Md. Hasemi Rafsan Jani Shohan
191-33-849
hasemi33-849@diu.edu.bd

Gesture Movement based phase angle control part:

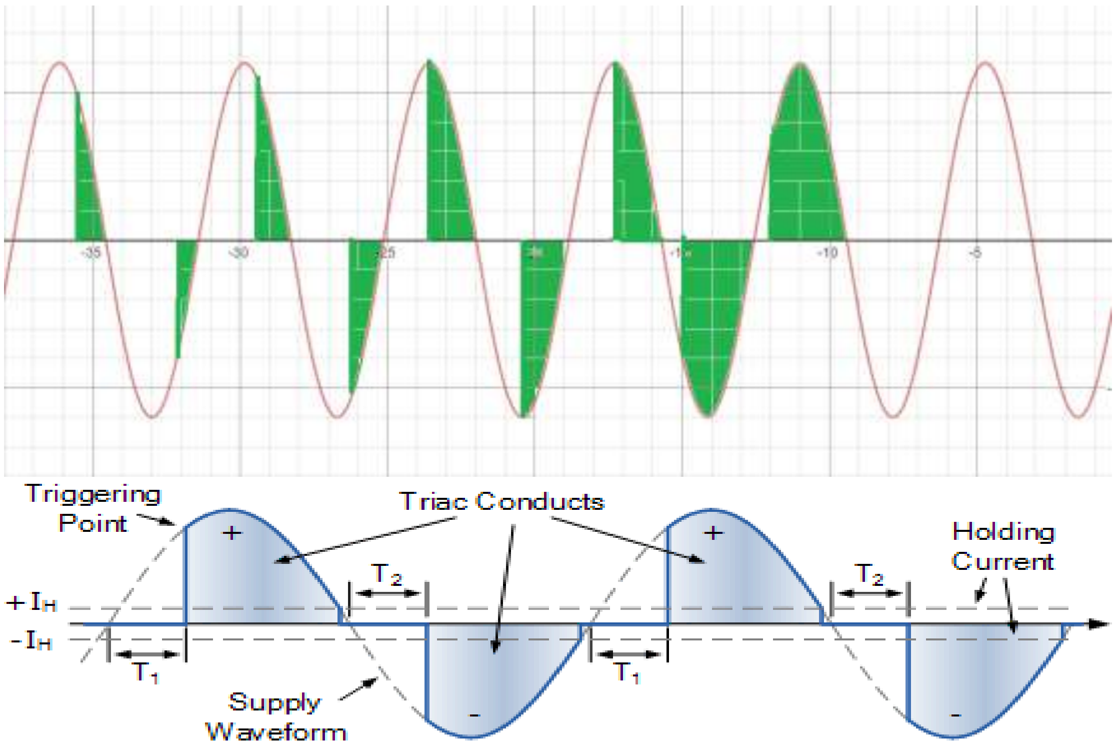


Fig 2.12 Gesture Control phase Voltage Triggering point view.[35][34]

Computers can comprehend human body language with the aid of gesture recognition. Instead of relying solely on text-based or graphical user interfaces, this promotes the development of a more powerful connection between humans and technology (GUIs). The movements of the human body are interpreted by the computer camera in this gesture recognition experiment. The computer then uses this information as input to manage apps. In order to adjust the voltage phase amplitude level, this project aims to provide an interface that dynamically captures human hand gestures.
The Python programming language now has support for massive, multi-dimensional arrays and matrices, as well as a large number of high-level mathematical functions to work on these arrays, thanks to the NumPy package. A free machine learning resource is Mediapipe. Google's open-source Mediapipe machine learning library offers encapsulation for python, js, and other languages and includes some solutions for face and gesture detection. A high-fidelity hand and finger tracking solution is MediaPipe Hands. It employs machine learning (ML) to extrapolate 21 crucial 3D hand

details from a single frame. It can be used to obtain the coordinates of the hand's important spots. **[28][29][30][31][32][33][34]**

