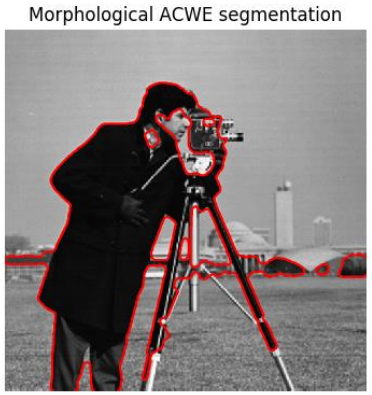
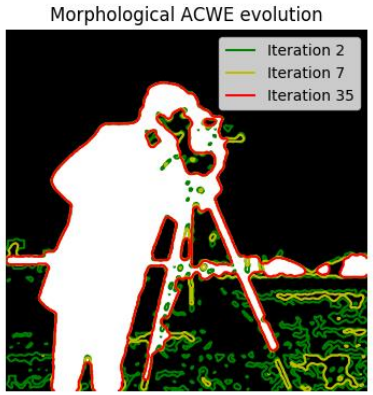
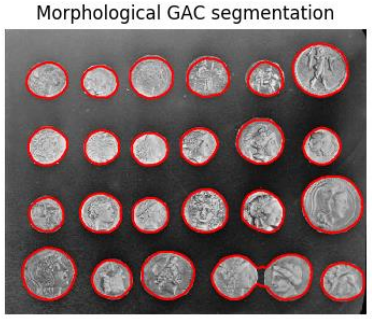
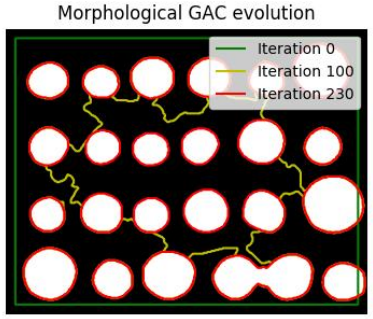


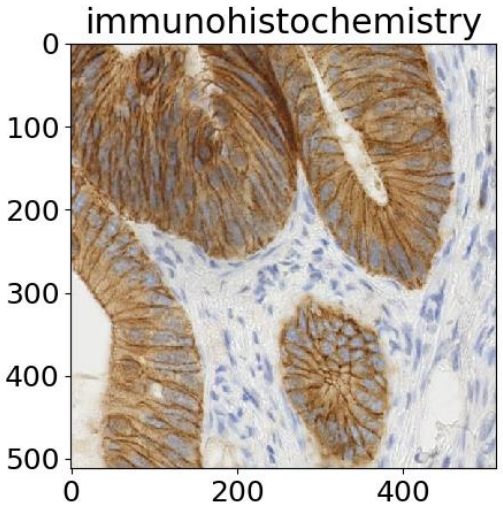


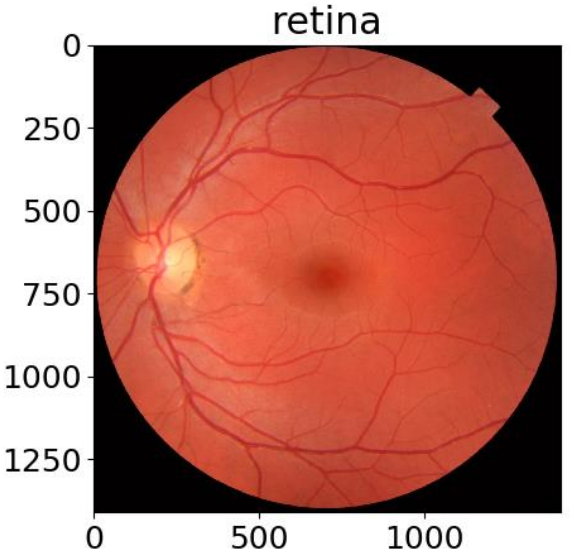




scikit-image

image processing in python





Fundamentos de procesamiento digital de imágenes con scikit-image

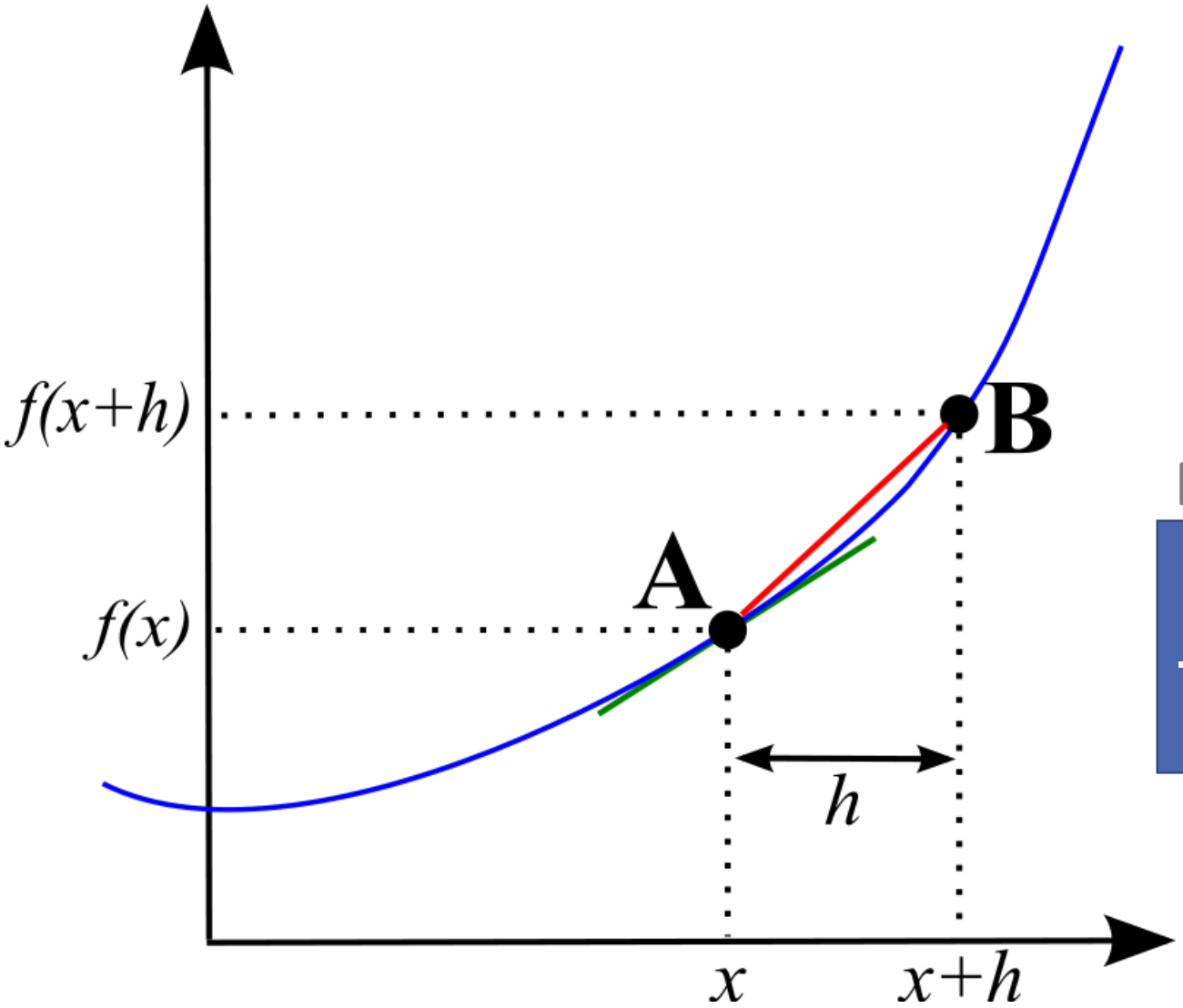
Rodolfo E. Escobar U.

Lo esencial: Matrices

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1j} \\ a_{21} & a_{22} & \cdots & a_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} \end{bmatrix}$$

En matemáticas, una matriz es un arreglo bidimensional de números. Las dimensiones de una matriz suelen denotarse por $m \times n$ dónde m es el número de filas y n el de columnas.

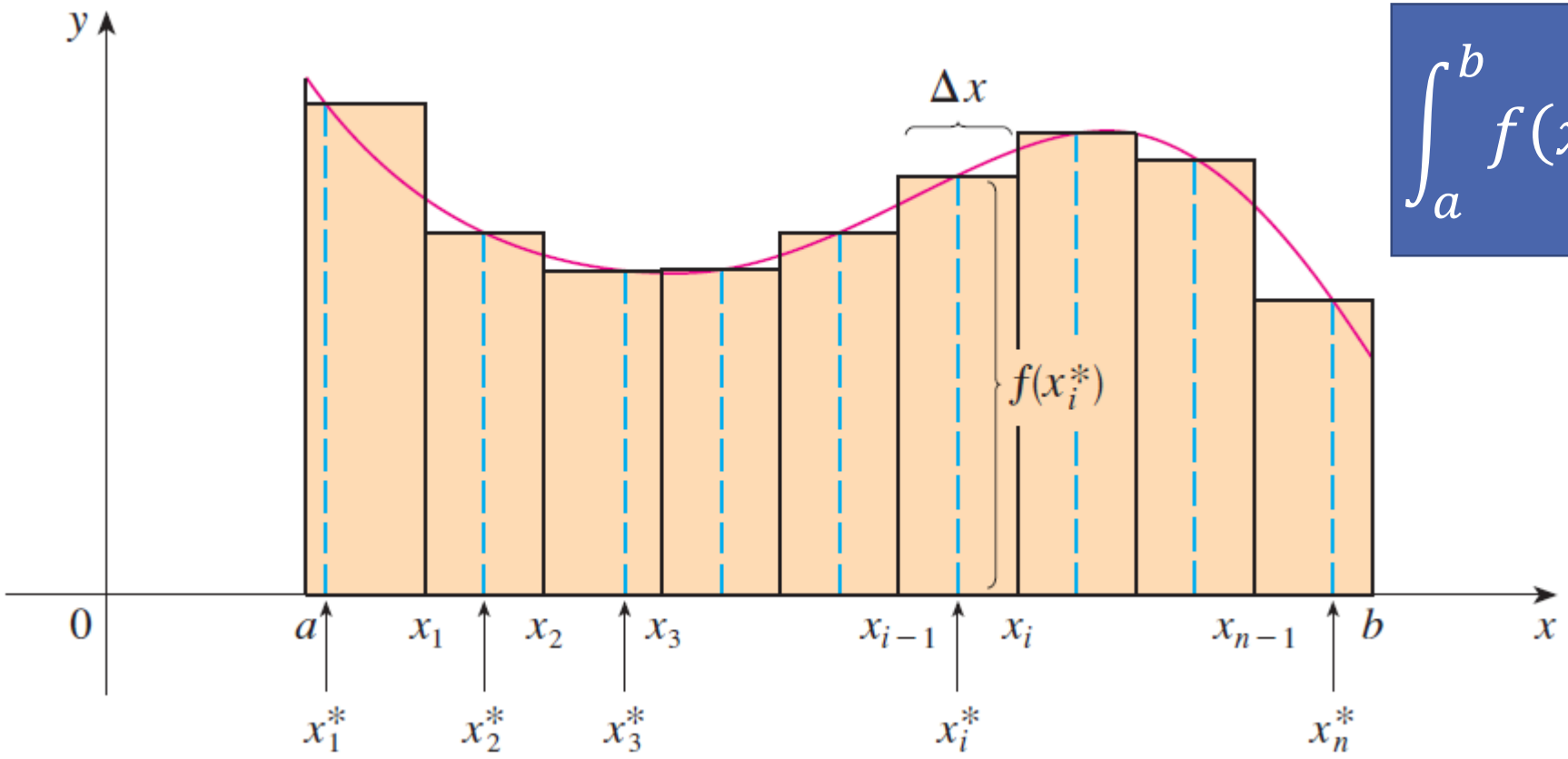
Lo esencial: Cálculo



Derivada

$$\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Lo esencial: Cálculo



Integral

$$\int_a^b f(x)dx = \lim_{n\to\infty} \sum_{i=1}^{n} f(x_i^*)\Delta x$$

Lo esencial: Tipos de dato

Tipo	Descripción	Rango
int8_t	Enteros signados de 8 bits	-128 a 127
uint8_t	Enteros sin signo de 8 bits	0 a 255
int16_t	Enteros signados de 16 bits	−32,768 a 32,767
uint16_t	Enteros sin signo de 16 bits	0 a 65,535
float	Racionales de 4 bytes	1.2×10^{-38} a 3.4×10^{38}
double	Racionales de 8 bytes	2.3×10^{-308} a 1.7×10^{308}

