



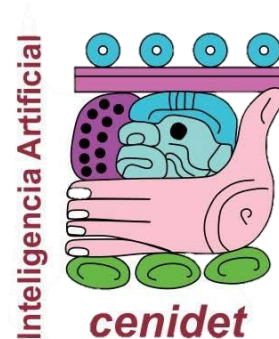
TECNOLÓGICO
NACIONAL DE MÉXICO

cenidet[®]
Centro Nacional de Investigación
y Desarrollo Tecnológico



Introducción al procesamiento digital de imágenes

M. En C. Daniel Flores Guerrero



11 de agosto del 2025
Cuernavaca Mor.

Contenido

01

Procesamiento de imágenes

02

Aplicaciones

03

Historia y evolución

04

Fundamentos y representación
de imágenes digitales

05

Muestreo y cuantización

06

Tipos de imágenes

07

Escalas de color

08

Librerías para procesamiento de
imágenes

09

Operaciones básicas en
imágenes

10

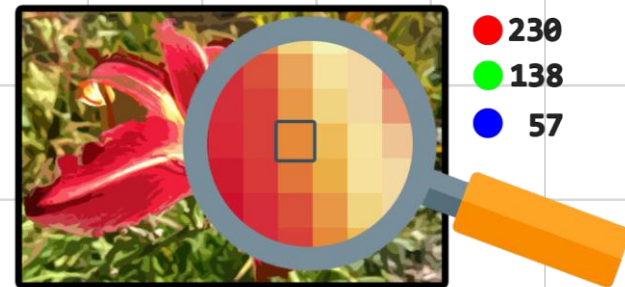
Histograma

11

Técnicas básicas de
manipulación

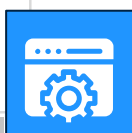
1. Procesamiento de imágenes

El Procesamiento Digital de Imágenes (PDI) implica la manipulación de imágenes digitales utilizando computadoras para transformarlas o extraer información de utilidad de ellas a través de operaciones y algoritmos.



Procesos de bajo nivel:

Tareas primitivas como reducción de ruido, mejora de contraste o nitidez. Tanto entradas como salidas son imágenes.



Procesos de nivel medio:

Segmentación, descripción de objetos y clasificación de regiones. La entrada suele ser una imagen y la salida atributos.

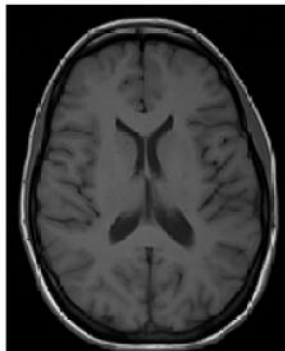


Procesos de alto nivel:

Interpretación de objetos reconocidos, tareas de visión por computadora.

2. Aplicaciones

Original MRI



After Skull Stripping

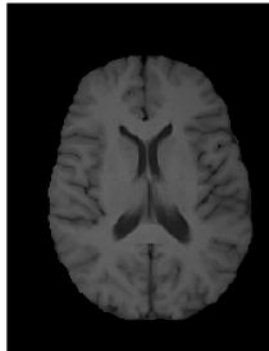
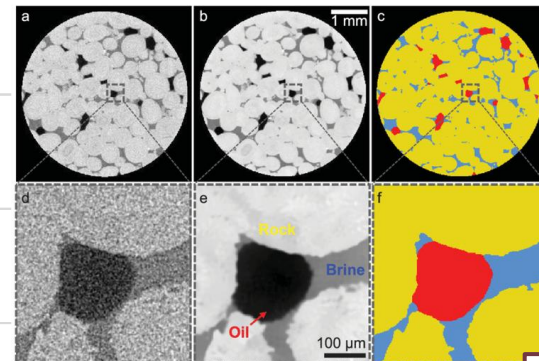
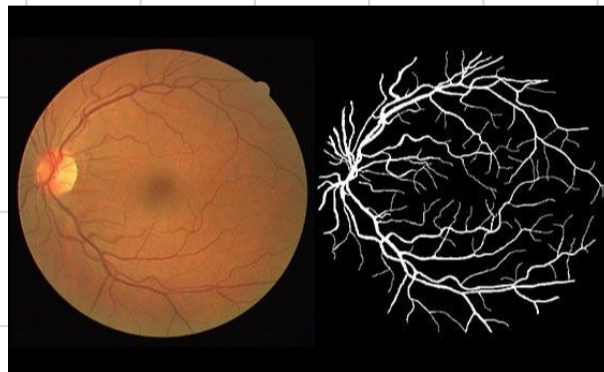


Image after segmentation



Image after segmentation and
morphological processing



3. Historia y evolución

Se introducen los conceptos de computadora digital moderna gracias a John Von Neumann, sentando las bases para el procesamiento de imágenes mediante programas almacenados y control secuencial.

1940

Se desarrollan técnicas de procesamiento digital para aplicaciones médicas.

1970

El crecimiento de Internet y la digitalización masiva impulsa el uso del procesamiento de imágenes en comunicación, multimedia, cámaras digitales y videovigilancia.

