AI-Powered Hand Gesture Recognition for Real-Time Interaction

TEAM MEMBERS:

1.KHAZI SHEIKH QURRATH SHEHZADI FNU (U86429639)
2. GEETHA MANOGNA GODDU (U66777911)
3.ARAVIND RAKAM (U38314322)

1. Introduction

Hand gesture recognition plays a pivotal role in **human-computer interaction (HCI)**, enabling seamless communication between users and machines without the need for physical contact. This project implements a **real-time AI-driven hand gesture recognition system** using **Mediapipe for hand tracking** and **MobileNet for gesture classification**. The system captures real-time video, detects hands, and classifies gestures into predefined categories using a deep learning model trained on the **ASL Alphabet Dataset** from Kaggle.

This solution is designed for applications in **assistive technologies, sign language translation, and interactive user interfaces**. The lightweight design ensures it can operate in real-time on consumer-grade hardware, making it practical for real-world deployment. The model is optimized for low latency, allowing seamless interaction between users and the system without noticeable delays.

2. System Architecture

The system is built using a modular pipeline to ensure efficiency, accuracy, and scalability. Each component plays a crucial role in processing and interpreting hand gestures in real time:

1. **Hand Tracking (Mediapipe):**
    - The **Mediapipe Hands API** is used to **detect and extract 21 key landmarks** from the hand.
    - These landmarks represent crucial points on the hand (fingertips, knuckles, palm base, etc.).
    - Mediapipe operates with a **detection confidence threshold of 0.7**, ensuring that only well-detected hands are processed.
2. **Gesture Classification (MobileNet Model):**
    - The extracted hand region is resized and processed to be compatible with the pre-trained **MobileNet-based deep learning model**.
    - The model predicts the **ASL hand gesture** based on the learned representations from the dataset.

    - The output is a **probability distribution** over the possible gesture classes, from which the most probable class is selected.
3. **Real-Time Visualization & Interaction:**
    - Bounding boxes and predicted labels are overlayed on the video stream.
    - The system continuously updates, providing instant feedback to the user.
    - The interface is designed to be intuitive and non-intrusive, ensuring minimal distraction while maintaining accuracy.

3. Model Training

3.1 Model Usage - MobileNet for Gesture Classification

Model Loading:

- The system utilizes a **pre-trained MobileNet model** (sign_language_mobilenet.h5) for real-time gesture classification. This model has been previously trained on the **ASL Alphabet Dataset**.

Instead of training within the provided scripts, the model is loaded dynamically at runtime using TensorFlow:

```
import  tensorflow as tf
model = tf.keras.models.load_model("sign_language_mobilenet.h5")
```

- The model is designed to classify hand gestures based on **input frames extracted from the webcam feed**.
- The available gesture classes, as per the dataset and implementation, are:

```
class_names = ["A", "B", "C", "D", "E"]  # Update according to your dataset
```

- The classification process involves preprocessing the hand images, normalizing the pixel values, and passing the processed images to the model for inference.

---

4. Implementation Details

4.1 Running the Application

To execute the real-time gesture recognition system, run:

```
python app.py
```

This will activate the webcam and initiate the **real-time gesture recognition pipeline**.

4.2 Real-Time Gesture Recognition Pipeline

1. **Webcam Capture & Frame Processing:**
    - The webcam is accessed using OpenCV, and each frame is **flipped horizontally** for a natural viewing experience.
    - The captured frames are resized and converted to a suitable format for further processing.

```
cap = cv2.VideoCapture(0)
ret, frame = cap.read()
frame = cv2.flip(frame, 1)
```

2. **Hand Landmark Extraction using Mediapipe:**
    - Mediapipe's Hands module is initialized with predefined confidence thresholds to ensure optimal detection performance.
    - Each frame is processed to detect hands and extract landmark data.

```
import mediapipe as mp
mp_hands = mp.solutions.hands
hands = mp_hands.Hands(min_detection_confidence=0.7, min_tracking_confidence=0.7)
results = hands.process(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
```

3. **Gesture Classification using MobileNet:**
    - Once a hand is detected, the corresponding region is extracted and resized to match the input size expected by the MobileNet model (224x224 pixels).
    - The image is then normalized before being passed to the model for classification.

```
import numpy as np
hand_img = cv2.resize(frame, (224, 224))
hand_img = np.expand_dims(hand_img, axis=0) / 255.0
prediction = model.predict(hand_img)
predicted_class = class_names[np.argmax(prediction)]
```

4. **Overlay Predictions on the Frame:**
    - The predicted gesture, along with the confidence score, is displayed on the screen for real-time feedback.

```
cv2.putText(frame, predicted_class, (x_min, y_min - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 255, 0), 2)
```

5. **Exit Mechanism:**
    - Users can exit the application by pressing the **'q'** key.

