

Reporte de Práctica: Procesamiento de Imágenes con OpenCV

Nombre: Eduardo Correa Flores

Materia: Visión Artificial

Tema: Umbralización de imágenes

Lenguaje: Python

Objetivo:

Aprender a aplicar diferentes técnicas de umbralización usando OpenCV para:

1. Convertir imágenes a escala de grises.
 2. Aplicar umbralización global y adaptativa.
 3. Comparar los resultados de distintos métodos de binarización.
-

Descripción de la Práctica:

Se utilizó la imagen bookpage.jpg para realizar los siguientes procesos de umbralización:

1. Carga de la Imagen Original

python

Copy

Download

```
img = cv2.imread('bookpage.jpg')
```

- La imagen se carga en formato **BGR** (color por defecto en OpenCV).

2. Umbralización Global Simple

python

Copy

Download

```
retval, threshold = cv2.threshold(img, 12, 255, cv2.THRESH_BINARY)
```

- **Parámetros:**
 - 12: Valor de umbral.

- 255: Valor máximo para píxeles que superan el umbral.
- cv2.THRESH_BINARY: Tipo de umbralización (binaria).
- **Resultado:** Imagen binarizada donde los píxeles >12 se convierten a 255 (blanco) y el resto a 0 (negro).

3. Conversión a Escala de Grises

python

Copy

Download

```
grayscaled = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

- La imagen se convierte a escala de grises para aplicar métodos que requieren un solo canal.

4. Umbralización Global en Escala de Grises

python

Copy

Download

```
retval2, threshold2 = cv2.threshold(grayscaled, 12, 255, cv2.THRESH_BINARY)
```

- Similar al primer umbral, pero aplicado a la versión en grises.

5. Umbralización Adaptativa Gaussiana

python

Copy

Download

```
gaus = cv2.adaptiveThreshold(grayscaled, 255,  
cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 115, 1)
```

- **Parámetros:**
 - 115: Tamaño del bloque (ventana para calcular el umbral local).
 - 1: Constante restada del umbral calculado.
- **Ventaja:** Mejor para imágenes con iluminación variable.

6. Método de Otsu

python

Copy

Download

```
retval3, otsu = cv2.threshold(grayscaled, 125, 255, cv2.THRESH_BINARY +  
cv2.THRESH_OTSU)
```

- **Características:**
 - Calcula automáticamente el umbral óptimo.
 - Ideal para imágenes con histogramas bimodales.

7. Visualización de Resultados

python

Copy

Download

```
cv2.imshow('original', img)  
cv2.imshow('threshold', threshold)  
cv2.imshow('threshold2', threshold2)  
cv2.imshow('gaus', gaus)  
cv2.imshow('otsu', otsu)  
cv2.waitKey(0)  
cv2.destroyAllWindows()
```

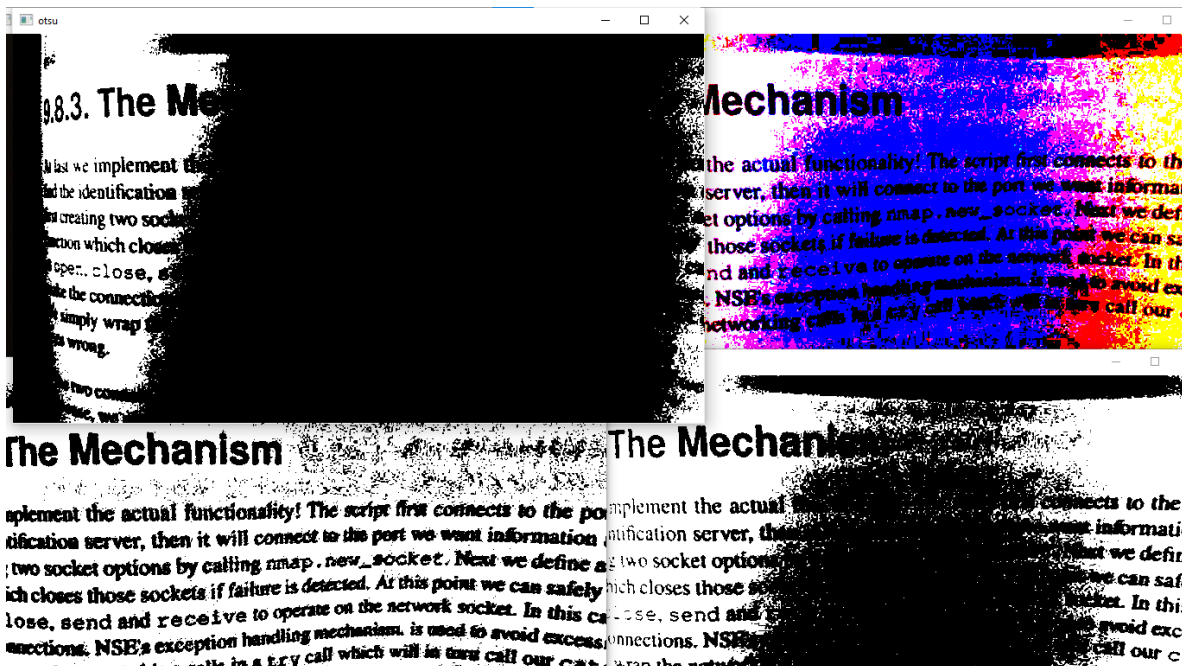
- Se muestran todas las imágenes procesadas en ventanas separadas.

Resultados Obtenidos:

Imagen	Descripción
--------	-------------

original	Imagen a color sin modificar.
----------	-------------------------------

Imagen	Descripción
threshold	Binarización global (color). Los píxeles >12 son blancos.
threshold2	Binarización global (escala de grises).
gaus	Umbralización adaptativa (Gaussiana). Efectivo en zonas con sombras.
otsu	Método automático de Otsu. Umbral óptimo sin necesidad de especificarlo.



Conclusión:

- La **umbralización global** es simple pero limitada para imágenes con iluminación irregular.
- La **umbralización adaptativa** (Gaussiana) ofrece mejores resultados en condiciones variables.

- **Otsu** es ideal cuando no se conoce el valor de umbral óptimo, ya que lo calcula automáticamente.