Digital Image Processing Lab

Course Code: CSEL - 4104

Presentation on

Hand Gesture Recognition

A Digital Image Processing Approach

Presented by

Maria Akter Abin(B190305040)

Najnin Sultana Shirin(B190305031)

Supervised by

Dr.Md.Zulfiker Mahmud

Professor

Dept. of Computer Science & Engineering

Jagannath University, Dhaka

Table of Contents

01. Introduction

02. Challenges & Goal

03. Key Features

04. Implementation

05. Diagram and charts

06. Methodology

07. Image Acquisition

08. Preprocessing

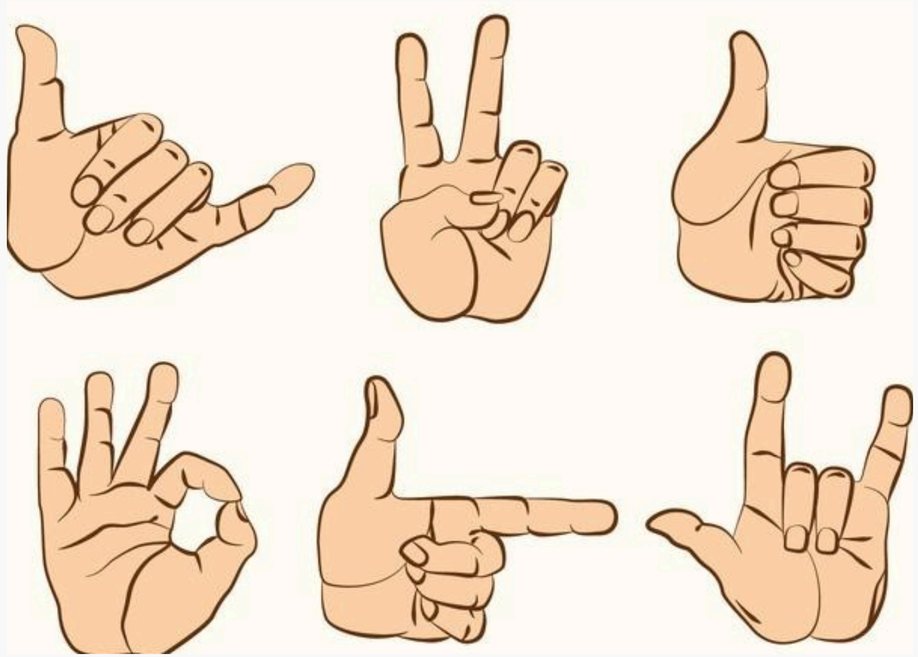
09. Classification

10. Gesture Output

11. Conclusion

Introduction

Hand Gesture Recognition **(HGR)** is a field of computer science that focuses on **recognizing** and **interpreting** human hand gestures. It involves using **Digital Image Processing** techniques to analyze images or video frames.



- A gesture is a **non-verbal** communication in which visible body communicates **particular** message.
- **Motion of body** that contains information.
- **Human Computer Interaction Gesture**

Challenges in Real-Time HGR

- Real-time Processing
- Robustness to Environmental Factors
- Accurately segmenting the hand from the background

Project Goals

Goal-1

Develop a robust and accurate real-time Hand Gesture Recognition (HGR) system

Goal-2

Implement an efficient HGR pipeline capable of processing image frames in real-time

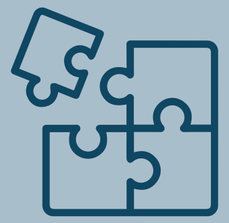
Goal-3

Create effective hand segmentation techniques to isolate hand regions, such

Key Features

Real-Time Gesture Recognition

- Detects and recognizes hand gestures from live input.



Multi-Gesture Support

- Supports a variety of hand gestures for different use cases.

Image processing & Machine Learning

- Utilizes OpenCV for image processing and detecting hand landmarks for gesture classification.

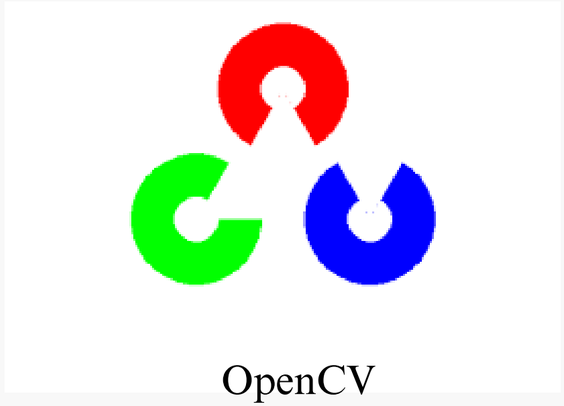
Customizable

- Easily extendable to add new gestures or improve the accuracy of the model.