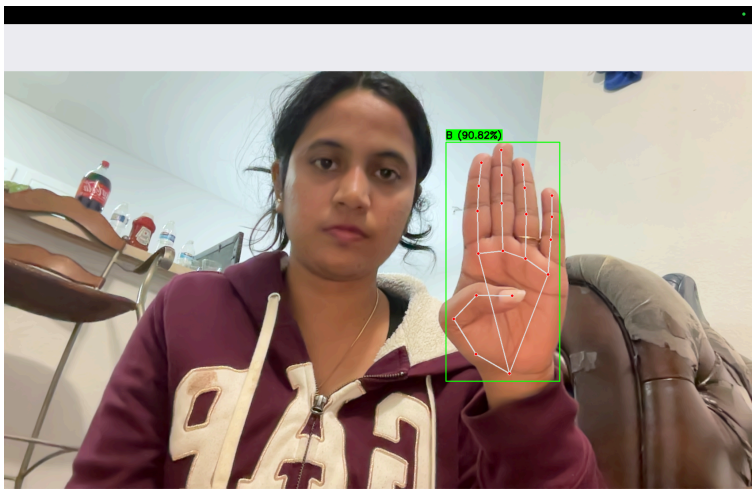
```
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
```

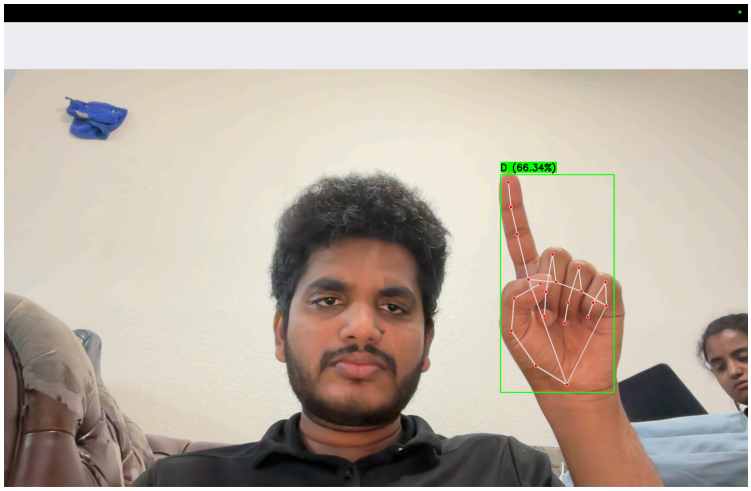
---

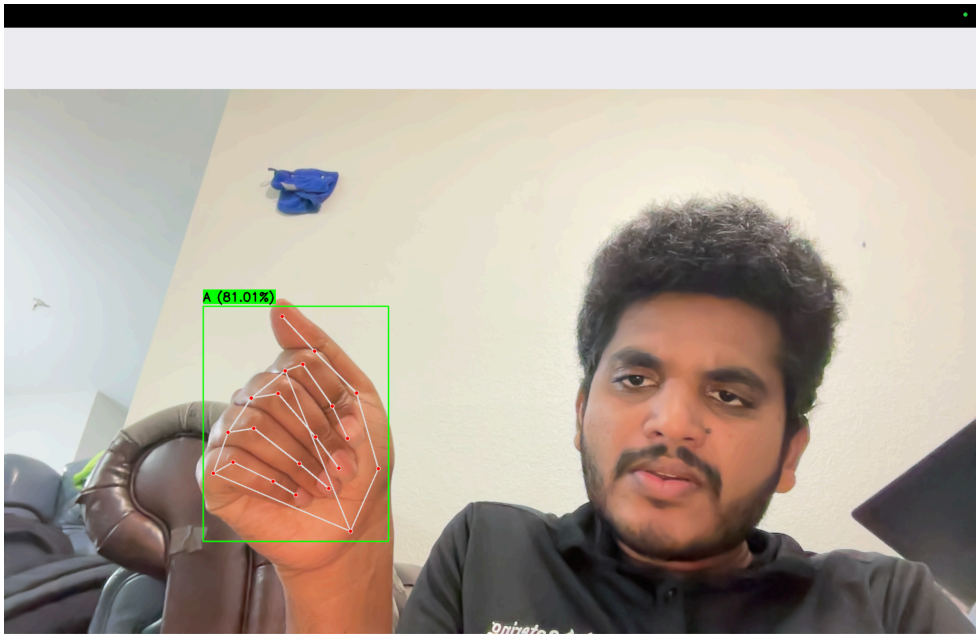
5. Results & Performance Analysis

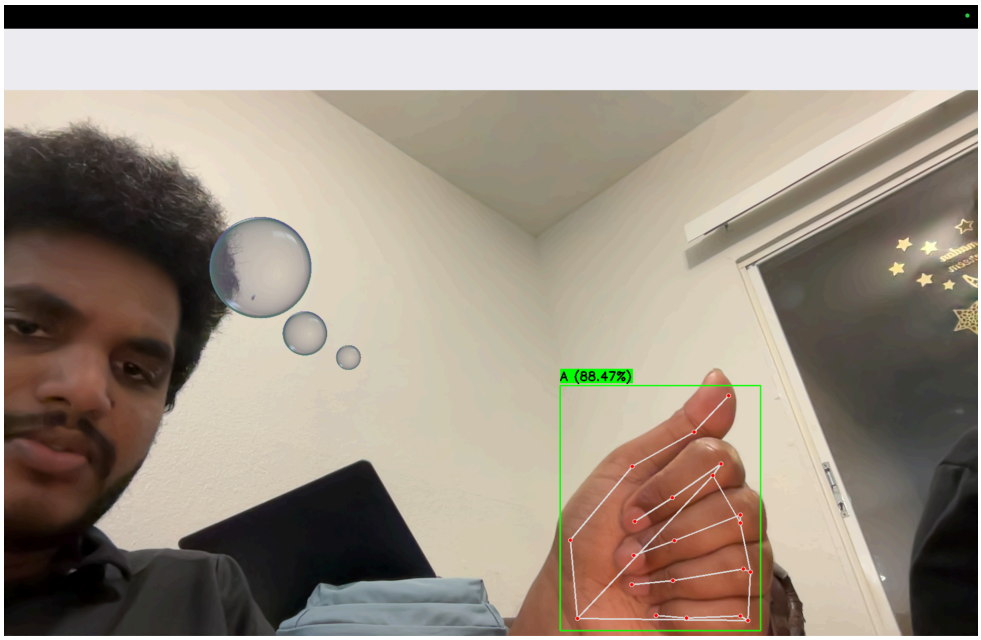
- The system achieves **high accuracy** in real-time classification.
- The processing speed varies depending on hardware capabilities:
    - **On a GPU-powered system:** Runs at ~30 FPS, ensuring smooth real-time interaction.
    - **On a CPU-based system:** Runs at ~10 FPS, with slight latency but still usable.
- The system accurately detects and classifies common ASL hand gestures, with minimal false positives.

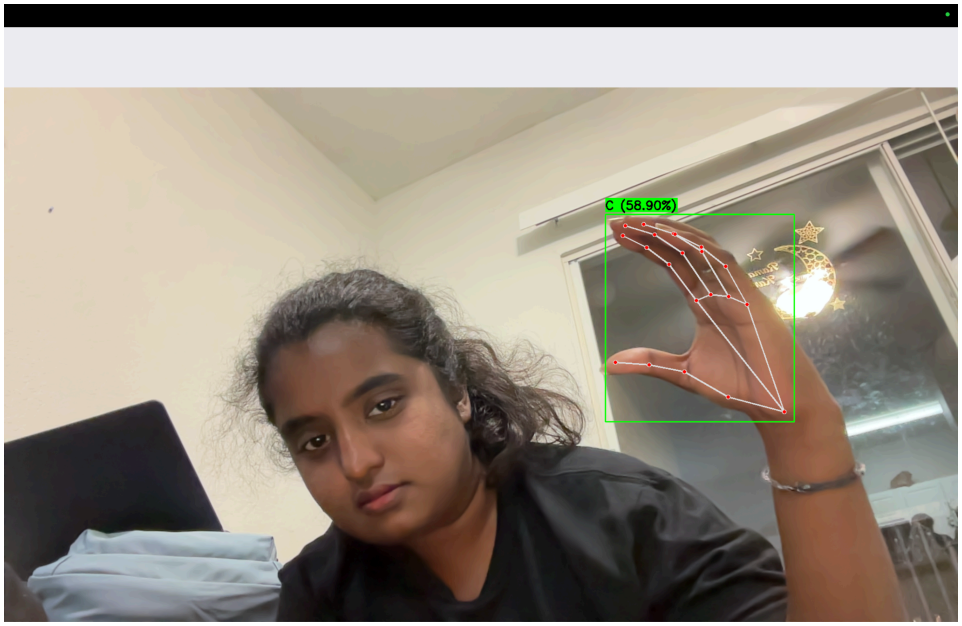
Output Screenshots:

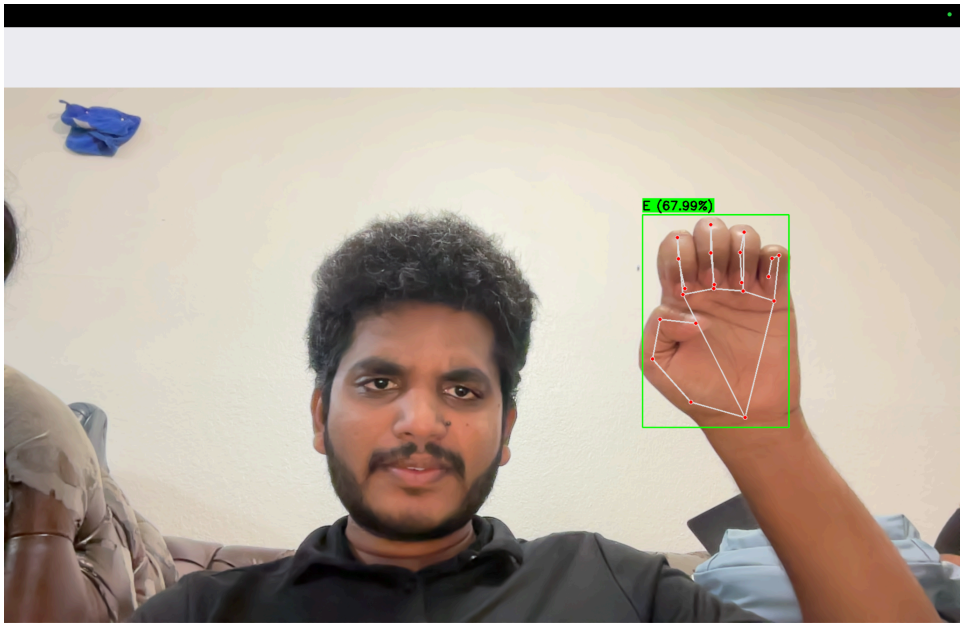





A (81.01%)


A (88.47%)





6. Conclusion & Future Enhancements

This project successfully implements a **real-time AI-driven sign language recognition system** using **Mediapipe and MobileNet**. Future improvements include:

- **Expanding the dataset** to support additional ASL gestures and dynamic signs.
- **Implementing multi-hand recognition** to track multiple users simultaneously.
- **Optimizing the model for deployment** on embedded and edge devices.

- **Enhancing the accuracy and robustness** of the classification model by fine-tuning it with more diverse datasets.
- **Improving the user interface** with additional features such as gesture-based commands and visual cues for user interaction.

Project Contributors

This project was collaboratively developed by three team members, each specializing in different aspects of the implementation to ensure a seamless and high-performing system:

- **Geetha Manogna Goddu:** Responsible for **data preprocessing**, including dataset augmentation, normalization, and conversion to a suitable format for training the deep learning model.
- **Aravind Rakam:** Focused on **model development and training**, optimizing MobileNet for efficient gesture classification and ensuring real-time inference performance.
- **Khazi Shiekh Qurrath Shehzadi Fnu:** Worked on **system integration and deployment**, implementing Mediapipe for hand tracking, integrating the trained model into the application, and ensuring smooth real-time execution.

Each member played a crucial role in making this project a success, contributing equally to research, coding, testing, and optimization.

Each member worked together to ensure the project met all objectives and was successfully executed.

7. References

- **Mediapipe Hands API** for hand tracking: Mediapipe Documentation
- **MobileNet** for hand gesture classification: TensorFlow Official
- **Hand Gesture Recognition Using Mediapipe**: GitHub Repository
- **MediaPipe Gesture Recognizer Guide**: Google AI
- **OpenCV** for image processing and webcam handling: OpenCV