

CSE 252D: Advanced Computer Vision

Manmohan Chandraker

Lecture 17: Object Detection 3



Virtual classrooms

- Virtual lectures on Zoom
 - Only host shares the screen
 - Keep video off and microphone muted
 - But please do speak up (remember to unmute!)
 - Slides uploaded on webpage just before class
- Virtual interactions on Zoom
 - Ask and answer plenty of questions
 - “Raise hand” feature on Zoom when you wish to speak
 - Post questions on chat window
 - Happy to try other suggestions!
- Lectures recorded and upload on Canvas
 - Available under “My Media” on Canvas

Overall goals for the course

- Introduce the state-of-the-art in computer vision
- Study principles that make them possible
- Get understanding of tools that drive computer vision
- Enable one or all of several such outcomes
 - Pursue higher studies in computer vision
 - Join industry to do cutting-edge work in computer vision
 - Gain appreciation of modern computer vision technologies
- This is a great time to study computer vision!

Papers for Fri, May 28

- You Only Look Once: Unified, Real-Time Object Detection
 - <https://arxiv.org/abs/1506.02640>
- Focal Loss for Dense Object Detection
 - <https://arxiv.org/abs/1708.02002>
- Single-Shot Refinement Neural Network for Object Detection
 - <https://arxiv.org/abs/1711.06897>
- CornerNet: Detecting Objects as Paired Keypoints
 - <https://arxiv.org/abs/1808.01244>

Papers for Wed, Jun 2

- Playing for Benchmarks
 - <https://arxiv.org/abs/1709.07322>
- Domain Adaptive Faster R-CNN for Object Detection in the Wild
 - <https://arxiv.org/abs/1803.03243>
- Fully Convolutional Adaptation Networks for Semantic Segmentation
 - <https://arxiv.org/abs/1804.08286>
- Label Efficient Learning of Transferable Representations across Domains and Tasks
 - <https://arxiv.org/abs/1712.00123>

Papers for Fri, Jun 4

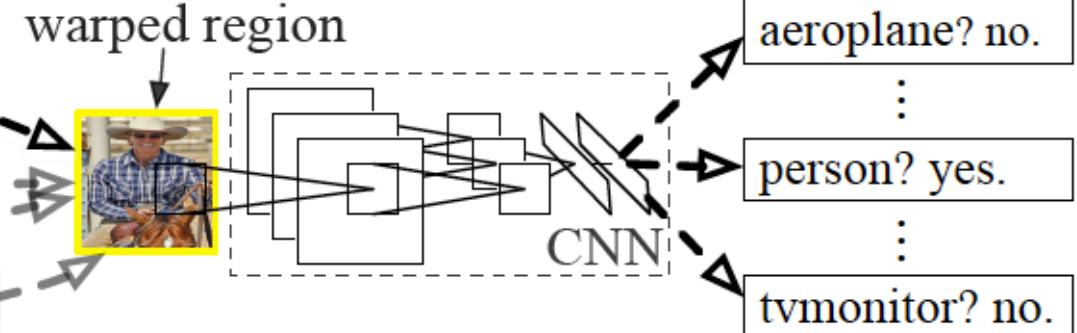
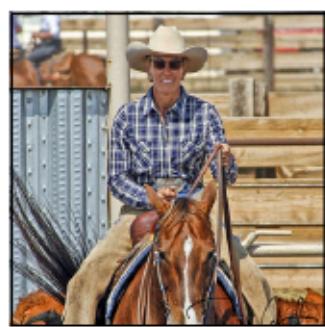
- A Simple Framework for Contrastive Learning of Visual Representations
 - <https://arxiv.org/abs/2002.05709>
- Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation
 - <https://arxiv.org/abs/1805.09806>
- Learning to Cluster Faces on an Affinity Graph
 - <https://arxiv.org/abs/1904.02749>
- UPSNet: A Unified Panoptic Segmentation Network
 - <https://arxiv.org/abs/1901.03784>
- Towards Universal Object Detection by Domain Attention
 - <https://arxiv.org/abs/1904.04402>

Announcements

- Final exam: Fri, Jun 11, 7-9:30pm
 - Held “in-class”, on Zoom
 - If time does not work for you, contact instructor by Wed, Jun 2
- Require you to apply or generalize concepts from class
 - Content: topics covered in lectures and lightning presentations
 - Exam will be open notes, can refer any material
 - Keep pen and paper handy
 - Only restriction: no communication with another human ☺
 - Enjoy the final as a celebration of things you learned in the course
- Sample final questions will be posted on course webpage
 - Some of those questions will be in the actual final!

