SIGN LANGUAGE RECOGNITION APP

Mokshak Ketan Dagli[1], Dr Preeti Savant[2]

[1]Post Graduate Student, [2]Assistant Professor in School of Computer Science and Information Technology,
JAIN (Deemed to be University), Bangalore

***Abstract*: It is very difficult for all the people living in the society to communicate with, the specially able people, mainly deaf and dumb. To remove the hindrance of people not able to understand the Sign Language this a solution that we have come up with. This is a mobile application which developed using the Flutter and Dart framework with the help of different packages like tensor flow lite, teachable machine, camera, TTS etc. The main objective of this idea is to convert the Sign Language into a human-readable and understandable format which is text and speech. This paper is all about the way this application is developed, and the various technologies used in developing the application.**

***Keywords*: flutter, tensor flow lite, teachable machine, TTS, cross-platform application.**

INTRODUCTION

In the current situation around the world, communication is playing a very important role in the lives of people. There are different types of people in society that connect to each other via Facebook, Twitter, Linked-In, neighbors, families etc. But it is very difficult that both parties to know each other's language or a common language that they can communicate easily with.

According to the latest survey done by the WHO (World Health Organization), around 15% people of the world population is specially able and at least 5 to 7% people out of them are deaf and dumb. The main problem faced by these deaf and dump people is that it is very difficult for them to communicate with each other and with the people around them in society.

These especially able people mainly deaf and dumb, use a special language to communicate, which is known as Sign Language. Sign Language is a way of explaining and sharing feelings with the help of signs and, in this growing world, it is very difficult for people to understand and learn this sign language. If a person speaks in Kannada and other person does not know the language, he/she can you a translator ex: Google Translate as a medium to communicate. The objective of Google Translate is to translate from any one language to another. Arguably [8], there has never been an issue for deaf people to access technology because it is a communication disability, not a physical challenge.

Therefore, to solve this problem, a mobile application has been developed which can be used as a translator or as a recognition system which can be deployed on a handy device like a smart phone which people carry with them in their day-to-day life with the ability to identify or recognize the hand signs depicted by a deaf and dumb person. This paper will give a detailed explanation about the flow of a Sign Language Recognition application, the technology used behind it along with the future works and how it can improve soon with the help of this growing technology in the field of mobile applications, recognition systems, etc.

LITERATURE REVIEW

According to what research we have done, we have came across many ways of implementing this idea of Sign Language Recognition. In this section we shall discuss the different ways how people have researched this idea with different methods of developing using various technologies.

While doing the literature review, we found out that there are multiple ways this ideology can be developed with its respective pros and cons, which we shall discuss later in this section. Firstly, all the researchers have stated which kind, or type of Sign Language Recognition system should be taken into consideration for recognition. There are mainly two types of Sign Language which are Static Sign Based and Dynamic Sign Based. The Static Sign base is that type of system where Static images are used for recognition. The dynamic Sign base is that type where live videos can be used as an input to predict the hand sign portrayed in them. Hong Li and et. al [1] used contour-based feature to recognize dynamic single hand sign.

The second / main thing to be analyzed is, after identifying which type of sign based system is to be used, we must identify using which technology we should develop the Sign Language Recognition system. Victoria Adewale [2] used K-Nearest Neighbor algorithm to develop the Sign Language recognition system. This idea had restricted him to deploying the application/program on to a computer/laptop enabled with Python with the necessary hardware requirements like the camera. The most difficult part of this idea was that it was very difficult to perform image segmentation and object recognition and to solve this, they used SURF and FAST algorithms in the field of machine learning.

Mahesh Kumar [3] used a combination of algorithm to develop this system. He used Linear Discriminant Analysis (LDA) as the main algorithm to which was used to predict and identify the hand sign. This idea also had the same problem where this program/application can only be deployed/executed over a computer system/laptop enabled with Python on it. This system used a combination of algorithms in the image cleaning/segmentation section where the input is broken down into frames as it was a dynamic sign based system, then an image is selected out of the frame, an algorithm for cleaning the image is used, another algorithm for identifying the sign in the image is used, etc. which made this a little complicated and difficult to implement.

