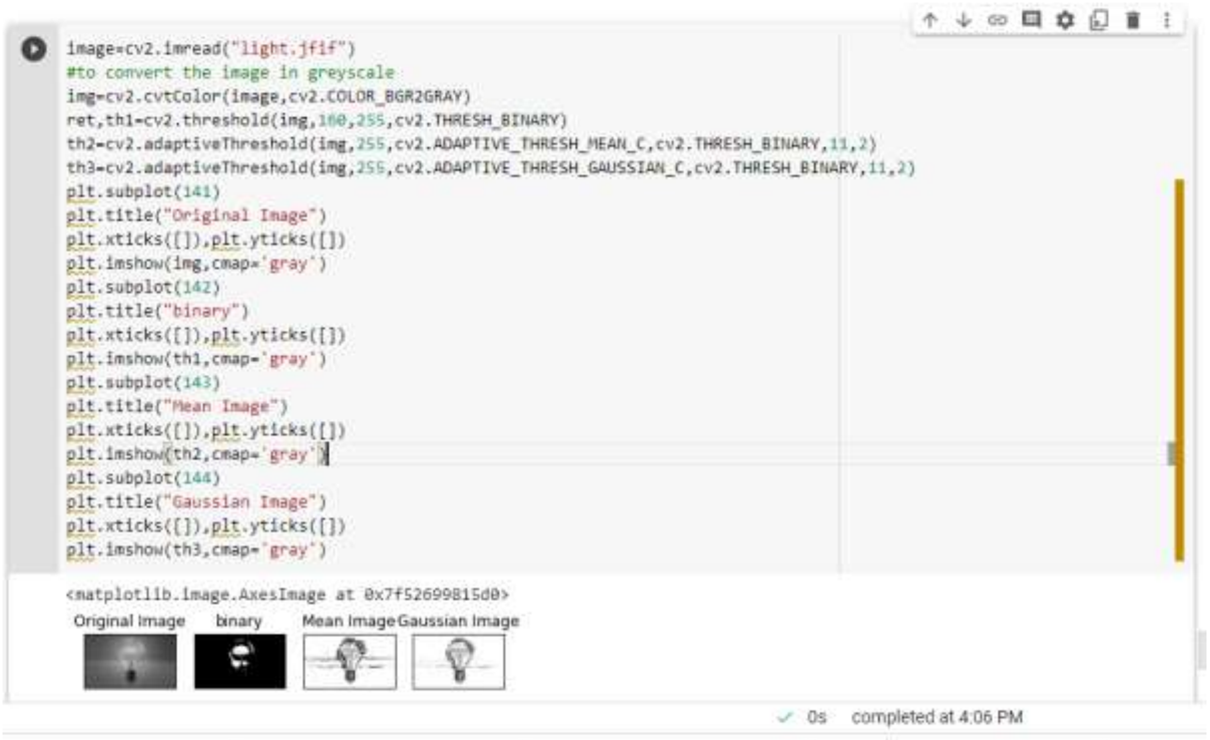
02_GAURAV BANE

```
image=cv2.imread("light.jfif")
#to convert the image in greyscale
img=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
ret,th1=cv2.threshold(img,160,255,cv2.THRESH_BINARY)
th2=cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_MEAN_C,cv2.THRESH
_BINARY,11,2)
th3=cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.TH
RESH_BINARY,11,2)
plt.subplot(141)
plt.title("Original Image")
plt.xticks([]),plt.yticks([])
plt.imshow(img,cmap='gray')
plt.subplot(142)
plt.title("binary")
plt.xticks([]),plt.yticks([])
plt.imshow(th1,cmap='gray')
plt.subplot(143)
plt.title("Mean Image")
plt.xticks([]),plt.yticks([])
plt.imshow(th2,cmap='gray')
plt.subplot(144)
plt.title("Gaussian Image")
plt.xticks([]),plt.yticks([])
plt.imshow(th3,cmap='gray')
```

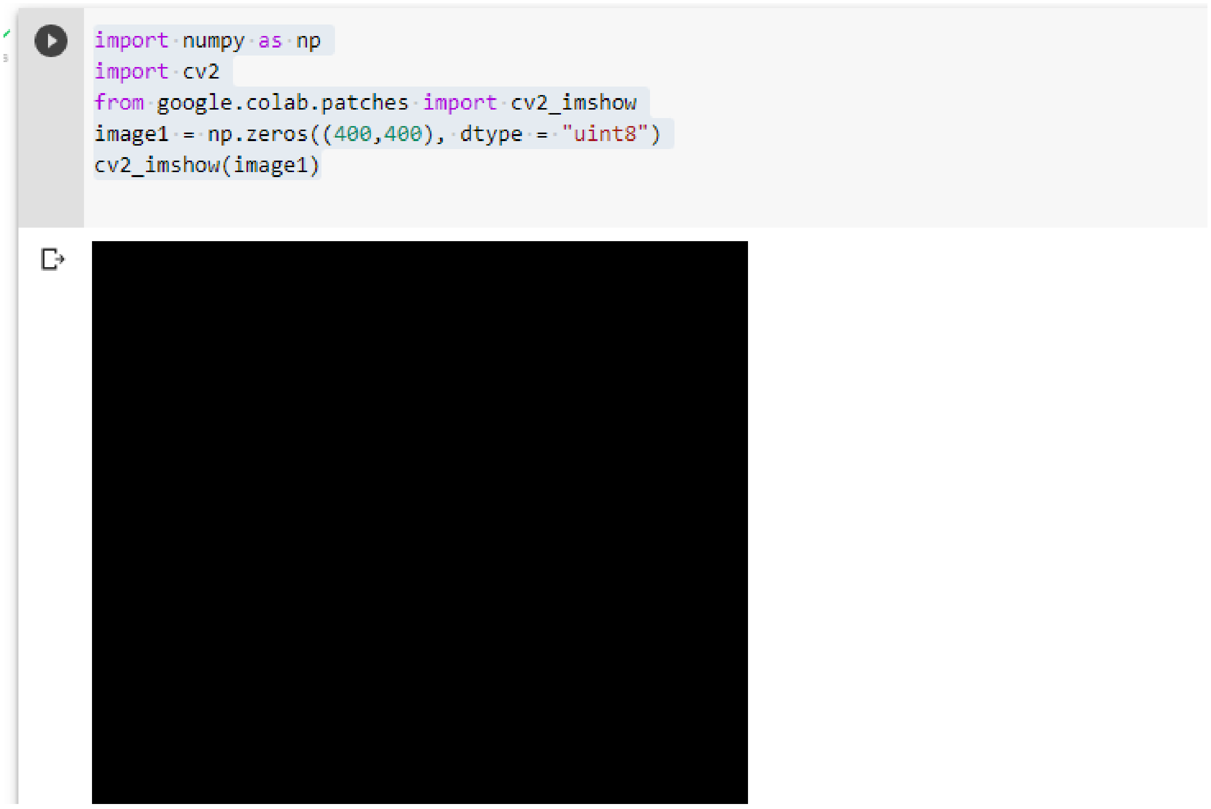


```
image=cv2.imread("light.jfif")
#to convert the image in greyscale
img=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
ret,th1=cv2.threshold(img,160,255,cv2.THRESH_BINARY)
th2=cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_MEAN_C,cv2.THRESH_BINARY,11,2)
th3=cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.THRESH_BINARY,11,2)
plt.subplot(141)
plt.title("Original Image")
plt.xticks([]),plt.yticks([])
plt.imshow(img,cmap='gray')
plt.subplot(142)
plt.title("binary")
plt.xticks([]),plt.yticks([])
plt.imshow(th1,cmap='gray')
plt.subplot(143)
plt.title("Mean Image")
plt.xticks([]),plt.yticks([])
plt.imshow(th2,cmap='gray')
plt.subplot(144)
plt.title("Gaussian Image")
plt.xticks([]),plt.yticks([])
plt.imshow(th3,cmap='gray')
```

```
<matplotlib.image.AxesImage at 0x7f52699815d0>
```

02_GAURAV BANE

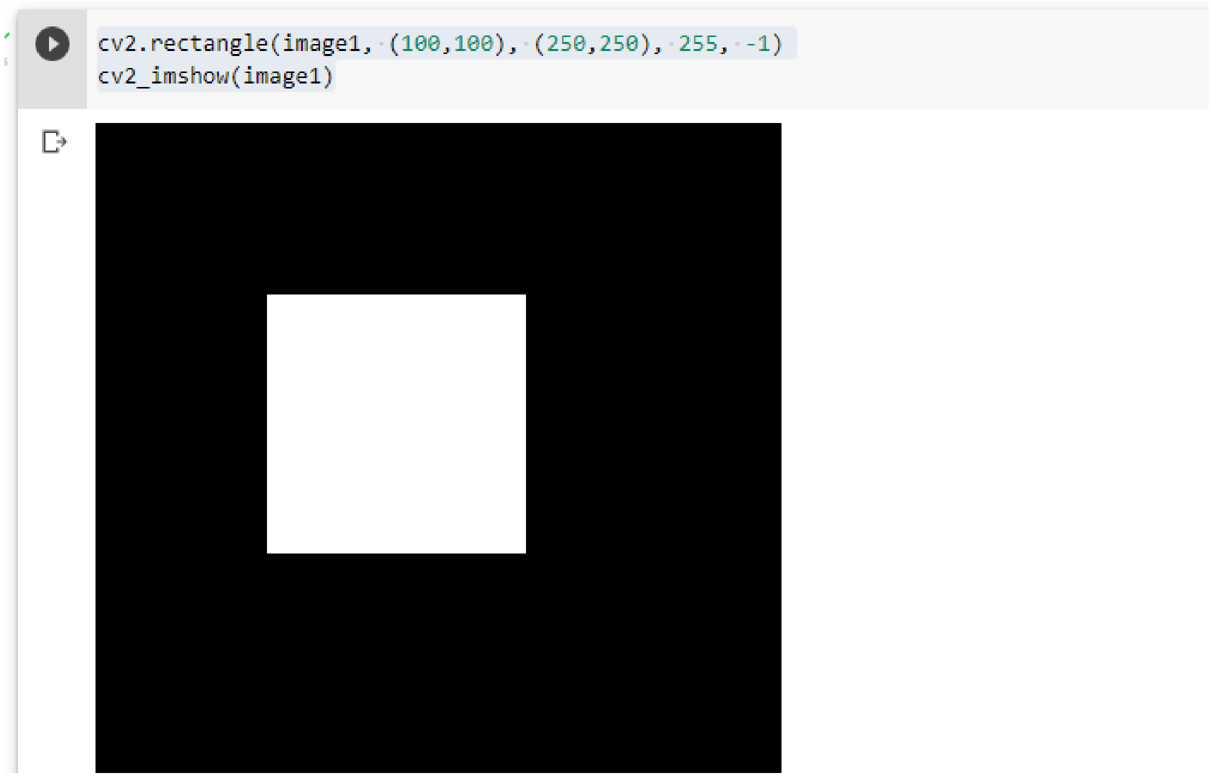
```
import numpy as np
import cv2
from google.colab.patches import cv2_imshow
image1 = np.zeros((400,400), dtype = "uint8")
cv2_imshow(image1)
```



```
import numpy as np
import cv2
from google.colab.patches import cv2_imshow
image1 = np.zeros((400,400), dtype = "uint8")
cv2_imshow(image1)
```

```
cv2.rectangle(image1, (100,100), (250,250), 255, -1)
cv2_imshow(image1)
```

