Dataiku Academy

Object Detection with Deep Learning

Classify images

data iku

Classify images with Deep Learning

Goal:

- Choose a unique label for the image

Cat

Dog

Classify images with Deep Learning

How:

- Convolutional Neural Networks (CNN) architectures
  - AlexNet [0], VGG16 [1], Inception [2], ResNet [3], etc.

- Convolution layers
  - Learned kernel convolution that extract image features

- Softmax activation at the top
  - Choose between mutually exclusive class

[0]: https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks [1]: https://arxiv.org/abs/1409.1556, [2]: https://arxiv.org/abs/1409.4842, [3]: https://arxiv.org/abs/1512.03385

Classify images with Deep Learning

Problem:

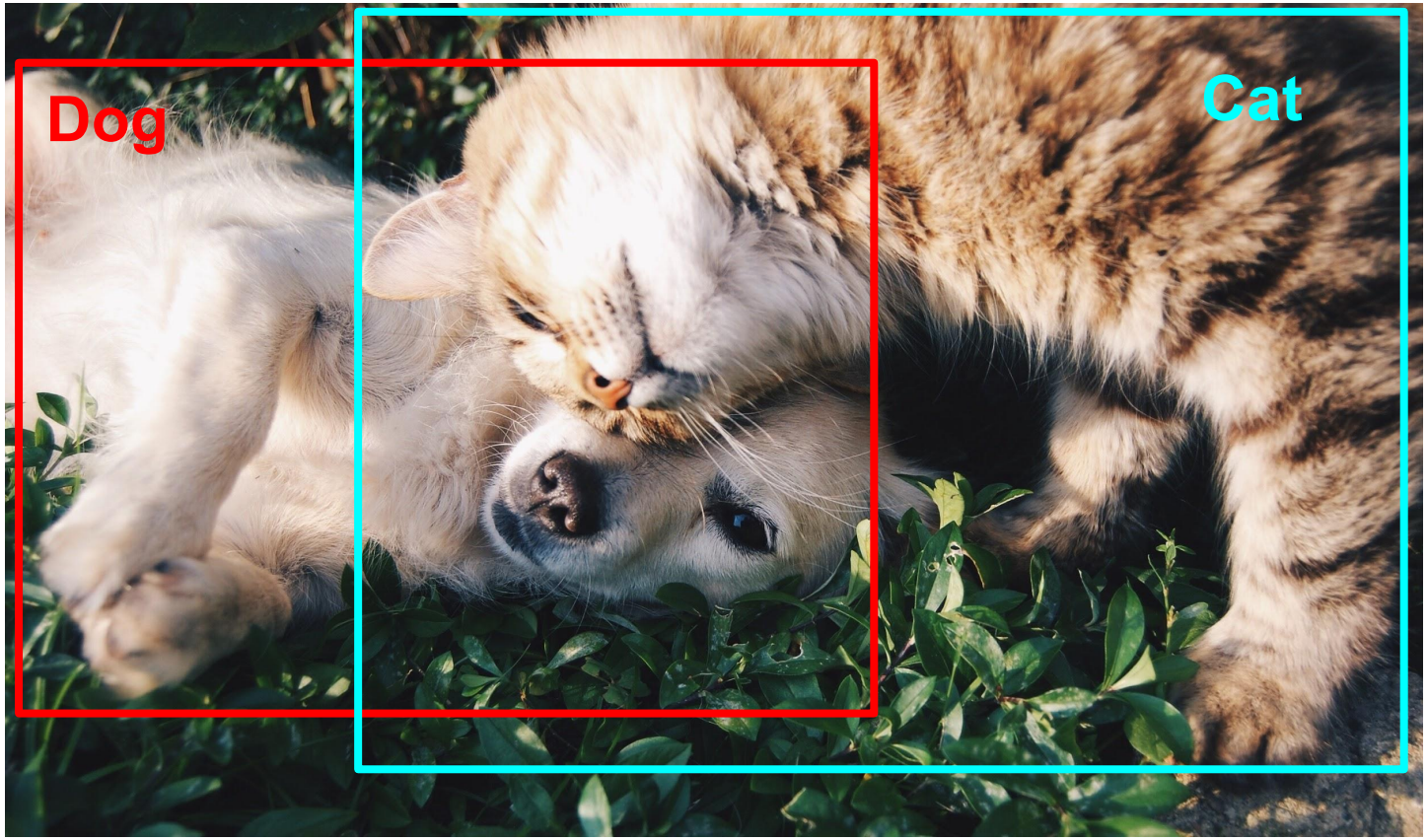
- When two classes are in the same image, is the image a “cat” or “dog”?...
- Where is the cat? On the left or on the right?

Detect objects in images

Detect Objects with Deep Learning

Solution:

- Use a object detection algorithm
  - Find for every objects in the image its label and position



Detect Objects with Deep Learning

Which algorithms?:

- Two big families [4]:
  - Single-stage
  - Two-stage

- Single-stage is fast but not accurate
  - Yolo [5, 6], SSD [7], RetinaNet [8]

- Two-stage is slow but accurate:
  - Faster-RCNN [9], Mask-RCNN [10]

[4]: https://arxiv.org/abs/1803.08707, [5]: https://arxiv.org/abs/1506.02640, [6]: https://arxiv.org/abs/1804.02767, [7]: https://arxiv.org/abs/1512.02325,

[8]: https://arxiv.org/abs/1708.02002, [9]: https://arxiv.org/abs/1506.01497, [10]: https://arxiv.org/abs/1703.06870

A two-stage algorithm:

Faster-RCNN

data iku

Faster-RCNN

The algorithm:

- An incremental improvement over:
  - Selective Search [11], R-CNN [12], Fast-RCNN [13]

- Two-stage algorithms work in … two stages:
  - A network proposing regions of interest (RoI)
  - Choose RoIs, classify, and regress bounding boxes coordinates

[11]: https://koen.me/research/pub/uijlings-ijcv2013-draft.pdf, [12]: https://arxiv.org/abs/1311.2524, [13]: https://arxiv.org/abs/1504.08083

Faster-RCNN

RoI proposal:

- Use a Region Proposal Network (RPN)

