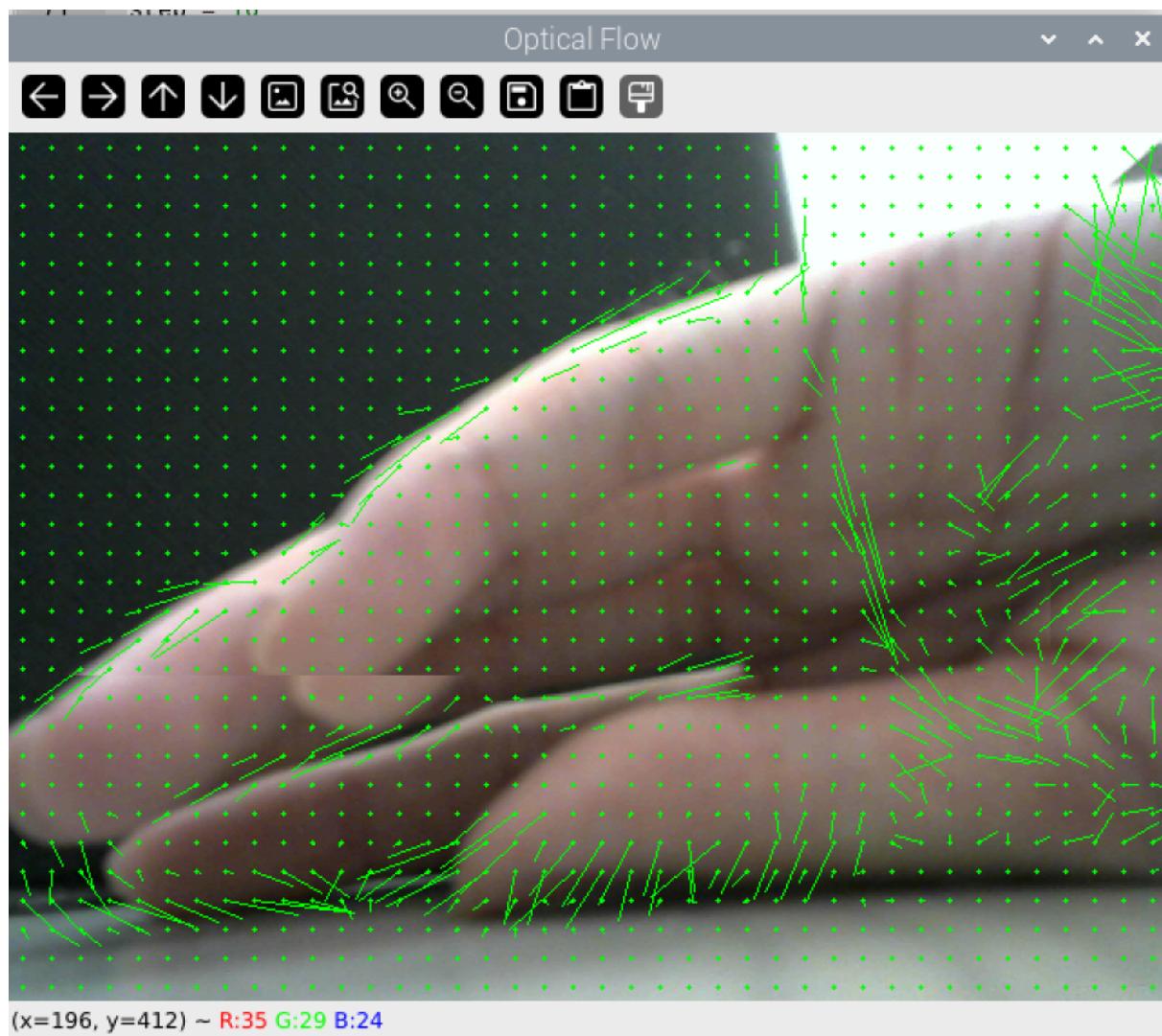


## INF2009 - Lab 4

[Modify the parameters (line 12/18) by looking into the OpenCV documentation and observe/note down the observations/conclusions.]

