

# Unidad 2 – Realce y Restauración de Imágenes

Tratamiento Digital de Imágenes

Sergio M. Nava Muñoz

s3rgio.nava@gmail.com

CIMAT/INFOTEC

2025-02-19

# Contenido

- Fin de aprendizaje
- Introducción
- Descomposición en canales RGB
- Representación de imágenes
- Visualización de un patch en RGB y sus canales
- Mezcla aditiva de colores – Modelo RGB
- Visualización de valores en escala de grises
- Histograma de una imagen
- Ecualización de histograma
- Ejemplo de ecualización de histograma
- Convolución en imágenes
- Filtrado espacial
- Filtro Promedio (Blur)
- Ejemplo paso a paso
- ¿Qué pasa en los bordes?

# Fin de aprendizaje

- Aplicar técnicas de mejora y restauración de imágenes digitales.
- Utilizar operaciones matriciales y filtros en OpenCV.
- Comprender fundamentos matemáticos de histogramas y filtrado.
- Preparar imágenes para tareas avanzadas de análisis visual.

# Introducción

- El realce mejora la apariencia **visual** de las imágenes.
- Facilita tareas como:
  - Detección de objetos
  - Reconocimiento de texturas (ej. imágenes médicas)
  - Segmentación y clasificación

# Descomposición en canales RGB

- Una imagen en color puede verse como **tres matrices en escala de grises**:
  - **Canal R (Rojo)**
  - **Canal G (Verde)**
  - **Canal B (Azul)**
- Cada canal almacena intensidades entre 0 y 255.
- La combinación de los tres forma la imagen en color.



Canal R



Canal G



Canal B



Imagen combinada

# Representación de imágenes

- Una imagen digital en escala de grises puede representarse como una **matriz**  $I$  de tamaño  $M \times N$ :

$$I = \begin{bmatrix} i_{11} & i_{12} & \cdots & i_{1N} \\ i_{21} & i_{22} & \cdots & i_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ i_{M1} & i_{M2} & \cdots & i_{MN} \end{bmatrix}$$

- Cada elemento  $i_{mn}$  es un valor de **intensidad** (0–255 en imágenes de 8 bits).

# Visualización de un patch en RGB y sus canales

## ► Code

### Canal R

```
[[171 137 145 137 130][151 137 136 123 105][142 158 154 141 112][160 182 157 141 114][173 170 122 106 99]]
```

### Canal G

```
[[124 94 105 105 104][104 94 99 91 82][ 97 117 119 111 89][118 143 125 114 93][132 134 91 81 80]]
```

### Canal B

```
[[70 41 53 54 55][50 39 46 40 32][40 61 65 57 37][60 86 68 59 40][76 76 34 25 24]]
```

### Juntos

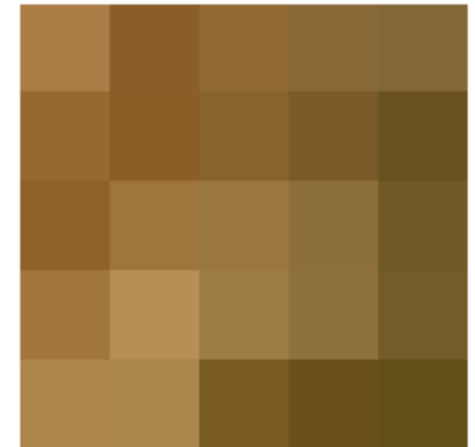
```
[[[171 124 70][137 94 41][145 105 53][137 105 54][130 104 55]]
```

```
[[151 104 50][137 94 39][136 99 46][123 91 40][105 82 32]]
```

```
[[142 97 40][158 117 61][154 119 65][141 111 57][112 89 37]]
```

