REAL TIME TRANSLATION TO REMOVE COMMUNICATION BARRIER

B. E. PROJECT

Submitted to Rashtrasant Tukdoji Maharaj Nagpur University, Nagpur
in Partial Fulfillment of the
Requirements for the Degree of BACHELOR OF ENGINEERING in
COMPUTER SCIENCE AND ENGINEERING.

By
VAIBHAVI BANABAKODE
(ID 17005005)
SHAMA ZAFAR
(ID 17005054)
GAURI TOSHNIWAL
(ID 17005062)

Guide
Dr. Latesh Malik
Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GOVERNMENT COLLEGE OF ENGINEERING NAGPUR 441108

2020-2021

2020-2021

REAL TIME TRANSLATION TO REMOVE COMMUNICATION BARRIER

B. E. PROJECT

Submitted to Rashtrasant Tukdoji Maharaj Nagpur University, Nagpur
in Partial Fulfillment of the
Requirements for the Degree of BACHELOR OF ENGINEERING in
COMPUTER SCIENCE AND ENGINEERING.

By
VAIBHAVI BANABAKODE
(ID 17005005)
SHAMA ZAFAR
(ID 17005054)
GAURI TOSHNIWAL
(ID 17005062)

Guide
Dr. Latesh Malik
Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GOVERNMENT COLLEGE OF ENGINEERING NAGPUR 441108

2020-2021

GOVERNMENT COLLEGE OF ENGINEERING NAGPUR 441108

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project/ dissertation entitled, "REAL TIME TRANSLATION TO REMOVE COMMUNICATION BARRIER", which is being submitted herewith for the award of B.E., is the result of the work completed by (1) Vaibhavi Banabakode, (2) Shama Zafar, (3) Gauri Toshniwal under supervision of Dr. Latesh Malik.

Dr. Latesh Malik Guide	Dr. Latesh Malik Head of Department	Dr. N. D. Ghawghawe Principal

DECLARATION

I hereby declare that the project entitled, "REAL TIME TRANSLATION TO REMOVE COMMUNICATION BARRIER" was carried out and written by me/ us under the guidance of Prof. DR. Latesh Malik, Head of Department, Department of Computer Science and Engineering, Govt. College of Engineering, Nagpur. This work has not previously formed the basis for the award of any degree or diploma or certificate nor has been submitted elsewhere for the award of any degree or diploma.

Date:
Place:

(1) Vaibhavi Banabakode
University Enrollment Number: 20181014217507

(2) Shama Zafar
University Enrollment Number: 20181014217506

(3) Gauri Toshniwal
University Enrollment Number: 20181014217494

ACKNOWLEDGEMENT

We feel immense pleasure and privilege in expressing deep sense of gratitude towards our guide Dr. Latesh Malik Head of Department, department of Computer Science and Engineering, Government College of Engineering Nagpur whose help and support, complete involvement, invaluable guidance and encouragement led us to complete this project work.

We express our sincere thanks to Dr. N.D. Ghawghawe Principal, Government College of Engineering Nagpur.

We are also thankful to our faculty members of the department of Computer Science and Engineering for their whole hearted support and kind cooperation from time to time.

We are especially thankful to Dr. Latesh Malik for providing us necessary facilities, their encouragement and valuable co-operation for completing our project work. We wish to express our gratitude towards our best effort and be a source of strength in the movement of despair and disappointment.

ABSTRACT

Communication becomes challenging due to factors like linguistic barriers, mode of communication, hearing and speech impairment. There is no one-stop solution to all these barriers of communication. Hence, there is a need for a solution that supports 1 . language translation through different input modes such as text, image and document. 2. Translating English text into Indian Sign Language and vice versa. The challenge also includes the unavailability of a publicly available dataset on Indian Sign Language and that Indian Sign Language is not standard across the country. In this work, we present two approaches to Sign Language recognition, dataset creation and real-time implementation. Also, we have used natural language processing to convert English text to Indian Sign Language. We aim to provide a one-stop solution by combining these modules with Language translation into a single digital interface (web application). Two datasets have been created: the first is a video dataset that involves 11 different words from Indian Sign Language with 59 videos for each word. The second dataset includes recording hand landmarks of 18 gestures from Indian Sign Language. The best performing model achieves 99% accuracy on the test dataset with reasonably good results in real-time.

CONTENT

Chapter 1

INTRODUCTION

In this chapter we will discuss the barriers of communication and ways to overcome the same, Introduction to Indian Sign Language and its types.

Communication is the way information is shared. It helps to understand people better by removing misunderstanding and creating clarity of thoughts and expression. Due to various reasons, communication can become challenging and these act as barriers to communication. Forms of communication barriers that prevent individuals from effective communication include language differences, difficulty in understanding unfamiliar accents, physical disabilities such as hearing problems or speech difficulties as well as the mode through which information is shared. These issues must be addressed, as existing systems majorly focus on translating one language into another but rarely do they focus on speech-impaired people, who need to communicate with the rest of the world using sign language. Also existing systems majorly focus only on a particular type of barrier making it cumbersome for the user to use a different system every time they face a different kind of barrier. Hence, there is a need for a one-stop solution to all kinds of barriers.

1.1: Overview

Being able to communicate is perhaps the most important life skill. Poor communication can stem from several issues and this has to be handled. Language and communication barriers can be challenging and there are a lot of types of hurdles they create.

Various Barriers to Communication:

- **Language Barriers:** Language and linguistic ability often act as an effective barrier to communication. Even if someone is communicating in the same language, the terminology or jargon associated with the language used in a message may act as a barrier if the receiver is not able to fully understand it.
- **Physiological Barriers:** Physiological barriers often result from the physical state of the receiver. Consider an example where a receiver who has reduced

hearing may not grasp the entirety of a spoken conversation delivered, especially if there is considerable background noise or some form of physical disabilities such as hearing problems.

- **Physical Barriers:** An important example of a physical barrier to communication could be the geographic distance between the receiver and sender. Over shorter distances, communication is often easier as more communication channels are available and less technology is required. Even though the advanced technology is serving in reducing the physical barriers, there is a need to understand the advantages and disadvantages of every communication medium so that the physical barriers can be removed using an appropriate channel.

- **Language Disabilities:** There are around 1.1 million deaf-and-dumb people in India. 98% of this population is illiterate, which makes sign language their only medium of conversation. Research shows that the trend of unfamiliarity with sign language is not expected to change anytime soon, as only 2% of deaf children attend school in India. Many people work with physical impediments to languages such as stuttering and hearing loss. These have no bearing on someone's ability to understand and do their job, but it can make communication more complicated and inefficient.

- **Mode of Communication**: A lot of times mode of communication could also make the process cumbersome like whether the information is in the form of text, image or speech/audio.

1.2: Sign Language

Sign language is a way of communication with the use of hands and other parts of the body. It is a visual language that uses a system of facial expressions, manual and body movements to communicate. Different sign languages are used in different countries as it is not a universal language. Linguists have identified at least 137 different sign languages.

Usually there are 4 ways to communicate in a sign language:

- **Fingerspelling**: Signer communicates letter by letter forming words.
- **Isolated**: Using signs specified for every word in the dictionary.
- **Continuous**: Uses a sequence of gestures that generate a meaningful sentence.
- **Non-manual features**: using tongue or mouth, facial expressions and body position.

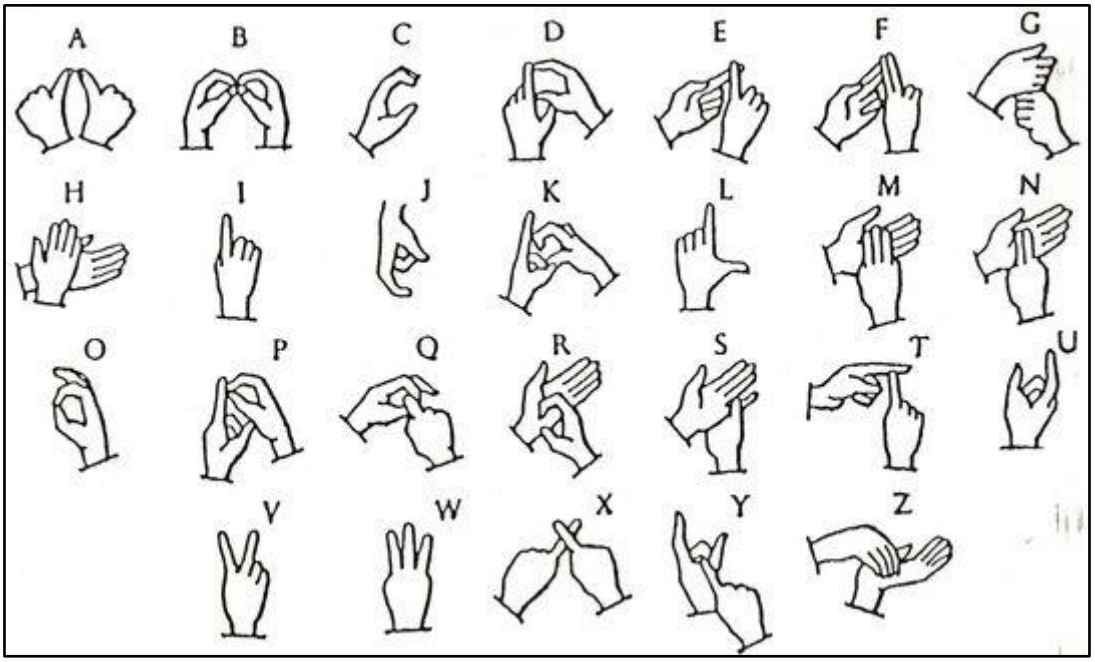

Fig 1.1 : Finger Spelling in Indian Sign Language

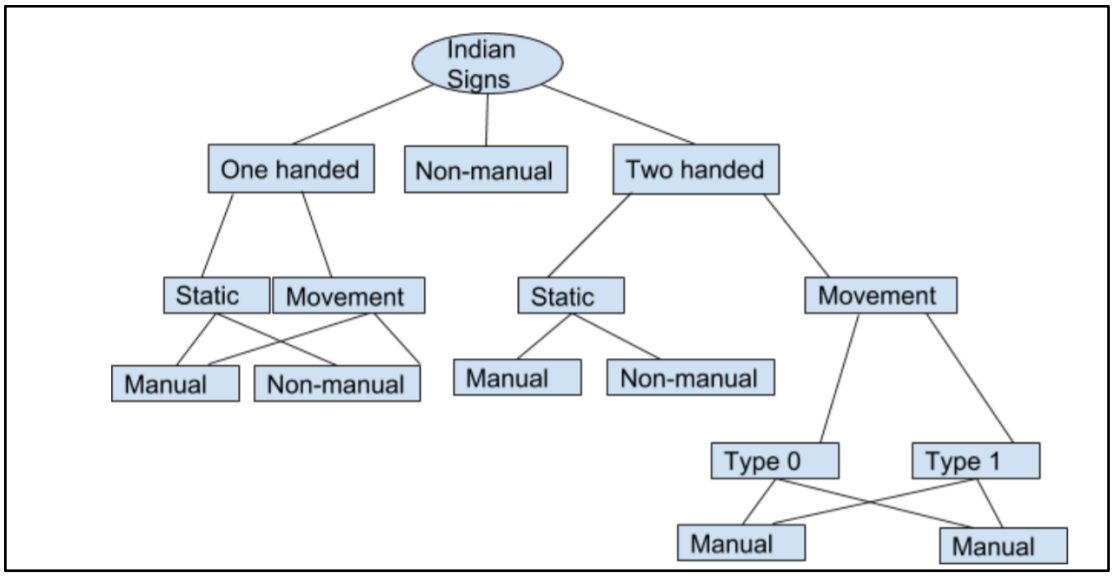

Fig 1.2: Indian Sign Language Type Hierarchy

The one handed signs are represented by static as well as dynamic movement of one hand. The static and dynamic movements are then classified into manual and non-manual signs. Two handed signs use both the hands for gesturing. It can be further classified into Type 0 and Type 1. Type 0 signs are those where the signer makes use of both the hands Type 1 signs are those where the use of one hand (dominant) is more compared to the other hand (non-dominant)

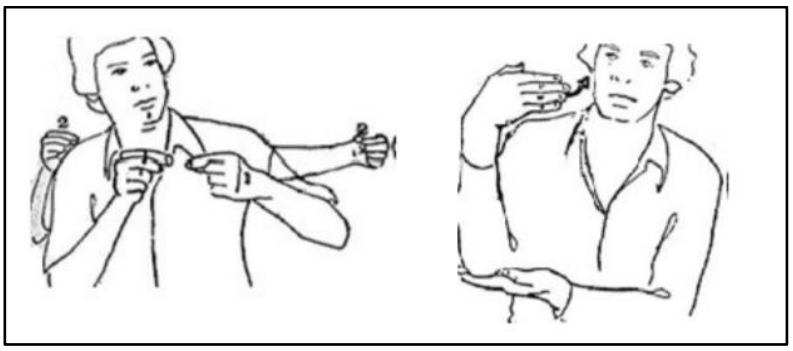

Fig 1.3: Two Handed Sign "long" (both the hands are moving) and "Flag" (only the dominant right hand is moving)

1.3: Problem With Existing System

Existing systems are majorly focusing only on a particular type of barrier making it cumbersome for the user to use a different system every time they face a different kind of barrier.

Also there are different issues coming to sign language translation. One popular implementation is using Gloves. The gloves don't have capability for written English input to glove movement output or the ability to hear language and then sign it to a deaf person, which means they do not provide reciprocal communication.

Some existing popular sign language translation application are listed below along with the their problems:

- **ProDeaf**

  ProDeaf (WebLibras) is a computer software that currently has a beta version in American Sign Language. The application cannot read sign language and turn it into word or text, so it only serves as a one-way communication.

- **SignAloud**

  SignAloud is a technology that incorporates a pair of gloves made by a group of students at University of Washington that transliterate American Sign Language (ASL) into English.

- **Kinect Sign Language Translator**

  Currently, this project is being worked on by researchers in the US to implement ASL translation. It's still a prototype, and the accuracy of translation is still not perfect.Also with this, Microsoft Kinect model 1414 is required.

These don’t focus on Indian linguistic communication. Indian Sign Language is a linguistically under-investigated language. Research and analysis on Indian Sign Language linguistics is also limited because of unavailability of standard sign dictionaries and the unavailability of such tools which provide any education for Indian Sign Language. In the interpretation between Sign language and verbal spoken language there is an intermediate step during which the Sign Language needs to be described by some written notation.

Many applications majorly focus on translating one language into another where users can type the text they want to translate, speak aloud, or take a photo of an image containing the text. Rarely do they focus on speech impaired people, Speech impaired people need to communicate with the rest of the world using sign language. There is a need for an elimination of the middle person who generally acts as a medium of translation. Hence there is a need for a platform which can bridge this barrier and implement machines as well as sign language translations.

1.4: Problem Statement

Formulating a technological solution to provide Real-time translation to remove:

1. Linguistic barriers and add ease of communication by providing various means of language translation such as text, image or document enabled input.
2. Sign language barriers using deep learning to translate sign language to text as well as NLP to translate text to sign language.

Hence, there is a need for a method or an application that can recognize sign language gestures so that communication is possible even if someone does not understand sign language. With this work, we intend to take a basic step in bridging this communication gap using Sign Language Translation.

Chapter 2

LITERATURE REVIEW

2.1: Introduction

In this section, the recent work in the area of automatic recognition of sign language is discussed. There are varied techniques available which can be used for recognition of sign language. Sign language recognition consists mainly of three steps: preprocessing, feature extraction and classification.

2.2: Previous Work

- Earlier work on Double hand sign recognition, Agrawal et al. [1] proposed a two stage recognition approach for 23 alphabets. Signs are performed by wearing red gloves on both hands for better segmentation. The segmented frames served as an input to feature extraction and recognition phase. In stage-1 the hand features are extracted which describes the overall shape of gestures, recognition is done through training feature vectors without using any classifier. In stage-2, a recognition criterion was tough and the feature vector had binary coefficient. Finally, an output shows whether the gesture is correct or not.

- In [2], a method for the recognition of 10 two handed Bangla characters using normalized cross correlation is proposed by Deb et al. A RGB color model is adopted to select heuristically threshold values for detecting hand regions and template based matching is used for recognition. However, this method does not use any classifier and is tested on limited samples.

- M.A. Mohandes [3] proposed a method for the recognition of the two handed Arabic signs using the Cyber Glove and support vector machine. Principal Component Analysis (PCA) feature is used for feature extraction. This method consists of 20 samples of 100 signs by one signer. 15 samples of each sign were used for training a Support Vector Machine to perform the recognition. The system was tested on the remaining 5 samples of each sign. A recognition rate of 99.6% on the testing data was obtained. When the number of signs in

the vocabulary increases, the support vector machine algorithm must be parallelized so that signs are recognized in real time. The drawback of this method was to employ 75% images for training and remaining 25% for testing.

- Work on two handed signs has been done in Rekha et al. [4]. Here, Principle Curvature Based Region (PCBR) is used as a shape detector, Wavelet Packet Decomposition (WPD-2) is used to find texture and complexity defects algorithms are used for finding features of the finger. The skin color model used here is YCbCr foregmenting hand region. The classifier used is Multi class non-linear support vector machines (SVM). The accuracy for static signs is 91.3%. However, three dynamic gestures are also considered which use Dynamic Time Warping (DTW). The feature extracted is the hand motion trajectory forming the feature vector. The accuracy for the same is 86.3%.

- In [5], threshold models have been designed to differentiate between signs and non-sign patterns of American Sign Language in Conditional Random Field (CRF). The recognition accuracy for this system is 93.5%.

- Aran et al. [6] have developed a system called the sign tutor for the automatic sign language recognition. The three stages in this system are: face and hand detector, analysis and classification. Users are made to wear colored gloves in order to easily detect the hand and remove occlusion problems. For both the hands,kalman filters are used for smoothening hand trajectories and thereafter, features for finding the hand shape are extracted. The classifier used is Hidden Markov Model (HMM). The dataset consisted of 19 signs from ASL. The accuracy for a signer-dependent system is 94.2%, while for a signer-independent system is 79.61%.

- Lekhashri and Pratap [7] developed a system for both static as well as dynamicISL gesture recognition. The Various features extracted are skin tone areas, temporal tracing and spatial filter velocimetry. This obtains the motion print of the image sequence. Then pattern matching is used to match the obtained motion prints with the training set which contains the motion print of

the trained image sequences. Then, the closest match is produced as the output.

- Nandy et al. [8] proposed an approach using direction histogram feature. The classification is done through Euclidean Distance and with also K-nearest neighbors and both are compared. In the dataset, only isolated hand gestures of 22 ISL signs are taken. The recognition rate is found to be 90%. The limitation of this approach is the poor performance in case of similar gestures.

- Sandjaja and Macros [9] also used color -coded gloves for tracking human hands easily. Multi-color tracking algorithm is used to extract the features. The
  recognition is done through the Hidden Markov Model. The dataset consisted of Filipno sign language numbers. The recognition rate is 85.52%.

- Quan [10] proposed extraction of multi-features for the image, where hand is the only object. These features are color histogram, 7 Hu moments, Gabor wavelet, Fourier descriptor, and SIFT features and used support vector machine for Classification.

- Bauer and Hienz [11] introduced the basic problems and difficulties that arise while performing continuous sign language recognition. In this paper, manual sign parameter information such as hand shape, hand location and orientation are extracted, forming the feature vector. The hand motion is not considered as it is handled separately by HMM topology, and hence not included in the feature vector. Gloves of different colors are worn by users to distinguish dominant and nondominant hand.For each sign,one HMM is modeled. The system was tested on 52 different signs of German Sign language (GSL) and 94% accuracy found for all features. For 97 signs, this accuracy drops to 91.7%.

- Alon et al. [12] proposed a unified framework for simultaneously performing temporal segmentation, spatial segmentation and recognition.There are three major contributions of this paper: Firstly, for the detection of the hand,

multiple candidates are detected in every frame and the hand is selected through a spatiotemporal matching algorithm. Secondly, for easy and reliable rejection of incorrect matches, a pruning framework is used which is based on classification. Thirdly, a sub-gesture reasoning algorithm is used that finds those models of gestures that falsely match to some parts of other large gestures. Skin color combined with motion cues is used for hand detection and segmentation.

- Shanableh et al. [13] presented a technique for Arabic Sign Language Recognition. This method works online as well as offline for the isolated gestures.he technique used uses varied spatio-temporal features. The features used for temporal contexts are forward, backward, and bidirectional predictions. After motion representation, spatial-domain features are extracted. For classification purposes, Bayesian classifiers and k-nearest neighbors are used. The accuracy of this system varies from 97% to 100%.

2.3: Steps in Sign Language Recognition

Sign language recognition comprises of three major steps: hand detection and segmentation, feature extraction and classification.

