Segmentation and Detection

CSCI 5299, 4/13/2020

Guest Lecture, Michelina Pallone (Google)

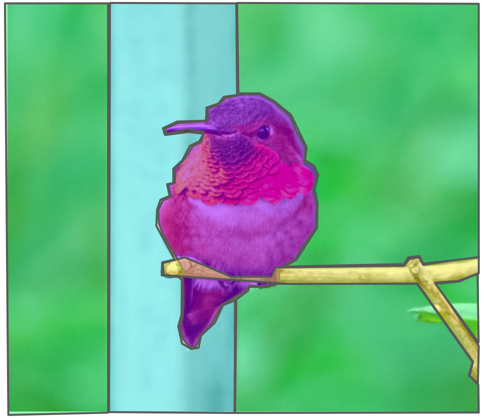
Classification

- AlexNet
- GoogLeNet
- VGG
- ResNet

Scores

- Hummingbird 0.8
- Robin 0.1
- Apple 0.05

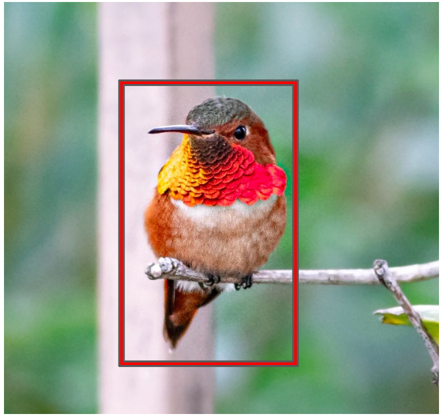
Semantic Segmentation



hummingbird, post, branch, leaves

Pixelwise labels, no objects

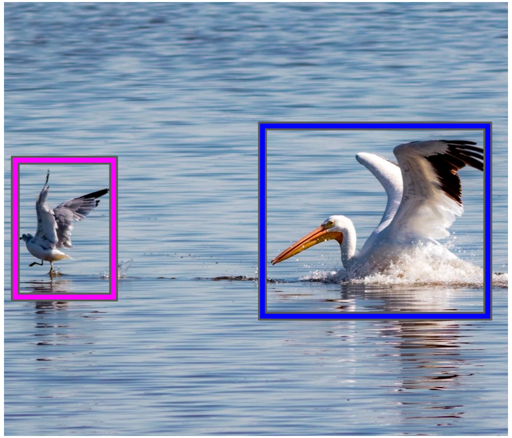
Localization



Hummingbird

Single object

Object Detection

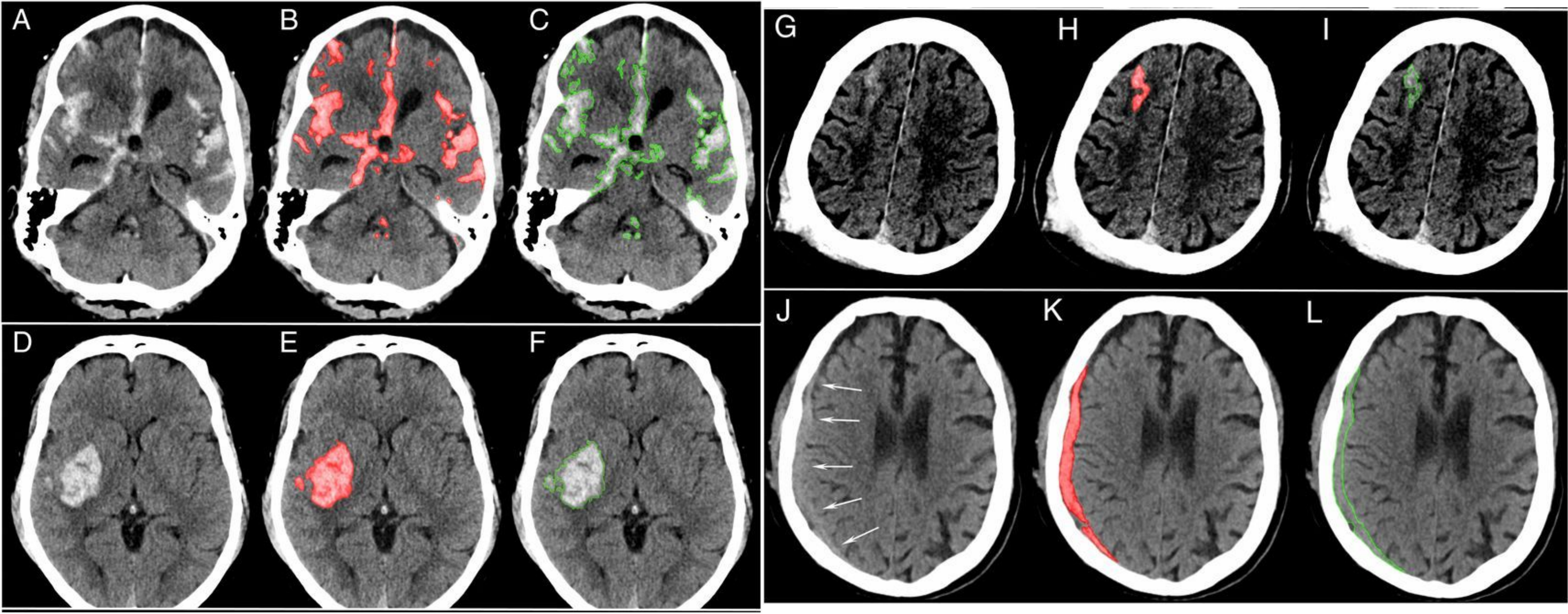


gull, pelican

Multi-object

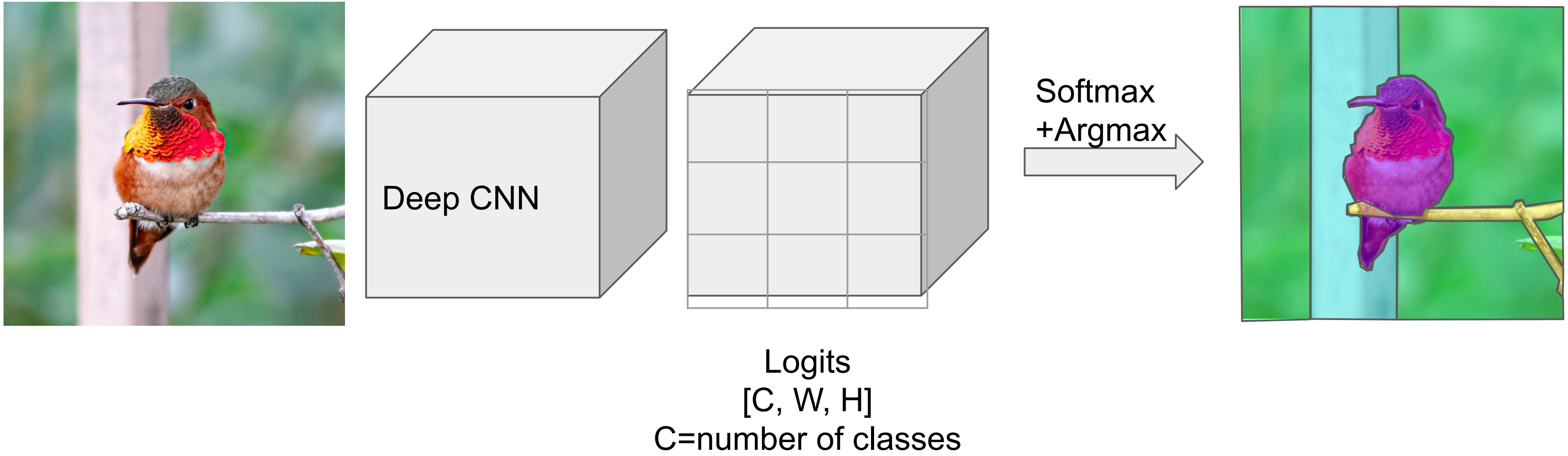
[Building Damage Assessment]

[Cityscapes data set]

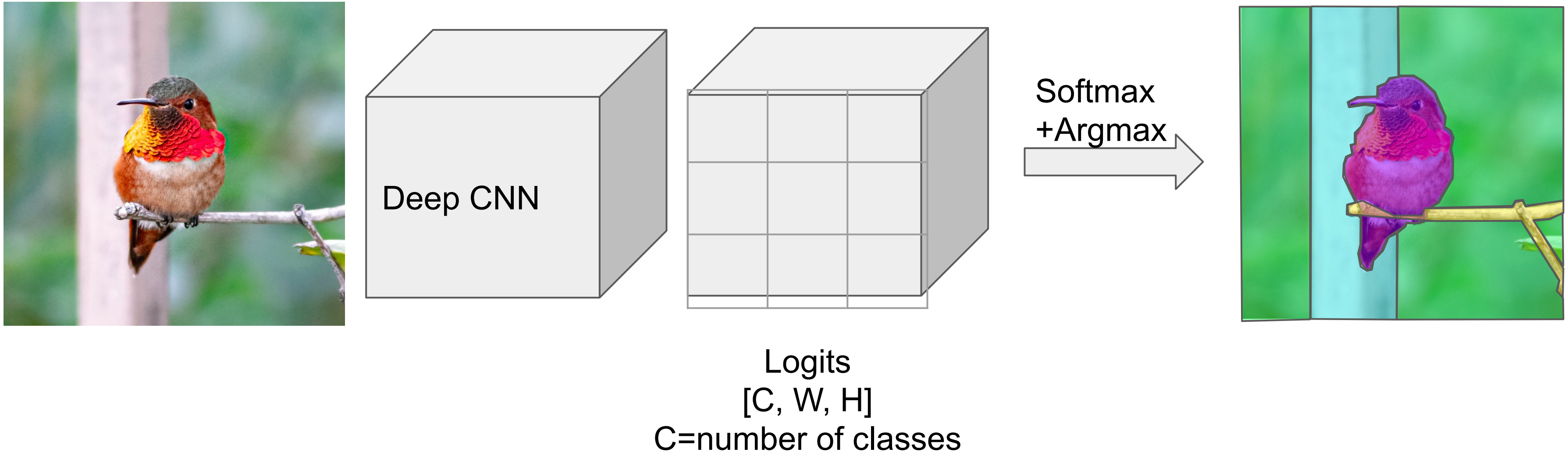


[Expert-level detection of acute intracranial hemorrhage on head computed tomography using deep learning]()

