

# Detección de objetos I

Visión por computador II

# Contenido

- Introducción
- Aplicaciones
- Retos
- R-CNN
- Faster R-CNN

# Introducción

- Los seres humanos pueden detectar e identificar fácilmente los objetos presentes en una imagen.
- El sistema visual humano es capaz de identificar múltiples objetos y detectar obstáculos sin pensar.
- La detección de objetos puede utilizarse para contar objetos en una escena y rastrear sus ubicaciones precisas, todo ello etiquetándolos con exactitud

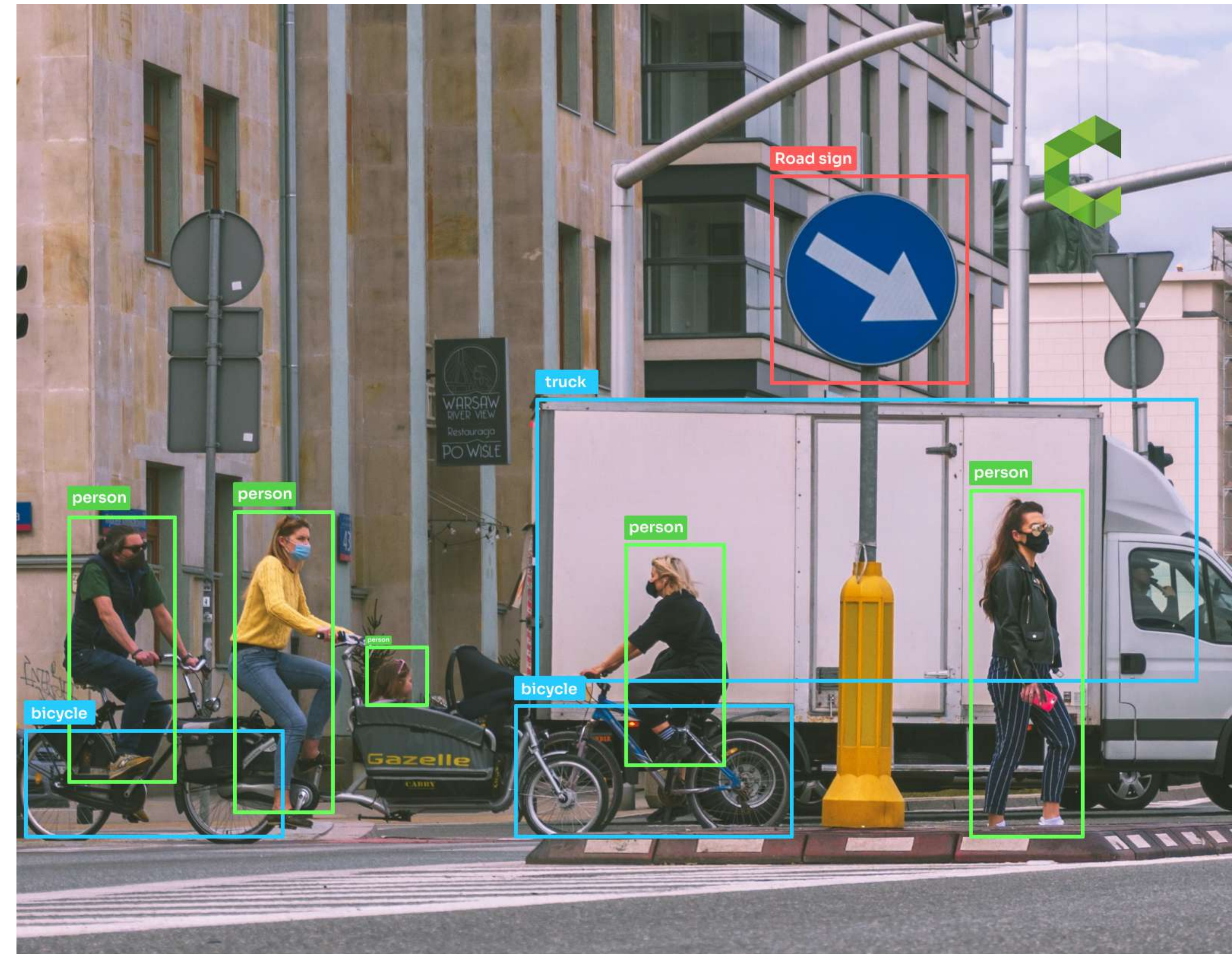


**¿Cuántos gatos hay en la imagen? ¿Dónde se encuentran?**



# ¿Qué es Detección de Objetos?

- La detección de objetos es una importante tarea de visión artificial
- Se utiliza para detectar instancias de objetos visuales de determinadas clases (humanos, animales, coches o edificios)
- El objetivo de la detección de objetos es desarrollar modelos que respondan a la pregunta: **¿Qué objetos están y dónde?**





# ¿Qué es Detección de Objetos?

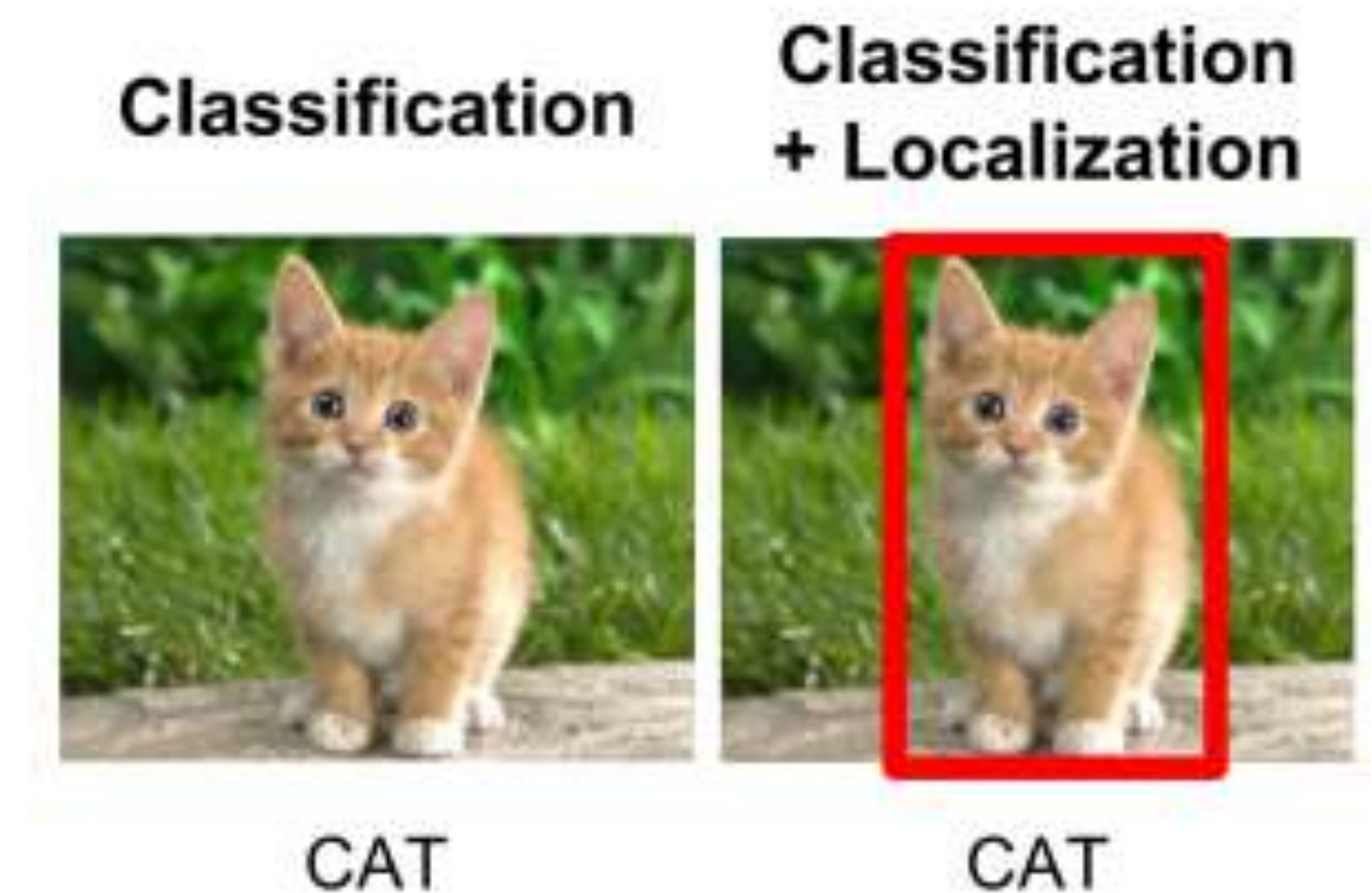
**La clasificación de imágenes** consiste en predecir la clase de un objeto en una imagen.

**La localización de objetos** consiste en identificar la ubicación de uno o varios objetos en una imagen y dibujar un \*recuadro alrededor\*

**La detección de objetos** combina estas dos tareas y localiza y clasifica uno o más objetos en una imagen.

*Dos términos que se usan para el reconocimiento de objetos son:*

- *Object detection*
- *Object recognition*



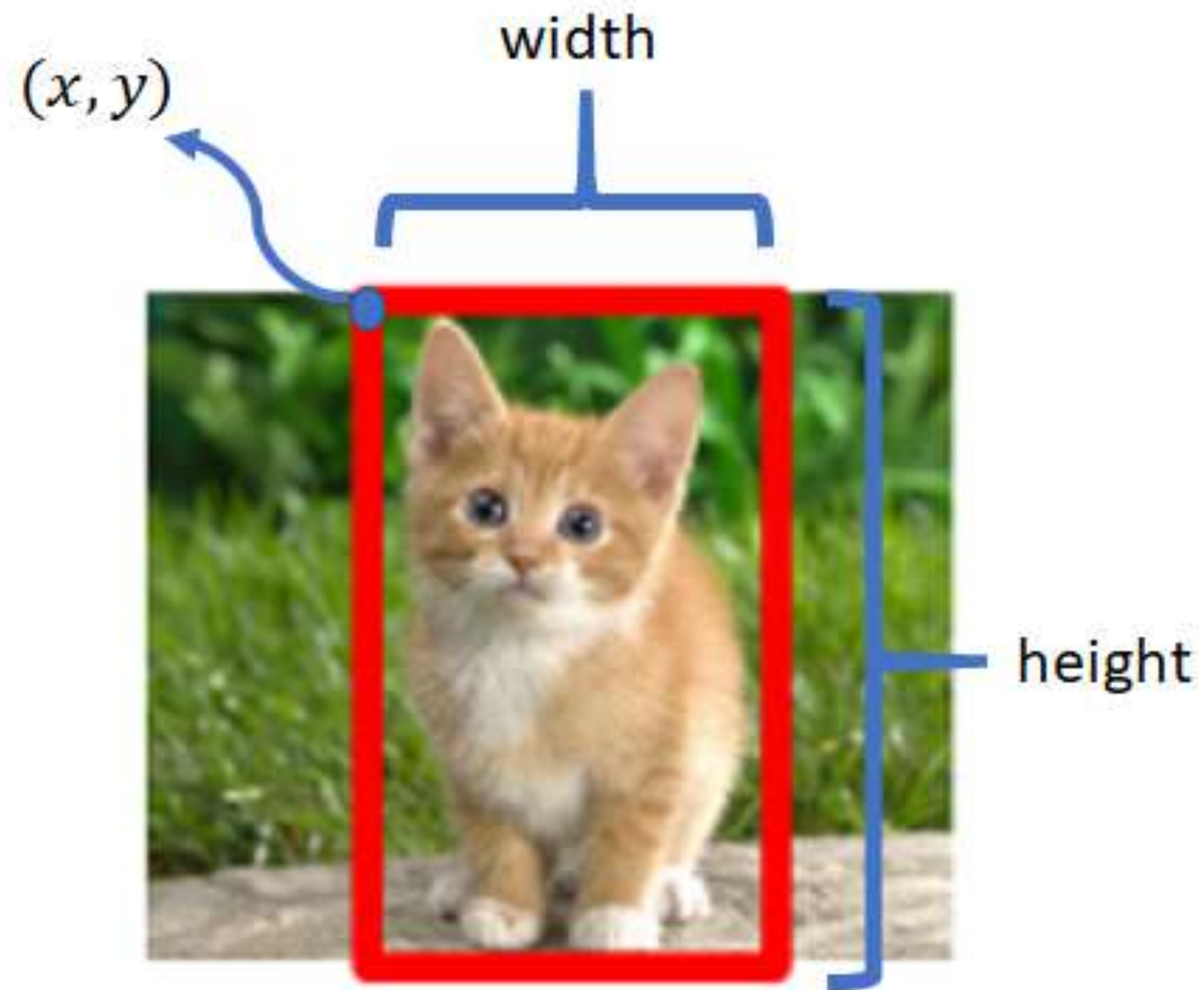
# ¿Qué es Detección de Objetos?



Cuántas variables por objeto debe predecir un modelo de detección?



# ¿Qué es Detección de Objetos?



Localization

+



N clases

# ¿Cómo son entrenados los modelos?

- El proceso de entrenamiento de un modelo para la *detección de objetos* es similar al *clasificación*, con una diferencia
- Los conjuntos de datos de detección de objetos **agrupan una imagen con una lista de objetos que contiene y su ubicación.**
- El modelo acepta una imagen como entrada y devuelve una lista de predicciones por cada objeto con:
  - Ubicación (Coordenadas)
  - Clase

