

# **VISIÓN ARTIFICIAL**

Temario de la asignatura Visión Artificial, correspondiente al primer cuatrimestre del Máster en Inteligencia Artificial de la Universidad Internacional de La Rioja.

**MÁSTER EN  
INTELIGENCIA  
ARTIFICIAL**

## TABLA DE CONTENIDO

Tema 1. Introducción a los sistemas de percepción .....	4
1.1. Sistema auditivo .....	4
1.2. Sistema visual .....	6
Tema 2. Elementos de un sistema de percepción .....	7
2.1. Captura de información .....	7
2.2. Procesamiento de la información.....	9
2.3. Toma de decisión.....	9
Tema 3. Captura y digitalización de señales .....	10
3.1. Sistemas de conversión analógico discreto .....	10
3.2. Muestreo y cuantificación .....	10
Tema 4. Fuentes y tipos de ruidos.....	13
4.1. Entropía .....	13
4.2. Fuentes perturbadoras de las señales .....	16
Tema 5. Detección y cancelación de anomalías .....	18
5.1. Métodos de identificación de anomalías .....	19
5.2. Eliminación de anomalías .....	20
Tema 6. Procesamiento de imagen. Operaciones elementales.....	23
6.1. Ajuste de intensidad.....	24
Tema 7. Procesamiento de imagen. Operaciones espaciales .....	29
7.1. Filtro paso bajo .....	30
7.2. Filtro paso alto.....	30
Tema 8. Procesamiento de señales. Filtrado y análisis de frecuencia.....	34
8.1. Series de Fourier .....	34
8.2. Transformada de Fourier .....	36
8.3. Transformada Discreta de Fourier .....	37
8.4. Transformada Rápida de Fourier .....	39
Tema 9. Procesamiento de imagen. Morfología matemática .....	40
9.1. Elemento estructural .....	40
9.2. Dilatación y erosión .....	41
9.3. Apertura y clausura .....	44

9.4. Gradiente morfológico .....	45
9.5. Top Hat .....	46
Tema 10. Procesamiento de imagen. Crecimiento de regiones .....	48
10.1. Segmentación por umbralización .....	49
10.2. Segmentación por crecimiento de regiones .....	52
Tema 11. Extracción de características. Propiedades estadísticas y frecuencias de la señal .....	59
11.1. Análisis en el dominio natural .....	61
11.2. Análisis en el dominio frecuencial .....	63
Tema 12. Extracción de características. Caracterización de textura en imágenes .....	65
12.1. Matriz de coocurrencia de Haralick (GCLM) .....	66
12.2. Patrones binarios locales (LBP) .....	68
12.3. Transformadas unitarias .....	72
Tema 13. Extracción de características. Procesamientos multiescala y métodos avanzados .....	74
13.1. Short Time Fourier Transform (STFT) .....	74
13.2. Wavelets .....	77
13.3. Filtros de Gabor .....	80
13.4. Scale Invariant Feature Transform (SIFT) .....	81
Tema 14. Decisión. Principios e implementación de algoritmos de ayuda en la toma de decisiones .....	83
14.1. Aprendizaje supervisado .....	84
14.2. Aprendizaje no supervisado .....	84
14.3. Equilibrio entre sesgo y varianza .....	84
14.4. Curse of dimensionality .....	85
Tema 15. Aplicaciones actuales del tratamiento de la señal .....	86
15.1. Biometría .....	86
15.2. Coches autónomos .....	90
15.3. Retos y limitaciones de la percepción computacional .....	91

## TEMA 1. INTRODUCCIÓN A LOS SISTEMAS DE PERCEPCIÓN

Los **sistemas de percepción** del ser humano que sirven como referencia para la implementación de técnicas artificiales.

### 1.1. SISTEMA AUDITIVO

El **sonido** se define como una **perturbación mecánica del medio** por el que se propaga en forma de onda. El sonido resulta de la vibración de ida y vuelta de las partículas del medio a través del cual se mueve dicha onda.

La **caracterización física** del sonido permite su medición y evaluación cuantitativa ante diferentes fenómenos:

- La **intensidad acústica de una onda sonora** viene dada por la rapidez promedio del flujo de energía acústica a través de un área unitaria, midiéndose en  $W/m^2$ .
- Los **niveles sonoros** son la medida de la energía sonora en dB. Se define el nivel de intensidad percibido ( $L$ ) a partir de la **Ley de Weber-Fechner** que establece una relación cuantitativa entre la magnitud de un estímulo físico y cómo es percibido. En relación con el sonido, esta ley explica cómo nuestra percepción del sonido y otros estímulos sensoriales sigue una escala logarítmica en lugar de una escala lineal:

$$L = 10 \log_{10} \left( \frac{I}{I_0} \right)$$

$$I = I_0 \cdot 10^{\frac{L}{10}}$$

- $L$  es el nivel de intensidad percibido en la escala de decibelios (dB), que es una escala logarítmica. Es la forma en que percibimos la intensidad del estímulo.
- $I$  es la intensidad del estímulo, es decir, la medida real de la intensidad en unidades físicas, como  $W/m^2$ , que sigue una escala lineal.
- $I_0$  es la intensidad de referencia que representa el umbral de audibilidad a 1 KHz en el aire libre y viene dada por:

$$I_0 = 10^{-12} W/m^2$$

Por tanto, para que percibir un cambio en la intensidad del sonido, la magnitud del estímulo debe aumentar de manera proporcional en lugar de sumarse de forma constante. Si la intensidad del sonido se incrementa en una cantidad fija, la diferencia percibida será cada vez menor. En cambio, para notar un cambio significativo, es necesario que la intensidad aumente en proporciones crecientes.

El **oído** es el órgano encargado de la transmisión de sonidos hasta el cerebro. El **oído humano** es capaz de **captar ondas sonoras** en el rango entre los **20 Hz y los 20 KHz de frecuencia**. Las ondas por debajo y por encima de estos límites se denominan **infrasonidos** y **ultrasonidos**, respectivamente.

El oído está caracterizado por diferentes **umbrales de funcionamiento**:

- **Umbral de audibilidad:** es el nivel de intensidad mínimo en espacio libre que el oído es capaz de oír. La máxima sensibilidad se tiene a 4 KHz.
- **Umbral de sensación:** se encuentra a 120 dB, respecto al umbral de referencia. A este nivel de intensidad se produce cierto cosquilleo en el oído. A partir de 140 dB se puede llegar a producir dolor.

La **voz humana** se caracteriza por un rango frecuencial entre 300 Hz y 3.4 KHz.

### Ejercicio sobre la Ley de Weber-Fechner en el sonido:

**Dos fuentes con una intensidad de 60 dB cada una. No se tiene en cuenta la separación entre ambas fuentes. ¿Cuál es el nivel de intensidad combinada?**

Primero, se convierten los niveles de intensidad ( $L_1, L_2$ ) de las fuentes que están en db (escala logarítmica) a  $W/m^2$  (escala lineal) para poder sumarlas linealmente.

$$I_1 = I_0 \cdot 10^{\frac{L_1}{10}} = 10^{-12} \cdot 10^{\frac{60}{10}} = 10^{-6} W/m^2$$

$$I_2 = I_0 \cdot 10^{\frac{L_2}{10}} = 10^{-12} \cdot 10^{\frac{60}{10}} = 10^{-6} W/m^2$$

$$I_{total} = I_1 + I_2 = 2 \cdot 10^{-6} W/m^2$$

Posteriormente se aplica la Ley de Weber-Fechner para calcular el nivel de intensidad combinada.

$$L_{total} = 10 \log_{10} \left( \frac{I_{total}}{I_0} \right) = 10 \log_{10} \left( \frac{2 \cdot 10^{-6}}{10^{-12}} \right) = 63.01 \text{ dB}$$

Esto indica que el nivel de intensidad del sonido (dB) crece en 3.01 dB cuando se dobla la intensidad lineal del sonido.

---

## 1.2. SISTEMA VISUAL

La **visión** es el fenómeno resultante de la percepción del color, forma y distancia de los objetos en un espacio. La visión se produce como resultado de la incidencia de la luz, caracterizada como una onda electromagnética, sobre la retina del ojo.

El **ojo humano**, similar a una cámara, capta la luz reflejada en los objetos. La **retina** es una membrana interior y en ella se encuentran las **células fotosensibles**:

- **Conos:** son muy poco sensibles a la luz. Son las células encargadas de la visión diurna. Actúan como filtros que responden a la luz incidente, capturando diferentes longitudes de onda. Estos se dividen en tres tipos: *Long* (L), *Medium* (M) y *Short* (S), que corresponden aproximadamente a los colores rojo, verde y azul, respectivamente.
- **Bastones:** son más sensibles a la luz. Son las células encargadas de la visión nocturna. Dado que solo existe un tipo de bastón, estos no son sensibles al color.

El concepto de **Just Noticeable Difference (JND)** o diferencia apenas apreciable se refiere al cambio mínimo en la intensidad de un estímulo ( $\Delta I$ ) necesario para que sea percibido. En términos visuales, es la mínima variación de luz que se necesita para notar la diferencia entre dos áreas de diferentes intensidades.

La **Ley de Weber** establece que el cambio necesario para percibir una diferencia ( $\Delta I$ ) es proporcional a la intensidad original del estímulo ( $I$ ). Es decir, cuanto mayor es la intensidad inicial, mayor debe ser el cambio para que sea perceptible.

Esto implica que los cambios de intensidad son más fáciles de percibir en áreas oscuras (bajas intensidades) que en áreas brillantes (altas intensidades), ya que la JND aumenta conforme la intensidad lo hace.

Al igual que sucede con el sonido, **la percepción del estímulo sigue una curva logarítmica**. Es decir, para percibir un cambio similar en la intensidad de la luz, es necesario un mayor incremento de la intensidad en áreas brillantes comparado con las oscuras. Matemáticamente:

$$\Delta I = I_0 \cdot 10^{\frac{L}{10}}$$

## TEMA 2. ELEMENTOS DE UN SISTEMA DE PERCEPCIÓN

Los **sistemas de percepción** buscan emular los procesos de la naturaleza. Por lo tanto, resulta lógico considerar ciertas simplificaciones al modelarlos de manera adecuada. En este contexto, los seres vivos realizan tres funciones de percepción fundamentales:

- **Captura de información:** este proceso recoge estímulos del entorno (movimientos, temperatura, amenazas, etc.) mediante señales físicas, mecánicas o químicas.
- **Procesamiento:** elimina los datos innecesarios, combina fuentes y ajusta la captación para evitar errores. Debe ser tan rápido como la captura para garantizar precisión, evitando sobrecarga o confusión.
- **Toma de decisión y aprendizaje:** la percepción del exterior busca optimizar la toma de decisiones, lo que mejora el aprendizaje al asociar estímulos con consecuencias almacenadas para futuras optimizaciones.

---

### 2.1. CAPTURA DE INFORMACIÓN

Existen numerosos parámetros que definen cómo es dicha la captura y puede discutirse cuáles son los más importantes. Sin embargo, los más esenciales son los siguientes:

- **Especificidad:** la capacidad de un sistema de captura de información para recoger fielmente los eventos sucedidos es clave.
- **Precisión:** es el error de medida de un determinado evento.
- **Sensibilidad:** es la capacidad que tiene un determinado sensor de recoger las fluctuaciones o cambios del evento que se mide.
- **Consumo y tamaño:** es necesario considerar el consumo y el tamaño en el diseño de los sistemas de percepción computacional pues suele ser cierto que consumo y tamaño son inversamente proporcionales a sensibilidad y precisión.

Es importante indicar que sensibilidad y precisión van de la mano, pero la precisión está más orientada a ver cómo difiere la realidad de lo que se ha medido, y la sensibilidad está más orientada a evaluar cómo se adapta a los cambios de lo que se mide.

---

### 2.1.1. Conversión analógico-digital (A/D)

La **conversión analógico-digital** es uno de los pasos esenciales para poder procesar cualquier información. Lo que se hace es **tomar fotografías de la realidad que se está midiendo**, convirtiendo las intensidades físicas a dígitos binarios.

Una de las principales ventajas de esta conversión es la información almacenada es infinitamente menor, lo que **ahorra espacio de almacenamiento**. Además, **permite y facilita la compresión de información**, puesto que la información ha sido convertida a formato digital. Sin embargo, la principal desventaja es la **pérdida de información**, ya que la señal se está muestreando.

---

### 2.1.2. Preprocesamiento de la información

El **preprocesamiento** consiste en ajustar y preparar la información capturada para que sea adecuada para su posterior procesamiento. El preprocesamiento puede realizarse, bien sobre la información en analógico, o bien después de la conversión a digital:

- En el caso de realizarlo **sobre la información en analógico**, suele tratarse de un preprocesamiento que involucra hardware como filtros basados en circuitos, mejores sensores, etc. Suele ser muy eficaz, pero poco flexible y configurable.
- En el caso de realizar el preprocesamiento **sobre la información ya digitalizada** (discretizada), este será eminentemente realizado por software o código. Suele ser más potente, pero trabaja sobre información ya deteriorada no solo por la captura de la señal, sino también por la propia conversión A/D.

Las tareas más comunes de preprocesamiento suelen cubrir los siguientes aspectos:

- **Eliminación de ruido:** el ruido, entendido como pérdidas o alteraciones en la calidad de la información, puede afectar procesos como la extracción de características y la toma de decisiones.
- **Detección de anomalías:** se produce cuando una de las medidas realizadas carece de sentido o está fuera de un rango determinado.
- **Corrección de errores:** subsana perturbaciones menores en la captura de información, como en imágenes, donde un píxel erróneo se ajusta según los vecinos.



---

## 2.2. PROCESAMIENTO DE LA INFORMACIÓN

Llegados a este punto, la información ya está depurada, limpia y disponible para realizar el **procesamiento de la información** propiamente dicho. Existen diferentes métodos de procesamiento de información existentes. Todos estos pueden aplicarse de forma secuencial y no suelen ser excluyentes:

- **Filtrado y suavizado de la información:** es una operación matemática que elimina o potencia detalles en señales o imágenes. Aunque útil para tareas como detección de objetos, es computacionalmente costosa y, debido a la pérdida de información, restaurar los datos originales es difícil y poco satisfactorio.
- **Segmentación y detección de errores:** es el proceso de dividir información en conjuntos con propiedades similares. Esta operación es computacionalmente costosa y presenta desafíos, como la detección automática del número de segmentos dentro de una imagen, un área aún en investigación. Para lograr una segmentación precisa, es esencial comparar características extraídas de la imagen.
- **Extracción de características:** es el paso siguiente a la segmentación y se refiere a resumir analíticamente cada región obtenida. Permite distinguir regiones con diferentes texturas, intensidades o patrones en señales unidimensionales.

---

## 2.3. TOMA DE DECISIÓN

La **toma de decisiones** es un proceso que busca seleccionar la mejor opción entre varias alternativas, basándose en la información disponible, el contexto y los objetivos a alcanza. En un ejemplo de lectura automática de matrículas, se deben asegurar aspectos como la cantidad de números y letras, identificar que se trata de una matrícula y asociar correctamente los píxeles a números específicos. Estos procesos se realizan en el módulo de toma de decisiones, que también puede ayudar a identificar objetos, como paisajes en una imagen.

## TEMA 3. CAPTURA Y DIGITALIZACIÓN DE SEÑALES

La **teoría de muestreo y cuantificación** nos da los umbrales a partir de los cuales podemos tener una calidad suficiente de señal o de imagen, sin necesidad de emplear mucho almacenamiento.

---

### 3.1. SISTEMAS DE CONVERSIÓN ANALÓGICO DISCRETO

Por **analógico** se entiende toda señal o imagen que es continua y que procede de una fuente o fenómeno físicos como puede ser un sonido. Este tipo de señales se caracterizan por tener todo el detalle de la realidad, no pierden nada de información cuando se transmiten de un punto a otro. Sin embargo, el almacenamiento de una señal analógica es complicado y muy poco eficiente.

Por **discreto (digital)** se entiende una señal o imagen que solo puede alcanzar unos determinados valores, por lo que está discretizada. Normalmente, una señal discreta/digital viene de muestrear y discretizar la señal analógica.

A continuación, se presenta un esquema general de un conversor A/D. Este conversor consta de las siguientes partes:

- En primer lugar, el **muestreador**, que es el encargado de convertir la señal analógica a discreta (digital) tomando fotos de la señal a una frecuencia determinada.
- En segundo lugar, un **cuantificador**, encargado de transformar la foto tomada en el muestreo y convertirla en un conjunto limitado de valores.
- Finalmente, un **codificador** que convierte el valor del cuantificador en binario o en un sistema numérico que pueda transmitirse.

---

### 3.2. MUESTREO Y CUANTIFICACIÓN

---

#### 3.2.1. Muestreo

El **muestreo** consiste en tomar fotos de la realidad (ya sea una señal o una imagen) a una determinada frecuencia de tal manera que la información manejada se reduce, pero no la información necesaria para comprender el mensaje. A continuación, se presenta un esquema general de muestreo. Dicho sistema consta de:

- Una señal de muestreo, conocida como **tren de pulsos**, que define la frecuencia y el tiempo de muestreo, es decir, durante cuánto tiempo se va a observar la señal.

- Un **interruptor** que, en función de la señal de muestreo, se abrirá o cerrará permitiendo tomar las fotos de la señal original.
- Por último, la **señal muestreada**.

De todos estos parámetros, solo hay uno que sí que puede modificarse: la frecuencia de muestreo. Dicha frecuencia viene definida por el teorema de Nyquist-Shannon.

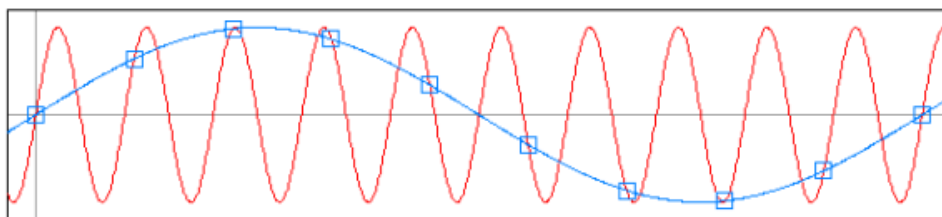
El **teorema de Nyquist-Shannon** establece que la frecuencia de muestreo debe ser al menos el doble de la frecuencia máxima de la señal que se desea muestrear:

$$f_{\text{muestreo}} \geq 2f_{\text{max}}$$

$$f_{\text{max}} \leq \frac{f_{\text{muestreo}}}{2}$$

Si la frecuencia de muestreo es exactamente  $2f_{\text{max}}$ , se dice que la señal está **muestreada en el límite de Nyquist**, lo cual puede generar problemas si hay ruido o variaciones en la señal. Para evitar estos problemas, en la práctica se suele usar una **frecuencia de muestreo un poco mayor** que el doble de la frecuencia máxima.

Existe un aspecto importante a tener en cuenta si no se cumple dicho teorema, el **aliasing**. Ocurre cuando se muestrea por debajo de la frecuencia de Nyquist. Como resultado, las altas frecuencias de la señal original se “confunden” con bajas frecuencias, y las muestras no representan correctamente la señal, lo que provoca distorsión.



**Figura 1.** Aliasing en una señal.

La forma general de un **término sinusoidal** es  $\cos(\omega \cdot t + \phi)$  o  $\sin(\omega \cdot t + \phi)$ . Donde  $\omega$  es la frecuencia angular.

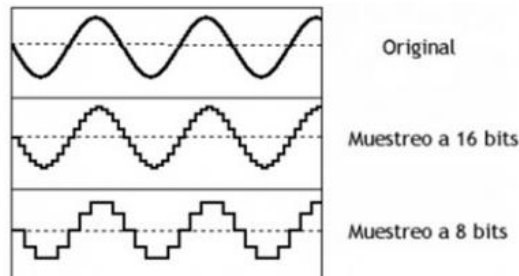
$$\omega = 2\pi \cdot f$$

$$f = \frac{\omega}{2\pi}$$

Para hallar la **frecuencia máxima** de una señal hay que **calcular la frecuencia para cada uno de sus términos sinusoidales**.

### 3.2.2. Cuantificación

La **cuantificación** es el proceso de convertir las muestras tomadas de la señal original en valores numéricos. Dichos valores numéricos poseen una resolución que dependerá del número de bits con los que cuente el procesador. Como se aprecia en la Figura 2, a mayor número de bits, mayor parecido con la señal original.



**Figura 2.** Ejemplo de cuantificación con diferentes niveles de calidad (bits).

En general, los sistemas de cuantificación proporcionan una **cuantificación uniforme**, pero en determinados casos, donde la riqueza de valores es más amplia, suele aumentarse el número de valores en una determinada región de la señal para que así se puede alcanzar mayor resolución donde más se necesita. La elección del número de bits, así como de la uniformidad del rango es algo que depende de la aplicación.

El **paso de cuantificación** define la precisión de dicha cuantificación. Un **paso de cuantificación pequeño significa una mejor precisión**, ya que los valores están más próximos a la señal original.

$$\Delta S = \frac{\Delta V}{2^b - 1}$$

- $\Delta V$  es el **rango dinámico** de la señal que consiste en la diferencia entre el voltaje máximo y el mínimo de la señal, o lo que es lo mismo, entre la amplitud máxima y mínima.

$$V_{rango} = V_{max} - V_{min} = \max(f(t)) - \min(f(t))$$

- $2^b$  es el número total de niveles de cuantificación que se define a partir del número de bits de resolución ( $b$ ).

Los **niveles de cuantificación** son los valores discretos que representan la señal después del proceso de cuantificación, y vienen dados por:

$$q_i = S_{min} + i \cdot \Delta S, \quad \text{para } i = 0, 1, \dots, 2^b - 1$$

$$\circ \quad q_0 = S_{min} = V_{min}$$

## TEMA 4. FUENTES Y TIPOS DE RUIDOS

El **ruido** es toda señal no deseada y de naturaleza aleatoria que modifica la intensidad de la señal original a percibir. En el mundo real, cualquier señal estará afectada por elementos incontrolables que generen ruido. Generalmente, este se superpone a la señal de interés en forma de ruido aditivo. Es decir, nuestra imagen o señal acústica recibida tendrá la siguiente forma:

$$S(t) = f(t) + r(t)$$

- $S$  es la señal recibida.
- $f$  es la señal original.
- $r$  es el componente de ruido.

Por tanto, la primera etapa en el tratamiento de una señal debe centrarse en la identificación y eliminación del mayor número de artefactos ruidosos posible.

La naturaleza aleatoria del ruido provoca que la medición de una misma señal no produzca siempre el mismo resultado. Por lo tanto, no estamos ante señales deterministas sino aleatorias (procesos estocásticos).

---

### 4.1. ENTROPÍA

La **entropía** hace referencia a la complejidad de la señal y se define como la cantidad de información de una fuente aleatoria (en promedio). Es decir, **mide la cantidad de información o incertidumbre en la imagen en bits**. Por tanto, el concepto de entropía sirve para caracterizar una **variable aleatoria**.

Dada una variable aleatoria discreta  $X$ , que toma valores del conjunto  $\{x_1, x_2, \dots, x_m\}$  y de acuerdo a una distribución de probabilidad  $p(X) = \{p(x_1), p(x_2), \dots, p(x_m)\}$ , Shannon definió la entropía de la siguiente forma:

$$H(X) = E\{-\log_2[p(X)]\} = - \sum_{i=1}^m \log_2[p(x_i)] \cdot p(x_i)$$

- $H(X)$  es la entropía de Shannon.
- $\log_2[p(x_i)]$  es la cantidad de información.
- $p(x_i)$  es la probabilidad de que un píxel tenga la intensidad  $i$ .

Así, valores poco probables de la variable están asociados con una mayor cantidad de información (efecto sorpresa) frente a valores de mayor probabilidad.

Por lo tanto, para **calcular la entropía de una imagen**, es necesario determinar la probabilidad ( $p(x_i)$ ) de cada uno de los posibles valores de  $X$ . En este caso, esto implica calcular la probabilidad de que un píxel tenga una intensidad  $x_i$ .