Segmentation



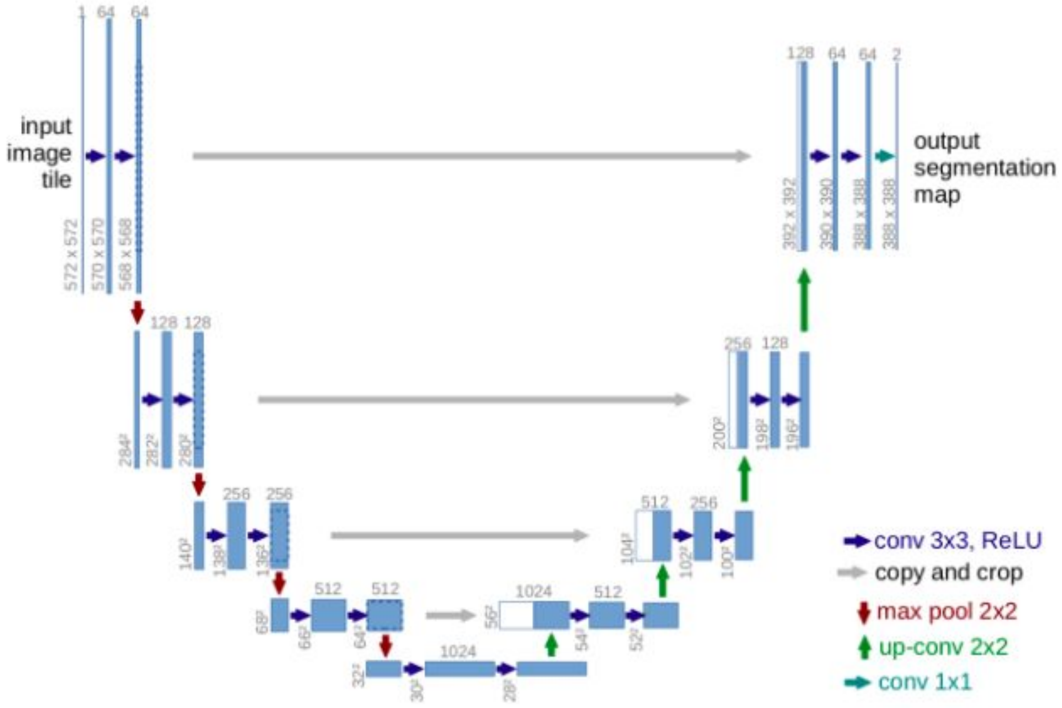
Segmentation



- Deep CNN includes downsampling to reduce computational costs.
  Then upsample back to the original size.

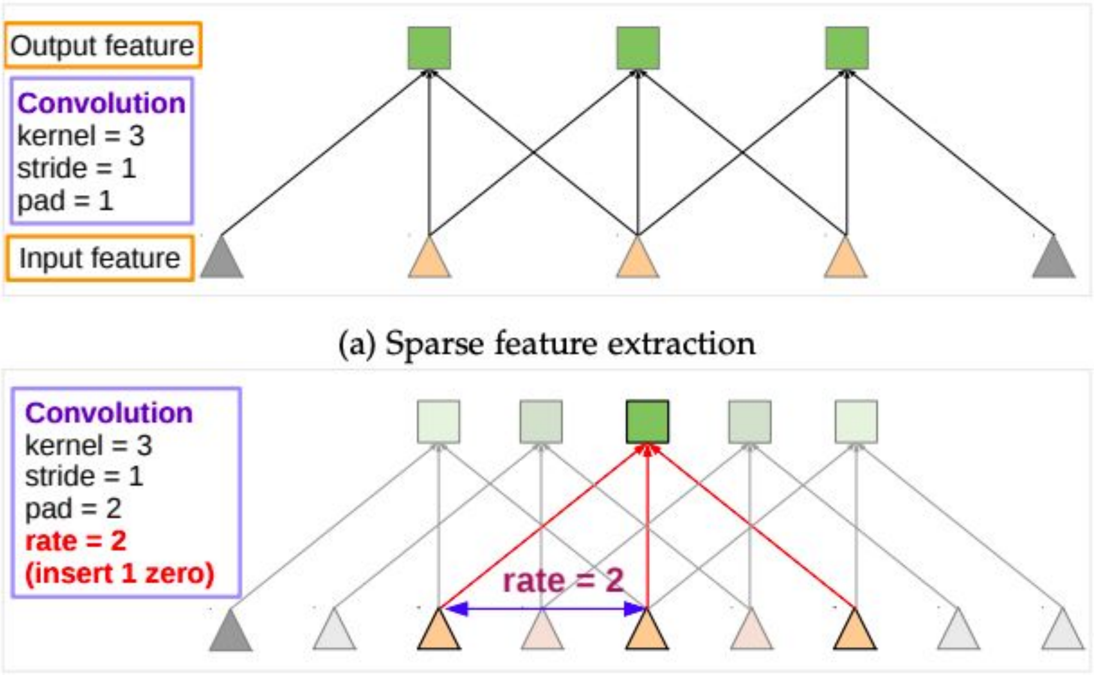
- Sum (or average) cross entropy loss on each pixel in the output

(2015) U-Net



- Influential in medical imaging
- Skip connections all low level features to propagate up to higher level features

(2016) DeepLab

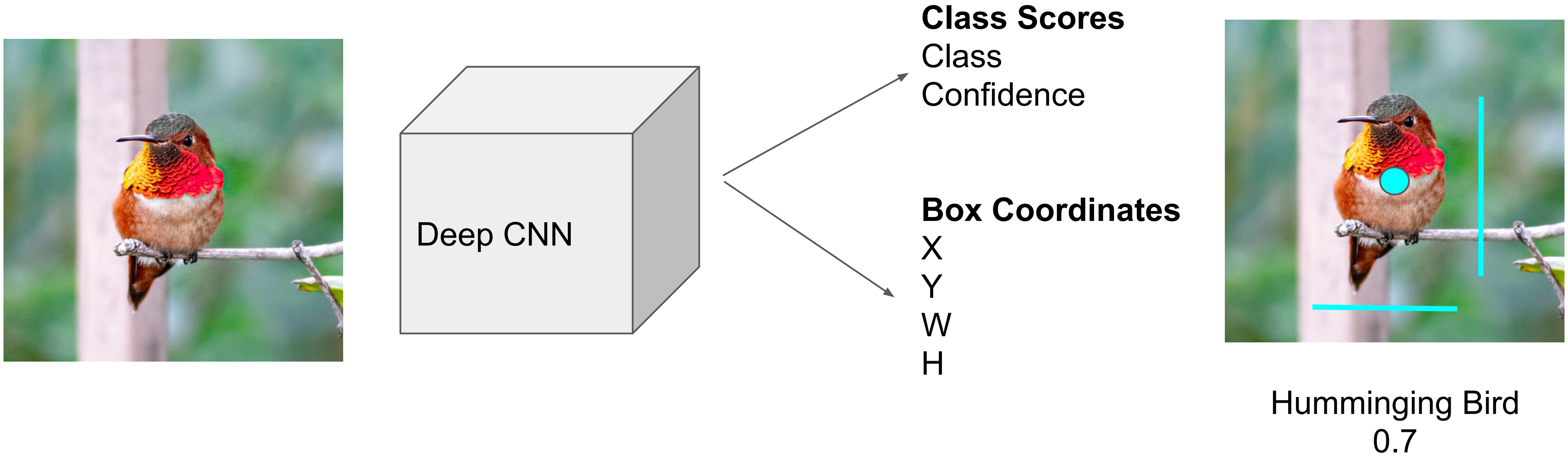


(a) Sparse feature extraction

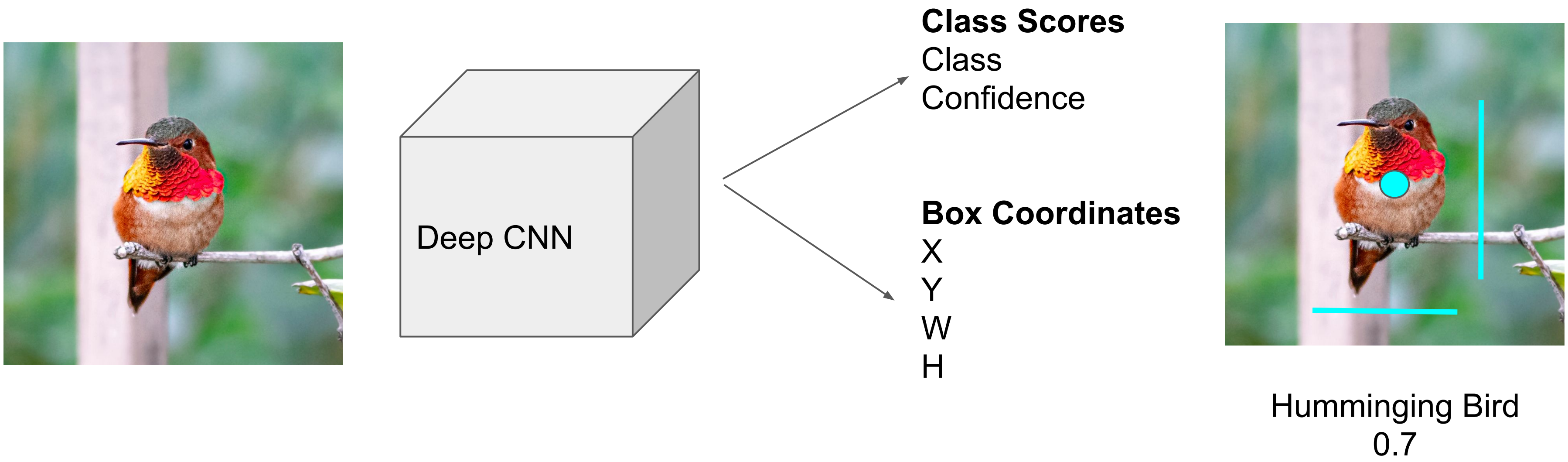
(b) Dense feature extraction

- DeepLab, DeepLab V3
- Atrous (Dilated) Convolutions
- Conditional Random Field (CRF)

