



# SignSpeak

“👉👉👉👉👉👉”

Created by- [Krekheds](#)

## Overview:

SignSpeak is a machine learning project that aims to analyze videos and images containing sign language gestures and interpret them into English text and speech. It is designed to help mute people communicate more easily with the world. The project uses computer vision and deep learning techniques to recognize and understand sign language.

## Key Features:

### Live translation:

Signspeak converts sign language gestures live through your webcam without delay.

### Text and speech both:

Signspeak not only converts sign language into speech but it also converts it into text.

### User friendly:

While creating this application we kept in mind to make it as easy to use as possible, so in order to use this application you just need to run it and press the button “start” and that’s all you have to do.

### Accuracy:

Currently even with small dataset it is performing over 97% accuracy (tested with tensorflow built in accuracy tester).

### Light weight:

Even being deep learning project, this project is light weight and can run easily on most of the mid computers nowadays with good fps. Being Tensorflow light model is also one of the reasons of being light weight, and optimized code is also one of the reasons.

## Technologies used:

This project employs a combination of powerful libraries to achieve its functionality:

### 1. MediaPipe:

- MediaPipe is an open-source framework developed by Google for building multimodal ML pipelines.expand\_more
- In this project, MediaPipe plays a crucial role by providing pre-trained hand landmark detection and tracking models.expand\_more
- It efficiently analyzes video frames and extracts the location and orientation of key points on the hands, allowing us to track the user's sign gestures.expand\_more

### 2. TensorFlow:

- TensorFlow is a popular open-source library for numerical computation and machine learning.expand\_more
- This project leverages TensorFlow's capabilities to build and train a deep learning model for sign language recognition.
- The model learns to identify various ASL signs from the hand pose data provided by MediaPipe.

### 3. Pygame:

- Pygame is a Python library specifically designed for creating interactive games and multimedia applications.expand\_more
- In this project, Pygame is used to build a user-friendly graphical user interface (GUI).
- The GUI provides a user interface for navigating the system, displaying translated text, and offering other user-oriented features.

### 4. OpenCV (Open-Source Computer Vision Library):

- OpenCV is a widely used library for real-time computer vision and image processing.expand\_more
- Capturing video frames from the webcam in real-time.expand\_more
- Preprocessing video frames before feeding them into MediaPipe and the TensorFlow model.
- Displaying the video feed with overlaid information (optional).

### 5. GTTS (Google Text-to-Speech) (Optional):

- While not necessarily included in this project by default, gTTS is an optional library that can be integrated to provide Text-to-Speech functionality.
- gTTS allows the system to convert the translated text into synthesized speech, offering an additional communication channel for the user.

## Model:

SignSpeak utilizes a powerful deep learning architecture based on Convolutional Neural Networks (CNNs) to effectively recognize sign language gestures.

### Model Architecture:

- **Sequential CNN:** The model incorporates six stacked convolutional layers, leveraging the spatial and temporal information efficiently.
- **Activation Functions:** Five of these layers employ the ReLU (Rectified Linear Unit) activation function, introducing non-linearity and enhancing the model's learning capabilities.
- **Output Layer:** The final layer utilizes the Softmax activation function, responsible for generating class probabilities for each potential sign language gesture.

### Training Details:

- **Epochs:** The model was trained for 1000 epochs, ensuring sufficient exposure to the training data and optimizing its performance.
- **Training Split:** A 75% training split was used, dedicating a significant portion of the dataset to training while reserving the remaining 25% for validation and testing.
- **Accuracy:** The model achieved an impressive 97% accuracy on the validation set, demonstrating its effectiveness in sign language recognition.
- **Optimizer:** The Adam optimizer was employed, known for its efficiency and adaptability in optimizing model parameters during training.
- **Batch Size:** A batch size of 128 data points was utilized, balancing computational efficiency and training stability.
- **Model Saving:** The trained model was saved in the TensorFlow Lite format, enabling efficient deployment on low tier devices.

### Data Details:

- **Features:** Each gesture is represented by 42 features, capturing essential characteristics for accurate recognition.
- **Target Value:** The model predicts one target value, corresponding to the specific sign language gesture identified.
- **Dataset Size:** The current dataset boasts 3000 data records, providing a substantial foundation for robust model training.
- **Number of Classes:** The model is currently trained to recognize 10 distinct sign language classes.

# Processing Pipeline:

SignSpeak takes raw video input from your webcam and transforms it into spoken and written English text in real-time through a series of processing steps:

## 1. Image Acquisition:

- The process starts by capturing a frame from your webcam, providing a raw image representing the current scene.

