DECSAI
Departamento de Ciencias de la Computación e I.A.
Universidad de Granada

Redes convolutivas

Fernando Berzal, berzal@acm.org

Redes convolutivas

- Motivación: Invarianza
- Redes convolutivas
  - Convolución
  - Mapas de características
  - Campos receptivos locales
  - Pooling
- Resultados experimentales y aplicaciones
  - Clasificación de imágenes
  - Detección de objetos

En la práctica…

Algoritmo de aprendizaje de redes multicapa

Aspectos que debemos considerar en su diseño:

- **Parámetros:** ¿Qué topología de red utilizamos?...
- **Optimización:** ¿Cómo obtenemos los pesos?
- **Generalización:** ¿Cómo conseguimos que la red funcione bien con datos distintos a los del conjunto de entrenamiento?
- **Invarianza:** ¿Cómo conseguimos que la red sea robusta frente a transformaciones comunes en los datos?

DECSAI
Departamento de Ciencias de la Computación e I.A.
Universidad de Granada

Redes convolutivas

Motivación

En la práctica: Invarianza

Técnicas [ad hoc] que nos permiten incorporar conocimiento previo en el diseño de una red neuronal:

- Restringir la arquitectura de la red
  (uso de conexiones locales,
  a.k.a. "campos receptivos").

- Restringir la selección de pesos de la red
  (uso de pesos compartidos por varias neuronas,
  i.e. "weight-sharing").

En la práctica: Invarianza

Motivación

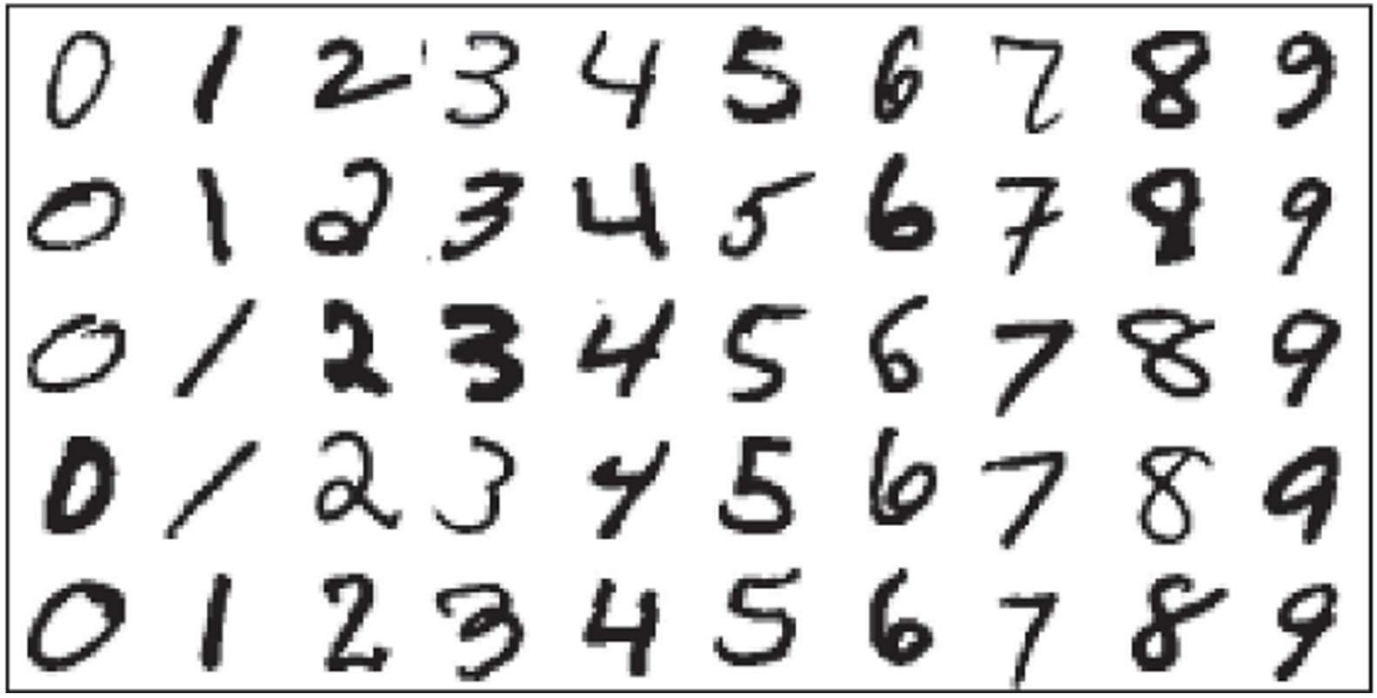
El reconocimiento de objetos es difícil por varios motivos:

- Segmentación (varios objetos en la misma imagen).
- Ocultación (partes ocultas detrás de otros objetos).
- Iluminación (valores de los píxeles determinados tanto por la iluminación ambiental como por el objeto al que corresponden).
- Punto de vista (cambios en el punto de vista causan cambios en la imagen que se capta).
- Deformación (los objetos pueden aparecer de varias deformados de muchas maneras [no afines]), p.ej. textos manuscritos.

En la práctica: Invarianza

Justificación

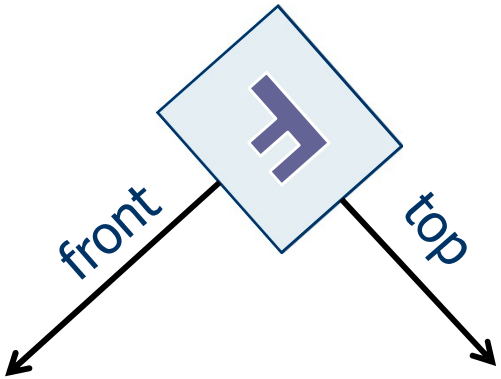
Nuestros sistemas visual y auditivo parecen haberse especializado para tratar sin problemas determinadas transformaciones (sin que seamos conscientes de su complejidad real).



En la práctica: Invarianza

Segmentación: Preprocesamiento de los datos

Identificamos una región alrededor de un objeto y la utilizamos como marco de coordenadas de referencia para un conjunto de píxeles normalizado (p.ej. MNIST).



No siempre es fácil:
¡Hay que reconocer el objeto para delimitar su región!

En la práctica: Invarianza

Por fuerza bruta

- En la fase de entrenamiento, utilizamos imágenes bien segmentadas y ya normalizadas.
- En la fase de prueba, probamos con distintas regiones (variando escala y orientación).

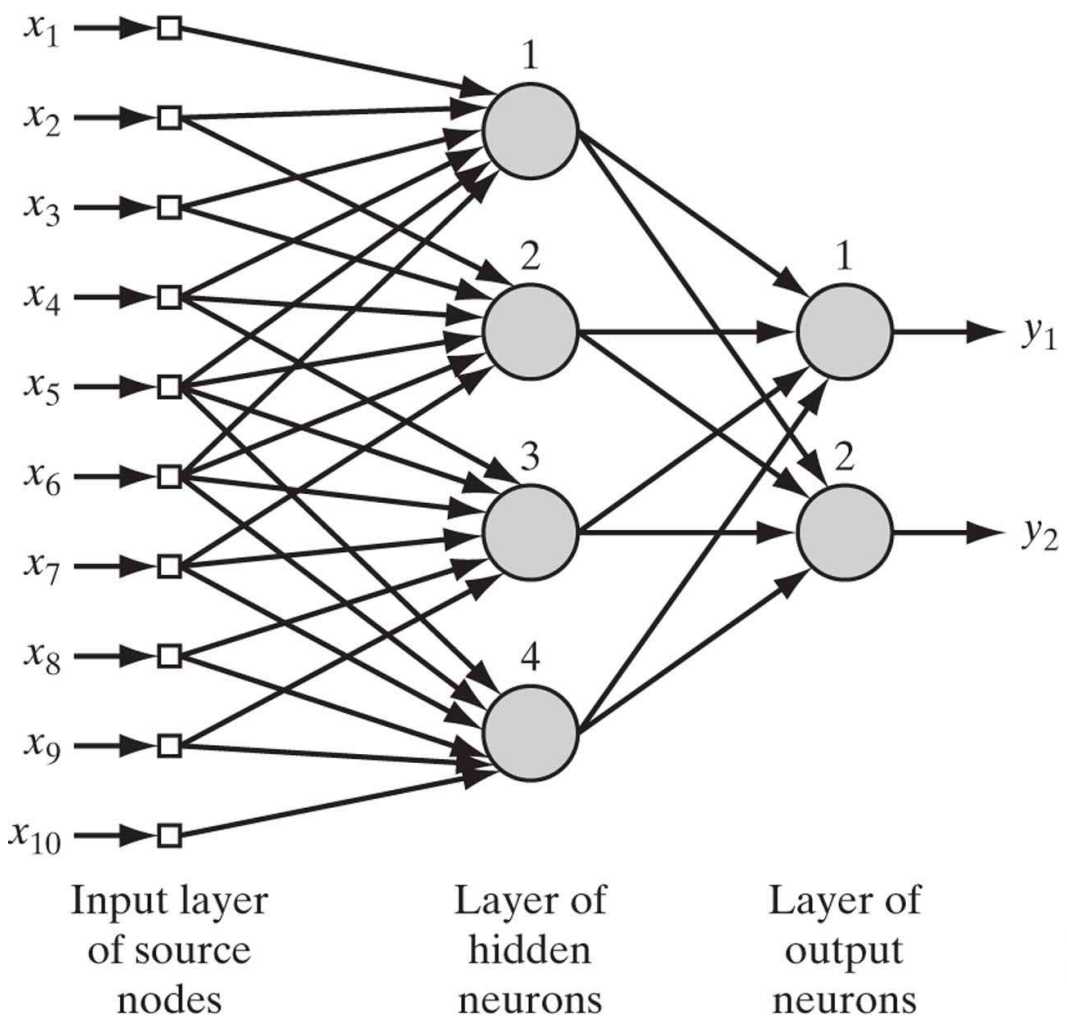
EJEMPLO

Reconocimiento de caras

Redes convolutivas

Campos receptivos

y pesos compartidos



[Haykin: "Neural Networks and Learning Machines", 3rd edition]

DECSAI
Departamento de Ciencias de la Computación e I.A.
Universidad de Granada

Redes convolutivas

Convolución

Redes convolutivas

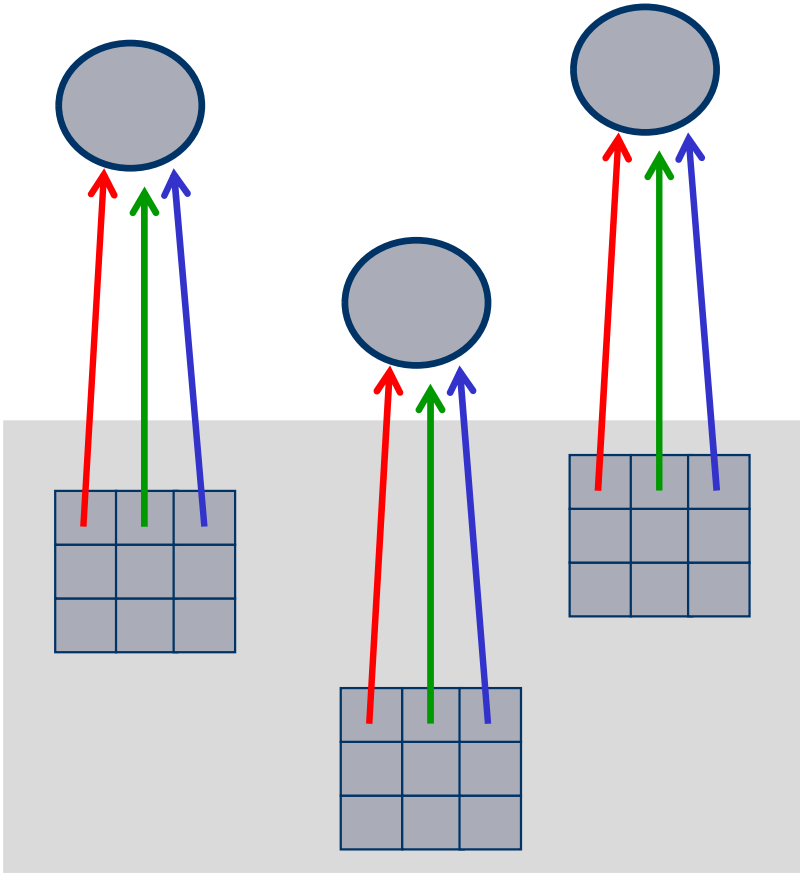
Ideas clave

- Se usan múltiples copias de los mismos detectores de características en distintas posiciones.
- La replicación reduce el número de parámetros (pesos) que deben ajustarse.
- Se pueden utilizar distintos tipos de detectores (cada uno con su "mapa" de detectores replicados): cada fragmento de la imagen se representa de distintas formas.

Redes convolutivas

Cada unidad detecta una característica en una región diferente de la imagen.

Todas comparten los mismos pesos.



Redes convolutivas

Convolución

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Imagen de entrada

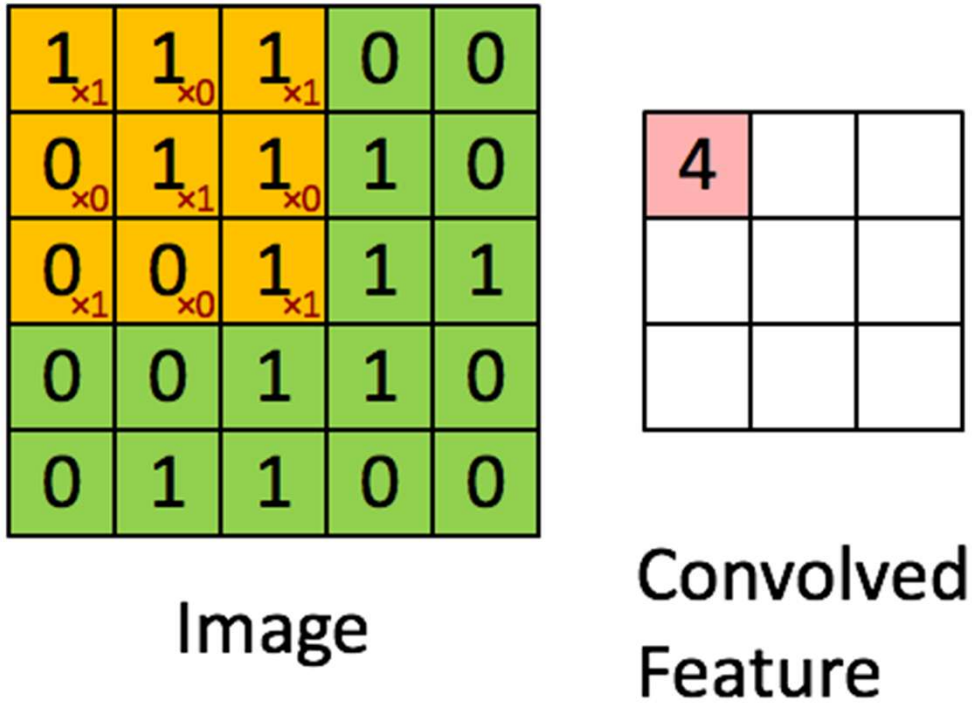
1	0	1
0	1	0
1	0	1

Máscara
filtro / kernel /
detector de características

http://ufldl.stanford.edu/tutorial/supervised/FeatureExtractionUsingConvolution/

Redes convolutivas

Convolución



http://ufldl.stanford.edu/tutorial/supervised/FeatureExtractionUsingConvolution/

Redes convolutivas

Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	

Redes convolutivas

Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Redes convolutivas

Ejemplos: GIMP

- Sharpen

0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0

- Blur

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

Redes convolutivas

Ejemplos: GIMP

- Edges

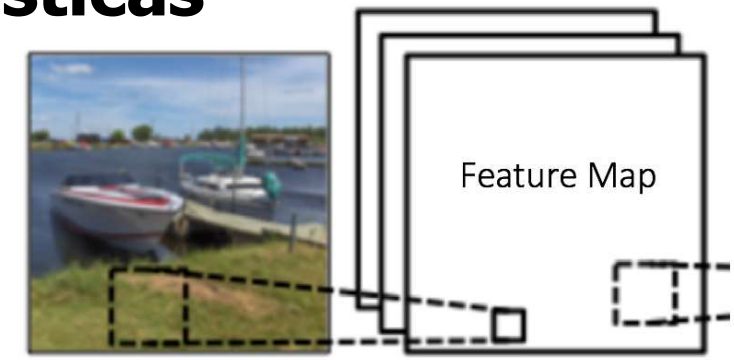
	0	1	0	
	1	-4	1	
	0	1	0	

- Emboss

	-2	-1	0	
	-1	1	1	
	0	1	2	

Redes convolutivas

Mapas de características ["Feature maps"]

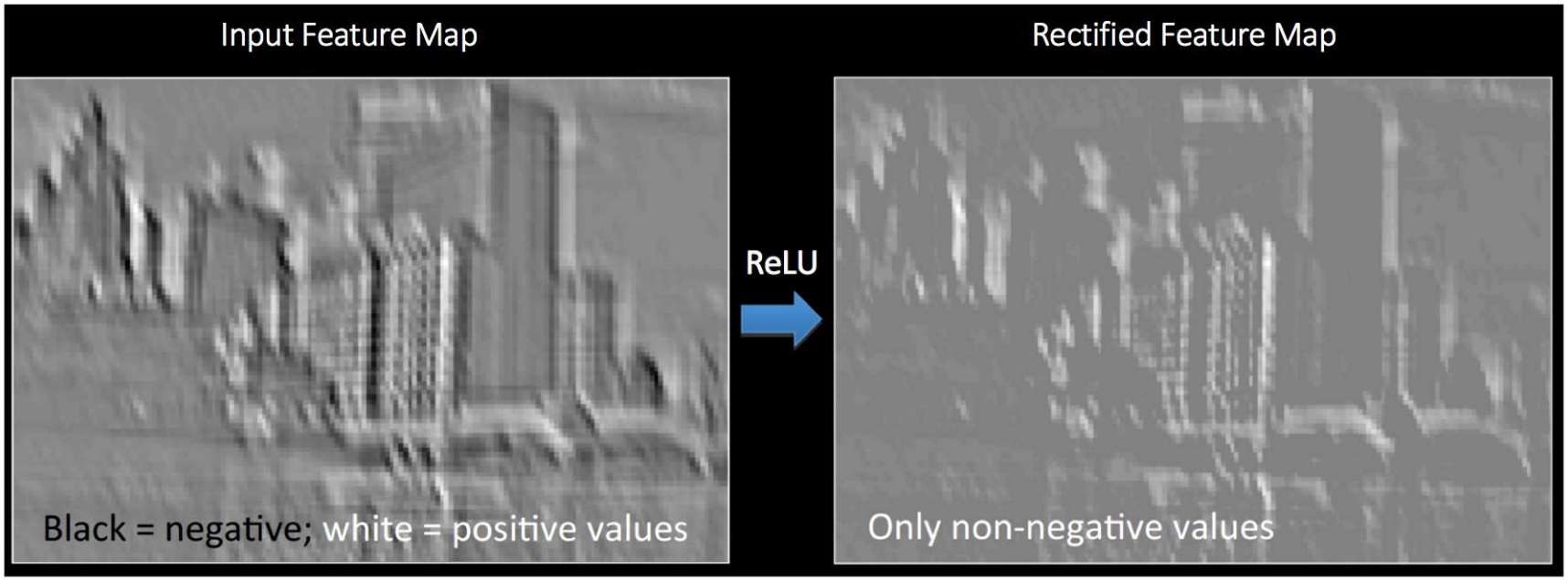


Feature Map having depth of 3 (since 3 filters have been used)

Redes convolutivas

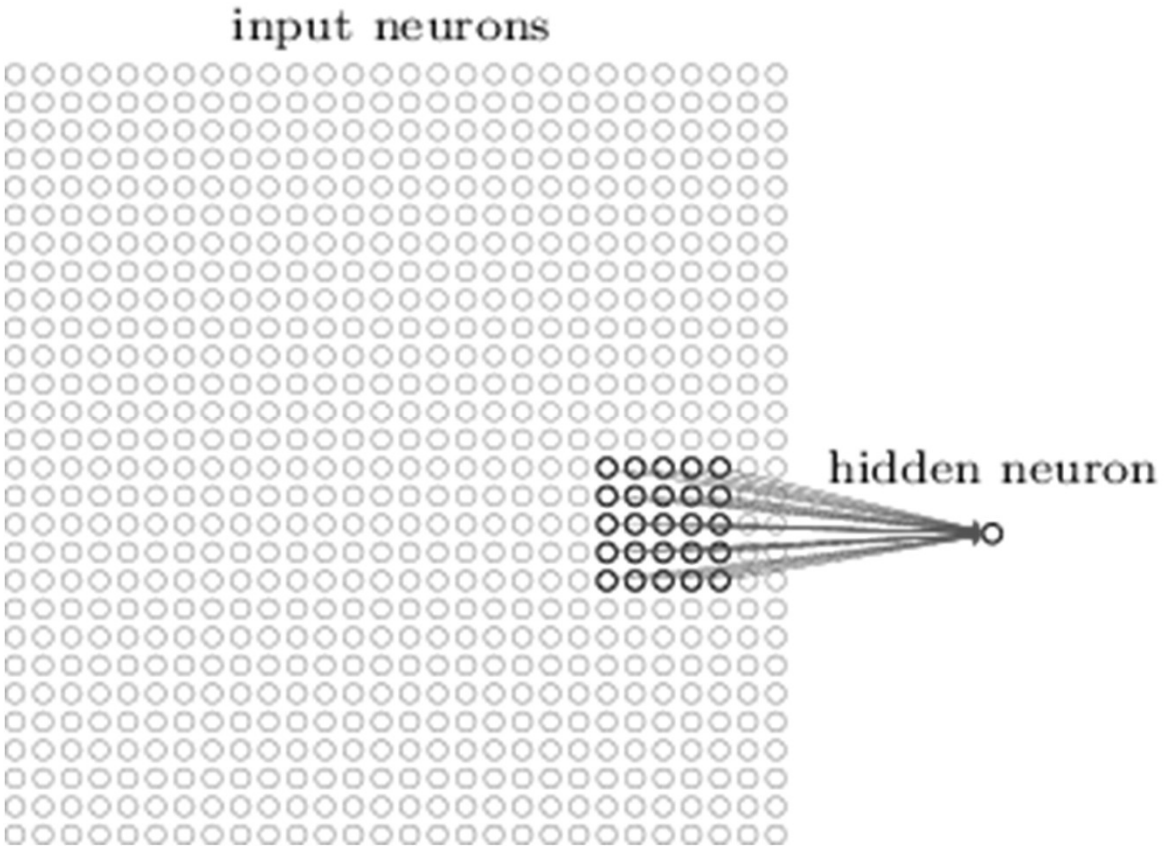
"Rectified feature maps"

ReLU @ convolutional layer



Redes convolutivas

Campos receptivos locales ["local receptive fields"]



http://neuralnetworksanddeeplearning.com/

Redes convolutivas

Backpropagation con restricciones sobre los pesos:

$$Para\ obligar\ w_1 = w_2$$

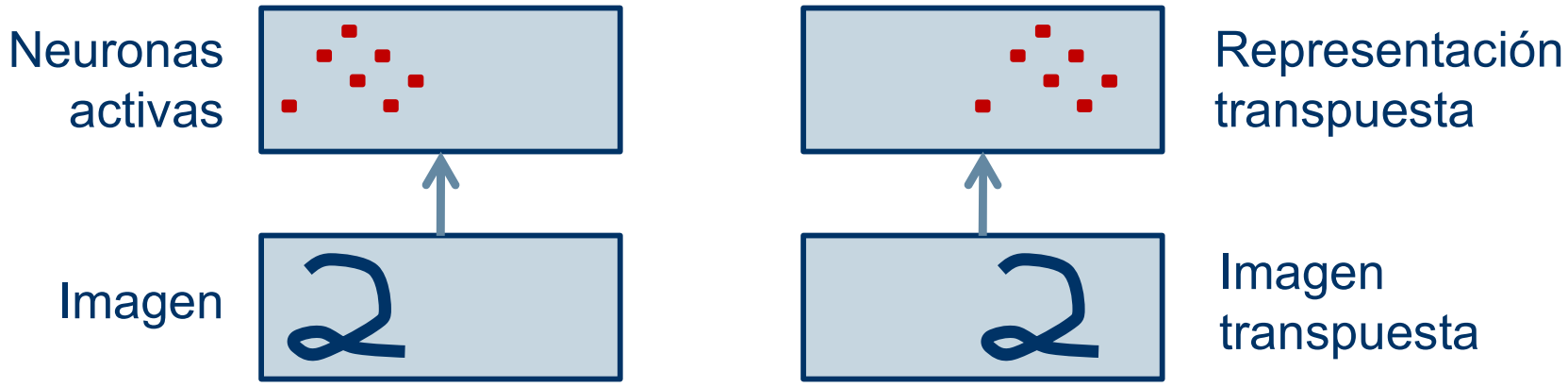
$$tenemos\ que\ garantizar\ \Delta w_1 = \Delta w_2$$

$$Calculamos\ \frac{\partial E}{\partial w_1}\ y\ \frac{\partial E}{\partial w_2}$$

$$Usamos\ \frac{\partial E}{\partial w_1} + \frac{\partial E}{\partial w_2}\ para\ \ w_1\ y\ w_2$$

Redes convolutivas

¿Qué consiguen los detectores replicados?



Los detectores replicados no hacen que la actividad neuronal sea invariante frente a traslaciones:
su actividad es "equivariante".

Si la característica resulta útil en algunas posiciones, la red incorpora detectores para esa característica en todas las posiciones.