1990

1920

1964

La NASA utiliza procesamiento digital para mejorar imágenes lunares captadas por la sonda Ranger 7, marcando la primera aplicación espacial y automatizada de este tipo de procesamiento.

1964

Con el abaratamiento del cómputo, el procesamiento digital de imágenes se expande a nuevas áreas como geografía, astronomía y arqueología.

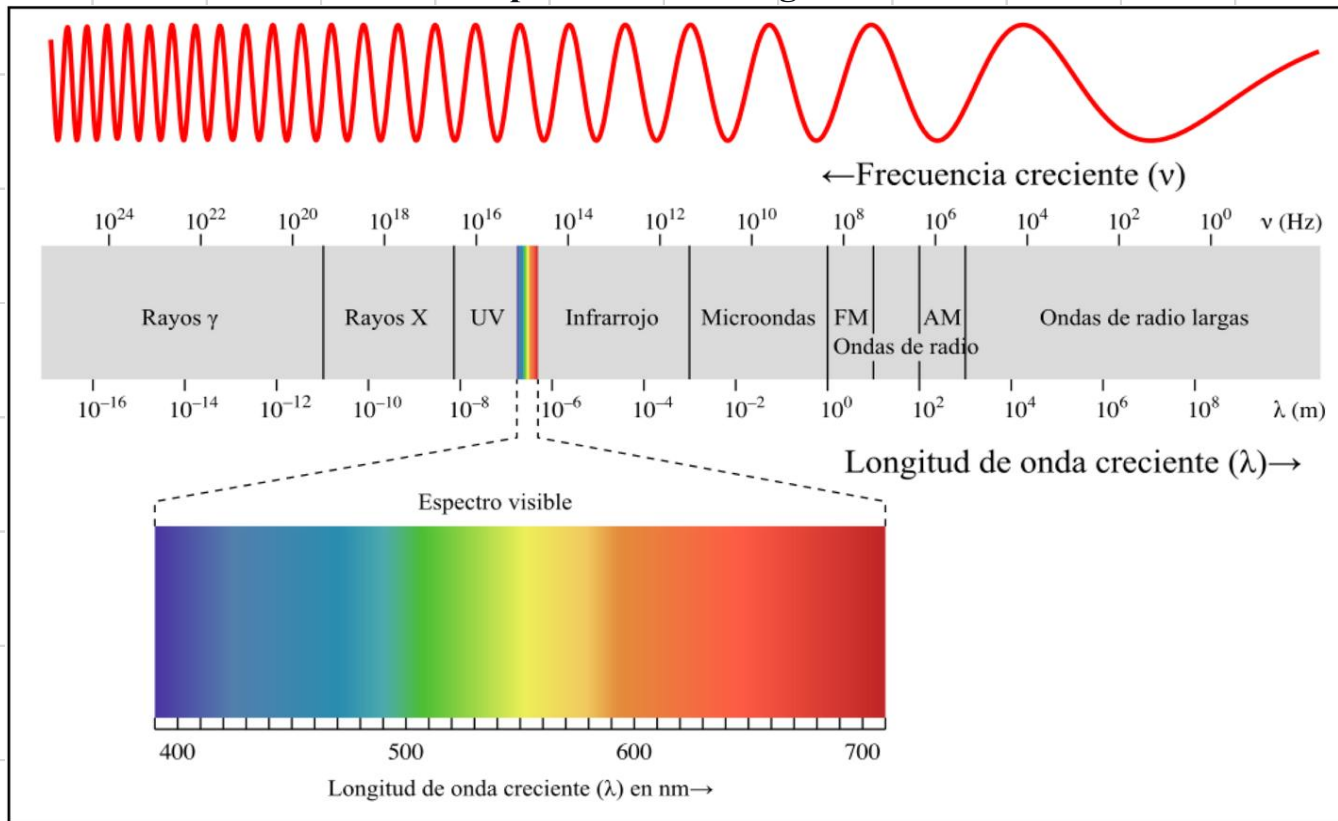
2000 - Presente

Se integra el procesamiento digital de imágenes en IA y visión por computadora, aplicándose en reconocimiento facial, vehículos autónomos, salud, industria y más.

Se envían imágenes digitales a través del atlántico desde Londres hasta Nueva York a través de un cable submarino (Bartlane) para la industria periodística pasando de una semana a 3 hrs.

