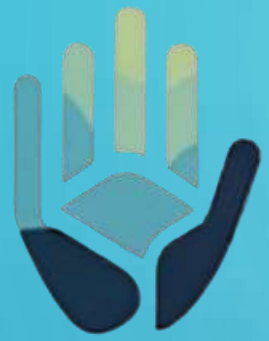


TalkPalm

MODULAR SIGN LANGUAGE TO SPEECH CONVERTER

- BIBHOR ROY

TalkPalm

INTRODUCTION:

- The primary aim of this project is to design a smart assistive technology for people with speech disability. Our project focuses on addressing the problems faced by the dumb(more specifically people who have speech disability) in conversing with other people with the help of sign language. Our prototype acts as an efficient and functional way to convert sign language into text. The text is further converted to speech so that other people can understand it easily. The respective hand signs are taken from webcam feed. Our prototype recognizes which letter the sign stands for and generates a word from it. Then it joins the letters to form sentences.

TalkPalm

WORKING PROCEDURE:

- Video Capture via webcam
- Preprocessing with OpenCV
- Hand Gesture classification
- Prediction Buffering
- Output Generation

TalkPalm

STEPS INVOLVED IN CREATING OUR PROTOTYPE:

- Training of model using Google Teachable Machine
- Import keras model and labels into python program
- Gesture recognition using OpenCV, keras and TensorFlow
- Majority voting for stability
- Text to speech using threading and PowerShell
- Displaying the word and sentences on window

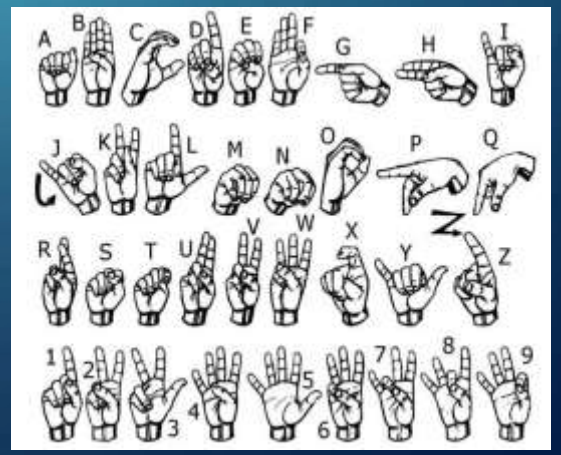
PYTHON LIBRARIES USED IN OUR PROTOTYPE:

- **TensorFlow**: For machine learning and deep learning applications.
- **NumPy**: For solving complicated mathematical equations involved in training the model
- **OpenCV**: For Computer vision and Gesture recognition.
- **Keras**: To efficiently integrate teachable machine model into our code.
- **Threading**: To handle the text to speech efficiently while the code runs.

TalkPalm

TRAINING OF AI MODEL USING GOOGLE TEACHABLE MACHINE:

- Google Teachable machine is a deep learning and neural network framework which helped us train the gesture recognition model easily. We trained the model to identify letters based on their respective hand signs. Over 200 images were used to train each later for utmost accuracy. Samples used are attached below. We could only train 5 letters (A,B,C,D,E).



TalkPalm

IMPORTING KERAS MODEL INTO THE PYTHON PROGRAM:

- The keras model was imported from teachable machine in h5 format containing all configurations and details of the training data.
- Parsing was also done to separate index from letters. Such as “0 A” to “A”