[Default]

```
11 # params for shi-tomasi corner detection
12 feature_params = dict( maxCorners = 100,
13                         qualityLevel = 0.3,
14                         minDistance = 7,
15                         blockSize = 7 )
16
17 # Parameters for lucas kanade optical flow
18 lk_params = dict( winSize = (15,15),
19                   maxLevel = 2,
20                   criteria = (cv2.TERM_CRITERIA_EPS | cv2.TERM_CRITERIA_COUNT, 10, 0.03))
21
```



[Modified]

feature\_params:

**maxCorners** = 100 to 300

This value results in more feature points being detected, which may improve tracking on complex features. However, there will be an increase in the computational load.

**qualityLevel** = 0.3 to 0.1

This value results in more corners being detected, inclusive of weaker ones. However, it may introduce noisy or unstable tracking.

**minDistance** = 7 to 15

This value results in lesser clutter by making the separation between detected corners further apart, which leads to better stability with the drawbacks of lesser features detected.

**lk\_params:**

**winSize** = (15,15) to (30,30)

This value results in the motion area that is being tracked to be larger, allowing for better tracking of objects that are moving quicker, with the downside of possible blurriness.

**maxLevel** = 2 to 5

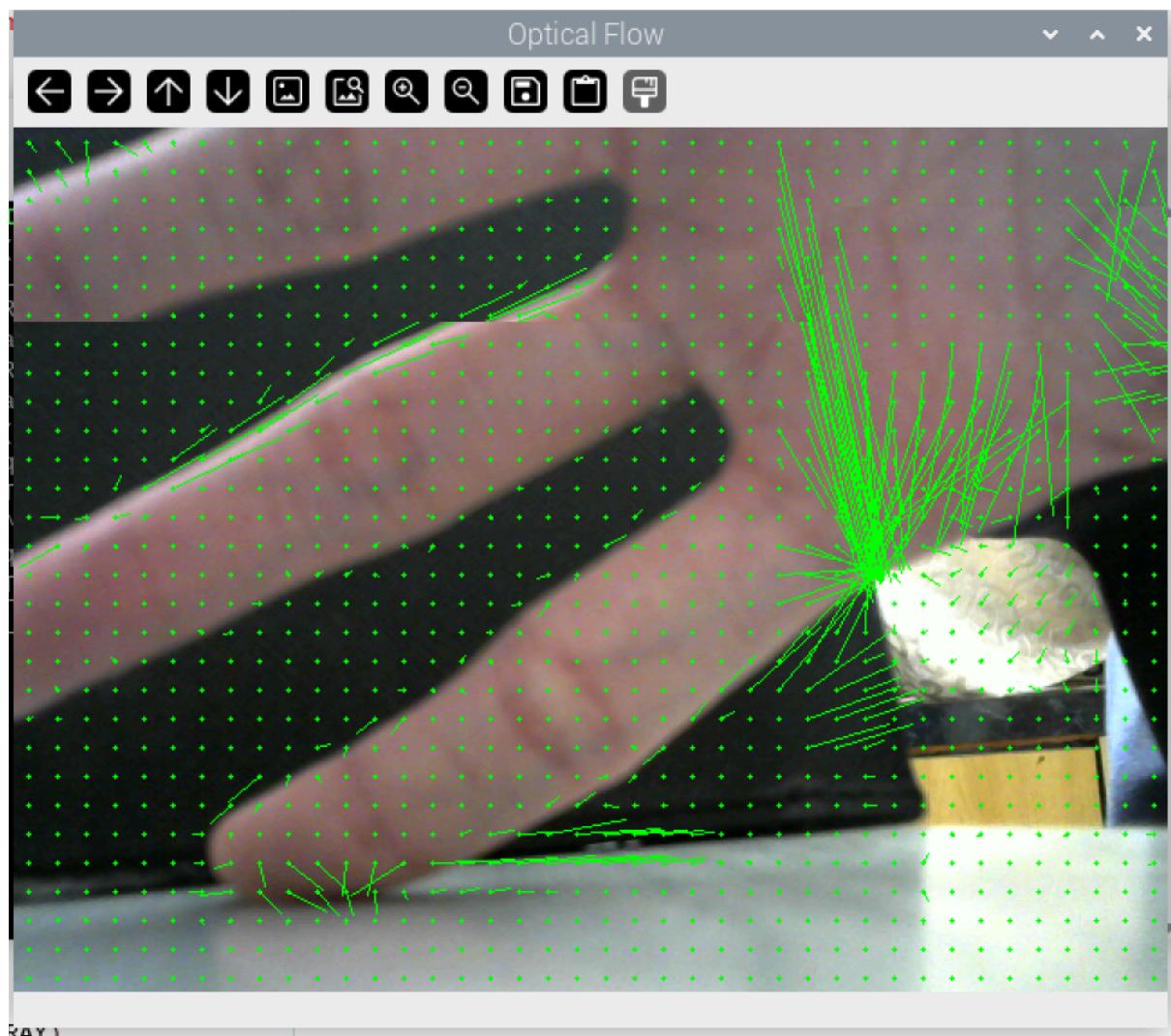
This value results in better tracking for faster motions with the drawback of slower computation.

