Feature Pyramid Networks (FPN) for Object Detection/feature semantic

Ray Kuo

2018-9-13 NTU

FPN

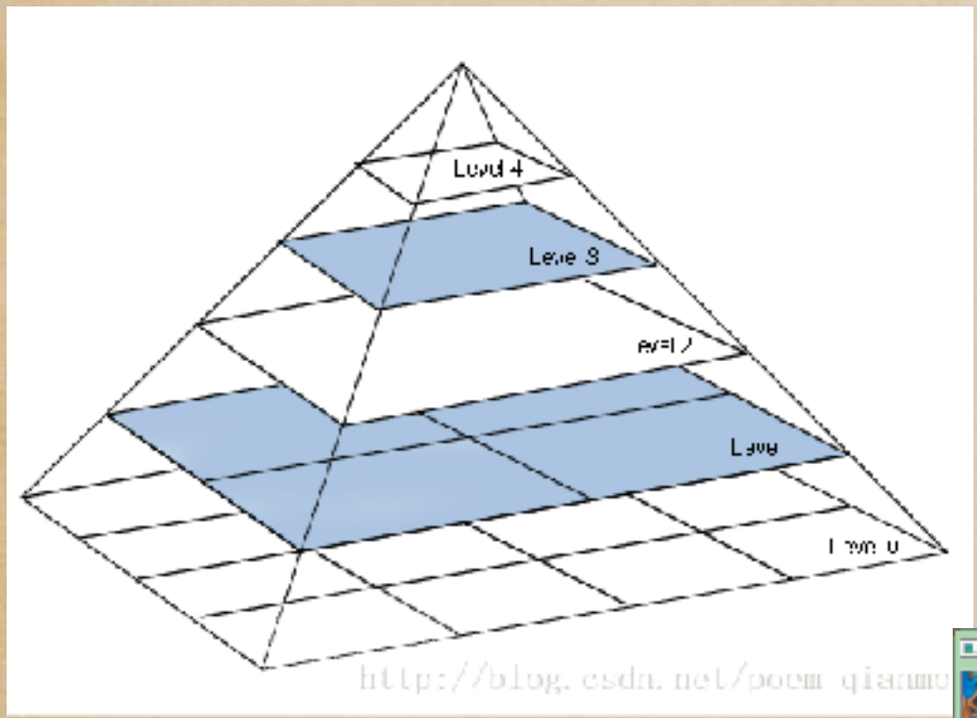
- How you get more information by original picture?
- ResNet backbone
- ConvNets
- COCO
- 6 FPS
- in various systems for detection and segmentation
- End-to-end learning on multiscale feature pyramid

SSD, FPN, RetinaNet, YOLOv3

(multi scale, small object, FG-BG, end-to-end)

- learn from region based object detectors
- learn from single shot object detectors
- Used trained network
- Multiple scale for multiple feature maps
- Reduce resolution/spatial dimension with structure for the semantic value
- prediction

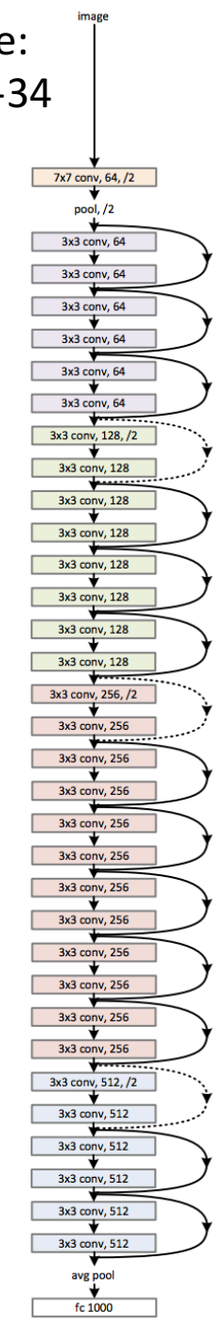
FPN, SIFT, HOG


Level 4
Level 3
Level 0
http://blog.csdn.net/poem qianmu

【原图】
【效果图】之一

1. Backbone ConvNet

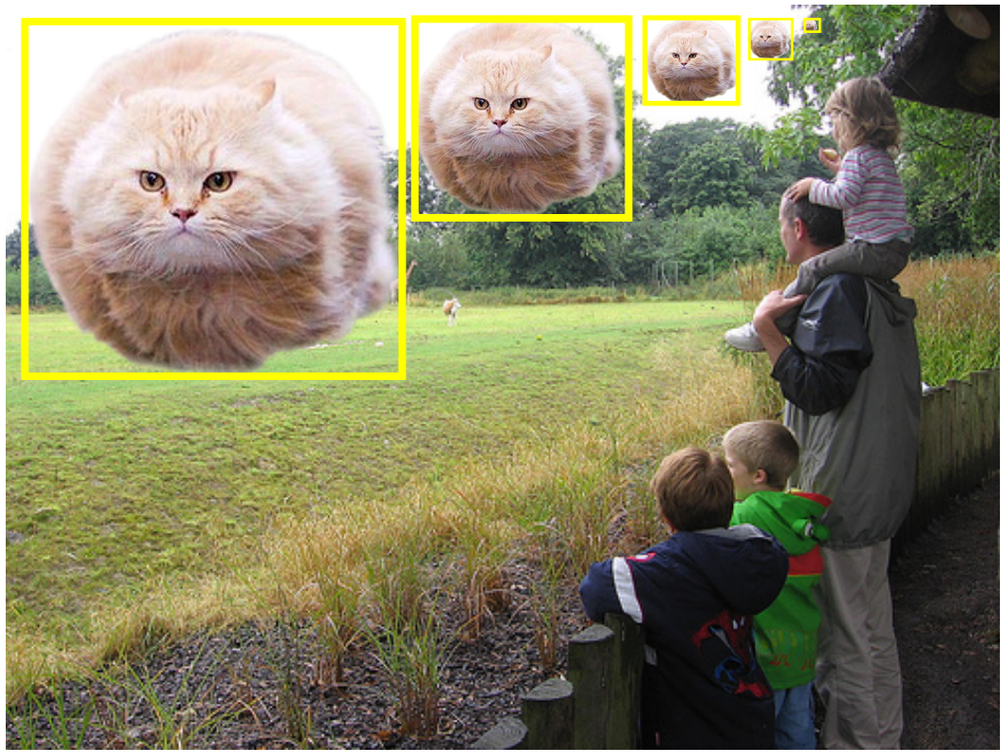
Example:
ResNet-34

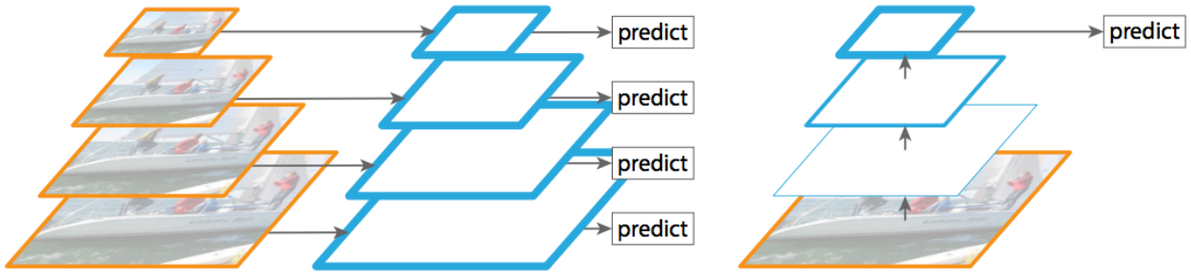


Use any standard ConvNet as a “backbone architecture”

- AlexNet, VGG, ResNet, Inception, Inception-ResNet, ResNeXt (poster 67 Tuesday morning), …
- Use “same” padding everywhere (preserves integer scales)
- Prefer fully convolutional networks (ignoring cls head; see next slide)
- Pre-train on ImageNet classification (or similar)

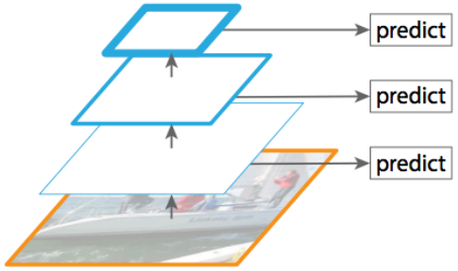
2. Scale Invariant Detection





(a) Featurized image pyramid
SLOW!
(Viola & Jones, HOG, DPM, multiscale Fast R-CNN, ...)

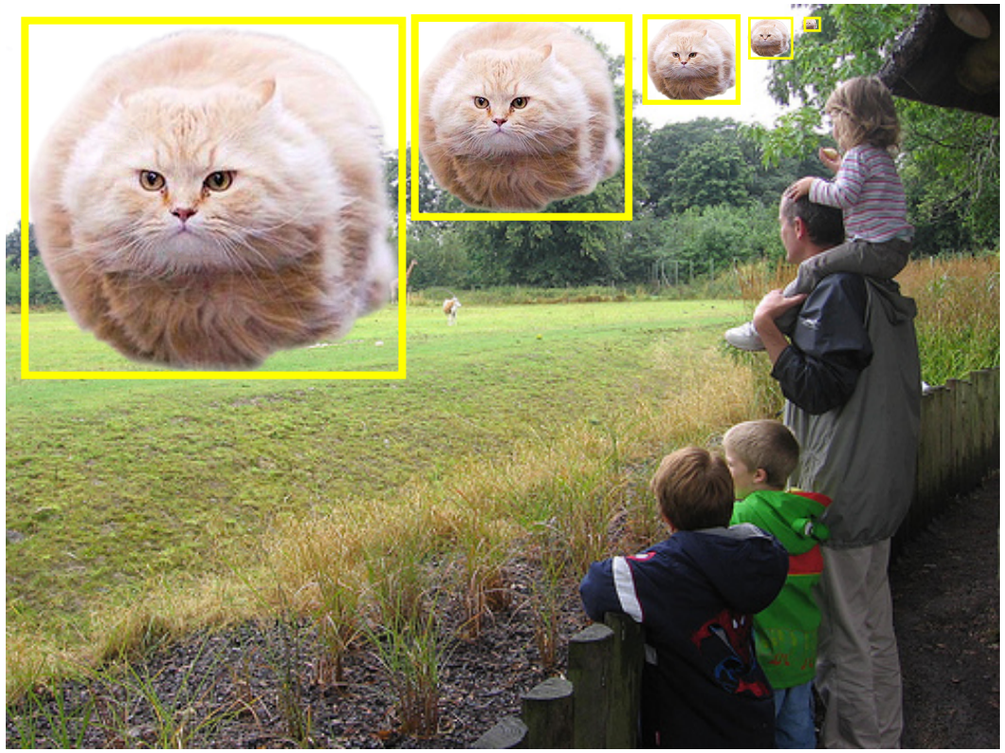
(b) Single feature map
Do nothing; give up; Fast
(Fast R-CNN, YOLO, ...)

