DEEP LEARNING

Embeddings en CNN

De una imagen a un vector

Dr. León Felipe Palafox Novack

lfpalafox@up.edu.mx

01 La idea central

02 Del píxel al concepto

03 Feature maps y pooling

04 Del clasificador al encoder

05 Búsqueda y clustering

06 Transfer learning y anomalías

07 Buenas prácticas

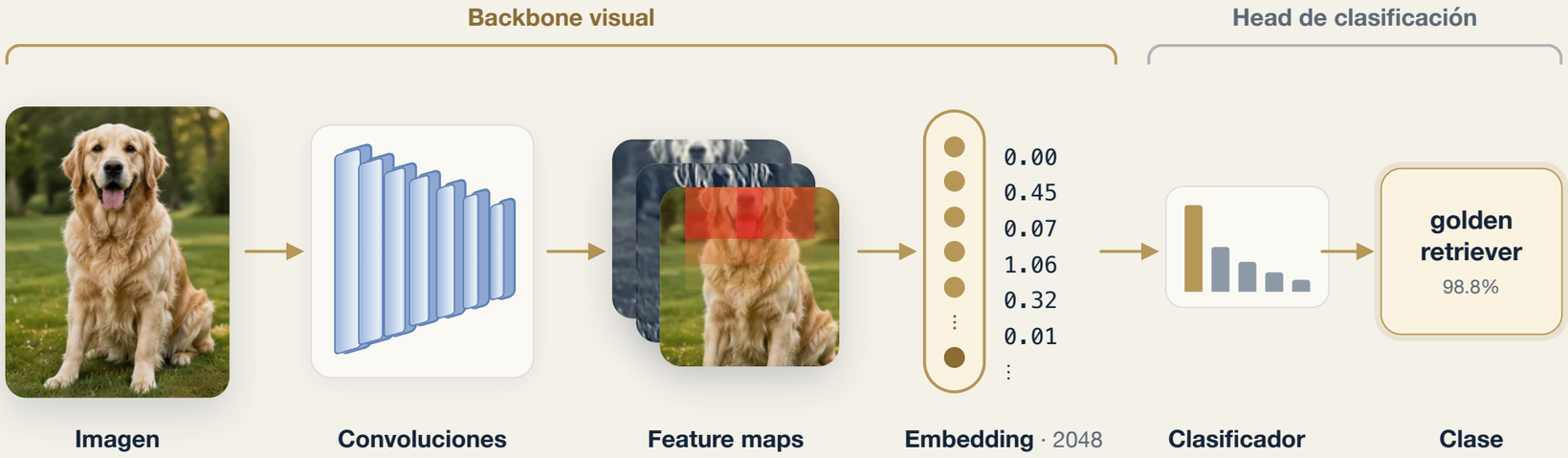
02

La idea central

Una CNN hace más que clasificar imágenes.

Antes de producir una etiqueta, la red construye una **representación numérica** de la imagen.

Esa representación es el **embedding**.



Números reales: esta foto pasó por una ResNet-50 entrenada en ImageNet. El embedding tiene 2,048 valores; aquí ves los primeros seis. Los mapas son activaciones reales de la red.

03

¿Qué es un embedding?

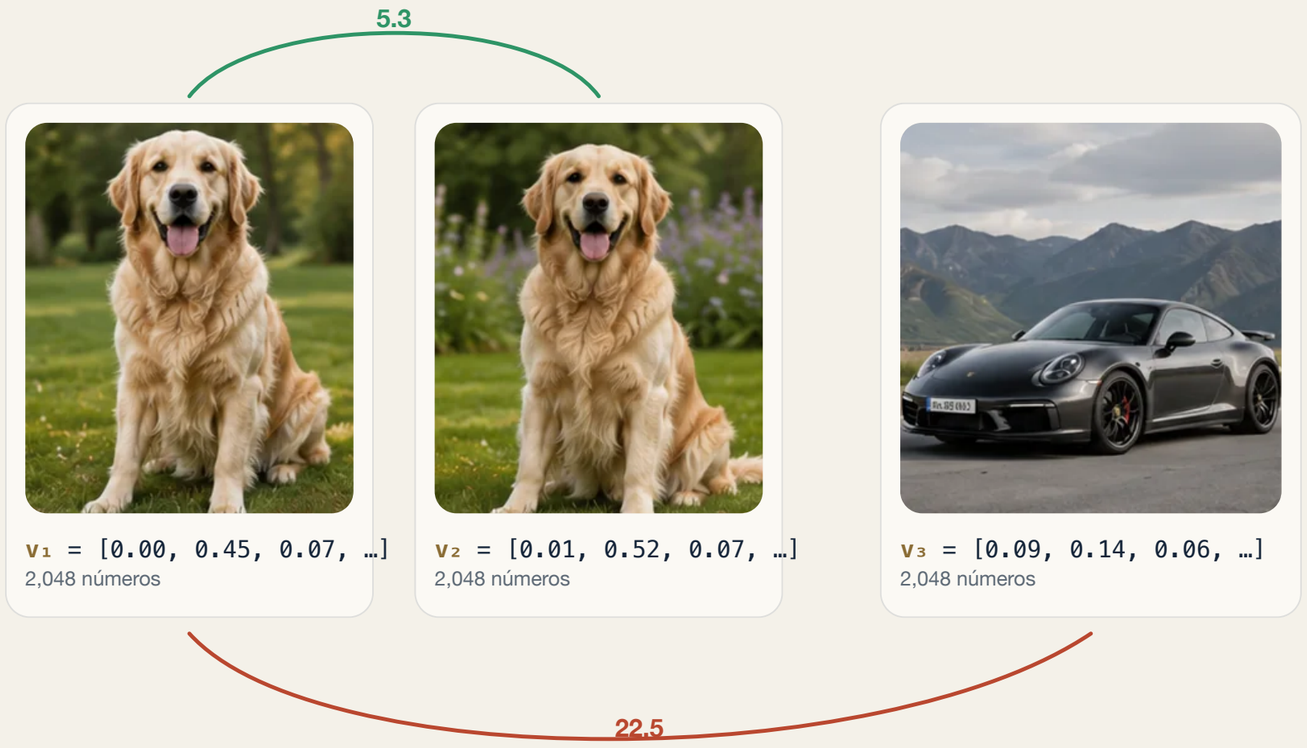
Un embedding es un **vector** que representa un objeto.

En visión por computadora, representa una imagen como una **lista de números**.

La meta: que el vector capture información visual útil.

Forma · Textura · Partes · Color · Estilo · Identidad del objeto · Similitud visual

Distancia euclidiana entre embeddings reales de ResNet-50. Similitud coseno: 0.97 (perro–perro) vs 0.51 (perro–auto).



