



INSTITUTO POLITÉCNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA DE
INGENIERÍA Y TECNOLOGÍAS AVANZADAS



CARRERA: INGENIERÍA TELEMÁTICA

UNIDAD DE APRENDIZAJE: MULTIMEDIA

PRACTIUCA: 3 Y 4

ALUMNO:

- GARNICA GONZALEZ CHRISTOPHER ALDAIR
- 2023640373

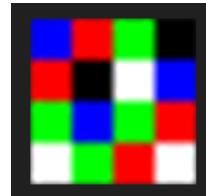
DOCENTE: NOE SIERRA ROMERO

FECHA: 12/02/25

GRUPO: 3TM2

Explicación Técnica: Binarización e Inversión de Imagen

El proceso de binarización es una técnica fundamental de segmentación de imágenes que simplifica la información visual, reduciendo la complejidad de 24 bits a solo 2 niveles de intensidad.



```
#Read and display data from BMP
file = open("./Images/example001.bmp", "rb")

firm = file.read(2)
print(firm)
file.seek(4)
pixel_first = file.read(3)
print(pixel_first)

file.seek(54,0)
no_pix = 0
while(True):
    pixel_data = file.read(3)
    if(len(pixel_data) > 0):
        print(pixel_data)
        no_pix += 3
    else:
        break
print("No Pixels: "+str(no_pix))
file.close()
```

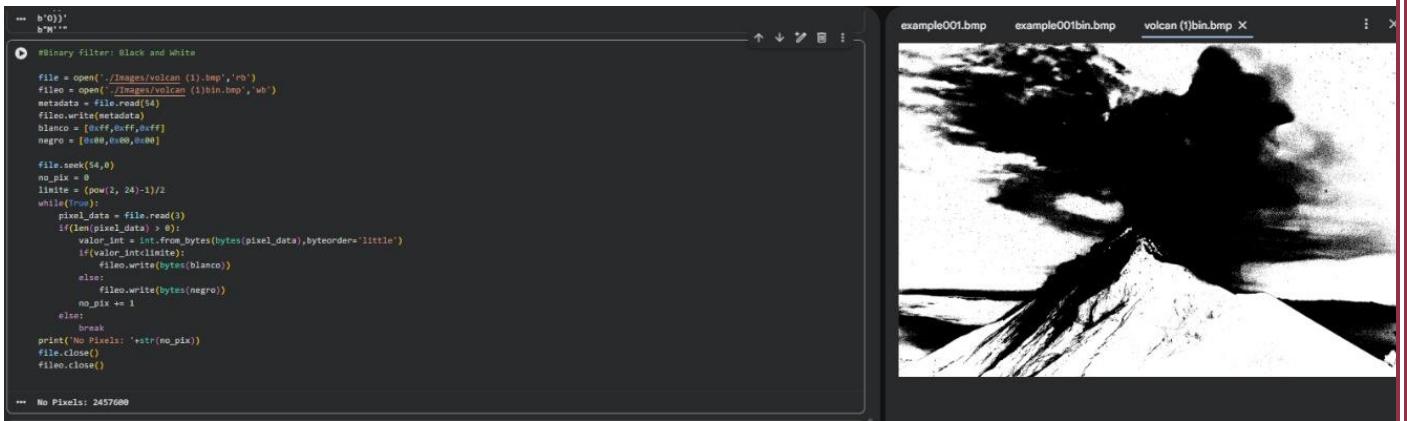
```
#Binary filter: Black and White
file = open("./Images/example001.bmp",'rb')
fileo = open("./Images/example001bin.bmp",'wb')
metadata = file.read(54)
fileo.write(metadata)
blanco = [0xFF,0xFF,0xFF]
negro = [0x00,0x00,0x00]

file.seek(54,0)
no_pix = 0
limite = (pow(2, 24)-1)/2
while(True):
    pixel_data = file.read(3)
    if(len(pixel_data) > 0):
        valor_int = int.from_bytes(bytes(pixel_data),byteorder='little')
        if(valor_int < limite):
            fileo.write(bytes(blanco))
        else:
            fileo.write(bytes(negro))
        no_pix += 3
    else:
        break
print("No Pixels: "+str(no_pix))
file.close()
fileo.close()
```