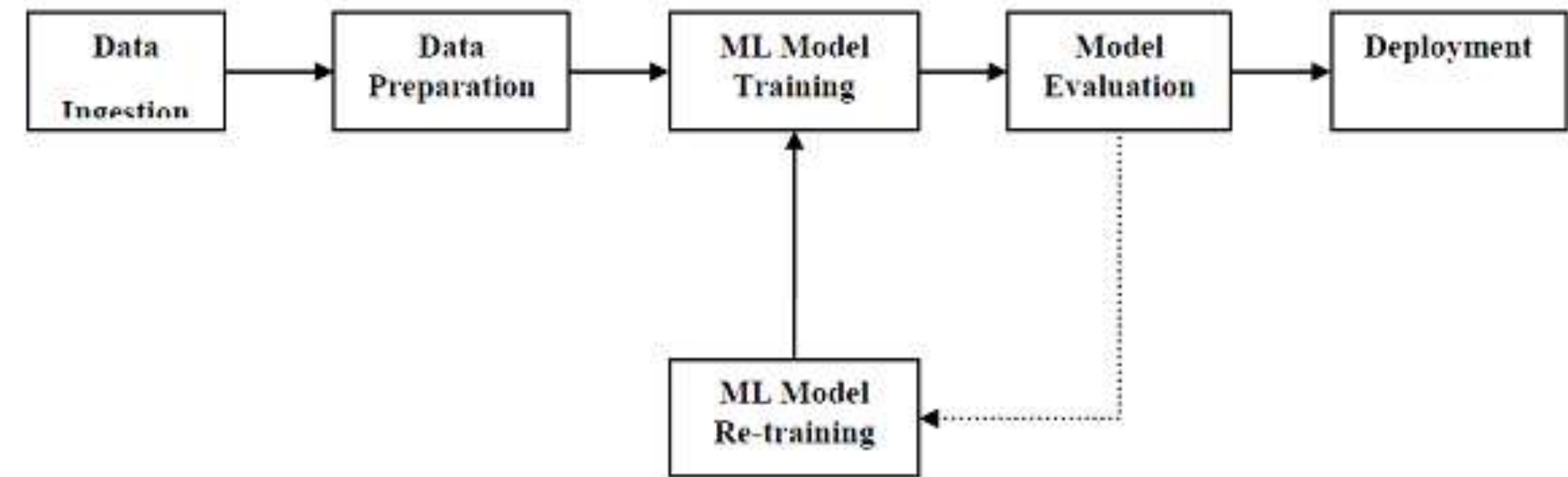
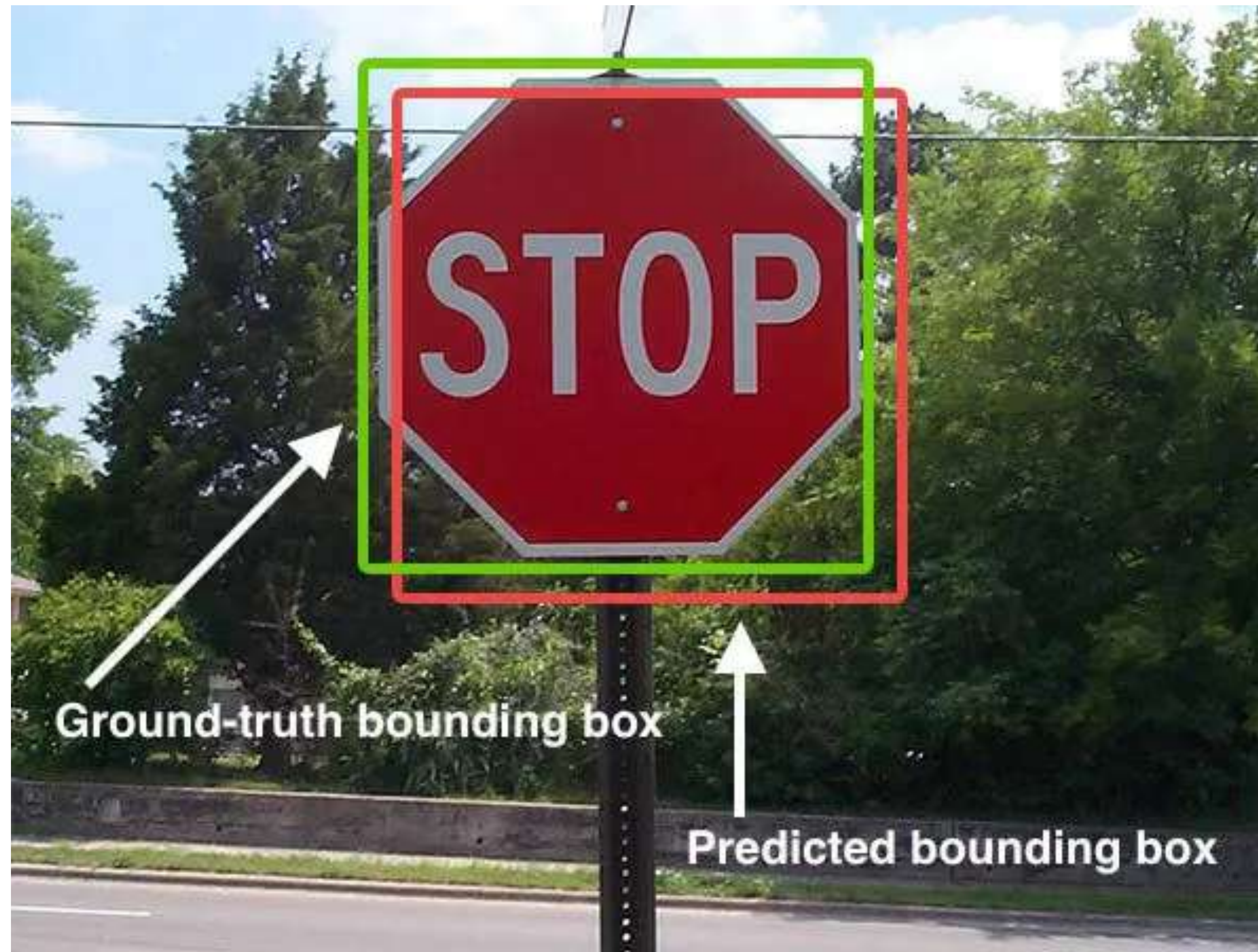


Diagrama de flujo de entrenamiento de modelos de ML



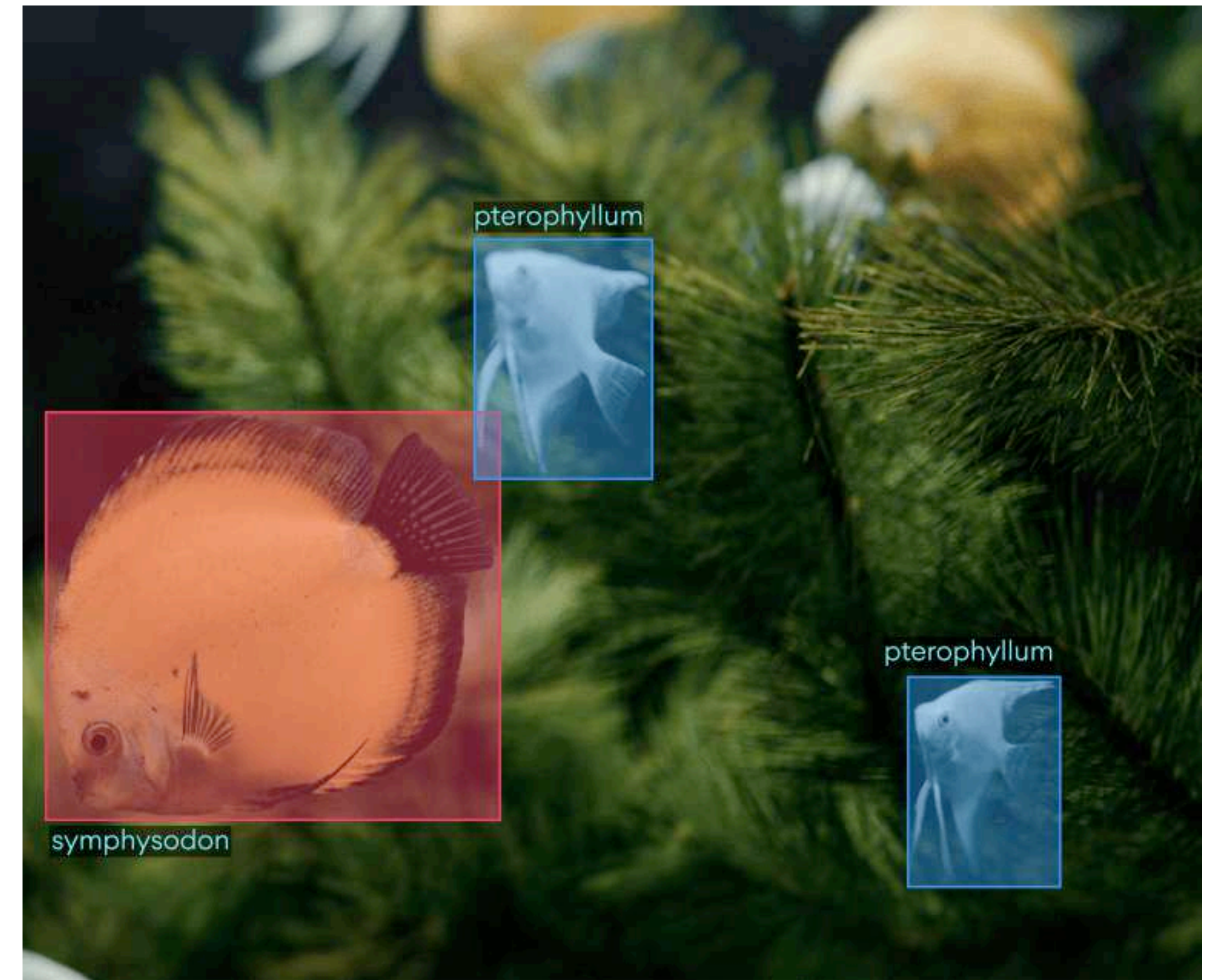


Con los elementos que hemos visto, ¿cómo sería la arquitectura de un modelo de detección de objetos?



# Aplicaciones

- La detección de objetos desempeña un papel importante en la comprensión de imágenes
- Es muy popular en la seguridad, el transporte, la medicina y aplicaciones militares
- Los casos de uso relacionados con la detección de objetos son muy diversos
- Existen formas casi ilimitadas para automatizar tareas manuales a partir de la detección de objetos





# Retail

- Los sistemas de recuento de personas colocados en tiendas se utilizan para recopilar información sobre cómo pasan el tiempo los clientes y su afluencia
- El análisis de clientes basado en IA para detectar y seguir a los clientes con cámaras ayuda a
  - Comprender la interacción y la experiencia del cliente
  - Optimizar la distribución de la tienda y hacer más eficientes las operaciones
- Un caso de uso popular es la detección de colas para reducir el tiempo de espera en las tiendas minoristas



<https://medium.com/@tagxdata/introduction-to-object-detection-for-computer-vision-and-ai-73a955b4d837>



# Conducción autónoma

- Los vehículos autónomos dependen de la detección de objetos para reconocer peatones, señales de tráfico, otros vehículos, etc.
- Por ejemplo, el *autopilot* de *Tesla* utiliza en gran medida la detección de objetos para percibir amenazas del entorno, como vehículos que circulan en sentido contrario u obstáculos.



<https://medium.com/@tagxdata/introduction-to-object-detection-for-computer-vision-and-ai-73a955b4d837>

<https://marketrealist.com/2019/08/tesla-autopilot-troubles-go-way-beyond-crashes/>



# Video vigilancia

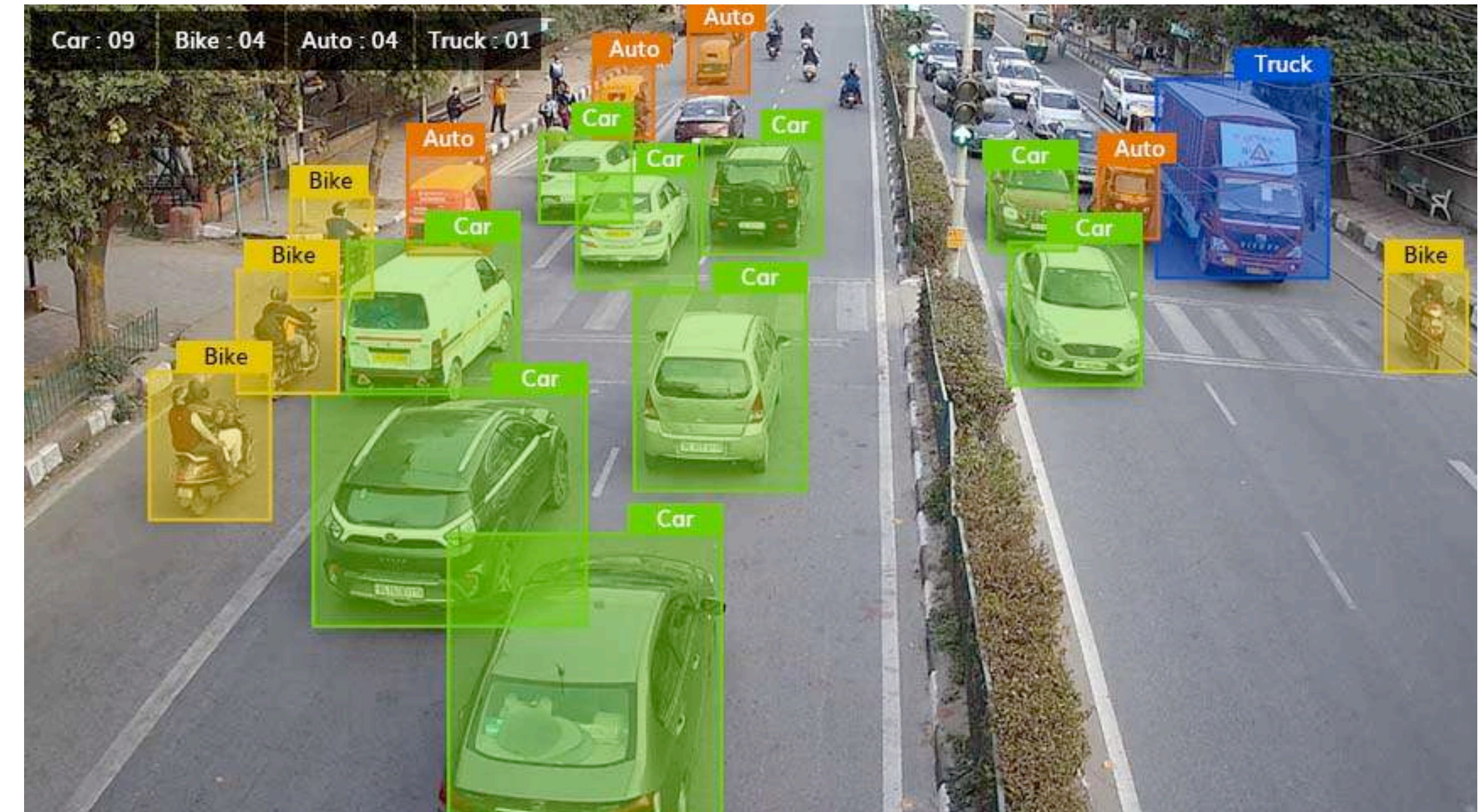
- Los modelos de detección de objetos son capaces de seguir a varias personas a la vez, en tiempo real, mientras se mueven por una escena determinada o a través de fotogramas de vídeo.
- El seguimiento personalizado proporciona información muy valiosa sobre seguridad, rendimiento y seguridad de los trabajadores, tráfico peatonal
- Ejemplo de detección de objetos en análisis de vídeo para la detección de personas en zonas peligrosas mediante cámaras de CCTV





# Monitoreo de tráfico

- Detección de vehículos con IA en el transporte.
- El reconocimiento de objetos se utiliza para:
  - Detectar y contar vehículos
  - Análisis del tráfico
  - Detectar coches que se detienen en zonas peligrosas, por ejemplo, en cruces o autopistas.



<https://medium.com/@tagxdata/introduction-to-object-detection-for-computer-vision-and-ai-73a955b4d837>  
<https://www.pushpak.ai/vehicle-counting-classification>



# Agricultura

Detección de animales en agricultura. La detección de objetos se utiliza en la agricultura para tareas como:

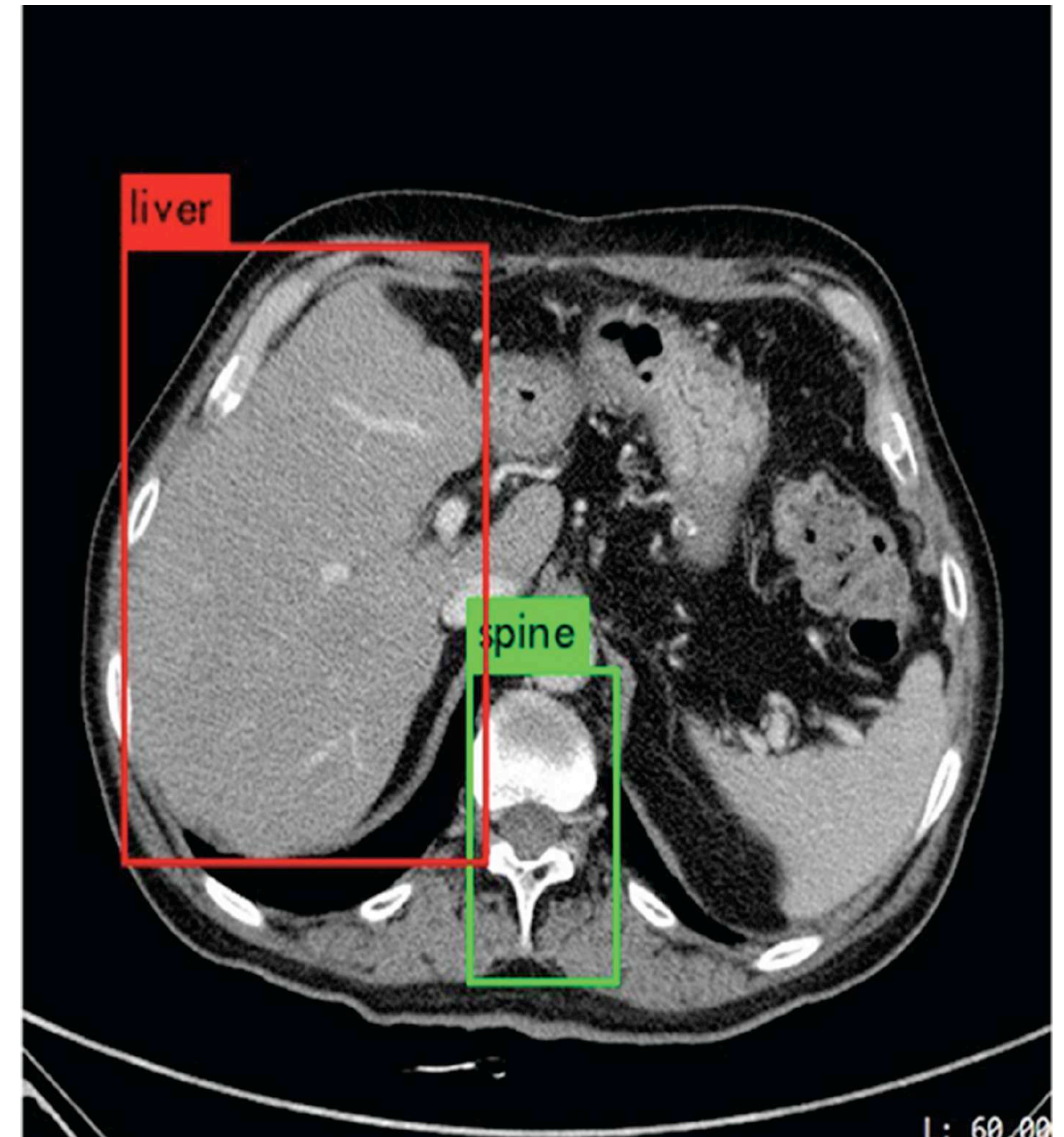
- Conteo
- Monitoreo de animales
- Evaluación de la calidad de los productos agrícolas
- Detección de enfermedades





# Cuidado de la salud

- “Medical object detection” es la tarea de identificar objetos médicos en una imagen.
- Dado que los diagnósticos médicos se basan en gran medida en el estudio de imágenes, escáneres y fotografías,
- La detección de objetos en tomografías y resonancias magnéticas se ha convertido en una herramienta para el diagnóstico de enfermedades.





