Lab 3 & 4 – Neo Lok Jun

Note: Screenshots of setting up of venv and installing requirements are not included.

Table of Contents

Lab 3

3. Introduction to Real-time Image Processing with Python (25 minutes)

1. Testing the code

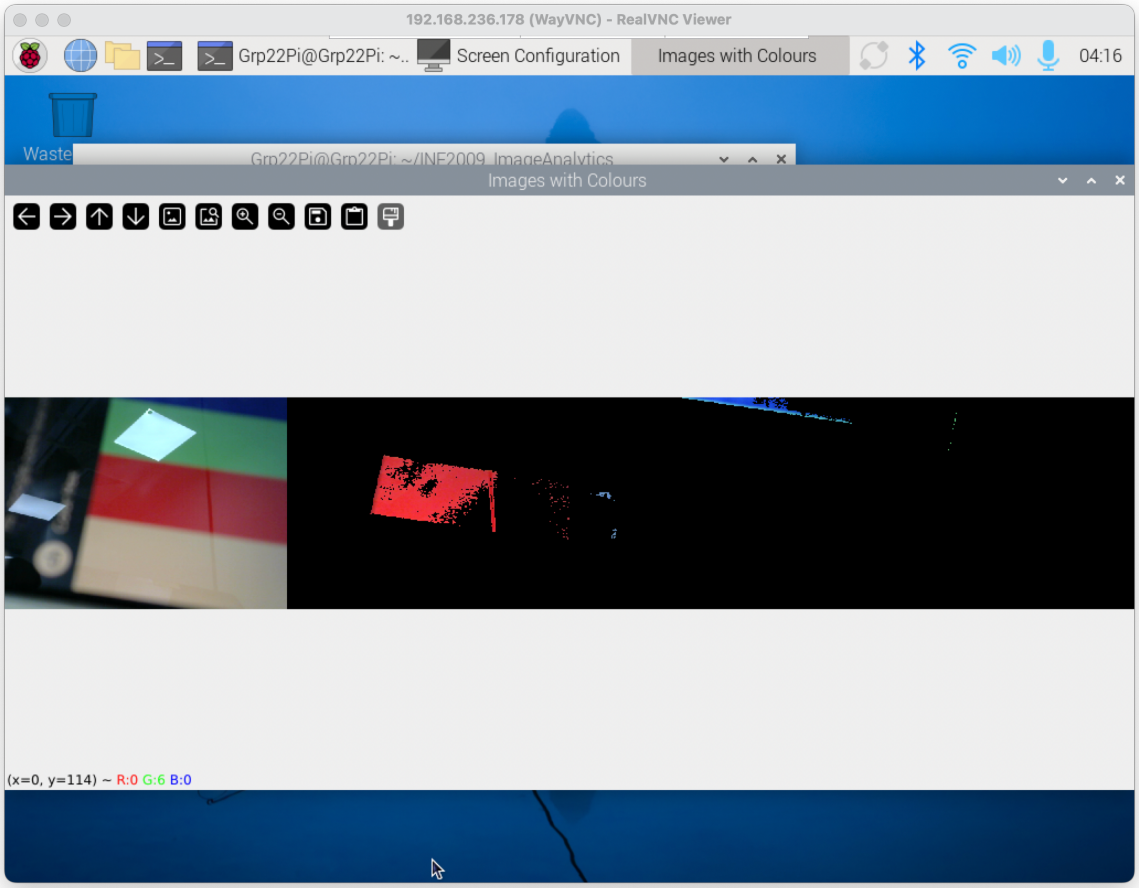


Figure 1: Testing code

2. Then, expand code to segment yellow (right most side, ([20, 100, 100], [30, 255, 255]))

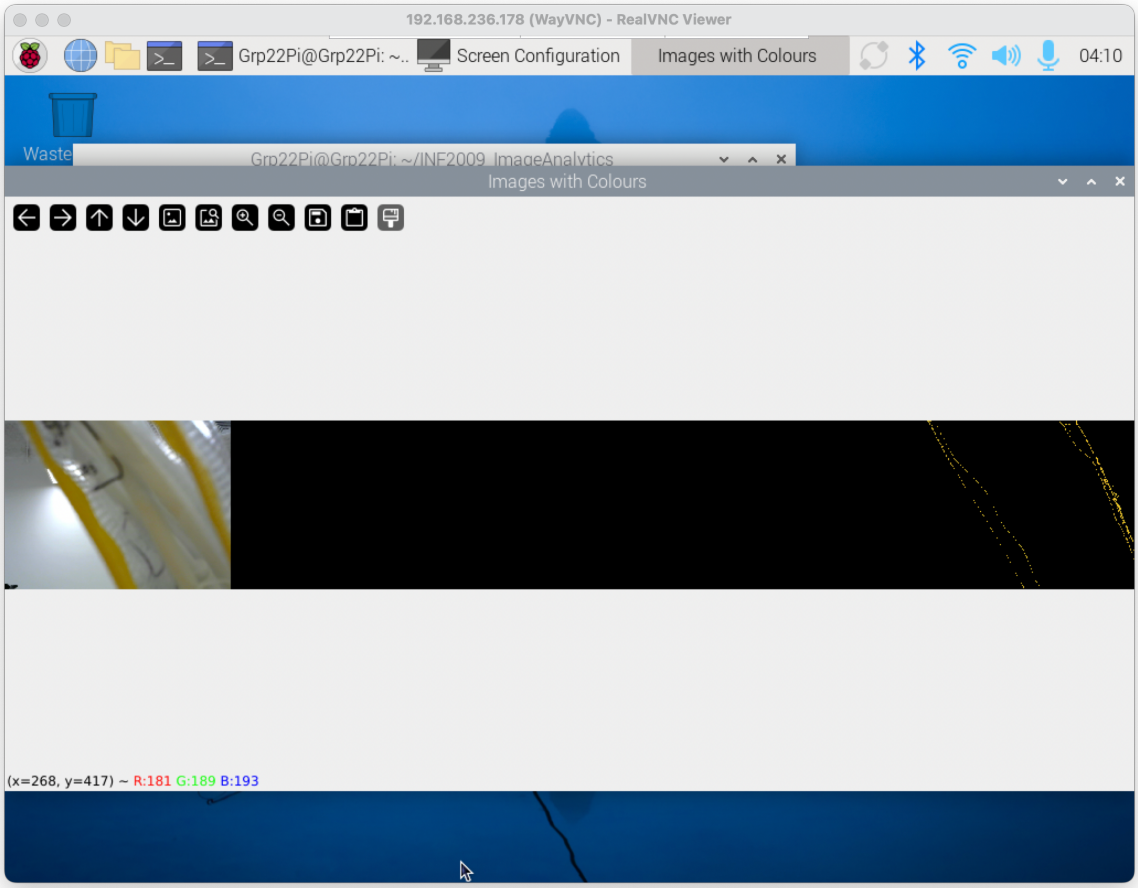


Figure 2: Added yellow

5. Real-time Image Analysis (25 minutes)

1. Run initial code

