Sign Language to Text Converter

Presented By:

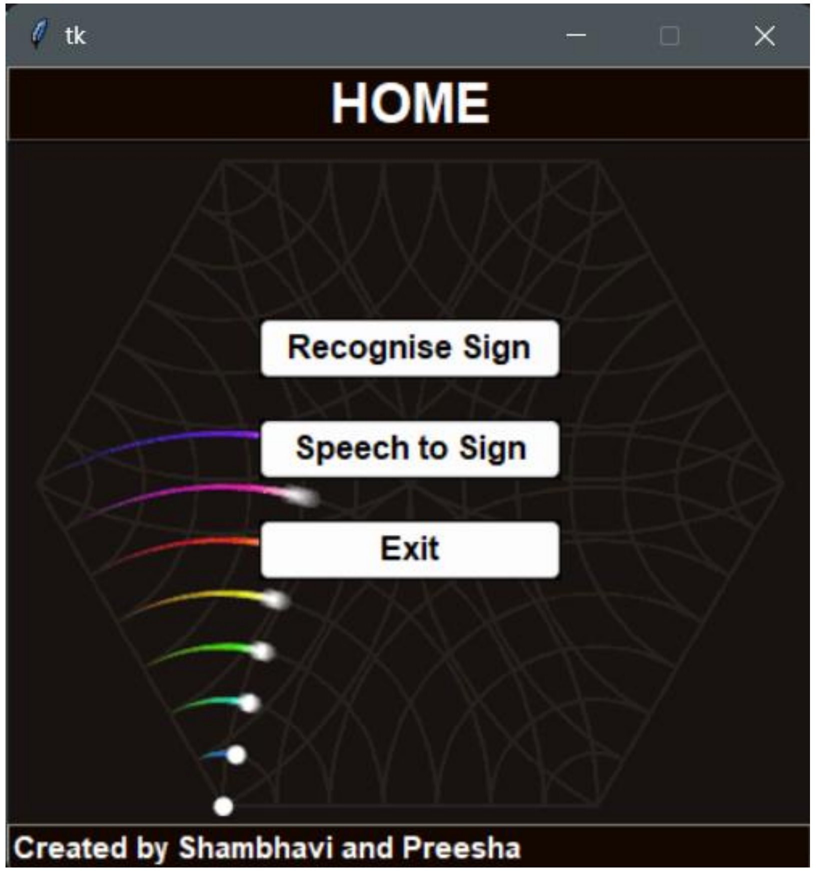
Shambhavi Shinde

Preesha Sheth

Content

Problem Statement

Develop an AI Agent capable of converting Indian sign language to text. This aim is to bridge communication gaps for those who use sign language, allowing them to interact more seamlessly with individuals and systems that do not understand sign language.

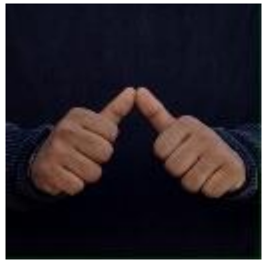


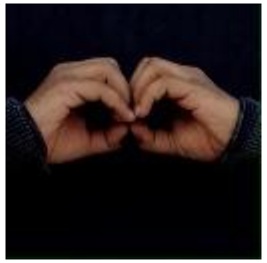
Objectives

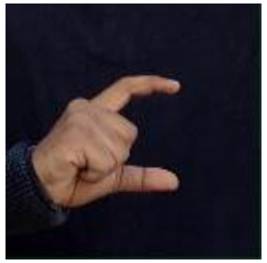
- Design a system that accurately recognizes and interprets various sign language gestures.
- Develop algorithms to convert these gestures into corresponding text outputs.

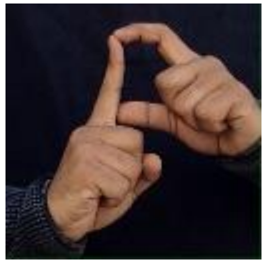
Assumptions

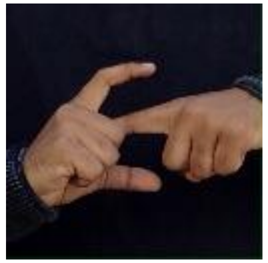
- The sign language gestures will be within the view of the agent’s sensors.
- Users will perform signs in a clear and discernible manner.
- The environment will have sufficient lighting for the agent’s visual sensors to function effectively.


