

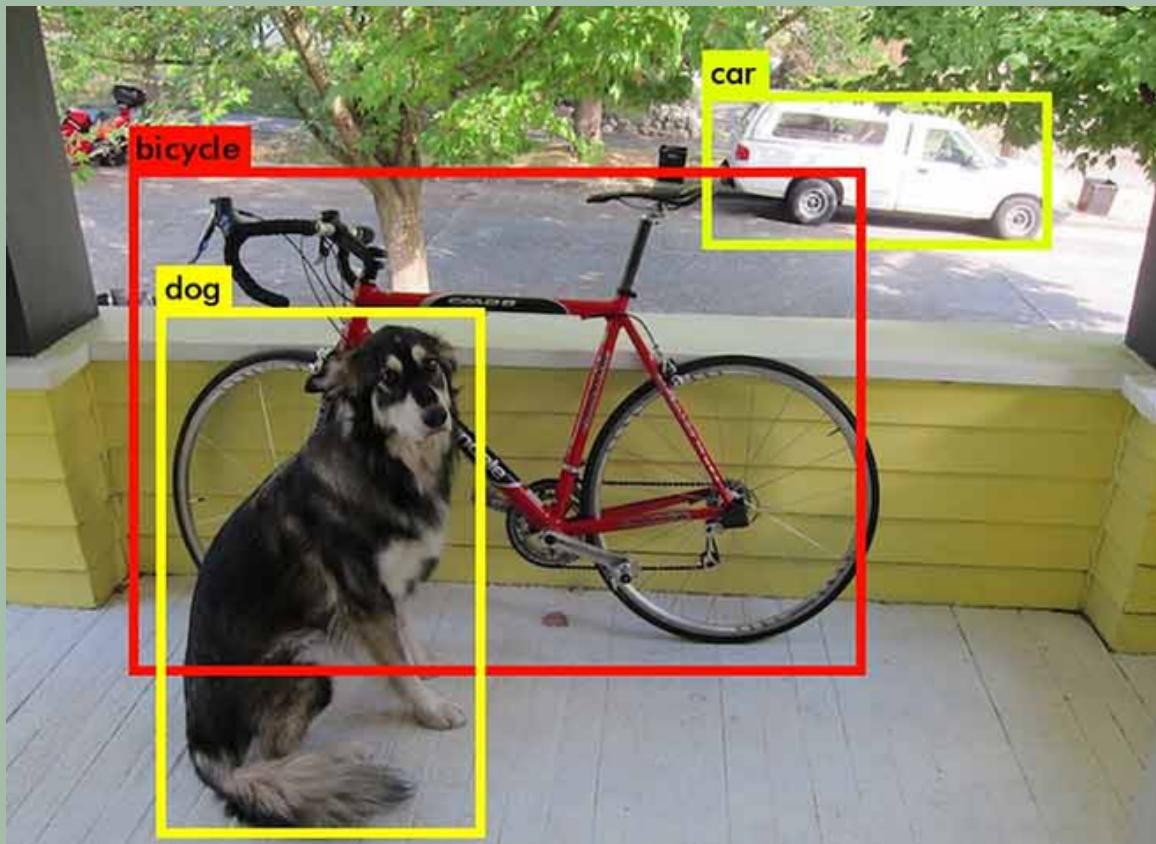
Localización de un objeto en una imagen



Introducción

La visión computacional es una rama que ha crecido demasiado en los últimos años, y una de las muchas tareas que busca resolver es la localización de objetos en una imagen.

El gran problema es que las entradas del algoritmo de localización no son fijas, pueden existir muchos objetos y estar posicionados en distintas regiones de la imagen, en diferentes imágenes.