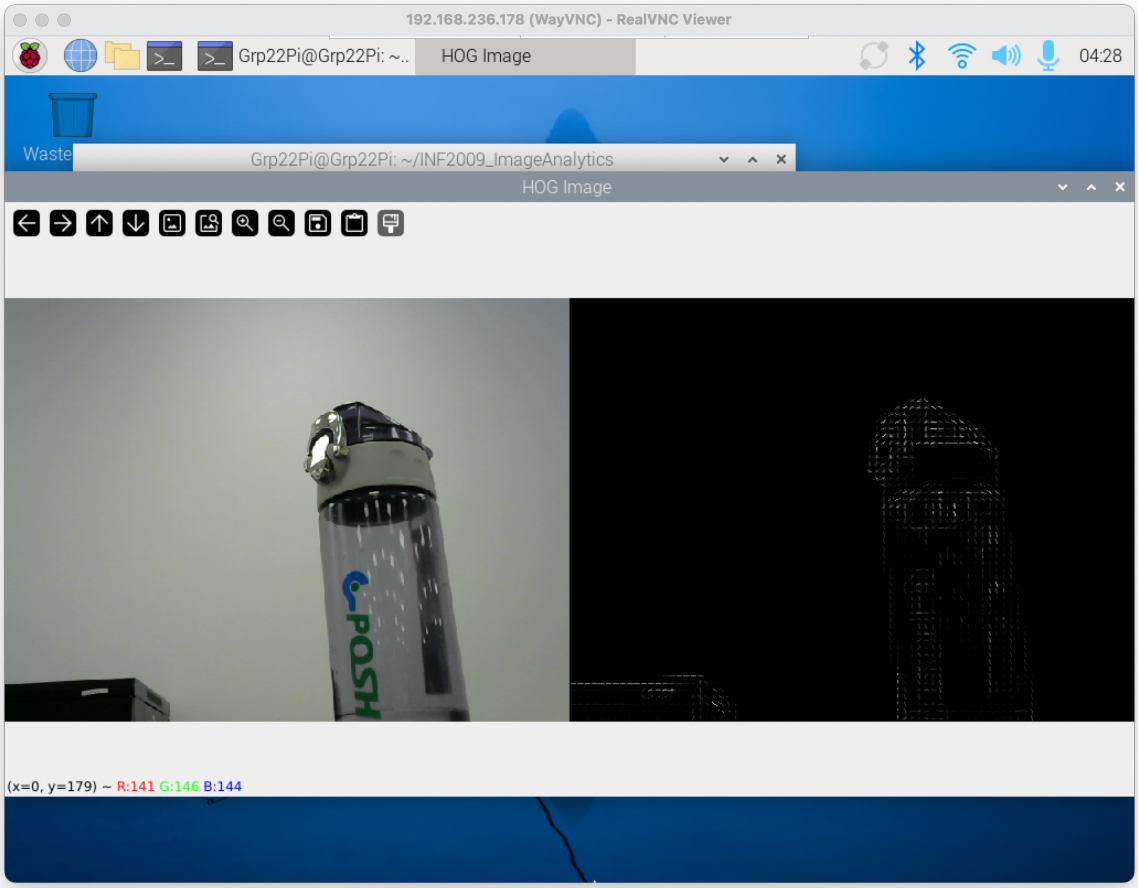


Figure 3: Testing code

2. Ran code with resizing of image (resulted in faster performance)

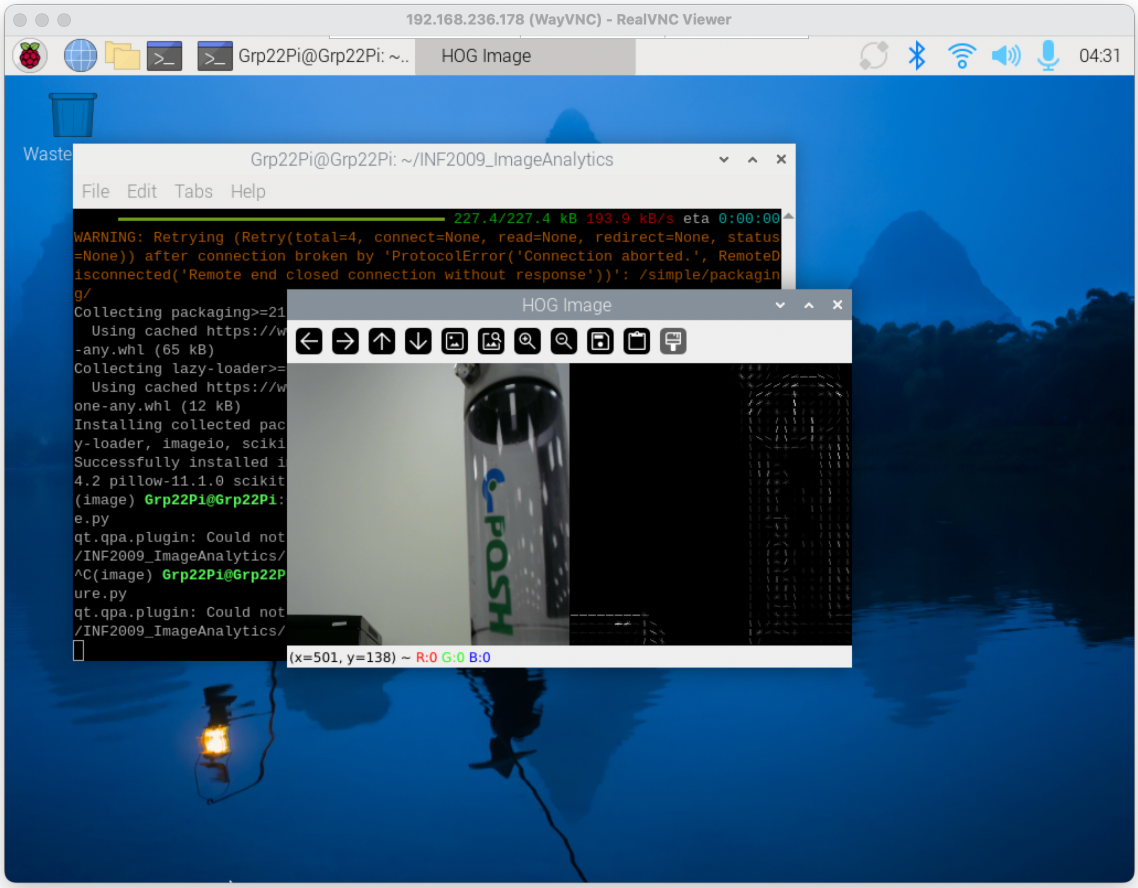


Figure 4: After implementing resize: Faster performance

3. Change patch size on line 25

Referencing https://selfdriving5.github.io/udacity/Self-Driving%20Car%20Engineer%20v5.0.0(us)/Part%2003-Module%2001-Lesson%2001_Object%20Detection/20.%20scikit-image%20HOG.html, I adjusted the patch size and tried out (16,16) pixels per cell and (4,4) for cells per block.

