**Python with Data Science**

! git clone https://github.com/hariharan849/DeepLearning.git

**#just showing selected img**

from google.colab.patches import cv2\_imshow

import cv2

image = cv2.imread("/content/bgr.jpg")

cv2\_imshow(image)



**#just showing selected img also in black N white**

from google.colab.patches import cv2\_imshow

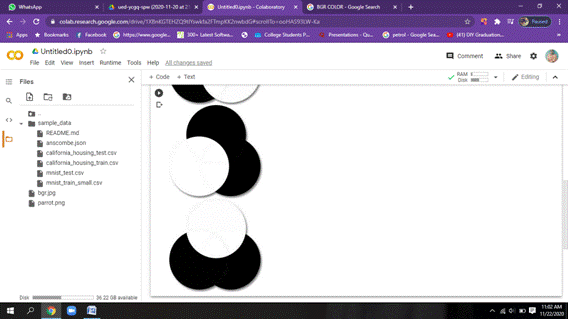
import cv2

image = cv2.imread("/content/bgr.jpg")

image1 = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

cv2\_imshow(image)

cv2\_imshow(image1)



**#just water image or reverse img(up &down)**

from google.colab.patches import cv2\_imshow

import cv2

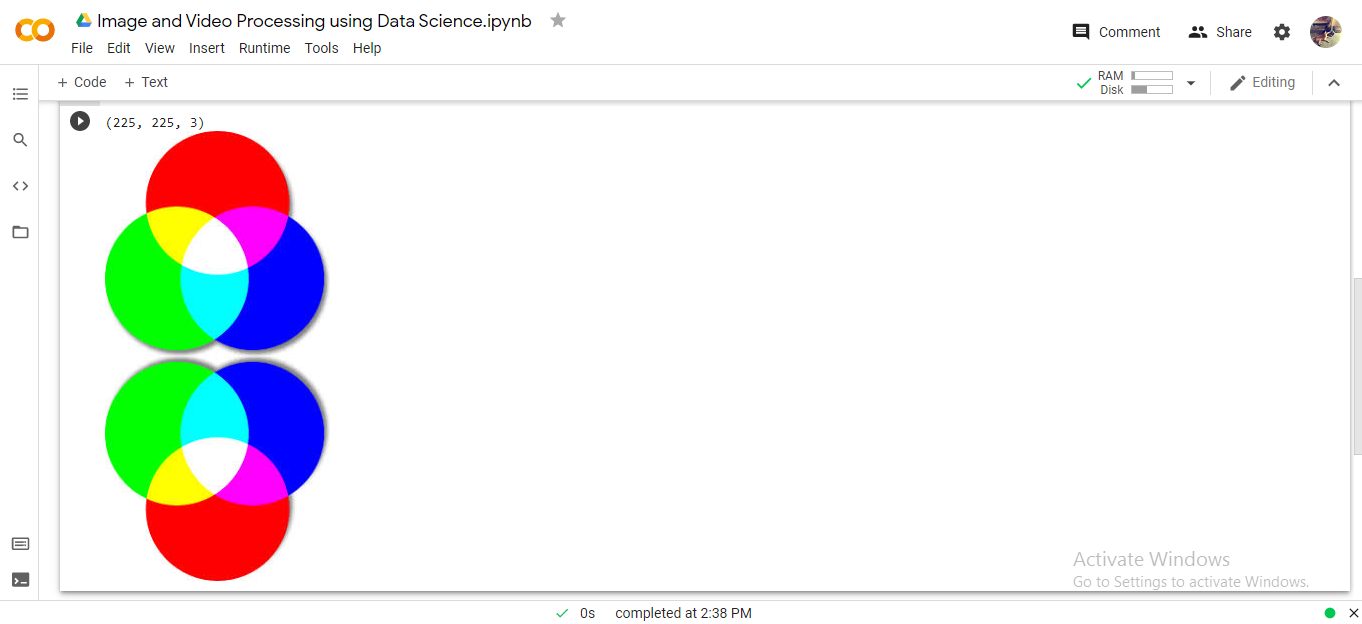
image = cv2.imread("/content/parrot.png")

image2 = cv2.flip(image,0)

print (image.shape)

cv2\_imshow(image)

cv2\_imshow(image2)



