Enhancing Bidirectional Sign Language Communication: Integrating YOLOv8 and NLP for Real-Time Gesture Recognition

by

Mubtasim Fuad Mozumder
20101463
Hasnat Jamil Bhuiyan
20101411
Ananna Majumder
20101183
Aneesha Sayara Chowdhury
20101029
Md Nafiz Mahfuz
20301365

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
January 2024

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Mubtasim Fuad Mozumder
20101463

Hasnat Jamil Bhuiyan
20101411

Ananna Majumder
20101183

Aneesha Sayara Chowdhury
20101029

Md Nafiz Mahafuz
20301365

Approval

The thesis titled "Bidirectional 3D Sign Language Conversion" submitted by

1. Mubtasim Fuad Mozumder (220101463)
2. Hasnat Jamil Bhuiyan (20101411)
3. Ananna Majumder (20101183)
4. Aneesha Sayara Chowdhury (20101029)
5. Md Nafiz Mahfuz (20301365)

of Fall, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 18, 2024.

Examining Committee:

Supervisor: (Member)

Nabuat Zaman Nahim
Lecturer
Department of Computer Science and Engineering
Brac University

Co-Supervisor: (Member)

Md. Sabbir Ahmed
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator: (Member)

Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department: (Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Sign language is a visual language expressed through physical movements instead of spoken words. Hands, eyes, facial emotions, and movement are all used as visual clues in this language. Although many hearing people also use sign language, it is predominantly utilized by those who are deaf or hard of hearing. The grammar and structural norms of sign language have developed over time, much like those of any spoken language. Most often we see that normal people who don't know sign language, struggle while communicating with people having hearing issues. It creates problems for the people to communicate and share their issues in case of emergency. The aim of this research is to take American Sign Language (ASL) data through real time camera footage and be able to convert the data and information into text. Moreover,the model can also convert text into sign language and show it to people who understand sign language, which can help us break the language barrier for the people who are in need. For recognising American Sign Language (ASL), we will be using the You Only Look Once(YOLO) model and Convolutional Neural Network (CNN) model. YOLO model is run in real time and automatically extracts discriminative spatial-temporal characteristics from the raw video stream without the need for any prior knowledge, eliminating design flaws.For converting text input to sign language, we will make a framework that will take a sentence as input, identify keywords from that sentence and then show a video where sign language is performed with respect to the sentence given as input. This framework will also function in real time.

Keywords: Sign Language; Convolutional Neural Network; You Only Look Once; American Sign Language

Acknowledgement

To begin with, we would like to thank The Almighty for blessing us with good health so that we could finish our thesis work in due time. Secondly, we would like to express our gratitude to our supervisor Nabuat Zaman Zahim and co-supervisor Md. Sabbir Ahmed for their constant guidance throughout our entire thesis work. It would have been very difficult to complete our thesis work without their guidance. Lastly, we would like to thank our families and friends for their constant support.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

CNN Convolutional Neural Network

HMM Hidden Markov Model

SVM Support Vector Machine

WHO World Health Organization

$YOLO$ You Only Look Once

Chapter 1

Introduction

When we think about communication, the first thing we think of is words. To an average human being, communication is simply the ability to convey and comprehend emotions and thoughts through the use of words, or, in short, spoken language. However, communication extends beyond spoken language. We subconsciously use facial expressions, gestures, and body language to communicate. One such language is sign language, which exists to break the barrier of communication between the hearing impaired and hearing people. Sign language is a form of visual language that makes use of 3D space, hand movements, and sometimes other parts of the body to convey meaning. It allows deaf or hard of hearing people to participate in conversations with equal status. Even though, like any other language, sign language has its own grammatical rules and linguistic structures, among the hundreds and thousands of languages in the world, sign language still struggles to be recognized as a commonly used language. According to the World Health Organization (WHO), over 1.5 billion people in the world live with hearing loss [53]. Despite such a huge chunk of the global population suffering from the issue, only a minority of people know how to effectively communicate with them due to a lack of awareness about sign language. This is where the importance of our research on sign language emerges, as our work provides a service to both the deaf and hearing communities, as the deaf people can express themselves without any barrier, and the hearing people can also interact with them without knowing sign language at first hand. Although there are more than 300 sign languages present in different regions of the world,for instance, British sign language in the UK, German sign language in Germany, Indian sign language in India and so on.[48] Since English is the language mostly used throughout the world, our work will be based on American Sign Language (ASL).

1.1 Research Problem

When communicating with individuals who are differently abled, specifically those who use sign language as a means of communication, there can be various challenges that arise for people who are not familiar with this form of communication. Here are some common issues that individuals may face when communicating with individuals who use sign language:

- Lack of knowledge or understanding: The majority of people have very little to no knowledge of sign language's structure. It could be difficult for them to

effectively communicate with persons that utilize sign language as a primary mode of communication owing to their lack of knowledge.

- Limited accessibility: Not every environment, public area, or communication medium is made to support those who use sign language. This inaccessibility can hinder effective communication and make it challenging for both parties to understand one another.

- Limited vocabulary: Important elements of sign language include precise hand gestures, facial expressions, and body language. Individuals who are not acquainted with this may misunderstand or miscommunicate as a result of misinterpreting the motions.

- Communication pace: Compared to spoken language, sign language conversations usually proceed more swiftly. As a result, sign language interactions usually proceed more swiftly than spoken language conversations. Those unfamiliar with sign language may find this overwhelming as it may be difficult to understand the rapid hand motions and facial expressions.

- Lack of confidence: Some people may feel insecure or self-conscious when trying to communicate with persons who use sign language. This feeling can be brought on by the fear of making mistakes, misinterpreting something, or feeling embarrassed, all of which can further impede effective communication.

- Limited availability of interpreters: In situations when communication support is needed, like in public events or professional settings, there may be an absence of qualified sign language interpreters. This could make it more difficult for those who use sign language to communicate effectively with others.

Effective communication is essential for everyone's safety and wellbeing during emergencies or natural disasters, including sign language users. There have been cases, nevertheless, where a dearth of competent sign language interpreters has made it extremely difficult for those who are deaf to obtain crucial information and emergency assistance. Several international tragedies, including the earthquake in Haiti in 2010 and the Hurricane Katrina disaster in the United States in 2005, brought this issue to light.. These events shed light on the need for im- proved emergency response systems that prioritize the communication needs of all individuals, including those who rely on sign language. In 2019, a significant step was taken towards recognizing sign languages at the international level. The United Nations (UN) General Assembly adopted a resolution calling for the inclusion of sign languages in all aspects of society, including education and public services. The resolution acknowledged that sign languages are fully-fledged languages with their own grammatical structures and cultural identities.

The main focus of this research is:
Whenever someone is in need or has to use sign language to communicate with others, our research will help both sides communicate smoothly and clear the barrier between sign language users and normal people.

1.2 Research Objective

- To create a comprehensive system for real-time conversion of American Sign Language (ASL) data acquired by camera footage into text, allowing deaf or hard of hearing people to communicate effectively.
- To create a rigged animated 3D model that can represent sign language gestures and movements, allowing hearing people to understand and communicate with sign language users.
- To investigate and assess the usefulness of the YOLO model and CNN model for real time sign language detection.
- To assess the performance and accuracy of the suggested models and systems using real-world sign language data in rigorous testing and validation, taking into account aspects such as recognition accuracy, translation quality, speed, and robustness.
- Evaluate the usability and user experience of the produced system using user studies and feedback from both sign language users and hearing people, with the goal of identifying areas for improvement and prospective enhancements.
- To make recommendations for incorporating the developed system into various communication platforms, public areas, and emergency response systems, with the goal of improving accessibility and inclusivity for people who use sign language as their primary mode of communication.
- To assess the performance and accuracy of the suggested models and systems using real-world sign language data in rigorous testing and validation, taking into account aspects such as recognition accuracy, translation quality, speed, and robustness.
- Evaluate the usability and user experience of the produced system using user studies and feedback from both sign language users and hearing people, with the goal of identifying areas for improvement and prospective enhancements.
- To make recommendations for incorporating the developed system into various communication platforms, public areas, and emergency response systems, with the goal of improving accessibility and inclusivity for people who use sign language as their primary mode of communication.

Chapter 2

Literature Review

2.1 Text input to sign language

Being bidirectional, one of the parts of our approach is the conversion of text input to sign language. From one of the papers we read, we got to figure out that neural network models are popular for this approach. RNN and CNN can be used for the translation part as they can incorporate multiple components of sign language translation. RNN helps to understand the sequence and CNN analyzes the part where visual is required. The model gets trained by using datasets where text and its corresponding ASL are in pairs. In this way the translation to sign language happens. From another paper, we found out an interesting approach to translate sentences in English to Indian sign language. For conversion of the sentence to the specific sign language mentioned, Lexical Functional Grammar has been used. It is a type of linguistic framework by which we can inspect the structure of natural language. Here, syntax and semantics are separated and it has categories like lexical and functional. This framework is very versatile which helps in different types of applications including natural language processing and also machine translation. So, with the help of Lexical Functional Grammar, they implemented the system.

2.2 Sign language to text

We saw a proposal of using Hidden Markov Model(HMM) to classify trajectory, orientation and resultant shape of sign language. There has been use of 3D translation and rotation of data using Ascension Technologies Flock of Birds devices [3]. While using 3D translation and rotation data and real time captured test data, it is possible to avoid the design flaws in our current research. However, use of the pre-trained CNN - VGG16 model for American Sign Language (ASL) character recognition has gained traction. VGG16 is developed by Vision Geometry Group from Oxford and is a vision model that uses Convolution Neural Network and the accuracy achieved is about 96 percent. In a research the authors have used Kinect Camera to obtain hand gestures features. After feature extraction, the authors have done training and testing of the gestures using multiclass SVM [22]. Use of technique which recognises hand gestures which was grounded in shape analysis. We saw neural networks classifying six static hand actions which have shown an accuracy of 86.38 percent [26].

Adding to that, we noticed the use of Microsoft Kinect and CNN accelerated by GPU(graphic processing unit) for gesture recognition as well in a paper. It has shown an accuracy of 92 percent and it was even more remarkably high for 20 actions [24]. For converting hand gestures to text, OpenCV and CNN were used to build an automated system. An ASL dataset will be brought to train this system [26]. Adding to that, use of weakly supervised training methods on video data of sign language gives meaningful sentences from gestures. For a stream of data, multiple streams of HMMs were applied in another paper. Each HMM stream will have CNN-LSTM models [8]. We also saw a paper use the Hierarchical Grassman Covariance Matrix (HGCM) for sign language recognition and representation. The model was tested on actual uninterrupted sign datasets and a motion based database(HDMO5) as well.

2.3 Related Works

Now let us discuss some of the research papers that inspired us to start working on our thesis topic in depth. We will be discussing the topic in two parts, which are mentioned as follows:

2.3.1 Text input to sign language

Neural network models such as RNN, CNN are well-known approaches of how to accomplish this feat when it comes to sign language translation. For example- in [27] the main deep learning architectures used by the author for the translation task are recurrent neural networks (RNNs) and convolutional neural networks (CNNs). These architectures are able to simulate both the linguistic and visual components of sign language translation. The model takes English text sentences as input and encodes them into numerical representations. Simultaneously, visual features are extracted from the corresponding ASL representations. RNNs capture the sequential nature of language, processing the text input and generating representations for each word in the sequence. CNNs analyze the visual input, extracting spatial patterns and relevant visual features.

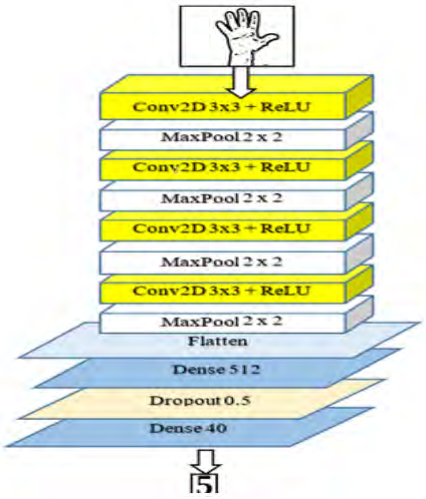

Figure 2.1: CNN Architecture [27]

The linguistic and visual representations are fused and attention mechanisms may be employed to weigh the importance of different words or visual features. Finally, the fused representations are decoded to generate sign language translations. The model is trained using a large dataset of text-to-ASL pairs, optimizing its parameters through back propagation and gradient descent. This proposed model's accuracy is 96.68 percent and it can recognize 40 motions in under 0.50 seconds.

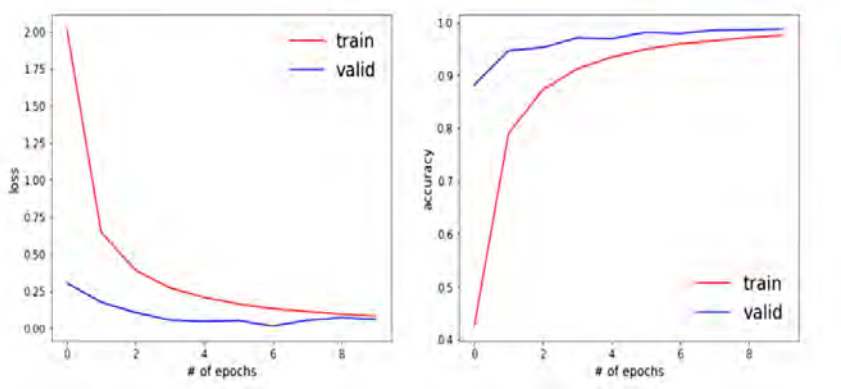

Figure 2.2: Loss & accuracy graph for the suggested model as a function of epochs [27]

In [5], the system proposed by the author has a unique approach to translating English sentences to Indian Sign Language (ISL). Their proposed system takes a text input and converts it to ISL with the help of Lexical Functional Grammar (LFG). The output is represented by using videos that were previously recorded. By using this system, they were able to achieve 89.4 percent accuracy on an evaluation based on 208 sentences where the majority of the errors occurred for compound sentences and sentences containing directional verbs.

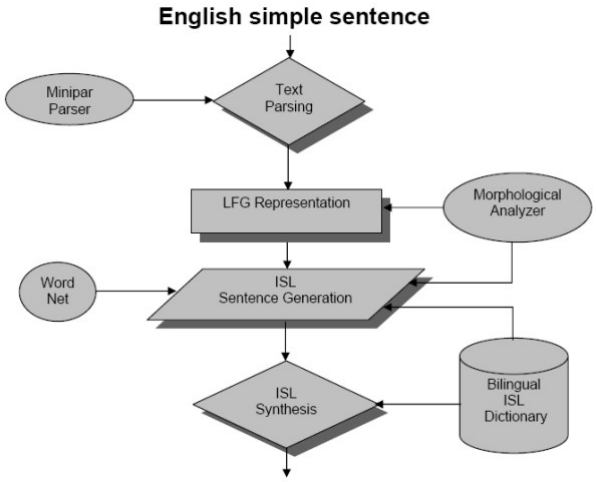

Figure 2.3: Architecture of the Text-to-ISL system [5]

	No. of Sentences	**Accuracy %**
Overall Corpus Size	208	89.4
Sentences without directional verbs	193	96.37
Sentences without compound constructions	201	92.53

Table 2.1: Evaluation Results [5]

In [10], we see an approach to transform Malayalam text to Indian Sign Language using animation. The input taken is a Malayalam word or some words and brings out the output in an animated style. Their system uses the Hamburg Notation System shortly known as HamNoSys for representing signs. It has a font set which has a good number of characters. The system has two parts here. The first part is the sign editor a database needs to be made. The database will contain HamNoSys notations of Malayalam words made by a sign language expert and it will be well defined. The second part is the translator which will convert it to ISL. The words in the database are mapped to strings of the notation system which is in turn mapped to specific tokens. By using that token, a SIGML file is generated. This file will help the avatar in the animation to generate the sign language. The final animated output is obtained by using JA SIGML URL app. Their system contained 100 words and out of those, 82 words which were converted to sign were correct and the rest needed some minor corrections.

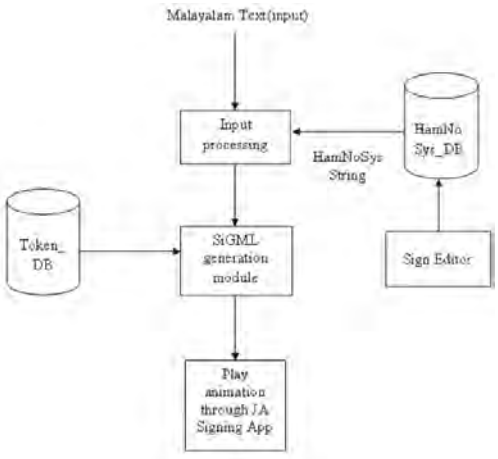

Figure 2.4: Architecture of the system [10]

In [6], we see an approach for converting Greek text to Greek sign language. The idea is based on teaching GSL (Greek Sign Language) and it is applied to a single

word only instead of sentences. Their work is grounded on a sign language database where coded knowledge of GSL can be found. For the translation part, they used Vsigns which is a web tool used for the synthesis of virtual signs. There is mention of another sign synthesis tool called SignSynth which uses Web3D and Perl. They displayed the output using animation where an animated character performed the sign.