# Aplicaciones

Otras aplicaciones:

- Detección de peatones
- Recuento de personas
- Detección de caras
- Detección de texto
- Detección de poses
- Reconocimiento de matrículas

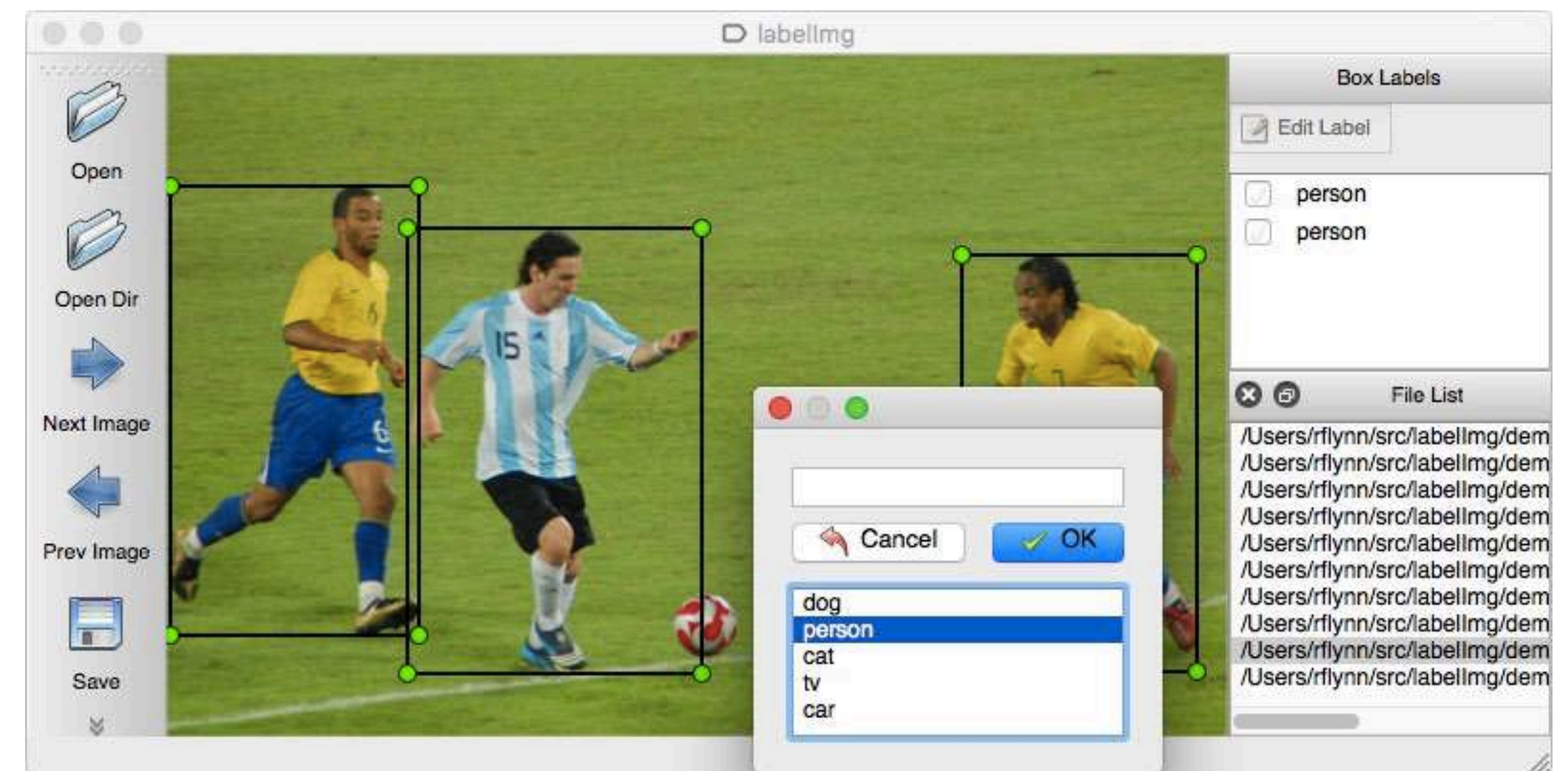




# Acerca de etiquetado de datos

- Los modelos de ML necesitan la mayor parte de enormes cantidad de datos de entrenamiento
- Los anotadores encargan de anotar manualmente cada imagen y generar una gran cantidad de datos etiquetados
- El reconocimiento de objetos es posible gracias a servicios de etiquetado de datos.

*\*\* Por lo general, la mayoría de las empresas de IA no emplean a sus trabajadores ni despliegan recursos para generar los conjuntos de datos*



<https://github.com/heartexlabs/labelImg>



# Retos

## Doble prioridad: clasificación y localización de objetos

- La detección de objetos tiene dos objetivos, la clasificación y la localización.
- Se utiliza una función de pérdida multitarea que penaliza tanto los errores de clasificación como los de localización.

**El término de clasificación** impone una pérdida logarítmica  
**El término de localización** es una pérdida *Smooth L<sub>1</sub>-loss* para los cuatro componentes que definen el rectángulo.

$$\ell(p, u, t^u, v) = \ell_c(pu) + \lambda_{[u \geq 1]} \ell_l(t^u, v)$$

- \* *La penalización por localización no se aplica a la clase de fondo cuando no hay ningún objeto presente,  $u=0$ .*
- \* *El parámetro  $\lambda$  puede ajustarse para dar más peso a la clasificación o a la localización.*



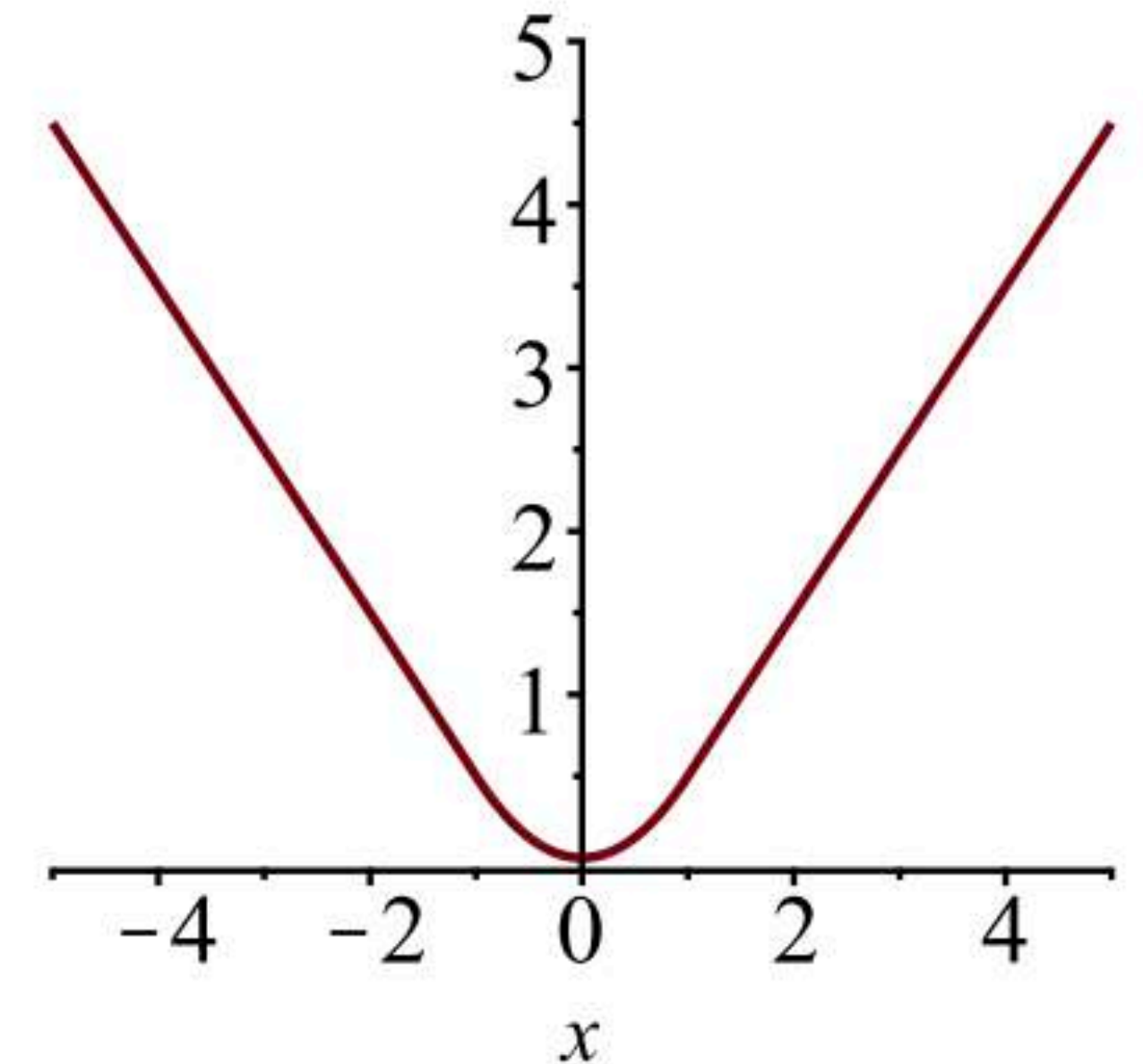
# Smooth L1 loss

$$\text{smooth}_{L1_{\text{plot}}} := \text{piecewise}(\text{abs}(x) < 1, 0.5 \cdot x^2, \text{abs}(x) - 0.5)$$

$$\begin{cases} 0.5 x^2 & |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

→

- Es usada en la regresión de los Bounding box
- Es menos sensible a outliers
- Necesita un LR menor
- En donde x es la distancia entre dos vectores

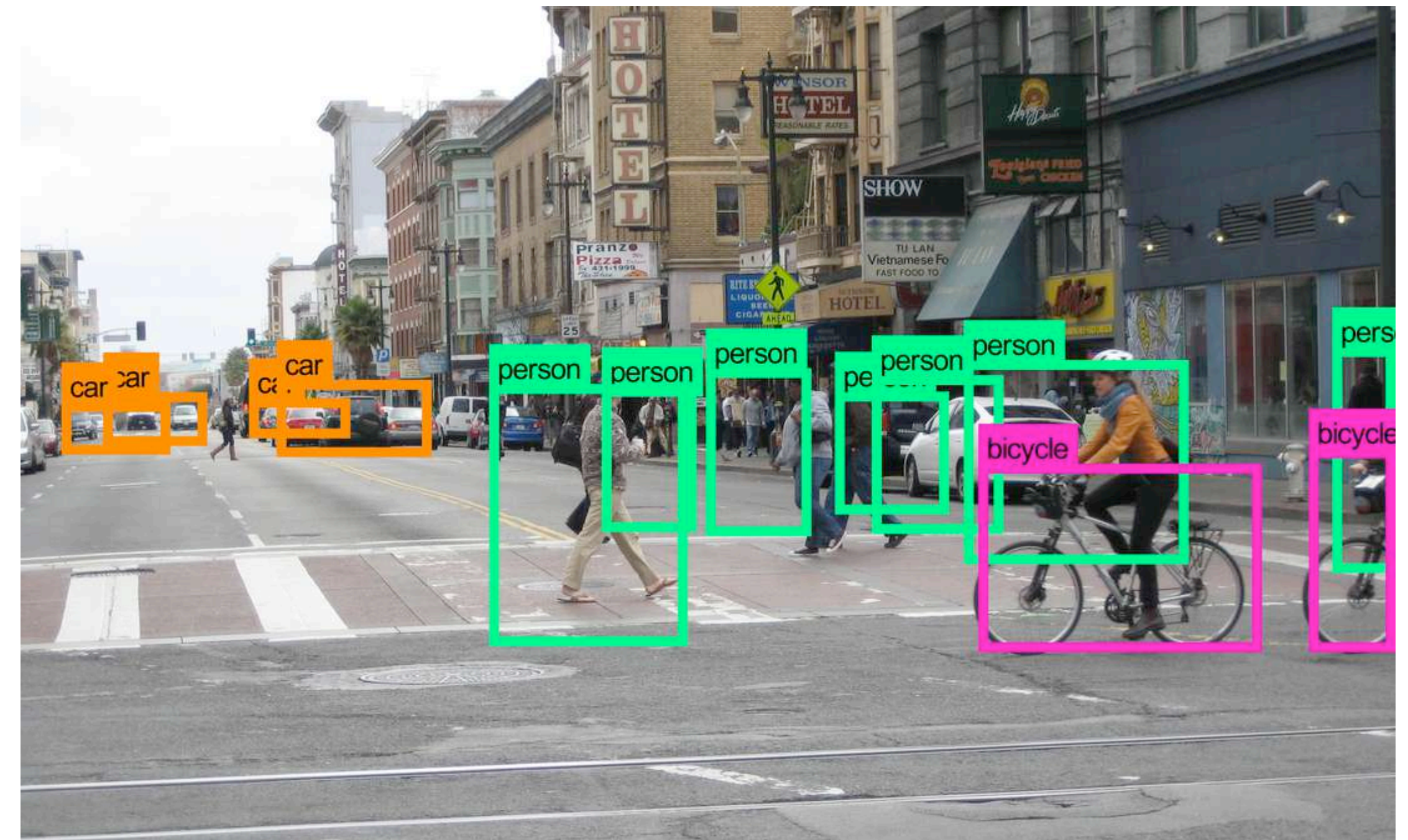




# Retos

## Detección en tiempo real

- Muchas de las aplicaciones vistas requieren análisis en tiempo real.
- Los vídeos suelen grabarse a un mínimo de 24 fps
- los algoritmos actuales de detección de objetos intentan encontrar un equilibrio entre velocidad y precisión



