A Project Report

(Review III report)

On

Sign Language and Emotion Detection

Submitted by **Team 579**

Rahul Gandluru – 20BCE2120

Rohan Kant – 20BKT0102

Gaurav Navada – 20BKT0128

Karthik Gopagoni – 20BKT0130

For

SmartBridge – Artificial Intelligence

June 2023

Table of Contents

Pg No.

Chapter 1 - Introduction

1.1 Overview

With real-time camera access, the "Sign Language and Emotion Detection" project intends to empower Deaf people by converting their signed language into written text. To analyse and comprehend the feelings exhibited by the users, the project also uses emotion detection utilising photographs. Convolutional Neural Networks (CNN) are used in the project as the main detection model, drawing on ideas and methods covered in the SmartBridge AI course. The Flask framework is used to deploy the project as a web page, making it accessible and user-friendly.

The communication barrier between Deaf individuals and the hearing community is a significant challenge. Although it is not widely understood by the public, sign language is the main form of communication for many Deaf people. This project aims to bridge this communication gap by providing real-time translation of sign language into written text. By using a camera feed and advanced AI techniques, the project captures the sign language gestures and converts them into textual form, enabling effective communication between Deaf individuals and those who do not understand sign language.

Additionally, the project goes beyond translation by incorporating emotion detection. Emotions play a crucial role in communication, conveying subtle nuances and intent. By analyzing the emotions expressed by individuals, the project aims to enhance communication experiences, fostering empathy and facilitating more meaningful interactions.

1.2 Purpose

This project aims to remove the barriers to communication that Deaf people encounter when interacting with the hearing community. Many Deaf people use sign language as their primary form of communication, although the general public does not always comprehend it. This project intends to close the communication gap between the hearing community and the deaf population by offering real-time translation of sign language into written text, enabling more effective and inclusive communication.

Further enhancing the communication experience for Deaf people is the integration of emotion recognition. Any type of communication relies heavily on emotions to convey meaning and intention.

Chapter 2 – Literature Survey

1. **Problem**: Limited sign language datasets –

   Sign language datasets are often small, making it challenging to train accurate models.

   Solution: We have implemented our own image dataset, where we input many images of each alphabet of the sign language.

   Reference:

   Camgoz, N.C., Hadfield, S., Koller, O., Bowden, R. (2018). Neural Sign Language Translation. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 234-250.

2. **Problem**: Real-time sign language recognition -

   Real-time recognition of sign language is crucial for effective communication, but it poses challenges due to the temporal nature of gestures.

   Solution: Using CNN for building the detection model with 90% accuracy and integrating it with Flask to access the real-time camera feed on a web page.

   Reference:

   Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2625-2634.

3. **Problem**: Less Accuracy Detection Model –

   The accuracy of image detection models are usually low due to reasons like overfitting, etc.

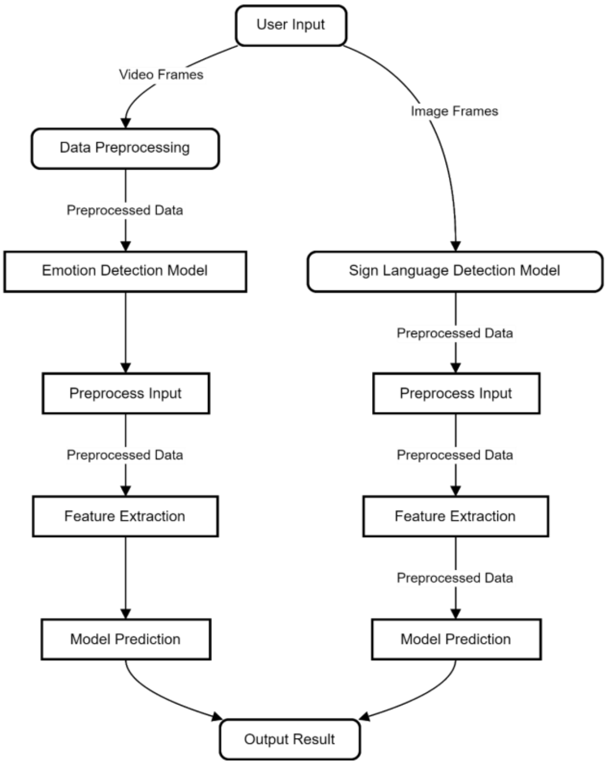
   Solution: Implemented Mediapipe library to detect hand landmarks to get a blueprint of the hand gesture, resulting in diversity amongst the alphabets of the sign language and thus easier to differentiate, and therefore making it more accurate.

