SIGN LANGUAGE TRANSLATOR

A Project work - I Report

Submitted in partial fulfillment of requirement of the

Degree of

BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING

BY

EN19CS303041 Ritik Ratnawat
EN19CS303055 Utkarsh Yeole
EN19CS303056 Vedant Deshmukh

Under the Guidance of

Dr. Harsh Pratap Singh

Department of Computer Science & Engineering
Faculty of Engineering
MEDI-CAPS UNIVERSITY, INDORE- 453331

Aug - Dec 2022

Report Approval

The project work **“Sign Language Translator”** is hereby approved as a creditable study of an engineering/computer application subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approved any statement made, opinion expressed, or conclusion drawn there in; but approve the “Project Report” only for the purpose for which it has been submitted.

Internal Examiner

Name : **Dr. Harsh Pratap Singh**

Designation : **Assistant Professor**

Affiliation :

External Examiner

Name :

Designation :

Affiliation :

Declaration

We hereby declare that the project entitled **“Sign Language Translator”** submitted in partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science Engineering (Sp. Data Science) completed under the supervision of **Dr. Harsh Pratap Singh, Assistant Professor, Department of Computer Science Engineering,** Faculty of Engineering, Medi-Caps University, Indore is an authentic work.

Further, I/we declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

Signature and name of the student(s) with date

Certificate

I/We, **Dr. Harsh Pratap Singh** certify that the project entitled **“Sign Language Translator”** submitted in partial fulfillment for the award of the degree of Bachelor of Technology/Master of Computer Applications by **Ritik Ratnawat, Utkarsh Yeole and Vedant Deshmukh** is the record carried out by him/them under my/our guidance and that the work has not formed the basis of award of any other degree elsewhere.

____

Dr. Harsh Pratap Singh
Department of Computer Science Engineering
Medi-Caps University, Indore

____

External Guide
Department of Computer Science Engineering

____

Dr. Pramod S. Nair

Head of the Department

Computer Science & Engineering

Medi-Caps University, Indore

Acknowledgements

I would like to express my deepest gratitude to Honorable Chancellor, **Shri R C Mittal,** who has provided me with every facility to successfully carry out this project, and my profound indebtedness to **Prof. (Dr.) Dileep K. Patnaik,** Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm has always boosted up my morale. I also thank **Prof. (Dr.) D K Panda,** Pro Vice Chancellor, **Dr. Suresh Jain,** Dean Faculty of Engineering, Medi-Caps University, for giving me a chance to work on this project. I would also like to thank my Head of the Department **Dr. Pramod S. Nair** for his continuous encouragement for betterment of the project.

I express my heartfelt gratitude to my Internal Guide/Project coordinator **Dr. Harsh Pratap Singh** without whose continuous help and support, this project would ever have reached to the completion.

Ritik Ratnawat
EN19CS303041
B.Tech. IV Year
Department of Computer Science & Engineering
Medi-Caps University, Indore

Utkarsh Yeole
EN19CS303055
B.Tech. IV Year
Department of Computer Science & Engineering
Medi-Caps University, Indore

Vedant Deshmukh
EN19CS303056
B.Tech. IV Year
Department of Computer Science & Engineering
Medi-Caps University, Indore

Executive Summary

As we all know that there is always a communication barrier between specially challenge people and Normal people. Although there are special languages that help us to communicate with them but it was not known by everyone or no one should learn it until and unless they doesn't need it. So, the problem is to how to remove these communication barrier with less efforts and faster results.

Proposed Solution

For solving these problem and removing these communication barrier with less efforts and faster results, We proposed a solution named as " Sign Language Translator " which acts as an interface between Normal people and specially challenged such deef or mute. For the solution of the problem, we use Machine learning and Deep learning concepts and build a model that act as mediator from translating the Sign language into English Language and vice verse in real time. So that one can understand the languages of each other with lesser efforts.

Value

This project provides portion of most effective and a reliable solution as per current technology advancements by providing a mobile application where user can capture the hand signs and that is translated to Text and Audio with the help of Machine Learning and Image Processing.

Final value and Thoughts

Considering today’s technology and needs this project may overcome the problem but an effective solution must be required by improving this project which is helpful to this needy people. Because every person has their opinion about the real-world problem and it might be possible that this people can give an idea to solve this real-world problems. So, it is good for us to make a communication between these people’s and normal peoples.

List of Figures

List of Tables

Table of Contents

Chapter - 1

INTRODUCTION

1.1 Introduction

Sign Language is the visual manner to convey the message for Deaf and Dumb Peoples. It is a combination of gestures, orientations, movements of hands, arms or body and facial expression. Like a normal language, Sign Language is also varying considering different factors. Various standard sign languages available in the real world are Indian Sign Language (ISL), American Sign Language (ASL) etc.

According to a great Author Paul J. Meyer, Communication – the human connection is the key to personal and career success [8]. Communication is the important term in real world to allow others and ourself to understand information more accurately and quickly. But it is not easy for the Deaf and Dumb peoples to communicate with the normal peoples as they all can't understand this Sign Language.

This project provides one of the solutions to increase the communication of Deaf and Dumb peoples with the normal peoples. In this digital era, Mobile application is the best solution for everyone to use, so by using the capabilities of Machine Learning and Image processing algorithms available in TensorFlow Library we make the working mobile application. In which, user has to capture image as input and got the output in terms of text and audio. This will ease the medium for those special peoples and in some cases using the Frequent Phrases feature of our application, there is no need of capturing the photos just they need to press single button.

1.2 Literature Review

Table 1.2.1 : Literature Review		
Sr. No.	**Title**	**Methodology**
1.	**A real-time portable sign language translation system[1]**	It uses the wireless system to process the data. To differentiate hand motion, they have inner sensors put into gloves to show the parameters as given by, posture, orientation, motion, defined of the hand in Taiwanese Sign Language could be recognize in no error. The hand gesture is considered by flex inner sensor and the palm size considered using the g sensor and the movement is considered using the gyroscope. Input signals would have to be consider for testing for the sign to be legal or not periodically. As the signal which was sampled can stay longer than the pre-set time, the legal gesture sent using phone via connectivity like Bluetooth for differentiating gestures and translates it. With the proposed architecture and algorithm, the accuracy OF 94% for gesture recognition which is quite satisfactory.
2.	**Automated Sign Language Interpreter[2]**	It demonstrates Instrumented gloves with audio out are the solution here. The gloves attached with various sensors are worn for sign interpretation. Hence, the proposed system solves the problem and helps the dumb people in communication with the rest of the world at low cost.

3.	**Vision-based sign language translation device [3]**	The proposed system which is an interactive application program developed using LABVIEW software and incorporated into a mobile phone. The sign language gesture images are acquired using the inbuilt camera of the mobile phone; vision analysis functions are performed in the operating system and provide speech output through the inbuilt audio device thereby minimizing hardware requirements and expense. The experienced lag time between the sign language and the translation is little because of parallel processing. This allows for almost instantaneous recognition from finger and hand movements to translation. This is able to recognize one handed sign representations of alphabets (A-Z) and numbers (0-9). The results are found to be highly consistent, reproducible, with fairly high precision and accuracy
4.	**Sign language interpreter using a smart glove [4]**	It demonstrates a novel approach of interpreting the sign language using the portable smart glove. LED-LDR pair on each finger senses the signing gesture and couples the analog voltage to the microcontroller.
5.	**A machine learning based approach for the detection and recognition of Bangla sign language [5]**	It demonstrates Hand Gesture recognition which is performed using HOG (Histogram of Oriented Gradients) for extraction of features from the gesture image and SVM (Support Vector Machine) as classifier. Finally, predict the gesture image with output text. This output text is converted into audible sound using TTS (Text to Speech) converter.

6.	**Speech Recognition Automation by ASR[9], [10]**	Author have presented multiple experiments to design a statistical model for deaf people for the conversion to sign language from the speech set. They have further made the system that automates the speech recognition by ASR by the help of animated demonstration and translation statistical module for multiple sets of signs. As they went ahead, they used the following approaches for the translation process, i.e., state transducer and phrase defined system. As of evaluation certain figures type have been followed: WER, BLEU after that comes the NIST. This paper demonstrates the process that translates the speech by automation recognizer having all three mentioned configurations. The paper came up with the result with finite type state transducer having the word error rate among the range of 28.21% and 29.27% for the output of ASR.
7.	**Indian Sign Language (ISL) Translation System For Sign Language Learning[6]**	It works in a continuous manner in which the sign language gesture series is provided to make a automate training set and providing the spots sign from the set from training. They have proposed a system with instance learning as density matrix algorithm that supervises the sentence and figures out the compound sign gesture related to it with a supervision of noisy texts. The set at first that they had used to show the continuous data stream of words is further taken as a training set for recognizing the gesture posture. They have experimented this set on a confined set of automated data that is used for training of them, identification for them and detection stored a subtle sign data to them. It has been stored around thirty sign language data that was extracted from the designed proposal.

