

Deep Learning

Abdelhak Mahmoudi
abdelhak.mahmoudi@um5.ac.ma

ENSIAS- 2021-2022

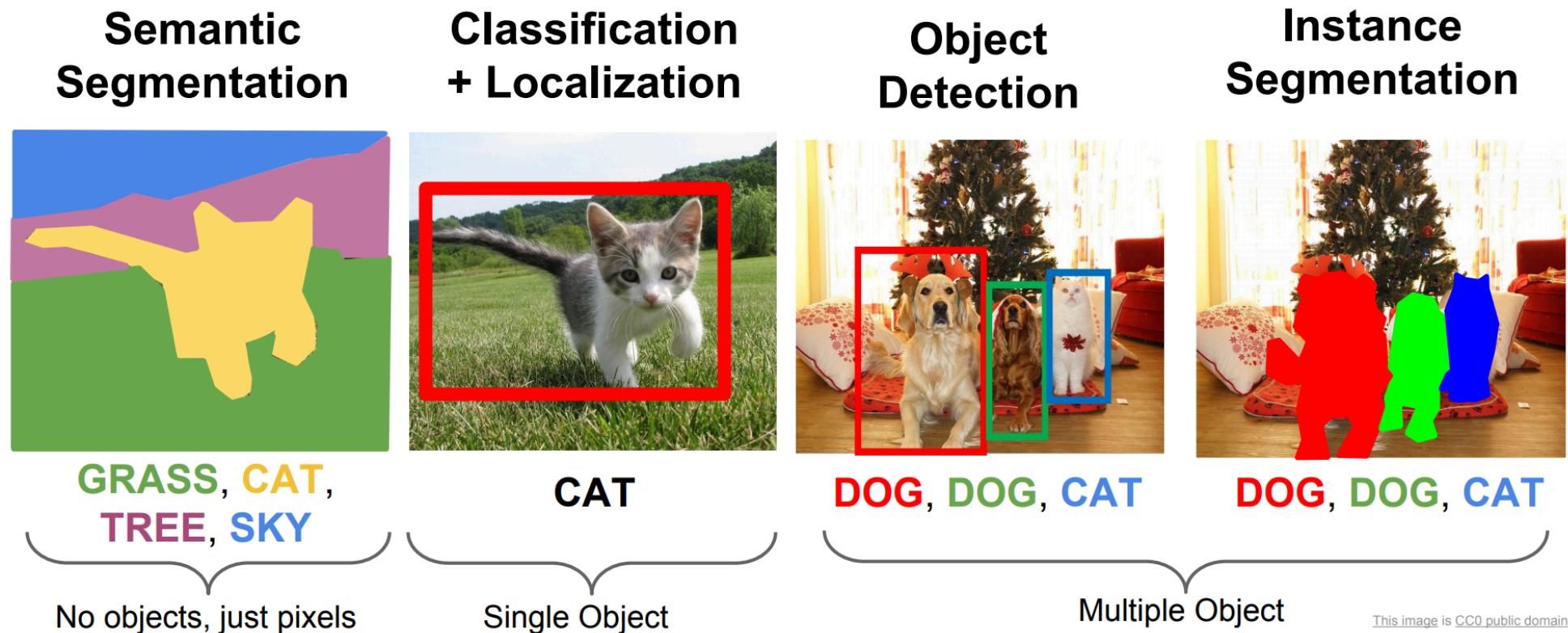
Content

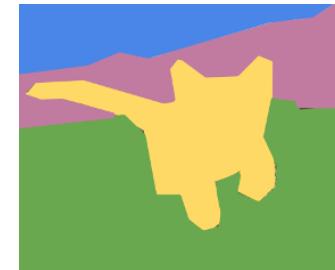
1. Deep Artificial Neural Networks
2. **Convolutional Neural Networks**
3. Sequence Models
4. Generative Models

CNN for Computer Vision

- Semantic Segmentation
- Classification + Localization
- Object Detection
- Instance Segmentation

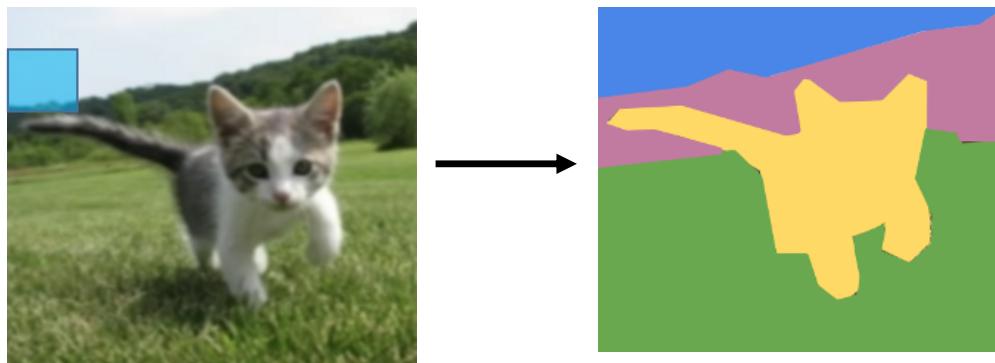
CNN for Computer Vision





Semantic Segmentation

- Sliding window ?
 - Classify center pixel with CNN
 - Very expensive
 - Not sharing features

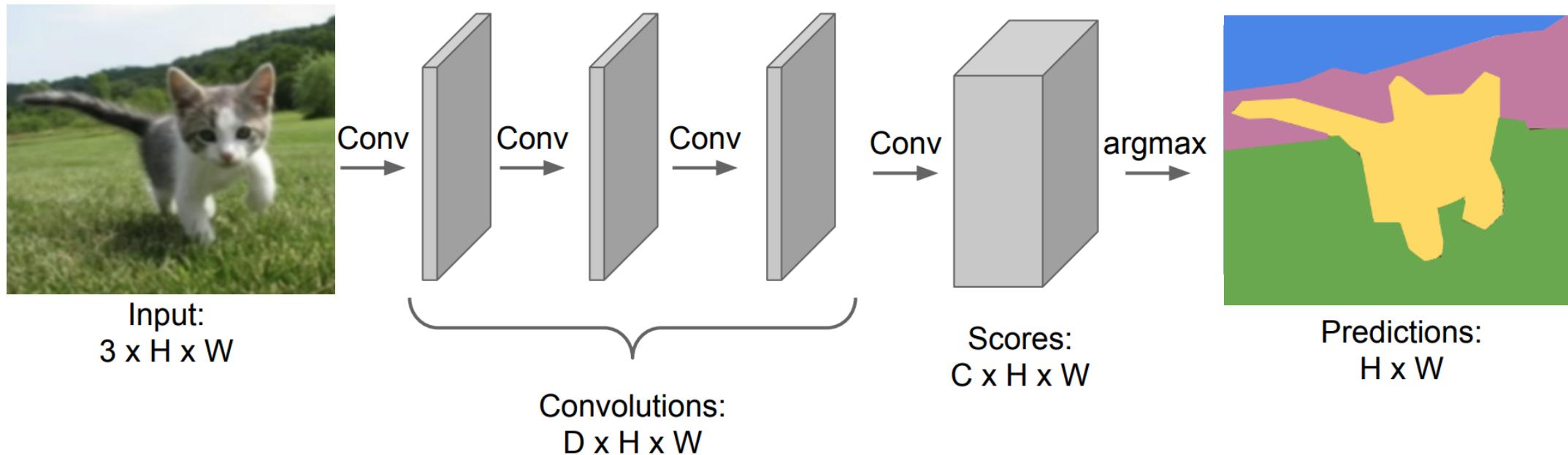




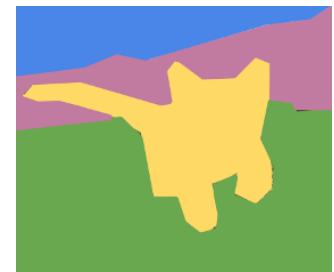
Semantic Segmentation

Fully Conv Nets

Preserve the size of the input
But still expensive



Semantic Segmentation



Fully Conv Nets
Symmetric with bottleneck



Input:
 $3 \times H \times W$

High-res:
 $D_1 \times H/2 \times W/2$

Convolution

(Encoding, downSampling, pooling)

