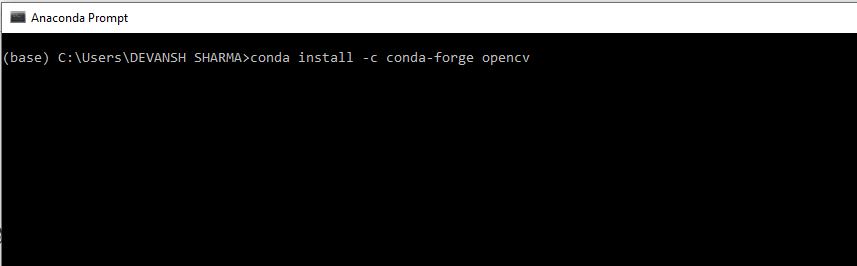
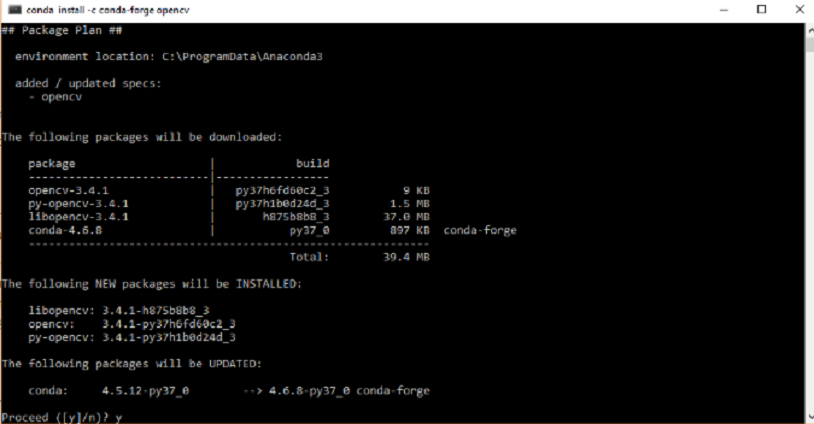
COMPUTER VISION

Installation of the OpenCV

### **Install OpenCV using Anaconda**

1. The first step is to download the latest Anaconda graphic installer for Windows from it [official site](https://www.anaconda.com/distribution/). Choose your bit graphical installer. You are suggested to install 3.7 working with Python 3.
2. Choose the graphical bit installer.
3. After installing it, open the Anaconda prompt and type the following command.



1. Press the Enter button and it will download all the related OpenCV configuration.

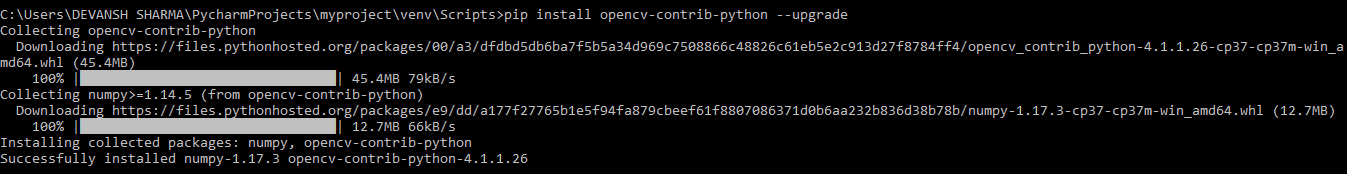
### **Install OpenCV in the Windows via pip**

1. OpenCV is a Python library so it is necessary to install Python in the system and install OpenCV using pip command:

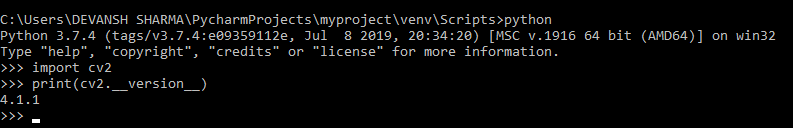
pip install opencv-contrib-python --upgrade

