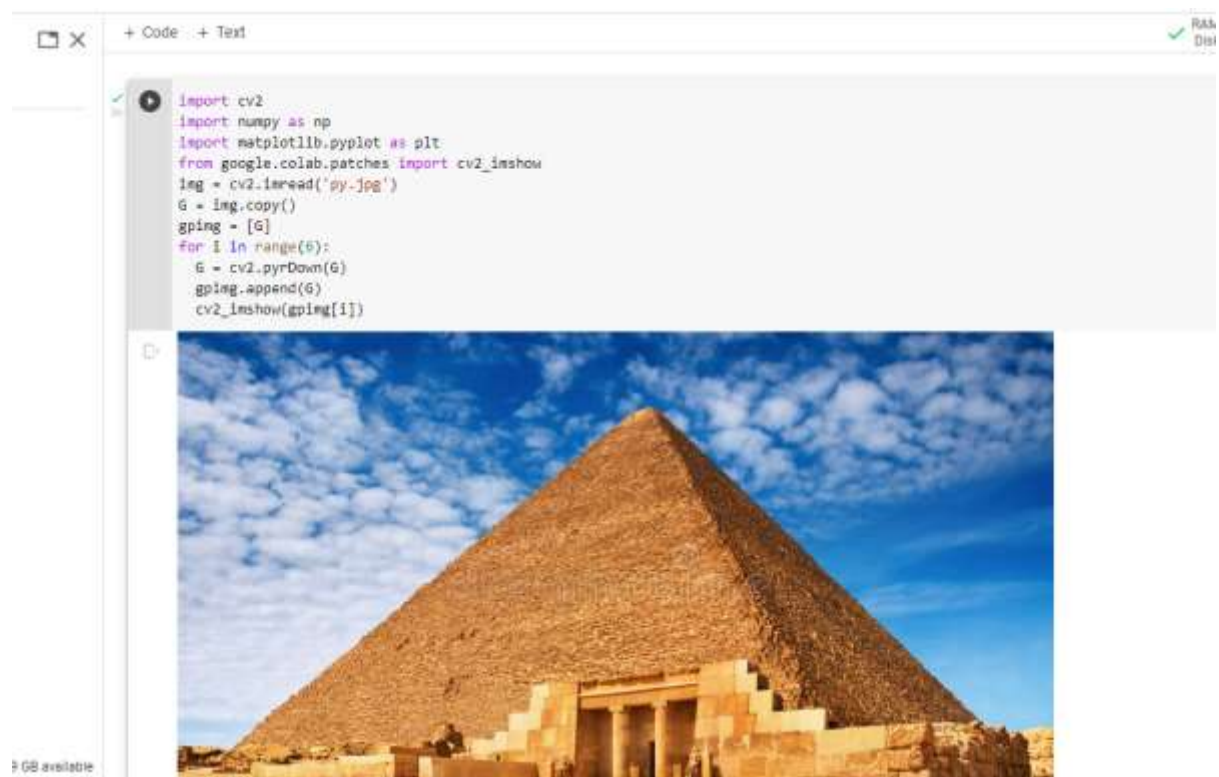


GAURAV BANE  
ROLL NO : 02

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from google.colab.patches import cv2_imshow
img = cv2.imread('img1.jpg')
G = img.copy()
gpimg = [G]
for i in range(6):
    G = cv2.pyrDown(G)
    gpimg.append(G)
    cv2_imshow(gpimg[i])
```



GAURAV BANE  
ROLL NO : 02




```
lpimg = [gpimg[5]]
cv2_imshow(lpimg[0])
for i in range(6,0,-1):
    GE=cv2.pyrUp(gpimg[i])
    GE=cv2.resize(GE,gpimg[i-1].shape[-2::-1])
    L=cv2.subtract(gpimg[i-1],GE)
    lpimg.append(L)
for i in range(6,0,-1):
    cv2_imshow(lpimg[i])
```

GAURAV BANE  
ROLL NO : 02

+ Code + Text

