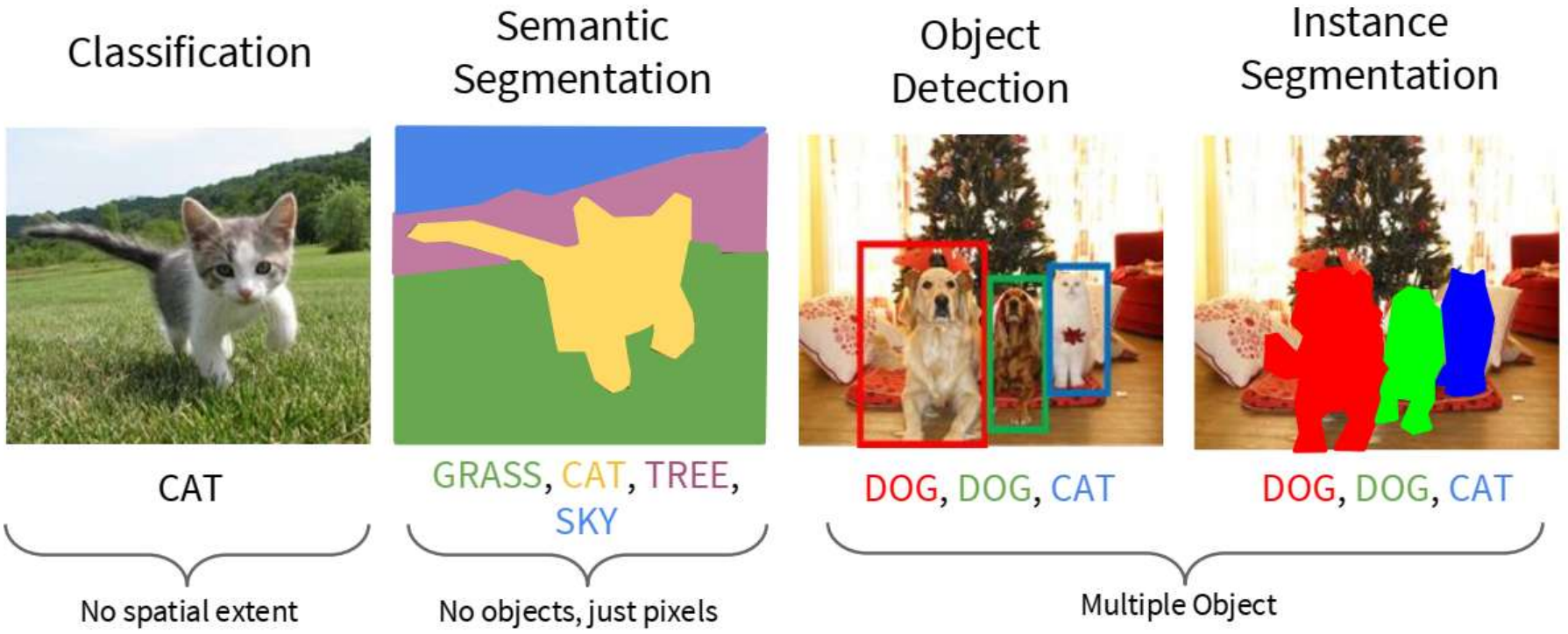
Visión Computacional

Ivan Sipiran

Tareas de Computer Vision



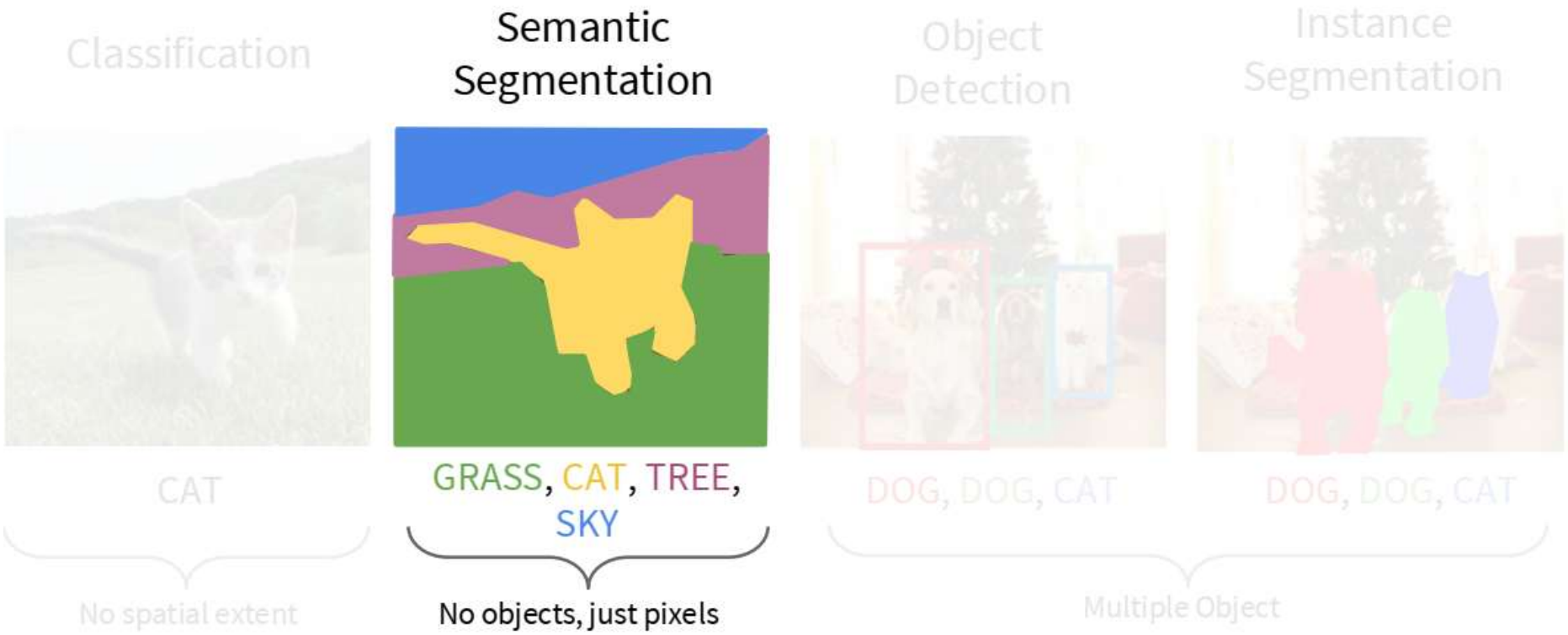
Clasificación

Tarea fundamental en CV

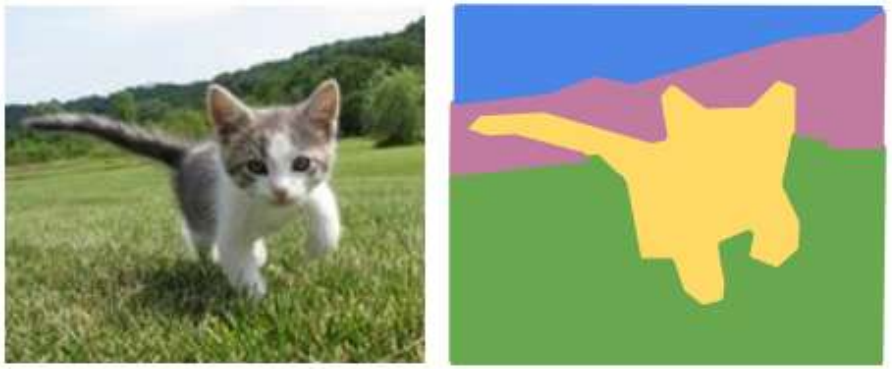
cat

De entre un conjunto posible de etiquetas

Segmentación semántica

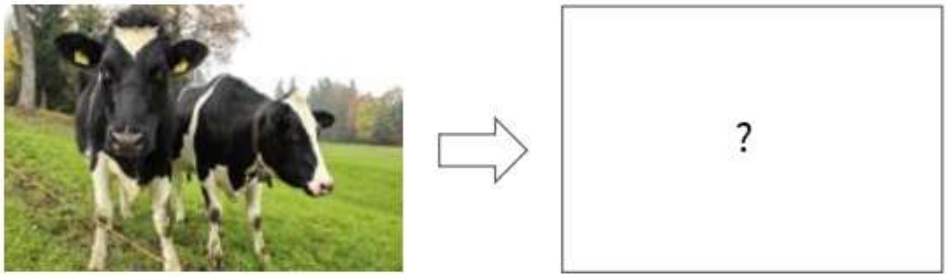


Segmentación semántica: Problema



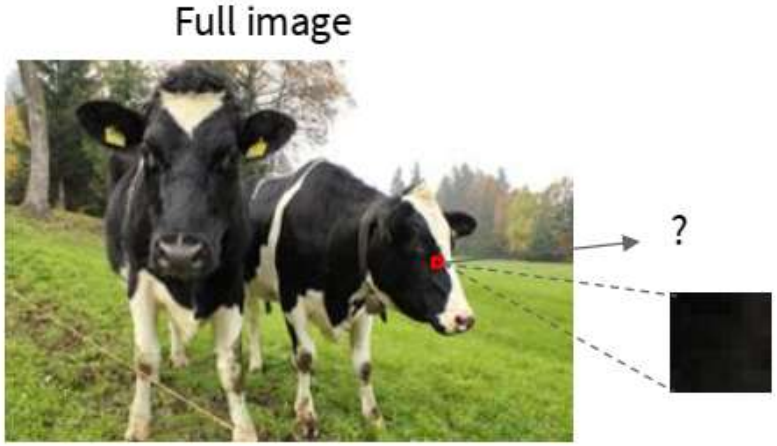
GRASS, CAT, TREE,
SKY, ...

Para cada imagen del dataset, cada píxel
tiene una etiqueta con una categoría semántica



Durante inferencia, clasificar cada píxel
De la imagen

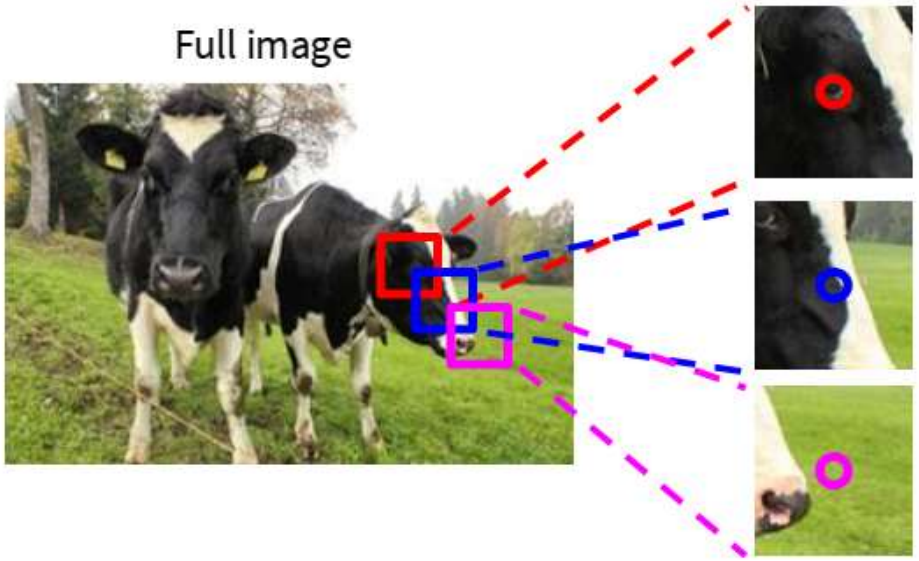
Segmentación semántica: Idea básica



Imposible clasificar sin un contexto

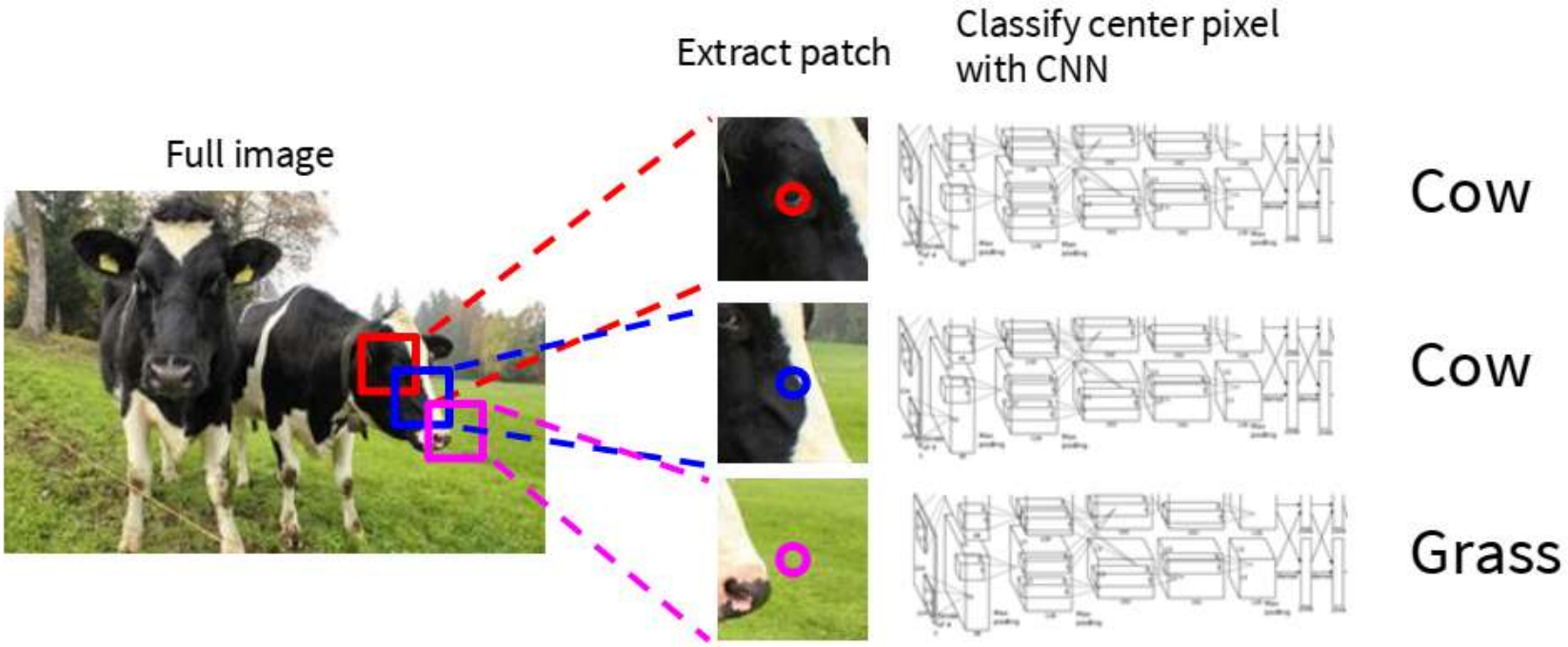
Cómo podemos incorporar contexto?

Segmentación semántica: Idea básica

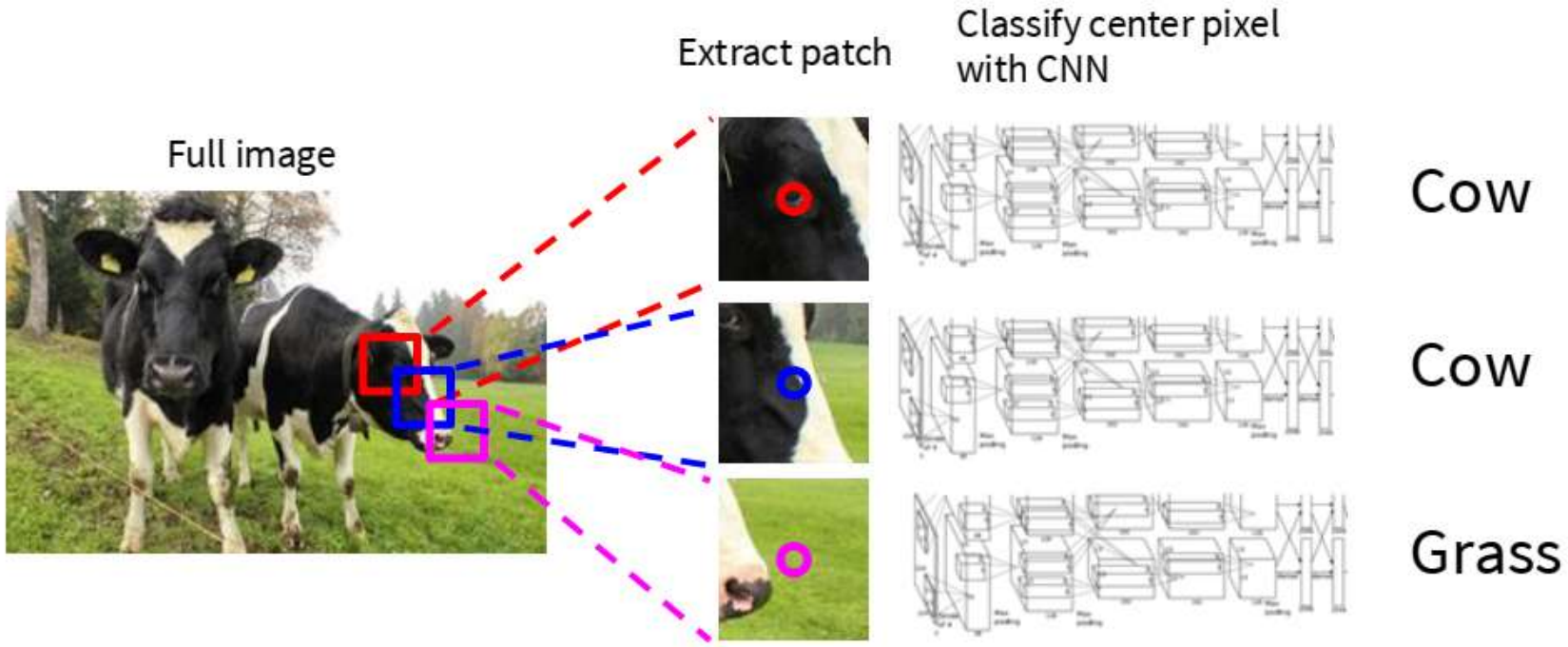


Sliding window

Segmentación semántica: Idea básica



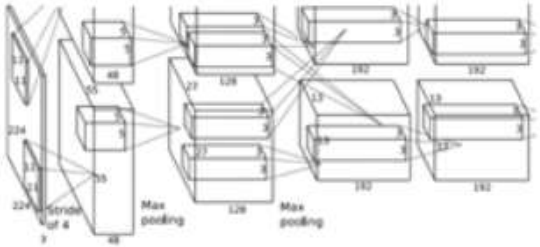
Segmentación semántica: Idea básica



Problema: muy ineficiente!
No se reusan features compartidas entre parches

Segmentación semántica: Convolución

Full image

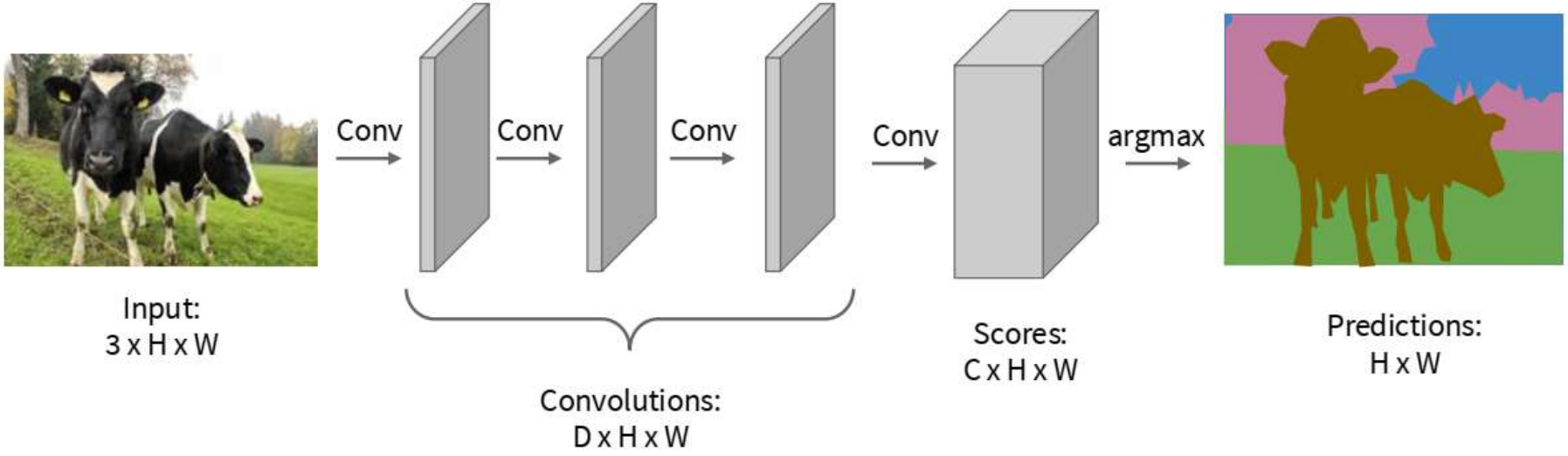


Idea intuitiva: codificar la imagen entera con ConvNet y hacer segmentación semántica directamente

Problema: arquitecturas de clasificación reducen las características espaciales de acuerdo a la profundidad de la red.
Segmentación semántica requiere que el tamaño de la salida sea igual que la entrada.

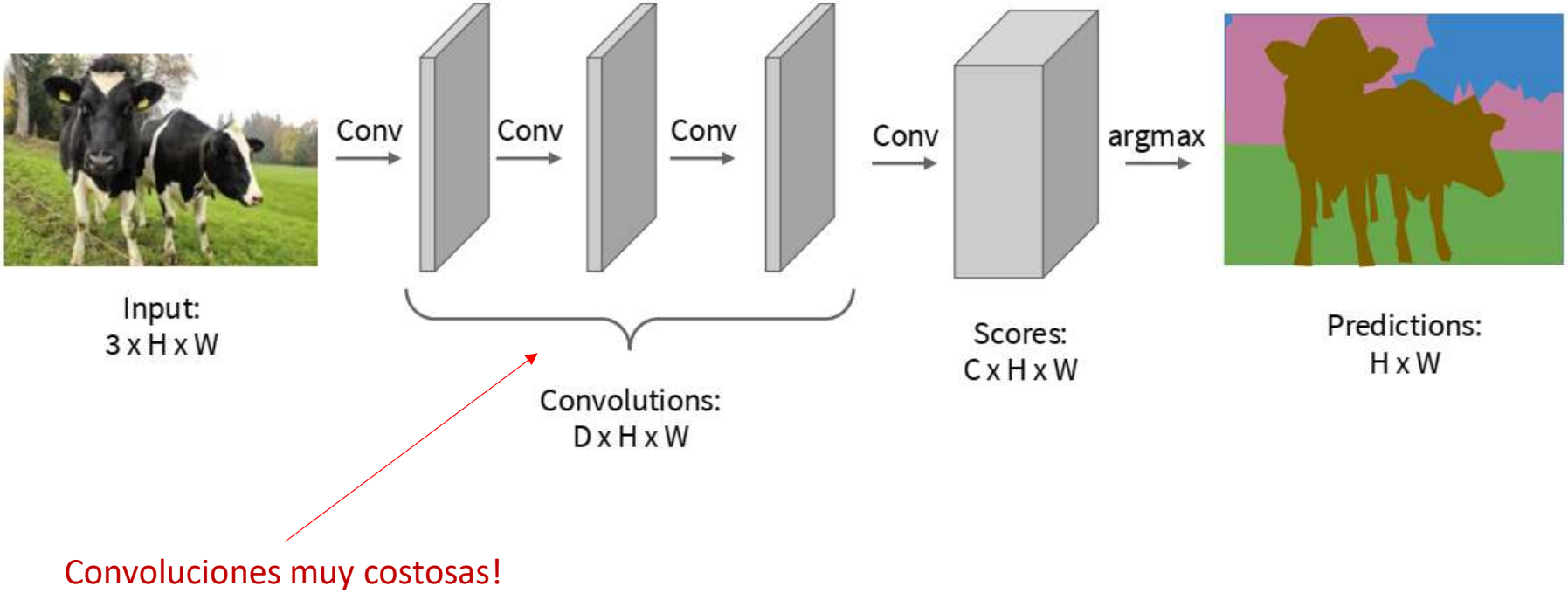
Segmentación semántica: Convolución

Diseñar una red con solo capas convolucionales sin downsampling para hacer predicciones a nivel de píxel



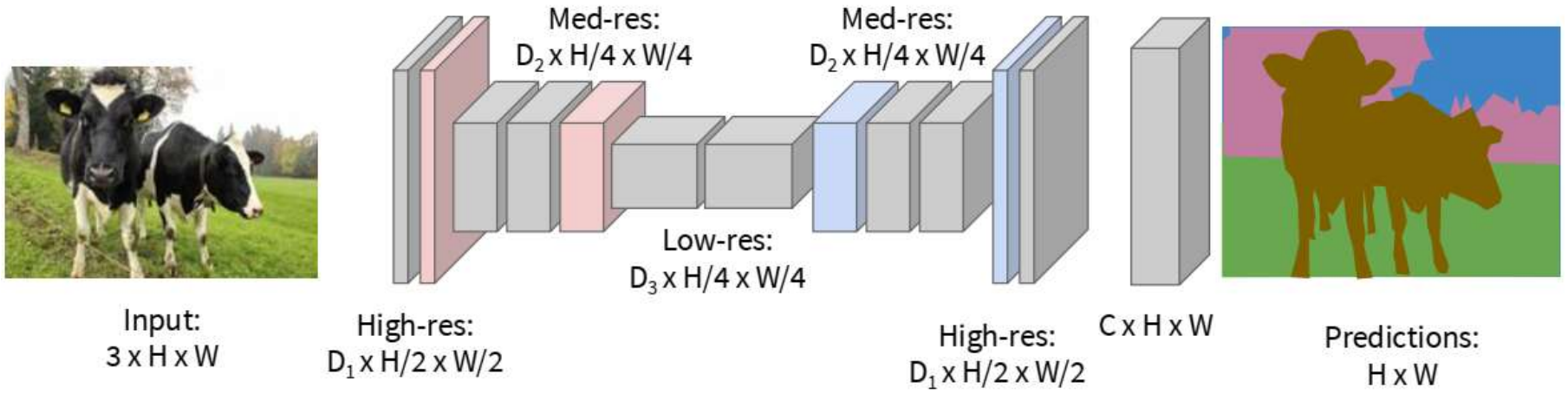
Segmentación semántica: Convolución

Diseñar una red con solo capas convolucionales sin downsampling para hacer predicciones a nivel de píxel



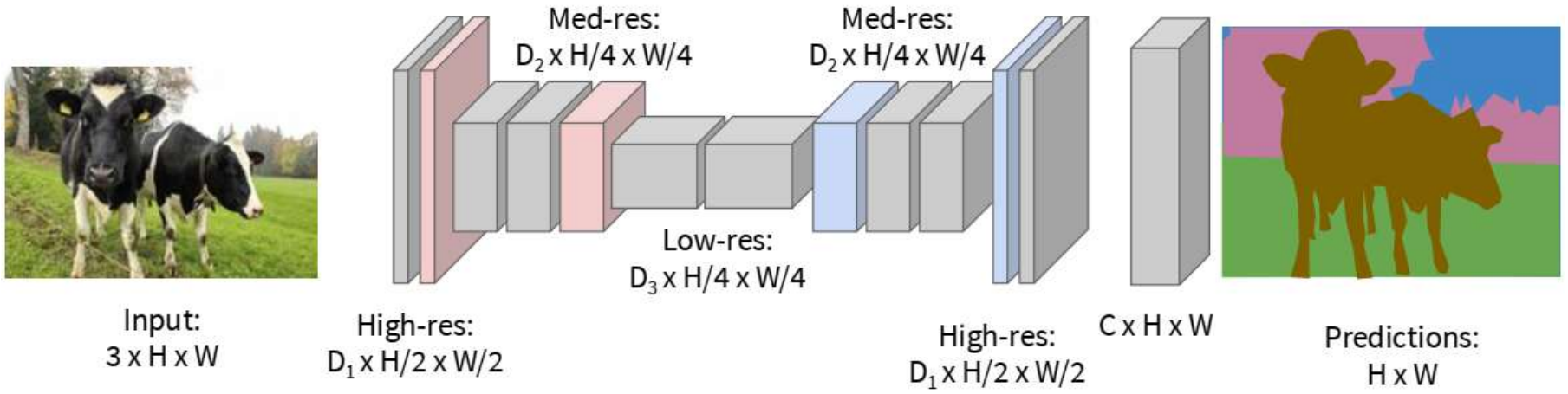
Segmentación semántica: Convolución

Idea: usar downsampling y upsampling dentro de la misma red.



Segmentación semántica: Convolución

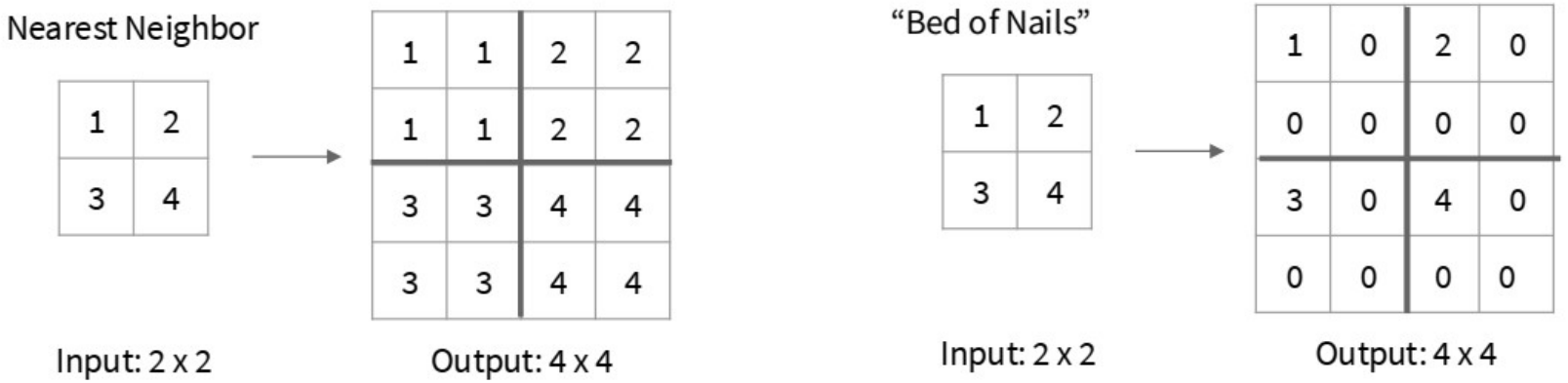
Idea: usar downsampling y upsampling dentro de la misma red.



