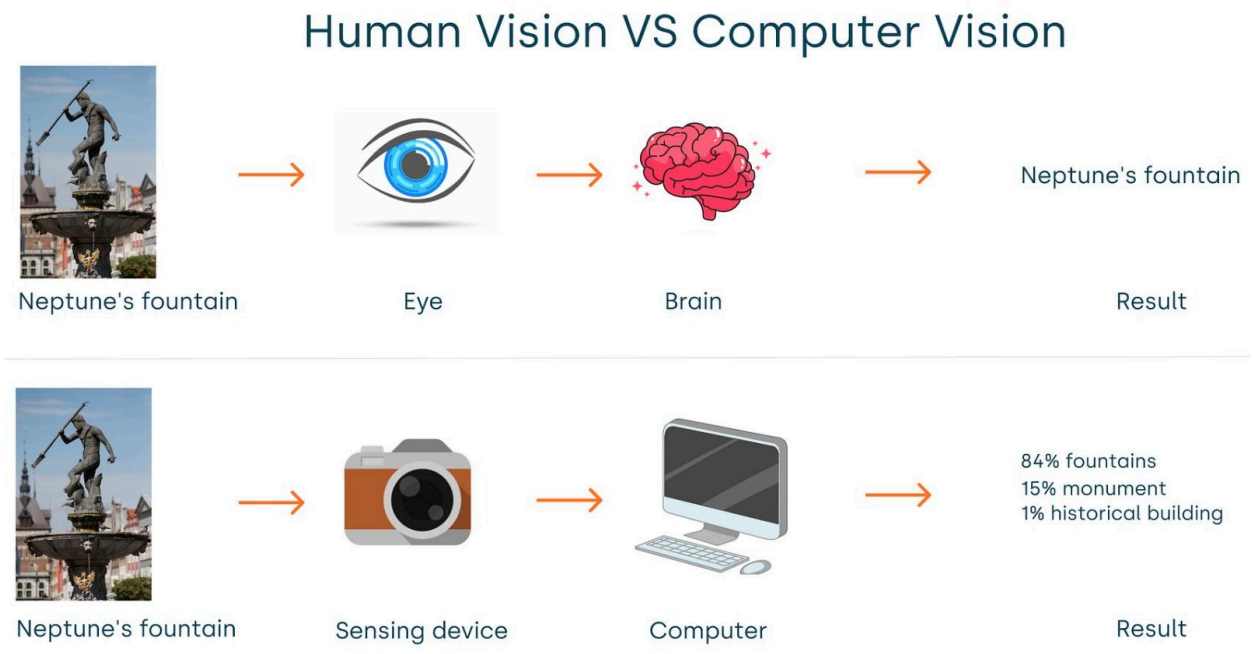
De píxeles a significado

Una imagen empieza como números. La tarea decide qué información buscamos.



El modelo conecta patrones visuales con una salida útil.

Las imágenes digitales son matrices de píxeles. La visión artificial aprende representaciones que permiten asociar esos valores con objetos, relaciones o regiones.

¿Qué salida necesitas?

La misma imagen admite tareas distintas.

Quieres contar coches y localizar cada uno.

- A Una etiqueta para toda la imagen.
- B Una caja y una categoría por objeto.
- C Una puntuación de confianza para la imagen.

Respuesta B. Exacto: la detección localiza cada instancia y permite contarla.

La clasificación da una etiqueta global. La detección devuelve cajas y categorías por instancia. La segmentación delimita píxeles. Otros problemas incluyen anomalías, pose, OCR, profundidad y superresolución.

Detectar es localizar

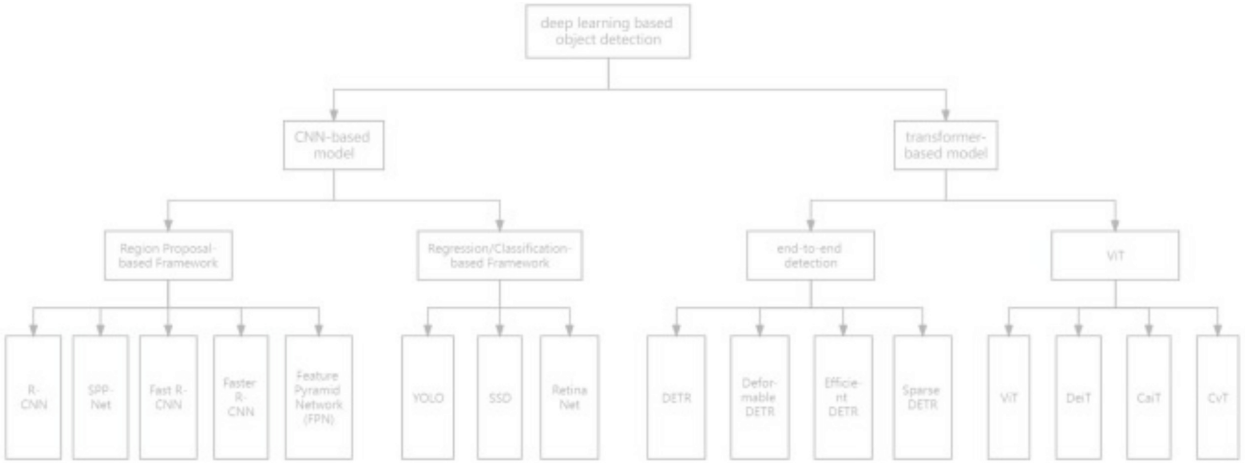
Cada detección combina una caja, una categoría y una puntuación.