A


B


C


D


E

Agent Introduction

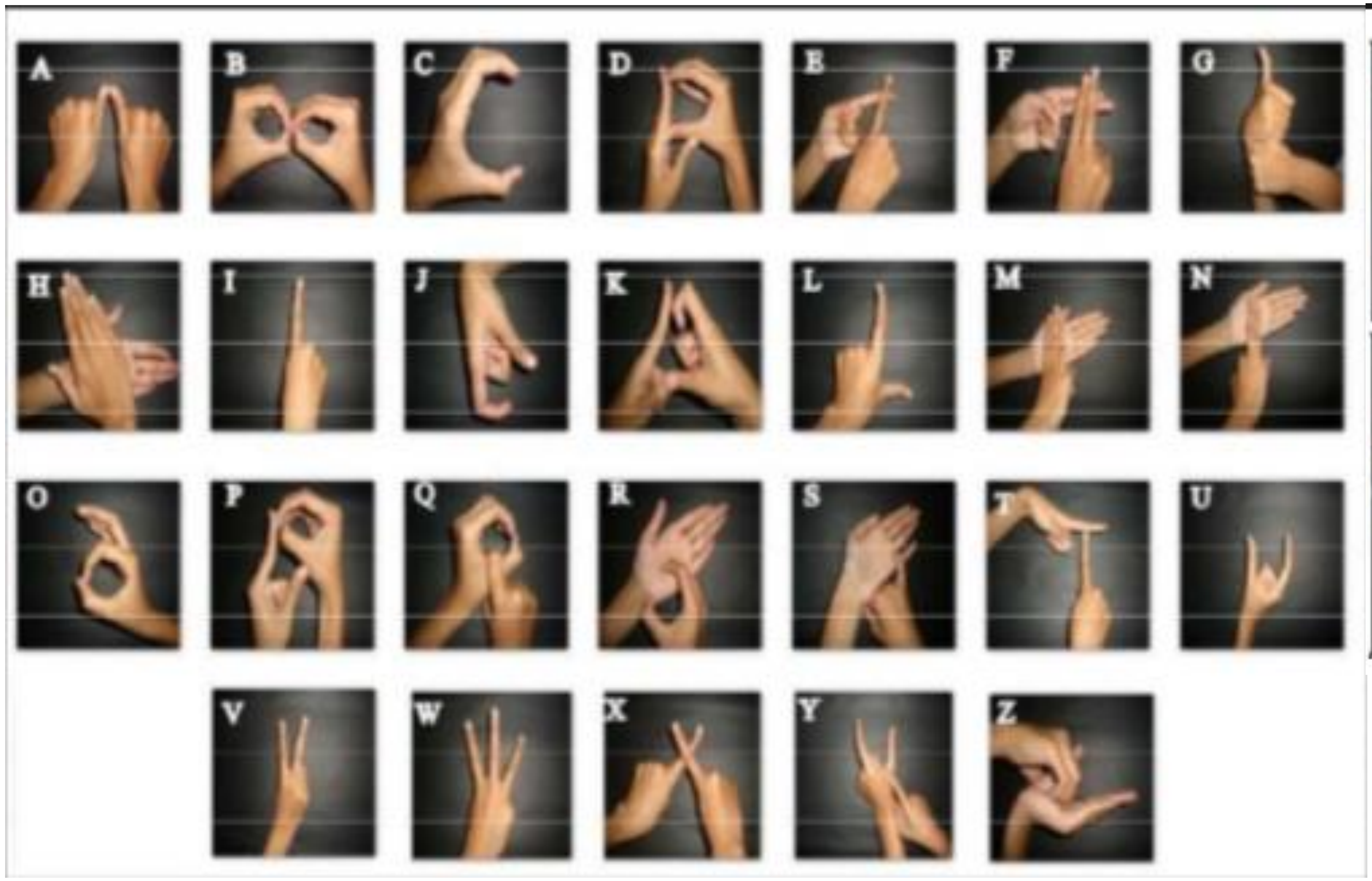
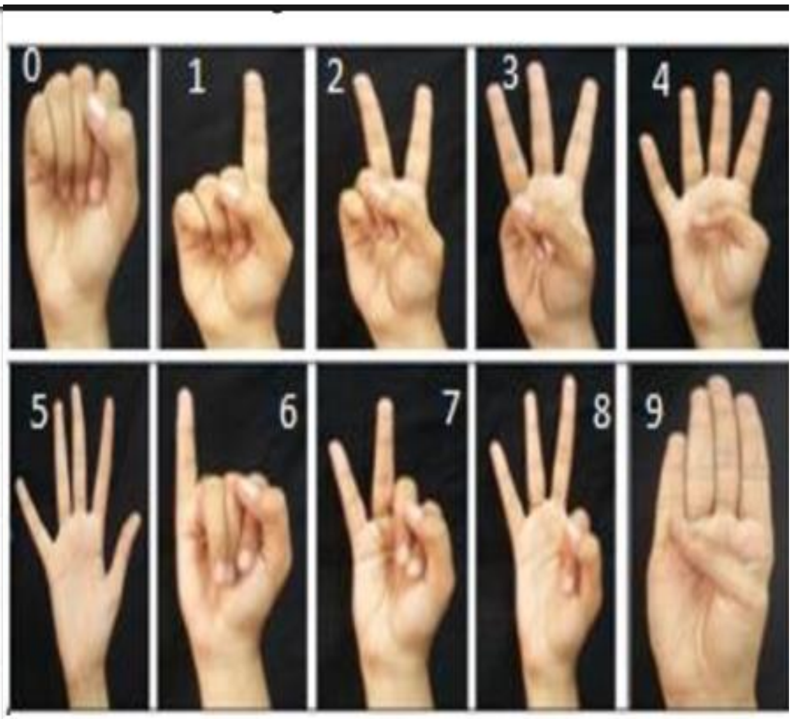
- Supervised-Learning based agent
- Visual and auditory sensors to detect and interpret sign language.
- It uses advanced machine learning models to recognize and translate sign language gestures into text , assisting in real-time communication.