1.3 Problem Statement

As we all know that there is always a communication barrier between specially challenge people and Normal people. Although there are special languages that help us to communicate with them but it was not known by everyone or no one should learn it until and unless they doesn't need it. So, the problem is to how to remove these communication barrier with less efforts and faster results.

1.4 Objectives

1. This project provides an interface to increase the communication of Deaf and Dumb peoples with the normal peoples.
2. Eliminating the need of any specialized external device to interpret the Sign languages.
3. Establishing a communication channel for deaf and muted people to make conversation with Normal people.
4. Reducing the efforts of understanding the different types of Sign languages.
5. Also release Audio outputs for faster understanding and conversation.

1.5 Significance

According to 2011 census of India, there are 63 million people which sums up to 6.3% of the total population, who are suffering from hearing problems. Out of these people, 76-89% of the Indian hearing challenged people have no knowledge of language either signed, spoken or written. The reason behind this low literacy rate is either the lack of sign language interpreters, unavailability of Indian Sign Language tool or lack of researches on Indian sign language.

The module for translation will help hearing disabled people to understand in an efficient and easy way by providing them with a video to convey them the message of text.

Our solution is different as we are using the technique of videos while other techniques are using the images for message conveyance. Also, the stop words have been removed so text processing efforts are also reduced.

1.6 Research Design

This research seeks to focus on related articles and studies that tackled deep learning in sign language translation into text, which fluctuated through the years 2017 to 2021. There were articles that did not make use of deep learning or NLP, which will not be included in the scope of this paper. Specifically, important details such as algorithms used in the study will be discussed. Alongside this, the different sources of input in the study and the type of content that the research papers have used to recognize sign language will be discussed. With the information gathered, a proposed system will be discussed.

There are a total of forty (40) papers obtained through the selection process; however, after filtering, only twenty (20) articles remain which fulfill the requirements based on their abstract and title. With a total of twenty (20) studies, IEEE Xplorer = 6, ACM = 6, Springer = 4, Elsevier = 2, Taylor and Francis Online = 1, and World Scientific = 1 were chosen and screened. The studies were chosen for their importance to deep learning, sign language recognition, and translation. The chosen studies were determined after screening the articles. The studies that satisfy the criteria for searching based on their relevance to the issue account for the small number of research available. The summary of the details of each study in sign language translation is shown in Table 1. Specifically, the table enumerates the different publishers of the paper, which are reputable sources such as ACM, Springer, Elsevier, and more. The different deep learning techniques are also listed, some of which uses a variation of neural networks, LTSM, and other classifier techniques.

1.7 Source of Data

Sign Language is a naturally evolved language like other oral languages. It is used by persons with deafness for day-to-day life communication. It is considered as a mother tongue of persons with deafness. Like other languages it is also an independent language with complex grammar. It involves naturally evolved visual-manual signs.

1. Indian Sign Language is the predominant sign language in the subcontinent of South Asia, used by at least 15 million deaf signers. We have taken our dataset from Kaggle. Our dataset contains images of hand gestures used in ISL. We have images for digits 1-9 and alphabets A-Z.

Link: https://www.kaggle.com/datasets/prathumarikeri/indian-sign-language-isl

2. The sentence-level completely labelled Indian Sign Language dataset for Sign Language Translation and Recognition (SLTR) research is developed. The ISL-CSLTR dataset assists the research community to explore intuitive insights and to build the SLTR framework for establishing communication with the deaf and dumb community using advanced deep learning and computer vision methods for SLTR purposes.
The Corpus Videos are provided in ISL-CSLTR: Indian Sign Language Dataset for Continuous Sign Language Translation and Recognition

Link : https://data.mendeley.com/datasets/kcmpdxky7p/1

1.8 Chapter Scheme

Chapter 1

These Chapter includes the complete introduction about the project. It also mentions the research done on the prior work done on the project. These chapter also tells about the objectives of the project and its significance in real-world. It also explains the process of research which tells about the how the research begins, number of research papers reviewed etc. It also mentions about the different sources of the data.

Chapter 2

These chapter includes about the different types of requirements needed to run and development of the project. It explains the experimental setup required for the development, debugging and testing of the project. These chapter also tells about the activities included in the development process. It also specifies all the functional, non-functional and technical requirements for the project. It describes all the UML diagrams to define the system design and processes of SDLC.

Chapter 3

These chapter focuses on the implementation and results of the project. It includes the code snippets, results, model evaluation and testing of the model. It also explains about the data generation process which includes Background Subtraction, Image Thresholding and Storage of Images. It also discuss about the frontend development and integration of ML model with the frontend to make it user - friendly.

Chapter 4

These chapter summarizes the whole process and results obtained after development and testing of the project. It also tells about the future scope of the project and scope of improvement in the project. It also lists all the references taken during the project development.

Chapter - 2

SYSTEM REQUIREMENTS AND SPECIFICATION

2.1 Experimental Setup

- **Computer Vision :** Computer vision works much the same as human vision, except humans have a head start. Human sight has the advantage of lifetimes of context to train how to tell objects apart, how far away they are, whether they are moving and whether there is something wrong in an image. Computer vision trains machines to perform these functions, but it has to do it in much less time with cameras, data and algorithms rather than retinas, optic nerves and a visual cortex. Because a system trained to inspect products or watch a production asset can analyse thousands of products or processes a minute, noticing imperceptible defects or issues, it can quickly surpass human capabilities.

- **Tkinter :** Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

- **Tensor Flow :** TensorFlow is an end-to-end open-source platform for machine learning. TensorFlow is a rich system for managing all aspects of a machine learning system; however, this class focuses on using a particular TensorFlow API to develop and train machine learning models.

- **Keras :** Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides.

- **Convolutional Neural Network (CNN) :** is a Deep Learning algorithm that can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

- **Background Subtraction:** The background subtraction method (BSM) is one of the most popular approaches to detecting objects. This algorithm works by comparing moving parts of a video to a background image and foreground image. This method is used to find foreground objects by isolating them while comparing them to the frame where no objects are present; it will find the differences between them and create a distance matrix. Basically, what it does is compare the difference in the value of two frames, one frame without an object and the other with objects to count, with the threshold value. The threshold value is predefined by using the first few frames of the video. Hence if the difference in the value of two frames is greater than the pre-set threshold value, the result is marked as a moving object detected.

- **Thresholding:** Image thresholding is a simple form of image segmentation. It is a way to create a binary image from a grayscale or full-colour image. This is typically done in order to separate "object" or foreground pixels from background pixels to aid in image processing.

2.2 Proposed Method

For the project implementation first of all backend is implemented by going through the following functions explained in a brief manner :

- **Data Collection / Generation :** For this project, we use real images taken from webcam and then it can be categorized into different classes according to their signs.

Table 2.2.1 : Image Statistics	
Sign	**Number of Images**
1	2400
2	2400
3	2400
4	2400
5	2400
6	2400
7	2400
8	2400
9	2400

- **Model Building :** For model building, we use python and its frameworks in which we add the images according to the classes and then by changing some of the attributes like epochs, batch size etc. And the last stage is of exporting model with given types i) Tensorflow ii) Tensorflow.js iii) Tensorflow Lite.

  → **Convolutional Neural Network (CNN) :** A convolutional neural network (CNN or ConvNet) is a network architecture for deep learning that learns directly from data. CNNs are particularly useful for finding patterns in images to recognize objects, classes, and categories. They can also be quite effective for classifying audio, time-series, and signal data.

  → **ADAM Optimizer :** This algorithm is used to accelerate the gradient descent algorithm by taking into consideration the 'exponentially weighted average' of the gradients. Using averages makes the algorithm converge towards the minima in a faster pace.

- **Integrating model with Desktop App :** The TensorFlow Lite Android Support Library makes it easier to integrate models into application. It provides high level APIs that help transform raw input data into the form required by the model, and interpret the model's output, reducing the amount of boilerplate code required. It supports common data formats for inputs and outputs, including images and arrays. It also provides pre and post processing units that perform tasks such as image resizing and cropping.

For front end implementation of the project is as follows :

- **Developing Desktop Application :** An interactive and operable desktop application is developed with minimalistic UI design so that all the operations are in approach to the user easily. User just need to capture images per words or live stream and then click single button to see the recognized text and single click to hear the audio of corresponding text.

