Week 7: Detection and instance segmentation

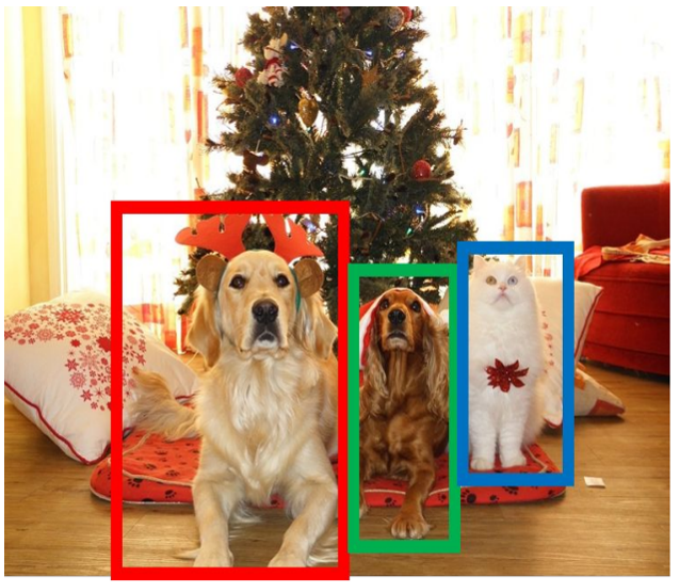
Instructor: Ruixuan Wang
wangruix5@mail.sysu.edu.cn

School of Data and Computer Science
Sun Yat-Sen University

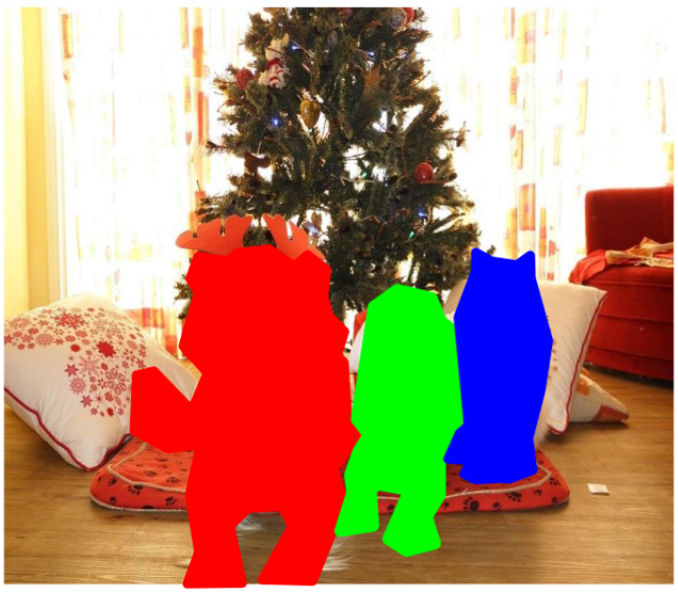
11 April, 2019

Object detection vs. instance segmentation

- Object (instance) detection: classify and localize object instances with bounding boxes
- Instance segmentation: semantically segment each instance
- Applications: self-driving, surveillance, medical image analysis



DOG, DOG, CAT



DOG, DOG, CAT

Figure from Stanford CS231n 2017 lecture 11

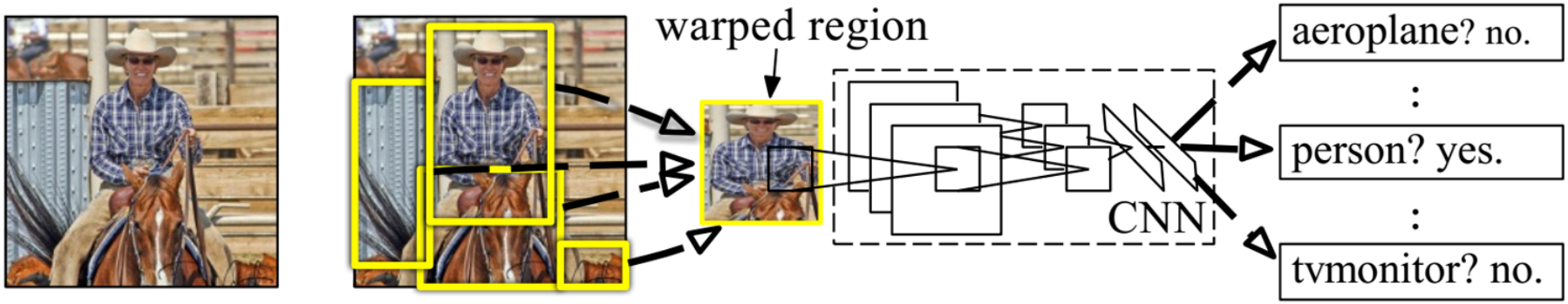
Two-stage approach

1^{st} stage: propose candidate object bounding boxes

2^{nd} stage: classify and fine-tune bounding-boxes

Region-based CNN (R-CNN) for object detection

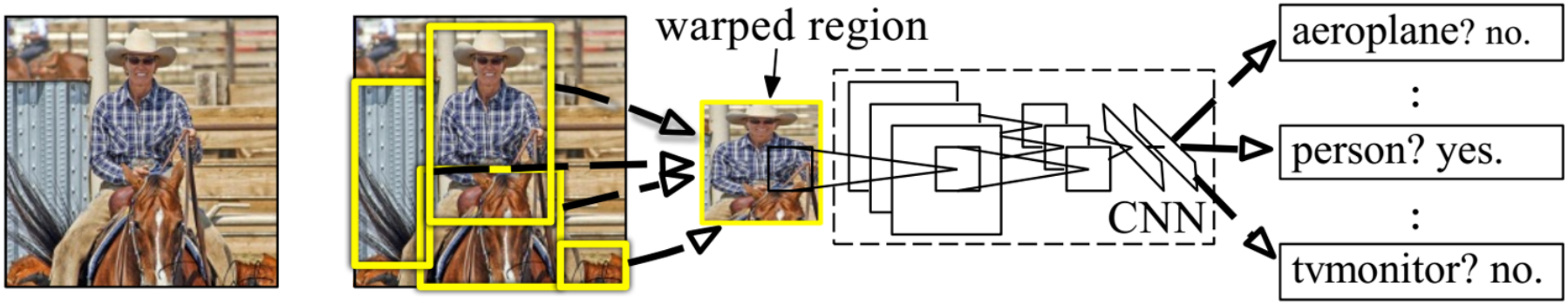
- Use CNN or CNN+SVM classifier for each proposal region
- Fine-tune bounding box location and size with regression
  $\triangle (x, y, w, h) = regressor(initial\ box)$
- Number of model output: $C + 4$ (for each box proposal)



Figures here and in the next two slides are from He's ppt "Deep learning gets way deeper - recent advances of deep learning for computer vision", 2016

Region-based CNN (R-CNN) for object detection

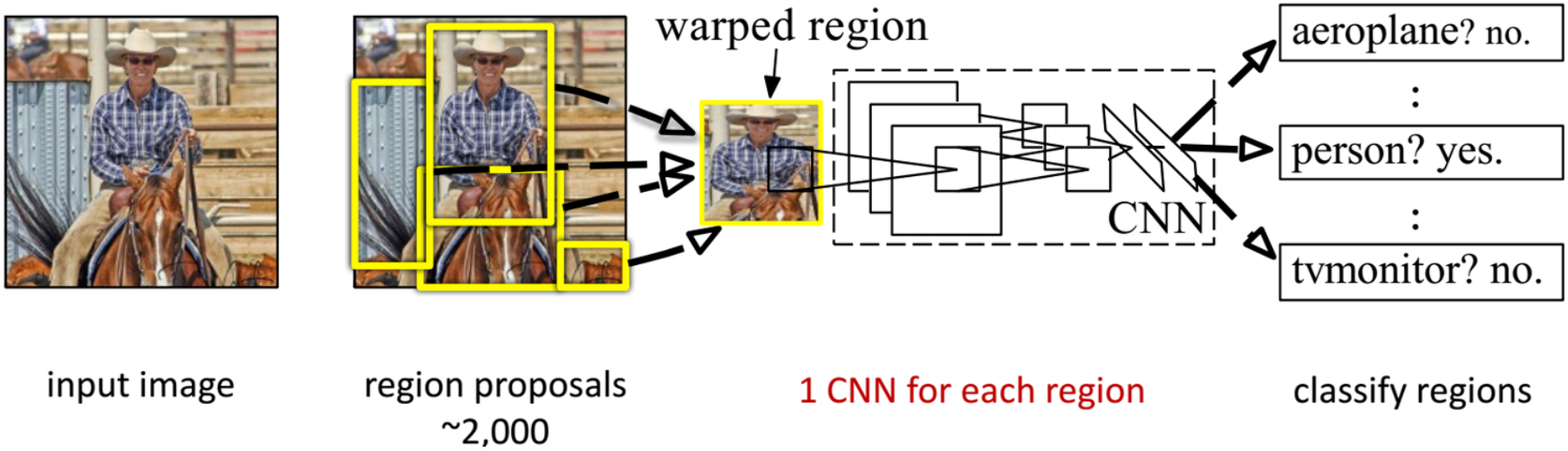
- Use CNN or CNN+SVM classifier for each proposal region
- Fine-tune bounding box location and size with regression
  $\triangle (x, y, w, h) = regressor(initial\ box)$
- Number of model output: $C + 4$ (for each box proposal)



Figures here and in the next two slides are from He's ppt "Deep learning gets way deeper - recent advances of deep learning for computer vision", 2016

Region-based CNN (R-CNN) for object detection

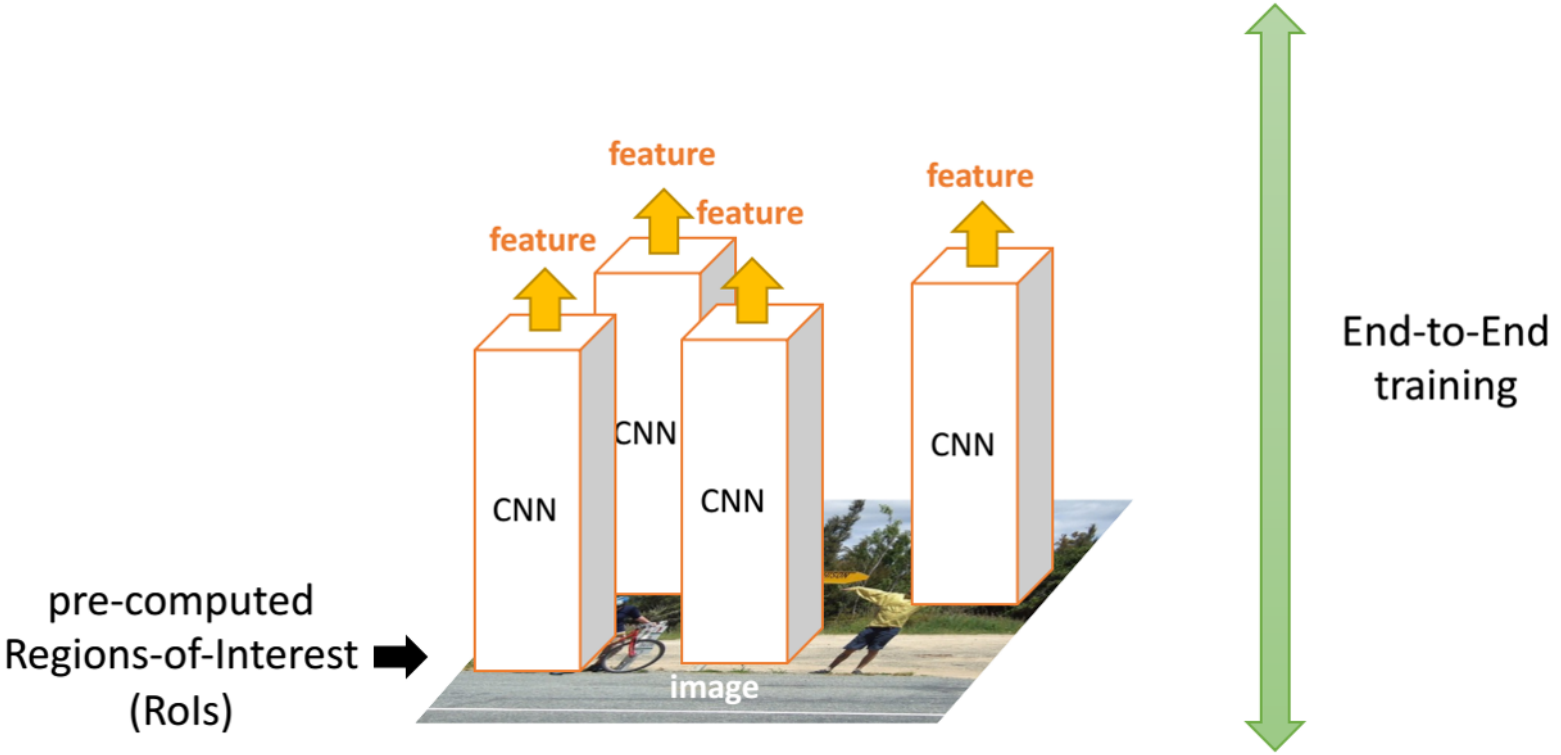
- Use CNN or CNN+SVM classifier for each proposal region
- Fine-tune bounding box location and size with regression
  $\triangle\ (x, y, w, h) = regressor(initial\ box)$
- Number of model output: $C + 4$ (for each box proposal)



Figures here and in the next two slides are from He's ppt "Deep learning gets way deeper - recent advances of deep learning for computer vision", 2016