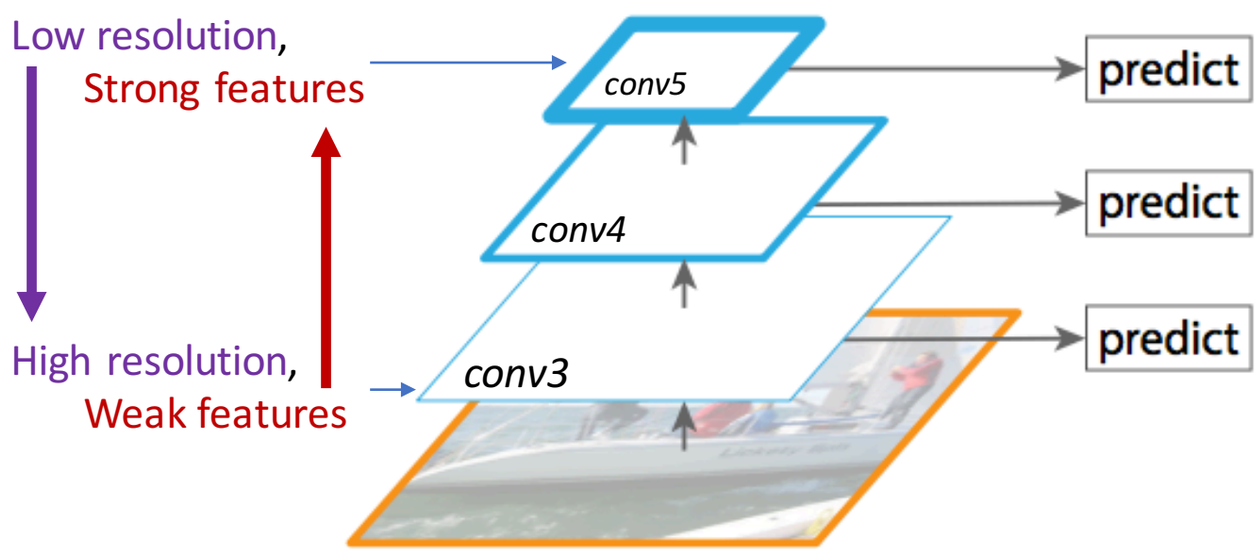


(c) Pyramidal feature hierarchy
Fast!
(≈ SSD, ...)

Popular strategies for detecting objects over large scale changes

Compromise Feature Quality, but Fast ("Free")

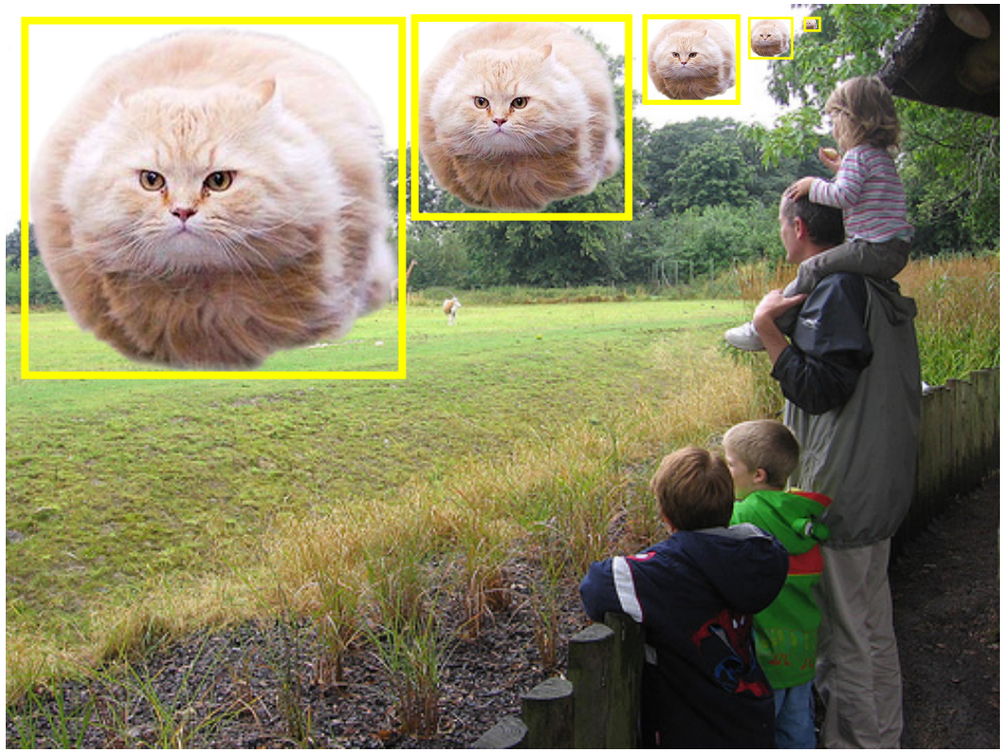


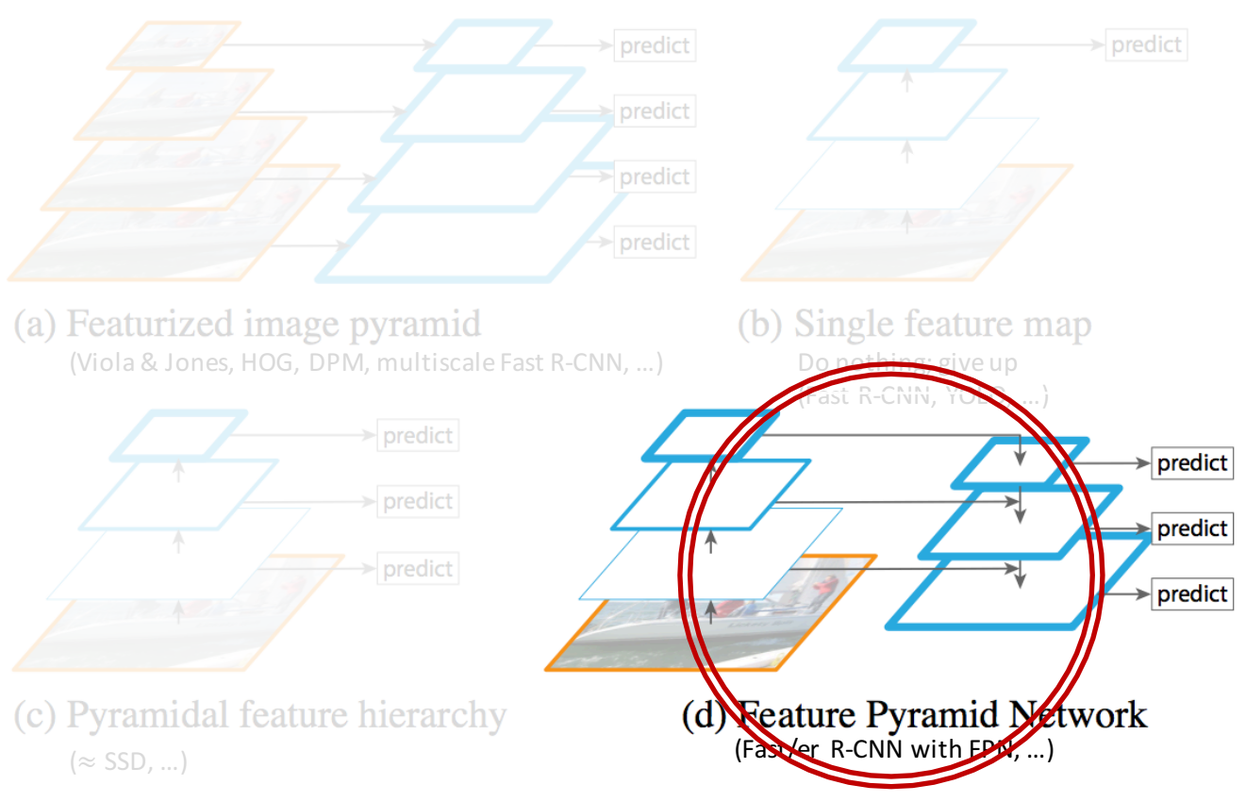


The "native" in-network feature pyramid poses an inherent tradeoff

2. … + Feature Pyramid Network (FPN)