Med-res:
 $D_2 \times H/4 \times W/4$

Low-res:
 $D_3 \times H/4 \times W/4$

Med-res:
 $D_2 \times H/4 \times W/4$

High-res:
 $D_1 \times H/2 \times W/2$

Transpose Convolution

Abdelhak Mahmoudi



Predictions:
 $H \times W$

Semantic Segmentation

2	6	4	5
7	3	3	8
2	1	0	0
4	3	4	3

Max pooling

7	8
4	4

7	0	8	0
0	0	0	0
4	0	4	0
0	0	0	0

Unpooling

0	0	0	0
7	0	0	8
0	0	0	0
4	0	4	0

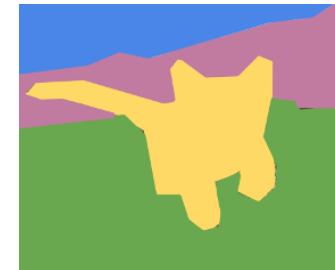
Max
Unpooling

7	7	8	8
7	7	8	8
4	4	4	4
4	4	4	4

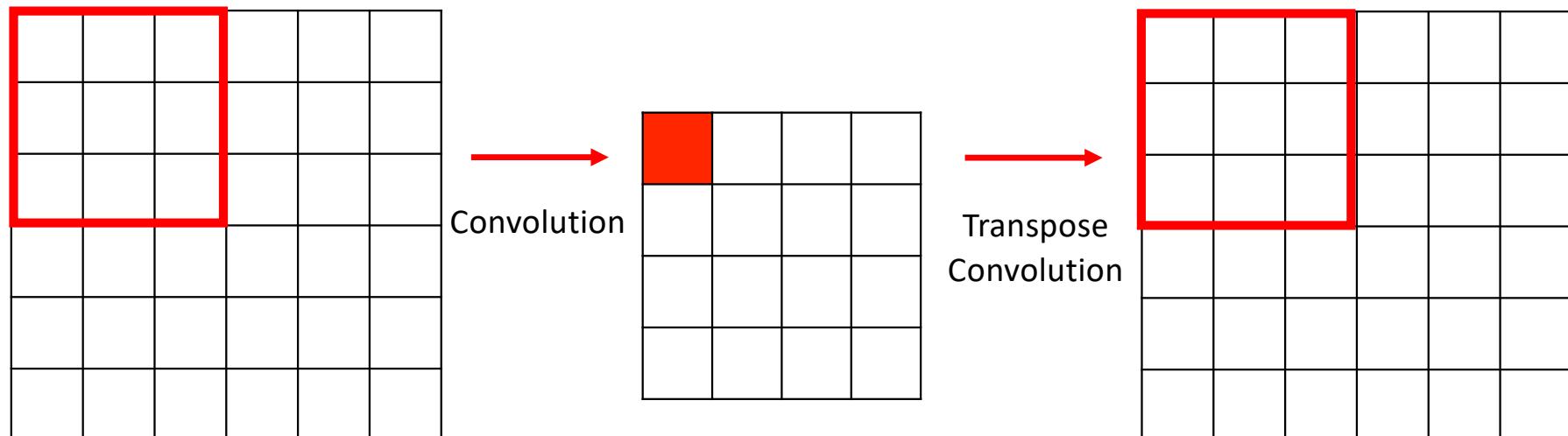
Nearest
Neighbor
Unpooling

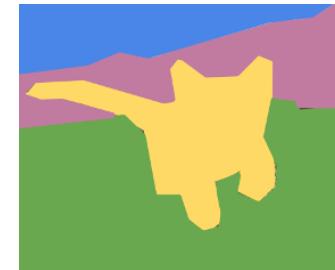
Abdelhak Mahmoudi



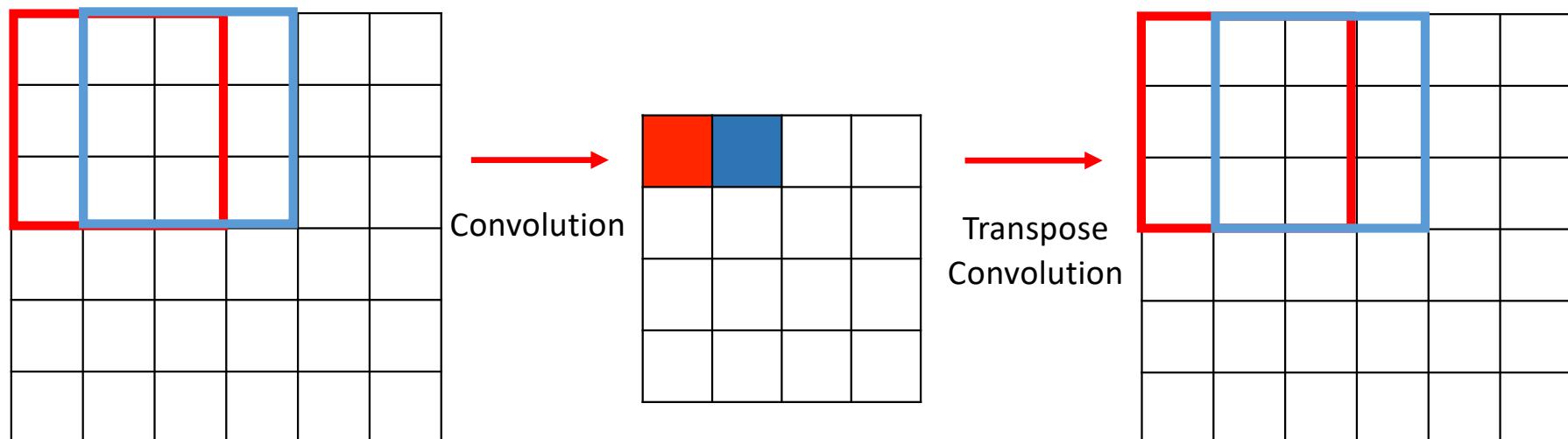


Semantic Segmentation



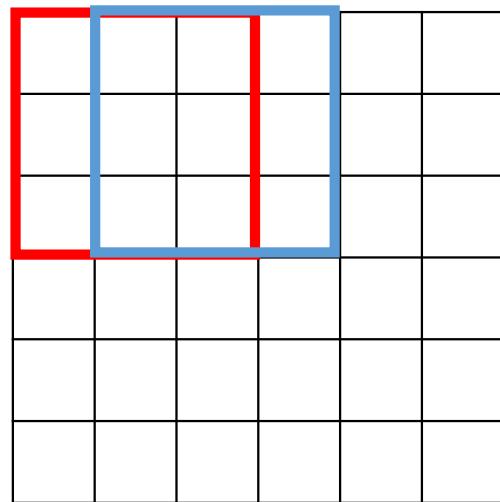


Semantic Segmentation

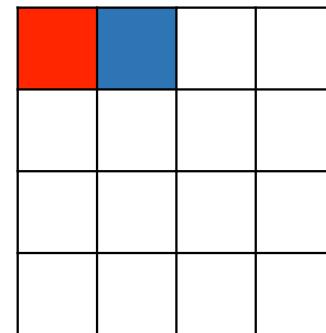




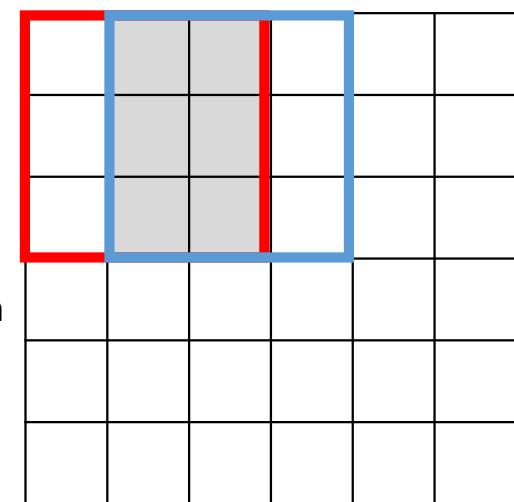
Semantic Segmentation



Convolution



Transpose
Convolution



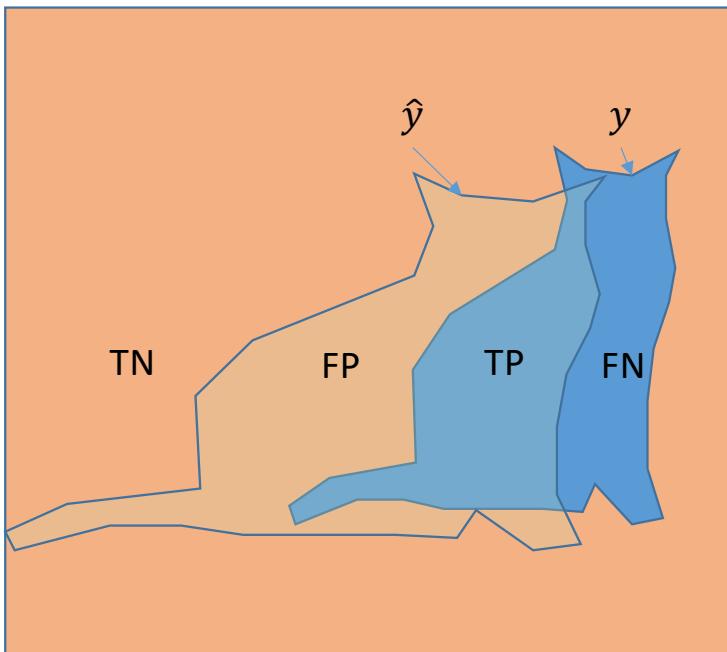


Semantic Segmentation

Example of Transpose Convolution

Input	Kernel	$=$				Output				
$\begin{matrix} 0 & 1 \\ 2 & 3 \end{matrix}$	$\begin{matrix} 0 & 1 \\ 2 & 3 \end{matrix}$	$\begin{matrix} 0 & 0 & \\ 0 & 0 & \\ & & \end{matrix}$	$+$	$\begin{matrix} & 0 & 1 \\ & 2 & 3 \end{matrix}$	$+$	$\begin{matrix} & & \\ 0 & 2 & \\ 4 & 6 & \end{matrix}$	$+$	$\begin{matrix} & & \\ 0 & 3 & \\ 6 & 9 & \end{matrix}$	$=$	$\begin{matrix} 0 & 0 & 1 \\ 0 & 4 & 6 \\ 4 & 12 & 9 \end{matrix}$

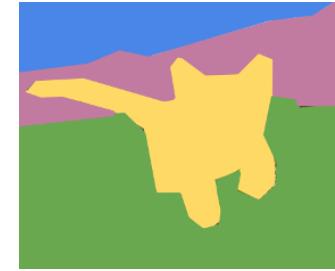
Semantic Segmentation Metrics



- $y = TP + FN$ = **Ground truth**
- $\hat{y} = TP + FP$ = **Prediction**
- Precision = $\frac{G \cap P}{G} = \frac{TP}{TP+FP}$
- Recall = $\frac{G \cap P}{P} = \frac{TP}{TP+FN}$
- IoU = Jaccard = $\frac{G \cap P}{G \cup P} = \frac{\sum \hat{y}y}{\sum \hat{y} + y - \hat{y}y}$
 $= \frac{TP}{TP + FN + FP}$
- Dice = F1 = $\frac{2G \cap P}{G \cup P + G \cap P} = \frac{2PR}{P+R}$
 $= \frac{2 \sum \hat{y}y}{\sum \hat{y} + y} = \frac{2TP}{2TP + FN + FP}$

