영상정보처리 12주차 과제 템플리트

이름: 김경민
학번:32200327

입력 이미지: 자유

구글 드라이브 마우팅 및 작업 경로로 이동

- 다음 쉘에 필요한 작업을 하시오.

```
from google.colab import drive
drive.mount('/gdrive')
%cd /gdrive/MyW Drive/Classroom/[22-1 영상정보처리] 2000004793-2022-1/ImageProcClass/Notebook-week1
!pwd
```

```
Mounted at /gdrive
/gdrive/My Drive/Classroom/[22-1 영상정보처리] 2000004793-2022-1/ImageProcClass/Notebook-week
/gdrive/My Drive/Classroom/[22-1 영상정보처리] 2000004793-2022-1/ImageProcClass/Notebook-week
```

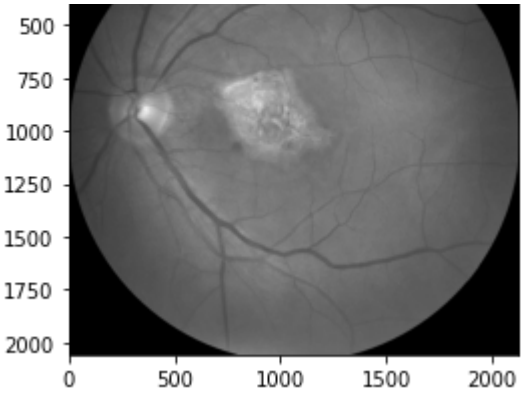
```
import cv2
from google.colab.patches import cv2_imshow
import numpy as np
import matplotlib.pyplot as plt

image_path = '../Dongkeun-OpenCV-ImgData/BAI_27_A0027.jpg' # Baidu에서 수집한 안저사진
image_path2 = '../Dongkeun-OpenCV-ImgData/srcThreshold.png'


# matplot color display
def show_with_matplotlib_jh(img, title):
    if img is None:
      print("show_with_matplotlib: Could not read the image.")
      return

    if img.shape[2] != 3:
      print()
      print("show_with_matplotlib: given image does not contains 3 channels")
      return

    # Convert BGR image to RGB:
    img_RGB = img[:, :, ::-1]

    # Show the image using matplotlib:
    plt.imshow(img_RGB)
    plt.title(title)
    plt.show()

# matplot grayscale display
```

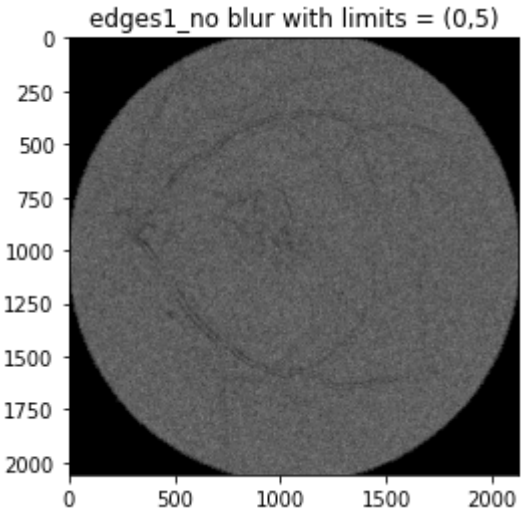
```
# matplot grayscale display
def show_with_matplotlib_gray_jh(img, title):
    if img is None:
        print("show_with_matplotlib_gray: Could not read the image.")
        return

    if img.ndim > 2:
        print()
        print("show_with_matplotlib: given image has more than 2 dim")
        return

    plt.imshow(img, cmap="gray")
    plt.title(title)
    plt.show()
```

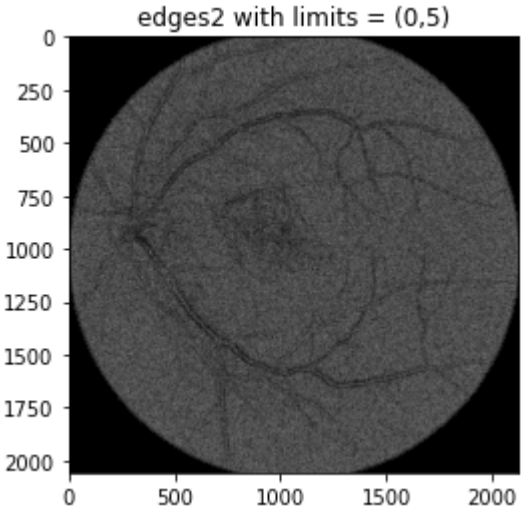
문제 1

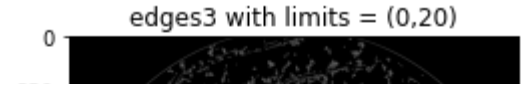
입력 이미지는 자유롭게 선택을 하여, Canny Algorithm 의 패러미터에 변경에 따른 결과를 보이고, 간단하게 이해한 바를 정리하시오.

```
src = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
show_with_matplotlib_gray_jh(src, "src")

# 노이즈 제거
blur = cv2.GaussianBlur(src,(5,5),0) # 5x5 가우시안 필터로 노이즈 제거

# Canny 알고리즘으로 edge 찾기
edges1 = cv2.Canny(src, 0,5)
edges2 = cv2.Canny(blur, 0,5) # 선명하고 가는 edge # threshold 범위 좁을수록 섬세한 edge 잡을 수 있
edges3 = cv2.Canny(blur, 0, 20) # 범위 넓어지면 threshold의 방향성 차이가 안 나기 때문에 노이즈나 잘

show_with_matplotlib_gray_jh(edges1, "edges1_no blur with limits = (0,5)")
show_with_matplotlib_gray_jh(edges2, "edges2 with limits = (0,5)")
show_with_matplotlib_gray_jh(edges3, "edges3 with limits = (0,20)") # 섬세한 선들은 잡아내지 못하고
```