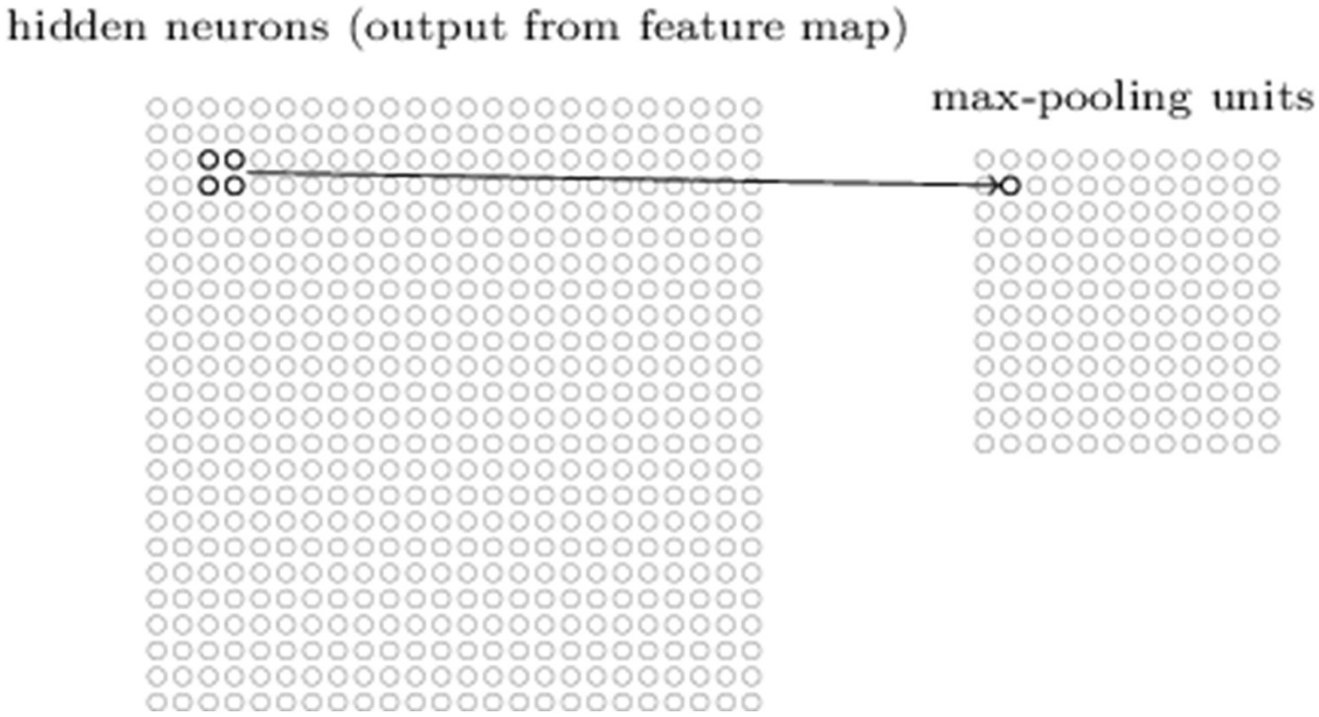
DECSAI
Departamento de Ciencias de la Computación e I.A.
Universidad de Granada

Redes convolutivas

Pooling

Redes convolutivas

Pooling (reducción de la dimensionalidad)

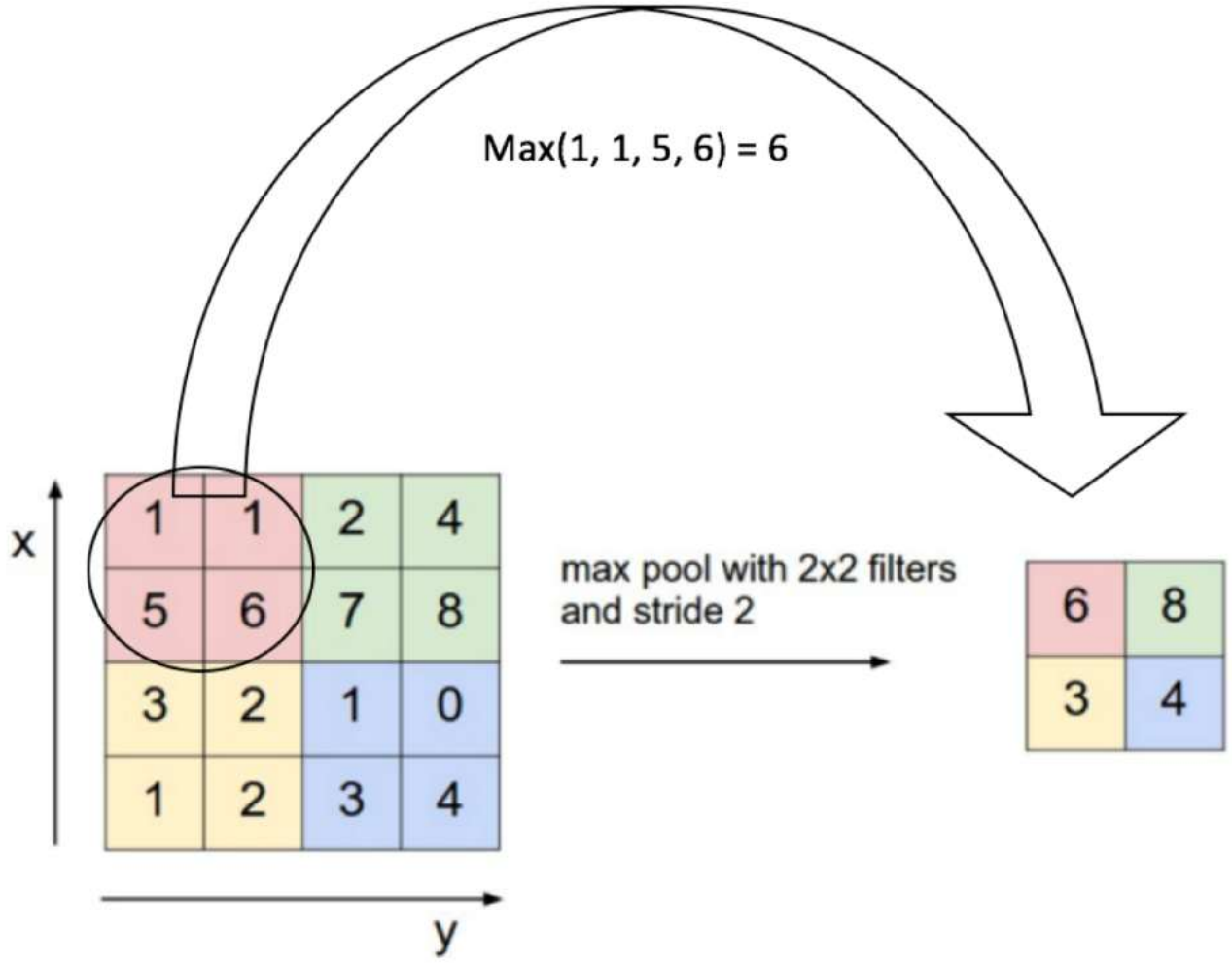


http://neuralnetworksanddeeplearning.com/

Redes convolutivas

Pooling

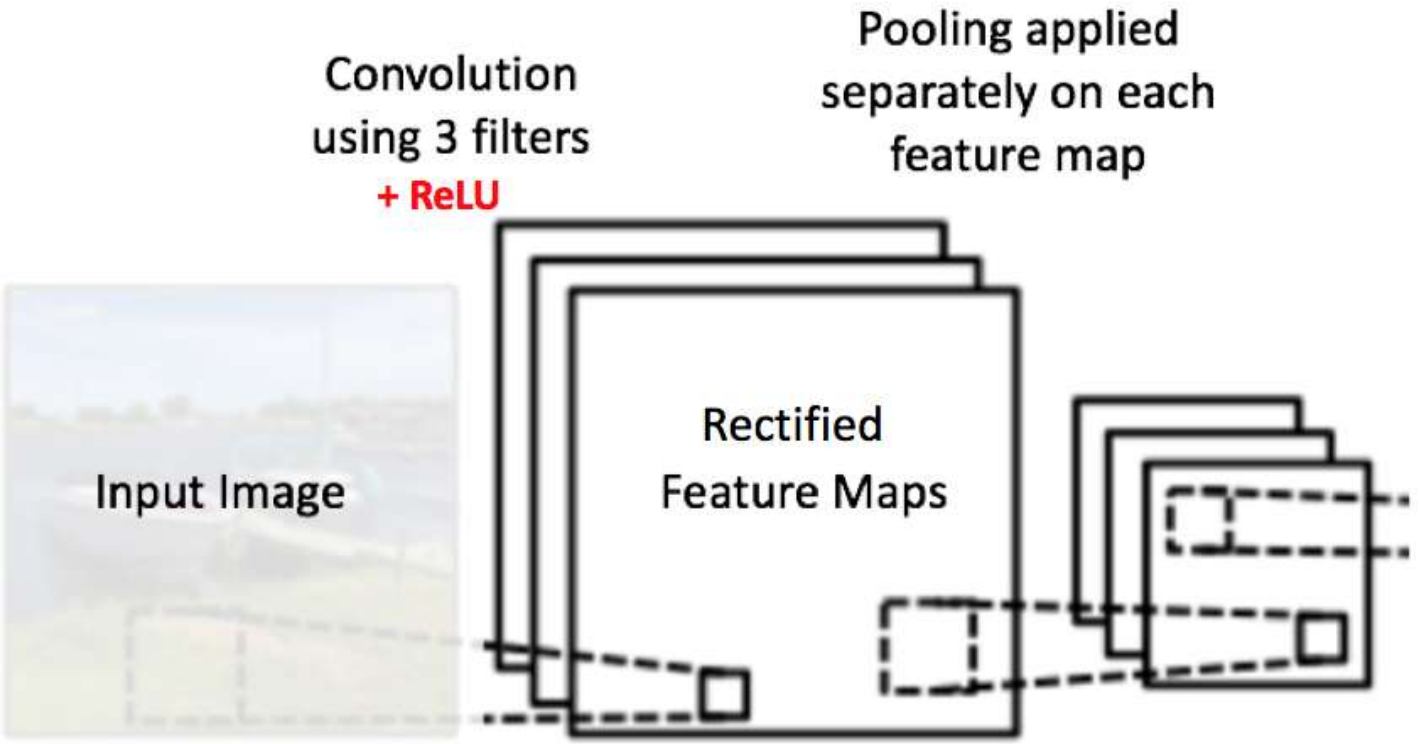
a.k.a. spatial pooling / subsampling / downsampling



Redes convolutivas

Pooling

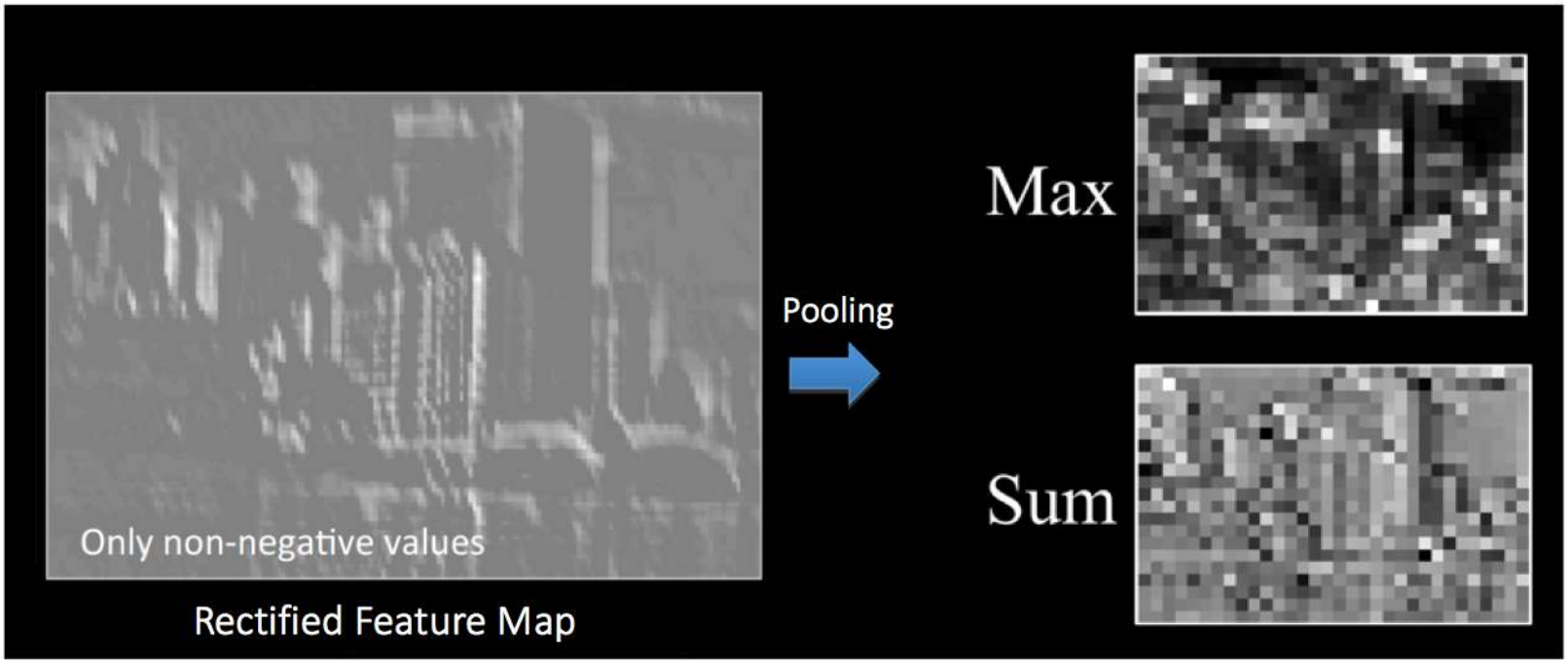
a.k.a. spatial pooling / subsampling / downsampling



Redes convolutivas

Pooling

a.k.a. spatial pooling / subsampling / downsampling



Redes convolutivas

Pooling

¿Qué se consigue agregando las salidas de receptores de características replicados?

- Una reducción del número de entradas de las siguientes capas de la red neuronal, lo que nos permite calcular más características distintas en paralelo.
- Una pequeña cantidad de invarianza frente a traslaciones en cada nivel (p.ej. 2x2 max-pooling).
- Problema: Tras varios niveles de "pooling", se pierde información acerca de la posición exacta de las cosas.

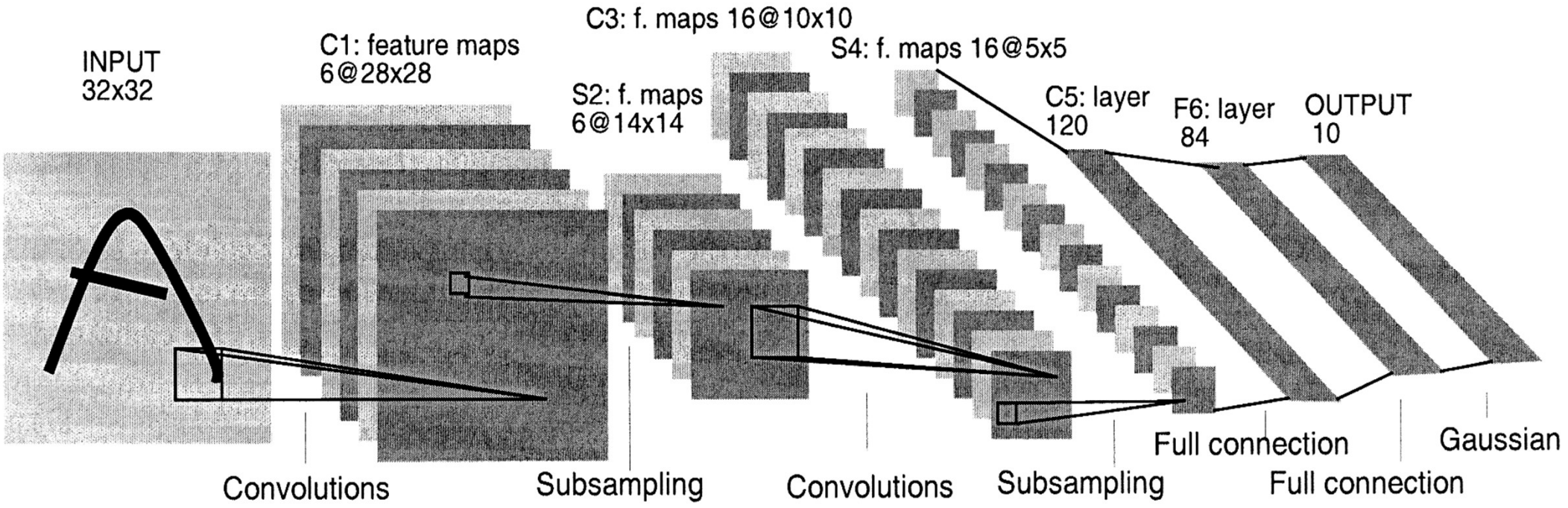
DECSAI
Departamento de Ciencias de la Computación e I.A.
Universidad de Granada

Redes convolutivas

Aplicaciones: Clasificación de imágenes

Aplicaciones

Clasificación de dígitos manuscritos (MNIST)



EJEMPLO: **LeNet5**

http://yann.lecun.com/exdb/lenet/

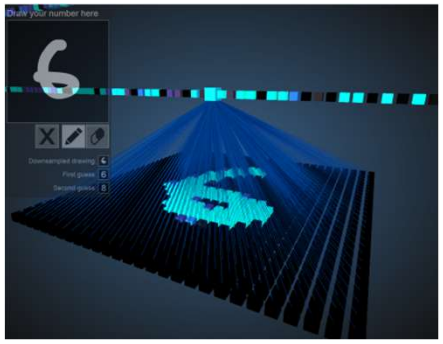
Aplicaciones

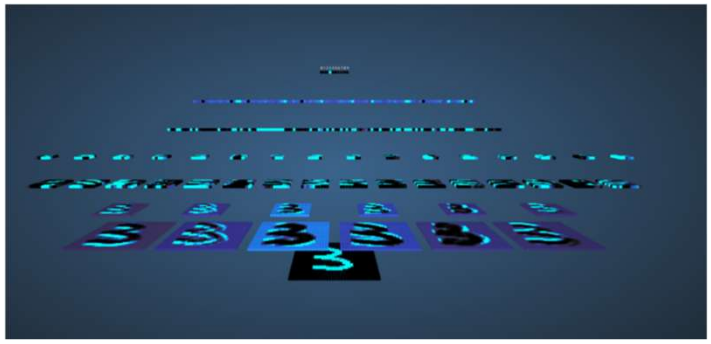
EJEMPLO: Los 82 errores de **LeNet5**

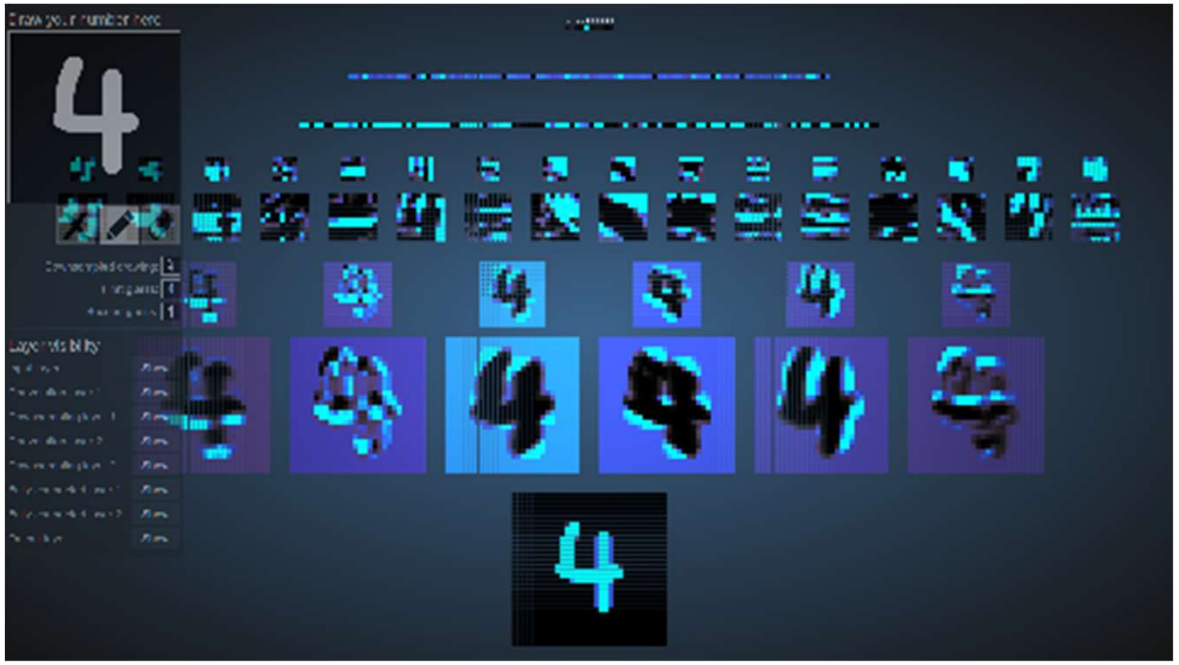
http://yann.lecun.com/exdb/lenet/

Aplicaciones

Clasificación de dígitos manuscritos (MNIST)





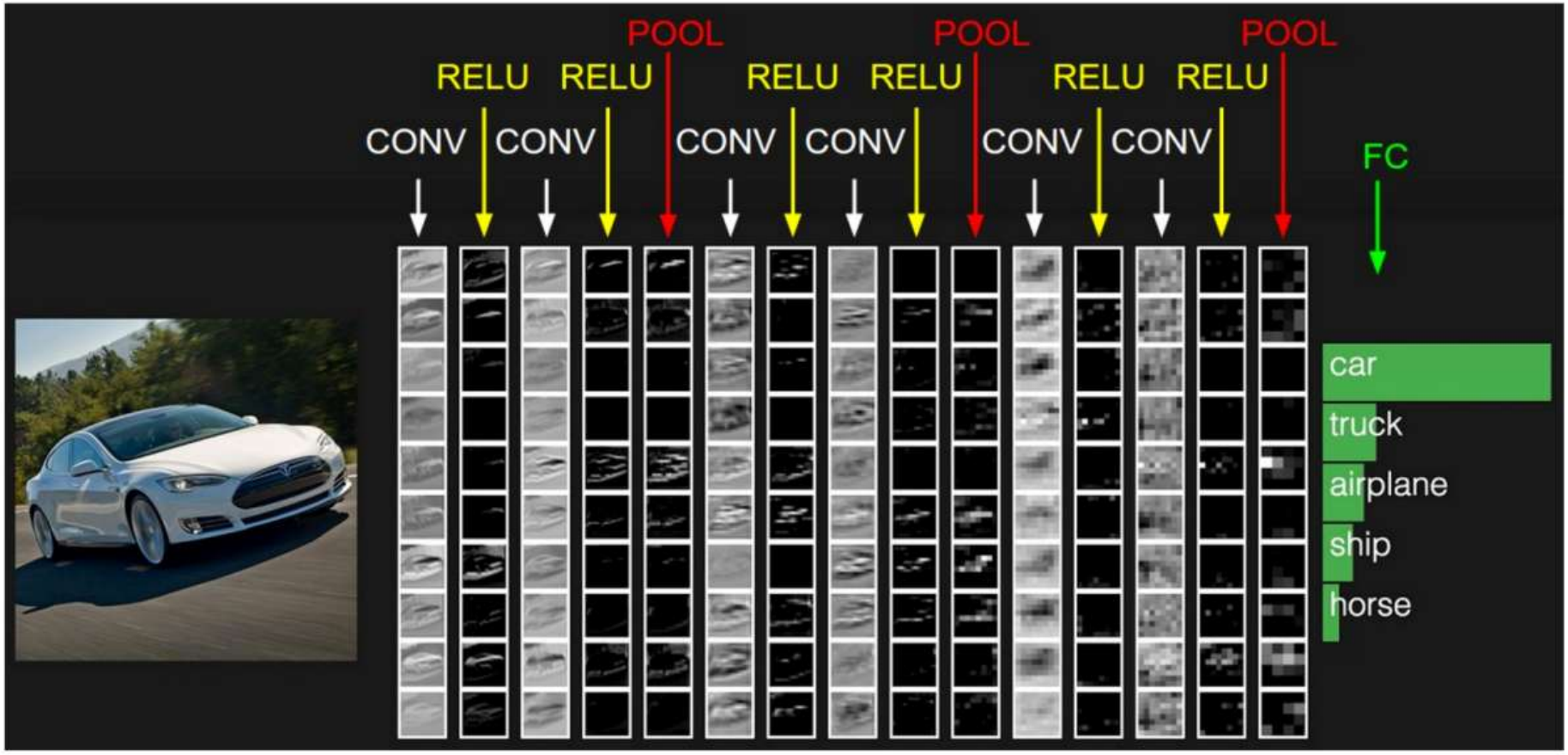


DEMO: MNIST

http://scs.ryerson.ca/~aharley/vis/conv/flat.html

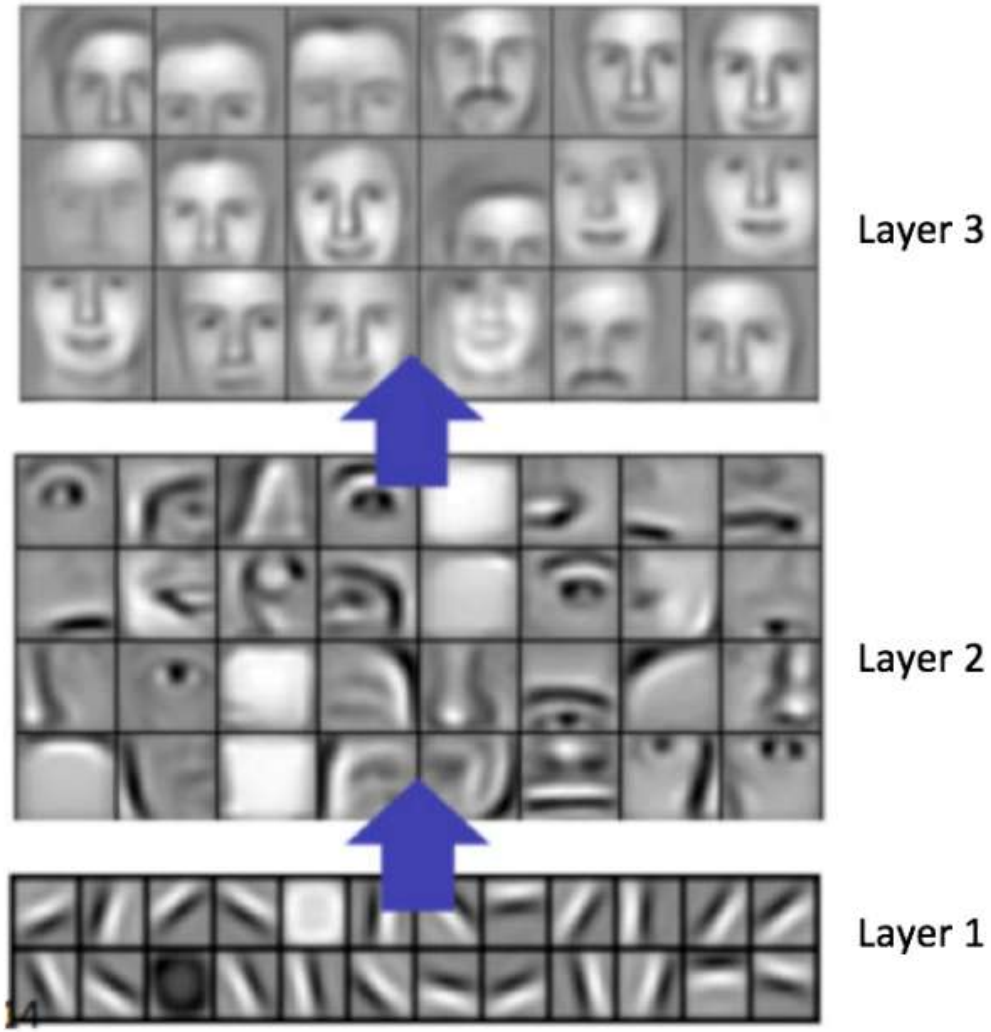
Aplicaciones

Clasificación de imágenes



Aplicaciones

Reconocimiento facial



DEMOS:

Buscar en YouTube

Aplicaciones

Clasificación de imágenes: Evolución

- AlexNet @ ILSVRC'2012: Alex Krizhevsky et al.
  University of Toronto: 60M parameters https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
- ZF Net @ ILSVSC'2013: Matthew Zeiler & Rob Fergus
  NYU: https://arxiv.org/abs/1311.2901
- GoogLeNet @ ILSVC'2014: Szegedy et al.
  Google "Inception module": 4M parameters http://arxiv.org/abs/1409.4842
- VGGNet @ ILSVC'2014 (2nd place):
  Oxford Visual Geometry Group
  http://www.robots.ox.ac.uk/~vgg/research/very_deep/
- ResNets [residual nets] @ ILSVC'2015:
  Microsoft Research: https://arxiv.org/abs/1512.03385
- DenseNet (2016):
  Densely-connected convolutional networks. http://arxiv.org/abs/1608.06993

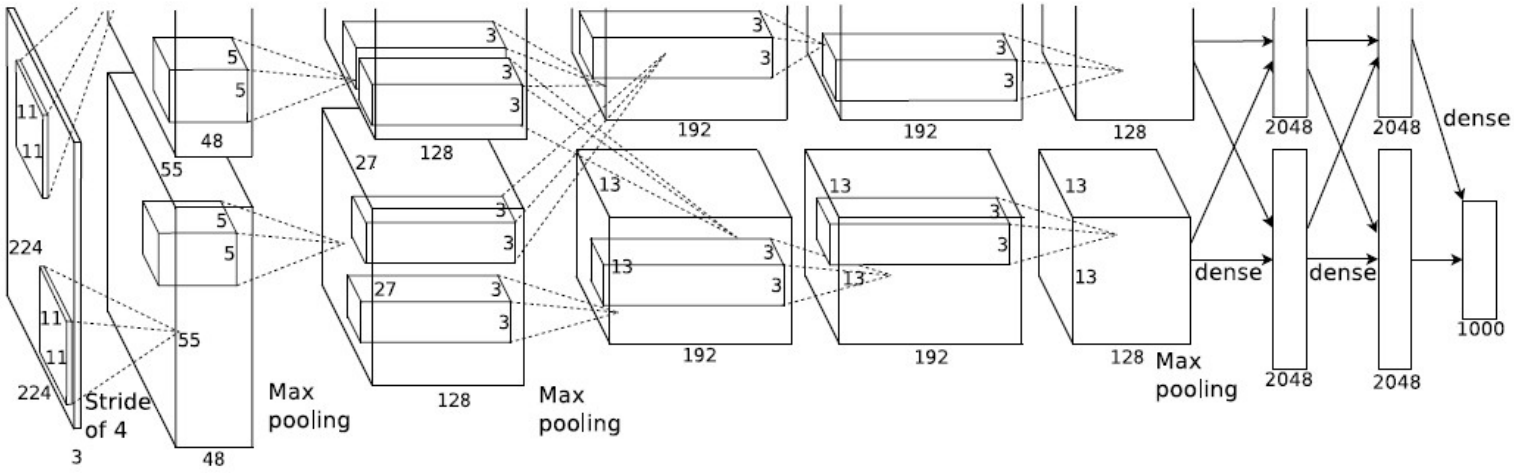
Aplicaciones

IMAGENET

Large Scale Visual Recognition Challenge

Red neuronal diseñada por Alex Krizhevsky (NIPS 2012)

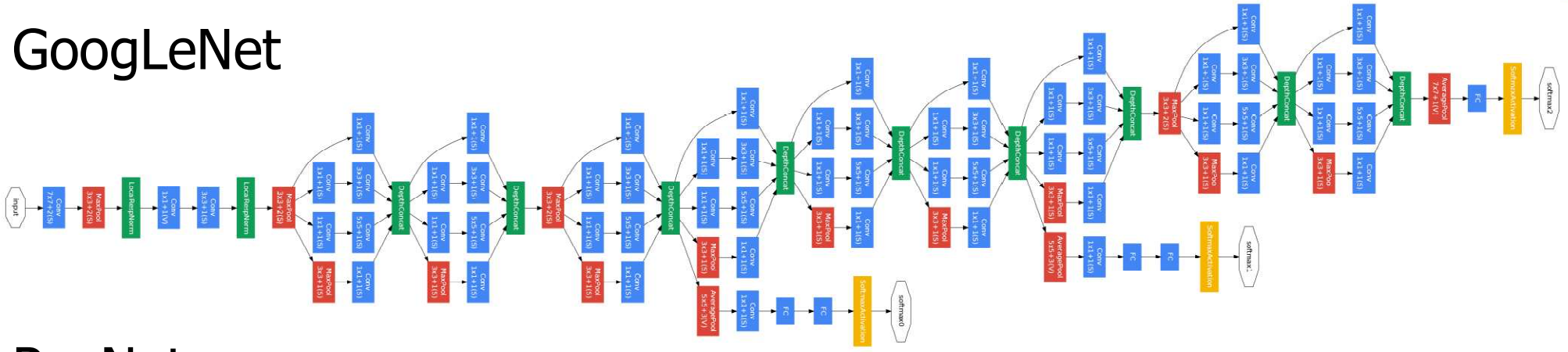
- "Deep network" con topología compleja: 7 capas ocultas (las primeras convolutivas, las últimas completamente conectadas).



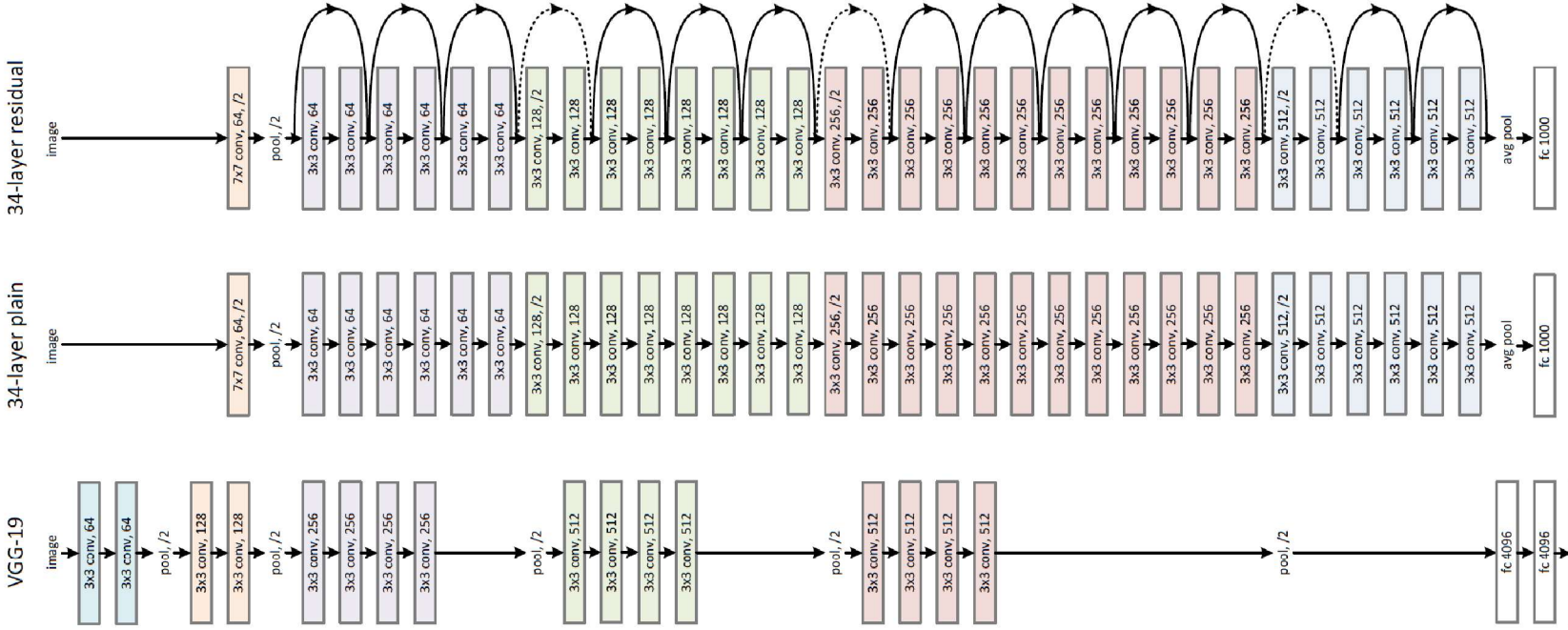
- Múltiples trucos para mejorar su capacidad de generalización ("image patches" & "dropout").

Aplicaciones

GoogLeNet

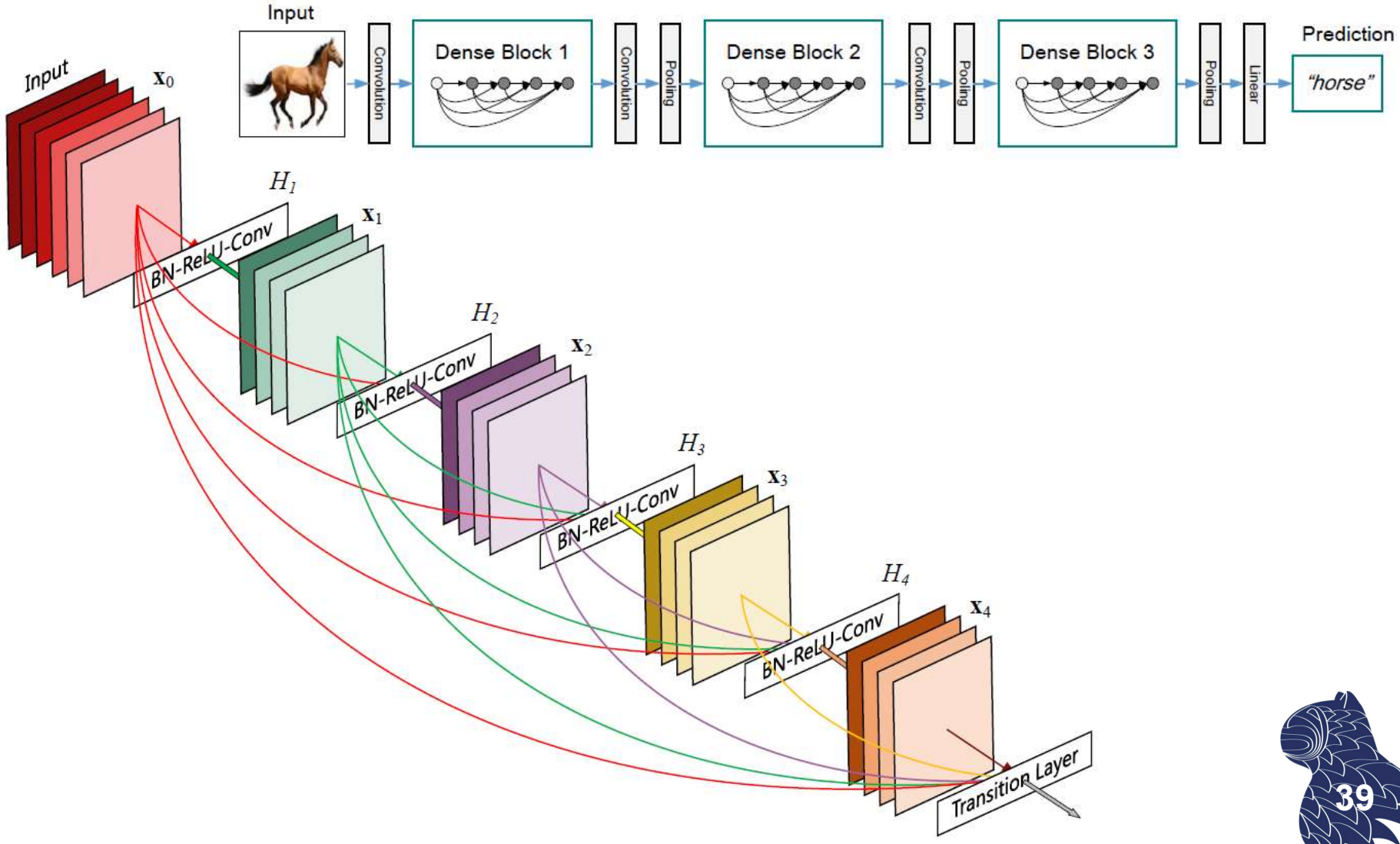


ResNets



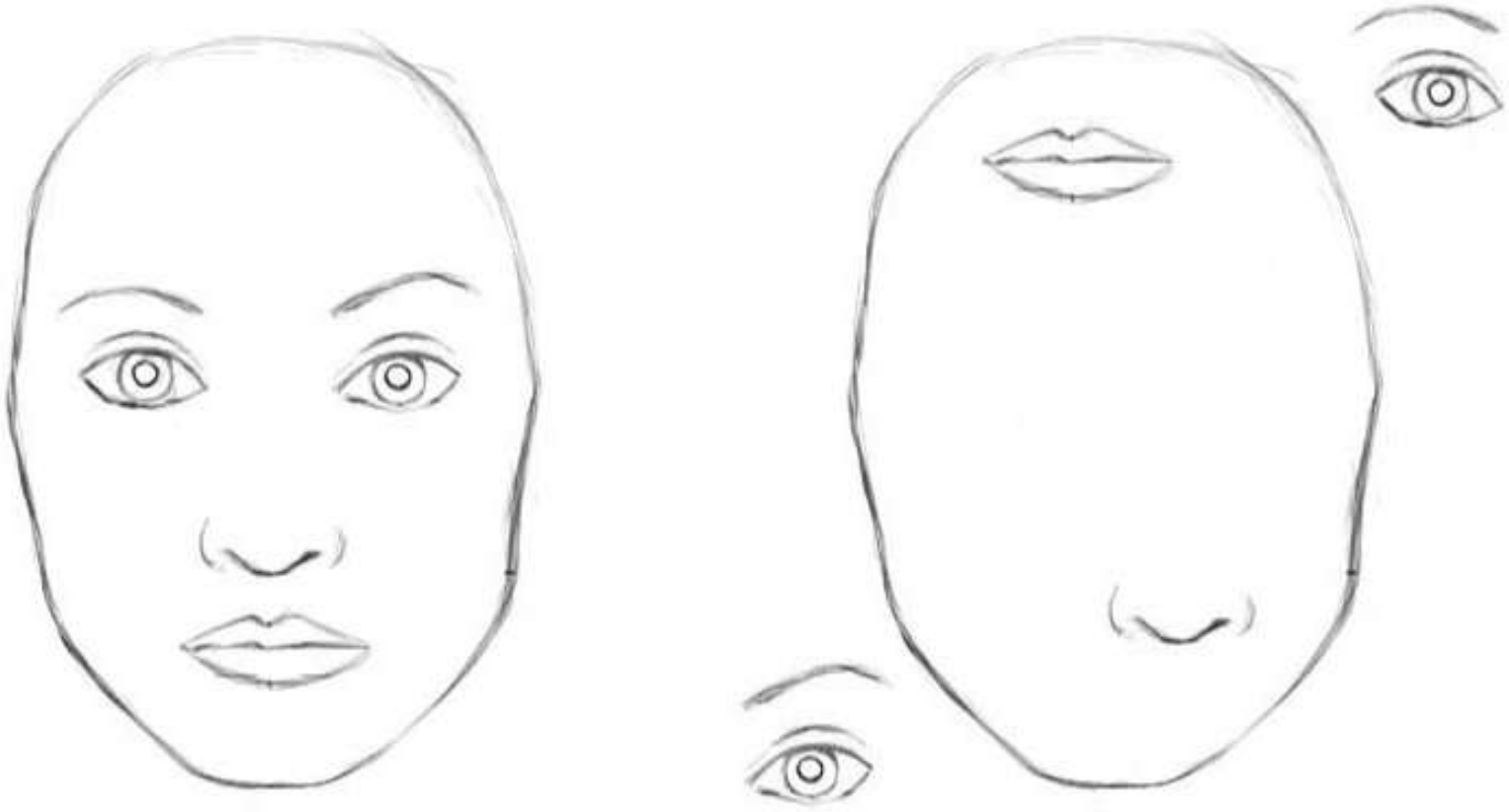
Aplicaciones

DenseNets



Limitaciones del Deep Learning

Las redes convolutivas [CNNs] funcionan muy bien en la práctica, pero…



… para una CNN, ambas imágenes son similares ☹

Limitaciones del Deep Learning

"Convolutional neural networks are doomed"
—Geoffrey Hinton

person	0.88

reddish orange color	0.78
light brown color	0.78
starlet	0.66
entertainer	0.66
female	0.60
woman	0.59
young lady (heroine)	0.59

person	0.90

light brown color	0.84
starlet	0.77
entertainer	0.77
female	0.65
woman	0.64
young lady (heroine)	0.64
reddish orange color	0.64
newsreader	0.50

coal black color	0.79

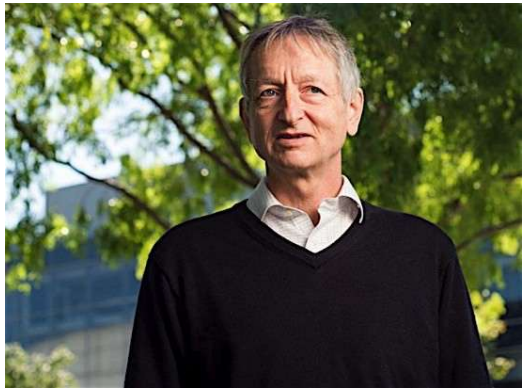
hairpiece (hair)	0.71
dress	0.71
maroon color	0.71
person	0.58
toupee (hairpiece)	0.58
woman	0.56
Earrings	0.55
female	0.50

Limitaciones del Deep Learning

- Las redes convolutivas detectan características, pero no su colocación relativa (traslación & rotación).

- Las redes convolutivas ignoran las posiciones relativas utilizando "pooling", un apaño que funciona sorprendentemente bien en la práctica:

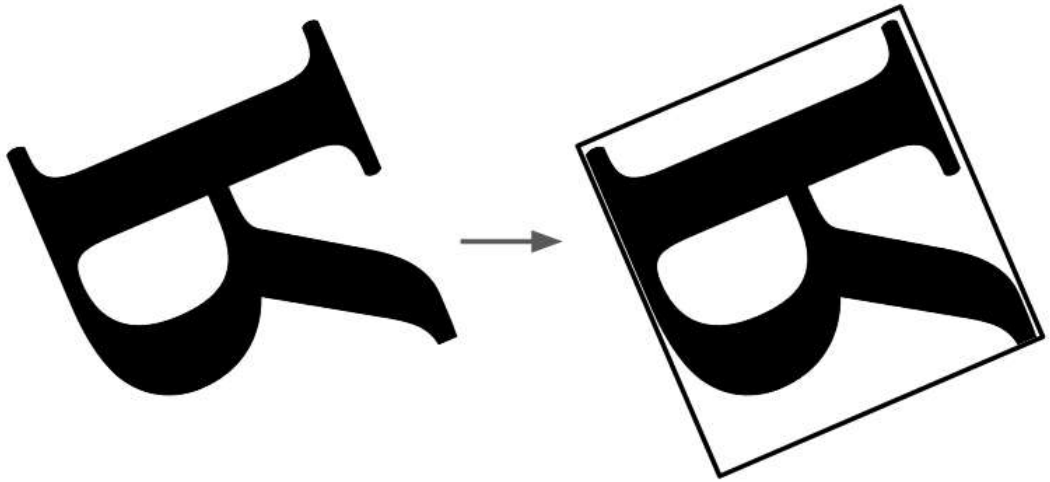
"The pooling operation used in convolutional neural networks is a big mistake and the fact that it works so well is a disaster." – Geoffrey Hinton



Limitaciones del Deep Learning

Problema clave de las redes convolutivas

La representación interna de una red convolutiva no tiene en cuenta las relaciones espaciales entre objetos, ni la jerarquía existente entre objetos simples y los objetos compuestos de los que forman parte.



Limitaciones del Deep Learning

Limitaciones

Falta de compresión (en sentido humano) …

The boy is holding a baseball bat.

DECSAI
Departamento de Ciencias de la Computación e I.A.
Universidad de Granada

Redes convolutivas

En la práctica: Invarianza & robustez

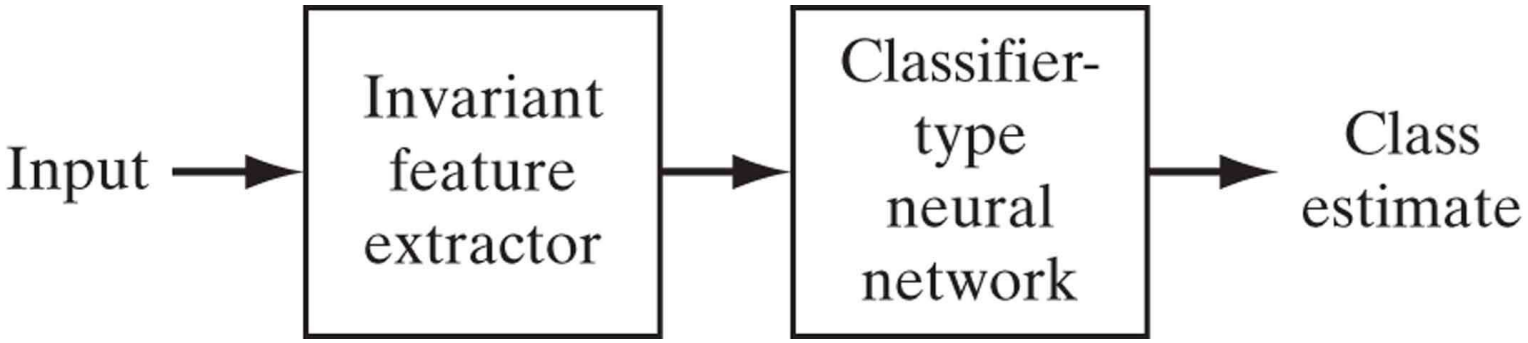
En la práctica: Invarianza

Técnicas que nos permiten diseñar redes robustas, invariantes frente a determinadas transformaciones (rotaciones o cambios de escala en imágenes; cambios de volumen, velocidad o tono en reconocimiento de voz):

- Invarianza por **extracción de características**.
- Invarianza por **estructura**.
- Invarianza por **entrenamiento**.