4. Fundamentos y representación de imágenes digitales

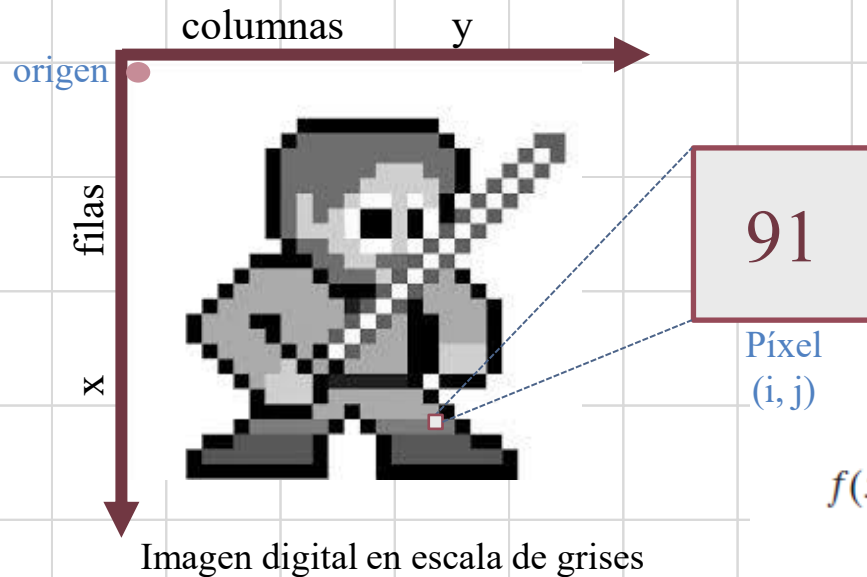
Espectro electromagnético



4. Fundamentos y representación de imágenes digitales

Pixel: Unidad más pequeña de una imagen digital

Imagen digital: Matriz de valores que representan la intensidad luminosa o la intensidad de color de un píxel.



Los valores estarán cuantizados en L niveles finitos y discretos de intensidad.

$$L = 2^k$$

Usualmente 8 bits

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

Matriz de Valores

4. Fundamentos y representación de imágenes digitales

Tensor: es básicamente una **caja de datos con varias dimensiones** que se utiliza para representar imágenes. Es como un contenedor que guarda números, pero la forma en que los organiza depende de cuántas dimensiones tenga:

0D → Escalar: Un solo número. Ejemplo: 7.

1D → Vector: Una fila de números. Ejemplo: [3, 5, 8].

2D → Matriz: Una tabla de números. Ej.: $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$

o dimensiones superiores.

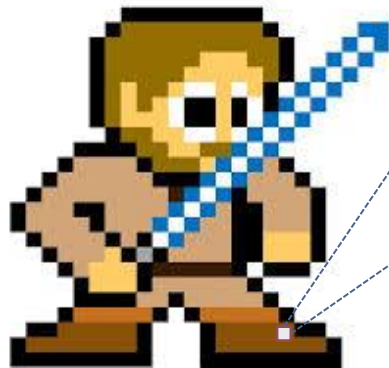
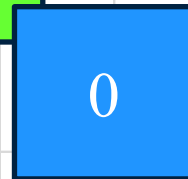
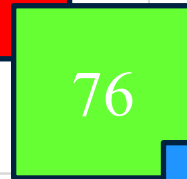
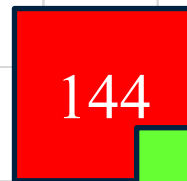
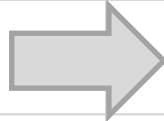


Imagen digital en RGB



Píxel

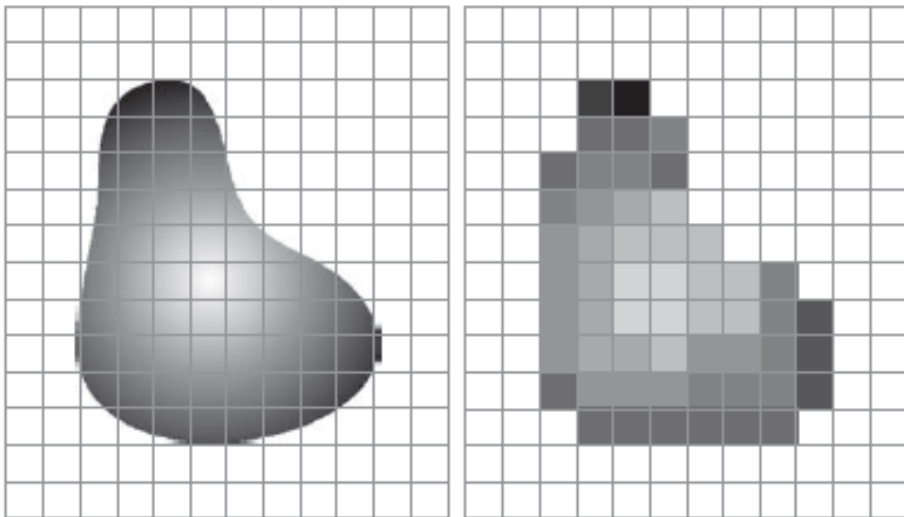


Valores del píxel en RGB

Idea clave: un tensor es la versión “pro” de la idea de vector y matriz, extendida a tantas dimensiones como quieras.

5. Muestreo y cuantización

Para crear una imagen digital se requiere convertir datos continuos a un formato digital, esto implica un proceso de muestreo y cuantización.

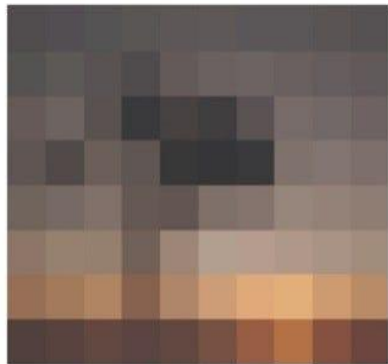


El muestreo viene determinado por la disposición y cantidad de sensores utilizados para generarla imagen.
-> Número de píxeles (resolución).

