Rodolfo Ferro
ferro@cimat.mx
https://rodolfoferro.xyz

BeePy 5.0

Mayo, 2026

Imágenes como datos

Redes neuronales y visión por computadora

Tabla de contenidos

Rodolfo Ferro (ferro@cimat.mx)

- Sr. Machine Learning | Computer Vision Engineer @ AIAEC
- Facilitador DS/AI @ DECI ENES León (UNAM) & EdgeHub School of Innovation.
- **Formación:** Esp. en Métodos Estadísticos (c/enf. en DS) @ CIMAT, BMath (UG), CSysEng (UVEG)
- **Experiencia:** Sr. SWE @ Bisconic (US), ML Engineer @ Vindoo.ai (España), Sherpa Digital de IA @ Microsoft México, AI Research Assistant @ CIMAT & AI Research Intern @ Harvard.

ferro@cimat.mx

Dato curioso

Imágenes como datos

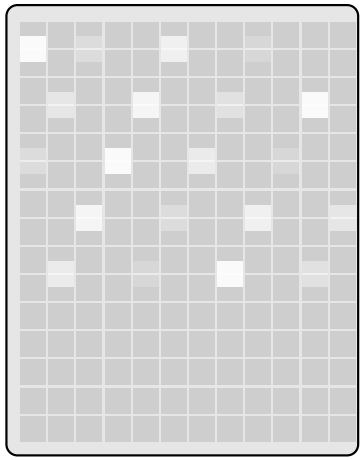
ferro@cimat.mx

Un humano ve una carta de Magic y sabe:

- Qué criatura o hechizo es
- Su rareza por la nota en el marco
- Si está en buen estado
- El color (su identidad)

Todo esto en **menos de 200 ms**.

Una máquina recibe:



$224 \times 224 \times 3 = 150\,528$ números

Pregunta central del taller

¿Cómo pasamos de una matriz de números a una comprensión semántica de la imagen?

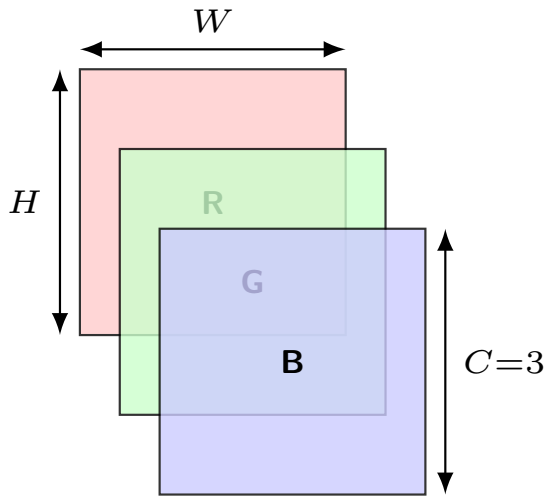
La imagen como matriz de píxeles

Una imagen digital es un **arreglo tridimensional**:

$$I \in \mathbb{R}^{H \times W \times C}$$

donde:

- H = altura en píxeles
- W = ancho en píxeles
- C = número de canales (1 ó 3)



Cada valor es un entero en $[0, 255]$ (uint8) o un real en $[0, 1]$ (float32).

Para reflexionar

Una imagen de 224×224 en RGB tiene 150 528 números. ¿Cómo elegimos cuáles importan para reconocer qué es la imagen?

