. For each alphabet letter, we have to upload a certain set of images in different lighting conditions. The set of images should be taken from different signers using the web camera. The dataset should contain a large set of images because the more data we feed the more accuracy we get.
- **Applying a proper machine learning algorithm for translation.**

  The selected dataset should be trained using a machine learning algorithm. Identifying a suitable algorithm is the most important thing in this context. The accuracy of the model is depending on the selecting algorithm. There are a lot of self-learning algorithms for image translation, but we need to select an algorithm that gives the most accurate results in real time translation.
- **Optimizing the overall model to real time translating.**

  The system will be slow because of the translation process. To achieve real time translation, we have to optimize the component. A unique optimization strategy should be added to the system. Some of the techniques are vectorizing functions, map-reduce model and calculating the math efficiently.

- **Building an API using the machine learning model.**

  As I mentioned earlier the main purpose of this component is to build a reusable API. The final objective is to wrap the machine learning model into an API that completes the project component. When the API is called it gets the image as a parameter and using the ML model it will predict the alphabet letter and pass it to the next component. The API will be integrated into the main component to serve its purpose.

4. METHODOLOGY

This section describes the way the project is handled. A methodology is an approach to the final product through different stages like requirement gathering, requirement analysis, planning, executing, and testing. The sign translation component in “EasyTalk” application also gone through the same steps. It has a significant research area in real time sign translation using machine learning.

There are some software development methodologies that can be used in the software development life cycle. To build our research project “EasyTalk”, we have used the waterfall model because the application requires more project guidance from the start to finish. It is much easier to control the system in various life cycle stages by using this waterfall approach.

4.1.System Overview

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 4.1 shows the system diagram of the component. The extracted sign images are fetched from before component and will be sent 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 have 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. The 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.

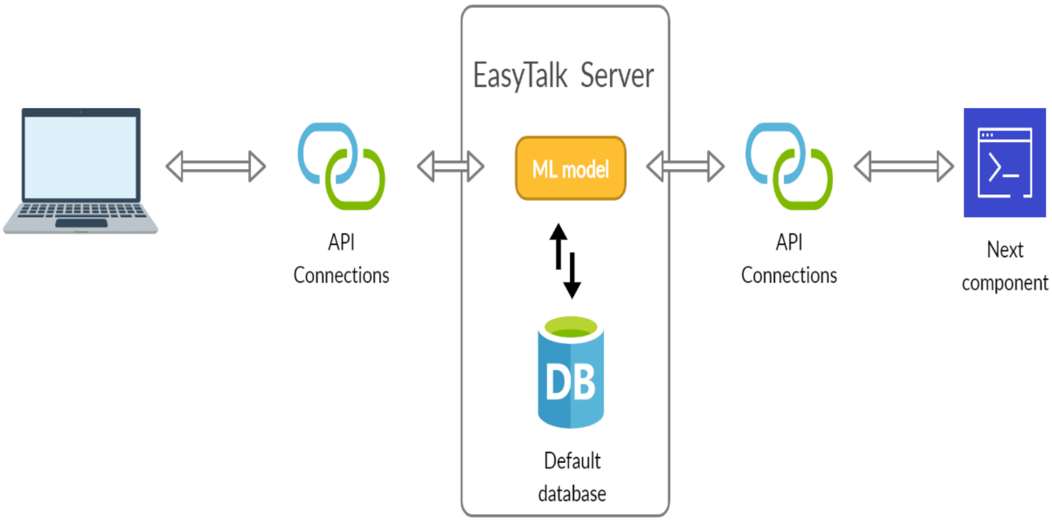


Figure 4.1: Component diagram.

4.1.1. Data Collection

To build this research component we have to go through several steps. The major part of 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.

