



**POLITECNICO  
DI TORINO**



# Data Science Lab

Image understanding  
Tasks and architectures

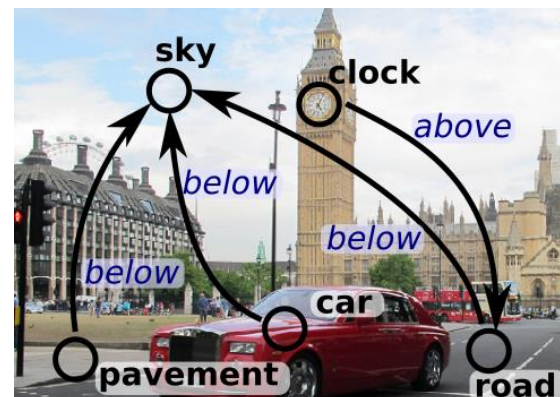
DataBase and Data Mining Group

Andrea Pasini



# Image understanding

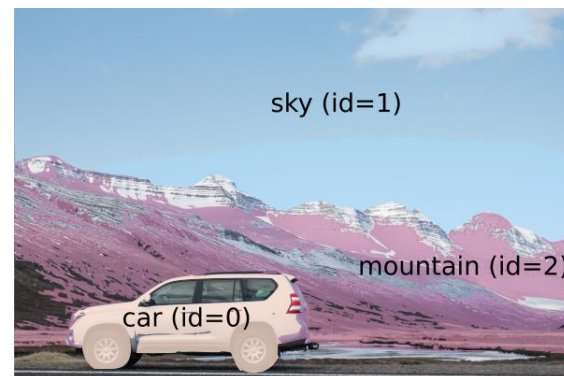
- Image understanding
  - Find objects inside images
  - Analyze their shape and position



- Applications
  - Image annotation (e.g., Google Photos, Pinterest)
  - Video annotation (e.g., YouTube)
  - Autonomous driving and robotics



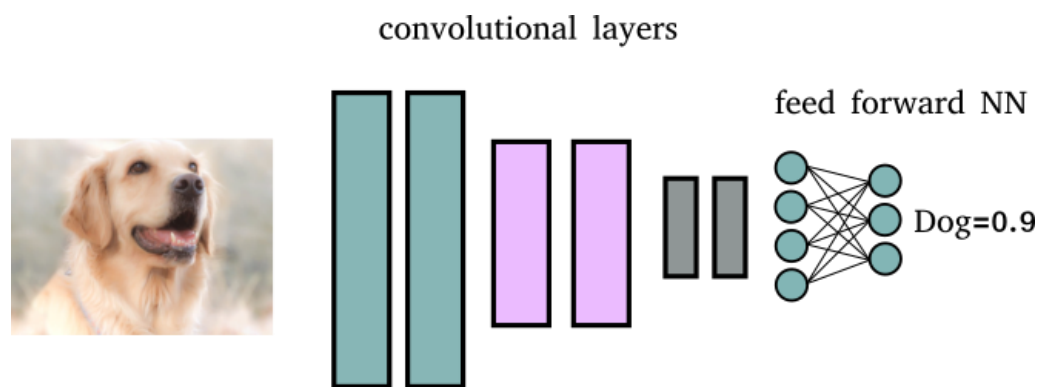
- **Taxonomy** of the tasks
  - Image classification
  - Object detection
  - Semantic segmentation
  - Instance segmentation
  - Panoptic segmentation





# Image classification

- Predict the probability of an image of belonging to a specific class



- **Convolutional** filters extract features from the input tensor
- Final layer is a fully connected **MLP**
  - Typically exploits the **softmax** function



## ■ Evaluation

- Precision, recall, f-measure

- Top-5 accuracy (% true positives):

Ground truth: Husky



Pred = (**0.45**, **0.42**, **0.063**, **0.031**, **0.017**, 0.012, 0.011,...)

Wolf   Husky   Setter   Goat   Horse

True positive

Top-5 predicted classes

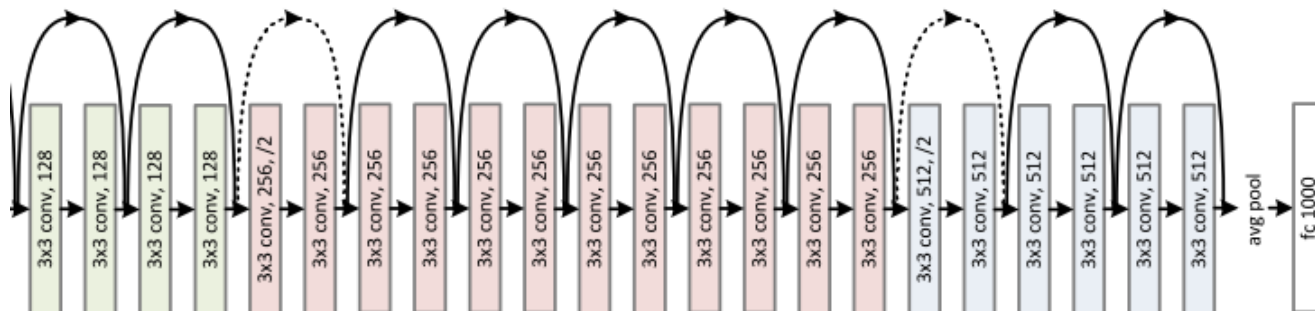


- Alexnet (2012) [1]



- Resnet (2015)<sub>[2]</sub>

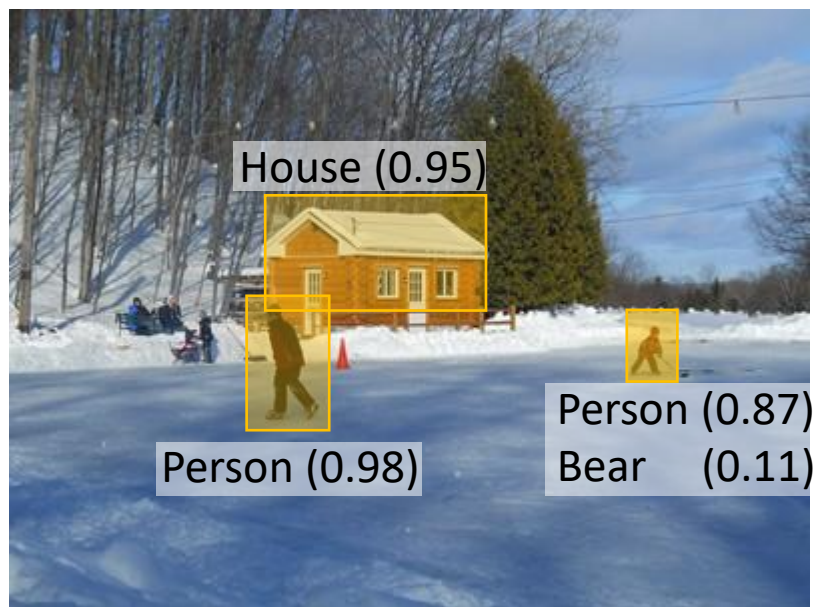
Top-5 Precision = 96% (Resnet152)





