REAL TIME SIGN LANGUAGE DETECTION

PALAK SHARMA . SANA FIRDOUS .SHRADDHA SUDHAKARAN

NOVELTY

01 Multi-Signer Real-Time Recognition

- Unlike traditional models that focus on single-sign recognition, our system distinguishes multiple signers simultaneously and generates separate sentences for each.
- This enhances real-world usability, especially in group conversations or classrooms.

02 Robustness to Low-Quality Videos

- Existing models struggle with poor lighting, motion blur, and low-resolution footage.
- Our dataset will include low-quality and noisy videos to improve generalization and ensure reliable recognition in diverse environments.

03 Advanced Real-Time Detection with YOLOv8

- We integrate YOLOv8 for real-time person detection, ensuring accurate tracking of signers before processing gestures.
- This improves performance by reducing misclassifications and focusing only on actual signers, ignoring background distractions.

04 Focus-Driven Sign Language Translation

- Our app will utilize advanced AI to accurately detect and translate sign language in real time. It intelligently isolates the signer, ignoring background movement and distractions, ensuring seamless and precise translation even in crowded environments.

MOTIVATION

01 Bridging the Communication Gap for the Deaf Community

- Millions of deaf and hard-of-hearing individuals face daily communication barriers.
- Most people do not understand sign language, limiting accessibility in education, workplaces, and social interactions.

02 Enhancing Accessibility in Real-World Scenarios

- Existing sign language models struggle with real-world conditions like multiple signers, poor lighting, and background noise.
- By training on multi-person and low-quality videos, we ensure better recognition accuracy in diverse environments (e.g., public spaces, hospitals, schools).

OBJECTIVES

Empowering the Deaf and Speech-Impaired Community

This project aims to provide a meaningful tool for individuals who rely on sign language, helping them communicate more confidently in environments where sign language is not commonly understood. It's about giving them independence, dignity, and a stronger voice in everyday interactions.

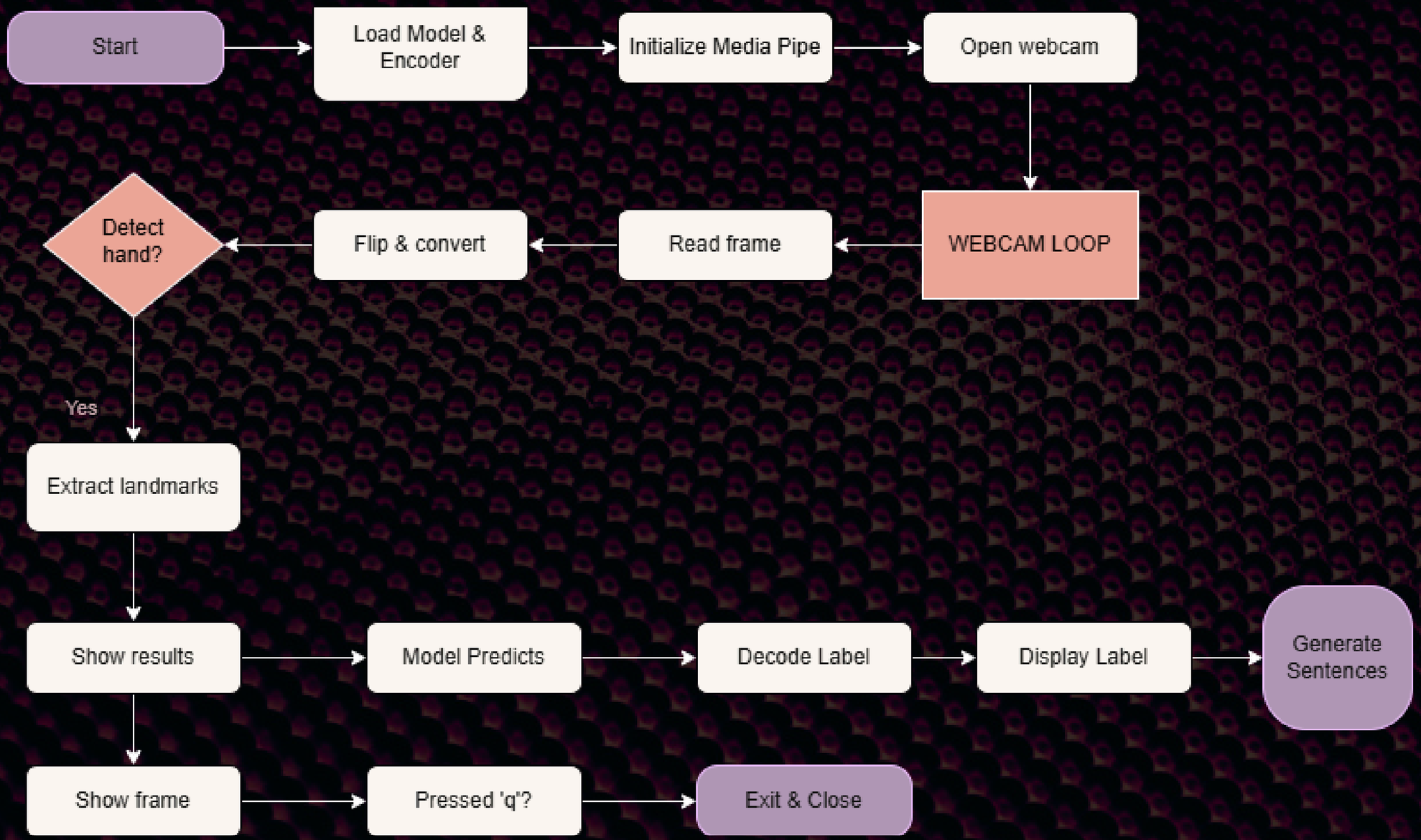
Addressing the Lack of Sign Language Awareness

