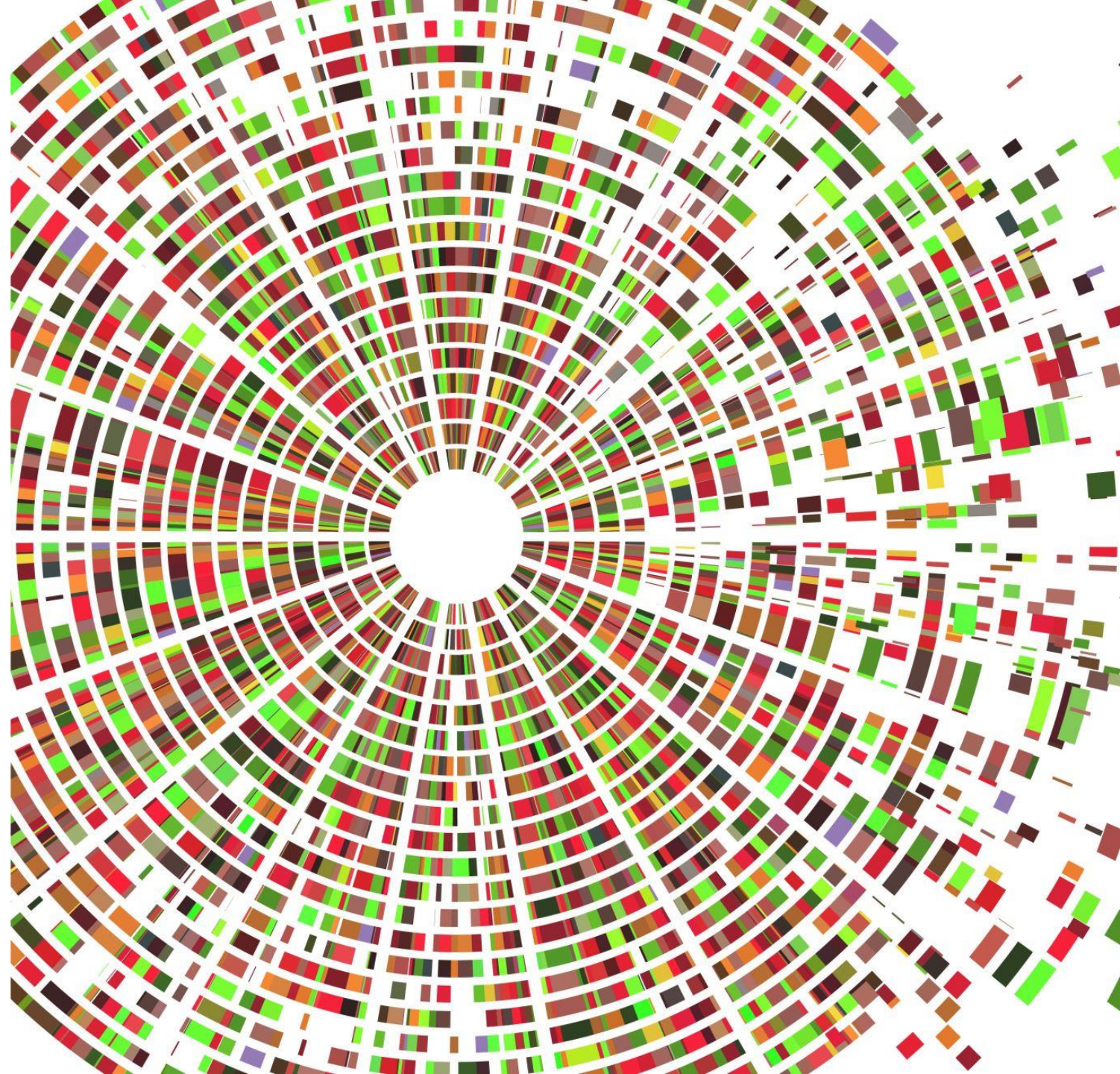


# Métodos avanzados de ciencia de datos

Prof. Emily Díaz



# Contenido



Segmentación de imágenes



Detección de objetos

# Segmentación y detección



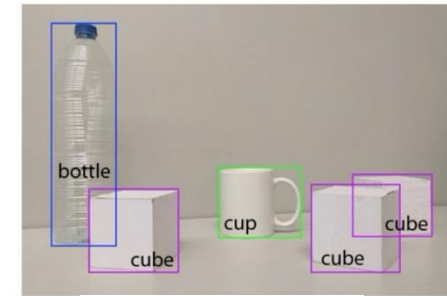


# Segmentación de imágenes

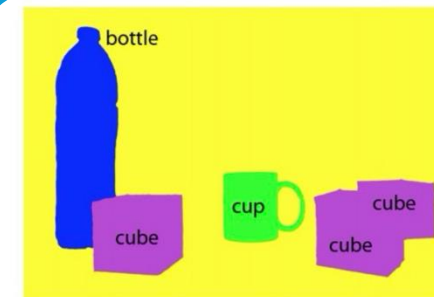
- Implica dividir una imagen en regiones o segmentos distintos, donde cada **segmento corresponde a un objeto o parte particular de la imagen**.
- A diferencia de la clasificación de imágenes tradicional, donde un modelo asigna una sola etiqueta a una imagen completa, **la segmentación asigna una etiqueta a cada píxel de la imagen**.
- Se tiende a utilizar **CNN** para identificar automáticamente los límites y las regiones de una imagen, lo que produce una predicción píxel por píxel que puede distinguir diferentes objetos, fondos o regiones.
- Los dos tipos más comunes:
  - **Semántica**: A cada píxel de la imagen se le asigna una etiqueta de clase, pero no hay distinción entre diferentes instancias de la misma clase.
  - **Instancia**: Identifica diferentes instancias del mismo objeto.



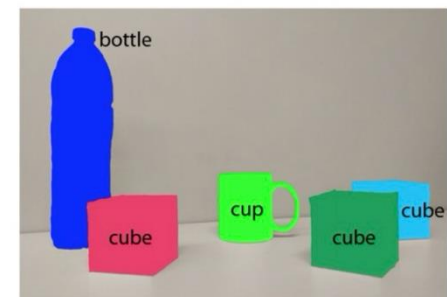
(a) **Clasificación**



(b) **Detección**



(c) **Segmentación semántica**



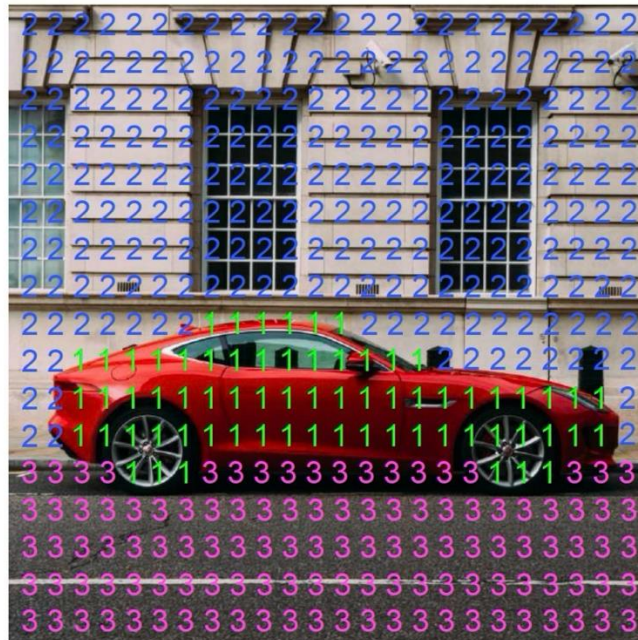
(d) **Segmentación de instancia**

# Segmentación de imágenes

- A diferencia de la clasificación, que simplemente indica qué objetos están presentes, la segmentación **proporciona la ubicación exacta y la forma de los objetos en la imagen**, lo que la hace mucho más informativa para aplicaciones del mundo real.
- Algunas de las arquitecturas más usadas para segmentación:
  - U-Net
  - Mask R-CNN
  - SegNet
  - Transformers
- Es una herramienta esencial para industrias como la atención médica, la conducción autónoma y el análisis de imágenes satelitales.