- The “conv layers” are ResNet

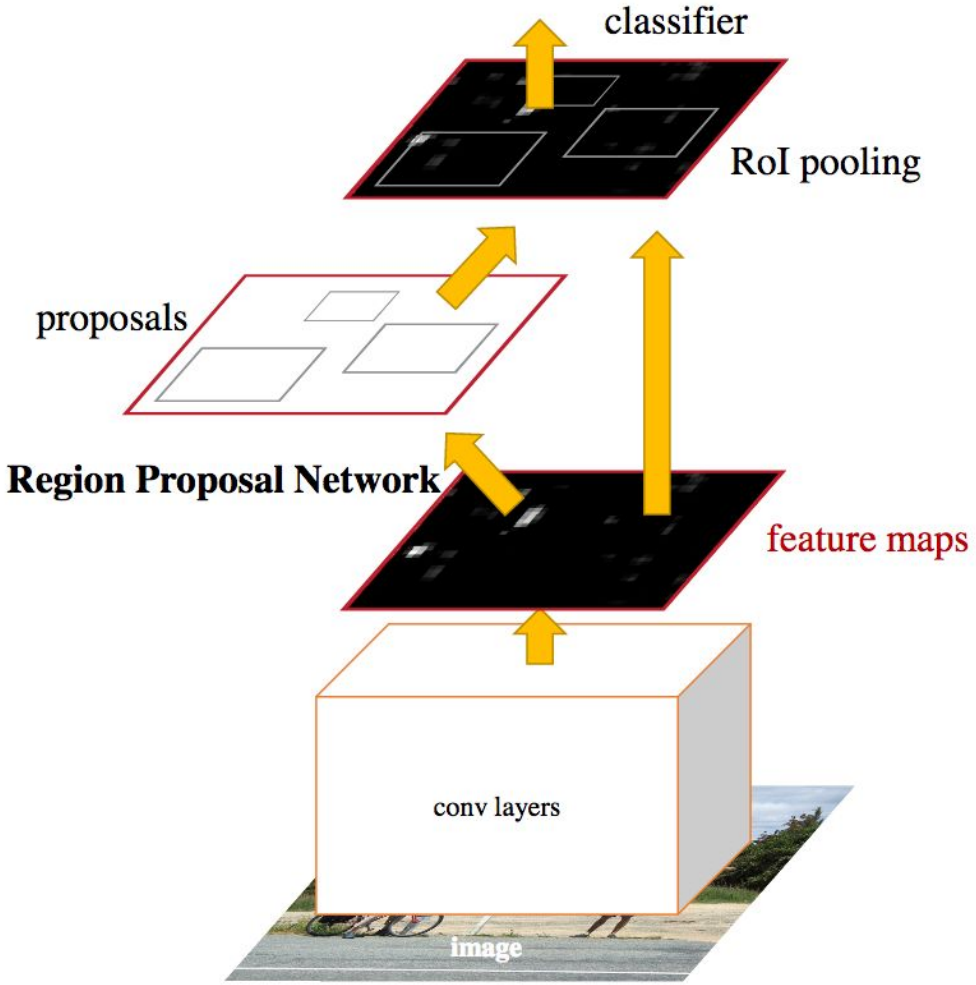


Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the ‘attention’ of this unified network.

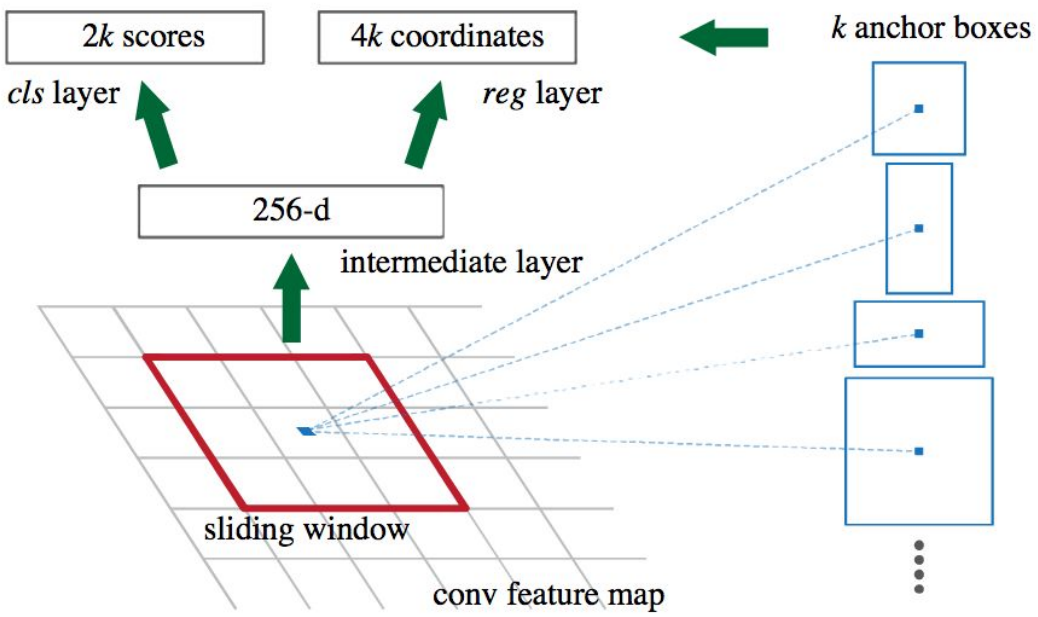
Faster-RCNN

Region Proposal Network:

- Sliding window with anchors of different sizes and ratios on the feature maps

- Binary score with softmax
  - Foreground vs Background

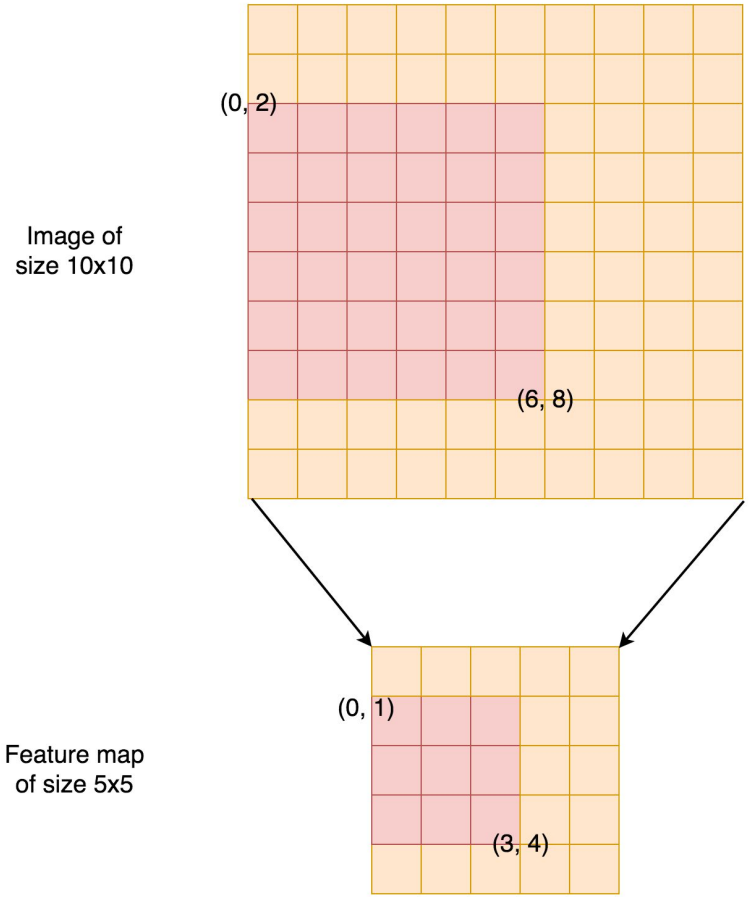
- Regression layer adapt the anchors to make it fit better the objects



Faster-RCNN

RoI pooling:

- We need to extract RoI on the feature maps.
- Problem: Not all RoI have the same shape



Extracted feature map

4	5	3
6	7	0
8	3	6

Creation of as-possible even sections

4	5	3
6	7	0
8	3	6

Max Pooling

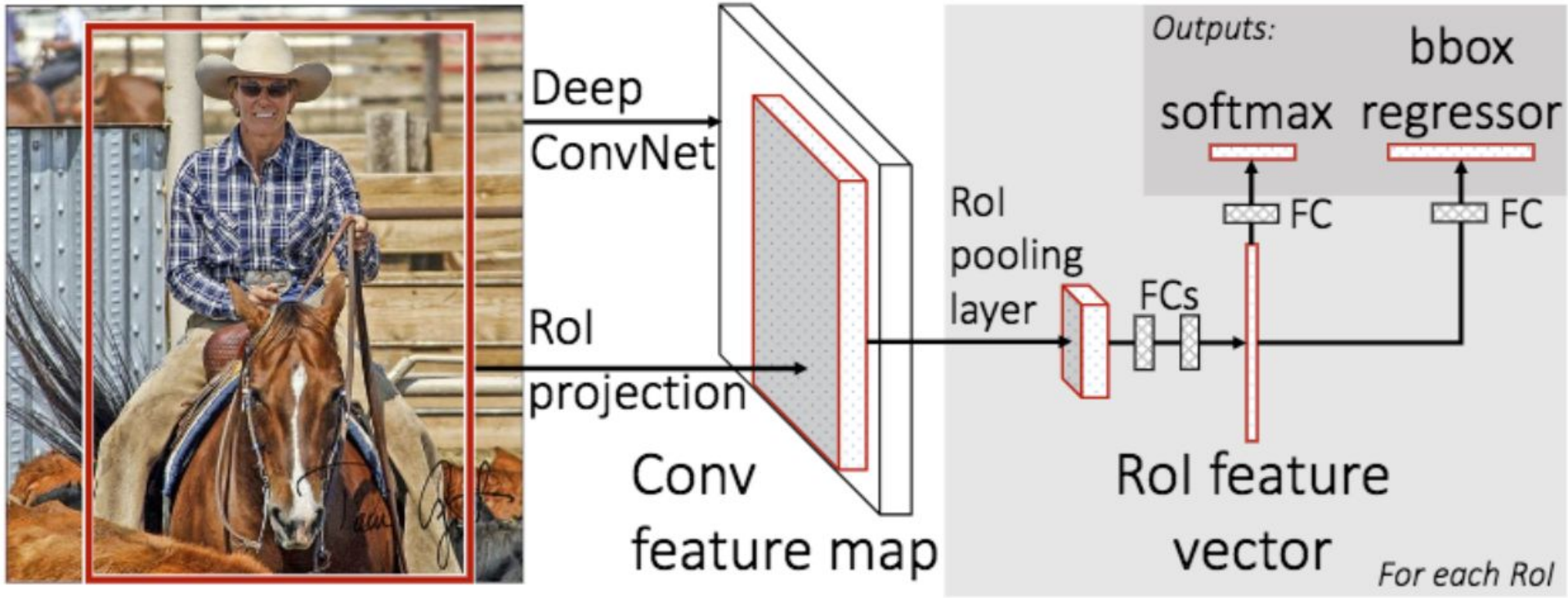
7	3
8	6

More pretty images here: https://arthurdouillard.com/2018/03/26/fast-rcnn

Faster-RCNN

Classifying and regressing:

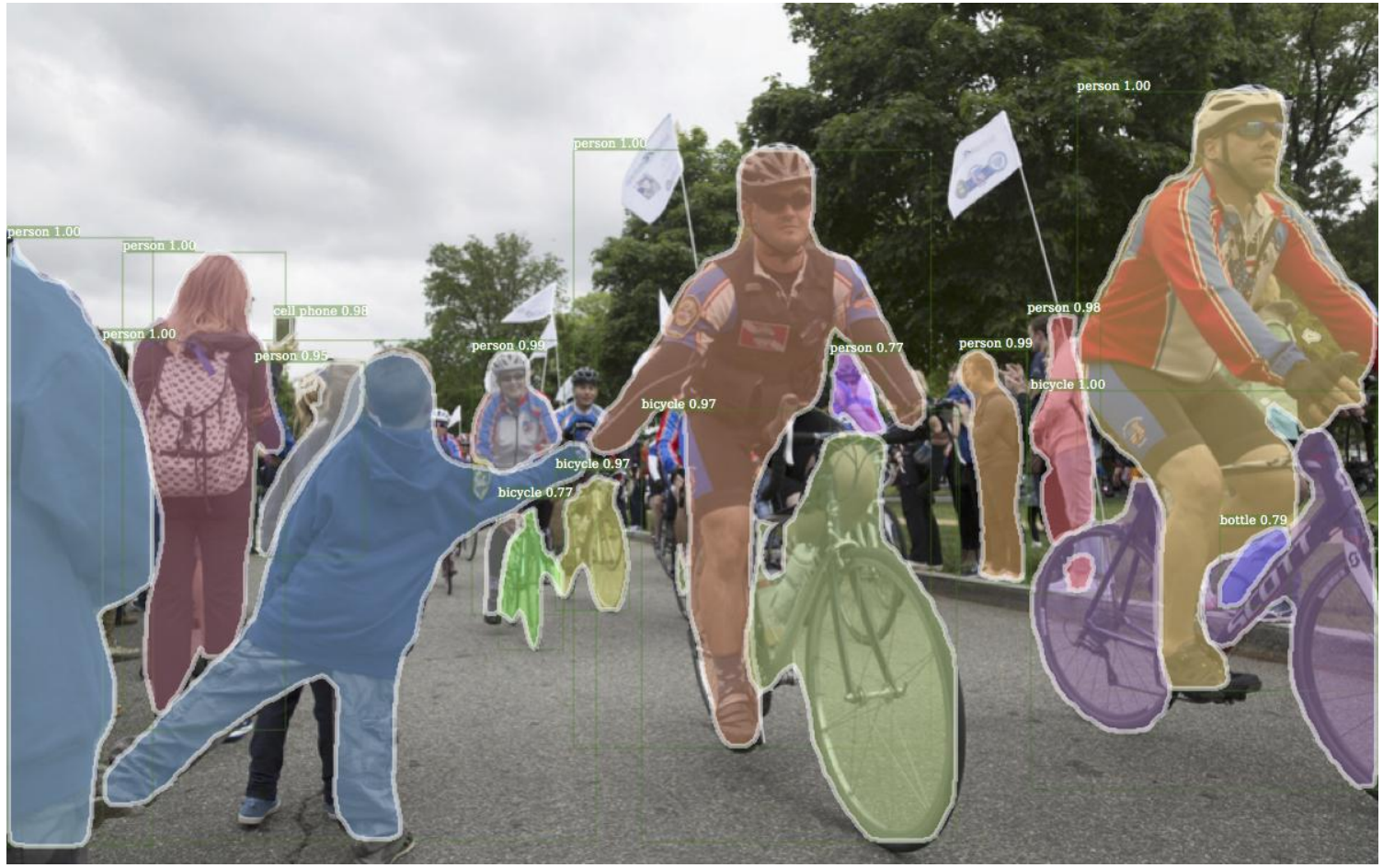
- After the RoI pooling, a few fully connected
- Two heads: One for classify each RoI, one to regress box coordinates



Mask-RCNN

Extending the algorithm:

- Faster-RCNN can be extended to do object segmentation



Mask-RCNN

RoI Align:

- In order to do *pixel-to-pixel* (aka segmentation) the RoI pooling must preserve the explicit per-pixel spatial correspondence
- Instead of discrete quantization (splitting the feature maps in four zones):
  - use bilinear interpolation
- Apply then max pooling

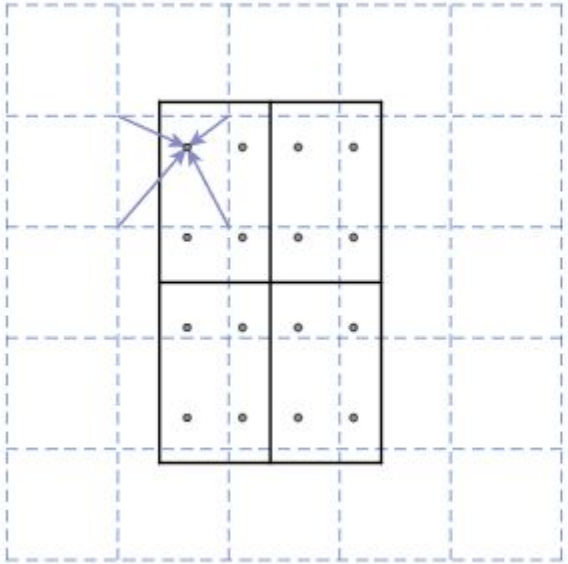


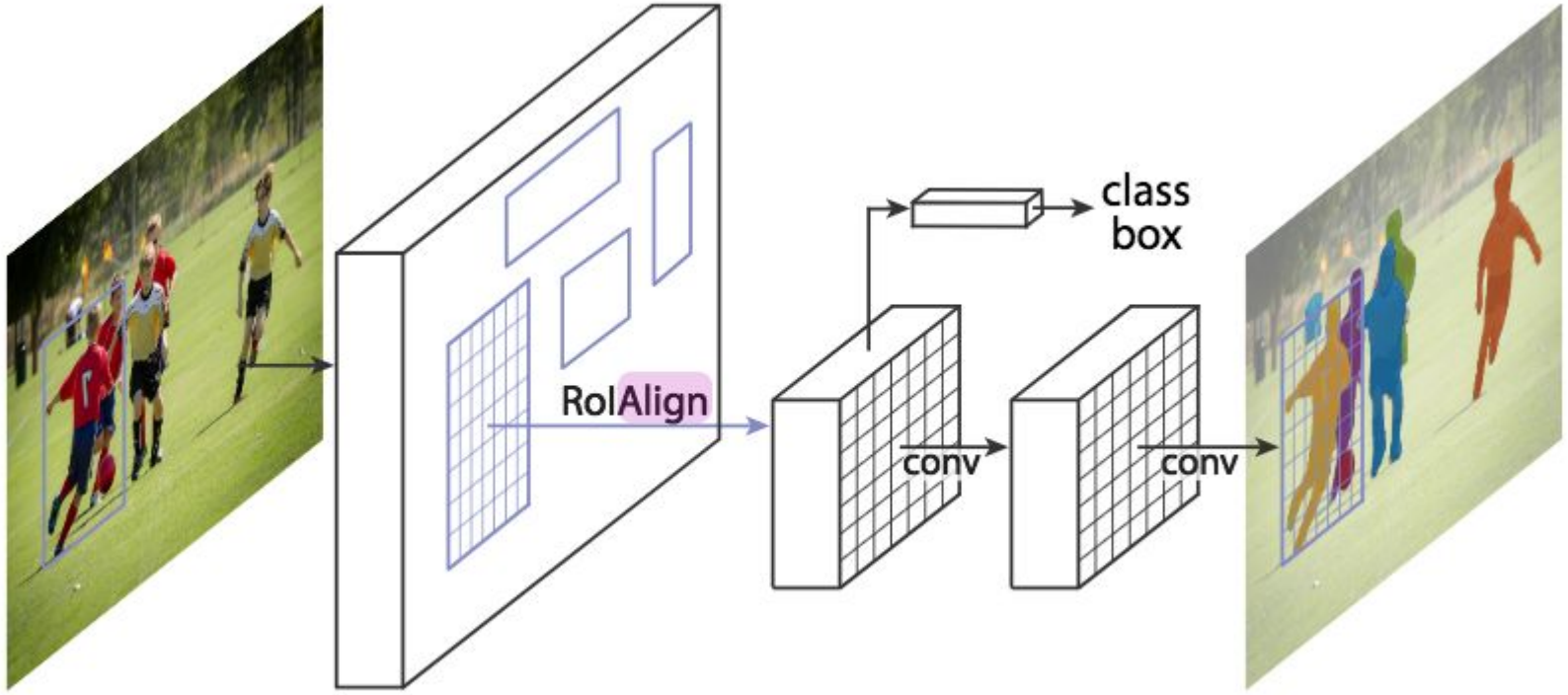
Figure 3. **RoIAlign:** The dashed grid represents a feature map, the solid lines an RoI (with 2×2 bins in this example), and the dots the 4 sampling points in each bin. RoIAlign computes the value of each sampling point by bilinear interpolation from the nearby grid points on the feature map. No quantization is performed on any coordinates involved in the RoI, its bins, or the sampling points.

