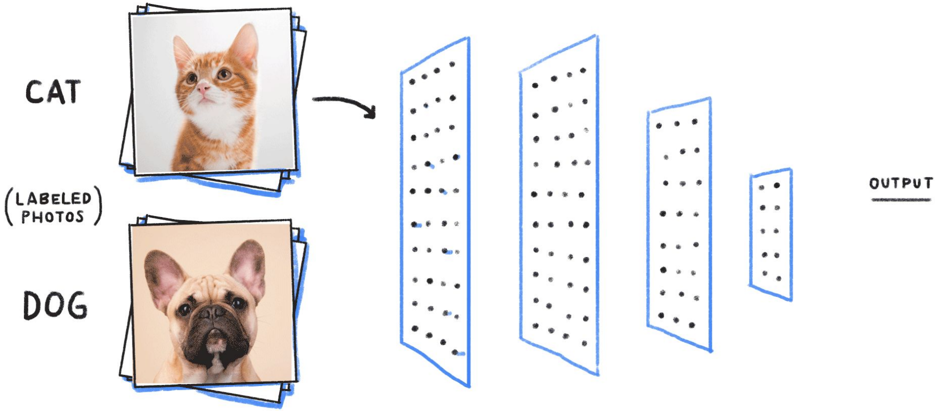
Object Detection

JunYoung Gwak

Motivation

Image classification

- Input: Image
- Output: object class



Motivation

Limitation of classification

- Multiple classes
- Location

i.e.

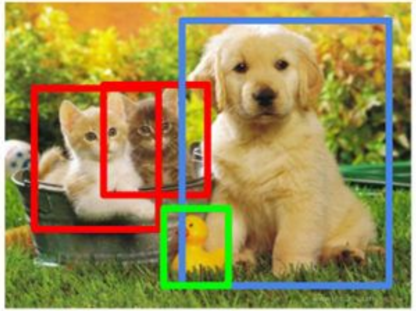
Object classification assumes

- Single class of object
- Occupies majority of the input image

Classification

CAT

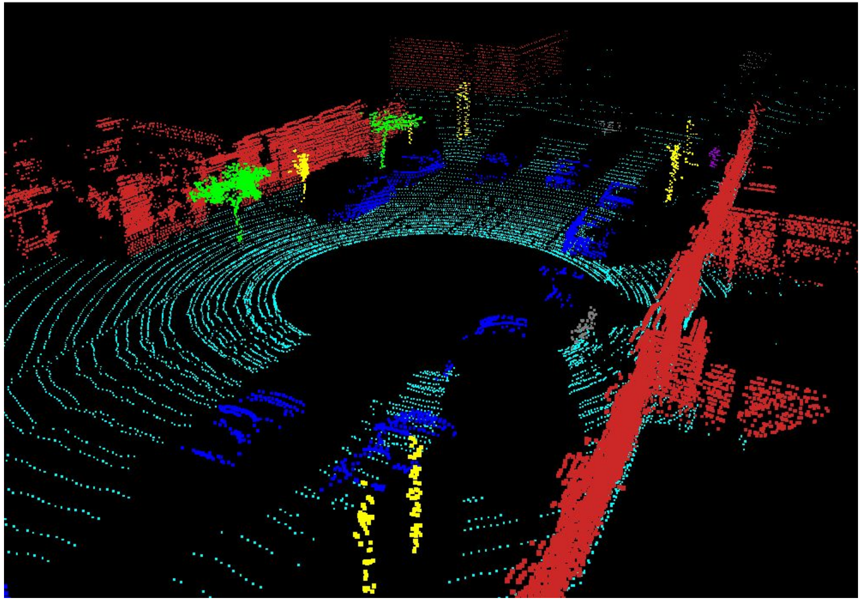
Object Detection

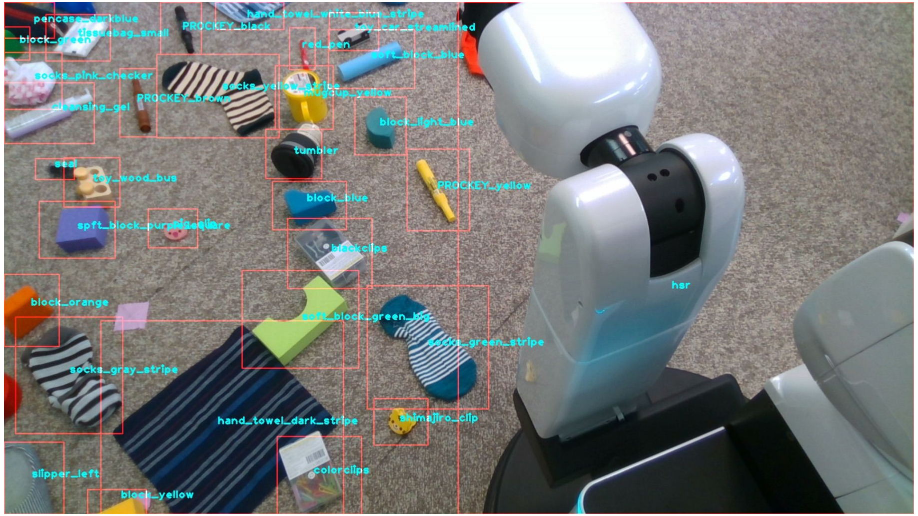


CAT, DOG, DUCK

Motivation

We need high-level understanding of the complex world

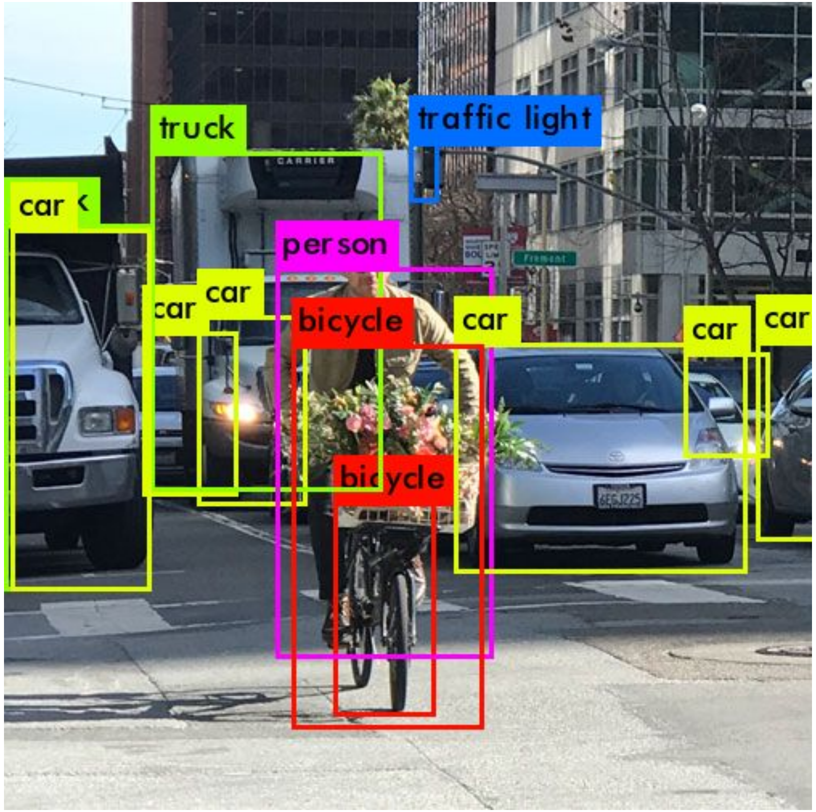




Problem Definition

Object Detection

- Input: Image
- Output: multiple instances of
  - object location (bounding box)
  - object class