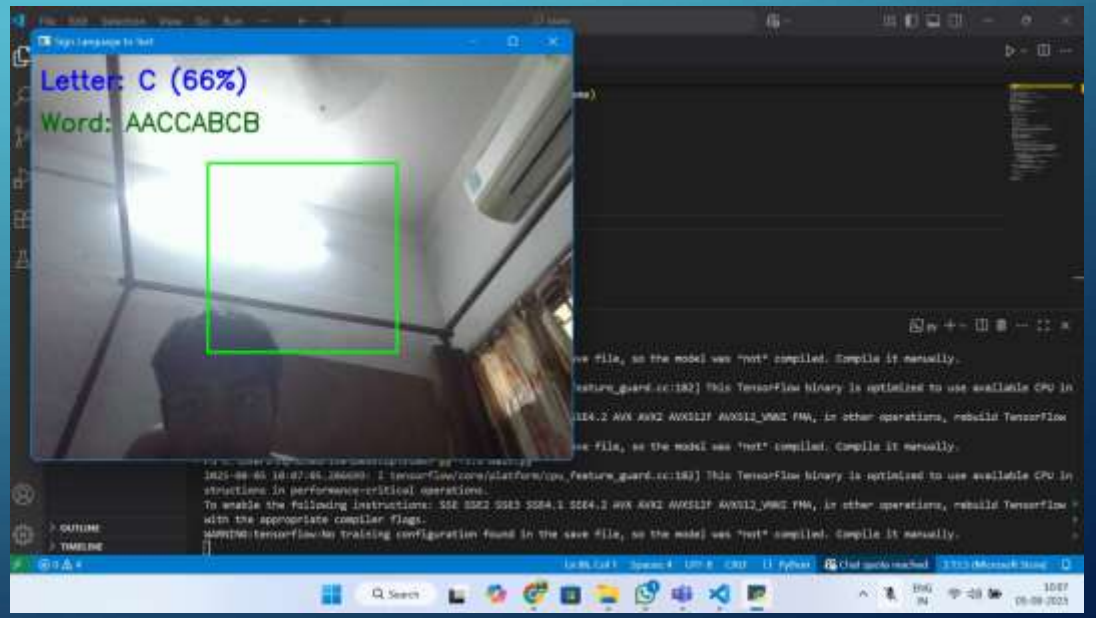
```
BASE_DIR = os.path.dirname(os.path.abspath(file)) MODEL_PATH = os.path.join(BASE_DIR,
    "keras_model.h5") LABELS_PATH = os.path.join(BASE_DIR, "labels.txt

model = keras.models.load_model(MODEL_PATH)

with open(LABELS_PATH, "r", encoding="utf-8") as f: raw_labels = [line.strip() for line in f
    if line.strip()]

def parse_label(line: str) -> str: parts = line.split() return " ".join(parts[1:]) if len
    (parts) > 1 and parts[0].isdigit() else line

class_names = [parse_label(line) for line in raw_labels] print(f"Loaded model and {len
    (class_names)} labels")
```

TalkPalm

GESTURE RECOGNITION USING OPENCV, TENSORFLOW AND KERAS:

- ROI(Region of Interest): 224x224 (same format as teachable machine for effective image extraction and higher accuracy.)
- If confidence score is less than 0.6, then it displays that no hand is detected.

```
ROI box (centered 224x224 inside 640x480)

x, y, w, h = 208, 128, 224, 224

try: while True: start_time = time.time()
ret, frame = camera.read()
    if not ret:
        continue

    frame = cv2.flip(frame, 1)
    display_frame = frame.copy()

    roi = frame[y:y+h, x:x+w]
    cv2.rectangle(display_frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

    resized = cv2.resize(roi, (224, 224), interpolation=cv2.INTER_AREA)
    input_image = np.asarray(resized, dtype=np.float32).reshape(1, 224, 224, 3)
    input_image = (input_image / 127.5) - 1.0

    prediction = model.predict(input_image, verbose=0)
    index = int(np.argmax(prediction))
    class_name = class_names[index] if 0 <= index < len(class_names) else "Unknown"
    confidence_score = float(prediction[0][index])

    # ------ NEW CODE: Detect "no hand" ------
    if class_name.lower() in ["nothing", "no hand", "none"] or confidence_score < 0.6:
        class_name = "No hand detected"
```



TalkPalm

TEXT TO SPEECH USING POWERSHELL AND THREADING

- The following code was to convert the text to speech.

```
def speak_word_threaded(word: str): def _speak():
    safe_word = word.replace("'", "''") command = ( "powershell -Command " ""Add-Typ
        -AssemblyName System.Speech; " "$s=New-Object System.Speech.Synthesis
        .SpeechSynthesizer; " f"$s.Speak('{safe_word}')"" ) os.system(command) threading
        .Thread(target=_speak, daemon=True).start()
```