**#for mirror image(right & left)**

from google.colab.patches import cv2\_imshow

import cv2

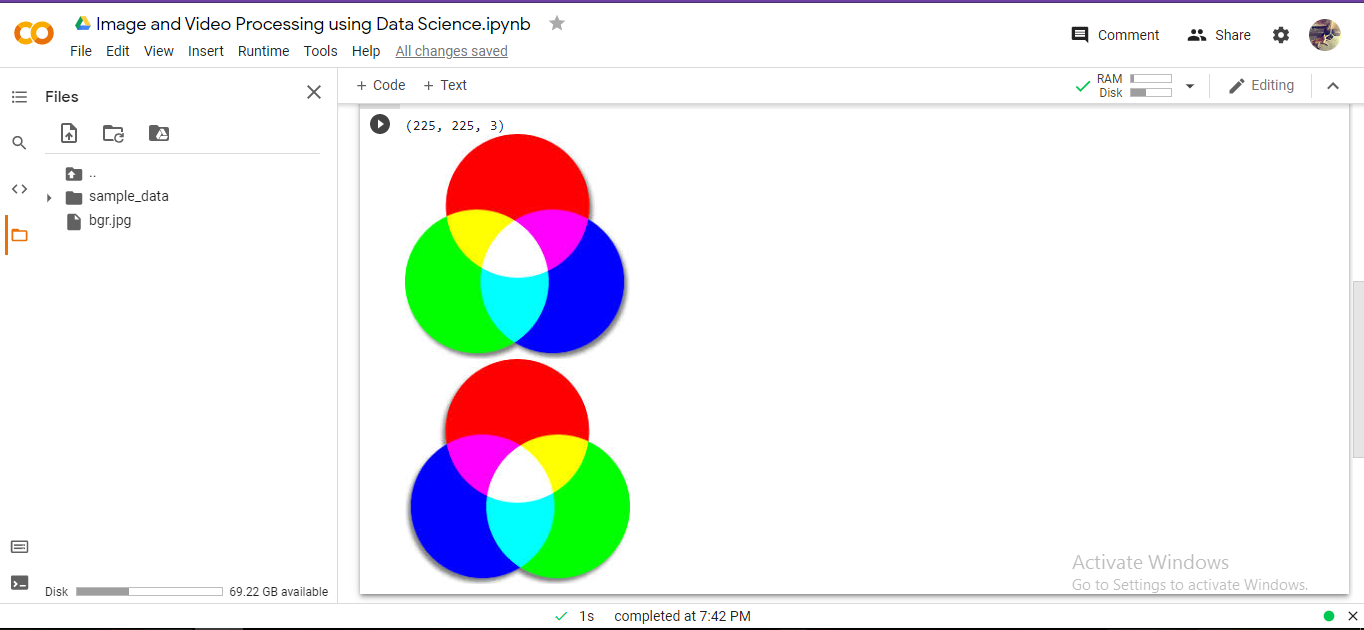
image = cv2.imread("/content/parrot.png")

image2 = cv2.flip(image,1)

print (image.shape)

cv2\_imshow(image)

cv2\_imshow(image2)



**#for left right n also up & down(both)**

from google.colab.patches import cv2\_imshow

import cv2

image = cv2.imread("/content/parrot.png")

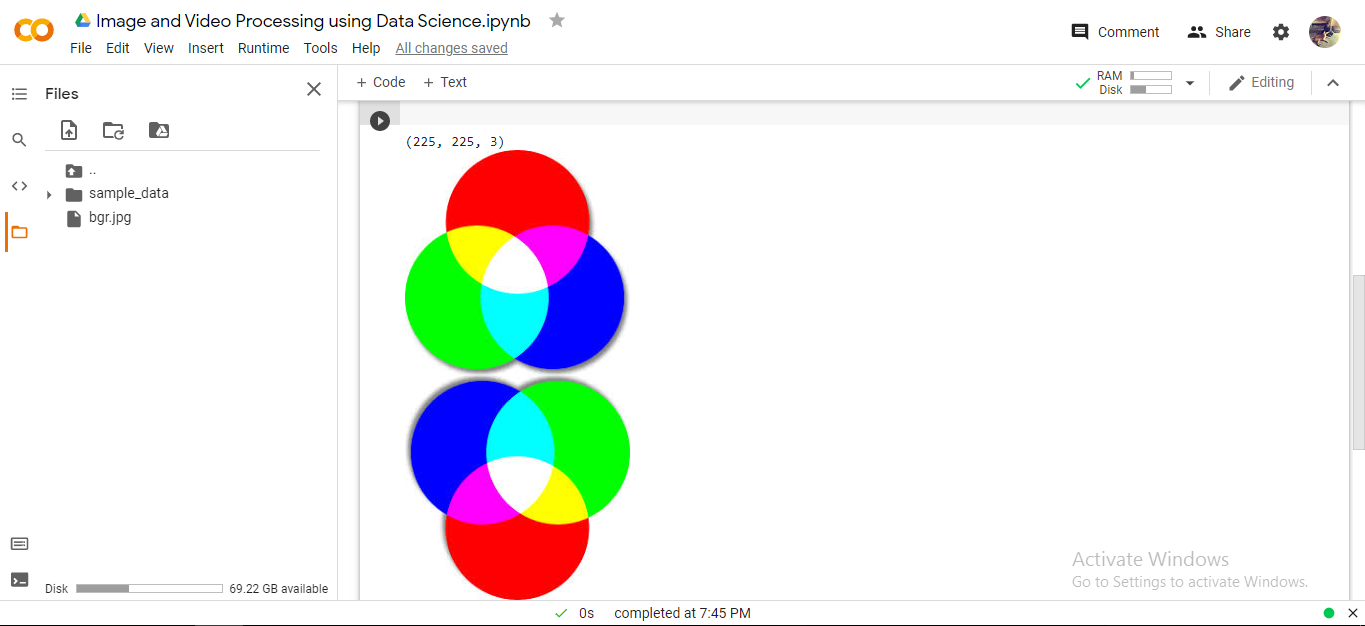
image2 = cv2.flip(image,-1)