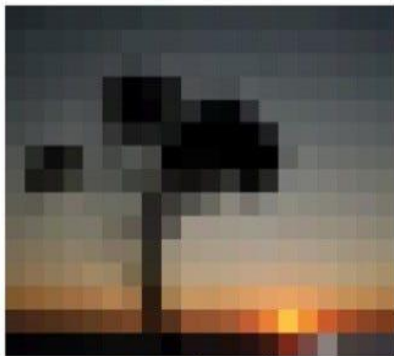
La digitalización de los valores de amplitud se denomina cuantización.
-> Número de colores o niveles de brillo por píxel.

La calidad de la imagen está determinada en parte por el número de muestras y los niveles discretos de intensidad utilizados en el muestreo y la cuantización.

Resolución



10 px de ancho



20 px de ancho



50 px de ancho



150 px de ancho



250 px de ancho



350px de ancho

Indica la cantidad de detalles que puede observarse en esta. El término es comúnmente utilizado en relación con imágenes, pero también se utiliza para describir cuán nítida es (calidad visual).

La resolución depende de:

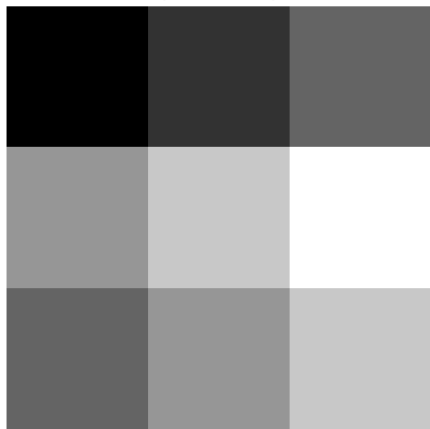
- El sensor
- Profundidad de bits
- Compresión
- Interpolación y reescalado

6. Tipos de imágenes

Escala de grises: Imágenes sin información de color, los píxeles varían en el rango del negro al blanco pasando por tonos de grises intermedios. Ejemplo: Radiografías.

A color: Imágenes con información de color, usualmente representadas en los modelos *RGB*, *CMY* o *HSI*.

Binarias: Imágenes con únicamente dos valores posibles por píxel, **blanco** y **negro**. Ejemplo. máscaras, texto.



