Hay más de 150 métodos de conversión de espacios de color disponibles en OpenCV. Pero veremos sólo dos, que son los más utilizados: BGR↔Gris y BGR↔VHS.

Para la conversión de color, utilizamos la función cv.cvtColor(input\_image, flag) donde flag determina el tipo de conversión.

Para BGR→Conversión de grises, usamos la bandera cv.COLOR\_BGR2GRAY . Lo mismo para BGR→HSV, usamos la bandera cv.COLOR\_BGR2HSV . Para obtener otras banderas, simplemente ejecute los siguientes comandos en su terminal Python:

import cv2 as cv  
flags = [i for i in dir(cv) if i.startswith('COLOR\_')]  
print( flags )

['COLOR\_BAYER\_BG2BGR', 'COLOR\_BAYER\_BG2BGRA', 'COLOR\_BAYER\_BG2BGR\_EA', 'COLOR\_BAYER\_BG2BGR\_VNG', 'COLOR\_BAYER\_BG2GRAY', 'COLOR\_BAYER\_BG2RGB', 'COLOR\_BAYER\_BG2RGBA', 'COLOR\_BAYER\_BG2RGB\_EA', 'COLOR\_BAYER\_BG2RGB\_VNG', 'COLOR\_BAYER\_BGGR2BGR', 'COLOR\_BAYER\_BGGR2BGRA', 'COLOR\_BAYER\_BGGR2BGR\_EA', 'COLOR\_BAYER\_BGGR2BGR\_VNG', 'COLOR\_BAYER\_BGGR2GRAY', 'COLOR\_BAYER\_BGGR2RGB', 'COLOR\_BAYER\_BGGR2RGBA', 'COLOR\_BAYER\_BGGR2RGB\_EA', 'COLOR\_BAYER\_BGGR2RGB\_VNG', 'COLOR\_BAYER\_GB2BGR', 'COLOR\_BAYER\_GB2BGRA', 'COLOR\_BAYER\_GB2BGR\_EA', 'COLOR\_BAYER\_GB2BGR\_VNG', 'COLOR\_BAYER\_GB2GRAY', 'COLOR\_BAYER\_GB2RGB', 'COLOR\_BAYER\_GB2RGBA', 'COLOR\_BAYER\_GB2RGB\_EA', 'COLOR\_BAYER\_GB2RGB\_VNG', 'COLOR\_BAYER\_GBRG2BGR', 'COLOR\_BAYER\_GBRG2BGRA', 'COLOR\_BAYER\_GBRG2BGR\_EA', 'COLOR\_BAYER\_GBRG2BGR\_VNG', 'COLOR\_BAYER\_GBRG2GRAY', 'COLOR\_BAYER\_GBRG2RGB', 'COLOR\_BAYER\_GBRG2RGBA', 'COLOR\_BAYER\_GBRG2RGB\_EA', 'COLOR\_BAYER\_GBRG2RGB\_VNG', 'COLOR\_BAYER\_GR2BGR', 'COLOR\_BAYER\_GR2BGRA', 'COLOR\_BAYER\_GR2BGR\_EA', 'COLOR\_BAYER\_GR2BGR\_VNG', 'COLOR\_BAYER\_GR2GRAY', 'COLOR\_BAYER\_GR2RGB', 'COLOR\_BAYER\_GR2RGBA', 'COLOR\_BAYER\_GR2RGB\_EA', 'COLOR\_BAYER\_GR2RGB\_VNG', 'COLOR\_BAYER\_GRBG2BGR', 'COLOR\_BAYER\_GRBG2BGRA', 'COLOR\_BAYER\_GRBG2BGR\_EA', 'COLOR\_BAYER\_GRBG2BGR\_VNG', 'COLOR\_BAYER\_GRBG2GRAY', 'COLOR\_BAYER\_GRBG2RGB', 'COLOR\_BAYER\_GRBG2RGBA', 'COLOR\_BAYER\_GRBG2RGB\_EA', 'COLOR\_BAYER\_GRBG2RGB\_VNG', 'COLOR\_BAYER\_RG2BGR', 'COLOR\_BAYER\_RG2BGRA', 'COLOR\_BAYER\_RG2BGR\_EA', 'COLOR\_BAYER\_RG2BGR\_VNG', 'COLOR\_BAYER\_RG2GRAY', 