En la práctica: Invarianza

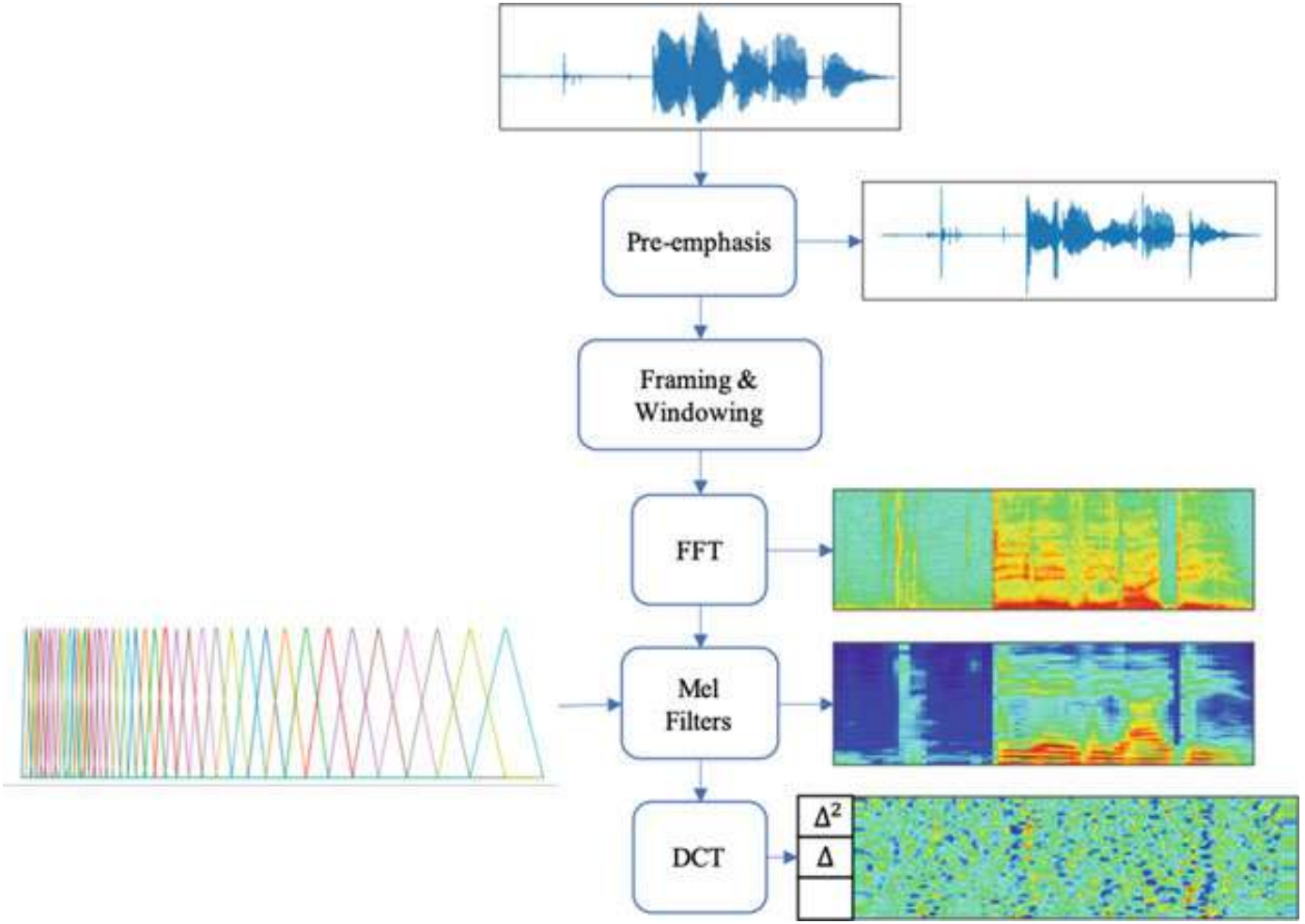
Invarianza por extracción de características
Preprocesamiento del conjunto de entrenamiento, del que se extraen características "esenciales" que sean invariantes con respecto a las transformaciones deseadas).

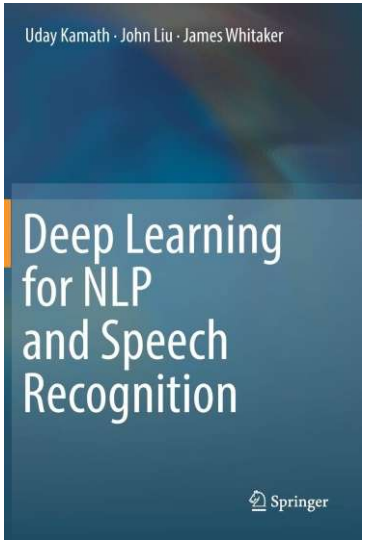


[Haykin: "Neural Networks and Learning Machines", 3rd edition]

En la práctica: Invarianza

Ejemplo:
Procesamiento de voz





En la práctica: Invarianza

Invarianza por estructura, p.ej. redes convolutivas

Red diseñada de forma que sus conexiones sinápticas hagan que versiones transformadas de la entrada produzcan salidas similares.

- Conectividad de la red ("local receptive fields")
- Restricciones sobre los pesos ("weight sharing")
- …

Menos tedioso que diseñar/extraer a mano características, aunque introduce prejuicios acerca de la forma particular de resolver el problema que tengamos en mente.

En la práctica: Invarianza

Invarianza por entrenamiento

Red entrenada a partir de un conjunto de entrenamiento ampliado, en el que se incluyen versiones transformadas de los ejemplos originales del conjunto de entrenamiento.

- El entrenamiento puede ser mucho más costoso.
- La optimización de los pesos de la red puede descubrir formas novedosas de utilizar la red multicapa
  … que a nosotros no se nos hayan ocurrido :-)
  … que nosotros nunca sepamos interpretar :-(

En la práctica: Invarianza

Invarianza por entrenamiento

LeNet5 utiliza conocimiento del problema para diseñar la estructura de la red (invarianza por estructura):

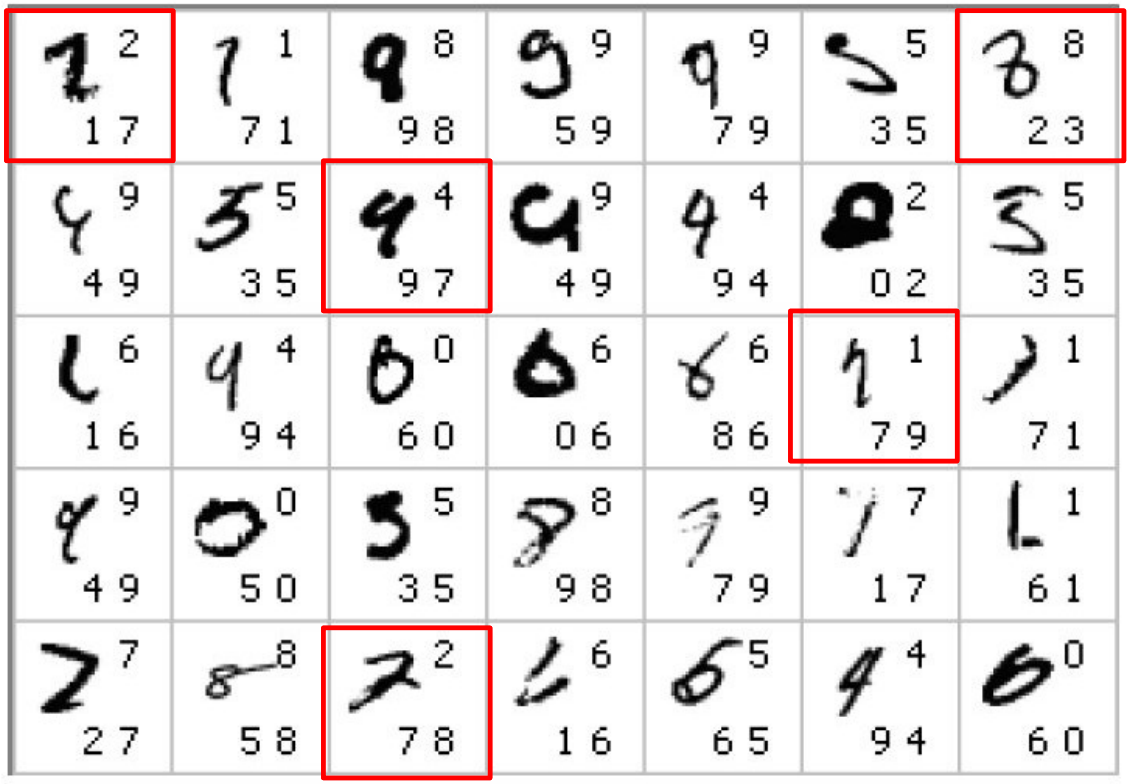
82 errores (0.82%)

Ciresan et al. (2010) crearon **cuidadosamente** nuevos ejemplos de entrenamiento aplicando distintos tipos de transformaciones, tras lo que entrenaron una "deep, dumb net" usando una GPU:

35 errores (0.35%)

En la práctica: Invarianza

Invarianza por entrenamiento



EJEMPLO: Los 35 errores de Ciresan et al. (2010)
http://arxiv.org/abs/1003.0358

En la práctica: Invarianza

Invarianza por entrenamiento

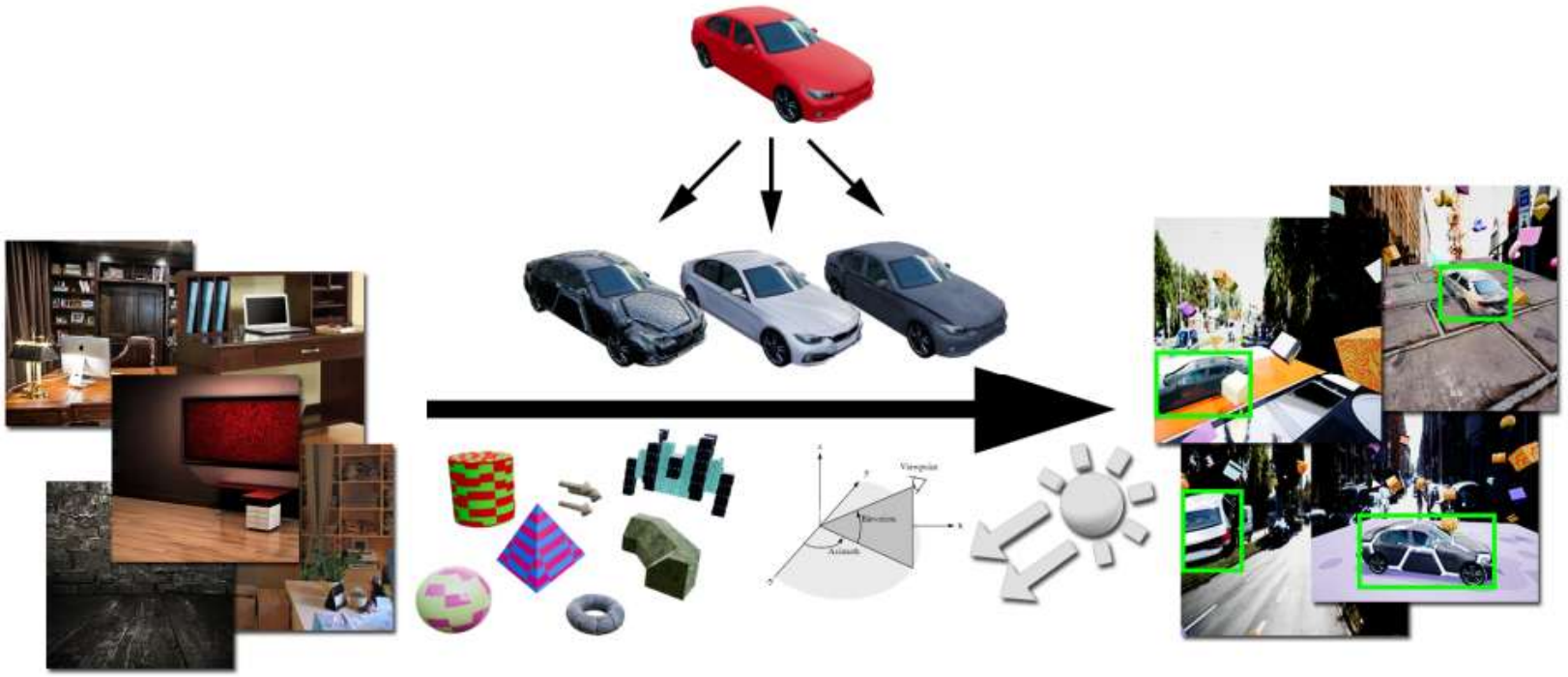


Figure 1. Domain randomization for object detection. Synthetic objects (in this case cars, top-center) are rendered on top of a random background (left) along with random flying distractors (geometric shapes next to the background images) in a scene with random lighting from random viewpoints. Before rendering, random texture is applied to the objects of interest as well as to the flying distractors. The resulting images, along with automatically-generated ground truth (right), are used for training a deep neural network.

Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization. CVPR 2018 Workshop on Autonomous Driving https://arxiv.org/abs/1804.06516

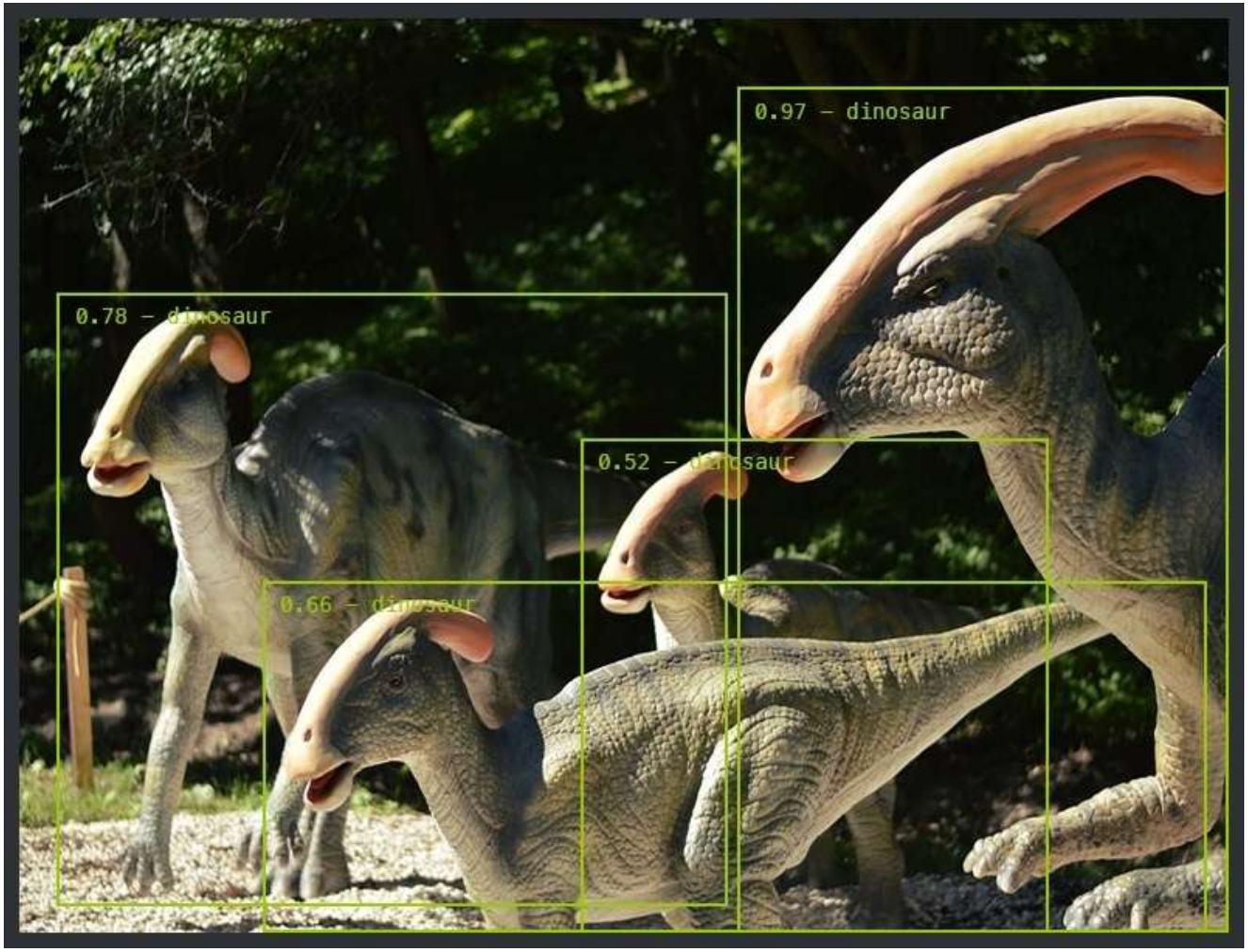
DECSAI
Departamento de Ciencias de la Computación e I.A.
Universidad de Granada

Redes convolutivas

Aplicaciones: Detección de objetos

Apéndice
Detección de objetos

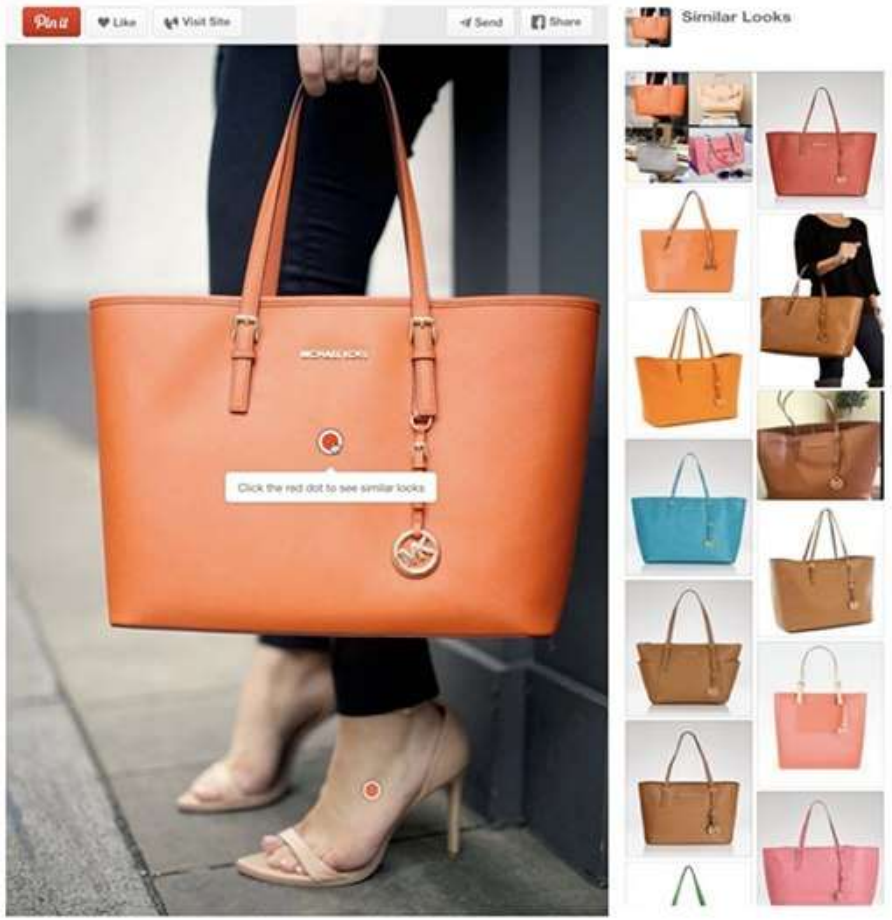
El problema



https://tryolabs.com/blog/2017/08/30/object-detection-an-overview-in-the-age-of-deep-learning

Apéndice
Detección de objetos

Aplicaciones: Visual Search Engine (Pinterest)



https://labs.pinterest.com/assets/paper/visual_search_at_pinterest.pdf
https://tryolabs.com/blog/2017/08/30/object-detection-an-overview-in-the-age-of-deep-learning

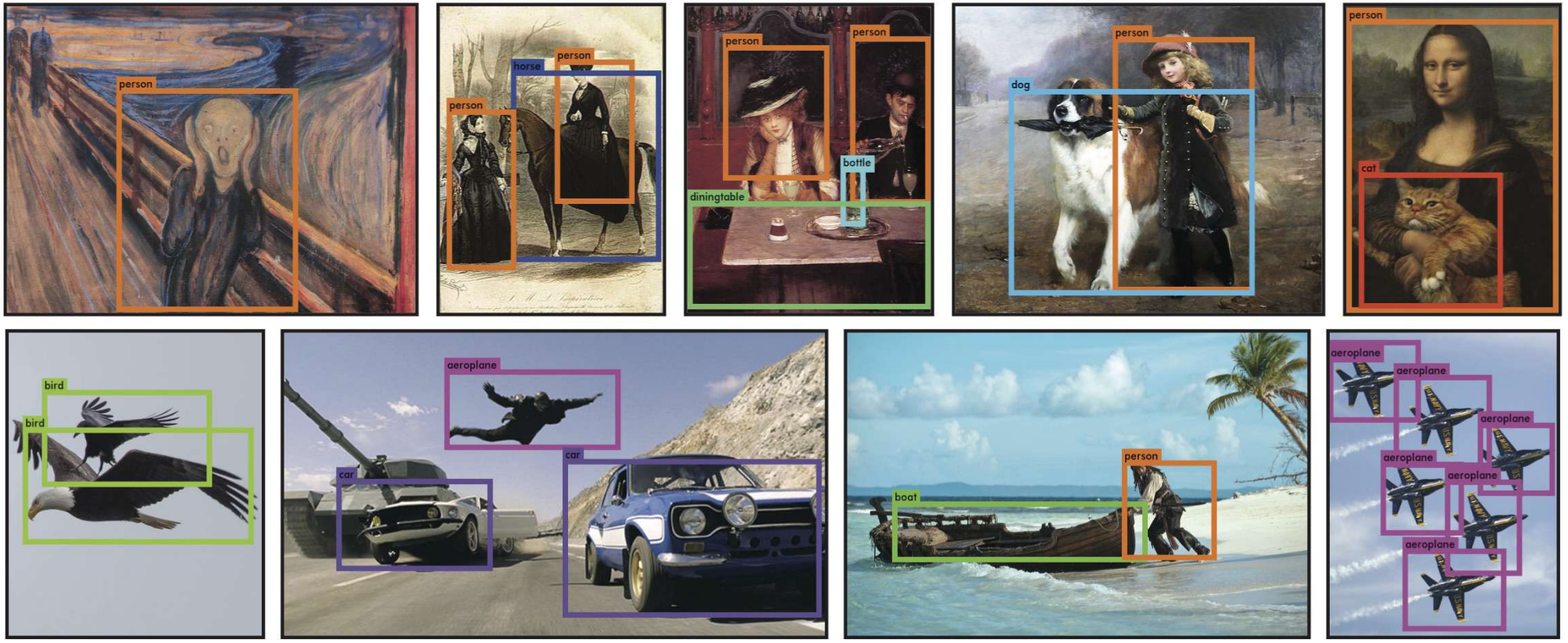
Apéndice
Detección de objetos

Aplicaciones: Análisis de imágenes aéreas

https://tensorflight.com/
https://tryolabs.com/blog/2017/08/30/object-detection-an-overview-in-the-age-of-deep-learning

Apéndice
Detección de objetos

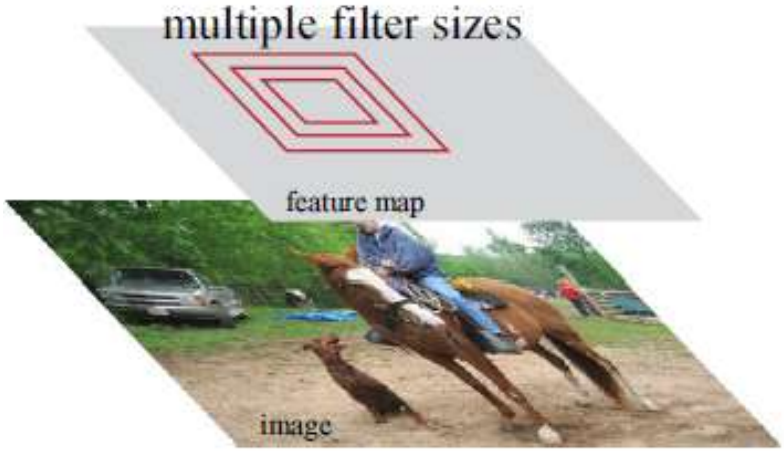
Problema técnico:



¿Número de objetos, tamaño y posición?

