GREY SCALE CODE

import cv2

#img=cv2.imread('./data/png.png')

img=cv2.imread('./data/png.png',cv2.IMREAD\_GRAYSCALE)

cv2.imshow('Image',img)

cv2.waitKey(0)

cv2.destroyALLWindows()

import matpotlib.pyplot as plt

plt.imshow(img)

plt.show()

EDGE DETECTION AND GREY SCALE CODE

import numpy as np

import cv2 as cv

from matplotlib import pyplot as plt

img = cv.imread('./data/png1.png', cv.IMREAD\_GRAYSCALE)

assert img is not None, "file could not be read, check with os.path.exists()"

edges = cv.Canny(img,100,200)

plt.subplot(121),plt.imshow(img,cmap = 'gray')

plt.title('Original Image'), plt.xticks([]), plt.yticks([])

plt.subplot(122),plt.imshow(edges,cmap = 'gray')

plt.title('Edge Image'), plt.xticks([]), plt.yticks([])

plt.show()

IMAGE RESIZING

import cv2

import matplotlib.pyplot as plt

img = cv2.imread('./data/png.png')

h, w, c = img.shape

print(f"Height and width of original image: {h}, {w}" )

new\_size = (450, 340) # new\_size=(width, height)

print(f"New height and width: {new\_size[1]}, {new\_size[0]}" )

resize\_img = cv2.resize(img, new\_size)

img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

resize\_img = cv2.cvtColor(resize\_img, cv2.COLOR\_BGR2RGB)

plt.subplot(121),plt.imshow(img), plt.title("Original Image")

plt.subplot(122), plt.imshow(resize\_img), plt.title("Resized Image")

plt.show()

GAUSIAN BLUR

import cv2

img = cv2.imread('./data/png.png')

avging = cv2.blur(img,(10,10))

cv2.imshow('Averaging',avging)

cv2.waitKey(0)

gausBlur = cv2.GaussianBlur(img, (5,5),0)

cv2.imshow('Gaussian Blurring', gausBlur)

cv2.waitKey(0)

medBlur = cv2.medianBlur(img,5)

cv2.imshow('Media Blurring', medBlur)

cv2.waitKey(0)

bilFilter = cv2.bilateralFilter(img,9,75,75)

cv2.imshow('Bilateral Filtering', bilFilter)

cv2.waitKey(0)

cv2.destroyAllWindows()