2.3 Data Flow Diagram

2.3.1 Data Flow Diagram (Level 0)

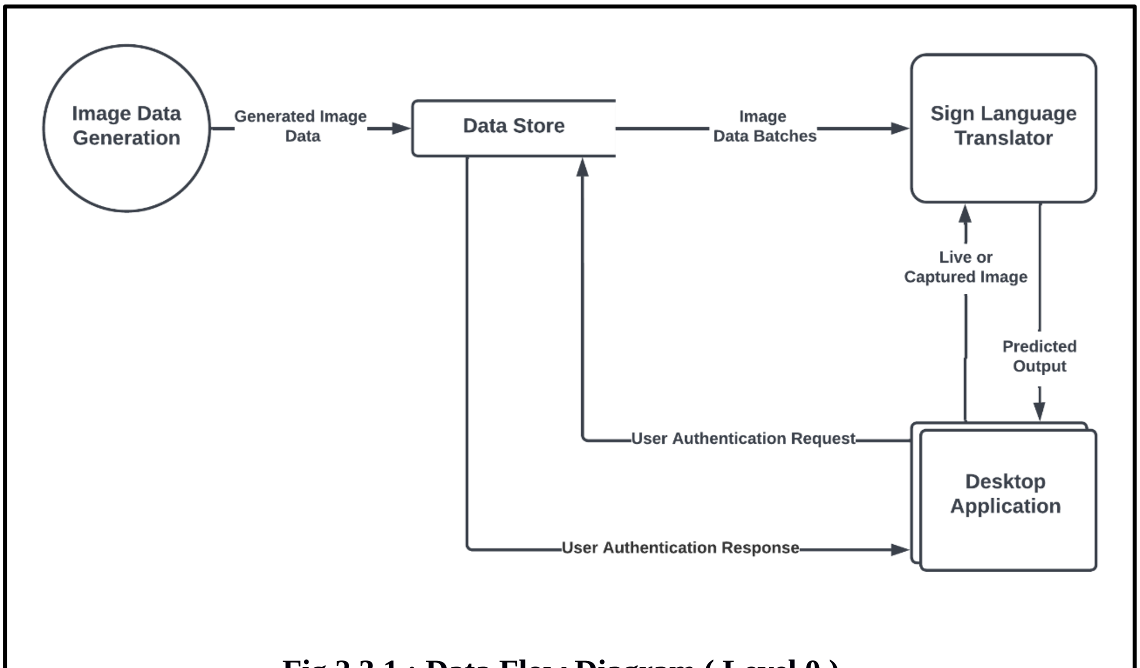

Fig 2.3.1 : Data Flow Diagram (Level 0)

2.3.2 Data Flow Diagram (Level 1)

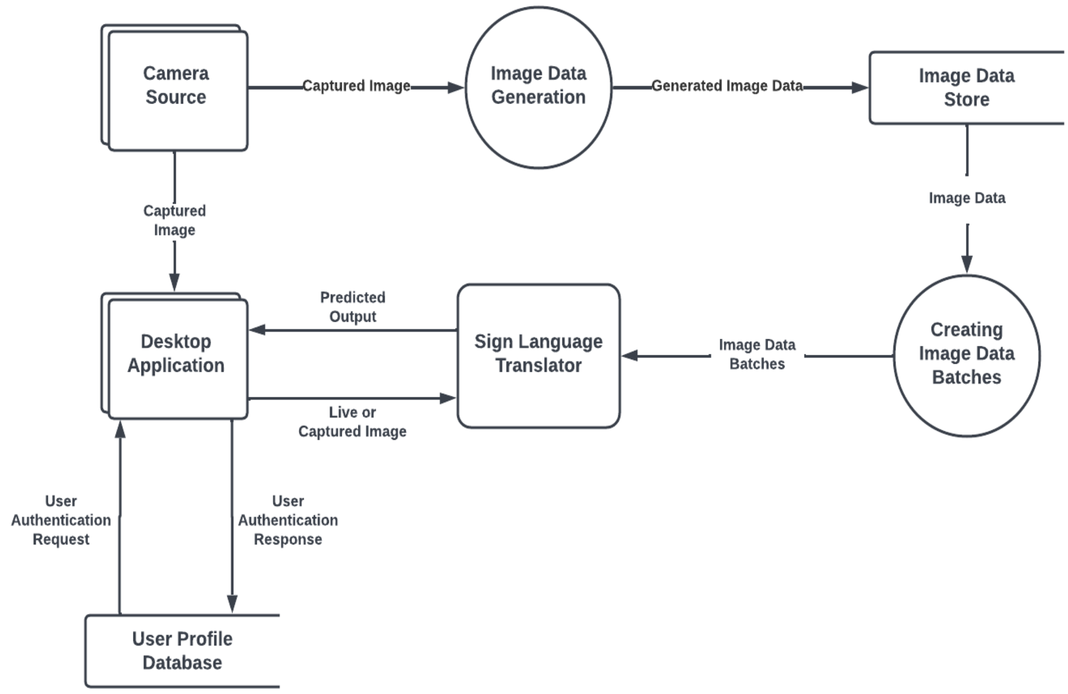

Fig 2.3.2 : Data Flow Diagram (Level 1)

In 1-level DFD, a context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main objectives of the system and breakdown the high-level process of 0-level DFD into subprocesses. Level 1 DFD breaks down the main process into subprocesses that can then be seen on a more deep level. Also, level 1 DFD contains data stores that are used by the main process.

2.3.3 Data Flow Diagram (Level 2)

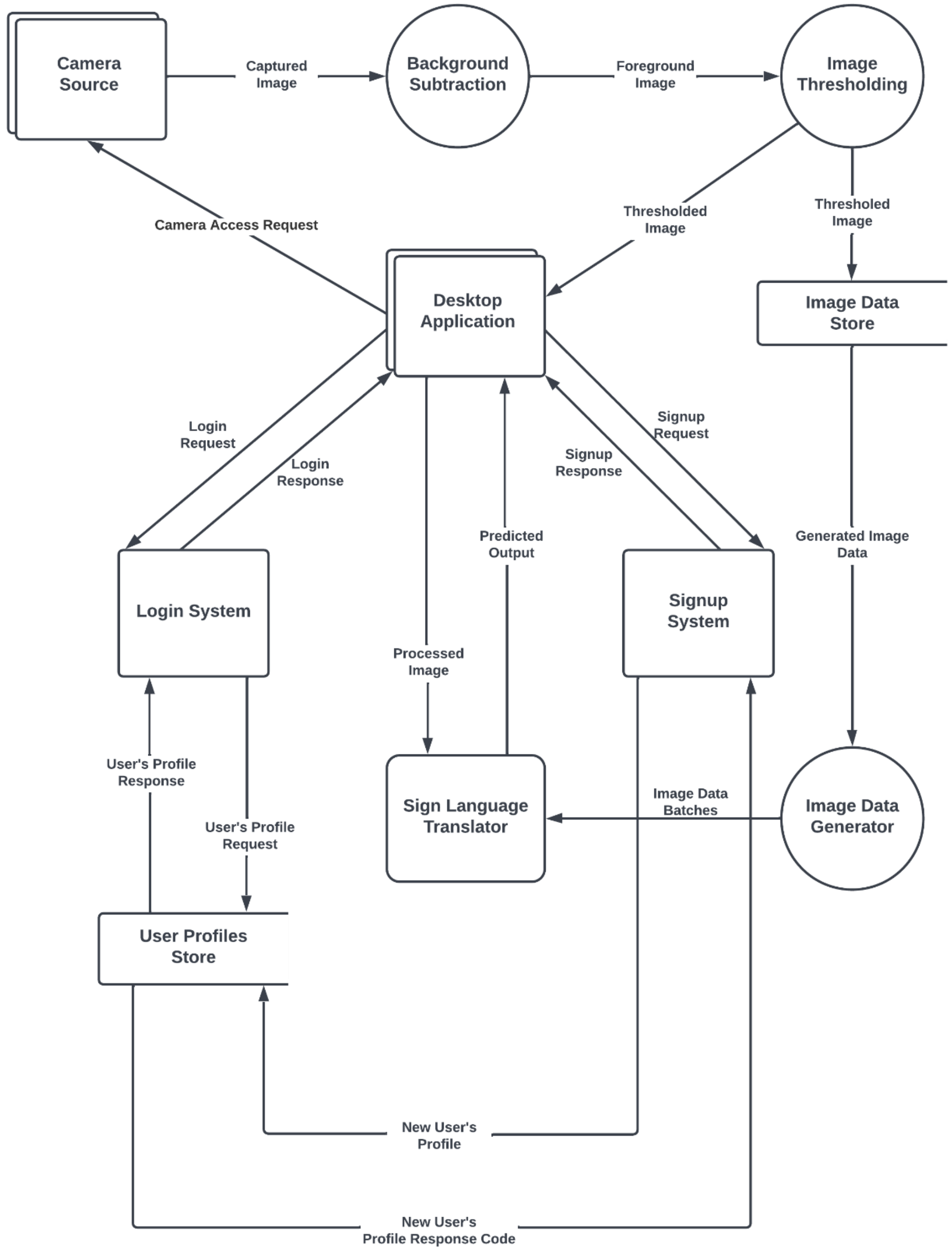

Fig 2.3.3 : Data Flow Diagram (Level 2)

- **Camera Source :** It is the external device used for both the process of Data Generation as well as taking input for translation of Sign Language into English Language.