   References:

   Pigou, L., Dieleman, S., & Kindermans, P. J. (2016). Sign language recognition using convolutional neural networks. In Proceedings of the 30th AAAI Conference on Artificial Intelligence.

Chapter 3 – Theoretical Analysis

3.1 Block Diagram



Emotion Detection Model:

- The user input, which are image frames, undergoes data preprocessing to convert the frames into a suitable format for the model.

- Preprocessed input data is fed into the Emotion Detection Model.

- The Feature Extraction block extracts relevant features from the preprocessed data.

- The extracted features are passed to the Model Prediction block, where a pre-trained CNN model predicts the emotion expressed in the input.

- The Output Result block obtains the predicted emotion from the model's prediction.

Sign Language Detection Model:

- The user input, which consists of image frames, is converted into a suitable format through data preprocessing.

- The Preprocess Input block normalizes the dimensions or scales the image frames to meet the requirements of the model.

- The Feature Extraction block extracts relevant features from the preprocessed image frames.

- The extracted features are passed to the Model Prediction block, where a pre-trained model predicts the corresponding sign language gesture or symbol.

- The Output Result block obtains the predicted sign language gesture from the model's prediction.

3.2 Hardware / Software Designing

Hardware Requirements:

1. Camera: The camera is a crucial hardware component for this project. It provides real-time access to the users by capturing the video stream. The camera should be compatible with your system and capable of capturing clear and high-quality video frames. Depending on the application, you may consider using a webcam, a built-in camera on a laptop, or an external camera.

Software Requirements:

1. Python: The project is implemented using the Python programming language. Therefore, you need to have Python installed on your system. Python provides a rich set of libraries and frameworks that are utilized throughout the code implementation. Make sure you have a compatible version of Python installed, preferably Python 3.x, which is commonly used for modern Python development.2.

2. OpenCV: OpenCV (Open Source Computer Vision Library) is a widely-used computer vision library. It provides a comprehensive set of functions and algorithms for image and video processing tasks. In this project, OpenCV is used for various operations such as capturing video frames from the camera, converting color spaces, cropping images, and drawing landmarks on the video frames. Install OpenCV using the command `pip install opencv-python`.

3. Mediapipe: Mediapipe is an open-source framework developed by Google. It simplifies the development of complex media processing pipelines, including hand detection, pose estimation, face detection, and more. In this project, Mediapipe is used specifically for hand detection. You can install Mediapipe using the command `pip install mediapipe`.

4. Keras and TensorFlow: Keras is a high-level neural networks API written in Python, and TensorFlow is a popular deep learning framework. They are used in this project for loading and running the pre-trained model for emotion recognition. Keras provides a user-friendly interface for defining and training deep learning models, while TensorFlow provides the backend for executing the computations efficiently. Install Keras and TensorFlow using the commands `pip install keras` and `pip install tensorflow`.

5. Flask: Flask is a lightweight and popular web framework for Python. It is used in this project to deploy the application as a web page. Flask simplifies the process of creating web applications and handling HTTP requests. You can install Flask using the Python package installer (pip) by running the command `pip install flask` in your terminal or command prompt.

6. HTML/CSS: The project involves deploying a web page to provide access to the application. While the provided code focuses on the backend implementation using Flask, you may need HTML and CSS knowledge to customize and enhance the web interface. The 'index.html' file referenced in the code should be present in

the appropriate template directory of the Flask application. HTML is used to structure the content of the web page, while CSS is used for styling and layout purposes.

Chapter 4 – Experimental Investigation

This experimental investigation focuses on the development and implementation of a real-time sign language translation system for deaf individuals. The system enables users to translate their sign language gestures into text or speech using camera access. Additionally, the investigation explores emotion detection from images to enhance the understanding of users' emotional expressions. The implementation leverages concepts learned in the SmartBridge AI class, particularly Convolutional Neural Networks (CNNs), for both sign language and emotion detection. The system is further integrated with Flask to deploy it as a web page, ensuring easy accessibility.