**criteria** = (... , 10, 0.03) to (... , 20, 0.01)

This value results in a greater iteration count and accuracy, allowing for the tracking to be more stable, with a downside of having an increased computational load.

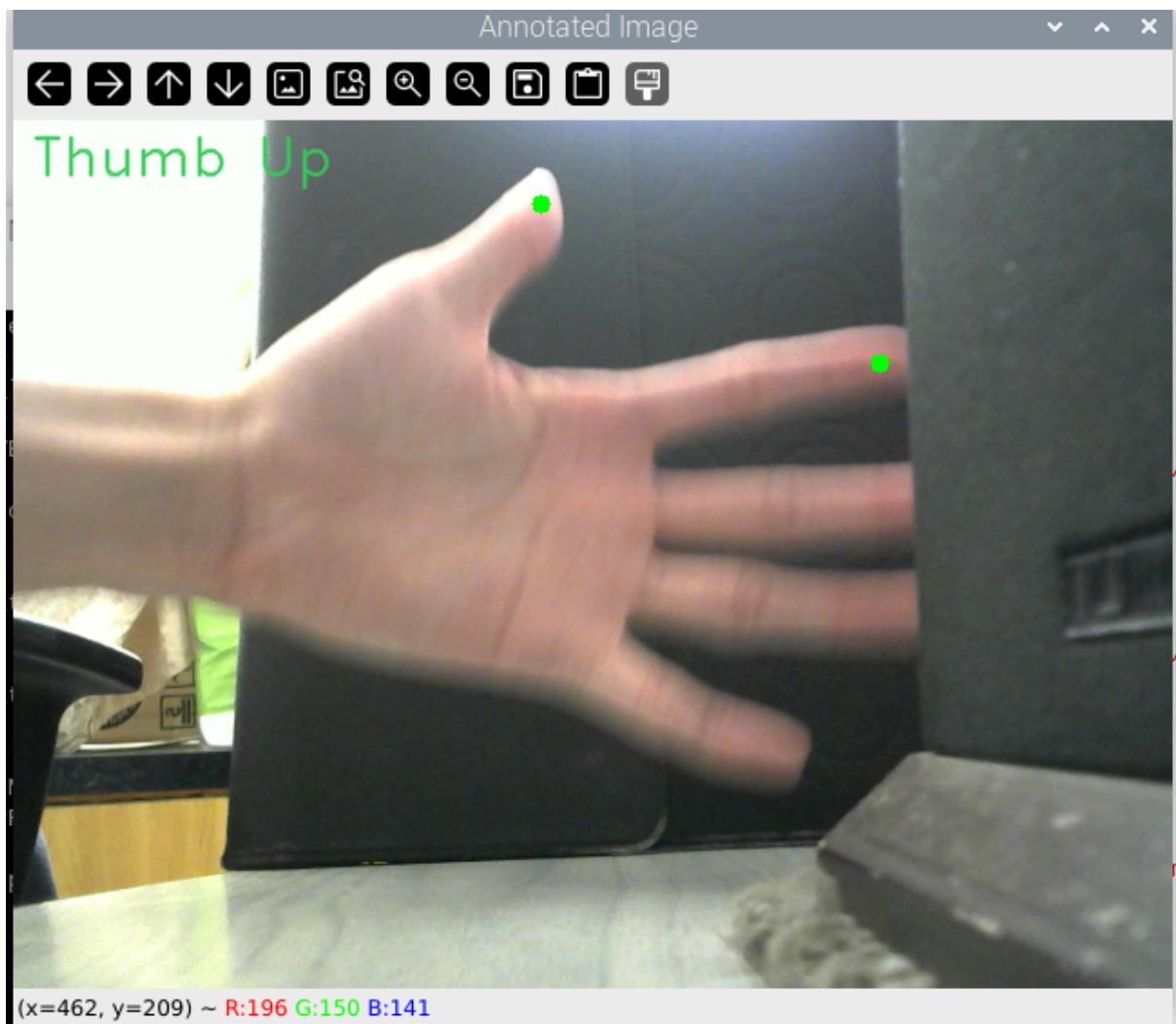
```
# params for ShiTomasi corner detection
feature_params = dict( maxCorners = 300,
                      qualityLevel = 0.1,
                      minDistance = 15,
                      blockSize = 7 )

# Parameters for lucas kanade optical flow
lk_params = dict( winSize = (30,30),
                  maxLevel = 5,
                  criteria = (cv2.TERM_CRITERIA_EPS | cv2.TERM_CRITERIA_COUNT, 20, 0.01))
```