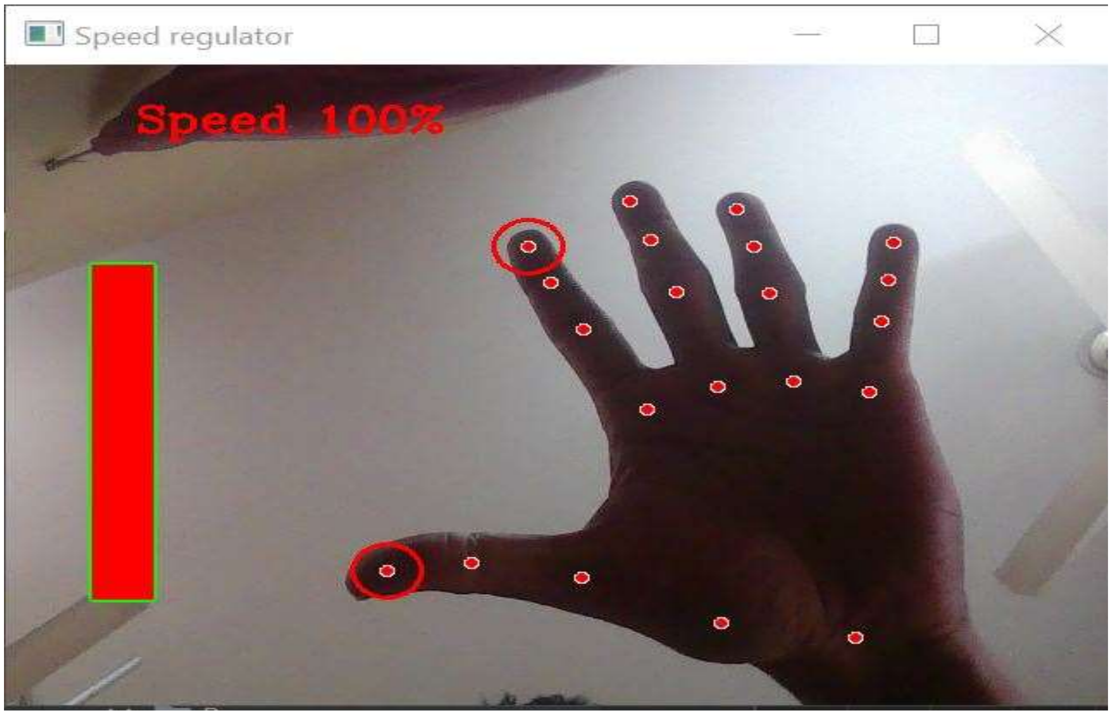


Fig 2.13 Gesture Control High Level gap between index and thumb finger .

I'll be utilizing this library's hands module to make two great projects in this article. The hands module generates a localisation of my hand based on 21 points. This means that if this project provide this module an image of showing hand, it will return a 21-point vector with the coordinates of 21 significant landmarks that are located on showging hand. **[28][29][30][31][32][33][34]**