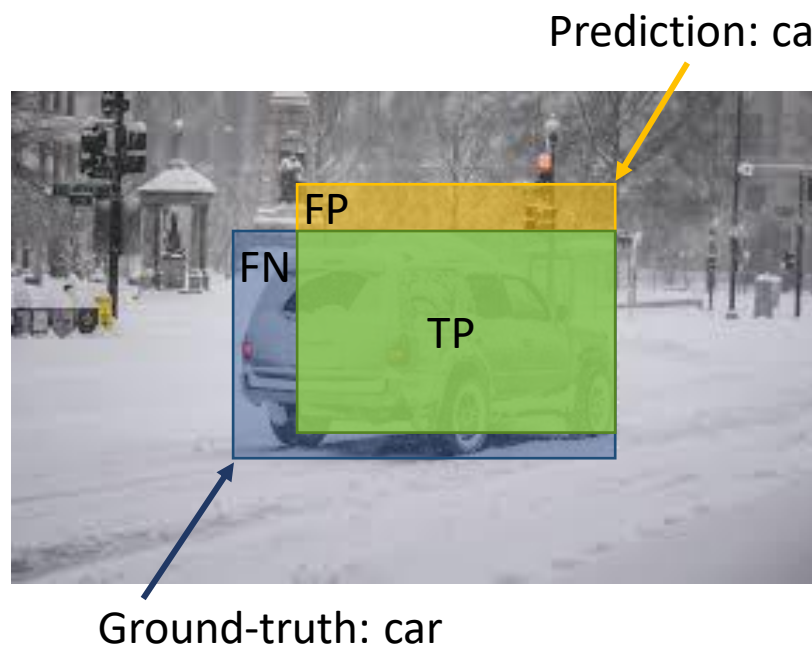
# Object detection

- Find **multiple objects** in the same image
- Each object is identified by a **bounding box**
- Return **class probabilities** for each bounding box





- Evaluation of a **single** detection
  - IoU (Intersection over Union) in range  $[0, 1]$



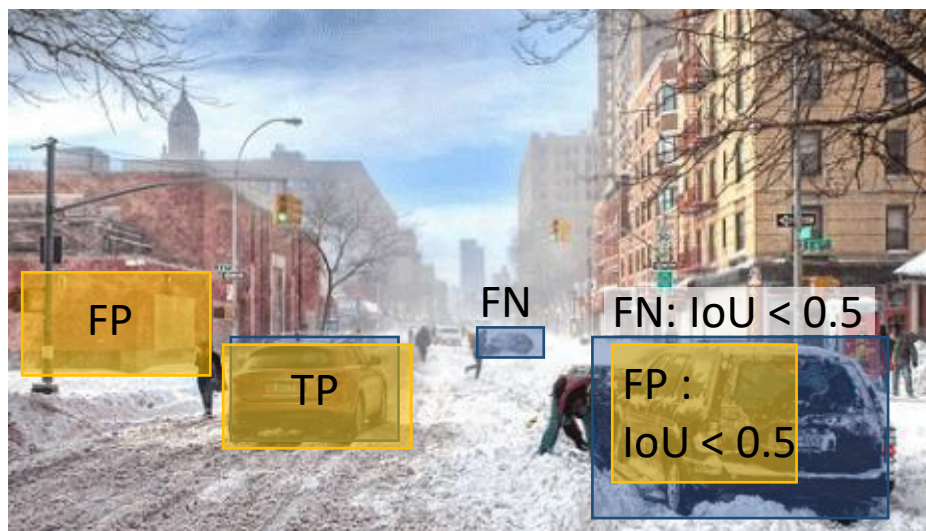
$$\text{IoU} = \text{intersection} / \text{union} \\ = \text{TP} / (\text{FP} + \text{FN} + \text{TP})$$





