Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Python Programming

Mini Project Report

On

Sign Language Interpreter using Random Forest Classifier in Python

By

Sanket Kharche (Roll number 39)
Vinay Yadav (Roll number 43)
Nia UK (Roll number 48)

Under the guidance of

Dr Pradnya Kulkarni

Department of Computer Engineering and Technology
School of Computer Science and Engineering
Dr. Vishwanath Karad MIT World Peace University
AY 2023-24 Semester IV

Table of Contents

List of Figures

1. Introduction

1.1 Sign Language and Communication Barriers:

Sign Language is a complete and natural language used by deaf and hard-of-hearing individuals to communicate. It utilizes hand gestures, facial expressions, and body language to convey meaning. ASL has its own grammar and syntax, distinct from spoken languages.

Despite the growing awareness of ASL and its importance, communication barriers still exist between those who rely on sign language and those who don't understand it. This can create challenges in various situations, including difficulty in social interactions, education and employment and even Daily Interactions

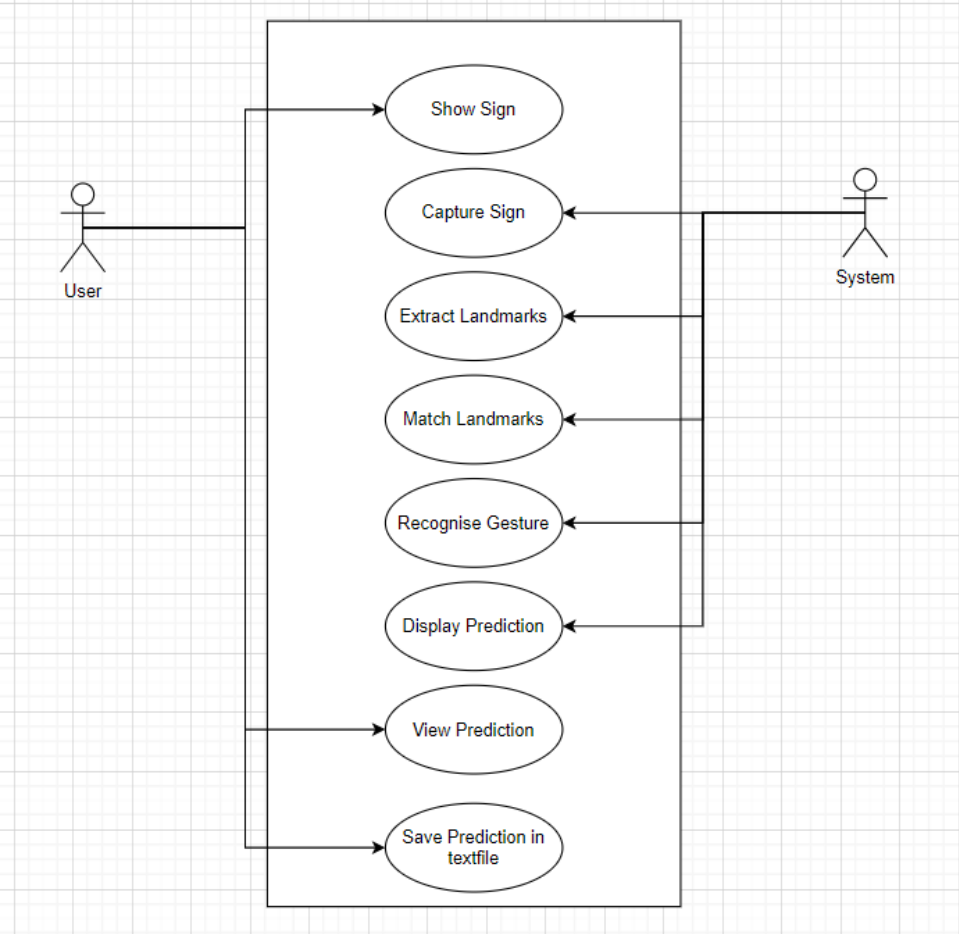
1.2 Sign Language Interpretation and Technology:

Sign language interpreters play a crucial role in bridging the communication gap between deaf and hearing communities. They translate spoken language into sign language and vice versa, facilitating communication in various settings.