Implementation

Software Tools used in HGR project making

Python 3.x



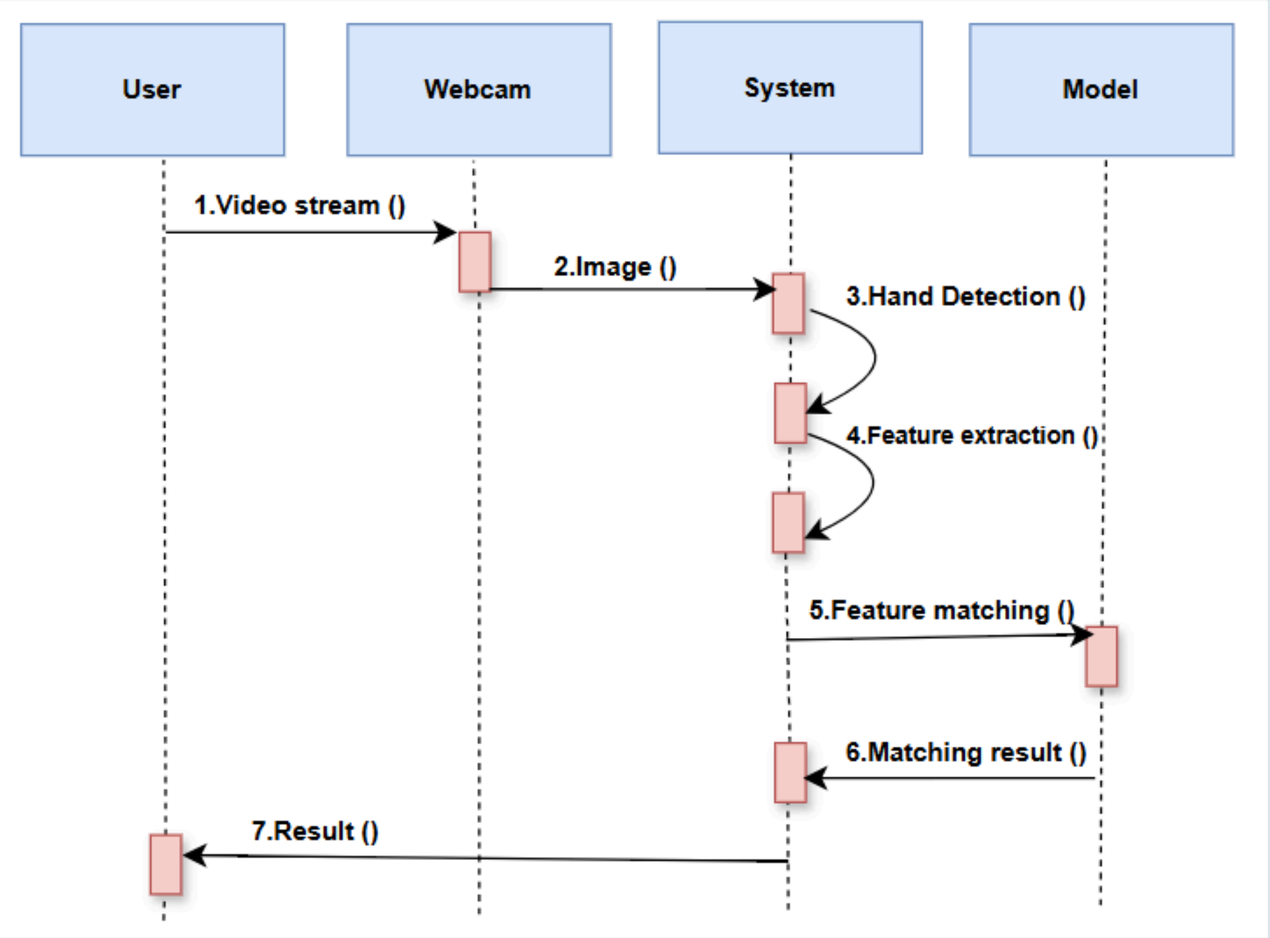
OpenCV

Numpy

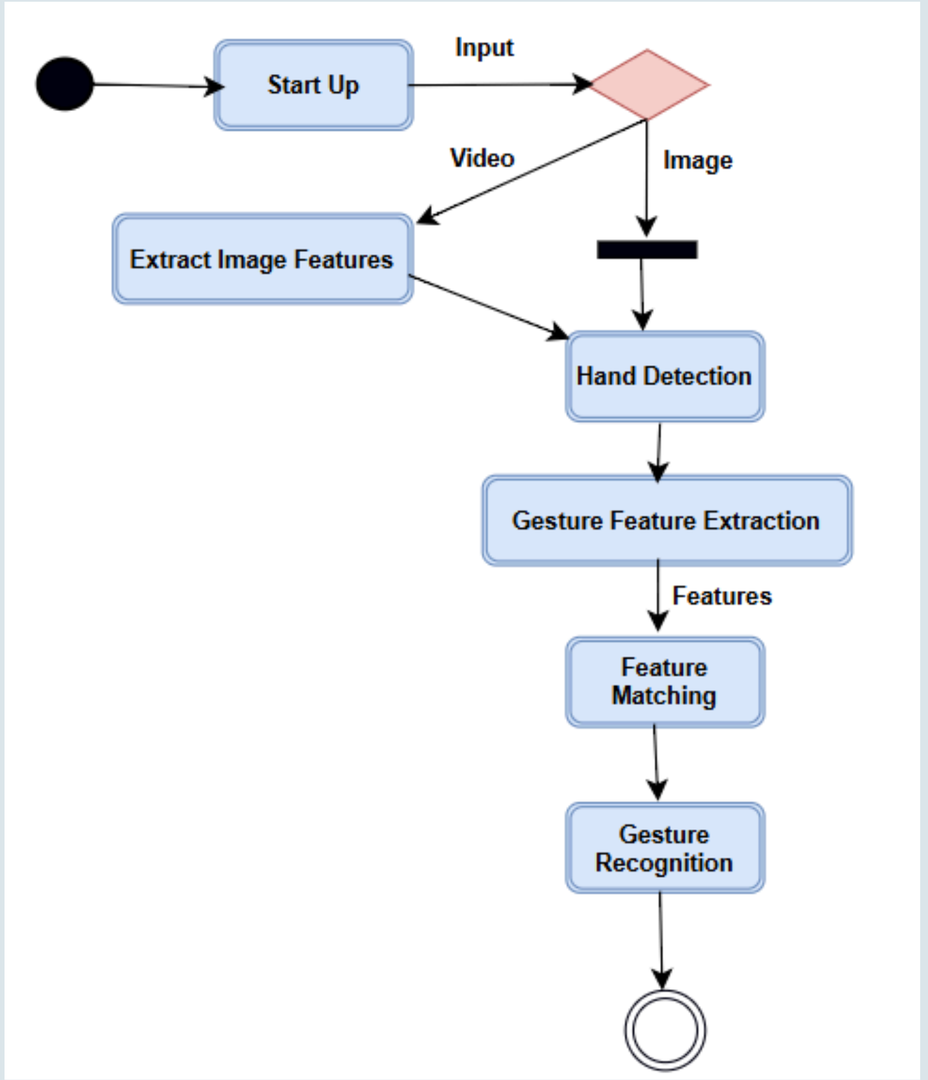
MediaPipe

TensorFlow

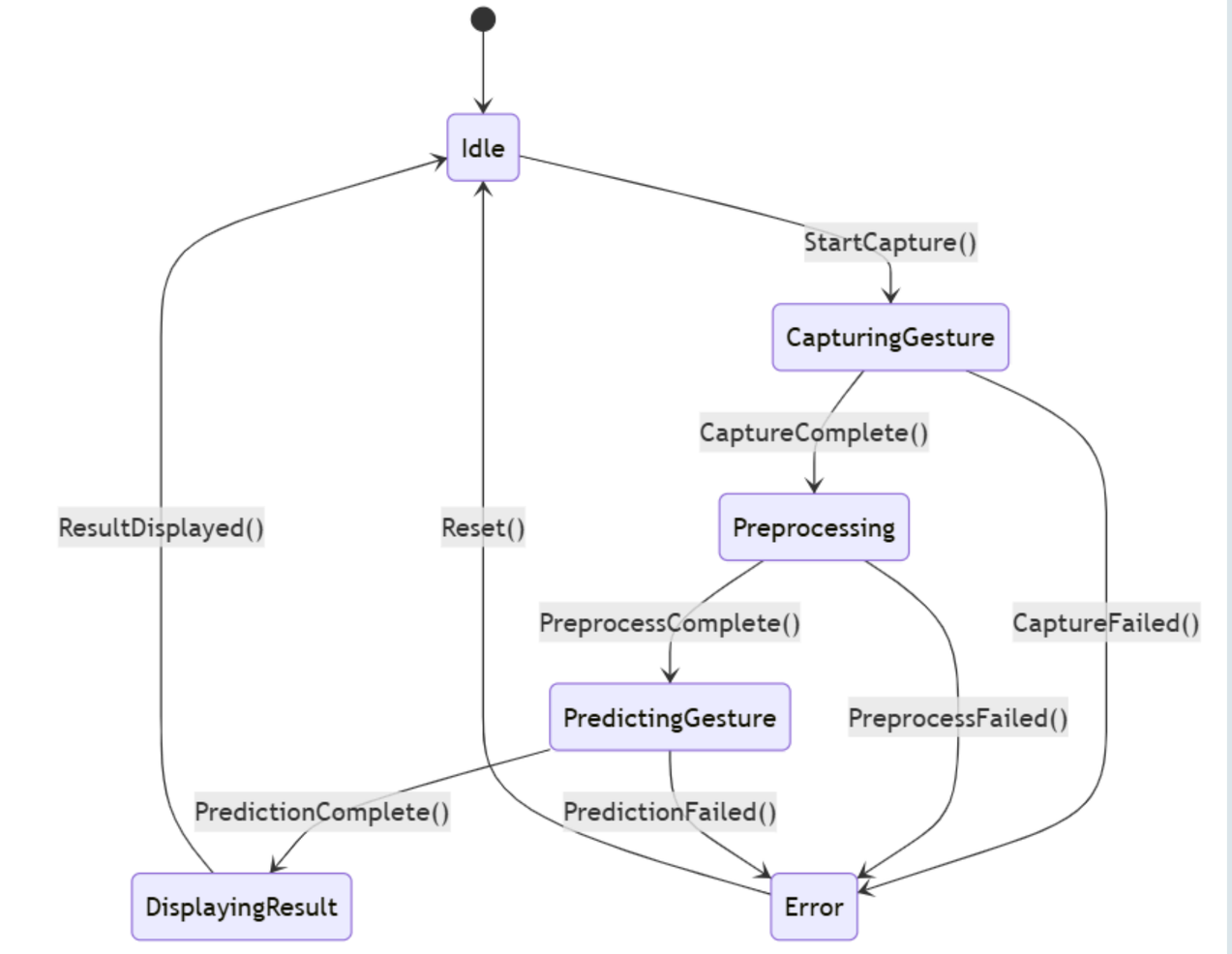
Sequence Diagram



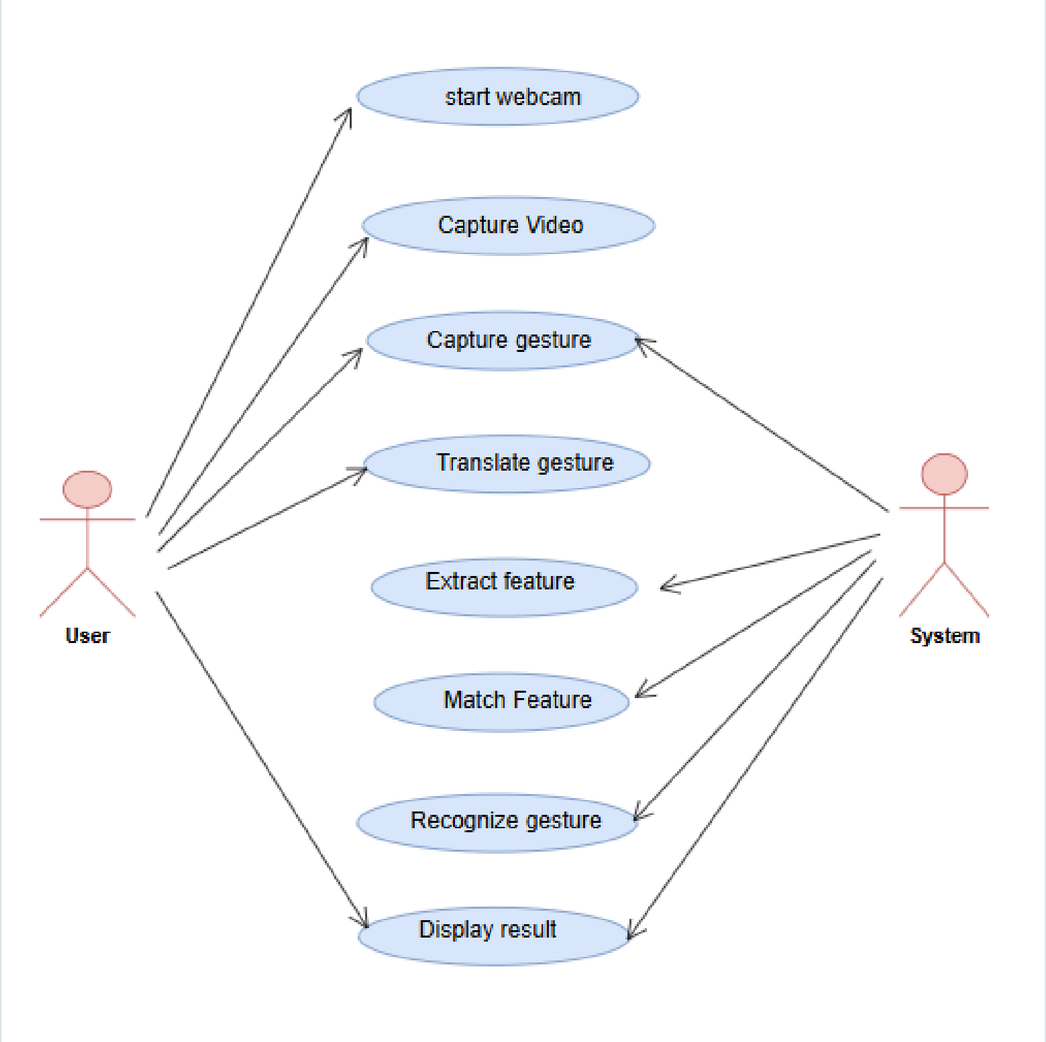
Flow Chart Diagram



Activity Diagram



Use Case Diagram



Project Flow

User

Webcam

Grabber

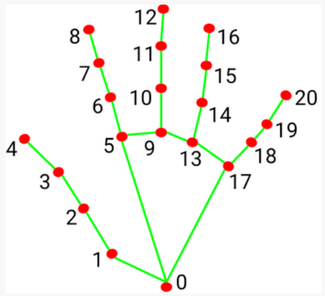
Detector

Recognized Gesture

Frame Grabber

Comparision