Recap

R-CNN



1. Input image

2. Extract region proposals (~2k)

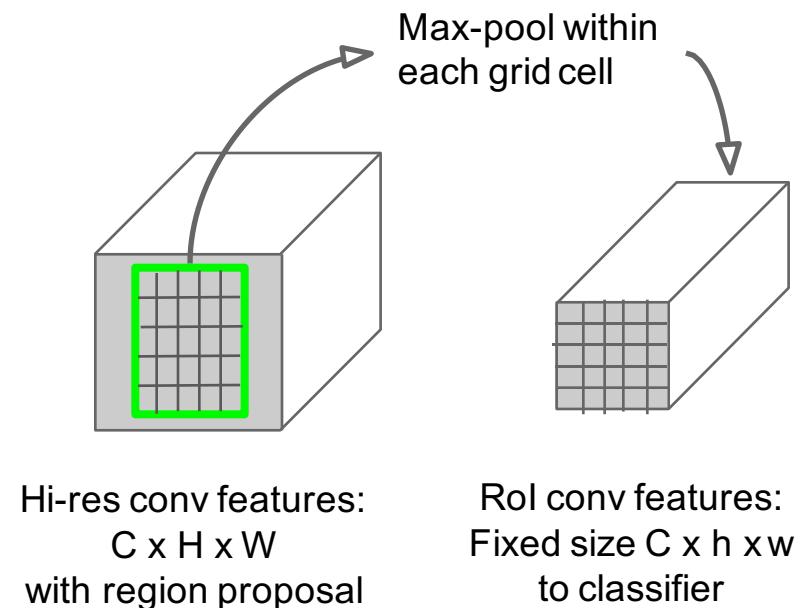
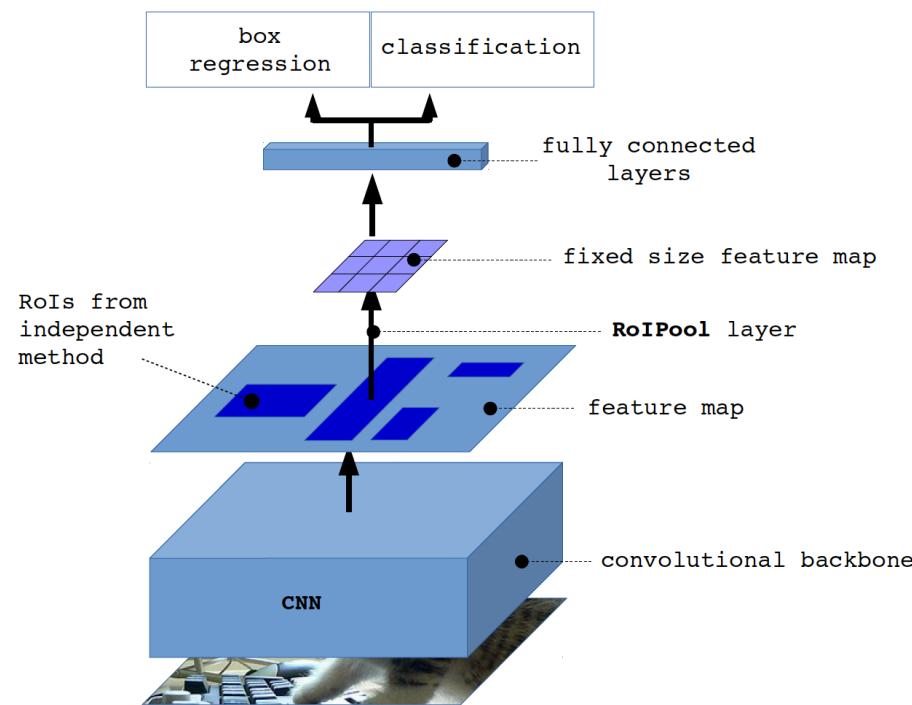
3. Compute CNN features

4. Classify regions

- Expensive
 - No feature reuse
 - Full forward pass of CNN for each region proposal
- Classification and regression are disjoint from feature extraction
 - CNN features not updated together with detection
- Complex training pipeline

[Girshick et al., Rich feature hierarchies]