'COLOR\_BAYER\_RG2RGB', 'COLOR\_BAYER\_RG2RGBA', 'COLOR\_BAYER\_RG2RGB\_EA', 'COLOR\_BAYER\_RG2RGB\_VNG', 'COLOR\_BAYER\_RGGB2BGR', 'COLOR\_BAYER\_RGGB2BGRA', 'COLOR\_BAYER\_RGGB2BGR\_EA', 'COLOR\_BAYER\_RGGB2BGR\_VNG', 'COLOR\_BAYER\_RGGB2GRAY', 'COLOR\_BAYER\_RGGB2RGB', 'COLOR\_BAYER\_RGGB2RGBA', 'COLOR\_BAYER\_RGGB2RGB\_EA', 'COLOR\_BAYER\_RGGB2RGB\_VNG', 'COLOR\_BGR2BGR555', 'COLOR\_BGR2BGR565', 'COLOR\_BGR2BGRA', 'COLOR\_BGR2GRAY', 'COLOR\_BGR2HLS', 'COLOR\_BGR2HLS\_FULL', 'COLOR\_BGR2HSV', 'COLOR\_BGR2HSV\_FULL', 'COLOR\_BGR2LAB', 'COLOR\_BGR2LUV', 'COLOR\_BGR2Lab', 'COLOR\_BGR2Luv', 'COLOR\_BGR2RGB', 'COLOR\_BGR2RGBA', 'COLOR\_BGR2XYZ', 'COLOR\_BGR2YCR\_CB', 'COLOR\_BGR2YCrCb', 'COLOR\_BGR2YUV', 'COLOR\_BGR2YUV\_I420', 'COLOR\_BGR2YUV\_IYUV', 'COLOR\_BGR2YUV\_YV12', 'COLOR\_BGR5552BGR', 'COLOR\_BGR5552BGRA', 'COLOR\_BGR5552GRAY', 'COLOR\_BGR5552RGB', 'COLOR\_BGR5552RGBA', 'COLOR\_BGR5652BGR', 'COLOR\_BGR5652BGRA', 'COLOR\_BGR5652GRAY', 'COLOR\_BGR5652RGB', 'COLOR\_BGR5652RGBA', 'COLOR\_BGRA2BGR', 'COLOR\_BGRA2BGR555', 'COLOR\_BGRA2BGR565', 'COLOR\_BGRA2GRAY', 'COLOR\_BGRA2RGB', 'COLOR\_BGRA2RGBA', 'COLOR\_BGRA2YUV\_I420', 'COLOR\_BGRA2YUV\_IYUV', 'COLOR\_BGRA2YUV\_YV12', 'COLOR\_BayerBG2BGR', 'COLOR\_BayerBG2BGRA', 'COLOR\_BayerBG2BGR\_EA', 'COLOR\_BayerBG2BGR\_VNG', 'COLOR\_BayerBG2GRAY', 'COLOR\_BayerBG2RGB', 'COLOR\_BayerBG2RGBA', 'COLOR\_BayerBG2RGB\_EA', 'COLOR\_BayerBG2RGB\_VNG', 'COLOR\_BayerBGGR2BGR', 'COLOR\_BayerBGGR2BGRA', 'COLOR\_BayerBGGR2BGR\_EA', 'COLOR\_BayerBGGR2BGR\_VNG', 'COLOR\_BayerBGGR2GRAY', 'COLOR\_BayerBGGR2RGB', 'COLOR\_BayerBGGR2RGBA', 'COLOR\_BayerBGGR2RGB\_EA', 'COLOR\_BayerBGGR2RGB\_VNG', 'COLOR\_BayerGB2BGR', 'COLOR\_BayerGB2BGRA', 'COLOR\_BayerGB2BGR\_EA', 'COLOR\_BayerGB2BGR\_VNG', 'COLOR\_BayerGB2GRAY', 'COLOR\_BayerGB2RGB', 'COLOR\_BayerGB2RGBA', 'COLOR\_BayerGB2RGB\_EA', 'COLOR\_BayerGB2RGB\_VNG', 'COLOR\_BayerGBRG2BGR', 'COLOR\_BayerGBRG2BGRA', 'COLOR\_BayerGBRG2BGR\_EA', 'COLOR\_BayerGBRG2BGR\_VNG', 'COLOR\_BayerGBRG2GRAY', 'COLOR\_BayerGBRG2RGB', 'COLOR\_BayerGBRG2RGBA', 'COLOR\_BayerGBRG2RGB\_EA', 'COLOR\_BayerGBRG2RGB\_VNG', 'COLOR\_BayerGR2BGR', 'COLOR\_BayerGR2BGRA', 'COLOR\_BayerGR2BGR\_EA', 'COLOR\_BayerGR2BGR\_VNG', 