Localization

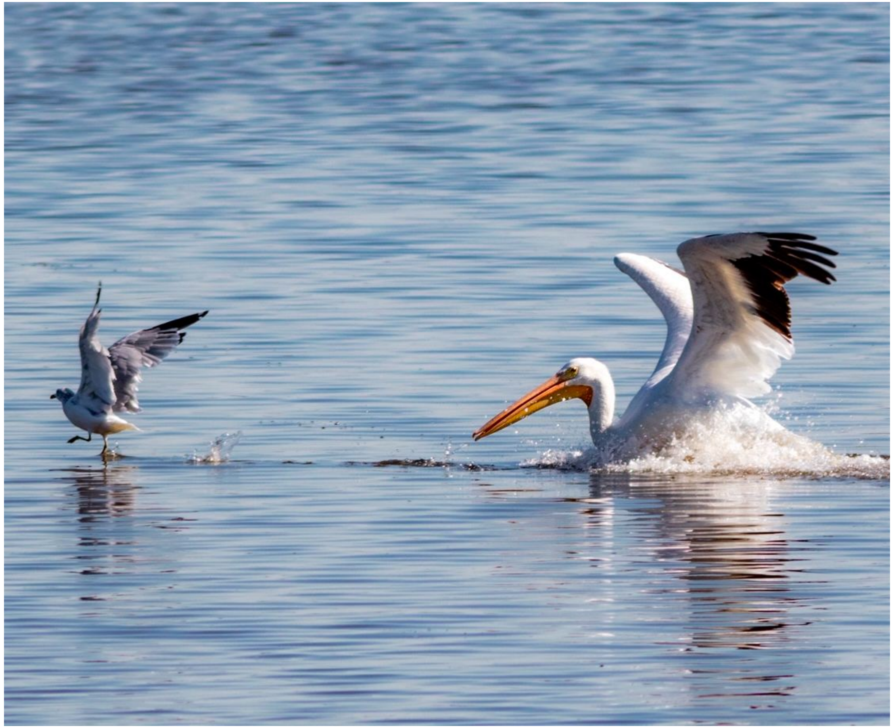


Localization

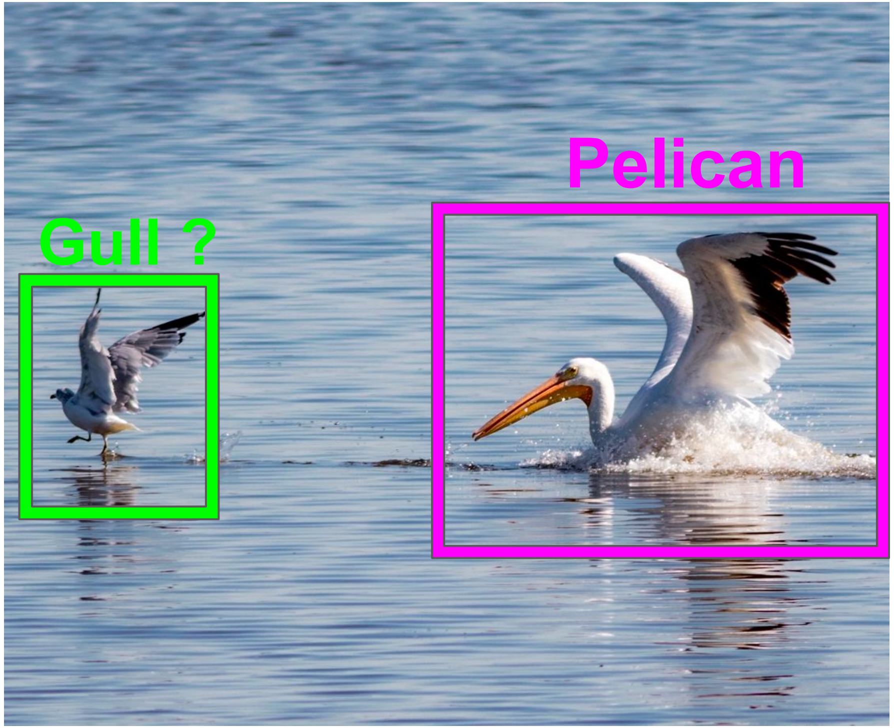


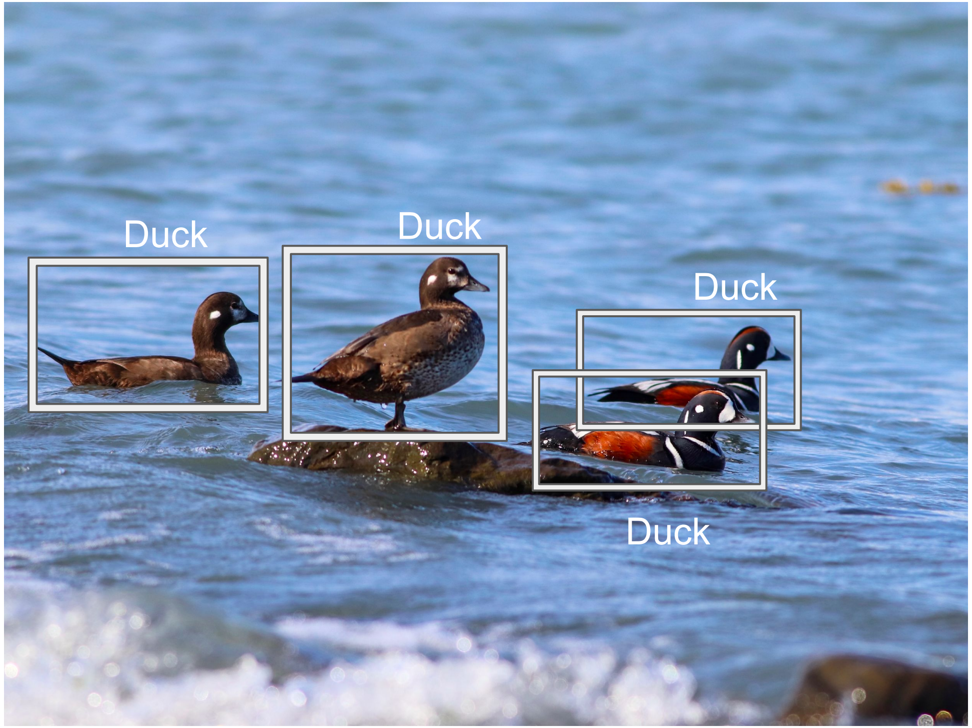
- Two losses: same softmax loss for classes and scores that you would use in image classification, some kind of regression loss for box coordinates.

