

05MIAR-Aprendizaje supervisado

Tema 3 – Extracción de características

Curso Abril 23/24

The logo consists of the lowercase letters "viu" in white, sans-serif font, centered within a solid orange rounded rectangle.

viu

Universidad
Internacional
de Valencia

¿Dónde estudiar el Tema 3?

Esta presentación

01. Materiales docentes > Tema 3 – Sesión adicional. Descriptores de textura

Scripts del Tema 3

Índice

3.1. Estadísticos unidimensionales

3.1.1. Conceptos básicos

3.1.2. Variables

3.1.3. Análisis de variables cualitativas

3.1.4. Análisis de variables cuantitativas

3.2. Características bidimensionales

3.2.1. Introducción a los datos 2D

3.2.2. Procesado de imagen

3.2.3. Segmentación de imágenes

3.2.4. Descriptores de forma

3.2.5. Descriptores de textura

Índice

3.1. Estadísticos unidimensionales

3.1.1. Conceptos básicos

3.1.2. Variables

3.1.3. Análisis de variables cualitativas

3.1.4. Análisis de variables cuantitativas

3.2. Características bidimensionales

3.2.1. Introducción a los datos 2D

3.2.2. Procesado de imagen

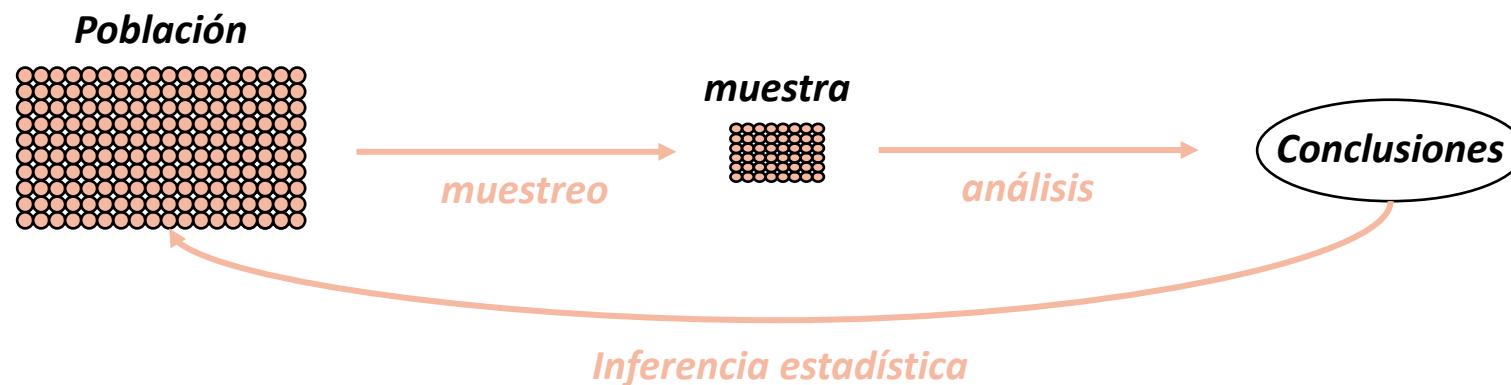
3.2.3. Segmentación de imágenes

3.2.4. Descriptores de forma

3.2.5. Descriptores de textura

3.1.1. Conceptos básicos

- **Población** – Es el conjunto de ítems (instancias, ocurrencias, individuos, sujetos, elementos, filas, etc.) que se desea estudiar, analizar o sobre el que se quiere extraer una serie de conclusiones. Por ejemplo: Todos los lotes de un producto que se va a fabricar en masa mediante un determinado proceso industrial.
- **Muestra** – Corresponde a cualquier subconjunto de la población. Es de vital importancia que la muestra sea representativa de la población. Para conseguirlo, se aplica un proceso llamado **muestreo aleatorio**. El objetivo es llegar a conocer a la población partiendo del análisis de una muestra.
- **Inferencia estadística** – Proceso de razonamiento por el que se obtienen conclusiones (con un margen de error conocido) sobre la población, a partir del análisis realizado sobre el conjunto de muestra.



3.1.1. Conceptos básicos

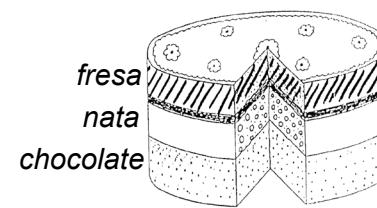
- Intención de voto ante unas Elecciones Generales en España.
 - *Población* → Número total de personas mayores de 18 años empadronadas en España.
 - *Muestra representativa* → Subconjunto balanceado de hombres y mujeres de diferente edad y distinta localización geográfica.
- Complicaciones de un determinado fármaco contra el Covid-19
 - *Población* → Número total de personas en el mundo.
 - *Muestra representativa* → Subconjunto balanceado de hombres y mujeres de diferentes edades y con diferentes complejiones, hábitos, afecciones, etc.
- Calidad de los ingredientes de una tarta.
 - *Población* → La tarta.
 - *Muestra* → Una porción de la tarta que contenga un trozo de cada uno de los ingredientes.

Ejemplos de muestreos erróneos

Población



Muestra



Índice

3.1. Estadísticos unidimensionales

3.1.1. Conceptos básicos

3.1.2. Variables

3.1.3. Análisis de variables cualitativas

3.1.4. Análisis de variables cuantitativas

3.2. Características bidimensionales

3.2.1. Introducción a los datos 2D

3.2.2. Procesado de imagen

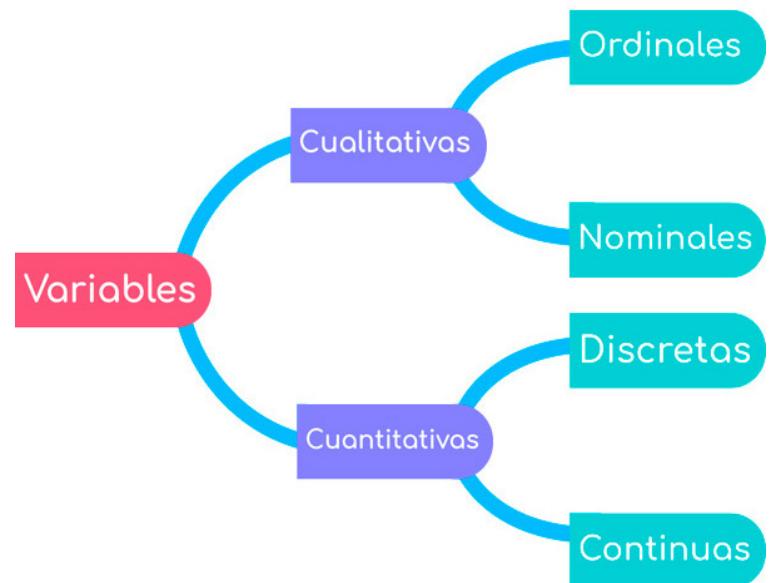
3.2.3. Segmentación de imágenes

3.2.4. Descriptores de forma

3.2.5. Descriptores de textura

3.1.2. Variables

Una **variable estadística** es cada una de las cualidades que poseen los individuos de una población. Es decir, una variable hace referencia al valor que toma una instancia (ítem, ocurrencia, observación, sujeto, fila, etc.) para una determinada característica (propiedad, atributo, columna, etc.). Dicho valor, como su propio nombre indica, puede variar en función del momento y/o del ítem en cuestión.



Las **variables ordinales** son aquellas cualidades no numéricas en las que existen un criterio de orden.

Las **variables nominales** son aquellas basadas en modalidades no numéricas que no admiten criterio de orden.

Las **variables discretas** son aquellas que solo pueden tomar un número finito de valores, es decir, son contables.

Las **variables continuas** son aquellas que pueden asumir un número infinito de valores en un rango continuo.

3.1.2. Variables

Variables cualitativas nominales



El sexo de una persona



El estado civil de una persona

Variables cualitativas ordinales



Satisfacción del cliente



Puesto conseguido en deporte

Variables cuantitativas discretas



El número de hijos



El número de empleados

Variables cuantitativas continuas



La estatura



El peso

Índice

3.1. Estadísticos unidimensionales

3.1.1. Conceptos básicos

3.1.2. Variables

3.1.3. Análisis de variables cualitativas

3.1.4. Análisis de variables cuantitativas

3.2. Características bidimensionales

3.2.1. Introducción a los datos 2D

3.2.2. Procesado de imagen

3.2.3. Segmentación de imágenes

3.2.4. Descriptores de forma

3.2.5. Descriptores de textura

3.1.3. Análisis de variables cualitativas

Tablas de frecuencia

Las **tablas de frecuencia** permiten resumir una muestra de datos cualitativos para facilitar su comprensión. De esta manera, se puede analizar la muestra de una población de manera organizada sin necesidad de recurrir a la muestra entera.

Ejemplo:

Se pretende conocer cuál es el nivel educativo en Valencia entre las personas de 30 y 40 años.

Para ello, se selecciona una muestra representativa de $K=500$ individuos empadronados en Valencia y que, al menos, acabaron sus estudios de secundaria.

Sujetos	Estudios
1	Grado
2	Bachillerato
3	Máster
4	Grado
5	Secundaria

● ● ●

499	Doctorado
500	Grado

Tabla de frecuencia



Estudios	Frecuencia absoluta	Frecuencia relativa	Frecuencia absoluta acumulada	Frecuencia relativa acumulada
Doctorado	21	0.0420	21	0.0420
Máster	52	0.1040	73	0.1460
Grado	145	0.2900	218	0.4360
Bachillerato	172	0.3440	390	0.7800
Secundaria	110	0.2200	500	1
TOTAL	500	1		

3.1.3. Análisis de variables cualitativas

Representación de los datos

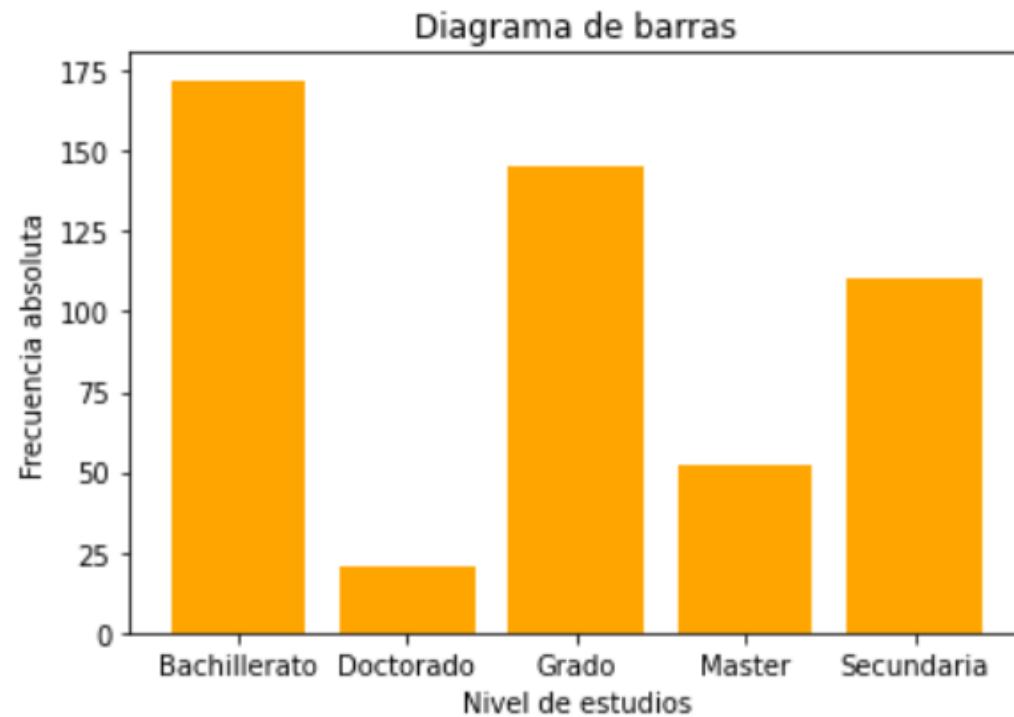
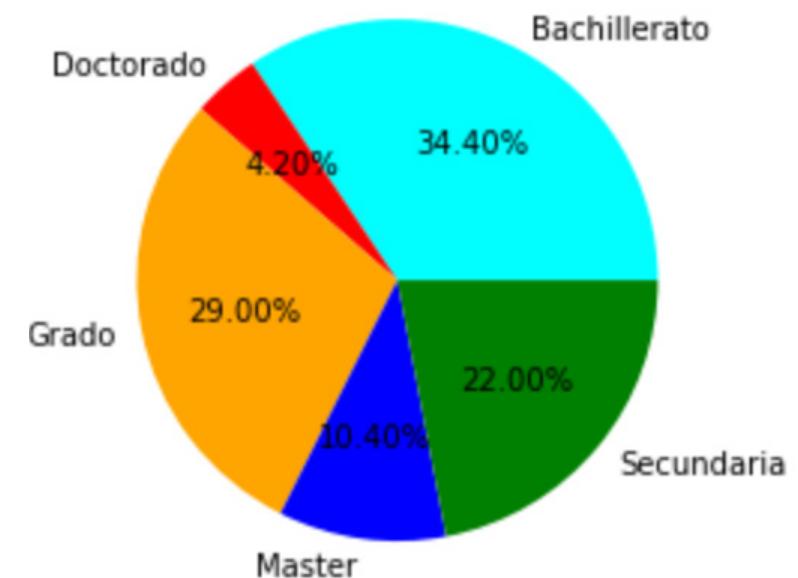


Diagrama de sectores



Índice

3.1. Estadísticos unidimensionales

3.1.1. Conceptos básicos

3.1.2. Variables

3.1.3. Análisis de variables cualitativas

3.1.4. Análisis de variables cuantitativas

3.2. Características bidimensionales

3.2.1. Introducción a los datos 2D

3.2.2. Procesado de imagen

3.2.3. Segmentación de imágenes

3.2.4. Descriptores de forma

3.2.5. Descriptores de textura

3.1.4. Análisis de variables cuantitativas

Medidas de tendencia central

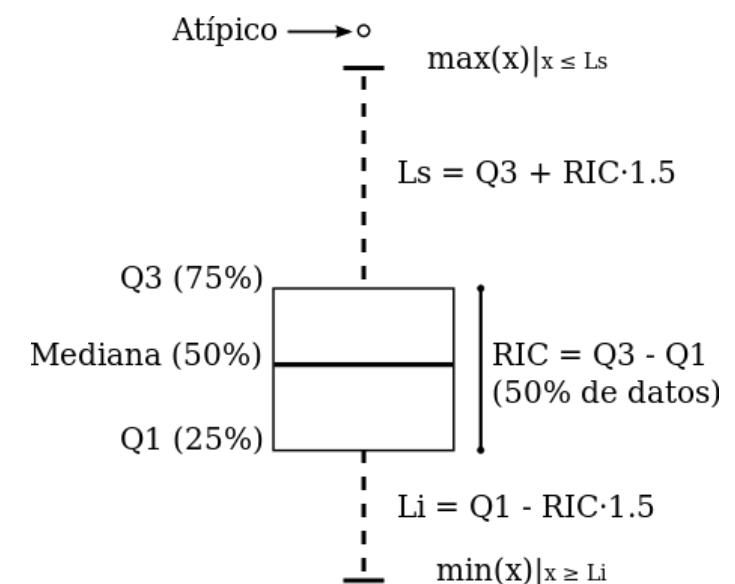
Media $\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$

Mediana

$$\left\{ \begin{array}{l} \text{Si } N \text{ es impar, } \tilde{X} \text{ es el valor en la posición } \frac{N}{2} \\ \text{Si } N \text{ es par, } \tilde{X} = \frac{1}{2} (X \left[\frac{N}{2} \right] + X \left[\frac{N}{2} + 1 \right]) \end{array} \right.$$

Moda Corresponde al valor más repetido del conjunto de datos X

Medidas de posición



3.1.4. Análisis de variables cuantitativas

Medidas de dispersión

Desviación típica

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N}}$$

Varianza

$$\sigma^2 = \frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N}$$

Medidas de distribución

Coeficiente de asimetría

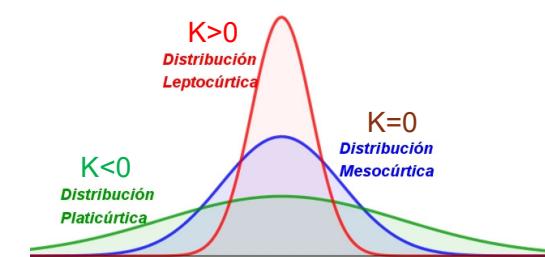
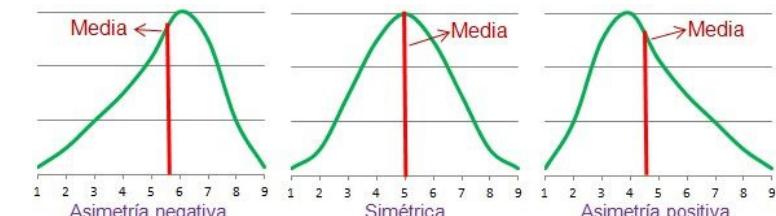
$$A = \frac{\sum_{i=1}^N (x_i - \bar{x})^3}{N \cdot \sigma^3}$$

Mide la simetría de la distribución de una variable respecto a la media aritmética.