Lo esencial: Tipos de dato

uint8_t

Unsigned Integer

Vector de Ponderación $\boldsymbol{P}$

$2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$

00000101

Vector Binario $\boldsymbol{B}$

$$Valor\ decimal = \boldsymbol{P} \cdot \boldsymbol{B}$$

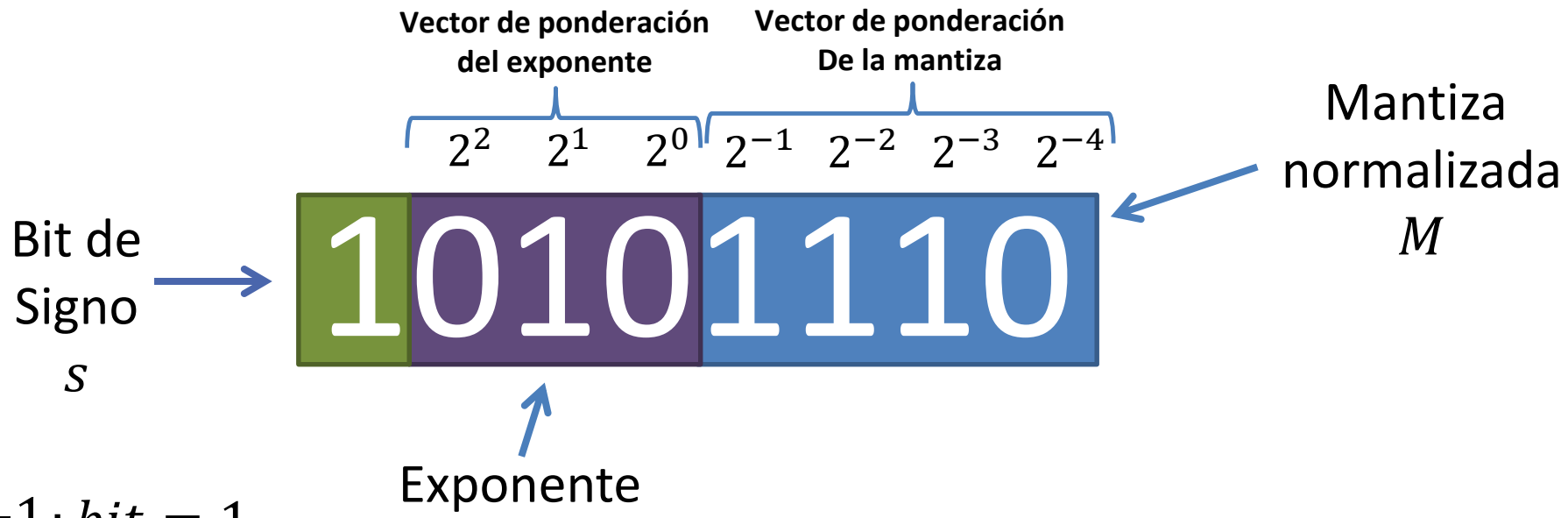
$$1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5$$

Rango: 0 a 255

Lo esencial: Tipos de dato

FLOAT8

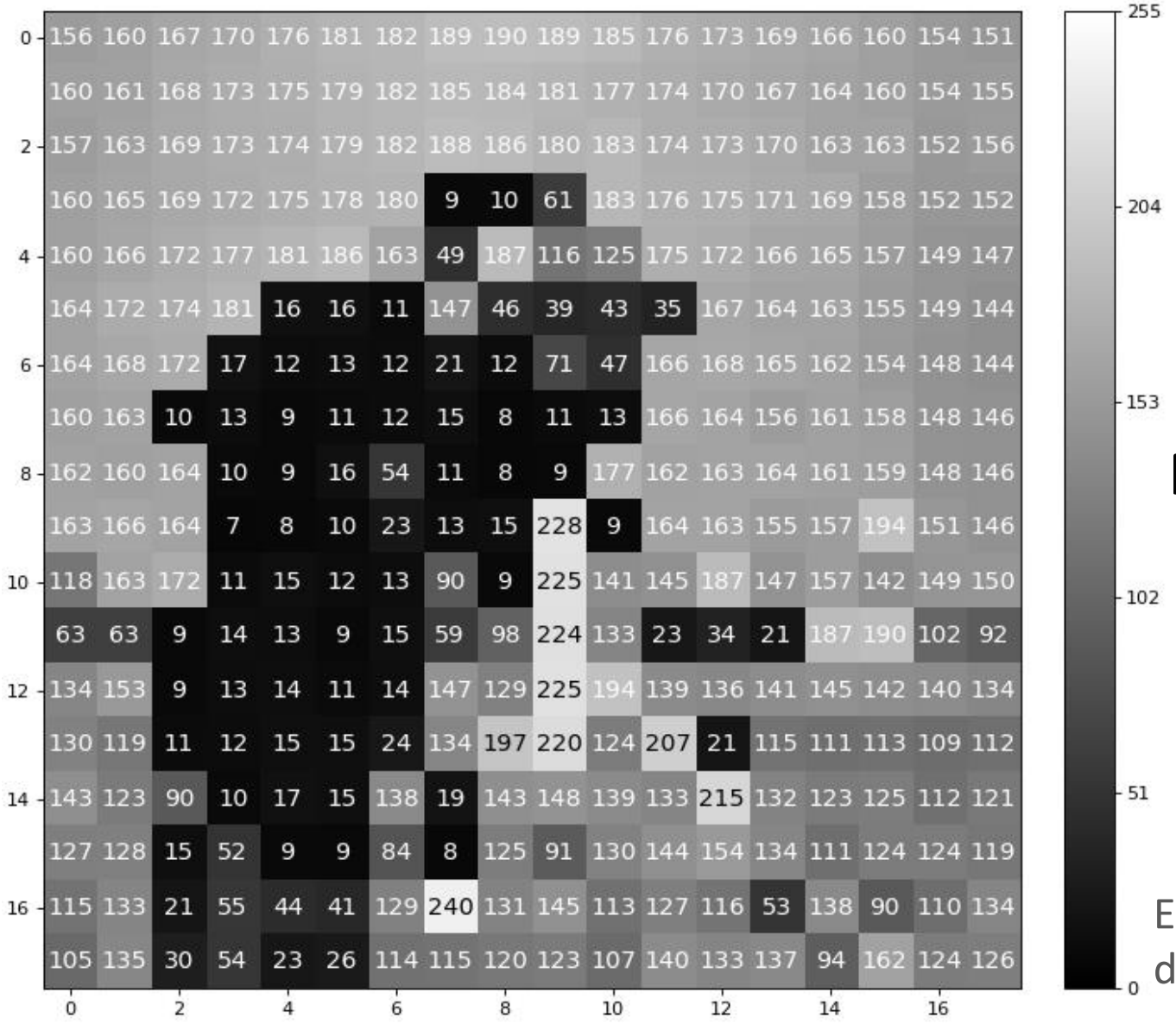
Floating Point



$$s\begin{cases} -1; bit = 1 \\ 1; bit = 0 \end{cases} \quad Valor\ decimal = s(M) \times 2^{exponente}$$

$$-1(1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3}) \times 2^{2} = -3.5$$

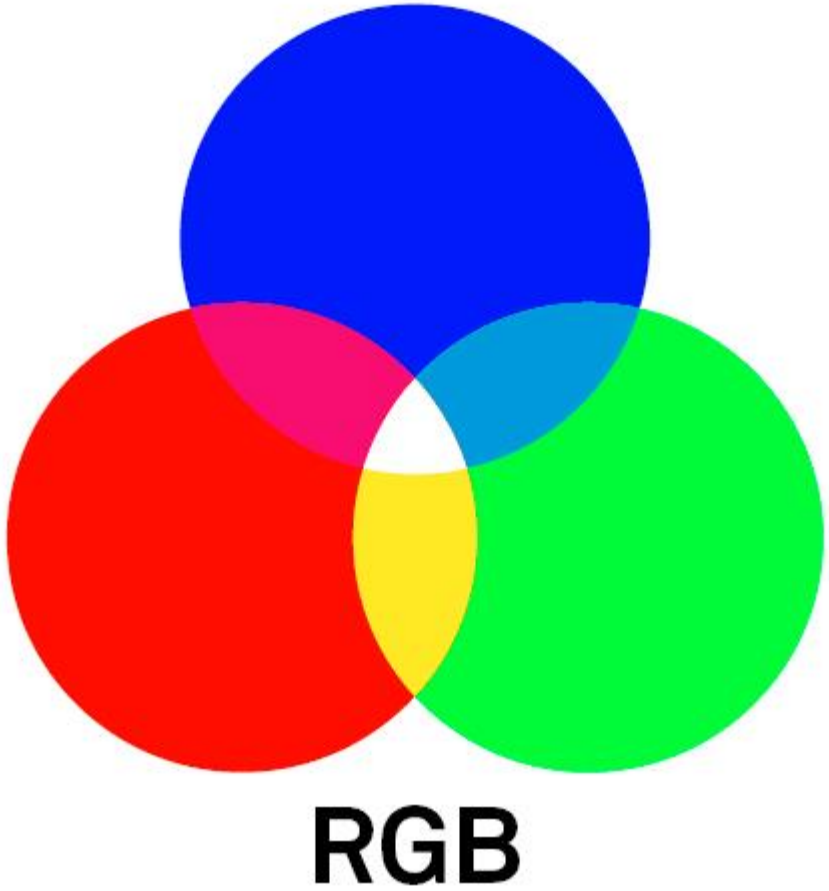
¿Qué es una imagen digital?



Repuesta corta. Es una matriz.

En este ejemplo, una matriz 18×18 de datos **uint8**.

Modelo aditivo de colores RGB



RGB es un **modelo de color** basado en la síntesis aditiva, con el que es posible representar un color mediante la mezcla por adición de los tres colores de luz primarios

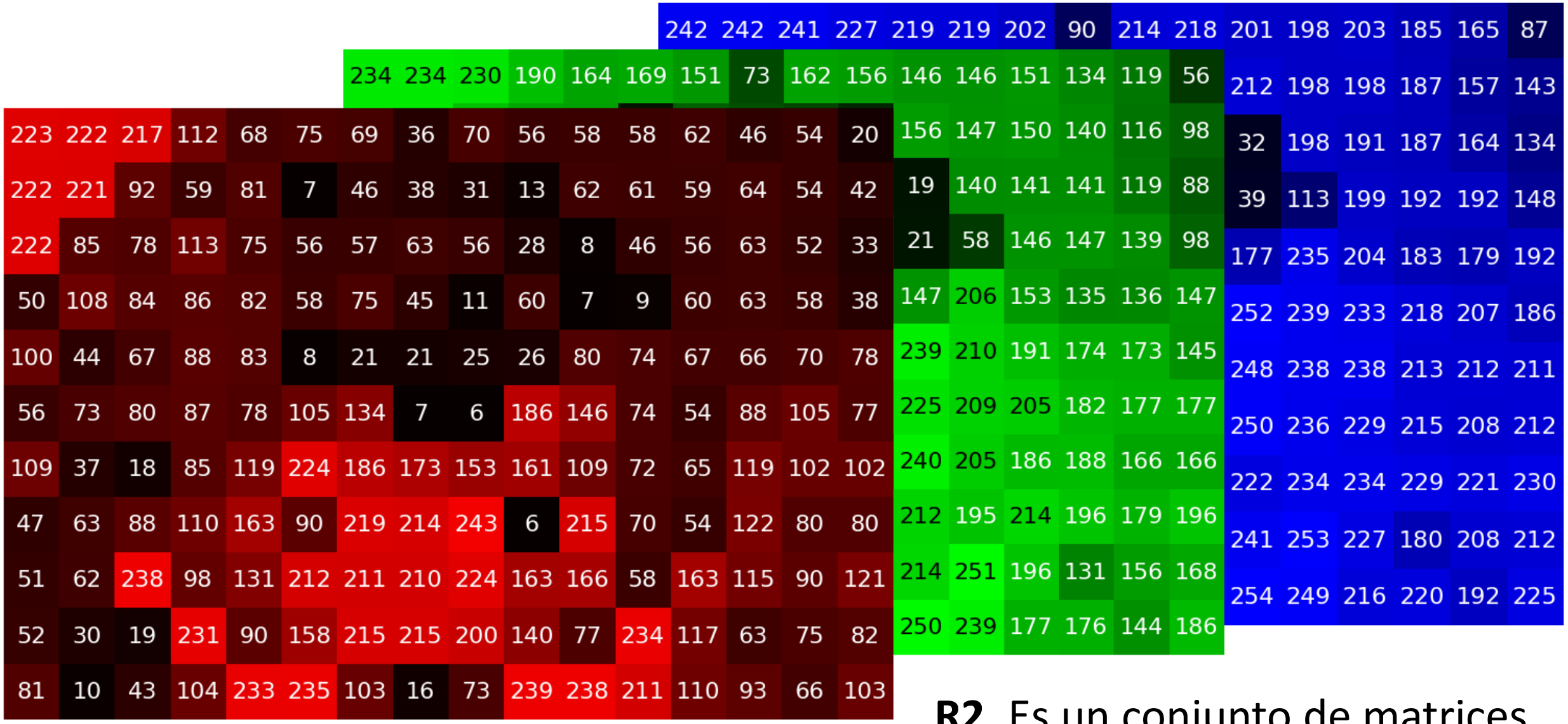
Modelo aditivo de colores RGB

Canal R

Canal G

Canal B

¿Qué es una imagen digital?