2.3.1: Hand Detection and Segmentation

Hand segmentation is the process of extracting handmeans pixels representing the hand are localized in the image and segmented from the background before recognition. In segmentation procedure, a number of restrictions are imposed on background, user and imaging [14]. Restrictions on the background and in imaging are commonly used. A controlled background greatly simplifies the task. It can vary from a simple light background [15] [16] to a dark background [17] [18]. Mostly a uniform background is used. In the case of restriction on users, the user can wear long sleeves [19]. The chosen color greatly helps the segmentation task. Skin color segmentation is another approach that can be used to detect the hand but the drawback of this method is that also finds faces, so we have to exclude the face from the image. Skin color can be modeled using simple histogram matching [20], mixtures of Gaussians [21]. The spaces used can be RGB (red, green and blue components) [22],

normalized RGB [23], YUV space [24], HSI (hue, saturation and intensity model) [25][26].

2.3.2: Features for Gesture Recognition

Feature extraction is the most important module in sign language recognition systems. Since the nature of every sign language and signs considered is different, the reliable features need to be selected. Feature extraction is aimed at finding the appropriate and most distinguishing features of the object. Sign language recognition can be accomplished using manual signs (MS) and non-manual signs (NMS). Manual signals includes features such as hand shape, hand position and hand motion whereas non manual signals include facial features, head and body motion. A lot of previous work has been done by extracting appearance based features. This is because these features are simple and have low computational time, and therefore can be used for real time applications.The feature descriptors can be classified as edge, corner, blob or region based descriptors. The various shape based descriptors can be contour based or region based. The region based shape descriptors are further classified as local or global descriptors. These features include region based descriptors (image moments, image eigenvectors, Zernike moments [27], Hu invariants [28], or grid descriptors) and edge based descriptors (contour representations [29], Fourier descriptors [30]). There are colour, motion and texture based descriptors also.

2.3.3: Classifier

Once features are computed, recognition of signs will be performed. Sign recognition will be decomposed into two main tasks: the popularity of isolated signs and the recognition of dynamic signs. Many recognition techniques include Support Vector Machine (SVM), template matching, neural networks, geometric feature classification, or other standard pattern recognition techniques. For dynamic sign recognition, the temporal context has to be considered. It's a sequence processing problem which will be accomplished by using Finite State Machines (FSM), Dynamic Time Warping (DTW), and Hidden Markov Models (HMM) to cite some techniques.

- In [31], a user independent framework for recognition of isolated Arabic sign language gestures has been proposed. For this, the user is required to wear gloves for the simplification of hand detection and segmentation. K-Nearest

Neighbor and polynomial networks are the two classifiers used in this paper and then these two classifiers' recognition rate are compared. Certain special devices like cyber gloves, or sensors can also be utilized for sign language recognition. These devices find the accurate position and motion of the hand. Though more accurate, these devices are cumbersome and prevent the natural interaction of the signer with the computer.]

- In [32], Back propagation and Kohonen's self-organizing network has been applied to recognize gestures related to American Sign Language (ASL) for 14 sign vocabulary. The overall accuracy of the system is 86% by using back propagation and reduces to 84% when Kohonen's network has been applied. However, the low recognition is due to insufficient training data, lack of abduction sensors or over constraining the network.

- In [33], Euclidian space and neural networks have been used for recognizing the hand gestures. They have defined some specific gestures and made a test on that. The achieved accuracy is 89%. Since different users perform the signs in different manners, the number of false positives is increased and the recognition rate is also low; the other factor being the occlusion of fingers.

- In [34], the authors present a hierarchical structure based on decision trees in order to be able to expand the vocabulary. The aim of this hierarchical structure is to decrease the number of models to be searched, which will enable the expansion of the vocabulary since the computational complexity is relatively low. They used a sensored glove and a magnetic tracker to capture the signs and achieved 83% recognition accuracy, at less than half a second average recognition time per sign, in a vocabulary of 5113 signs. One of the biggest challenges in sign language recognition arises in the case of continuous sign sentences, which means that a sign is succeeded and preceded by certain other signs, therefore forming a sentence of these signs. This is similar to the coarticulation problem in speech. The transition from the end of one sign to the start of another sign needs to be identified in order to find the isolated signs within the continuous signing. Such movement is known as movement epenthesis.

- In [35], a methodology based on Transition-Movement Models (TMMs) for large-vocabulary continuous sign language recognition is proposed. TMMs are used to handle the transitions between two adjacent signs in continuous signing. The transitions are dynamically clustered and segmented; then these extracted parts are used to train the TMMs. The continuous signing is modeled with a sign model followed by a TMM. The recognition is based on a Viterbi search, with a language model, trained sign models and TMM. The large vocabulary sign data of 5113 signs is collected with a sensored glove and a magnetic tracker with 3000 test samples from 750 different sentences. Their system has an average accuracy of 91.9%.

- Agrawal et al. [36] have proposed a user dependent framework for Indian Sign Language Recognition using redundancy removal from the input video frames. The skin color segmentation and face elimination is performed to segment the hand. Various hand shape, motion and orientation features are used to form a feature vector. Finally a MSVM is used to classify the signs with 95.9% accuracy.

2.4: Summary

A number of work has been accomplished within the case of static isolated symptoms of sign language recognition. Many researchers have used various strategies for the Same. Some result in high recognition accuracy on the high cost of computational complexity. At the same time some systems were simpler but much less accurate. Distinctive datasets referring to various different corners of the world are created which increase complexity and constraints. Numerous methods were proposed to resolve the 3 fundamental issues of vision-based totally gestural interfaces, particularly hand segmentation and detection, tracking and recognition. This chapter discusses all the methods and techniques available in sign recognition systems.

Chapter 3

WORK DONE

3.1: Dataset Creation and collection

This section deals with the dataset created or arranged for Indian Sign Language to English text module as well as English text to Indian Sign Language.

3.1.1 Dataset used for Text to Indian Sign Language

Indiansignlanguage.org: Indian Sign Language Portal by Ramakrishna Mission Vivekananda Educational and Research Institute (RKMVERI), Coimbatore, Tamil Nadu.

Description:

The dataset offers a huge collection of Indian Sign Language (ISL) signs. Each sign has an image, running video and threaded discussions. It is an ideal resource to use while you learn/teach Indian Sign Language. They are continually adding more signs and designing new services to empower the Deaf. **The Faculty of Disability Management and Special Education** provides quality education to all students with disabilities. Subjects will be presented through Sign Language by the subject teachers who are trained in Sign language.

Fig 3.1: Sample of indiansignlanguage.org dataset

3.1.2 Dataset Creation for Sign to Text: Approach 1

No standard dataset is available for Indian Sign Language with distinctly coloured hands. Therefore, a new video dataset is created consisting of 11 different categories of dynamic isolated signs of ISL with different light conditions and distinctly coloured hands which do not match any other object or the background in the frame. This was done to simplify the problem of hand segmentation within a frame.

Number of classes - 11
Number of train videos in each class - 50
Number of test videos in each class - 9

The 11 classes consist of 3 double hand signs and 8 single hand signs. Double hand sign classes include Perfect, Thank You and Strong while single hand involve See, Good, Morning, You, Mango, Eat, Tall and Sorry.

Fig 3.2: Samples from created video dataset

3.1.3 Dataset creation for Sign to Text: Approach 2

This approach involved the use of integrated Webcam to record 18 different dynamic ISL Signs and the corresponding 3D hand landmarks into train CSV files.
Final Dataset included two separate CSV files.

1. Single Handed (11 classes) - [rows(15,500), columns(85)]
2. Double Handed (7 classes) - [rows(10,500), columns(169)]

In both CSV each class has 1500 records. In Single Handed CSV 11 classes include signs such as See, Hello, Mango, Eat, Morning, Tall, Good, Sorry, What, This and You. In Double Handed CSV 7 classes include signs like Perfect, Thank You, Dance, Sleep, Time, Late and Work.

3.2: Technical Design Theory

TransApp is a Digital Interface to facilitate real-time Translation.

System of TransApp can be broadly divided into three parts:

1. **Digital Interface** (Web Application)
2. **Web Server**
3. **Backend Modules**
    a. Rest APIs
    b. Sign Language Detection Model
    c. English Text to Indian Sign Language Conversion Module
    d. Translation Helper Functions (Language text Input, Image Input, Document Input)

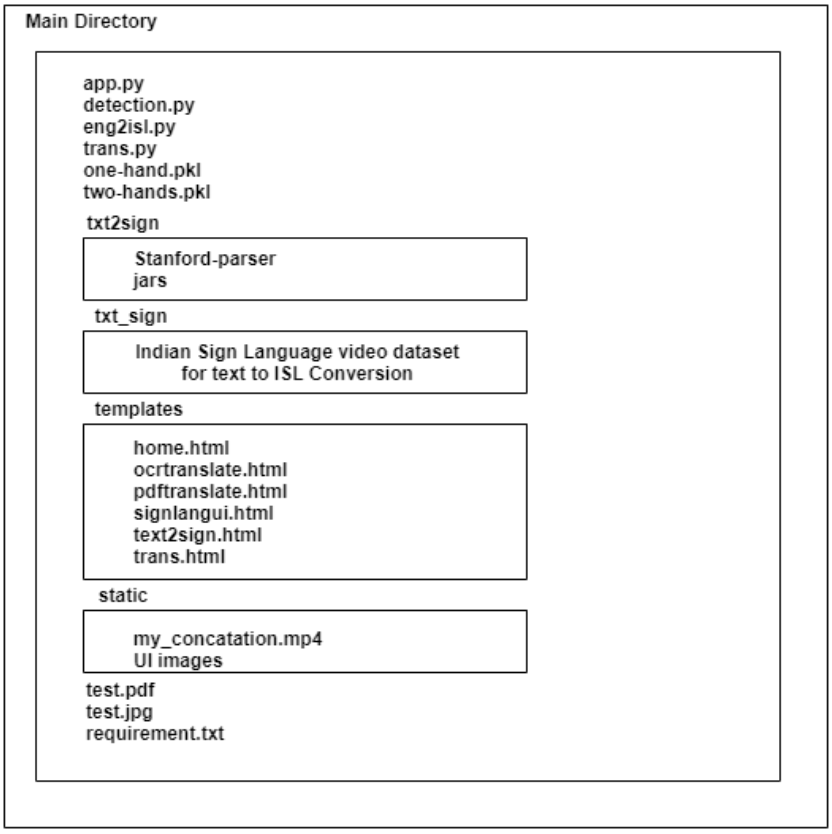

Fig 3.3: Project Directory

Digital Interface of TransApp is a Web Application through which real-time translation can take place. The interface requests data to the REST API developed in flask[40], a framework of python programming language with the help of Flask-RESTful which is an extension for Flask that adds support for quickly building REST APIs.

Interface - Server Communication

Communication between Interface and REST API follows the request-response model over the HTTP protocol.

API has following routes -

- /home
- /translate
- /pdftranslate
- /ocrtranslate
- /t2s
- /s2t
- /video_feed

- **Home Page:**

  The endpoint **"/home"** renders the landing page of the Webapp home.html, here User can choose different options like Language Translation, Document Translation, Image Translation, Text to ISL, ISL to text and the page will redirect the user to the the corresponding pages.

- **Language Translation:**

  This endpoint receives POST requests using the form data.

  Form data:

  {

  "message":msg,

  "languages":language

  }

  This activity asks for input text and destination language from the user. When a user clicks on a translate button, both input text and selected language are fetched by the server. Interface POSTs request for translation to the REST API on the route "**/translate**" after which it detects the original language of the

text. Next, it converts the text into the required destination language using googletrans which is a free and unlimited python library that implements Google Translate API. This uses the Google Translate Ajax API to make calls to such methods as detect and translate after which a response is sent that can be viewed on the updated trans.html page.

- **Image Translation:**

  This endpoint receives POST requests using the form data.

  Form data:

  {

  “file”:file,

  “languages”:language

  }

  This activity asks for Image and destination Language from the user. When a user clicks on a translate button, both image and selected language send in the form of form data. Interface POSTs request for translation to REST API on the route “**/ocrtranslate**”. At the server end, Language and Image is fetched from the form data the image is in the form fileobject hence werkzeug package is used to convert fileobject to appropriate filetype i.e. jpg/png and extracts the text from the given image using pytesseract package and then detects the original language of the text. Next, it converts the text into required destination language using googletrans and a response is sent which can be viewed on the updated ocrtranslate.html page.

- **Document Translation:**

  This endpoint receives POST requests using the form data.

  Form data:

  {

  “file”:file,

  “languages”:language

  }

  This activity asks for Document and destination Language from the user. When a user clicks on a translate button, both document and selected language are sent in the form of form data. Interface POSTs request for translation to

REST API on the route "**/pdftranslate**". At the server end, Language and Document is fetched from the form data the document is in the form fileobject hence werkzeug package is used to convert fileobject to appropriate filetype i.e. doc/pdf and extracts the text from the given image using textract and then detects the original language of the text. Next, it converts the text into required destination language using googletrans and a response is sent which can be viewed on the updated pdftranslate.html page.

- **Text to sign:**

  This endpoint receives POST requests using the form data.

  Form data:

  {

  "message":message

  }

  This activity requires the user to input a message to be converted to Indian Sign Language. When a user clicks on a translate button, the message is sent in the form of form data. Interface POSTs request for translation to REST API on the route "**/t2s**". At the server end, input messages are fetched from the form data. This is then passed to the Text to Sign Translation module which returns a corresponding Indian Sign Language Video created by the Module, this video response is sent which can be viewed on the updated text2sign.html page.

- **Sign to text:**

  The **'/s2t"** renders the signlangui.html page which hits the **"/video_feed" endpoint** to get the live feed input from the camera and pass it to the Sign Language Detection Module in form of frames, these frames are then processed by the module to get the predicted class and accuracy printed on the frames and the modified frames are returned to the "/video_feed" endpoint which then processes the frames as stream of bytes to livestream the output on signlangui.html page.

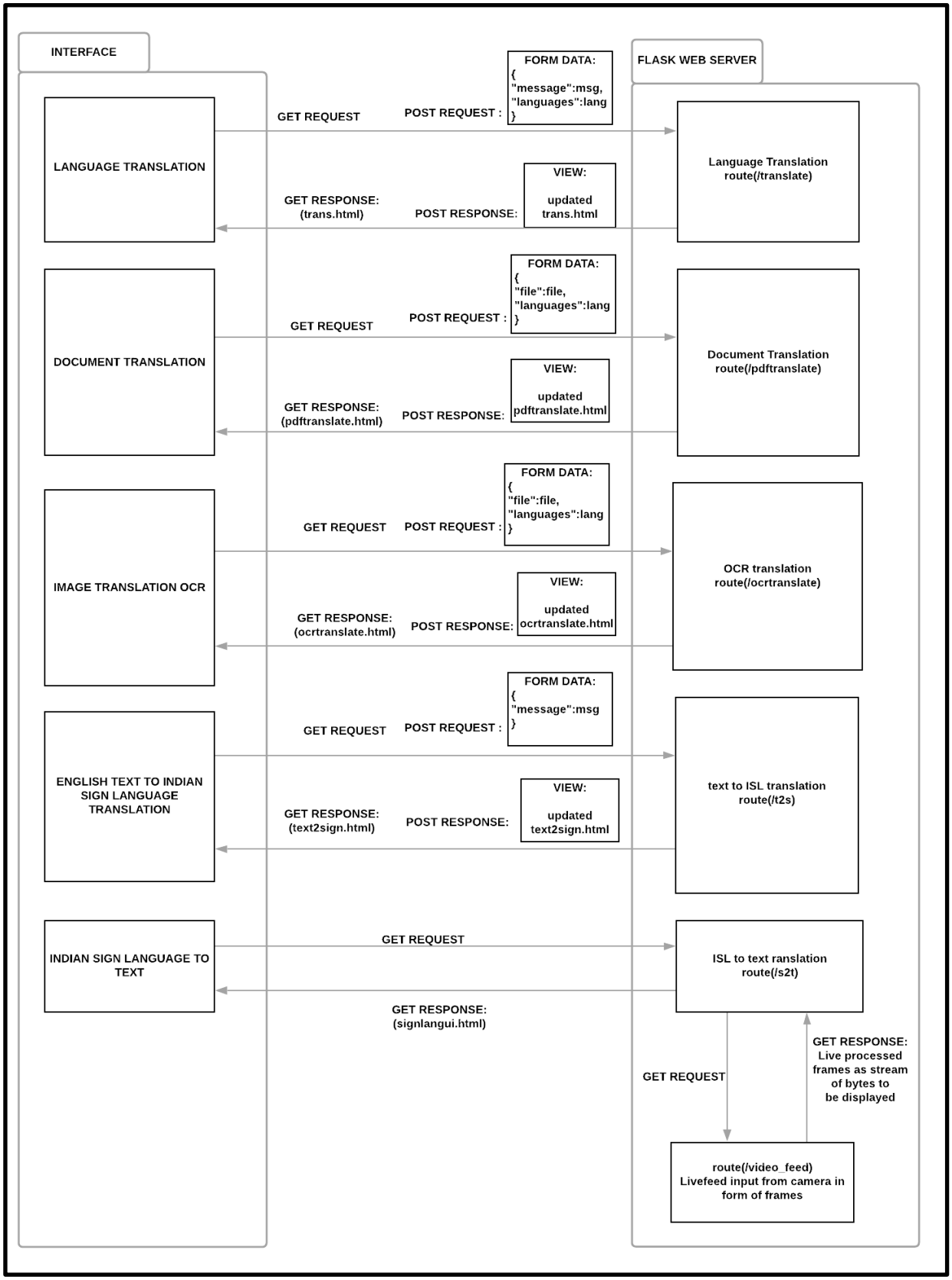

Fig 3.4: UML diagram showing Interface-server connection

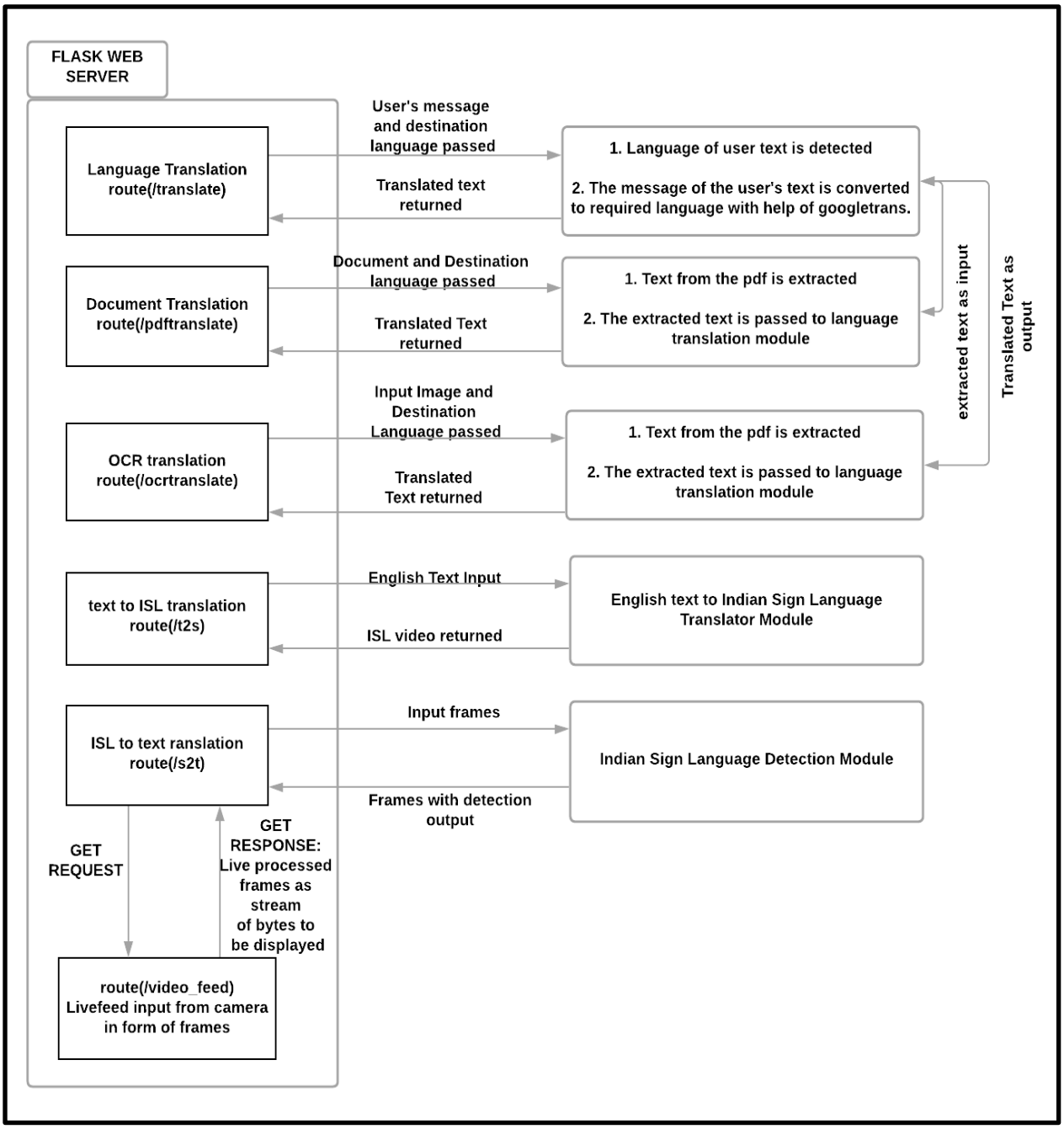

Fig 3.5: UMLdiagram showing server-backend connection

3.3: English text to Indian Sign Language

The objective of this module is to convert English text to ISL using Natural Language Processing (NLP) such that the output is structured according to ISL grammar rules. This approach makes use of ISL gesture videos for specific words to translate into ISL.