'COLOR\_BayerGR2GRAY', 'COLOR\_BayerGR2RGB', 'COLOR\_BayerGR2RGBA', 'COLOR\_BayerGR2RGB\_EA', 'COLOR\_BayerGR2RGB\_VNG', 'COLOR\_BayerGRBG2BGR', 'COLOR\_BayerGRBG2BGRA', 'COLOR\_BayerGRBG2BGR\_EA', 'COLOR\_BayerGRBG2BGR\_VNG', 'COLOR\_BayerGRBG2GRAY', 'COLOR\_BayerGRBG2RGB', 'COLOR\_BayerGRBG2RGBA', 'COLOR\_BayerGRBG2RGB\_EA', 'COLOR\_BayerGRBG2RGB\_VNG', 'COLOR\_BayerRG2BGR', 'COLOR\_BayerRG2BGRA', 'COLOR\_BayerRG2BGR\_EA', 'COLOR\_BayerRG2BGR\_VNG', 'COLOR\_BayerRG2GRAY', 'COLOR\_BayerRG2RGB', 'COLOR\_BayerRG2RGBA', 'COLOR\_BayerRG2RGB\_EA', 'COLOR\_BayerRG2RGB\_VNG', 'COLOR\_BayerRGGB2BGR', 'COLOR\_BayerRGGB2BGRA', 'COLOR\_BayerRGGB2BGR\_EA', 'COLOR\_BayerRGGB2BGR\_VNG', 'COLOR\_BayerRGGB2GRAY', 'COLOR\_BayerRGGB2RGB', 'COLOR\_BayerRGGB2RGBA', 'COLOR\_BayerRGGB2RGB\_EA', 'COLOR\_BayerRGGB2RGB\_VNG', 'COLOR\_COLORCVT\_MAX', 'COLOR\_GRAY2BGR', 'COLOR\_GRAY2BGR555', 'COLOR\_GRAY2BGR565', 'COLOR\_GRAY2BGRA', 'COLOR\_GRAY2RGB', 'COLOR\_GRAY2RGBA', 'COLOR\_HLS2BGR', 'COLOR\_HLS2BGR\_FULL', 'COLOR\_HLS2RGB', 'COLOR\_HLS2RGB\_FULL', 'COLOR\_HSV2BGR', 'COLOR\_HSV2BGR\_FULL', 'COLOR\_HSV2RGB', 'COLOR\_HSV2RGB\_FULL', 'COLOR\_LAB2BGR', 'COLOR\_LAB2LBGR', 'COLOR\_LAB2LRGB', 'COLOR\_LAB2RGB', 'COLOR\_LBGR2LAB', 'COLOR\_LBGR2LUV', 'COLOR\_LBGR2Lab', 'COLOR\_LBGR2Luv', 'COLOR\_LRGB2LAB', 'COLOR\_LRGB2LUV', 'COLOR\_LRGB2Lab', 'COLOR\_LRGB2Luv', 'COLOR\_LUV2BGR', 'COLOR\_LUV2LBGR', 'COLOR\_LUV2LRGB', 'COLOR\_LUV2RGB', 'COLOR\_Lab2BGR', 'COLOR\_Lab2LBGR', 'COLOR\_Lab2LRGB', 'COLOR\_Lab2RGB', 'COLOR\_Luv2BGR', 'COLOR\_Luv2LBGR', 'COLOR\_Luv2LRGB', 'COLOR\_Luv2RGB', 'COLOR\_M\_RGBA2RGBA', 'COLOR\_RGB2BGR', 'COLOR\_RGB2BGR555', 'COLOR\_RGB2BGR565', 'COLOR\_RGB2BGRA', 'COLOR\_RGB2GRAY', 'COLOR\_RGB2HLS', 'COLOR\_RGB2HLS\_FULL', 'COLOR\_RGB2HSV', 'COLOR\_RGB2HSV\_FULL', 'COLOR\_RGB2LAB', 'COLOR\_RGB2LUV', 'COLOR\_RGB2Lab', 'COLOR\_RGB2Luv', 'COLOR\_RGB2RGBA', 'COLOR\_RGB2XYZ', 'COLOR\_RGB2YCR\_CB', 'COLOR\_RGB2YCrCb', 'COLOR\_RGB2YUV', 'COLOR\_RGB2YUV\_I420', 'COLOR\_RGB2YUV\_IYUV', 'COLOR\_RGB2YUV\_YV12', 'COLOR\_RGBA2BGR', 'COLOR\_RGBA2BGR555', 'COLOR\_RGBA2BGR565', 'COLOR\_RGBA2BGRA', 'COLOR\_RGBA2GRAY', 'COLOR\_RGBA2M\_RGBA', 'COLOR\_RGBA2RGB', 'COLOR\_RGBA2YUV\_I420', 