

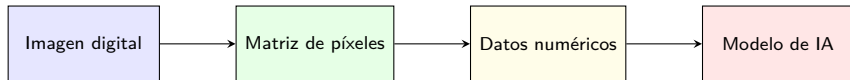
Procesado de Imagen Digital: Representación y Transformaciones

Antonio Falcó

Máster en IA y Fabricación Aditiva

Lectura 1 ampliada

De píxel a modelo de IA



Idea clave

Una imagen es un conjunto de datos que puede ser procesado, transformado y aprendido por un modelo de IA.

¿Qué es una imagen digital?

- ▶ Una imagen digital es una matriz de valores numéricos que representan la intensidad de luz (grises) o color (RGB).
- ▶ Cada elemento de la matriz es un **píxel** (*picture element*).
- ▶ Resolución: número de píxeles horizontales y verticales ($H \times W$).
- ▶ Profundidad de color: número de bits por píxel ($8 \text{ bits} \Rightarrow 256 = 2^8$ niveles).

34	50	100	120	120	130	165
23	55	110	105	140	140	170
50	75	110	112	138	138	178
90	95	140	160	180	180	200
110	115	150	175	175	210	220
145	175	185	195	230	230	240
145	175	185	195	230	240	

Representación matricial de imágenes

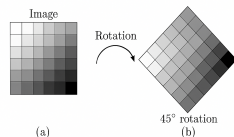
- ▶ Imagen en escala de grises:
 $I \in \mathbb{R}^{H \times W}$.
- ▶ Imagen RGB: $I \in \mathbb{R}^{H \times W \times 3}$.
- ▶ Canales: Rojo, Verde, Azul.
- ▶ Formatos: HWC (por canal) o CHW (por tensor).

Matriz de píxeles

		200		
				60
120				

Transformaciones geométricas básicas

- ▶ **Traslación:** mover la imagen en el plano.
- ▶ **Rotación:** $I'(x, y) = I(x \cos \theta - y \sin \theta, x \sin \theta + y \cos \theta)$.
- ▶ **Escalado:** cambiar el tamaño mediante interpolación.
- ▶ **Inversión:** reflejo horizontal/vertical.



Normalización de imágenes

Objetivo: llevar los valores de intensidad a un rango estándar para mejorar la estabilidad numérica y la comparabilidad entre imágenes.

1. Normalización lineal

$$I' = \frac{I - \min(I)}{\max(I) - \min(I)} \Rightarrow I' \in [0, 1]$$

- ▶ Se aplica canal a canal (R, G, B).
- ▶ Preserva el contraste relativo.
- ▶ Útil para visualizar y preparar datos antes de alimentar un modelo.

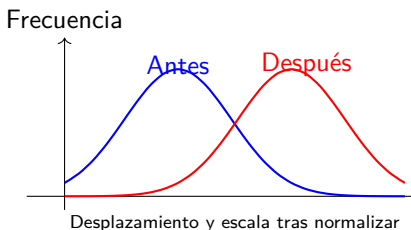
2. Estandarización estadística (z-score)

$$I' = \frac{I - \mu}{\sigma}$$

- ▶ Centra la distribución en media cero y varianza unitaria.
- ▶ Recomendado para redes profundas (CNN, autoencoders).
- ▶ Mejora la convergencia del descenso del gradiente.

Efectos visuales:

- ▶ En redes neuronales, evita gradientes explosivos o muertos.
- ▶ Homogeneiza la escala de entrada entre imágenes o lotes.
- ▶ Facilita la interpretación de activaciones.



Resumen

Normalizar no cambia la información visual, pero **mejora la estabilidad numérica y la capacidad de generalización** de los modelos de IA.

Conversión entre espacios: RGB \leftrightarrow HSV

Suposición: $R, G, B \in [0, 1]$ (si vienen en $[0, 255]$, dividir por 255).

$$C_{\max} = \max(R, G, B), \quad C_{\min} = \min(R, G, B), \quad \Delta = C_{\max} - C_{\min}.$$

Definición de HSV:

$$V = C_{\max}, \quad S = \begin{cases} 0, & C_{\max} = 0 \\ \Delta / C_{\max}, & \text{en otro caso} \end{cases}$$

$$H = \begin{cases} 60^\circ \cdot \frac{G-B}{\Delta} \text{ mód } 360^\circ, & C_{\max} = R \\ 60^\circ \cdot \left(\frac{B-R}{\Delta} + 2 \right), & C_{\max} = G \\ 60^\circ \cdot \left(\frac{R-G}{\Delta} + 4 \right), & C_{\max} = B \end{cases}$$

Notas:

- ▶ H (tono) define el color perceptual (0° =rojo, 120° =verde, 240° =azul).
- ▶ S mide la saturación y V el brillo o valor máximo.
- ▶ Útil en segmentación por color o detección de regiones homogéneas.



Conversión inversa (esquema):

$$\begin{aligned} C &= V \cdot S, \\ X &= C \cdot (1 - |((H/60) \text{ mód } 2) - 1|), \\ m &= V - C. \end{aligned}$$

Asignar (R', G', B') según el sector de H y sumar m a cada canal.

