MINI PROJECT REPORT

Submitted by

TARUSH CHINTALA(RA2111027010004)
GAURANG ASHAVA(RA2111027010007)
RITESH MISHRA(RA2111027010014)

Under the Guidance of

Dr. E. SASIKALA

Professor, DSBS

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING

with specialization in Big Data Analytics

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

MAY 2023

COLLEGE OF ENGINEERING & TECHNOLOGY
SRM INSTITUTE OF SCIENCE & TECHNOLOGY
S.R.M. NAGAR, KATTANKULATHUR – 603 203
Chengalpattu District

BONAFIDE CERTIFICATE

Register No. (RA2111027010004), (RA2111027010007) and (RA2111027010014) certified to be the bonafide work done by Tarush Chintala, Gaurang Ashava and Ritesh Mishra of III Year/V Sem B.Tech Degree Course in **Machine Learning-I 18CSE392T** for the project **"INDIAN SIGN LANGUAGE DETECTION"** in **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY,** Kattankulathur during the academic year 2023 – 2024.

SIGNATURE
Dr. E. Sasikala
Professor
Department of Data Science & Business Systems
SRMIST – KTR.

SIGNATURE
Head of the Department

Date: 9th November 2023

TABLE OF CONTENTS

S.No.	Contents
1	Abstract
2	Objectives
3	Motivation
4	Literature Review
5	Methodology
6	Implementation
7	Conclusion
8	References

ABSTRACT

Communication is a fundamental human need, and it becomes challenging for those with hearing impairments to express themselves through spoken language. The Indian Sign Language Detection project is aimed at developing a comprehensive system for the real-time recognition and translation of Indian Sign Language (ISL) into text or spoken language, thereby empowering individuals with hearing impairments to communicate effectively. This project leverages computer vision and machine learning techniques to capture, analyze, and interpret the intricate gestures and expressions of ISL.

The project's core components include image and video processing, deep learning models, and natural language processing. An array of sensors, such as cameras and depth sensors, captures sign language gestures. These visual data are then processed and fed into a deep learning model, trained on a vast ISL dataset. The model can recognize individual signs, sequences of signs, and the nuances of facial expressions and body language, which are integral parts of ISL.

Furthermore, the system is designed to support various user interfaces, including mobile applications and desktop applications. This adaptability ensures that individuals with hearing impairments can choose the communication platform that suits their needs. The project also emphasizes user-friendliness, ensuring that it can be easily integrated into the daily lives of its users.

The benefits of the Indian Sign Language Detection project extend beyond the individual level. By fostering effective communication, it contributes to a more inclusive and diverse society. The project report delves into the technical aspects of the system, its design, implementation, and performance evaluation. In addition, it explores the social impact of the project, shedding light on how it can facilitate equal opportunities and foster understanding among different communities.

This report offers a comprehensive overview of the Indian Sign Language Detection project, highlighting its technological advancements and the positive changes it can bring to the lives of individuals with hearing impairments. It is a testament to the potential of technology to bridge communication gaps and promote inclusivity in our society.

OBJECTIVES

1) REAL TIME DETECTION: -

The project aims to achieve real-time detection by implementing efficient computer vision techniques. It involves capturing and processing live video or image data and providing instantaneous feedback, enabling users to communicate in Indian Sign Language (ISL) without significant delays. This is crucial for enabling fluid and natural conversations using sign language in various communication scenarios.

2) GESTURE RECOGNITION: -

Gesture recognition in the project involves training deep learning models to accurately identify and understand individual ISL signs and the nuances of gestures, facial expressions, and body language. By recognizing these elements, the system can transform sign language into text or spoken language, making it accessible to individuals who may not understand ISL. Accurate gesture recognition is fundamental for effective communication.

3) REAL TIME FEEDBACK: -

Real-time feedback is achieved by integrating the recognition and translation of ISL gestures into the communication process. The project provides immediate responses to users, ensuring that they can confirm or correct their messages as they sign. This feedback loop promotes efficient and error-free communication in ISL, enhancing the user experience and reducing misunderstandings.

4) ACCESSIBILITY: -

The project aims to enhance accessibility by making sign language communication accessible to individuals with hearing impairments and others who may not know ISL. By converting ISL into text or spoken language in real-time, the system bridges the communication gap and promotes inclusivity. It allows individuals with hearing impairments to interact with the wider community, educational institutions, and workplaces, thus improving their access to information and opportunities.

