

INF2009 - Lab 3

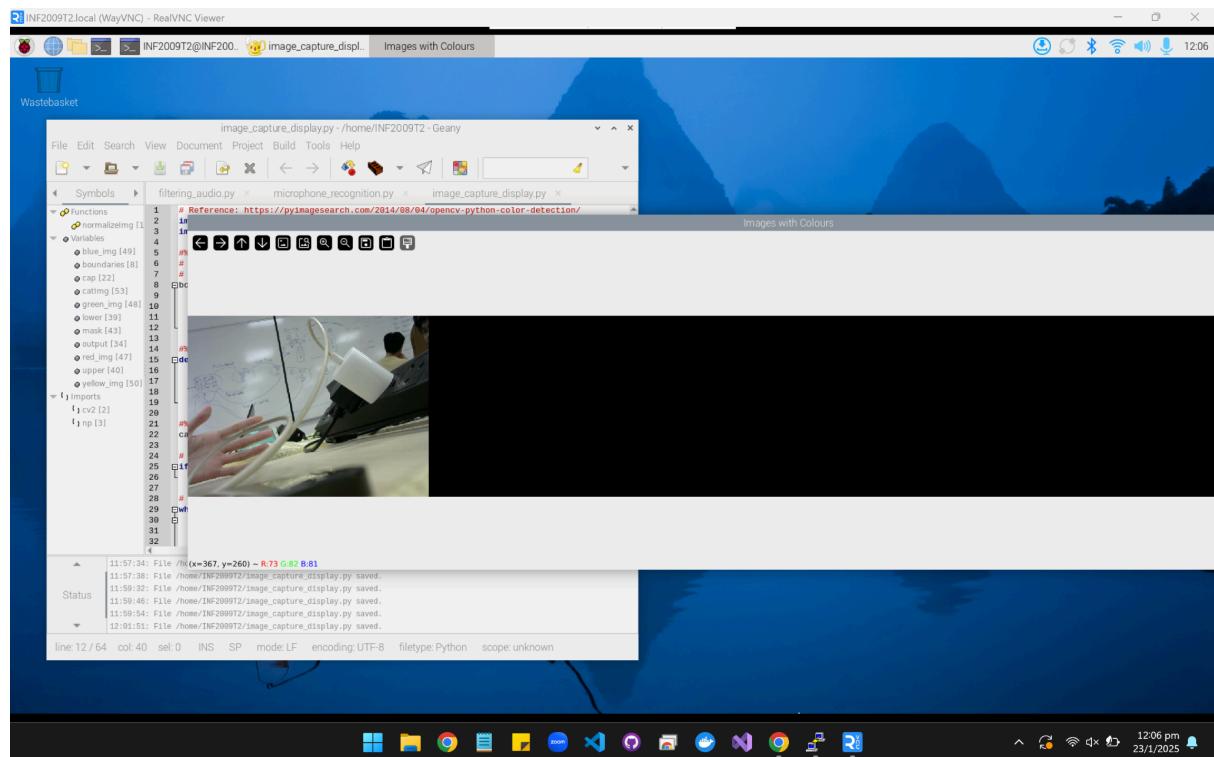
[Expand the code to segment another colour - Yellow]

```
# Refer to https://docs.opencv.org/3.4/da/d97/tutorial_threshold_inRange.html
boundaries = [
    ([17, 15, 100], [50, 56, 200]), # For Red
    ([86, 31, 4], [220, 88, 50]), # For Blue
    ([25, 90, 4], [62, 200, 50]), # For Green
    ([0, 120, 120], [80, 255, 255])] # For Yellow

# Output is appended to be of size Pixels X 3 (for R, G, B)
red_img = normalizeImg(output[0])
green_img = normalizeImg(output[1])
blue_img = normalizeImg(output[2])
yellow_img = normalizeImg(output[3])

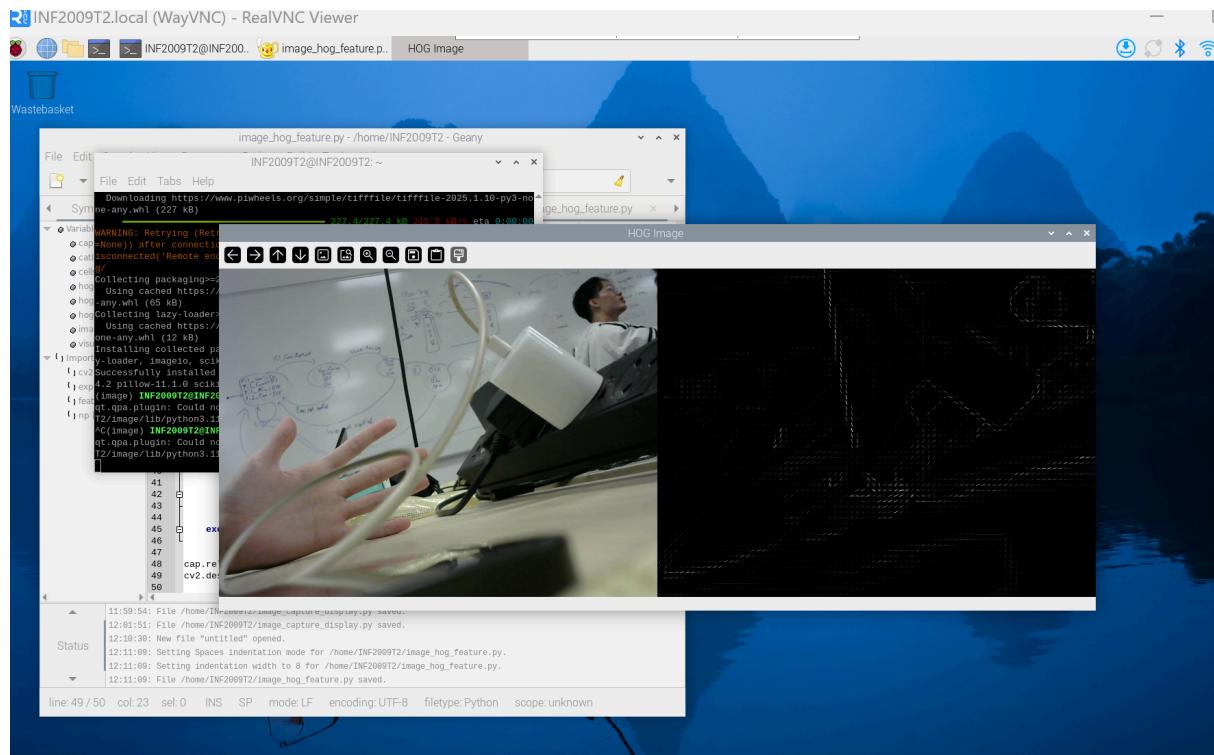
# horizontal Concatination for displaying the images and colour segmentations
catImg = cv2.hconcat([frame,red_img,green_img,blue_img,yellow_img])
cv2.imshow("Images with Colours",catImg)
```