ferro@cimat.mx

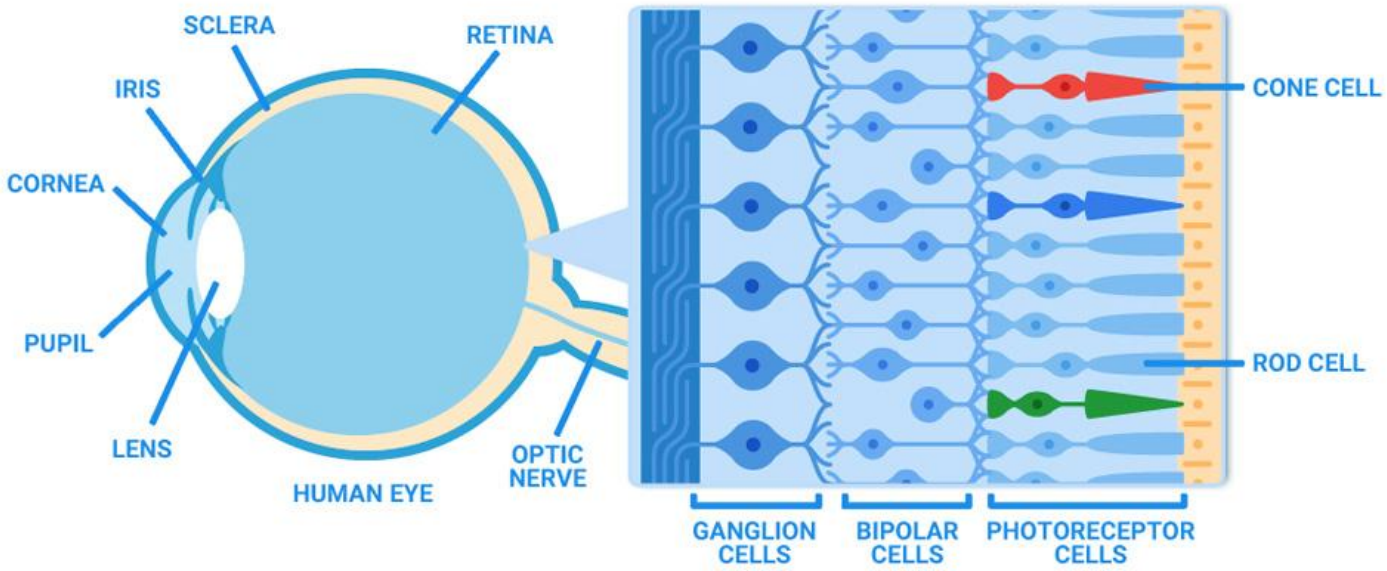
Tipos de imagen según sus canales

Tipo	Canales	Shape	Ejemplo
Escala de grises	1	$H \times W$	Rayos X, microscopía
RGB	3	$H \times W \times 3$	Fotos cotidianas
RGBA	4	$H \times W \times 4$	PNG con transparencia
Hiperespectral	n	$H \times W \times n$	Imágenes satelitales
Volumétrica (3D)	—	$D \times H \times W$	MRI, CT scan

En este taller

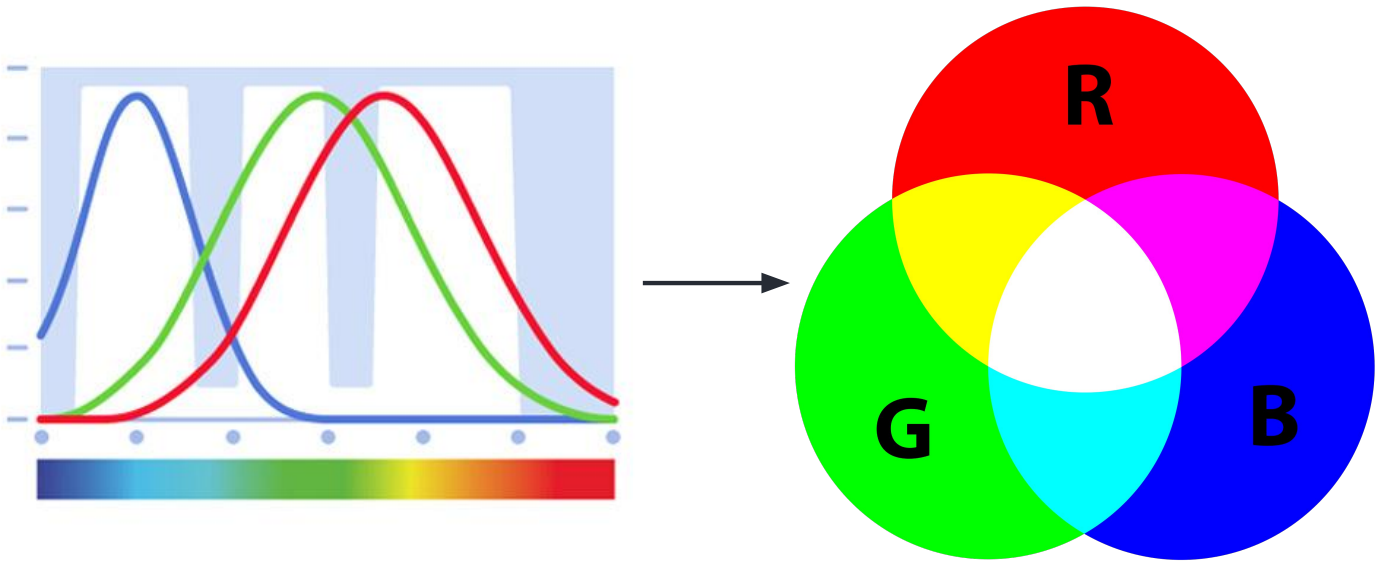
Trabajaremos con imágenes **RGB** de células sanguíneas. Son pequeñas (28×28 px en BloodMNIST) pero suficientes para ilustrar todos los conceptos.