GRS



Finger count

Gesture Database

System Control

Hand Landmark and contour

Methodology

1.Image acquisition

2.Pre-processing

- **i)** Hand Detection
- **ii)** Cropping
- **iii)** Resizing and Centering
- **iv)** Gray-scale Conversion
- **v)** Binarization
- **vi)** Noise Removal
- **vii)** Blob Removal

3. Model Training

4. Real-Time Gesture Recognition

- **i)** Color Conversion
- **ii)** Hand Landmark Detection
- **iii)** Drawing Landmarks

5. Testing and Evaluation

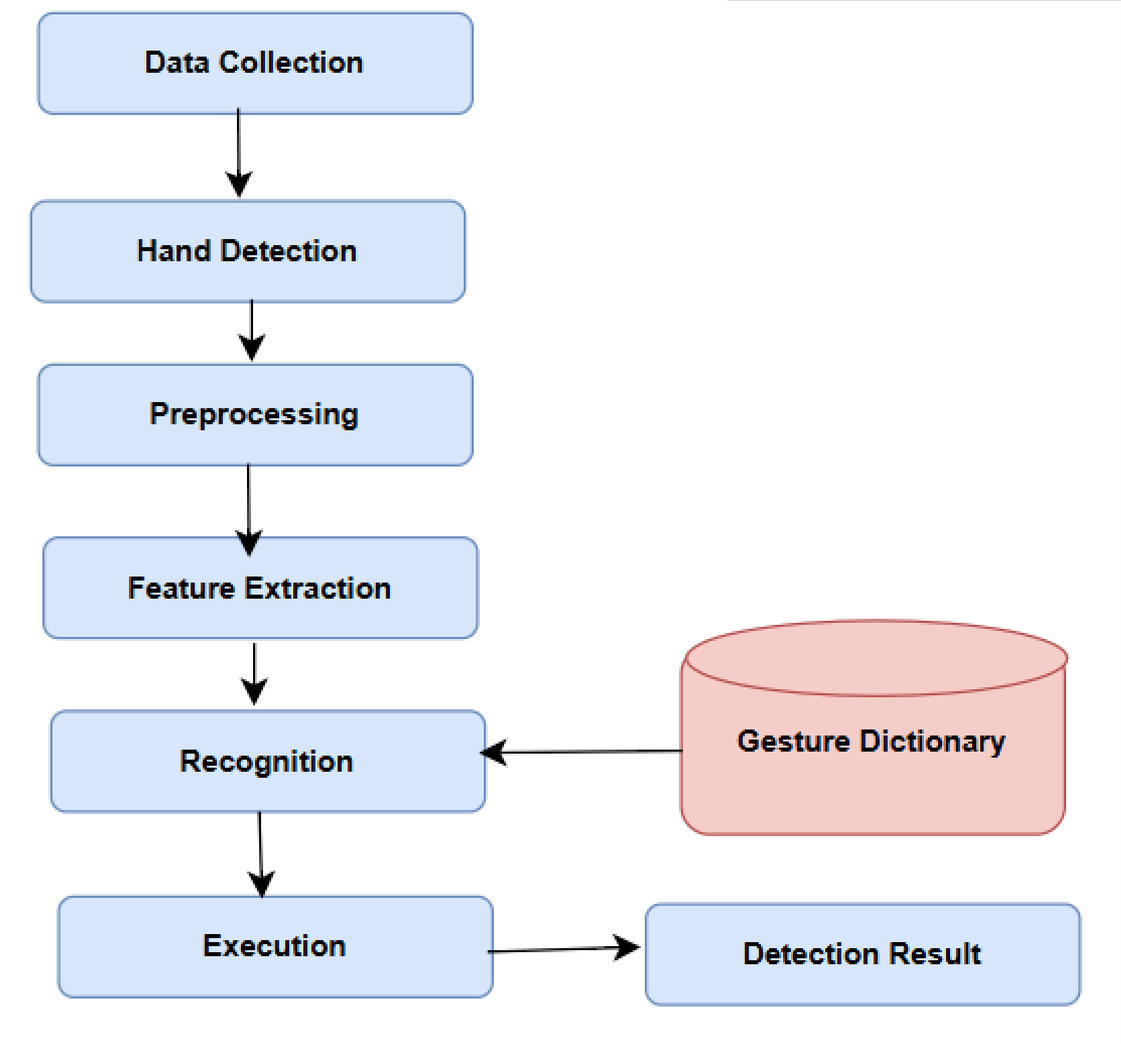


Image Acquisition

- The first step in hand gesture recognition is **Image Acquisition**. This involves capturing **images or video** frames of the hand performing various **gestures**.
- **Webcams:** Use a **webcam** to capture hand gestures with OpenCV and detect hands with the **HandDetector** .

Preprocessing

Before feeding images into the HGR system, We preprocessed data to improve their quality and facilitate subsequent analysis.