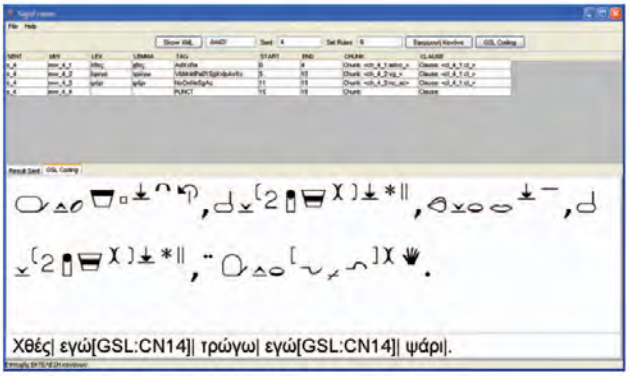

Figure 2.5: Coded text of Greek Sign Language grammar [6]

In [16], we see the making of a system where text in English language is taken as input and then it is first translated to HamNoSys representation. This is afterward converted into SiGML. SiGML is a markup language. The XML tags of HamNoSys are contained in it. A mapping system is used to link the text to the HamNoSys notation. For converting to SiGML, a database of the alphabets of HamNoSys is kept. From there, the xml tags are created. By conducting experiments, the authors claim that their system has an accuracy of 87 percent. This work may not be a direct example of text-to-sign language conversion which we expect. However, this provides us with insights into converting text to a signed notation system.

In [9], we found a proposal for a system where text will be represented in Indian Sign Language (ISL) through HamNoSys generation and conversion to SiGML. In their work, the HamNoSys was made for a word and then it was entered into the system. The HamNoSys was then coordinated with its corresponding SiGML tag from the database. Afterwards, a SiGML file was made from this, and then through a SiGML app the sign of the word was displayed using an avatar. The authors claim that they have tested their system on 250 words and they want to extend their work by making a large database and include more movement of hands and expressions.

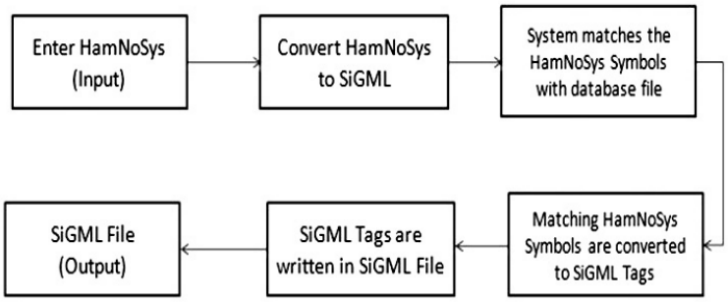

Figure 2.6: Workflow of the system [9]

The primary focus of the proposed model in [28] is the automatic conversion of Punjabi text to Indian Sign Language (ISL). The translation module fetches the corresponding HamNoSys notation for the input word from the database which is then converted into SiGML markup format. The output from the model is given in two formats: 3D animation of a human avatar and natural video recordings of sign language experts.

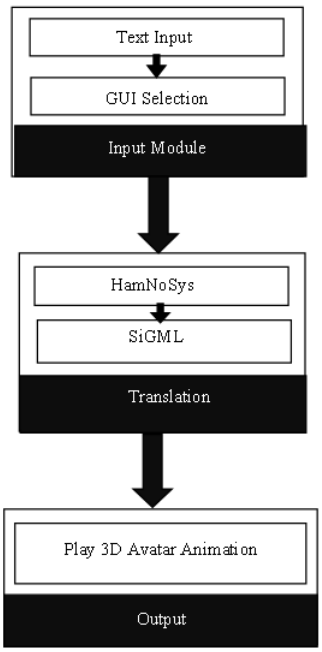

Figure 2.7: Illustration of the translation system [28]

In [12], the authors worked on translating Thai text into Thai Sign Language (TSL) is called "Intelligent Thai Text – Thai Sign Translation for Language Learning" (IT3STL), the architecture takes into account changes in syntax and semantics and consists of five steps. The modular architecture of IT3STL incorporates distinct knowledge bases for effortless upkeep. Users can submit Thai sentences into the interface for TSL translation, and the result will provide word information and sign representations. Evaluation shows that the translation is accurate and that users are satisfied, particularly those who are deaf or have hearing impairments. As a useful and intuitive language learning tool for Thai and TSL, IT3STL exhibits promise.

In [19], the author proposed a model that takes both example based and rule-based Interlingua approaches to convert Arabic Text to Arabic Sign Language(ArSL). An example-based approach is applied if the sentence already exists in the database, if not, a rule-based approach is applied. Automatic natural language processing (ANLP) technologies are used in rule-based learning. The result is vowelized following stem-based stemming and root-based stemming. Afterwards, syntactic reordering is done. Finally, the text is converted into a sign sentence which is animated using gif images. In situations when the analyzer gives more than one result for an Arabic word, the last one is taken.

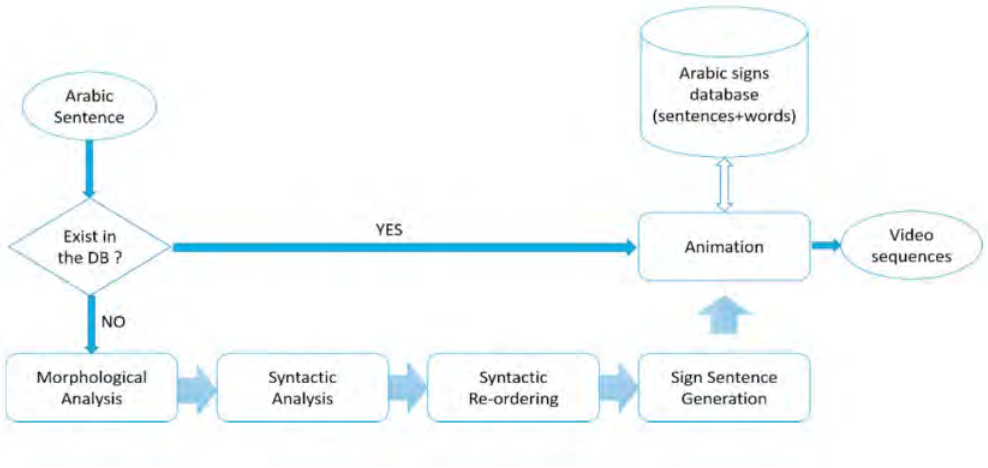

Figure 2.8: Architecture of English text to ISL Synthetic Animation System [19]

In [44], we see a text-to-sign language conversion system for Indian Sign Language (ISL) which takes into account the language's distinctive alphabet and syntax. This accepts input in alphabets or numerals and converts it into an ISL sign. The system uses preprocessing techniques for correct input handling, such as grayscale conversion, edge identification, and database comparison. Pattern recognition and matching parameters produced from input text or images are checked against a database of Marathi sign language elements. The output differs according to the input type: sign language input produces related text, while text input produces sign language.

In [18], a solution for speech-to-sign language conversion is proposed that uses Google API and Natural Language Processing. Speech acquisition, conversion to text with the Google Speech-to-Text API, NLP-based text processing, and matching with a visual sign word library are all part of the systematic four-stage process. The algorithm blends matched videos and displays the entire text in sign language. The system outperforms other approaches in terms of accuracy.

In [4], a prototype Arabic Text to Arabic Sign Language Translation System for the Deaf is presented in this paper. The system, which was developed in collaboration with deaf signers, makes use of an instructional ArSL corpus and an example-based methodology. Errors in earlier systems are a result of misunderstandings regarding ArSL's reliance on Arabic. Leave-one-out cross-validation evaluation results in a word error rate of 46.7 percent and a position error rate of 29.4 percent. While

highlighting the need for a larger corpus, the system has promise. Being the only system of its kind now in use for translating Arabic text to ArSL, it stands out for its potential for advancement and its distinctive contributions to this specialized sector.

In [12], the authors explore Arabic Sign Language (ArSL) recognition using image-based techniques, evaluating various visual descriptors. The proposed system employs the Histograms of Oriented Gradients (HOG) descriptor and a one versus-all soft-margin Support Vector Machine (SVM), achieving a 63.5 percent accuracy for Arabic alphabet gestures. Per-class performance varies, and future work involves investigating kernel SVM for further enhancement and assigning relevant weights to features. The study highlights challenges in ArSL recognition due to similar gestures and contributes to the development of effective image-based systems for communication by hearing and speech-impaired individuals.

2.3.2 Sign language Input to Text

In [43], we see the authors attempt to recognize the English alphabet and gestures in sign language and produce the accurate text version of the sign language used. The authors used CNN and computer vision to achieve their goals. Their model focuses on capturing the unique gestures in the picture so that the desired letter can be identified. They have used different datasets for training and testing. Here are some pictures of the training data. We can see that they have used several images to train a single alphabet. It is done so that the model gets well trained and can identify the gesture of an alphabet even if the user is not very sharp in using sign language. So, the overall work done by them is excellent. However, if they can recognize facial expressions as well then it will be better as then the output produced will be sentences instead of letters.

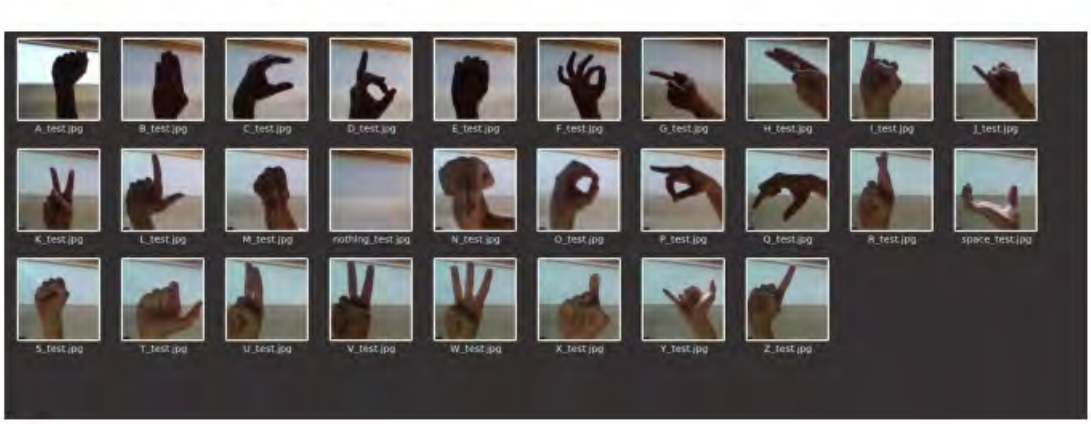

Figure 2.9: Sample pictures of training data [43]

Figure 2.10: Sample pictures of Training data for letter A [43]

In [31], we can see that the author wanted to clear out the environmental barrier that was hindering the accuracy and results in different research. He was working on computer vision-based hand gesture recognition. In the research, the main objective was to remove the background issue and they used bidirectional communication through image showing, which was kind of unique for the time and we were highly motivated while searching through the paper. They mainly used the Hidden Markov Model, Convolutional Neural Network(CNN), CNN VGG-16 method, and Python.

Figure 2.11: This snapshot shows the lettering inserted and the hand gesture picture given out by the model which was pre-inserted. [31]

In [41], the researchers were keen on making a real-time sign language conversion model. They mainly used ASL databases, advanced sign language grammar, CNN VGG-16, and Gated Recurrent Unit (GRU). Using GRU, they changed the frame rates from 20 to 5 and the method took out the frames needed for the CNN model to clarify and detect the grammars and figure out the actual sentence that the user was trying to say. As they reduced the frame rates, it was much easier to detect and was faster than other competing research. They also used notations from the advanced ASL database which made the work much simpler and easier to detect. Moreover, they also used stick posed estimation which was possible due to the video capture and multi-image process rendering. While this technology is still expensive to use, it can be done using computer vision. It marks the points and makes a low

poly image of a human figure. This makes the computation and detection more accurate.

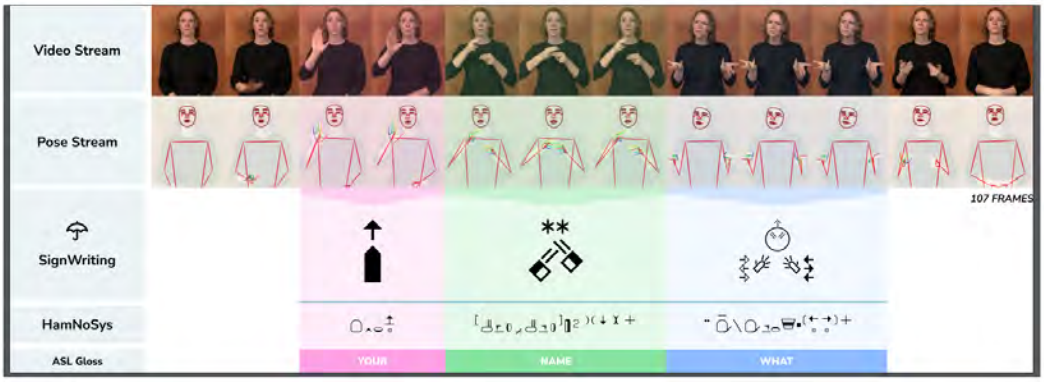

Figure 2.12: While there is a video running the model detects the images with the proper hand gestures and gives the output below. [41]

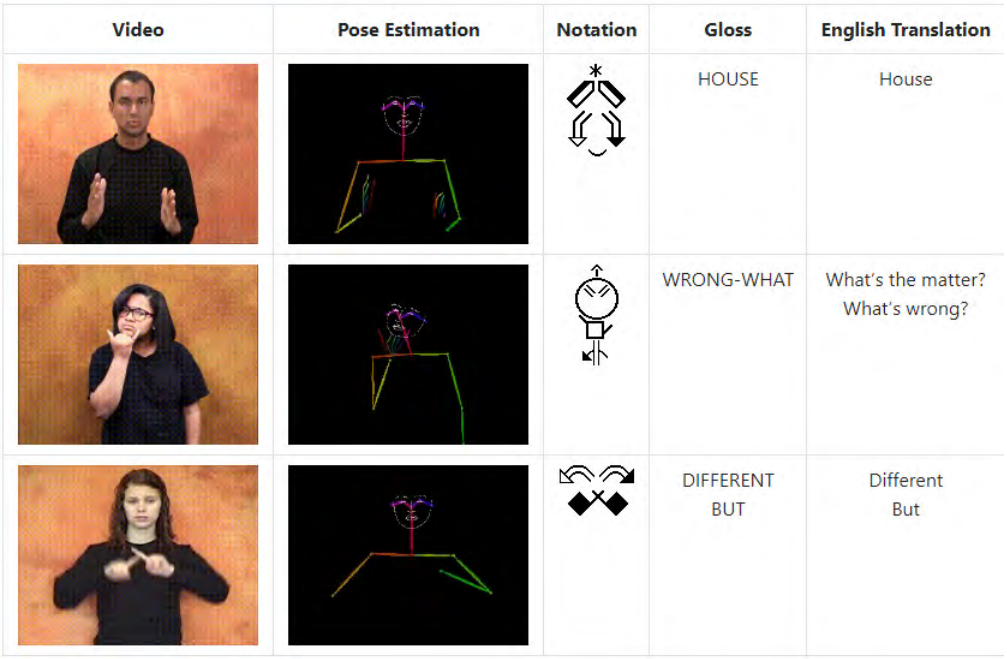

Figure 2.13: While the video is running there is a stick pose estimation which directly indicates the notations and makes the detection faster [41]

In [7], the researchers worked on reviewing multiple works on the recognition of Indian Sign Language (ISL). One such review caught our attention. The researchers worked on dual hand gesture recognition. Their datasets contained a plethora of images of 26 alphabets for ISL for dual hand gestures. They did feature extraction from the images using the Histogram of Orientation Gradient(HOG) and the Histogram of Edge Frequency(HOEF). Then the gestures were classified using a Support Vector Machine (SVM). By using HOEF for feature extraction, they achieved an accuracy of 98.1 percent. Adding to that, from their research they claimed that HOEF is better than HOG.