Cómo se hace el upsampling?

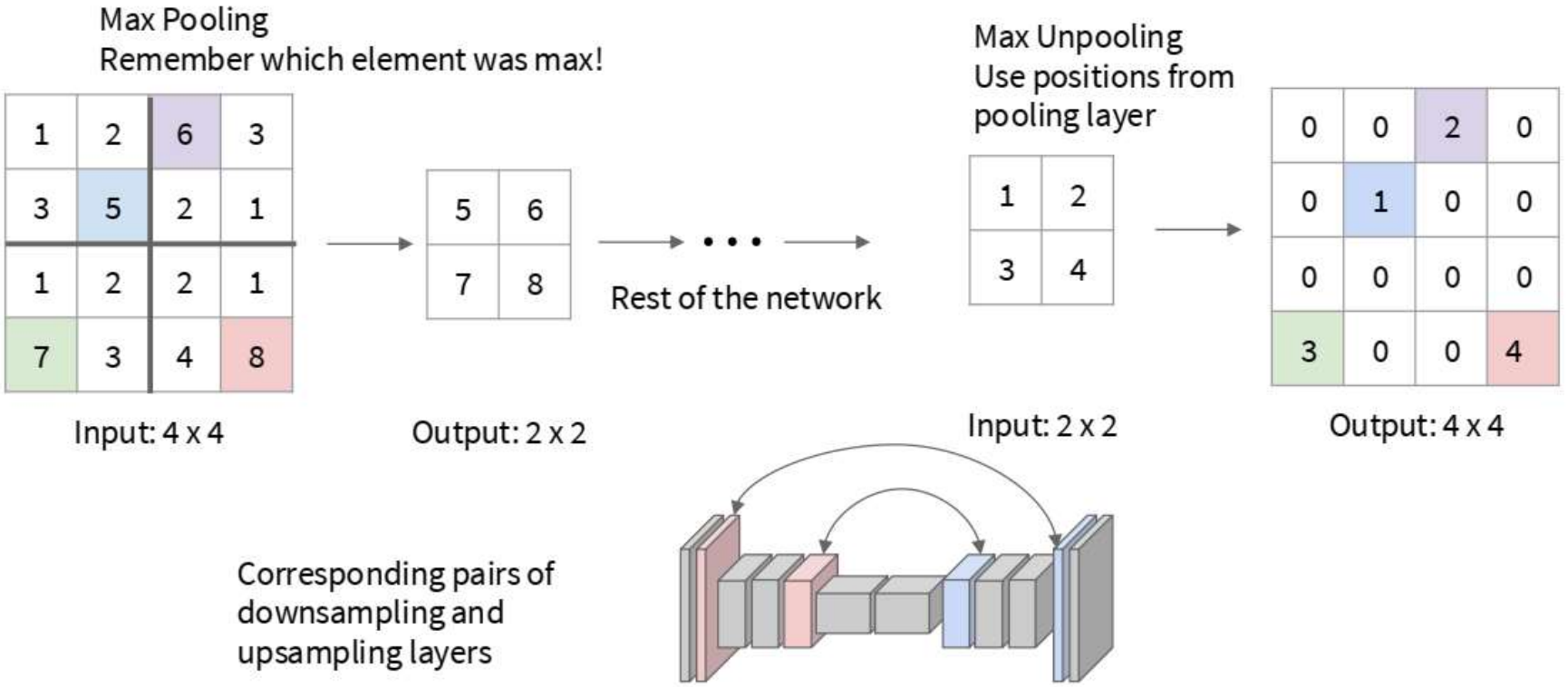
Segmentación semántica: Convolución

Upsampling: Unpooling



Segmentación semántica: Convolución

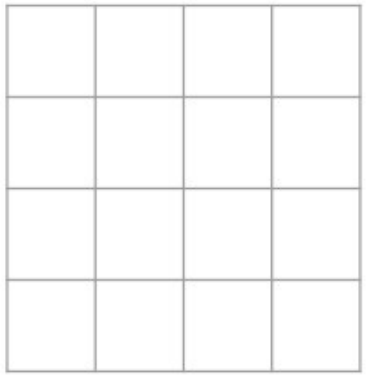
Upsampling: Max-Unpooling



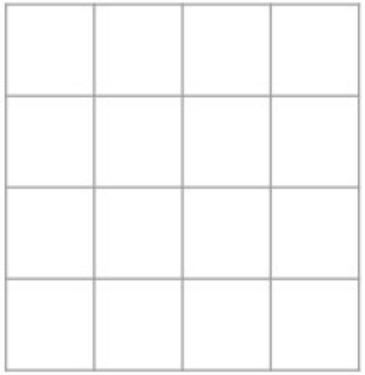
Segmentación semántica: Convolución

Se puede aprender el upsampling

Recordemos: convolución normal 3x3 , stride 1, pad 1



Input: 4 x 4

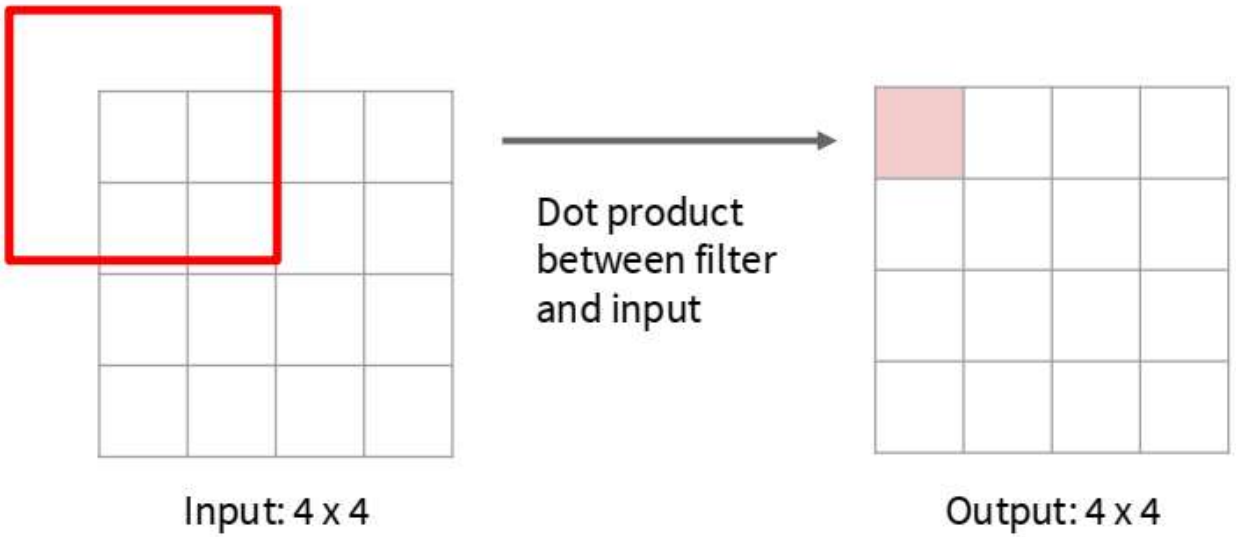


Output: 4 x 4

Segmentación semántica: Convolución

Se puede aprender el upsampling

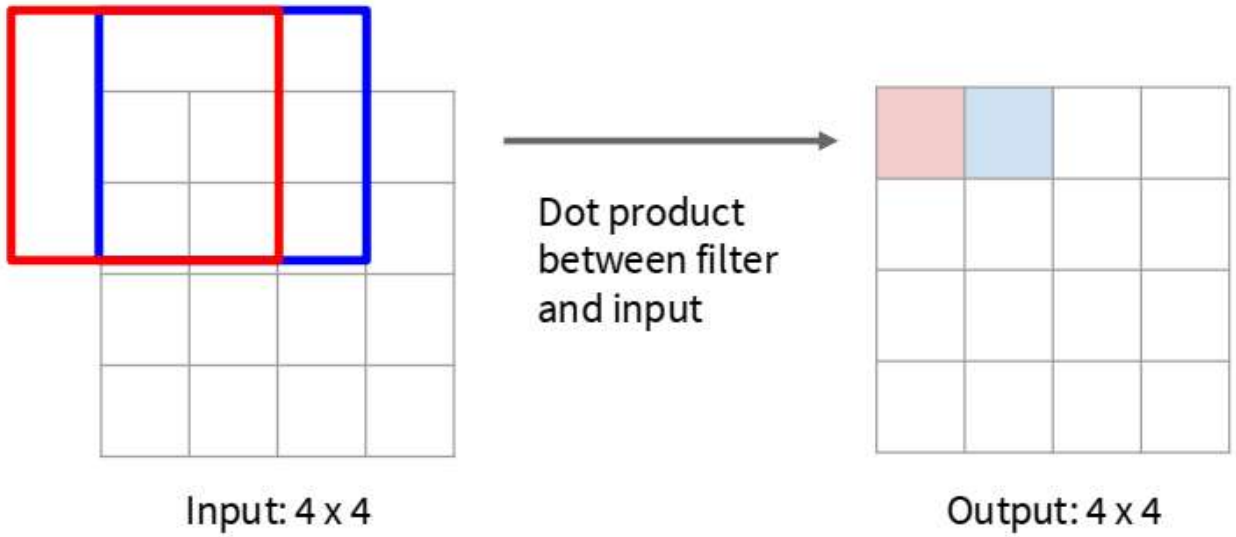
Recordemos: convolución normal 3x3 , stride 1, pad 1



Segmentación semántica: Convolución

Se puede aprender el upsampling

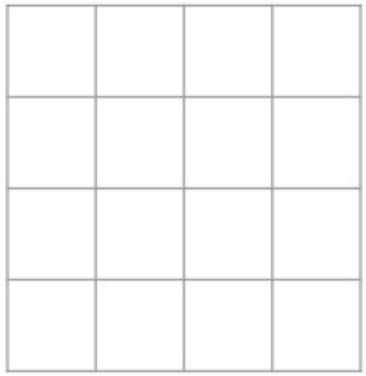
Recordemos: convolución normal 3x3 , stride 1, pad 1



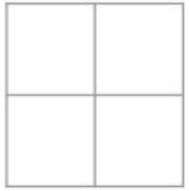
Segmentación semántica: Convolución

Se puede aprender el upsampling

Recordemos: convolución normal 3x3 , **stride 2**, pad 1



Input: 4 x 4

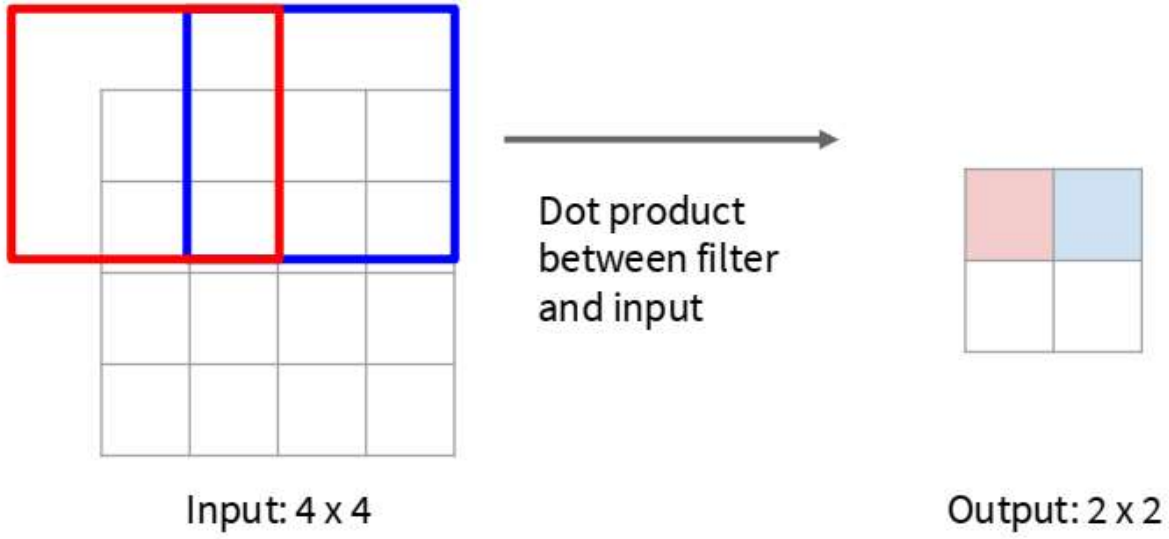


Output: 2 x 2

Segmentación semántica: Convolución

Se puede aprender el upsampling

Recordemos: convolución normal 3x3 , **stride 2**, pad 1



Filtro se mueve dos píxeles en el input por cada pixel en el output

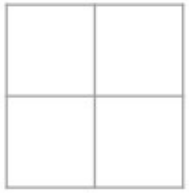
Stride: proporción entre movimiento en el input y en el output

Podemos interpretar convolución con stride como “downsampling aprendible”

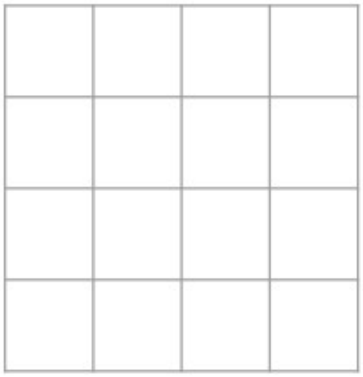
Segmentación semántica: Convolución

Se puede aprender el upsampling

Idea: convolución transpuesta 3x3 , **stride 2**, pad 1



Input: 2 x 2

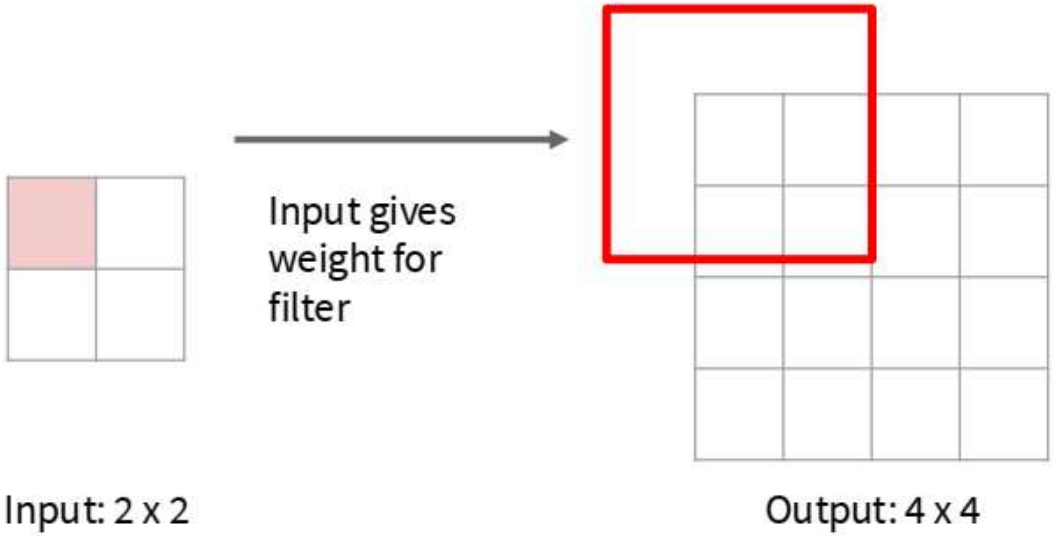


Output: 4 x 4

Segmentación semántica: Convolución

Se puede aprender el upsampling

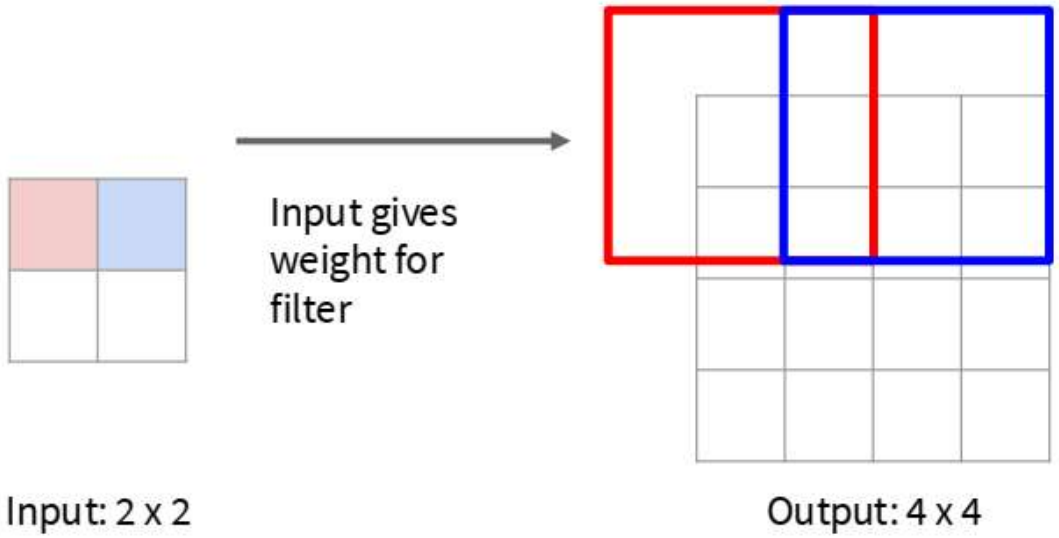
Idea: convolución transpuesta 3x3 , **stride 2**, pad 1



Segmentación semántica: Convolución

Se puede aprender el upsampling

Idea: convolución transpuesta 3x3 , **stride 2**, pad 1



Filtro se mueve dos píxeles en el output por cada pixel en el input

Stride: proporción entre movimiento en el output y en el input

Segmentación semántica: Convolución

Se puede aprender el upsampling

Idea: convolución transpuesta 3x3 , **stride 2**, pad 1

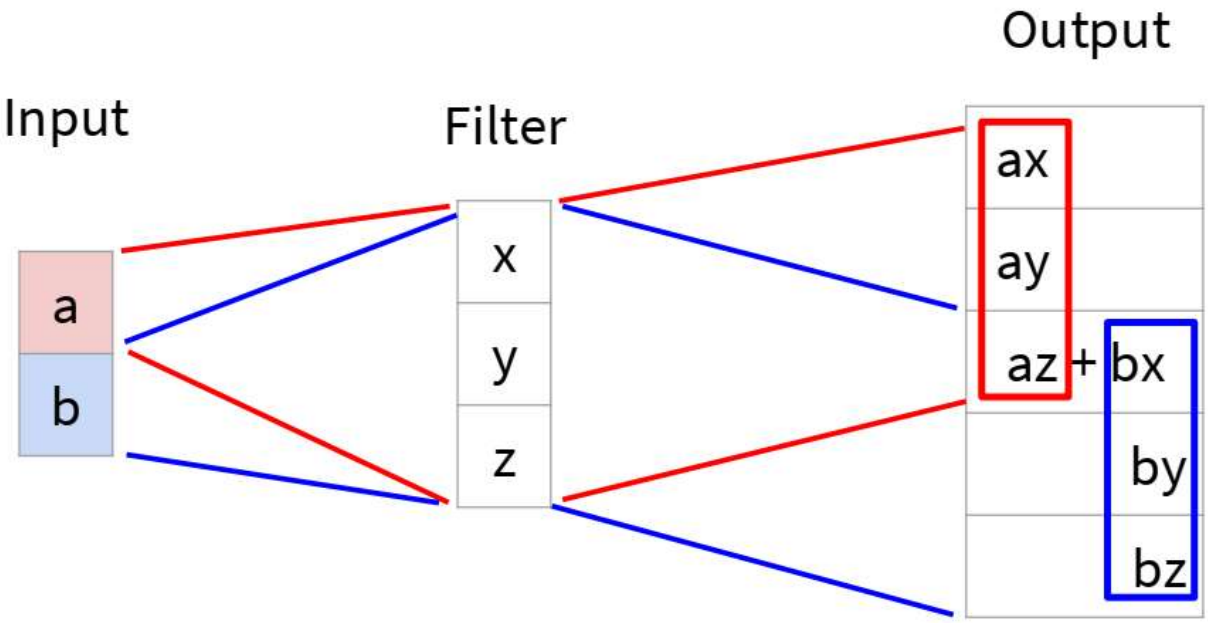
Sumas en donde el output hace overlap

Filtro se mueve dos píxeles en el output por cada pixel en el input

Stride: proporción entre movimiento en el output y en el input

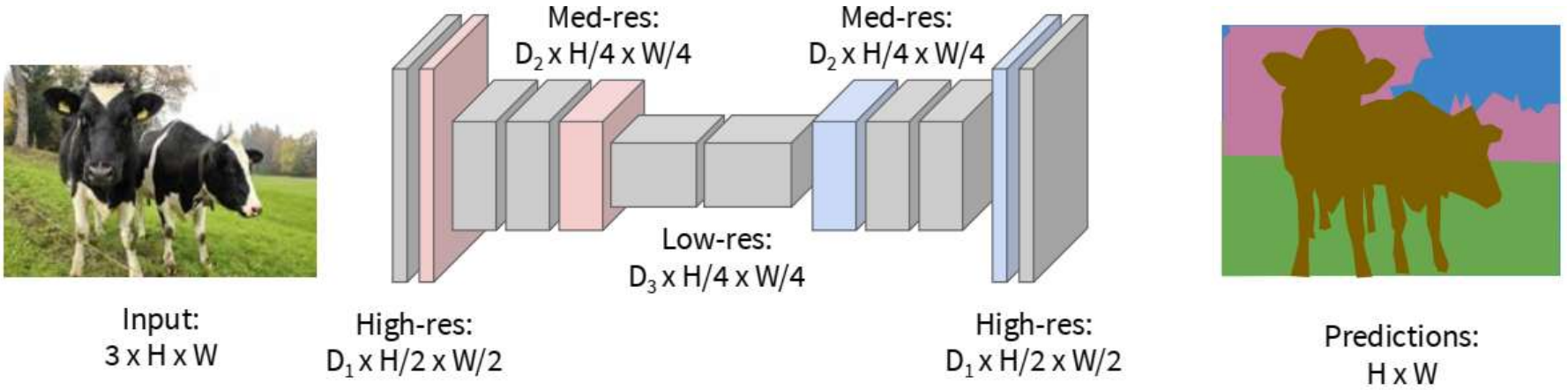
Segmentación semántica: Convolución

Se puede aprender el upsampling



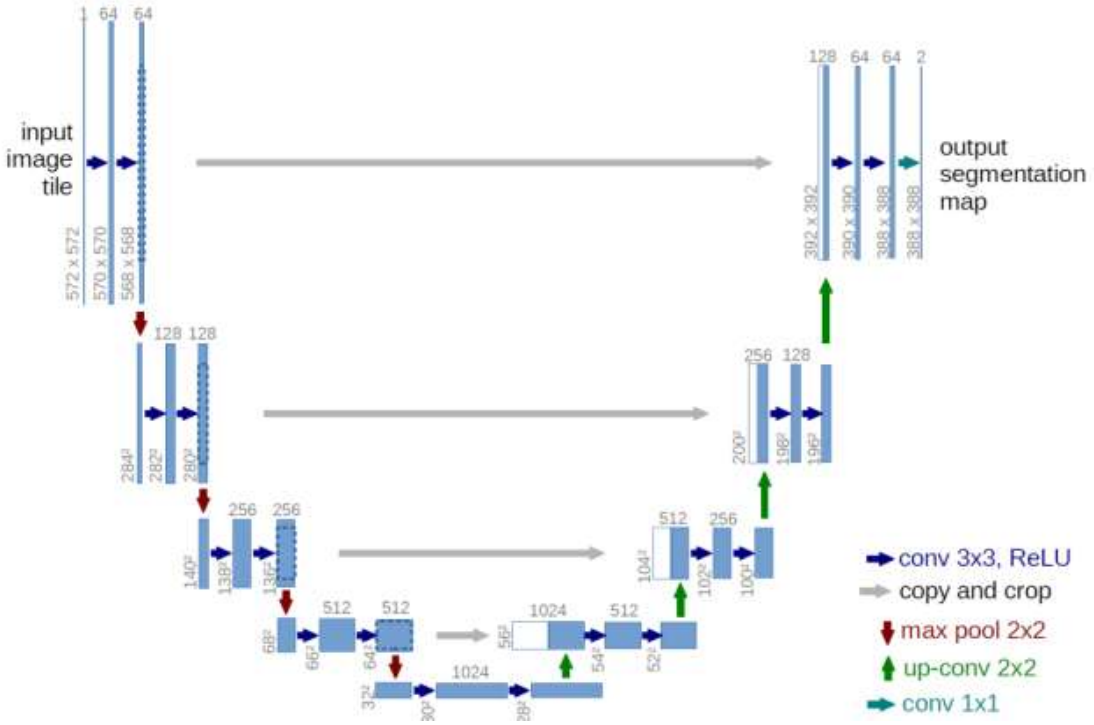
Output contiene copias del filtro ponderadas por el input, con sumas en donde hay overlap