# Object detection

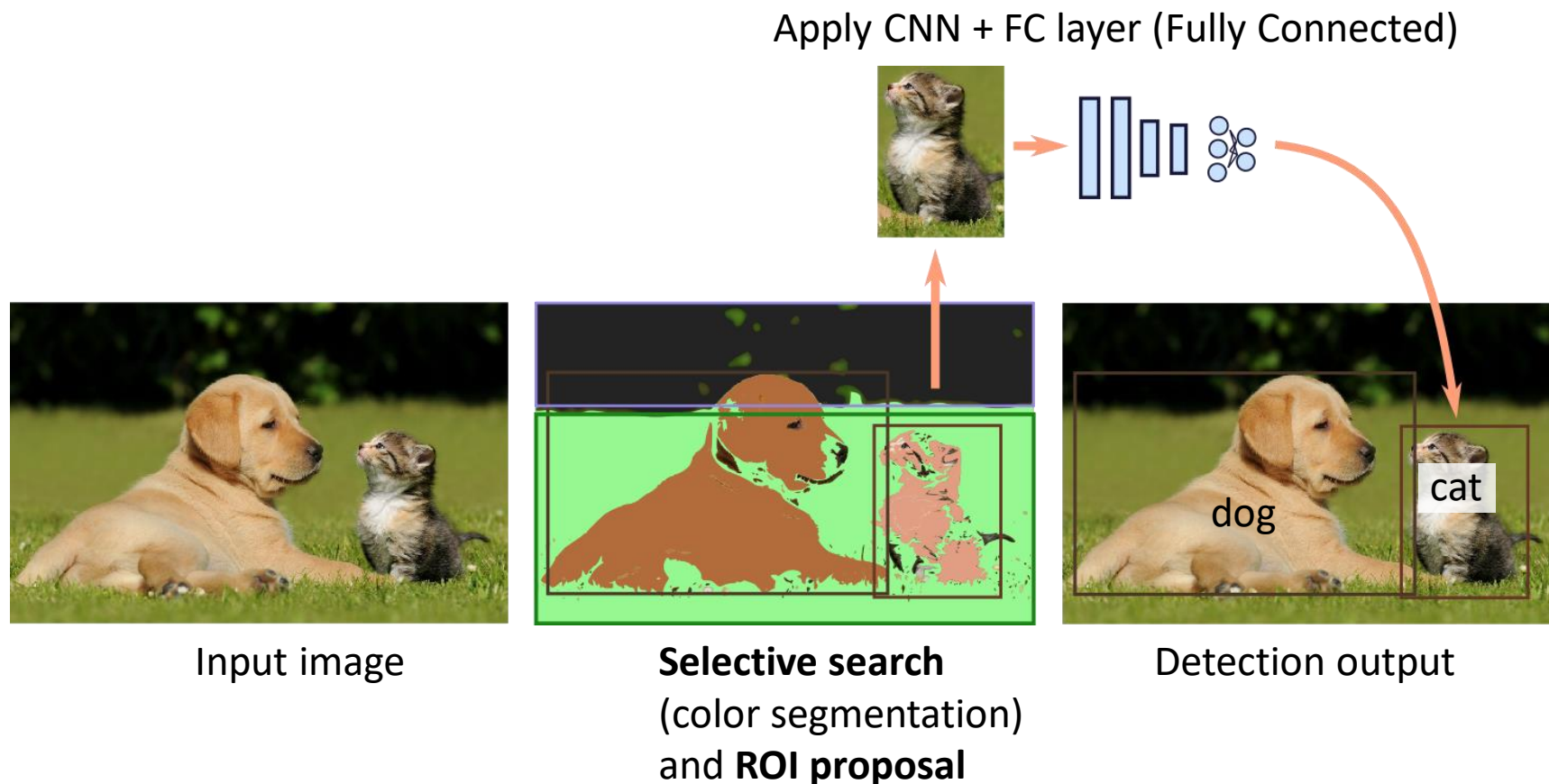
- Evaluation of multiple detections
  - Mean Average Precision (mAP)
    - See appendix





# Object detection

- **R-CNN** (2014) [3]: Find regions, then classify with CNN
  - ROI = Region Of Interest

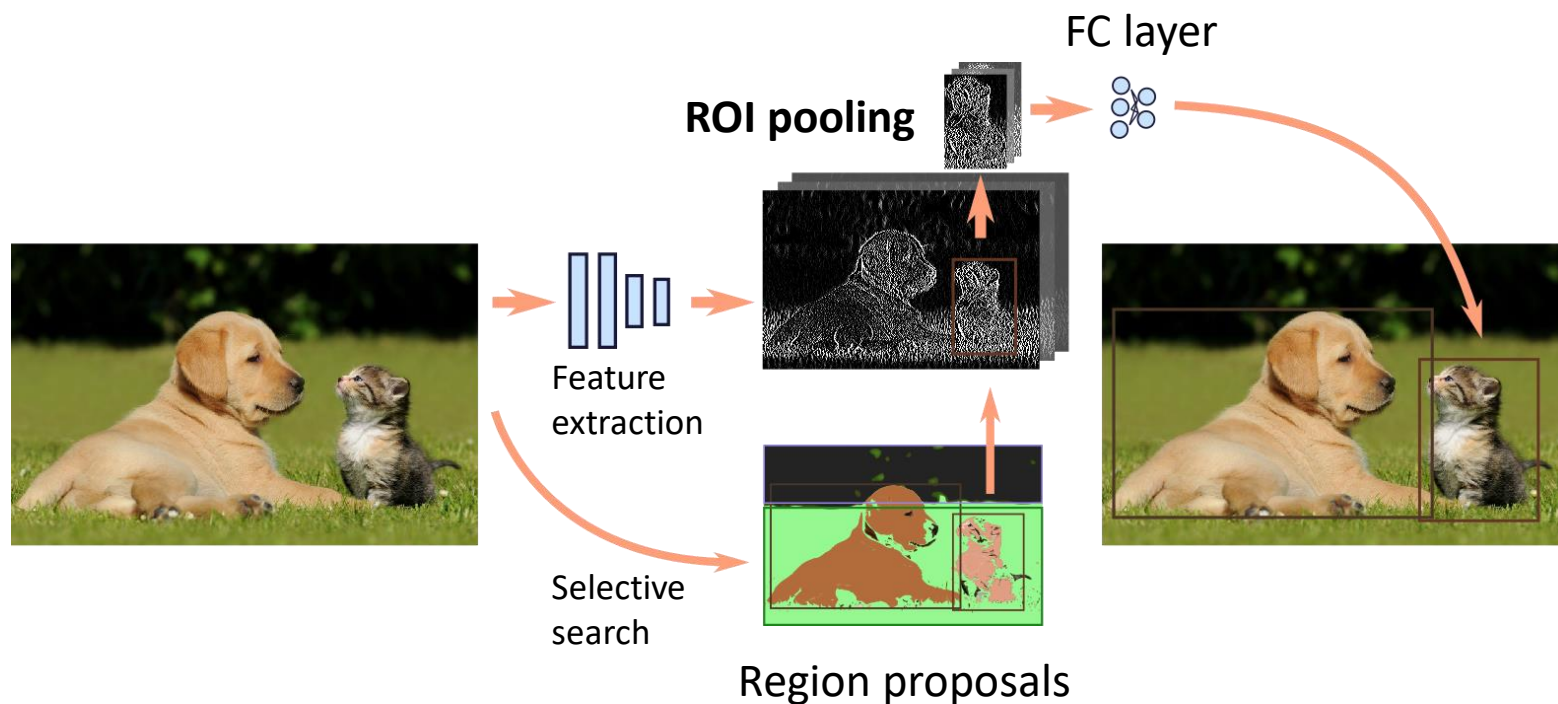




- R-CNN issues
  - Selective search is **slow** (sometimes inaccurate)
  - CNN applied **multiple** times (2K+ regions per image)
- Solution
  - Fast R-CNN:
    - Extract features with CNN only **once**
    - Apply **ROI pooling**