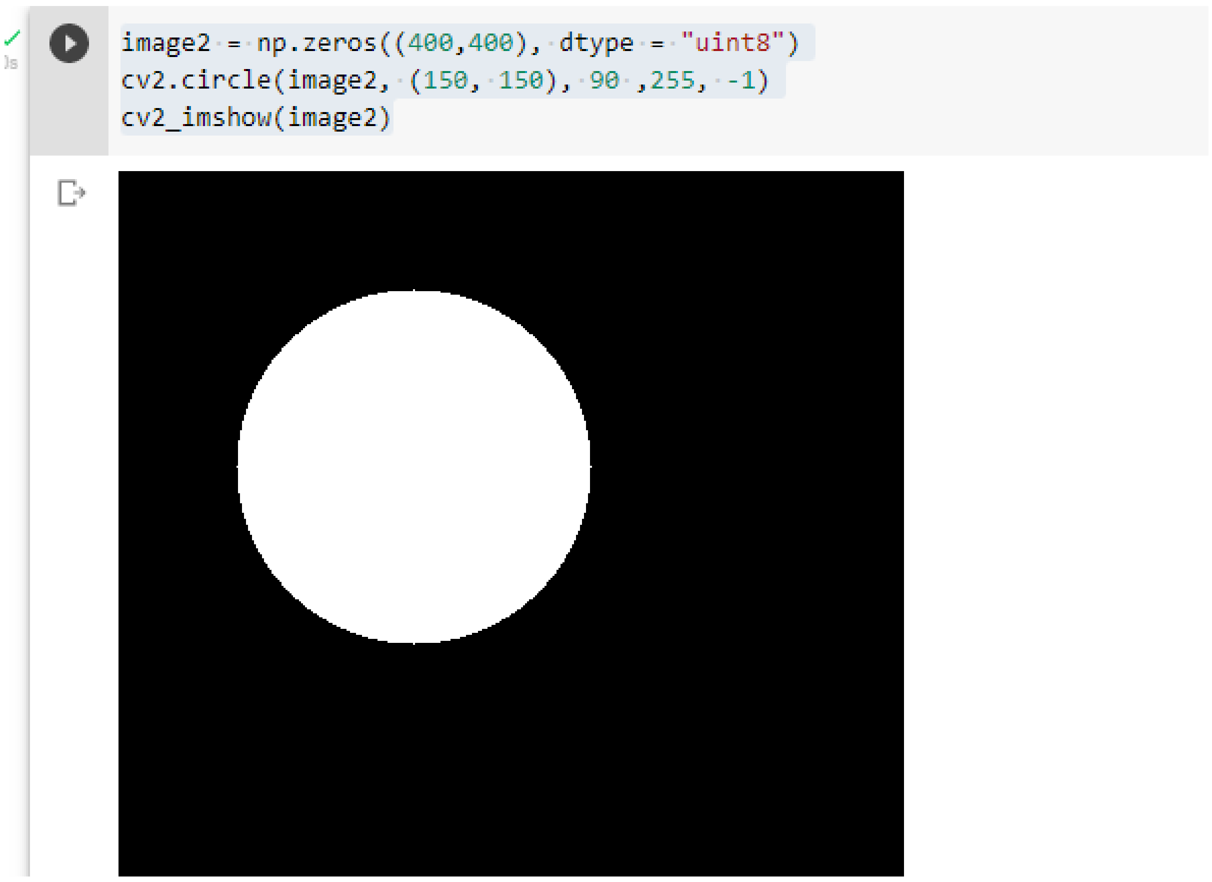
02_GAURAV BANE



```
cv2.rectangle(image1, (100,100), (250,250), 255, -1)
cv2_imshow(image1)
```

```
image2 = np.zeros((400,400), dtype = "uint8")
cv2.circle(image2, (150, 150), 90 ,255, -1)
cv2_imshow(image2)
```

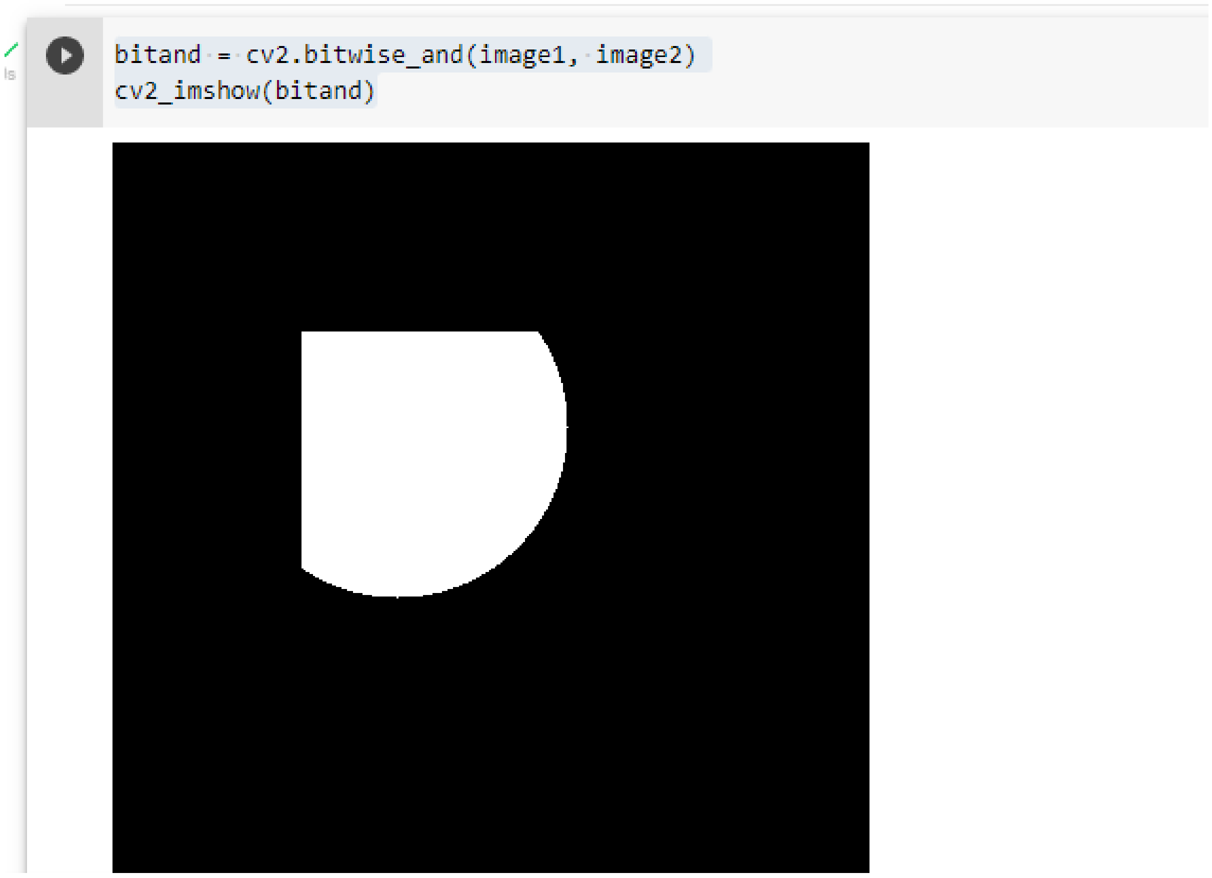
02_GAURAV BANE



```
image2 = np.zeros((400,400), dtype = "uint8")
cv2.circle(image2, (150, 150), 90 ,255, -1)
cv2_imshow(image2)
```

```
bitand = cv2.bitwise_and(image1, image2)
cv2_imshow(bitand)
```

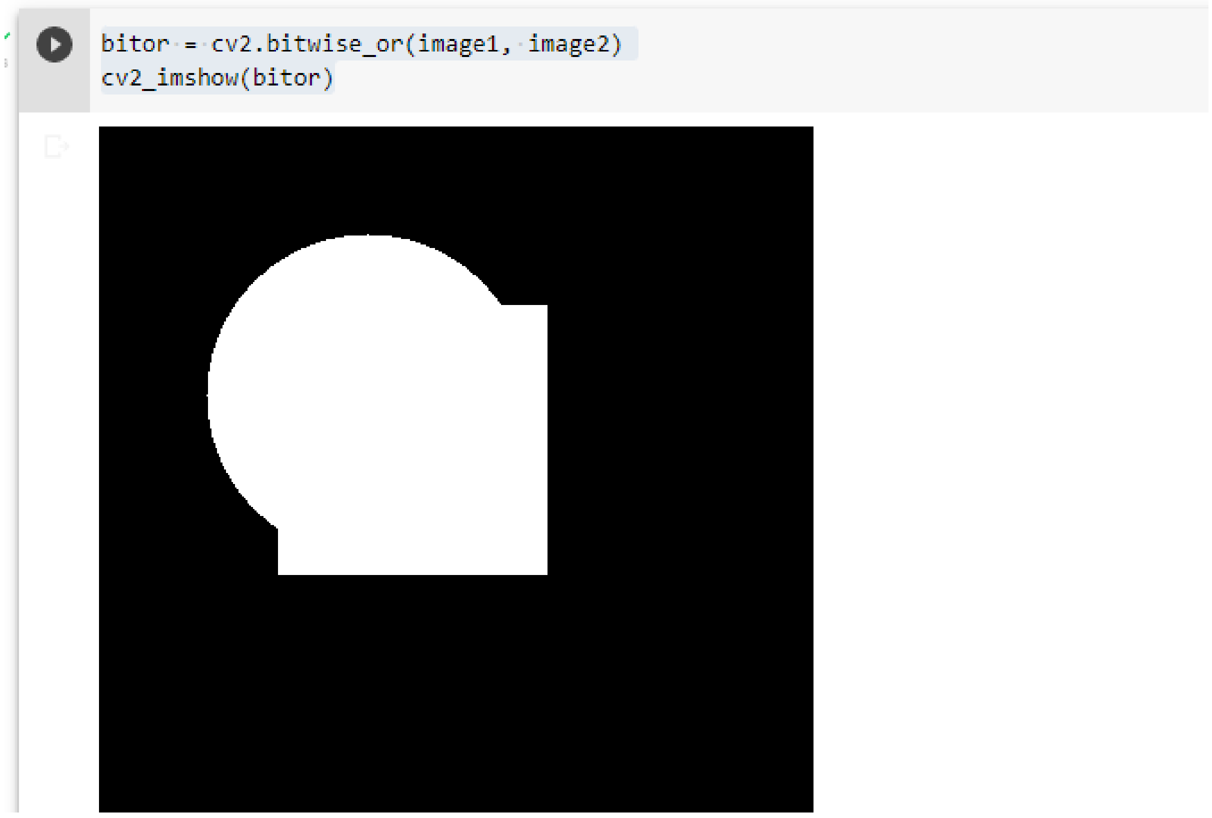
02_GAURAV BANE



```
bitand = cv2.bitwise_and(image1, image2)
cv2_imshow(bitand)
```

```
bitor = cv2.bitwise_or(image1, image2)
cv2_imshow(bitor)
```

