



UNIVERSIDAD DE CHILE

# Deep Learning

Deeper, Better, \_\_\_\_\_, Stronger than Machine Learning

---

Valentin Barriere

Universidad de Chile – DCC

CC6204, Primavera 2024

# Object Detection

# Introduction

---

# Outline : Introduction

## **Introduction**

Datasets

Models

Object Detection

Segmentation

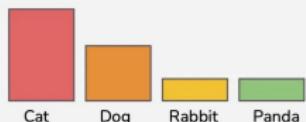
Others and SOTA

# Object Detection

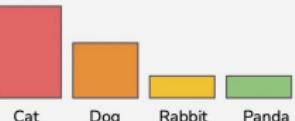
## Object Detection and Localization



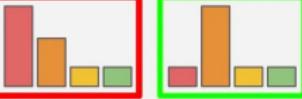
Classification  
Algorithm



Detection &  
Localization Algorithm



Multi-class Detection &  
Localization Algorithm



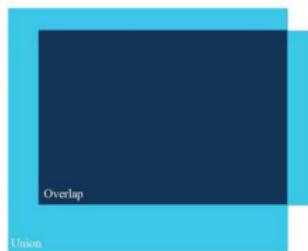
# How to evaluate?

How to evaluate the prediction of an bounding box?



- █ Ground truth
- █ Prediction

$$IoU = \frac{\text{area of overlap}}{\text{area of union}}$$

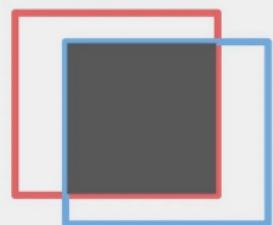


$IoU \geq 0.5$

# Intersection over Union

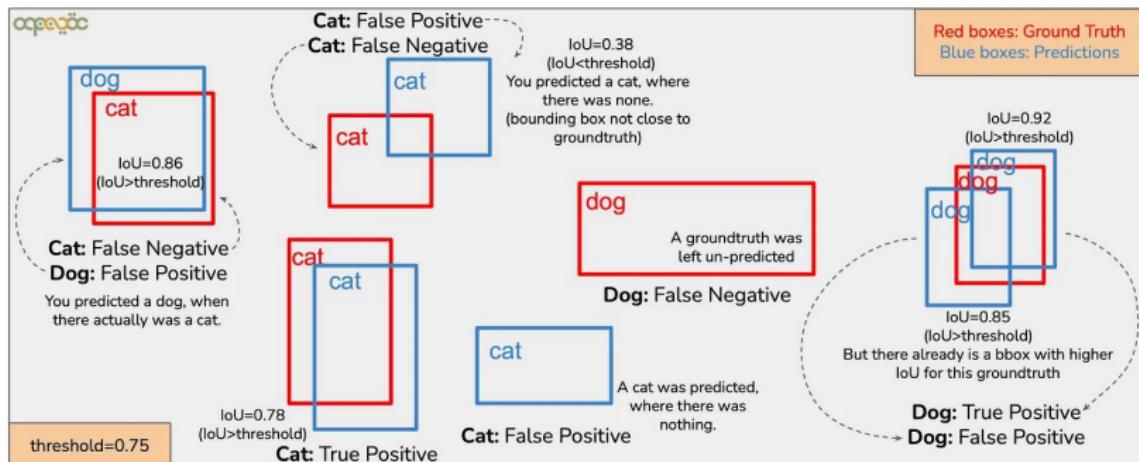
## Intersection over Union (IoU)

Ratio between the area of intersection and the area of union of 2 bounding boxes

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


# Intersection over Union

## Object Detection and Localization - IoU, True Positive, False Positive, False Negative

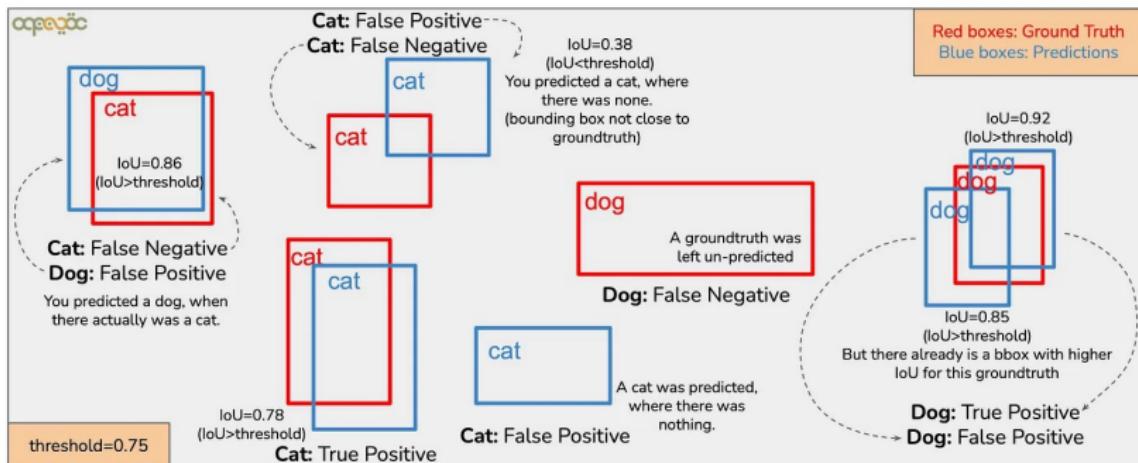


@\_aqueelanwar  
in aqeelanwarmingalik

Threshold	Class	# GroundTruth	# predictions	TP	FP	FN	Precision	Recall
0.75	Cat	3	3	1	2	2	1/3	1/3
	Dog	2	3	1	2	1	1/3	1/2
0.35	Cat	3	3	2	1	1	2/3	2/3
	Dog	2	3	1	2	1	1/3	1/2

# Intersection over Union

## Object Detection and Localization - IoU, True Positive, False Positive, False Negative



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

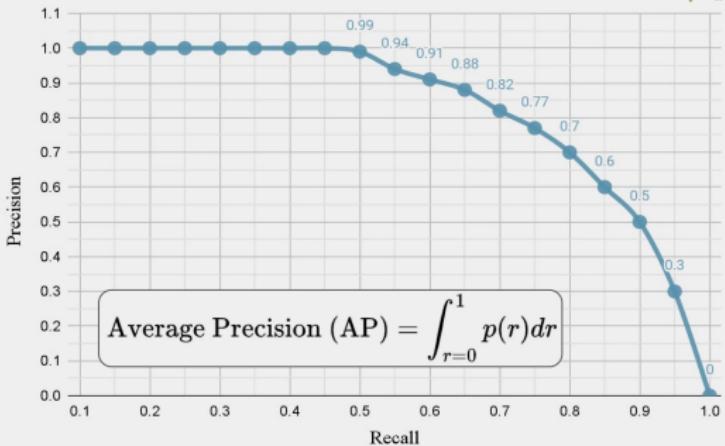
Threshold	Class	# GroundTruth	# predictions	TP	FP	FN	Precision	Recall
0.75	Cat	3	3	1	2	2	1/3	1/3
	Dog	2	3	1	2	1	1/3	1/2
0.35	Cat	3	3	2	1	1	2/3	2/3
	Dog	2	3	1	2	1	1/3	1/2