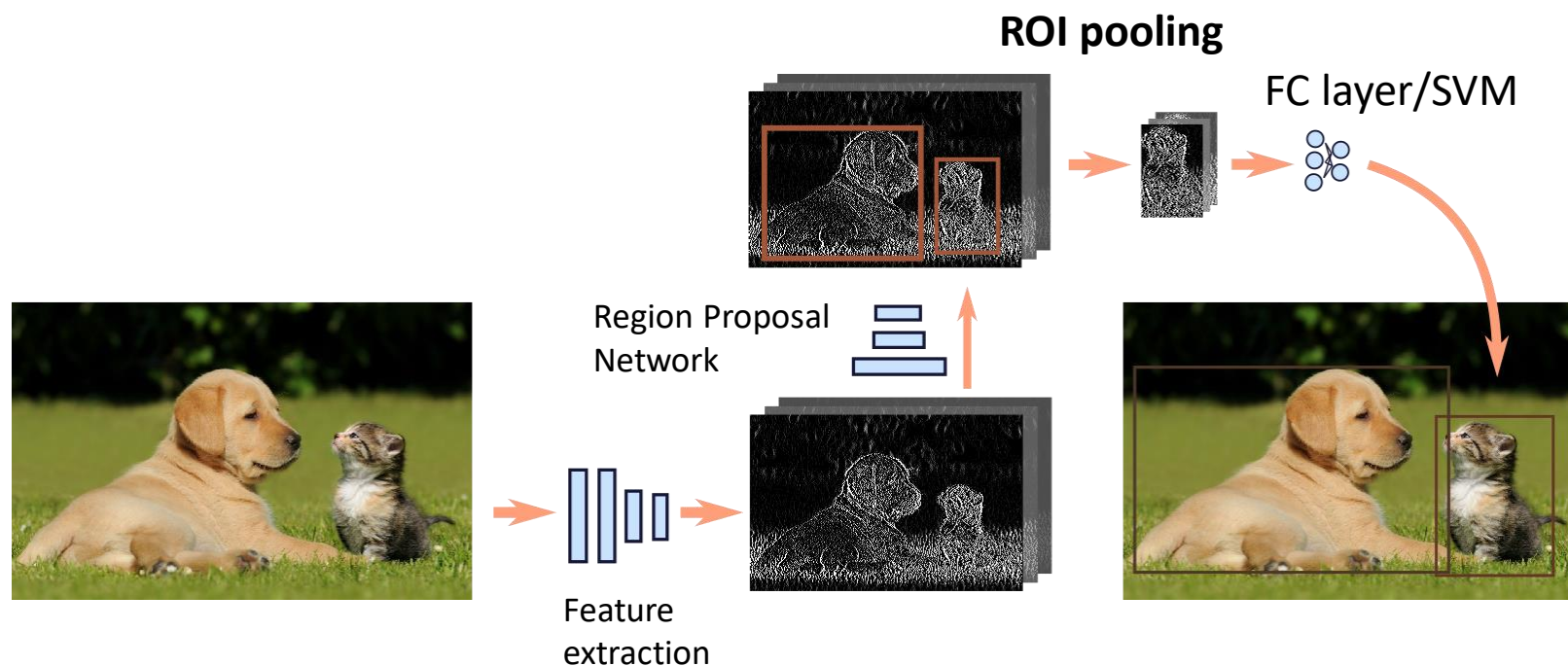


- **Fast R-CNN (2015)** [4]: ROI pooling





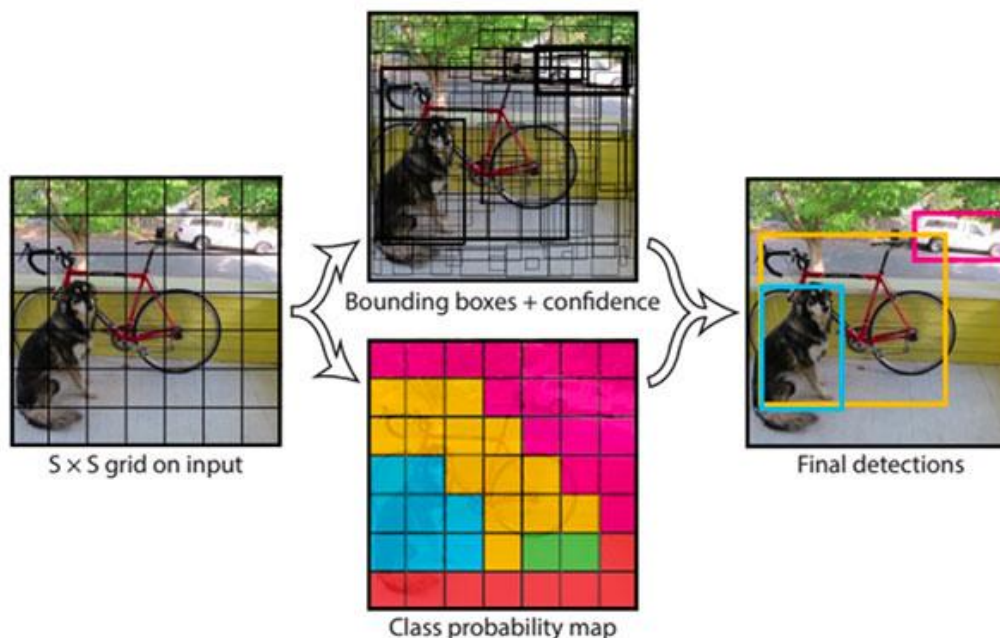
- **Faster R-CNN (2015) [5]:** Replace selective search with **region proposal network**





# Object detection

- Another famous model:
  - **YOLO - You Only Look Once – (2016)** [6]
    - **30-200 FPS** (Pascal Titan X GPU), YOLO v4
    - Up to **0.57** mean Average Precision (mAP) on Microsoft COCO dataset












# Image segmentation

- Predict the probability of each pixel of belonging to a specific class
- **Heavy** computation

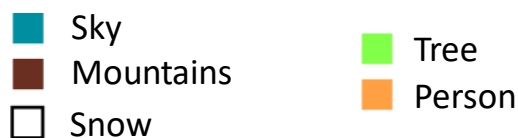


- |   |  |
|---|--|
|  Sky       |  Tree   |
|  Mountains |  Person |
|  Snow      |  |