We recognized a major gap — most people do not understand sign language, which creates unintentional exclusion. Our objective is to raise awareness by showing how simple tech solutions can make communication more inclusive and accessible for everyone.

Demonstrating the Social Impact of Student Innovation

By creating our own dataset and developing this system from the ground up, we want to prove that students can build socially impactful, real-world solutions. This project reflects how passion, purpose, and technology can come together to solve problems that truly matter.

FLOWCHART


Start
Load Model & Encoder
Initialize Media Pipe
Open webcam
WEBCAM LOOP
Read frame
Flip & convert
Detect hand?
Yes
Extract landmarks
Show results
Model Predicts
Decode Label
Display Label
Generate Sentences
Show frame
Pressed 'q'?
Exit & Close

DEMO

RESULTS

DATASET DESCRIPTION

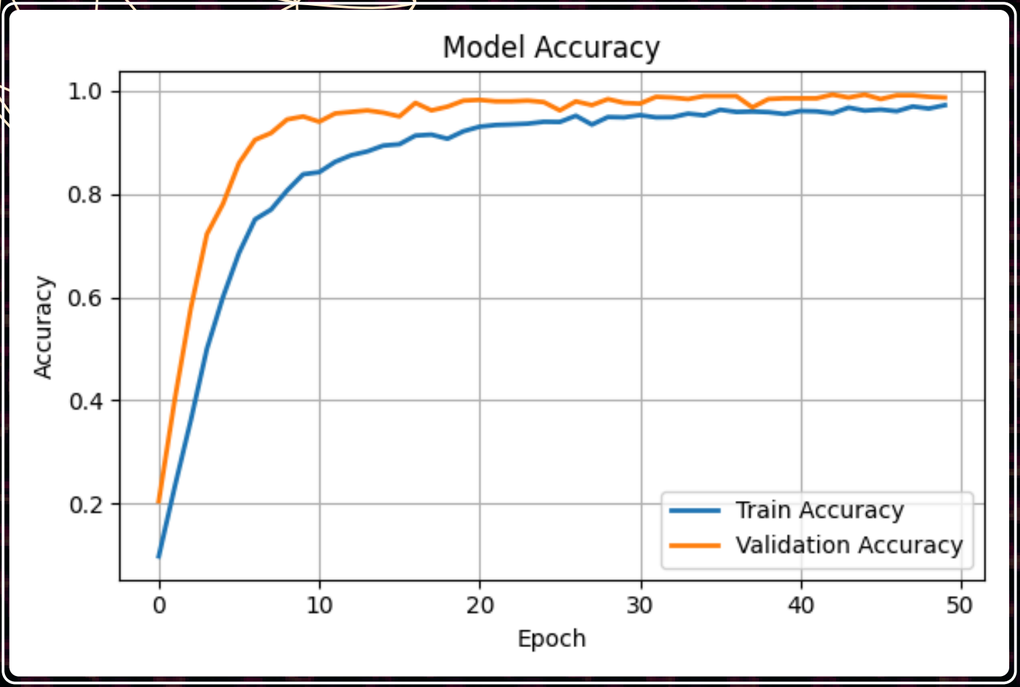
This dataset was self-recorded using a custom-built sign language capture system leveraging MediaPipe and OpenCV. It was designed to recognize American Sign Language (ASL) finger spelling gestures through landmark data collected from webcam video.