print (image.shape)

#prints in matrix form(row,col,colorspace)

cv2\_imshow(image)

cv2\_imshow(image2)



**#for grey image**

from google.colab.patches import cv2\_imshow

import cv2

image = cv2.imread("/content/parrot.png", 0)

print(image.shape)

#only(row,col)is printed since color space is made to grey

cv2\_imshow(image)

****

**#bgr**

from google.colab.patches import cv2\_imshow

import cv2

image = cv2.imread("/content/bgr.jpg")

cv2\_imshow(image)

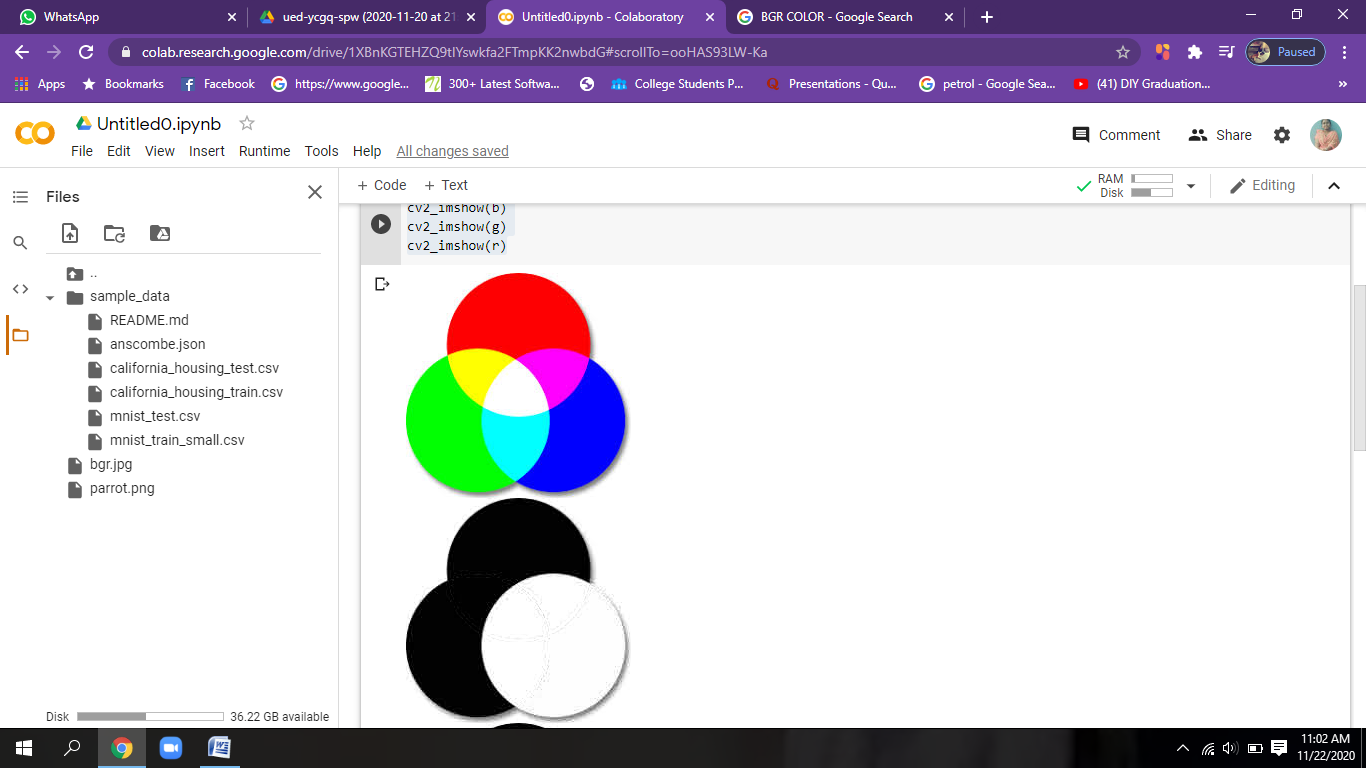
(b, g, r)= cv2.split(image)