Segmentación semántica: Convolución



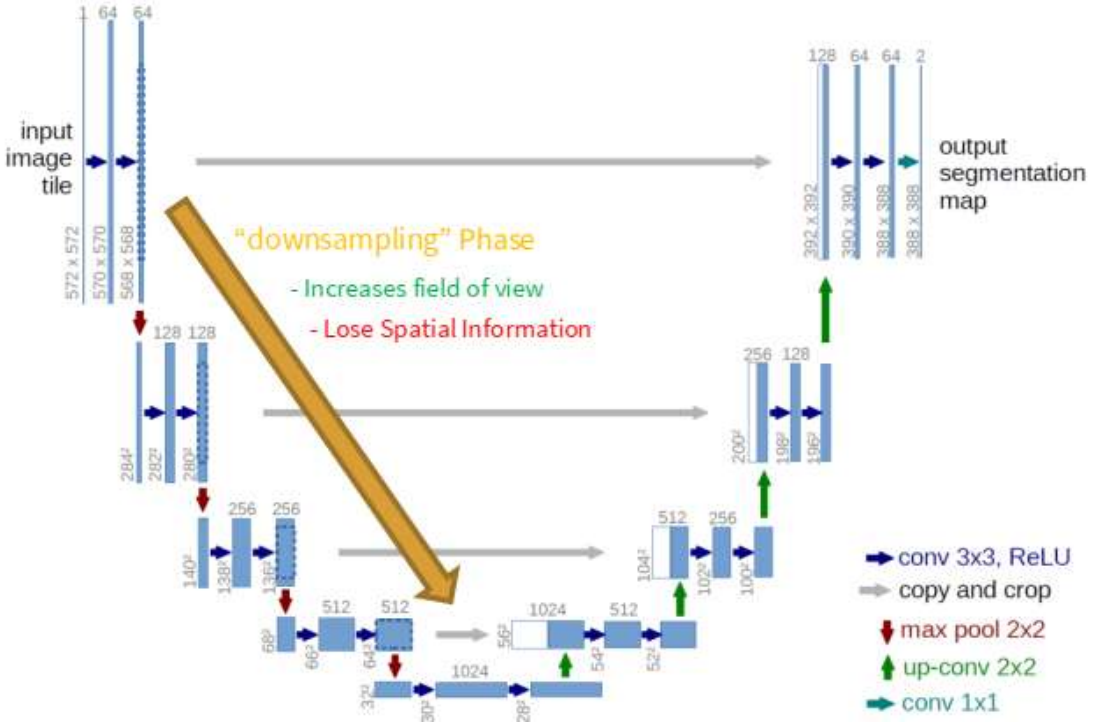
Segmentación semántica: Convolución

UNet



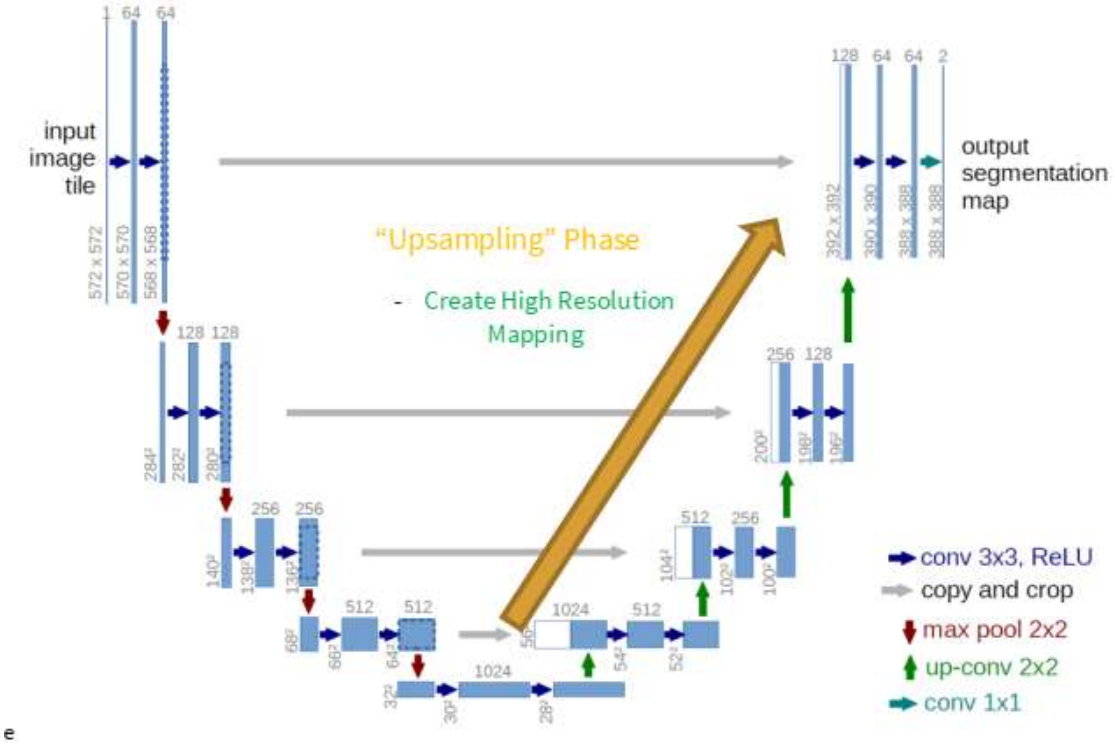
Segmentación semántica: Convolución

UNet



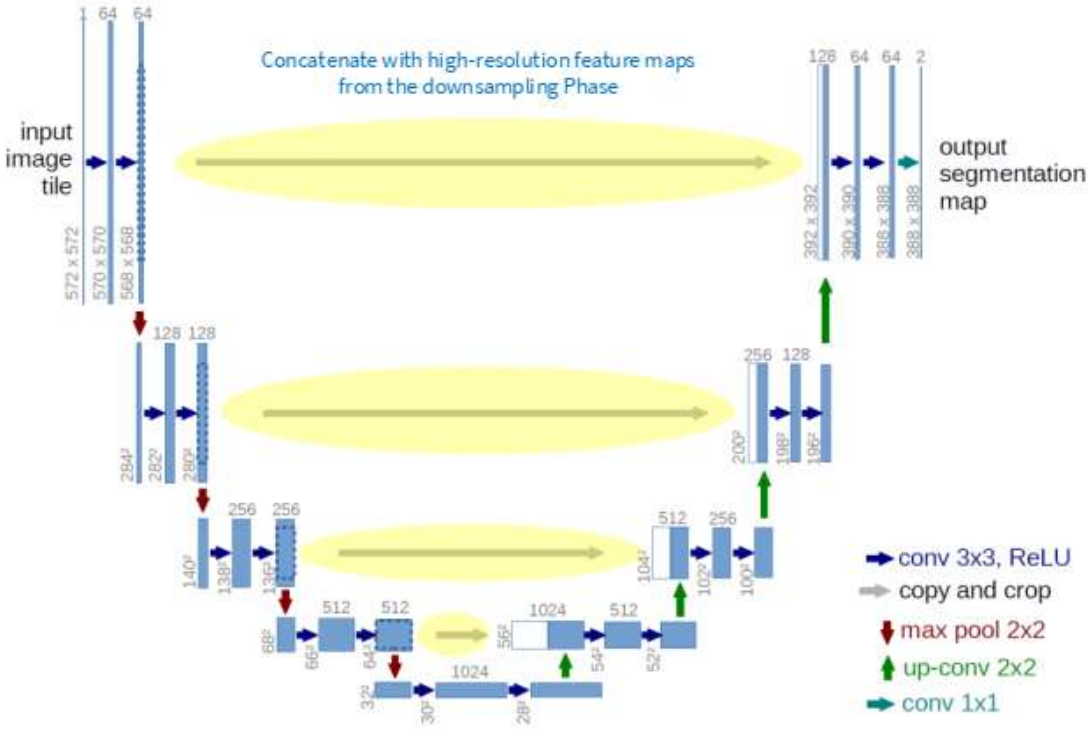
Segmentación semántica: Convolución

UNet



Segmentación semántica: Convolución

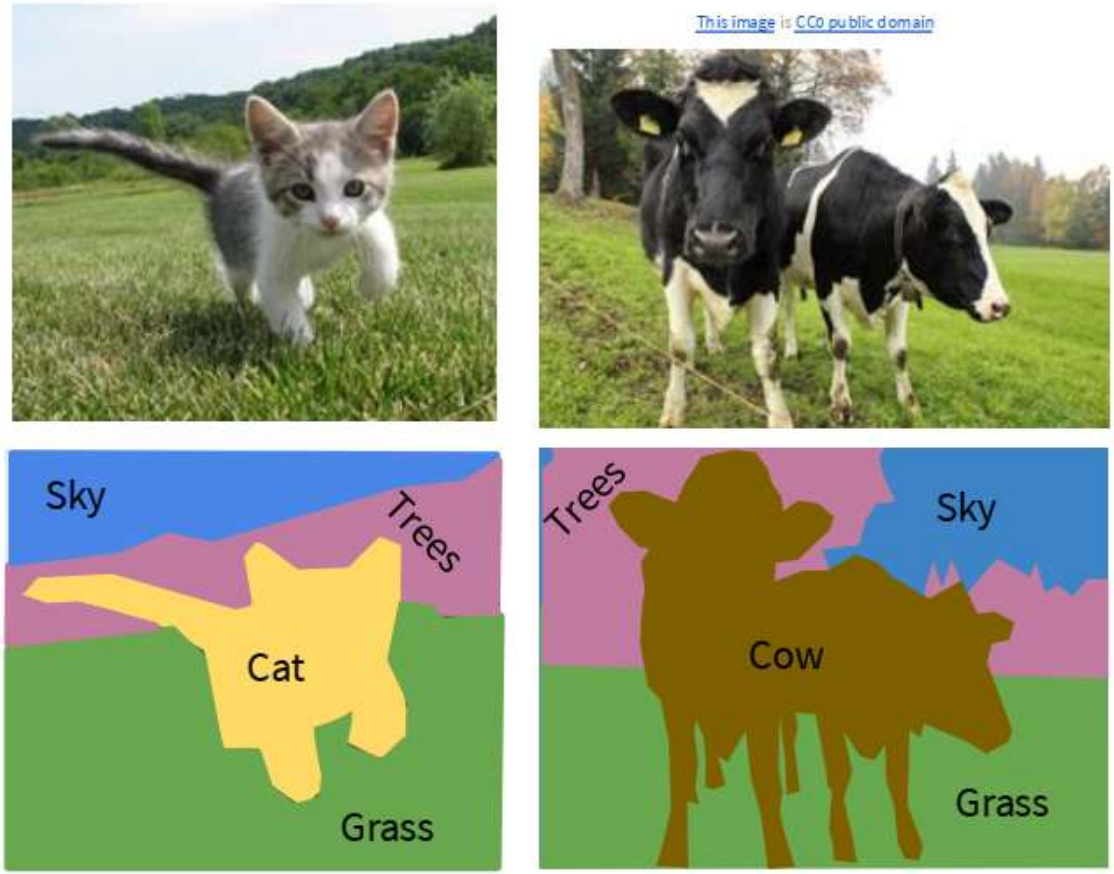
UNet



Segmentación semántica

Etiquetar cada píxel de la imagen con una clase

No se diferencian instancias!



Detección de objetos

Classification

CAT

No spatial extent

Semantic Segmentation

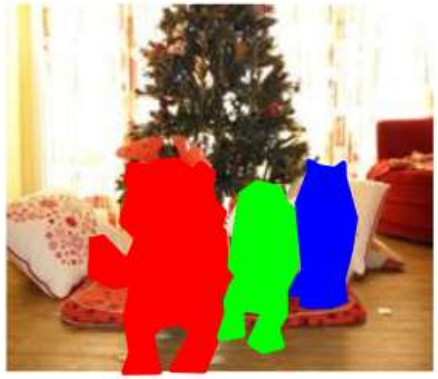
GRASS, CAT, TREE, SKY

No objects, just pixels

Object Detection

DOG, DOG, CAT

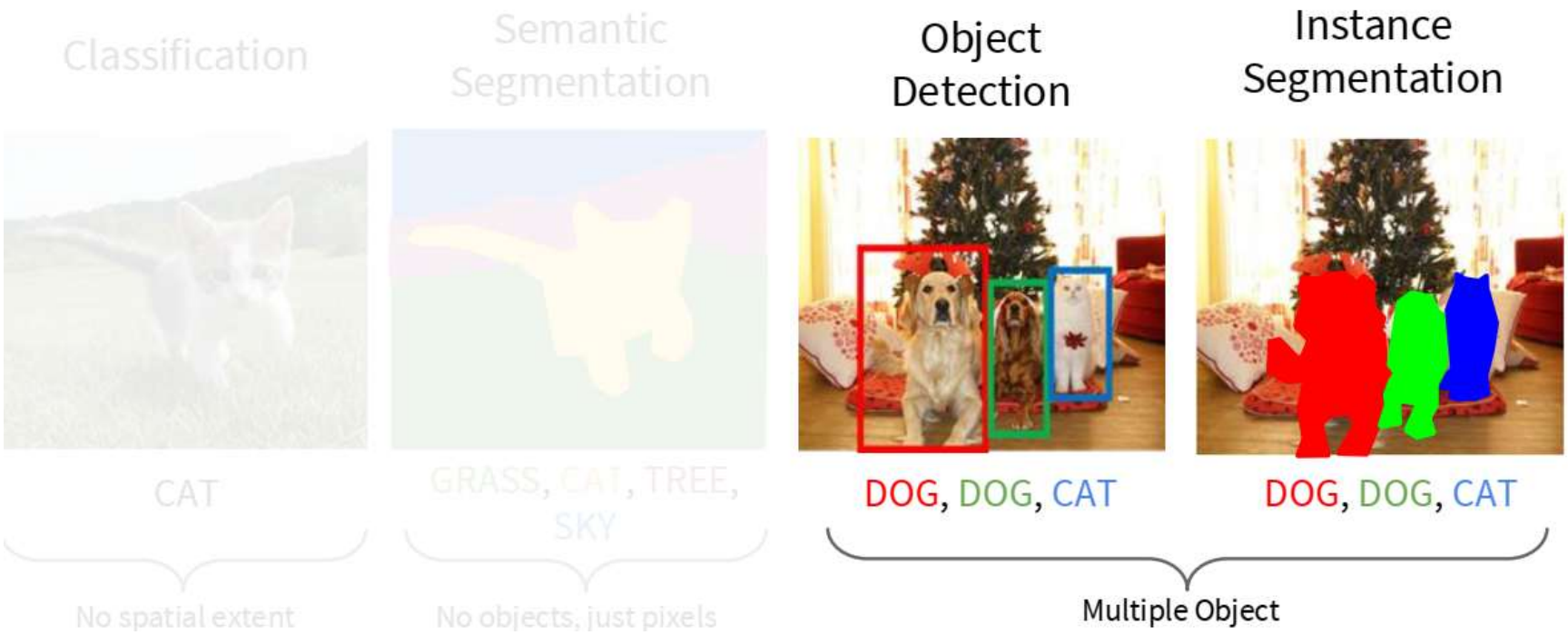
Instance Segmentation



DOG, DOG, CAT

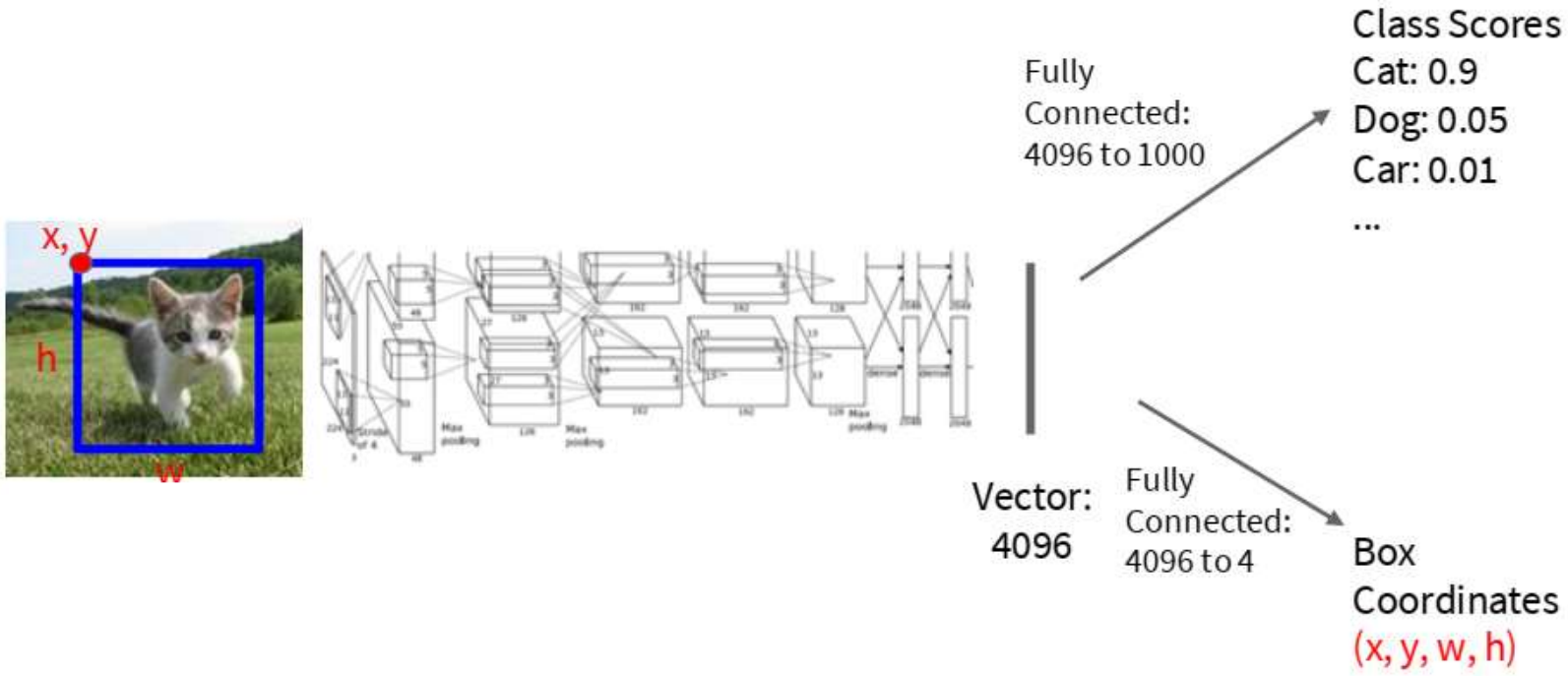
Multiple Object

Detección de objetos



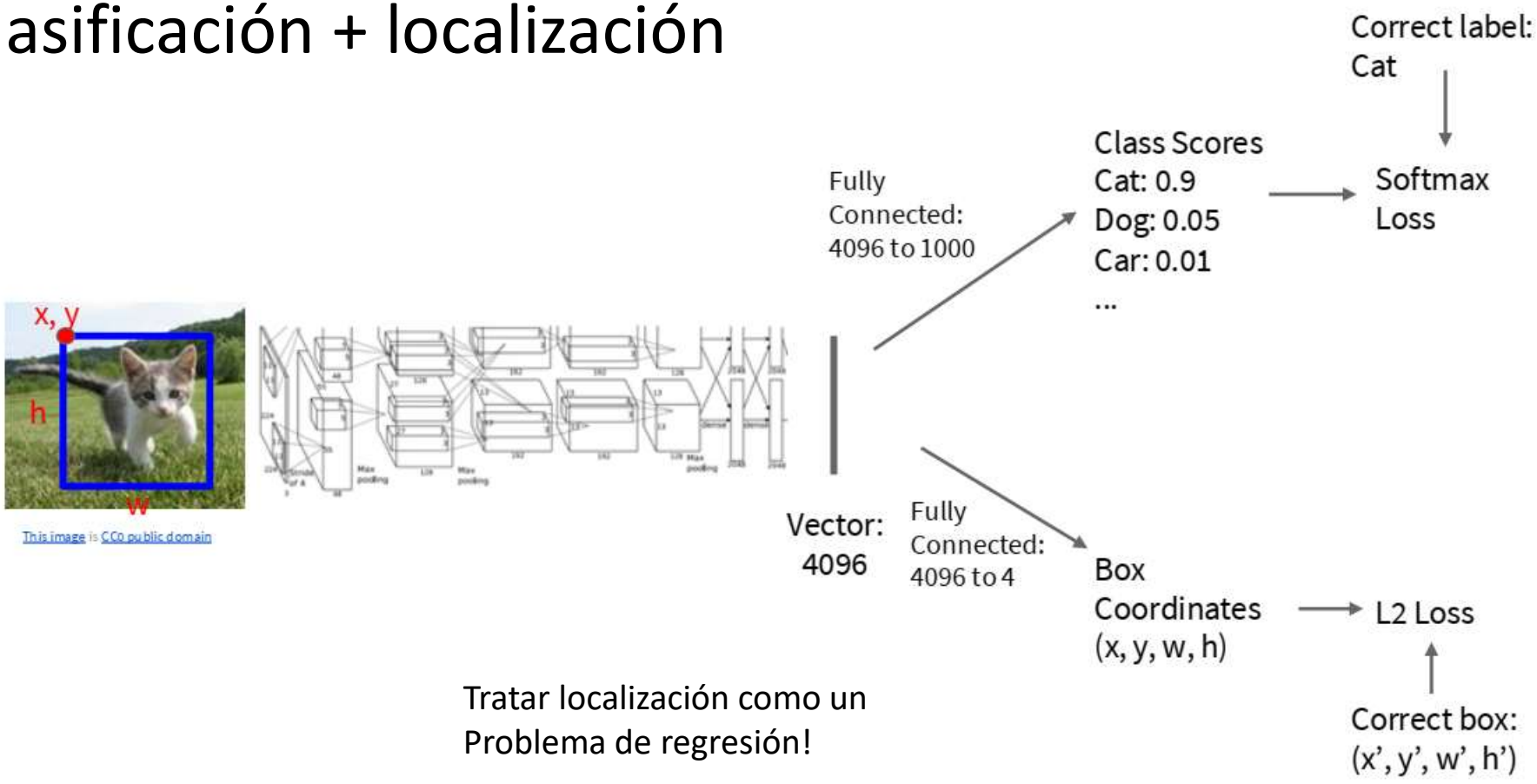
Detección de objetos: un objeto

Clasificación + localización



Detección de objetos: un objeto

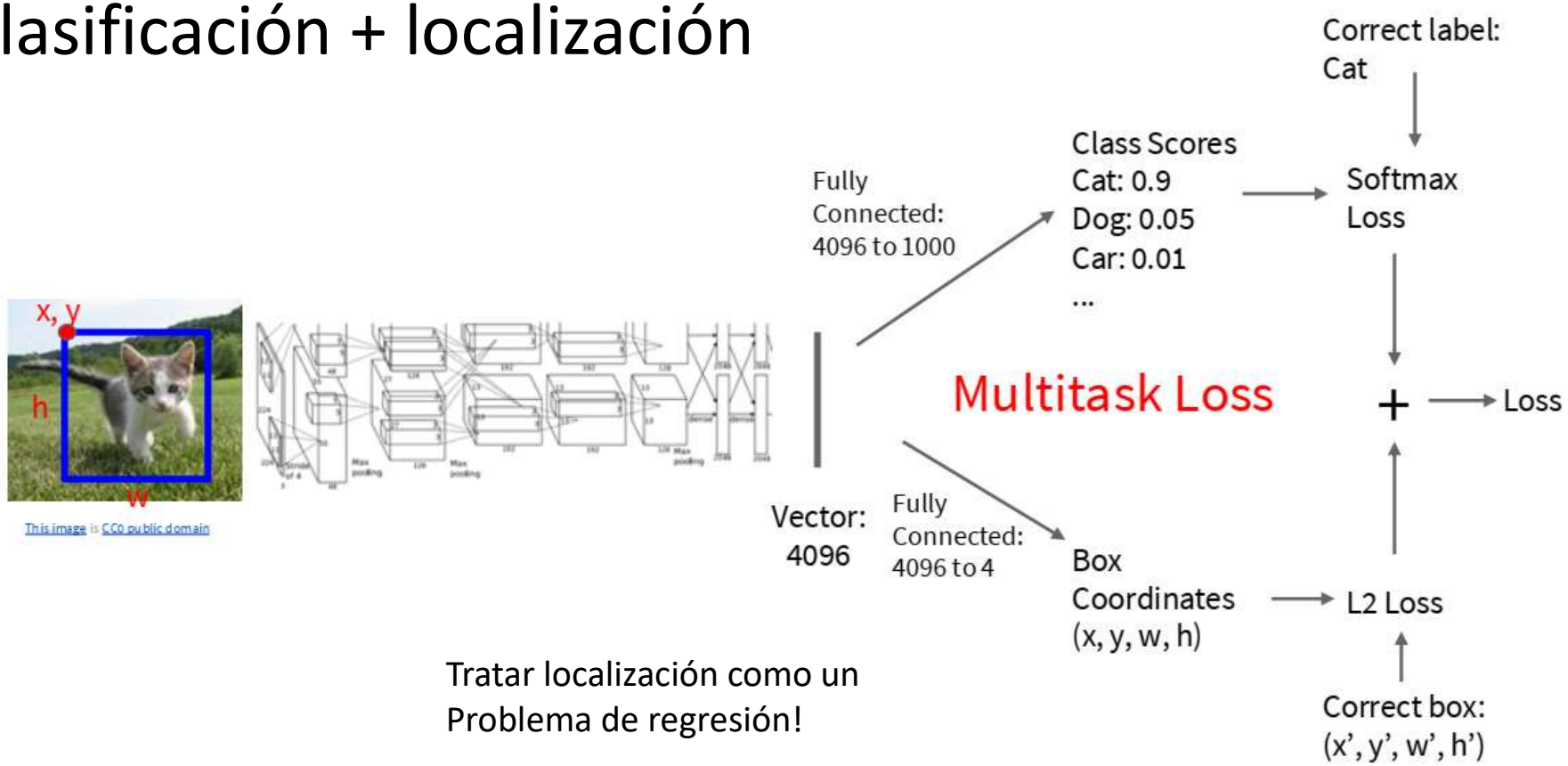
Clasificación + localización



Tratar localización como un
Problema de regresión!

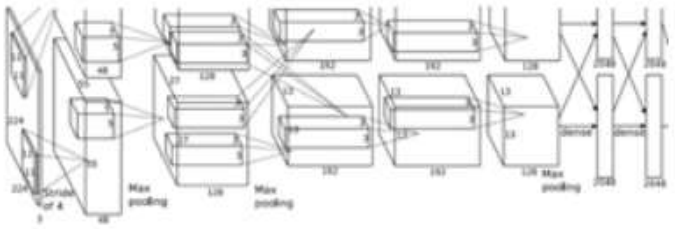
Detección de objetos: un objeto

Clasificación + localización

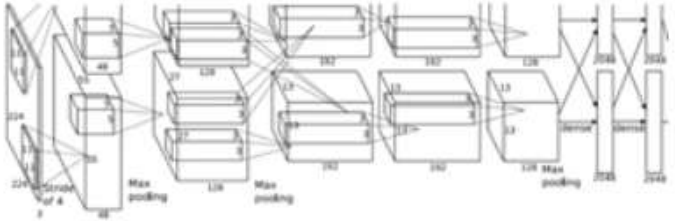


Tratar localización como un
Problema de regresión!

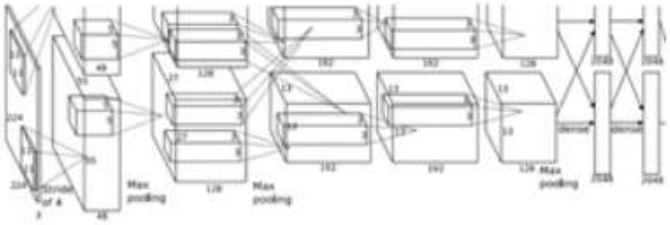
Detección de objetos: múltiples objetos



CAT: (x, y, w, h)



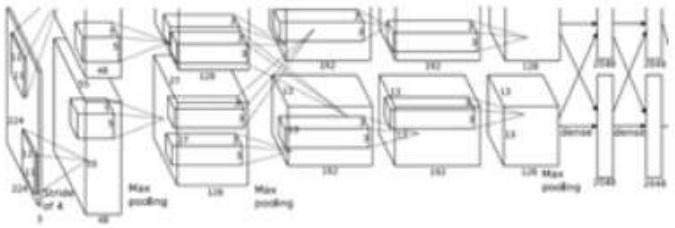
DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)



DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
….

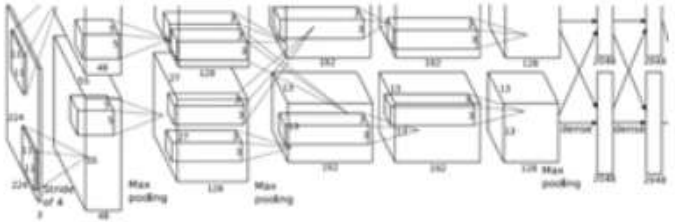
Detección de objetos: múltiples objetos

Cada imagen necesita un número diferente de salidas



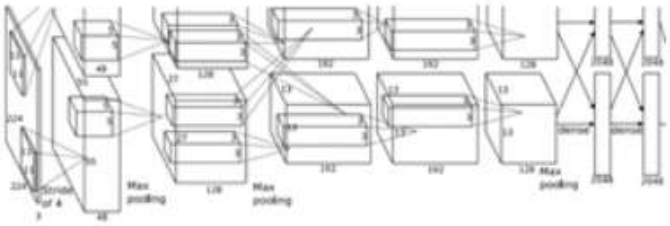
CAT: (x, y, w, h)

4 números



DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)

12 números

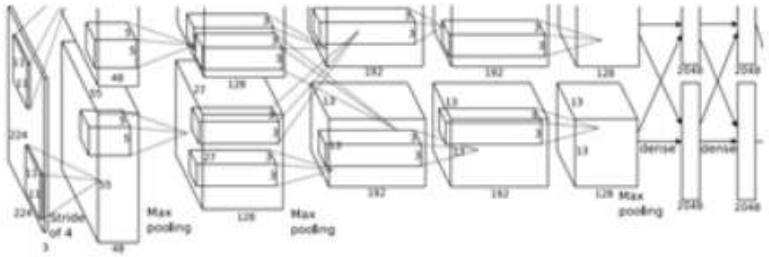


DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
....

Muchos números

Detección de objetos: múltiples objetos

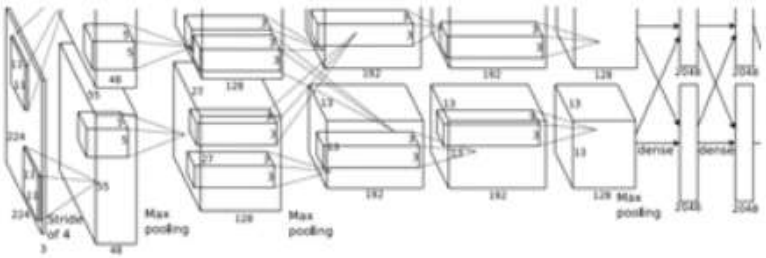
Aplicar una CNN a muchos diferentes crops de la imagen,
CNN clasifica cada crop como objeto o fondo



Dog? NO
Cat? NO
Background? YES

Detección de objetos: múltiples objetos

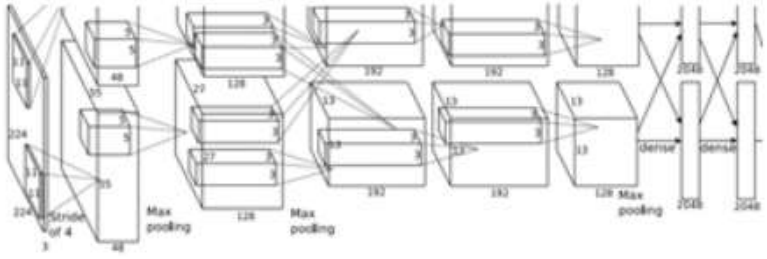
Aplicar una CNN a muchos diferentes crops de la imagen,
CNN clasifica cada crop como objeto o fondo



Dog? YES
Cat? NO
Background? NO

Detección de objetos: múltiples objetos

Aplicar una CNN a muchos diferentes crops de la imagen,
CNN clasifica cada crop como objeto o fondo

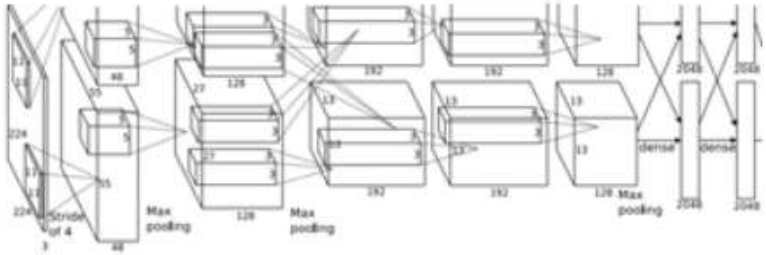


Dog? YES
Cat? NO
Background? NO

Detección de objetos: múltiples objetos

Cuál es el problema con este enfoque?

Aplicar una CNN a muchos diferentes crops de la imagen,
CNN clasifica cada crop como objeto o fondo

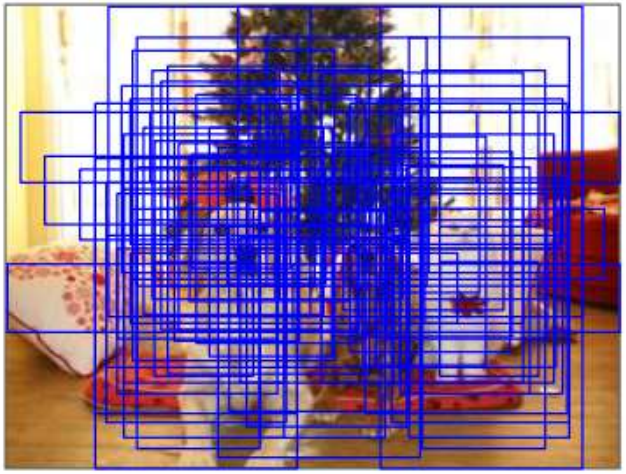


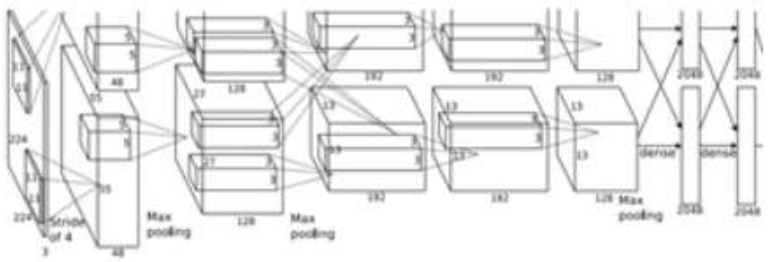
Dog? NO
Cat? YES
Background? NO

Detección de objetos: múltiples objetos

Necesitas aplicar CNN a un gran número de Lugares, escalas y aspect ratios. Es muy caro!