# Average Precision

## Precision Recall Curve (PR Curve)

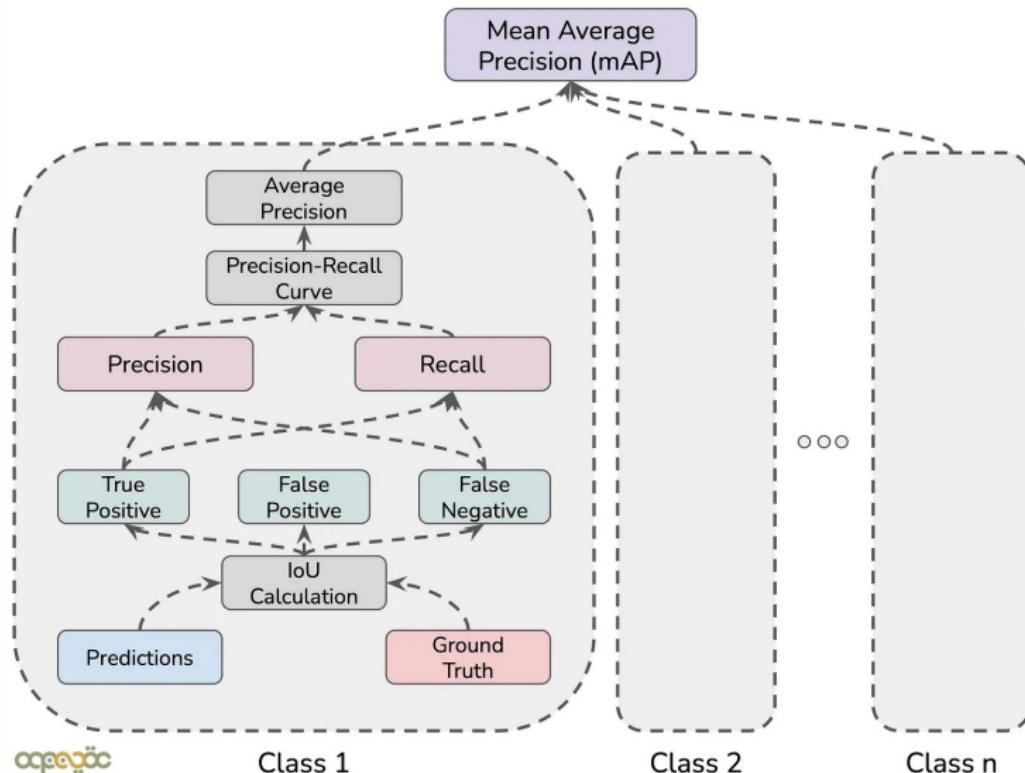
Conf. Thresh.	Recall	Precision	Effect
0.95	0.1	1	More FN
0.9	0.15	1	
0.85	0.2	1	
0.8	0.25	1	
0.75	0.3	1	
0.7	0.35	1	
0.65	0.4	1	
0.6	0.45	1	
0.55	0.5	0.99	
0.5	0.55	0.94	
0.45	0.6	0.91	
0.4	0.65	0.88	
0.35	0.7	0.82	
0.3	0.75	0.77	
0.25	0.8	0.7	
0.2	0.85	0.6	
0.15	0.9	0.5	
0.1	0.95	0.3	
0.05	1	0	More FP

The **smaller the probability confidence threshold**, the higher the number of detections made by the model, and the lower the chances that the ground-truth labels were missed and hence **higher the recall** (Generally, but not always). On the other hand, the **higher the confidence threshold**, the more confident the model is in what it predicts and hence **higher the precision** (Generally, but not always).



# mean Average Precision

## How to calculate mean average precision (mAP)



# Concretely Calculating AP

## Calculating Average Precision from PR Curve

### Approach 1: Sample-and-hold

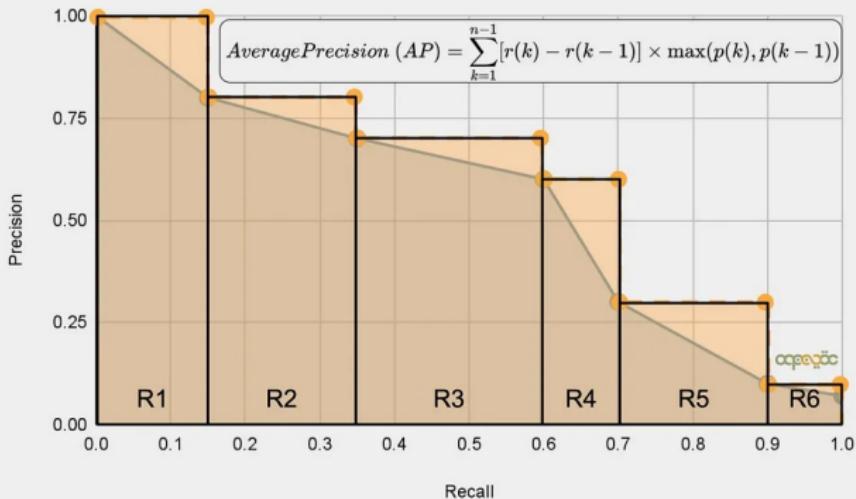
- For each precision-recall pair ( $j=0, \dots, n-1$ ), the area under the PR curve can be found by approximating the curve using rectangles.
- The width of such rectangles can be found by taking the difference of two consecutive recall values ( $r(k), r(k-1)$ ), and the height can be found by taking the maximum value of the precision for the selected recall values i.e.

$$w = r(k) - r(k-1)$$
$$h = \max(p(k), p(k-1))$$

- AP can be calculated by the sum of the areas of these rectangles.

@\_aqeelanwar

aqeelanwormalik



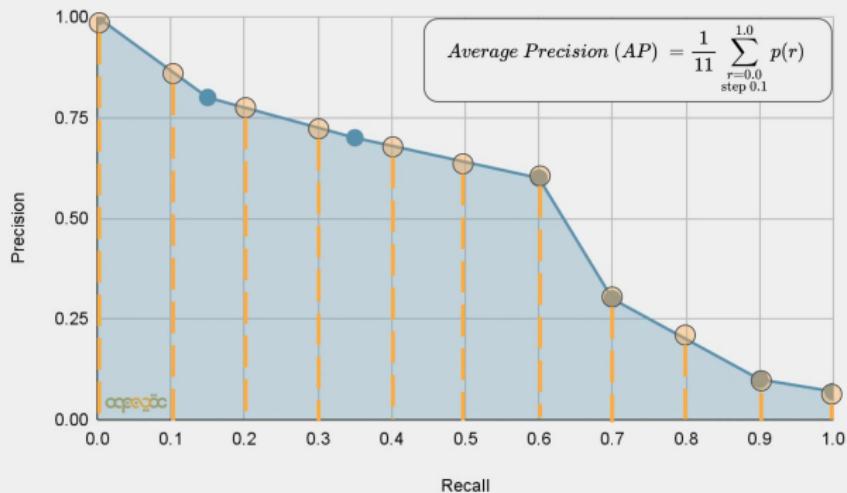
# Concretely Calculating AP

## Calculating Average Precision from PR Curve

### Approach 2: Interpolation and 11-point average

- The precision values for the 11 recall values from 0.0 to 1.0 with an increment of 0.1 is calculated
- These 11 points can be seen as orange samples in the figure on the right
- AP can be calculated by taking the mean of these 11 precision values i.e.

