Practica 6
Github

https://github.com/Leogaltre/Practica-6-Segmentaci-n-de-Colores-en-Video.git

Tipo de letra de escritura en Word para mostrar la programación -> 3ds tamaño 11

Tipo de Letra de Python -> Courier New tamaño 12

Objetivo

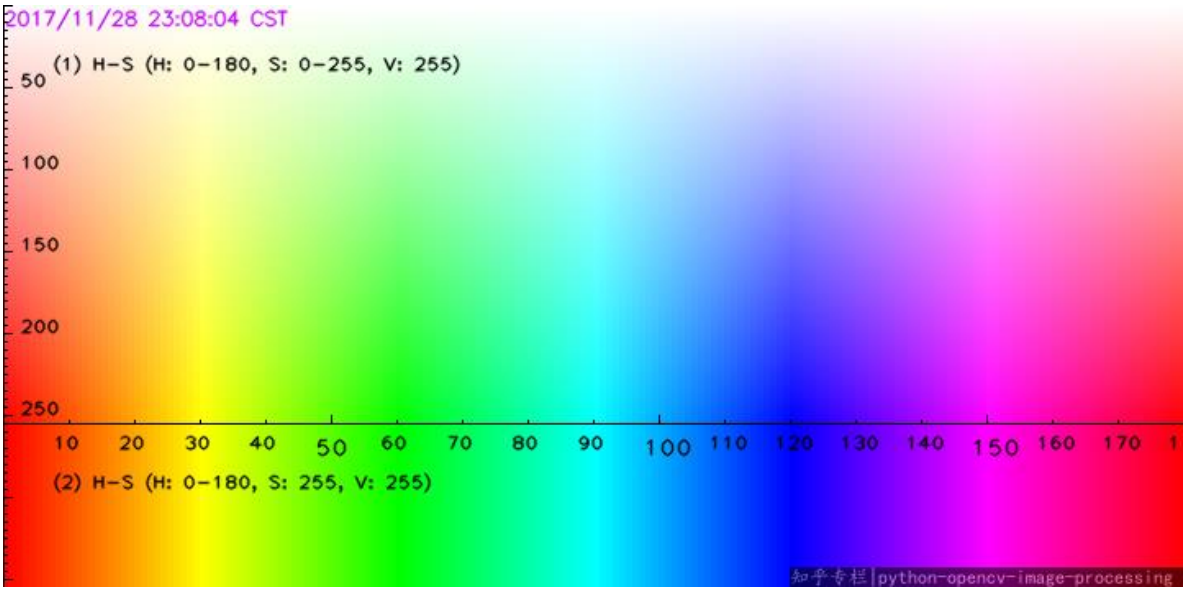
¿Qué es HSV?

Es un espacio cuyos componentes son: Matiz, Saturación y Brillo (Hue, Saturation y Value), Se usa este espacio porque es más sencillo identificar el color o los colores a detectar:

H -> 0 a 179

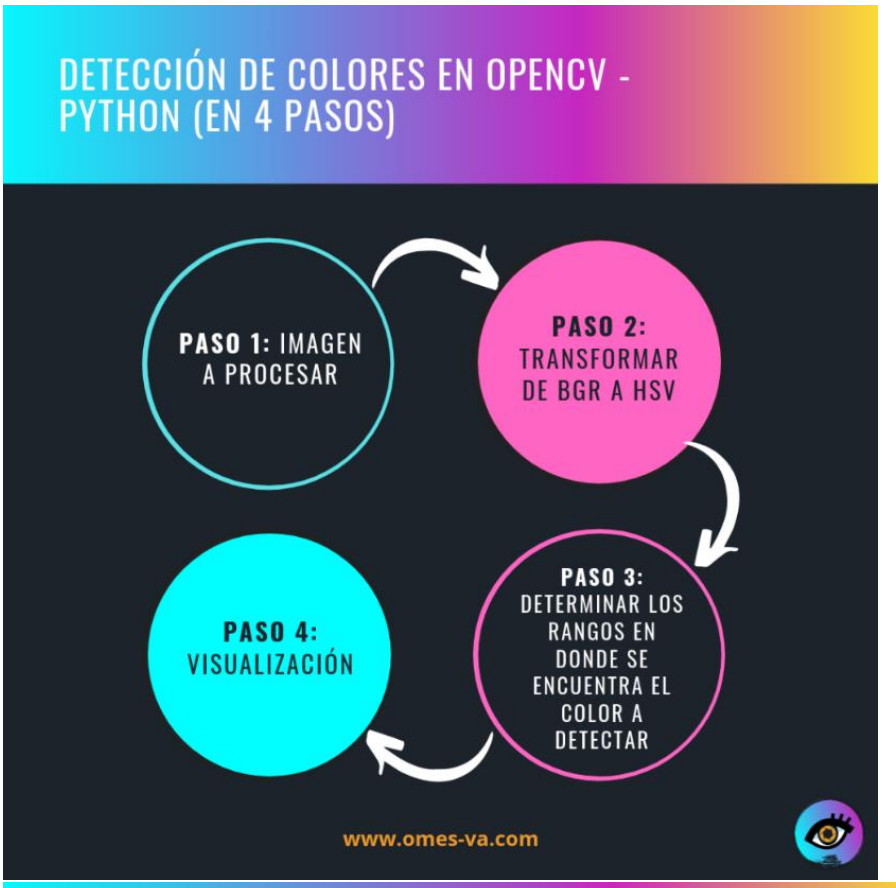
S -> 0 a 255

V -> 0 a 255



¿Qué es segmentación?

Aislamiento de rango de colores que se quiere extraer a partir de una imagen, mediante el uso del espacio de HSV


DETECCIÓN DE COLORES EN OPENCV - PYTHON (EN 4 PASOS)
PASO 1: IMAGEN A PROCESAR
PASO 2: TRANSFORMAR DE BGR A HSV
PASO 3: DETERMINAR LOS RANGOS EN DONDE SE ENCUENTRA EL COLOR A DETECTAR
PASO 4: VISUALIZACIÓN
www.omes-va.com


DETECCIÓN DE COLORES EN OPENCV - PYTHON (EN 4 PASOS)
PASO 1: IMAGEN A PROCESAR
Se necesita una imagen o fotogramas con los cuales trabajar, por ello este paso puede darse a través de:
Lectura de una imagen
Video
LEE VISUALIZA Y GUARDA IMÁGENES
OpenCV y Python
OpenCV
CAPTURA GUARDA Y LEE VIDEO
OpenCV y Python

DETECCIÓN DE COLORES EN OPENCV - PYTHON (EN 4 PASOS)

PASO 2:
TRANSFORMAR DE BGR A HSV

Cuando una imagen es leída en OpenCV, por defecto la lee en BGR, por ello en este paso necesita ser transformada al espacio de color HSV. Para ello se emplea la siguiente línea:

```
imagenHSV = cv2.cvtColor(imagen, cv2.COLOR_BGR2HSV)
```

- imagenHSV: Imagen resultante
- imagen: Imagen BGR
- cv2.COLOR_BGR2HSV: Transformación BGR a HSV

¿Por qué HSV?
HSV, es un espacio de color cuyos componentes son: Matiz, Saturación y Brillo (Hue, Saturation and Value). Se usa este espacio de color debido a que es más sencillo determinar el o los colores a detectar, para ello se emplea especial atención al componente H, que corresponde al matiz. En OpenCV sus componentes pueden tomar los siguientes valores:

- H: 0 a 179
- S: 0 a 255
- V: 0 a 255

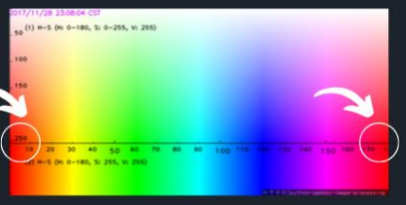
DETECCIÓN DE COLORES EN OPENCV - PYTHON (EN 4 PASOS)

Otra forma de encontrar los rangos del color que se desea detectar es, tomar una porción de la imagen de entrada en donde se encuentre dicho color. Luego podrás determinar los rangos.

PASO 3:
DETERMINAR LOS RANGOS EN DONDE SE ENCUENTRA EL COLOR A DETECTAR

Se necesita construir dos arrays en donde esté presente el límite inicial y final en HSV del color a detectar. Se busca en H donde podría estar ubicado dicho color, mientras que para S y V se pueden usar rangos de entre 100 a 255 y de 20 a 255 respectivamente.

Si se desease detectar el color rojo, en H se encuentra en dos lugares, al principio y al final. Entonces se debe establecer entre que valores se encuentra, por ejemplo de 0 a 8, y de 175 a 179.