Mask-RCNN

Segmentation:

- Keep the Faster-RCNN heads:
  - Classification and regression
- Add new head computing a binary mask for each possible objects

- The classification head will determine which mask to keep



A single-stage algorithm:

RetinaNet

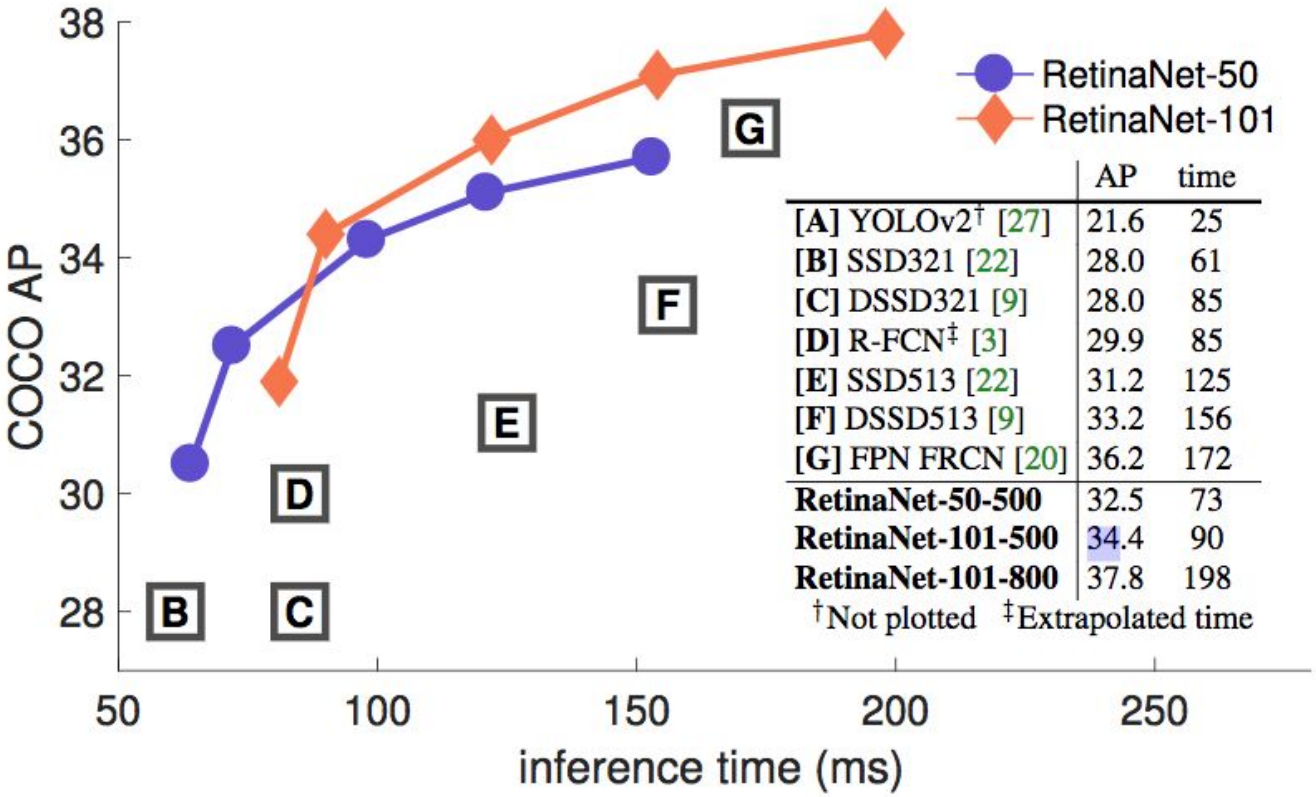
data iku

RetinaNet

The algorithm:

- Best accuracy among single-stage … and two-stage algorithms
- Faster than two-stage, but still way slower than Yolo

- Recent YoloV3 *may* be more accurate than RetinaNet (depending on the benchmark)



	AP	time
[A] YOLOv2† [27]	21.6	25
[B] SSD321 [22]	28.0	61
[C] DSSD321 [9]	28.0	85
[D] R-FCN‡ [3]	29.9	85
[E] SSD513 [22]	31.2	125
[F] DSSD513 [9]	33.2	156
[G] FPN FRCN [20]	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	37.8	198

†Not plotted ‡Extrapolated time

RetinaNet

The algorithm:

- Two stage algorithms remove easy background during first stage
- Single stage algorithms struggle to deal with huge amount of easy background

- RetinaNet adds the *Focal Loss* that discard easy background.

