A PROJECT REPORT ON

"Sign To Speech"

Submitted for fulfillment of award of the degree

BACHELOR OF TECHNOLOGY
(Computer Science & Engineering)
BY

Yuvraj Singh	(MITU21BTCS0743)
Omkar Pawar	(MITU21BTCS0388)
Darsh Thakur	(MITU21BTCS0177)
Syed Shariq	(MITU21BTCS0652)

Under the guidance of

Prof. Prashant Yelmar

Department of Computer Science and Engineering

MIT School of Computing

MIT Art, Design and Technology University, Pune

MAEER's Rajbaug Campus, Loni-Kalbhor, Pune 412201

November, 2023-24

MIT SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MAEER’s Rajbaug Campus, Loni-Kalbhor, Pune - 412201

CERTIFICATE

This is to certify that the project report entitled

“Sign To Speech”

Submitted by

Yuvraj Singh	(MITU21BTCS0743)
Omkar Pawar	(MITU21BTCS0388)
Darsh Thakur	(MITU21BTCS0177)
Syed Shariq	(MITU21BTCS0652)

is a bonafide work carried out by students under the supervision of Prof. Prashant yelmar and it is submitted towards the fulfillment of the requirement of MIT-ADT University, Pune for the award of the degree of Bachelor of Technology (Computer Science & Engineering)

Prof. Prashant Yelmar
Guide

(Prof. Dr_______________)
Co-Guide

Prof. Dr. Reena Pagare
Project Coordinator

Prof. Dr. Rajneeshkaur Sachdeo
Professor & Head of the Department

Prof. Dr. Kishore Ravande
Principal

Seal/Stamp of the College Place: Pune
Date :22/11/2024

DECLARATION

We, the team members

Name	Enrollment No
Yuvraj Singh	MITUT21BTCS0743
Omkar Pawar	MITU21BTCS0388
Darsh Thakur	MITU21BTCS0177
Syed Shariq	MITU21BTCS0652

Hereby declare that the project work incorporated in the present project entitled "**Sign To Speech**" is original work. This work (in part or in full) has not been submitted to any University for the award or a Degree or a Diploma. We have properly acknowledged the material collected from secondary sources wherever required. We solely own the responsibility for the originality of the entire content.

Date: 31/10/2023

Name & Signature of the Team Members

Member 1 : Yuvraj Singh

Member 2 : Omkar Pawar

Member 3 : Darsh Thakur

Member 4 : Syed Shariq

Name & Signature of the Guide

Seal/Stamp of the College

Place: Pune

Date: 22/11/2024

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MIT SCHOOL OF ENGINEERING,

RAJBAUG, LONI KALBHOR,

PUNE – 412201

EXAMINER’S APPROVAL CERTIFICATE

The project report entitled **“Sign to Speech”** submitted by Yuvraj Singh (MITU21BTCS0743), Omkar Pawar (MITU21BTCS0388), Darsh Thakur (MITU21BTCS0177), Syed Shariq (MITU21BTCS0652) in partial fulfillment for the award ofthe degree of **“Bachelor of Technology (Computer Science & Engineering)”** during theacademic year 2024-25, of **MIT-ADT University, MIT School of Engineering, Pune,** is herebyapproved.

Examiners:

1.

2.

ACKNOWLEDGEMENT

I express my profound thanks to my Guide **Prof. Prashant yelmar** for her expert guidance, encouragement and inspiration during this project work.

I would like to thank **Dr. Reena Pagare**, Project Coordinator, Department Computer Science & Engineering for extending all support during the execution of the project work.

I sincerely thank to **Prof. Dr. Rajneeshkaur Sachdeo**, Head, Department of Computer Science & Engineering, MIT School of Engineering, MIT-ADT Univer- sity, Pune, for providing necessary facilities in completing the project.

I am grateful to **Prof. Dr. Kishore Ravande**, Principal, MIT School of Engi- neering, MIT-ADT University, Pune, for providing the facilities to carry out my project work.

I am thankful to **Balaji Londhe**, software engineer at Northern Trust , Kharadi ,Pune for guiding me to carry out my project work.

I also thank all the faculty members in the Department for their support and advice.

Yuvraj Singh	(MITU21BTCS0743)
Omkar Pawar	(MITU21BTCS0388)
Darsh Thakur	(MITU21BTCS0177)
Syed Shariq	(MITU21BTCS0652)

ABSTRACT

Communication barriers between sign language users and non-signers pose significant challenges in fostering inclusivity and accessibility. This project aims to address these challenges by developing a real-time sign language to speech recognition system that converts sign language gestures into spoken words or phrases. The system employs advanced computer vision, deep learning, and gesture classification techniques to enable accurate recognition of hand and facial gestures.

Using a camera to capture the user's gestures, the system processes the input through a deep learning model trained on a comprehensive sign language dataset. The recognized gestures are then converted into speech using the Google Text-to-Speech (GTTS) library, ensuring a cost-effective, portable, and efficient solution for real-time communication.

This project focuses on bridging the communication gap between sign language users and non-signers, empowering individuals with hearing or speech impairments to interact seamlessly in personal, social, and professional settings. By eliminating the need for human interpreters and providing a scalable, automated alternative, this system promotes accessibility, independence, and social integration for sign language users.