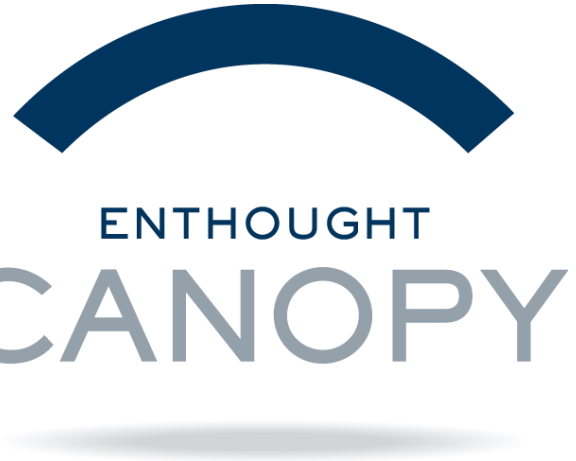
R2. Es un conjunto de matrices.

scikit-image
image processing in python

Scikit-image es una biblioteca de código abierto para **Python** que cuenta con algoritmos de segmentación, transformaciones geométricas, manipulaciones de espacio de color, análisis, filtrado, morfología, detección de características, etc. Su uso esta orientado al **procesado *offline* de imágenes**.

scikit-image.org/docs/dev/api/api.html

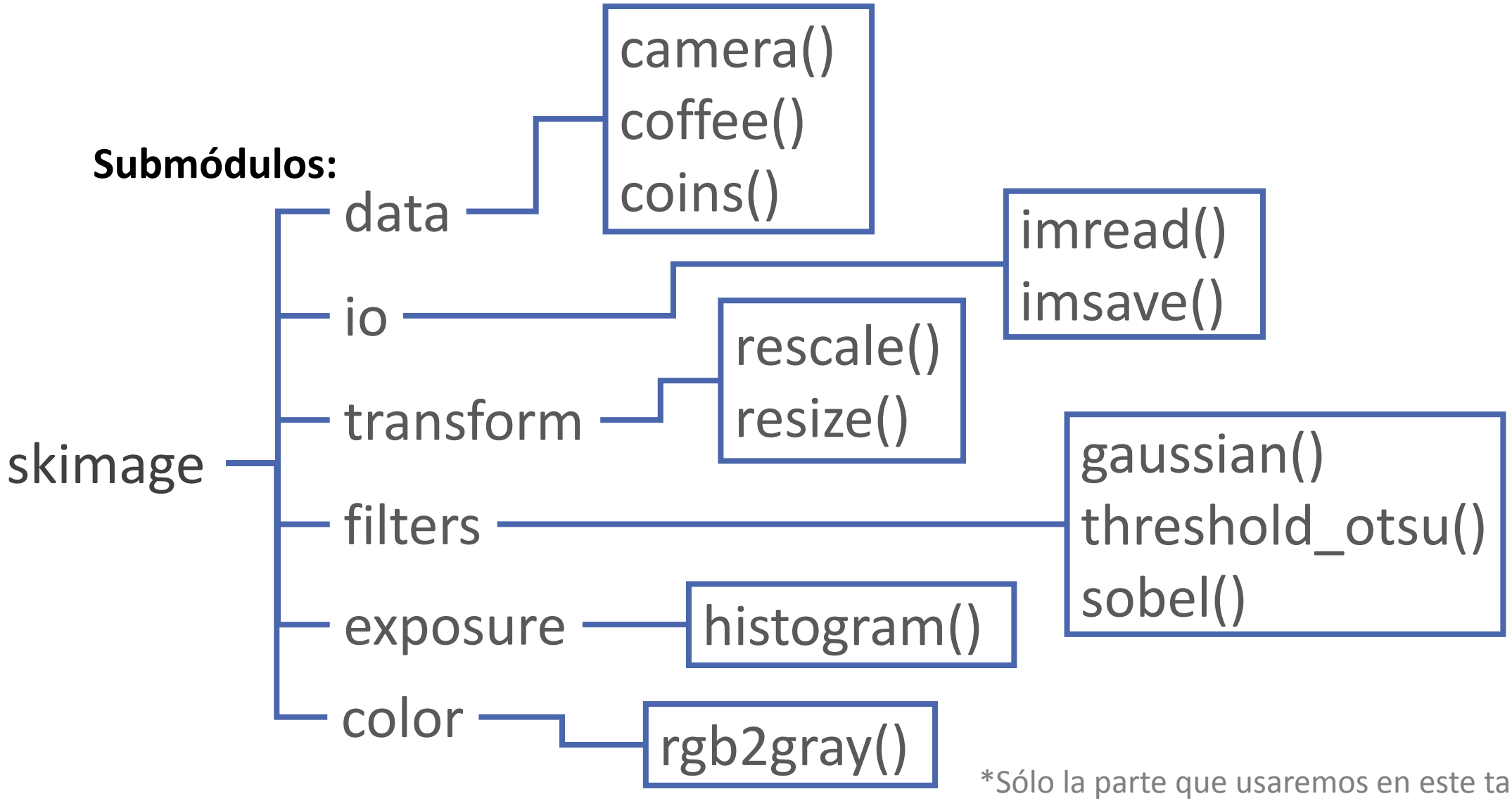
Viene pre-instalada junto con las dos principales distribuciones de Python:



O puede instalarse manualmente con:

```
pip install numpy scipy matplotlib scikit-image
```

Mapa* de scikit-image



*Sólo la parte que usaremos en este taller

Práctica #1

Cargar una imagen de la base de datos de ejemplo y visualizarla.

(El “Hola Mundo” del PDI)

Práctica #2

Extraer y visualizar de manera separada los canales de una imágen RGB .

Convolución

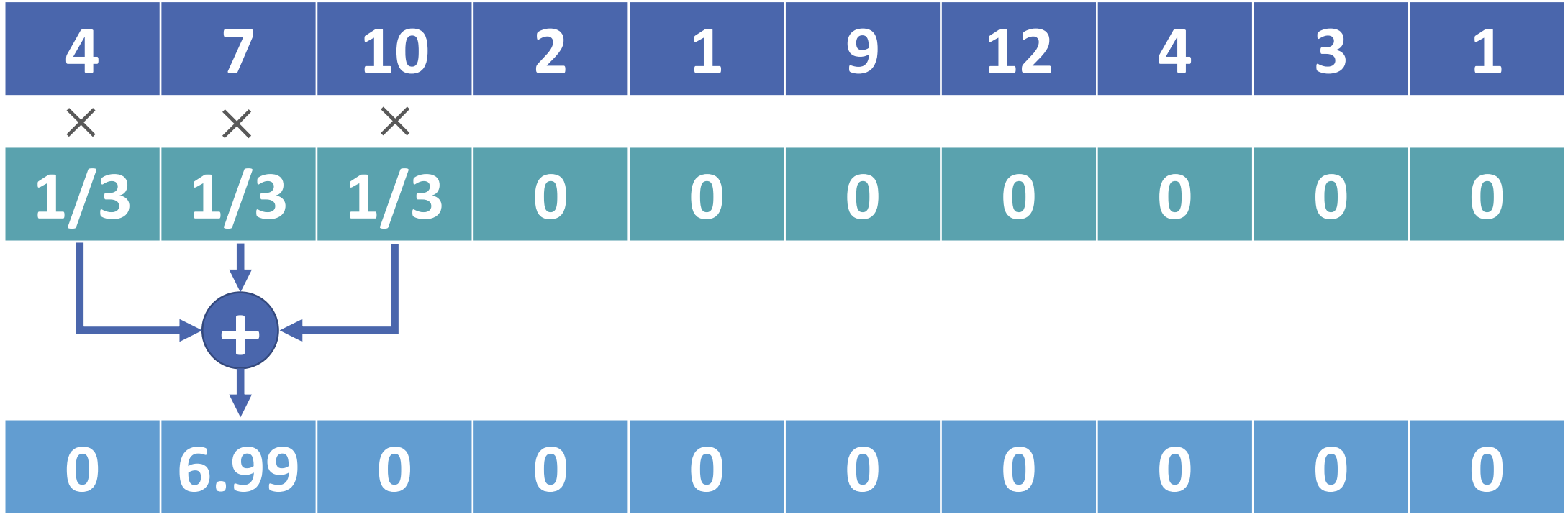
$$f(t) * g(t) = \int_0^t f(\tau) g(t-\tau) d\tau$$

https://commons.wikimedia.org/wiki/File:Convolution_of_box_signal_with_itself2.gif