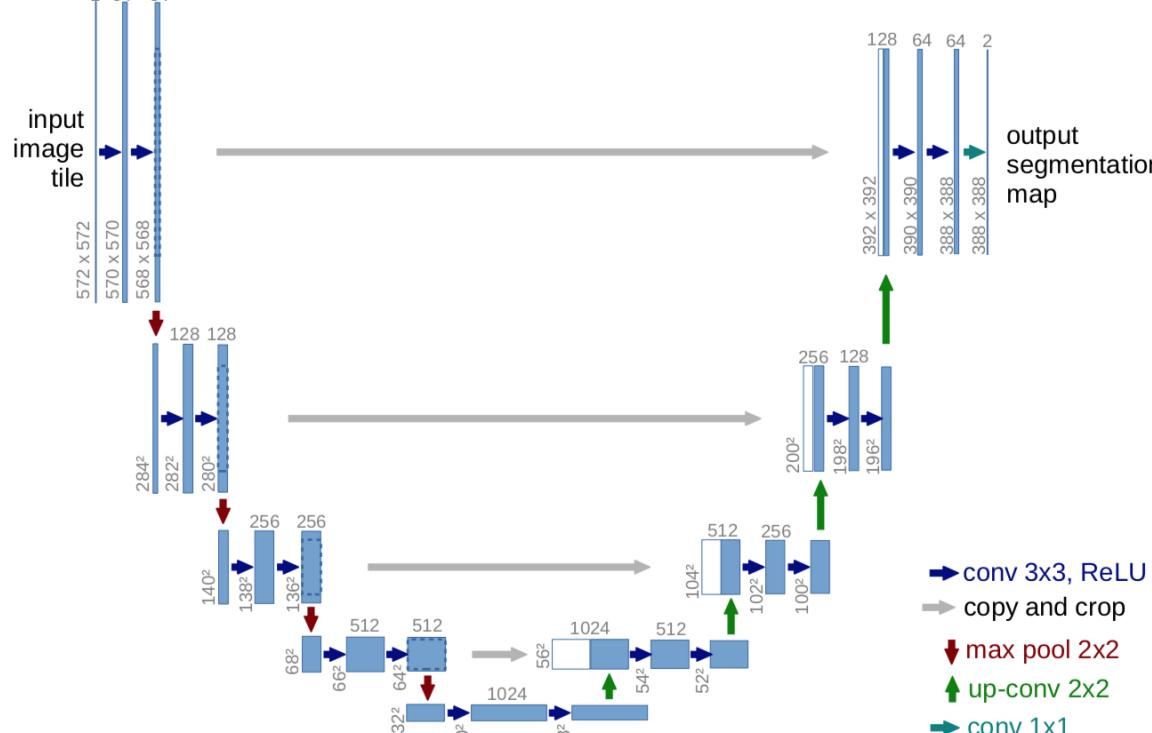
Non differentiable

differentiable



Semantic Segmentation

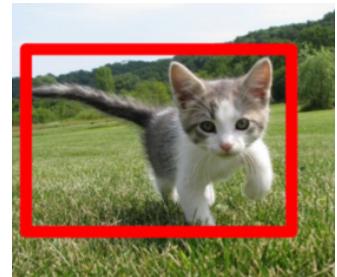
U-net



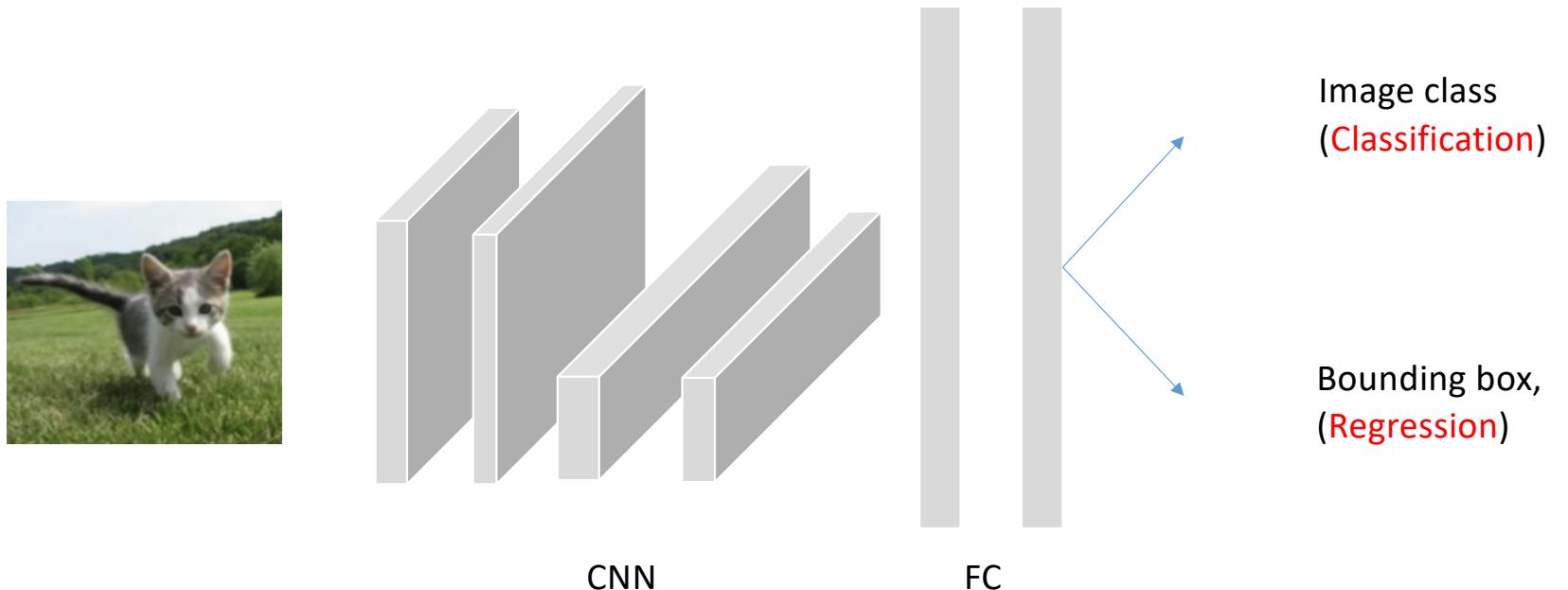
Abdelhak Mahmoudi

14

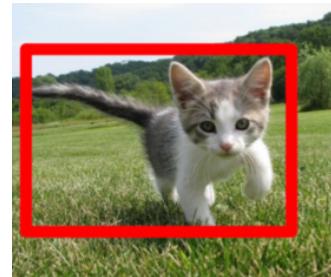
Classification + Localization



- Classification + Rgression



Classification + Localization



Object Recognition



Face Recognition



Pose Recognition



Find a bounding box

$$b_x, b_y, b_h, b_w$$

Find landmarks

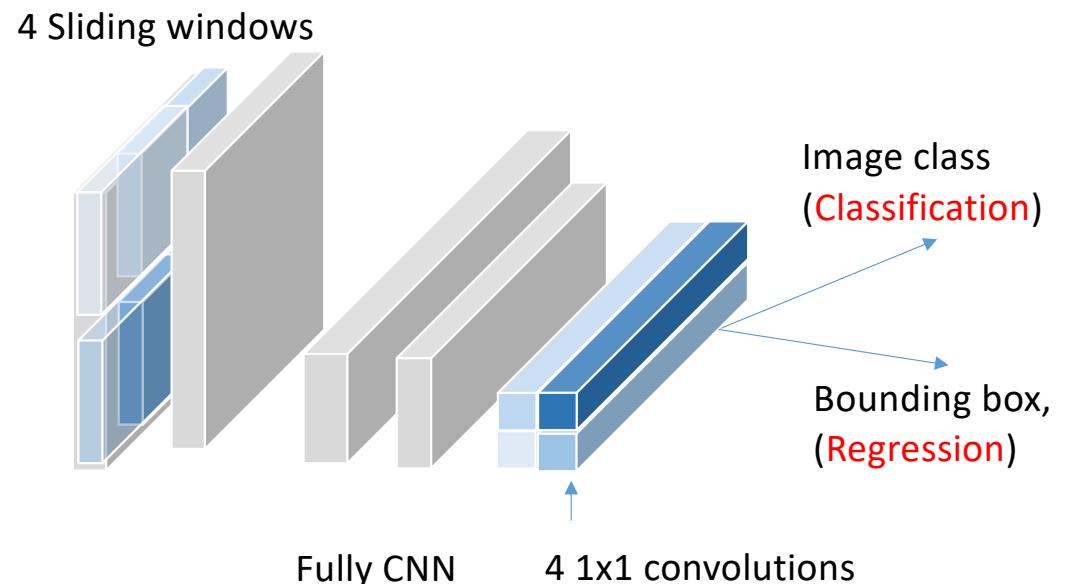
$$l_{1x}, l_{1y},; l_{2x}, l_{2y}; l_{4x}, l_{4y} \dots; l_{64x}, l_{64y}$$