TalkPalm

DISPLAYING THE WORDS AND SENTENCES:

- The following code was used to display the generated words and sentences.

```
if top_prediction == "No hand detected":
            cv2.putText(display_frame, "Sentence: No hand detected", (10, 40),
                        cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
        else:
            label_text = f"Letter: {top_prediction} ({round(avg_conf*100)}%)"
            cv2.putText(display_frame, label_text, (10, 40),
                        cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2)

        current_time = time.time()
        if avg_conf > confidence_threshold and (current_time - last_add_time) >= interval:
            if top_prediction.lower() != "no hand detected":
                predicted_letters
```

TalkPalm

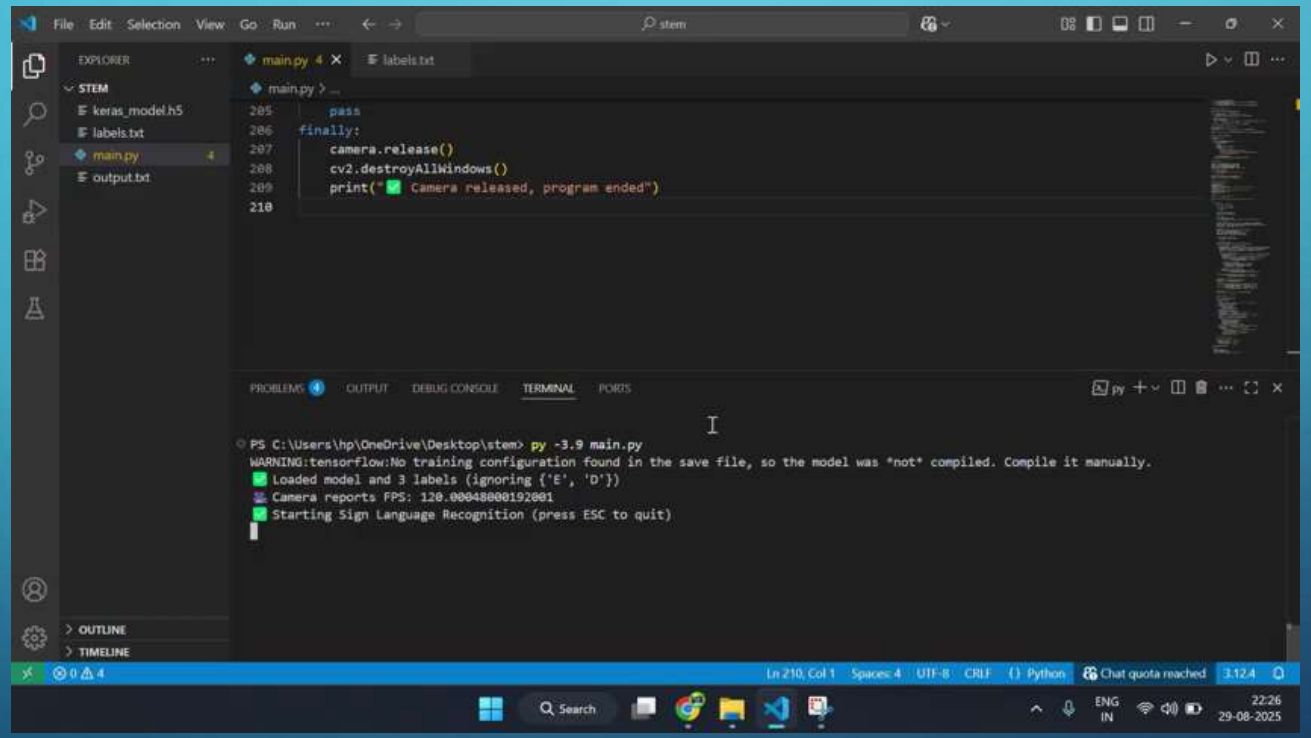
FUTURE SCOPE:

- Enhance model accuracy with expanded datasets.
- Incorporate multi language sign recognition.
- Improve UI/UX
- Integrate with wearable communication devices.

TalkPalm

REAL TIME DEMONSTRATION:

- Demonstration video of our prototype is attached below:



THANK YOU!