La entropía puede calcularse tanto para **variables discretas** como **continuas**. En el caso de las **imágenes**, los píxeles toman valores discretos, por lo que la entropía se calcula mediante sumatorios. En cambio, en el caso de **señales sin discretizar**, donde los valores son continuos, la suma de probabilidades se reemplaza por una **integral**, que acumula la incertidumbre o la información a lo largo de todos los posibles valores de intensidad.

La **distribución de probabilidad** se puede adoptar dos formas distintas:

- **Uniforme:** se distribuye equitativamente entre todos los valores posibles.
- **No uniforme:** los valores siguen una función de densidad de probabilidad  $f(x)$ , que debe **normalizarse** para que la suma de probabilidades sea 1.

La normalización de dicha función  $f(x)$  se hace de forma diferente según el tipo de variable:

- Valores discretos:

$$p(x_i) = \frac{f(x_i)}{\sum_{x_i} f(x_i)}$$

- Valores continuos:

$$p(x) = \frac{f(x)}{\int_{x_{min}}^{x_{max}} f(x) dx}$$

Si la **distribución de probabilidad** varía en función de **rangos de intensidades**, la entropía se calcula de manera independiente en cada uno de esos rangos:

- **Entropía  $H(X)$  para valores discretos en función de rangos:**

$$H(X) = - \sum_{r \in R} \sum_{i=r_{min}}^{r_{max}} \log_2[p(x_i)] \cdot p(x_i) dx$$

- $R$  es el conjunto de rangos distintos en los que se distribuye la probabilidad.
- $r_{min}$  y  $r_{max}$  son los valores mínimo y máximo de cada rango  $r$ , respectivamente.

- **Entropía  $H(X)$  para valores continuos en función de rangos:**

$$H(X) = - \sum_{r \in R} \int_{r_{min}}^{r_{max}} \log_2[p(x)] \cdot p(x) dx$$

La **intensidad media de la imagen** se puede calcular a partir de la probabilidad de cada píxel ( $p(x_i)$ ) de la siguiente forma:

$$I_{media} = \sum_{i=0}^{N-1} i \cdot p(x_i)$$

- $N$  es el número total de intensidades distintas en la imagen.

---

#### 4.1.1. Entropía diferencial

La **entropía diferencial** o **tasa de entropía** es la **variación de la entropía de la señal debida a la inclusión de una nueva muestra**. A partir de una señal de longitud  $N$  dada por  $x = (x_1, x_2, \dots, x_N)$  y compuesta por la secuencia de la realización de  $N$  variables aleatorias. La entropía del proceso estocástico viene dada por:

$$\begin{aligned} H_N &= E - \log_2[p(x_1, x_2, \dots, x_N)] = \\ &= \int_{-\infty}^{\infty} \log_2[p(x_1, x_2, \dots, x_N)] \cdot p(x_1, x_2, \dots, x_N) dx_1, \dots, dx_N \end{aligned}$$

- $p(x_1, x_2, \dots, x_N)$  es la función de densidad de probabilidad conjunta de las variables que componen el proceso estocástico.

A partir de esta expresión, el **ratio de entropía** de la señal se define de la siguiente forma:

$$E_N = \lim_{N \rightarrow \infty} H_{N+1} - H_N$$

Existen diferentes métodos para la estimación de la entropía de una señal. En este caso, se propone la **entropía aproximada** como procedimiento de estimación. Este algoritmo consiste en estimar la entropía de subseries de longitud  $m$  y  $m + 1$ , de forma que la diferencia de ambas estimaciones proporciona el valor final de entropía. A continuación, se describe cómo obtener la estimación de la entropía para subseries de longitud  $m$ . Para nuestra serie original  $x = (x_1, x_2, \dots, x_N)$ :

- Se extraen todas las subseries de longitud  $m$  representadas por  $x_i^{(m)}$ .

- Posteriormente, se define una tolerancia  $r$  que permite identificar el número  $N^{(m)}(i)$  de las subseries  $x_j^{(m)}$  que se asemejan a  $x_i^{(m)}$ , de acuerdo a una métrica de la distancia entre ambas  $d[x_i^{(m)}, x_j^{(m)}] \leq r$ .
- Por tanto, la probabilidad de encontrar en la serie original una subserie similar a  $x_i^{(m)}$  viene dada por la siguiente expresión:

$$C^{(m)}(i) = \frac{N^{(m)}(i)}{N - m + 1}$$

- $C^{(m)}(i)$  proporciona una estimación discreta de la función de densidad de probabilidad  $f(x^{(m)})$ .
- $N - m + 1$  es el número total de subseries de longitud  $m$  que podemos encontrar en la serie original.

Así, la entropía de los procesos representados por  $x^{(m)}$  se calcula como:

$$H_N^{(m)} = - \frac{\sum_{i=1}^{N-m+1} C^{(m)}(i)}{N - m + 1}$$

El **ratio de entropía** de la serie (proceso estocástico) inicial, es decir, el incremento de entropía por la inclusión de una nueva muestra se obtiene como:

$$E_N = H_N^{(m+1)} - H_N^{(m)}$$

---

## 4.2. FUENTES PERTURBADORAS DE LAS SEÑALES

Las señales eléctricas se ven afectadas por diferentes tipos de ruido:

- **Atmosférico:** viene dado por las señales eléctricas derivadas de las descargas naturales que tienen lugar bajo la ionosfera.
- **Producido por el hombre (ruido industrial):** se refiere a los artefactos eléctricos generados por fuentes como automóviles, motores eléctricos, interruptores, líneas de alto voltaje...
- **Impulsivo o shot:** origina la aparición de valores anómalos (*outliers*) en la señal. Se caracteriza por un aumento brusco de la intensidad durante un corto periodo de tiempo.
- **Galáctico:** perturbaciones producidas más allá de la atmósfera terrestre:
  - Solar: el sol es una gran fuente de emisión de energía en forma de radiación electromagnética.



- Cósmico: otras estrellas próximas a nuestro planeta que también emiten energía en forma de radiación electromagnética.
- **Térmico**: se debe a la agitación aleatoria de los electrones en los elementos de un circuito electrónico. Se trata de una fuente de **ruido inevitable** que siempre va a estar presente en un sistema de captación y procesado de señales.
- **Parpadeo o flicker (1/f)**: su potencia decae por debajo de 1 KHz cuando la frecuencia aumenta. Por lo que tiene mayor impacto sobre las bajas frecuencias.

Ante una fuente de información afectada por artefactos ruidosos, el valor de la **relación señal a ruido** (SNR, *signal to noise ratio*) indica cuantitativamente la calidad de la señal de interés:

$$SNR = 10 \log_{10} \left( \frac{P_S}{P_N} \right)$$

- $P_S$  es la potencia de la señal (dB).
- $P_N$  es la potencia del ruido (dB).

## TEMA 5. DETECCIÓN Y CANCELACIÓN DE ANOMALÍAS

La **detección de anomalías (outliers)** tiene como objetivo la identificación de valores atípicos en la fuente de información. La aparición de *outliers* en una señal o imagen refleja la existencia de ruido, generalmente de tipo impulsivo. Existen diferentes tipos de anomalías:

- **Anomalías puntuales:** se refiere a una muestra que es significativamente diferente de las demás. Estas anomalías se representan por valores extremos que se desvían considerablemente de los demás datos.

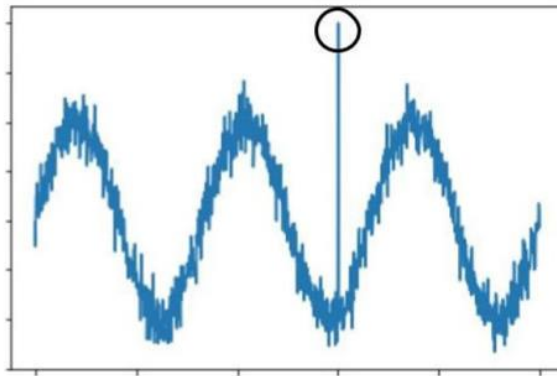


Figura 3. Ejemplo de una anomalía puntual en una señal temporal.

- **Anomalías contextuales:** refiere a una muestra que es considerada anómala en un contexto específico, pero no necesariamente en otro. Cada muestra de datos se define teniendo en cuenta los siguientes atributos:
  - Atributos contextuales: se usan para determinar el contexto (o vecindad) para esa muestra. Vienen dados por la naturaleza de la fuente de datos.
  - Atributos de comportamiento: definen el carácter no contextual de una instancia, es decir, representa el valor de la muestra.

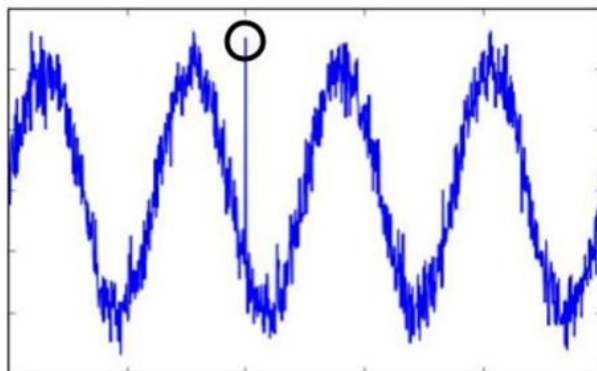


Figura 4. Ejemplo de anomalía contextual en una serie temporal.

- **Anomalías colectivas:** cuando una colección de instancias de datos relacionadas es anómala con respecto a todo el conjunto de datos. Las instancias de datos individuales en una anomalía colectiva pueden no ser anomalías por sí mismas, pero su ocurrencia conjunta como colección es anómala.

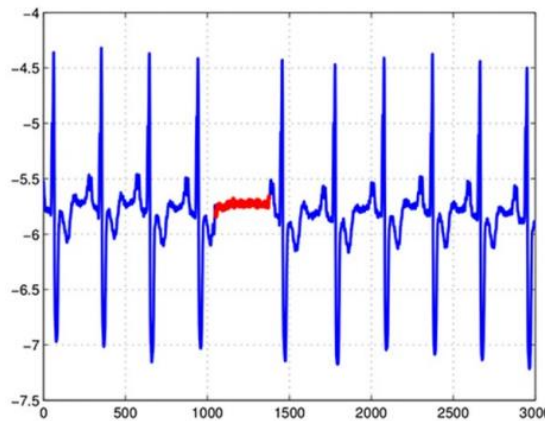


Figura 5. Ejemplo de una anomalía colectiva en una serie temporal.

## 5.1. MÉTODOS DE IDENTIFICACIÓN DE ANOMALÍAS

La configuración de **detección de anomalías** que se utilizará depende de las etiquetas disponibles en el conjunto de datos. Así, podemos distinguir entre tres tipos principales:

- **Métodos supervisados:** se dispone de un conjunto de datos en los que cada muestra está asociada a una etiqueta que indica si es o no una anomalía. A partir de estos datos, se puede entrenar un clasificador que capture el patrón característico de la anomalía. Posteriormente, este clasificador se emplearía como identificador automático de estos valores atípicos. Sin embargo, la identificación de anomalías basada en técnicas supervisadas no está muy extendida, pues el punto de partida es la suposición de que las anomalías son conocidas y etiquetadas correctamente.

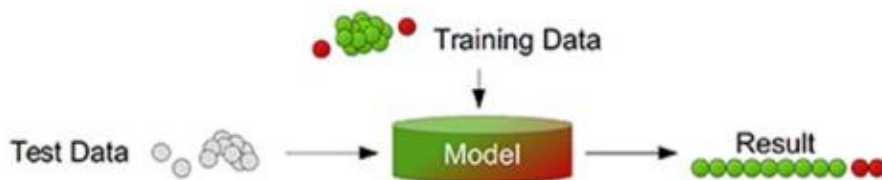
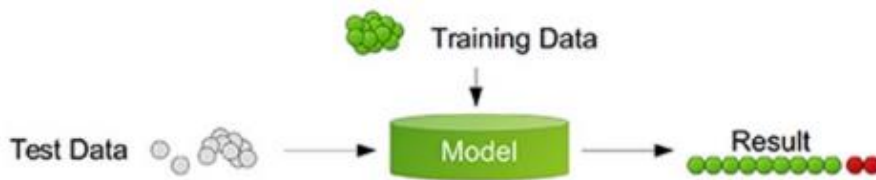


Figura 6. Esquema del método supervisado para detección de anomalías.

- **Métodos semisupervisados:** se emplea un conjunto de datos para entrenar un modelo. Sin embargo, en este caso, los datos de entrenamiento contienen únicamente muestras no anómalas.



**Figura 7.** Esquema del método semisupervisado para la detección de anomalías.

- **Métodos no supervisados:** es el procedimiento más flexible para la implementación de un método de identificación de anomalías pues no requiere de ningún tipo de conocimiento previo sobre los datos (etiquetas). La idea es que un algoritmo de detección de anomalías no supervisado califique los datos únicamente en función de las propiedades intrínsecas de estos. Normalmente, las **distancias o densidades** se utilizan para dar una estimación de lo que es normal y lo que es un valor atípico.



**Figura 8.** Esquema del método no supervisado para la detección de anomalías.

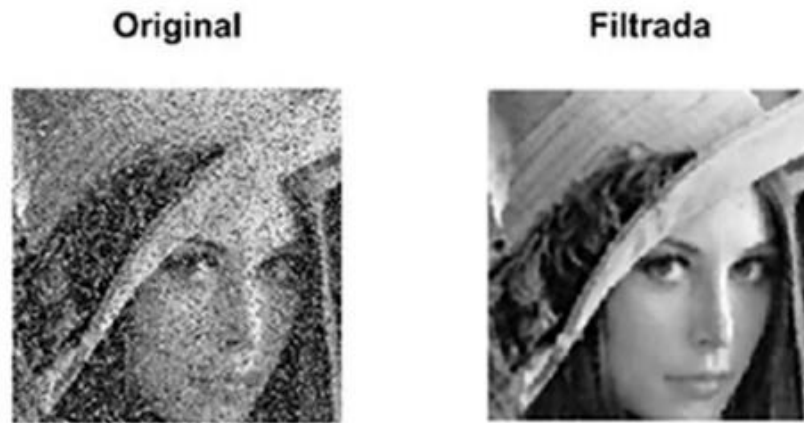
## 5.2. ELIMINACIÓN DE ANOMALÍAS

A continuación, se explican los procedimientos no supervisados más comunes para la eliminación de anomalías en señales. En cada uno de estos métodos, se genera una nueva imagen, lo que significa que la **modificación de un píxel no afecta a los siguientes** durante el proceso de transformación.

- **Filtro de mediana:** es una operación que se aplica punto a punto mediante una ventana deslizante. En el caso de señales unidimensionales como series temporales, se trata de una ventana de longitud  $N$ , mientras que en imágenes la ventana se define en ambas coordenadas y es de tamaño  $N \times N$ . El valor de  $N$  es impar, ya que la ventana se centra en el punto de la señal que se pretende filtrar. Así, el valor resultante en este punto viene dado por la mediana de los puntos considerados por la ventana.

$$\text{Mediana } (n \text{ es impar}) = x_{\frac{(n+1)}{2}}$$

$$\text{Mediana } (n \text{ es par}) = \frac{x_{\frac{n}{2}} + x_{\frac{n+1}{2}}}{2}$$



**Figura 9.** Efecto del filtro de mediana sobre una imagen con ruido impulsivo “sal y pimienta”.

- **Filtro de Hampel:** similar a la de la mediana, pero más efectivo. Reemplaza el punto central de la ventana por el valor correspondiente a la mediana únicamente cuando su diferencia absoluta con la mediana es mayor a un cierto umbral. Evita modificar píxeles de manera innecesaria.
- **Filtro de media:** similar al de la mediana, pero calculando la media aritmética de los píxeles contenidos en la ventana.

$$\mu = \frac{\sum_{i=1}^N x_i}{N}$$

- **Filtro de media geométrica:** similar al filtro de media, pero tiende a perder menos detalles de la imagen.

$$\mu_{geo} = \left( \prod_{i=1}^N x_i \right)^{\frac{1}{N}}$$

- **Filtro de media armónica:** reduce ruido tipo sal o ruido gaussiano, pero falla en presencia de ruido tipo pimienta.

$$\mu_{arm} = \frac{N}{\sum_{i=1}^N \frac{1}{x_i}}$$

- **Filtro máximo:** útil para detectar puntos claros, y reduce ruido de tipo pimienta.
- **Filtro mínimo:** útil para detectar puntos oscuros, y reduce ruido de tipo sal.
- **Técnicas estadísticas:** se basa en la utilización de la función de densidad de probabilidad de los datos  $f(x)$ . Se requieren fijar inicialmente un umbral

que determina la definición de anomalía en nuestro conjunto de datos. Estas son algunas técnicas que pueden ser utilizadas para la estimación de esta función:

- Histograma: a partir de las muestras de una variable, esta se discretiza mediante la división de su dominio en un número limitado de intervalos de igual tamaño e identificados por su punto medio. Estos puntos representan los valores discretos que la variable puede tomar. Así, se obtiene la frecuencia (probabilidad) asociada a cada posible valor a partir del conjunto total de datos inicial contando el número de muestras de la variable que caen en cada intervalo.

Existen diferentes reglas para obtener el número óptimo de intervalos a considerar. Entre ellas, una de las más comunes es la **regla de Freedman-Diaconis**:

$$T = \frac{2 \cdot RIC}{\sqrt[3]{n}}$$

- $T$  es el número de intervalos.
  - $RIC$  es el rango intercuartílico.
- Estimación paramétrica: se asume que la función de densidad de probabilidad que caracteriza estadísticamente a la variable es de tipo normal. Por tanto, la media y varianza de esta distribución son los parámetros a obtener. Para ello, se emplean las estimaciones derivadas de la muestra disponible:

$$\mu = \frac{\sum_{i=1}^N x_i}{N}$$

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

## TEMA 6. PROCESAMIENTO DE IMAGEN. OPERACIONES ELEMENTALES

Una imagen se puede definir como una función matemática  $f(x, y)$  donde:  $(x, y)$  son las coordenadas espaciales de la matriz y  $f(x, y)$  representa la intensidad o color en esa posición (un único valor para escala de grises y un vector para imágenes a color).

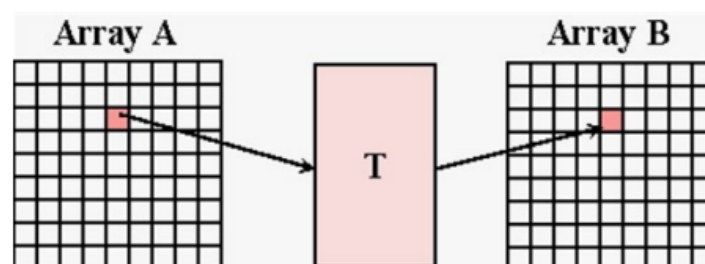
La **resolución** de una imagen está definida por el número de píxeles que es la unidad mínima de información (cada celda de la matriz).

La **profundidad de bits** es la relación entre los bits y los píxeles. Cada valor (escala de grises) o canal (color) es cuantificado utilizando  $2^b$  niveles donde  $b$  es el número de bits. Para una imagen binaria hace falta un único bit para representarla ( $2^1$ , 2 valores: 0 (negro) y 1 (blanco)). Las imágenes estándar utilizan 8 bits ( $2^8$ , 256 valores: 0 (negro) y 255 (blanco)).

El **realce de una imagen** tiene como objetivo optimizar su calidad para facilitar la extracción de información, ya sea por parte de un ser humano o una máquina. Existen diferentes operaciones para el realce de una imagen. En este tema, se pone el foco en las **operaciones elementales de tipo punto a punto**. Estas operaciones se caracterizan por que el valor de un píxel en la imagen procesada es función únicamente del valor de ese mismo píxel en la imagen original:

$$B(x, y) = T(A(x, y))$$

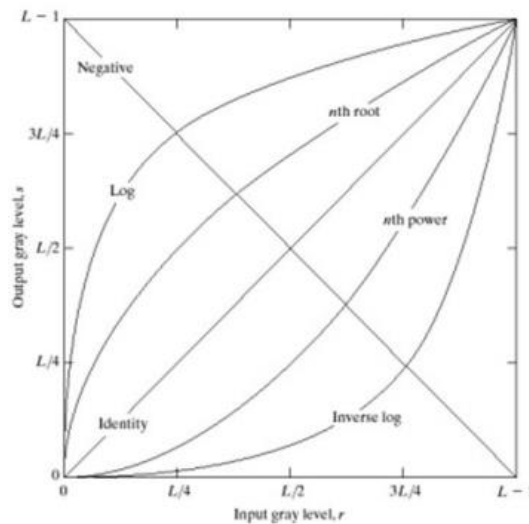
- $B$  es la imagen procesada.
- $A$  es la imagen original.
- La coordenada  $(x, y)$  identifica al píxel sobre el que se aplica la operación.
- $T$  es la función u operador de transformación.



**Figura 10.** Aplicación de un operador punto a punto  $T$  sobre una imagen  $A$  y dando como resultado una imagen  $B$ .

## 6.1. AJUSTE DE INTENSIDAD

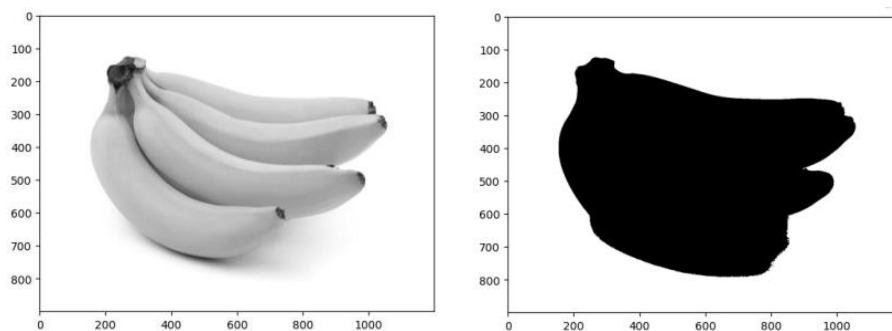
Las **operaciones de ajuste de intensidad** consisten en diferentes expresiones del operador  $T$  de la ecuación anterior. La Figura 11 recoge algunas de las operaciones de transformación más comunes para una imagen con  $L$  niveles de intensidad diferentes.



**Figura 11.** Perfil de las funciones de transformación punto a punto más comúnmente empleadas.

Algunas de estas operaciones de ajuste de intensidad son las siguientes:

- **Umbralización (Thresholding):** se usa para resaltar objetos de interés. Según un umbral predefinido, los valores de intensidad inferiores a este se asignan a 0 (negro), mientras que los valores superiores se establecen en 255 (blanco).



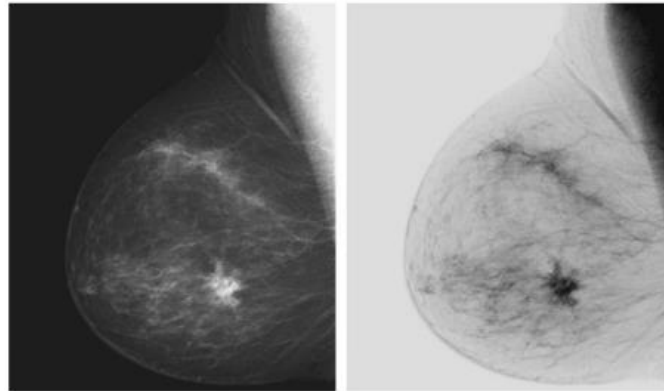
**Figura 12.** Umbralización de una imagen.

- **Negativo:** representa la imagen invertida respecto a la original. Tiene especial utilidad para realzar estructuras de mayor intensidad, es decir, de tonos más próximos al color blanco (que representa el valor máximo de intensidad) encerradas en áreas oscuras de la imagen.

$$T(u) = L - u$$



- $T(u)$  es el nuevo valor de intensidad para un determinado pixel.
- $L$  es el mayor nivel de intensidad que puede tomar un pixel (ej. 255).
- $u$  es el valor de intensidad para un determinado pixel.