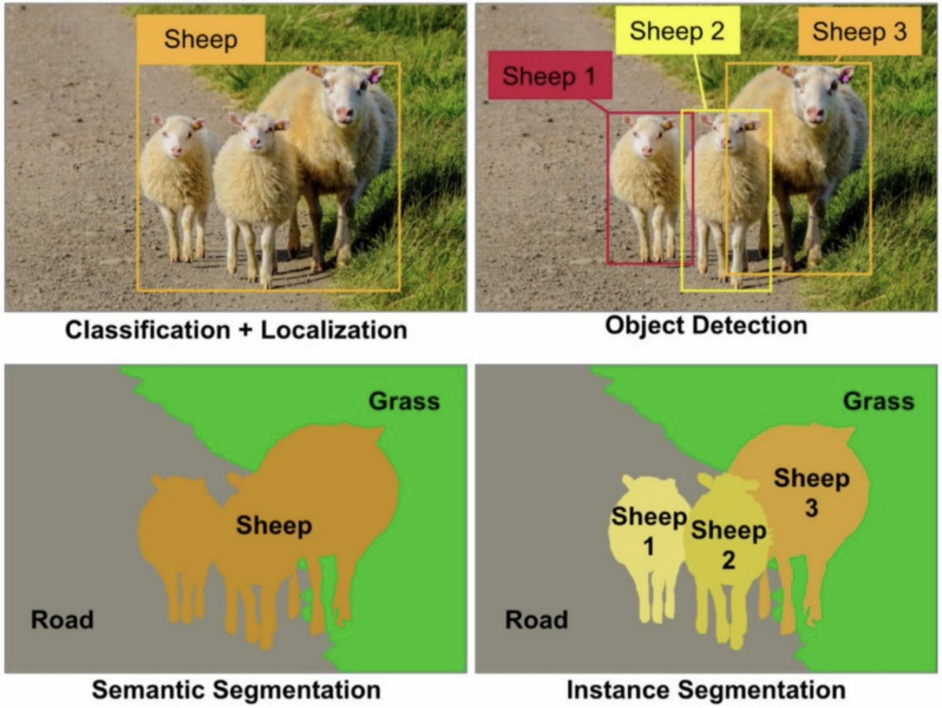
Problem Definition

Object Detection

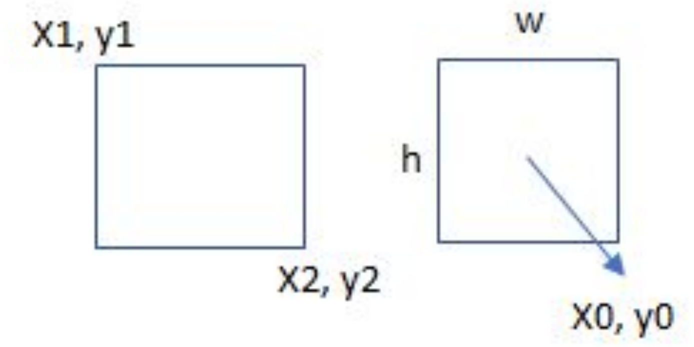
- Input: Image
- Output: multiple **instances** of
  - object location (bounding box)
  - object class

Instance:

- Distinguishes individual objects, in contrast to considering them as a same single semantic class

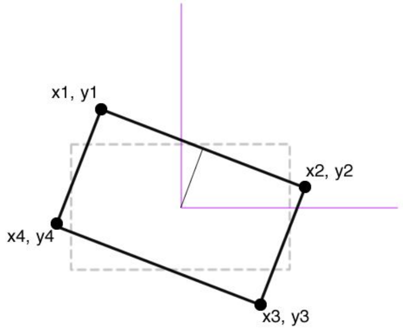


Problem Definition



Object Detection

- Input: Image
- Output: multiple instances of
  - **object location (bounding box)**
  - object class



Bounding box:

- Rigid box that confines the instance
- Multiple possible parameterizations
  - (width, height, center x, center y)
  - (x1, y1, x2, y2)
  - (x1, y1, x2, y2, rotation)

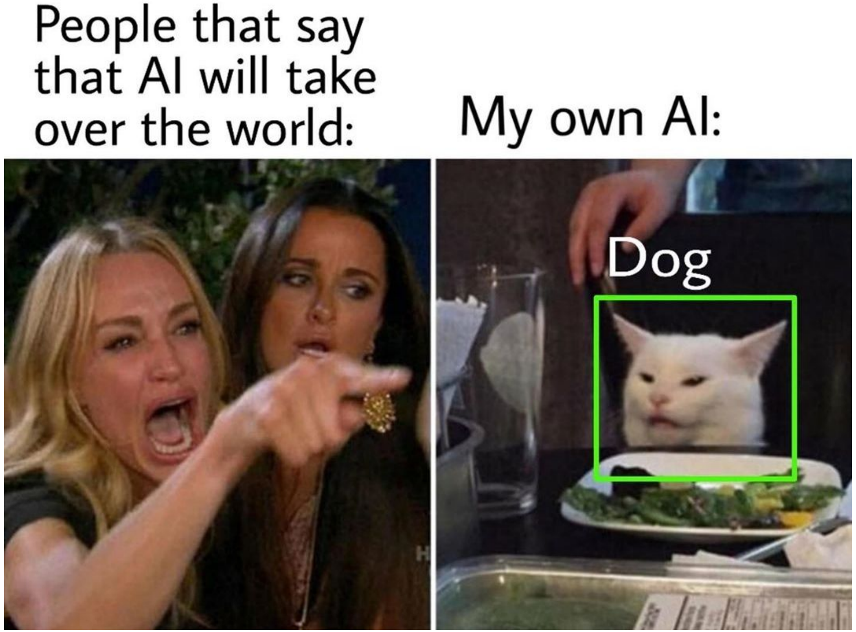
Problem Definition

Object Detection

- Input: Image
- Output: multiple instances of
  - object location (bounding box)
  - **object class**

Object class:

- Semantic class of the instance
  - Similar to object classification task, by predicting a vector of scores



Modern Object Detection Architecture (as of 2017)

- Multiple important works around 2014-2017 which built the basis of modern object detection architecture
    - R-CNN
    - Fast R-CNN
    - Faster R-CNN
    - SSD
    - YOLO (v2, v3)
    - FPN
    - Fully convolutional
    - ...

	YOLO								YOLOv2
batch norm?		✓	✓	✓	✓	✓	✓	✓	✓
hi-res classifier?			✓	✓	✓	✓	✓	✓	✓
convolutional?				✓	✓	✓	✓	✓	✓
anchor boxes?				✓	✓				
new network?					✓	✓	✓	✓	✓
dimension priors?						✓	✓	✓	✓
location prediction?						✓	✓	✓	✓
passthrough?							✓	✓	✓
multi-scale?								✓	✓
hi-res detector?									✓
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	**78.6**

Let’s dissect the modern (2017) object detection architecture!
⇒ Detectron

Modern Object Detection Architecture (as of 2017)

Stage 1

- For every output pixel (given by backbone networks)
  - For every anchor boxes
    - Predict bounding box offsets
    - Predict anchor confidence
- Suppress overlapping predictions using non-maximum suppression

(Optional, if two-stage networks) Stage 2

- For every region proposals
  - Predict bounding box offsets
  - Predict its semantic class

Modern Object Detection Architecture (as of 2017)

Stage 1

- **For every output pixel** (given by backbone networks)
  - For every anchor boxes
    - Predict bounding box offsets
    - Predict anchor confidence
- Suppress overlapping predictions using non-maximum suppression

(Optional, if two-stage networks) Stage 2

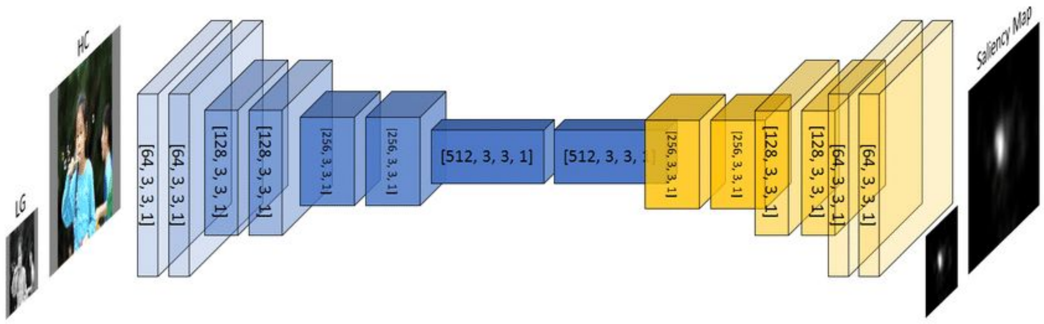
- For every region proposals
  - Predict bounding box offsets
  - Predict its semantic class

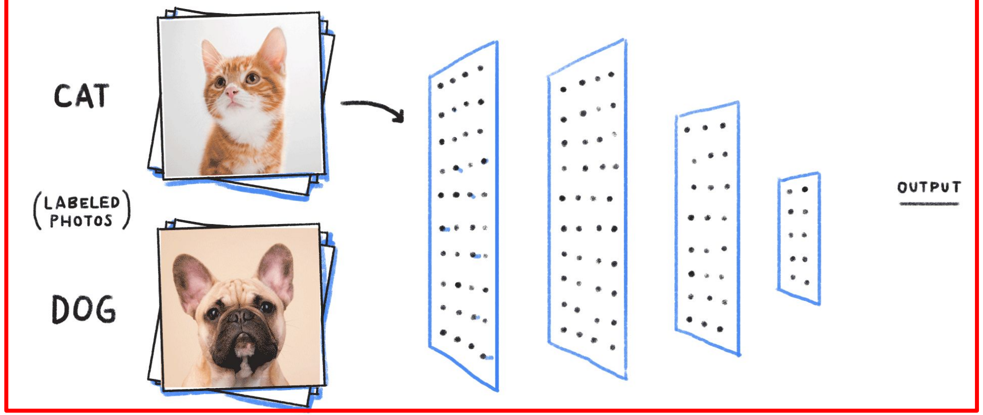
Modern Object Detection Architecture (as of 2017)

Fully Convolutional

Every pixel makes prediction!

