Henry Kou: 204921239, Team 6, EE180DA **Week 0 Lab Report**

1. **Action Items (plan)**
    a. Join group - completed (Group 6)
    b. Purchase parts - completed
    c. Tutorial 0 Intro to SW and CV
        i. Task 1 - **completed**

Learned about the basics of git interface from remote to local repositories. Link to the github remote repo is https://github.com/kenryhou2/180DA-WarmUp .

ii. Task 2 - **completed**

Installed miniconda and added test.txt file to the local repo. Then uploaded to remote.

iii. Task 3 - **completed**

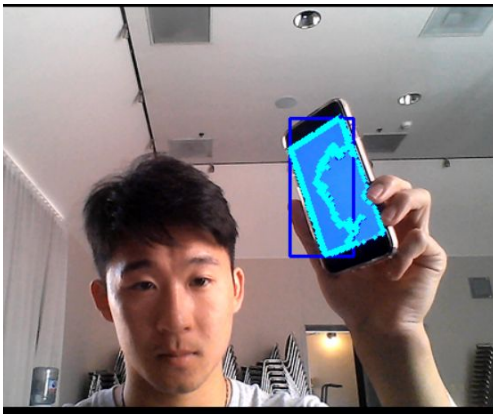
Created and executed code snippet in Sublime and miniconda command line. Then, pushed to remote repo.

File output:

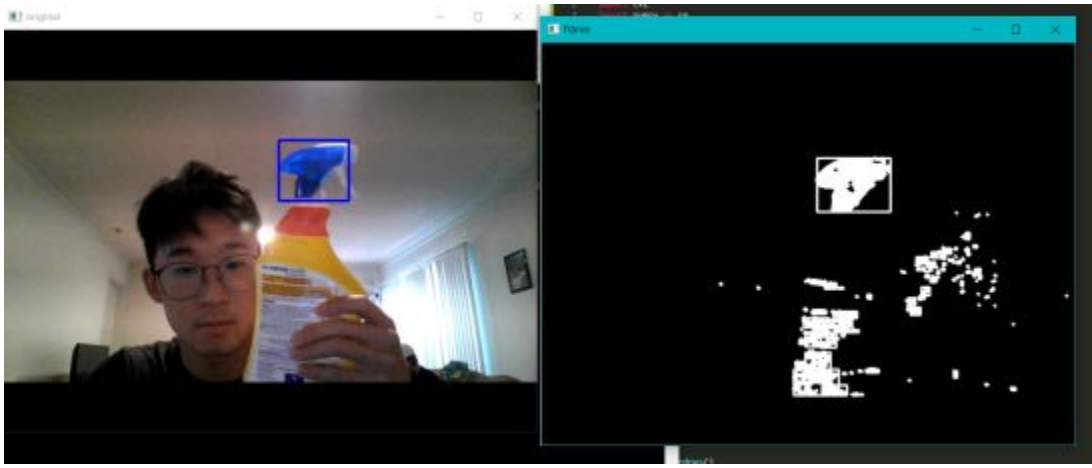
```
(hkou_env_tutorial) C:\Users\Henry Kou\Desktop\EE180DA\tutorial\180DA-WarmUp>python test.py
ECE_180_DA_DB--Best class ever
```

iv. Task 4 - **completed**

HSV and RGB Bounded Box



Bound box with RGB thresholding



Bound box with HSV of non-phone object

Typically, HSV is better for object tracking with cameras because there is added dimension with saturation and light/value ranges.Therefore, the tracking of an object by HSV will be more robust with lighting changes than with RGB. The RGB result had impartial bounding box tracking.

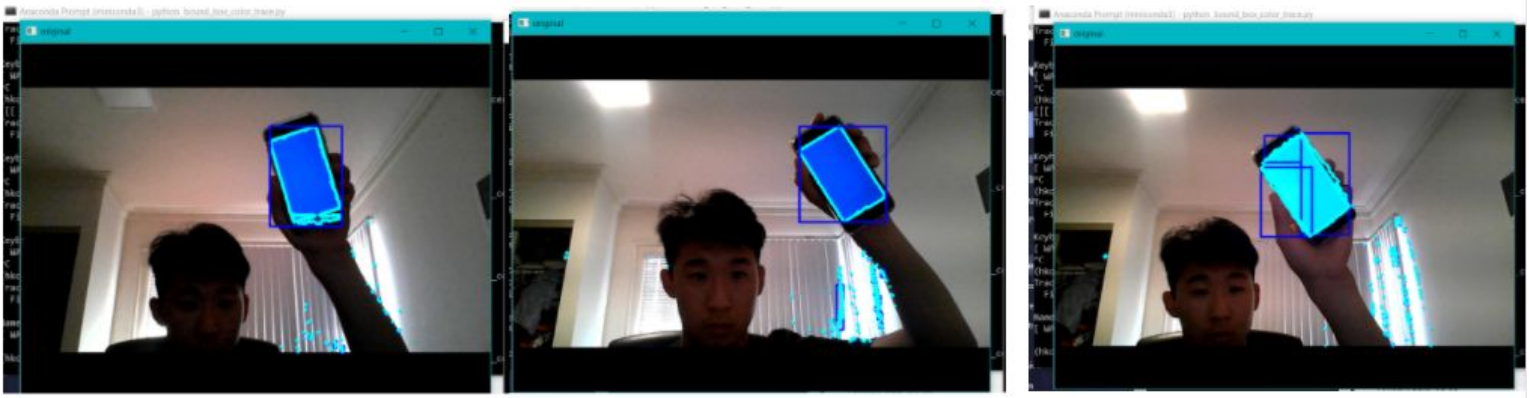
The threshold range for HSV was [20, 153, 153] between high and low thresholds and the threshold for RGB was [70, 70, 60]

Color	Upper Threshold	Lower Threshold	Threshold Range
HSV (blue)	[130,255,255]	[110,50,50]	[20, 153, 153]
RGB (blue)	[92,137,255]	[12,67,195]	[70, 70, 60]

When using HSV to track the phone, there was a slight variation in performance depending on the ambient light. However, with HSV, the robustness of the threshold range allowed it to perform relatively well, enabling decent tracking of the target object with a single bounding box.