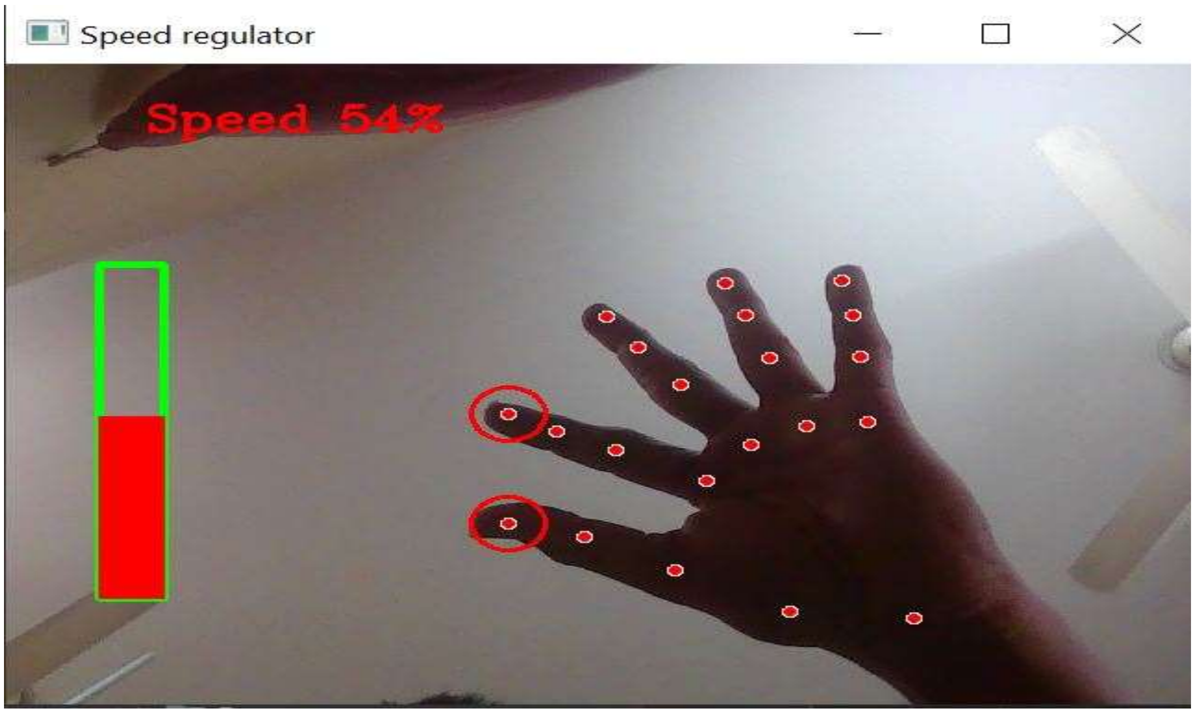


Fig 2.14 Gesture Control Midium Level gap between index and thumb finger.