Find bones

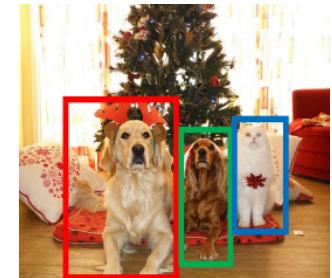


Object Localization

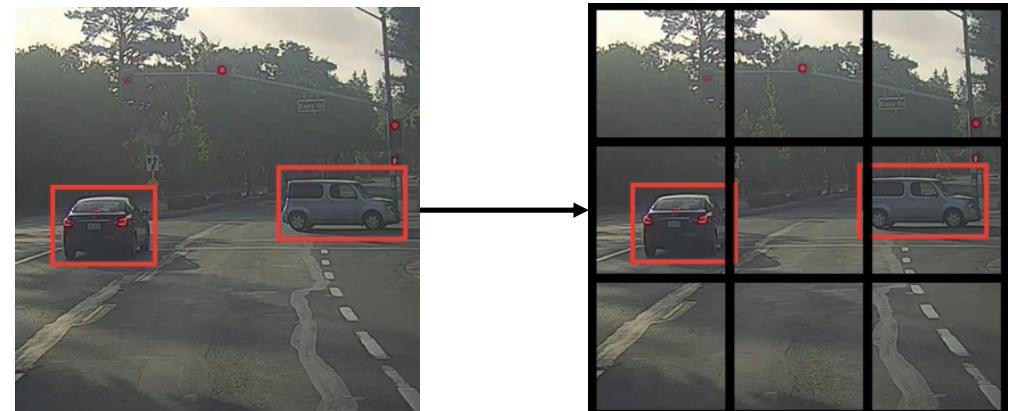
- Multiple Objects
- Different Objects
- **Problem**
 - Need to apply CNN to huge number of sliding windows to all locations and with different scales. Very computationally expensive!
- **1st Solution**
 - Fully ConvNets



Object Localization

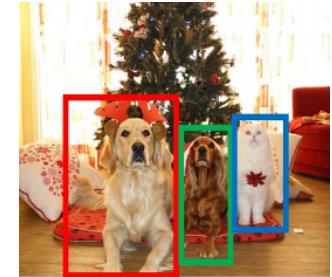


- Multiple Objects
- Different Objects
- **Problem**
 - Non accurate bounding boxes
- **2nd Solution**
 - YOLO (You Only Look Once)



Assign to the appropriate cell of the grid

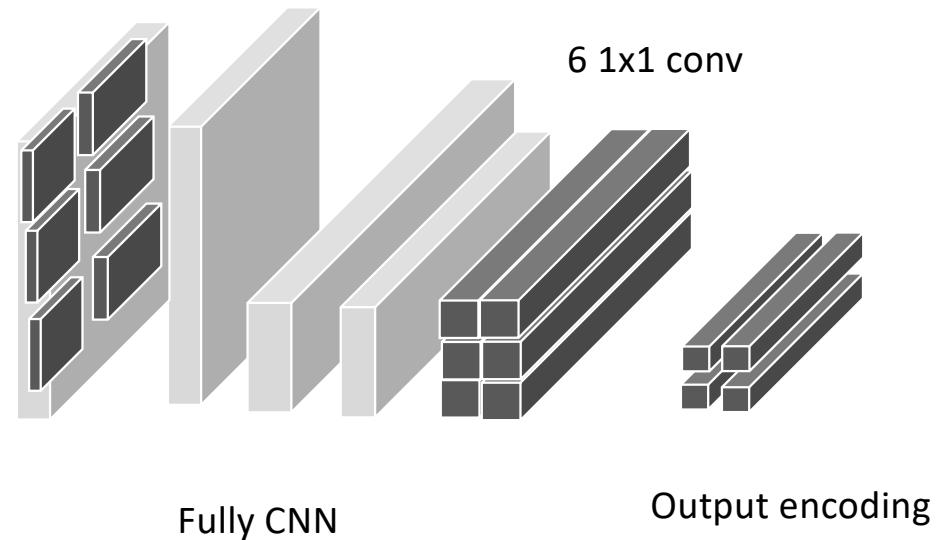
Redmon et al, “You Only Look Once:
Unified, Real-Time Object Detection”, CVPR 2016



Object Localization

- Multiple Objects
- Different Objects
- **Problem**
 - Non accurate bounding boxes
- **2nd Solution**
 - YOLO (You Only Look Once)
 - Last YOLO-V4 (**April 2020**)

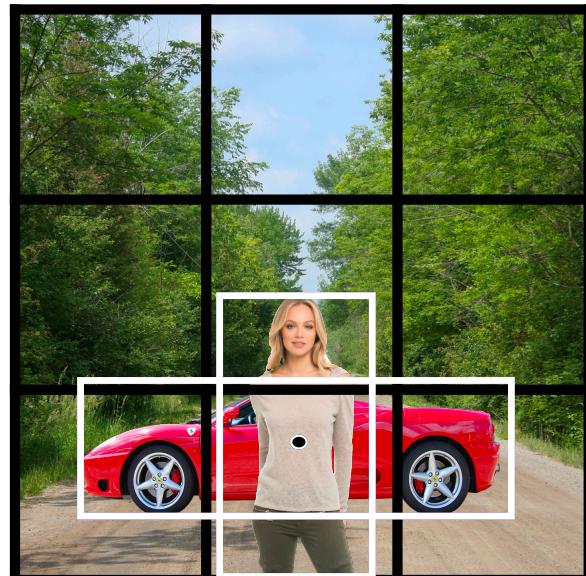
Grid of 6 cells



Object Localization

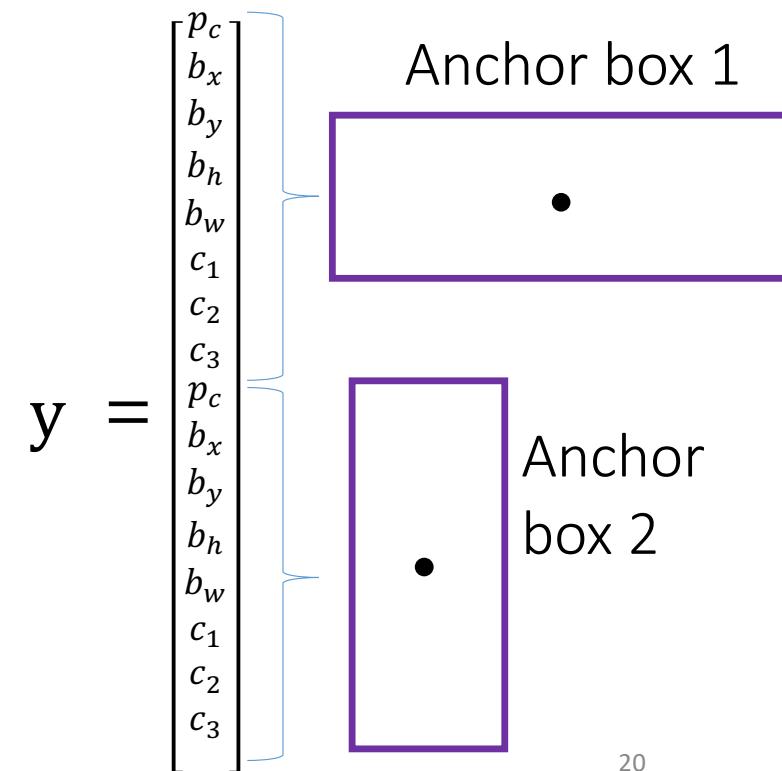


Pedestrian	Car	motorcycle
$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$



