PONDICHERRY UNIVERSITY

(A CENTRAL UNIVERSITY)

Puducherry -605014

DEPARTMENT OF COMPUTER SCIENCE

(Master of Computer Science)

2nd REVIEW PROJECT REPORT

ON

Sign Language Detection

NAME: ASWANTH E K

REGISTRATION NO: 23370009

UNDER THE GUIDANCE OF: DR. R.SUBRAMANIAN

SIGNATURE OF THE GUIDE:

Sign Language Detection

Project Abstract:

Sign language is a crucial means of communication for individuals with hearing impairments. However, the language barrier between sign language users and non-signers poses a communication challenge. This project focuses on developing a sign language detection system using Python, OpenCV, and machine learning techniques. The system captures hand gestures through a webcam, processes the images, and classifies them using a trained model. The output is then displayed in real-time, aiding communication and accessibility.

Introduction

Sign language detection is an innovative application of computer vision that leverages deep learning to recognize hand gestures. Traditional methods of sign language translation require manual interpretation, which is not always available. By using machine learning and computer vision, we can automate this process and make communication more inclusive. This project implements a sign language detection system that captures and classifies gestures, translating them into readable text for real-time communication.

Project Overview

The project consists of four main components:

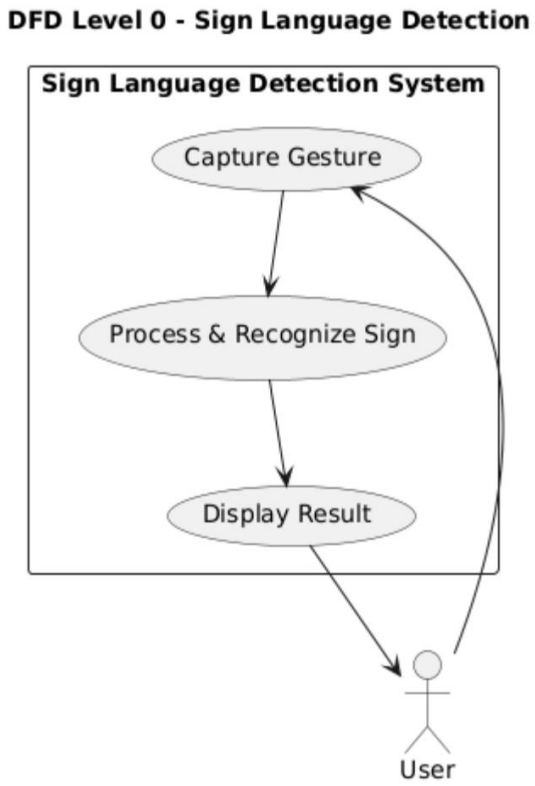
1. Data Collection – Capturing images of different hand gestures for training.

