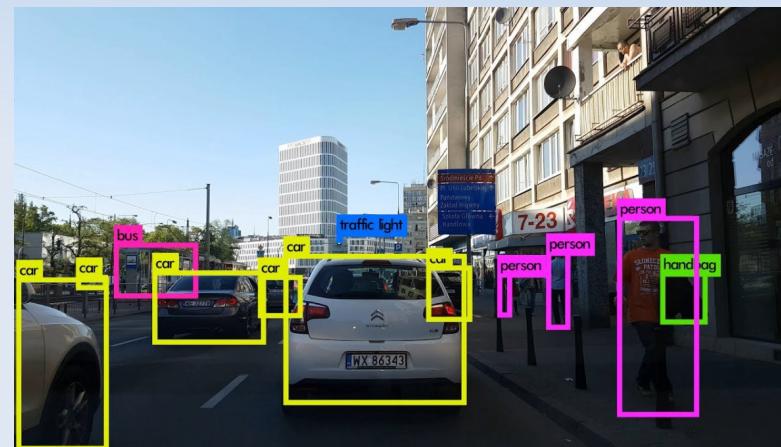


Módulo 2

Procesado de Imagen y Visión por Computador

Grado en Ingeniería en Stmas. Telecomunicación



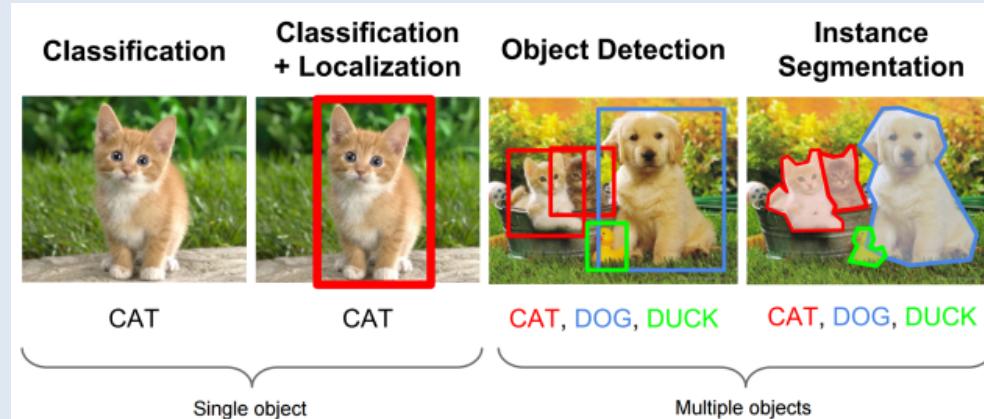
Detección de objetos en tiempo real
(YOLO)