Curtosis

$$K = \frac{\sum_{i=1}^N (x_i - \bar{x})^4}{N \cdot \sigma^4} - 3$$

Mide el grado de apuntamiento o achatamiento de la distribución de frecuencia, respecto a una distribución normal



Índice

3.1. Estadísticos unidimensionales

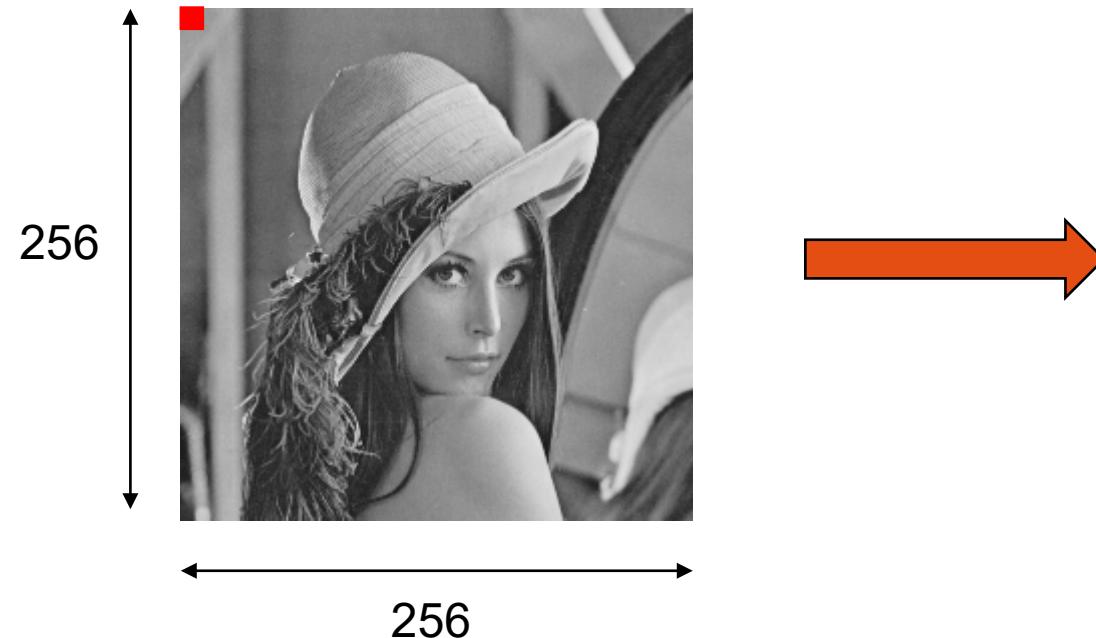
- 3.1.1. Conceptos básicos
- 3.1.2. Variables
- 3.1.3. Análisis de variables cualitativas
- 3.1.4. Análisis de variables cuantitativas

3.2. Características bidimensionales

- 3.2.1. Introducción a los datos 2D**
- 3.2.2. Procesado de imagen
- 3.2.3. Segmentación de imágenes
- 3.2.4. Descriptores de forma
- 3.2.5. Descriptores de textura

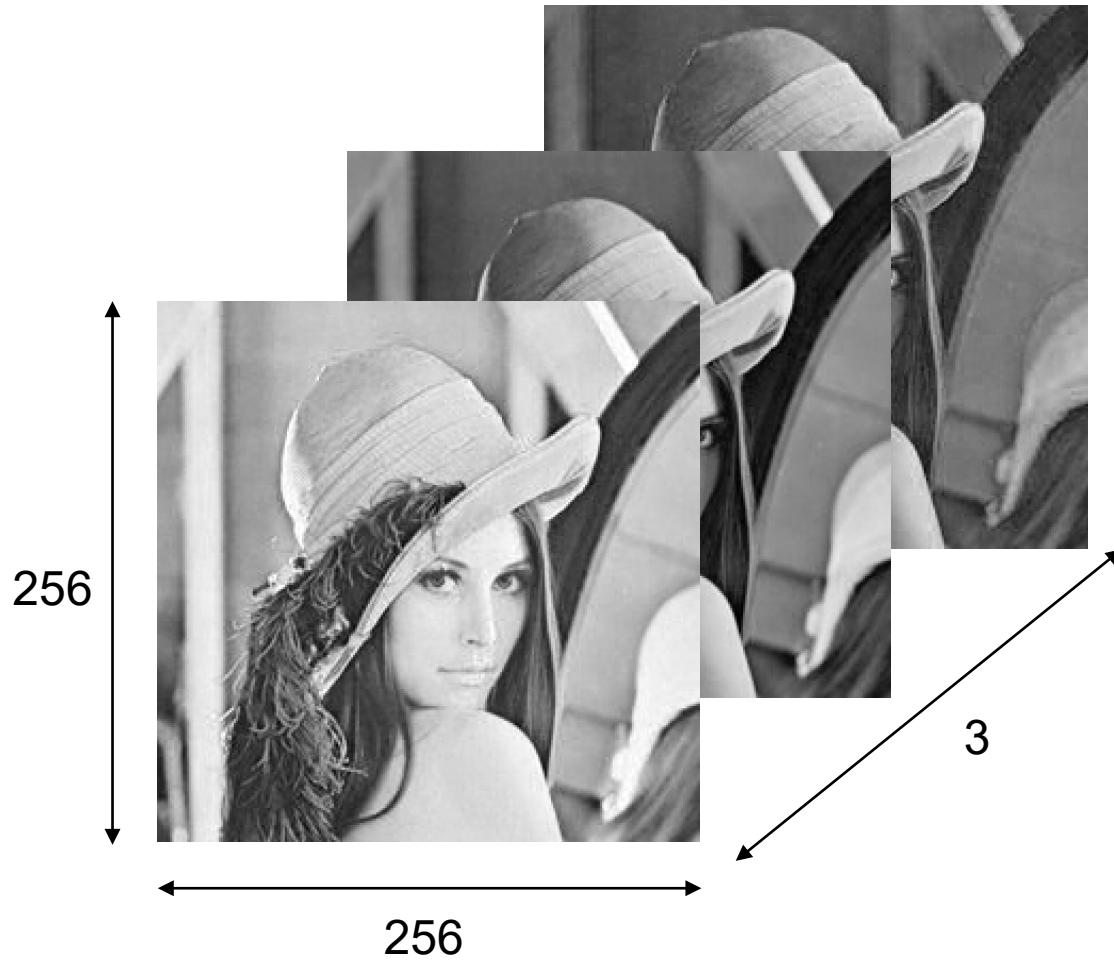
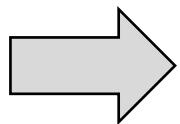
3.2.1. Introducción a los datos 2D

Una **imagen** es una función bidimensional $f(x,y)$ donde x e y representan las coordenadas espaciales y el valor de f , la intensidad de la imagen en dicho punto.

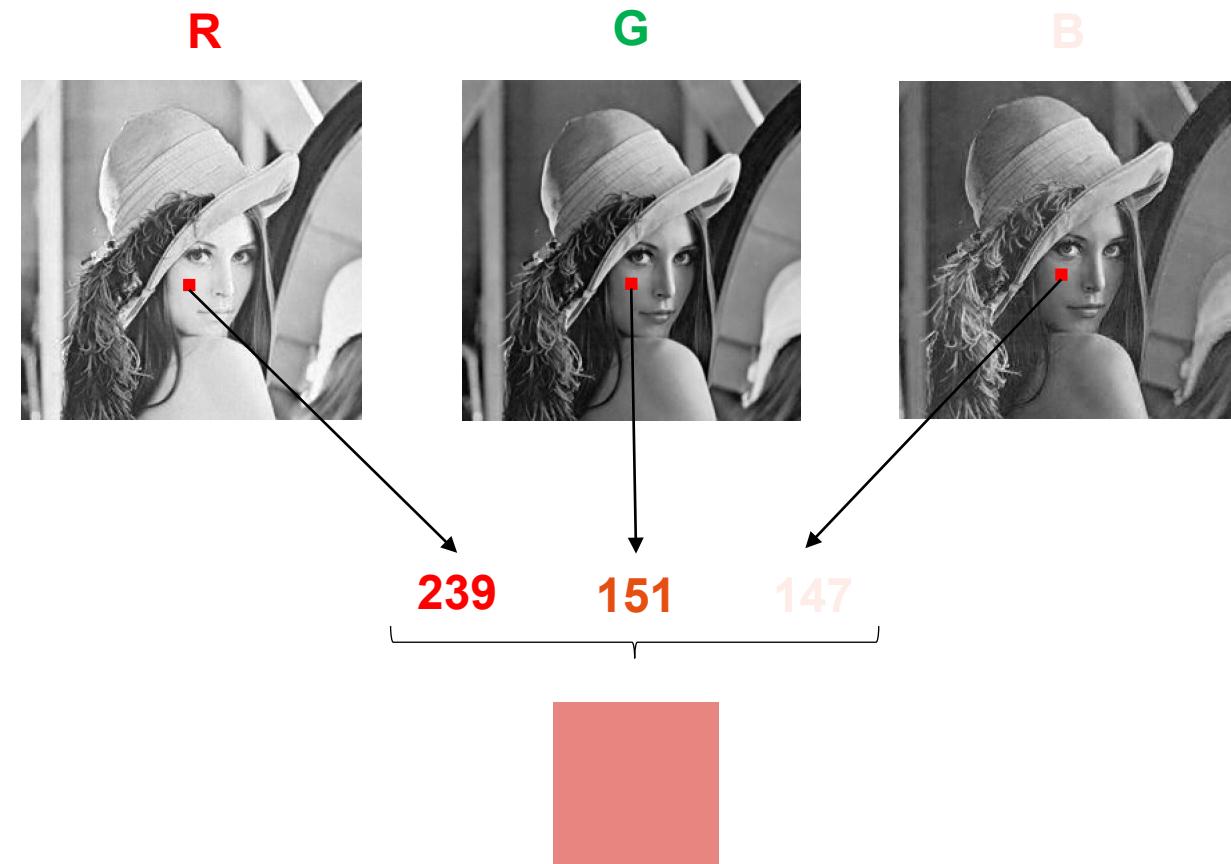
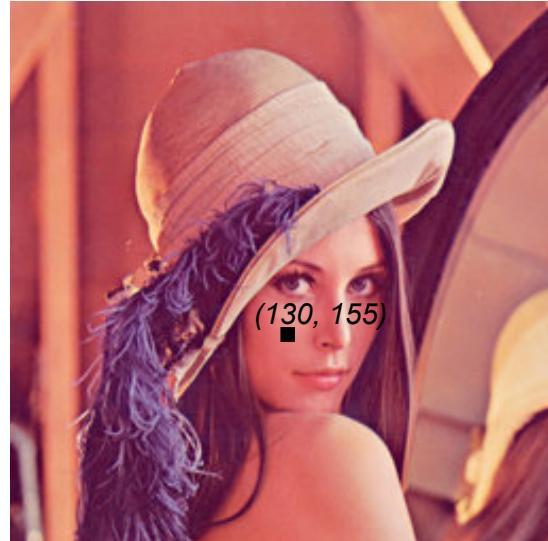


59	15	59	59	15	140
15	59	59	15	59	175
15	59	15	59	15	152
59	59	59	59	15	76
59	15	59	59	59	152
76	76	152	152	140	152
76	152	152	140	140	152
76	152	140	175	140	152

3.2.1. Introducción a los datos 2D



3.2.1. Introducción a los datos 2D



Índice

3.1. Estadísticos unidimensionales

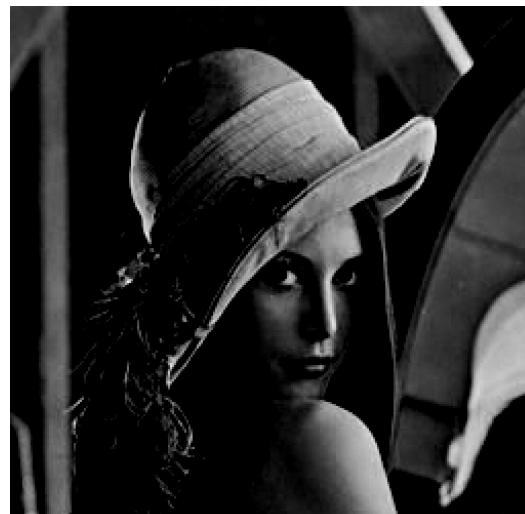
- 3.1.1. Conceptos básicos
- 3.1.2. Variables
- 3.1.3. Análisis de variables cualitativas
- 3.1.4. Análisis de variables cuantitativas

3.2. Características bidimensionales

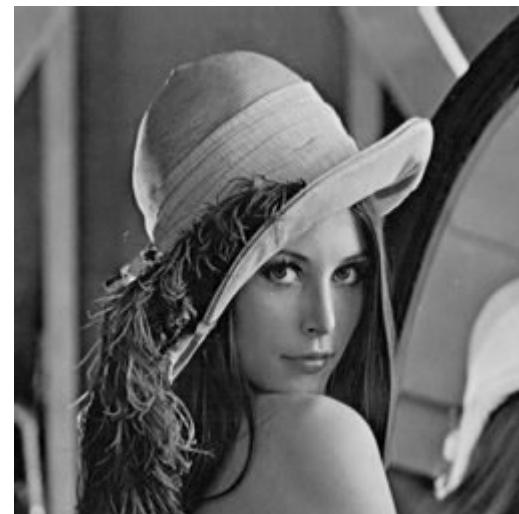
- 3.2.1. Introducción a los datos 2D
- 3.2.2. Procesado de imagen**
- 3.2.3. Segmentación de imágenes
- 3.2.4. Descriptores de forma
- 3.2.5. Descriptores de textura