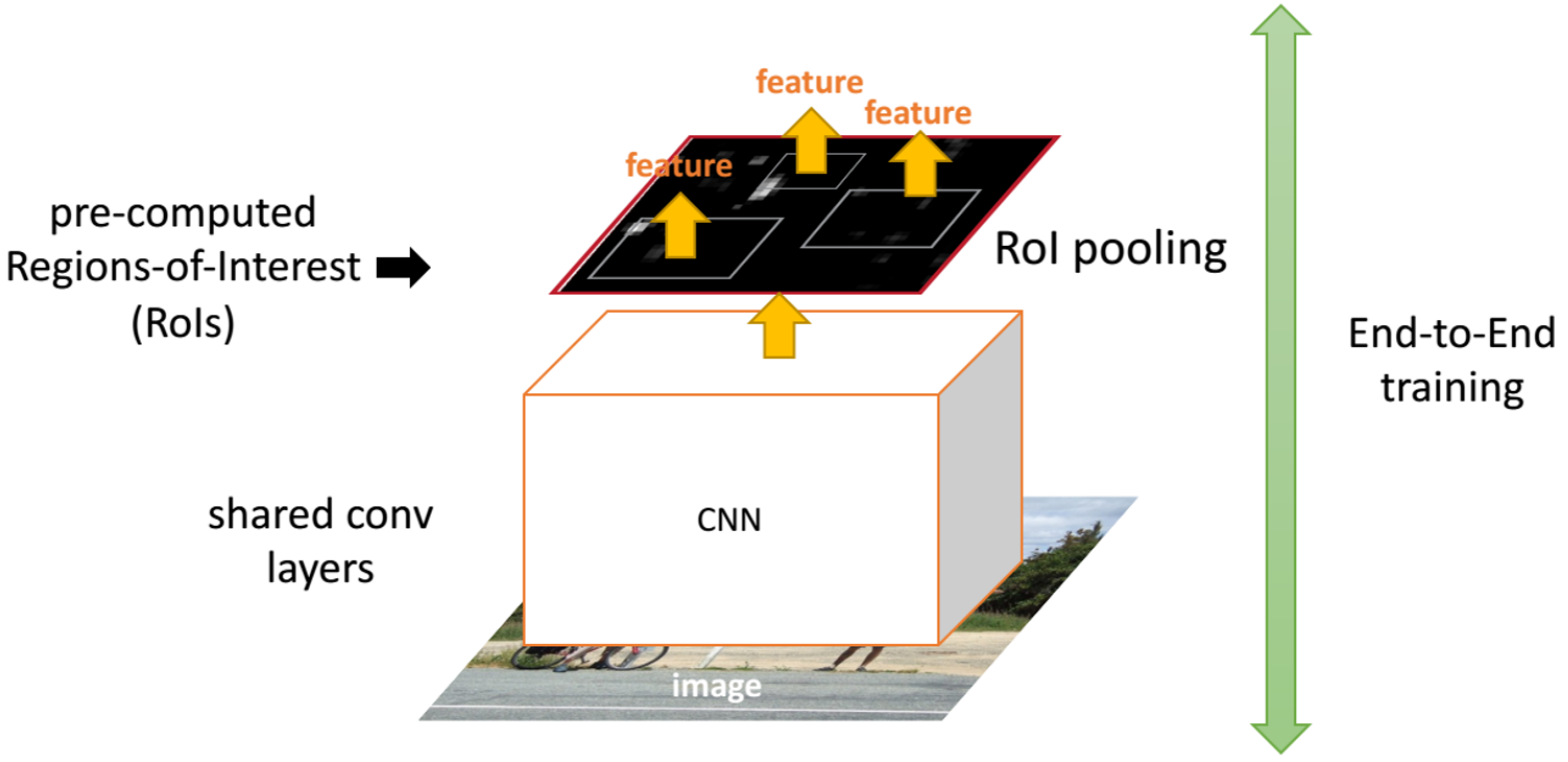
R-CNN (cont')

- Region proposals are pre-computed for each image
- Same CNN is applied multiple times, once for each proposal
- So, slow inference



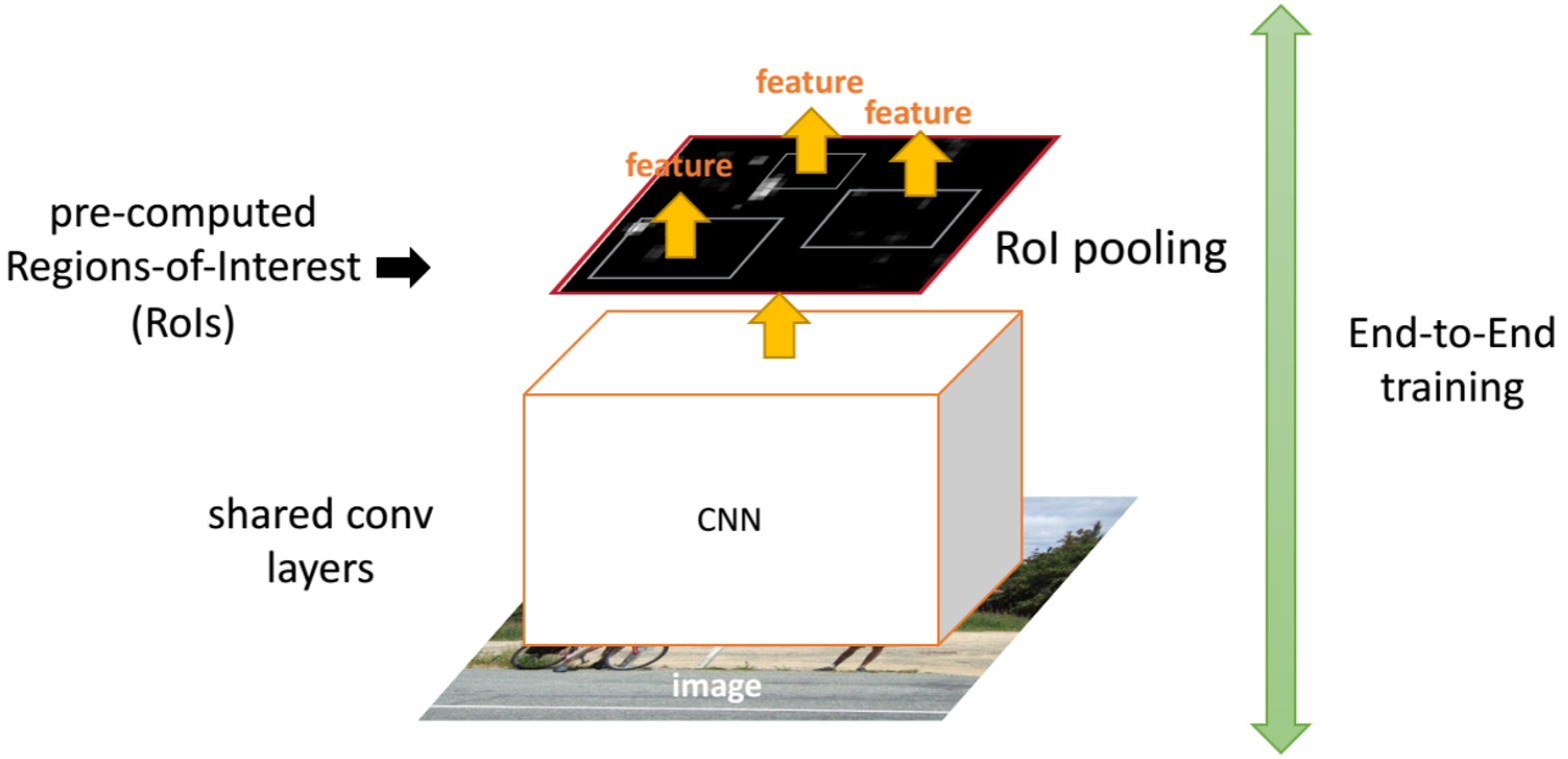
Fast R-CNN

- Region proposals are extracted from feature maps
- CNN is applied only once for all proposals
- So, fast inference
- Note: classifier and regression applied multiple times



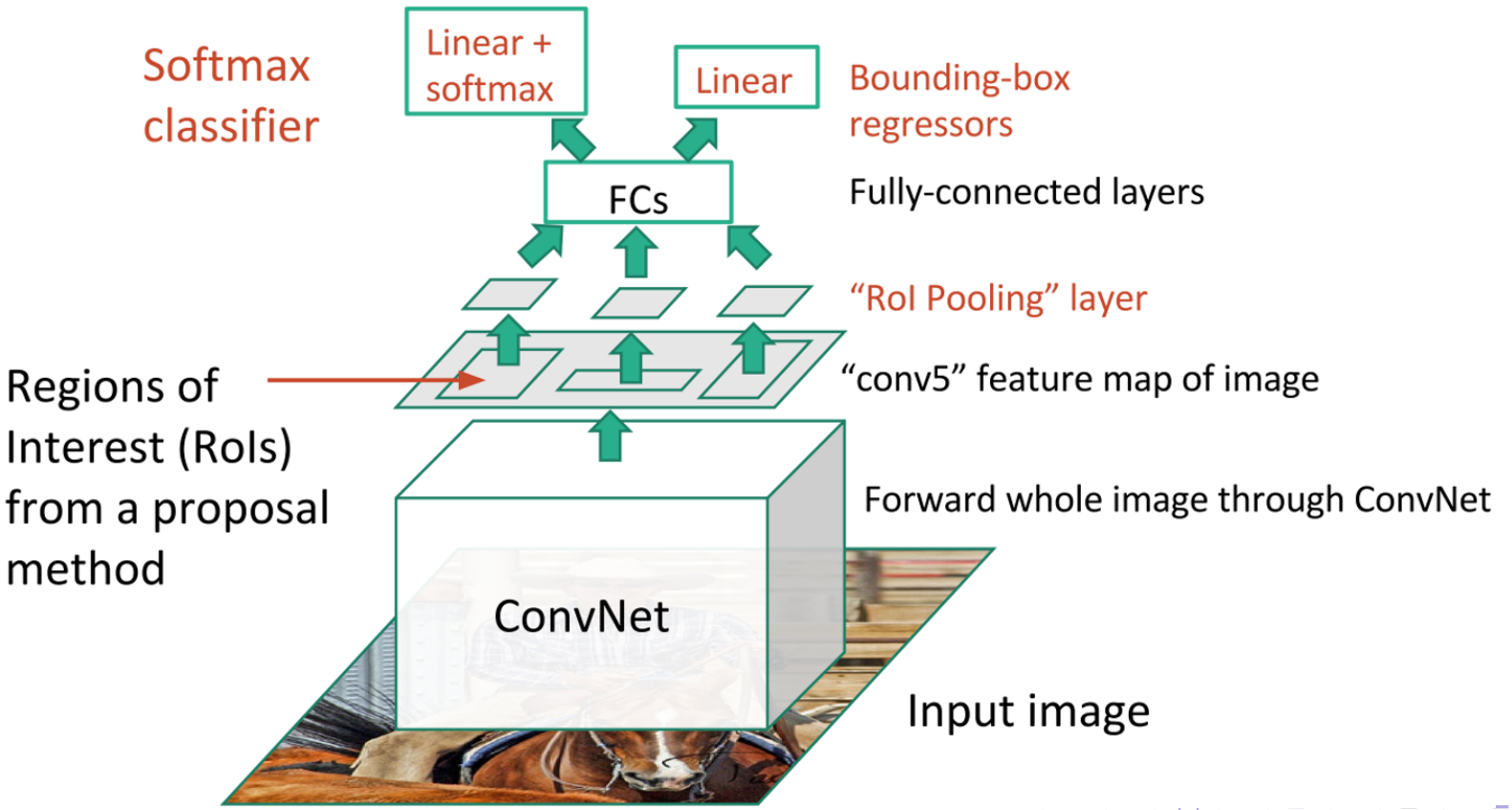
Fast R-CNN

- Region proposals are extracted from feature maps
- CNN is applied only once for all proposals
- So, fast inference
- Note: classifier and regression applied multiple times

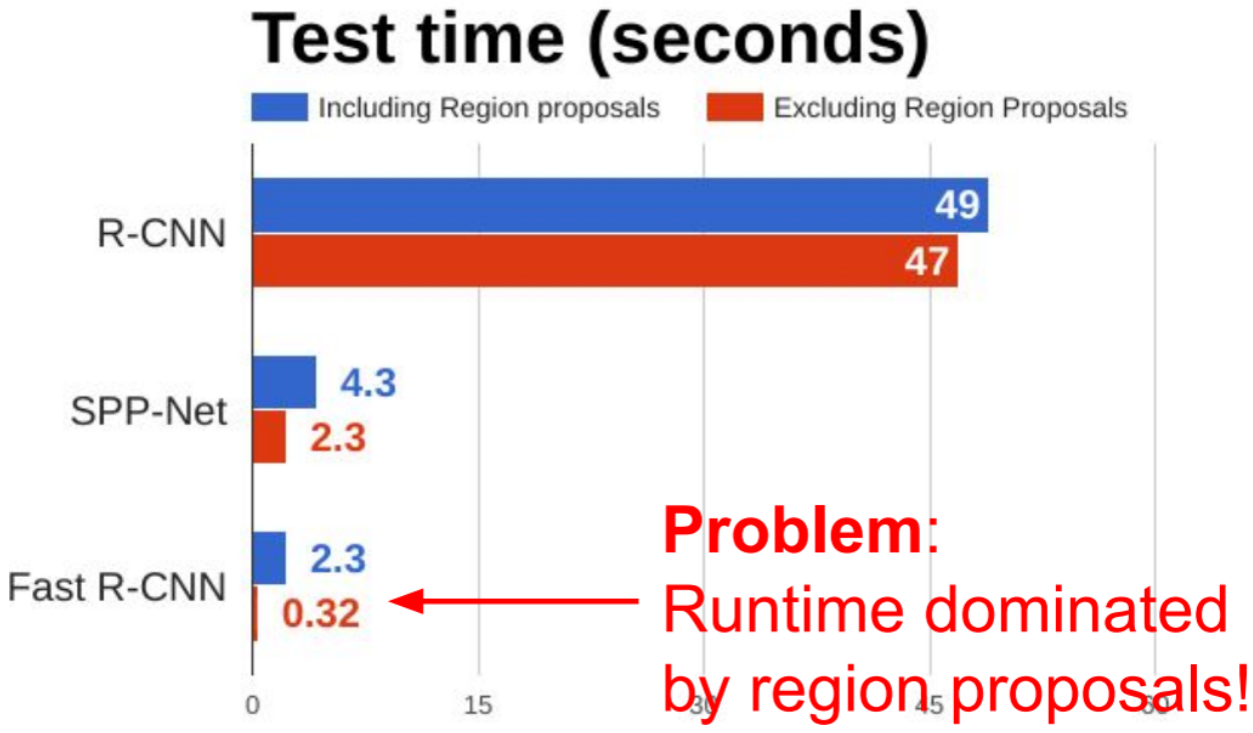


Fast R-CNN (cont')

- RoI pooling layer: pool each RoI to a pre-fixed $S \times S$ region
- Every RoI pooling output is connected to FC layers for classification and bounding box offset prediction
- Training loss $L = L_{cls} + L_{box}$