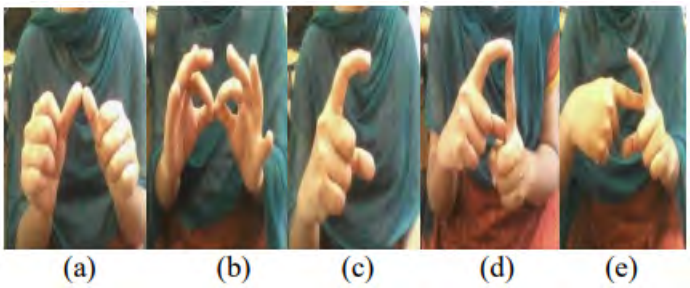

Figure 2.14: ISL signs for different letters [41]

In [35], we noticed that Indonesian sign language recognition was done using the YOLO method. They used a YOLOv3 pre-trained model making adjustments to the number of classes as they felt were needed. They used both image data and also video data. The system's performance was incredibly high during using image data and it was comparatively low while using video data.

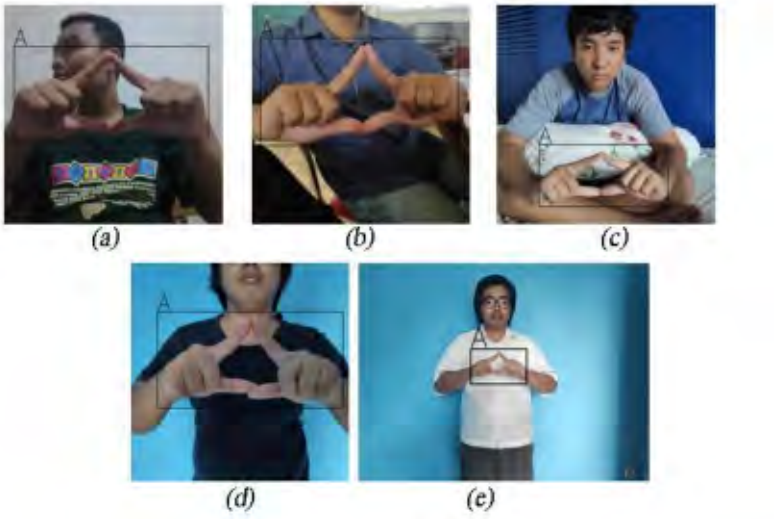

Figure 2.15: Indonesian sign language signs for letters or words [35]

In [17], Malaysian sign language recognition system was developed based on convolutional neural network and YOLOv3 model. The system was made for real-time analysis. They worked on alphabet here mainly and tested their system on the Darknet framework after doing preprocessing. However, the results of the testing did not show a very high level of accuracy in the detection. They plan to work on their sign language detection on a deeper level and larger scale to improve the performance of their system.

In [37], we see the researchers working on making an Italian sign language recognition system that identifies letters of the Italian alphabet in real-time. For their system, they used two models, CNN and VGG-19. As their system was based on the interaction between humans and computers, they used OpenCV which is a Python library along with the two models. After running experiments on both models, they

found that the pre-trained VGG-19 model performed better than their ordinary CNN model. The VGG-19 model gave an accuracy of 99 percent and the CNN model was 97 percent accurate.

In [2], we found the researchers worked on a system that is used to recognize Korean sign language. The system identifies the sign language performed and converts it into text form in the Korean language. For understanding the movement of hands and fingers and identifying it, data gloves came into use here as a sensing instrument. To make their system efficient, they used a fuzzy min-max neural network and also suggested a method for effective categorization of motions.

In [36], we see sign language detection work done on Arabic sign language. The researchers applied deep learning to their research work. Their dataset consists of static images where each image represents a letter in Arabic. They selected some of the best deep learning architectures by searching related works and conducting some tests. Based on the test results, they selected AlexNet as their main architecture and made their model based on it which resulted in a very high accuracy rate. They examined the model in real time and the accuracy rate was well over 90 percent.

In [42], the authors worked on a real-time American Sign Language (ASL) to text conversion system with convolutional Neural Network (CNN) and Artificial Neural Network. Resulting in a notable accuracy for recognizing ASL alphabets, the vision-based approach uses OpenCV for dataset creation and Gaussian haze for highlight extraction. Overcoming the challenges of filter selection outperforms existing research without specialized devices by the two-layer algorithm. The project eliminates the need for interpreters as well as offers potential applications to other sign languages. Future examination plans to upgrade precision utilizing foundation deduction, consolidate discourse acknowledgment, and perceive extra letter sets, with the chance of extension as a web or portable application.

In [32], the authors have developed an Android app that can convert real-time ASL input to speech or text. Their dataset consists of a total of 27 classes, 26 for the English alphabet and 1 class for "Space". SVM is used to train the proposed model based on three array parameters: direction method, kernel, and dimensionality reduction type which are then compared to find the parameter values that give the maximum accuracy and minimum average processing time.

Sl. No	Different combination of methods in SVM				
	Detection Method	*Kernel*	*Dimensionality Reduction*	*Average per Image processing time (MilliSecond)*	*Accuracy*
1	Contour Masking	Linear	None	18.2	97.45
2	Contour Masking	Linear	PCA	18.0	97.98
3	Contour Masking	RBF	None	18.4	98.12
4	Contour Masking	RBF	PCA	17.8	98.34
5	Skeleton	Linear	None	17.9	98.22
6	Skeleton	Linear	PCA	17.7	98.25
7	Skeleton	RBF	None	18.4	98.56
8	Skeleton	RBF	PCA	18.0	98.89
9	Canny Edges	Linear	None	18.1	98.52
10	Canny Edges	Linear	PCA	17.5	98.67
11	Canny Edges	RBF	None	18.3	98.74
12	Canny Edges	RBF	PCA	15.0	98.82

Figure 2.16: Performance comparison of all array parameters in SVM [32]

Canny edges, RBF, and PCA were used in the SVM algorithm after the comparative analysis. Finally, testing is performed on 20 percent of the dataset using the parameter values stated above to compare and choose the optimal approach.

The model in [38] uses machine learning to translate Indian sign language (ISL) to text in real time. The authors propose the use of a computer webcam to take images of the hand signs. The images are then converted to black and white threshold maps from RGB images and further passed on to a neural network to be processed and classified between the 26 alphabets or digits from 0 to 9. The accuracy of the model is 97.3 percent for the test dataset.

In [29], emphasizing Bangla Sign Language (BdSL), the authors review various systems developed for sign language recognition, including those for American Sign Language (ASL) and Japanese Sign Language (JSL). The paper then focuses on existing BdSL-related research, providing a thorough examination of 15 selected studies. It divides research into two categories: one-way communication (from BL to BdSL or vice versa) and bi-directional communication. The suggested method, which incorporates both BL to BdSL and BdSL to BL conversions, aims for fluent communication with deaf individuals. Building a large database, creating grammatical rules, and implementing machine learning for rule-based conversion are all part of the technique. The proposed solution overcomes shortcomings in previous techniques and attempts to create a well-developed two-way communication system for BdSL users.

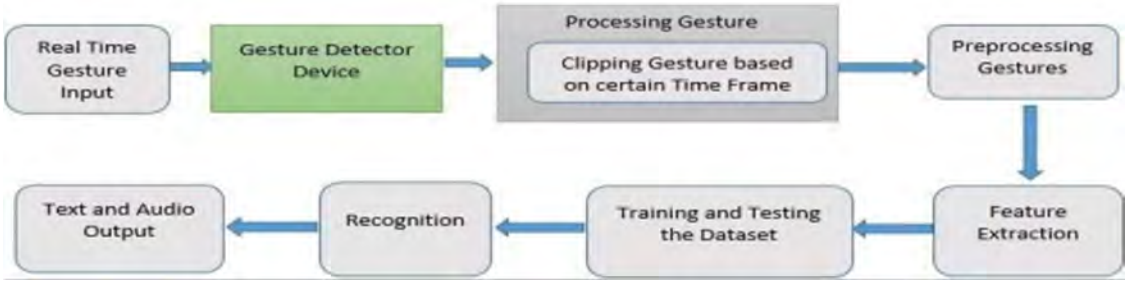

Figure 2.17: Conversion of BdSL to Bangla spoken language [29]

In [21], the authors developed a system that allows the creation of a double-handed gesture recognition module based on MEMS sensors. The system captures double-handed gestures using MEMS sensors. These gestures are subsequently processed by a hidden Markov model (HMM) recognition module. This module aims to recognize and interpret complex hand gestures, laying the foundation for the succeeding conversion. Once they have been detected, a sign-to-text conversion technique based on hidden Markov models is used. This stage translates the interpreted movements into bilingual text, bridging the gap between visual sign language and written language. The transformed text is subsequently processed by a text-to-speech synthesizer, which employs hidden Markov models once more. This synthesizer converts written words into synthetic voice, allowing the unimpaired listener to understand the impaired speaker's intended message. The suggested system performs admirably, obtaining 98 percent accuracy for single-handed motions and 97 percent accuracy for double-handed gestures. This precision is critical for maintaining effective and dependable communication between the two groups.

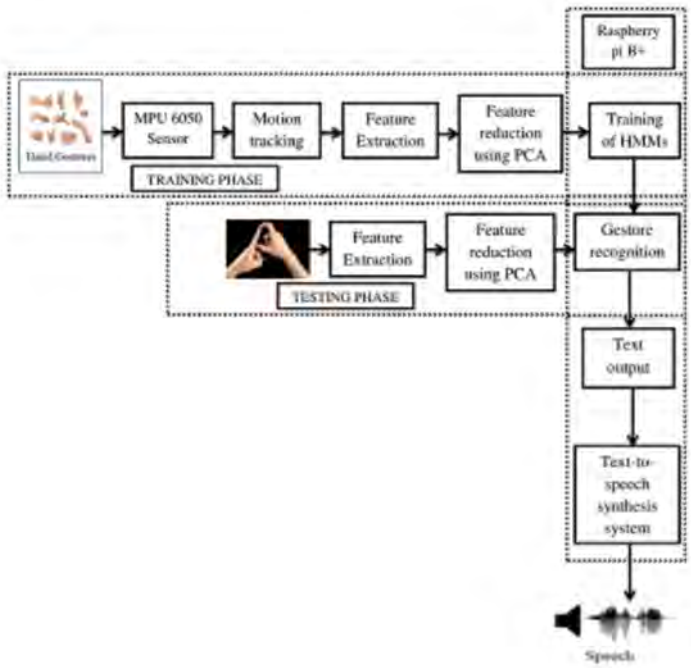

Figure 2.18: Gesture to speech conversion module [21]

In [39], Deep learning is used to create a revolutionary Indian Sign Language (ISL) detection and translation framework. To address ISL difficulties, it applies MediaPipe and a hybrid CNN model for accurate pose extraction. Google Text-to-Speech assists with audio creation in a variety of Indian languages. The system performs better than earlier techniques, achieving 94 percent accuracy after being tested and reviewed for convincing improvements. The proposal has the potential to improve communication and accessibility for those with hearing and speech difficulties.

In [15], smart gloves with sensors and a microprocessor are used to convert Bangla sign language into text. Flex sensors measure finger bending, On the other hand, accelerometers and gyroscopes record hand position and movement. The method processes and transmits data via Bluetooth to a PC, where it dynamically matches information with a database to build word soundtracks, with that are played on speakers. On-board tests show efficacy and evaluations with 8 individuals and 248 samples of 72 words show a system accuracy of 99.5 percent. In Bangla sentences, the user can change word pronunciation based on tense and pronoun variations.

In [13], there is the use of a Sign Language Recognition System (SLRS) in human-computer interaction, specifically translating solitary Arabic word signs into text. Hand segmentation, tracking, feature extraction, and classification are the four primary stages of this system. For hand segmentation, a dynamic skin detector based on face color tone is used, followed by a unique skin-blob tracking technique to recognize and track hands. The method is tested using a dataset of 30 isolated words relevant to hearing-impaired children's daily school life, with 83 percent of the words displaying varied occlusion statuses. On the other hand, in the signer-independent mode, the findings show a recognition rate of 97 percent. Furthermore, at = 5, the suggested occlusion resolution strategy outperforms previous methods by 2.57 percent in defining hand and head positions.

In [11], this paper focuses on subword recognition in Chinese Sign Language (CSL) using surface electromyography (sEMG), accelerometer (ACC), and gyroscope (GYRO) sensors. The study assesses the categorization capacities of these sensors, both alone and in combination, for three typical sign components: one-handed or two-handed signs, hand orientation, and hand amplitude. For CSL subword recognition, an efficient tree-structure classification framework is developed, to effectively fuse input from all three sensor types. The study included eight people, and recognition procedures were carried out under varied testing situations with a target set of 150 CSL subwords. The suggested framework outperformed the competition in seven different testing situations, including single-sensor, paired-sensor, and three-sensor fusion. The total recognition accuracies in user-specific and user-independent tests were 94.31 percent and 87.02 percent, respectively. The results pave the way for the development of a large-vocabulary sign language recognition system based on sEMG, ACC, and GYRO sensors.

The authors of [23] propose using a web camera to capture and process videos in real time without the need to store the data. The features required are extracted from videos of users showing sign languages and converted into static images. Following

this, the static images are processed using MATLAB and Simulink image processing algorithms. The sign is identified using the pixel values around the hands and head; the sign is finally transmitted to the Bluetooth receiver for the audio to be played.

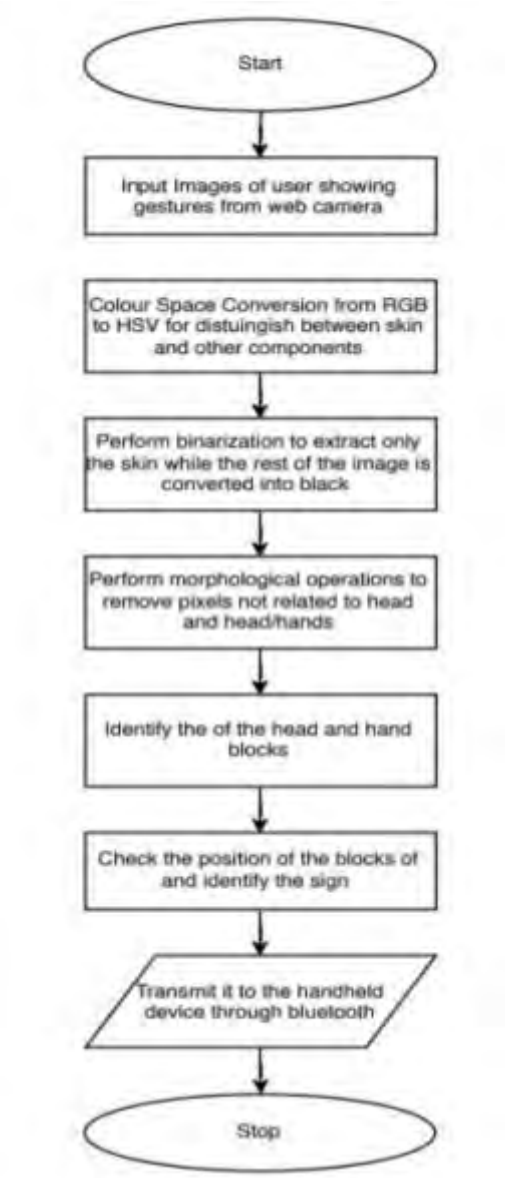

Figure 2.19: Flowchart for Image preprocessing and Feature Extraction [23]

In [25], a comprehensive system targeted at enabling communication for people with speech and hearing impairments has been highlighted. With a specific emphasis on sign language users. The device detects hand motions and converts them into human-audible messages using flex sensors accelerometer sensors and a Raspberry Pi controller. The device is for two-way communication by transforming human voice into hand gestures for people who have hearing or speech problems. In an emergency, the system automatically sends hand motion information to the user's contacts as text. The system tackles the communication issues that mute people face by providing an advanced solution that improves their safety and accessibility.

In [20], a system is proposed for the communication issues faced by deaf and dumb people. This system uses picture recognition and deep learning. It acts as a translator between sign language and text using the MediaPipe framework. The proposed system consists of three modules, image input, image processing, and classification. The system intends to provide an affordable and accessible solution for converting sign language into text. By capturing hand motions, processing photos, and classifying them, it promotes better communication for deaf individuals in their contact with the community.

In [1], there is a real-time Alphabetic Sign Language to Speech Conversion VR System, which focuses on American Sign Language (ASL), uses a Data Glove with tact switches and a windowed template matching method to recognize ASL alphabets in real-time, followed by speech synthesis. Contact points help recognize certain ASL motions when there is confusion. It gives a natural communication method with a recognition rate of 15-20 samples per second, which is helpful for those who have hearing impairments.

