



# **ceti**

**CENTRO DE ENSEÑANZA  
TÉCNICA INDUSTRIAL**

Maestro: Mauricio Alejandro Cabrera Arellano

Alumno: Alejandro Retana Rubio 22110315

Materia: Visión Artificial

Practica 7

Fecha: 09-06-2025

## Práctica: Remoción de Ruido y Realce de Detalles en Imágenes

En esta práctica se aplicaron técnicas de procesamiento digital de imágenes para **reducir el ruido y resaltar detalles** en una imagen en escala de grises. Primero, se utilizó un **filtro Gaussiano** para suavizar la imagen y eliminar pequeñas imperfecciones. Luego, se aplicaron **operaciones morfológicas**:

- **TopHat**, que destaca detalles claros sobre un fondo oscuro,
- **BlackHat**, que resalta detalles oscuros sobre un fondo claro.

Estas operaciones son útiles en tareas como detección de bordes, segmentación y preprocesamiento para visión por computadora. Finalmente, los resultados se visualizaron usando gráficos para comparar visualmente el efecto de cada paso.

Codigo

```

        # Importar las librerías necesarias
        import cv2 # OpenCV para procesamiento de imágenes
        import numpy as np # Operaciones numéricas eficientes
        import matplotlib.pyplot as plt # Visualización de imágenes

        # Cargar la imagen en escala de grises
        img = cv2.imread('luffy5.png', cv2.IMREAD_GRAYSCALE)
        # Leer imagen como escala de grises para facilitar el procesamiento

        # ----- Aplicar filtro Gaussiano para suavizar -----
        img_suavizada = cv2.GaussianBlur(img, (5, 5), 0)
        # Suaviza la imagen y reduce el ruido utilizando un kernel 5x5

        # ----- Definir kernel estructurante -----
        kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (9, 9))
        # Kernel morfológico rectangular de 9x9 píxeles

        # ----- Operaciones morfológicas -----
        tophat = cv2.morphologyEx(img_suavizada, cv2.MORPH_TOPHAT, kernel)
        # TopHat: resalta regiones claras más pequeñas que el entorno

        blackhat = cv2.morphologyEx(img_suavizada, cv2.MORPH_BLACKHAT, kernel)
        # BlackHat: resalta regiones oscuras más pequeñas que el entorno

        # ----- Visualización de resultados -----
        plt.figure(figsize=(10, 6)) # Configurar tamaño de la figura

        # Imagen original
        plt.subplot(2, 2, 1)
        plt.imshow(img, cmap='gray')
        plt.title('Imagen Original')
        plt.axis('off')

        # Imagen suavizada
        plt.subplot(2, 2, 2)
        plt.imshow(img_suavizada, cmap='gray')
        plt.title('Filtro Gaussiano')
        plt.axis('off')

        # Resultado TopHat
        plt.subplot(2, 2, 3)
        plt.imshow(tophat, cmap='gray')

```

```
plt.title('TopHat')
plt.axis('off')

# Resultado BlackHat
plt.subplot(2, 2, 4)
plt.imshow(blackhat, cmap='gray')
plt.title('BlackHat')
plt.axis('off')

# Ajustar espacios entre subgráficas
plt.tight_layout()
plt.show()
```

Figure 1