- **Background Subtraction :** The background subtraction method (BSM) is one of the most popular approaches to detecting objects. This algorithm works by comparing moving parts of a video to a background image and foreground image. This method is used to find foreground objects by isolating them while comparing them to the frame where no objects are present.

- **Image Thresholding :** Image thresholding is a simple form of image segmentation. It is a way to create a binary image from a grayscale or full-colour image. This is typically done in order to separate "object" or foreground pixels from background pixels to aid in image processing.

- **Image Data Store :** It is the database used for storing the images captured during the process of Data Generation as well as in the process of Translation to improve accuracy.

- **Image Data Generator :** Keras Image Data Generator is used for getting the input of the original data and further, it makes the transformation of this data on a random basis and gives the output resultant containing only the data that is newly transformed. It does not add the data. Keras image data generator class is also used to carry out data augmentation where we aim to gain the overall increment in the generalization of the model.

- **Sign Language Translator :** It is ML model which is used to translate the Sign Language into English Language. It uses the Convolutional Neural Network (CNN) to recognize the patterns and objects and ADAM optimizer which helps to improve accuracy and efficiency of the Model.

- **Desktop Application :** These is a Frontend for Sign Language Translator with which user interact to use the features of the translator with some additional features.

- **Login System :** It is a system which is integrated with the Desktop Application for the purpose of User Authentication.

- **Signup System :** It is a system which is integrated with the Desktop Application for the purpose of Data collection of the New User and assigning the credentials for using the Sign Language Translator.

- **User Profile Store :** It is database which is used by Login and SignUp system to store and retrieve the user's data for User Authentication and Data Collection.

2.4 Functional Requirements

- **Live Capturing Video :** The system should take the live feed from the User for further analysis.

- **Background Subtraction :** The system should be able to clip foreground object from background.

- **Thresholding :** The system should separate "object" or foreground pixels from background pixels to aid in image processing.

- **Sign Translation :** The system be to able to translate a Sign into its meaning in other language.

- **User Authentication :** The system should have the functionalities of Login and Sign up to authenticate the User and build individual sessions.

2.4 Non – Functional Requirements

- **Accuracy :** Since we will give the priority to the accuracy of the software, the performance of the Music Recommender will be based on its accuracy on recommendations.

- **Failure handling System :** Components may fail independently of others. Therefore, system components must be built so they can handle failure of other components they depend on.

- **Openness :** The system should be extensible to guarantee that it is useful for a reasonable period of time.

- **Security :** User profile information will be used, so data security is one of the most important concern of the system.

- **Usability :** The software will be embedded in a website. It should be scalable designed to be easily adopted by a system.

- **Reliability :** The system should have accurate results and fast responses to user's changing habits.

- **Hardware Constraints :** To use Sign Language Translation system, user should enter from a personal computer, mobile device, tablet etc.

2.5 Technical Specifications

- 4 GB RAM
- 5 GB free disk space
- Intel Core i3 processor or above
- 40 GB Cache memory
- Web Camera (Integrated / External)
- Python 3.9 or above

2.6 UML Diagrams

1. Use Case Diagram

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.

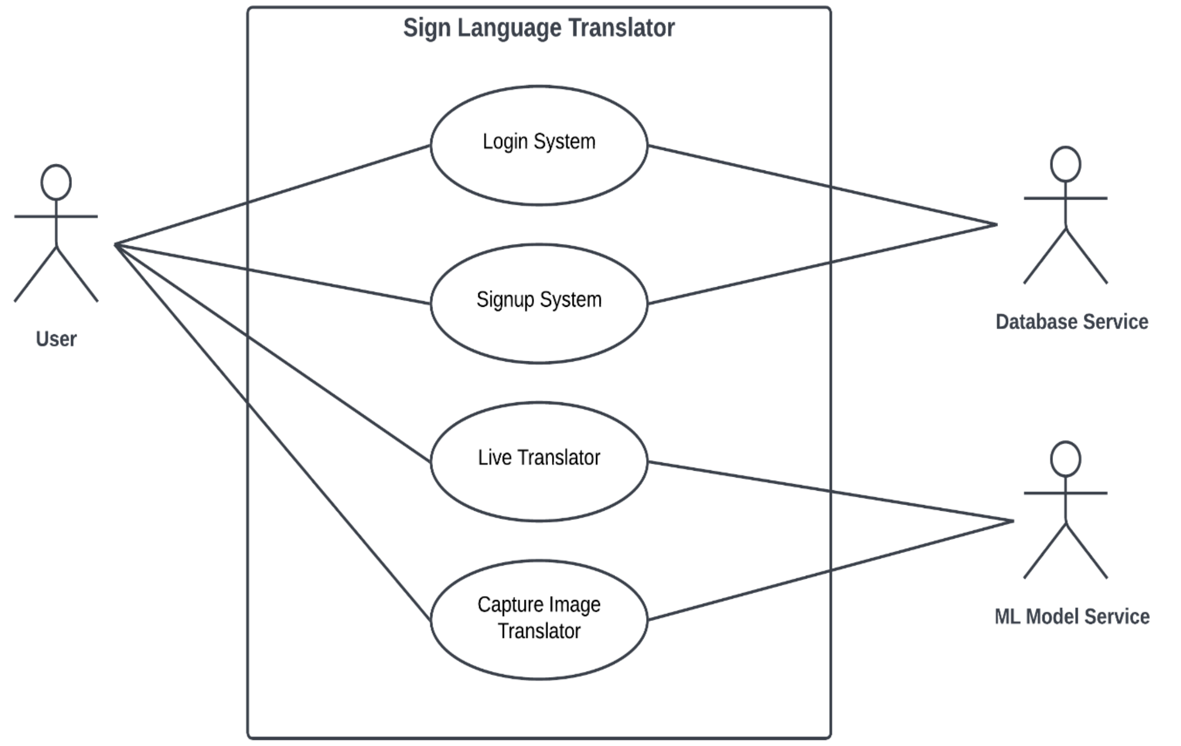

Fig 2.7.1 : Use Case Diagram

2. Activity Diagram

The activity diagram is used to demonstrate the flow of control within the system rather than the implementation. It models the concurrent and sequential activities.The activity diagram helps in envisioning the workflow from one activity to another. It put emphasis on the condition of flow and the order in which it occurs. The flow can be sequential, branched, or concurrent, and to deal with such kinds of flows, the activity diagram has come up with a fork, join, etc. It is also termed as an object-oriented flowchart. It encompasses activities composed of a set of actions or operations that are applied to model the behavioral diagram.