'COLOR\_RGBA2YUV\_IYUV', 'COLOR\_RGBA2YUV\_YV12', 'COLOR\_RGBA2mRGBA', 'COLOR\_XYZ2BGR', 'COLOR\_XYZ2RGB', 'COLOR\_YCR\_CB2BGR', 'COLOR\_YCR\_CB2RGB', 'COLOR\_YCrCb2BGR', 'COLOR\_YCrCb2RGB', 'COLOR\_YUV2BGR', 'COLOR\_YUV2BGRA\_I420', 'COLOR\_YUV2BGRA\_IYUV', 'COLOR\_YUV2BGRA\_NV12', 'COLOR\_YUV2BGRA\_NV21', 'COLOR\_YUV2BGRA\_UYNV', 'COLOR\_YUV2BGRA\_UYVY', 'COLOR\_YUV2BGRA\_Y422', 'COLOR\_YUV2BGRA\_YUNV', 'COLOR\_YUV2BGRA\_YUY2', 'COLOR\_YUV2BGRA\_YUYV', 'COLOR\_YUV2BGRA\_YV12', 'COLOR\_YUV2BGRA\_YVYU', 'COLOR\_YUV2BGR\_I420', 'COLOR\_YUV2BGR\_IYUV', 'COLOR\_YUV2BGR\_NV12', 'COLOR\_YUV2BGR\_NV21', 'COLOR\_YUV2BGR\_UYNV', 'COLOR\_YUV2BGR\_UYVY', 'COLOR\_YUV2BGR\_Y422', 'COLOR\_YUV2BGR\_YUNV', 'COLOR\_YUV2BGR\_YUY2', 'COLOR\_YUV2BGR\_YUYV', 'COLOR\_YUV2BGR\_YV12', 'COLOR\_YUV2BGR\_YVYU', 'COLOR\_YUV2GRAY\_420', 'COLOR\_YUV2GRAY\_I420', 'COLOR\_YUV2GRAY\_IYUV', 'COLOR\_YUV2GRAY\_NV12', 'COLOR\_YUV2GRAY\_NV21', 'COLOR\_YUV2GRAY\_UYNV', 'COLOR\_YUV2GRAY\_UYVY', 'COLOR\_YUV2GRAY\_Y422', 'COLOR\_YUV2GRAY\_YUNV', 'COLOR\_YUV2GRAY\_YUY2', 'COLOR\_YUV2GRAY\_YUYV', 'COLOR\_YUV2GRAY\_YV12', 'COLOR\_YUV2GRAY\_YVYU', 'COLOR\_YUV2RGB', 'COLOR\_YUV2RGBA\_I420', 'COLOR\_YUV2RGBA\_IYUV', 'COLOR\_YUV2RGBA\_NV12', 'COLOR\_YUV2RGBA\_NV21', 'COLOR\_YUV2RGBA\_UYNV', 'COLOR\_YUV2RGBA\_UYVY', 'COLOR\_YUV2RGBA\_Y422', 'COLOR\_YUV2RGBA\_YUNV', 'COLOR\_YUV2RGBA\_YUY2', 'COLOR\_YUV2RGBA\_YUYV', 'COLOR\_YUV2RGBA\_YV12', 'COLOR\_YUV2RGBA\_YVYU', 'COLOR\_YUV2RGB\_I420', 'COLOR\_YUV2RGB\_IYUV', 'COLOR\_YUV2RGB\_NV12', 'COLOR\_YUV2RGB\_NV21', 'COLOR\_YUV2RGB\_UYNV', 'COLOR\_YUV2RGB\_UYVY', 'COLOR\_YUV2RGB\_Y422', 'COLOR\_YUV2RGB\_YUNV', 'COLOR\_YUV2RGB\_YUY2', 'COLOR\_YUV2RGB\_YUYV', 'COLOR\_YUV2RGB\_YV12', 'COLOR\_YUV2RGB\_YVYU', 'COLOR\_YUV420P2BGR', 'COLOR\_YUV420P2BGRA', 'COLOR\_YUV420P2GRAY', 'COLOR\_YUV420P2RGB', 'COLOR\_YUV420P2RGBA', 'COLOR\_YUV420SP2BGR', 'COLOR\_YUV420SP2BGRA', 'COLOR\_YUV420SP2GRAY', 'COLOR\_YUV420SP2RGB', 'COLOR\_YUV420SP2RGBA', 'COLOR\_YUV420p2BGR', 'COLOR\_YUV420p2BGRA', 'COLOR\_YUV420p2GRAY', 'COLOR\_YUV420p2RGB', 'COLOR\_YUV420p2RGBA', 'COLOR\_YUV420sp2BGR', 'COLOR\_YUV420sp2BGRA', 'COLOR\_YUV420sp2GRAY', 'COLOR\_YUV420sp2RGB', 'COLOR\_YUV420sp2RGBA', 'COLOR\_mRGBA2RGBA']