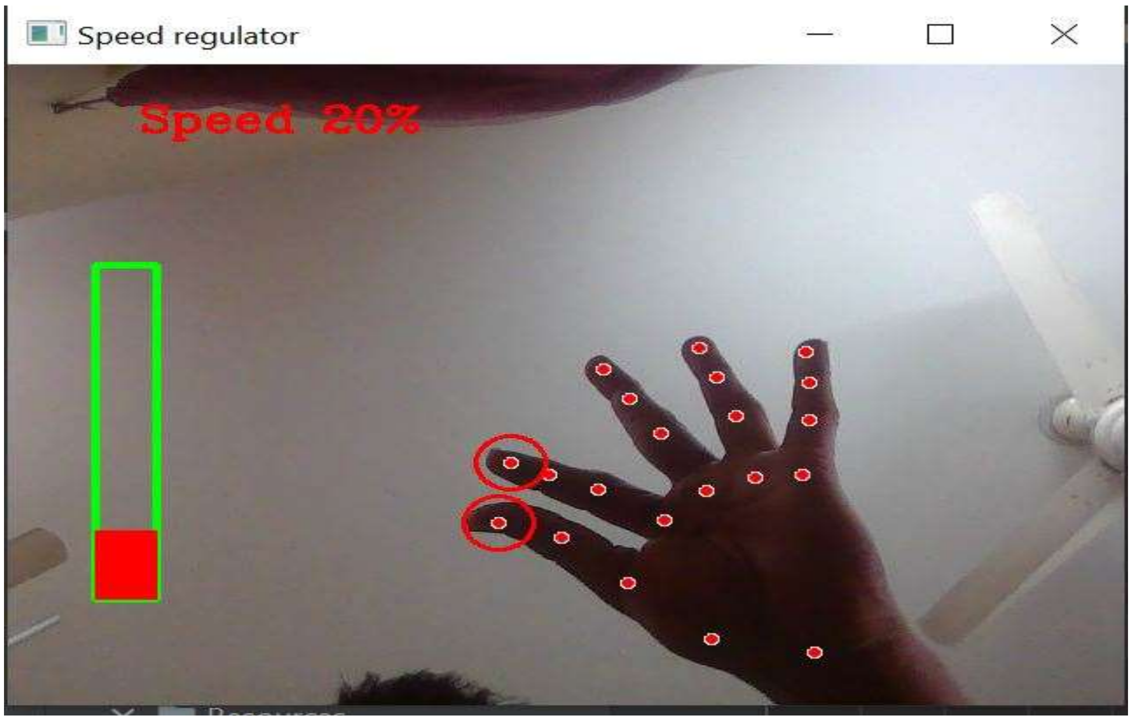


Fig 2.15 Gesture Control Low Level gap between index and thumb finger

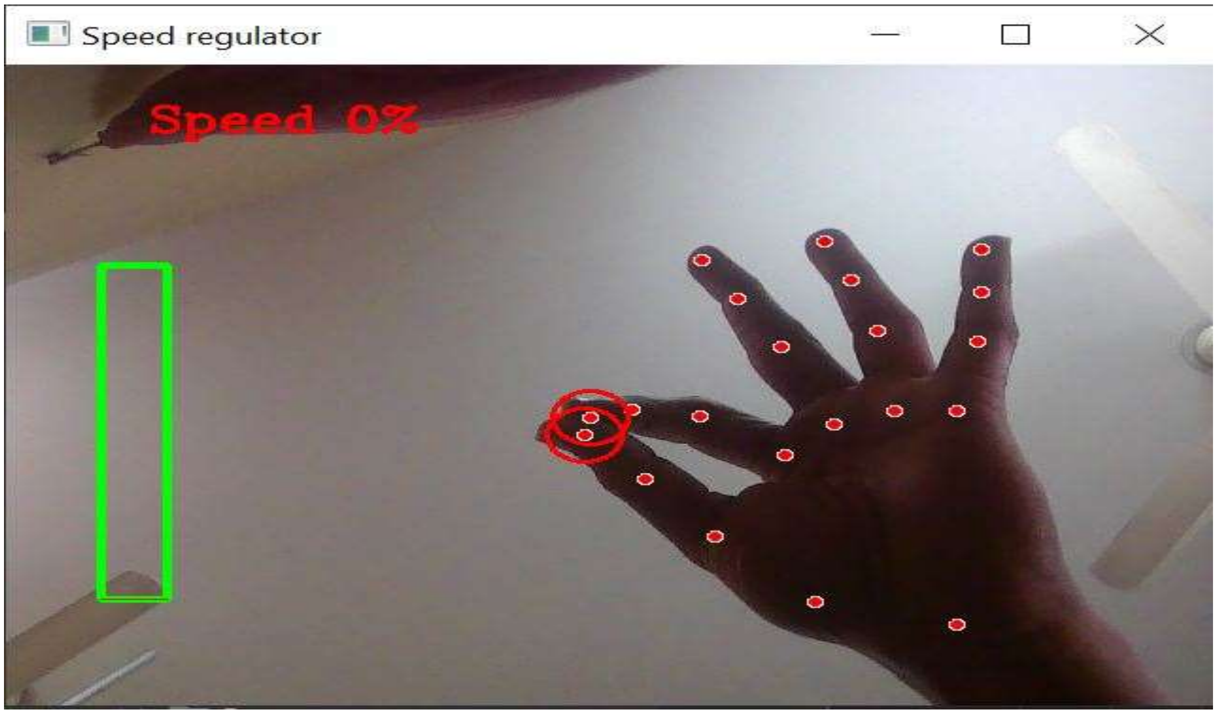


Fig 2.16 Gesture Control Zero Level gap between index and thumb finger .

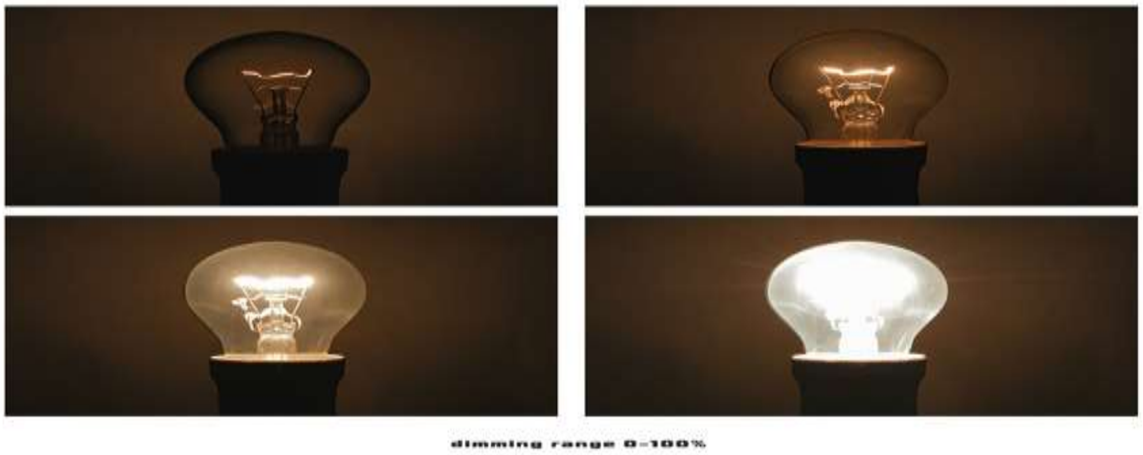


Fig 2.17 After Gesture Control Zero Level to High Level Voltage amplitude and brightness fluctuate Result.[35]