P Vijayalakshmi [4] used a unique way to identify hand Sign produced by people which was using an IOT device developed by them. In this idea, they developed an IOT device which is enabled with various sensors such as Flex sensors (measure degree to which the finger is bent), Accelerometer sensors (measure the degree to which the finger is tilted), Tactile sensors (measure the physical interaction between fingers) and many more sensors which were all connected to a single system called as the Arduino micro-controller board. This board is used to manage all the devices connected to it, acting as a bridge to pass collected data to the back-end system for further processing and identifying the hand sign depicted by the person. A back-end algorithm was also developed which takes all these inputs passed by the micro-controller into a probable value of hand sign based on the input provided to the system. This idea is a little difficult to implement but has a lot of potential in the coming future where IOT is going to play a major role in the world of technology, health-care, business, etc.

P. S Rajam [7] developed a unique way to predict hand signs. The ideology used here was to use unique combinations of binary values where each binary value states a different hand sign. In this technique, the upwards and the downwards fingers are detected and, based on this, the binary code is generated. Based on the generated binary code, that value is cross-checked with the saved values. If the match is present or if the value exists, the corresponding hand sign stored on this value is represented as the predicted value, else it displays cannot identify the hand sign. The problem is that the dataset used in this model is limited to 32 hand signs.

IMPLEMENTATION

This application has been implemented in three stages. The first stage is the registration and login section, second section being the learning section and developing the model which can detect the hand sign and the third being the recognition system developed using Flutter and Dart along with various other libraries which we shall discuss later in this section.

These types of applications can be developed and have already been developed in different ways, such as python programming applications, etc. The main problem with these types of applications is that they can be used only on a computer system, or a laptop enabled with python and various hardware services such as camera, speaker, etc. But the idea used behind this application is completely different. Here we are going to develop a mobile application which can be installed on any smartphone which is of type Android or IOS as this application is developed using the technology developed by Google known as Flutter and Dart. Flutter and Dart is a framework owned by Google which can used to develop cross- platform applications using a single code base. This means that you write your code using the Dart language, and you can easily deploy it over different platforms such as Android, IOS, Windows, macOS, Linux, and Web Applications along with a design system of Material UI which can also be changed based on user preference. You can also use Fluent UI to design your application, which is a design system used by Windows applications that gives the same user experience as a window's application.

The flow of the application can be seen in figure 1. Here, we first ask the users to create an account on the application via phone number or email address and password which is developed using another technology or service provided by Google known as Firebase. Firebase is a Back-end as a Service system which can be easily integrated into various platforms such as Android, IOS, Web applications, etc. which can be used as an authentication system, storage system used to store data in the form of tables (Firebase Real-time Database) or in the form of NoSQL structure (Firebase Firestore), etc. If the user has already created an account on the application, he/she can directly log in either via the Email Address and Password used while creating the account or via the Phone Number which, when entered and submitted, receives an OTP on the entered mobile number allowing the user to ensure that the security of his/her account has been taken care of completely. After logging into the system or signing in to the system, the user has navigated to User Dashboard Section/Screen. This section/screen provides the user with two options.