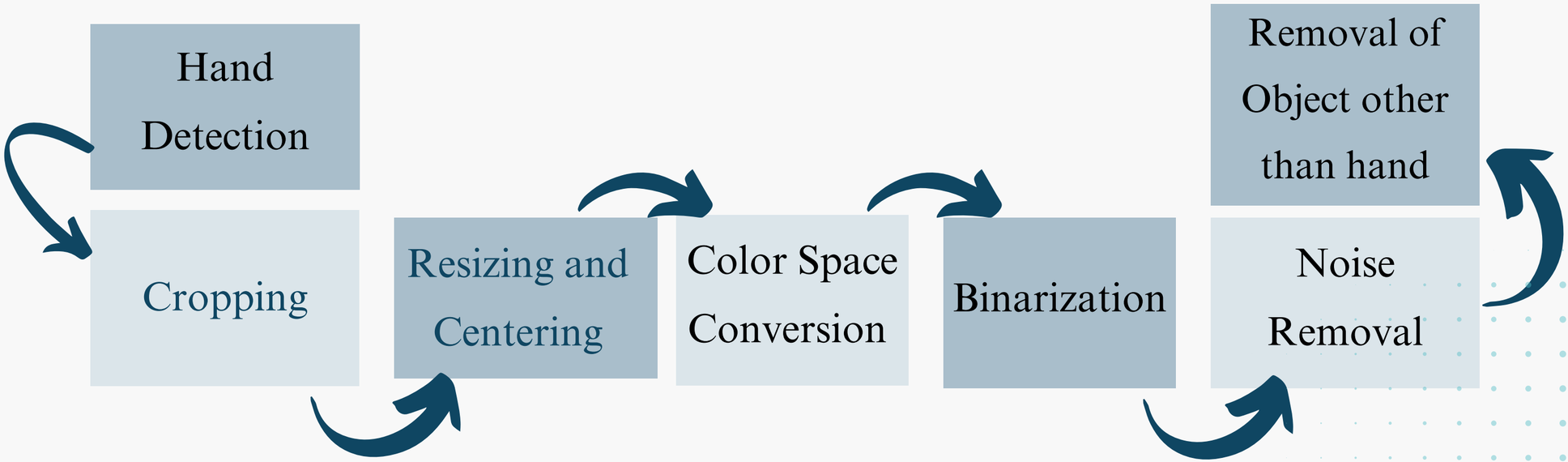


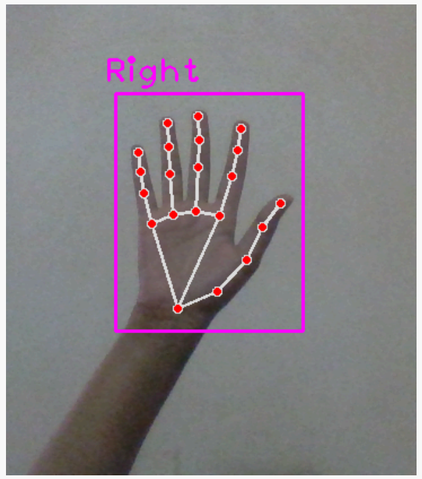
Image Preprocessing(Deatils)

1.Hand Detection: Detects and locates hands using **bounding boxes**.

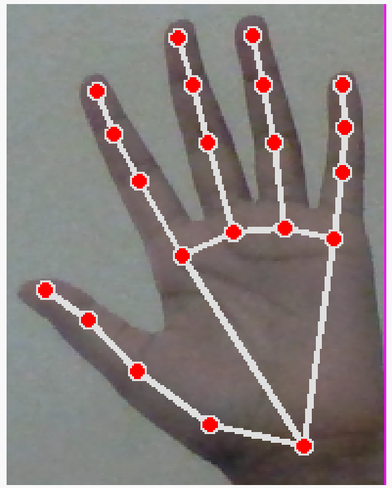
2.Image Cropping: Extracts only the **hand region** from each frame.

3.Resizing and Centering: Resizes the cropped image to fit within a fixed square size (**300 x 300** pixels), centering it within a **white background**.

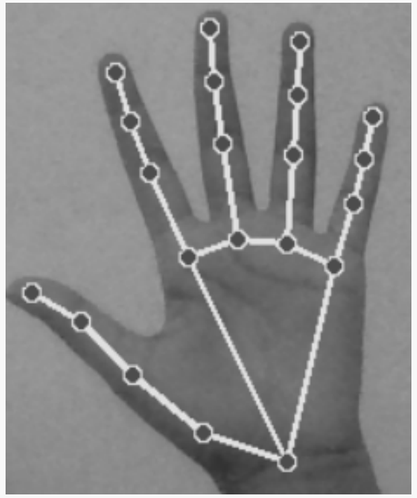
4.Gray-scale Conversion: Converts the centered image to **gray-scale.**



1



2

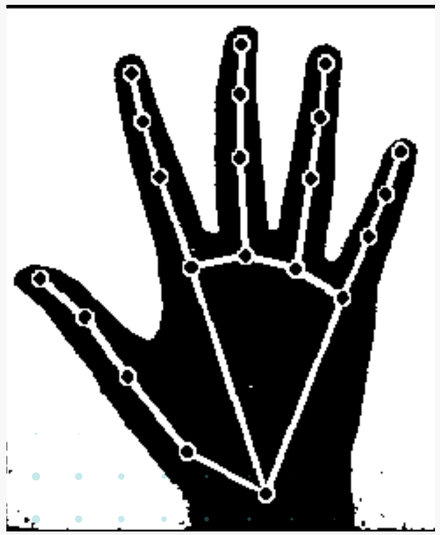


4

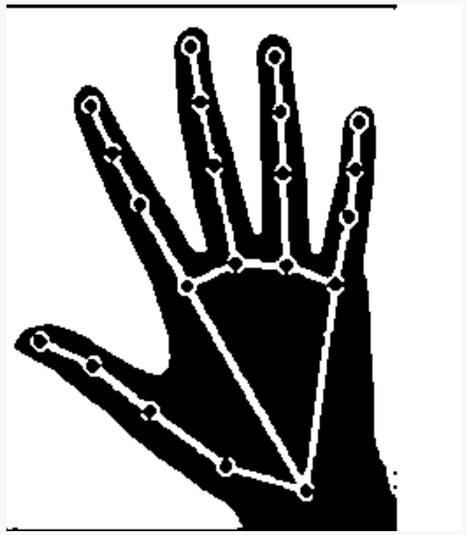
5.Binarization: Converts the **gray-scale** image to a **binary (black and white)** image using **thresholding**.

6.Noise Removal: Applies a **median filter** to reduce noise in the binary image.

7.Blob Removal: Removes **small components** (unwanted parts) from the image, keeping only **larger components** (like the hand).