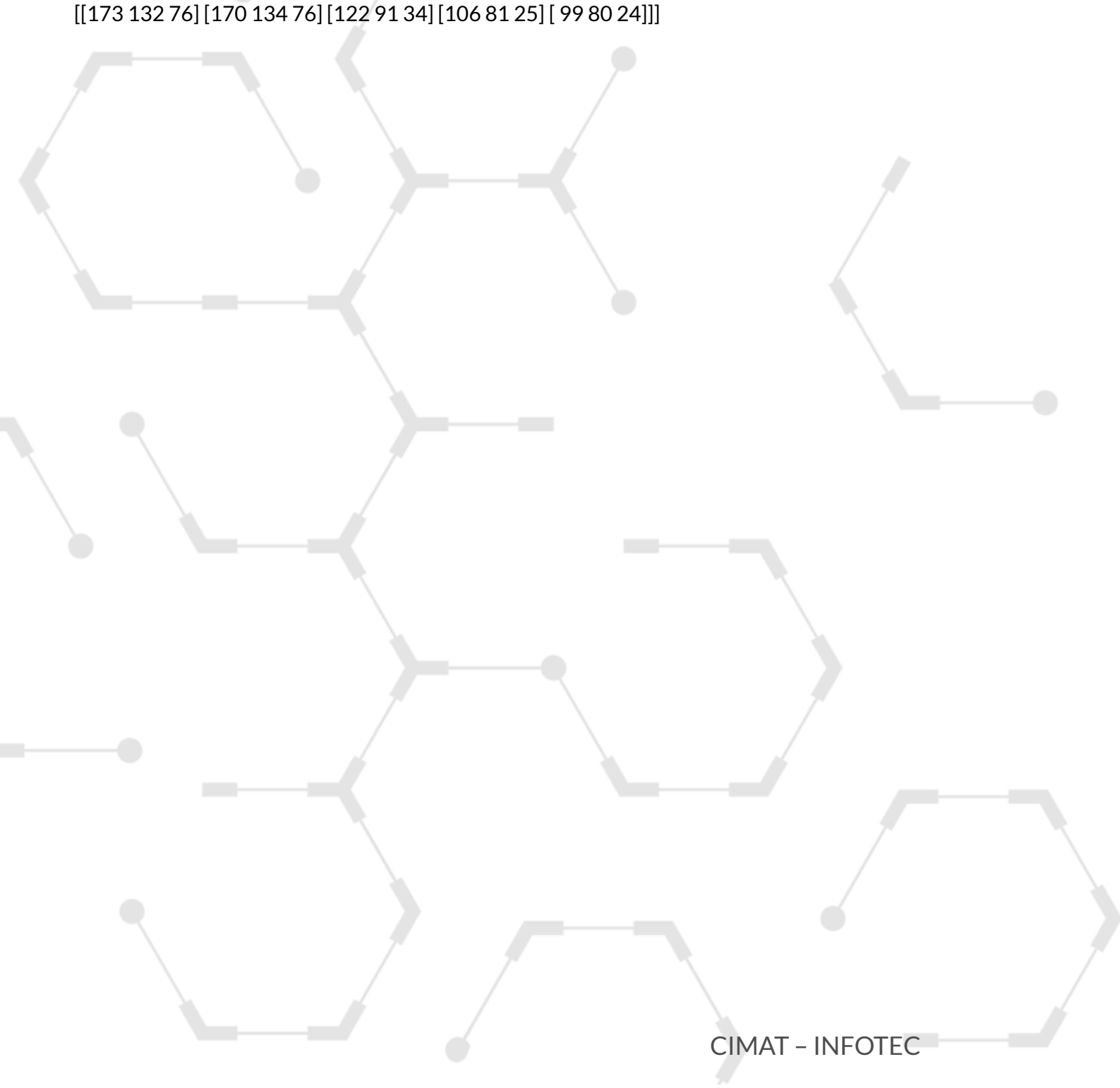
```
[[160 118 60][182 143 86][157 125 68][141 114 59][114 93 40]]
```