Escala de grises Matriz 2D

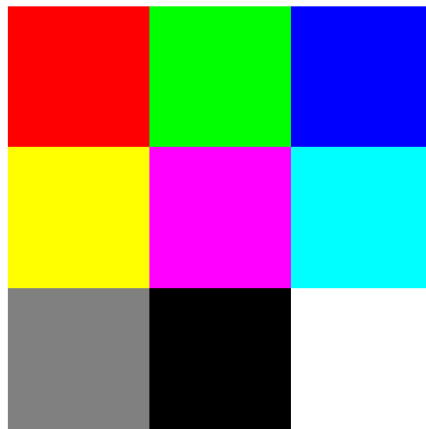


Imagen a color (RGB)
Tensor 3D

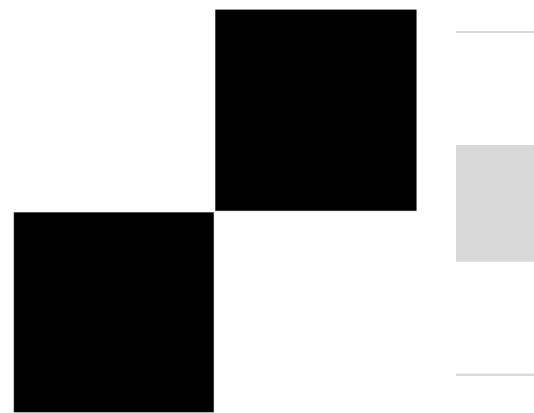


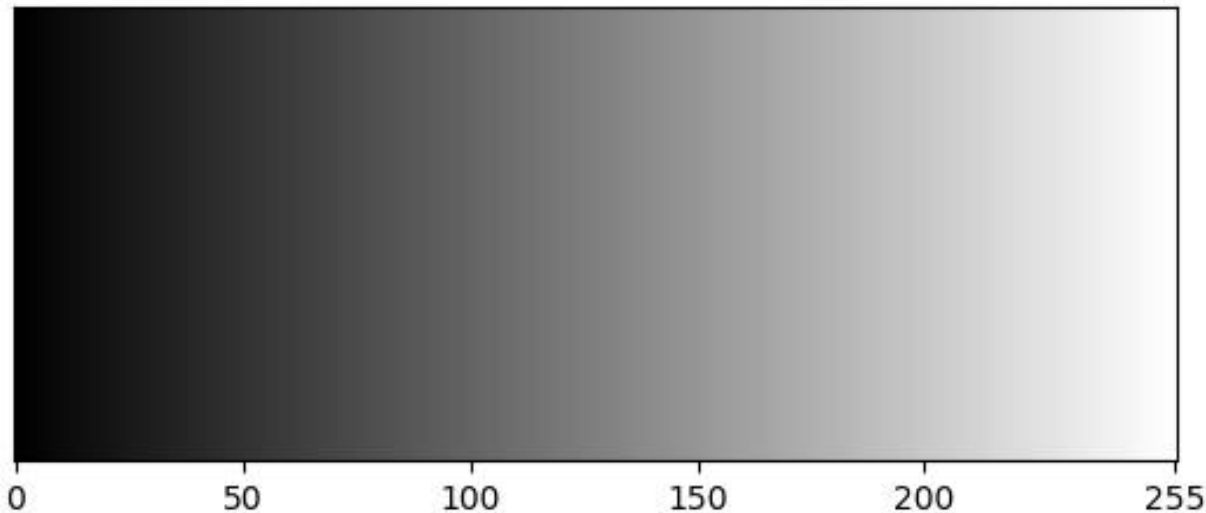
Imagen binaria
Matriz 2D

7. Escalas de color

Escala de grises

Modelo en el que cada píxel de una imagen tiene un único valor de intensidad que varía entre el negro y el blanco. Este valor suele expresarse en un rango de 0 (negro) a 255 (blanco) en imágenes de 8 bits. Las imágenes en escala de grises no contienen información de color, lo que las hace más simples y menos costosas computacionalmente.

$$L = 2^8 = 256$$



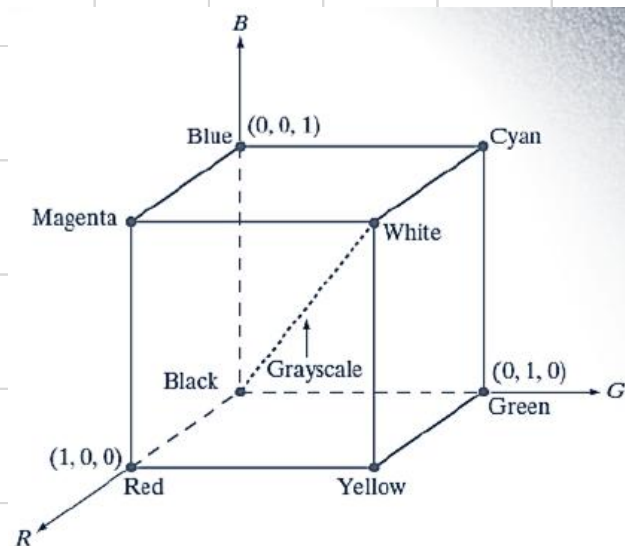
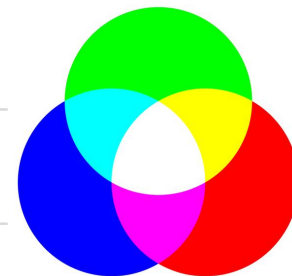
7. Escalas de color

Modelo RGB

El modelo **RGB** (**R**ed, **G**reen, **B**lue) es un sistema de representación del color basado en la combinación de tres colores primarios: rojo, verde y azul. Cada color en una imagen RGB se forma mediante la mezcla de distintas intensidades de estos tres componentes, típicamente en un rango de **0** a **255**.

Este modelo es aditivo, lo que significa que al combinar los tres colores a máxima intensidad se obtiene el blanco.

Es el modelo mas común en dispositivos digitales como monitores, cámaras y sensores.



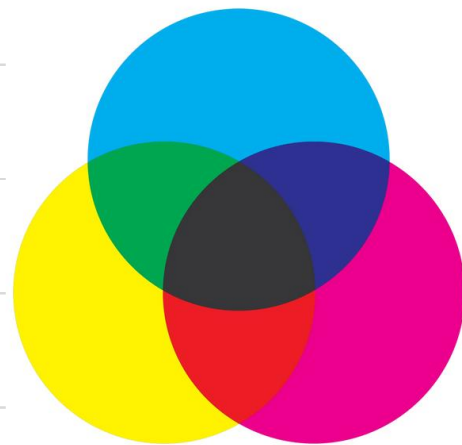
7. Escalas de color

Modelo CMY

El modelo CMY (*Cyan, Magenta, Yellow*) está basado en los colores secundarios del modelo de luz (RGB) y primarios para pigmentos, es un modelo de color sustractivo utilizado principalmente en la impresión.

En este modelo, los colores se crean restando luz de un fondo blanco.

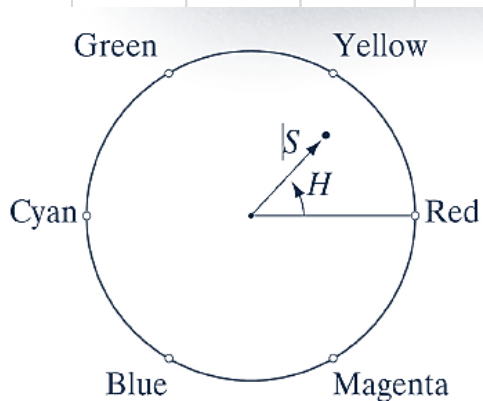
CMYK: Incorpora el componente negro (K), ya que en la práctica la mezcla de colores en CMY resulta en un gris oscuro.