- In contrast to previous works in image classification





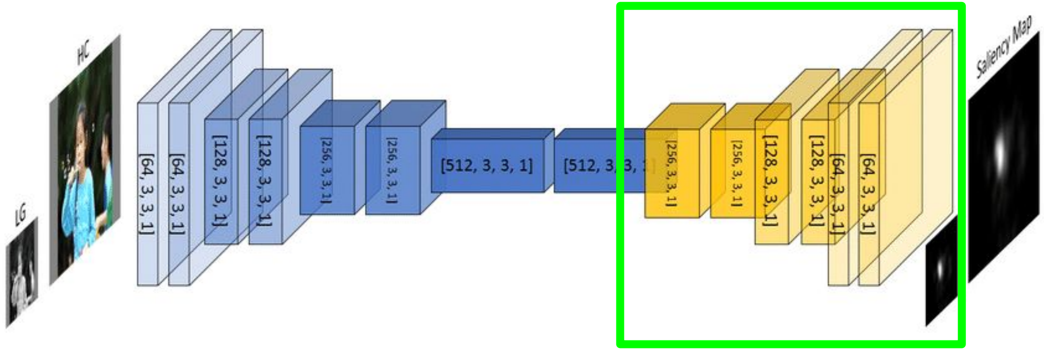
Modern Object Detection Architecture (as of 2017)

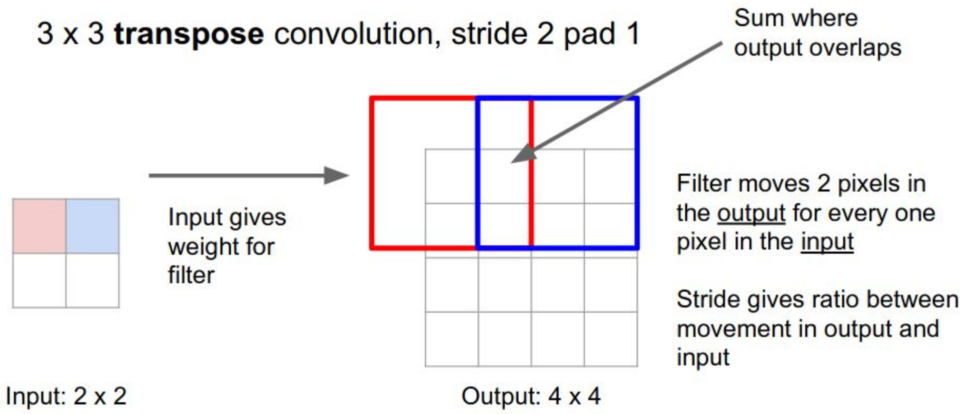
Fully Convolutional

Every pixel makes prediction!

Key notions

- **Conv Transpose** / unpooling operation: Recover the resolution of the input image





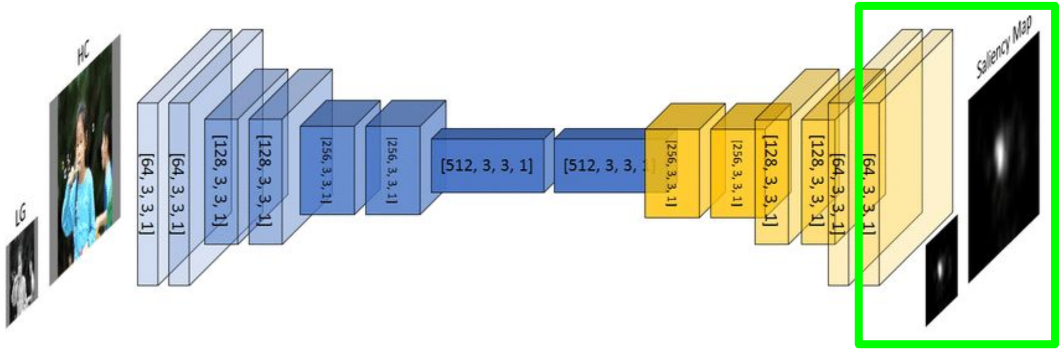
Modern Object Detection Architecture (as of 2017)

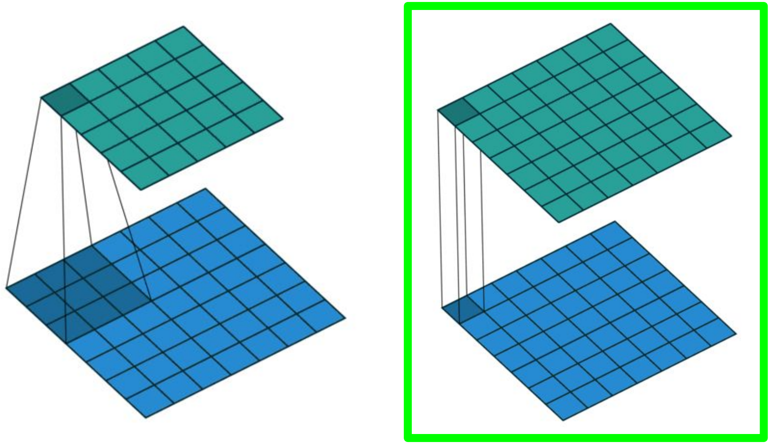
Fully Convolutional

Every pixel makes prediction!

Key notions

- Conv Transpose / unpooling operation
- **1x1 convolution** pixel-wise fully connected layers



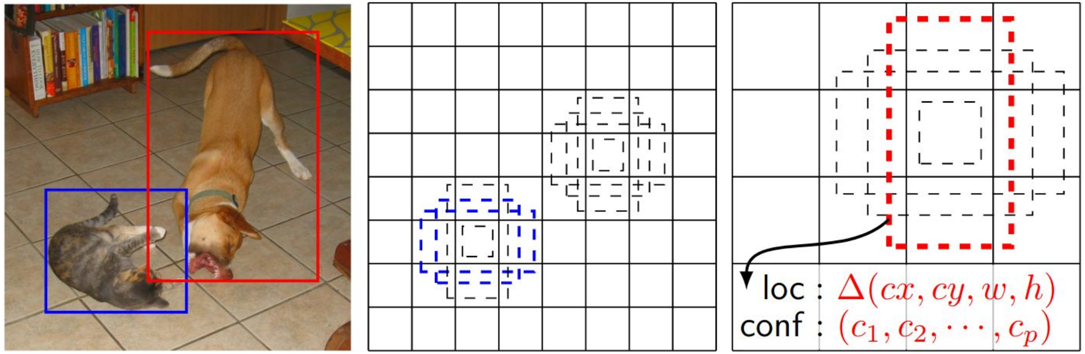


Modern Object Detection Architecture (as of 2017)

Fully Convolutional

Every pixel makes prediction!

⇒ Every pixel predicts bounding boxes that are centered at its location



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

Modern Object Detection Architecture (as of 2017)

Stage 1

- For every output pixel (given by backbone networks)
    - **For every anchor boxes**
        - Predict bounding box offsets
        - Predict anchor confidence
- Suppress overlapping predictions using non-maximum suppression

(Optional, if two-stage networks) Stage 2

- For every region proposals
    - Predict bounding box offsets
    - Predict its semantic class

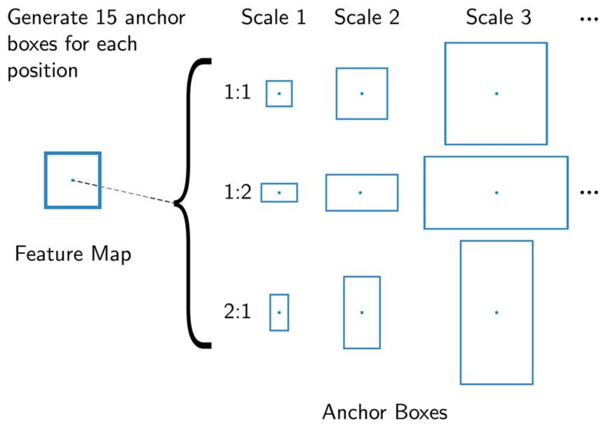
Modern Object Detection Architecture (as of 2017)

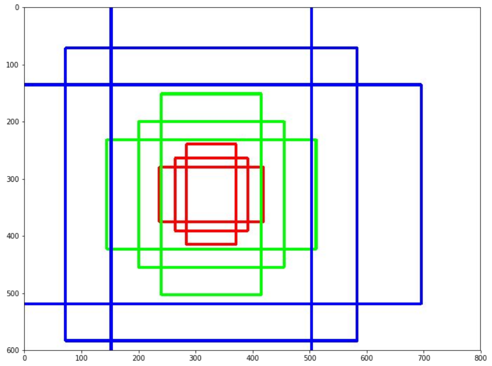
Anchor boxes

Neural network prefers **discrete** prediction over continuous regression!