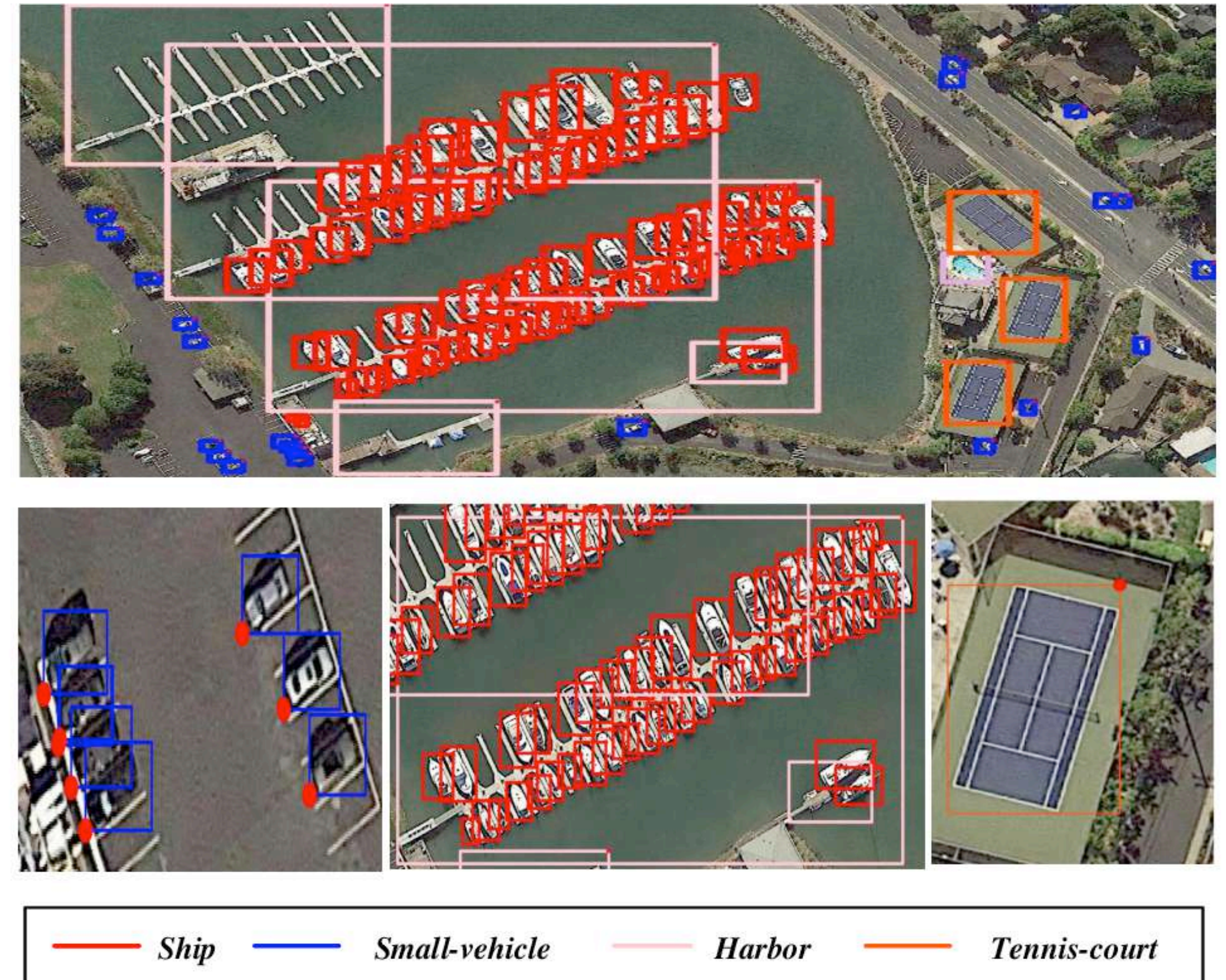
<https://medium.com/analytics-vidhya/object-detection-using-regions-with-cnn-features-557392e22f84>



# Retos

## Múltiples escalas y relaciones de aspecto

- Los elementos de interés pueden aparecer en una amplia gama tamaños y relación entre ancho y alto (*Aspect ratio*)
- Las imágenes con gran diferencia de tamaños son un reto hoy en día





# Retos

## 4. Datos limitados

- La limitada cantidad de datos anotados que se dispone para la detección de objetos es un obstáculo importante
- Aunque el número de clases suele ser menor en los problemas de detección, el etiquetado preciso de los datos es tedioso
- Se requiere una alta cantidad de datos para la correcta localización de los objetos



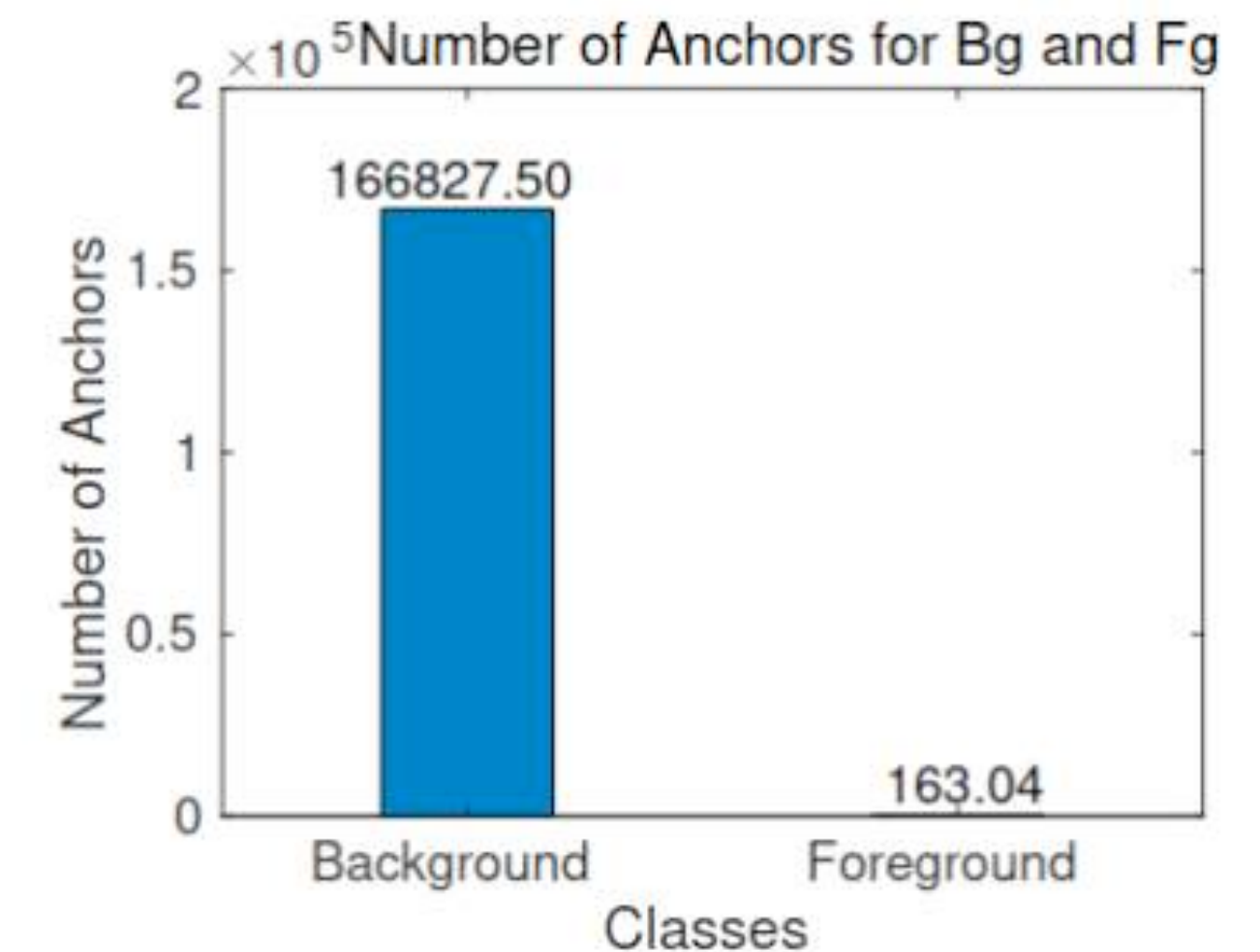
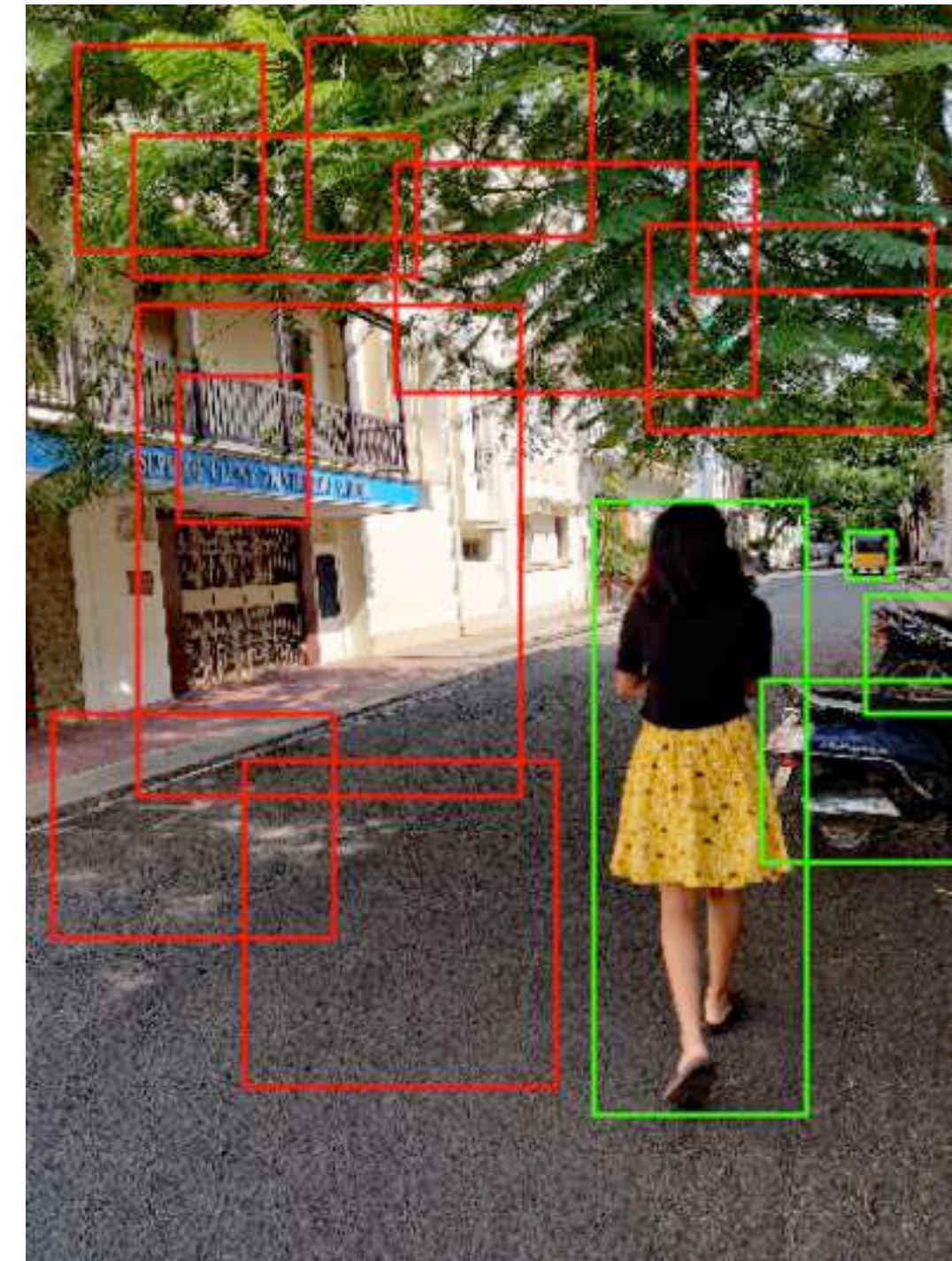
2021 - Object Detection in Densely Packed Scenes via Semi-Supervised Learning with Dual Consistency



# Retos

## 5. Class imbalance

- El desequilibrio de clases resulta ser un problema en la mayoría de los problemas de clasificación y de detección
- En una fotografía típica lo mas probable es que contenga unos pocos objetos principales y que el resto de la imagen esté llena de fondo.





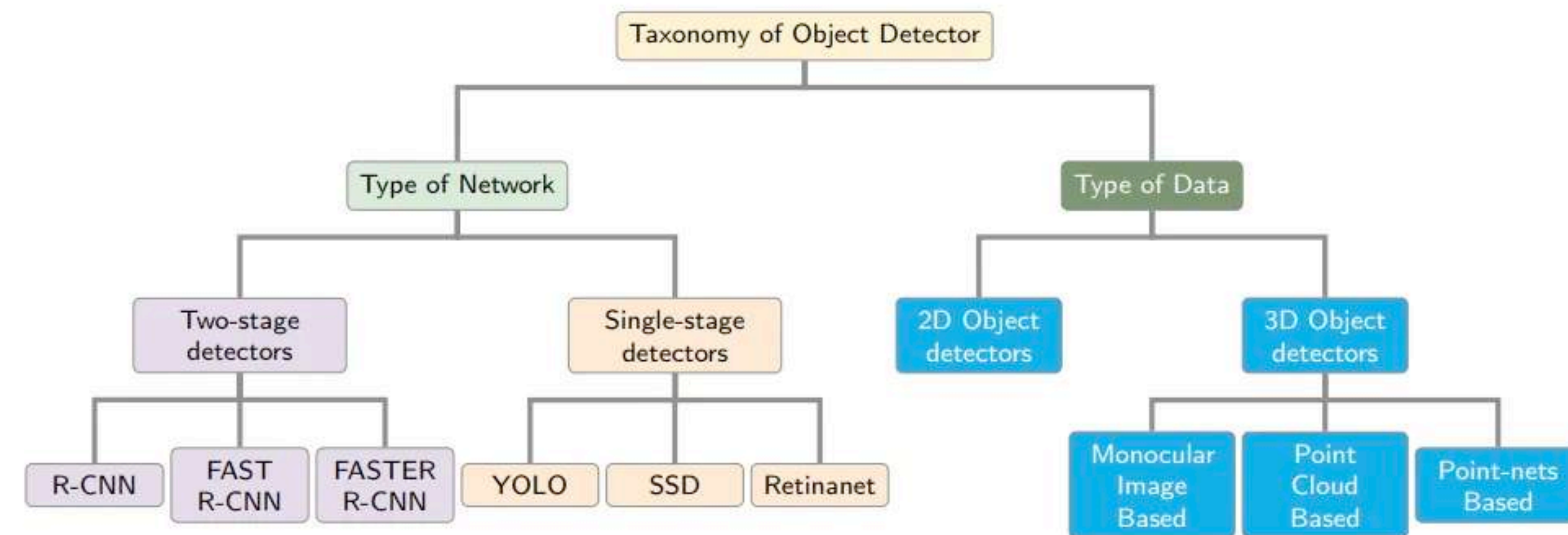
# Algoritmos

- Hay muchos algoritmos para la detección de objetos, cada uno tiene sus pros y contras.
- Tres algoritmos principales que se utilizan en la industria (nivel introductorio)

1. Faster R-CNN

2. SSD

3. YOLO



*2022 - Object Detection in Autonomous Vehicles: Status and Open Challenges*



# Sliding window

- **Una solución convencional** es usar una ventana deslizante para buscar cada posición dentro de la imagen
- Diferentes objetos o incluso el mismo tipo pueden tener diferentes relaciones de aspecto y tamaños dependiendo del tamaño del objeto y la distancia desde la cámara.
- Este proceso es extremadamente lento si utilizamos CNN para la clasificación de imágenes en cada ubicación.