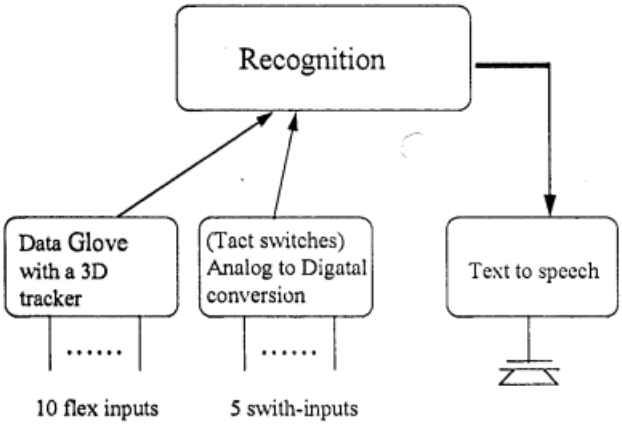

Figure 2.20: The system diagram [1]

In [14], the authors worked on gesture detection, feature extraction, and image processing systems for turning signs into text. Pre-processing manual segmentation, and feature extraction using Eigenvalues. Eigenvectors are the four modules that make up the system. The Linear Discriminant Analysis (LDA) technique has been used for sign identification. It transforms recognized motions into text and audio. This method rejects the need for interpreters by enabling smooth communication between sign language people and non-signers.

In [40], the authors worked on proposing a technology for turning sign language into text and aims to improve communication barriers for deaf and mute persons. The suggested solution uses computer vision and deep learning techniques, including MediaPipe for key point recognition, data pre-processing, labeling, feature synthesis, and LSTM neural network training. The system recognizes hand movements in real-time and converts them into text output. The method addresses the growing demand for effective communication solutions, particularly given the expected growth in worldwide hearing impairment. By merging cutting-edge technologies and

algorithms, the system provides a promising tool for bridging communication gaps, not only offering real-time sign language interpretation but also functioning as a significant learning tool.

In [30], the authors worked on a sign language-to-voice turning system that uses image processing and machine learning. Using a camera, hand movements are collected and analyzed using image techniques such as the OTSU approach. The linear classification method is used in machine learning, which is further advanced by the Naive Bayes classifier. The combination of image processing and Naive Bayes algorithms improves the system's performance, resulting in an effective method for identifying and interpreting sign language motions to voice.

Chapter 3

Background

3.1 Text to Sign Language

We have developed a webpage to take input and translate the words into sign language for this part. For this website, the Django framework was used, along with Javascript, HTML, and CSS for the front end. The python framework Django follows the MVT structure. In this website, we have three main pages: HomePage, Sign up and Log-in. The homepage features a navigation bar with links to the Sign Up and Login pages. It also includes a video of sign language representation of the world “Hello”. Below the video, there is a call-to-action button labeled ”Click to Start.” This button directs the user to the Log-in page. The new users can create accounts in the sign up page using necessary credentials like username and password.

Figure 3.1: Homepage of our website

Users are redirected to their personal dashboard, which includes a text input field, after logging in. Users can submit audio input by speaking sentences aloud by using the microphone icon button located beneath the text input box. The website translates spoken words into text by using the JavaScript Web Speech API for speech recognition.

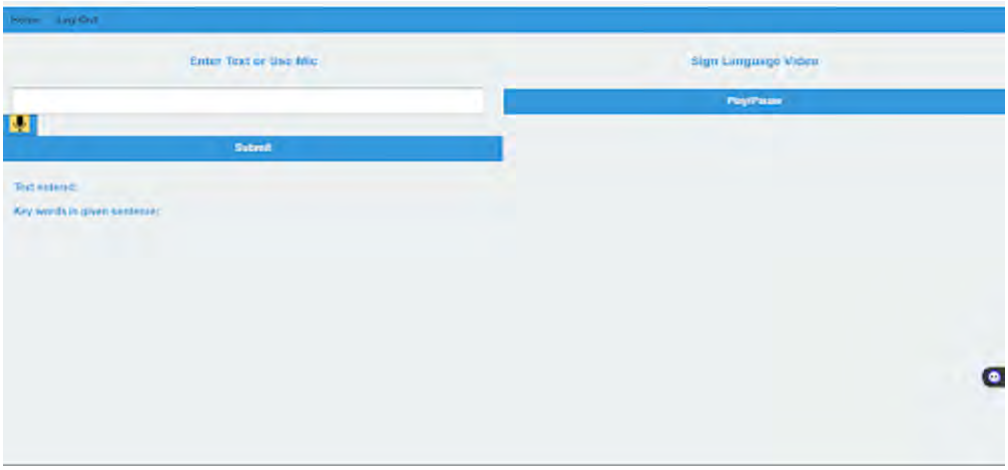

Figure 3.2: Dashboard of the users

Users can click the "Submit" button after they have typed their sentence or provided audio input. This action triggers the processing of the input, extracting keywords from the sentence, and generating the corresponding sign language representation. A list of keywords extracted from the submitted sentence is shown beneath the text input area. On the right side of the dashboard, there is a play/pause button. Users can click on that button to watch the videos demonstrating the sign language representation of each keyword.

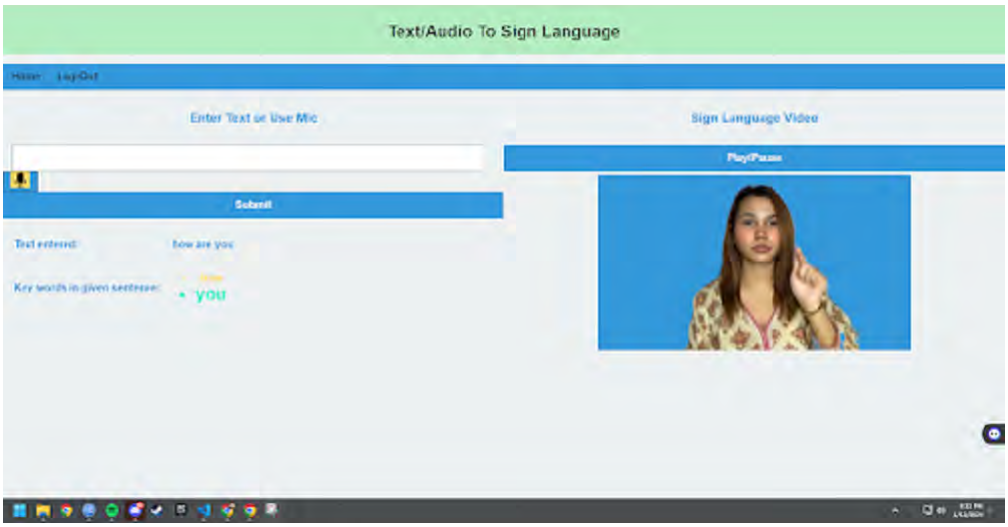

Figure 3.3: Converting text input to sign language

To extract keywords from a sentence we are using Natural language processing (NLP). The "Natural language toolkit (NLTK)" Python library is being used, which includes features like tokenization, stemming, POS-tagging, and more. Tokenization is the first technique we use in our work. The process [49] of splitting a sentence into smaller units is known as tokenization. Sentences can be tokenized in many different kinds of ways, but we've employed word tokenization which splits a sentence by words.Tokenizing the sentence "I am happy" will, for instance, result in an array of individual words like ["I", "am", "happy"]. Next, parts-of-speech (POS) tagging is being applied. The technique [33] of assigning a part-of-speech to each word in

a sentence is known as POS tagging in NLP. For POS tagging, we are utilizing the NLTK library's pos tag function. After that, we have a list of stop words. In text processing, these are common terms that are frequently omitted in order to highlight the more significant words. Lastly, lemmatization is accomplished using WordNetLemmatizer function. The process [50] of changing any sort of word to its basic root mode is called lemmatization. Words are being lemmatized based on their pos tags.

3.2 Sign Language to Text

3.2.1 Yolo Architecture

You Only Look Once, or YOLO, is a popular real-time object recognition algorithm. YOLO takes an input image and divides it into a grid. Predicting [52] bounding boxes and confidence scores for objects inside a grid cell is the responsibility of that grid cell. The head, [34] neck, and backbone make up the three essential components of YOLO's architecture.

1. Backbone: This component consists of a convolutional neural network (CNN) layer that extracts key features from an input image. Usually, a dataset for classification is used to train this component. In the first layers, the backbone extracts lower-level features from the input images and in the deeper layers, it extracts higher-level features.

2. Neck: This component is an intermediate between the backbone and the head. This part might have more CNN layers. It enhances the characteristics that the backbone extracted.

3. Head: The last component, it is in charge of making predictions using the information provided by the other two components. Head usually comprises task-specific subnetworks.

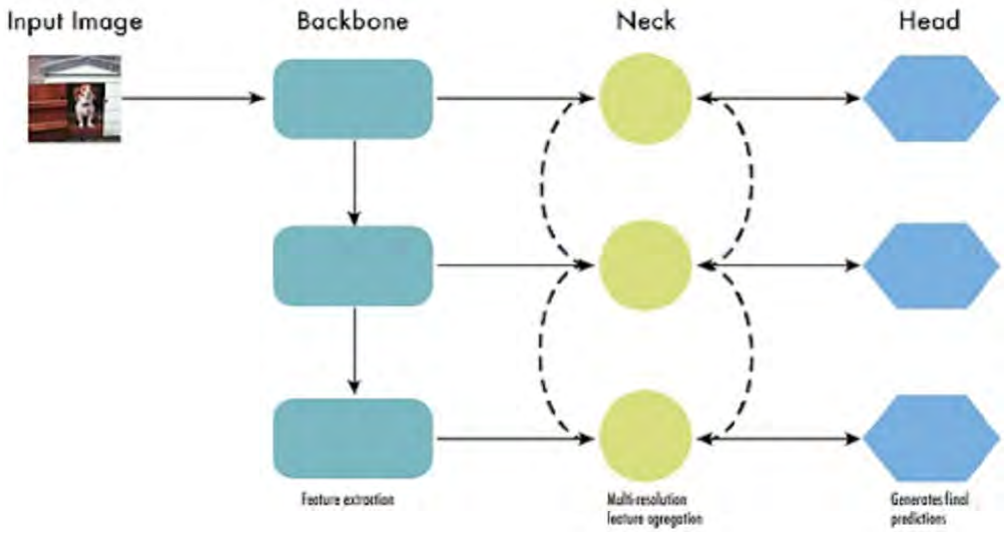

Figure 3.4: Architecture of modern YOLO models [34]

We are working with two versions of the YOLO model: YOLOv5 and YOLOv8.

3.2.1.1 YOLOv5

CSPDarknet53 [47], or Cross Stage Partial, is the backbone of YOLOv5. The feature map is divided into two parts by CSPnet, which works on each part independently before merging them via a cross-stage hierarchy. Because the map is divided into two parts and the layer works on each individually, YOLOv5 requires fewer parameters because each section requires fewer details. As a result, computation time is also decreased and YOLOv5 can make predictions more quickly.

Spatial Pyramid Pooling (SPP) and a modified Path Aggregation Network (PANet) are used by the neck of YOLOv5. The CSPNet technique has been applied to PANet in YOLOv5. This network is used to enhance the model's object identification capabilities. This network helps in improving the model's comprehension of data after modification. SPP is a technique that looks at various areas of a picture and extracts the most relevant feature without causing the model to slow down. To make YOLOv5 more efficient, a version of SPP called SPPF has been incorporated.

Three CNN layers make up the head of YOLOv5, which predicts scores, object classes, and bounding box locations.

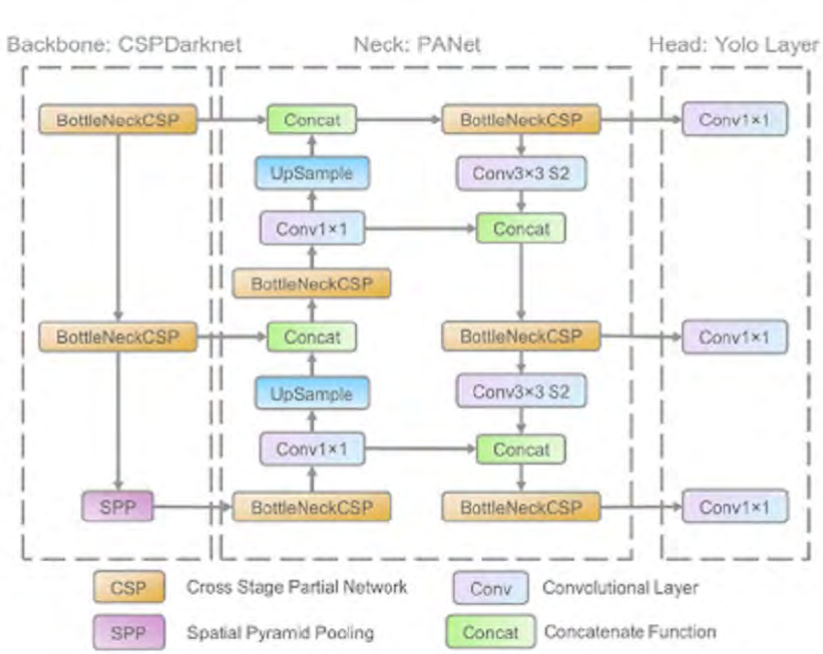

Figure 3.5: Architecture of YOLOv5 [47]

3.2.1.2 YOLOv8

YOLOv8 [34] makes use of the same CSPDarknet53 backbone as YOLOv5, but it also includes a new module called C2f to enhance detection accuracy. FPN [51], which more effectively collects features from different backbone levels, is also used in the neck of the YOLOv8. The PANet, which is incorporated into the head architecture, has the ability to identify objects of varying sizes or that may be partially hidden.

When it comes to accuracy, YOLOv8 provides greater accuracy than YOLOv5 because YOLOv8 [34] uses an anchor-free model which has a decoupled head which can identify an object and classify it independently. The usage of new loss functions has improved the object detection performance of YOLOv8.

3.2.2 CNN Architecture

Convolutional neural network (CNN) is a network architecture of deep learning which is very useful for object detection. CNN [46] architecture has two main parts: Feature extraction and a fully connected layer. Feature extraction is the process of identifying and extracting distinct parts of an image for analysis. Following looking at all these details, the fully connected layer predicts an object's class by combining all the data from previous layers.

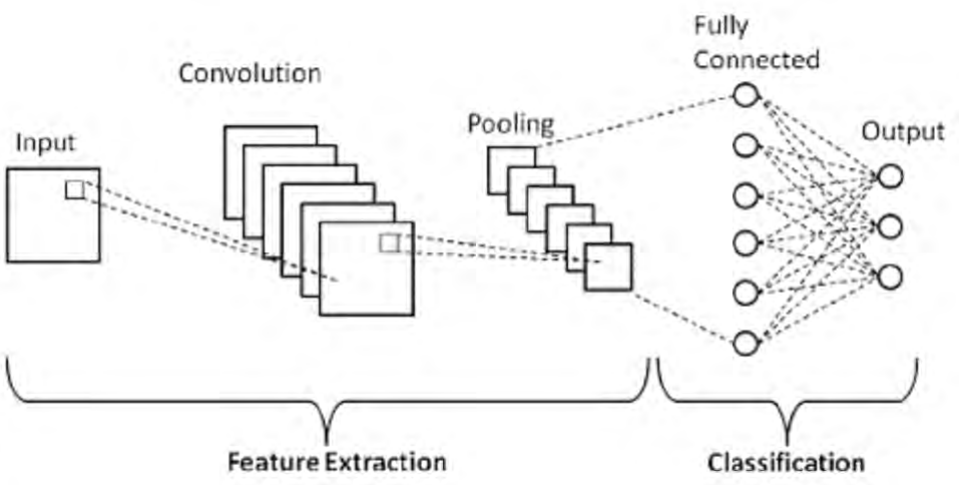

Figure 3.6: CNN Architecture [46]

1. Input layer: An image or collection of images is CNN's input.

2. Convolutional layers: Convolutional layers comprise the majority of CNNs. They perform convolutional operations to the input data by using filters, also known as kernels. By swiping over the input image and extracting local patterns, these filters are able to capture components such as textures, edges, and forms. In this case, a mathematical operation is performed between the filter and the input image. A feature map is this layer's output, which is passed to the next layer.

3. Pooling layers: The most significant features are highlighted and the computational burden is decreased by downsampling the spatial dimensions of the feature maps using pooling layers. Max pooling and average pooling are two common pooling methods.

4. Fully connected layers: Traditional neural network layers are fully linked layers, in which every neuron in the layer above and layer below is coupled to

every other neuron. Typically, these layers come before the output layer. This involves feeding the fully linked layer with the flattened input image from the preceding layers.

5. Dropout: A dropout layer, which reduces the size of the model by removing a few neurons from the neural network during training, is used to avoid overfitting.

6. Activation function: The component that determines whether or not information should be forwarded is known as the activation function. It aids in approximating the network's complexity. The network gains non-linearity from this function. ReLu, softmax are two of the most commonly used activation functions.

3.2.3 Mediapipe Framework

We are going to incorporate mediapipe framework with cnn. It is an [45] open-source framework that can be used to create pipelines that process any kind of sensory data, including audio and video and analyzes multimedia data in real-time, especially in the fields of computer vision and machine learning. Because of the modular design of the framework, developers may create and mix various components to create intricate pipelines for tasks like position estimation, hand tracking, and face detection.