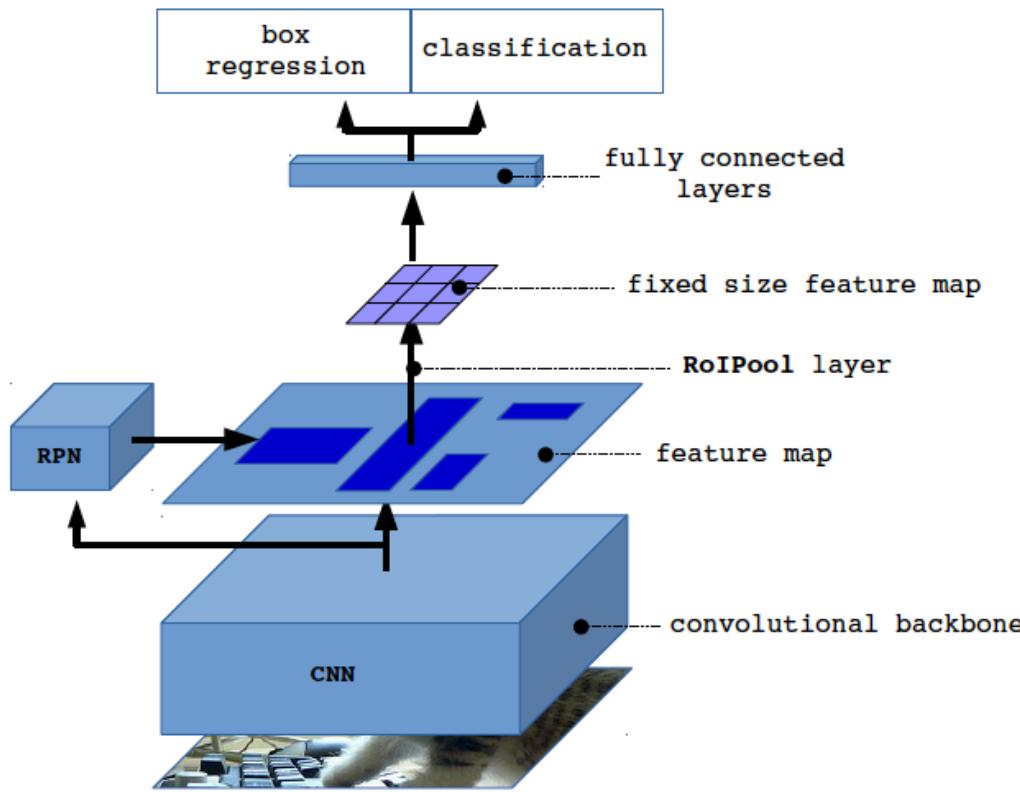
Fast R-CNN



- **RoI pooling:** connects convolutional layers to classifier
 - Features and classifier trained together
- Feature extraction just once for the whole image
 - Faster inference and training
- Proposals still from external mechanism

[Girschick, Fast R-CNN]

Faster R-CNN

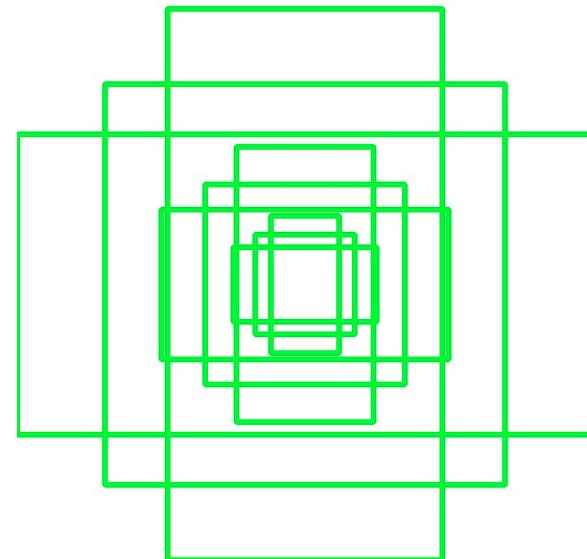
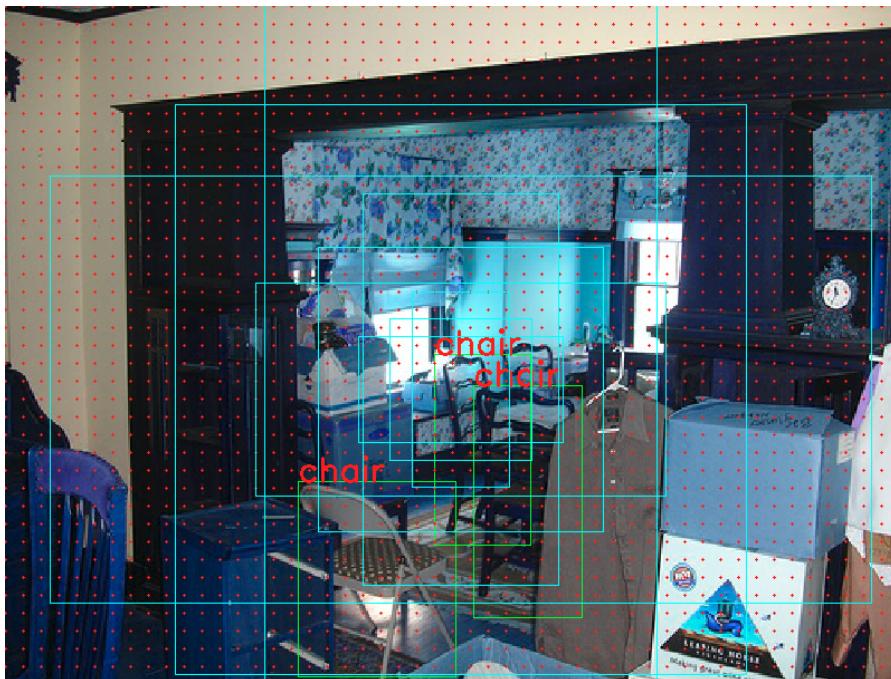


- Proposals still from separate mechanism in Fast R-CNN
- Insert a **Region Proposal Network (RPN)** after last convolutional layer
- RPN trained to produce region proposals directly, no need for external region proposals!
- After RPN, use RoI Pooling and an upstream classifier and regressor just like Fast R-CNN

[Ren et al., Faster R-CNN]

Faster R-CNN: Anchors

- Consider single object localization
 - Simply have a classification and regression head
- But now consider multiple objects, possibly overlapping
 - Regression for objects will interfere with each other
- Anchors: a set of reference positions on the feature map
 - Anchor with high ground truth overlap responsible for regressing position
 - Determines reference and spatial extent for predicting object