- Evaluation:
  - **Pixel accuracy**: % of correct pixels
    - Also can be separated for **each class**
  - **IoU(class)**:  $(\text{n. correct pixels}) / (\text{pred} \cup \text{g-truth})$



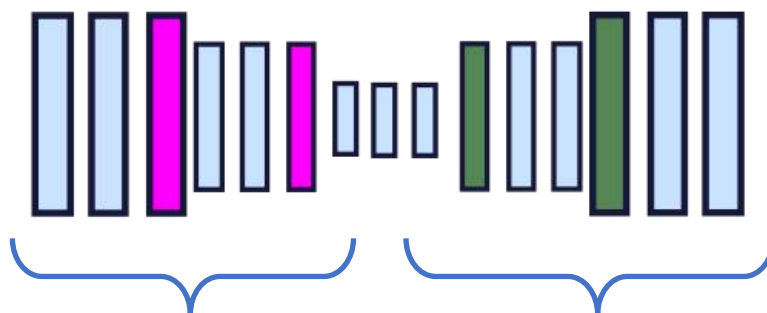




# Image segmentation

- Methods:
  - Encoder decoder networks

- Convolutional layer
- (down) pooling
- Upsampling (e.g., bilinear interpolation)



Encoder

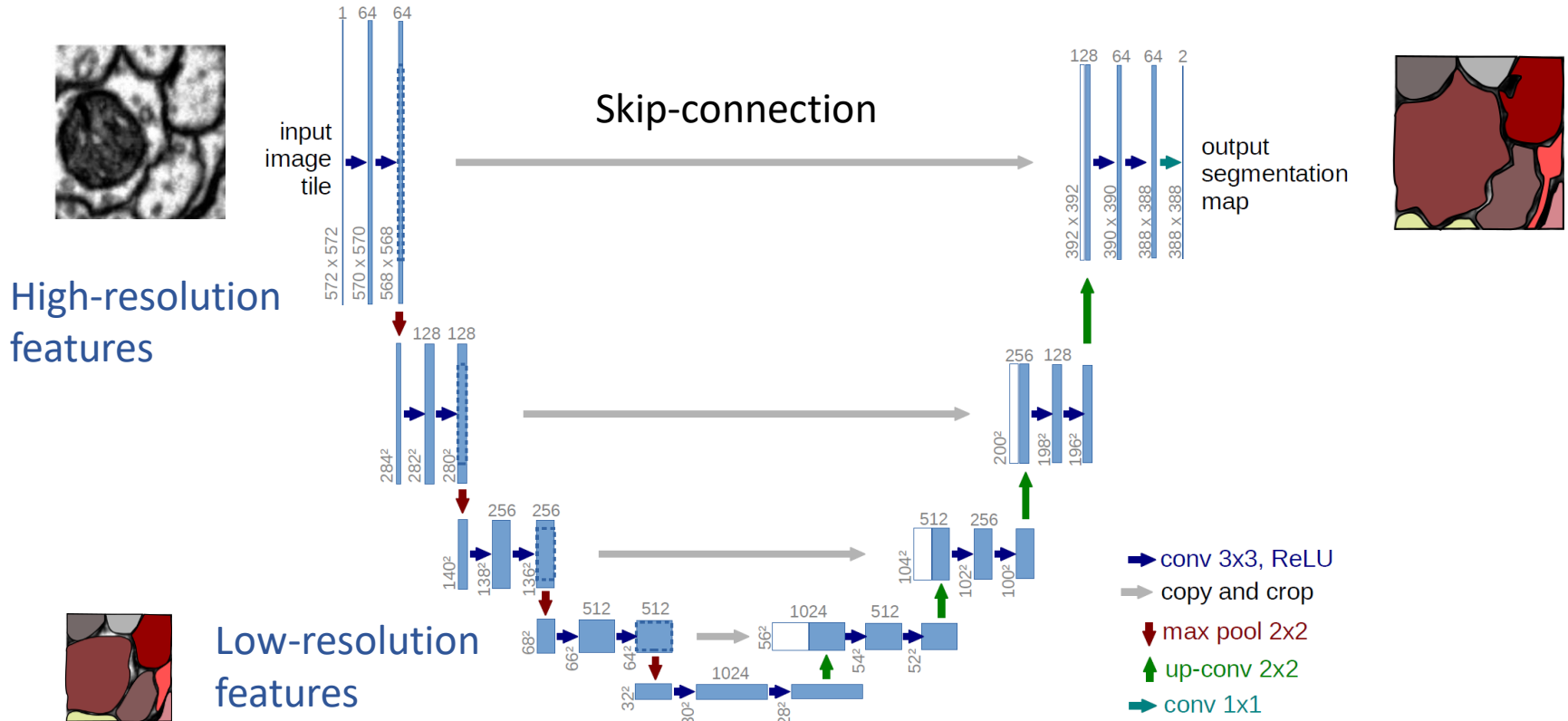
Decoder





- U-Net (2015) [7]

