A PROJECT REPORT ON

Sign Language Detection System

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)

SUBMITTED BY

PRANJAL JAGDISH PATIL	EXAM ROLL NO. B150054429
SUMEET SANWAL	EXAM ROLL NO. B150054480
RIDDHI SHANKAR TAKAWALE	EXAM ROLL NO. B150054484

UNDER THE GUIDANCE OF
PROF.MANISH JANSARI

DEPARTMENT OF COMPUTER ENGINEERING
PUNE INSTITUTE OF COMPUTER TECHNOLOGY
DHANKAWADI, PUNE – 43

SAVITRIBAI PHULE PUNE UNIVERSITY
2021-2022

CERTIFICATE

This is to certify that the project report entitles

"Sign Language Detection System"

Submitted by

Pranjal Jagdish Patil	EXAM ROLL NO. B150054429
Sumeet Sanwal	EXAM ROLL NO. B150054480
Riddhi Shankar Takawale	EXAM ROLL NO. B150054484

are the bonafide students of this institute and the work has been carried out by them under the supervision of **Prof. Manish Jansari** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

Prof. Manish Jansari
Guide
Dept. of Computer Engineering

Dr. Geetanjali V. Kale
Head,
Dept. of Computer Engineering

Dr. R. Sreemathy
Principal,
Pune Institute of Computer Technology

Place: Pune
Date:

ACKNOWLEDGEMENT

It gives us great pleasure in presenting the project report on 'Sign Language Detection System'.

We would like to take this opportunity to thank **Prof. Manish Jansari** for giving us all the help and guidance we needed. We are really grateful to him for his constant support, encouragement and motivation. His valuable suggestions were very helpful.

We are also grateful to **Dr. Geetanjali V. Kale**, Head, Department of Computer Engineering, PICT for her indispensable support and suggestions.

Pranjal Patil
Sumeet Sanwal
Riddhi Takawale

ABSTRACT

Creating a web application that uses a webcam to capture a video of a person signing gestures and translate it into corresponding text and speech in real time. The translated sign language gesture will be acquired in text which is further converted into audio providing more ease to the user in communicating. In this manner we are developing a video sequence sign language translator system.

We propose a modified short-term memory (LSTM) for continuous gesture sequence or continuous SLR that detects connected gesture sequences. It is based on dividing continuous signals into sub-units and connecting them with neural networks. Thus, the consideration of a different combination of sub-units is not required during training.

Keywords

LSTM, Continuous SLR, Joint Keypoints,RNN

Contents

List of Abbreviations

Abbreviation	Full form
SLR	Sign Language Recognition
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Overview

American Sign Language (ASL) is a natural syntax with the same etymological properties as spoken languages, with a completely different grammar, ASL can be expressed through the creation of physical actions. In the Americas, people who are deaf, a reliable source of irrationality. There is no official or standard form of sign language. Languages of different symbols guess in certain places.

As the number of deaf people increased, so did the need for translators. Reducing the communication gap between listening to people with disabilities and becoming a desire to have some effective communication between them all. Sign-language translation is one of the fastest growing forms of research these days and is a natural way of communicating with people with hearing impairments. A hand gesture device that can detect deaf people may be able to communicate with the deaf without the need for an interpreter. The system is designed to automatically convert ASL to text and speech content.

1.2 Motivation

The intent to reduce efforts and make communication easier and convenient for mute using the proposed system which intends to understand some very fundamental elements of signal language and to translate them to text and speech in real time helping them understand and communicate better without the translator.

1.3 Problem Definition and Objectives

1.3.1 Problem Definition

This work is focused on providing a real time communication system to serve mute by Creating a desktop application that uses a webcam to capture a video of a person signing gestures and translate it into corresponding text and speech in real time making communication easier and instantaneous for hearing and speech impaired.

1.3.2 Objectives

- To provide an efficient and accurate way to convert sign language into text and speech for the hearing impaired.
- To reduce efforts and make communication easier and convenient for mute.

1.4 Project Scope and Limitations

1.4.1 Project Scope

The study focuses on developing a working app that could be used by mute to communicate what they want to say without the need of a human translator.

Convolutional neural networks are great for a 1 to 1 relation; given an image of a sign, it generates fixed-size labels, like the class of the sign in the image. However, What CNNs cannot do is accept a sequence of vectors. That's where Recurrent Neural Networks (RNNs) are used. RNNs allow us to understand the context of a video frame, relative to the frames that came before it. We use a special type of rnn called lstm that allows our network to learn long term dependencies.

It not just recognizes individual words but fluffy fledged sentences with proper grammar. We propose on using A special kind of rnn model called lstm which that allows our network to learn long term dependencies and hence form proper sentences by alinging probability to each possible sentence and then choose the one with the highest probability as the response making it highly efficient in forming sentences from real time sign language.

1.4.2 Limitations

Some of the limitations of the proposed framework are -

- The image takes consideration of 2D image frames and not 3D image frames. swipe to wave, circle to go, Keytap to walk, Screencap to advance uses x, y, and z coordinated, whereas our model is limited to x and y coordinates.
- The System Recognition is limited to smaller set of sign language action to achieve a higher accuracy.

1.5 Methodologies Of Problem Solving

A Neural Network is a machine learning is based around the human brain, thus we create an Artificial Neural Network using algorithms enabling the computer to learn by incorporating new data and try to behave like human brain. The units learn how to convert input into the desired output, e.g. picture of a hand gesture is an input while the label of gesture recognized is an output, forming the basis of automated recognition. The computer learns from training data, which were labelled initially. A task for a neural network is to recognize an object, where it is a large collection of frames(images) obtained from videos belonging to a certain class, such as a hand sign and the computer analyzes the patterns using key points of hands, face ,body shape in the images fed to it, which helps it to classify when new images are fed.

Model construction

- It includes the following steps :
  - Using OpenCV to connect with the user
  - Collecting key points of hands, face and body shape
  - Optimizing the key points
  - Creating training data to process in the model
  - Storing and labeling the data in folders
  - Feature extraction and optimization using LSTM

Model training

- Model is trained once it has been constructed appropriately. It includes training of model using training data and expected output for this data.
- It's look this way: model.fit(training_data, expected_output).

Model Testing

- In this phase a another set of data is loaded.It is used to verify accuracy truly.This data set contains unseen data that has never been seen by model.Model can be saved after the model training is complete, and it is understood that the model shows the right result.

Model Evaluation

- Finally, the saved model can be used in the real world. This means that the model can be used to evaluate new data.

CHAPTER 2

LITERATURE SURVEY

2.1 Literature Survey

Fully Convolutional Networks for Continuous Sign Language Recognition

Ref: [1] Non-stop signal language recognition (SLR) is a tough challenge that calls for gaining knowledge of on each spatial and temporal dimensions of signing body sequences. maximum recent work accomplishes this through the usage of CNN and RNN hybrid networks. but, training these networks is normally non-trivial, and maximum of them fail in studying unseen sequence styles, inflicting an unsatisfactory performance for on line reputation. on this paper, we advise a fully convolutional network (FCN) for on line SLR to simultaneously examine spatial and temporal features from weakly annotated video sequences with most effective sentence-stage annotations given. A gloss feature enhancement (GFE) module is delivered in the proposed community to put into effect higher series alignment studying. The proposed network is end-to-cease trainable without any pre-training. We conduct experiments on large scale SLR datasets. Experiments display that our technique for continuous SLR is powerful and plays well in on line recognition.