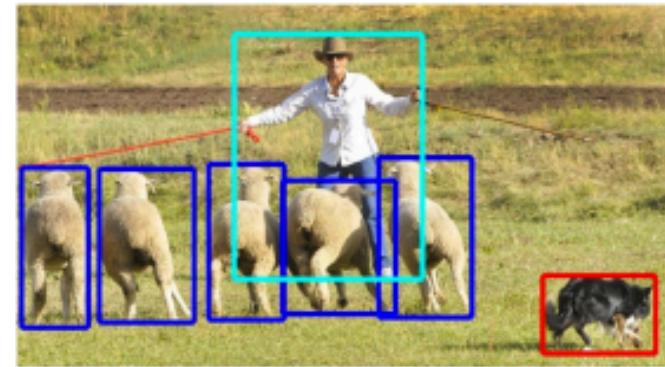


Typically: 3 scales and 3 aspect ratios

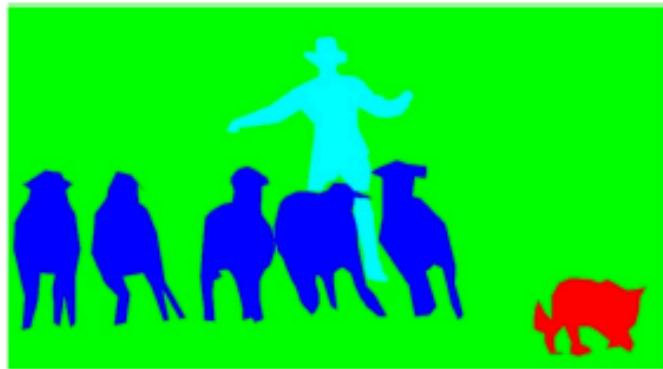
Instance Segmentation



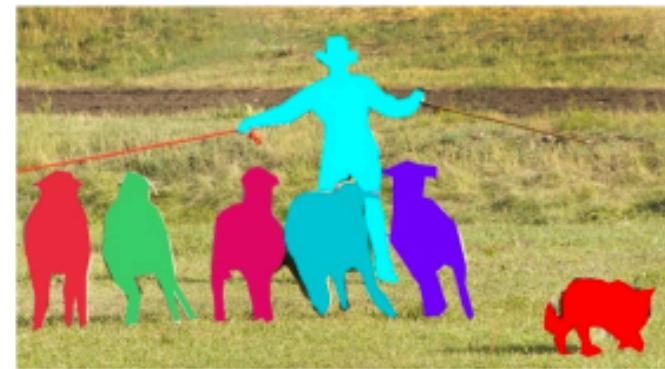
Image classification



Object detection



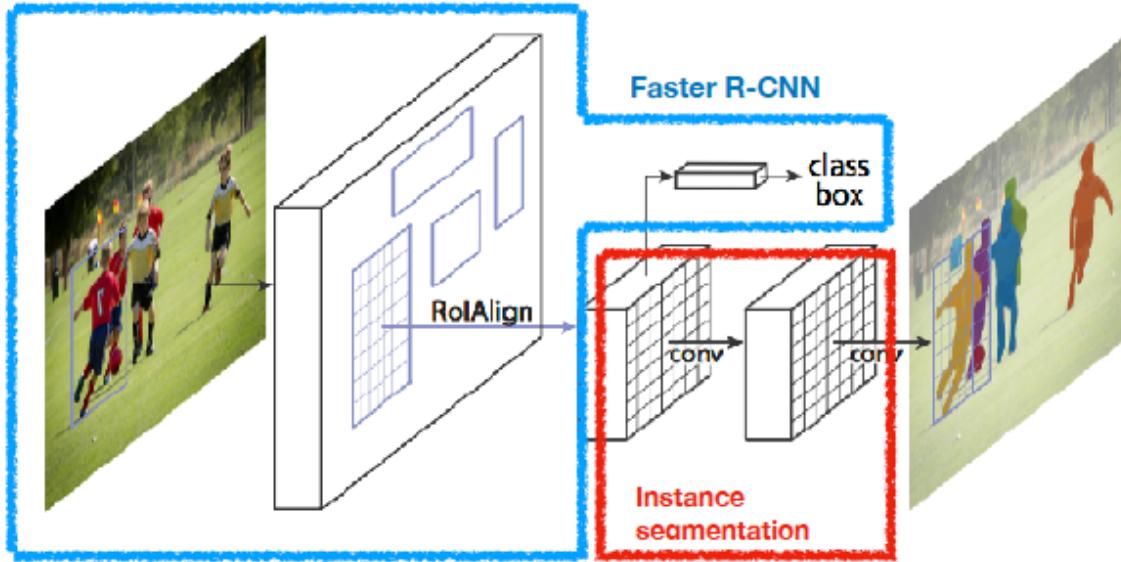
Semantic segmentation



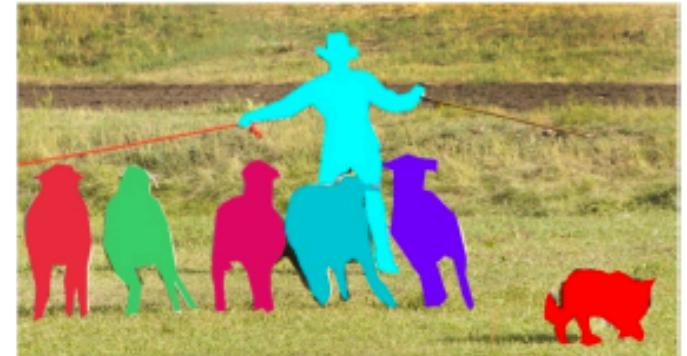
Instance segmentation

[Lin et al., MS COCO]