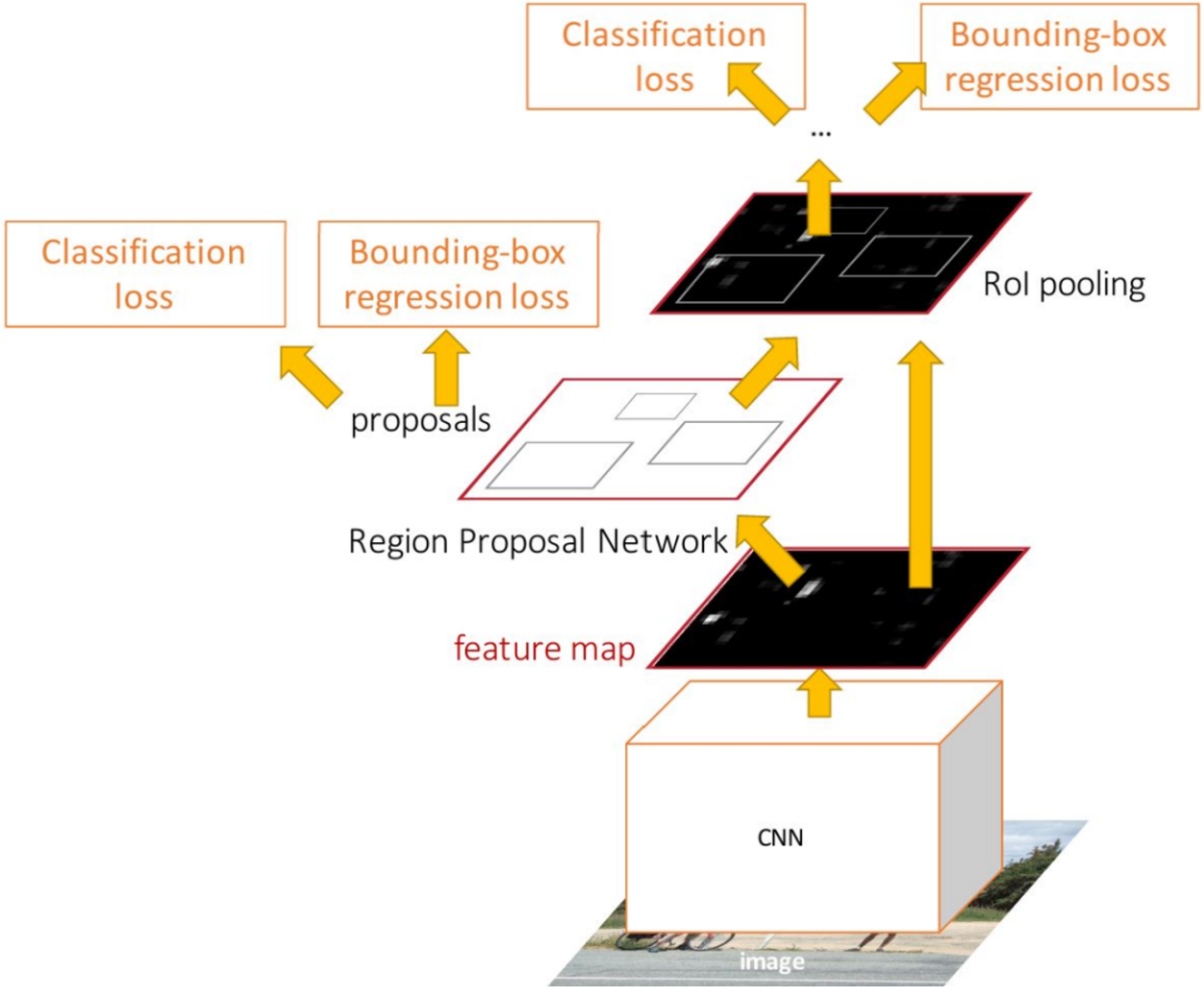


Fast R-CNN (cont')



Faster R-CNN: make CNN do proposals

- Region Proposal Network (RPN) predicts proposals
- Jointly train 4 losses; fast inference (5fps)

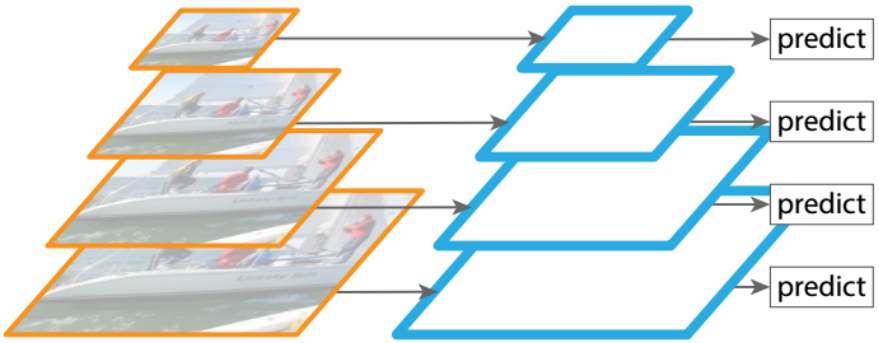


Not only faster ...

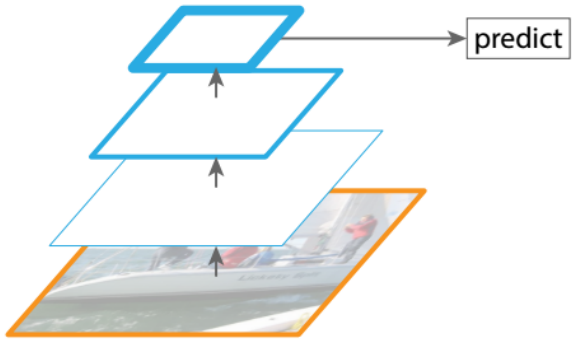
We also need more accurate detections!

Feature pyramid network (FPN) for object detection

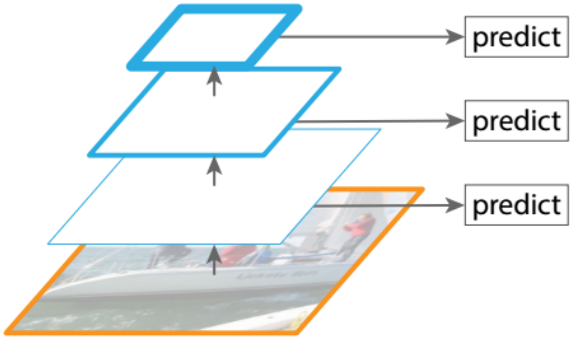
- All recent recognition and detection challenge winners use multi-scale testing on featurized image pyramids (fig. a). Key observation: features at all levels are *semantically strong*!
- However, expensive computation during prediction



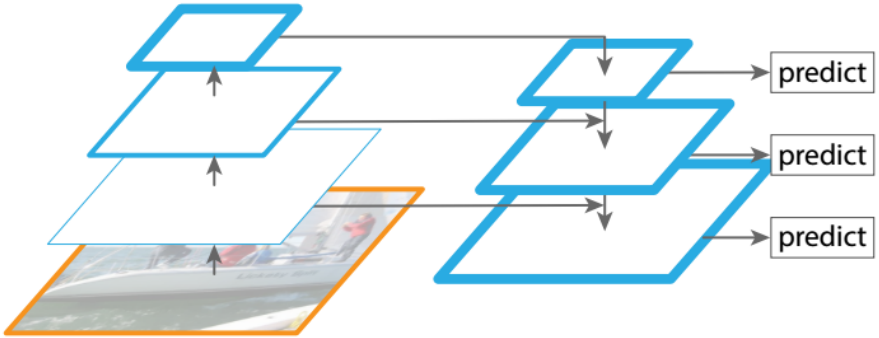
(a) Featurized image pyramid



(b) Single feature map



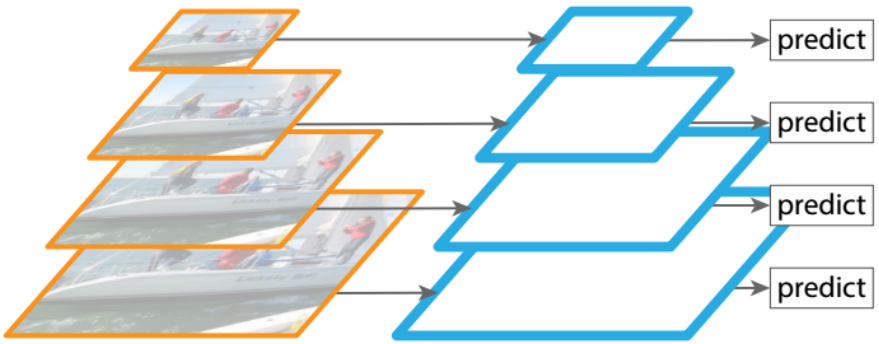
(c) Pyramidal feature hierarchy



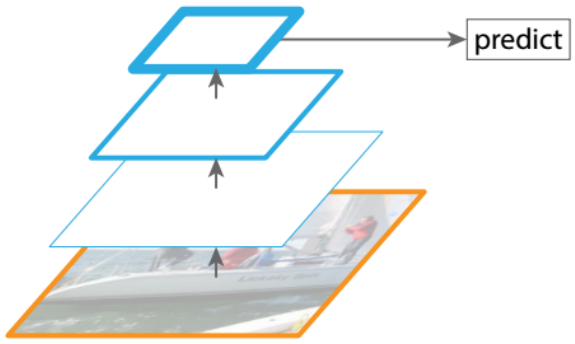
(d) Feature Pyramid Network

Feature pyramid network (FPN) for object detection

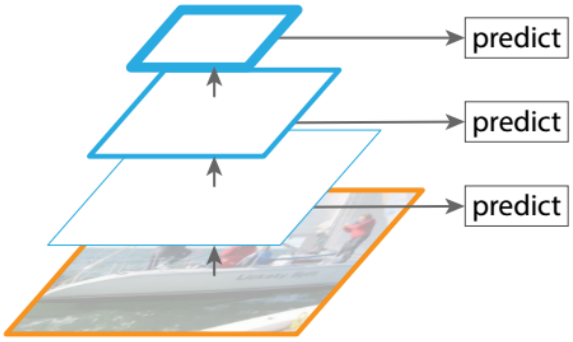
- Recently, use single scale features for faster detection (fig. b)
- High-resolution feature maps have low-level features (fig. c)



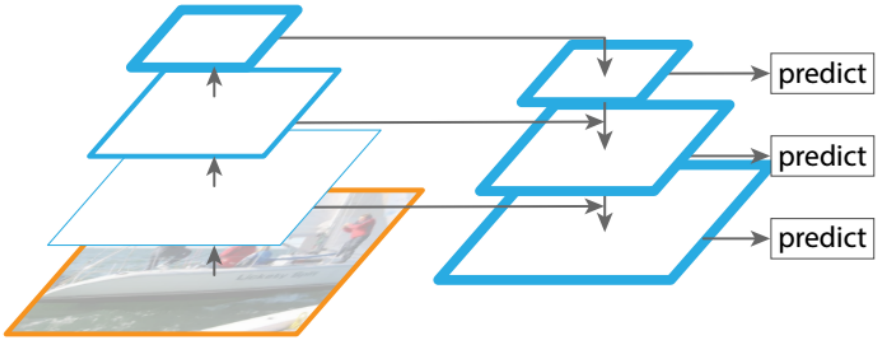
(a) Featurized image pyramid



(b) Single feature map



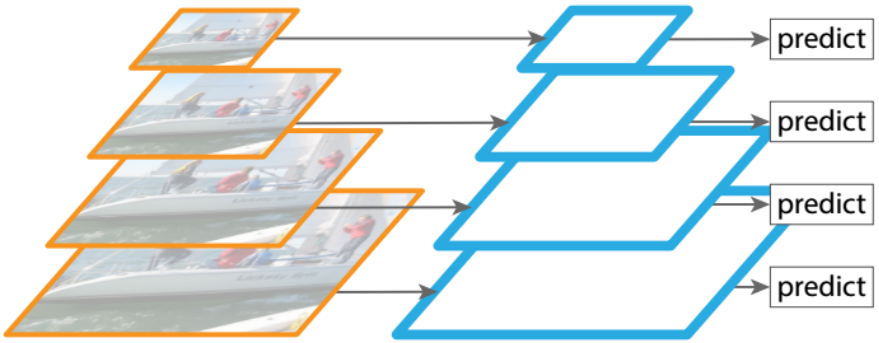
(c) Pyramidal feature hierarchy



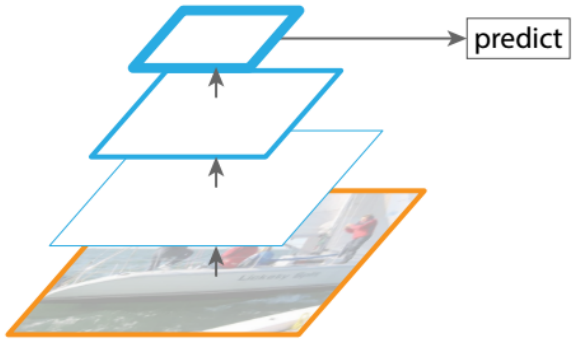
(d) Feature Pyramid Network

Feature pyramid network (FPN) for object detection

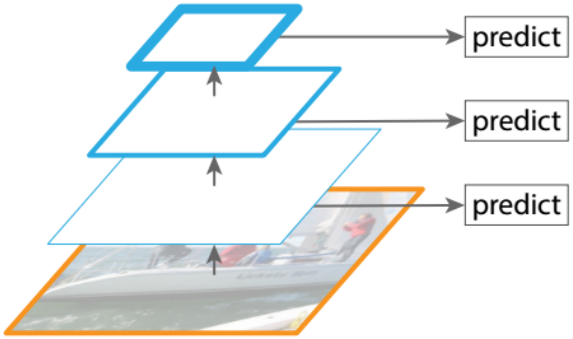
- Feature pyramid network (fig. d): rich semantics at all levels



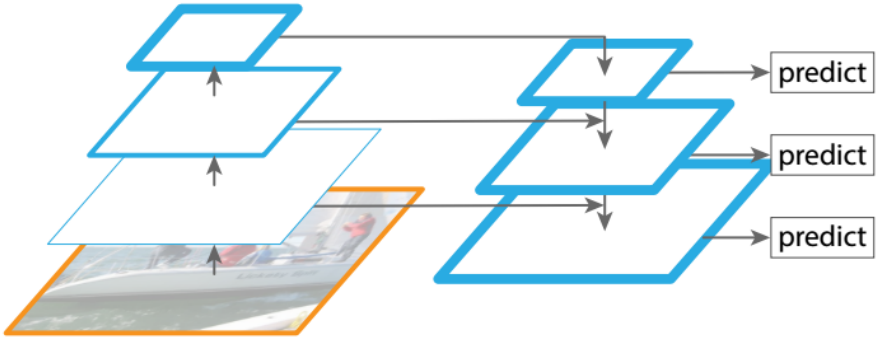
(a) Featurized image pyramid



(b) Single feature map



(c) Pyramidal feature hierarchy

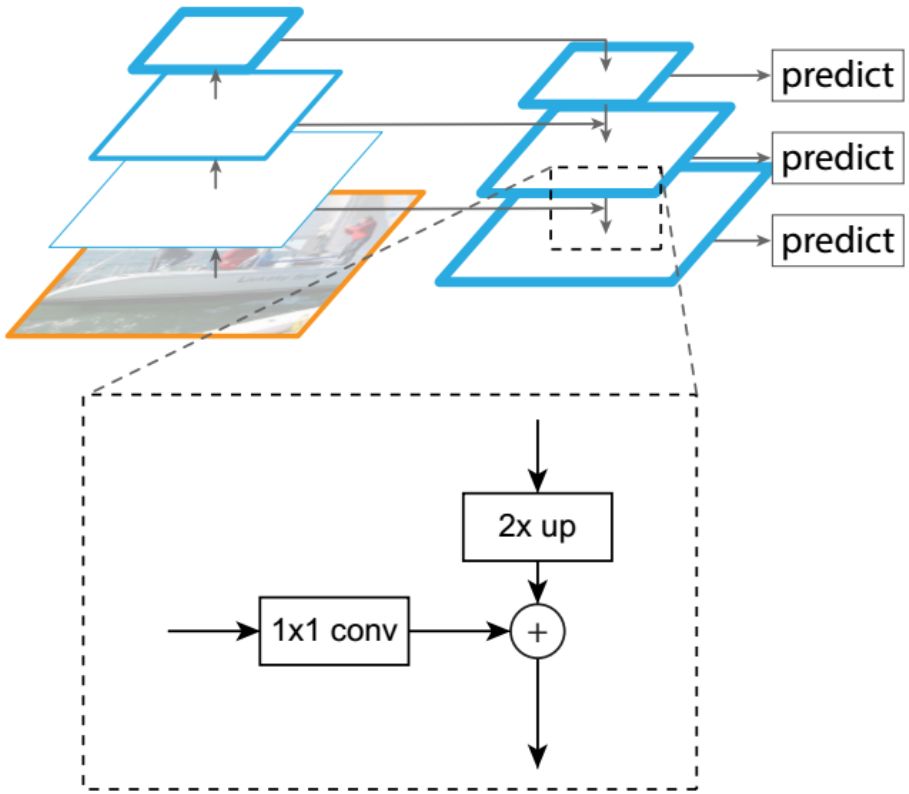


(d) Feature Pyramid Network

Figures here and in the next two slides from Lin, Dollar, Girshick, He, Hariharan, Belongie, 'Feature pyramid network for object detection', CVPR, 2017