This project makes advantage of our device's camera. It recognizes my hand as having points on it so that it can measure the space between the tips of my thumb and index fingers. The Voltage amplitude of the gadget is directly proportional to the distance between points 4 and 8.
Track down hand landmarks Measure the distance between the tips of Showing thumb and index finger. Map the distance between the tips of Showingr thumb and index finger using a Voltage amplitude range. In my situation, the distance between the tips of my thumb and index finger fell between the range of 30 to 350, and the voltage level fell within the range of 0,100. **[28][29][30][31][32][33][34]**

Gesture movement based Voltage Amplitude Controller Making Steps:

Video Capture: Use OpenCV's VideoCapture class to access the webcam video feed in order to capture it. use Python tools like Mediapipe NumPy, Pandas, SciPy, and scikit-learn to help with this process.

Video pre-processing: To isolate the hand region in the frame, convert the video frame to grayscale and use thresholding.

Identify and extract the hand contour from the thresholded image using OpenCV's contour detection to recognize hand motions.

Recognize gestures: Using hand contour data, train a computer learning system, such as a convolutional neural network (CNN), to recognize particular hand motions.

Control voltage: Create code to relate changes in my system's phase voltage to recognized gestures.

Put the system in place: Create a comprehensive system that combines all of the parts and employs gesture control to change the phase voltage via the camera.

Identify gestures: Train a machine learning model, such as a Convolutional Neural Network (CNN), to recognize specific gestures based on the processed video frames.

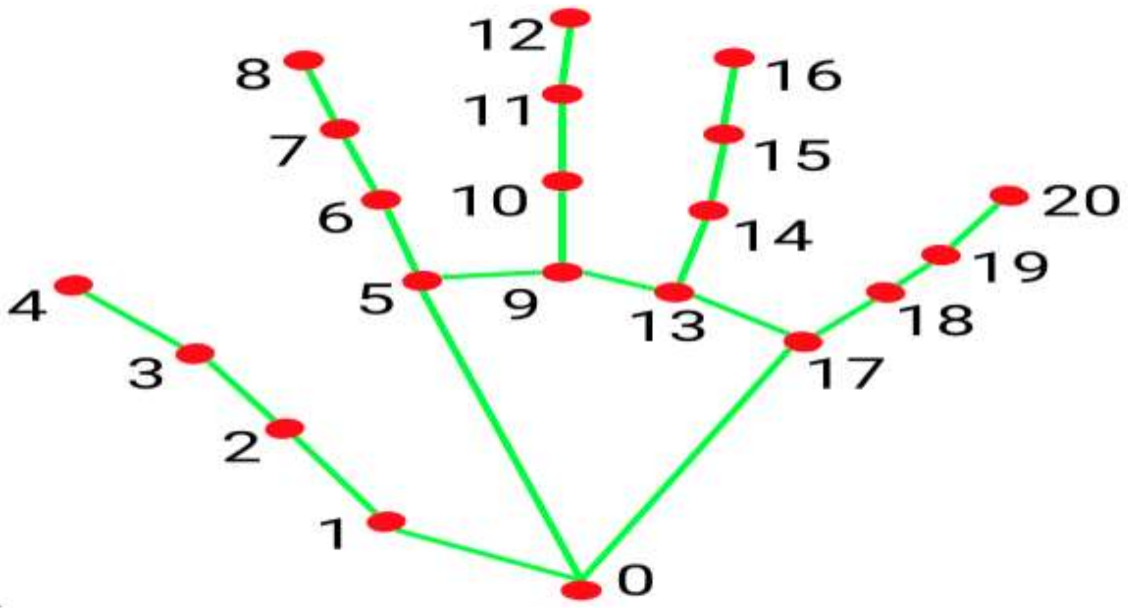
Map gestures to control: Write code that maps the recognized gestures to changes in the phase voltage of this project. Integrate system components: Integrate all of the components into a complete system that uses gesture control to adjust the phase voltage based on the webcam input.

An open-source platform for developing hand and body tracking applications is called MediaPipe. It includes a pre-trained TensorFlow-based hand identification model that may be used to identify hands in real-time video captured by a webcam or mobile device. Single-shot multi-box detection (SSD) is the foundation of the hand detection model, which has been real-time performance enhanced.