Sign language recognition

Ref: [2] This paper introduces a novel program to assist in communicating with those who have a voice and hearing impairment. Discusses advanced sign language recognition and grammar. The designed algorithm is capable of extracting signals from video programs under a dense and flexible background using skin color separation. It distinguishes between vertical and shift touches and releases the appropriate feature vector. These are divided using Vector Support Machines. Speech recognition is built on a standard module - the Sphinx. The test results show a satisfactory distinction of symptoms under various backgrounds and high accuracy in comparison with touch and visual perception.

A Modified LSTM Model for Continuous Sign Language Recognition Using Leap Motion

Ref: [3] Sign language facilitates communication between the deaf and the general public. A number of sign language (SLR) systems have been developed by researchers, but they are limited to just one touch of signals. In this paper, we propose a modified short-term memory (LSTM) modified continuous touch sequence or continuous SLR that detects connected touch sequences. It is based on dividing continuous signals into sub-units and comparing them with sensory networks. Therefore, consideration of a different combination of smaller units is not necessary during training. The proposed system has been tested in 942 signed Indian Sign Language (ISL) sentences. These punctuation marks are identified by using 35 different punctuation marks. 72.3% accuracy and 89.5% accuracy are recorded in signed sentences and individual punctuation marks, respectively.

Real time Indian Sign language recognition

Ref: [?] Sign languages are a collection of previously defined languages that use sign language to convey information. We are considering the problem of recognizing finger spelling in real-time Indian Sign Language (ISL). We collected a deep RGB image data set using the

Microsoft XBOX360 Kinect Camera to separate 36 different touches (alphabets and numbers). The system detects hand gestures as embedded and returns the corresponding visual character as a real-time exit on the monitoring screen. In classification we used the Deep Convolutional Neural Network and gained 89.30% accuracy.

Real-Time Translation of Indian Sign Language using LSTM

Ref: [4] Sign language is the only medium of communication with a speech impediment while everyone communicates verbally. The project aims to close this communication gap by proposing a new way to translate dynamic and dynamic signals into Indian Sign Language and transform them into speech. A sensor glove, with flexible sensors to detect the flexion of each finger and IMU to learn hand shape, is used to collect data about actions. This data is then transmitted wirelessly and segmented into the results of the corresponding speech. LSTM networks are explored and used for segmentation of touch data due to their ability to learn long-term dependencies. The designed model can distinguish 26 touches with 98% accuracy, indicating the feasibility of using LSTM-based LSTM sensor networks for sign language translation.

Table 2.1: Literature survey

Paper	**Overview**	**Technology used**	**Feature**
Fully Convolutional Networks for Continuous Sign Language Recognition [1]	The proposed network does not need any pre-training and is end to end trained. They have conducted experiments on two large scale SLR datasets.	CNN ,RNN, fully convolutional network (FCN), gloss feature enhancement (GFE)	Proposes a FCN for online SLR to regularly learn spatial and temporal features from loosely combined video sequences
Real-Time Translation of Indian Sign Language using LSTM	This paper proposes a novel approach to interpret the static and dynamic signs in the Indian Sign Language and convert them to speech.	LSTM, sensor glove,Gesture recognition	The designed model can classify 26 gestures with an accuracy of 98%, showing good results by LSTM model for SLR.
Sign language recognition	This paper presents a system to help in communicating for hearing and speaking disabled.	Image processing, Speech Recognition, Gesture recognition	Distinguishes between static and dynamic gestures and extracts the appropriate feature vector and relatively high accuracy in gesture and speech recognition.
A Modified LSTM Model for Continuous SLR Using Leap Motion	This paper proposes a modified long short-term memory (LSTM) model for continuous SLR that recognizes a sequence of connected gestures.	LSTM,SVM,radio frequency identification	Modified long short-term memory (LSTM) model for continuous SLR that recognizes a sequence of connected gestures.
Real time Indian Sign language recognition	The paper presents a system which takes in a hand gesture as input and returns the corresponding recognized character as output in real time on the monitor screen.	CNN	Predicts the 'alphanumeric' gesture of the ISL system by using segmented RGB hand-gestures which shows good accuracy while predicting results both offline and online.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Assumptions & Dependencies

3.1.1 Assumptions

1. The application will detect the ASL action and predict the meaning into text based on probabilities.

2. Improved method for sign language recognition and conversion of interpreted text to speech.

3.1.2 Dependencies

1. The interaction of user with camera and the Program will be implemented using OpenCV Library.

2. The keypoints of hands face and pose will be extracted using mediapipe Holistic Library.

3. the training of model will be done using Keras tensorflow library and LSTM layers.

4. The plotting of the datapoints is achieved using matplotlib library to get a visual aspect of data set.

3.2 Functional Requirements

- **T**his feature will translate the recognized gesture into the textual meaning of the gesture and display the translated text to the user.
- **T**his feature will also have an option to play an audio of the recognized gesture.
- **S**ystem will recognize the appropriate movement of the hands,face,pose and will search its database to match the movement with the pre-defined gestures. After matching system, will add the meaning of the sign to the opened file.

  The system requirements can be specified as follows:

- User should be able to perform sign that represent digit number
- User should be able to perform sign that represent alphabet letter
- User should be able to perform sign that represent word
- User should be able to perform sign that represent sentence
- User should be able to see the translation of sign to text or sign to speech

  Normal Flow of Events:

- User selects the communication mode
- User opens a camera
- User performs the movement
- Gesture is recognized and has a match
- The text is added to the file and displayed to the user and also speech out the words or sentence

3.3 External Interface Requirements

3.3.1 User Interface

People who want to communicate with the dumb(mute) and are not familiar with the sign language. They can use this application to interpret what the other person is saying. It is achieved by capturing the action of the person.

3.4 Non-Functional Requirements

3.4.1 Performance Requirements

- Real time - the system should provide the recognition of signs and their translation to speech in an unnoticeable time for its users.

- Accuracy – signs should not be confusing and the system should recognize appropriate sign.
- Environment – the system should provide real time recognition with high accuracy in low light conditions as well.

3.4.2 Security Requirements

- The system should provide natural interaction to its users. The hearing impaired person needs to worry nothing else, just for performing signs.

3.4.3 Software Quality Attributes

The following are some key quality characteristics that will be important to objectively judge the end result and feasibility of the product:

- Extensibility
- Reliability
- Accuracy
- Robustness
- Maintainability
- Intuitive GUI
- Availability

3.5 System Requirements

3.5.1 Database Requirements

We have created our own database by recording 150 video frames for each sign.