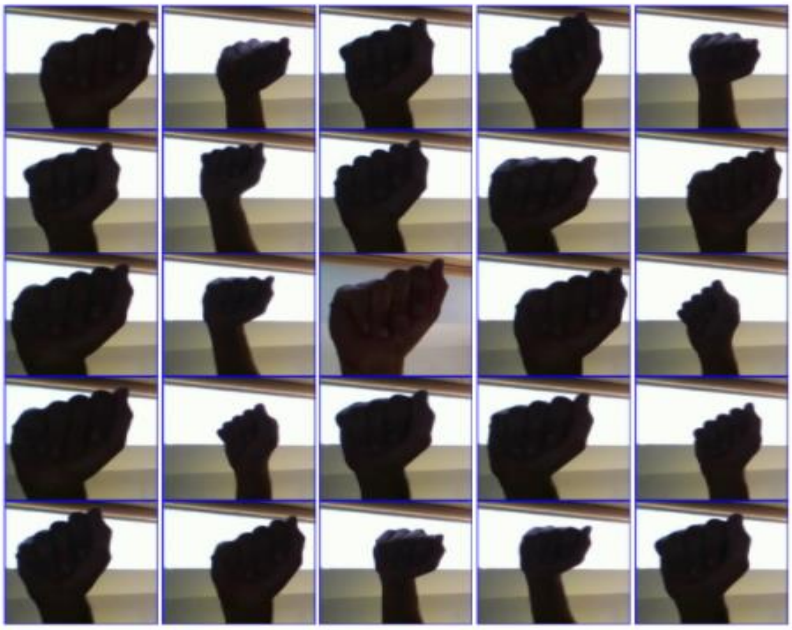


Figure 4.2: Collected sign images.

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

4.1.2. 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. In 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 the main advantage of using this algorithm [18].

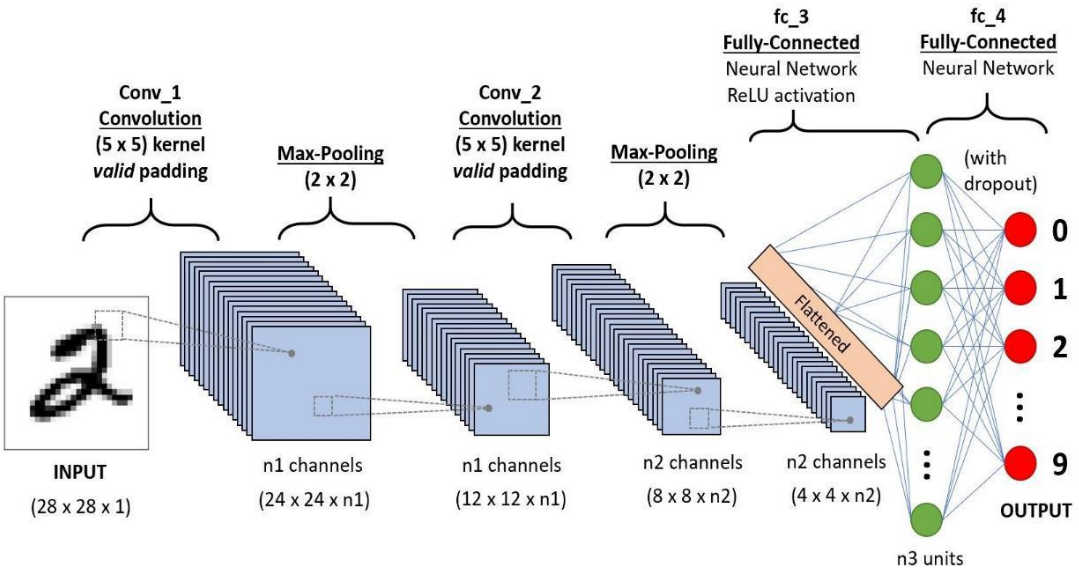


Figure 4.3: Process in the CNN algorithm.