Apéndice
Detección de objetos

Una primera solución
Ventana deslizante



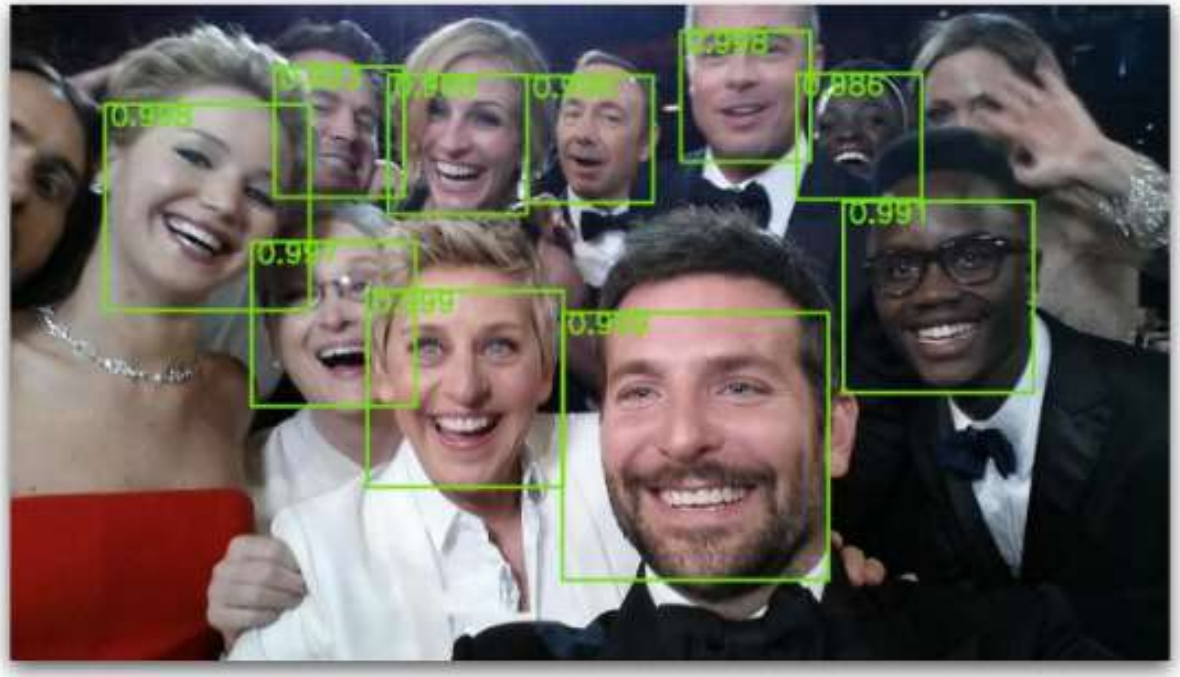
Apéndice
Detección de objetos

Soluciones clásicas

Paul Viola & Michael Jones (Compaq):

"Robust Real-time Object Detection", 2001

- Miles de clasificadores utilizando características Haar.
- Detección de caras en tiempo real (p.ej. cámaras)



Apéndice
Detección de objetos

Soluciones clásicas

Gradientes de una imagen
(cambios del color de un pixel individual con respecto a sus vecinos)

Vector $$\nabla f(x,y) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} f(x+1,y) - f(x-1,y) \\ f(x,y+1) - f(x,y-1) \end{bmatrix}$$

Magnitud $$g = \sqrt{g_x^2 + g_y^2}$$

Dirección $$\theta = \arctan\left(g_y / g_x\right)$$

Soluciones clásicas

Gradientes de una imagen

El cálculo del gradiente se puede realizar mediante una operación de convolución con un kernel o máscara de convolución:

$$\mathbf{G}_x = [-1, 0, 1] \qquad \mathbf{G}_y = [+1, 0, -1]^\top$$

En Python:

```
import numpy as np
import scipy.signal as sig

data = np.array([[…], […], …, […]])
G_x = sig.convolve2d(data, np.array([[-1, 0, 1]]), mode='valid')
G_y = sig.convolve2d(data, np.array([[-1], [0], [1]]), mode='valid')
```

Apéndice
Detección de objetos

Soluciones clásicas

HOG [Histogram of Oriented Gradients]

U.S. Patent 1986 – CVPR'2005

e.g. Detección de peatones

Input example

Average gradients

Input image

Histogram of Oriented Gradients

Apéndice
Detección de objetos

Soluciones clásicas

HOG [Histogram of Oriented Gradients]

U.S. Patent 1986 – CVPR'2005

1. Se calculan los gradientes de la imagen (vector, magnitud y dirección)

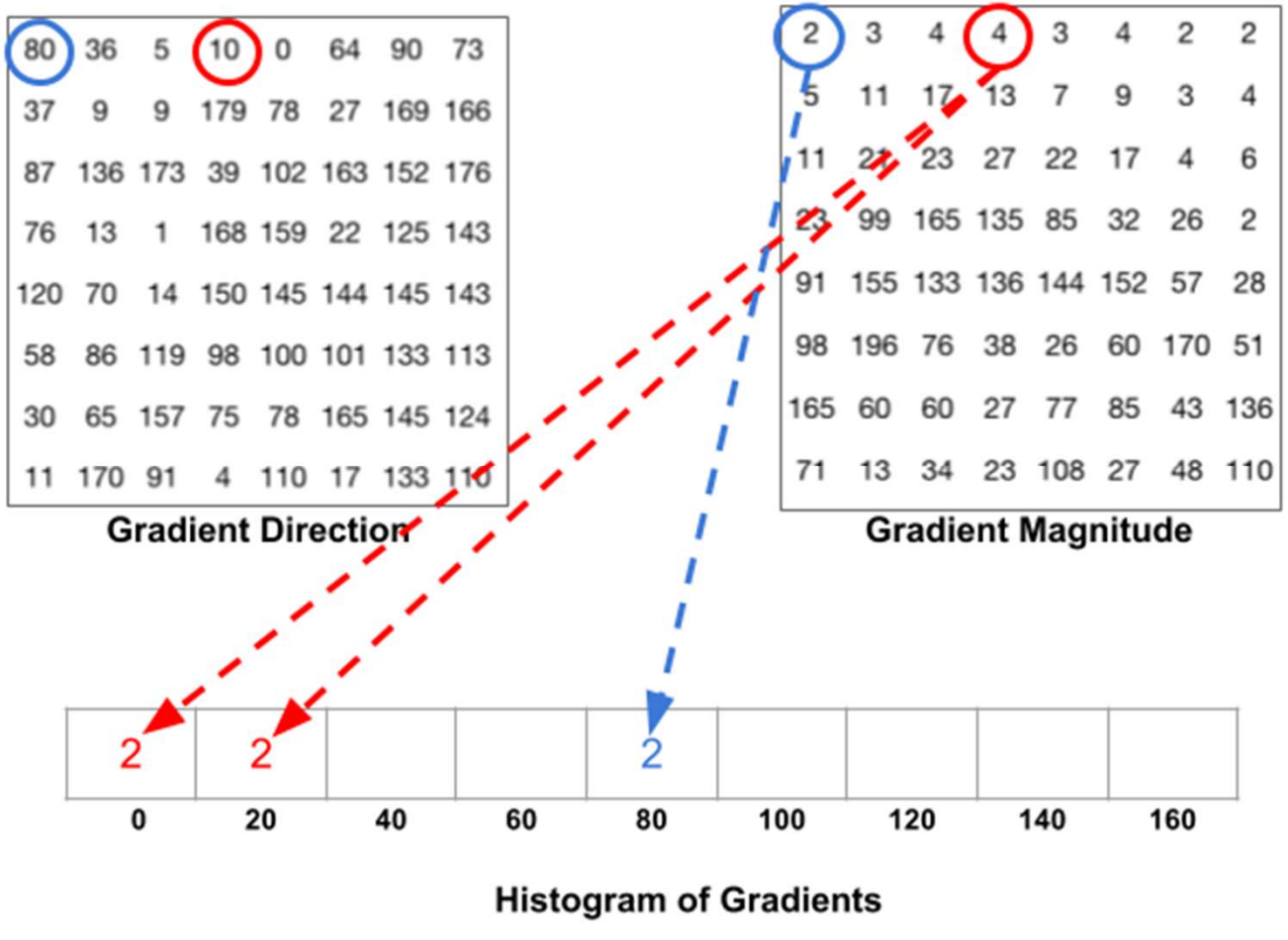
2. Para hacer el histograma más estable ante pequeñas distorsiones en la imagen, se divide ésta en celdas de 8x8. Las magnitudes del gradiente en cada celda de 64 píxeles se acumulan en 9 "buckets" en función de su dirección (de 20 en 20 grados, de 0º a 180º).
   NOTA: Si la dirección del gradiente cae justo entre dos "buckets", su magnitud se reparte entre ellos.

Apéndice
Detección de objetos

Soluciones clásicas

HOG [Histogram of Oriented Gradients]

U.S. Patent 1986 – CVPR'2005



Apéndice
Detección de objetos

Soluciones clásicas

HOG [Histogram of Oriented Gradients]
U.S. Patent 1986 – CVPR'2005

3. Recorremos bloques de 2x2 celdas (16x16 píxeles). Para cada bloque, 4 histogramas de 4 celdas se concatenan en un vector de dimensión 36 (9x4) y se normaliza (para que tenga magnitud 1).

El vector HOG final de la imagen es la concatenación de todos los vectores de bloques y se puede utilizar como entrada para un clasificador (p.ej. SVM).

Apéndice
Detección de objetos

Soluciones clásicas

Segmentación de imágenes

Algoritmo de Felsenszwalb (IJCV'2004)

Representa una imagen como un grafo no dirigido G=(V,E), con un vértice por píxel y una arista que conecta dos píxeles con un peso proporcional a la disimilitud entre los píxeles (color, posición, intensidad…).

La imagen se segmenta dividiendo en conjunto de vértices en múltiples componentes conectadas (píxeles similares pertenecerán a la misma componente conexa, píxeles distintos serán asignados a componentes distintas).

Apéndice
Detección de objetos

Soluciones clásicas

Segmentación de imágenes

Algoritmo de Felsenszwalb (IJCV'2004)

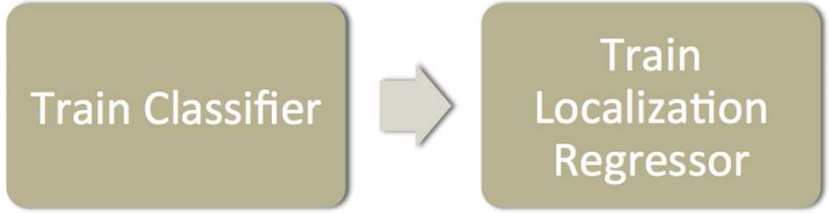
Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

OverFeat (NYU'2013)

Modelo pionero a la hora de integrar en una única red CNN la detección, localización y clasificación de objetos.

IDEA:



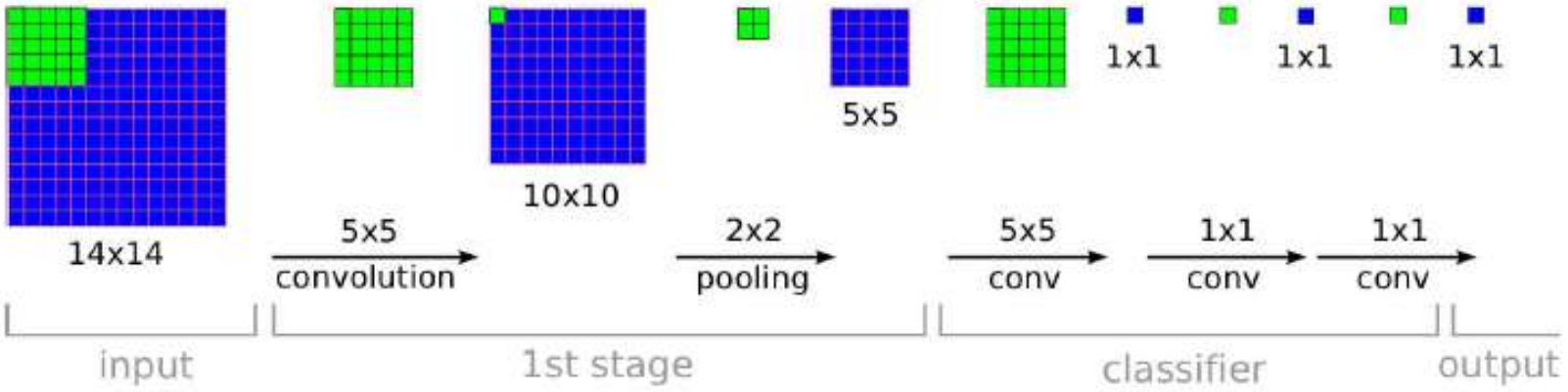
1. Clasificar imágenes en diferentes localizaciones sobre regiones en múltiples escalas de la imagen (con una ventana deslizante).
2. Predecir la caja [bounding box] que envuelve los objetos con un regresor entrenado sobre la propia red convolutiva.

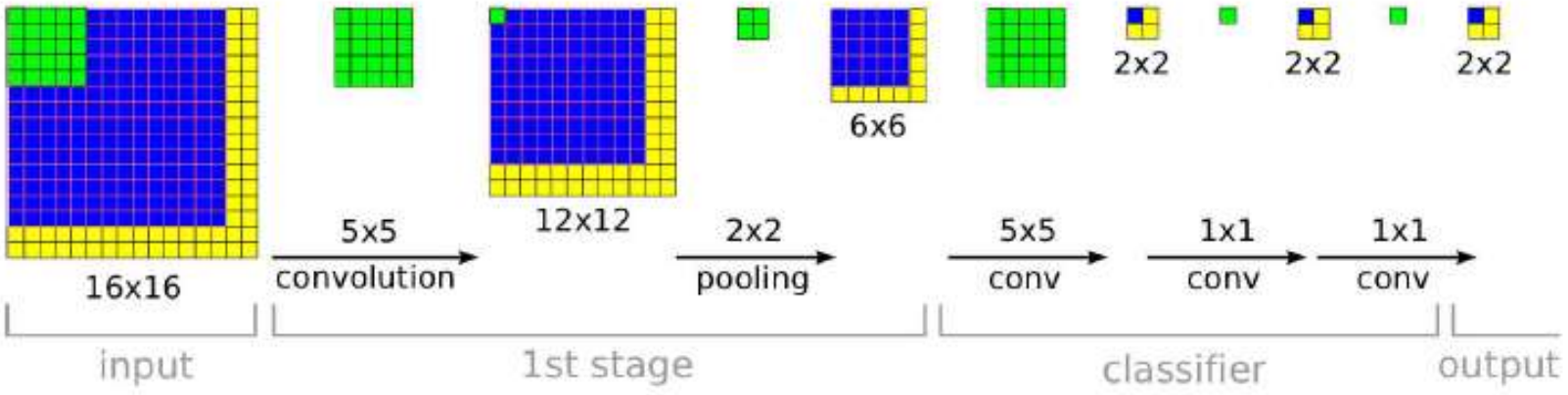
Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

OverFeat (NYU'2013)

Multi-scale sliding window using CNNs





Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

OverFeat (NYU'2013)

ENTRENAMIENTO:

1. Red CNN (similar a AlexNet).
2. Reemplazar la capa de salida de clasificación con un regresor por clase que predice la localización del objeto: 4 salidas (x_{left}, x_{right}, y_{top}, y_{bottom}).

USO:

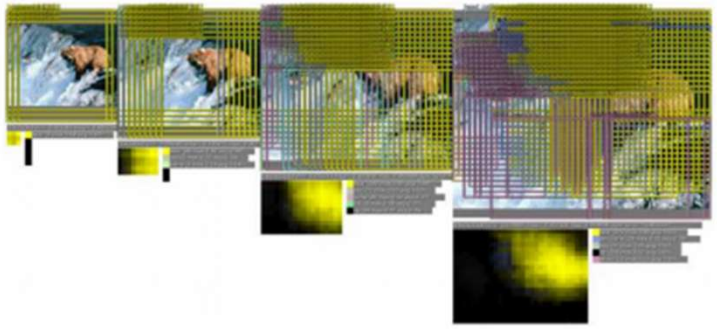
1. Clasificar usando la red CNN.
2. Predecir la localización para las regiones clasificadas.
3. Combinar localizaciones solapadas.

Apéndice
Detección de objetos

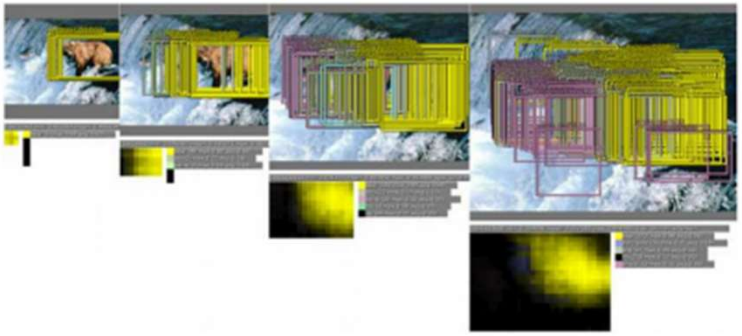
Soluciones basadas en "deep learning"

OverFeat (NYU'2013)

Window positions + score maps



Box regression outputs



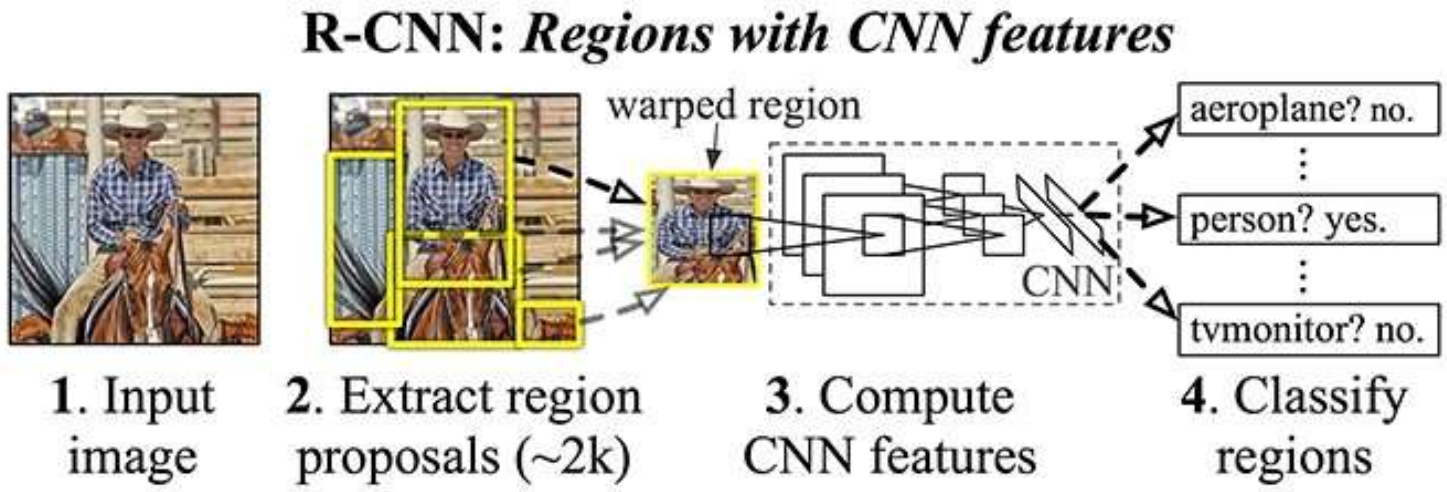
Final Predictions

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

R-CNN (UC Berkeley '2014)

Regions with CNN features / Region-based CNN



Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

R-CNN (UC Berkeley '2014)

ALGORITMO

1. Se identifica un número manejable de regiones candidatas (~2k) que puedan contener un objeto: búsqueda selectiva para obtener regiones de interés [RoI: region of interest].

2. Se utiliza una red convolutiva para clasificar esas regiones de interés (de forma completamente independiente cada una de ellas).

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

Búsqueda selectiva [selective search]

Algoritmo para proponer regiones que potencialmente contengan objetos de interés:

1. Aplicar el algoritmo de segmentación de imágenes de Felzenszwalb y Huttenlocher (IJCV'2004) para obtener un conjunto inicial de regiones en la imagen.

2. Utilizar un algoritmo greedy para ir agrupando regiones iterativamente (clustering jerárquico aglomerativo).

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

Búsqueda selectiva [selective search]

Algorithm 1: Hierarchical Grouping Algorithm

DontPrintSemicolon **Input**: (colour) image
Output: Set of object location hypotheses L

Obtain initial regions $R = \{r_1, \cdots, r_n\}$ using Felzenszwalb and Huttenlocher (2004) Initialise similarity set $S = \emptyset$;
foreach *Neighbouring region pair* (r_i, r_j) **do**
  Calculate similarity $s(r_i, r_j)$;
  $S = S \cup s(r_i, r_j)$;

while $S \neq \emptyset$ **do**
  Get highest similarity $s(r_i, r_j) = \max(S)$;
  Merge corresponding regions $r_t = r_i \cup r_j$;
  Remove similarities regarding r_i : $S = S \setminus s(r_i, r_*)$;
  Remove similarities regarding r_j : $S = S \setminus s(r_*, r_j)$;
  Calculate similarity set S_t between r_t and its neighbours;
  $S = S \cup S_t$;
  $R = R \cup r_t$;
Extract object location boxes L from all regions in R;

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

Búsqueda selectiva [selective search]

Se pueden usan cuatro medidas de similitud complementarias para agrupar regiones:

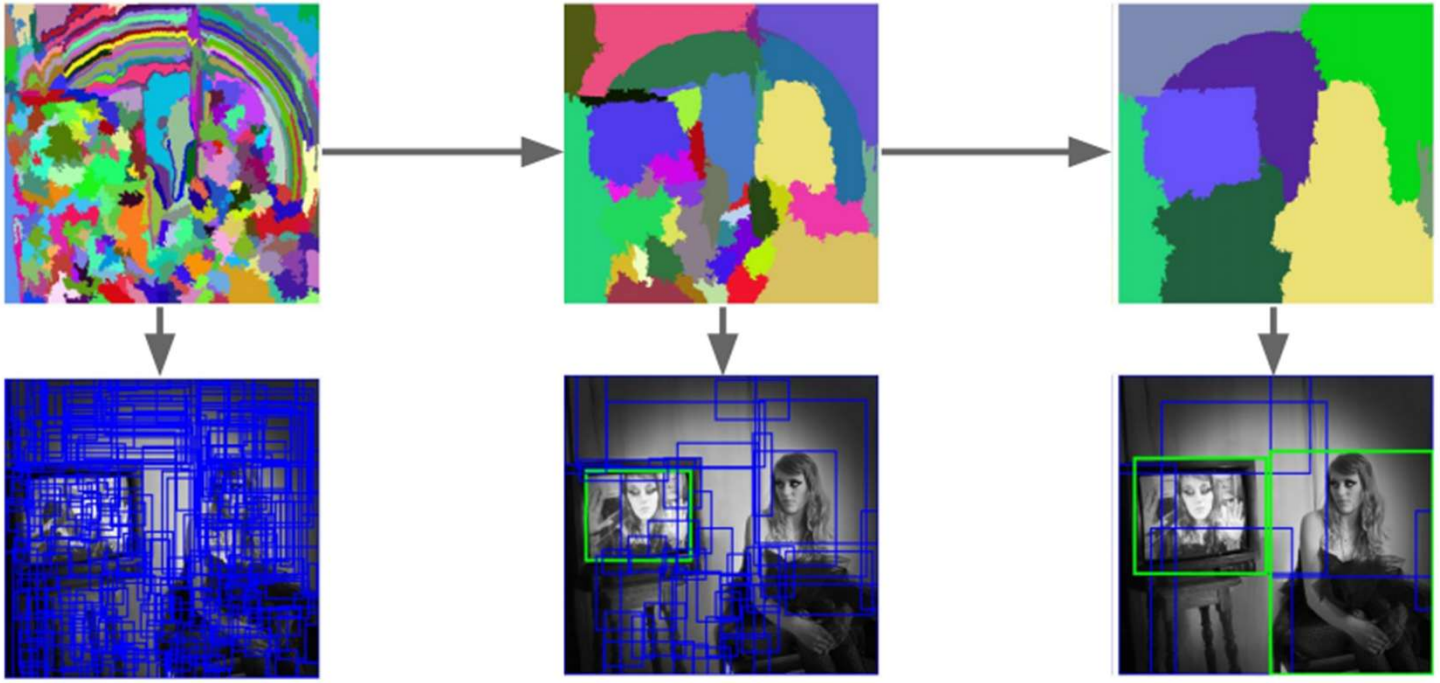
- Color
- Textura, p.ej. SIFT (ICCV'1999)
- Tamaño (se procuran fusionar regiones pequeñas)
- Forma (regiones que rellenan huecos en otras)

Usando combinaciones, se consiguen distintas estrategias de selección de regiones potencialmente interesantes…

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

Búsqueda selectiva [selective search]



Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

R-CNN (UC Berkeley '2014)

USO

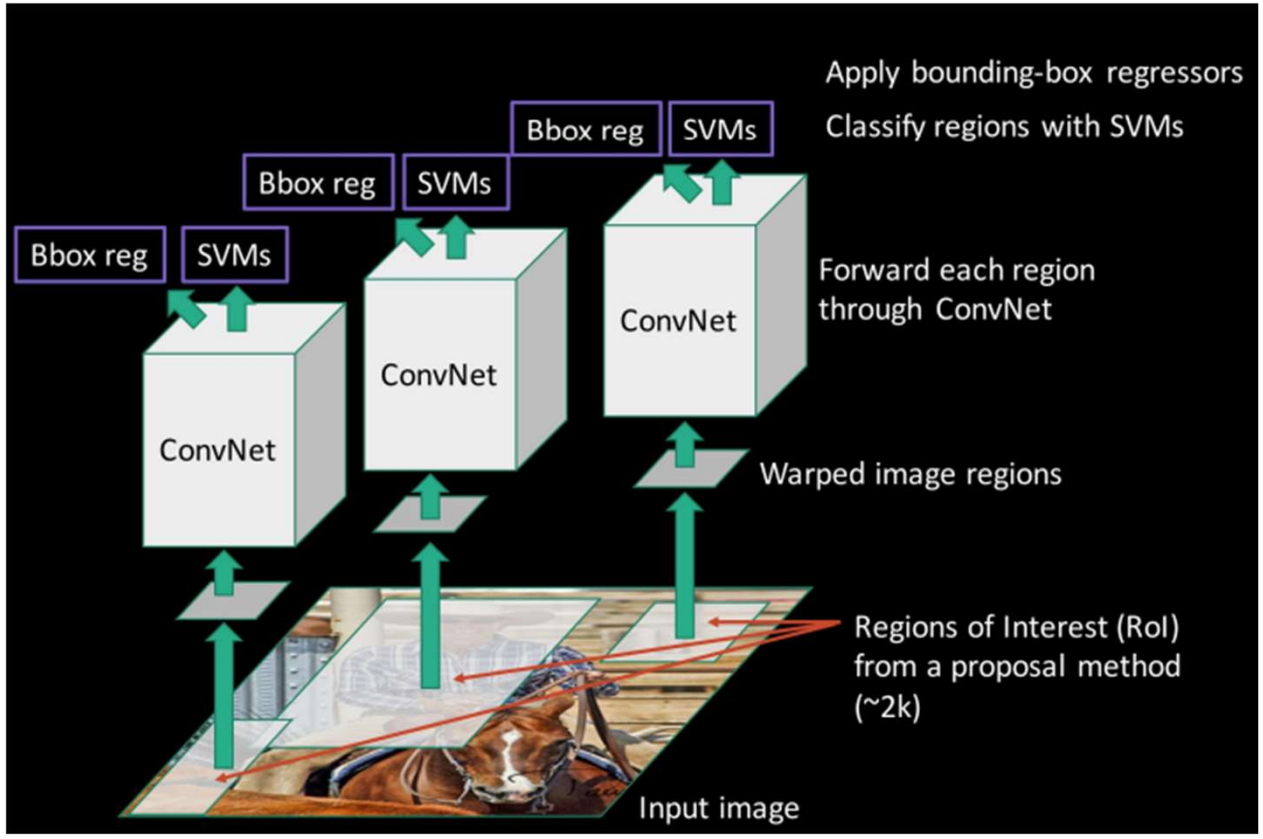
Dada una red CNN entrenada para clasificar imágenes:

1. Se proponen regiones de interés (búsqueda selectiva).
2. Se deforman [warp] las regiones de interés para que tengan el tamaño requerido por la CNN.
3. Se clasifican las regiones (añadiendo una clase extra correspondiente al fondo, sin objeto de interés).
4. Para reducir errores de localización, se usa un regresor para predecir la caja que envuelve el objeto.