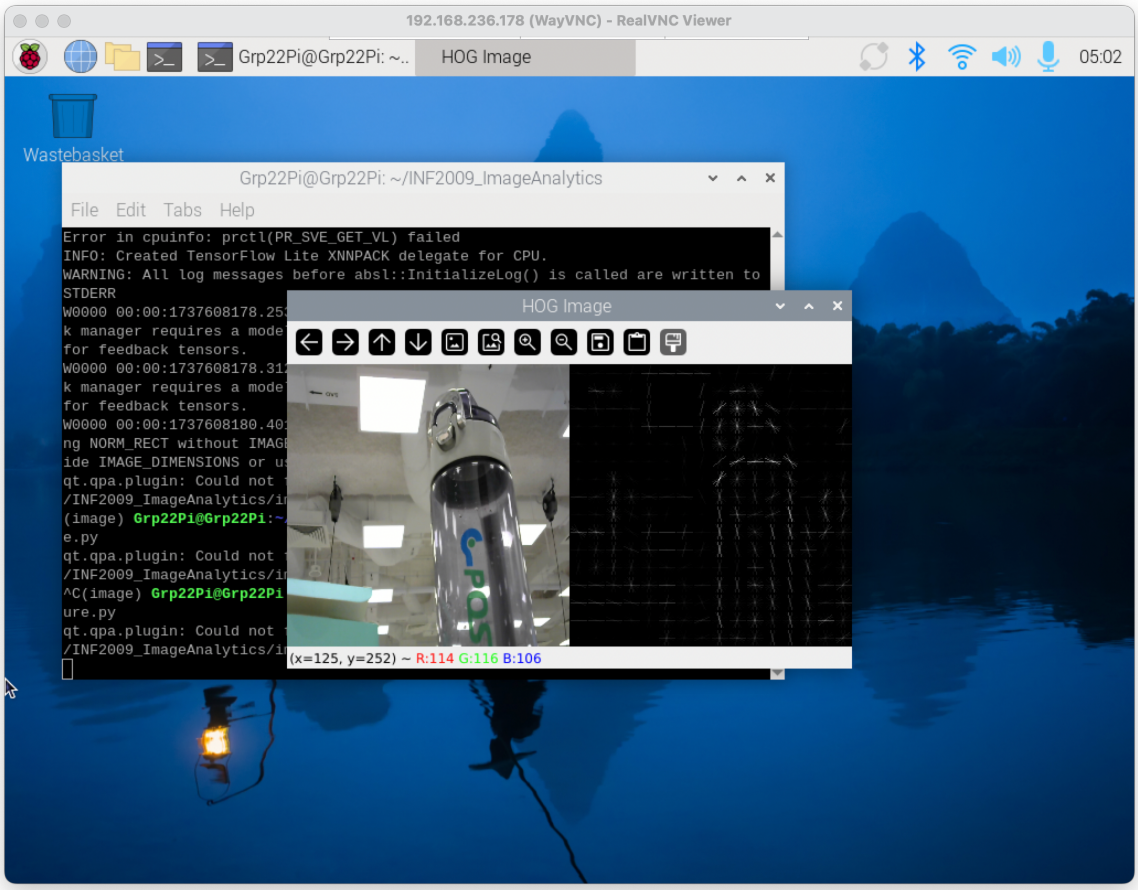


Figure 5: Modified patch size to 16,16) pixels per cell and (4,4) for cells per block.

Naturally, the white lines in the feature extraction became “longer”.

4. Extract HoG Features for Identification

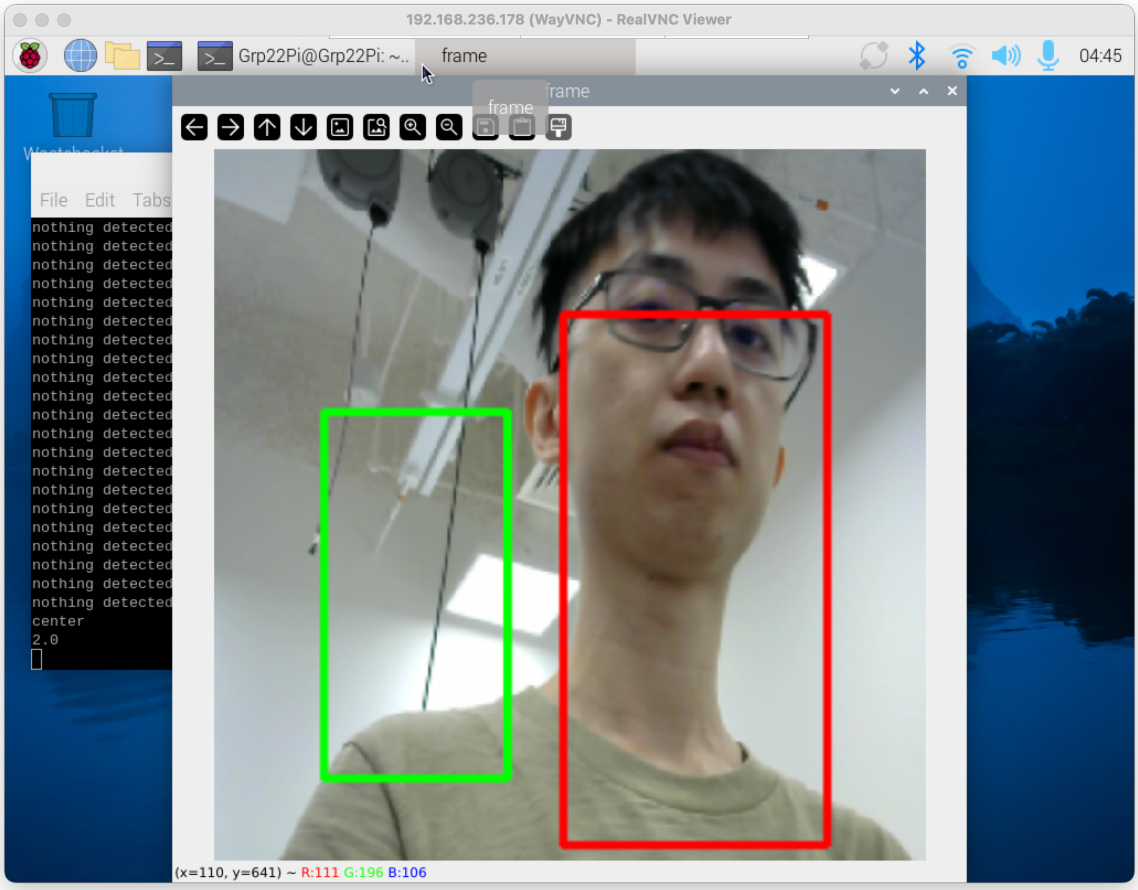


Figure 6: Identification using HoG feature extraction

(Feature extraction was not very good, failed to detect face on multiple occasion)