3.3.1: Existing Approach

So far the work done in this field is majorly implemented on American Sign Language or British Sign Language, but not much development has been done on ISL. The underlying architecture for most of systems for English text to ISL are based on:

1) Direct translation: Direct dictionary based mapping approach is used to convert english words to corresponding ISL gestures.
2) Transfer based architecture: Grammar rules specific to sign language are being applied so as to define a proper translation from one language system to another. Although this approach so far does not have a proper translation for ISL grammar and has a lot of inconsistencies.

3.3.2: Indian Sign Language Grammar

Indian Sign language has its own grammar and is much different from the manual representation of spoken English or Hindi. Following are the distinct features that it has:

1) Number representation is done with hand gestures for each hand.

2) Signs for family relationships are preceded by male or female.

3) In interrogative sentences, all the WH questions are placed in the back of the sentence. E.g. “You go where?” is used instead of “Where are you going?”.

4) It also consists of many non-manual gestures such as mouth pattern, mouth gestures, body postures, head position and eye gaze.

5) The past, present and future tense is represented by signs for before, then and after.

7) Always sign in the Present Tense.

English Grammar	Indian Sign Language Grammar
English grammar is well structured and a lot of research work has been carried out to define the rules for it. English grammar follows the subject-verb-object order.	ISL is invented by deaf and a little work has been done to study the grammar of this language. The structure of sentences of ISL follows the subject-object-verb order[13].
English language uses various forms of verbs and adjectives depending upon the type of the sentence. Also, a lot of inflections of the words are used in English sentences.	ISL does not use any inflections (gerund, suffixes, or other forms), it uses the root form of the word.
English language has much larger dictionary	Indian sign language has a very limited dictionary, approximately 1800 words[10].
Question word in interrogative sentences is at the start in English	In Indian sign language, the question word is always sentence final
A lot of helping verbs, articles, and conjunctions are used in the sentences of English	In Indian sign language, no conjunctions, articles or linking verbs are used

Table 3.1: Comparison of Grammar of English and ISL

3.3.3: Algorithm Design

The system consists of following 5 modules:

1) English parser to parse the English text.

2) Sentences reordering module based on ISL grammar rules as described above.

3) Eliminator for eliminating stop words such as “a”, “an”, “the” etc.

4) Stemming to get the root words of each word.

5) Video conversion module.

The input to the system is a written English text which is parsed using the stanford parser to create a phrase structure based on its grammar representation. After parsing, reordering is done to meet ISL grammar needs since, English text follows Subject-Verb-Object structure whereas ISL follows Subject-Object-Verb structure along with variation of negative and interrogative sentences. After this unwanted words are removed using a list of stopwords provided, as ISL has only those words which have meanings and all helping words like linking verbs, articles etc are not used and hence removed. The output of which is sent to the lemmatization module which reduces each of the words to its root form. The gesture words not present in the dictionary are given out alphabetically as output.

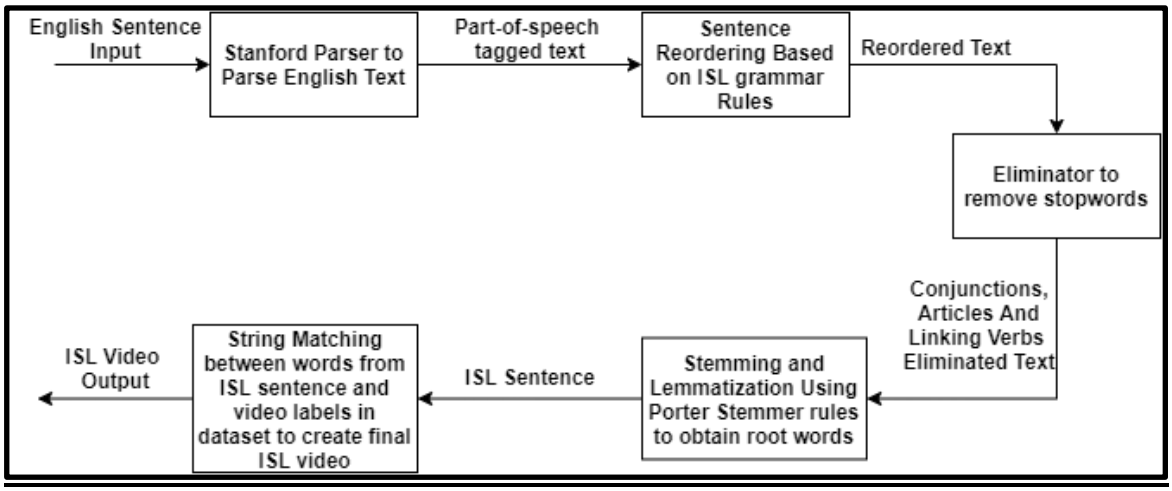

Fig 3.6: Workflow of text to Indian Sign Language Module

- **Parsing of the Input English Text:**

  To carry out rule based conversion of one language to another, grammatical structure of both the source and target language must be known. Parsing is the answer to acquiring this grammatical structure. Stanford parser is capable of producing three different outputs, part-of-speech tagged text, context free grammar representation of phrase structure and type dependency representation. The parser uses Penn tree tags for parsing the English sentence.

- **Elimination of Stop Words:**

  Since Indian Sign Language deals with words associated with some meaning, unwanted words are removed these include various parts of speech such as TO, POS(possessive ending), MD(Modals), FW(Foreign word), CC(coordinating conjunction), some DT(determiners like a, an, the), JJR, JJS(adjectives, comparative and superlative), NNS, NNPS(nouns plural, proper plural), RP(particles), SYM(symbols), Interjections, non-root verbs.

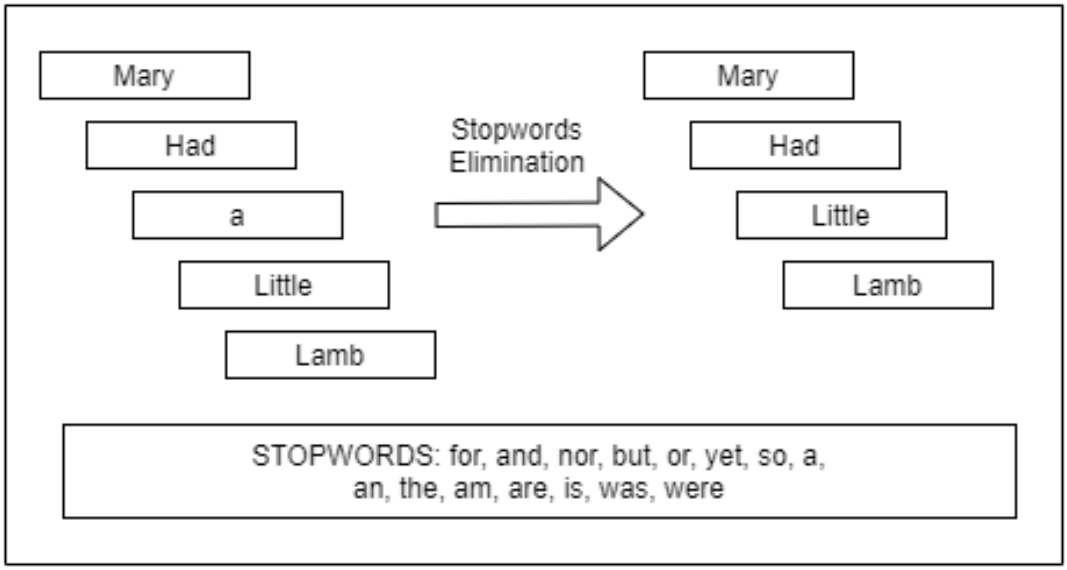

Fig 3.7: Removal of Stop words

- **<u>Lemmatization and Synonym replacement:</u>**

  Indian sign language uses root words in their sentences. So we convert them to root form using Porter Stemmer rules. Along with this, each word is checked in a bilingual dictionary; if a word does not exist, it is tagged to its synonym containing the same part of speech.

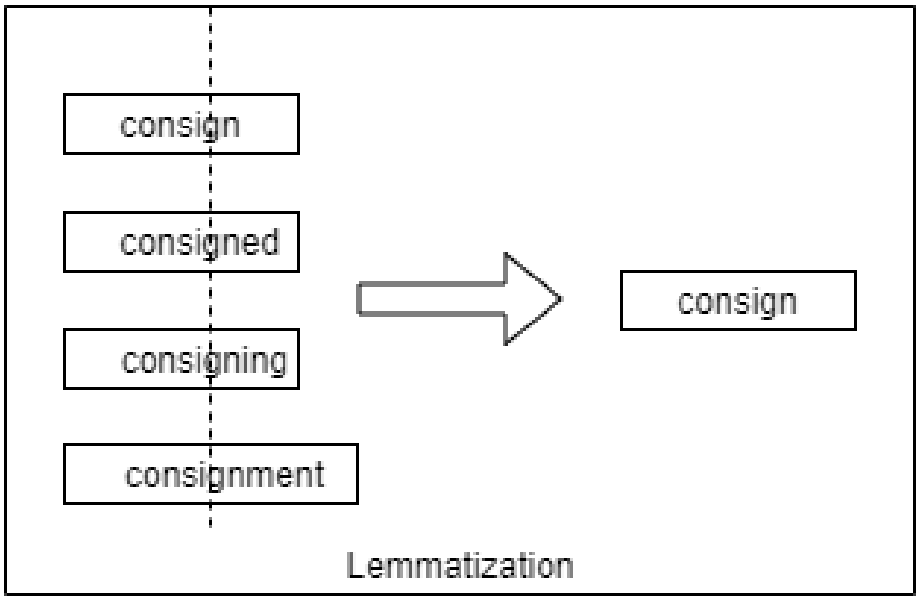

Fig 3.8: Lemmatization example

- **Video Conversion Stage:**

  After the completion of above stages we get the ISL transformed text , the program will then find matches from the dataset available for each of the words. This will be based on the basic string matching algorithm between the processed input text and labels of videos present in the labelled dataset.If a label is not found then the particular word will be broken into its corresponding label and the letter signs will be added to the output video. Finally a display of a set of videos as a sequence one after the other can be seen on the screen.

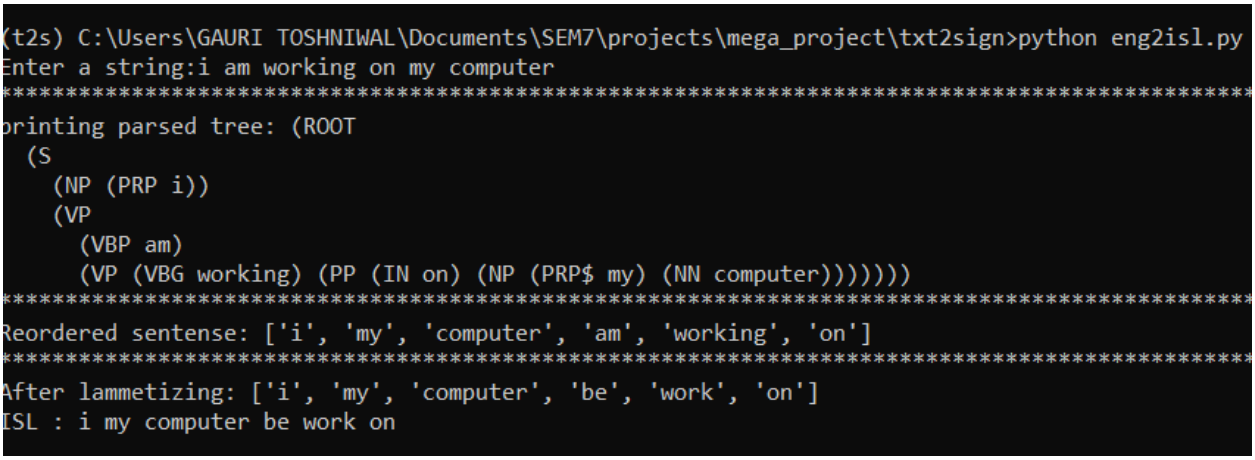

Fig 3.9: Backend showing complete process of conversion from english to Indian Sign Language sentence

Some examples of English sentences and corresponding Indian Sign Language sentences

Sr. No.	English Sentence	ISL sentence
1	I am working on my computer	I my computer be work on
2	It is raining outside	It be rain outside
3	What is your name ?	Your name what be
4	He is going to the university	He university be go to

Table 3.2: English sentences and corresponding Indian Sign Language sentences

3.4: Indian Sign Language to English Text (Approach 1)

The objective of this module is to convert ISL gestures (isolated) to english text using Convolutional Neural Network (CNN) to work on spatial features and Recurrent Neural Network (RNN) for the temporal features of gestures.

In this approach, we have made an attempt to recognize ISL gestures on isolated signs with the use of a vision-based method. A dataset has been created separately with significant video samples and larger gesture variants so as to avoid ambiguity. This is done so that the final CNN model can generalize better.

3.4.1: Convolutional Neural Network (CNN)

The invention of neural networks as a machine learning technique is very much inspired from how our brain works and is structured. Just like a brain functions, a network has learning units called neurons. These neurons learn to produce output signals (e.g. the label "cat") from input signals (e.g. picture of a cat) forming the basis of how automated recognition works.

A convolutional neural network (CNN, or ConvNet) is an artificial neural network which follows a feedforward technique in which connection patterns between the neurons are inspired from the way the animal visual cortex is organized.

A general CNN model has a set of repetitive blocks of neurons which are applied across time (for audio signals) or space (for images). These blocks of neurons are recognized as 2-dimensional convolutional networks for images, which are repeatedly applied over each section of image. In case of speech input, they can be interpreted as 1-dimensional convolutional kernels applied across time windows.

When a CNN is going through the process of training, the weights of these repeated blocks are shared, i.e. the gradients of the weights learned over various sections of image are averaged.

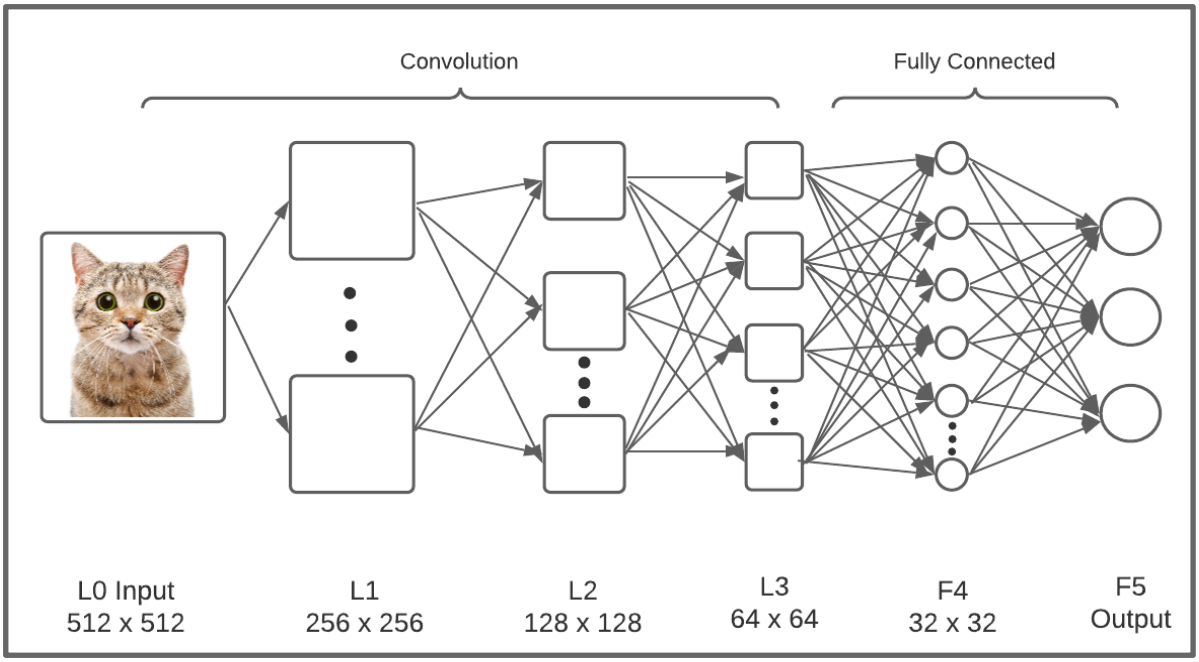

Fig 3.10: Convolutional Neural Network

- **Convolution**

  Convolution filters are the first layer that receives an input signal. It is a process where the complete network takes in whatever it has learned in the past and tries to label the input signal using that knowledge. If the network has observed that the input signal looks like cat images that it has seen before, the "cat" reference signal will be mixed into or convoluted with the input signal. The output produced during the course will be passed onto the next layers in the network.

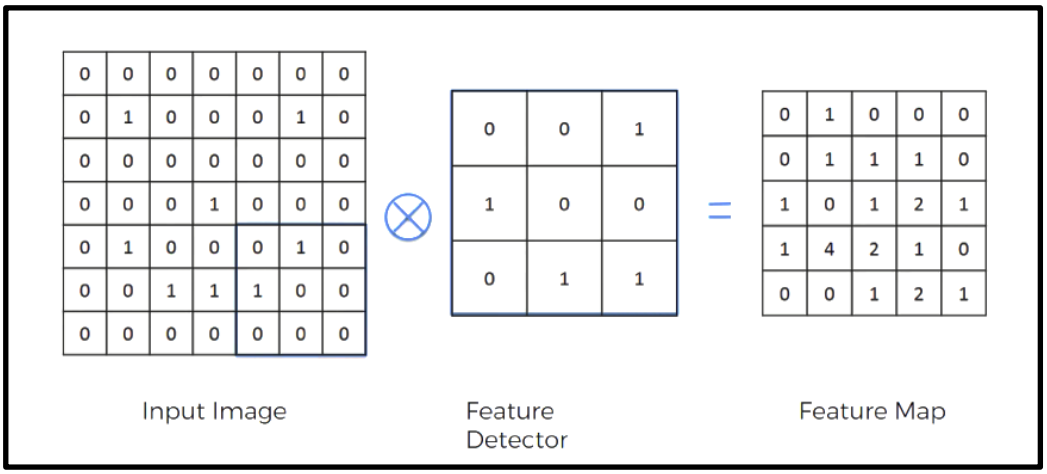

Fig 3.11: Convolution operation applied on input image

The way convolution operation is carried out, it has a tendency of being transnational invariant. This means that each convolution filter is representing an FOI (feature of interest). E.g whiskers, fur, nose etc. and the CNN combines these features to learn which of these comprise the resulting reference i.e. cat. The output signal strength of a neuron is not dependent on the position of these features but simply whether such features are present or not.

For example, suppose we want to perform convolution operation on a 64x64x3 (64x64 image with 3 channels R,G and B respectively) with a 5x5x3 filter. This 5*5*3 filters would be taken and will have to slide over the complete image. Once the filter is put over one section of the image of cat, take the dot product between the filter and the section of the input image.

- **Subsampling**

  Another feature of using CNN is the use of a process called subsampling. We can reduce the sensitivity of the filters to noise and variations of the input image from the convolution layer and smoothen it. This can be achieved by taking the averages over a sample of the signal or by taking the maximum.

  Examples of subsampling methods (for image signals) include the reduction of the size of the original image, or reduction of color contrast across red, green, blue (RGB) channels.

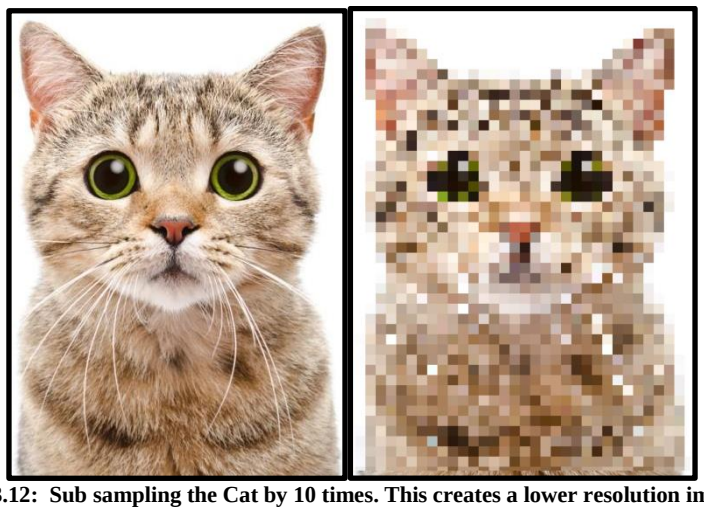

Fig 3.12: Sub sampling the Cat by 10 times. This creates a lower resolution image.

- **Pooling**

  Another building block which can be used in CNN is a pooling layer. Its function is to reduce the amount of computations and parameters by progressively reducing the spatial size of the representation in the network. A pooling layer can be operated independently on each of the feature maps of the representation.

  One of the most common approaches used in pooling is max pooling where the representative of a region is taken by finding the maximum of it.

  For example in the following diagram a 2x2 region is replaced by the maximum value in it.