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 4.4 illustrates the code segment which shows the RGB pixel intensities of the image which is captured through a 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])
```

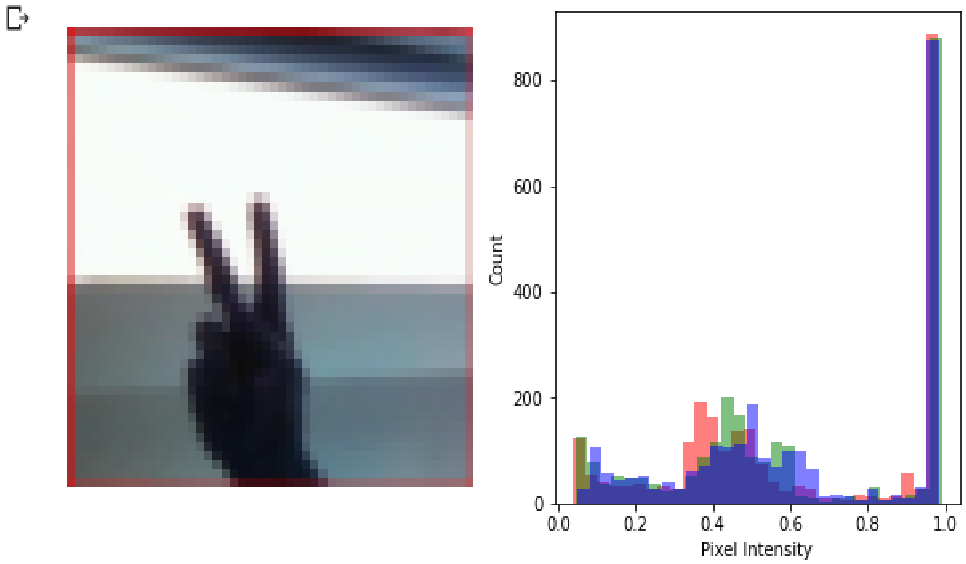


Figure 4.4: Code segment and output for pixel intensity.

4.1.3. 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 is sent to the API, it gets the sign image and does the classification. Identified alphabet will be sent as a response.

4.2. Technologies and Resources

4.2.1. Software Requirements

- Python

The core part of the component is a machine learning model. So that before development we have to choose the best programming language which can support various libraries. We choose Python because it has many support libraries for image understanding. For this project, we have used Python 3.8 version.

- 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 that 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.

4.2.2. Hardware Requirements

As the project has a machine learning model there is a basic need for a good hardware requirement to train, test, and to run the model. They are,

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

4.3. Flow of Project

4.3.1. Requirements Gathering and Analysis

Requirement gathering is a very essential aspect in software development life cycle. I did a review of the literature in my field of operation as a research-based initiative. In this way, research papers are read, and specifications are obtained via these documents. The ideas and requirements gathered from these papers are very much useful, but we must not consider them as our real requirements. We must get into the field and need to gather information from the real stakeholders. Then only we can decide what are our requirements and what is not. So, we met some people from mute and deaf schools who are familiar with local sign language translations. We gathered information through interviews and questioners. It is easy to get an idea of how the application should operate and what could be enforced, by learning what they expect. The main aim of "Gathering Requirements" is to define what the application should do. If the requirement collecting steps are correctly performed, then the other steps can be handled quite smoothly. So, this step is very much important, and it should be conducted in a proper way to succeed in the project.