Input

- Visual input of sign language hand gestures using a camera.
- Audio input for feedback and commands, using a microphone.

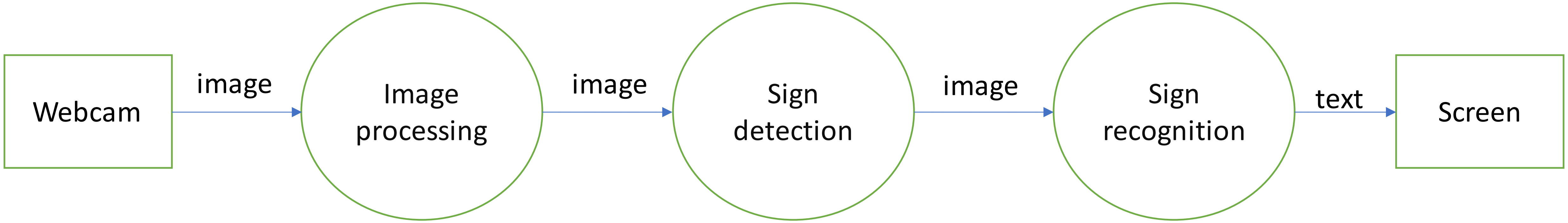
- **DATASET**

Data set consisting of 26 alphabets of Indian Sign Language and 9 numbers. Data set consists of 1200 images for each character.


A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

0
1
2
3
4
5
6
7
8
9

Output

- Text displayed on a screen or device.



Components Used

- Camera for visual gesture recognition.
- Microphone for audio input.
- Display screen for text output.
- Processing unit equipped with AI algorithms.

P.E.A.S.

- **Performance measure:**
  - Accuracy: High accuracy in gesture recognition and translation.
  - Speed: Real-time processing and response to ensure smooth communication.
  - User Experience: Intuitive and user-friendly.
- **Environment:**
  - Indoor Use: Optimized for controlled indoor environments like offices, schools, or homes.
  - Variable Conditions: Capable of functioning in different lighting and minor background noise conditions.
  - Interaction: Designed to be used in one-on-one or small group interactions.

P.E.A.S.

- **Actuators:**
  - Visual Display: To show the translated text.
- **Sensors:**
  - Camera: High-resolution camera for capturing sign language gestures.
  - Microphone: For receiving auditory commands or feedback.

