

# Curso de Detección y Segmentación de Objetos con TensorFlow

Sergio Paniego Blanco



@sergiopaniego





# Requisitos del curso



**Redes Neuronales con Python y Keras.**



**Redes Neuronales Convolucionales con Python y Keras.**



**Profesional de Redes Neuronales con TensorFlow.**



# Proyectos del curso

- **Object detection en un problema de conducción autónoma.**
- **Segmentación semántica en un problema de conducción autónoma.**



# ¿Por qué realizar este curso?

- Conocer técnicas más allá de clasificación dentro de computer vision.
- Utilizar modelos pre-entrenados de object detection y segmentation.
- Conocer diferentes datasets utilizados por la comunidad.

**¿Qué es la visión  
computarizada y  
cuáles son sus tipos?**



# Ejemplos de casos de uso





# Clasificación



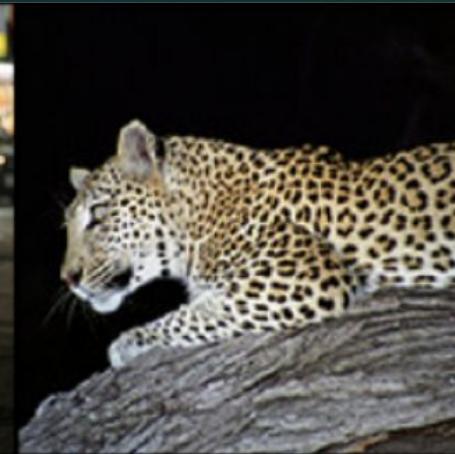
mite



container ship



motor scooter



leopard

mite	mite	container ship	motor scooter	leopard
black widow		lifeboat	go-kart	jaguar
cockroach		amphibian	moped	cheetah
tick		fireboat	bumper car	snow leopard
starfish		drilling platform	golfcart	Egyptian cat



# Clasificación



grille



mushroom



cherry

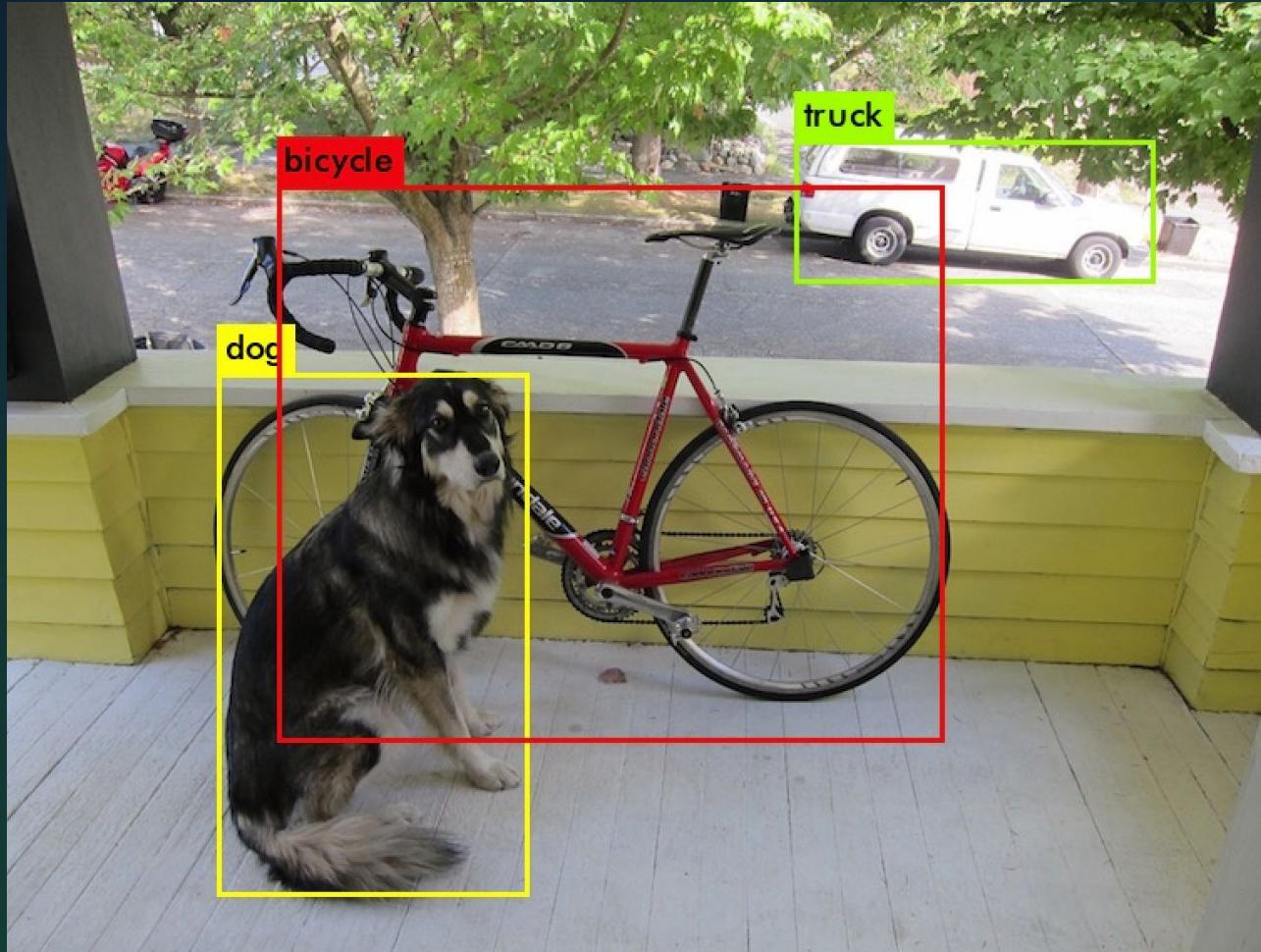


Madagascar cat

convertible		agaric	dalmatian	squirrel monkey
grille		mushroom	grape	spider monkey
pickup		jelly fungus	elderberry	titi
beach wagon		gill fungus	ffordshire bullterrier	indri
fire engine		dead-man's-fingers	currant	howler monkey