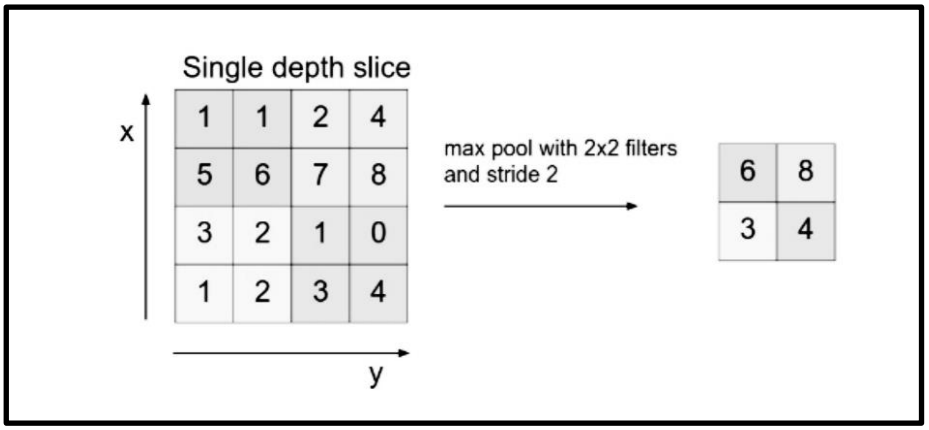

Fig 3.13: Max Pooling

- **Activation**

  In a CNN model architecture, the activation layer is present to control the way with which signal flows from one layer to the next, imitating in a similar fashion how neurons are fired in our brain. As we have learnt that output signals which are related with past references made, would result in activating more neurons. Thereby enabling the signals to be propagated efficiently for the classification.

  CNN uses a wide variety of complex activation functions, the most common function is the Rectified Linear Unit (ReLU), which is favored for its faster training speed.

- **Fully Connected**

  The last layers in the CNN network are fully connected layers, which means that neurons of the preceding layers are connected to every other neuron in subsequent layers. This mimics high level reasoning where all possible pathways from the input to output are considered.

3.4.2: Inception

In this approach, we've used the Inception v3 model of the TensorFlow library. Inception itself is a huge image classification model with millions of parameters that can differentiate various kinds of images. As we have used transfer learning, we have only trained the final layer of the Inception network, such that heavy computations can be avoided and training will end in a reasonable amount of time. Inception -v3 is trained for the ImageNet Large Visual Recognition Challenge using the data from 2012 where it reached a top -5 error rate of as low as 3.46%.

We have performed the transfer learning on the Inception model i.e we downloaded the pre-trained Inception v3 model (trained on ImageNet Dataset consisting of 1000 classes), and added a new final layer corresponding to the number of categories, in our case there are 11 classes as mentioned in the dataset details. Finally we trained the final layer on the dataset.

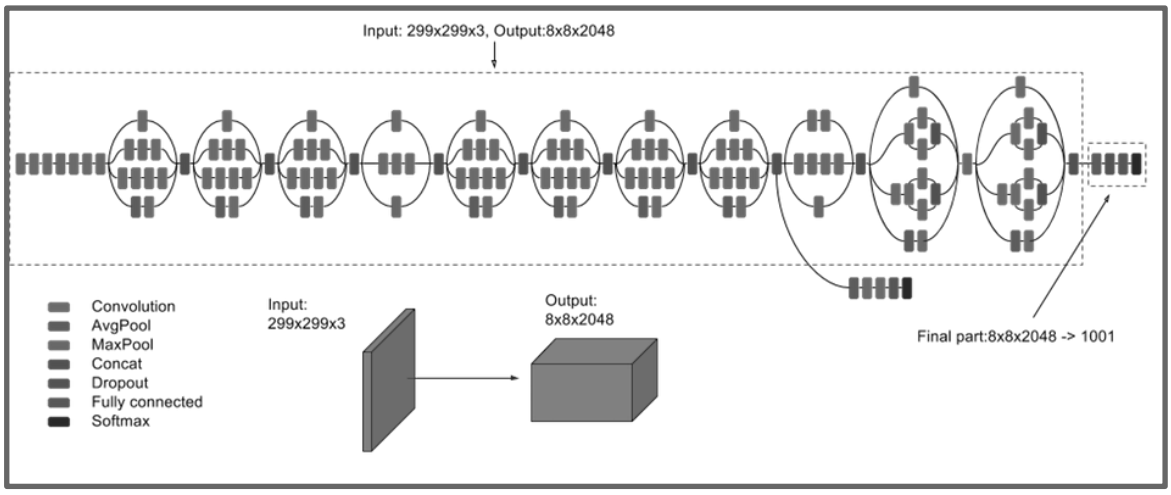

Fig 3.14: Inception-v3 Pre-Trained Model

This kind of information that makes it possible for the model to differentiate among 1,000 classes is also useful for distinguishing other objects. By using this pre-trained network, we are using that information as input to the final classification layer that distinguishes our dataset and generates predictions for test and train dataset.

3.4.3: Recurrent Neural Network (RNN)

An RNN is a type of neural network which has loops in them enabling information to be carried across neurons while taking input. In the following diagram, a part of some RNN, A, takes some input x_t and produces an output of value h_t. The structure of this loop allows the information to flow step by step in the network. The way the recurrent network is making a decision at time step t1 will affect the decision it will make one moment later at next time step t2. In summary, a recurrent neural network takes present input and output of the previous input which combines together to determine in what way they will respond to new data.

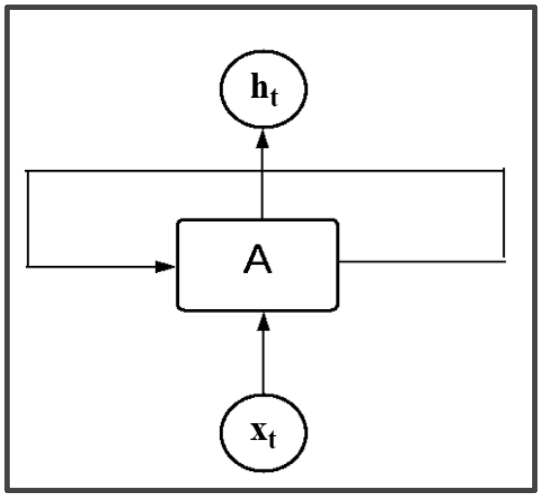

Fig 3.15: A chunk of RNN

A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor. Consider what happens if the loop is unrolled:

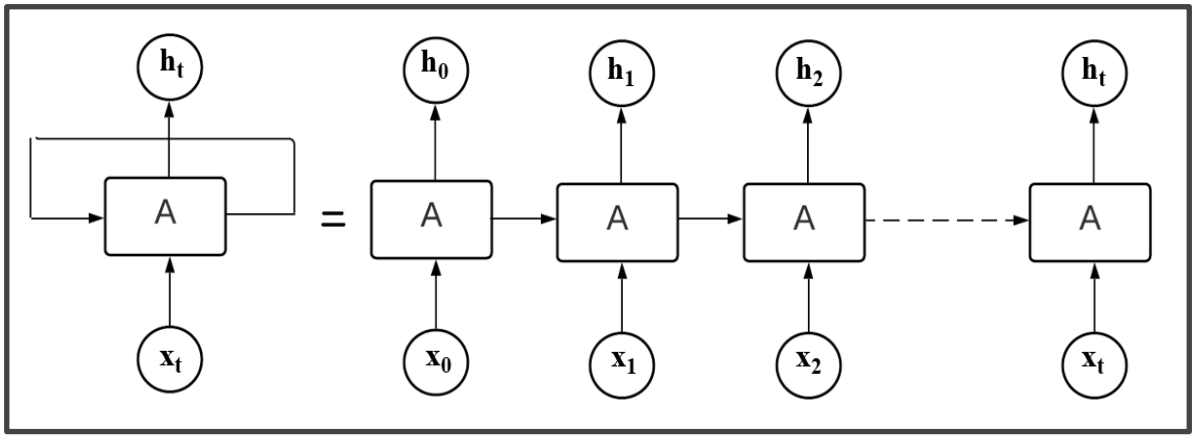

Fig 3.16: An Unrolled recurrent neural network

This chain-like nature reveals that recurrent neural networks are intimately related to sequences and lists. They're the natural architecture of a neural network to use for such data. The sequential information is preserved in the recurrent network's hidden state, which manages to span many time steps as it cascades forward to affect the processing of each new example.

- **Exploding and Vanishing Gradient Problem**

  When we look at the RNN in theory, they are very much capable of handling long term dependency. But unfortunately in practice, RNN don't seem to be able to learn and retain that. The gradient expresses the change in all weights with regard to the change in error. Since the layers and time steps of deep neural networks relate to each other through multiplication, gradients are susceptible to vanishing or exploding if not taken care.

3.4.4: Long Short-Term Memory Units (LSTMs)

A variation of recurrent net with "Long Short-Term Memory Units" LSTMs, was proposed by the German researchers Sepp Hochreiter and Juergen Schmidhuber as a solution to the vanishing gradient problem.

LSTMs help preserve the error that can be backpropagated through time and layers. By maintaining a more constant error, they allow recurrent nets to continue to learn over many time steps (over 1000), thereby opening a channel to link causes and effects remotely.

LSTMs are explicitly designed to avoid the long term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn.

Architecture of RNN Model Used:

- The RNN model created is based on LSTM units.
- The first layer is an input layer used to feed input to the upcoming layers. Its size is determined by the size of the input being fed. Our Model is a wide network consisting of a single layer of 256 LSTM units.
- The input layer is followed by a fully connected layer with softmax activation. In Fully Connected every neuron is connected to every neuron of the previous layer. The fully connected layer consists of as many neurons as there are categories/classes.
- Finally a regression layer is used to apply a regression (linear or logistic) to the provided input. We used adam (Adaptive Moment Estimation) which is a stochastic optimizer, as a gradient descent optimizer to minimize the provided loss function "categorical_crossentropy" (which calculates the errors).

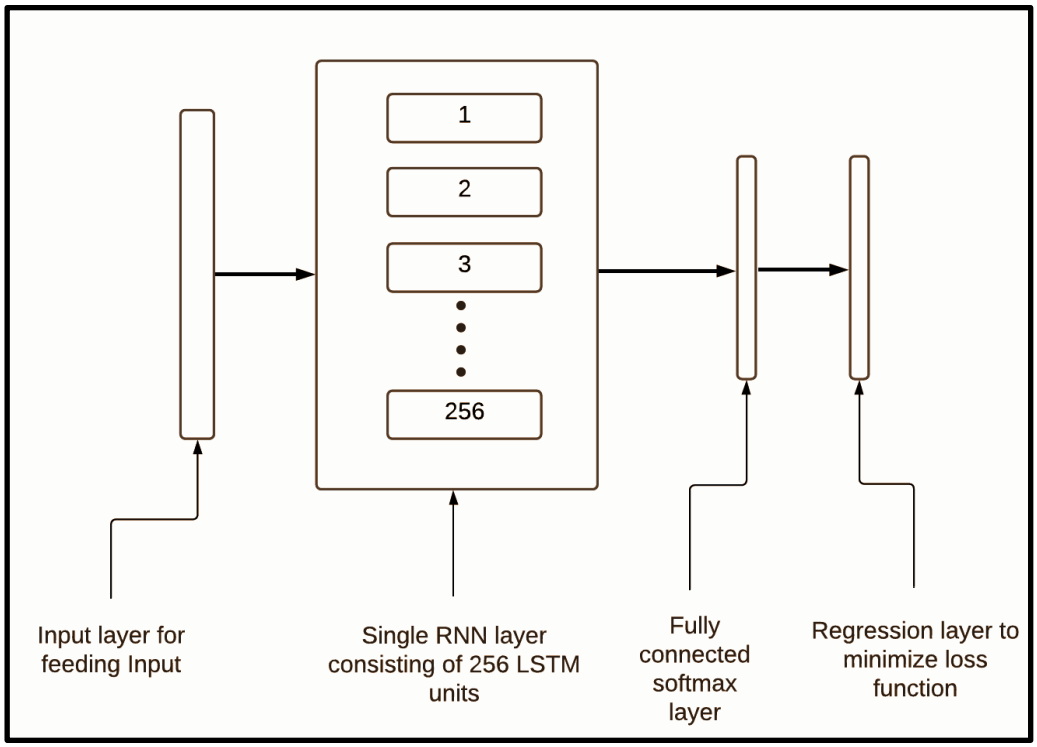

Fig 3.17: RNN model used in Approach 1

A wider RNN network with 512 LSTM units and another deep RNN network with three layers of 64 LSTM units each was also tried and tested on the same input. After testing these on a sample of the dataset, it was found that the wide model with 256 LSTM units performed the best and therefore only the wide model was used for training and testing on a complete dataset.

3.4.5: Algorithm Design

A. As video sequences contain both the temporal as well as the spatial features, a pipeline of CNN and RNN is used.
B. First, frames from the multiple video sequences of each gesture are extracted to train on CNN.
C. After this, noise from the frames i.e background, body parts other than hand are removed to extract more relevant features from the frame.
D. To train the model on the spatial features of the video sequences of gestures, we have used the Inception V3 model which is a deep CNN. It was trained on the grayscale frames obtained from the video sequences of train data.
E. Store the train and test frame predictions. The model obtained in the above step for the prediction of frames will be used.
F. Trained CNN model was used and forwarded to make predictions for individual frames to obtain a sequence of predictions.
G. Now this sequence of prediction or pool layer outputs was given to RNN to train on the temporal features.
H. RNN (recurrent neural network) is used to train the model on the temporal features so as to relate the frames of video in the course of time.
I. Using the predictions by CNN as input for RNN, 93.93% accuracy was obtained.

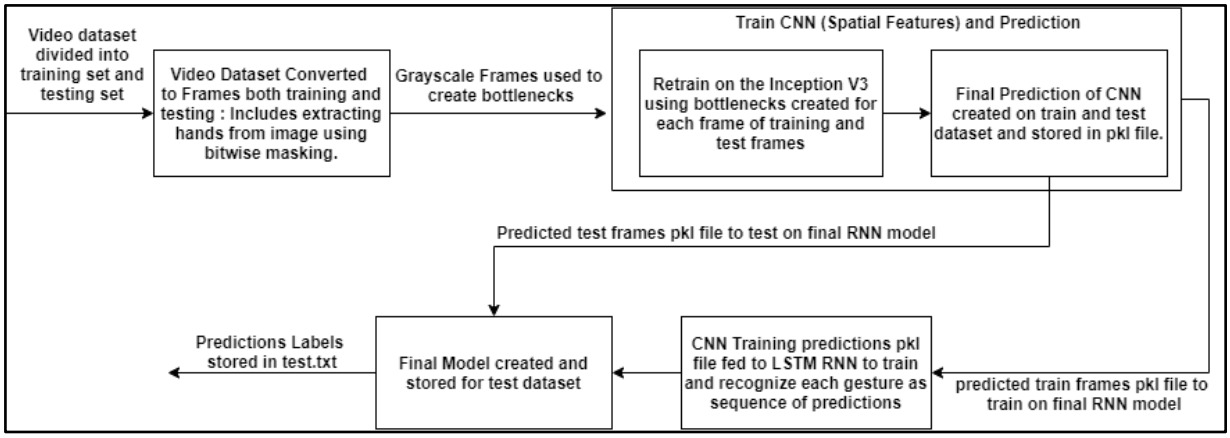

Fig 3.18: Overview of Indian Sign Language to Text Conversion (Approach 1)

- **<u>Frame Extraction and Background Removal:</u>**

  Each video gesture is broken down into a sequence of frames. Frames are then processed to remove all the noise from the image that is everything except hands. The final image consists of a grayscale image of hands to avoid color specific learning of the model.

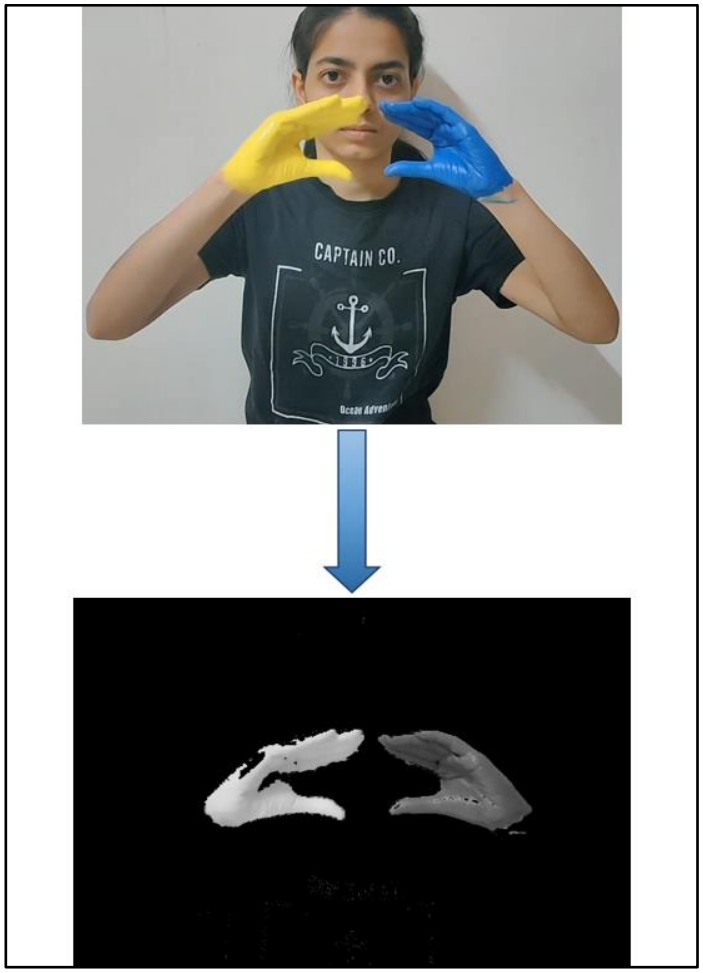

Fig 3.19: One of the extracted frames (Above) and frames after extracting hands (Below)

Fig 3.20: Series of frames from the sample gesture after background removal

- **Train CNN (Spatial Features) and Prediction:**

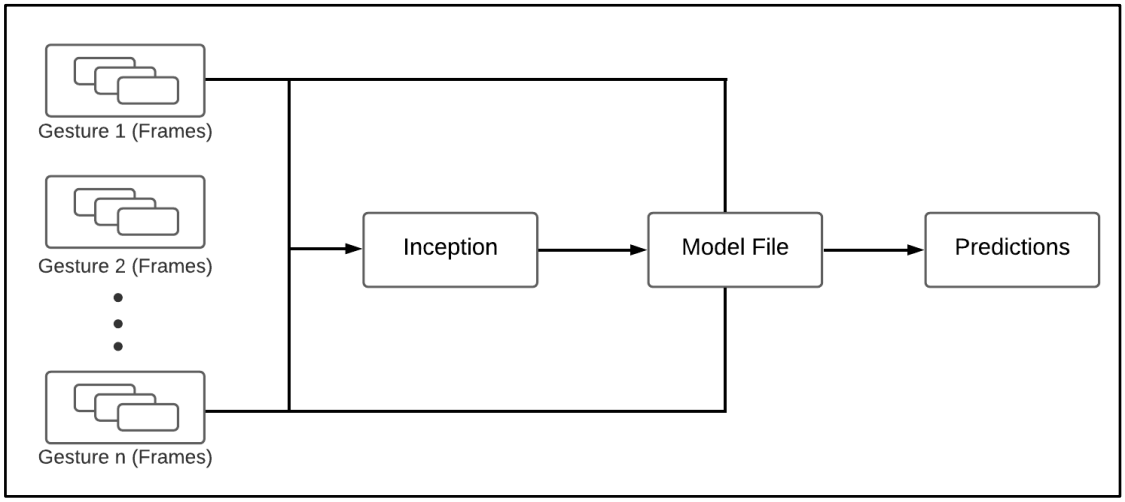

Fig 3.21: Overview of CNN model generating predictions of test and train frames

The first row in the illustration below is the video of a “Morning” gesture. The second row shows the set of frames extracted from it. The third row shows the sequence of predictions for each frame by CNN after training it.