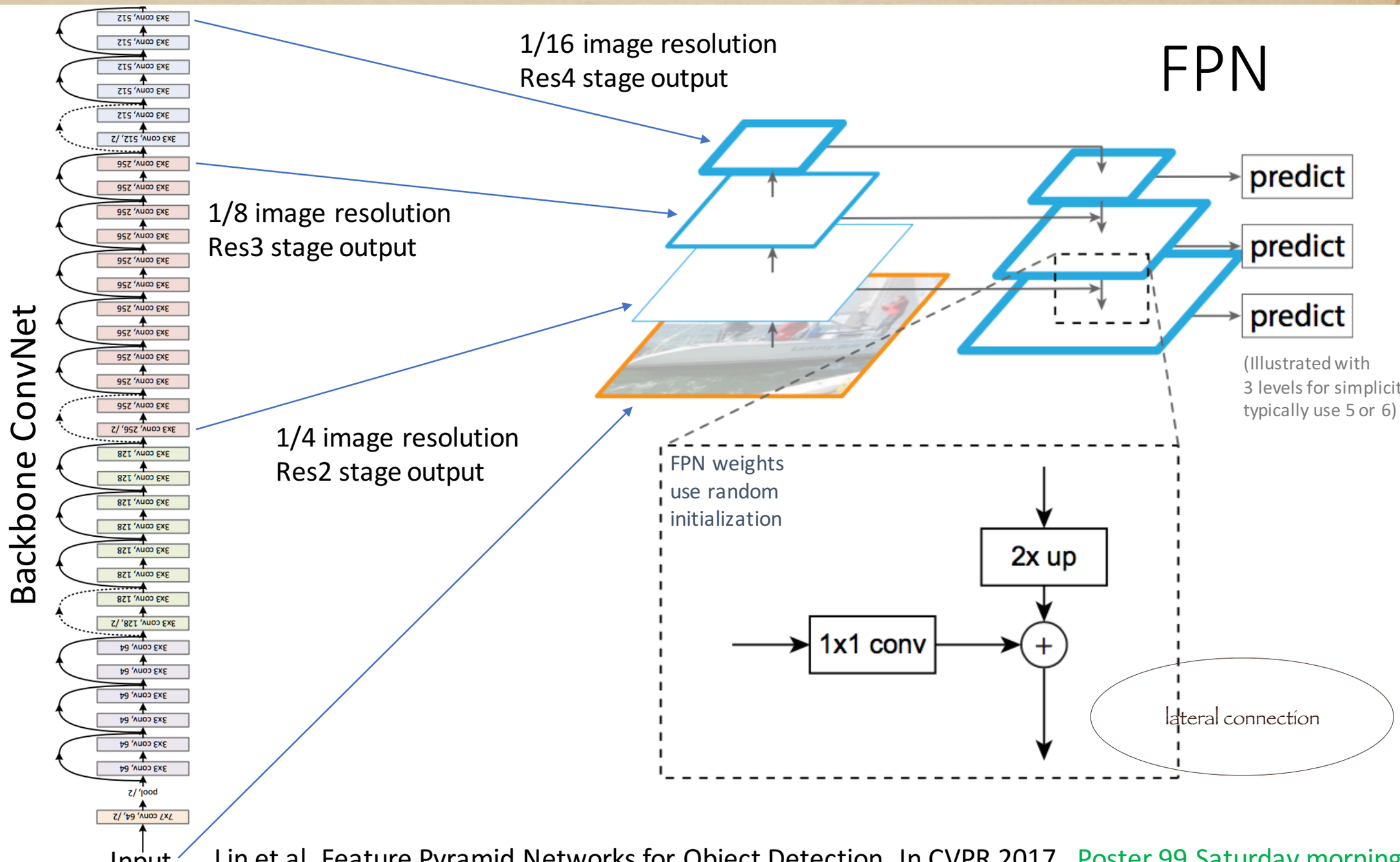
[Optional, but recommended]





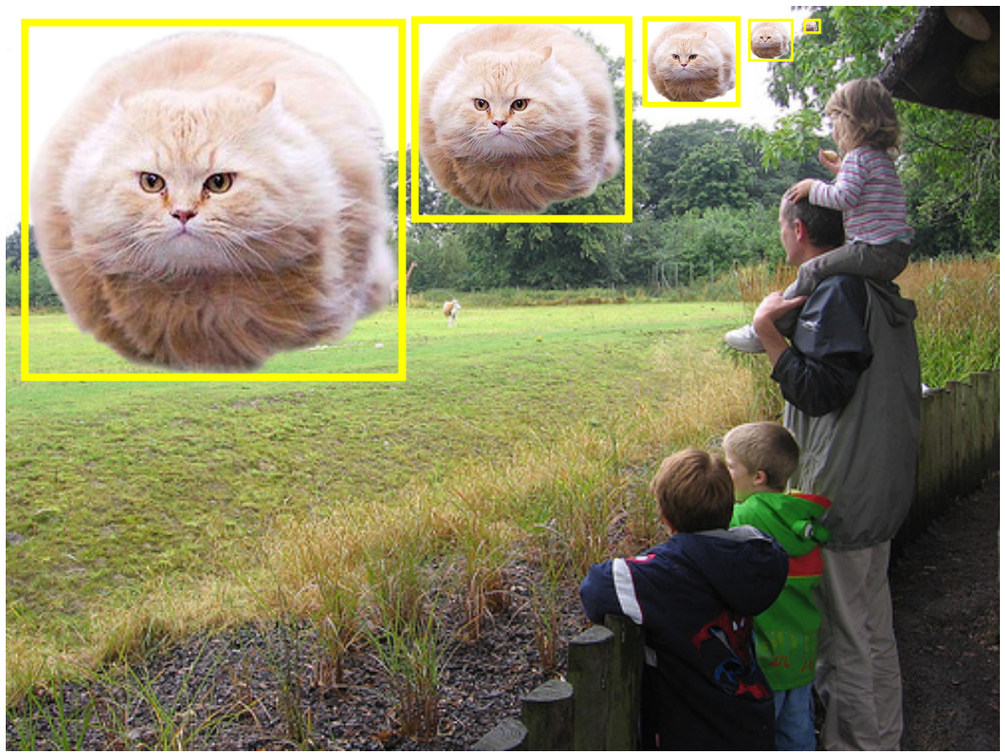
Lin et al. Feature Pyramid Networks for Object Detection. In CVPR 2017. Poster 99 Saturday morning. See also: Shrivastava's TDM

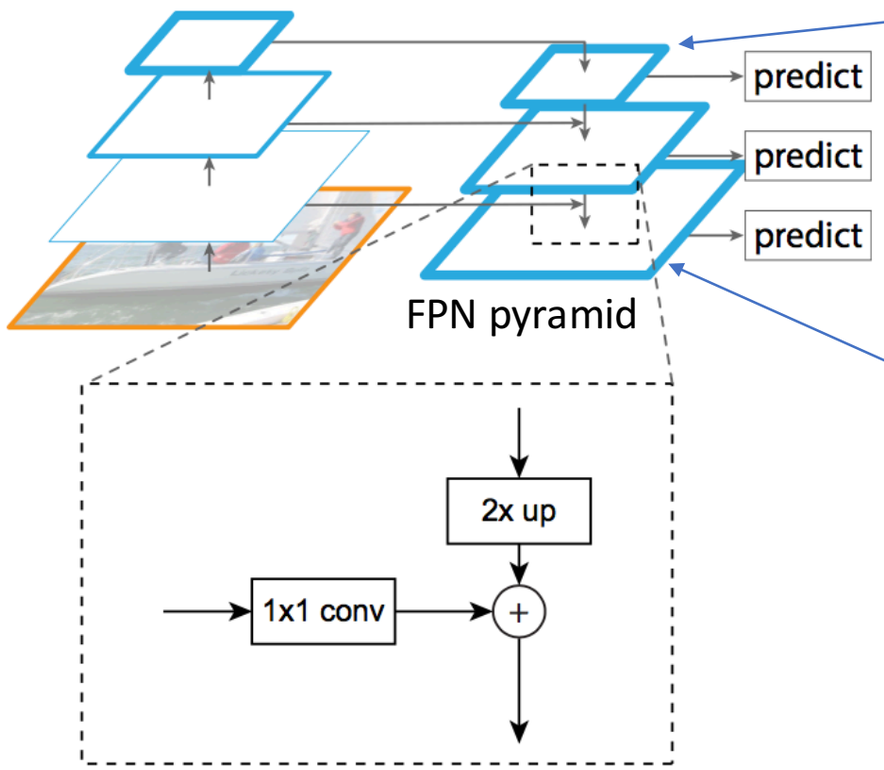
FPN



Lin et al. Feature Pyramid Networks for Object Detection. In CVPR 2017. Poster 99 Saturday morning.