3.2.2. Procesado de imagen

- Transformaciones de intensidad - Cambio de brillo



Original – 50



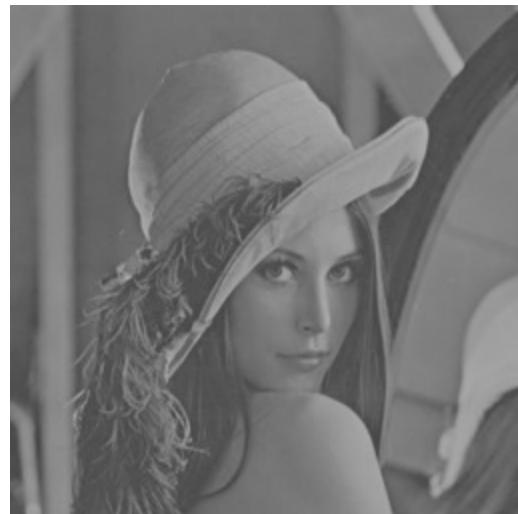
Original



Original + 50

3.2.2. Procesado de imagen

- Transformaciones de intensidad - Cambio de contraste



Original – 50



Original

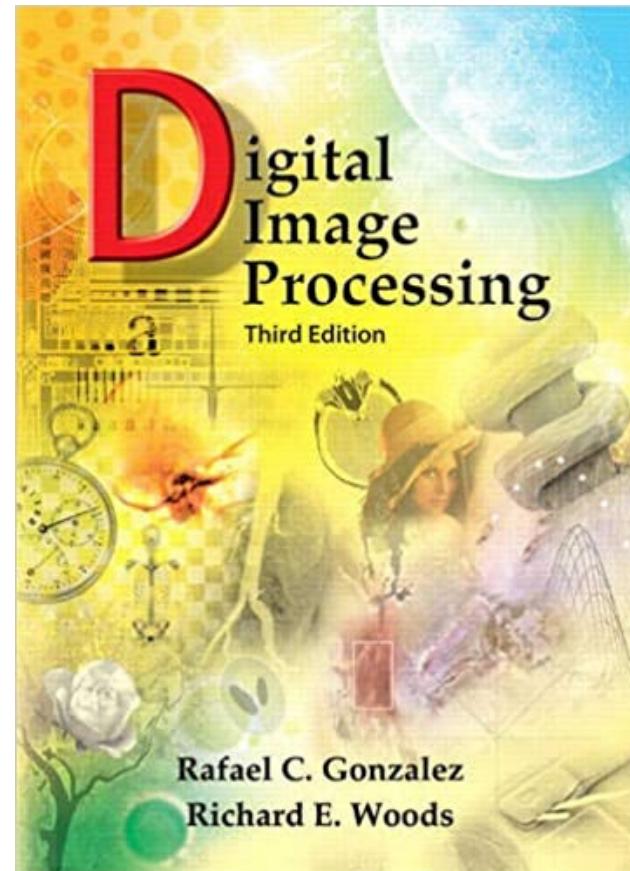


Original + 50

3.2.2. Procesado de imagen

Otras transformaciones de intensidad

- Transformación lineal a tramos
- Cambio de linealidad
- Negativo
- Ajuste del histograma
- Ecualización del histograma



3.2.2. Procesado de imagen

- Espacios de color – RGB a CMYK

$$R' = R/255$$

$$G' = G/255$$

$$B' = B/255$$

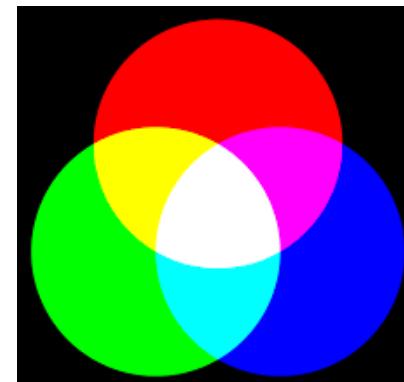
$$K = 1 - \max(R', G', B')$$

$$C = (1 - R' - K) / (1 - K)$$

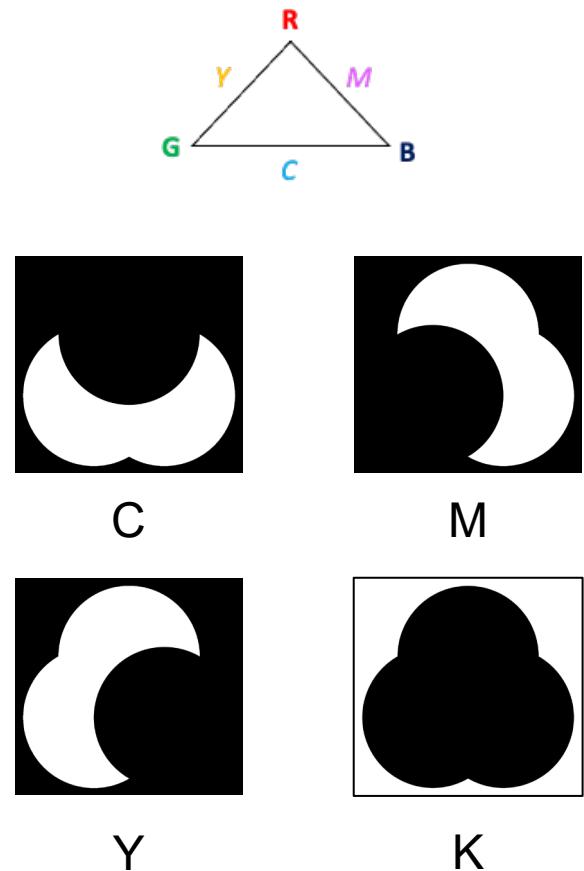
$$M = (1 - G' - K) / (1 - K)$$

$$Y = (1 - B' - K) / (1 - K)$$

CMYK



Original



3.2.2. Procesado de imagen

- Espacios de color – Otras transformaciones

```
import cv2
img = cv2.imread('Lena_RGB.png')

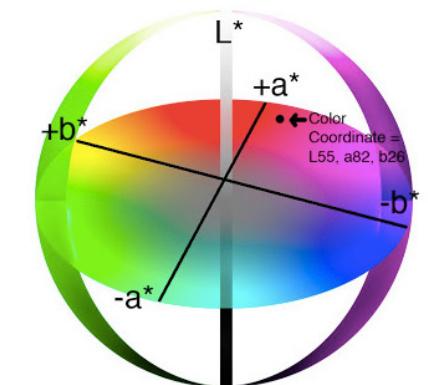
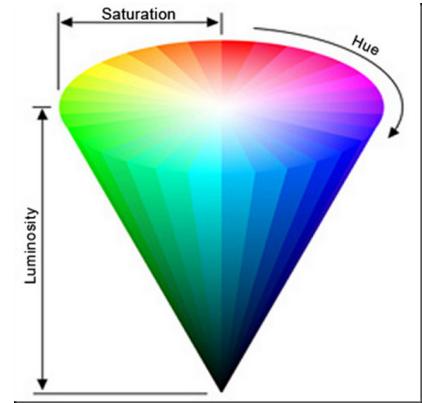
gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # gray-scale

HSV = cv2.cvtColor(img, cv2.COLOR_BGR2HSV) # (H)ue, (S)aturation and (V)alue

Lab = cv2.cvtColor(img, cv2.COLOR_BGR2Lab) # (L)uminosidad, a-b colores complementarios

YCrCb = cv2.cvtColor(img, cv2.COLOR_BGR2YCrCb) # Y-Luma, Cr-Cb crominancia rojo y azul
```

https://docs.opencv.org/3.4/d8/d01/group__imgproc__color__conversions.html



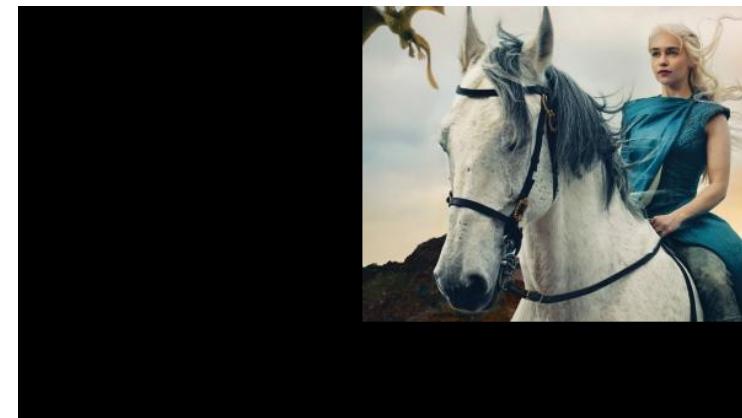
3.2.2. Procesado de imagen

- **Transformaciones geométricas - Traslación**

Una **traslación** es el desplazamiento de la posición de un objeto manteniendo las dimensiones de la imagen original. Si se conoce la magnitud del desplazamiento (t_x, t_y) en las direcciones x e y , respectivamente, se puede escribir la matriz de transformación M como:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

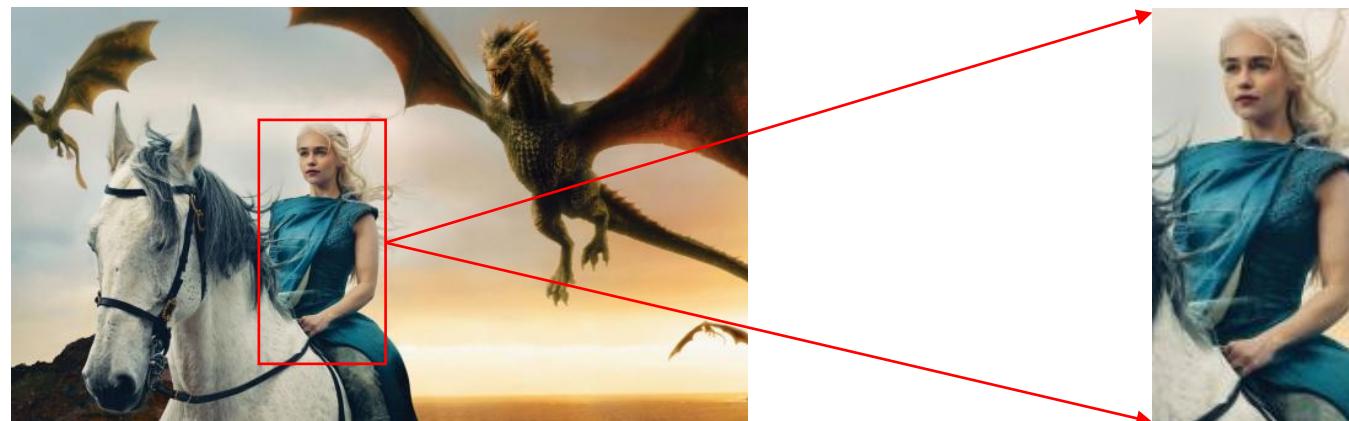
$$M = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix}$$



3.2.2. Procesado de imagen

- **Transformaciones geométricas - Cropping**

El **cropping** consiste en la selección y extracción de una determinada zona de la imagen que consideraremos como región de interés (ROI, del inglés – *region of interest*) para centrar los algoritmos únicamente en esa parte específica. Esto es muy interesante cuando queremos extraer características únicamente de una región concreta y el resto de la imagen puede no nos interesa o introduce ruido.



3.2.2. Procesado de imagen

- **Transformaciones geométricas - Rotación**

La **rotación** de una imagen consiste en aplicar un determinado ángulo θ a una matriz de rotación M .

OpenCV permite además **personalizar más la rotación** multiplicando por un factor de escala y modificando el centro de rotación.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$M = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$



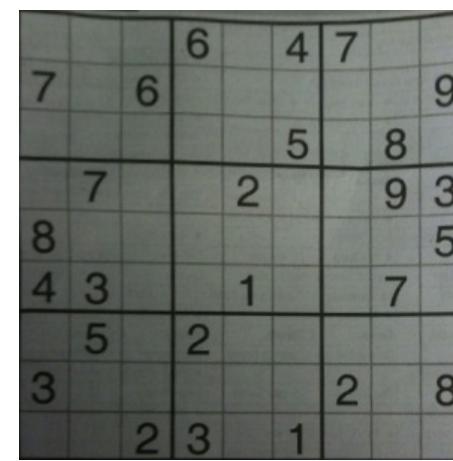
3.2.2. Procesado de imagen

- **Transformaciones geométricas - Perspectiva**

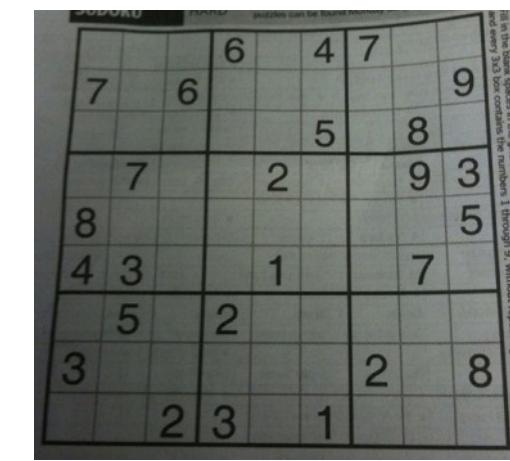
La **transformación de perspectiva** consiste en la rotación de la superficie de proyección (superficie de perspectiva) en un cierto ángulo alrededor θ . La idea de la transformación de perspectiva es: dada una imagen en el plano 2D, colocarla en una posición cualquiera del espacio 3D y después proyectarla sobre el plano de imagen $Z=1$.



Original



Perspectiva



Cropping

3.2.2. Procesado de imagen

- **Transformaciones geométricas - Flipping**

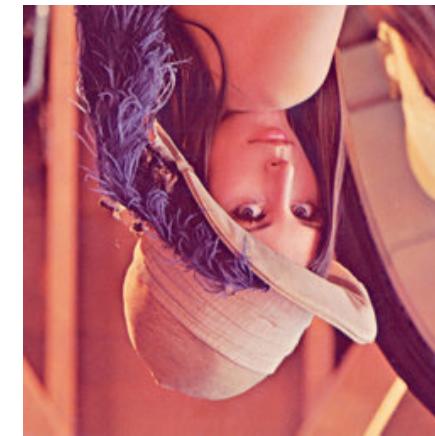
Las técnicas de **flipping** permiten voltear la imagen de diversas maneras para obtener la posición más conveniente a la hora de resolver un problema. Este método puede ser útil cuando las imágenes de una base de datos están adquiridas de manera diferente y es necesario establecer una posición estándar para poder extraer características uniformes.



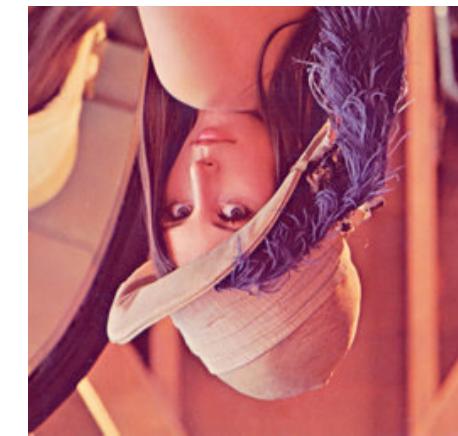
Original



Flip horizontal



Flip vertical



Flip hor. + vert.

Índice

3.1. Estadísticos unidimensionales

- 3.1.1. Conceptos básicos
- 3.1.2. Variables
- 3.1.3. Análisis de variables cualitativas
- 3.1.4. Análisis de variables cuantitativas

3.2. Características bidimensionales

- 3.2.1. Introducción a los datos 2D
- 3.2.2. Procesado de imagen
- 3.2.3. Segmentación de imágenes**
- 3.2.4. Descriptores de forma
- 3.2.5. Descriptores de textura

3.2.3. Segmentación de imágenes

■ Introducción

Normalmente, para extraer características basadas en **descriptores de forma**, es necesario, previamente, llevar a cabo alguna técnica de segmentación que permita delimitar la región de interés que se desea analizar.

Los **algoritmos de segmentación** se basan en particionar la imagen digital en diferentes segmentos (objetos) para simplificar la representación de la imagen y facilitar la comprensión de su contenido.