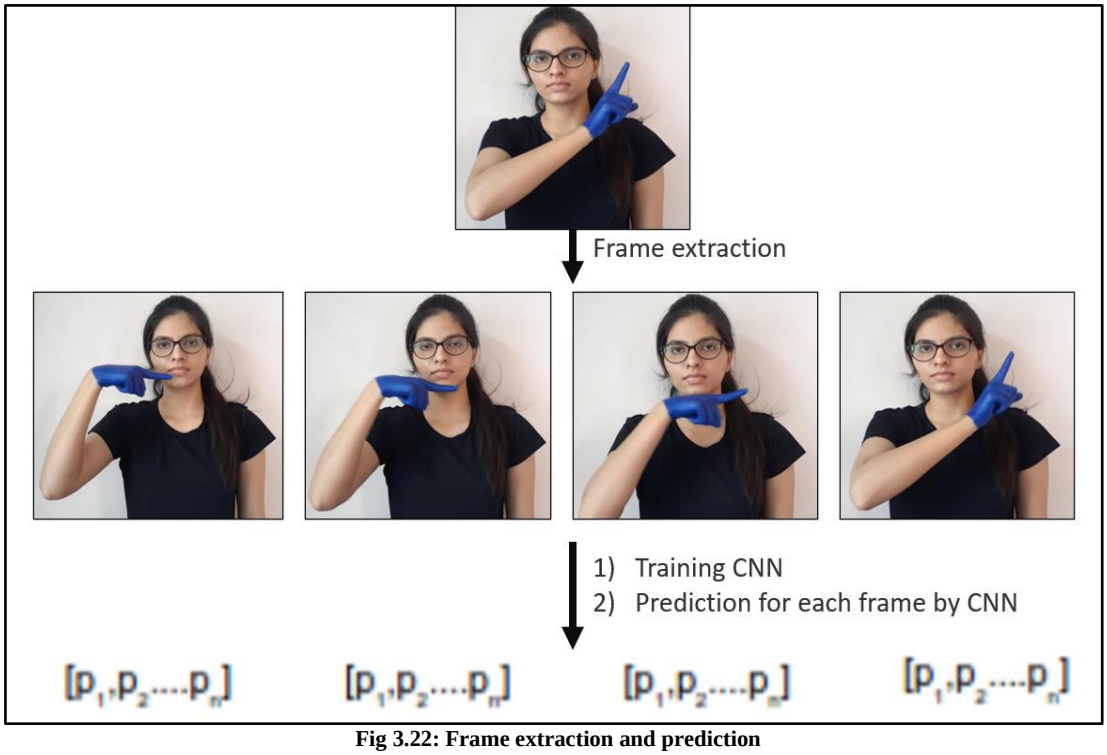

Fig 3.22: Frame extraction and prediction

- **Training RNN (Temporal Features):**

  The videos for each gesture are fed to RNN as sequence of predictions of its constituent frames. The RNN learns to recognize each gesture as a sequence of predictions. After the Training of RNN completes a model file is created.

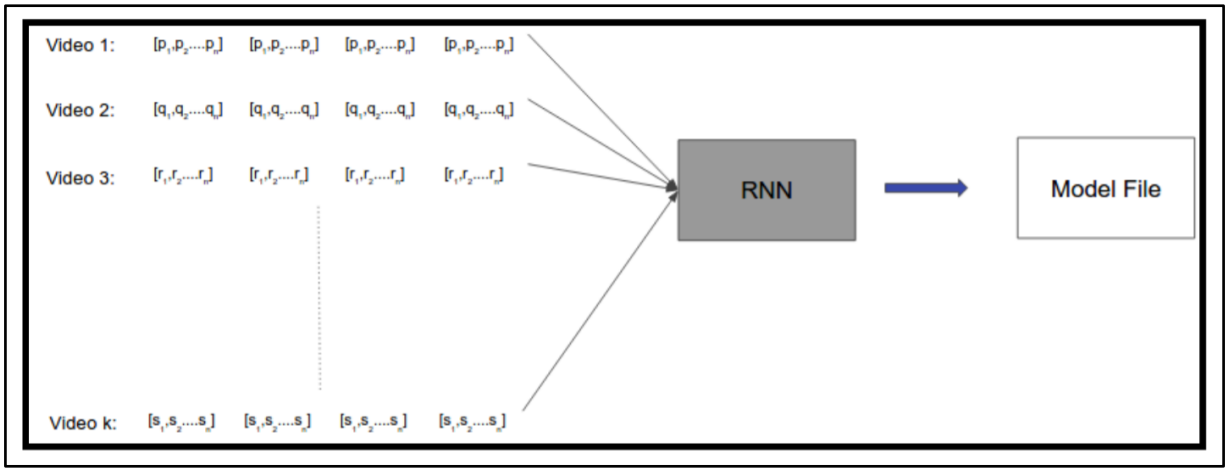

Fig 3.23: Overview of RNN model processing predictions generated by CNN

The predictions of CNN for frames of the test set were fed to the trained model for test.

3.5: Indian Sign Language to English Text (Approach 2)

The objective of this module is to perform real-time detection of ISL with the help of mediapipe and classification module.

In this approach video feed will be taken as input with the help of an integrated webcam in which users will perform some signs of ISL. The frames from the video feed are used for detection of signs. At first the module will identify if the sign performed by the user is single hand sign or double hand sign after segregation of hand frames are passed through respective trained modules and the detected result along with its accuracy is shown on the screen of webapp.

3.5.1: Mediapipe

Mediapipe [38] is an open source cross-platform and graph-based framework developed by Google used for building customized ML solutions for live and streaming media.

MediaPipe Hands is a solution for hand tracking and reliance on high fingers. It uses machine learning (ML) to integrate 21 3D landmarks from just one frame.

MediaPipe Hands uses an ML pipeline with multiple models working together: A palm detection model that works on a full-blown image and returns a box attached to a hand-binding. A hand-crafted model that operates in a fixed image region defined by a palm detector and retrieves a high 3D key manually.

After the palm detection over the whole image our subsequent hand landmark model performs precise keypoint localization of 21 3D hand-knuckle coordinates inside the detected hand regions via regression, that is direct coordinate prediction. The model learns a consistent internal hand pose representation and is robust even to partially visible hands and self-occlusions.

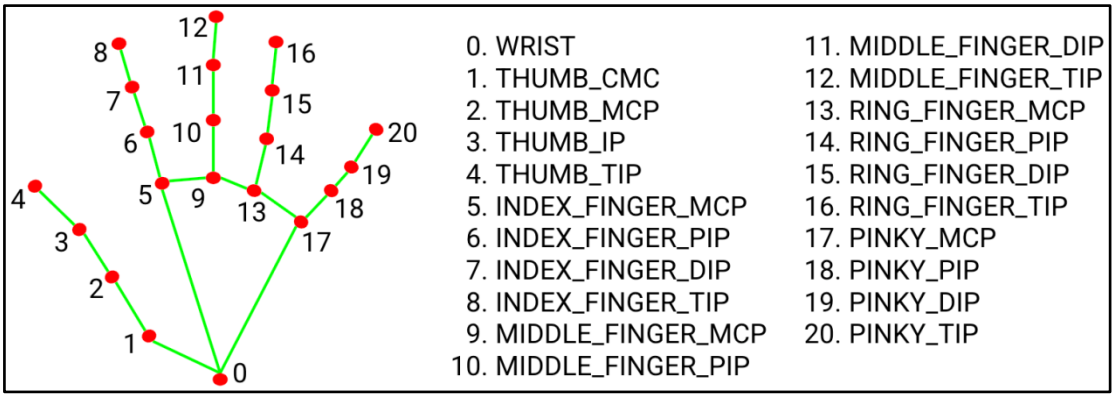

Fig 3.24: Mediapipe Landmarks

3.5.2 Supervised Machine Learning Classifiers:

In supervised learning, classification is a technique for categorizing a given set of data into classes; it can be implemented on both structured and unstructured kinds of data. It is basically used to determine which class the dependent belongs to based on one or more independent variables. The classes are often referred to as labels, targets or categories.

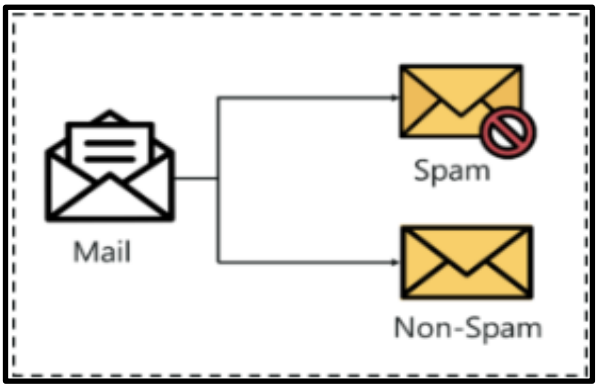

Fig 3.25: Classification Example

The classification predictive models task is to approximate the mapping function from input variables to discrete output variables. The main goal is to identify which class/category the new data will fall into. The most common classification problems are – speech recognition, face detection, handwriting recognition, document classification, etc. It can be either a binary classification problem or a multi-class problem.

1. Binary Classification – It is a type of classification with two outcomes, for eg – either true or false.
2. Multi-Class Classification – The classification with more than two classes, in multi-class classification each sample is assigned to one and only one label or target.

Below are some classifiers used in model:

Logistic Regression:

It is a classification algorithm in machine learning that uses one or more independent variables to determine an outcome. The outcome is measured with a dichotomous variable meaning it will have only two possible outcomes.

The goal of logistic regression is to find a best-fitting relationship between the dependent variable and a set of independent variables. It is better than other binary classification algorithms like nearest neighbor since it quantitatively explains the factors leading to classification.

Ridge Classifier:

The Ridge Classifier, based on the Ridge regression method, converts the label data into {-1, 1} and solves the problem with the regression method. The highest value in prediction is accepted as a target class and for multiclass data multi-output regression is applied.

Ensemble methods for classification:

Ensemble method in machine learning is basically a technique used to combine several supervised learning models that are individually trained and the results merged in various ways to achieve one optimal predictive model which will give the final prediction. This result has higher predictive power than the results of any of its constituting learning algorithms independently.

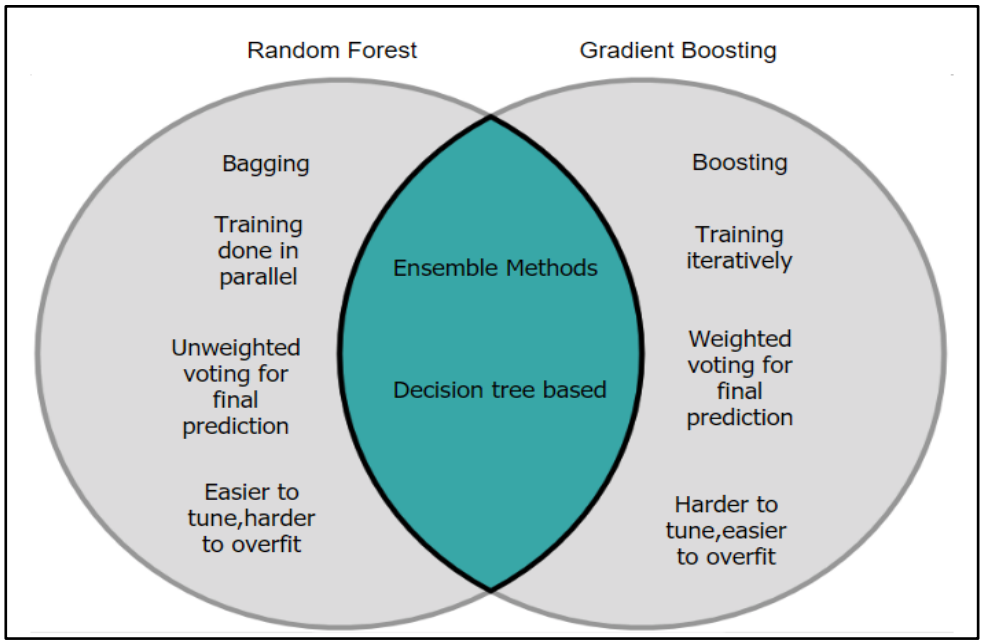

Fig 3.26: Emsembled Classifiers

Bagging: Bagging is an acronym for 'Bootstrap Aggregation' and is used when the goal is to reduce the variance of a decision tree classifier. Bagging is a parallel method that fits different, considered learners independently from each other, making it possible to train them simultaneously. The objective is to create several subsets of data from training samples chosen randomly with replacement. Each collection of subset data is used to train their decision trees.These multi datasets are used to train multiple models in parallel. The average of all the predictions from different ensemble models is calculated. The majority vote gained from the voting mechanism is considered when classification is made. Bagging decreases the variance and is more robust than a single decision tree classifier.

Boosting: Boosting is used to create a collection of predictors. In this technique, the sequential ensemble method iteratively adjusts the weight of observation as per the last classification tree. If an observation is incorrectly classified, it increases the weight of that observation. The term 'Boosting' in a layman language, refers to algorithms that convert a weak learner to a stronger one. It decreases the bias error and builds strong predictive models.

Consecutive trees (random sample) are fit and at every step, the goal is to improve the accuracy from the prior tree.

Data points mispredicted in each iteration are spotted, and their weights are increased so that the next hypothesis is more likely to classify it correctly. This process converts weak learners into better performing models.

Random Forest Classifier:

Random forest classifier is a supervised learning algorithm based on bagging i.e bootstrap aggregation. The general idea of the bagging method is that a combination of learning models increases the overall result.

Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

Random forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

Algorithm:

A. Select random samples from a given dataset.
B. Construct a decision tree for each sample and get a prediction result from each decision tree.
C. Perform a vote for each predicted result.
D. Select the prediction result with the most votes as the final prediction.

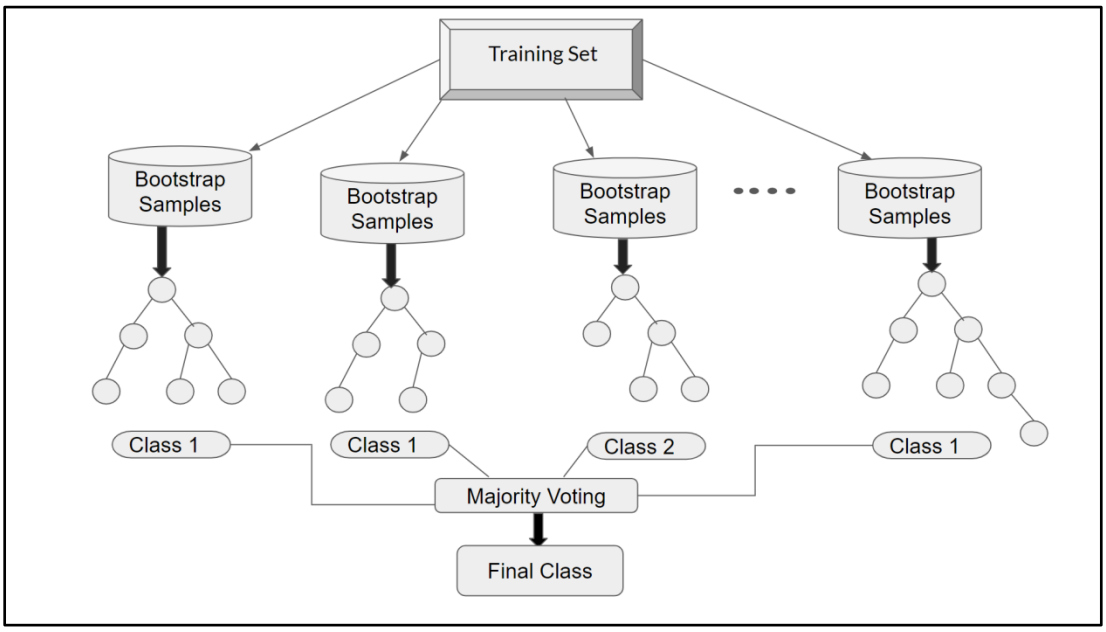

Fig 3.27: Random Forest Classifier

Deep decision trees may suffer from overfitting, but random forests prevent overfitting by creating trees on random subsets. The main reason is that it takes the average of all the predictions, which cancels out the biases.

Gradient Boosting:

Gradient boosting classifier is a boosting ensemble method. Boosting is a way to combine (ensemble) weak learners, primarily to reduce prediction bias. Instead of creating a pool of predictors, as in bagging, boosting produces a cascade of them, where each output is the input for the following learner. Typically, in a bagging algorithm trees are grown in parallel to get the average prediction across all trees, where each tree is built on a sample of original data. Gradient boosting, on the other hand, takes a sequential approach to obtaining predictions instead of parallelizing the

tree building process. In gradient boosting, each decision tree predicts the error of the previous decision tree—thereby *boosting* (improving) the error (gradient).

Algorithm:

A. Initialize predictions with a simple base model..
B. Calculate residual (actual-prediction) value.
C. Build another shallow decision tree that predicts residual based on all the independent values.
D. Update the original prediction with the new prediction multiplied by learning rate.
E. Repeat steps two through four for a certain number of iterations (the number of iterations will be the number of trees).

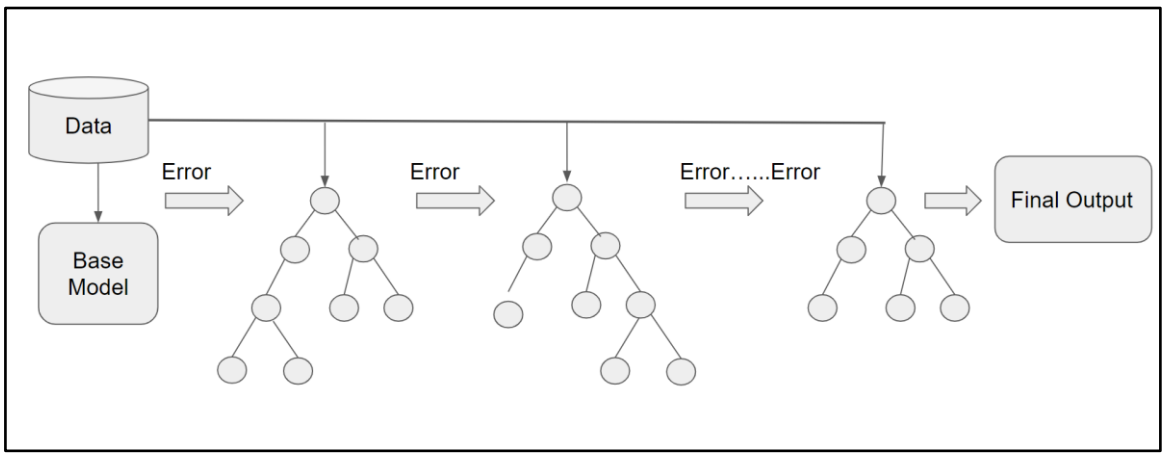

Fig 3.28: Gradient Boosting Classifier

3.5.3 Dataset Collection and Training of Model:

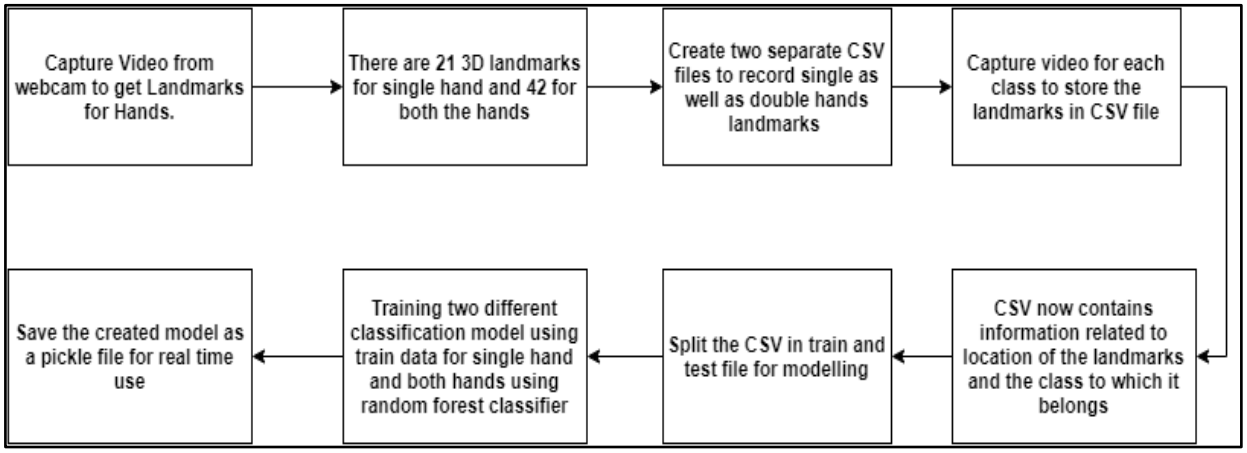

Fig 3.29: Model Training Flowchart

Dataset Collection:

Step 1: Capture live video feed with the help of opencv module and draw landmarks for both the hand with the help of mediapipe module. There are 21 3D landmarks for single hands and 42 3D landmarks for both hands.

Step 2: For collecting records of 3D landmarks we created two separate CSV files, one for recording 21 3D landmarks of a single hand and another one for 42 landmarks of both hands.

Step 3: To collect landmark records for each class in respective CSVs. Users have to perform signs for each class in front of the webcam. First column in both CSV contains class information and the further column contains information related to relative location of landmarks in the frame.

Training and Testing of model:

Step1: For training and testing purposes both CSV files are split into 70/30. Random 70% of CSV file record is used for training of the model and 30% of record is used to test the trained model.

Step2: Now the 70% record from each CSV is used to train through 4 different Machine Learning Classification models.