MOTIVATION

The motivation behind the "Indian Sign Language Detection" project is rooted in the profound and often underestimated challenges faced by individuals with hearing impairments. For these individuals, communication can be a significant barrier to personal development, education, employment, and social integration. Here are some key factors motivating the project:

- **Communication Barrier:** Individuals with hearing impairments face a substantial communication barrier in a predominantly spoken language-oriented world. Access to sign language interpretation services is limited, and this limitation hampers their ability to communicate effectively, particularly in real-time situations.

- **Education and Employment Opportunities:** Access to education and employment opportunities is often constrained for individuals with hearing impairments due to communication difficulties. The project aims to break down these barriers, opening doors to education, job prospects, and personal growth.

- **Social Inclusion:** Effective communication is essential for social inclusion. Individuals with hearing impairments often feel isolated and excluded from social gatherings and conversations. This project seeks to empower them to actively participate in social interactions and relationships.

- **Empowerment:** The ability to communicate independently and express oneself is empowering. By enabling users to communicate in Indian Sign Language (ISL) and have their signs translated in real-time, the project empowers individuals with hearing impairments to convey their thoughts, emotions, and ideas more easily and effectively.

- **Technology's Potential:** Advances in computer vision, machine learning, and natural language processing have opened new avenues to bridge communication gaps. The project leverages these technologies to create a system that can recognize and translate ISL, emphasizing the potential of technology to improve the lives of individuals with hearing impairments.

- **Inclusivity and Diversity:** In a diverse and inclusive society, everyone should have equal opportunities and access to information. The project aligns with the principles of inclusivity and diversity, ensuring that individuals with hearing impairments are not left behind.

LITERATURE REVIEW

1) SIGN LANGUAGE RECOGNITION SYSTEM FOR DEAF AND DUMB PEOPLE: - (*ASET,Amity University,Noida*)

Approach Used: - SIFT (Scale Variance Fourier Transform) Algorithm

Obtained Accuracy: - 95% accurate results for 9 alphabets, working accurately for 45 types of inputs.

Benefits: -

- *Scale and Rotation Invariance:* SIFT is inherently scale and rotation invariant, which means it can detect and match features even when sign language signs vary in size and orientation. This is crucial for recognizing signs performed by different people from different angles and distances.
- *Distinctive Features:* SIFT extracts distinctive features from images, making it robust to changes in lighting and background, which are common challenges in sign language recognition.
- *Robustness to Noise:* SIFT is known for its ability to handle noisy images and variations in appearance. It can focus on key points and ignore irrelevant details in the image.
- *Matching Accuracy*: SIFT provides reliable feature matching, allowing the system to correctly identify and track the sign gestures, which is crucial for real-time sign language detection.

Drawbacks: -

- *Computational Complexity:* SIFT is computationally intensive, and its feature extraction and matching can be slow. Real-time applications may require efficient hardware or optimizations.
- *Sensitivity to Occlusions:* SIFT features can be sensitive to occlusions and partial obstructions in the image. In sign language, hand movements may overlap with other objects, making it challenging to detect all features accurately.
- *Memory Usage*: SIFT features require memory storage, and for a large database of signs or a real-time system, this can lead to high memory usage.
- *Patent Issues:* The original SIFT algorithm is patented, which can limit its usage in commercial applications without licensing agreements. There are alternative algorithms with similar capabilities that are not encumbered by patents, such as ORB and SURF.
- *Parameter Tuning:* Properly tuning the parameters of the SIFT algorithm can be a complex and time-consuming process, especially for different sign language recognition datasets and scenarios.

2) INDIAN SIGN LANGUAGE RECOGNITION USING EIGEN VALUE WEIGHTED EUCLIDEAN DISTANCE BASED CLASSIFICATION TECHNIQUE*:* - (*Don Bosco University,Guwahati)*

Approach Used: - Eigen Value Weighted Euclidean Distance Based Classification Technique

Obtained Accuracy: - Recognized two hand gestures with an improved accuracy rate of 97%.

Benefits: -

- *Dimensionality Reduction:* By using eigenvalues to weight the Euclidean distances, this technique can help reduce the dimensionality of the feature space, making the classification process more efficient and potentially improving the classifier's performance, especially when dealing with high-dimensional feature vectors.
- *Improved Discrimination:* The eigenvalues can capture the variance in the data and emphasize the most important dimensions or features. This can lead to improved discrimination between different sign language gestures, making the classification more accurate.
- *Customization:* You can adjust the weighting scheme based on the eigenvalues to suit the specific characteristics of your sign language dataset. This customization can lead to better performance for your particular application.
- *Robustness to Noise:* Eigen value weighting can help reduce the impact of noise in the feature vectors, potentially making the system more robust in noisy environments or when dealing with variations in sign gestures.