$$(AP) = \frac{1}{11} \sum_{r=0.0}^{1.0} p(r)$$



@\_aqueelanwar

aqeelanwarmalik

## Datasets

---

# Outline : Datasets

---

Introduction

**Datasets**

Models

Object Detection

Segmentation

Others and SOTA

# Datasets I

## COCO (Common Objects in Context) [7]

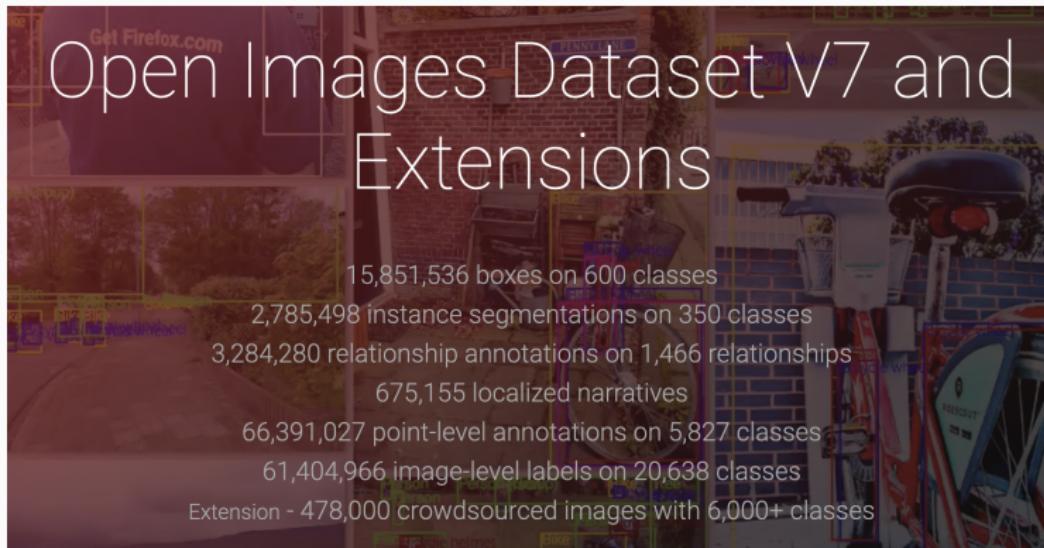
- **Type:** Bounding Boxes, Segmentation Masks
- **Details:** The COCO dataset is one of the most widely used for object detection and segmentation. It includes over 330,000 images, with 80 object categories, and provides annotations for bounding boxes, segmentation masks, keypoints, and captions.
- **Applications:** Object detection, segmentation, human pose estimation.



## Datasets II

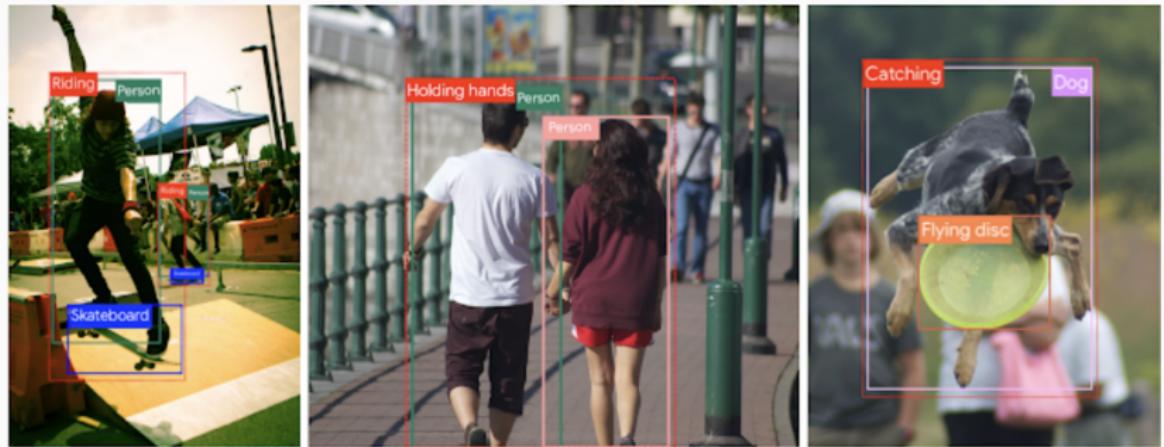
### Pascal VOC [1]

- **Type:** Bounding Boxes, Segmentation Masks
- **Details:** 20 object classes.
- **Applications:** detection, semantic segmentation, classification.



**Figure 1: Open Images Dataset [6]**

# Visual Relationship



**Figure 2:** Open Image contains rich annotations, like visual relationships

# Models

---

# Outline : Models

---

Introduction

Datasets

**Models**

Object Detection

Segmentation

Others and SOTA

# Outline : Object Detection

Introduction

Datasets

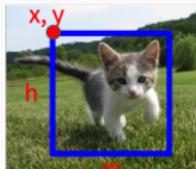
**Models**

Object Detection

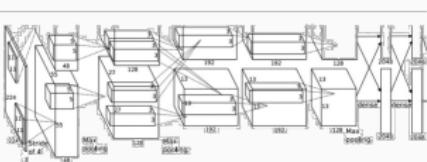
Segmentation

Others and SOTA

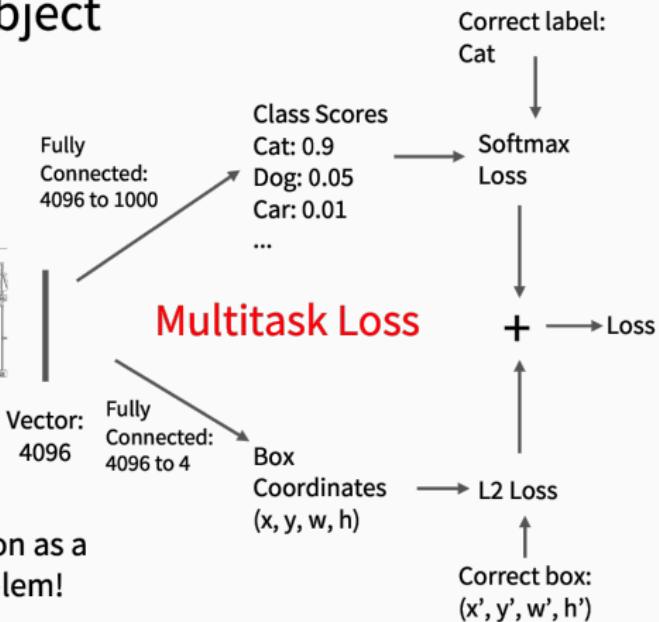
## Object Detection: Single Object (Classification + Localization)



This image is CC0 public domain



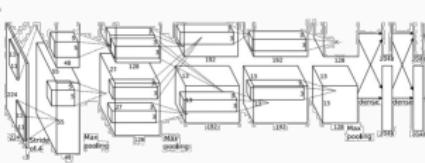
Treat localization as a  
regression problem!



Bounding boxes are represented by 4-uples with the starting point, height and width

# Basis

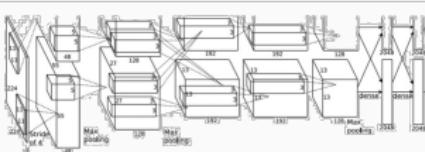
## Object Detection: Multiple Objects



Each image needs a different number of outputs!