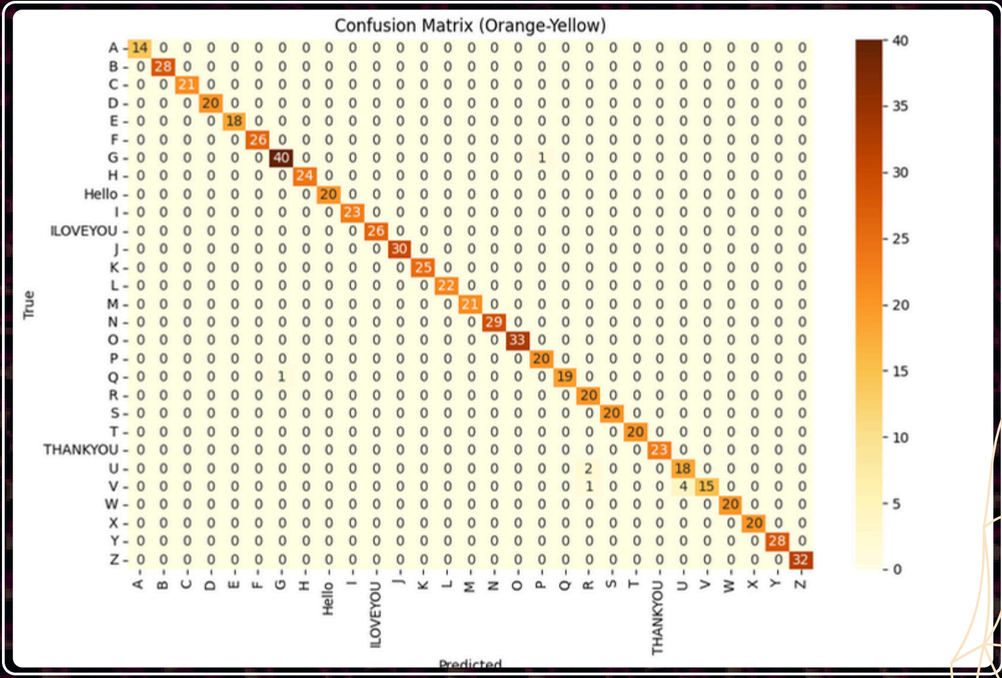
Recording Details:

- Method: Real-time webcam capture using OpenCV and MediaPipe Hands
- Approx: 3000+ words
- Data Collected: 21 hand landmarks per frame (only x and y coordinates — no z or 3D depth)
- Features: 42 total per sample (21 points × 2 values)
- Storage: All samples saved in a structured .csv file with 1 label column

```python
# --- Setup Mediapipe Hands ---
mp_hands = mp.solutions.hands
hands = mp_hands.Hands(static_image_mode=False, max_num_hands=1, min_detection_confidence=0.7)
mp_draw = mp.solutions.drawing_utils

# --- Create/Open Dataset File ---
if not os.path.exists('landmarks_dataset.csv'):
    # If file doesn't exist, create with header
    columns = []
    for i in range(21):
        columns += [f'x{i}', f'y{i}', f'z{i}']
    columns.append('label')
    df = pd.DataFrame(columns=columns)
    df.to_csv('landmarks_dataset.csv', index=False)
else:
    # If file exists, load it
    df = pd.read_csv('landmarks_dataset.csv')
# --- Open Webcam ---
cap = cv2.VideoCapture(1)
print("Starting webcam... Press 's' to save landmarks, 'q' to quit.")
current_label = input("Enter the label for this recording session (e.g., A, B, C, etc.): ").upper()

while cap.isOpened():
    success, frame = cap.read()
    if not success:
        print("Failed to grab frame.")
        break

    frame = cv2.flip(frame, 1)
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = hands.process(frame_rgb)
```

RESULTS

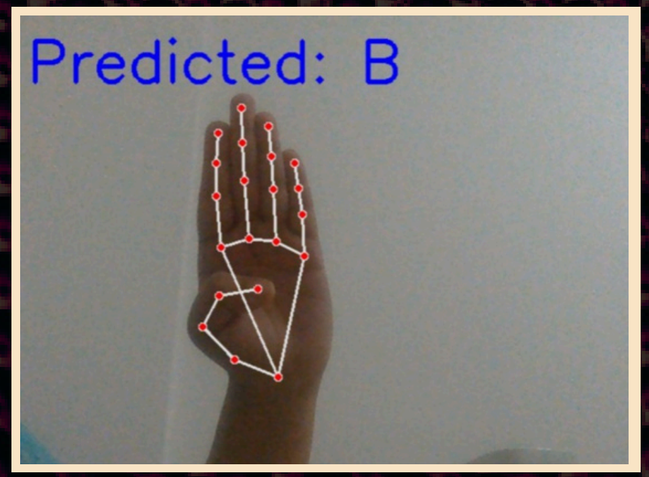




accuracy			0.99	710
macro avg	0.99	0.99	0.99	710
weighted avg	0.99	0.99	0.99	710

RESULT ANALYSIS

1. **Python** used as the core language with OpenCV for webcam-based video streaming
2. **MediaPipe Hands** detects up to 2 hands simultaneously, extracting 21 landmarks per hand in real time
3. A pre-trained Keras (**TensorFlow**) model predicts the ASL letter based on hand landmark positions
4. Each hand's 63 features (x, y, z of 21 points) are passed through the model for classification
5. Prediction stability is ensured using a frame queue + debounce logic (majority vote across 7 frames)
6. **LabelEncoder** (scikit-learn) is used to decode numeric predictions into letter labels
7. Predictions are displayed live on the video as Person 1 and Person 2, each with their own label
8. All built and tested using **Streamlit**, with standard webcam hardware



CONCLUSION

- Built a full-fledged ASL finger spelling system using real-time landmark detection and a self-trained deep learning model.
- Self-recorded dataset ensured clean labeling and gesture consistency, improving model reliability.
- Unlike traditional models, our system supports two-person detection, generating separate live sentences per signer.
- Added frame stability filtering to reduce noisy predictions, making real-time translation smoother and more accurate.
- The project stands out for its real-world usability, accessibility focus, and novel multi-signer support — all in an interactive app.

THANK YOU