[Change the patch size in line 25 (feature.hog) and observe the changes in the results] [No HoG]



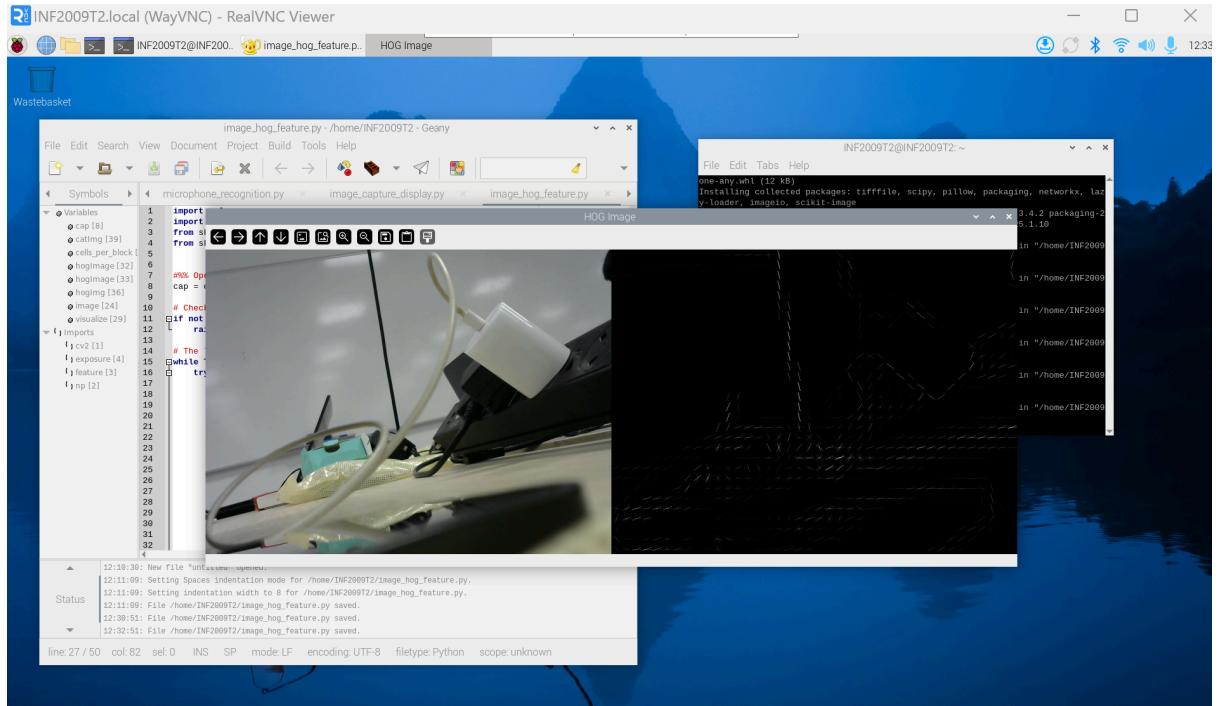
Default Patch Size [8,8]

```
# Extract the HoG features from the image
(H, hogImage) = feature.hog(image, orientations=9, pixels_per_cell=(8, 8),
cells_per_block=(2, 2), transform_sqrt=True, block_norm="L1",
visualize=True)
```