ferro@cimat.mx



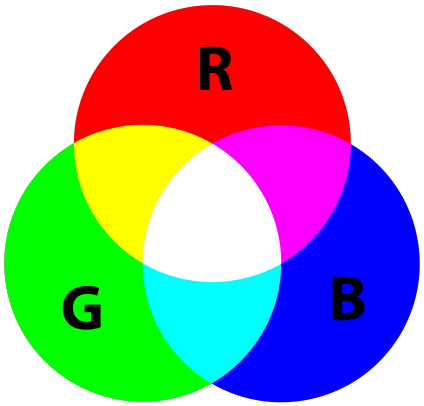
ferro@cimat.mx

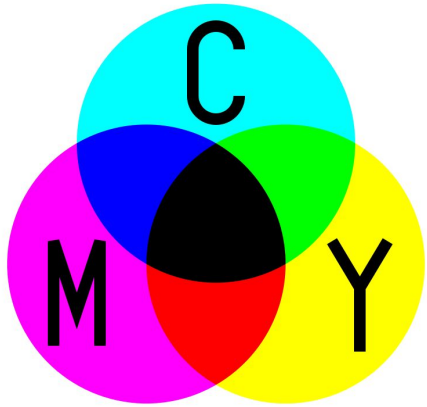
"How EnChroma Color Blind Glasses Work." EnChroma. http://enchroma.com/technology/

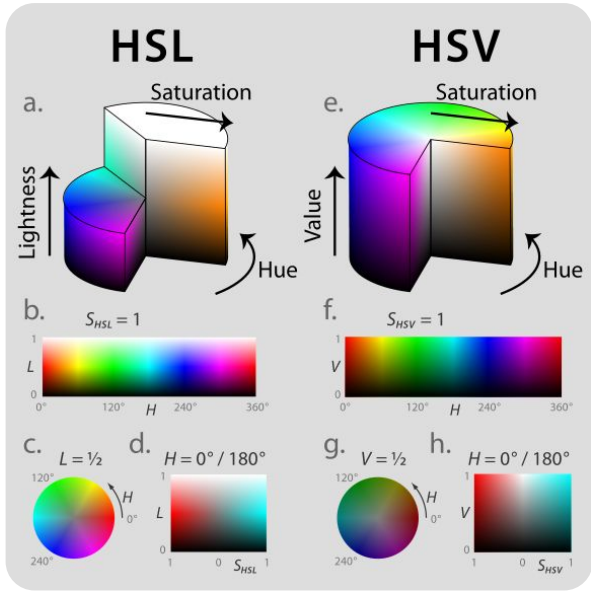


ferro@cimat.mx

"How EnChroma Color Blind Glasses Work." EnChroma. http://enchroma.com/technology/







ferro@cimat.mx

RGB — el más común

- Intuitivo para humanos
- No separa luminosidad de color
- Sensible a cambios de iluminación

HSV (Hue, Saturation, Value)

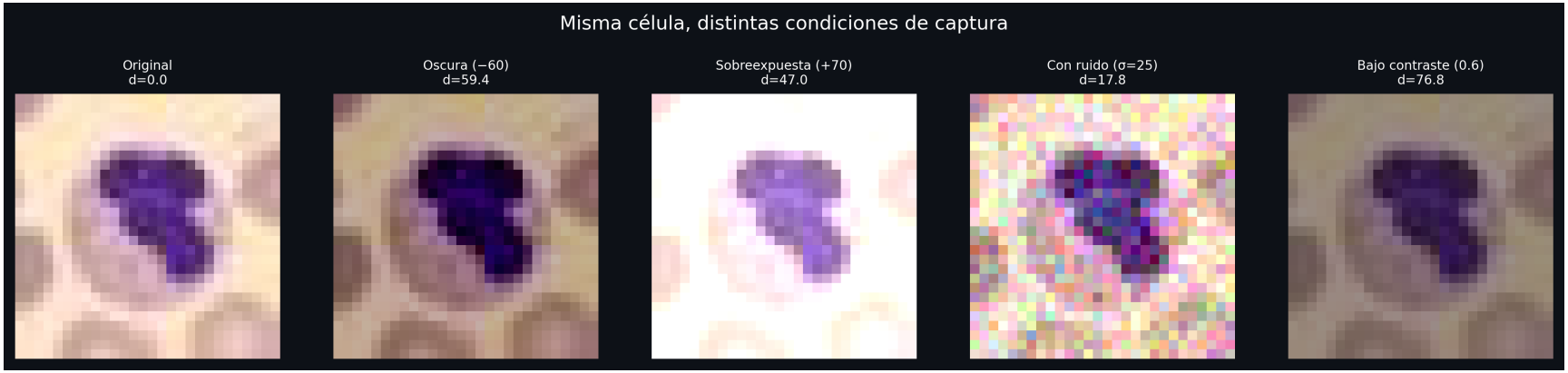
- Separa el *tono* del *brillo*
- Útil para segmentación por color
- Ej.: detectar células por tinción