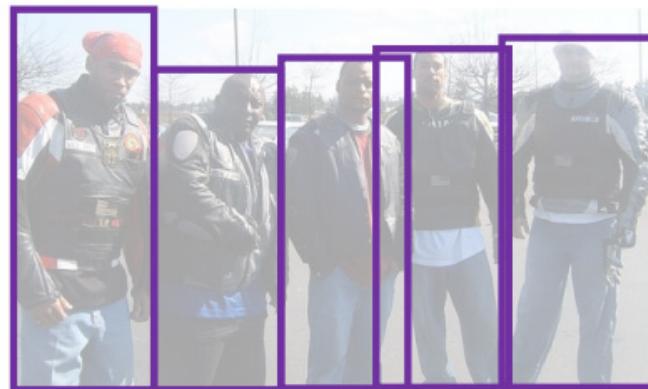
Las técnicas de segmentación permiten **asignar una etiqueta a cada píxel** de la imagen de manera que los píxeles con la misma etiqueta comparten características similares.

Existen dos tipos de segmentación:

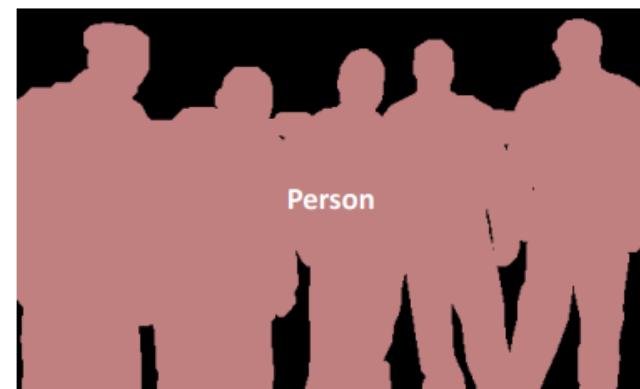
- **Segmentación semántica.** Consiste en un problema de clasificación a nivel de píxel que permite asignar una etiqueta de objeto a cada píxel para delimitar regiones de interés. Por ejemplo, diferenciar un grupo de personas del fondo de la imagen.
- **Segmentación de instancia.** Consiste en una tarea más compleja que implica llevar a cabo una detección de objetos para asignar una etiqueta de instancia a cada objeto. Es decir, la segmentación de instancia permite identificar objetos que comparten las mismas propiedades como objetos independientes. Por ejemplo, diferenciar cada persona de un grupo de personas en una imagen.

3.2.3. Segmentación de imágenes

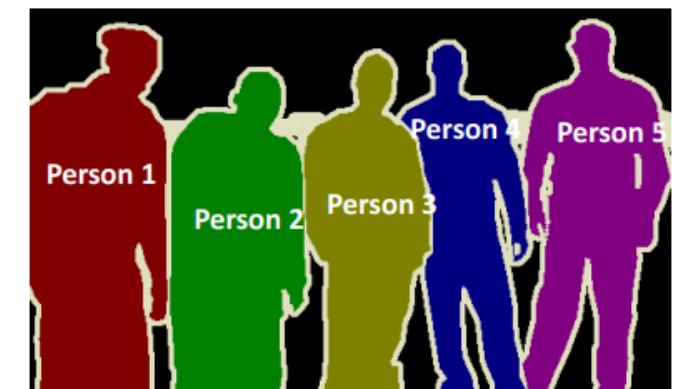
- Introducción



Detección de objetos



Segmentación semántica

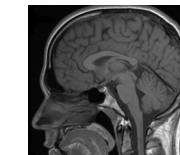
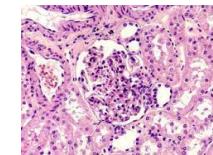


Segmentación de instancia

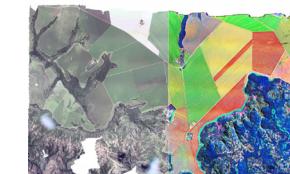
3.2.3. Segmentación de imágenes

■ Aplicaciones

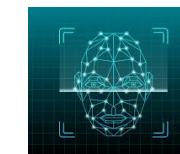
Sistemas de ayuda al diagnóstico mediante análisis de imágenes médicas. Por ejemplo: resonancia magnética, TAC, imagen histológica, imágenes de fondo de ojo, radiografías, ultrasonido, etc.



Reconocimiento de objetos. Por ejemplo, detección de peatones, localización de objetos en imágenes por satélite (carreteras, bosques, cultivos, etc.)



Reconocimiento de patrones en temas de seguridad. Por ejemplo, reconocimiento de caras (*face recognition*), reconocimiento de huellas dactilares (*fingerprint recognition*) o reconocimiento del iris.



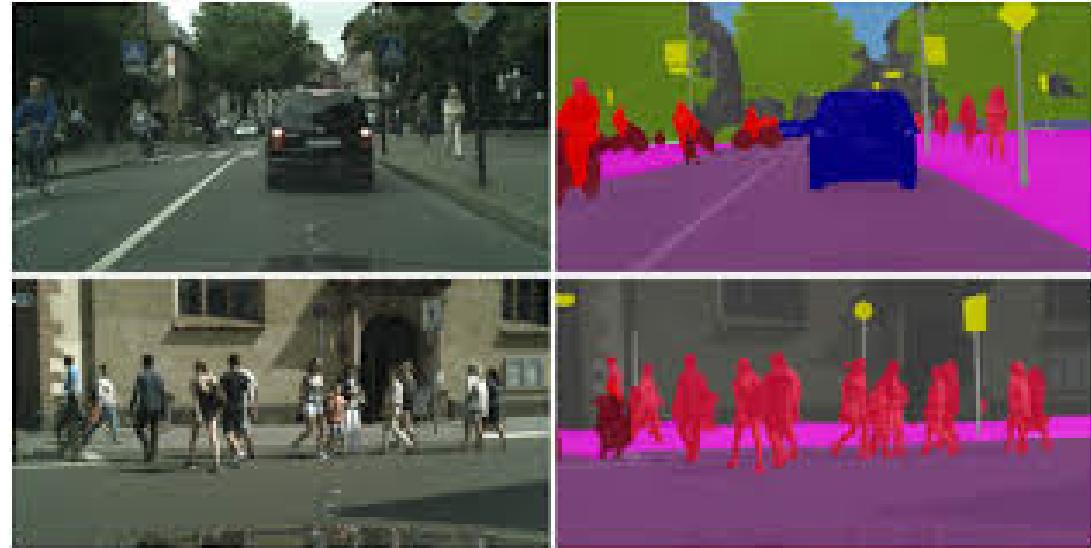
Sistemas de control de tráfico y conducción autónoma. Por ejemplo, detección de señales de tráfico, reconocimiento de matrículas, *self driving*.



3.2.3. Segmentación de imágenes

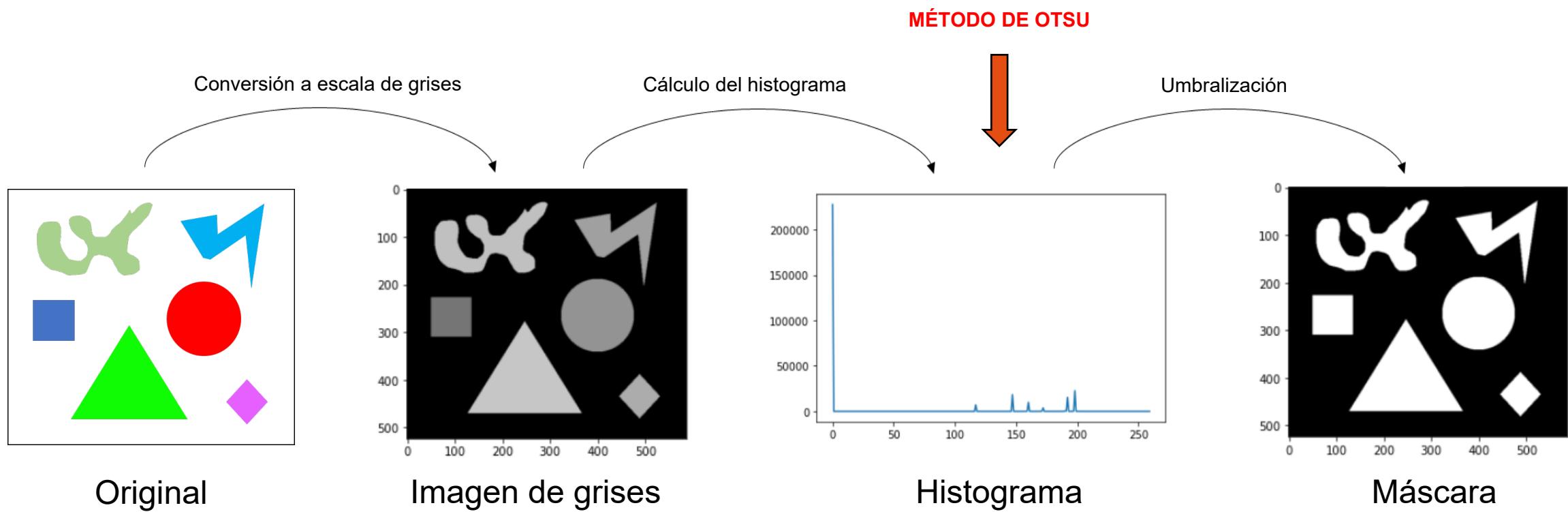
- **Técnicas**

- Métodos de umbralización (*thresholding*)
- **Métodos basados en el espacio de color**
- **Detección de bordes**
- Crecimiento de regiones (*región growing*)
- Métodos de agrupación (*clustering*)
- **Watershed**
- Autoencoders (Redes Neuronales Convolucionales)



3.2.3. Segmentación de imágenes

- Técnicas – Thresholding



3.2.3. Segmentación de imágenes

- Técnicas – Thresholding + Otsu

El **método de Otsu** es un procedimiento no paramétrico que permite obtener el umbral óptimo maximizando la varianza entre clases. Búsqueda exhaustiva.

Algoritmo. Se parte de una imagen con **N** píxeles que puede tomar **L** valores de grises diferentes

$$c_1 = [0, 1, 2, \dots, t]$$

1. Los píxeles se dividen en dos clases

$$c_2 = [t + 1, t + 2, \dots, L]$$

2. Se define la probabilidad del nivel de gris i

$p_i = f_i/N$, siendo f_i la frecuencia de repetición del nivel de gris i

3. Se calculan las medias para cada una de las clases

$$\mu_1 = \sum_{i=1}^t \frac{i \cdot p_i}{\omega_1(t)} \quad \text{donde} \quad \begin{cases} \omega_1(t) = \sum_{i=1}^t p_i \\ \omega_2(t) = \sum_{i=t+1}^L p_i \end{cases}$$

4. Se obtiene la intensidad media total de la imagen

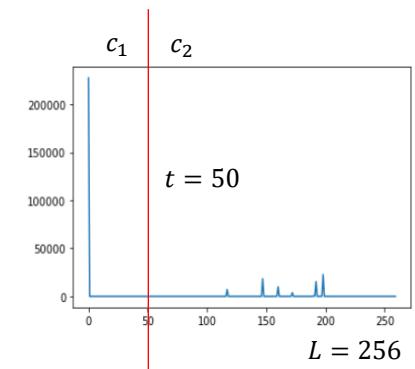
$$\mu_T = \omega_1 \cdot \mu_1 + \omega_2 \cdot \mu_2$$

5. Se calcula la varianza entre clases

$$\sigma^2 = \omega_1(\mu_1 - \mu_T)^2 + \omega_2(\mu_2 - \mu_T)^2$$

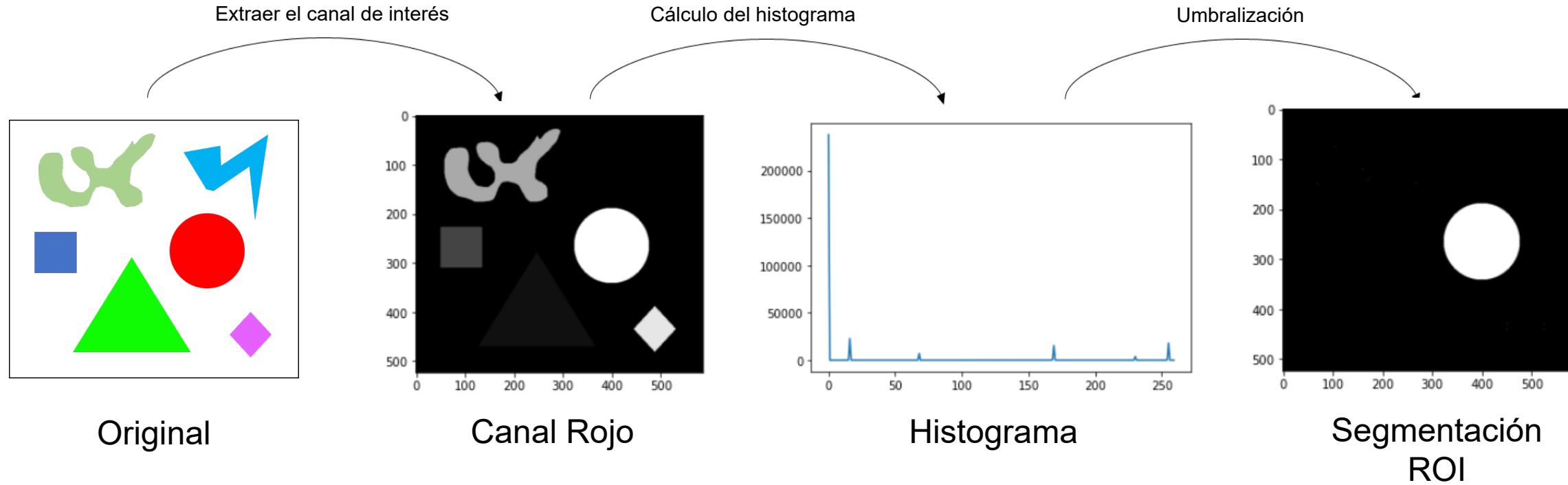
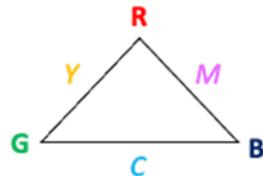
6. Se selecciona como el umbral óptimo aquel que maximiza la varianza

$$t^* = \text{Max}\{\sigma^2(t)\}$$



3.2.3. Segmentación de imágenes

- Técnicas – Espacios de color



3.2.3. Segmentación de imágenes

- Técnicas – Detección de bordes con Canny

1. Detector de bordes con Sobel

El **operador Sobel** se basa en el **cálculo de la primera derivada** para obtener el **gradiente de la intensidad** de una imagen en cada píxel. Es decir, **mide cambios de intensidad en la imagen**. Para ello, aplica **dos máscaras de convoluciones con kernels de 3x3 elementos**, uno para los **cambios horizontales** y otro para los **verticales**. De esta manera, se obtienen G_x y G_y que representan las **aproximaciones de las derivadas de intensidades**.

En cada píxel de la imagen, **se combinan las aproximaciones de las derivadas** para obtener la **magnitud G** y la **dirección Θ** del gradiente. La dirección de los píxeles puede ser en estos ángulos (0, 45, 90, o 135)

2. Filtrado de bordes mediante la supresión non-máximo

Esta técnica permite **adelgazar los bordes** basándose en el **gradiente**. El objetivo es **eliminar los píxeles que no corresponden a un borde**. Se **compara** el valor de intensidad de **cada píxel con el valor de intensidad de los píxeles vecinos en la dirección del gradiente**. Si es **mayor**, ese píxel se considera **borde**.

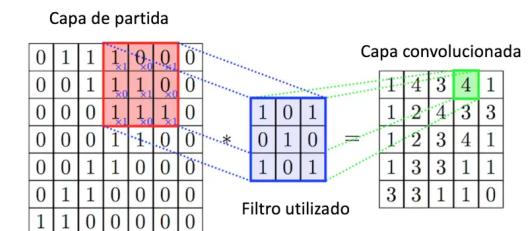
3. Umbralización por histéresis

El umbral por histéresis se centra en establecer **dos umbrales (uno máximo y otro mínimo)**.

- Si el valor del píxel es **mayor que el umbral máximo**, el píxel se considera parte del **borde**.
- Si el valor del píxel es **menor que el umbral mínimo**, el píxel **no** se considera parte del **borde**.
- Si el valor del píxel está **entre el mínimo y el máximo umbral**, será **borde si está conectado a un píxel que sí es parte del borde**.