Detection

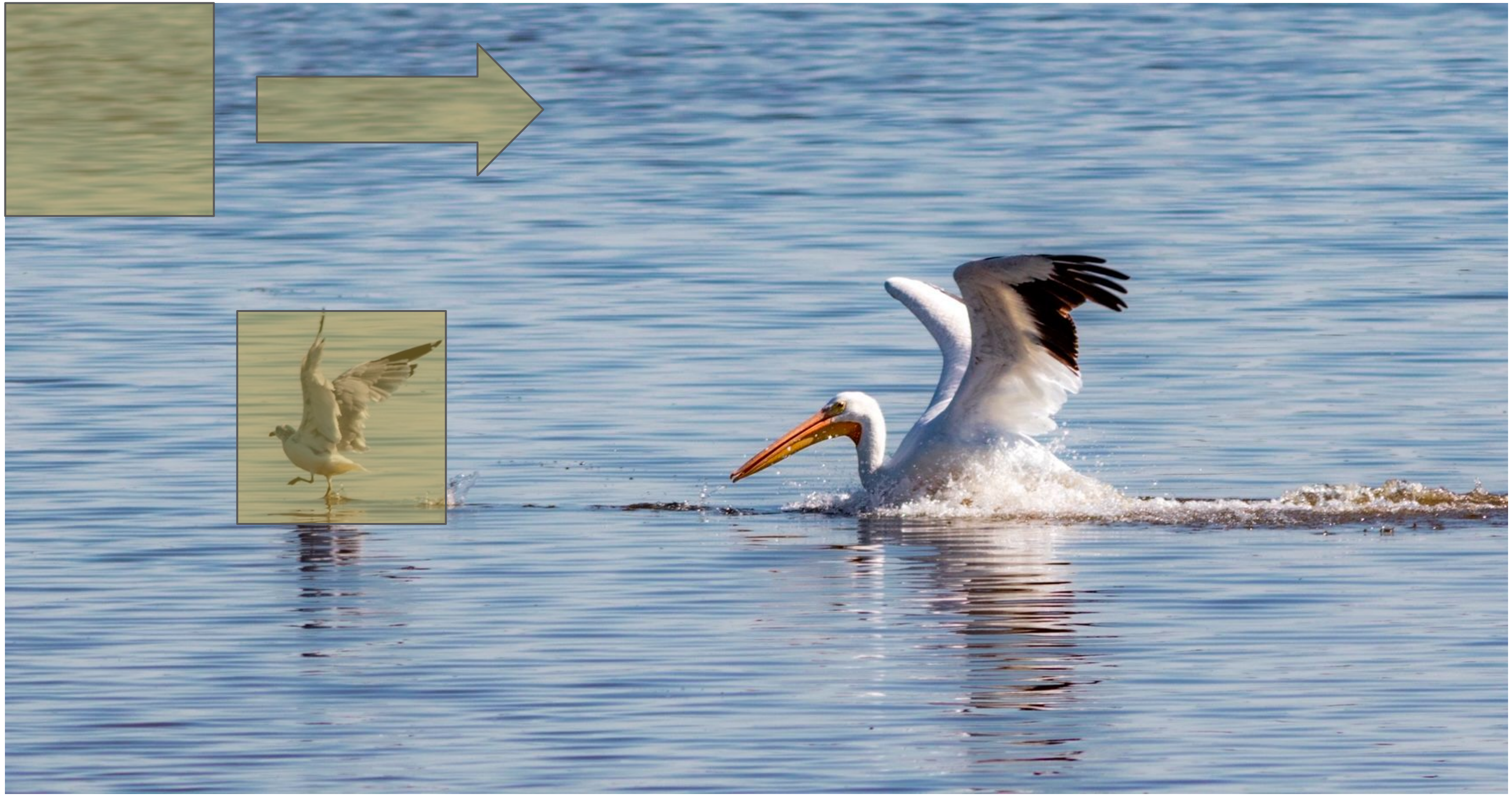


Detection

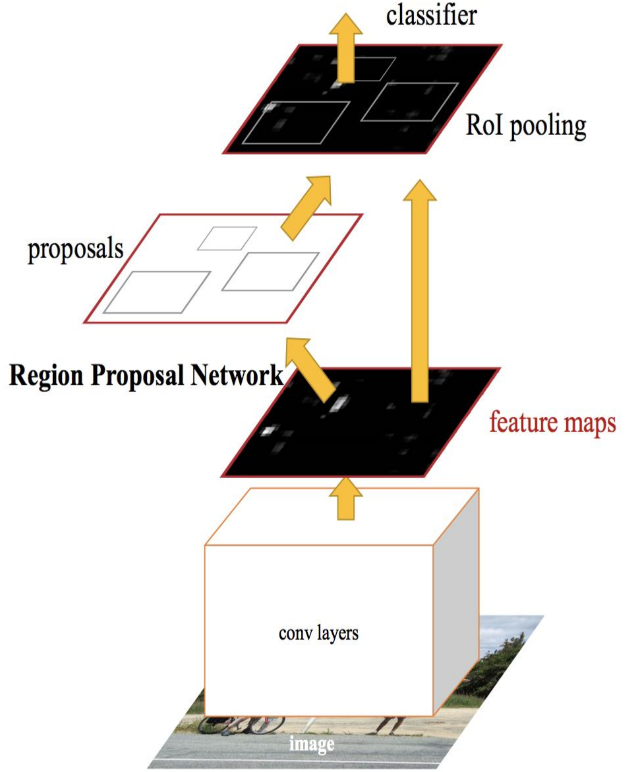


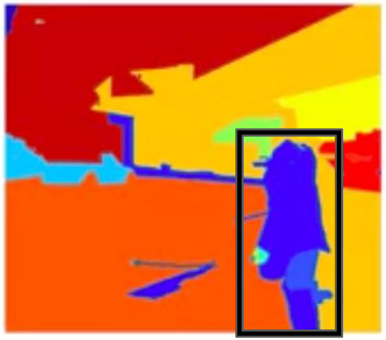


Detection



2 Stage Detection: Regions with CNN (R-CNN)





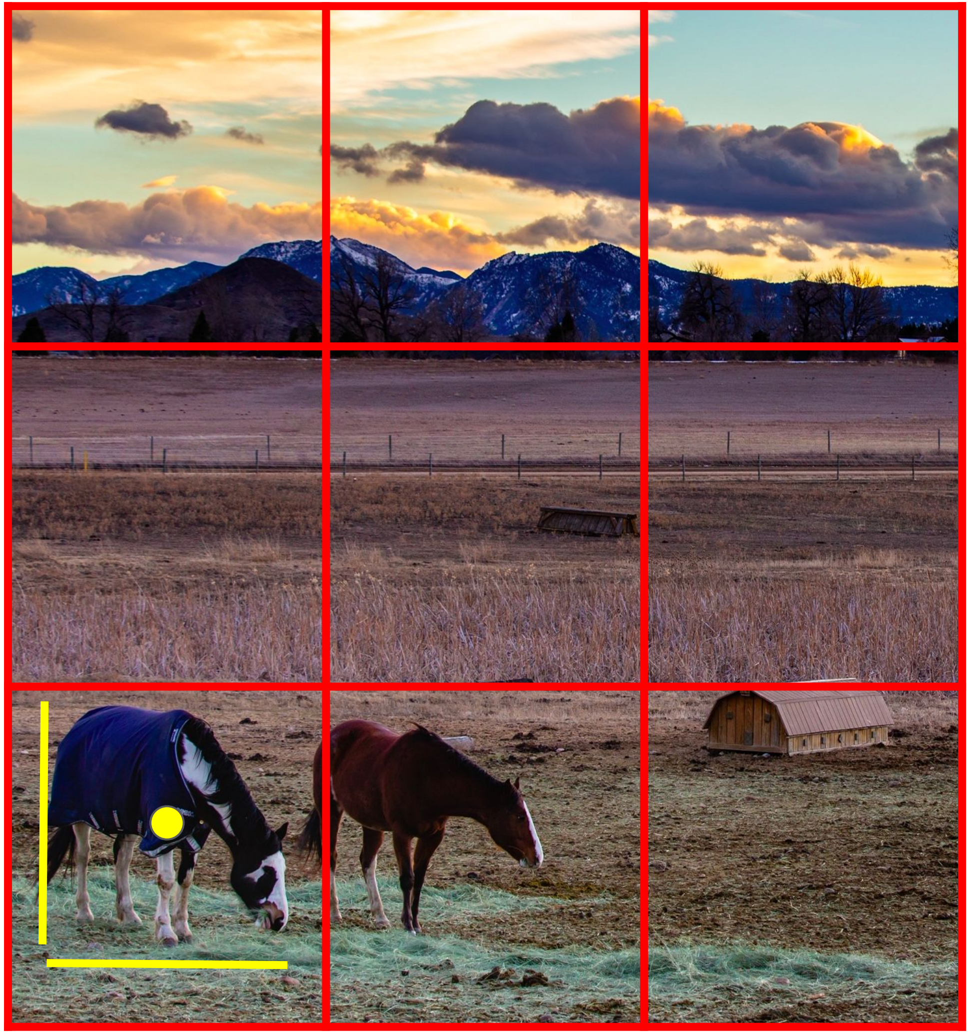
- Narrow down candidate regions via semantic segmentation (Region Proposal Network)
- Train and inference times are slow
  - R-CNN (2014)
  - Fast R-CNN (2015)
  - Faster R-CNN (2015)

One Stage Detection

- YOLO (You only look once 2016)
- SSD (Single Shot Detection 2016)
- Only one pass through the network
  - Simple
  - Fast
- Accuracy is ok

One Stage Detection

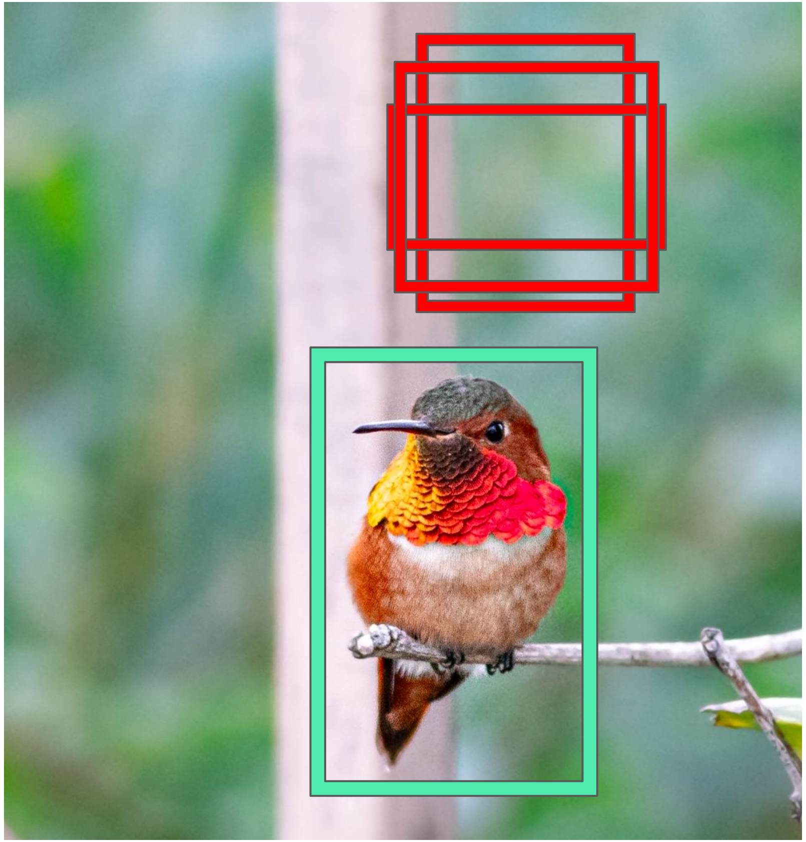
- YOLO (You only look once)
- SSD (Single Shot Detection)
- Only one pass through the network
    - Simple
    - Fast
- Accuracy is ok