3.5.2 Software Requirements

- Operating System:Windows,Mac,Linux
- SDK: OpenCV,TensorFlow,Keros,Numpy

3.5.3 Hardware Requirements

The Hardware Interfaces Required are:

- Camera:Good quality,3MP
- Mouse: Scroll or Optical Mouse or Touch Pad
- GPU: 4GB dedicated

- Ram: Minimum 8GB or higher
- HDD: 10GB or higher
- Processor: Intel Pentium 4 or higher
- Keyboard: Standard 110 keys keyboard

3.6 Analysis Models : SDLC Model To Be Applied

SDLC is a process which is followed for any software development project, within an organization. This project makes use of the Agile model. Agile model refers to a development process which is a combination of incremental and iterative models. This type of software development model is basically used for the project which has small incremental builds.

1. Brainstorming on the Problem Statement: In this phase, we understood and discussed the problem statement with our project guide. We studied the basics of Deep Learning and Neural Networks and thoroughly analyzed our problem statement and it's scope. Requirements were revisited every month to ensure rechecking of the requirements.

2. Requirement Analysis: During this phase, we gathered all the requirements of our system and decided to explore various languages, frameworks, tools, softwares, API's and developed Data Sets we can use in our project. During this phase, the System Requirement Specification (SRS) was prepared as per the given format consisting of all the different requirements such as system requirement, funtional requirements, system requirements(both H/W S/W), system requirements,etc.

3. Design: In this phase, the technical details of the project were decided. We worked on the system design of our application and decided how various components and modules of the application would interact with each other. We also designed the structure of the LSTM architecture which would be used to create the model . Thus, the overall design of the system was done in this stage. class design, data flow design along with the system architecture.

4. Implementation: After the design stage, we started with the implementation stage, where the actual coding of the application was done. In this phase, coding of Sign Language Recognition System was carried out by using the technology CNN and RNN discussed in the previous phase. This phase consisted of coding of Web based application, and the model of the application. First, the Data set were loaded according to the requirements. Then, implementation of the trained model and Web Application started.

5. Testing: This phase was carried out after coding out the respective modules. Unit tests were done to test the accuracy of the application model. Web application was tested manually individually as well as after their integration. Edge cases were also highlighted out and handled successfully.

6. Deployment: In the deployment phase, we made sure that the environment was setup correctly and we deployed the application modules to their respective environments. Once the application was deployed on Flask, any additional requirements or bugs experienced at run-time were also fixed.

CHAPTER 4

SYSTEM DESIGN

4.1 System Architecture

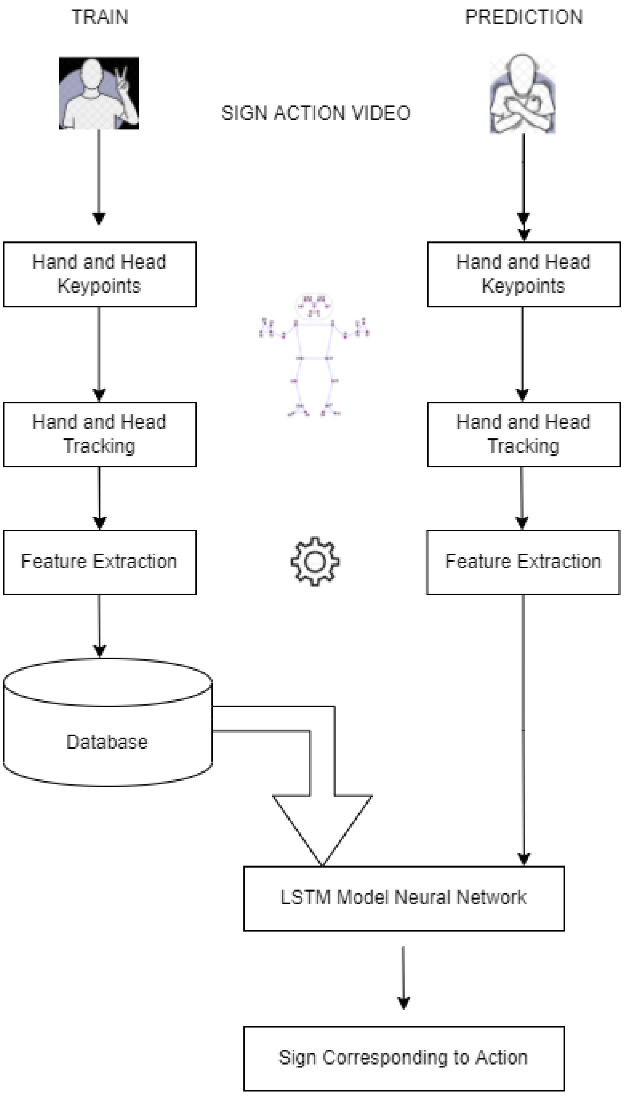

Figure 4.1: System Architecture Diagram

4.2 Module Split Up

In this section, we introduce our proposed sign language recognition system that aims at recognizing person signing gestures for ASL and translate it into text and speech.

4.2.1 Web Application

The web application provides the user to record a video and upload it for the model to process and generate a message in the form of text and audio. It also provides the user with the functionality to play the audio generated by the model.

4.2.2 Model

The LSTM Model uses the video provided to it by the user to be broken into frames from which it analyses the key points of hands, face and body shape. It further optimizing the key points to obtain high probability in determining the message from the video by processing each frames and generate the message with the highest probablity score which is then converted to audio and send back to the web application with the help of an API.

4.3 Data Flow Diagram

The data flow diagram illustrates the flow of data throughout the life time of the application and that is as follows.