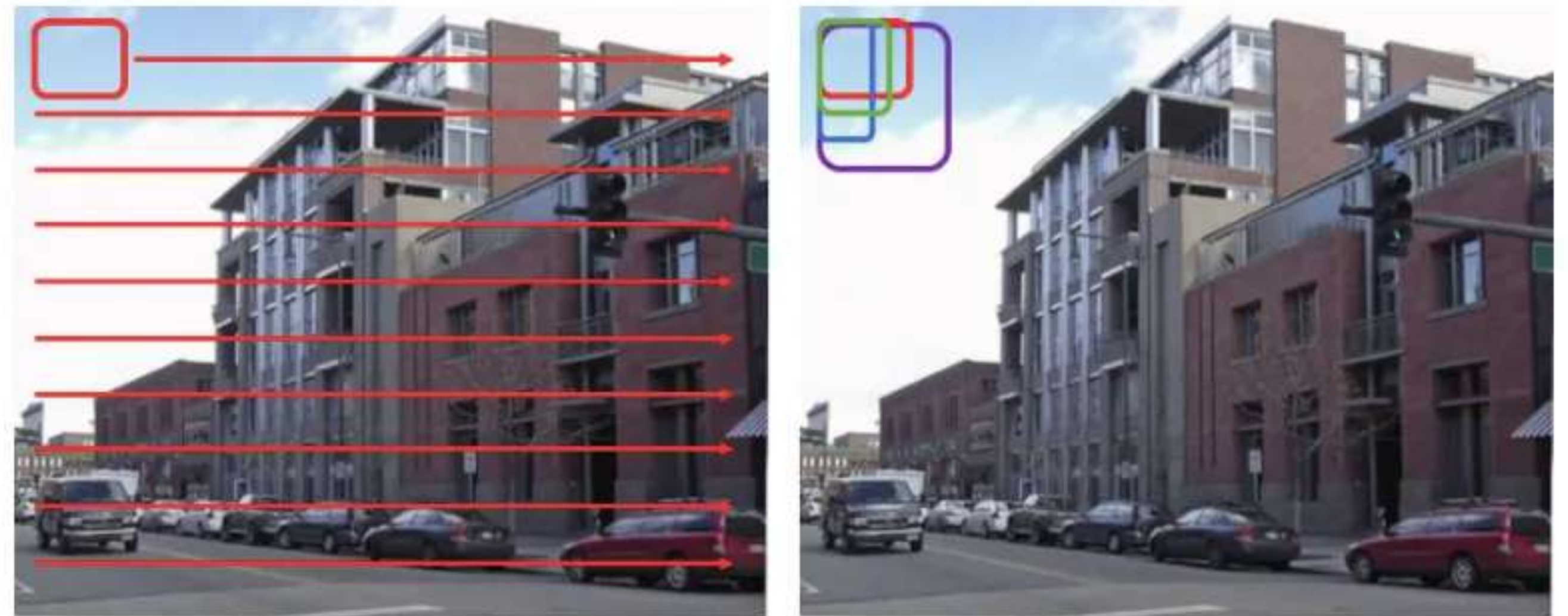


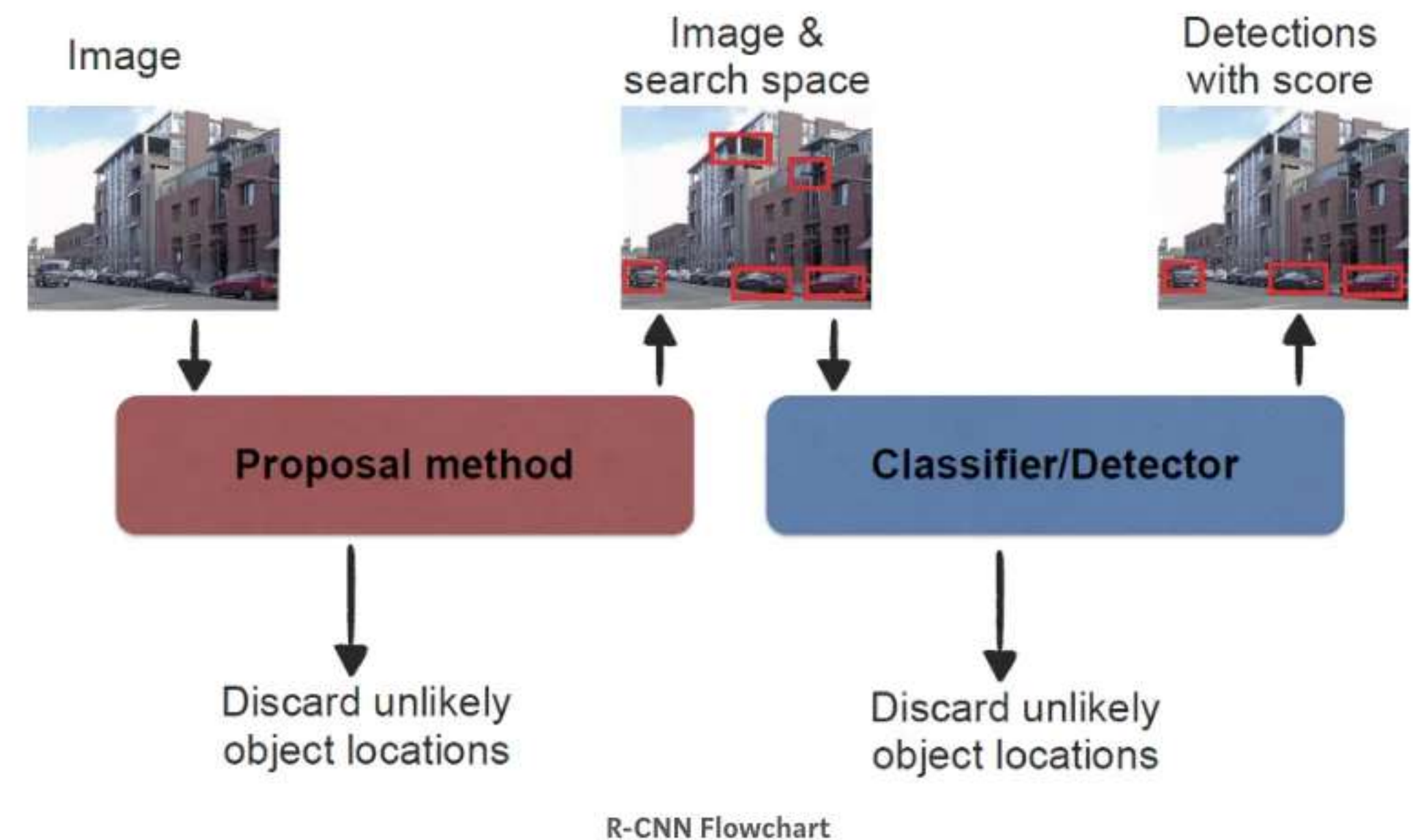
Illustration of Sliding Window (Left) with Different Aspect Ratios and Sizes (Right)



# R-CNN

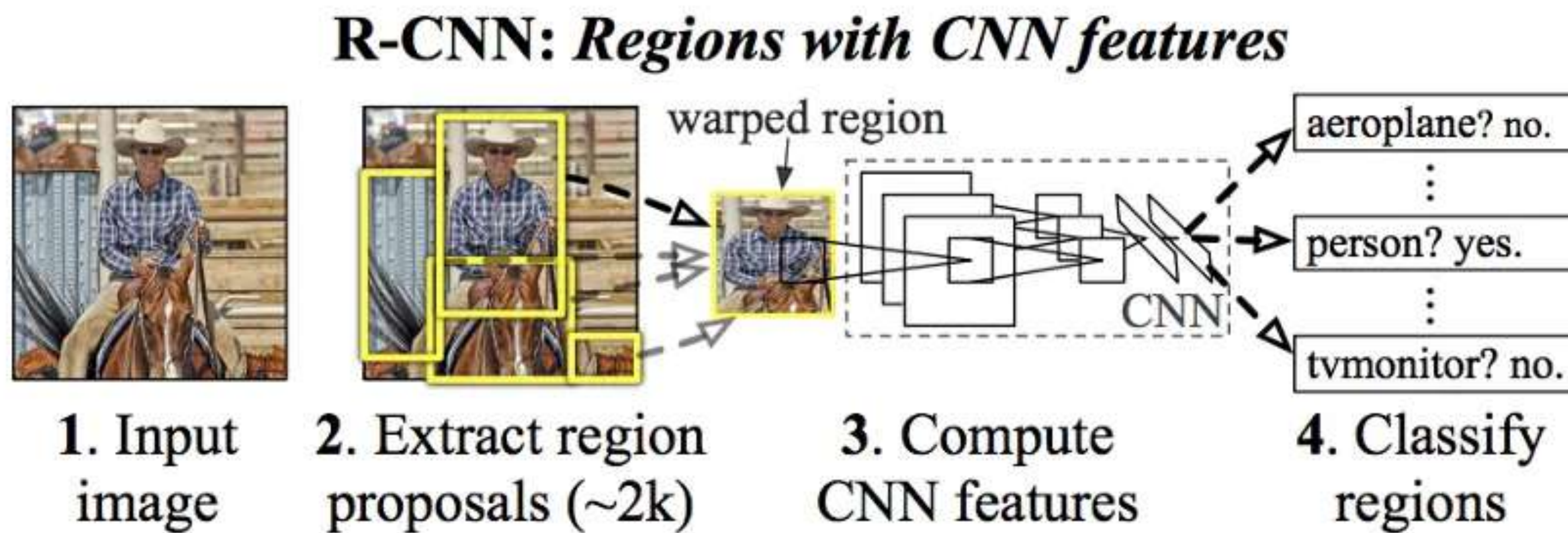
## Region-based Convolutional Neural Network (R-CNN)

- Es un detector de dos etapas
  1. Proposición de regiones
  2. Clasificación de regiones
- Basado este hay dos implementaciones comúnmente usadas: *Fast RCNN* y *Faster RCNN*





# R-CNN



*2014 - Rich feature hierarchies for accurate object detection and semantic segmentation*

- A. Primero **extraemos las regiones de interés** con un algoritmo como la *búsqueda selectiva*
- B. Redimensionar todos los regiones y **pasarlas por una CNN para la clasificación**



# Propuestas de regiones

- **Selective Search** es un algoritmo de propuesta de regiones que agrupa regiones en función de la intensidad de sus píxeles
- Agrupa los píxeles basándose en la agrupación jerárquica de píxeles similares.



<https://pyimagesearch.com/2020/06/29/opencv-selective-search-for-object-detection/>



# Selective Search

## Selective search

1. Comienza por sobre-segmentar la imagen basándose en la intensidad de los píxeles mediante un método de segmentación basado en gráficos
2. Añade todos los recuadros correspondientes a las partes segmentadas a la lista de propuestas regionales
3. Agrupa los segmentos adyacentes en **función de la similitud**
4. Ir al paso 2



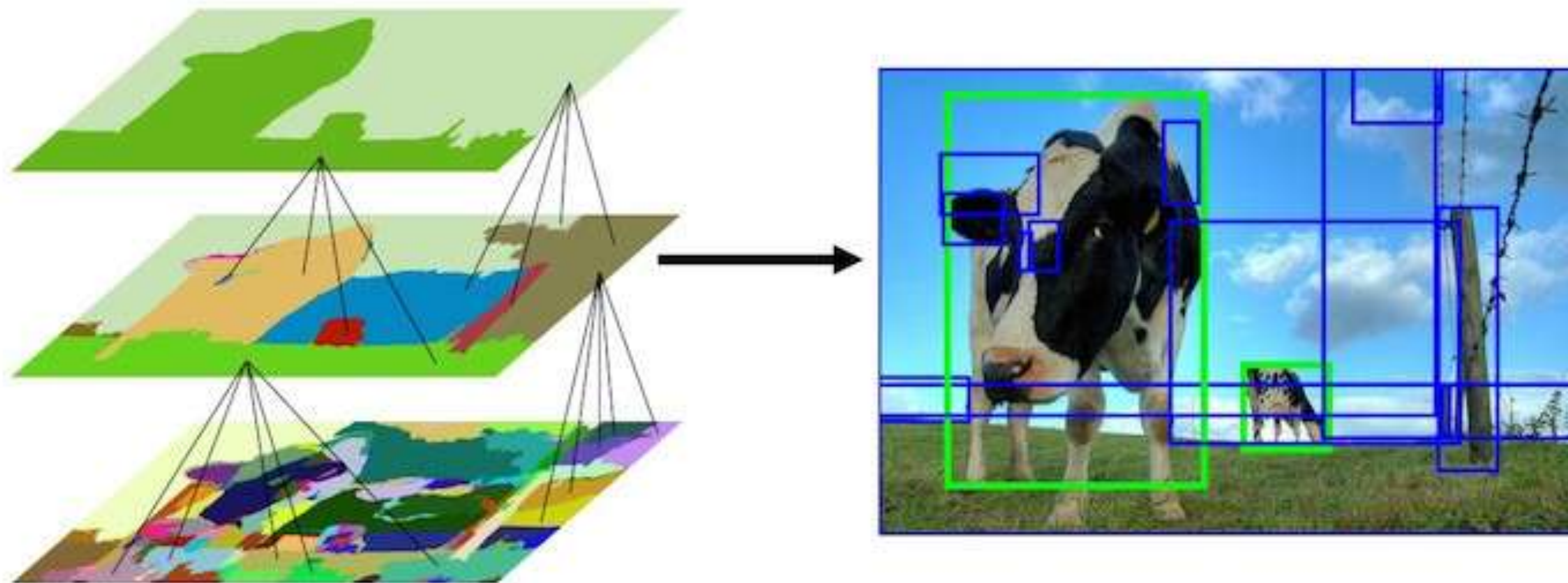
Imagen original



Over-segmented



# Selective Search

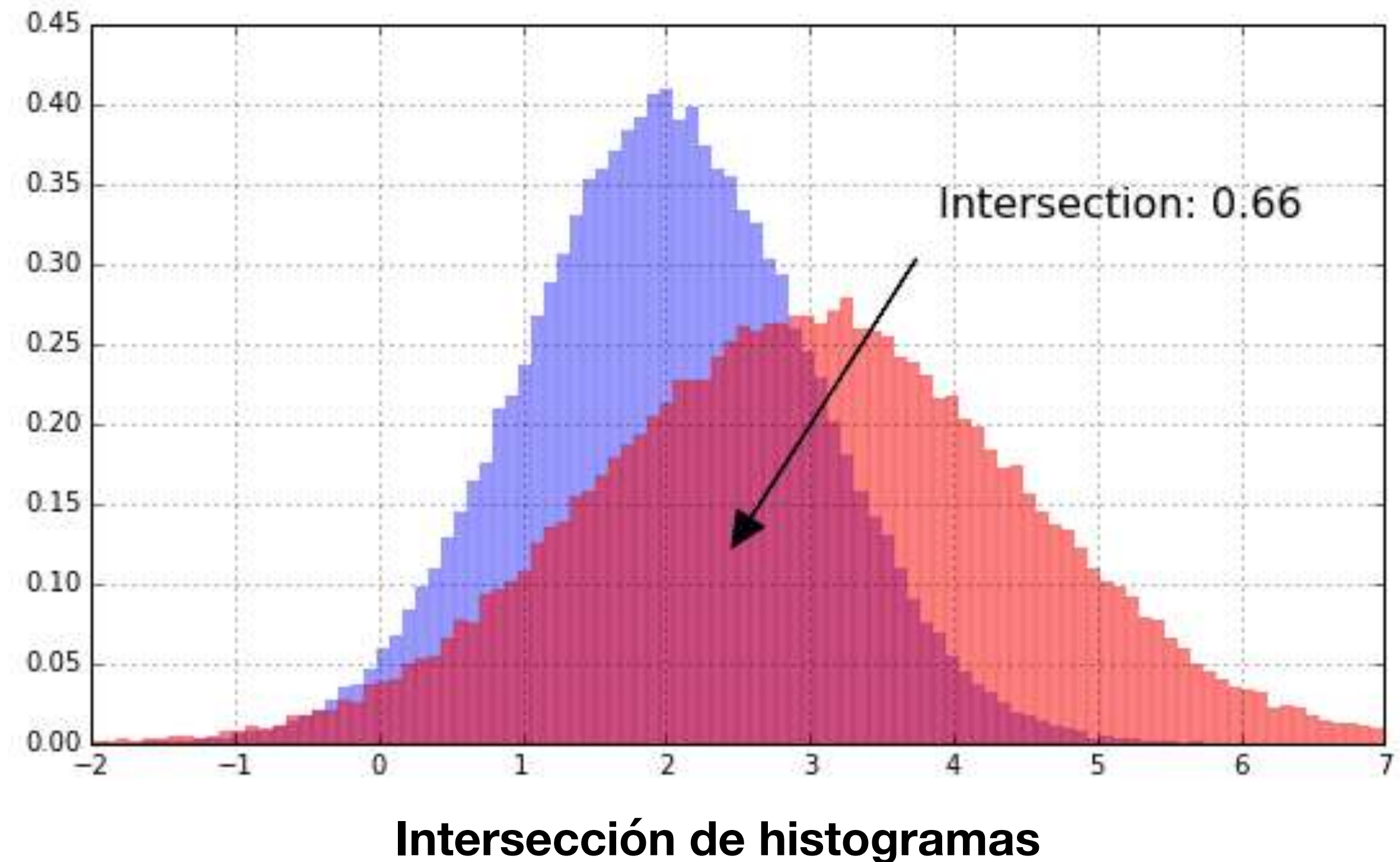


- En cada iteración, se forman segmentos más grandes y se añaden a la lista de propuestas de regiones
- Por lo tanto, creamos propuestas de regiones de segmentos más pequeños a segmentos más grandes en un enfoque ascendente