2. Preprocessing & Feature Extraction – Cleaning and enhancing images for better recognition.

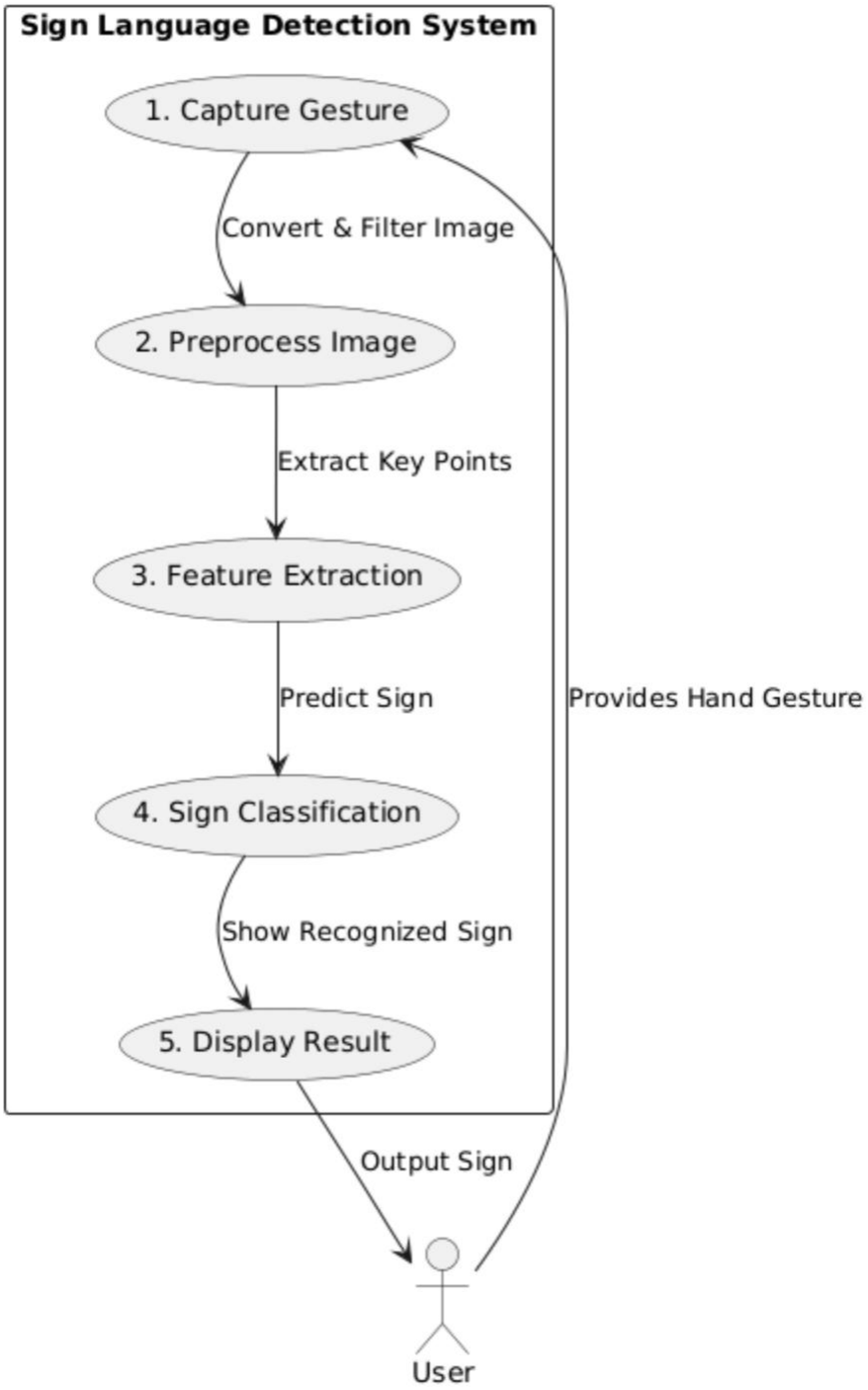
3. Model Training – Training a deep learning model to classify hand gestures.

4. Real-Time Detection & UI Implementation – Implementing a real-time system with a user-friendly interface.

DFD Diagrams



DFD Level 1 - Sign Language Detection



Components of the Project

1. Data Collection (Image Capturing)

Before training a model, we need a dataset of hand gestures representing different signs. The provided script captures images from a webcam and stores them in labeled directories.

Key Features of the Script:

- Captures multiple images per sign using OpenCV.
- Uses UUID-based unique filenames for image storage.
- Organizes images into folders like Hello, Yes, No, etc.
- Allows users to exit the program by pressing 'q'.

Model Training with Machine Learning

After preprocessing, the images are used to train a machine learning model. Popular models include:

1**. CNN (Convolutional Neural Networks) –** Best suited for image classification.
2. **YOLO V5 (You Only Look Once) –** If real-time sign detection is required.