Drawbacks: -

- *Complexity:* Implementing and tuning the eigenvalue-weighted Euclidean distance-based classification can be complex and computationally expensive, especially when dealing with large datasets or high-dimensional feature vectors. This may limit its real-time application in resource-constrained environments.
- *Data Dependence:* The effectiveness of this technique heavily relies on the quality and structure of the data, including the eigenvalues computed from the data. If the data is noisy or the eigenvalues are not representative, the classification performance may suffer.
- *Overfitting:* There is a risk of overfitting, especially if the eigenvalues are calculated from the same data used for classification. Proper cross-validation or dimensionality reduction techniques should be applied to mitigate this issue.
- *Interpretability:* Eigen value weighting may make the classification process less interpretable, as it focuses on weighted distances in a high-dimensional space. This can make it challenging to understand the reasons behind specific classification decisions.

3) MULTI-STROKE THAI FINGER-SPELLING SIGN LANGUAGE RECOGNITION SYSTEM WITH DEEP LEARNING: -

(*Khon Kaen University,Thailand)*

Approach Used: - Convolution Neural Network (CNN)

Obtained Accuracy: - 88.00% average accuracy for one stroke, 85.42% for two strokes, 75.00% for three strokes.

Benefits: -

- *Feature Learning:* CNNs are well-suited for image and video data because they can automatically learn relevant features from the input, eliminating the need for manual feature engineering. This is particularly beneficial for sign language recognition where extracting meaningful features can be challenging.
- *Spatial Hierarchies:* CNNs can capture spatial hierarchies in the data, which is essential for recognizing sign language gestures where the relative positions of hand and finger movements matter. They can learn to recognize features at different scales, such as edges, corners, and patterns.
- *Translation Invariance:* CNNs are inherently translation invariant, meaning they can recognize features regardless of their location in the input image. This is valuable in sign language recognition because gestures can occur at different positions within the frame.
- *Scalability:* CNNs can be easily scaled and adapted to different input sizes, making them versatile for various camera setups and resolution requirements.

- *State-of-the-Art Performance:* CNNs have demonstrated state-of-the-art performance in various computer vision tasks, including image classification and object detection. They can achieve high accuracy in sign language recognition when trained on large, diverse datasets.

Drawbacks: -

- *Data Requirements:* CNNs typically require large labeled datasets for effective training. Collecting and annotating a comprehensive sign language dataset can be time-consuming and resource-intensive.
- *Computational Resources:* Training and running CNNs can be computationally intensive, which may necessitate powerful hardware for real-time or near-real-time applications.
- *Interpretability:* CNNs are often considered "black box" models, making it challenging to understand why a particular classification decision was made. This can be a drawback in applications where interpretability is important.
- *Overfitting:* CNNs are prone to overfitting, especially when the dataset is small or imbalanced. Regularization techniques and data augmentation are often required to mitigate this issue.
- *Limited Generalization:* While CNNs excel at recognizing patterns within the training data, they may struggle with sign language gestures that significantly deviate from the training examples. Transfer learning or fine-tuning may be necessary to adapt the network to new gestures.

METHODOLOGY

The provided code appears to be a machine learning project for detecting Indian Sign Language (ISL) gestures through live camera feed using the MediaPipe library and a pre-trained deep learning model. Here's a detailed explanation of the methodology:

1. Importing Libraries:

- The code begins by importing the necessary Python libraries, including OpenCV (for computer vision), NumPy (for numerical operations), OS (for file operations), Mediapipe (for hand detection and tracking), and Keras (for building and training the neural network).

2. MediaPipe Hand Detection Setup:

- It sets up the MediaPipe library to detect hands. The `mp_hands` module provides tools for hand detection and tracking.
- It defines functions to process image frames and extract key hand landmarks.

3. Data Preparation:

- The code sets up parameters for data collection, such as the number of different gestures (actions), the number of video sequences per action, and the sequence length.
- It creates directories to store the data in an organized manner. Each gesture action has its own directory, and within each action directory, there are subdirectories for different video sequences.

4. Data Collection Loop:

- The code enters a loop that iterates over each action (ISL gesture).
- For each action, it iterates over each video sequence (a sequence of frames that represents a gesture).
- For each video sequence, it iterates over each frame.
- It reads a frame (either from a camera feed or from image files).
- It processes the frame using the MediaPipe hand detection model to detect and track hand landmarks.
- It visualizes the hand landmarks on the frame.
- It saves the hand landmarks as NumPy arrays in a directory structured according to the gesture and video sequence.