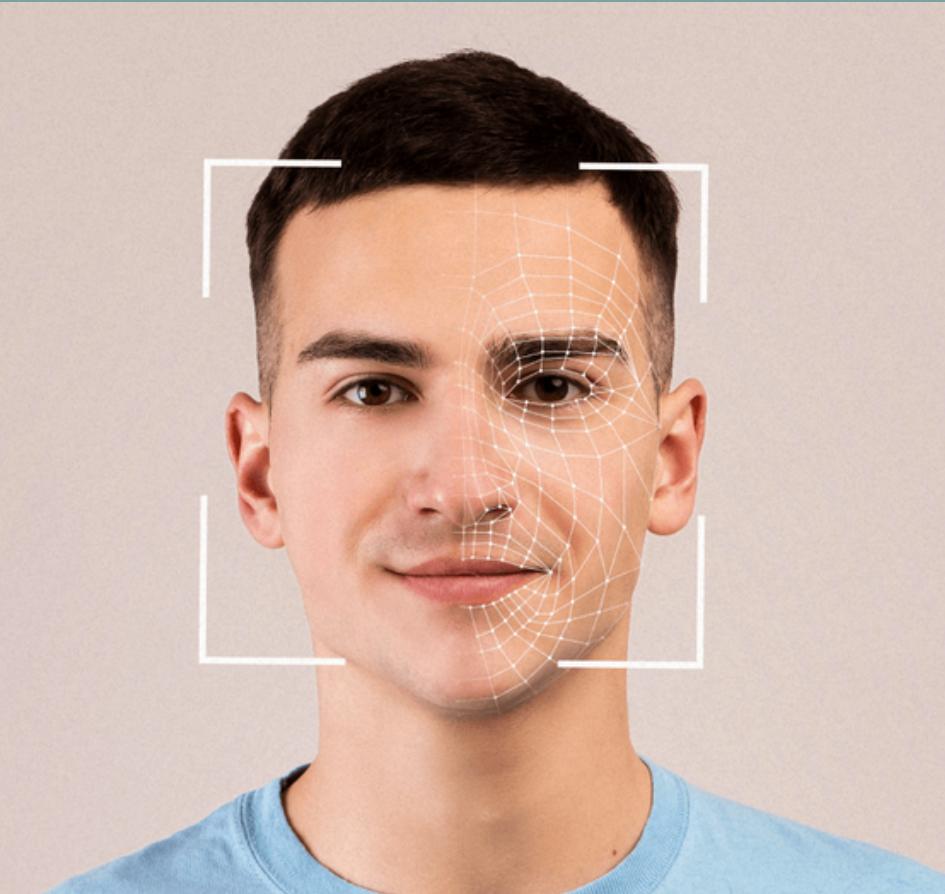


Hipótesis...

Al entrenar la red neuronal fast R-CNN utilizando conjuntos de datos compuestos exclusivamente por imágenes que contienen un solo objeto, se espera lograr una especialización más efectiva en la detección de ese objeto específico. Al proporcionar a la red un conjunto de entrenamiento más homogéneo y enfocado, se espera que aprenderá características distintivas del objeto sin distracciones de otros elementos en la escena. Este enfoque tiene el potencial de mejorar el rendimiento de la red en aplicaciones prácticas donde la detección precisa de un solo objeto es esencial

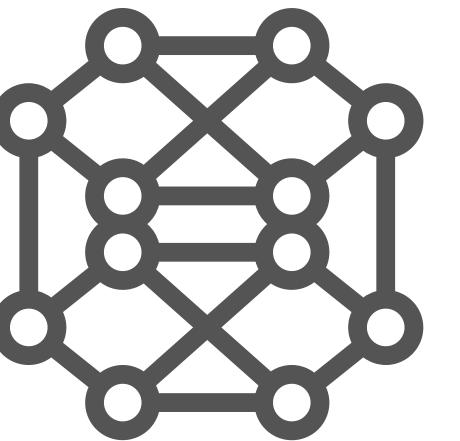
Objetivo

Utilizar una red neuronal pre-entrenada con estructura fast R-CNN para lograr la detección de un solo objeto en una imagen, queremos intentar especializar más la red para que sea más precisa y logre la localización del objeto con mayor precisión, dentro de nuestro objetivo está el evaluar la red para intentar llegar a un buen funcionamiento.



Recursos y métodos utilizados

- Arquitectura: Fast R-CNN, específicamente es la resnet50
- Dataset: COCO
- Entorno de desarrollo: Kaggle



coco: Common Objects in Context (2017)

- Ampliamente utilizado en el campo de visión computacional
- Se puede utilizar para resolver distintas tareas
- La mayoría de las imágenes son de personas



coco: Retos

- Como tiene tantas aplicaciones, tuvimos que realizar un filtrado de los datos
- Era un dataset muy pesado, y con datos muy variantes
- El filtrado ayudó a normalizarlos
- Sólo nos interesaba un campo de las anotaciones: **box (bounded box)**

Algunas particularidades del dataset:

- Tiene aproximadamente 91 categorías
- Las anotaciones (JSON) cuentan con:
 - id de la anotación
 - id de la imagen a la que pertenece segmentación
 - bounded box
 - id de la categoría del objeto