1. Sign Language Translation:

- The first aspect of the investigation involves sign language translation using a CNN model. The implementation incorporates the MediaPipe library for hand detection and tracking. By utilizing the pre-trained MediaPipe hand detection model, the system detects and locates the hand region within the camera frame. The bounding box coordinates of the hand are extracted for subsequent processing.

- To accurately capture hand gestures, the hand region is cropped based on the bounding box coordinates. The cropped image is then resized to a fixed dimension, typically 224x224 pixels, to maintain consistency for the CNN model. Preprocessing steps include converting the image to an array and expanding the dimensions to match the input requirements of the CNN model.

- The trained CNN model is loaded to predict the sign language gesture from the cropped hand image. The model assigns a class label corresponding to the predicted sign gesture, which is then mapped to the appropriate text representation. To provide real-time feedback to the user, the text representation is overlaid on the video frame using OpenCV functions. Additionally, landmarks on the hand region are visualized using the MediaPipe library.

2. Emotion Detection from Images:

- The second aspect of the investigation focuses on emotion detection from images using another CNN model. While the sign language translation occurs in real time, the system switches to image-based emotion detection. Instead of processing a continuous video stream, a single image is captured from the camera feed to analyze the user's emotional state.

- Similar to sign language translation, the MediaPipe library is used to detect and track the hand region within the captured image. The bounding box coordinates of the hand region are extracted to crop the image and isolate the hand area for emotion detection. Measures are taken to ensure that the cropped region remains within the boundaries of the image to avoid processing errors.

- The cropped image is resized to a fixed size, often 224x224 pixels, to maintain consistency with the input requirements of the CNN model. The preprocessed image is then passed through the loaded CNN model, which predicts the emotional state based on the captured hand gesture. The output of the model represents the predicted emotion class, such as happy, sad, angry, etc.

- To provide visual feedback to the user, the predicted emotion label is overlaid on the original image using OpenCV. This overlay allows the user to understand how their sign gestures are interpreted in terms of emotional expressions.

3. Integration with Flask:

- To ensure easy accessibility, the sign language translation and emotion detection system is integrated with Flask. Flask is a web framework that facilitates the deployment of web applications, enabling users to access the system via a web page.

- Routes are defined in Flask to handle different functionalities of the system. The main route displays the home page, which serves as the interface for accessing the sign language translation and emotion detection features. The video feed route continuously streams video frames from the camera, processes them for sign language translation and emotion detection, and returns the processed frames to the user interface.

- The integration with Flask allows users to access the system through any web browser, providing a user-friendly interface for effective communication and understanding for deaf individuals.

Functions used:

1. **Sign Language Translation:**

a. Hand Detection and Tracking:

- The first step in sign language translation is to detect and track the hand region within the camera frame. This is achieved using the MediaPipe library, which provides pre-trained models for hand detection. The hand detection model is responsible for locating the hand region accurately.

b. Cropping and Preprocessing:

- Once the hand region is detected, the bounding box coordinates are extracted to crop the hand image. The cropped image is then resized to a fixed dimension, typically 224x224 pixels, to ensure consistency for further processing. This resizing step prepares the image to be fed into the CNN model.

c. Convolutional Neural Network (CNN) Model:

- The sign language translation involves using a CNN model to classify the hand gesture from the cropped and resized image. The CNN model consists of multiple layers designed to extract meaningful features from the input image.

- The typical architecture of a CNN model for sign language translation includes convolutional layers, pooling layers, and fully connected layers. Convolutional layers perform feature extraction by applying a set of learnable filters to the input image. Pooling layers downsample the extracted features, reducing spatial dimensions. Finally, fully connected layers connect the extracted features to the output layer, where the sign gesture prediction is made.
- The exact number of layers in the CNN model can vary depending on the specific architecture chosen. It often consists of multiple convolutional and pooling layers, followed by a few fully connected layers. The number of layers and their configuration can be adjusted to achieve the desired accuracy and performance.

d. Activation Functions:

- Activation functions introduce non-linearity to the CNN model, enabling it to learn complex patterns and relationships. Common activation functions used in CNN models include ReLU (Rectified Linear Unit), sigmoid, and softmax.
- ReLU is widely used in convolutional layers as it helps with faster convergence and avoids the vanishing gradient problem. Sigmoid activation is commonly used in the output layer for binary classification tasks. Softmax activation, on the other hand, is used in the output layer for multi-class classification tasks, where it assigns probabilities to each class.