No Compromise on Feature Quality, still Fast



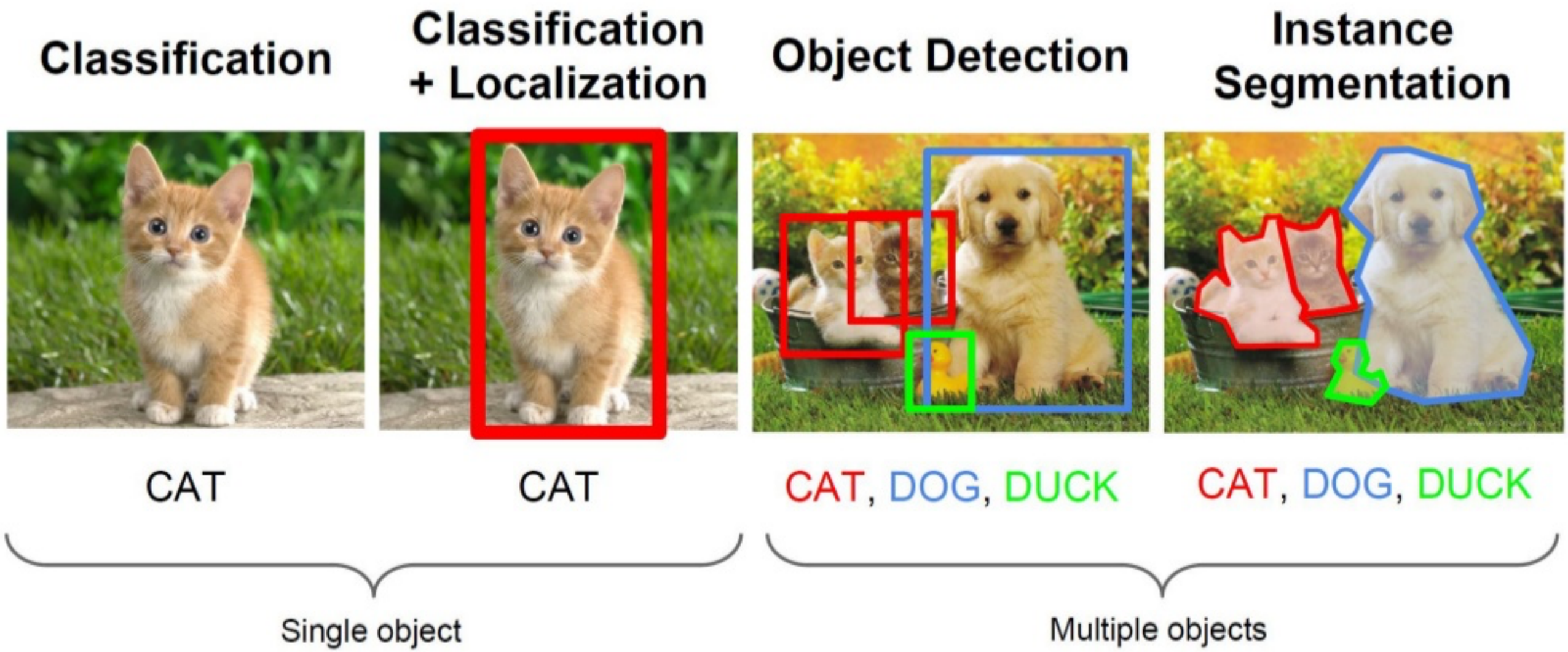


Low resolution,
Strong features

High resolution,
Strong features

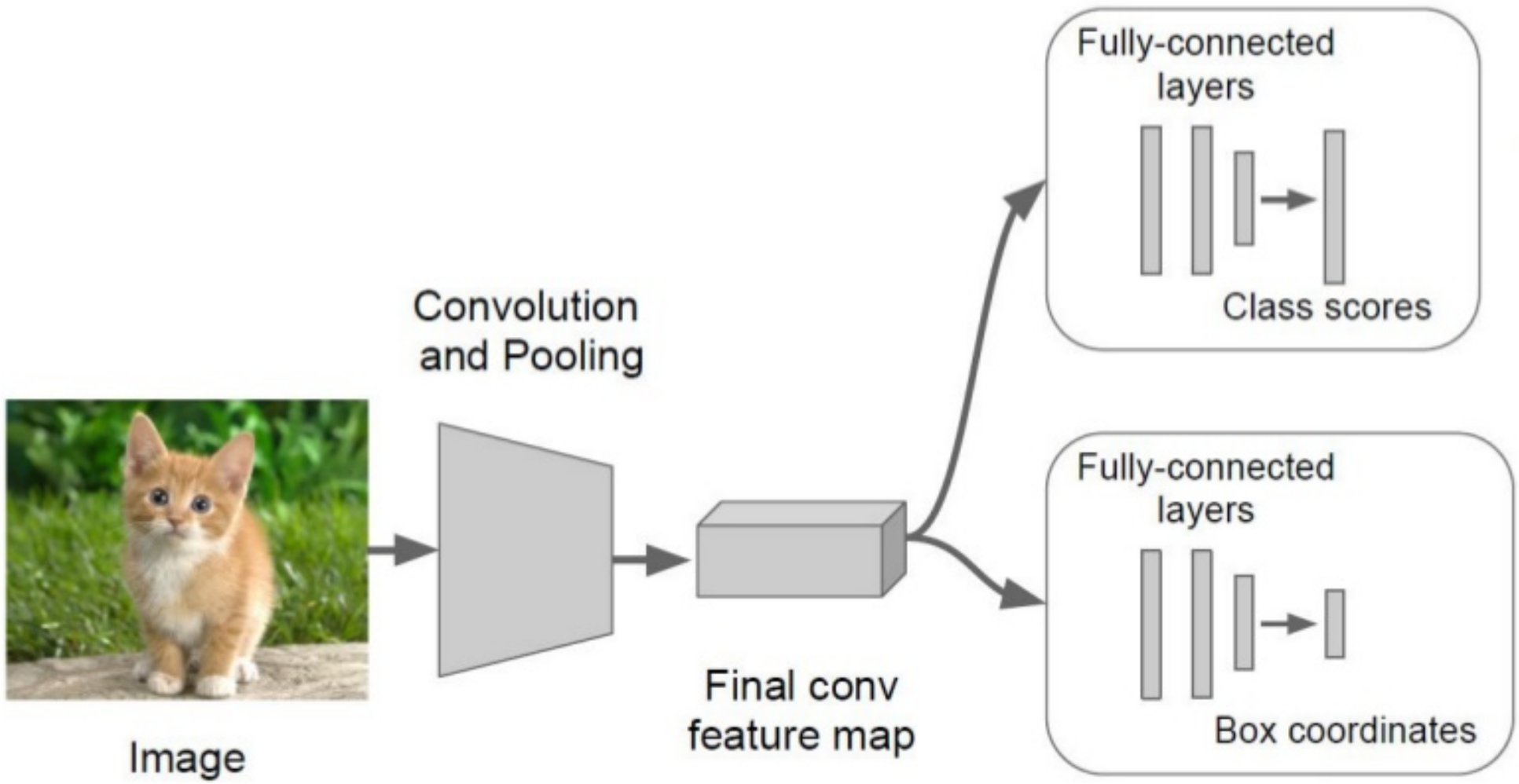
Lin et al. Feature Pyramid Networks for Object Detection. In CVPR 2017. Poster 99 Saturday morning. See also: Shrivastava's TDM

Computer Vision Tasks



Source: CS231n Object detection http://cs231n.stanford.edu/slides/2016/winter1516_lecture8.pdf

Classification + Localization: Model



Classification Head:

- C Scores for C classes

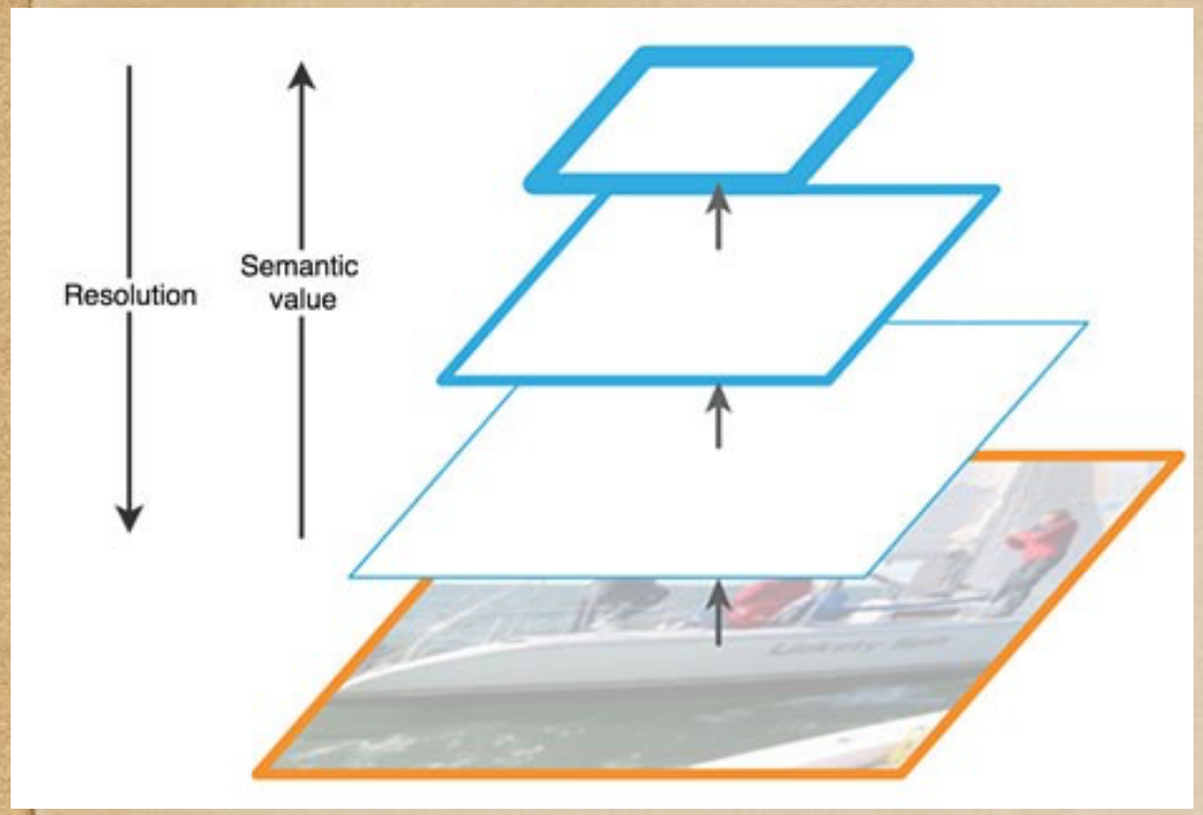
Localization Head:

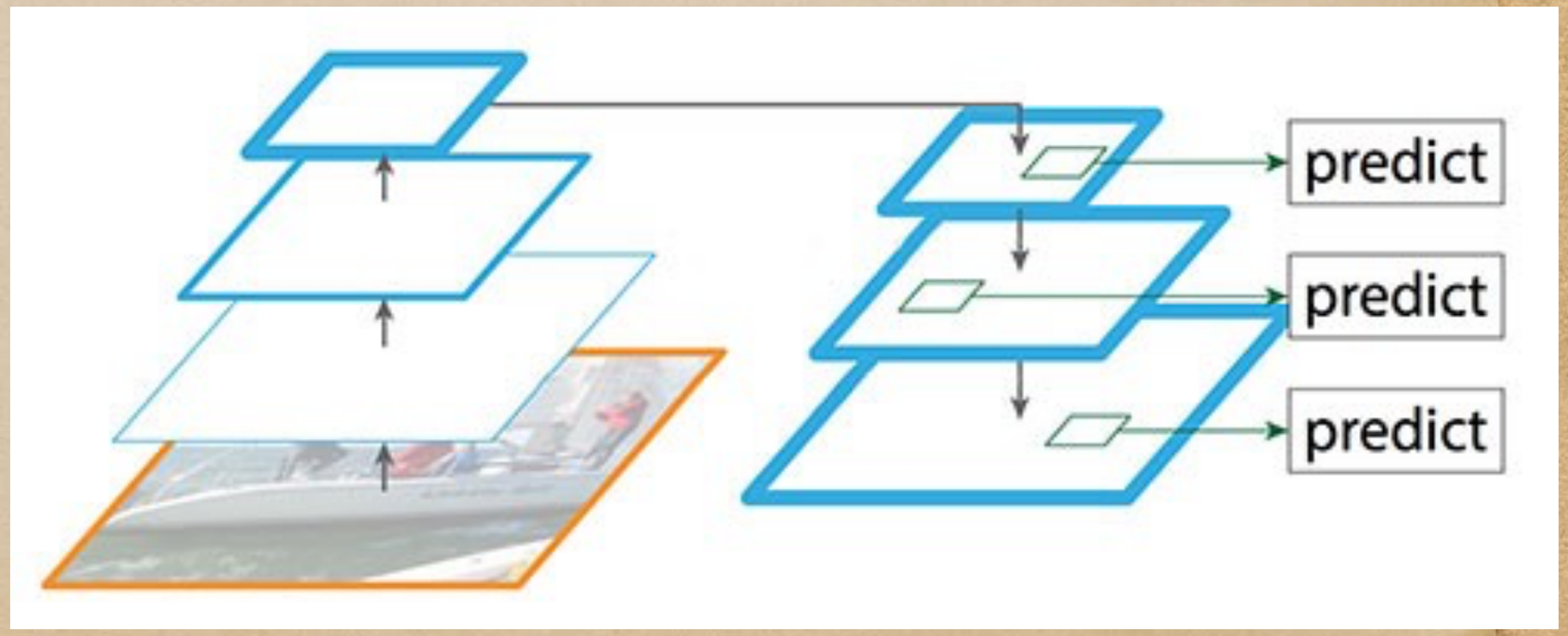
- Class agnostic: (x,y,w,h)
- Class specific: (x,y,w,h) X C

Object Detection: Datasets

2007	2013	2015
Pascal VOC	ImageNet ILSVRC	MS COCO
• 20 Classes • 11K Training images • 27K Training objects	• 200 Classes • 476K Training images • 534K Training objects	• 80 Classes • 200K Training images • 1.5M Training objects
Was de-facto standard, currently used as quick benchmark to evaluate new detection algorithms.	Essentially scaled up version of PASCAL VOC, similar object statistics.	More categories and more object instances in every image. Only 10% of images contain a single object category, 60% in Pascal. More small objects than large objects.

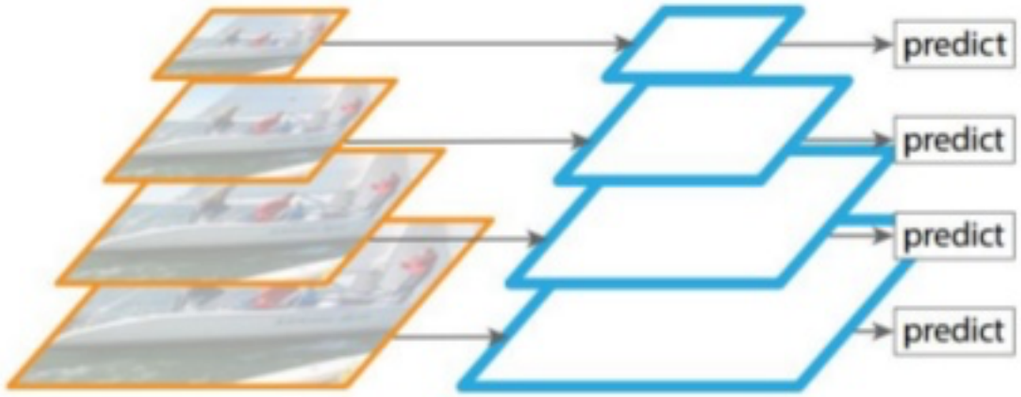
COCO Examples


Resolution
Semantic value


predict
predict
predict

Multi-Scale Object Detection

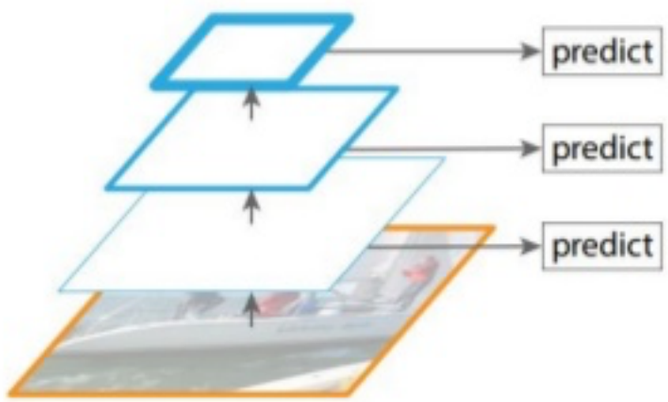
- Multi-scale object detection using image pyramid
  - Predict different scales by applying same model at different image resolutions.
- Classic method.
- But also, in OverFeat.
- Slow. Requires multiple evaluation of the same model.



(a) Featurized image pyramid

Multi-Scale Object Detection

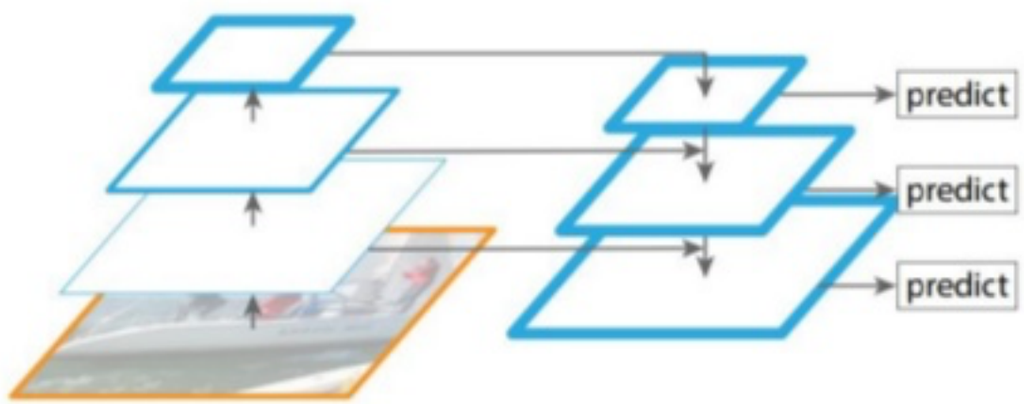
- Predict different object sizes at different feature scales.
- Same as SSD.
- Good features resolution for small objects
- But features are much weaker than in deeper layers.



(c) Pyramidal feature hierarchy

Feature Pyramid Network (FPN)

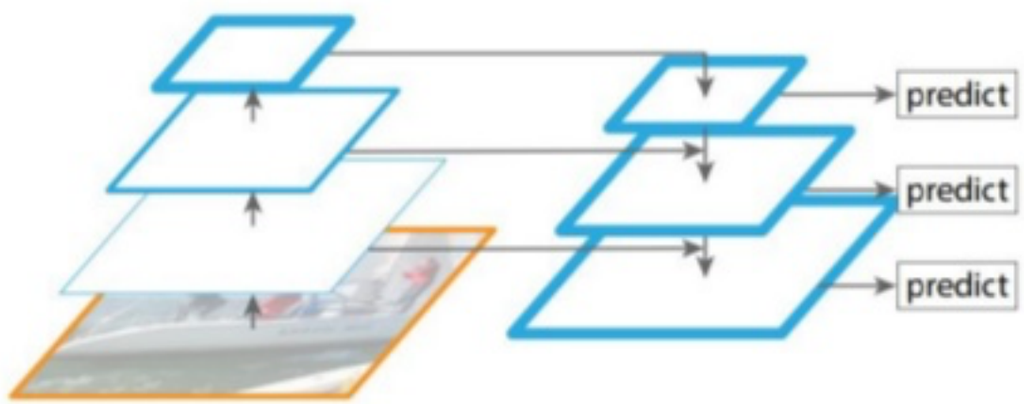
- Single model (same in training as in testing).
- Good features resolution for small objects.
- Strong features in all layers.
- Almost no overhead over SSD (= Fast).



