آماده سازی نوت بوک

> 1. قبل از هر کاری، با استفاده از مسیر نوشته شده در خط پایین، یک کپی از این نوتبوک در گوگل درایو خودتان بسازید و تمرین را در آن نسخه حل کنید.

File --> Save a copy in Drive

> 2. برای دسترسی به تصاویر مورد نیاز در این تکلیف، بدون اعمال هیچ تغییری در بلوک زیر، آن را اجرا کنید. با این کار فایل های مربوط به تکلیف (تصاویر) دانلود و در فولدر کولب شما قرار داده می شوند. انجام این مرحله پس از هر بار قطع شدن از کولب و اتصال دوباره، ضروری است. یعنی اگر مثلا یک سوال را حل کردید و بعد کولب را بستید یا مدتی با آن کار نکردید و اتصالتان به طور خودکار قطع شد، در اقدام بعدیتان برای نوشتن بقیه‌ی تمرین، حتما این بلوک باید دوباره اجرا شود.

RUN THIS BLOCK WITHOUT ANY CHANGE to download the data

Show code

Imports

فراخوانی کتابخانه ها

[ ] ↳ *1 cell hidden*

Any Helper Functions

در صورت نیاز یا برای راحتی خودتان می‌توانید توابع کمکی (مثلا برای عملیات‌های پر تکرار) این جا تعریف کنید (همه در همین بلوک).

[ ] ↳ *1 cell hidden*

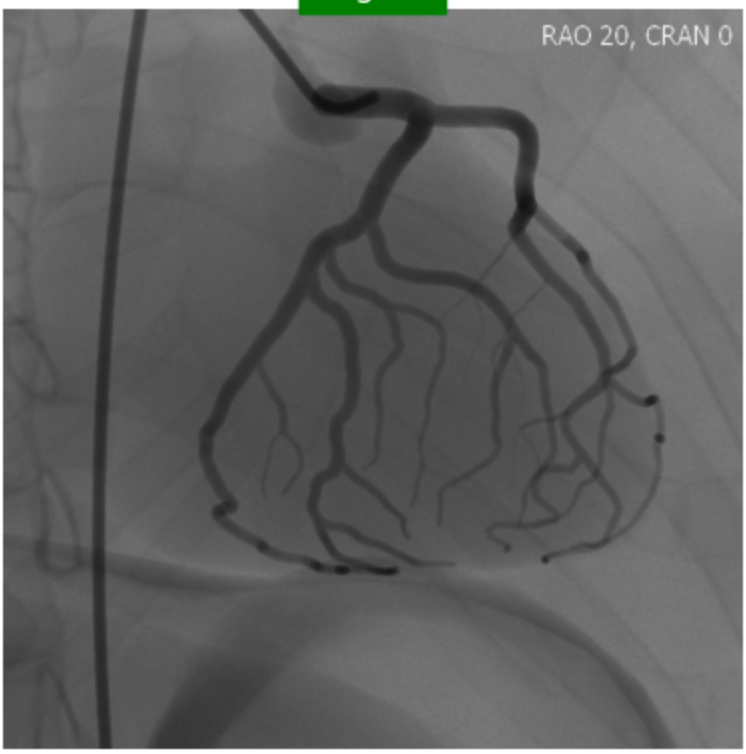
Question 5: Edge Detection (15%)

Q5 - Part 1 (Sobel) (5%)