CAT: (x, y, w, h)

4 numbers

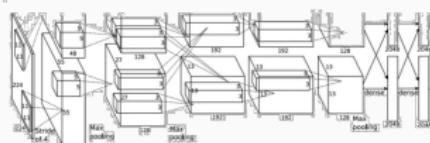


DOG: (x, y, w, h)

12 numbers

DOG: (x, y, w, h)

CAT: (x, y, w, h)



DUCK: (x, y, w, h)

Many numbers!

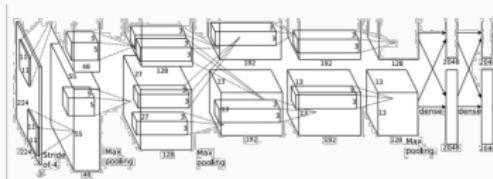
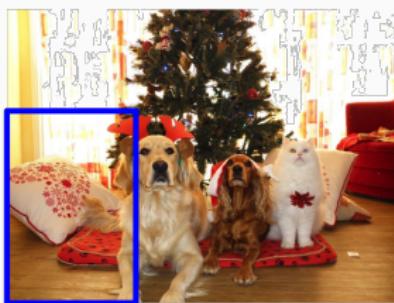
DUCK: (x, y, w, h)

....

Bounding boxes are represented by 4-uples with the starting point, height and width

## Object Detection: Multiple Objects

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

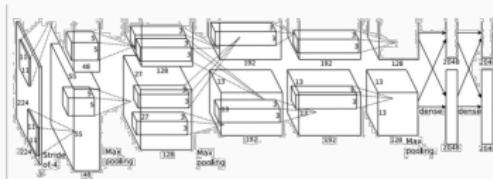


Dog? NO  
Cat? NO  
Background? YES

Bounding boxes are represented by 4-uples with the starting point, height and width

## Object Detection: Multiple Objects

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

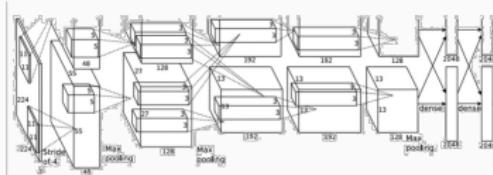


Dog? YES  
Cat? NO  
Background? NO

Bounding boxes are represented by 4-uples with the starting point, height and width

## Object Detection: Multiple Objects

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

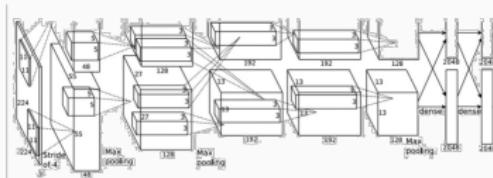


Dog? YES  
Cat? NO  
Background? NO

Bounding boxes are represented by 4-uples with the starting point, height and width

## Object Detection: Multiple Objects

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



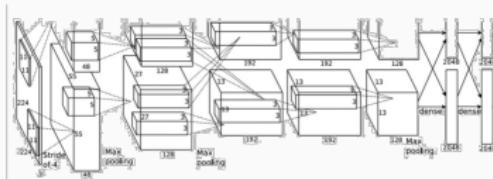
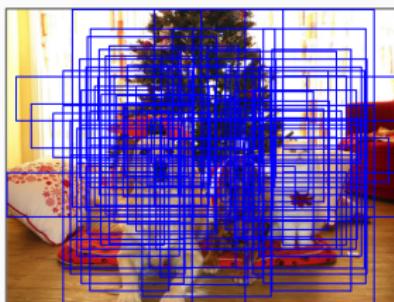
Dog? NO  
Cat? YES  
Background? NO

Q: What's the problem with this approach?

Bounding boxes are represented by 4-uples with the starting point, height and width

## Object Detection: Multiple Objects

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO  
Cat? YES  
Background? NO

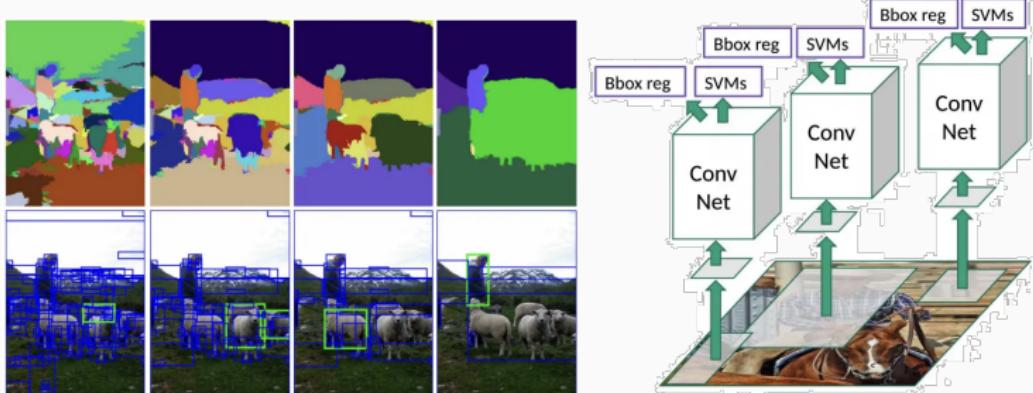
Problem: Need to apply CNN to huge number of locations, scales, and aspect ratios, very computationally expensive!

Bounding boxes are represented by 4-uples with the starting point, height and width

# R-CNN [3]

Three-stage algorithm:

- Algorithm to generate candidate regions in images (Selective Search)
- Extract candidate region feature with CNN pre-trained
- Classification (object) + Regression (bounding box): SVM + least squares regressor. 25/75 training ratio on object/background



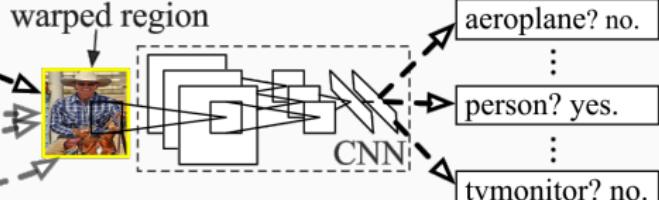
**Figure 3: Train time: Selective Search + 2 Loss Functions**

# R-CNN [3]

Three-stage algorithm:

- Algorithm to generate candidate regions in images (Selective Search)
- Extract candidate region feature with CNN pre-trained
- Classification (object) + Regression (bounding box): SVM + least squares regressor. 25/75 training ratio on object/background

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



1. Input image

2. Extract region proposals (~2k)

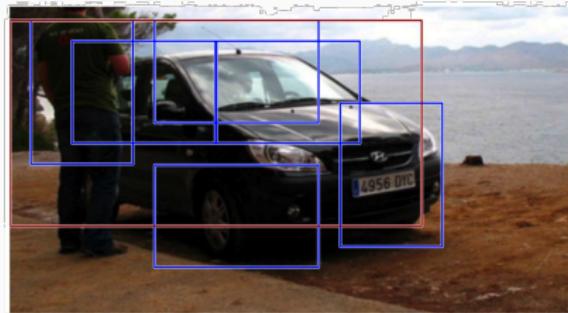
3. Compute CNN features

4. Classify regions

Figure 3: Test time: 2k region proposals, feature extraction + classification

## R-CNN [3]: Select the best boxes

- Suppression of non-maximums: many bounding boxes associated with the same object could be generated.
- Sort bounding boxes by confidence value. Choose the one with the highest value.
- Remove all bounding boxes with  $\text{IoU} \geq 0.5$  and same class as the chosen one.
- Repeat until no more bounding boxes can be removed.