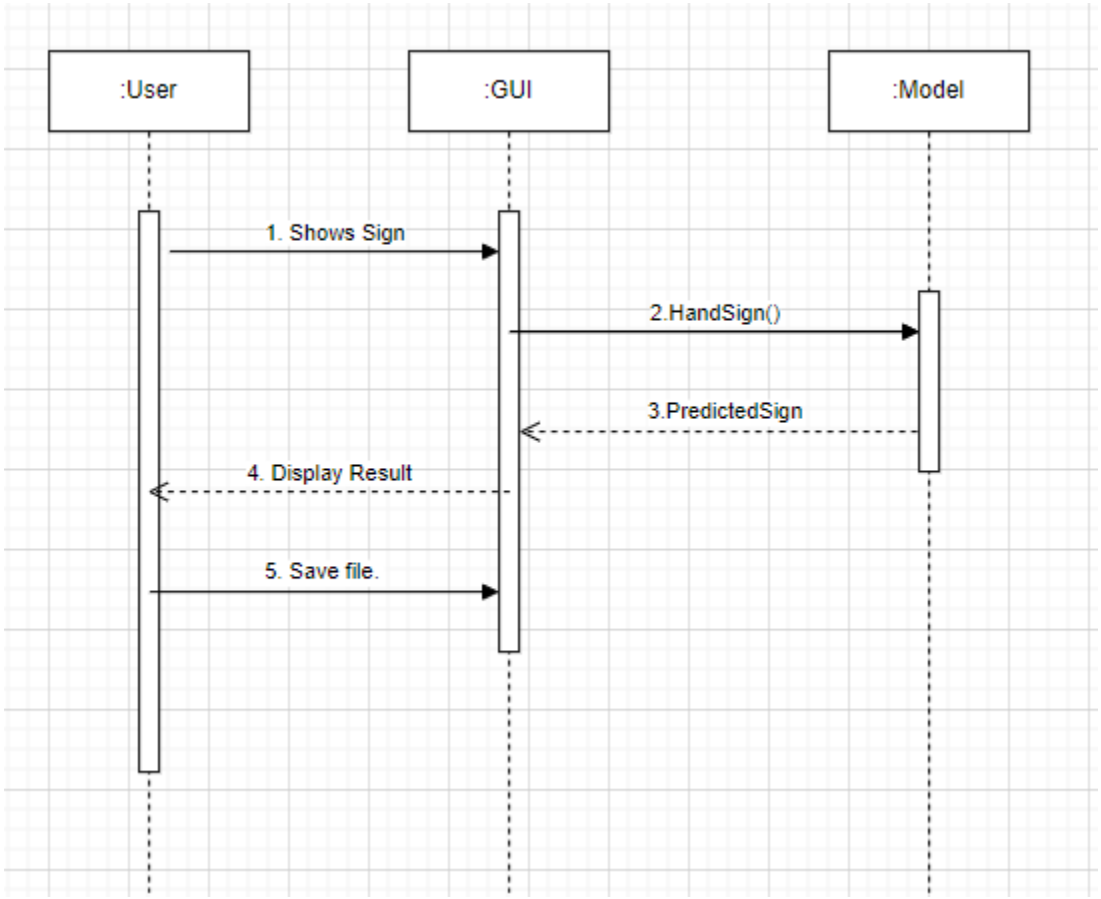
2. Problem Statement:

Limited access to sign language interpreters creates communication barriers for deaf and hard-of-hearing individuals. We propose a mini Sign Language interpreter application using Random Forest Classifier to provide a more accessible and real-time interpretation solution.

3. Use Case Diagram:



4. Sequence Diagram:



5. Tasks Performed:

5.1: Data Acquisition and Preprocessing:

5.1.1: Data Collection:

- Created a custom dataset of sign languages im

5.1.2: Data Preprocessing and Feature Extraction:

- Utilize Media Pipe's hand landmark detection to extract keypoint coordinates from the pre-processed sign language images

5.3: Machine Learning Model Development and Training:

5.3.1: Model Building:

- Utilized Pretrained Model of Random Forest for sign language classification

5.3.2: Model Training:

- Trained Model on collected dataset containing hand landmark features corresponding to labels

5.3.2: Model Evaluation:

- Evaluation metrics of accuracy are calculated during training to gauge performance

5.4: UI Development:

- Designed and Coded UI for application using Tkinter
- UI displays a live video feed of user's hand and overlays the predicted sign language character

6. Technology Used:

6.1: Hardware:

- Memory: 6gb RAM as recommended for smooth operation
- Storage: Storage space for storing trained model and pre-processed data
- Webcam

6.2: Software:

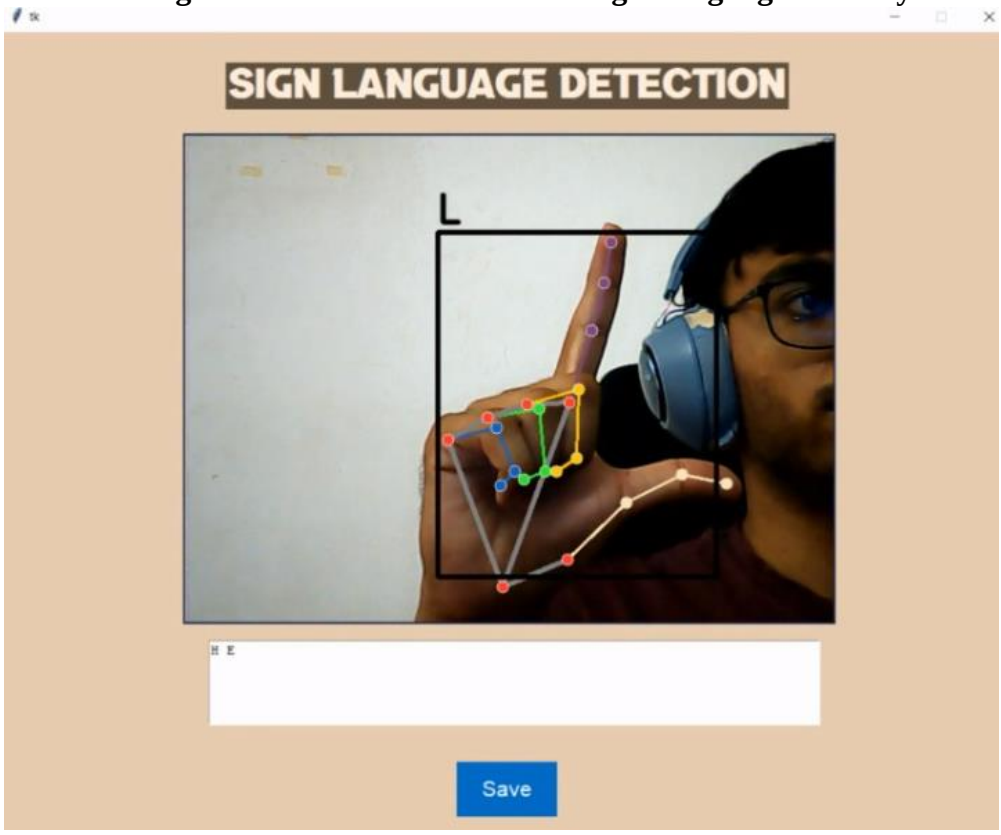
- Operation System: Windows/macOS
- Programming Language: Python 3.12.0

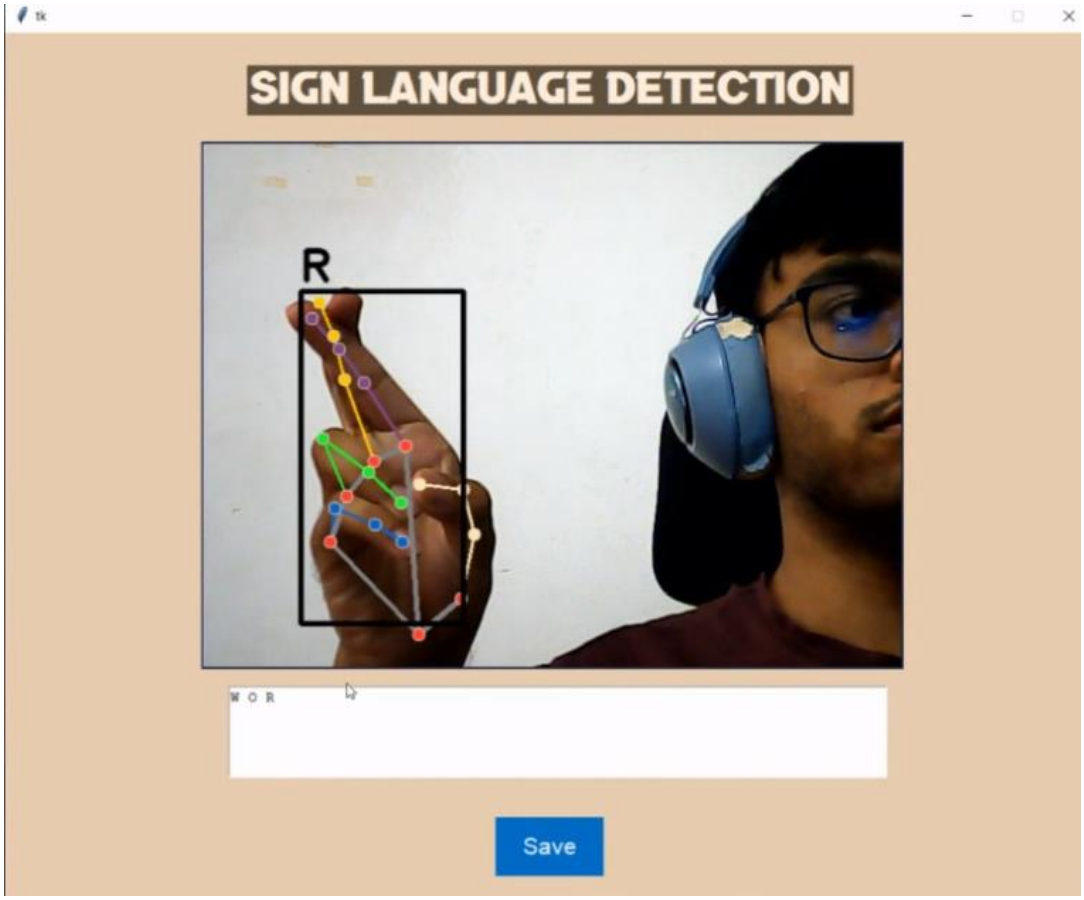
6.3: Libraries:

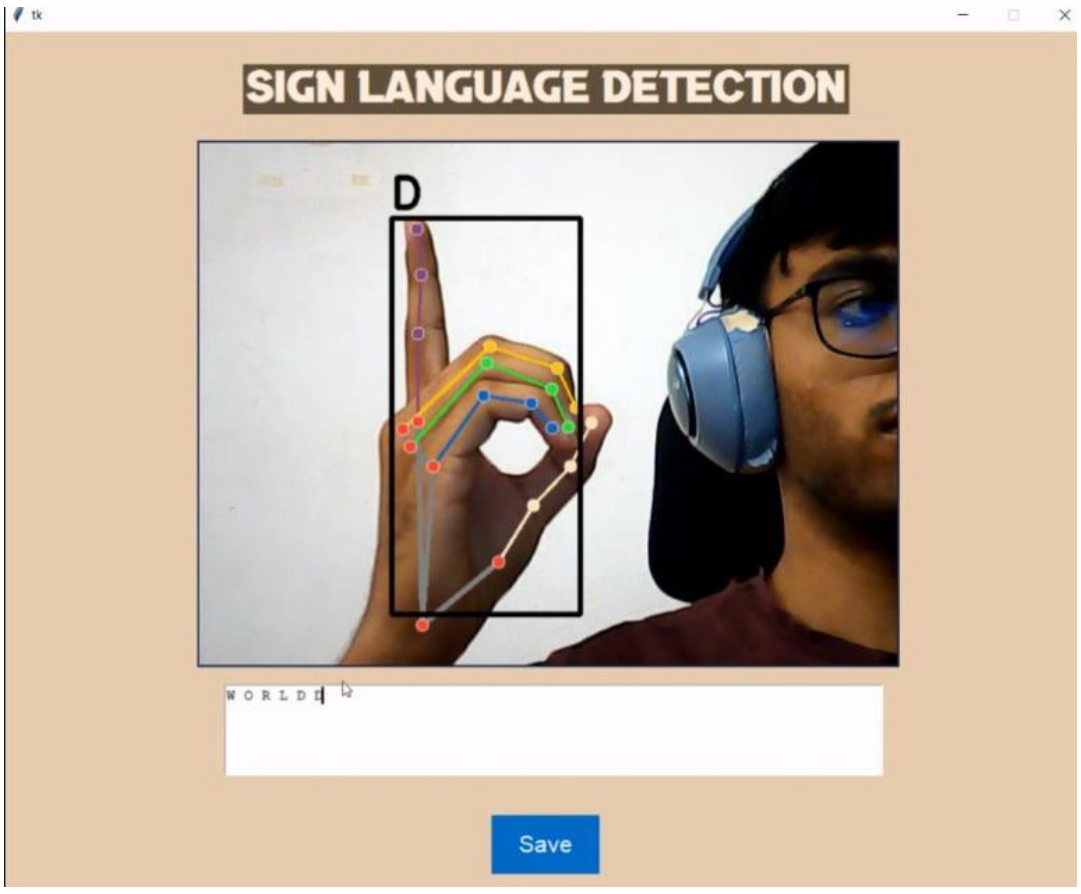
- OpenCV
- Mediapipe
- Scikit Learn
- Pickle
- Matplotlib
- NumPy
- Tkinter