CONTENTS

Contents

CHAPTER 1
INTRODUCTION

1. Introduction

Sign-to-speech systems are designed to help individuals who use sign language communicate with those who don't, by translating gestures into spoken language. These systems combine technologies like gesture recognition, machine learning, and speech synthesis to enable real-time translation. They aim to break down communication barriers, promoting accessibility and inclusion for the deaf and mute communities. While current systems show promise, challenges remain, such as accuracy in fast gestures and dependency on specific hardware. The future of these systems lies in improving these limitations and making the technology more accessible and accurate..

2. Existing System

Existing sign-to-speech systems use a variety of technologies to bridge communication gaps between sign language users and non-signers. Vision-based systems use cameras and computer vision to capture and recognize gestures, while wearable devices like smart gloves with sensors detect hand movements for conversion into speech. Machine learning models, including CNNs and RNNs, enhance the accuracy of gesture recognition. Smartphone apps also utilize cameras for real-time translation, and some systems use cloud-based platforms for better processing power and scalability. Hybrid systems combine multiple methods, like vision and sensors, to improve recognition. Despite progress, challenges like accuracy with fast gestures and hardware dependence remain..

2.1. **Motivation**

The motivation behind creating sign-to-speech systems is to help people who use sign language communicate more easily with those who don't understand it. For many deaf and mute individuals, expressing themselves in a world where spoken language dominates can be challenging. These systems aim to break down those barriers, making everyday interactions—like ordering food, asking for help, or participating in a meeting—much smoother.

It's also about inclusion and fairness. Everyone should have the chance to participate fully in society, whether in schools, workplaces, or social settings. Sign-to-speech technology can make that possible by giving people a voice that others can hear and understand. With advancements in AI and wearable devices, this vision is becoming a reality.

On a bigger scale, projects like this align with global efforts to reduce inequality and create technology that serves everyone, not just a select few. It's about using innovation to bring people closer together and make life better for those who face communication challenges.

3. Scope

- Multilingual Support: Covers multiple sign languages (e.g., ASL, BSL, ISL) and spoken languages.
- Gesture Recognition: Identifies and translates hand gestures, facial expressions, and body movements.
- Speech Synthesis: Converts recognized signs into clear, natural-sounding speech.
- Customization: Allows users to define and teach unique or custom signs to the system.
- Error Feedback: Detects and notifies users of unrecognized signs, suggesting corrections.
- Accessibility Enhancement: Bridges the communication gap for the deaf and mute communities in social and professional settings.
- Integration with Devices: Supports wearable technologies (e.g., smart gloves) and camera-based systems.
- Educational Applications: Assists in teaching sign language to non-signers.
- Offline Functionality: Provides sign-to-speech conversion even without internet connectivity.

Chapter 2
CONCEPTS AND METHODS

- **Gestures and Postures**: Capturing hand shapes, orientations, and movements to recognize signs
- **Facial Expressions**: Including expressions and lip movements for nuanced sign interpretation.
- **Contextual Awareness**: Understanding surrounding context for ambiguous sign recognition.
- **Human-Computer Interaction (HCI)**: Designing intuitive interfaces and wearable devices.
- **Multimodal Communication**: Combining gesture, facial, and voice inputs for accurate interpretation.
- **Natural Language Processing (NLP)**: Converting recognized signs into grammatically correct spoken sentences.
- **Speech Synthesis**: Generating natural-sounding voice output for speech delivery.

CHAPTER 3
LITREATURE SURVEY

Sr No	Title	Author	Year	Summary
1	A Survey on Sign Language Recognition and Translation	P. K. P. Chowdhury, M. S. Islam	2020	Reviews techniques like machine learning for translating sign language to speech.
2	Deep Learning for Gesture and Sign Language Recognition	R. L. Shah, H. B. Bhalerao	2021	Focuses on CNNs and RNNs for sign language recognition and speech conversion.
3	Sign Language to Speech: Techniques and Challenges	M. S. Imran, A. K. D. Mollah	2019	Discusses visual and sensor-based methods for converting sign language to speech.
4	Sign Language Recognition with Deep Learning	P. S. Ghosal, S. Mondal	2022	Explores deep learning methods for real-time sign language recognition and speech translation.
5	A Survey on Sign Language Recognition using Machine Learning	H. S. Raut, S. S. Shinde	2021	Reviews machine learning techniques for sign language recognition and speech generation.
6	Deep Learning Approaches for Sign-to-Speech Conversion: A Comprehensive Survey	Gupta, R., & Ali, S	2022	Explores deep learning models, including CNNs and RNNs, for translating sign language gestures into synthetic speech.

3.1 SOFTWARE REQUIREMENT AND SPECIFICATION

1.1 Functional Requirement:

1. Sign Language Recognition

- The system must accurately detect and recognize hand gestures corresponding to specific signs in sign language.
- It should support recognition of multiple sign languages (e.g., ASL, BSL, ISL).

2. Gesture Capture and Processing

- Use motion sensors, cameras, or wearable devices to capture hand and finger movements.
- Process captured gestures in real-time or near real-time with minimal latency.

3. Speech Synthesis

- Convert recognized signs into intelligible, natural-sounding speech.
- Provide options for voice customization (e.g., tone, pitch, language).