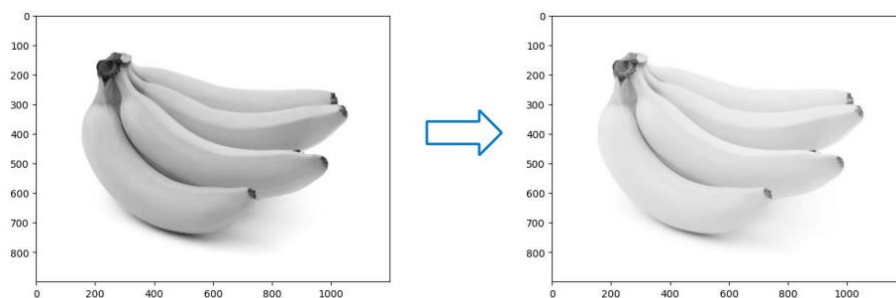


**Figura 13.** Negativo de una imagen.

- **Transformación logarítmica:** se emplea cuando se pretende expandir el rango de intensidad de los píxeles oscuros al mismo tiempo que se equiparan entre sí los píxeles más claros.

$$T(u) = C \ln(1 + u)$$

- $C$  es una constante:  $C = \frac{L-1}{\ln(1+r_{max})}$
- $r_{max}$  es el valor máximo de los píxeles de entrada.

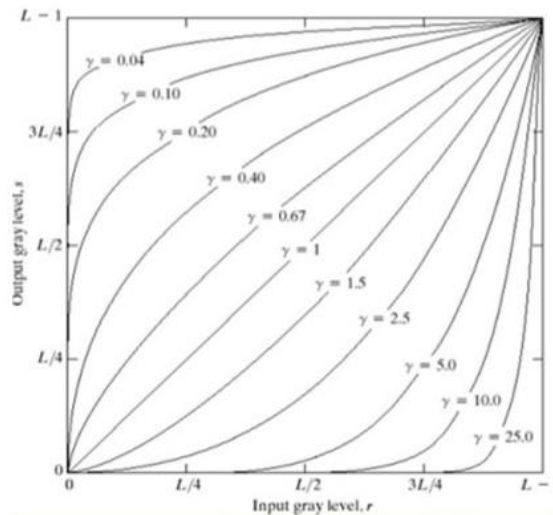


**Figura 14.** Transformación logarítmica de una imagen.

- **Corrección gamma (ley de potencia):** el valor gamma debe ser siempre mayor que 0.

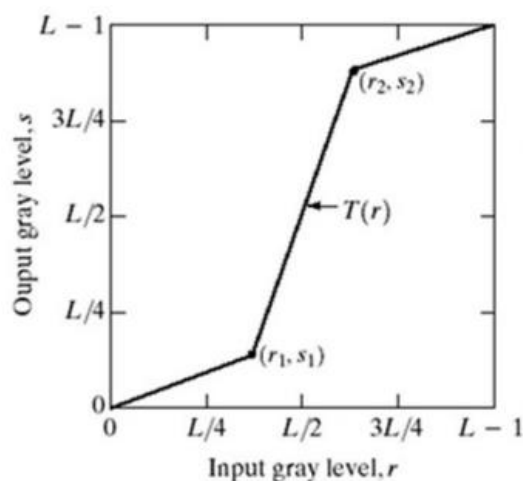
$$T(u) = C \cdot u^\gamma$$

- $\gamma > 1$  **aumenta el contraste en las áreas oscuras** (similar a la transformación logarítmica, pero con más posibilidades). El histograma resultante tendrá una asimetría positiva, en el que habrá más píxeles con intensidades bajas.
- $\gamma < 1$  **aumenta el contraste en las áreas claras**. El histograma resultante tendrá una asimetría negativa, en el que habrá más píxeles con intensidades altas.



**Figura 15.** Funciones de transformación punto a punto obtenidas de acuerdo a la ley de potencias.

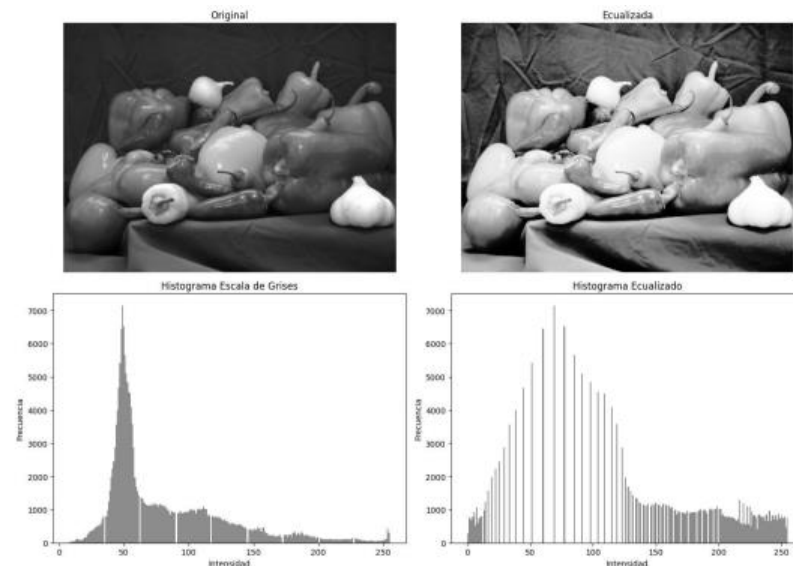
- **Función definida a trozos:** el usuario divide el rango de intensidades de entrada en tantos tramos como se desee. Una de las aplicaciones más comunes de este tipo de funciones es el realce del contraste en una imagen.



**Figura 16.** Perfil de la función de transformación definida a tramos para conseguir el realce de contraste.

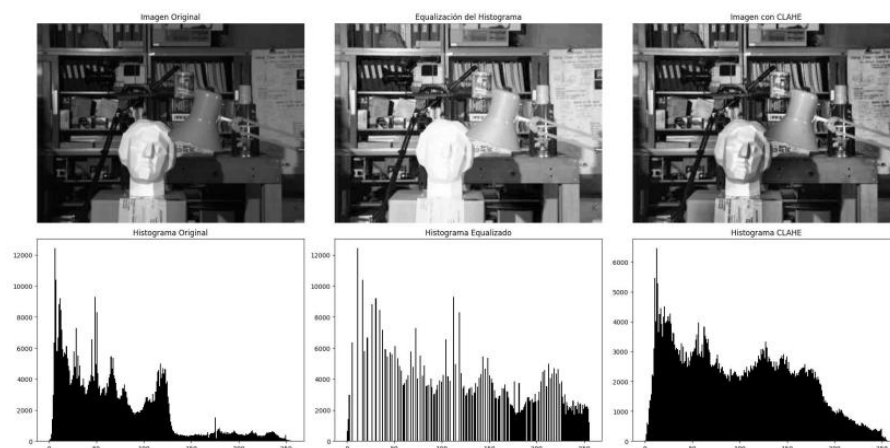
- **Ecualización del histograma:** mejora el contraste de una imagen ajustando la distribución de los niveles de intensidad de los píxeles. Transforma la

imagen para que tenga una distribución uniforme de intensidades. Es decir, **redistribuye los valores de intensidad** para hacer que la imagen tenga una **gama más amplia de intensidades**. Los píxeles que originalmente tienen intensidades similares (por ejemplo, muchos píxeles con una intensidad baja) se redistribuyen de manera que la imagen cubra mejor todo el rango de intensidades, no concentrando todos los píxeles en una pequeña área del histograma.



**Figura 17.** Ecuilización del histograma en una imagen.

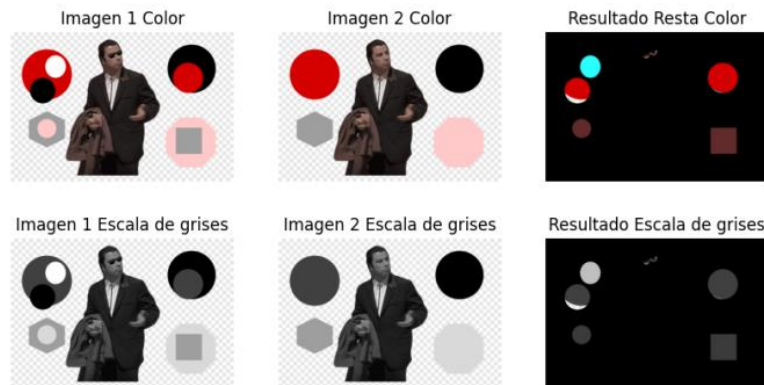
Una versión mejorada de la ecuilización del histograma es **CLAHE** (*Contrast Limited Adaptive Histogram Equalization*). Esta técnica de procesamiento de imágenes ajusta el contraste de manera local, trabajando en pequeñas regiones de la imagen (bloques), en lugar de aplicarlo globalmente como en la ecuilización tradicional. CLAHE permite configurar parámetros clave, como el número de bloques (*n\_tiles*) y el límite de contraste (*clip\_limit*). Esto la convierte en una herramienta especialmente eficaz para imágenes con iluminación desigual o bajo contraste.



**Figura 18.** Comparación entre ecuilización de histograma y CLAHE en una imagen.

- **Operadores aritméticos:** la resta y la suma son los dos operadores más útiles en las etapas de preprocesado para conseguir el realce de una imagen.
  - Resta: permite realzar las diferencias entre dos imágenes.

$$C(u) = A(u) - B(u)$$



**Figura 19.** Operación resta a partir de dos imágenes.

- Suma: permite obtener la imagen promedio a partir de un conjunto, lo que permite reducir el ruido de captación (asumiendo un ruido Gaussiano de media 0).

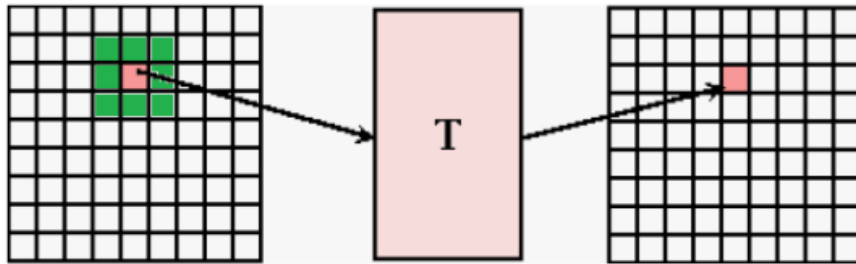
$$C(u) = \frac{\sum_{i=1}^M A_i(u)}{M}$$



**Figura 20.** Operación suma a partir de imágenes con ruido Gaussiano de media 0.

## TEMA 7. PROCESAMIENTO DE IMAGEN. OPERACIONES ESPACIALES

En el presente tema se aborda la utilización de operadores más complejos para el realce de imagen. Concretamente, se estudian diferentes **operaciones espaciales**. Para este tipo de operadores, el valor del píxel resultante es **función del píxel** correspondiente en la imagen original **y de la vecindad de este**.



**Figura 21.** Aplicación de un operador espacial T sobre una imagen A y dando como resultado una imagen B.

Las **operaciones lineales** de interés consisten en **multiplicar** cada píxel en el vecindario **por el correspondiente coeficiente, y sumar** para obtener la respuesta en cada punto  $(x, y)$ . Se selecciona el vecindario alrededor del píxel. Los coeficientes están ordenados en una matriz llamada **filtro, máscara, kernel o ventana**.

La máscara tiene columnas y filas impares de forma que se centra en el píxel bajo estudio. En función de cómo se aplique esta máscara, existen dos tipos de filtrados lineales:

- **Correlación:** consiste en superponer *kernel* (máscara) sobre la imagen y calcular la suma de productos (escalar) de los valores de los píxeles.
- **Convolución:** similar a la correlación, pero antes de aplicar el *kernel*, este se rota  $180^\circ$  (es decir, se invierte tanto el orden de las filas como el de los elementos dentro de cada fila.). Necesaria cuando los **kernels no son simétricos**. La inversión que realiza la convolución garantiza que la dirección del filtro se preserve adecuadamente.

Cuando la máscara se centra en uno de estos píxeles, parte de ella queda fuera de la imagen. Por tanto, hay distintas estrategias a seguir:

- Aplicar la máscara solo a aquellos píxeles que estén situados a una distancia  $\geq \frac{P-1}{2}$  píxeles del borde, en el caso de una máscara  $P \times P$ .

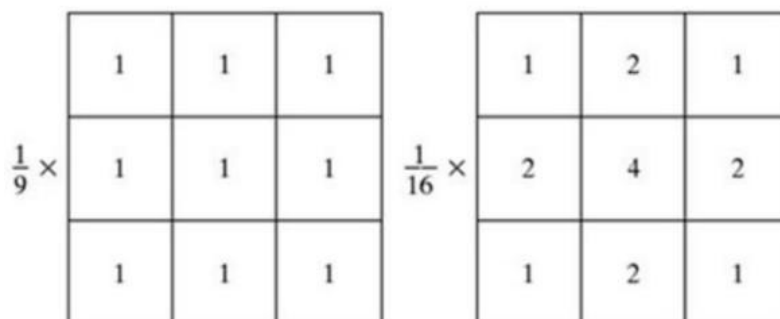
- Aplicar solo a los píxeles cubiertos por la máscara, es decir, la vecindad de un píxel solo incluirá aquellos que estén dentro de la imagen.
- Posibilidad de añadir filas o columnas a la imagen con un valor determinado.

### 7.1. FILTRO PASO BAJO

El **filtro paso bajo** tiene como objetivo atenuar las variaciones bruscas en una imagen, suavizándola y reduciendo el ruido presente en ella. Por lo tanto, atenúa frecuencias altas (bordes y detalles finos) y conserva frecuencias bajas (áreas suaves).

El efecto del filtro paso bajo se obtiene mediante el **cálculo del promedio** a partir del conjunto de píxeles definidos por la máscara, por tanto, los **coeficientes** de esta **son mayores que cero**. Además, a fin de preservar la componente continua de la señal original, estos **coeficientes son normalizados** de forma que **la suma de todos ellos sea la unidad**.

Cuando una máscara tiene coeficientes iguales en todos sus elementos, se denomina **máscara promedio**. Si, en cambio, los elementos de la máscara tienen pesos asociados según su posición, se conoce como **máscara promedio ponderada**.



**Figura 22.** Ejemplo de máscaras promedio y promedio ponderada respectivamente.

### 7.2. FILTRO PASO ALTO

El **filtro paso alto** tiene como objetivo resaltar las diferencias o cambios abruptos en la imagen, como bordes y texturas. Así, aplica un efecto de agudizado (*sharpening*) y también se utiliza para corregir el desenfoque.

El efecto del filtro paso alto se obtiene mediante máscaras con coeficientes tales que lleven a cabo cálculos de diferenciación. Los **coeficientes suman 0** y están **basados en aproximaciones la derivada**.

La **derivada de primer orden (gradiente)** produce bordes más gruesos. Por lo tanto, se usa para **detectar bordes gruesos o áreas con cambios bruscos en la**

**intensidad.** Presenta una respuesta más acusada a cambios con forma de escalón en el nivel de intensidad de la imagen. Es ideal para encontrar bordes que son claramente visibles, como los que aparecen en objetos o estructuras de la imagen.

La **derivada de segundo orden (Laplaciano)** tiene una respuesta más pronunciada a detalles más finos como líneas delgadas o puntos aislados. Por lo tanto, se usa para **detectar bordes más finos o detalles pequeños**, como líneas delgadas, puntos aislados o discontinuidades sutiles en la imagen. También se utiliza cuando se quiere una respuesta más precisa en las transiciones más complejas de intensidad.

De acuerdo con las propiedades mencionadas, existen diferentes tipos de máscaras para detectar bordes:

- **Operador Prewitt (gradiente) X:** detecta cambio en intensidad en la dirección horizontal  $dI/dx$ . Tiene máscaras  $3 \times 3$ . Identifica bordes verticales.

-1	0	1
-1	0	1
-1	0	1

**Figura 23.** Operador Prewitt X que detecta bordes verticales donde la transición es de oscuro a claro.

1	0	-1
1	0	-1
1	0	-1

**Figura 24.** Operador Prewitt X que detecta bordes verticales donde la transición es de claro a oscuro.

- **Operador Prewitt (gradiente) Y:** detecta cambio en intensidad en la dirección vertical  $dI/dy$ . Tiene máscaras  $3 \times 3$ . Identifica bordes horizontales.

-1	-1	-1
0	0	0
1	1	1

**Figura 25.** Operador Prewitt Y que detecta bordes horizontales donde la transición es de oscuro a claro.

1	1	1
0	0	0
-1	-1	-1

**Figura 26.** Operador Prewitt Y que detecta bordes horizontales donde la transición es de claro a oscuro.

- **Operador Sobel:** mejora sobre el operador gradiente básico. Asigna un peso mayor a los píxeles centrales de la convolución, lo que produce un efecto de suavizado y reduce la sensibilidad al ruido.

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

**Figura 27.** Operador Sobel que detecta bordes verticales donde la transición es de oscuro a claro.

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

**Figura 28.** Operador Sobel que detecta bordes horizontales donde la transición es de oscuro a claro.

- **Operador Roberts:** identifica bordes en dirección diagonal ( $45^\circ$  y  $135^\circ$ ). Tiene máscaras  $2 \times 2$ . Respuesta débil a los bordes y sensible a la presencia de ruido.

1	0
0	-1

0	1
-1	0

**Figura 29.** Operador Roberts que detecta bordes diagonales. Se aplica sobre la celda en negrita.

- **Operador Bordes diagonales:** identifica bordes diagonales.

-1	-1	4
-1	4	-1
4	-1	-1

**Figura 30.** Operador Bordes diagonales en la dirección  $135^\circ$ .



- **Operador Laplaciano:** es un **filtro de segundo orden**. Identifica discontinuidades en cualquiera de las direcciones de la imagen (carácter isotrópico).

1	1	1
1	-8	1
1	1	1

**Figura 31.** Operador Laplaciano.

- Realce de detalles:  $I_{enhanced} = I_{original} + \lambda \cdot I_{Laplacian}$
- Reducción de ruido:  $I_{smoothed} = I_{original} - \lambda \cdot I_{Laplacian}$
- **Operador Canny:** es el más efectivo para la detección de bordes. Cuenta con una serie de pasos:

1. Se aplica un filtro paso bajo de tipo gaussiano, cuya característica es que los coeficientes más cercanos al centro tienen mayor valor, y todos los coeficientes están normalizados para que su suma total sea 1. Este filtro suaviza la imagen y ayuda a reducir el ruido presente en ella.

$$\begin{bmatrix} 0.0751 & 0.1238 & 0.0751 \\ 0.1238 & 0.2042 & 0.1238 \\ 0.0751 & 0.1238 & 0.0751 \end{bmatrix}$$

2. Se calcula el gradiente de la imagen en las diferentes direcciones.
3. Se aplica una transformación sobre los píxeles no máximos que consiste en poner a 0 el valor de intensidad de los píxeles que no forman parte de un máximo local.
4. Se aplica un umbral sobre la intensidad de los píxeles para binarizar la imagen y quedarnos con las estructuras que potencialmente pueden corresponderse con bordes.
5. Se eliminan aquellas estructuras más débiles que no están conectadas con bordes mayores.



**Figura 32.** Resultado de la aplicación del operador Canny sobre una imagen.

## TEMA 8. PROCESAMIENTO DE SEÑALES. FILTRADO Y ANÁLISIS DE FRECUENCIA

El análisis en frecuencia es una de las herramientas más poderosas e innovadoras en el procesamiento de la información. Una **señal compleja, aperiódica, aleatoria o caótica** es una señal unidimensional en la que se observan variaciones de intensidad y voltaje, resultado de la superposición de múltiples señales.

El **análisis en frecuencia** es la herramienta encargada de, dada la señal continua, analizar si esta proviene de la suma de dos o más señales y cómo de intensas (potencia) son dichas señales.

La herramienta principal de análisis en el dominio de la frecuencia es la **Transformada de Fourier**, que permite:

- Entender qué componentes de frecuencia posee una determinada señal temporal y cuál es la intensidad de dichas componentes.
- Separar comportamientos temporales de comportamientos de frecuencia.
- Realizar filtrados en frecuencia y separar un determinado número de señales de forma sencilla y con razonamiento matemático, cuya labor en el análisis temporal habría sido imposible.

Para entenderla completamente, es útil comenzar con las series de Fourier y luego avanzar hacia la transformada en sí.

---

### 8.1. SERIES DE FOURIER

Las **series de Fourier** son una forma de **representar funciones periódicas** (señales que se repiten en intervalos regulares) como una suma infinita de senos y cosenos. La idea detrás de esta representación es que cualquier función periódica puede ser descompuesta en una serie de ondas sinusoidales de diferentes frecuencias, amplitudes y fases.

La fórmula general de la serie de Fourier es:

$$f(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cdot \cos(2\pi \cdot n \cdot f_0 \cdot t) + b_n \cdot \sin(2\pi \cdot n \cdot f_0 \cdot t))$$

- $f(t)$  es la función periódica que se quiere representar.
- $a_0$  es el coeficiente de la componente de frecuencia cero (promedio de la señal).

- $a_n$  y  $b_n$  son los coeficientes de Fourier que determinan la amplitud de las ondas seno y coseno de frecuencia  $n$ .
- $f_0$  es la frecuencia base, relacionada con el periodo  $T$  de la señal mediante  $f_0 = \frac{1}{T}$ .
- $n$  es el número de la frecuencia armónica (es decir, 1, 2, 3, ...).

La serie de Fourier divide la señal  $f(t)$  en una suma de componentes sinusoidales con frecuencias múltiplos de una frecuencia fundamental  $f_0$ . Cada término de la suma tiene un valor específico de amplitud, que está determinado por los coeficientes  $a_n$  y  $b_n$ , y un desplazamiento de fase que está reflejado en la forma seno o coseno de cada término.

Los coeficientes de Fourier se calculan utilizando las siguientes fórmulas integrales:

- **Coeficiente  $a_0$  (promedio de la señal):** da el valor medio de la función sobre un periodo  $T$ . Es una medida de la componente de frecuencia cero (promedio).

$$a_0 = \frac{1}{T} \int_0^T f(t) dt$$

- **Coeficiente  $a_n$  (amplitud de la componente coseno):** representa la amplitud de la componente coseno de frecuencia  $n \cdot f_0$ . Se calcula mediante la integral del producto de la señal  $f(t)$  y el coseno  $n \cdot f_0$  de la frecuencia durante un ciclo  $T$ .

$$a_n = \frac{2}{T} \int_0^T f(t) \cdot \cos(2\pi \cdot n \cdot f_0 \cdot t) dt$$

- **Coeficiente  $b_n$  (amplitud de la componente seno):** representa la amplitud de la componente seno de frecuencia  $n \cdot f_0$ . Se calcula de manera similar al coeficiente  $a_n$ , pero usando el seno en lugar del coseno.

$$b_n = \frac{2}{T} \int_0^T f(t) \cdot \sin(2\pi \cdot n \cdot f_0 \cdot t) dt$$

Cuando hacemos referencia al espacio complejo, Fourier quiere incluir funciones  $f(t)$  que tengan valores en el campo complejo, es decir, con el término  $j = \sqrt{-1}$ .

Evidentemente, no existe ninguna función que pueda medirse que dé un valor complejo, pero muchos fenómenos físicos se modelan con estos, principalmente los campos electromagnéticos, por lo que si se quiere extender las series de Fourier a esos fenómenos físicos, es necesario llevar las series de Fourier a este espacio.

Para eso está la **fórmula de Euler**:

$$e^{j\cdot\phi} = \cos(\phi) + j \cdot \sin(\phi)$$

---

## 8.2. TRANSFORMADA DE FOURIER

La **Transformada de Fourier** sirve para extender la serie de Fourier a funciones no periódicas (periódicas con período infinito). Convierte una función de tiempo en una función de frecuencia.

La **Transformada de Fourier** de una señal  $f(t)$ , que es una función en el dominio del tiempo, se define como:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) \cdot e^{-j\cdot\omega\cdot t} dt$$

- $f(t)$  es la señal en el dominio del tiempo.
- $F(\omega)$  es la señal transformada en el dominio de la frecuencia.
- $e^{-j\cdot\omega\cdot t}$  es una función compleja que puede ser descompuesta usando la fórmula de Euler:  $e^{-j\cdot\omega\cdot t} = \cos(\omega \cdot t) - j \cdot \sin(\omega \cdot t)$
- $\omega$  es la frecuencia angular, que está relacionada con la frecuencia normal  $f$  mediante la relación  $\omega = 2\pi \cdot f$ .
- $j$  es la unidad imaginaria  $j = \sqrt{-1}$ .