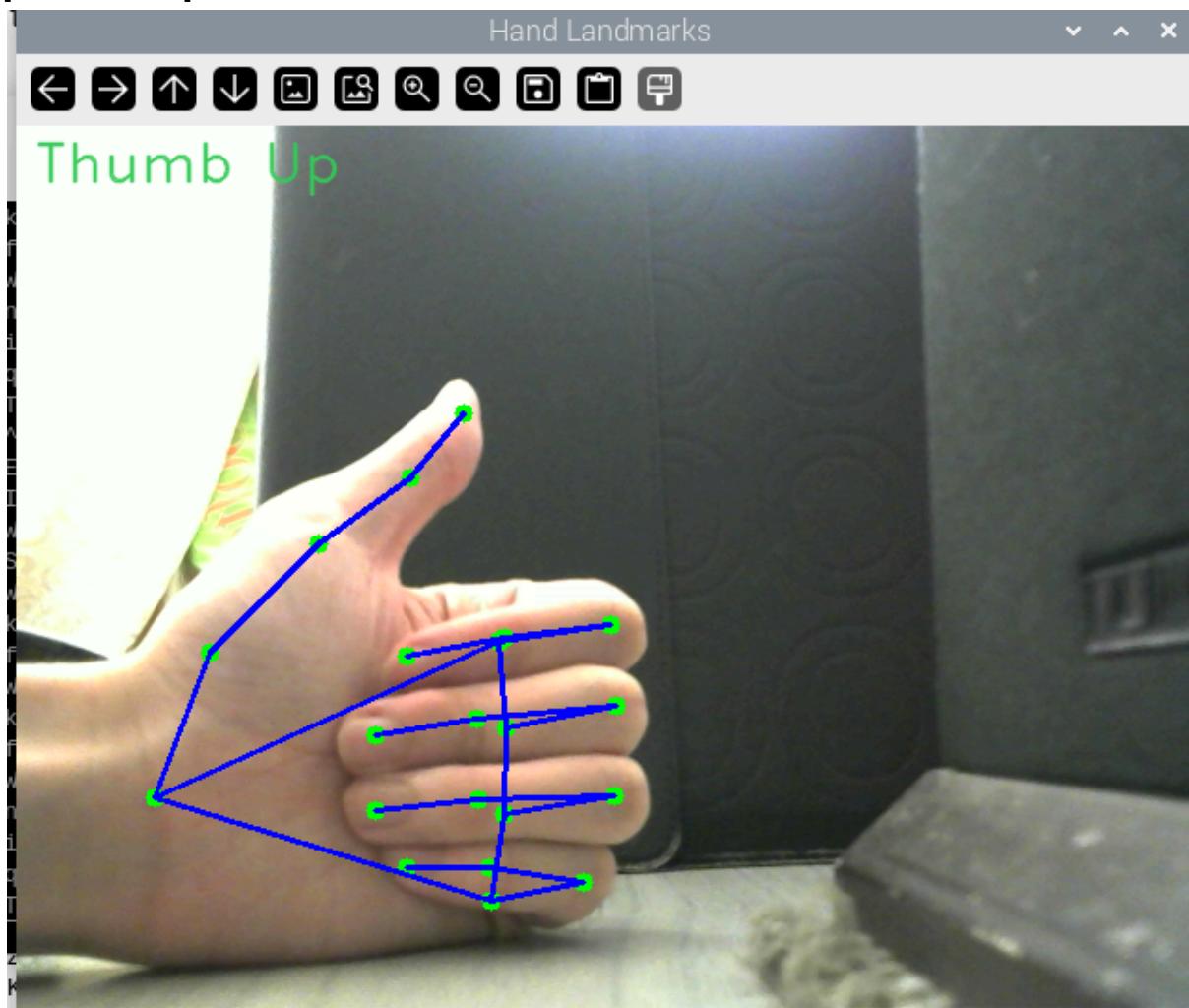


**[Modify the code to show all the 21 finger points and observe the same while moving the hand.]**

[Default]