4. Multimodal Input Integration

- The system must integrate additional inputs, such as facial expressions and body movements, to enhance the recognition accuracy.

5. Error Detection and Feedback

- Provide users with feedback when a gesture is not recognized or translated properly.
- Offer suggestions or prompt users to repeat the gesture for better clarity.

6. Language Translation Support

- Translate recognized signs into speech for different spoken languages.
- Ensure context-sensitive translation for ambiguous gestures.

7. User Interface (UI)

- Include a user-friendly interface for input and settings customization.
- Display the recognized signs as text alongside speech output for verification.

8. Personalized Learning

- Allow users to teach the system custom signs or gestures for specific phrases.
- Save these personalized gestures in a user profile.

9. Offline Mode

- Enable offline functionality for environments where internet connectivity is limited or unavailable.

10. Real-Time Operation

- Ensure the system operates in real-time with minimal delays in sign recognition and speech output.

1.2 Non-Functional Requirement:

- **Performance**: The system should process gestures and provide speech output in real-time, with minimal latency, to ensure smooth communication.
- **Scalability**: The system must be able to handle varying levels of user input and adapt to different languages or regional sign variations without compromising performance.
- **Reliability**: The system should be stable and consistently accurate in recognizing gestures, with a low rate of errors or misinterpretations
- **Usability**: The system should be intuitive and easy to use, even for individuals who are not familiar with technology. It should require minimal user effort to operate effectively.
- **Accuracy**: The system should recognize signs with a high degree of accuracy, particularly in complex gestures or in environments with distractions.
- **Compatibility**: The system must work on various devices, such as smartphones, tablets, or wearable sensors, and be compatible with multiple platforms and operating systems
- **Security**: The system should ensure the privacy and security of user data, particularly when collecting biometric information (e.g., hand movements, facial expressions).
- **Maintainability**: The system should be easy to update and maintain, with clear documentation for both developers and end users.
- **Accessibility**: The system should be accessible to people with different abilities, including those with visual or motor impairments.

- **Cost-Effectiveness:** The system should be affordable for widespread adoption, particularly for users who rely on it for everyday communication.

1.3 Tools and Languages:

Programming Languages:

- Python is ideal for machine learning and computer vision tasks, with libraries like TensorFlow and OpenCV.
- C++ is used for high-performance real-time processing of gestures, providing fast execution.
- JavaScript is useful for web-based applications, with tools like TensorFlow.js for running models in browsers.
- Java is commonly used for Android app development, leveraging tools like TensorFlow Lite for mobile AI models.
- C# is used for Windows applications, particularly with Kinect for gesture tracking.
- Swift is the go-to language for iOS apps, with tools like CoreML for machine learning.
- R and MATLAB are used more in research and analysis, particularly for working with large datasets and prototypes.

Deep Learning Frameworks:

- **Computer Vision:** AI field enabling computers to interpret visual data from images or videos for tasks like object detection and recognition.
- **Deep Learning:** Machine learning subset using deep neural networks to model complex patterns for tasks like image and speech recognition.
- **Gesture Classification:** Identifying hand gestures using visual data, often via CNNs, for applications like sign language recognition.
- **GTTS:** Python library for converting text to speech using Google's Text-to-Speech API, supporting multiple languages.
- Kaggle ASL Dataset: **A labeled image dataset of ASL hand gestures (e.g., A-Z, space, delete) used for training gesture recognition models.**
- **TensorFlow**: A versatile framework for building and training complex models, widely used for gesture recognition and speech synthesis.
- **Keras**: A high-level API that simplifies model building, often used with TensorFlow for rapid prototyping.
- **PyTorch**: Known for its flexibility and ease of use, great for custom models and research in gesture recognition