Show code

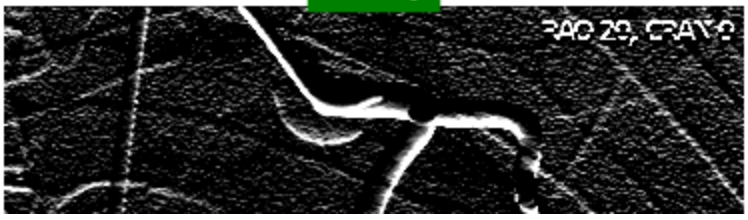
```
(-0.5, 719.5, 719.5, -0.5)
```

original



sobel_x

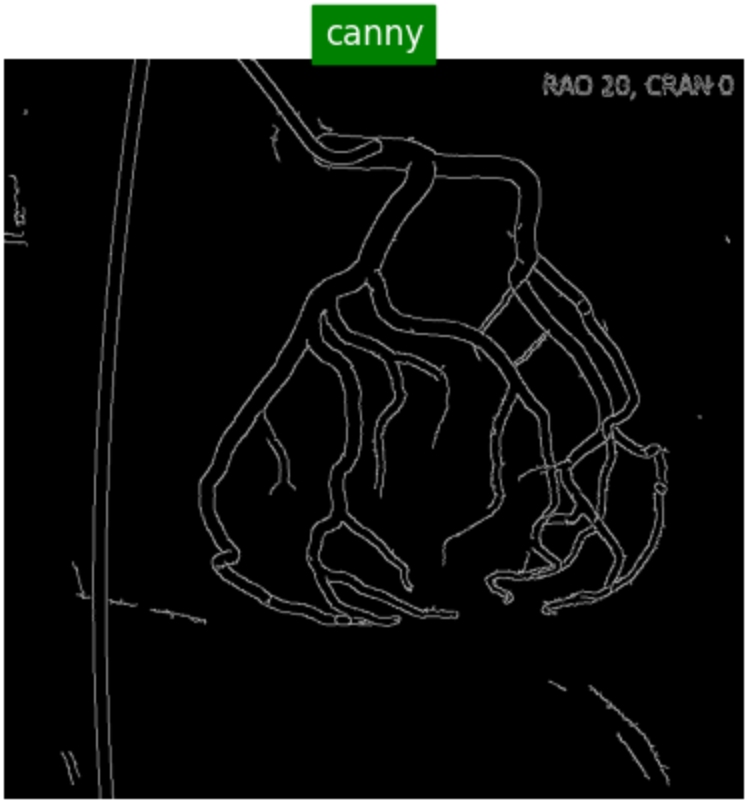
sobely



Q5 - Part 2 (Canny) (5%)

Show code

```
(-0.5, 719.5, 719.5, -0.5)
```



Q5 - Part 3 (LoG) (5%)

Show code

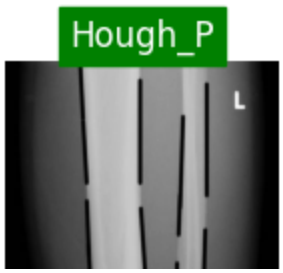
```
(-0.5, 719.5, 719.5, -0.5)
```

Question 6: Hough Transform (15%)

Q6 - Part 1 (60%)

Show code

```
(-0.5, 369.5, 999.5, -0.5)
```



Q6 - Part 2 (40%)