# ¿Cuál es la variable respuesta en este caso?

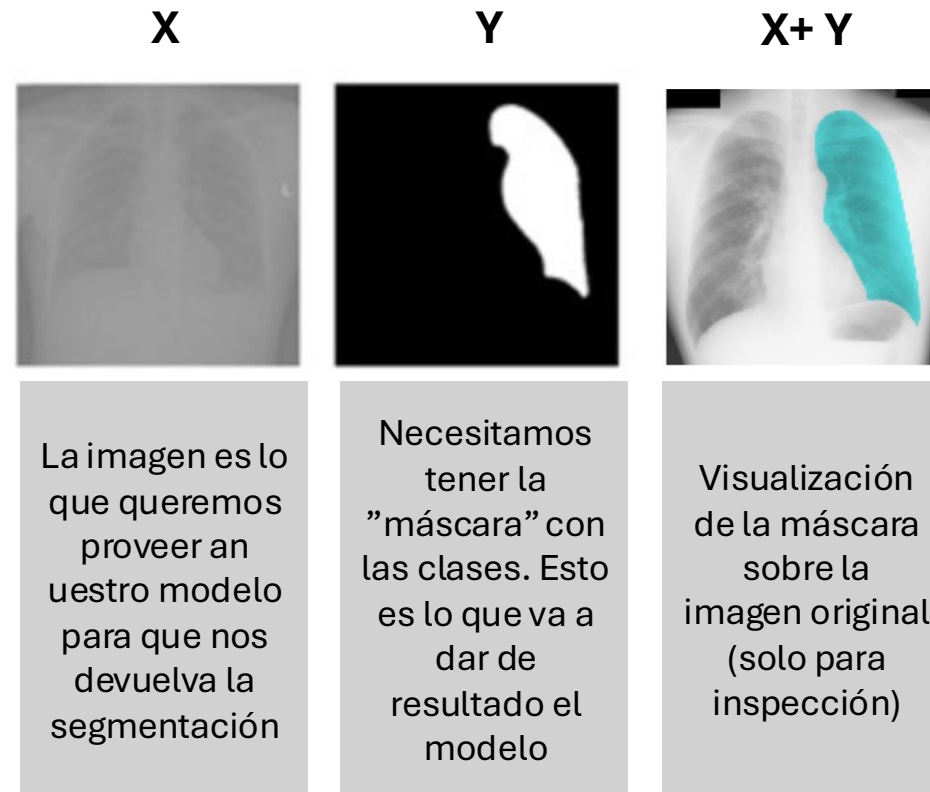


- 1 – Car
- 2 – Building
- 3 – Road



A segmentation map

# Segmentación: Clasificación a nivel de pixel



- Puede ser una máscara binaria (0s y 1s) o multi-clase (cada pixel tiene un número de clase asignado)

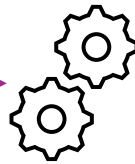
# Clasificación a nivel de pixel

Datos de entrada

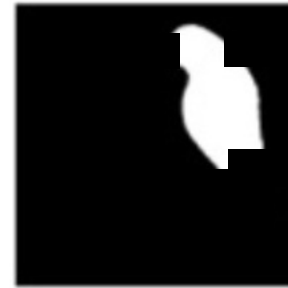


Entrenamiento

Red  
Neuronal



Resultado (mapa  
de clases  
predicho)



Para cada pixel, busca la probabilidad más alta de cada clase y así lo clasifica



Para cálculo del  
error y mejora de  
la red

Mapa de clases  
real

¿De qué tamaño es el mapa de clases?



# DICE: Métricas de rendimiento y función de pérdida

DICE (DSC): **Medida de superposición entre dos conjuntos**, da más peso a la superposición entre los píxeles predichos y los de real al considerar tanto el tamaño de la intersección como el tamaño total de los conjuntos.

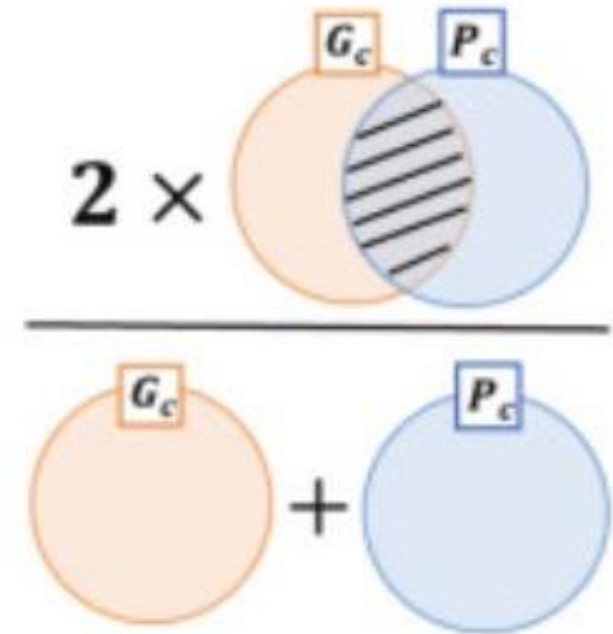
**Un DICE de 1 es traslape perfecto (bueno) y de 0 nada de superposición (malo)**

$$DSC = \frac{2|A \cap B|}{|A| + |B|}$$

$|A|$  = es el número de píxeles en la segmentación predicha

$|B|$  = es el número de píxeles en la segmentación real

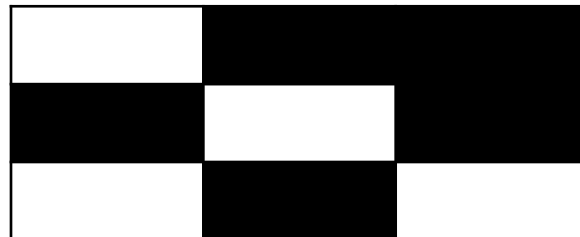
$|A \cap B|$  = es el número de píxeles en la intersección entre las regiones predichas y de la verdad fundamental.