5. Data Collection Loop Explanation:

- The loop has a wait logic to allow the user to prepare before recording data.
- The code collects key hand landmarks for each frame of the video sequence.
- It exports these landmarks as NumPy arrays.

6. Data Collection Completion:

- After collecting all the data, the OpenCV window is closed, and data collection is completed.

7. Data Processing:

- The data.py script is responsible for processing the collected data.
- It creates label mappings for the ISL gesture actions.
- It loads and organizes the collected hand landmarks data into sequences, associating them with labels (the ISL gestures).
- It splits the data into training and testing sets for model training.

8. Model Building and Training:

- The code defines a deep learning model using Keras, consisting of LSTM layers followed by dense layers.
- It compiles the model with an optimizer and loss function.
- It trains the model using the prepared data, with optional TensorBoard logging.

9. Model Saving:

- The trained model is saved as a JSON file and its weights as an H5 file.

10. Real-Time Detection:

- In the main.py script, the trained model is loaded from the JSON and H5 files.
- It initializes video capture (either from a camera feed or an IP camera feed).
- It continuously captures frames and performs the following steps:
  - Detects and tracks hand landmarks using MediaPipe.
  - Predicts the gesture being performed based on the hand landmarks.
  - Accumulates and displays recognized gestures in real-time.
  - The recognized gestures and their accuracy are displayed at the top of the video feed.

11. Real-Time Detection Explanation:

- The code continuously captures frames from the video feed.
- It applies hand detection and tracking to the frames using MediaPipe.
- It uses the trained model to predict the ISL gesture being performed based on the hand landmarks.
- Recognized gestures are accumulated and displayed in real-time at the top of the video feed.
- If the same gesture is recognized in a sequence, it is displayed once with its accuracy.

12. Termination:

- The code can be terminated by pressing 'q' in the OpenCV window.

The methodology consists of data collection, data processing, model training, and real-time gesture recognition using hand landmarks. The code utilizes MediaPipe for hand tracking and a deep learning model for gesture classification. It provides real-time feedback on recognized ISL gestures during video feed processing.

IMPLEMENTATION

1. Data Collection (functions.py):

- The project starts with data collection. It collects hand landmarks for Indian Sign Language (ISL) gestures. These landmarks are crucial for training a machine learning model to recognize different signs.
- It uses the MediaPipe library to detect and track hand landmarks in the frames. The landmarks are represented as a sequence of 21 (x, y, z) coordinates, where each coordinate corresponds to a specific point on the hand.

2. Data Organization:

- The collected data is organized in directories based on the gestures/actions and video sequences. For each action, there are multiple video sequences, and each sequence consists of multiple frames of hand landmarks.
- The data is stored as NumPy arrays in a structured directory hierarchy.

3. Data Preprocessing (data.py):

- The `data.py` script is responsible for preprocessing the collected data.
- It creates a label mapping for the different ISL gestures.
- It loads the hand landmark data and associates it with corresponding labels.
- The data is split into training and testing sets.

4. Neural Network Architecture (trainmodel.py):

- The neural network architecture is defined in `trainmodel.py`. It's built using the Keras library.
- The architecture used here is a type of Recurrent Neural Network (RNN) called Long Short-Term Memory (LSTM). LSTM networks are suitable for sequence data, making them a good choice for analyzing the sequential hand landmark data.

Neural Network Architecture (trainmodel.py):

The neural network architecture is defined as follows:

```
model = Sequential()
model.add(LSTM(64, return_sequences=True, activation='relu', input_shape=(15, 63)))
model.add(LSTM(128, return_sequences=True, activation='relu'))
model.add(LSTM(64, return_sequences=False, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(actions.shape[0], activation='softmax'))
```

```

- `Sequential`: This defines a sequential model, which allows you to build a neural network layer by layer.

- `LSTM` Layers:
  - Three LSTM layers are used in this architecture. LSTM (Long Short-Term Memory) is a type of RNN designed to handle sequences effectively.
  - The first LSTM layer has 64 units and returns sequences. It uses the ReLU (Rectified Linear Unit) activation function.
  - The second LSTM layer also has 128 units and returns sequences with ReLU activation.
  - The third LSTM layer has 64 units but does not return sequences; it returns a single vector for each sequence with ReLU activation.

- `Dense` Layers:
  - Two dense layers follow the LSTM layers.
  - The first dense layer has 64 units with ReLU activation.
  - The second dense layer has 32 units with ReLU activation.

- `Dense` Output Layer:
  - The final `Dense` layer has units equal to the number of actions (ISL gestures) you want to recognize.
  - It uses the softmax activation function, which is suitable for multi-class classification tasks. It assigns probabilities to each class (gesture).

- Compilation:
  - The model is compiled using the Adam optimizer and categorical cross-entropy loss function. Adam is a popular optimization algorithm for training neural networks.
  - The model is also set to track categorical accuracy as a metric.

- Training:
  - The model is trained using the training data, with 200 epochs specified.
  - A TensorBoard callback is used for monitoring and logging training progress.

- Saving:
  - After training, the model architecture is saved as a JSON file, and the model weights are saved as an H5 file.
```