Patch 5x5 RGB

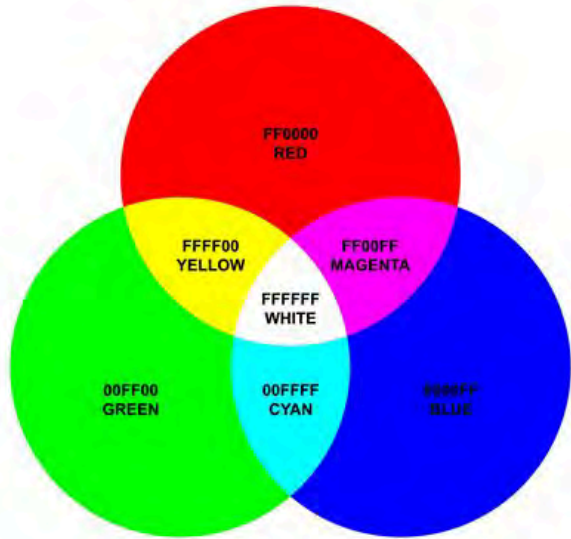




[[173 132 76][170 134 76][122 91 34][106 81 25][ 99 80 24]]



# Mezcla aditiva de colores – Modelo RGB



RGB color model venn diagram

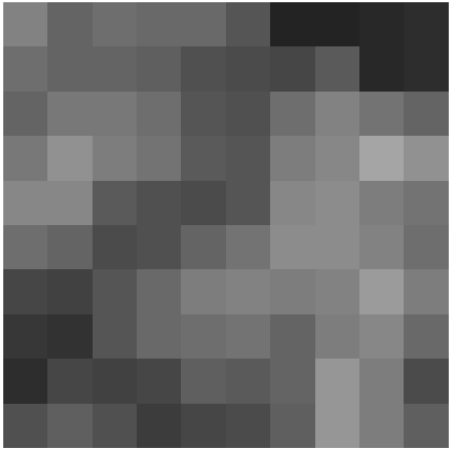
## Explicación:

- El modelo **RGB** (Red, Green, Blue) es **aditivo**.
- Combinaciones:
  - Rojo + Verde = Amarillo
  - Verde + Azul = Cian
  - Azul + Rojo = Magenta
  - Rojo + Verde + Azul (máxima intensidad) = Blanco
- Cada canal varía de **0 a 255**, permitiendo representar millones de colores.
- Es el modelo usado en **pantallas, cámaras y procesamiento digital de imágenes**.

# Visualización de valores en escala de grises

## ► Code

Sección 10x10 en escala de grises



```
[[132 101 111 109 106 88 39 37 45 48]
 [112 101 104 95 83 76 72 91 41 46]
 [104 123 123 114 90 80 110 131 116 104]
 [124 148 128 116 93 85 128 139 165 145]
 [138 138 94 82 79 89 135 140 129 118]
 [111 100 78 84 101 117 140 141 130 114]
 [ 72 67 88 109 127 134 126 134 158 128]
 [ 58 51 90 105 111 115 100 126 135 107]]
```

# Histograma de una imagen

- El histograma es la frecuencia de aparición de intensidades.
- Formalmente:

$$h(r_k) = n_k$$

donde  $r_k$  es un nivel de gris y  $n_k$  el número de píxeles con ese valor.