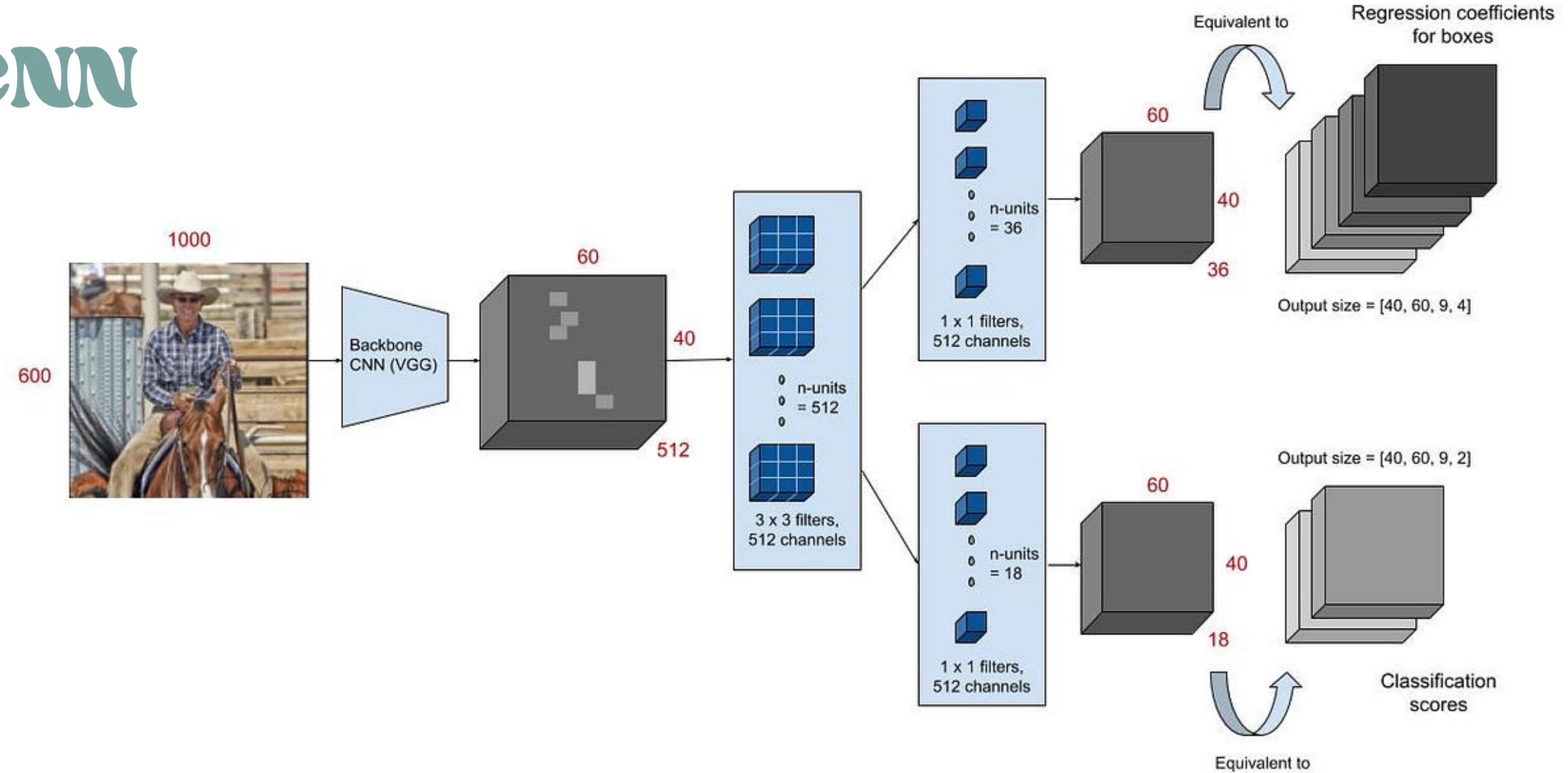
R-CNN

- Consisten en usar la búsqueda selectiva para obtener propuestas de región a partir de las cuales discriminar.
- Se apoya de un algoritmo (greedy) denominado búsqueda selectiva
- Las propuestas son deformadas a una imagen cuadrada, que en una red convolucional es aplanada a un vector de características.
- Este vector de características es dado como entrada a una máquina de soporte vectorial de forma que prediga la presencia de un objeto en la propuesta de región

R-CNN: búsqueda selectiva

1. Generar una subsegmentación inicial, generamos muchas regiones candidatas.
2. Utilizar un algoritmo voraz para combinar de manera recursiva regiones similares en regiones más grandes.
3. Utilizar las regiones generadas para producir las propuestas finales de regiones candidatas.