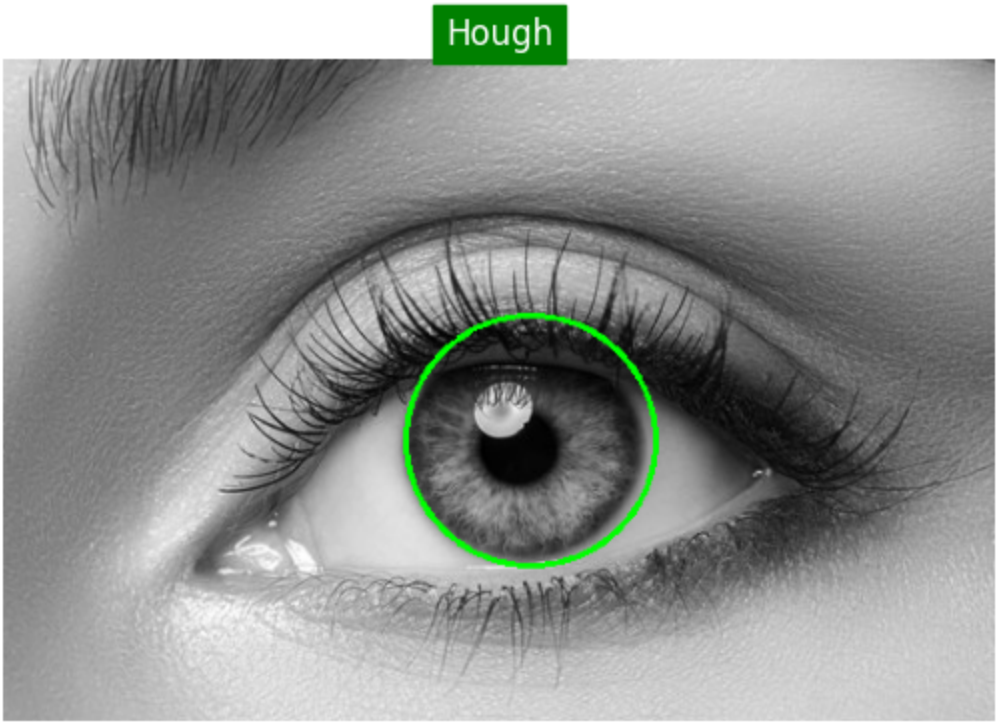
```
#@title Q6 - Part 2 (40%)
###

# ENTER YOUR CODE HERE.
image = cv2.imread('eye.jpg', cv2.IMREAD_GRAYSCALE)
img = cv2.medianBlur(image,5)
cimage = cv2.cvtColor(image,cv2.COLOR_GRAY2BGR)
plt.figure()
plt.imshow(image, cmap="gray", vmin=0, vmax=255)
plt.title('original', color='white', backgroundcolor='green')
plt.axis('off')

circles = cv2.HoughCircles(img,cv2.HOUGH_GRADIENT,2,15,param1=220,param2=120,minRadius=0,max

circles = numpy.uint16(numpy.around(circles))
print(circles)
for i in circles[0,:]:
    cv2.circle(cimage,(i[0],i[1]),i[2],(0,255,0),2)
plt.figure()
plt.imshow(cimage, cmap="gray", vmin=0, vmax=255)
plt.title('Hough', color='white', backgroundcolor='green')
plt.axis('off')
###
```

```
[[[287 207  68]]]
(-0.5, 539.5, 359.5, -0.5)
```



Question 7: Feature-Based Registration (20%)

```
###

# ENTER YOUR CODE HERE.
x11 = 90
y11 = 112
x12 = 265
y12 = 213

x21 = 115
y21 = 81
x22 = 340
y22 = 194

x31 = 74
y31 = 137
x32 = 214
y32 = 230

A = numpy.array([[x11, y11, 1], [x21, y21, 1], [x31, y31, 1]])
b = numpy.array([[x12,y12], [x22, y22], [x32, y32]])

x = numpy.matmul(numpy.linalg.inv(A),b)

print("a = ", x)
###
```

```
(217, 181)
a =  [[  2.27906977   0.40310078]
 [ -0.58139535   0.9379845 ]
 [125.          71.66666667]]
```

Question 8: Similarity-based Segmentation (20%)

Q8 - Part 1

```
#@title Q8 - Part 1
###

# ENTER YOUR CODE HERE.
image = cv2.imread('Color_MRI.png')
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
###
```

Q8 - Part 2

```
#@title Q8 - Part 2
###

# ENTER YOUR CODE HERE.
x11 = 370
y11 = 400
x12 = 247
y12 = 271
###
```

Q8 - Part 3 (5%)

```
#@title Q8 - Part 3 (5%)
###

# ENTER YOUR CODE HERE.
new_image = numpy.zeros_like(image)
new_image[x11, y11] = image[x11, y11]
new_image[x12, y12] = image[x12, y12]
###
```

Q8 - Part 4 (50%)

```
#@title Q8 - Part 4 (50%)
###
# ENTER YOUR CODE HERE.
def growing_region(image, d, type_t, seed, max_iterations=100):
    result = numpy.zeros_like(image, dtype=numpy.uint8)
    x = image[seed]
    result[seed] = x

    for _ in range(max_iterations):
        if type_t == 'static':
            threshold = x
        elif type_t == 'variable':
            threshold = numpy.mean(image[result == x])
            threshold = numpy.array([threshold,threshold,threshold])


        mask = cv2.inRange(image, threshold - d, threshold + d)
        result = cv2.bitwise_and(result, result, mask=~mask)
        result[mask > 0] = x

    return result

###
```