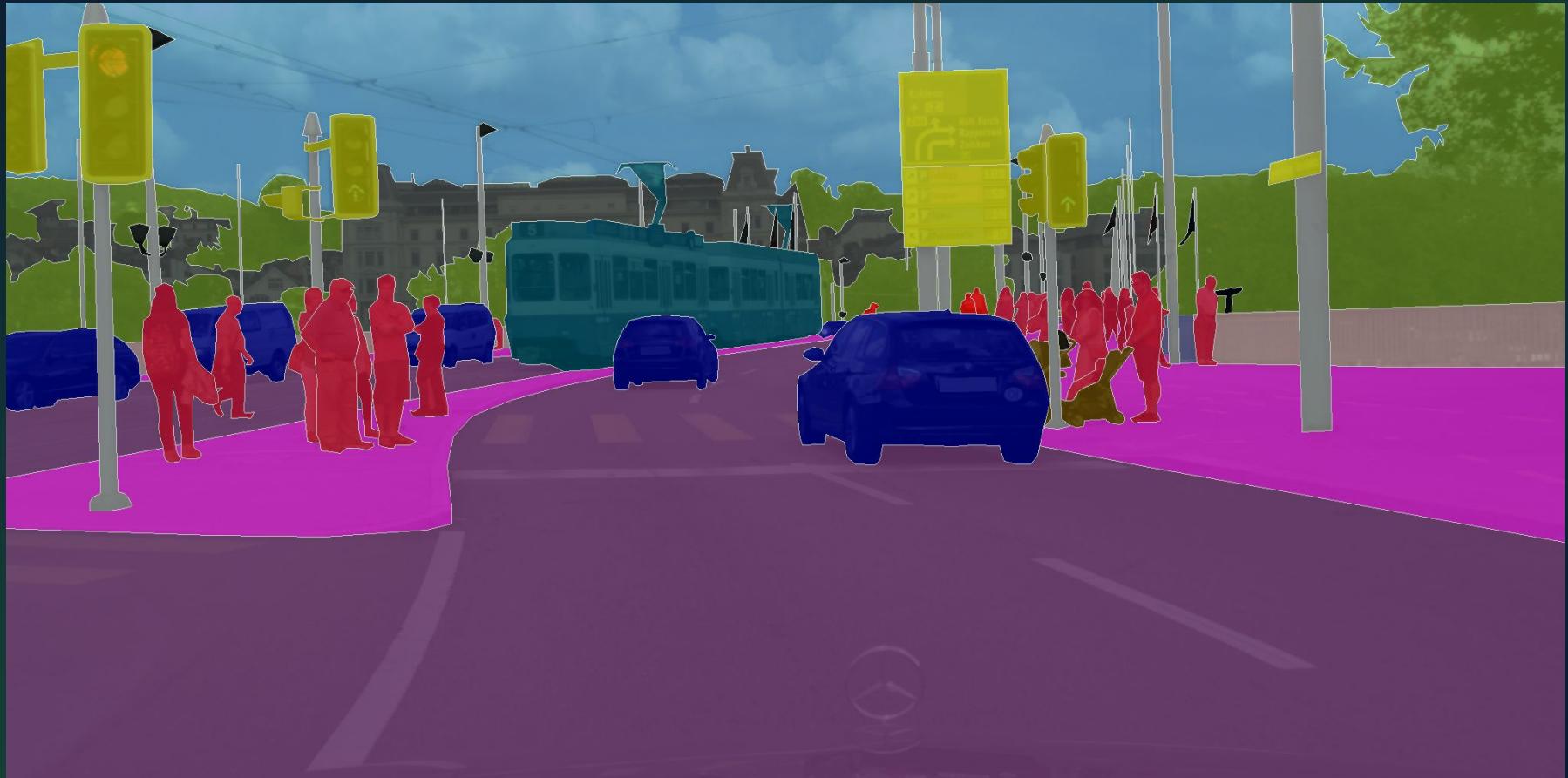


# Object detection





# Segmentation



# Introduciendo detección de objetos I



# Localización + detección

Classification



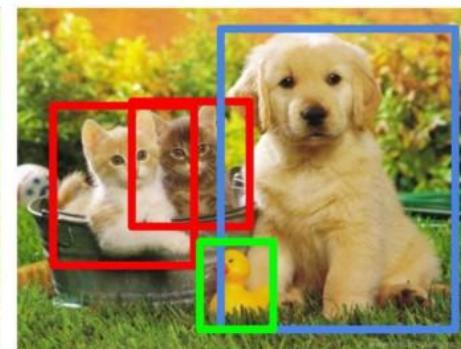
CAT

Classification  
+ Localization



CAT

Object Detection



CAT, DOG, DUCK

Instance  
Segmentation



CAT, DOG, DUCK

Single object

Multiple objects

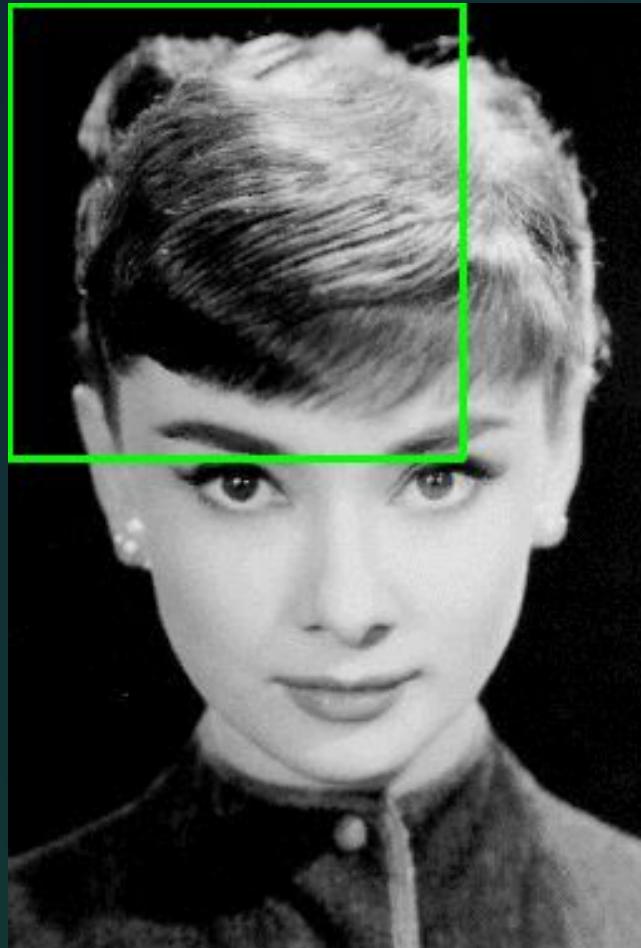


# Ejemplos de casos de uso



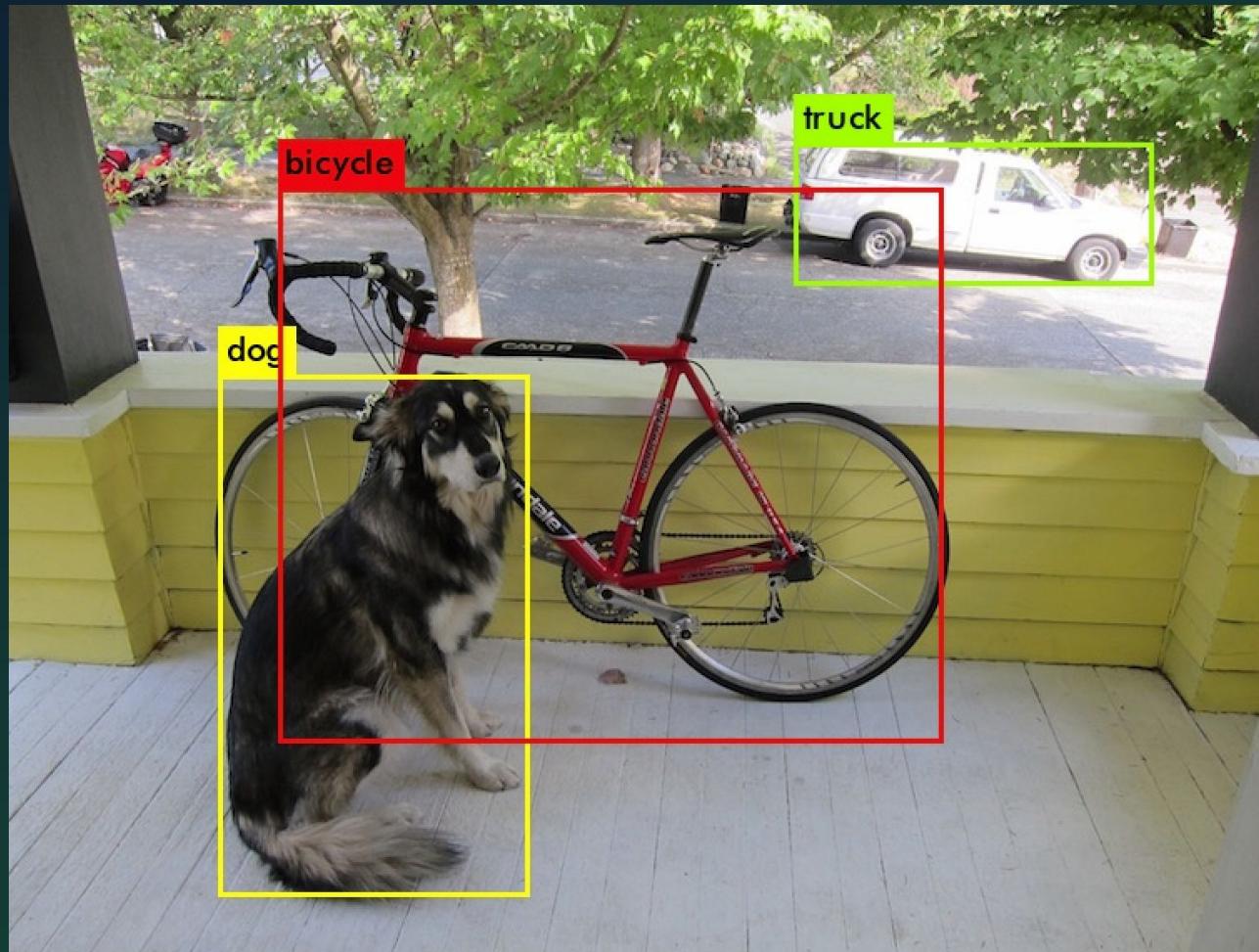


# Sliding window

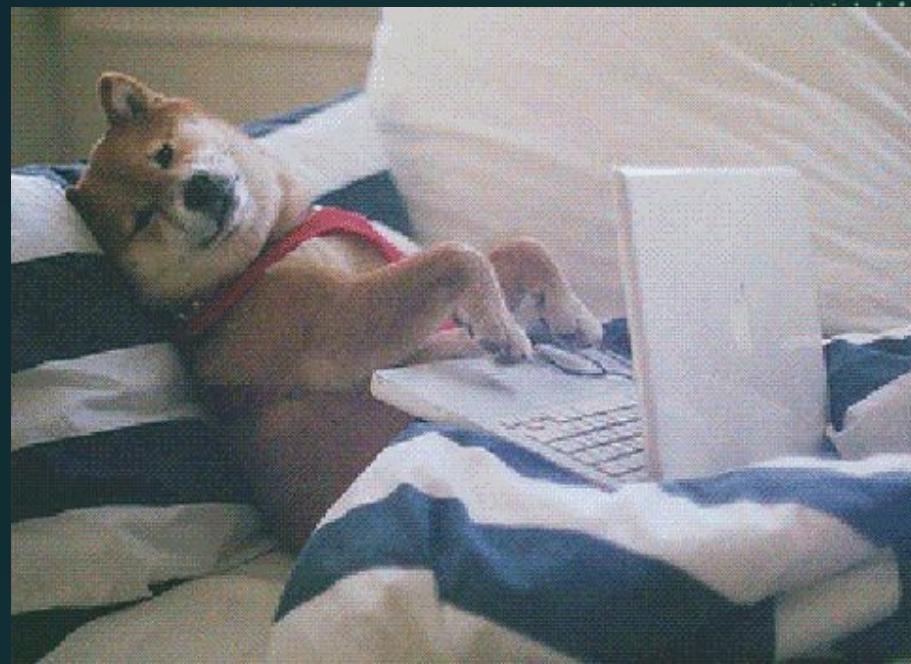




# Bounding box



co



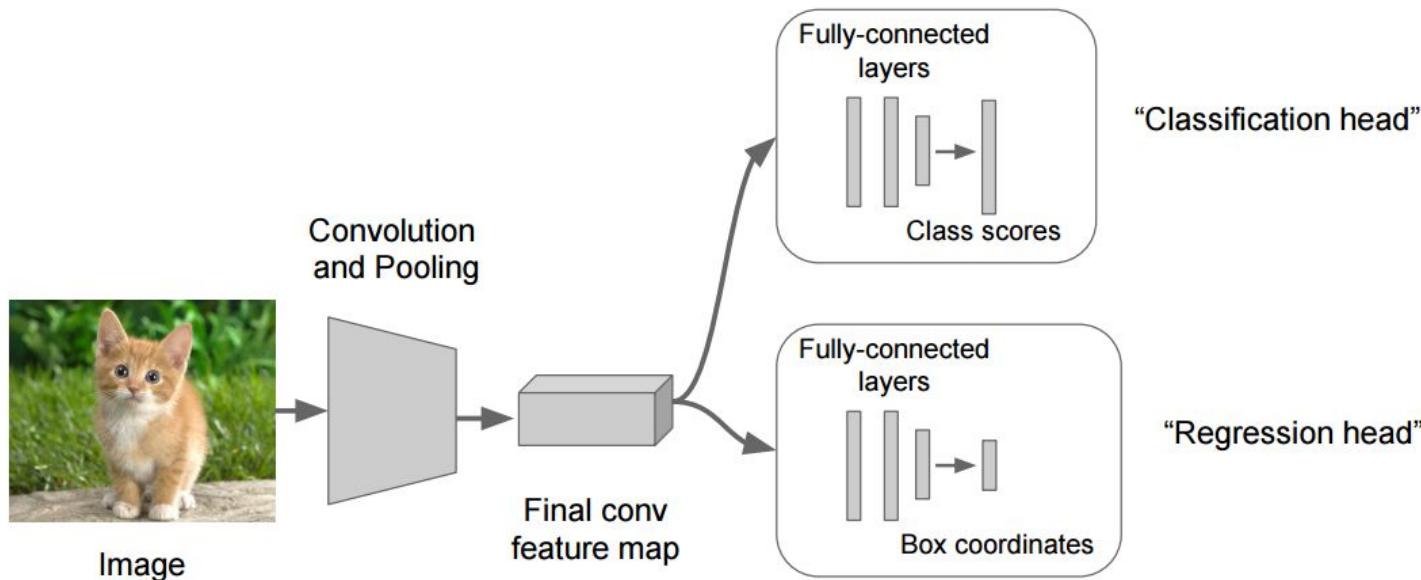
# **Introducción a object detection**

**Backbone, non-max suppression y  
métricas**



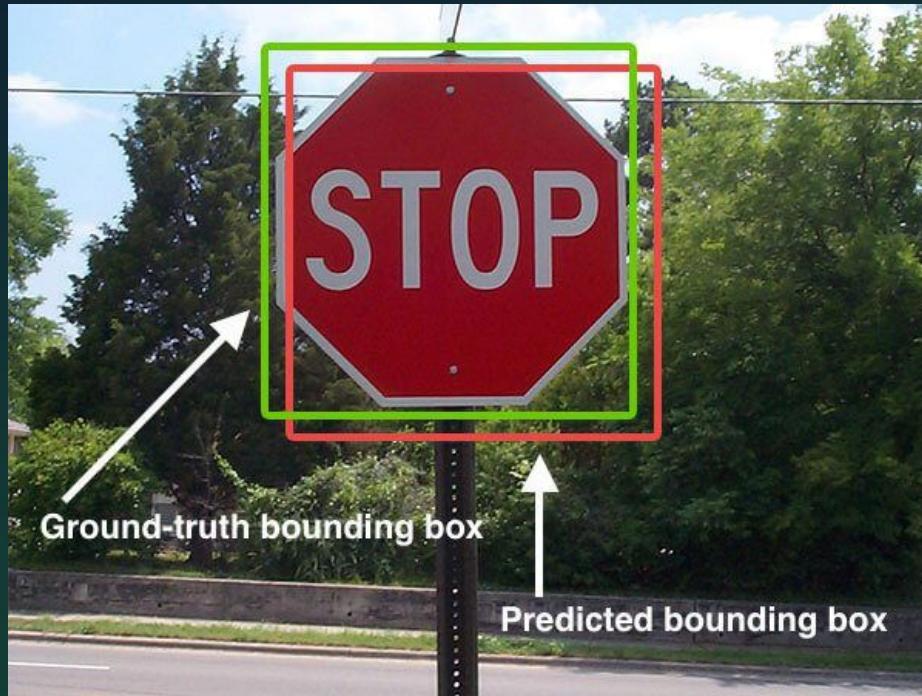
# Backbone

- Red de clasificación pre-entrenada sin últimas capas
- 2 cabezas de salida

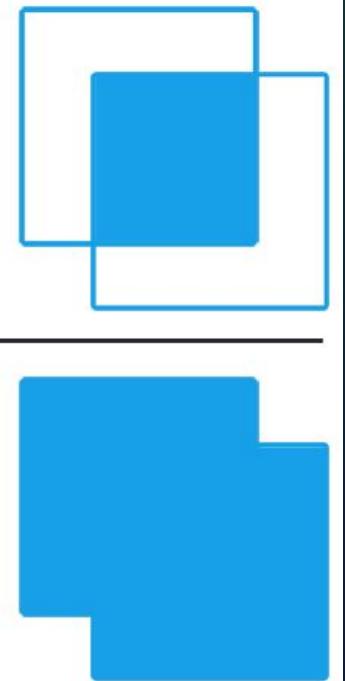




# Intersección entre uniones (IoU)



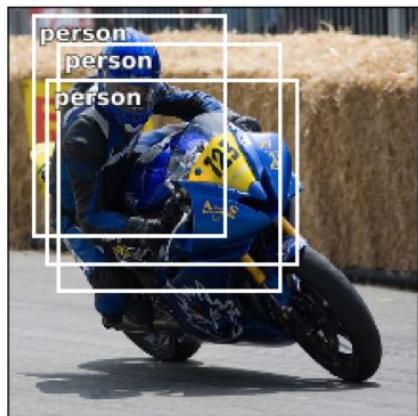
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



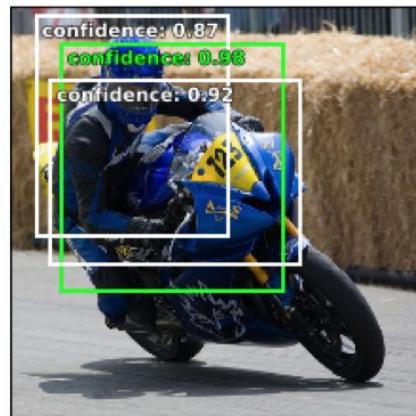


# Non-max suppression

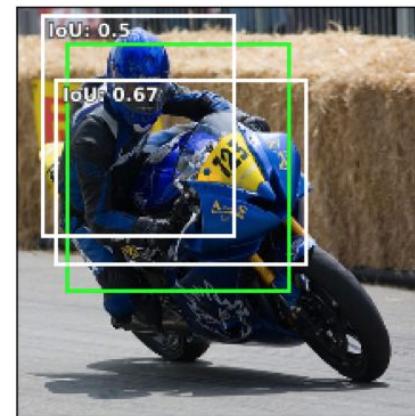
For each class...



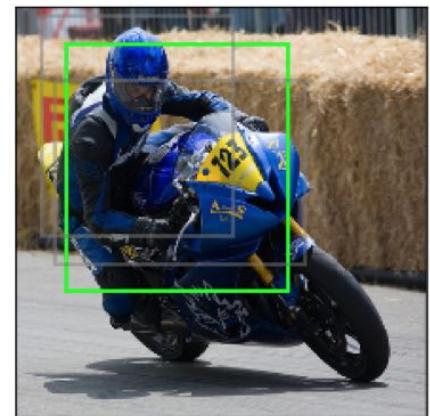
After filtering out low confidence predictions, we may still be left with **redundant detections**



Select the bounding box prediction with the **highest confidence**



Calculate the IoU between the **selected box** and all remaining predictions



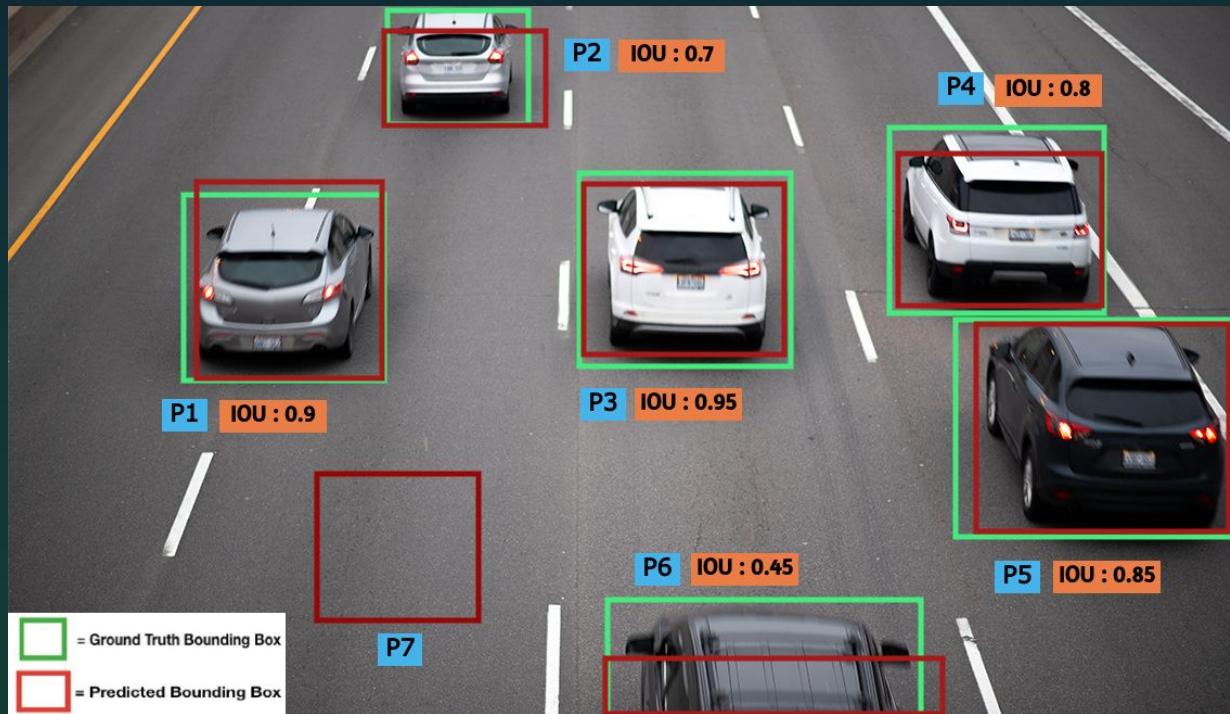
Remove any boxes which have an IoU score above some defined threshold