04

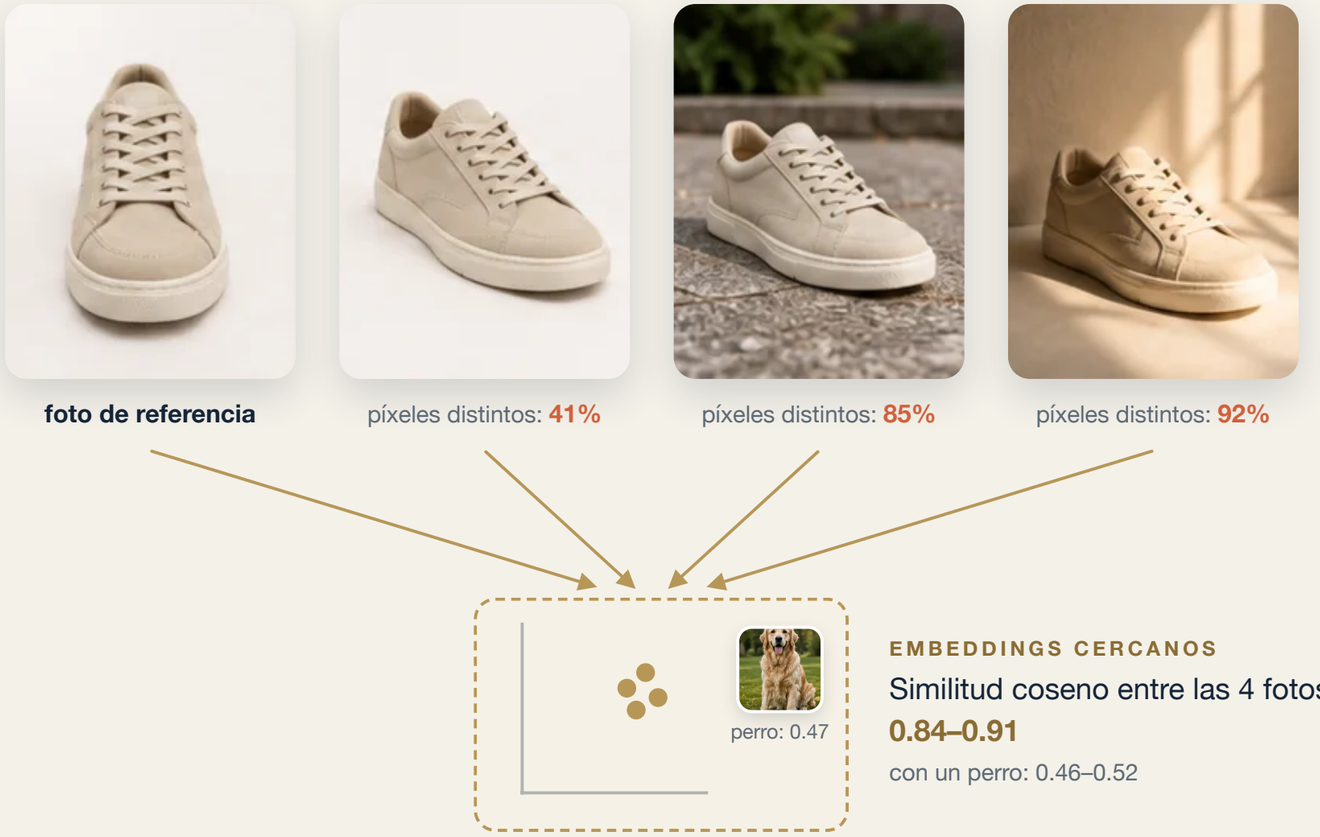
No queremos comparar píxeles

Los píxeles son **frágiles**.

Dos imágenes del mismo objeto pueden tener valores de píxel muy distintos por:

- Iluminación
- Escala
- Rotación
- Fondo
- Perspectiva
- Oclusión

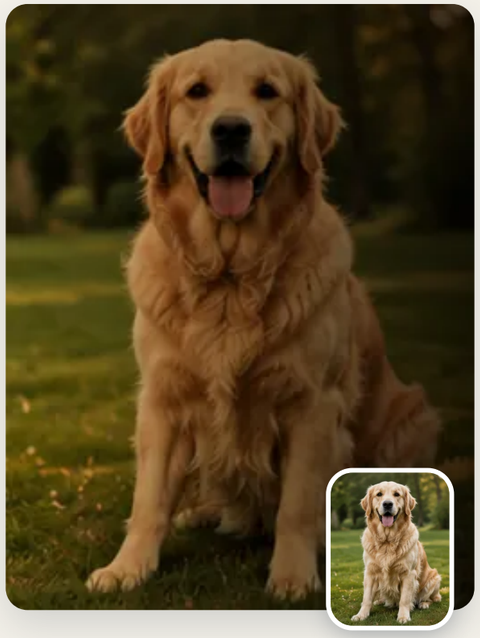
El embedding busca capturar **significado visual**, no solo píxeles.



EXPERIMENTO

Mismo perro, otros píxeles

- Original
- Iluminación
- Escala
- Rotación
- Fondo
- Perspectiva
- Oclusión



Iluminación

PÍXELES QUE CAMBIARON

97%

del recorte de 224×224 que ve la red (cambio mayor a 10%)