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

R-CNN (UC Berkeley '2014)

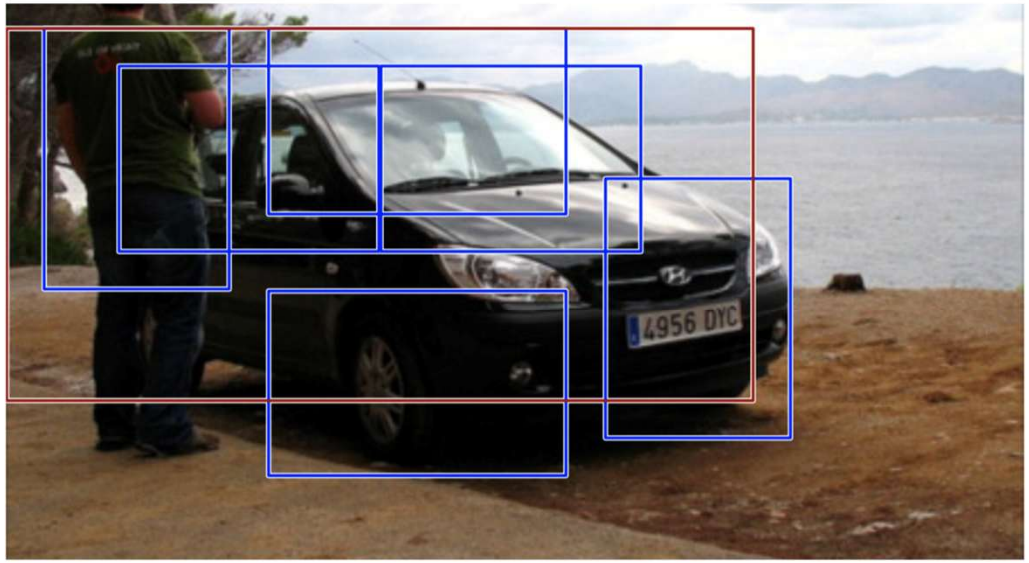


Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

R-CNN (UC Berkeley '2014)

AJUSTES: Supresión de no-máximos



Before non-max suppression

After non-max suppression

Para evitar la detección repetida del mismo objeto, se utiliza un algoritmo greedy que elimina solapamientos.

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

Fast R-CNN (Microsoft Research '2014)

R-CNN es costoso computacionalmente: búsqueda selectiva para producir 2k RoI, clasificación de 2k RoI, regresión para determinar la localización de los objetos de interés.

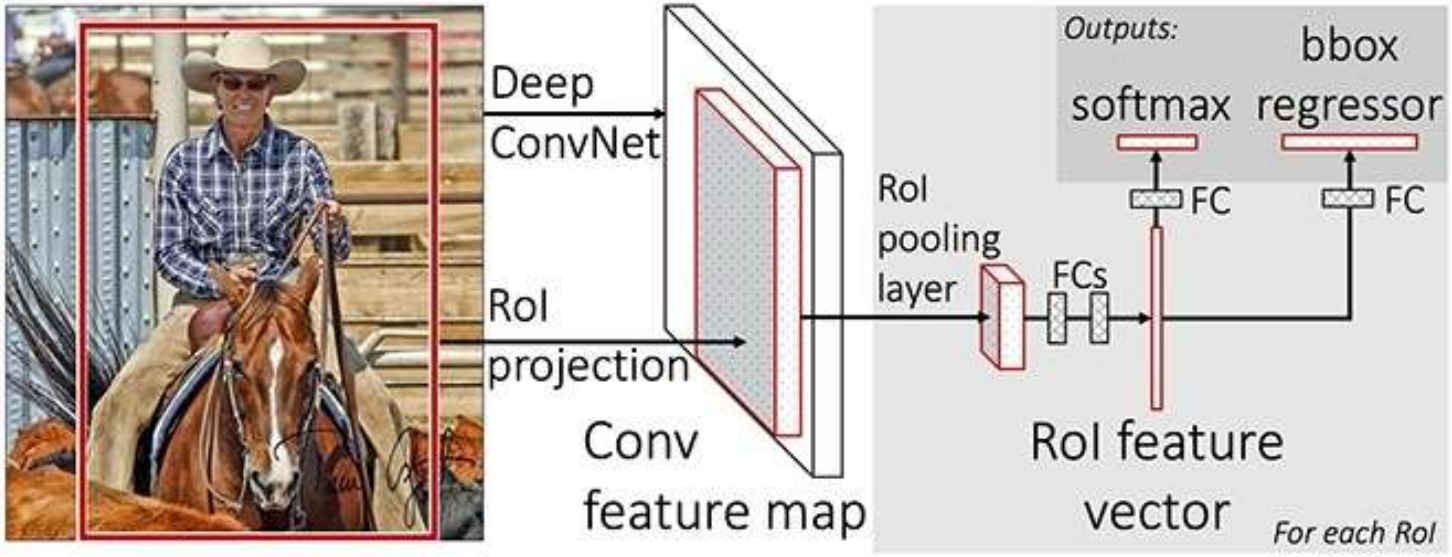
Fast R-CNN unifica los modelos independientes para aprovechar los cálculos compartidos: agrega todas las regiones en una única CNN que se usa tanto de clasificador como de regresor..

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

Fast R-CNN (Microsoft Research '2014)

Region of Interest (RoI) Pooling

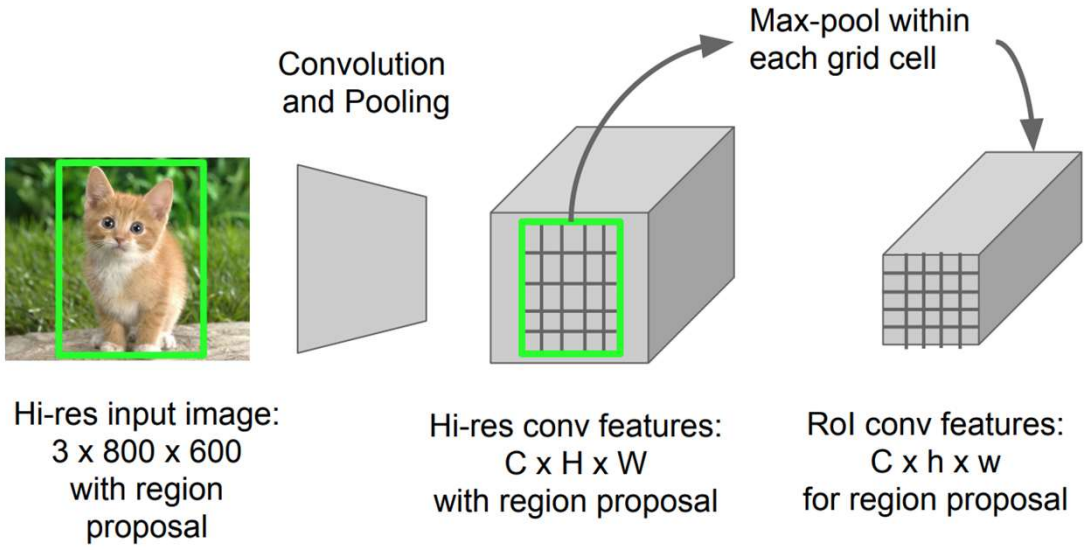


Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

Fast R-CNN (Microsoft Research '2014)

RoI Pooling (como max pooling), convierte una imagen de tamaño hxw en una ventana pequeña de tamaño fijo HxW.

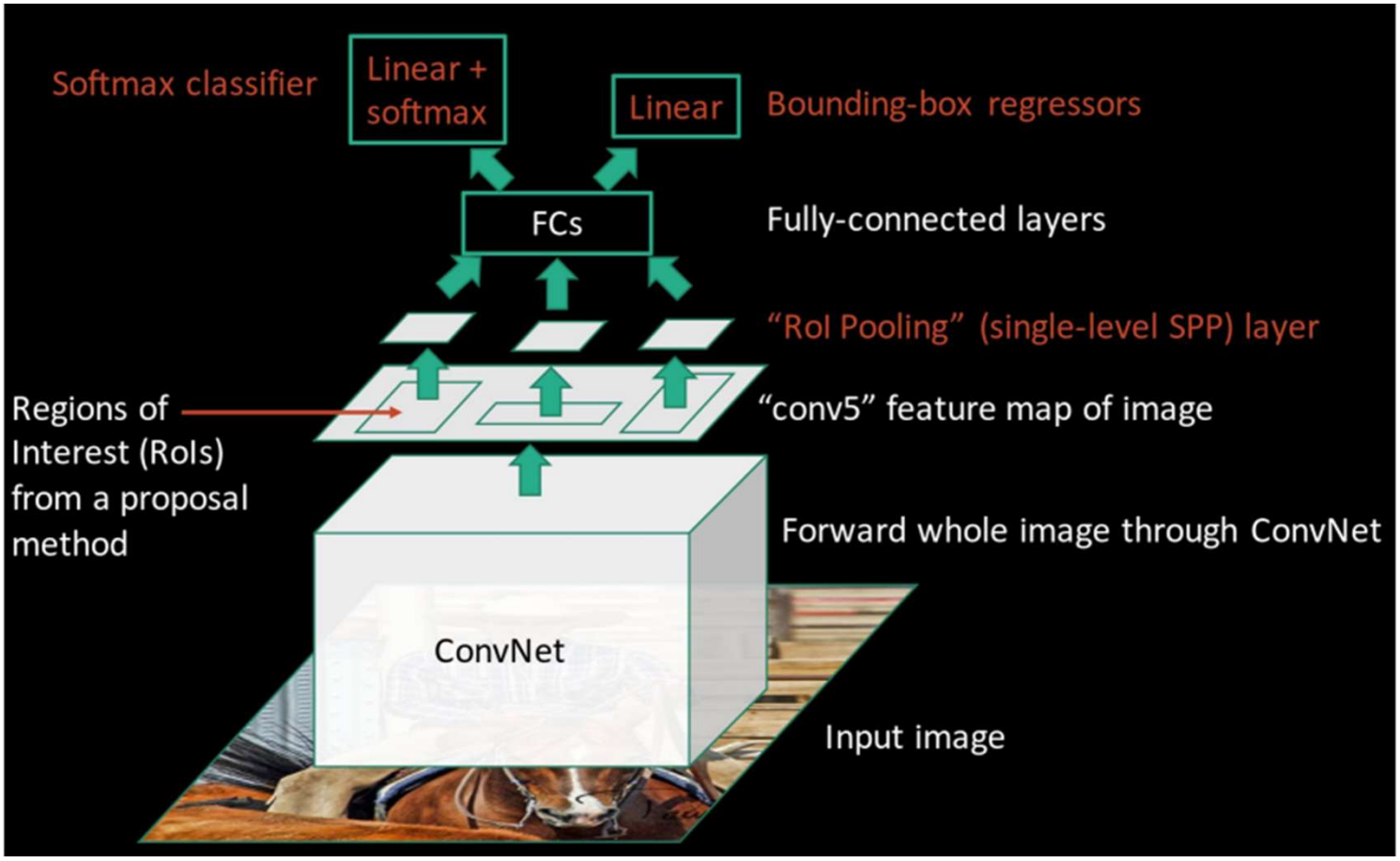


La entrada se divide en un grid HxW

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

Fast R-CNN (Microsoft Research '2014)



Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

Fast R-CNN (Microsoft Research '2014)

	R-CNN	Fast R-CNN
Training Time:	84 hours	**9.5 hours**
(Speedup)	1x	**8.8x**
Test time per image	47 seconds	**0.32 seconds**
(Speedup)	1x	**146x**
mAP (VOC 2007)	66.0	**66.9**

Apéndice
Detección de objetos

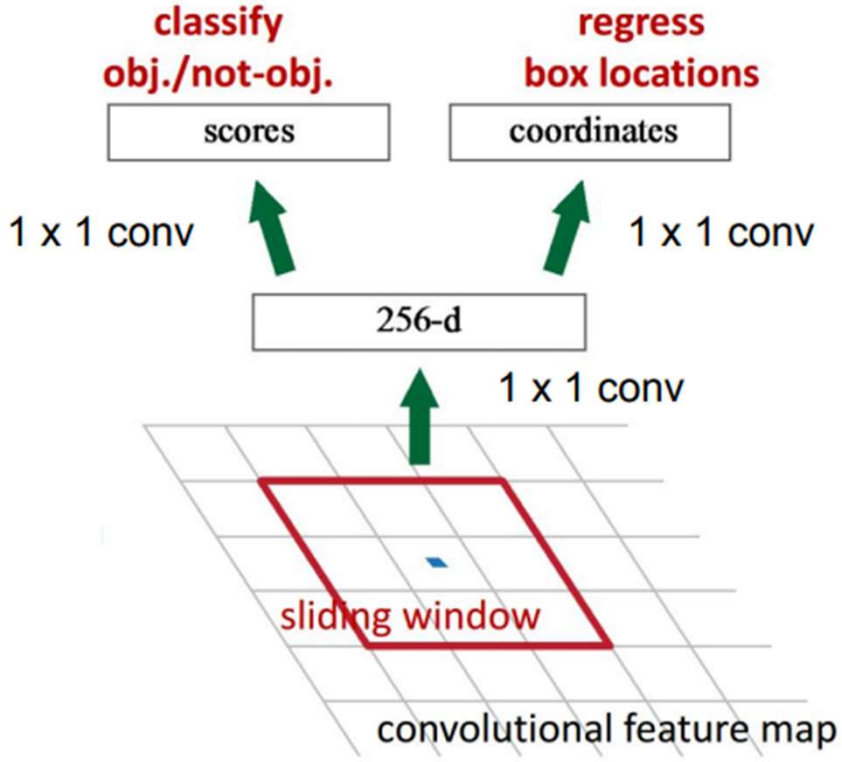
Soluciones basadas en "deep learning"

Faster R-CNN (Microsoft Research '2016)

Fast R-CNN no incluía la proposición de regiones de interés en la propia red, incluyámosla…

RPN

Region Proposal Network

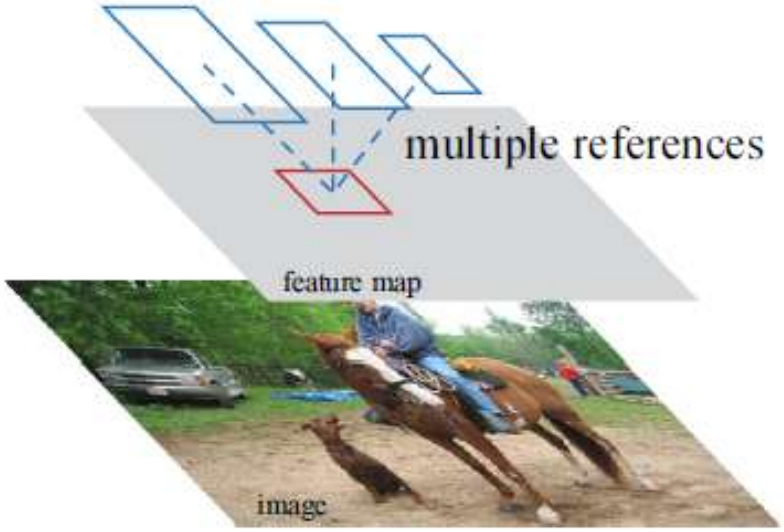


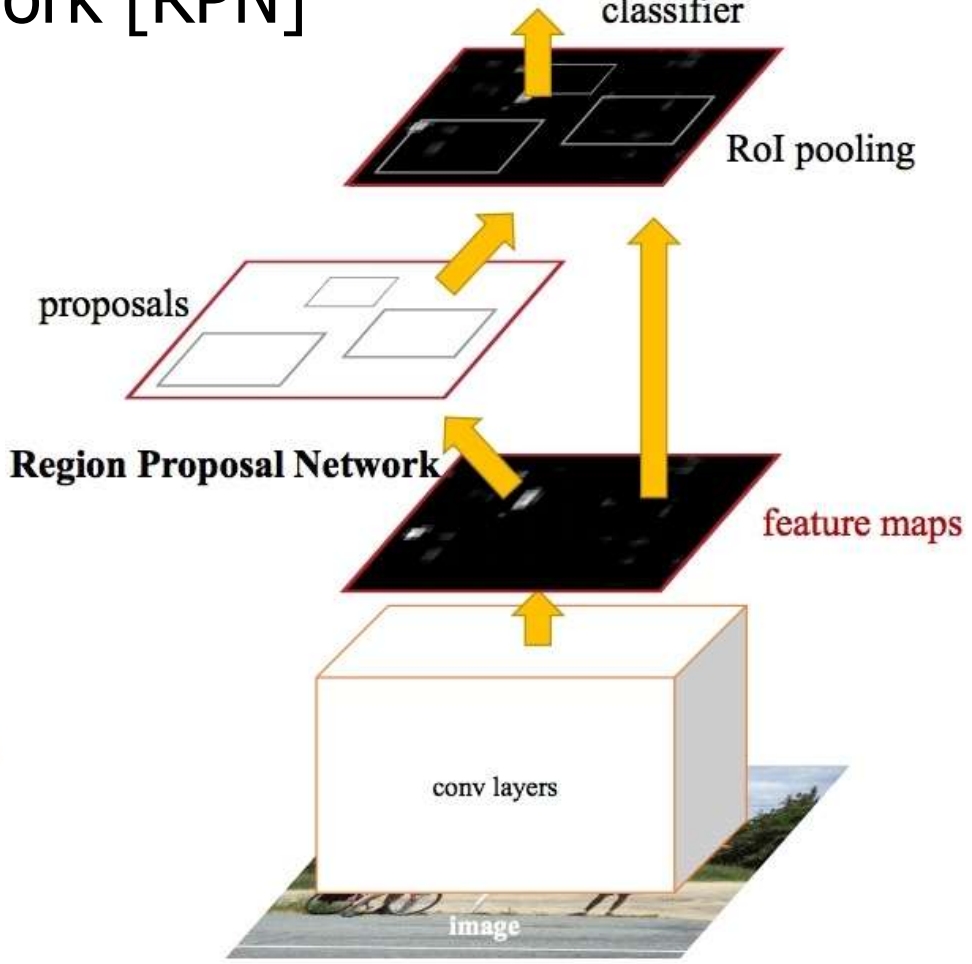
Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

Faster R-CNN (Microsoft Research '2016)

Region proposal network [RPN]





Apéndice
Detección de objetos

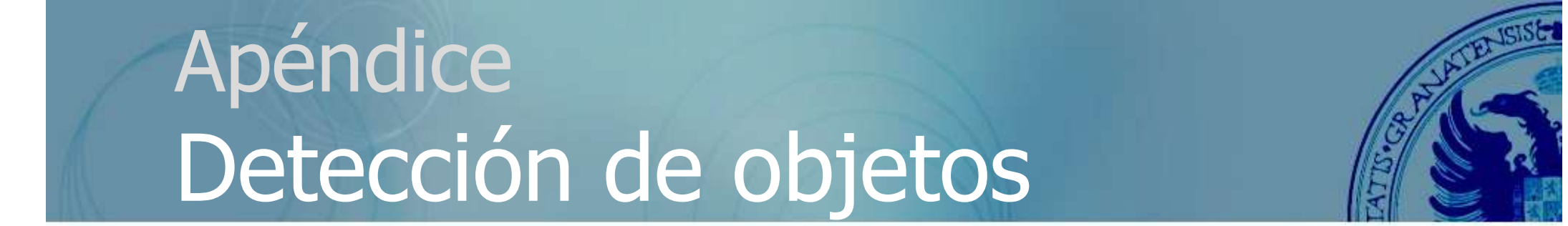
Soluciones basadas en "deep learning"

Faster R-CNN (Microsoft Research '2016)

Region proposal network [RPN]

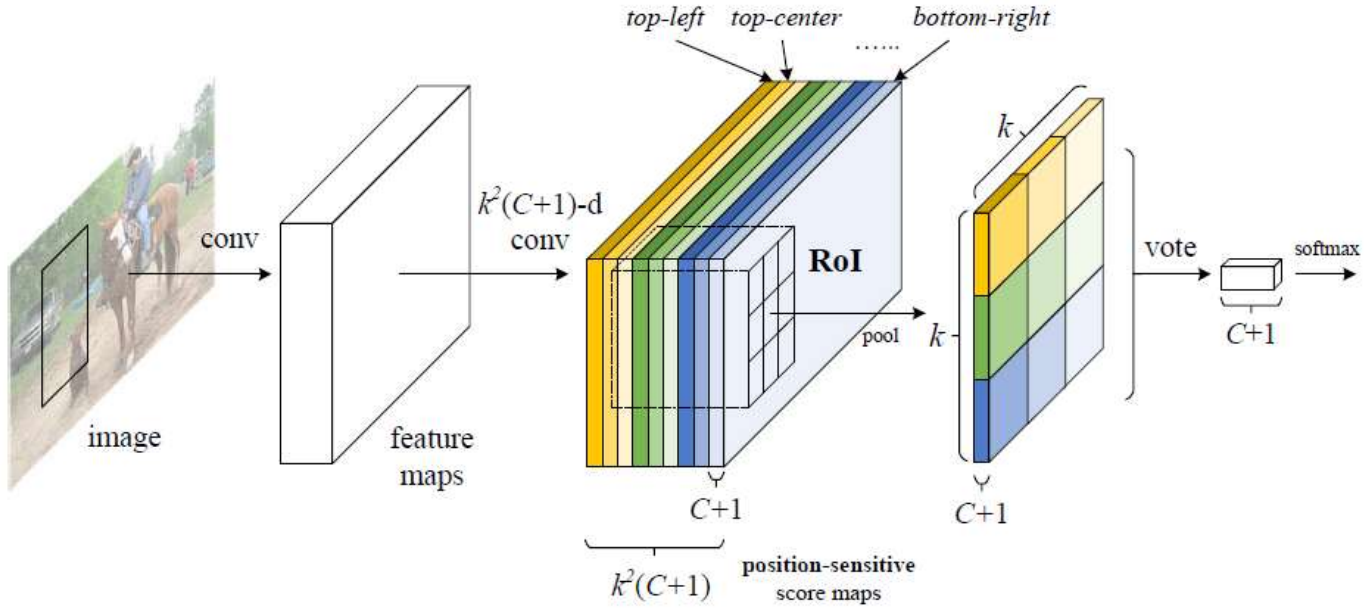
	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image (with proposals)	50 seconds	2 seconds	**0.2 seconds**
(Speedup)	1x	25x	**250x**
mAP (VOC 2007)	66.0	**66.9**	**66.9**

Apéndice
Detección de objetos



Soluciones basadas en "deep learning"

Más variantes… de Faster R-CNN

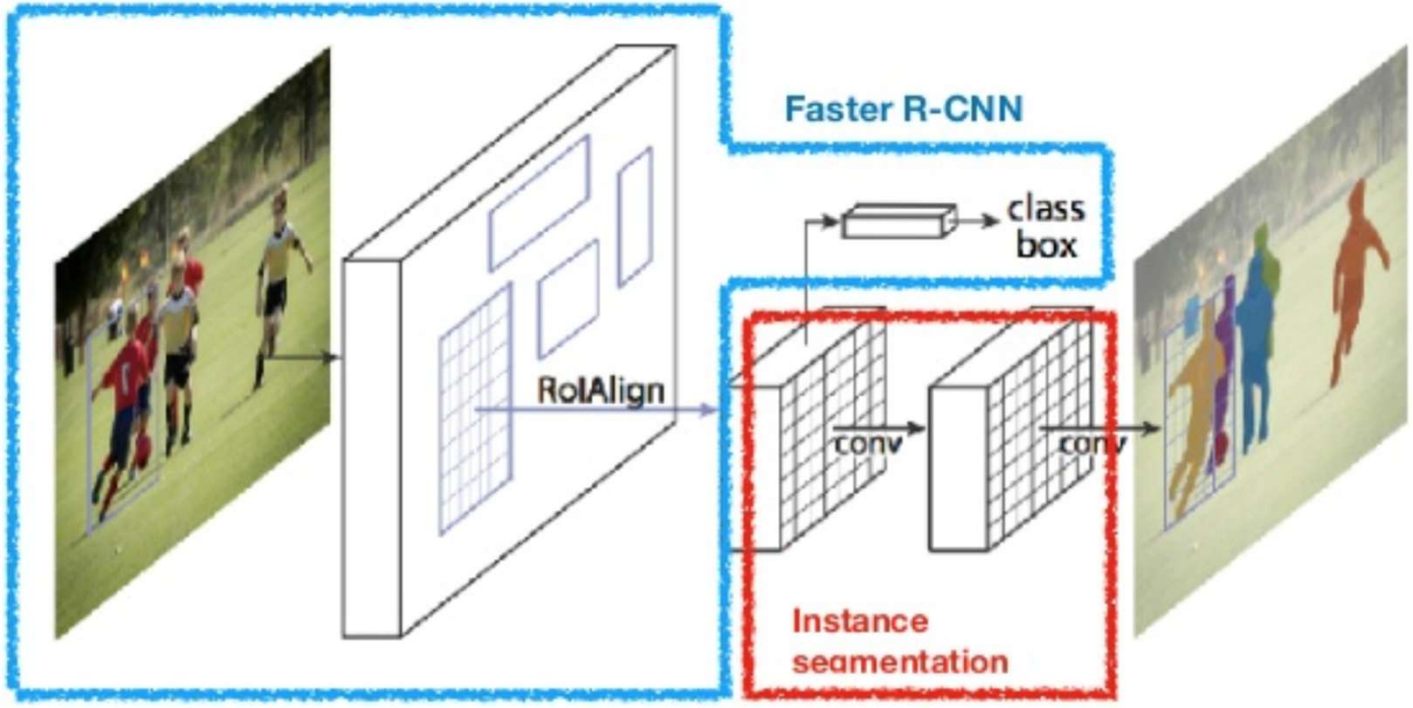


R-FCN [Region-based Fully Convolutional Network]
Microsoft Research, 2016

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

Más variantes… de Faster R-CNN



Mask R-CNN

Segmentación de imágenes a nivel de píxeles

Facebook AI Research, ICCV'2017

Apéndice
Detección de objetos

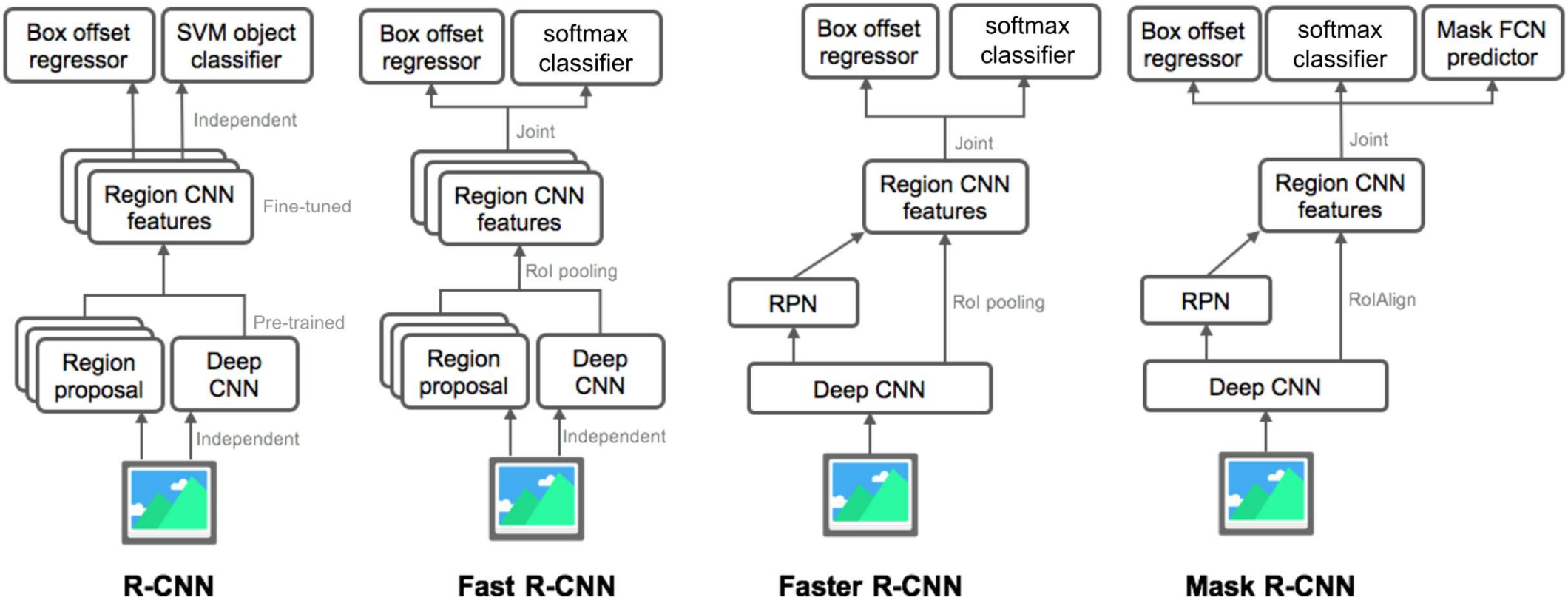
Soluciones basadas en "deep learning"

Más variantes… de Faster R-CNN

Mask R-CNN (Facebook, ICCV'2017)

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"



La familia R-CNN

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

Detectores de una y dos etapas

El problema de los modelos de la familia R-CNN es que la detección sucede en dos fases: primero se proponen regiones de interés (búsqueda selectiva o RPN) y después se clasifican esas regiones de interés.