Chapter 4

Dataset Description

4.1 Text to sign language

From what we have researched so far, we can say that there is a lack of suitable dataset to work on text to sign language translation. Thus, for ASL, we have created our own dataset. Videos representing alphabets, digits 0 through 9, and the most frequently used words in the English language are included in this dataset. In total, there are 150 videos.

Figure 4.1: Dataset of Text to Sign Language

4.2 Sign language to text

4.2.1 YOLOv5 and v8 model dataset

We created a database of our own for ASL. Here we created classes for each sign. There are 22 classes in total. After creating images for each class and while resizing them at 640*480px, we annotated each of the data samples according to their specific

class. After this we split the classes into 80[Train]:20[Test] ratio. These are the classes of our dataset with an image of each of the classes. The classes are:

1. 'Are you Hurt_OR_I am hurt_',
2. 'Are You Thirsty_OR_I am Thirsty',
3. 'Be Careful',
4. 'Did you eat_OR_I want to eat',
5. 'Dont do that',
6. 'Go There',
7. 'Hello',
8. 'I am Cold_OR_Are you cold',
9. 'I am mad_OR_Why are you mad',
10. 'I am nervous',
11. 'I am Sad_OR_Why are you sad',
12. 'I am Sorry',
13. 'I dont understand_OR_Do you understand',
14. 'I love you',
15. 'I Need Help',
16. 'Need a bandage',
17. 'No',
18. 'Please_OR_Kindly',
19. 'Stop it_Or_Stop This',
20. 'Thank You',
21. 'Why are you Crying,',
22. 'Yes'

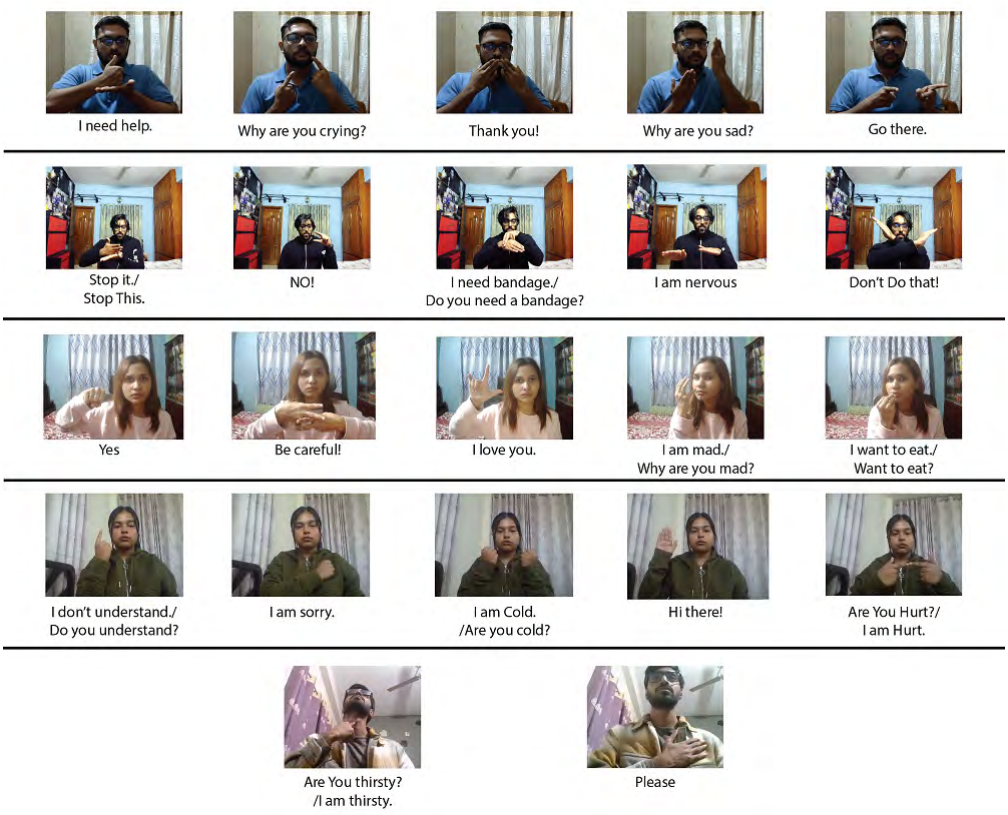

Figure 4.2: Total 22 sample images of each of the classes.

Furthermore, most often the data of a single sentence has some movement including hand gestures. Keeping that in mind, we have used samples which include movement for those specific sentences. We got a good amount of accuracy because of that when we are using movement gestures. We took 80 image samples for each class in the train data and 20 image samples for each class in test data. Overall we got 2200 data samples, where we used 1760 samples for training and 440 samples for testing our data. Moreover, for annotating each and every picture we used the "labelImg" tool.

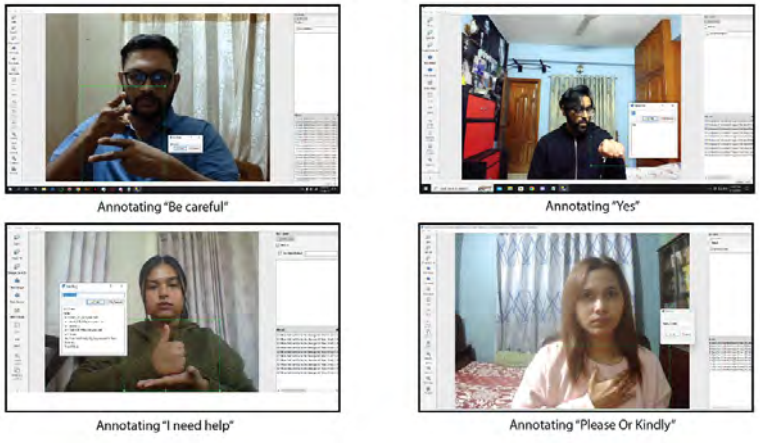

Figure 4.3: Annotating using "labelImg"

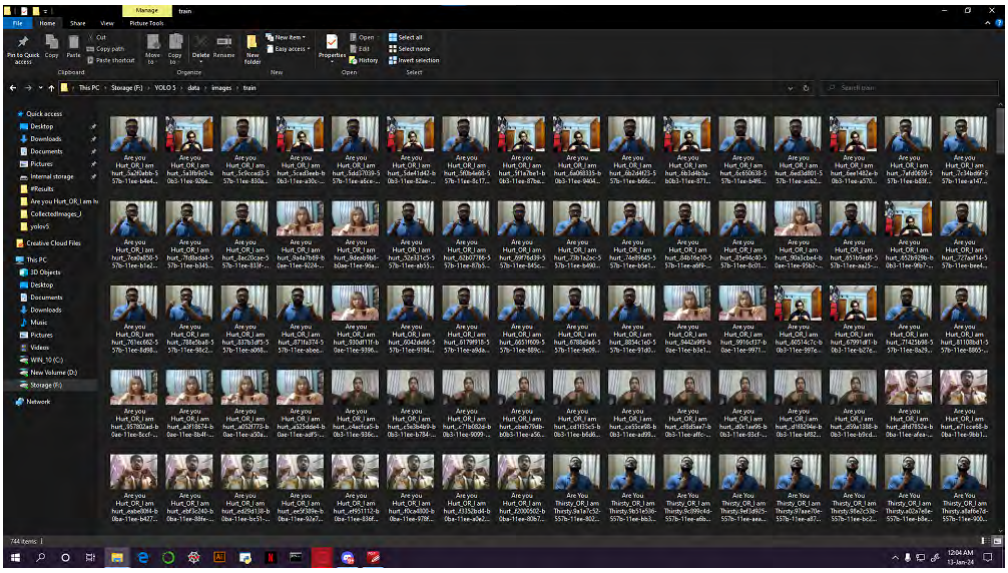

Figure 4.4: Snapshot of the testing data

4.2.2 Dataset of model based on CNN and mediapipe

The dataset here is basically coordinate points taken from each frame. The coordinates here are taken for each and every class. Using the webcam and python's opencv library, we collected data for each class. We have made hand sign gestures from different angles and at different parts of the frame while collecting data so that the model gets better trained and can detect the hand sign performed even if it has been performed from a different angle in the frame. The classes for the hand sign gestures where both hands are needed are provided with more data compared to the classes where only one hand is needed to perform hand sign gestures. This is done because it is comparatively more difficult to identify hand signs with both hands for this model. The model gets trained based on these coordinate points after preprocessing is done. The dataset has been made for the following 10 classes:

1. Hello there
2. I love you
3. I am sorry
4. Please
5. I need help
6. Go there
7. Why are you crying?
8. Be careful
9. Stop it
10. Don't do that

Chapter 5

Model Description and Result analysis

5.1 Text to Sign Language

We will implement spoken language sentences to sign language translation using Natural Language Processing (NLP). The framework will utilize the Natural Language Toolkit (nltk) for part-of-speech tagging to identify the tense of the input sentence. A dictionary will be used to hold verb counts for various tenses, and a list of predefined stop words will be utilized for filtering. After the words have been lemmatized, particular tense-indicating words like "Before," "Will," or "Now" will be added in accordance with the determined tense.

We will need a dataset containing videos corresponding to words. The dataset will contain most used/common English words and alphabets. The words will be added to a list after each word in the input sentence and its matching part-of-speech tag have been processed and lemmatized according to their POS tag. For every word in the list, the model will look for video files in the dataset that match it. The word will be split up into individual characters if there isn't a video available in the dataset. Finally, the processed words and the original sentence will be passed to a html template for rendering. The primary benefit of this framework is that it does not require system modifications when a new word is added to the dataset.

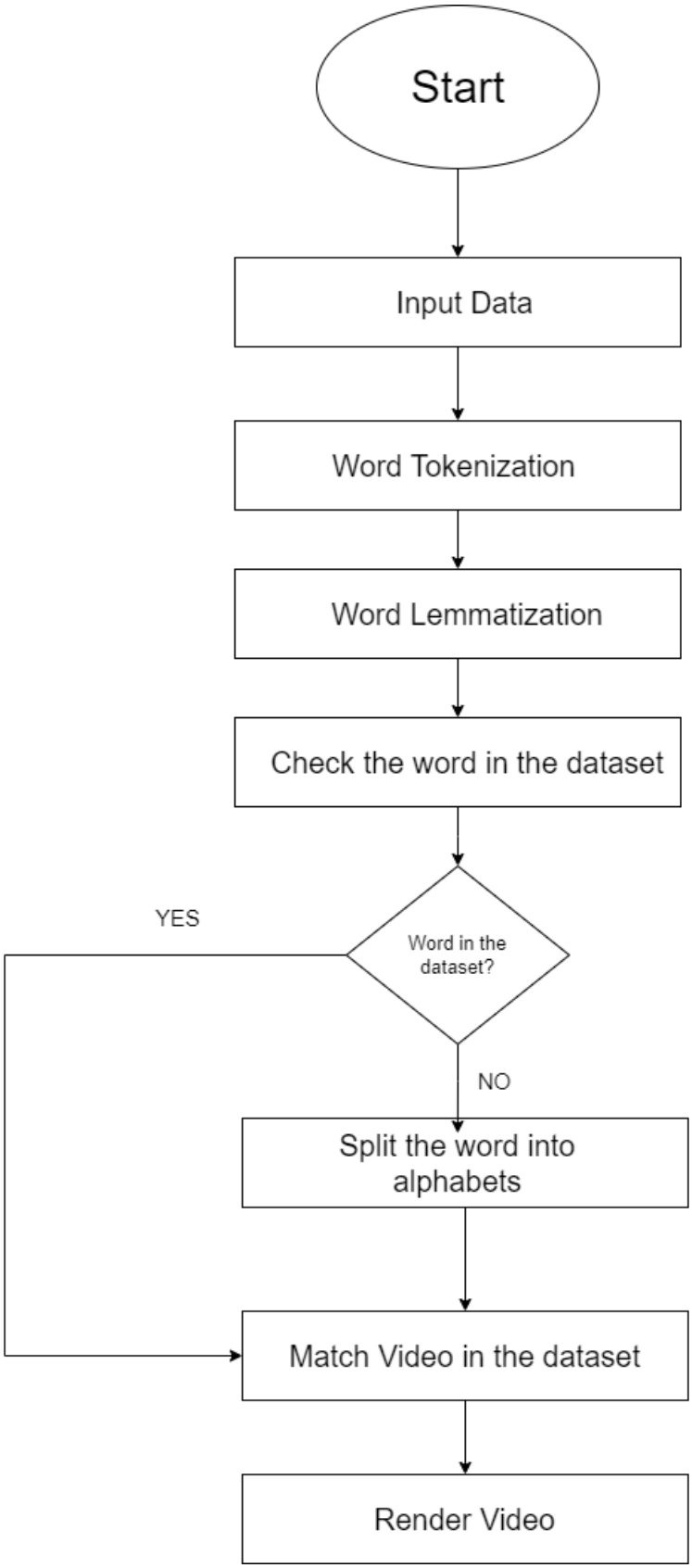

Figure 5.1: workflow of text to sign language translation

5.1.1 Result

It is possible to provide textual or audio input. The sentence is divided into key words when input is received.

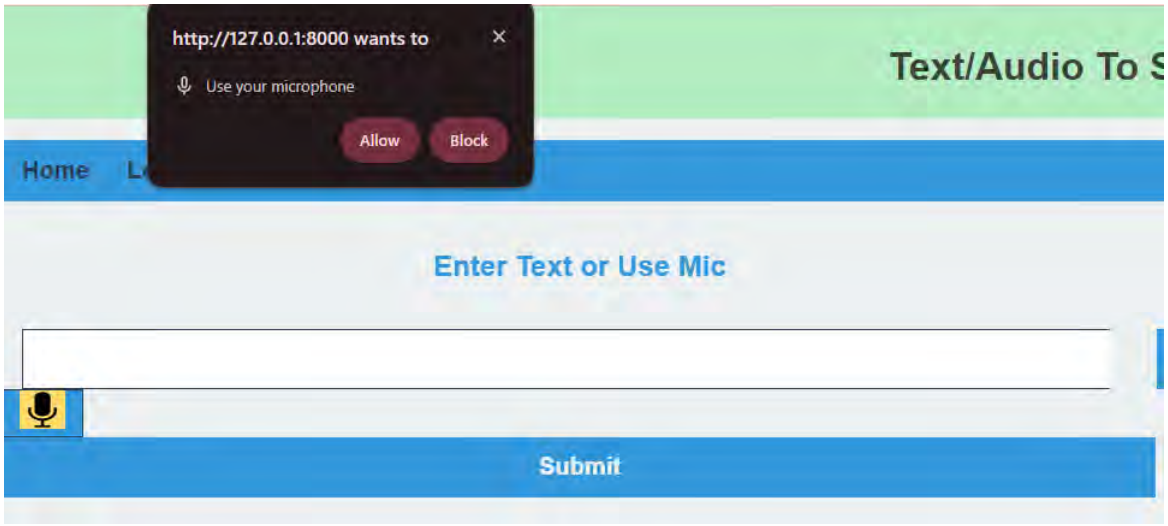

Figure 5.2: Taking input

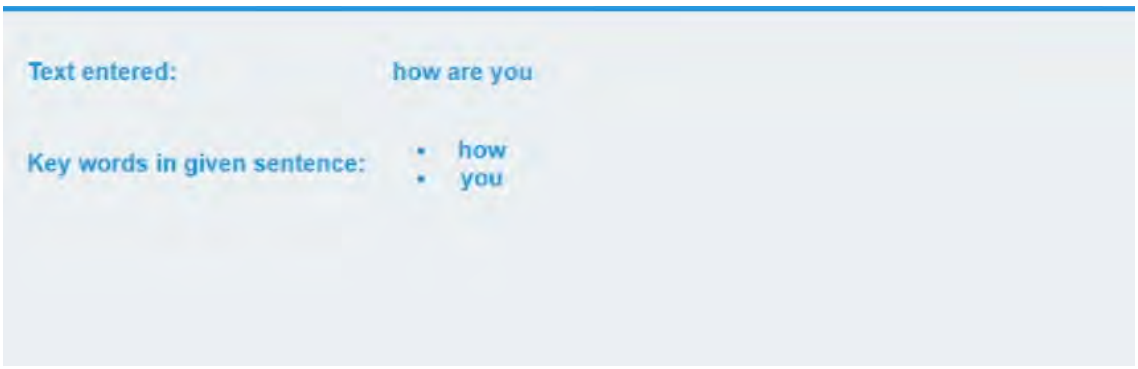

Figure 5.3: Key words in a sentence

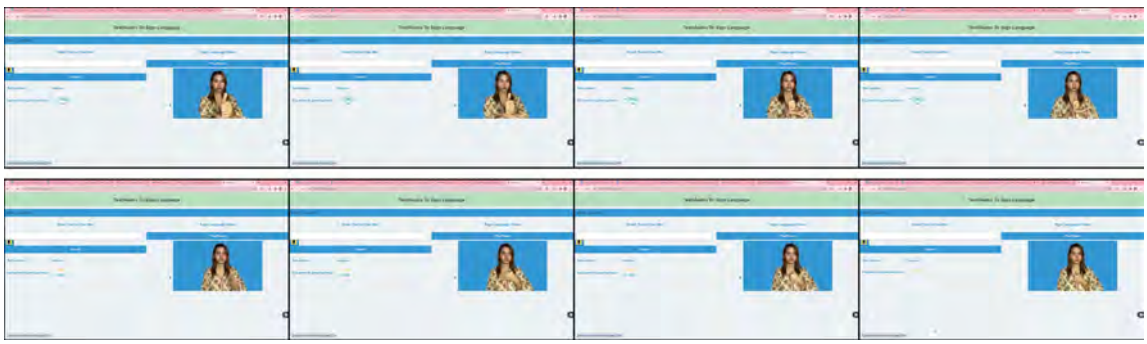

Figure 5.4: Frames of translating text input to sign language

Youtube Link for Text to Sign Language website video:
https://youtu.be/emInlNMSDHA

5.2 Sign language to text :

5.2.1 YOLO Models

5.2.1.1 Yolo v5 and v8 model overview

For our sign language detection, we are applying the YOLO v5 and v8 models, YOLO- v8 is a state-of-the-art model in the ongoing time. This YOLO model is an upgraded version of its predecessors. It uses image and video data to train and can detect the gestures and other detection scenarios with almost 100 percent accuracy if implemented correctly. YOLO v8 uses a custom data loader mosaic which helps the data to be loaded into the model while training and testing the raw data. However, the data collection is more likely the same with v5 and v8 models. Both of these models are compatible with multiple data collection styles, which are:

1. Single shot detection where the entire image is detected as the whole data. In this case we will not get individual data for a specific item but an overall summary of the detection.