Caja · categoría · puntuación de confianza

La puntuación ordena las predicciones según el modelo. No equivale automáticamente a una probabilidad calibrada y no dice por sí sola si la caja está bien situada.

Tres familias de detectores

Dos etapas, una etapa y detectores con transformer organizan el problema de formas diferentes.



El árbol sirve como mapa. Nos centraremos en las ideas que separan las familias.

Las taxonomías cambian con el tiempo y hay modelos híbridos. Esta clasificación ayuda a situar los ejemplos del tema, no a decidir qué detector es mejor sin medirlo.

Cómo cambió el recorrido

Tres etapas cambiaron cómo se proponen, predicen y asignan las cajas.

2014–2016 · Dos etapas

R-CNN propone regiones y después las clasifica.

2015→ · Una etapa

YOLO predice cajas y clases en una sola pasada.

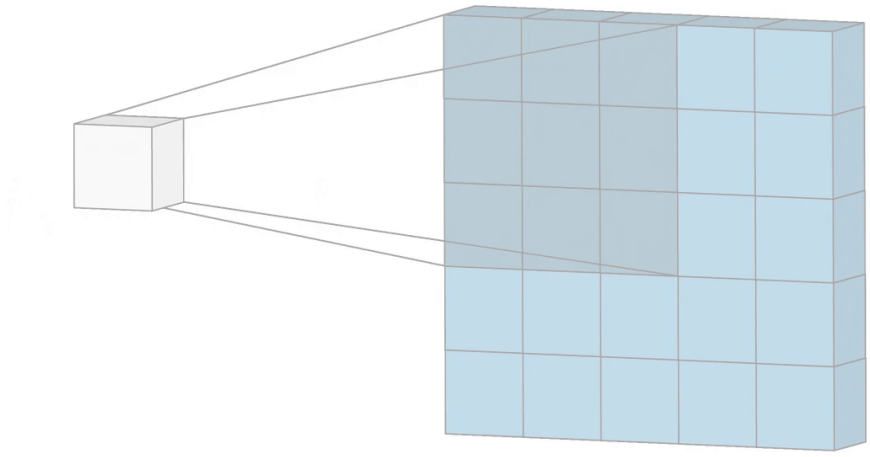
2020→ · Transformers

DETR aprende queries y asignación uno a uno.

Las fechas sitúan la aparición de las ideas, no el fin de cada familia. Los modelos de dos etapas, una etapa y transformer siguen coexistiendo, y hay diseños híbridos.

Patrones, capa a capa

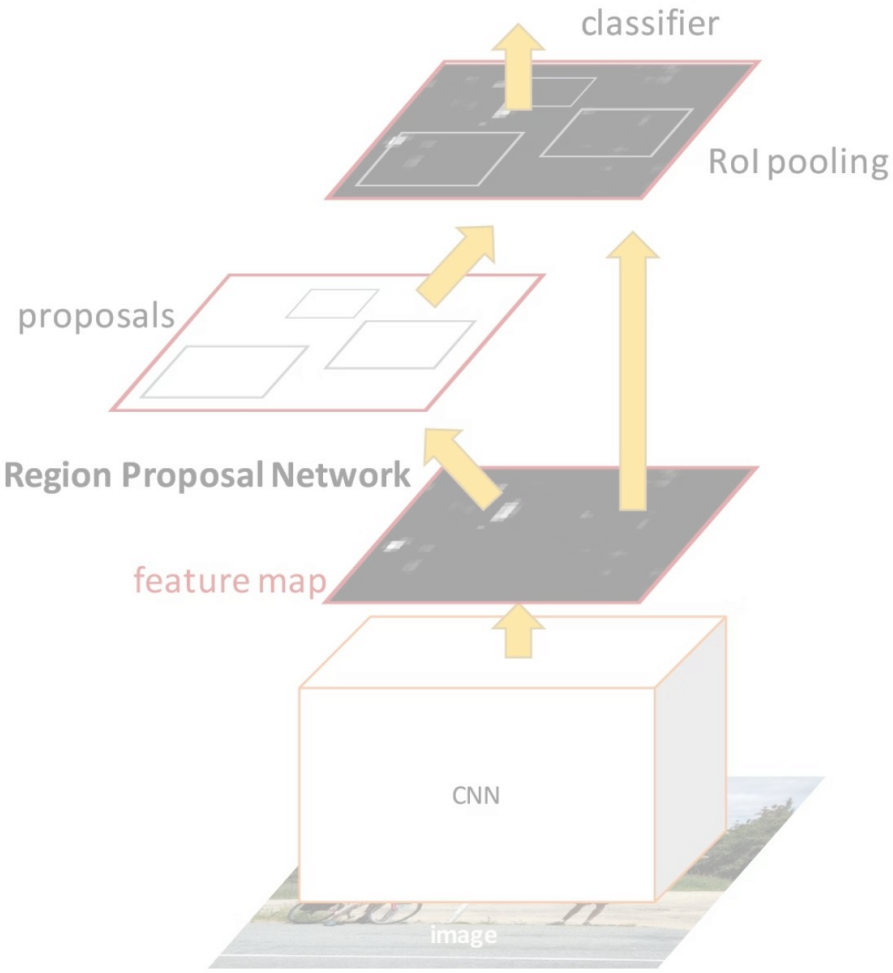
Las convoluciones extraen rasgos locales que capas posteriores combinan.



