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()

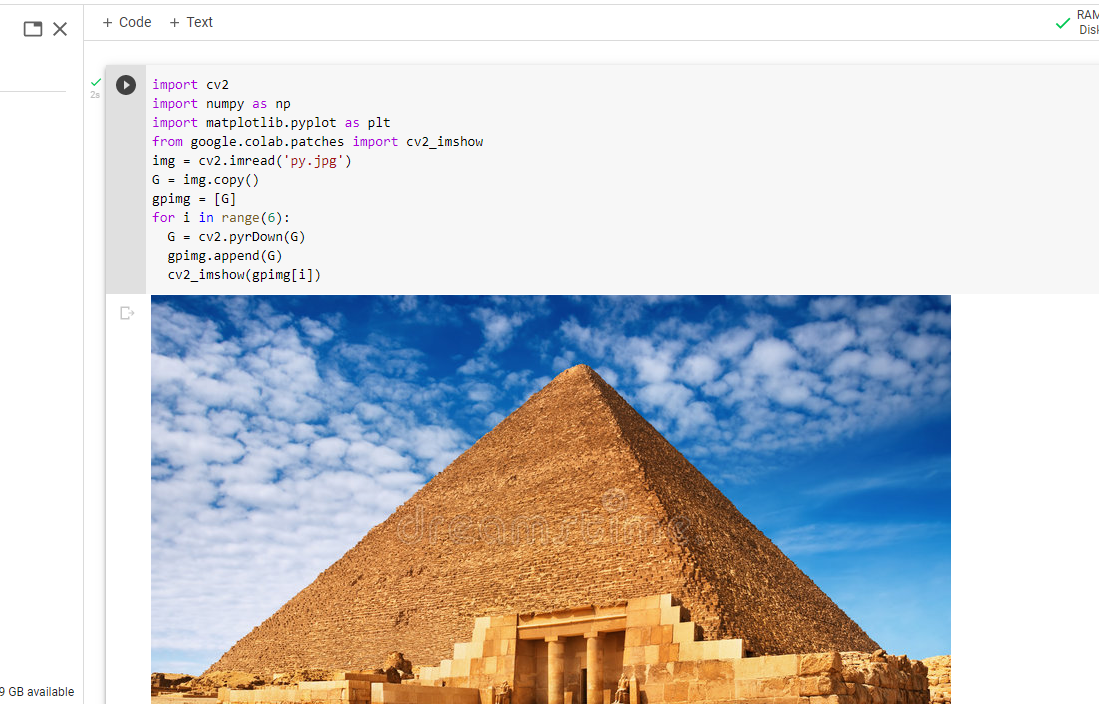
gpimg = [G]

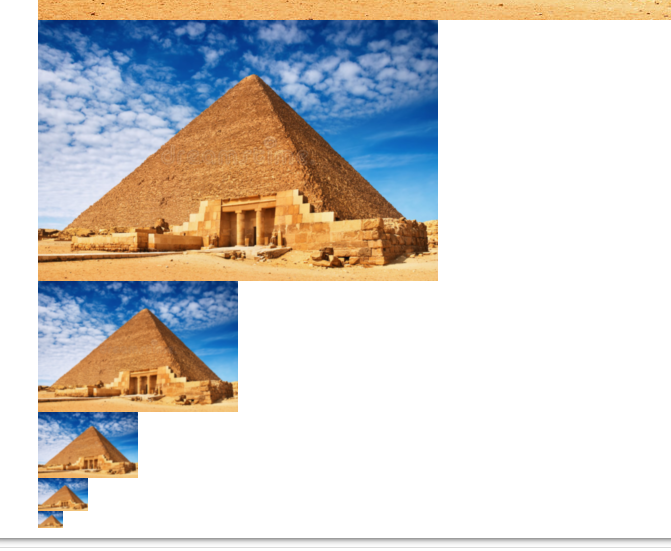
for i in range(6):

  G = cv2.pyrDown(G)

  gpimg.append(G)

  cv2\_imshow(gpimg[i])





lpimg = [gpimg[5]]

cv2\_imshow(lpimg[0])

for i in range(6,0,-1):