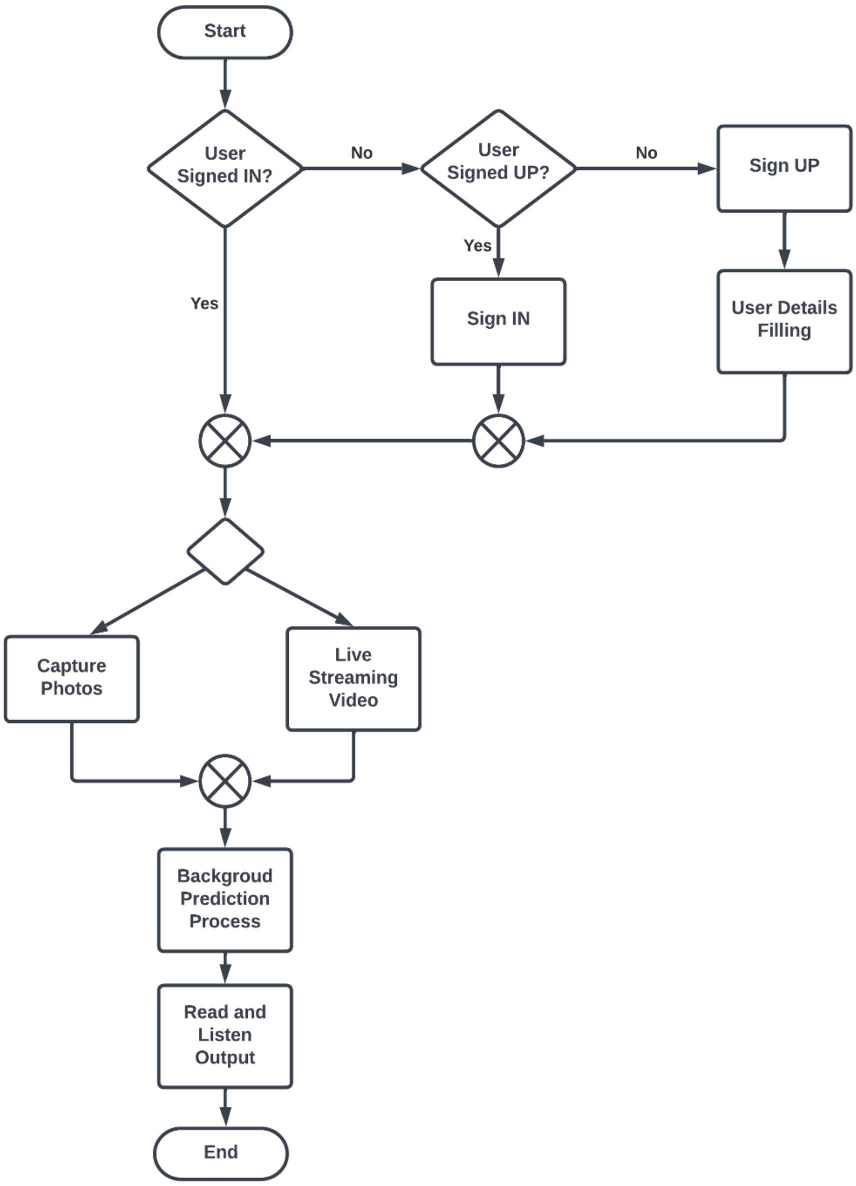

Fig 2.7.2 : Activity Diagram

3. Deployment Diagram

The deployment diagram visualizes the physical hardware on which the software will be deployed. It portrays the static deployment view of a system. It involves the nodes and their relationships. It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths.

The main purpose of the deployment diagram is to represent how software is installed on the hardware component. It depicts in what manner a software interacts with hardware to perform its execution.

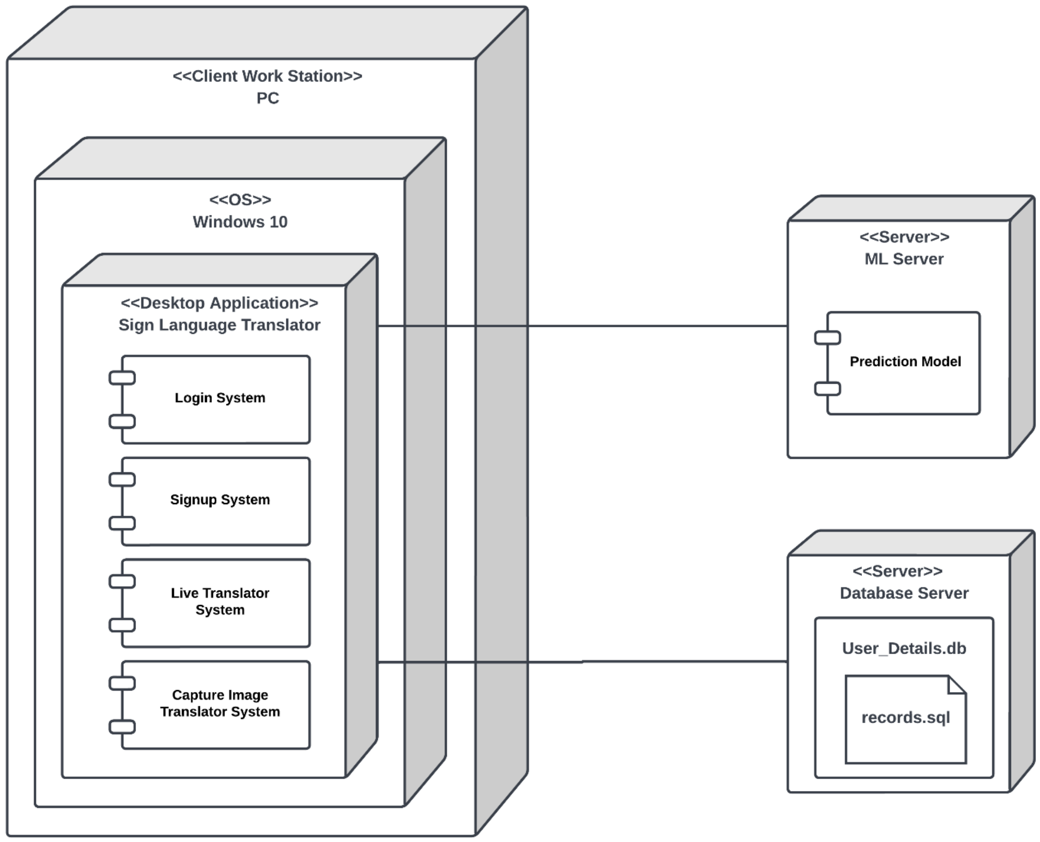

Fig 2.7.3 : Deployment Diagram

Chapter - 3

IMPLEMENTATION & RESULTS

3.1 Data Collection

For this project, we use real images taken from webcam and then it can be categorized into different classes according to their signs.

Function for generating the Data for Signs

In [8]:
```python
def getDataforElement(element):
    cam = cv2.VideoCapture(0)

    num_frames = 0
    nums_imgs_taken = 0

    while True:
        ret, frame = cam.read()

        frame = cv2.flip(frame, 1)
        frame_copy = frame.copy()

        roi = frame[ROI_top:ROI_bottom, ROI_left:ROI_right]

        gray_frame = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
        gray_frame = cv2.GaussianBlur(gray_frame, (9, 9), 0)

        if num_frames < 60:
            calculate_accumulated_avg(gray_frame, accumulated_weight)

            if num_frames <= 59:
                cv2.putText(frame_copy, "FETCHING BACKGROUND...PLEASE WAIT",
                            (80, 400), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0,0,255), 2)

        elif num_frames <= 300:
            hand = segment_hand(gray_frame)
            cv2.putText(frame_copy, "Adjust hand...Gesture for " + str(element),
                        (200, 400), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,255),2)


            if hand is not None:
                thresholded, hand_segment = hand

                # Draw contours around hand segment
                cv2.drawContours(frame_copy, [hand_segment + (60, 60)], -1, (255, 0, 0),1)

                cv2.putText(frame_copy, str(num_frames)+" For " + str(element),(70, 45), cv2.FONT_HERSHEY_SIMPLEX,
                            1, (0,0,255), 2)
                cv2.imshow("Thresholded Hand Image", thresholded)

        else:
            hand = segment_hand(gray_frame)

            if hand is not None:
                thresholded, hand_segment = hand

                cv2.drawContours(frame_copy, [hand_segment + (60, 60)],
                                 -1, (255, 0, 0),1)
                cv2.putText(frame_copy, str(num_frames), (70, 45), cv2.FONT_HERSHEY_SIMPLEX,
                            1, (0,0,255), 2)
                cv2.putText(frame_copy, str(nums_imgs_taken) + " images For " + str(element),
                            (200, 400), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,255), 2)
                cv2.imshow("Thresholded Hand Image", thresholded)
```

```python

                if nums_imgs_taken <= 300:
                    cv2.imwrite(r"D:\\My Learning\\7th Semester\\Project Work - I\\Translator\\gesture\\train\\"
                                +str(element)+"\\" + str(nums_imgs_taken+300) + '.jpg', thresholded)
                    print(nums_imgs_taken, end=" ")
                else:
                    break

                nums_imgs_taken += 1

            else:
                cv2.putText(frame_copy, 'No hand detected...', (200, 400), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,255), 2)

        # Drawing ROI on frame copy
        cv2.rectangle(frame_copy, (ROI_left, ROI_top), (ROI_right,ROI_bottom), (255,128,0), 3)

        # increment the number of frames for tracking
        num_frames += 1

        # Display the frame with segmented hand
        cv2.imshow("Sign Detection", frame_copy)

        # Closing windows with Esc key...(any other key with ord can be used too.)
        k = cv2.waitKey(1) & 0xFF
        if k == 27:
            break

    # Releasing the camera & destroying all the windows...
    cv2.destroyAllWindows()
    cam.release()
```

Fig 3.1.1 : Data Collection

3.2 Model Building :

For model building, we use python and its frameworks in which we add the images according to the classes and then by changing some of the attributes like epochs, batch size etc. And the last stage is of exporting model with given types i) Tensorflow ii) Tensorflow.js iii) Tensorflow Lite.

