Informe de práctica 2

Para el desarrollo de la práctica se utilizaron las librerías OpenCV (cv2), NumPy y datetime. OpenCV se empleó para la captura, procesamiento y visualización de los fotogramas del video; NumPy permitió definir y manipular los rangos de color como arreglos numéricos; mientras que datetime se utilizó para generar nombres únicos de archivo basados en la hora del sistema al momento de guardar las imágenes resultantes.

Se definieron dos rangos de color en el espacio **HSV (Hue, Saturation, Value)**. Estos rangos fueron establecidos para detectar diferentes tonalidades de piel, considerando que la iluminación y las variaciones cromáticas pueden afectar la uniformidad de un solo rango.

El uso de dos rangos complementarios permite ampliar la cobertura de detección, mejorando la precisión en la segmentación de las áreas correspondientes al tono de piel.

El espacio de color **HSV** se seleccionó en lugar de **BGR**, ya que este último representa directamente los canales de color azul, verde y rojo, lo cual no facilita la separación entre el color y la intensidad luminosa. En cambio, el modelo HSV permite distinguir el componente de **tono (Hue)** de la **intensidad (Value)**, siendo más robusto ante cambios de iluminación. Esto facilita la detección de regiones con un color específico, como la piel, sin que se vean afectadas por la cantidad de luz presente en la escena.



Figura 1: Ejemplo de detección de piel

Durante la ejecución, cada fotograma se convierte de **BGR** a **HSV** mediante **cv2.cvtColor()**. Luego, se aplican los dos rangos de detección utilizando **cv2.inRange()**, generando máscaras binarias donde los píxeles dentro del rango son blancos (valor 255) y los demás son negros (valor 0). Las dos máscaras se combinan con **cv2.add()** para obtener una máscara final que agrupa todas las áreas detectadas.

Finalmente, se aplica la máscara combinada al fotograma original mediante la operación **cv2.bitwise\_and()**. Este proceso conserva únicamente los píxeles correspondientes al color detectado y coloca en negro el resto de la imagen. El resultado se almacena al frame al presionar la tecla ‘S’, generando un archivo de imagen que contiene únicamente las regiones segmentadas correspondientes a la piel.

Repositorio: <https://github.com/militem/computervision>

**Conclusiones**

1. El uso del espacio de color HSV permitió una segmentación más precisa y estable en comparación con el modelo BGR, ya que separa la información de color de la iluminación, reduciendo la sensibilidad del sistema ante cambios de brillo o sombras en los fotogramas del video.
2. La aplicación de dos rangos de detección complementarios mejoró la cobertura en la identificación de tonos de piel, permitiendo obtener una máscara más completa y robusta frente a variaciones cromáticas naturales o condiciones de iluminación no uniformes.

**Punto extra**

Para desarrollar el desenfoque en una cámara hemos usado la librería YOLOv8 de ultralytics, quien se encarga en realizar el efecto de desenfoque del fondo en tiempo real, usando máscaras para detectar a las personas y filtros como el Gaussiano para el desenfoque para luego combina las máscaras para tener el efecto deseado.