One Stage Detection

- YOLO (You only look once)
- SSD (Single Shot Detection)
- Only one pass through the network
  - Simple
  - Fast
- Accuracy is ok

RetinaNet Return of 1 stage



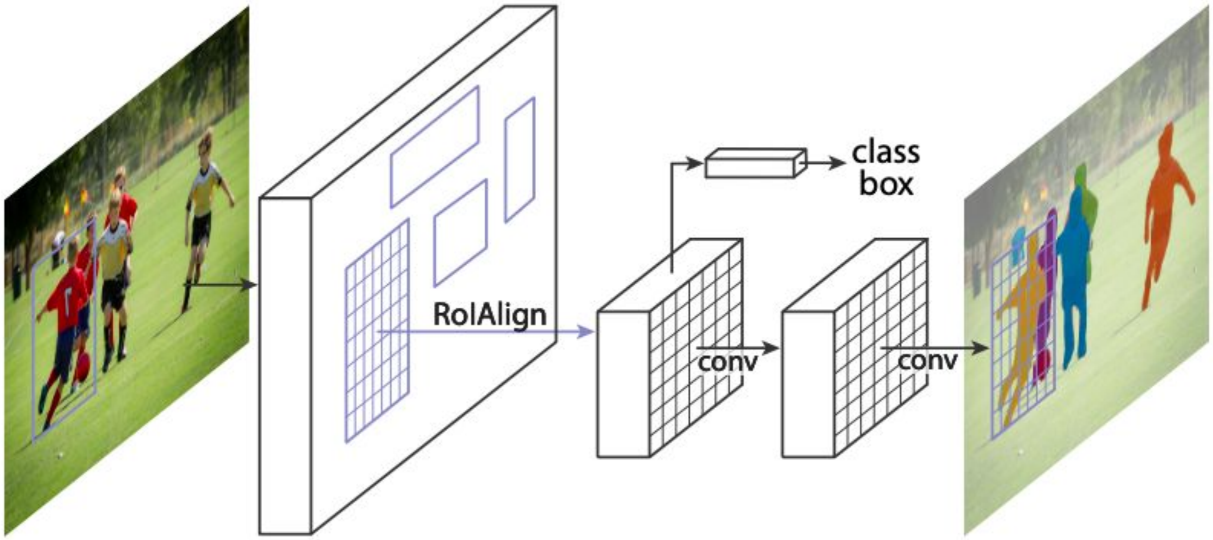
- Traditional problem with one-stage detectors: An extreme class imbalance causes “easy” background samples to overwhelm the loss function

- **Focal Loss**: Keep the loss from easy background samples from overwhelming the loss from sparse hard samples

Instance segmentation

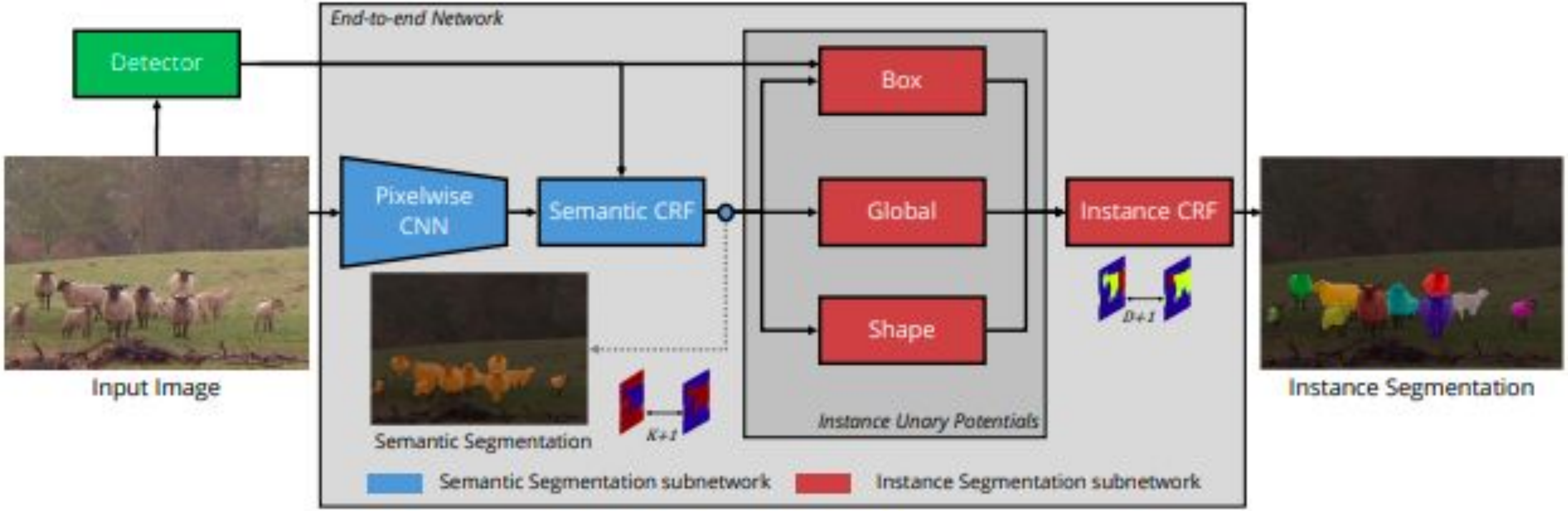
Instance segmentation

Detection-based approaches ("Top Down")



- [Mask R-CNN]
- [ShapeMask: Learning to Segment Novel Objects by Refining Shape Priors]

Grouping-based approaches ("Bottom up")



- Pixelwise Instance Segmentation with a Dynamically Instantiated Network (above)
- Semantic Instance Segmentation with a Discriminative Loss Function
- InstanceCut: from Edges to Instances with MultiCut

Panoptic segmentation

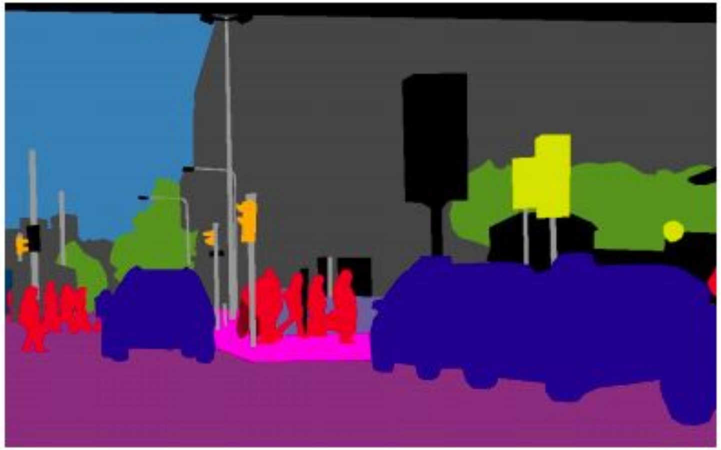
Water?

Rocks?

- **Things**: Objects that come in discrete, countable instances
    - Well suited to instance segmentation
    - Examples: Cars, people, animals

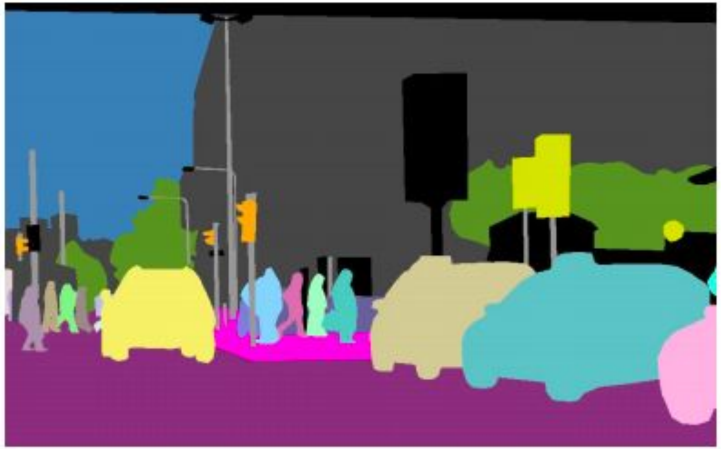
- **Stuff**: Amorphous background regions
    - Better suited to semantic segmentation
    - Examples: Sky, grass, water