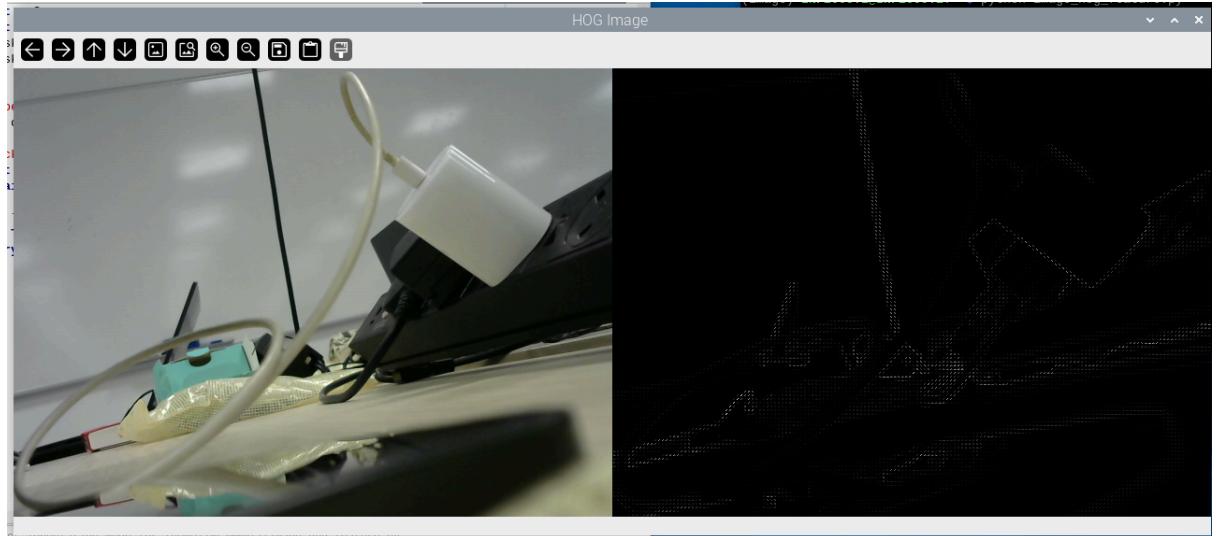
Patch Size [16,16] - very chopped up HoG image

```
# Extract the HoG features from the image
(H, hogImage) = feature.hog(image, orientations=9, pixels_per_cell=(16, 16),
cells_per_block=(2, 2), transform_sqrt=True, block_norm="L1",
visualize=True)
```



Patch Size [4,4] - smoother HoG image

```
# Extract the HoG features from the image
(H, hogImage) = feature.hog(image, orientations=9, pixels_per_cell=(4, 4),
cells_per_block=(2, 2), transform_sqrt=True, block_norm="L1",
```



[Try the sample code to detect the face based on Mediapipe's approach]