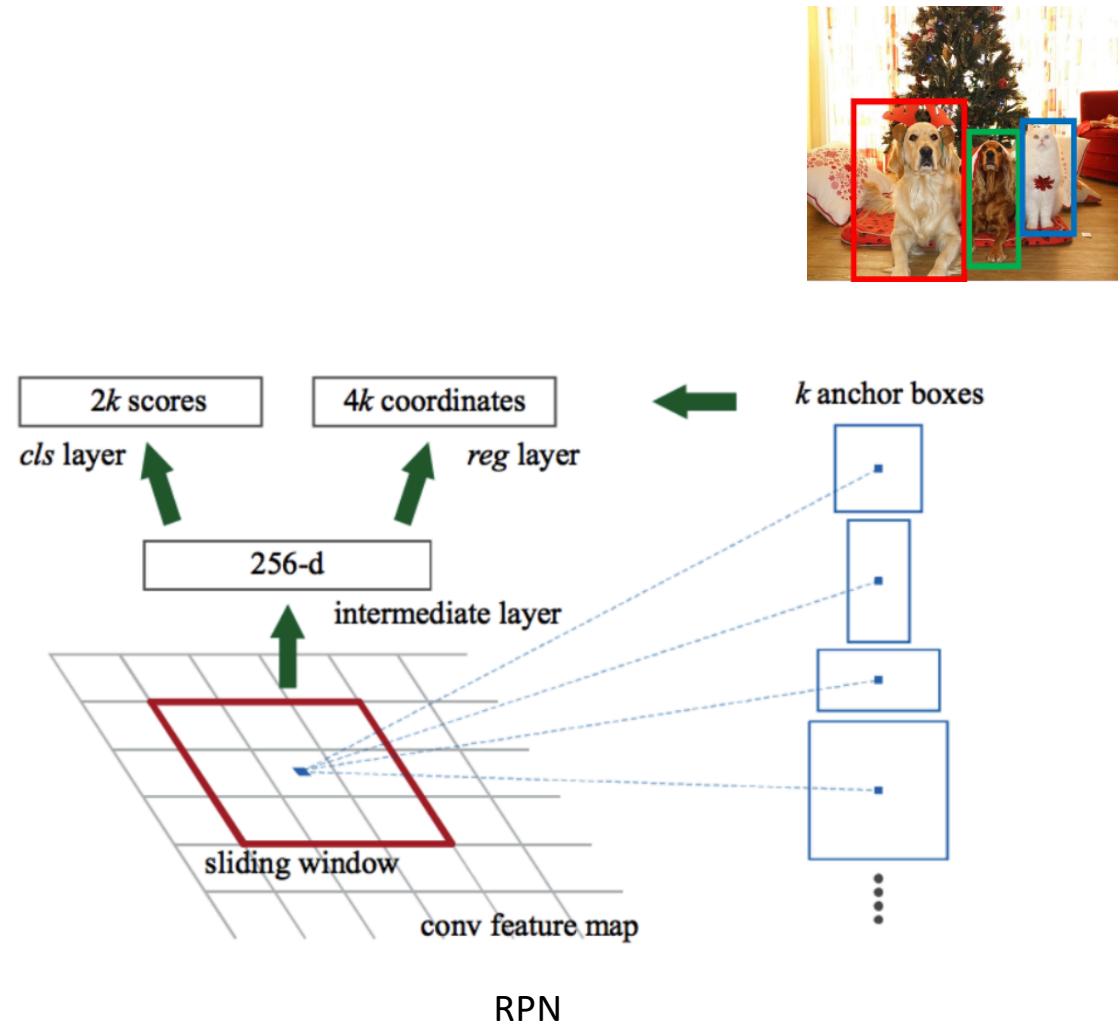
$$\mathbf{y} = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \\ p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

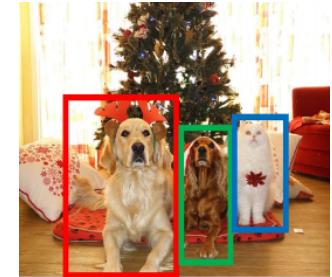
Abdelhak Mahmoudi



Object Localization

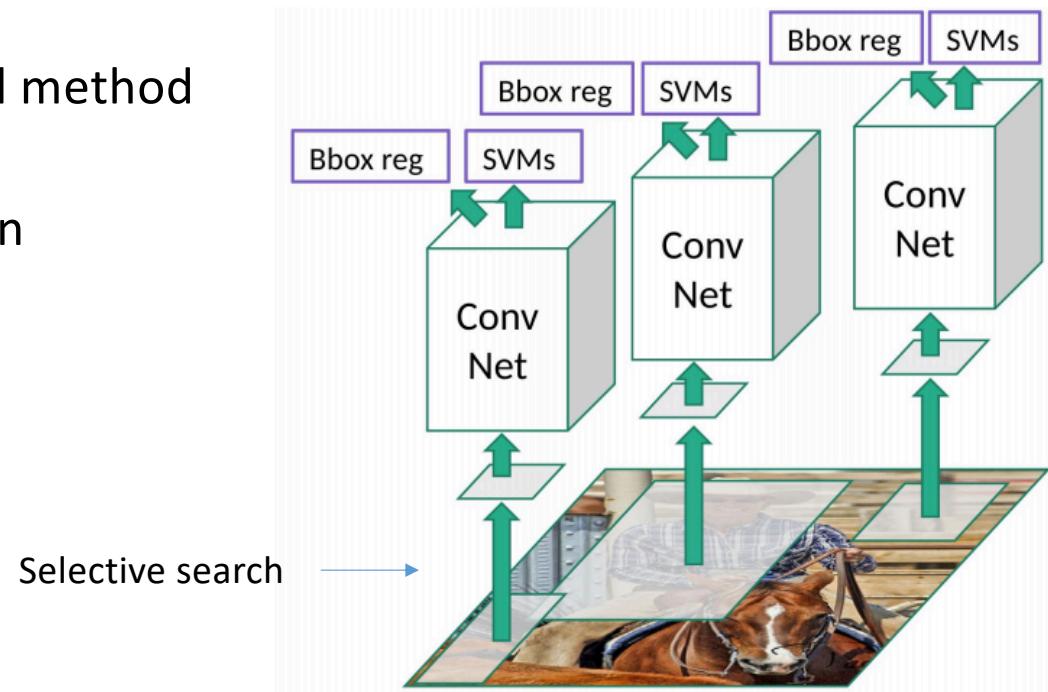
- Multiple Objects
- Different Objects
- **Problem**
 - Need to apply CNN to huge number of sliding windows to all locations and with different scales. Very computationally expensive!
- **3rd Solution**
 - Use Region Proposal Network (RPN)