Dado que la Transformada de Fourier es una representación de la señal en el dominio de la frecuencia, también existe una operación inversa que permite recuperar la señal original en el dominio del tiempo. La **Transformada Inversa de Fourier** está dada por la fórmula:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) \cdot e^{j\cdot\omega\cdot t} d\omega$$

Los **coeficientes de la Transformada de Fourier**,  $F(\omega)$ , son números complejos, lo que significa que tienen dos componentes importantes:

- **Magnitud (  $|F(\omega)|$  )**: indica cuánta de una frecuencia particular está presente en la señal. Es el tamaño del coeficiente complejo.

$$|F(\omega)| = \sqrt{\text{Re}(F(\omega))^2 + \text{Im}(F(\omega))^2}$$

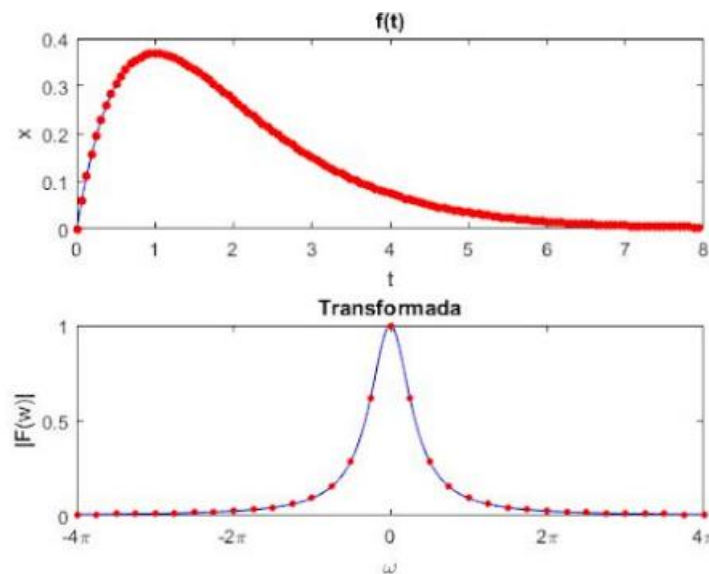
- **Fase ( $\phi(\omega)$ ):** indica el desfase de la componente de frecuencia en relación con el tiempo, lo cual es crucial para reconstruir la señal en el tiempo correctamente.

$$\phi(\omega) = \arctan\left(\frac{\text{Im}(F(\omega))}{\text{Re}(F(\omega))}\right)$$

Esto se puede escribir como:

$$F(\omega) = |F(\omega)| \cdot e^{j\phi(\omega)}$$

El **espectro de frecuencias** es una representación visual que muestra cómo están distribuidas las frecuencias en una señal. Básicamente, te permite ver qué componentes de frecuencia están presentes en la señal y con qué intensidad. Se genera a partir de la magnitud de los coeficientes de Fourier  $|F(\omega)|$  en función de la frecuencia angular  $\omega$ .



**Figura 33.** Espectro de frecuencias para una determinada señal.

### 8.3. TRANSFORMADA DISCRETA DE FOURIER

La **Transformada Discreta de Fourier (DFT)** es una versión discreta de la Transformada de Fourier, utilizada cuando los datos disponibles no son continuos, sino una serie finita de muestras (como ocurre en la mayoría de los sistemas digitales). Es ampliamente utilizada en procesamiento digital de señales, imágenes y otras áreas donde los datos están cuantificados.

Dada una señal discreta  $x[n]$ , que es una serie de  $N$  muestras en el tiempo, su Transformada Discreta de Fourier se define como:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j \frac{2\pi}{N} k \cdot n}, \quad k = 0, 1, \dots, N-1$$

- $X[k]$  es el valor transformado en el dominio de la frecuencia discreta para el índice  $k$ .
- $x[n]$  es el valor de la señal en el tiempo discreto en el índice  $n$ .
- $N$  es el número total de muestras.
- $e^{-j \frac{2\pi}{N} k \cdot n}$  es el núcleo complejo que relaciona el tiempo con la frecuencia.

Para recuperar la señal original  $x[n]$  a partir de su transformada  $X[k]$ , se usa la fórmula de la **Transformada Inversa Discreta de Fourier**:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \cdot e^{j \frac{2\pi}{N} k \cdot n}, \quad n = 0, 1, \dots, N-1$$

La **magnitud y la fase** se calculan de la misma manera que en la Transformada de Fourier, pero aplicadas a cada uno de los coeficientes discretos  $X[k]$ , los cuales representan las componentes de frecuencia de la señal.

El **espectro de frecuencias** se obtiene de la misma manera que en la Transformada de Fourier. Sin embargo, en el caso de la Transformada Discreta de Fourier, es necesario dividir el espectro de frecuencias en componentes discretas. Para ello, se emplea la siguiente fórmula:

$$f_k = \frac{k}{N} \cdot f_s$$

- $f_k$  es la frecuencia asociada con el índice  $k$ .
- $f_s$  es la frecuencia de muestreo (en Hz).
- $k$  es el índice de la frecuencia en la DFT (va de 0 a  $N-1$ ).

$X[k]$  no es la frecuencia en sí, sino la **cantidad de contribución de la frecuencia  $f_k$  en la señal**. El conjunto de todos los  $X[k]$  forma el espectro de la señal (representación de la señal en el dominio de la frecuencia).

Por lo tanto, en la **gráfica del espectro de frecuencias**, el eje vertical muestra la magnitud de cada frecuencia  $k$ , que indica la intensidad de cada componente de la señal, mientras que en el eje horizontal están las frecuencias reales  $f_k$  en Hz.

Esto permite **observar la distribución de la intensidad** (a través de la magnitud  $|X[k]|$ ) **para cada frecuencia en el espectro**, representada en términos de frecuencias reales en Hz, en lugar de usar la frecuencia angular  $\omega$  como en la Transformada de Fourier Continua.

El cálculo directo de la DFT implica  $O(n^2)$  operaciones debido a las sumas anidadas, lo cual puede ser costoso para señales largas. Por eso, se utiliza la **Transformada Rápida de Fourier (FFT)**, un algoritmo optimizado que reduce la complejidad computacional a  $O(n \log(n))$ .

---

#### 8.4. TRANSFORMADA RÁPIDA DE FOURIER

La **Transformada Rápida de Fourier (FFT)** usa propiedades matemáticas de la DFT, como la periodicidad y simetría de los núcleos exponenciales, para reorganizar los cálculos de forma más eficiente. Esto reduce el número de operaciones necesarias a  $O(n \log(n))$ .

## TEMA 9. PROCESAMIENTO DE IMAGEN. MORFOLOGÍA MATEMÁTICA

La **morfología matemática** es una de las ramas de las matemáticas que estudia el análisis y tratamiento de estructuras geométricas basándose en topología y geometría.

La principal utilidad de la morfología matemática es la de **simplificar el aspecto visual de una imagen**, pero conservando sus características o composición principales.

Puede aplicarse también a señales unidimensionales, pero su mayor uso está concentrado en las imágenes binarias y en las imágenes de escala de grises.

Su utilidad se centra en el **limpiado de imágenes ya segmentadas** o la eliminación de detalles tanto en la imagen original como en la imagen segmentada para que los algoritmos de extracción de características puedan obtener resultados más precisos.

### 9.1. ELEMENTO ESTRUCTURAL

La función del **elemento estructural** es similar a la función de máscara en un filtro gaussiano de imágenes: es un **elemento matricial** (mínimo de  $3 \times 3$  píxeles) que recorre la imagen de forma iterativa y sobre el cual los operadores morfológicos actúan.

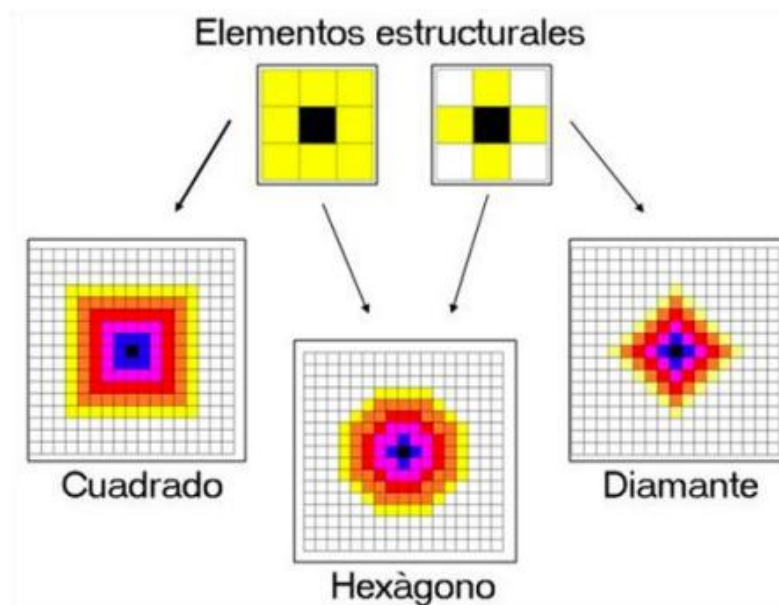


Figura 34. Ejemplos de elementos estructurales con diferentes formas.

Dependiendo de las características de la imagen y del objetivo que se desea alcanzar, se deberán seleccionar diferentes **elementos estructurales (SE)**. Es



importante destacar que el elemento estructural circular, o de disco, suele representarse mediante un hexágono, ya que un círculo no puede aproximarse de manera exacta en una matriz pixelada.

---

## 9.2. DILATACIÓN Y EROSIÓN

La **dilatación y la erosión** son los operadores morfológicos esenciales. A partir de ellos, se crean el resto de los operadores. Es importante destacar que la erosión no es la inversa de la dilatación ni viceversa.

---

### 9.2.1. Dilatación

La **dilatación** consiste en potenciar y aumentar los contornos de una imagen binaria, haciendo que **los detalles se magnifiquen e incluso que algunos objetos dentro de una imagen, que originalmente estén separados, se fusionen**. La dilatación se define matemáticamente como:

$$I \oplus B = \delta_B(I) = \{x \mid (B_x \cap I) \neq \emptyset\}$$

- $I$  es la imagen original.
- $B$  es el elemento estructural.
- $B_x$  es el elemento estructural  $B$  trasladado para que su origen coincida con el píxel  $x$  en la imagen.
- $B_x \cap I$  es la intersección entre el elemento estructural trasladado y la imagen original. Esta intersección indica si  $B$ , colocado en  $x$ , coincide con algún píxel activo ( $\neq 0$ ) en  $I$ .
- $\neq \emptyset$  indica que la intersección no es vacía, es decir, hay al menos un píxel en común entre el elemento estructurante trasladado y la imagen.

El valor dilatado de un píxel  $x$  es el **máximo valor de la imagen en la ventana** definida por el elemento estructurante  $B$  cuando su origen está en  $x$ . Esto significa que, para cada píxel en la imagen, se toma la vecindad definida por  $B$  y se reemplaza el valor del píxel por el valor máximo dentro de esa vecindad. El resultado es que las áreas brillantes de la imagen se expanden.

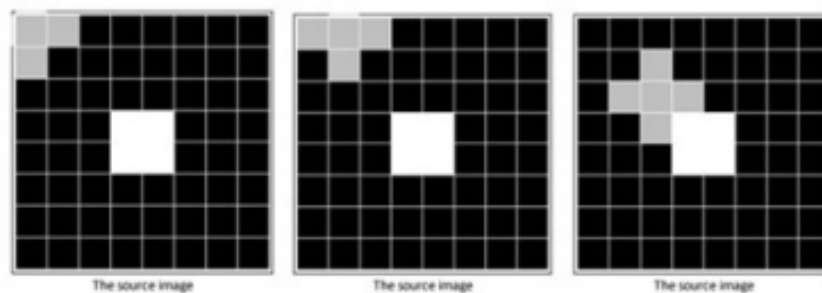
Los píxeles **fuera del borde de la imagen (padding)** se asignan al **valor mínimo**, evitando así su influencia en la dilatación.

Este tipo de funcionalidad tiene mucha utilidad cuando, tras una segmentación, hay objetos que se han quedado separados, pero cercanos en la imagen. La manera de implementar computacionalmente el operador dilatación es de forma

iterativa, comprobando píxel a píxel dentro de la imagen qué relación tienen el elemento estructural ( $B$ ) y la imagen original ( $I$ ).

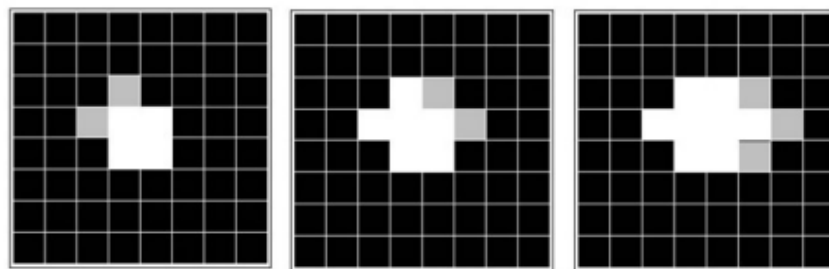
En este caso, el operador mirará si el elemento estructural ( $B$ ) coincide al menos en un píxel con la imagen original ( $I$ ). En ese caso, añadirá a dicha imagen el elemento estructural ( $B$ ), haciendo que la imagen aumente en la zona de los contornos y manteniéndose constante en el interior de esta.

Independientemente del elemento estructural empleado, la imagen resultante después de aplicar la dilatación ha de ser mayor (en el sentido de número de píxeles en blanco) que en la imagen original.



**Figura 35.** Elemento estructural en forma de cruz y la imagen sobre el que se va a realizar la operación morfológica.

Cuando el elemento estructural, toca en al menos un píxel, la zona de blancos dentro de la imagen original es entonces donde el operador dilatación aplica su definición. Es decir, en aquellas posiciones donde ambos coincidan se hará la unión de ambos operadores, tal y como puede verse en la siguiente imagen.



**Figura 36.** Paso a paso de cómo el elemento estructural entra en contacto con una zona dentro de la imagen con píxeles diferentes a cero (negro).

En este caso, y puesto que se trata de una dilatación, se conserva la unión de ambos elementos. Finalmente, tras recorrer toda la imagen, el resultado es que el cuadrado se ha dilatado, ha aumentado de tamaño con la forma del elemento estructural. En ningún momento se aprecia que se haya conservado la forma inicial (se trataba de un cuadrado) y tiene forma de cruz.

### 9.2.2. Erosión

La principal utilidad de la **erosión** como operador morfológico es el de **eliminar detalles, reducir contornos y, en algunos casos, desunir objetos**. La erosión se define matemáticamente como:

$$I \ominus B = \varepsilon_B(I) = \{x \mid B_x \subseteq I\}$$

- $A$  es la imagen original.
- $B$  es el elemento estructural.

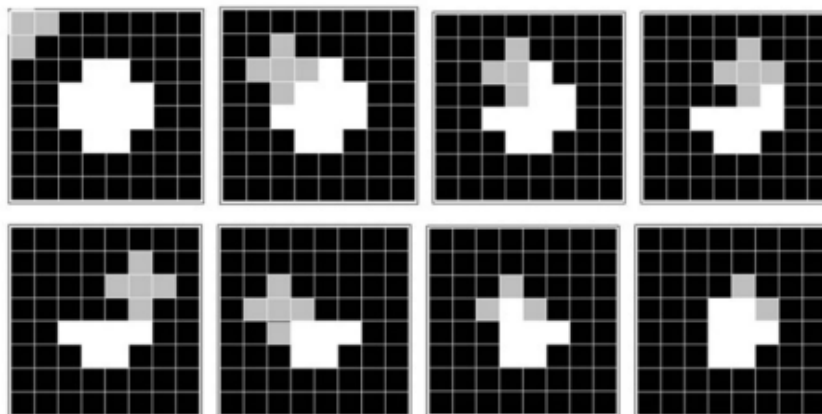
El valor dilatado de un píxel  $x$  es el **mínimo valor de la imagen en la ventana** definida por el elemento estructurante  $B$  cuando su origen está en  $x$ .

Los **píxeles fuera del borde de la imagen (padding)** se asignan al **valor máximo**, evitando así su influencia en la erosión.

La erosión no conserva la forma original. Tras usar este operador, la forma resultante puede haberse erosionado tanto que no conserve la morfología original. La manera de implementar computacionalmente el operador erosión es también de forma iterativa, comprobando píxel a píxel dentro de la imagen qué relación tienen el elemento estructural ( $B$ ) y la imagen original ( $I$ ).

En este caso, el operador mirará píxel a píxel si el elemento estructural ( $B$ ) está contenido íntegramente en la imagen original ( $I$ ). En ese caso, se quedará únicamente con la posición/píxel donde ambos elementos ( $I$  y  $B$ ) coincidan. De esta manera, el contorno de la imagen se reduce considerablemente.

En el proceso de erosión, el número de píxeles blancos en la imagen se reducirá.



**Figura 37.** Proceso de erosión con un operador en forma de cruz.

### 9.3. APERTURA Y CLAUSURA

A diferencia de los operadores anteriores, la **apertura y clausura** son los primeros operadores combinación de varias operaciones morfológicas. Su mayor propiedad es que son capaces de respetar, en la medida de lo posible, la morfología adicional.

Con lo cual, tienen las propiedades de la erosión y la dilatación, pero **manteniendo la forma original**.

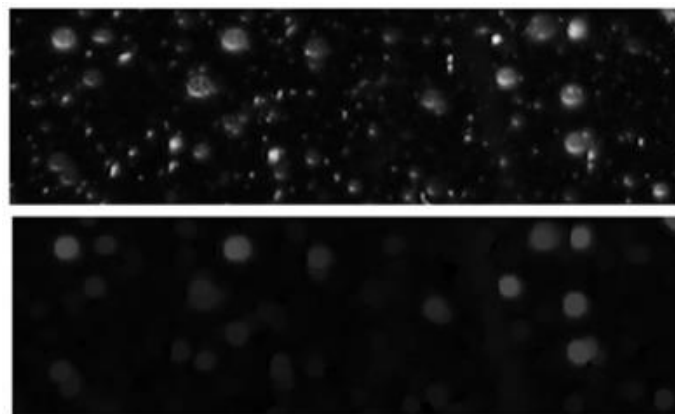
#### 9.3.1. Apertura

La **apertura** (*opening*) sirve para **separar objetos**, ya que elimina los posibles detalles que haya en la imagen en función del elemento estructural. Es un operador morfológico que consiste en la sucesión de una **erosión seguida de una dilatación**. Matemáticamente, la apertura se define como:

$$\begin{aligned} I \circ B &= (I \ominus B) \oplus B = \\ &= \gamma_B(I) = \delta_{B'}[\varepsilon_B(I)] \end{aligned}$$

- $I$  es la imagen original.
- $B$  es el elemento estructural usado en la erosión.
- $B'$  es el elemento estructural usado en la dilatación.

En el ejemplo de la Figura 38, se ve el resultado de aplicar el operador morfológico apertura con un elemento estructural disco de 5 píxeles de radio. El efecto es el de haber eliminado los detalles (puntos) más pequeños, manteniendo los objetos más grandes.



**Figura 38.** Resultado de aplicar el operador morfológico apertura.

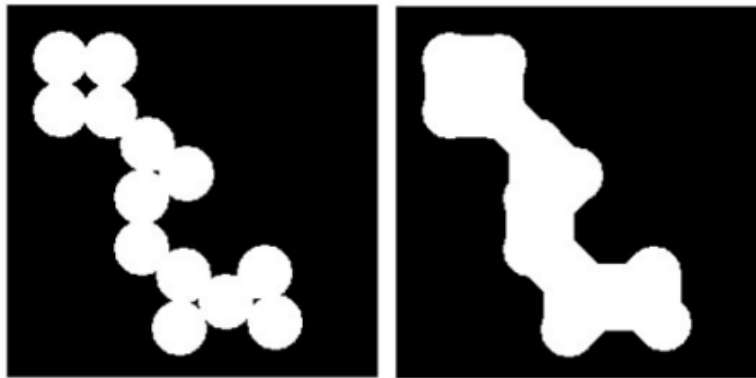
### 9.3.2. Clausura

La **clausura** (*closing*) suele emplearse para **reafirmar y consolidar formas** dentro de una imagen. Es un operador morfológico que realiza primeramente una **dilatación y posteriormente una erosión**. Matemáticamente, la clausura se define como:

$$I \bullet B = (I \oplus B) \ominus B = \\ = \phi_B(I) = \varepsilon_{B'}[\delta_B(I)]$$

- $I$  es la imagen original.
- $B$  es el elemento estructural usado en la dilatación.
- $B'$  es el elemento estructural usado en la erosión.

Como podemos apreciar en la Figura 39, el operador ha transformado la figura inicial en un solo bloque.



**Figura 39.** Clausura con un disco de radio 20 píxeles.

### 9.3.3. Técnica Apertura/Clausura

La técnica de **apertura/clausura** (*opening/closing*) consiste en aplicar primero la operación de apertura y luego la de clausura. Esta técnica es útil para **limpiar el ruido positivo** (eliminando pequeños objetos blancos) y, a continuación, **rellenar huecos o conectar regiones relevantes** de la imagen. Es eficaz para **simplificar la estructura de los objetos** en la imagen, manteniendo sus formas principales y eliminando detalles innecesarios.

## 9.4. GRADIENTE MORFOLÓGICO

El **gradiente morfológico** es un operador que resalta las variaciones en la intensidad del píxel en un entorno definido por el elemento estructural. Hay tres tipos de operadores gradiente:

- **Gradiente de Beucher:** se define como la **diferencia aritmética** entre la **dilatación** y la **erosión**. La idea intuitiva es la de **detectar bordes en la imagen**; ya que la dilatación aumenta una imagen y la erosión la disminuye, la diferencia ha de ser similar a los bordes de una imagen.

$$\rho_B(I) = (I \oplus B) - (I \ominus B) = \delta_B(I) - \varepsilon_B(I)$$

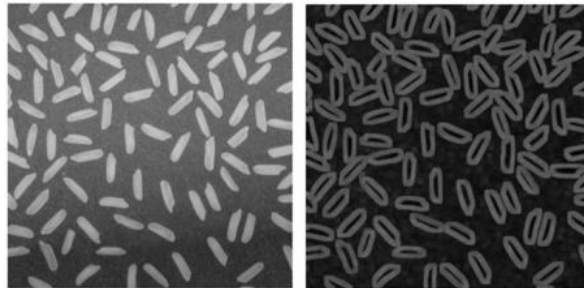
- **Gradiente medio por erosión o gradiente interno:** se obtiene con la **diferencia aritmética** entre la **imagen original** y la **imagen erosionada**. **Resalta los bordes internos** de los objetos mientras **suaviza las regiones externas**.

$$\rho_B^-(I) = I - (I \ominus B) = I - \varepsilon_B(I)$$

- **Gradiente medio por dilatación o gradiente externo:** se obtiene como la **diferencia aritmética** entre la **imagen dilatada** y la **imagen original**. **Resalta los bordes externos** de los objetos mientras **suaviza las regiones internas**.

$$\rho_B^+(I) = (I \oplus B) - I = \delta_B(I) - I$$

La Figura 40 muestra un ejemplo de gradiente morfológico de Beucher aplicado a una imagen con granos de arroz.



**Figura 40.** Gradiente morfológico de Beucher con un disco de radio 3 píxeles.

## 9.5. TOP HAT

El operador **Top Hat** se define como la diferencia entre la imagen original y el operador morfológico apertura o clausura. En función de cómo se realice existen dos tipos de operadores Top Hat:

- **White Top Hat:** se define como la **diferencia aritmética** entre la **imagen original** y el resultado de su **apertura**. **Resalta los detalles brillantes** (blancos) mientras que **suaviza el fondo**. Es la forma más habitual de conservar los objetos pequeños y eliminar los objetos grandes. La apertura se debe realizar con un elemento estructurante intermedio entre el tamaño

máximo de los objetos que se quieren conservar, pero menor que el de los objetos que se quieren eliminar.

$$WTH(I) = I - \gamma(I)$$

- **Black Top Hat:** se define como la **diferencia aritmética** entre la **clausura** de una imagen y la **imagen original**. **Resalta los detalles oscuros** mientras que **suaviza el fondo**.

$$BTH(I) = \phi(I) - I$$

## TEMA 10. PROCESAMIENTO DE IMAGEN. CRECIMIENTO DE REGIONES

La **segmentación** consiste en **dividir una imagen** en regiones/segmentos con **propiedades similares**. Esas propiedades pueden venir definidas bien por el color, la textura o incluso por el hecho de estar acotados por contornos muy bien definidos.