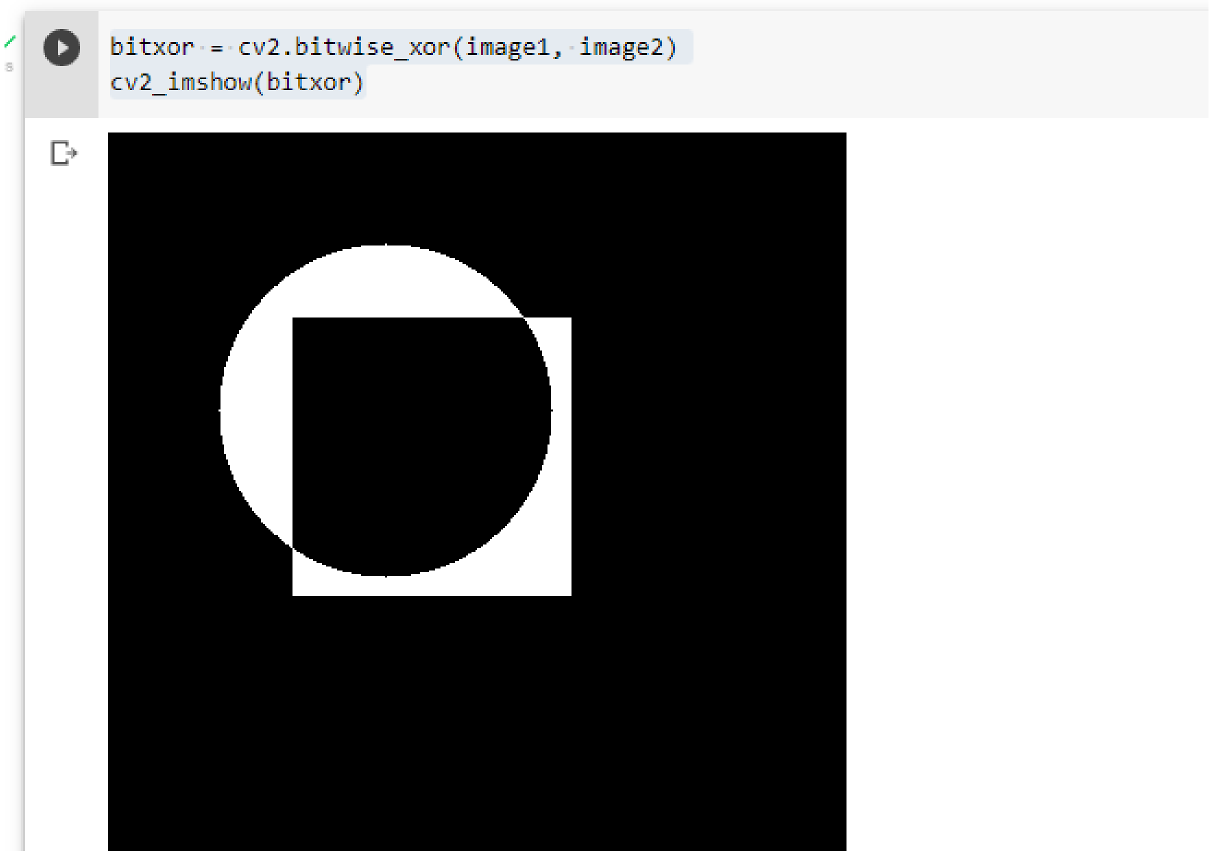
02_GAURAV BANE



```
bitor = cv2.bitwise_or(image1, image2)
cv2_imshow(bitor)
```

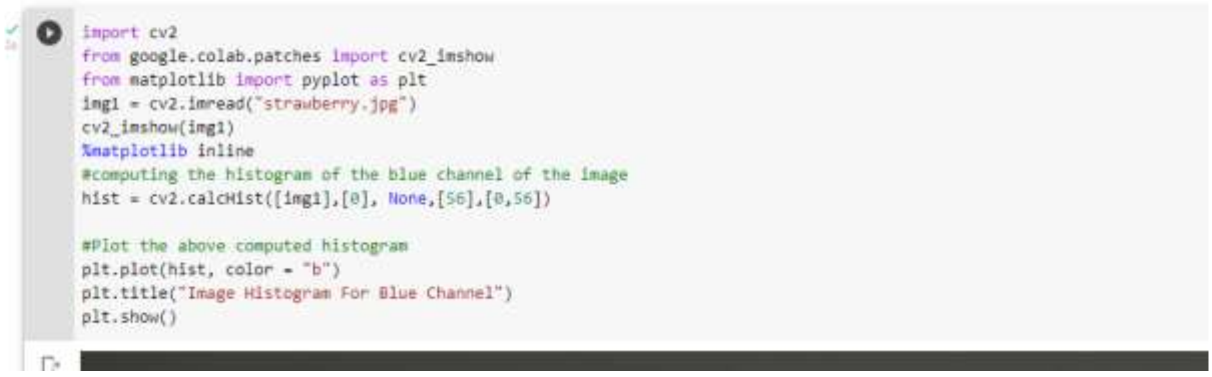
```
bitxor = cv2.bitwise_xor(image1, image2)
cv2_imshow(bitxor)
```

02_GAURAV BANE

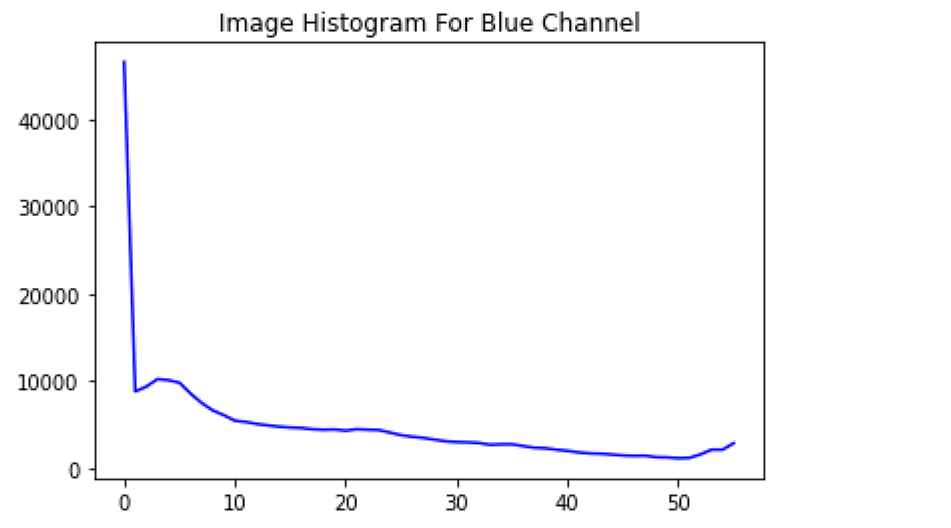


```
bitxor = cv2.bitwise_xor(image1, image2)
cv2_imshow(bitxor)
```

```
import cv2
from google.colab.patches import cv2_imshow
from matplotlib import pyplot as plt
img1 = cv2.imread("strawberry.jpg")
cv2_imshow(img1)
%matplotlib inline
#computing the histogram of the blue channel of the image
hist = cv2.calcHist([img1],[0], None,[256],[0,256])

#Plot the above computed histogram
plt.plot(hist, color = "b")
plt.title("Image Histogram For Blue Channel")
plt.show()
```

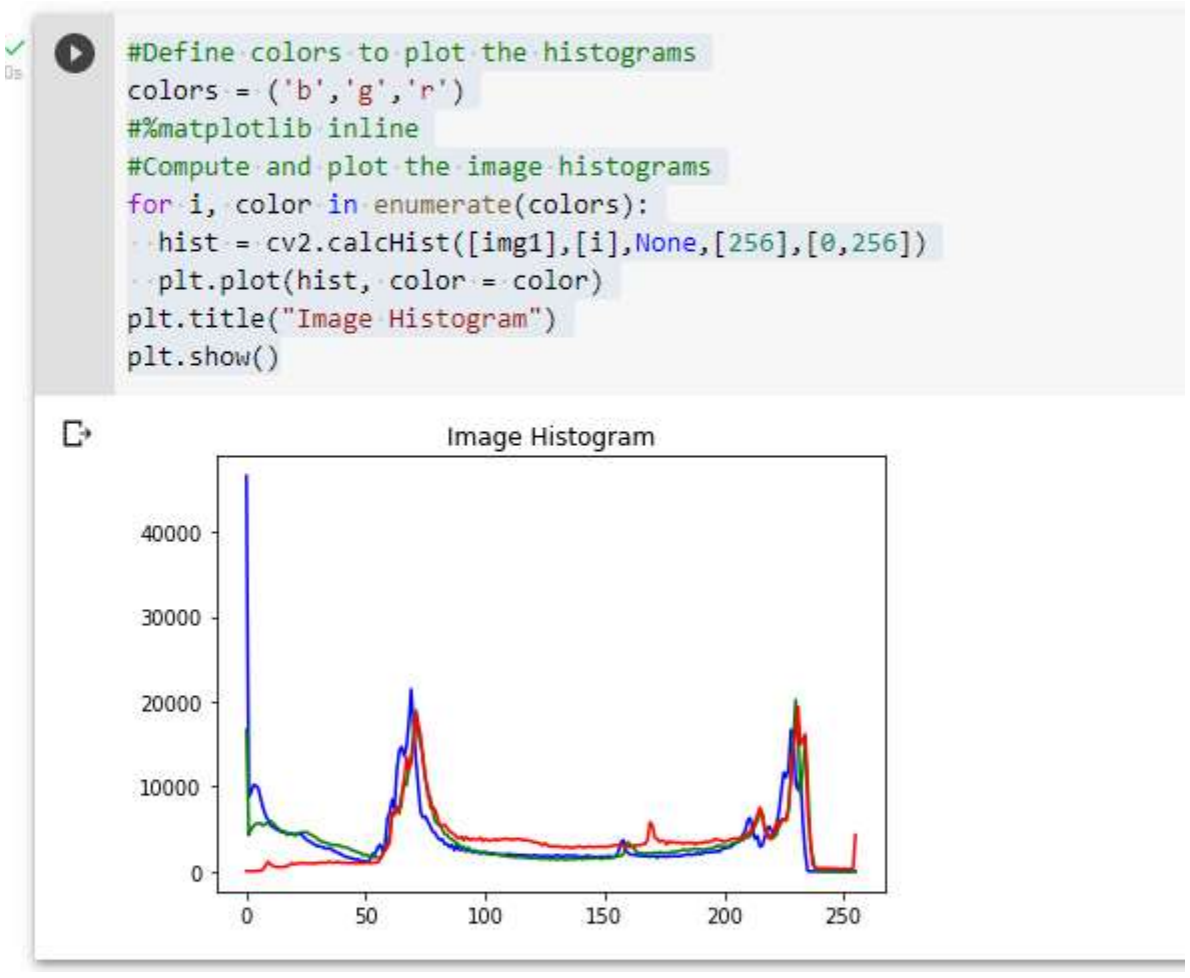


```
import cv2
from google.colab.patches import cv2_imshow
from matplotlib import pyplot as plt
img1 = cv2.imread("strawberry.jpg")
cv2_imshow(img1)
%matplotlib inline
#computing the histogram of the blue channel of the image
hist = cv2.calcHist([img1],[0], None,[56],[0,56])

#Plot the above computed histogram
plt.plot(hist, color = "b")
plt.title("Image Histogram For Blue Channel")
plt.show()
```



```
#Define colors to plot the histograms
colors = ('b','g','r')
#%matplotlib inline
#Compute and plot the image histograms
for i, color in enumerate(colors):
  hist = cv2.calcHist([img1],[i],None,[256],[0,256])
  plt.plot(hist, color = color)
plt.title("Image Histogram")
plt.show()
```

