2024 5th IEEE Global Conference for Advancement in Technology (GCAT)
Karnataka, India. Oct 4-6, 2024

Vad-Anuvad: Speech to Sign Language & Sign Language to Speech

Yash S. Brid, Manas P. Abhyankar, Pranav P. Nair, Divya N. Shah and Dr. Sudhir N. Dhage
Sardar Patel Institute of Technology
Mumbai, India
yash.brid@spit.ac.in,manas.abhyankar@spit.ac.in,pranav.nair@spit.ac.in, divya.shah@spit.ac.in, sudhir_dhage@spit.ac.in

2024 5th IEEE Global Conference for Advancement in Technology (GCAT) | 979-8-3503-7668-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/GCAT62922.2024.10923966

***Abstract*—A thorough explanation of a speech-to-sign language and sign language-to-speech converter system is provided in this work. Detecting and recognizing sign language, translating it to text, and then translating that text to speech are all part of the sign language-to-speech system. Our smartphone application uses the Flutter API for text-to-speech translation and YOLOv5 (You Only Look Once) as its object detection model to recognize sign language. When it came to object detection, YOLO was first evaluated with several iterations. Based on the outcomes of the model training procedure, YOLOv5 produced the most accurate findings. We created a speech-to-text model and compared its accuracy to the Flutter API for the purpose of translating speech to sign language. The Flutter API was selected as the more realistic model to translate speech to text. We found two ways to translate text to sign language: one way is to show pictures of signs, and the other is to use a SiGMl player based on HamNoSys that creates 3D models to display sign language. Our main objective was to develop an end-to-end integrated mobile application that would function as a barrier to entry for users of any technical skill level by utilizing our machine learning model to execute detections using the power of a local computer.**

***Index Terms*—YOLO, Object Detection, HamNoSys, Flutter**

I. Introduction

For individuals who are deaf or hard of hearing, communication poses significant challenges. Despite their fluency in sign language, the effectiveness of their communication depends on the listener's ability to understand the signs—a skill that most people lack. The absence of a readily available translator further complicates interactions, and relying on lip reading often results in misunderstandings. These challenges highlight the need for a more reliable and accessible communication method. In response, we have developed an innovative solution for real-time translation. Our application tries to convert speech into text and then translates that text into sign language in real-time, facilitating smoother and more accurate communication compared to traditional methods. This approach not only bridges the communication gap but also enhances the overall interaction experience for both parties.

Effective communication remains one of the most significant challenges for individuals who are deaf or hard of hearing. Although many of these individuals are proficient in sign language, their ability to communicate hinges on the other person's understanding of these signs—an ability that is not widespread among the general population. The lack of readily available translators and the high probability of miscommunication through lip reading further exacerbate these challenges. Addressing these issues necessitates innovative solutions that bridge the communication gap and facilitate smoother interactions.

In this research, we aimed to develop a comprehensive system that translates speech into sign language and vice versa in real-time. Our primary focus was on leveraging advanced technologies such as motion capture and gesture recognition to convert human sign language into audio. This capability allows a person who is unable to speak to control a system that translates their movements into spoken words, enabling them to effectively communicate with larger audiences. The system is designed with an emphasis on high efficiency and accuracy to ensure that it meets the communication needs of users.

Additionally, the application we developed converts speech into text, and subsequently into sign language, in real-time. This functionality is intended to facilitate more comfortable and accurate conversations, enhancing the overall communication experience compared to traditional methods.

The structure of this paper is as follows: In Section II, we provide an extensive literature review, where we examine various studies and methodologies pertinent to our research. Section III outlines the project's workflow, including data collection, and the training process of our machine learning model. In Section IV, we detail the development of the mobile application that incorporates our model. Finally, Section V concludes the paper, summarizing our findings and discussing potential future research directions.

II. Literature Survey

Using a Hidden Markov Model (HMM), the study by [1] focuses primarily on the recognition of English speech. The researchers looked for ways to change several settings of the HMM to improve its performance. Using the Aurora 2 English language database, the accuracy of English speech recognition was tested in four distinct noisy environments. The test results showed that the suggested method could greatly increase the English voice recognition systems' accuracy.

The challenge of translating spoken English into American Sign Language (ASL) was tackled by the authors in [2], who proposed the "Speech to Sign Language Interpreter System (SSLIS)." This system provides real-time video-based ASL translation from spoken English. It uses the Viterbi beam search to obtain the final recognition output after dynamically creating a search graph from observations. In addition to pre-recorded videos, the system uses markers and the ASL manual alphabet. Each marker corresponds to a single basic word in the SSLIS sign language database. These elements are implemented in accordance with the Signed English manual. Still, the study discovered that several factors affected the accuracy of the system significantly, suggesting that creating an audio model and building a database of dictation tasks could improve accuracy.