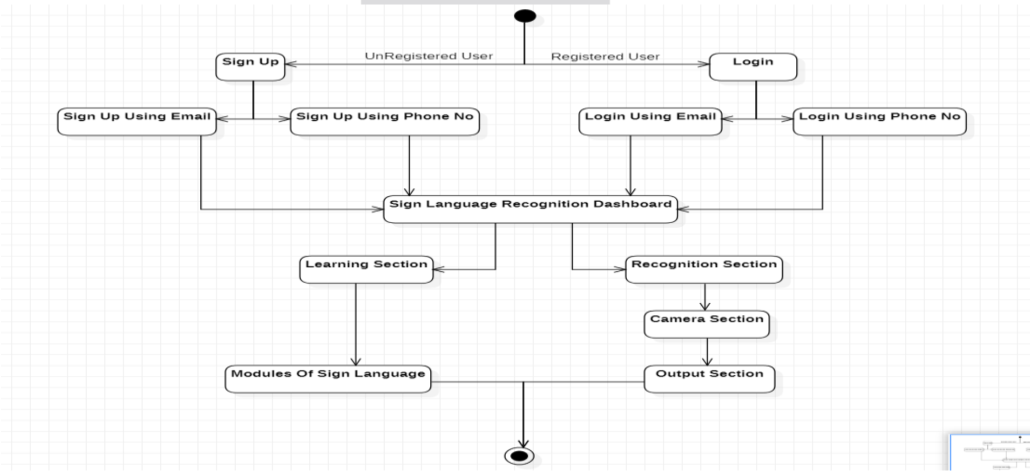


Figure 1: Flow Chart of the Application

The first option is the Learning section or the Learning screen. This screen allows the user to learn various hand signs which are depicted as images with the label below it. This data depicted on this screen has been retrieved from the Firebase Firestore database which is a NoSQL structured database, which allows the user to store data in a non-relational format (Tables). The data stored for each hand sign is an individual document with two properties, one being the path to the image and another being the value of what the sign is. The image path referred to here is the link to the image uploaded on the Firebase cloud storage. The Firebase cloud storage is a cloud database which can be used to store various kinds of data such as images, audio files, videos, etc. over a cloud system.

After the learning section, the second option which is available on the user's dashboard is the recognition system, which is the main functionality of the application. This section/screen will be explained in detail later in this section and finally comes the output/result screen which is used to display the output in text/sound.

The system architecture of our application can be seen in figure 2. The system architecture is a diagram/representational design which is used to depict the whole activity diagram or the flow of the application in a detailed format. Each part / component of the system architecture displays the input parameters it takes and the function it can perform for a better and deeper understanding of the flow of the application. Ex The Phone Number Login Screen has two input parameters where the first input parameter is the phone number of the user once entered and clicked on send OTP. An OTP from the back-end system is generated to the registered/entered phone number which then needs to be entered on the screen, which becomes the second parameter of the screen. This eventually leads to calling a function on the screen being getUserCredentials(). This architecture helps the developers to better understand the hierarchy that is to be followed and the detailed information about each and every component in the application.