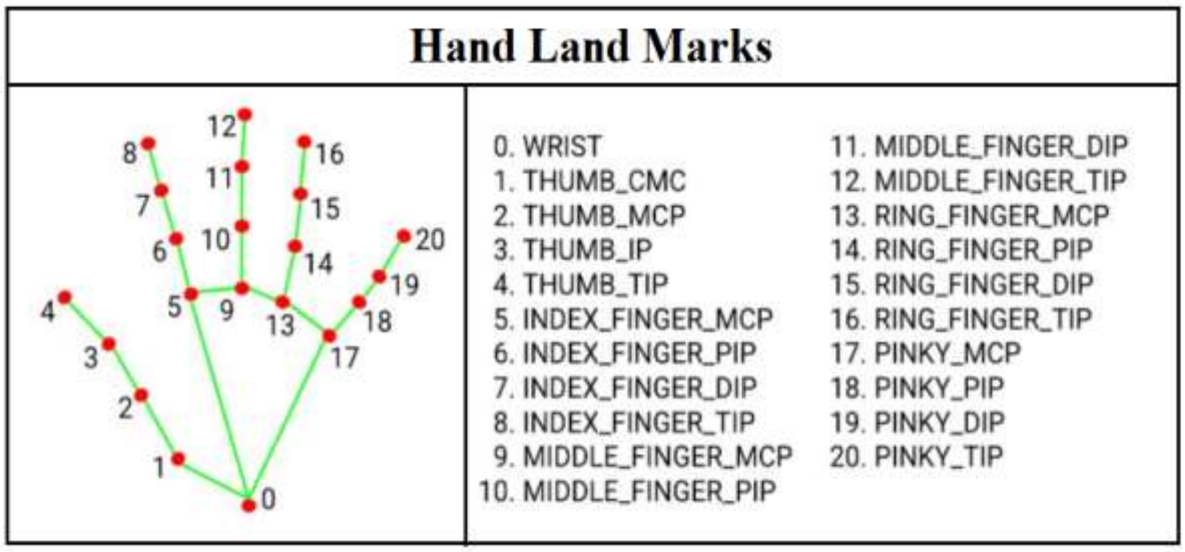
Install MediaPipe: To install MediaPipe and its dependencies, refer to the installation guidelines provided by operating system.

Run the hand detection pipeline: To run the hand detection pipeline on a video stream from a webcam or mobile camera, use MediaPipe's command line interface.

Imagine the outcomes: By processing the output in Ir own code or using MediaPipe's built-in visualizers, i can visualize the hand detection pipeline's output of hand bounding boxes and land marks in vdeo stream.



the **Fig 3.24 21 Hands Landmarks point** **[35]**



Hand Land Marks

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	

Fig 3.21 25 Hands Landmarks point Specification [35]

Customize the pipeline: I can alter the configuration file for the hand detection pipeline or provide Ir own code to handle the output. For instance, I may control a user interface or work with 3D objects using the hand detection information.
The shape and location of the hand can be described using hand landmarks, which are certain places on the hand like the fingertips, knuckles, and wrist. The hand bounding boxes produced by the hand detection model are used in MediaPipe to estimate hand landmarks.
A machine learning model that was trained on a sizable dataset of hand photographs is used to calculate the hand landmarks. The image plane's landmarks' 2D coordinates are output by the model. The landmarks can then be applied to tasks like 3D hand tracking, hand posture estimation, and hand gesture recognition.

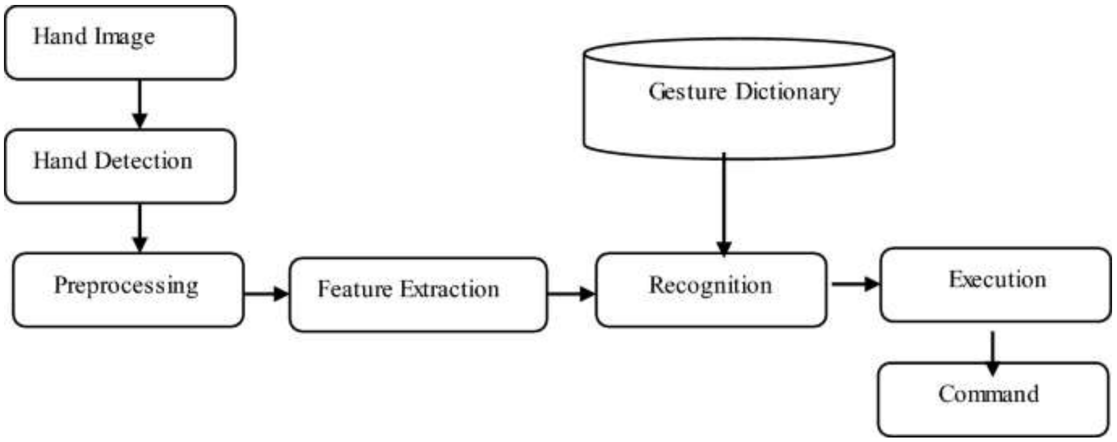


Fig 3.26 Gesture Control Steps 1 [35] [30] [32]

