IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.53

Volume 4, Issue 2, April 2024

Sign Language Recognition using Python

Prof. Samgir K. R[1], Sakshi D. Phadatare[2], Suraj H. Pisal[3], Soham D. Pise[4]

Professor, Department of Computer Science and Engineering[1]

Students, Department of Computer Science and Engineering[2,3,4]

Navsahyadri Education Society's Group of Institutions, Polytechnic, Pune, Maharashtra, India

Abstract: *Sign language recognition (SLR) using Python offers a promising avenue for enhancing communication accessibility for the deaf and hard of hearing community. In this project, we explore the implementation and scope of a computer system for SLR, leveraging Python's rich ecosystem of libraries and frameworks for machine learning and computer vision. The project encompasses data collection, preprocessing, model selection, training, evaluation, and deployment phases. Data collection involves gathering a diverse dataset of sign language gestures, while preprocessing encompasses tasks such as resizing, normalization, and augmentation. Model selection involves choosing a suitable architecture, with convolutional neural networks (CNNs) being a prevalent choice for image-based tasks like SLR. Following model selection, the system undergoes training, during which the model learns the patterns and features of different sign language gestures. Evaluation metrics such as accuracy, precision, recall, and F1-score are utilized to assess the model's performance on a separate test dataset. Finally, the trained model is deployed into a Python application, providing a user-friendly interface for real-time gesture recognition. Through this project, we aim to contribute to the advancement of assistive technologies for the deaf and hard of hearing community, fostering inclusive communication environments through the utilization of Python-based SLR systems.*

Keywords: Sign language recognition

I. INTRODUCTION

To introduce sign language recognition using Python, you can start by explaining the concept of sign language and its importance for the deaf and hard of hearing community. Then, you can outline the steps involved in sign language recognition, which typically include:

1. Data Collection
2. Preprocessing
3. Model Selection
4. Training
5. Evaluation
6. Deployment
7. User Interface

You can implement sign language recognition using Python libraries such as TensorFlow, Keras, OpenCV, and scikit-learn for machine learning tasks. Additionally, there are pre-trained models and datasets available to facilitate the development process. You can provide code snippets and examples to demonstrate each step of the process in your introduction.

II. LITERATURE SURVEY

A literature survey of sign language recognition using Python involves looking at different studies and research articles about how computers can understand and interpret sign language using the Python programming language. This survey explores various techniques, such as deep learning with neural networks, to teach computers to recognize different hand movements and gestures that represent words or letters in sign language. It also considers the availability of datasets containing sign language examples for training these systems and the challenges faced, like varying lighting conditions and background distractions. Researchers use metrics like accuracy and precision to measure how well these systems work, and they look at practical applications, such as helping people who are deaf communicate better or assisting in educational tools for learning sign language.

III. OBJECTIVE

- Develop a real-time sign language recognition system using Python.
- Achieve high accuracy in recognizing signs from video input.
- Design an intuitive user interface for easy interaction.
- Ensure robustness to variations in lighting conditions and hand orientations.
- Enable integration with other applications or platforms.
- Strive for scalability to accommodate a wide range of signs and users.

IV. PROPOSED METHODOLOGY

Sign language recognition using Python involves a multi-stage approach. First, data collection is conducted by capturing video recordings of sign language gestures, either using a dedicated camera setup or existing video datasets. Next, preprocessing techniques such as frame extraction, normalization, and noise reduction are applied to the video data to enhance its quality and suitability for analysis. Then, feature extraction methods, such as deep learning-based feature learning or handcrafted feature engineering, are employed to extract relevant information from the preprocesses video frames. Subsequently, a classification model, such as a convolutional neural network (CNN) or recurrent neural network (RNN), is trained on the extracted features to recognize and classify different sign language gestures. Finally, the trained model is evaluated using appropriate metrics and fine-tuned as necessary to improve its performance. This methodology leverages Python libraries such as OpenCV, TensorFlow, and Keras for implementation, ensuring scalability, flexibility, and compatibility with existing sign language recognition frameworks.

V. ADVANTAGES AND APPLICATIONS

5.1 ADVANTAGES

- Ease of learning.
- Vast community support.
- Rich Ecosystem of libraries.
- Cross-platform compatibility.

5.2 APPLICATION

- Sign language recognition using python is diverse and impactful.

VI. CONCLUSION AND FUTURE SCOPE

In conclusion, using Python for sign language recognition offers numerous advantages, including ease of learning, vast community support, a rich ecosystem of libraries, cross-platform compatibility, and rapid prototyping capabilities. Moving forward, the future scope of sign language recognition using Python looks promising, with opportunities for further advancements in accuracy, real-time processing, and user interface design. Continued research and development in this field could lead to more accessible and effective communication tools for the deaf and hard of hearing community, as well as broader applications in education, accessibility, and human-computer interaction interfaces.

REFERENCES

[1]. https://medium.com/@20it105/sign-language-recognition-using-python-74ef7ea43181
[2]. https://ieeexplore.ieee.org/document/9996001/
[3]. https://chat.openai.com/share/c5951ee6-586b-4697-bcf1-6dbef6b11703
[4]. https://chat.openai.com/share/ce7ec8f0-779e-483c-b6a9-107ae7a69f56