LAB (CIE L*a*b*)

- Perceptualmente uniforme
- L^*: luminosidad, a^*b^*: cromaticidad
- Estándar en imágenes médicas

ferro@cimat.mx

Una misma célula puede verse **muy diferente** en píxeles:



Distancias en píxeles:

Original ↔ Original: **0.00**
Original ↔ Oscura: **43.67**
Original ↔ Ruido: **22.67**
Original ↔ Otra célula: **85.21**

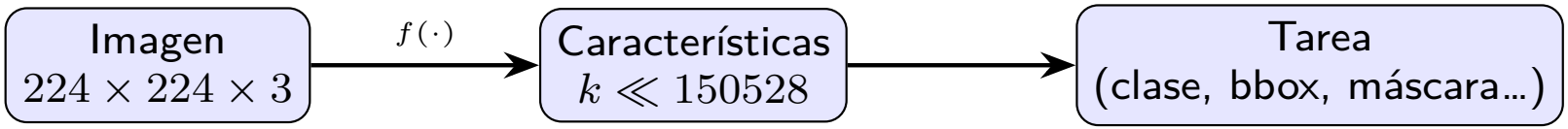
Problema

La distancia pixel a pixel no distingue bien entre “la misma célula bajo otra iluminación” y “una célula diferente”.

Representación de información: ¿qué es una característica?

Definición informal

Una **característica** (*feature*) es cualquier función $f : \mathbb{R}^{H \times W \times C} \to \mathbb{R}^k$ que transforma una imagen en una representación más compacta y útil para la tarea.



La pregunta es: **¿cómo definimos f?**

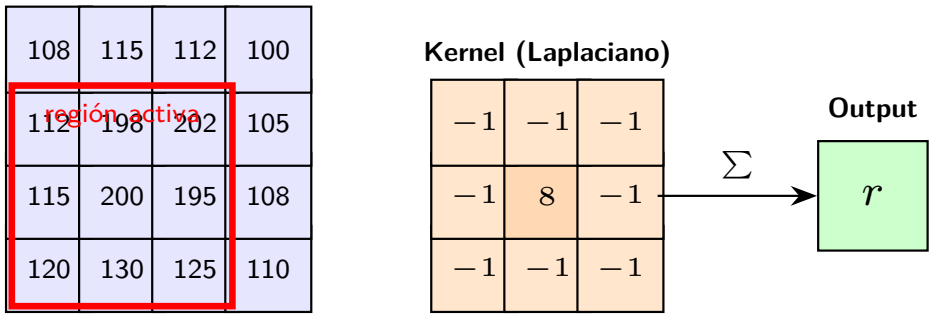
- **Diseñada a mano** (ingeniería de características clásica)
- **Aprendida** de los datos (deep learning)

ferro@cimat.mx

Características diseñadas a mano: convoluciones clásicas

Una **convolución** aplica un filtro (kernel) K deslizándolo sobre la imagen:

$$(I * K)[i, j] = \sum_m \sum_n I[i + m,\ j + n] \cdot K[m, n]$$



Ejemplos de kernels clásicos:

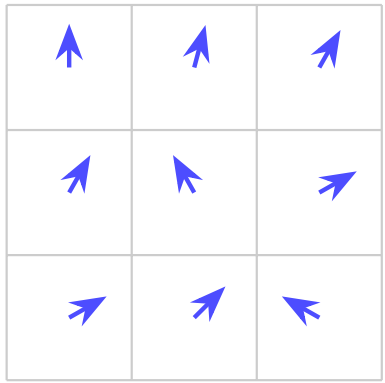
- **Gaussiano**: suavizado, reducción de ruido
- **Sobel/Canny**: detección de bordes
- **Laplaciano**: bordes en todas las direcciones
- **Sharpen**: realzar detalles

[ver Notebook 1, celda de filtros]

ferro@cimat.mx

Histogram of Oriented Gradients (HOG) — Dalal & Triggs, 2005

Idea: En lugar de los píxeles, describir la *distribución local de orientaciones de borde*.



Gradientes locales

Pasos:

1. Calcular gradiente $(\partial_x I, \partial_y I)$ en cada píxel
2. Obtener magnitud y ángulo del gradiente
3. Dividir imagen en celdas de 8×8 px
4. Histograma de ángulos (9 bins) por celda
5. Normalizar por bloques de 2×2 celdas

Resultado: vector de k números que describe la *forma* con invarianza parcial a cambios de iluminación.

