

# SignSpeak - Project Setup Guide

## Step 1: Install Required Dependencies

Open a terminal and install the necessary libraries:

```
pip install opencv-python mediapipe tensorflow numpy pandas matplotlib pyttsx3 googletrans tkinter
```

opencv-python - For webcam access and image processing.

mediapipe - For real-time hand tracking.

tensorflow - To train and run deep learning models.

numpy, pandas, matplotlib - For data processing and visualization.

pyttsx3 - Offline text-to-speech (TTS).

googletrans - For translating text if needed.

tkinter - To build the GUI.

## Step 2: Set Up the Project Structure

Create a project folder and set up the required files:

```
mkdir SignSpeak
```

```
cd SignSpeak
```

```
mkdir dataset models gui
```

```
touch main.py hand_tracking.py gesture_recognition.py tts.py gui.py
```

main.py - Runs the complete program.

hand\_tracking.py - Handles hand detection using OpenCV & Mediapipe.

gesture\_recognition.py - Trains and loads the deep learning model.

tts.py - Converts recognized gestures into speech.

gui.py - Manages the user interface.

## Step 3: Dataset Collection (Sign Language Images)

1. Download a Sign Language Dataset (like ASL or create a custom dataset).
2. Store images in the 'dataset/' folder (each folder represents a gesture).
3. Preprocess the images (resize, normalize, augment) before training.

## Step 4: Train the Gesture Recognition Model

1. Use a CNN-based model (TensorFlow/Keras) to classify gestures.
2. Save the trained model in the 'models/' folder.
3. Test the model to ensure accuracy is at least 90%.

## Step 5: Implement Text-to-Speech (TTS)

Use pyttsx3 for offline TTS or Google TTS for online speech conversion.

Convert recognized gestures into words and speak them aloud.

## Step 6: Build the GUI

Use Tkinter or PyQt to create an interactive interface.

Show real-time webcam feed, recognized gestures, and audio output.

## Step 7: Optimize & Test the Project

Reduce latency for real-time performance.

Handle multi-threading for webcam, recognition, and TTS simultaneously.

Test on different hand sizes, lighting conditions, and skin tones for robustness.

Now, Let's Start!

Do you want to start with hand tracking first or dataset collection?