El **etiquetado** en el contexto de la segmentación de imágenes se refiere al proceso de **asignar una etiqueta o identificador a cada región o segmento identificado** en la imagen después de haber sido dividida. Esta etapa es crucial para **interpretar y comprender** el significado de cada región y facilitar el análisis posterior de la imagen segmentada.

La segmentación de imágenes se puede clasificar en varias categorías:

- **Semántica:** asigna a cada pixel de una imagen una etiqueta que representa la clase o categoría a la que pertenece.

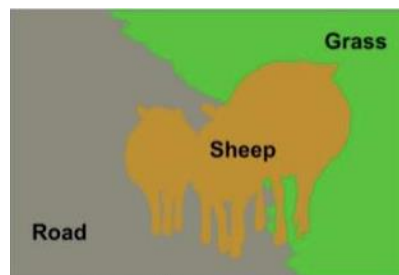


Figura 41. Imagen segmentada semánticamente.

- **Instancias:** extiende a la semántica, ya que asigna a cada pixel una etiqueta única para diferenciar y seguir instancias individuales de objetos, incluso si pertenecen a la misma categoría.



Figura 42. Imagen segmentada en instancias.

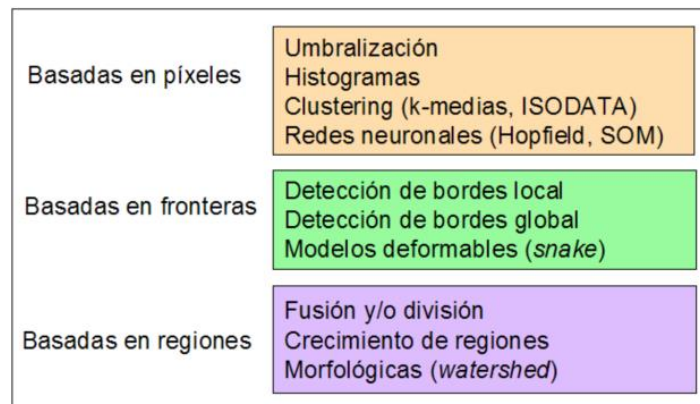
Los **algoritmos de segmentación** se basan, generalmente, según sus dos propiedades principales:



- **Discontinuidad:** obtener una partición de la imagen en base a **cambios bruscos de la intensidad**, lo que suele indicar presencia de bordes. Estos bordes pueden definir regiones dentro de la imagen.
- **Similitud:** identificar regiones que son similares de acuerdo con un **criterio predefinido**. Enfoques en esa categoría donde se incluyen **umbralización y crecimiento de regiones**.

En base las propiedades anteriores, es posible realizar la segmentación de diversas maneras:

- **Basadas en píxeles (similitud):** agrupación de los píxeles de una imagen en conjuntos homogéneos respecto de una o varias características.
- **Basadas en fronteras (discontinuidad):** la segmentación de imágenes es una operación de bajo nivel relativa a la partición de una imagen mediante la localización de bordes.
- **Basadas en regiones (similitud):** la segmentación de imágenes es el proceso de dividir una imagen en distintas regiones, pertenecientes a diferentes objetos, tales que cada región es homogénea pero si se unen dos regiones adyacentes dejan de serlo.



**Figura 43.** Técnicas de segmentación en función de en qué se basan.

### 10.1. SEGMENTACIÓN POR UMBRALIZACIÓN

En la **segmentación por umbralización** se elige un **valor umbral** basado en la distribución de intensidades de píxeles en la imagen. Cada píxel de la imagen se compara con el umbral establecido. Si la intensidad del píxel es **mayor que el umbral**, se le asigna un valor binario de 1 (**objeto de interés**); si es **menor o igual al umbral**, se le asigna un valor binario de 0 (**fondo**).

La umbralización puede realizarse utilizando **múltiples niveles** (umbrales); sin embargo, **presenta limitaciones** cuando los objetos en la imagen tienen un amplio rango de niveles de gris, el fondo no es uniforme o existe presencia de ruido.

El umbral puede **calcularse de forma iterativa** para maximizar su precisión:

- **Umbral global básico:**

1. Calcular el umbral inicial  $T$  como la intensidad media de la imagen.
2. Segmentar la imagen según  $T$ :

$$g(x, y) = 1 \text{ si } f(x, y) > T; \quad g(x, y) = 0 \text{ si } f(x, y) \leq T$$

3. Calcular las intensidades medias  $m_1$  y  $m_2$  de los dos grupos obtenidos en el paso 2.
4. Calcular el nuevo umbral como:

$$T = \frac{(m_1 + m_2)}{2}$$

5. Repetir los pasos 2-4 hasta que la diferencia entre dos valores sucesivos de  $T$  sea menor a una cierta tolerancia.
6.  $g(x, y)$  es la imagen segmentada.

El procedimiento descrito se ha aplicado para el caso de un único umbral (2 clases). No obstante, puede generalizarse para  $C$  clases.

- **Umbral global óptimo (método de Otsu):**

1. Calcular el histograma de la imagen. Sea  $n(i)$  el número de píxeles con intensidad  $i$  de una imagen  $N \times M$ .
2. Normalizar el histograma para obtener las probabilidades de ocurrencia de cada intensidad de píxeles de la imagen.

$$p(i) = \frac{n(i)}{N \cdot M}$$

3. Calcular la probabilidad de asignar un píxel a cada clase dado un umbral  $T$  (candidato):

$$P_1(T) = \sum_{i=0}^T p(i); \quad P_2(T) = \sum_{i=T+1}^{L-1} p(i)$$

4. Calcular las medias ponderadas de intensidades para ambas clases de píxeles.

$$\mu_1(T) = \frac{\sum_{i=0}^T i \cdot p(i)}{P_1(T)}; \mu_2(T) = \frac{\sum_{i=T+1}^{L-1} i \cdot p(i)}{P_2(T)}$$

5. Calcular la media ponderada global de intensidades:

$$\mu_T = \sum_{i=0}^{L-1} i \cdot p(i)$$

6. Calcular la varianza ponderada entre clases utilizando las probabilidades, las medias ponderadas y la media global ponderada.

$$\sigma_B^2 = P_1(T)[\mu_1(T) - \mu_T]^2 + P_2(T)[\mu_2(T) - \mu_T]^2$$

7. Itera sobre los posibles umbrales  $T = [0, L - 1]$  y calcula la varianza ponderada entre clases para cada uno.
8. El umbral óptimo  $T^*$  es aquel que maximiza la varianza ponderada entre clases. Es decir, es el umbral que separa las clases de píxeles de manera más eficiente.

El procedimiento descrito se ha aplicado para el caso de un único umbral (2 clases). No obstante, puede generalizarse para  $C$  clases, donde la varianza ponderada se calcula de la siguiente manera:

$$\sigma_B^2 = \sum_{i=1}^C P_i(T) \cdot [\mu_i(T) - \mu_T]^2$$

Algunas técnicas que se pueden aplicar para mejorar la eficacia de la umbralización son las siguientes:

- Suavizar la imagen para eliminar detalles pequeños.
- Mejorar el aspecto del histograma considerando solo los píxeles que forman los bordes o cercanos a los bordes (usando operadores como el gradiente o el laplaciano).
- Considerar umbrales locales (*variable thresholding*) mediante una ventana.
  - Umbral medio adaptativo: calcula el promedio de los valores de los píxeles en una ventana.

- Umbral gaussiano adaptativo: calcula una suma ponderada gaussiana de los valores de los píxeles en una ventana. Los valores de la ventana más cercanos al píxel actual tienen un peso mayor.

---

## 10.2. SEGMENTACIÓN POR CRECIMIENTO DE REGIONES

El **crecimiento de regiones** es una técnica **no supervisada** que, a partir de píxeles individuales, busca identificar regiones más grandes mediante un proceso de agrupamiento tanto local como global. Esta técnica consiste en **hacer crecer un grupo de píxeles o subregiones** siguiendo un **criterio de crecimiento predefinido**. El crecimiento de regiones puede ser:

- **Bottom-up o basado en semillas**: partiendo de píxeles singulares trata de encontrar regiones más globales basándose en agrupación tanto local como global.
- **Top-down o Split & Merge**: trata de dividir la imagen en secciones más pequeñas de forma iterativa, viendo hasta qué punto dos regiones pueden unirse.

---

### 10.2.1. Crecimiento de regiones basado en semillas

El **crecimiento de regiones basado en semillas** es el más intuitivo y para ello hacen falta los siguientes ingredientes, que serán comunes al resto de métodos de crecimiento de regiones.

- Un conjunto de píxeles  $p_1, p_2, \dots, p_n$  que servirán como semillas.
- La **posición** de esos píxeles vendrá dada por la propia posición que ocupen dentro de la imagen, y las **propiedades (criterio de similitud)** de cada píxel, por la función  $I(p_i)$  donde se elige el nombre  $I$ , ya que comúnmente será la intensidad de color de dicho píxel. El criterio de similitud depende del tipo de imagen utilizada: intensidad, textura o color. Estos son **criterios locales** y no tienen aspectos globales como el tamaño y la forma de las regiones.
- Una **función de similaridad** entre dos píxeles  $p_i$  y  $p_j$ . Esta similaridad vendrá dada por  $S(p_i, p_j)$  y podrá aplicarse bien entre píxeles o bien entre regiones como posteriormente veremos.
- Una **función de pertenencia** a una región; dada  $\pi_r(p_i)$  donde  $r$  hace referencia a la región en concreto y  $p_i$  a un píxel cualquiera. Esta función es 1 si  $S(r, p_i) > T$ , es decir, si la función de similaridad supera un cierto umbral. En cualquier otro caso, será 0. Dicho umbral se fijará de forma

automática por el algoritmo o bien de forma manual, si así lo decide el usuario.

Con lo cual, un algoritmo de crecimiento de regiones basado en semillas lo que intenta es encontrar el número mínimo de regiones que maximiza todas y cada una de las funciones de similaridad y de pertenencia dentro de cada región.



**Figura 44.** Ejemplo de crecimiento de regiones basado en semillas.

Este proceso iterativo cuenta con algunas limitaciones:

- Necesita una **buena inicialización**, es decir, requiere de intervención humana que indique dónde comenzar a crecer.
- Este tipo de algoritmos son computacionalmente muy costosos por dos motivos:
  - Son **algoritmos iterativos y no paralelizables**, ya que hasta que no acabe una iteración no puede comenzar la siguiente.
  - Las **funciones** que exploran si un píxel debe pertenecer o no a una región son **complejas de evaluar y costosas de implementar**, lo que hace que estos algoritmos sean dependientes de estas funciones de evaluación.

### 10.2.2. Crecimiento de regiones basado en Split & Merge

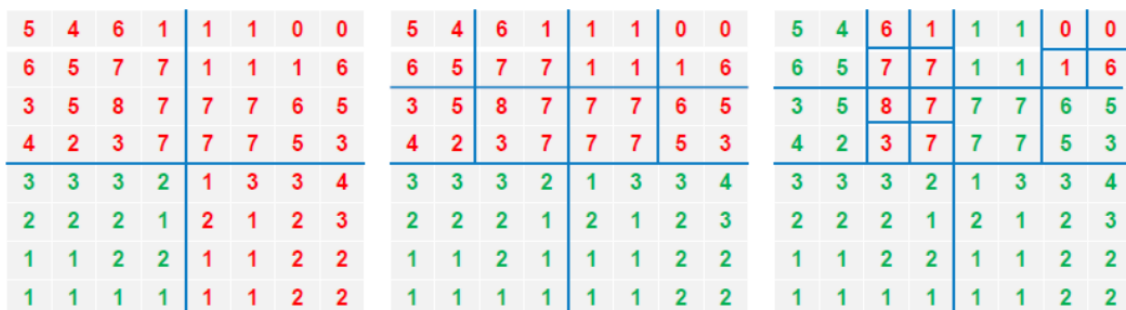
El **crecimiento de regiones basado en Split & Merge** es un enfoque utilizado para segmentar una imagen en regiones homogéneas en función de un criterio de similitud (intensidad, textura o color). Este método combina dos fases:

- **División (split):** la imagen se divide inicialmente en bloques pequeños, generalmente en 4 partes (**quadtree**), aunque es posible elegir cualquier número de particiones según se desee. Cada bloque se analiza para verificar si cumple con un criterio de homogeneidad, como una intensidad uniforme

o un nivel de color similar. Si no se cumple con este criterio, el bloque se subdivide en partes más pequeñas para un análisis más detallado.

- **Fusión (merge):** después de la fase de división, se analizan los bloques adyacentes que puedan ser homogéneos entre sí. Si dos o más bloques cumplen con el criterio de homogeneidad global, se fusionan para formar una región más grande. Este proceso se repite hasta que no sea posible fusionar más bloques.

En el siguiente ejemplo, el criterio de similitud utilizado es la intensidad, con un umbral definido de 3. Así, si un bloque (región) no cumple con este criterio de similitud, se procede a dividirlo (Figura 45). Al finalizar la división, se lleva a cabo el proceso de fusión, uniendo aquellas regiones adyacentes que sí cumplen con el criterio de similitud (Figura 46).



**Figura 45.** Fase de *split* usando el crecimiento de regiones basado en *Split & Merge*.



**Figura 46.** Resultado de las fases de *split* (izquierda) y *merge* (derecha) usando el crecimiento de regiones basado en *Split & Merge*.

### 10.2.3. Crecimiento de regiones basado en Watershed

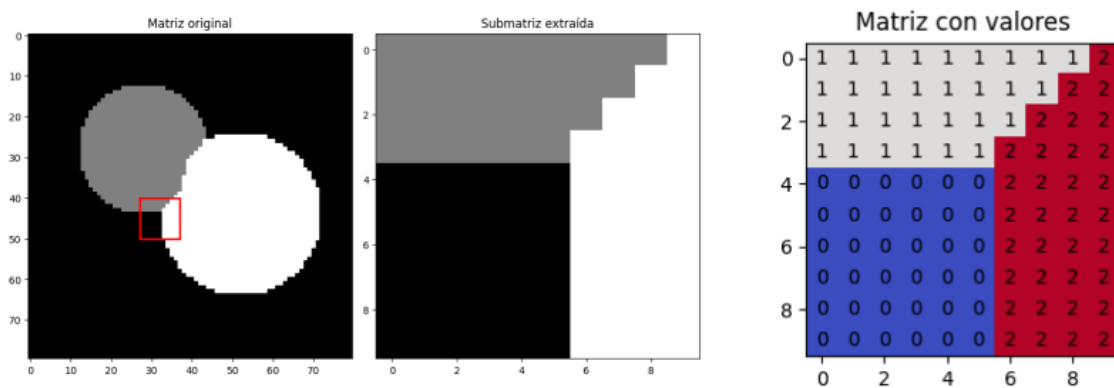
La **transformada de Watershed** tiene como objetivo **segmentar una imagen en niveles de grises**. Trata la imagen como una superficie topográfica, donde los píxeles más oscuros representan valles y los más claros representan elevaciones.

Esta segmentación cuenta con los siguientes pasos:

1. Transformación de la imagen de entrada en una imagen cuyas cuencas de captación son los objetos de interés que queremos identificar.

2. Identificación de los mínimos locales que actuarán como puntos iniciales para iniciar el proceso de segmentación.
3. “Inundación” de la imagen desde los mínimos locales. Asignar etiquetas a los píxeles que se inundan conjuntamente. Cada región formada representa una cuenca hidrográfica.

Por lo tanto, tras el proceso de “inundación”, la transformada de Watershed produce una **matriz de etiquetas**, en la cual cada valor entero positivo representa las posiciones asociadas a las diferentes cuencas de captación en la imagen.



**Figura 47.** Ejemplo de la transformada de Watershed.

La segmentación basada en Watershed **produce normalmente sobresegmentación** y puede deberse a diferentes aspectos como la diferencia de texturas o patrones, ruido o incluso cambios en las tonalidades del color. Para limitar el número de regiones factibles en la segmentación mediante Watershed, es común aplicar las siguientes estrategias:

- Filtrar la imagen de entrada (**filtros espaciales, morfológicos**, etc.)
- Usar el **gradiente** (operadores como Sobel, Prewitt o Canny) o la **transformada de la distancia** (mide la distancia de cada píxel a los bordes más cercanos en una imagen binaria).

#### 10.2.4. Crecimiento de regiones basado en grafos

La segmentación basada en grafos es un enfoque que modela una imagen como un grafo  $G(V, E)$ , donde los píxeles o regiones de la imagen actúan como **nodos** ( $V$ ), y las relaciones entre ellos, como similitud o diferencia, se representan mediante **aristas ponderadas** ( $E$ ).

Esta segmentación consta de algoritmos que tratan de dividir el grafo en  $k$  subgrafos o *clusters*:

- **Graph-Cuts:** tiene como objetivo **minimizar el coste de corte**, el cual se define como la suma de los pesos de las aristas que conforman el *cut-set*. El *cut-set* es el conjunto de aristas cuyos vértices pertenecen a dos subconjuntos disjuntos de una partición  $C = (V_A, V_B)$  del grafo. Matemáticamente, el coste de corte entre dos subconjuntos  $V_A$  y  $V_B$  de vértices se expresa como:

$$cut(V_A, V_B) = \sum_{u \in V_A, v \in V_B} W(u, v)$$

Cada uno de los subgrafos representa un segmento de la imagen. En este algoritmo, existe el riesgo de que queden pequeños elementos aislados. Para mitigar este problema, se puede **normalizar el proceso de segmentación**, favoreciendo la creación de subgrafos de mayor tamaño:

$$Ncut(V_A, V_B) = \frac{cut(V_A, V_B)}{assoc(V_A, V)} + \frac{cut(V_B, V)}{assoc(V_B, V)}$$

$$assoc(V_A, V) = \sum_{u \in V_A, v \in V} W(u, v)$$

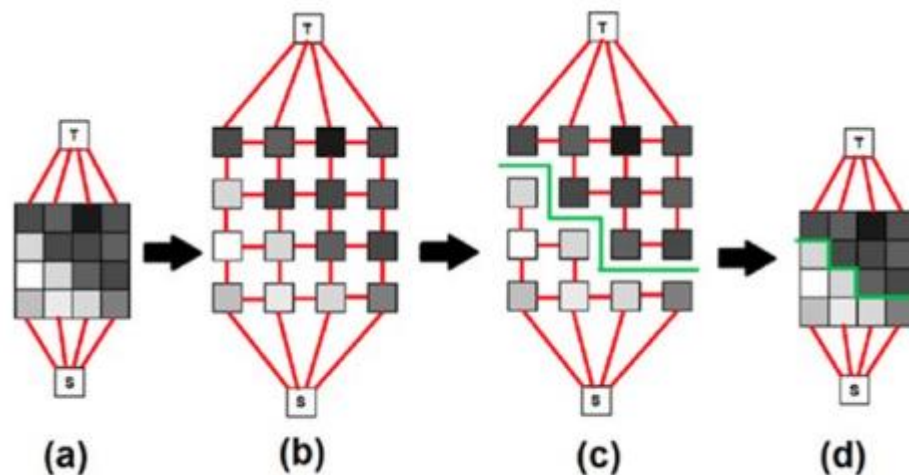


Figura 48. Funcionamiento simplificado de *graph-cuts*.

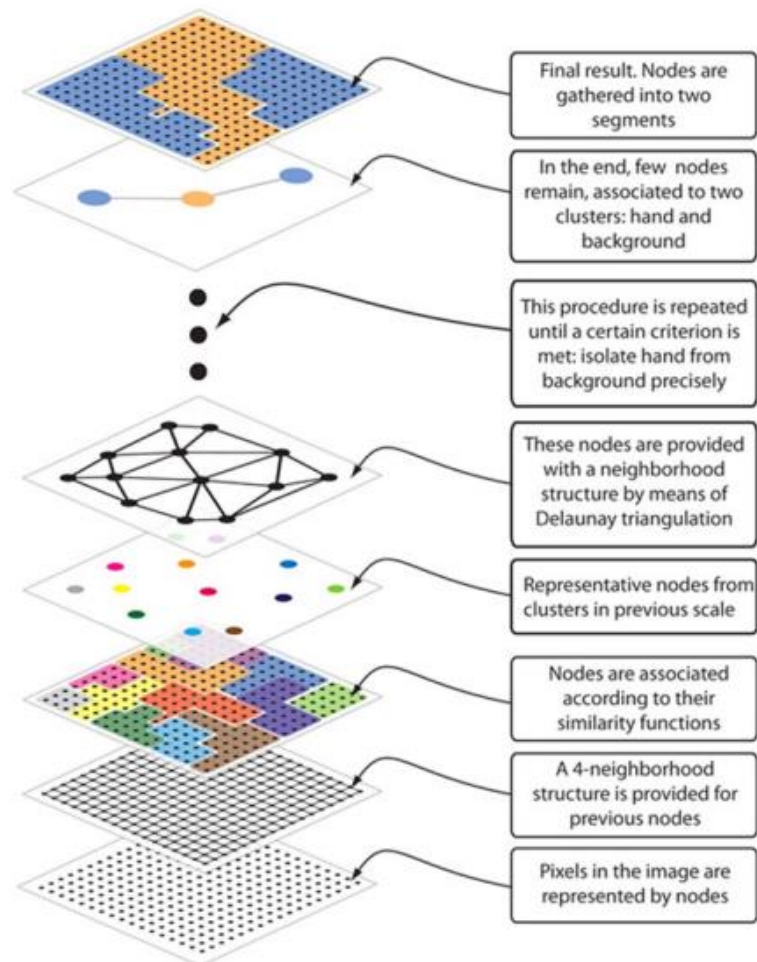
- **Agregación multiescala:** es un proceso de segmentación que comienza con la imagen original, donde cada píxel es un nodo. **Se agrupan los nodos más próximos según un criterio de similitud**, lo que produce una segmentación inicial.

Luego, **se calcula un representante (como el valor medio) para cada grupo**, creando una nueva capa con menos nodos, pero cada uno representando un conjunto de nodos iniciales.



Este proceso de agregación se repite en cada escala hasta alcanzar un número determinado de segmentos o cumplir con una condición de calidad.

Finalmente, se reconstruye la imagen deshaciendo las agrupaciones de cada nivel.



**Figura 49.** Procedimiento de agregación multiescala comenzando por la imagen original (parte inferior) y terminando con la imagen segmentada (parte superior).

#### 10.2.5. Segmentación basada en contornos activos

La **segmentación basada en contornos activos** o basado en Gradient Vector Flow (GVF) utiliza **curvas paramétricas (contornos serpiente)** para delimitar regiones de interés y alcanzar una segmentación automática. Estas **curvas evolucionan de manera dinámica**, adaptándose para identificar contornos que minimicen determinados criterios, tales como la energía de la imagen.

Esta segmentación cuenta con los siguientes pasos:

1. **Inicialización:** definir una curva (contorno) inicial. Esto puede hacerse de manera manual o mediante algún método de inicialización automática.

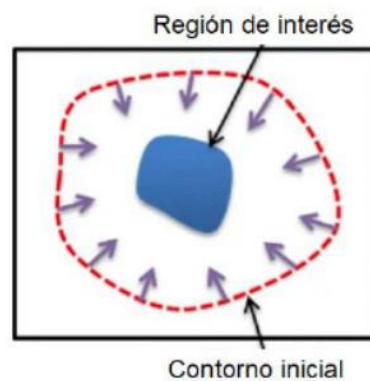
**2. Parametrización de la curva o contorno.**

3. **Definir la función objetivo:** cuantifica la adecuación de la posición y forma de la curva en relación con las características de la imagen. Una función objetivo puede ser maximizar la suma cuadrada de la magnitud del gradiente, lo que es equivalente a minimizar:

$$E = - \sum_{i=0}^{n-1} \|\nabla n_{\sigma} * I(v_i)\|$$

4. **Optimización de la curva:** evolucionar la curva para minimizar la función de energía (objetivo).

