OBJECT DETECTION

To Perform object Detection on Images

#import cv2

- We Have Imported Required Libraries like Numpy and cv2
- Numpy is used for Numerical Operations
- Cv2 is used for image processing

#def color_prediction(n):

if n == 0:

name = "Orange"

lowerHSV = (0, 100, 100)

upperHSV = (115, 255, 255)

return name, lowerHSV, upperHSV

#if n == 1:

name = "Apple"

lowerHSV = (0, 100, 100)

upperHSV = (10, 255, 255)

return name, lowerHSV, upperHSV

- This line is used to define the color if n==0 means that color is come to under (0,100,100) and upper limit is (115,255,255) this color should be Orange.
- color if n==1 means that color is come to under (0,100,100) and upper limit is (10,255,255) this color should be Red

#image = cv2.imread("/content/object.jpg")

- This Reads an Image File using opencv imread function and stores in img

#HSVimage= cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

- the code reads the "object.jpg" image and converts it to the HSV color space using cv2.cvtColor().
- If it is, it displays the original image and the HSV image once and sets the flag to True, so the images are not displayed again in subsequent iterations

#rects = {}

- Create a empty list to append the boundaries of the image
- This code is used to detect the area of given image and detect the boundaries of given image

#for i in range(2):

name, lowerHSV, upperHSV = color_prediction(i)

mask = cv2.inRange(HSVimage, lowerHSV, upperHSV)

contours, _ = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)

- This code is used to define the Mask function in range of HSVimage and Upper , lower HSV image
- To define the contours function using key word cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
- Cv2.RETR_EXTERNAL used to detect the outer boundary of object in image
- Cv2.CHAIN_APPROX_NONE used to store the accurate value of contours points

 # if len(contours) > 0:

biggest = sorted(contours, key=cv2.contourArea, reverse=True)[0]

rects = cv2.boundingRect(biggest)

x, y, w, h = rects

cv2.rectangle(image, (x, y), (x+w, y+h), (255, 0, 0), 2)

cv2.putText(image, name, (x, y), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 2)

- This code is used to predict the contours area using ContourArea keyword
- **cv2.boundingRect** This function used to create an approximate rectangle along with the image.
- And we create a blue rectangle boundary box for given object
- We put text for the object to predict the orange and apple

#cv2_imshow(image)

cv2.waitKey(0)

cv2.destroyAllWindows()

- Showing the image using the keyword cv2.imshow
- Finally, all OpenCV windows are closed when the loop exits, ending the program.