SIMILITUD DEL EMBEDDING CON LA ORIGINAL

0.98

perro vs auto: 0.50

similitud coseno de los vectores de 2048 dimensiones

LA RED DICE

golden retriever · 99%

→ / ← recorren las transformaciones · también puedes hacer clic en cada botón

06

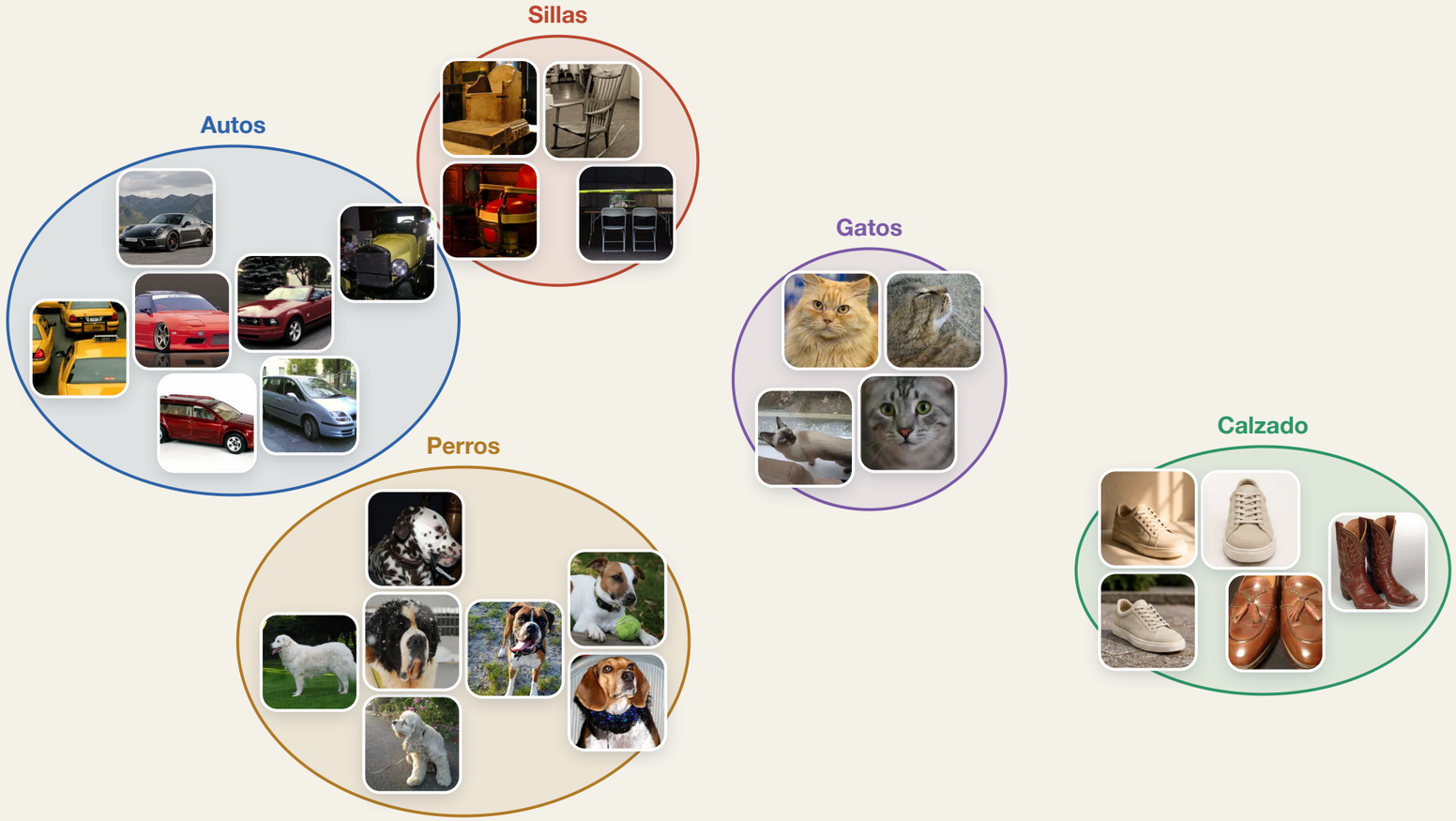
Embeddings como coordenadas

Después de pasar por una CNN, cada imagen se vuelve **un punto**.

La posición depende de lo que la red aprendió a reconocer.

Imágenes similares quedan **cerca**; imágenes diferentes, **lejos**.

27 fotos reales · embeddings de ResNet-50 (2048 dimensiones) proyectados a 2D con t-SNE.



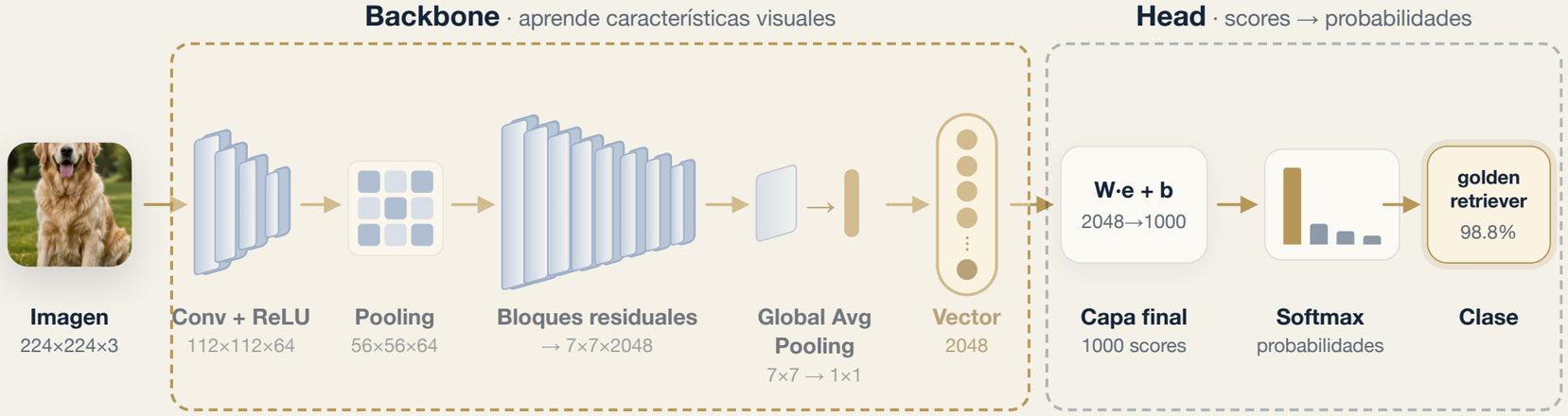
07

Recordatorio: una CNN clasificadora

Una CNN clasificadora tiene **dos partes** principales.

Backbone: aprende características visuales.

Head de clasificación: convierte la representación en probabilidades de clase.



Formas reales de ResNet-50 para una imagen de 224×224: la parte espacial se reduce 32 veces (224 → 7) mientras los canales crecen de 3 a 2048.