Para determinar los rangos, en OpenCV se usa:

```
imagenBinaria = cv2.inRange(imagenHSV, limiteIni, limiteFin)
```

- imagenBinaria: Imagen resultante
- imagenHSV: Imagen HSV
- limiteIni: Límite inicial
- limiteFin: Límite final

Si se necesitara sumar de dos imágenes (como en el caso del color rojo en el cual se obtendrán 2 imágenes binarias):

```
imagenResult = cv2.add(imagen1, imagen2)
```

- imagenResult: Imagen resultante
- imagen1: Imagen 1
- imagen2: Imagen 2

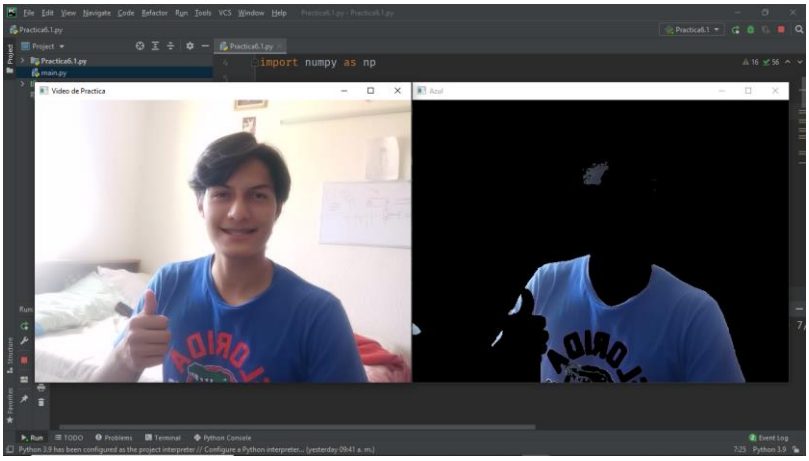
Código

```
# Programa completo Error
import cv2
import numpy as np

# cap = cv2.VideoCapture('Practica.mp4')
# cap = cv2.VideoCapture(1)
cap = cv2.VideoCapture(0)

# Convertir la imagen RGB a HSV, que es HSV?? ----------------------------
--
visualiza_un_HSV = np.uint8([[[0, 0, 255]]]) #[[[200, 0, 0]]]
azulbajo = np.array(cv2.cvtColor(visualiza_un_HSV, cv2.COLOR_BGR2HSV))
print(visualiza_un_HSV)
print(azulbajo)
# HSV -> [0 - 179, 0 - 255, 0 - 255]
# HSV -> [Hue, Saturation, Value]
# HSV -> [Matiz, Saturación, Valor]

# Definir intervalo del color para el Azul en HSV
azulbajo = np.array([100, 50, 50])  #[200,0,0] - 110
azulalto = np.array([130, 255, 255])  #[255,0,0] - 120

# Definir intervalo del color para el Verde en HSV
verdebajo = np.array([40, 50, 50]) #[0,175,0] - 50
```

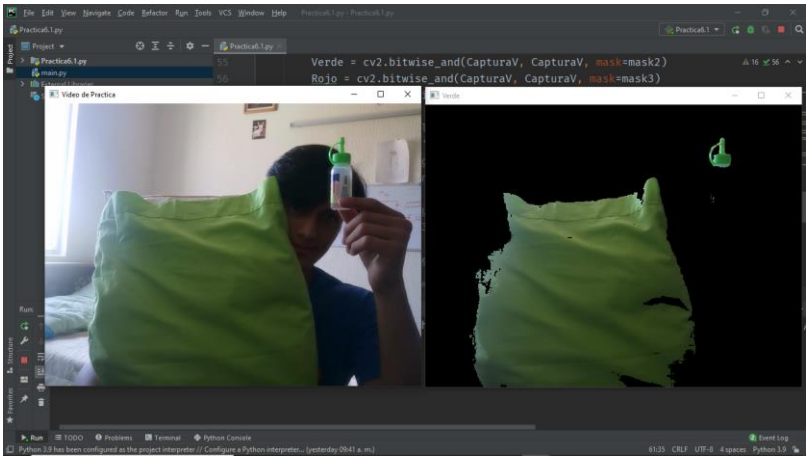
```
verdealto = np.array([80, 255, 255]) #[0,255,0] - 80

# Definir intervalo del color para el Rojo en HSV
rojobajo1 = np.array([0, 50, 50])    #[0,0,255] - 0
rojoalto1 = np.array([20, 255, 255])    #[0,0,150] - 10

rojobajo2 = np.array([165, 50, 50])    #[0,0,255] - 0
rojoalto2 = np.array([179, 255, 255])    #[0,0,150] - 10

rojobajo = cv2.add(rojobajo1, rojobajo2)
rojoalto = cv2.add(rojoalto1, rojoalto2)

# Definir intervalo del color para el Blanco en HSV
blancobajo = np.array([0, 0, 150])    #[200,200,200]
blancoalto = np.array([0, 0, 255])    #[255,255,255]

while True:
    ret, CapturaV = cap.read()
    if ret == True:
        # hsv = cv2.cvtColor(cap, cv2.COLOR_BGR2HSV)
        CapturaV_HSV = cv2.cvtColor(CapturaV, cv2.COLOR_BGR2HSV)

        # Umbralizar la imagen HSV para obtener solo los rango de colores
        mask1 = cv2.inRange(CapturaV_HSV, azulbajo, azulalto)
        mask2 = cv2.inRange(CapturaV_HSV, verdebajo, verdealto)
        mask3 = cv2.inRange(CapturaV_HSV, rojobajo, rojoalto)
        mask4 = cv2.inRange(CapturaV_HSV, blancobajo, blancoalto)

        # Bitwise (# Bit a bit) and mask and original image
        Azul = cv2.bitwise_and(CapturaV, CapturaV, mask=mask1)
        Verde = cv2.bitwise_and(CapturaV, CapturaV, mask=mask2)
        Rojo = cv2.bitwise_and(CapturaV, CapturaV, mask=mask3)
        Blanco = cv2.bitwise_and(CapturaV, CapturaV, mask=mask4)

        # cv2.imshow('Azul', Azul)
        # cv2.imshow('Verde', Verde)
        cv2.imshow('Rojo', Rojo)
        # cv2.imshow('Blanco', Blanco)
        cv2.imshow('Video de Practica', CapturaV)
        # Para video preguardado
        # if cv2.waitKey(350) & 0xFF == ord('n'):
        #     break
        # Para video Streeming
        if cv2.waitKey(1) & 0xFF == ord('n'):
            break

cap.release()
cv2.destroyAllWindows()
```