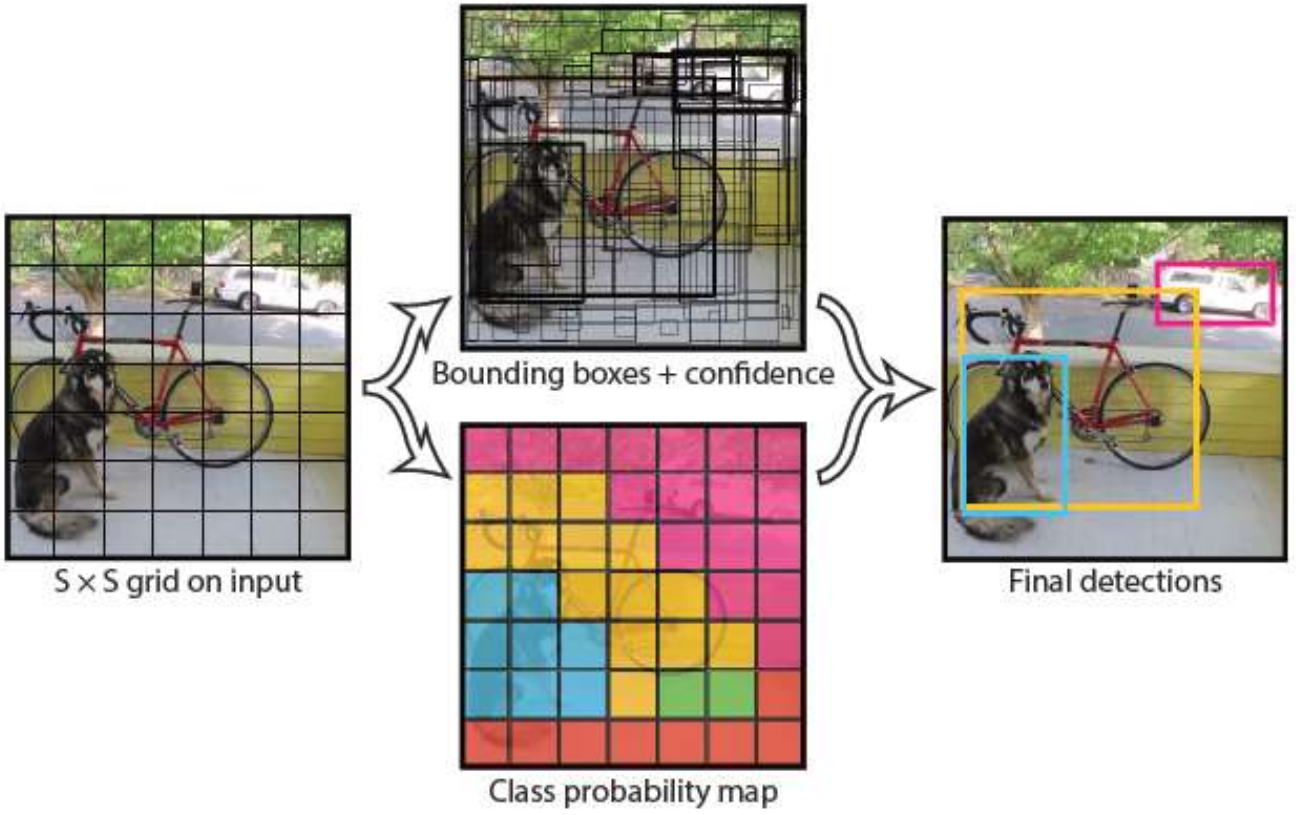
Existen algoritmos que evitan la fase de proposición de regiones y detectan los objetos directamente sobre sus posibles localizaciones (más rápido y simple, aunque tal vez algo menos efectivo), p.ej. YOLO, SSD, RetinaNet.

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

YOLO: You Only Look Once '2015

Primer detector de objetos en tiempo real (100Hz)



Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

YOLO: You Only Look Once '2015

El problema de detección como un problema de regresión:

- Se divide la imagen en un grid SxS.
- Para cada celda del grid, se predicen
  - B cajas (4 coordenadas + confianza)
  - C scores (C clases)

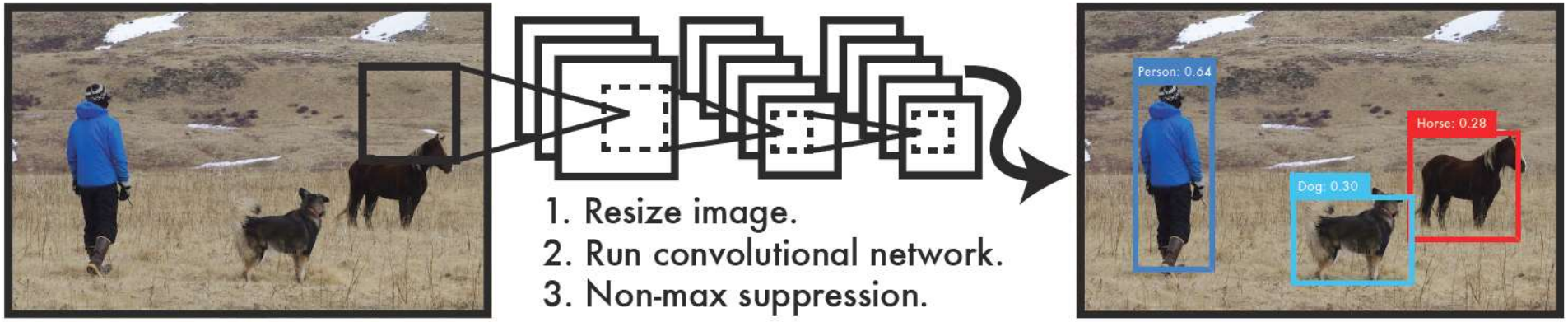
Red CNN de regresión
de una imagen a un tensor S x S x (5B+C)

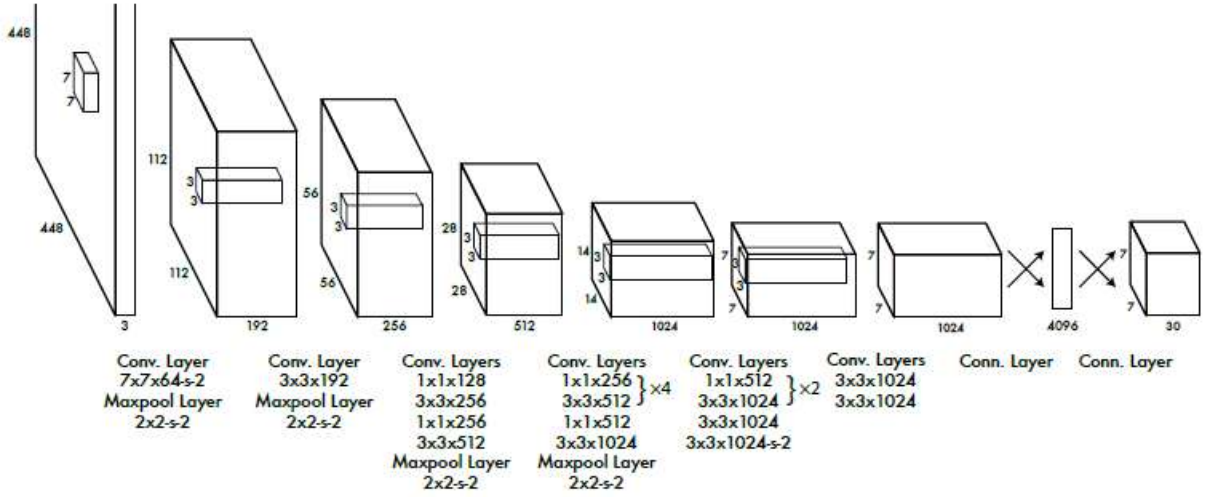
Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

YOLO: You Only Look Once '2015

Red convolutiva única



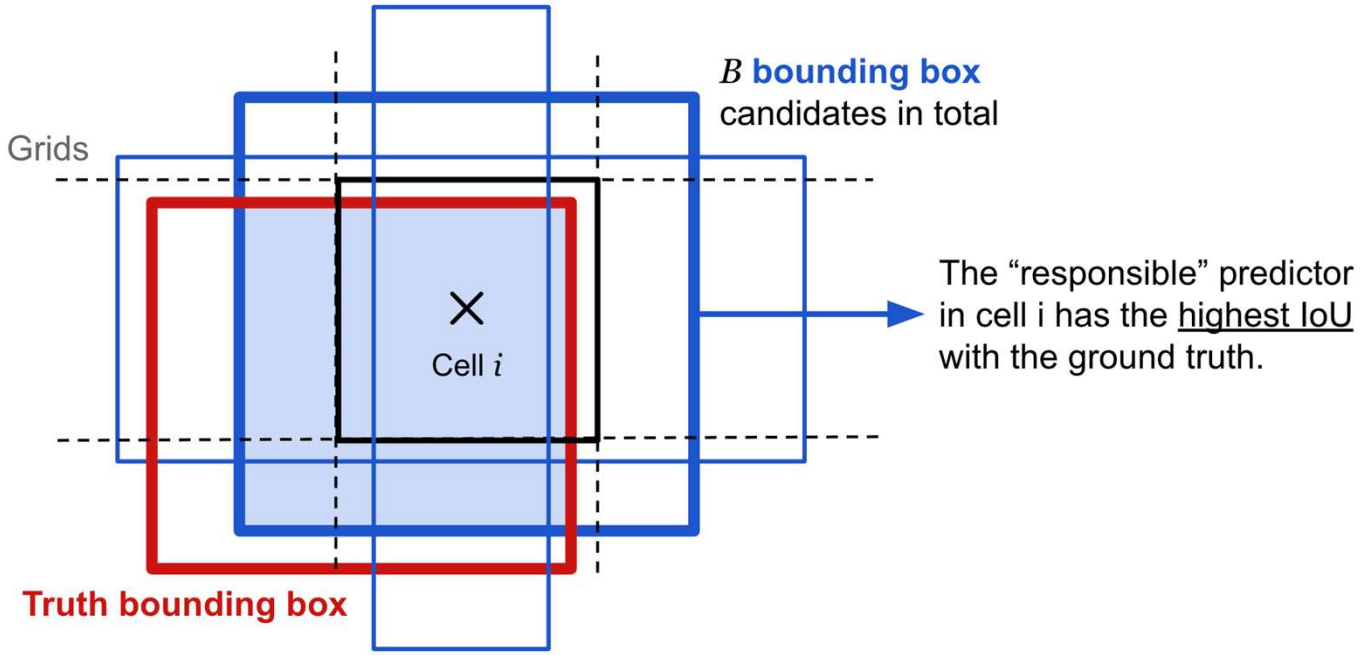


Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

YOLO: You Only Look Once '2015

Bounding box



Para cada celda,
se proponen B bounding boxes (SxSxB en total)

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

YOLO: You Only Look Once '2015

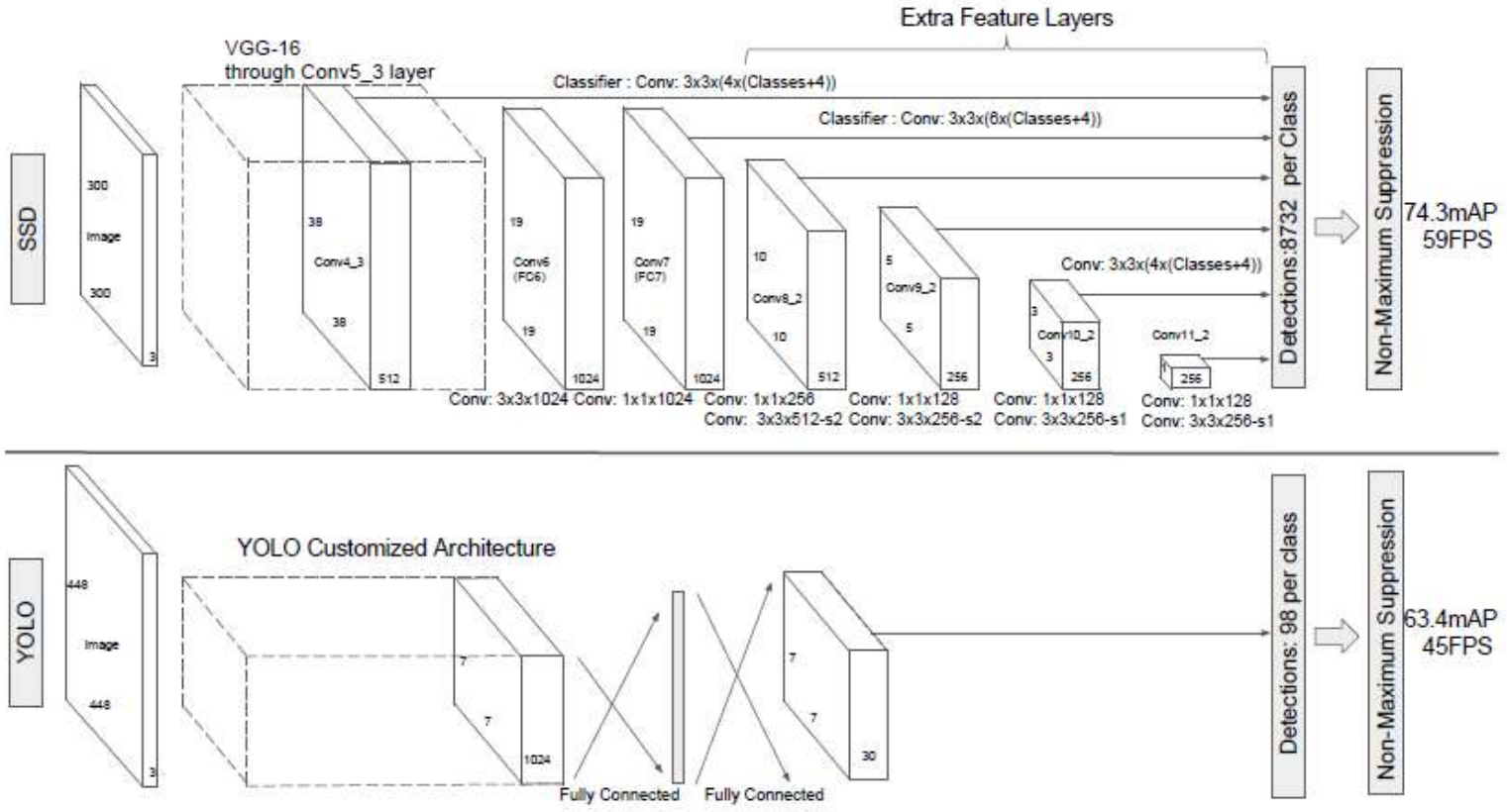
- YOLO es muy rápido
  (detección de objetos en tiempo real).

- Sin embargo,
  debido al número reducido de BB candidatas,
  no es demasiado bueno reconociendo
  objetos de formas irregulares
  o grupos de objetos pequeños.

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

Más variantes… de YOLO



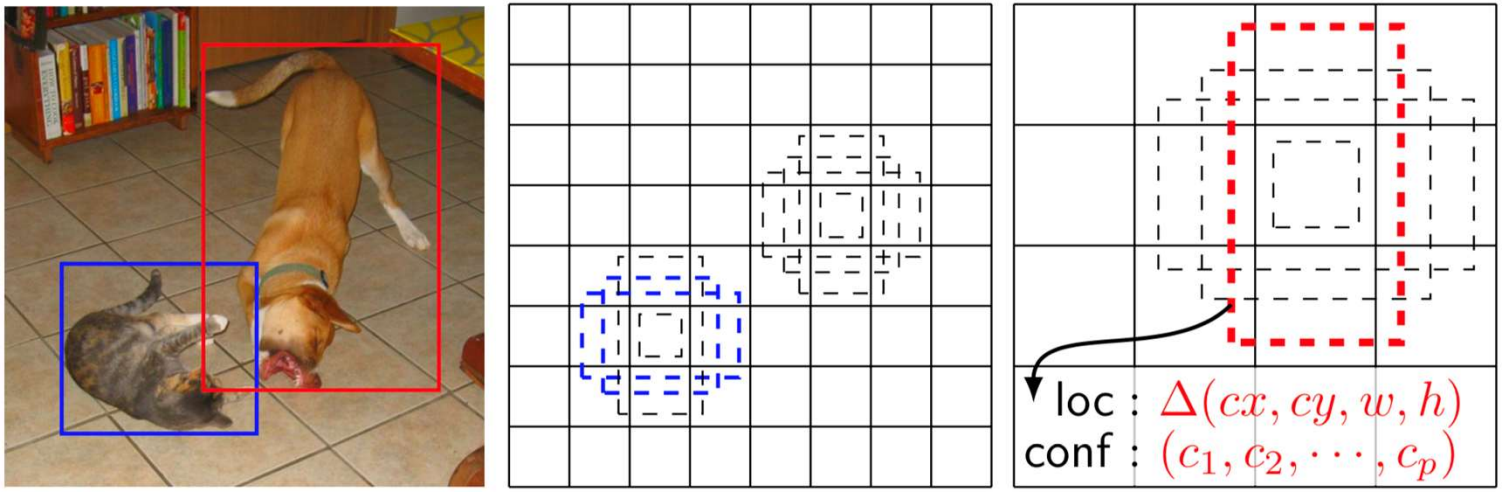
SSD [Single Shot Detector]

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

Más variantes… de YOLO

En lugar de dividir la imagen en un grid de tamaño arbitrario, se predicen los offsets para los "anchor boxes" de cada localización del mapa de características de la CNN.



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

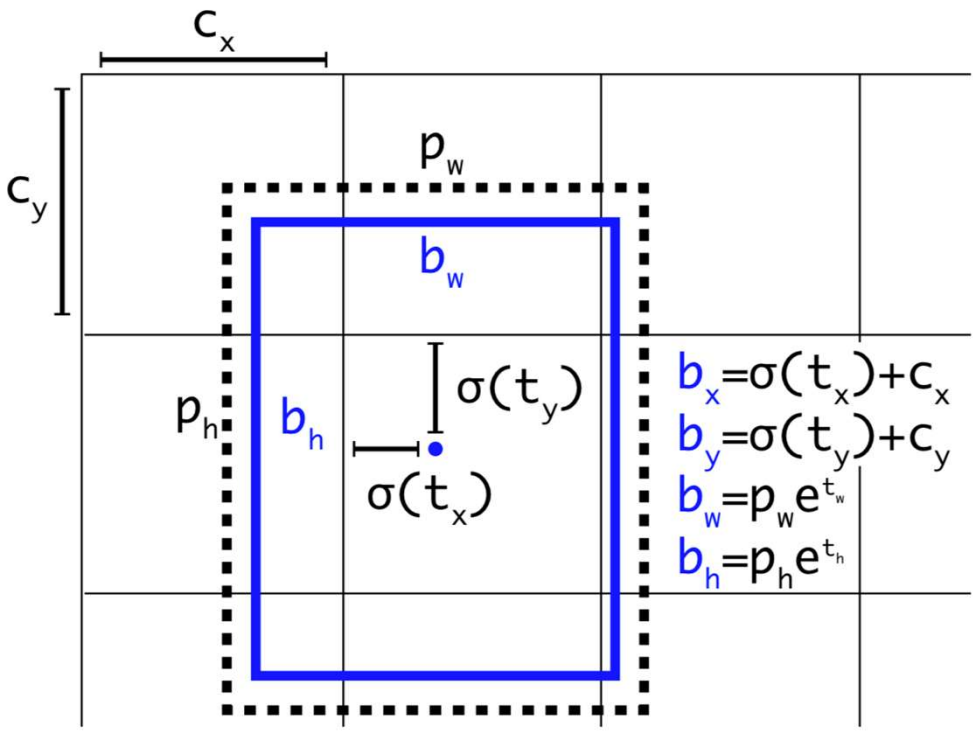
SSD [Single Shot Detector]

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

Más variantes… de YOLO

YOLOv2
bounding box
location prediction



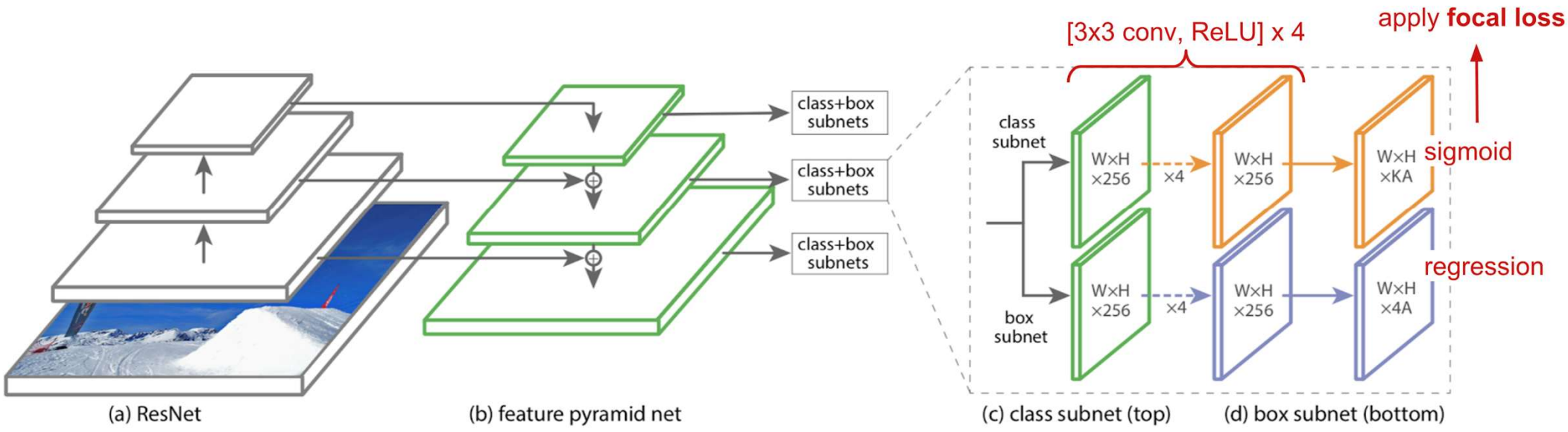
2016: **YOLOv2**, **YOLO9000**
2018: **YOLOv3**
2020: **YOLOv4**
2021: **YOLOF, YOLOX**
2024: **YOLOv10**