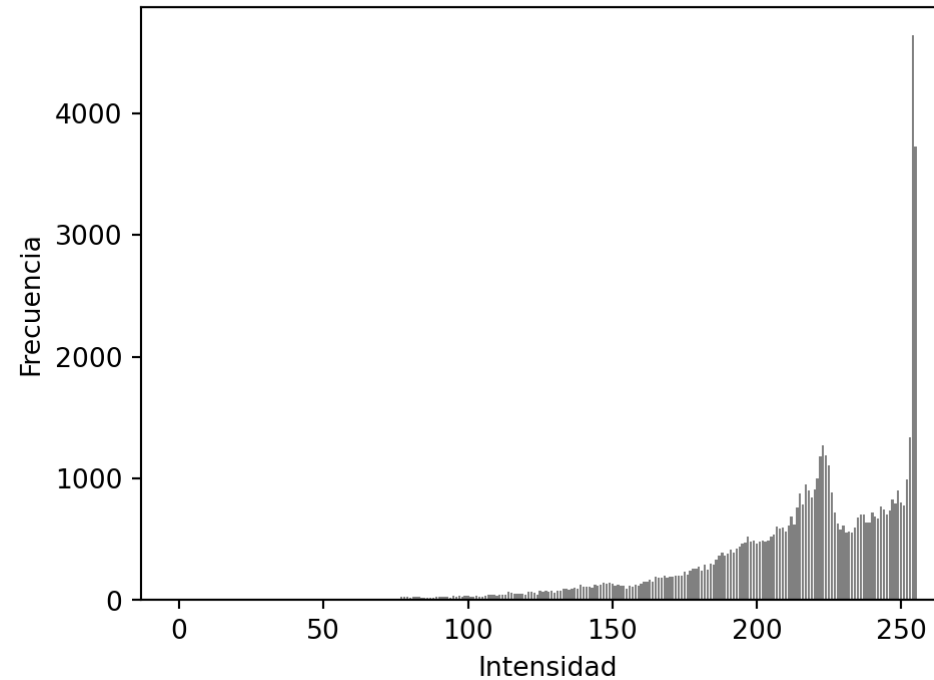
# Código: Construcción del histograma

## ► Code

Imagen 256x256



Histograma



# Ecualización de histograma

- Busca redistribuir los niveles de gris para mejorar el contraste.
- Función de transformación acumulativa:

$$s_k = (L - 1) \sum_{j=0}^k \frac{n_j}{MN}$$

donde:

- $L$  = número de niveles (256 en 8 bits).
- $M \times N$  = número total de píxeles.

# Ejemplo de ecualización de histograma

Consideremos una imagen de  $M \times N = 8$  píxeles, con intensidades de 3 bits ( $L = 8$  niveles, de 0 a 7):

$$I = [3, 3, 4, 5, 6, 6, 6, 7]$$

## 1. Histograma de frecuencias

Intensidad ( $r_k$ )	Frecuencia ( $n_k$ )
0	0
1	0
2	0
3	2
4	1
5	1
6	3
7	1

## 2. Probabilidades y acumuladas

Probabilidad:  $p(r_k) = n_k/8$

$r_k$	$p(r_k)$	CDF $\sum_{j=0}^k p(r_j)$
3	0.25	0.25
4	0.125	0.375
5	0.125	0.500
6	0.375	0.875
7	0.125	1.000



### 3. Nueva asignación de intensidades

$$s_k = (L - 1) \sum_{j=0}^k p(r_j)$$

Con  $L - 1 = 7$ :

$r_k$	CDF	$s_k$
3	0.25	$7 \cdot 0.25 = 1.75 \approx 2$
4	0.375	$7 \cdot 0.375 = 2.63 \approx 3$
5	0.50	$7 \cdot 0.50 = 3.50 \approx 4$
6	0.875	$7 \cdot 0.875 = 6.13 \approx 6$
7	1.00	$7 \cdot 1.00 = 7.00$

## 4. Imagen ecualizada

Imagen original:

$[3, 3, 4, 5, 6, 6, 6, 7]$

Imagen ecualizada:

$[2, 2, 3, 4, 6, 6, 6, 7]$

**Conclusión:** La ecualización redistribuyó los valores, extendiendo mejor el contraste entre 2 y 7.

# Código: Ecualización con OpenCV

## ► Code

# Convolución en imágenes

- La **convolución** es la operación matemática que permite aplicar un **kernel** o **máscara** sobre una imagen.
- Cada píxel resultante se obtiene como combinación ponderada de sus vecinos:

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b H(s, t) \cdot f(x + s, y + t)$$

