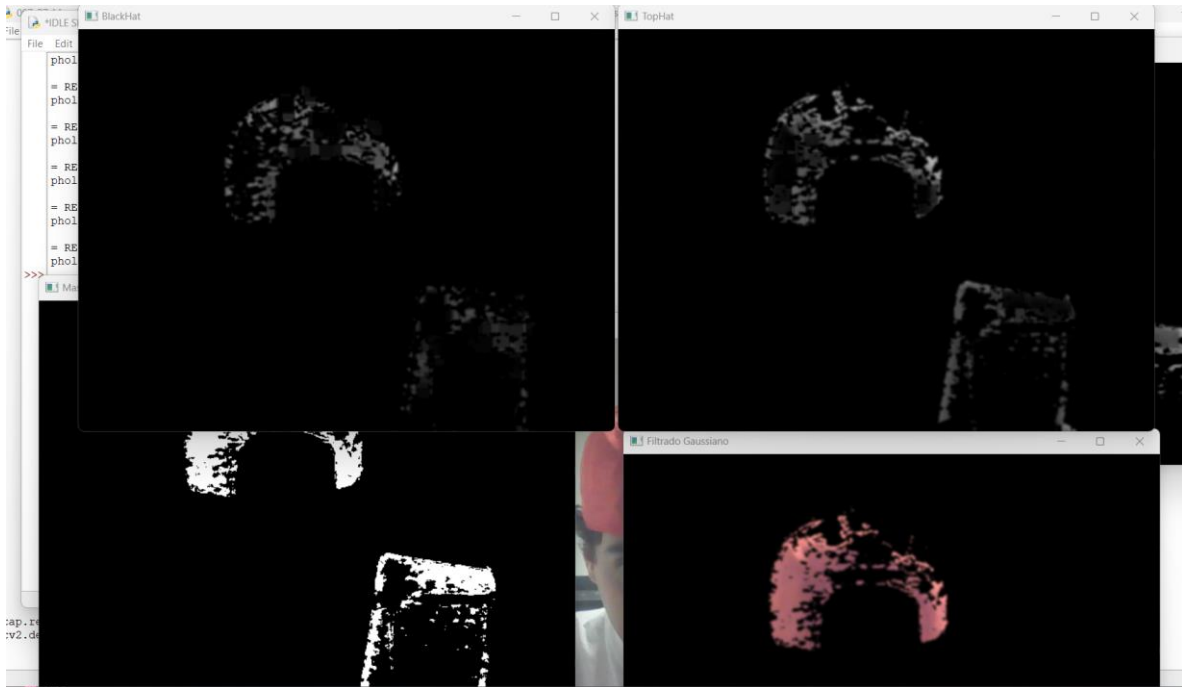


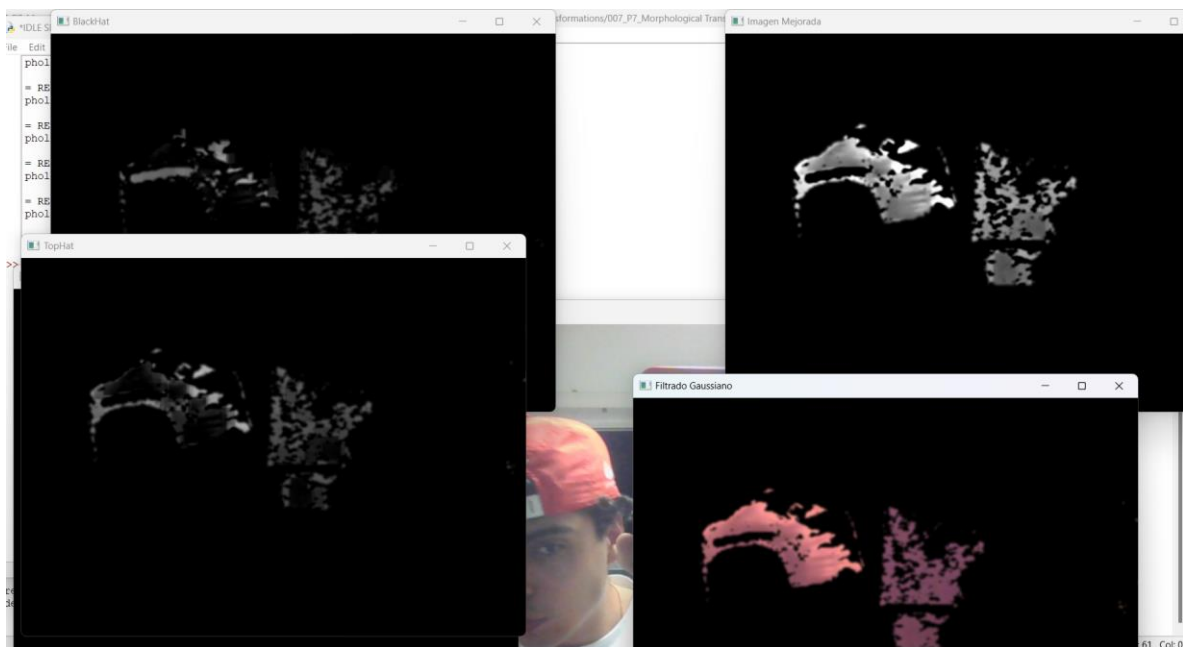
007_P7_Morphological Transformations

- Inicie sesión en mi Github
- Creé un nuevo archivo para la práctica 7 en Python
- Vi el tutorial del video de la práctica 7 de Python “Morphological Transformations Python Tutorial”
- Puse el código que viene de ejemplo, y vi lo que tenía que hacer con lo que nos puso el profesor que es Remover Ruido – Lineal y morfológicamente y cómo principal objetivo es: Remover ruido de la detección F+ y F- hacer las funciones TOPHAT y BLACKHAT.
- Lo ejecuté y le pedí ayuda a chat que es parecido a lo de la práctica anterior y ya tomé la captura del código y su respectiva evidencia.
- Después hice lo que pedía en la práctica 7 ya lo antes mencionado y me salió un poco raro porque no mejoraba la resolución de mi cámara y tuve que agregar un filtro gaussiano el cual sirve para suavizar la imagen aplicando una media ponderada donde los píxeles más cercanos al centro tienen más peso.
- Lo compilé y me salió las ventanas con las imágenes con sus respectivos ruidos de la detección F+ y F- hacer las funciones TOPHAT y BLACKHAT.
- Tome la segunda prueba de la evidencia y tomé nota de mis errores.
- Creé el reporte .
- Subí todos los archivos a mi GithubDesktop sobre la práctica 7 “Morphological Transformations”

PRUEBA 1:



PRUEBA 2:



Mi código Práctica 7:

```
import cv2

import numpy as np

cap = cv2.VideoCapture(0)

# Kernel para operaciones morfológicas
kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (15, 15))

while True:
    ret, frame = cap.read()