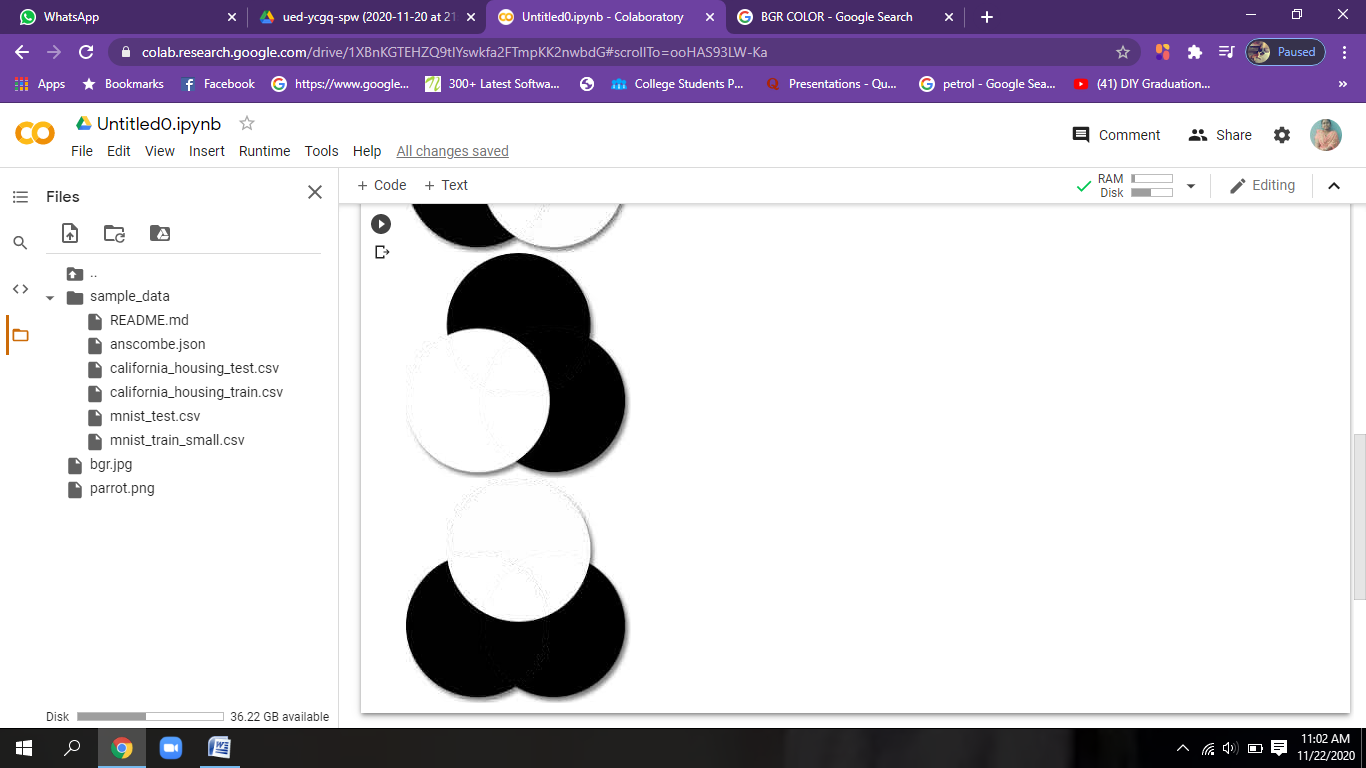
cv2\_imshow(b)

cv2\_imshow(g)

cv2\_imshow(r)

(for example scroll below)





**#back to merge into a single pic**

from google.colab.patches import cv2\_imshow

import cv2

image = cv2.imread("/content/bgr.jpg")

cv2\_imshow(image)

image1 = cv2.merge((b,g,r))

cv2\_imshow(image1)