After gathering the requirements, we did a requirement analysis. This stage is conducted because there are many requirements and we have to select what is possible and what is not possible. There were some requirements with incomplete or inconsistent data, and some were unclear. These found concerns were addressed and explained with our supervisor before further proceeding with the project. So, we exclude some of them and others were altered with additional information. At the same time, we also analyzed other Sri Lankan sign language interpreters to get more accurate requirements.

4.3.2. Feasibility Study

The aim of this process is to consider, calculate and evaluate the shortcomings of the possibilities in the proposed project. Other uses of feasibility studies are to control financial feasibility, predicting operating chance, the possibility of technology, and business valuation. The main restrictions understood are the project's time and potential.

Technical Feasibility

The technological prospect stresses the fact that existing technical capabilities are taken into consideration and that they are valid for the likely needs of the proposed application. It is an approximation that the creation can express the current conceptual criteria which can be calculated as a means of checking that the design is theoretically possible. During the technical analysis, the proposed system centered primarily on interpretation program related APIs, machine learning algorithms and source codes. From this feasibility study, we decided to use Python as a programming language because it provides various libraries which are useful and feasible for our project. The IDE's and other supporting tools were analyzed in the study.

Financial Feasibility

The financial viability evaluation is intended to regulate the positive financial aid that would be offered by the proposed system. Presented on the basis of a cost over benefit study including evidence like needed technology, tools, conveyance and other resources. Upon evaluating all the costs versus the incentives, the application's advantages exceeded the costs. So, the project being planned should be calculated as a project that is commercially feasible.