1. Binarización (Filtro Blanco y Negro)

Para convertir la imagen del volcán a blanco y negro, el algoritmo evalúa la intensidad total de cada píxel. Se establece un umbral crítico (normalmente el valor medio del rango):

- Lógica: Si el valor entero del píxel es mayor al umbral, se asigna Blanco (0xFFFFFFFF). Si es menor, se asigna Negro (0x0000000).
- Propósito: Esta técnica es esencial para el reconocimiento de formas y la extracción de características, ya que separa el "objeto" (el volcán y la lava brillante) del "fondo" (el cielo o las sombras).



The terminal window shows the following code:

```
... b'0')'
b'M'...
#Binary filter: Black and White

file = open('Imagen/volcan (1).bmp','rb')
fileo = open('Imagen/volcan (1)bin.bmp','wb')
metadata = file.read(54)
fileo.write(metadata)
blanco = [0xFF,0xFF,0xFF]
negro = [0x00,0x00,0x00]

file.seek(54,0)
no_pix = 0
limite = (pow(2, 24)-1)/2
while(1):
    pixel_data = file.read(3)
    if(len(pixel_data) > 0):
        valor_int = int.from_bytes(bytes(pixel_data),byteorder='little')
        if(valor_int>limite):
            fileo.write(bytes(blanco))
        else:
            fileo.write(bytes(negro))
        no_pix += 1
    else:
        break
print("No Pixels: "+str(no_pix))
file.close()
fileo.close()

... No Pixels: 2457600
```

The terminal also displays the output: "No Pixels: 2457600".

Two images are shown in the terminal window:

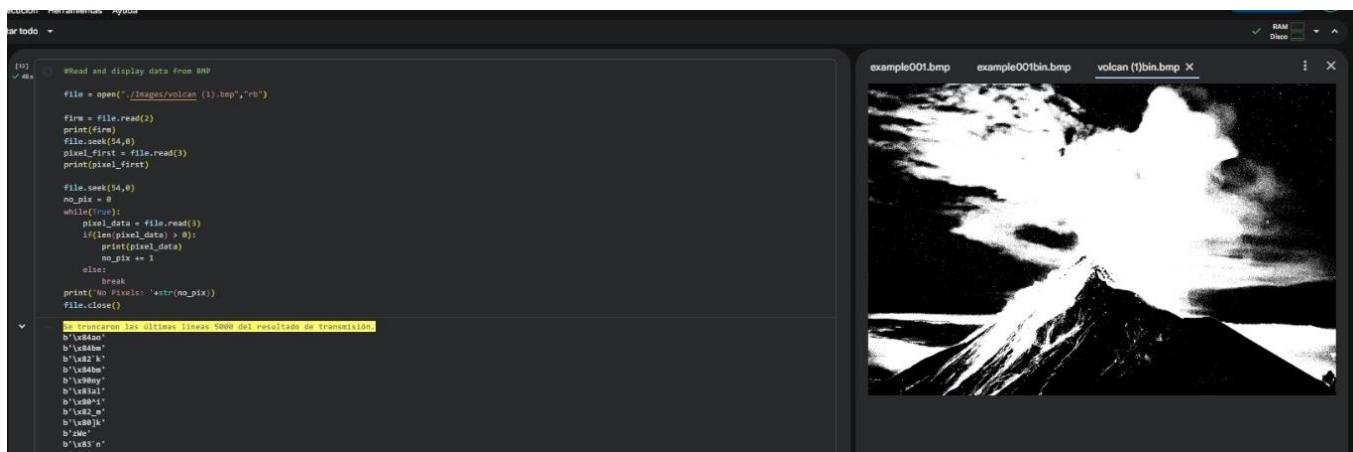
- example001.bmp: The original color image of a volcano.
- example001bin.bmp: The binary (black and white) version of the image.
- volcan (1)bin.bmp: The binary version of the image, which appears identical to example001bin.bmp.