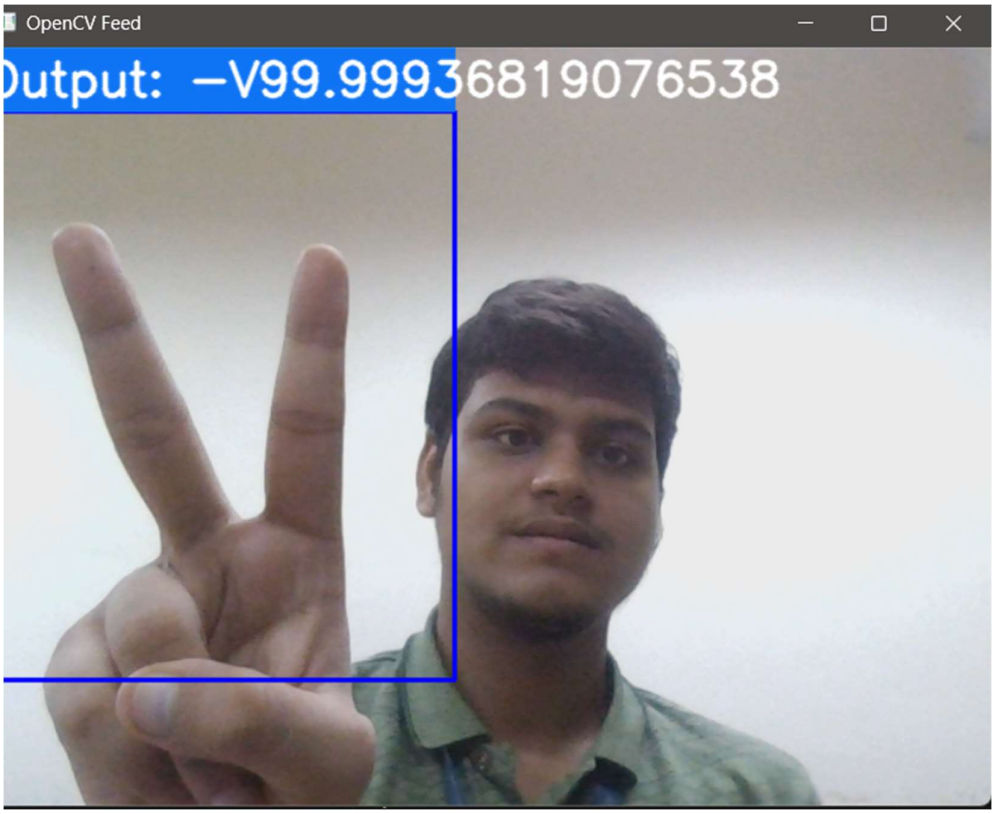
Real-Time Gesture Recognition (main.py):

