**เปลี่ยน Colorspaces**

เป้าหมาย

* ในบทแนะนำนี้คุณจะได้เรียนรู้วิธีแปลงรูปภาพจากพื้นที่สีหนึ่งไปยังพื้นที่อื่นเช่น BGR \ leftrightarrowGray, BGR \ leftrightarrowHSV ฯลฯ
* นอกจากนั้นเราจะสร้างแอปพลิเคชันซึ่งจะแยกแยะวัตถุสีออกในวิดีโอ
* คุณจะได้เรียนรู้ฟังก์ชันต่อไปนี้: **cv2.cvtColor ()** , **cv2.inRange ()**ฯลฯ

เปลี่ยนพื้นที่สี

มีวิธีการแปลงสีพื้นที่มากกว่า 150 แบบที่มีอยู่ใน OpenCV แต่เราจะมองเข้าไปในสองส่วนเท่านั้นที่ใช้กันอย่างแพร่หลายคือ BGR \ leftrightarrowGrey และ BGR \ leftrightarrowHSV

สำหรับการแปลงสีเราใช้ฟังก์ชัน cv2.cvtColor (input\_image, flag)โดยที่flagกำหนดประเภทของ Conversion

สำหรับ BGR \ลูกศรขวาแปลงสีเทาเราจะใช้ flag cv2.COLOR\_BGR2GRAY ในทำนองเดียวกันสำหรับ BGR \ลูกศรขวาHSV เราจะใช้ flag cv2.COLOR\_BGR2HSV หากต้องการรับแฟล็กอื่นให้เรียกใช้คำสั่งต่อไปนี้ในเทอร์มินัล Python ของคุณ:

**>>> import** **cv2**

**>>>** flags = [i **for** i **in** dir(cv2) **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\_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\_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\_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\_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\_BayerGB2BGR', 'COLOR\_BayerGB2BGRA', 'COLOR\_BayerGB2BGR\_EA', 'COLOR\_BayerGB2BGR\_VNG', 'COLOR\_BayerGB2GRAY', 'COLOR\_BayerGB2RGB', 'COLOR\_BayerGB2RGBA', 'COLOR\_BayerGB2RGB\_EA', 'COLOR\_BayerGB2RGB\_VNG', 'COLOR\_BayerGR2BGR', 'COLOR\_BayerGR2BGRA', 'COLOR\_BayerGR2BGR\_EA', 'COLOR\_BayerGR2BGR\_VNG', 'COLOR\_BayerGR2GRAY', 'COLOR\_BayerGR2RGB', 'COLOR\_BayerGR2RGBA', 'COLOR\_BayerGR2RGB\_EA', 'COLOR\_BayerGR2RGB\_VNG', 'COLOR\_BayerRG2BGR', 'COLOR\_BayerRG2BGRA', 'COLOR\_BayerRG2BGR\_EA', 'COLOR\_BayerRG2BGR\_VNG', 'COLOR\_BayerRG2GRAY', 'COLOR\_BayerRG2RGB', 'COLOR\_BayerRG2RGBA', 'COLOR\_BayerRG2RGB\_EA', 'COLOR\_BayerRG2RGB\_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']**

**บันทึก :** สำหรับช่วง HSV ช่วงเฉดสีคือ Hue [0,179], ช่วงความอิ่มตัวคือ Saturation [0,255] และช่วงค่าอยู่ที่ Value [0,255] โปรแกรมต่างๆใช้เครื่องชั่งที่แตกต่างกัน ดังนั้นถ้าคุณกำลังเปรียบเทียบค่า OpenCV กับค่าเหล่านี้คุณจะต้องปรับค่าช่วงเหล่านี้ให้เป็นปกติ

การติดตามวัตถุ

ตอนนี้เรารู้วิธีการแปลงภาพ BGR เป็น HSV เราสามารถใช้เพื่อดึงวัตถุสีออกได้ ใน HSV จะแสดงสีได้ง่ายกว่าพื้นที่สี RGB ในโปรแกรมของเราเราจะพยายามดึงวัตถุสีฟ้าออก ดังนั้นนี่คือวิธีการ:

* ถ่ายวิดีโอแต่ละเฟรม
* แปลงจาก BGR เป็น HSV color-space
* เราตั้งเกณฑ์ภาพ HSV สำหรับช่วงของสีฟ้า
* ตอนนี้ดึงวัตถุสีฟ้าเพียงอย่างเดียวเราสามารถทำสิ่งที่อยู่ในภาพที่เราต้องการ

ด้านล่างนี้เป็นรหัสที่แสดงความคิดเห็นในรายละเอียด:

**import** **cv2**

**import** **numpy** **as** **np**

cap = cv2.VideoCapture(0)

**while**(1):

*# Take each frame*

\_, frame = cap.read()

*# Convert BGR to HSV*

hsv = cv2.cvtColor(frame, cv2.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 = cv2.inRange(hsv, lower\_green, upper\_green)

*# Bitwise-AND mask and original image*

res = cv2.bitwise\_and(frame,frame, mask= mask)

cv2.imshow('frame',frame)

cv2.imshow('mask',mask)

cv2.imshow('res',res)

k = cv2.waitKey(5) & 0xFF

**if** k == 27:

**break**

cv2.destroyAllWindows()



**บันทึก :** นี่เป็นวิธีที่ง่ายที่สุดในการติดตามวัตถุ เมื่อคุณเรียนรู้การทำงานของรูปทรงแล้วคุณสามารถทำสิ่งต่างๆมากมายได้เช่นหา centroid ของวัตถุนี้และใช้เพื่อติดตามวัตถุวาดแผนภาพโดยการขยับมือของคุณต่อหน้ากล้องและสารพัด

วิธีการหาค่า HSV เพื่อติดตาม

นี่เป็นคำถามทั่วไปที่พบใน stackoverflow.com มันเป็นเรื่องง่ายมากและคุณสามารถใช้ฟังก์ชั่นเดียวกัน*cv2.cvtColor*() แทนที่จะส่งผ่านภาพคุณเพียงแค่ส่งค่า BGR ที่คุณต้องการ ตัวอย่างเช่นในการหาค่า HSV ของ Green ลองทำตามคำสั่งใน terminal ของ Python:

**>>>** green = np.uint8([[[0,255,0 ]]])

**>>>** hsv\_green = cv2.cvtColor(green,cv2.COLOR\_BGR2HSV)

**>>> print** hsv\_green

[[[ 60 255 255]]]

*ตอนนี้คุณใช้ [H-10, 100,100] และ [H +10, 255, 255] เป็นขีดล่างและส่วนบนตามลำดับ นอกเหนือจากวิธีนี้คุณสามารถใช้เครื่องมือแก้ไขรูปภาพเช่น GIMP หรือตัวแปลงแบบออนไลน์เพื่อค้นหาค่าเหล่านี้ได้ แต่อย่าลืมปรับช่วง HSV*