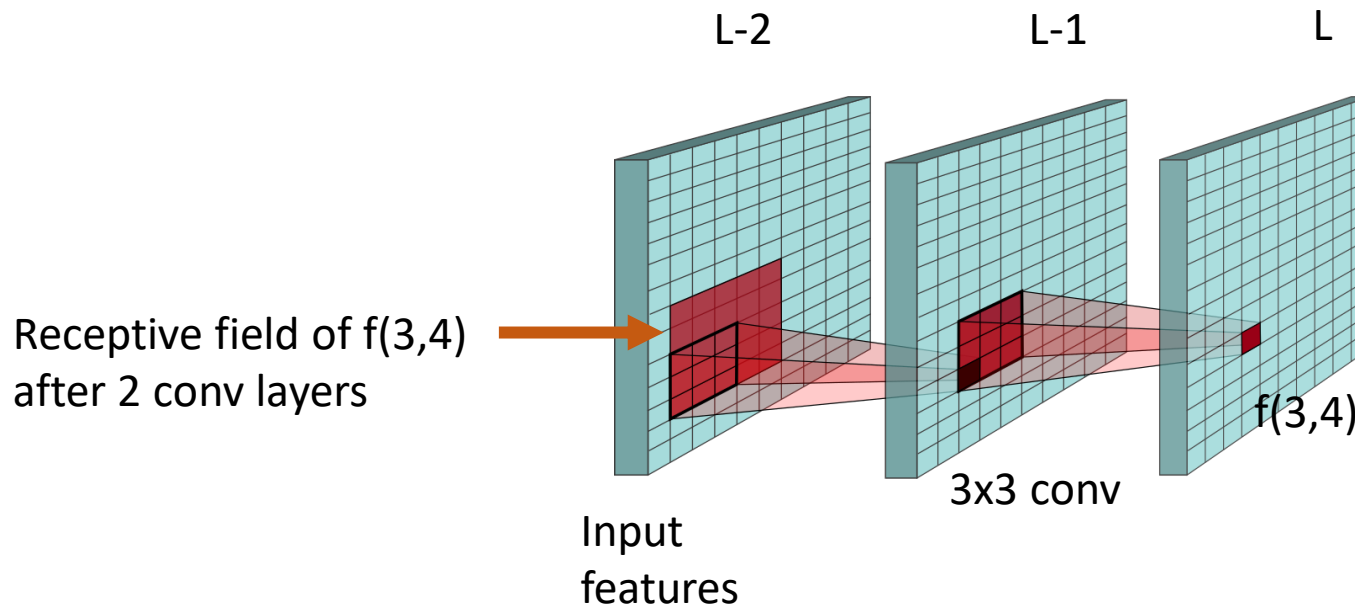
- **skip-connections:** provide high-resolution information to the decoder





## ■ Receptive field

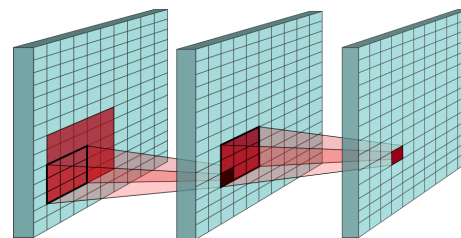
- How much **context** is it retrieved by feature  $f(3,4)$ ?



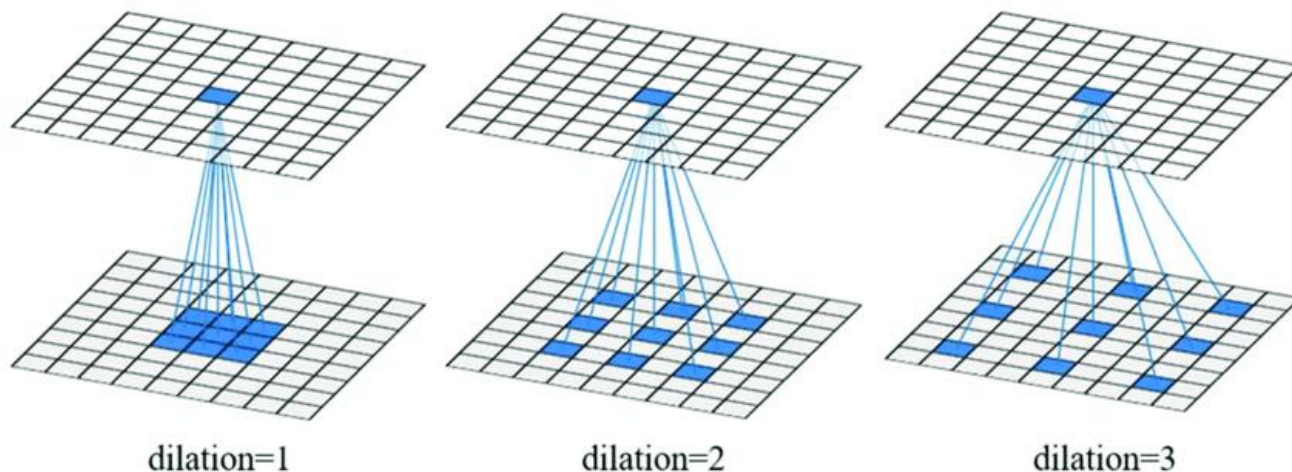


# Image segmentation

- How to increase receptive field?



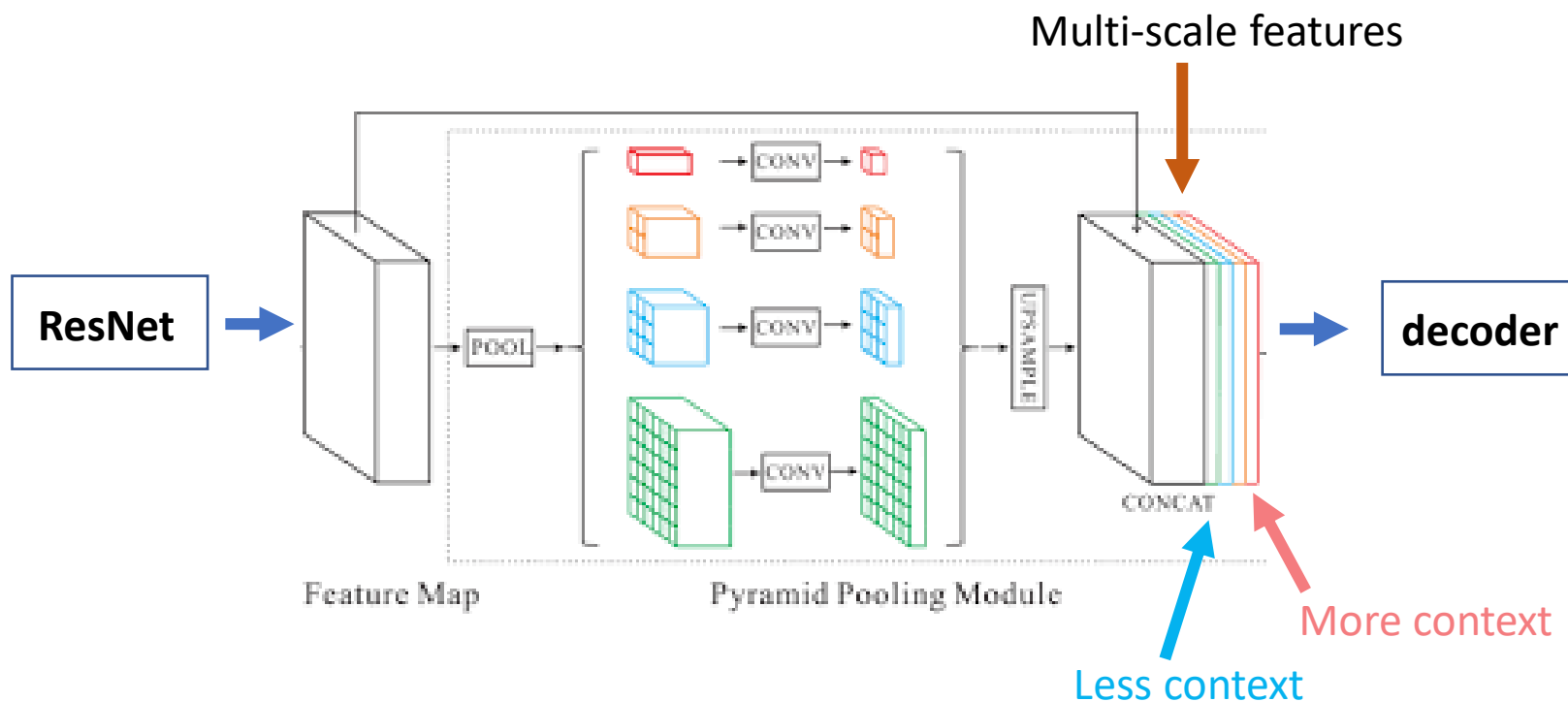
- Dilated Convolution (2016) [8]
  - **bigger sparse filters** to get more context