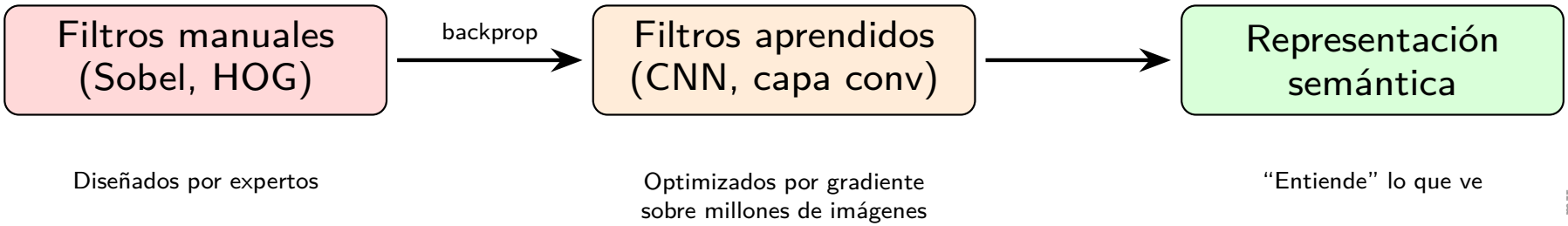
¿Por qué importa HOG hoy?

Ilustra el principio de *feature engineering*: codificar conocimiento del dominio en la representación. Las CNNs aprenden descriptores similares en sus primeras capas, **automáticamente**.

ferro@cimat.mx

De filtros manuales a filtros aprendidos

Imágenes como datos



En una CNN, cada filtro es un tensor
$K \in \mathbb{R}^{k_h \times k_w \times C_{in}}$

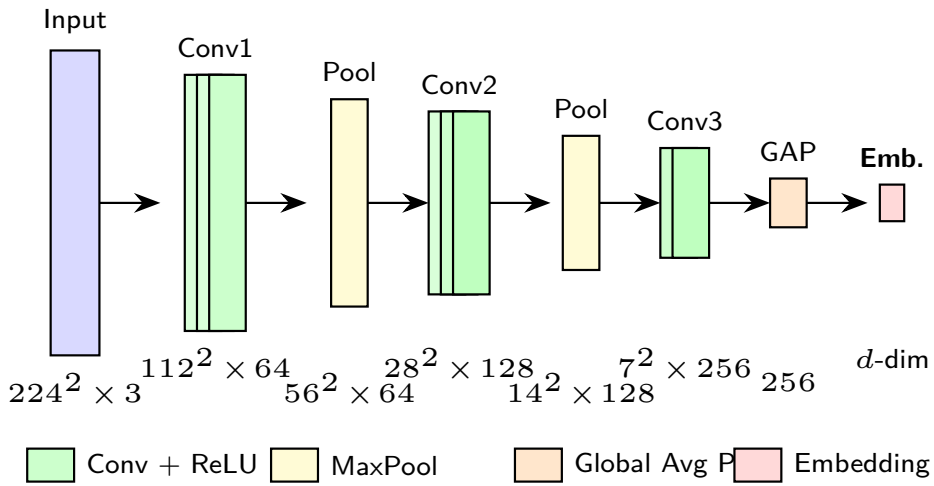
Sus valores **no están fijos**: se aprenden minimizando la pérdida en el conjunto de entrenamiento.

¿Cuántos filtros?

Capa típica: 64–512 filtros en paralelo
$\rightarrow$ cada uno detecta un patrón diferente
$\rightarrow$ la salida es un *mapa de características* $\in \mathbb{R}^{H' \times W' \times C_{out}}$

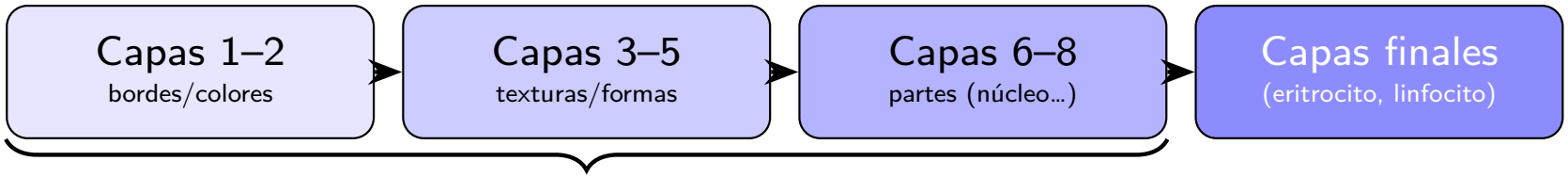
ferro@cimat.mx

Arquitectura de una CNN — vista general



- Las capas convolucionales reducen el *espacio* pero aumentan los *canales*: más filtros, patrones más abstractos
- El Global Average Pooling colapsa los mapas espaciales a un vector fijo: el **embedding**

Las CNNs aprenden una **jerarquía de características**:



Se pueden reusar para otras tareas (*transfer learning*)