6. Real-time Image Feature Analysis for Face Capture and Facial Landmark Extraction (20 minutes)

1. Mediapipe feature analysis

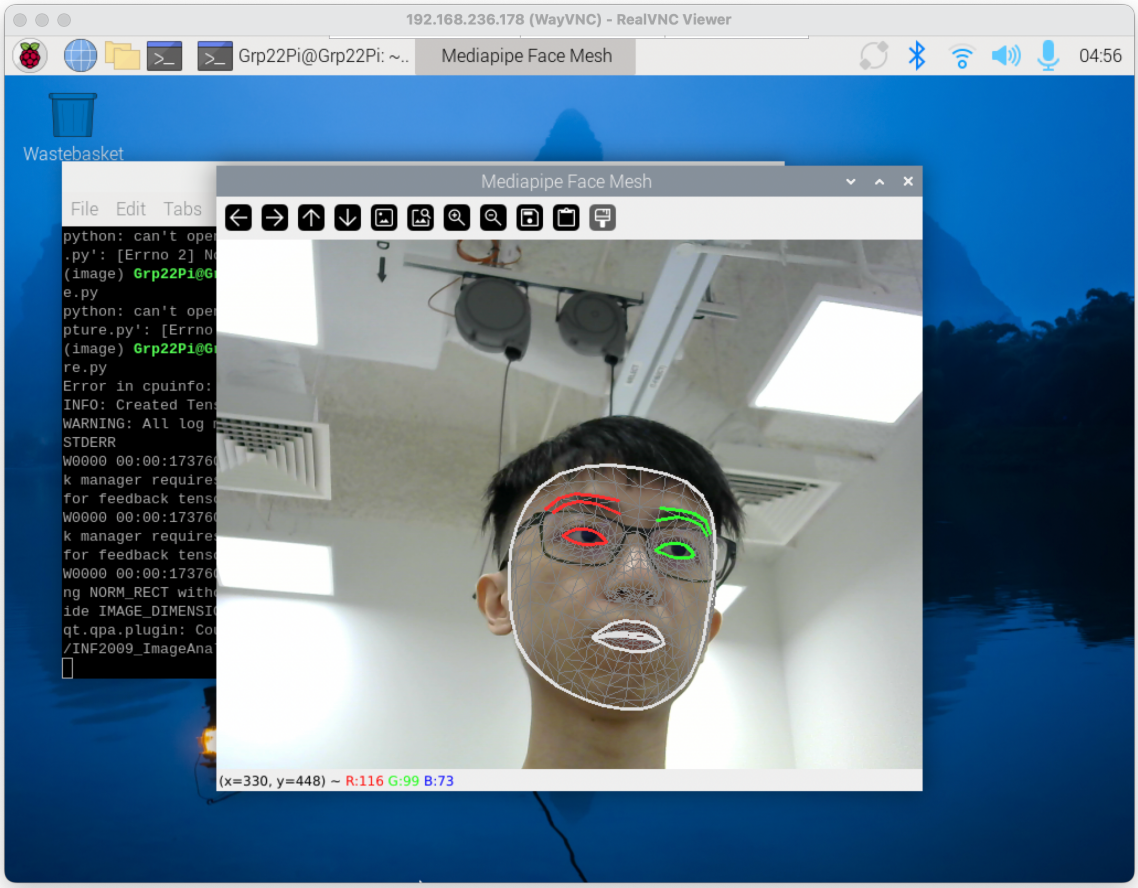
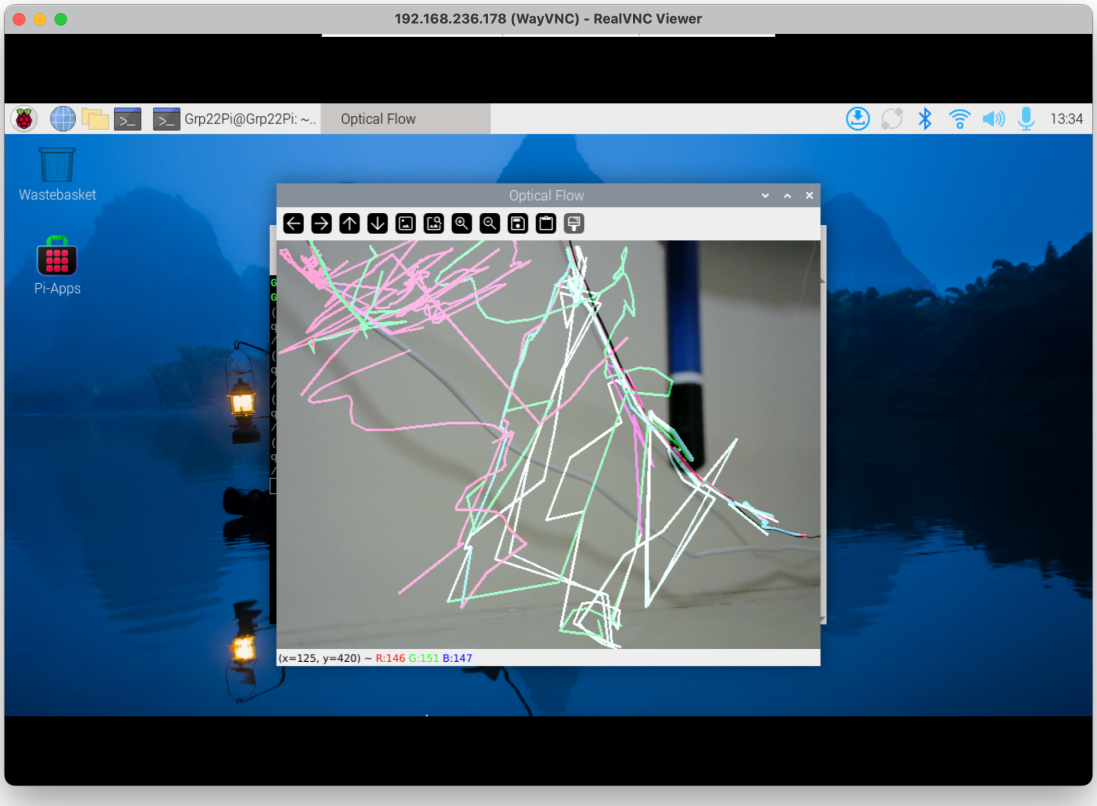


Figure 7: Mediapipe's approach

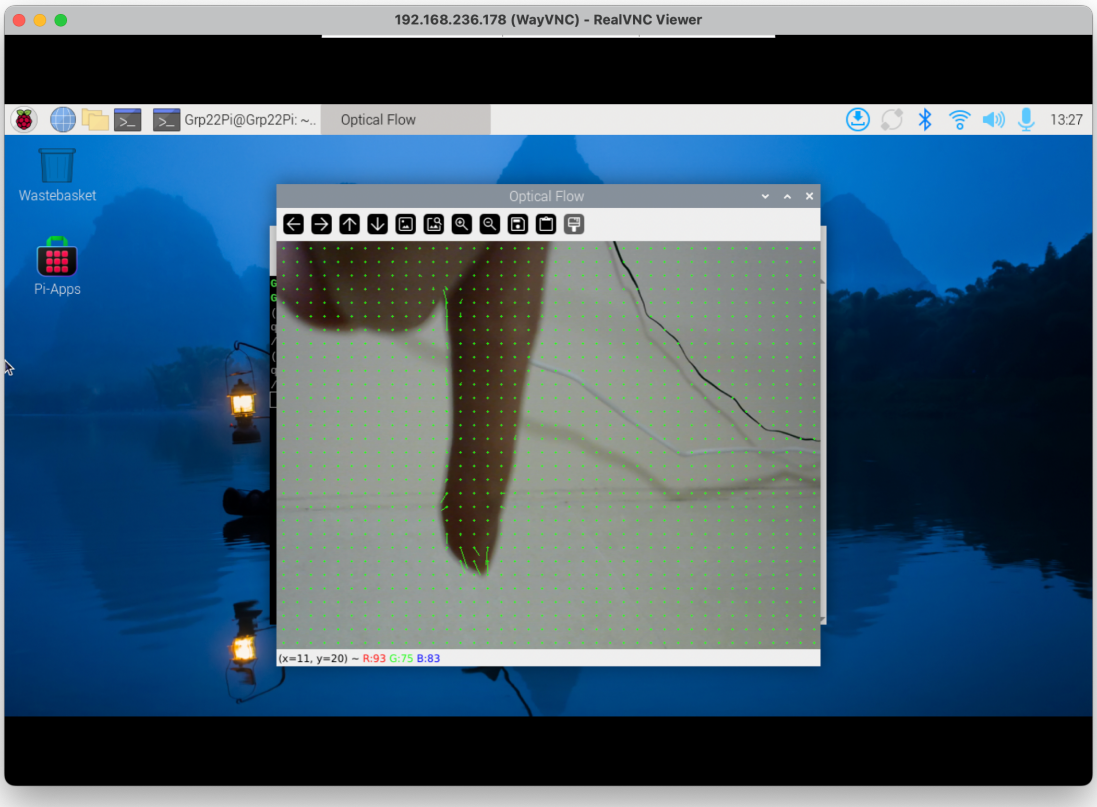
Lab 4

Introduction to real-time video processing on raspberry pi (20 minutes)