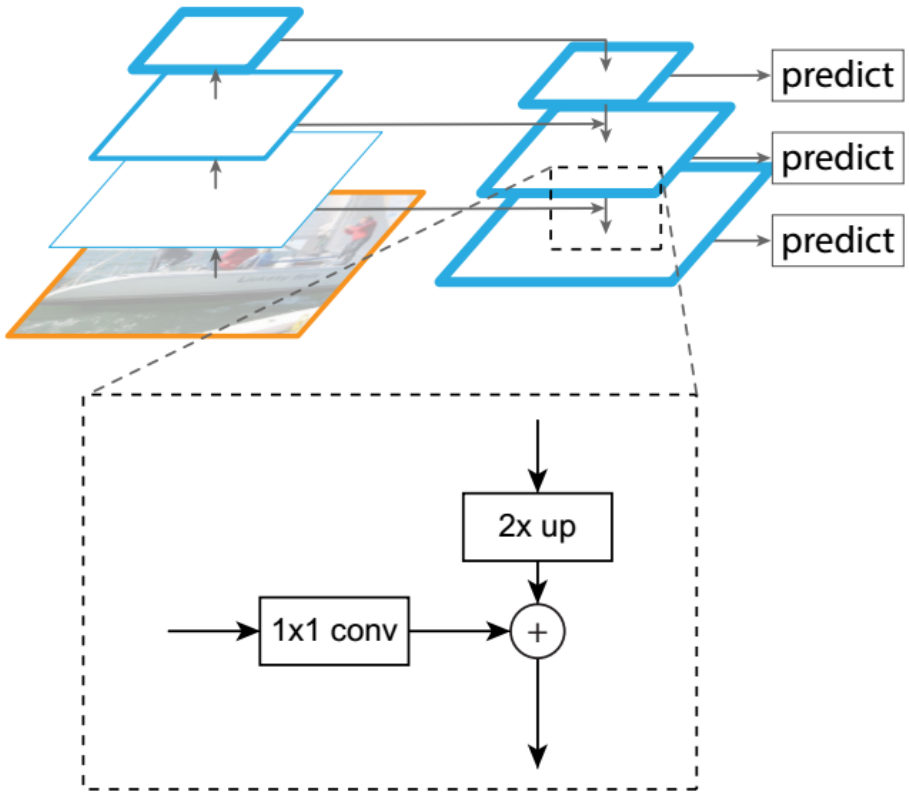
FPN

- Top-down path: upsample spatially coarser but semantically stronger from higher levels; embed semantics into lower levels
- All levels use shared classifiers/regressors
- Each RoI is assigned to one level for object detection
- Only use a single-scale input image



FPN

- Top-down path: upsample spatially coarser but semantically stronger from higher levels; embed semantics into lower levels
- All levels use shared classifiers/regressors
- Each RoI is assigned to one level for object detection
- Only use a single-scale input image



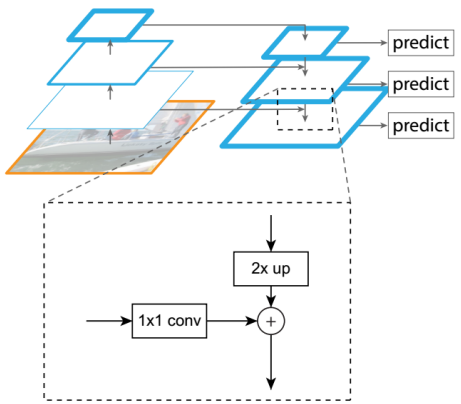
FPN (cont')

FPN is a network architecture

- Independent of backbone networks like ResNet
- Can be used for Region Proposal Network, Fast R-CNN

FPN vs. FCN in head architecture

- FCN head is connected to last upsampled conv layer and fully convolutoinal
- FPN head is connected to each upsampled layer and ends with either fully connected layers for classification/regression or with fully convolutional layer for segmentation



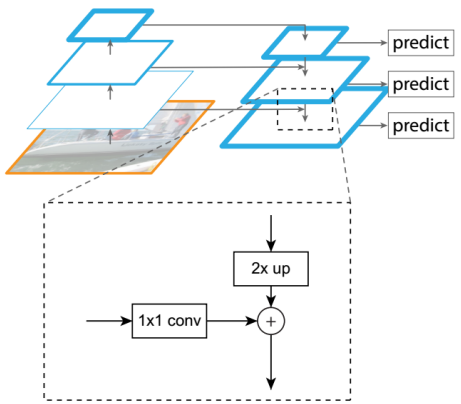
FPN (cont')

FPN is a network architecture

- Independent of backbone networks like ResNet
- Can be used for Region Proposal Network, Fast R-CNN

FPN vs. FCN in head architecture

- FCN head is connected to last upsampled conv layer and fully convolutoinal
- FPN head is connected to each upsampled layer and ends with either fully connected layers for classification/regression or with fully convolutional layer for segmentation



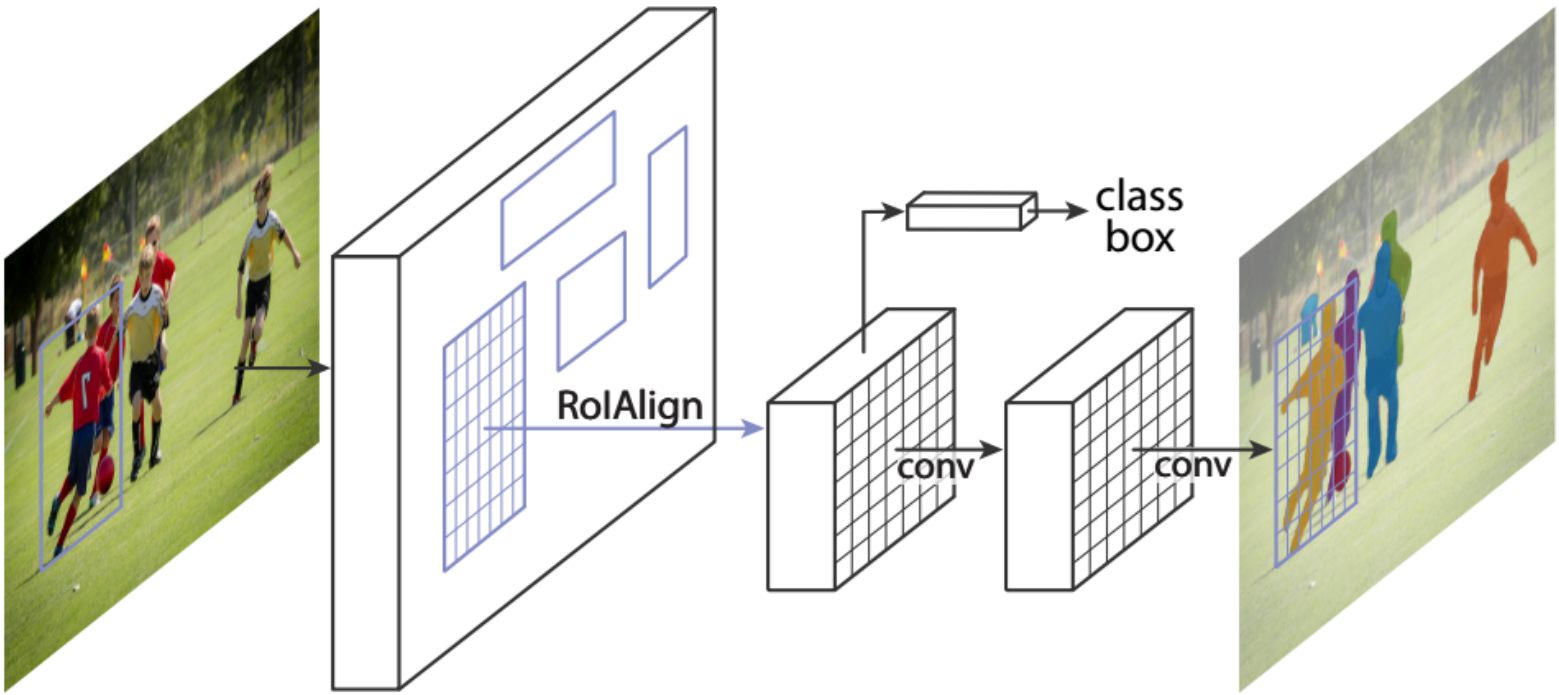
So far ...

Seen object detections and semantic segmentation!

Combine them for instance segmentation!

Mask R-CNN

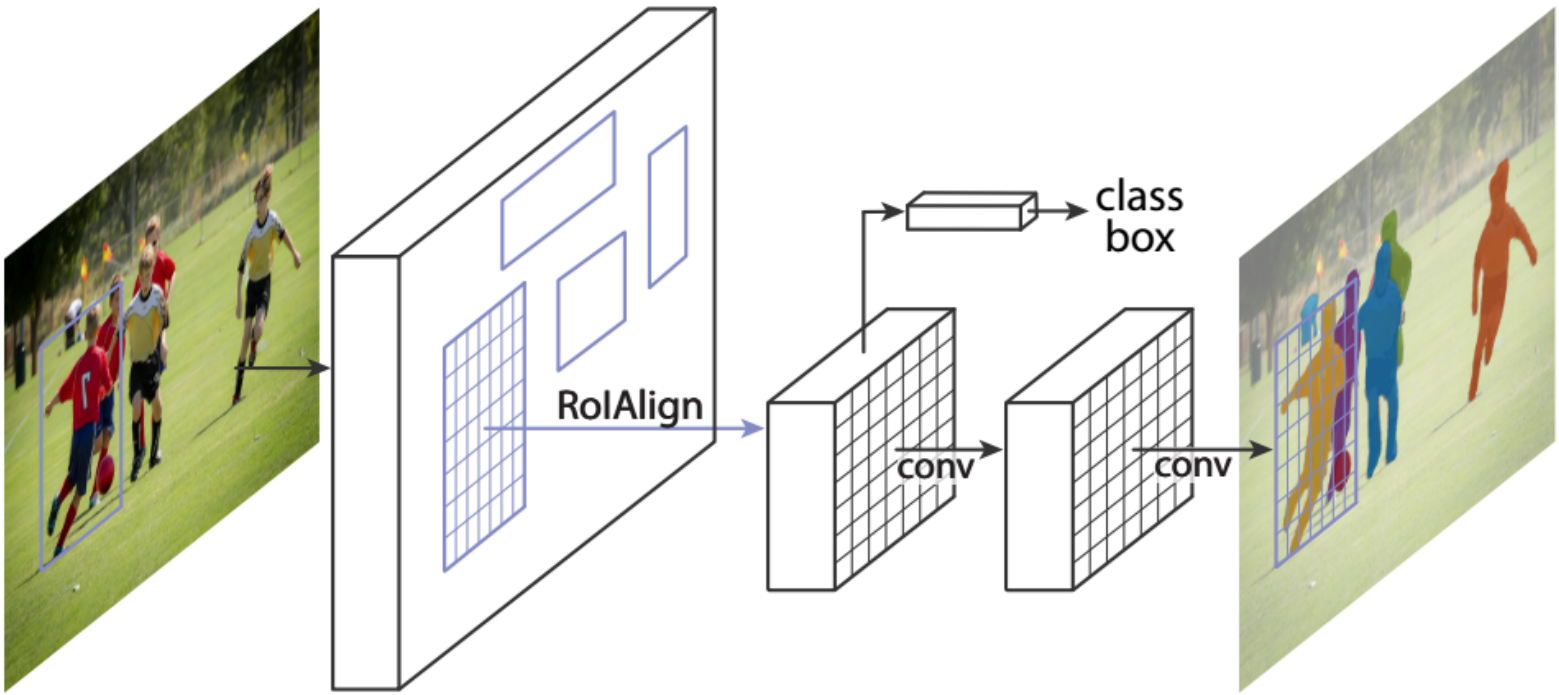
- Extends Faster R-CNN by adding a sub-network (FCN) branch to predict segmentation masks on each RoI
- Mask R-CNN = Faster R-CNN + FCN
- Segmentation mask output: K (i.e., # classes) channels



Figures here and in the next three slides from He, Gkioxari, Dollar, Girschick, 'Mask R-CNN', ICCV 2017

Mask R-CNN

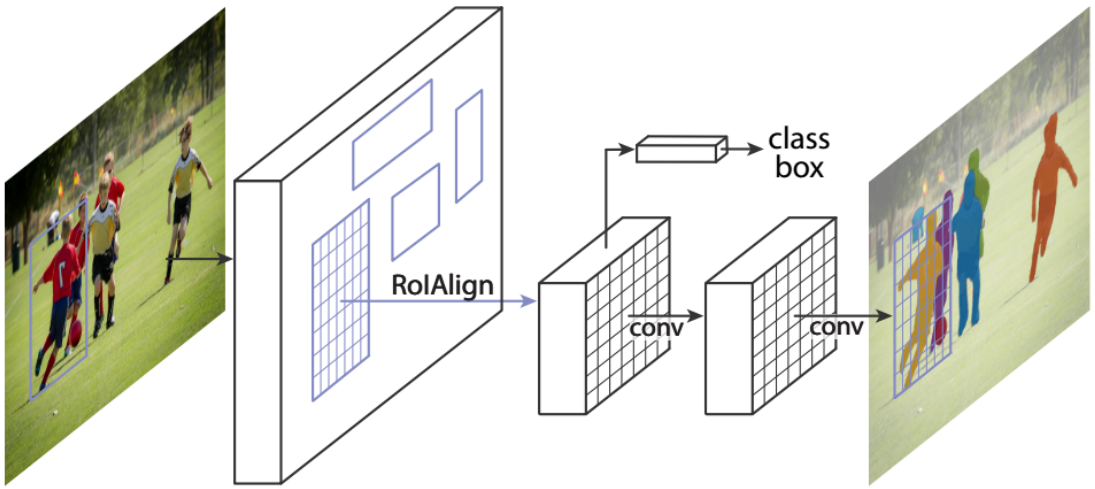
- Extends Faster R-CNN by adding a sub-network (FCN) branch to predict segmentation masks on each RoI
- Mask R-CNN = Faster R-CNN + FCN
- Segmentation mask output: K (i.e., # classes) channels