Repeat with next highest confidence prediction until no more boxes are being suppressed

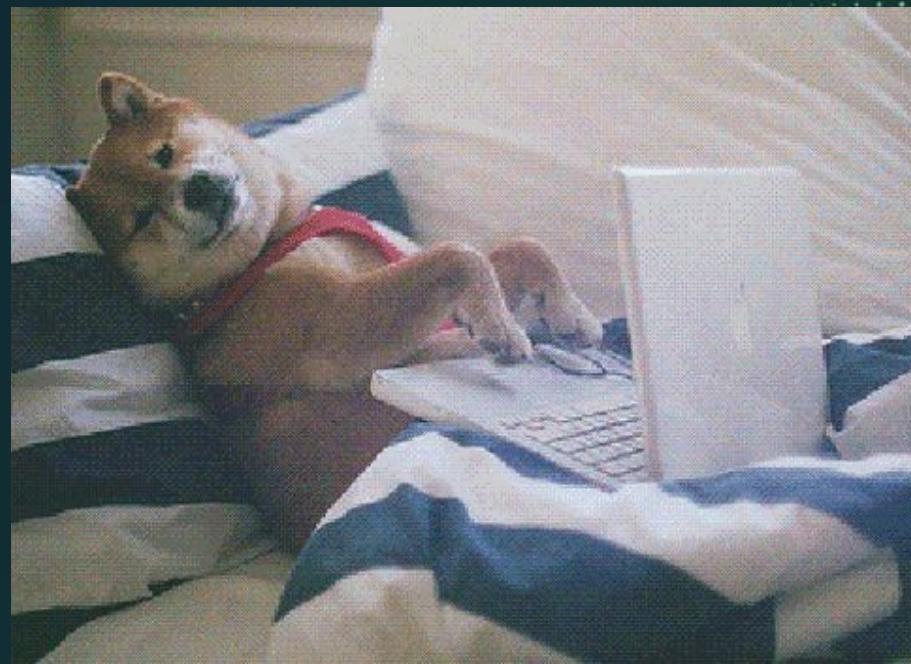


# Métricas interesantes

- Mean average precision (mAP).
- Número de frames por segundo (FPS).



co



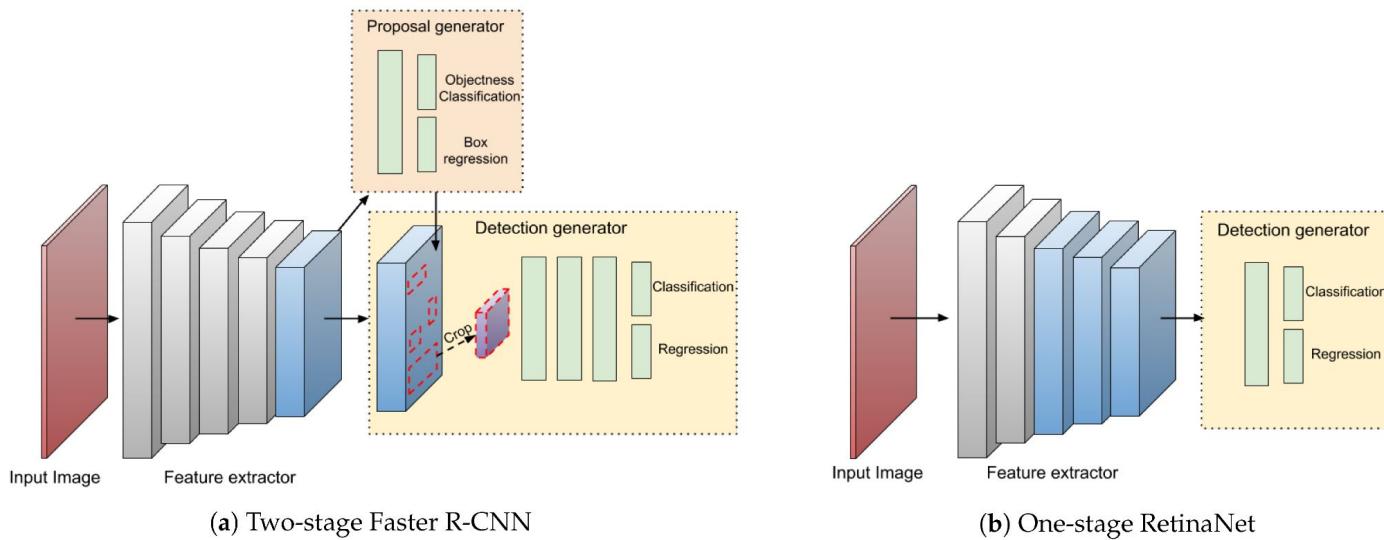
# **Tipos de arquitecturas en deteccción de objetos**

**De una etapa vs. multietapa**



# Una etapa vs. multietapa

- Una etapa: realizan la detección + localización en un solo pase hacia delante de la red.
- Multietapa: el proceso se realiza utilizando múltiples etapas.





# Multietapa

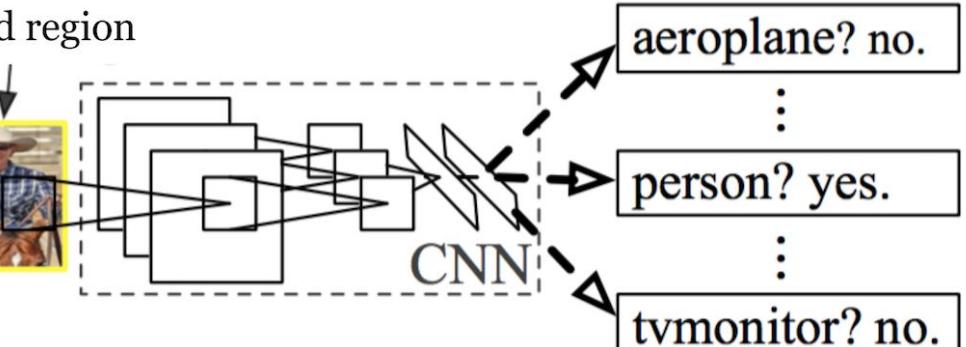


1. Input images

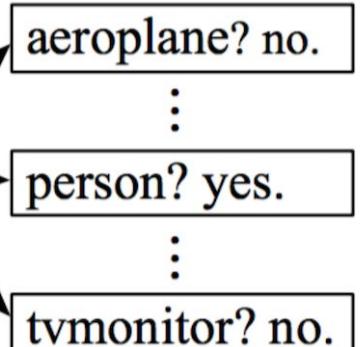


2. Extract region  
proposals (~2k)

Warped region



3. Compute CNN features



4. Classify regions



# Multietapa

- Se generan propuestas de regiones (selective search).
- Se pasan las propuestas por una CNN para analizarlas y extraer sus características.
- Finalmente se clasifican esas regiones.



# Selective search para generar las region proposals





# Una etapa

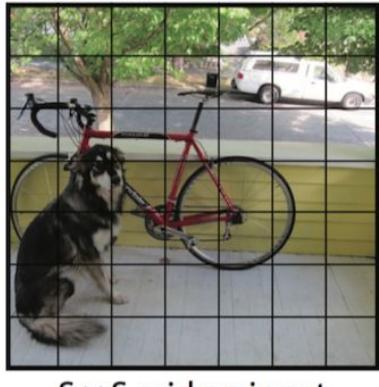
- **Extracción de características + clasificación de objetivos + regresión de posiciones en una sola etapa.**
- **Se genera un grid sobre la imagen.**
- **Anchor box.**



# Una etapa

$S \times S \times B$  bounding boxes

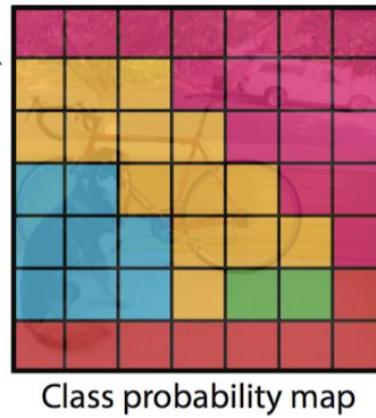
**confidence** =  $Pr(\text{object}) \times \text{IoU}(\text{pred, truth})$



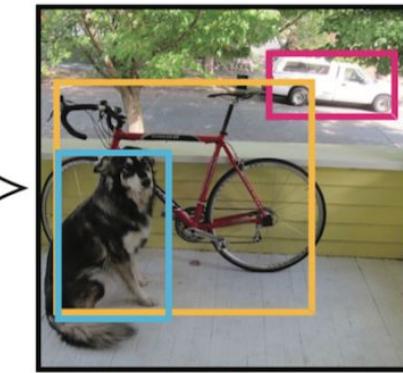
$S \times S$  grid on input



Bounding boxes + confidence



Class probability map



Final detections

$Pr(\text{Class}_i \mid \text{object})$

# Arquitecturas relevantes en detección de objetos



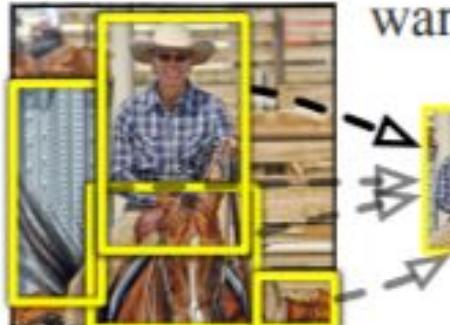
# Familia R-CNN (Fast, Faster)

- R-CNN: multietapa.