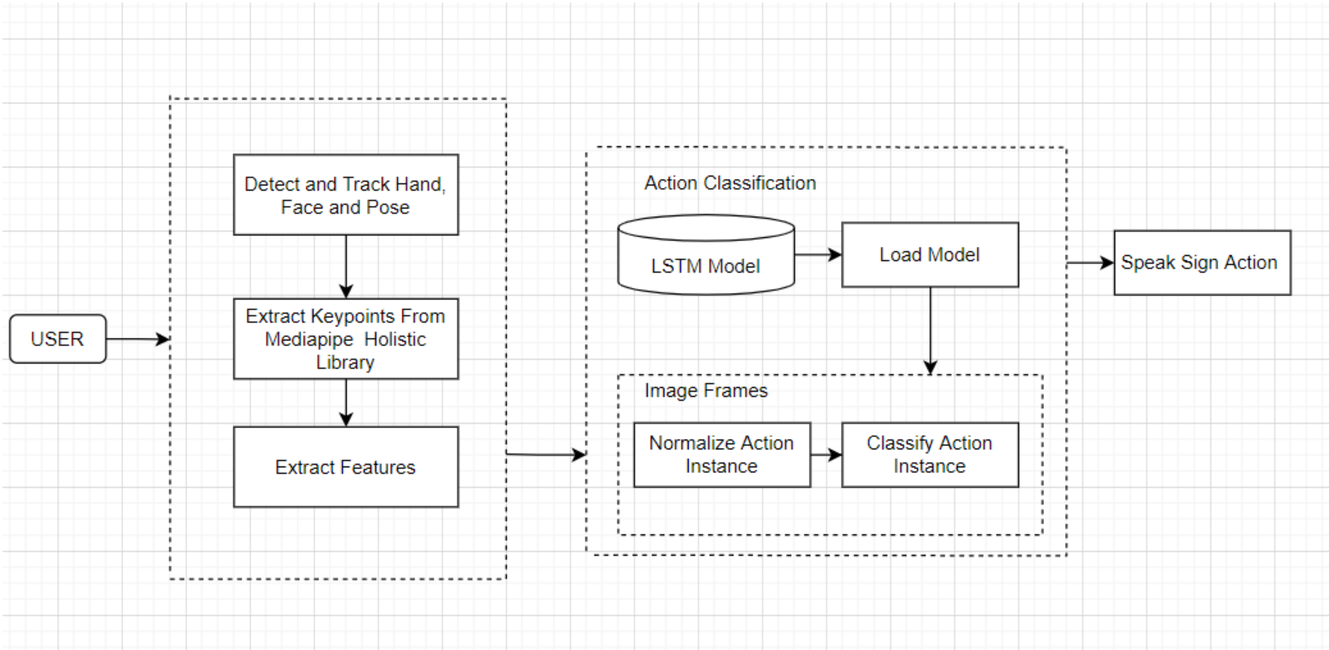

Figure 4.2: Dataflow Diagram

4.4 UML Diagrams

4.4.1 Class Diagram

The components of the application can be divided into three main components and illustrated below:

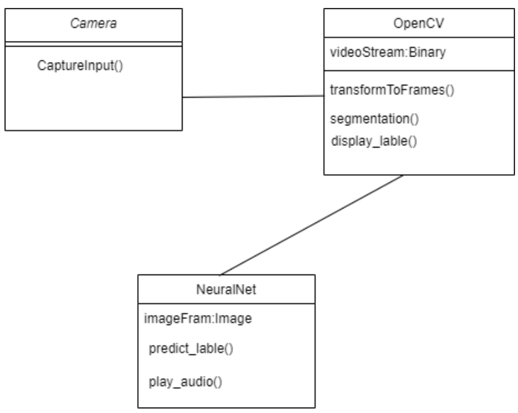

Figure 4.3: Class Diagram

4.4.2 Use Case Diagram

The major use cases of the application are illustrated here , with users and system as the actors interacting with different modules.

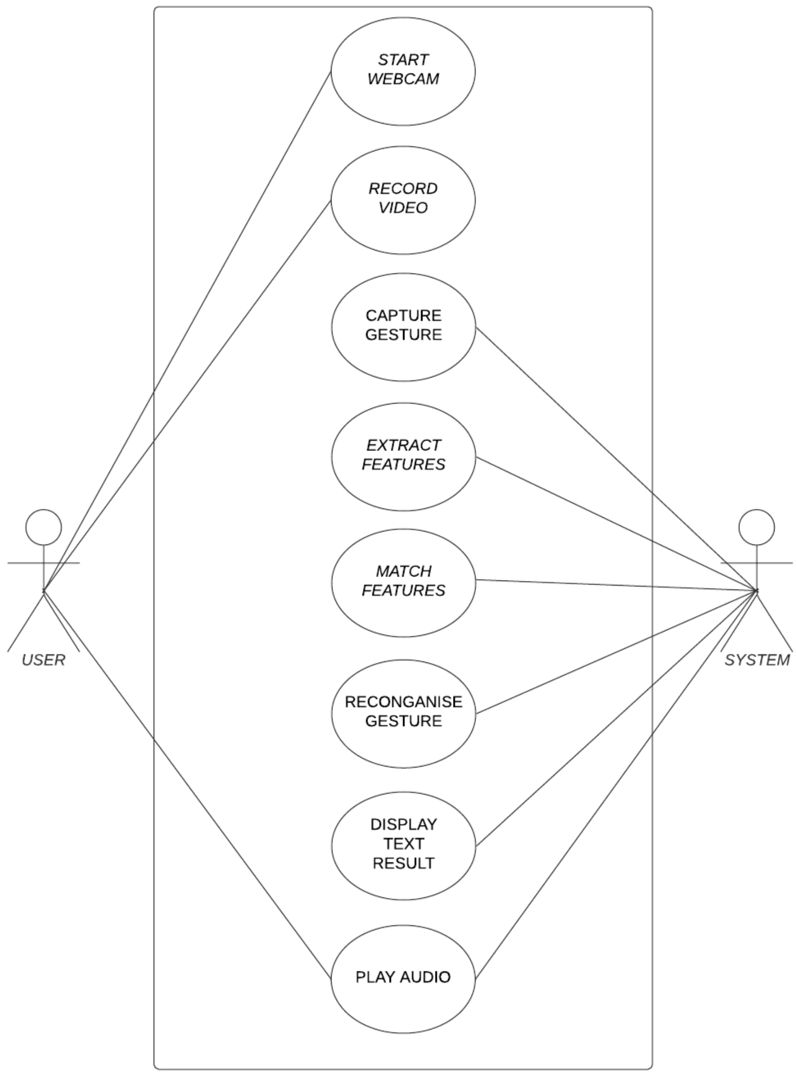

Figure 4.4: Use Case Diagram

4.4.3 Sequence Diagram

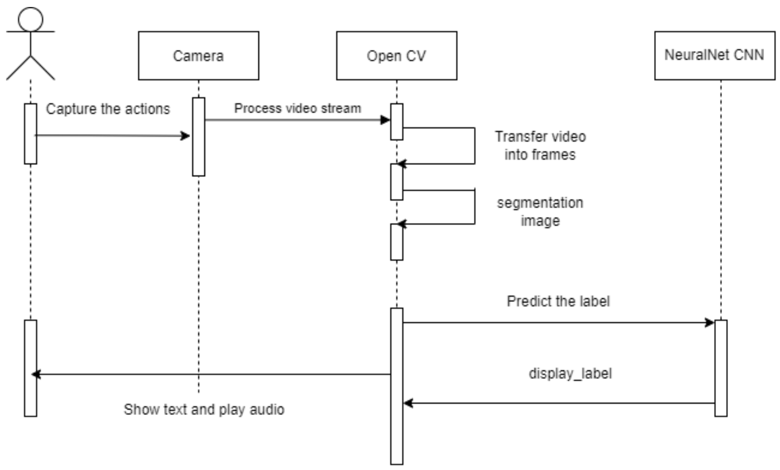

Figure 4.5: Sequence Diagram with Major Functionalities

CHAPTER 5

PROJECT PLAN

5.1 Project Estimate

5.1.1 Time Estimates

- Finalizing the Problem Statement: August 2021
- Designing the System Architecture: September 2021
- Implementation of the system: November 2021 - February 2022
- Testing: March 2022
- Deployment: April 2022

5.1.2 Reconciled Estimates

- Cost Estimate: The software used for the application will incur no cost as it is all released under free open source licenses.