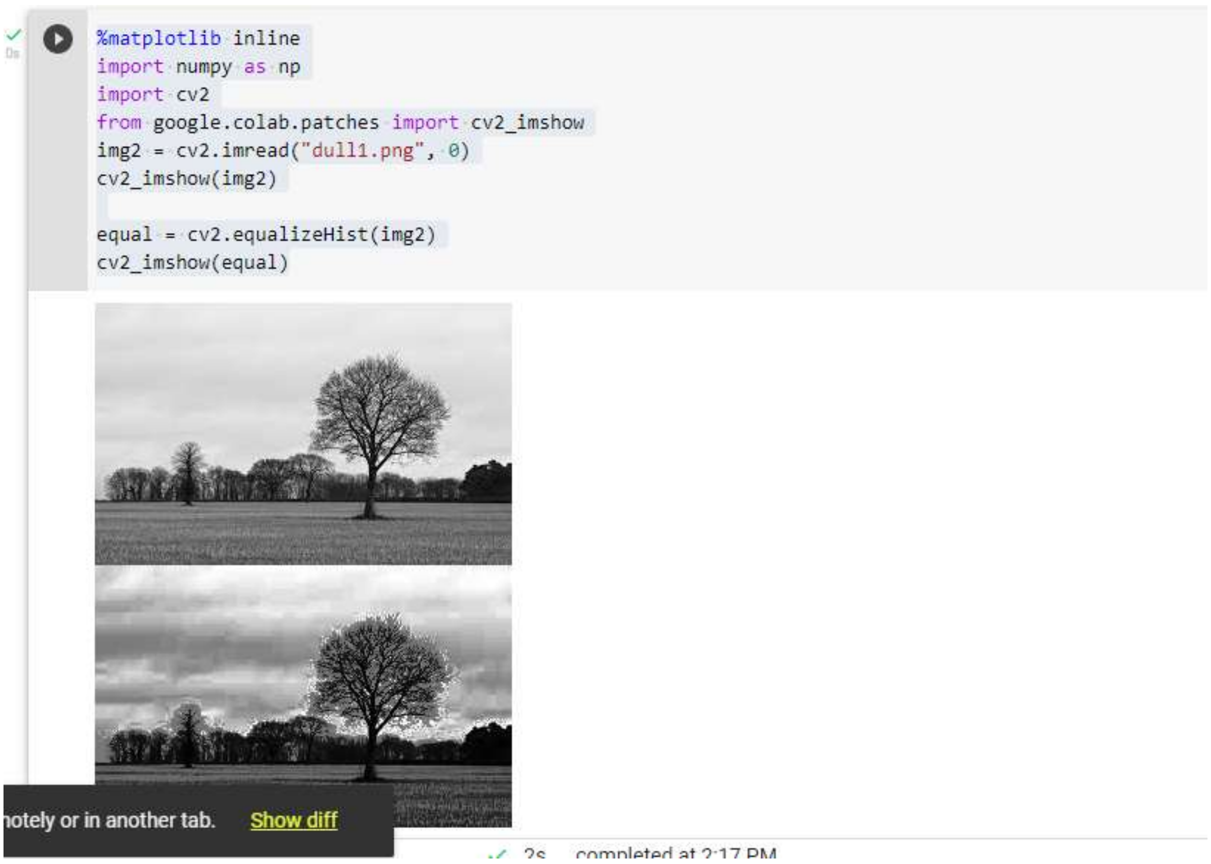
02_GAURAV BANE



```
#Define colors to plot the histograms
colors = ('b','g','r')
#%matplotlib inline
#Compute and plot the image histograms
for i, color in enumerate(colors):
  hist = cv2.calcHist([img1],[i],None,[256],[0,256])
  plt.plot(hist, color = color)
plt.title("Image Histogram")
plt.show()
```

```
%matplotlib inline
import numpy as np
import cv2
from google.colab.patches import cv2_imshow
img2 = cv2.imread("dull1.png", 0)
cv2_imshow(img2)

equal = cv2.equalizeHist(img2)
cv2_imshow(equal)
```

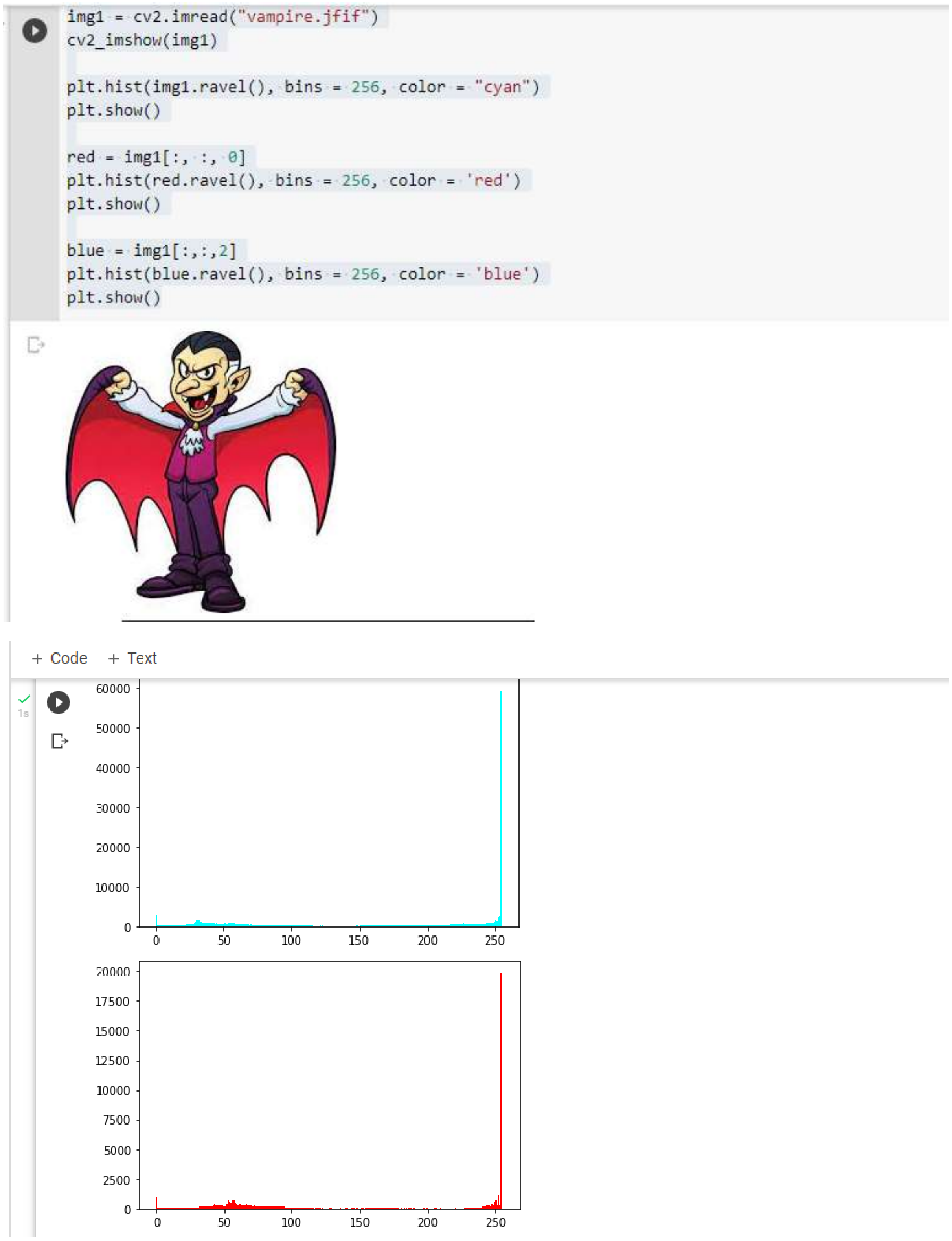
02_GAURAV BANE



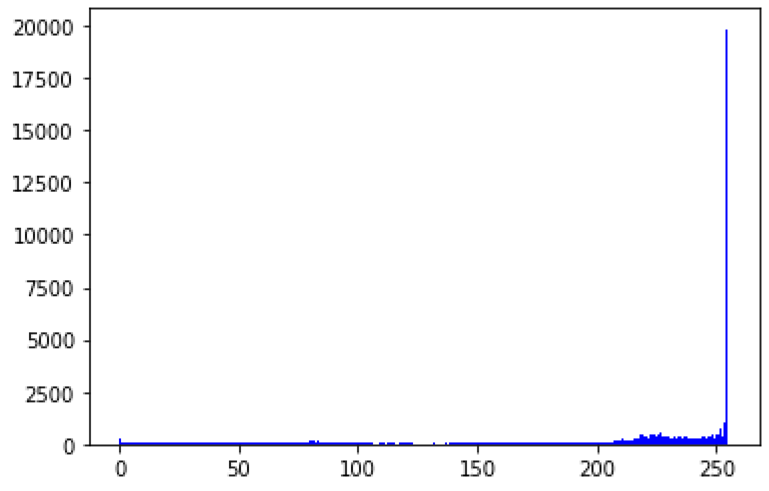
```
%matplotlib inline
import numpy as np
import cv2
from google.colab.patches import cv2_imshow
img2 = cv2.imread("dull1.png", 0)
cv2_imshow(img2)

equal = cv2.equalizeHist(img2)
cv2_imshow(equal)
```

```
img1 = cv2.imread("vampire.jfif")
cv2_imshow(img1)

plt.hist(img1.ravel(), bins = 256, color = "cyan")
plt.show()

red = img1[:, :, 0]
plt.hist(red.ravel(), bins = 256, color = 'red')
plt.show()

blue = img1[:,:,2]
plt.hist(blue.ravel(), bins = 256, color = 'blue')
plt.show()
```

02_GAURAV BANE

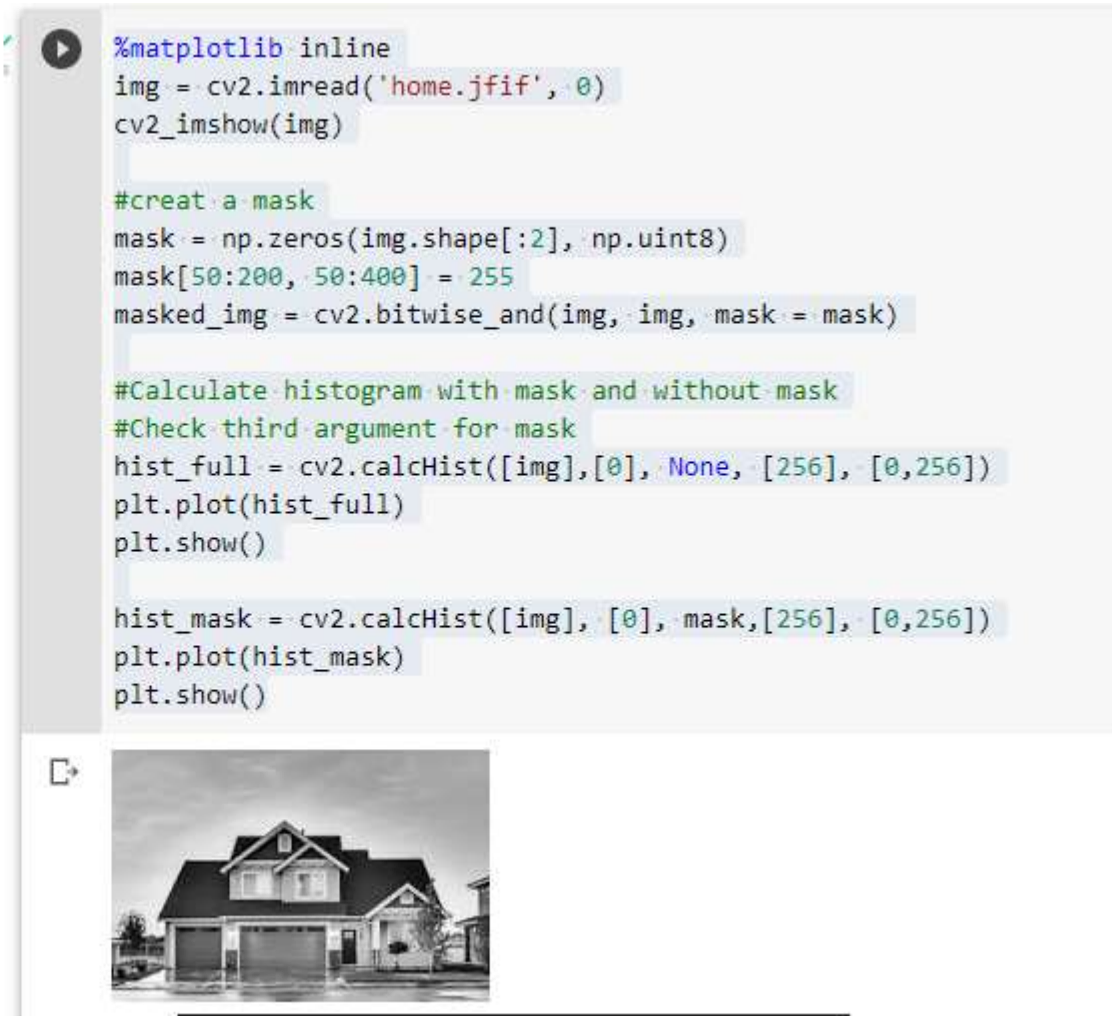


```
img1 = cv2.imread("vampire.jfif")
cv2_imshow(img1)

plt.hist(img1.ravel(), bins = 256, color = "cyan")
plt.show()

red = img1[:, :, 0]
plt.hist(red.ravel(), bins = 256, color = 'red')
plt.show()

blue = img1[:,:,2]
plt.hist(blue.ravel(), bins = 256, color = 'blue')
plt.show()
```