Las desventajas de GVF incluyen la dificultad para configurar adecuadamente todos sus parámetros, la posibilidad de que no converja en algunos casos y el mayor coste computacional cuando se aplica a imágenes a color.



**Figura 50.** Funcionamiento simplificado de la segmentación basada en contornos activos.

## TEMA 11. EXTRACCIÓN DE CARACTERÍSTICAS. PROPIEDADES ESTADÍSTICAS Y FRECUENCIALES DE LA SEÑAL

La **definición de características en el procesamiento de imágenes** es el conjunto de propiedades extraídas de una imagen que facilitan su descripción, análisis y clasificación. Se dividen en **distintos tipos** según la información relevante para la tarea:

- **Color:** característica fundamental utilizada para describir imágenes en términos de sus componentes de color. Se representa comúnmente en modelos de color como RGB, HSV.
- **Textura:** describe patrones repetitivos o variaciones de intensidad en una imagen que representan propiedades de la superficie (suavidad, rugosidad o repeticiones). Métodos (descriptores) como GCLM, LBP o Filtros de Gabor cuantifican estas características.
- **Forma:** se refiere a las características geométricas de los objetos en una imagen, como bordes, contornos o características específicas como esquinas. Ejemplos incluyen SIFT y SURF.
- **Estructura:** describe la disposición y organización de elementos en una imagen, incluyendo las relaciones espaciales entre diferentes componentes. Es fundamental para tareas como: Análisis de Componentes Conexas, Esqueleto o Relaciones entre regiones.

La **extracción de características** persigue convertir la complejidad visual en información significativa y manejable. En el procesamiento de imágenes se pueden destacar los siguientes **aspectos clave**:

- **Representación compacta:** una imagen almacena detalles sobre intensidad, color y posición. La extracción de características condensa estos datos en un conjunto más reducido, preservando la esencia informativa clave.
- **Invariancia a transformaciones:** la extracción de características busca invariancia a transformaciones geométricas, como rotaciones o cambios en la escala.
- **Mejora la eficiencia computacional:** al resumir la información visual en características clave, la extracción de características mejora la eficiencia computacional al reducir la dimensionalidad de los datos.

Esta extracción de características consiste en la **síntesis de la información** contenida en una señal, permitiendo representar sus aspectos más relevantes de manera compacta y eficiente. Este proceso está guiado por el conocimiento del problema, seleccionando aquellas propiedades que mejor describan el fenómeno de interés.

Para llevar a cabo esta **síntesis**, es necesario:

- Caracterizar el comportamiento de la señal **midiendo** algunas de sus propiedades (aquellas más relevantes para el problema de interés).
- Como **resultado** se obtiene un conjunto de atributos que **cuantifican** diferentes cualidades de la fuente de información.

Existen **diferentes enfoques** para el proceso de **extracción de características**:

- **Extracción manual:** requiere la identificación y descripción de las características relevantes para un problema específico. Un ejemplo simple es calcular la media de una señal dentro de una ventana temporal.
- **Extracción automática:** emplea algoritmos especializados o redes neuronales profundas para extraer características sin intervención humana, optimizando el proceso en tareas complejas.

Es importante no confundir **Feature Selection** (selección de características), que consiste en elegir las más relevantes de un conjunto dado, con **Feature Extraction** (extracción de características), que transforma los datos en un nuevo espacio representativo.

La **extracción de características** de una señal se puede realizar en **dos dominios distintos**:

- **Dominio natural:** consiste en **analizar la señal tal y como es capturada** en su representación original. Las características se extraen directamente de los valores que la señal tiene en su formato crudo. Algunos ejemplos prácticos son:
  - Voz: se analiza la amplitud de la señal de voz en el tiempo ( $t$ ), es decir, cómo varía la intensidad del sonido con el paso del tiempo.
  - Imagen: Se analiza la intensidad de los píxeles en las coordenadas espaciales ( $x, y$ ).
- **Dominio frecuencial:** representa la señal en el dominio de la frecuencia para analizar patrones de repetición o componentes cíclicos. Esta

transformación se realiza típicamente usando herramientas como la **Transformada de Fourier**.

---

### 11.1. ANÁLISIS EN EL DOMINIO NATURAL

Los métodos basados en el **análisis en el dominio natural** buscan entender y modelar las señales observadas (como imágenes o vídeos) a través de dos enfoques principales:

- **Caracterización estadística:** trata de modelar la señal como una **variable aleatoria** (significa tratar sus valores como muestras de un proceso aleatorio, donde cada punto de la señal tiene cierta probabilidad de tomar un valor específico). Se asume que la señal  $f$  (variable aleatoria) está compuesta de  $n$  muestras u observaciones. Sea  $f_j$  una observación de la señal  $f$  con  $j = 1, 2, \dots, n$ . Sea  $n_j$  el número de muestras de la señal  $f$  con el valor  $f_j$  entonces el número total de muestras de la señal (denotado por  $n$ ) es igual a la suma de las cantidades de cada valor posible  $f_j$  ( $\sum_j n_j = n$ ).

A partir de esta modelización, es posible describir la señal mediante diferentes **medidas estadísticas** que resumen sus características principales. Estas métricas permiten analizar la distribución y variabilidad de los datos, facilitando su interpretación y procesamiento. Algunas de las más utilizadas son:

- Máximo y mínimo: representan los valores más altos y bajos de los datos. La diferencia entre ambos valores define el **rango dinámico** de la señal.
- Media: es el promedio de los datos.
- Mediana: es el valor que divide los datos en dos partes iguales cuando están ordenados.
- Moda: es el valor que aparece con mayor frecuencia.
- Percentiles: dividen los datos en 100 partes iguales. Un percentil específico indica el valor por debajo del cual cae un porcentaje de los datos.
- Rango intercuartílico: es la diferencia entre el tercer cuartil ( $Q_3$ ) y el primer cuartil ( $Q_1$ ).
- Momento Central de Orden K: se define como la expectación matemática de la variable centrada en su media elevada a la potencia  $k$ .

$$\mu_k = E[(f - \mu)^k]$$

- $E$  representa el operador de esperanza matemática, que básicamente calcula el promedio ponderado de los valores.
  - $\mu$  es el valor medio de la variable aleatoria  $f$ .
- Primer momento central: ( $\mu_1 = 0$ ) mide la desviación promedio de los valores respecto a la media. Esta medida **siempre da cero**, porque al calcular las desviaciones, las positivas y las negativas se cancelan. Por eso no aporta información útil y no se utiliza.

$$\mu_1 = E[f - \mu]$$

- Momento central de orden 2: ( $\mu_2 = \sigma^2$ ) **mide la varianza** ( $\sigma^2$ ), que dice qué tan dispersos están los valores alrededor de la media. Si la varianza es pequeña, los valores están **muy cerca de la media** (los datos son homogéneos). Si la varianza es grande, los valores están **muy dispersos** respecto a la media.

$$\mu_2 = E[(f - \mu)^2] = \sigma^2 = \sum_{i=1}^n \frac{(x_i - \mu)^2}{n}$$

- Momento central de orden 3 (Asimetría o skewness): ( $\mu_3 = \gamma$ ) **mide la asimetría** de una distribución de datos respecto a su media. Es decir, indica si los datos están equilibrados alrededor de la media o si tienden a estar más concentrados hacia un lado (izquierda o derecha).

$$\mu_3 = \gamma = \frac{E[(f - \mu)^3]}{\sigma^3}$$

- $\gamma = 0$  indica que la distribución es simétrica.
  - $\gamma > 0$  indica que la distribución tiene una asimetría positiva (hay valores atípicos hacia el lado derecho).
  - $\gamma < 0$  indica que la distribución tiene una asimetría negativa (hay valores atípicos hacia el lado izquierdo).
- Momento central de orden 4 (Curtosis o Kurtosis): ( $\mu_4 = \delta$ ) es una medida estadística que **describe la forma de la distribución de los datos**, especialmente la **presencia de valores extremos (colas)**. En

otras palabras, analiza si los datos tienen una distribución más plana o más “puntiaguda” que una distribución normal.

$$Kurtosis(K) = \delta - 3 = \frac{E[(f - \mu)^4]}{\sigma^4} - 3 = \sum_{i=1}^n \frac{\frac{x_i - \mu}{\sigma}}{\frac{n}{\sigma^4}} - 3$$

- $k > 0$  indica curtosis positiva (leptocúrtica).
  - $k < 0$  Indica curtosis negativa (platicúrtica).
- **Teoría de la Información:** cuantifican la complejidad/regularidad de la señal (cómo de predecible es o qué grado de caos existe en la serie).
    - **Entropía aproximada (ApEn):** es una herramienta matemática que se utiliza para **medir la complejidad o irregularidad** en una serie temporal o conjunto de datos. Clasifica sistemas complejos con pocas muestras. Es especialmente útil para el análisis de señales fisiológicas y detección de patrones anómalos. Para una serie temporal con  $N$  muestras, la ApEn viene dada por:

$$ApEn(m, r, N) = \phi^m(r) - \phi^{m+1}(r)$$

- $m$  es la longitud de las subseries de la serie temporal que se comparan entre sí.
  - $r$  es el factor de tolerancia que define la distancia máxima para que dos subseries se consideren similares.
  - $\phi^m(r)$  es la probabilidad de que dos subseries de longitud  $m$  sean similares en un grado menor o igual a  $r$ .
- **Complejidad de Lempel-Ziv (LZC):** es una técnica de análisis no lineal que **mide la complejidad o aleatoriedad** de una secuencia de datos. Se basa en contar cuántos patrones distintos aparecen en una secuencia. Cuanto más **aleatoria o compleja** sea una secuencia, **más información** se necesita **para codificarla**. Por el contrario, si la secuencia tiene **muchos patrones repetidos**, es **más simple** y necesita **menos información para ser descrita**.

---

## 11.2. ANÁLISIS EN EL DOMINIO FRECUENCIAL

El **dominio frecuencial** se utiliza para **analizar señales en términos de sus frecuencias** (en lugar de en el tiempo). Esto permite identificar componentes importantes, como el rango de frecuencias que están presentes y su intensidad.

La **densidad espectral de potencia (DSP)** es una herramienta que permite analizar **cómo se distribuye la energía de una señal** en el dominio de la frecuencia. Muchas señales tienen características clave que son más evidentes en el dominio frecuencial, como los tonos dominantes en un audio o las frecuencias relevantes en una vibración. Cuando se analiza una señal en el dominio frecuencial, podemos extraer las siguientes características clave:

- **Frecuencia fundamental:** es la **frecuencia más baja** presente en la señal, que a menudo corresponde al tono base o repetición principal.
- **Armónicos:** son **frecuencias múltiplos enteros** de la frecuencia fundamental.
- **Ancho de banda:** representa el **rango de frecuencias** en el cual se concentra la mayor parte de la energía de una señal. Es la diferencia entre las frecuencias superior ( $f_2$ ) e inferior ( $f_1$ ), donde la potencia de la señal está por encima de cierto umbral (3 dB). El ancho de banda define cuánto “espacio” ocupa una señal en el espectro frecuencial. Señales con mayor ancho de banda tienden a transmitir más información.
- **Formantes:** son los **máximos locales** en el espectro de frecuencias. Permiten distinguir los sonidos del habla humana.
- **Entropía espectral (SE):** mide **cómo se distribuye la energía** de una señal en el dominio de la frecuencia. Es una forma de entender qué tan uniforme o concentrada está la potencia de una señal en las diferentes frecuencias. Trata la distribución de potencia normalizada de la señal en el dominio de la frecuencia como una distribución de probabilidad y calcula la entropía de Shannon de esta.

Para una señal  $X(n)$  con espectro de potencia  $S(w) = |X(w)|$ , donde  $X(w)$  es la transformada discreta de Fourier de  $X(n)$ , la distribución de probabilidad  $P(w)$  es:

$$P(w) = \frac{S(w)}{\sum_i S(i)}$$

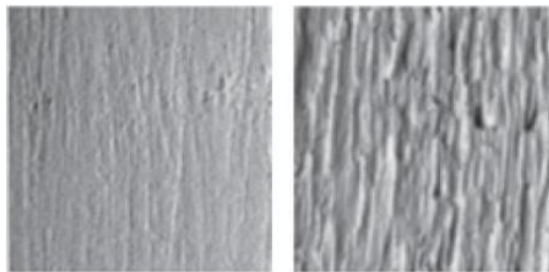
La entropía espectral es entonces:  $H_n = \frac{-\sum_{w=1}^N P(w) \cdot \log_2[P(w)]}{\log_2(N)}$



## TEMA 12. EXTRACCIÓN DE CARACTERÍSTICAS. CARACTERIZACIÓN DE TEXTURA EN IMÁGENES

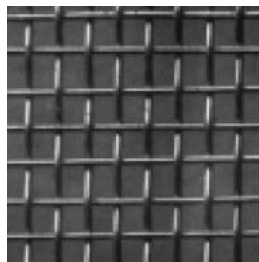
La **textura** se refiere a las **variaciones de intensidad** en una imagen que **forman patrones repetitivos o estocásticos**, proporcionando información visual sobre la **estructura superficial y la disposición espacial** de los objetos. Es una propiedad intrínseca que ayuda a **identificar y distinguir entre diferentes superficies y materiales**. Además, está asociada con cualidades intuitivas descritas por términos como los siguientes:

- **Rugosidad:** se caracteriza por las **variaciones dentro de áreas pequeñas** y puede ser evaluada mediante la observación de cambios en regiones de escala reducida.



**Figura 51.** Ejemplo de imágenes con una textura suave y rugosa.

- **Regularidad:** se refiere a la propiedad de las variaciones a lo largo de una textura completa y se manifiesta mediante la existencia de **patrones periódicos dominantes**.



**Figura 52.** Imagen con una textura regular.

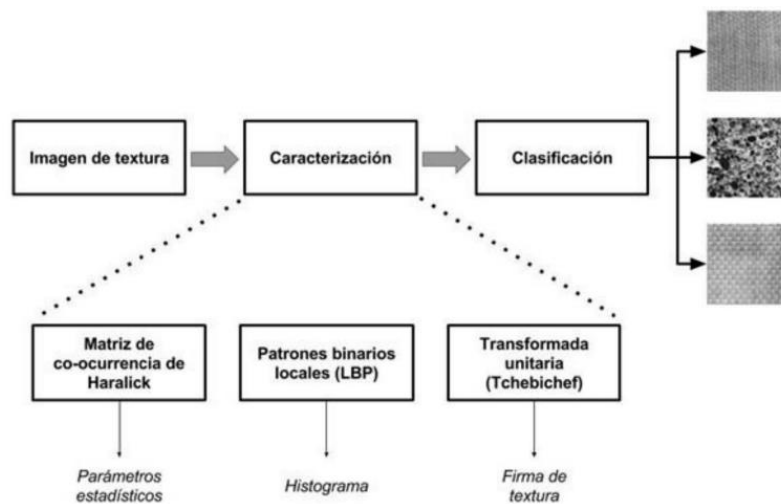
- Direccionalidad: se refiere a la **orientación de los bordes**, dado que estos tienen un impacto significativo en la percepción visual de la imagen. La estimación de la direccionalidad puede llevarse a cabo mediante el **análisis de las orientaciones principales de los bordes**, lo que proporciona información clave sobre la orientación dominante de las estructuras presentes en la imagen.



**Figura 53.** Imagen con una textura regular con direccionalidad vertical.

La caracterización de la textura es **útil para la segmentación de imágenes**, ya que la información textural facilita la segmentación de imágenes, dividiendo la escena en regiones homogéneas basadas en la textura.

El **textón** es la **unidad elemental de la textura** que se repite a lo largo de las dimensiones de la imagen.



**Figura 54.** Principales herramientas para la caracterización de la textura en una imagen.

### 12.1. MATRIZ DE COOCURRENCIA DE HARALICK (GCLM)

La textura está caracterizada por la relación espacial entre las intensidades correspondientes a sus píxeles. La **matriz de coocurrencia de Haralick (GLCM)** refleja las relaciones espaciales entre píxeles vecinos a una **distancia “ $d$ ”** y **dirección “ $\theta$ ”**.

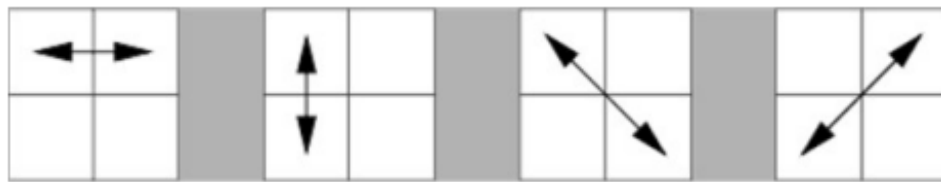
Esta matriz es cuadrada y de **dimensión  $N_g$** , donde  $N_g$  es el número de niveles utilizados para la cuantificación de los valores de intensidad (gris) en la imagen. Se genera **contando el número de veces** que un píxel con valor de intensidad  $i$  ( $N_g = i$ ) es adyacente a un píxel con valor  $j$  ( $N_g = j$ ).

A continuación se describen los pasos básicos para calcular la GLCM:

1. **Definición de la dirección ( $\theta$ ) y la distancia ( $d$ ):** se elige una dirección y una distancia específica para evaluar la relación entre píxeles.

2. **Construcción de pares de niveles de gris:** se recorre la imagen seleccionando pares de píxeles de acuerdo con la dirección y la distancia establecidas.
3. **Creación de la matriz de coocurrencia:** la matriz es cuadrada y tiene dimensiones igual al número de niveles de gris presentes en la imagen.
4. **Normalización de la matriz:** dividir cada elemento por la suma total de la matriz. Esto proporciona una matriz de probabilidad que refleja la probabilidad de que dos niveles de gris aparezcan juntos en la imagen.

Comúnmente, se emplea  $N_g = 8$  y se calculan las ocho matrices posibles (una por cada ángulo diferente).



**Figura 55.** Cuatro direcciones posibles para la identificación de píxeles adyacentes en una imagen y el cómputo de la matriz de coocurrencia de Haralick.

A partir de estas matrices, se **calculan características** como las que se describen a continuación:

- **Momento angular de segundo orden (ASM – Angular Second Moment):** mide la uniformidad en la imagen.

$$ASM = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} P(i, j)^2$$

- **Energía:**

$$Energía = \sqrt{ASM} = \sqrt{\sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} P(i, j)^2}$$

- **Contraste:** mide la variación en la imagen.

$$Contraste = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i - j)^2 \cdot P(i, j)$$

- **Entropía:** representa el grado de desorden en la imagen.

$$Entropía = - \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} P(i,j) \cdot \log_2[P(i,j)]$$

- **Media:**

$$\mu_x = \sum_{i=0}^{N_g-1} i \cdot P_x(i), \quad \mu_y = \sum_{j=0}^{N_g-1} j \cdot P_y(j)$$

- $P_x(i) = \sum_{j=0}^{N_g-1} P(i,j)$
- $P_y(j) = \sum_{i=0}^{N_g-1} P(i,j)$

- **Varianza:** mide la dispersión alrededor de la media.

$$\sigma_x^2 = \sum_{i=0}^{N_g-1} (i - \mu_x)^2 \cdot P_x(i), \quad \sigma_y^2 = \sum_{j=0}^{N_g-1} (j - \mu_y)^2 \cdot P_y(j)$$

- **Correlación:** mide la relación lineal entre los valores de la matriz.

$$Correlación = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \frac{(i - \mu_x) \cdot (j - \mu_y) \cdot P(i,j)}{\sigma_x \cdot \sigma_y}$$

---

## 12.2. PATRONES BINARIOS LOCALES (LBP)

El operador de **patrones binarios locales (LBP)** permite obtener una **transformación píxel a píxel** de la imagen original. El valor transformado de un píxel se obtiene a partir de la **comparación de su valor de intensidad** con los píxeles vecinos. Este operador destaca por su **simplicidad de cálculo** y su **robustez frente al ruido**.

A continuación se describen los pasos básicos para calcular los LBP:

1. Los **píxeles se comparan con sus vecinos**, asignándoles un valor binario (0 o 1) según su intensidad en relación con el píxel central.
2. Estos códigos binarios locales se concatenan para formar un **patrón binario**.
3. Posteriormente, se genera un **número decimal a partir de este código**.

4. La imagen completa se escanea, generando códigos binarios locales que se agrupan en un **histograma**, capturando la distribución de estos patrones.
5. El histograma resultante describe la **textura de la imagen**.

En algunos casos, los valores de  $P$  (número de vecinos) y  $R$  (radio) pueden generar **posiciones de vecinos** que **no coinciden exactamente con coordenadas enteras** en la imagen. Para abordar esto, es necesario aplicar **interpolación** para estimar la intensidad en esas posiciones.

Cuando los vecinos no están en posiciones de píxeles exactos, sus coordenadas se obtienen mediante:

$$x_{vecino} = x_{central} + R \cdot \cos(i \cdot 30)$$

$$y_{vecino} = y_{central} + R \cdot \sin(i \cdot 30)$$

- $i$  toma valores de 0 a  $P - 1$  (en este caso,  $P = 12$ ), y cada vecino se distribuye en un ángulo de  $\frac{360}{P}$  (en este caso,  $30^\circ$ ).

Si las coordenadas obtenidas **no son enteras**, se debe interpolar la intensidad de los píxeles más cercanos en la cuadrícula de la imagen:

- Si la **coordenada es semi-entera** (Ejemplo: (3, 4.5)): se utilizan dos píxeles cercanos para la **interpolación lineal**.

$$I(x, y) = I(x_1, y_1) + (k - k_1) \cdot \frac{I(x_2, y_2) - I(x_1, y_1)}{k_2 - k_1}$$

- $k$  es la coordenada no entera ( $x$  o  $y$ ), por lo que  $k_1$  y  $k_2$  son las coordenadas enteras más cercanas en  $K$ .
- Si la **coordenada no es entera** (Ejemplo: (3.75, 4.30) ): se usa **interpolación bilineal** con los cuatro píxeles más cercanos.

$$I(x, y) = I(x_1, y_1) \cdot (1 - a) \cdot (1 - b) + I(x_2, y_1) \cdot a \cdot (1 - b) + I(x_1, y_2) \cdot (1 - a) \cdot b + I(x_2, y_2) \cdot a \cdot b$$

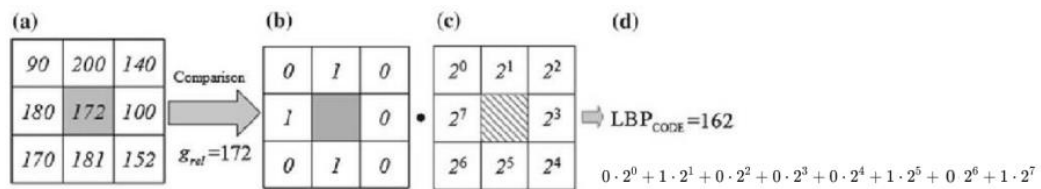
- $x_1$  y  $x_2$  son las coordenadas enteras más cercanas en  $X$ .
- $y_1$  y  $y_2$  son las coordenadas enteras más cercanas en  $Y$ .
- $a = \frac{x - x_1}{x_2 - x_1}$
- $b = \frac{y - y_1}{y_2 - y_1}$

El **operador original de LBP** se obtiene de la siguiente forma para cada píxel de la imagen:

$$LBP(P, R) = \sum_{p=0}^P s(g_p - g_c) \cdot 2^p$$

$$s(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$

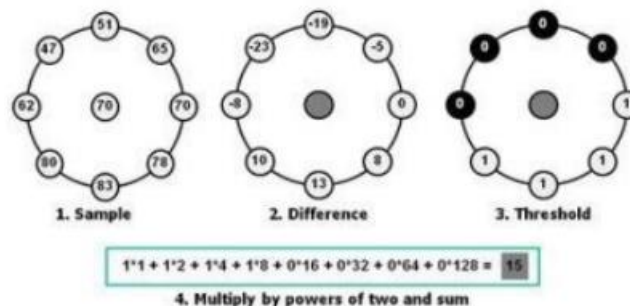
- $g_c$  es el nivel de gris del píxel central.
- $g_p$  corresponde a los niveles de gris de  $P$  píxeles equidistantes sobre un círculo de radio  $R$ .