R-CNN



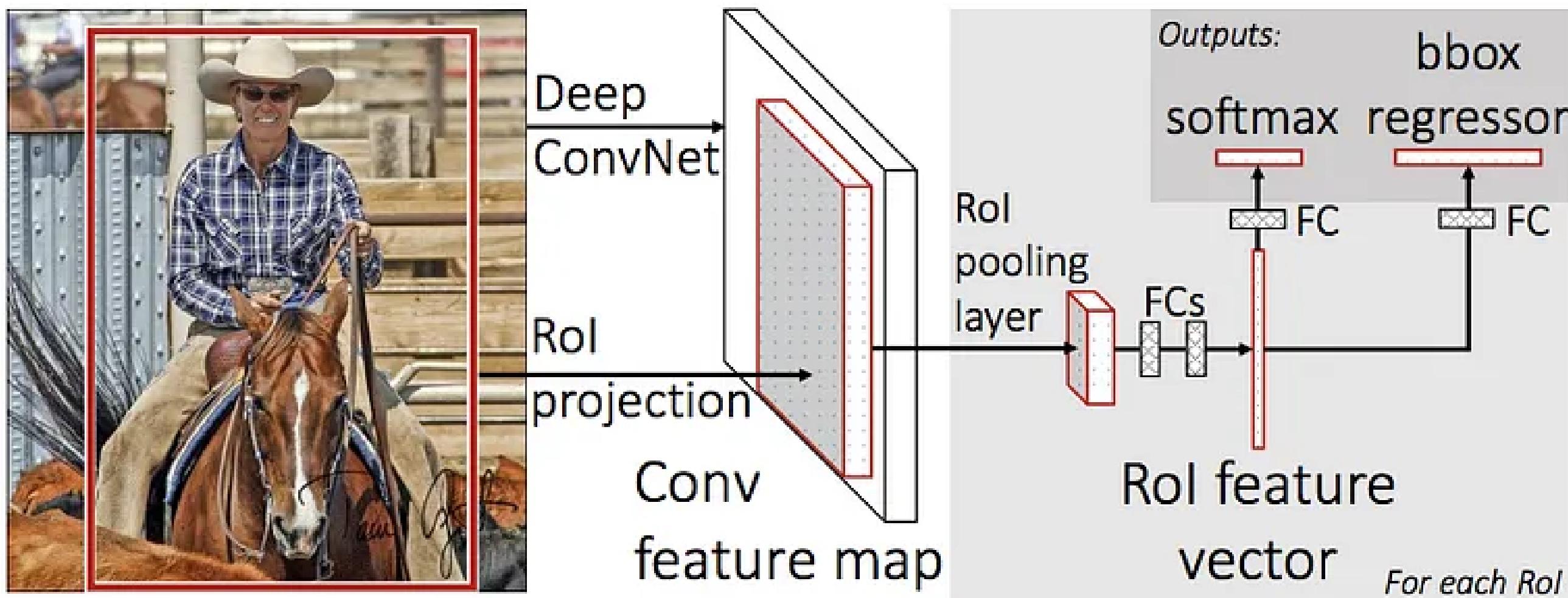
Fast R-CNN

- La red tiene un algoritmo estático que puede no ajustarse a los datos al ser de naturaleza glotona.
- Se resolvió sustituyendo el algoritmo de búsqueda selectiva, por alimentar la misma red convolucional con la imagen de entrada con el fin de generar un mapa de características, con este se puede con ayuda de la búsqueda selectiva, identificar las propuestas de región e igual que con la red R-CNN se puedan reescalar a imágenes cuadradas pero ahora con una capa de RoI pooling (Region of Interest)