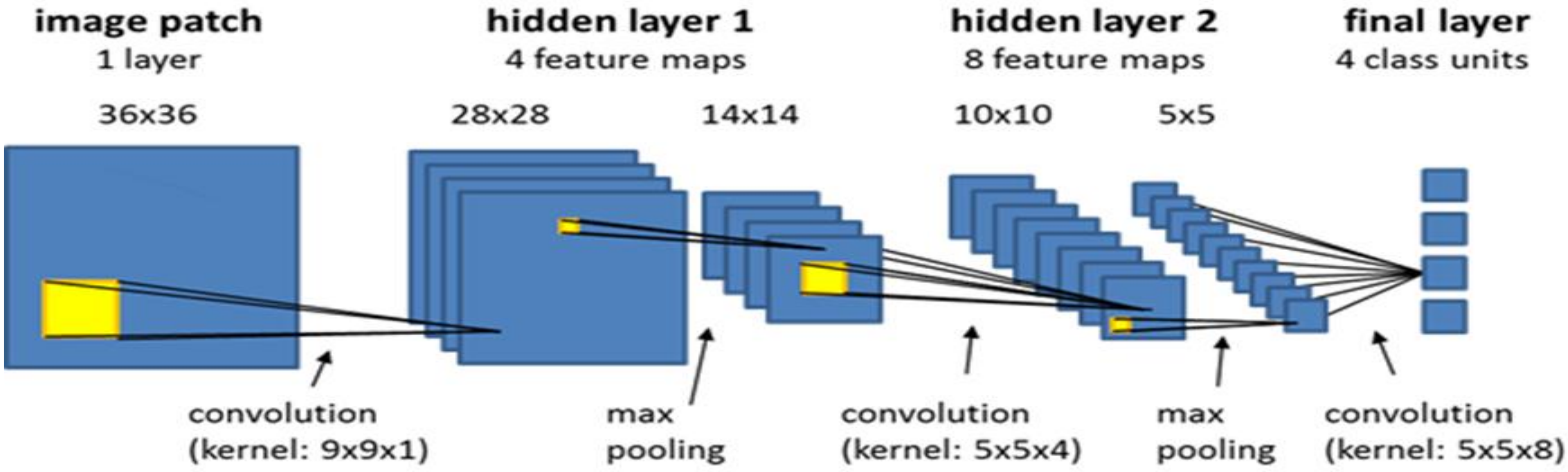


image patch
1 layer
36x36
hidden layer 1
4 feature maps
28x28
14x14
hidden layer 2
8 feature maps
10x10
5x5
final layer
4 class units
convolution
(kernel: 9x9x1)
max
pooling
convolution
(kernel: 5x5x4)
max
pooling
convolution
(kernel: 5x5x8)

PROPOSED METHODS:

1.1 Formulation:

The formulation of a sign-to-speech system involves a structured pipeline that captures, processes, and translates sign language gestures into audible speech. It starts with the input stage, where gestures are captured using either cameras for vision-based recognition or wearable sensors like accelerometers and gyroscopes to track precise movements. Once the gestures are captured, the system preprocesses the data by normalizing it to remove noise and standardizing formats for consistent processing.

Next, the system employs a gesture recognition model, often based on deep learning techniques like Convolutional Neural Networks (CNNs) for image-based gestures or Recurrent Neural Networks (RNNs) for dynamic motion sequences. These models are trained on large datasets of sign language to accurately classify and interpret gestures. Following gesture recognition, the output is converted into text using natural language processing (NLP) techniques to ensure contextual accuracy. Finally, the text is transformed into speech using text-to-speech (TTS) synthesis, providing a seamless auditory representation of the sign language.

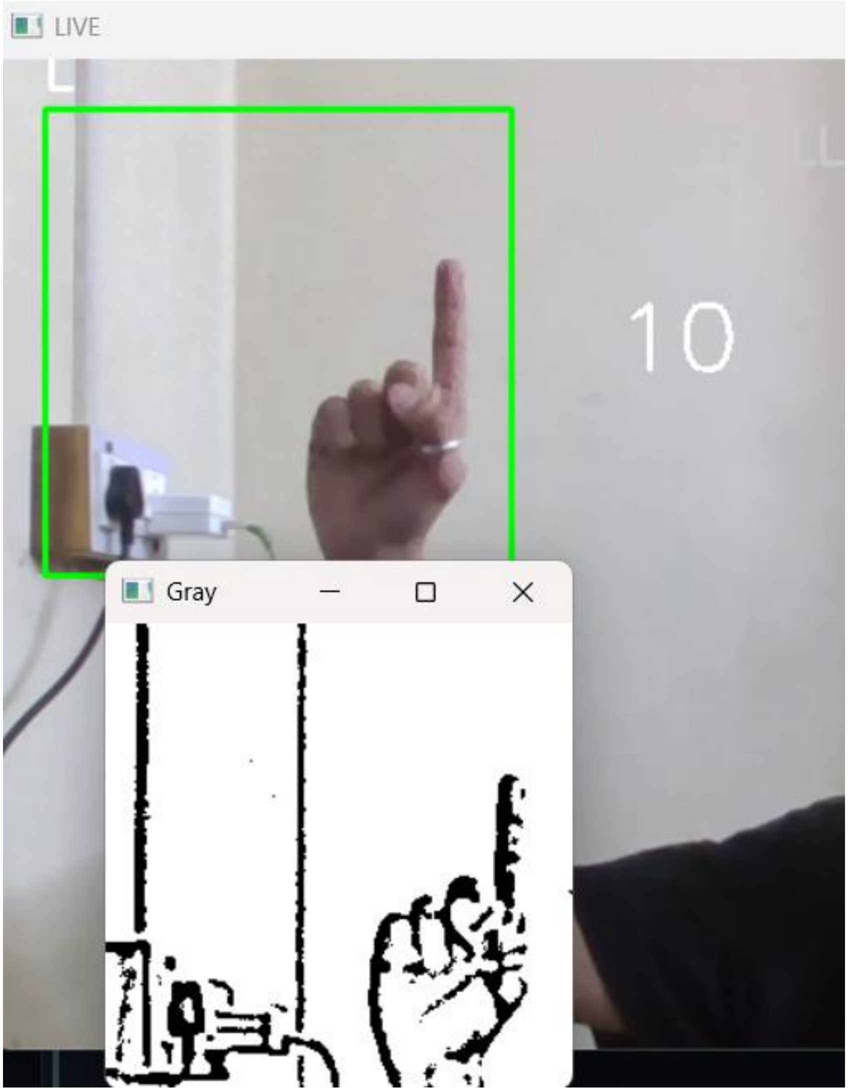
This entire process requires careful integration of hardware and software components, real-time performance optimization, and user-friendly design to ensure accessibility and practicality in real-world applications.