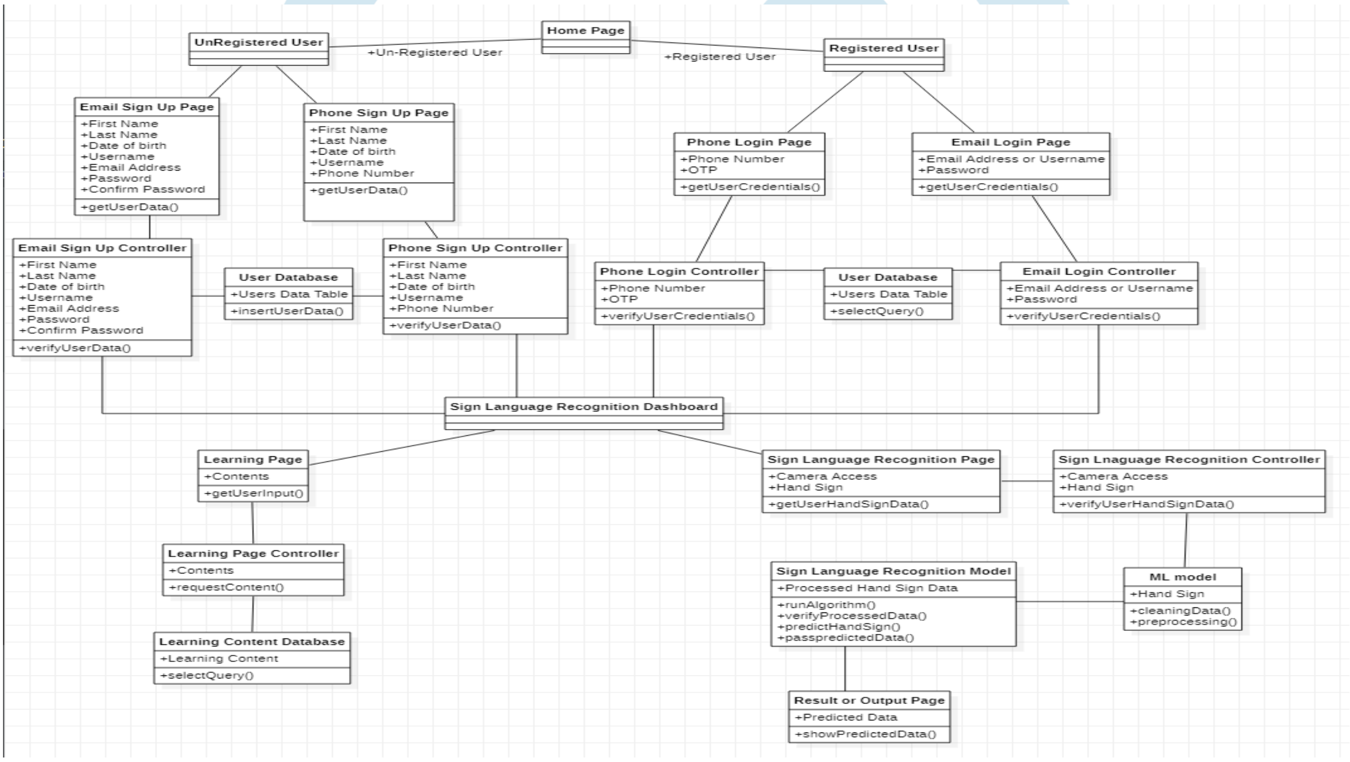


Figure 2: Architecture of the Application

The first stage of the application, as seen here in the system architecture, explains how the user interacts with the application and how he/she makes it available for them to get access to the user dashboard, which is by creating an account or logging into the application if already created an account. This section allows users to create accounts either with a phone number or email address and password and once done, this data is stored and managed at Firebase Authentication. The Firebase Authentication system allows you to create user, manage users and delete users for your application. Once the users creates an account with the application, the details like Name, DOB, etc. are stored at the Firebase Real-time database, which is relational database used to store data in the form of tables with the help of rows and columns. Once the user enters the details on the screen and clicks on create account, the data is collected and passed by the controller to the back-end system to create the account. It is the role of the controller to manage and maintain the validation of the data collected on the screen.

Once the user logs in and gets access to the user dashboard, the two available options are the Learning section and the Recognition section. The Learning section allows all the users to learn and understand how each sign is based on the label and image depicted on this screen.

The second option available on the user's dashboard screen is the recognition section, which is an important part of the application. This section has various steps which are followed one after the other to complete the cycle of the recognition process. So, once the user clicks on the recognition system, the application checks if the user has already provided permission to access the camera and if not, it requests the user to provide access for it as to recognize the hand sign that the camera is the source of input hardware device for the system.

The recognition system for this application was developed using a web-based tool which is known as Teachable Machine. Various other tools, libraries, technologies can be used as a recognition system, such as using Open CV, machine learning algorithm, existing machine learning models like classification, regressions etc, which can help you predict the probability of the image passed as an input. The whole recognition application can be broken down into 2 stages, which are explained in detail below.

1) Dataset

For our application, we have developed a custom dataset which is a group of images from various sources and from various angles. The images of the hand sign from different angles are taken into consideration to make the dataset more and more accurate. If we provide a single image in a huge number, then it can give us results, but it might not guarantee us accuracy. So, to make the application more accurate, different images of a single hand sign have been used and images of a hand sign from different angles have also been grouped into it. As seen in the Fig 3 after generating the dataset, we finally add the dataset to the Teachable machine to generate the model for the given dataset.

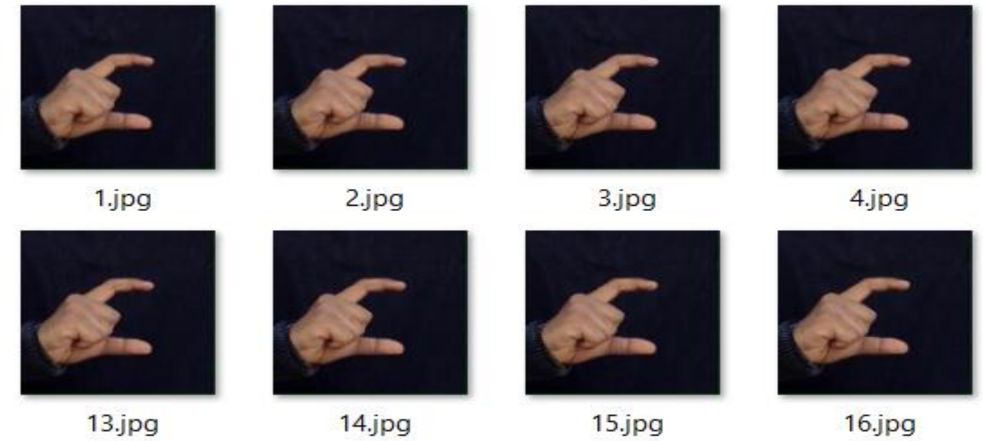


Figure 3: Dataset Sample

2) Model Generation

After the dataset has been generated and all the images required to develop the model have been collected, we now must a develop the machine learning model which can help us predict the hand sign, and we also have to convert it into a file format which can be easily integrated into a mobile application understandable format. There are different ways the model can be generated. You can either write your own python scripts with the help of different machine learning models, or you can use the Open CV to generate the model, etc.