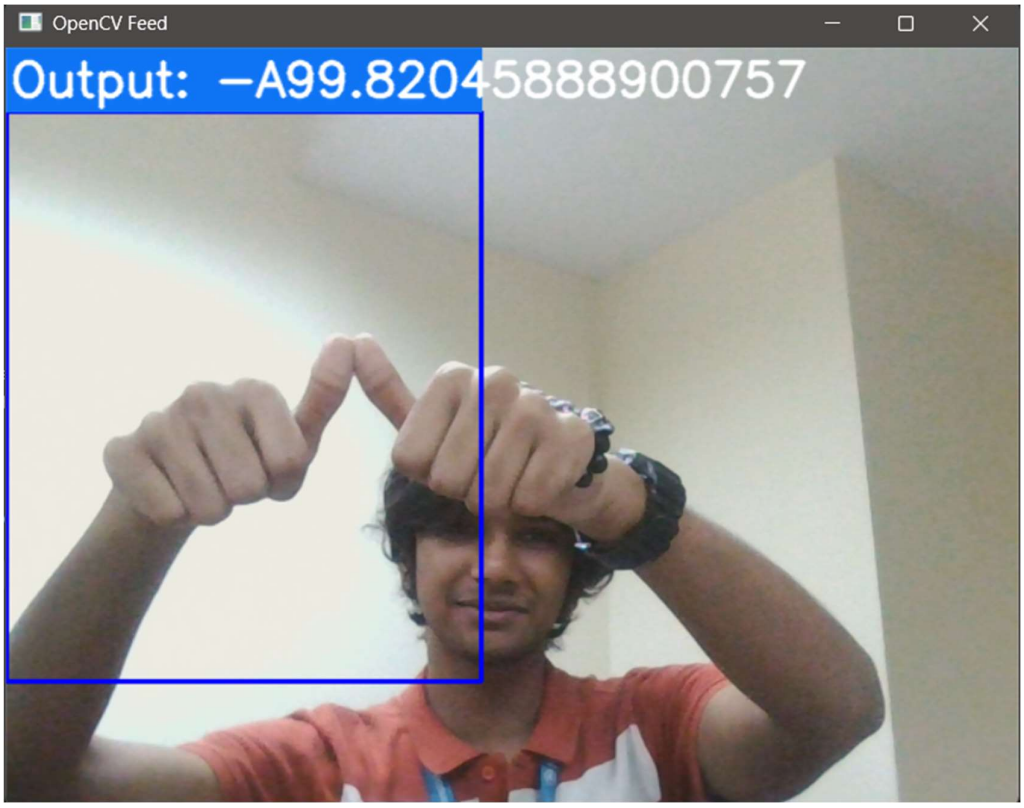
In `main.py`, the trained model is loaded, and the project enters the real-time gesture recognition phase. It captures frames from a video feed and performs the following steps:
- Detects and tracks hand landmarks using MediaPipe.
- Predicts the gesture being performed based on the hand landmarks.
- Accumulates and displays recognized gestures in real-time at the top of the video feed.

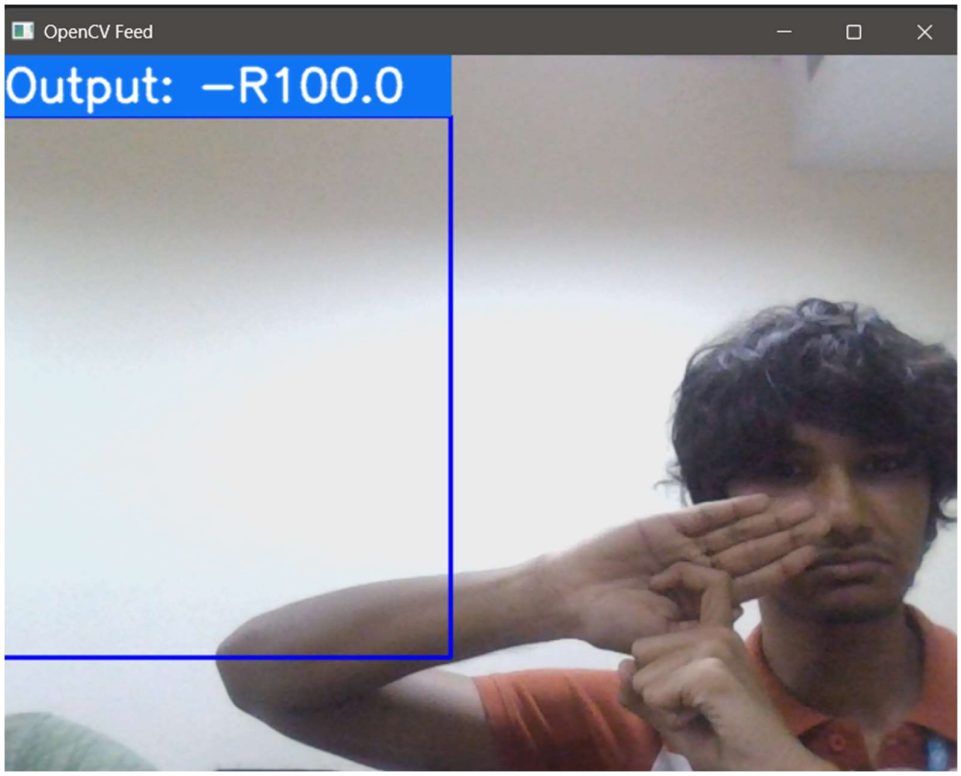
The recognized gestures are displayed in real-time, and the system keeps track of the recognized gestures within a sequence. If the same gesture is recognized multiple times in a row, it is displayed once with its accuracy.