# Similitud entre recuadros

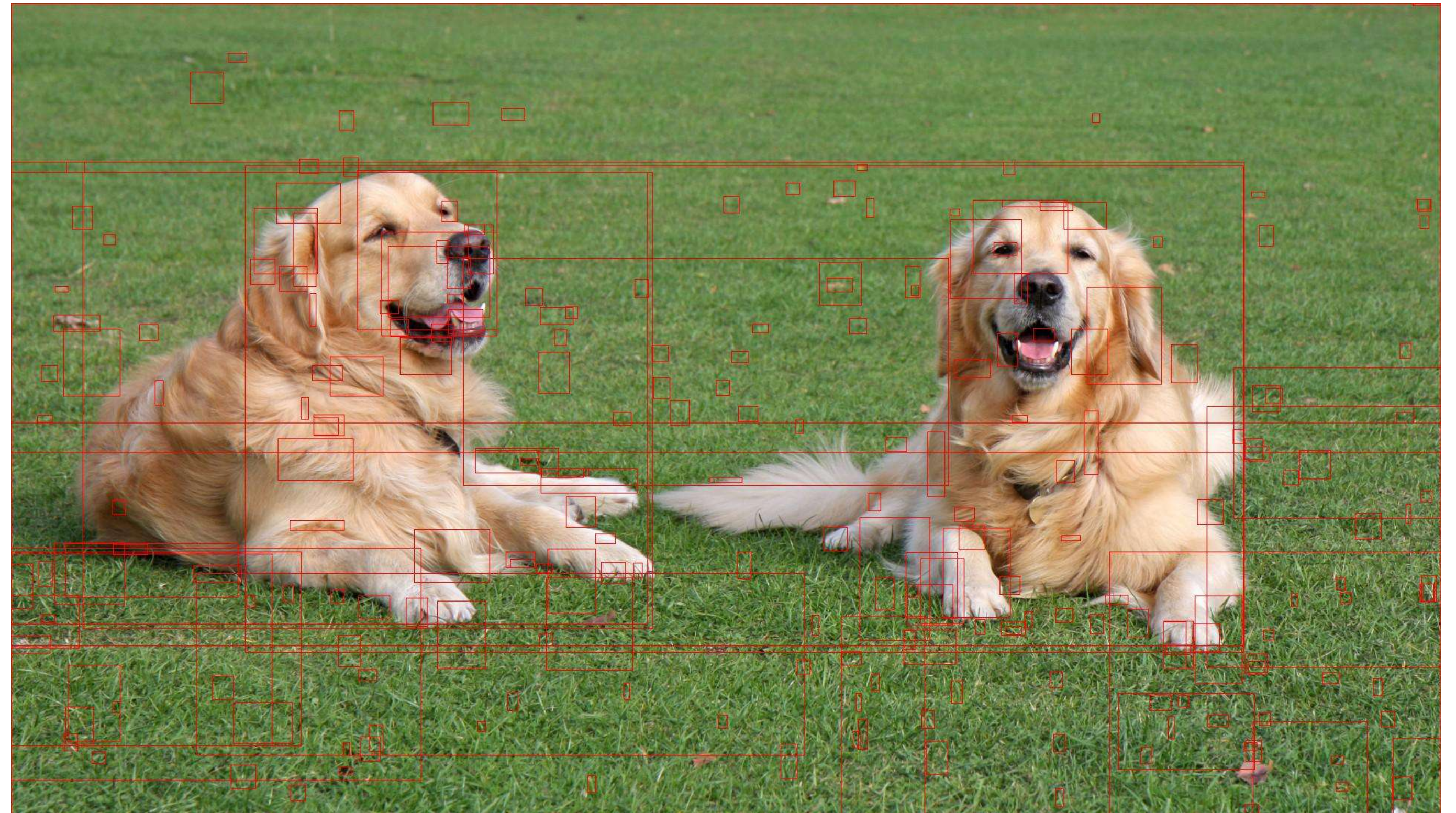
- La búsqueda selectiva utiliza **4 medidas de similitud**
  1. Intersección de histogramas de:
    - Color
    - Textura (derivados de color)
  2. Tamaño: anima a las regiones más pequeñas a fusionarse antes
  3. Compatibilidad de forma: lo bien que encajan dos regiones entre sí
- Similitud final:
$$s(r_i, r_j) = \alpha_1 S_{color} + \alpha_2 S_{texture} + \alpha_3 S_{size} + \alpha_4 S_{shape}$$





# Selective Search

- Es necesario etiquetar cada una de las regiones propuestas usando las etiquetas manuales.
- A cada región propuesta se le debe asignar a qué clase pertenece **dependiendo de su intersección con la etiqueta**




Dogs: top 250 region proposals



# IOU (Intersection over Union)

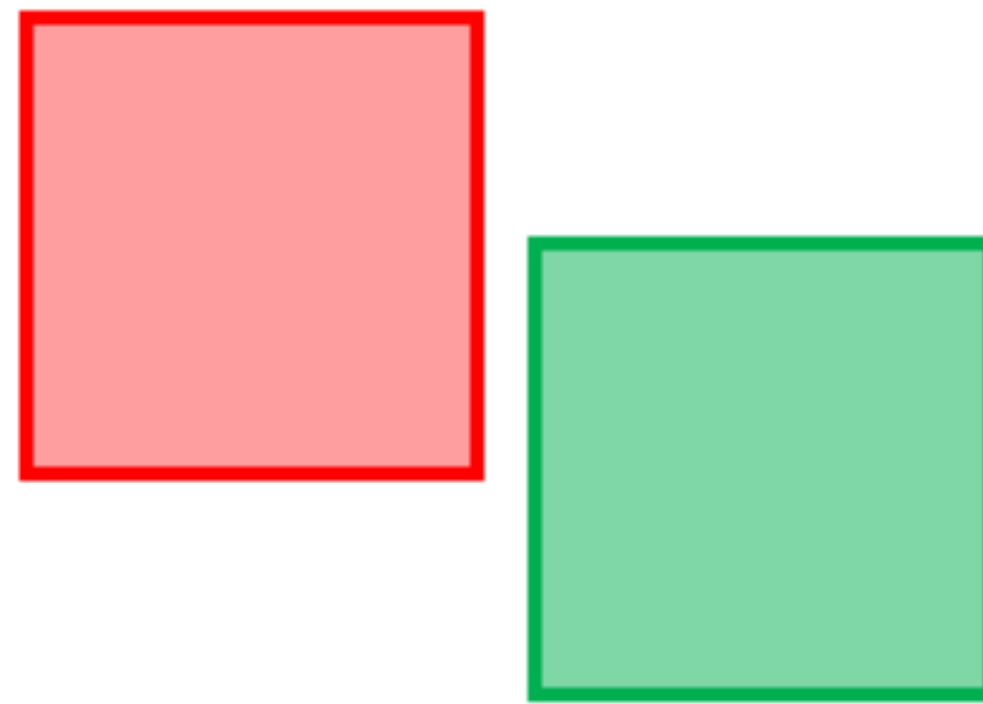
- Es un término utilizado para describir el grado de superposición (*overlap*) de rectángulos (*Bounding Box*)
- Cuanto mayor sea la región de superposición, mayor será el IOU
- IOU se utiliza principalmente en aplicaciones relacionadas con la detección de objetos

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


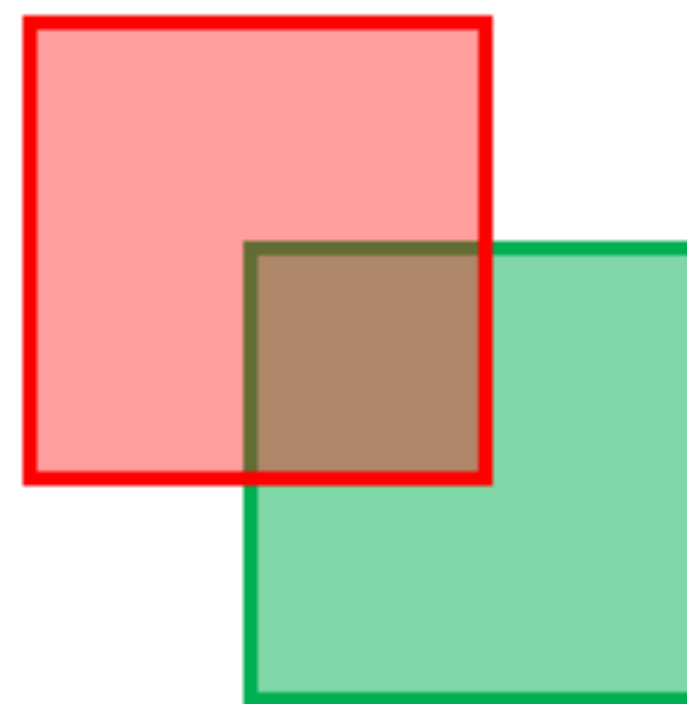
Intersection Over Union (IOU)



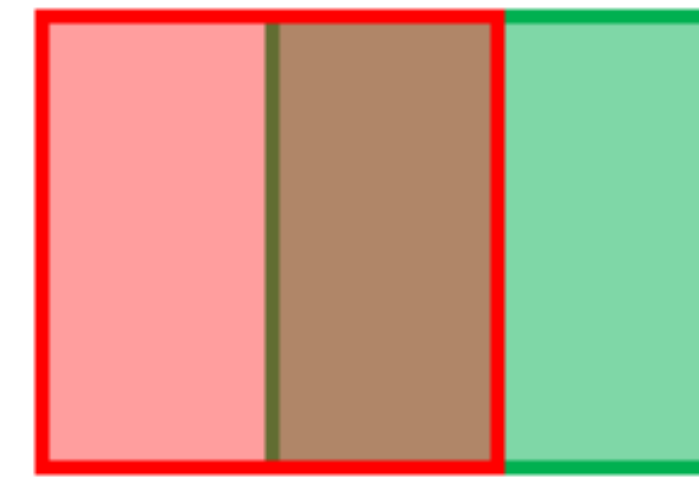
# IOU (Intersection over Union)