**#bgr to gray**

from google.colab.patches import cv2\_imshow

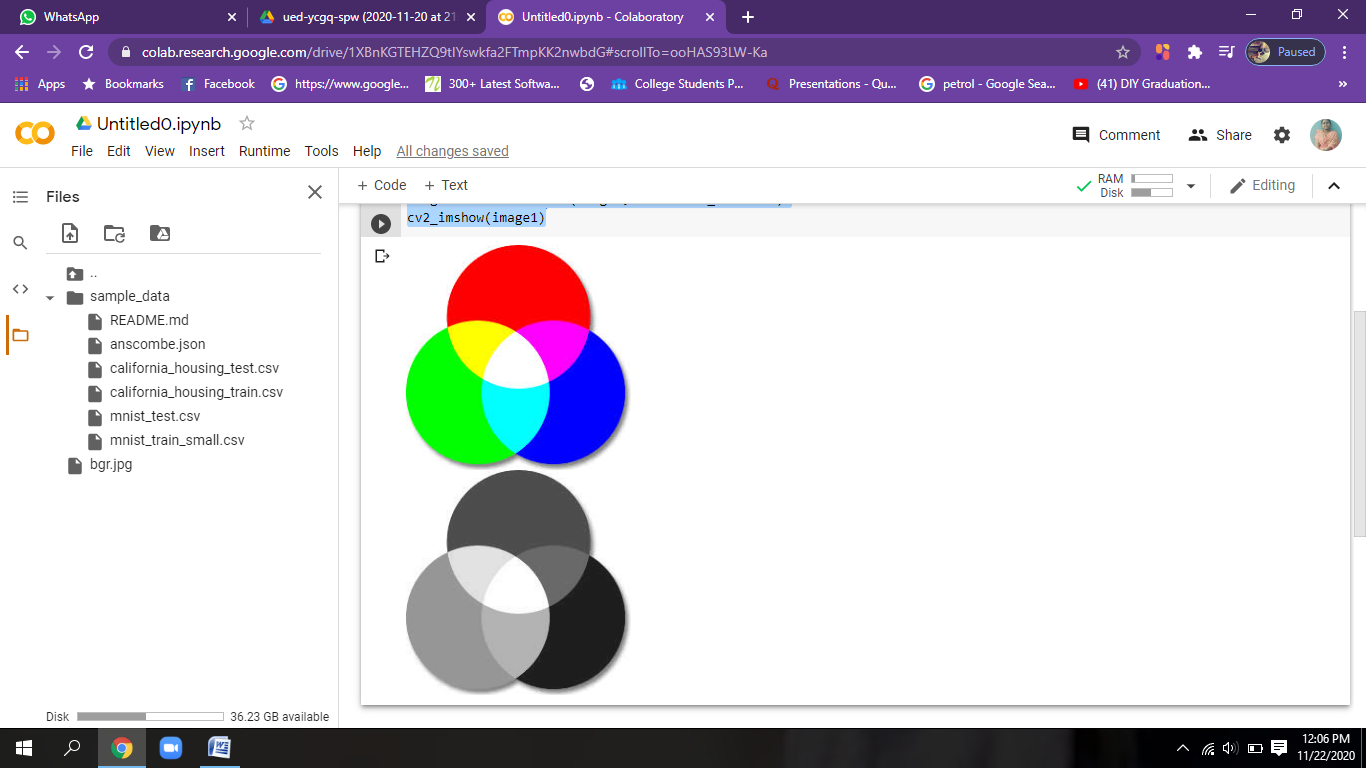
import cv2

image = cv2.imread("/content/bgr.jpg")

cv2\_imshow(image)

image1 = cv2.cvtColor(image , cv2.COLOR\_BGR2GRAY)

cv2\_imshow(image1)