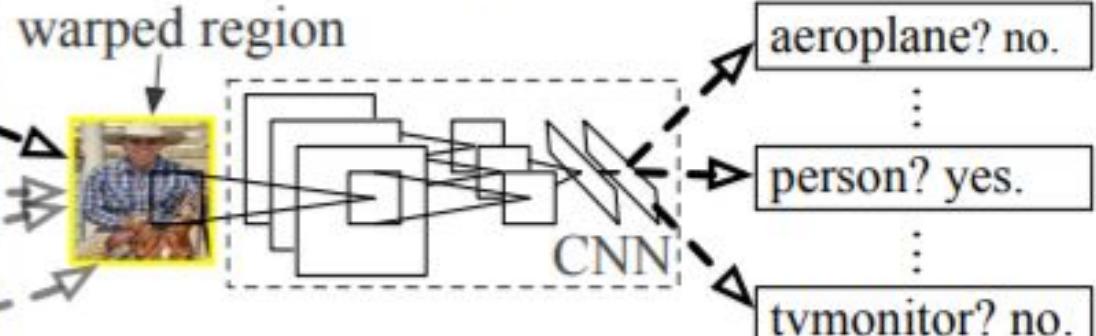
**R-CNN: *Regions with CNN features***



1. Input image



2. Extract region proposals (~2k)



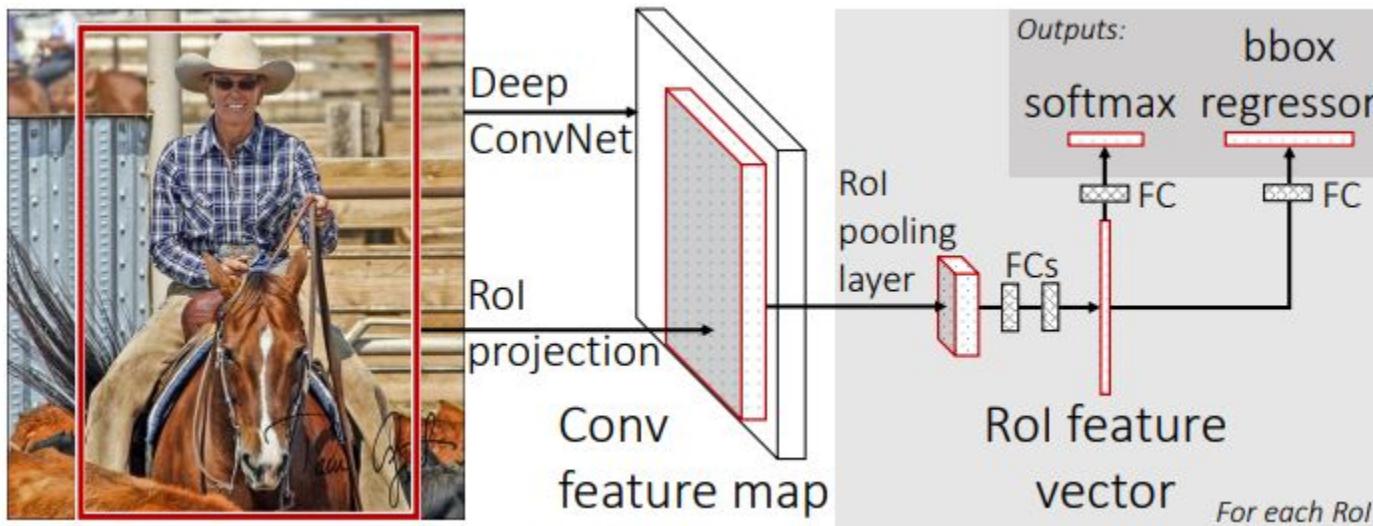
3. Compute CNN features

4. Classify regions



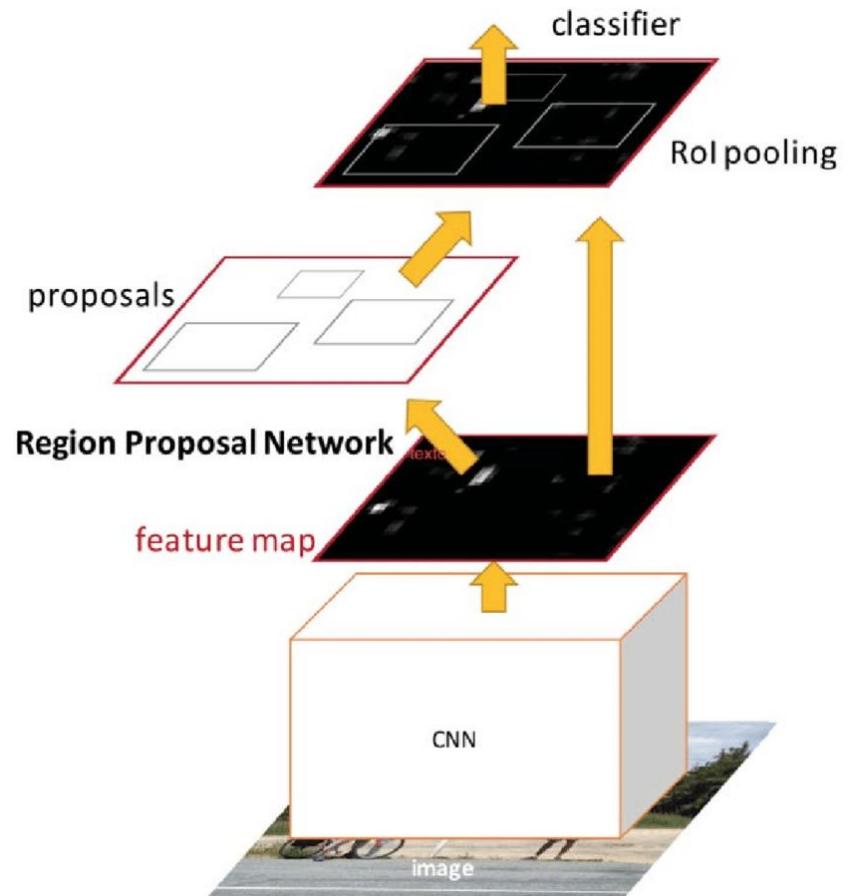
# Familia R-CNN (Fast, Faster)

- **Fast R-CNN: detecta y clasifica de forma conjunta. Mejora la velocidad.**



# Familia R-CNN (Fast, Faster)

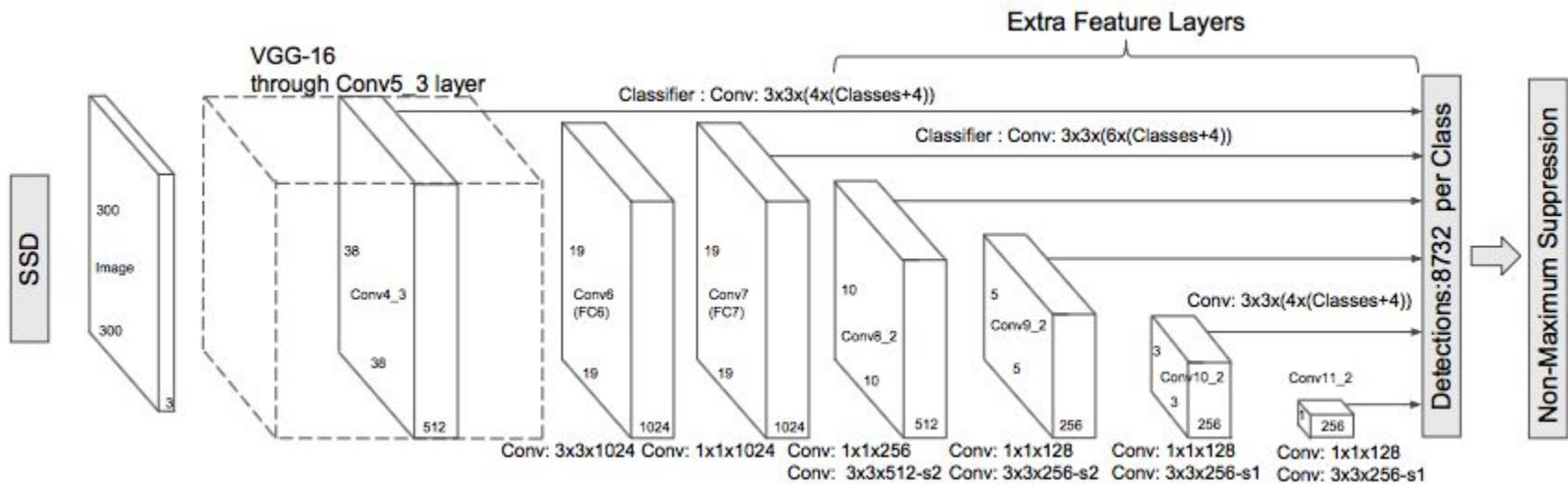
- **Faster R-CNN:** implementa Region Proposal Networks (RPN) en vez de Selective search.





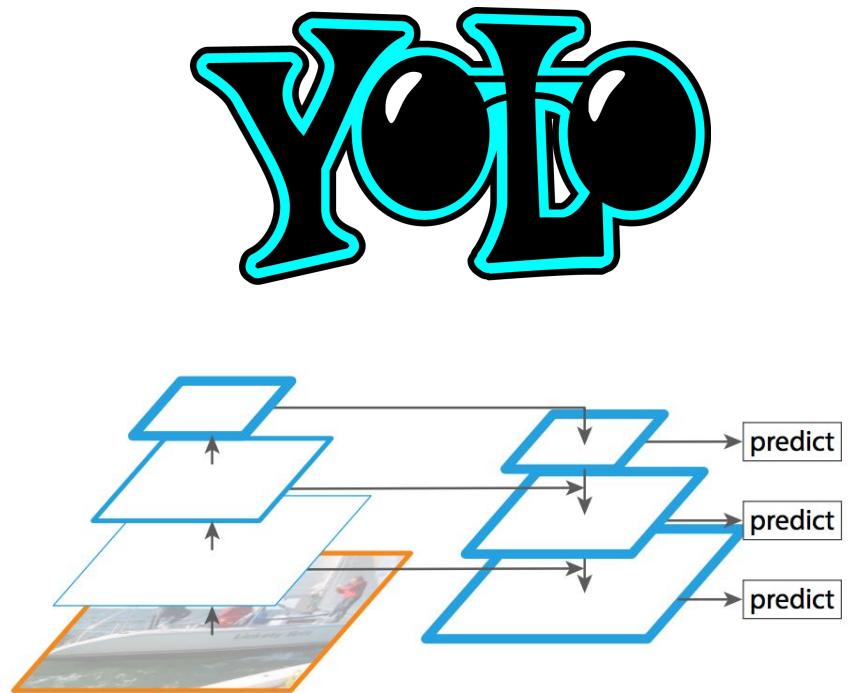
# SSD

- Arquitectura de una etapa.
- VGG-16 como backbone.



# YOLO

- Arquitectura de una etapa.
- Una de las arquitecturas más famosas.
- Mejoras incrementales en versiones sucesivas y variantes.
- Anchor boxes.
- Feature pyramid network (FPN).
- Backbone: Darknet

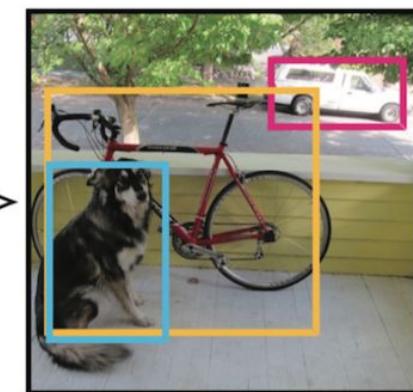
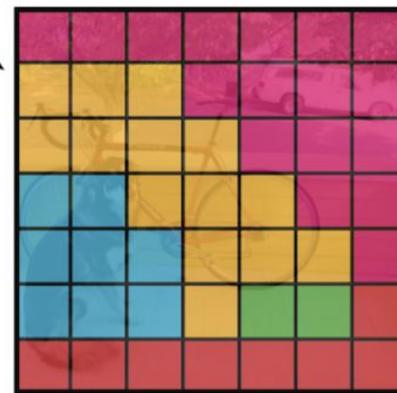




# YOLO

$S \times S \times B$  bounding boxes

**confidence** =  $Pr(\text{object}) \times \text{IoU}(\text{pred, truth})$

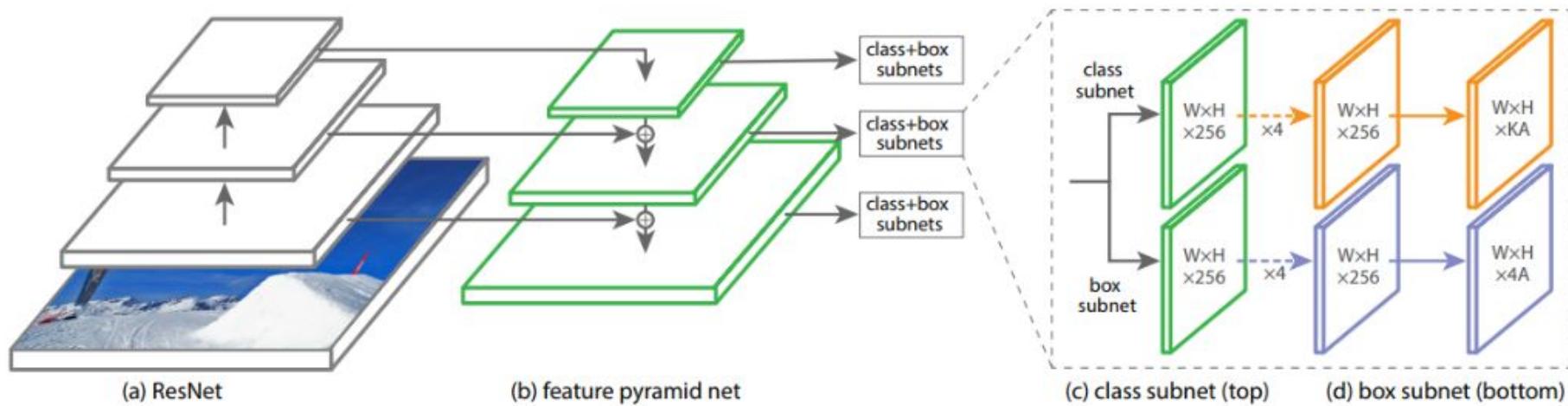


$Pr(\text{Class}_i | \text{object})$



# RetinaNet

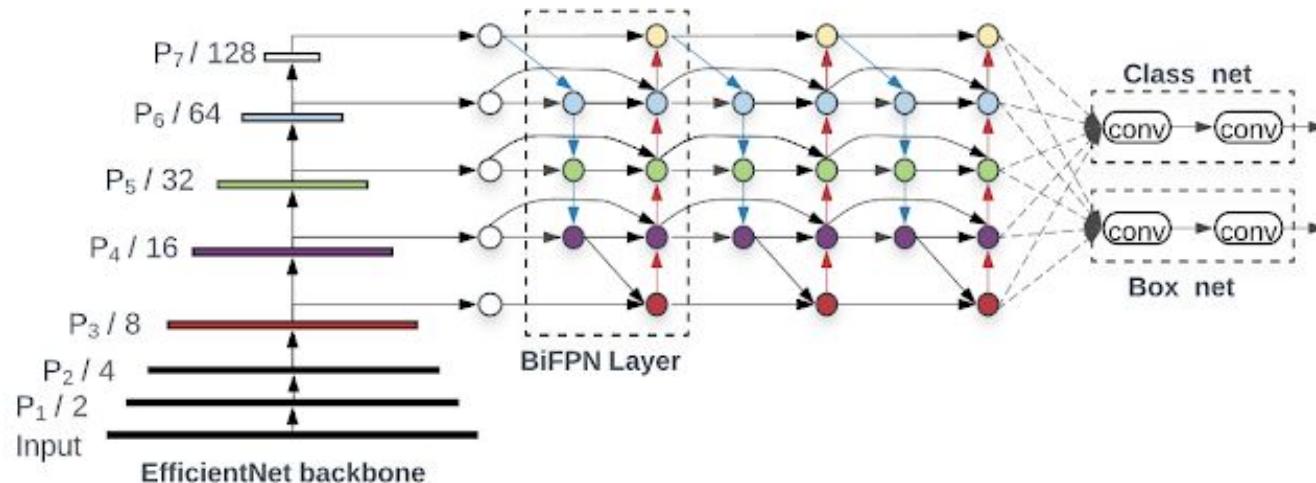
- Arquitectura de una etapa.
- Backbone ResNet.





# EfficientDet

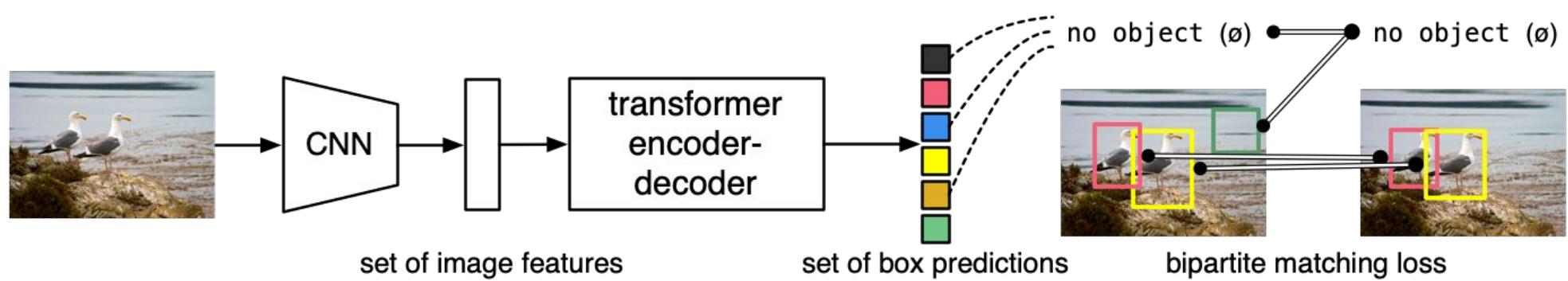
- Familia de modelos.
- Basados en EfficientNet.
- Objetivo eficiencia.
- Reduce el número de parámetros.