Color Picker

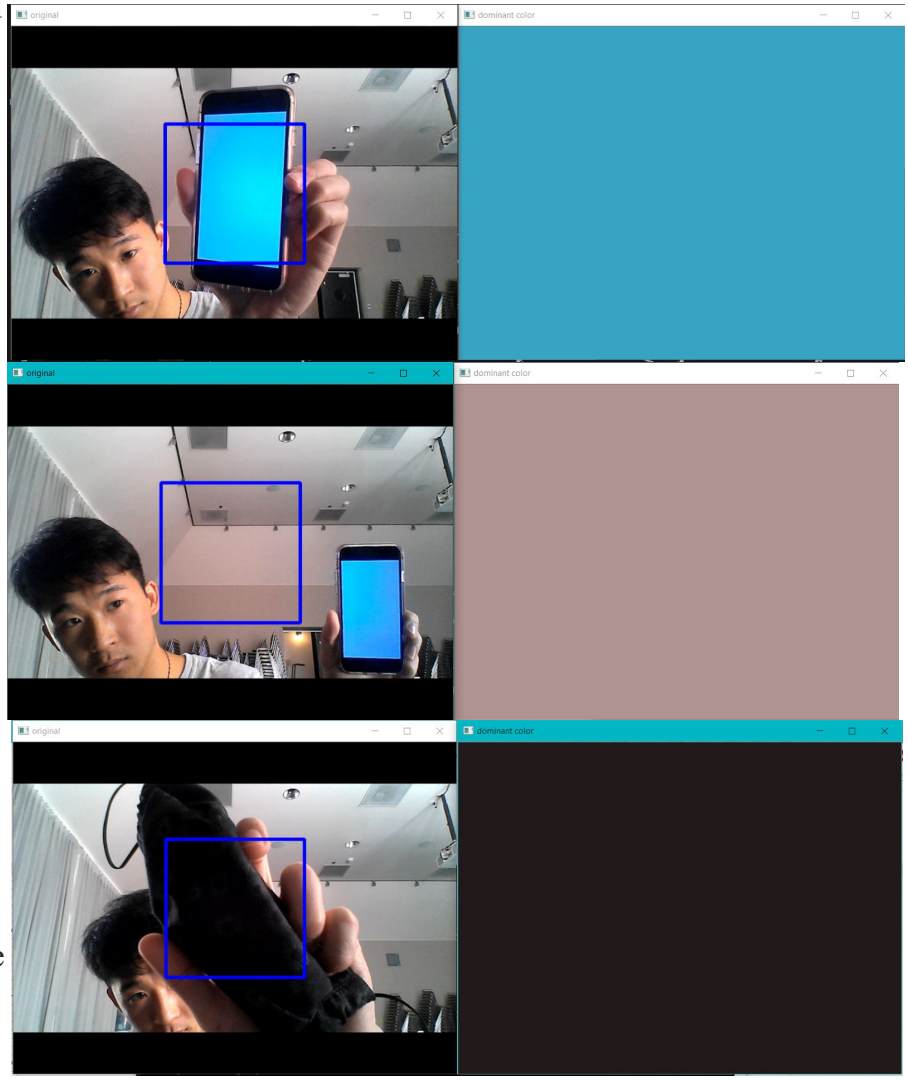
Bound box with HSV thresholding (Color Picker)



without ambient light | with ambient light | ambient light and full phone brightness

- Using HSV, created a bounding box across the blue hue
- compared its performance under different lighting conditions with a phone object and non phone object.
- For the non-phone object, the tracking was fairly good, since the external light conditions did not exceed the threshold range in the code.
- With a phone object with the blue hue at full brightness, the camera could not pick up the blue color as well, and the bounding box segmented into several smaller boxes The reasoning for this could be that the camera quality and brightness of the phone exceeded the thresholding range of the code.

Dominant Color



- Using the kmeans method of sklearn import, found the dominant color(RGB) within a target subimage of the video capture.
- Used a phone device to change the target area's brightness and color.
- The system was indeed robust to the changes and was still able to find the dominant color despite varying brightness and color change.
- Using a non-phone object under different brightness, the dominant color shifted accordingly to become less saturated.
- Phone and non-phone trials showed the dominant color system was equally robust to brightness changes in both.

2. Challenges and Todo Items

- When performing the dominant color video capture, the latency of the camera frames became very high due to the fit() function from sklearn.
- The video capture was not doing a good job at producing real-time results.
- potential improvement would be to substitute the fit() function for lower latency.
- For color masking using HSV and RGB, an additional improvement can be made by limiting the noise pick-up for the bounding boxes to more accurately track the target object.
- One way this can be done is setting a low bound of bound box dimensions so that small false-positives can be neglected.