Transfer learning

En lugar de entrenar desde cero, tomamos una red pre-entrenada en ImageNet y *ajustamos* solo las últimas capas para nuestra tarea.

ferro@cimat.mx

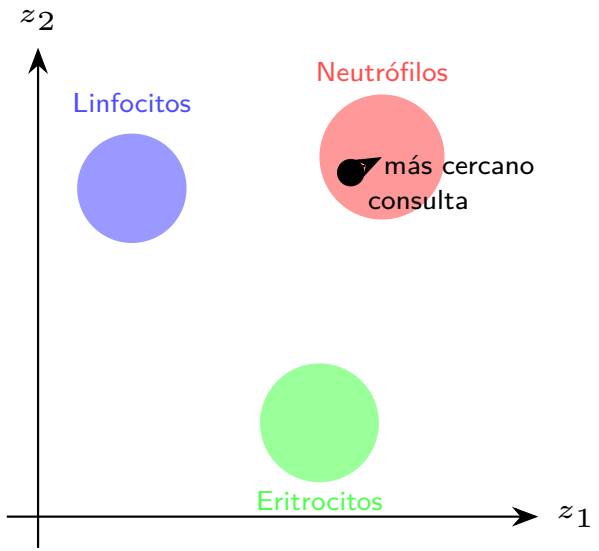
El **embedding** $\mathbf{z} \in \mathbb{R}^d$ es la salida de la CNN antes de la capa de clasificación.

Propiedad clave:
Imágenes visualmente similares $\Rightarrow$ vectores cercanos en $\mathbb{R}^d$

Medida de similitud — coseno:

$$\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \, \|\mathbf{v}\|} \in [-1, 1]$$

- 1: idénticos (mismo ángulo)
- 0: ortogonales (sin relación)
- -1: opuestos



Tarea	Output	Ejemplo biomédico
Clasificación	Etiqueta global	Tipo de célula sanguínea
Detección	Bounding boxes	Localizar núcleos celulares
Segmentación semántica	Máscara por clase	Separar citoplasma / núcleo
Segmentación de instancias	Máscara por objeto	Contar células individuales
Extracción de features	Vector $\mathbb{R}^d$	Indexar imágenes histológicas
Comparación	Similitud $\in [0, 1]$	¿Misma célula en dos fotos?
Imagen $\rightarrow$ texto	Descripción libre	Reporte de hallazgos radiológicos

ferro@cimat.mx

Output: una distribución de probabilidad sobre K clases.

$$\hat{p}_k = \text{softmax}(\mathbf{W}\mathbf{z})_k, \quad \sum_{k=1}^{K} \hat{p}_k = 1$$

Pérdida: entropía cruzada

$$\mathcal{L} = -\sum_k y_k \log \hat{p}_k$$

En BloodMNIST: 8 clases

neutrófilos, eosinófilos, basófilos, linfocitos, monocitos, eritrocitos inmaduros, plaquetas, eritrocitos

[ver descripción del dataset en Notebook 2]

ferro@cimat.mx

Output: lista de bounding boxes + clase + confianza.

Cada box: $(x, y, w, h, \text{clase}, p)$

Modelos populares:

- YOLO (v5, v8, v11) — tiempo real
- Faster R-CNN — más preciso
- DETR — basado en transformers

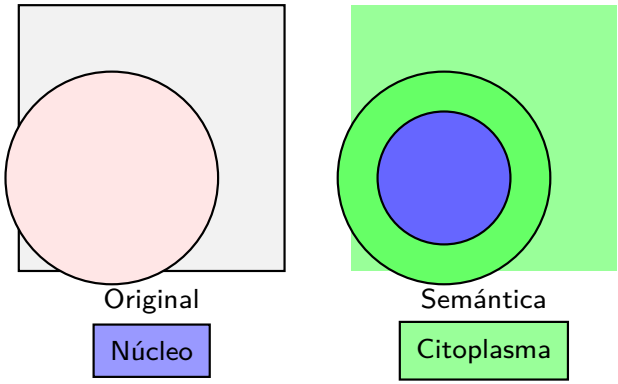
Aplicación:
Contar automáticamente células por tipo en un frotis sanguíneo (hemograma automatizado).

Semántica: cada píxel recibe una etiqueta de clase.

Modelos: U-Net (muy común en biomedicina), DeepLab.

Instancias: distingue entre objetos del mismo tipo (célula 1, célula 2, …).