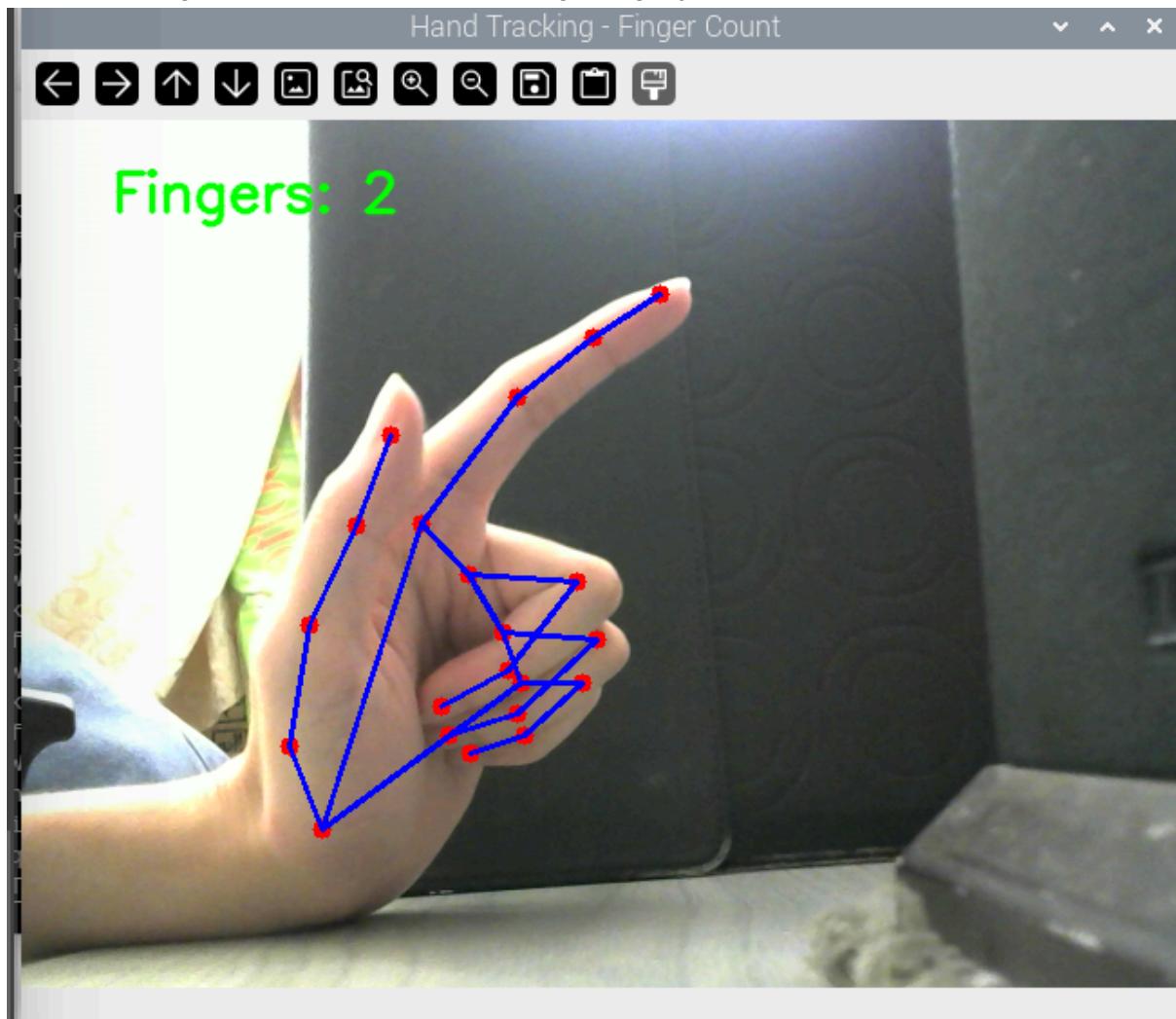


[21 landmarks]