$$DSC_c = \frac{2 \times \text{Intersection}(G_c, P_c)}{|G_c| + |P_c|}$$

# Ejemplo de cálculo DICE – Mapa de etiquetas binario

**Valor real**

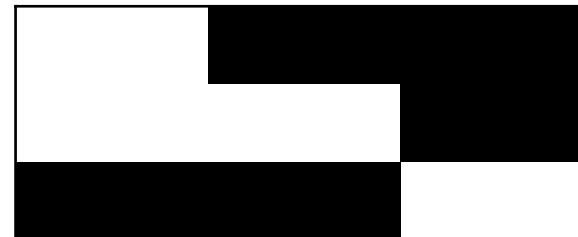
<b>1</b>	<b>0</b>	<b>0</b>
<b>0</b>	<b>1</b>	<b>0</b>
<b>1</b>	<b>0</b>	<b>1</b>



**Visto como  
máscara/imagen**

**Predicho**

<b>1</b>	<b>0</b>	<b>0</b>
<b>1</b>	<b>1</b>	<b>0</b>
<b>0</b>	<b>0</b>	<b>1</b>



**Visto como  
máscara/imagen**

**Dice**

$$DSC = \frac{2|A \cap B|}{|A| + |B|}$$

$$DSC = \frac{2 * 3}{4 + 4} = \frac{6}{8} = 0.75$$

*Clase “0” no se cuenta como  
acierto. Es llamada  
“background”/“fondo”*

# Ejemplo de cálculo DICE con varias clases

**Valor real**

<b>0</b>	<b>1</b>	<b>1</b>
<b>0</b>	<b>2</b>	<b>0</b>
<b>2</b>	<b>0</b>	<b>1</b>



**Visto como  
máscara/imagen**

**Predicho**

<b>0</b>	<b>1</b>	<b>0</b>
<b>2</b>	<b>2</b>	<b>0</b>
<b>2</b>	<b>0</b>	<b>1</b>



**Visto como  
máscara/imagen**

$$DSC_1 = \frac{2 * 2}{3 + 2} = \frac{4}{5} = 0.8$$

$$DSC_2 = \frac{2 * 2}{2 + 3} = \frac{4}{5} = 0.8$$

$$DSC_{media} = \frac{0.8 + 0.8}{2} = 0.8$$

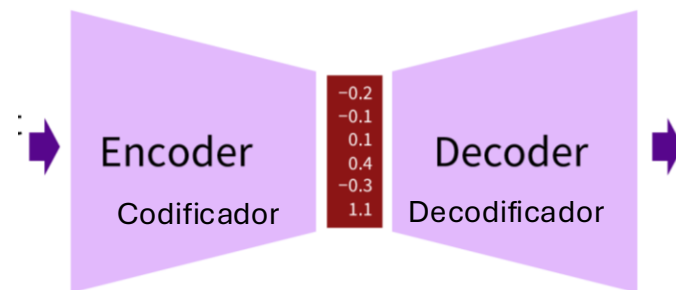
*Calculamos por separado cada clase y la luego agregamos con la media. También podría usarse ponderación si existe un criterio*

# Comparación de clasificación y segmentación

	Clasificación	Segmentación
Objetivo	Asignar <b>una etiqueta</b> a toda la imagen	Asignar una etiqueta para <b>cada pixel</b> en la imagen
Resultado	Probabilidades de cada clase o etiqueta predicha	Clasificación píxel por píxel (etiqueta o probabilidad para cada píxel)
Arquitectura	Típicamente CNNs: VGG, ResNet, etc	Codificador-decodificador: U-Net, FCN, DeepLab, etc
Función de pérdida	Entropía cruzada categórica	Entropía cruzada categórica o pérdida <b>DICE</b>

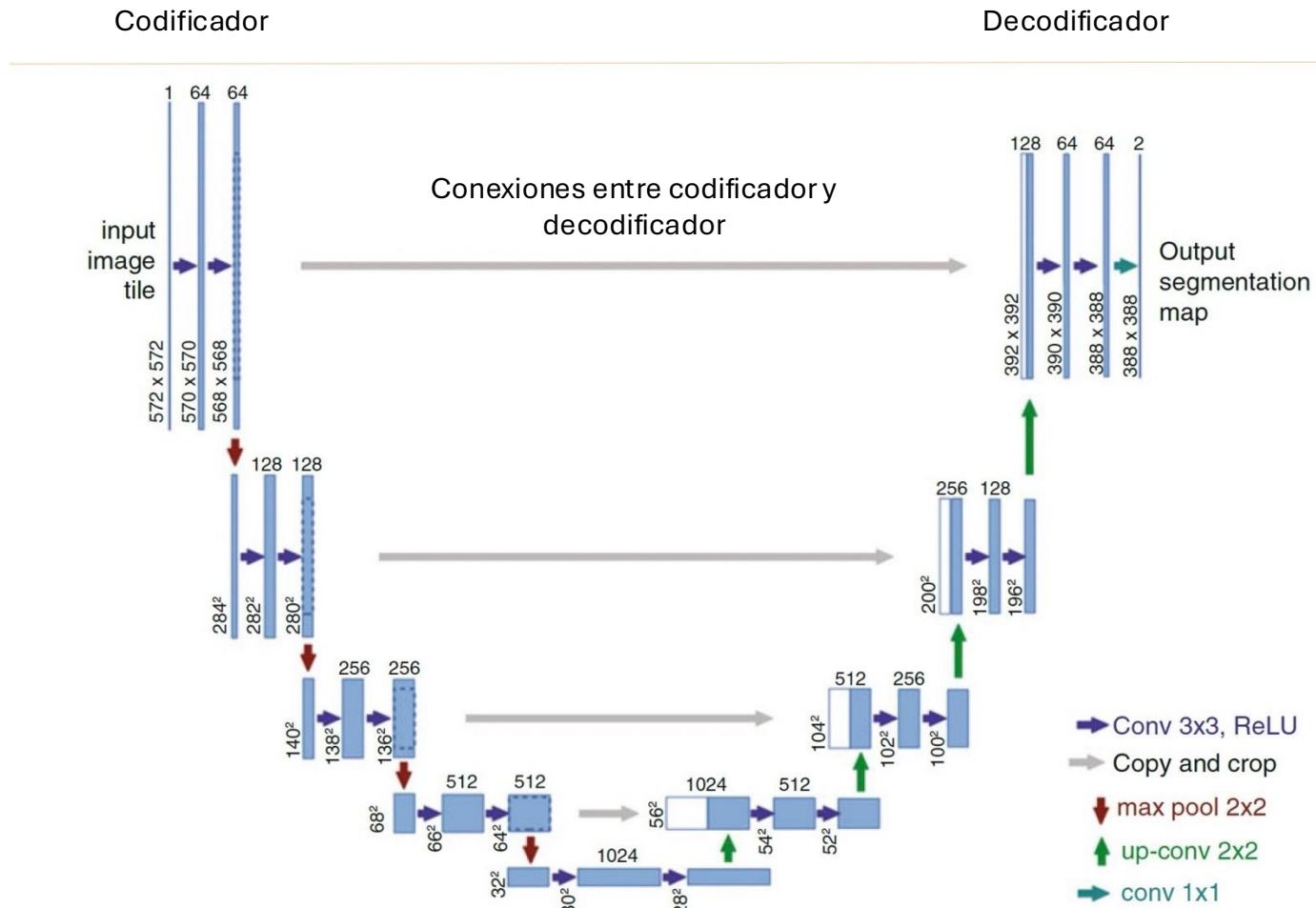


# Arquitecturas para segmentar tiende a tener 3 componentes



- **Codificador:** al igual que las redes de clasificación, extrae características a través de capas convolucionales y de agrupamiento.
- **Decodificador:** aumenta el tamaño de las características hasta el tamaño de la imagen de entrada original, lo que crea predicciones a nivel de píxel.
- **Conexiones de salto:** muchos modelos de segmentación (por ejemplo, U-Net) **utilizan conexiones de salto para retener información de alta resolución al pasar directamente las características del codificador al decodificador. Retiene información espacial**

# Estructura de U-Net



U-Net architecture. Blue boxes are the feature maps. Channel numbers are denoted above each box, while the tensor sizes are denoted on the lower left. White boxes show the concatenations and arrows indicate various operations.

Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Springer, Berlin,

# Ventajas y desventajas de U-net

## **Ventajas**

Flexible y utilizable con la mayoría de tareas que tengan máscaras

Da buen rendimiento en general (al menos de base)

Poderosa en escenarios con limitado número de datos

## **Desventajas**

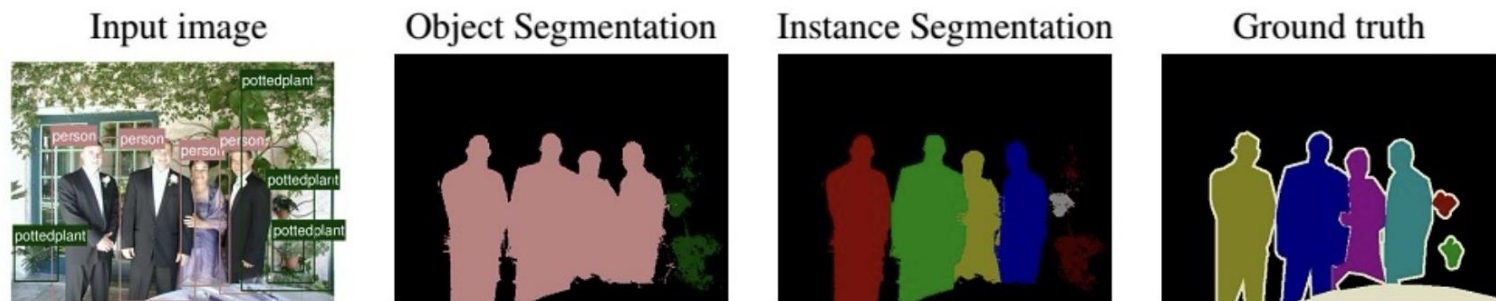
Imágenes muy grandes necesitan alta memoria de GPU

Toma tiempo significativo de entrenamiento

No hay muchos modelos pre-entrenados con esta arquitectura

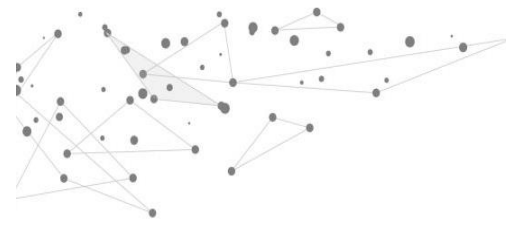
# Diferencias de segmentación de instancia

- **Problema:** el número de instancias es desconocido
- La evaluación no se realiza a nivel de píxeles
- Se crean varias máscaras, una para cada instancia de objeto individual. Cada píxel pertenece a un objeto específico y se etiqueta con su clase (por ejemplo, "perro", "automóvil") y su número de instancia (por ejemplo, persona 1, persona 2).
- El tipo de red neuronal tiende a ser distinta, por ejemplo: Mask R-CNN
- Para la función de pérdida utiliza una combinación de:
  - Pérdida de detección de objetos (regresión y clasificación de cuadro delimitador).
  - Pérdida de máscara de segmentación (entropía cruzada binaria) aplicada a cada objeto detectado.

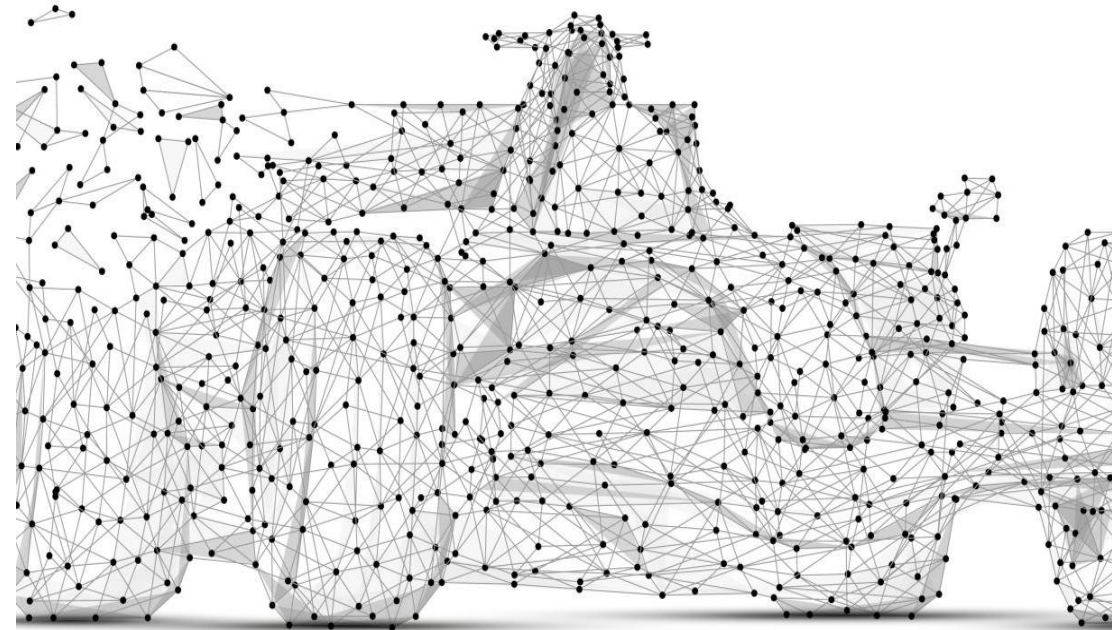




# Otros modelos más avanzados en este tema



- Arquitecturas **Transformers** han revolucionado las redes neuronales en todas sus aplicaciones, incluyendo segmentación
- En los últimos años, **DINO y SAM** han surgido como modelos transformers innovadores con distintos propósitos en términos de aplicaciones y tareas.
- **DINO (Distillation with No Labels)** es un modelo de **aprendizaje autosupervisado** para tareas de visión basado en **Vision Transformers (ViTs)**.
  - Está diseñado para aprender **representaciones visuales enriquecidas** a partir de imágenes **sin utilizar ningún dato etiquetado**.
  - Una vez entrenado, **las representaciones aprendidas se pueden aplicar a diversas tareas**, como clasificación de imágenes, detección de objetos y segmentación semántica, **incluso con un ajuste fino mínimo**.
- **SAM (Segment Anything Model)**: Introducido por Meta AI, es un modelo de segmentación generalizado **capaz de segmentar objetos en imágenes según indicaciones del usuario o de forma automática**.
  - La idea clave es que SAM puede segmentar cualquier objeto en una imagen con una supervisión mínima (por ejemplo, utilizando un punto, un cuadro o una indicación de texto de forma libre), de ahí el nombre "Segmentar cualquier cosa".