Aplicar una CNN a muchos diferentes crops de la imagen, CNN clasifica cada crop como objeto o fondo





Dog? NO
Cat? YES
Background? NO

Region proposals: Selective Search

- Encontrar regiones con potenciales objetos
- Selective Search entrega 2000 regiones en pocos segundos usando CPU

R-CNN

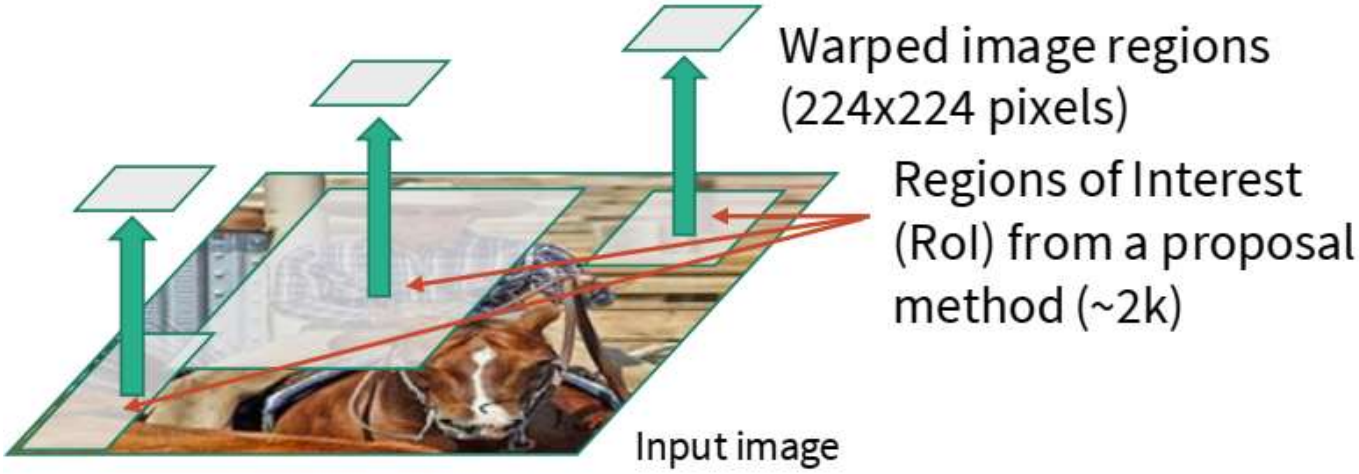
Input image

R-CNN

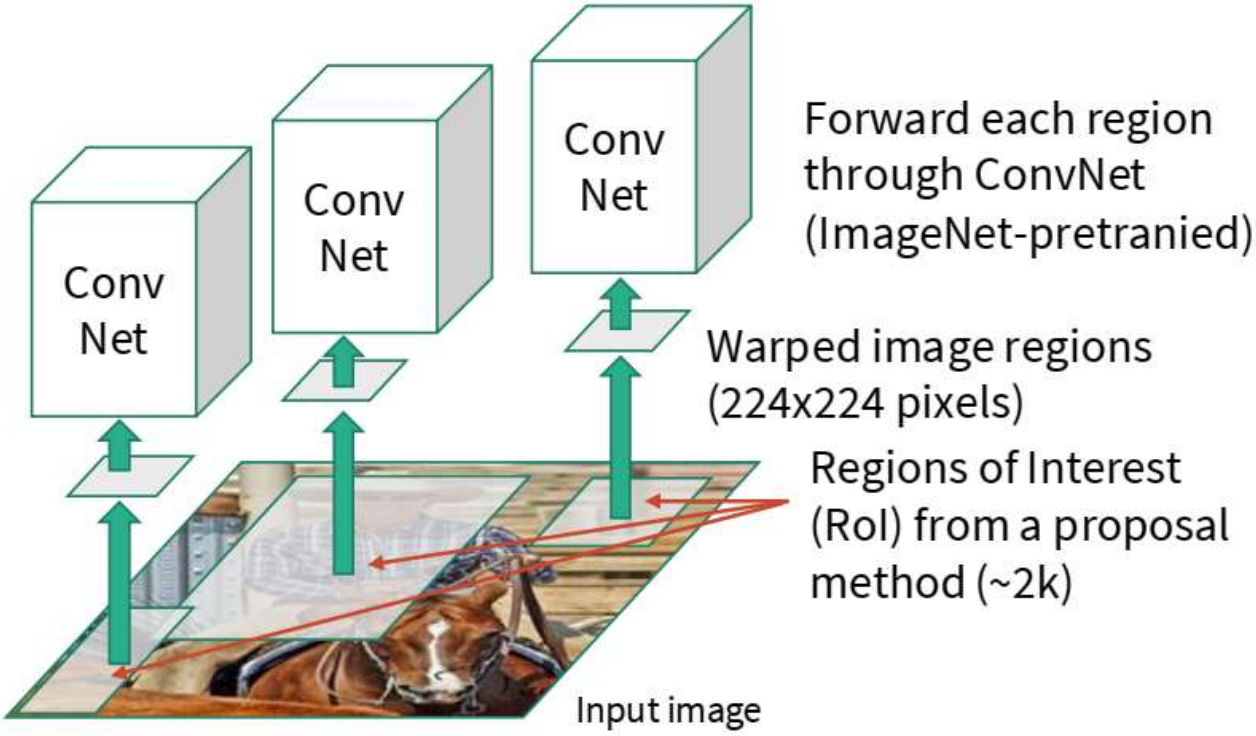
Regions of Interest (RoI) from a proposal method (~2k)

Input image

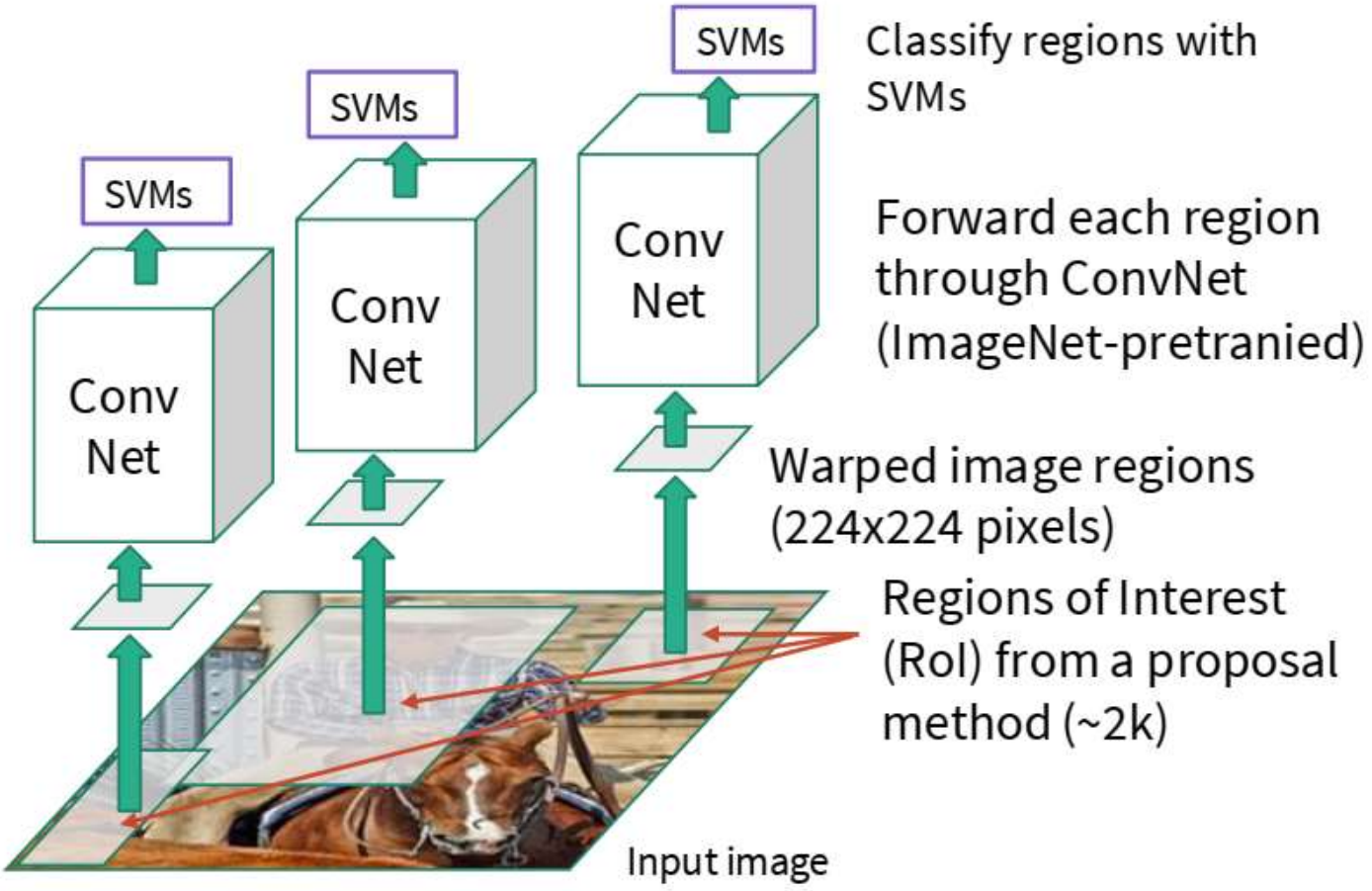
R-CNN


Warped image regions
(224x224 pixels)
Regions of Interest
(RoI) from a proposal
method (~2k)
Input image

R-CNN

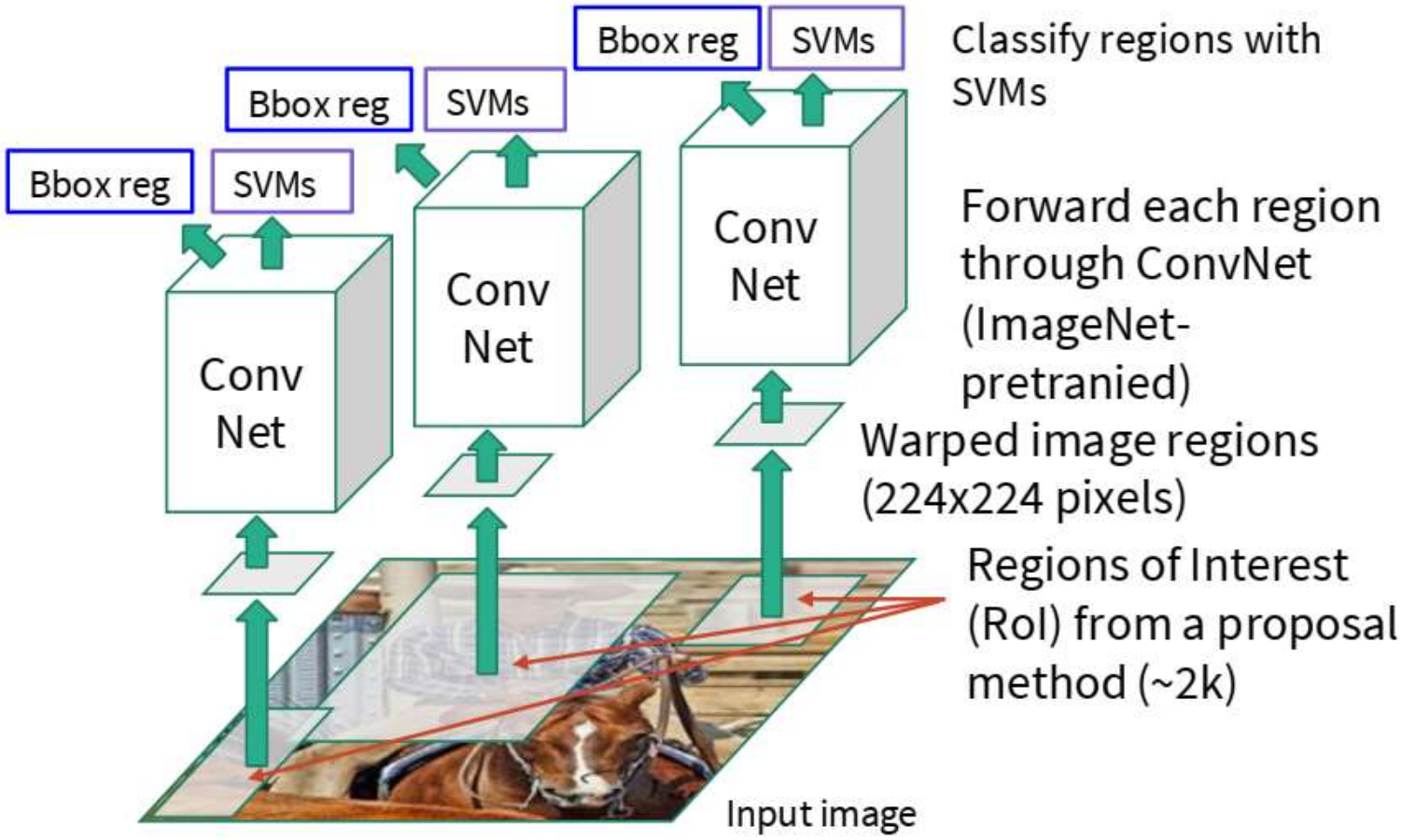

Conv Net
Conv Net
Conv Net
Forward each region through ConvNet (ImageNet-pretranied)
Warped image regions (224x224 pixels)
Regions of Interest (RoI) from a proposal method (~2k)
Input image

R-CNN

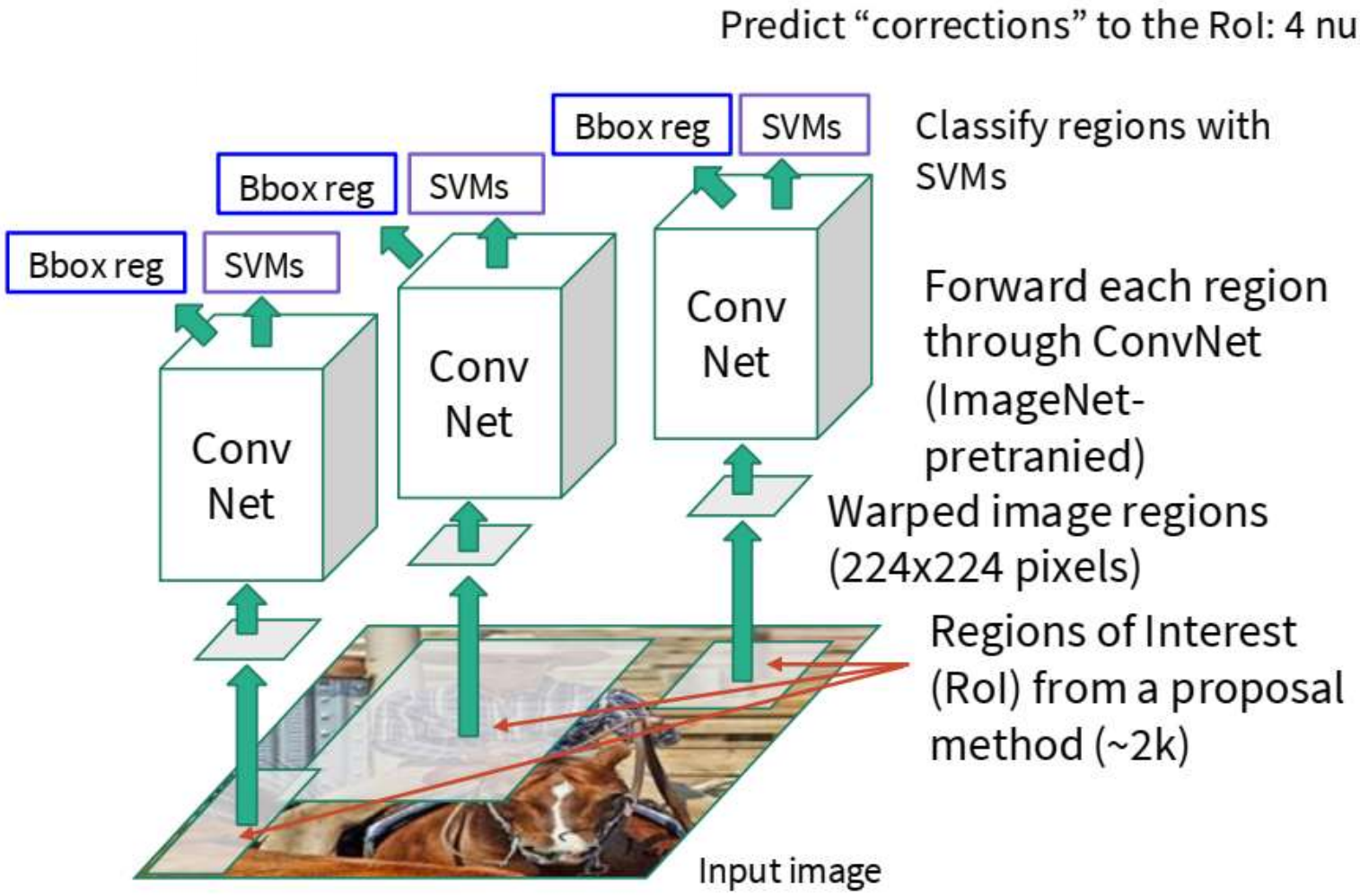

SVMs
SVMs
SVMs
Conv Net
Conv Net
Conv Net
Classify regions with SVMs
Forward each region through ConvNet (ImageNet-pretranied)
Warped image regions (224x224 pixels)
Regions of Interest (RoI) from a proposal method (~2k)
Input image

R-CNN

Predict “corrections” to the RoI: 4 numbers: (dx, dy, dw, dh)



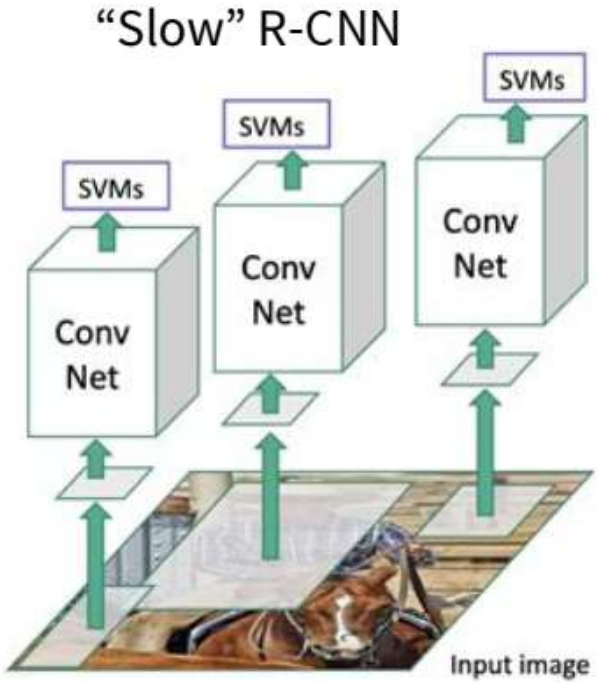
R-CNN



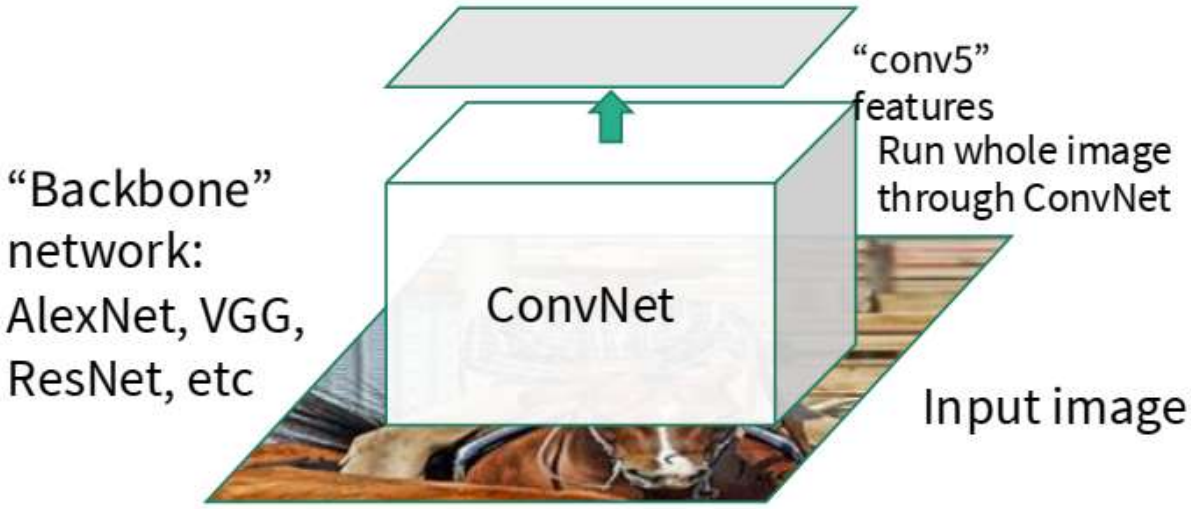
Problema: 2k predicciones
Para cada imagen

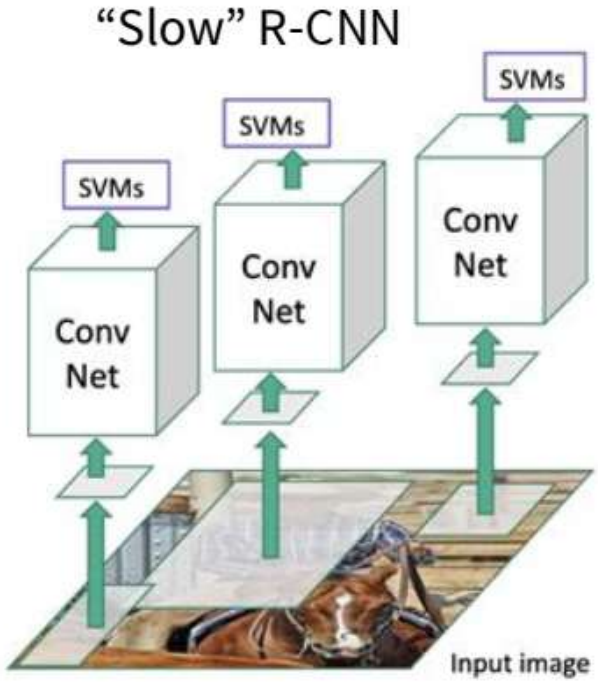
Fast R-CNN

Input image

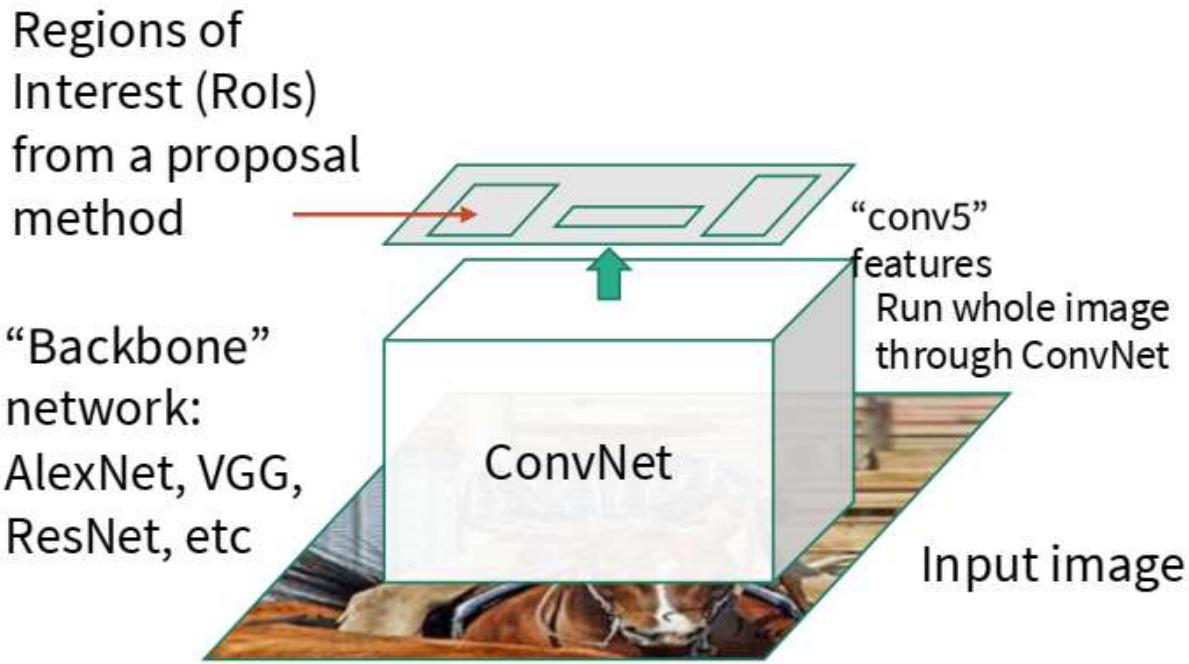


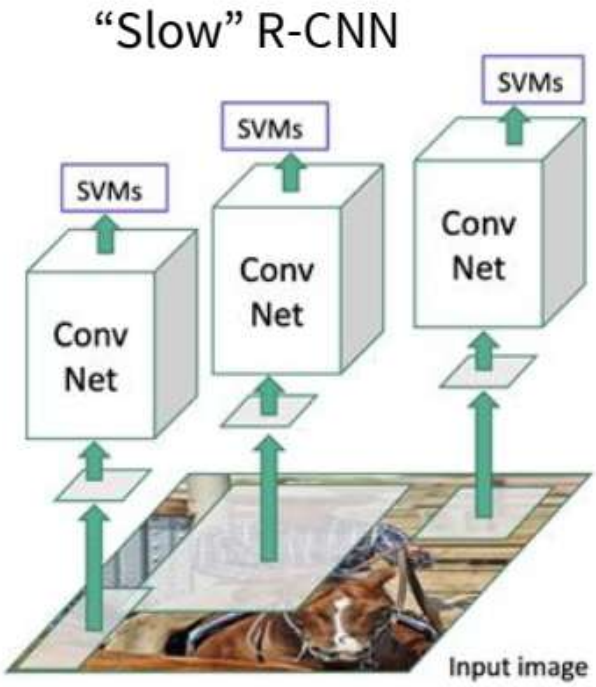
Fast R-CNN


"Backbone" network: AlexNet, VGG, ResNet, etc
ConvNet
"conv5" features
Run whole image through ConvNet
Input image

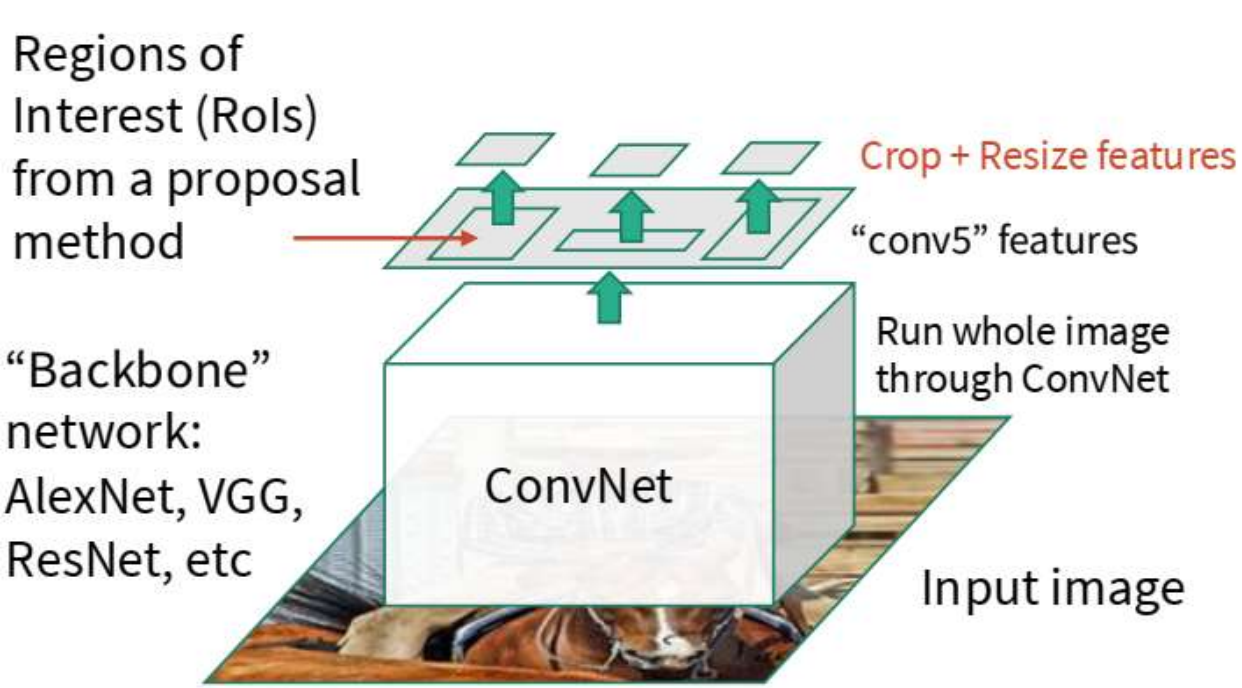

"Slow" R-CNN
SVMs
SVMs
SVMs
Conv Net
Conv Net
Conv Net
Input image

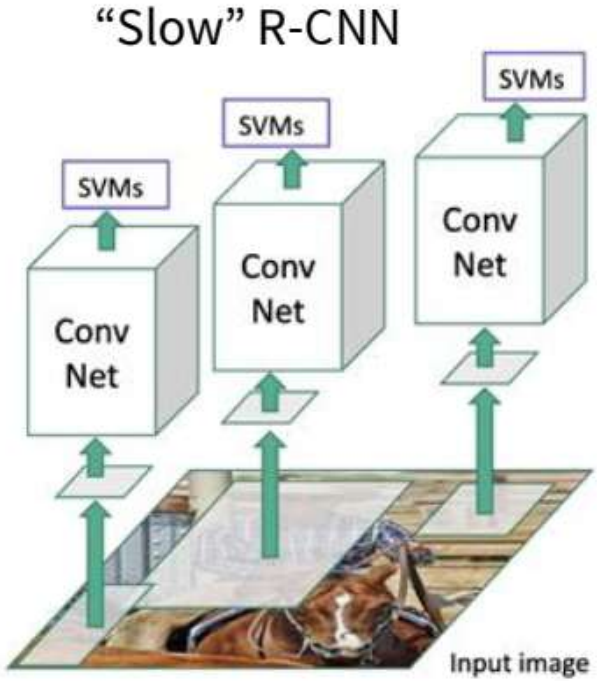
Fast R-CNN


Regions of Interest (RoIs) from a proposal method
"Backbone" network: AlexNet, VGG, ResNet, etc
"conv5" features
Run whole image through ConvNet
ConvNet
Input image