7. Output and Visualization Screenshots:

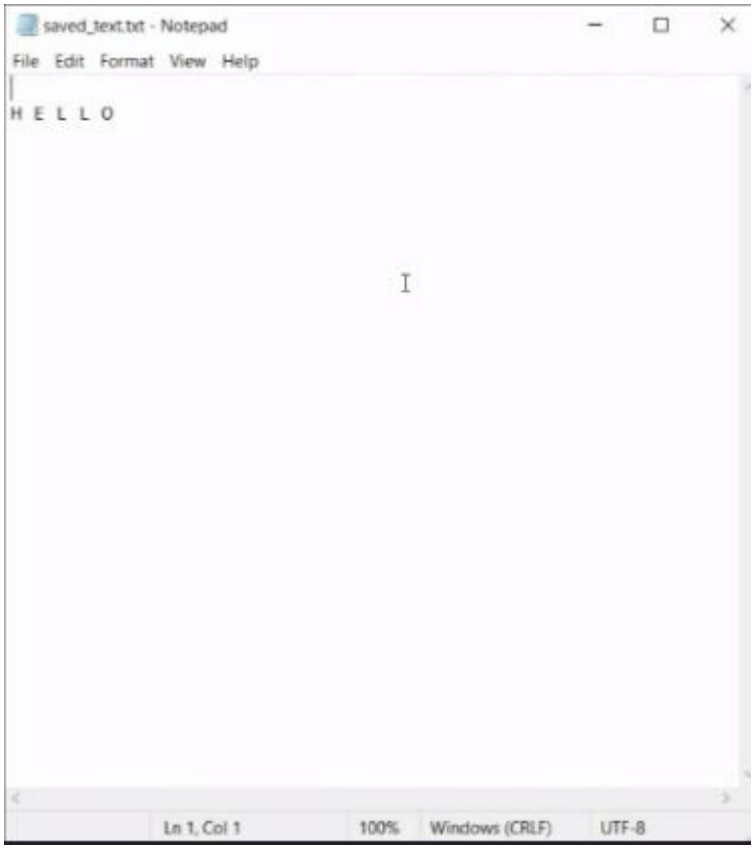
7.1 Gesturing "H E L L O W O R L D" in Sign Language letter by letter:




tk
SIGN LANGUAGE DETECTION
R
W O R
Save


tk
SIGN LANGUAGE DETECTION
D
W O R L D
Save

7.2 Exporting “H E L L O W O R L D” from Sign Language Interpreter to text file:



8. Conclusion:

This project successfully developed a desktop ASL interpreter application utilizing a trained Random Forest Classifier model. Users can capture signs through webcam. The application preprocesses the data, extracts relevant features, and leverages the Random Forest Classifier model for sign recognition. The interpreted text is displayed on the user interface. User can also save their interpreted signs on text files.

This project demonstrates the potential of Random Forest Classifier for real-world applications like ASL interpretation. By focusing on core functionalities, the application provides an accessible tool for communication and language learning. Future advancements could involve expanding the vocabulary support, incorporating real-time sign language translation, or integrating error handling for complex scenarios. Overall, this project serves as a foundation for further development in the field of ASL interpretation technology.

9. References:

9.1: Research Papers:

[1] Fang Biyi, Co Jillian, and Zhang Mi. 2017. DeepASL: Enabling ubiquitous and non-intrusive word and sentence-level sign language translation.

[2] Ira Cohen, Nicu Sebe, Ashutosh Garg, Lawrence S Chen and Thomas S. Huang, "Facial expression recognition from video sequences: temporal and static modeling", Computer Vision and Image Undertaking, pp. 91, February 2003.

[3] Chen, J.K. Sign Language Recognition with Unsupervised Feature Learning; CS229 Project Final Report; Stanford University: Stanford, CA, USA, 2011.

[4] Muthu Mariappan, H.; Gomathi, V. Real-Time Recognition of Indian Sign Language. In Proceedings of the International Conference on Computational Intelligence in Data Science, Haryana, India, 6–7 September 2019.

[5] Anderson, R., Wiryana, F., Chandra, M., Putra, G.: Sign language recognition application systems for deaf-mute people: a review based on input-process-output. Procedia Comput. Sci. 116, 441–448 (2017)

9.2: Websites:

[1]https://medium.com/@harshdeepsingh_35448/understanding-random-forests-aa0ccecdbbbb

[2]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[3] https://www.javatpoint.com/machine-learning-random-forest-algorithm