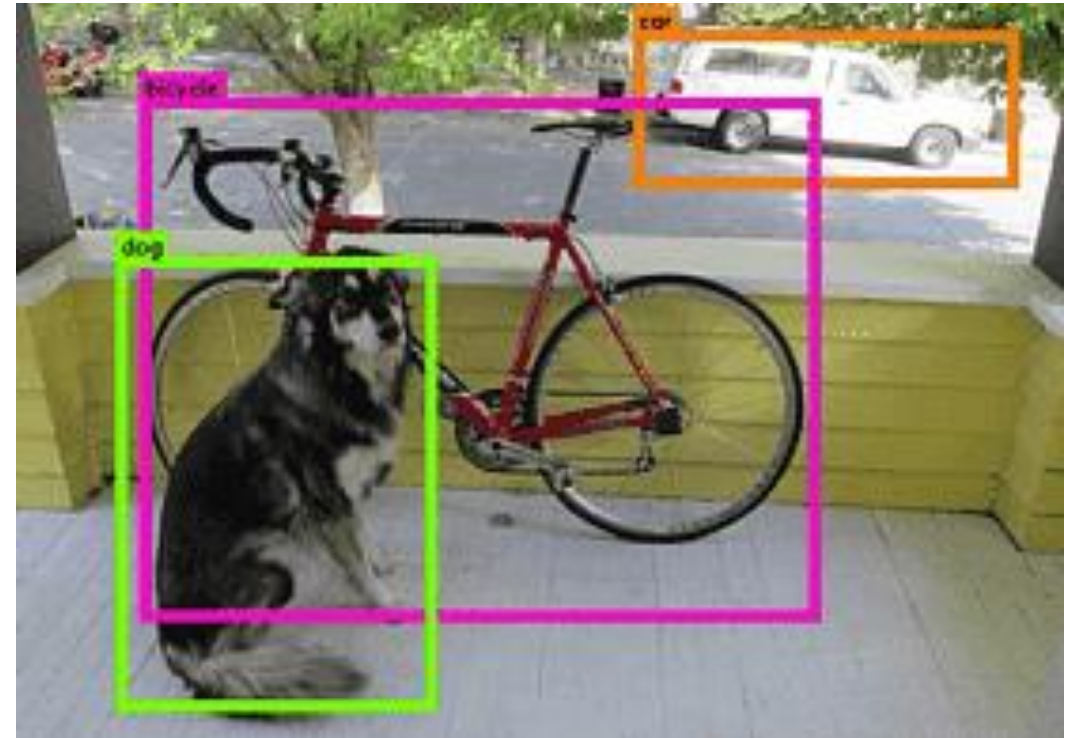




# Detección de objetos

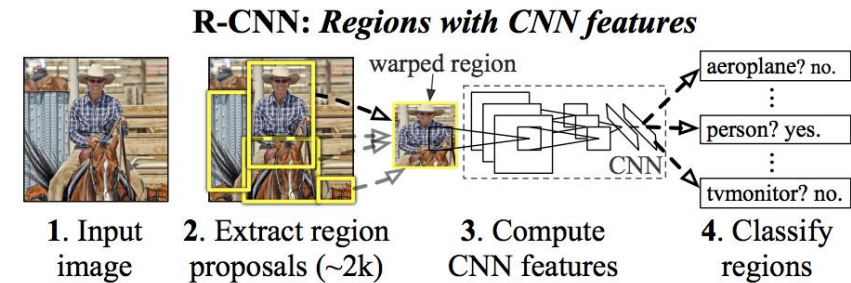
# Qué es detección de objetos?

- La detección de objetos implica identificar y localizar objetos dentro de una imagen.
- No solo clasifica los objetos, sino que también proporciona su ubicación espacial dibujando cuadros delimitadores a su alrededor.
- Básicamente, el objetivo de la detección de objetos es doble:
  1. **Clasificación:** identificar la categoría de cada objeto (p. ej., perro, automóvil, persona).
  2. **Localización:** determinar dónde se encuentra cada objeto en la imagen mediante la predicción de cuadros delimitadores.



# Arquitecturas R-CNN: Region-based Convolutional Neural Network

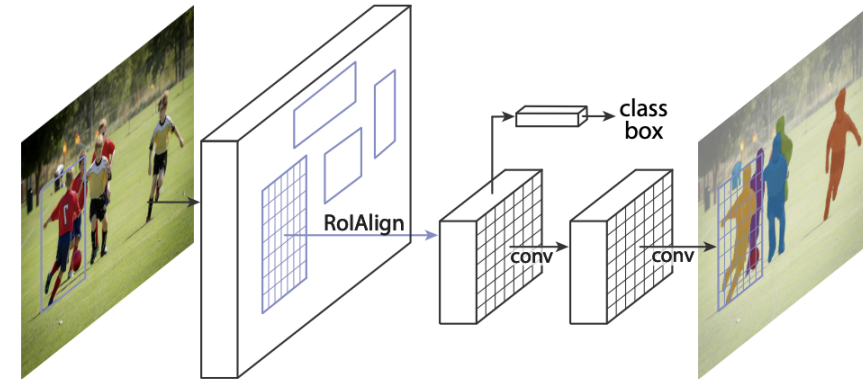
- La familia de arquitecturas son enfoques fundamentales para la **detección de objetos** y, posteriormente, la **segmentación de instancias**.
- R-CNN (2014) fue uno de los primeros enfoques basados en CNN para la detección de objetos. Introdujo la idea de utilizar la **búsqueda selectiva para proponer regiones de interés (ROI)** a partir de la imagen y luego **clasificar estas regiones utilizando una red neuronal convolucional**.
- Pasos de R-CNN
  - **Propuesta de región:** Búsqueda selectiva genera aproximadamente 2000 propuestas de región a partir de la imagen (posibles cuadros delimitadores que podrían contener objetos).
  - **Extracción de características:** para cada propuesta de región, aplica una CNN (como AlexNet) para extraer vectores de características de longitud fija.
  - **Clasificación:** estos vectores de características se introducen en las SVM (máquinas de vectores de soporte) para clasificar los objetos.
  - **Regresión de cuadro delimitador:** un modelo de regresión independiente ajusta las coordenadas del cuadro delimitador para lograr un mejor ajuste.