5



6

7

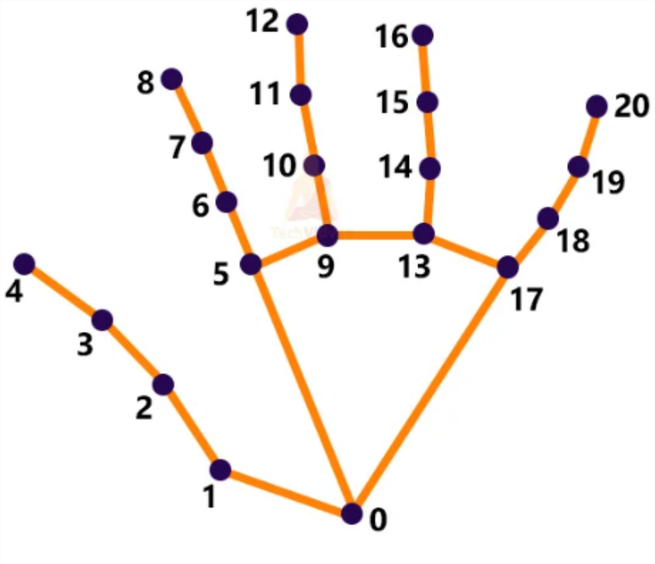
Gesture Recognition & Landmark Detection

Model Training:

- **Image Labeling:** Assigns **labels** to images based on gesture classes.

Real-Time Gesture Recognition:

- **Color Conversion:** Converts frames from **BGR to RGB** for **Mediapipe** compatibility.
- **Hand Landmark Detection:** Identifies **key points on the hand** (e.g., fingertips, knuckles).
- **Drawing Landmarks:** Visualizes **hand landmarks** and connections on each frame.



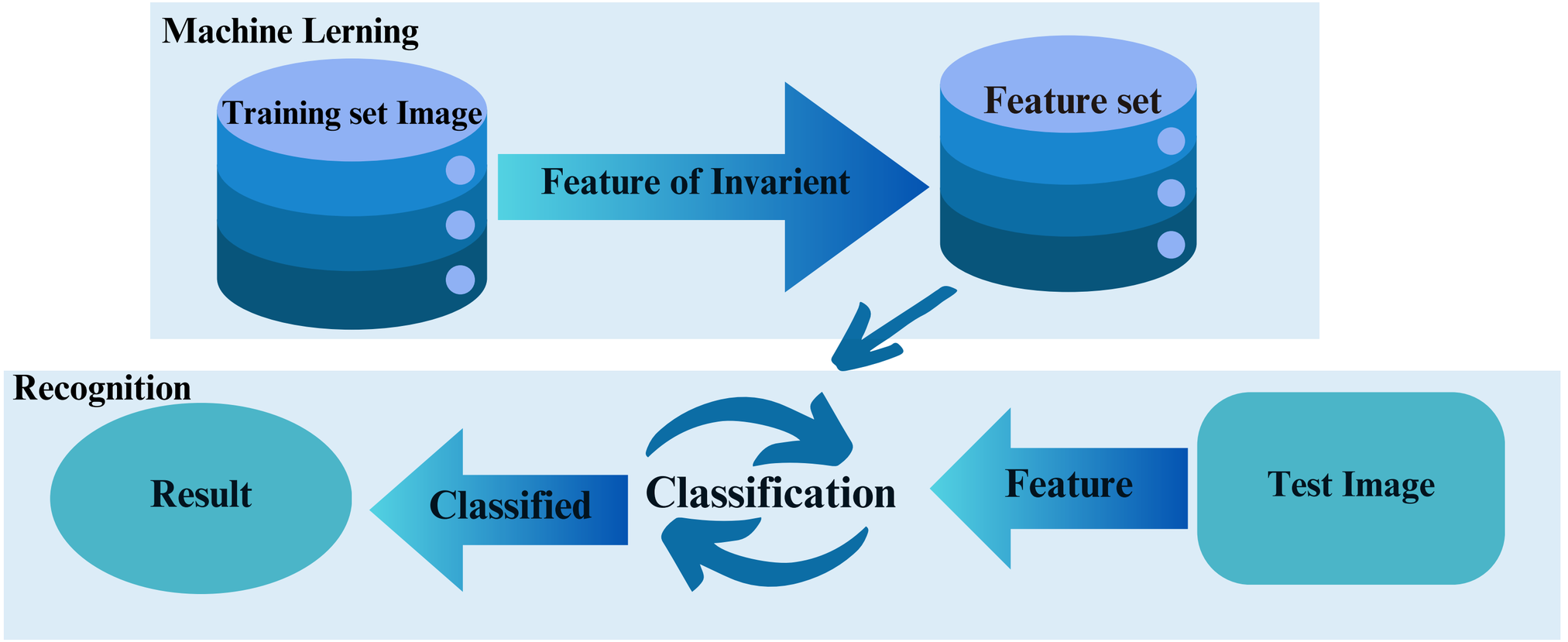
0.	WRIST	**11.**	MIDDLE_FINGER_DIP
1.	THUMB_CMC	**12.**	MIDDLE_FINGER_TIP
2.	THUMB_MCP	**13.**	RING_FINGER_MCP
3.	THUMB_IP	**14.**	RING_FINGER_PIP
4.	THUMB_TIP	**15.**	RING_FINGER_DIP
5.	INDEX_FINGER_MCP	**16.**	RING_FINGER_TIP
6.	INDEX_FINGER_PIP	**17.**	PINKY_MCP
7.	INDEX_FINGER_DIP	**18.**	PINKY_PIP
8.	INDEX_FINGER_TIP	**19.**	PINKY_DIP
9.	MIDDLE_FINGER_MCP	**20.**	PINKY_TIP
10.	MIDDLE_FINGER_PIP		

Classification

Image Classification has two steps:

1.Machine learning

2.Recognition

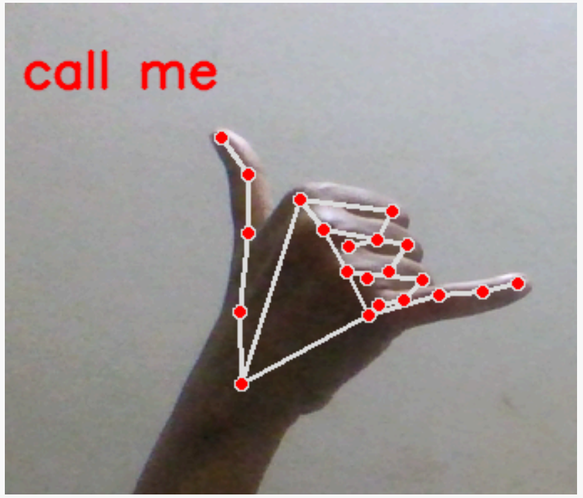


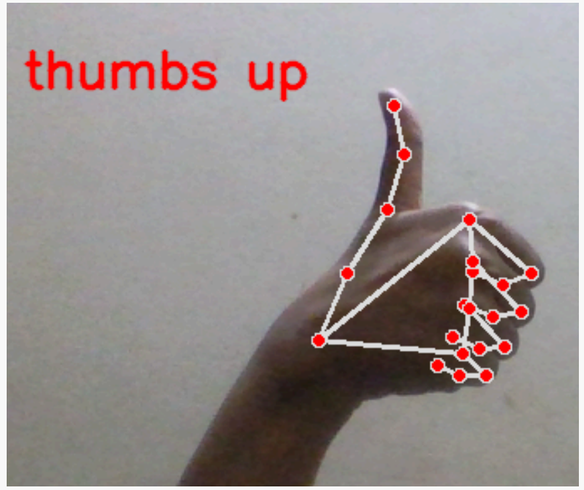
Gesture Output

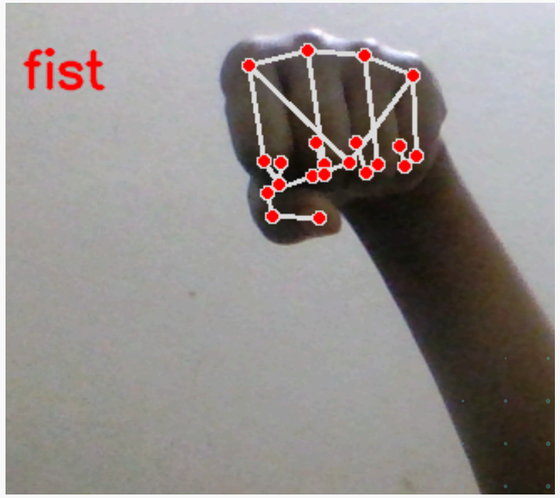
Testing and Evaluation:

- **Frame Processing:** Continuously **processes each frame** to detect hands and predict gestures.
- **Text Overlay:** **Displays predicted gesture label** on the video frame for real-time feedback.

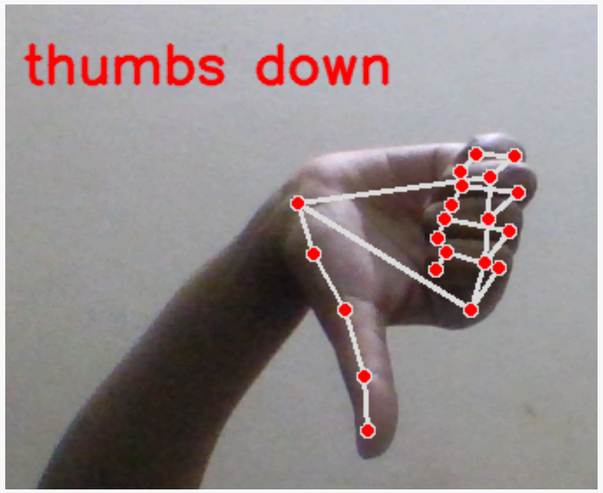
Our Predicted Gestures: We worked on some gestures like thumbs up, thumbs down, okay, peace, call me, stop etc.