7. Escalas de color

Modelo HSI

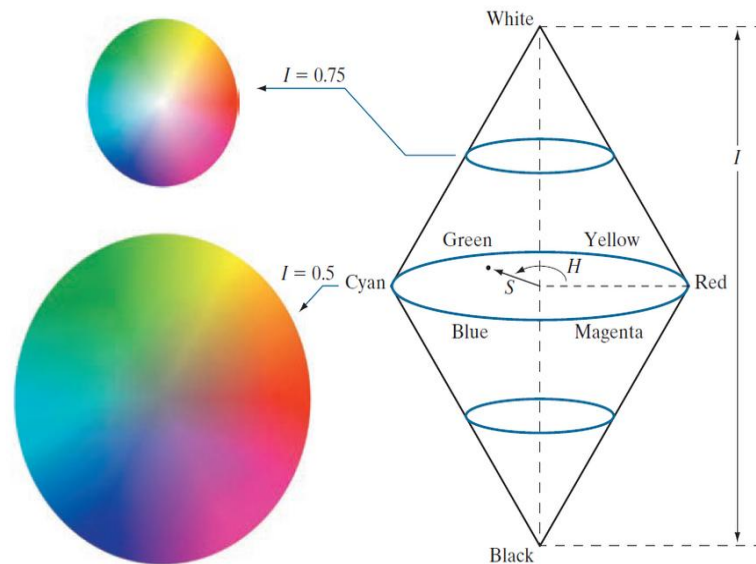
El modelo HSI (*Hue, Saturation, Intensity*) está basado en como los humanos observamos y describimos el color, en términos de Tono, Saturación e Intensidad (brillo).



Tono: Describe un color puro.

Saturación: Mide que tanto se diluye el color puro con luz blanca.

Intensidad: Representa la cantidad de luz percibida.



El modelo separa la información de color (tono y saturación) de la de intensidad.

Tipos de formato de imagen

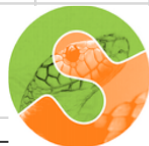
Para crear una imagen digital se requiere convertir datos continuos a un formato digital, esto implica un proceso de muestreo y cuantización.

Formato	Extensión	Compresión	Transparencia	Uso típico
JPEG	.jpeg/.jpg	Con pérdida	No	Tamaño reducido en espacio, pero con degradación de calidad de imagen (Fotografías, imágenes complejas)
PNG	.png	Sí	Sí	Gráficos, logos, transparencias
TIFF	.tiff	Sí	Sí	Impresión profesional, archivos
GIF	.gif	Sí	Sí	Animaciones, gráficos simples
BMP	.bmp	Sí	Sí	Archivos no comprimidos, pero son apropiados para transferencia en internet.

8. Librerías para procesamiento de imágenes



Escrita nativamente en C++. Puede emplearse para detección de objetos, reconocimiento facial, análisis de movimiento, procesamiento de video, etc. Su sintaxis puede resultar poco intuitiva.



scikit-image
image processing in python

Librería científica para el análisis y procesamiento de imágenes.

Utiliza *NumPy arrays* como estructura de datos fundamentales para representar imágenes como matrices y es compatible con NumPy, SciPy, matplotlib y scikit-learn.

Ofrece herramientas para segmentación, morfología, detección de bordes, transformaciones geométricas y extracción de características, entre otras funciones.



Librería orientada al manejo básico y procesamiento de imágenes. Permite abrir, guardar, redimensionar, rotar, recortar y aplicar efectos simples a imágenes con una sintaxis clara y directa. Rápido acceso a píxeles y visualización de imágenes.

9. Operaciones básicas en imágenes

Son aquellas manipulaciones fundamentales que permiten **analizar, mejorar o extraer información visual**. Son la base para tareas más avanzadas como detección de objetos, reconocimiento facial o clasificación de imágenes

1. Lectura y visualización

- Leer una imagen desde un archivo.
- Mostrarla en pantalla.
- **Librerías:** *OpenCV, PIL, Matplotlib.*

Python

```
import cv2  
img = cv2.imread('imagen.jpg')  
cv2.imshow('Imagen', img)  
cv2.waitKey(0)
```

9. Operaciones básicas en imágenes

2. Conversión de color

- Convertir entre espacios de color:
 - RGB \leftrightarrow Grayscale
 - RGB \leftrightarrow HSV
 - RGB \leftrightarrow LAB

Útil para simplificar la información o extraer características específicas.

Python

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

9. Operaciones básicas en imágenes

3. Transformaciones geométricas

- **Escalado** (resizing): cambiar el tamaño.
- **Rotación**: girar la imagen.
- **Traslación**: moverla en el plano.
- **Reflexión** (flip): espejar horizontal o vertical.
- **Interpolación bilineal/bicúbica** en procesos como zoom.

Python

```
img_resized = cv2.resize(img, (200, 200))  
img_flipped = cv2.flip(img, 1) # Horizontal
```

9. Operaciones básicas en imágenes

4. Operaciones aritméticas y lógicas

- Suma o resta de imágenes (por ejemplo, para detectar cambios).
- AND, OR, XOR entre imágenes binarias.
- Escalado de brillo o contraste.