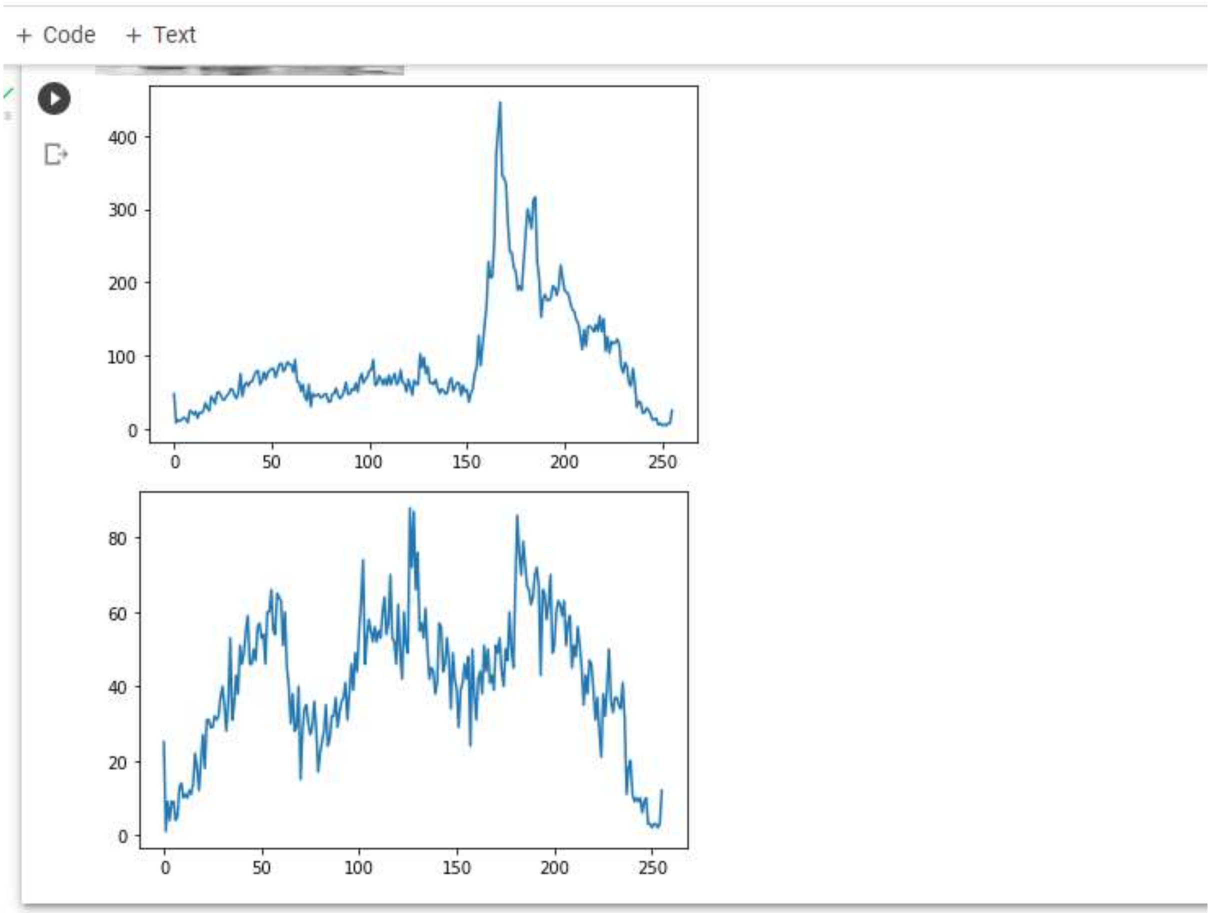
```
%matplotlib inline
img = cv2.imread('home.jfif', 0)
cv2_imshow(img)

#creat a mask
mask = np.zeros(img.shape[:2], np.uint8)
mask[50:200, 50:400] = 255
masked_img = cv2.bitwise_and(img, img, mask = mask)

#Calculate histogram with mask and without mask
#Check third argument for mask
hist_full = cv2.calcHist([img],[0], None, [256], [0,256])
plt.plot(hist_full)
plt.show()

hist_mask = cv2.calcHist([img], [0], mask,[256], [0,256])
plt.plot(hist_mask)
plt.show()
```

02_GAURAV BANE



```
%matplotlib inline
img = cv2.imread('home.jfif', 0)
cv2_imshow(img)

#creat a mask
mask = np.zeros(img.shape[:2], np.uint8)
mask[50:200, 50:400] = 255
masked_img = cv2.bitwise_and(img, img, mask = mask)

#Calculate histogram with mask and without mask
#Check third argument for mask
hist_full = cv2.calcHist([img],[0], None, [256], [0,256])
plt.plot(hist_full)
plt.show()

hist_mask = cv2.calcHist([img], [0], mask,[256], [0,256])
plt.plot(hist_mask)
plt.show()
```

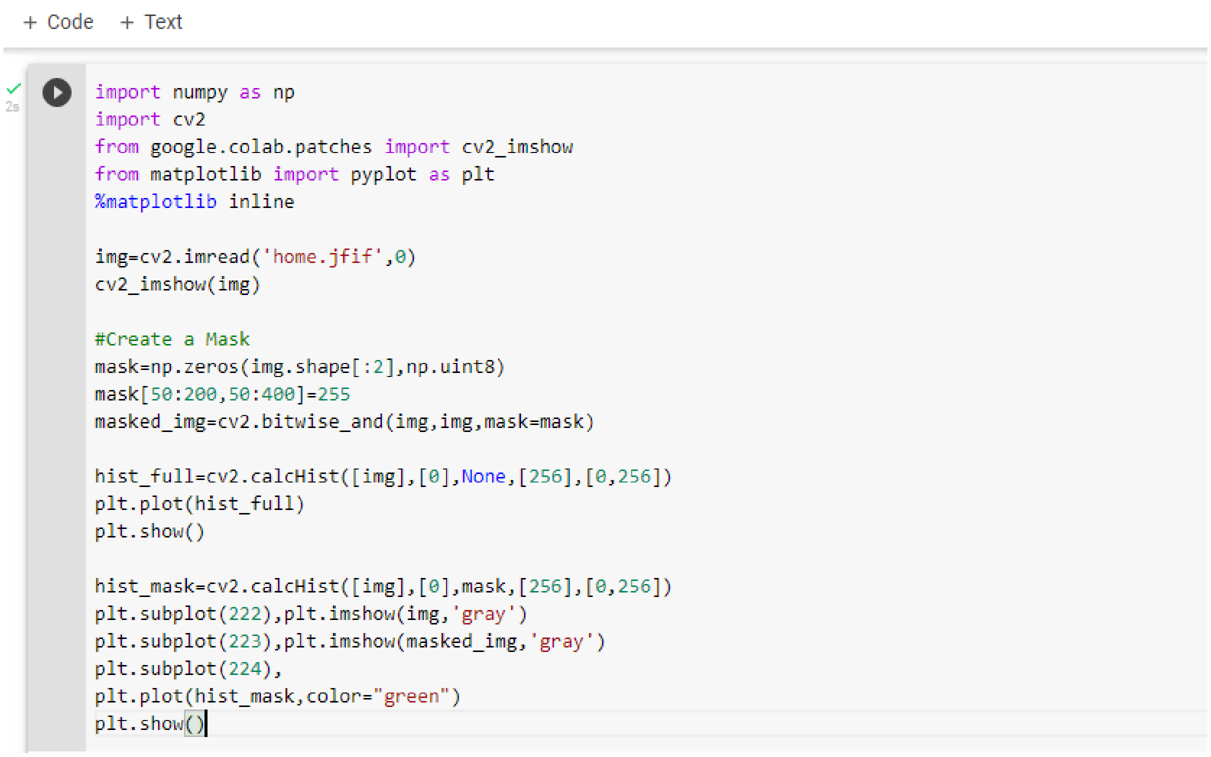
02_GAURAV BANE



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

img=cv2.imread('home.jfif',0)
cv2_imshow(img)

#Create a Mask
mask=np.zeros(img.shape[:2],np.uint8)
mask[50:200,50:400]=255
masked_img=cv2.bitwise_and(img,img,mask=mask)

hist_full=cv2.calcHist([img],[0],None,[256],[0,256])
plt.plot(hist_full)
plt.show()

hist_mask=cv2.calcHist([img],[0],mask,[256],[0,256])
plt.subplot(222),plt.imshow(img,'gray')
plt.subplot(223),plt.imshow(masked_img,'gray')
plt.subplot(224),
```

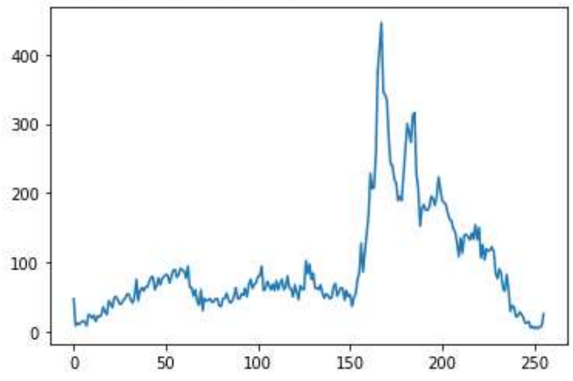
02_GAURAV BANE

```
plt.plot(hist_mask,color="green")
plt.show()
```

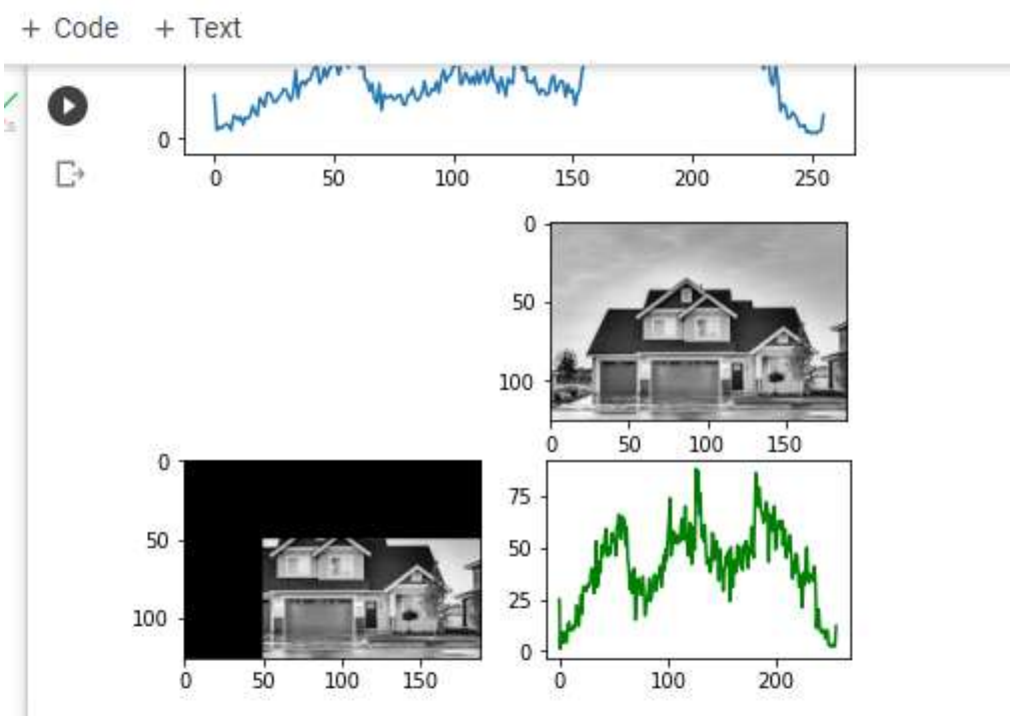


```
import numpy as np
import cv2
from google.colab.patches import cv2_imshow
from matplotlib import pyplot as plt
%matplotlib inline

img=cv2.imread('home.jfif',0)
cv2_imshow(img)