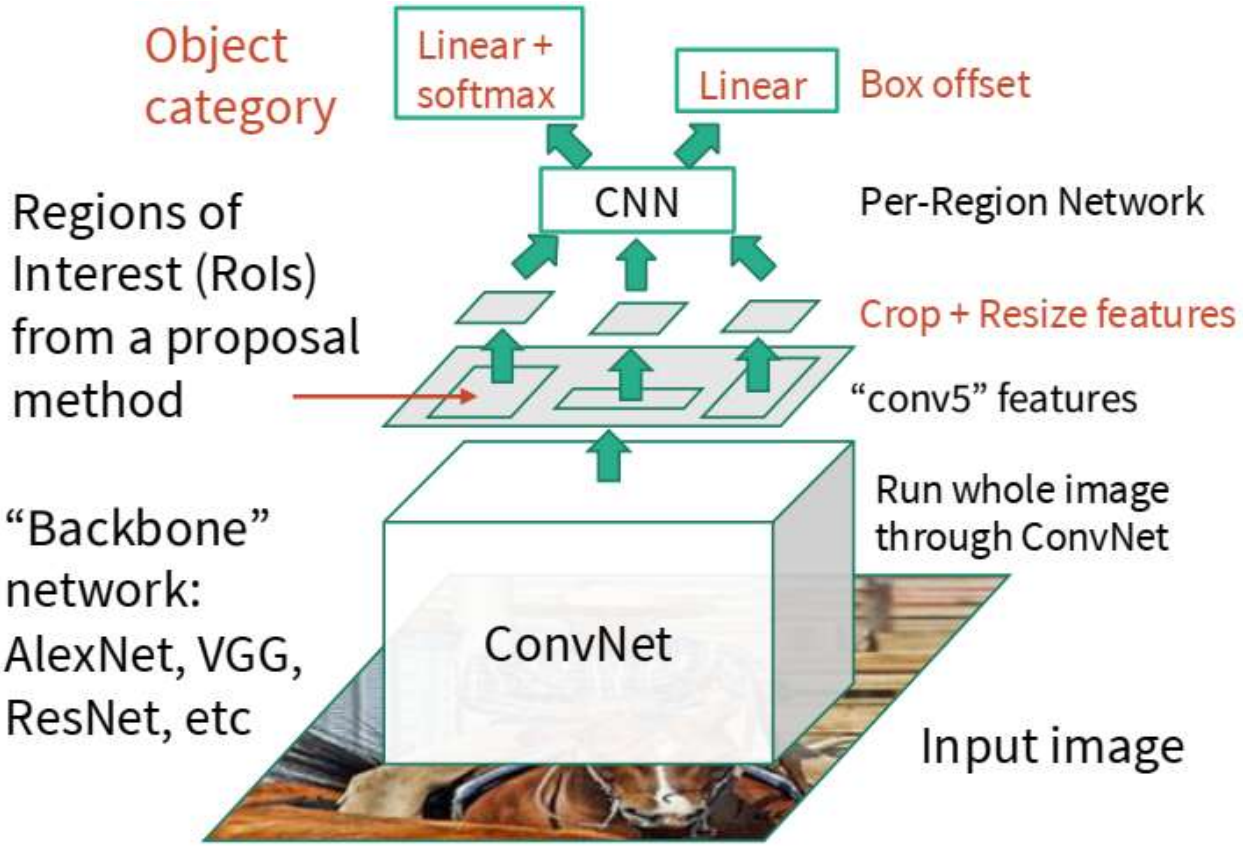

"Slow" R-CNN
SVMs
SVMs
SVMs
Conv Net
Conv Net
Conv Net
Input image

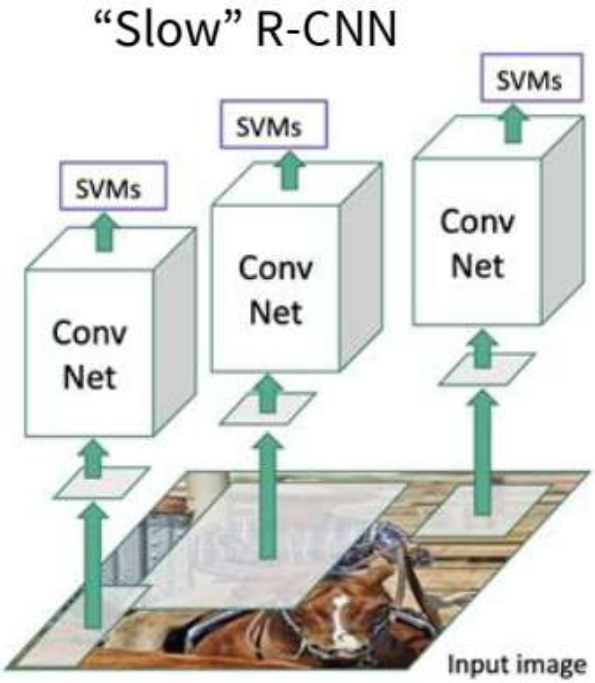
Fast R-CNN


Regions of Interest (RoIs) from a proposal method
"Backbone" network: AlexNet, VGG, ResNet, etc
Crop + Resize features
"conv5" features
Run whole image through ConvNet
ConvNet
Input image

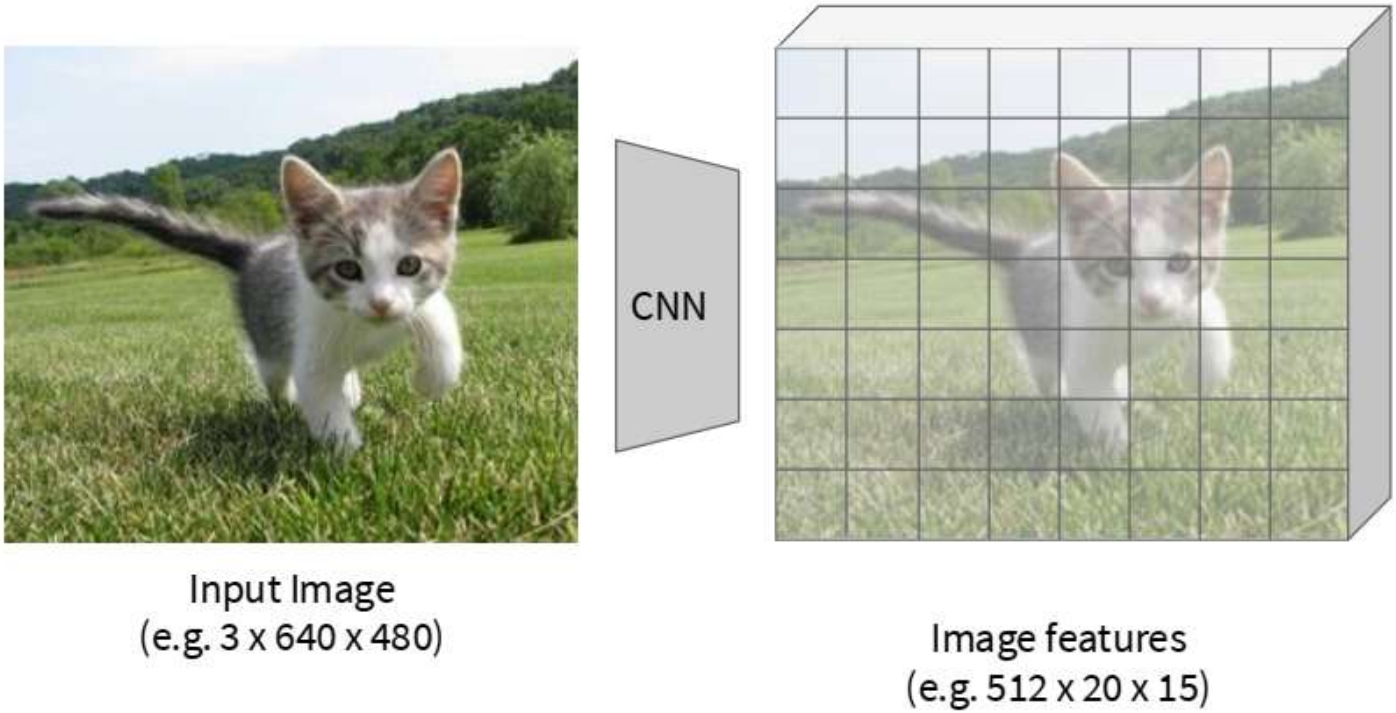

"Slow" R-CNN
SVMs
SVMs
SVMs
Conv Net
Conv Net
Conv Net
Input image

Fast R-CNN


Object category
Linear + softmax
Linear
Box offset
CNN
Per-Region Network
Regions of Interest (RoIs) from a proposal method
Crop + Resize features
"conv5" features
"Backbone" network: AlexNet, VGG, ResNet, etc
Run whole image through ConvNet
ConvNet
Input image


"Slow" R-CNN
SVMs
SVMs
SVMs
Conv Net
Conv Net
Conv Net
Input image

Region Proposal Network



Region Proposal Network

Input Image
(e.g. 3 x 640 x 480)

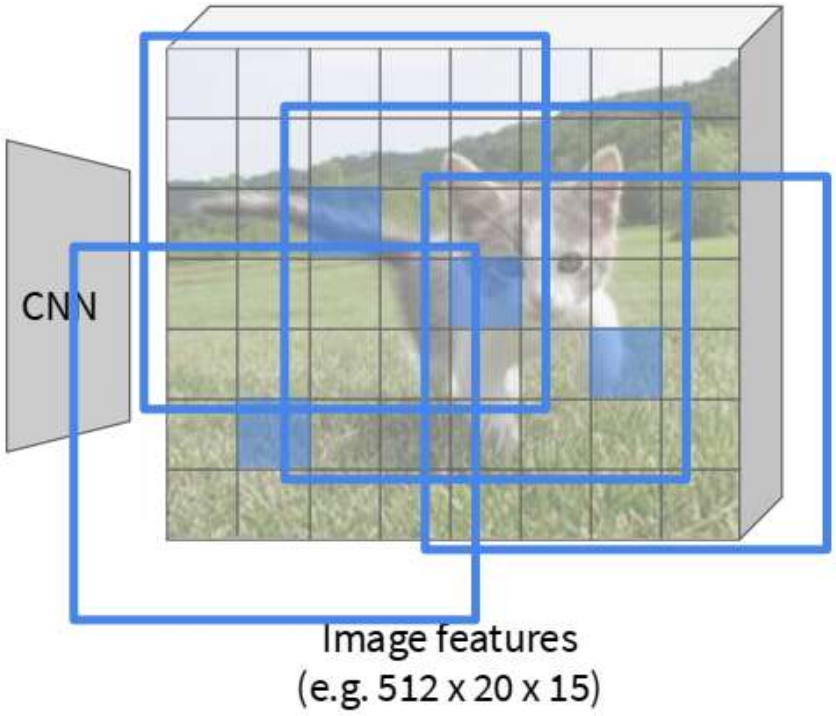


Image features
(e.g. 512 x 20 x 15)

Imagina un anchor box de
Tamaño fijo en cada punto del
Feature map

Region Proposal Network

Input Image
(e.g. 3 x 640 x 480)

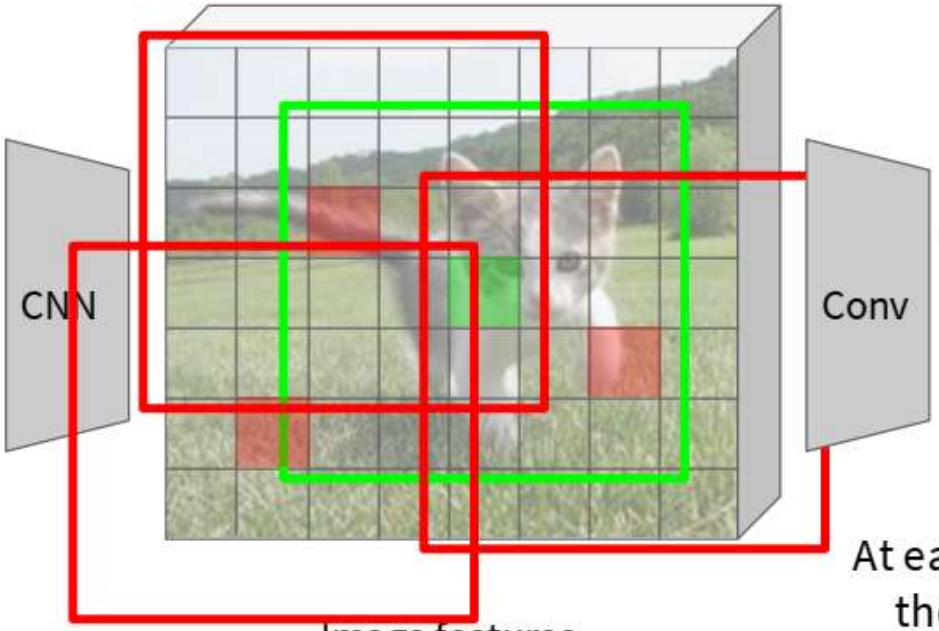


Image features
(e.g. 512 x 20 x 15)

At each point, predict whether the corresponding anchor contains an object (binary classification)

Region Proposal Network

Input Image
(e.g. 3 x 640 x 480)

CNN

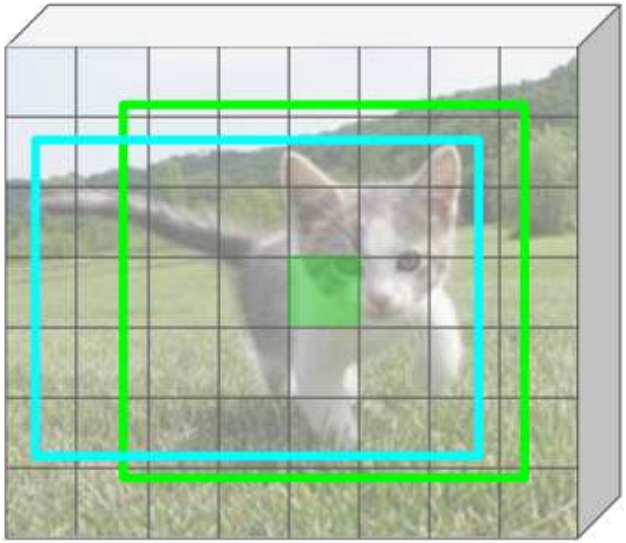


Image features
(e.g. 512 x 20 x 15)

Conv

Anchor is an object?
1 x 20 x 15

Box corrections
4 x 20 x 15

For positive boxes, also predict a corrections from the anchor to the ground-truth box (regress 4 numbers per pixel)

Region Proposal Network

En la práctica se usan K diferentes
Anchor boxes de diferentes tamaño
Y escalas en cada punto

Input Image
(e.g. 3 x 640 x 480)

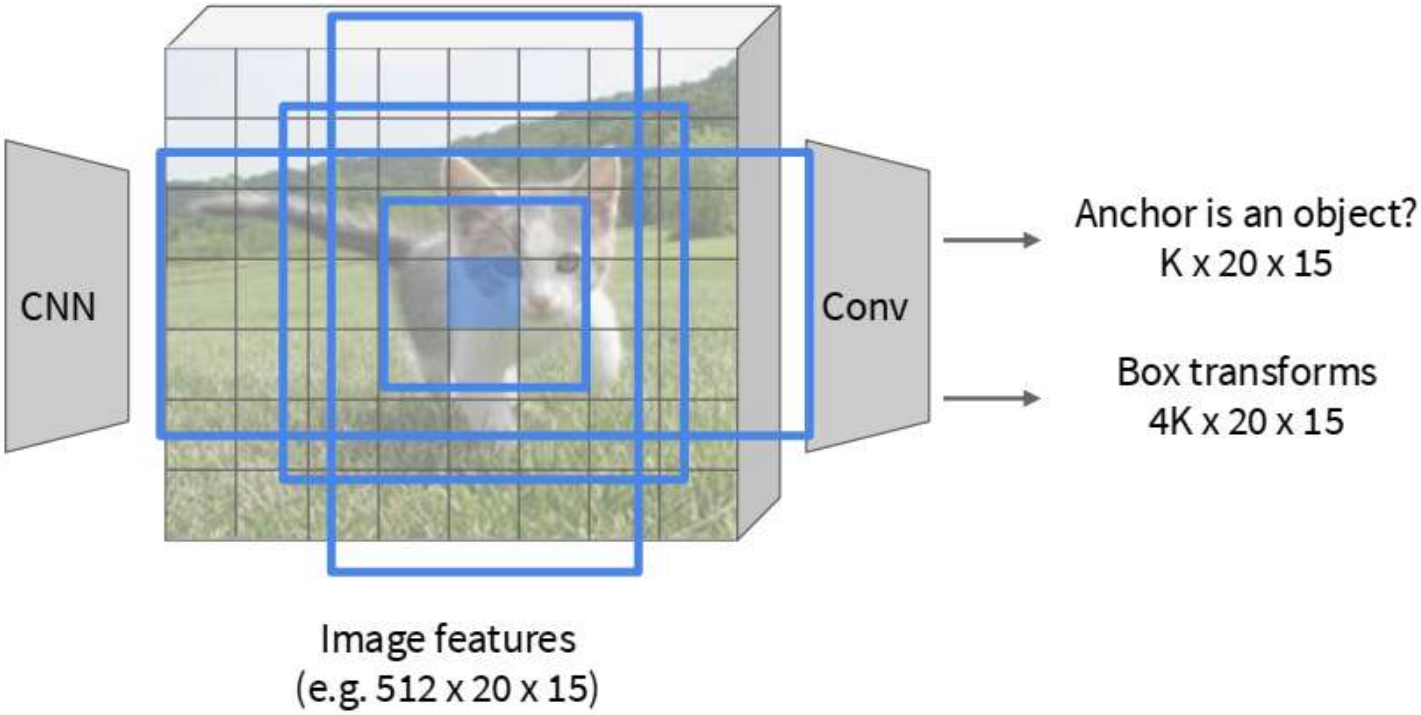


Image features
(e.g. 512 x 20 x 15)

Region Proposal Network

En la práctica se usan K diferentes
Anchor boxes de diferentes tamaño
Y escalas en cada punto

Input Image
(e.g. 3 x 640 x 480)

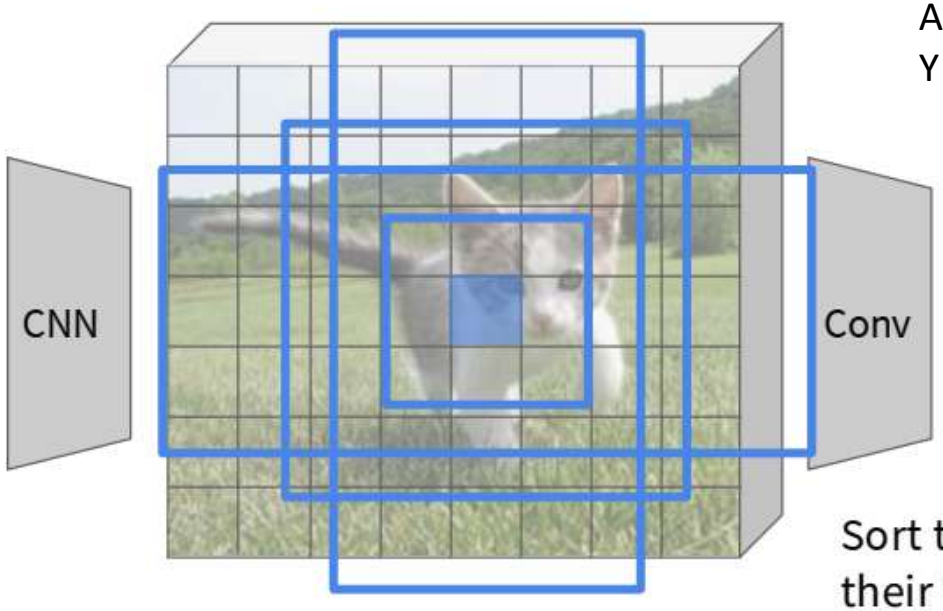


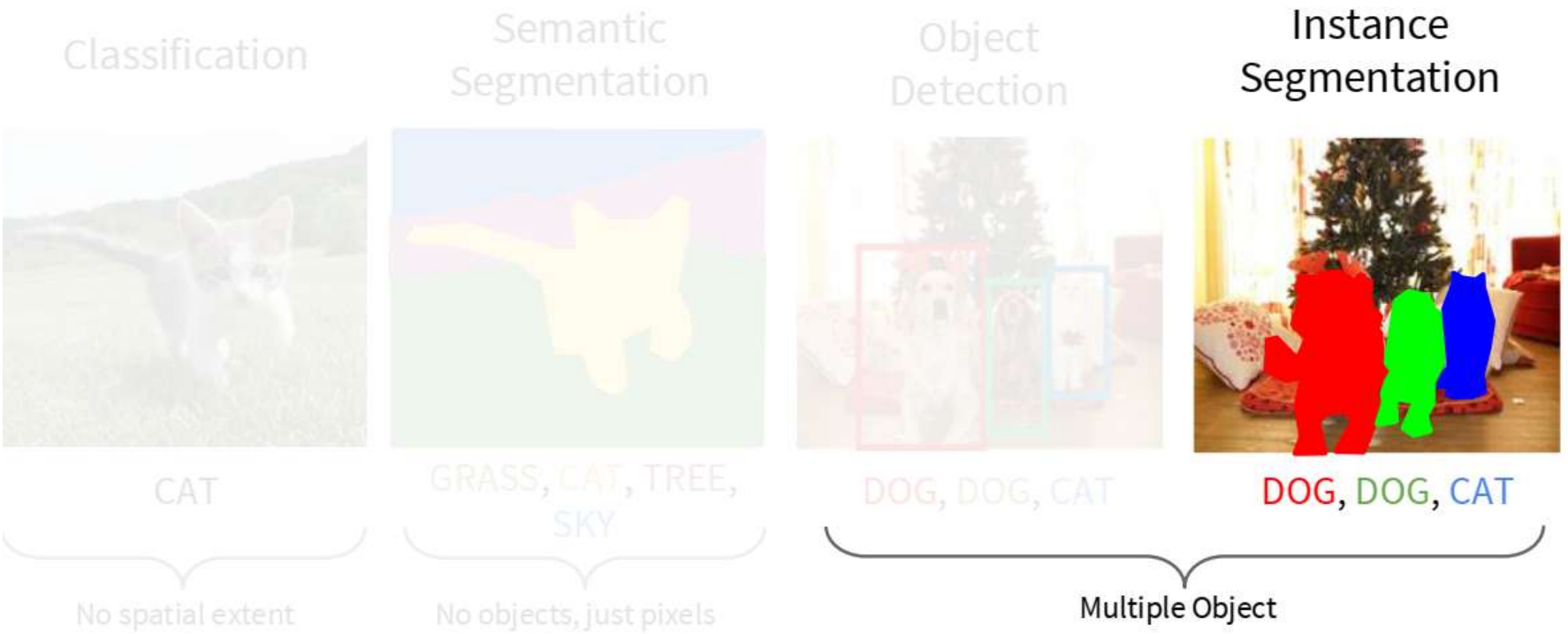
Image features
(e.g. 512 x 20 x 15)

Anchor is an object?
K x 20 x 15

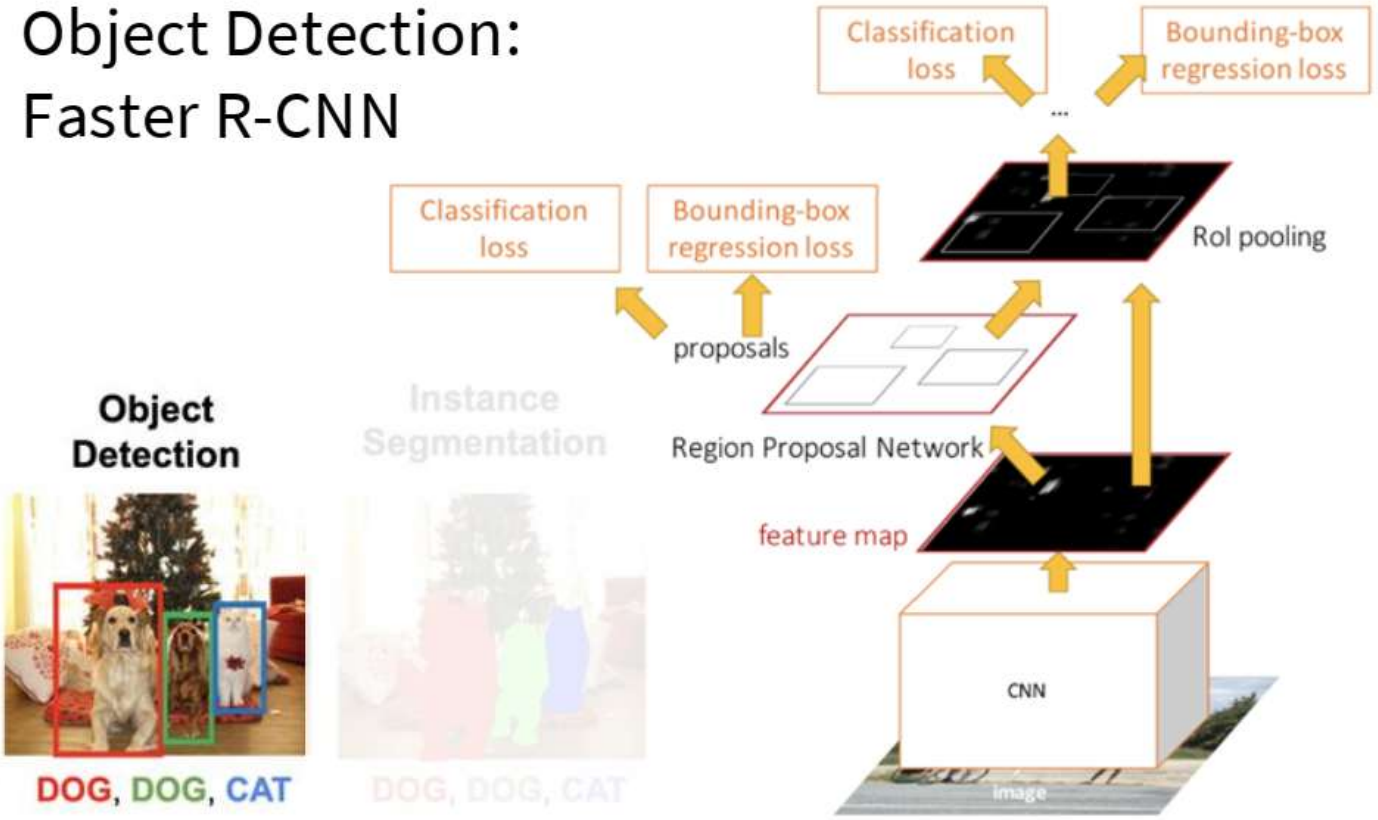
Box transforms
4K x 20 x 15

Sort the K*20*15 boxes by their "objectness" score, take top ~300 as our proposals

