Lab TASK 5

Introduction to OpenCV

we will cover the most basic and important concepts of OpenCV intuitively:

1. Reading an image
2. Extracting the RGB values of a pixel
3. Extracting the Region of Interest (ROI)
4. Resizing the Image
5. Rotating the Image
6. Drawing a Rectangle
7. Displaying text

# Importing the OpenCV library

import cv2

# Reading the image using imread() function

image = cv2.imread('image.jpg')

# Extracting the height and width of an image

h, w = image.shape[:2]

# Displaying the height and width

print("Height = {}, Width = {}".format(h, w))

# Extracting RGB values.

# Here we have randomly chosen a pixel

# by passing in 100, 100 for height and width.

(B, G, R) = image[100, 100]

# Displaying the pixel values

print("R = {}, G = {}, B = {}".format(R, G, B))

# We can also pass the channel to extract

# the value for a specific channel

B = image[100, 100, 0]

print("B = {}".format(B))

# We will calculate the region of interest

# by slicing the pixels of the image

roi = image[100 : 500, 200 : 700]

cv2.imshow("ROI", roi)

cv2.waitKey(0)

# resize() function takes 2 parameters,

# the image and the dimensions

resize = cv2.resize(image, (500, 500))

cv2.imshow("Resized Image", resize)

cv2.waitKey(0)

# Calculating the ratio

ratio = 800 / w

# Creating a tuple containing width and height

dim = (800, int(h \* ratio))

# Resizing the image

resize\_aspect = cv2.resize(image, dim)

cv2.imshow("Resized Image", resize\_aspect)

cv2.waitKey(0)

# We are copying the original image,

# as it is an in-place operation.

output = image.copy()

# Using the rectangle() function to create a rectangle.

rectangle = cv2.rectangle(output, (1500, 900),

                        (600, 400), (255, 0, 0), 2)

# Copying the original image

output = image.copy()

# Adding the text using putText() function

text = cv2.putText(output, 'OpenCV Demo', (500, 550),

                cv2.FONT\_HERSHEY\_SIMPLEX, 4, (255, 0, 0), 2)

# Python code to read image

import cv2

# To read image from disk, we use

# cv2.imread function, in below method,

img = cv2.imread("geeksforgeeks.png", cv2.IMREAD\_COLOR)

# Creating GUI window to display an image on screen

# first Parameter is windows title (should be in string format)

# Second Parameter is image array

cv2.imshow("image", img)

# To hold the window on screen, we use cv2.waitKey method

# Once it detected the close input, it will release the control

# To the next line

# First Parameter is for holding screen for specified milliseconds

# It should be positive integer. If 0 pass an parameter, then it will

# hold the screen until user close it.

cv2.waitKey(0)

# It is for removing/deleting created GUI window from screen

# and memory

cv2.destroyAllWindows()

#import cv2, numpy and matplotlib libraries

import cv2

import numpy as np

import matplotlib.pyplot as plt

img=cv2.imread("geeks.png")

#Displaying image using plt.imshow() method

plt.imshow(img)

#hold the window

plt.waitforbuttonpress()

plt.close('all')

#import cv2, numpy and matplotlib libraries

import cv2

import numpy as np

import matplotlib.pyplot as plt

img=cv2.imread("geeks.png")

# Converting BGR color to RGB color format

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

#Displaying image using plt.imshow() method

plt.imshow(RGB\_img)

# hold the window

plt.waitforbuttonpress()

plt.close('all')

# Python program to explain cv2.imread() method

# importing cv2

import cv2

# path

path = r'geeksforgeeks.png'

# Using cv2.imread() method

# Using 0 to read image in grayscale mode

img = cv2.imread(path, cv2.IMREAD\_GRAYSCALE)

# Displaying the image

cv2.imshow('image', img)

cv2.waitKey(0)

cv2.destroyAllWindows()