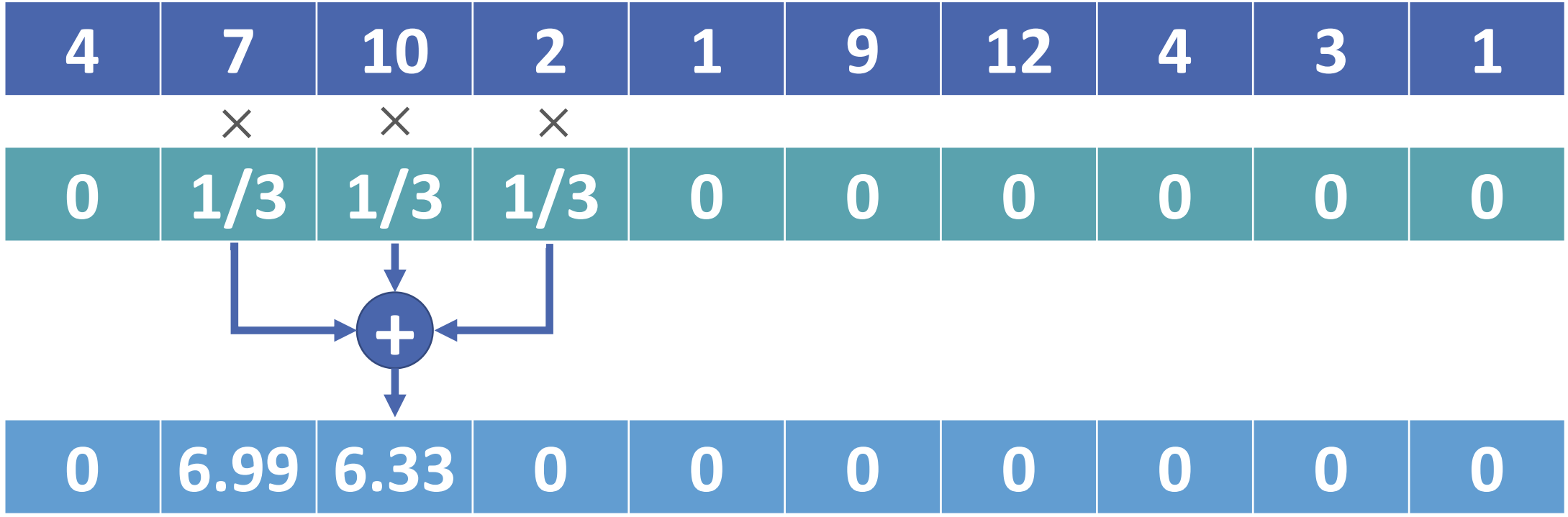
Promedio móvil

4	7	10	2	1	9	12	4	3	1

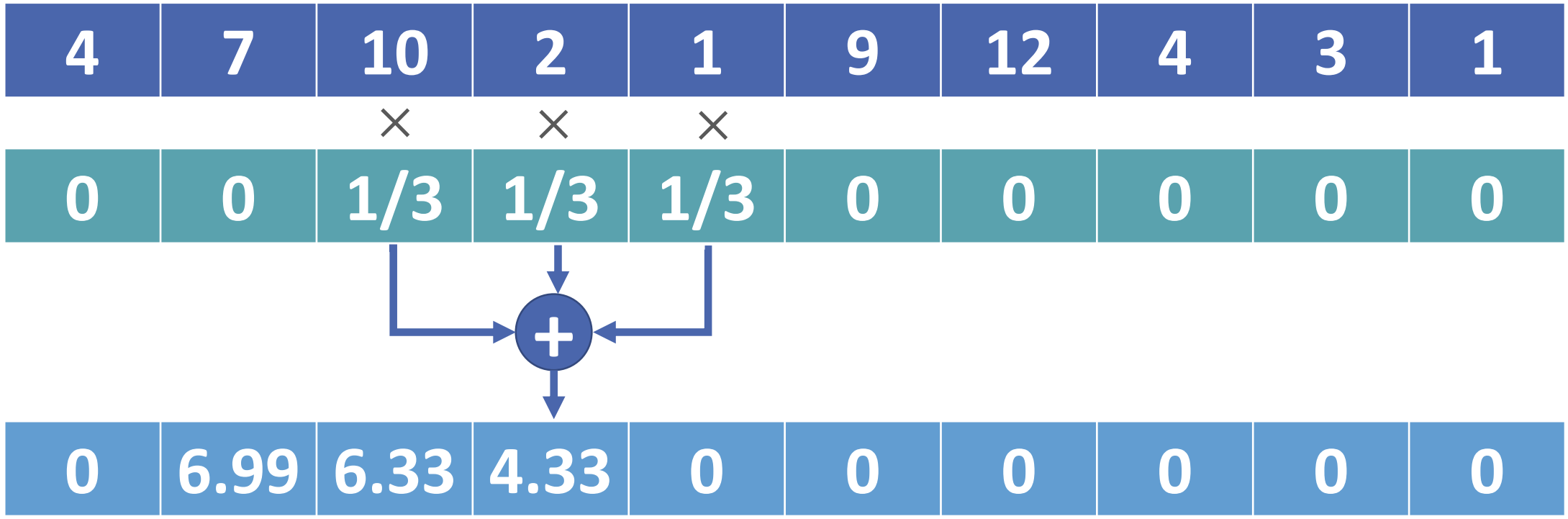
Promedio móvil



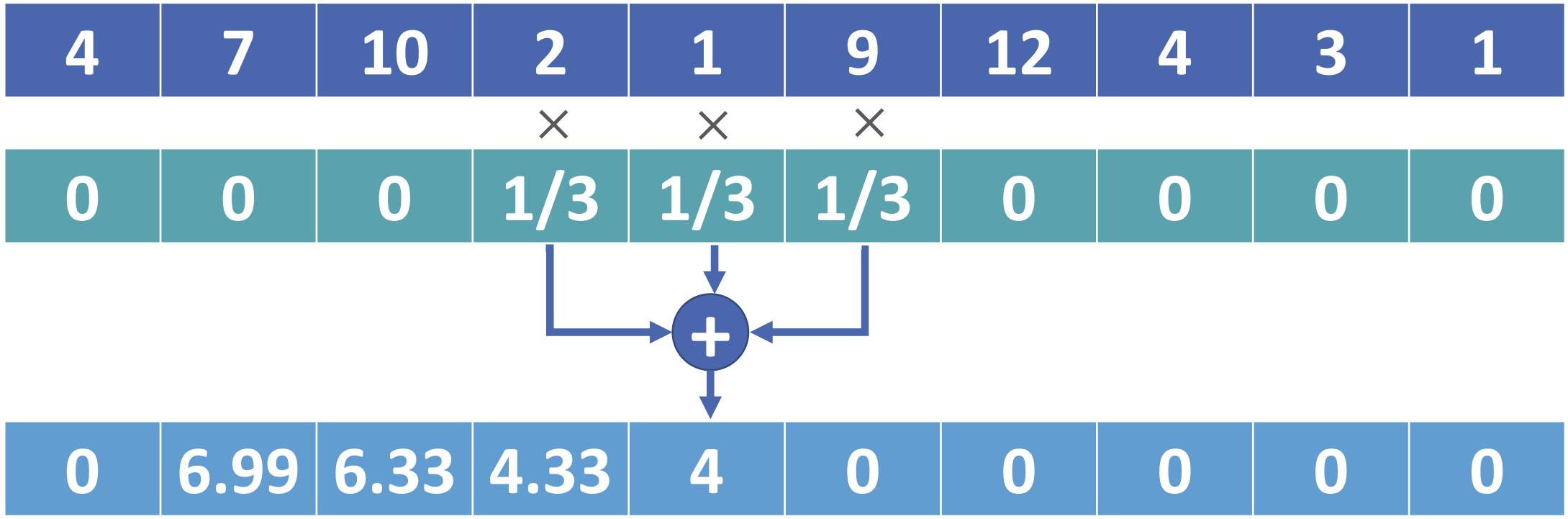
Promedio móvil



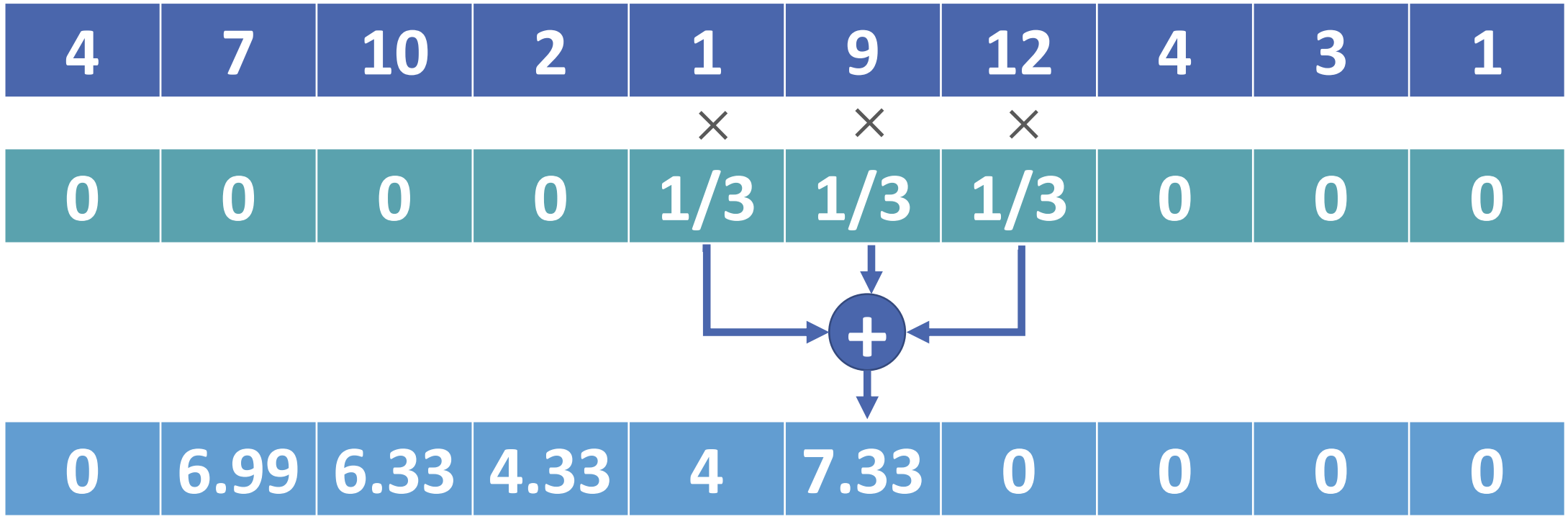
Promedio móvil



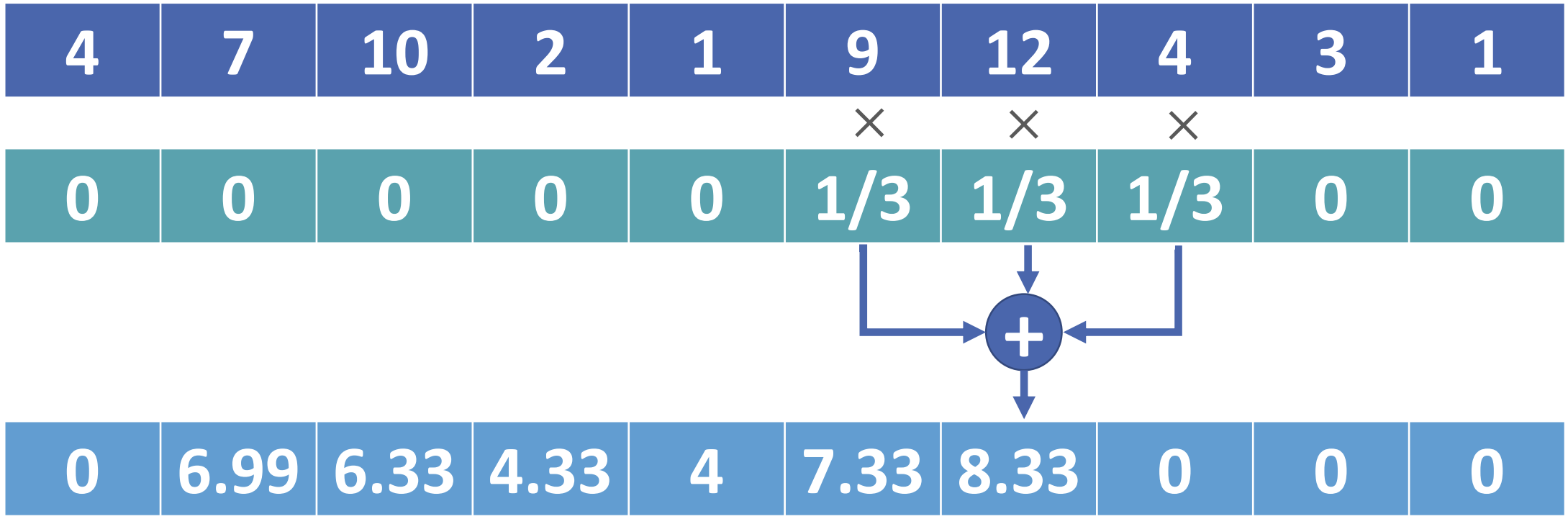
Promedio móvil



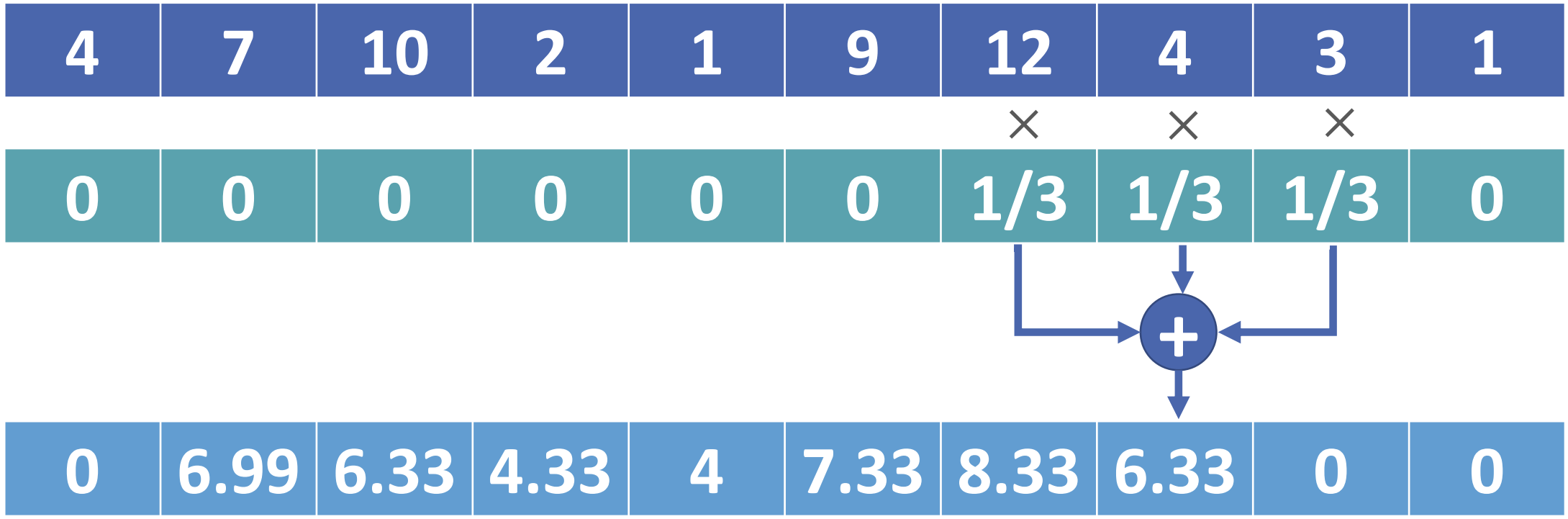
Promedio móvil



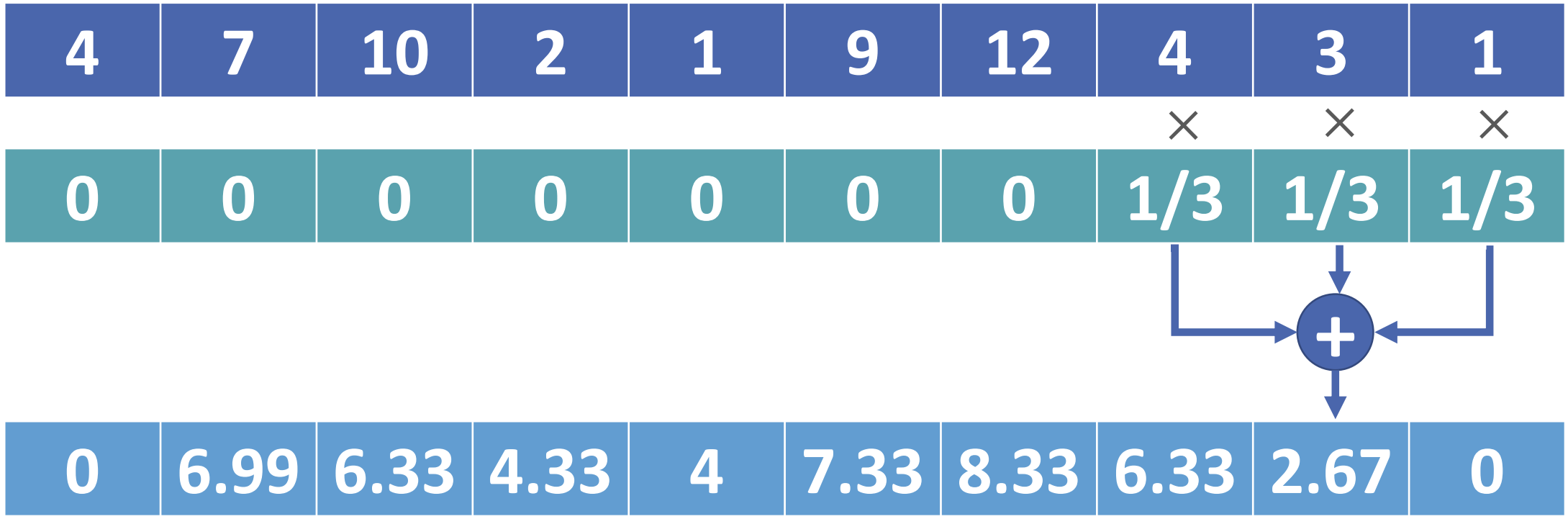
Promedio móvil



Promedio móvil



Promedio móvil



Promedio móvil con numpy

In [15]:

```
import numpy as np
from scipy import signal

V = [4,7,10,2,1,9,12,4,3,1]
V = np.array(V)
H = np.ones(3)*1/3.0
Vm = signal.convolve(V,H,mode='valid')

np.set_printoptions(precision=2)
print("V = " + str(V))
print("Vm = " + str(Vm))
```

```
V = [ 4  7 10  2  1  9 12  4  3  1]
Vm = [7.   6.33 4.33 4.   7.33 8.33 6.33 2.67]
```