**#bgr to HSV**

from google.colab.patches import cv2\_imshow

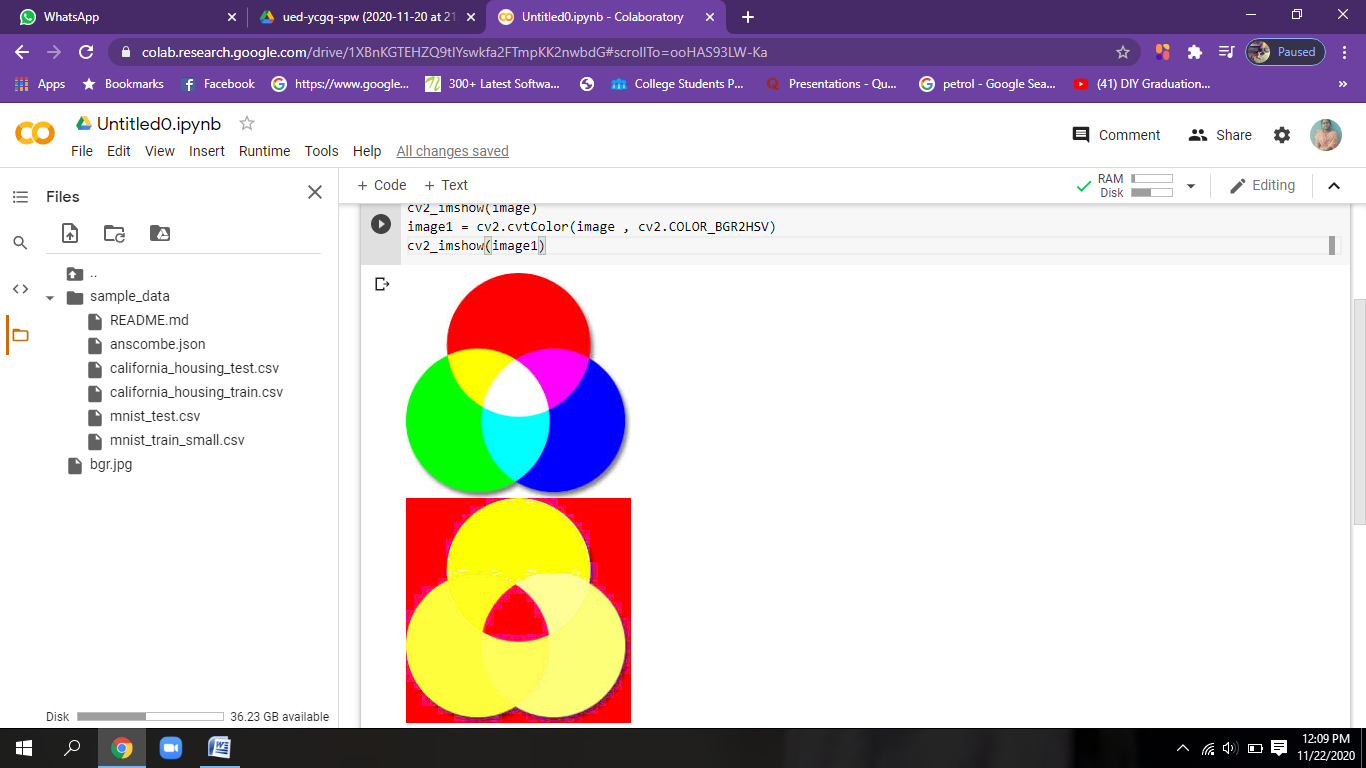
import cv2

image = cv2.imread("/content/bgr.jpg")

cv2\_imshow(image)

image1 = cv2.cvtColor(image , cv2.COLOR\_BGR2HSV)

cv2\_imshow(image1)



**#saves an image to a specified file**

from google.colab.patches import cv2\_imshow

import cv2

image = cv2.imread("/content/bgr.jpg")

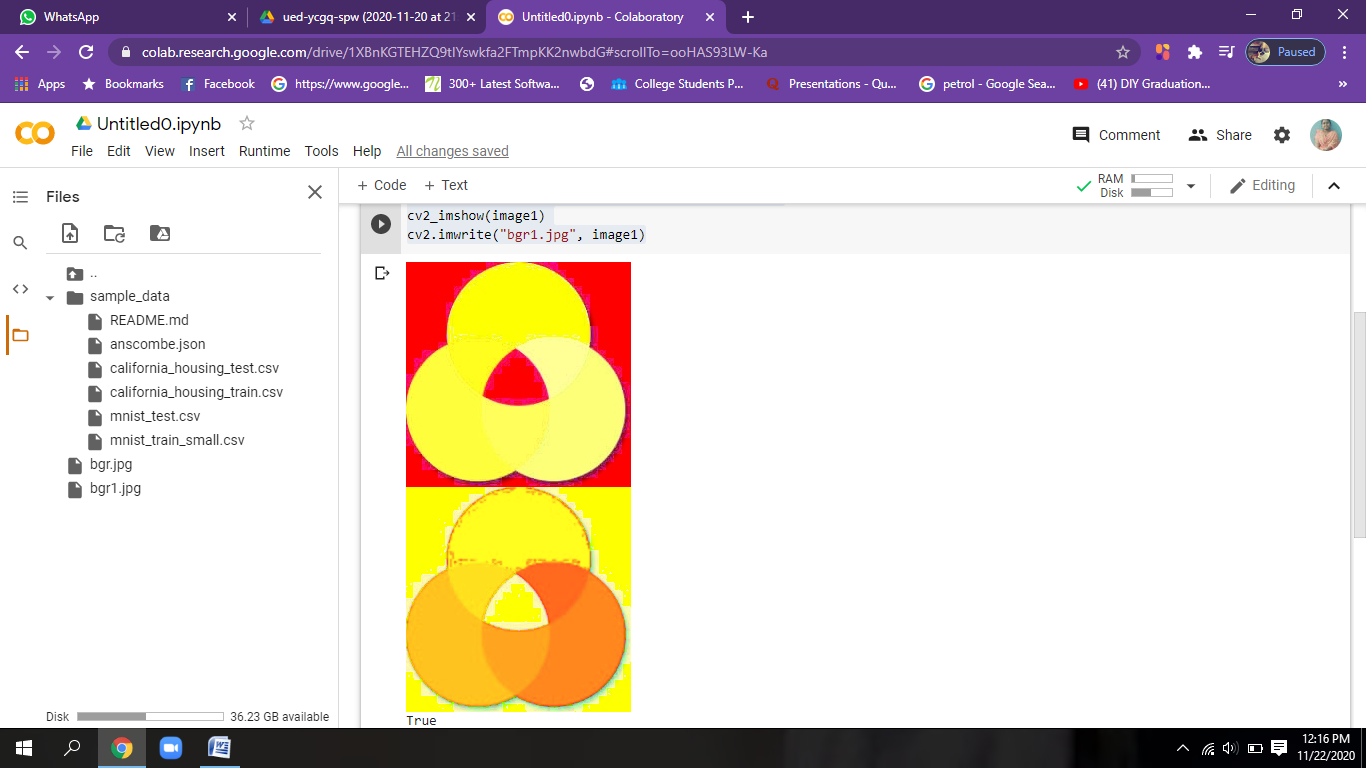
cv2\_imshow(image)