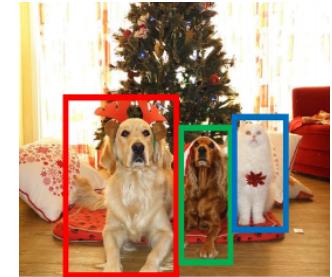




Object Localization

- **RCNN**
 - 1) Selective Search Proposal method
 - 2) CNN
 - 3) Classification + Regression

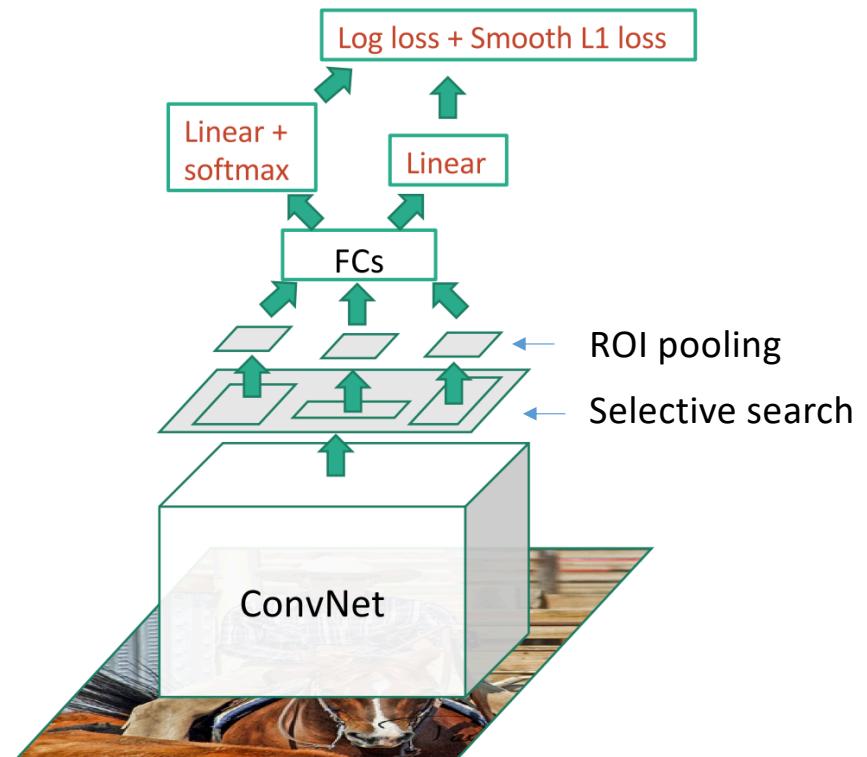




Object Localization

- **Fast-R-CNN**

- 1) CNN
- 2) Selective Search Proposal method
- 3) ROI pooling
- 4) Classification + Regression



Grshick, “Fast R-CNN”, ICCV 2015.