08

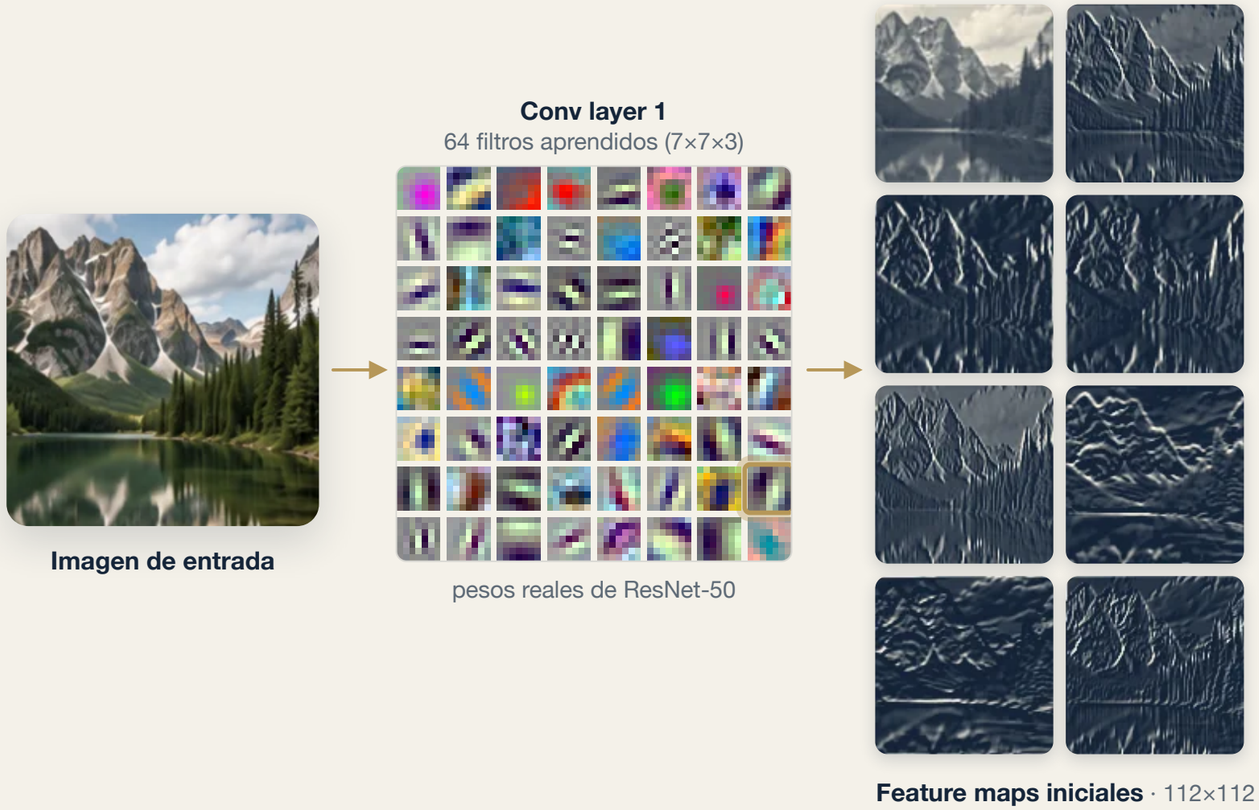
De píxeles a patrones locales

Las primeras capas detectan **patrones simples**.

EJEMPLOS

- Bordes
- Esquinas
- Texturas
- Cambios de color
- Líneas orientadas

Estas capas todavía están cerca del mundo de los píxeles.



09

De patrones locales a conceptos visuales

Las capas profundas combinan patrones simples en **conceptos más complejos**.

LA RED APRENDE COMBINACIONES DE

- Bordes
- Texturas
- Partes de objetos
- Formas
- Configuraciones visuales



La red tiende a capturar estas abstracciones de manera **jerárquica**, no de forma rígida ni exclusiva.

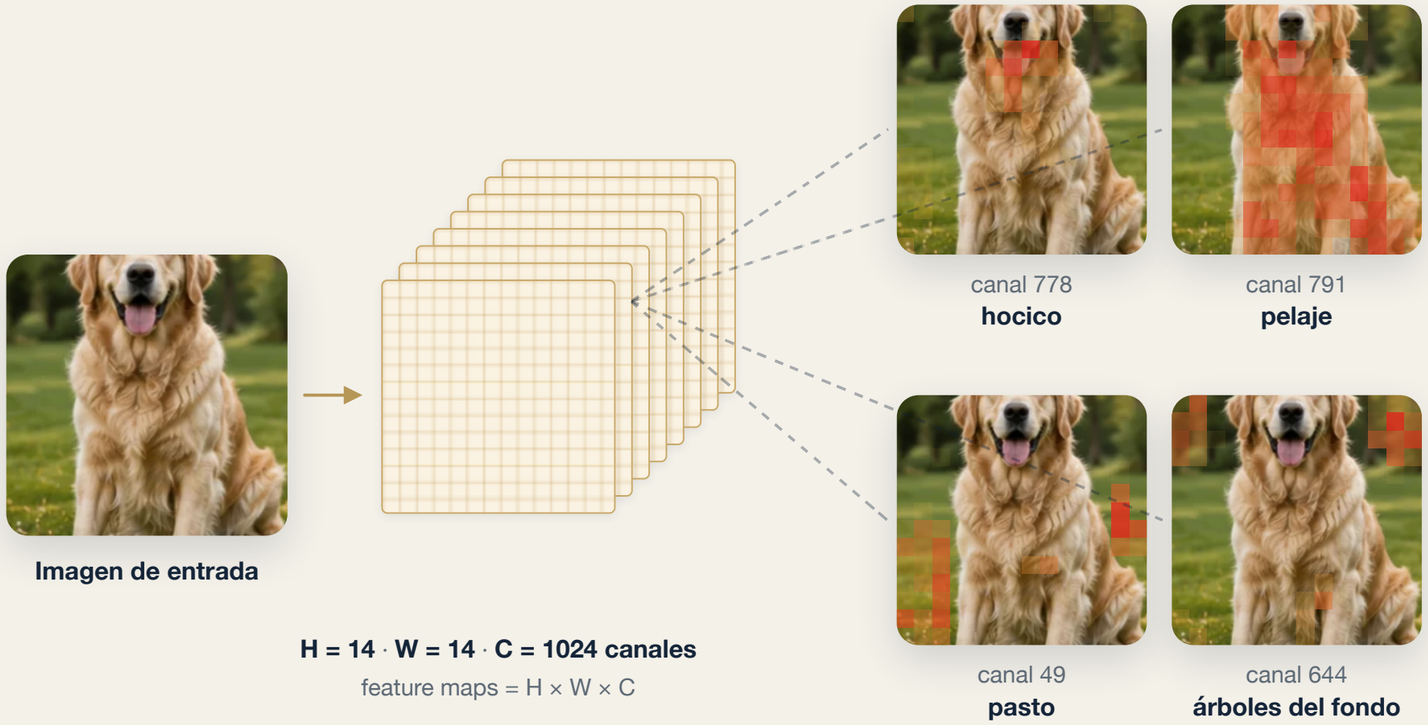
10

Feature maps: un volumen de activaciones

Una capa convolucional no produce un solo número.

Produce un **volumen** de activaciones: **alto × ancho × canales**.

Cada canal puede responder a un tipo de patrón aprendido.