Las CNN incorporan localidad e invariancia aproximada a traslaciones. Siguen siendo componentes útiles en muchos detectores; el contexto global y la atención son decisiones de arquitectura, no una carencia universal de todas las CNN.

Primero regiones, después clases

Una RPN propone regiones; ROI Pooling reutiliza features; otra cabeza clasifica y refina.

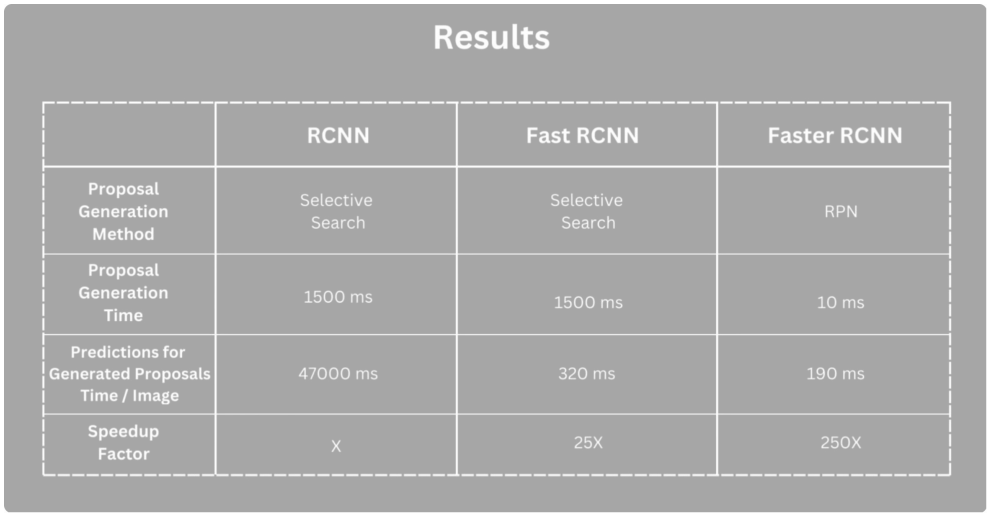


Más candidatos permiten refinar mejor, pero aumentan cálculo y latencia.

R-CNN, Fast R-CNN y Faster R-CNN son hitos de esta familia. La primera etapa puede producir cientos de candidatos. La segunda extrae features de cada región, asigna una clase y ajusta la caja. Es una opción razonable cuando la exactitud pesa más que la latencia, pero las cifras concretas dependen de implementación, datos y hardware.

Eliminar trabajo repetido

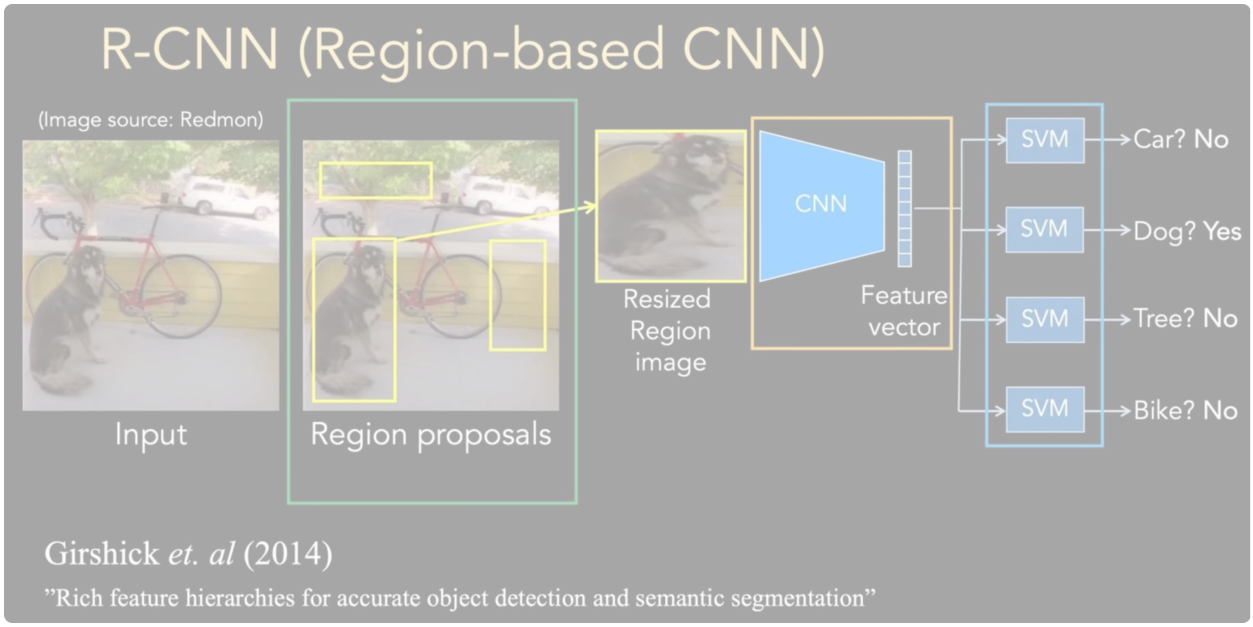
Cada cambio de diseño buscó compartir más cálculo entre las regiones de una imagen.