3.2.1 Creating CNN Layers

In [14]:

```
model = Sequential()

# 1st Convolution Layer
model.add(Conv2D(filters=32, kernel_size=(3,3), padding="same", input_shape=(64,64,3)))
model.add(BatchNormalization())
model.add(Activation("relu"))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.25))

# 2nd Convolution Layer
model.add(Conv2D(filters=64, kernel_size=(3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation("relu"))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.25))

# 3rd Convolution Layer
model.add(Conv2D(filters=128, kernel_size=(3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation("relu"))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.25))

# 4th Convolution Layer
model.add(Conv2D(filters=128, kernel_size=(3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation("relu"))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.25))


# Flattening
model.add(Flatten())

# 1st Fully Connected Layer
model.add(Dense(64))
model.add(BatchNormalization())
model.add(Activation("relu"))
model.add(Dropout(0.25))


# 2nd Fully Connected Layer
model.add(Dense(128))
model.add(BatchNormalization())
model.add(Activation("relu"))
model.add(Dropout(0.25))


# 3rd Fully Connected Layer
model.add(Dense(128))
model.add(BatchNormalization())
model.add(Activation("relu"))
model.add(Dropout(0.25))


model.add(Dense(9, activation='softmax'))
```

Fig 3.2.1.1 : Creating CNN Layers

```
In [15]:  1  model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 64, 64, 32)        896

 batch_normalization (BatchN  (None, 64, 64, 32)       128
 ormalization)

 activation (Activation)     (None, 64, 64, 32)        0

 max_pooling2d (MaxPooling2D  (None, 32, 32, 32)       0
 )

 dropout (Dropout)           (None, 32, 32, 32)        0

 conv2d_1 (Conv2D)           (None, 32, 32, 64)        18496

 batch_normalization_1 (Batc  (None, 32, 32, 64)       256
 hNormalization)

 activation_1 (Activation)   (None, 32, 32, 64)        0

 max_pooling2d_1 (MaxPooling  (None, 16, 16, 64)       0
 2D)

 dropout_1 (Dropout)         (None, 16, 16, 64)        0

 conv2d_2 (Conv2D)           (None, 16, 16, 128)       73856

 batch_normalization_2 (Batc  (None, 16, 16, 128)      512
 hNormalization)

 activation_2 (Activation)   (None, 16, 16, 128)       0

 max_pooling2d_2 (MaxPooling  (None, 8, 8, 128)        0
 2D)

 dropout_2 (Dropout)         (None, 8, 8, 128)         0

 conv2d_3 (Conv2D)           (None, 8, 8, 128)         147584

 batch_normalization_3 (Batc  (None, 8, 8, 128)        512
 hNormalization)

 activation_3 (Activation)   (None, 8, 8, 128)         0

 max_pooling2d_3 (MaxPooling  (None, 4, 4, 128)        0
 2D)

 dropout_3 (Dropout)         (None, 4, 4, 128)         0

 flatten (Flatten)           (None, 2048)              0

 dense (Dense)               (None, 64)                131136

 batch_normalization_4 (Batc  (None, 64)               256
 hNormalization)

 activation_4 (Activation)   (None, 64)                0

 dropout_4 (Dropout)         (None, 64)                0

 dense_1 (Dense)             (None, 128)               8320

 batch_normalization_5 (Batc  (None, 128)              512
 hNormalization)

 activation_5 (Activation)   (None, 128)               0

 dropout_5 (Dropout)         (None, 128)               0

 dense_2 (Dense)             (None, 128)               16512

 batch_normalization_6 (Batc  (None, 128)              512
 hNormalization)

 activation_6 (Activation)   (None, 128)               0

 dropout_6 (Dropout)         (None, 128)               0

 dense_3 (Dense)             (None, 9)                 1161

=================================================================
Total params: 400,649
Trainable params: 399,305
Non-trainable params: 1,344
_________________________________________________________________
```

Fig 3.2.1.2 : Model Summary

3.2.2 Training CNN

The process of adjusting the value of the weights is defined as the “training” of the neural network. Firstly, the CNN initiates with the random weights. During the training of CNN, the neural network is being fed with a large dataset of images being labelled with their corresponding class labels

```
In [18]:  1 history = model.fit(train_batches, epochs=10, callbacks=[reduce_lr, early_stop], validation_data=test_batches)

Epoch 1/10
1752/1752 [==============================] - 338s 192ms/step - loss: 2.5223 - accuracy: 0.1392 - val_loss: 2.3924 - val_accur
acy: 0.1155 - lr: 1.0000e-04
Epoch 2/10
1752/1752 [==============================] - 347s 198ms/step - loss: 2.3226 - accuracy: 0.1766 - val_loss: 2.2617 - val_accur
acy: 0.1558 - lr: 1.0000e-04
Epoch 3/10
1752/1752 [==============================] - 378s 216ms/step - loss: 2.1238 - accuracy: 0.2371 - val_loss: 2.1064 - val_accur
acy: 0.2155 - lr: 1.0000e-04
Epoch 4/10
1752/1752 [==============================] - 352s 201ms/step - loss: 1.9411 - accuracy: 0.2960 - val_loss: 1.9590 - val_accur
acy: 0.3581 - lr: 1.0000e-04
Epoch 5/10
1752/1752 [==============================] - 323s 184ms/step - loss: 1.7702 - accuracy: 0.3632 - val_loss: 1.8894 - val_accur
acy: 0.3505 - lr: 1.0000e-04
Epoch 6/10
1752/1752 [==============================] - 326s 186ms/step - loss: 1.6280 - accuracy: 0.4223 - val_loss: 1.8189 - val_accur
acy: 0.3498 - lr: 1.0000e-04
Epoch 7/10
```

Fig 3.2.2.1 : Training CNN

3.2.3 Model Analysis

Post Training quantization is a conversion technique that can reduce model size and inference latency, while also improving CPU and hardware accelerator inference speed, with a little degradation in model accuracy. Thus, it's widely used to optimize the model. We can achieve the better accuracy of model by tweaking some the training hyperparameters.

→ **Epochs :** more epochs could achieve better accuracy until it converges but training for too many epochs may lead to overfitting.

→ **Dropout - rate :** The rate for dropout, avoid overfitting. None by default.

→ **Learning - rate :** The rate to train the model.

→ **Batch - size :** A batch is a set of samples used in one iteration of training.

→ **Samples – Amount :** Amount of sample input data for training.

Here, we are using 80% of input data for training purpose and 20% for the test purpose. The loss of model decreases by increasing the samples per classes as shown in below given graphs.

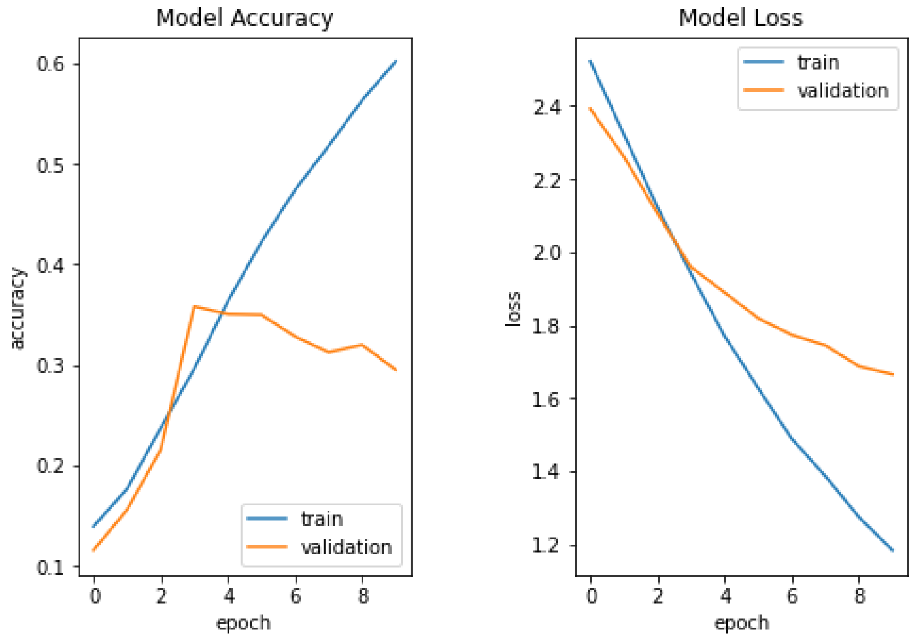

Fig 3.2.3.1 : Model Analysis

3.3 Developing Desktop Application

An interactive and operable desktop application is developed with minimalistic UI design so that all the operations are in approach to the user easily. User just need to capture images per words or live stream and then click single button to see the recognized text and single click to hear the audio of corresponding text.