# DETR (Transformers)

- Nueva tendencia: basado en transformers.



# Utilizando un dataset de detección de objetos



# Datasets interesantes

- Generales:
  - COCO dataset
  - PASCAL VOC
  - ImageNet
- Especializados:
  - KITTI
  - nuScenes
  - VisDrone

# ¿Cómo descubrir datasets?

- Webs oficiales
- TensorFlow datasets
- Kaggle datasets
- Papers with code

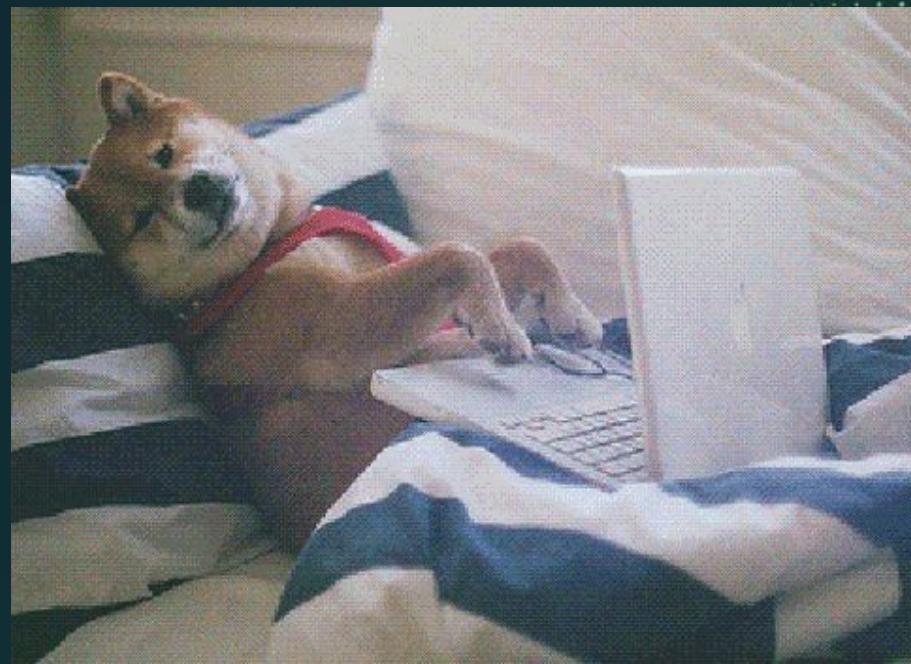


Papers With Code

kaggle<sup>TM</sup>



co

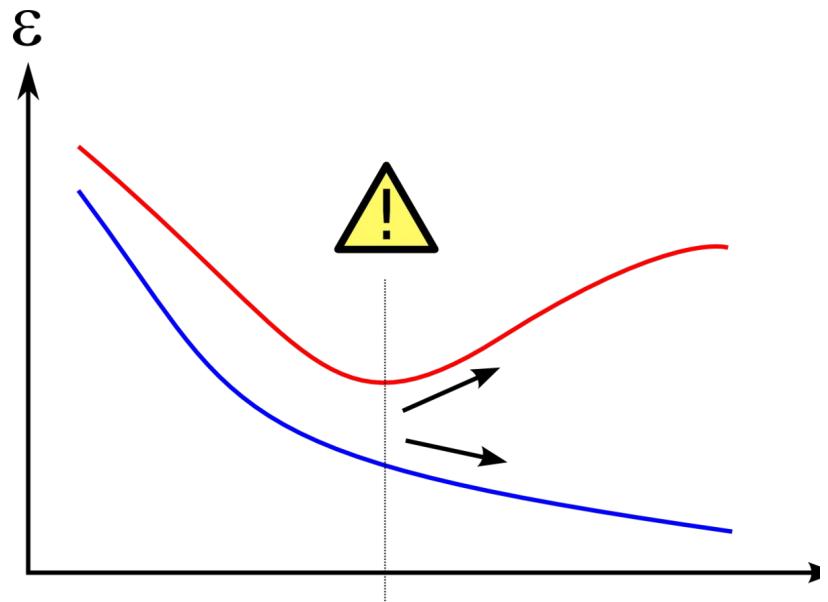
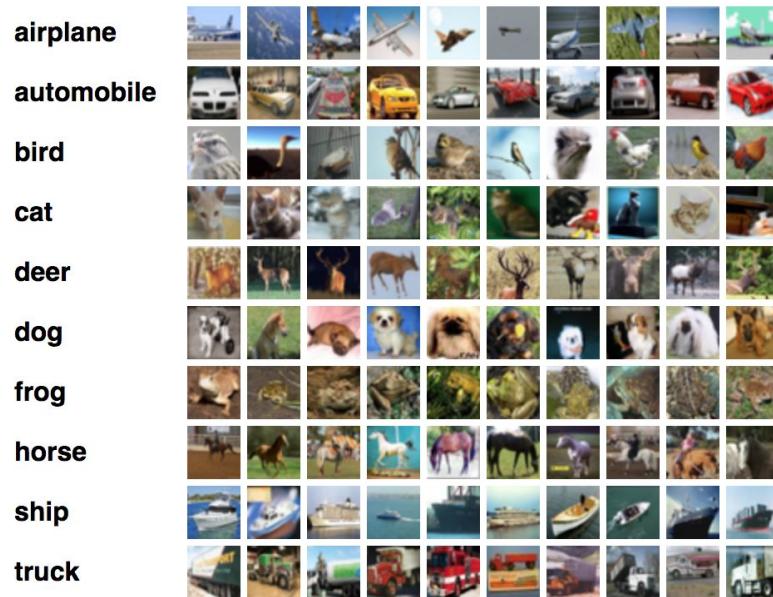


# Aumentado de datos con Alumentations



# Necesidades de sistema de deep learning

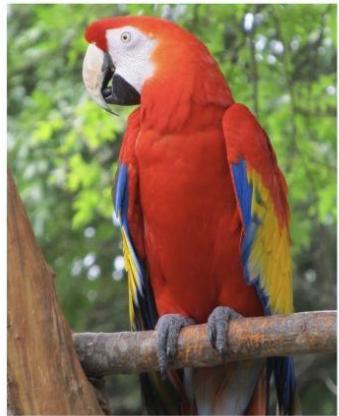
## Datos – Evitar overfitting





# Ejemplo de aumentado de datos

Original image



augmentation



Horizontal Flip



Crop



Median Blur



Contrast



Hue / Saturation / Value



Gamma





# ¿Cómo incluirlo dentro de nuestro proyecto?





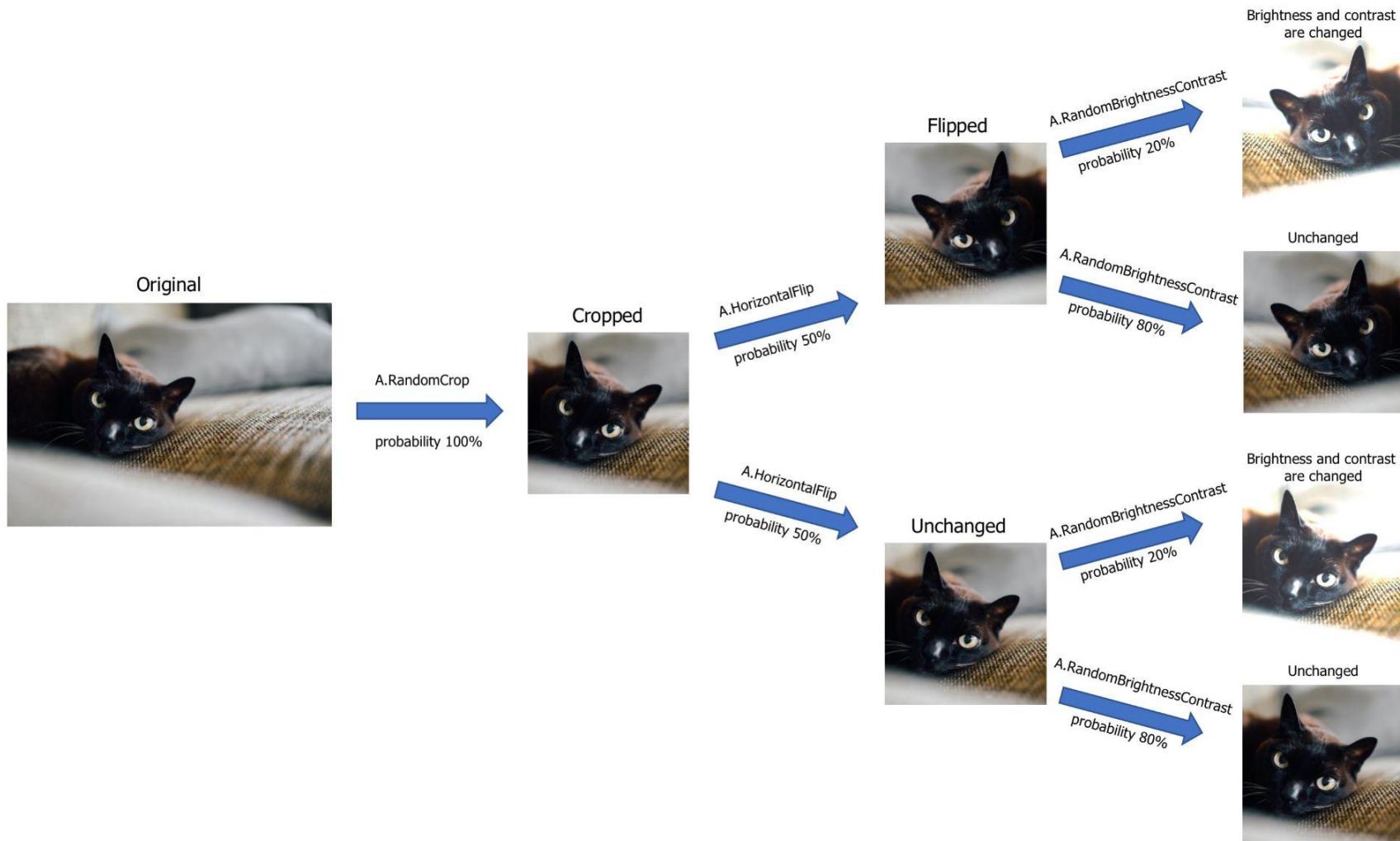
# Albumentations

- Librería open source.
- PyTorch, TensorFlow.
- Clasificación, detección, segmentación.
- Integración muy sencilla.
- Pipelines.

A large, bold, red letter 'A' with a black outline, centered on the right side of the slide.