We are going to use a Teachable machine to generate a machine learning model. Teachable machine is a web-based tool which can help you a generate machine learning model in an effective and efficient manner. On the [6] Google's Teachable Machine website there are 3 options that can choose used to create machine learning, namely image classification mode, voice classification and pose classification. The method used by teachable machines for model generation is the classification method of machine learning. In our case, we provided different classes with images relevant to their sign language and passed them to the Teachable machine to generate the machine learning model as shown in the Fig 4. You can easily customize your classes and add more precision to the developing machine learning model before the model generation starts.

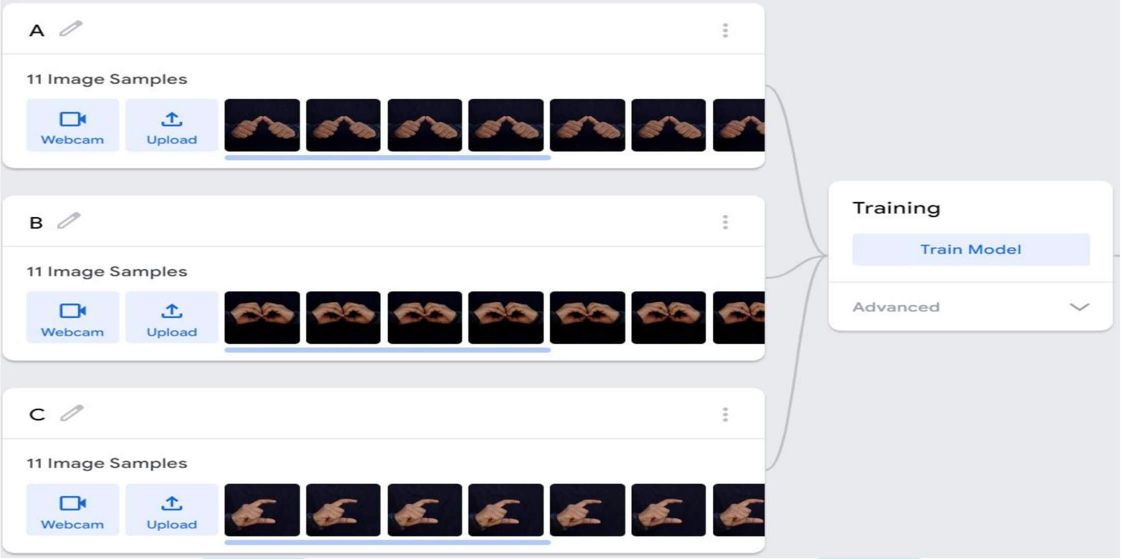


Figure 4: Model Generation

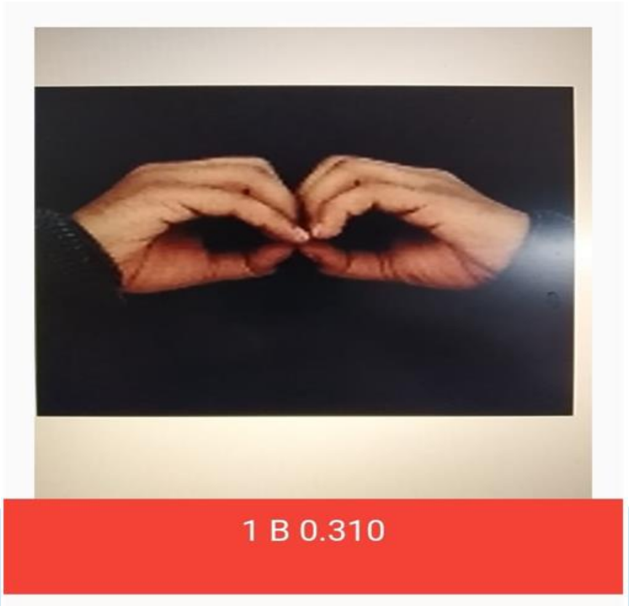
After the model of the Teachable machine with the machine learning technique of classification has been developed, we now need to convert the machine learning model into a mobile usable format which is the TensorFlow lite format. TensorFlow Lite (TFlite) [5] is a lightweight, fast and cross-platform open-source machine learning framework specifically designed for mobile and IOT. We converted the machine learning model into a TFlite file format as there is no ways to integrate the complete machine learning model into a Flutter mobile application. So, we use the TFlite package of dart to access the TFlite formatted files to perform recognition into the system.