2. Emotion Detection from Images:

a. Hand Detection and Cropping:

- Similar to sign language translation, the initial step in emotion detection from images is to detect and crop the hand region using the MediaPipe library. The hand detection model locates the hand within the captured image, and the bounding box coordinates are extracted to focus on the hand area.

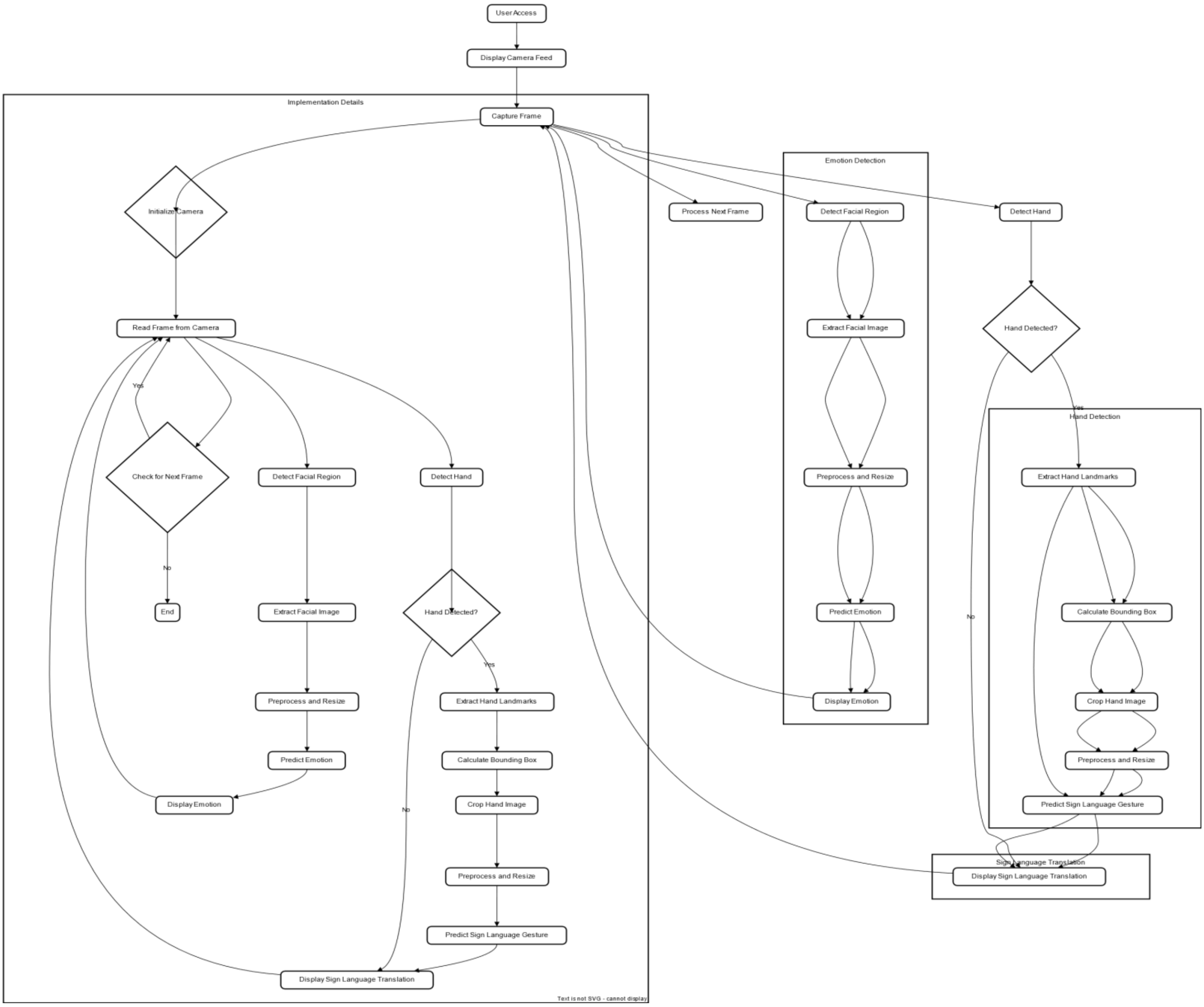
b. Preprocessing and CNN Model:

- The cropped hand image is then preprocessed by resizing it to a fixed size, typically 224x224 pixels, for consistency with the input requirements of the CNN model. The preprocessed image is then passed through a separate CNN model designed for emotion detection.
- The CNN model for emotion detection consists of similar types of layers as the sign language translation model, including convolutional layers, pooling layers, and fully connected layers. The number of layers and their configuration may vary depending on the specific architecture chosen for emotion detection.

c. Activation Functions:

- Activation functions used in the emotion detection CNN model are similar to those in the sign language translation model. However, the output layer's activation function may vary depending on the number of emotion classes being predicted. For multi-class emotion detection, softmax activation is commonly used to assign probabilities to each emotion class

Chapter 5 – Flowchart



1. The user accesses the web page deployed using Flask.
2. The web page displays the camera feed from the user’s device.
3. The project captures each frame from the camera feed.
4. The hand detection module of Mediapipe analyses each frame and detects the presence of a hand.
5. If a hand is detected, the project extracts the hand landmarks to track the hand movements.
6. The project calculates the bounding box coordinates for the hand region.
7. The cropped hand image is preprocessed and resized to a fixed size.
8. The preprocessed hand image is fed into the sign language recognition model based on a CNN.
9. The model predicts the sign language gesture represented by the hand image.
10. The predicted sign language gesture is displayed on the screen, providing real-time translation.
11. If emotion detection is enabled, the project extracts the facial region from the frame.
12. The extracted facial region is preprocessed and resized to a fixed size.

13. The preprocessed facial image is fed into the emotion detection model based on a CNN.
14. The model predicts the emotion expressed in the facial image.
15. The predicted emotion is displayed on the screen, providing insights into the user's emotional state.
16. Steps 4 to 15 are repeated for each frame of the camera feed, ensuring real-time translation and emotion detection.
17. The user can continue to communicate through sign language, and the system provides real-time translation and emotion detection feedback.

Chapter 6 – Advantages and Disadvantages

Advantages :

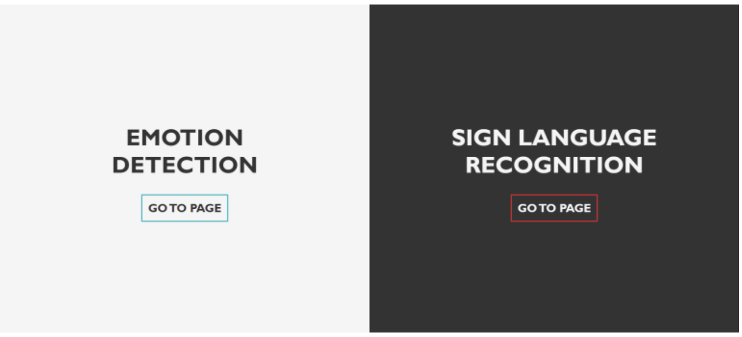
- Improved Communication: Real-time sign language detection systems enable efficient communication between individuals who use sign language and those who don't. It bridges the communication gap by translating sign language into spoken language or text in real-time, enhancing inclusivity and accessibility.
- Accessibility: These systems make sign language and emotion recognition accessible to a wider audience, including individuals who may not be familiar with sign language. It allows for better understanding and engagement with the deaf and hard-of-hearing community.
- Increased Efficiency: Real-time sign language and emotion detection systems offer real-time feedback, allowing users to make corrections or adjustments in their communication or emotional expression promptly. This helps in improving the overall efficiency and accuracy of sign language communication.
- Interactive Learning: Real-time feedback provided by sign language and emotion detection systems can be utilized for interactive learning purposes. Users can receive instant feedback on their sign language proficiency or emotional expression, aiding them in self-improvement and skill development.