⇒ Preselect **templates** of bounding boxes to alleviate regression problem

⇒ Let neural network classify the **anchor box** and **small refinement** of it





Modern Object Detection Architecture (as of 2017)

Stage 1

- For every output pixel
    - For every anchor boxes
        - **Predict bounding box offsets**
        - Predict anchor confidence
- Suppress overlapping predictions using non-maximum suppression

(Optional, if two-stage networks) Stage 2

- For every region proposals
    - Predict bounding box offsets
    - Predict its semantic class

Modern Object Detection Architecture (as of 2017)

Bounding box refinement

Given

- Anchor box size (p_w, p_h)
- Output pixel center location (p_x, p_y)

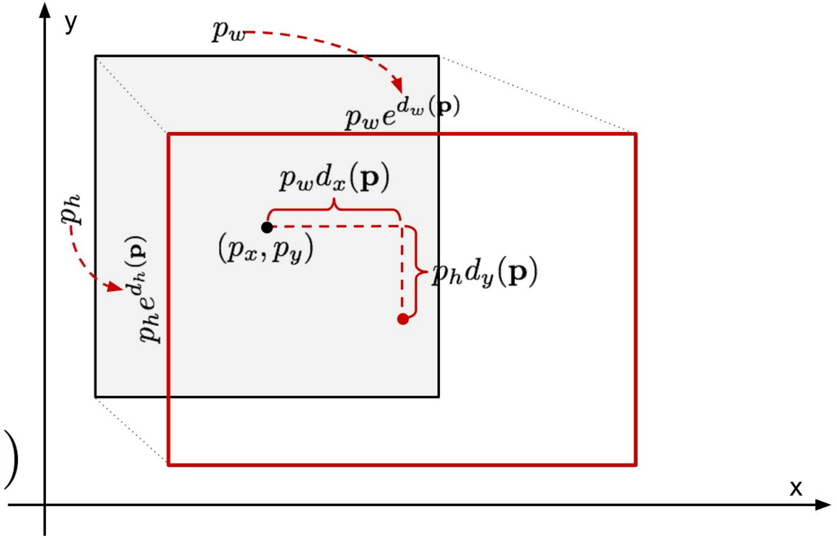
Predict bounding box refinement toward b

- Log-scaled scale relative ratio

$$d_w = \log(b_w/p_w), d_h = \log(b_h/p_h)$$

- Relative center offset

$$d_x = (b_x - p_x)/p_w, d_y = (b_y - p_y)/p_h$$



Modern Object Detection Architecture (as of 2017)

Stage 1

- For every output pixel
  - For every anchor boxes
    - Predict bounding box offsets
    - **Predict anchor confidence**
- Suppress overlapping predictions using non-maximum suppression

(Optional, if two-stage networks) Stage 2

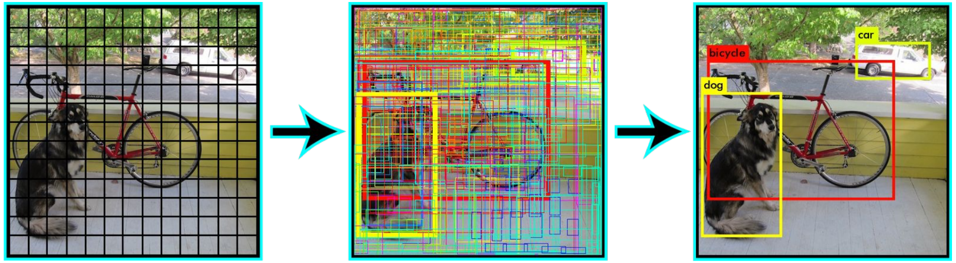
- For every region proposals
  - Predict bounding box offsets
  - Predict its semantic class

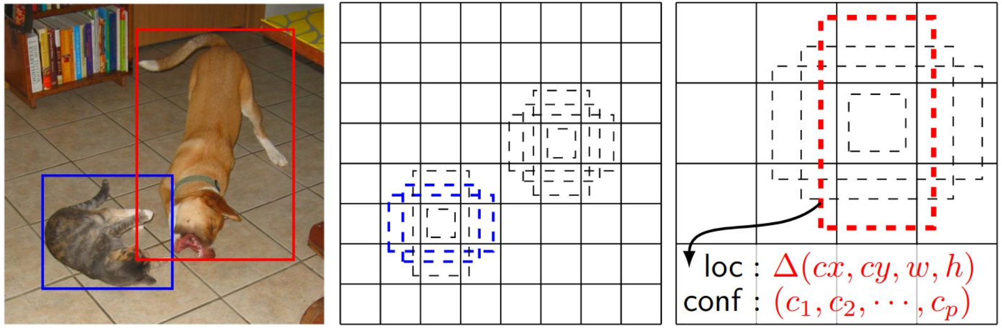
Modern Object Detection Architecture (as of 2017)

Bounding box classification

For each predicted bounding box,

- Predict **confidence** of the box
  ex) binary cross-entropy loss
- (Optional, if 1-stage network)
  Predict **semantic class** of the instance
  ex) categorical cross-entropy loss





(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

Modern Object Detection Architecture (as of 2017)

Stage 1

- For every output pixel
  - For every anchor boxes
    - Predict bounding box offsets
    - Predict anchor confidence
- **Suppress overlapping predictions using non-maximum suppression**

(Optional, if two-stage networks) Stage 2

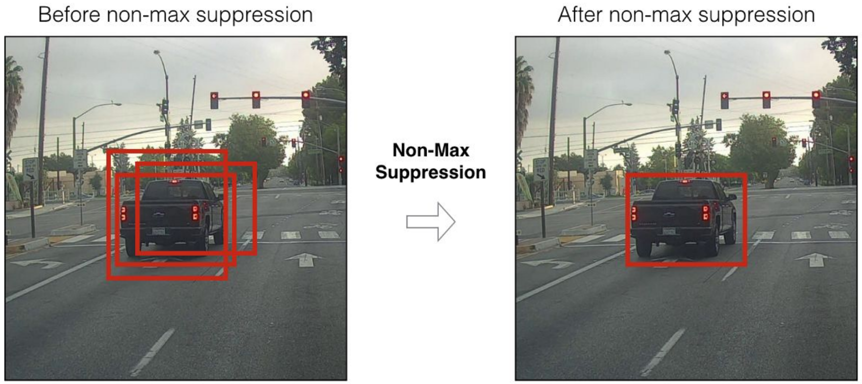
- For every region proposals
  - Predict bounding box offsets
  - Predict its semantic class

Modern Object Detection Architecture (as of 2017)

Non-maximum suppression

The resulting prediction contains multiple predictions of same instance. Heuristics to remove redundant detections

- For all predictions, in descending order of the prediction confidence
    - If the current prediction heavily overlaps with any of the final predictions:
        - Discard it
    - Else
        - Add it to the final prediction



Modern Object Detection Architecture (as of 2017)

Stage 1

- For every output pixel
  - For every anchor boxes
    - Predict bounding box offsets
    - Predict anchor confidence
- Suppress overlapping predictions using non-maximum suppression

(Optional, if two-stage networks) Stage 2

- For every region proposals
  - Predict bounding box offsets
  - Predict its semantic class
- Suppress overlapping predictions using non-maximum suppression