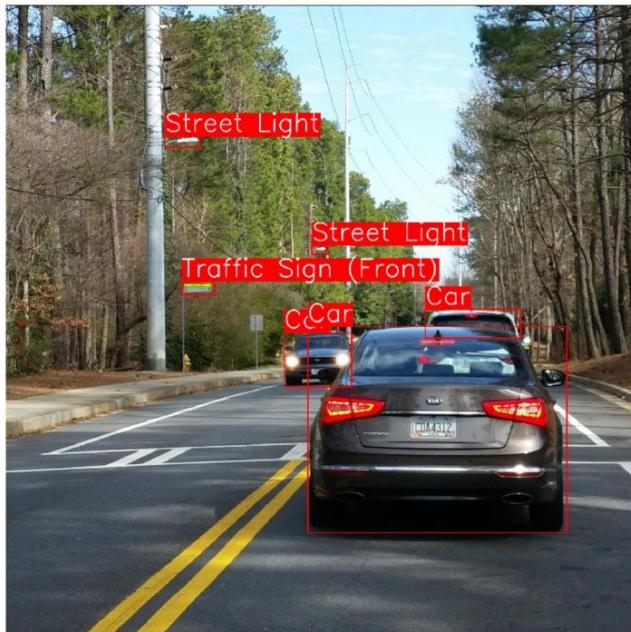


# Pipeline



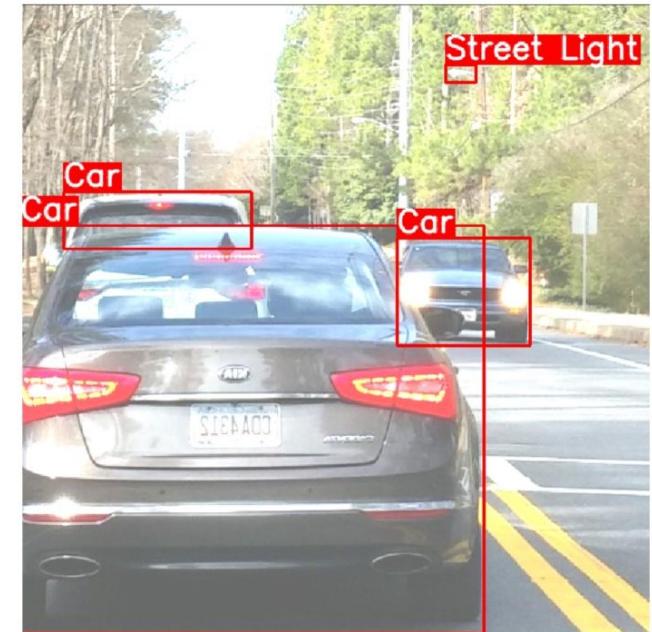


# Soporte para object detection



Brightness adjustment

pixel-level augmentations

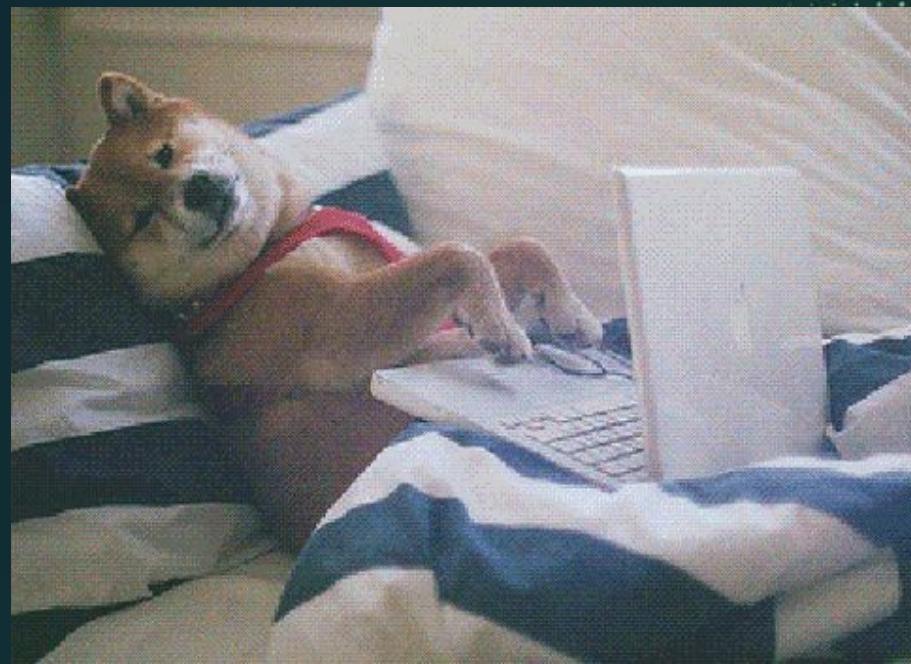


Mirroring

Cropping a part of the image

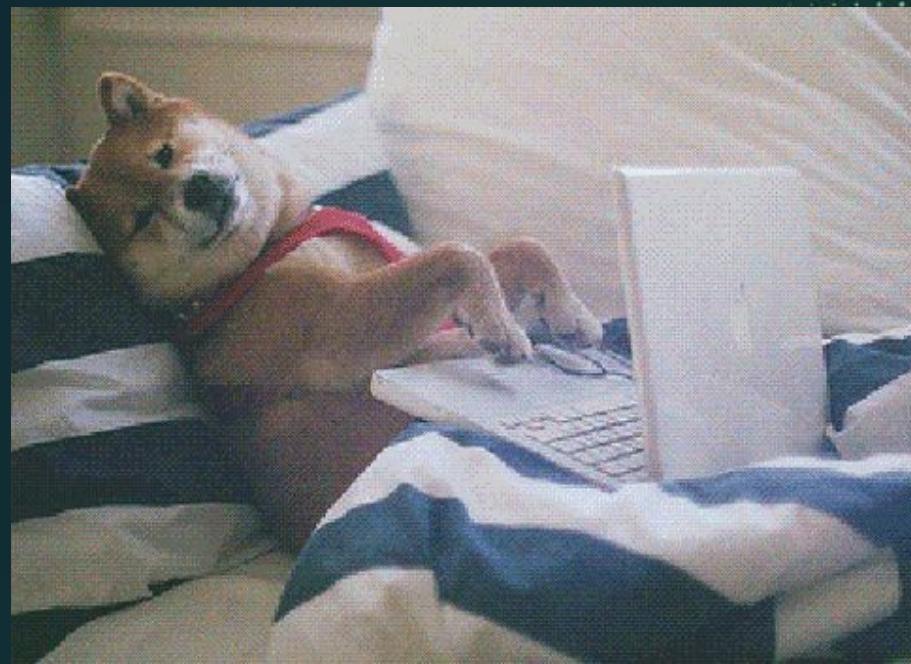
spatial-level augmentations

co



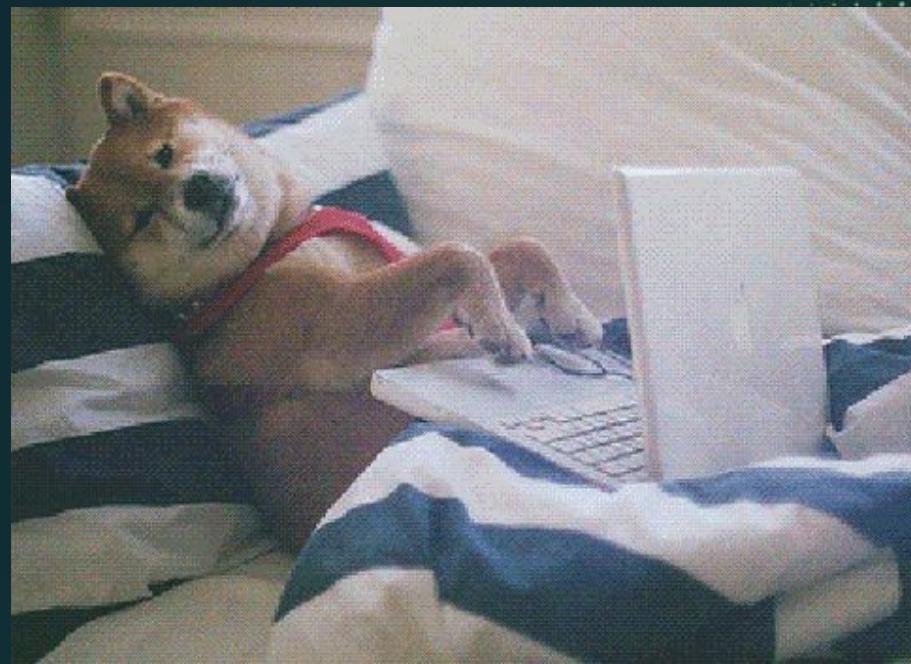
# **Utilizando un modelo de object detection pre-entrenado**

co



# Fine-tuning en detección de objetos

co

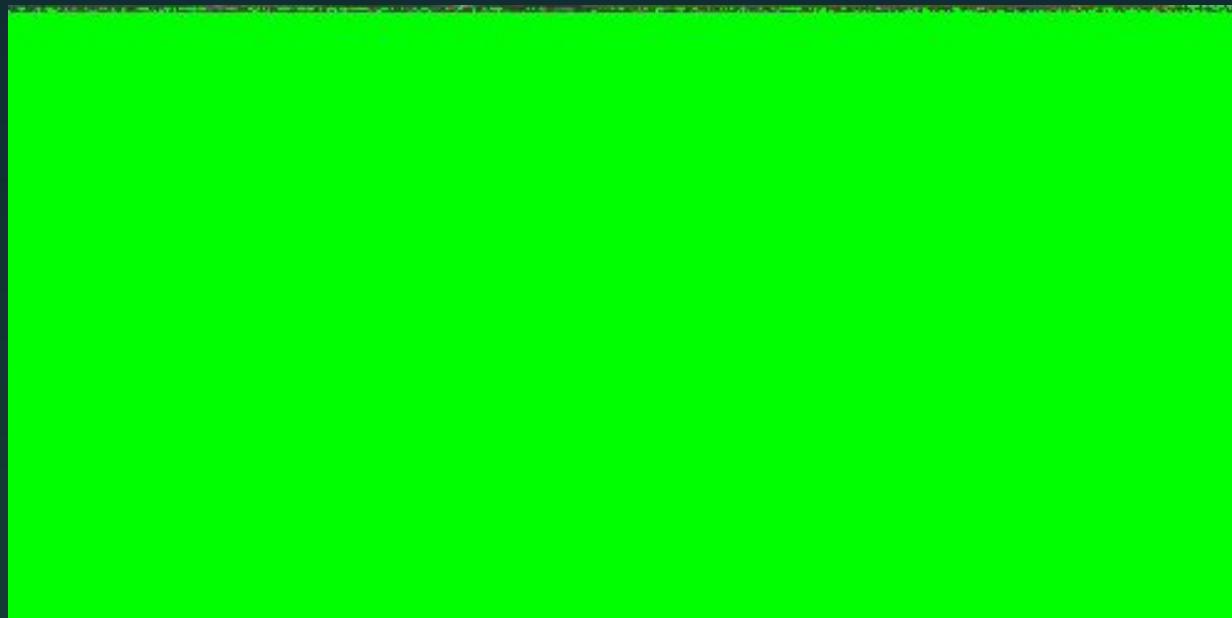


# Introduciendo la segmentación de objetos



# Segmentación

- Una de las tareas clave en CV.
- Entendimiento de la escena.





# Segmentación

- Un paso más allá:

**Clasificación → Localización/Detección → Pixel a pixel**





# Ejemplos de casos de uso





# Métricas

- Precisión a nivel de píxeles
- IoU
- Coeficiente de Dice (F1 score)

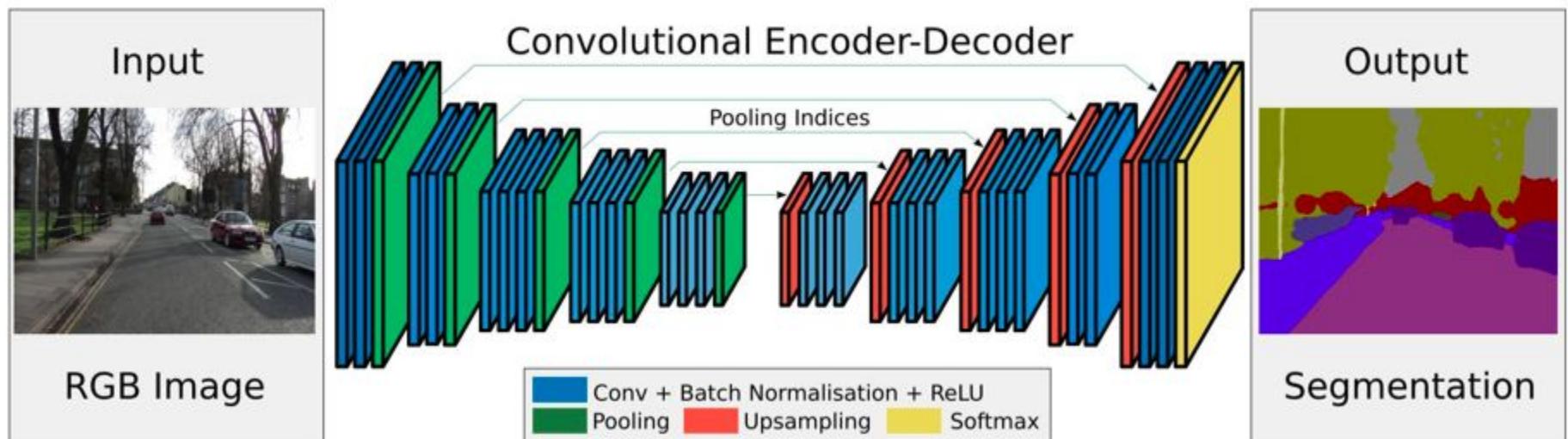


# Arquitectura encoder-decoder

- Dos partes diferenciadas: encoder y decoder.
- Encoder genera características discriminativas en resolución baja.
- Decoder proyecta características en el espacio de píxeles, aumentando la resolución.