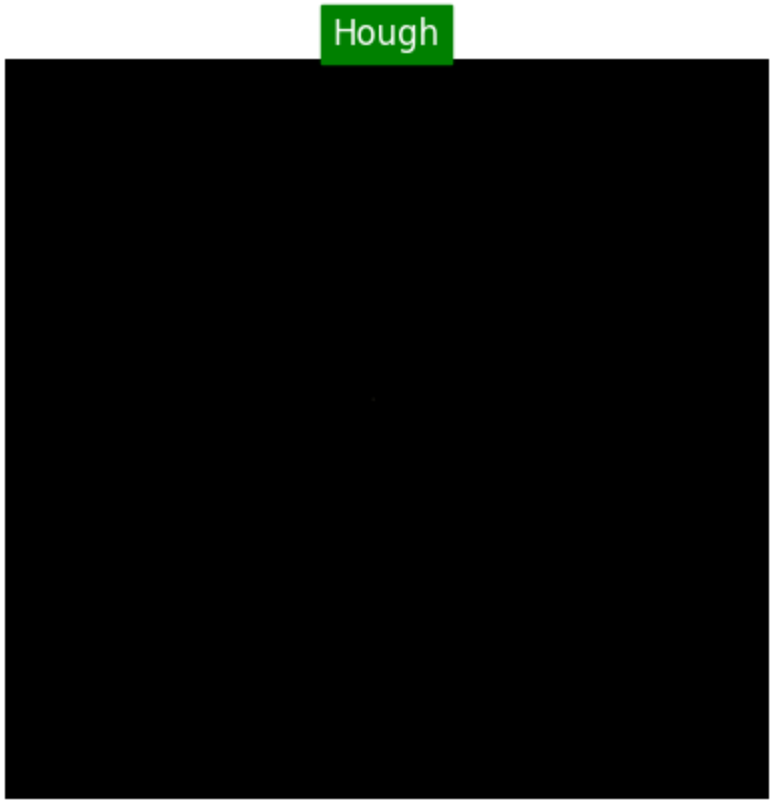
Q8 - Part 5 (15%)

```
#@title Q8 - Part 5 (15%)
###

# ENTER YOUR CODE HERE.

plt.figure()
plt.imshow(growing_region(new_image, 10, "static", (x11, y11), 200), cmap="gray", vmin=0, vm
plt.title('Hough', color='white', backgroundcolor='green')
plt.axis('off')
###
```

```
(-0.5, 830.5, 805.5, -0.5)
```



Q8 - Part 6 (10%)

```
#@title Q8 - Part 6 (10%)
###
image1 = cv2.imread('Color_MRI.png')
image1 = cv2.cvtColor(image1, cv2.COLOR_BGR2RGB)

image2 = cv2.imread('Color_MRI.png')
image2 = cv2.cvtColor(image2, cv2.COLOR_BGR2RGB)

image3 = cv2.imread('Color_MRI.png')
image3 = cv2.cvtColor(image3, cv2.COLOR_BGR2RGB)
```

Q8 - Part 7 (5%)

```
#@title Q8 - Part 7 (5%)
###

# ENTER YOUR CODE HERE.
print(200)
###
```

```
200
```

Q8 - Part 8 (15%)

```
#@title Q8 - Part 8 (15%)
###

# ENTER YOUR CODE HERE.
plt.figure()
plt.imshow(static1, cmap="gray", vmin=0, vmax=255)
plt.title('static1', color='white', backgroundcolor='green')
plt.axis('off')

plt.figure()
plt.imshow(static2, cmap="gray", vmin=0, vmax=255)
plt.title('static2', color='white', backgroundcolor='green')
plt.axis('off')
plt.figure()

plt.imshow(variable1, cmap="gray", vmin=0, vmax=255)
plt.title('variable1', color='white', backgroundcolor='green')
plt.axis('off')
plt.figure()
```