1. We can install it without extra modules by the following command:

pip install opencv-python



1. Open the command prompt and type the following code to check if the OpenCV is installed or not.



# OpenCV Read and Save Image

# import cv2

# import matplotlib.pyplot as plt

# kernel = np.ones((5, 5), np.uint8)

# print(kernel)

# path = "E:\COMPUTER VISION\IMAGE FOLDER\CAT.jpg"

# img = cv2.imread(path)

# imgGray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

# plt.imshow(imgGray, cmap='gray')

# plt.axis('off') # Optional: To hide the axis ticks and labels

# plt.show()

# output:

# 

# OpenCV Basic Operation on Images

## **Accessing and Modifying pixel values**

**import** numpy as np

**import** cv2

img = cv2.imread("E:\COMPUTER VISION\IMAGE FOLDER\CAT.jpg ",1)

pixel = img[100,100]

print(pixel)

**Output:**

[190 166 250]

## **Accessing Image Properties**

**import** cv2

# read image

img = cv2.imread(r' E:\COMPUTER VISION\IMAGE FOLDER\CAT.jpg,1)

# height, width, number of channels in image

height = img.shape[0]

width = img.shape[1]

channels = img.shape[2]

size1 = img.size

print('Image Dimension    : ',dimensions)

print('Image Height       : ',height)

print('Image Width        : ',width)

print('Number of Channels : ',channels)

print('Image Size  :', size1)

**Output:**

Image Dimension : (4, 1, 3)

Image Height : 4

Image Width : 1

Number of Channels : 3

Image Size : 12

## **Image ROI (Region of Interest)**

## **Splitting and Merging Image channels**

b,g,r = cv2.split(img)

img = cv2.merge((b,g,r))

## **Making Borders for Images**

**import** cv2 as cv

**import** numpy as np

from matplotlib **import** pyplot as plt

BLUE = [255,0,0]

img1 = cv.imread(r' E:\COMPUTER VISION\IMAGE FOLDER\CAT.jpg ',1)

replicate = cv.copyMakeBorder(img1,10,10,10,10,cv.BORDER\_REPLICATE)

reflect = cv.copyMakeBorder(img1,10,10,10,10,cv.BORDER\_REFLECT)

reflect101 = cv.copyMakeBorder(img1,10,10,10,10,cv.BORDER\_REFLECT\_101)

wrap = cv.copyMakeBorder(img1,10,10,10,10,cv.BORDER\_WRAP)

constant= cv.copyMakeBorder(img1,10,10,10,10,cv.BORDER\_CONSTANT,value=BLUE)

plt.subplot(231),plt.imshow(img1,'gray'),plt.title('ORIGINAL')

plt.subplot(232),plt.imshow(replicate,'gray'),plt.title('REPLICATE')

plt.subplot(233),plt.imshow(reflect,'gray'),plt.title('REFLECT')

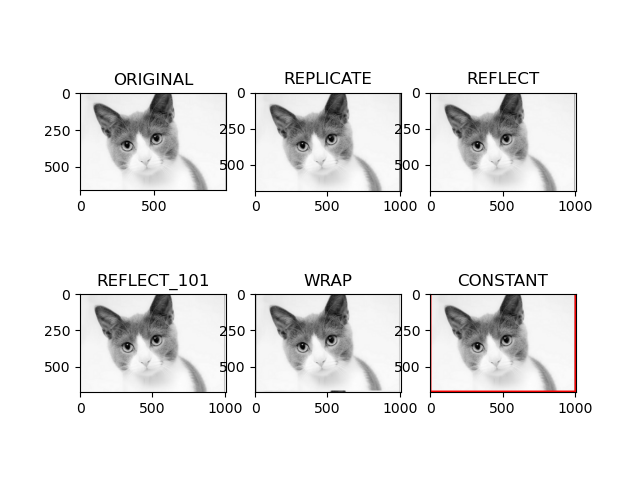
plt.subplot(234),plt.imshow(reflect101,'gray'),plt.title('REFLECT\_101')

plt.subplot(235),plt.imshow(wrap,'gray'),plt.title('WRAP')

plt.subplot(236),plt.imshow(constant,'gray'),plt.title('CONSTANT')

plt.show()