Promedio móvil en 2D

3	1	9	1	0
1	7	6	2	3
2	1	0	8	1
0	4	2	4	0
7	9	5	1	2

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Promedio móvil en 2D

$3\cdot\frac{1}{9}$	$1\cdot\frac{1}{9}$	$9\cdot\frac{1}{9}$	1	0
$1\cdot\frac{1}{9}$	$7\cdot\frac{1}{9}$	$6\cdot\frac{1}{9}$	2	3
$2\cdot\frac{1}{9}$	$1\cdot\frac{1}{9}$	$0\cdot\frac{1}{9}$	8	1
0	4	2	4	0
7	9	5	1	2

0	0	0	0	0
0	3.3	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Promedio móvil en 2D

3	$1\cdot\frac{1}{9}$	$9\cdot\frac{1}{9}$	$1\cdot\frac{1}{9}$	0
1	$7\cdot\frac{1}{9}$	$6\cdot\frac{1}{9}$	$2\cdot\frac{1}{9}$	3
2	$1\cdot\frac{1}{9}$	$0\cdot\frac{1}{9}$	$8\cdot\frac{1}{9}$	1
0	4	2	4	0
7	9	5	1	2

0	0	0	0	0
0	3.3	3.9	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Promedio móvil en 2D

3	1	$9\cdot\frac{1}{9}$	$1\cdot\frac{1}{9}$	$0\cdot\frac{1}{9}$
1	7	$6\cdot\frac{1}{9}$	$2\cdot\frac{1}{9}$	$3\cdot\frac{1}{9}$
2	1	$0\cdot\frac{1}{9}$	$8\cdot\frac{1}{9}$	$1\cdot\frac{1}{9}$
0	4	2	4	0
7	9	5	1	2

0	0	0	0	0
0	3.3	3.9	3.3	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Promedio móvil en 2D

3	1	9	1	0
$1\cdot\frac{1}{9}$	$7\cdot\frac{1}{9}$	$6\cdot\frac{1}{9}$	2	3
$2\cdot\frac{1}{9}$	$1\cdot\frac{1}{9}$	$0\cdot\frac{1}{9}$	8	1
$0\cdot\frac{1}{9}$	$4\cdot\frac{1}{9}$	$2\cdot\frac{1}{9}$	4	0
7	9	5	1	2

0	0	0	0	0
0	3.3	3.9	3.3	0
0	2.5	0	0	0
0	0	0	0	0
0	0	0	0	0

Promedio móvil en 2D

3	1	9	1	0
1	$7\cdot\frac{1}{9}$	$6\cdot\frac{1}{9}$	$2\cdot\frac{1}{9}$	3
2	$1\cdot\frac{1}{9}$	$0\cdot\frac{1}{9}$	$8\cdot\frac{1}{9}$	1
0	$4\cdot\frac{1}{9}$	$2\cdot\frac{1}{9}$	$4\cdot\frac{1}{9}$	0
7	9	5	1	2

0	0	0	0	0
0	3.3	3.9	3.3	0
0	2.5	7.8	0	0
0	0	0	0	0
0	0	0	0	0

Promedio móvil en 2D

3	1	9	1	0
1	7	$6\cdot\frac{1}{9}$	$2\cdot\frac{1}{9}$	$3\cdot\frac{1}{9}$
2	1	$0\cdot\frac{1}{9}$	$8\cdot\frac{1}{9}$	$1\cdot\frac{1}{9}$
0	4	$2\cdot\frac{1}{9}$	$4\cdot\frac{1}{9}$	$0\cdot\frac{1}{9}$
7	9	5	1	2

0	0	0	0	0
0	3.3	3.9	3.3	0
0	2.5	7.8	2.9	0
0	0	0	0	0
0	0	0	0	0

Promedio móvil en 2D

3	1	9	1	0
1	7	6	2	3
$2\cdot\frac{1}{9}$	$1\cdot\frac{1}{9}$	$0\cdot\frac{1}{9}$	8	1
$0\cdot\frac{1}{9}$	$4\cdot\frac{1}{9}$	$2\cdot\frac{1}{9}$	4	0
$7\cdot\frac{1}{9}$	$9\cdot\frac{1}{9}$	$5\cdot\frac{1}{9}$	1	2

0	0	0	0	0
0	3.3	3.9	3.3	0
0	2.5	7.8	2.9	0
0	3.3	0	0	0
0	0	0	0	0

Promedio móvil en 2D

3	1	9	1	0
1	7	6	2	3
2	$1\cdot\frac{1}{9}$	$0\cdot\frac{1}{9}$	$8\cdot\frac{1}{9}$	1
0	$4\cdot\frac{1}{9}$	$2\cdot\frac{1}{9}$	$4\cdot\frac{1}{9}$	0
7	$9\cdot\frac{1}{9}$	$5\cdot\frac{1}{9}$	$1\cdot\frac{1}{9}$	2

0	0	0	0	0
0	3.3	3.9	3.3	0
0	2.5	7.8	2.9	0
0	3.3	3.8	0	0
0	0	0	0	0

Promedio móvil en 2D

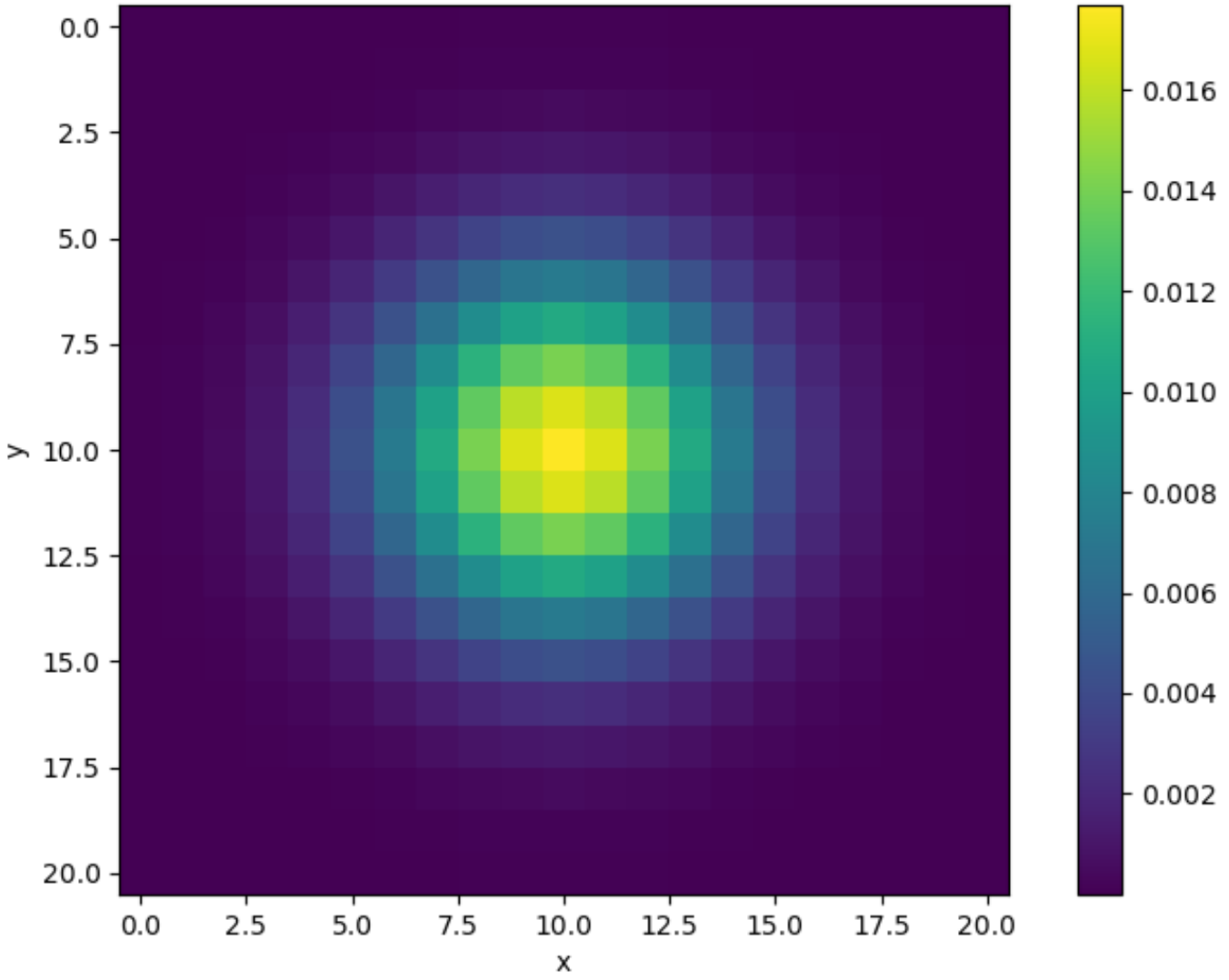
3	1	9	1	0
1	7	6	2	3
2	1	$0\cdot\frac{1}{9}$	$8\cdot\frac{1}{9}$	$1\cdot\frac{1}{9}$
0	4	$2\cdot\frac{1}{9}$	$4\cdot\frac{1}{9}$	$0\cdot\frac{1}{9}$
7	9	$5\cdot\frac{1}{9}$	$1\cdot\frac{1}{9}$	$2\cdot\frac{1}{9}$

0	0	0	0	0
0	3.3	3.9	3.3	0
0	2.5	7.8	2.9	0
0	3.3	3.8	2.5	0
0	0	0	0	0

Práctica #3

Realizar un filtrado de promedio con un Kernel de 11x11 a *Camera Man*.

Kernel Gaussiano

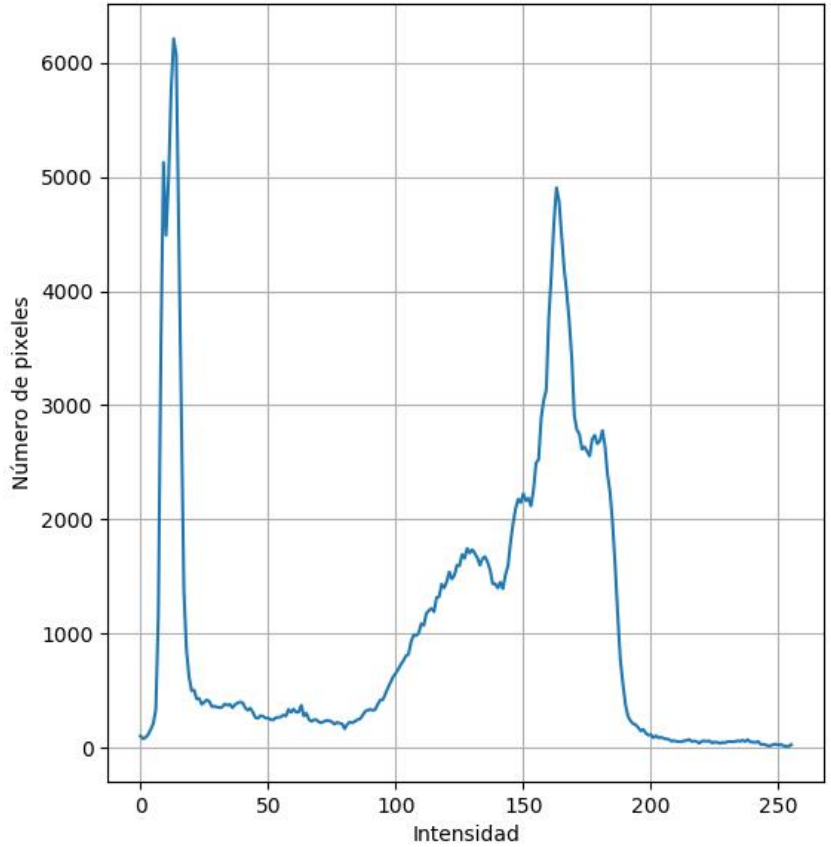


$$H(x,y) = Ae^{-\left(\frac{(x-x_0)^2}{2\sigma_X^2}+\frac{(y-y_0)^2}{2\sigma_Y^2}\right)}$$

Práctica #4

Realizar un filtrado gaussiano con $\sigma = 10$ a *Camera Man*.