Operational Feasibility

Operational feasibility is to control how the conceptual plan tackles the challenges and how it satisfies the criteria which were recorded in the SDLC requirement collection and review phase. In order to get a clear understanding of the objectives in the suggested solution, requirements were previously gathered by meeting sign language interpreters. The requirements which are documented were collected via discussions and surveys. So that we could add new functionalities and features to the system and confirm the user needs. Therefore, the proposed system was decided to be an operationally feasible product.

4.3.3. Time Frame

For a good development, project planning is more important. So, we maintained a project backlog. This component's major tasks are broken into user stories and completed in basic sprints. Each month we planned the tasks to be completed and the task which has impediments and issues were moved to the next sprint. Figure 4.5 illustrates the milestone of the component and followed by are user stories and project Gantt chart.

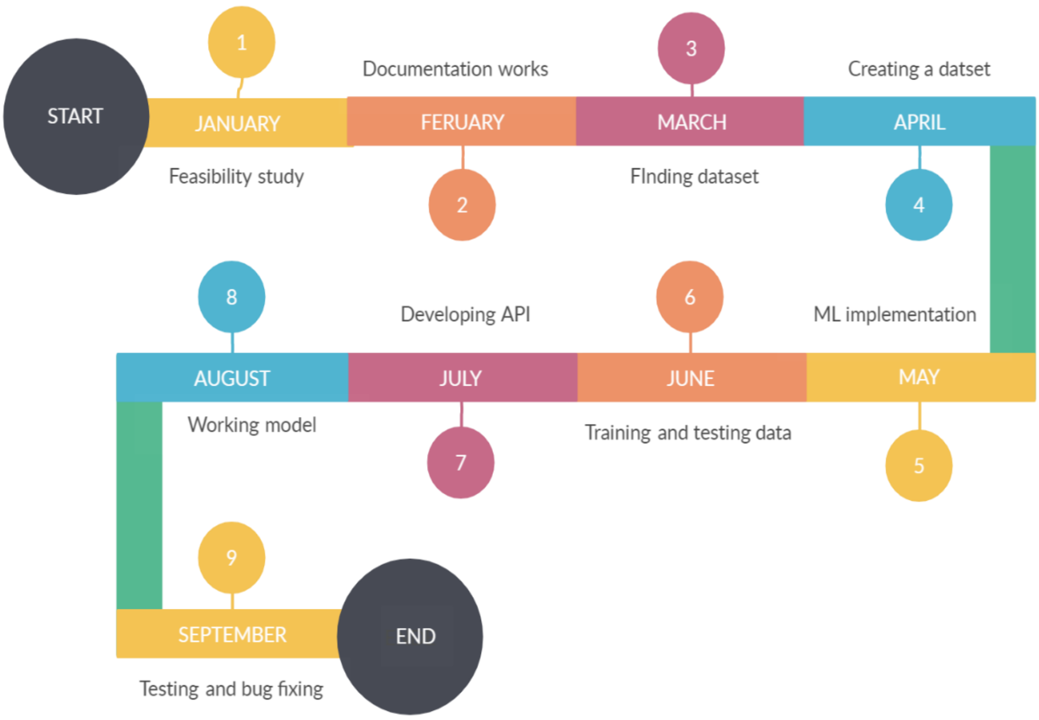


Figure 4.5: Research component milestone.

User stories are divided into four main parts and issues are created accordingly.

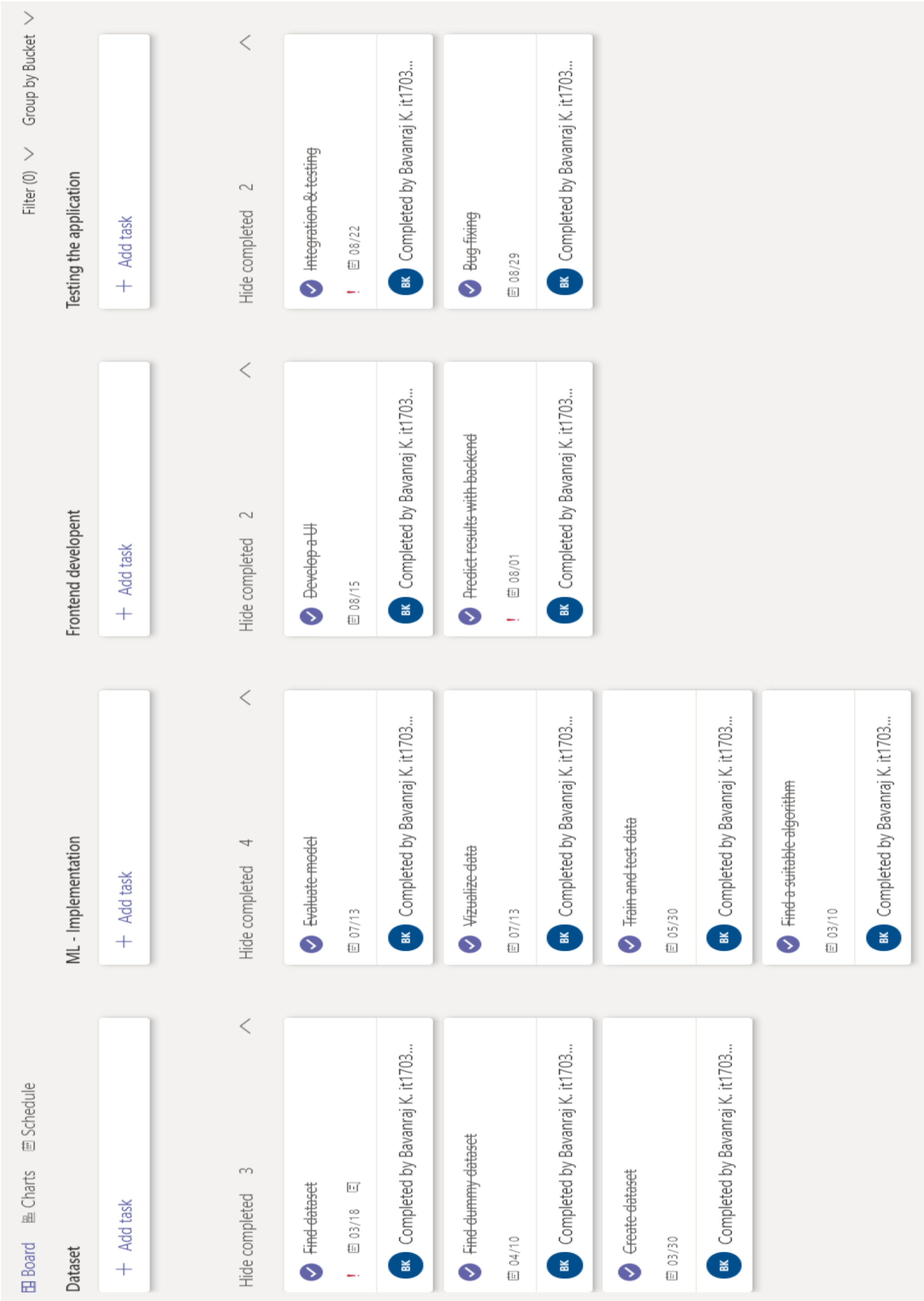


Figure 4.6: Tasks assigned for the component.

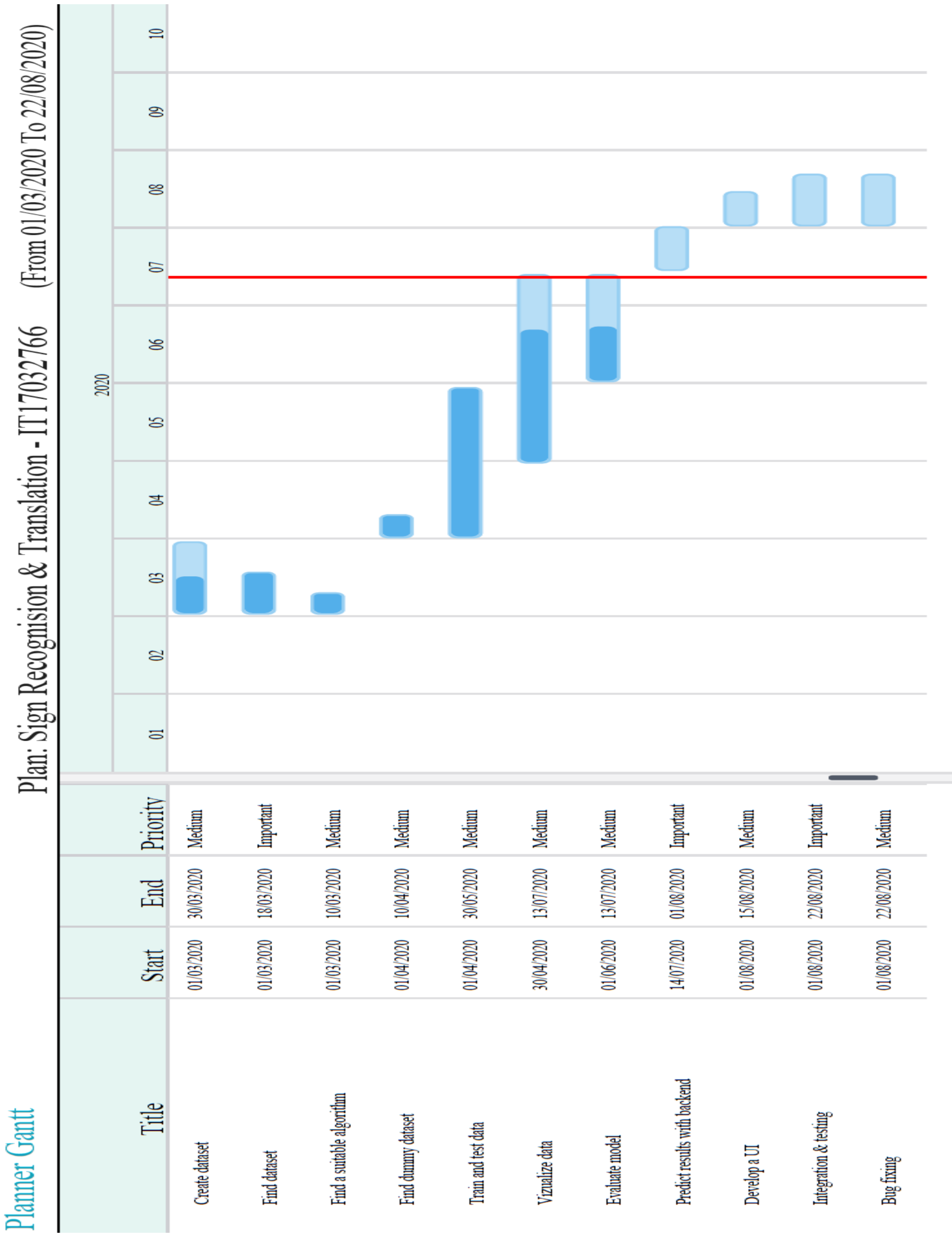


Figure 4.7: Project Gantt chart.

4.4. Commercialization Aspects of the Product

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.

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.

4.4.1. 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.

For example, consider the scenario below:

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

5. TESTING & IMPLEMENTATION

Testing components are more essential in the 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 has several other testing methodologies. To provide the 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 5.1 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)
```