  GE=cv2.pyrUp(gpimg[i])

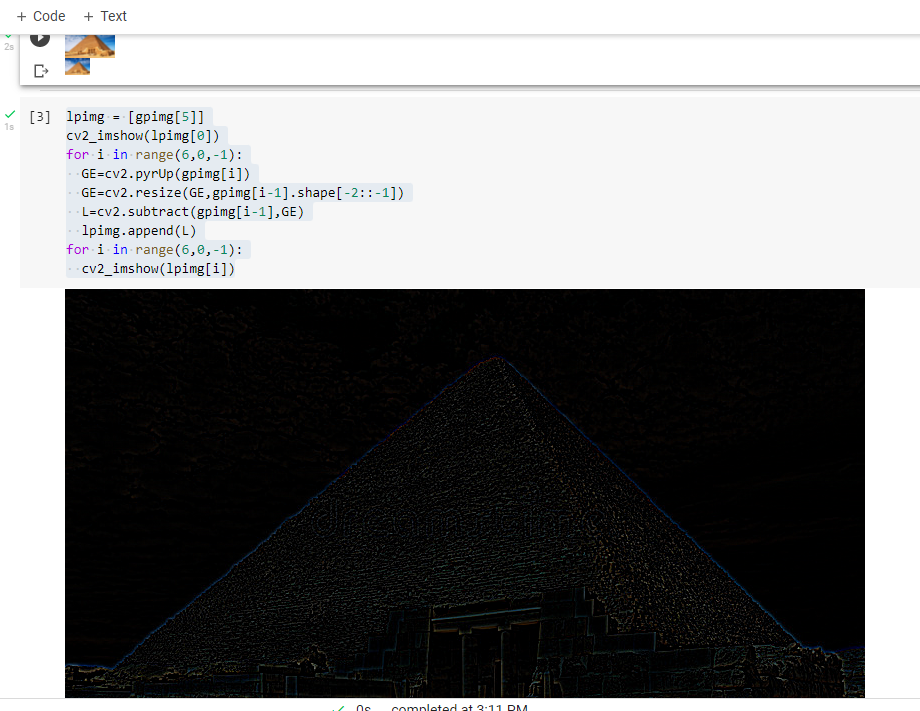
  GE=cv2.resize(GE,gpimg[i-1].shape[-2::-1])

  L=cv2.subtract(gpimg[i-1],GE)

  lpimg.append(L)

for i in range(6,0,-1):

  cv2\_imshow(lpimg[i])



A picture containing graphical user interface

Description automatically generated

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)

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:]))

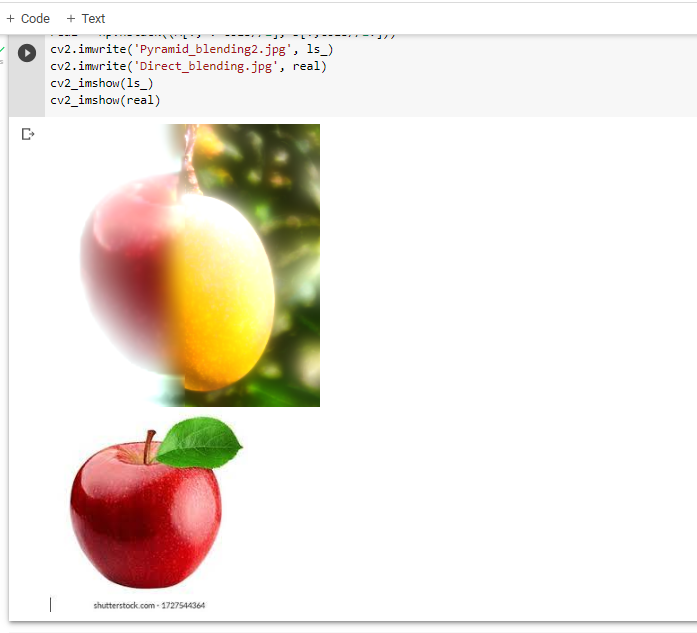
cv2.imwrite('Pyramid\_blending2.jpg', ls\_)

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)