Seguimiento de objetos Ahora que sabemos cómo convertir una imagen BGR a HSV, podemos usar esto para extraer un objeto coloreado. En HSV, es más fácil representar un color que en el espacio de color BGR. En nuestra aplicación, intentaremos extraer un objeto de color azul. Así que aquí está el método:

Toma cada fotograma del vídeo. Convertir del espacio de color BGR al espacio de color HSV Umbramos la imagen HSV para una gama de color azul. Ahora extrae el objeto azul solo, podemos hacer lo que queramos con esa imagen. A continuación se muestra el código que se comenta detalladamente:

import cv2 as cv  
import numpy as np  
cap = cv.VideoCapture(0)  
while(1):  
 # Take each frame  
 \_, frame = cap.read()  
 # Convert BGR to HSV  
 hsv = cv.cvtColor(frame, cv.COLOR\_BGR2HSV)  
 # define range of blue color in HSV  
 lower\_blue = np.array([110,50,50])  
 upper\_blue = np.array([130,255,255])  
 # Threshold the HSV image to get only blue colors  
 mask = cv.inRange(hsv, lower\_blue, upper\_blue)  
 # Bitwise-AND mask and original image  
 res = cv.bitwise\_and(frame,frame, mask= mask)  
 cv.imshow('frame',frame)  
 cv.imshow('mask',mask)  
 cv.imshow('res',res)  
 k = cv.waitKey(5) & 0xFF  
 if k == 27:  
 break  
cv.destroyAllWindows()

[ WARN:0@30.644] global cap\_v4l.cpp:982 open VIDEOIO(V4L2:/dev/video0): can't open camera by index  
[ERROR:0@30.644] global obsensor\_uvc\_stream\_channel.cpp:156 getStreamChannelGroup Camera index out of range

---------------------------------------------------------------------------  
error Traceback (most recent call last)  
Cell In[4], line 8  
 6 \_, frame = cap.read()  
 7 # Convert BGR to HSV  
----> 8 hsv = cv.cvtColor(frame, cv.COLOR\_BGR2HSV)  
 9 # define range of blue color in HSV  
 10 lower\_blue = np.array([110,50,50])  
  
error: OpenCV(4.8.1) /io/opencv/modules/imgproc/src/color.cpp:182: error: (-215:Assertion failed) !\_src.empty() in function 'cvtColor'

¿Cómo encontrar valores de HSV para rastrear? Esta es una pregunta común que se encuentra en stackoverflow.com . Es muy simple y puedes usar la misma función, cv.cvtColor() . En lugar de pasar una imagen, simplemente pasa los valores de BGR que desea. Por ejemplo, para encontrar el valor HSV de Verde, pruebe los siguientes comandos en una terminal Python:

Ejercicio: Intente encontrar una manera de extraer más de un objeto coloreado, por ejemplo, extraer objetos rojos, azules y verdes simultáneamente.

import cv2  
import numpy as np  
# Para un video  
#ap = cv2.VideoCapture('nombre\_del\_video.mp4')  
  
# Para una imagen  
img = cv2.imread('resources/apple.jpg')  
lower\_red = np.array([0, 100, 100])  
upper\_red = np.array([10, 255, 255])  
  
lower\_green = np.array([40, 40, 40])  
upper\_green = np.array([80, 255, 255])  
  
lower\_blue = np.array([100, 100, 100])  
upper\_blue = np.array([140, 255, 255])  
hsv = cv2.cvtColor(img, cv2.COLOR\_BGR2HSV)  
  
mask\_red = cv2.inRange(hsv, lower\_red, upper\_red)  
mask\_green = cv2.inRange(hsv, lower\_green, upper\_green)  
mask\_blue = cv2.inRange(hsv, lower\_blue, upper\_blue)  
  
result\_red = cv2.bitwise\_and(img, img, mask=mask\_red)  
result\_green = cv2.bitwise\_and(img, img, mask=mask\_green)  
result\_blue = cv2.bitwise\_and(img, img, mask=mask\_blue)  
  
cv2.imshow('Result Red', result\_red)  
cv2.imshow('Result Green', result\_green)  
cv2.imshow('Result Blue', result\_blue)  
cv2.waitKey(0)  
cv2.destroyAllWindows()

qt.qpa.plugin: Could not find the Qt platform plugin "wayland" in "/home/dread/miniconda3/envs/tf/lib/python3.9/site-packages/cv2/qt/plugins"