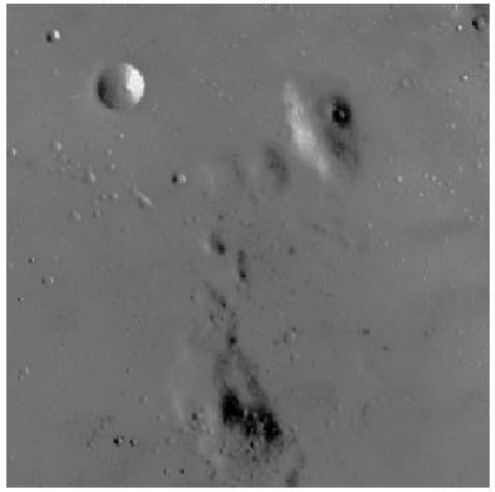
Histograma



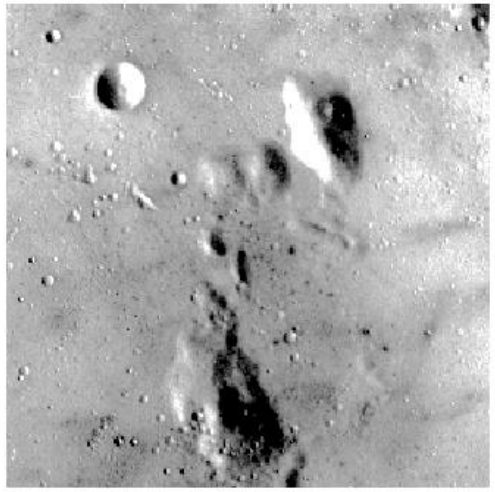
Un histograma es una representación grafica del número de pixeles que hay de un valor determinado dentro de la imagen .

Histograma

Imagen de bajo contraste

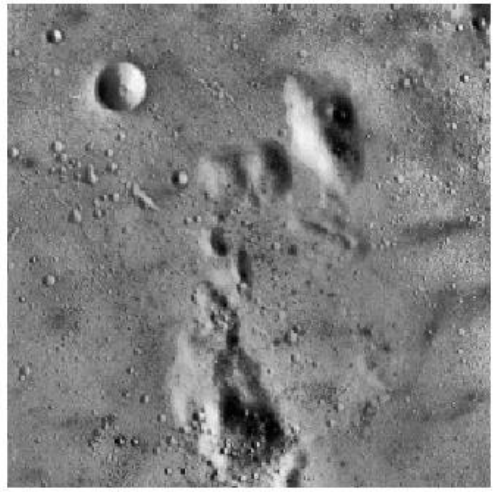


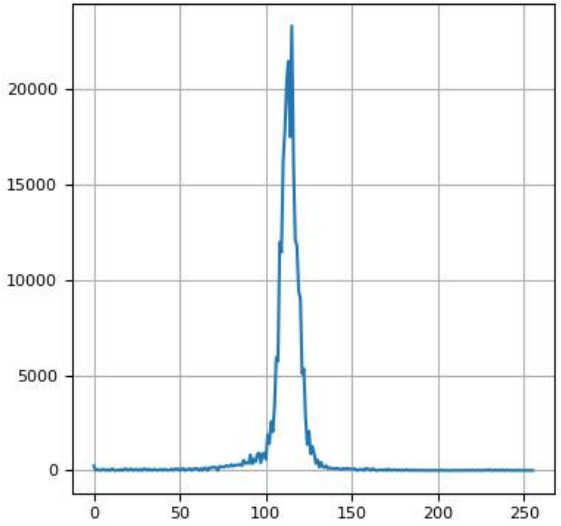
Contraste aumentado

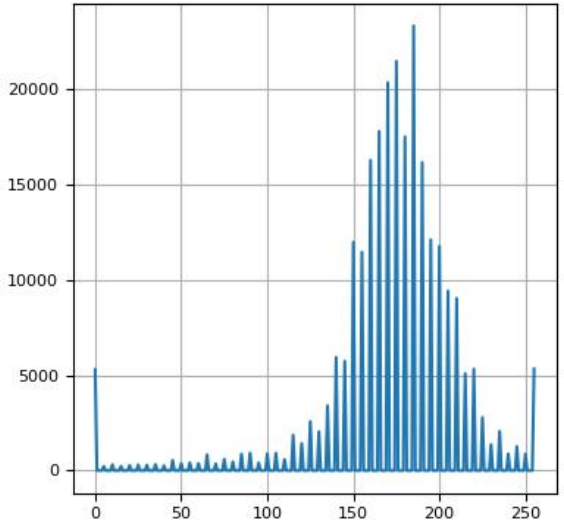


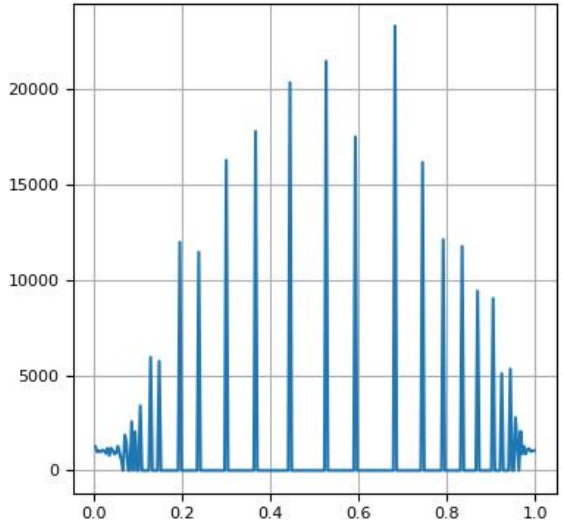
Histograma equalizado

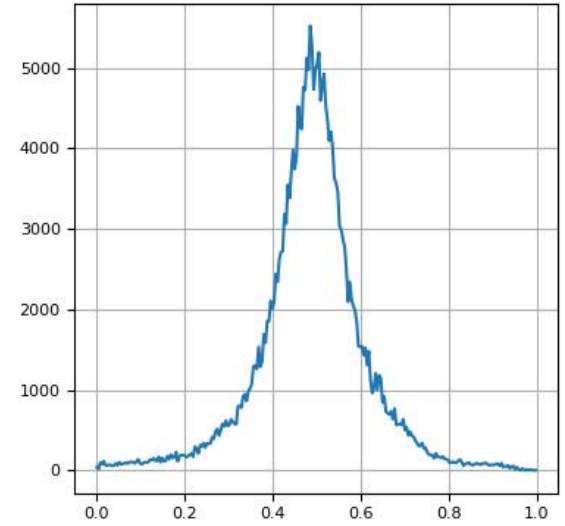
Equalización adaptativa







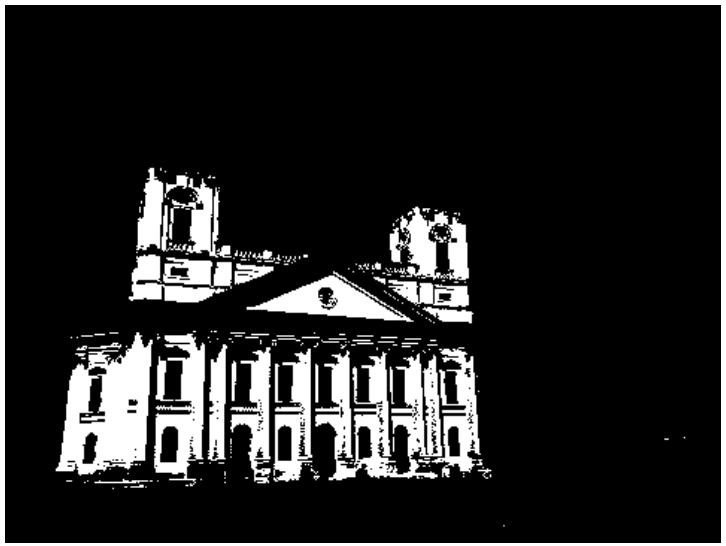




Mejora de contraste

skimage.exposure.rescale_intensity(image, in_range, out_range)

Umbrales y binarización



Se llama ***binarización*** al resultado de la operación lógica de relación ($<$,$>$,$\leq$,$\geq$,$=$) entre la imagen digital y un valor de umbral.

Ejemplo. Si el umbral es 121, la operación lógica $A(x, y) < 121$ dará verdadero sólo si el pixel en (x, y) tiene un valor menor a 121.

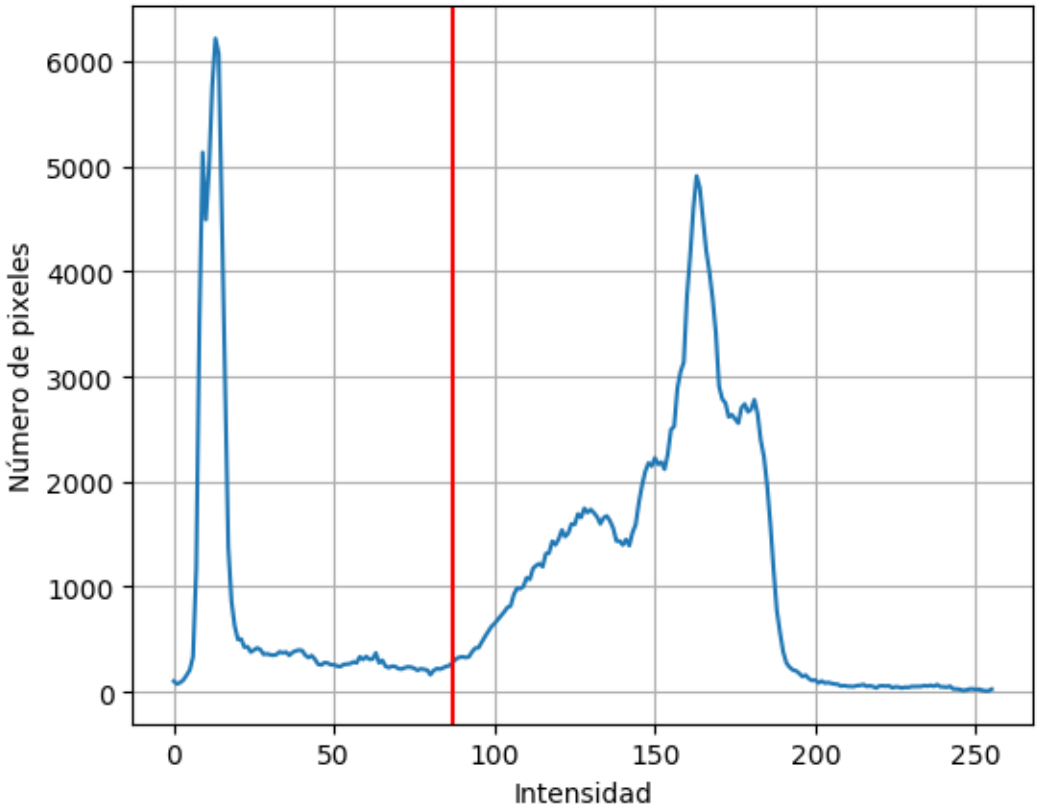
Umbrales y binarización

20	15	14	12
3	14	122	100
5	32	97	108
21	10	8	16

$P(x, y) > 90$

F	F	F	F
F	F	T	T
F	F	T	T
F	F	F	F

Método de Otsu

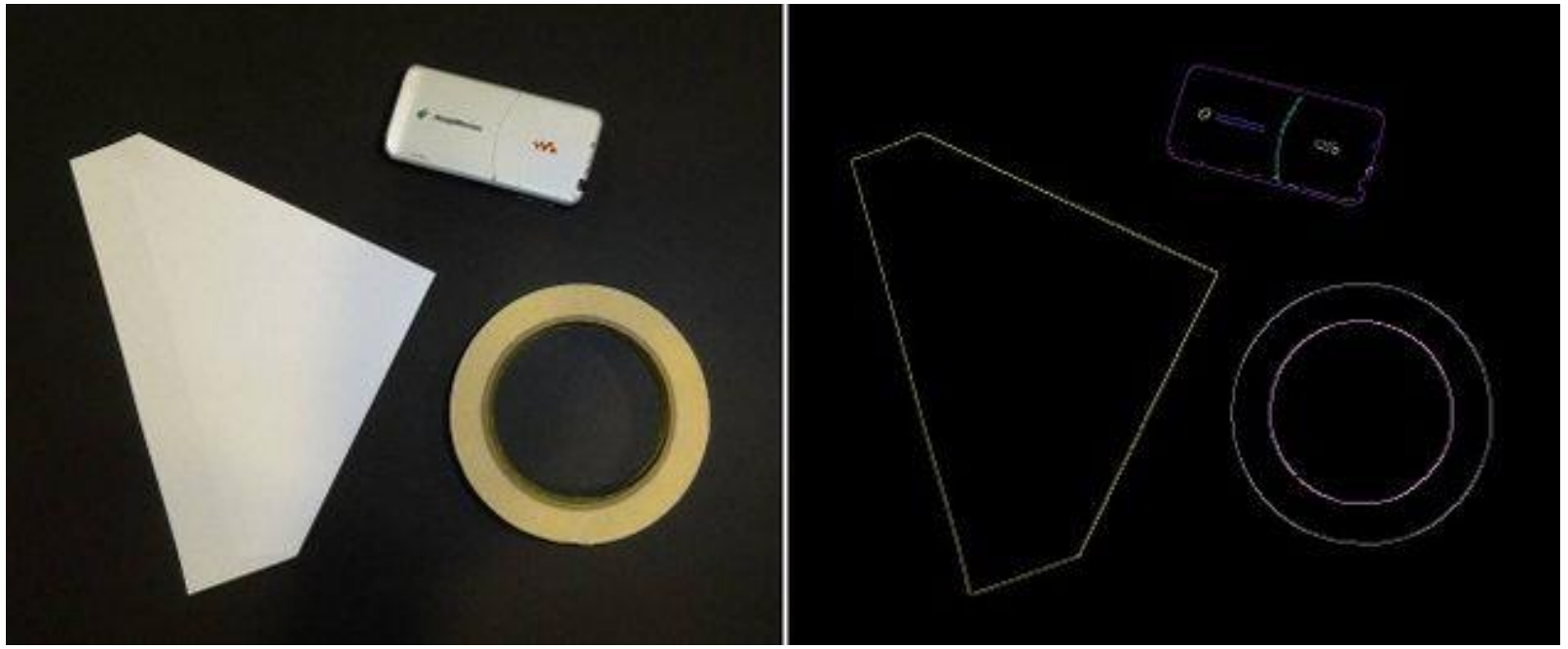


El método de Otsu es un algoritmo que calcula el valor de umbral como un punto que divide al histograma en dos clases: objeto y fondo.