# Arquitectura encoder-decoder



# Tipos de segmentación



# Segmentación

## Other Computer Vision Tasks

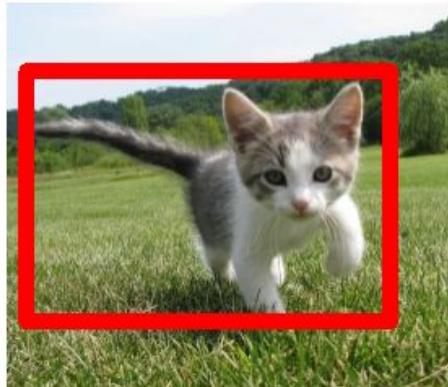
Semantic  
Segmentation



GRASS, CAT,  
TREE, SKY

No objects, just pixels

Classification  
+ Localization



CAT

Single Object

Object  
Detection



DOG, DOG, CAT

Multiple Object

Instance  
Segmentation



DOG, DOG, CAT

This image is CC0 public domain



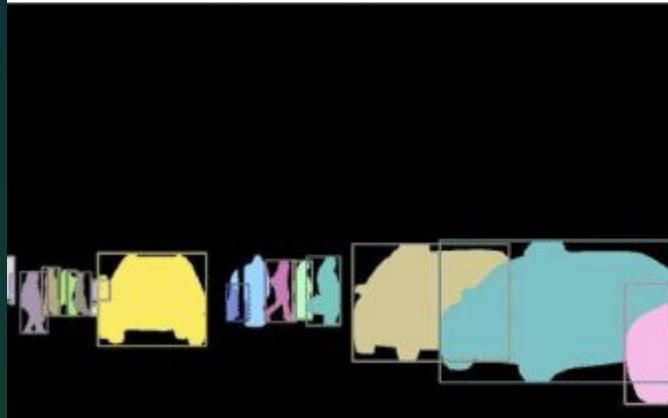
# Segmentación



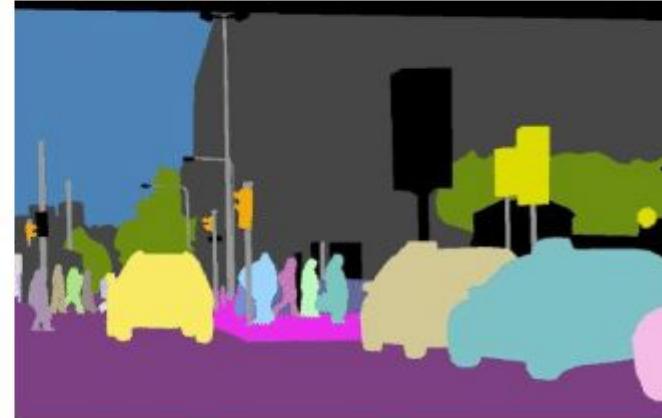
(a) Image



(b) Semantic Segmentation



(c) Instance Segmentation



(d) Panoptic Segmentation

# Arquitecturas relevantes en segmentación

# Arquitecturas

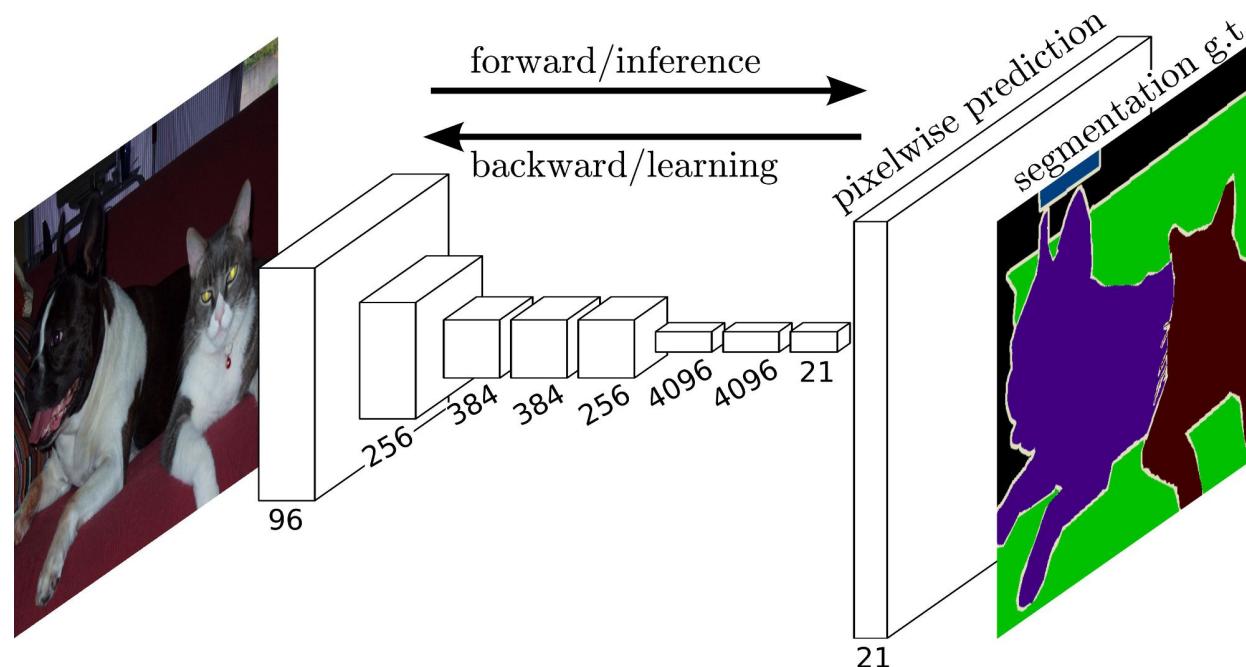
- Fully Convolutional Network
- U-Net
- DeepLab





# Fully Convolutional Network

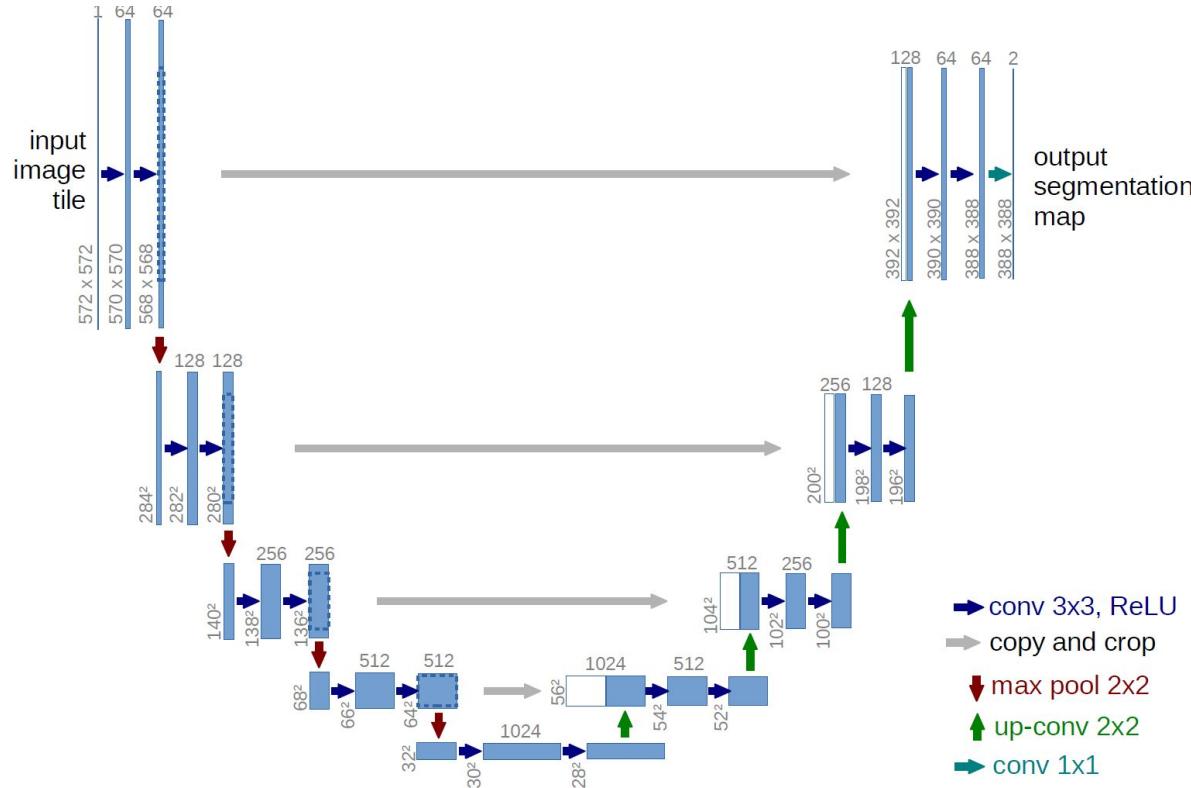
- Encoder AlexNet
- Skip connections





# U-Net

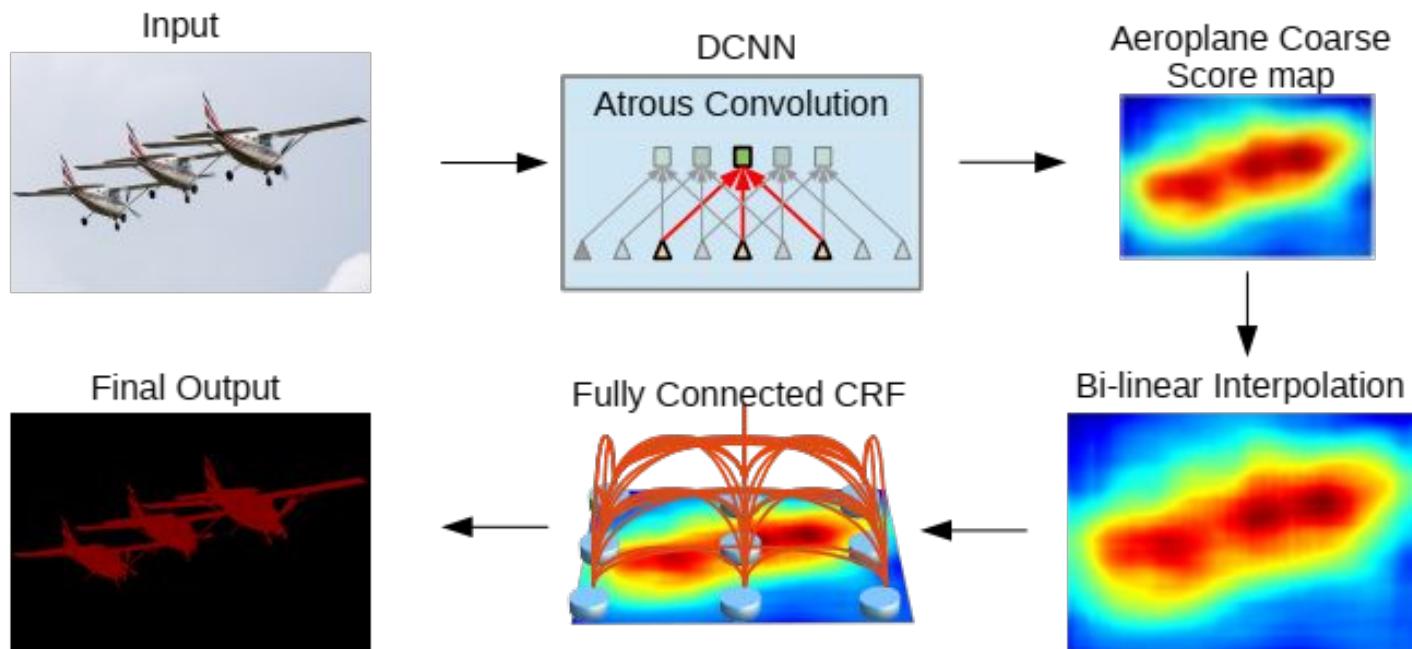
- Camino de contracción para contexto.
- Camino de expansión simétrico para localización.





# DeepLab

- Atrous convolution
- Conditional Random Fields (CRF)



**¿Cómo es un dataset  
de segmentación?**

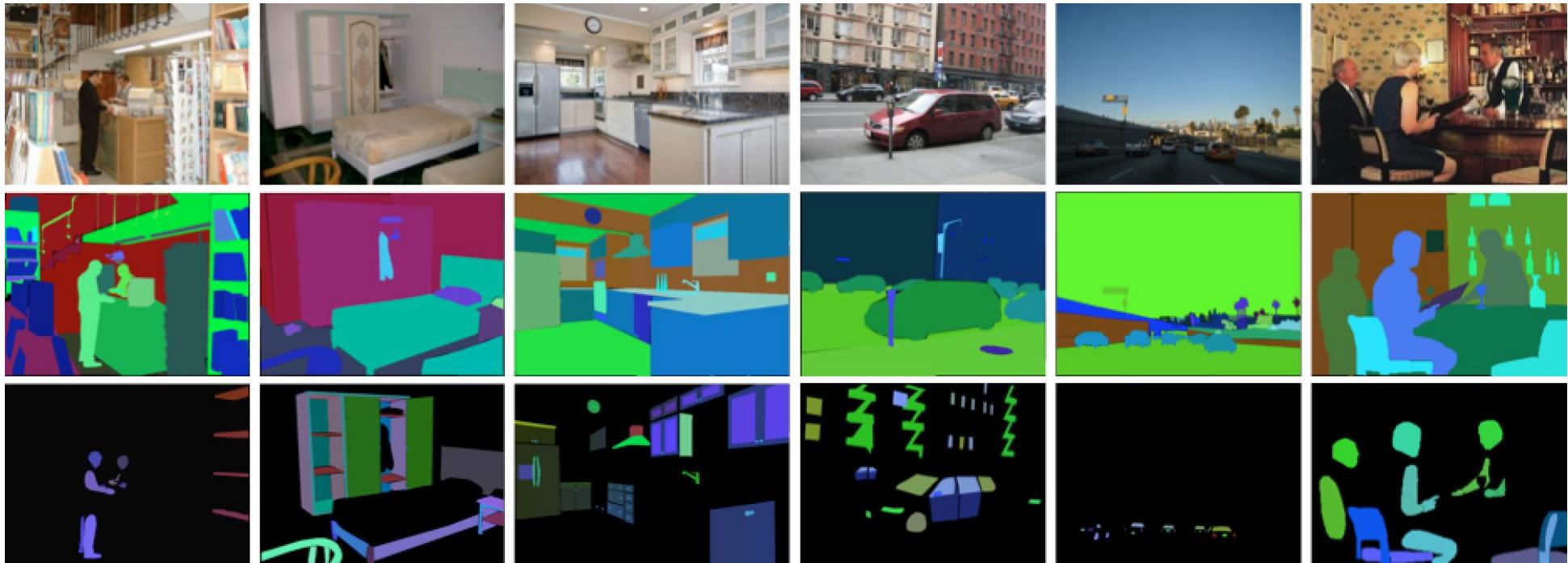


# Datasets interesantes

- COCO dataset
- PASCAL VOC
- CityScapes
- KITTI
- ADE20K
- DAVIS (Densely Annotated Video Segmentation)
- BDD100K