```
[3] lpimg = [gpimg[5]]
cv2_imshow(lpimg[0])
for i in range(6,0,-1):
    GE=cv2.pyrUp(gpimg[i])
    GE=cv2.resize(GE,gpimg[i-1].shape[-2::-1])
    L=cv2.subtract(gpimg[i-1],GE)
    lpimg.append(L)
for i in range(6,0,-1):
    cv2_imshow(lpimg[i])
```



completed at 3:11 PM

+ Code + Text

```
[3]
```



GAURAV BANE

ROLL NO : 02

```
A = cv2.imread('apple.jfif')
B = cv2.imread('mango.jpg')

#generate Gaussian pyramid for A
G = A.copy()
gpA = [G]
for i in range(6):
    G = cv2.pyrDown(G)
    gpA.append(G)

#generate Gaussian pyramid for B

G = B.copy()
gpB = [G]
for i in range(6):
    G = cv2.pyrDown(G)
    gpB.append(G)

lpA = [gpA[5]]
for i in range(6, 0, -1):
    GE = cv2.pyrUp(gpA[i])
    GE = cv2.resize(GE, gpA[i - 1].shape[-2::-1])
    L = cv2.subtract(gpA[i - 1], GE)
    lpA.append(L)

#Generate Laplacian Pyramid for B
lpB = [gpB[5]]
for i in range(6, 0, -1):
    GE = cv2.pyrUp(gpB[i])
    GE = cv2.resize(GE, gpB[i - 1].shape[-2::-1])
    L = cv2.subtract(gpB[i - 1], GE)
    lpB.append(L)

#Now add left and right halves of images in each level
LS = []
lpAc = []
for i in range(len(lpA)):
    b = cv2.resize(lpA[i], lpB[i].shape[-2::-1])
    lpAc.append(b)

j = 0
for i in zip(lpAc, lpB):
    la, lb = i
    rows, cols, dpt = la.shape
    ls = np.hstack((la[:, 0:cols//2], lb[:, cols//2:]))
    j = j + 1
    LS.append(ls)
```

GAURAV BANE  
ROLL NO : 02

```
ls_ = LS[0]
for i in range(1,6):
    ls_ = cv2.pyrUp(ls_)
    ls_ = cv2.resize(ls_, LS[i].shape[-2::-1])
    ls_ = cv2.add(ls_, LS[i])

#image with direct connecting each half
B = cv2.resize(B, A.shape[-2::-1])
real = np.hstack((A[:, : cols//2], B[:,cols//2:]))
cv2.imwrite('Pyramid_blending2.jpg', ls_)
cv2.imwrite('Direct_blending.jpg', real)
cv2_imshow(ls_)
cv2_imshow(real)
```

```
+ Code + Text

A = cv2.imread('apple.jpg')
B = cv2.imread('aango.jpg')

#generate Gaussian pyramid for A
G = A.copy()
gpA = [G]
for i in range(5):
    G = cv2.pyrDown(G)
    gpA.append(G)

#generate Gaussian pyramid for B
G = B.copy()
gpB = [G]
for i in range(5):
    G = cv2.pyrDown(G)
    gpB.append(G)

lpA = [gpA[5]]
for i in range(5, 0, -1):
    GE = cv2.pyrUp(gpA[i])
    GE = cv2.resize(GE, gpA[i-1].shape[-2::-1])
    I = cv2.subtract(gpA[i-1], GE)
    lpA.append(I)