Colores Mostrados

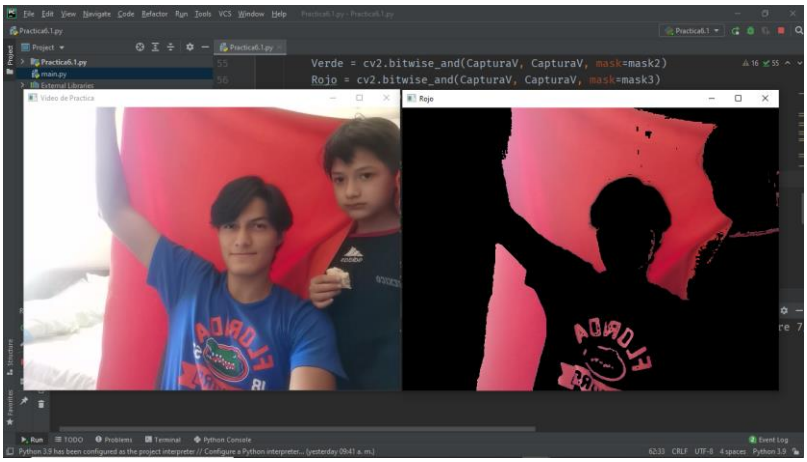
Azul



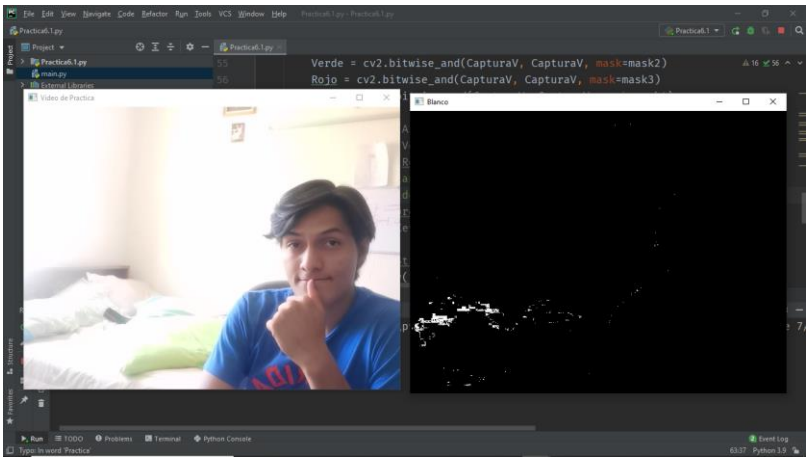
Verde



Rojo



Blanco



El color blanco es difícil de conseguir ya que, por la iluminación, puede tomar tonos desde azules, rojos, verdes y grises tenues que salen del rango establecido del color blanco

Todos estos colores dependen de la iluminación del ambiente ya que no es la misma en todos los lugares.

Referencias:

HSV y Segmentación

https://omes-va.com/deteccion-de-colores/

RGB a HSV (Fallo en H tiene un rango de 0 - 360 y no de 0 - 179)

https://www.peko-step.com/es/tool/hsvrgb.html

Ver después:

https://programmerclick.com/article/60941814035/