#Create a Mask
mask=np.zeros(img.shape[:2],np.uint8)
mask[50:200,50:400]=255
masked_img=cv2.bitwise_and(img,img,mask=mask)

hist_full=cv2.calcHist([img],[0],None,[256],[0,256])
plt.plot(hist_full)
plt.show()

hist_mask=cv2.calcHist([img],[0],mask,[256],[0,256])
plt.subplot(222),plt.imshow(img,'gray')
plt.subplot(223),plt.imshow(masked_img,'gray')
plt.subplot(224),
plt.plot(hist_mask,color="green")
plt.show()
```

+ Code + Text

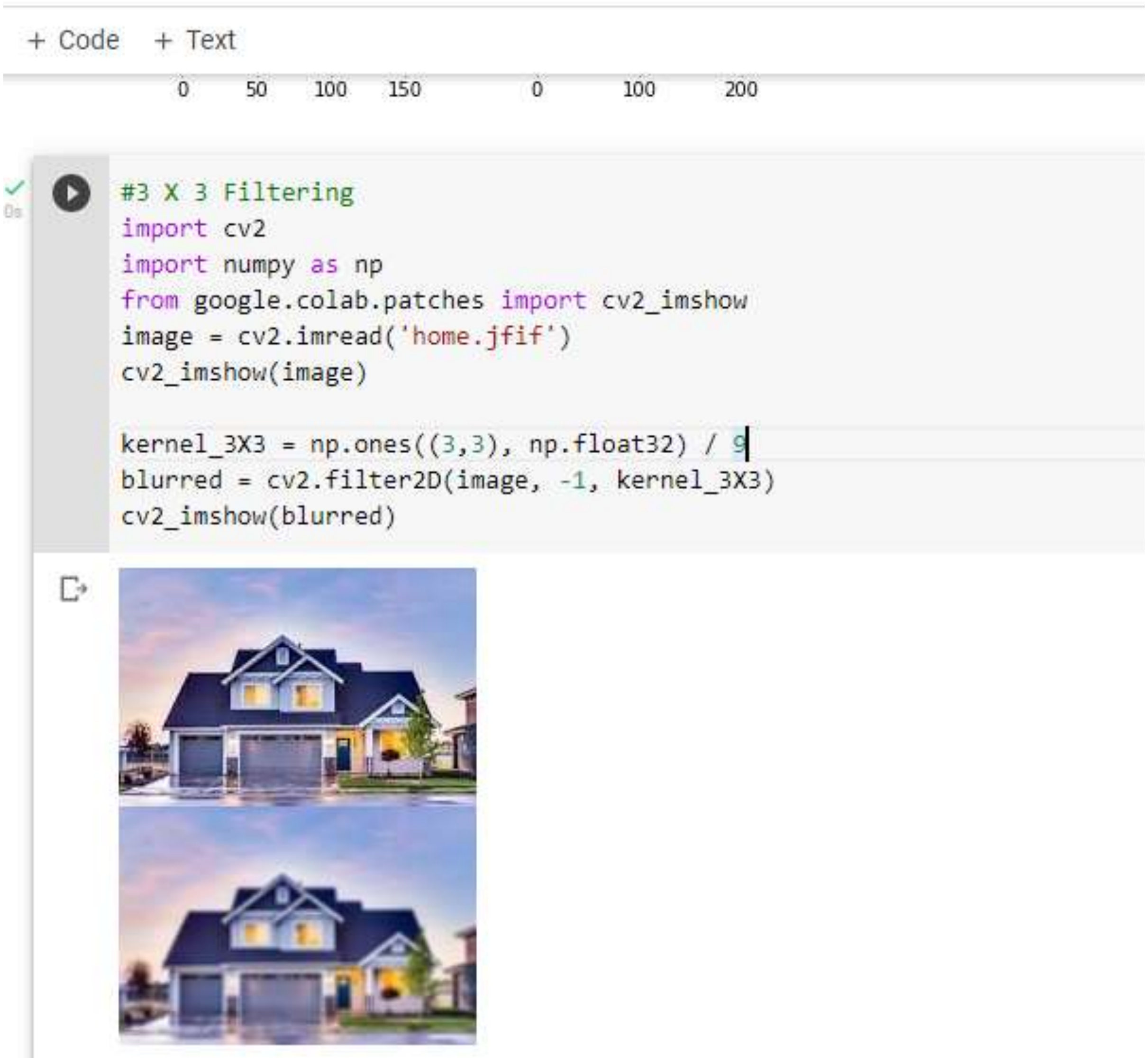


02_GAURAV BANE



```
#3 X 3 Filtering
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
image = cv2.imread('home.jfif')
cv2_imshow(image)

kernel_3X3 = np.ones((3,3), np.float32) / 9
blurred = cv2.filter2D(image, -1, kernel_3X3)
cv2_imshow(blurred)
```

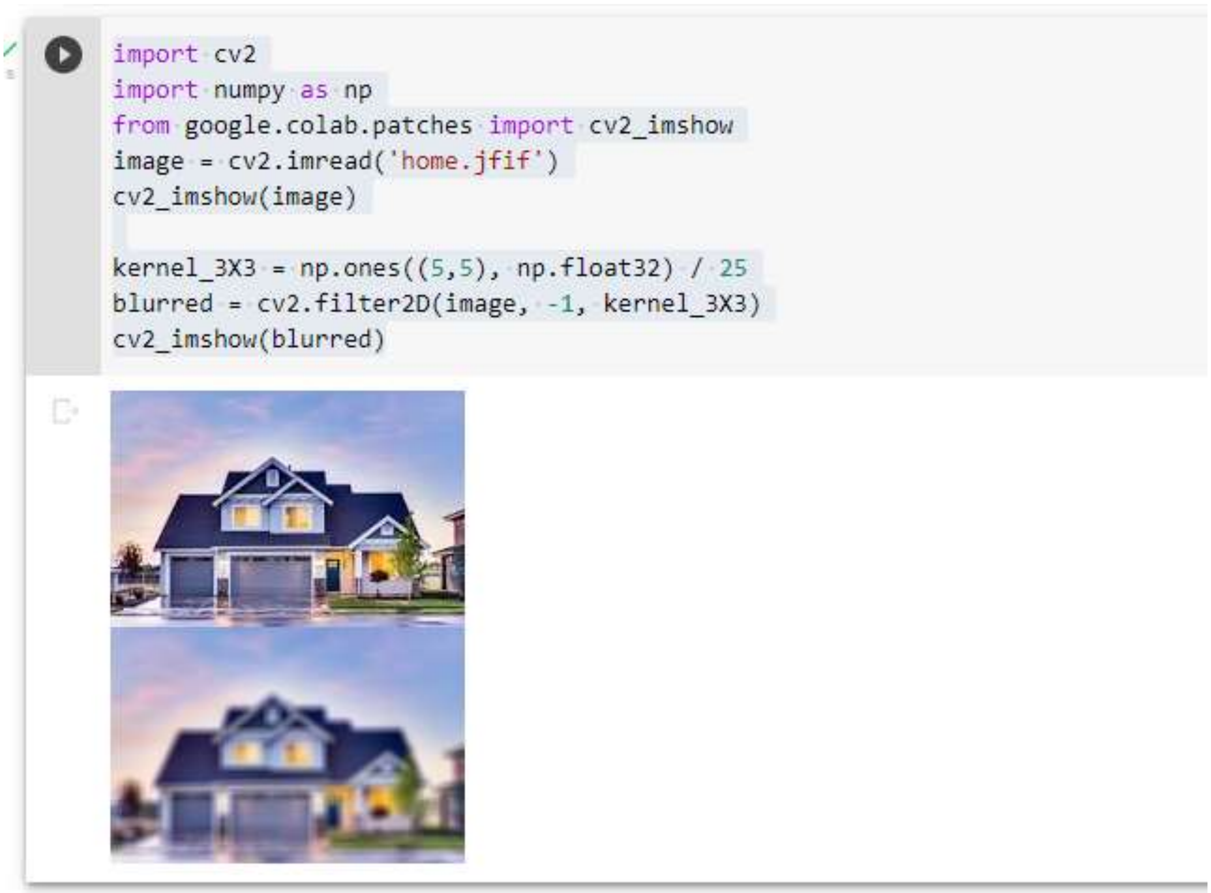
02_GAURAV BANE



```
#3 X 3 Filtering
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
image = cv2.imread('home.jfif')
cv2_imshow(image)

kernel_3X3 = np.ones((3,3), np.float32) / 9
blurred = cv2.filter2D(image, -1, kernel_3X3)
cv2_imshow(blurred)
```

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
image = cv2.imread('home.jfif')
cv2_imshow(image)

kernel_3X3 = np.ones((5,5), np.float32) / 25
blurred = cv2.filter2D(image, -1, kernel_3X3)
cv2_imshow(blurred)
```

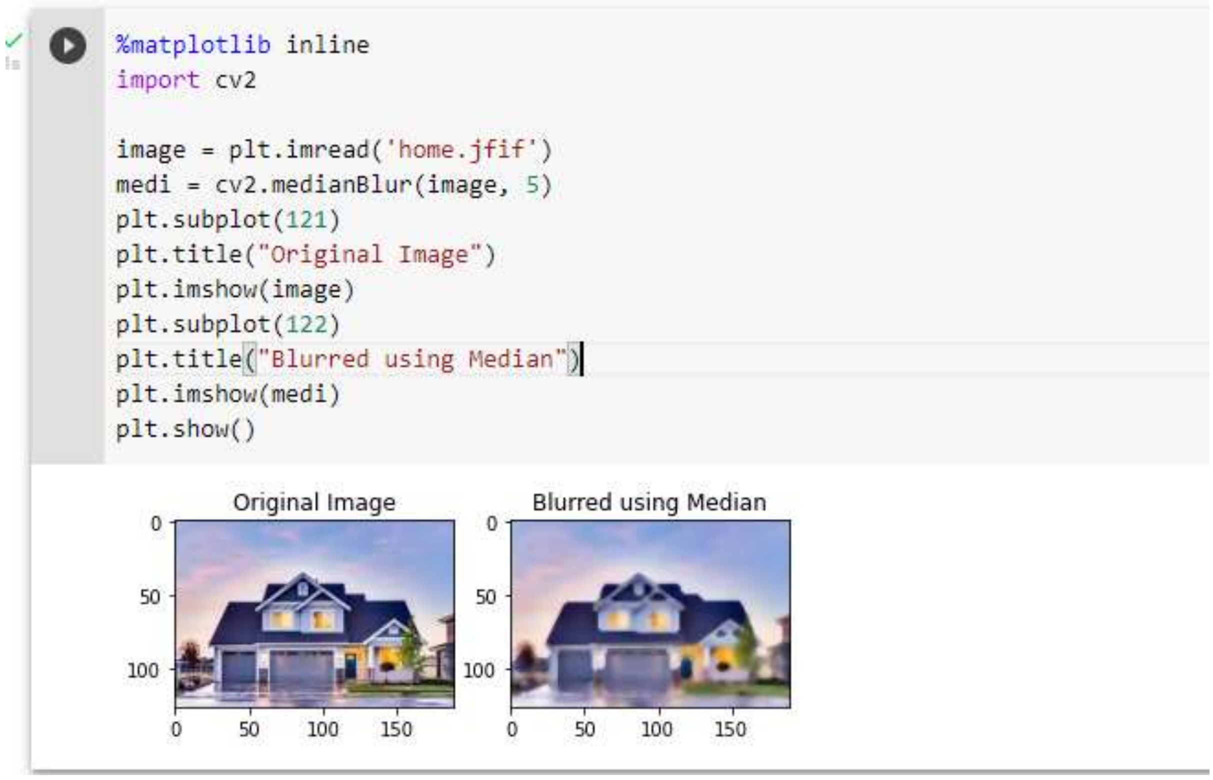
02_GAURAV BANE



```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
image = cv2.imread('home.jfif')
cv2_imshow(image)

kernel_3X3 = np.ones((5,5), np.float32) / 25
blurred = cv2.filter2D(image, -1, kernel_3X3)
cv2_imshow(blurred)
```

```
%matplotlib inline
import cv2

image = plt.imread('home.jfif')
medi = cv2.medianBlur(image, 5)
plt.subplot(121)
plt.title("Original Image")
plt.imshow(image)
plt.subplot(122)
plt.title("Blurred using Median")
plt.imshow(medi)
plt.show()
```

