

Sesión 3:

Redes Neuronales Convolucionales para Segmentación de
Imágenes y Detección de Objetos

Elvin Mark Munoz Vega

Table of contents

1. Objetivos
2. Segmentación de Imágenes
3. Detección de objetos

Objetivos

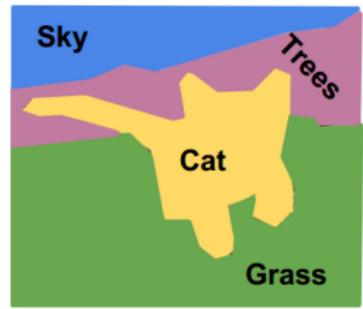
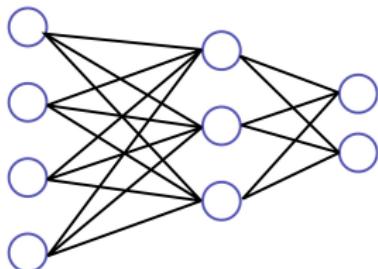
Objetivos

1. Aprender como los Fully Convolutional Networks (FCN) ayudan en la tarea de segmentación de imágenes.
2. Aprender sobre la extracción de Regions of Interest (ROI) y los Region Proposal Networks (RPN) y su papel en la detección de objetos en una imagen.

Segmentación de Imágenes

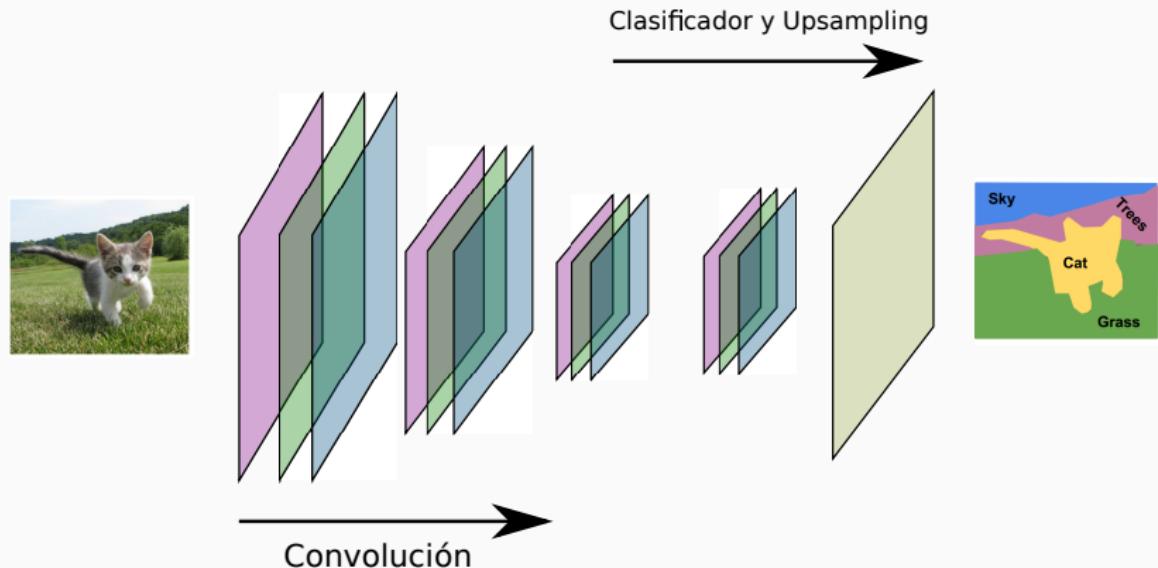
En que consiste la tarea de segmentación de imágenes?

Esta tarea consiste en transformar una imagen en otra imagen de las mismas dimensiones, pero en donde el valor de cada pixel nos indique a que clase el pixel de la imagen original pertenece. En otras palabras queremos dividir la imagen en pedazos o “blobs” de cada clase.