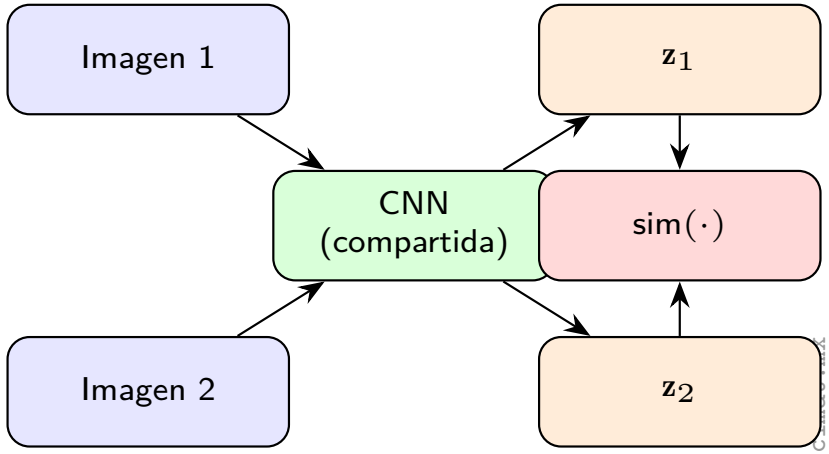
Modelos: Mask R-CNN, SAM (Segment Anything).



ferro@cimat.mx

Extracción de features:
Usar el embedding como representación densa de la imagen.

- Indexar grandes colecciones de imágenes
- Búsqueda por similitud visual (sin etiquetas)
- Clustering / exploración de datos



Comparación (Siamese / contrastive):
Dado un par (I_1, I_2), ¿son del mismo tipo?

$$s = \text{sim}(\mathbf{z}_1, \mathbf{z}_2) > \tau?$$

Aplicación: verificar si dos frotis pertenecen al mismo paciente, o si una célula es anómala.

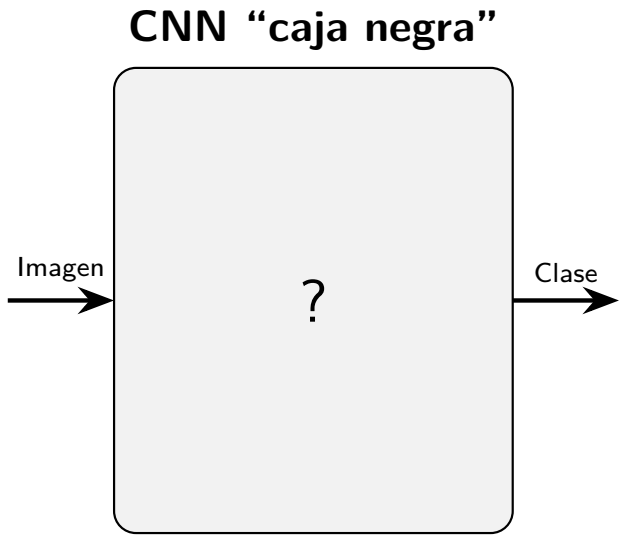
La caja negra y por qué importa abrirla

Una CNN puede alcanzar 95%+ de precisión y aun así **estar mirando lo incorrecto**.

Ejemplos reales de fallos por "shortcuts":

- Red que clasificaba neumonía mirando la regla de escala en la radiografía
- Clasificador de raza de perro que aprendió el fondo (nieve → Husky, sala → Labrador)
- Detector de melanoma que aprendió que las marcas de dermatoscopio → maligno

En medicina, esto puede costar vidas.

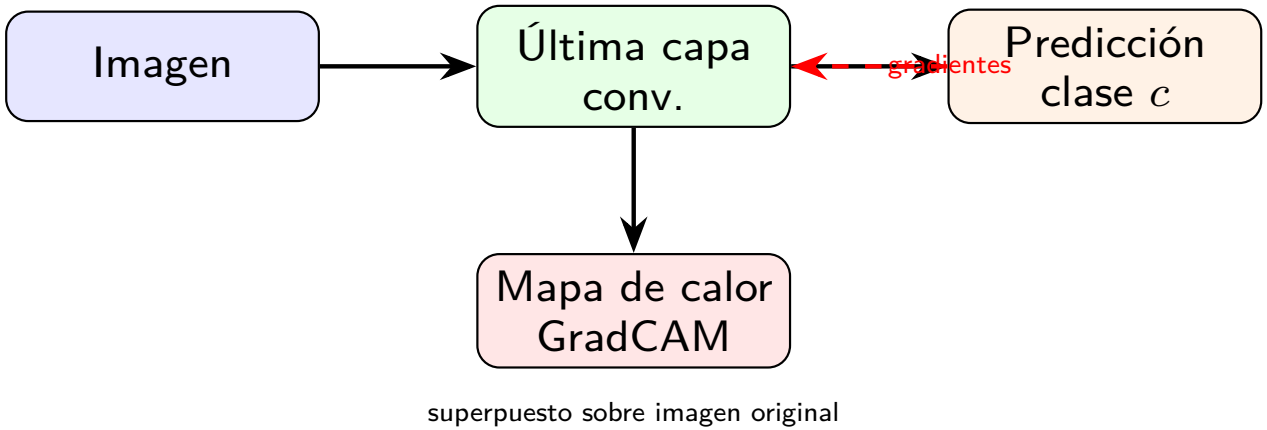


Precisión alta ≠ modelo confiable

ferro@cimat.mx

Gradient-weighted Class Activation Mapping
Selvaraju et al., ICCV 2017

Pregunta: ¿qué regiones de la imagen influyeron más en la predicción de la clase c?