    if not ret:
        break

    # Convertir a HSV
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    # ----- RANGO COMPLETO PARA COLOR ROJO -----
    lower_red1 = np.array([0, 100, 100])
    upper_red1 = np.array([10, 255, 255])
    lower_red2 = np.array([160, 100, 100])
    upper_red2 = np.array([179, 255, 255])

    # Crear las dos máscaras y unir las
    mask1 = cv2.inRange(hsv, lower_red1, upper_red1)
    mask2 = cv2.inRange(hsv, lower_red2, upper_red2)
```

```
mask = cv2.bitwise_or(mask1, mask2)

# Aplicar máscara a la imagen original
res = cv2.bitwise_and(frame, frame, mask=mask)

# ----- FILTRADO PARA REMOVER RUIDO LINEAL -----
blurred = cv2.GaussianBlur(res, (7, 7), 0)

# Convertir a escala de grises para morfología
gray = cv2.cvtColor(blurred, cv2.COLOR_BGR2GRAY)

# ----- OPERACIONES MORFOLÓGICAS -----
tophat = cv2.morphologyEx(gray, cv2.MORPH_TOPHAT, kernel)
blackhat = cv2.morphologyEx(gray, cv2.MORPH_BLACKHAT, kernel)

# Mejorar imagen combinando resultados
enhanced = cv2.add(gray, tophat)
enhanced = cv2.subtract(enhanced, blackhat)

# Mostrar resultados
cv2.imshow('Original', frame)
cv2.imshow('Mascara Roja', mask)
cv2.imshow('Filtrado Gaussiano', blurred)
cv2.imshow('TopHat', tophat)
cv2.imshow('BlackHat', blackhat)
cv2.imshow('Imagen Mejorada', enhanced)
```

```
# Salir con ESC
```

```
k = cv2.waitKey(5) & 0xFF
```

```
if k == 27:
```

```
    break
```

```
cap.release()
```

```
cv2.destroyAllWindows()
```

ENLACE DE MI REPOSITORIO DE GITHUB:

<https://github.com/Ing-OscarValencia/VisionArtificialOGV-2025.git>