RetinaNet

Focal Loss:

- Designed to down-weight the loss from easy examples

- Example with 2 classes:
  Foreground and background

$$\mathrm{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise.} \end{cases} \tag{1}$$

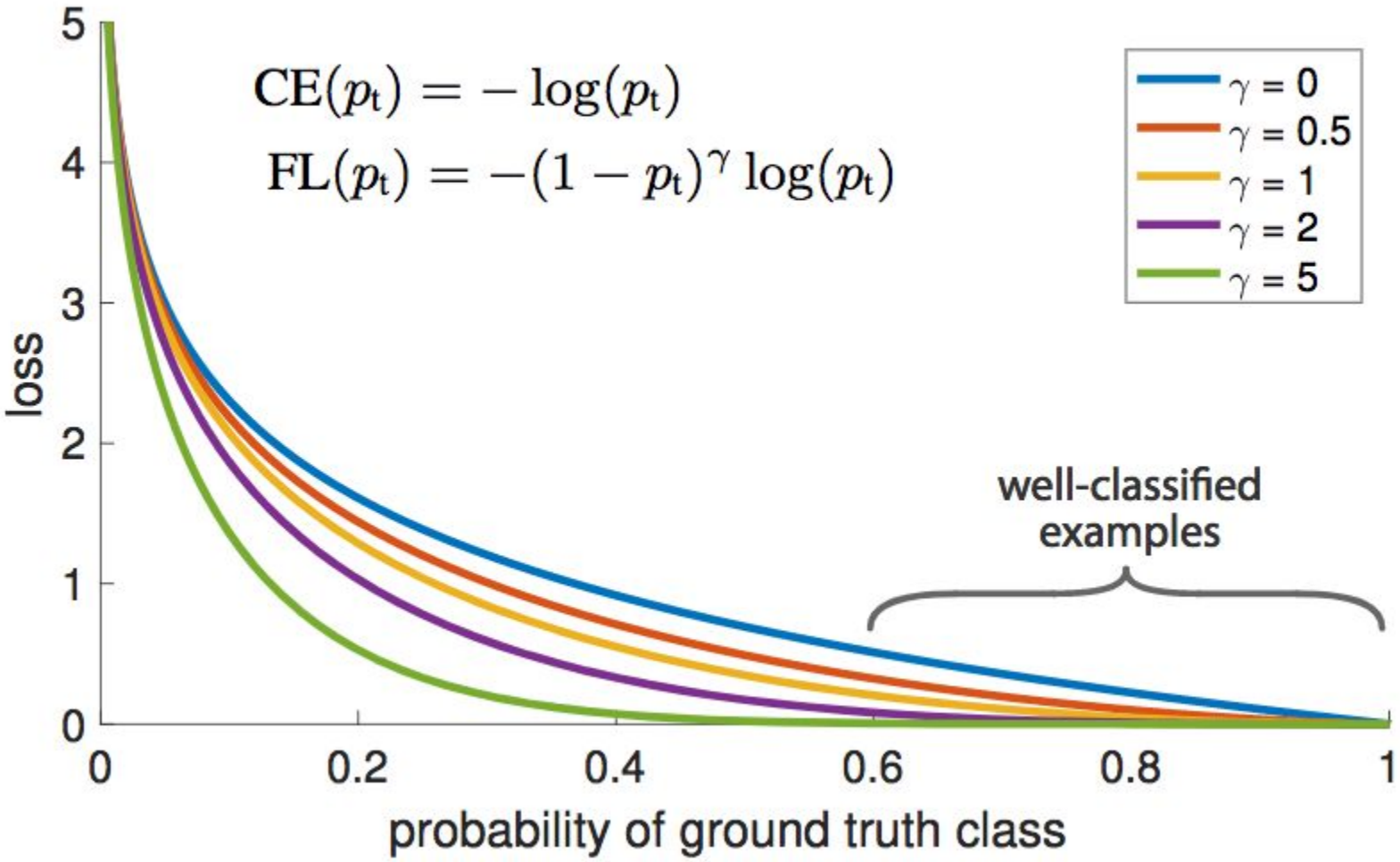
In the above $y \in \{\pm 1\}$ specifies the ground-truth class and $p \in [0, 1]$ is the model's estimated probability for the class with label $y = 1$. For notational convenience, we define p_t:

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise,} \end{cases} \tag{2}$$

$$\mathrm{FL}(p_t) = -(1-p_t)^{\gamma} \log(p_t).$$

RetinaNet

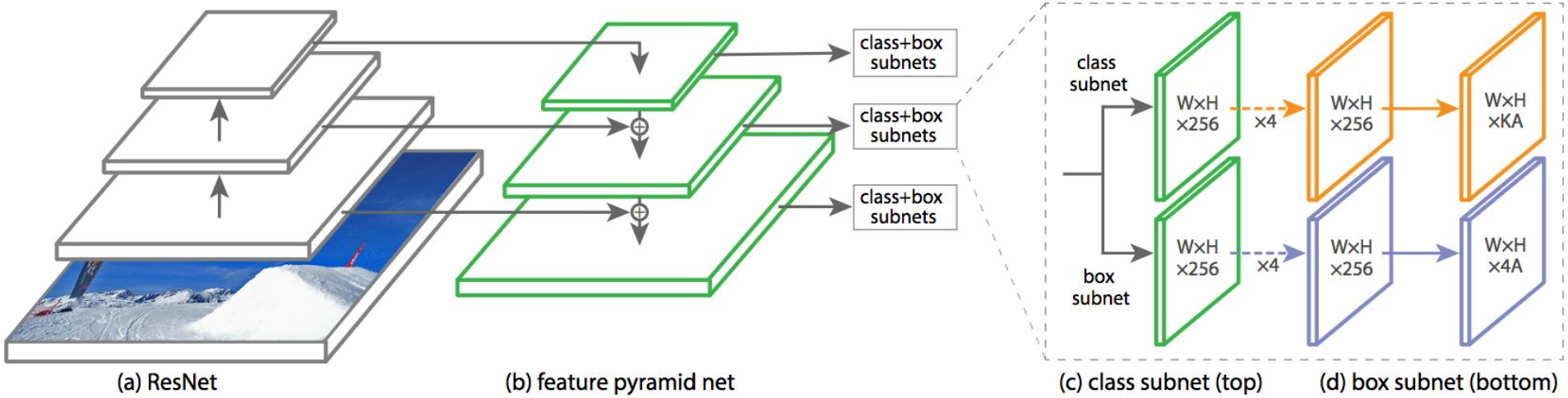
Focal Loss:



RetinaNet

The architecture:

- ResNet + Feature Pyramid Network (FPN)[14] + 2 Fully Connected Networks (FCN) [15]