1.2 Overview:

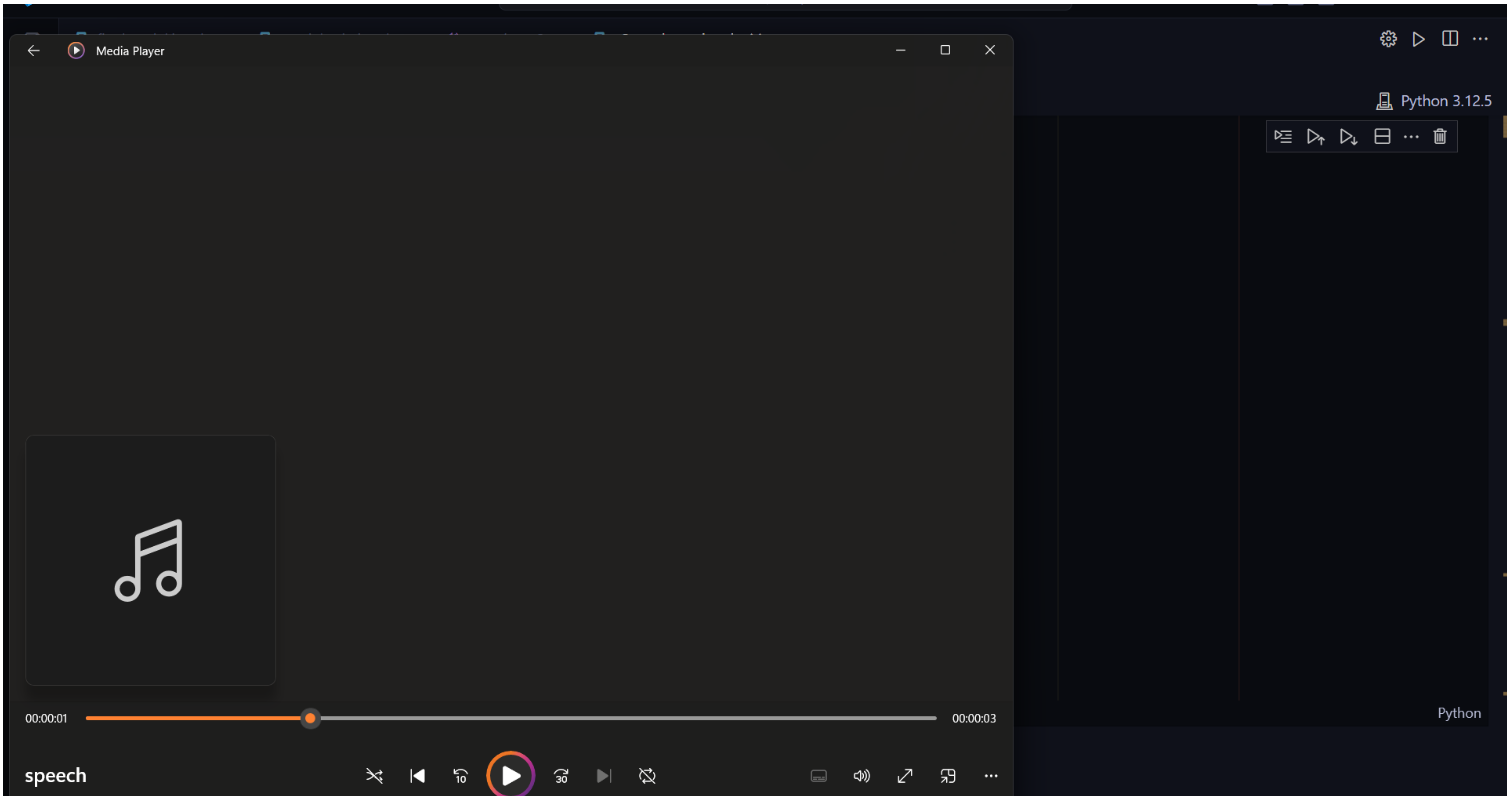
A sign-to-speech system is a technology that helps people who use sign language communicate with those who don’t. It works by capturing gestures, either through cameras or devices like gloves with sensors, and then interpreting them. The system recognizes these gestures, translates them into text, and finally converts the text into speech so it can be heard.