**IoU = 0**



**IoU = 0.142**



**IoU = 0.333**

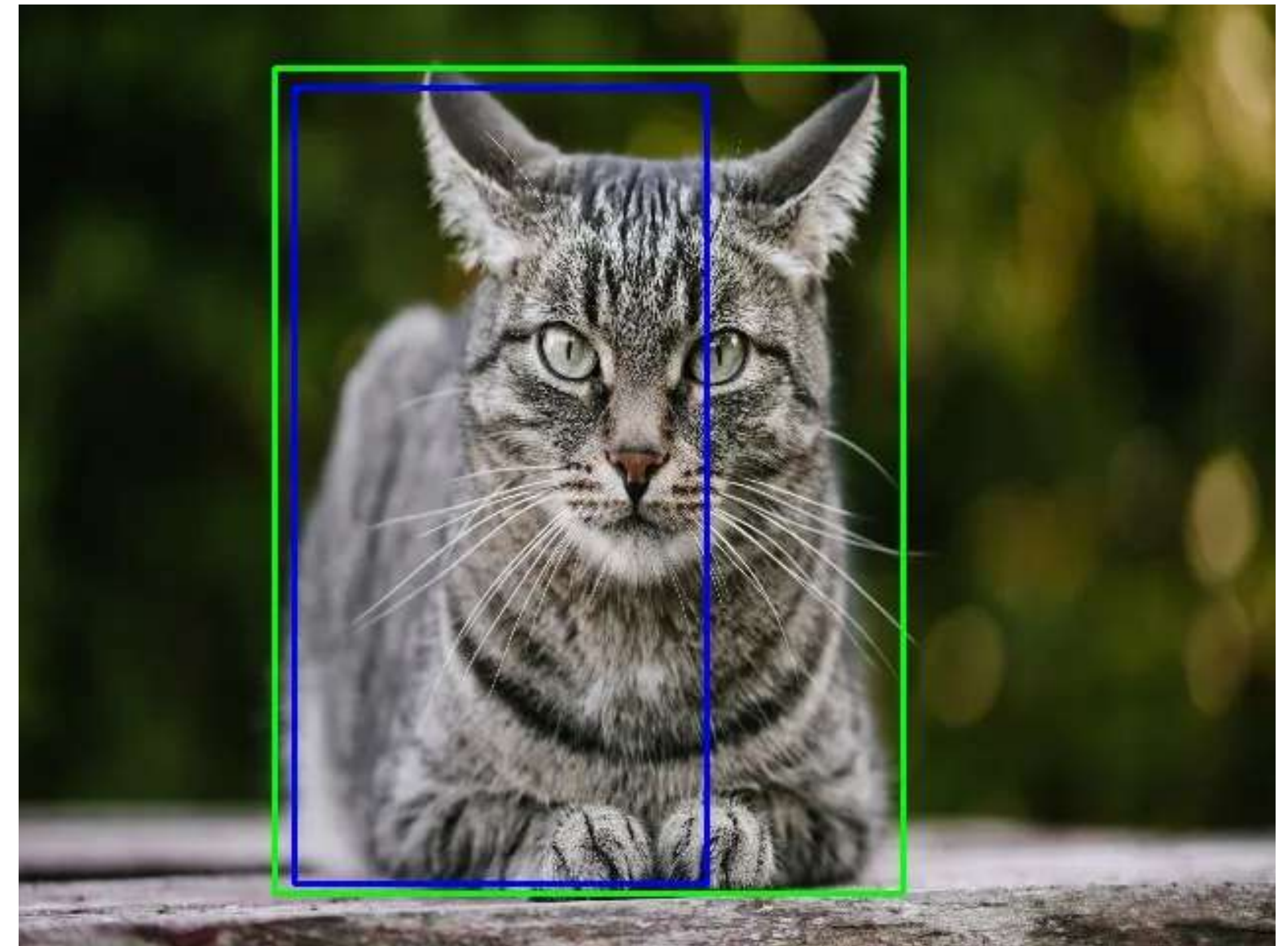


**IoU = 1**



# Label regions

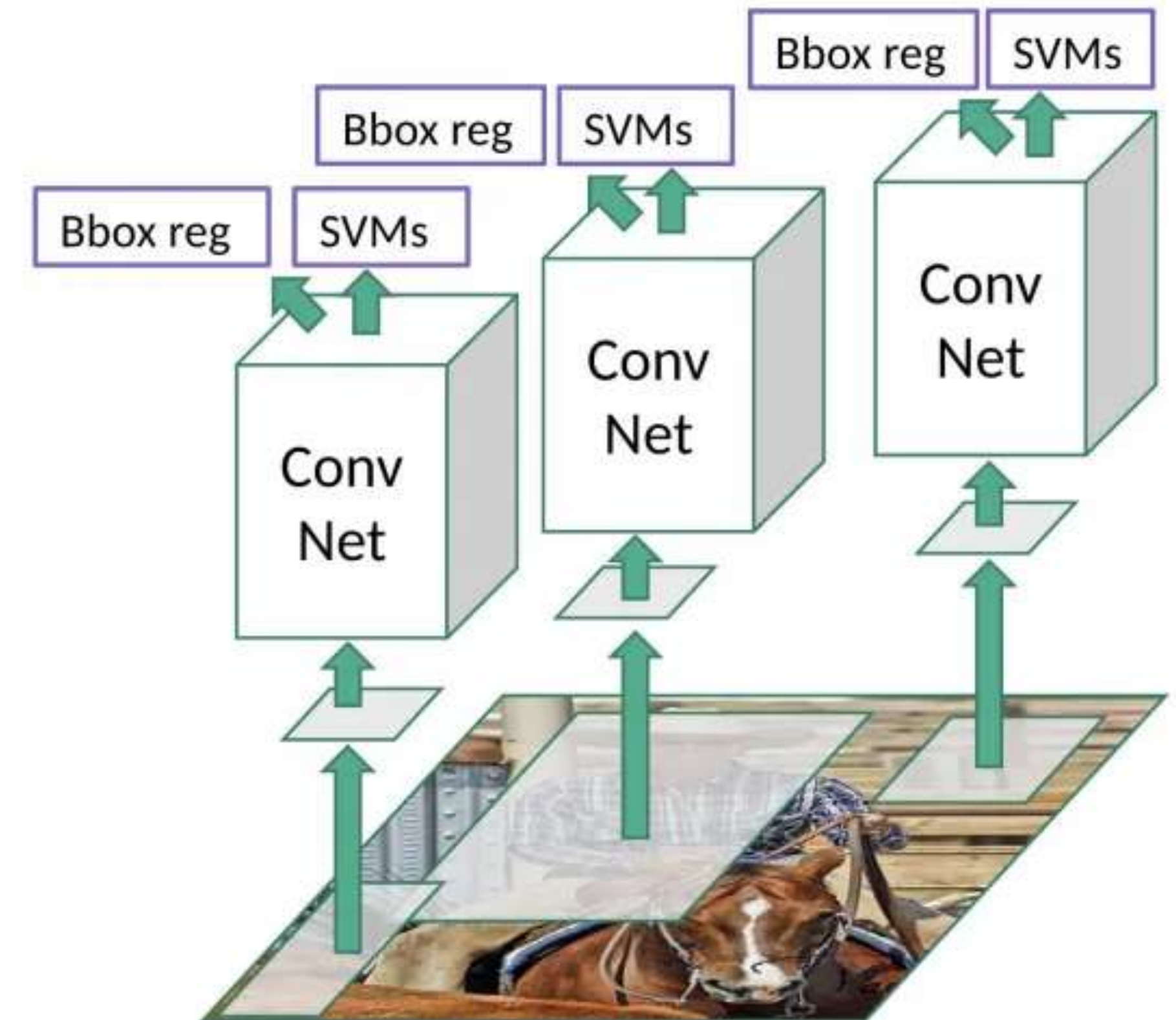
- Después de extraer nuestra propuesta de región, también tenemos que etiquetarlas para el entrenamiento.
- **Todas las propuestas que tengan un *IOU* de al menos 0.5 con la región de la etiqueta le asignamos la clase de la etiqueta**
- Todas las demás propuestas de región que tengan un *IOU* inferior a 0.3 se etiquetan como fondo.
- El resto simplemente se ignoran.





# Classification

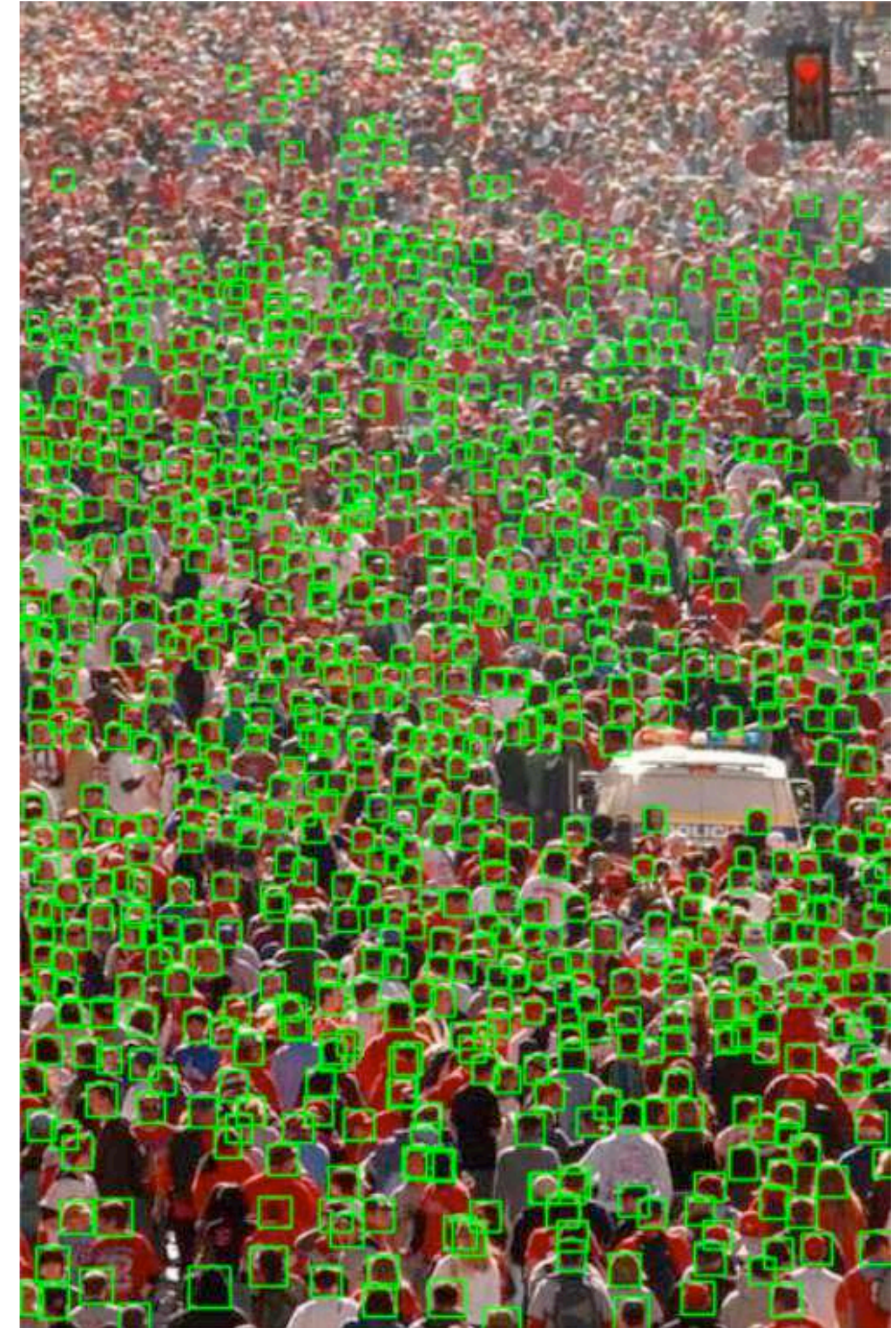
- Las regiones candidatas se deforman en un cuadrado y se introducen en una CNN que produce un vector de características como salida.
- La CNN actúa como un extractor de características
- las características extraídas se introducen en una *SVM* para clasificar la presencia del objeto dentro de esa propuesta de región candidata.
- También se predicen 4 valores de desplazamiento de las coordenadas del rectángulo para aumentar la precisión





# Problemas con R-CNN

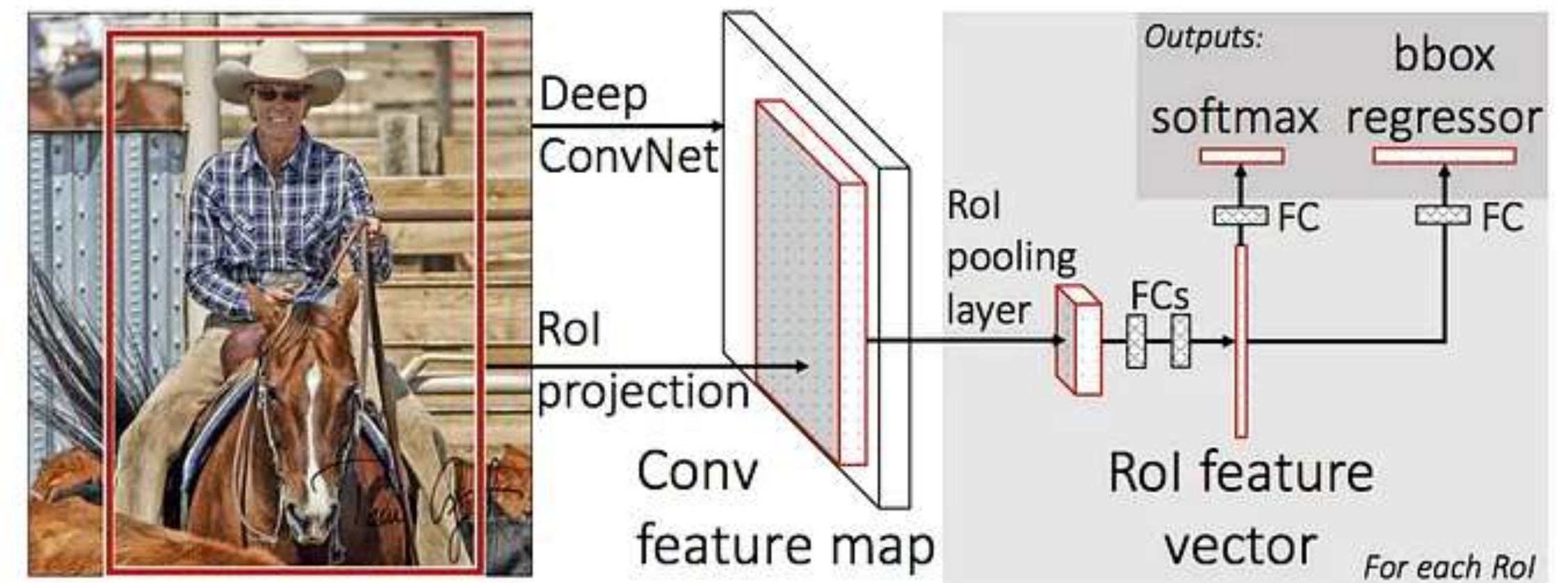
- El tiempo necesario para entrenar la red es enorme, ya que habría que clasificar cada región (~2000)
- No puede aplicarse en tiempo real, ya que tarda unos ~47 segundos por cada imagen
- El algoritmo de búsqueda selectiva es un algoritmo fijo. Por lo tanto, no se produce ningún aprendizaje en esa fase.





# Fast-RCNN

- El mismo autor resolvió algunos de los problemas para construir un algoritmo de detección de objetos más rápido y lo llamó *Fast R-CNN*
- El enfoque es similar al del algoritmo R-CNN, Pero:
- **Alimentamos la CNN con la imagen de entrada para generar un mapa convolucional de características.**