image1 = cv2.cvtColor(image , cv2.COLOR\_BGR2HSV)

cv2\_imshow(image1)

cv2.imwrite("bgr1.jpg", image1)

(refer img down, bgr1 location is created as specified in imwrite)



**#resize image**

from google.colab.patches import cv2\_imshow

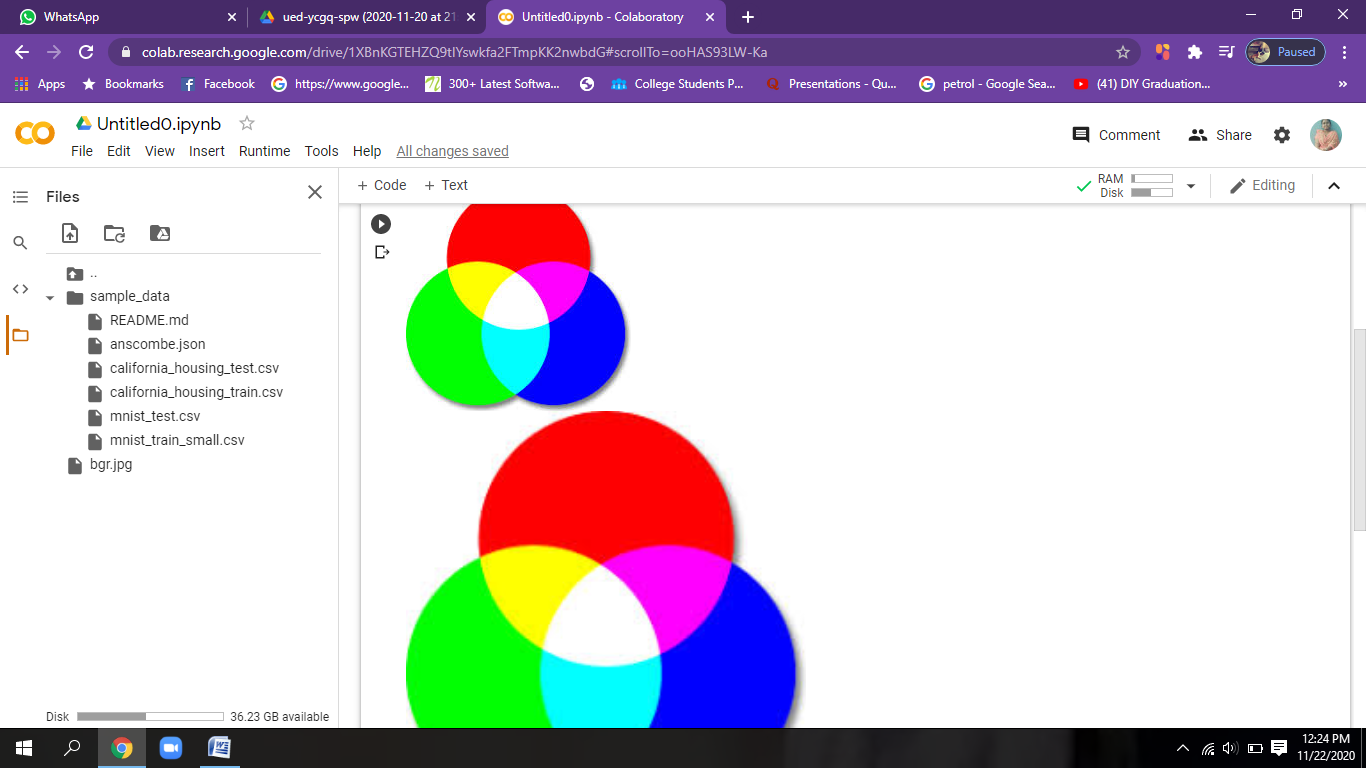
import cv2

image = cv2.imread("/content/bgr.jpg")

image1 = cv2.resize(image,(400,400), interpolation = cv2.INTER\_LINEAR)

cv2\_imshow(image)

cv2\_imshow(image1)