AI Module Used

- **Module 1: Introduction to AI**

The foundational understanding for conceptualizing/framing the problem .

- **Module 6: Reasoning Under Uncertainty**

For handling the uncertainties and variabilities in sign language recognition, such as different lighting conditions, hand shapes, and signing style.

- **Module 8: Learning from Examples**

Training the agent using machine learning models (like CNNs for gesture recognition) using examples of sign language.

- **Module 9: AI Applications and Ethics**

An AI application for assistive technology requires careful consideration of ethical implications and user-centric design.

Algorithm

- **Convolutional Neural Network (CNN):**
  - Used for processing visual input.
  - CNNs excel at capturing spatial hierarchies in images, making them ideal for recognizing sign language gestures.
  - Training involves a large dataset of sign language gestures

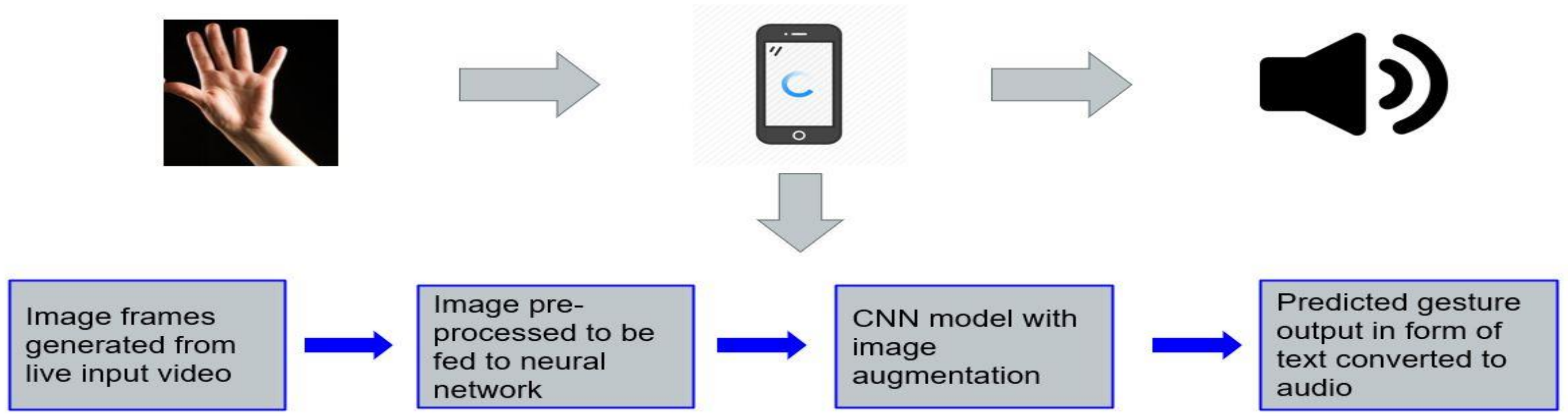
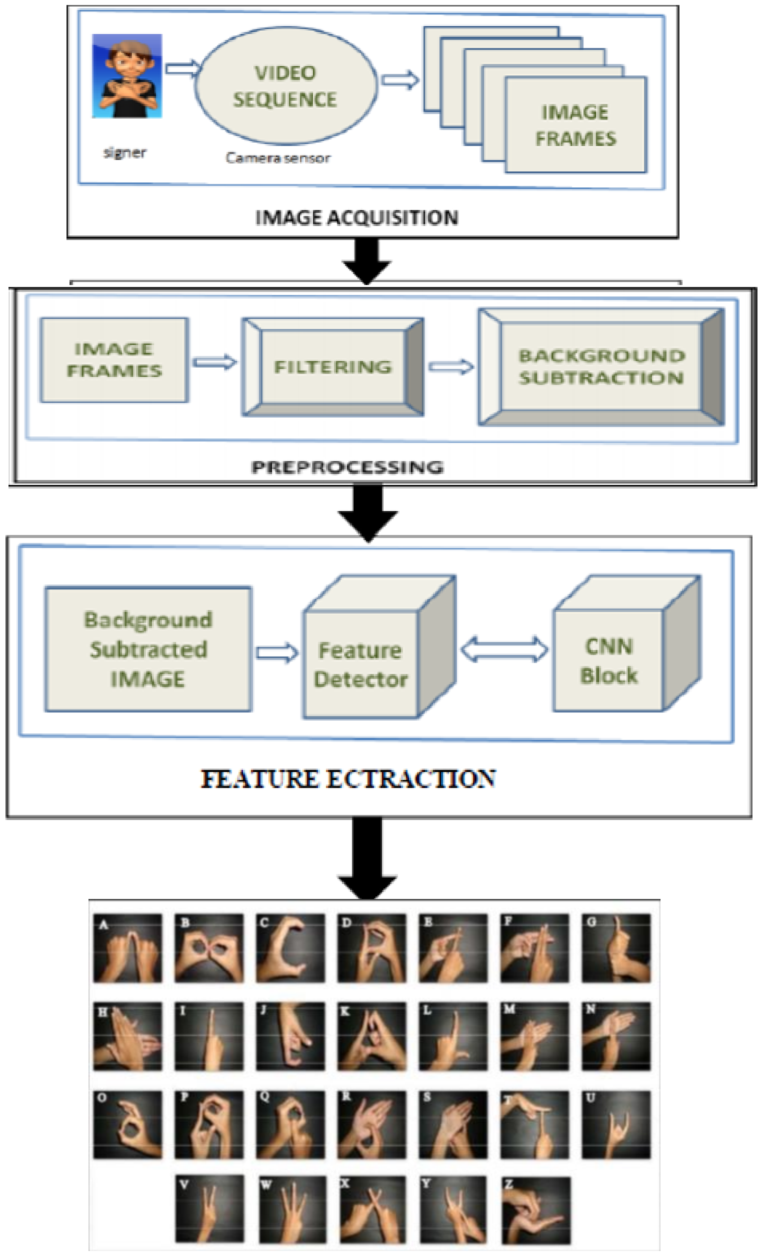