11

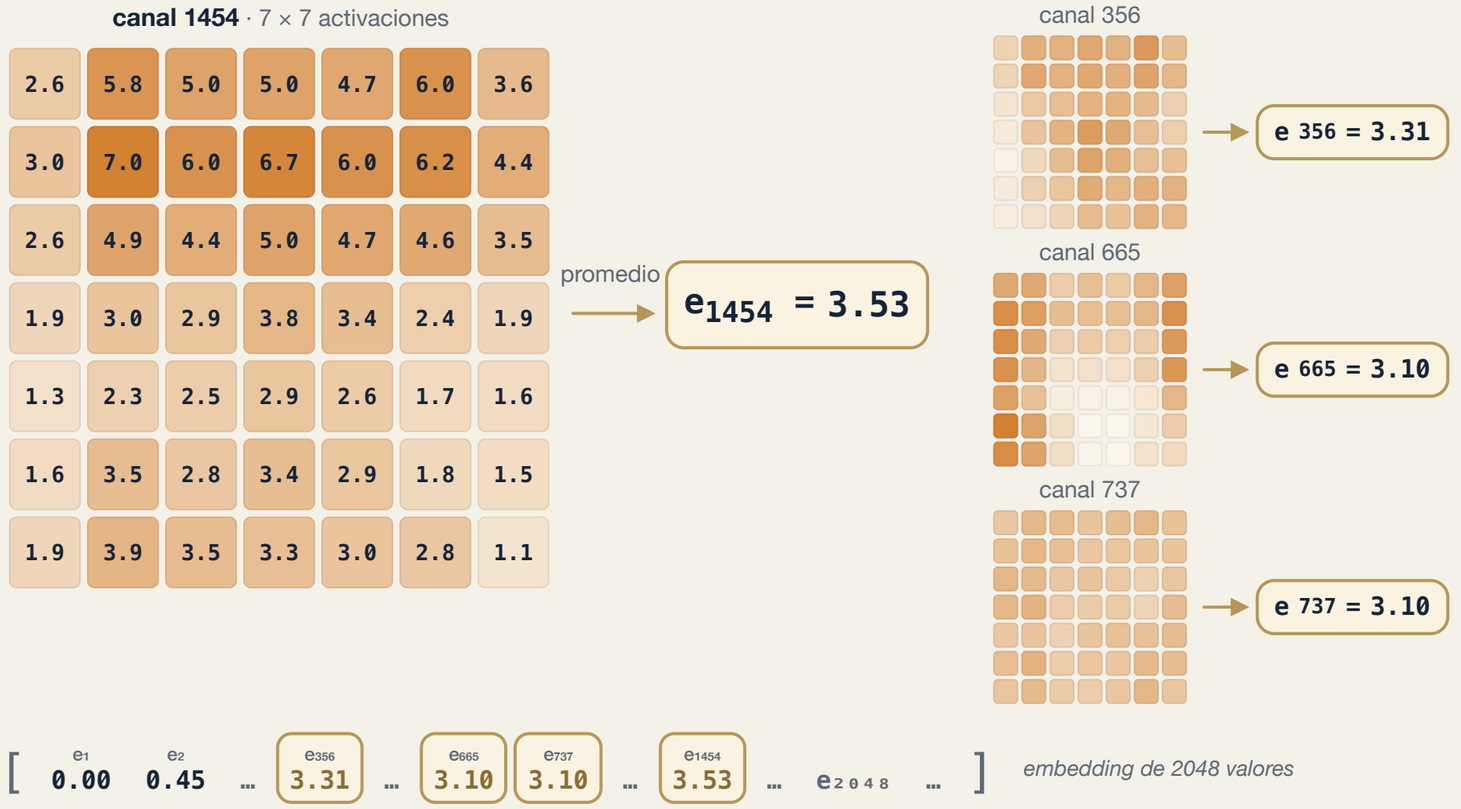
Cómo se comprime un volumen en un vector

Para obtener un embedding pasamos de **mapas espaciales** a un **vector**.

Operación común: **Global Average Pooling**.

Convierte	$H \times W \times C$ 7 × 7 × 2048
en	$1 \times 1 \times C$ 1 × 1 × 2048
y luego	$[e_1, e_2, \ldots, e_C]$ 2048 números

Cada canal se promedia espacialmente para producir **una coordenada** del vector.



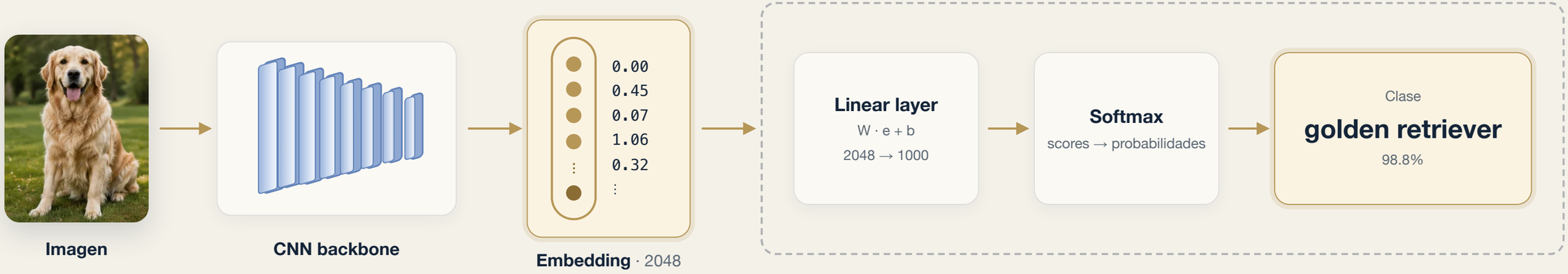
12

El embedding vive antes del clasificador

El embedding está justo **antes de la decisión final**.

La CNN primero genera una representación; después, el clasificador la usa para producir una clase.

embedding → clasificador → probabilidades



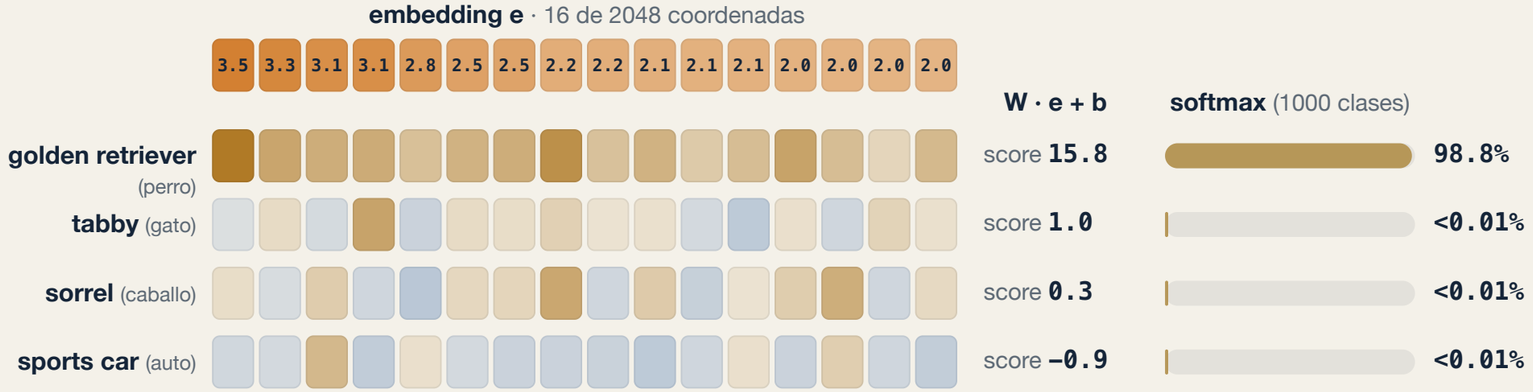
13

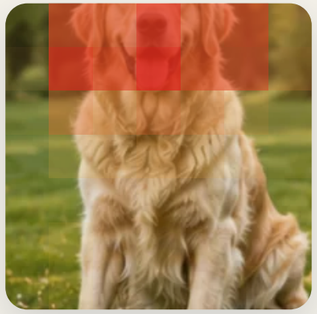
¿Qué hace realmente la última capa?

La última capa aprende una **frontera entre clases**.

Toma el embedding y calcula un **score** por categoría: una fila de W por clase.

No descubre la imagen desde cero: **lee la representación** que el backbone ya construyó.





¿QUÉ MIRA LA FILA «GOLDEN RETRIEVER»?

Sus 2048 pesos, aplicados a los mapas de 7×7 de la última capa, encienden la **cara del perro**.