2. Grid-based detection where the data image is divided into equal grids and each of the grid cells are responsible for predicting bounding boxes and classifying the bounding object inside them.

3. Bounding box prediction, here the model predicts the bounding boxes with coordinates and dimensions. However, we will have to specify the class and annotate the boxes in each of the data and train the model with that specific information in order to predict.

4. Anchor boxes, here the model uses anchor boxes to improve the accuracy of bounding boxes to predict. The anchor boxes are predefined shapes that help the model better to adapt to objects of different sizes and aspect ratios.

From here bounding box prediction was more suitable for us as our hands move constantly and we will have to deal with more than one shape and aspect ratios. Therefore, we added our hand gestures with bounding boxes as it can be easily manipulated and the end result was satisfactory for both of the models.

5.2.1.2 Workflow for YOLO v5 and v8 Models

As we already know that the data collection for YOLO v5 and v8 are the same as both of them share a similar architectural pattern.First we took the sample data one by one using a custom code where we first assigned the classes and numbers. After taking each and every class we used, we had to annotate each of the sample pictures with its corresponding class so that the actual annotation in real time may work. We preprocess the same data which we collected and run our decided epochs for both of the models. Here we trained our data with 80 percent of the created dataset and validated it with the other 20 percent. We took the best fit model we got from both of the training sessions and got the desired output.

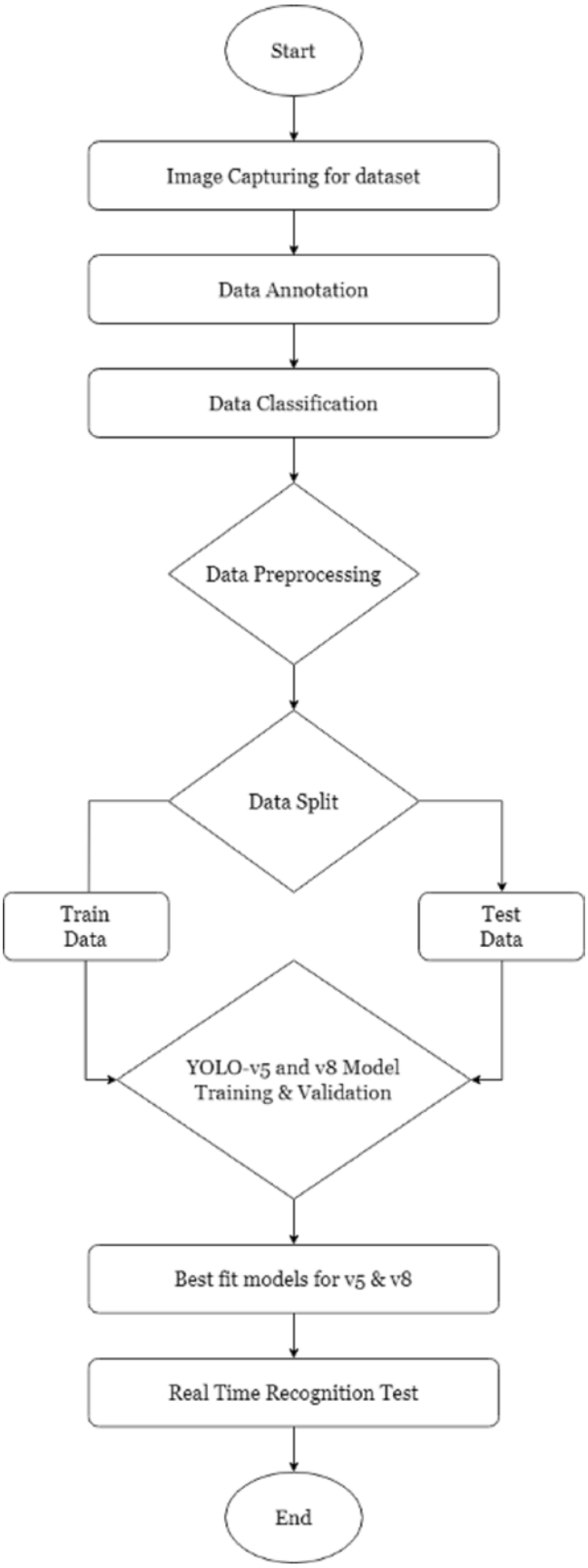

Figure 5.5: Workflow for YOLO v5 and v8 models.

5.2.1.3 YOLO v8 model Real time output

Here, we ran 50 epochs for our YOLO v8 model with a total of 22 classes.

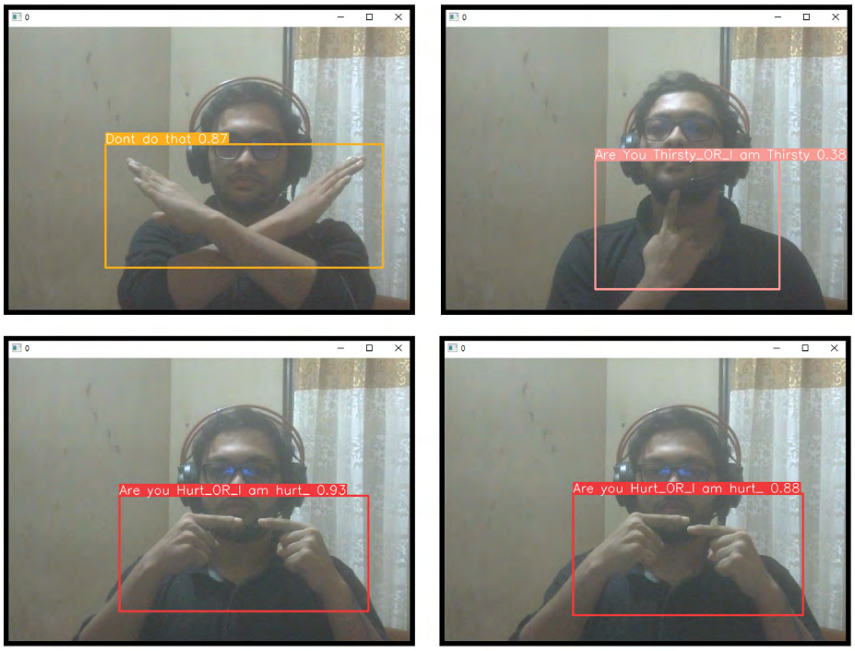

Figure 5.6: Real time signs of some classes from v8 model.

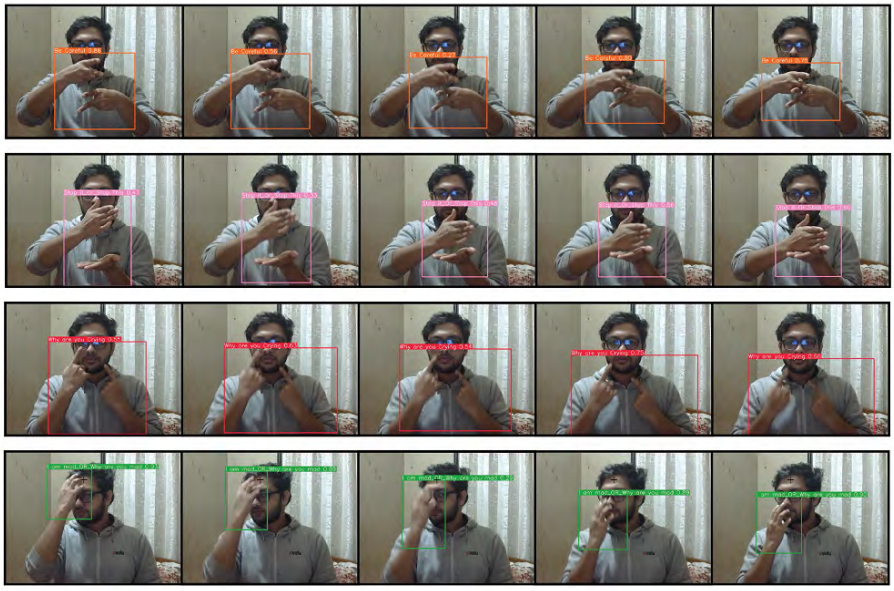

Figure 5.7: Frames of real time moving sign language classes from v8 model.

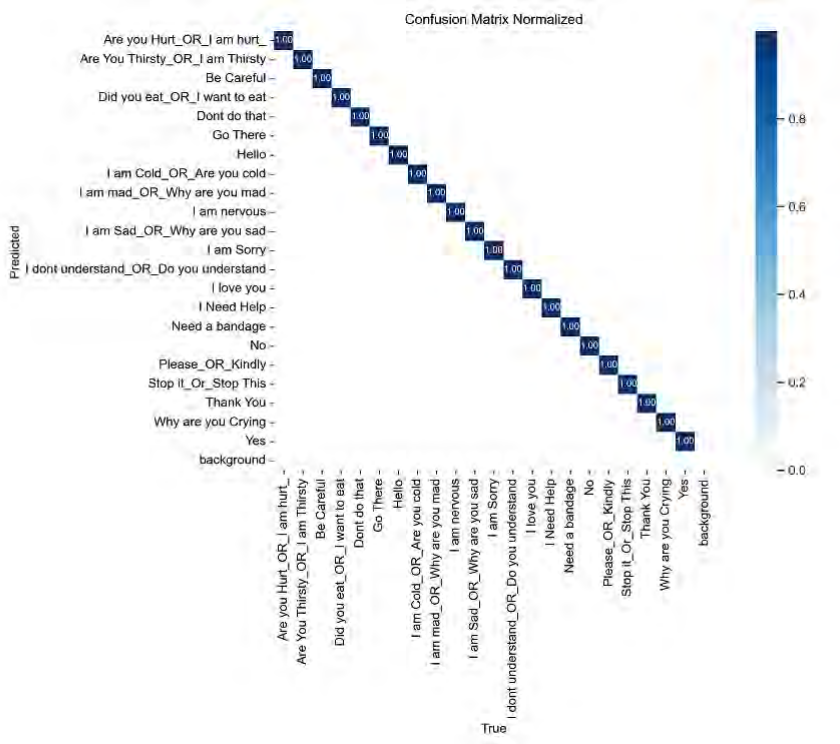

Figure 5.8: Confusion matrix of YOLO v8 model.

So from the matrix we can see that all classes are correctly detected.

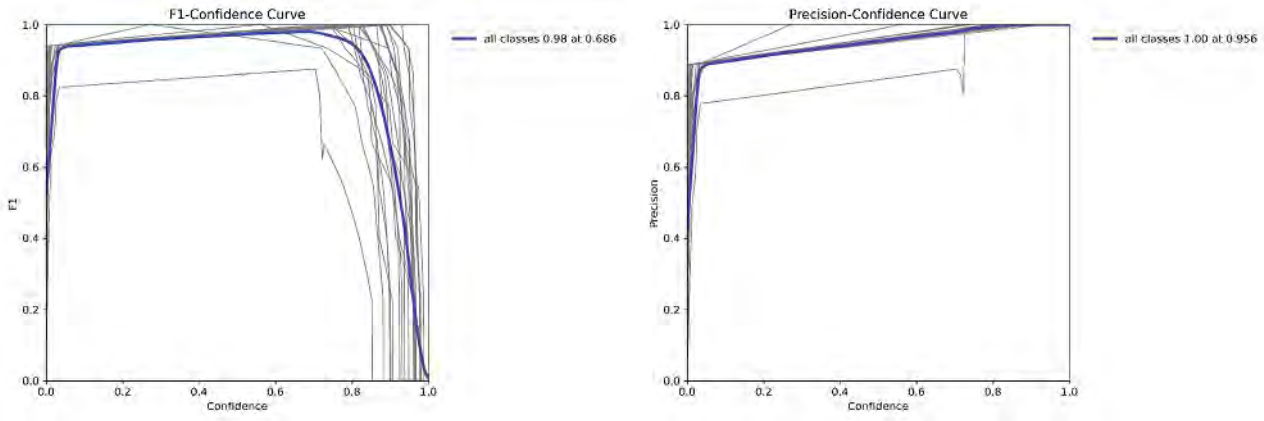

Figure 5.9: F1 confidence curve and precision graph of YOLO v8 model.

Here, we can see that the confidence curve for the v8 model is 68.6 percent and the precision level is 95.6 percent for all classes.

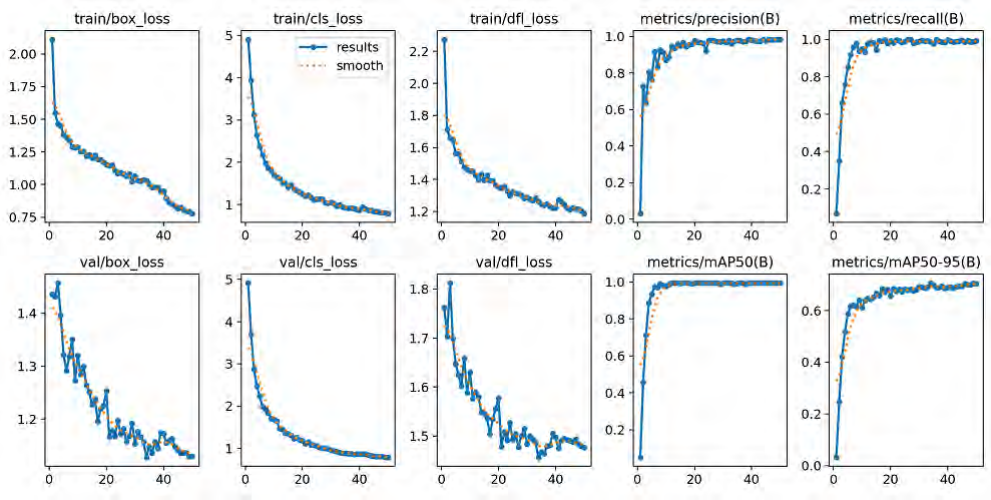

Figure 5.10: Validation graphs of YOLO v8 model.

Here, we see the training and validation box loss, class loss, object loss, precision metrics and recall metrics consecutively for the v8 model.

5.2.1.4 YOLO v5 model Real time output

Here, we ran 100 epochs for our YOLO v5 model with the total of 22 classes.