## 2. Preprocessing:

- This stage prepares the image for further analysis:
  - **Color Space Conversion:** The image's color palette might be converted to a format more suitable for hand detection, like grayscale or HSV (Hue, Saturation, Value).
  - **Mirroring:** Depending on the expected hand orientation, the image may be mirrored to ensure consistency with the training data.

## 3. Hand Landmark Detection:

- The preprocessed image is fed to MediaPipe, a powerful library for pose estimation.
- MediaPipe identifies and extracts the coordinates of key hand landmarks (e.g., fingertips, palm center) within the image frame.

## 4. Feature Extraction:

- The extracted hand landmark coordinates are transformed into relevant features suitable for the model's input.
- This might involve calculating distances between landmarks, angles between fingers, or other geometric features that capture the essential information about the hand gesture.

## 5. Gesture Classification:

- The preprocessed features are fed into the pre-trained Convolutional Neural Network (CNN) model.
- The model analyzes the features and outputs a class probability distribution, indicating the likelihood of each possible sign language gesture.

## 6. Label Mapping and Output Generation:

- Based on the most likely class predicted by the model:
  - The corresponding sign language gesture label is retrieved.
  - This label is used to:

- **Speech Synthesis:** Text-to-speech software converts the label into spoken English audio, played through your device's speakers.
- **Text Display:** The label is displayed on the screen, providing a written representation of the recognized gesture.

## Future Developments:

The SignSpeak team is actively expanding the model's capabilities by:

### Increasing the number of recognized classes:

- Continuously adding new sign language gestures to the model's repertoire.

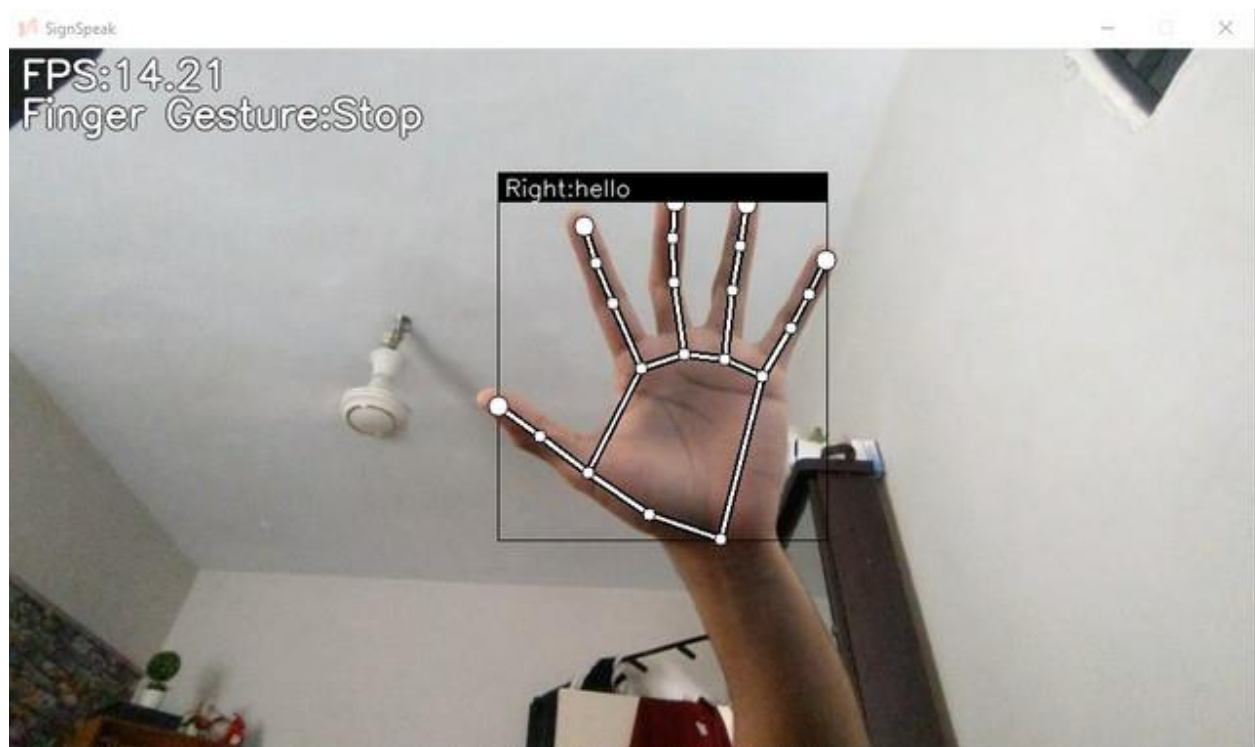
### Enhancing accuracy:

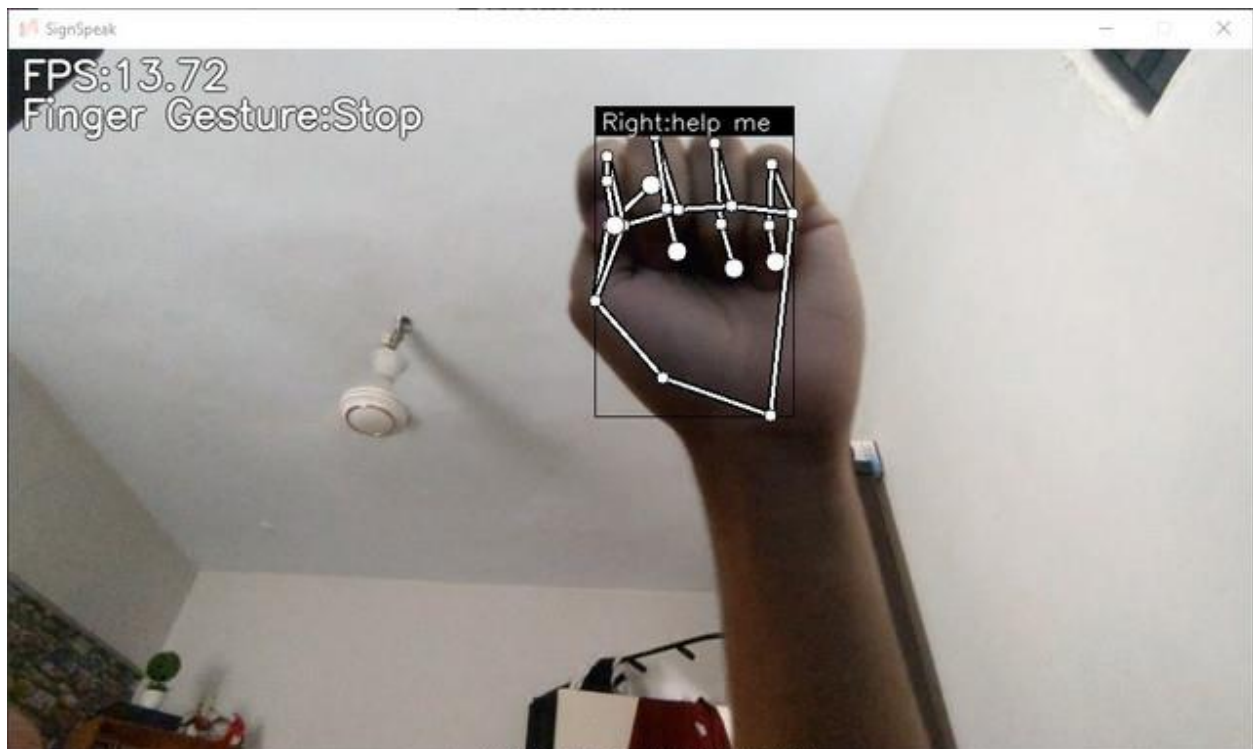
- Experimenting with advanced deep learning techniques to further improve recognition performance.

### Real-time optimization:

- Optimizing the model for real-time sign language interpretation on various devices.

## Snapshots:





## Open Source and Community Driven:

SignSpeak is proud to be an open-source project, available under the Apache 2.0 license. This means you have the freedom to:

### Use and modify the code:

- Feel free to integrate SignSpeak into your projects, modify it to suit your needs, and contribute your improvements back to the community.



### **Distribute freely:**

- Share SignSpeak with others, whether for personal or commercial use, without any restrictions.

### **Collaborate and contribute:**

- We encourage participation and contributions from everyone! Join the community to discuss ideas, share your expertise, and help us make SignSpeak even better.

By choosing an open-source license, we believe in fostering a collaborative environment where everyone can benefit from this technology and help advance its capabilities. We welcome developers, researchers, and anyone passionate about making communication more accessible to join us on this journey.

### **Additional Resources:**

- Apache License 2.0: <https://www.apache.org/licenses/LICENSE-2.0>
- SignSpeak GitHub Repository: <https://github.com/DanyalAbbas/SignSpeak>