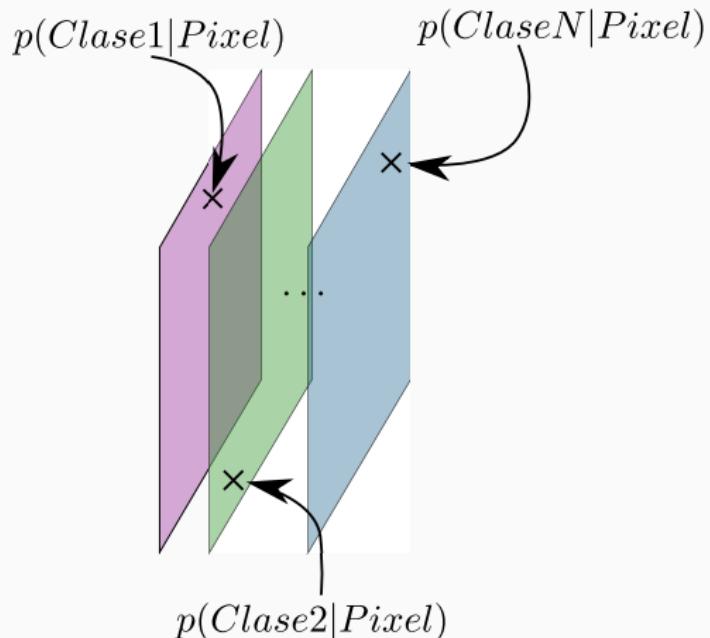
Fully Convolutional Networks (FCN)

Los Fully Convolutional Networks, estan conformados enteramente de capas convolucionales, es decir no hay ningun tipo de fully connected layers en la red, de ahi el nombre de FCN. Los capas convolucionales van extrayendo propiedades locales de la imagen, para luego ser clasificadas y ampliadas para tener una imagen del mismo tamaño que el original.



Fully Convolutional Networks (FCN)

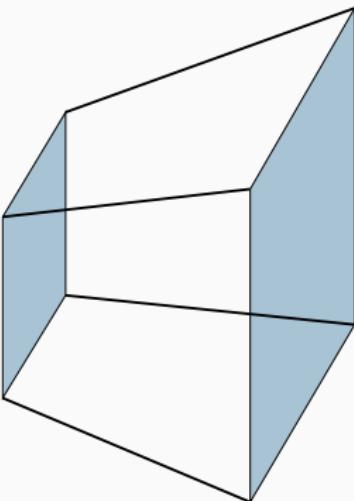
El output de los Fully Convolutional Networks consiste no de una sola imagen, sino de una colección de imágenes (que depende del número de clases), donde los pixeles de cada imagen representan la probabilidad de que el pixel pertenezca a una clase determinada.



Deconvolution Layers

Realiza el proceso inverso de las redes convolucionales. En lugar de extraer propiedades de la imagen, utiliza las propiedades previamente extraídas para reconstruir la imagen.

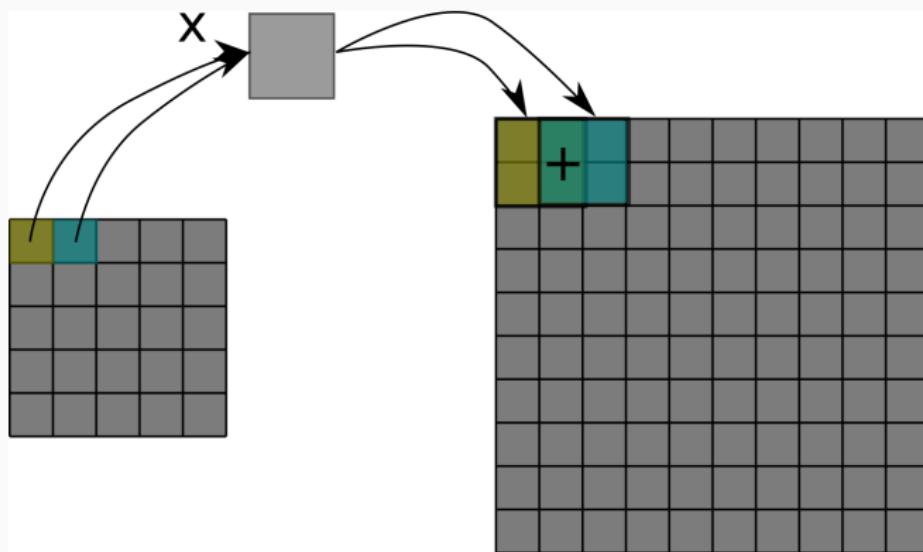
Deconvolución



Reconstrucción basado en las características extraídas

Deconvolution Layers

Esta operación consiste en usar cada elemento del input y multiplicarlo con un kernel, que se va ir desplazando de acuerdo al “striding” que le indiquemos, para ir reconstruyendo el output. Se le puede aplicar un padding inverso (realizar un cropping) a el output.

