

**Centro de Enseñanza Técnica Industrial**

**Visión Artificial.**

**Ingeniería Mecatrónica.**



**Practica# 5: Recuperar información y mostrarla con los diferentes umbrales en una misma imagen.**

**Platel: CETI Colomos.**

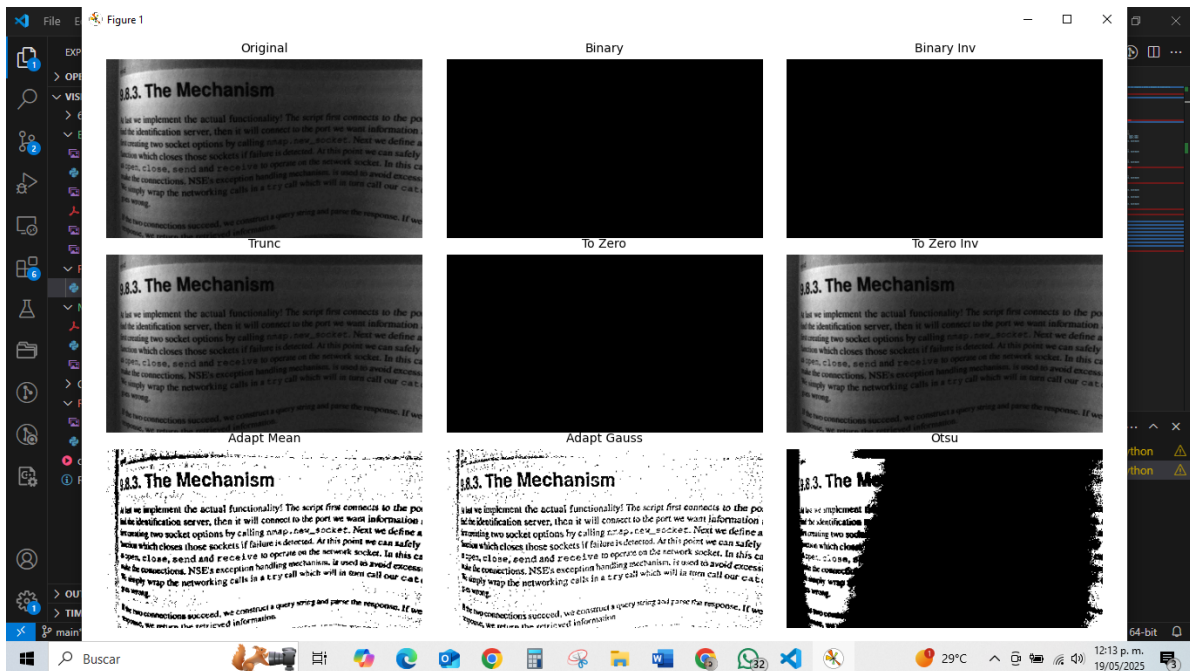
**Nombre: Ruiz Macías Luis Enrique - 21310196**

**Grado/Grupo: 6°G**

**Objetivo:** Mediante el uso de los diferentes umbrales, hacer regresión y modificar una imagen con el fin de recuperar la información escrita en una pagina de libro de una imagen.

En el presente código se muestra una imagen proporcionada previamente con la finalidad de realizar una aplicación de diferentes filtros, esto para formar una nueva imagen con diferentes características, con el fin de mostrar la información de la página del libro.

Esto se logra mediante la detención de umbrales y mediante el destacamento de fondos y relieves o bordes de la imagen.



A continuación, se muestra el código presente:

```
import cv2
import matplotlib.pyplot as plt

# Cargar imagen en escala de grises
img = cv2.imread('bookpage.jpg', cv2.IMREAD_GRAYSCALE)
img = cv2.resize(img, None, fx=0.5, fy=0.5)
if img is None:
    print("Error: No se pudo cargar la imagen")
    exit()
```

# Umbral fijo para métodos básicos

umbral = 127

max\_valor = 255

# 1. Binary

\_, th\_binary = cv2.threshold(img, umbral, max\_valor, cv2.THRESH\_BINARY)

# 2. Binary Invertido

\_, th\_binary\_inv = cv2.threshold(img, umbral, max\_valor, cv2.THRESH\_BINARY\_INV)

# 3. Truncado

\_, th\_trunc = cv2.threshold(img, umbral, max\_valor, cv2.THRESH\_TRUNC)

# 4. To Zero

\_, th\_tozero = cv2.threshold(img, umbral, max\_valor, cv2.THRESH\_TOZERO)

# 5. To Zero Invertido

\_, th\_tozero\_inv = cv2.threshold(img, umbral, max\_valor, cv2.THRESH\_TOZERO\_INV)

# 6. Umbral adaptativo - Media

th\_mean = cv2.adaptiveThreshold(img, max\_valor, cv2.ADAPTIVE\_THRESH\_MEAN\_C, cv2.THRESH\_BINARY, 11, 2)

# 7. Umbral adaptativo - Gaussiano

```
th_gauss = cv2.adaptiveThreshold(img, max_valor,
cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
cv2.THRESH_BINARY, 11, 2)
```

```
# 8. Otsu (automático)
```

```
_, th_otsu = cv2.threshold(img, 0, max_valor, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)
```

```
# ----- Mostrar en una sola ventana con matplotlib -----
```

```
titulos = [
```

```
    'Original', 'Binary', 'Binary Inv', 'Trunc',
```

```
    'To Zero', 'To Zero Inv', 'Adapt Mean', 'Adapt Gauss', 'Otsu'
```

```
]
```

```
imagenes = [
```

```
    img, th_binary, th_binary_inv, th_trunc,
```

```
    th_tozero, th_tozero_inv, th_mean, th_gauss, th_otsu
```

```
]
```

```
plt.figure(figsize=(12,8))
```

```
for i in range(9):
```

```
    plt.subplot(3, 3, i+1)
```

```
    plt.imshow(imagenes[i], cmap='gray')
```

```
    plt.title(titulos[i], fontsize=10)
```

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

```
plt.tight_layout(pad=2.0) # Ajusta los márgenes entre subplots
```

```
plt.show()
```