1. LogisticRegression
2. RidgeClassifier
3. RandomForestClassifier
4. GradientBoostingClassifier

Testing result on 30% record for Single Hand Model:

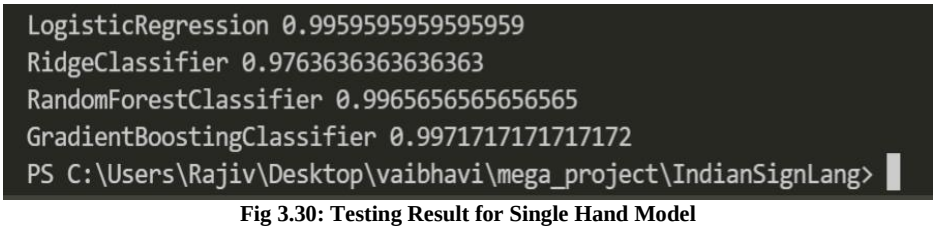

Fig 3.30: Testing Result for Single Hand Model

Testing result on 30% record for Double Hand Model:

```
LogisticRegression 0.9990476190476191
RidgeClassifier 1.0
RandomForestClassifier 0.9996825396825397
GradientBoostingClassifier 0.9993650793650793
PS C:\Users\Rajiv\Desktop\vaibhavi\mega_project\IndianSignLang>
```

Fig 3.31: Testing Result for Double Hand Model

Step3: For Real-Time detection we used Random Forest Classifier to train the model because it is giving best accuracy when tested on 30% in both models and it does not suffer over which is saved into pickle file for real-time use. Two seperate pickle files are generated one for single hand and other for both hands for more accurate result.

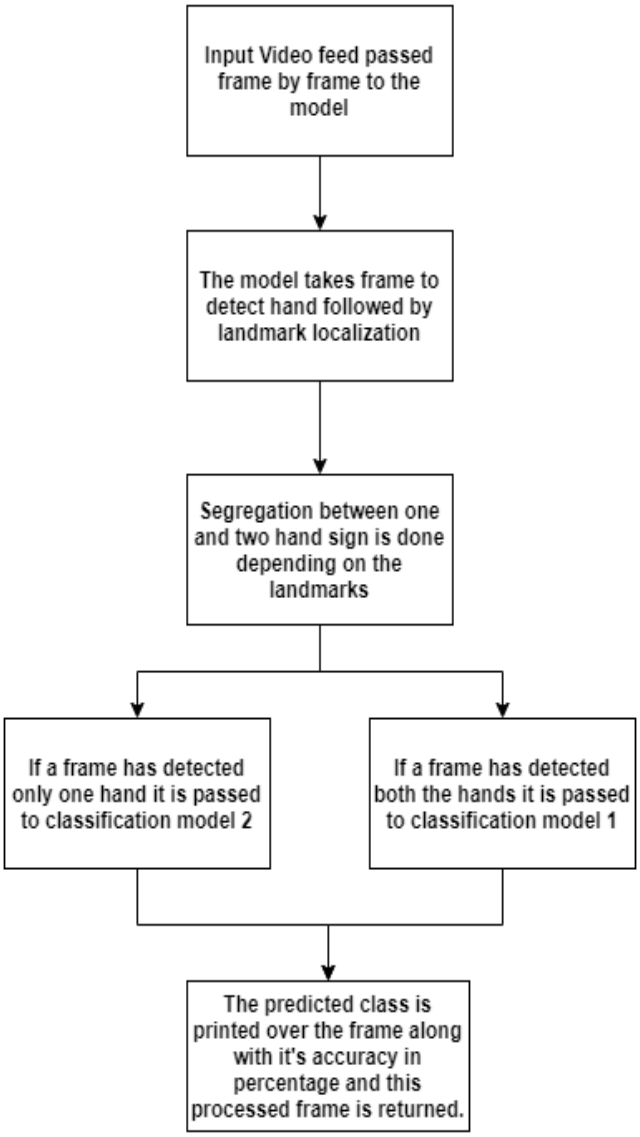

Fig 3.32: ISL model Flowchart

Step1: User has to perform signs of ISL in front of the webcam . Live input video feed is read frame by frame using opencv module and with the help of mediapipe hand is localized and 3D landmarks are drawn on hand.

Step2: Frames read from video feed are passed to a detection module which firstly segregates between one and two hand signs depending on the recorded landmarks.

Step3: If the detection module identifies one hand as a frame based on the landmark then recorded landmarks from frame are passed through model 2 which has Single hand trained pickle file model.
Else if both hand landmarks recorded from frames is passed through model 2 which contain Double hand trained pickle file model.

Step4: Predicted class is displayed on screen along with its accuracy.

Chapter 4:

RESULTS

This section deals with the user interface regarding the front end of TransApp. First page is the landing page displaying various modules as:

- Language Translation: This module will help users translate any text to a language of choice.
- Image Translation: This will enable users to translate the content of any image file into another language.
- Document Translation: Users can upload any document in pdf or doc format and it will be translated into a language of choice.
- Text to ISL translation: This module can translate any text/sentence in English to corresponding ISL gestures.
- ISL to English Text: Users can sign gestures in front of the webcam and the predicted gesture will be displayed.

The navigation bar is also provided for easy access to other modules within the webapp.

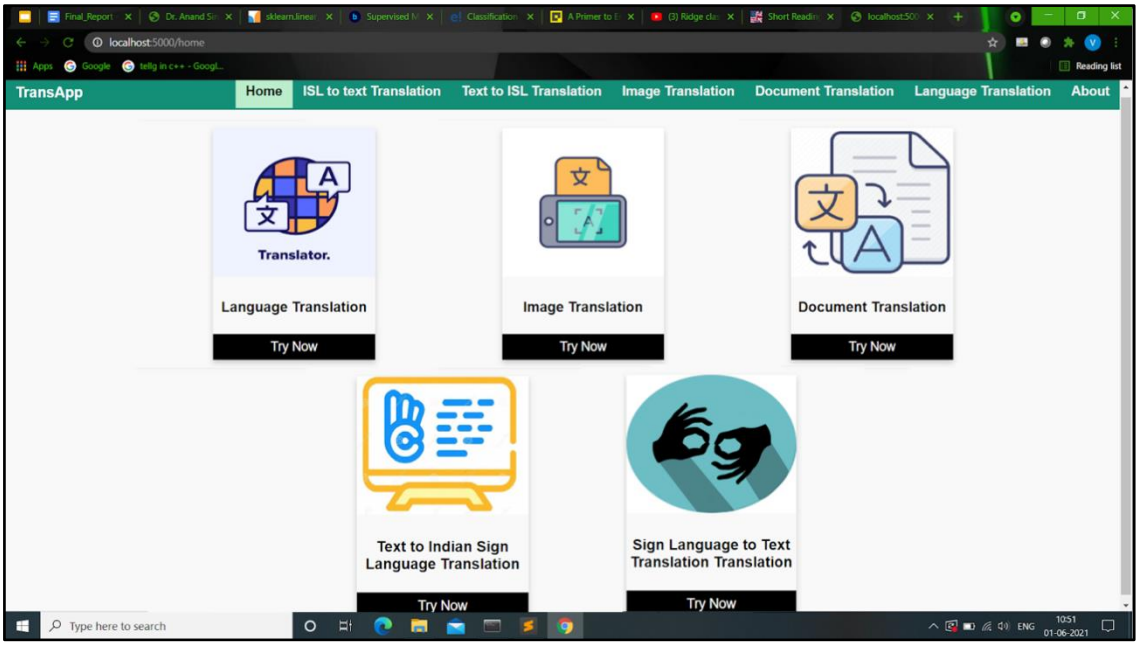

Fig 4.1: Landing Page of TransApp

4.1: Language Translation

This module can help users translate any text to the desired language of choice.

- The module supports automatic detection of input text as well as provides a wide range of language options to choose from.
- Users can click on “Please Select a Language” and a drop-down menu will list the languages.
- The translated text will be displayed in a separate box beside the input field after the user has pressed the “Translate” button.

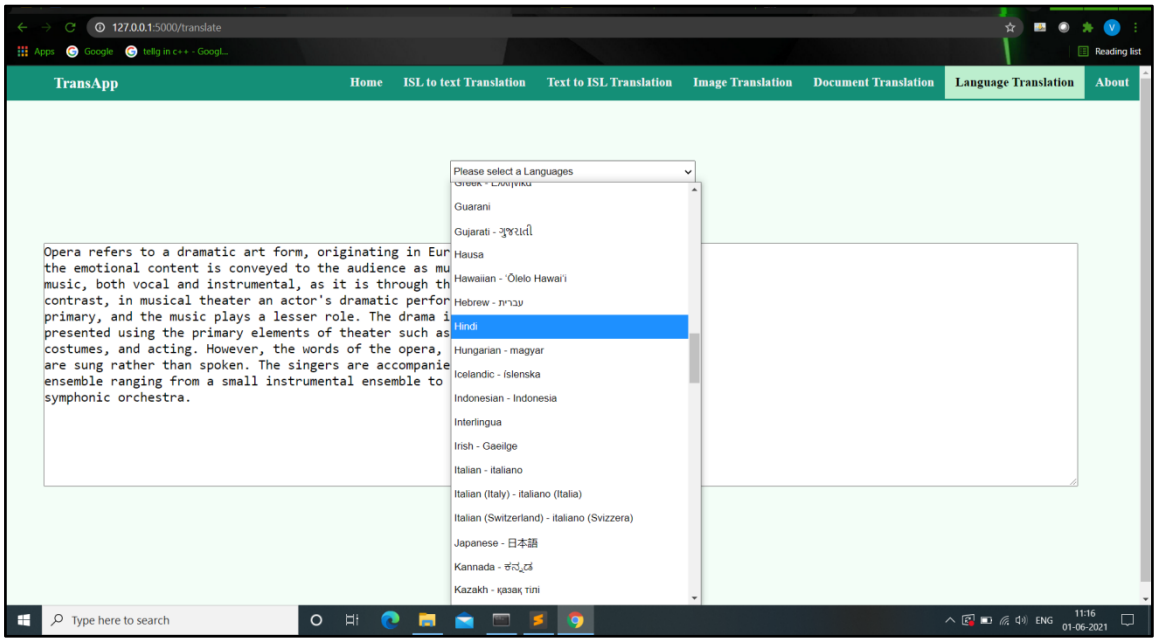

Fig 4.2: UI of Language Translator

A sample of English text translated into Hindi

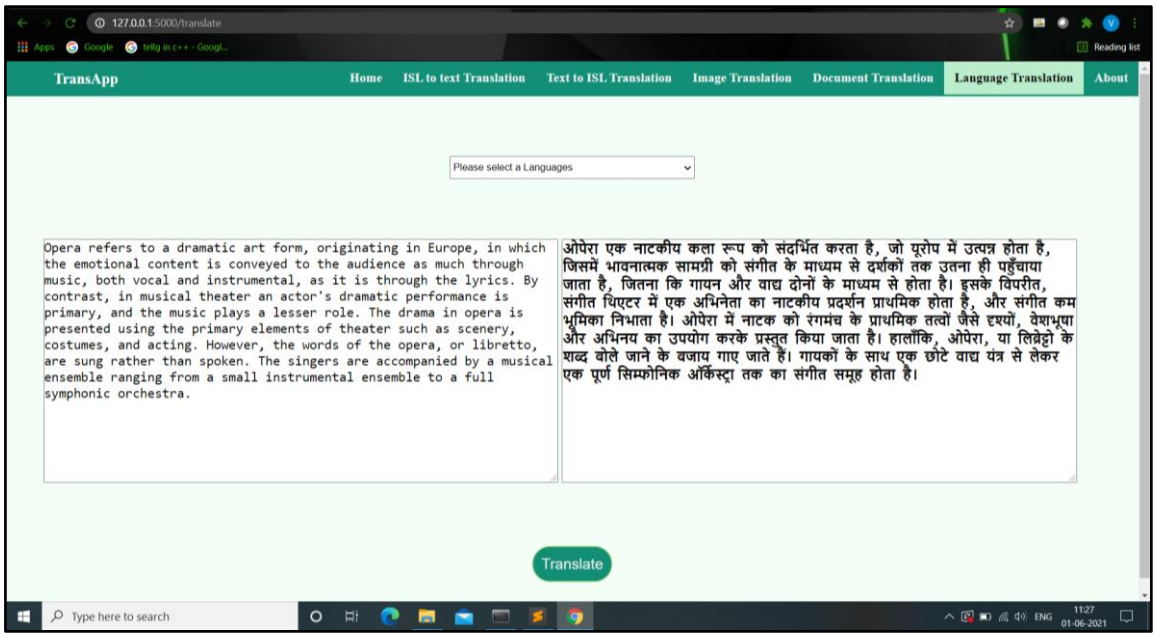

Fig 4.3: UI of Language Translator

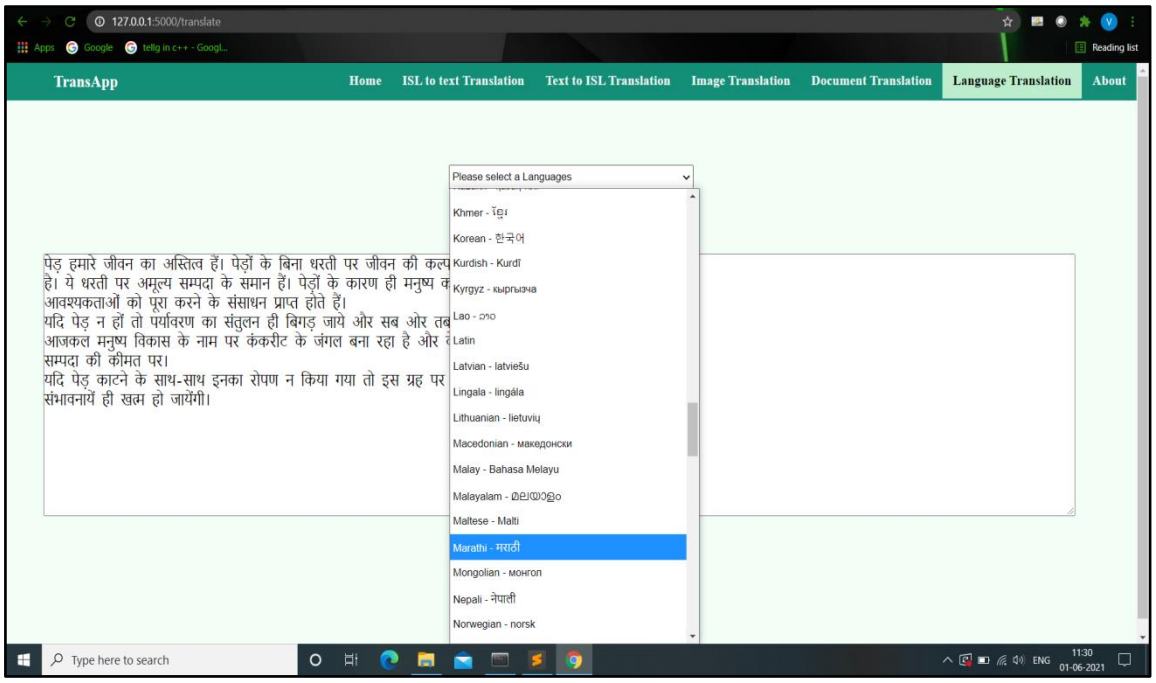

Fig 4.4: UI of Language Translator

A sample of text in Hindi translated into Marathi and displayed in the output box.

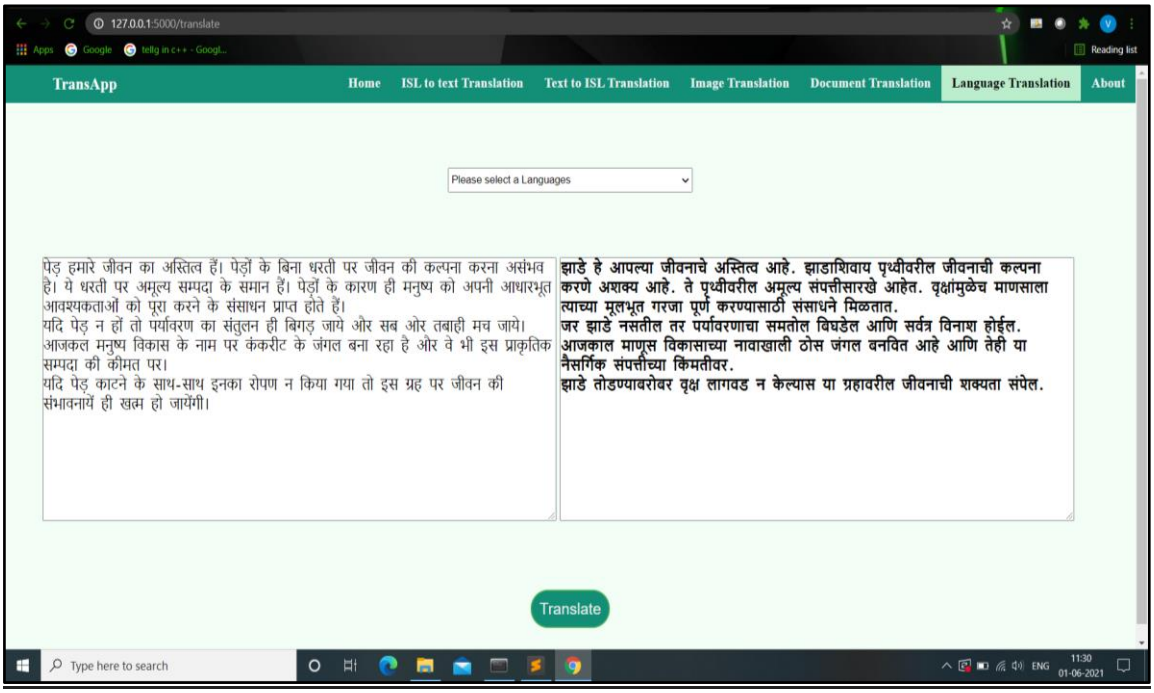

Fig 4.5: UI of Language Translator

4.2: Document translation

This module can help users translate any document file (e.g pdf or word file) to the desired language of choice.

- The UI first shows an option to upload a file by clicking on the “Choose File” button. This will open a window using which users can select the required file from their device’s storage.
- Next, an option is provided to select languages from a drop-down menu.
- After clicking on the “Translate” button, the content of the uploaded file will be extracted and translated into the destination language. This module also supports automatic detection of language.
- This translated text is then displayed on the box below.

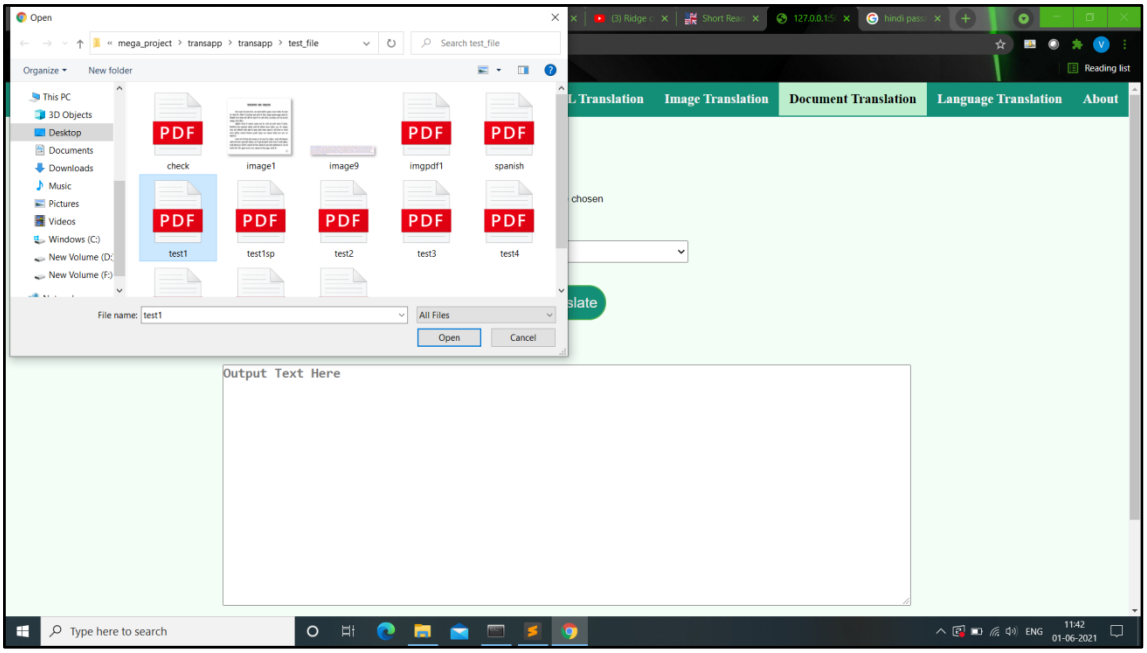

Fig 4.6: UI of Document Translator

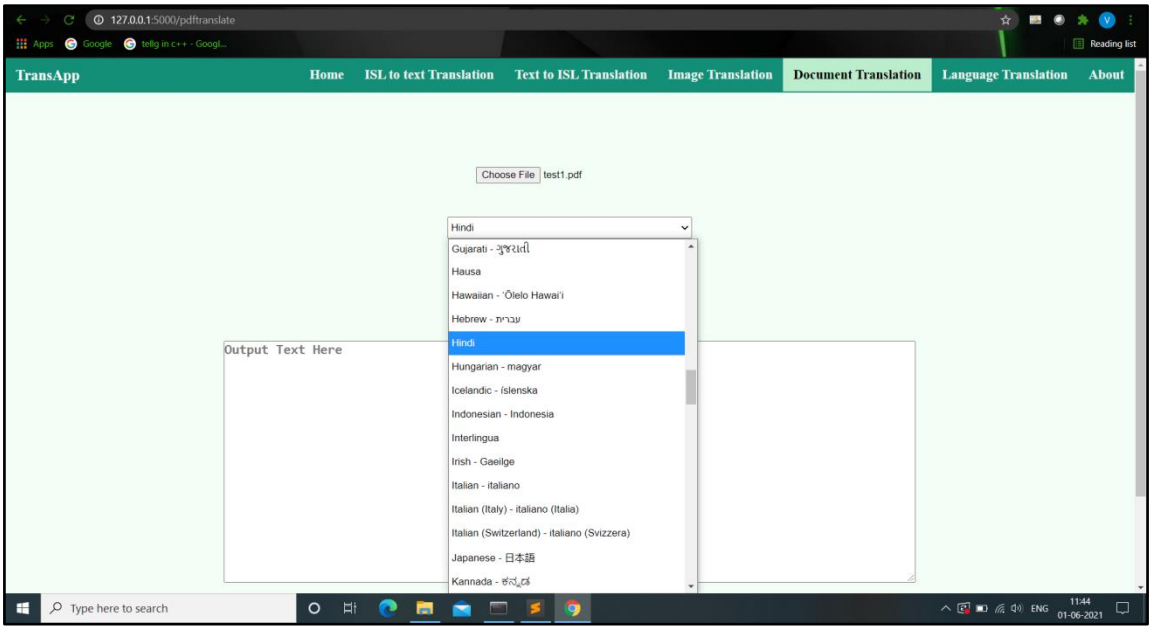

Fig 4.7: UI of Document Translator

The content of the test1.pdf file in English is translated into Hindi.