5.1.3 Human Resources

- Number of people: 3
- Skills: Python,Machine Learning,CNN,RNN.
- Client: In our project, the users can belong to any age group, or section
- Stakeholders: Stakeholders are often members of the project team. Considering the project, stakeholders will be the people who will be using the application, i.e, the clients along with project team.

5.1.4 Development Resources

- Hardware: i5 Processor, Minimum 8GB of RAM, At least 2GB of free disk space
- Software: Linux/Windows OS, Solidity, python, Web Browser (preferably Google Chrome or Firefox)

5.2 Risk Management

Risks are a natural part of developing applications. While risk is normally affiliated with negative connotations, a mature understanding of risk defines it as an essential element that helps corporates to seize new opportunities and conquer new frontiers, that is, if risk is handled properly. Hence, the management of risk becomes an important part of developing an application.

5.2.1 Risk Identification & Analysis

The risk analysis is performed using the following guidelines:

Risk Probability

a.	High Probability	$75\% \leq x \leq 100\%$
b.	Medium High Probability	$50\% \leq x \leq 75\%$
c.	Medium-Low Probability	$25\% \leq x \leq 50\%$
d.	Low Probability	$0\% \leq x \leq 25\%$

Table 5.1: Risk Probability Levels

Risk Impact

a.	Very High	Catastrophic
b.	High	Critical
c.	Medium	Moderate
d.	Low	Marginal

Table 5.2: Risk Impact Severity

5.3 Project Schedule

5.3.1 Project Task Set

- Literature Survey (T1)
- Creating Development Environment (T2)
- Developing Model (T3)
- Training and Testing Model(T4)
- Developing Frontend (T5)
- Integration of Frontend and Backend (T6)
- Drafting Documentation (T7)

5.3.2 Task Network

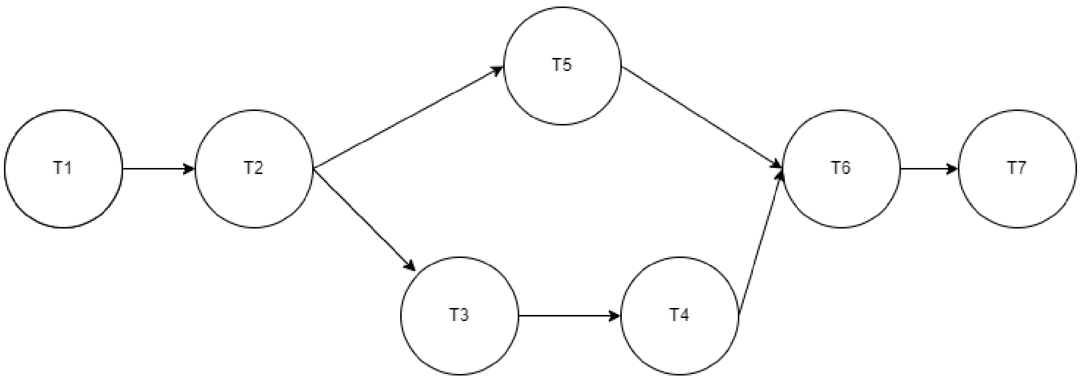

Figure 5.1: Task Network

5.3.3 Timeline Chart

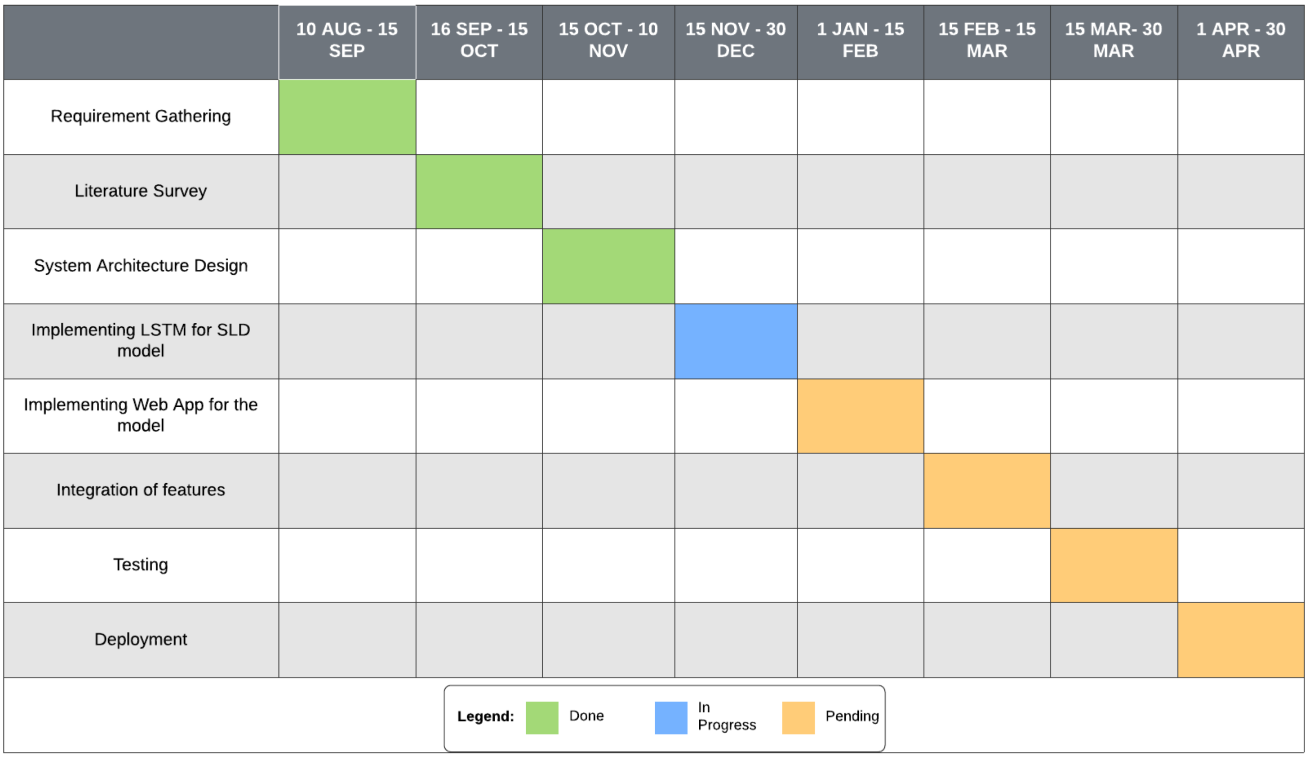

Figure 5.2: Timeline Chart

5.4 Team Organization

The project will be implemented by three developers working as a team, team structure and roles for the project is identified.