# Image segmentation

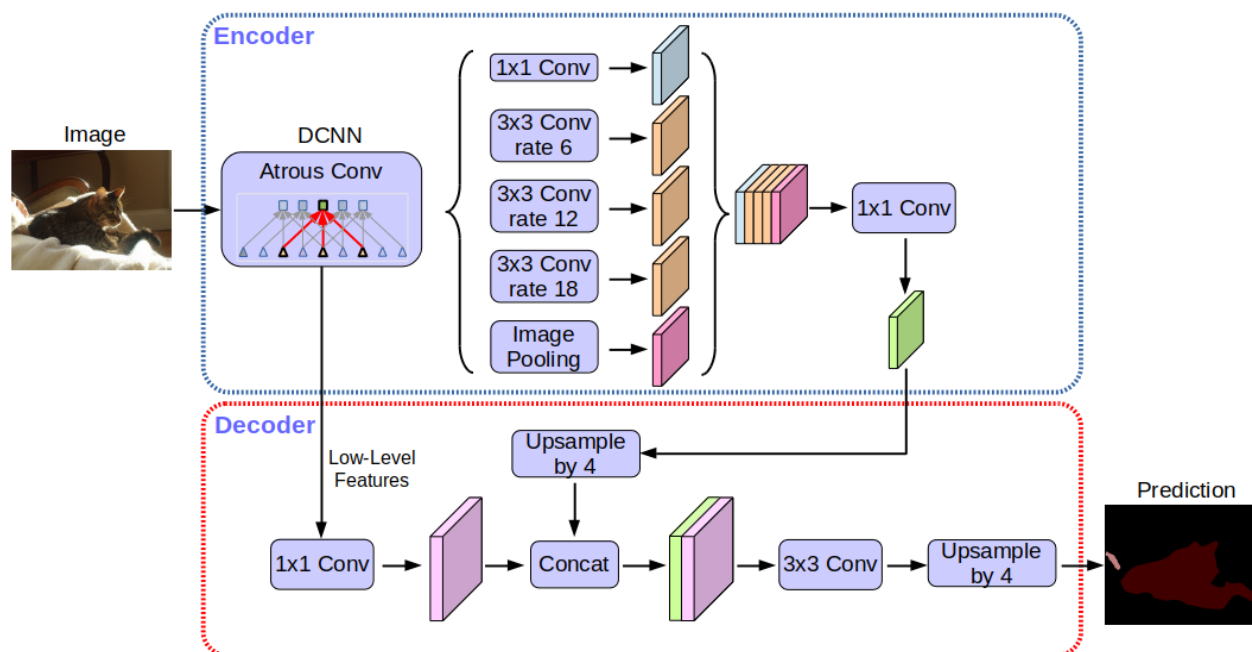
- PSPNet (2016) [9] – 81.2 mIoU on Cityscapes
  - **Pyramid pooling:** multi-size pooling filters
    - Allows capturing global image **context**





# Image segmentation

- DeepLab V3 (2017) <sup>[10]</sup> – 81.3 mIoU on Cityscapes
  - **Atrous Spatial Pyramid Pooling (ASPP):**
    - = Atrous (dilated) Convolution + pyramid pooling
  - **Multi-size** conv filters (similar to Pyramid Pooling)



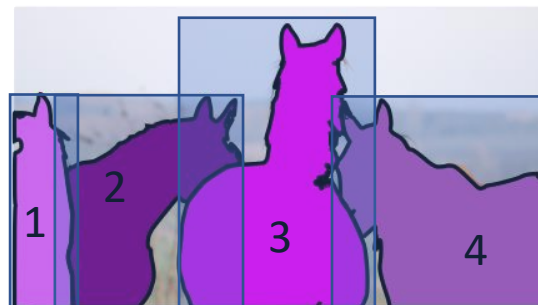


# Instance segmentation

- Segmentation does not distinguish between **instances**
- Instance segmentation: Detect bounding box + **mask**