Mask R-CNN for Instance Segmentation



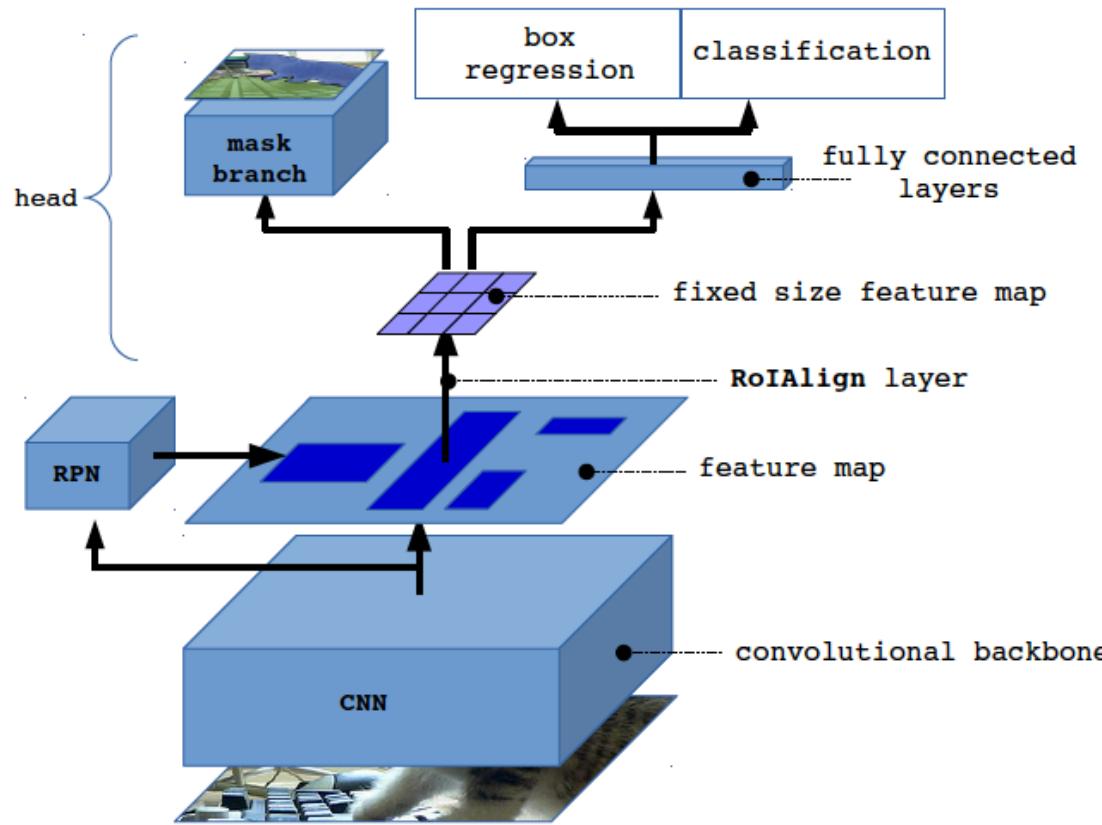
- Two choices:
 - Predict instance first, then classify
 - Segment first, then break into instances
- Mask R-CNN is instance-first
- Also applicable to human pose estimation with keypoints as one-hot masks



Instance segmentation

[He et al., Mask R-CNN]