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

Más variantes… de YOLO

Featurized Image Pyramid + Focal Loss



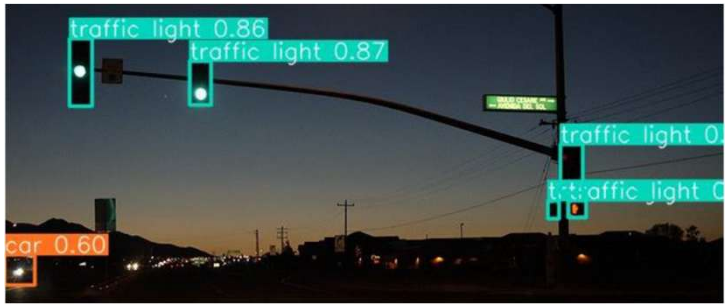
RetinaNet: Detección de objetos a distintas escalas

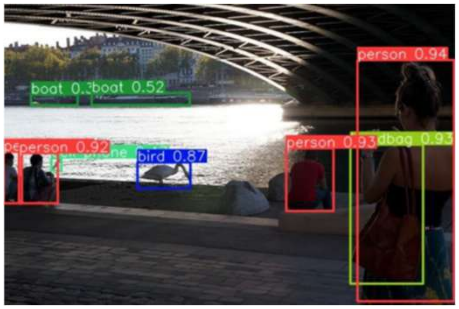
Facebook AI Research, 2018

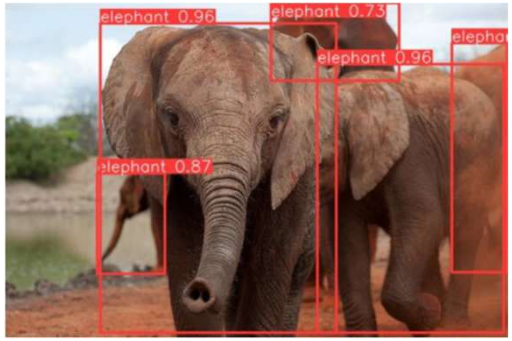
Apéndice
Detección de objetos

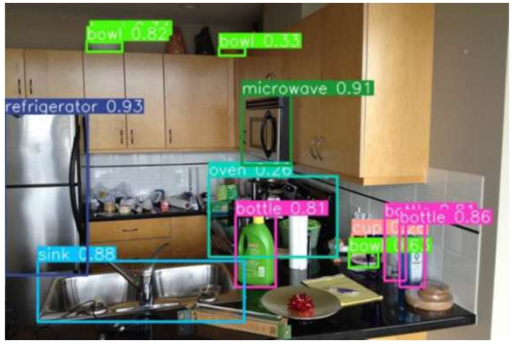
Soluciones basadas en "deep learning"

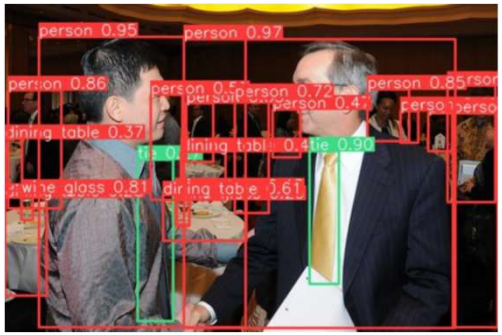
Más variantes… de YOLO







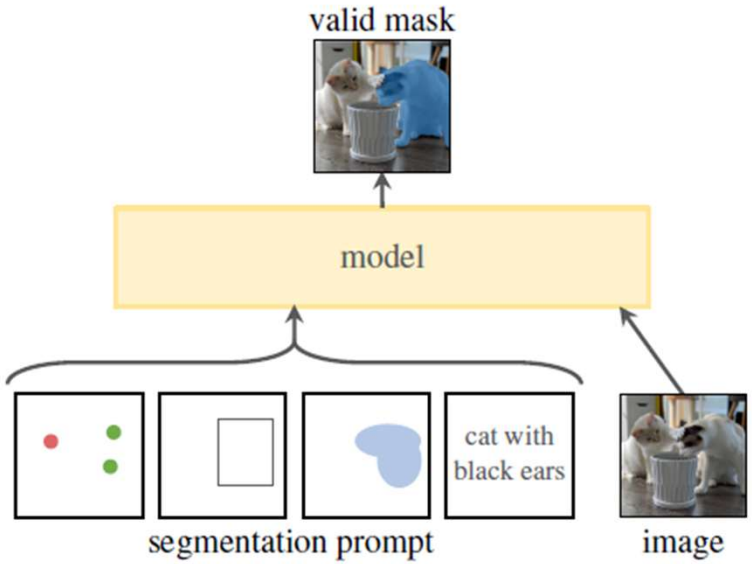




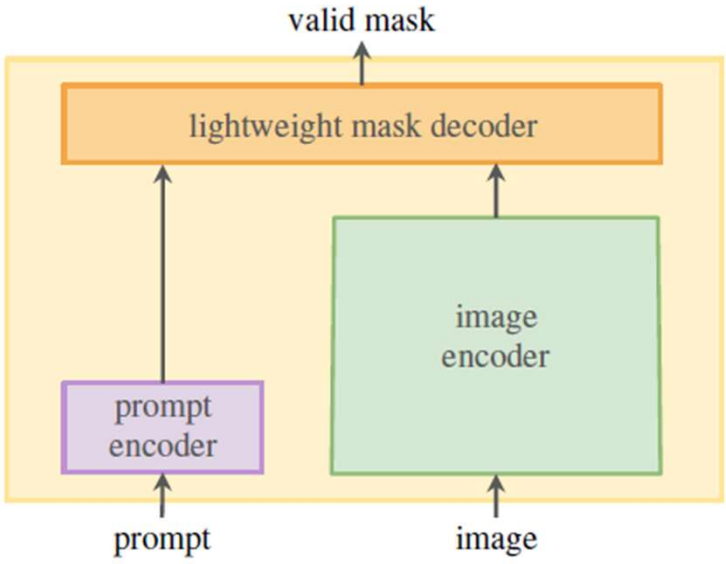
YOLOv10: Real-Time End-to-End Object Detection
NeurIPS'2024

Apéndice
Detección de objetos

SAM [Segment Anything Model]



(a) **Task**: promptable segmentation



(b) **Model**: Segment Anything Model (**SAM**)

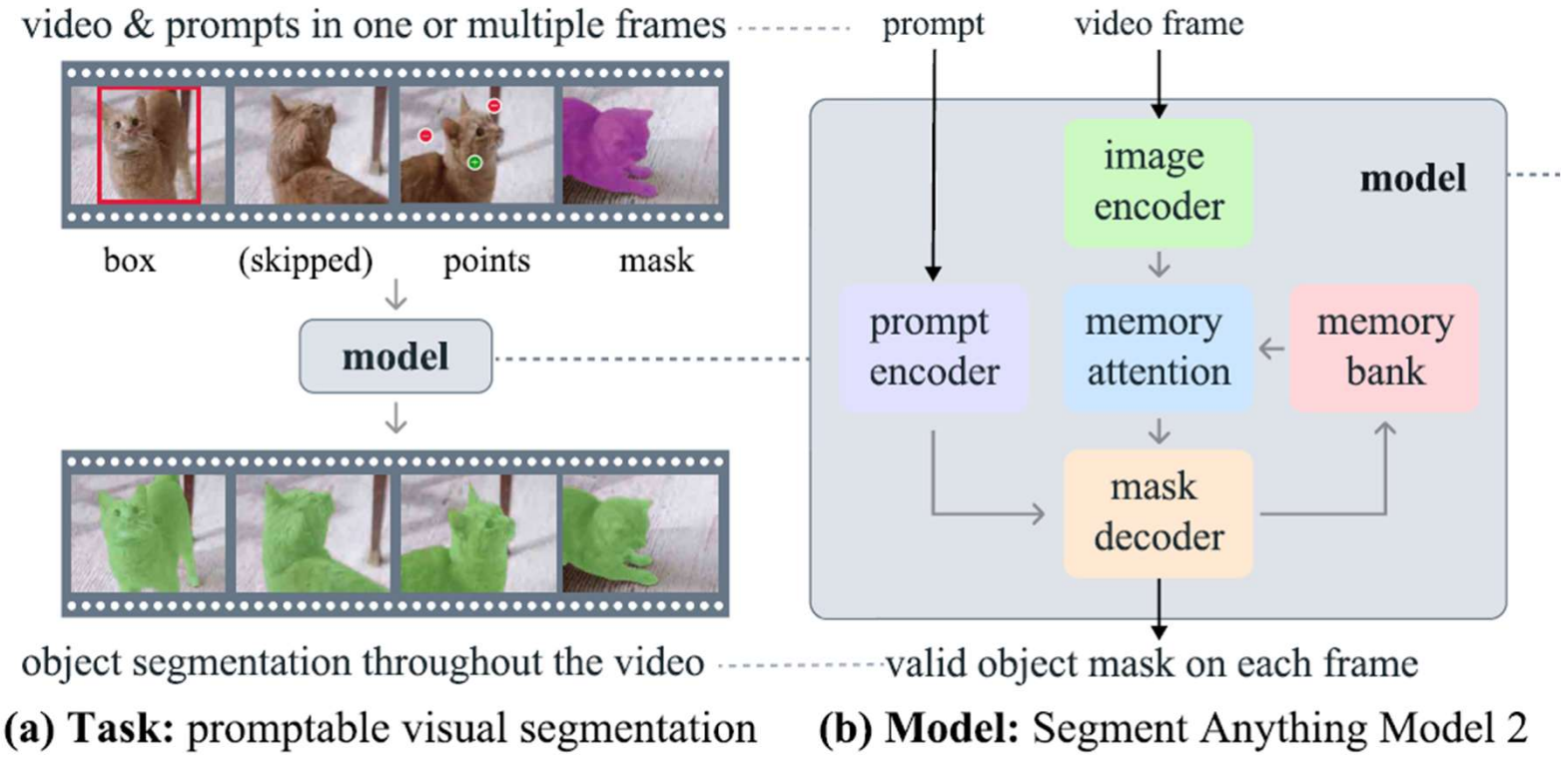
Meta AI

https://segment-anything.com/

SAM (2023)

Apéndice
Detección de objetos

SAM 2 [Segment Anything Model 2]



(a) Task: promptable visual segmentation **(b) Model:** Segment Anything Model 2

Meta AI

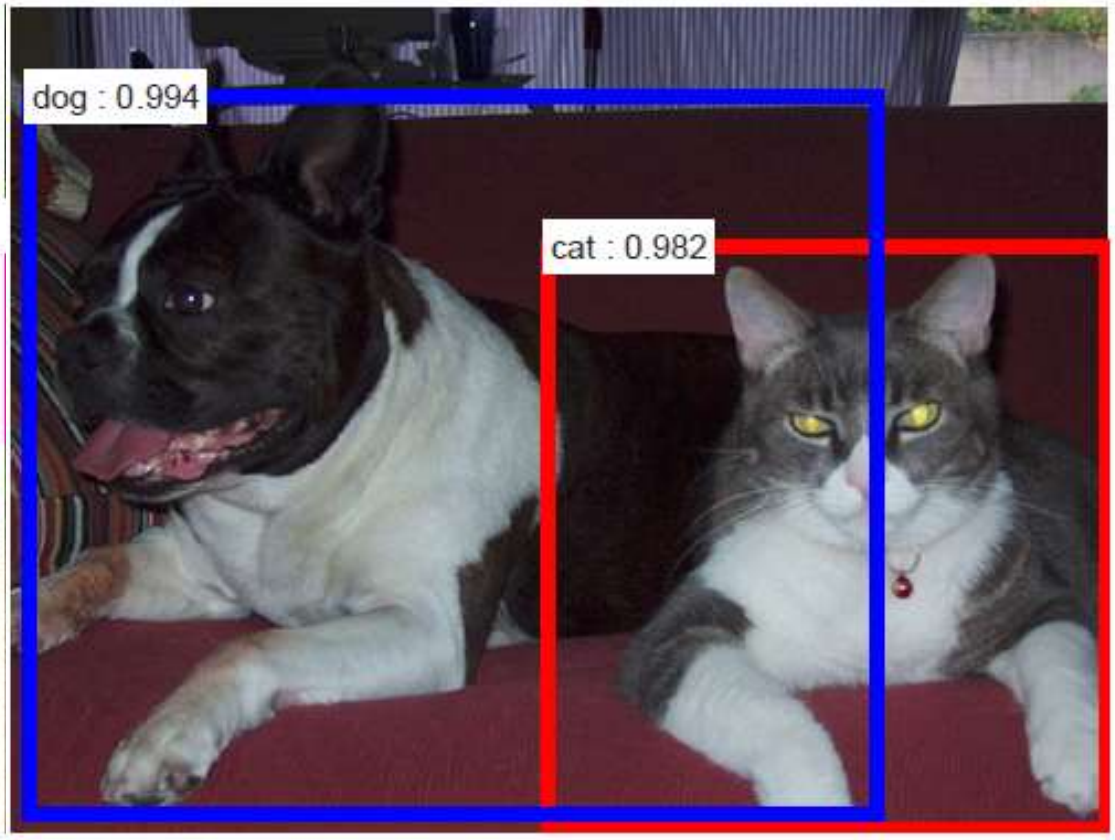
https://ai.meta.com/sam2/

SAM v2 (2024)

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

Resultados



Apéndice
Detección de objetos

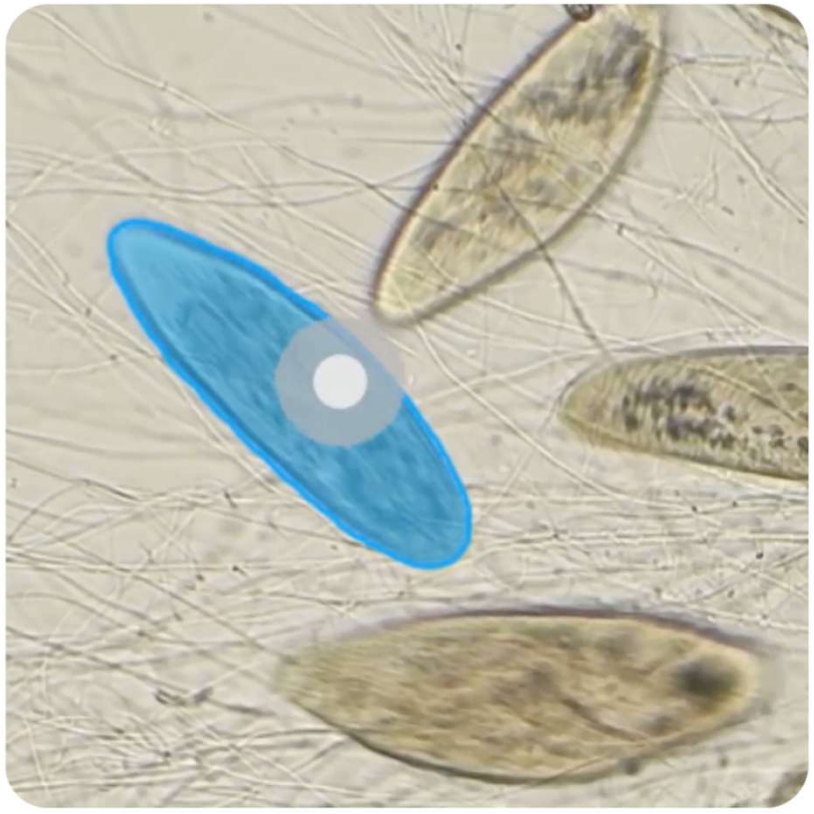
Soluciones basadas en "deep learning"

Resultados @ 2018

Apéndice
Detección de objetos

Soluciones basadas en "deep learning"

Resultados @ 2024



Apéndice
Detección de objetos

Tutoriales y demos

Técnicas tradicionales

- HOG [Histogram of Oriented Gradients]
  https://www.learnopencv.com/histogram-of-oriented-gradients/
  http://mccormickml.com/2013/05/09/hog-person-detector-tutorial/

Técnicas basadas en deep learning

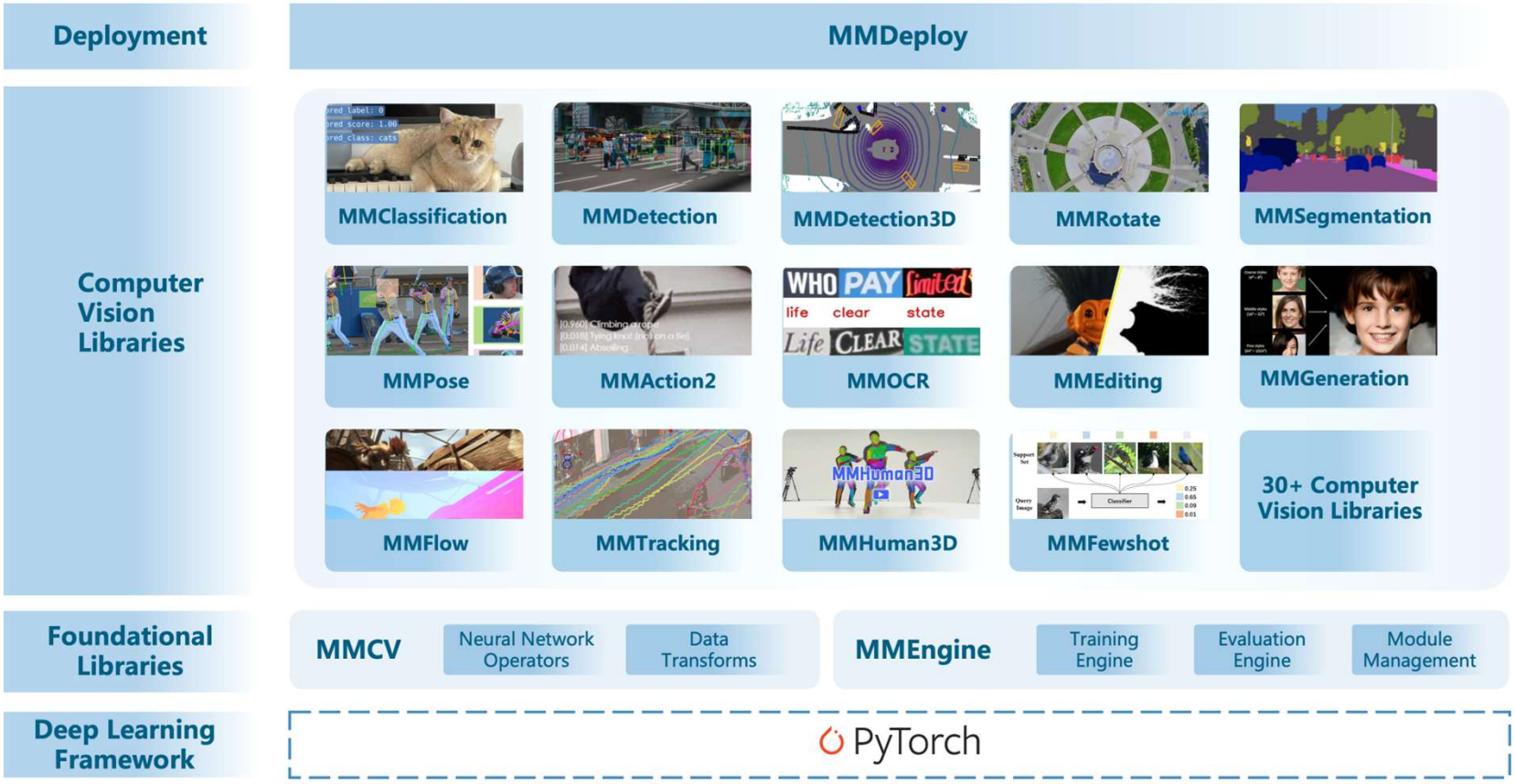
- YOLO Object Detection
  https://pjreddie.com/darknet/yolo/
  https://www.youtube.com/watch?v=4eIBisqx9_g
  https://github.com/llSourcell/YOLO_Object_Detection
- OpenMMLab Detection Toolbox and Benchmark
  https://github.com/open-mmlab/mmdetection

Apéndice
Detección de objetos

Herramientas

https://openmmlab.com/

https://github.com/open-mmlab



Apéndice
Detección de objetos

Referencias

Lilian Weng (OpenAI):

"Object Detection for Dummies"

- Part 1: Gradient Vector, HOG, and SS
- Part 2: CNN, DPM, and OverFeat
- Part 3: R-CNN Family
- Part 4: Fast Detection Models (e.g. YOLO)

https://lilianweng.github.io/, December 2017

Cursos

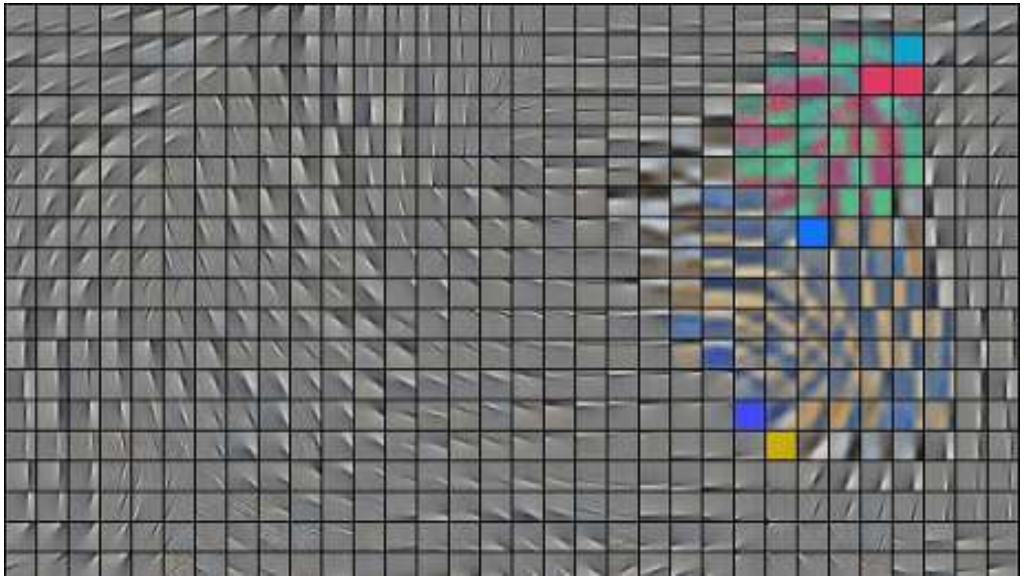
Neural Networks for Machine Learning

by Geoffrey Hinton

(University of Toronto & Google)

https://www.coursera.org/course/neuralnets

coursera



UNIVERSITY OF TORONTO

Cursos

Deep Learning Specialization

by Andrew Ng, 2017

- Neural Networks and Deep Learning
- Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization
- Structuring Machine Learning Projects
- Convolutional Neural Networks
- Sequence Models

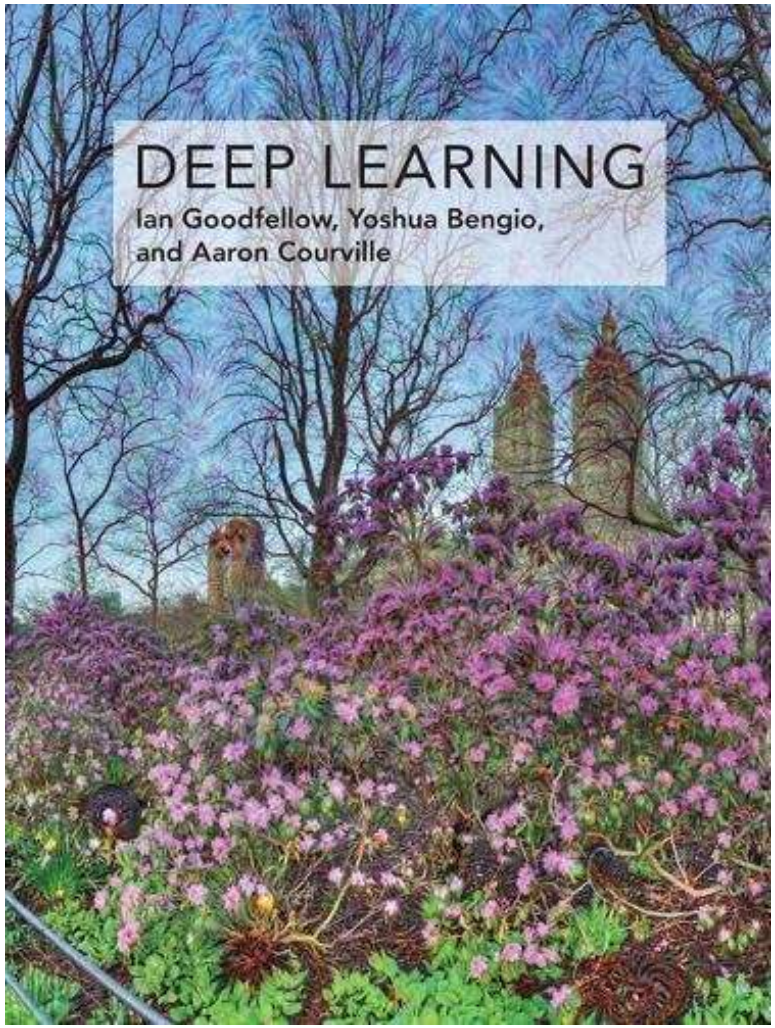
deeplearning.ai

https://www.coursera.org/specializations/deep-learning

Bibliografía

Lecturas recomendadas

Ian Goodfellow,
Yoshua Bengio
& Aaron Courville:
Deep Learning
MIT Press, 2016
ISBN 0262035618



http://www.deeplearningbook.org

Bibliografía

Lecturas recomendadas

Fernando Berzal:
Redes Neuronales & Deep Learning

CAPÍTULO 13
Redes convolutivas