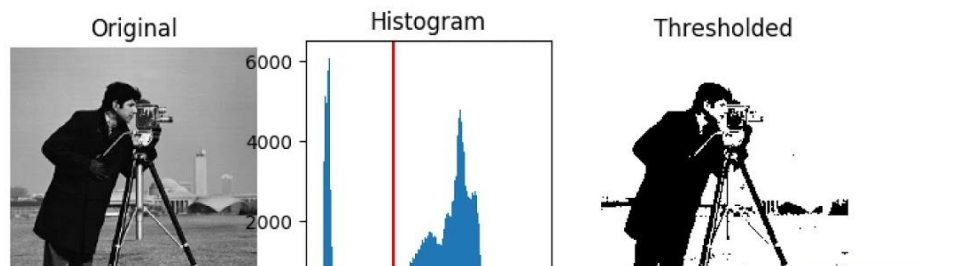


**Integrantes:** Jhenry Lopez – Eduardo Coro

**Fecha:**19/03/24

## Método Otsu



El método de Otsu es una técnica de segmentación de imágenes utilizada para separar automáticamente los objetos y el fondo en una imagen en escala de grises. Fue desarrollado por Nobuyuki Otsu en 1979 y es ampliamente utilizado en el procesamiento digital de imágenes y visión por computadora.

La idea principal detrás del método de Otsu es encontrar el umbral óptimo que maximiza la varianza entre las clases de píxeles de primer plano y de fondo en una imagen.

### Explicación

El umbral óptimo separa la imagen en dos clases, una que contiene píxeles de primer plano y otra que contiene píxeles de fondo. Imagina que estás mirando una foto en blanco y negro, quieres separar las partes claras de las partes oscuras automáticamente, sin tener que hacerlo manualmente. El método de Otsu te ayuda a hacer esto.

Primero, mira la foto y trata de encontrar un valor que divida claramente las partes claras de las oscuras, el método de Otsu hace esto automáticamente. Imagina que tienes una línea que atraviesa la foto y divides los píxeles en dos grupos: los que están más allá de la línea son oscuros y los que están antes son claros.

Para encontrar este umbral óptimo, el método de Otsu calcula la varianza intra-clase para todos los posibles umbrales y luego selecciona el umbral que minimiza la varianza intra-clase o, de manera equivalente, maximiza la varianza inter-clase.

## Ejemplo de Aplicación de Código en Python

```
import cv2

# Cargar imagen en escala de grises
img = cv2.imread('Mono.png', 0)

# Aplicar umbral de Otsu
_, thresh = cv2.threshold(img, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)

# Crear ventana con el tamaño de la imagen
cv2.namedWindow('Imagen binarizada', cv2.WINDOW_NORMAL)
cv2.resizeWindow('Imagen binarizada', 600, 600) # ajusta el tamaño como desees

# Mostrar imagen binarizada
cv2.imshow('Imagen binarizada', thresh)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

[3] ✓ 5.1s

## Resultado aplicando Otsu



## Código

```
import cv2

# Cargar imagen en escala de grises
img = cv2.imread('Monin.jpeg', 0)

# Aplicar umbral de Otsu
_, thresh = cv2.threshold(img, 0, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)

# Crear ventana con el tamaño de la imagen
cv2.namedWindow('Imagen binarizada', cv2.WINDOW_NORMAL)
cv2.resizeWindow('Imagen binarizada', 600, 600) # ajusta el tamaño como
desees
```

```
# Mostrar imagen binarizada
cv2.imshow('Imagen binarizada', thresh)
cv2.waitKey(0)
cv2.destroyAllWindows()
```