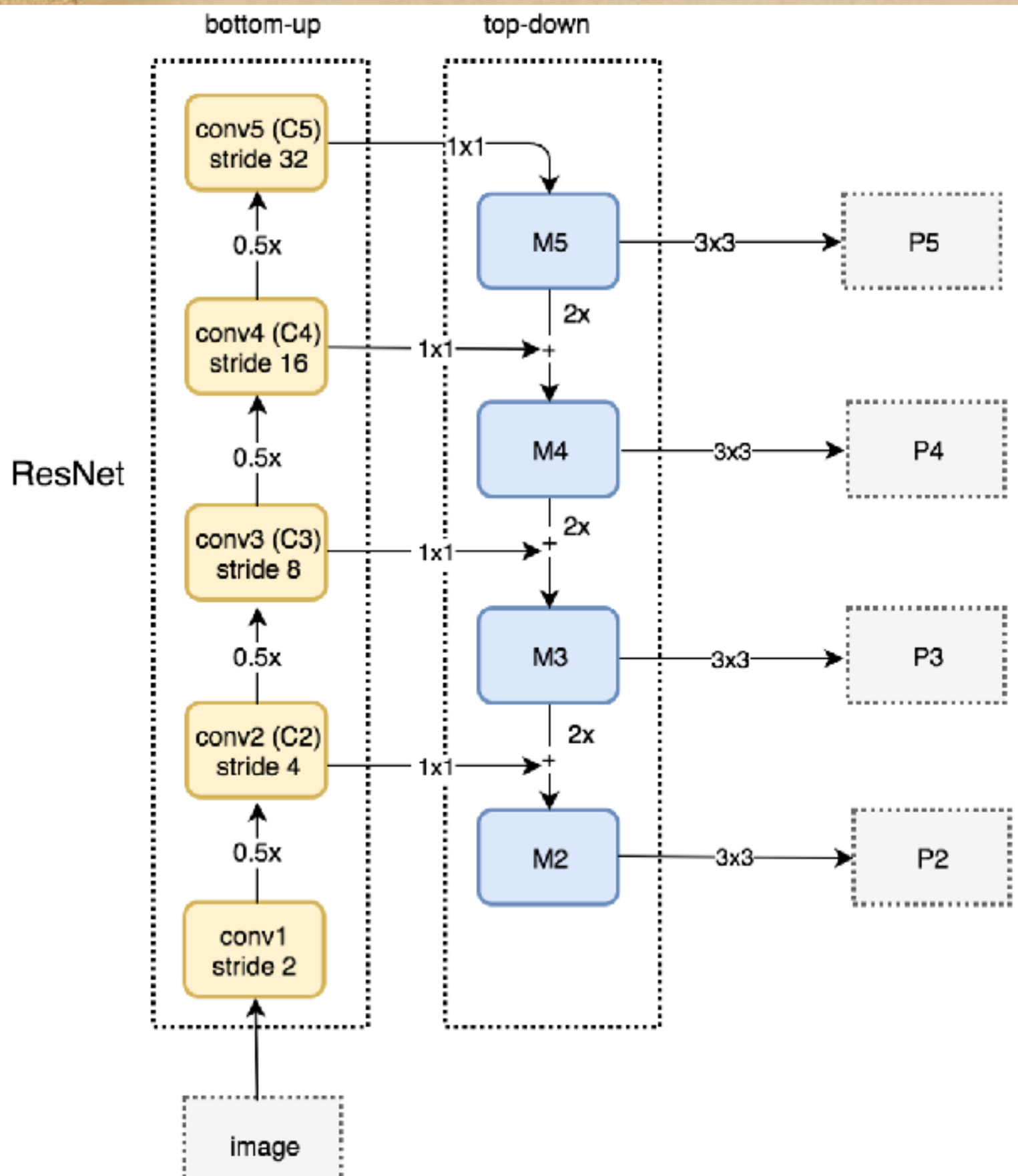
(d) Feature Pyramid Network

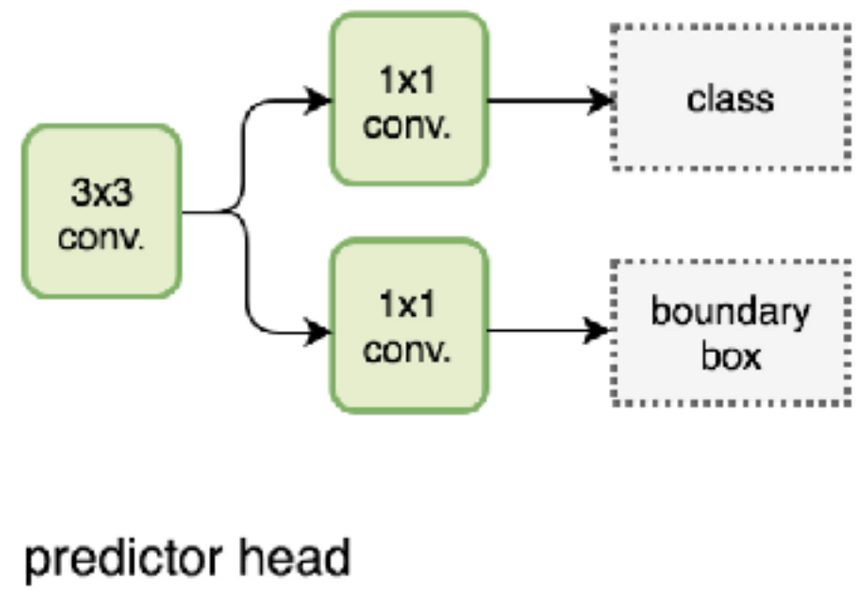
Feature Pyramid Network (FPN)

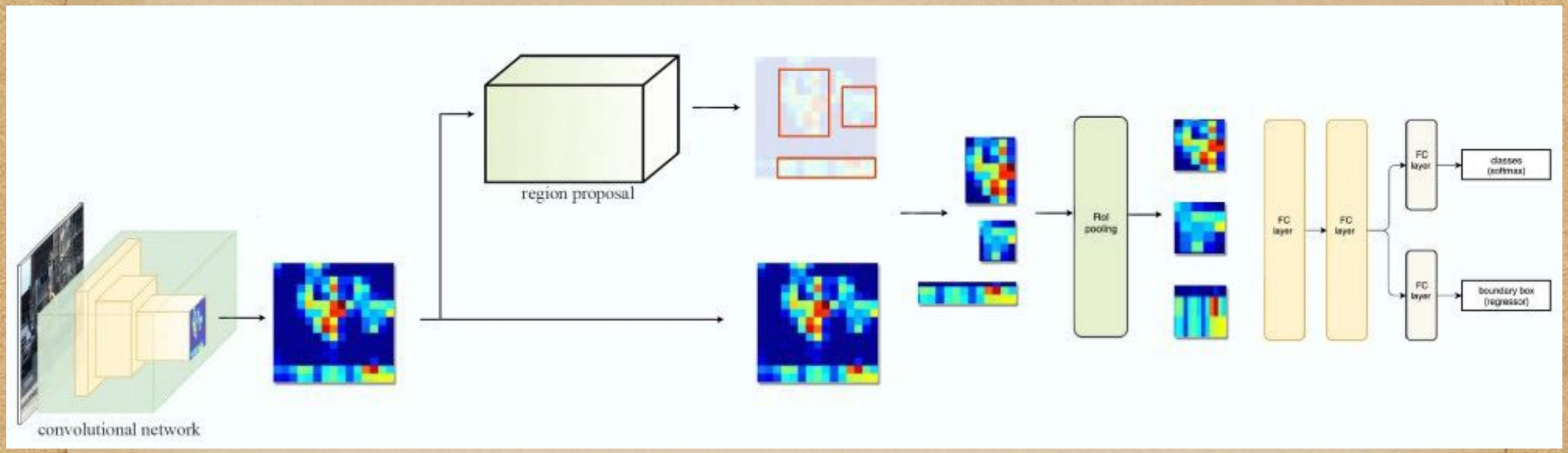
- Single model (same in training as in testing).
- Good features resolution for small objects.
- Strong features in all layers.
- Almost no overhead over SSD (= Fast).

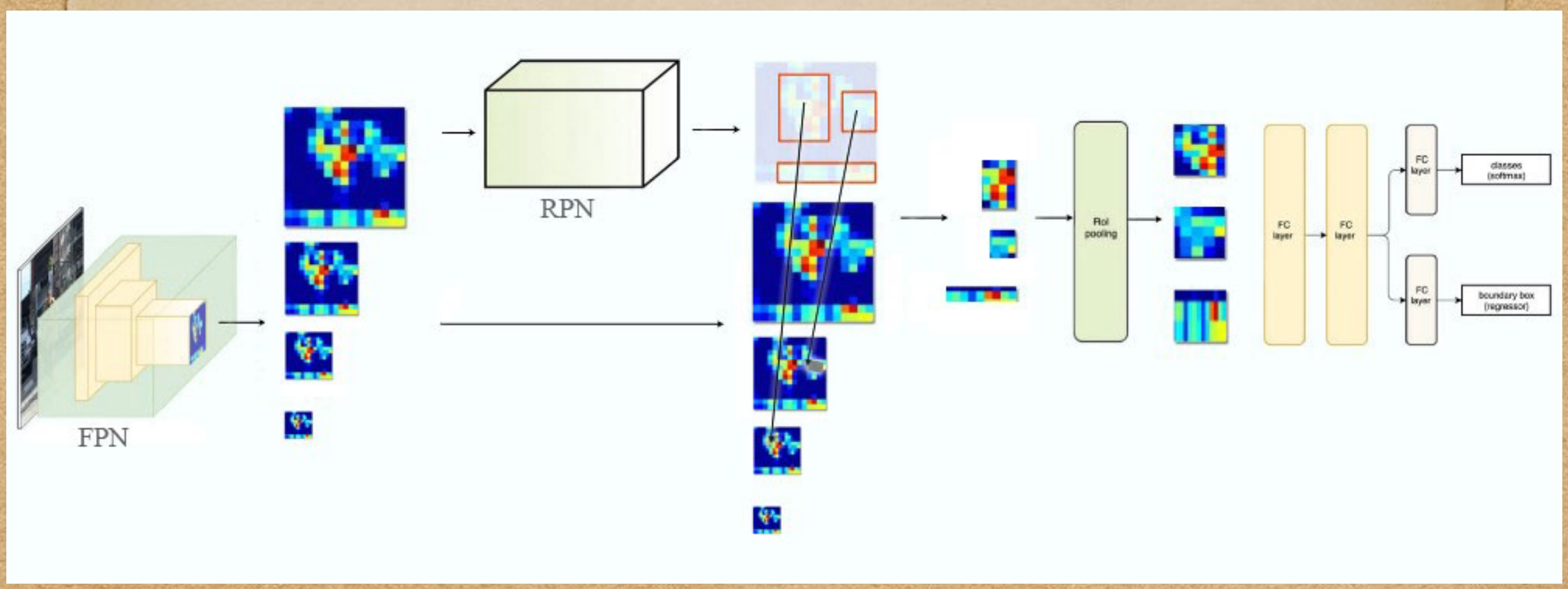


(d) Feature Pyramid Network


bottom-up
top-down
ResNet
conv5 (C5)
stride 32
0.5x
conv4 (C4)
stride 16
0.5x
conv3 (C3)
stride 8
0.5x
conv2 (C2)
stride 4
0.5x
conv1
stride 2
image
1x1
1x1
1x1
1x1
M5
M4
M3
M2
2x
2x
2x
3x3
3x3
3x3
3x3
P5
P4
P3
P2