**#image transation(moving via width and height)**

from google.colab.patches import cv2\_imshow

import cv2

import numpy as np

image = cv2.imread("/content/bgr.jpg")

(rows, cols, channels) = image.shape

print(image.shape)

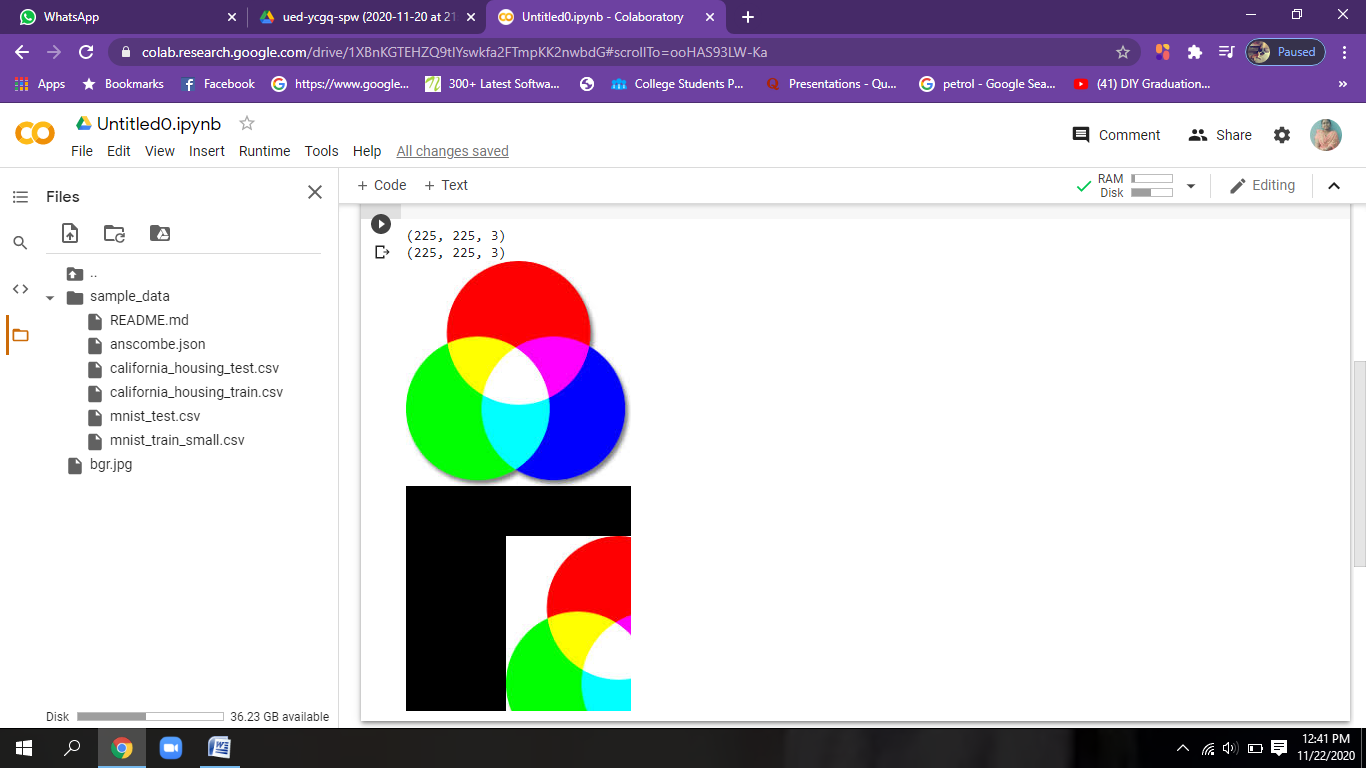
M = np.float32([[1,0,100],[0,1,50]])

image1 = cv2.warpAffine(image, M, (cols,rows))

cv2\_imshow(image)

cv2\_imshow(image1)

(NOTE:height-50;width-100)



**#flip image by angle**

from google.colab.patches import cv2\_imshow

import cv2

import numpy as np

image = cv2.imread("/content/bgr.jpg")

(rows, cols, channels) = image.shape

print(image.shape)

M = cv2.getRotationMatrix2D((cols/2,rows/2), 45, 1)

image1 = cv2.warpAffine(image, M, (cols,rows))

cv2\_imshow(image)

cv2\_imshow(image1)

(NOTE:image is rotated by 45 degree)