(a) image



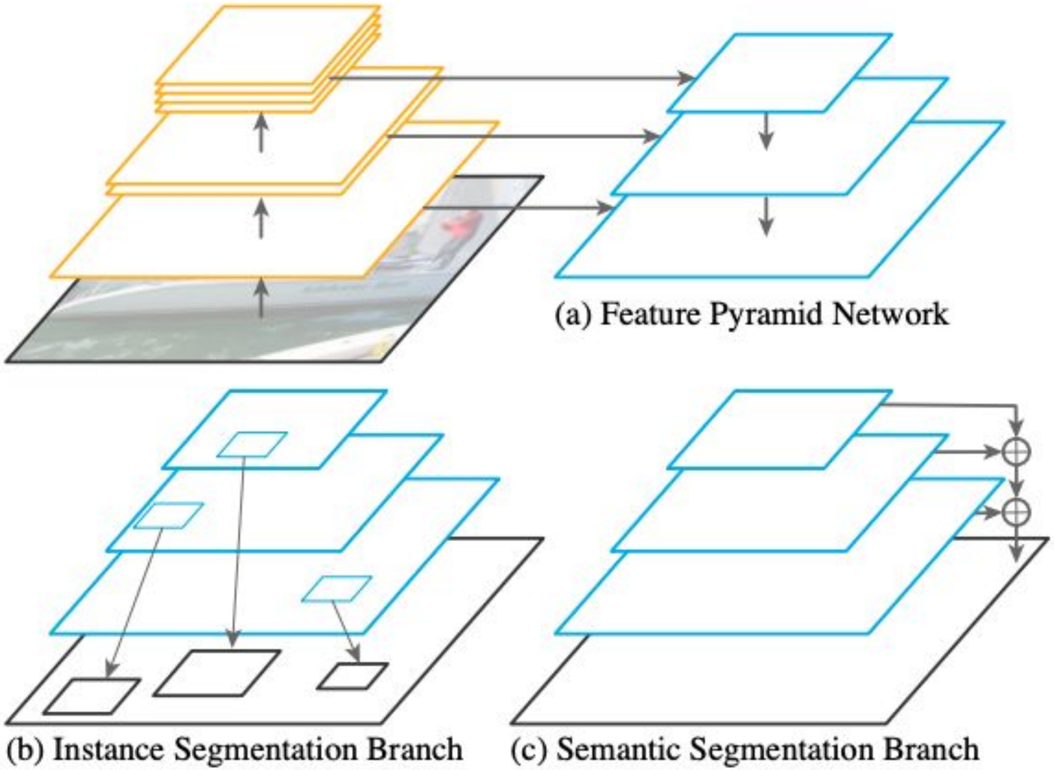
(b) semantic segmentation

(c) instance segmentation



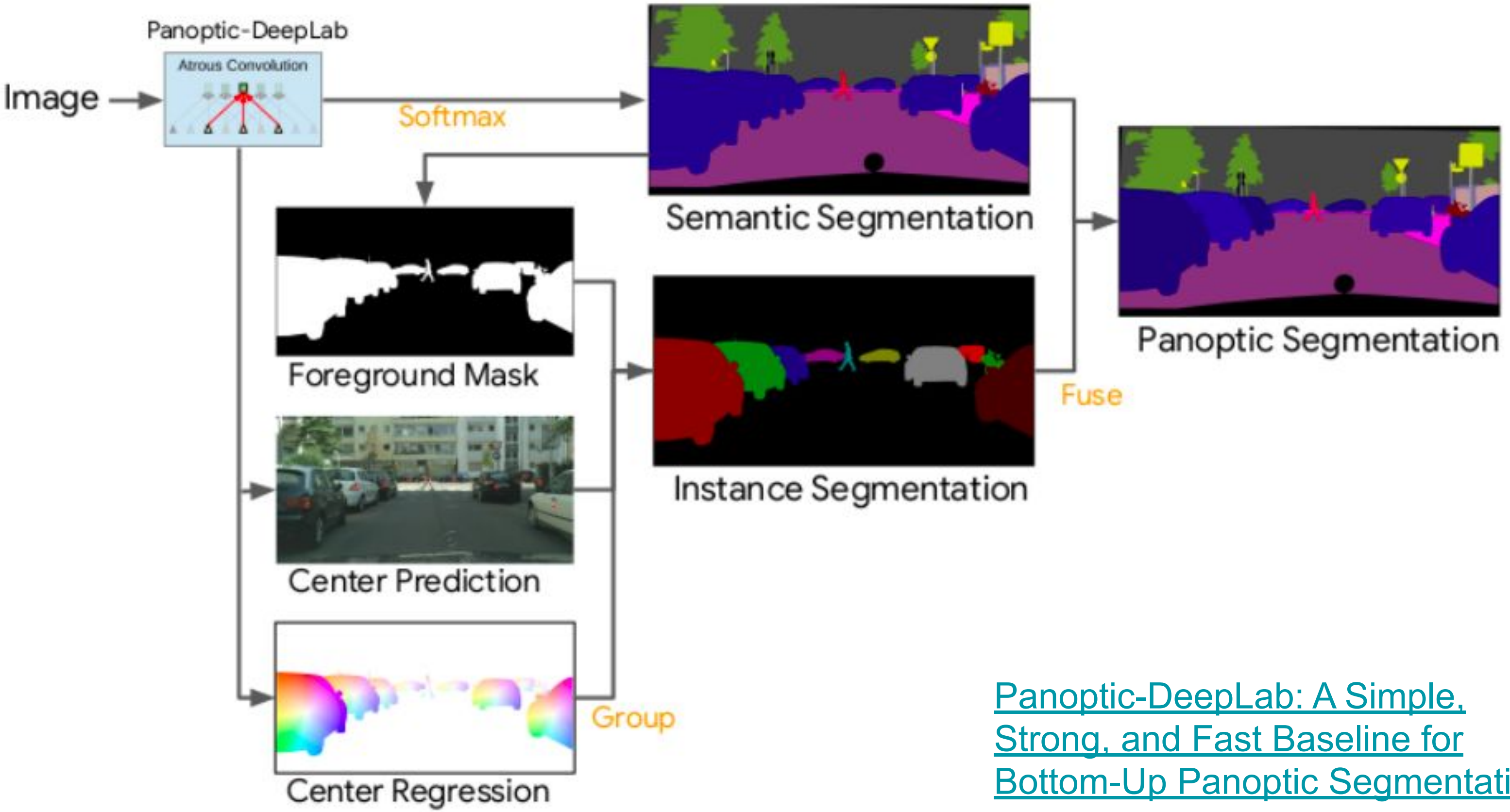
(d) panoptic segmentation

Top Down



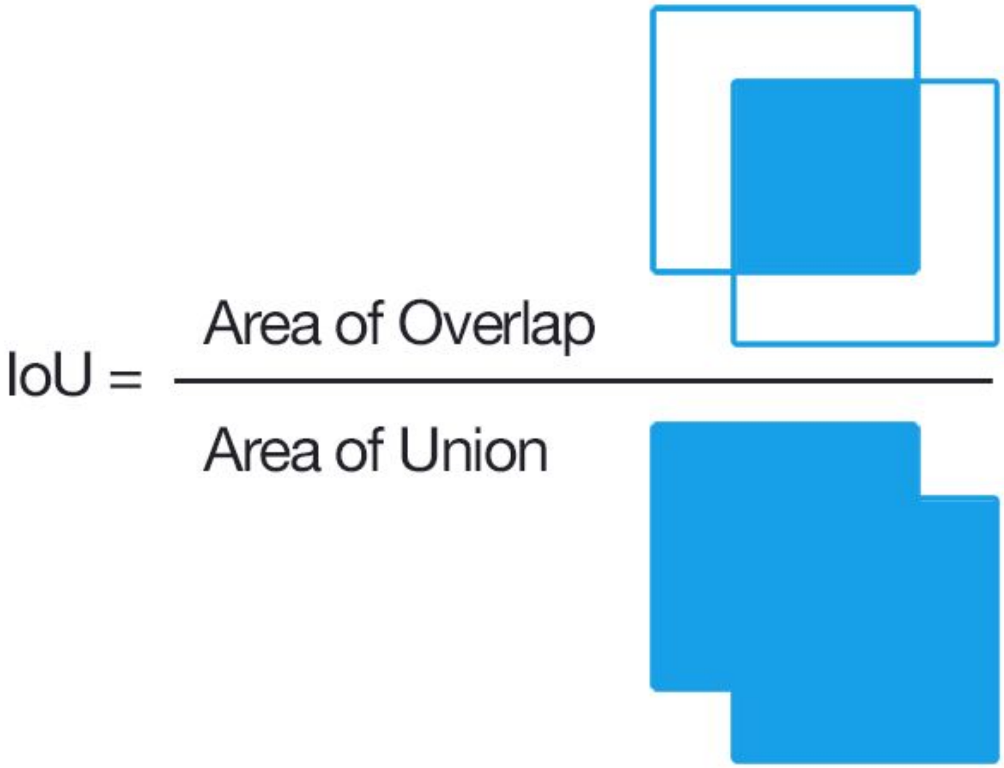
[Panoptic Feature Pyramid Networks]

Bottom Up



[Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation]