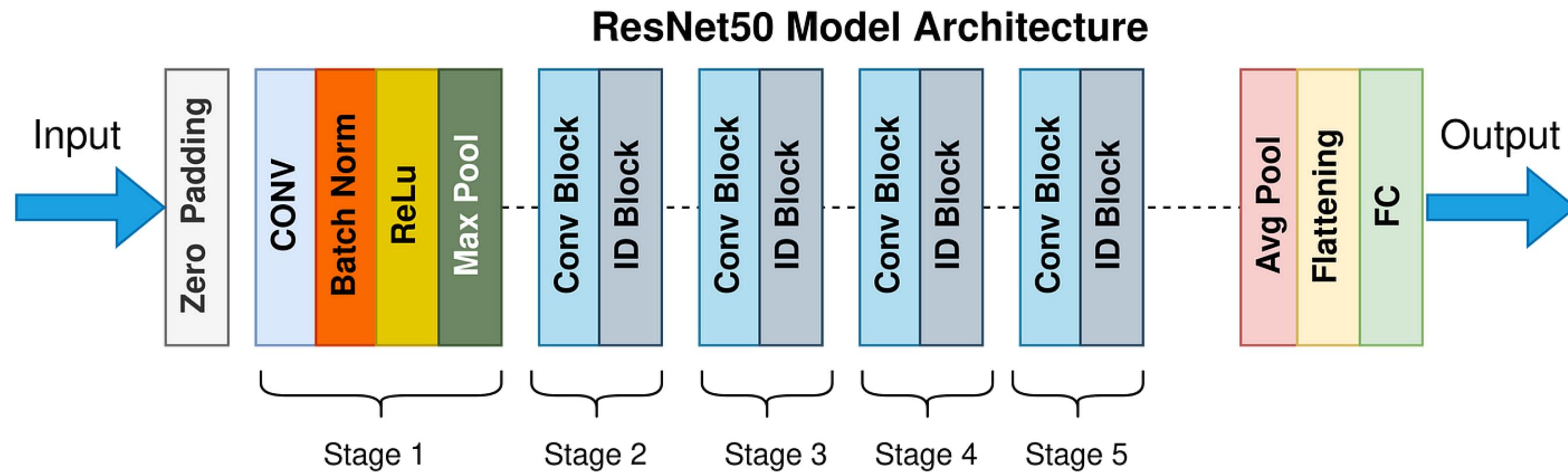
Fast R-CNN

- Esta última capa es la encargada de predecir la clase de los objetos así como los valores de ajuste de la caja.
- **Faster RCNN**: evita el uso de la búsqueda selectiva para encontrar propuestas de regiones. Ya que idénticamente que la red Fast R-CNN produce un mapa de características, solo que en lugar de usar un algoritmo glotón para la búsqueda de propuestas de región usa una red para predecir las regiones conocida como Region Proposal Network (RPN)