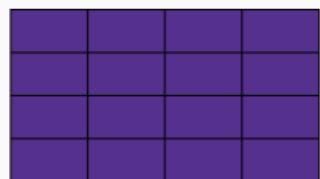


Deconvolution layers: Example

1	2	0
3	1	1
2	0	2


$$^{-1}$$

1	3
0	1



Deconvolution layers: Example

1	2	0
3	1	1
2	0	2



1	3
0	1

1*1	1*3		
1*0	1*1		

Deconvolution layers: Example

1	2	0
3	1	1
2	0	2



1	3
0	1

1	3 + 2*1	2*3	
0	1 + 2*0	2*1	

Deconvolution layers: Example

1	2	0
3	1	1
2	0	2



1	3
0	1

1	5	6+0*1	0*3
0	1	2+0*0	0*1

Deconvolution layers: Example

1	2	0
3	1	1
2	0	2



1	3
0	1

1	5	6	0
0+3*1	1+3*3	2	0
3*0	3*1		

Deconvolution layers: Example

1	2	0
3	1	1
2	0	2



1	3
0	1

1	5	6	0
3	$10+1*1$	$2+1*3$	
0	$3+1*0$	$1*1$	

Deconvolution layers: Example

1	2	0
3	1	1
2	0	2



1	3
0	1

1	5	6	0
3	11	$5+1*1$	$1*3$
0	3	$1+1*0$	$1*1$

Deconvolution layers: Example

1	2	0
3	1	1
2	0	2



1	3
0	1

1	5	6	0
3	11	6	3
0+2*1	3+2*3	1	1
2*0	2*1		

Deconvolution layers: Example

1	2	0
3	1	1
2	0	2



1	3
0	1

1	5	6	0
3	11	6	3
2	9+0*1	1+0*3	1
0	2+0*0	0*1	

Deconvolution layers: Example

1	2	0
3	1	1
2	0	2



1	3
0	1

1	5	6	0
3	11	6	3
2	9	$1+2*1$	$1+2*3$
0	2	$0+2*0$	$2*1$

Deconvolution layers: Example

1	2	0
3	1	1
2	0	2



1	3
0	1

1	5	6	0
3	11	6	3
2	9	3	7
0	2	0	2

Detección de objetos

Que es la detección de objetos?

Esta tarea consiste en detectar ciertos objetos dentro de una imagen. La diferencia con la anterior tarea de segmentación radica en el hecho de que no tenemos que especificar cada pixel que pertenece a cada objeto, pero solo encontrar los “boundary boxes” que los encapsulan.



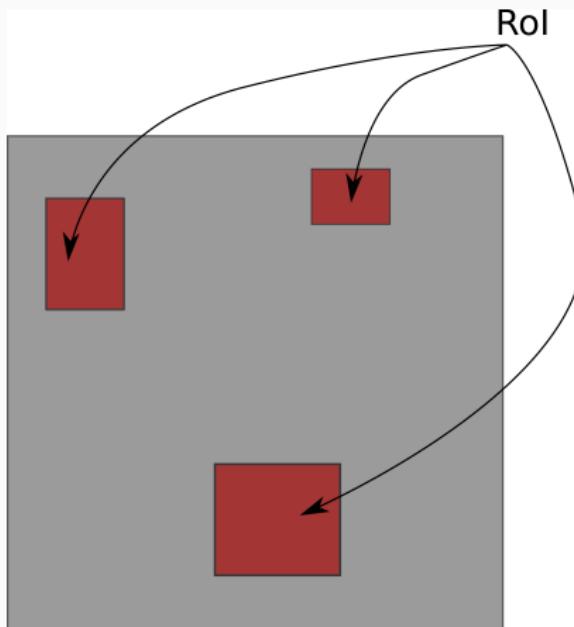
Que es la detección de objetos?

Esta tarea consiste en detectar ciertos objetos dentro de una imagen. La diferencia con la anterior tarea de segmentación radica en el hecho de que no tenemos que especificar cada pixel que pertenece a cada objeto, pero solo encontrar los “boundary boxes” que los encapsulan.