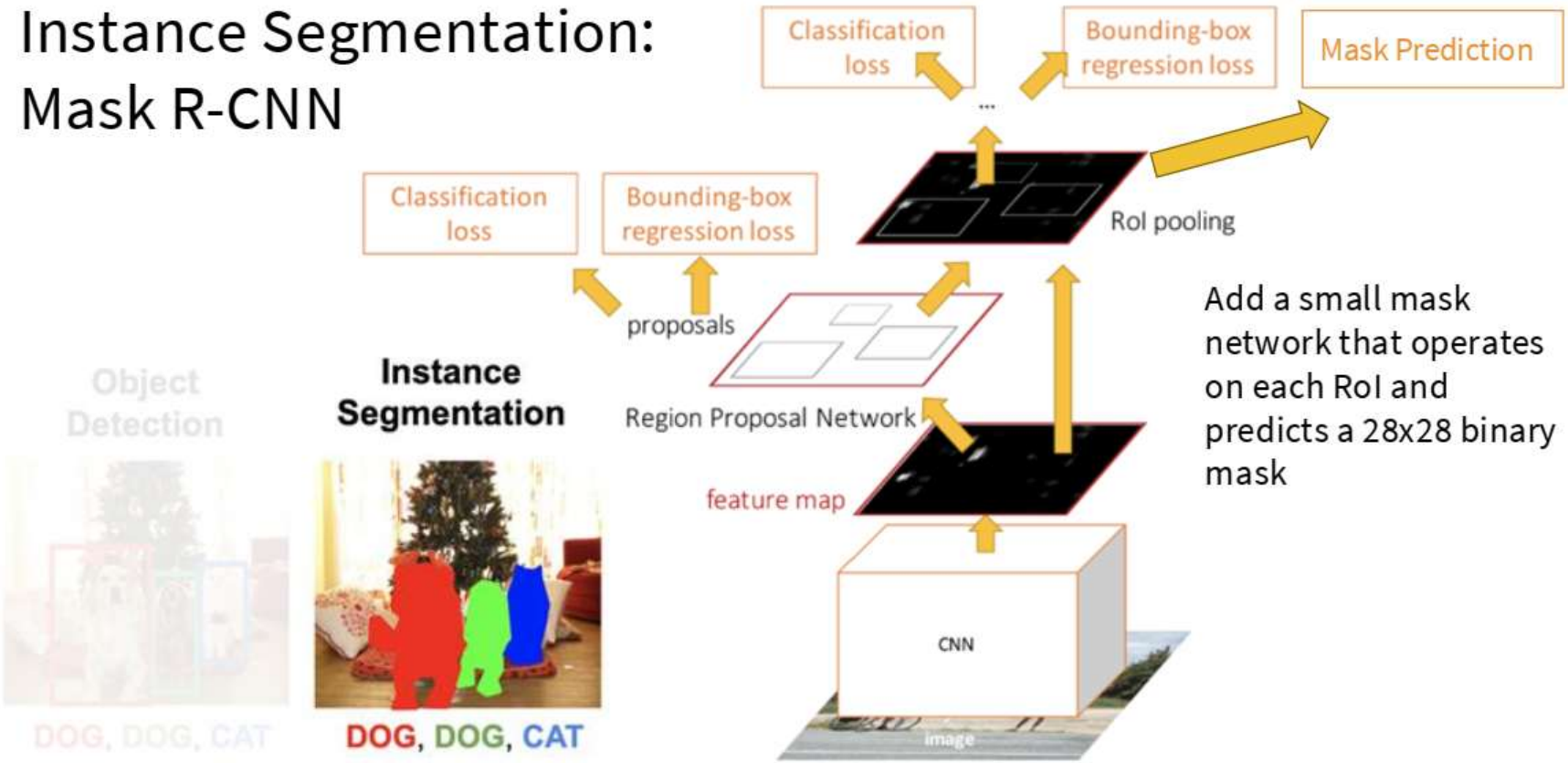
Instance Segmentation



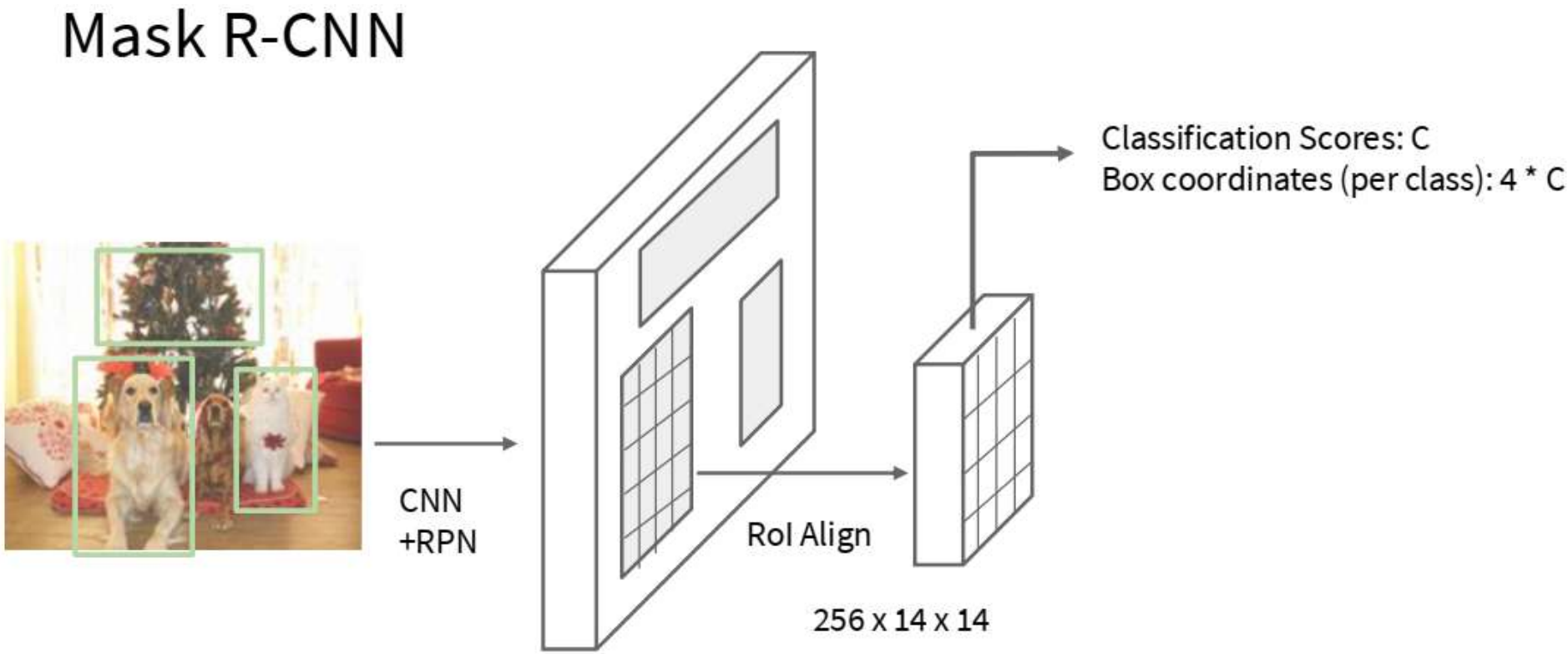
Instance Segmentation



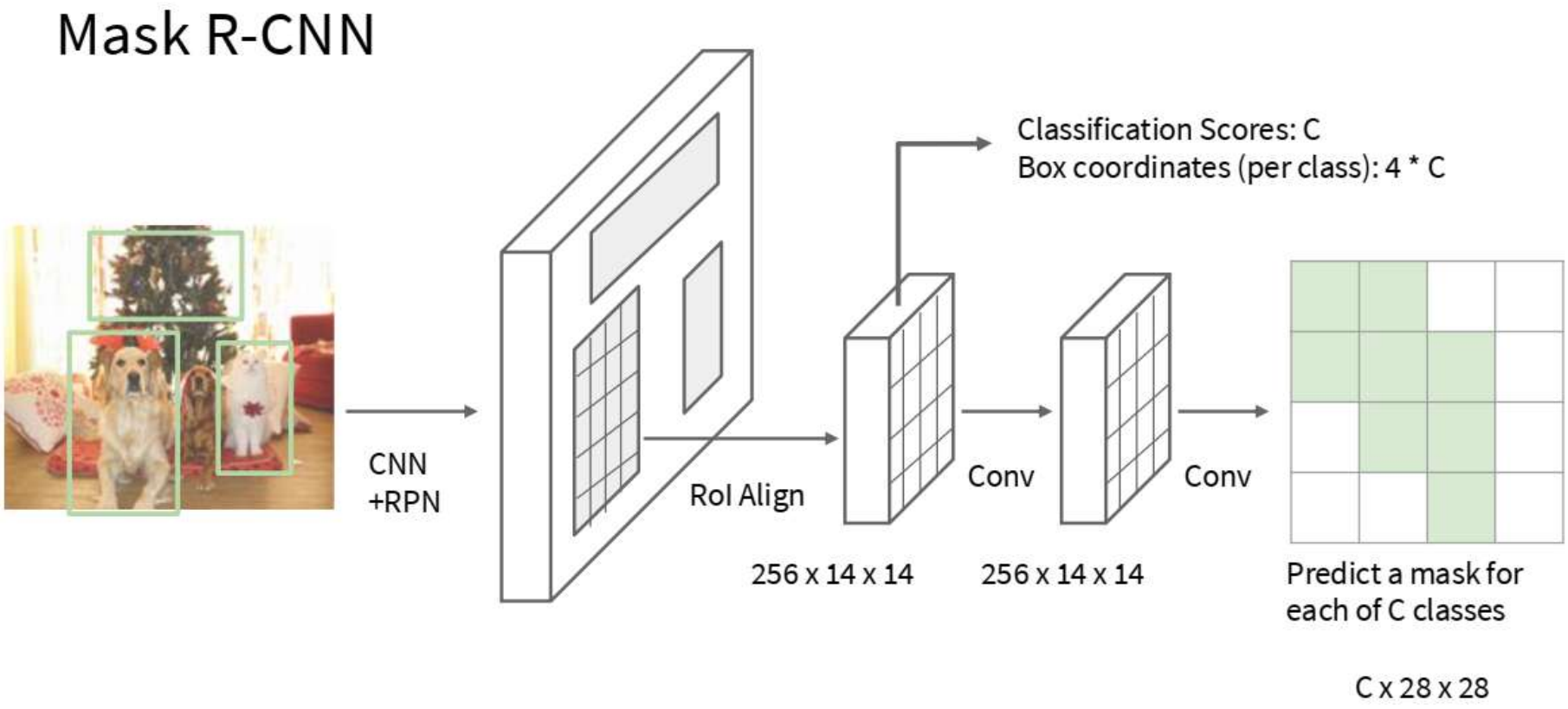
Instance Segmentation



Instance Segmentation

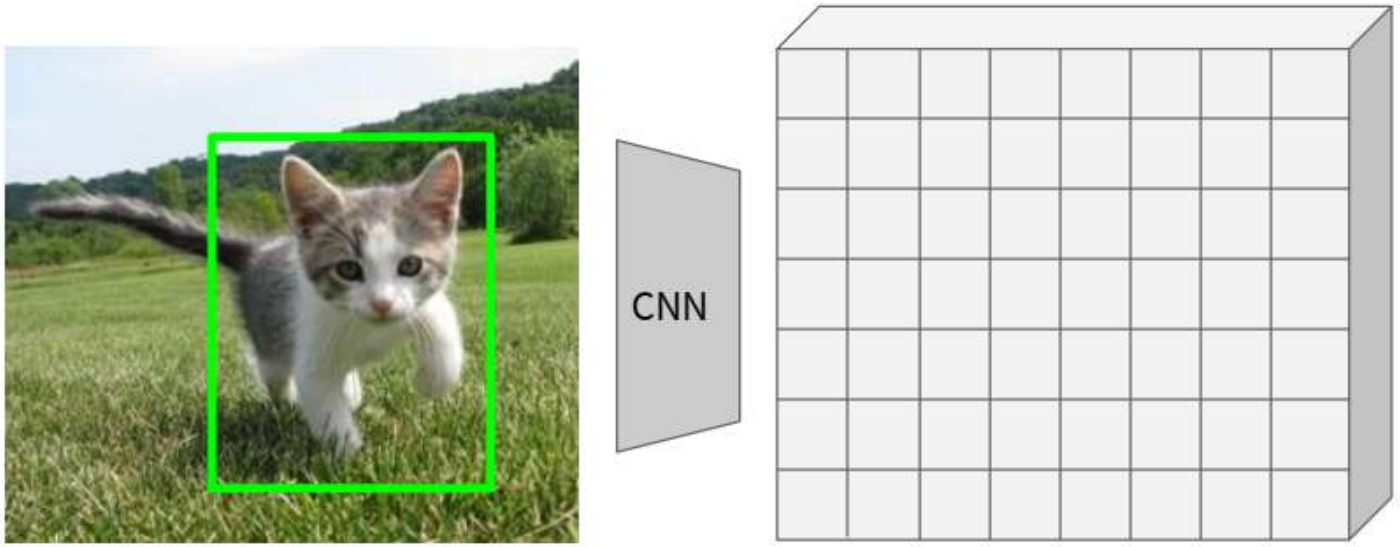


Instance Segmentation

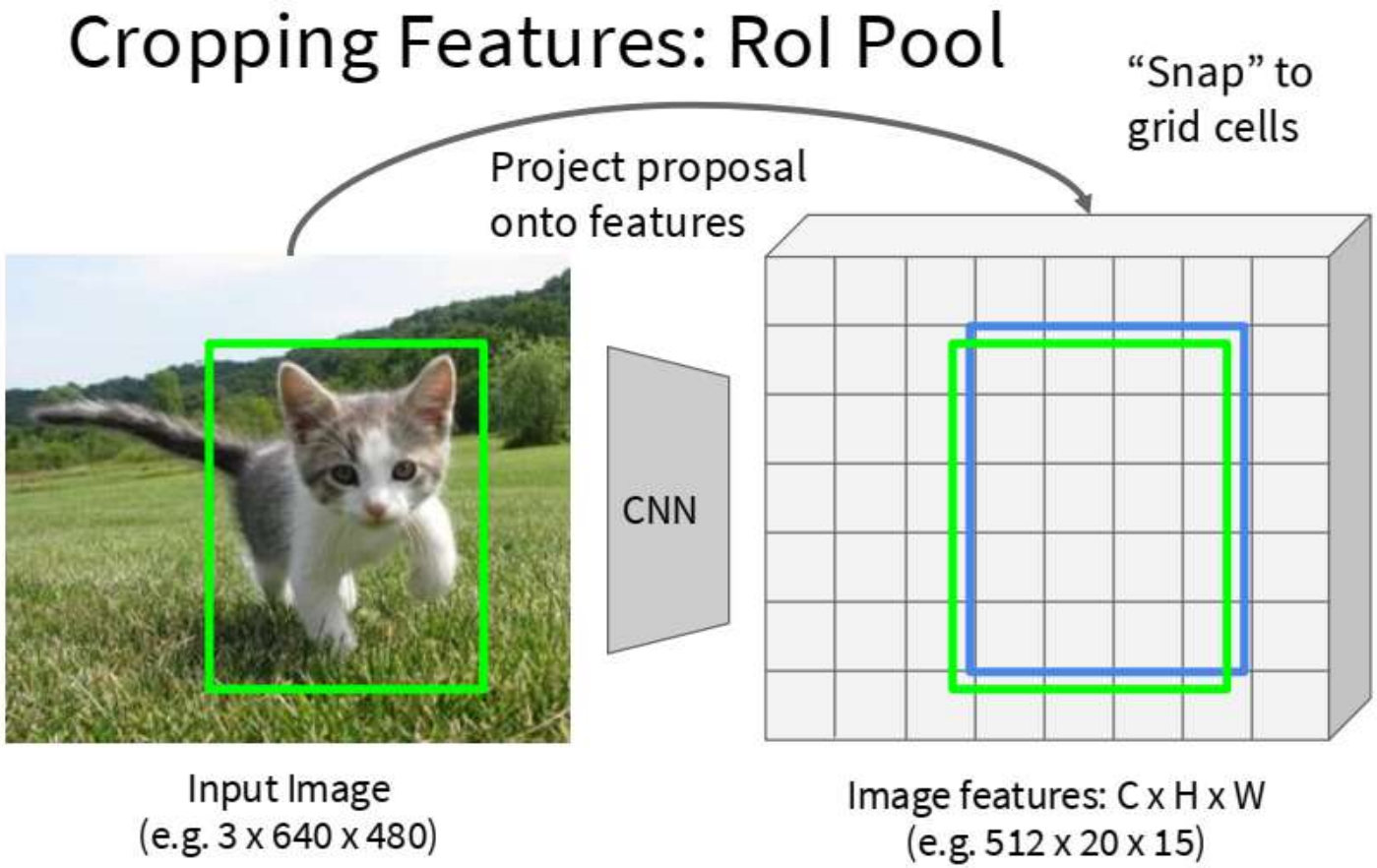


Instance Segmentation

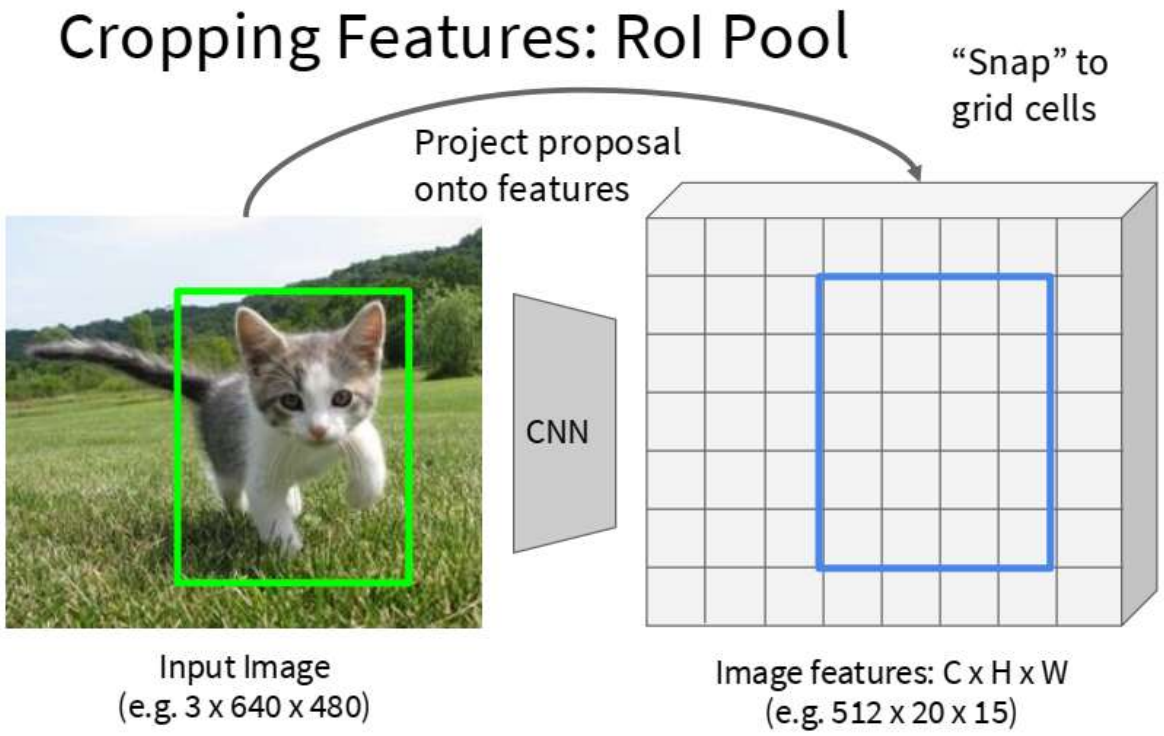
ROI Pool



Instance Segmentation



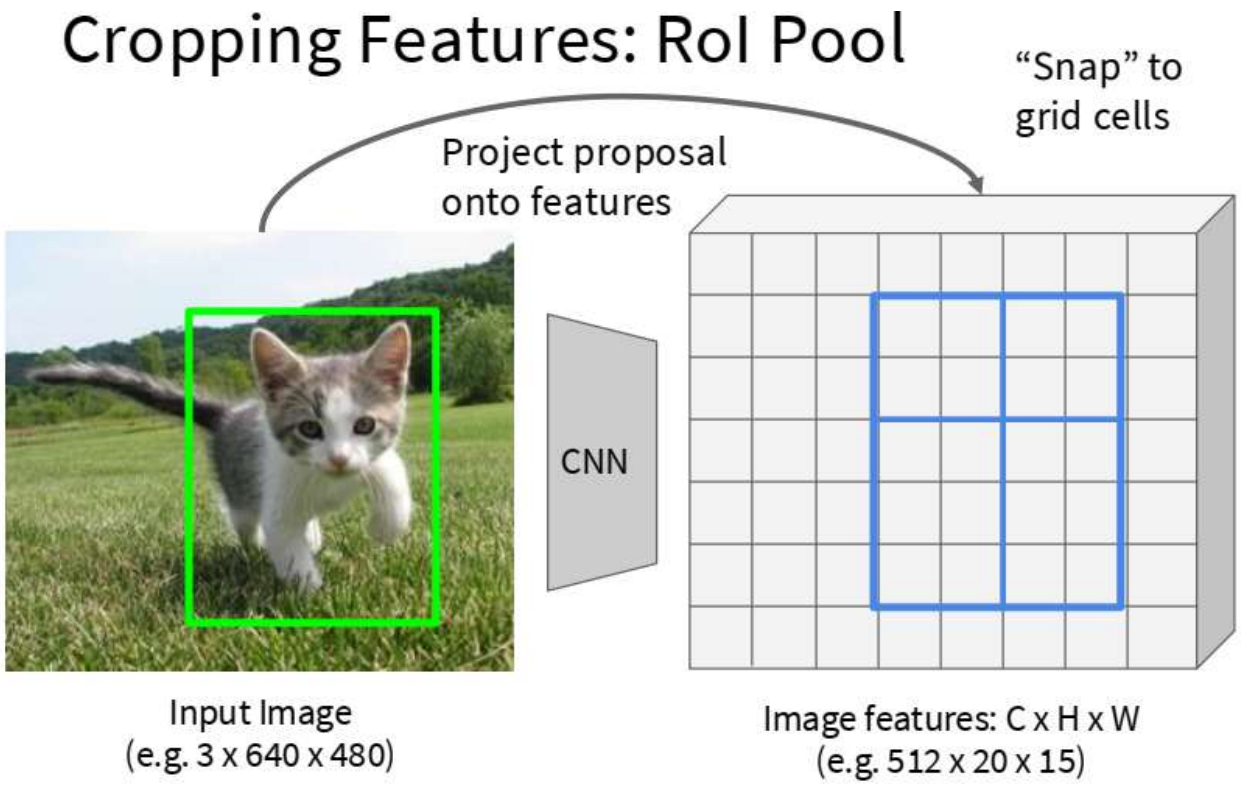
Instance Segmentation



Cómo transformamos la región 512x5x4 a 512x2x2, por ejemplo?

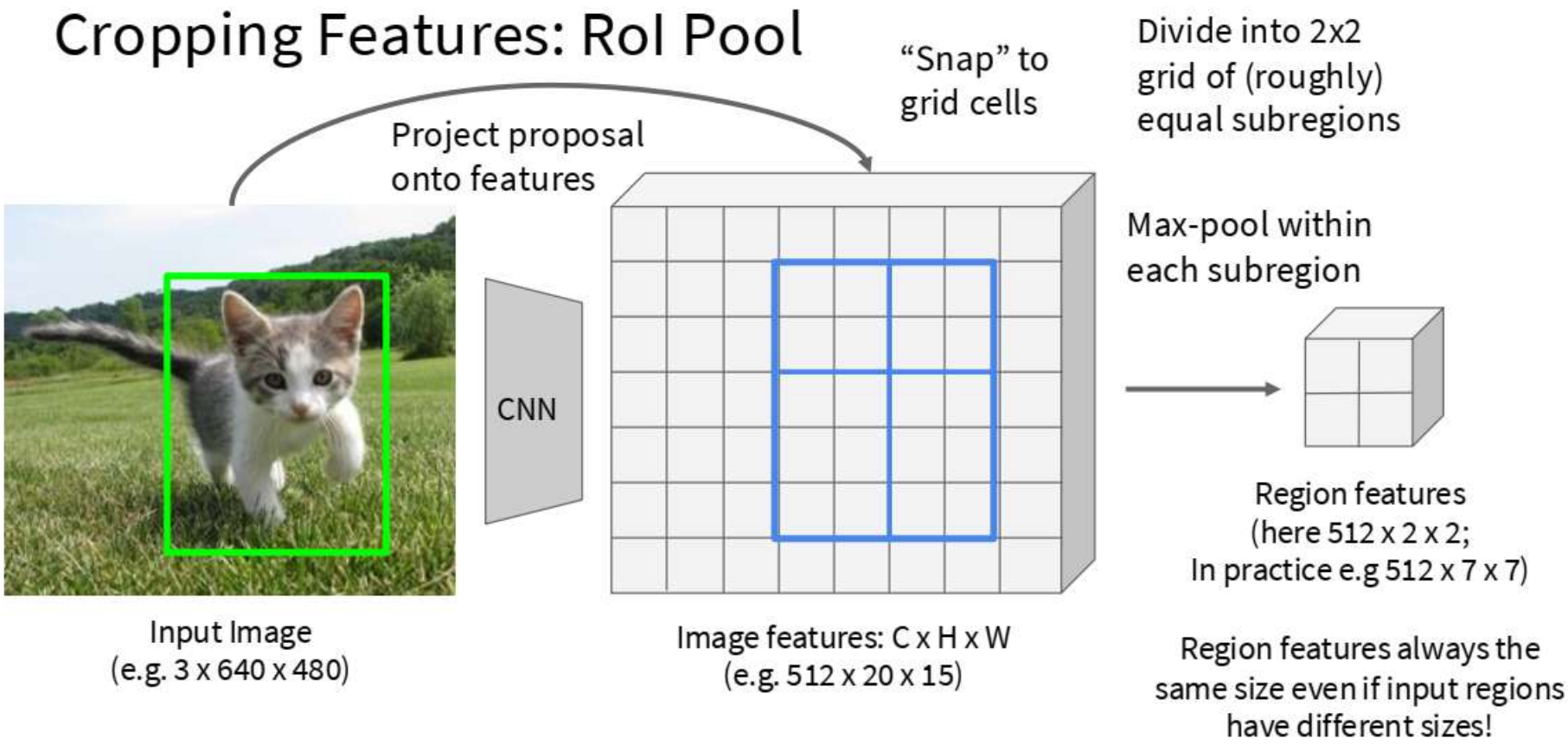
Instance Segmentation

Dividir en grid de 2x2 de regiones más o menos iguales

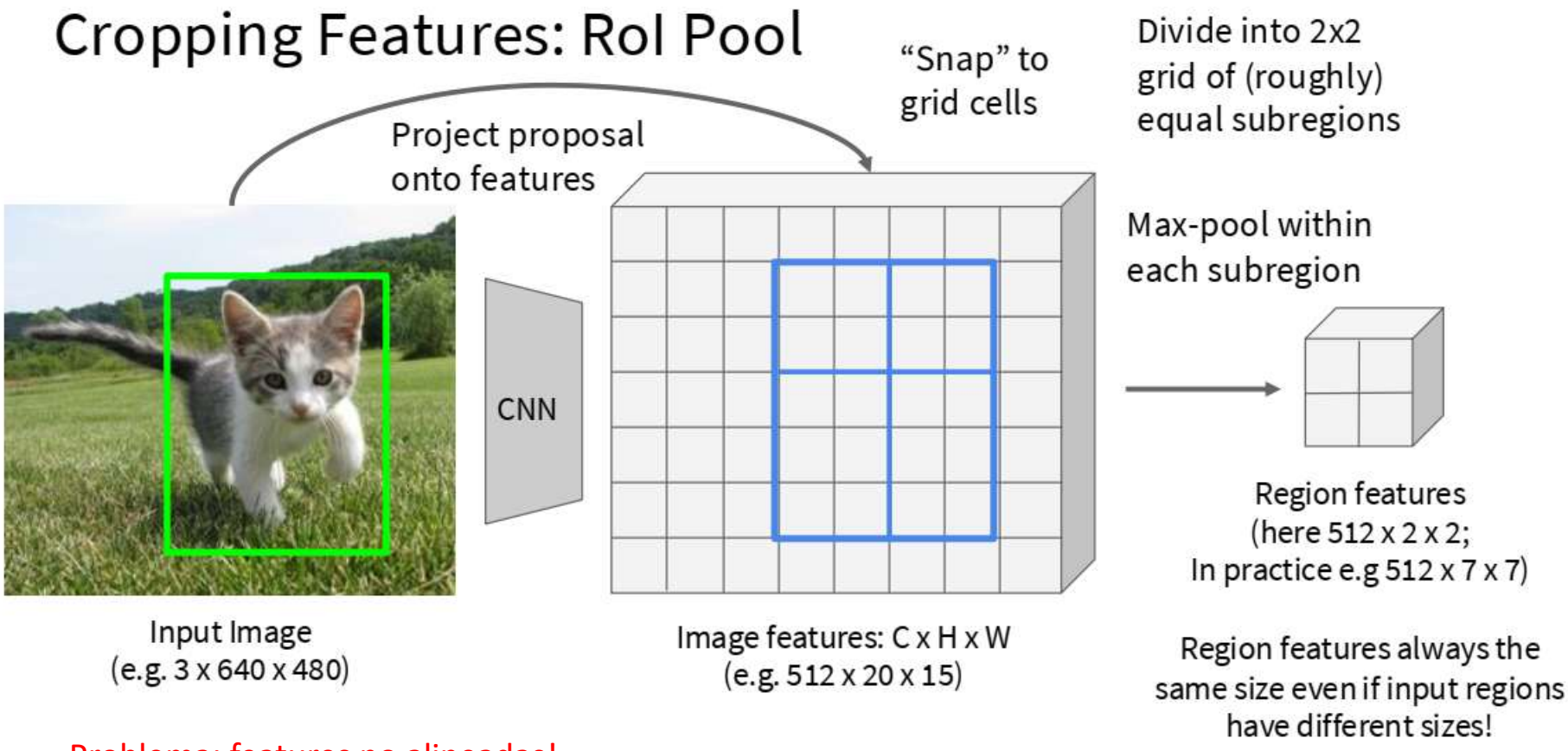


Cómo transformamos la región 512x5x4 a 512x2x2, por ejemplo?