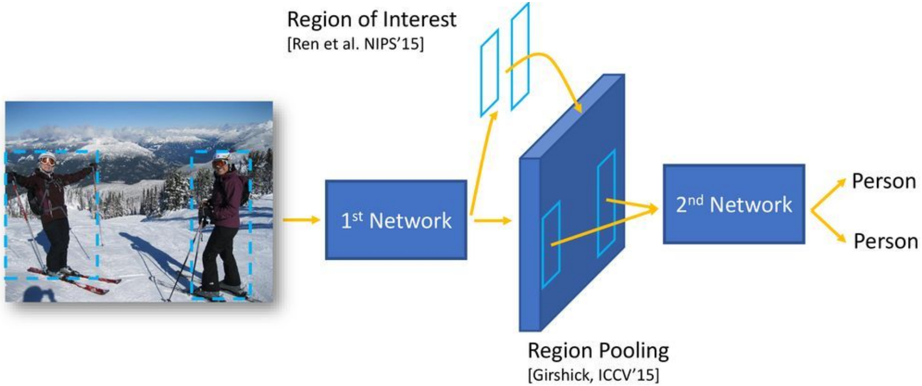
Modern Object Detection Architecture (as of 2017)

Two-stage networks

Second network to **refine** the prediction by the first network

Pro

- Better predictions
  - Better localization
  - Better precision



Con

- Non-standard operation (not favorable for embedded system)
- Slower

Modern Object Detection Architecture (as of 2017)

Stage 1

- For every output pixel
  - For every anchor boxes
    - Predict bounding box offsets
    - Predict anchor confidence
- Suppress overlapping predictions using non-maximum suppression

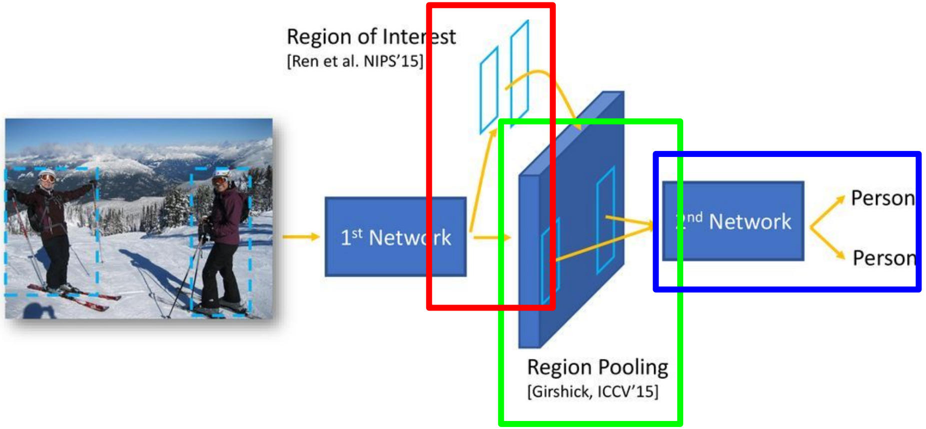
(Optional, if two-stage networks) Stage 2

- **For every region proposals**
  - **Predict bounding box offsets**
  - **Predict its semantic class**
- Suppress overlapping predictions using non-maximum suppression

Modern Object Detection Architecture (as of 2017)

For every region proposal from the fist stage

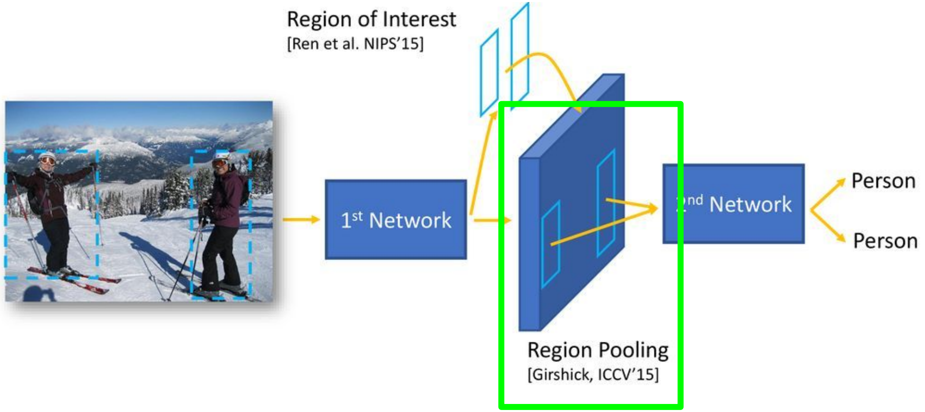
- Extract fixed-size feature corresponding to the region proposal
  Using the extracted features,
  - Predict bounding box offsets
  - Predict its semantic class



Modern Object Detection Architecture (as of 2017)

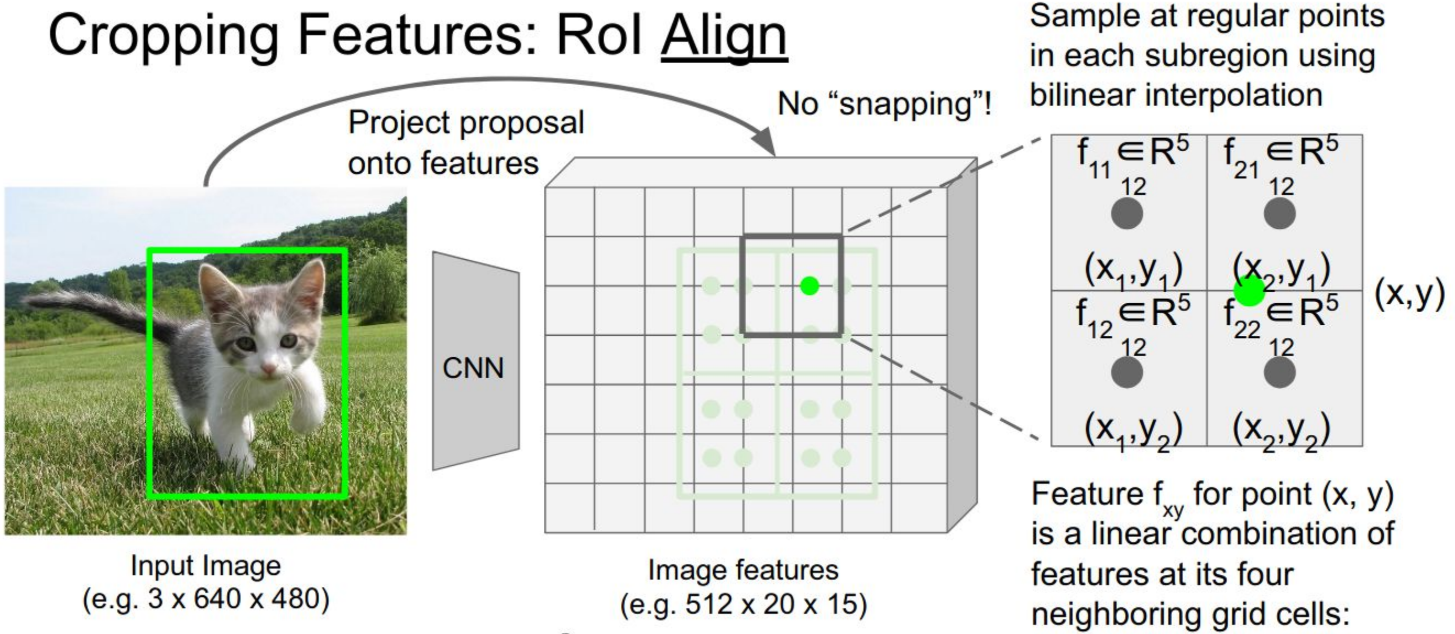
For every region proposal from the fist stage

- **Extract fixed-size feature corresponding to the region proposal**
  Using the extracted features,
    - Predict bounding box offsets
    - Predict its semantic class



Modern Object Detection Architecture (as of 2017)

ROI Align: For every region proposal from the fist stage, extract fixed-size feature



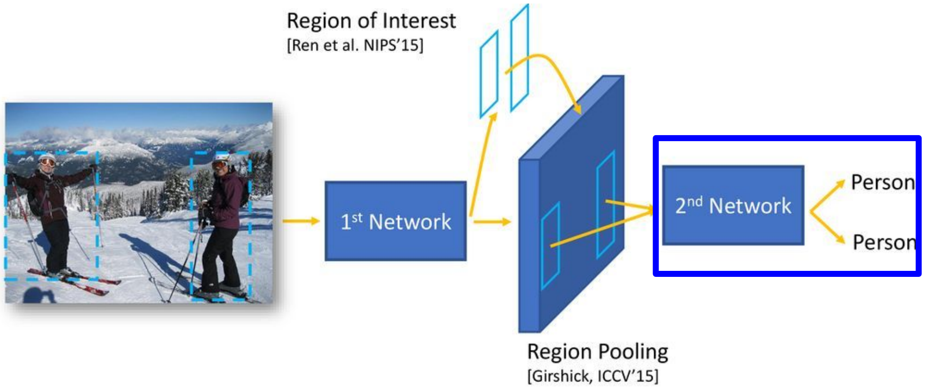
$$f_{xy} = \sum_{i,j=1}^{2} f_{i,j} \max(0, 1 - |x - x_i|) \max(0, 1 - |y - y_j|)$$