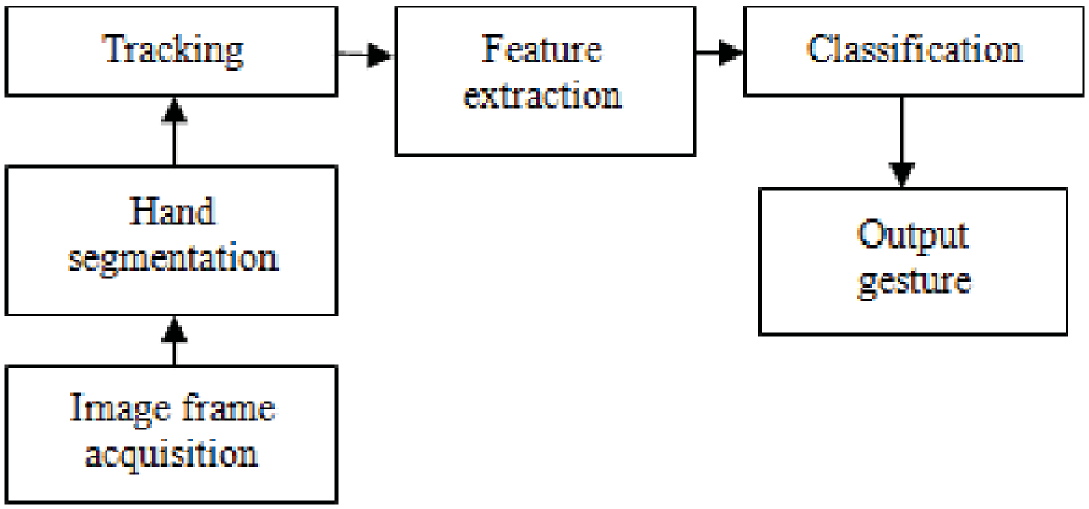


Fig 3.27 Gesture Control Steps 2 [35] [30] [32]

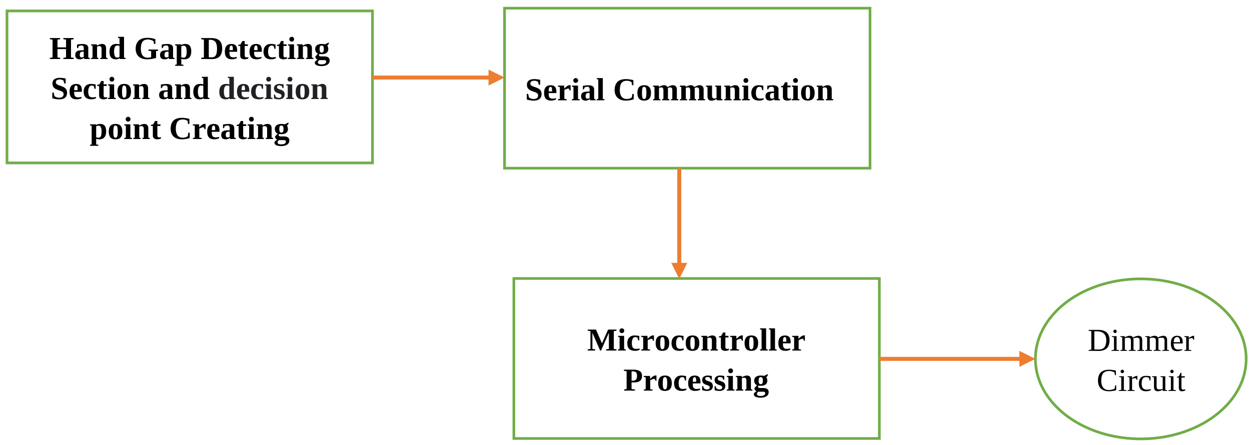

Hand Gap Detecting Section and decision point Creating
Serial Communication
Microcontroller Processing
Dimmer Circuit

Fig 3.28 Block steps of Gesture Control Development