TOP 5 DE 1000 CLASES

golden retriever	98.8%
Labrador retriever	0.30%
kuvasz	0.19%
Brittany spaniel	0.16%
tennis ball	0.12%

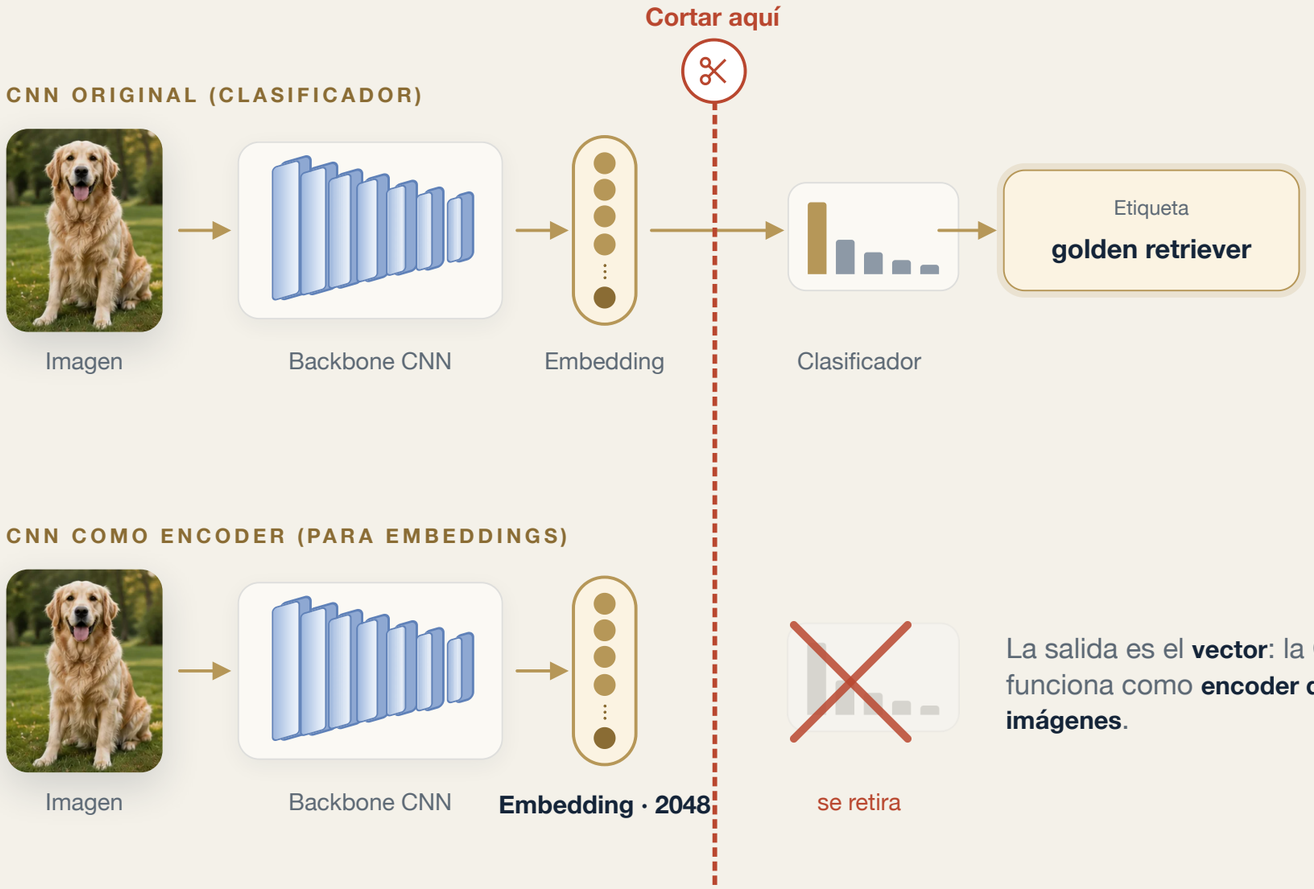
14

El truco principal: quitar la última capa

Para obtener embeddings, **quitamos la última capa** de clasificación.

La CNN deja de ser solo un clasificador.

Ahora funciona como un **encoder de imágenes**.



La salida es el **vector**: la CNN funciona como **encoder de imágenes**.

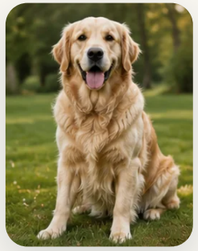
15

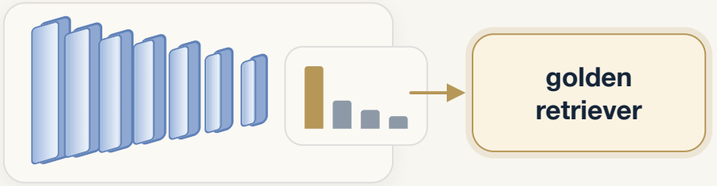
Clasificador vs extractor de embeddings

Misma base convolucional. **Diferente salida.**

Clasificador

Entrada: imagen





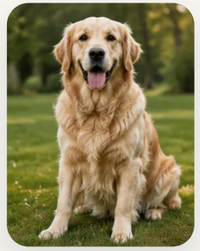
golden retriever

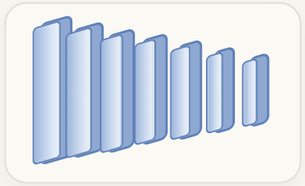
Salida: clase

CNN completa
(backbone + head)

Extractor de embeddings

Entrada: imagen





Salida: vector

0.00
0.45
0.07
1.06
⋮
0.28
0.02

2048 valores

CNN sin última capa
(backbone)

La salida ya no es una etiqueta. **Es una representación.**

16

Ejemplo concreto con ResNet-50

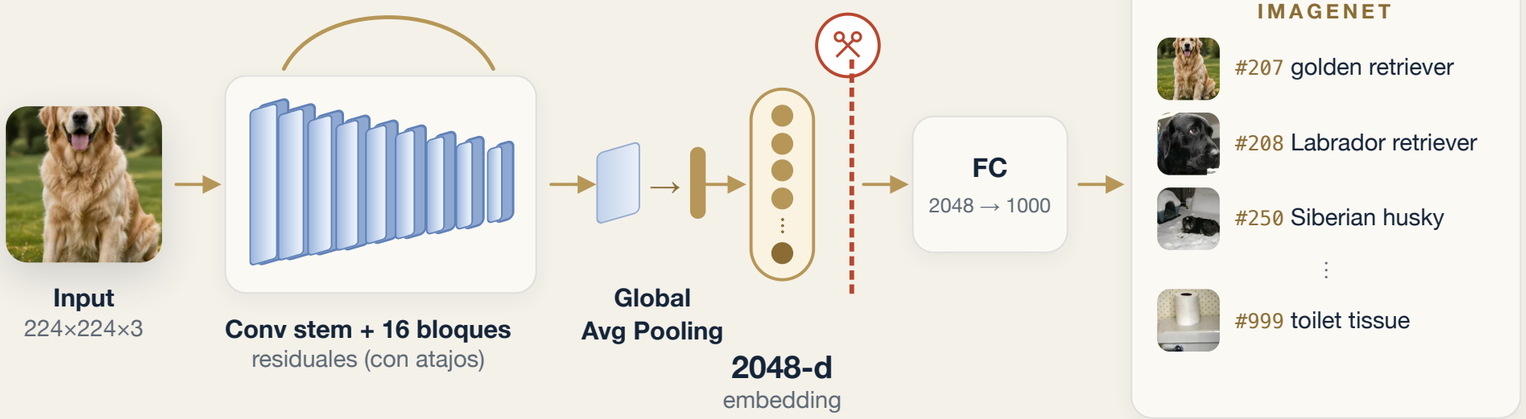
Una ResNet entrenada para clasificar se divide en **backbone convolucional** y **capa fully connected final**.