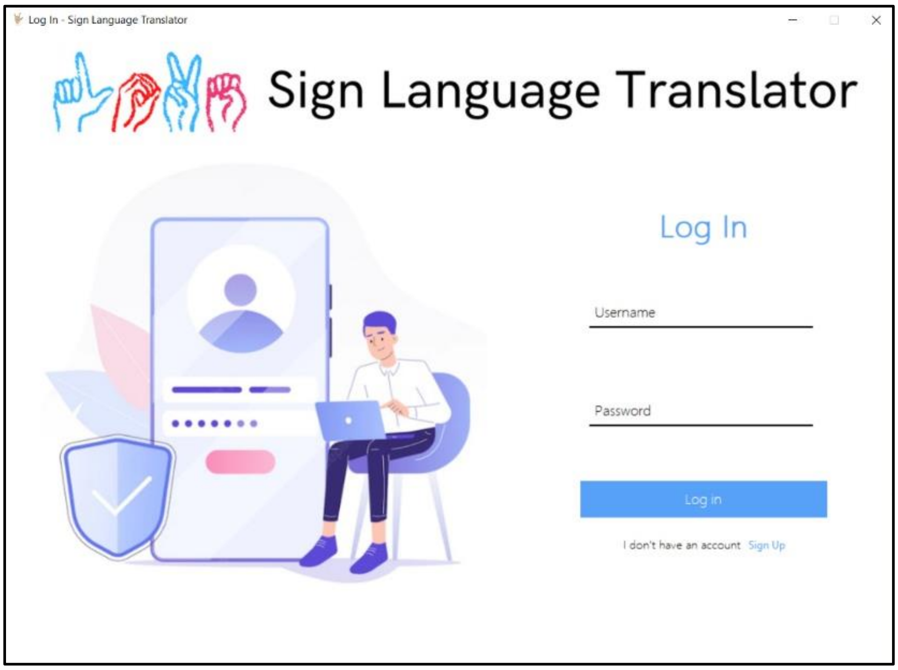

Fig 3.3.1 : Login UI

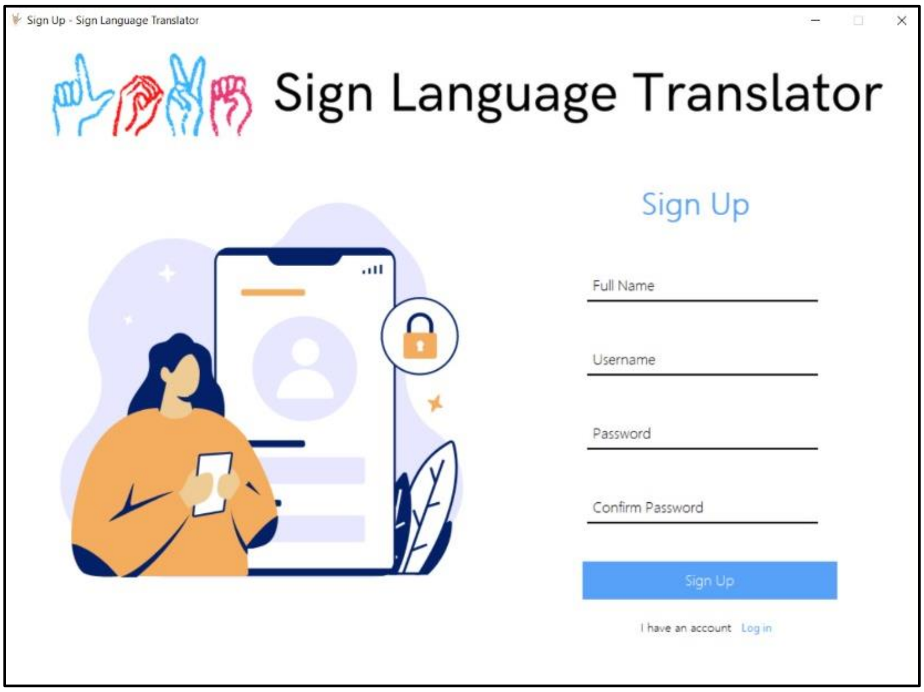

Fig 3.3.2 : SignUp UI

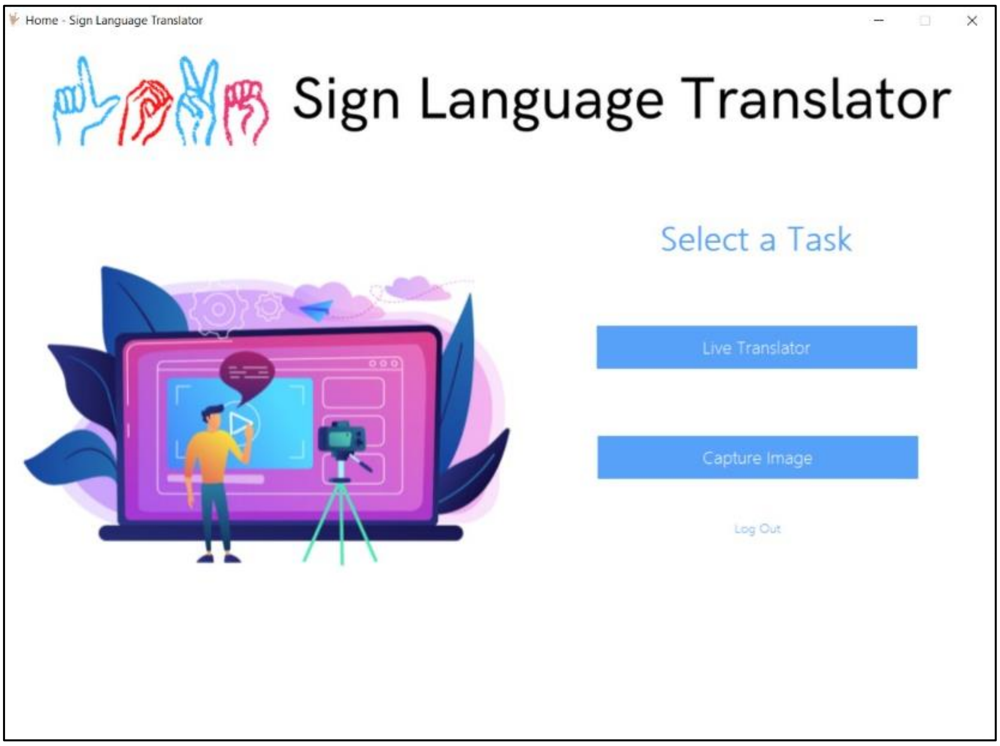

Fig 3.3.3 : Options Window

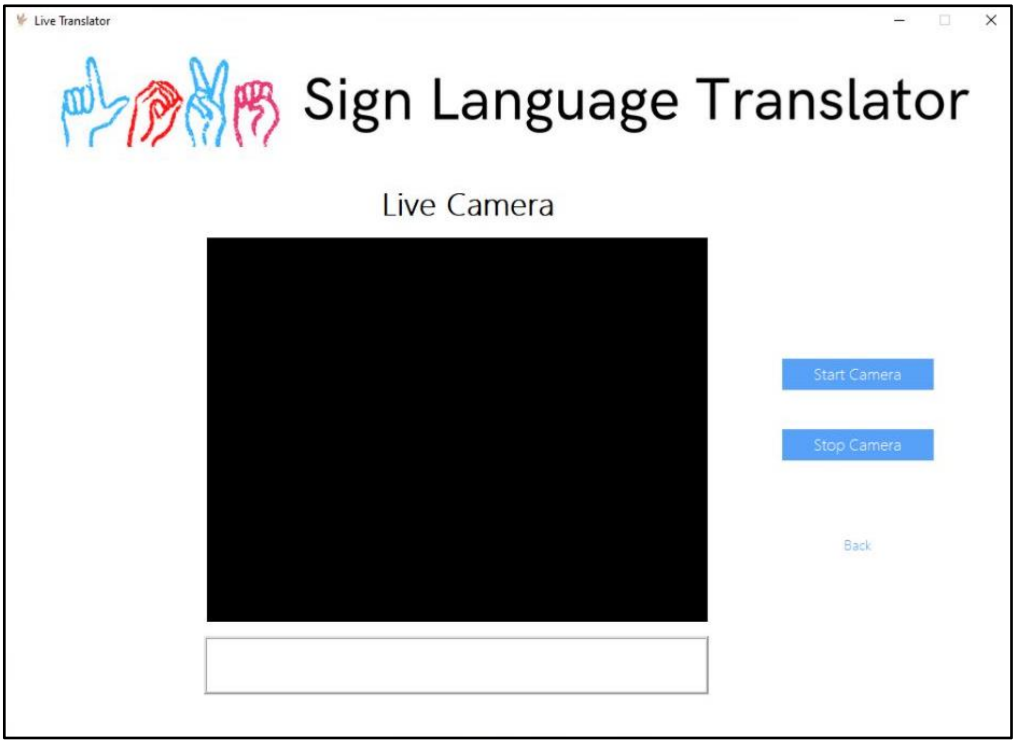

Fig 3.3.4 : Live Translator Window (Off Camera)