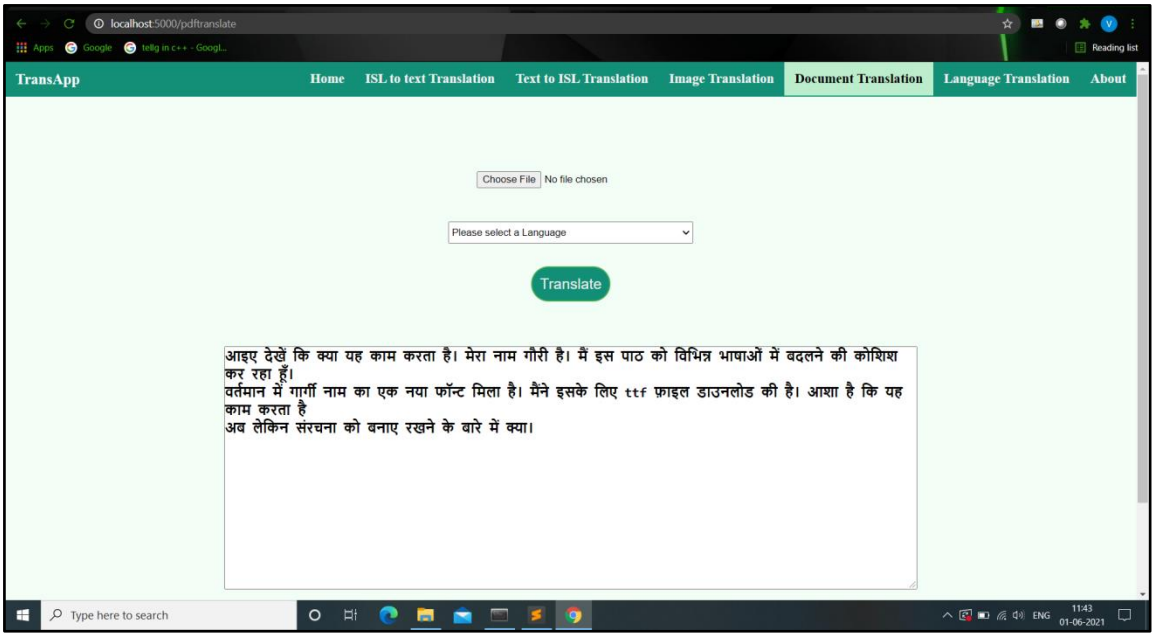

Fig 4.8: UI of Document Translator

4.3: Image Translation

This module can help users translate the content of an image file (e.g jpg, png etc) by extracting it using Optical Character Recognition (OCR). The extracted text is then translated to the destination language of choice.

- The UI first displays an option “Choose File” to select the image file from users’ storage to upload it.
- The backend then extracts text from the image using OCR and the text is extracted.
- An option “Please select a language” displays a list of languages for the user to choose from.
- After selecting the destination language, the user can click on the “Translate” button and the translated text will be displayed in the output box.

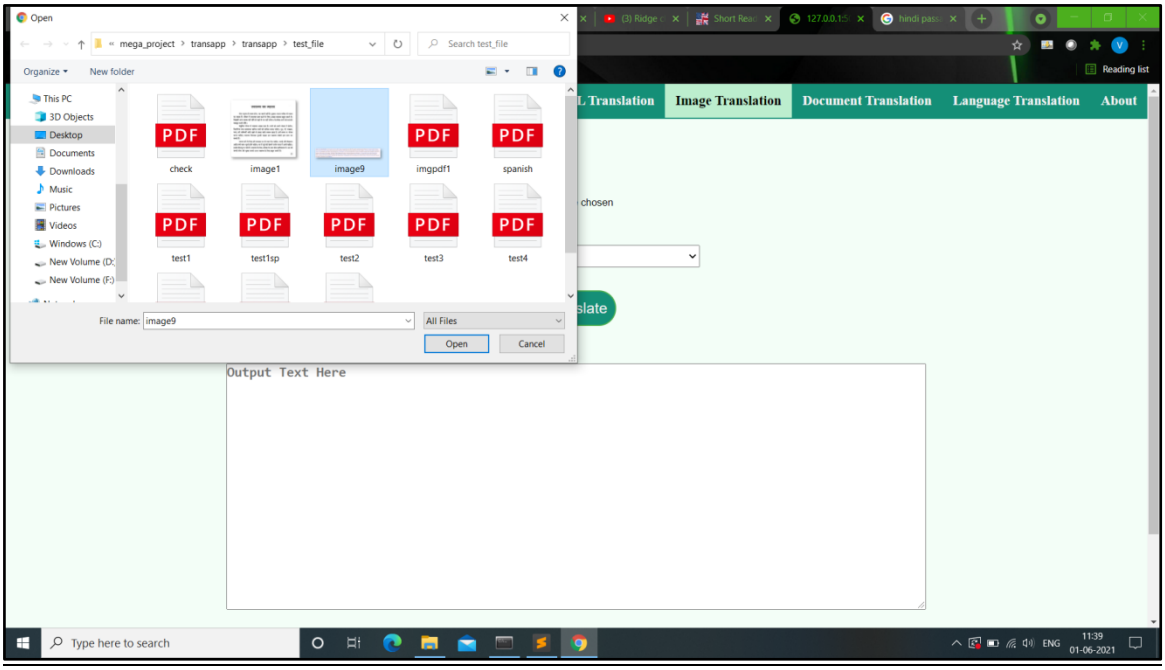

Fig 4.9: UI of Image Translator

Below is a sample file ‘image9.jpg’ taken as an input file and the destination language chosen as “Hindi”..

Vietnam has submitted films for the Academy Award for Best Foreign Language Film since 1993. The award is presented annually by the U.S. Academy of Motion Picture Arts and Sciences to a feature-length motion picture produced outside the United States that contains primarily non-English dialogue. *The Scent of Green Papaya*, directed by Trần Anh Hùng *(pictured)*, was Vietnam's first submission for the 1993 awards. It is the only Vietnamese film to secure a nomination and was the first nomination received by a Southeast Asian country in the category. *The Scent of Green Papaya* and the three subsequent Vietnamese submissions – Hồ Quang Minh's *Gone, Gone Forever Gone* (1996), Tony Bui's *Three Seasons* (1999) and Hùng's *Vertical Ray of the Sun* (2000) – were directed by overseas Vietnamese directors and chosen without any support councils, deriving solely from the directors' relationship with foreign partners. *The Buffalo Boy* was the first selection by Vietnam's Ministry of Culture and Information, following an invitation to participate in 2006. (**Full list...**)

Fig 4.10: Sample Input PDF file

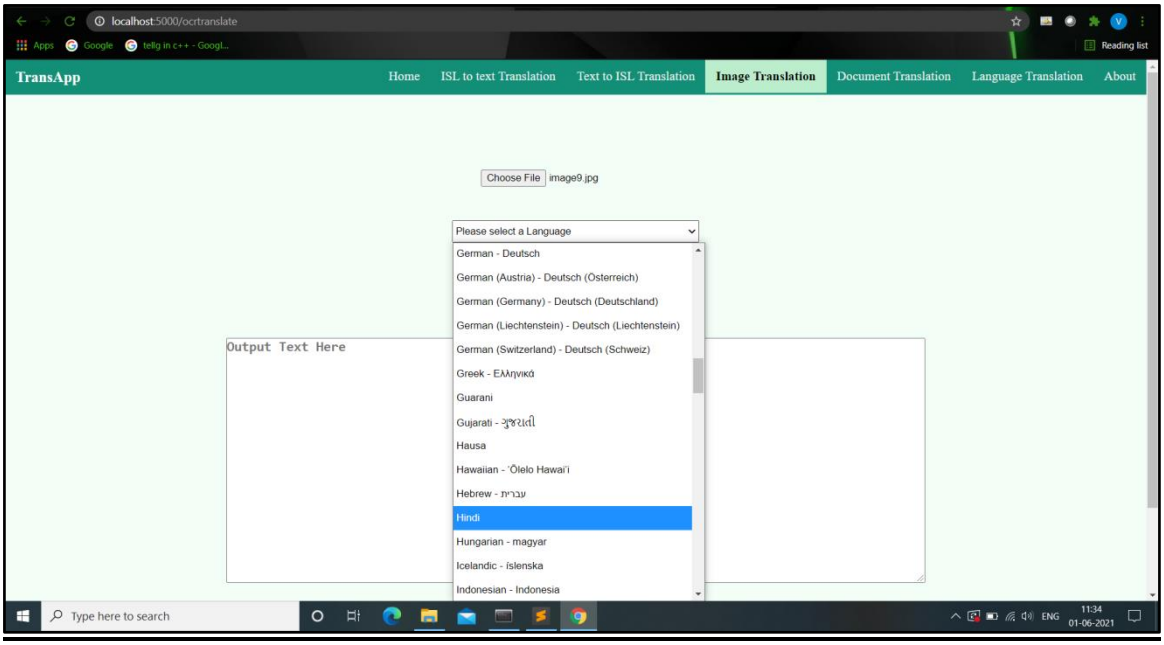

Fig 4.11: UI of Image Translator

The content of the image file is now extracted and the translated text in Hindi is displayed in the output box.

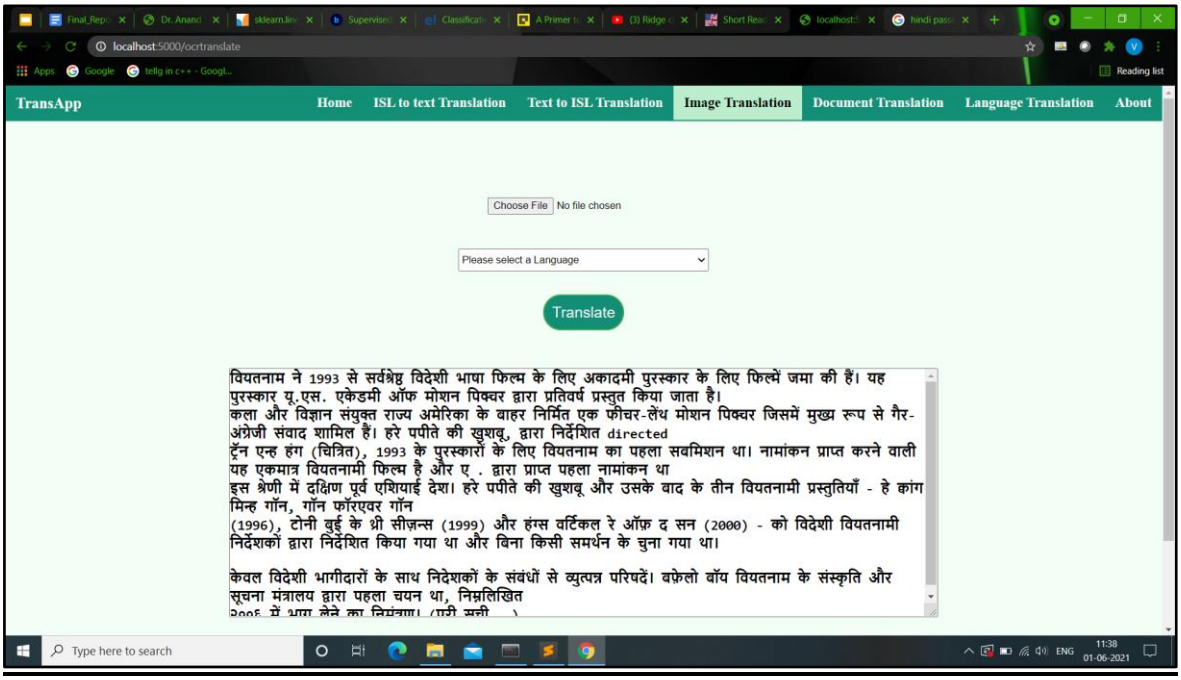

Fig 4.12: UI of Image Translator

4.4: Text to Sign Language Conversion Module

This module can facilitate users to translate English text to corresponding ISL gestures.

- This first takes the input sentence from users in the left box as shown in the below image.
- After the user clicks on the “Translate” button, in the backend the English text is parsed to create a phrase structure based on ISL’s grammar representation.
- After parsing, reordering is done and unwanted words are removed using a list of stopwords provided and the output of which is lemmatized.
- The sentence in ISL format is displayed in the output box and the concatenated video of ISL gestures is played on the screen.
- The gesture words not present in the dictionary are given out alphabetically as output.

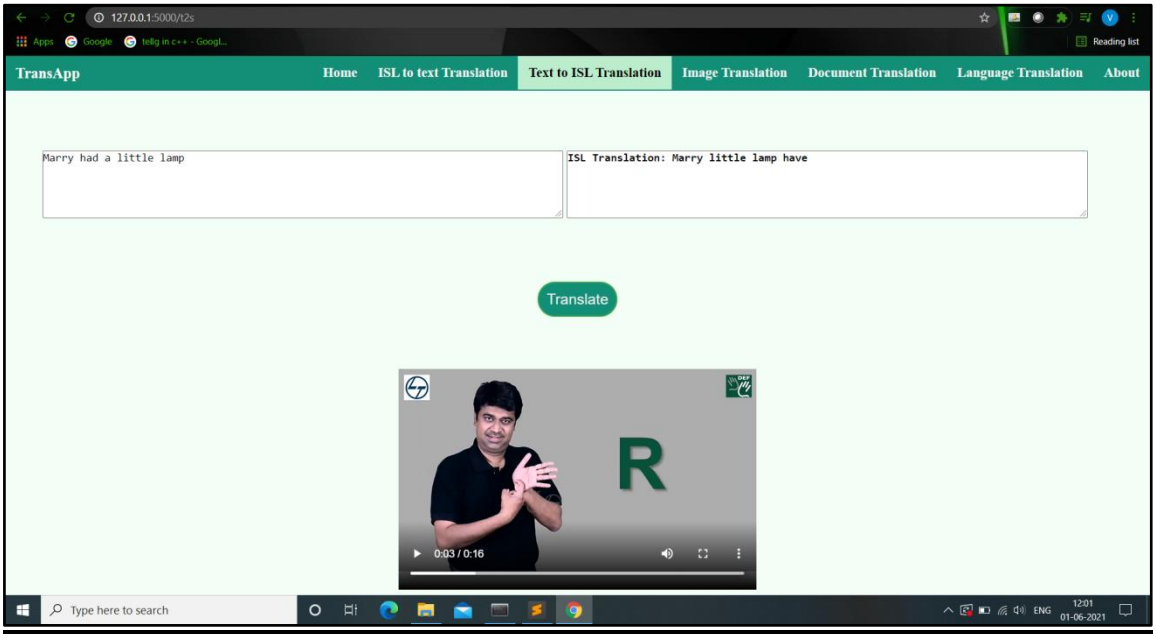

Fig 4.13: UI of English text to ISL gestures

4.5: Sign Language to Text Conversion Module

This feature will help users sign the gestures in ISL and get the result as English words when the trained model in the backend has recognized it.

- When the user navigates to "ISL to Text Translation" option, the webcam integrated to the system will capture the live video feed of the user in front of it, enabling them to sign various words present in ISL (this depends on the dataset created and trained in the backend).
- Once the user has performed a gesture provided in our dataset, landmarks associated with the hands are drawn on the screen.
- For single hand, 21 landmarks are drawn and 42 for double hand. After this, the model file in the backend will recognize the gesture.
- The output gesture in English text will be displayed on the screen in the top left corner along with the accuracy in percentage the model has predicted.