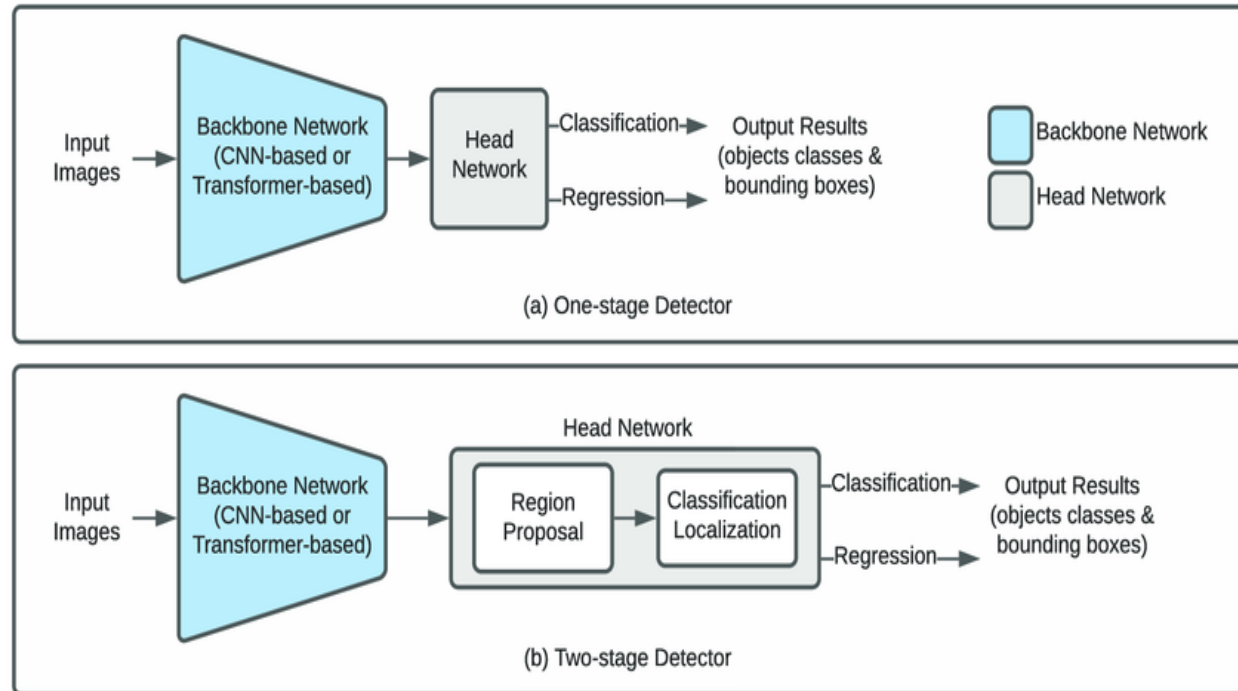


# Arquitecturas R-CNN: Region-based Convolutional Neural Network

- **Faster R-CNN (2016):** Introdujo varias innovaciones que redujeron tanto el costo computacional como el uso de memoria y consolidó todo el proceso de detección en una única red más eficiente. **Introdujo una red de propuestas de región (RPN) completamente aprendible**, eliminando la necesidad de un método de propuesta de región externo y lento como la búsqueda selectiva.
- **Mask R-CNN (2017):** Amplía Faster R-CNN añadiendo una rama para la **segmentación de instancias**. Mientras que Faster R-CNN detecta objetos y proporciona cuadros delimitadores, **Mask R-CNN también genera una máscara por píxel para cada objeto detectado**, lo que permite la segmentación de instancias.
- Entre estos modelos, **Mask R-CNN es el más reciente y potente**, especialmente para tareas que requieren tanto la detección de objetos como la segmentación de instancias. **Se utiliza ampliamente en aplicaciones que necesitan precisión de píxeles** (por ejemplo, imágenes médicas, conducción autónoma), **pero puede ser más lento en comparación con los detectores en tiempo real**.

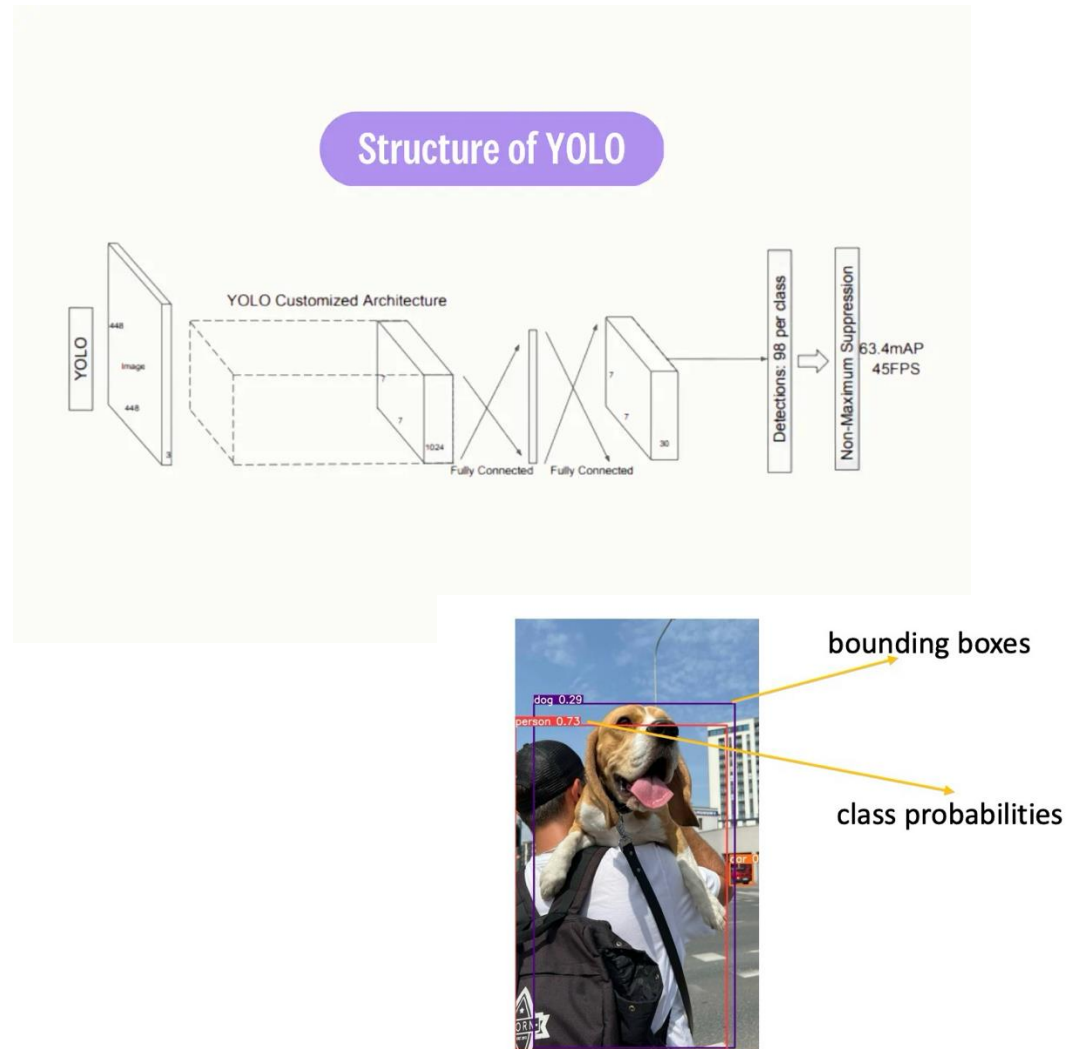


# Métodos de 1 etapa vs 2 etapas



- La principal diferencia entre las redes de detección de objetos de 1 etapa y de 2 etapas radica en **cómo manejan el proceso de proponer regiones de objetos y realizar la clasificación y localización** (regresión del cuadro delimitador).
- Ejemplo de dos etapas: Familia R-CNN
- Ejemplo de una etapa: YOLO
- **Los de una etapa son más rápidos pero menor rendimiento.**
- **Los de una etapa son mejores candidatos para aplicaciones en tiempo real**