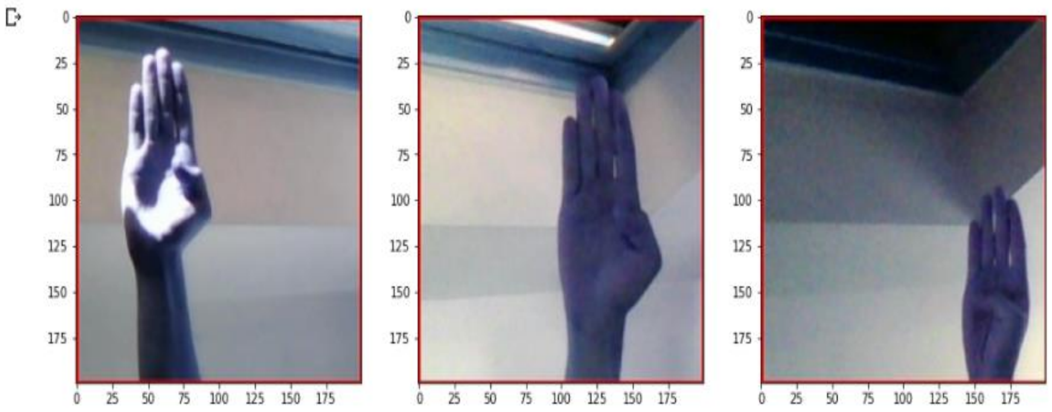


Figure 5.1: Sample output of the dataset.