Idea central
Los gradientes de la clase c respecto a los mapas de características de la última capa convolucional nos dicen qué canales (y qué regiones) son más importantes para esa predicción.

ferro@cimat.mx

GradCAM: algoritmo paso a paso

1. **Forward pass:** obtener los mapas de características $A^k \in \mathbb{R}^{H' \times W'}$ de la última capa conv.

2. **Backward pass:** calcular los gradientes de la puntuación de clase y^c respecto a cada mapa A^k:

$$\frac{\partial y^c}{\partial A^k_{ij}}$$

3. **Pesos de importancia:** promedio global de los gradientes (Global Average Pooling sobre los gradientes):

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}}$$

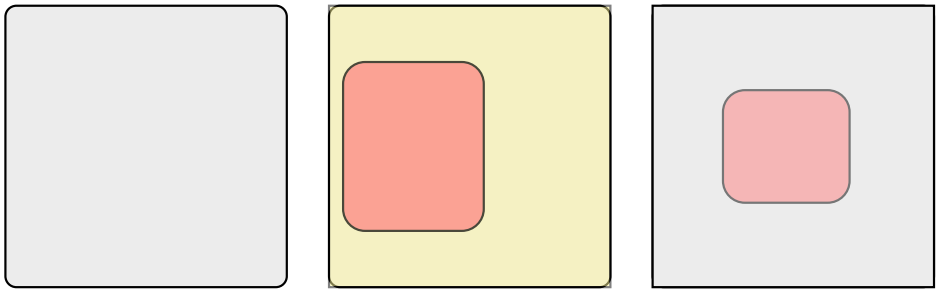
4. **Mapa de activación:** combinación lineal ponderada, seguida de ReLU (solo regiones con efecto positivo):

$$L^c_{\text{GradCAM}} = \text{ReLU}\left(\sum_k \alpha_k^c \, A^k \right)$$

5. **Visualización:** escalar L^c al tamaño original y superponer como mapa de calor.

ferro@cimat.mx

GradCAM: resultado esperado



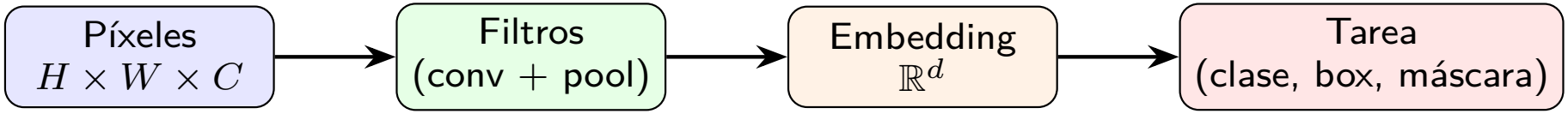
Original GradCAM Superposición

¿Qué buscar en el mapa?

- ¿La red mira el **núcleo** (forma lobulada en neutrófilos)?
- ¿O el **citoplasma** (granularidad en eosinófilos)?
- ¿O la **forma global** (bicóncava en eritrocitos)?

ferro@cimat.mx

Resumen: del píxel al concepto



¿Qué "observó" la red?
$\Rightarrow$ **GradCAM**

Aprendido hoy:

- Imagen = matriz de números
- Espacios de color y variabilidad
- Filtros clásicos (Sobel, HOG)
- CNNs: filtros aprendidos, jerarquía
- Embeddings y similitud coseno
- Panorama de tareas en visión
- GradCAM: abriendo la caja negra

Ahora en Notebook 2:

- Dataset BloodMNIST: 8 tipos de células sanguíneas
- Clasificación con EfficientNet
- GradCAM desde cero en PyTorch
- Matriz de similitud de embeddings
- Conexión con el detector de cartas

ferro@cimat.mx

¡Hora de poner manos al código!

`Notebook 1:` *La imagen como dato*

`Notebook 2:` *Clasificación con CNNs e Interpretabilidad con GradCAM*

ferro@cimat.mx

Recursos

- MedMNIST: `medmnist.com`
- GradCAM original: Selvaraju et al., ICCV 2017
- Feature visualization: `distill.pub/2017/feature-visualization`
- timm models: `huggingface.co/timm`