Image frames generated from live input video
Image pre-processed to be fed to neural network
CNN model with image augmentation
Predicted gesture output in form of text converted to audio

Flowchart

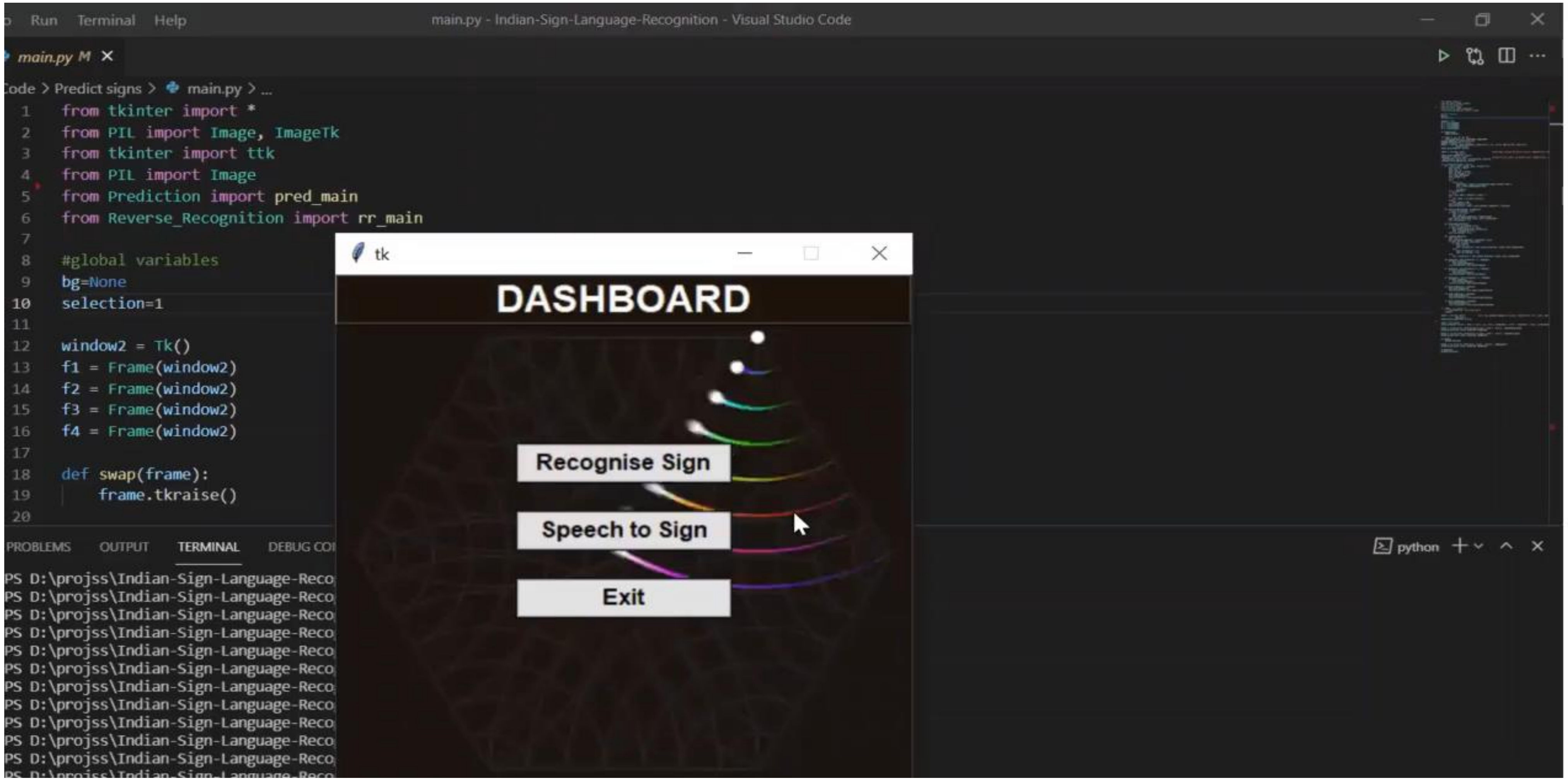


signer
VIDEO SEQUENCE
Camera sensor
IMAGE FRAMES
IMAGE ACQUISITION
IMAGE FRAMES
FILTERING
BACKGROUND SUBTRACTION
PREPROCESSING
Background Subtracted IMAGE
Feature Detector
CNN Block
FEATURE ECTRACTION
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

LIBRARIES USED

- **cv2**: Used for image processing and capturing video from the camera.
- **imutils**: Used for resizing the video frames.
- **numpy**: Used for numerical operations on the image data.
- **os**: Used for file and directory operations.
- **pickle**: Used for loading the pre-trained CNN model.
- **pyttsx3**: Used for text-to-speech conversion.
- **tensorflow and keras**: Used for making predictions with the loaded CNN model.
- **threading**: Used for running the text-to-speech functionality in a separate thread.
- **tkinter**: Used for displaying error messages.
- **pySpellChecker**: Used for correcting misspelled words in the recognized text.

Working Video



main.py - Indian-Sign-Language-Recognition - Visual Studio Code

```
