

Rodrigo Hernández Zavala

Fecha

Ing. Mecatrónica

22310215

Filtros de color

Visión Artificial

Este código realiza una **captura de video en tiempo real desde la cámara**, detecta los colores **rojo, verde y azul** en el espacio **HSV**, y aplica un efecto de **pantalla verde (green screen)**, además de mostrar el video en formato **YUV**. A continuación te explico paso a paso:

✓ ¿Qué hace este código?

1. Captura de video

```
python
CopiarEditar
cap = cv2.VideoCapture(0)
```

- Abre la cámara por defecto (0).
 - Verifica si la cámara está disponible (`cap.isOpened()`).
-

2. Bucle de procesamiento de video

```
python
CopiarEditar
while True:
    ret, frame = cap.read()
```

- Lee cada fotograma del video.
 - Redimensiona el fotograma a 640x480 para optimizar el procesamiento.
-

3. Conversión a HSV

```
python
CopiarEditar
hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
```

- Convierte el color de BGR (OpenCV por defecto) a **HSV**, que es más adecuado para segmentar colores.
-

4. Detección de color rojo

```
python
```

```
python
CopiarEditar
lower_red1 / upper_red1 # Rango bajo
lower_red2 / upper_red2 # Rango alto
```

- El rojo en HSV está en dos zonas: cerca de 0° y cerca de 180°, por eso se usan **dos rangos**.
 - Se crea una **máscara binaria** para cada rango con `cv2.inRange()`, y se combinan.
-

5. Detección de color verde y azul

```
python
CopiarEditar
mask_green = cv2.inRange(hsv, lower_green, upper_green)
mask_blue = cv2.inRange(hsv, lower_blue, upper_blue)
```

- Se definen rangos HSV para verde y azul y se generan sus máscaras.
-

6. Aplicación de las máscaras

```
python
CopiarEditar
res_red = cv2.bitwise_and(frame, frame, mask=mask_red)
```

- Usa `bitwise_and` para **mostrar solo los píxeles** donde la máscara es blanca (color detectado).
 - Se hace esto para rojo, verde y azul.
-

7. Efecto de Green Screen

```
python
CopiarEditar
mask_inv_green = cv2.bitwise_not(mask_green)
green_removed = cv2.bitwise_and(frame, frame, mask=mask_inv_green)
```

- Invierte la máscara del verde (zonas **no verdes**).
 - Aplica esa máscara para **eliminar el color verde del fondo**, simulando un "green screen".
-

8. Conversión a YUV

```
python
```

```
CopiarEditar
yuv = cv2.cvtColor(frame, cv2.COLOR_BGR2YUV)
```

- Convierte el video a **espacio de color YUV**, usado en compresión de video y TV.
-

9. Visualización de resultados

```
python
CopiarEditar
cv2.imshow(...)
```

Se muestran varias ventanas:

- Video Original
 - Filtro Rojo, Filtro Verde, Filtro Azul
 - Green Screen (Verde Eliminado)
 - Video en YUV
-

10. Salir con la tecla 'q'

```
python
CopiarEditar
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
```

- Termina el bucle cuando se presiona la tecla 'q'.
-

¿Aplicaciones prácticas?

- **Visión artificial y robótica** para detectar colores específicos.
- **Green screen** en efectos de video.
- **Seguimiento de objetos** por color.
- Preprocesamiento para segmentación y reconocimiento.