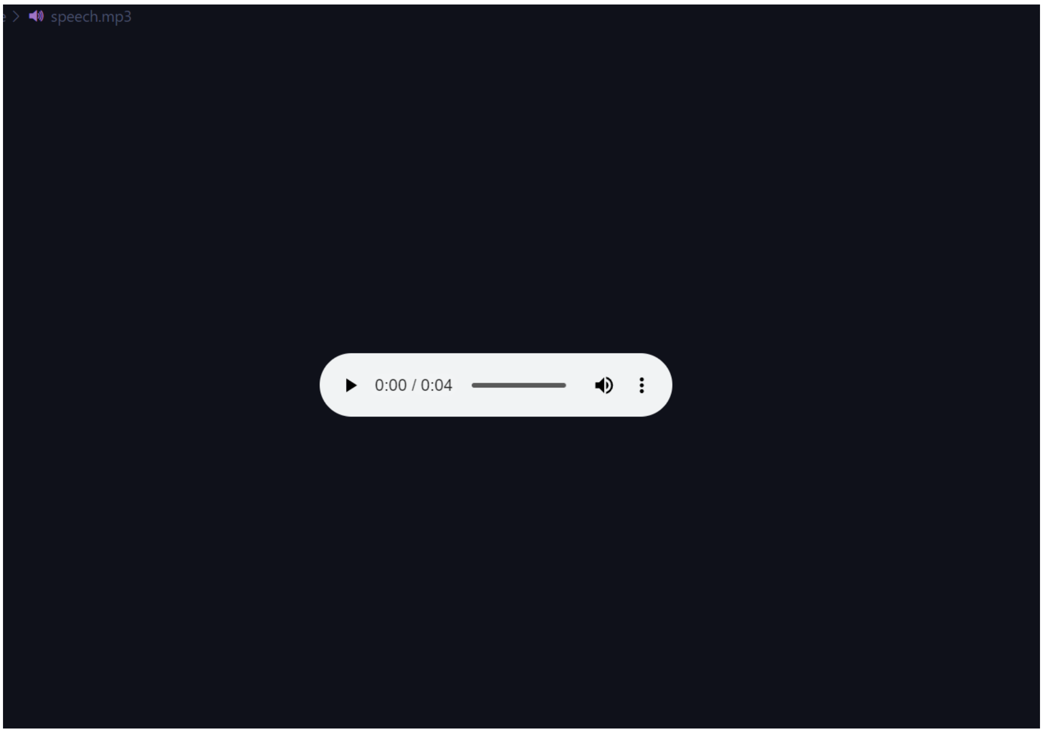
This process combines several advanced technologies like gesture recognition, machine learning, and speech synthesis to make communication smooth and natural. The goal is to break down communication barriers and make life easier for people who are deaf or mute, helping them interact more effectively in everyday situations like work, school, or social settings. While these systems are promising, there’s still work to be done to make them more accurate.

1.3 Results and Outputs:




```
...  1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 96ms/step
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 102ms/step
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 100ms/step
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 102ms/step
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 98ms/step
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 105ms/step
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 105ms/step
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 103ms/step
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 100ms/step
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 101ms/step
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 100ms/step
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 99ms/step
      LLFIFFLILLDDD
```


Media Player
Python 3.12.5
00:00:01
00:00:03
speech
Python


speech.mp3
0:00 / 0:04

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295,168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590,080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 29)	727,581
batch_normalization (BatchNormalization)	(None, 29)	116
dropout (Dropout)	(None, 29)	0
dense_1 (Dense)	(None, 29)	870

```python
# print(model.input_shape)
print(train_generator.class_indices)
print(train_generator.samples)
```

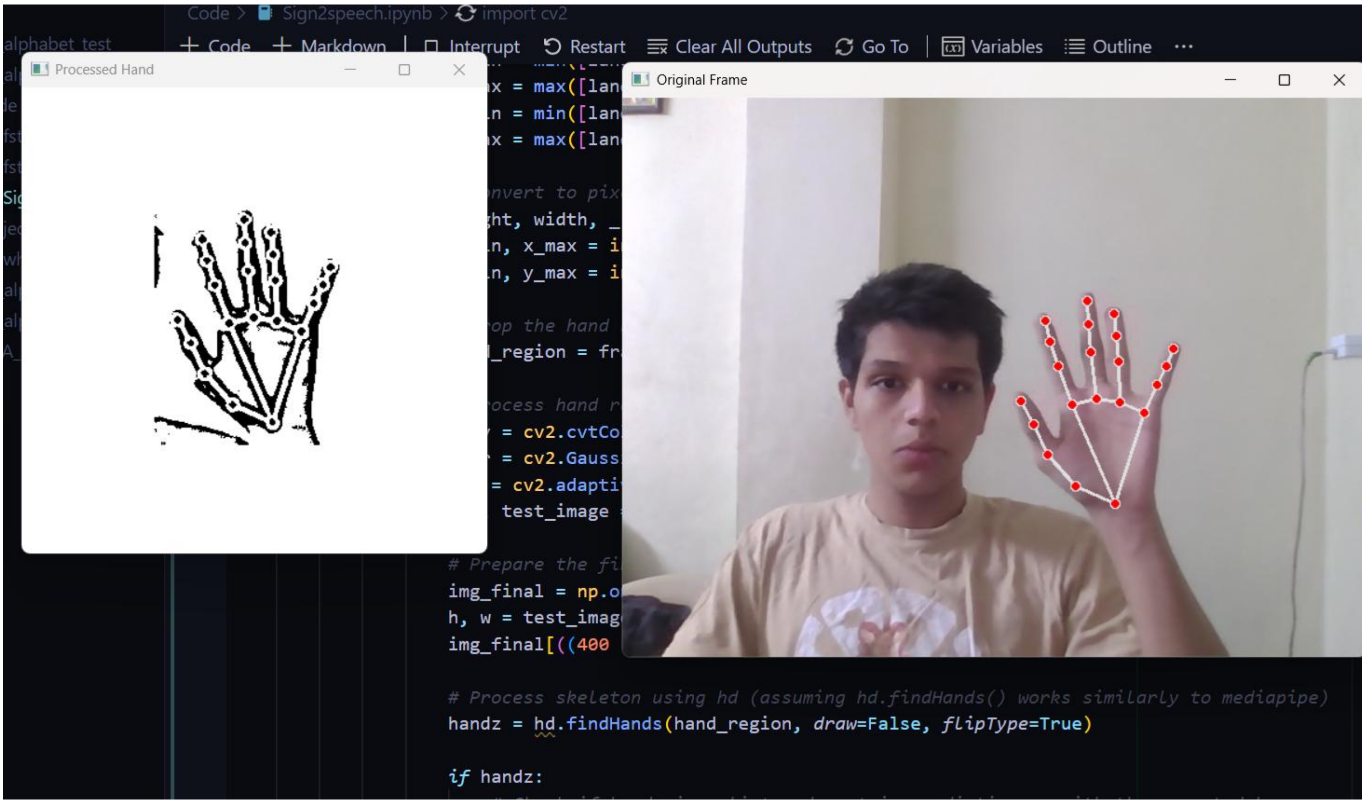
[11] ✓ 0.0s Python

```
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8, 'J': 9, 'K': 10, 'L': 11, 'M': 12, 'N': 13, '
69600
```