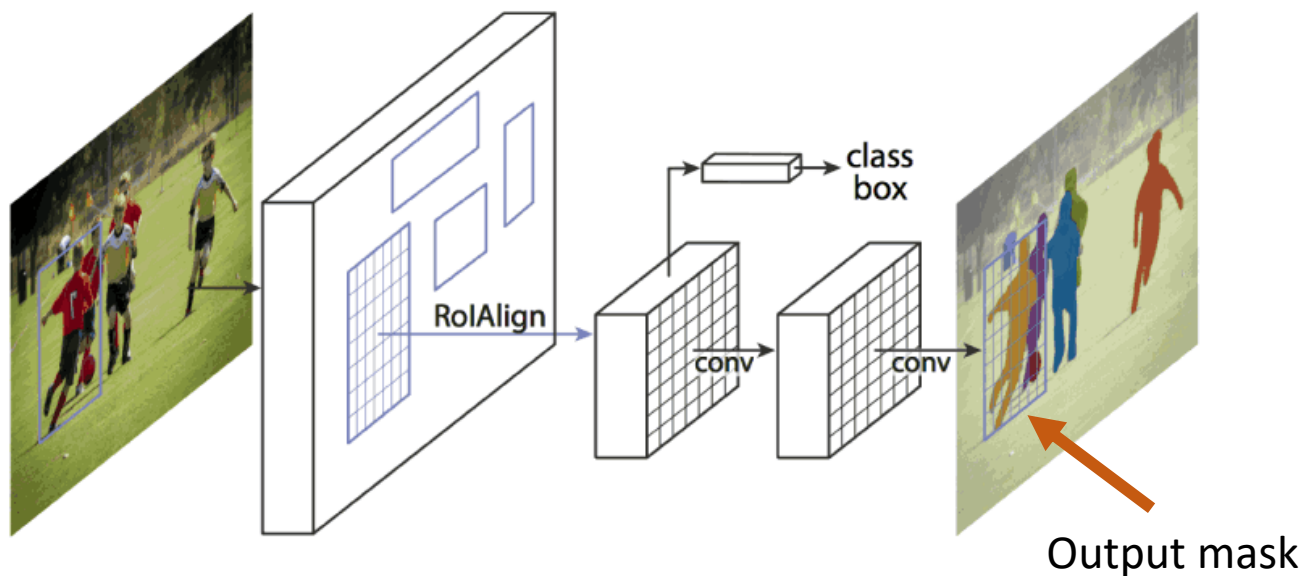
Semantic segmentation



Instance segmentation



- Mask RCNN (2017) [11]
  - Same structure as Faster-RCNN
  - Includes conv layers to generate a **mask** for each bbox







- Instance segmentation: What about uncountable objects?
  - Sky, grass, vegetation, dirt, road, ...
- **Panoptic segmentation**
  - Pan-optic, all you can see
    - Semantic segmentation for **stuff** (uncountable objects)
    - Instance segmentation for **objects**
  - Challenge launched in 2019 by **Microsoft COCO** <sup>[12]</sup>



## ■ Panoptic segmentation output

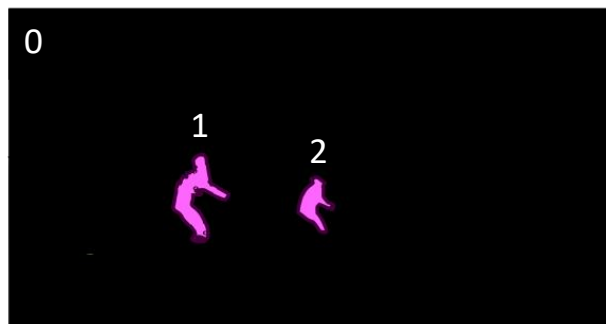
- Class matrix + instance matrix



- Tree
- Sky
- Mountains
- Snow
- Person



Stuff/Instance classes



Instance ids



- Methods:
  - **Heuristics** to merge results from semantic and instance segmentation
  - Single neural network:
    - **Panoptic Feature Pyramid Networks** (2019) [13]
      - **Unified** feature extraction (FPN)
      - Mask R-CNN + semantic segmentation branch
      - Merge with heuristic
    - **MaX-DeepLab** (2020) [14]
      - Based on transformers



# Bibliography

1. Imagenet classification with deep convolutional neural networks
2. Deep residual learning for image recognition
3. Rich feature hierarchies for accurate object detection and semantic segmentation
4. Fast r-cnn
5. Faster r-cnn: Towards real-time object detection with region proposal networks
6. You only look once: Unified, real-time object detection
7. U-Net: Convolutional Networks for Biomedical Image Segmentation
8. Multi-Scale Context Aggregation by Dilated Convolutions
9. Pyramid Scene Parsing Network
10. Rethinking Atrous Convolution for Semantic Image Segmentation
11. Mask r-cnn
12. Panoptic segmentation
13. Panoptic feature pyramid networks
14. MaX-DeepLab: End-to-End Panoptic Segmentation with Mask Transformers

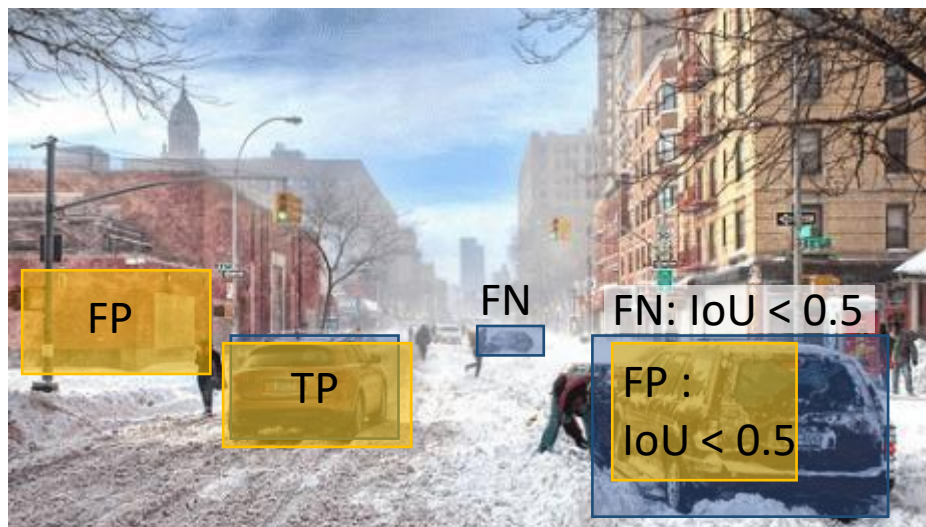


## Appendix



# Object detection (mAP)

- Evaluation of multiple detections
  - Take all **bboxes** with prediction confidence  $> \text{thr}$ 
    - TP =  $\text{IoU} \geq 0.5$  and same class, FP = otherwise
  - Take all bboxes from **ground truth**
    - FN =  $\text{IoU} < 0.5$  with any prediction with the same class





# Object detection (mAP)

- Evaluation of multiple detections
  - Compute precision, recall for each class
    - Varying *thr* value
  - Average Precision (AP):
    - area under precision recall curve