Fast R-CNN

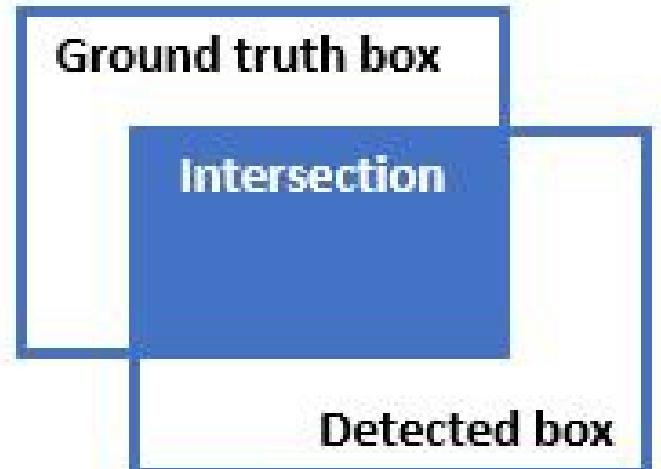


ResNet 50: capas usadas



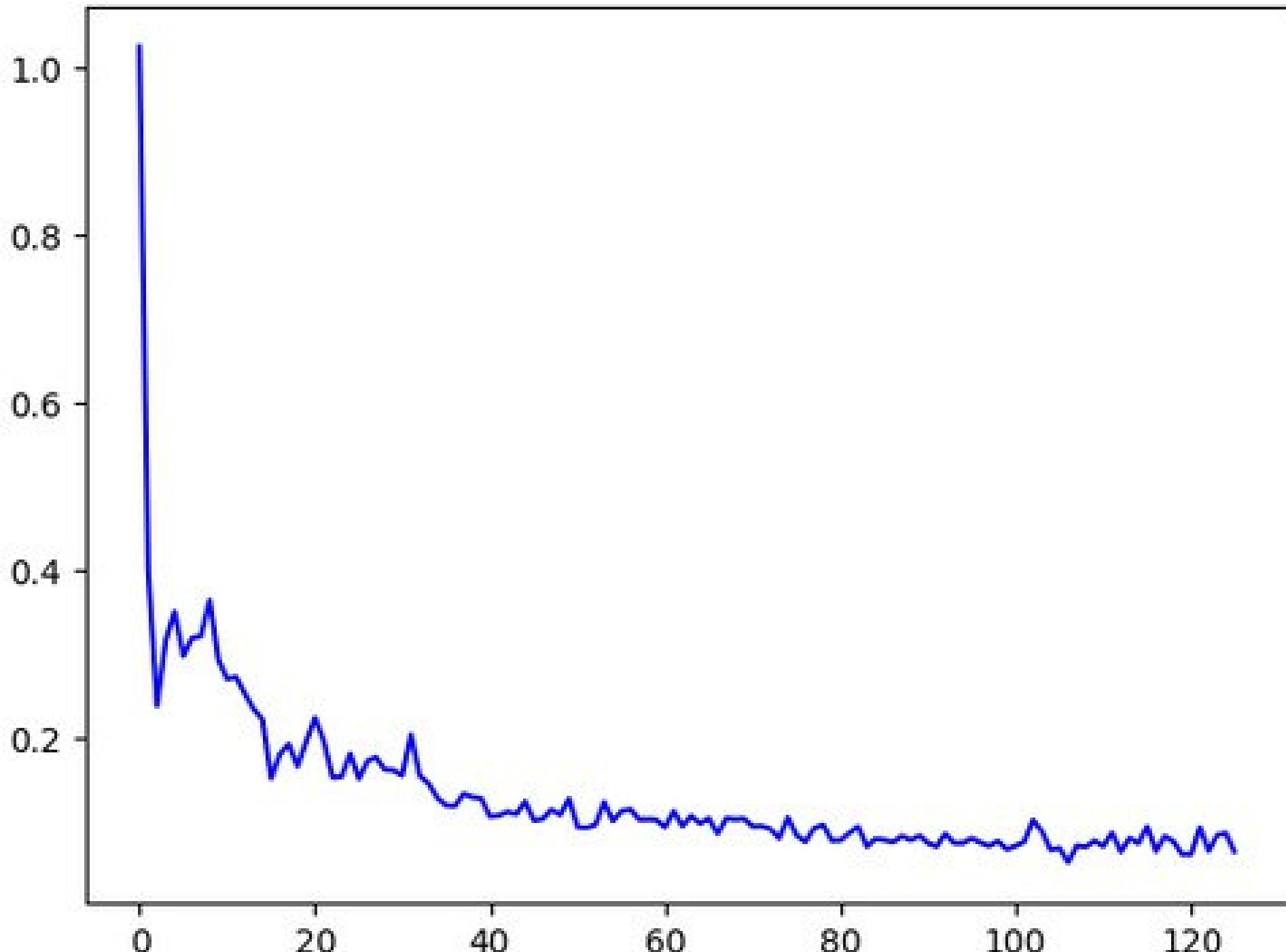
Evaluación: Intersection over union

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



Resultados: función de riesgo

Ejecutando el modelo con 3 épocas,
todo el conjunto de datos y batches
de 10



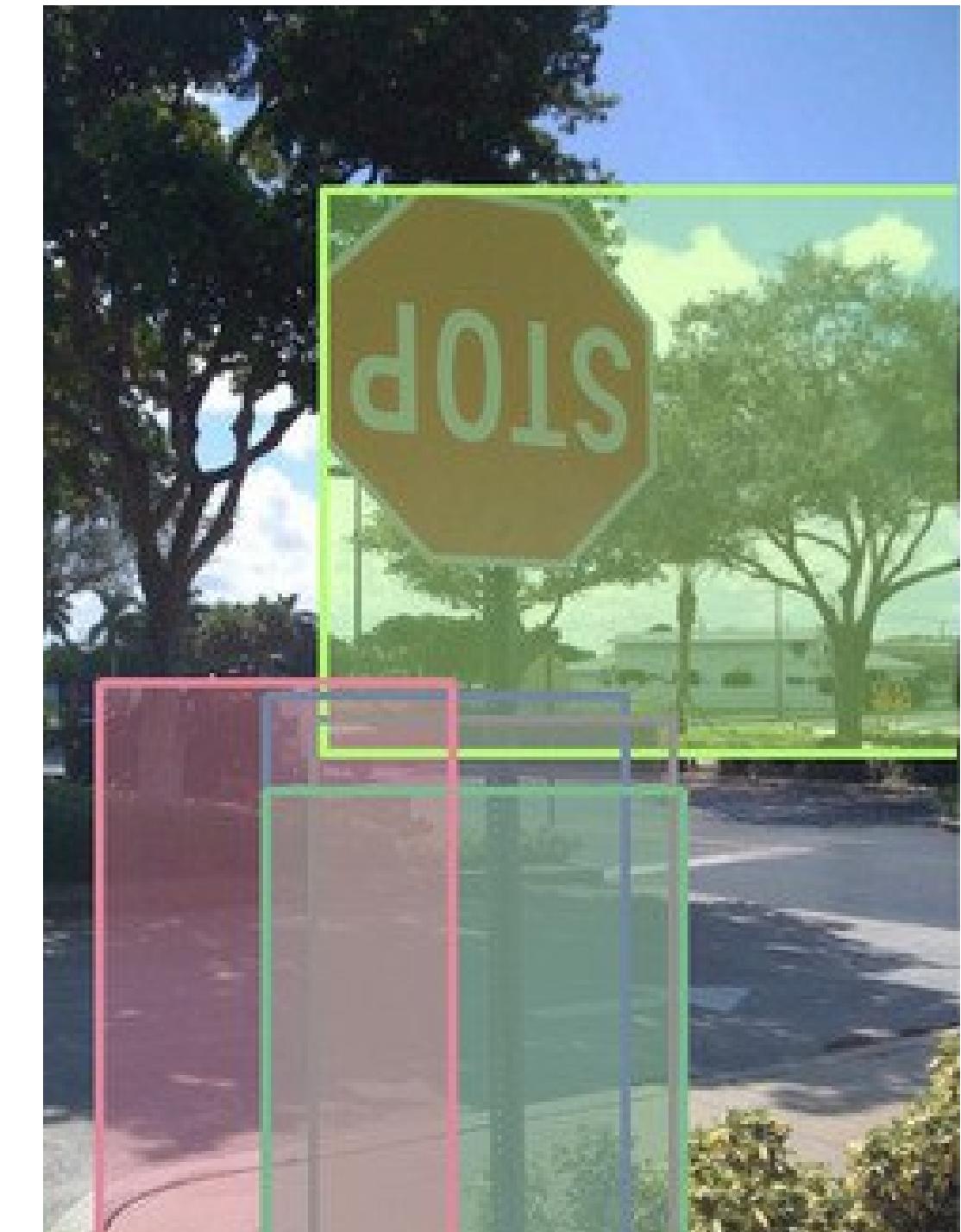
Resultados



Resultados



Resultados



¿QUÉ PROBLEMAS IMPLICA LA LOCALIZACIÓN DE OBJETOS EN UNA IMAGEN?

- La entrada de nuestro programa será siempre muy variable, tanto en el tamaño de las imágenes como en la cantidad de objetos que puede llegar a haber
- El procesamiento de las regiones que tendremos en la imagen es muy pesado, ya que tenemos muchísimas regiones
- Diferenciar cuando una región pertenece o no al mismo objeto

Conclusiones

La detección de imágenes es un problema muy complejo que implica resolver muchos retos, como los que mencionamos anteriormente. Sin embargo para poder entrenar bien un modelo es necesario mucha infraestructura y recursos, ya que las imágenes son un tipo de dato muy complejo, y procesarlas por una red compleja toma mucho tiempo. Entrenar una red para detectar un solo objeto puede ser complicado, porque la mayoría de las imágenes hay más de un objeto (aunque no estén clasificados), o algunos objetos se componen por varios objetos.

El dataset de COCO tiene la virtud de que son imágenes en contexto, pero eso implica que no hay objetos aislados, sería bueno ver como entrena un modelo con imágenes de objetos aislados.

Bibliografía

- Gandhi, R. (2018, July 9). R-CNN, fast R-CNN, Faster R-CNN, YOLO - object detection algorithms. Medium. <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>
- Ren, S., He, K., Girshick, R., & Sun, J. (2017b). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149. <https://doi.org/10.1109/tpami.2016.2577031>

Gracias

Los máquinas