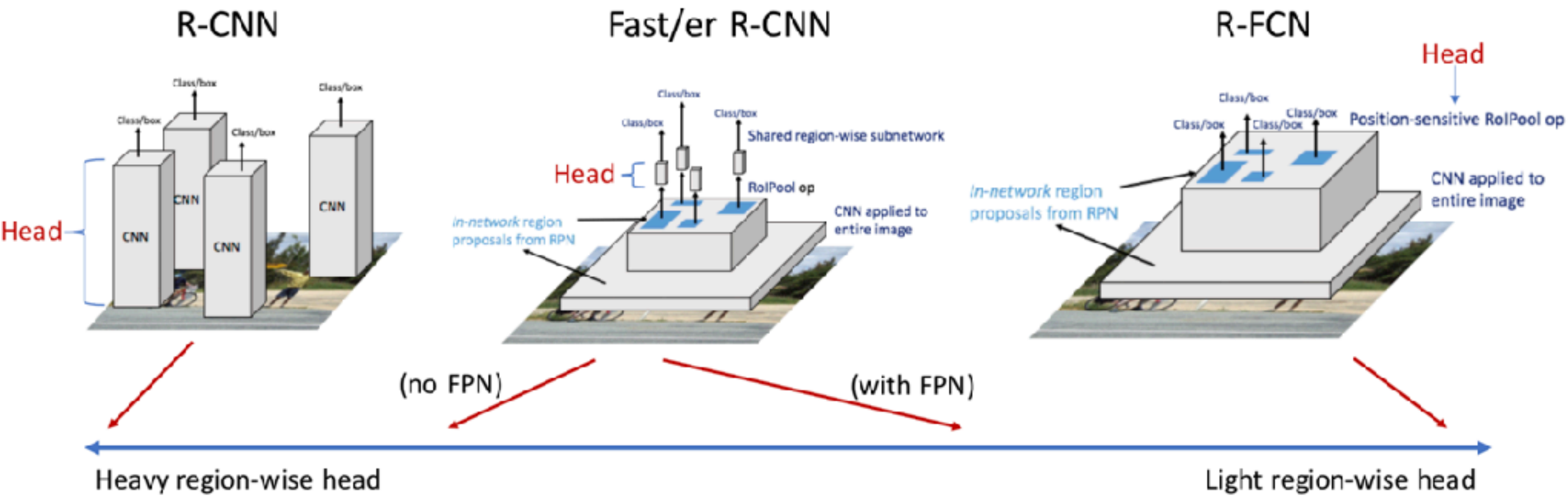

3x3
conv.
1x1
conv.
1x1
conv.
class
boundary
box
predictor head


region proposal
convolutional network
RoI pooling
FC layer
FC layer
FC layer
classes (softmax)
FC layer
boundary box (regressor)


RPN
FPN
RoI pooling
FC layer
FC layer
FC layer
classes (softmax)
FC layer
boundary box (regressor)

Networks on Convolutional Feature Maps

Shifting computation between the "head" and the "trunk"



RoIs of different scales to the pyramid levels

feature pyramid as if it were produced from an image pyramid

RoI of width w and height h (on the input image to the network) to the level P_k of our feature pyramid by:

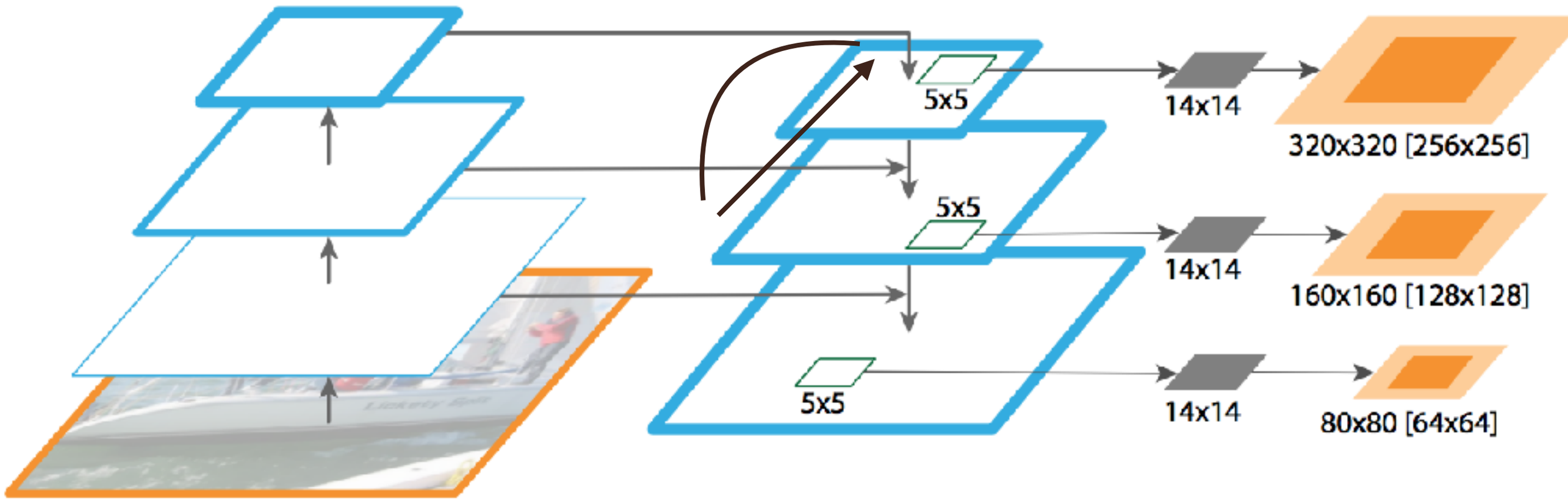
$$k = \lfloor k_0 + \log_2(\sqrt{wh}/224) \rfloor.$$

224 is the canonical ImageNet pre-training size

k_0 is the target level on which an RoI with $w \times h = 224^2$

Segmentation

Mask for instance segmentation



Feature Pyramid Network (FPN)

- How important is top-down enrichment?
- How important are lateral connections?
- How important are pyramid representations?

RPN	feature	# anchors	lateral?	top-down?	AR^{100}	AR^{1k}	AR^{1k}_s	AR^{1k}_m	AR^{1k}_l
(a) baseline on conv4	C_4	47k			36.1	48.3	32.0	58.7	62.2
(b) baseline on conv5	C_5	12k			36.3	44.9	25.3	55.5	64.2
(c) **FPN**	$\{P_k\}$	200k	✓	✓	**44.0**	**56.3**	**44.9**	**63.4**	66.2
Ablation experiments follow:									
(d) bottom-up pyramid	$\{P_k\}$	200k	✓		37.4	49.5	30.5	59.9	**68.0**
(e) top-down pyramid, w/o lateral	$\{P_k\}$	200k		✓	34.5	46.1	26.5	57.4	64.7
(f) only finest level	P_2	750k	✓	✓	38.4	51.3	35.1	59.7	67.6

Table 1. Bounding box proposal results using RPN [29], evaluated on the COCO `minival` set. All models are trained on `trainval35k`. The columns "lateral" and "top-down" denote the presence of lateral and top-down connections, respectively. The column "feature" denotes the feature maps on which the heads are attached. All results are based on ResNet-50 and share the same hyper-parameters.

Feature Pyramid Network (FPN)

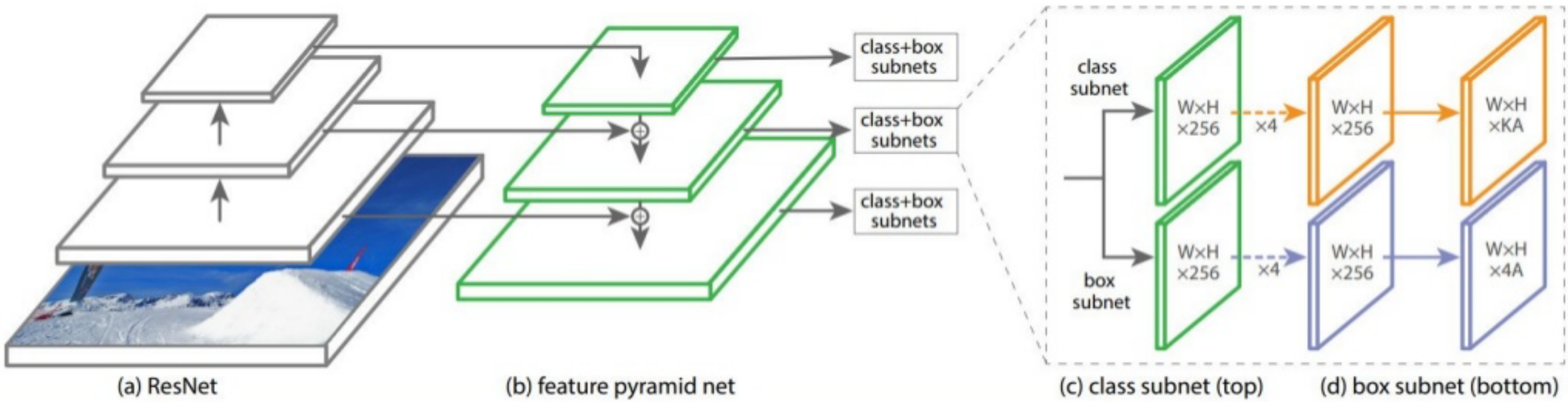
- How important is top-down enrichment?
- How important are lateral connections?
- How important are pyramid representations?