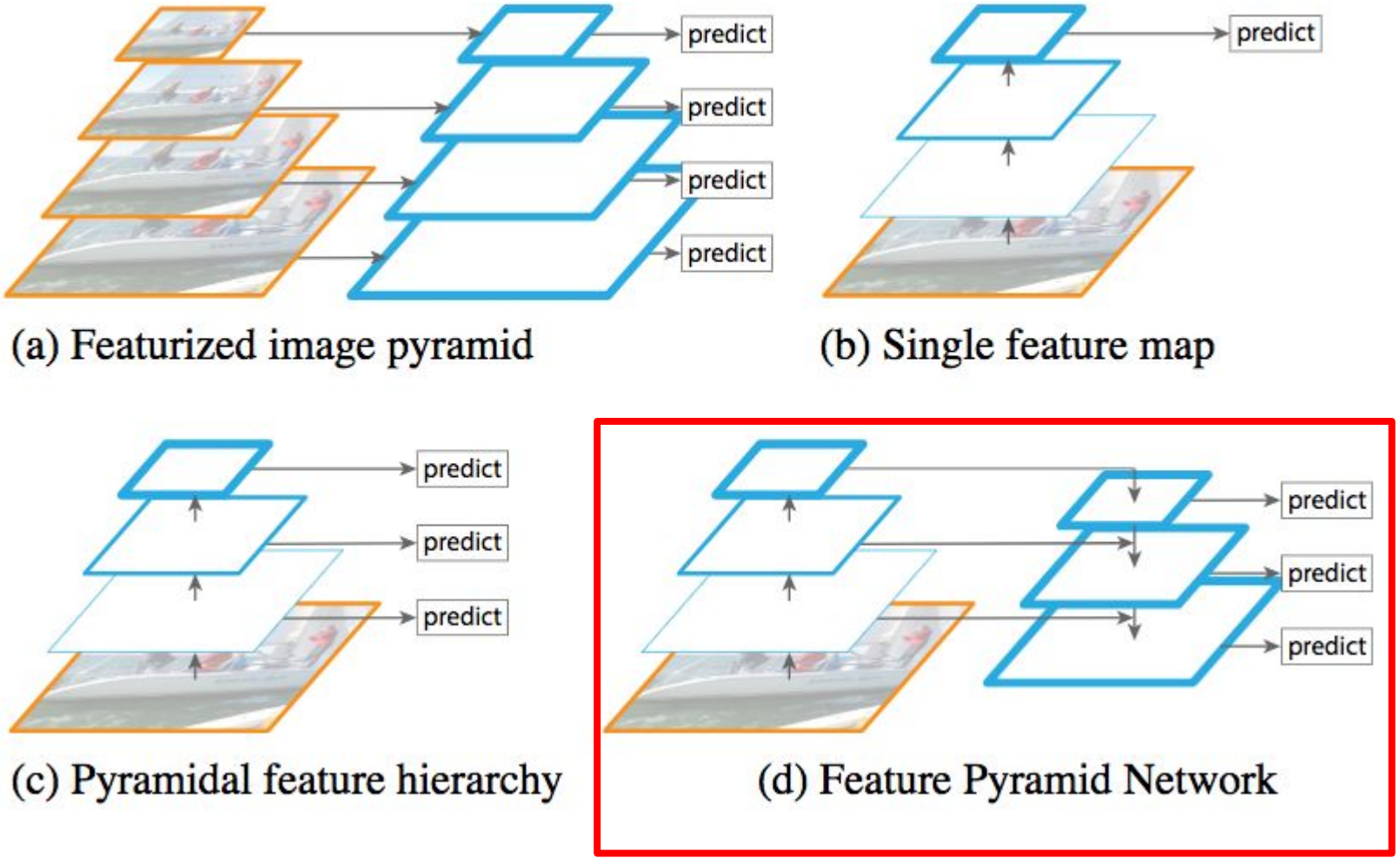


[14]: https://arxiv.org/abs/1612.03144, [15]: https://arxiv.org/abs/1605.06409

RetinaNet

Feature Pyramid Network:

- Scale invariant
- Share feature maps between several ResNet layers



(a) Featurized image pyramid

(b) Single feature map

(c) Pyramidal feature hierarchy

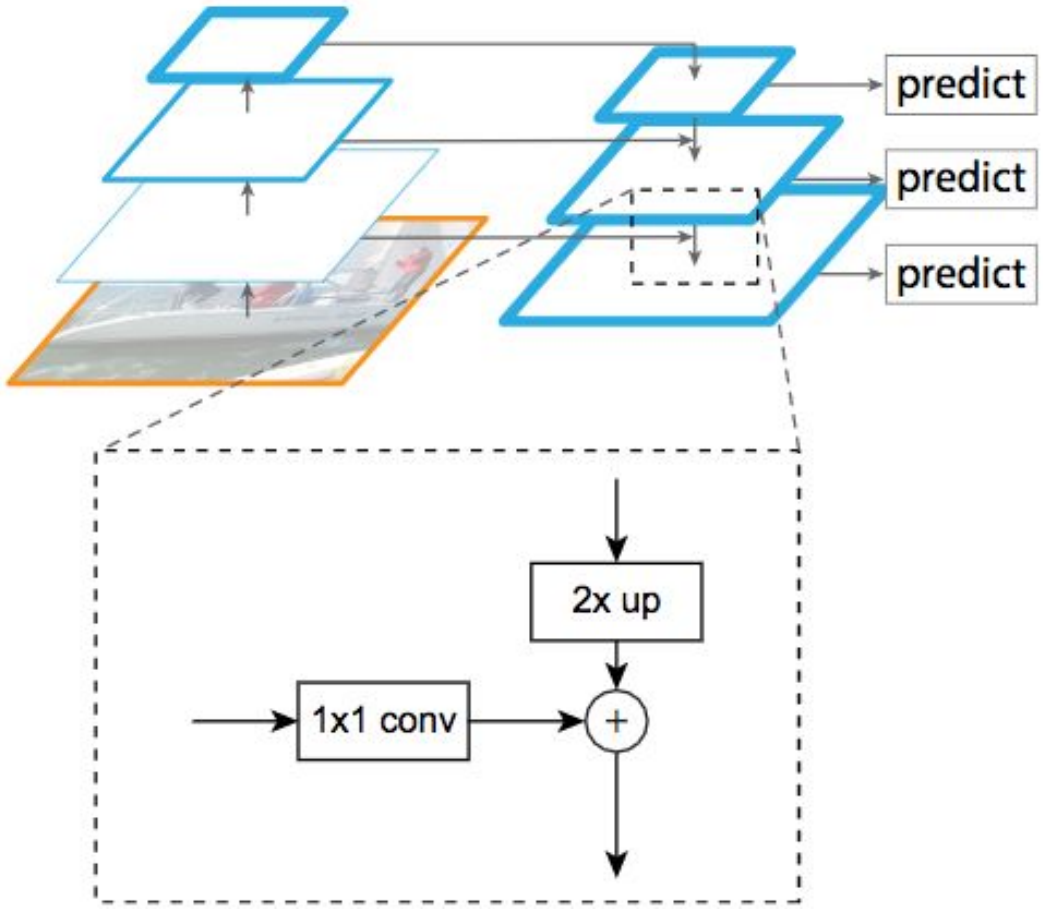
(d) Feature Pyramid Network

RetinaNet

Feature Pyramid Network:

- Scale invariant
- Share feature maps between several ResNet layers

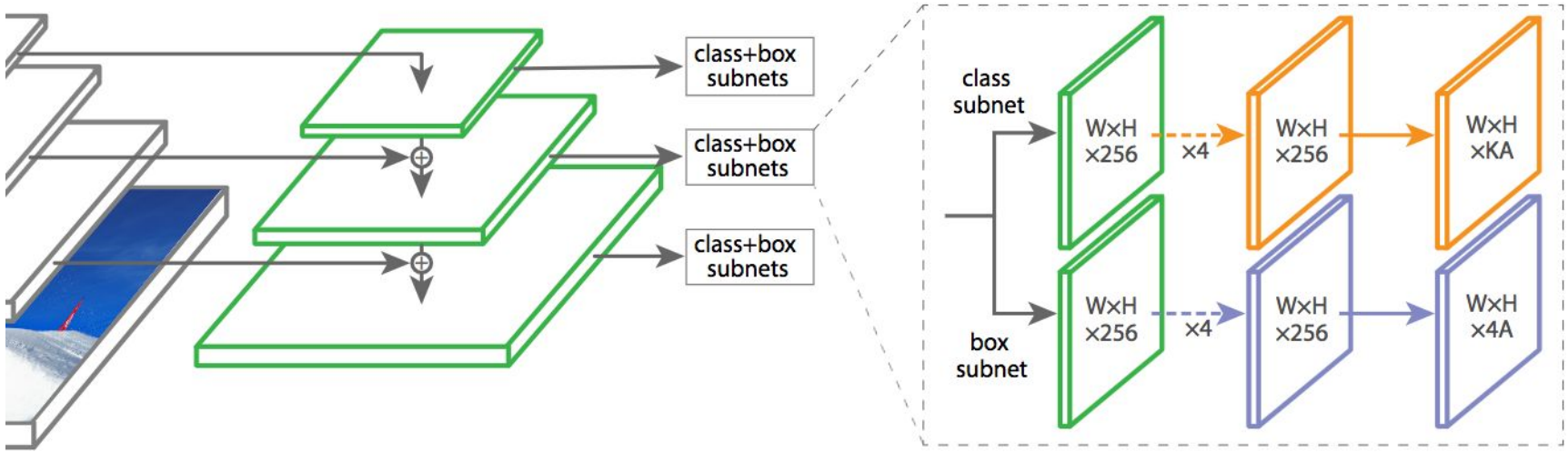
- Combination of fine-grained features and high-level features



RetinaNet

Fully Connected Networks:

- Subnets at each level share parameters
- Because it is only convolution, input feature maps can be of different sizes



(b) feature pyramid net (c) class subnet (top) (d) box subnet (bottom)

Common post-processing

data iku

Common post-processing

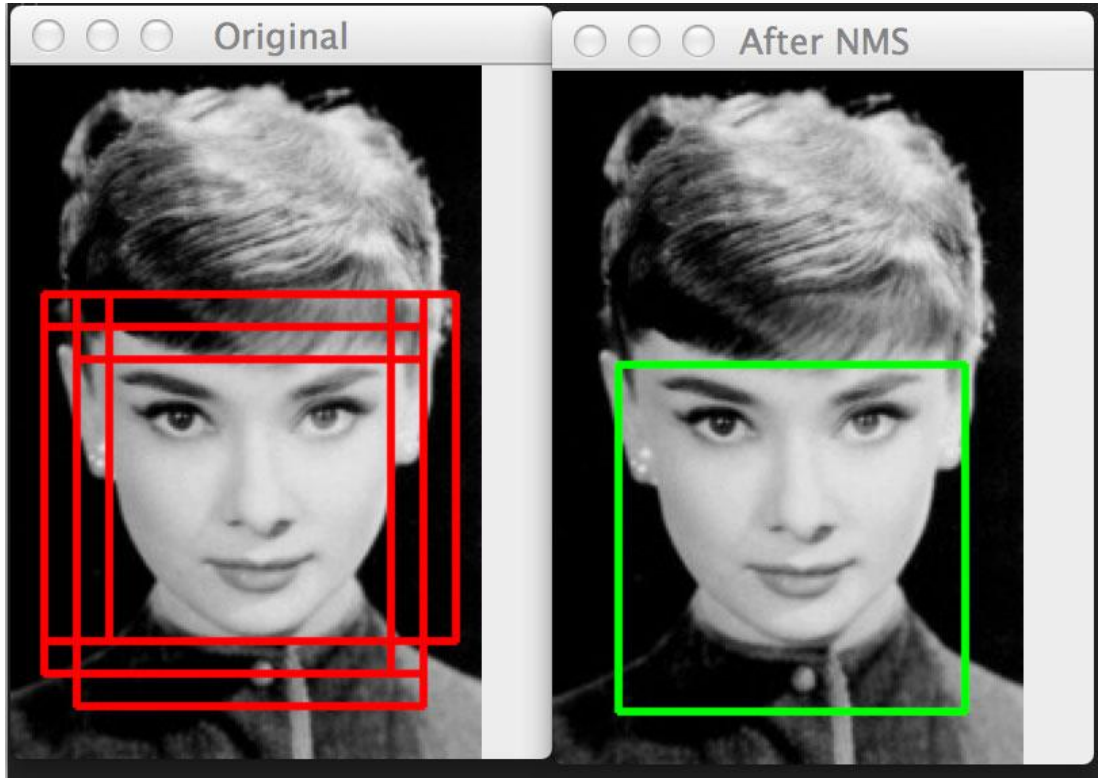
Confidence:

- Each class score can be seen as “confidence”. Keep only those above threshold.
  - Usually 0.5. In the case of satellite/aerial imagery 0.3-0.4.

Non-Max Suppression (NMS):

- Strips overlapping boxes
- Can be based on Jaccard index.

  0.3 is a good start.



Thank you for attending!

data
iku