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

Modern Object Detection Architecture (as of 2017)

For every region proposal from the fist stage

- Extract fixed-size feature corresponding to the region proposal
  Using the extracted features,
  - **Predict bounding box offsets**
  - **Predict its semantic class**



Modern Object Detection Architecture (as of 2017)

Bounding box refinement

Given

- Region Proposal box size (p_w, p_h)
- Output pixel center location (p_x, p_y)

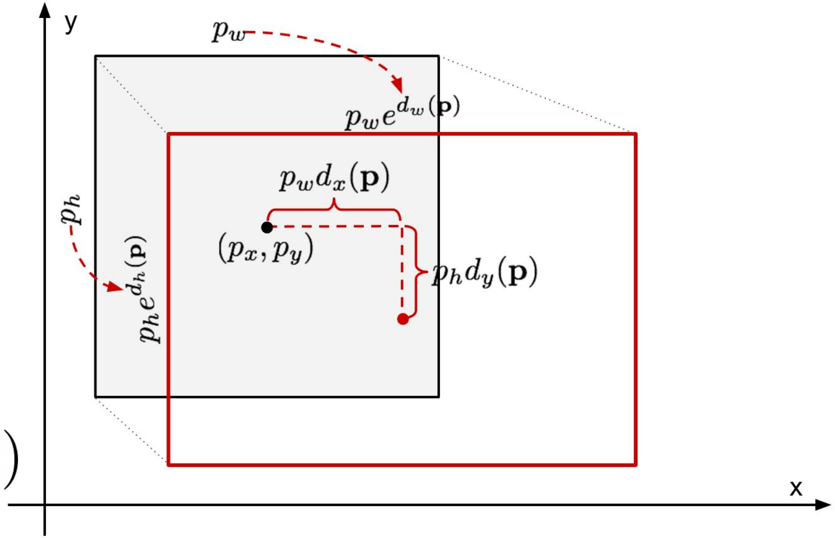
Predict bounding box refinement toward b

- Log-scaled scale relative ratio

$$d_w = \log(b_w/p_w), d_h = \log(b_h/p_h)$$

- Relative center offset

$$d_x = (b_x - p_x)/p_w, d_y = (b_y - p_y)/p_h$$



Modern Object Detection Architecture (as of 2017)

Stage 1

- For every output pixel
    - For every anchor boxes
        - Predict bounding box offsets
        - Predict anchor confidence
- Suppress overlapping predictions using non-maximum suppression

(Optional, if two-stage networks) Stage 2

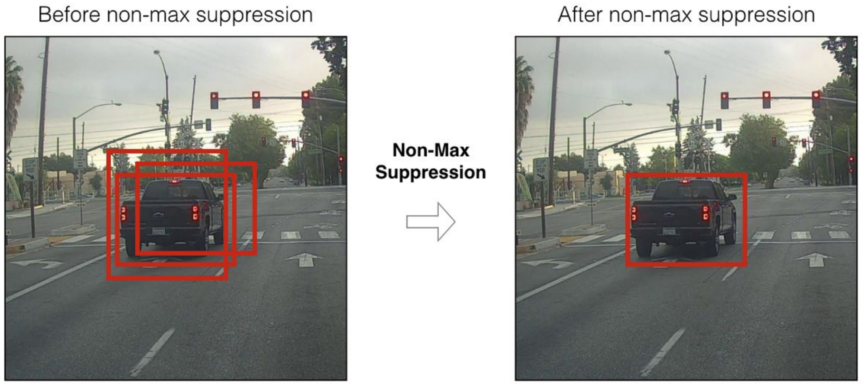
- For every region proposals
    - Predict bounding box offsets
    - Predict its semantic class
- **Suppress overlapping predictions using non-maximum suppression**

Modern Object Detection Architecture (as of 2017)

Non-maximum suppression

The resulting prediction contains multiple predictions of same instance. Heuristics to remove redundant detections

- For all predictions, in descending order of the prediction confidence
    - If the current prediction heavily overlaps with any of the final predictions:
        - Discard it
    - Else
        - Add it to the final prediction



Modern Object Detection Architecture (as of 2017)

Stage 1

- For every output pixel
  - For every anchor boxes
    - Predict bounding box offsets
    - Predict anchor confidence
- Suppress overlapping predictions using non-maximum suppression

(Optional, if two-stage networks) Stage 2

- For every region proposals **(features from corresponding layer of pyramid)**
  - Predict bounding box offsets
  - Predict its semantic class
- Suppress overlapping predictions using non-maximum suppression

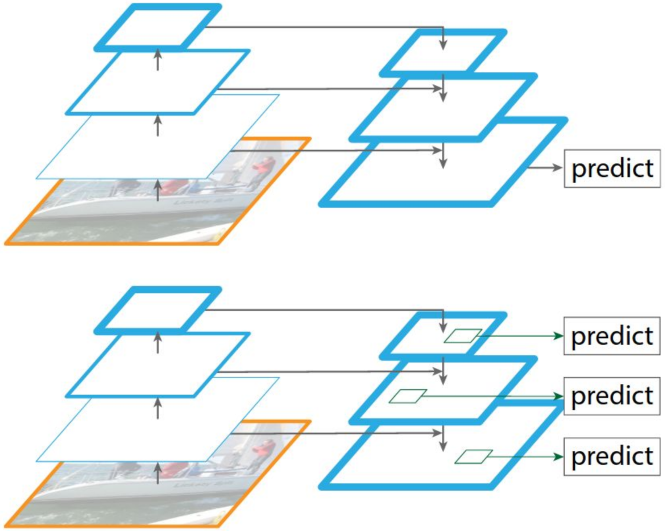
Modern Object Detection Architecture (as of 2017)

Feature Pyramid Networks

Key observation:

Deeper layers of the network has larger receptive fields

⇒ For ROIAlign, extract features for larger bounding boxes from deeper layers of network



$$k = \lfloor k_0 + \log_2(\sqrt{wh}/224) \rfloor$$

Modern Object Detection Architecture (as of 2017)

Stage 1

- For every output pixel
  - For every anchor boxes
    - Predict bounding box offsets
    - Predict anchor confidence
- Suppress overlapping predictions using non-maximum suppression

(Optional, if two-stage networks) Stage 2

- For every region proposals (features from corresponding layer of pyramid)
  - Predict bounding box offsets
  - Predict its semantic class
- Suppress overlapping predictions using non-maximum suppression

Evaluation Metrics

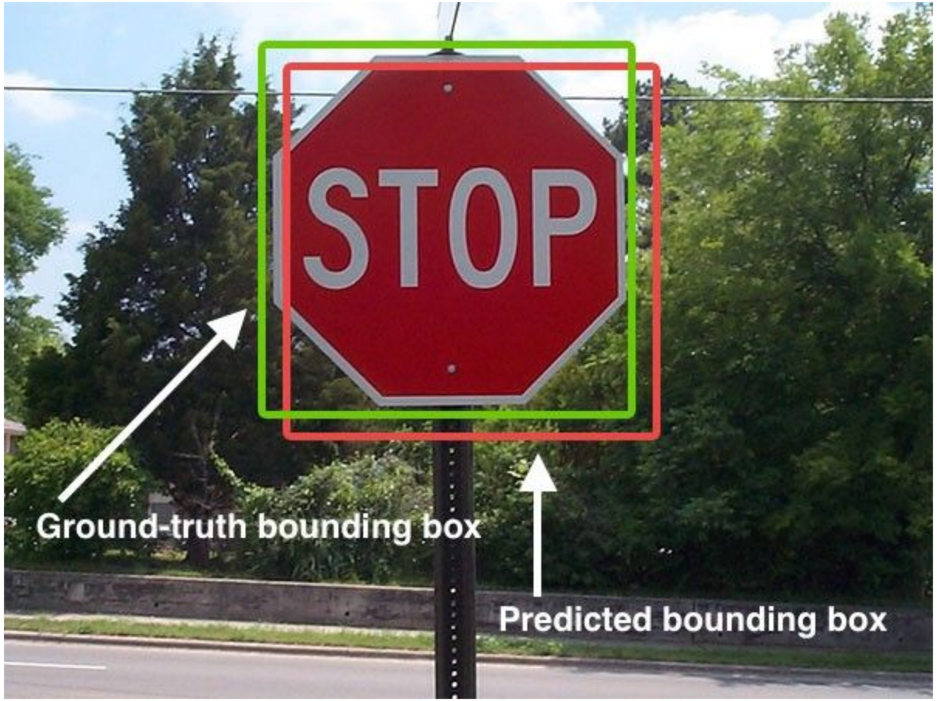
Given:

Single ground-truth bounding box

Single prediction bounding box

Output:

How well are we doing?