Figures here and in the next three slides from He, Gkioxari, Dollar, Girschick, 'Mask R-CNN', ICCV, 2017

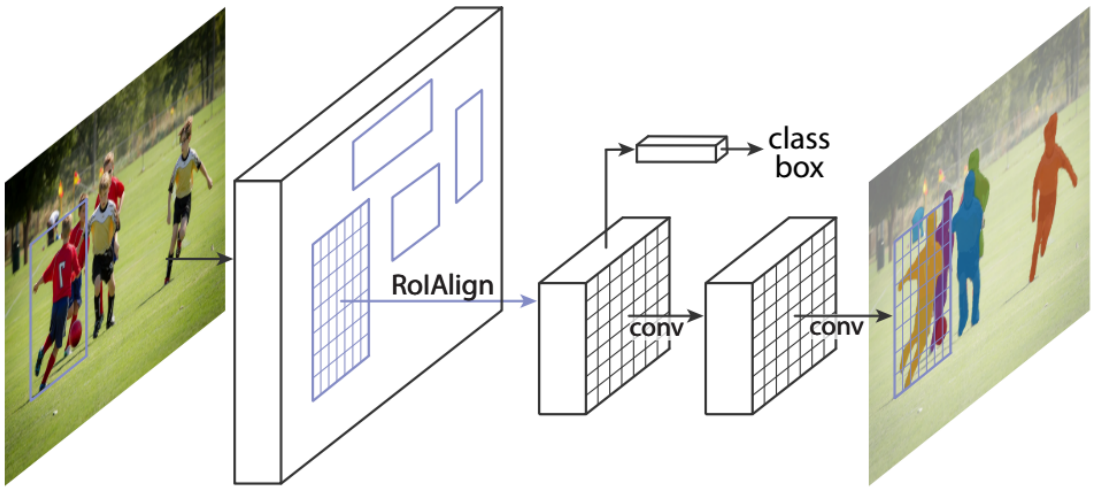
Mask R-CNN

- Training loss $L = L_{cls} + L_{box} + L_{mask}$



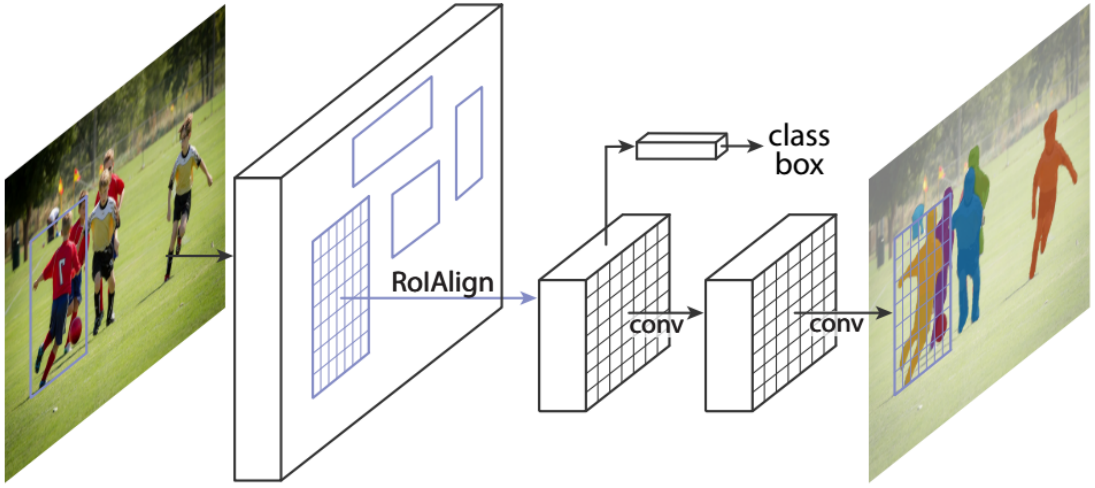
Mask R-CNN

- Training loss $L = L_{cls} + L_{box} + L_{mask}$
- Novelty 1: decouple segmentation and classification tasks, i.e., generate masks for every class without competition among classes (or: per-pixel binary categorization).
  Other FCNs: perform per-pixel multi-class categorization!



Mask R-CNN

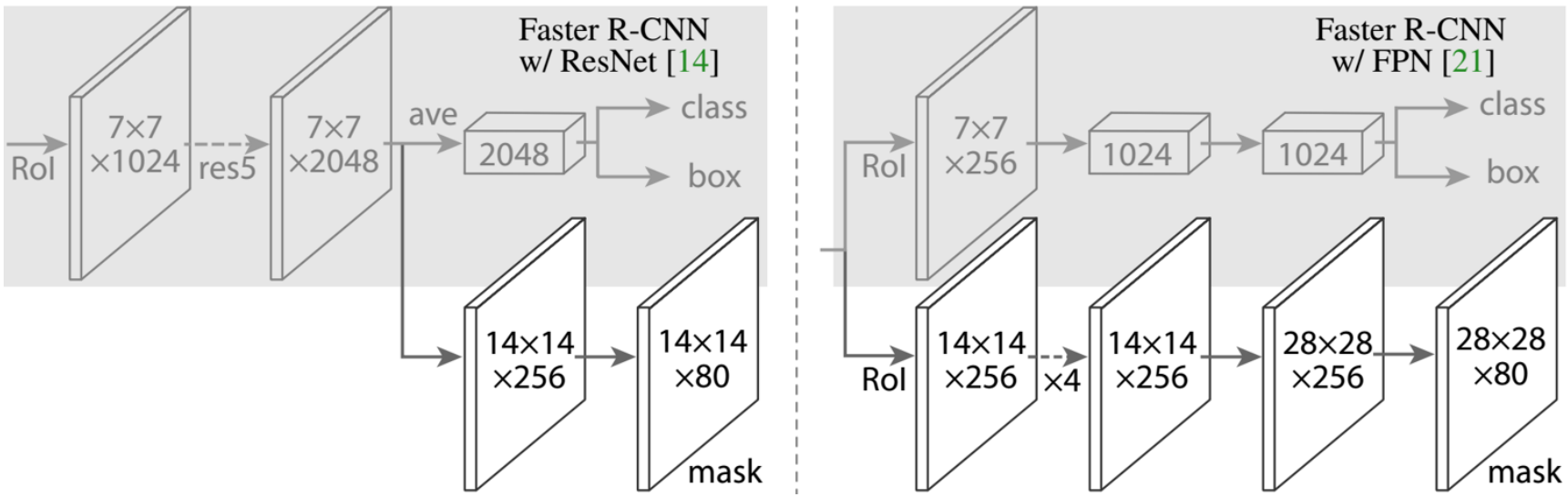
- Training loss $L = L_{cls} + L_{box} + L_{mask}$
- Novelty 1: decouple segmentation and classification tasks, i.e., generate masks for every class without competition among classes (or: per-pixel binary categorization).
  Other FCNs: perform per-pixel multi-class categorization!
- Novelty 2: RoIAlign - avoid quantization of RoI locations and bins during RoIPool, e.g., use $x/16$ instead of $\lfloor x/16 \rfloor$



Mask R-CNN

Network architectures: backbone + head

- backbone: first four conv blocks of ResNet or FPN
- head architecture, e.g.,



Mask R-CNN: result

- Outperform COCO challenge 2016 winners in bounding-box object detection, instance segmentation, etc.
- Still fast: 5 fps

One-stage approach

Above are two-stage models!

In comparison, one-stage models:

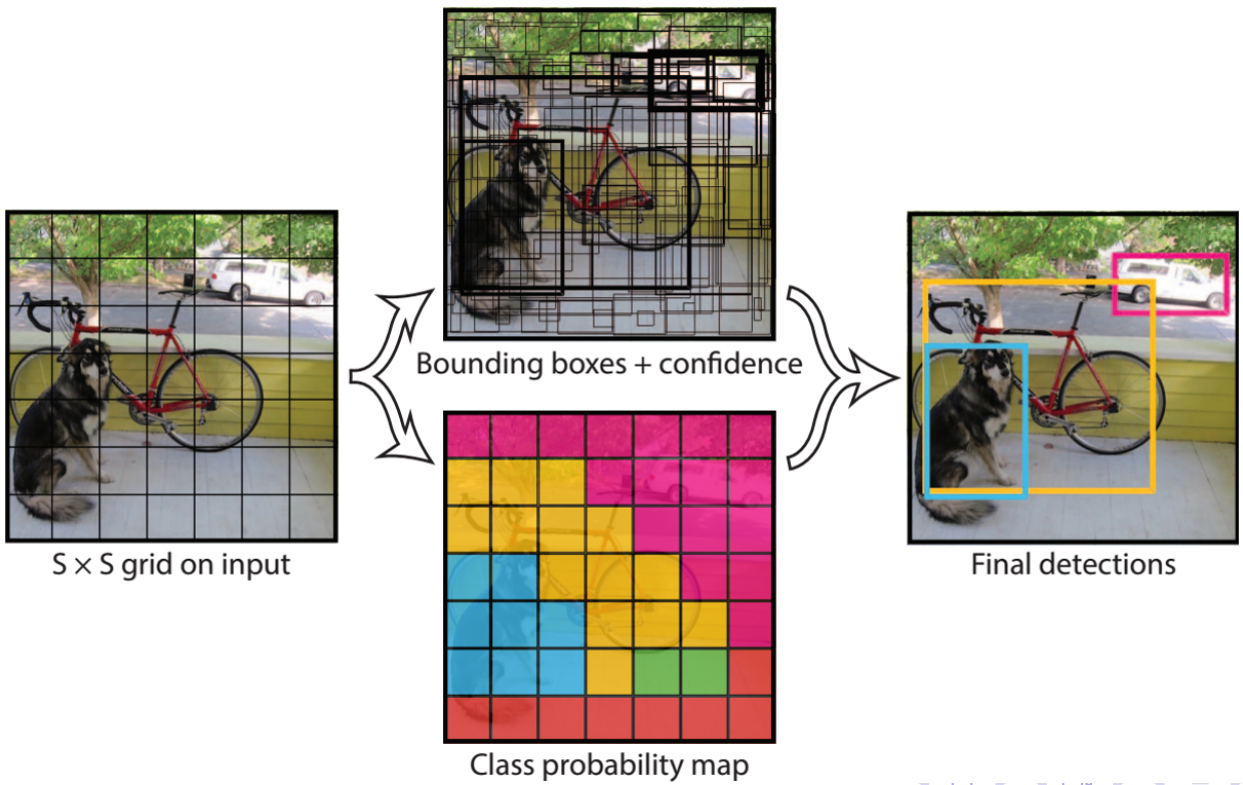
No need region proposal stage!

Simultaneously predict boxes and class probabilities!

Fast (real-time), but lower detection accuracy!

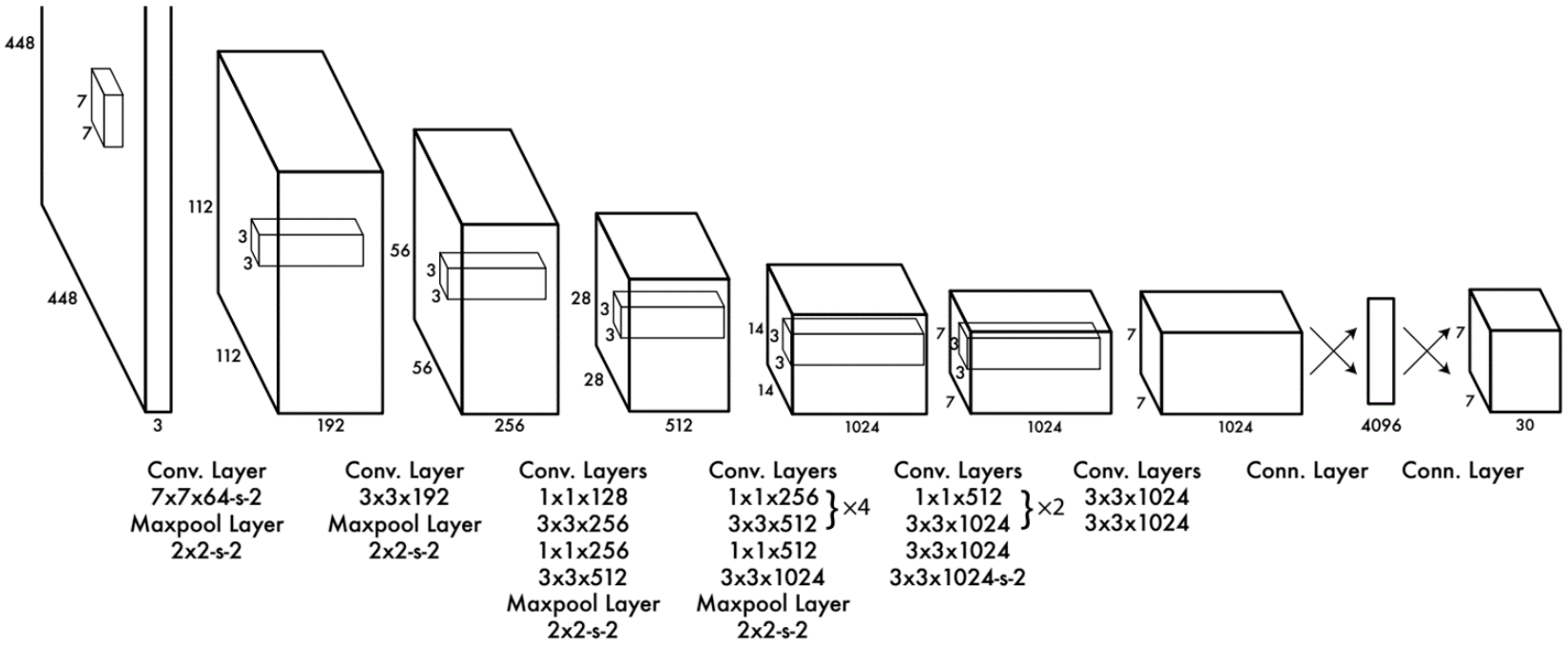
YOLO: You Only Look Once

- Divide input image into an $S \times S$ grid
- If the centre of an object falls into a grid cell, that grid cell is responsible for detecting that object.
- Each cell: predicts B boxes & confidences, C class probabilities (Confidence: IoU between predicted and ground-truth boxes)