# Modelo YOLO: You Only Look Once

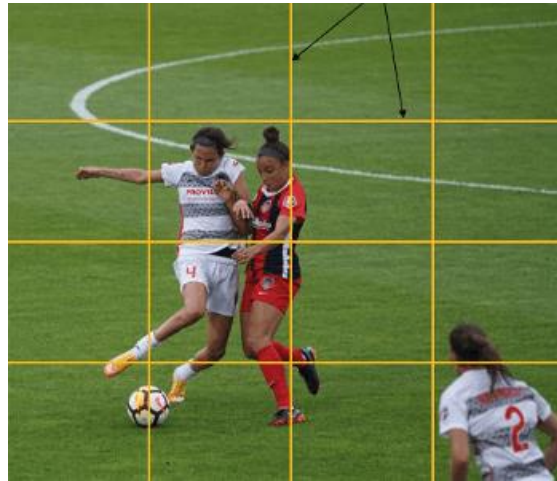


- Yolo fue propuesto en 2015 y ha evolucionado a distintas versiones hasta la 10 en mayo del 2024
- Enmarca el problema de detección de objetos como **una regresión** en lugar de una tarea de clasificación, **separando espacialmente cuadros delimitadores** y asociando **probabilidades a cada imagen detectada** utilizando una única **red neuronal convolucional (CNN)**.
- La arquitectura de YOLO tiene 24 capas convolucionales, cuatro capas de agrupación máxima y dos capas completamente conectadas.

# Pasos YOLO

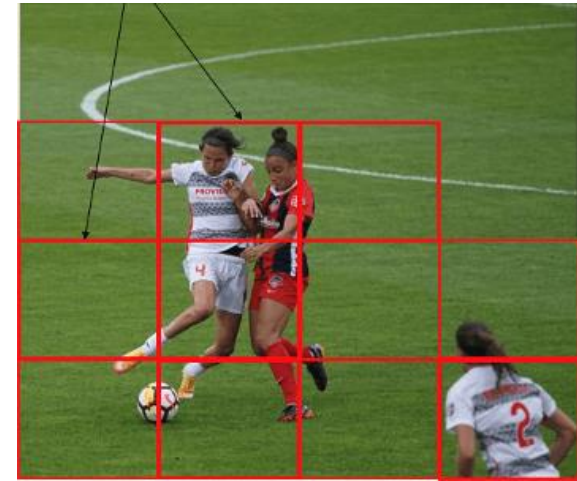


Imagen de entrada



Bloques residuales

Cada celda de la cuadrícula es responsable de localizar y predecir la clase del objeto que cubre, junto con el valor de probabilidad/confianza.

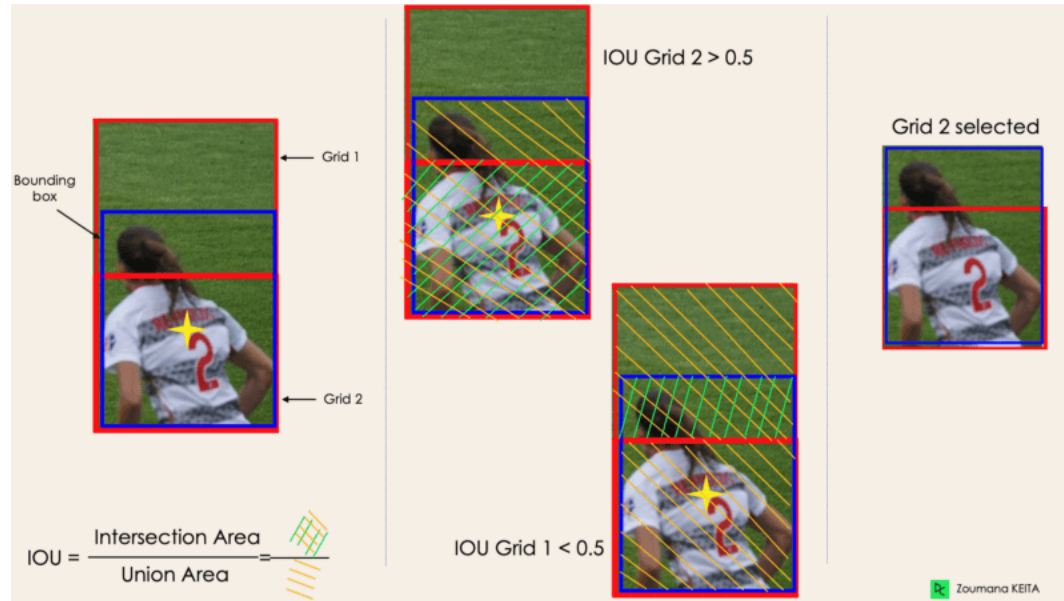


Regresión de cuadro delimitador

Determina los cuadros delimitadores correspondientes a los rectángulos, resaltando todos los objetos de la imagen.