Comparación histórica: los tiempos solo se interpretan con su hardware y protocolo originales.

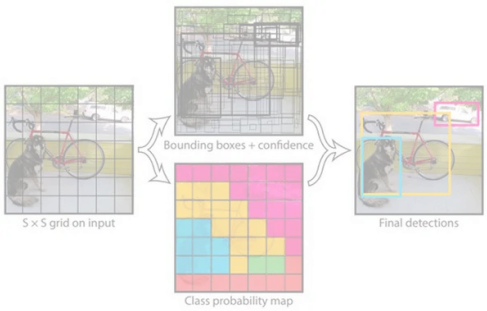
Una propuesta se convierte en candidato

R-CNN ilustra el recorrido de una región desde la imagen hasta la clase.



Una pasada, muchas predicciones

YOLOv1 popularizó la idea de producir cajas y clases directamente desde la imagen.



¿Qué describe la idea de un detector de una etapa?

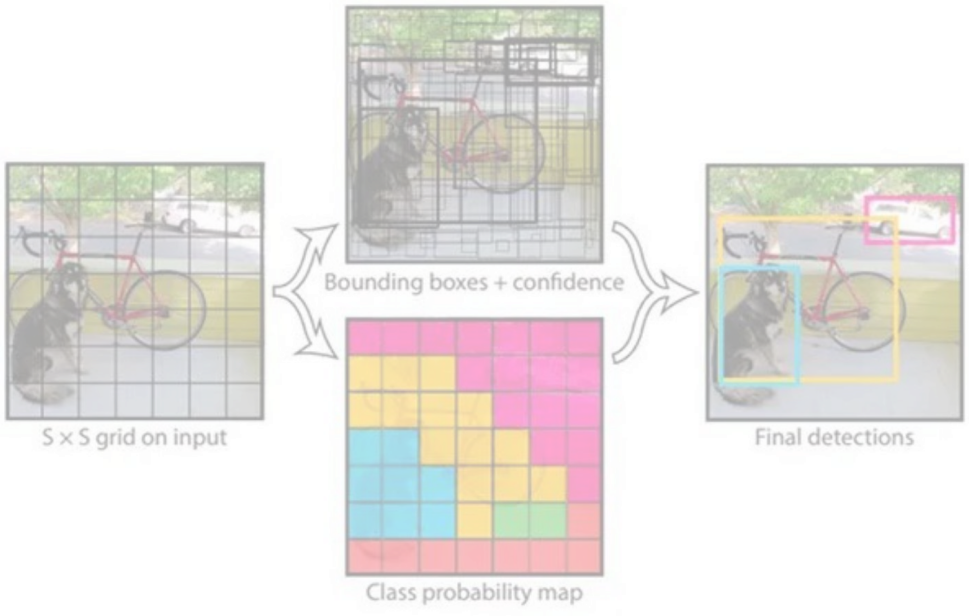
- A Propone regiones y las clasifica en un segundo recorrido.
- B No necesita imágenes de entrenamiento.
- C Predice localización y categoría en una misma ruta de detección.

Respuesta C. Exacto: esa es la idea central, aunque los detalles cambian entre versiones.

La cuadrícula S×S es una explicación histórica de YOLOv1. Las variantes posteriores cambiaron detalles importantes, por lo que no debe describirse toda la familia moderna con ese único diagrama.

