SIGN LANGUAGE DETECTION

Presented to:

Mr. Asit Kumar Das

Presented by:

Group: StarFleet

Sai Ayushman Nayak
Pratyush Kumar Panigrahi
Bibhuti Bhushan Mohanty
Amarjyoti Senapati
Biswajit Nayak

"STOP DREAMING
AND START
DOING"

PROJECT INTRODUCTION

This project focuses on creating a machine learning-based system to detect and recognize sign language gestures. Sign language plays a crucial role in communication for individuals with hearing or speech impairments. By leveraging computer vision and machine learning, this system aims to bridge communication gaps and promote inclusivity.

The system recognizes gestures such as "Hello," "Thank You," "Please," and "A" using a dataset of over 100 samples per gesture. The project uses tools like TensorFlow and OpenCV to achieve high accuracy in gesture detection.

PROBLEM STATEMENT

Communication barriers between hearing-impaired individuals and non-sign language users create significant challenges in daily interactions. While sign language serves as a vital communication tool, its understanding is not universal, leading to social and accessibility gaps. Current solutions, such as human interpreters or specialized devices, are often costly or impractical for widespread use. This project addresses the need for an efficient, automated system that leverages machine learning to detect and interpret sign language gestures in real-time, providing an accessible and scalable solution to bridge the communication gap.

OBJECTIVES

Recognize basic sign language gestures using machine learning

Enable real-time gesture detection and interpretation

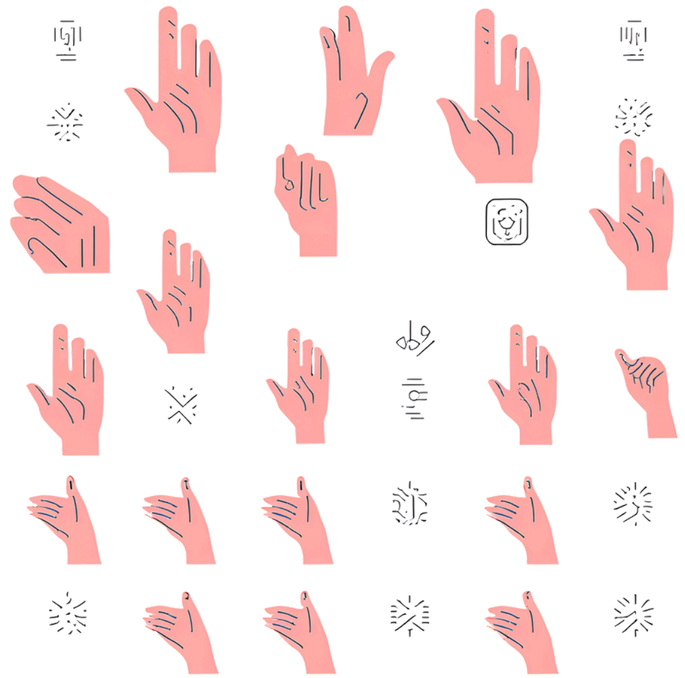
Enhance communication for hearing-impaired individuals

Bridge gaps between sign language users and non-users

Support scalability for additional gestures in the future

Deliver an efficient, accessible, and user-friendly solution

MOTIVATION



1. *To address the lack of widespread understanding of sign language by developing an automated recognition system*

2. *To leverage machine learning to create a practical, real-time solution for bridging communication barriers*

3. To contribute to the development of assistive technologies that empower individuals with hearing or speech impairments

LITERATURE SURVEY

Numerous research studies have explored sign language detection using various approaches:

Sign Language Recognition Using Convolutional Neural Networks (CNNs)

Depth Sensors and Wearable Services

Real-Time Gesture Recognition

TOOLS AND TECHNIQUES USED

PYTHON

TENSORFLOW

OPENCV

NUMPY

PANDAS

MATPLOTLIB

PROJECT DETAILS

Data Set:

- The dataset contains over 400 images, divided into folders corresponding to gestures such as “Hello,” “Thank You,” “Please,” and “A.”
- Each folder has more than 100 labeled images captured under varying conditions to improve model robustness.

Models:

- Trained using TensorFlow and stored in .h5 format.
- Labels for each model are stored in corresponding .txt files.

SCRIPTS:

- datacollection.py: Captures and labels images for training.
- test.py: Evaluates the model’s performance on unseen data.

IMPLEMENTATION DETAILS

The implementation comprises three primary stages..

Data Collection:

Images of hand gestures are collected using a webcam. The script captures images, labels them, and stores them in folders. Preprocessing steps include resizing, normalization, and augmentation.

Model Training:

A Convolutional Neural Network (CNN) is trained on the dataset. The model architecture includes:

INPUT LAYER:

Processes images of size 224x224 pixels.

CONVOLUTIONAL LAYERS:

Extract spatial features.

DENSE LAYERS:

Classify gestures based on extracted features.

Testing and Evaluation:

The trained model is tested on unseen data to evaluate its accuracy, precision, and recall.

CONCLUSION

The project successfully demonstrates a machine learning-based system for recognizing basic sign language gestures. The use of CNNs and a well-structured dataset ensures high accuracy and robustness.

FUTURE WORK

- Expanding the dataset to include additional gestures and variations.
- Optimizing the model for deployment on mobile or web platforms.
- Incorporating real-time feedback with audio or text outputs for enhanced usability.

REFERENCES

- **TensorFlow Documentation: https://www.tensorflow.org/**
- **OpenCV Documentation: https://opencv.org/**
- **Python Official Documentation: https://www.python.org/**
- **Teachable Machine:https://teachablemachine.withgoogle.com/**
- **Youtube Link:https://youtu.be/EiNyiqx1u2E**

THANK YOU