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

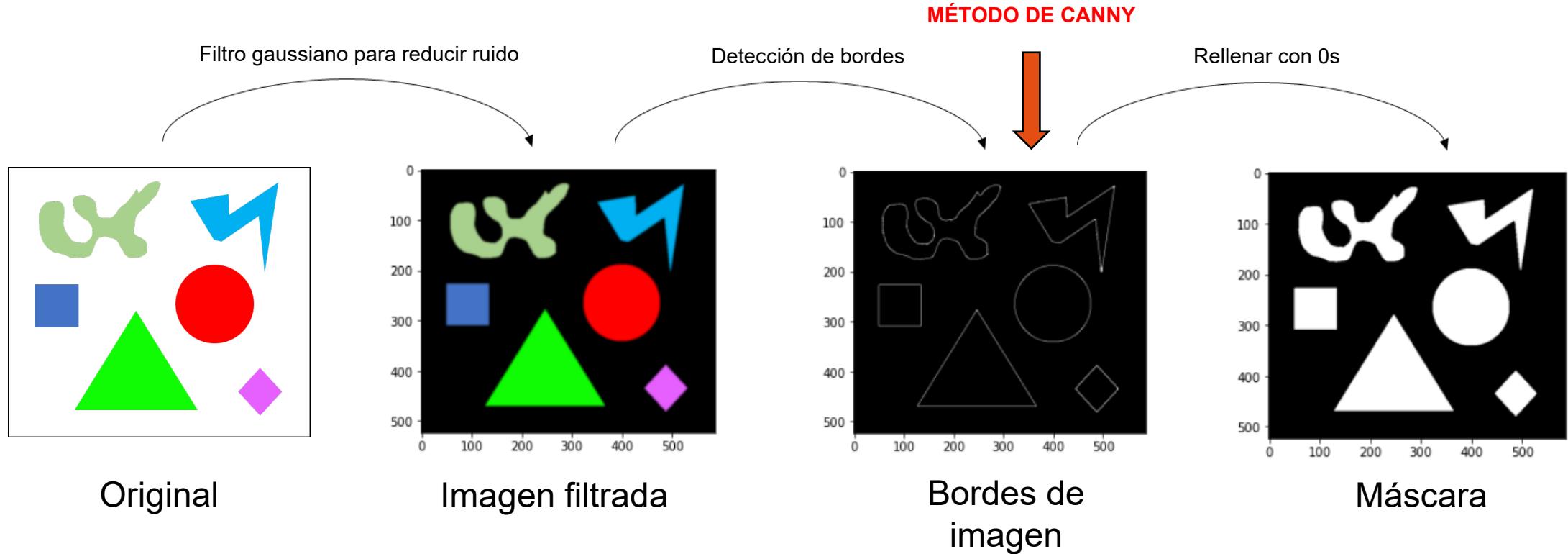
$$G = \sqrt{G_x^2 + G_y^2} \quad \Theta = \arctan\left(\frac{G_y}{G_x}\right)$$



3.2.3. Segmentación de imágenes

■ Técnicas – Detección de bordes

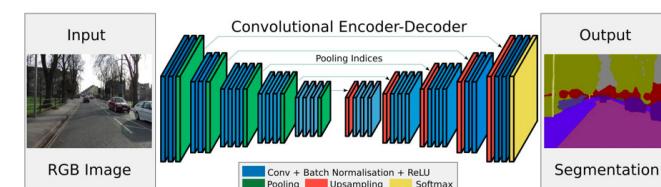
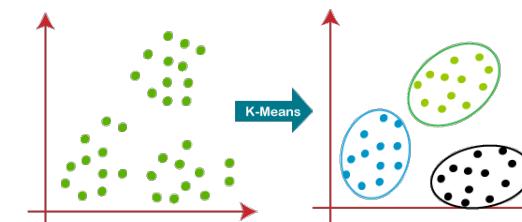
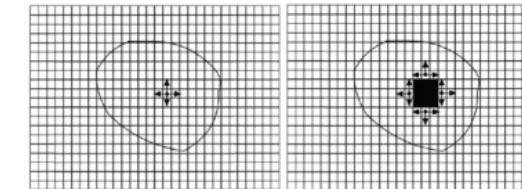
Cuando hablamos de **bordes de una imagen**, hablamos de los píxeles donde hay un **cambio de intensidad**.



3.2.3. Segmentación de imágenes

- **Técnicas - Otras**

- Métodos de umbralización (thresholding)
- Métodos basados en el espacio de color
- Detección de bordes
- Crecimiento de regiones (región growing)
- **Métodos de agrupación (*clustering*)**
- **Watershed**
- **Autoencoders (Redes Neuronales Convolucionales)**



Índice

3.1. Estadísticos unidimensionales

- 3.1.1. Conceptos básicos
- 3.1.2. Variables
- 3.1.3. Análisis de variables cualitativas
- 3.1.4. Análisis de variables cuantitativas

3.2. Características bidimensionales

- 3.2.1. Introducción a los datos 2D
- 3.2.2. Procesado de imagen
- 3.2.3. Segmentación de imágenes
- 3.2.4. Descriptores de forma**
- 3.2.5. Descriptores de textura

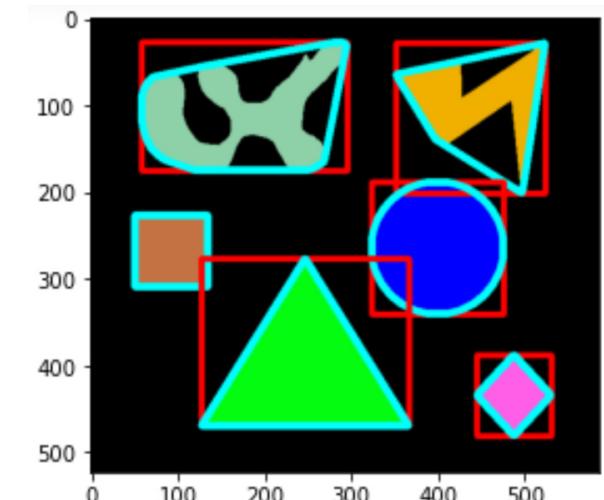
3.2.4. Descriptores de forma

▪ Descriptores geométricos

Los **descriptores geométricos** analizan la forma estructural de **objetos binarios** y sus **contornos**. Se pueden utilizar para **identificar objetos** y **reconocer patrones** en una imagen.

Algunos ejemplos son:

- **Área** → Mide el número de píxeles que contiene la región de interés.
- **Bounding box (bbox)** → Consiste en el rectángulo más pequeño capaz de albergar la región de interés (rectángulo rojo).
- **Área de la bbox** → Número de píxeles que contiene la bounding box.
- **Convex hull image** → Consiste en el polígono convexo más pequeño capaz de albergar la región de interés (polígono cyan).
- **Área de la convex image** → Mide el número de píxeles dentro de la convex image.
- **Excentricidad** → Es la relación de la distancia entre los focos de la elipse y su longitud del eje principal. Una elipse con excentricidad=0 es un círculo, mientras que una excentricidad =1 denota un segmento de línea.
- **Diámetro equivalente** → Consiste en el diámetro de un círculo cuya área sea la misma que la región de interés



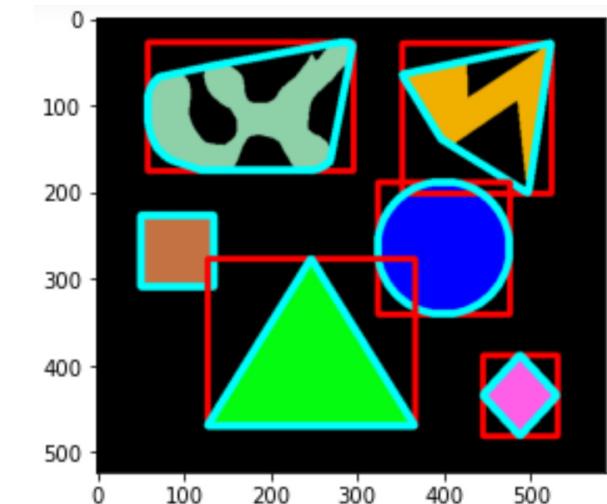
3.2.4. Descriptores de forma

■ Descriptores geométricos

- **Extensión** → Se define como el ratio de píxeles en la región con respecto al número total de píxeles contenidos en la bounding box.
- **Longitud del eje mayor** → Se refiere a la longitud del eje mayor de una elipse que tiene el mismo segundo momento central normalizado que la región.
- **Longitud del eje menor** → Igual para el eje menor de la elipse.
- **Orientación** → Es el ángulo que forma la elipse anterior con respecto al eje horizontal. El valor se expresa en grados en un rango de [-90, 90].
- **Perímetro** → Se define como el número de píxeles que forman parte del contorno del objeto de interés.
- **Solidez** → Corresponde con la proporción de píxeles que contiene el área respecto a la convex hull image. En el caso de polígonos regulares, su valor será 1.

A partir de estos descriptores se podrían extraer otros de interés como:

- **Compacidad** → Mide la circularidad del objeto. Se calcula como el cociente entre el área y el perímetro al cuadrado.
- **Diámetro de Feret** → Se define como el cociente entre el área del objeto y la bounding box.



Índice

3.1. Estadísticos unidimensionales

- 3.1.1. Conceptos básicos
- 3.1.2. Variables
- 3.1.3. Análisis de variables cualitativas
- 3.1.4. Análisis de variables cuantitativas

3.2. Características bidimensionales

- 3.2.1. Introducción a los datos 2D
- 3.2.2. Procesado de imagen
- 3.2.3. Segmentación de imágenes
- 3.2.4. Descriptores de forma
- 3.2.5. Descriptores de textura**

3.2.4. Descriptores de forma

- Descriptores de textura – Matriz de coocurrencias

La **matriz de coocurrencias** mide la frecuencia en la que un píxel con nivel de intensidad i es adyacente de un píxel con nivel de intensidad j . La adyacencia se define a través de un **número de píxeles vecinos d** y una **orientación (ángulo, α)**. A partir de estos valores, es posible extraer diferentes características como: el **contraste**, la **correlación**, la **disimilitud**, la **homogeneidad**, la **energía** y la **segundo momento angular**.

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

Imagen

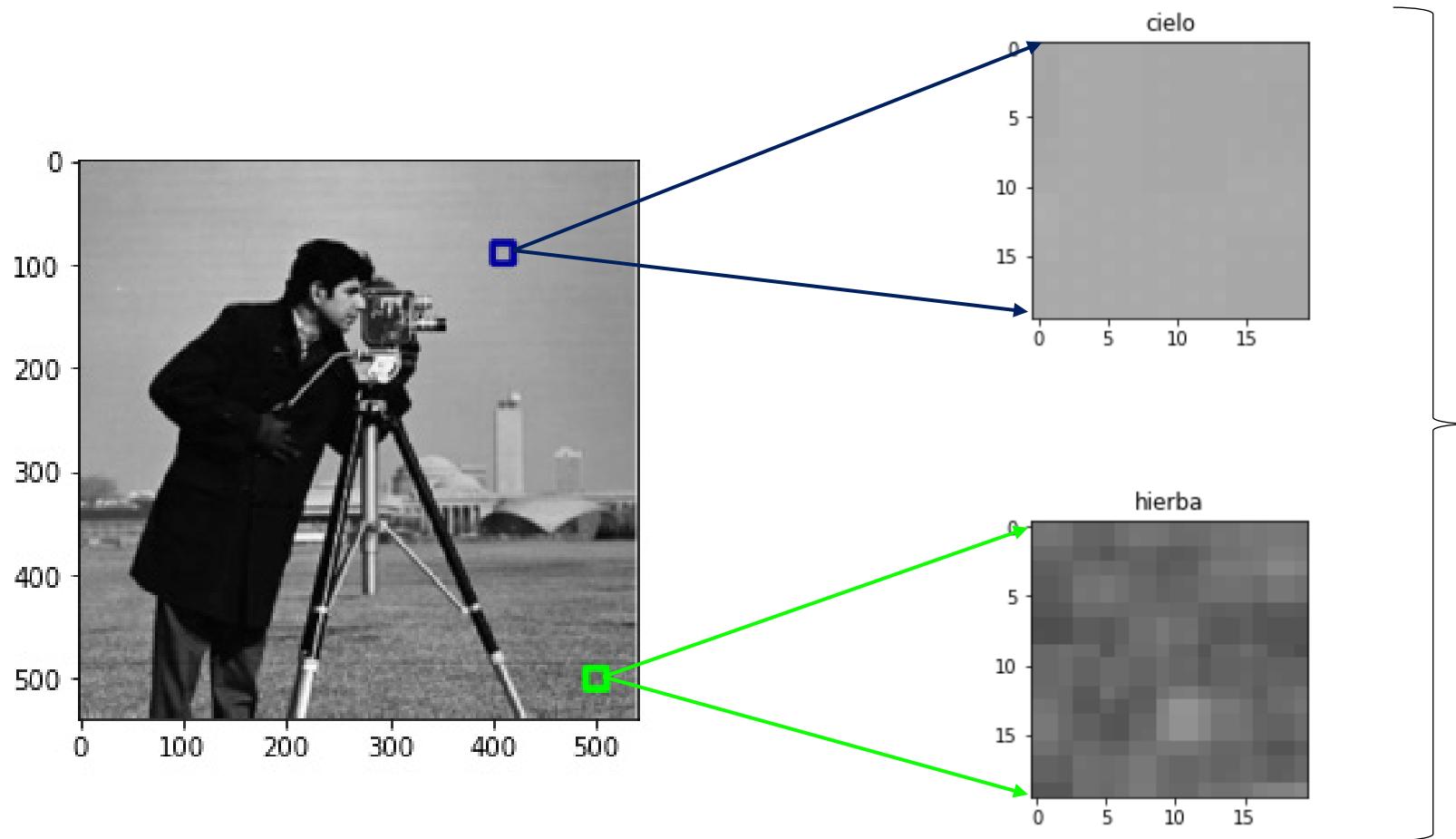
$$\begin{array}{l} \alpha = 0^\circ \\ d = 1 \\ \longrightarrow \end{array}$$

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

Matriz de coocurrencias 8x8

3.2.4. Descriptores de forma

- Descriptores de textura – Matriz de coocurrencias



	cielo	hierba
contraste	3.68	333.393
disimilitud	1.4	13.8533
homogeneidad	0.512471	0.0921279
ASM	0.142717	0.00510556
Energía	0.377779	0.0714532
Correlación	0.190521	-0.0705127

3.2.5. Descriptores de textura

- Descriptores de textura – Matriz de coocurrencias

La **matriz de coocurrencias** mide la frecuencia en la que un píxel con nivel de intensidad i es adyacente de un píxel con nivel de intensidad j . La adyacencia se define a través de un **número de píxeles vecinos d** y una **orientación (ángulo, α)**. A partir de estos valores, es posible extraer diferentes características como: el **contraste**, la **correlación**, la **disimilitud**, la **homogeneidad**, la **energía** y la **segundo momento angular**.