Before non-max suppression



After non-max suppression

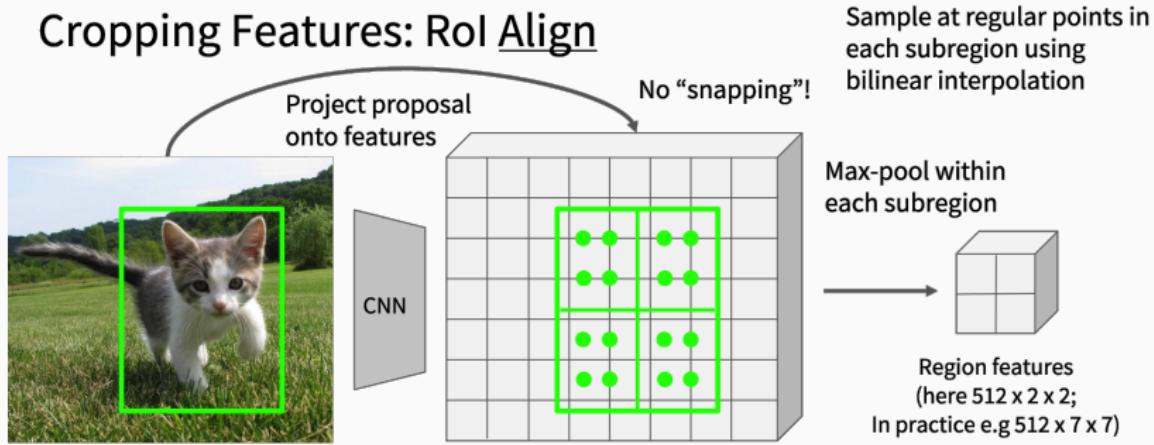
# Fast R-CNN [2]

## Idea

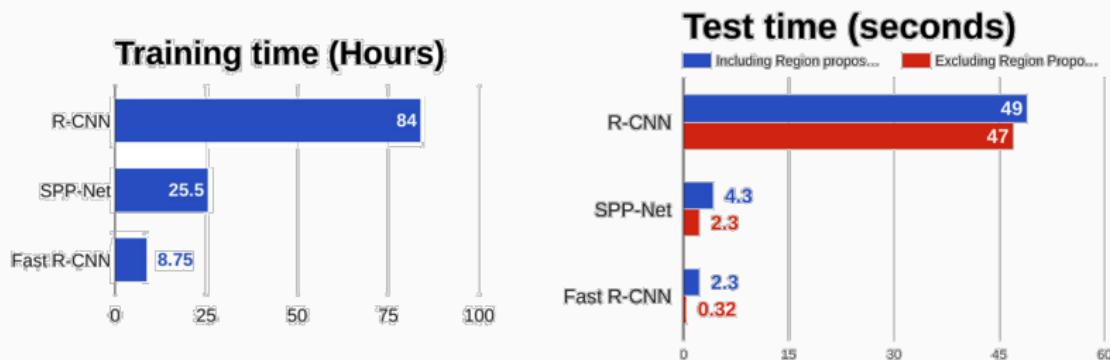
Why not run the CNN just once per image and then find a way to share that computation across the  $\approx 2000$  proposals?

- Warp the Rots directly on the activation maps
- Use a FC layers and then the 2 losses
- Convolution operation is done only once per image and a feature map is generated from it

## Cropping Features: RoI Align

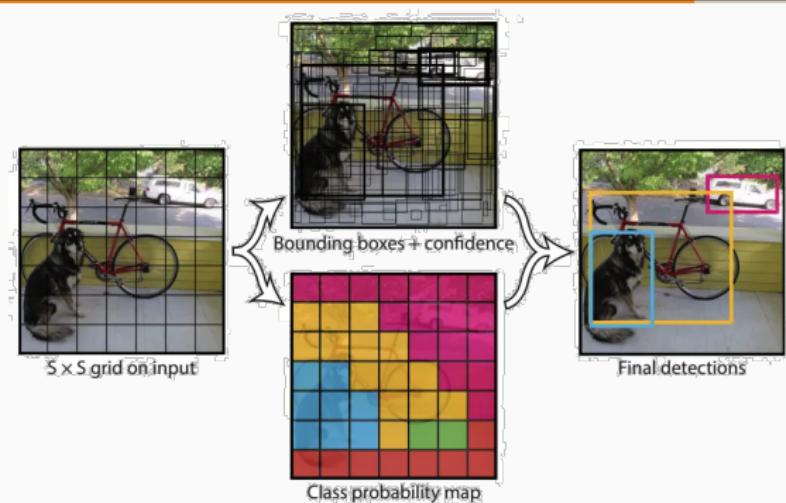


# Fast R-CNN [2]



**Figure 4:** Way Faster than R-CNN

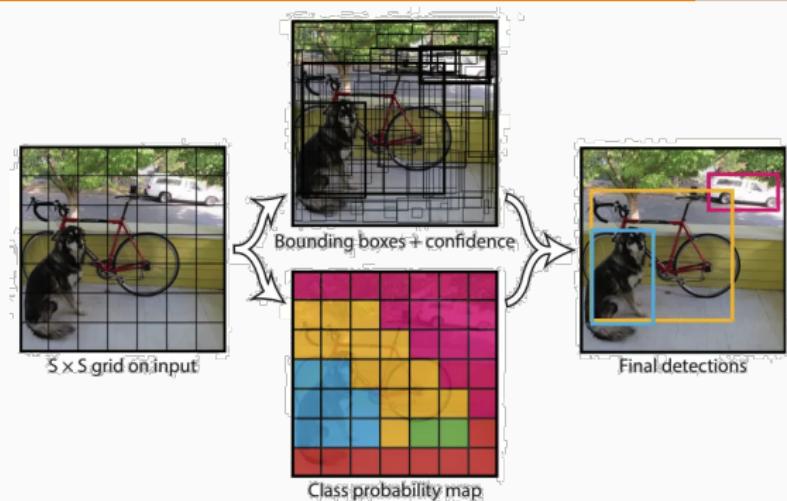
# YOLO [8]



For every cell in the grid predict  $B$  (number maximum of Boxes):

- Bounding box size and center coordinate (4 dims)
- Probability of bounding box (1 dim)
- Classes probabilities ( $C$  dims)

# YOLO [8]



For every cell in the grid predict  $B$  (number maximum of Boxes):

- Bounding box size and center coordinate (4 dims)
- Probability of bounding box (1 dim)
- Classes probabilities ( $C$  dims)