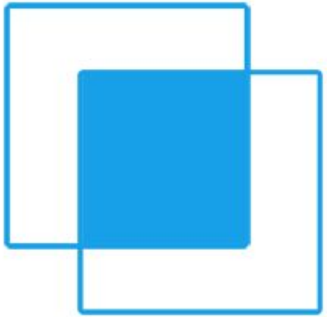


Evaluation Metrics

Given:
Single ground-truth bounding box
Single prediction bounding box

Output:
How well are we doing?

Intersection over Union (IoU)



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

Evaluation Metrics

Given:

Multiple ground-truth bounding box

Multiple prediction bounding box

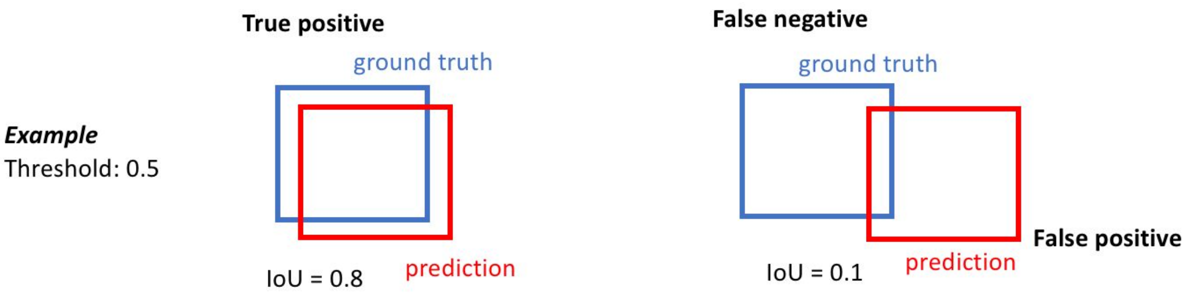
Output:

How well are we doing?

Evaluation Metrics

Match: if all of the conditions are true

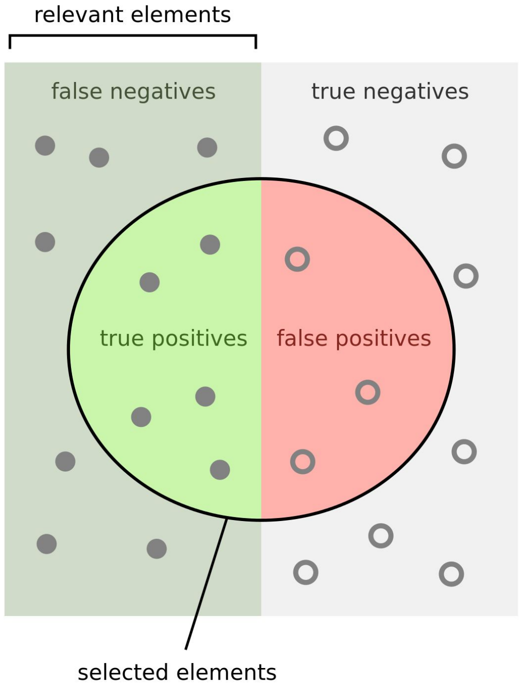
- IoU is between ground-truth and prediction box is above certain threshold
- Their semantic classes are the same
- Only consider 1-to-1 matching.

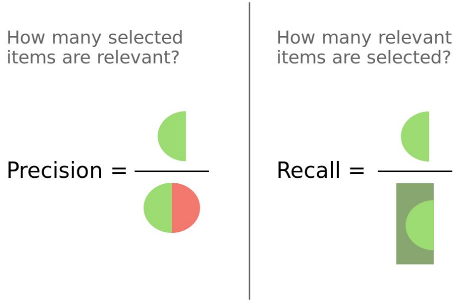


Evaluation Metrics

- **True positive (TP)**: For ground-truth, if there exists a matching prediction
- **False negative (FN)**: For ground-truth, if there is no matching prediction
- **False positive (FP)**: For prediction, if there exists no matching groundruth

- **Precision**: TP / (TP + FP)
- **Recall**: TP / (TP + FN)



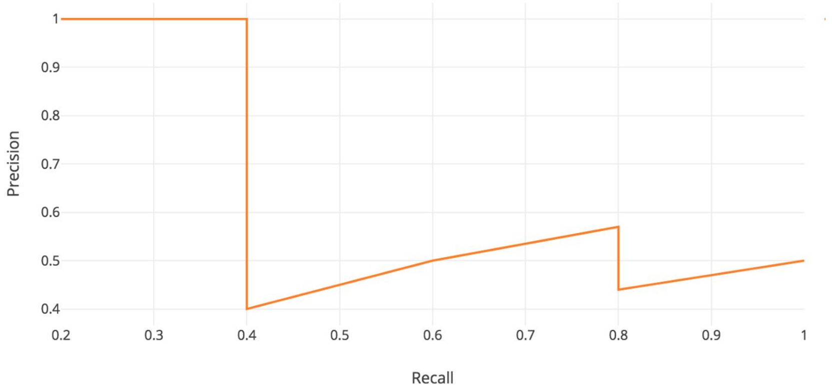


Evaluation Metrics

Average Precision (AP)

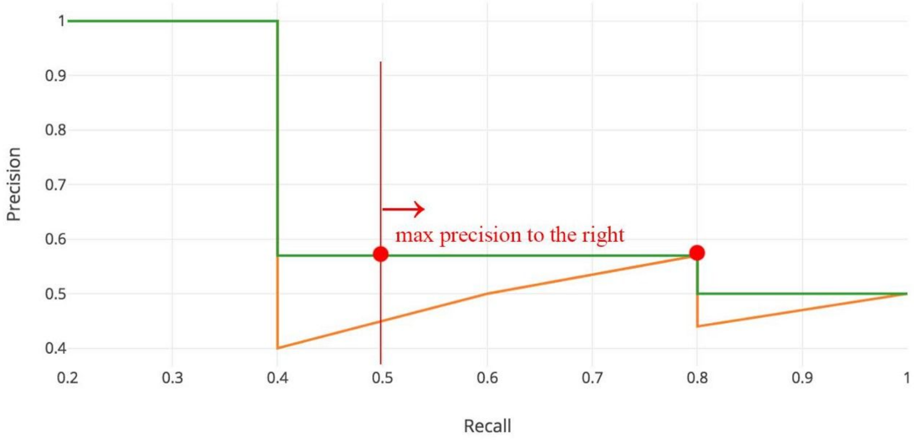
- Go through every prediction in descending order of the prediction confidence
- Calculate and plot Precision / Recall at every step
- Area below the Precision/Recall plot (integral of precisions) is **Average Precision (AP)**

Rank	Correct?	Precision	Recall
1	True	1.0	0.2
2	True	1.0	0.4
3	False	0.67	0.4
4	False	0.5	0.4
5	False	0.4	0.4
6	True	0.5	0.6
7	True	0.57	0.8
8	False	0.5	0.8
9	False	0.44	0.8
10	True	0.5	1.0



Evaluation Metrics

- To make AP more stable to the score ordering, we sometimes take max precision to the right of the AP plot
- We alter the match IoU threshold and take average of them to compute mAP
  - Average of (AP evaluated at matching IoU threshold 0.5, 0.55, 0.6, …, 0.95)