Instance Segmentation

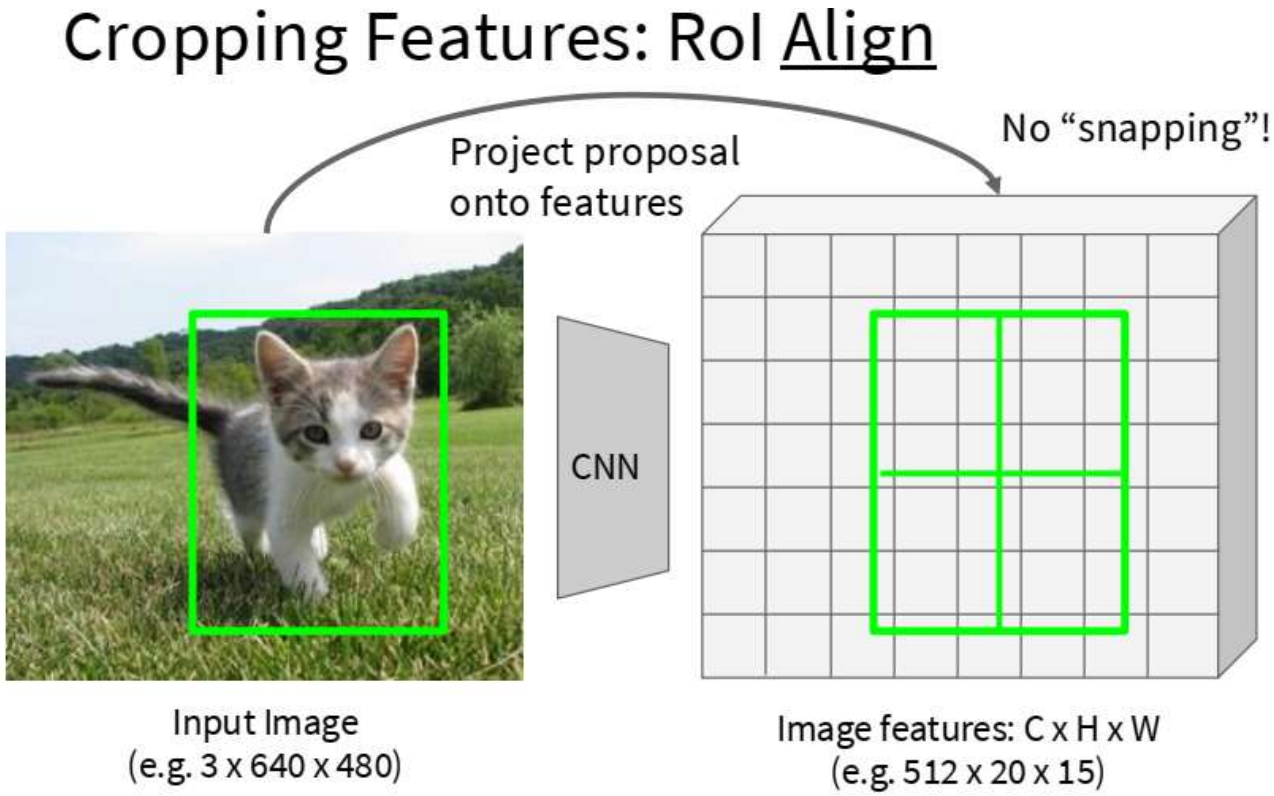


Instance Segmentation

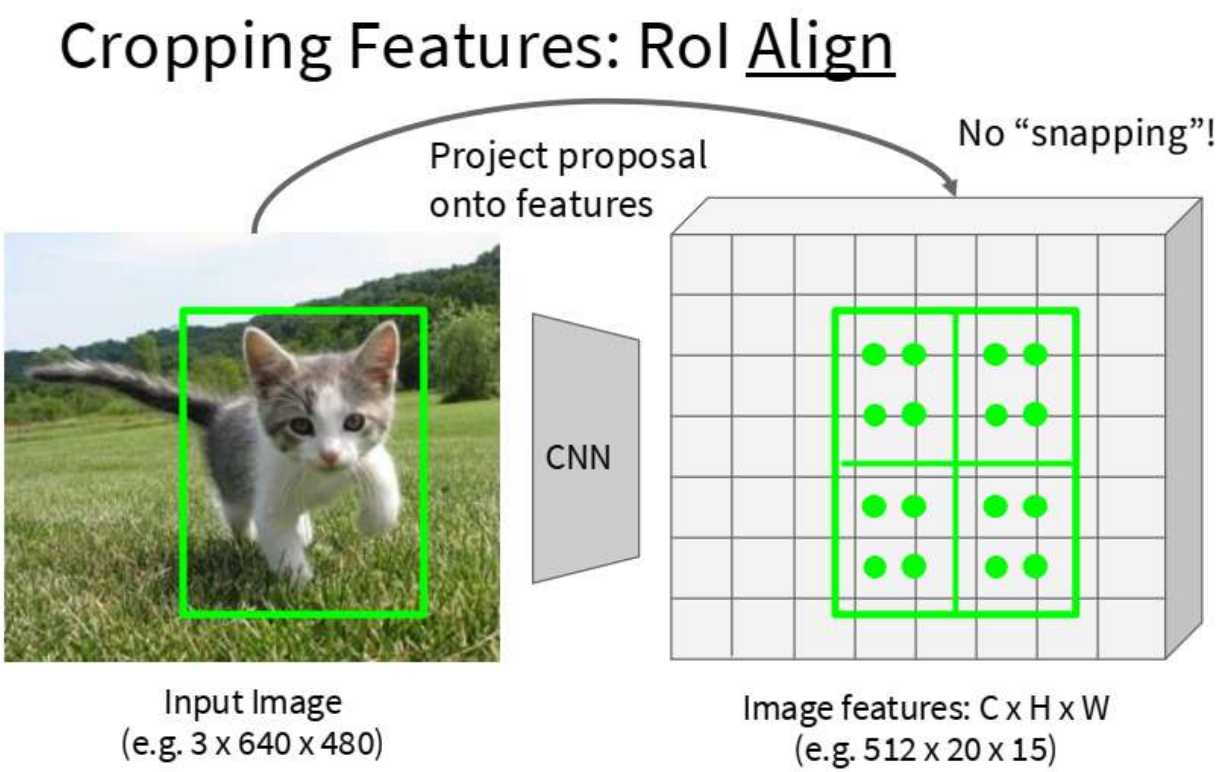


Problema: features no alineadas!

Instance Segmentation

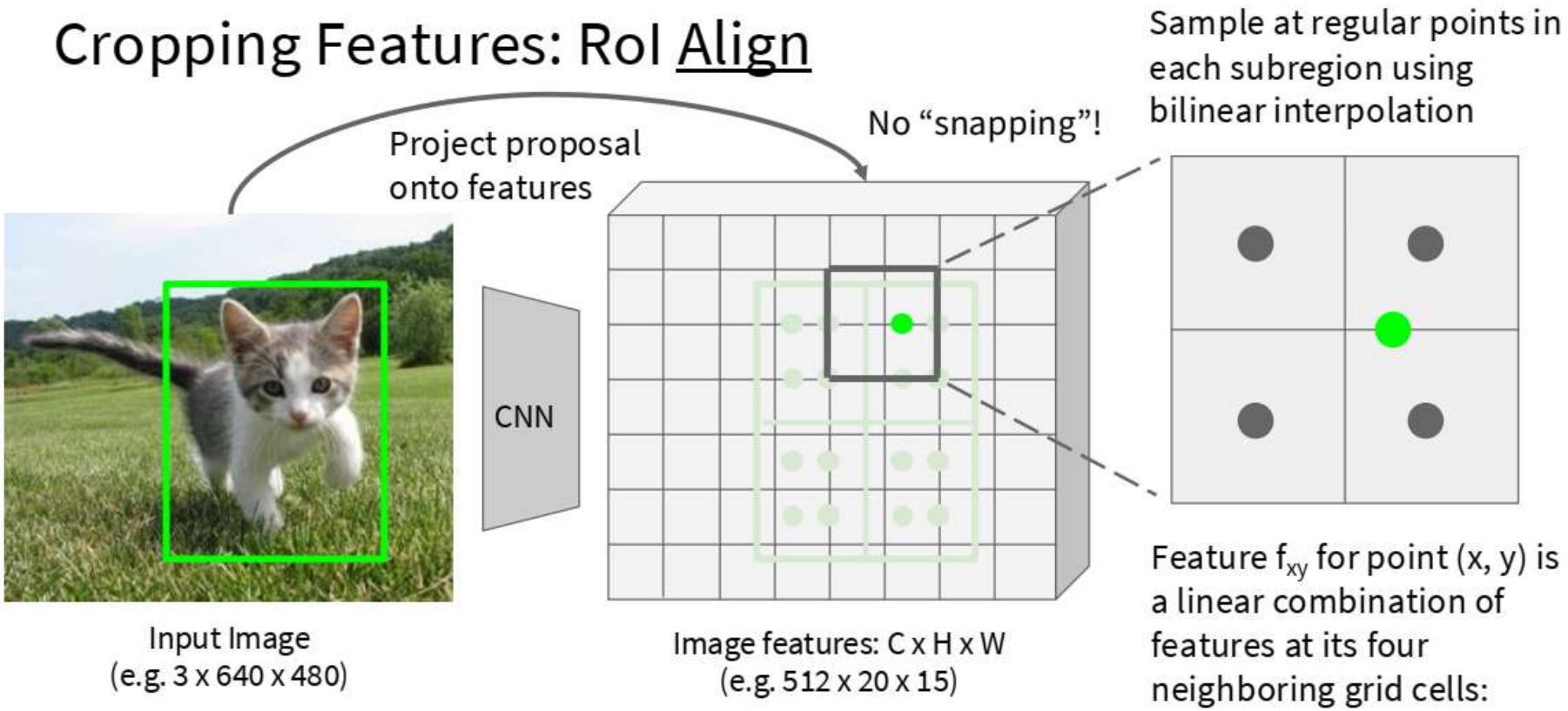


Instance Segmentation

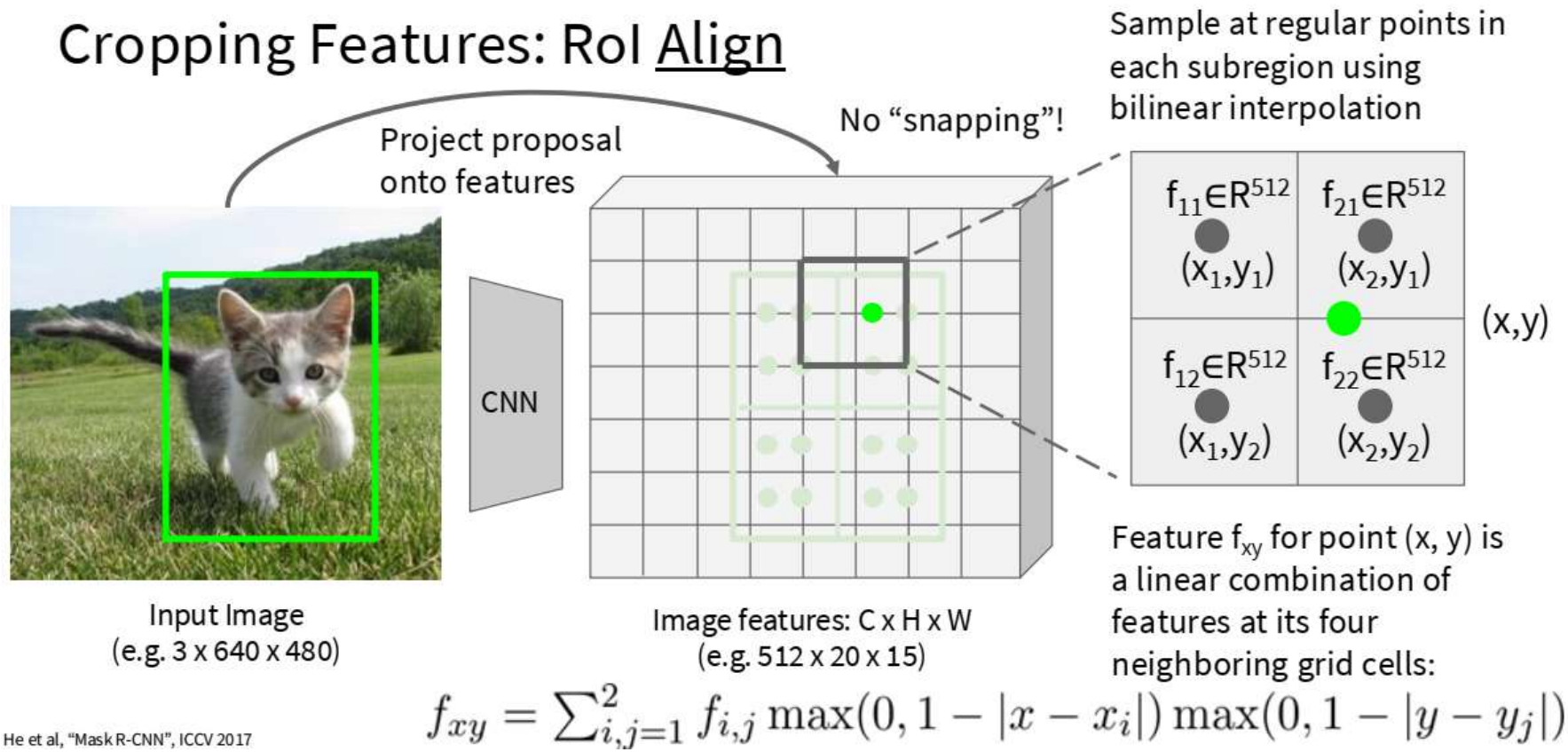


Sample at regular points in each subregion using bilinear interpolation

Instance Segmentation

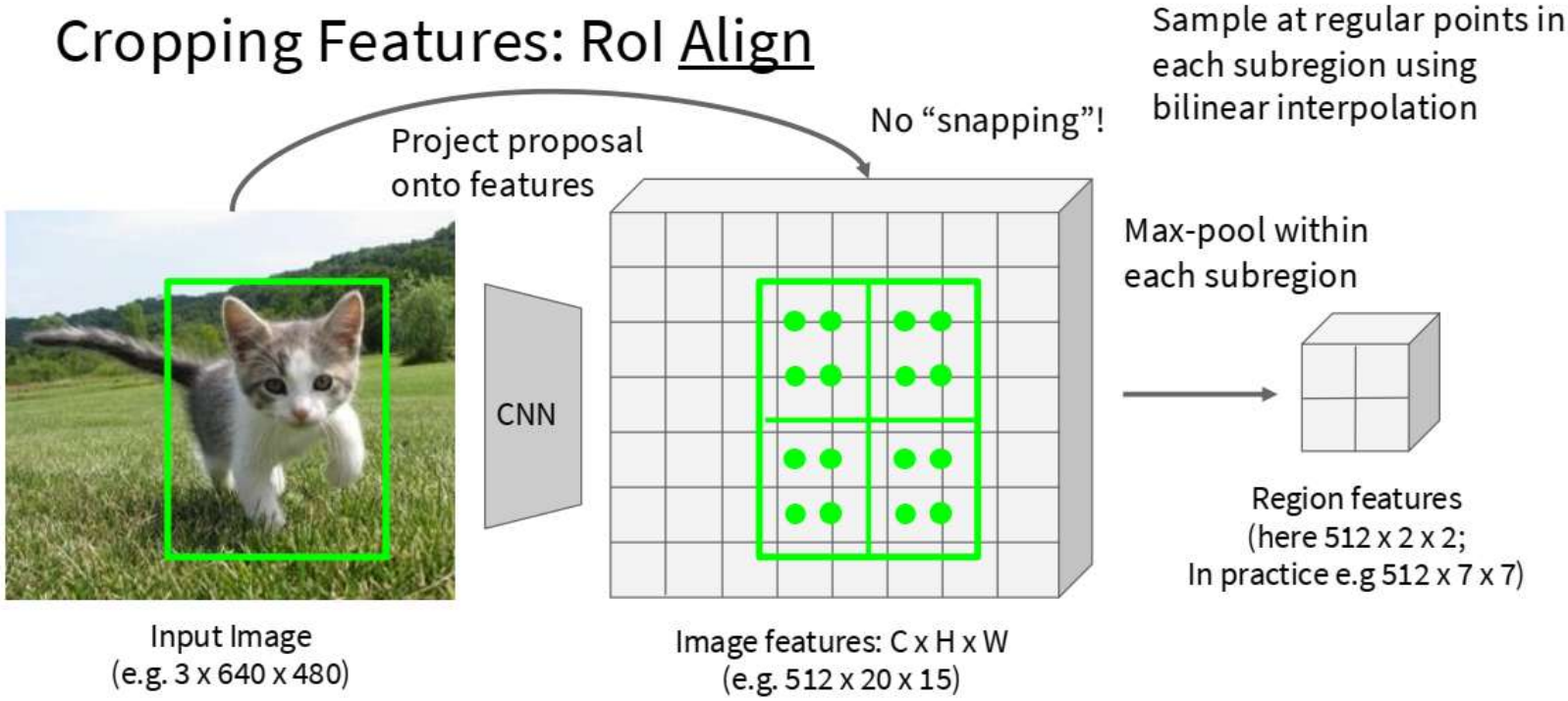


Instance Segmentation

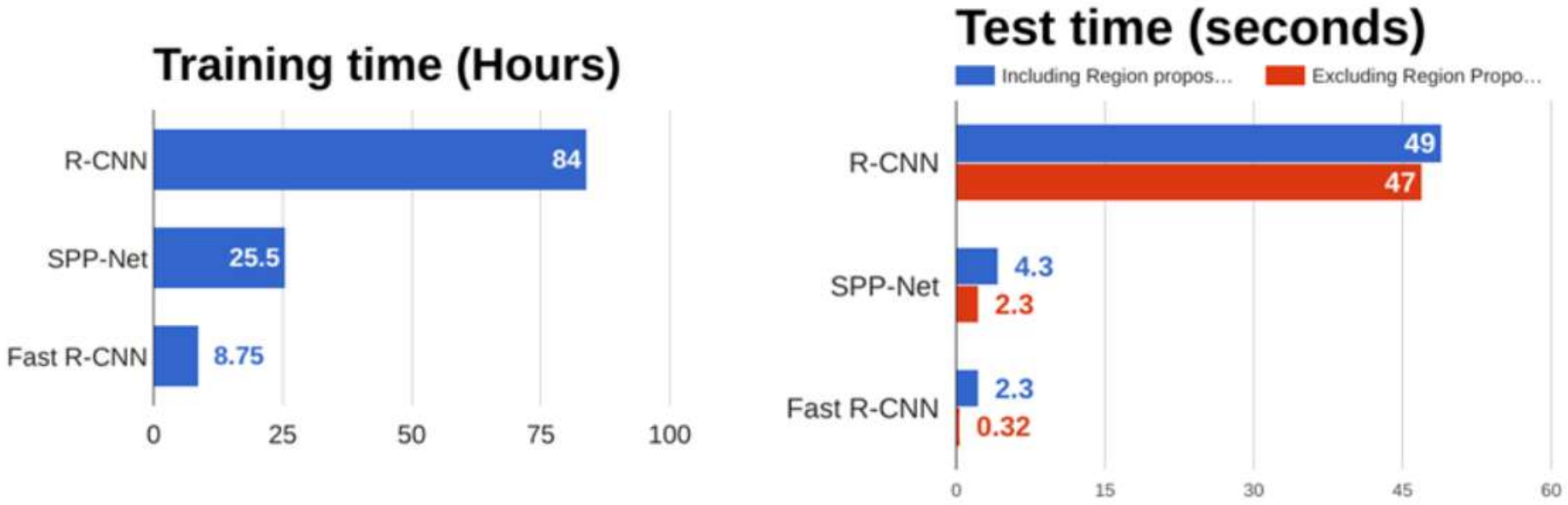


He et al, "Mask R-CNN", ICCV 2017

Instance Segmentation

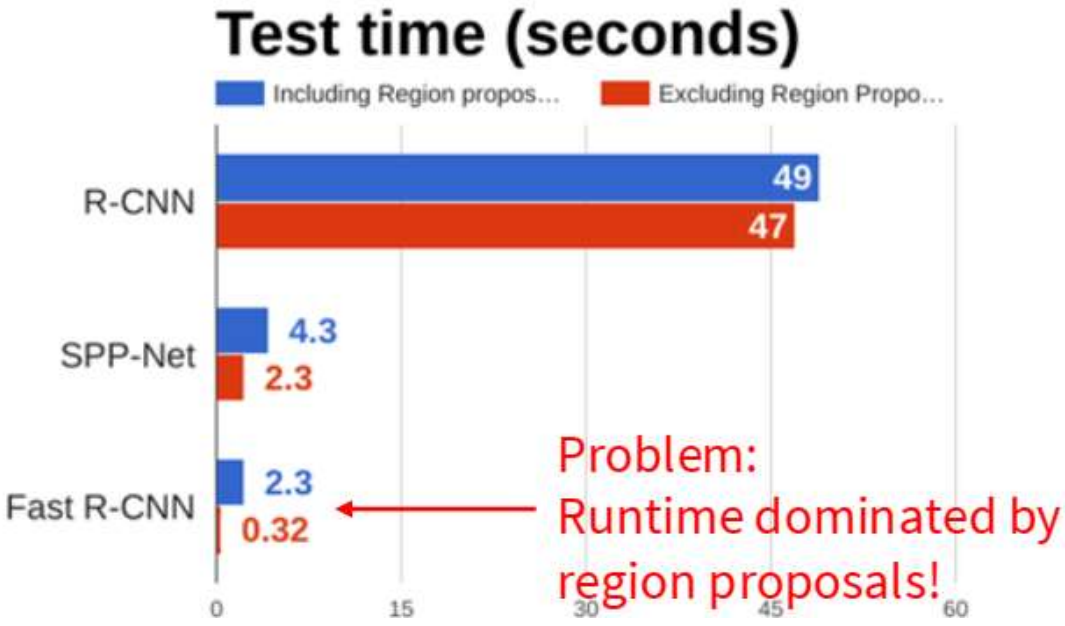


Timing


Training time (Hours)
R-CNN
84
SPP-Net
25.5
Fast R-CNN
8.75
0
25
50
75
100
Test time (seconds)
Including Region propos…
Excluding Region Propo…
R-CNN
49
47
SPP-Net
4.3
2.3
Fast R-CNN
2.3
0.32
0
15
30
45
60

Timing


Training time (Hours)
R-CNN
84
SPP-Net
25.5
Fast R-CNN
8.75
0
25
50
75
100

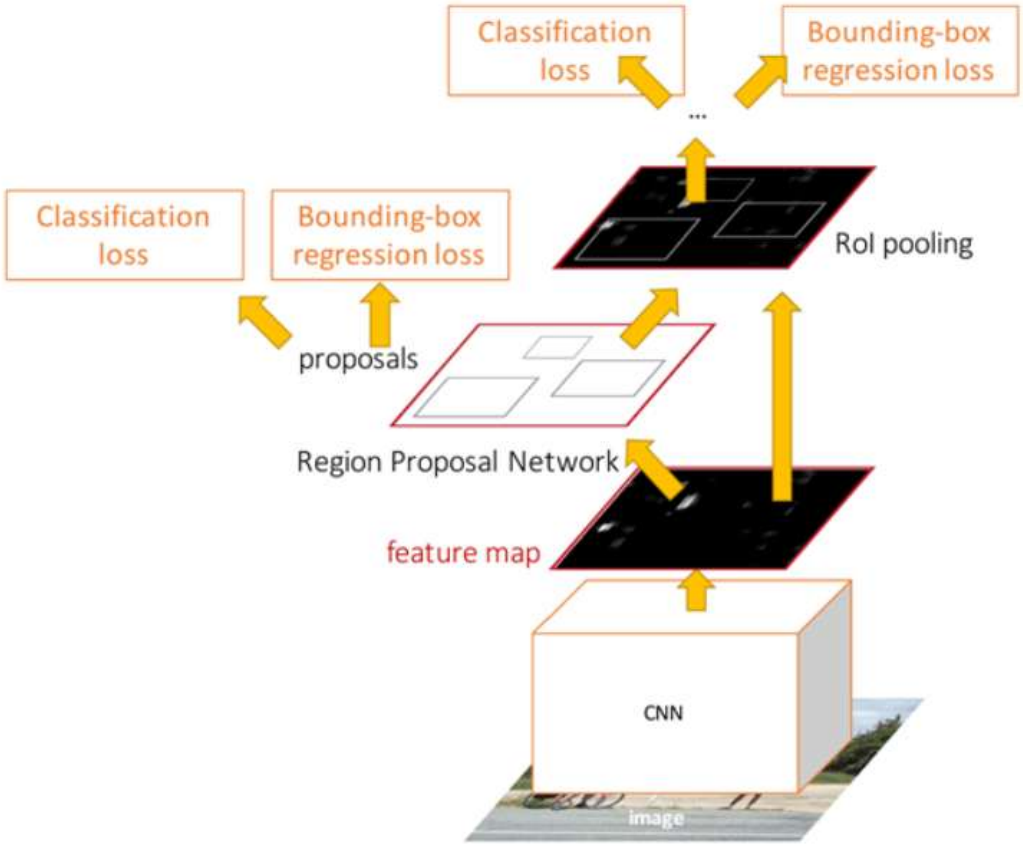

Test time (seconds)
Including Region propos...
Excluding Region Propo...
R-CNN
49
47
SPP-Net
4.3
2.3
Fast R-CNN
2.3
0.32
0
15
30
45
60
Problem:
Runtime dominated by
region proposals!

Faster R-CNN

Que la CNN se encargue de los proposals!

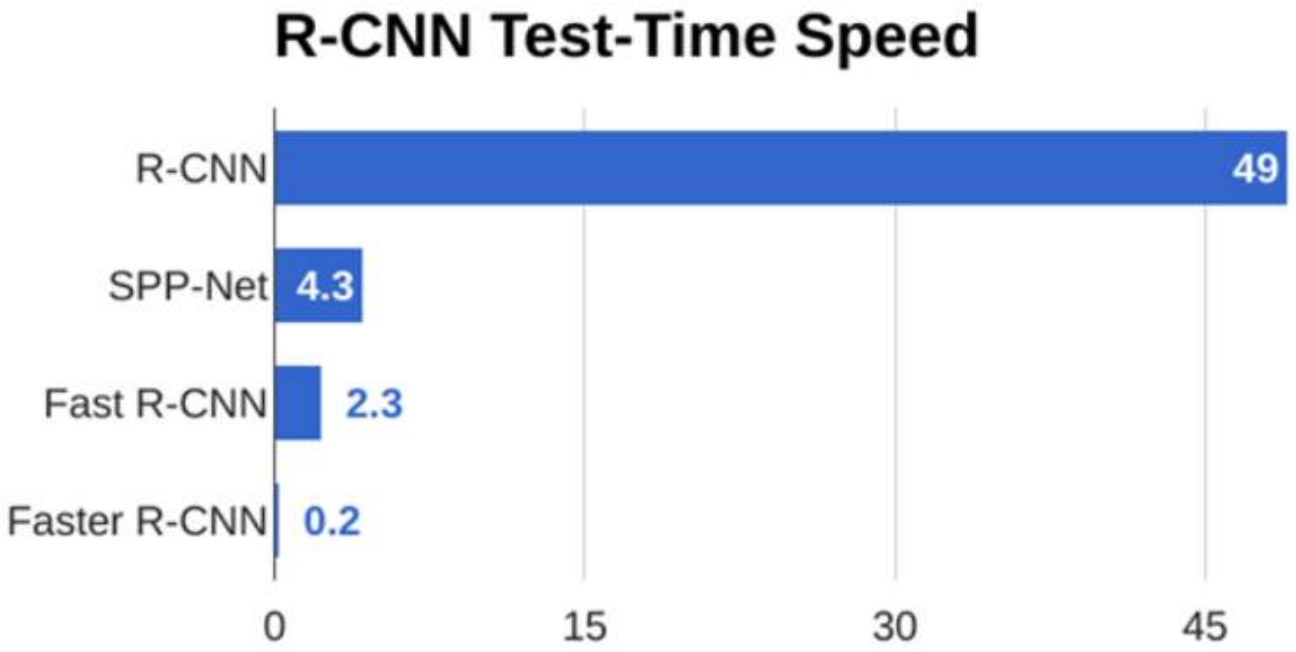
Se entrena en conjunto:

- RPN clasifica objeto/no objeto
- RPN regresión coordenadas box
- Score de clasificación final
- Coordenadas de box final



Faster R-CNN

Que la CNN se encargue de los proposals!



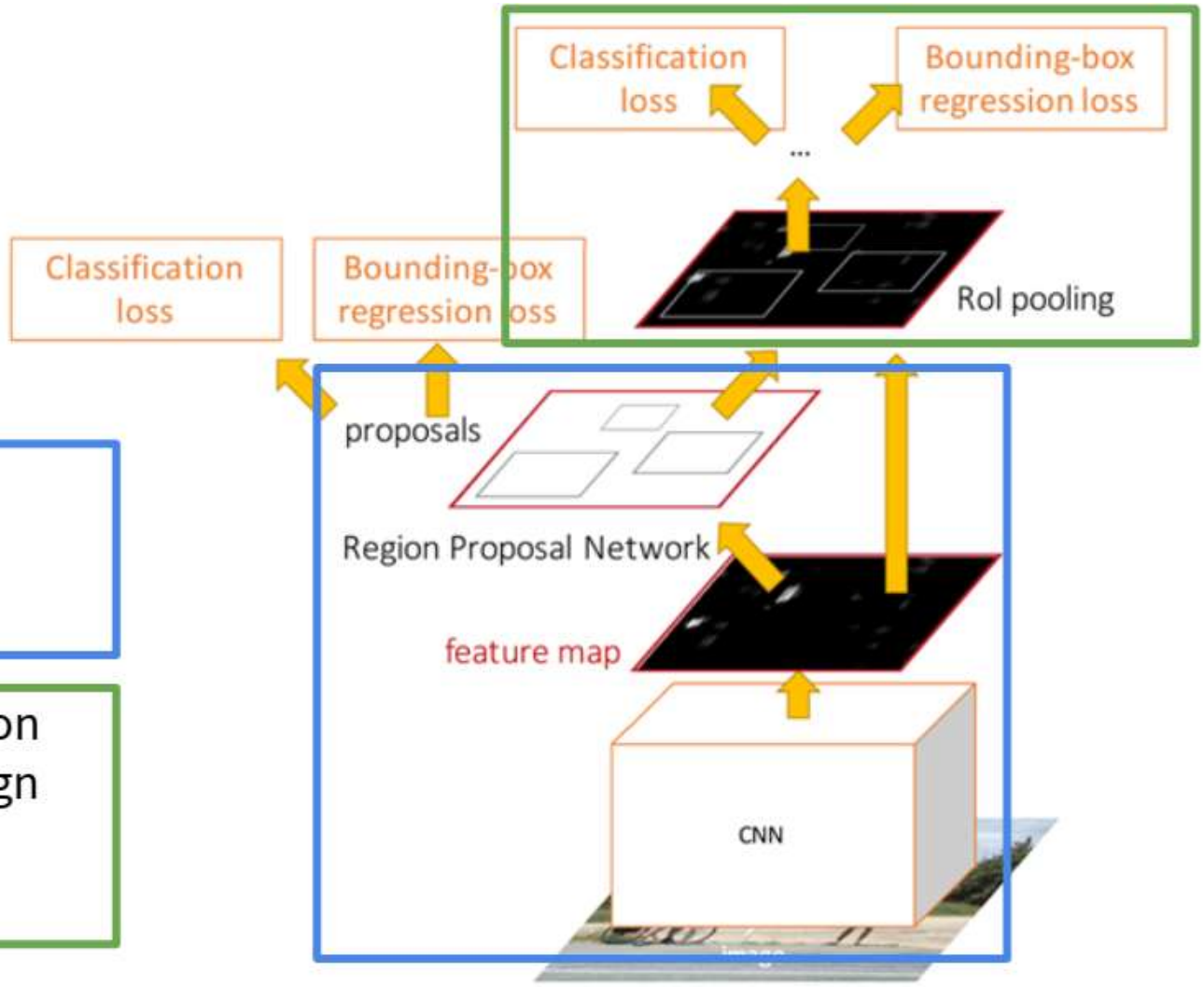
Faster R-CNN:

Make CNN do proposals!