⇒ **Vector of size  $B * (1 + 4 + C)$**

# Outline : Segmentation

Introduction

Datasets

**Models**

Object Detection

Segmentation

Others and SOTA

# From faster RCNN [9] to Mask-RCNN [4]

## Object Detection: Faster R-CNN

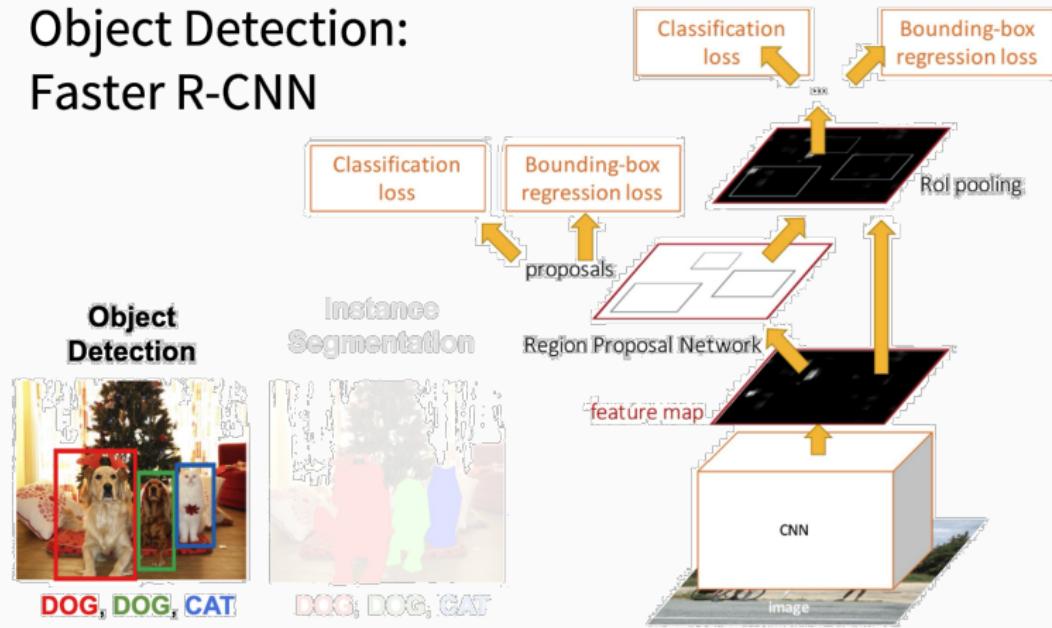
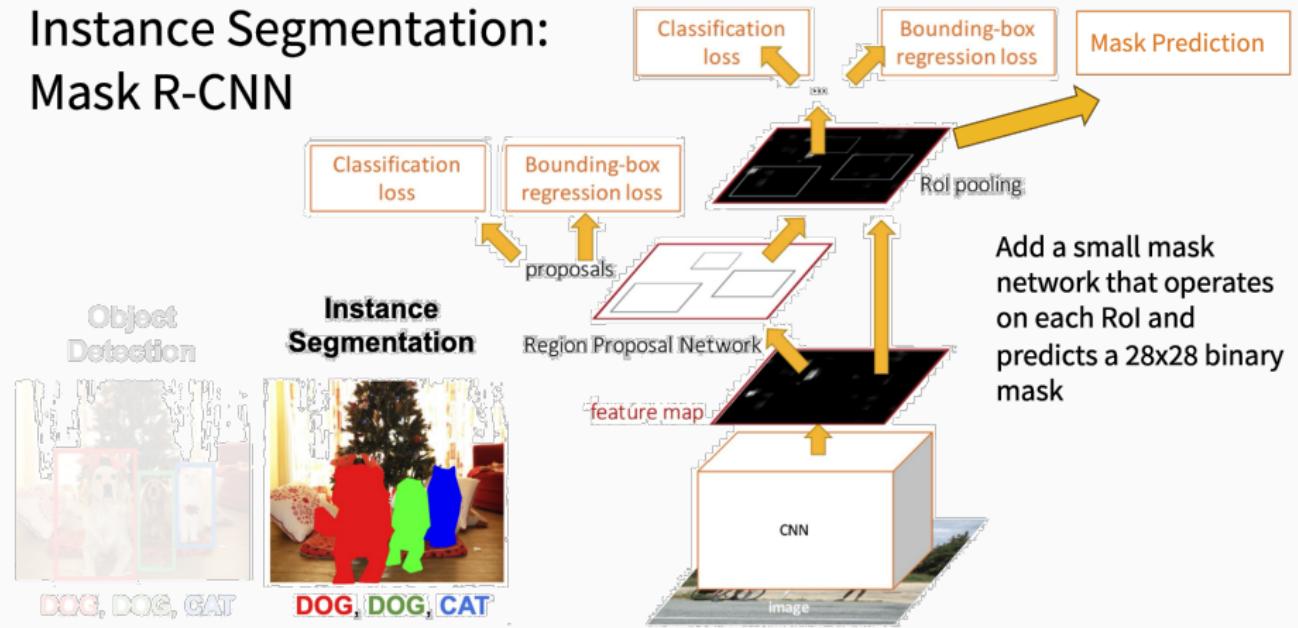