Intersection over Union



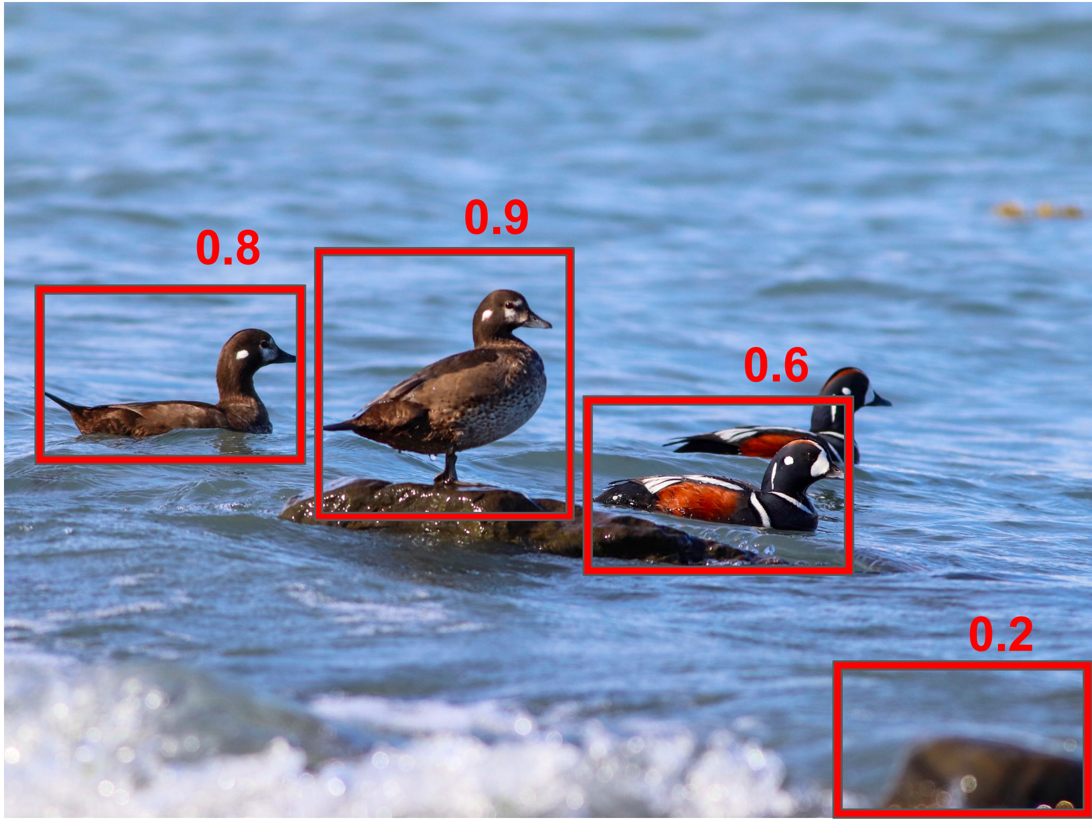
- For **stuff** IoU is a standard evaluation metric
- For **things,** you decide what IoU is a true positive (IoU = 0.5 is common)

Predicted Values	Actual Values: Positive	Actual Values: Negative
Positive	True Positive (TP)	False Positive (FP, Type I Error)
Negative	False Negative (FN, Type II Error)	True Negative (TN)

$$\text{Precision} = \frac{\#\,TP}{\#\,TP + \#\,FP}$$

With confidence threshold of 0:

PR = 3 / (3 + 1) = 0.75

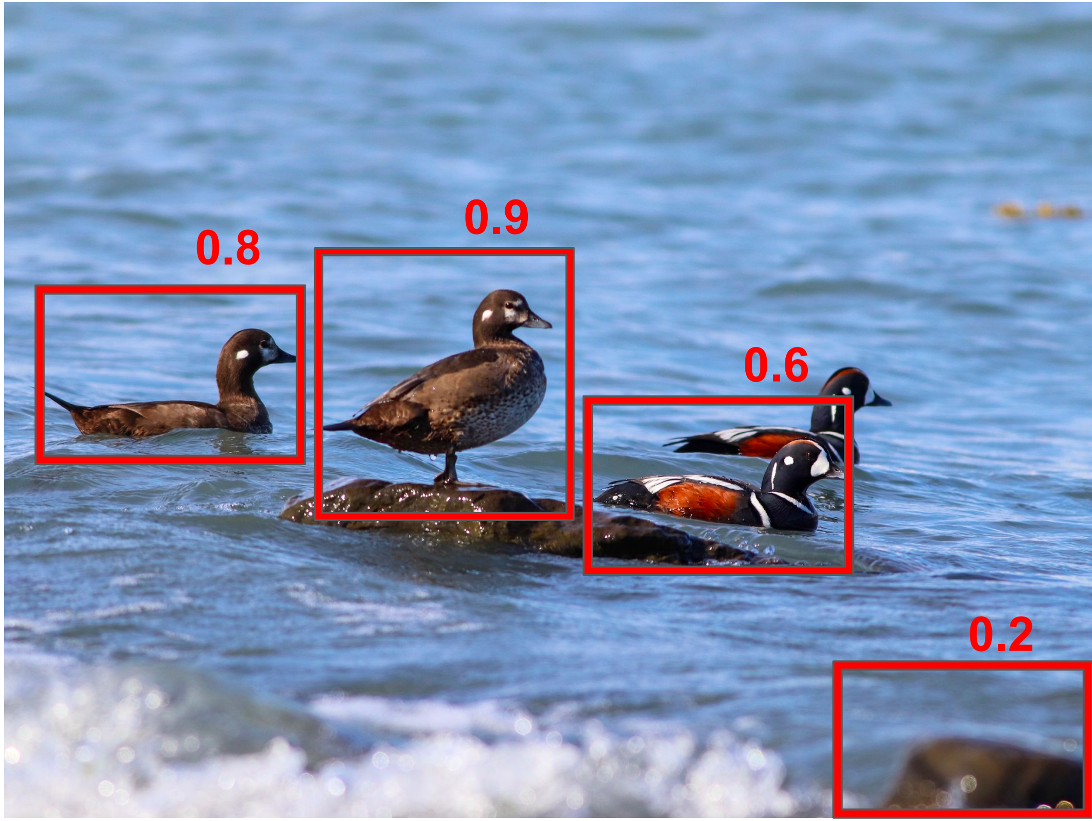


Model predictions with confidences

$$\text{Precision} = \frac{\#\,TP}{\#\,TP + \#\,FP}$$

With confidence threshold of 0.5:

PR = 3 / (3 + 0) = 1

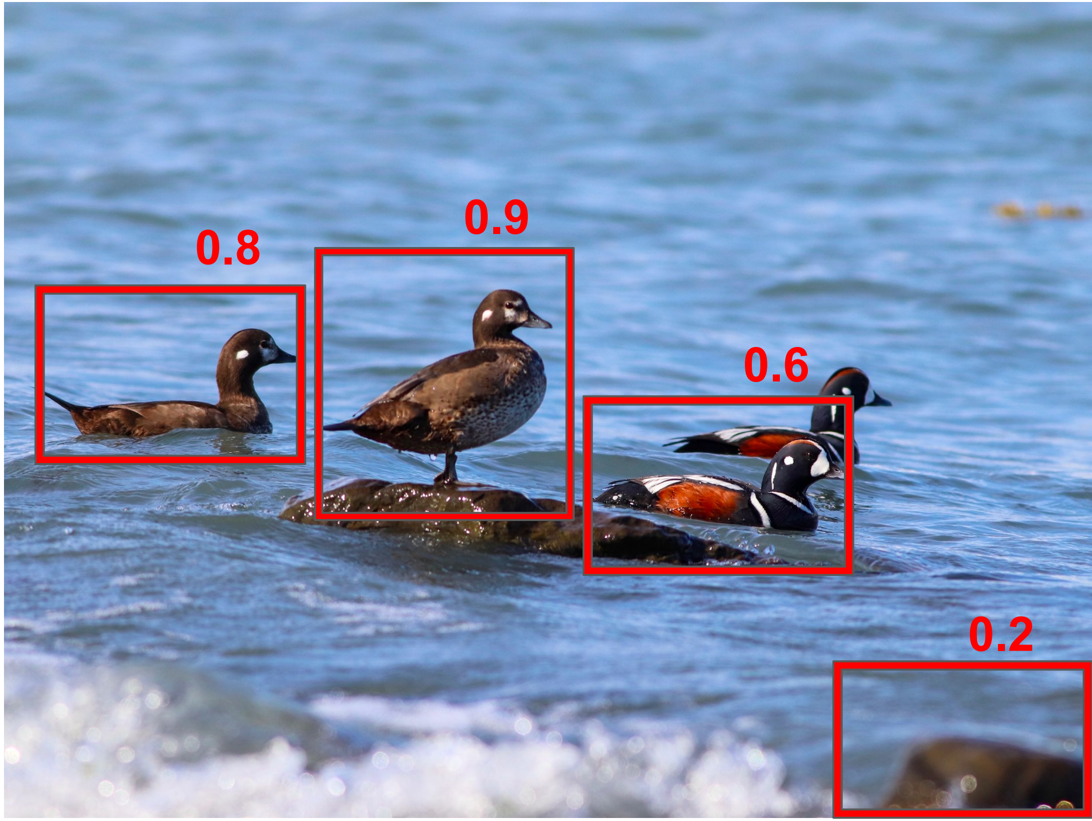


Model predictions with confidences

$$\text{Recall} = \frac{\#\,TP}{\#\,TP + \#\,FN}$$

With confidence threshold of 0:

PR = 3 / (3 + 1) = 0.75

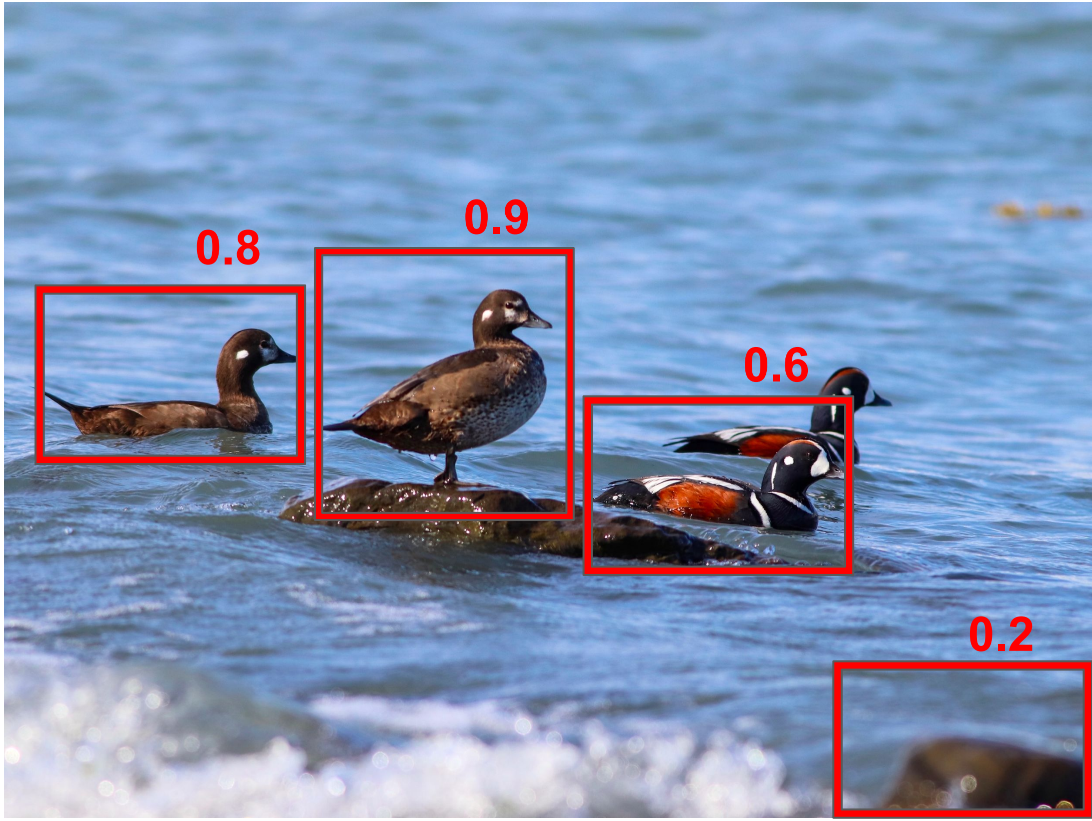


Model predictions with confidences

$$\text{Recall} = \frac{\#\,TP}{\#\,TP + \#\,FN}$$

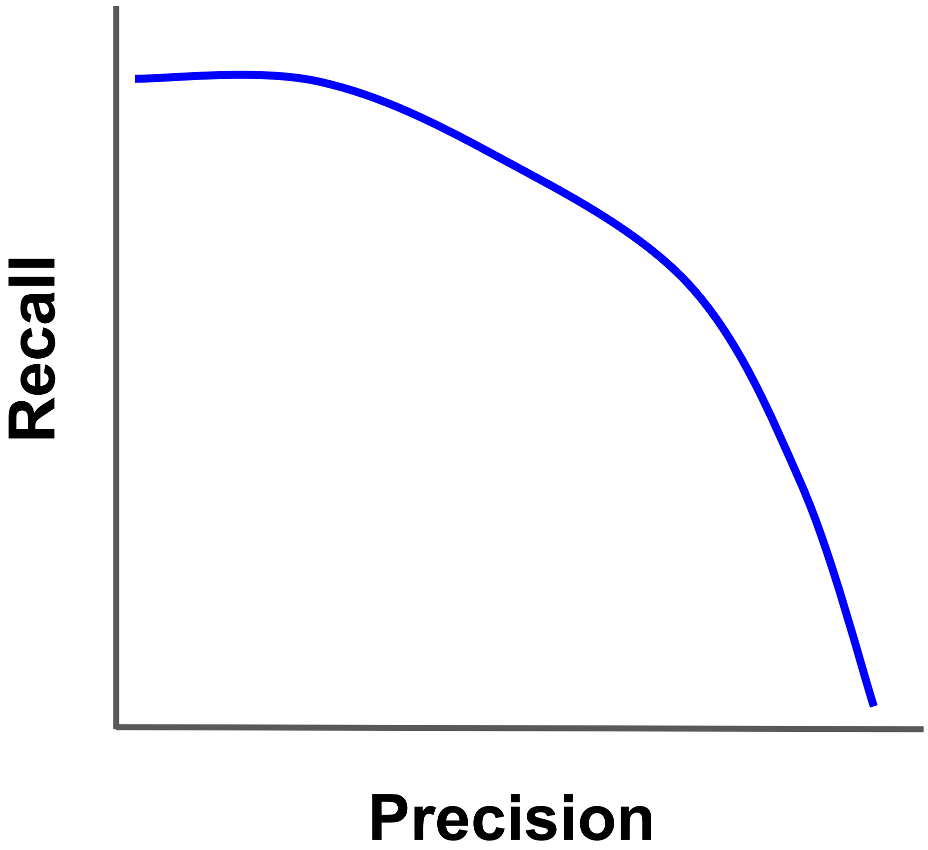
With confidence threshold of 0.7:

PR = 2 / (2 + 2) = 0.5



Model predictions with confidences

Precision Recall Curves



- Plot precision vs recall at all the confidence thresholds
- Average Precision (AP) is the area under the curve
- Mean Average Precision (mAP) is the mean of all the APs for each class
- Similarly, you will see mean IOU (mIOU) for semantic segmentation metrics

COCO
Common Objects in Context

info@cocodataset.org

Home People Dataset Tasks Evaluate

Detection Leaderboard

BBOX: Dev Standard15 Chal15 Chal16 Chal17

SEGM: Dev Standard15 Chal15 Chal16 Chal17 Chal18

Copy to Clipboard Export to CSV

Search:

	AP	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}	AR^{1}	AR^{10}	AR^{100}	AR^{S}	AR^{M}	AR^{L}	date
Megvii (Face++)	0.526	0.730	0.585	0.343	0.556	0.660	0.391	0.645	0.689	0.513	0.727	0.827	2017-10-05
CM-CV AR	0.525	0.717	0.578	0.352	0.550	0.642	0.392	0.647	0.689	0.528	0.720	0.821	2019-07-26
Night owl	0.519	0.704	0.570	0.342	0.548	0.647	0.391	0.640	0.680	0.498	0.713	0.824	2019-08-05
Alibaba Turing Lab	0.514	0.694	0.563	0.336	0.540	0.639	0.388	0.638	0.679	0.499	0.711	0.818	2019-08-05
UCenter	0.510	0.705	0.558	0.326	0.539	0.648	0.392	0.640	0.678	0.497	0.720	0.829	2017-10-05
MSRA	0.507	0.717	0.566	0.343	0.529	0.627	0.379	0.638	0.690	0.524	0.720	0.824	2017-10-05
DL-61	0.507	0.708	0.569	0.337	0.534	0.626	0.383	0.639	0.686	0.516	0.718	0.821	2018-08-15
FAIR Mask R-CNN	0.503	0.720	0.558	0.328	0.537	0.627	0.380	0.622	0.659	0.485	0.704	0.800	2017-10-05

Public Benchmarks

- COCO
- Cityscapes
- Mapillary Vistas

Evaluating models for real world applications

- Mapping:
  - False positive: Add a non-existent building to the map
  - False negative: Missing a building on the map
- Autonomous Vehicles:
  - False positive: Vehicle detects a non-existent stop sign, stops, and gets rear ended
  - False negative: Vehicle drives through a stop sign and causes an accident
- Medicine:
  - False positive: Unnecessary procedures -> higher healthcare costs, strain health care system
  - False negative: A serious disease goes untreated


N 95th St
N 95th St
N 95th St
N 95th St

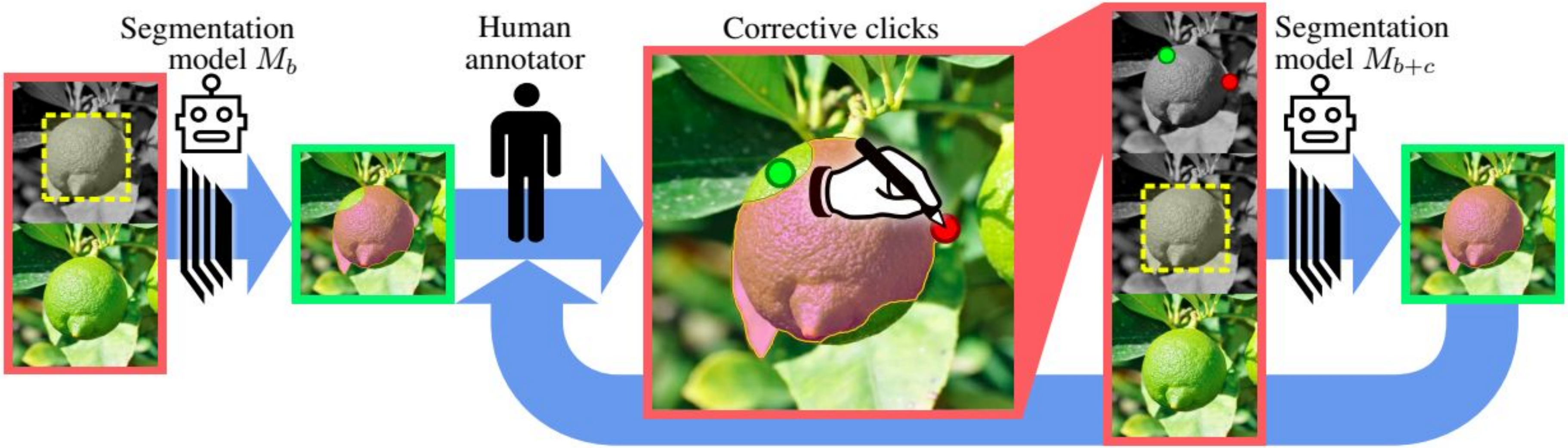

N 95th St

N 95th St


North 95th Street

North 95th Street

Resource Constraints: Data and Labeling



- Large-scale interactive object segmentation with human annotators
- Interactive Full Image Segmentation by Considering All Regions Jointly

Resource Constraints: Computation

- Autonomous vehicles need to do inference on device
  - MobileNet
- Very deep backbones (e.g. ResNet) are expensive at train and inference time
  - Start with a pre-trained backbone
- High training cost to search for your architecture
- Bigger data sets yield better results but require disk space