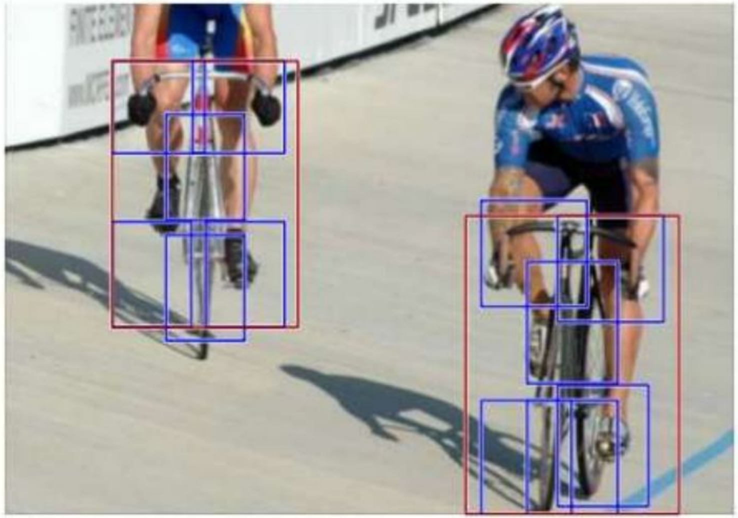


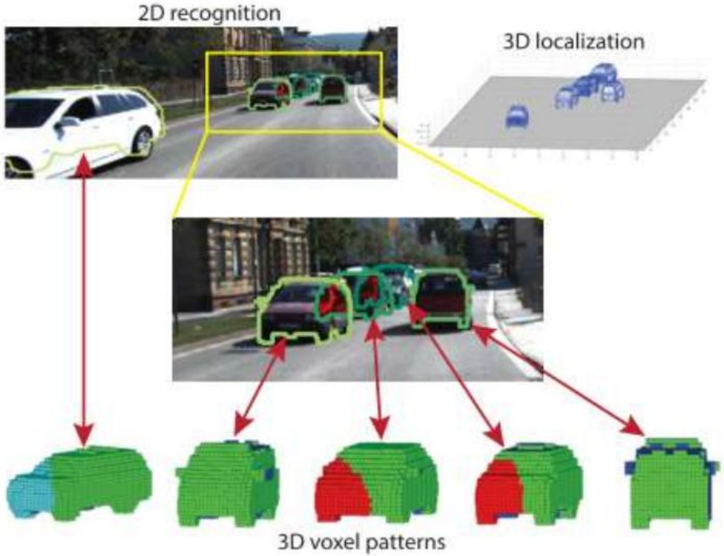
Extensions of 2D Object Detection

- 3D Object Detection
- Instance Segmentation
- Mesh R-CNN
- … and more

3D Object Detection

- 2D bounding boxes are not sufficient
  - Lack of 3D pose, Occlusion information, and 3D location

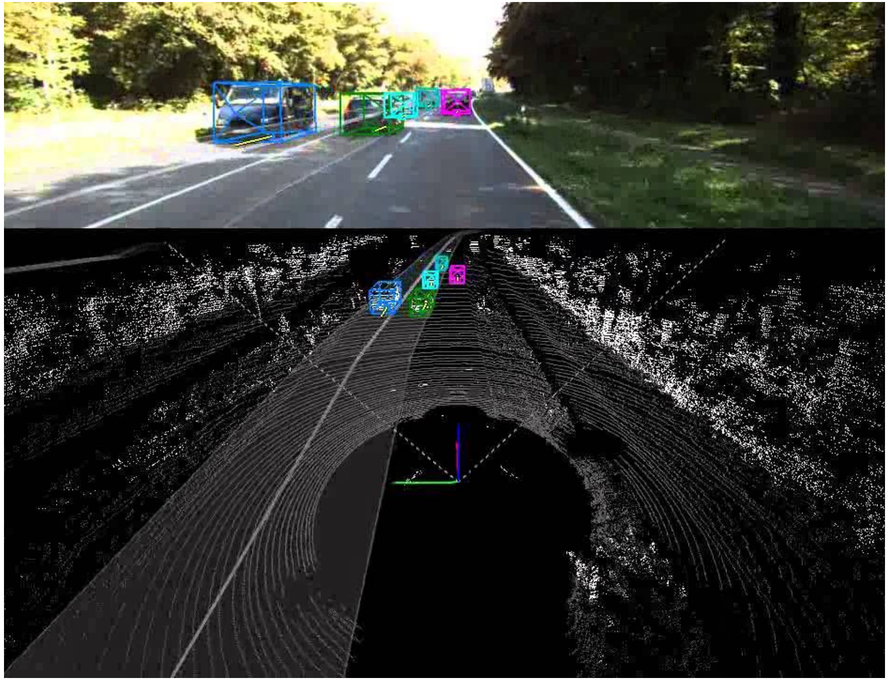




3D Object Detection

Input

- 2D image and/or
- 3D point clouds



Output

- 3D bounding box
  (center location: x, y, z
  bounding box size: w, h, l
  rotation around gravity axis: θ)

The overall pipeline is not too different from that of 2D

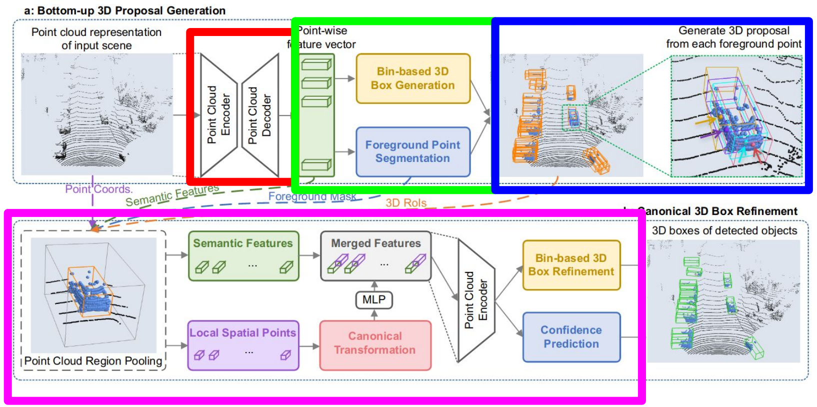
3D Object Detection

Stage 1

- For every output pixel (from backbone)
  - For every anchor boxes
    - Predict bounding box offsets
    - Predict anchor confidence

(Optional, if two-stage networks) Stage 2

- For every region proposals
  - Predict bounding box offsets
  - Predict its semantic class



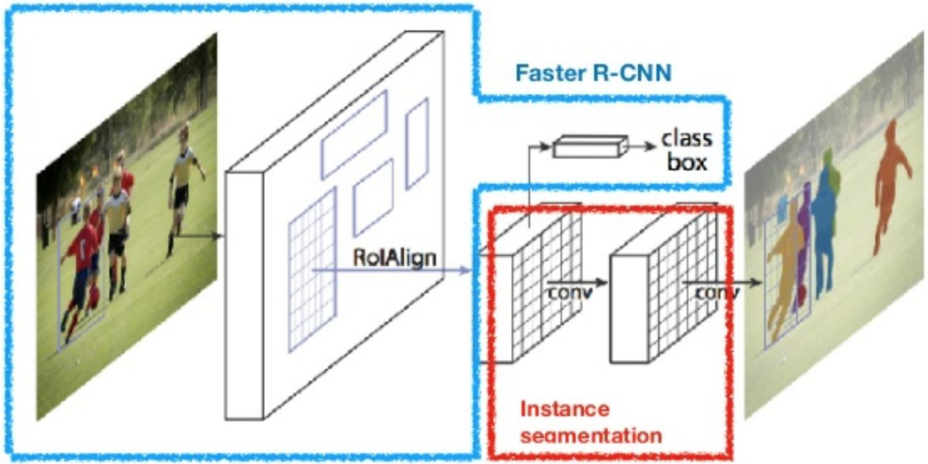
For example,
Point R-CNN

Instance Segmentation

Mask R-CNN

Stage 3

- For every detected instance, predict instance mask

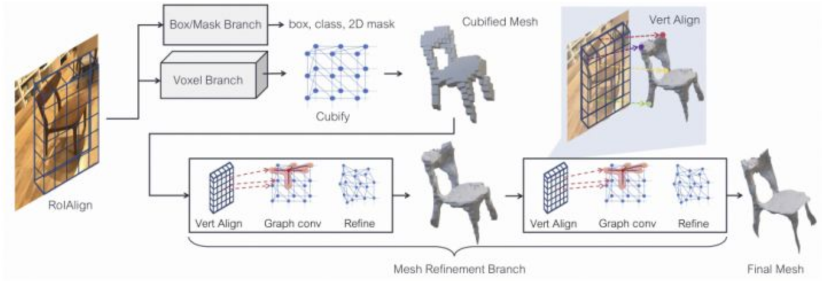


Mesh R-CNN

Mesh R-CNN

Stage 3

- For every detected instance, predict 3D voxels and meshes



Conclusion

Stage 1

- For every output pixel
  - For every anchor boxes
    - Predict bounding box offsets
    - Predict anchor confidence
- Suppress overlapping predictions using non-maximum suppression

(Optional, if two-stage networks) Stage 2

- For every region proposals (features from corresponding layer of pyramid)
  - Predict bounding box offsets
  - Predict its semantic class
- Suppress overlapping predictions using non-maximum suppression