Figure 5: Faster R-CNN

# From faster RCNN [9] to Mask-RCNN [4]

## Instance Segmentation: Mask R-CNN



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

Figure 5: Mask R-CNN

# Mask-RCNN [4]

## Mask R-CNN

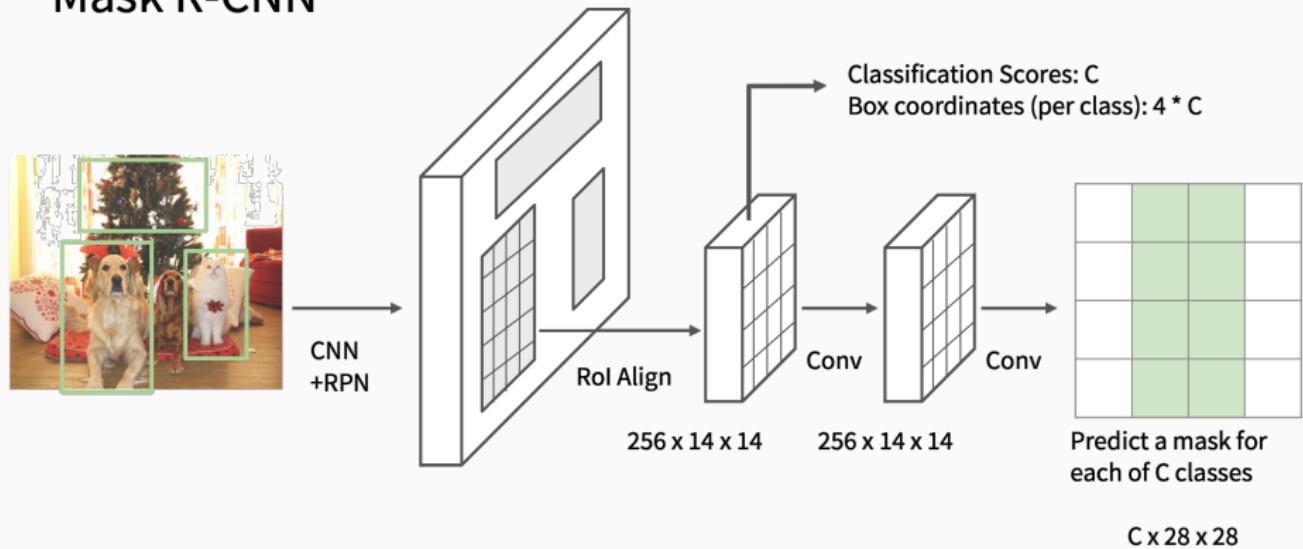
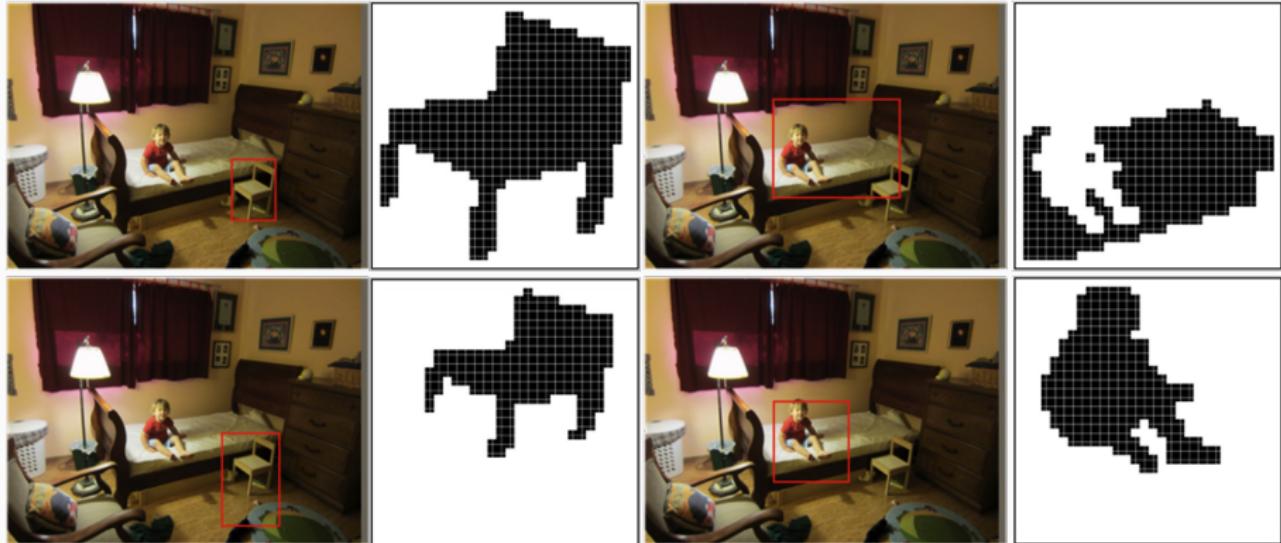


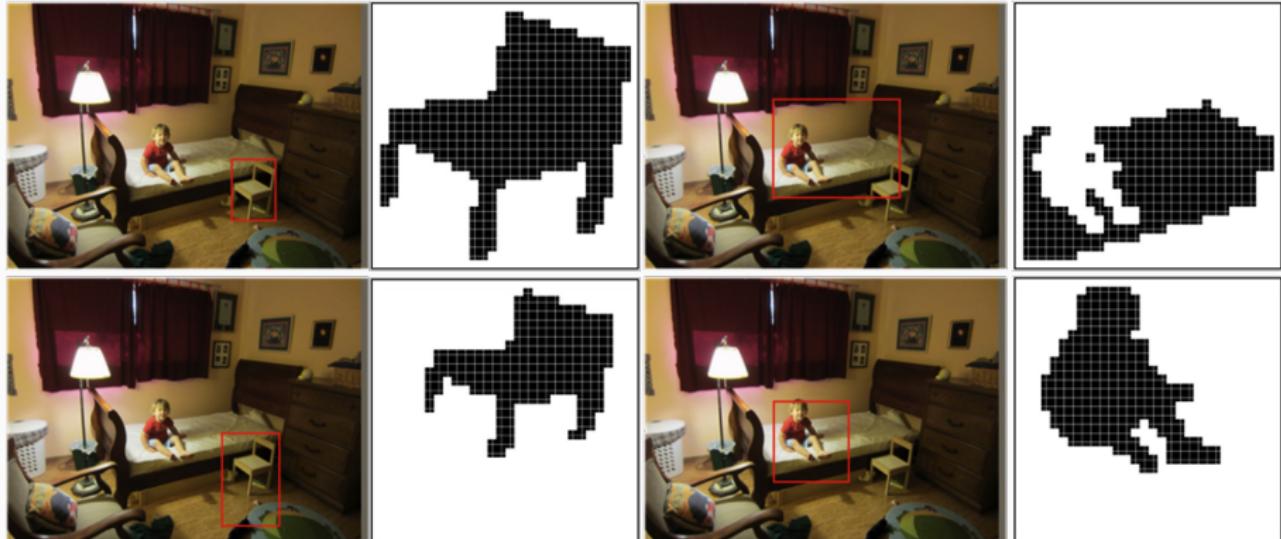
Figure 6: Mask R-CNN

# Mask-RCNN [4]



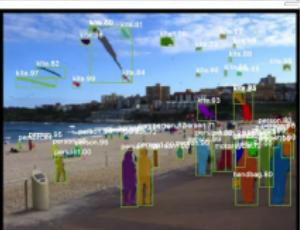
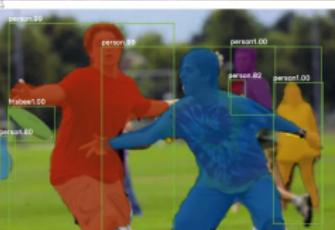
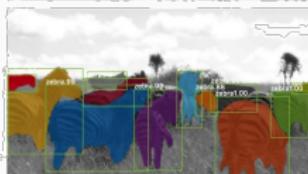
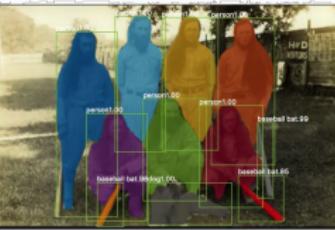
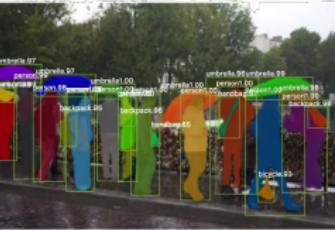
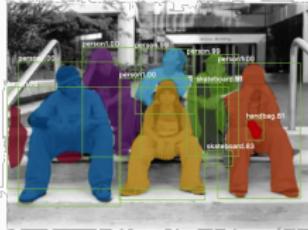
**Figure 7:** Mask R-CNN

# Output Example

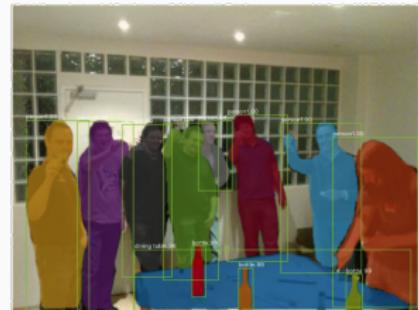
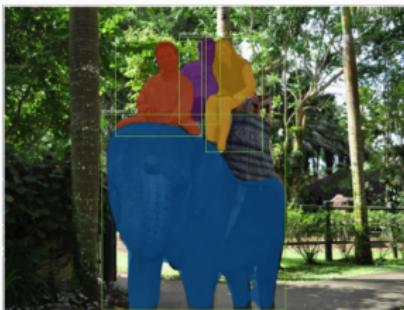