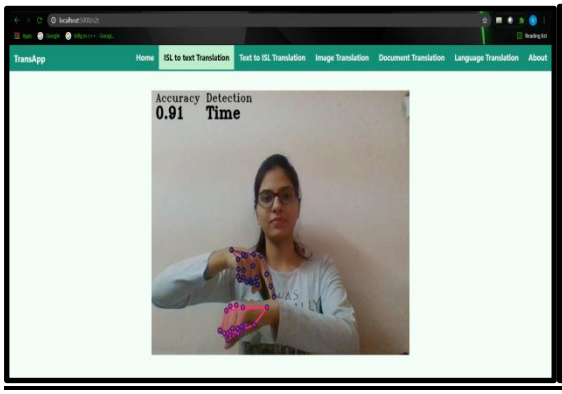

Fig 4.14: Detected Sign: "Time" Accuracy: 91%

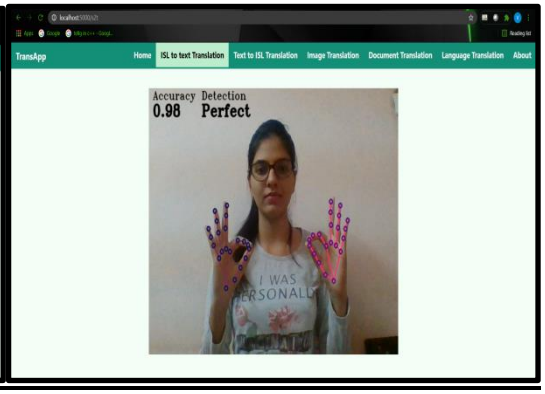

Fig 4.15: Detected Sign: "Perfect" Accuracy: 91%

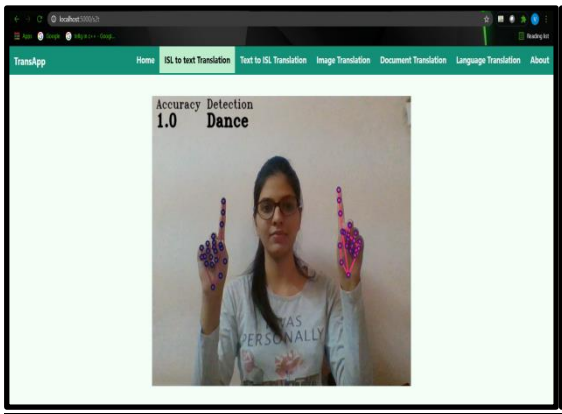

Fig 4.16: Detected Sign: "Dance" Accuracy: 100%

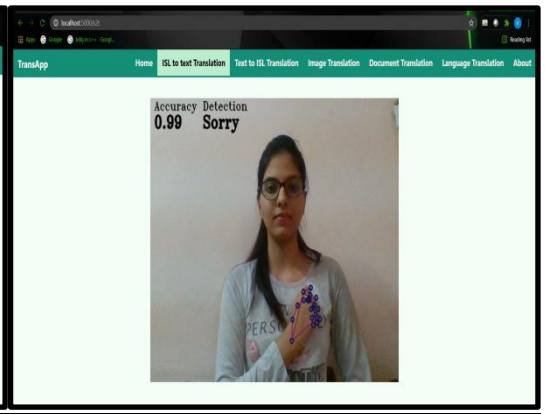

Fig 4.17: Detected Sign: "Sorry" Accuracy: 99%

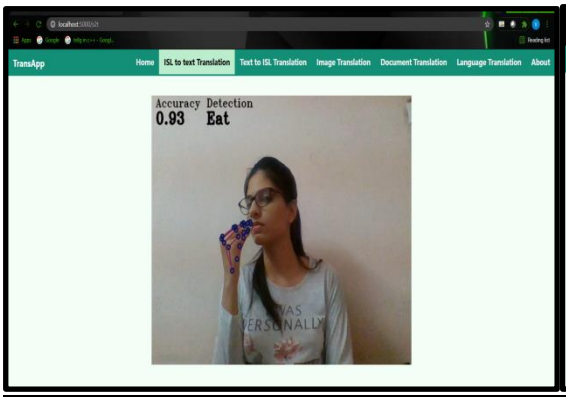

Fig 4.18: Detected Sign: “Eat” Accuracy: 93%

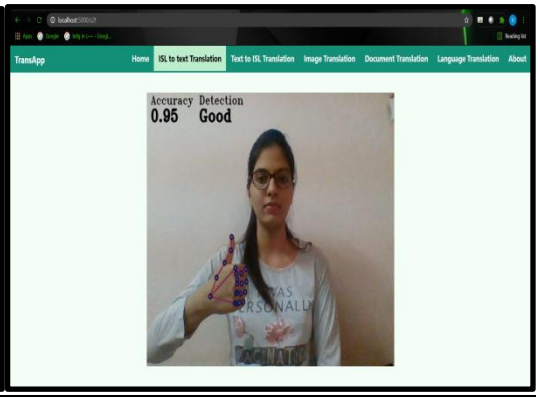

Fig 4.19: Detected Sign: “Good” Accuracy: 95%

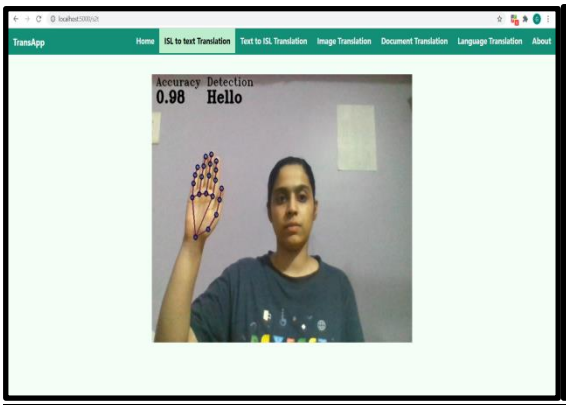

Fig 4.20: Detected Sign: “Hello” Accuracy: 98%

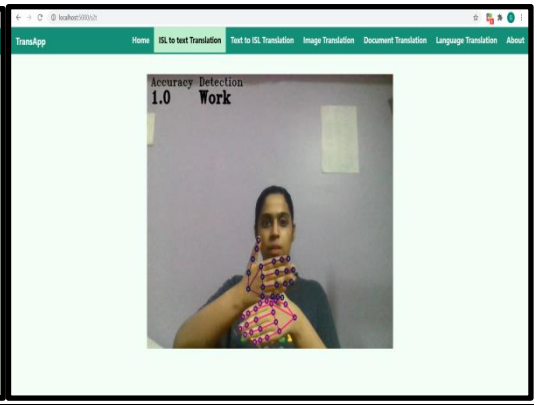

Fig 4.21: Detected Sign: “Work” Accuracy: 100%

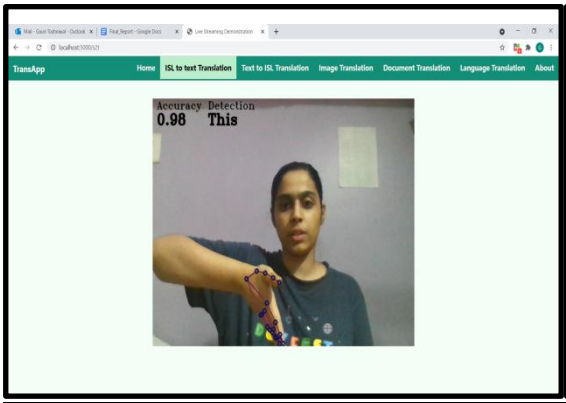

Fig 4.22: Detected Sign: “This” Accuracy: 98%

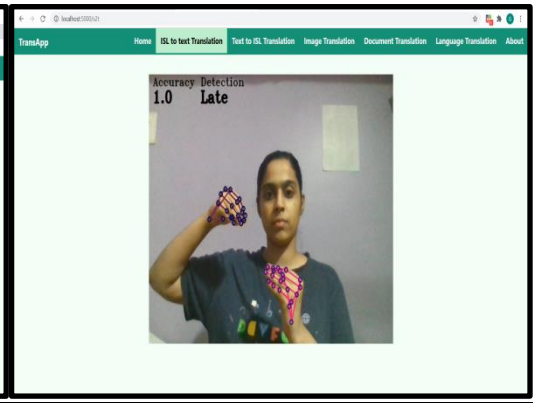

Fig 4.23: Detected Sign: “Late” Accuracy: 100%

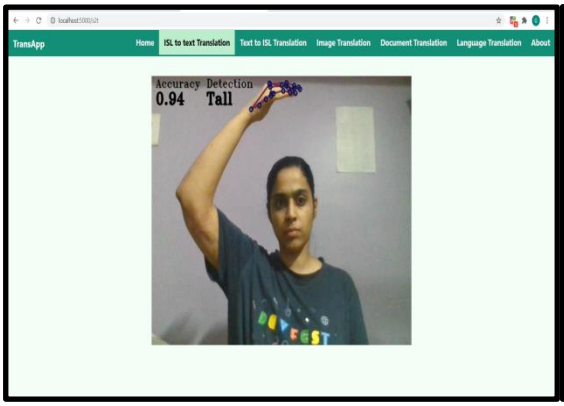

Fig 4.24: Detected Sign: “Tall” Accuracy: 94%

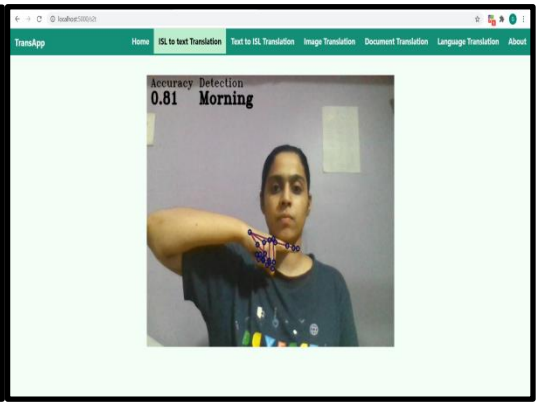

Fig 4.25: Detected Sign: “Morning” Accuracy: 81%

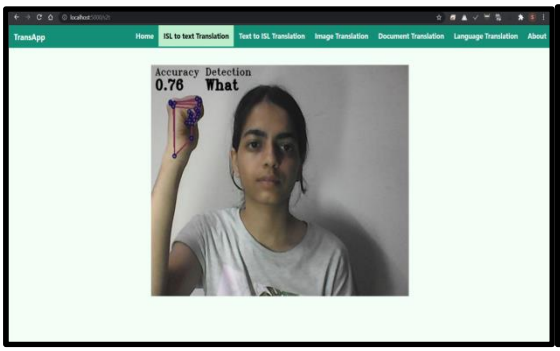

Fig 4.26: Detected Sign: “What” Accuracy: 76%

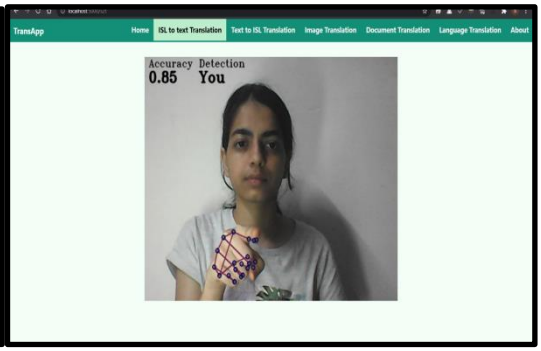

Fig 4.27: Detected Sign: “You” Accuracy: 85%

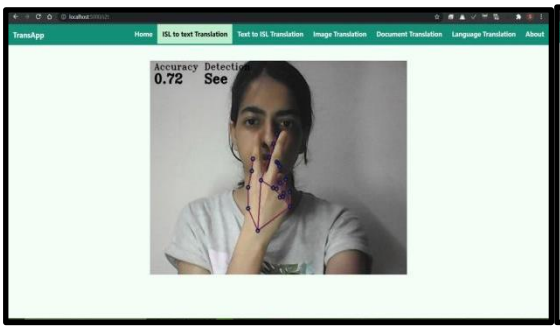

Fig 4.28: Detected Sign: “See” Accuracy: 72%

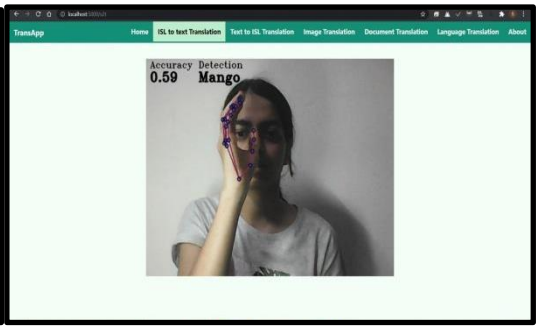

Fig 4.29: Detected Sign: “Mango” Accuracy: 59%

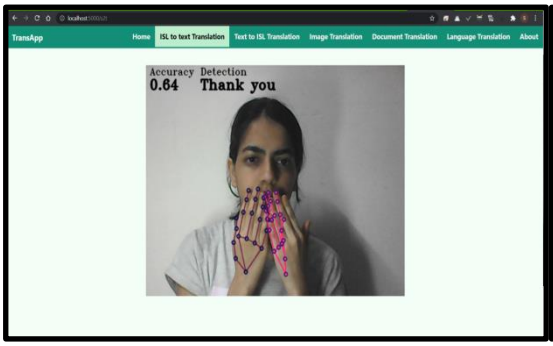

Fig 4.30: Detected Sign: “Thank You” Accuracy: 64%

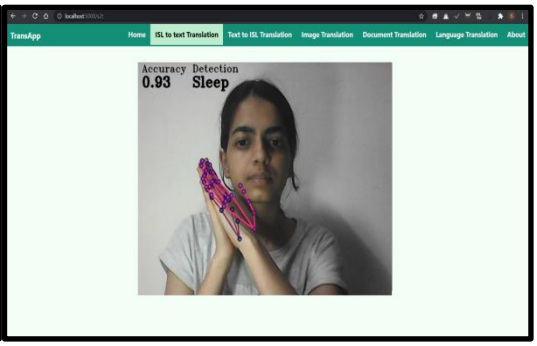

Fig 4.31: Detected Sign: “Sleep” Accuracy: 93%

Chapter 5
CONCLUSION AND SUMMARY

This work presents two approaches to Sign Language recognition, dataset creation and real-time implementation. Also, use of natural language processing to convert English text to Indian Sign Language and provide a one-stop solution by combining these modules with Language translation into a single digital interface (web application).

Two datasets have been created:

The first is a video dataset (649 videos) that involves 11 different categories of dynamic isolated signs of ISL with different light conditions and distinctly coloured hands which do not match any other object or the background in the frame. The 11 classes consist of 3 double hand signs and 8 single hand signs. Double hand sign classes include **Perfect**, **Thank You** and **Strong** while single hand involve **See**, **Good**, **Morning**, **You**, **Mango**, **Eat**, **Tall** and **Sorry**. 59 videos were recorded for every class 50 used for training and 9 for testing.

The second dataset includes recording hand landmarks of 18 different dynamic ISL Signs and the corresponding 3D hand landmarks into CSV files.Dataset included two separate CSV files. Single Handed (11 classes) - [rows(15,500), columns(85)], Double Handed (7 classes) - [rows(10,500), columns(169)]. In both CSV each class has 1500 records. In Single Handed CSV 11 classes include signs S**ee, Hello, Mango, Eat, Morning, Tall, Good, Sorry, What, This** and **You**. In Double Handed CSV 7 classes include signs **Perfect, Thank You, Dance, Sleep, Time, Late** and **Work.**

Two approaches to Detect Indian Sign Language:

Approach 1 - To train the model on spatial features, the inception model which is a deep convolutional neural network (CNN) has been used and recurrent neural network (RNN) has been used to train the model on temporal features. The proposed model was able to achieve an accuracy of 93.93% over the testing videos.

Approach 2 - Landmarks recorded were used as dependent variables and the corresponding Indian Sign Language word as Target variable hence reducing it to a classification problem, Random Forest Classifier Obtained accuracy of 99.96% on the test dataset with reasonably accurate results in real-time.

Translation system for English text to Indian sign language has also been presented. The major components of the system are conversion module(converts the English sentence to ISL sentence based on the grammatical rules), Elimination module(eliminates the unwanted words from the ISL sentence),Lemmatization module(converts each word of the ISL sentence to root word), and String matching module(converts the ISL sentence to video output).

These Modules i.e. Sign to Text Conversion and Text to Sign Conversion have been integrated with Language Translation Modules which can take input in text, image and document form to provide complete translation functionality addressing various barriers of communication.

REFERENCES

[1] I. Agarwal, S. Johar, and Dr. J. Santhosh, "A Tutor For The Hearing Impaired (Developed Using Automatic Gesture Recognition)," International Journal of Computer Science, Engineering and Applications (IJCSEA) vol.1, no.4, 2011.

[2] K. Deb, H. P. Mony, and S. Chowdhury, "Two-Handed Sign Language Recognition for Bangla Character Using Normalized Cross Correlation," Global Journel of Science and Technology, vol. 12, Issue 3, 2012.

[3] M. A. Mohandes," Recognition of Two-handed Arabic Signs using the CyberGlove," The Fourth International Conference on Advanced Engineering Computing and Applications in Sciences, 2010

[4] J. Rekha, J. Bhattacharya, and S. Majumder, "Shape, Texture and Local Movement Hand Gesture Features for Indian Sign Language Recognition," in 3rd International Conference on Trends in information science and computing, pp. 30-35, 2011.

[5] H.D. Yang, S. Sclaroff, and S.W. Lee, "Sign Language Spotting with a Threshold Model Based on Conditional Random Fields," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 31, pp. 1264-1277, 2009.

[6] O. Aran, I. Ari, L. Akarun, B. Sankur , A. Benoit , A. Caplier, P. Campr, A. H. Carrillo, and F. Xavier Fanard, "SignTutor: An Interactive System for Sign Language Tutoring," IEEE feature article, pp. 81-93, 2009.

[7] B. Lekhashri and A. ArunPratap,"Use of motion-print in sign language recognition," IEEE National Conference on Innovations in Emerging Technology (NCOIET), pp. 99-102, 2011.

[8] A. Nandy, J. S. Prasad, S. Mondal, P. Chakraborty, and G.C. Nandi, "Recognition of Isolated Indian Sign Language Gesture in Real Time," BAIP , Springer LNCS-CCIS, Vol. 70, pp. 102-107, 2010. References Dept. of CEA, GLAU 41

[9] I. N. Sandjaja, and N. Marcos, "Sign language number recognition," Fifth International Joint Conference on INC, IMS,IEEE, pp. 1503-1508, 2009. References Dept. of CEA, GLAU 42

[10] Y. Quan,"Chinese Sign Language Recognition Based On Video Sequence Appearance Modeling, "5th IEEE Conference on Industrial Electronics and Applications, pp. 1537- 1542, 2010.

[11] B. Bauer and H. Hienz, “Relevant features for video-based continuous sign language recognition,” In Automatic Face and Gesture Recognition, pages 440–445, 2000.

[12] J. Alon, V. Athitsos, Q. Yuan and S. Scalaroff, “A Unified Framework for Gesture Recognition and Spatiotemporal Gesture Segmentation,” in IEEE trans. on pattern analysis and machine intelligence, vol. 31, no. 9, 2009.

[13] T. Shanableh, K. Assaleh and M. Al-Rousan,” Spatio-Temporal Feature-Extraction Techniques for Isolated Gesture Recognition in Arabic Sign Language,” in IEEE trans. on systems, man, and cybernatics, vol. 37, no. 3, 2007.

[14] V.Pavlovic, R. Sharma, T. Huang, “Visual Interpretation of Hand Gestures forHuman – Computer interaction: A review”, in IEEE trans. on pattern analysis and machine intelligence, vol. 19, no. 7, 1997.

[15] M. K. Bhuyan, D. Ghosh, and P.K. Bora, “Threshold Finite State Machine for Vision based Gesture Recognition,” In Proceedings of the INDICON Annual IEEE Conference, 2005.

[16] S. Muller, S. Eickeler, and G. Rigoll, “Crane Gesture Recognition using Pseudo 3-d Hidden Markov Models,” In Proceedings of the International Conference on Automatic Face and Gesture Recognition, 2000.

[17] C.C. Chang and C. Y. Liu, ”Modified Curvature Scale Space Feature Alignment Approach for Hand Posture Recognition,” In Proceedings of the International Conference on Image Processing, 2003.

[18] K. Hoshino and T. Tanimoto, “Real time Estimation of Human Hand Posture for Robot Hand Control,” In Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, 2005.

[19] H. Jang, J. H. Do, J. Jung, K. H. Park, and Z. Z. Bien, “View-Invariant Hand Posture Recognition for Soft-Remocon-System,” In Proceedings of the International Conference on Intelligent Robots and Systems, 2004. References Dept. of CEA, GLAU 43

[20] A. Rocha, D. C. Hauagge, J. Wainer, and S. Goldenstein, “Automatic Fruit and Vegetables Classification from Images,” Computer and Electronics in Agriculture, vol. 70, pp. 96-104, 2010.

[21] T. Kurata, T. Okuma, M. Kourogi, and K. Sakaue, “The Handmouse: GMM HandColor Classification and Mean Shift Tracking,” In Proceedings of the IEEE

International Conference on Computer Vision - Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 2001.

[22] M. C. Cabral, C. H. Morimoto, and M. K. Zuffo, “On the usability of Gesture Interfaces in Virtual Reality Environments,” In Proceedings of the Latin American Conference on Human-Computer Interaction, 2005.

[23] H. Jang, J. H. Do, J. Jung, K. H. Park, and Z. Z. Bien, “View-invariant Hand Posture Recognition for Soft-Remocon-System,” In Proceedings of the International Conference on Intelligent Robots and Systems, 2004.

[24] R. O. Hagan and A. Zelinsky, “Visual Gesture Interfaces for Virtual Environments,” In Proceedings of the 1st Australasian User Interface Conference, 2000.

[25] Y. Zhu and G. Xu, “A Real-time Approach to the Spotting, Representation, and Recognition of Hand Gestures for Human-Computer Interaction,” Computer Vision and Image Understanding, 2002. References Dept. of CEA, GLAU 44

[26] A. Cassidy, D. Hook, and A. Baliga, “Hand Tracking using Spatial Gesture Modeling and Visual Feedback for a Virtual DJ System,” In Proceedings of the 4th IEEE International Conference on Multimodal Interfaces, 2002.

[27] Y. Sribooruang, P. Kumhom, and K. Chamnongthai, “Hand Posture Classification using Wavelet Moment Invariant,” In Proceedings of the IEEE Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2004.

[28] M. K. Hu, “Visual Pattern Recognition by Moment Invariants,” IRE Transactions on Information Theory, vol. 8, pp. 179-187, 1962.

[29] Y. Hamada, N. Shimada, and Y. Shirai, “Hand Shape Estimation under Complex Backgrounds for Sign Language Recognition,” Proc. of 6th Int. Conf. on Automatic Face and Gesture Recognition, 2004, pp. 589 -594.

[30] C. W. Ng and S. Ranganath, “Real-Time Gesture Recognition System and Application,” Image Vision and Computing, 2002.

[31] T. Shanableh, and K. Assaleh, “Arabic Sign Language Recognition in User Independent Mode,” IEEE International Conference on Intelligent and Advanced Systems , 2007.

[32] M. B. Waldron, and S. Kim, “Isolated ASL Sign Recognition System for Deaf Persons,” IEEE Transactions of Rehabilitation Engineering, vol. 3, no. 3, September 1995.

[33] G. R. S. Murthy, and R. S. Jadon, “Hand Gesture Recognition using Neural Networks,” IEEE, 2010.

[34] G. Fang, W. Gao, and D. Zhao, “Large Vocabulary Sign Language Recognition based on Fuzzy Decision Trees,” IEEE Transactions on Systems, Man, and Cybernetics, part A, vol. 34, no. 3, pp. 305-314, 2004.

[35] G. Fang, W. Gao, and D. Zhao, “Large-Vocabulary Continuous Sign Language Recognition Based on Transition-Movement Models,” IEEE Transactions on Systems, Man and Cybernetics, part A, vol. 37, no. 1, pp. 1-9, 2007.

[36] S. C. Agrawal, A. S. Jalal and C. Bhatnagar, “Redundancy removal for isolated gesture in Indian sign language and recognition using multi-class support vector machine,” Int. J. Computational Vision and Robotics, Vol. 4, pp 23-38, Nos. 1/2, 2014.

[37] ISL dataset https://indiansignlanguage.org/ by RKMVERI, The Faculty of Disability Management and Special Education.

[38] Mediapipe open source pipeline framework https://mediapipe.dev/ developed by Google - CNX.

[39] Real-Time Sign Language Gesture (Word) Recognition from Video Sequences using CNN and RNN developed by Sarfaraz Masood, Adhyan Srivastava, Harish Chandra Thuwal and Musheer Ahmad https://www.springerprofessional.de/en/real-time-sign-language-gesture-word-recognition-from-video-sequ/15606280

[40] Flask Webapp Framework https://opensource.com/article/18/4/flask by Nicholas Hunt-Walker, 02, Apr, 2018.

[41]Classifiers in supervised machine learning, https://builtin.com/data-science/supervised-machine-learning-classification

by Badreesh Shetty, August 5, 2020.

[42] Masood, S., Thuwal, H.C., Srivastava, A.: American sign language character recognition using convolutional neural networks. In: Proceedings of Smart Computing and Informatics, pp. 403–412. Springer, Singapore (2018).