5.4.1 Team Structure

Table 5.3: Team Structure

Roles	Responsibilities	Participant(s)
Project Guide	1. Manages project in accordance to the project plan. 2. Reviews project elements approved by the company. 3. Provides overall project direction. 4. Directs team members toward project objectives	Prof.Manish Jansari
Project Participant	1. Communicate project goals, status and progress throughout the project to personnel in their area. 2. Identify risk and issues Identify functional and non-functional requirements.	Pranjal Patil Sumeet Sanwal Riddhi Takawale

5.4.2 Management reporting and communication

- Face to Face discussion in college (PICT)
- Discussion with Project Guide
- Keeping track of implementation in Log Book
- Discussions via google meets
- Discussions via email

CHAPTER 6

PROJECT IMPLEMENTATION

6.1 Overview of Project Modules

Web Application

The web application provides the user to record a video and upload it for the model to process and generate a message in the form of text and audio. It also provides the user with the functionality to play the audio generated by the model.

Model

The LSTM Model uses the video provided to it by the user to be broken into frames from which it analyses the key points of hands, face and body shape. It further optimizing the key points to obtain high probability in determining the message from the video by processing each frames and generate the message with the highest probablity score which is then converted to audio and send back to the web application with the help of an API.

Model construction

- It includes the following steps :
  - Connecting user to program using OpenCV
  - Collecting key points of hands, face and body shape
  - Extracting the key points
  - Creating training data to be processed in the model
  - Storing and labeling the data in folders
  - Feature extraction and optimization using LSTM

6.2 Tools And Technologies Used

- Jupyter NoteBook
- LSTM
- CNN
- RNN
- Streamlift
- Image Recognition

6.3 Algorithm Details

Recurrent neural network

Recurrent neural networks (RNN) uses Sequential data.RNNs are a class of neural networks.It is helpful in modeling sequence data.RNNs are derived from feedforward networks and they exhibit similar behavior to how human brain functions.In Simple words ,recurrent neural networks produce predictive results in sequential data that other algorithms can't. In a RNN the information cycles through a loop. When it makes a decision on the basis of the current input and also from the inputs it received previously.

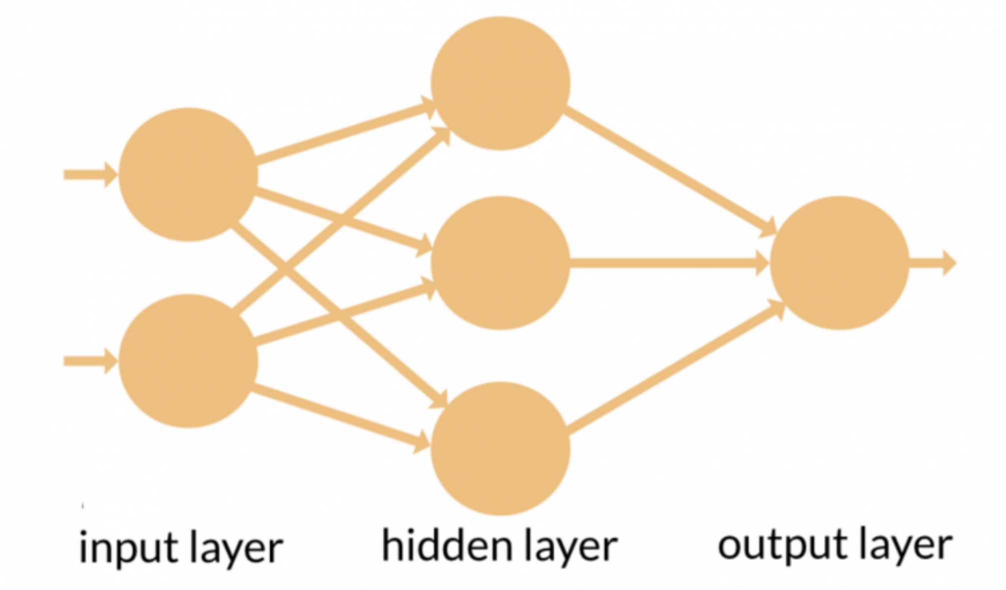

Figure 6.1: Recurrent Neural Networks

Long Short Term Memory

The Long Short Term Memory(LSTM) Model uses the video provided to it by the user to be broken into frames from which it analyses the key points of hands, face and body shape. It further optimizing the key points to obtain high probability in determining the message from the video by processing each frames and generate the message with the highest probablity score which is then converted to audio and send back to the web application with the help of an API.

CHAPTER 7

SOFTWARE TESTING

7.1 Types Of Testing

The different levels of testing strategies are applied at different phases of software development in order to make sure that the system does not have errors This testings are :

Unit Testing

Unit testing is performed on individual models as they are completed and becomes executable. It is confined only to the designer's requirements.It is different from and should be preceded by other techniques, including: Inform Debugging,Code Inspection.

Black Box testing

This testing consists of some test cases are generated as input conditions that fully execute all functional requirements for the program.

White Box testing

In this testing,the test cases are generated on the logical decisions and the logic of each module by drawing flow graphs of that module are tested on all the cases. It is used to generate the test cases in the following cases:

- Execute internal data structures to ensure their validity.
- Execute all loops at their boundaries and within their operational bounds.
- Guarantee that all independent paths have been executed

Integration Testing

In Integration testing software and subsystems work together a whole. It tests the interface of all the modules just to make sure that when models are integrated together they behave properly. It is performed by developers, especially at the lower, module to module level. Testers work in higher levels.

System Testing

It contains house testing of the entire system before delivery to the user. The target is to satisfy the user the system meets all requirements of the client's specifications.If a company has one testing organization then it conducts the tests. Test data may range from and generated to production. Requires test scheduling to plan and organize:

- Inclusion of changes.
- Test data to use.

One common approach is graduated testing: as system testing progresses and fewer and fewer defects are found, the code is frozen for testing for increasingly longer time periods.

Acceptance Testing

Acceptance Testing is a pre-delivery testing.In this testing entire system is tested at client's site on real world data to find errors.

7.2 Test cases & Test Results

Table 7.1: Test Verification

Test case	**Input description**	**Expected output**	**Test status**
Loading model	Initializing trained model and starting the application	Loaded model without errors	pass
Converting video to frames	Capturing video and converting it into frames	Image frames of captured video	pass
Recognize hand gesture	Image frame that contains face and hand object	label	Pass

CHAPTER 8

RESULTS

8.1 Outcome

Our project on sign language recognition system is used to recognize sign language actions.It can detect not only image signs gestures but also sign actions.It gives the output in the form of text as well as in speech.