De RGB a YCbCr y CIE Lab

Corrección gamma: muchos dispositivos usan *sRGB*, que no es lineal. Para cálculos fotométricos, se usa:

$$R_{\text{lin}} = f_{\gamma}^{-1}(R), \quad G_{\text{lin}} = f_{\gamma}^{-1}(G), \quad B_{\text{lin}} = f_{\gamma}^{-1}(B).$$

RGB \rightarrow **YCbCr** (BT.601, valores en $[0, 1]$):

$$Y = 0,299 R + 0,587 G + 0,114 B,$$

$$Cb = \frac{1}{2} \frac{B - Y}{1 - 0,114}, \quad Cr = \frac{1}{2} \frac{R - Y}{1 - 0,299}.$$

Uso: compresión de vídeo o imagen; separa luminancia (Y) y crominancia (Cb, Cr).

RGB \rightarrow **CIE Lab** (resumen):

1. Convertir a espacio **XYZ** (D65 típico).
2. Aplicar función $f(t)$ con $\delta = 6/29$:

$$f(t) = \begin{cases} t^{1/3}, & t > \delta^3 \\ \frac{t}{3\delta^2} + \frac{4}{29}, & \text{en otro caso.} \end{cases}$$

3. Calcular:

$$L^* = 116 f(Y/Y_n) - 16, \quad a^* = 500[f(X/X_n) - f(Y/Y_n)], \quad b^* = 200[f(Y/Y_n) - f(Z/Z_n)].$$

Ejemplo numérico: RGB \rightarrow HSV y YCbCr

Color de partida: $(R, G, B) = (255, 128, 0)$. Normalizamos a $[0, 1]$:

$$R = 1, \quad G = \frac{128}{255} \approx 0,502, \quad B = 0.$$

RGB \rightarrow HSV (con $C_{\max} = \max(R, G, B)$, $C_{\min} = \min(R, G, B)$, $\Delta = C_{\max} - C_{\min}$):

$$C_{\max} = 1, \quad C_{\min} = 0, \quad \Delta = 1.$$

$$V = C_{\max} = 1, \quad S = \frac{\Delta}{C_{\max}} = \frac{1}{1} = 1.$$

Como $C_{\max} = R$,

$$H = 60^\circ \cdot \left(\frac{G - B}{\Delta} \right) \bmod 360^\circ = 60^\circ \cdot (0,502 - 0) \approx 30,1^\circ.$$

$$\Rightarrow \text{HSV} \approx (H, S, V) = (30^\circ, 1, 1).$$

RGB \rightarrow YCbCr (BT.601 aproximado en $[0, 1]$):

$$Y = 0,299 R + 0,587 G + 0,114 B \approx 0,299 + 0,587 \cdot 0,502 + 0 \approx 0,594.$$

$$Cb = \frac{1}{2} \frac{B - Y}{1 - 0,114} = \frac{1}{2} \frac{0 - 0,594}{0,886} \approx -0,335, \quad Cr = \frac{1}{2} \frac{R - Y}{1 - 0,299} = \frac{1}{2} \frac{1 - 0,594}{0,701} \approx 0,290.$$

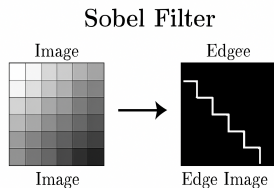
$$\Rightarrow \text{YCbCr} \approx (Y, Cb, Cr) = (0,594, -0,335, 0,290).$$

Espacio	Valores (aprox.)
RGB (normalizado)	(1, 0,502, 0)
HSV	(30°, 1, 1)
YCbCr	(0,594, -0,335, 0,290)

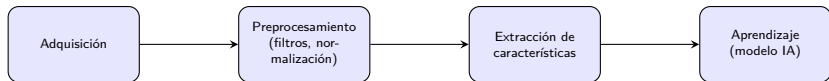
Nota: Las ligeras diferencias dependen del redondeo y de la definición exacta/estándar usada (p.ej.,

Filtros y convolución

- ▶ Un filtro es una operación local que modifica cada píxel según sus píxeles vecinos.
- ▶ **Kernel:** matriz pequeña (3x3, 5x5) con coeficientes.
- ▶ **Operación de convolución:**
$$(I * K)(x, y) = \sum_{i,j} I(x - i, y - j) K(i, j).$$
- ▶ Ejemplos: filtro de media, Sobel (bordes).



De la imagen al modelo de IA



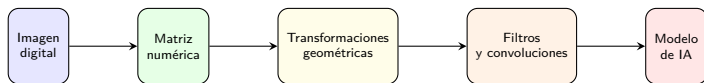
Objetivo

Entender cómo los datos visuales se transforman en información útil mediante modelos de aprendizaje automático.

Conclusión y práctica en Colab

- ▶ Repasar los conceptos vistos (slicing, inversión, filtrado, normalización).
- ▶ Abrir el cuaderno: `IA_intro_colab_Lecture1.ipynb`.
- ▶ Modificar parámetros en bloques `*TRY IT*` y observar el efecto.
- ▶ Ejercicios propuestos:
 1. Visualizar una imagen y sus canales.
 2. Implementar un filtro de media 3x3.
 3. Comparar filtrado Sobel horizontal y vertical.

Resumen visual de la sesión



Este flujo resume la transición desde la representación numérica hasta la comprensión visual basada en IA. Cada etapa será explorada en profundidad a lo largo del curso.