4. Real-Time Sign Detection and Translation

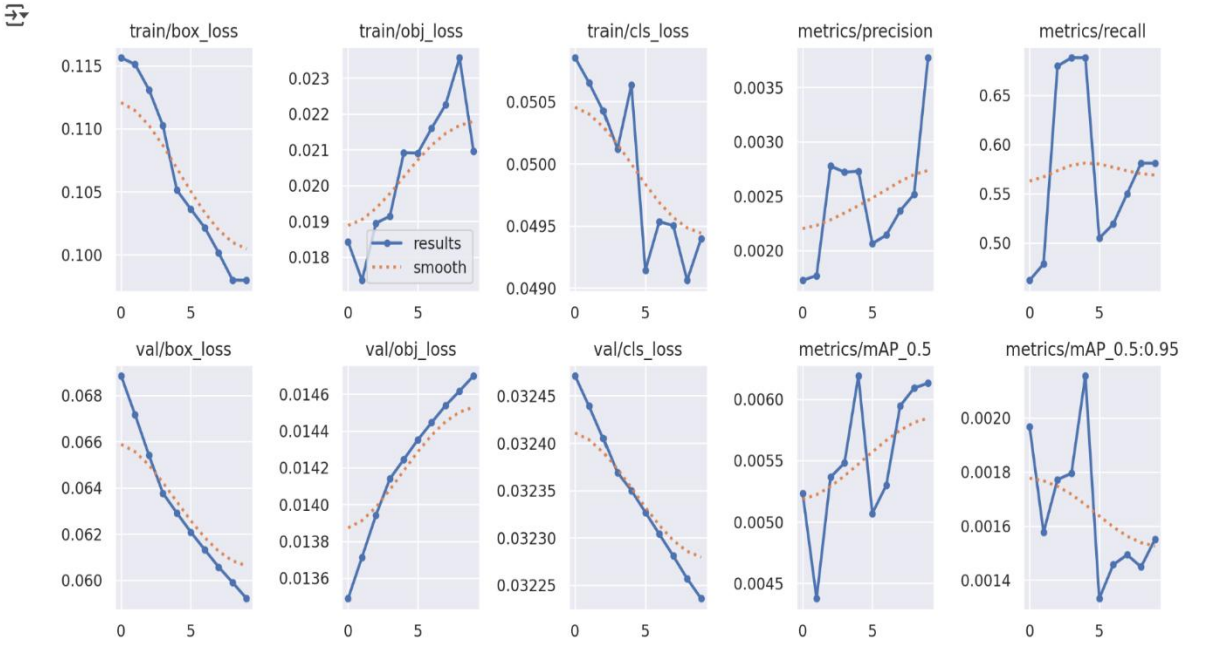
Once the model is trained, real-time detection is implemented using OpenCV.
Output: The camera detects a hand gesture and displays the corresponding sign name on the screen.

Challenges & Future Improvements

- **Lighting Conditions –** Can affect hand detection accuracy.
- **Sign Variations –** Different people sign slightly differently.
- **Real-time Processing –** Faster CNN models can improve performance.
- **Integration with NLP –** Convert signs into text/speech using NLP models

Evaluate Custom Yolov5 Detector Performance

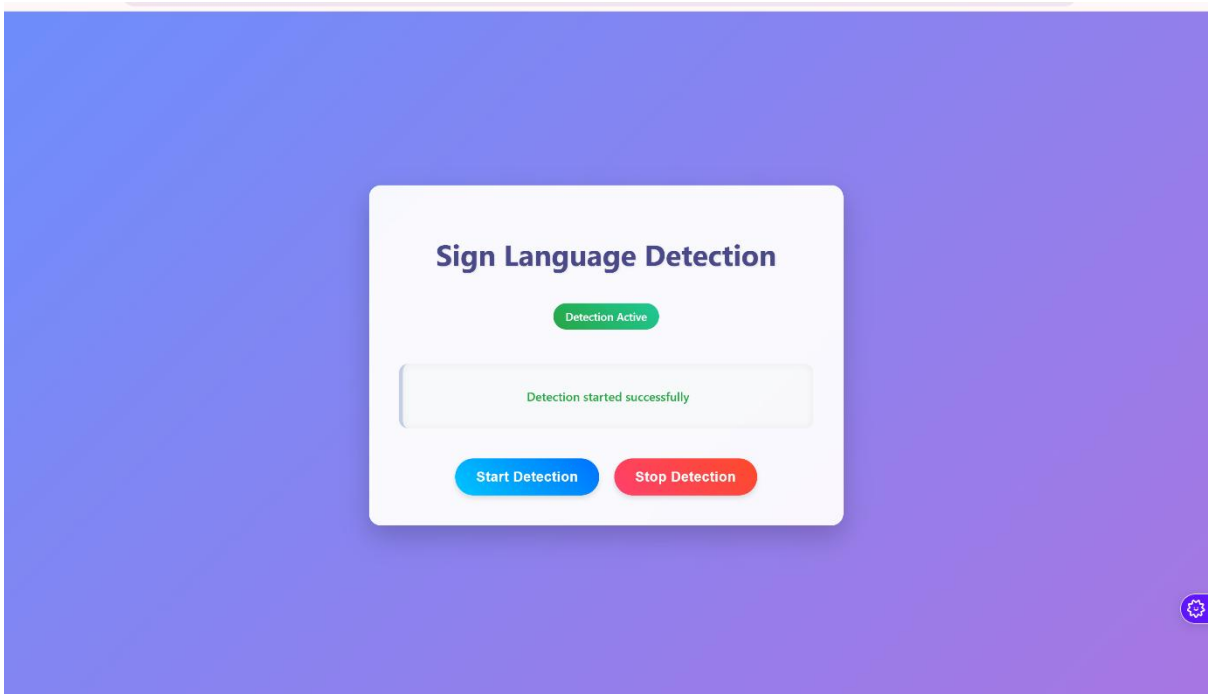
```
from utils.plots import plot_results
Image(filename='/content/yolov5/runs/train/yolov5s_results/results.png',width=1000)
```