**Figura 56.** Cálculo del operador LBP para un píxel.

El procedimiento para el cálculo del LBP es el siguiente:

1. Tomar la muestra de píxeles vecinos (habitualmente se recorren en sentido horario, tomando como referencia el píxel superior).
2. Calcular la diferencia del píxel central con estos.
3. Aplicar el umbral 0 sobre esta diferencia.
4. Se transforma la cadena binaria obtenida en el número decimal correspondiente (multiplicando cada bit por las potencias de 2 de izquierda a derecha, es decir,  $2^0, 2^1, \dots, 2^7$ ).



**Figura 57.** Etapas involucradas en el cálculo del operador LBP.

Por tanto, cada píxel de la imagen es asignado con una de las  $2^P$  posibles etiquetas (puesto que se genera una cadena de  $P$  bits). Comúnmente, el radio  $R$  es tal que

$P = 8$ . El histograma de estas  $2^8 = 256$  etiquetas diferentes se puede usar como un **descriptor de textura**.

Dado el elevado número de contenedores del histograma resultante, el resultado de la transformación tenderá a generar histogramas con un número significativo de 0, pues muchas de las  $2^P$  etiquetas posibles no habrán sido asignadas a ningún píxel. Además, el resultado de la transformación variará si la imagen original es rotada un determinado ángulo.

A fin de superar estas limitaciones, se propuso una **extensión del operador LBP** original basada en la definición de los llamados **patrones uniformes**. Esta extensión se inspiró en el hecho de que algunos patrones binarios aparecen con mayor frecuencia en imágenes de textura que otros.

Un patrón binario local se denomina **uniforme** si su cadena de bits contiene como **máximo dos transiciones** de 0 a 1 o viceversa. Por ejemplo:

- Los patrones 00000000 (0 transiciones), 00001110 (2 transiciones) y 11110011 (2 transiciones) son uniformes.
- Mientras que los patrones 00110110 (4 transiciones) y 10101101 (6 transiciones) no lo son.

En función del **número de transiciones**:

- **Menos transiciones:** indica una región más homogénea donde los cambios entre intensidades de píxeles vecinos son menos frecuentes. Esto podría corresponder a áreas de textura uniforme.
- **Más transiciones:** indica una región más variada o texturizada donde hay más cambios abruptos en las intensidades de píxeles vecinos. Esto podría corresponder a bordes o áreas con cambios rápidos de textura en la imagen.

El **LBP uniforme se obtiene rotando circularmente cada patrón de bits al valor mínimo**. Por ejemplo, las secuencias 1000011, 1110000 y 0011100 surgen de diferentes rotaciones del mismo patrón local y todas corresponden a la secuencia 0000111.

Por lo tanto, el histograma resultante del **LBP uniforme tiene menos bins que el LBP tradicional** porque agrupa los patrones no uniformes en un solo *bin*. Esto se hace porque el objetivo de LBP uniforme es reducir la complejidad y capturar solo aquellos patrones que son más significativos (los uniformes), ya que se asume que estos son los más comunes en las texturas. Para el caso de  $P = 8$ , hay 59 patrones uniformes posibles y un bin adicional para los patrones no uniformes. Esto reduce

la cantidad de *bins* activos, pasando de 256 posibles valores en el LBP tradicional a un número menor en el LBP uniforme.

El histograma obtenido a partir del LBP permite extraer diversas características para describir la textura de una imagen, como la media, varianza, asimetría, curtosis y entropía, entre otras.

---

### 12.3. TRANSFORMADAS UNITARIAS

Las **transformadas unitarias** permiten representar imágenes en un espacio transformado mediante la proyección sobre **funciones ortonormales**, generalmente sinusoidales. Estas funciones **capturan la textura** de una imagen, ya que esta se caracteriza por la repetición de estructuras.

Ejemplos de transformadas incluyen la **de Fourier** y la **del coseno**, pero en este caso se destaca la **transformada de Chebyshev**, cuyos coeficientes, llamados **momentos de Chebyshev**.

Los **momentos de Chebyshev** son un conjunto de momentos ortogonales basados en **polinomios de Chebyshev**. Proporcionan una representación compacta de la distribución espacial de las intensidades de píxeles de una imagen. Para calcular los momentos de Chebyshev, la imagen se transforma utilizando los polinomios de Chebyshev, y los coeficientes resultantes representan los momentos. Estos coeficientes pueden utilizarse como características para diversas aplicaciones de análisis de imágenes.

Formalmente, el **momento de una imagen**,  $T_{pq}$ , donde  $p$  y  $q$  son enteros positivos y  $s = p + q$  denotando **su orden**, es calculado proyectando la imagen en una base de polinomios (polinomios de Chebyshev).

$$T_{pq} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} t_{pq}(x, y) \cdot f(x, y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} t_p(x) \cdot t_q(y) \cdot f(x, y)$$

- $t_p(x)$  y  $t_q(y)$  se calculan de forma recursiva:

$$t_0(x) = 1$$

$$t_1(x) = x$$

$$t_n(x) = 2x \cdot t_{n-1}(x) - t_{n-2}(x), \quad \text{para } n \geq 2$$

El momento mide la **correlación entre la imagen**  $f(x, y)$  y el **correspondiente polinomio**  $t_{pq}(x, y)$ .



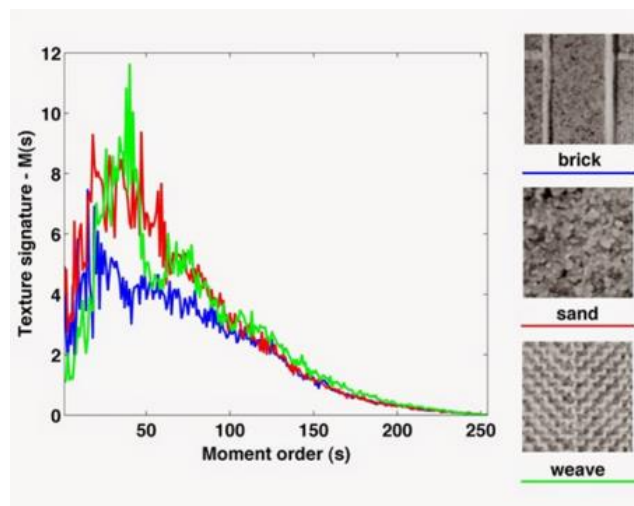
Para caracterizar la textura de una imagen, generalmente, **se calcula el promedio de las magnitudes absolutas de los momentos  $T_{pq}$  para un orden específico**. Esto permite analizar la distribución de las frecuencias presentes en la imagen.

- **Momentos de orden bajo** (con  $p$  y  $q$  pequeños) capturan las **frecuencias bajas**, es decir, patrones grandes.
- **Momentos de orden alto** (con  $p$  y  $q$  grandes) capturan **frecuencias altas**, es decir, patrones pequeños.

Una vez que se tienen las magnitudes de los momentos  $T_{pq}$  para cada orden, se genera una **curva de firma de textura**. Esta curva muestra cómo se repiten las estructuras en la imagen según el orden de los momentos.

La curva resultante  $M(s)$ , función del orden  $s$ , te da información sobre la **frecuencia de los patrones** presentes en la textura:

- Si la curva tiene **picos en órdenes bajos** (valores pequeños de  $p$  y  $q$ ), significa que la textura tiene **patrones grandes y repetitivos** (por ejemplo, ladrillos).
- Si la curva tiene **picos en órdenes altos** (valores grandes de  $p$  y  $q$ ), significa que la textura tiene **patrones más pequeños y detallados** (por ejemplo, arena o tejido).



**Figura 58.** Firmas de textura para tres imágenes diferentes derivadas de los momentos Chebyshev de la imagen original.

A partir de esta curva, se pueden extraer diversas características para construir un vector de características, incluyendo valores estadísticos, extremos locales, pendientes en los extremos, entre otras.

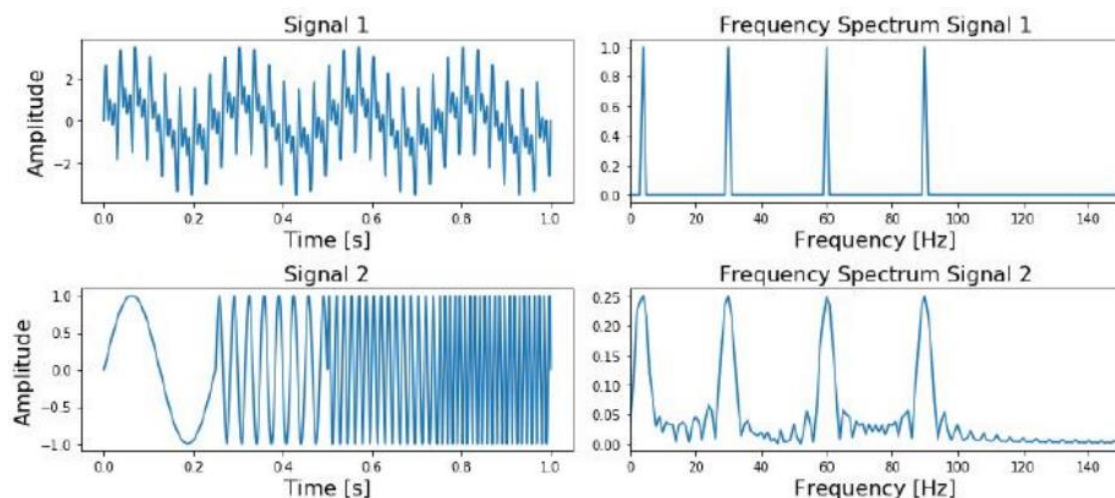
## TEMA 13. EXTRACCIÓN DE CARACTERÍSTICAS. PROCESAMIENTOS MULTIESCALA Y MÉTODOS AVANZADOS

Un problema tan sencillo de definir como puede ser la identificación de un objeto independientemente de su rotación o traslación es altamente complejo usando las extracciones de características presentadas hasta ahora, por no decir imposible.

Es por eso por lo que surgen otras herramientas como son la transformada Wavelet, los filtros de Gabor o el algoritmo SIFT (*Scale Invariant Feature Transformation*) que son capaces de proporcionar características más robustas.

### 13.1. SHORT TIME FOURIER TRANSFORM (STFT)

La **Transformada de Fourier de Tiempo Corto (STFT)**, permite estudiar cómo cambia las componentes de frecuencia de una señal a lo largo del tiempo o el espacio. Es una extensión de la Transformada de Fourier tradicional que está diseñada para **señales no estacionarias**, es decir, componentes frecuenciales varían en diferentes momentos o regiones.



**Figura 59.** Señales estacionaria (arriba) y no estacionaria (abajo).

La STFT resuelve este problema al aplicar la Transformada de Fourier en **ventanas de tiempo o espacio**, permitiendo analizar la evolución de las frecuencias en diferentes regiones.

La STFT se lleva a cabo en tres etapas principales:

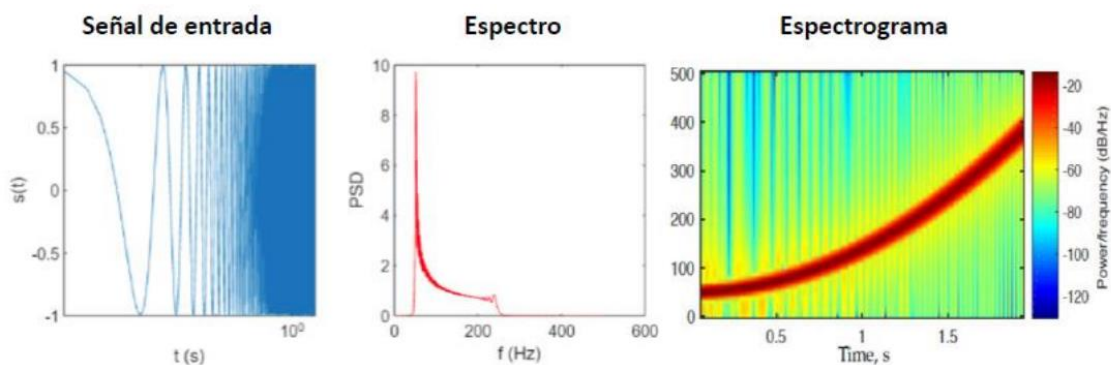
1. **Segmentación y aplicación de ventana:** la señal de entrada se divide en segmentos de igual longitud mediante una ventana deslizante. La elección del tipo de ventana y su tamaño es fundamental, ya que influye directamente en la resolución tiempo-frecuencia de la STFT.

2. **Cálculo de la Transformada de Fourier:** se aplica la Transformada de Fourier a cada uno de los segmentos obtenidos en la etapa anterior, permitiendo analizar la composición espectral en cada instante de tiempo.
3. **Generación del espectrograma:** los espectros resultantes se ensamblan para construir el espectrograma, una representación visual que muestra la evolución de las amplitudes de las diferentes frecuencias a lo largo del tiempo.

La STFT de una señal  $x(t)$  se define matemáticamente como:

$$STFT\{x(t)\}(\tau, \omega) = \int_{-\infty}^{\infty} x(t) \cdot \omega(t - \tau) \cdot e^{-j\omega \cdot t} dt$$

- $x(t)$  es la señal original.
- $\omega(t - \tau)$  es una función ventana centrada en  $\tau$ .
- $e^{-j\omega \cdot t}$  es el núcleo de la Transformada de Fourier.
- $\omega$  representa la frecuencia en el dominio transformado.



**Figura 60.** Ejemplo de la STFT de una señal.

La **selección del tamaño de la ventana** en la STFT implica un compromiso entre la resolución temporal y la resolución en frecuencia:

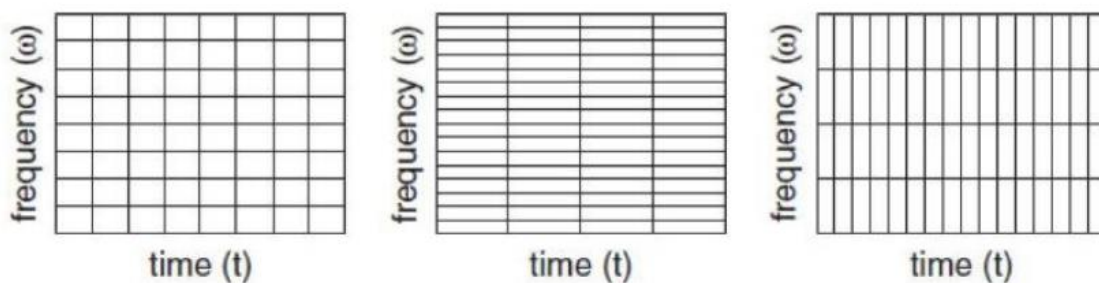
- **Ventana estrecha:** se utilizan ventanas cortas cuando es necesario **detectar cambios rápidos en la señal**. Proporcionan una **alta resolución temporal**, lo que permite distinguir eventos muy cercanos en el tiempo. Sin embargo, al reducir la longitud de la ventana, se **pierde precisión en la identificación de frecuencias** específicas.
- **Ventana ancha:** se emplean ventanas más largas para mejorar la **precisión en la identificación de frecuencias**. Al abarcar más ciclos de la señal, permiten diferenciar con mayor exactitud frecuencias cercanas. No

obstante, el aumento en la longitud de la ventana **disminuye la capacidad de detectar eventos breves en el tiempo**.

El **Principio de Incertidumbre de Gabor** establece que existe una limitación fundamental en la resolución simultánea de tiempo y frecuencia:

- Si se mejora la resolución temporal utilizando ventanas más cortas, se reduce la precisión en la identificación de frecuencias.
- Si se mejora la resolución en frecuencia con ventanas más largas, se pierde precisión en el tiempo.

Por ello, la elección del tamaño de la ventana debe basarse en los requerimientos específicos del análisis, priorizando la resolución en tiempo o en frecuencia según la aplicación.



**Figura 61.** Diferencia en las STFT dependiendo del tipo de ventana utilizada.

La Transformada de Fourier de Tiempo Corto (STFT) es un proceso matemáticamente invertible, lo que permite recuperar la señal original mediante la **Transformada Inversa de STFT (ISTFT)**. El proceso de inversión es el siguiente:

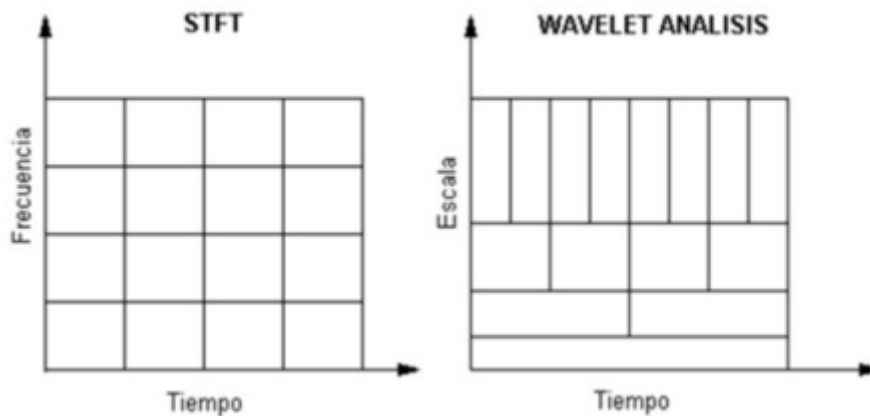
1. Se toman los coeficientes de la STFT correspondientes a cada ventana en el dominio tiempo-frecuencia.
2. Se aplica la **Transformada de Fourier Inversa** a cada uno de los segmentos obtenidos.
3. Cada segmento reconstruido se multiplica por la misma ventana utilizada en la STFT.
4. Finalmente, los segmentos se combinan mediante el método **overlap-add**, superponiéndolos para obtener la señal final en el dominio del tiempo.

Para lograr una reconstrucción precisa, es fundamental elegir adecuadamente el tipo de ventana y el grado de solapamiento. Además, se debe cumplir la **condición de suma constante (COLA)**, que garantiza que la suma de las ventanas superpuestas sea uniforme en el tiempo, evitando distorsiones en la señal reconstruida.

### 13.2. WAVELETS

El análisis **Wavelet** representa el paso lógico siguiente a la STFT; es una técnica que, mediante **ventanas con regiones de tamaño variable**, permite:

- El uso de intervalos grandes de tiempo en aquellos segmentos en los que se requiere mayor precisión en baja frecuencia.
- Regiones más pequeñas donde se requiere información en alta frecuencia.



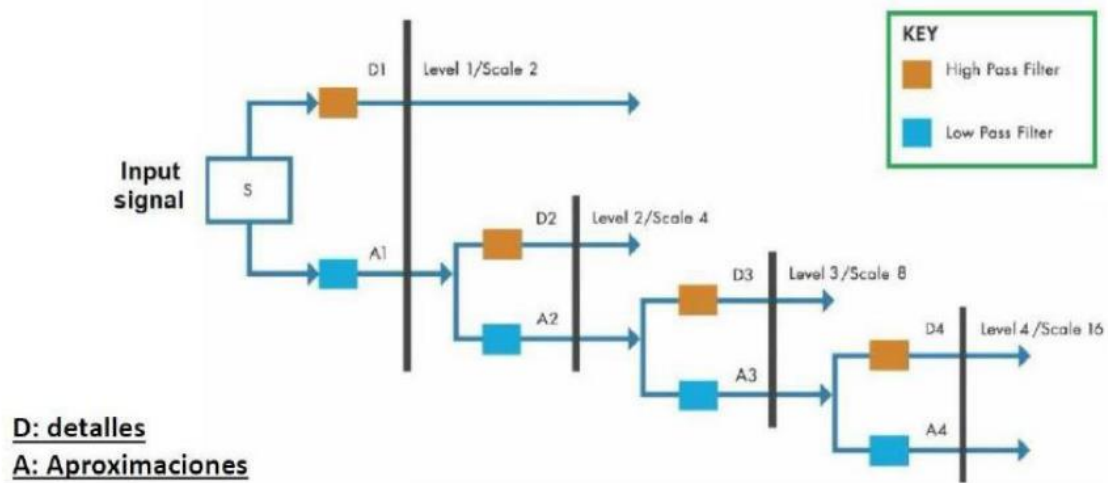
**Figura 62.** Comparativa de esquema entre STFT y análisis con Wavelets.

#### 13.2.1. Transformada de Wavelet Discreta (DWT)

La **Transformada Wavelet Discreta (DWT)** es un método de análisis de señales que descompone la señal en diferentes niveles de detalle y aproximación. Su proceso consta de cuatro pasos principales:

1. **Filtrado de paso bajo (aproximaciones, A):** se aplica un filtro de paso bajo para retener las componentes de baja frecuencia, que contienen la información principal de la señal.
2. **Filtrado de paso alto (detalles, D):** se aplica un filtro de paso alto para extraer las componentes de alta frecuencia, resaltando detalles y bordes de la señal.
3. **Submuestreo:** las señales resultantes de ambos filtros se submuestran, reduciendo su tamaño a la mitad para optimizar el procesamiento.
4. **Iteración del proceso:** la descomposición se repite sobre la señal de aproximación obtenida en el paso 1, generando niveles sucesivos de análisis con mayor detalle y precisión.

Este proceso jerárquico permite representar la señal en distintos niveles de resolución, facilitando aplicaciones como compresión, eliminación de ruido y análisis de características en el dominio de la frecuencia.



**Figura 63.** Esquema de la Transformada de Wavelet Discreta (DWT).

### 13.2.2. Transformada de Wavelet Continua (CWT)

La **Transformada Wavelet Continua (CWT)** es una técnica de análisis de señales que permite descomponer una señal en función de diferentes escalas y traslaciones, proporcionando una representación detallada de su evolución en el tiempo y la frecuencia.

A diferencia de la Transformada Wavelet Discreta (DWT), que trabaja con un conjunto discreto de escalas y posiciones, la CWT emplea una **función wavelet continua**, lo que permite analizar la contribución de distintas frecuencias a lo largo del tiempo con mayor precisión.

La CWT de una señal  $x(t)$  se define matemáticamente como:

$$C(a, b) = \int_{-\infty}^{\infty} x(t) \cdot \frac{1}{\sqrt{|a|}} \cdot \psi \left( \frac{t-b}{a} \right) dt$$

- $x(t)$  es la señal original.
- $\psi(t)$  es la función de wavelet madre, que se usa como base para analizar la señal.
- $a$  es el **parámetro de escala** ( $a > 0$ ) que determina el ancho de la wavelet.
  - Valores pequeños de  $a$  comprimen la wavelet, permitiendo detectar detalles finos en la señal.

- Valores grandes de  $a$  expanden la wavelet, capturando componentes de baja frecuencia.
- $b$  es el **parámetro de traslación** que determina la posición de la wavelet en el tiempo.
  - Al variar  $b$ , la wavelet se desplaza a lo largo de la señal, permitiendo analizar distintas secciones

### 13.2.3. Wavelet de Haar

La **wavelet de Haar** es una de las **wavelets más simples** y es ampliamente utilizada en aplicaciones iniciales de análisis wavelet debido a su facilidad de implementación y eficiencia computacional.

Su función madre  $\psi(t)$  se define de como:

$$\psi(t) = \begin{cases} 1 & \text{si } 0 \leq t < \frac{1}{2} \\ -1 & \text{si } \frac{1}{2} \leq t < 1 \\ 0 & \text{en otro caso} \end{cases}$$

### 13.2.4. Wavelet de Morlet

La **wavelet de Morlet** se caracteriza por su capacidad para **representar patrones oscilatorios**, lo que la hace ideal para detectar **texturas, bordes y estructuras repetitivas** en una imagen. Combina una **onda sinusoidal** con una **ventana gaussiana**, lo que le permite enfocarse en diferentes **frecuencias** sin perder la localización en el tiempo o en la imagen.