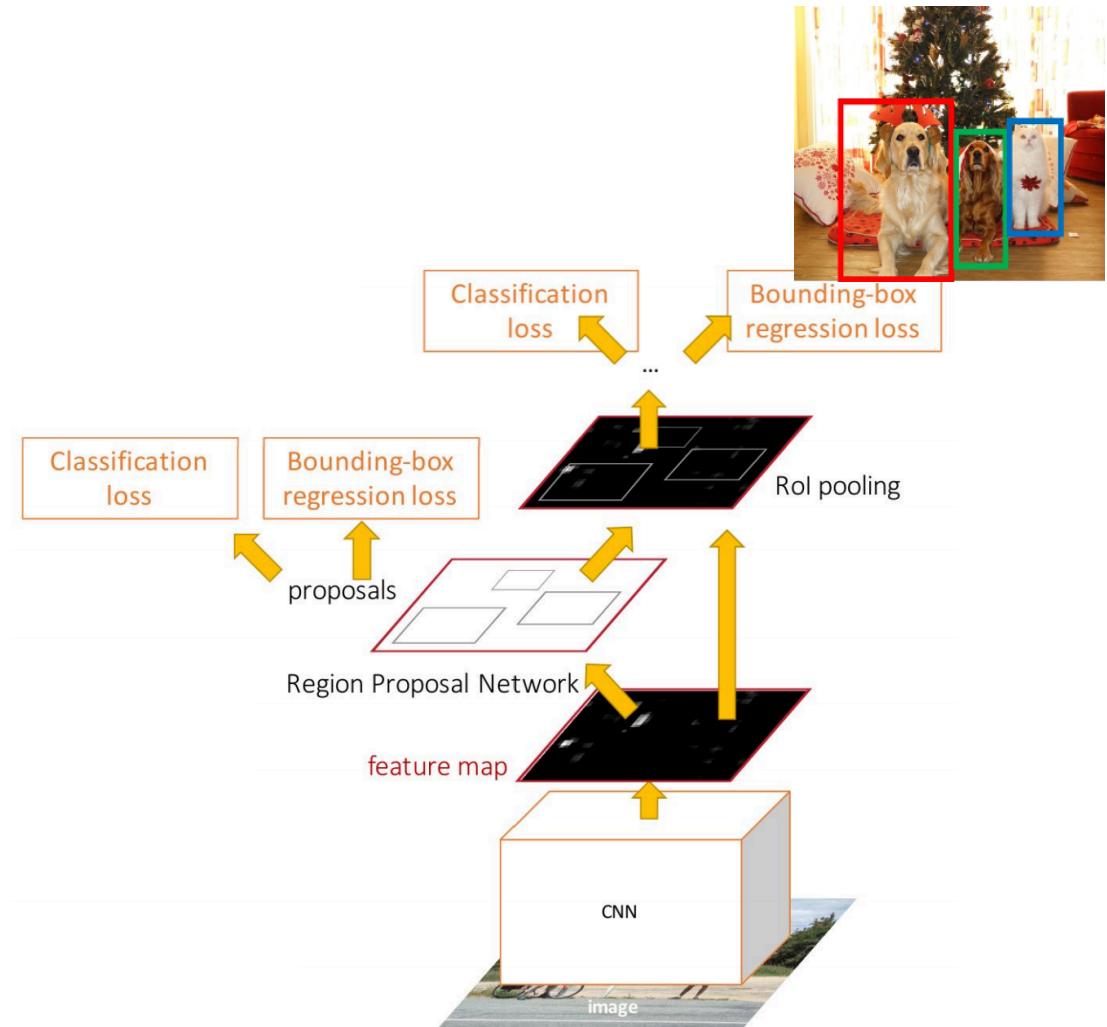
Abdelhak Mahmoudi

23

Object Localization

- **Faster-R-CNN**

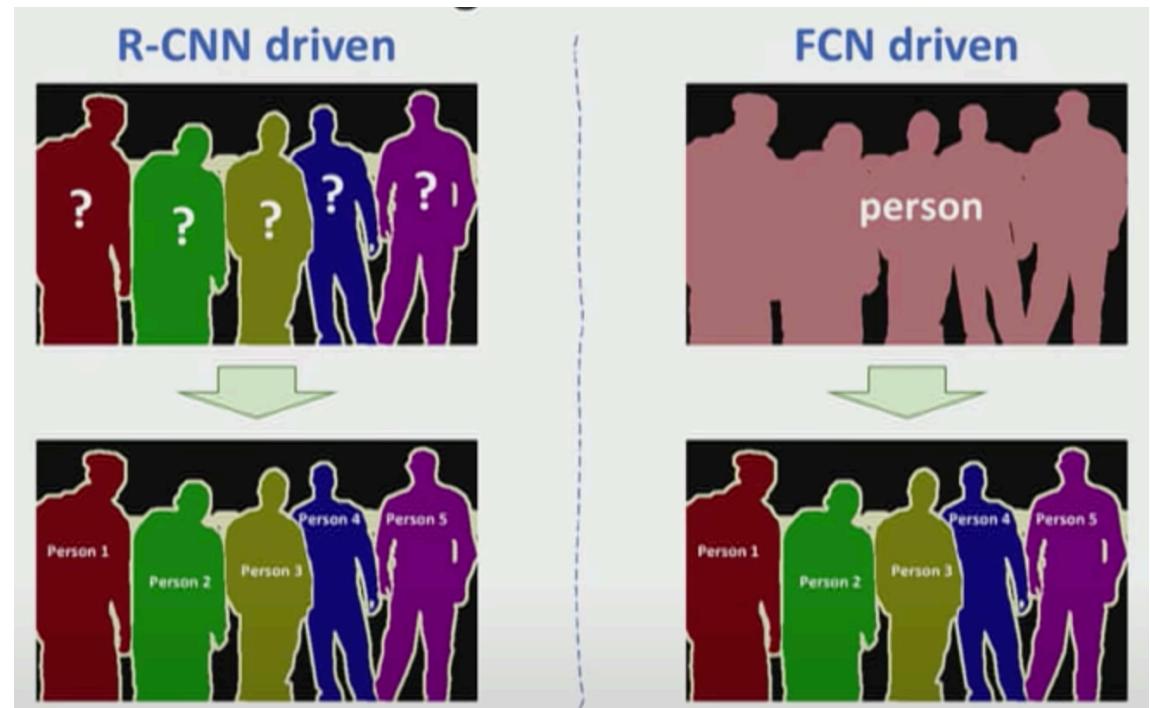
- 1) CNN
- 2) Region Proposal Network (RPN)
- 3) ROI pooling
- 4) Classification + Regression



Instance Segmentation



- Mask-RCNN
 - Combine Faster RCNN with FCNN



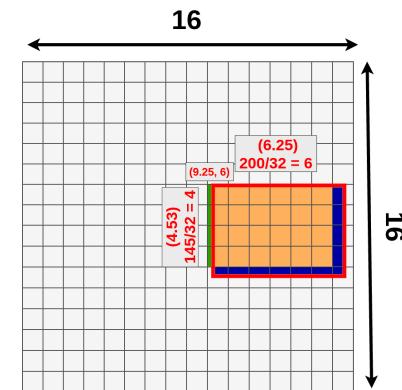
He et al, “Mask R-CNN”, arXiv 2017

Figures copyright Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, 2017.

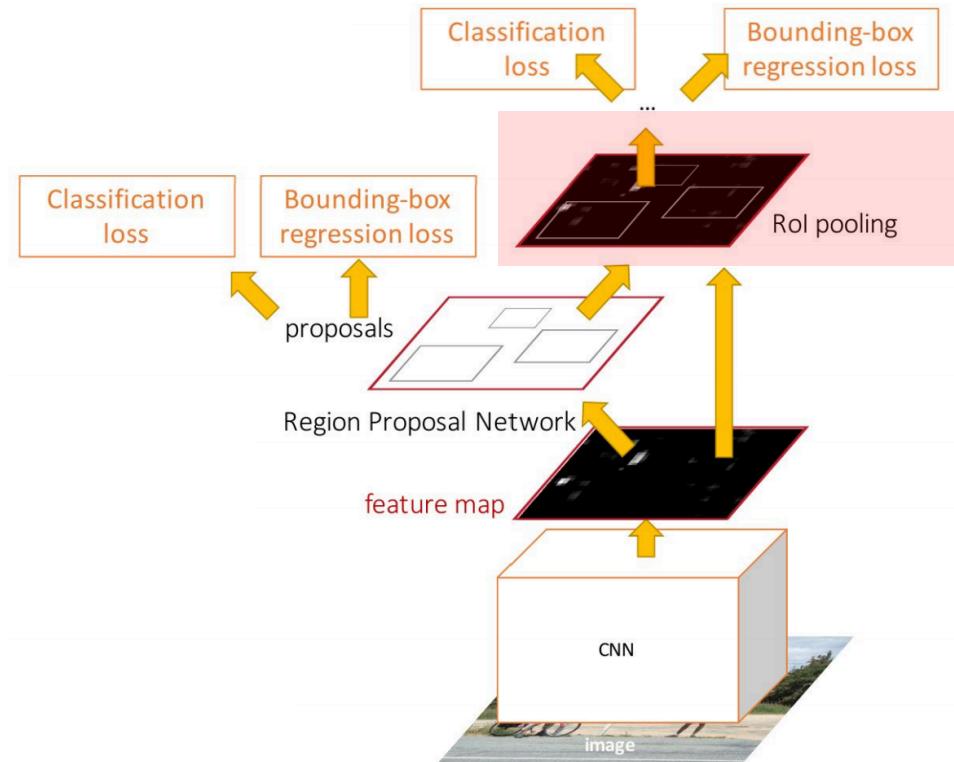
Instance Segmentation



- Mask-RCNN