The goal of the work by [3] is to create a translation system made up of several modules that translates English audio to text that is parsed to produce a representation of structural grammar. This representation is analyzed using the grammar rules of Indian Sign Language (ISL), and stop words are removed from the rearranged phrase. Words in ISL cannot have conjugations, so lemmatization and stemming are used to return words to their original forms. Next, a dictionary with videos of the matching signs is compared to each word. The suggested model converts audio input into animated sign language with success. As the ISL dictionary grows, more advancements might be made to enable more thorough translations.

An additional noteworthy system that interprets American Sign Language (ASL) gestures uses an AcceleGlove and a two-link arm skeleton; it is reported in [4]. This system's instrumented part breaks down ASL gestures into discrete phoneme sequences called poses and motions. Software modules that have undergone independent training and testing on volunteers with different hand sizes and signing abilities are able to recognize these sequences. Thirty one-handed signs, representing a subset of the ASL vocabulary, were used to assess the overall sign recognizer. The system's scalability was impressive: even without retraining, its accuracy increased to 95% when the lexicon was expanded to 176 signs. In terms of classification, this is a major advancement over conventional hidden Markov models (HMMs) and neural networks (NNs).

A related method is used by [5], which makes use of flex sensors to track the movements and gestures of the wearer and computes various values according to predetermined circumstances. A Global System for Mobile Communications (GSM) module, which speaks text messages into voice, is used by this system to create and send messages. The system was developed and tested by the authors using an ASL dataset. For those who are hard of hearing, the flex sensor technology offers a useful and effective way to translate and recognize sign language in real time, improving accessibility and communication.

III. Sign Language Detection Model

A. Initial Research

Our initial approach to detect sign language in images or video frames involved utilizing YOLOv4 technology. After implementing our model with this technology we made several observations:

- **Training Time**: The model took about 4–5 hours to train on a dataset of 200 photos using the YOLOv4 framework on Google Colab, which enables GPU-accelerated training.
- **Accuracy**: The detections performed by the model were not accurate in the majority of scenarios. This inaccuracy was a significant limitation in our initial approach.
- **Processing Time**: Processing a 2 to 3-second video on a local computer took around 30 minutes, which is inefficient for real-time applications.

We switched to YOLOv5, a PyTorch-based version of the YOLO architecture that is known for its precise and lightweight models, as a result of these difficulties. This change was made with the intention of making our detection system more accurate and effective.

B. Transition to YOLOv5

The decision to adopt YOLOv5 was driven by several advantages it offers over previous versions:

- **Improved Performance**: YOLOv5 models are more accurate and faster, making them suitable for real-time applications.
- **Ease of Use**: Being based on PyTorch, YOLOv5 is easier to implement and customize, with better community support and more comprehensive documentation.
- **Efficiency**: YOLOv5's architecture is optimized for both speed and accuracy, addressing the limitations we faced with YOLOv4.

We sought to improve detection accuracy and decrease processing time by utilizing YOLOv5, making the system more useful for real-world applications. Our ability to recognize objects has significantly improved since this shift, laying the groundwork for future project advancements.

C. Data Preparation

An online resource provided labelled photos for educational reasons. More images were added to the training dataset to ensure that outliers are included in the dataset and that the model's performance is not compromised. In order to label specific areas of the supplied image—in this case, the sign language—and save the coordinates of the object in question in a suitable text file for these additional pictures, we used the LabelIMG tool. Following that, a round of image enhancement was performed, which involved recalculating the coordinates of the sign language hand and rotating, tilting, blurring, and altering the contrast of a few photos. There were a total of 1000 labeled images in this collection.