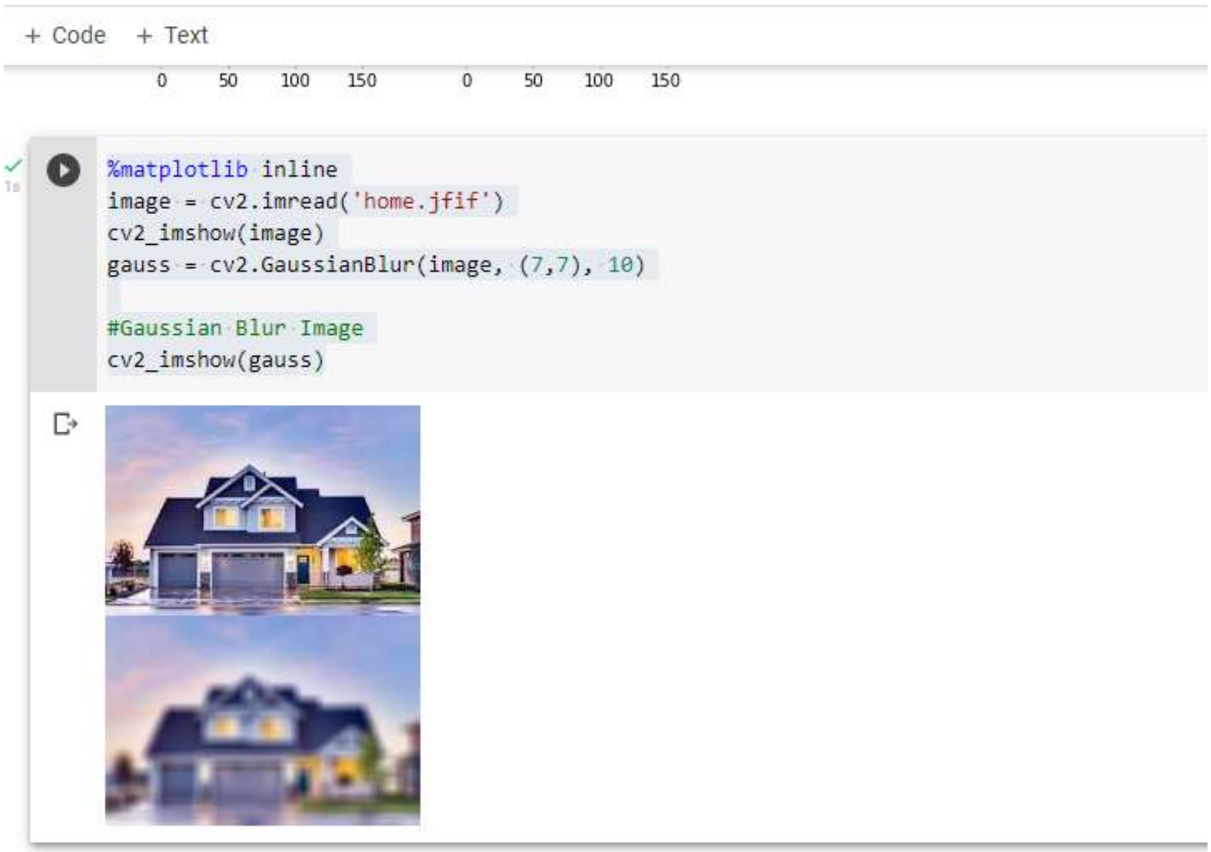
02_GAURAV BANE



```
%matplotlib inline
import cv2

image = plt.imread('home.jfif')
medi = cv2.medianBlur(image, 5)
plt.subplot(121)
plt.title("Original Image")
plt.imshow(image)
plt.subplot(122)
plt.title("Blurred using Median")
plt.imshow(medi)
plt.show()
```

```
%matplotlib inline
image = cv2.imread('home.jfif')
cv2_imshow(image)
gauss = cv2.GaussianBlur(image, (7,7), 10)

#Gaussian Blur Image
cv2_imshow(gauss)
```

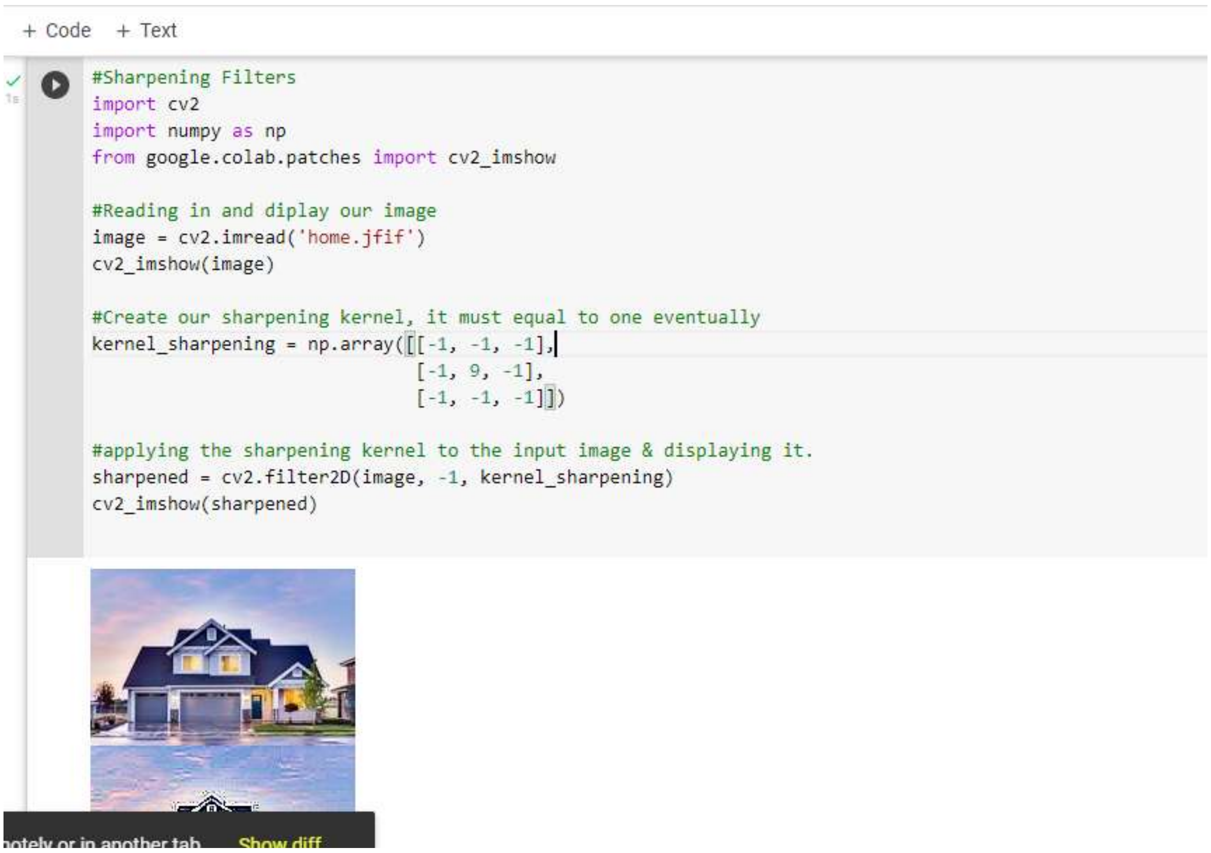
02_GAURAV BANE



```
%matplotlib inline
image = cv2.imread('home.jfif')
cv2_imshow(image)
gauss = cv2.GaussianBlur(image, (7,7), 10)

#Gaussian Blur Image
cv2_imshow(gauss)
```

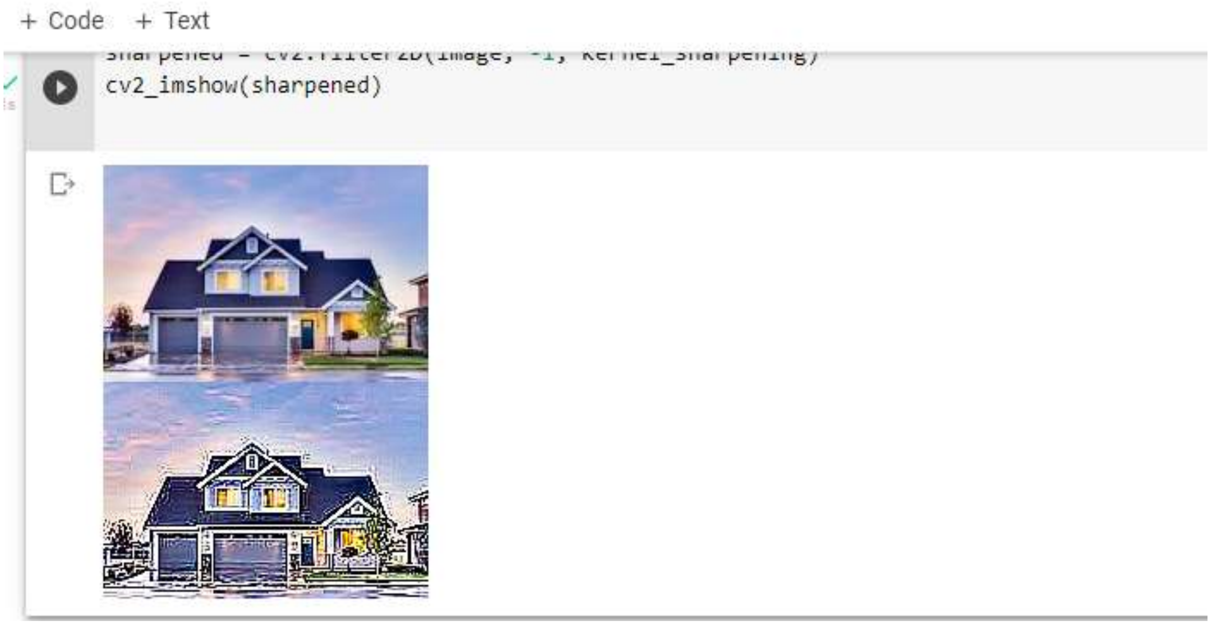
```
#Sharpening Filters
import cv2
import numpy as np
from google.colab.patches import cv2_imshow

#Reading in and diplay our image
image = cv2.imread('home.jfif')
cv2_imshow(image)

#Create our sharpening kernel, it must equal to one eventually
kernel_sharpening = np.array([[-1, -1, -1],
                              [-1, 9, -1],
                              [-1, -1, -1]])

#applying the sharpening kernel to the input image & displaying it.
sharpened = cv2.filter2D(image, -1, kernel_sharpening)
cv2_imshow(sharpened)
```

02_GAURAV BANE



+ Code + Text

```
#Sharpening Filters
import cv2
import numpy as np
from google.colab.patches import cv2_imshow

#Reading in and diplay our image
image = cv2.imread('home.jfif')
cv2_imshow(image)

#Create our sharpening kernel, it must equal to one eventually
kernel_sharpening = np.array([[-1, -1, -1],
                              [-1, 9, -1],
                              [-1, -1, -1]])

#applying the sharpening kernel to the input image & displaying it.
sharpened = cv2.filter2D(image, -1, kernel_sharpening)
cv2_imshow(sharpened)
```