8.2 Screen Shots

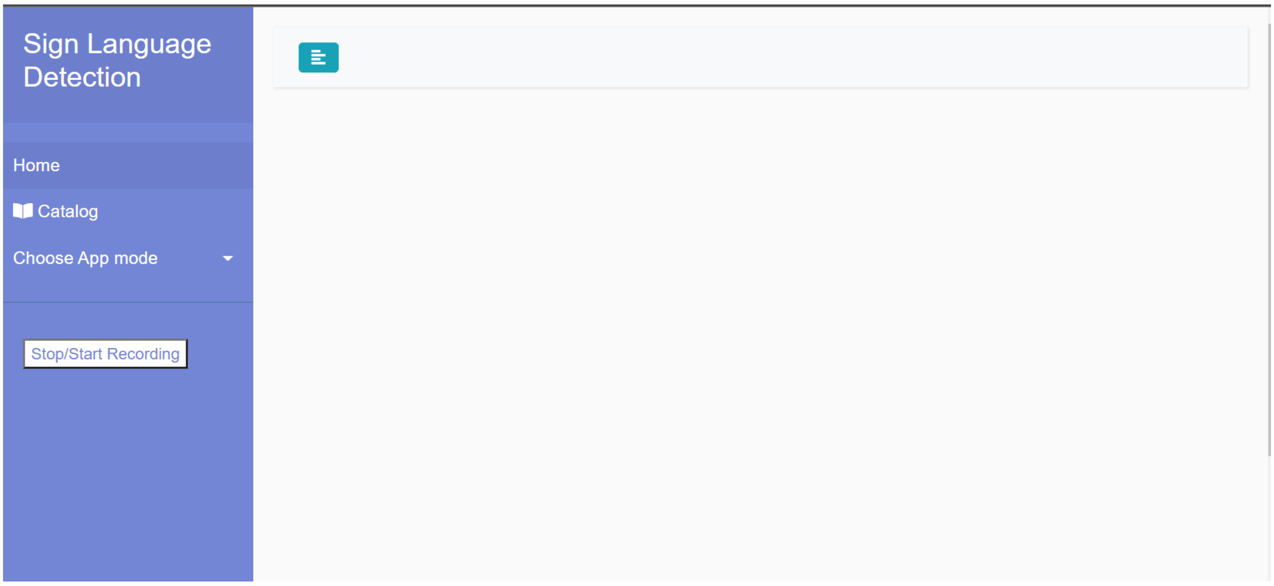

Figure 8.1: Home Page

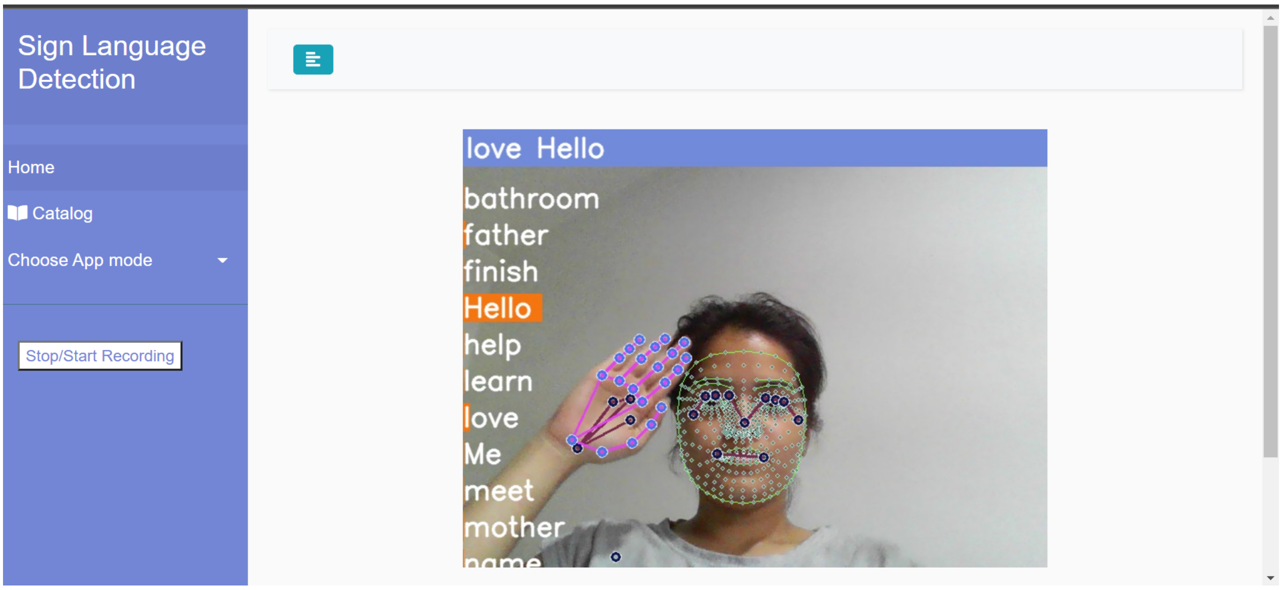

Figure 8.2: Sign Detection 1

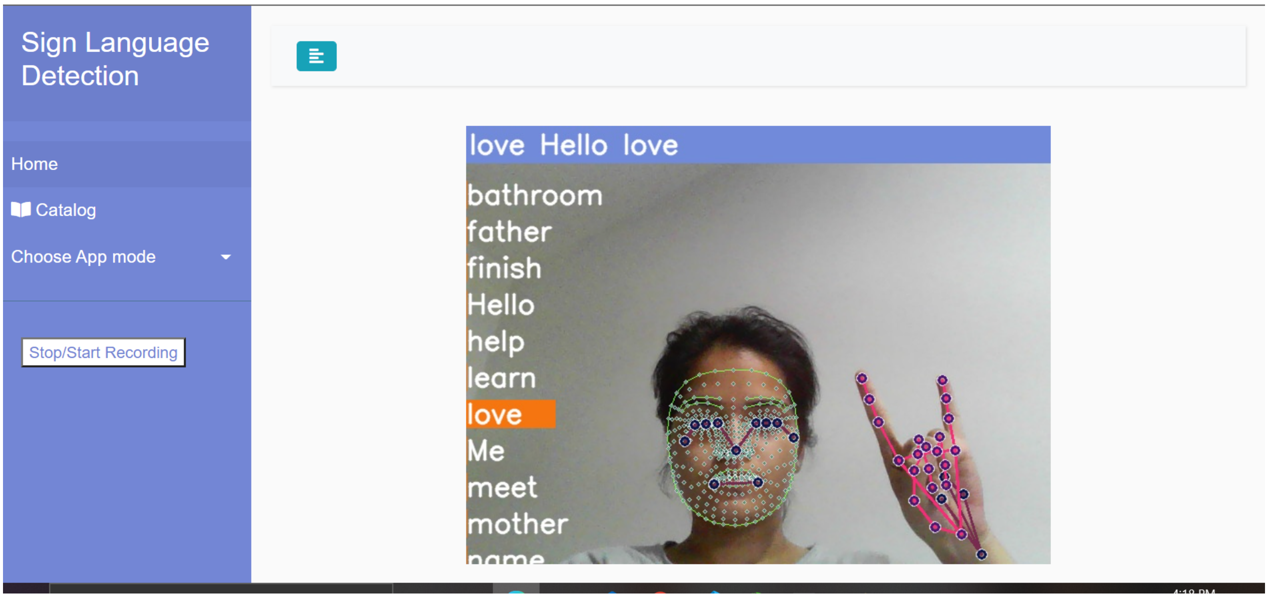

Figure 8.3: Sign Detection 2

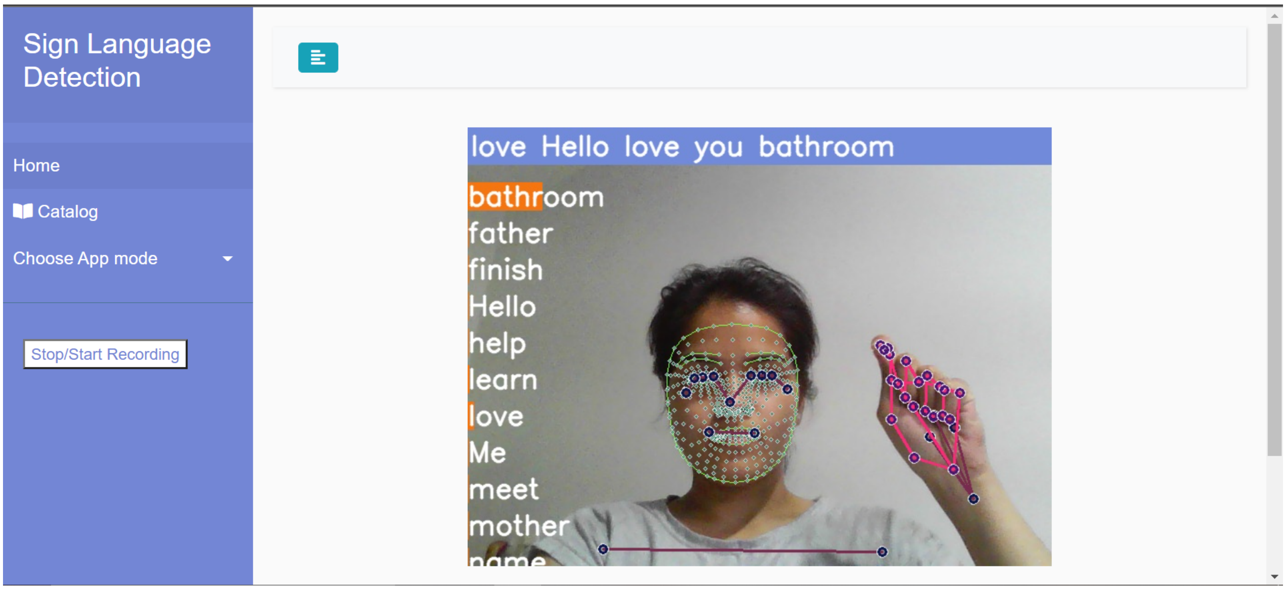

Figure 8.4: Sign Detection 3

Figure 8.5: Catalog

CHAPTER 9

CONCLUSION & FUTURE WORK

9.1 Conclusion

This paper presents a detailed mapping analysis that elucidates the implementation of Sign Language Action Recognition System and proposes a plan for developing a system that can detect not only image signs gestures but also sign actions. First, the current SLR flaws were determined and new Video frames recognition are discussed using the LSTM model. This report tries to put forward strong case for a high accuracy SLR sytem using fewer Dataset. The detailed system architecture is also provided.

9.2 Future Scope

The proposed sign language recognition system used to recognize sign language actions can be further extended to detect the bending of each finger using sensor glove, with flex sensors and an IMU to read the orientation of the hand. It can be used to collect data about the actions.This also increases readability. More training data can be added to detect the action with more accuracy. This project can further be extended to convert the signs to speech and sentences.

A system for animation of sign language for a person to converse with a deaf person could be added where the user types a message they want to convey and an animation doing sign language for the same communicates the message to the deaf person.

9.3 Applications

- Provide a way to convert sign language into text or voice for the hearing impaired so that they can communicate with others more efficiently and accurately.

APPENDIX

Appendix A: Mathematical Model

INPUT: - Let S is the Whole System Consist of
S- I, P, O

Where, I =input. I= U, Q

U = User U = {u1, u2,....... un}

Q = Query Q'{q1, q2.........qn}

P= Process P = CNN- Long Short Term Memory Model

CNN = Convolutional Neural Network

OUTPUT: The predicted result will be the output of the system

Prediction = model. Predict (test data)

Accuracy = (accuracy score(Y test,Y pred)*100)

Appendix C: Plagiarism Report

Curiginal

Document Information

Analyzed document	B_E_PROJECT_62_REPORT.pdf (D135214270)