Fig. 1: Labelling images using the LabelIMG tool

D. Training with YOLOv5

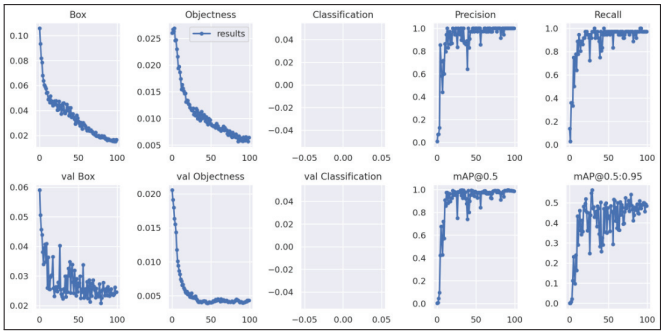

Fig. 2: Training Results - YOLOv5

To use the YOLOv5 framework to train our model, we set up the project's structure so that YOLOv5 can identify the necessary "train" and "valid" folders and access the files within them. Simultaneously, we defined the training parameters that determine the character of our training. YOLOv5 provides four distinct types of models (all pre-trained on the COCO dataset), each with its own size of neural network. We chose YOLOv5s after evaluating all of the other types of models. It has the shortest training time and retains a fair level of accuracy when completing detections. With the batch size set to 8 images and the epoch set to 125, we trained the model. The model was trained in 40 minutes with the greatest accuracy using the above configuration.

To enhance training results, YOLOv5 integrates sophisticated picture augmentations, most notably by utilizing the Mosaic Dataloader that was initially shown in YOLOv4. By combining four separate training photos into one composite image, this augmentation strategy enhances the model's exposure to a variety of circumstances in a single training instance. We made great use of this strategy in our training schedule, as Figure 4 shows. The training figures show that YOLOv5 was able to recognize targets with an astounding 97% accuracy. The model was rigorously tested in a variety of difficult settings in order to validate these findings. These included situations involving two signs, dimly light or poorly contrasted pictures, and photos with items that were rotated.

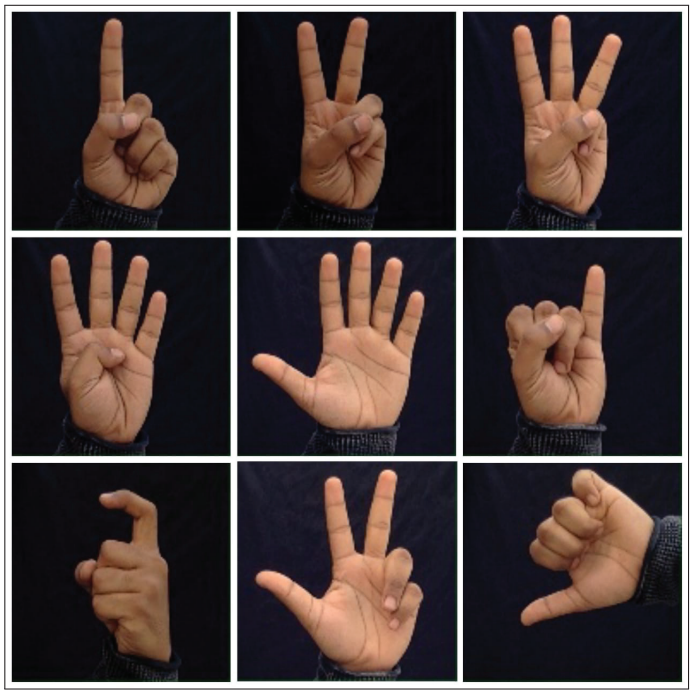

Fig. 3: Image augmentation - YOLOv5

The durability of the reported training accuracy is confirmed by the consistently strong performance in these several tests.

E. Sign Language Detection

```
0  Orientation and script detection (OSD) only.
1  Automatic page segmentation with OSD.
2  Automatic page segmentation, but no OSD, or OCR.
3  Fully automatic page segmentation, but no OSD. (Default)
4  Assume a single column of text of variable sizes.
5  Assume a single uniform block of vertically aligned text.
6  Assume a single uniform block of text.
7  Treat the image as a single text line.
8  Treat the image as a single word.
9  Treat the image as a single word in a circle.
10 Treat the image as a single character.
11 Sparse text. Find as much text as possible in no particular order.
12 Sparse text with OSD.
13 Raw line. Treat the image as a single text line, bypassing hacks that are Tesseract-specific.
```

Fig. 4: Different page segmentation modes of Tesseract

Following the extraction of the coordinates for the sign languages from the output produced by our model, we labeled the cropped image in order to translate the sign language contained therein into a string that could be placed in the application's database for later usage.

F. Implementation

The two primary functionalities of our project, "Sign Language to Speech" and "Speech to Sign," are each designed to make it easier for the deaf and hearing-impaired community to communicate. We go over each functionality's implementation procedure in detail below.