Universidad
de Alcalá

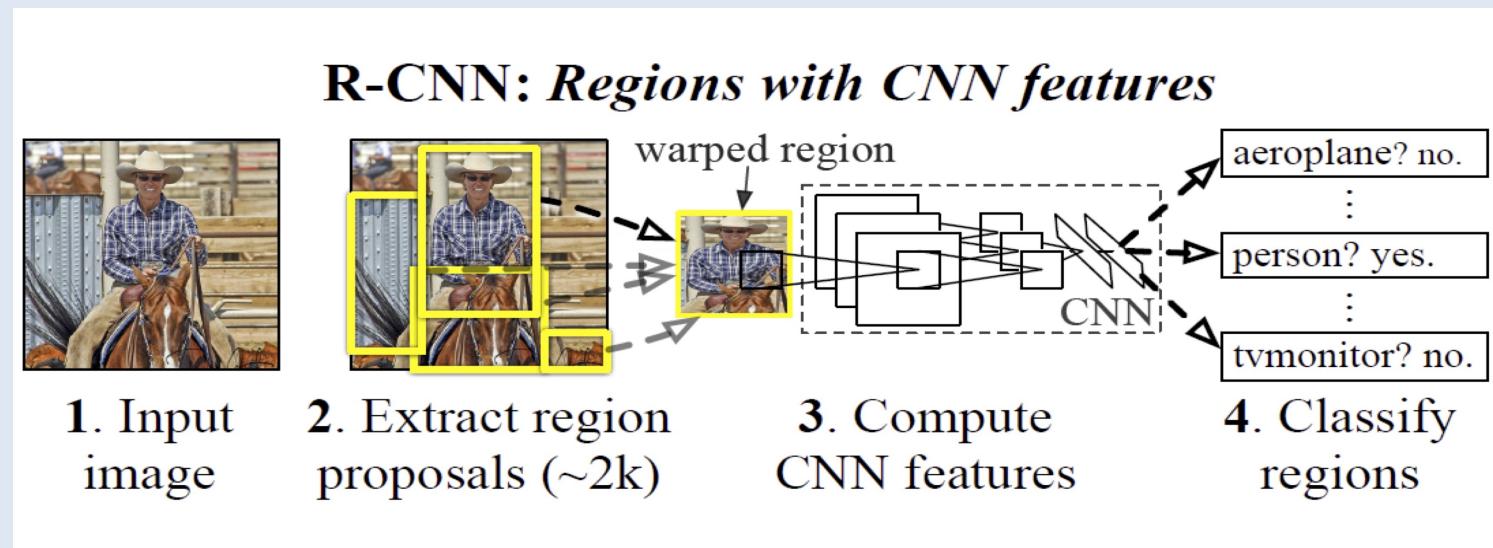
Detección de objetos

- **Clasificación:** asignar una imagen a una categoría en función de su contenido. ¿Qué hay en esta imagen?
- **Localización de objetos:** permite localizar los objetos en la imagen. ¿Qué hay y dónde en la imagen?
- **Detección de objetos:** encontrar todos los objetos en una imagen.
- **Segmentación:** etiquetar todos los píxeles de la imagen en una categoría.



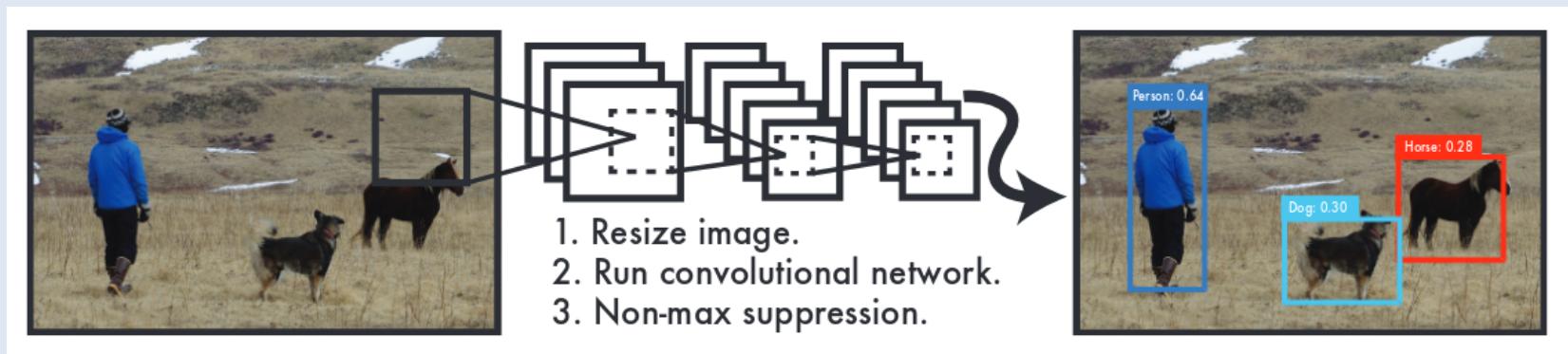
Aproximaciones

- **Algoritmos basados en clasificación:** detectan regiones de interés en la imagen (ROIs) y clasifican cada una de ellas. Ejemplos: RCNN, FastRCNN, ...
 - Inconveniente: predicción para cada ROI (lentitud)



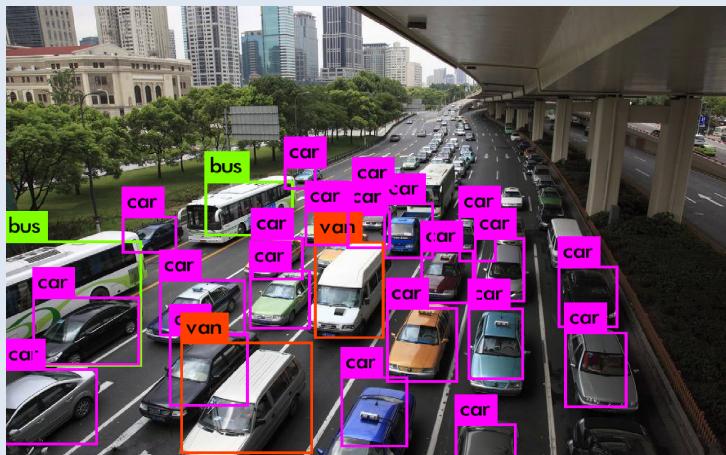
Aproximaciones

- **Algoritmos basados en regresión:** detectan objetos de una sola pasada sin detectar ROIs previamente.
 - Ejemplos: YOLO (You only look once) y SSD (Single Shot Multibox Detector)



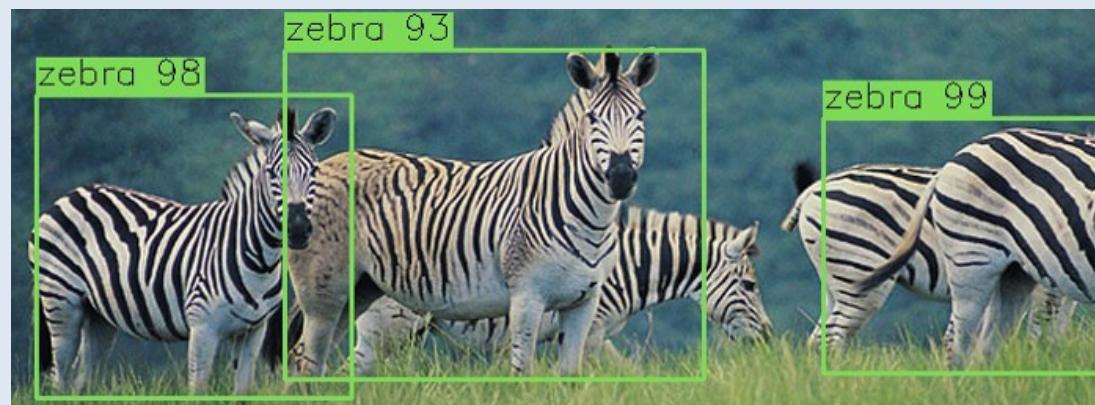
Aplicaciones

- Reconocimiento de caracteres (OCR)
- Vehículos inteligentes
- Robótica
- Vídeo-vigilancia (conteo de objetos)
- Detección facial
- Detección de armas, etc.



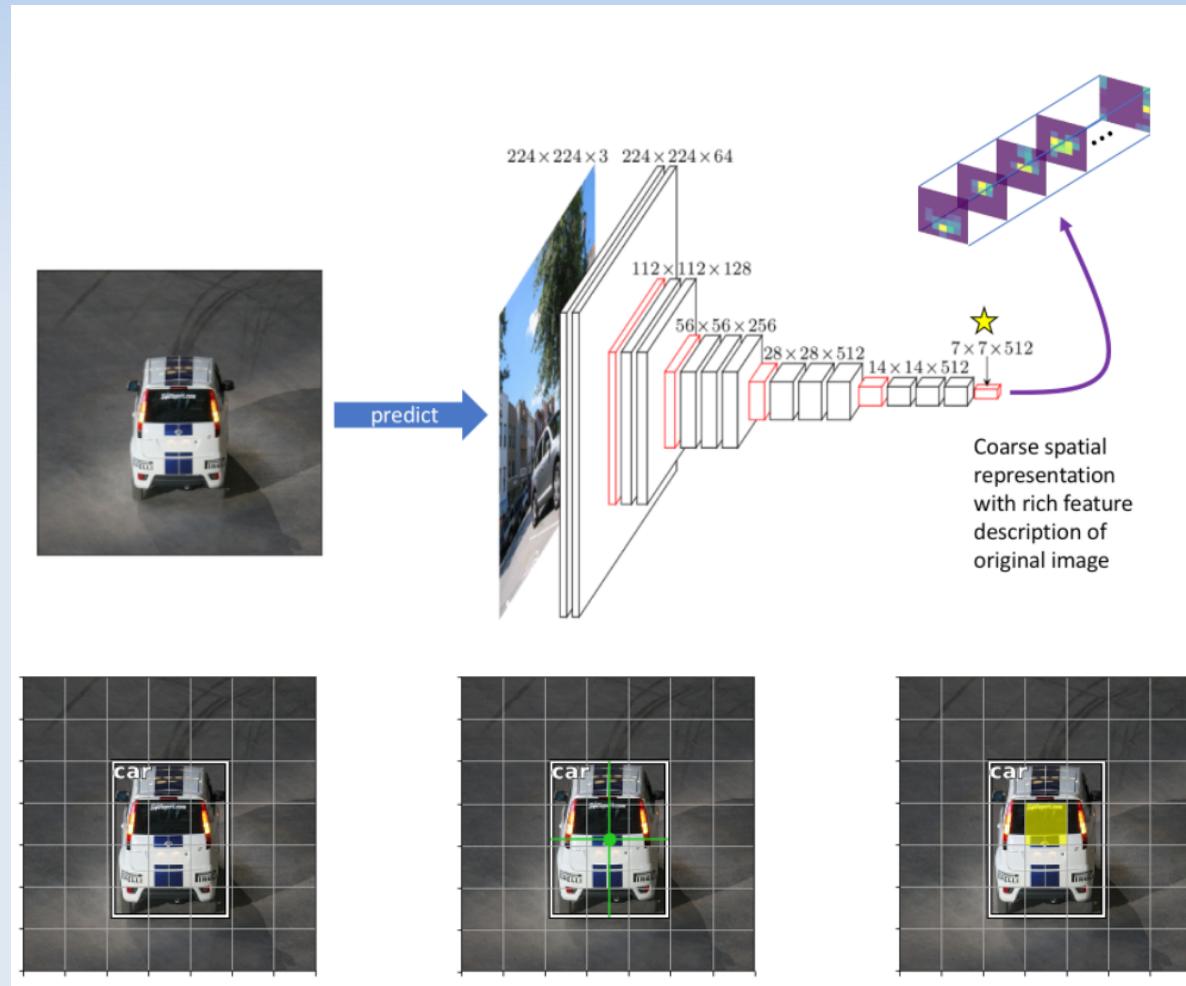
YOLO

- **'You Only Look Once: Unified, Real-Time Object Detection'** by Joseph Redmon, Santosh Divvala, Ross Girshick and Ali Farhadi (2015).
- Entrada: imagen.
- Salida: 'bounding-box' de objetos detectados y predicciones de clases.
- ¿Cómo? Mediante una CNN



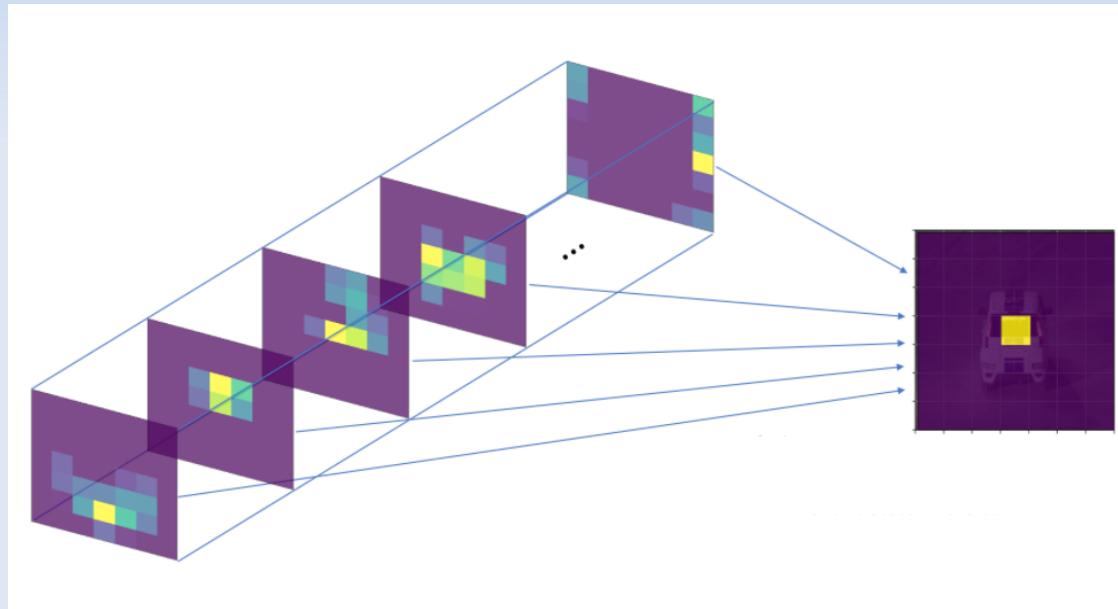
YOLO

- CNN: Mapa de características de la imagen original.



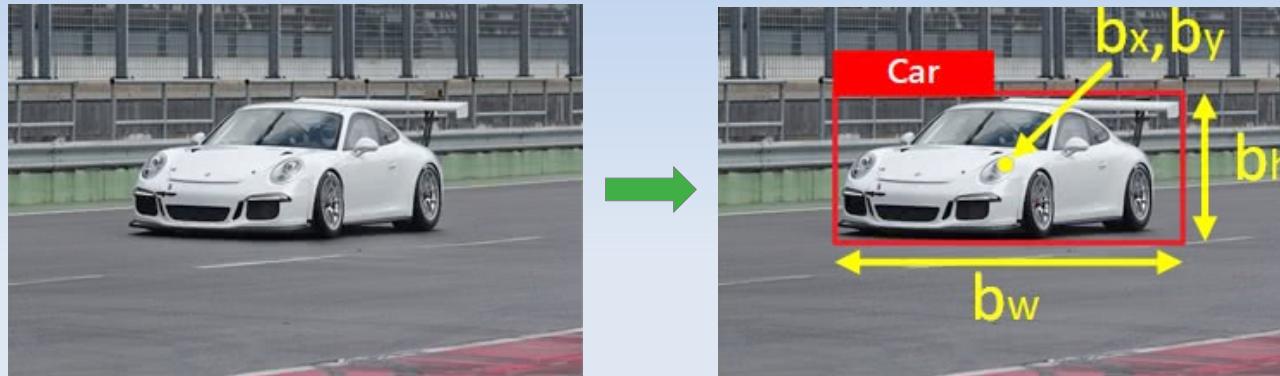
YOLO

- CNN: combina características de salida para activar celda del objeto.



YOLO

- Codificación de salida



$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

p_c : '1' (objeto detectado). '0' (objeto no detectado).
 b_x : coordenada x del centro del objeto.
 b_y : coordenada y del centro del objeto
 b_h : altura del bounding-box
 b_w : anchura del bounding-box
 $C=[0.001, 0.998, 0.001]$ con clases: 1 ('pedestrian'), 2 ('car') y 3 ('motorcycle')

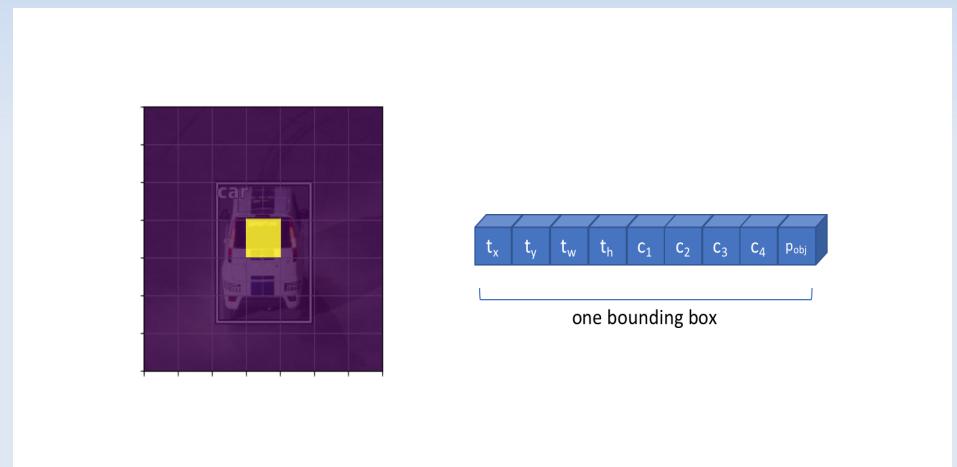
YOLO

- Rejilla de celdas (SxS)



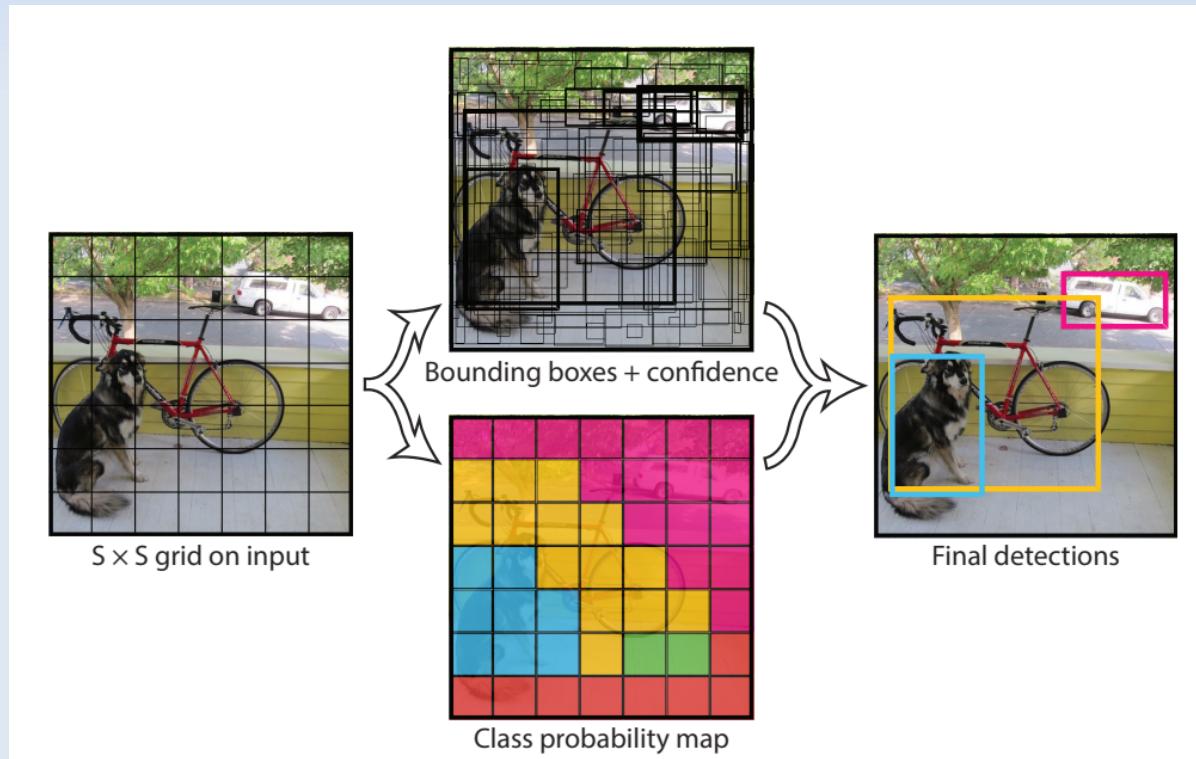
YOLO

- Cada celda es responsable de predecir objetos cuyo centro 'cae' en ella.



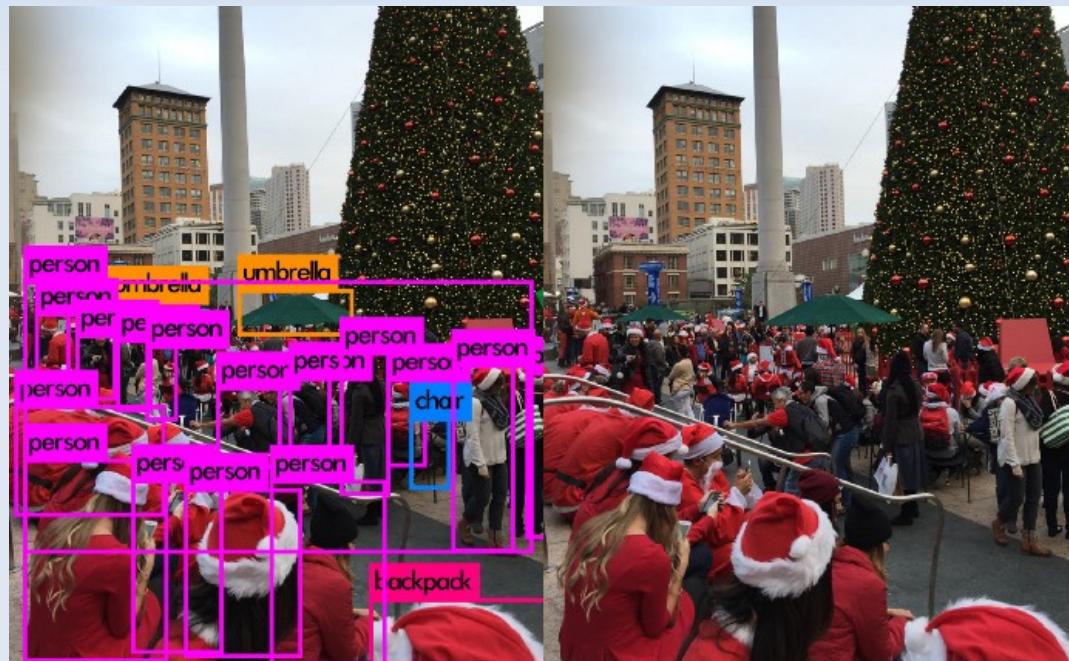
YOLO

- YOLO en detalle:
 - class confidence score=box confidence score x class probability
 - $Pr(Class_i) = Pr(Class_i|Object) * Pr(Object)$



YOLO

- Limitación en la detección de objetos muy próximos:
 - Problemas con grupos de objetos pequeños



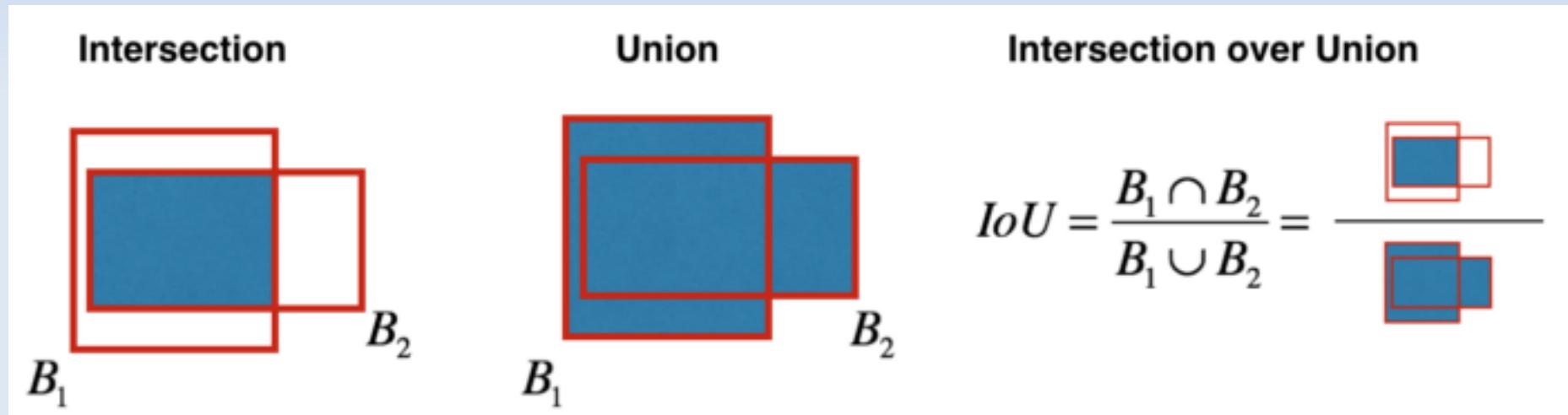
Non-maximal suppression

- Non-maximal suppression: evitar múltiples detecciones de un mismo objeto.



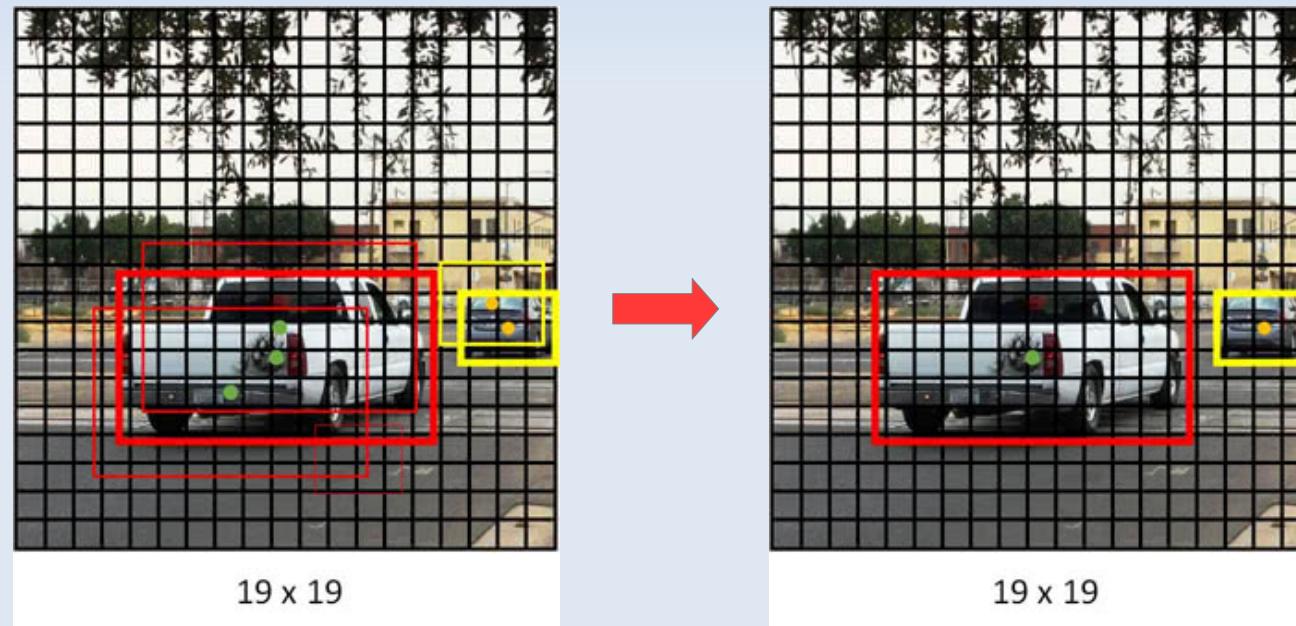
Non-maximal suppression

- Intersección sobre unión (IoU)



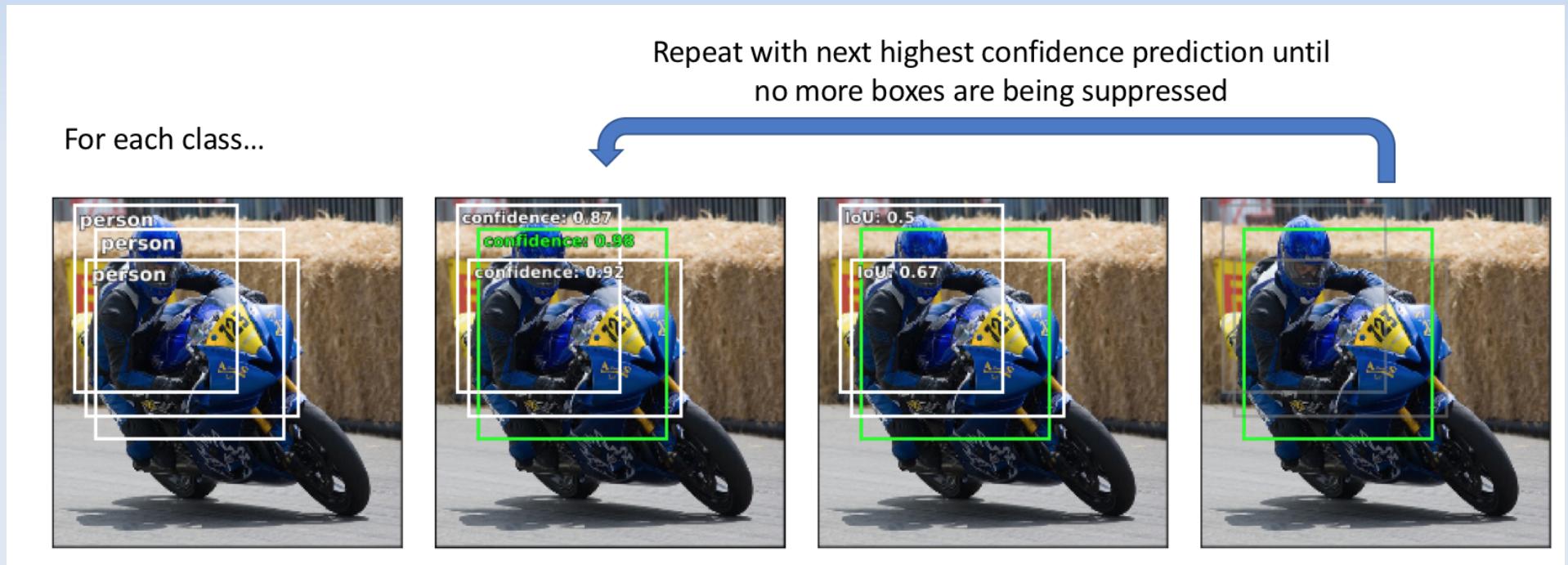
Non-maximal suppression

- ‘Non-max suppression’:
 - Descarta detecciones con ’score’ bajo.
 - Descarta detecciones de la misma clase con IoU por encima de un valor.



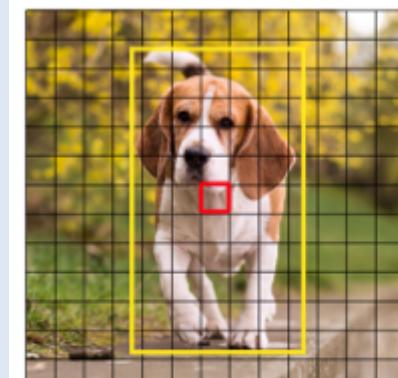
Non-maximal suppression

- ‘Non-max suppression’: se realiza, independientemente, para cada clase.

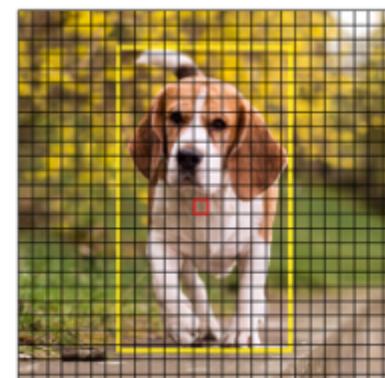


YOLO

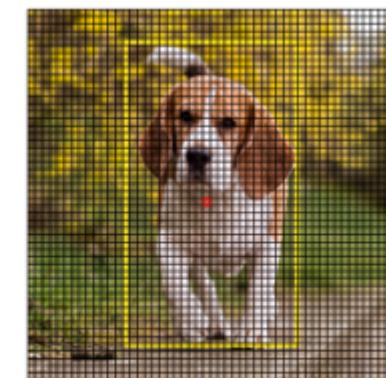
- Versiones:
 - YOLOv1 (2016): una única detección por celda
 - YOLOv2 (2016): introduce mejoras en detección de objetos cercanos y 'anchor box'
 - YOLOv3 (2018): detección a diferentes escalas.
 - YOLOv4 (2020): optimización de velocidad y precisión.



13×13



26×26



52×52

Referencias

- "You only look once: unified, real-time object detection". J. Redmon, S. Divvala, R. Girshick, A. Farhadi. Computer Vision and Pattern Recognition. 2016.
- Enlaces de interés:
 - https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088
 - <https://www.dlogy.com/blog/gentle-guide-on-how-yolo-object-localization-works-with-keras-part-2/>
 - <https://www.jeremyjordan.me/object-detection-one-stage/>