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

Imagen

$$\begin{array}{l} \alpha = 0^\circ \\ d = 1 \\ \longrightarrow \end{array}$$

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

Matriz de coocurrencias 8x8

3.2.5. Descriptores de textura

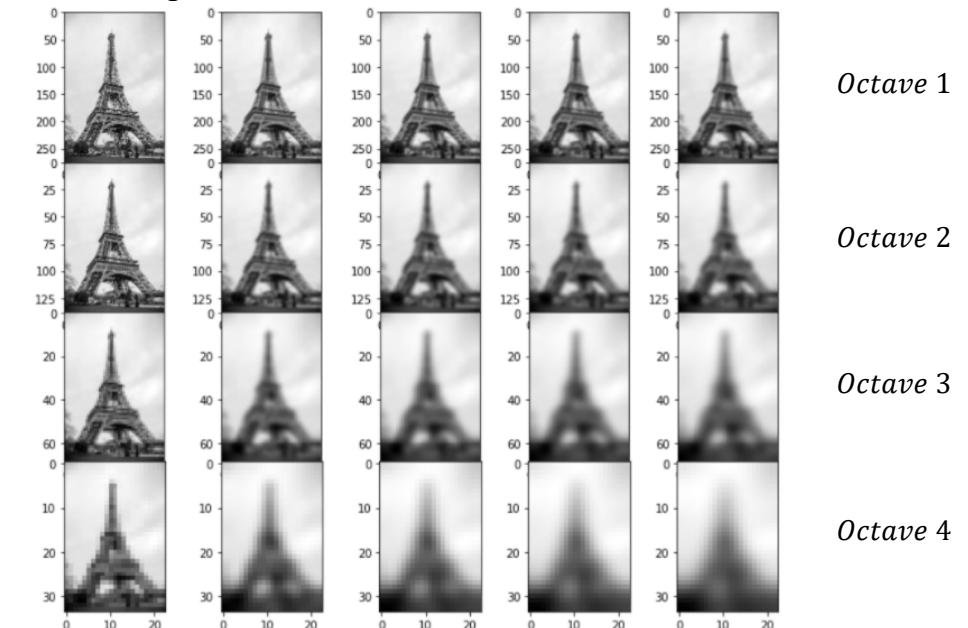
- **Descriptores de textura – Scale-Invariant Feature Transforms (SIFT)**

El **algoritmo SIFT** se utiliza en aplicaciones que involucran un escalado de la imagen. Sirve para la detección de esquinas, bordes o texturas, siendo estable frente a perturbaciones y transformaciones que pueda sufrir la imagen. Es invariante a posición, escala, rotación, iluminación y contraste. El algoritmo consta de las siguientes fases:

1. Detección de extremos en la escala-espacio. La escala-espacio se crea aplicando convoluciones entre la imagen original (I) y un kernel gaussiano (G) con diferentes escalas (σ).

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$
$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x^2+y^2)/2\sigma^2}$$

Todas las imágenes (L) generadas forman una “octave”. Este proceso se repite para obtener varias “octaves”, donde el tamaño de cada una de ellas es la mitad de la anterior.

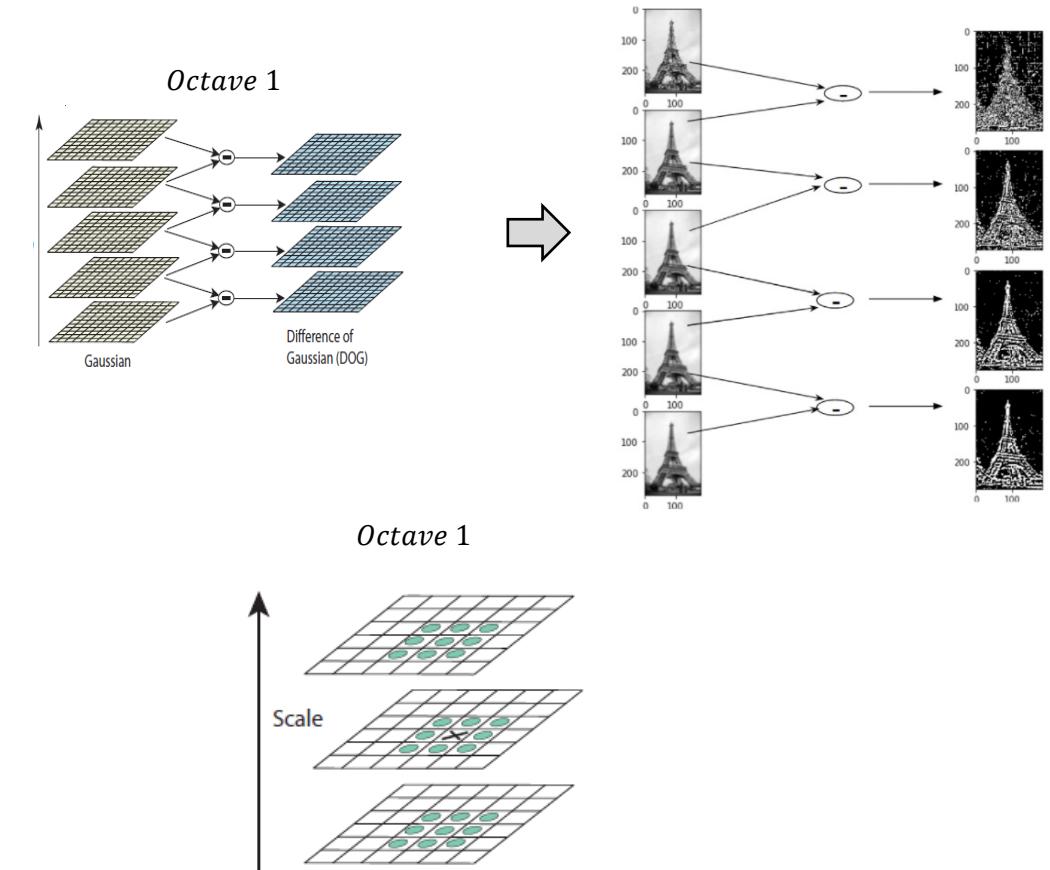


3.2.5. Descriptores de textura

- Descriptores de textura – Scale-Invariant Feature Transforms (SIFT)

2. Diferencia de Gaussianas (DOG). Se genera un set de imágenes basado en la diferencia de las imágenes tras ser convolucionadas con el kernel gaussiano. Estas diferencias de gaussianas son muy buenas para encontrar puntos de interés en las imágenes. El procedimiento se aplica para todas las octavas como la diferencia entre una imagen con dos escalas diferentes, por ejemplo, σ y $k\sigma$.

3. Detección de posibles puntos de interés. Cada píxel de una imagen es comparada con los 8 píxeles vecinos de esa misma imagen, y con los 9 píxeles (en esa misma posición) en las escalas directamente posterior y anterior. Si el valor del píxel es un extremo local (mayor o menor intensidad), ese píxel se convierte en un posible punto de interés.



3.2.5. Descriptores de textura

- **Descriptores de textura – Scale-Invariant Feature Transforms (SIFT)**

4. Localización de puntos de interés. El punto anterior genera muchos posibles puntos de interés, por lo que en este paso hay que quedarse únicamente con los que son útiles. Se aplican series de Tylor (segundo orden) sobre cada posible punto de interés para lidiar con los keypoints de bajo contraste. Si la intensidad es menor que un determinado umbral (0.03), se rechaza. Además, la Diferencia de Gausianas (DoG) presenta una alta respuesta para bordes, por lo que muchos keypoints se detectan cerca de los bordes. No obstante, tenemos que eliminar aquellos que no sean robustos al ruido. Para ello, se utiliza la “Hessian matrix (H)” de 2x2 que permite obtener la curvatura principal.

5. Asignación de la orientación. Una vez ya tenemos localizados los keypoints estables frente a cambios de escala, bajo contraste y ruido, asignamos una orientación a cada punto de interés para hacerlo invariante a las rotaciones. Se selecciona un vecindario alrededor de un punto de interés y se calcula la magnitud y la dirección del gradiente. La orientación se calcula mediante un histograma de 36 bins que cubre los 360 grados. Si por ejemplo la dirección del gradiente en un punto es 57.17 grados, esto corresponde al bin 50-59 del histograma. La cantidad que se añade a dicho bin es la magnitud del gradiente, por ejemplo, 16.64. Tras hacer esto con todos los píxeles del vecindario del punto de interés, el histograma tendrá un pico en cada punto. El valor más alto del histograma y aquellos picos mayores del 80% se tienen en cuenta para calcular la orientación. Esto crea puntos de interés con la misma localización y escala, pero diferentes direcciones.

35	40	41	45	50
40	40	42	46	52
42	46	50	55	55
48	52	56	58	60
56	60	65	70	75

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

$$\Theta = \arctan\left(\frac{G_y}{G_x}\right)$$



$$G_x = 55 - 46 = 9 \\ G_y = 56 - 42 = 14$$

$$Magnitud = G = 16.64$$

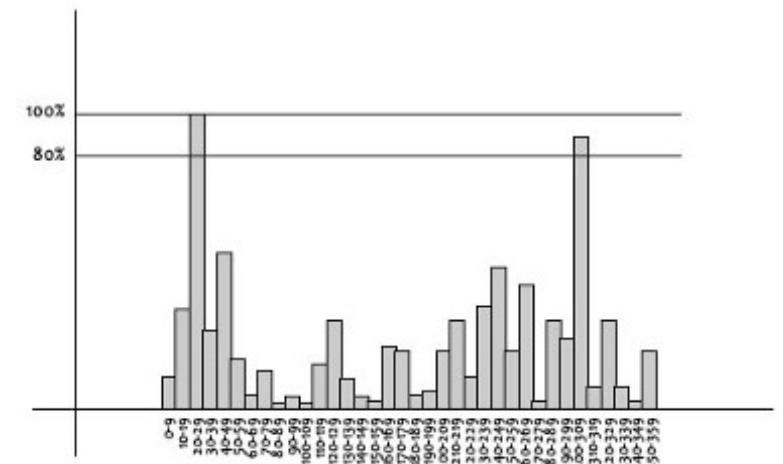
$$Dirección = \Theta = 57.17$$

3.2.5. Descriptores de textura

- **Descriptores de textura – Scale-Invariant Feature Transforms (SIFT)**

4. Localización de puntos de interés. El punto anterior genera muchos posibles puntos de interés, por lo que en este paso hay que quedarse únicamente con los que son útiles. Se aplican series de Tylor (segundo orden) sobre cada posible punto de interés para lidiar con los keypoints de bajo contraste. Si la intensidad es menor que un determinado umbral (0.03), se rechaza. Además, la Diferencia de Gausianas (DoG) presenta una alta respuesta para bordes, por lo que muchos keypoints se detectan cerca de los bordes. No obstante, tenemos que eliminar aquellos que no sean robustos al ruido. Para ello, se utiliza la “Hessian matrix (H)” de 2x2 que permite obtener la curvatura principal.

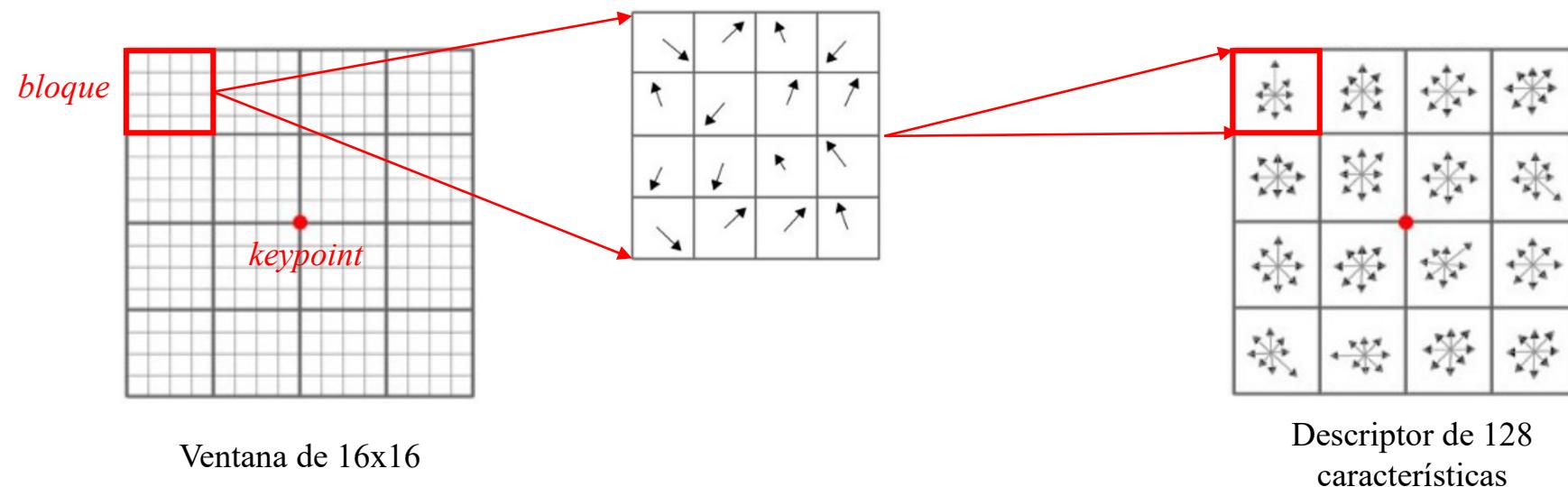
5. Asignación de la orientación. Una vez ya tenemos localizados los keypoints estables frente a cambios de escala, bajo contraste y ruido, asignamos una orientación a cada punto de interés para hacerlo invariante a las rotaciones. Se selecciona un vecindario alrededor de un punto de interés y se calcula la magnitud y la dirección del gradiente. La **orientación** se calcula mediante un histograma de 36 bins que cubre los 360 grados. Si por ejemplo la dirección del gradiente en un punto es 57.17 grados, esto corresponde al bin 50-59 del histograma. La cantidad que se añade a dicho bin es la **magnitud** del gradiente, por ejemplo, 16.64. Tras hacer esto con todos los píxeles del vecindario del punto de interés, el histograma tendrá un pico en cada punto. El valor más alto del histograma y aquellos picos mayores del 80% se tienen en cuenta para calcular la orientación. Esto crea puntos de interés con la misma localización y escala, pero diferentes direcciones.



3.2.5. Descriptores de textura

- Descriptores de textura – Scale-Invariant Feature Transforms (SIFT)

6. Descriptor del punto de interés. Una vez cada punto de interés tiene su localización, escala y orientación, es momento de calcular el descriptor que permite identificar un mismo punto en una imagen transformada a partir de la original. Para ello, se coge una ventana de 16x16 alrededor del punto de interés. Dicha ventana se divide en bloques de 4x4 y para cada bloque, se calculan la magnitud y las orientaciones de los píxeles obteniendo un histograma de orientación de 8 bins. De esta manera, se obtienen 16 histogramas de 8 bins, dando lugar a un fingerprint de 128 características.



3.2.5. Descriptores de textura

- **Descriptores de textura – Speeded-Up Robust Features (SURF)**

El **descriptor SURF** es una versión mejorada del algoritmo SIFT para la detección y descripción de puntos de interés. Se puede utilizar para el reconocimiento o el tracking de objetos, para reconstrucción 3D, segmentación basada en movimiento, etc. La principal ventaja es que es hasta 3 veces más rápido que el algoritmo SIFT y esto permite ser aplicado en problemas que requieren solución en tiempo real.

El algoritmo consta de las siguientes etapas:

1. **Cálculo de la matriz Hessiana.** A diferencia de SIFT, que se basaba en el cálculo de diferencia de gaussianas (DoG), este algoritmo se basa en el cálculo de la Hessiana H . Esta matriz representa la derivada segunda para cada dirección en cada punto (horizontal, vertical y diagonal). Se calcula el determinante de H que expresa el cambio local alrededor de un punto x .