Fast R-CNN	proposals	feature	head	lateral?	top-down?	AP@0.5	AP	AP_s	AP_m	AP_l
(a) baseline on conv4	RPN, $\{P_k\}$	C_4	conv5			54.7	31.9	15.7	36.5	45.5
(b) baseline on conv5	RPN, $\{P_k\}$	C_5	2*fc*			52.9	28.8	11.9	32.4	43.4
(c) **FPN**	RPN, $\{P_k\}$	$\{P_k\}$	2*fc*	✓	✓	**56.9**	**33.9**	**17.8**	**37.7**	**45.8**
Ablation experiments follow:										
(d) bottom-up pyramid	RPN, $\{P_k\}$	$\{P_k\}$	2*fc*	✓		44.9	24.9	10.9	24.4	38.5
(e) top-down pyramid, w/o lateral	RPN, $\{P_k\}$	$\{P_k\}$	2*fc*		✓	54.0	31.3	13.3	35.2	45.3
(f) only finest level	RPN, $\{P_k\}$	P_2	2*fc*	✓	✓	56.3	33.4	17.3	37.3	45.6

Table 2. Object detection results using **Fast R-CNN** [11] on *a fixed set of proposals* (RPN, $\{P_k\}$, Table 1(c)), evaluated on the COCO `minival` set. Models are trained on the `trainval35k` set. All results are based on ResNet-50 and share the same hyper-parameters.

Focal Loss for Dense Object Detection

- Can we train a single stage detector to be as accurate as two stage detectors?
- Contributions:
  - RetinaNet: Single stage object detector based on FPN backbone.
  - New loss.



FPN

- End-to-end learning on multi-scale feature pyramid
- ROI mask, prediction
- IOU
- Header through pyramid for parameters sharing and prediction
- Segmentation with lateral/skip connection
- DeepMask/SharpMask were trained on image crops for predicting instance segments and object/non-object scores.
- Region Proposal + pyramid representations
-

In-network design

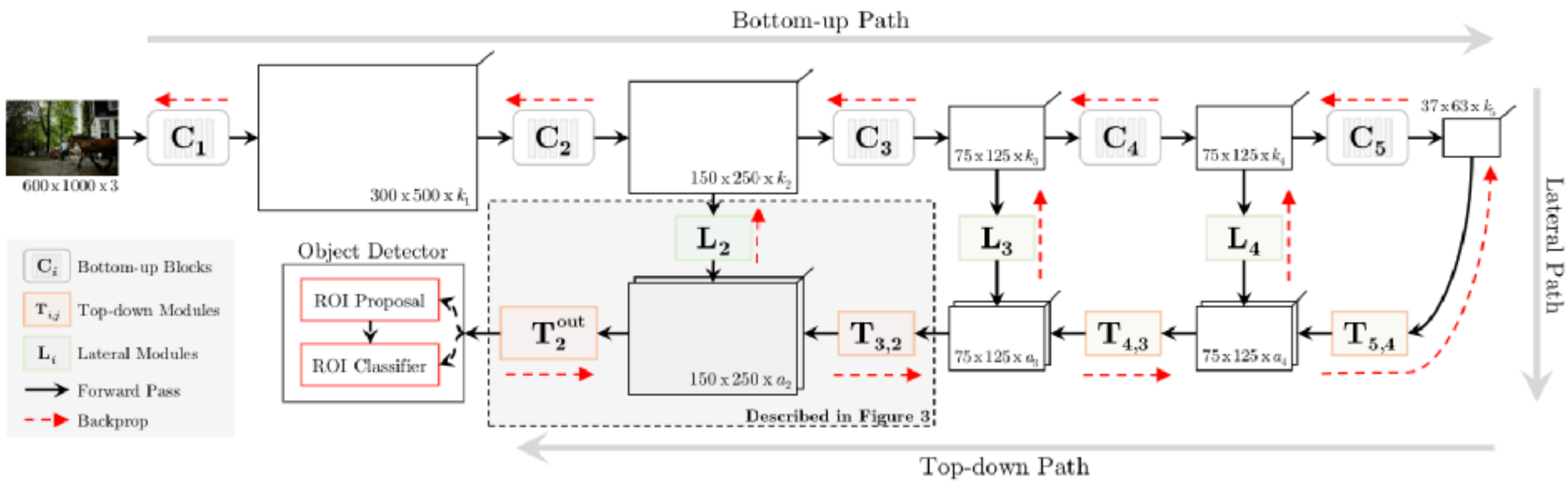


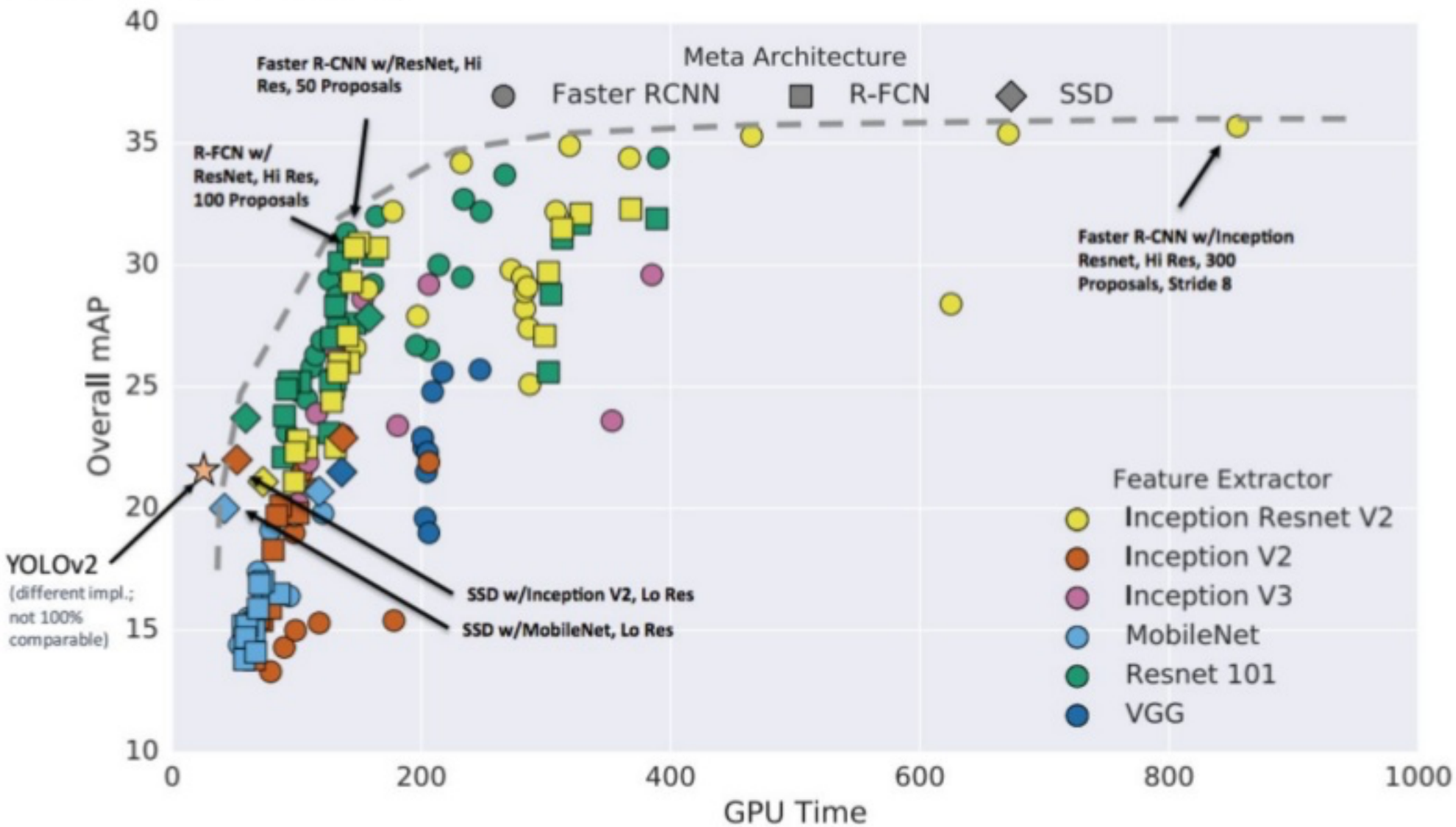
Figure 2. The illustration shows an example of **Top-Down Modulation** (TDM) Network, which is integrated with the bottom-up network with lateral connections. $\mathbf{C}_i$ are bottom-up, feedforward feature blocks, $\mathbf{L}_i$ are the lateral modules which transform low level features for the top-down contextual pathway. Finally, $\mathbf{T}_{j,i}$, which represent flow of top-down information from index j to i. Individual components are explained in Figure 3 and 4.

In-network design

(Faster R-CNN on ResNet)

- Bounding box proposal (AR +8.0)
- Object detection (AP +2.3 COCO)
  (AP + 3.8 PAS CAL)
- Mask proposal on image pyramid
  improve instance segmentation AR and speed
- RPN design for binary classification and bounding box regression
- Network head 3x3 followed by two sibling 1x1, extract features for parameters and for segmentation
- Prediction head to ROI of all levels followed by two FC and ReLU
- DeepMask/SharpMask framework for both training and inference. 5×5 MLP to predict 14×14 masks by FC, Half octaves are handled by an MLP on 7x7.
  MLP=RPN

Speed/accuracy trade-offs for modern convolutional object detectors (Google)



Frameworks

- Caffe
  - Faster R-CNN: https://github.com/rbgirshick/py-faster-rcnn
  - SSD: https://github.com/weiliu89/caffe/tree/ssd
- Tensorflow Object Detection API:
  - https://github.com/tensorflow/models/tree/master/research/object_detection
- Detectron:
  - https://github.com/facebookresearch/Detectron
- Many more re-implementations in different languages...

Honorable mentions

- VGG16: https://arxiv.org/abs/1409.1556
- ResNet: https://arxiv.org/abs/1512.03385
- Inception-ResNet: https://arxiv.org/abs/1602.07261
- ResNeXt: https://arxiv.org/abs/1611.05431
- Xception: https://arxiv.org/abs/1610.02357
- DenseNet: https://arxiv.org/abs/1608.06993
- MobileNet: https://arxiv.org/abs/1704.04861
- SqueezeNet: https://arxiv.org/abs/1602.07360

FIN