1. LucasKanadeOpticalFlow

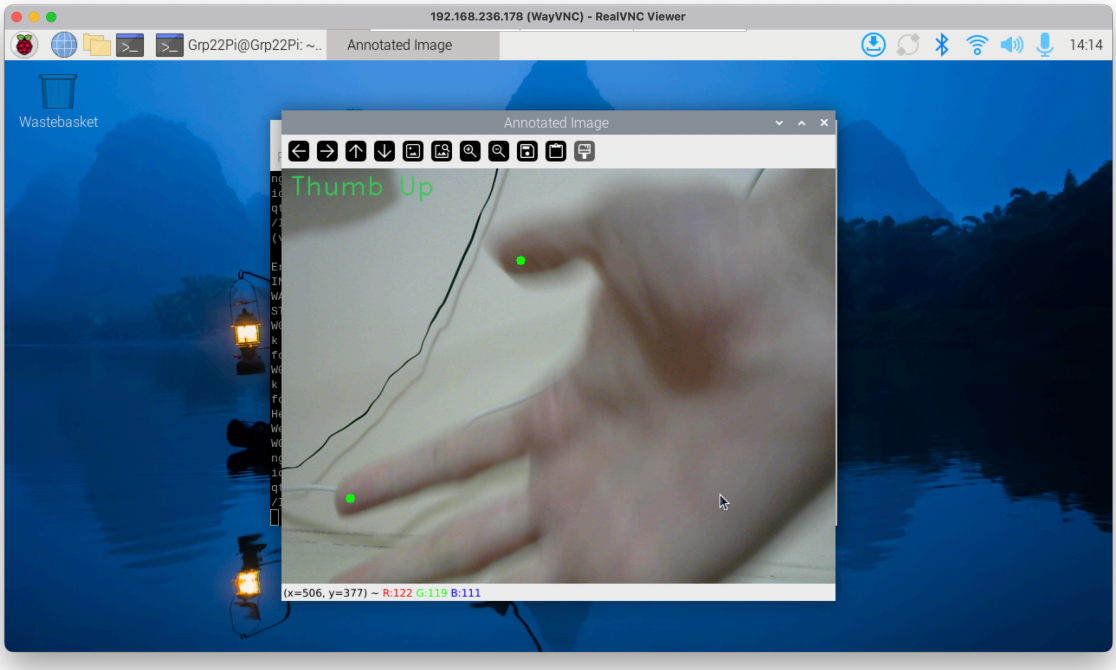


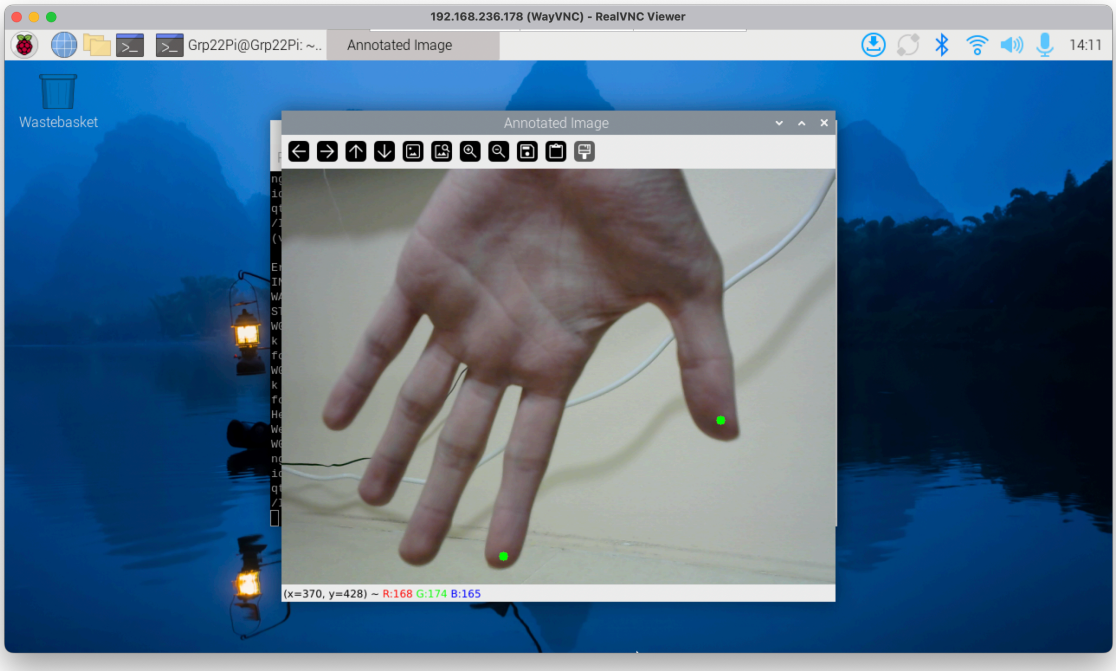
2. DenseOpticalFlowByLines



5. Advanced Video Analytics (40 minutes)

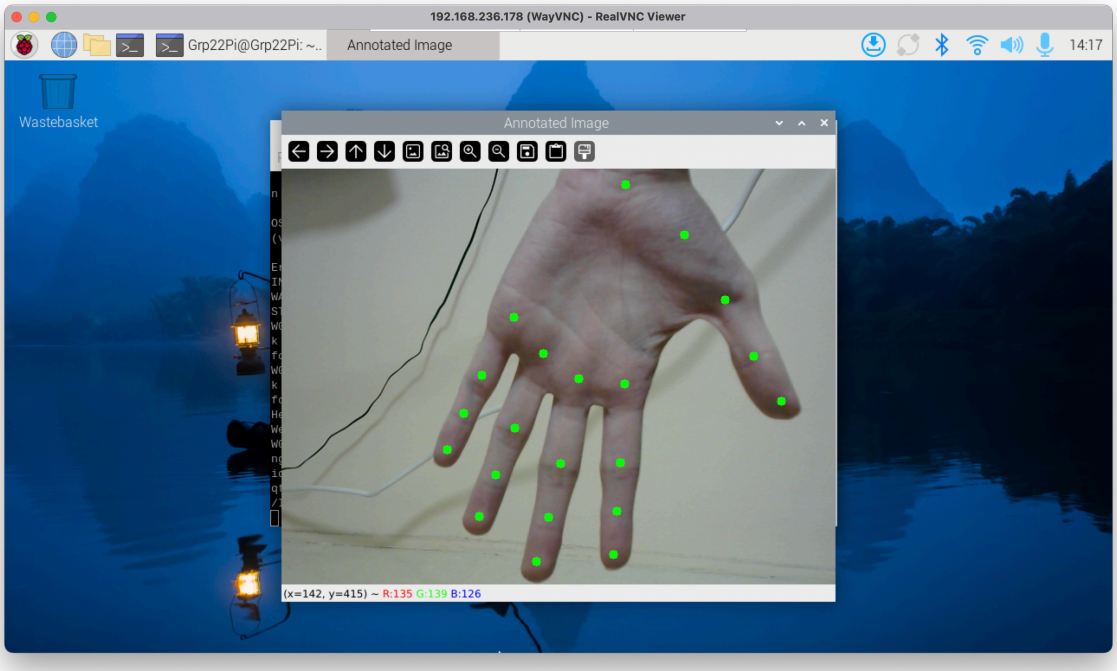
Handmark detection - Run default code





Handmark detection – All 21 finger points

```
# Loop through the detected hands to visualize.
for idx in range(len(hand_landmarks_list)):
    hand_landmarks = hand_landmarks_list[idx]

    for landmark in hand_landmarks:
        x = int(landmark.x * frame.shape[1])
        y = int(landmark.y * frame.shape[0])
        cv2.circle(frame, (x, y), 5, (0, 255, 0), -1)
```