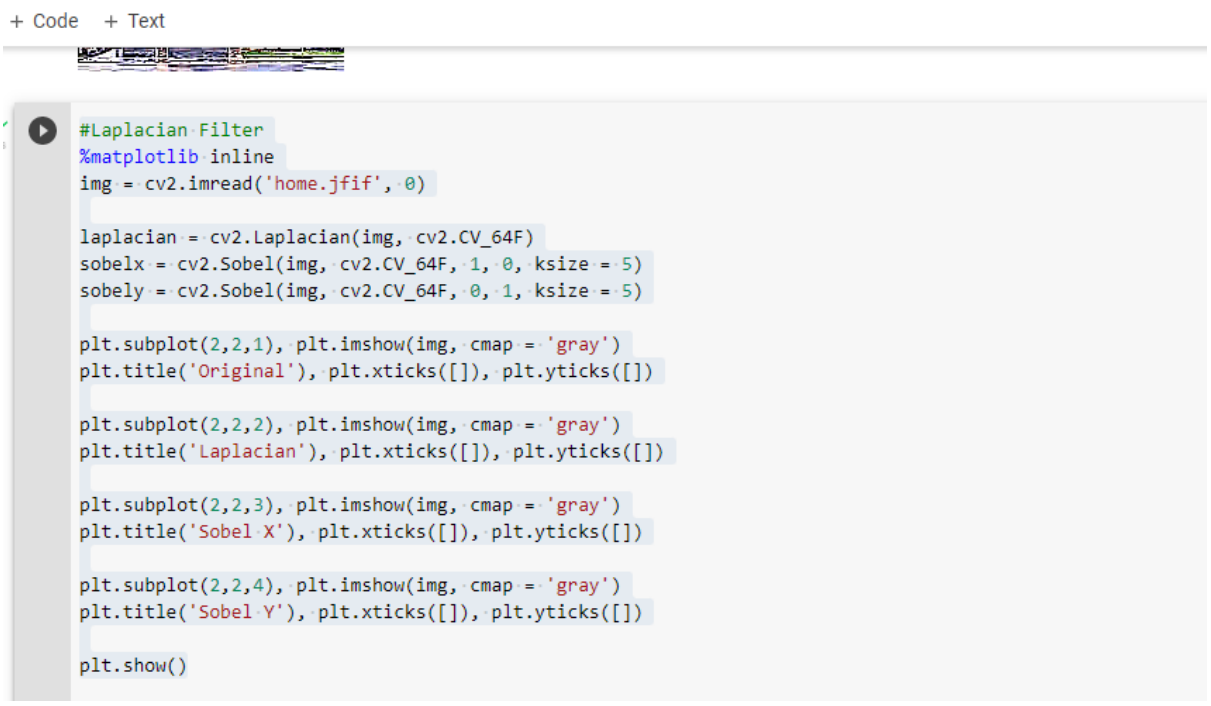
+ Code + Text

```
sharpened = cv2.filter2D(image, -1, kernel_sharpening)
cv2_imshow(sharpened)
```

```
#Laplacian Filter
%matplotlib inline
img = cv2.imread('home.jfif', 0)

laplacian = cv2.Laplacian(img, cv2.CV_64F)
sobelx = cv2.Sobel(img, cv2.CV_64F, 1, 0, ksize = 5)
```

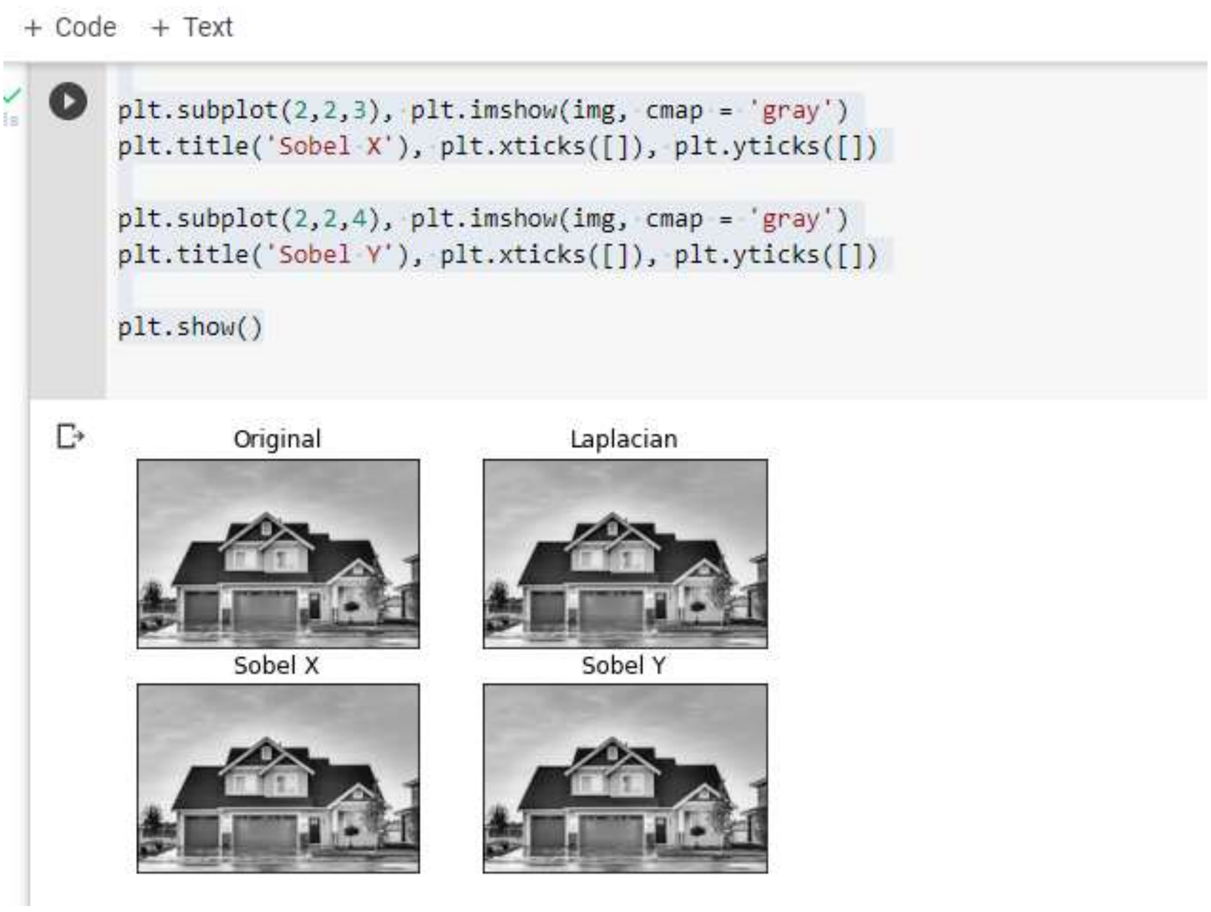
02_GAURAV BANE

```
sobely = cv2.Sobel(img, cv2.CV_64F, 0, 1, ksize = 5)

plt.subplot(2,2,1), plt.imshow(img, cmap = 'gray')
plt.title('Original'), plt.xticks([]), plt.yticks([])

plt.subplot(2,2,2), plt.imshow(img, cmap = 'gray')
plt.title('Laplacian'), plt.xticks([]), plt.yticks([])

plt.subplot(2,2,3), plt.imshow(img, cmap = 'gray')
plt.title('Sobel X'), plt.xticks([]), plt.yticks([])

plt.subplot(2,2,4), plt.imshow(img, cmap = 'gray')
plt.title('Sobel Y'), plt.xticks([]), plt.yticks([])

plt.show()
```



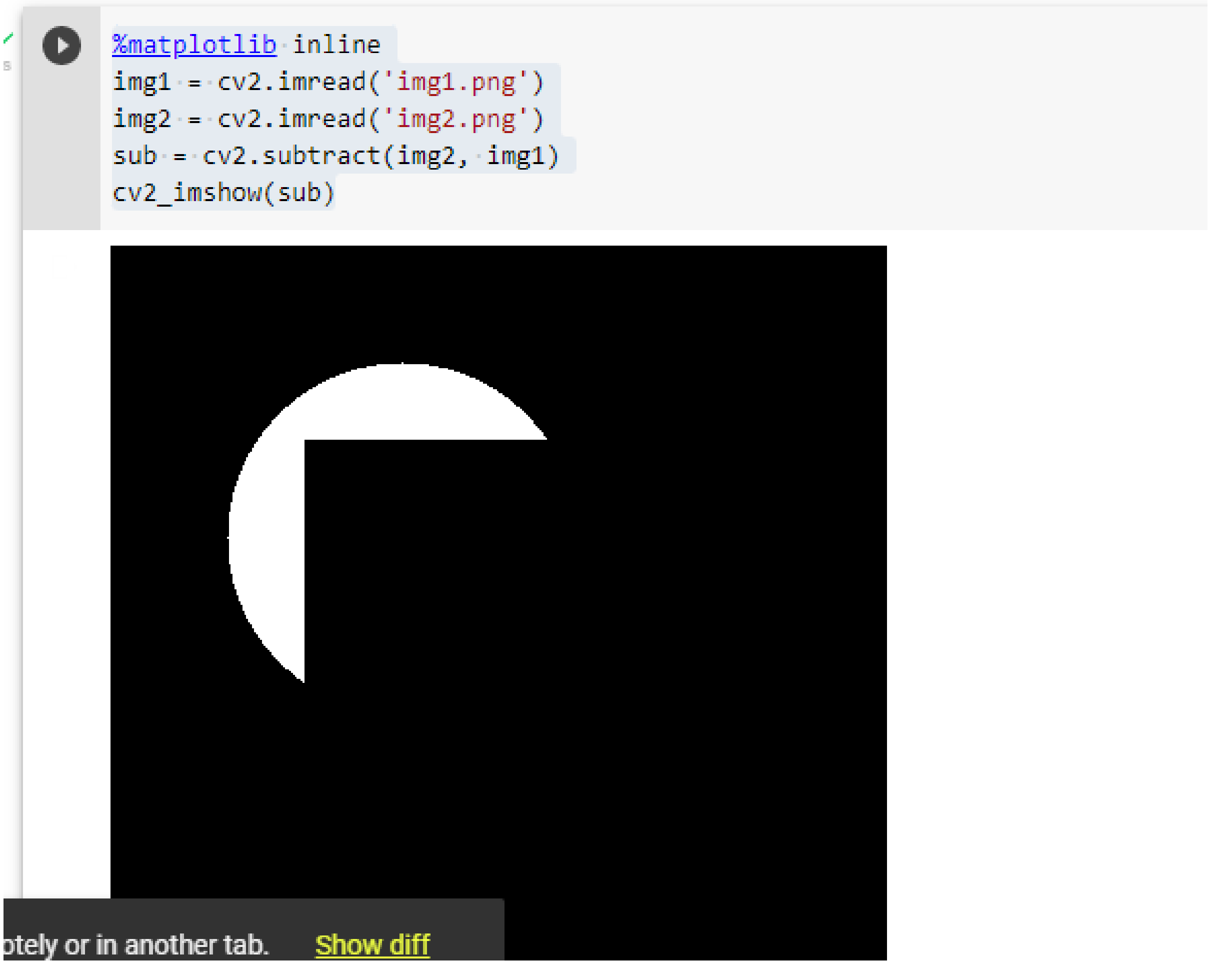
```
#Laplacian Filter
%matplotlib inline
img = cv2.imread('home.jfif', 0)

laplacian = cv2.Laplacian(img, cv2.CV_64F)
sobelx = cv2.Sobel(img, cv2.CV_64F, 1, 0, ksize = 5)
sobely = cv2.Sobel(img, cv2.CV_64F, 0, 1, ksize = 5)

plt.subplot(2,2,1), plt.imshow(img, cmap = 'gray')
plt.title('Original'), plt.xticks([]), plt.yticks([])

plt.subplot(2,2,2), plt.imshow(img, cmap = 'gray')
plt.title('Laplacian'), plt.xticks([]), plt.yticks([])

plt.subplot(2,2,3), plt.imshow(img, cmap = 'gray')
plt.title('Sobel X'), plt.xticks([]), plt.yticks([])

plt.subplot(2,2,4), plt.imshow(img, cmap = 'gray')
plt.title('Sobel Y'), plt.xticks([]), plt.yticks([])

plt.show()
```

02_GAURAV BANE



```
plt.subplot(2,2,3), plt.imshow(img, cmap = 'gray')
plt.title('Sobel X'), plt.xticks([]), plt.yticks([])

plt.subplot(2,2,4), plt.imshow(img, cmap = 'gray')
plt.title('Sobel Y'), plt.xticks([]), plt.yticks([])

plt.show()
```

```
%matplotlib inline
img1 = cv2.imread('img1.png')
img2 = cv2.imread('img2.png')
sub = cv2.subtract(img2, img1)
cv2_imshow(sub)
```



```
%matplotlib inline
img1 = cv2.imread('img1.png')
img2 = cv2.imread('img2.png')
sub = cv2.subtract(img2, img1)
cv2_imshow(sub)
```