# Filtrado espacial

- Consiste en aplicar una máscara o kernel  $H$  a la imagen:

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b H(s, t) \cdot f(x + s, y + t)$$

- Donde:
  - $f(x, y)$  = imagen original
  - $g(x, y)$  = imagen filtrada
  - $H$  = kernel de tamaño  $(2a + 1) \times (2b + 1)$

# Ejemplo de kernel promedio 3x3

$$H = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

# Filtro Promedio (Blur)

- El valor del píxel se reemplaza por el **promedio aritmético** de los píxeles en una ventana  $k \times k$ .

$$g(x, y) = \frac{1}{k^2} \sum_{s=-a}^a \sum_{t=-b}^b f(x + s, y + t)$$

- Kernel:

$$H = \frac{1}{k^2} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

# Ejemplo paso a paso

Imagen local (parche 3x3):

$$\begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \\ 70 & 80 & 90 \end{bmatrix}$$

Kernel de suavizado:

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Resultado para el píxel central =  $\frac{1}{9} (10 + 20 + \dots + 90) = 50$ .



# ¿Qué pasa en los bordes?

- Al aplicar un kernel, algunos vecinos quedan **fuera de la imagen**.
- Estrategias comunes de relleno (*padding*):
  - **Zero padding**: rellena con ceros.
  - **Replicate padding**: repite el valor de la orilla.
  - **Reflect padding**: refleja los valores como en un espejo.
  - **Valid convolution**: solo se calculan píxeles donde el kernel cabe completo.

# Ejemplo en OpenCV

## ► Code



# Filtro Gaussiano

# Filtro de Mediana

- Es un método **no lineal**.
- En una ventana  $k \times k$ , reemplaza el valor del píxel por la **mediana** de los valores en esa ventana.

$$g(x, y) = \text{mediana}\{f(x + s, y + t) \mid -a \leq s, t \leq a\}$$

- Preserva bordes mejor que los filtros lineales.
- Muy eficaz contra ruido impulsivo (“sal y pimienta”).

# Resumen Comparativo

Filtro	Fórmula	Naturaleza	Ventajas	Desventajas
<b>Promedio (Blur)</b>	Promedio aritmético	Lineal	Simple, rápido	Borra bordes y detalles
<b>Gaussiano</b>	Ponderación con distribución normal	Lineal	Suavizado natural, controlado por $\sigma$	Más costoso computacionalmente
<b>Mediana</b>	Mediana de intensidades	No lineal	Preserva bordes, elimina ruido impulsivo	Menos eficiente en ventanas grandes

# Código: Filtros en OpenCV

## ► Code

Original



Blur 3x3



Gaussian 11x11



Median 5



# Comparación visual y análisis

- El filtro **blur** suaviza pero genera pérdida de detalle.
- El filtro **gaussiano** suaviza de forma más natural.
- El filtro **mediana** elimina ruido tipo “sal y pimienta” y preserva bordes.

# Aprende

Consulta y toma notas de los siguientes materiales:

- Peguero Núñez, P. D. [Realce y restauración de imágenes](#)
- Giménez-Palomares, F. et al. (2016). [Convolución y filtrado de imágenes](#)
- León-Batallas, A. et al. (2020). [Revisión de métricas](#)
- Mazet, V. (2021). [Histogram transformations](#)
- Mazet, V. (2021). [Filtering](#)
- OpenCV. (s.f.). [Operaciones aritméticas](#)



# Conclusión

- El **histograma** describe la distribución de intensidades.
- La **ecualización** mejora el contraste redistribuyendo los niveles.
- Los **filtros espaciales** suavizan o reducen ruido:
  - Media → suaviza más, pero pierde bordes.
  - Gaussiano → transición suave, natural.
  - Mediana → preserva bordes, elimina ruido impulsivo.
- Comprender la **matemática de matrices y convolución** es clave para usar y diseñar filtros.