

Engkanchanith Tan
005834717

Lab 1: Introduction to Software Dev and CV

Task 1:

Github repo: www.github.com/Kanchanith/180DA-WarmUp/

Kanchanith / 180DA-WarmUp Public

Code Issues Pull requests Actions Projects Wiki Security

main 1 branch 0 tags Go to file Add file Code

Engkanchanith Tan blah why is deleting so hard =.= d0f5f9d 14 minutes ago 13 commits

File/Commit	Description	Time
png code testing	moved all png files into one folder	21 minutes ago
README.md	first commit	last week
opencv.py	added source	27 minutes ago
opencv_cluster.py	added source	26 minutes ago
test.py	added test.py	last week
test.txt	added test.txt	last week

Task 2:

Kanchanith / 180DA-WarmUp Public

Code Issues Pull requests

main 180DA-WarmUp / test.txt

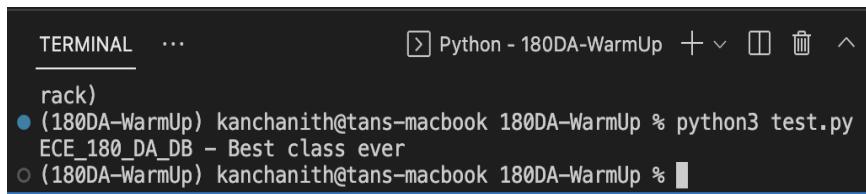
Engkanchanith Tan added test.txt

0 contributors

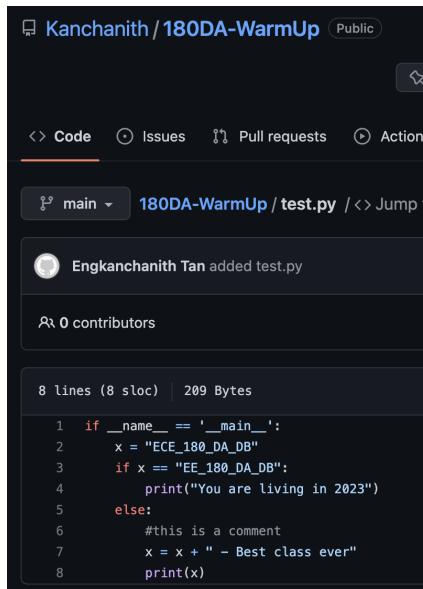
1 lines (1 sloc) | 15 Bytes

```
1 # 180DA-WarmUp
```

Task 3:



```
TERMINAL ... Python - 180DA-WarmUp + ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ 
rack)
● (180DA-WarmUp) kanchanith@tans-macbook 180DA-WarmUp % python3 test.py
ECE_180_DA_DB - Best class ever
○ (180DA-WarmUp) kanchanith@tans-macbook 180DA-WarmUp %
```



Kanchanith / **180DA-WarmUp** (Public)

Code Issues Pull requests Action

main 180DA-WarmUp / test.py / Jump

Engkanchanith Tan added test.py

0 contributors

8 lines (8 sloc) | 209 Bytes

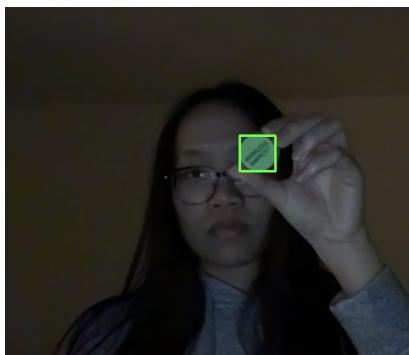
```
1 if __name__ == '__main__':
2     x = "ECE_180_DA_DB"
3     if x == "EE_180_DA_DB":
4         print("You are living in 2023")
5     else:
6         #this is a comment
7         x = x + " - Best class ever"
8     print(x)
```

Task 4:

1. HSV better. Hue is from 50 - 70. Saturation and Value are from 90 - 255. Green is between 128 - 255. Blue and Red are from 2 - 121.



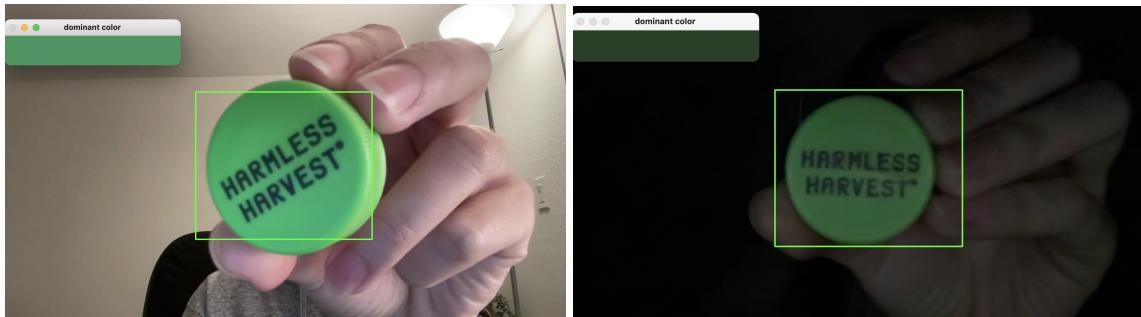
2. Turned off the lights: the ability to track the object went noticeably down: a lot of flickerings happening. It was harder to track the object. In the normal condition, the process was seamless: the box followed the object. RGB works better than HSV.