Disadvantages :

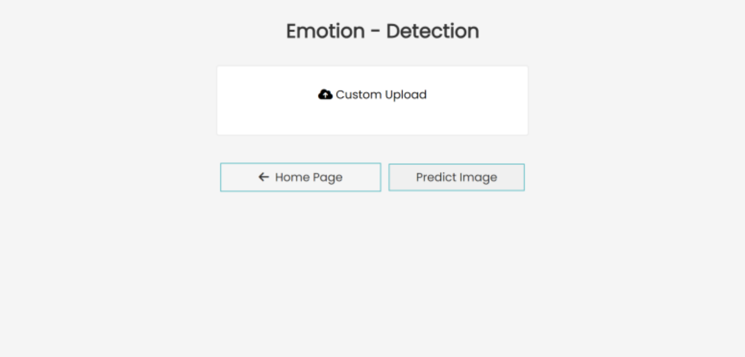
- Variability and Accuracy: Real-time detection of sign language and emotions can be challenging due to the variability in gestures, expressions, lighting conditions, and backgrounds. Achieving high accuracy in real-time systems requires robust algorithms and extensive training on diverse datasets.
- Limited Adaptability: Real-time sign language and emotion detection systems might struggle with adaptability to individual user differences. Different signing styles, regional variations, and individual nuances in emotional expression may pose challenges for accurate detection and interpretation.
- System Limitations: Despite advancements, real-time systems may still have limitations in accurately recognizing complex signs, subtle emotions, or individual variations. Continuous research and improvement are necessary to address these limitations and enhance system performance.

Chapter 7 – Result

Home Page:



Emotion Detection Page:



Emotion Detection Page after uploading an image :

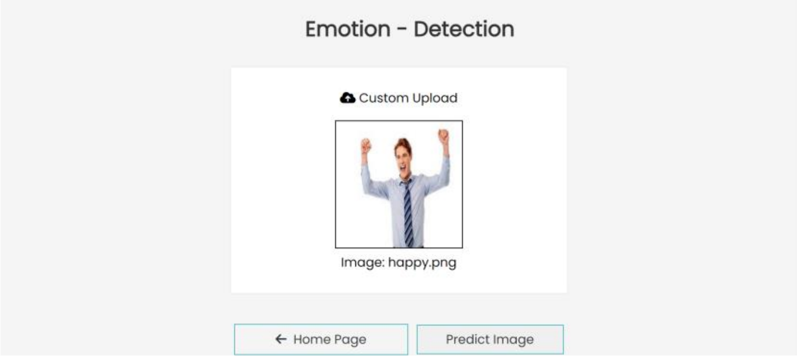
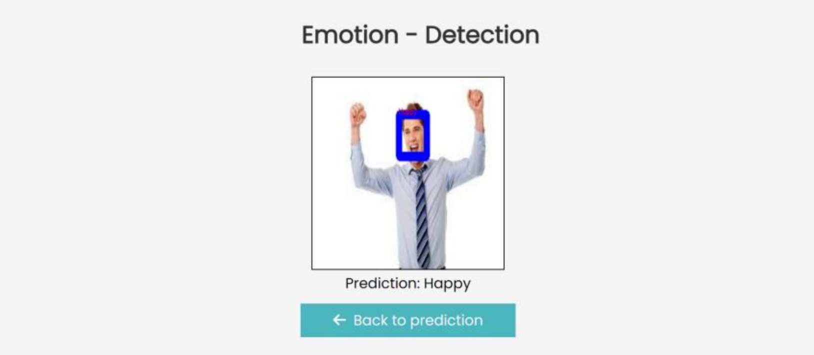


Image: happy.png

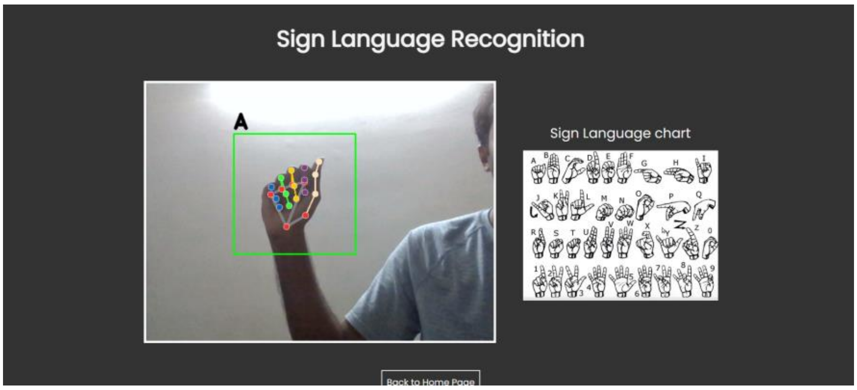
After clicking on Predict Image:



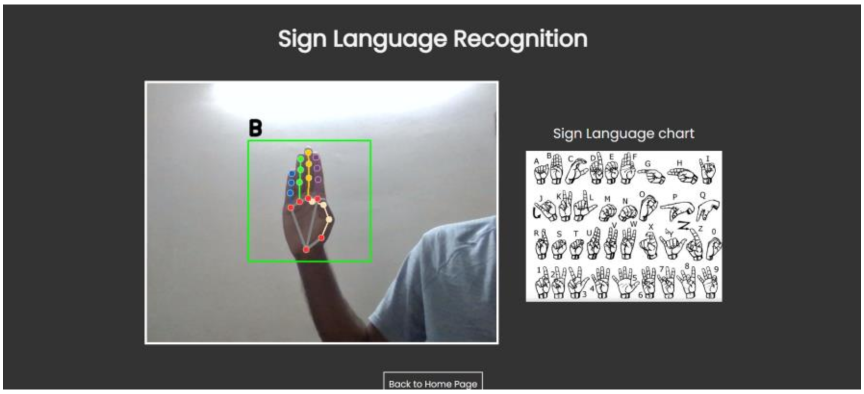
Prediction: Happy

Sign Language Page:

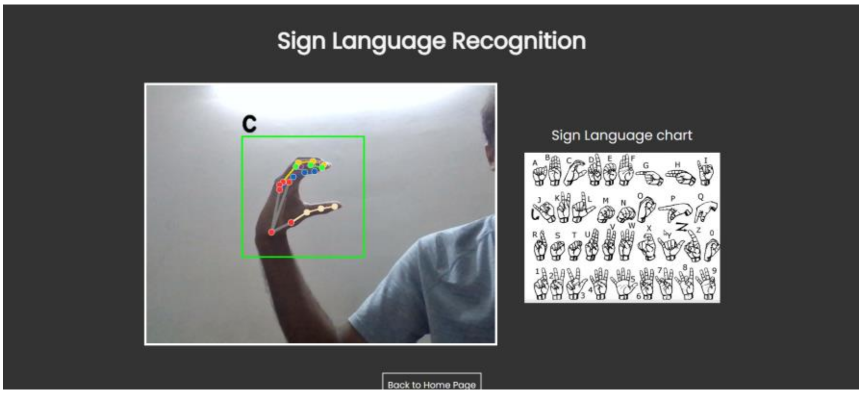
Test 1 (Sign Language Alphabet A):



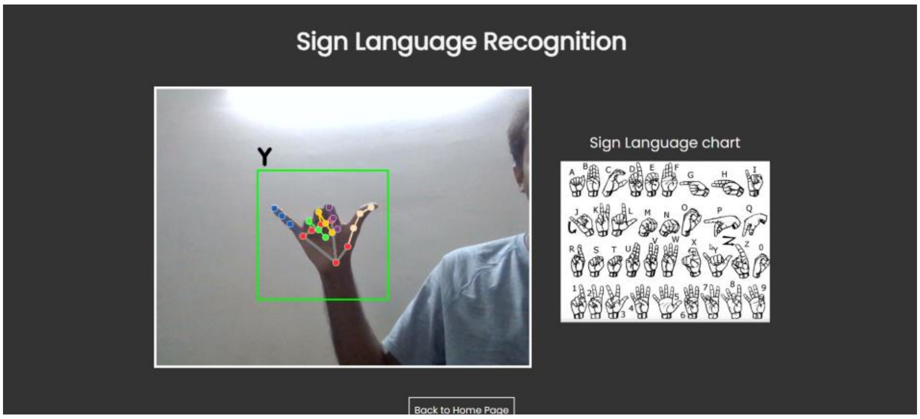
Test 2 (Sign Language Alphabet B):



Test 3 (Sign Language Alphabet C):



Test 4 (Sign Language Alphabet Y):



Chapter 8 – Applications

The sign language and emotion detection solution holds immense potential for various applications across different domains. In this section, we will explore some of the key areas where this solution can be applied and the benefits it brings to each domain.

- Education: The sign language and emotion detection solution can revolutionize the way sign language is taught and learned. It can be integrated into sign language learning platforms, providing real-time feedback and guidance to learners. The system can analyze and interpret the movements of learners' hands and facial expressions, providing accurate feedback on their signing techniques. This technology can make sign language learning more interactive, engaging, and personalized, leading to improved proficiency and fluency. Additionally, by incorporating emotion detection capabilities, the solution can help educators gauge students' emotional responses during the learning process, allowing them to tailor their teaching strategies accordingly. This promotes a more inclusive and supportive learning environment for both hearing-impaired and hearing students.