Faster R-CNN is a
Two-stage object detector

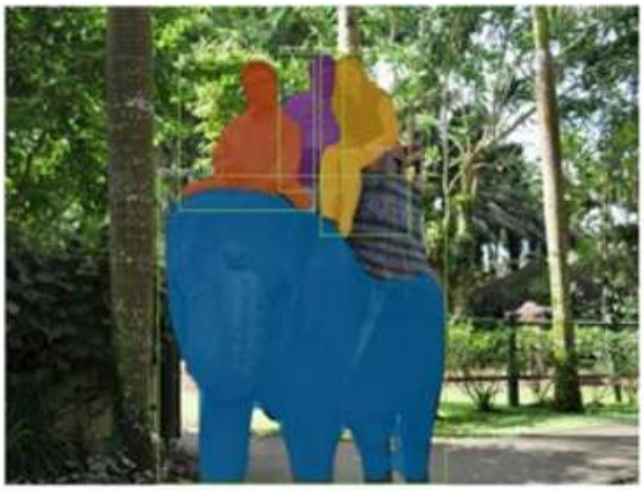
First stage: Run once per image
- Backbone network
- Region proposal network

Second stage: Run once per region
- Crop features: RoI pool / align
- Predict object class
- Prediction bbox offset



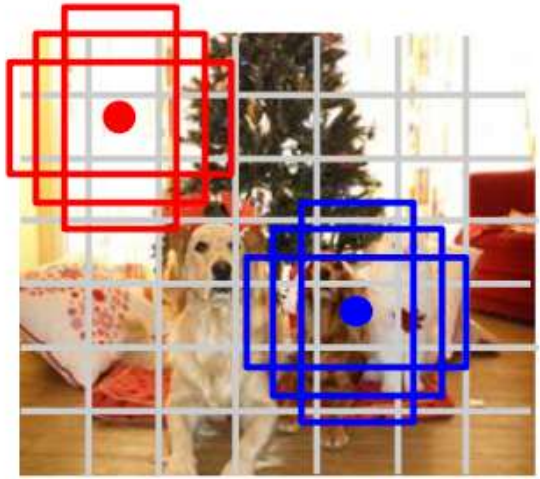
Instance Segmentation

Instance Segmentation


Detección de Objetos Single-Stage

Input image
3 x H x W



Divide image into grid
7 x 7

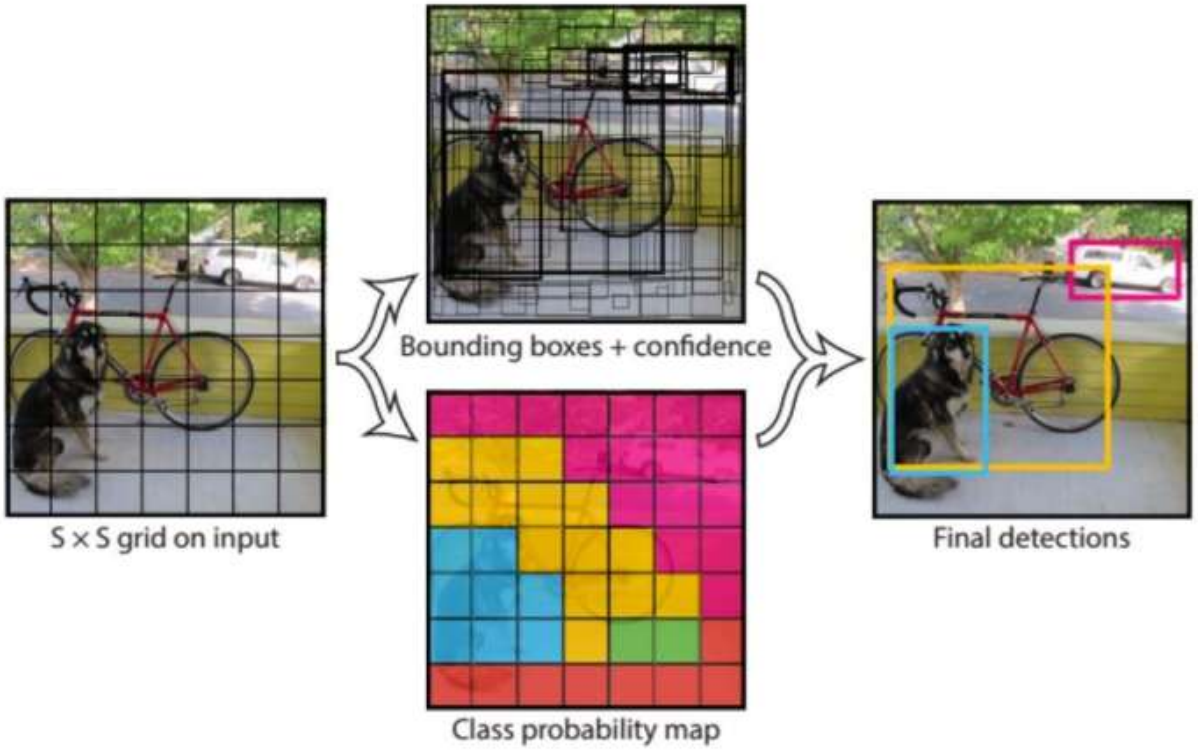
Image a set of base boxes
centered at each grid cell
Here B = 3

Dentro de cada celda:

- Regresión de cada B base box a un box final con 5 números (dx, dy, dh, dw, confidence)
- Predecir scores para cada una de las C clases (background es una clase).
- Parece una RPN, pero específico para las clases

Output:
7 x 7 x (5 * B + C)

YOLO (You Only Look Once)



YOLO (You Only Look Once)

SxS Grid

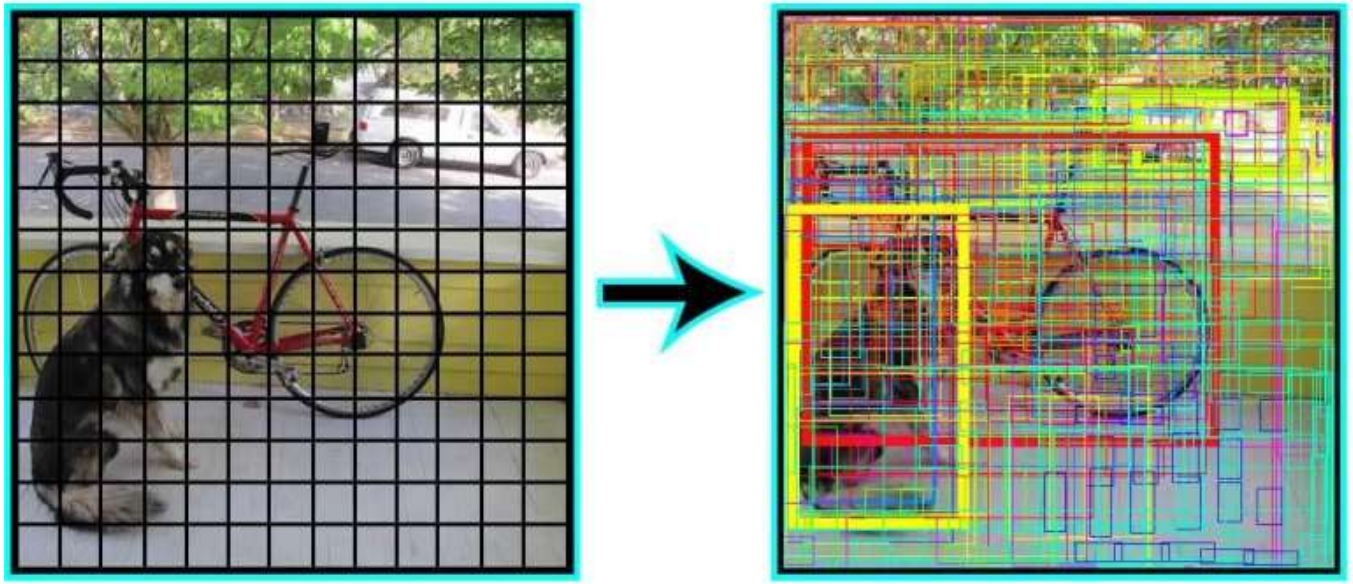
YOLO (You Only Look Once)

SxS Grid

Para cada box generar:

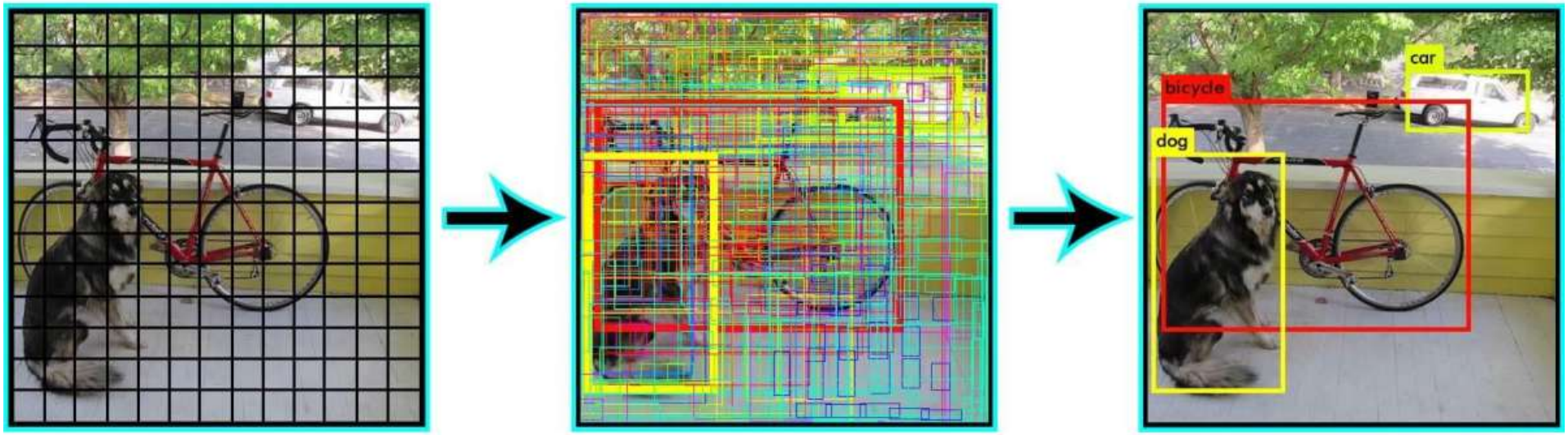
- P(object): probabilidad de que el box contenga un objeto
- B bounding boxes (x, y, d, w)
- P(class): probabilidad de pertenecer a una clase

YOLO (You Only Look Once)



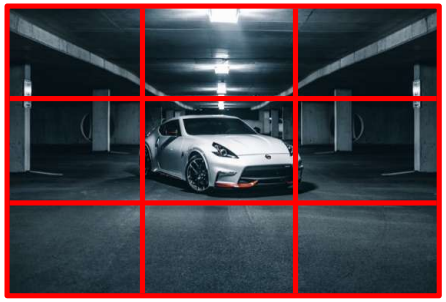
Muchos bounding boxes con
Diferentes probabilidades

YOLO (You Only Look Once)



Yolo (You Only Look Once)

- Imagen se particiona en bloques
- Cada bloque es responsable de sus detecciones



Clases
1. Person
2. Car
3. Motorbike

y is 3 x 3 x 2 x 8

$y =$

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

Hay objeto?

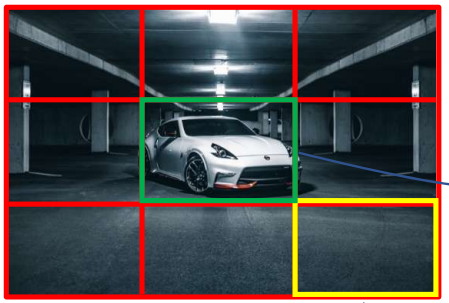
Bounding box

Prob. clases

Un anchor es un marco rectangular
En donde puede ocurrir una detección

Cada celda tiene B anchors ->

Yolo (You Only Look Once)

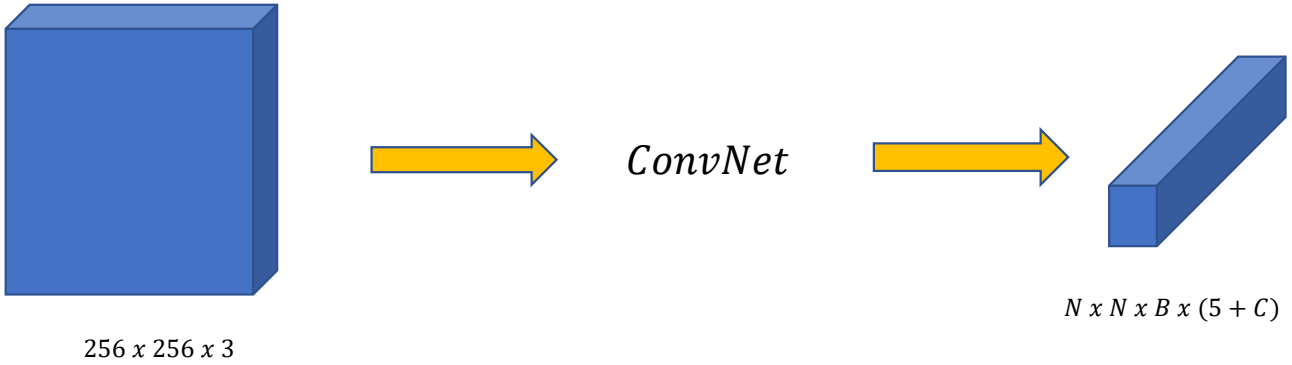


$$y = \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix} \quad \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

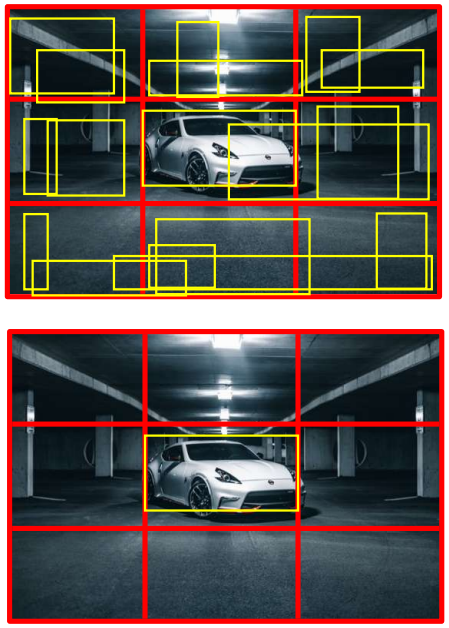
Clases

1. Person
2. Car
3. Motorbike

Yolo (You Only Look Once)



Yolo (You Only Look Once)

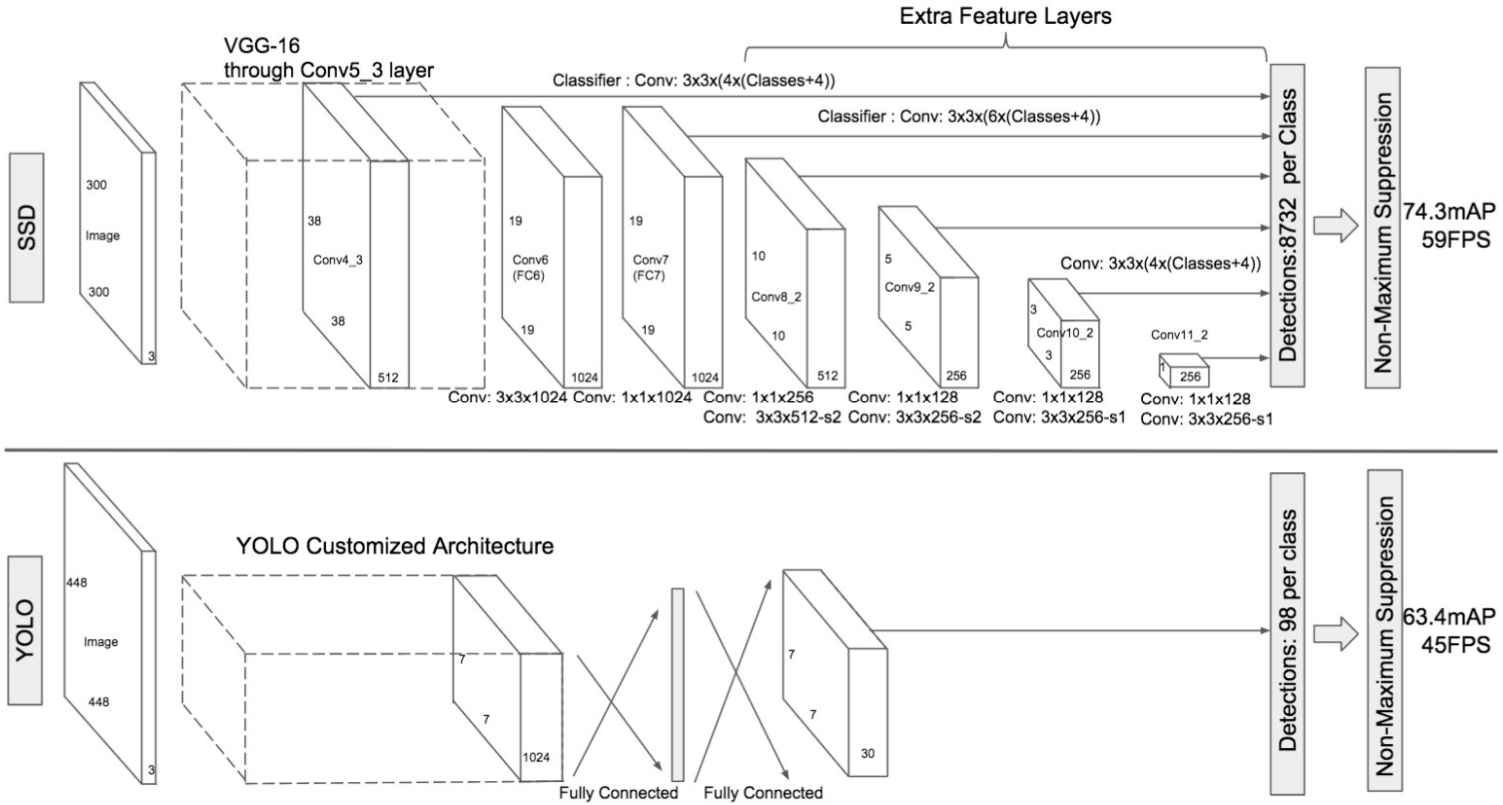


Para cada celda hay B anchor boxes

Se remueven las bajas confidencias

Para cada clase, usamos supresión de no máximos parea generar predicciones finales

Single Shot Detection



Comparison of methods

Model	PASCAL VOC 2007	PASCAL VOC 2010	PASCAL VOC 2012	COCO 2015 (IoU=0.5)	COCO 2015 (IoU=0.75)	COCO 2015 (Official Metric)	COCO 2016 (IoU=0.5)	COCO 2016 (IoU=0.75)	COCO 2016 (Official Metric)	Real Time Speed
R-CNN	x	62.4%	x	x	x	x	x	x	x	No
Fast R-CNN	70.0%	68.8%	68.4%	x	x	x	x	x	x	No
Faster R-CNN	78.8%	x	75.9%	x	x	x	x	x	x	No
R-FCN	82.0%	x	x	53.2%	x	31.5%	x	x	x	No
YOLO	63.7%	x	57.9%	x	x	x	x	x	x	Yes
SSD	83.2%	x	82.2%	48.5%	30.3%	31.5%	x	x	x	No
YOLOv2	78.6%	x	x	44.0%	19.2%	21.6%	x	x	x	Yes

Evaluación

Datos

Datos para localizar objetos.

- Big datasets
- Tagged datasets

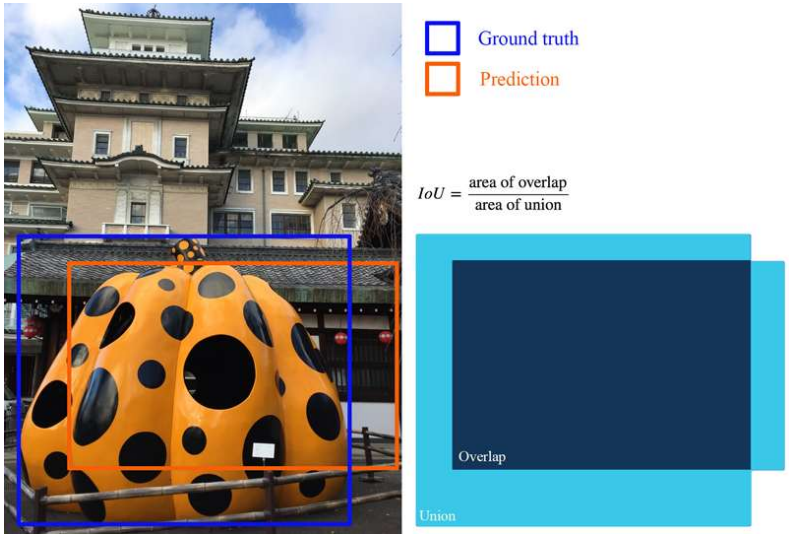
Pascal VOC

COCO dataset

Evaluación de detección

Dado un groundtruth y las detecciones de un algoritmo, cómo sabemos si el algoritmo lo hace bien?

- Una detección es exitosa si existe un alto overlap entre bounding boxes



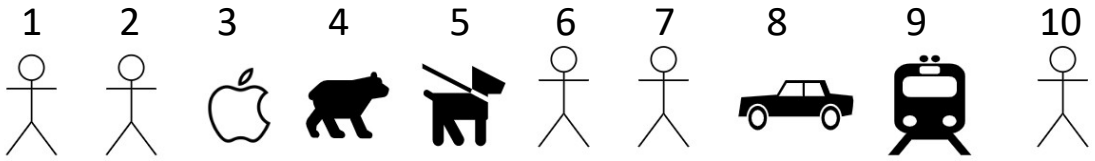
IoU >= 0.5

Evaluación de detección

La métrica más común es el Average Precision (AP). Esta métrica es computada para cada Clase en el dataset. Veamos un ejemplo:

- Tenemos un dataset con una clase *persona*

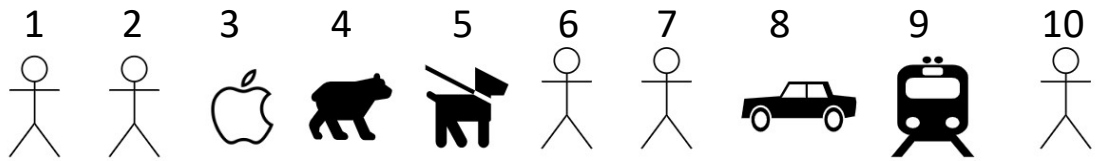
Colectamos todas las ocurrencias en donde el algoritmo predice una persona y Ordenamos de acuerdo a la confidencia



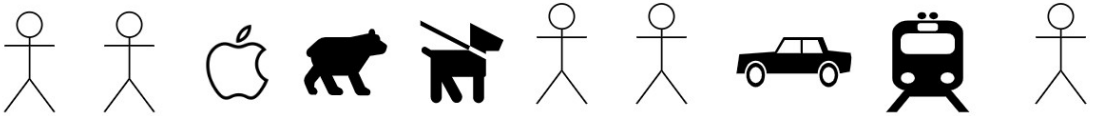
Evaluación de detección

Medimos la precisión y recall en cada posición de la lista

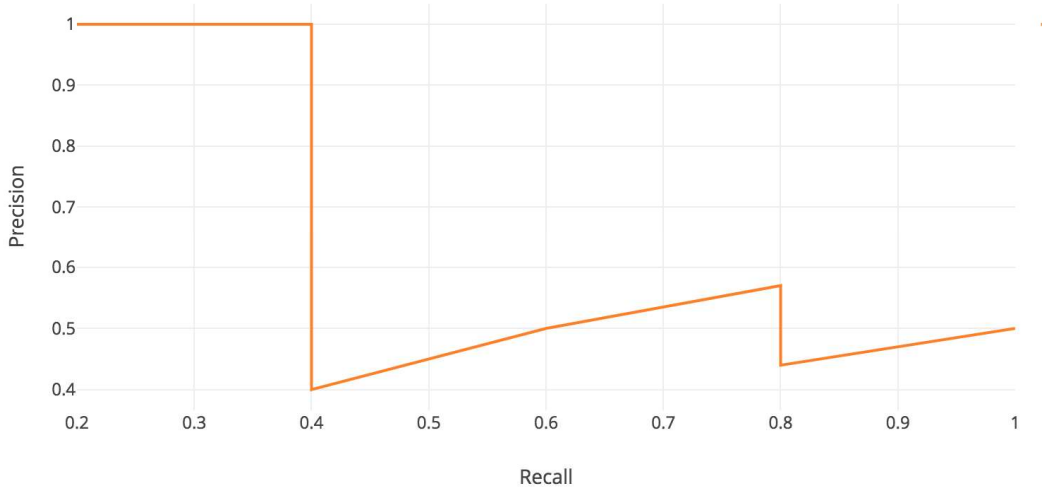
- Precision: número de personas/elementos en lista
- Recall: número de personas/número total de personas

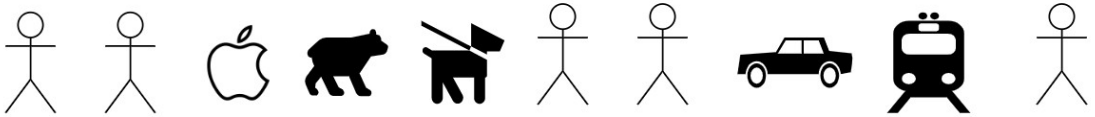


P.	1	1	0.67	0.5	0.4	0.5	0.57	0.5	0.44	0.5
R.	0.2	0.4	0.4	0.4	0.4	0.6	0.8	0.8	0.8	1.0

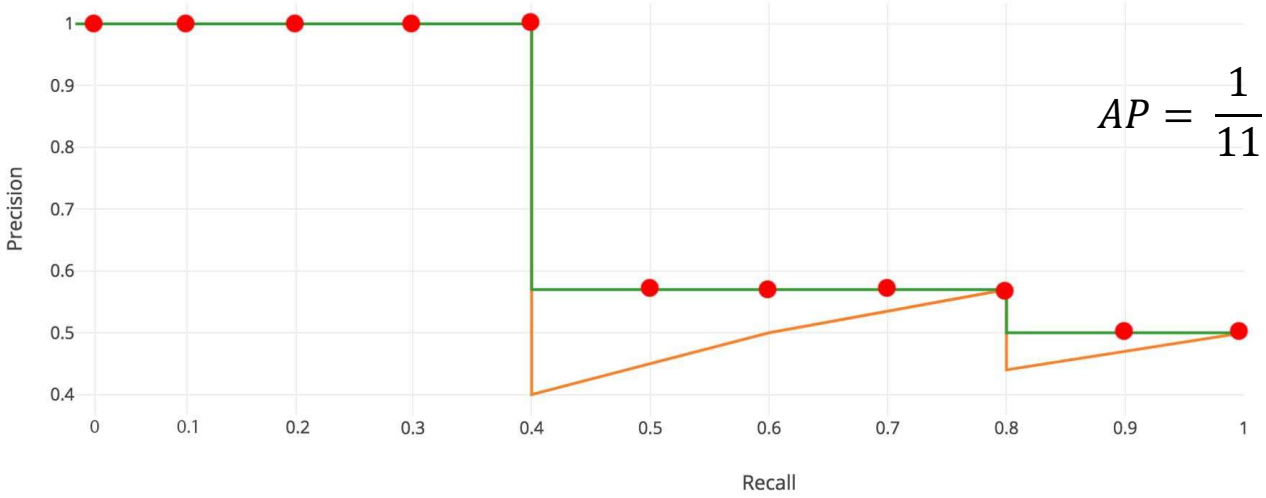
	1	2	3	4	5	6	7	8	9	10
										
P.	1	1	0.67	0.5	0.4	0.5	0.57	0.5	0.44	0.5
R.	0.2	0.4	0.4	0.4	0.4	0.6	0.8	0.8	0.8	1.0

Ploteamos la curva recall vs. precision: precisión promedio es el área bajo la curva



	1	2	3	4	5	6	7	8	9	10
										
P.	1	1	0.67	0.5	0.4	0.5	0.57	0.5	0.44	0.5
R.	0.2	0.4	0.4	0.4	0.4	0.6	0.8	0.8	0.8	1.0

En la práctica, se toman valores interpolados de recall y valores monotónicamente decreciente de precisión



$$AP = \frac{1}{11}(5 * 1 + 4 * 0.57 + 2 * 0.5) = 0.75$$

Para clase *persona*

Mean Average Precision (mAP) es el Promedio de AP para todas las clases