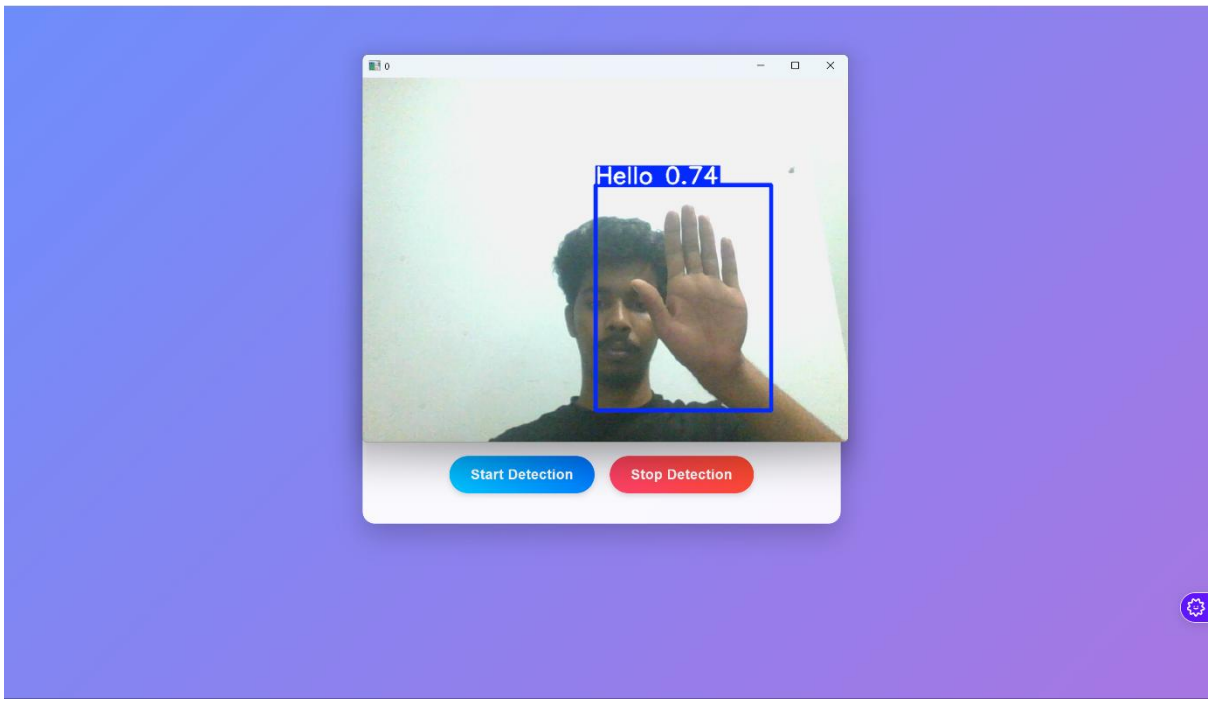


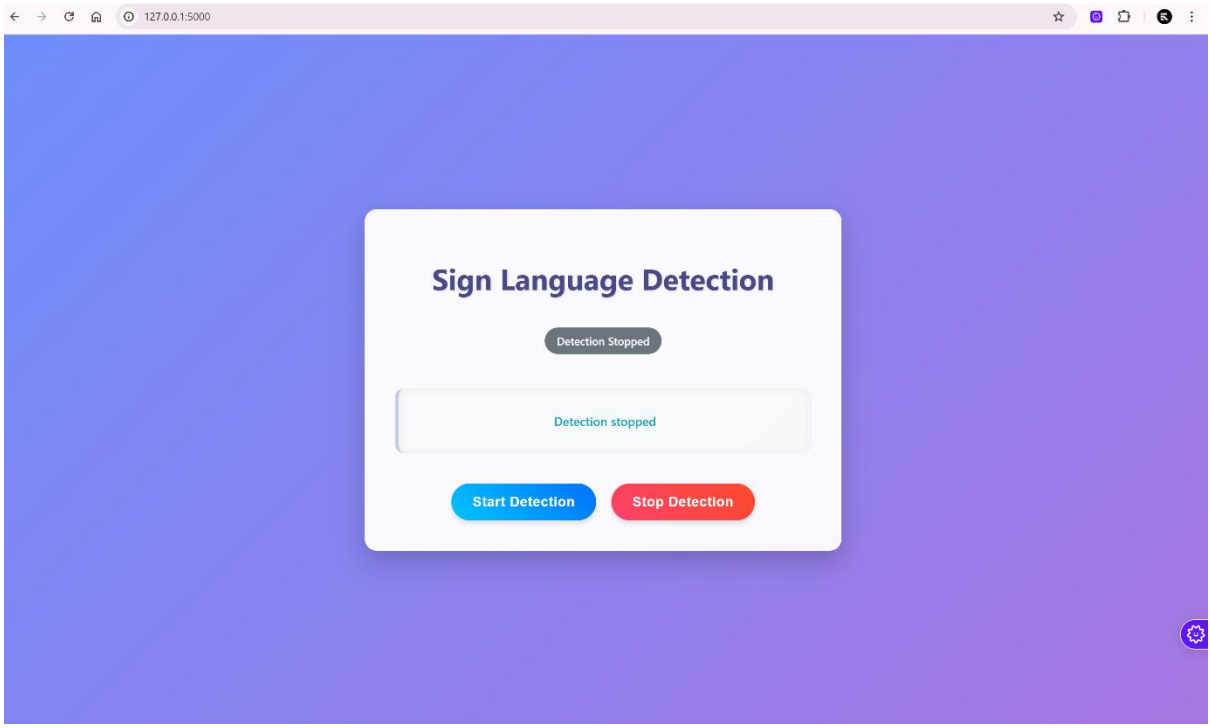
```
print("GROUND TRUTH TEST DATA:")
Image(filename='/content/yolov5/runs/train/yolov5s_results/train_batch0.jpg',width=900)
```

```
GROUND TRUTH TEST DATA:
```

Frontend UI


Sign Language Detection
Detection Active
Detection started successfully
Start Detection
Stop Detection


Hello 0.74
Start Detection
Stop Detection


127.0.0.1:5000
Sign Language Detection
Detection Stopped
Detection stopped
Start Detection
Stop Detection

Conclusion

In this project, we successfully developed a real-time Sign Language Detection system using Python, Machine Learning, YOLOv5, and OpenCV. The system is capable of detecting and interpreting hand gestures corresponding to sign language, offering an effective solution to bridge the communication gap between hearing-impaired individuals and others. By creating a custom dataset and annotating it using LabelImg, we ensured that the model was tailored to our specific gesture classes. The use of YOLOv5 allowed us to achieve fast and accurate object detection, making the system suitable for real-time applications. The integration of a user-friendly interface further enhances accessibility and usability. Overall, this project demonstrates the potential of AI-powered vision systems in creating more inclusive communication tools and highlights the importance of technology in promoting accessibility and social integration.