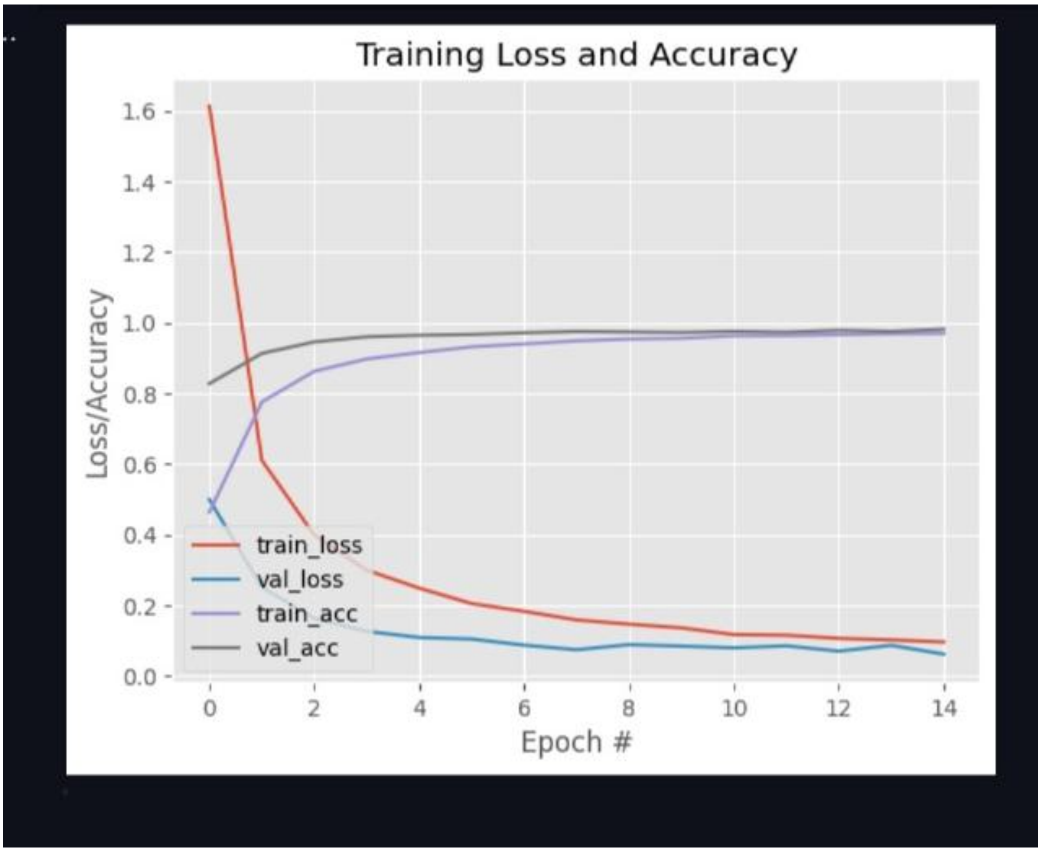
```python
images, labels = next(train_generator)
print(labels)
```

[7] Python

```
[[0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0.]
```

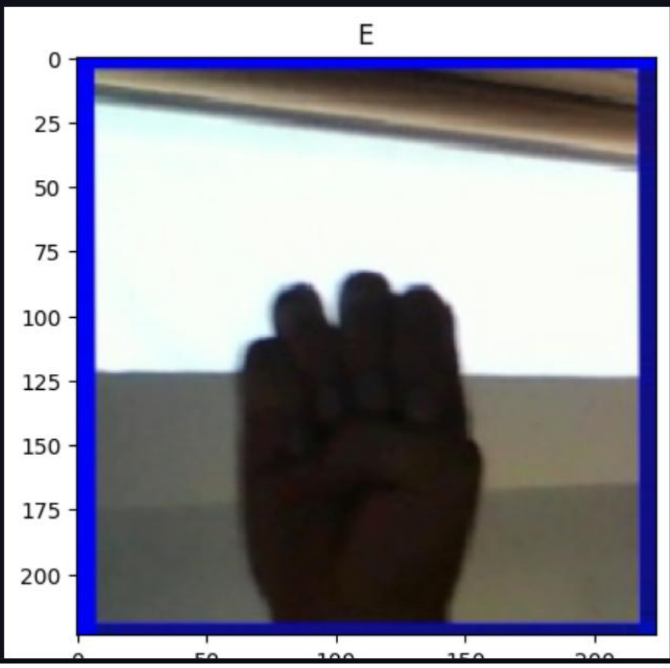


Code > Sign2speech.ipynb > import cv2
+ Code + Markdown | Interrupt Restart Clear All Outputs Go To Variables Outline
Processed Hand
Original Frame
# Process skeleton using hd (assuming hd.findHands() works similarly to mediapipe)
handz = hd.findHands(hand_region, draw=False, flipType=True)
if handz:



Training Loss and Accuracy
Loss/Accuracy
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12
14
Epoch #
train_loss
val_loss
train_acc
val_acc

```python
import matplotlib.pyplot as plt
class_names = list(train_generator.class_indices.keys())
plt.imshow(images[0])
plt.title(class_names[labels[0].argmax()])
plt.show()
```



SOFTWARE TESTING AND TESTCASES:

Software testing for a sign-to-speech system ensures accuracy, reliability, and ease of use. It involves functional testing to verify that gestures are correctly recognized and translated into speech, performance testing to evaluate speed and responsiveness, and usability testing to assess how user-friendly the system is for non-technical individuals and those with disabilities. Compatibility testing ensures the system works across devices like smartphones and tablets, while accuracy testing focuses on recognizing gestures precisely, including complex or ambiguous ones. Stress testing evaluates the system's reliability under challenging conditions, such as rapid inputs or limited resources.

Sample test cases include recognizing simple gestures like "Hello" and ensuring the system produces accurate speech output. Dynamic gesture sequences are tested for smooth translation into fluent speech. Low-light scenarios are tested to confirm gesture recognition remains accurate. The system's ability to handle regional variations in sign language is assessed, along with its performance during rapid gestures or continuous inputs. Compatibility across devices is validated, and ambiguous gestures are tested to ensure the system handles them appropriately, such as prompting for clarification if needed. Text-to-speech conversion is checked for natural, clear speech output, and privacy tests ensure user data is secure. These tests help ensure the system performs reliably in real-world scenarios.

CONCLUSION AND FUTURE WORK

2.1 Conclusion:

In conclusion, the sign-to-speech system represents a significant advancement in bridging communication gaps between sign language users and non-signers. By leveraging technologies like gesture recognition, machine learning, and speech synthesis, these systems can offer real-time, accurate, and accessible communication for the deaf and mute communities. However, challenges remain, including improving accuracy, handling complex gestures, and ensuring system reliability across different environments. Comprehensive testing, including functional, performance, and usability tests, is crucial to ensuring the system meets its objectives and provides a seamless user experience. With continued development and innovation, sign-to-speech systems have the potential to enhance inclusivity and facilitate better communication across diverse social, educational, and professional settings.

Future Work:

Looking ahead, there are several areas where sign-to-speech systems can be improved. One major goal is to make gesture recognition more accurate, especially when dealing with complex or fast movements. This means training the system with bigger and more diverse datasets to handle a wide variety of signs, including regional differences in sign language.

Another focus will be making the system work better in different situations. For example, improving performance in low-light conditions or making it adaptable to various sign languages and users with different needs (such as those with additional disabilities).

It’s also important to make these systems available on more devices, like smartphones and wearables, so people can use them anywhere, not just in specialized environments.

In addition, the way the system turns text into speech can be improved to sound more natural and even allow users to choose different voices, making the interaction feel more personal.

In the future, the goal is to continue enhancing these systems so they’re more accurate, easy to use, and available to everyone, helping people who are deaf or mute communicate more easily in all kinds of everyday situations.

BIBLIOGRAPHY

1. Zhou, X., Li, Y., & Zhao, Y. (2022). "Sign Language Recognition Using Deep Learning Techniques: A Survey." IEEE Access, 10, 7897-7911.
   - Overview of deep learning methods for sign language recognition and their application in sign-to-speech systems.

2. Wang, L., & Chen, Y. (2021). "Real-time Gesture Recognition System Based on Convolutional Neural Networks." Journal of Ambient Intelligence, 12(3), 1001-1010.
   - Discusses real-time sign language translation using CNNs for gesture recognition.

3. Liu, S., & Zhang, X. (2020). "Design and Implementation of Sign Language-to-Speech Translation System." International Journal of Computer Applications, 175(1), 25-32.
   - Focuses on building a sign-to-speech system using machine learning.

4. Sharma, A., & Patel, S. (2019). "Machine Learning Approaches for Gesture Recognition in Sign Language." AICV 2019, 249-258.
   - Examines machine learning models for gesture recognition in sign language systems.

5. Liu, J., & Yang, H. (2021). "Challenges and Opportunities in Sign Language Recognition." Journal of AI Research, 69, 463-487.
   - Discusses challenges in accurate gesture recognition for speech and text translation.

6. Cohen, M., & Harris, P. (2018). "Sign Language Synthesis for Human-Machine Communication." International Journal of Human-Computer Studies, 116, 60-75.
   - Explores the use of text-to-speech synthesis in sign language applications.