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

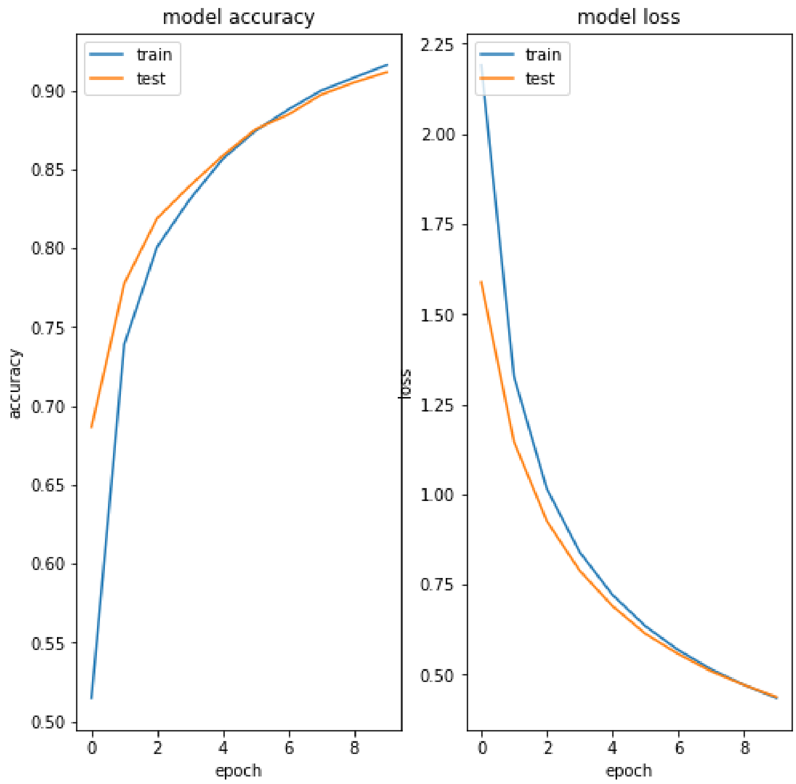


Figure 5.2: Model accuracy and loss in training and testing.

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 unit of the component is tested for the best functionality. After passing unit tests they were integrated into 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 into the main system. In the system, the 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 5.3: Manual test cases.

6. RESULTS & DISCUSSION

6.1. Results

The component I built for the research project is a reusable component. If anyone wants to use it in their system, they can simply call the relevant API method in their code. In the "EasyTalk" system it serves its purpose as the second module, connecting first and third modules. The whole purpose of this component is to provide an image classification tool. The classifications are done on an average accuracy of 91%. The latency for image detection is almost one second. When using this API, HTTP request is sent with JSON objects containing sign image data and dataset. The response also will be in the same format containing the alphabet letter.

6.2.Discussion

The machine learning model gives a decent accuracy but there are still problems in the classification phase. Sign representation for letters "I" and "J" are quite similar in ASL. It is difficult to detect through a low-resolution web camera. Some signs are shown with movements from different angles and it also makes the system to struggle in classification. With more data collection and training, I think we can fix these issues in future works. Table 6.1 shows the satisfaction level of each outcome in my component.

Table 6.1: Satisfaction level of the outcomes.

Outcome	**Review**
Accuracy of the ML model	Good
Quality of the API	Good
Time taken for translation	Average
Reusability of the component	Good

6.3. Research Findings

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 I tried to build an API that accepts different sign languages and it does the work quite normally.

6.4. Summary of Contribution

Table 6.2: Summary of contribution.

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.

7. 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 main research area addressed in this report is sign recognition and translation. It is one of the major components of the system. Basically, this component is built as an API that contains a machine learning model in it. This API fetches a sign image with a dataset and identifies the English alphabet letter through the machine learning model. Then the identified letter will be passed to the next component in “EasyTalk”. The main objective of this component is to develop an API which is useful for sign language related developments. Existing systems are using a single sign language method and if anyone needs to build a similar system for different languages then, they have to develop from the beginning. But this research outcome removes that burden and provides them an interface to classify images. The only thing they have to do is, to send an HTTP request to the component with a valid image and dataset. The other problem addressed by the component is real time translation. The main requirement of an interpreter is its performance. This component can classify images with an average latency of 0.9 seconds which is a decent performance but not the best. In future developments, we can expect that performance and accuracy can be improved with the advancement of technology.

REFERENCES

[1] World Health Organization, "Deafness and hearing loss." https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss (accessed Feb. 22, 2020).

[2] World Health Organization, "World report on disability." https://www.who.int/disabilities/world_report/2011/report/en/ (accessed Feb. 22, 2020).

[3] SaraEgo1, "Communication between deaf and hearing society," 2014. https://storify.com/SaraEgo01/communication-between-deafand-%0Ahearing-society-mor (accessed Feb. 20, 2020).

[4] Department of Census and Statistics, "Persons with disability in hearing/speaking by cause of disability and sex." http://www.statistics.gov.lk/PopHouSat/Des_Chra.asp (accessed Feb. 22, 2020).

[5] Rohana Special School, "An Introduction to Sri Lankan Sign Language," 2007. http://www.rohanaspecialschool.org/wp-content/uploads/2010/08/AnIntroductionToSriLankanSignLanguage.pdf (accessed Jan. 13, 2020).

[6] University of Moratuwa, "KATHANA." http://lms.uom.lk/sf/shantha/Project-web-sites/2009-10/PI-33-%0Akathana/overview.html (accessed Jul. 06, 2020).

[7] Microsoft Azure, "Azure Kinect DK Documentation." https://docs.microsoft.com/en-us/azure/kinect-dk/ (accessed Feb. 22, 2020).

[8] Leap motion developer, "Leap Motion Developer documentation." https://developer.leapmotion.com/documentation/ (accessed Feb. 22, 2020).