Python

```
brighter = cv2.convertScaleAbs(img,  
alpha=1.2, beta=30)
```

9. Operaciones básicas en imágenes

5. Filtrado (convoluciones)

- **Suavizado (blur):** reduce ruido.
 - Filtros: media, gaussiano, mediana.
- **Realce de bordes:** resalta contornos.
 - Filtros: Sobel, Laplaciano, Canny.
- Se aplican mediante **kernels**.

6. Segmentación básica

- Umbralización (*thresholding*): separar fondo y objeto.
- Binarización de la imagen.
- Segmentación por color o intensidad.

7. Operaciones morfológicas

- Se aplican a imágenes binarias para limpiar o refinar formas.
- Erosión: elimina ruido pequeño.
- Dilatación: agranda objetos blancos.
- Apertura y cierre.

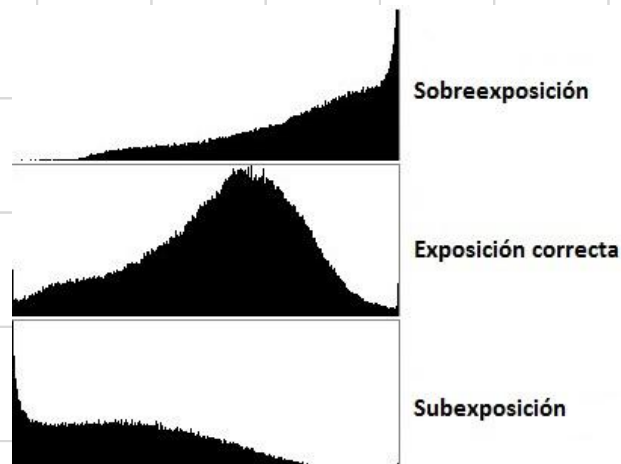
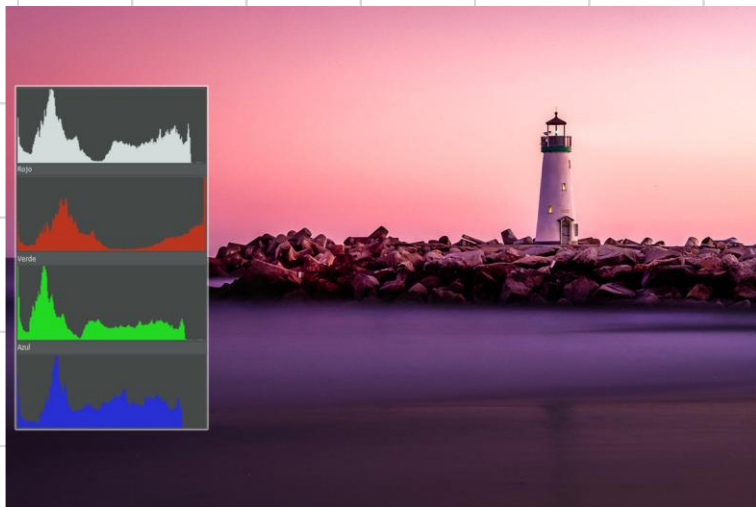
8. Detección de bordes

- Identifica cambios abruptos de intensidad.
- Operadores: *Sobel, Prewitt, Laplaciano, Canny*.

10. Histograma

El histograma de una imagen es una representación gráfica de la frecuencia de aparición de los diferentes niveles de intensidad (en escala de grises) o valores de color (en imágenes RGB).

- Ver la distribución de intensidades.
- Equalización del histograma: mejora el contraste



Ejemplo: Detección de imágenes subexpuestas o sobreexpuestas.

11. Técnicas básicas de manipulación

Umbralización y Binarización

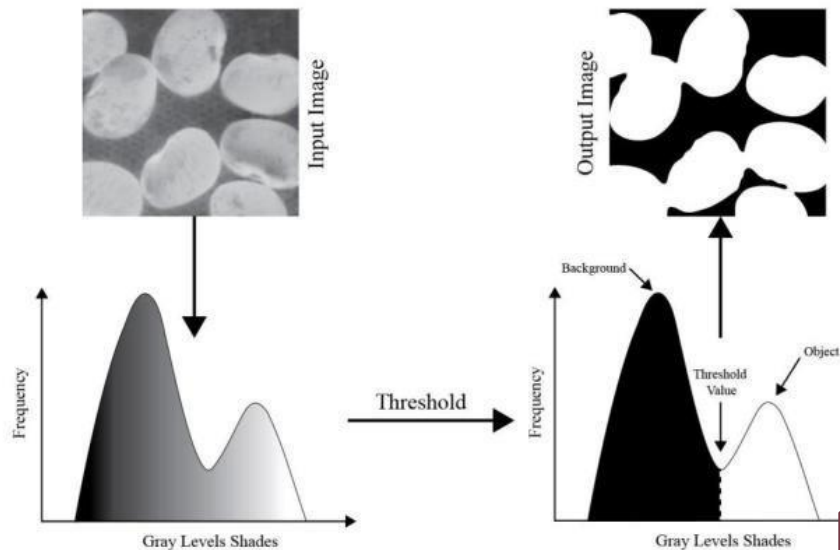
La umbralización una técnica de segmentación que convierte una imagen en escala de grises a una imagen binaria, comparando cada píxel con un valor umbral.