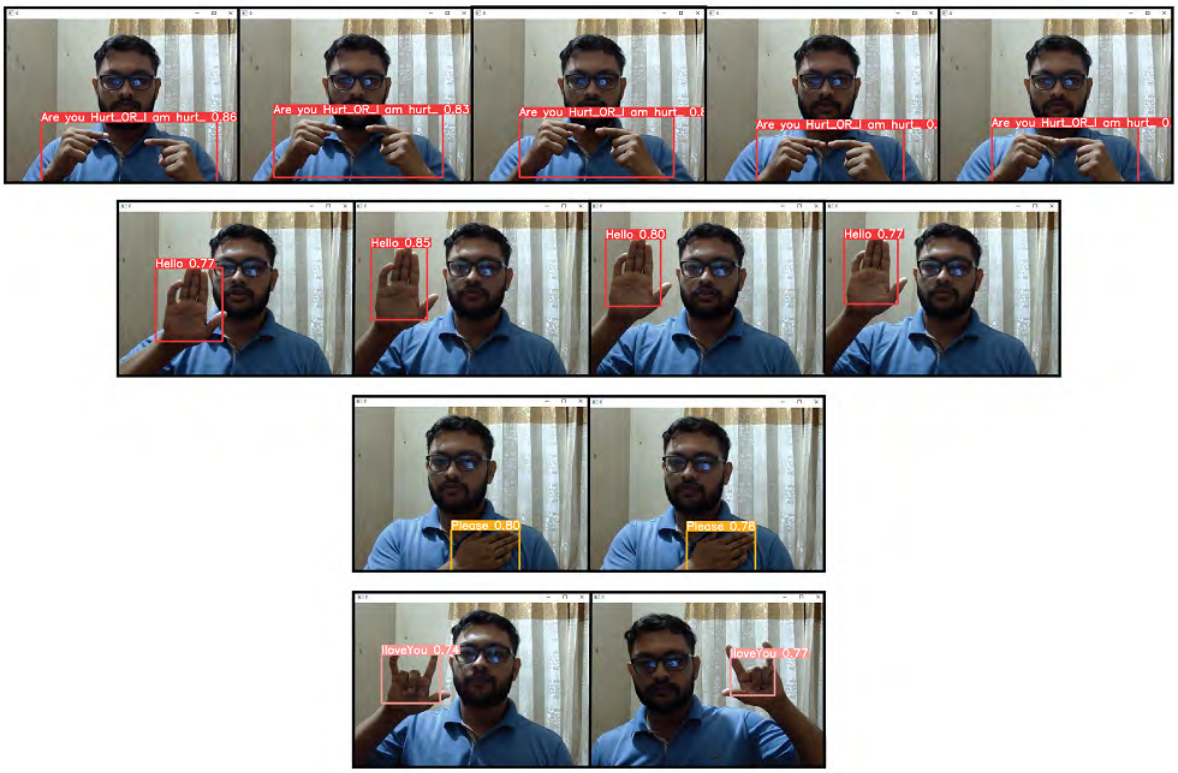

Figure 5.11: Real time detection output of our YOLO v5 model.

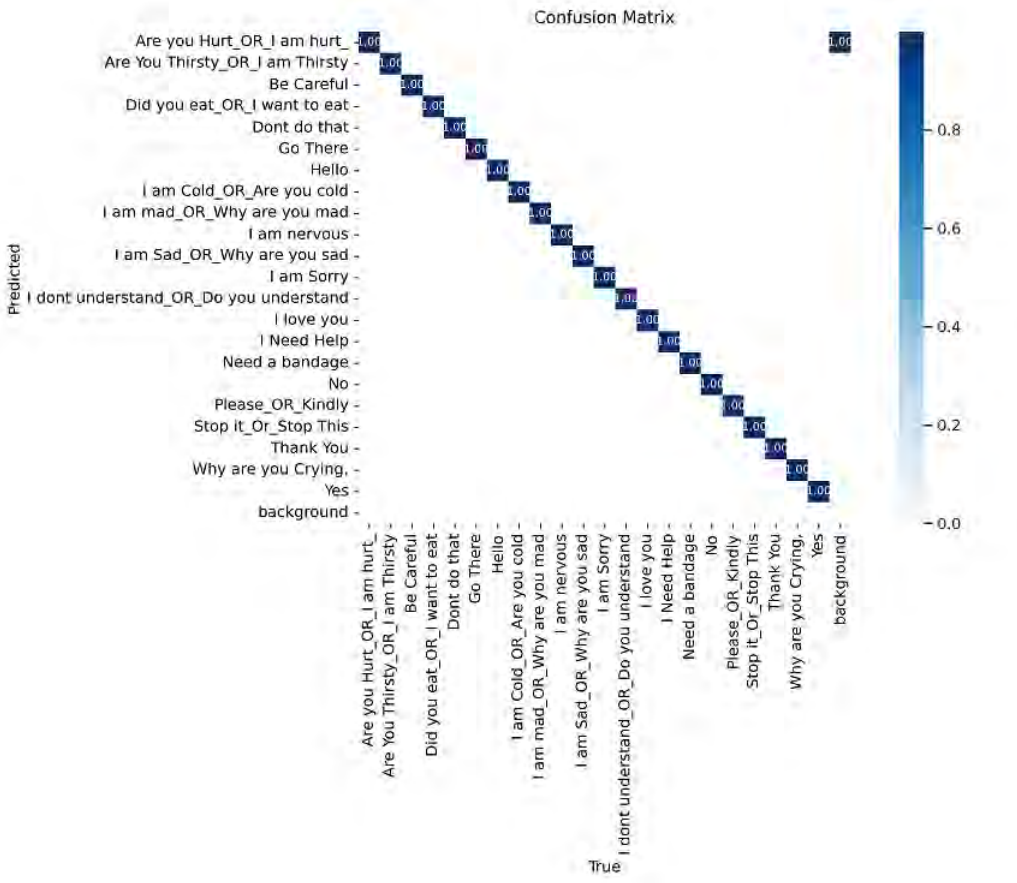

Figure 5.12: Confusion matrix of YOLO v5 model.

So from the matrix we can see that all classes are correctly detected.

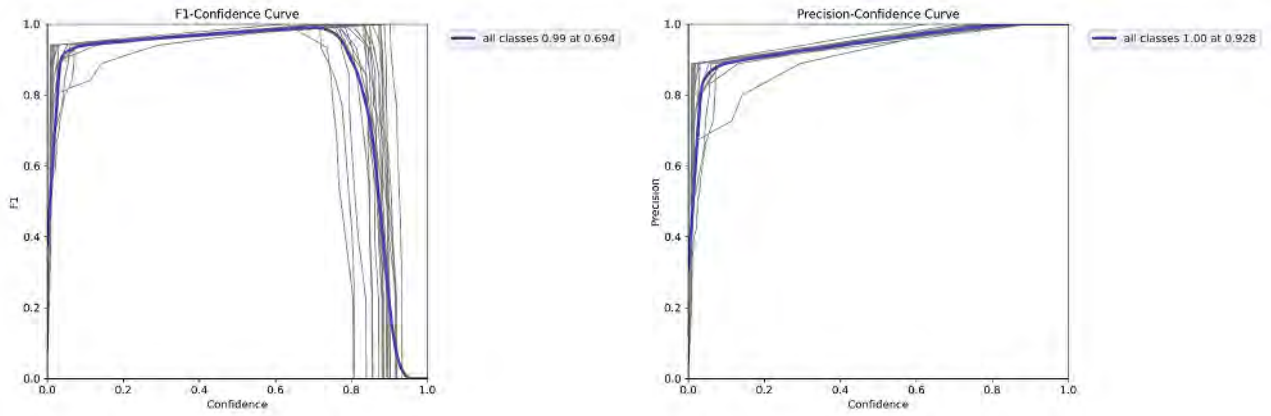

Figure 5.13: F1 confidence curve and precision graph of YOLO v5 model.

Here, we can see that the confidence curve for the v5 model is 69.4 percent and the precision level is 92.8 percent for all classes.

In the following figure, we see the training and validation box loss, object loss, class loss, precision metrics and recall metrics consecutively for the v5 model. As we ran the model for 100 epochs. The curve is much smoother than the v8 model.

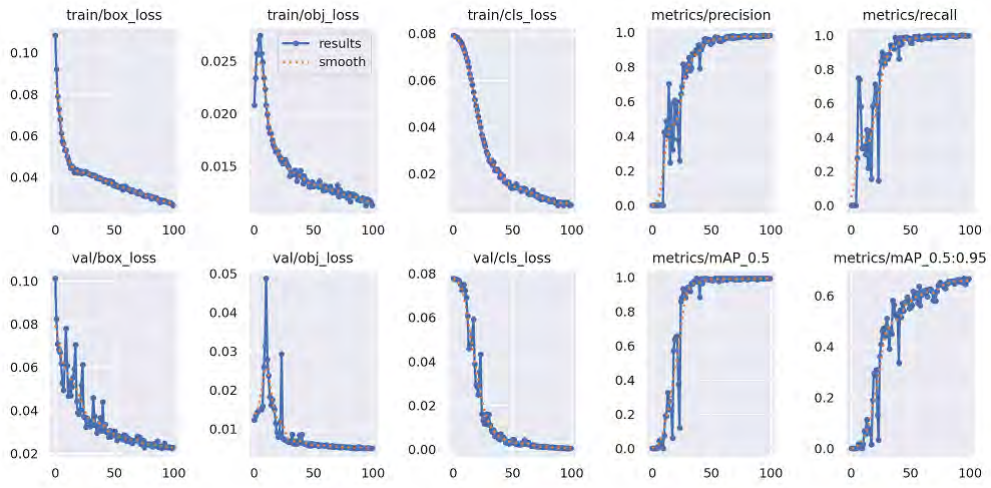

Figure 5.14: Validation graphs of YOLO v5 model.

5.2.2 Model based on CNN and mediapipe framework

5.2.2.1 Overview and workflow of the model

We are also doing sign language detection with the help of a simple CNN model implemented by python's openCV library. One kind of deep neural network that excels at image-based tasks is the convolutional neural network (CNN), which is especially useful for object detection. In short, finding items inside an image or video frame is known as object detection. Thanks to its ability to automatically learn hierarchical features from input data, CNNs have shown to be effective in this endeavor.

Let us dive into details about how the model is working. We are using python's opencv library to capture frames from webcam or any other camera. The frames are 2D matrices basically which hold pixel values and we have to identify in which part of the image the hands are being used to perform sign language. Using the mediapipe framework, we will identify the joints in the user's hand which look like a skeleton-like structure. These joints are also called landmarks.We are going to use it to train a feed forward neural network which is simple and lightweight and ultimately it will be used to detect the user's hand sign. The skeleton-like structure which tells us about the joints in the user's hand is significant to our work. Because if we trained directly using the whole image then we would need a huge amount of training data in different environments and we also need to take plenty of other things into consideration like different hand and finger sizes. This would make the work more complex. So, the skeleton-like structure simplifies our work a lot.

Before training, the dataset is preprocessed. The objective in this step is to standardize and normalize the input data so that it is suitable to be fed to the CNN model.

The dataset has been trained on the CNN model. The architecture of this model is heavily influenced from the official Tensorflow documentation of a fundamental and stark CNN model. Since that model was well trained and tested for an enormous dataset , we were confident that following it will be beneficial for making ours. Here,

in the CNN there are three Conv2D layers and max-pooling layers between them. When the model gets deeper, the width and height dimensions start to reduce. Because of this, we get more output channels in each Conv2D layer. When the work of the conovulational base is done, dense layers get added on top. Before going through two dense layers, the final output from the convolutional base is flattened into a 1D vector. In the final dense layer, the number of outputs will be the number of classes provided.

We trained the model on 10 classes. The number of data provided in some classes differs from other classes based on how complex the sign is. The train and test ratio here was 75[train]:25[validation]. The model underwent 500 epochs with a batch size of 128 during the training process. The accuracy rate was 94.94 percent.

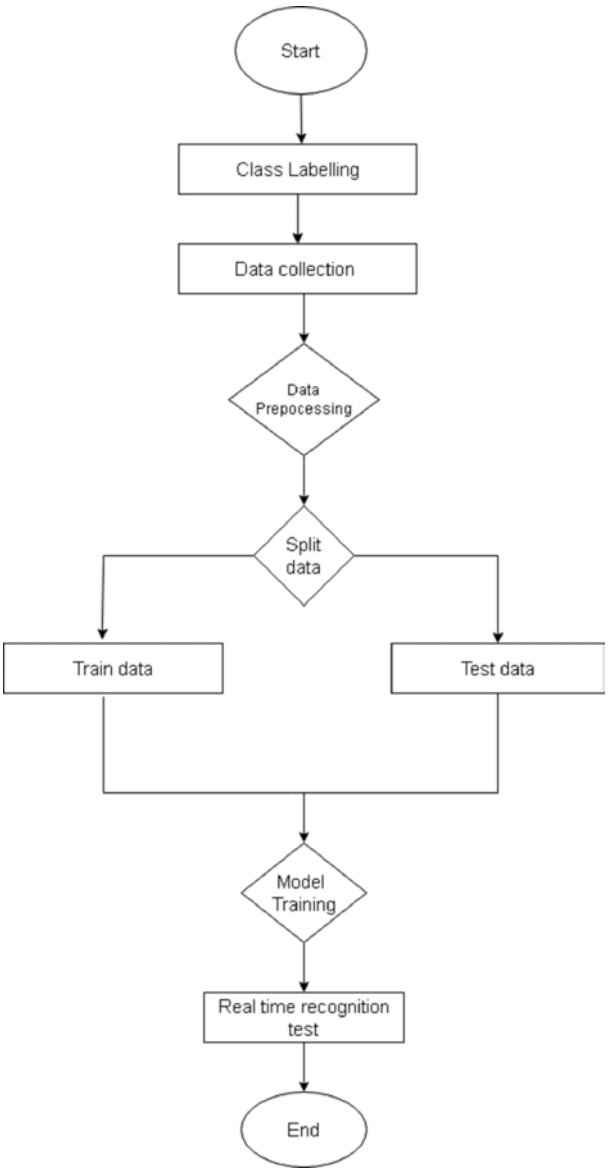

Figure 5.15: Workflow of the model

5.2.2.2 Real time output

Figure 5.16: Real time output detection of the model

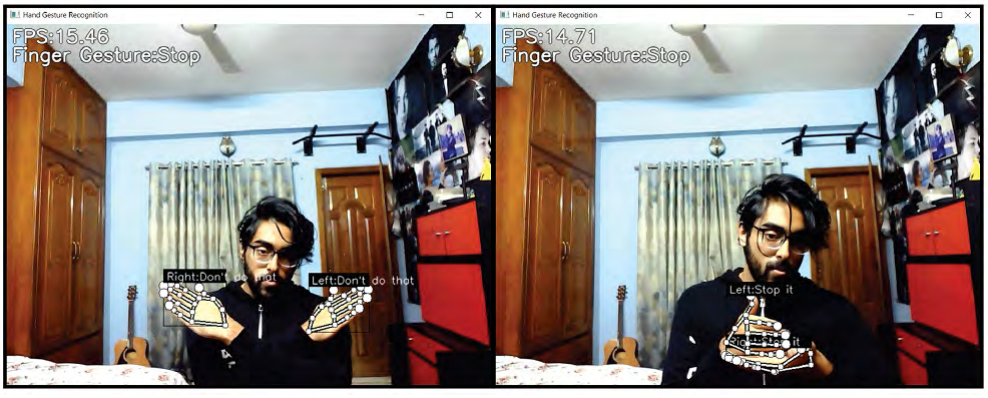

Figure 5.17: Real time output detection of the model

So, from the snapshots presented above we can see that the model is able to recognize the hand signs performed in real time.

Classification Report

	precision	recall	f1-score	support
0	1.00	1.00	1.00	13
1	1.00	1.00	1.00	16
2	1.00	1.00	1.00	11
3	0.86	1.00	0.92	6
4	0.96	0.96	0.96	28
5	1.00	0.74	0.85	23
6	0.86	1.00	0.92	37
7	0.95	1.00	0.98	41
8	1.00	0.75	0.86	16
9	0.96	0.98	0.97	46
accuracy			0.95	237
macro avg	0.96	0.94	0.95	237
weighted avg	0.95	0.95	0.95	237

Figure 5.18: Classification report

Here, we see that the precision, recall and f1-scores of some classes are comparatively lower. The hand signs performed in those classes are difficult for this model to recognize despite providing enough data in those classes.

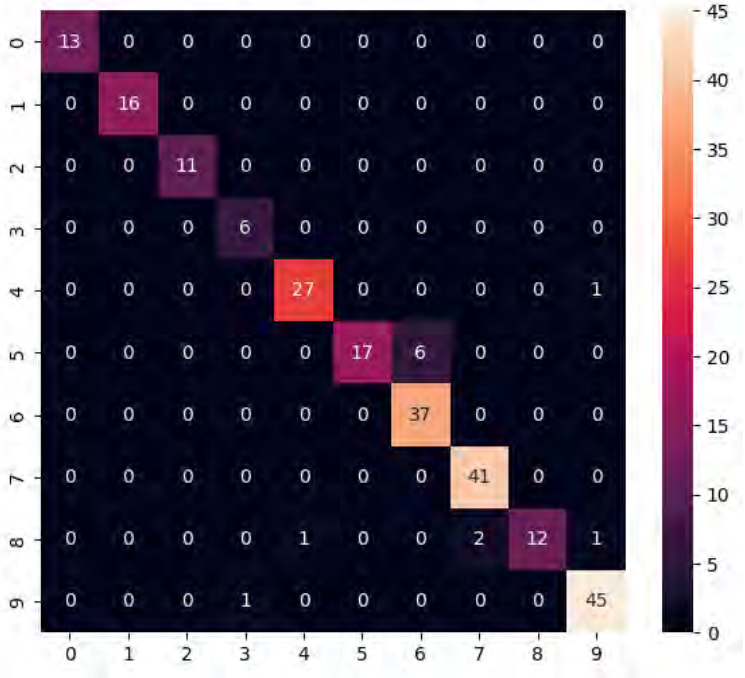

Figure 5.19: Correlation matrix

From the correlation matrix here, we can see that all classes are being identified. But sometimes some classes are being falsely identified. For example- 'Be careful' (Class-07) has been misclassified as 'Stop it' (Class-08) sometimes. Again, 'I need help' (Class-04) has also been misclassified as 'Stop it' (Class-08). So, the model cannot always predict accurately.