Para extraer embeddings usamos la salida **antes de la capa final**.

2048
dimensiones del embedding

≈25.6 M
parámetros en total

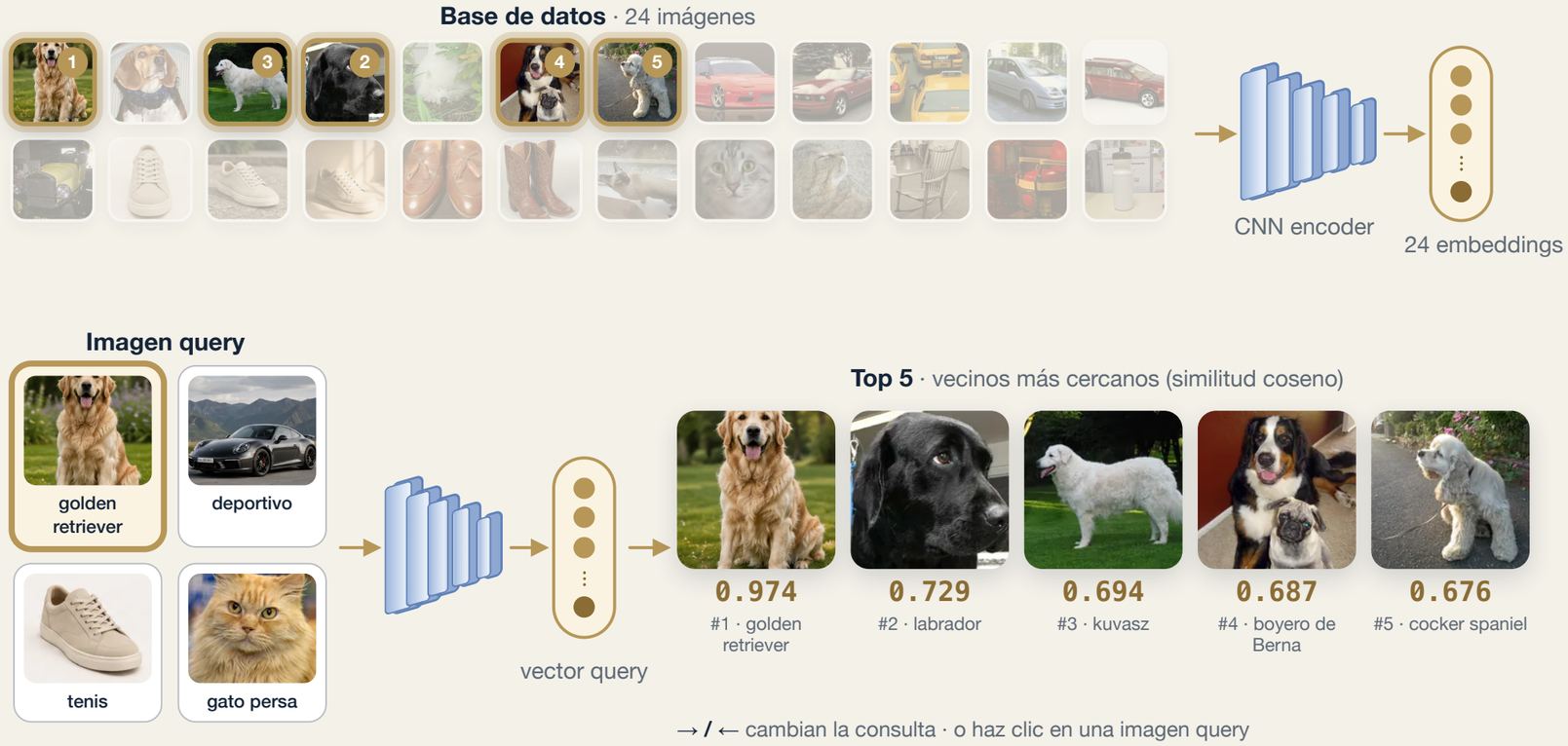
2,049,000
en la capa final (8%)



CASO 1

Búsqueda visual por similitud

1. Pasamos todas las imágenes por la **CNN encoder**.
2. Guardamos sus **embeddings**.
3. Pasamos una imagen nueva por el **mismo encoder**.
4. Buscamos los **vectores más cercanos**.
5. Regresamos las **imágenes más similares**.



CASO 2

Clustering sin etiquetas

Los embeddings permiten **explorar colecciones** de imágenes sin etiquetas.

Imágenes sin etiqueta
↓
CNN encoder
↓
Embeddings
↓
Clustering
↓
Grupos visuales

Clustering: k-means · DBSCAN · HDBSCAN



k-means (k = 5) colorea cada foto según su grupo.

RESULTADO

20 de 20

fotos en el grupo correcto, sin usar ninguna etiqueta.