1) Sign Language to Speech: Deaf or mute people can communicate with others by using the "Sign Language to Speech" feature, which translates sign language gestures into audible speech and visual text display. Several crucial steps are involved in the implementation:

- **Video Streaming:** The application captures real-time video using the camera on a mobile device, ensuring continuous streaming of the user's gestures for analysis.
- **Gesture Detection:** We use the YOLOv5 model, a cutting-edge deep learning model renowned for its quickness and precision in object recognition. Specifically, this model is trained to recognize different gestures in sign language from the input of videos.
- **Real-time Conversion:** Following recognition of a gesture, it is converted to English text. After that, this text is sent to a specially created Flutter plugin that turns text into audible speech.
- **Display and Speech Output:** Alongside the speech output, the text is also displayed on the mobile device's screen in real-time, ensuring that the user can visually confirm the speech output.

Limitations: The current implementation is restricted to English, limiting its utility to English-speaking users or those familiar with English sign language.

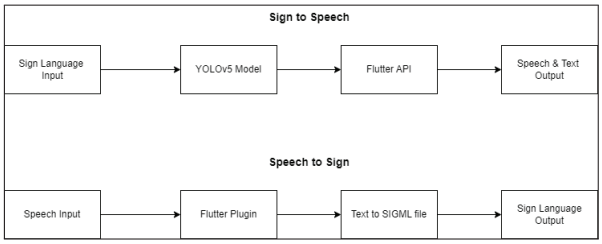

Fig. 5: Different page segmentation modes of Tesseract

2) Speech to Sign: With the help of the "Speech to Sign" feature, users who are deaf or have hearing impairments can access spoken content by translating spoken words into animated sign language. The following are the steps involved:

- **Speech Recognition:** First, we tried creating our own speech-to-text model from scratch. But because of its better accuracy and dependability, we decided to incorporate the Flutter API for speech recognition because the performance was not at its best.
- **Text Processing:** The recognized speech is then converted into text, which is subsequently tokenized to isolate individual words.
- **Animation Mapping:** For each tokenized word, the application fetches a corresponding SiGML (Signing Gesture Markup Language) file. These files contain predefined animations for various signs.
- **Animation Playback:** The fetched SiGML files are sent to a SiGML player integrated within our application, which then plays back the sign language animations corresponding to the spoken words.

Challenges and Enhancements: This phase of implementation has revealed shortcomings in current technology's ability to convert complex speech patterns into sign language precisely. Future improvements will aim to enhance the precision of gesture animations, broaden the range of signable terms, and [illegible] ds.

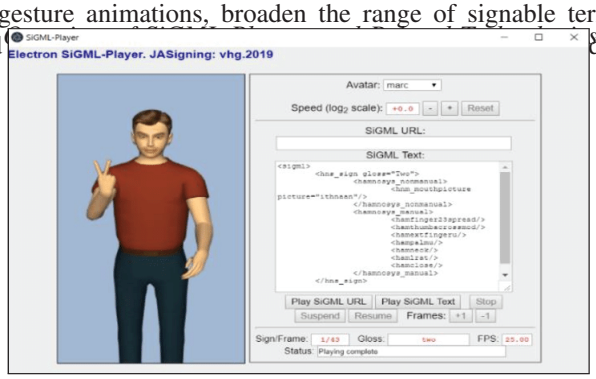

Fig. 6: Different page segmentation modes of Tesseract

SiGML (Signing Gesture Markup Language) is a specialized language designed for representing sign language gestures in a digital format. SiGML players, software tools that interpret these XML-based SiGML files, animate 3D avatars to perform the corresponding sign language gestures, offering a consistent and accurate visualization method. These players are integral in creating accessible content for deaf and hard-of-hearing individuals, enhancing communication, education, and accessibility through interactive and visual aids.

a) Core Components: include **SiGML Files**, which provide structured descriptions of sign language gestures; **3D Avatars**, virtual characters that execute these gestures; and the Animation Engine, software that animates these avatars based on the SiGML files.

b) Functionality: of SiGML players extends to rendering sign language gestures in real-time or pre-recorded formats, with customization options that allow users to alter avatar appearances and signing speeds. Interactivity is another critical feature, enabling users to input text and receive sign language gestures as immediate feedback.

c) Applications: of SiGML technology are diverse, supporting educational platforms in teaching sign language, enhancing digital content accessibility for deaf users, and facilitating communication between hearing and non-hearing individuals by translating text or speech into sign language.

d) Technical Aspects: involve the integration of SiGML players into various digital applications, adhering to standards that ensure the consistent and accurate representation of sign language. However, the **Challenges** faced include the complexity of sign language, which involves capturing nuanced and regional variations, and the realism required in animations to make the avatars' performances lifelike and understandable, necessitating advanced animation techniques.