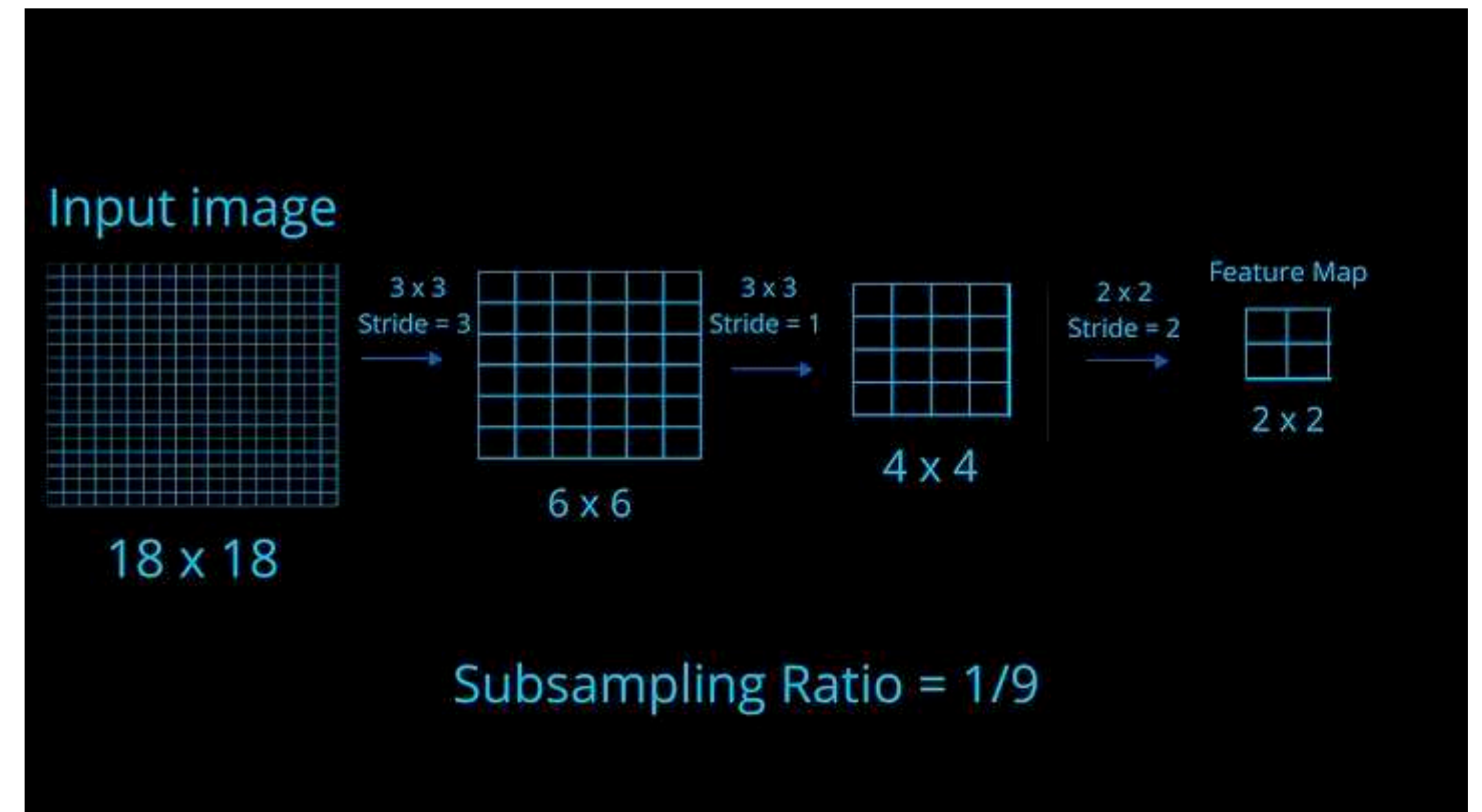


2015 - Fast R-CNN



# ROI projection

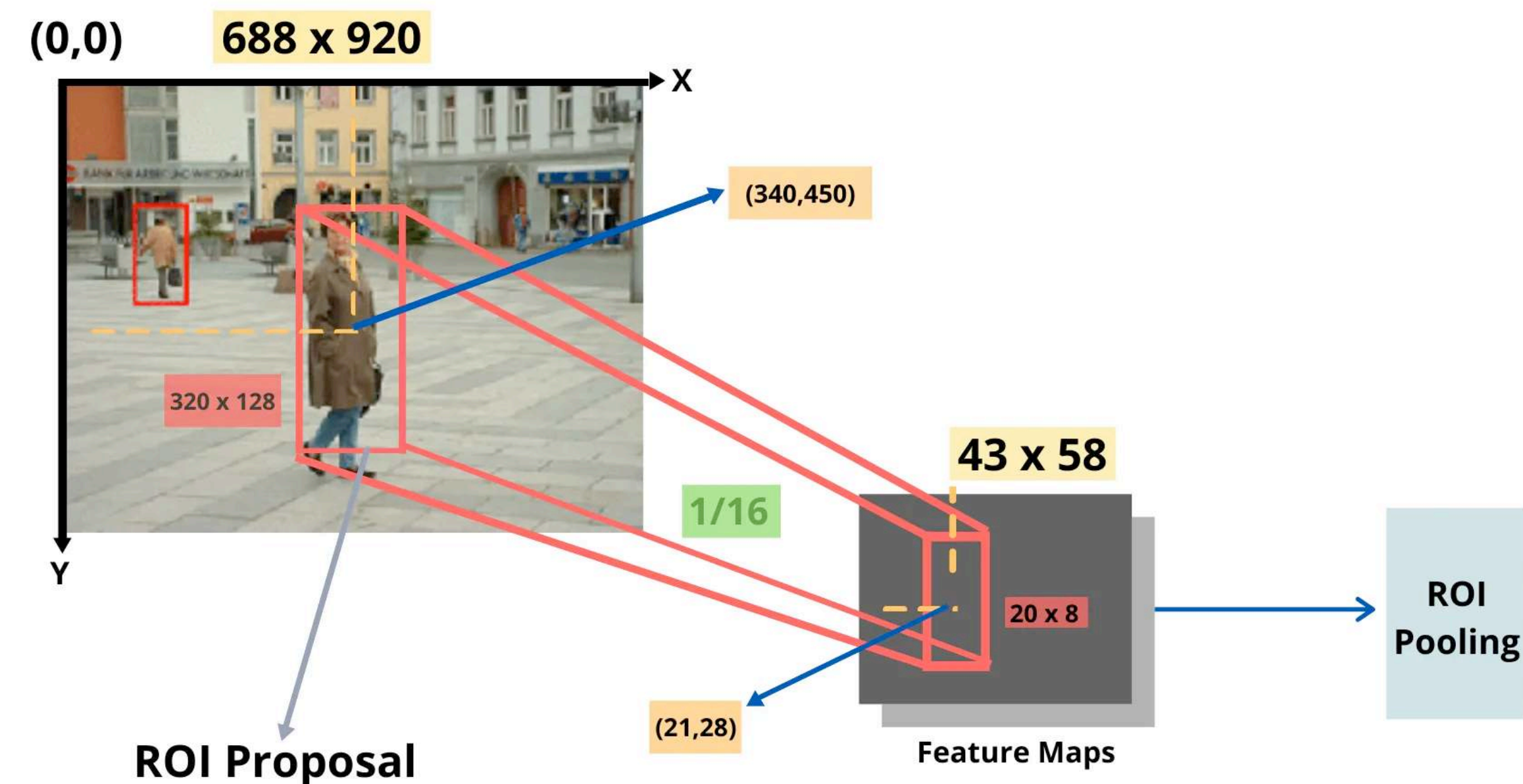
- Las regiones generadas por la búsqueda selectiva se proyectan sobre los mapas de características generados por la CNN.
- Este proceso se denomina Proyección ROI (Region Of Interest).
- Antes de la Proyección ROI, es necesario conocer la Relación de Submuestreo.





# ROI projection

- La idea de la proyección ROI es que obtenemos las coordenadas del rectángulo a partir de las propuesta de regiones y necesitamos proyectarlas sobre el mapa de características
- Una imagen de  $688 \times 920$  se alimenta a una CNN cuya relación de *submuestreo* es  $1/16$
- El mapa de características resultante tiene un tamaño de  $43 \times 58$
- Coordenadas son escaladas usando la transformación geométrica de escalado

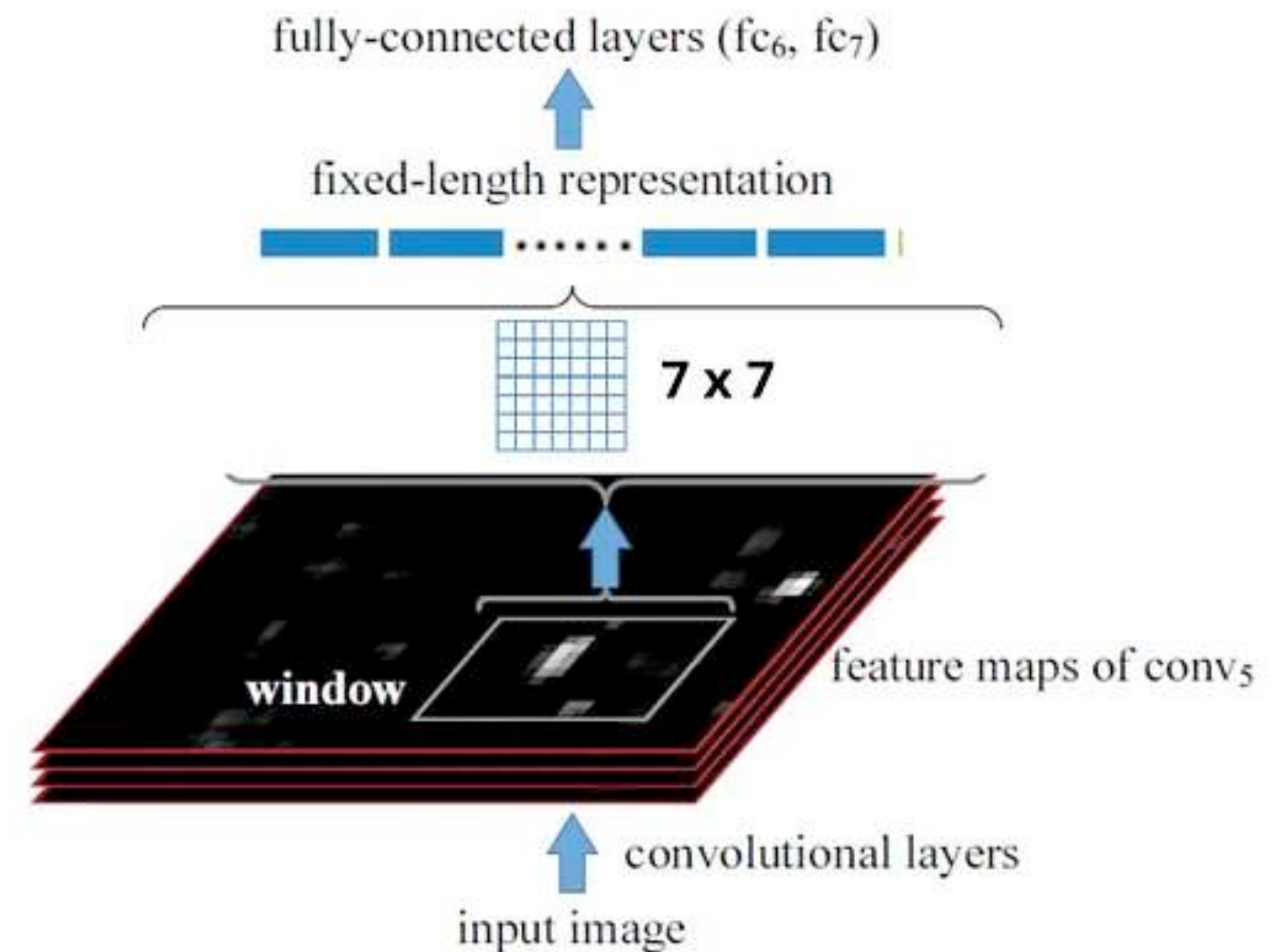


<https://towardsdatascience.com/understanding-fast-r-cnn-and-faster-r-cnn-for-object-detection-adbb55653d97>



# Fast-RCNN

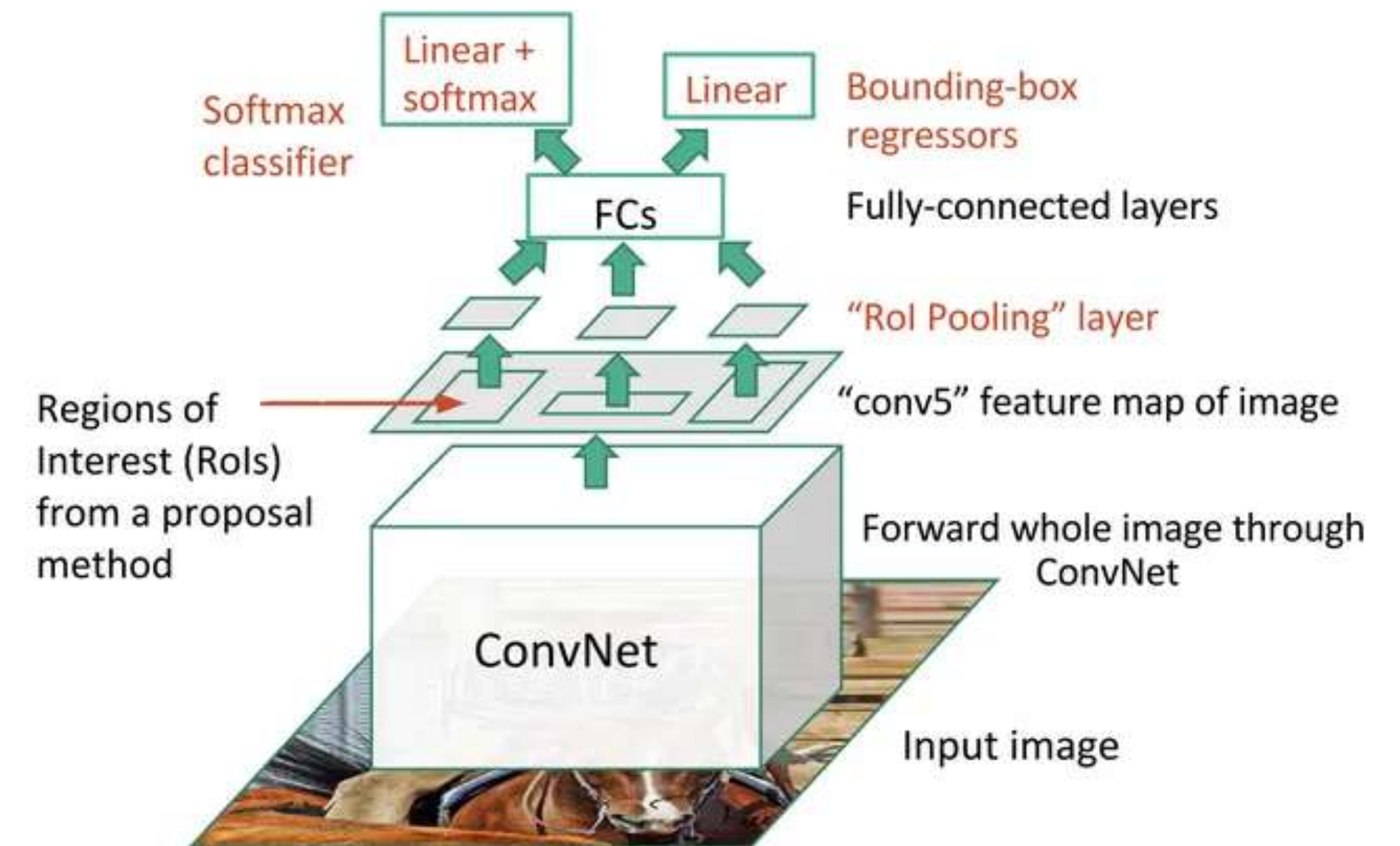
- La razón principal por la que se utilizan imágenes de tamaño fijo para la red se debe a las capas FC
- Esperan vectores de tamaño fijo, ya que hay pesos fijos asignados.
- Para resolver este problema, los autores de Fast R-CNN propusieron la idea de ROI Pooling
- La proyección de ROI en el mapa de características **se transforma en dimensiones fijas.**





# Fast-RCNN

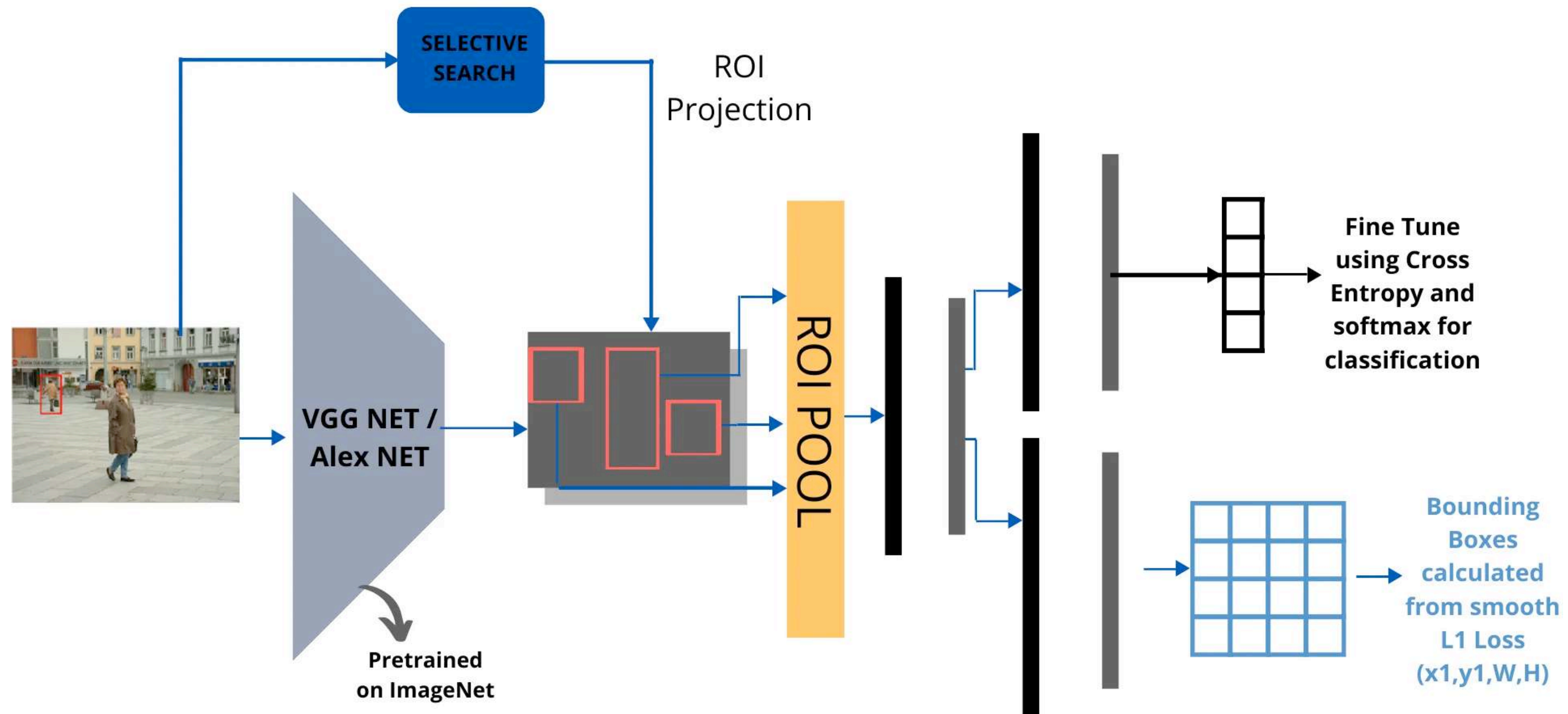
- Proyectamos la región de las propuesta en el mapa
- La deformamos en cuadrados utilizando una capa de *ROI Pooling*
- **Region of Interest Pooling:** ROI pooling produce mapas de características de tamaño fijo a partir de entradas no uniformes mediante un *max-pooling*



2015 - Fast R-CNN



# Arquitectura final de Fast RCNN





# Ejercicio

1. Leer el archivo en formato COCO ("bbox" : [x,y,width,height])
2. Implementar una función para calcular el IoU de dos rectángulos
3. Determinar los rectángulos con mayor IoU

$\text{IoU} =$