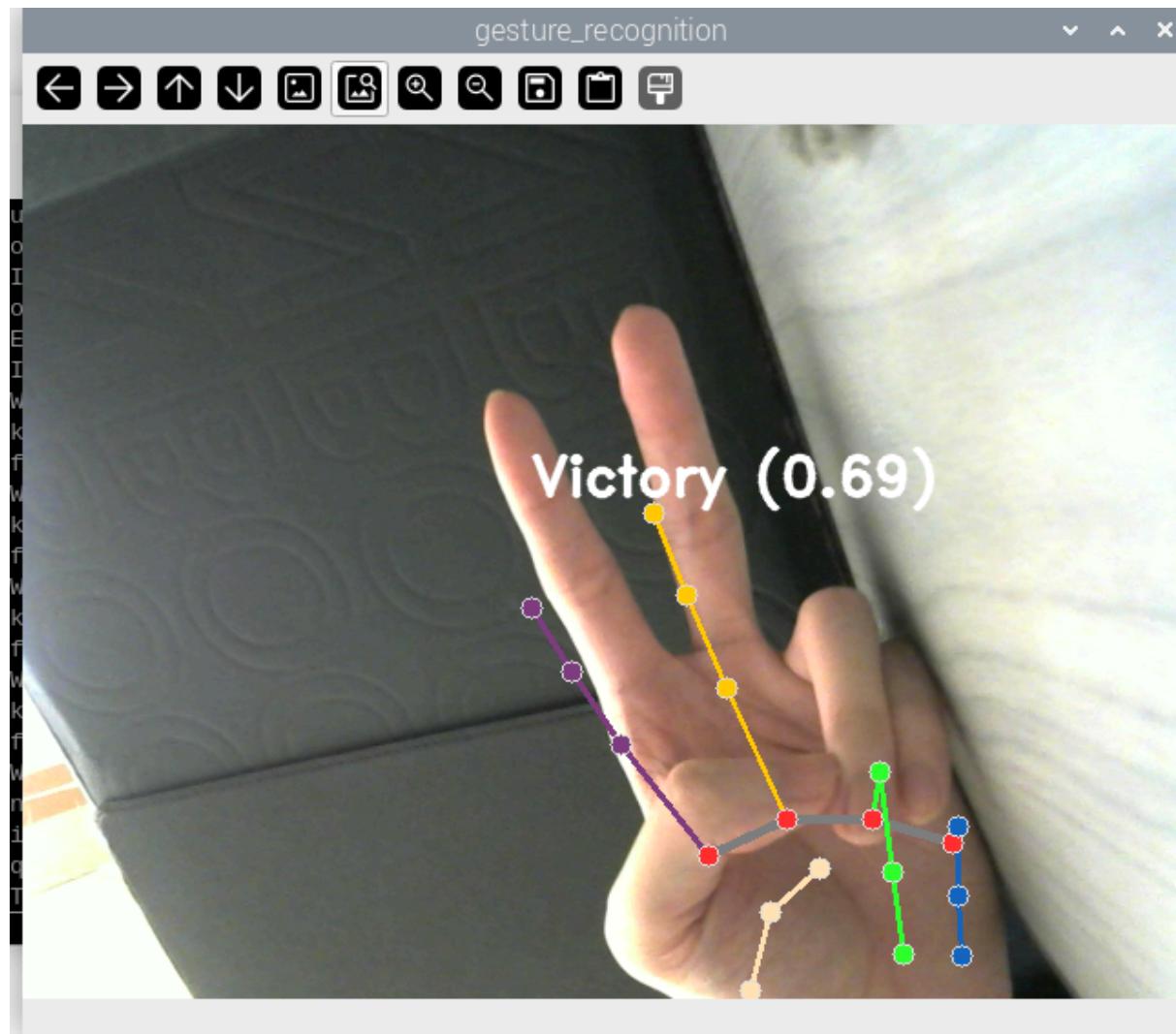
**[Modify the code to predict the number of fingers and display the same overlaid on the image as text (e.g. if four fingers are raised, display '4' on the screen and if three fingers on one hand and two on the other, the display should be '5').]**

[number of fingers up = counted via checking if finger joints are above base]



[Based on the above code, write a code to do object detection based video summarization (e.g. for a video with only frames having a cellphone)]

[Default]



[cell phone detection - blacked out for privacy due to reflection on phone screen]