# Ejemplo: ADE20K dataset



# Cómo descubrir datasets

- Webs oficiales
- TensorFlow datasets
- Kaggle datasets
- Papers with code

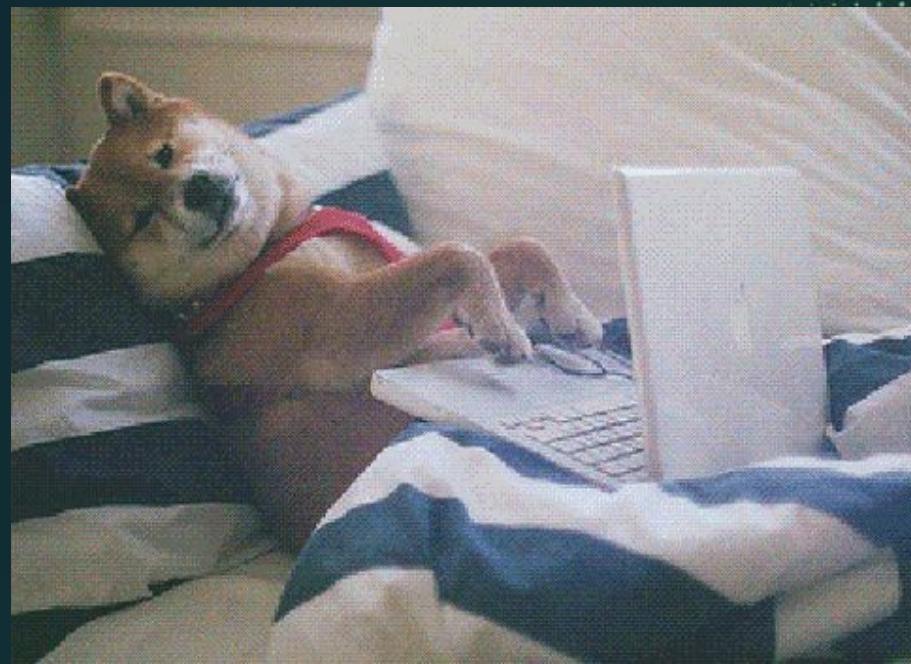


Papers With Code

kaggle™

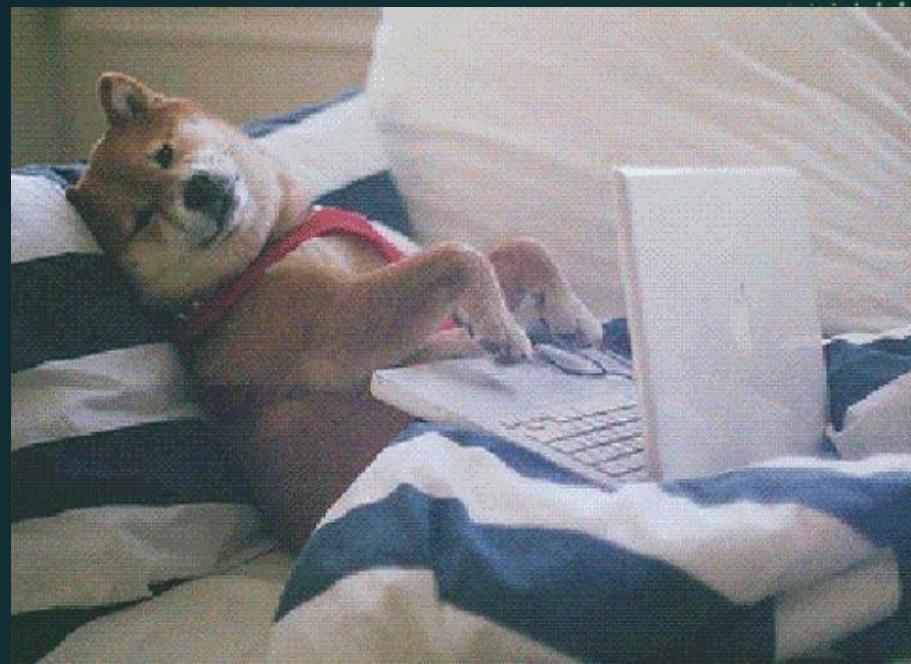


co



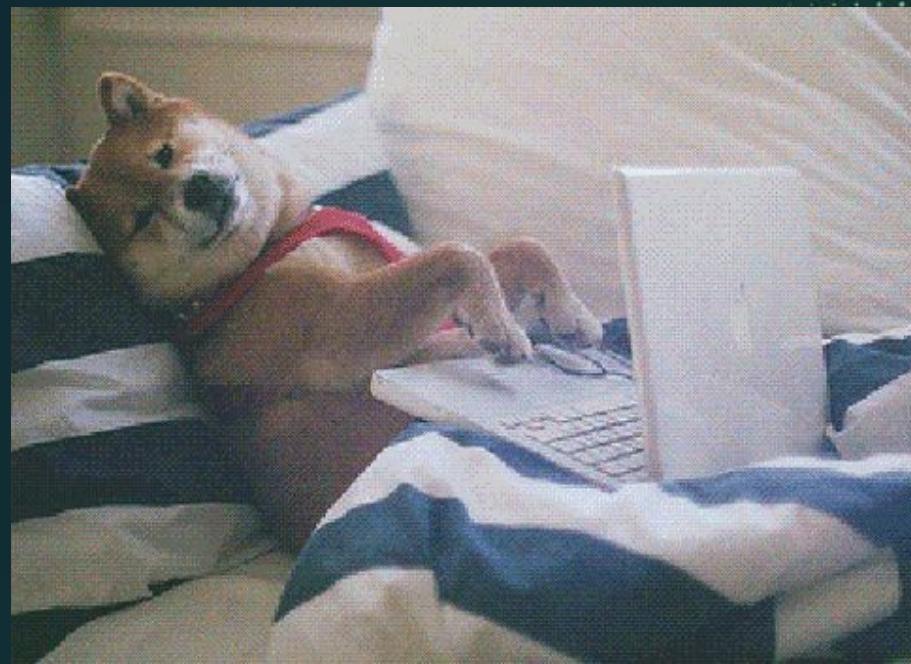
# Utilizando un dataset de segmentación

co



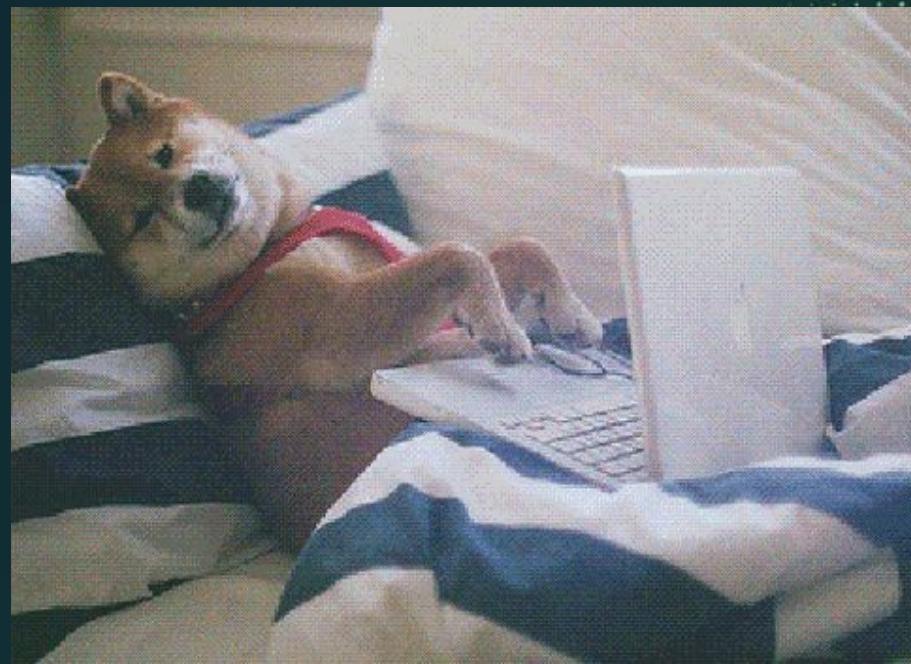
# Visualización de nuestro dataset de segmentación

co



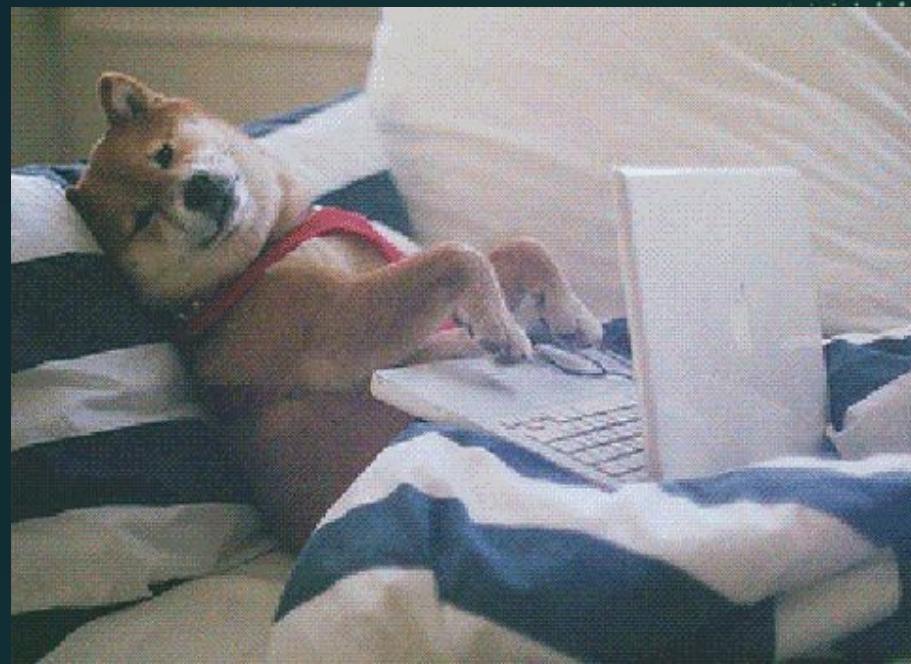
# **Creando red neuronal U-Net para segmentación**

co



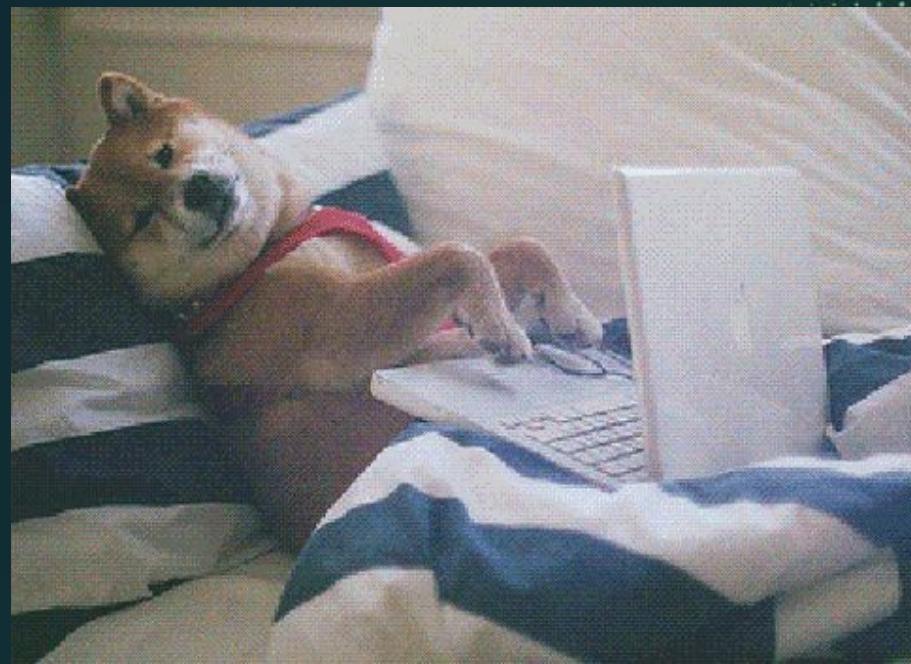
# **Entrenando y estudiando una red de segmentación**

co



# Generando predicciones con modelo de object segmentation

co



# **Conociendo el estado de la cuestión en CV**



# Herramientas interesantes

- Papers with code



- Kaggle

kaggle

- Arxiv

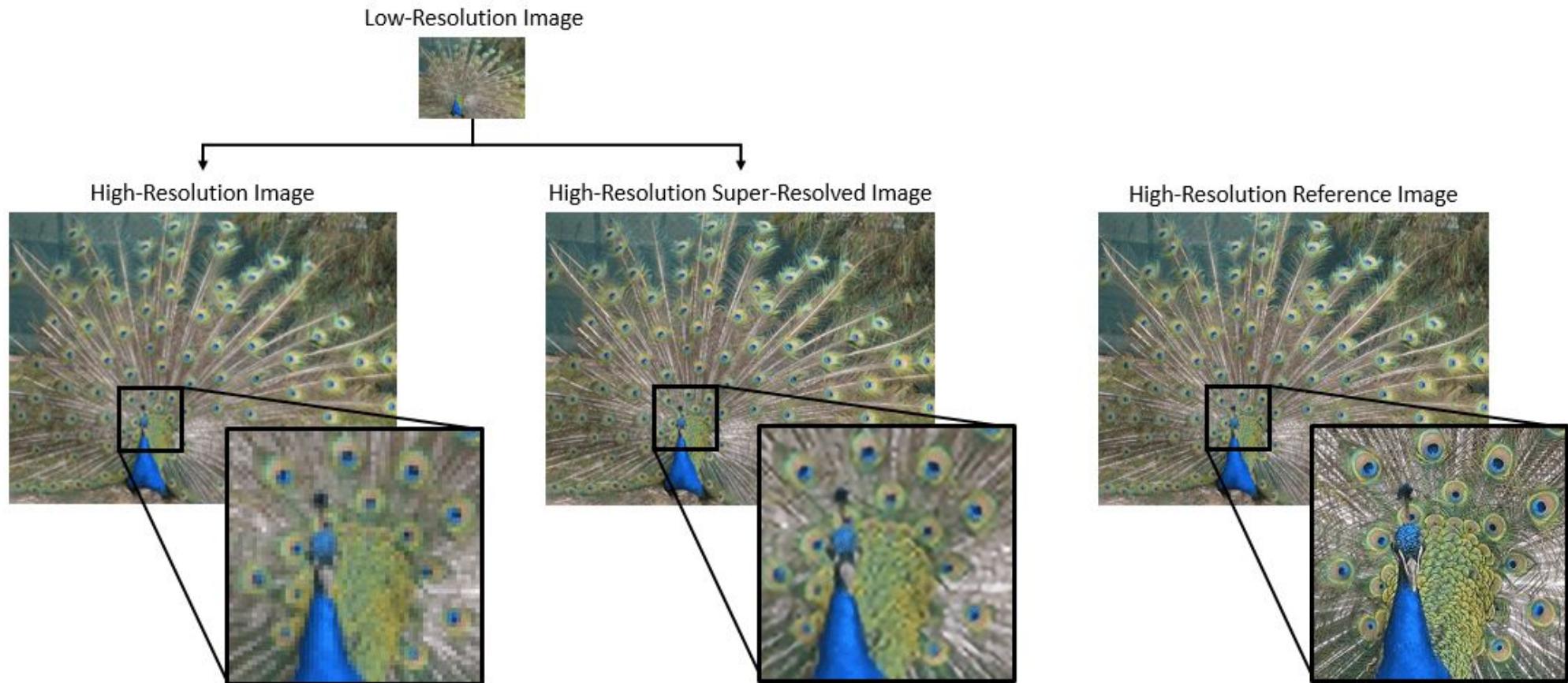


- Hugging Face





# Otras áreas interesantes





# Otras áreas interesantes