3. Only a small range of green was picked up by the camera. Some green shades does not work. The process works better when the brightness is low. When the brightness is high (almost full) the camera couldn't detect the color at all. Colors other than green does not work at all.

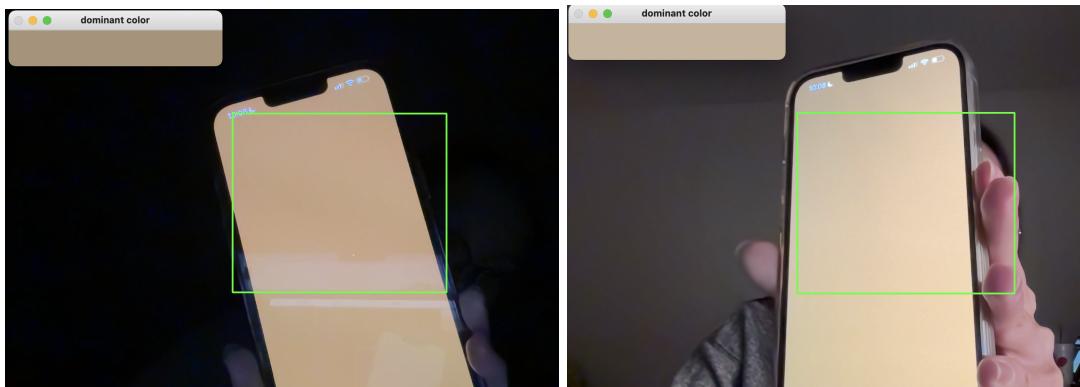


4. Non-phone object: normal vs low light



The dominant color changes its shade to mimic the shade that the camera is detecting.

Phone: low vs full brightness



We can make the same conclusion that the dominant color changes to match the color that the camera sees.

The robustness to brightness is the same for both objects.

Code

Task 4.1:

```
2  source:
3  OpenCV documentation:
4  - https://docs.opencv.org/3.0-beta/doc/py\_tutorials/py\_gui/py\_video\_display.html
5  - https://docs.opencv.org/4.x/df/d9d/tutorial\_py\_colorspaces.html
6  - https://docs.opencv.org/4.x/df/d9d/tutorial\_py\_colorspaces.html
7  - https://docs.opencv.org/4.x/d4/d73/tutorial\_py\_contours\_begin.html
8
9  ...
10
11 import numpy as np
12 import cv2 as cv
13
14 cap = cv.VideoCapture(0)
15
16 while(True):
17     # Capture frame-by-frame
18     ret, frame = cap.read()
19
20     # Convert frame color from BGR to HSV
21     hsv = cv.cvtColor(frame, cv.COLOR_BGR2HSV)
22
23     # Range of green
24     hsv_lower_green = np.array([50,90,90])
25     hsv_upper_green = np.array([70,255,255])
26
27     bgr_lower_green = np.array([2,128,2])
28     bgr_upper_green = np.array([121,255,121])
29
30     # Applying the threshold to isolate the object
31     mask = cv.inRange(frame, bgr_lower_green, bgr_upper_green)
32     # mask = cv.inRange(hsv, hsv_lower_green, hsv_upper_green)
33     contours, hierarchy = cv.findContours(mask, cv.RETR_TREE, cv.CHAIN_APPROX_SIMPLE)
34
35     if (len(contours) > 0):
36         c = max(contours, key = cv.contourArea)
37         x,y,w,h = cv.boundingRect(c)
38         cont_img = cv.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),5)
39         # cont_img = cv.drawContours(frame, contours, -1, 255, 3)
40
41     # Display the resulting frame
42     cv.imshow('frame',frame)
43     if cv.waitKey(1) & 0xFF == ord('q'):
44         break
45
46 # When everything done, release the capture
47 cap.release()
48 cv.destroyAllWindows()
```

Task 4.4:

```
2  source:
3  - https://code.likeagirl.io/finding-dominant-colour-on-an-image-b4e075f98
4  - https://docs.opencv.org/4.x/dc/d45/tutorial\_py\_drawing\_functions.html
5  - https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
6  - https://docs.opencv.org/3.4/d3/d2f/tutorial\_py\_basic\_ops.html
7
8  ...
9  import numpy as np
10 import cv2 as cv
11 import matplotlib.pyplot as plt
12 from sklearn.cluster import KMeans
13
14 cap = cv.VideoCapture(0)
15
16 # a matrix of RGB
17 bar = np.zeros((50, 300, 3), dtype="uint8")
18
19 while(True):
20     # Capture frame-by-frame
21     ret, frame = cap.read()
22
23     # Convert color from BGR to RGB
24     RGB_frame = cv.cvtColor(frame, cv.COLOR_BGR2RGB)
25
26     # Draw a rectangle and apply Image ROI
27     focus = frame[300:800, 650:1250]
28     rect = cv.rectangle(frame,(650,300),(1250,800),(0,255,0),3)
29
30     # 3D to 2D
31     focus = focus.reshape((focus.shape[0] * focus.shape[1],3))
32
33     # Apply kmeans
34     clt = KMeans(n_clusters=1) #cluster number
35     clt.fit(focus)
36
37     # Transfer RGB value of dominant color to list
38     bar[:] = clt.cluster_centers_[0]
39
40     # Display the resulting frame
41     cv.imshow('frame', frame)
42     cv.imshow('dominant color',bar)
43     if cv.waitKey(1) & 0xFF == ord('q'):
44         break
45
46 # When everything done, release the capture
47 cap.release()
48 cv.destroyAllWindows()
```