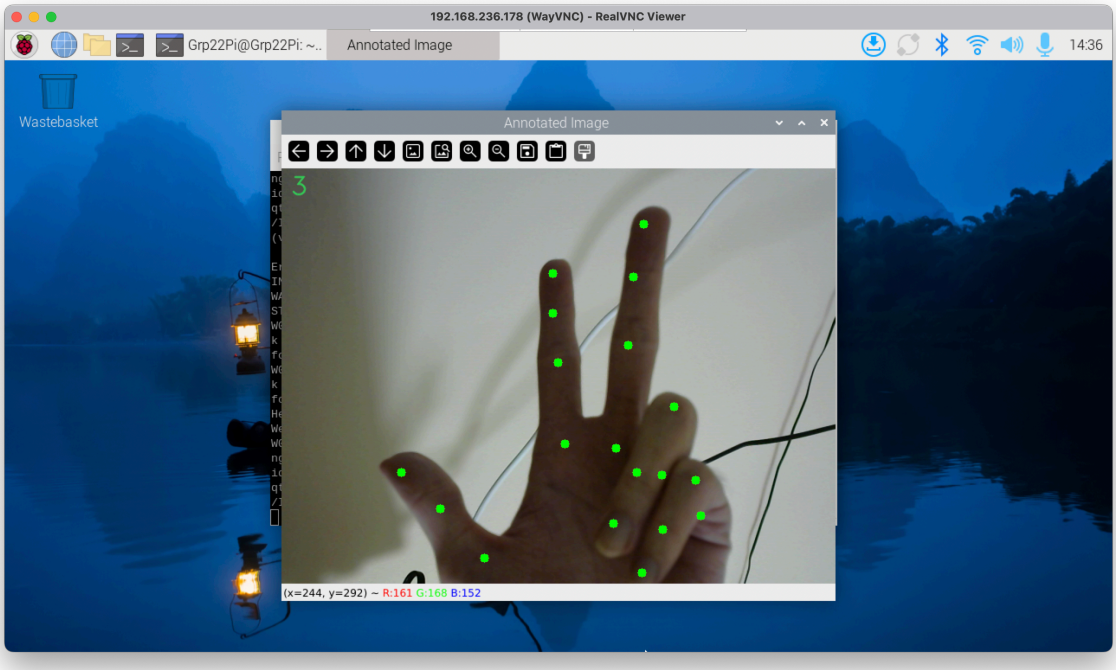


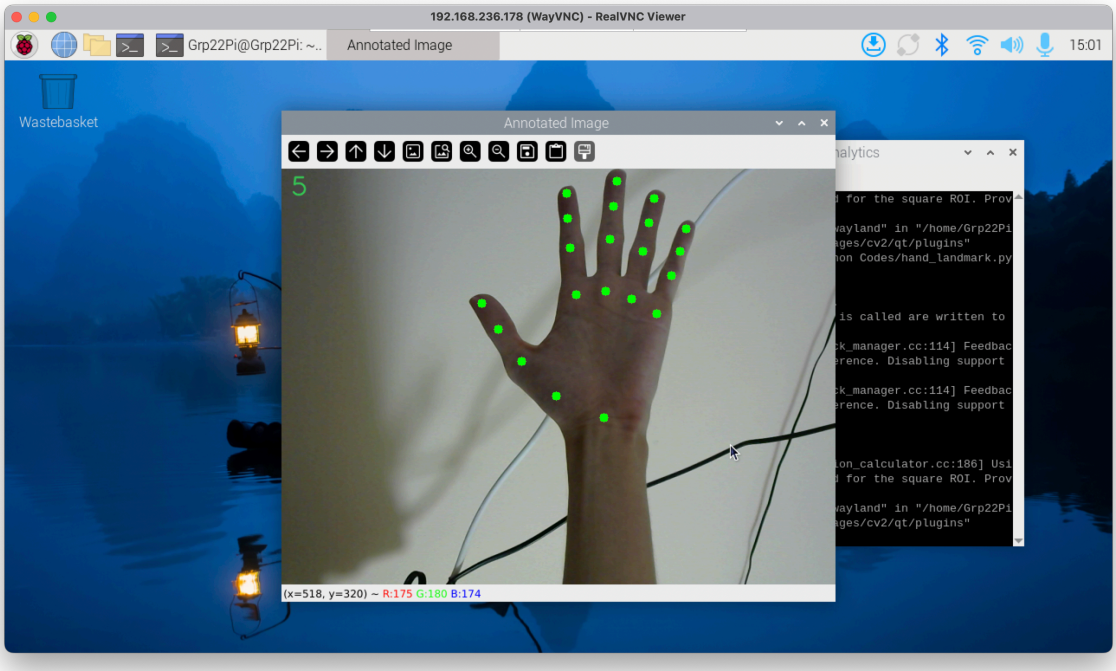
Handmark detection – Detect number of fingers up

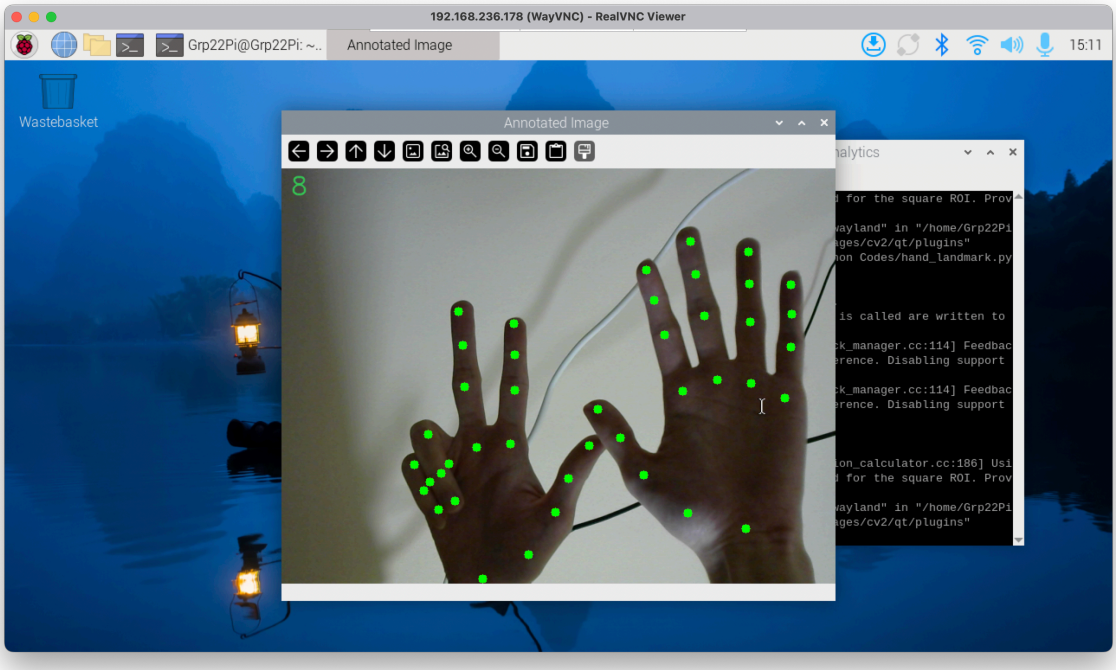
The following code implementation assumes that:

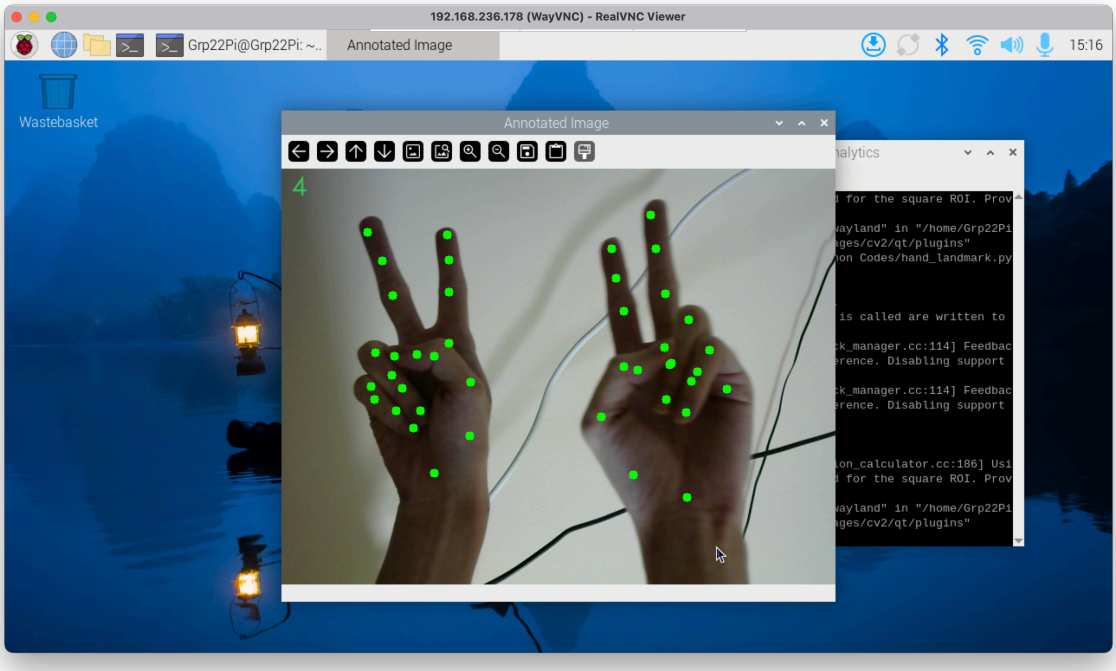
1. Tip of the finger must be in the image to be counted as pointed up
2. Tip of the finger must be above the immediate lower part of the finger to be considered “pointed up”
3. Threshold of 0.05 is added to account for thumb or slanted finger positions (if y position of tip of finger is too close to lower part, ignore it)

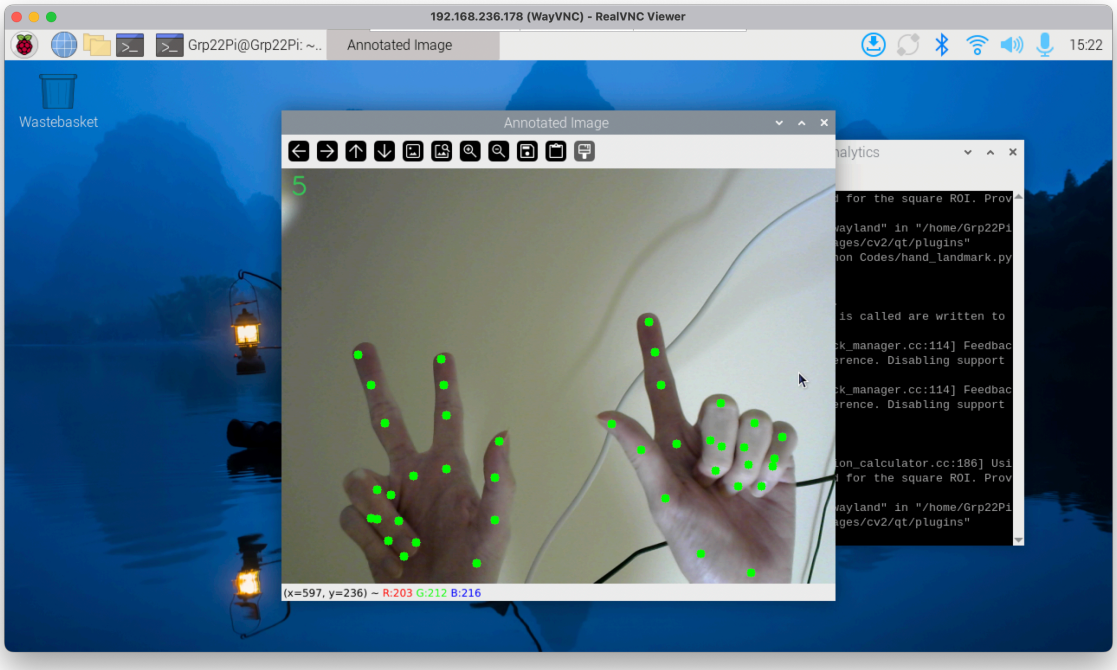
```
threshold = 0.05
for i in range(4,21, 4):
    # if i == 4: threshold = 0.035
    # check if tip of finger is below immediate lower part, and >0 to make sure finger is on screen
    if hand_landmarks[i].y + threshold < hand_landmarks[i-1].y and hand_landmarks[i].y > 0 and hand_landmarks[i].x > 0:
        num_fingers += 1

if num_fingers and idx == (len(hand_landmarks_list)-1):
    cv2.putText(frame, str(num_fingers), (10,30),
                cv2.FONT_HERSHEY_DUPLEX,
                FONT_SIZE, HANDEDNESS_TEXT_COLOR, FONT_THICKNESS, cv2.LINE_AA)

cv2.imshow('Annotated Image', frame)
```

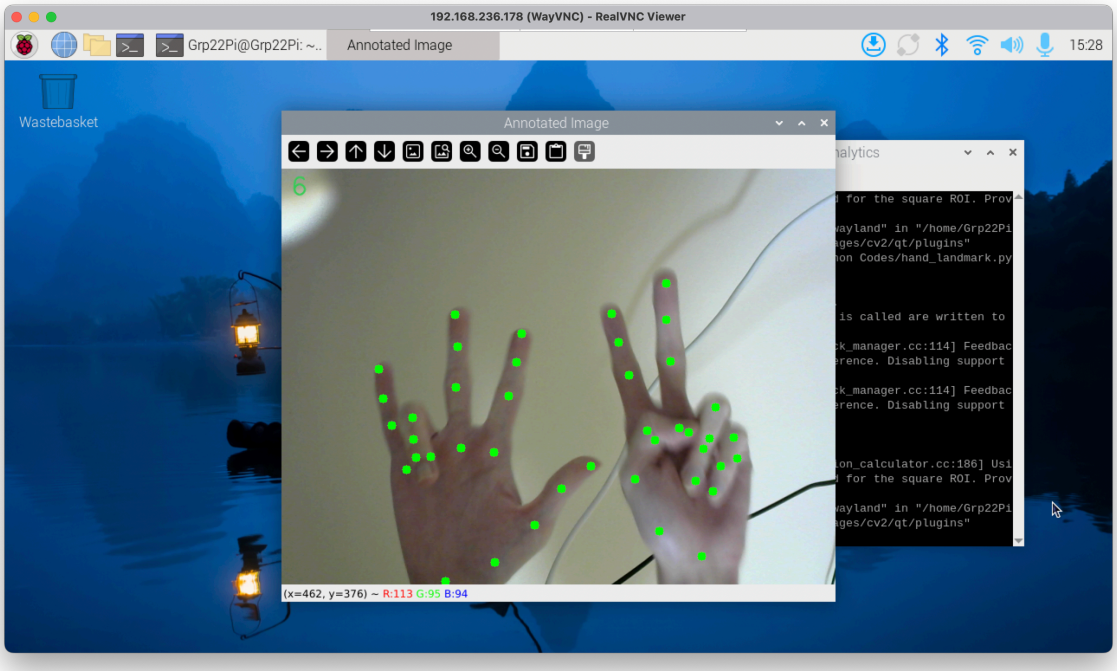

192.168.236.178 (WayVNC) - RealVNC Viewer
Grp22Pi@Grp22Pi: ~..
Annotated Image
14:36
Wastebasket
Annotated Image
3
(x=244, y=292) ~ R:161 G:168 B:152