YOLO: You Only Look Once

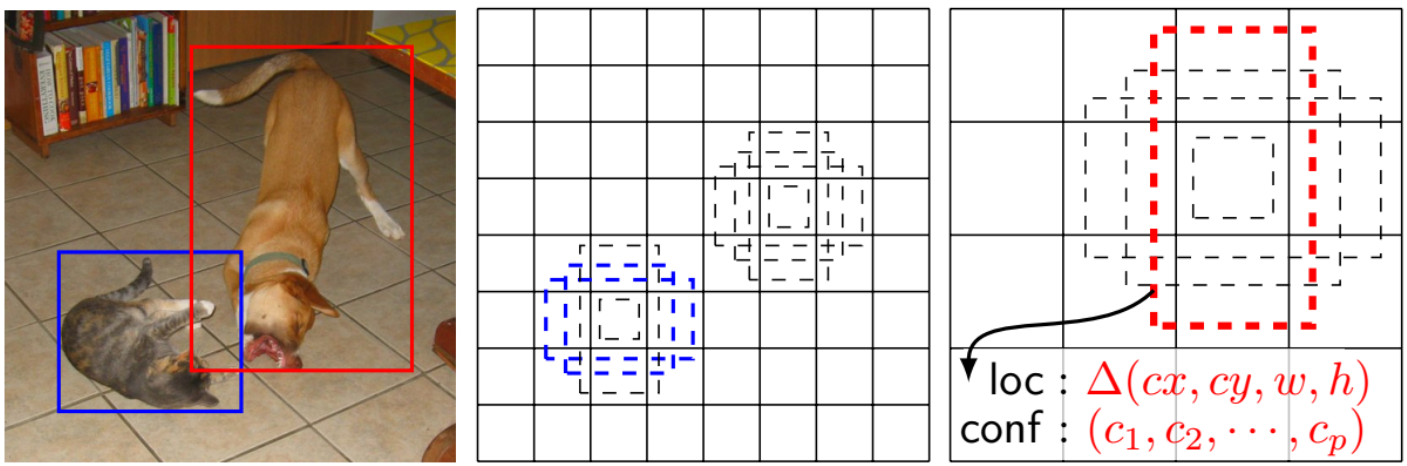
- FC layer: use features from entire image to predict each box
- Output of network: $S \times S \times (B * 5 + C)$, e.g., $B = 2$, $C = 20$
- Non-maximal suppression to fix multiple nearby detections
- Inference: 45fps; cannot detect small objects in groups



Figures here and in the previous slide from Redmon, Divvala, Girshick, Farhadi, 'You only look once: unified, real-time object detection', CVPR, 2016

SSD: single shot multibox detector

- Use multiple layers of feature maps
- B (4 or 6) default boxes generated at each location per layer
- Dashed blue & red: positive boxes (good match with objects)
- $\triangle\ (cx, cy, w, h)$: predicted offset (from red box) by network to well match ground-truth box
- $(c_1, c_2, \ldots, c_p)$: predicted class probabilities for red box
- Minimize loss $L = L_{cls} + L_{box}$

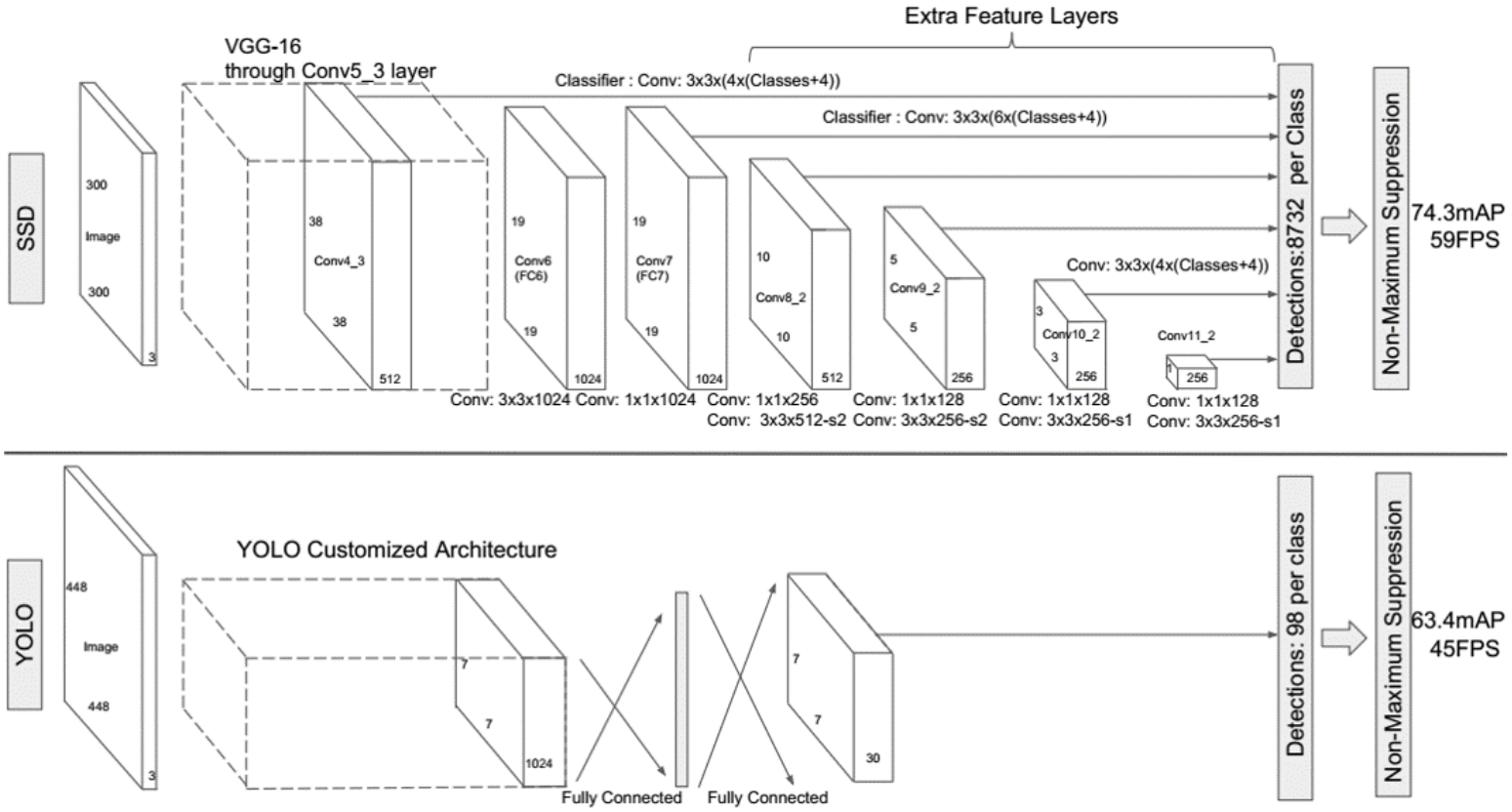


(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

Figures here and in next slide from Liu et al., 'SSD: single shot multibox detector', ECCV, 2016

SSD: single shot multibox detector

- Train different predictors (with small filters) for different aspect ratio detection per layer
- For a $S \times S$ feature map: $S \times S \times B \times (C + 4)$ outputs
- Inference: faster: ~ 60fps; more accurate than YOLO



Why one-stage models have lower accuracy?

One-stage models: faster, but *lower* accuracy.

Why?

They densely sample locations, scales, aspect ratios.

So: extreme foreground-background class imbalance during training!

Why one-stage models have lower accuracy?

One-stage models: faster, but *lower* accuracy.

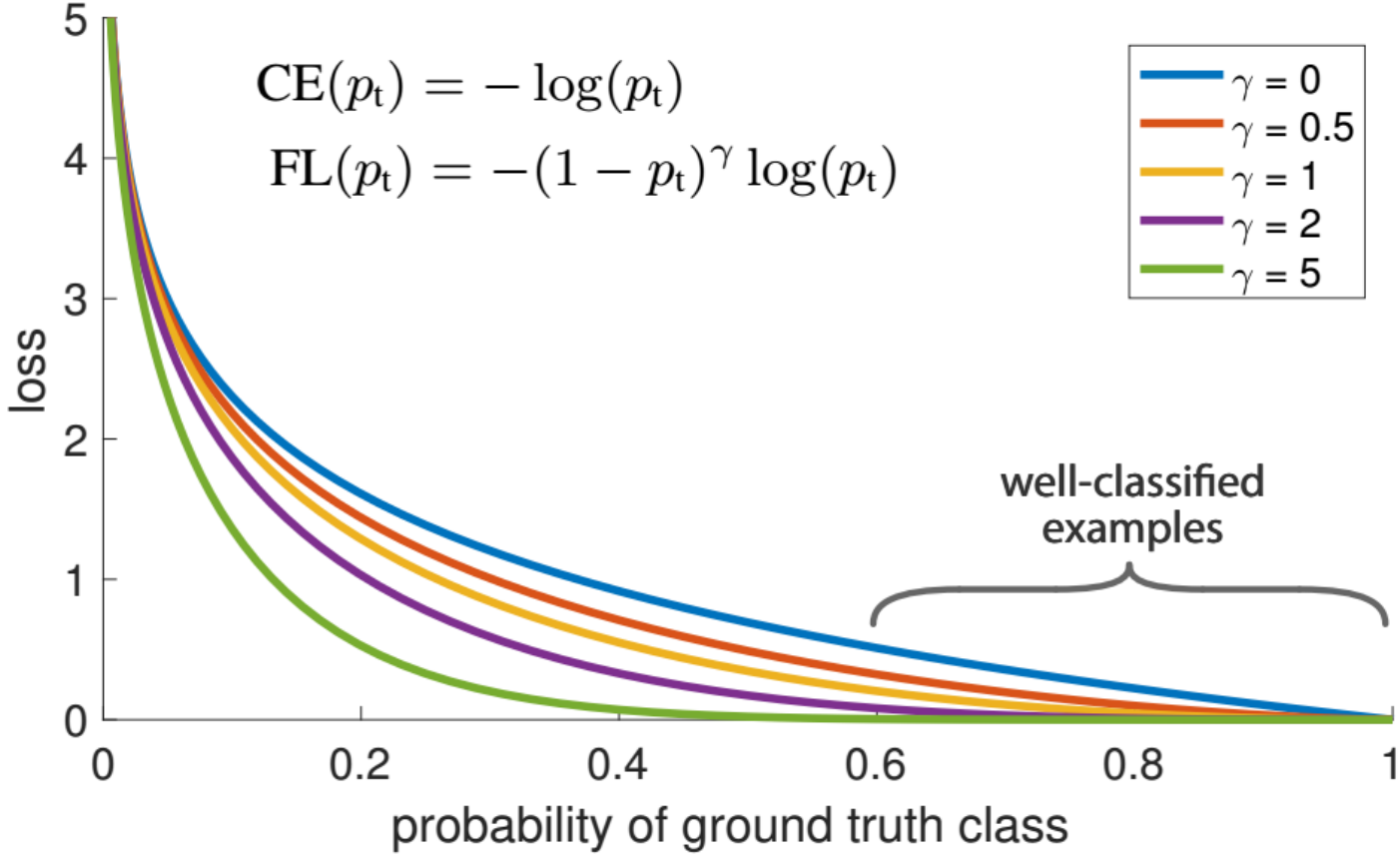
Why?

They densely sample locations, scales, aspect ratios.

So: extreme foreground-background class imbalance during training!

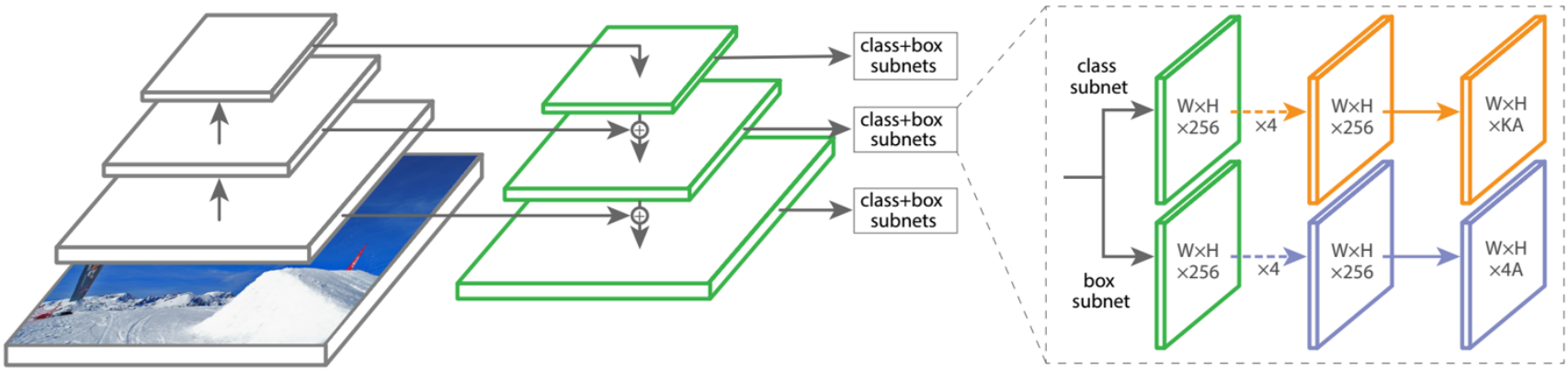
Solution: focal loss

- Focal loss down-weights the loss for well-classified examples
- It automatically focuses on hard examples during training
- CE: cross-entropy loss; FC: focal loss; p_t: model prediction



RetinaNet

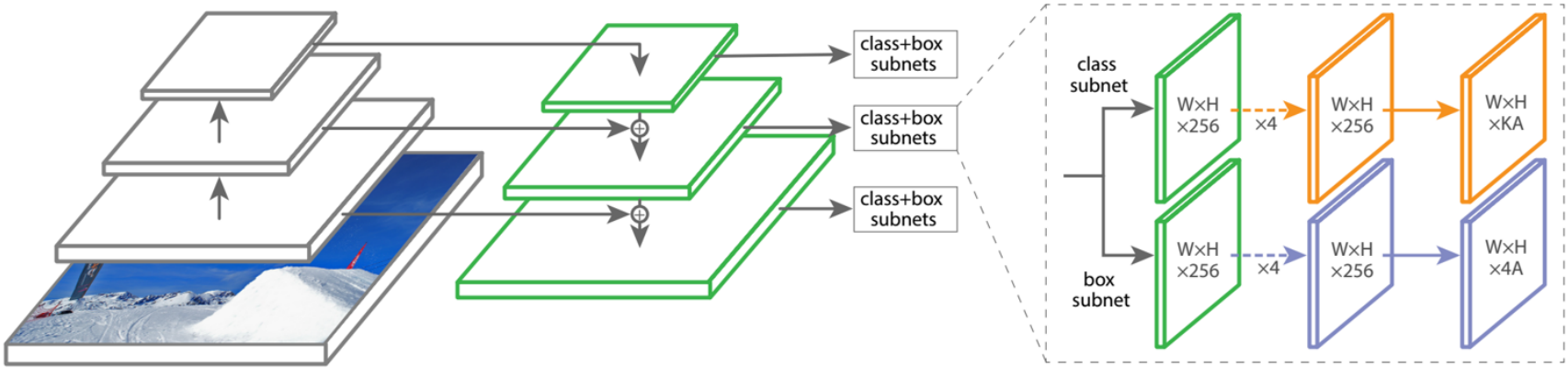
- RetinaNet: ResNet+FPN+2 FCNs
- *One-stage* model for object detection
- Parameters of two subnets shared across all levels
- Focal loss for classification subnet
- Merge detections across levels, non-maximum suppression
- Achieve state-of-the-art results when trained with focal loss
- A: number of default ('anchor') boxes at each location



Figures here and in previous slide from Lin, Goyal, Girshick, He, Dollar, 'Focal loss for dense object detection', ICCV, 2017