Su función madre  $\psi(t)$  se define de como:

$$\psi(t) = \pi^{-\frac{1}{4}} \cdot e^{j \cdot \omega_0 \cdot t} \cdot e^{-\frac{t^2}{2}}$$

### 13.2.5. Análisis Multi-resolución (MRA) con wavelets

El **Análisis Multi-resolución (MRA) con wavelets** es una técnica que permite descomponer una señal en distintos niveles de resolución, separando sus componentes de frecuencia. El MRA descompone la señal en varios **niveles de detalle**  $D_k$  y una **aproximación** final. Cada nivel corresponde a una banda de frecuencia específica dada por:

$$\left[ \frac{F_S}{2^{k+1}}, \frac{F_S}{2^k} \right]$$

- $F_s$  es la frecuencia de muestreo.
- **Niveles altos:** detalles de alta frecuencia.
- **Niveles intermedios:** frecuencias medias.
- **Niveles bajos:** frecuencias bajas (tendencia).
- **Último nivel:** aproximación global (tendencia a largo plazo).

### 13.3. FILTROS DE GABOR

Los filtros de Gabor están diseñados para imitar las células de la **corteza visual** de algunos mamíferos, lo que les permite extraer características en diferentes orientaciones y escalas. Son altamente efectivos para **resaltar detalles y patrones** en imágenes, siendo fundamentales en aplicaciones como el reconocimiento de objetos, clasificación de texturas, detección de bordes y análisis de imágenes médicas. Su **sensibilidad a la textura y orientación** les permite adaptarse a diversas características visuales, mientras que su **naturaleza multiescala** permite capturar información en distintos niveles de detalle.

Además, al ajustar el parámetro de frecuencia espacial  $f$ , los filtros de Gabor pueden afinarse para responder selectivamente a las frecuencias específicas presentes en la imagen.

Un filtro de Gabor en 2D consiste en una combinación de una onda sinusoidal y un filtro gaussiano:

$$G_{real}(x, y) = e^{-\frac{1}{2} \left( \frac{x'^2 + \gamma \cdot y'^2}{\sigma^2} \right)} \cdot \cos(2\pi \cdot f \cdot x')$$

$$G_{imag}(x, y) = e^{-\frac{1}{2} \left( \frac{x'^2 + \gamma \cdot y'^2}{\sigma^2} \right)} \cdot \sin(2\pi \cdot f \cdot x')$$

- $x'$  e  $y'$  son las coordenadas transformadas.  $\theta$  es el ángulo de orientación del filtro de Gabor.

$$\begin{cases} x' = x \cdot \cos(\theta) + y \cdot \sin(\theta) \\ y' = -x \cdot \sin(\theta) + y \cdot \cos(\theta) \end{cases}$$

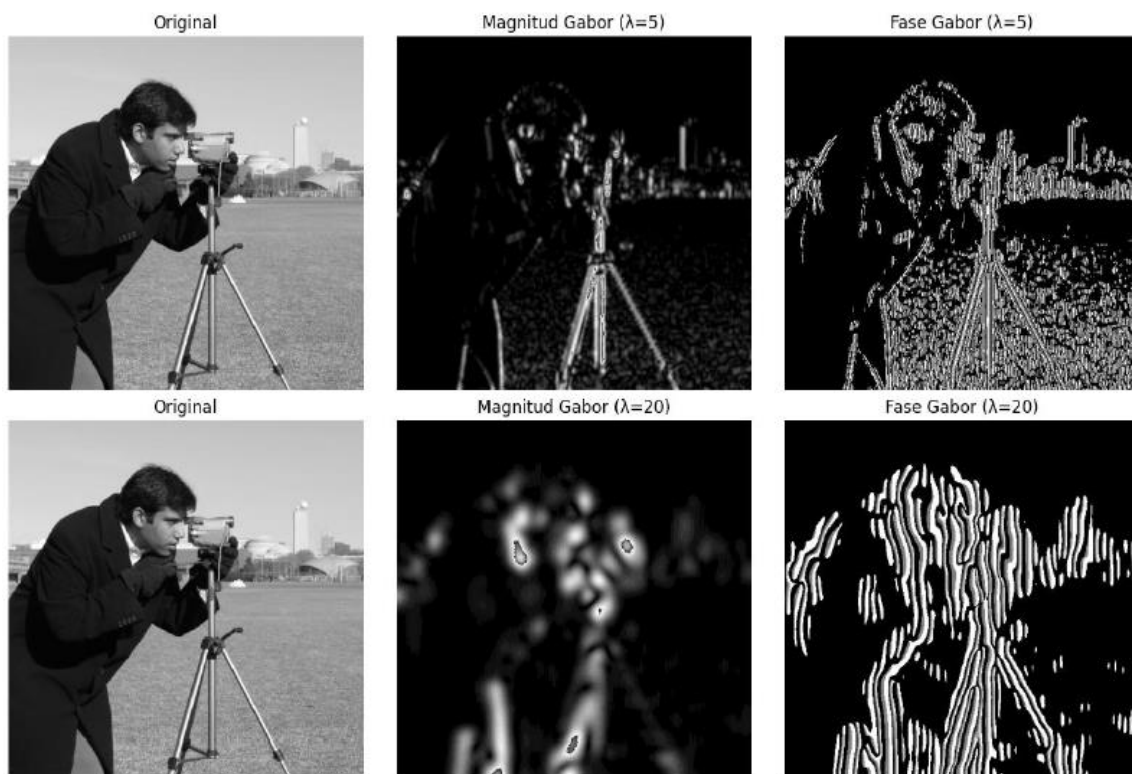
- $\sigma$  es la desviación estándar (anchura de la función gaussiana).
- $f = \frac{1}{\lambda}$  es la frecuencia de la componente sinusoidal.
- $\gamma$  es la relación aspecto del filtro (controla la elongación del filtro).



Una **longitud de onda** ( $\lambda$ ) más corta resulta en una **mayor sensibilidad a los detalles finos** de la imagen. Sin embargo, una **longitud de onda más larga** implica una menor sensibilidad a los detalles finos, pero una **mejor capacidad de capturar patrones más gruesos o texturas más grandes** en la imagen.

El filtro de Gabor tiene una **parte real** y una **parte imaginaria**, por lo que se puede calcular su **magnitud** y **fase** de la misma forma que la explicada durante la Transformada de Fourier.

La **magnitud** indica la intensidad de la respuesta del filtro en cada punto de la imagen, siendo útil para la **detección de texturas y bordes**. Por otro lado, la **fase** proporciona información sobre la **orientación y la estructura de los patrones** presentes en la imagen.



**Figura 64.** Efecto del filtro de Gabor en una imagen con diferentes longitudes de onda.

#### 13.4. SCALE INVARIANT FEATURE TRANSFORM (SIFT)

**Scale Invariant Feature Transform (SIFT)** es una técnica utilizada para detectar y describir puntos de interés (*features*) en imágenes de manera que estos puntos sean **invariantes a cambios de escala, rotación, e incluso iluminación o ruido**. Esta propiedad **permite encontrar coincidencias** entre imágenes tomadas desde diferentes perspectivas o bajo distintas condiciones de luz.

El proceso de la Transformada SIFT es el siguiente:

### 1. Detección de puntos clave:

- Construcción de la pirámide de escalas: se generan versiones de la imagen a distintas escalas para detectar puntos de interés robustos en diferentes niveles de resolución.
- Detección de Extremos (*Keypoint Detection*): se identifican los puntos donde la imagen presenta máximos o mínimos locales a través de la pirámide de escalas. Esto permite que los puntos clave sean invariantes a la escala de la imagen.

2. **Asignación de orientación**: se calcula la dirección principal de cada punto de interés para hacerlo invariante a rotaciones. Esto asegura que los puntos clave se mantengan consistentes, independientemente de la orientación de la imagen.

3. **Cálculo del descriptor**: se genera un vector descriptor que describe la vecindad de cada punto de interés, utilizando los gradientes de intensidad. Este descriptor facilita la comparación entre puntos de interés de diferentes imágenes, permitiendo encontrar coincidencias incluso si las imágenes varían en escala, rotación o iluminación.

Este proceso hace que SIFT sea una herramienta poderosa para la correspondencia de puntos en tareas como el reconocimiento de objetos, la reconstrucción 3D y la estimación de movimiento en visión por computador.



**Figura 65.** Efecto de aplicar SIFT sobre una imagen.

## TEMA 14. DECISIÓN. PRINCIPIOS E IMPLEMENTACIÓN DE ALGORITMOS DE AYUDA EN LA TOMA DE DECISIONES

El **sistema de reconocimiento de patrones** se basa en inferir diferencias entre categorías de objetos a partir de ejemplos. Su desarrollo implica un proceso de optimización guiado por un algoritmo iterativo y una métrica de evaluación. La implementación del sistema de decisión se fundamenta en estos principios, que básicamente corresponden a **Machine Learning**.

A partir de las señales o imágenes, se extraen las características relevantes que describen sus formas y patrones. Estas características se organizan en un **vector de características**, formando un **dataset** donde cada ejemplo (señal o imagen) está representado por un conjunto de atributos derivados de su contenido. Dichas características son esenciales para **entrenar modelos de aprendizaje automático**, tanto supervisados como no supervisados, utilizando un conjunto de entrenamiento.

En el **aprendizaje supervisado**, el modelo aprende a partir de ejemplos etiquetados, mientras que en el **aprendizaje no supervisado**, el modelo identifica patrones sin necesidad de etiquetas. En ambos casos, las características extraídas de las señales o imágenes son la base sobre la que el modelo realiza predicciones o agrupaciones.

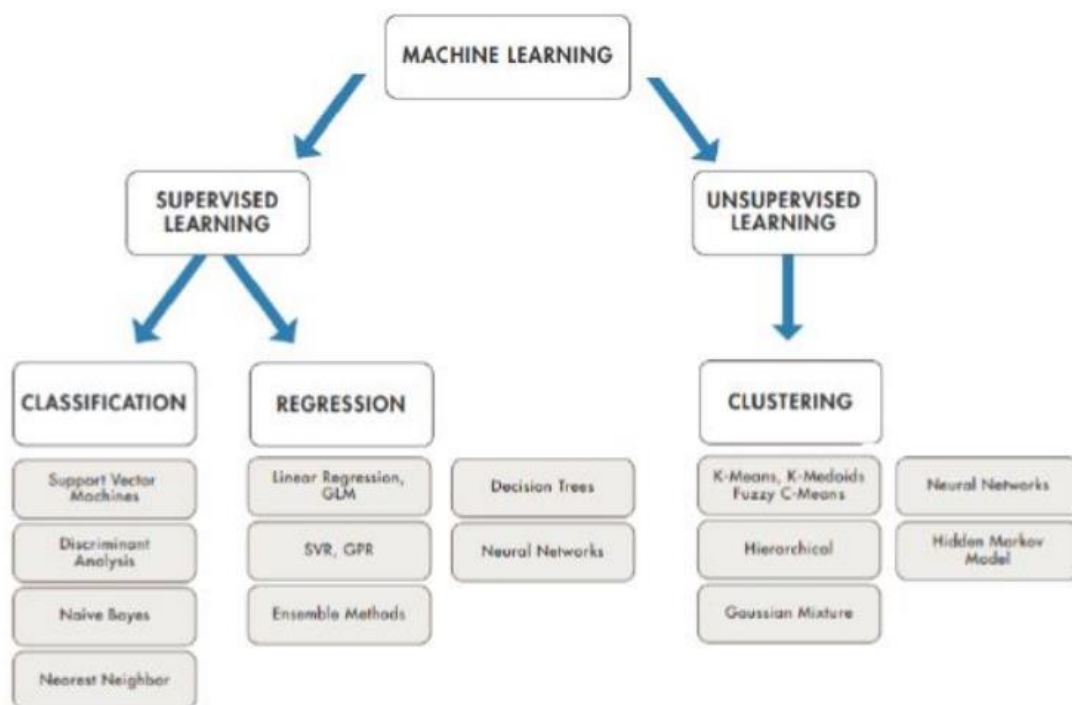


Figura 66. Esquema del *Machine Learning*.

---

### 14.1. APRENDIZAJE SUPERVISADO

En el **aprendizaje supervisado**, los algoritmos se entrenan con un **conjunto de datos etiquetados (conjunto de entrenamiento)** que contiene entradas emparejadas con sus salidas deseadas.

El **propósito** es que **el modelo aprenda a establecer una relación entre las entradas y las salidas correctas**, lo que permite predecir con precisión las salidas para nuevas entradas.

Dependiendo del tipo de problema, se puede tratar de un **problema de regresión**, que busca predecir un valor numérico continuo, o de **clasificación**, que tiene como objetivo predecir una clase dentro de un conjunto de posibles categorías.

Algunos de los algoritmos utilizados para abordar estos problemas incluyen **SVM, Decision Trees, Random Forest**, entre otros.

---

### 14.2. APRENDIZAJE NO SUPERVISADO

En el **aprendizaje no supervisado**, los algoritmos exploran la estructura inherente de un conjunto de datos sin la guía de salidas deseadas predefinidas.

En lugar de aprender a predecir una salida específica, **el modelo busca descubrir patrones, relaciones o agrupaciones** entre los datos lo que permite revelar información oculta o desconocida previamente.

Dependiendo del tipo de problema, se puede tratar de un **problema de agrupamiento**, que busca encontrar patrones en los datos mediante la relación de grupos de datos con similitud entre sí, o de **detección de anomalías**, que tiene como objetivo encontrar datos que no siguen patrones establecidos en el resto del conjunto de datos

Algunos de los algoritmos utilizados para abordar estos problemas incluyen **K-Means, HMM**, entre otros.

---

### 14.3. EQUILIBRIO ENTRE SESGO Y VARIANZA

Cuando se evalúa un modelo de aprendizaje automático, generalmente se busca un equilibrio entre la capacidad del modelo para ajustar a los datos de entrenamiento (**sesgo**) y su capacidad para generalizar bien a datos no vistos (**varianza**).

- **Sesgo (bias):** se refiere a la capacidad para capturar la verdadera relación entre las variables de entrada y la variable objetivo.

- **Sesgo alto:** indica una simplificación excesiva de esta relación, lo que resulta en un rendimiento deficiente incluso en los datos de entrenamiento.
- **Sesgo bajo:** sugiere una captura precisa de la complejidad de los datos de entrenamiento, pero puede inducir al sobreajuste, comprometiendo la capacidad del modelo para generalizar correctamente a nuevos datos.
- **Varianza:** se refiere a la sensibilidad del modelo a las pequeñas variaciones en los datos de entrenamiento.
  - **Varianza alta:** puede llevar al modelo a ajustarse demasiado a los datos de entrenamiento, incluso capturando el ruido presente en ellos. Esto puede resultar en un rendimiento deficiente al generalizar a datos no vistos, ya que el modelo no puede generalizar más allá de los datos de entrenamiento.
  - **Varianza baja:** puede llevar a un ajuste insuficiente, resultando en un rendimiento bajo en los datos de entrenamiento, aunque el modelo podría generalizar bien a datos no vistos.



Figura 67. Efecto de la varianza y el sesgo en las predicciones.

#### 14.4. CURSE OF DIMENSIONALITY

El **curse of dimensionality** se refiere al fenómeno en el que el **rendimiento de los algoritmos de aprendizaje automático empeora a medida que aumenta el número de características** (dimensiones) de los datos. A medida que crece la dimensionalidad, los datos se dispersan más, lo que hace más difícil encontrar patrones y puede llevar a un sobreajuste o mayor complejidad computacional.

## TEMA 15. APLICACIONES ACTUALES DEL TRATAMIENTO DE LA SEÑAL

## 15.1. BIOMETRÍA

La **biometría** es un conjunto de técnicas de reconocimiento de patrones que permiten identificar y verificar a una persona de forma única. La **seguridad de la información** se basa en **tres pilares**:

- **Lo que sabes:** contraseñas y claves (usuario y contraseña, sistemas de clave pública y privada).
- **Lo que tienes:** tokens como mensajes SMS o tarjetas de coordenadas.
- **Lo que eres:** identificación mediante rasgos biométricos (huella digital, iris, retina).

Los sistemas biométricos tienen los siguientes **usos**:

- **Verificación:** se compara la muestra biométrica con el patrón almacenado para confirmar la identidad.
- **Identificación:** se compara la muestra biométrica con todos los patrones almacenados para determinar quién es el usuario.



**Figura 68.** Rasgos físicos más comunes empleados en el reconocimiento biométrico.

El **proceso habitual de registro** (*enrollment*) en estos sistemas sigue los siguientes **pasos**:

1. **Captura de la información** biométrica mediante sensores.
2. **Extracción de características** para generar un patrón.
3. **Almacenamiento** en una base de datos.

Las técnicas más utilizadas incluyen **reconocimiento facial y biometría de iris**, con algoritmos avanzados de segmentación y extracción de características.

---

### 15.1.1 Reconocimiento facial

El **reconocimiento facial** se basa en características geométricas del rostro, como la distancia entre ojos o la proporción de la nariz.

Los **procesos clave** que se siguen son los siguientes:

1. **Segmentación:** se suele emplear detección de color de piel. Métodos más avanzados incluyen sensores infrarrojos o de temperatura.
2. **Extracción de características:** el método más utilizado es **eigenfaces**, basado en Análisis de Componentes Principales (PCA), que permite descomponer la imagen en vectores propios para mejorar la precisión con un coste computacional razonable.



**Figura 69.** Ejemplo visual de eigenfaces.

La **ventaja** del reconocimiento facial es que es poco invasivo y el usuario no necesita colaborar activamente. Sin embargo, como **desventaja**, factores externos como iluminación, uso de gafas, barba o heridas pueden afectar la precisión del reconocimiento.

---

### 15.1.2. Reconocimiento de iris

El reconocimiento de iris es uno de los métodos más precisos debido a la unicidad de la textura del iris, incluso entre los dos ojos de una misma persona.

Los **procesos clave** que se siguen son los siguientes:

1. **Segmentación:**
  - Basada en color: poco precisa debido a variaciones de iluminación.
  - Basada en forma: detecta zonas circulares en el ojo, utilizando algoritmos de crecimiento de regiones.
  - Basada en textura: más utilizada, especialmente con sensores infrarrojos.

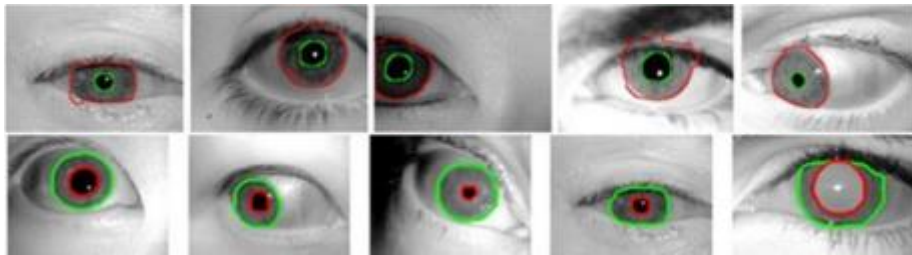


## 2. Extracción de características:

- Se utilizan **filtros de Gabor**, junto con otros métodos como **Markov Random Fields, Wavelets y DCT**.
- Se aplica una transformación de coordenadas polares a cartesianas para facilitar la comparación.

## 3. Comparación y Precisión:

- Métodos simples como **distancia euclídea** o **distancia de Hamming** hacen que la verificación sea rápida y eficiente.
- La segmentación con cámaras infrarrojas requiere enfoques avanzados basados en forma o textura.



**Figura 70.** Ejemplo de segmentación de iris y pupila.

---

### 15.1.3. Otras técnicas biométricas

A continuación, se presenta un resumen breve de otras técnicas biométricas con los algoritmos de segmentación y extracción de características más comunes:

- **Huella digital:** sin duda, una de las técnicas biométricas más empleadas, y no por ella la más sencilla.
  - Sistemas de captura: suelen ser sistemas de infrarrojos o escáneres.
  - Segmentación: muy sencilla, ya que se delega en el sensor toda la dificultad de la captura de la huella. Suele tratarse de una segmentación basada en *clustering* de dos clases junto con morfología matemática para eliminar errores.
  - Extracción de características: propiedades geométricas de los patrones de las líneas de la huella. Dichas propiedades son muy similares a los algoritmos SIFT.
- **Palmprint:** esta técnica consiste en leer las líneas de la palma de la mano, no solo las más visibles, también la textura; similar a la huella del dedo, pero en la palma de la mano.



- Sistemas de captura: muy variados, aunque los más usados son los que emplean fotografías infrarrojas.
- Segmentación: más compleja que en la huella; en este caso hay que identificar correctamente la palma de la mano y hacer una segmentación basada en intensidad.
- Extracción de características: muy similares a la de la huella digital.
- **Geometría de mano**: esta técnica permite identificar a las personas en función de la forma de su mano.
  - Sistemas de captura: muy sencillos, bastaría con una cámara del móvil.
  - Segmentación: cuanto más sencillo sea el dispositivo de captura, más compleja será la segmentación. En este caso, la segmentación suele conllevar agregación multiescala, segmentación basada en color o crecimiento de regiones.
  - Extracción de características: está orientada a medidas relativas de los dedos, mano y muñeca.
- **Venas de la mano y retina**:
  - Sistema de captura: infrarrojo obligatoriamente para poder ver a través de la piel.
  - Segmentación: basada en color y en intensidades.
  - Extracción de características: suele usarse filtros de Gabor y algoritmos que simulan redes comparando las venas a redes de nodos.
- **Firma**:
  - Sistema de captura: muy especializado; o bien un sensor con un bolígrafo específico (*online*), o bien una fotografía de una firma (*offline*).
  - Segmentación: en el caso *online* no hay segmentación. En el *offline*, la segmentación es muy sencilla: binarización de la imagen.
  - Extracción de características: se suelen utilizar métodos basados en parametrización de curvas y propiedades parecidas a SIFT.

---

## 15.2. COCHES AUTÓNOMOS

Los **coches autónomos** están equipados con sensores avanzados que superan la capacidad humana para detectar y procesar información. La **percepción computacional** en estos vehículos sigue **tres fases**:

- **Captura de información:** utilizan sensores como **infrarrojos, radares, cámaras, acelerómetros y posicionamiento GPS**. Futuras implementaciones podrían incluir **drones para mejorar la percepción del tráfico**.
- **Procesamiento de información:** un **computador a bordo**, muchas veces ubicado en el maletero, se encarga del procesamiento en tiempo real. La sincronización de datos es crucial para evitar errores en la toma de decisiones. Se almacena toda la información procesada para auditorías en caso de accidentes. Maneja distintos tipos de datos:
  - Unidimensionales: sensores de humedad y temperatura.
  - Bidimensionales: imágenes y vídeos.
  - Tridimensionales: radares para detectar profundidad.

Los algoritmos empleados son los siguientes:

- Segmentación: separación de fondo, detección de coches y objetos en la vía.
- Detección de bordes: identificación de líneas y límites de la carretera.
- Extracción de características:
  - **Wavelets y filtros de Gabor** para detectar objetos en movimiento.
  - **SIFT** para identificar señales de tráfico.
- Corrección de ruido: para evitar errores en la detección de señales bajo condiciones adversas.
- **Toma de decisiones y ética:** actualmente, ante situaciones anómalas, el coche avisa al conductor. Quedan cuestiones pendientes como la **responsabilidad en accidentes o multas**, y dilemas éticos sobre **a quién priorizar en situaciones de emergencia**.

---

### 15.3. RETOS Y LIMITACIONES DE LA PERCEPCIÓN COMPUTACIONAL

Los **sistemas de percepción computacional** aún tienen mucho margen de mejora, ya que los algoritmos actuales requieren alta especificidad, perdiendo su capacidad de generalización. Técnicas como el *Deep Learning* han acercado la meta de lograr una percepción más flexible, pero aún hay desafíos pendientes.

Algunos **retos clave** que se presentan son los siguientes:

- **Sesgos:** riesgo de que los algoritmos reproduzcan los sesgos humanos.
- **Ética:** ¿hasta qué punto se puede confiar en la toma de decisiones de una máquina?
- **Falta de generalidad:** ¿es posible crear un algoritmo capaz de entender imágenes sin depender de múltiples clasificadores predefinidos?
- **Arte:** si estos sistemas pueden percibir imágenes, ¿podrían también crearlas de manera original?

A pesar de estos desafíos, la percepción computacional sigue evolucionando con gran interés y avances constantes.