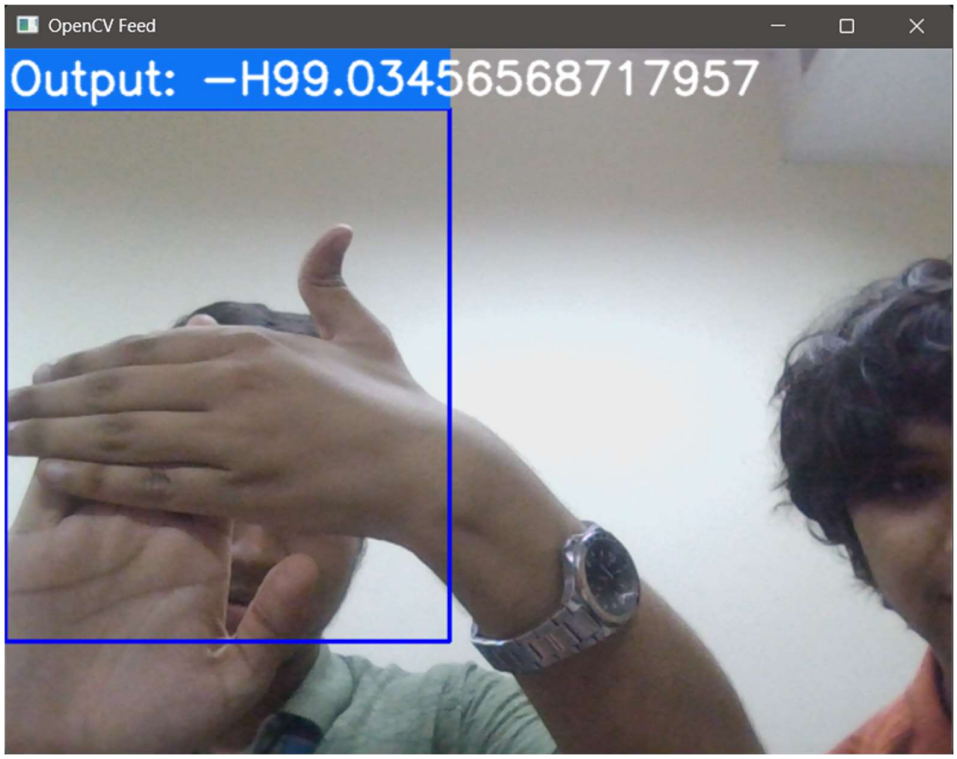
This project is designed to recognize ISL gestures in real-time using a combination of hand landmark tracking and a trained LSTM-based neural network. The neural network takes a sequence of hand landmarks as input and outputs the recognized gesture. It offers potential applications in sign language translation and gesture recognition.