from tkinter import *
from PIL import Image, ImageTk
from tkinter import ttk
from PIL import Image
from Prediction import pred_main
from Reverse_Recognition import rr_main

#global variables
bg=None
selection=1

window2 = Tk()
f1 = Frame(window2)
f2 = Frame(window2)
f3 = Frame(window2)
f4 = Frame(window2)

def swap(frame):
    frame.tkraise()
```

```
PS D:\projss\Indian-Sign-Language-Reco
PS D:\projss\Indian-Sign-Language-Reco
PS D:\projss\Indian-Sign-Language-Reco
PS D:\projss\Indian-Sign-Language-Reco
PS D:\projss\Indian-Sign-Language-Reco
PS D:\projss\Indian-Sign-Language-Reco
PS D:\projss\Indian-Sign-Language-Reco
PS D:\projss\Indian-Sign-Language-Reco
PS D:\projss\Indian-Sign-Language-Reco
PS D:\projss\Indian-Sign-Language-Reco
PS D:\projss\Indian-Sign-Language-Reco
```

Our learning

- **Figuring Out Gesture Recognition:**

Learning how to make a computer recognize sign language using CNNs.

- **Translating Signs to Words:**

Tackling the challenge of turning sign language gestures into understandable text using Machine learning model.

- **Speed is Key:**

Realizing the importance of making everything work fast enough for real-time conversation.

REFERENCES

- **https://www.kaggle.com/datasets/soumyakushwaha/indian-sign-language-dataset**