IV. VAD-ANUVAD - MOBILE APP

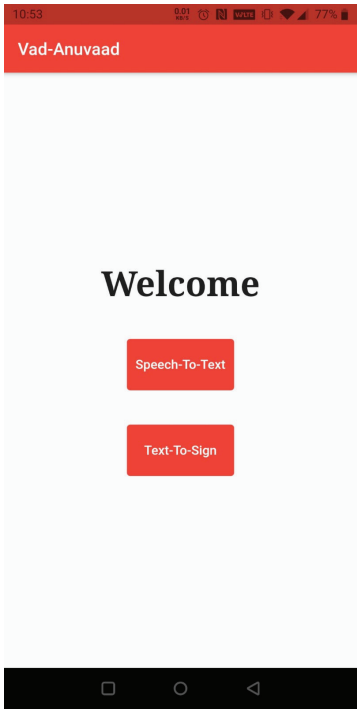

Fig. 7: Home Page

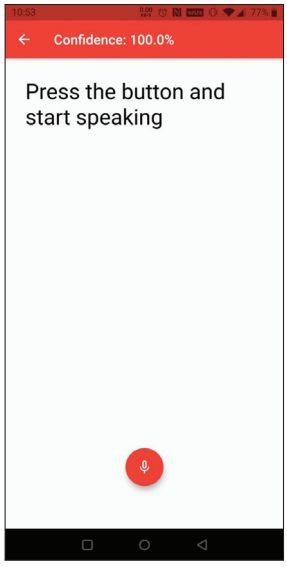

Fig. 8: Speeech to text

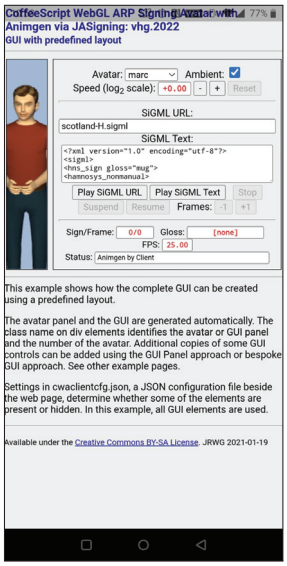

Fig. 9: Sign page

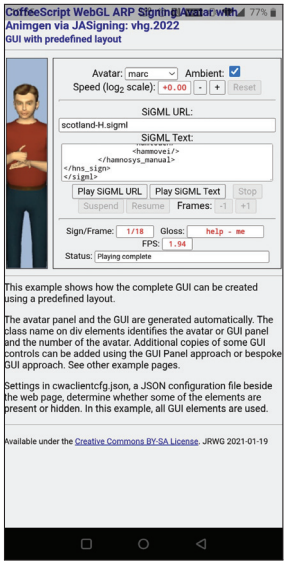

Fig. 10: Demo sign display

V. CONCLUSION

We were able to put into practice a system that receives user audio as input, uses a machine learning model to translate the audio into text, and then translates the text into sign language in real time with the least amount of delay between the user's input and the sign that is displayed.

Additionally, we have successfully constructed a system that receives user visual input as input, converts it into text using a machine learning model, and then outputs speech in real time with the least amount of delay between the user's input and the speech.

We used YOLOv5 as the foundation for our machine learning model and enhanced its implementation to meet our needs and give the user additional features. In the first phase,

we fed the model all the pictures and video frames directly, but we soon discovered that if we enhanced the images we got—for example, by cropping, adjusting the color contrast, or straightening the image—the model would have better test data to work with and the results would be more accurate.

It was also a whole new experience for us to be able to integrate the machine learning model with the front end of our application. To increase system performance and enable any user to perform object detection using available local resources, we implemented asynchronous function calls.

REFERENCES

[1] L. Cuiling, "English speech recognition method based on hidden markov model," in *2016 International Conference on Smart Grid and Electrical Automation (ICSGEA)*, pp. 94–97, 2016.

[2] K. Khalil, K. El-Darymli, O. Khalifa, and H. Enemosah, "Speech to sign language interpreter system (sslis)," 01 2006.

[3] H. Monga, J. Bhutani, M. Ahuja, N. Maid, and H. Pande, *Speech to Indian Sign Language Translator*. 12 2021.

[4] J. Hernandez-Rebollar, N. Kyriakopoulos, and R. Lindeman, "A new instrumented approach for translating american sign language into sound and text," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pp. 547–552, 2004.

[5] B. A. Manoj, "Conversion of sign language to text and speech and prediction of gesture," 2020.