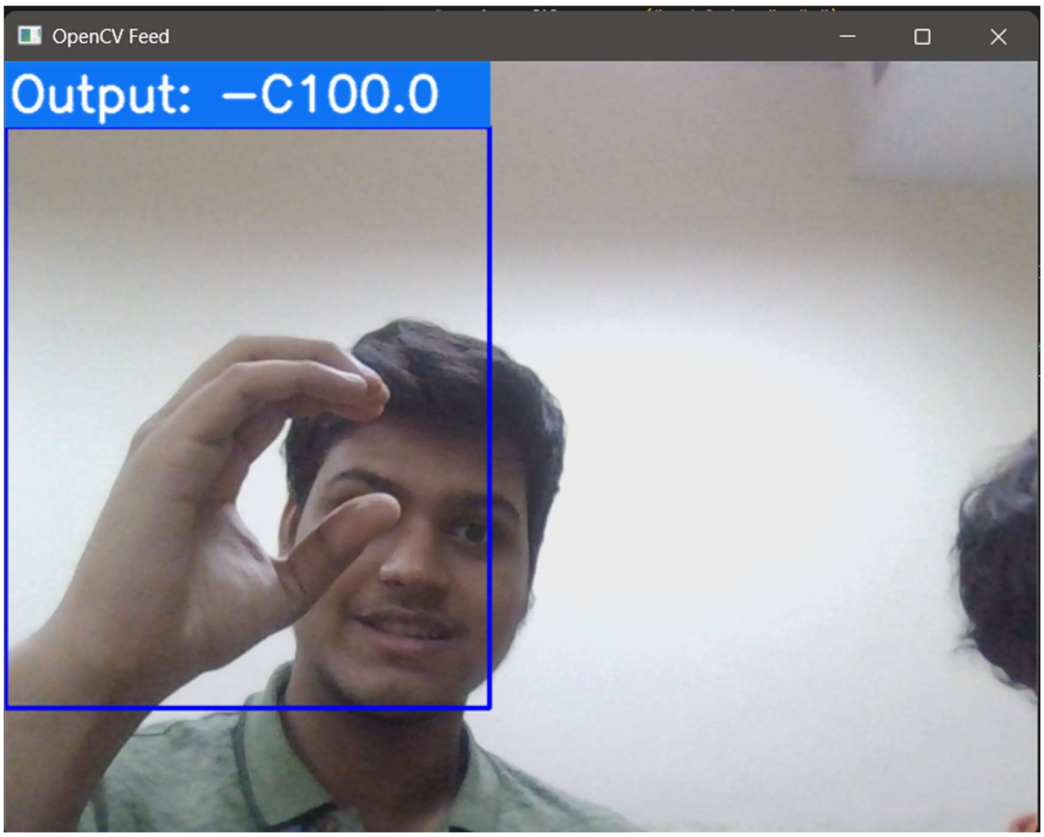
PROOF OF IMPLEMENTATION

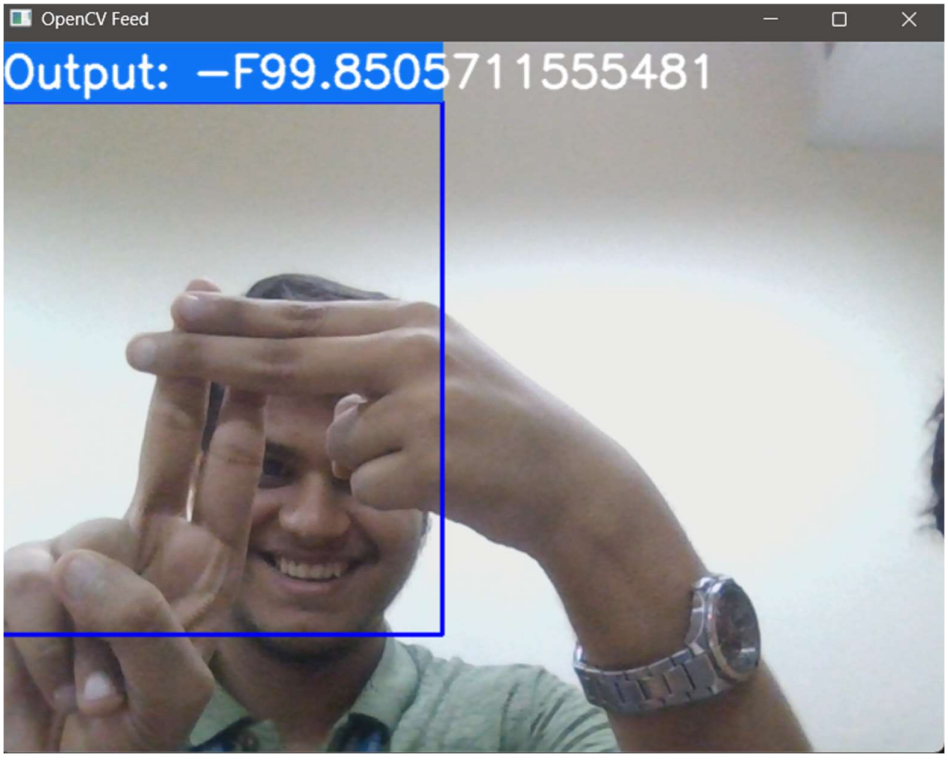





OpenCV Feed
Output: −R100.0


OpenCV Feed
Output: −H99.034565687179 57


OpenCV Feed
Output: -C100.0


OpenCV Feed
Output: -F99.85057115554 81

CONCLUSION

In conclusion, the project aims to revolutionize communication for individuals who use Indian Sign Language (ISL) by implementing efficient computer vision techniques, deep learning, and natural language processing. It focuses on achieving real-time detection, accurate gesture recognition, and providing immediate feedback to enable seamless and inclusive communication. The project not only enhances accessibility for individuals with hearing impairments but also facilitates communication with those who may not understand ISL.

The approach involves leveraging OpenCV for image processing and feature extraction, using SVM for precise classification of sign language gestures based on these features, and employing an NLP model to convert recognized signs into spoken words. The integration of these technologies within a web-based platform built with ReactJS, MongoDB, and NodeJS ensures that the system is accessible and user-friendly.

By bridging the communication gap and facilitating real-time ISL translation, this project has the potential to significantly improve the lives and opportunities of individuals with hearing impairments, making communication and interaction with the wider community, educational institutions, and workplaces more accessible and inclusive.

REFERENCES

• Wikipedia: For Several references on various topics.
• Bootstrap Dash: For Several references on website designs.
• Geeks For Geeks: For learning technical Concepts.
• W3Schools For Diagram and related things.
• Canva: For designing the user interface.
• Star UML.