La cuadrícula de YOLOv1

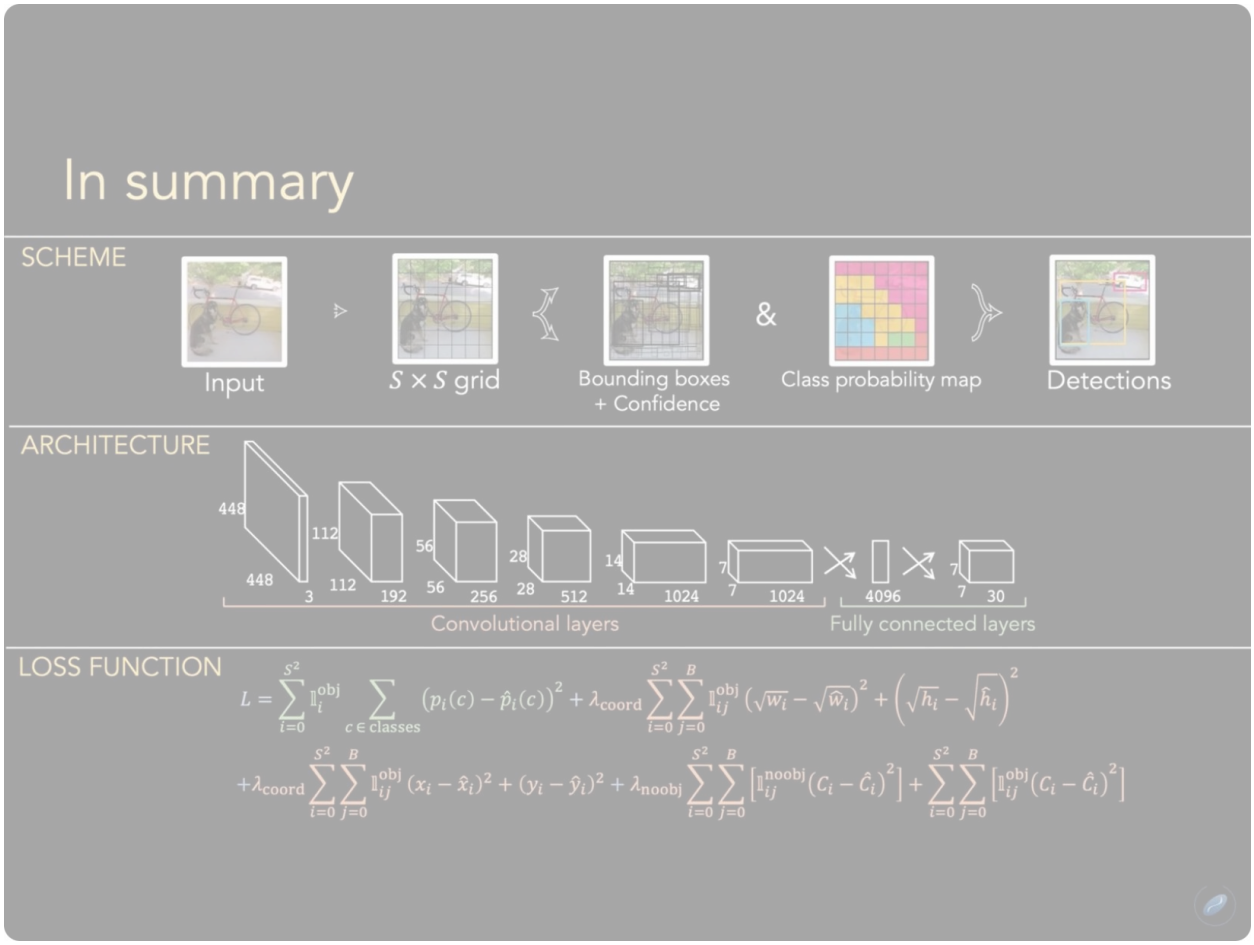
En la formulación original, una celda responde por objetos cuyo centro cae en ella.



Esquema histórico de YOLOv1, no una descripción de todas las versiones posteriores.

De la imagen a las cajas

YOLOv1 usa una cuadrícula y predice cajas, confianza y clases de forma conjunta.



Una sola CNN conecta la imagen completa con todas las predicciones.

En YOLOv1, la celda que contiene el centro del objeto asume su detección. Cada celda predice cajas, confianza y probabilidades de clase. Las versiones posteriores cambiaron asignación, backbones, objetivos y posprocesado; la cuadrícula es una referencia histórica, no una plantilla universal.

La familia siguió cambiando

Cada versión cambió una parte del entrenamiento, la arquitectura o la experiencia de uso.

2016 · v1
Predicción directa.

2018 · v3
Varias escalas.

2020 · v4
CSP y Mosaic.

2020 · v5
Flujo PyTorch.

2022 · v7
E-ELAN.

2023 · v8
Anchor-free y multitarea.

2024 · v9
PGI y GELAN.

La cronología resume hitos del material original. “YOLO” no identifica una única arquitectura ni una única organización. Las comparativas entre generaciones deben indicar implementación, checkpoint, resolución, hardware y protocolo; no se puede inferir el mejor modelo solo por el número de versión.

Qué intenta simplificar DETR

Las CNN aportan localidad; los detectores clásicos añaden heurísticas para relacionar y depurar predicciones.

Contexto

Las capas locales necesitan combinarse para relacionar zonas lejanas.

Sesgo

La localidad es útil, pero condiciona qué patrones se aprenden con facilidad.

Heurísticas

Anchors y NMS añaden reglas e hiperparámetros al pipeline.

Estas propiedades no convierten a toda CNN en insuficiente: son decisiones con ventajas y costes. DETR propone otra formulación basada en atención, queries y predicción de conjuntos para optimizar el detector de extremo a extremo y evitar anchors y NMS en su diseño original.

Consultas para objetos

DETR predice un conjunto de candidatos a partir de características visuales y consultas aprendidas.