 - Combine Faster RCNN with FCNN
 - Replace ROI Pooling by ROI alignment or ROI Warping



<https://towardsdatascience.com/@kemalpiro>



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

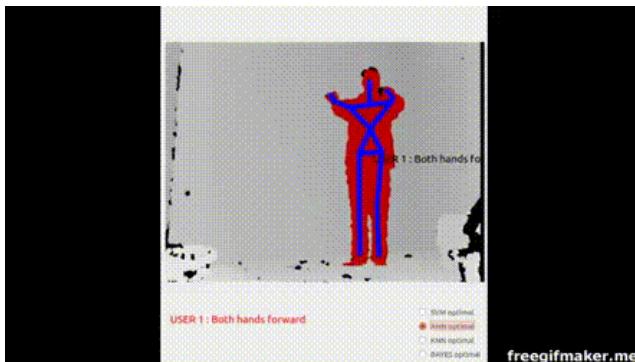
Figures copyright Kaiming He, Georgia Gkioxari,
Piotr Dollár, and Ross Girshick, 2017.

Abdelhak Mahmoudi

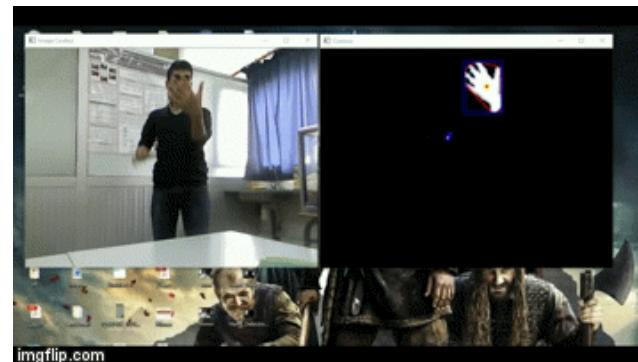
26

CNN for Computer Vision at LIMIARF Lab

Pose recognition



Hand Gestures recognition



Hand Signs recognition



<https://www.youtube.com/c/AbdelhakMahmoudi>