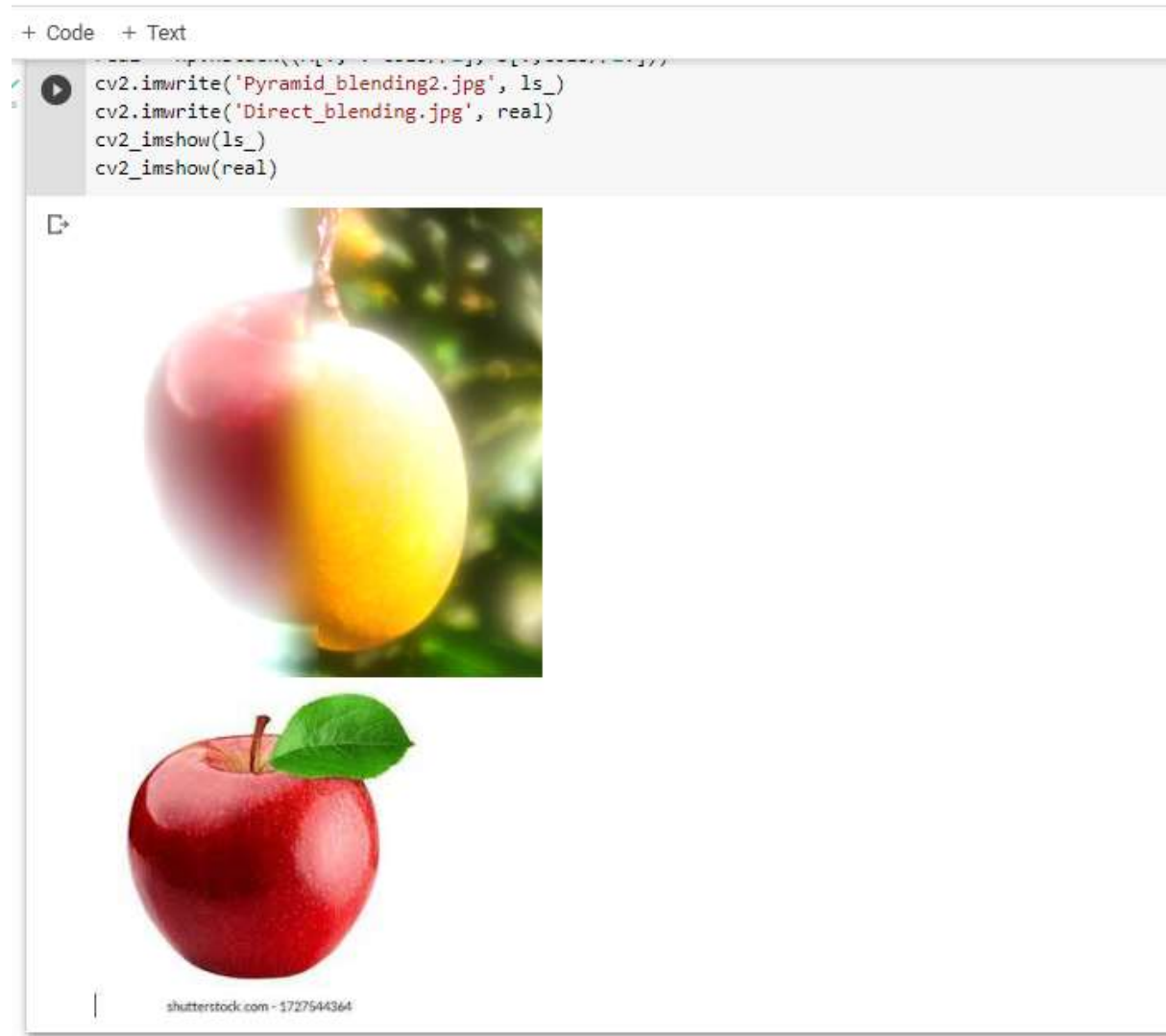
#Generate Laplacian Pyramid for B
lpB = [gpB[5]]
for i in range(5, 0, -1):
    GE = cv2.pyrUp(gpB[i])
    GE = cv2.resize(GE, gpB[i-1].shape[-2::-1])
    I = cv2.subtract(gpB[i-1], GE)
    lpB.append(I)

#Now add left and right halves of images in each level
ls = []
lpAc = []
for i in range(len(lpA)):
    b = cv2.resize(lpA[i], lpB[i].shape[-2::-1])
    lpAc.append(b)

j = 0
for i in zip(lpAc, lpB):
    la, lb = i
    rows, cols, dpt = la.shape
    ls = np.hstack((la[:,0:cols//2], lb[:, cols//2:]))
    j = j + 1
    ls.append(ls)

ls_ = ls[0]
for i in range(1,6):
    ls_ = cv2.pyrUp(ls_)
    ls_ = cv2.resize(ls_, ls[i].shape[-2::-1])
    ls_ = cv2.add(ls_, ls[i])