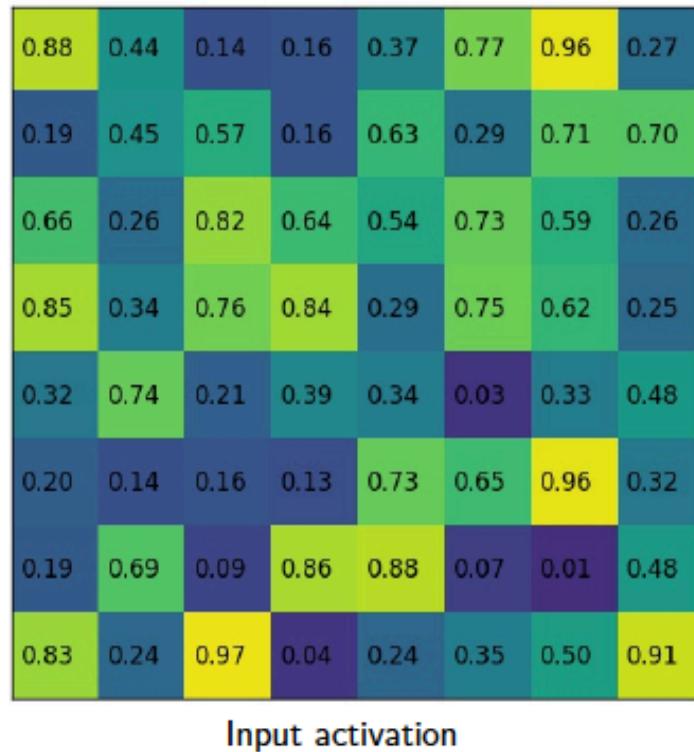
Mask R-CNN



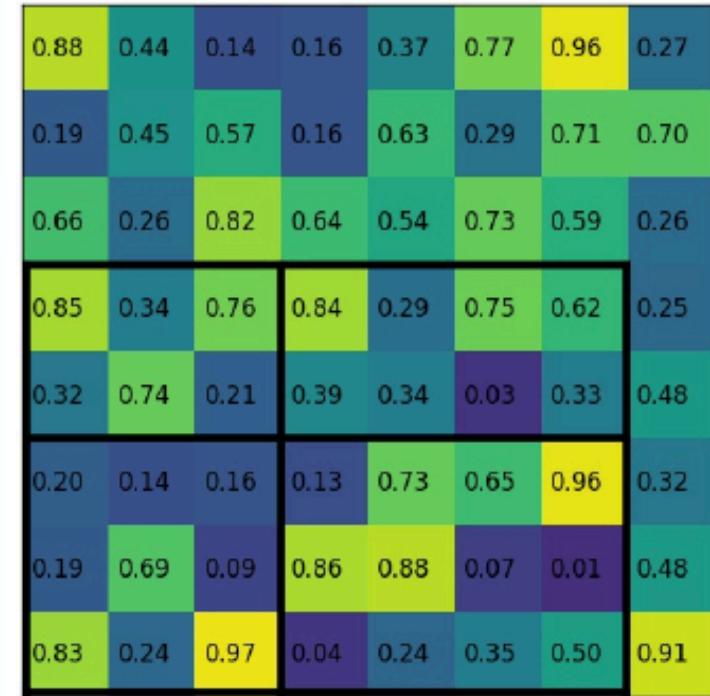
- Predict mask in parallel to class (rather than before class)
- Decouple mask (FCN) and class prediction (Faster R-CNN)
- ROIAlign: quantization-free for better localization
- Training (2 days on 8 GPUs) and inference (200ms) speed

[He et al., Mask R-CNN]

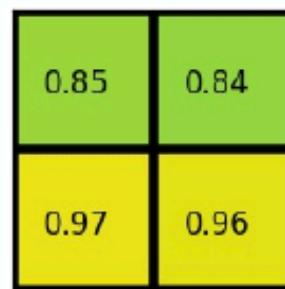
RoI-Pooling in Faster R-CNN



Faster R-CNN
RoIPool



Region projection and pooling sections

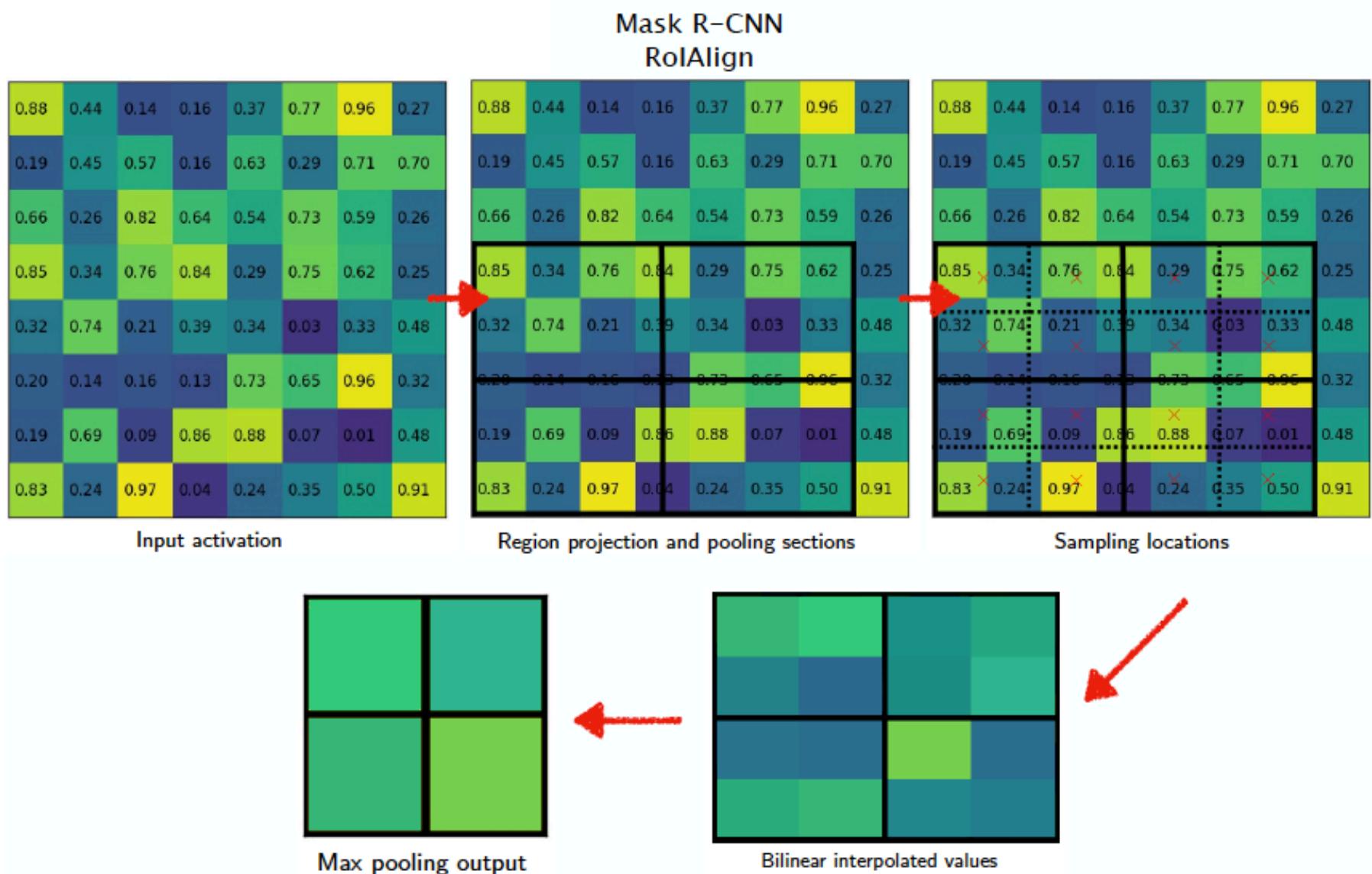


Max pooling output

[Credit: deepsense.io]

Fine for classification which is robust to small translations, but need better for pixel-level masks

RoI-Align in Mask R-CNN



Decoupled segmentation and classification

- Decoupling achieved with per-class binary masks (sigmoid)
- Alternative in FCN: multinomial masks (softmax)

$$L = \underline{L_{cls}} + \underline{L_{box}} + \underline{L_{mask}}$$

A. Fast R-CNN B.

$K \cdot (m \times m)$ sigmoid outputs:
→ pixel-wise binary classification
→ one mask for each class, no competition

L_{mask} : mean binary cross-entropy

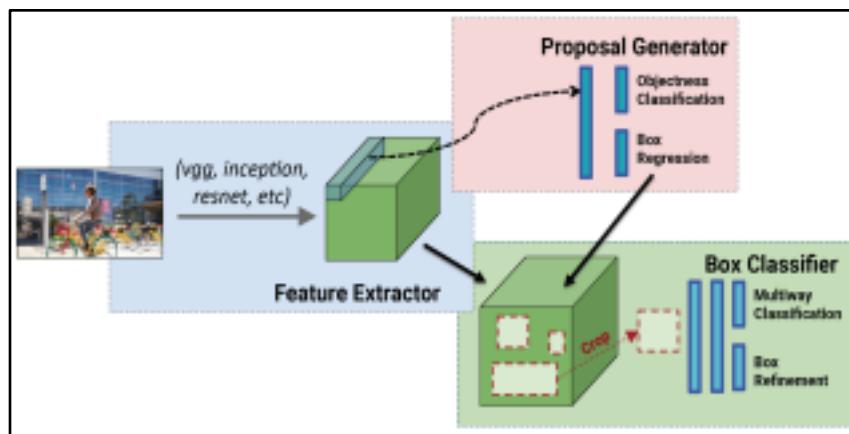
- Generate mask for every class without competition between classes
- Dedicated classification branch to predict class label to select output mask

	AP	AP ₅₀	AP ₇₅
<i>softmax</i>	24.8	44.1	25.1
<i>sigmoid</i>	30.3	51.2	31.5
	+5.5	+7.1	+6.4

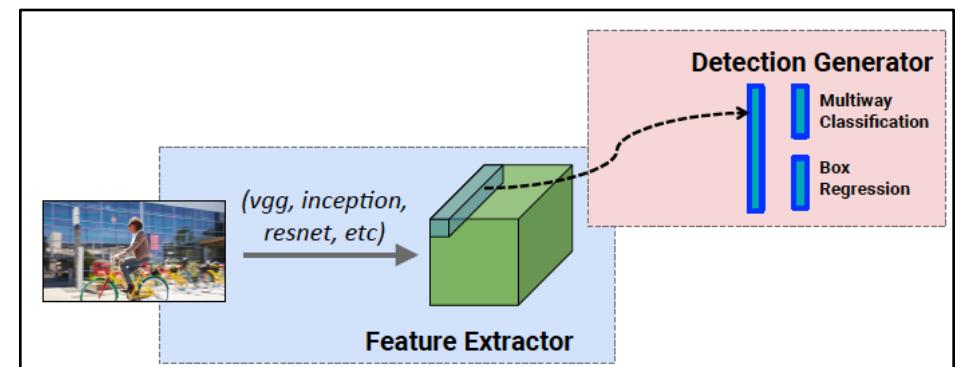


Single-Shot Detector

- Two step process of Faster R-CNN can be expensive
 - Box classification and regression are done twice
 - RPN produces boxes, but they are used to pool features
 - Separate classifier then used for evaluation
- Key motivation for SSD: achieve good accuracy-speed trade-off
 - Use something similar to RPN for directly scoring anchors
 - Extract anchors at multiple scales to get accuracy with minimal overhead