2. Inversión de Colores (Negativo)

En el caso de la inversión ("viceversa"), el algoritmo aplica un operador de complemento:

- Lógica: Se resta el valor del píxel actual al valor máximo posible (255). En términos binarios, esto equivale a aplicar un operador NOT.
- Resultado: Las zonas de alta luminiscencia (la ceniza y el humo blanco) se convierten en las zonas más oscuras, permitiendo identificar detalles que en la imagen original podrían estar saturados por el brillo.



```
#!/usr/bin/python
#Read and display data from BMP
file = open("./Images/volcan (1).bmp","rb")
firs = file.read(2)
file.seek(54,0)
pixelsFirst = file.read(3)
print(pixelsFirst)

file.seek(54,0)
no_pix = 0
while(True):
    pixel_data = file.read(3)
    if(len(pixel_data) > 0):
        print(pixel_data)
        no_pix += 1
    else:
        break
print("No Pixels: "+str(no_pix))
file.close()

#se truncaron las ultimas lineas: 5000 del resultado de transmision
b'\x8d\xaa'
b'\x8d\xbb'
b'\x8d\xcc'
b'\x8d\xdd'
b'\x8e\xee'
b'\x8e\xff'
b'\x8f\x00'
b'\x8f\x01'
b'\x8f\x02'
b'\x8f\x03'
b'\x8f\x04'
b'\x8f\x05'
b'\x8f\x06'
b'\x8f\x07'
b'\x8f\x08'
b'\x8f\x09'
b'\x8f\x0a'
b'\x8f\x0b'
b'\x8f\x0c'
b'\x8f\x0d'
b'\x8f\x0e'
b'\x8f\x0f'
b'\x8f\x10'
b'\x8f\x11'
b'\x8f\x12'
b'\x8f\x13'
b'\x8f\x14'
b'\x8f\x15'
b'\x8f\x16'
b'\x8f\x17'
b'\x8f\x18'
b'\x8f\x19'
b'\x8f\x1a'
b'\x8f\x1b'
b'\x8f\x1c'
b'\x8f\x1d'
b'\x8f\x1e'
b'\x8f\x1f'
b'\x8f\x20'
b'\x8f\x21'
b'\x8f\x22'
b'\x8f\x23'
b'\x8f\x24'
b'\x8f\x25'
b'\x8f\x26'
b'\x8f\x27'
b'\x8f\x28'
b'\x8f\x29'
b'\x8f\x2a'
b'\x8f\x2b'
b'\x8f\x2c'
b'\x8f\x2d'
b'\x8f\x2e'
b'\x8f\x2f'
b'\x8f\x2g'
b'\x8f\x2h'
b'\x8f\x2i'
b'\x8f\x2j'
b'\x8f\x2k'
b'\x8f\x2l'
b'\x8f\x2m'
b'\x8f\x2n'
b'\x8f\x2o'
b'\x8f\x2p'
b'\x8f\x2q'
b'\x8f\x2r'
b'\x8f\x2s'
b'\x8f\x2t'
b'\x8f\x2u'
b'\x8f\x2v'
b'\x8f\x2w'
b'\x8f\x2x'
b'\x8f\x2y'
b'\x8f\x2z'
```



Resumen de la Práctica 3 y 4

Estas prácticas se centraron en la **reducción extrema de datos** y la **manipulación de la luminancia**. Se logró transformar la imagen del volcán mediante lógica booleana aplicada directamente a los bytes del archivo BMP.

- **Impacto en Multimedia:** Se comprendió que la binarización no es solo un efecto visual, sino una herramienta de pre-procesamiento para sistemas de visión artificial.
- **Relación con la Esteganografía:** El manejo de umbrales en blanco y negro es el principio básico para ocultar "marcas de agua" o mensajes ocultos en canales específicos de una imagen (como el canal alfa o el bit menos significativo), donde la información se codifica en estados de presencia/ausencia de señal.