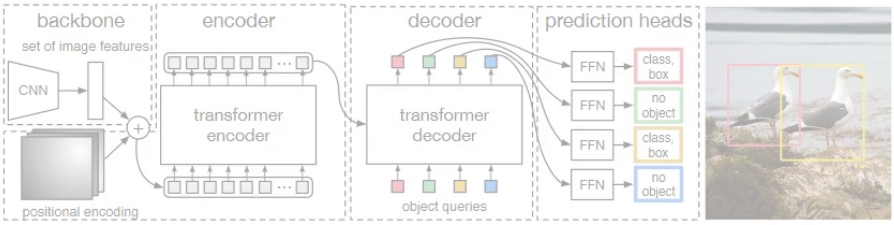


Fig. 2: DETR uses a conventional CNN backbone to learn a 2D representation of an input image. The model flattens it and supplements it with a positional encoding before passing it into a transformer encoder. A transformer decoder then takes as input a small fixed number of learned positional embeddings, which we call *object queries*, and additionally attends to the encoder output. We pass each output embedding of the decoder to a shared feed forward network (FFN) that predicts either a detection (class and bounding box) or a "`no object`" class.

Una consulta puede acabar asociada a una caja y una clase.

El diseño original utiliza un encoder y un decoder transformer. Los detalles concretos, como número de consultas o backbone, pertenecen a una versión específica.

Asignar antes de aprender

Durante el entrenamiento, DETR relaciona predicciones y objetos reales uno a uno.

¿Para qué sirve la asignación bipartita en DETR durante el entrenamiento?

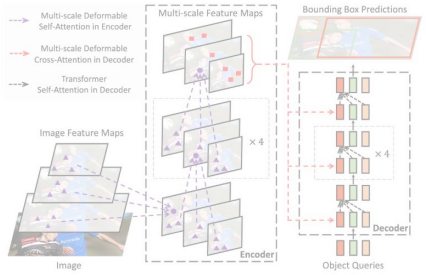
- A Relacionar predicciones y objetos reales uno a uno al calcular la pérdida.
- B Aplicar NMS a las cajas de cada imagen al predecir.
- C Crear automáticamente etiquetas para entrenar.

Respuesta A. Exacto: define qué objetivo supervisa cada predicción.

La asignación bipartita define objetivos para calcular la pérdida durante entrenamiento. No es el mismo proceso que aplicar supresión de no máximos durante inferencia.

Del backbone a las cajas

RF-DETR combina features multiescala, atención deformable y queries que se refinan.



01 **Combina** features de varias escalas.

02 **Alinea** puntos relevantes con atención deformable.

03 **Consulta** el encoder mediante object queries.

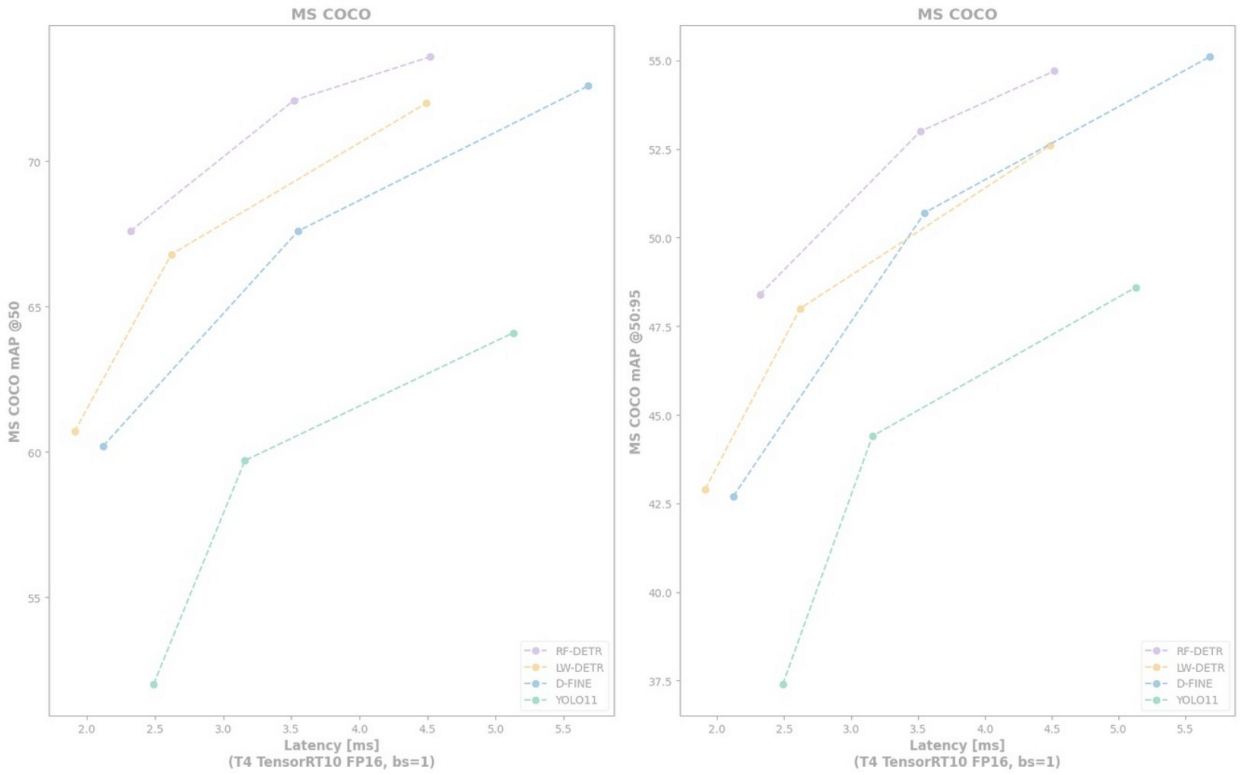
04 **Refina** las predicciones en el decoder.