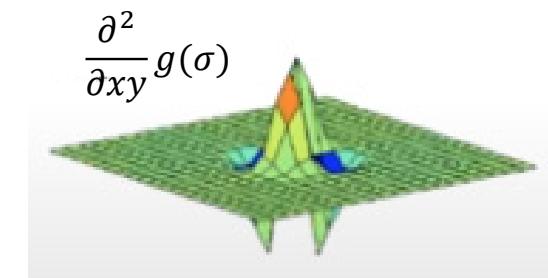
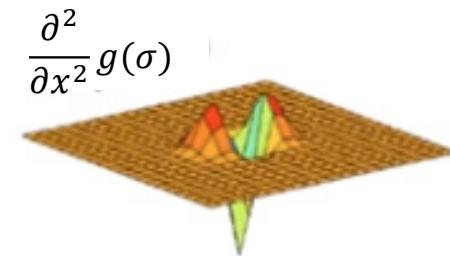
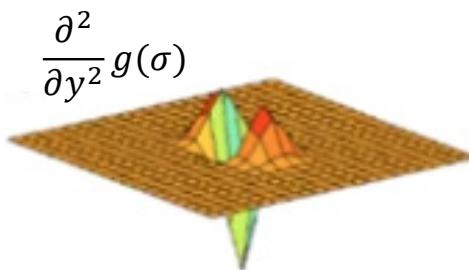
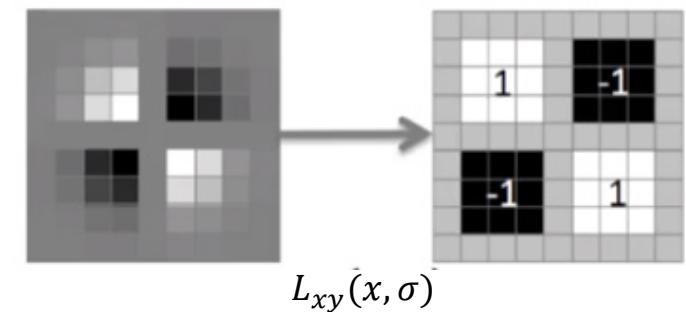
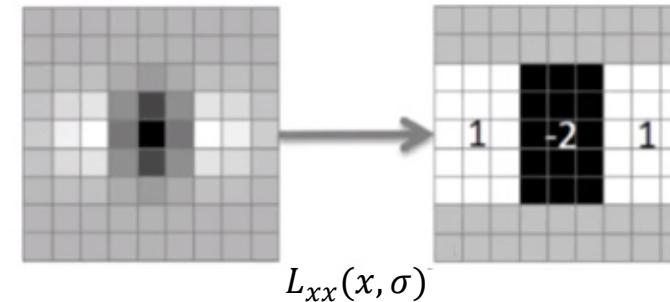
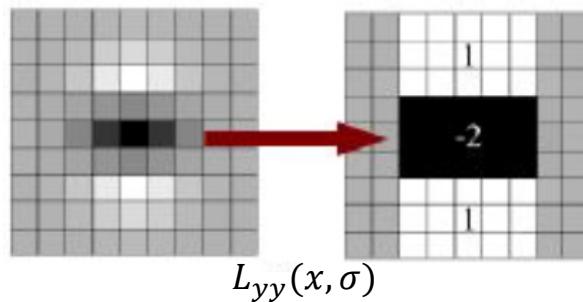
$$H(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix}$$

A diferencia de SIFT, donde se genera una pirámide gaussiana de imágenes (imágenes cada vez más pequeñas, en SURF se genera una **pirámide gaussiana de filtros**). En cada escala se duplica el tamaño del filtro (horizontal y vertical). Finalmente, se calculan los máximos locales del determinante a diferentes escalas.

3.2.5. Descriptores de textura

- **Descriptores de textura – Speeded-Up Robust Features (SURF)**

Lo que se hace es **aproximar los filtros a una estructura más simple** mediante el cálculo de la Hessiana para acelerar el proceso.



3.2.5. Descriptores de textura

- Descriptores de textura – Speeded-Up Robust Features (SURF)

Se calcula la **imagen integral** $O(x)$ a partir de la imagen original. Cada nuevo píxel de la imagen integral corresponde a la suma de todos los píxeles previos por filas y por columnas. Aunque el cálculo de la matriz integral pueda ser un poco costoso, solo hay que calcularla una vez, ya que las dimensiones de la imagen no van a cambiar. De esta manera, el cálculo de las máscaras (recordemos que en SURF lo que varía es la máscara de filtros) se obtiene muy rápido a partir de sumas y restas de valores pre-calculados.

1	1	1
1	1	1
1	1	1

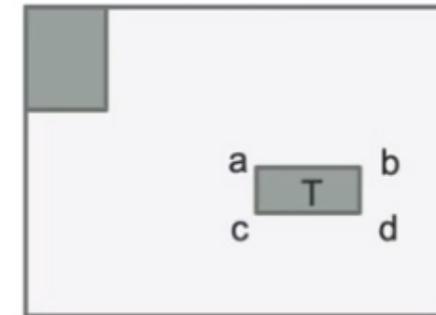
$$O(x,y) = \sum_{i=0}^x \sum_{j=0}^y I(i,j)$$



1	2	3
2	4	6
3	6	9

Imagen original

Imagen integral



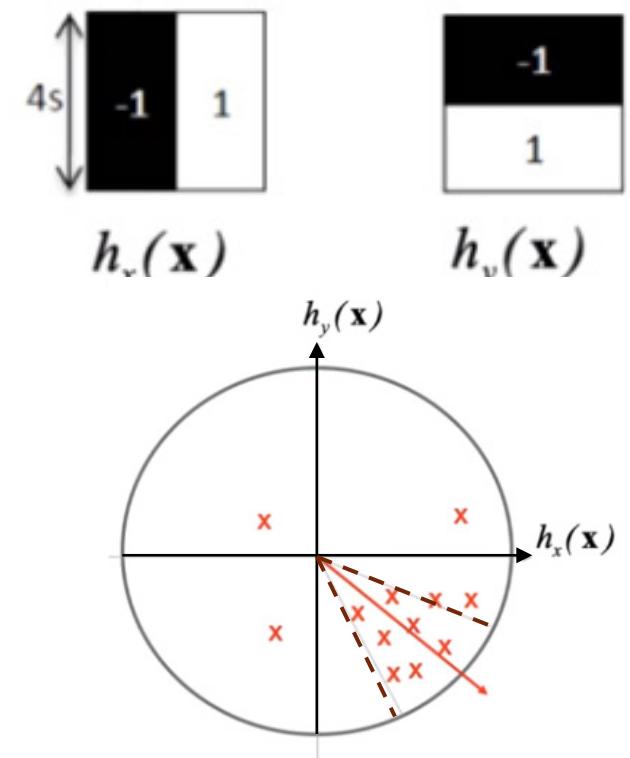
$$T = O(\mathbf{d}) - O(\mathbf{c}) - O(\mathbf{b}) + O(\mathbf{a})$$

3.2.5. Descriptores de textura

- **Descriptores de textura – Speeded-Up Robust Features (SURF)**

2. Cálculo de la orientación. Una vez localizados los puntos de interés mediante el cálculo de la matriz Hessiana, se calculan las derivadas horizontales y verticales sobre la imagen original. Pero, en este caso, aprovechando que se tiene calculada la imagen integral, se realizan derivadas por bloque (wavelet). Estas derivadas se basan en sumar o restar un determinado valor a los píxeles que aparecen en la ventana. El tamaño de la máscara se ajusta en función de la escala (s). En la escala original la máscara sería 4×4 píxeles.

Los cálculos se realizan para todos los píxeles de la ventana de vecindad alrededor del punto de interés para evitar el ruido. Esto da lugar a un valor de orientación de gradiente para cada píxel de la ventana. Estos valores se representan en un diagrama circular que se recorre en arcos de 60° para encontrar el régión donde hay más cantidad de puntos. Dichos puntos corresponde a la dirección del gradiente de cada píxel de la ventana de vecindad.



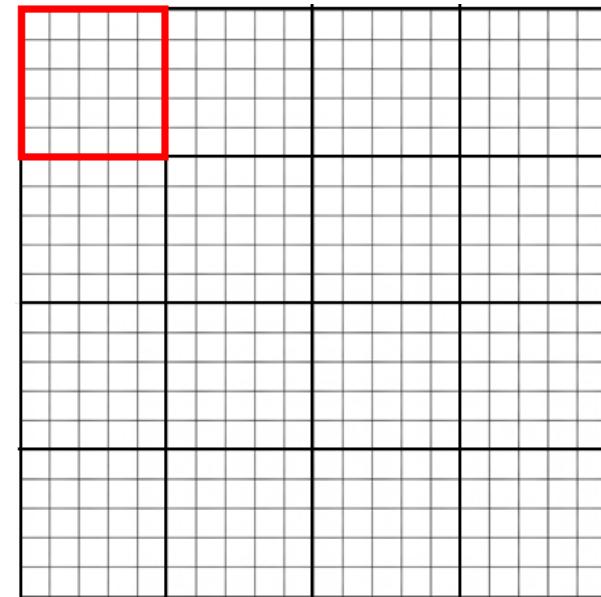
3.2.5. Descriptores de textura

- **Descriptores de textura – Speeded-Up Robust Features (SURF)**

3. Cálculo del descriptor SURF. Se selecciona una ventana de 20x20 píxeles alrededor del punto de interés. A partir de dicha ventana, se obtienen 16 subregiones de 5x5 píxeles cada una. Para cada subregión se calcula la derivada wavelet y se extraen 4 características en un vector v . De esta forma, para cada ventana de 20x20 píxeles se obtienen $4 \times 16 = 64$ características que describen el punto de interés.

$$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|).$$

También es posible extender la versión del algoritmo SURF a 128 características para aumentar el nivel de distinción. Las sumas de d_x y $|d_x|$ se computan por separado para $d_y < 0$ y $d_y \geq 0$. Lo mismo para d_y y $|d_y|$. De esta manera, se duplican el número de características por subregión, dando lugar a $8 \times 16 = 128$ características.



3.2.5. Descriptores de textura

- **Descriptores de textura – Histogram of oriented gradients (HOG)**

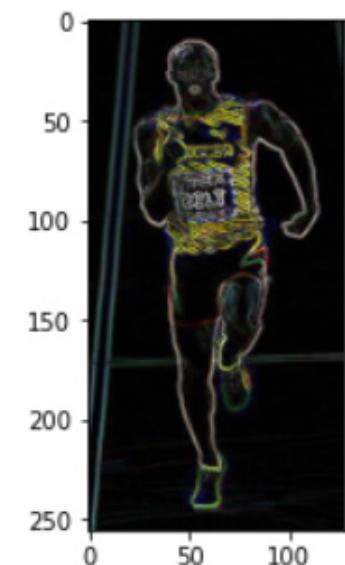
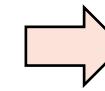
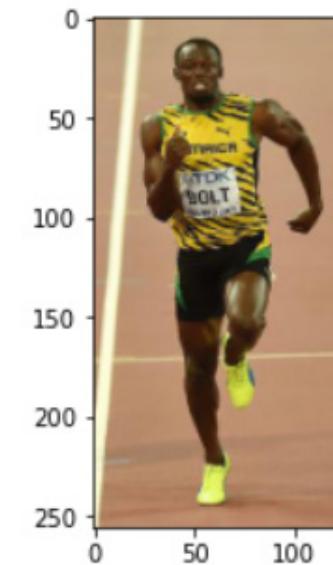
El **histograma de gradientes** utiliza la distribución de la dirección de los gradientes como características, ya que la magnitud del gradiente presenta una alta respuesta a los cambios de intensidad elevados. Esto permite obtener información sobre la forma de los objetos.

El algoritmo consta de los siguientes pasos:

1. Cálculo de los gradientes de la imagen. Se calculan los gradientes horizontales y verticales mediante el filtrado de la imagen con un kernel de tamaño 1. Se puede aplicar el algoritmo de Sobel visto anteriormente. Una vez se han obtenido las matrices G_x y G_y que representan las aproximaciones de las derivadas de intensidades, se calcula la magnitud y la dirección del gradiente.

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

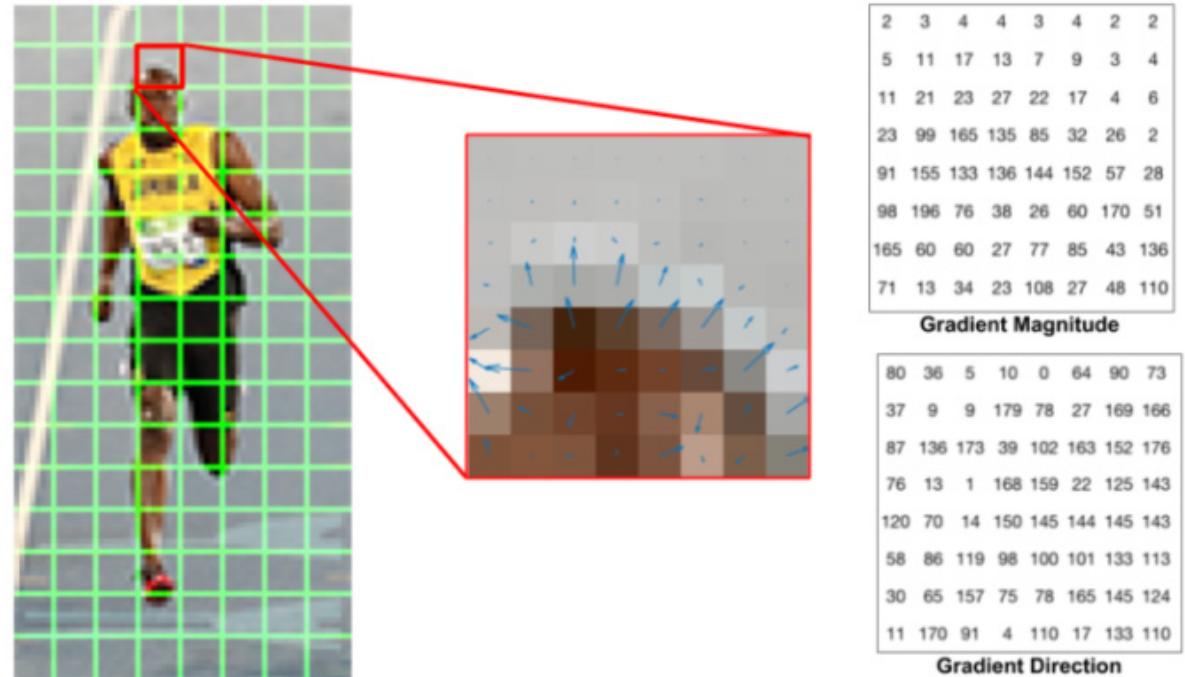
$$\Theta = \arctan\left(\frac{G_y}{G_x}\right)$$



3.2.5. Descriptores de textura

- **Descriptores de textura – Histogram of oriented gradients (HOG)**

2. Cálculo de gradientes en celdas de 8x8. El hecho de escoger celdas de 8x8 píxeles es porque cada píxel tendrá 2 valores (magnitud y dirección), lo que da lugar a un total de 128 números. Estos 128 valores se pueden representar en un histograma de 9 bins correspondiente a los ángulos 0, 20, 40, 60, 80, 100, 120, 140 y 160. Estas 9 características dan lugar a una representación óptima que no solo es la más compacta, sino que también es robusta frente al ruido. Como se puede observar, la dirección del gradiente se evalúa entre 0 y 180 grados, en lugar de 0 y 360. Esto se denomina “unsigned” gradients porque la dirección del gradiente se representa en valor absoluto.