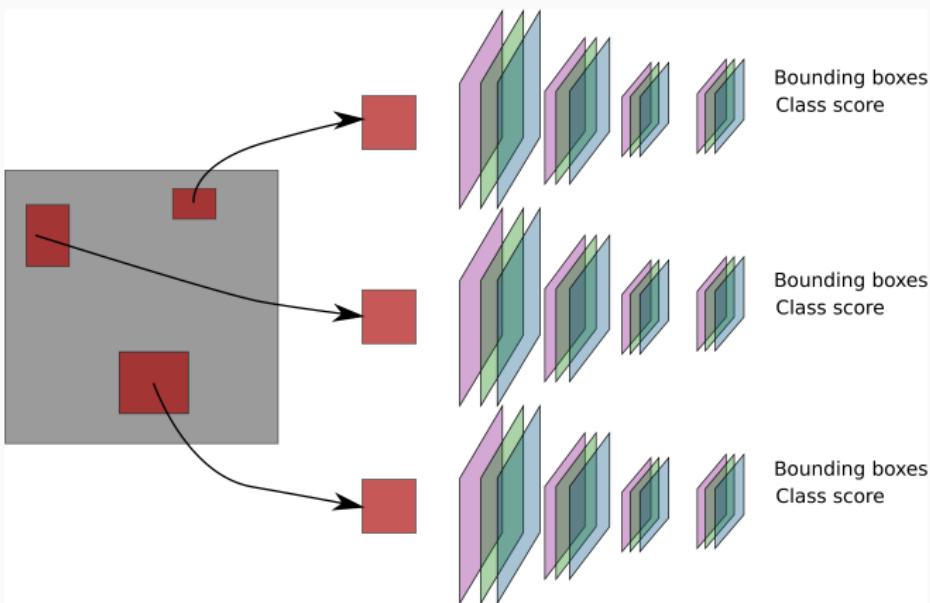
Region of Interest (ROI)

Extraer las regiones donde es más probable encontrar un objeto. Se utilizan algoritmos de procesamiento de imágenes como HoG (Histogram of Gradients).



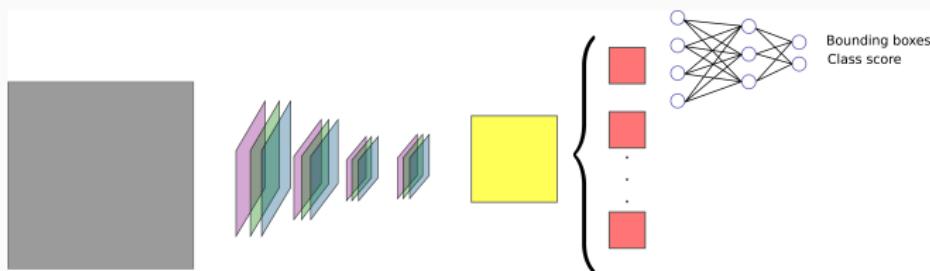
R-CNN

Encontrar a que clase pertenece cada región preseleccionada. Simple clasificación de imágenes.



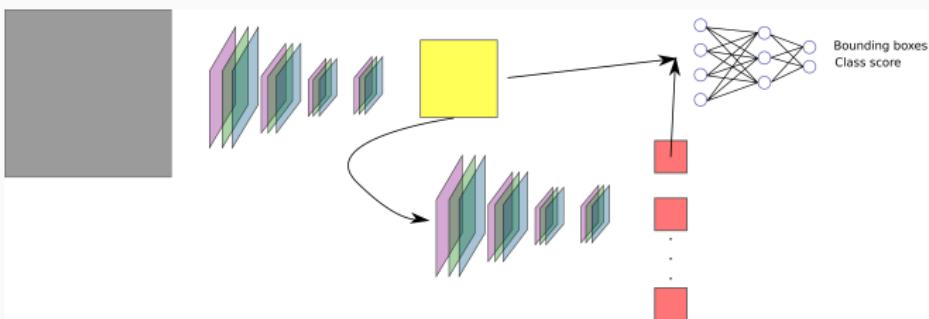
Fast R-CNN

Primero extraer las características o “features” de la imagen para no tener que hacerlo luego para cada region seleccionada. Reducción del costo computacional \Rightarrow más rapido (más cercano a aplicaciones reales).



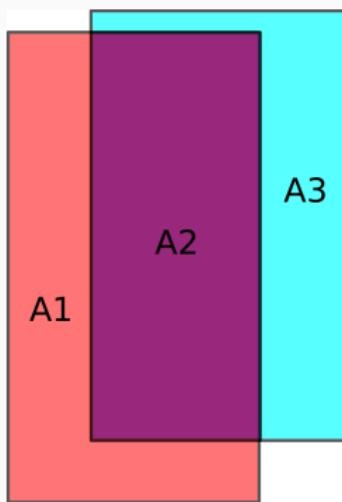
Faster R-CNN

En lugar de usar algoritmos clásicos para seleccionar las regiones de interés, mejor usar otra red convolucional.



Como medir el “loss” de los bounding boxes

Para medir que tan correcto se predijeron los bounding boxes con respecto a los bounding boxes que se dieron de referencia se utiliza la cantidad IoU: Intersection over Union.



$$\text{IoU} = \frac{\text{A2}}{\text{A1} + \text{A2} + \text{A3}}$$