Intensity Controller

The power to control a bulb's brightness is now at your finger tips. LITERALLY!!!
This video processing solution detects your fingertips and tracks the pinch/zoom motion of the fingertips and determines the corresponding light intensity. This solution is a 3-stage pipeline where, we first have the input video coming in. The next stage is to identify the hands in this video and track the necessary landmarks of the fingers. Finally, the distance between the desired landmarks is calculated and this value is extrapolated to intensity.

Figure 1: Intensity Controller Pipeline

Hand Tracker

The very first step after receiving the video (in our case, the video stream is fed in from the laptop's webcam using OpenCV with a minimal frame delay so as to not disrupt the video) is to first identify the hands and determine the important landmarks which are then tracked. Mediapipe from google offers an ML based tracker library which acts as a foundation for our hand tracking module. We determine the fingertips from the landmarks ((x, y) co-ordinates) and store the relevant information into a dictionary for further processing.

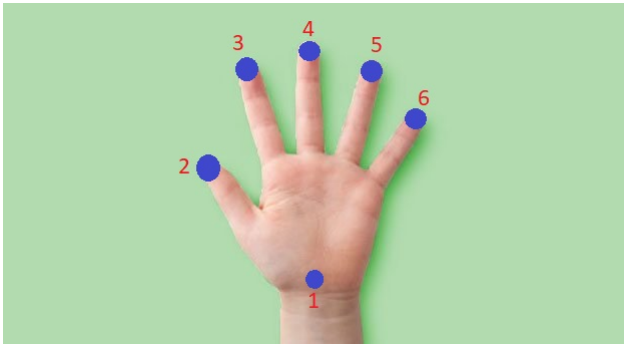

Figure 2: Landmarks of all the fingertips and wrist tip

Finger Detector

Now that we have the landmarks of the fingers in the frame, the next task would be to identify the number of fingers that are up. This feature enables the user to select the color of the bulb as well prior to adjusting its intensity. The user can show the desired number as in Fig (3) and choose the color option as shown in the table below. The user can use both hands as well to show the desired choice (say, the user wants to choose Green (Number 2), he/she can show one finger in each hand or two fingers in one hand. Both would be detected as 2).

Number	Color
0	Default: [127, 127, 127]
1	Red: [255, 0, 0]
2	Green: [0, 255, 0]
3	Blue: [0, 0, 255]

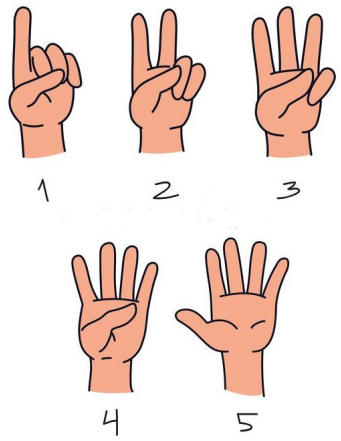

Figure 3: Count the number of fingers

The choice of choosing the desired color sequence is only a one time operation and after the choice is done, the color information in the MQTT payload will be static. The video processing module determines the the fingers information by checking the landmarks. The two landmarks used here are the fingertip and the point below it (which acts as the reference). If the fingertip landmark is above the reference landmark, the finger is considered to be up. If the user selects a wrong number (which is not in the table) or doesn't show any number for 10s, the default choice is selected.

Distance Calculator

Now that we have the desired landmarks and the choice of our color, the next step is to process these landmarks and determine the user operation (pinch or zoom). In order to maintain a fixed notion, we process information pertaining to the right hand and a frame with both or no hands implies the user is happy with the current intensity. The need to track the second (left) hand in frame to stop the operation is needed so as to avoid jitters when moving the right hand out of frame while stopping the operation. This intensity value is published via MQTT and the homeassistant, which has subscribed to this topic, regulates the bulb intensity.

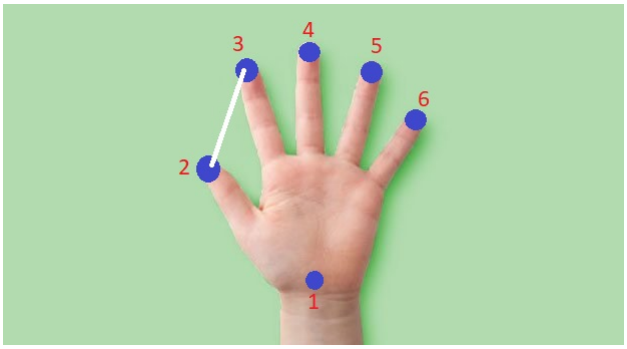

Figure 4: Distance measurement between thumb and index finger to determine user operation

The direction of displacement between landmarks 1 and 2 from Fig. (4) helps determine which hand is in frame ($+ve \Rightarrow RightHand, -ve \Rightarrow LeftHand$). Now that we have the landmarks and conditions to start and stop, we can now measure the distance. We isolate the thumb (landmark 2) as (x1, y1) and index finger (landmark 3) as (x2, y2) co-ordinates and calculate the euclidean distance (Eq.1) which returns the distance in the range 0 to 100. The euclidean distance is curbed to 100 due to the video frame resolution.Since the light intensity can go upto 255, we extrapolated this euclidean distance to span from 0 to 255 (Eq.2) and this value is then published via MQTT.

$$d(thumb, index) = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \tag{1}$$

$$d(extrapolate) = ((d(thumb, index) - 0.0)/100.0) * 255.0 \tag{2}$$

where 0.0 and 100.0 are the min and max values of the euclidean distance range and 255.0 is the max value of the new range(Intensity).