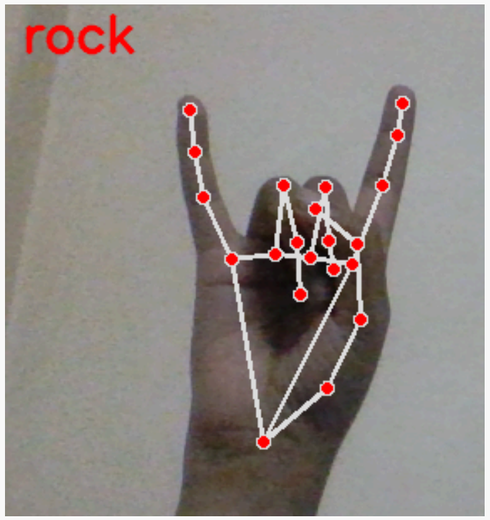


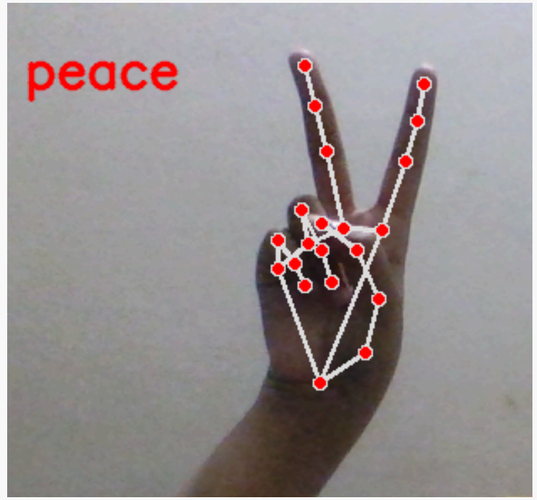


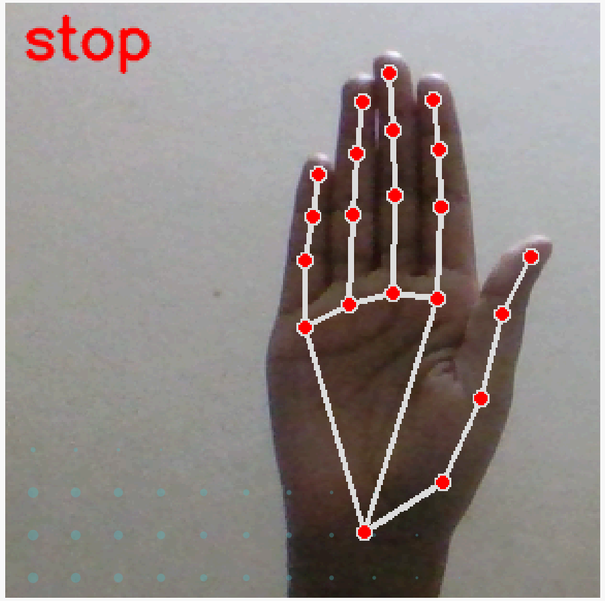


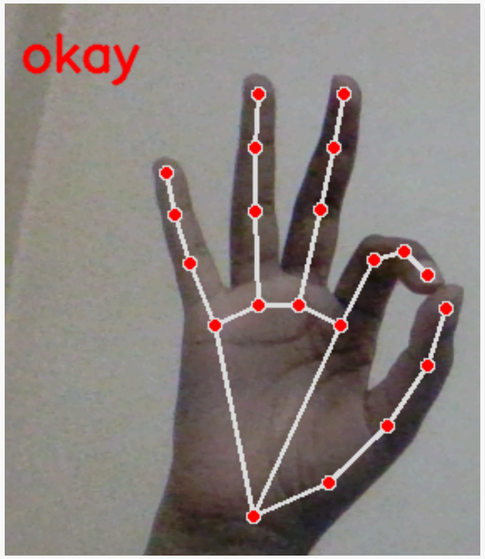
Gestures Output

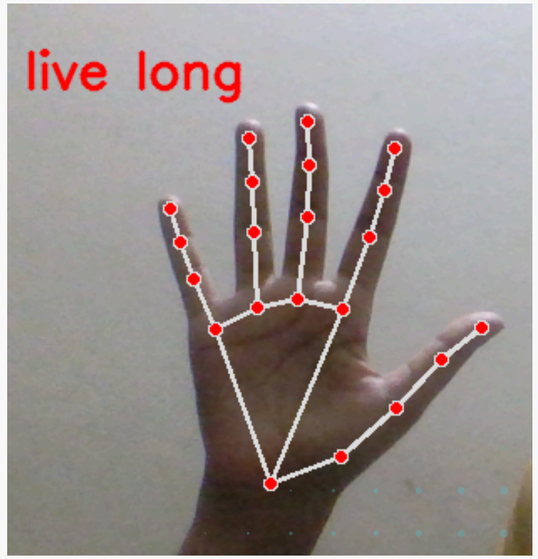












Conclusion

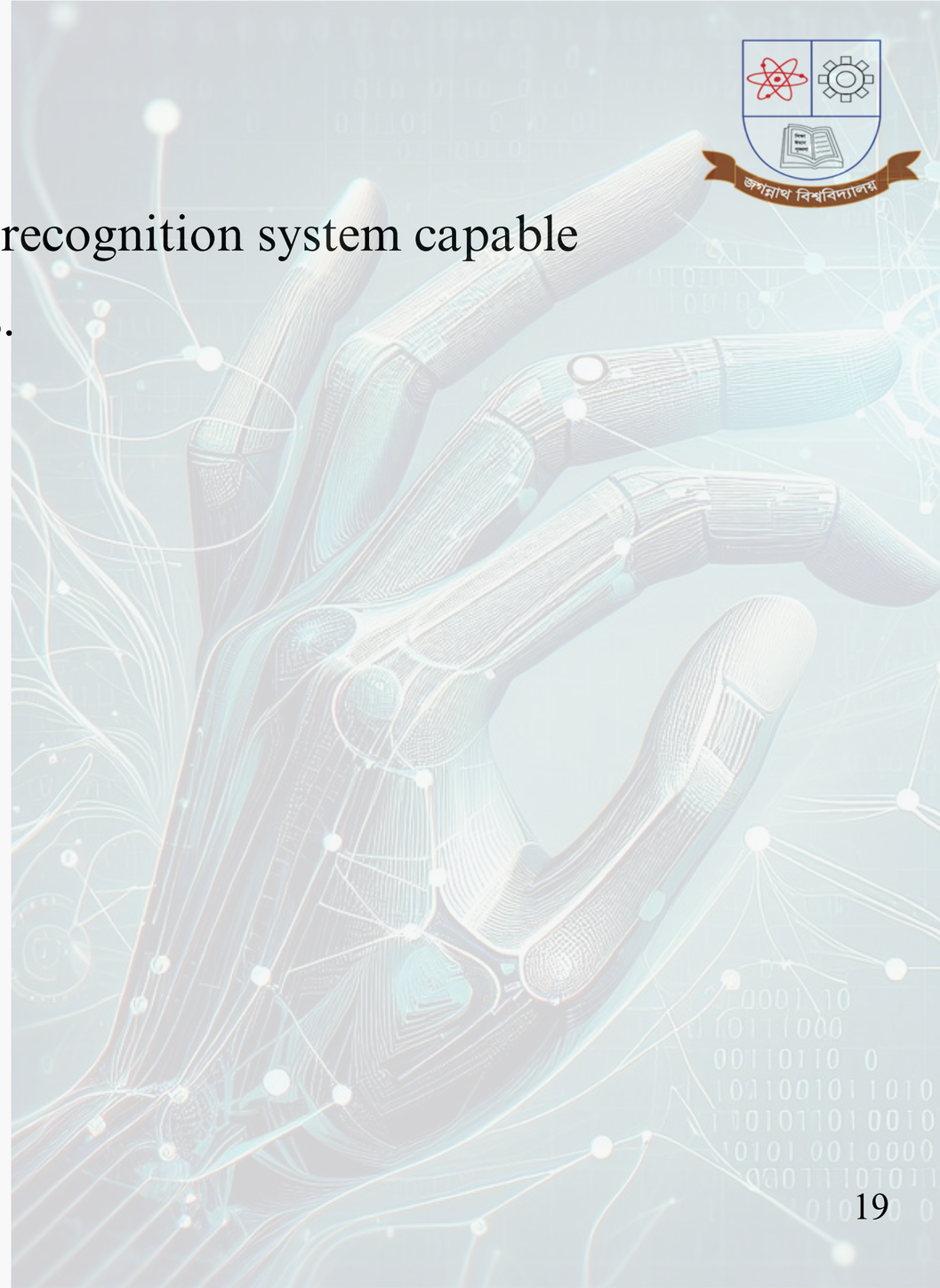
This project successfully developed a hand gesture recognition system capable of **accurately recognizing** a variety of hand gestures.

Future Improvements

- Expanding Gesture Library
- Improving Robustness
- User-Centric Design

Applications

- Robotics
- Artificial Intelligence
- Controlling computer through gesture



References

1.**IEEE Xplore: Real-Time Hand Gesture Recognition -** https://ieeexplore.ieee.org/document/8756576

2.ResearchGate: Hand Gesture Recognition- https://www.researchgate.net/publication/264117080_Real_Time_Hand_Gesture_Recognition_for_Computer_Interaction

3. https://techvidvan.com/tutorials/hand-gesture-recognition-tensorflow-opencv/

Thank you

Any question?