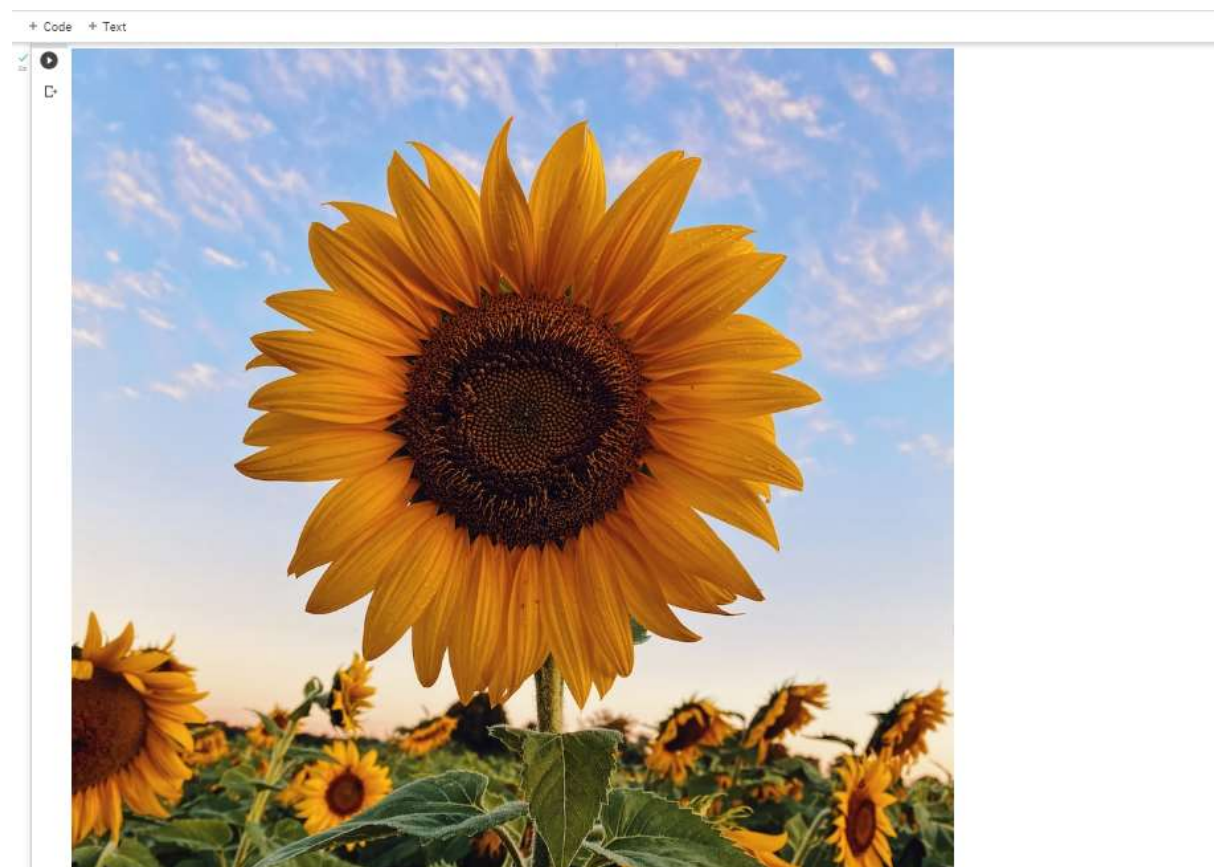
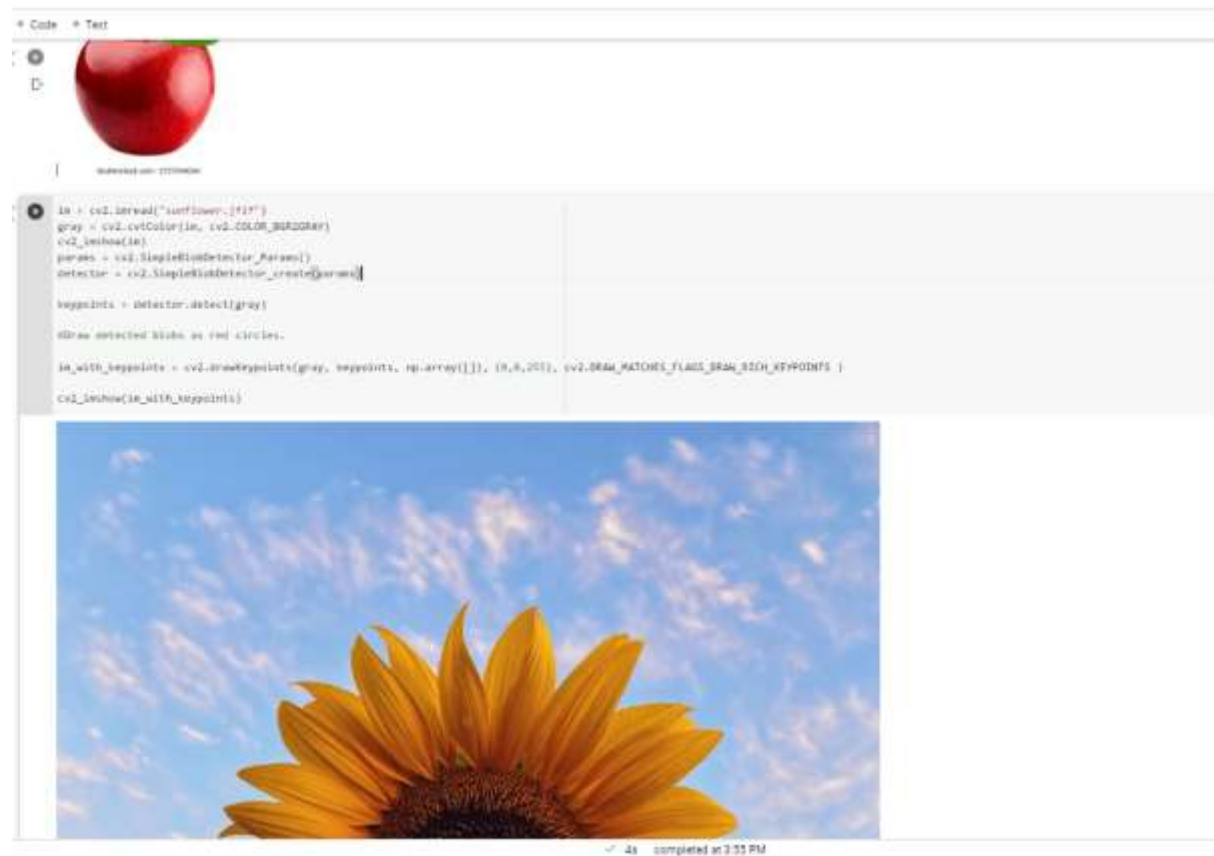
#image with direct connecting each half
B = cv2.resize(B, A.shape[-2::-1])
real = np.hstack((A[:, : cols//2], B[:,cols//2:]))
cv2.imwrite('Pyramid_blending2.jpg', ls_)
cv2.imwrite('Direct_blending.jpg', real)
cv2_imshow(ls_)
cv2_imshow(real)
```



```
im = cv2.imread("sunflower.jfif")  
gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)  
cv2.imshow(im)  
params = cv2.SimpleBlobDetector_Params()  
detector = cv2.SimpleBlobDetector_create(params)  
  
keypoints = detector.detect(gray)  
  
#Draw detected blobs as red circles.  
  
im_with_keypoints = cv2.drawKeypoints(gray, keypoints, np.array([]), (0  
,0,255), cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS )  
  
cv2.imshow(im_with_keypoints)
```



GAURAV BANE  
ROLL NO : 02



GAURAV BANE  
ROLL NO : 02