RetinaNet

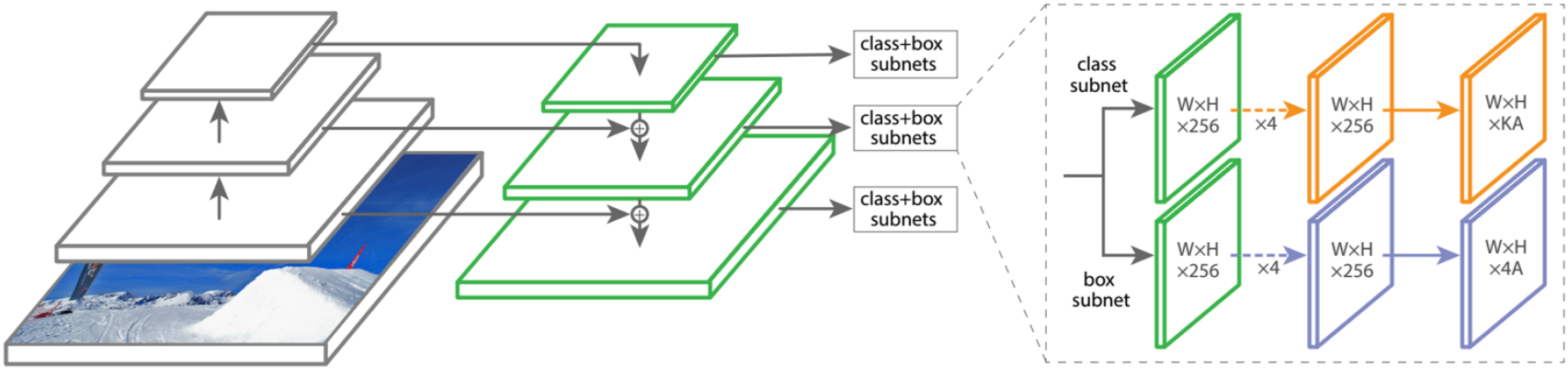
- RetinaNet: ResNet+FPN+2 FCNs
- *One-stage* model for object detection
- Parameters of two subnets shared across all levels
- Focal loss for classification subnet
- Merge detections across levels, non-maximum suppression
- Achieve state-of-the-art results when trained with focal loss
- A: number of default ('anchor') boxes at each location



Figures here and in previous slide from Lin, Goyal, Girshick, He, Dollar, 'Focal loss for dense object detection', ICCV, 2017

RetinaNet

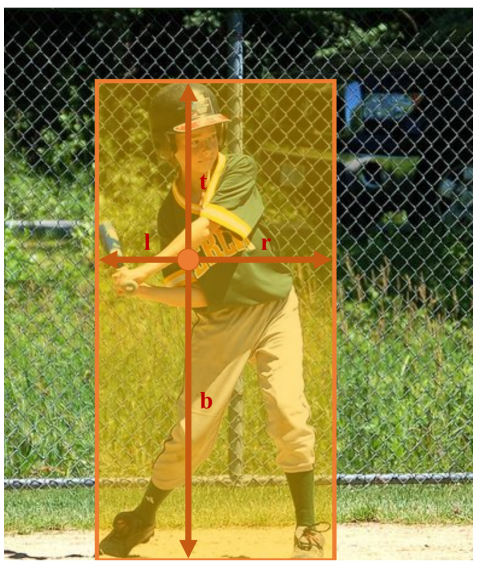
- RetinaNet: ResNet+FPN+2 FCNs
- *One-stage* model for object detection
- Parameters of two subnets shared across all levels
- Focal loss for classification subnet
- Merge detections across levels, non-maximum suppression
- Achieve state-of-the-art results when trained with focal loss
- A: number of default ('anchor') boxes at each location



Figures here and in previous slide from Lin, Goyal, Girshick, He, Dollar, 'Focal loss for dense object detection', ICCV, 2017

FCOS: fully convolutional one-stage object detection

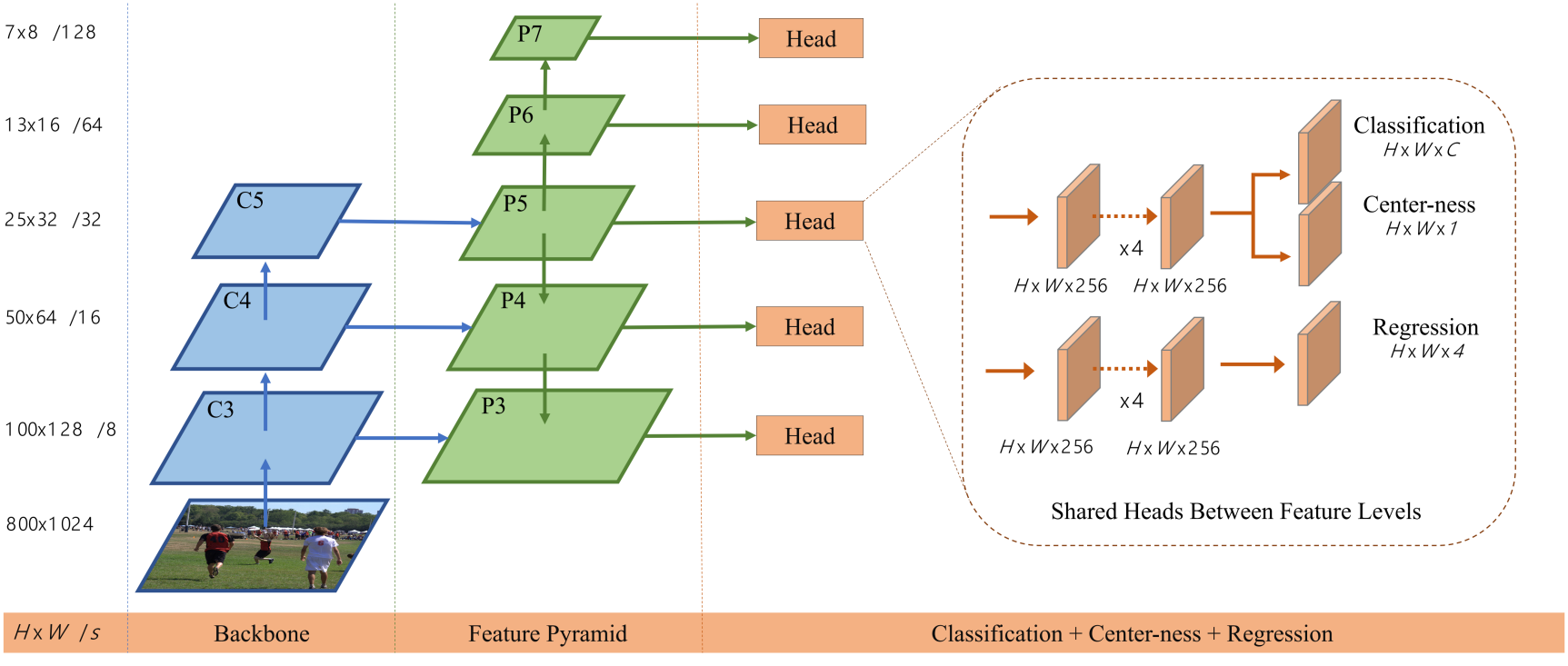
- Bounding box free: avoid hyper-parameters (size, aspect ratio)
- Predict distances from *each* foreground pixel to 4 boundaries (left, top, right, bottom) of the object



Figures here and in next slide from Tian, Shen, Che, He, 'FCOS: fully convolutional one-stage object detection', arXiv, 2019

FCOS (cont')

- Architecture similar to RetinaNet; loss different
- Different levels of feature maps for objects of different sizes
- *One-stage* model for object detection

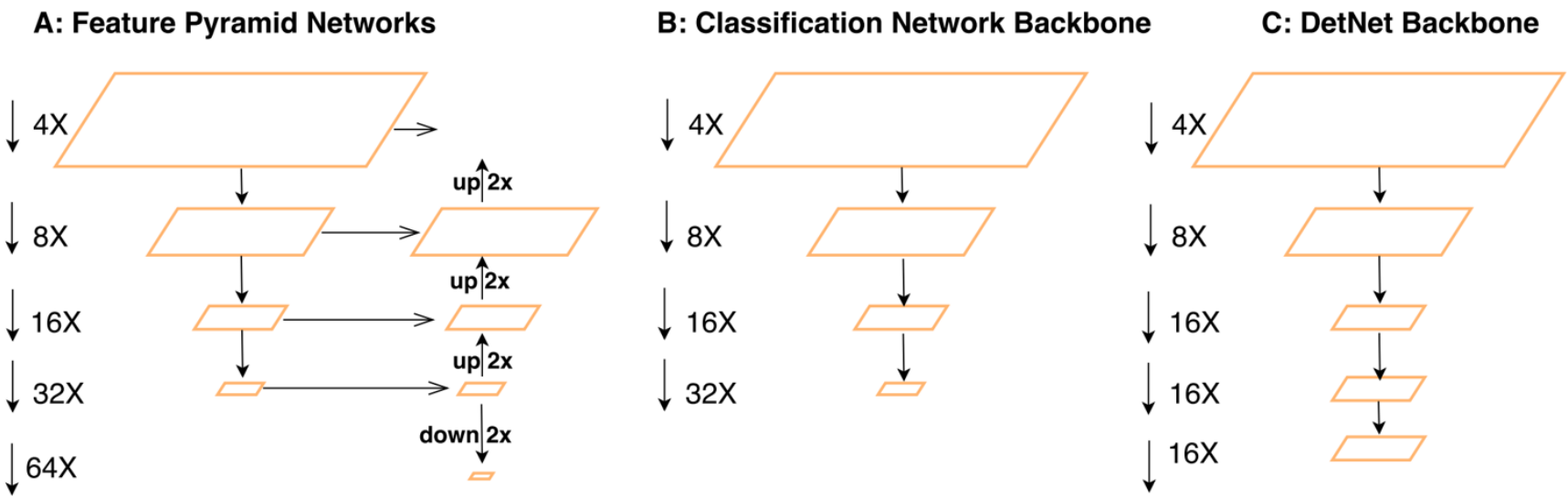


Issue for both one-stage and two-stage models

Above *detection* models depend on network pretrained on ImageNet for *classification* task

DetNet: a backbone network for object detection

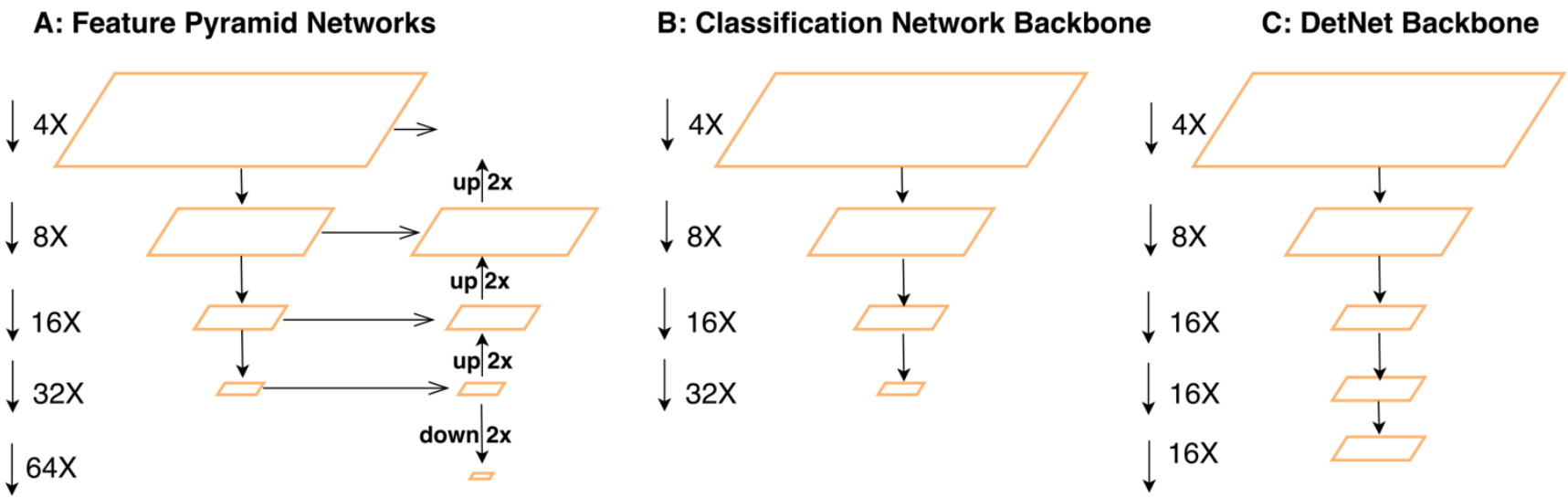
- Large stride (downsampling) is harmful for localization (fig. B)
- Large stride causes missing of small objects (fig. B)
- Shallower layers have low semantic information (fig. B)
- Even in FPN, context cues of small objects also miss (fig. A)



Figures here and in next five slides from Li, Peng, Yu, Zhang, Deng, Sun, 'DetNet: a backbone network for object detection', ECCV, 2018

DetNet: a backbone network for object detection

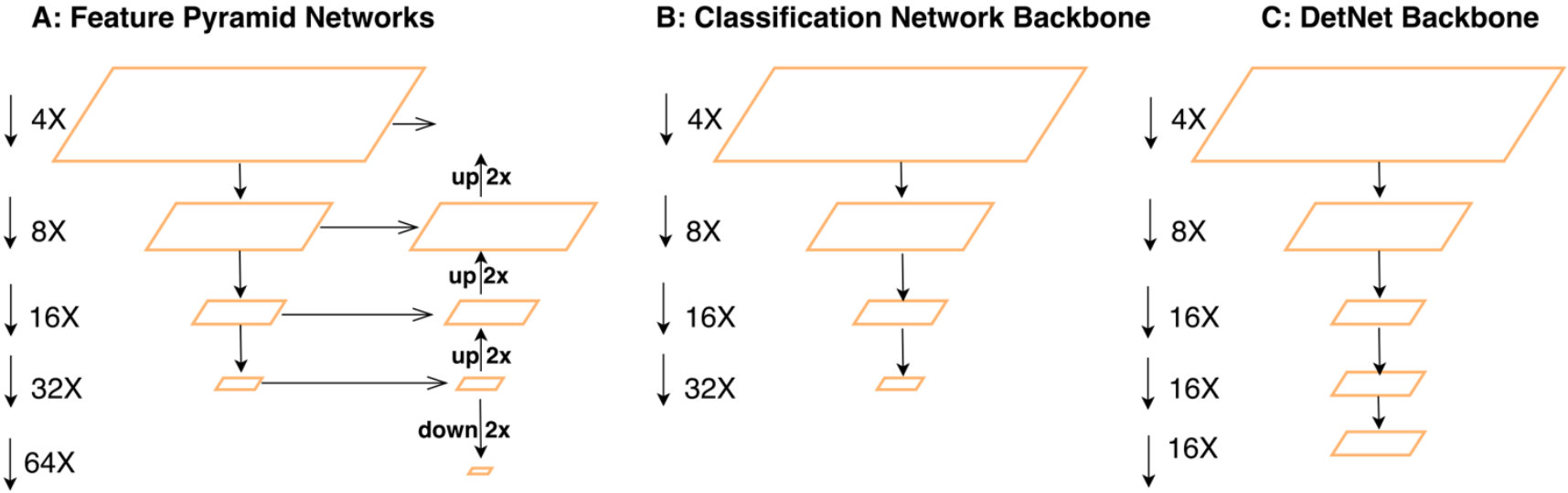
- Large stride (downsampling) is harmful for localization (fig. B)
- Large stride causes missing of small objects (fig. B)
- Shallower layers have low semantic information (fig. B)
- Even in FPN, context cues of small objects also miss (fig. A)