192.168.236.178 (WayVNC) - RealVNC Viewer
Grp22Pi@Grp22Pi: ~..
Annotated Image
15:01
Wastebasket
Annotated Image
5
(x=518, y=320) ~ R:175 G:180 B:174


192.168.236.178 (WayVNC) - RealVNC Viewer
Grp22Pi@Grp22Pi: ~..
Annotated Image
15:11
Wastebasket
Annotated Image
8
for the square ROI. Prov
wayland" in "/home/Grp22Pi
ages/cv2/qt/plugins"
ion Codes/hand_landmark.py
is called are written to
k_manager.cc:114] Feedbac
erence. Disabling support
k_manager.cc:114] Feedbac
erence. Disabling support
ion_calculator.cc:186] Usi
for the square ROI. Prov
wayland" in "/home/Grp22Pi
ages/cv2/qt/plugins"

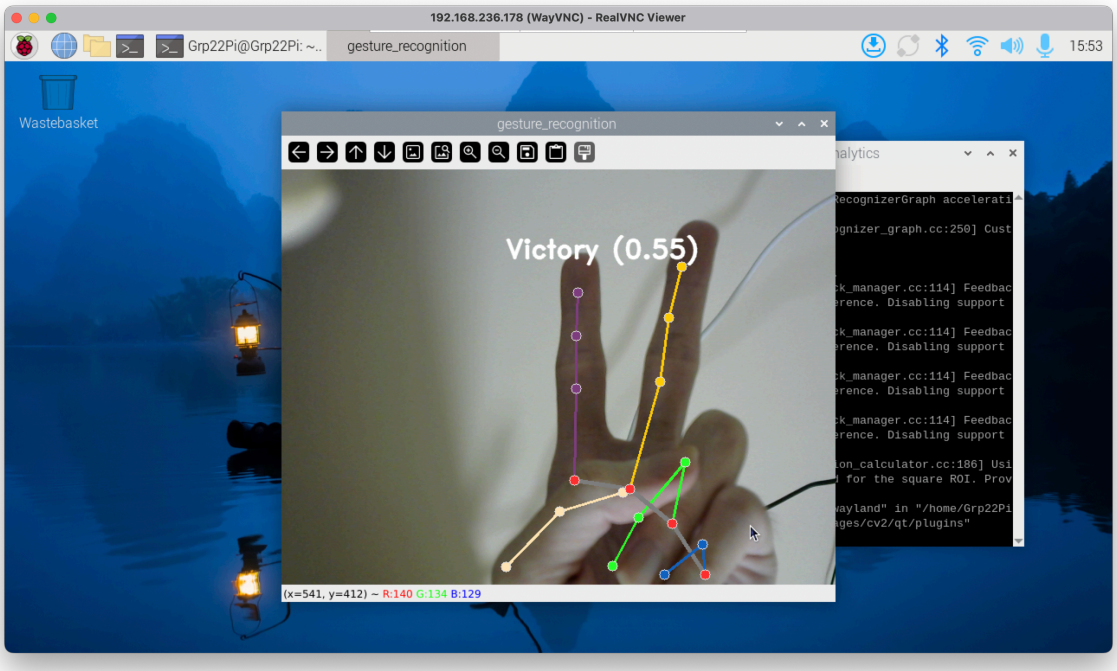

192.168.236.178 (WayVNC) - RealVNC Viewer
Grp22Pi@Grp22Pi: ~..
Annotated Image
15:16
Wastebasket
Annotated Image
4
for the square ROI. Prov
wayland" in "/home/Grp22Pi
ages/cv2/qt/plugins"
ion Codes/hand_landmark.py
is called are written to
k_manager.cc:114] Feedbac
erence. Disabling support
k_manager.cc:114] Feedbac
erence. Disabling support
ion_calculator.cc:186] Usi
for the square ROI. Prov
wayland" in "/home/Grp22Pi
ages/cv2/qt/plugins"



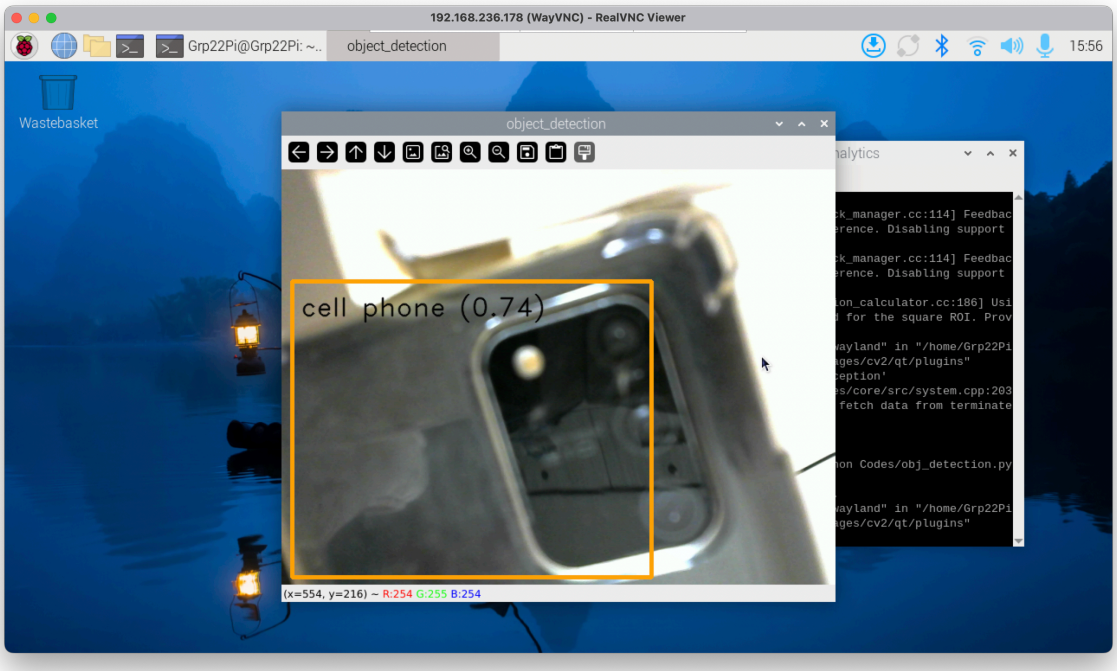


6. Advanced Video Analytics (20 minutes)

Hand_gesture



Object Detection (Default code/ Live object detection)



Object Detection (Object Detection based Video Summarization)

Video from: https://videos.pexels.com/video-files/9558563/9558563-sd_426_226_25fps.mp4

Create a videowriter (avi is used because opencv supports avi only)

```
out = cv2.VideoWriter('output.avi',
                      cv2.VideoWriter_fourcc('M', 'J', 'P', 'G'),
                      cap.get(cv2.CAP_PROP_FPS),
                      (int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)),int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)))
                     )
```

Write frame only if cell phone is detected

```
        if "cell phone" in category_name:
            phone_detected = 1

    detection_frame = current_frame
    detection_result_list.clear()

    if phone_detected:
        #print("Writing...")
        out.write(current_frame)
        pass
```

Then close video writer at the end

```
out.release()
```

The output video only has a remaining length of 3 seconds

(due to VNC lag, I exported video from RPi to a windows pc, and viewed it from there)

Original video had 10s, output video only has 3 seconds.