3.2.5. Descriptores de textura

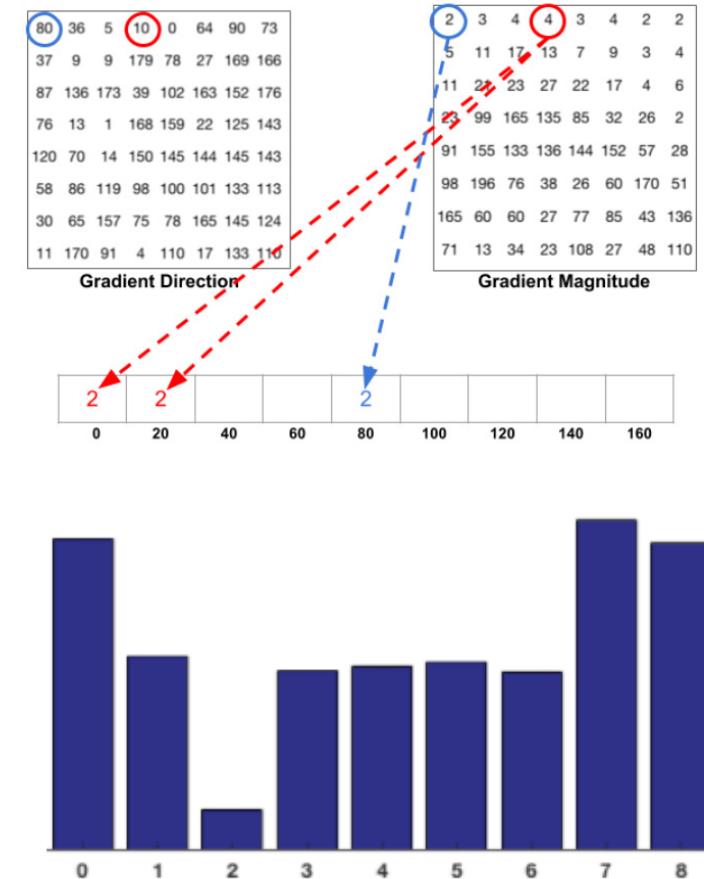
- **Descriptores de textura – Histogram of oriented gradients (HOG)**

3. Cálculo del histograma de gradientes de 9-bins. Una vez calculadas la magnitud y la dirección de los gradientes de cada píxel en la celda de 8x8, se construye un histograma de 9-bins donde cada bin representa una dirección y el valor de dicho bin viene determinado por la magnitud.

- Si la dirección coincide con la etiqueta de un bin específico, entonces se añade íntegramente el valor de su magnitud a dicho bin (azul).
- Si la dirección toma valores entre dos bins, la magnitud correspondiente se añadirá a los dos bins de manera proporcional (rojo).

Nota: En el caso de que la dirección sea mayor de 180° , la magnitud se añadirá de manera proporcional entre los bins de 160 y 0, ya que 0 equivale a 180° .

Finalmente, se crea el histograma de gradientes con las contribuciones de todos los píxeles de la celda 8x8.



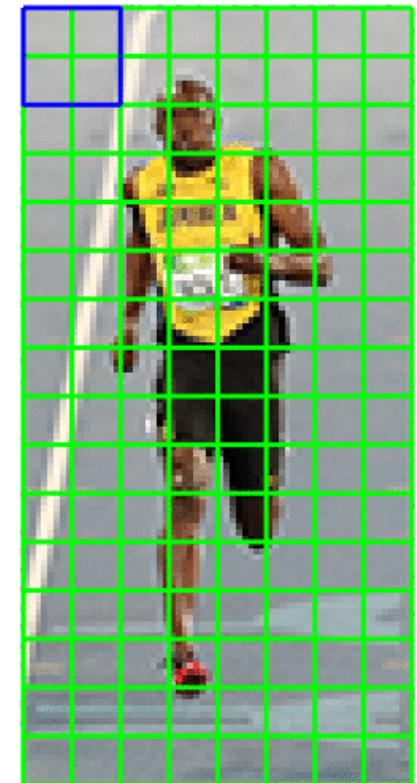
3.2.5. Descriptores de textura

- **Descriptores de textura – Histogram of oriented gradients (HOG)**

4. **Normalización del histograma de gradientes.** El objetivo es que el descriptor sea invariante frente a variaciones en la intensidad de los píxeles, es decir, invariante a los cambios de contraste y/o luminosidad. Para ello, se recorre la imagen en tamaños de ventana de 16x16, es decir, en cada paso se cogerán 4 celdas de 8x8. De esta manera, se obtendrán 4 histogramas de 9 bins que dan lugar a un vector de características de 36 elementos. Estos elementos se normalizan de acuerdo a una regla (L1 o L2, por ejemplo). Después, la ventana se desplaza una celda (8 píxeles) y se vuelven a calcular y normalizar los 36 elementos.

5. **Cálculo del histogram of oriented gradients (HOG).** En nuestro caso particular, el tamaño de la imagen era de 128x64 píxeles. Por tanto, en el eje horizontal hay 7 celdas de 16x16 píxeles (con solapamiento de celdas como se muestra en la animación) y en el eje vertical, 15 celdas.

Puesto que en cada celda de 16x16 píxeles obteníamos un vector de características con 36 elementos, el descriptor final tendrá un total de $7 \times 15 \times 36 = 3780$ características que son representativas de la imagen bajo estudio.



3.2.5. Descriptores de textura

- Descriptores de textura – Local Binary Patterns (LBP)

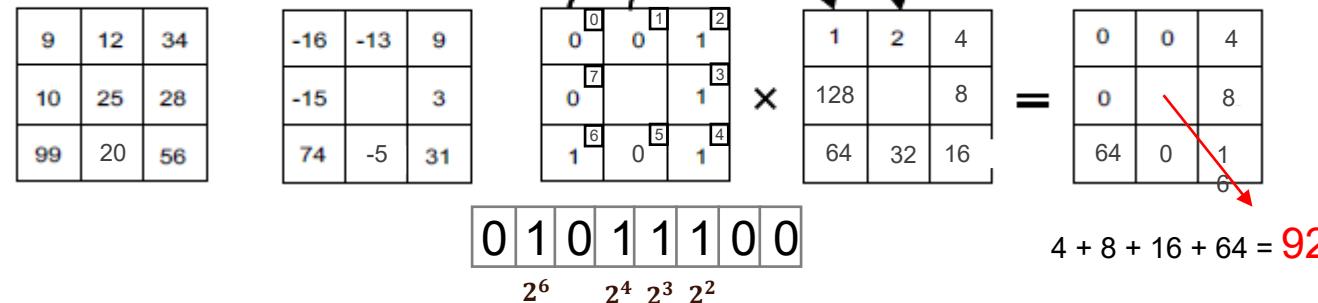
Los **Local Binary Patterns (LBP)** consisten en un operador que describe patrones de la imagen a nivel local proporcionando buenos resultados en aplicaciones de reconocimiento de texturas y en identificación de objetos y formas. El algoritmo para extraer los LBP consiste en un protocolo de ventana deslizante donde el valor del píxel central se modifica a partir del valor de intensidad de sus píxeles vecinos.

Paso 1. Se calcula la diferencia de intensidad entre los píxeles vecinos y el píxel central.

Paso 2. Se construye una matriz auxiliar a partir de la umbralización de las diferencias con 0s en los valores negativos y 1s en los positivos.

Paso 3. La matriz construida da lugar a una cadena binaria que se multiplica por una matriz de referencia.

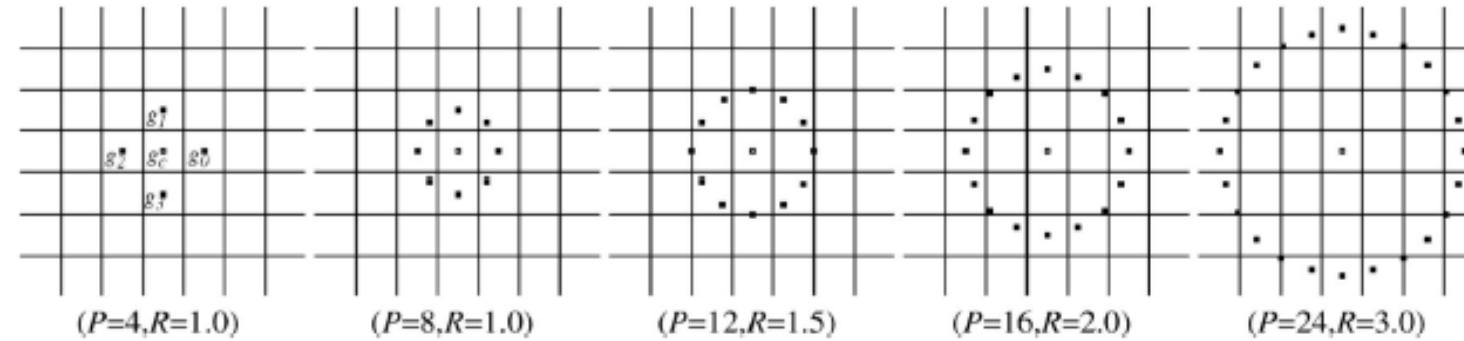
Paso 4. Se suman todos los resultados para obtener el LBP del píxel central.



3.2.5. Descriptores de textura

- **Descriptores de textura – Local Binary Patterns (LBP)**

Para hacer el descriptor **invariante a transformaciones de intensidad**, aparecen los $LBP_{P,R}$, donde P es el número de píxeles vecinos alrededor del píxel de interés y R , es el radio de dicha frecuencia. P determina el espacio angular y R la resolución espacial.



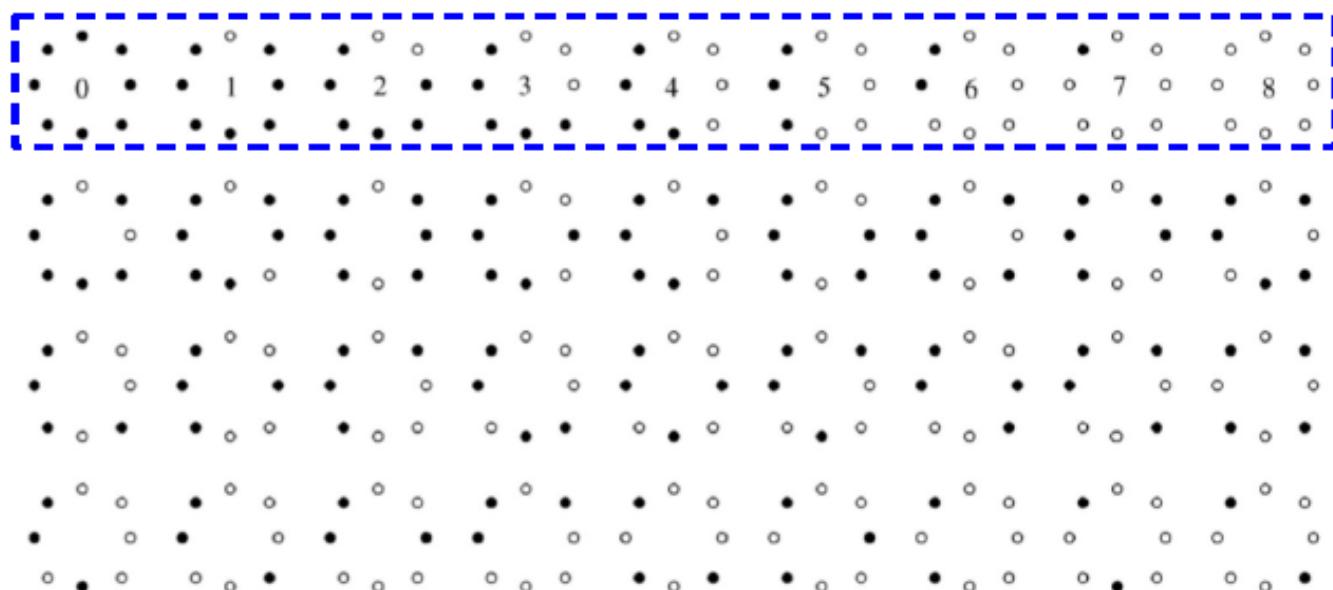
Sin embargo, el descriptor en este punto es sensible a transformaciones de rotación. Para eliminar ese efecto, se define el operador $LBP_{P,R}^{ri}$ que asigna un identificador único a todas aquellas cadenas binarias que presentan el mismo número de transiciones. Se llama transición a los cambios de 0 a 1, y viceversa, que aparecen en una cadena.

3.2.5. Descriptores de textura

- Descriptores de textura – Local Binary Patterns (LBP)

Por ejemplo, dadas dos cadenas binarias diferentes (11100001 y 01111000), cuyo número de transiciones es el mismo, se asigna un único identificador normalizado poniendo todos los 0s a la izquierda y los 1s a la derecha. De esta manera, ambas cadenas quedarían representadas por la cadena 00001111, donde se observan 2 transiciones (una de 0 a 1 y otra de 1 a 0).

Así pues, siendo $P=8$ el número de píxeles vecinos, los LBP normales podían tener hasta $2^8 = 256$ posibles valores de patrones diferentes, mientras que el nuevo descriptor invariante a la rotación $LBP_{P,R}^{ri}$ presenta únicamente $(P+1)*4=36$ valores distintos.



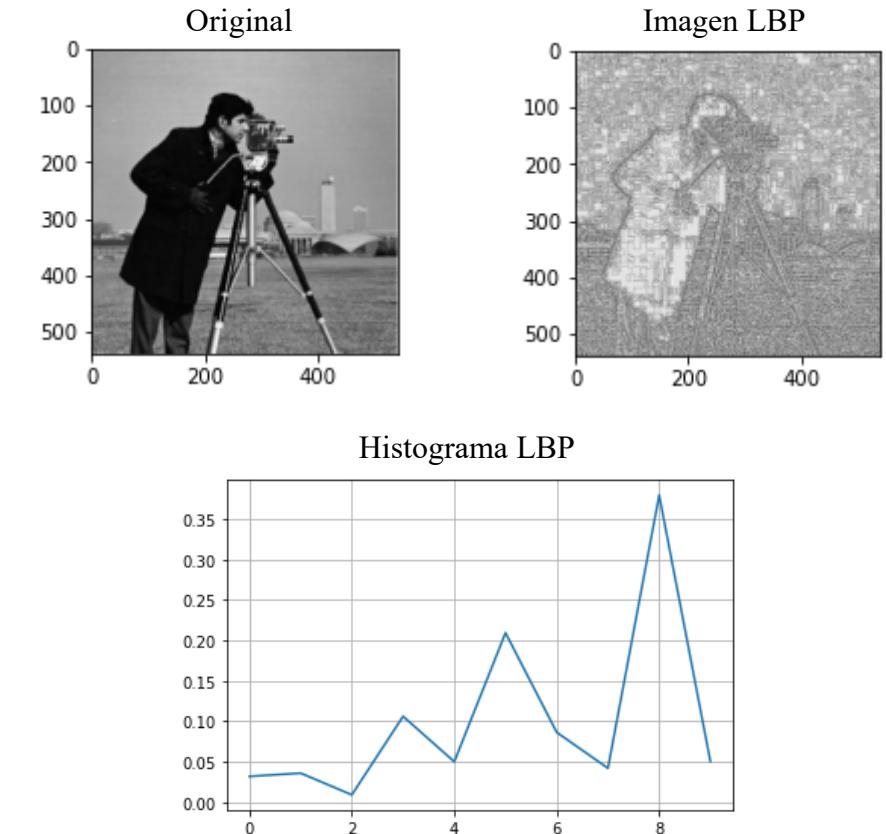
3.2.5. Descriptores de textura

- **Descriptores de textura – Local Binary Patterns (LBP)**

Más adelante se dedujo que muchos de esos 36 patrones presentaban información redundante, así que se introdujo el concepto de **uniformidad** dando lugar al descriptor $LBP_{P,R}^{riu^2}$ → **LBP uniformemente invariante a rotación.**

Se demostró que los patrones con un número de transiciones menor o igual a 2 tenían una capacidad discriminativa mucho mayor que el resto, por lo que se definieron dichos patrones como **LBP uniformes**. El número total de LBPs uniformes es $P+1$. En el caso anterior, donde $P=8$, observamos que hay 9 LBP uniformes (cuadro azul), ya que dichos patrones presentan un número de transiciones menor o igual a 2. El resto de los LBP no uniformes (los 27 restantes) se agrupan todos en un único conjunto “no uniforme”.

De esta manera, el descriptor **LBP uniformemente invariante a rotación** $LBP_{P,R}^{riu^2}$ presenta $P+2$ posibles valores: de 0 a 8 para los patrones uniformes y la etiqueta adicional para los patrones no uniformes





viu

**Universidad
Internacional
de Valencia**

universidadviu.com

De:
 Planeta Formación y Universidades