Para cada píxel $f(x, y)$

$$g(x, y) = \begin{cases} 1 & \text{si } f(x, y) \geq T \\ 0 & \text{si } f(x, y) < T \end{cases}$$

Tipos de umbral:

- Fijo
- Adaptativo

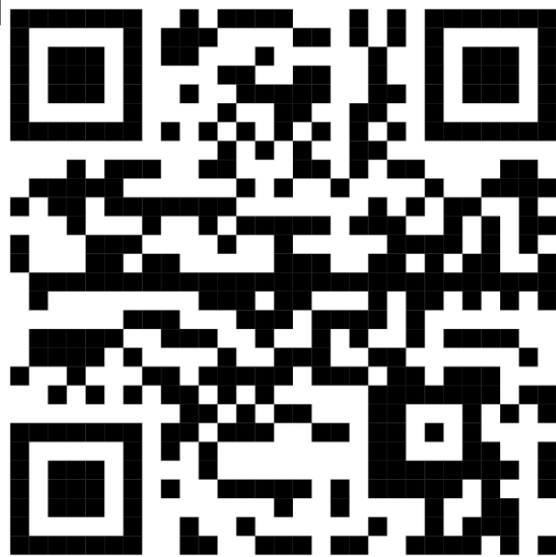


01

```
git clone  
git@github.com:iscwarrior/ProcesamientoDeImagenes2025.git
```



02



¿Por qué son importantes estas operaciones?

Estas operaciones permiten:

- Preparar la imagen para análisis posterior.
- Reducir ruido.
- Aumentar la calidad visual.
- Extraer características útiles para algoritmos de clasificación, detección, etc.



Referencias

- Gonzalez, R. C., & Woods, R. E. (2018). Digital Image Processing (4th ed.). Pearson.
- Zhang, Y., Liu, S., & Wang, S. (2022). A novel medical image fusion method based on structure-preserving filter and deep features. *Information Fusion*, 86, 40–52. <https://doi.org/10.1016/j.inffus.2022.05.005>
- Wang, G., Zhang, J., & Chen, Y. (2021). Deep learning for tomographic image reconstruction. *Nature Machine Intelligence*, 3(10), 943–952. <https://doi.org/10.1038/s42256-021-00407-z>
- Li, X., He, R., & Sun, Z. (2021). Learning discriminative deep face representation with visualization. *IEEE Transactions on Image Processing*, 30, 3697–3709. <https://doi.org/10.1109/TIP.2021.3071703>
- Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, 187, 27–48. <https://doi.org/10.1016/j.neucom.2015.09.116>
- Zhang, K., Zuo, W., & Zhang, L. (2017). FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Transactions on Image Processing*, 27(9), 4608–4622. <https://doi.org/10.1109/TIP.2018.2839891>
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 9351, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28
- Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600–612. <https://doi.org/10.1109/TIP.2003.819861>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 1097–1105.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. <https://doi.org/10.1109/CVPR.2016.90>



Referencias

- Szegedy, C., Liu, W., Jia, Y., et al. (2015). Going deeper with convolutions. CVPR 2015, 1–9. <https://doi.org/10.1109/CVPR.2015.7298594>
- Liu, Y., Zhang, D., Zhang, Y., & Wang, Y. (2023). Lightweight deep learning for real-time image super-resolution. IEEE Transactions on Multimedia, 25, 1896–1907. <https://doi.org/10.1109/TMM.2022.3147112>
- Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. CVPR 2009, 248–255. <https://doi.org/10.1109/CVPR.2009.5206848>
- Xu, Y., Mo, T., Feng, Q., Zhong, P., Lai, M., & Chang, E. I. (2014). Deep learning of feature representation with multiple instance learning for medical image analysis. IEEE Transactions on Neural Networks and Learning Systems, 25(5), 1019–1031. <https://doi.org/10.1109/TNNLS.2013.2297683>
- Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations (ICLR). <https://arxiv.org/abs/1409.1556>
- Buades, A., Coll, B., & Morel, J. M. (2005). A non-local algorithm for image denoising. CVPR 2005, 60–65. <https://doi.org/10.1109/CVPR.2005.38>
- Dong, C., Loy, C. C., He, K., & Tang, X. (2016). Image super-resolution using deep convolutional networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(2), 295–307. <https://doi.org/10.1109/TPAMI.2015.2439281>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. CVPR 2016, 779–788. <https://doi.org/10.1109/CVPR.2016.91>
- Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. CVPR 2017, 5967–5976. <https://doi.org/10.1109/CVPR.2017.632>
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al. (2014). Generative adversarial nets. Advances in Neural Information Processing Systems, 27, 2672–2680.



TECNOLÓGICO
NACIONAL DE MÉXICO

cenidet[®]
Centro Nacional de Investigación
y Desarrollo Tecnológico



Gracias



d17ce031@cenidet.tecnm.mx



LinkedIn: df-guerrero



GitHub: iscwarrior