Submitted	2022-05-02T10:41:00.0000000
Submitted by	Manish
Submitter email	mrjansari@pict.edu
Similarity	5%
Analysis address	mrjansari.pict@analysis.urkund.com

Sources included in the report

SA	**Pune Institute of Computer Technology / Final Project Report.pdf** Document Final Project Report.pdf (D122188713) Submitted by: anu_dondekar@yahoo.co.in Receiver: anu_dondekar.pict@analysis.urkund.com	1
SA	**Pune Institute of Computer Technology / GroupID29.pdf** Document GroupID29.pdf (D108558407) Submitted by: rakulkarni@pict.edu Receiver: rakulkarni.pict@analysis.urkund.com	4
SA	**Pune Institute of Computer Technology / BE_Report.pdf** Document BE_Report.pdf (D74264858) Submitted by: mstakalikar@pict.edu Receiver: mstakalikar.pict@analysis.urkund.com	2
SA	**Pune Institute of Computer Technology / BE_Report_53.pdf** Document BE_Report_53.pdf (D108485411) Submitted by: aachandorkar@pict.edu Receiver: aachandorkar.pict@analysis.urkund.com	4
SA	**Pune Institute of Computer Technology / BE_Project_Final.pdf** Document BE_Project_Final.pdf (D108705070) Submitted by: anu_dondekar@yahoo.co.in Receiver: anu_dondekar.pict@analysis.urkund.com	1
SA	**Lit survey on Real time sign lang Recognition System (1).doc** Document Lit survey on Real time sign lang Recognition System (1).doc (D125927103)	1

1/15

Figure 9.1: Plagiarism Report

REFERENCES

References

[1] Cheng K.L., Yang Z., Chen Q., Tai YW. (2020) Fully Convolutional Networks for Continuous Sign Language Recognition. In: Vedaldi A., Bischof H., Brox T., Frahm JM. (eds) Computer Vision – ECCV 2020. ECCV 2020. Lecture Notes in Computer Science, vol 12369. Springer, Cham.

[2] A. Kumar, K. Thankachan and M. M. Dominic, "Sign language recognition," 2016 3rd International Conference on Recent Advances in Information Technology (RAIT), 2016, pp. 422-428, doi: 10.1109/RAIT.2016.7507939.

[3] Anshul Mittal, Pradeep Kumar, Partha Roy, Raman Balasubramanian "A Modified LSTM Model for Continuous Sign Language Recognition Using Leap Motion" August 2019IEEE Sensors Journal 19(16):7056-7063. DOI:10.1109/JSEN.2019.2909837

[4] E. Abraham, A. Nayak and A. Iqbal, "Real-Time Translation of Indian Sign Language using LSTM," 2019 Global Conference for Advancement in Technology (GCAT), 2019, pp. 1-5, doi: 10.1109/GCAT47503.2019.8978343.