***Programming Environment***

*Python - 3.11 and Opencv and Matplotlib Library*

*GNU Octave - 8.4.0*

***Output***

*-Octave Output*

metin, ekran görüntüsü, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

*-Python Output*

gökyüzü, deniz feneri, bina, dış mekan içeren bir resim

Açıklama otomatik olarak oluşturuldu deniz feneri, bina, gökyüzü, bulut içeren bir resim

Açıklama otomatik olarak oluşturuldu

bina, gökyüzü, deniz feneri, dış mekan içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

***Reflections***

*Image processing codes are similar in opencv and octave. Drawing histogram in octave is easier than in python.*

***Source Code***

*-Octave Codes*

*% Read Image*

*img = imread('images/lighthouse.png');*

*% Show Image*

*figure;*

*subplot(2, 2, 1);*

*imshow(img);*

*title('RGB Image');*

*% Convert to gray scale image*

*gray = rgb2gray(img);*

*% Show grayscale image*

*subplot(2, 2, 2);*

*imshow(gray);*

*title('Grayscale Image');*

*% Rotate 45 degree the image*

*pkg load image;*

*rotated\_img = imrotate(img, 45);*

*% Show rotated image*

*subplot(2, 2, 3);*

*imshow(rotated\_img);*

*title('Rotated Image');*

*% Draw the grayscale images's histogram*

*subplot(2, 2, 4);*

*imhist(gray);*

*title('Grayscale Image Histogram');*

*xlabel('Pixel Intensity');*

*ylabel('Number Of Pixel');*

*-Python Codes*

from matplotlib import pyplot as plt  
import cv2 as cv  
  
#Read Image  
img = cv.imread('images/lighthouse.png')  
cv.namedWindow('colored',cv.WINDOW\_AUTOSIZE)  
cv.imshow('colored',img)  
cv.waitKey(0)  
  
#Convert to gray scale image  
gray = cv.cvtColor(img,cv.COLOR\_BGR2GRAY)  
#Show grayscale image  
cv.imshow('gray',gray)  
cv.waitKey(0)  
  
#Rotate 45 degree the image  
rot\_mat = cv.getRotationMatrix2D((img.shape[1] / 2, img.shape[0] / 2), 45, 1)  
rotated\_img = cv.warpAffine(img, rot\_mat, (img.shape[1], img.shape[0]), flags=cv.INTER\_CUBIC)  
  
#Show rotated image  
cv.imshow('rotated', rotated\_img)  
cv.waitKey(0)  
  
#Draw the grayscale images's histogram  
plt.figure()  
plt.hist(gray)  
plt.xlabel("Pixel Intensity")  
plt.ylabel("Number of Pixels")  
plt.title("Histogram of Grayscale Image")  
plt.show()