Práctica #5

Calcular el valor de umbral con el método de Otsu y obtener una imagen binarizada de *Chelsea The Cat*. Graficar histograma.

Detección de bordes



La detección de bordes es una herramienta fundamental que tiene como objetivo la identificación de puntos en una imagen digital en la que el brillo de la imagen cambia ***abruptamente***.

Detección de bordes: Operador de Sobel

Sea A una imagen (de un solo canal)

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A \qquad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * A$$

$$G = \sqrt{G_x^2 + G_y^2}$$

Imagen de magnitud del gradiente de cambios en la imagen A.

Detección de bordes: Operador de Sobel

0	0	1	1	1	1
0	0	1	1	1	1
0	0	1	1	1	1
0	0	1	1	1	1

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Detección de bordes: Operador de Sobel

0	0	1	1	1	1
0	0	2	1	1	1
0	0	1	1	1	1
0	0	1	1	1	1

Bordes verticales

0	0	0	0	0	0
0	4	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Detección de bordes: Operador de Sobel

0	0	0	1	1	1
0	0	0	2	1	1
0	0	0	1	1	1
0	0	1	1	1	1

Bordes verticales

0	0	0	0	0	0
0	4	4	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Detección de bordes: Operador de Sobel

0	0	-1	0	1	1
0	0	-2	0	2	1
0	0	-1	0	1	1
0	0	1	1	1	1

Bordes verticales

0	0	0	0	0	0
0	4	4	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Detección de bordes: Operador de Sobel

0	0	1	-1	0	1
0	0	1	-2	0	2
0	0	1	-1	0	1
0	0	1	1	1	1

Bordes verticales

0	0	0	0	0	0
0	4	4	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Detección de bordes: Operador de Sobel

0	0	1	1	1	1
0	0	1	1	1	1
0	0	2	1	1	1
0	0	1	1	1	1

Bordes verticales

0	0	0	0	0	0
0	4	4	0	0	0
0	4	0	0	0	0
0	0	0	0	0	0

Detección de bordes: Operador de Sobel

0	0	1	1	1	1
0	0	0	1	1	1
0	0	0	2	1	1
0	0	0	1	1	1

Bordes verticales

0	0	0	0	0	0
0	4	4	0	0	0
0	4	4	0	0	0
0	0	0	0	0	0

Detección de bordes: Operador de Sobel

0	0	1	1	1	1
0	0	-1	0	1	1
0	0	-2	0	2	1
0	0	-1	0	1	1

Bordes horizontales

0	0	0	0	0	0
0	4	4	0	0	0
0	4	4	0	0	0
0	0	0	0	0	0

Detección de bordes: Operador de Sobel

0	0	1	1	1	1
0	0	1	-1	0	1
0	0	1	-2	0	2
0	0	1	-1	0	1

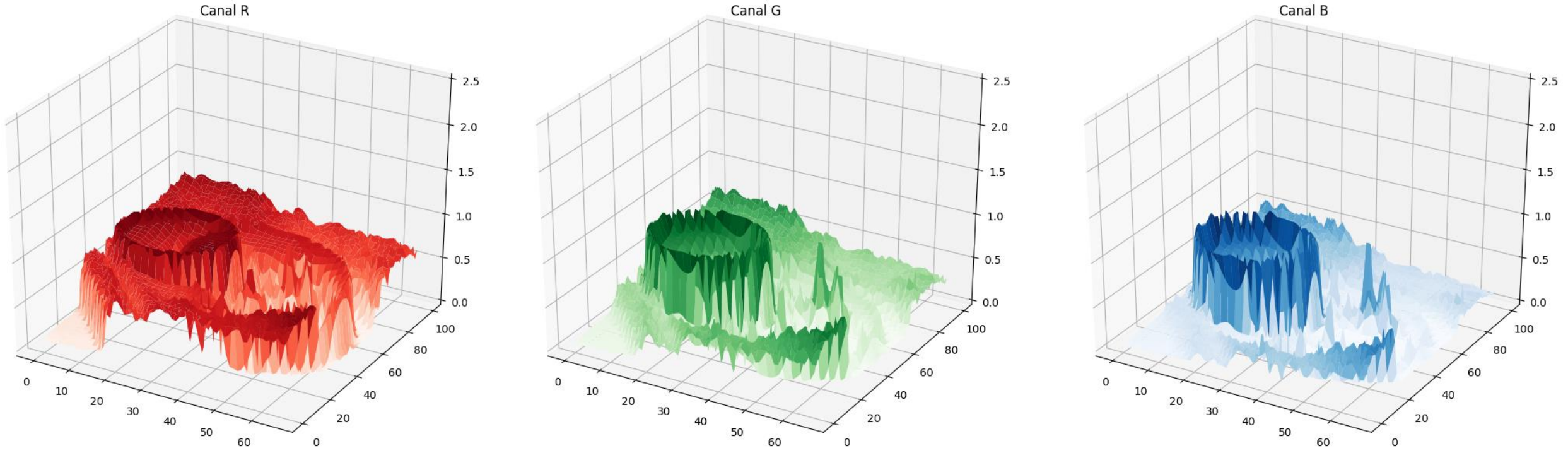
Bordes verticales

0	0	0	0	0	0
0	4	4	0	0	0
0	4	4	0	0	0
0	0	0	0	0	0

Práctica #6

Aplicar el operador de Sobel a *Coins*.

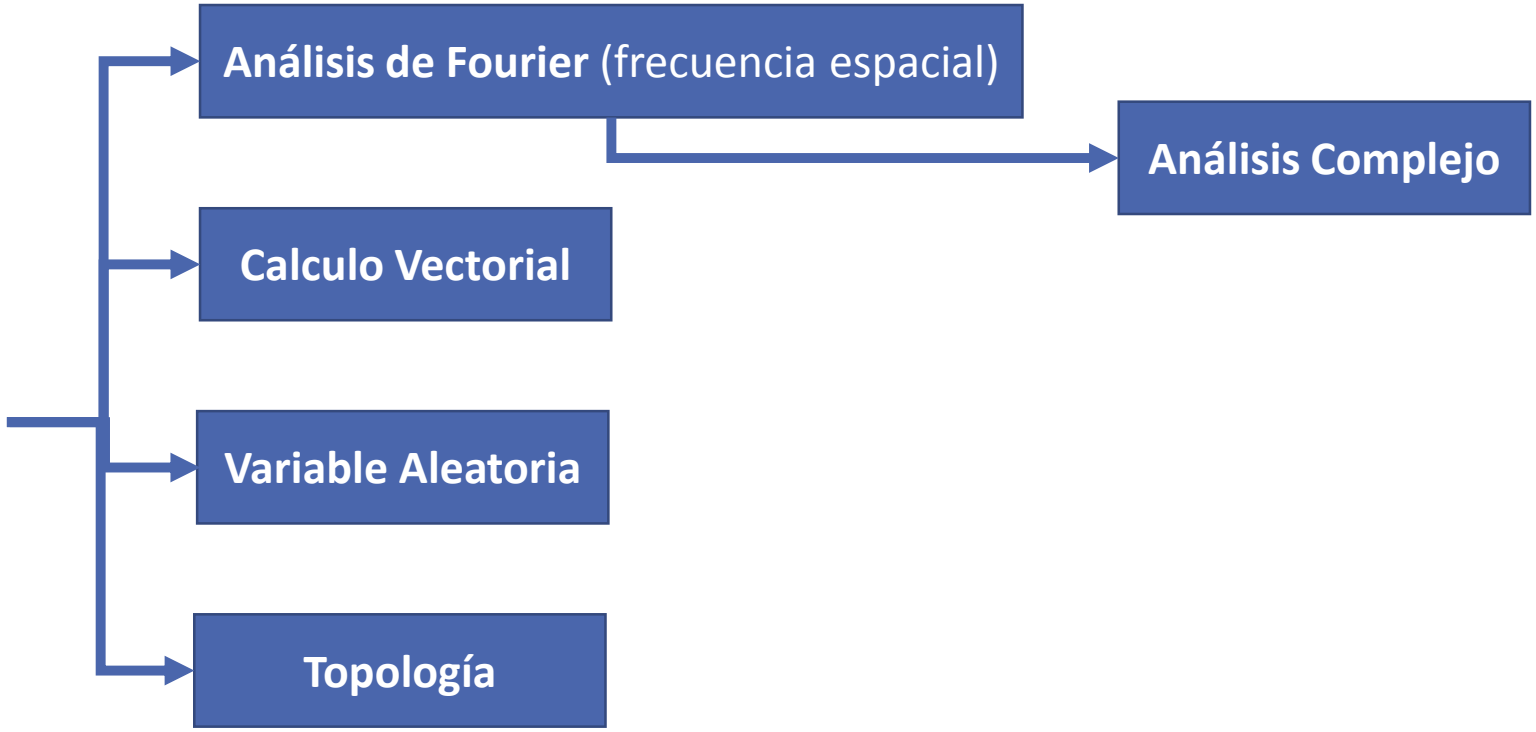
¿Qué es una imagen digital?



Respuesta elaborada. Es un conjunto de funciones de dos variables $\boldsymbol{F}(x, y)$.

Imágenes como funciones

$$\boldsymbol{F}(x,y) = \begin{bmatrix} F_R(x,y) \\ F_G(x,y) \\ F_B(x,y) \end{bmatrix}$$

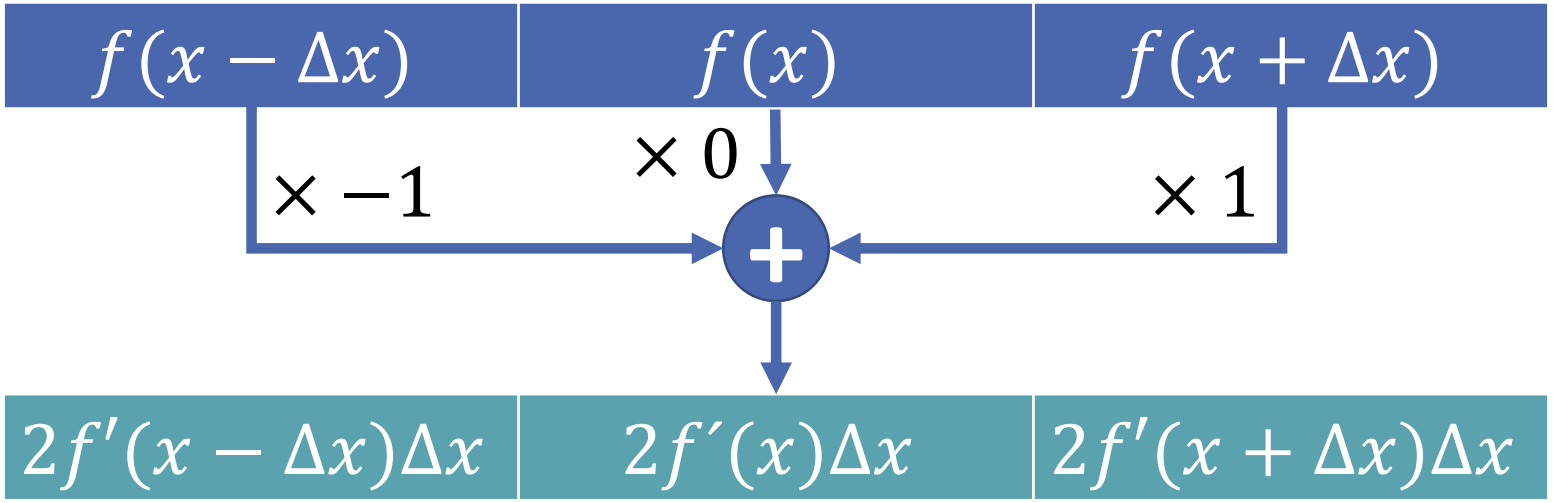


Considerar a las imágenes como funciones abre una enorme variedad de métodos de análisis y herramientas matemáticas para procesamiento.