**Output:**



Read Image and Convert to Grayscale using Python

import cv2

import numpy as np

kernel = np.ones((5,5),np.uint8)

print(kernel)

path = "E:/COMPUTER VISION/IMAGE FOLDER/flower.jpg”

img =cv2.imread(path)

imgGray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

cv2.imshow("GrayScale",imgGray)

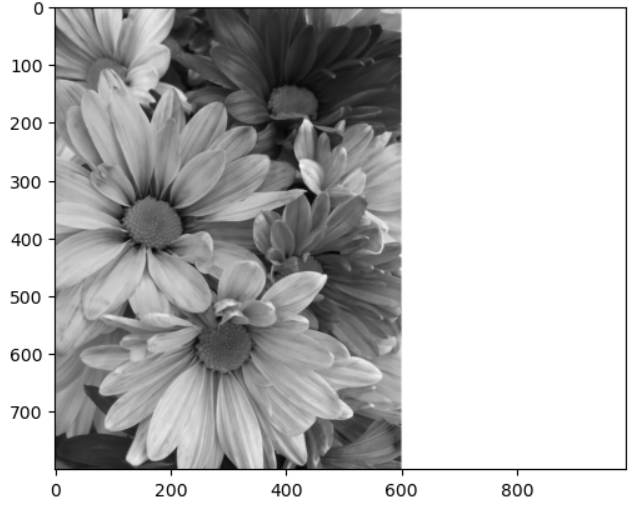
import matplotlib.pyplot as plt

plt.imshow(imgGray, cmap='gray')

plt.axis('off')

plt.show()

OUTPUT:



Basic Operations to Convert image to show outline Canny function in Python

import cv2

import matplotlib.pyplot as plt

import numpy as np

kernel = np.ones((5,5),np.uint8) print(kernel)

path = "C:/Users/divya/OneDrive/Documents/COMPUTER VISION/CANNY.jpg"

img =cv2.imread(path)

imgGray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

imgBlur = cv2.GaussianBlur(imgGray,(7,7),0)

imgCanny = cv2.Canny(imgBlur,100,200)

# Display the image using plt.imshow

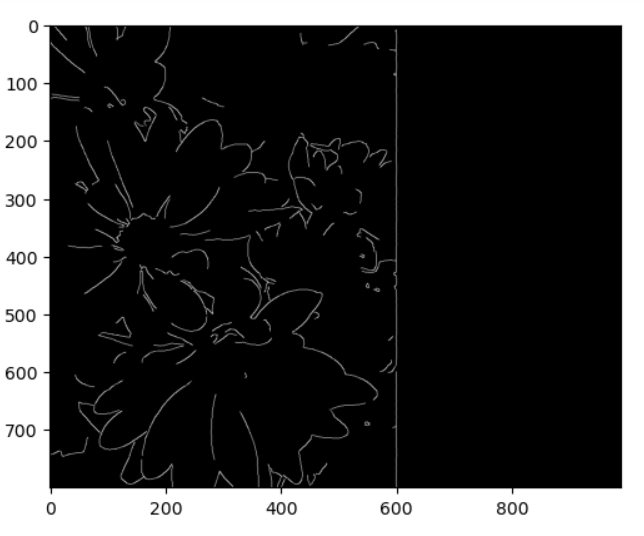
plt.imshow(imgCanny, cmap='gray')

plt.show()

cv2.imshow("Img Canny",imgCanny)

cv2.waitKey(0)

OUTPUT:



To Perform Basic Operations to Read Image and Convert to Blur using GaussianBlur.

import cv2

import matplotlib.pyplot as plt

import numpy as np

kernel = np.ones((5,5),np.uint8)

print(kernel)

path = "E:/COMPUTER VISION/IMAGE FOLDER/flower.jpg"

img =cv2.imread(path)

imgGray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

imgBlur = cv2.GaussianBlur(imgGray,(7,7),0)

plt.imshow(imgBlur, cmap='gray')

plt.axis('off')

plt.show()

cv2.imshow("Img Blur",imgBlur)

cv2.waitKey(0)

OUTPUT:

