

# Detección de objetos

Métodos Generativos, curso 2025-2026

---

Guillermo Iglesias, guillermo.iglesias@upm.es

Jorge Dueñas Lerín, jorge.duenas.lerin@upm.es

Edgar Talavera Muñoz, e.talavera@upm.es

5 de noviembre de 2025

Escuela Técnica Superior de Ingeniería de Sistemas Informáticos | UPM



# Introducción

Hasta ahora, la única **aplicación** que se ha visto de las **Convolutional Neural Networks (CNNs)** son los clasificadores. Sin embargo existe una variedad enorme de posibilidades a la hora de procesar imágenes.

Dicho esto, las redes que ya han sido estudiadas comparten ciertas características a la hora de **procesar imágenes**:

- Extracción de características.
- Modelos jerárquicos.
- Composición de redes profundas.

Todo esto puede ser **trasladado** a otras aplicaciones de la **visión por computador**.

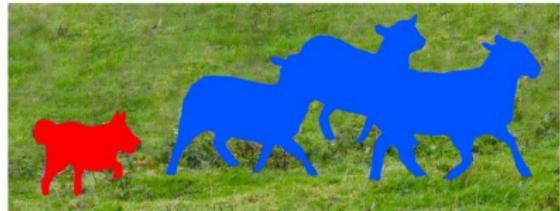
# Introducción

Durante esta sesión se estudiarán las siguientes aplicaciones de la visión por computador:

- Segmentación semántica (semantic segmentation).
- Segmentación de instancias (instance segmentation).
- Detección y localización de objetos (object localization and detection).



Image Recognition



Semantic Segmentation

[1]

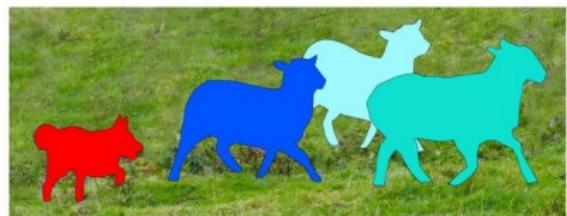
# Introducción

Durante esta sesión se estudiarán las siguientes aplicaciones de la visión por computador:

- Segmentación semántica (semantic segmentation).
- Segmentación de instancias (instance segmentation).
- Detección y localización de objetos (object localization and detection).



Image Recognition



Instance Segmentation

[2]

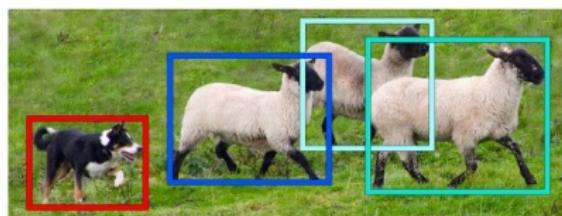
# Introducción

Durante esta sesión se estudiarán las siguientes aplicaciones de la visión por computador:

- Segmentación semántica (semantic segmentation).
- Segmentación de instancias (instance segmentation).
- Detección y localización de objetos (object localization and detection).



Image Recognition



Object Detection

[3]

## Segmentación semántica

---

## Definición del problema

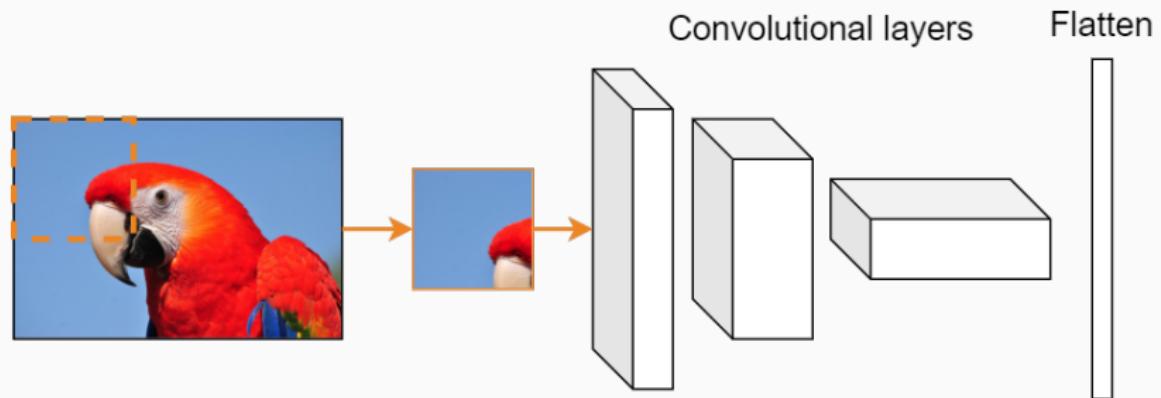
La **segmentación semántica** consiste en realizar una **clasificación** de cada uno de los **píxeles** que conforman una imágen.

De esta manera se le asigna a cada píxel una **etiqueta**, correspondiente al tipo de objeto que está **representando**.

Cada una de las **salidas de la red** representa una **clasificación multi-clase** de los distintos posibles objetos presentes en la imagen. De esta manera cada uno de los **píxeles de salida** se activa con una función **softmax**.

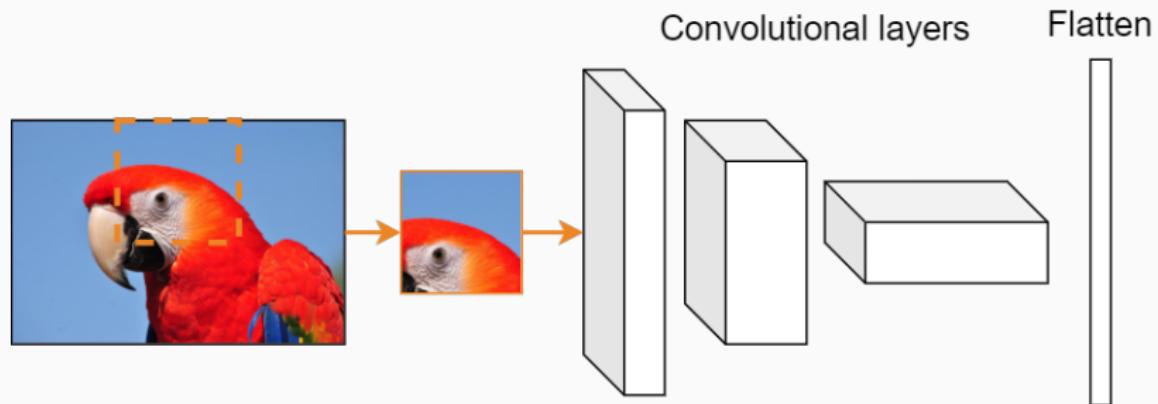
## Ventana deslizante

Uno de los modelos más **sencillos** a la hora de realizar segmentación es utilizar un **clasificador** que reciba cada una de las posibles ventanas de la imagen. Por cada ventana este realizará una **claseficación** del los píxeles contenidos en ella.



## Ventana deslizante

Uno de los modelos más **sencillos** a la hora de realizar segmentación es utilizar un **clasificador** que reciba cada una de las posibles ventanas de la imagen. Por cada ventana este realizará una **claseficación** del los píxeles contenidos en ella.

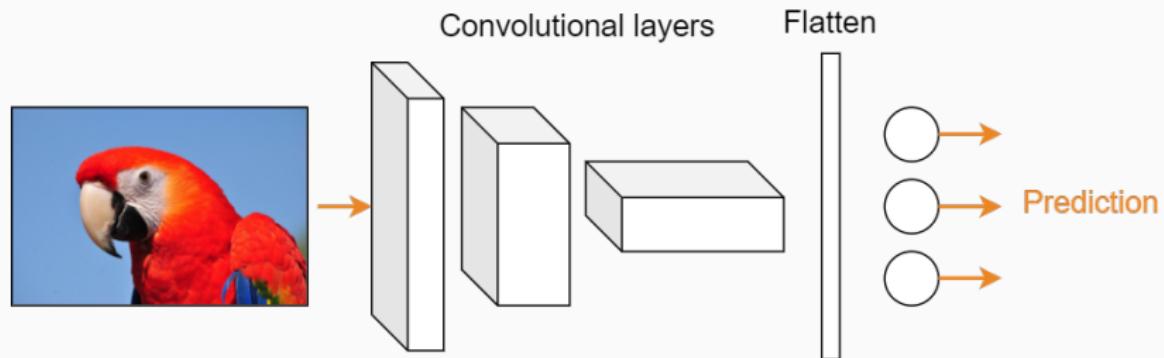


Sin embargo esta aproximación tiene multitud de inconvenientes:

- Ineficiencia computacional.
- Sólo es capaz de analizar una porción pequeña de la imagen.
- Pérdida de información.

# Red completamente convolucional

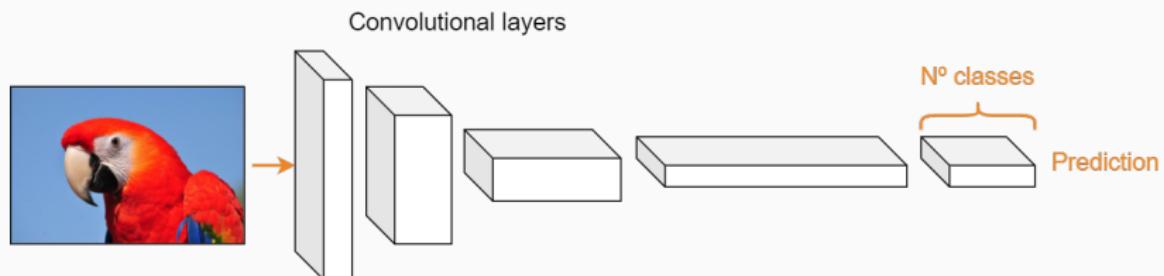
Una **CNN estándar** está compuesta por una serie de capas convolucionales, seguidas de capas de Pooling y finalmente un **perceptrón multicapa** para realizar predicciones.



# Red completamente convolucional

Pero a la hora de realizar **segmentación semántica** no es necesaria la sección del **perceptrón**, ya que no se va a predecir la etiqueta de la imagen.

Una red **completamente convolucional** (fully convolutional CNN)[4] elimina esta parte de la red, usando únicamente capas convolucionales.

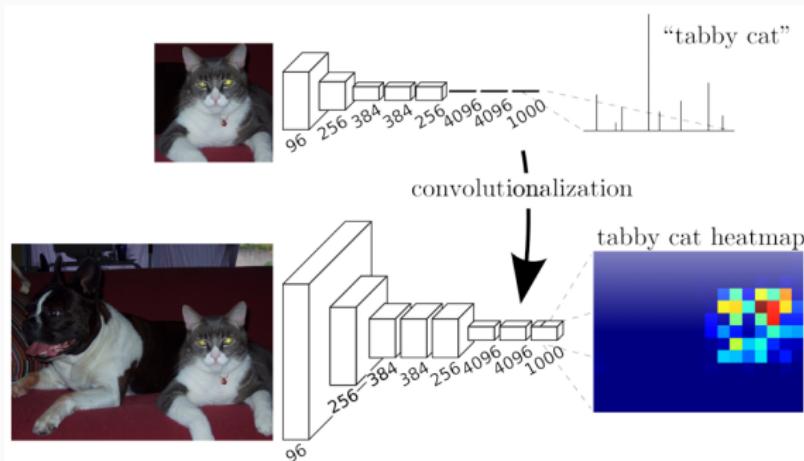


\* *la última capa se encarga de realizar la predicción para cada sección de la imagen, a través de una activación softmax.*

# Red completamente convolucional

Las ventajas de esta arquitectura son:

- Puede tratar **imágenes de cualquier tamaño**.
- **Eficiencia computacional** y de **parámetros**.
- Reutilización de **información**.
- Campo receptivo **grande**.

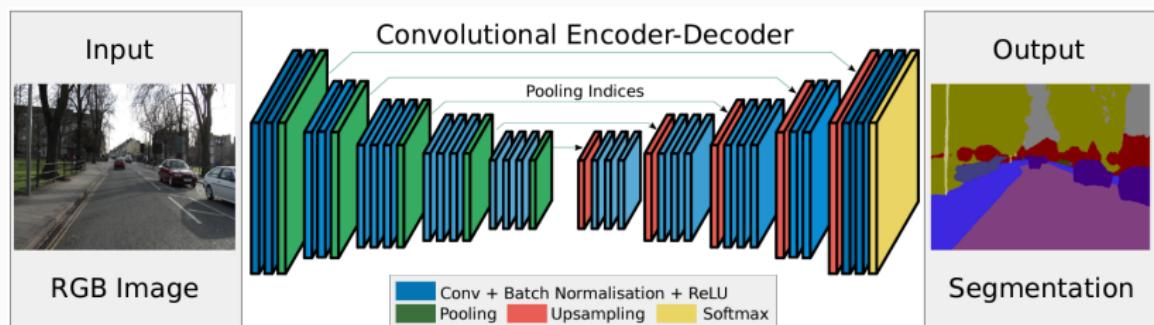


[4]

# Autoencoder

Una mejor solución para realizar segmentación semántica es el uso de **autoencoders**, propuesta con la red **SegNet**[5].

La estructura de **encoder-decoder** permite extraer la información más relevante de la imagen y **reconstruir** la misma **identificando los elementos** que aparecen en ella.



[5]

# UpSampling2D

Para realizar el **cambio de dimensionalidad** ascendente existen distintas alternativas.

## UpSampling2D (MaxUnPooling)

1	2
3	4

1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

# Bed of nails unpooling

Para realizar el **cambio de dimensionalidad** ascendente existen distintas alternativas.

## Bed of nails unpooling

1	2
3	4

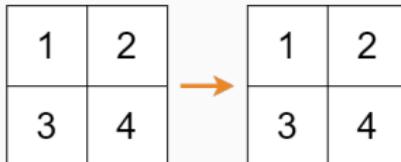
1	0	2	0
0	0	0	0
3	0	4	0
0	0	0	0

# Bed of nails unpooling

Esta arquitectura también propone un nuevo mecanismo para realizar **UpSampling** de la información.

## Max unpooling with memory

1	0	2	1
1	0	1	1
1	3	4	4
2	0	2	1

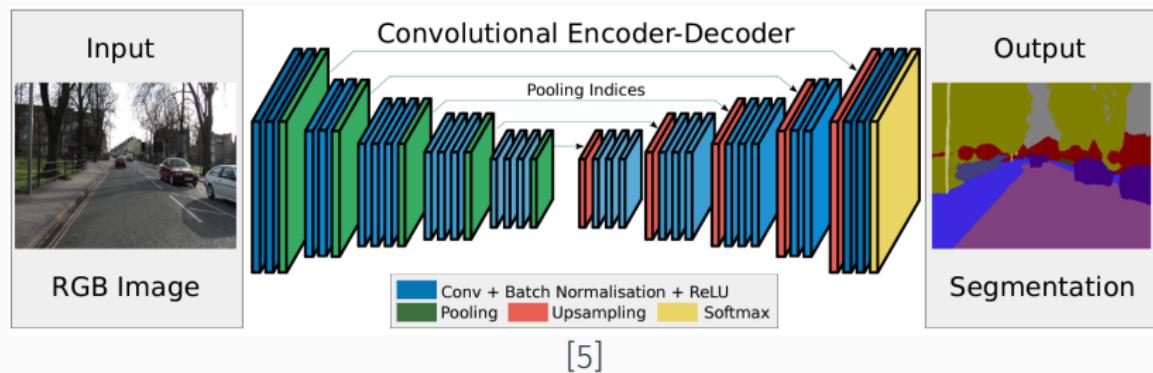


1	0	2	0
1	0	0	0
0	3	4	4
0	0	0	0

# Autoencoder

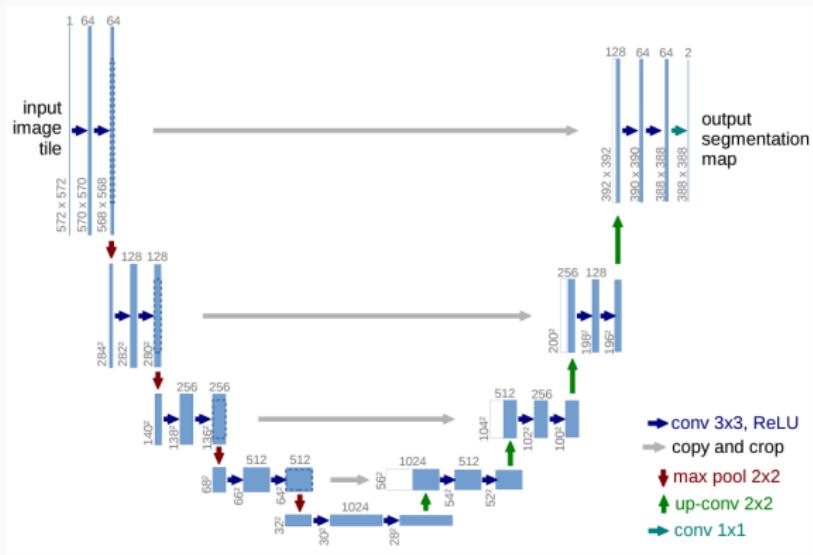
Algunas de las características de esta arquitectura son:

- Uso de **Visual Geometry Group (VGG)** como arquitectura de referencia para el **encoder**.
- MaxPooling **con memoria**.
- **Batch normalization** y **ReLU** como activación.



# U-Net

La arquitectura **U-Net**[6] tiene como objetivo conseguir **imágenes de salida** con una mayor definición. Para ello se definen ciertas **skip connections** con el objetivo de no perder información sobre la composición de la imagen.



[6]

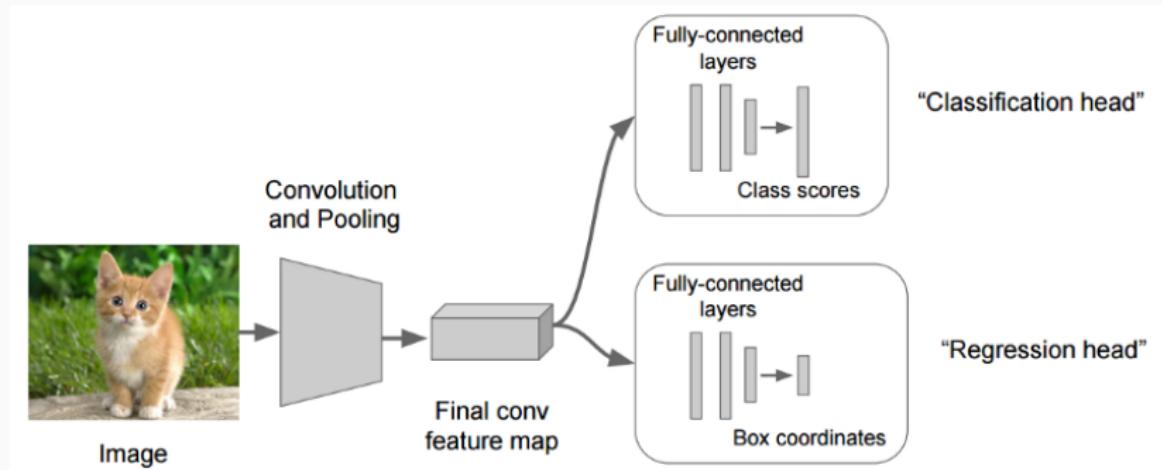
## Localización y detección de objetos

---

# Clasificación y localización

La **solución más simple** para realizar detección de objetos consiste en dividir la salida en **dos tareas distintas**:

- Clasificación de objetos
- Localización de objetos



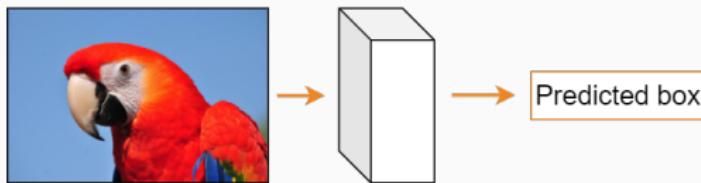
[7]

# Detección de objetos

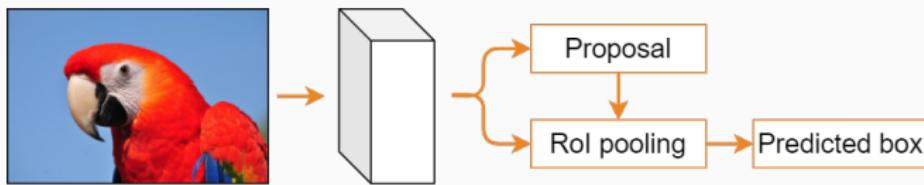
Existen dos grandes ramas a la hora de tratar con la **detección de objetos**:

- Enfoques de una etapa.
- Enfoques de dos etapas.

**One stage approach**

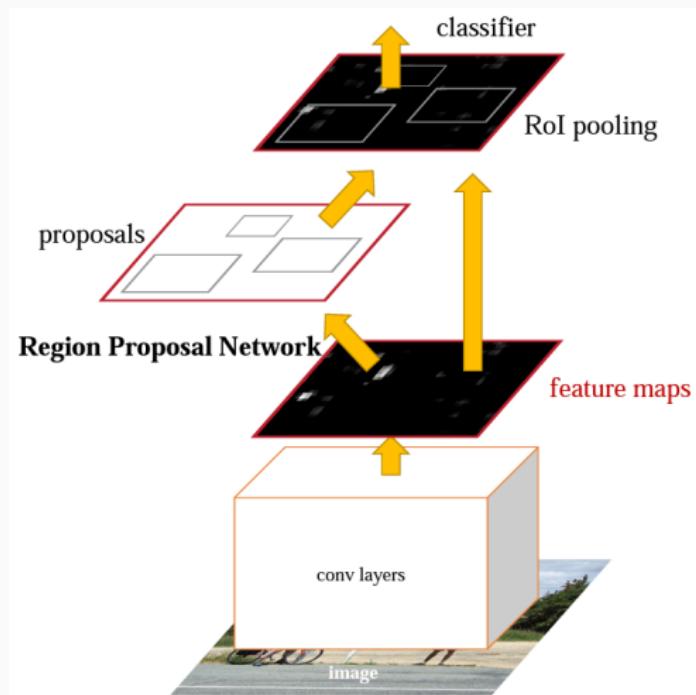


**Two stage approach**



# Faster R-CNN

La arquitectura **Faster R-CNN**[8] es una arquitectura en **2 etapas**.

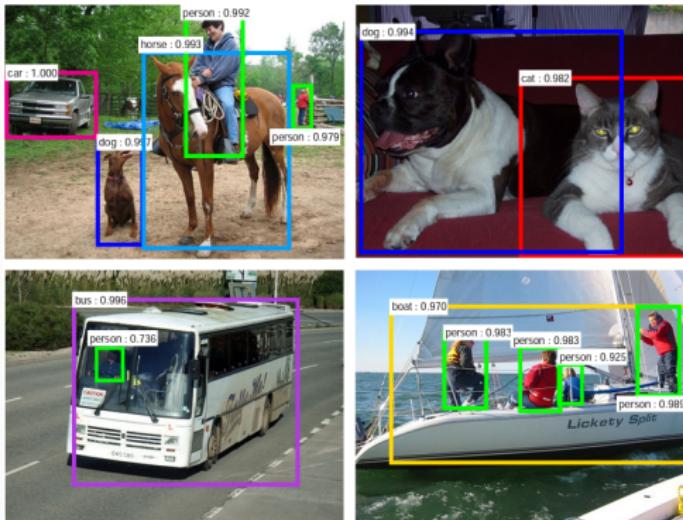


[8]

# Faster R-CNN

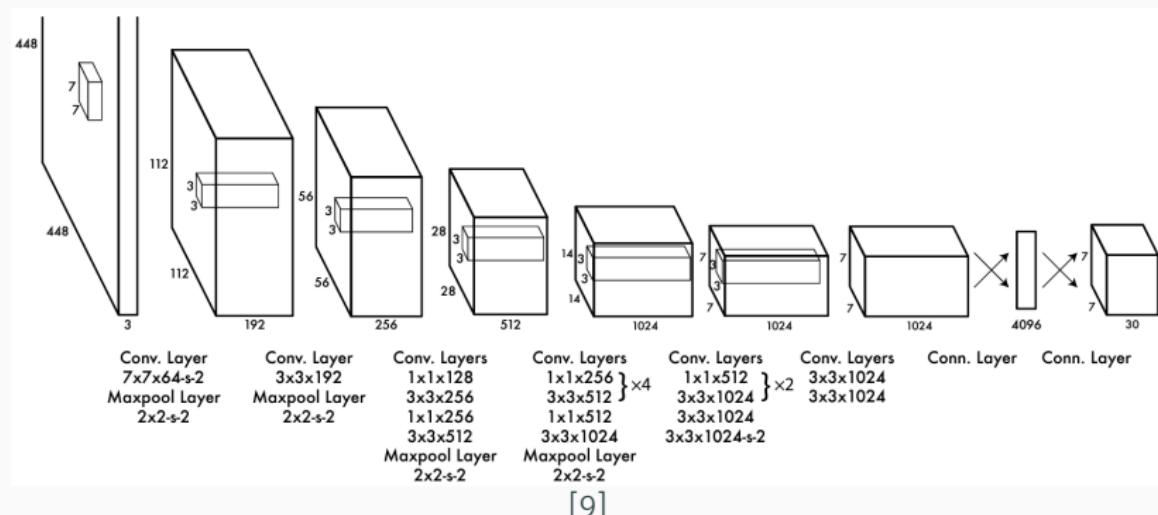
La arquitectura se basa en 2 redes neuronales:

- **Region Proposal Network**: realiza predicciones a través de una ventana deslizante.
- **Feature Pyramid Network**: Se encarga de generar bounding boxes de mayor calidad.



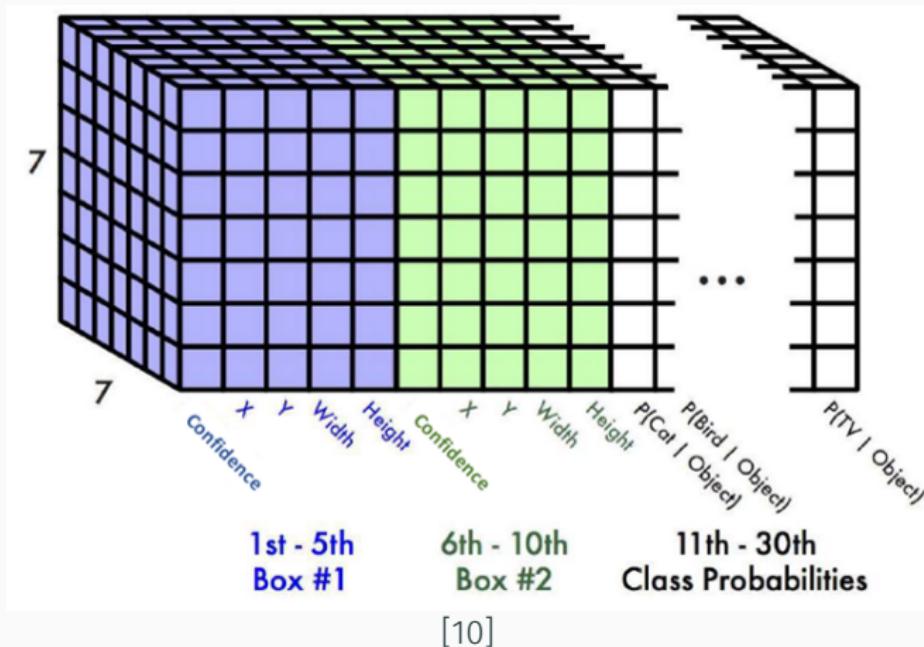
# YOLO

La arquitectura de You Only Look Once (YOLO) está compuesta por 24 capas con estilo de VGG.



# YOLO

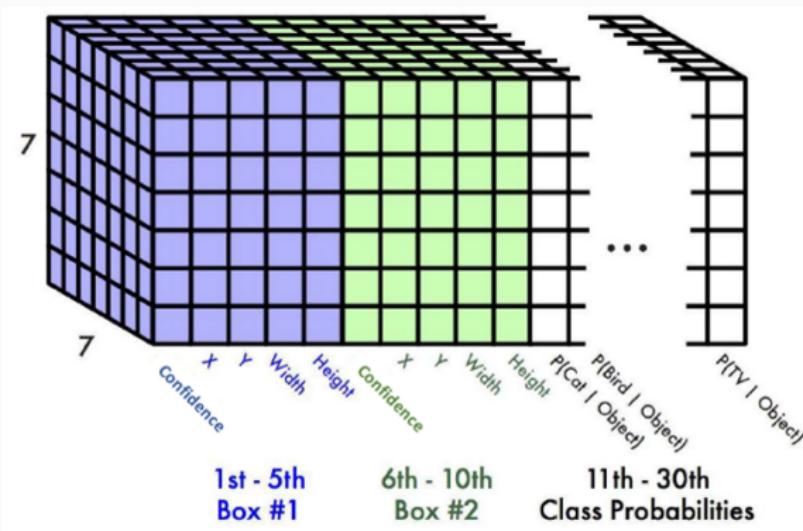
La **última capa** de la red se encarga de **predecir dos bounding boxes** por cada **ventana** de  $7 \times 7$ .



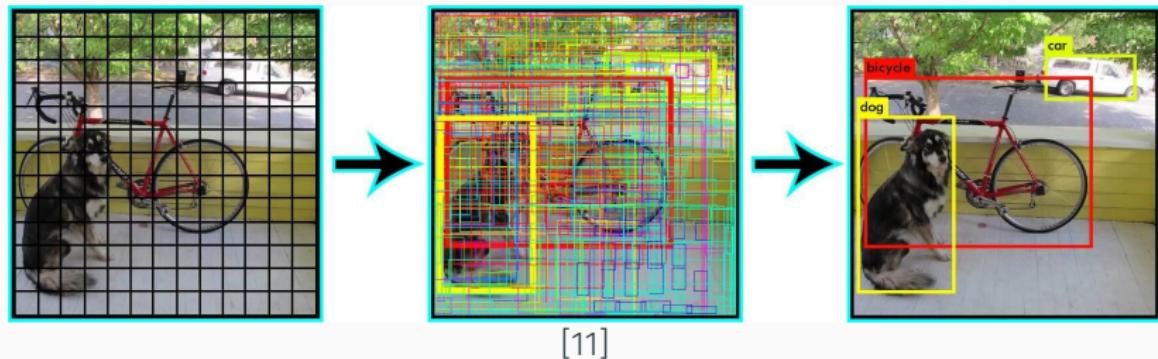
# YOLO

Por cada división se calculan:

- **Dos bounding boxes:**
  - 4 coordenadas de posición ( $x, y, w, h$ ).
  - 1 valor de confianza en la caja.
- **20 probabilidades de clase**



Finalmente se realiza un depurado de las bounding boxes detectadas a través de non-maximum suppression.



## Referencias i

---

- [1] John Wilson (AI Pool).  
**Semantic segmentation image.**  
<https://ai-pool.com/d/could-you-explain-me-how-instance-segmentation-works>.  
[Online; accessed September, 2022].
  
- [2] John Wilson (AI Pool).  
**Instance segmentation image.**  
<https://ai-pool.com/d/could-you-explain-me-how-instance-segmentation-works>.  
[Online; accessed September, 2022].

## Referencias ii

- [3] John Wilson (AI Pool).  
**Object detection image.**  
<https://ai-pool.com/d/could-you-explain-me-how-instance-segmentation-works>.  
[Online; accessed September, 2022].
- [4] Jonathan Long, Evan Shelhamer, and Trevor Darrell.  
**Fully convolutional networks for semantic segmentation.**  
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [5] Vijay Badrinarayanan, Ankur Handa, and Roberto Cipolla.  
**Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling.**  
*arXiv preprint arXiv:1505.07293*, 2015.

## Referencias iii

---

- [6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox.  
**U-net: Convolutional networks for biomedical image segmentation.**  
In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [7] Ravindra Parmar (Towards Data Science).  
**Clasification and localization image.**  
<https://towardsdatascience.com/detection-and-segmentation-through-convnets-47aa42de27ea>.  
[Online; accessed September, 2022].
- [8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun.  
**Faster r-cnn: Towards real-time object detection with region proposal networks.**  
*Advances in neural information processing systems*, 28, 2015.

- [9] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi.  
**You only look once: Unified, real-time object detection.**  
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [10] Sik-Ho Tsang (Towards Data Science).  
**Yolo final layer image.**  
<https://towardsdatascience.com/yolov1-you-only-look-once-object-detection-e1f3ffec8a89>.  
[Online; accessed September, 2022].
- [11] Gilbert Tanner.  
**Yolo non-maximum suppression image.**  
<https://gilberttanner.com/blog/yolo-object-detection-introduction/>.  
[Online; accessed September, 2022].

## Contribuciones de las diapositivas

- Autor original de las diapositivas: Guillermo Iglesias Hernández
- Extensión de contenido: Jorge Dueñas Lerín