3.4 Integrating model with Desktop Application

It provides high-level APIs that help transform raw input data into the form required by the model, and interpret the model's output, reducing the amount of boilerplate code required. It supports common data formats for inputs and outputs, including images and arrays. It also provides pre- and post-processing units that perform tasks such as image resizing and cropping.

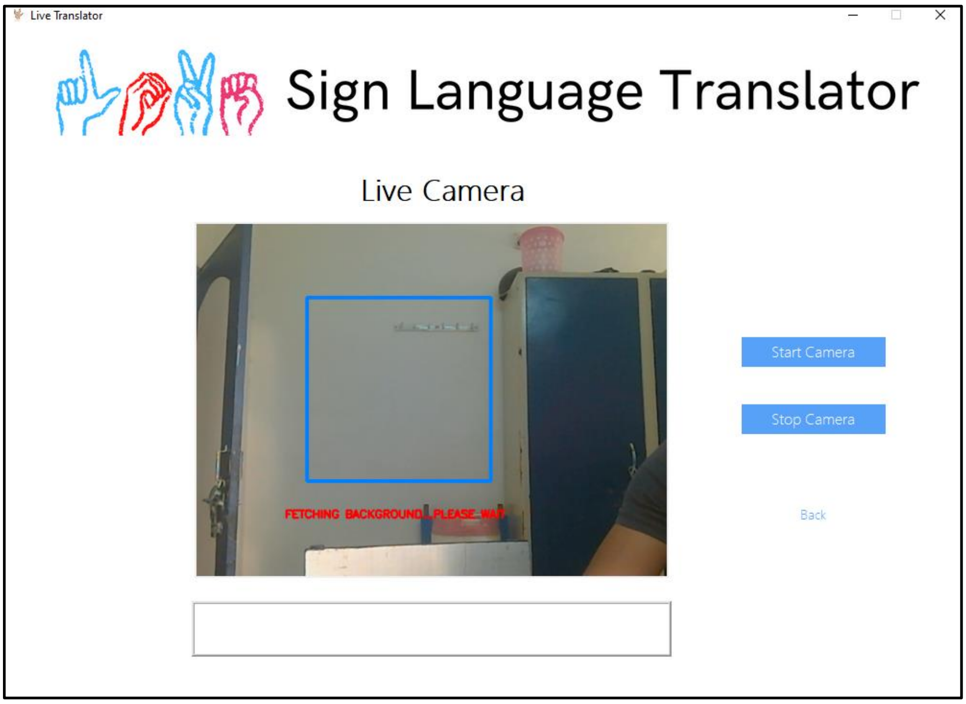

Fig 3.4.1 : Live Translator Window (On Camera)

3.5 Results and Discussions

The project reflects a potential solution towards Hand Sign Language Recognition and Translation to Audio and Text. It is preferred that hand signs are made on plain background and it neither too far nor too nearby camera, so that the predicted output is accurate. Currently project works on some the statements or words but for the real-world purpose, we have to add more classes and more samples to get accurate output.

Hypothesis for the behaviour of application can be made as, the input data provided for training model is not considering in all circumstances and it might be possible that user can use this with abstract background and also it is possible that all users has their own different shape and size of hands, so the output may be inaccurate. Despite of above hypothesis that presents special conditions, this project puts light over a great solution against the problem. It showcases the way of using the current technology, the algorithms within it and how to train the machine for such scenarios. The various algorithms that are capable and can also be developed to increase their efficiency for better prediction. In the control testing environment, they have shown a great accuracy.

Way of presenting it make it more operable for the user by providing good UI design and controlled options over the application. With more advancement the predictions can be taken close to reality.

Chapter – 4
FUTURE SCOPE & CONCLUSIONS

4.1 Summary and Conclusions

Our project is about a system which can support the communication between deaf and ordinary people. The aim of the project is to provide a complete dialog without knowing sign language. Earlier communication is not so important, but today in the growing world communication is the important term for personal and career success. So this is a project which help Deaf and Dumb people to communicate with others and put their thoughts in the real world. This project provides portion of most effective and a reliable solution as per current technology advancements by providing a mobile application where user can capture the hand signs and that is translated to Text and Audio with the help of Machine Learning and Image Processing.

Considering today's technology and needs this project may overcome the problem but an effective solution must be required by improving this project which is helpful to this needy people. Because every person has their opinion about the real-world problem and it might be possible that this people can give an idea to solve this real-world problems. So, it is good for us to make a communication between these people's and normal peoples.

4.2 Future Scope

1. Since deaf people are usually deprived of normal communication with other people, they have to rely on an interpreter or some visual communication. Now the interpreter can not be available always, so this project can help eliminate the dependency on the interpreter.
2. The system can be extended to incorporate the knowledge of facial expressions and body language too so that there is a complete understanding of the context and tone of the input speech.
3. A mobile and web based version of the application will increase the reach to more people.
4. Integrating hand gesture recognition system using computer vision for establishing 2-way communication system.

Bibliography

[1] L. Kau, W. Su, P. Yu and S. Wei, "A real-time portable sign language translation system," 2015 IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS), Fort Collins, CO, 2015, pp. 1-4, doi: 10.1109/MWSCAS.2015.7282137.

[2] Automated Sign Language Interpreter, 2018 Eleventh International Conference on Contemporary Computing (IC3), Department of Computer Science and Information Technology, JIIT, Noida.

[3] Vision-based sign language translation device, 2013 International Conference on Information Communication and Embedded Systems (ICICES), Department of Biomedical Engineering, SRM University, Kattankulathur, TamilNadu, India. [4] Sign language interpreter using a smart glove, 2014 International Conference on Advances in Electronics Computers and Communications, DOI: 10.1109/ICAECC.2014.7002401.

[4] Sign language interpreter using a smart glove, 2014 International Conference on Advances in Electronics Computers and Communications, DOI: 10.1109/ICAECC.2014.7002401.

[5] A machine learning based approach for the detection and recognition of Bangla sign language, 2016 International Conference on Medical Engineering, Health Informatics and Technology (MediTec).

[6] Indian Sign Language (ISL) Translation System For Sign Language Learning M. Jerin Jose , V. Priyadharshni , M. Suresh Anand , A. Kumaresan , N. Mohan Kumar

[7] Tensorflow Lite Model Building. Available: https://www.tensorflow.org/lite/guide

[8] “Communication – the human connection is the key to personal and career success” – Paul J. Meyer Available: https://www.brainyquote.com/quotes/paul_j_meyer_190945

[9] R. San Segundo, B. Gallo, J. M. Lucas, R. Barra-Chicote, L. F. D'Haro and F. Fernandez, "Speech into Sign Language Statistical Translation System for Deaf People," in IEEE Latin America Transactions, vol. 7, no. 3, pp. 400- 404, July 2009, doi: 10.1109/TLA.2009.5336641.

[10] Bhargav Hegde, Dayananda P, Mahesh Hegde, Chetan C, “Deep Learning Technique for Detecting NSCLC”, International Journal of Recent Technology and Engineering (IJRTE), Volume-8 Issue-3, September 2019, pp. 7841-7843.

[11] Heet D. Patel, Avneet Saluja, “Sign Language Recognition and Translator Application”, International Research Journal of Engineering and Technology (IRJET), Volume: 08 Issue: 09, September 2021, e-ISSN: 2395-0056, p-ISSN: 2395-0072.