**Figure 8:** Training Examples

## Output Examples



# Output Examples



# Outline : Others and SOTA

---

Introduction

Datasets

**Models**

Object Detection

Segmentation

Others and SOTA

# Transformers-based: Detectron2

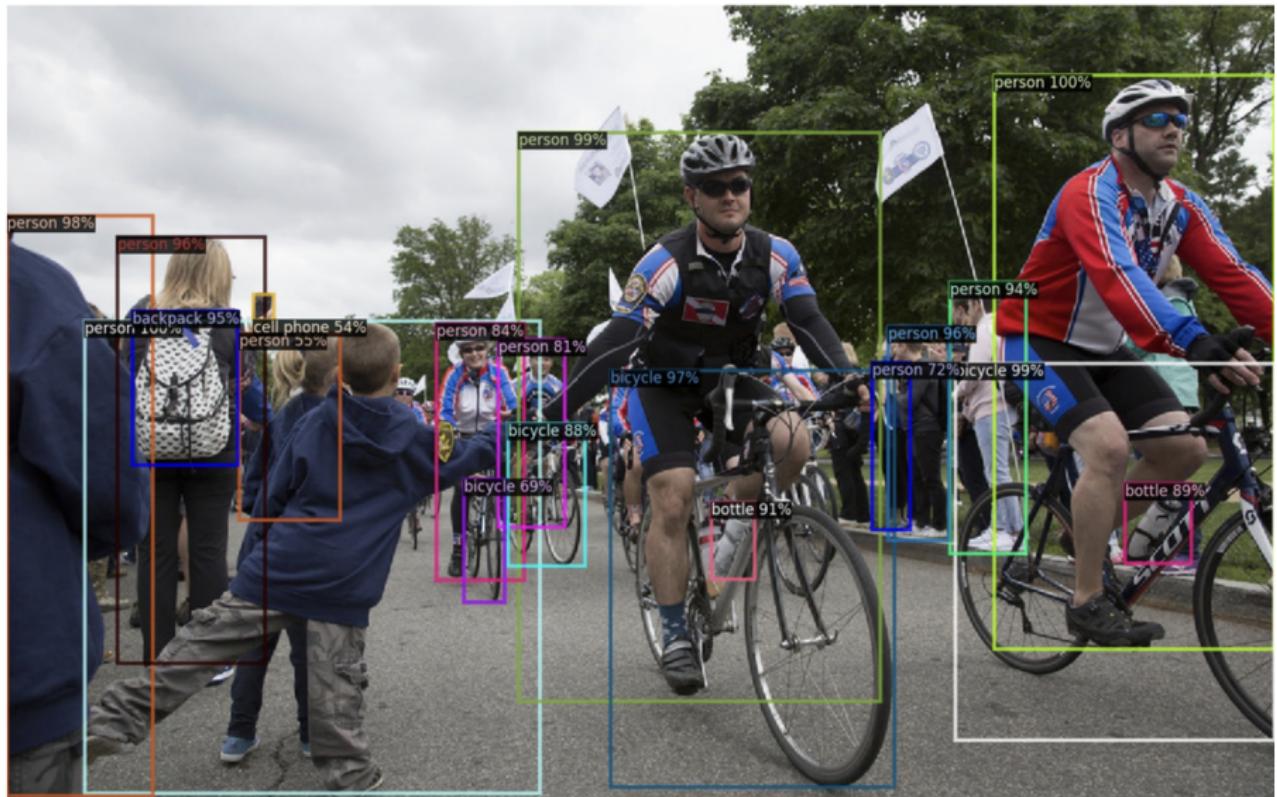


Figure 9: Detectron2 available [online](#)

# Transformers-based: Detectron2



Figure 9: Detectron2 available [online](#)

# Transformers-based: Detectron2



Figure 9: Detectron2 available [online](#)

# Transformers-based: Detectron2



Figure 9: Detectron2 available [online](#)

# Transformers-based: Detectron2



Figure 9: Detectron2 available [online](#)

# Yolov11

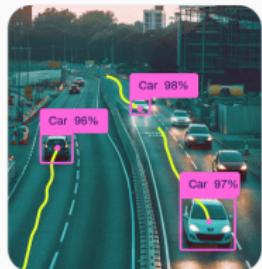
Classify

Detect

Segment

Track

Pose



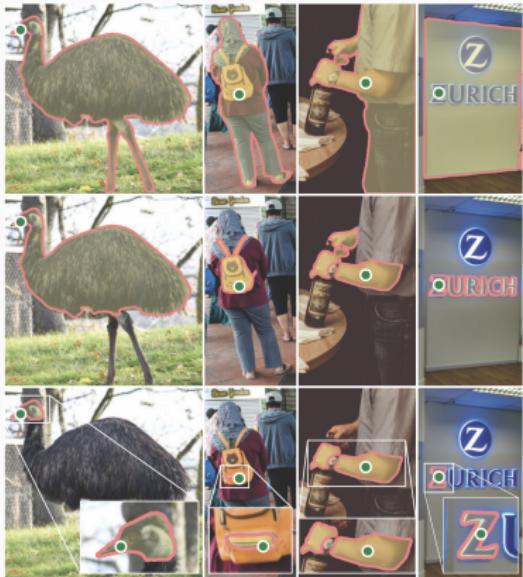
Many more tasks are now possible using the new off-the-shelf models, such as Yolov11.

# Segment Anything Model [5]



- SAM can segment automatically

# Segment Anything Model [5]



- SAM can segment automatically
- SAM can **interacts with the user** to find the right object

# Segment Anything Model [5]



- SAM can segment automatically
- SAM can **interacts with the user** to find the right object
- Other similar models exists such as SEEM [10]

**Questions?**

## References i

-  M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman.  
**The pascal visual object classes (VOC) challenge.**  
International Journal of Computer Vision, 88(2):303–338, 2010.
-  R. Girshick.  
**Fast R-CNN.**  
In Proceedings of the IEEE International Conference on Computer Vision, volume 2015 Inter, pages 1440–1448, 2015.
-  R. Girshick, J. Donahue, T. Darrell, and J. Malik.  
**Rich feature hierarchies for accurate object detection and semantic segmentation.**  
In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 580–587, 2014.

## References ii

-  K. He, G. Gkioxari, P. Dollár, and R. Girshick.  
**Mask R-CNN.**  
IEEE Transactions on Pattern Analysis and Machine Intelligence,  
42(2):386–397, 2020.
-  A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson,  
T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and  
R. Girshick.  
**Segment Anything.**  
2023.
-  A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin,  
J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, and  
others.  
**The open images dataset v4: Unified image classification,  
object detection, and visual relationship detection at scale.**  
International journal of computer vision, 128(7):1956–1981, 2020.

## References iii

-  T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick.  
**Microsoft COCO: Common objects in context.**  
In ECCV, volume 8693 LNCS, pages 740–755, 2014.
-  J. Redmon, S. Divvala, R. Girshick, and A. Farhadi.  
**You only look once: Unified, real-time object detection.**  
In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 2016-Decem, pages 779–788. IEEE, 2016.
-  S. Ren, K. He, R. Girshick, and J. Sun.  
**Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks Shaoqing.**  
In Neurips, 2015.

-  X. Zou, J. Yang, H. Zhang, F. Li, L. Li, J. Gao, and Y. J. Lee.  
**Segment Everything Everywhere All at Once.**  
2023.