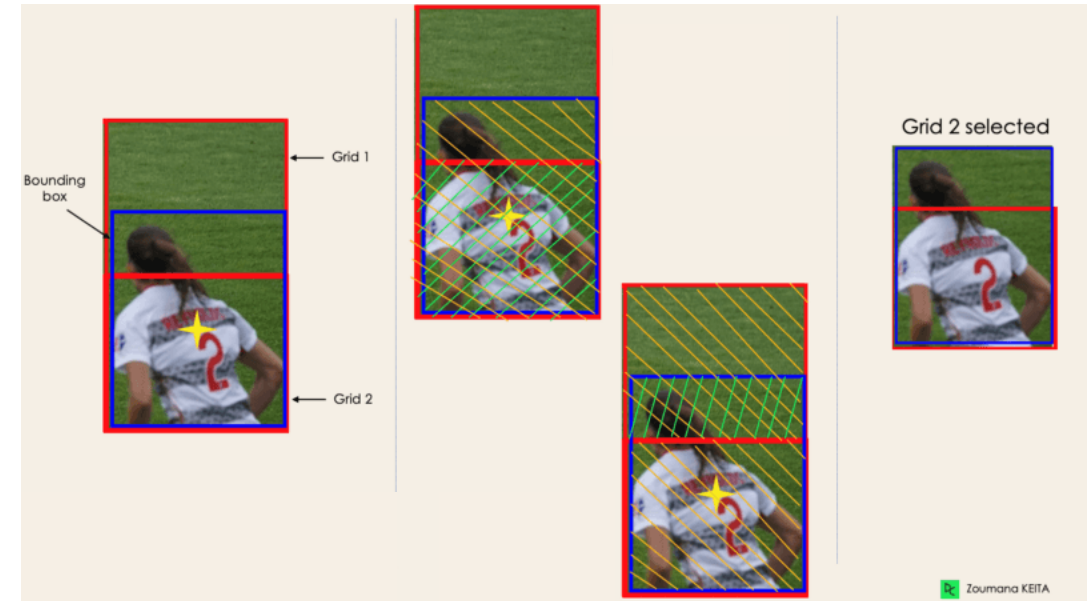


# Pasos YOLO



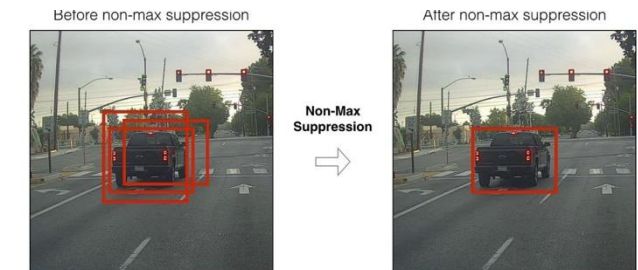
## Intersección sobre unión (IOU)

Descarta cuadros de cuadrícula para conservar solo aquellos que sean relevantes mediante un punto de corte basado en IOU



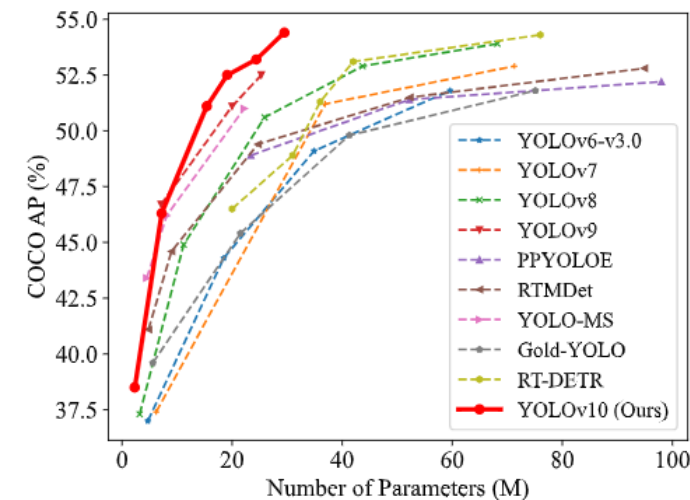
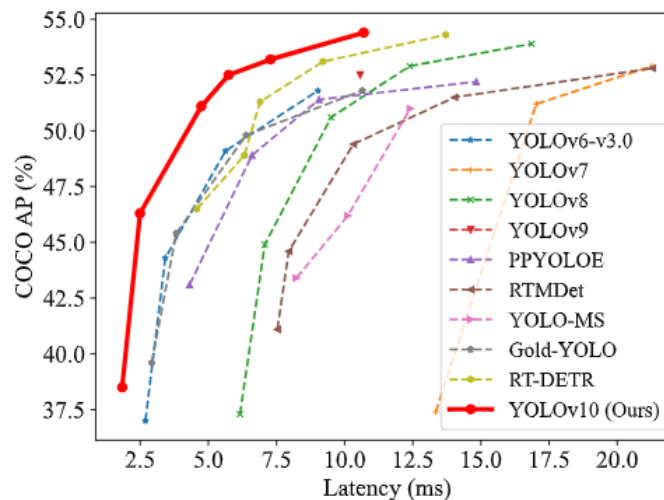
## Supresión no máxima o NMS

Filtro adicional conserva sólo las casillas con la puntuación de probabilidad de detección más alta.




# Ultimo YOLO: v10

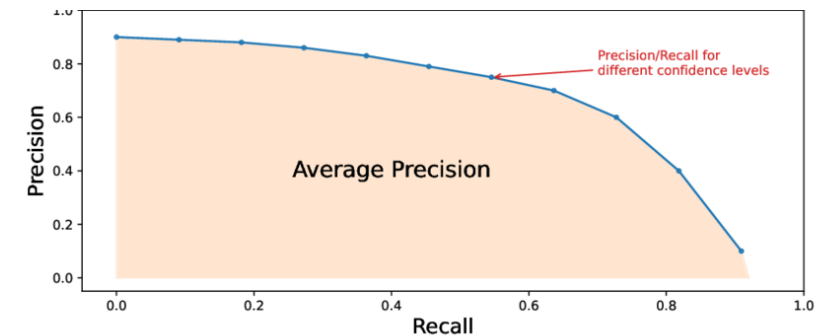
- Más actual: version 10 – 2024
- YOLOv10 incorpora técnicas más sofisticadas para equilibrar la velocidad con una alta precisión de detección.
- Los modelos YOLO originales **eran más rápidos pero menos precisos, especialmente con objetos más pequeños.**
- YOLOv10, por otro lado, está diseñado tanto para la velocidad como para la precisión, integrando innovaciones arquitectónicas modernas como la **autoatención** y los **diseños de bloques adaptativos** que le permiten superar a sus predecesores.



# Métricas de precision de detección de objetos

- **AP** (Average Precision): La precisión promedio (AP) es la métrica principal que se utiliza en la detección de objetos, y **mAP** es el promedio de AP en todas las clases de objetos. Mide tanto la precisión (cuántos de los objetos detectados son correctos) como la recuperación (cuántos de los objetos reales se detectan).
- El mAP se calcula hallando el **área bajo la curva de precisión-recall** para cada clase y luego **promediando estos valores en todas las clases**. Cuanto mayor sea el mAP, mejor será el rendimiento de detección del modelo.
- Precisión y Recall
- F1 Score
- IoU

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$




# Ejercicios - ejemplo

# Ejemplo 1

- Una red neuronal para segmentación fue utilizada para segmentar un tumor sobre tomografías computarizadas del cerebro.
- Las imágenes son de 4x4 y en escalas grises.
- Calcular cuanto se reduce el tamaño en la primera convolución si aplicamos una convolución de 2x2 con paso =1 y sin relleno
- Cuál modelo basado en CNN es mejor utilizar en este caso? Por qué?
- Obtener el IoU y DICE para el siguiente resultado

**Real**

0	0	1	0
0	1	1	0
0	1	1	0
0	1	0	0

**Predicho**

0	0	0	0
0	0	1	0
0	1	1	1
0	0	0	0



# Ejemplo 2

- Queremos detectar perezosos, por lo que creamos un modelo YOLO para esto.
- Obtenemos la predicción para una imagen:
  - **Caja real (GT o ground truth):**  $[x_{GT}, y_{GT}, w_{GT}, h_{GT}] = [100, 150, 200, 100]$  - El cuadro delimitador comienza en (100, 150) y tiene un ancho de 200 y una altura de 100.
  - **Predicción (P):**  $[x_P, y_P, w_P, h_P] = [120, 140, 180, 120]$  - El cuadro delimitador predicho comienza en (120, 140) y tiene un ancho de 180 y una altura de 120.
- Obtener la intersección sobre union (IoU) para este caso