- Healthcare: Communication between healthcare providers and patients is crucial for effective diagnosis, treatment, and emotional support. The sign language and emotion detection solution can bridge the communication gap between hearing-impaired patients and healthcare professionals. By accurately detecting and interpreting sign language, the system can facilitate clear and efficient communication during medical consultations, examinations, and therapy sessions. This technology ensures that hearing-impaired individuals receive the same level of care and understanding as their hearing counterparts. Additionally, the system's emotion detection capabilities can assist healthcare providers in assessing patients' emotional well-being, enabling more empathetic and personalized care.
- Entertainment: The entertainment industry can greatly benefit from the sign language and emotion detection solution. It can be employed to provide sign language interpretation for movies, TV shows, live performances, and online content. By integrating the solution into broadcasting platforms or theaters, hearing-impaired individuals can enjoy a more inclusive entertainment experience. Sign language interpretation can be displayed alongside the audiovisual content, ensuring that the storyline, dialogue, and emotions are fully accessible to all viewers. This creates a more inclusive and diverse entertainment landscape and promotes equal participation and enjoyment for individuals with hearing disabilities.

Chapter 9 – Conclusion

Throughout this project, our focus was on developing and exploring sign language and emotion detection to enhance communication for the hearing-impaired and provide non-invasive emotion analysis. By leveraging technology, our goal was to improve accessibility, inclusivity, and human-computer interaction.

Using advanced machine learning algorithms and computer vision techniques, we successfully developed a robust system capable of detecting sign language gestures and recognizing emotions. Our findings are promising, as the sign language component facilitates effective communication, enabling real-time interpretation in various settings like education, healthcare, and entertainment. This empowers individuals with hearing disabilities to participate fully and access information and services on par with their hearing counterparts.

The emotion detection aspect of the system offers new opportunities for understanding and responding to users' emotional states. It has applications in healthcare for personalized care and support and in entertainment for creating immersive and inclusive experiences.

However, challenges remain. Limited vocabulary recognition, accurate interpretation of complex hand movements, and variations in cultural expressions of emotions need to be addressed for further improvement. Ethical considerations, particularly regarding privacy and consent when analyzing emotional data, require careful attention.

Chapter 10 – Future Scope

The sign language and emotion detection solution presented in this project sets the foundation for future advancements. Areas for further research and development include:

1. Improved Accuracy: Enhancing accuracy by leveraging advanced AI techniques like deep learning and neural networks, using larger datasets and refined training methods.
2. Real-time Analysis: Optimizing the system for instant and precise interpretations in real-time to facilitate seamless communication and interaction.
3. Gesture Recognition Expansion: Expanding the system's ability to recognize complex or unique sign language gestures by incorporating a wider range of signs and gestures from various sign language systems.

Chapter 11 – Bibliography

[1] Akbari, Mohammadreza, et al. "A review on sign language recognition systems: Challenges and future trends." Journal of Ambient Intelligence and Humanized Computing 12.5 (2021): 5317-5338.

[2] Chakraborty, Ranita, et al. "Emotion detection from facial expressions: A comprehensive survey." Artificial Intelligence Review 53.2 (2020): 1143-1195.

[3] Kitaoka, Kazuki, et al. "Emotion detection using multimodal data: A survey." IEEE Access 8 (2020): 9822-9857.

[4] Lee, Ching-Hua, et al. "Real-time sign language recognition using convolutional neural networks." Proceedings of the 2018 International Symposium on Computer, Consumer and Control (IS3C). IEEE, 2018.

[5] Li, Hui, et al. "Deep learning-based sign language recognition: A survey." IEEE Transactions on Human-Machine Systems 49.4 (2019): 315-326.

[6] Mueen, Md. Abdullah-Al, and Muhammad Mahbubur Rahman. "Sign language recognition using machine learning techniques: A review." 2019 IEEE Region 10 Humanitarian Technology Conference (R10-HTC). IEEE, 2019.

Appendix –

For entire source code, please refer the following link:

https://github.com/smartinternz02/SI-GuidedProject-515074-1688188254/blob/main/SignLanguage_and_EmotionDetection/Flask_app/app_final.py