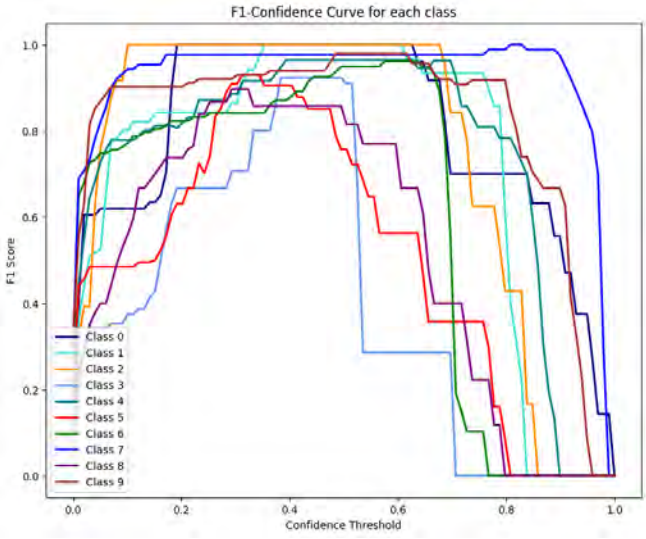

Figure 5.20: F1 confidence curve

From the F1 confidence curve, we can deduce that most classes have an f1 confidence score on or over 0.8 . One or two classes have a score between 0.7 and 0.8 .

Model	**Classes**	**Batch Size**	**Epochs**	**Accuracy (%)**	**F1 Score**
YOLO v8	22	16	50	95.6	0.686
YOLO v5	22	64	100	92.8	0.694
CNN and mediapipe Framework	10	128	500	94.94	0.95

Table 5.1: Model Performance Comparison

Although the model based on CNN and mediapipe framework shows a high accuracy, it sometimes makes false detection which has previously been mentioned above. The model is unable to always perfectly identify the hand signs where both hands have been used to perform the sign despite providing enough data. So, it is difficult for this model to identify or recognize comparatively complex hand sign gestures. After seeing its performance on the number of classes we have trained it in, we have come to the conclusion that this model is not suitable for large scale work.

The YOLO models on the other hand have performed much better than the CNN model. Though the YOLO v5 model shows a little bit less accuracy than the CNN model, it does not do any type of false detection. The YOLO v8 model also does not have any case of making any wrong detection. The YOLO v8 model functions better than the YOLO v5 model. It is because YOLO v5 shows an accuracy of 92.8 percent after training the model for 100 epochs. Whereas the YOLO v8 model shows an accuracy of 95.6 percent after training the model for just 50 epochs. Therefore we can say that the YOLO v8 model is the best one here for real time sign language detection out of all the models used.

Chapter 6

Limitation and Future work

6.1 Text to Sign

Certain sentences that contain stop words (For example - ‘ - apostrophe) that we utilized for filtering are not compatible with the framework. One further drawback is that the video outputs are not smooth. Before each word is depicted on a sign, there is a pause before and after each sign is shown. We want to work on these problems in the future and also make an app version of the site with a 3D model with more smoother transitions.

Figure 6.1: Frames of an input sentence with a stop word

6.2 Sign to text

It is possible to work on video data in the YOLO v8 model. Though we have not used video data in our dataset for sign language recognition, it is something we aim to incorporate in our work in the future. Moreover, we also intend to add words and letters to our dataset while constructing proper sentences for the user. As we know that sometimes if we want to form a sentence we also need letters and words to describe, that’s why we intend to enrich our dataset as well. We also aim to add facial expression recognition in our work as there are some signs where the facial expression is important to determine and put emphasis on what the sign language user is trying to say. Furthermore, if we can get enough resources to increase the computational power, we can work on a large dataset where there will be hundreds of classes. In addition to that, we plan to make an app version of this model. By the app, the sign language which will be performed can be recognized. To top it off, we would like to say that our work here on ASL detection can also be applied to other sign languages as well. Therefore, if it is possible, we would like to work on detecting other sign languages.

Chapter 7

Conclusion

According to the World Health Organization (WHO), with 1.5 billion people in the world already suffering from hearing loss, there stands a chance for the number to increase to over 2.5 billion by 2050.[53] The deaf community is deprived of basic human rights like health care, education and even minimum wage jobs simply because of their inability to communicate with the hearing people using spoken language. Our YOLO and CNN based models for sign to text and adding to that the framework we made for converting text into sign language aim to bridge this communication gap that has been prevalent in the community for ages by providing the fastest real time solution. Only an insignificant number of hearing people know how to communicate in sign language due to their lack of awareness or uninterest. Our bidirectional sign language conversion models and framework, will be able to capture and convert images containing sign language to spoken language and vice versa. This will ensure an equal spot for the deaf people in the society as well as the working class as both the communities will be able to overcome this language barrier. In conclusion, our system will be helpful for both hearing and hearing impaired people to communicate effectively with one another. Hence, shortening the existing communication gap.

Bibliography

[1] R.-H. Liang and M. Ouhyoung, "A real-time continuous alphabetic sign language to speech conversion vr system," in *Computer graphics forum*, Wiley Online Library, vol. 14, 1995, pp. 67–76.

[2] J.-S. Kim, W. Jang, and Z. Bien, "A dynamic gesture recognition system for the korean sign language (ksl)," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 2, pp. 354–359, 1996.

[3] C. Vogler and D. Metaxas, "Adapting hidden markov models for asl recognition by using three-dimensional computer vision methods," in *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, vol. 1, 1997, 156–161 vol.1. DOI: 10.1109/ICSMC.1997.625741.

[4] S. Dangsaart, K. Naruedomkul, N. Cercone, and B. Sirinaovakul, "Intelligent thai text–thai sign translation for language learning," *Computers & Education*, vol. 51, no. 3, pp. 1125–1141, 2008.

[5] T. Dasgupta, S. Dandpat, and A. Basu, "Prototype machine translation system from text-to-indian sign," *NLP for Less Privileged Languages*, vol. 19, 2008.

[6] D. Kouremenos, S.-E. Fotinea, E. Efthimiou, and K. Ntalianis, "A prototype greek text to greek sign language conversion system," *Behaviour & Information Technology*, vol. 29, no. 5, pp. 467–481, 2010.

[7] A. V. Nair and V. Bindu, "A review on indian sign language recognition," *International journal of computer applications*, vol. 73, no. 22, 2013.

[8] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen, "Sign language recognition using convolutional neural networks," in *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part I 13*, Springer, 2015, pp. 572–578.

[9] K. Kaur and P. Kumar, "Hamnosys to sigml conversion system for sign language automation," *Procedia Computer Science*, vol. 89, pp. 794–803, 2016.

[10] M. S. Nair, A. Nimitha, and S. M. Idicula, "Conversion of malayalam text to indian sign language using synthetic animation," in *2016 International Conference on Next Generation Intelligent Systems (ICNGIS)*, IEEE, 2016, pp. 1–4.

[11] X. Yang, X. Chen, X. Cao, S. Wei, and X. Zhang, "Chinese sign language recognition based on an optimized tree-structure framework," *IEEE journal of biomedical and health informatics*, vol. 21, no. 4, pp. 994–1004, 2016.

[12] R. Alzohairi, R. Alghonaim, W. Alshehri, and S. Aloqeely, "Image based arabic sign language recognition system," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 3, 2018.

[13] N. B. Ibrahim, M. M. Selim, and H. H. Zayed, "An automatic arabic sign language recognition system (arslrs)," *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 4, pp. 470–477, 2018.

[14] M. Kumar, "Conversion of sign language into text," *International Journal of Applied Engineering Research*, vol. 13, no. 9, pp. 7154–7161, 2018. [Online]. Available: https://www.ripublication.com/ijaer18/ijaerv13n9_90.pdf.

[15] S. Sarker and M. M. Hoque, "An intelligent system for conversion of bangla sign language into speech," in *2018 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, IEEE, 2018, pp. 513–518.

[16] M. Varghese and S. K. Nambiar, "English to sigml conversion for sign language generation," in *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, IEEE, 2018, pp. 1–6.

[17] M. Asri, Z. Ahmad, I. A. Mohtar, and S. Ibrahim, "A real time malaysian sign language detection algorithm based on yolov3," *International Journal of Recent Technology and Engineering*, vol. 8, no. 2, pp. 651–656, 2019.

[18] R. Bharti, S. Yadav, S. Gupta, *et al.*, "Automated speech to sign language conversion using google api and nlp," in *Proceedings of the International Conference on Advances in Electronics, Electrical & Computational Intelligence (ICAEEC)*, 2019.

[19] M. Brour and A. Benabbou, "Atlaslang mts 1: Arabic text language into arabic sign language machine translation system," *Procedia computer science*, vol. 148, pp. 236–245, 2019.

[20] P. Ghaste, R. Khandelwal, S. Ambapkar, Z. Ansari, *et al.*, "Sign language interpretation and conversion to text," *National Journal of Computer and Applied Science*, 2019.

[21] R. Jayapriya, P. Vijayalakshmi, *et al.*, "Development of mems sensor-based double handed gesture-to-speech conversion system," in *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, IEEE, 2019, pp. 1–6.

[22] S.-K. Ko, C. J. Kim, H. Jung, and C. Cho, "Neural sign language translation based on human keypoint estimation," *Applied sciences*, vol. 9, no. 13, p. 2683, 2019.

[23] D. N. Kumar, M. Madhukar, A. Prabhakara, A. V. Marathe, S. S. Bharadwaj, *et al.*, "Sign language to speech conversion—an assistive system for speech impaired," in *2019 1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE)*, IEEE, 2019, pp. 272–275.

[24] A. Mittal, P. Kumar, P. P. Roy, R. Balasubramanian, and B. B. Chaudhuri, "A modified lstm model for continuous sign language recognition using leap motion," *IEEE Sensors Journal*, vol. 19, no. 16, pp. 7056–7063, 2019.

[25] S. Vigneshwaran, M. S. Fathima, V. V. Sagar, and R. S. Arshika, "Hand gesture recognition and voice conversion system for dump people," in *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, IEEE, 2019, pp. 762–765.

[26] H. Wang, X. Chai, and X. Chen, "A novel sign language recognition framework using hierarchical grassmann covariance matrix," *IEEE Transactions on Multimedia*, vol. 21, no. 11, pp. 2806–2814, 2019.

[27] B. A. Dabwan, "Convolutional neural network-based sign language translation system," *International Journal of Engineering, Science and Mathematics*, vol. 9, no. 6, pp. 47–57, 2020.

[28] A. S. Dhanjal and W. Singh, "An automatic conversion of punjabi text to indian sign language," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 7, no. 28, e9–e9, 2020.

[29] M. N. Huda, I. Alam, and S. A. Siddique, "Automated bangla sign language conversion system: Present and future," 2020.

[30] P. C. Narvekar, S. Mungekar, A. Pandey, and M. Mahadik, *Sign language to speech conversion using image processing and machine learning*, Aug. 2020. [Online]. Available: https://ijtre.com/wp-content/uploads/2021/10/2020071224.pdf.

[31] A. Thakur, P. Budhathoki, S. Upreti, S. Shrestha, and S. Shakya, "Real time sign language recognition and speech generation," *Journal of Innovative Image Processing*, vol. 2, no. 2, pp. 65–76, 2020.

[32] K. Tiku, J. Maloo, A. Ramesh, and R. Indra, "Real-time conversion of sign language to text and speech," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, IEEE, 2020, pp. 346–351.

[33] A. Vidhya. "Part of speech (pos) tagging, dependency parsing, and constituency parsing in nlp." Accessed on: Insert Access Date Here. (2020), [Online]. Available: https://www.analyticsvidhya.com/blog/2020/07/part-of-speechpos-tagging-dependency-parsing-and-constituency-parsing-in-nlp/.

[34] A. N. not available, "Title not available," *Journal of Low Power Electronics and Applications*, vol. 5, no. 4, p. 83, 2021, Accessed on: Insert Access Date Here. DOI: 10.3390/jlpea5040083. [Online]. Available: https://www.mdpi.com/2504-4990/5/4/83.

[35] S. Daniels, N. Suciati, and C. Fathichah, "Indonesian sign language recognition using yolo method," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, vol. 1077, 2021, p. 012 029.

[36] Z. Alsaadi, E. Alshamani, M. Alrehaili, A. A. D. Alrashdi, S. Albelwi, and A. O. Elfaki, "A real time arabic sign language alphabets (arsla) recognition model using deep learning architecture," *Computers*, vol. 11, no. 5, p. 78, 2022.

[37] V. J. Schmalz, "Real-time italian sign language recognition with deep learning," in *CEUR Workshop Proceedings*, CEUR Workshop Proceedings, vol. 3078, 2022, pp. 45–57.

[38] S. Thakar, S. Shah, B. Shah, and A. V. Nimkar, "Sign language to text conversion in real time using transfer learning," in *2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)*, IEEE, 2022, pp. 1–5.

[39] S. Deshpande and R. Shettar, "Hand gesture recognition using mediapipe and cnn for indian sign language and conversion to speech format for indian regional languages," in *2023 7th International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, IEEE, 2023, pp. 1–7.

[40] A. Kamble, J. Musale, and R. Chalavade, "Conversion of sign language to text," May 2023, Accessed: Jan. 08, 2024. [Online]. Available: https://www.ijraset.com/research-paper/conversion-of-sign-language-to-text#introduction.

[41] Y. G. (yoav.goldberg@biu.ac.il), *Sign Language Processing — research.sign.mt*, https://research.sign.mt, [Accessed 25-May-2023].

[42] A. Kumar, A. Gupta, B. Chaurasia, and C. Mishra, "Sign language to text conversion using cnn (for deaf and mute people),"

[43] J. Singh and D. Singh, "Sign language and hand gesture recognition using machine learning techniques: A comprehensive review," *Modern Computational Techniques for Engineering Applications*, pp. 187–211,

[44] S. Sudeep, *Text to sign language conversion by using python and database of images and videos*, www.academia.edu, Accessed: Jan. 08, 2024. [Online]. Available: https://www.academia.edu/40531482/Text_to_Sign_Language_Conversion_by_Using_Python_and_Database_of_Images_and_Videos.

[45] A. information not available. "Introduction to mediapipe: A computer vision library." Accessed on: Insert Access Date Here. (Year not available), [Online]. Available: https://viso.ai/computer-vision/mediapipe/.

[46] A. information not available. "Understanding the basic cnn architecture." Accessed on: Insert Access Date Here. (Year not available), [Online]. Available: https://www.upgrad.com/blog/basic-cnn-architecture/.

[47] A. information not available. "Understanding yolov5." Accessed on: Insert Access Date Here. (Year not available), [Online]. Available: https://iq.opengenus.org/yolov5/.

[48] A. information not available. "What are the different types of sign language?" Accessed on: Insert Access Date Here. (Year not available), [Online]. Available: https://www.signsolutions.uk.com/what-are-the-different-types-of-sign-language/.

[49] DataCamp. "What is tokenization?" Accessed on: Insert Access Date Here. (Year not available), [Online]. Available: https://www.datacamp.com/blog/what-is-tokenization.

[50] Engati. "Lemmatization." Accessed on: Insert Access Date Here. (Year not available), [Online]. Available: https://www.engati.com/glossary/lemmatization.

[51] M. S. Shahab. "Yolo v8." Accessed on: Insert Access Date Here. (Year not available), [Online]. Available: https://medium.com/@muhammadshabrozshahab/yolo-v8-104f1375242c.

[52] A. Vidhya. “Yolo explained.” Accessed on: Insert Access Date Here. (Year not available), [Online]. Available: https://medium.com/analytics-vidhya/yolo-explained-5b6f4564f31.

[53] World Health Organization. “Hearing loss.” Accessed on: Insert Access Date Here. (Year not available), [Online]. Available: https://www.who.int/health-topics/hearing-loss#tab=tab_1.