[9] G. García-Bautista, F. Trujillo-Romero, and S. O. Caballero-Morales, "Mexican sign language recognition using kinect and data time warping algorithm," *2017 Int. Conf. Electron. Commun. Comput. CONIELECOMP 2017*, 2017, doi: 10.1109/CONIELECOMP.2017.7891832.

[10] D. Naglot and M. Kulkarni, "Real time sign language recognition using the Leap Motion Controller," *Proc. Int. Conf. Inven. Comput. Technol. ICICT 2016*, vol. 2016, pp. 1–5, 2016, doi: 10.1109/INVENTIVE.2016.7830097.

[11] Y. Xue, S. Gao, H. Sun, and W. Qin, "A Chinese sign language recognition system using leap motion," *Proc. - 2017 Int. Conf. Virtual Real. Vis. ICVRV 2017*, pp. 180–185, 2017, doi: 10.1109/ICVRV.2017.00044.

[12] H. V. Verma, E. Aggarwal, and S. Chandra, "Gesture recognition using kinect for sign language translation," *2013 IEEE 2nd Int. Conf. Image Inf. Process. IEEE ICIIP 2013*, pp. 96–100, 2013, doi: 10.1109/ICIIP.2013.6707563.

[13] M. Hassan, K. Assaleh, and T. Shanableh, "User-Dependent Sign Language Recognition Using Motion Detection," *Proc. - 2016 Int. Conf. Comput. Sci. Comput. Intell. CSCI 2016*, pp. 852–856, 2017, doi: 10.1109/CSCI.2016.0165.

[14] Y. Chen and W. Zhang, "Research and implementation of sign language recognition method based on Kinect," *2016 2nd IEEE Int. Conf. Comput. Commun. ICCC 2016 - Proc.*, pp. 1947–1951, 2017, doi: 10.1109/CompComm.2016.7925041.

[15] K. Dabre and S. Dholay, "Machine learning model for sign language interpretation using webcam images," *2014 Int. Conf. Circuits, Syst. Commun. Inf. Technol. Appl. CSCITA 2014*, pp. 317–321, 2014, doi: 10.1109/CSCITA.2014.6839279.

[16] S. Shahriar *et al.*, "Real-Time American Sign Language Recognition Using Skin Segmentation and Image Category Classification with Convolutional Neural Network and Deep Learning," *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, vol. 2018-Octob, no. October, pp. 1168–1171, 2019, doi: 10.1109/TENCON.2018.8650524.

[17] Ahanna.Org, "Spiritually Enlightened Deaf Community through Pure Buddhism," *Ahanna(listen)*, 2016. http://www.ahanna.org/en/about/ (accessed Jul. 06, 2020).

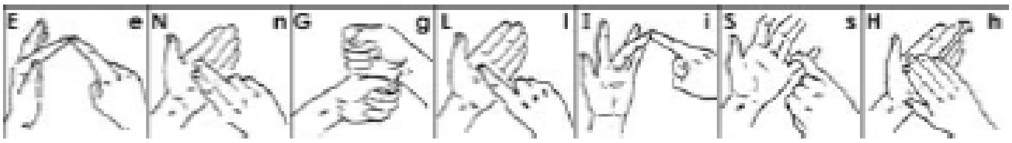
[18] Techopedia, "Convolutional Neural Network (CNN)." https://www.techopedia.com/definition/32731/convolutional-neural-network-cnn (accessed Feb. 22, 2020).

APPENDICES

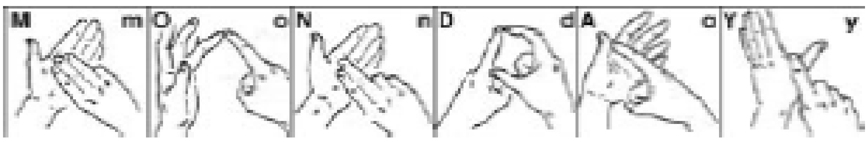
Appendix A

Appendix B

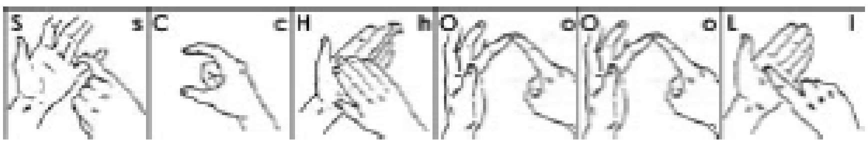
හෝඩිය - Alphabets



ඉංග්‍රීසි
English



සඳුදා
Monday



පාසල
School