Figures here and in next five slides from Li, Peng, Yu, Zhang, Deng, Sun, 'DetNet: a backbone network for object detection', ECCV, 2018

DetNet: a backbone network for object detection

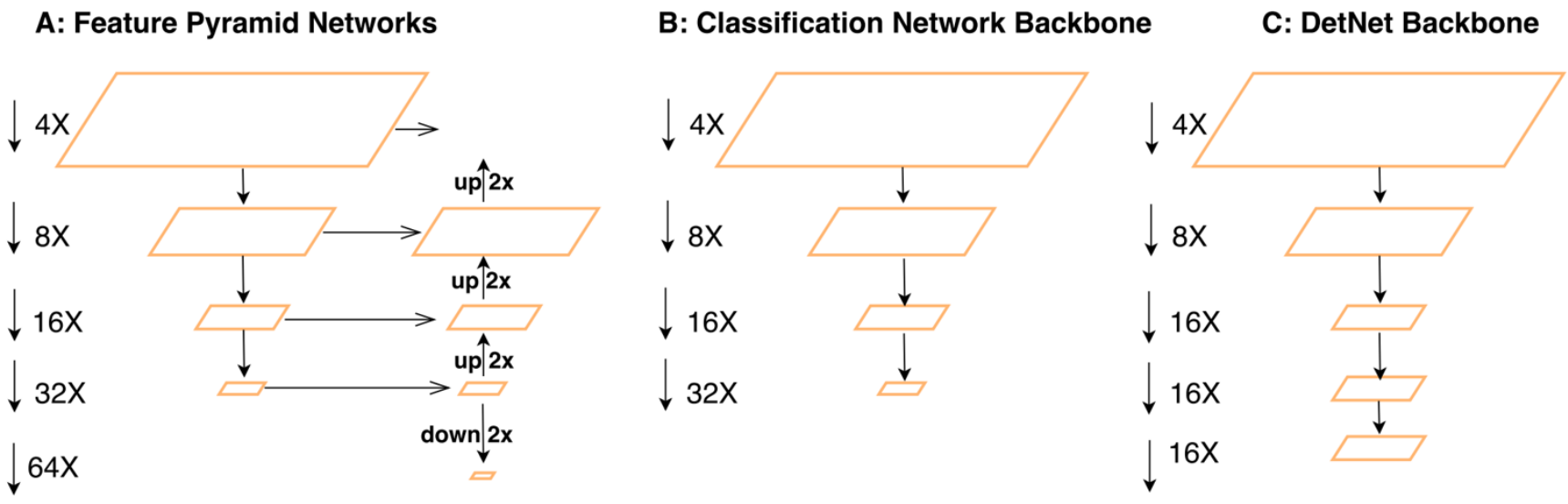
- Large stride (downsampling) is harmful for localization (fig. B)
- Large stride causes missing of small objects (fig. B)
- Shallower layers have low semantic information (fig. B)
- Even in FPN, context cues of small objects also miss (fig. A)



Figures here and in next five slides from Li, Peng, Yu, Zhang, Deng, Sun, 'DetNet: a backbone network for object detection', ECCV, 2018

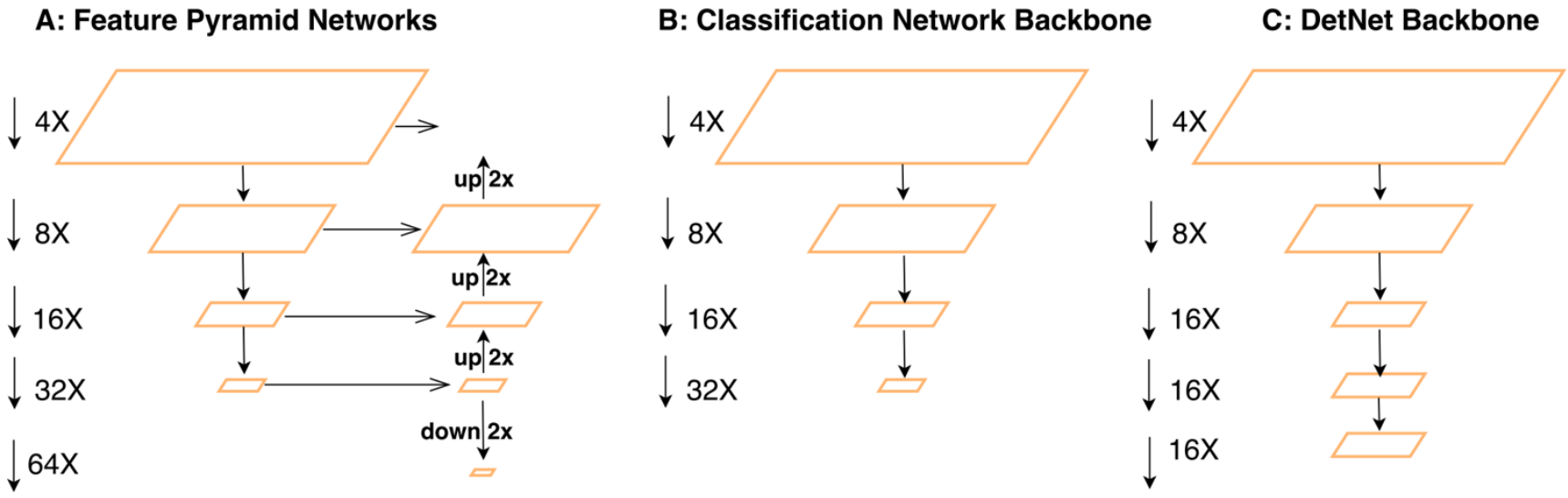
DetNet: a backbone network for object detection

- DetNet: designed directly for object detection and pretrained on ImageNet dataset (fig. C)
- Larger feature map size and also larger receptive field
- Better in locating large objects and detect small objects



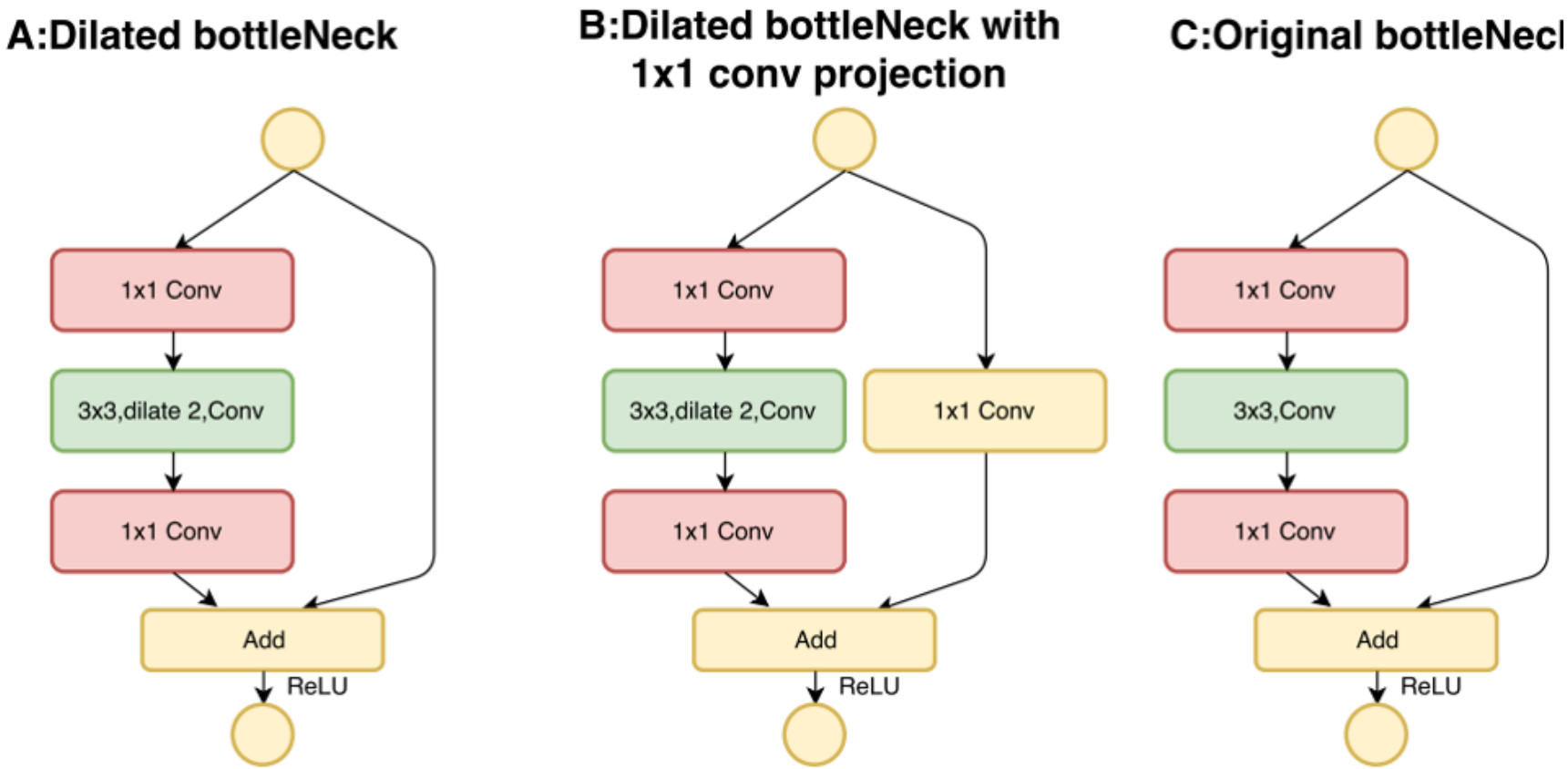
DetNet: a backbone network for object detection

- DetNet: designed directly for object detection and pretrained on ImageNet dataset (fig. C)
- Larger feature map size and also larger receptive field
- Better in locating large objects and detect small objects
- Overall structure like DeepLab, but for different tasks



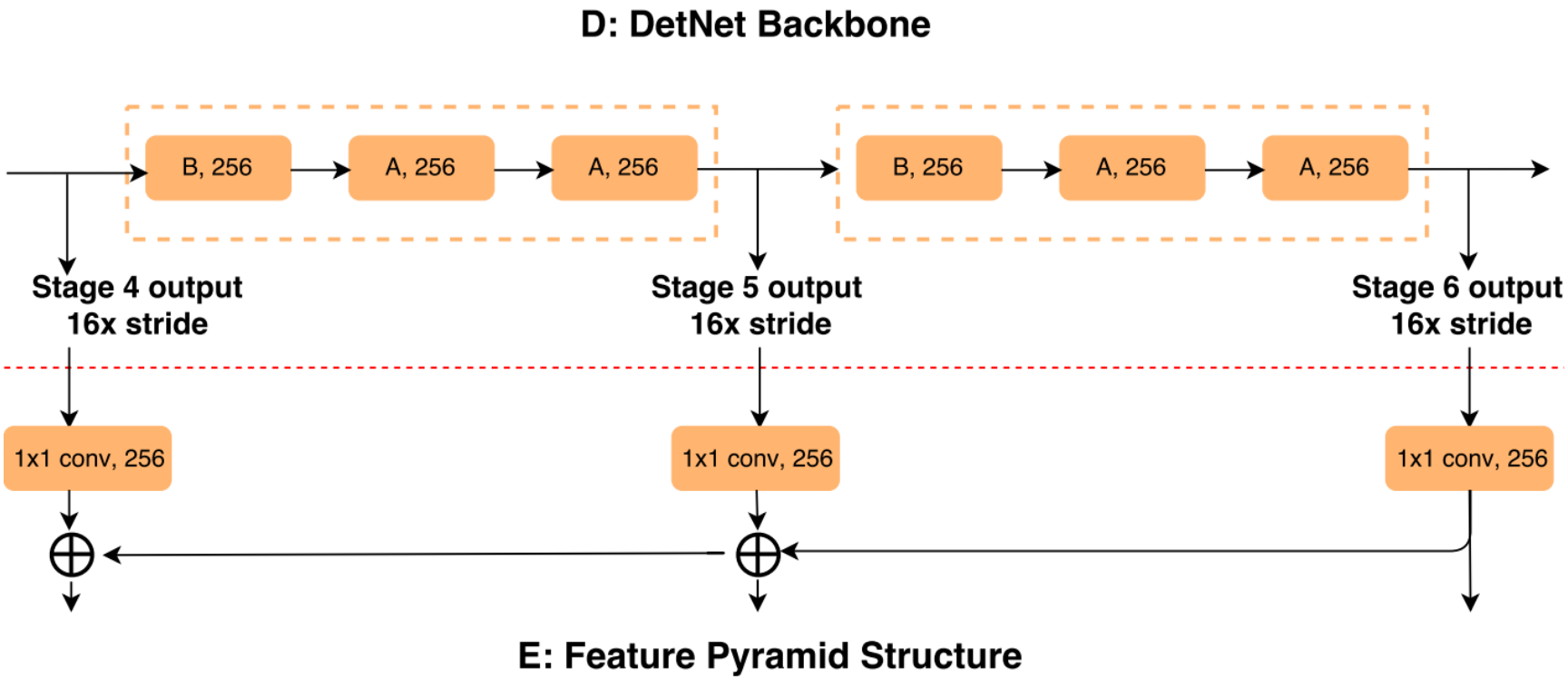
DetNet: a backbone network for object detection

- Novelty: dilated residual unit (A), dilated residual unit with 1×1 convolutional projection (B)
- Dilated convolution enlarges receptive field (as in DeepLab)



DetNet: a backbone network for object detection

- Easy to integrate DetNet with any detector, e.g., FPN
- 1×1 convolution (i.e., type B) helps create a new stage



DetNet with FPN: state-of-the-art result

- Large objects: accurate localization
- Small objects: detected, few missing

Summary

- From R-CNN to Mask R-CNN: faster, more accurate
- One-stage models: real-time, becoming more accurate
- Backbone network designed & pretrained specifically for detection
- Evolved fast! What are next innovations?

Further reading:

- Liu et al., Path aggregation network for instance segmentation, CVPR, 2018
- Hu et al., Learning to segment every thing, CVPR, 2018
- Wang et al., Region proposal by guided anchoring, arXiv, 2019

Comments on 1st assignment

- Be clear, specific, concise
- Both short (<100 words) and long (better 600-800) summary
- Highlight the key innovation in short summary
- Follow points listed on Week 1's last slide
- Say more about limitations and possible improvement
- Describe experiments in words, but concisely
- Just maximally one or two tables/figures/formulae
- If using figure, add figure caption below
- Delete redundant and non-academic information
- Not too many short paragraphs; not too long paragraphs
- Not use bullet points; link paragraphs smoothly
- 'we'/'I' to be replaced by 'the author(s)'/'the method', etc.
- Not select papers from low-rank conferences/journals