Operador de Sobel como derivada numérica

$$f'(t) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \approx \frac{f(x - \Delta x) - f(x + \Delta x)}{2\Delta x}$$

$$0 < \Delta x \ll 0$$

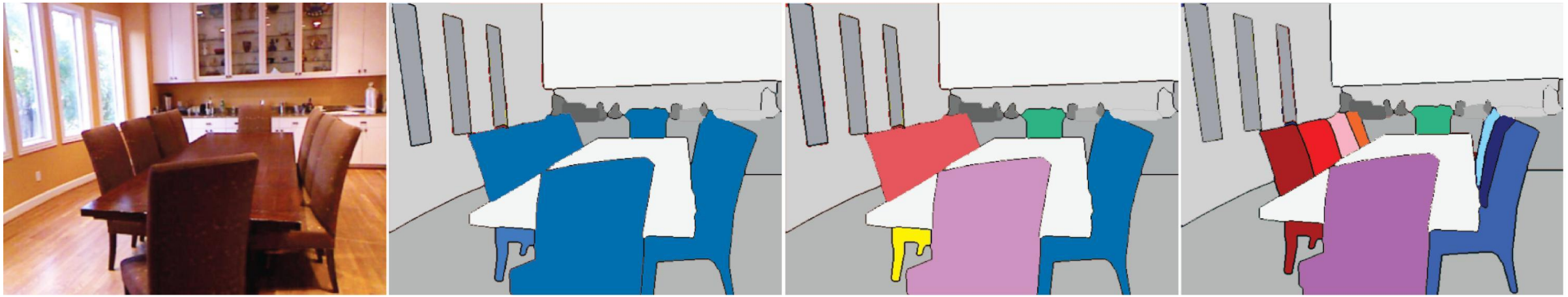


Los kernels de Sobel de las operaciones G_x y G_y son versiones escaladas de las derivadas numéricas parciales a lo largo de x y y.

Práctica #7

Obtener el campo de gradiente de *Coffee* (para uno de sus canales o su versión en escala de grises).

Segmentación de imágenes



Es el proceso de división de imágenes en regiones de interés.

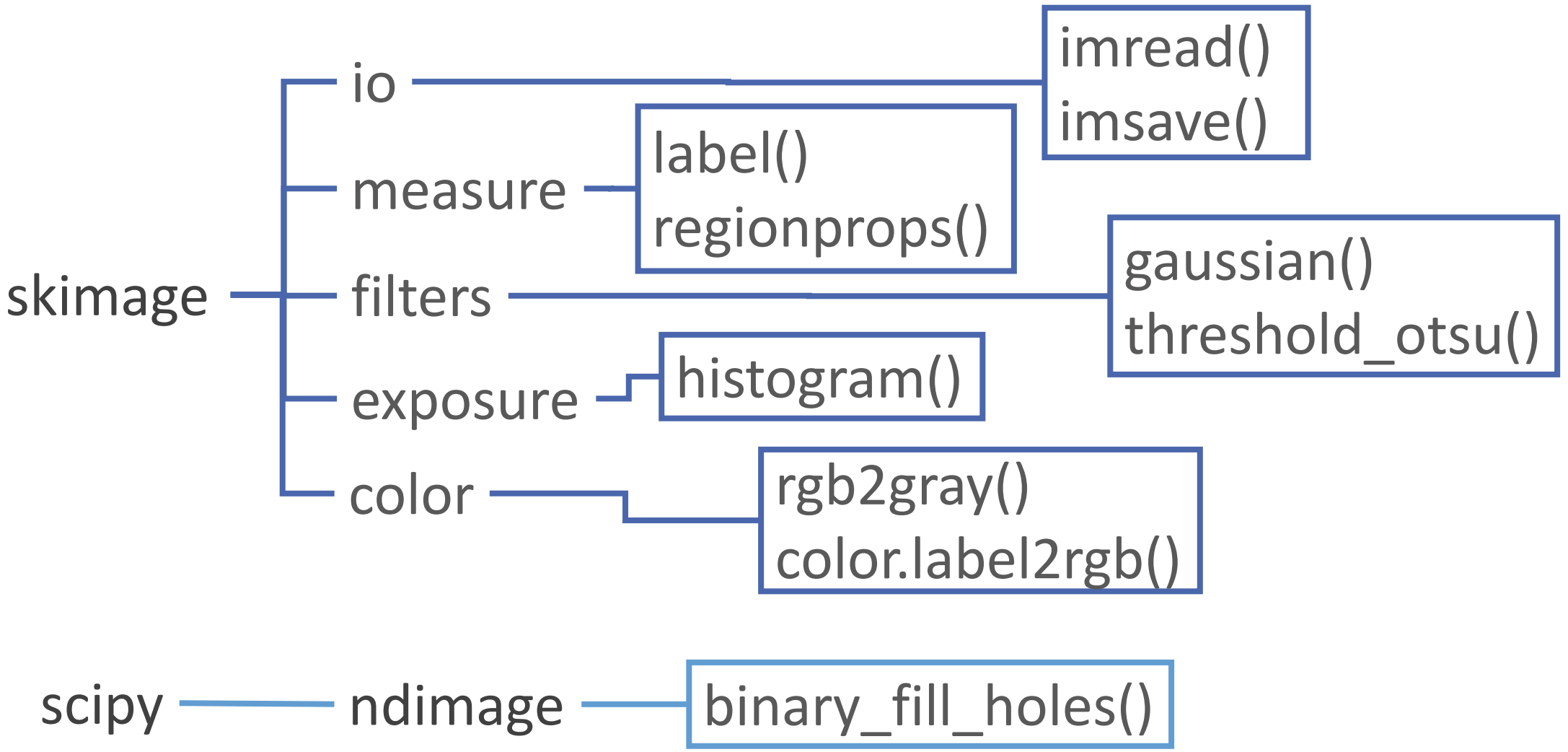
Puede dividirse en dos tipos:

- Segmentación por características (PDI, CV Clásica)
- Segmentación semántica (Machine Learning, Deep Learning)

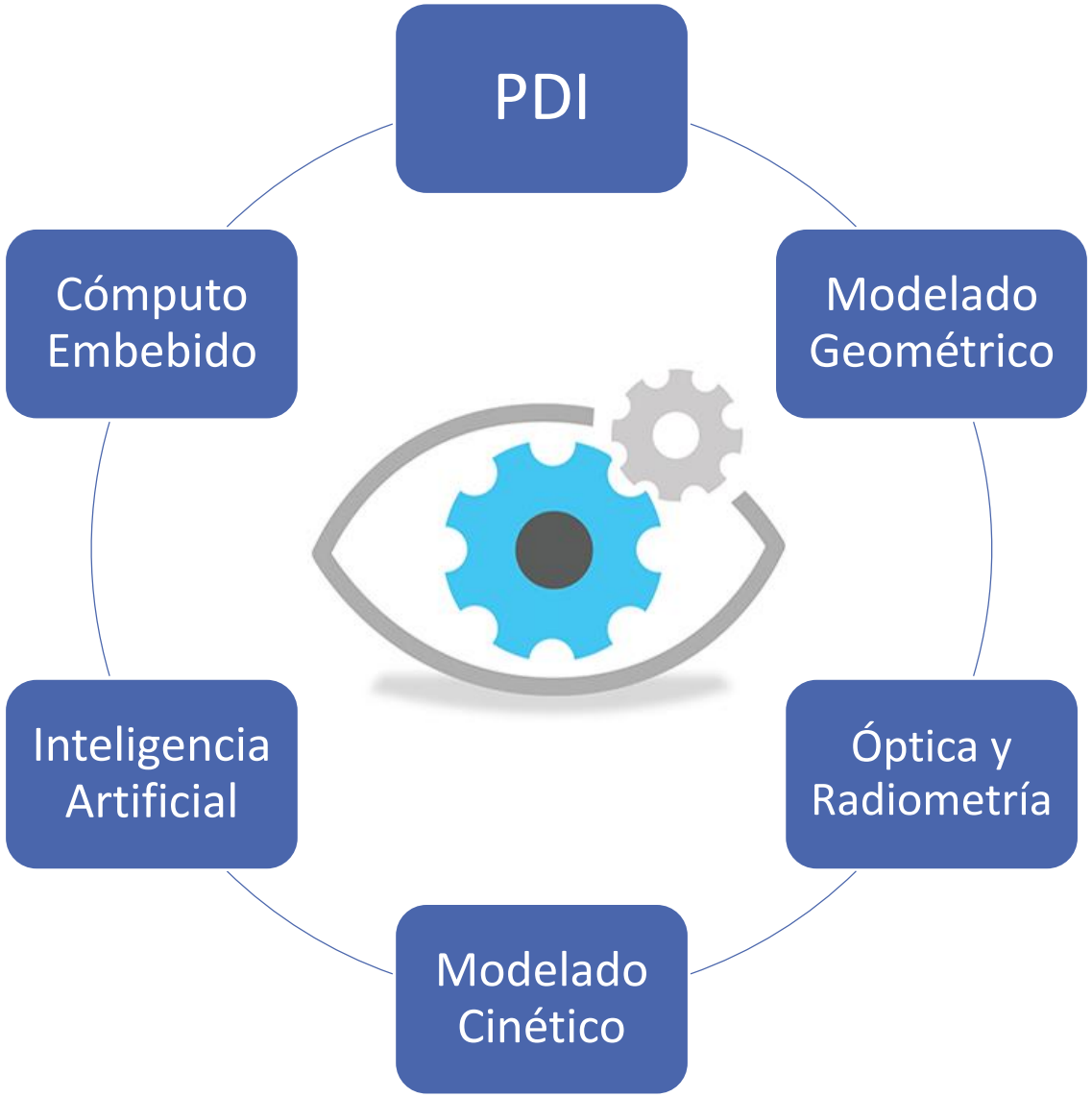
Práctica Final

Realizar una segmentación por umbral de la siguiente imagen:

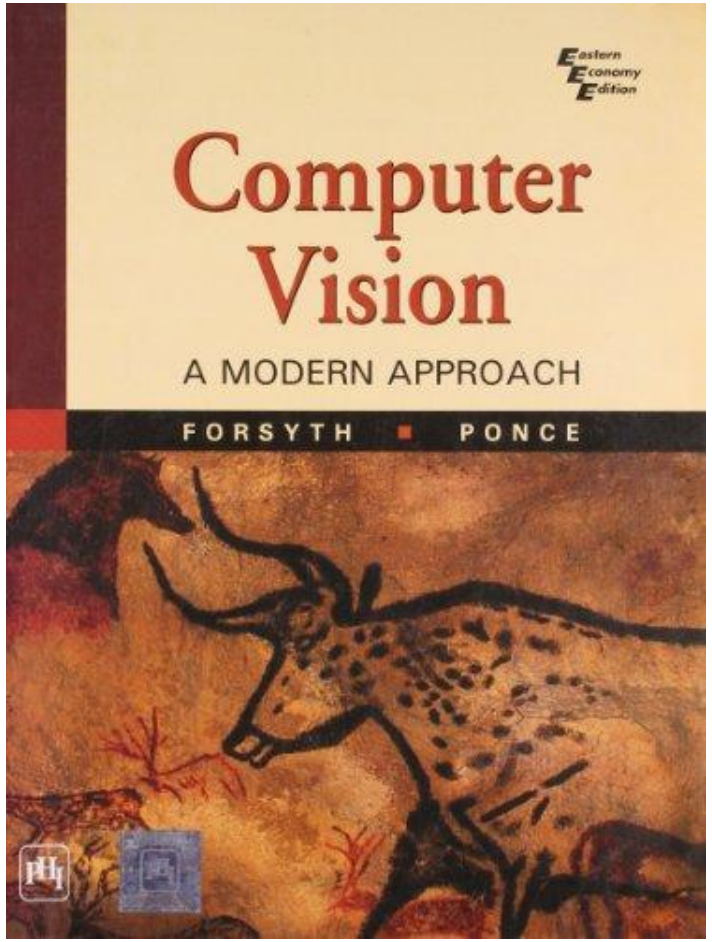
Mapa de la práctica final



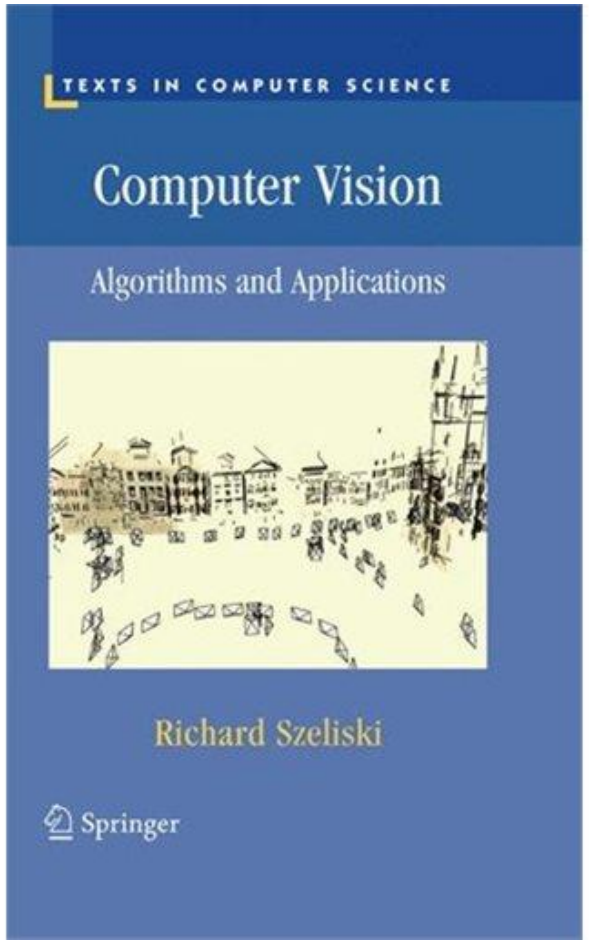
Observaciones finales: Visión por Computadora



Observaciones finales: Bibliografía



Computer Vision: A Modern Approach
Forsyth - Ponce



Computer Vision: Algorithms and Applications
Richard Szelinski

AJC
Yucatán

Contacto

r.escurib@gmail.com

r.e.escobar.3