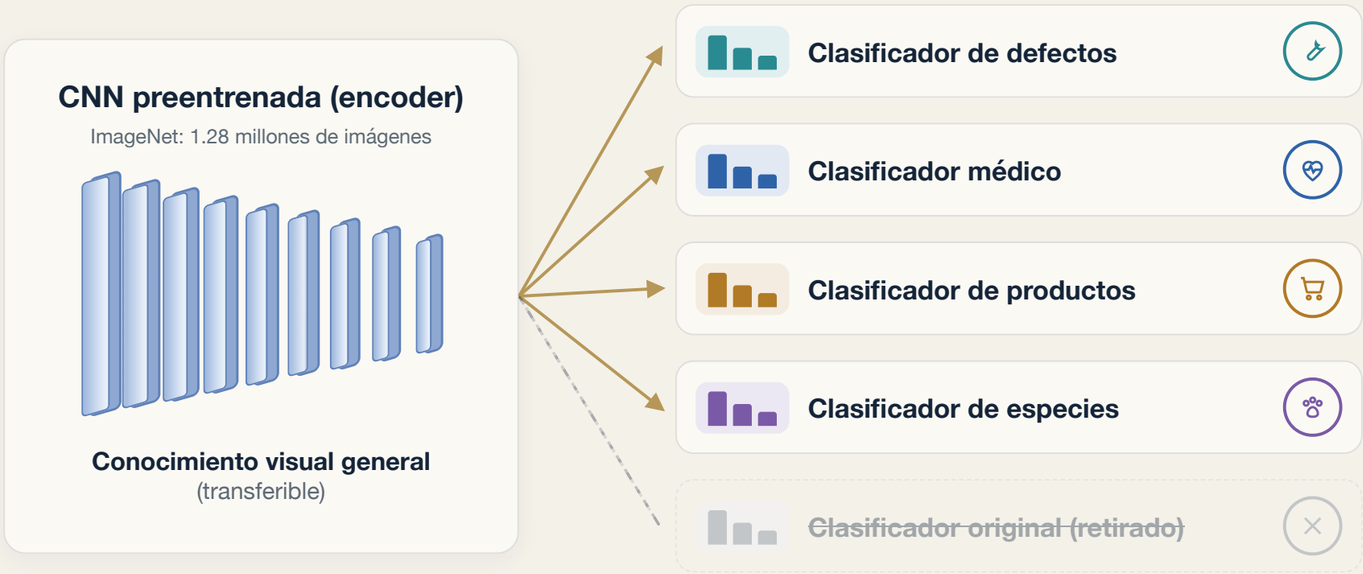
CASO 3

Transferencia de aprendizaje

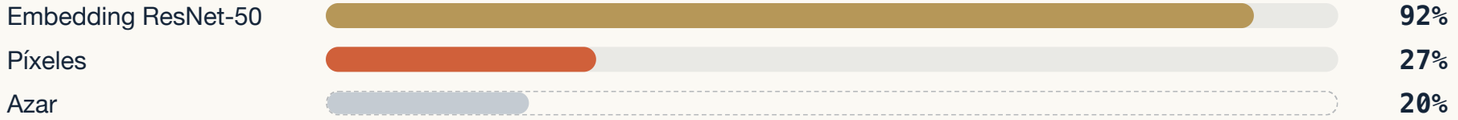
Una CNN entrenada en millones de imágenes ya aprendió **patrones visuales generales**.

Podemos reutilizar ese conocimiento para una tarea nueva, sobre todo con **pocos datos etiquetados**.

> El **embedding** es la parte transferible; la cabeza (**head**) se adapta a la nueva tarea.



EXPERIMENTO REAL · 1 FOTO ETIQUETADA POR CLASE (5 CLASES, 39 FOTOS)



CASO 4

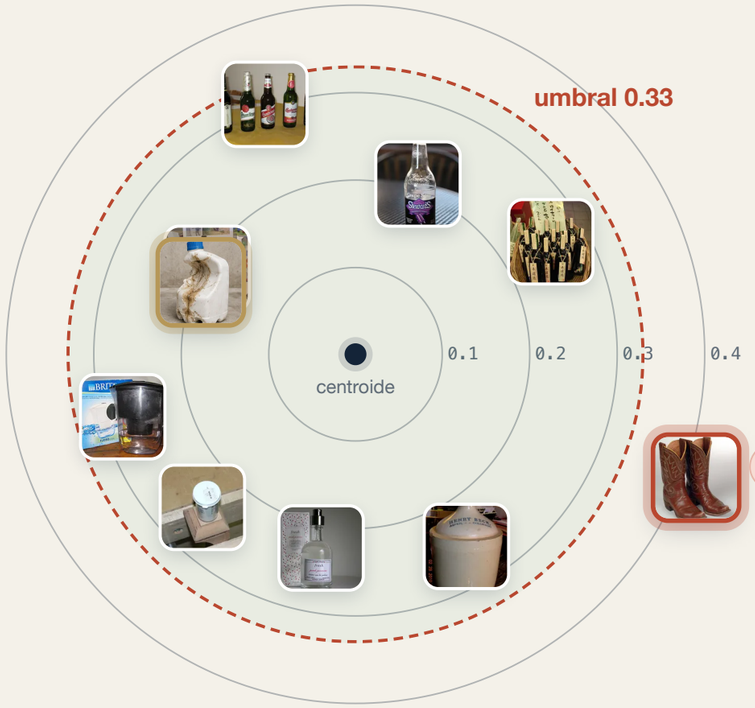
Detección de anomalías

Si las imágenes normales forman una **región compacta**, una imagen extraña puede aparecer **lejos**.

IDEA

- ✓ **normal** = cerca del grupo
- ! **anómalo** = lejos del grupo

> El embedding ayuda a medir **qué tan diferente** es una imagen… según la similitud que aprendió.





Garrafón dañado (de la diapositiva original): distancia **0.20**, **dentro** del grupo.

ResNet-50 lo describe como «water jug»: no mira manchas ni golpes. Para defectos, el embedding debe aprender *esa* similitud.

21

¿Qué hace que un embedding sea bueno?

Un buen embedding conserva **la similitud que importa** para la tarea.

Pero «similar» no siempre significa lo mismo:

- **Para identificar objetos:** perro cerca de perro.
- **Para buscar estilo:** objetos rojos cerca de objetos rojos.
- **Para detectar defectos:** productos normales cerca de normales.

> ⚖ Un embedding no es «la verdad» de la imagen. Es una **representación aprendida** para cierto tipo de similitud.



→ / ← cambian de espacio: las mismas fotos, otra noción de «parecido».

22

Decisiones al extraer embeddings

DECISIONES IMPORTANTES

1. ¿Qué CNN usamos?
2. ¿Preentrenada o desde cero?
3. ¿De qué capa sacamos el embedding?
4. ¿Congelamos el backbone o hacemos fine-tuning?
5. ¿Normalizamos el vector?
6. ¿Qué métrica usamos para comparar?

Extraer embeddings es sencillo conceptualmente, pero **las decisiones de diseño cambian el resultado**.



Similitud coseno
compara la dirección



perro – perro	0.974
perro – auto	0.505

Distancia euclidiana
con vectores de norma 1



perro – perro	0.228
perro – auto	0.995

Vectores tal cual | **Normalizados (L2)**

$$\|a - b\|^2 = 2 - 2 \cdot \cos(a, b)$$

23

Cierre: clasificación vs representación

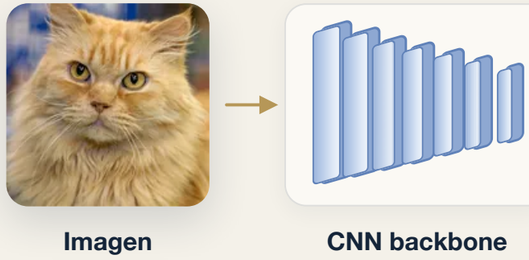
IDEA FINAL

Una CNN puede ser **mucho más** que un clasificador: puede ser un **encoder visual**.

La última capa produce **etiquetas**. La capa anterior produce **representaciones**.

Y esas representaciones permiten buscar, comparar, agrupar, transferir aprendizaje y detectar anomalías.

La clasificación es una salida.
La representación es el activo reutilizable.



Clasificación
Predice etiquetas (la salida tradicional)

Búsqueda visual
Encuentra imágenes similares

Clustering
Agrupa imágenes por similitud

Transfer learning
Reutiliza la representación en tareas nuevas

Anomaly detection
Detecta instancias fuera de patrón

El **embedding** es el **puente** entre una CNN entrenada y muchas aplicaciones prácticas de visión por computadora.