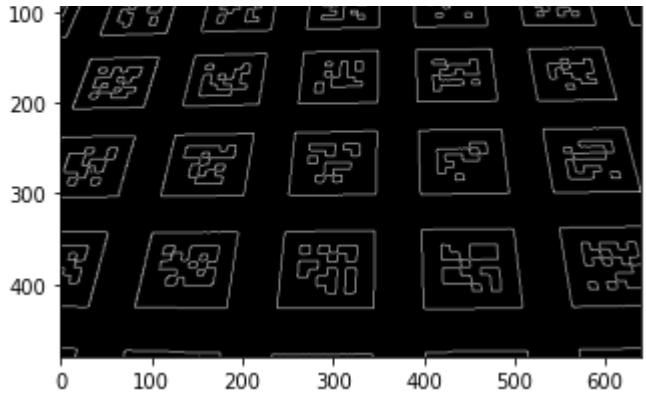


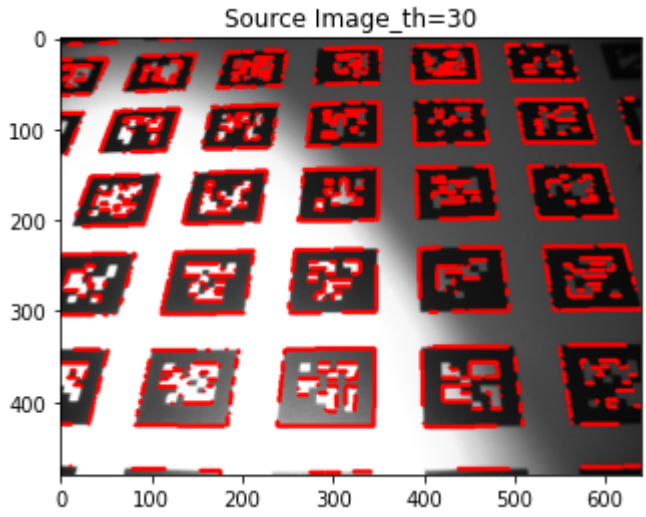
문제 2

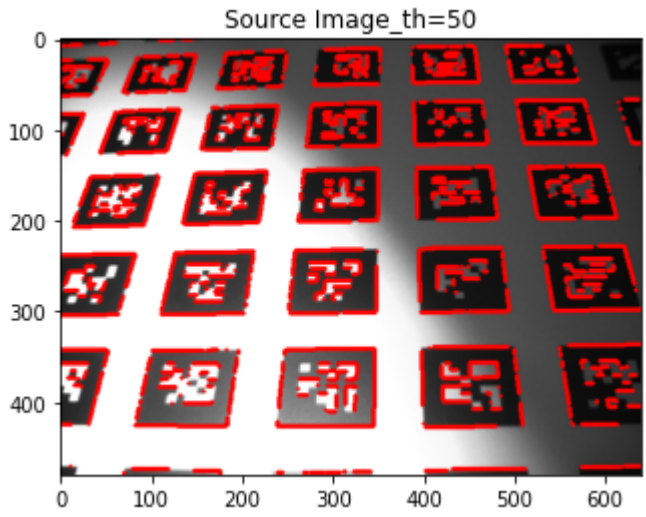
입력 이미지는 자유롭게 선택을 하여, 통계적 Hough Transform 에 사용되는 패러미터 변경에 따른 결과를 보이고, 간단하게 이해한 바를 정리하시오.

```
src = cv2.imread(image_path2)
gray = cv2.cvtColor(src,cv2.COLOR_BGR2GRAY)

def draw_line(lines,text):
  for line in lines:
    x1, y1, x2, y2   = line[0]
    cv2.line(src, (x1,y1), (x2,y2), (0,0,255), 3) # draw a red line on the source image
```

```
  show_with_matplotlib_jh(src, text)

#1
edges = cv2.Canny(gray, 50, 100)
show_with_matplotlib_gray_jh(edges, "Edges by Canny")

#2
lines = cv2.HoughLinesP(edges, rho=1, theta=np.pi/180.0, threshold=30) # 숫자가 작을수록 검출되는 선
#print(lines)
draw_line(lines,"Source Image_th=30") # 자잘한 선들이 띄엄띄엄 많음

lines = cv2.HoughLinesP(edges, rho=1, theta=np.pi/180.0, threshold=50)
draw_line(lines,"Source Image_th=50")

lines = cv2.HoughLinesP(edges, rho=1, theta=np.pi/180.0, threshold=100)
draw_line(lines,"Source Image_th=100") # 자잘한 선은 없어지고 이어지는 선이 많음 # missing point 를
```







✓ 2초 오전 1:15에 완료됨

reCAPTCHA 서비스에 연결할 수 없습니다. 인터넷 연결을 확인한 후 페이지를 새로고침하여 reCAPTCHA 보안문자를 다시 로드하세요.