After the machine learning model has been integrated into the system, we now have to develop the sign language recognition section on the mobile application where it uses the TFlite file, which is a machine learning model to detect the hand sign and populate the necessary result based on the input. So, we do that with the help of different dart packages of which TFlite is one of them. We also use the TFlite helper package to manage the interpreter of the recognition system and the instance of the machine learning model in the background. We also use a camera package using which we can manage the camera of the device and use it based on our needs. The Flutter TTS package is also used to generate text to speech.

So, as this recognition system is a dynamic sign-based one and as the sign changes, the value also changes as the machine learning model recognizes the sign. So, to add more enhancements and under friend ability to the application, we represent the result in two formats. The first format is a simple human understandable format which is text displayed on the screen parallelly at the time of recognition. The second format is voice/speech which is achieved using the TTS package of the dart. So, the predicted value will be passed to a box with a button which will generate the text into speech format and make it easier for the user to understand the predicted hand sign.

RESULTS

The model currently used in the system has the capability to detect only 16 hand signs as only a number of classes were provided to develop the model. To enhance the model and to make it detect more hand signs, a number of classes can be created along with its relevant images and, after training, it can easily predict those hand signs as well.



1 B 0.310

Figure 5: Detection System

To make the model improve its accuracy to predicting the hand sign right, make sure to provide different hand signs with different types of images where these hand signs are represented or displayed at different angles, etc. Fig 5 gives a small glance of the prediction and the system developed to predict hand signs.

CONCLUSION

To help the people of society, an ideology has been developed which can be used to detect or predict hand signs. This paper gives detailed information on how this idea has been developed, the technologies used to develop the system, and how this can be improved along with a few results. This paper also gives a small summary of different research and the techniques used to solve this problem. This is a simple mobile application which can detect signs in human-readable format which is text and speech.

REFERENCES

[1] H . Li and M. Greenspan, “Model-based segmentation and recognition of dynamic gestures in continuous video streams,” Pattern Recognition., vol. 44, no. 8, pp. 1614–1628, 2011.
[2] V. Adewale and A. Olamiti, “Conversion of Sign Language to text and speech using machine learning techniques,” JOURNAL OF RESEARCH AND REVIEW IN SCIENCE, vol. 5, no. 1, 2018.
[3] M. Kumar, “Conversion of Sign Language into text,” Ripublication.com. [Online]. Available: https://www.ripublication.com/ijaer18/ijaerv13n9_90.pdf. [Accessed: 19-Jan-2022].
[4] P. Vijayalakshmi and M. Aarthi, “Sign language to speech conversion,” in 2016 International Conference on Recent Trends in Information Technology (ICRTIT), 2016, pp. 1–6.
[5] Li Shuangfeng. TensorFlow Lite: On-Device Machine Learning Framework[J]. Journal of Computer Research and Development, 2020, 57(9): 1839-1853.
[6] D. Agustian, P. P. G. P. Pertama, P. N. Crisnapati and P. D. Novayanti, "Implementation of Machine Learning Using Google's Teachable Machine Based on Android," 2021 3rd International Conference on Cybernetics and Intelligent System (ICORIS), 2021, pp. 1-7, doi: 10.1109/ICORIS52787.2021.9649528.
[7] P. S. Rajam and G. Balakrishnan, “Real time Indian Sign Language Recognition System to aid deaf-dumb people,” in 2011 IEEE 13th International Conference on Communication Technology, 2011, pp. 737–742.
[8] Harditya, Arya. (2020). Indonesian Sign Language (BISINDO) As Means to Visualize Basic Graphic Shapes Using Teachable Machine. 10.2991/assehr.k.201202.045.