05 **Entrega** cajas sin anchors ni NMS.

La descripción del material usa un backbone visual preentrenado de la familia DINOv2, features multiescala, atención deformable y un decoder que refina iterativamente las queries. La arquitectura exacta depende de la variante y del checkpoint cargado por el notebook.

Elegir requiere condiciones

Calidad, latencia, licencia y mantenimiento solo se comparan con versión, datos y hardware explícitos.

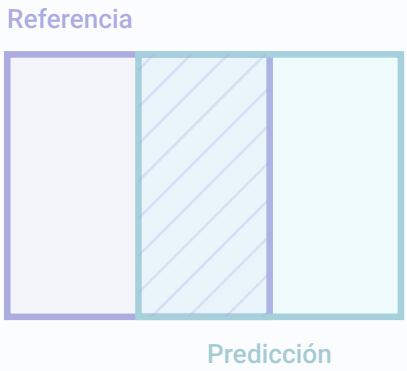


Una gráfica tiene sentido solo con su protocolo de medida y versiones concretas.

RF-DETR es la elección docente de este curso. Eso no demuestra que sea el mejor detector para cada producto: la decisión requiere una evaluación en el dominio, condiciones de despliegue y revisión de licencias de la versión concreta.

¿Cuánto coinciden las cajas?

IoU = área compartida ÷ área total ocupada.



IoU

0.33

Solapamiento parcial.

IoU vale 1 cuando ambas cajas coinciden y 0 cuando no comparten área. Para decidir TP, FP y FN también se tiene en cuenta la clase, la correspondencia uno a uno y un protocolo de evaluación.

Contar aciertos y fallos

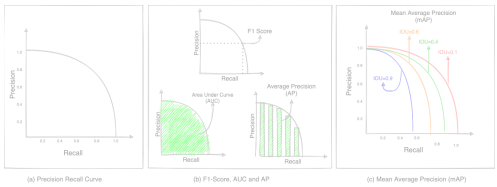
Una predicción puede acertar, duplicar un objeto o dejar un objeto sin detectar.

TP	FP	FN
Caja y clase que corresponden a un objeto real.	Falsa alarma o predicción duplicada.	Objeto real que quedó sin predicción.

En detección no se suele usar un recuento práctico de verdaderos negativos de fondo. Las ubicaciones sin objetos son enormes y no se enumeran como predicciones negativas individuales.

Precisión y recall responden a preguntas distintas

La precisión mira las predicciones; el recall mira los objetos reales.



(a) Precision Recall Curve (b) F1-Score, AUC and AP (c) Mean Average Precision (mAP)

Precisión = TP / (TP + FP). Recall = TP / (TP + FN).

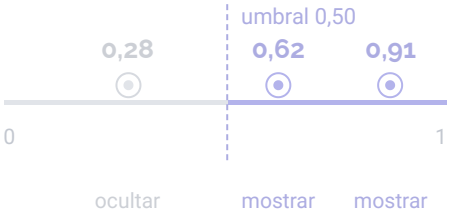
Hay 3 TP, 1 FP y 2 FN. ¿Cuáles son precisión y recall?

- A Precisión 0,60; recall 0,75.
- B Precisión 0,75; recall 0,75.
- C Precisión 0,75; recall 0,60.

Respuesta C. Exacto: 3/(3+1) y 3/(3+2).

Filtrar no cambia las cajas

El umbral de confianza decide qué predicciones se muestran; IoU evalúa su solapamiento.



El filtro cambia qué predicciones ves, no sus coordenadas.

Subes el umbral de confianza y mantienes fijas las predicciones candidatas.

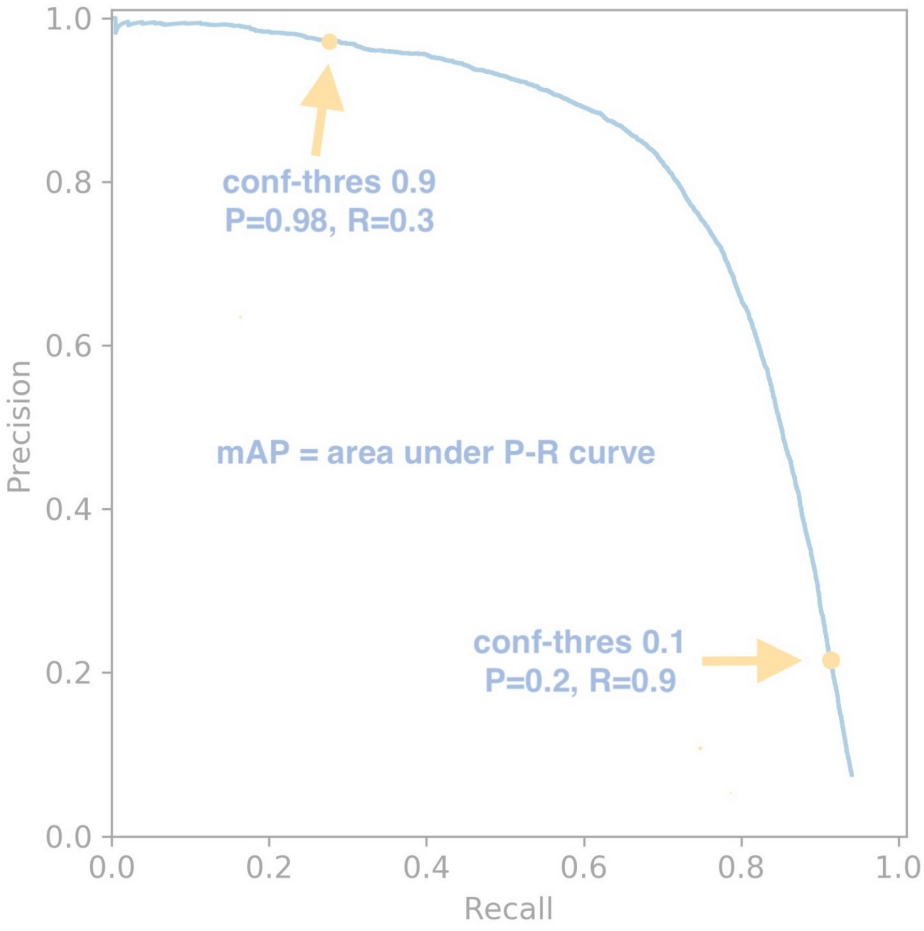
- A Se conservan las mismas o menos cajas.
- B Aparecen categorías nuevas.
- C Aumenta necesariamente la IoU de todas las cajas restantes.

Respuesta A. Exacto: el filtro elimina las puntuaciones que quedan por debajo del nuevo umbral.

Al subir el umbral suelen desaparecer predicciones de baja puntuación: pueden bajar los falsos positivos y también aumentar los falsos negativos. La curva precisión-recall estudia ese intercambio recorriendo muchos umbrales.

Una curva resume umbrales

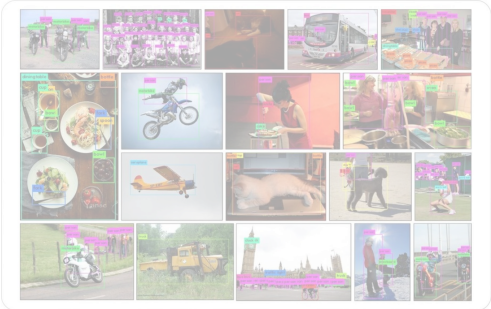
AP resume precisión y recall al recorrer puntuaciones. mAP agrega AP según un protocolo.



AP50 y AP promediada entre varios umbrales IoU responden a protocolos distintos.

El vocabulario depende de los datos

Un checkpoint aprende las clases de sus datos. Tu aplicación puede necesitar otras.



Tu checkpoint solo conoce clases COCO y necesitas detectar un defecto industrial nuevo.

- A Bajar mucho el umbral para crear la categoría.
- B Evaluar adaptación con datos etiquetados o un modelo adecuado al nuevo vocabulario.
- C Aumentar el tamaño de las cajas existentes.

Respuesta B. Exacto: la nueva categoría exige una estrategia y una evaluación en el dominio.

Mide donde vas a usarlo

Un benchmark no sustituye datos representativos, latencia medida y análisis de errores.

Datos que representen escenas, clases y condiciones reales.

Latencia medida en el sistema de destino.

Errores revisados: falsas alarmas, objetos omitidos y casos límite.

Un detector logra mejor AP en un benchmark publicado. ¿Basta para elegirlo?

- A Sí, cualquier mejora de AP garantiza mejor producción.
- B Sí, si es el modelo más reciente.
- C No: hay que medir calidad y latencia con datos y condiciones representativos.

Respuesta C. Exacto: la elección depende del problema, del entorno y de los errores que aceptas.

Ahora, con tu propia imagen

Prueba RF-DETR en tu copia del cuaderno.

1. Guarda una copia en Drive.
2. Ejecuta el detector con una imagen.
3. Cambia el umbral de confianza: **0,25 · 0,50 · 0,75**.

Un detector cerrado reconoce sus clases; el siguiente bloque añadirá lenguaje y zero-shot.

Abrir en Colab ↗

En Colab: Archivo → Guardar una copia en Drive. Empieza por inferencia. La disponibilidad de GPU varía; el entrenamiento necesita además el dataset indicado en el notebook. La comparación entre 0,25, 0,50 y 0,75 permite observar qué predicciones desaparecen sin confundir confianza con IoU. El siguiente tema amplía el vocabulario cerrado mediante modelos que relacionan visión y lenguaje.