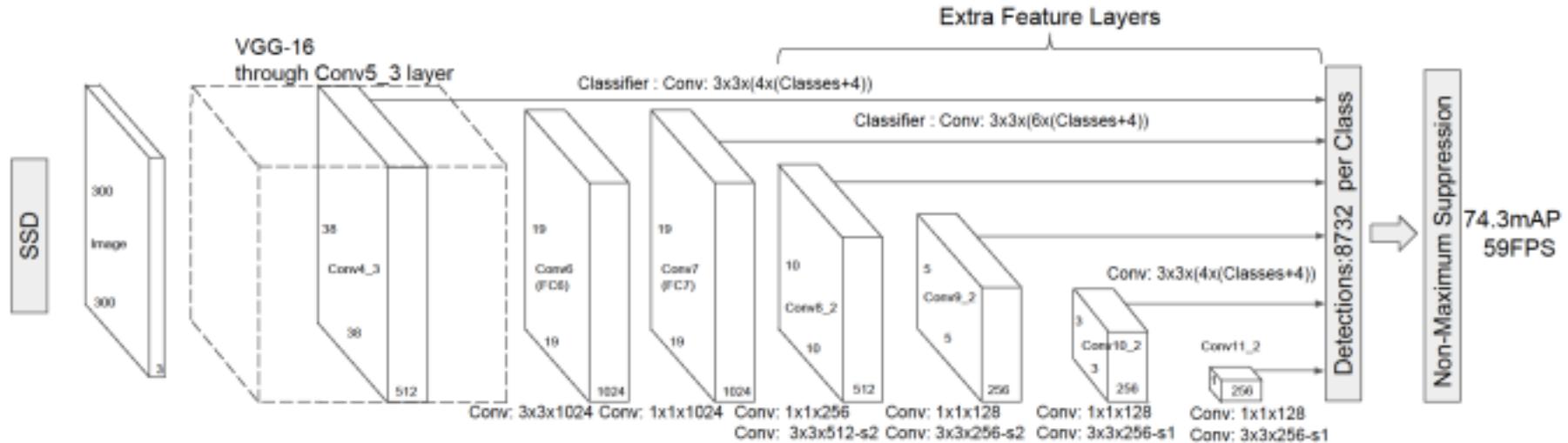


Faster R-CNN



SSD

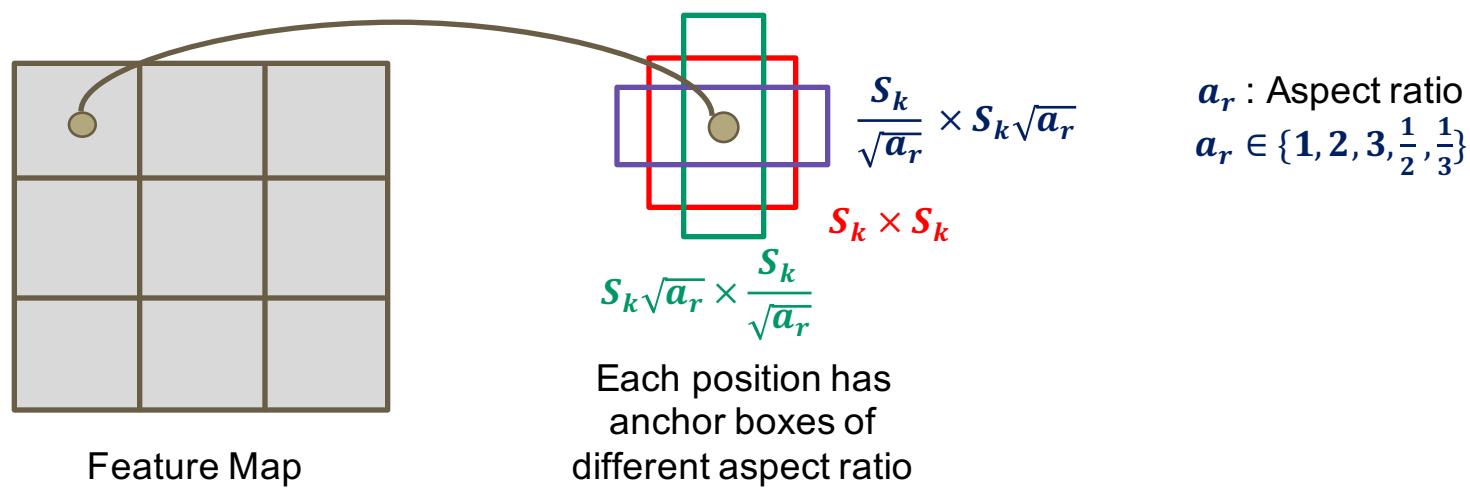
Single-Shot Detector



- Standard VGG (or ResNet) base network
- Add further layers with progressively decreasing size
 - Used for predicting detections at multiple scales
 - Layers with wider receptive fields expected to detect larger objects

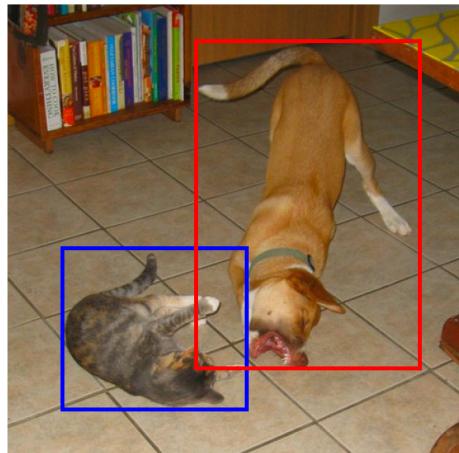
[Liu et al., SSD]

SSD: Default boxes (anchors)

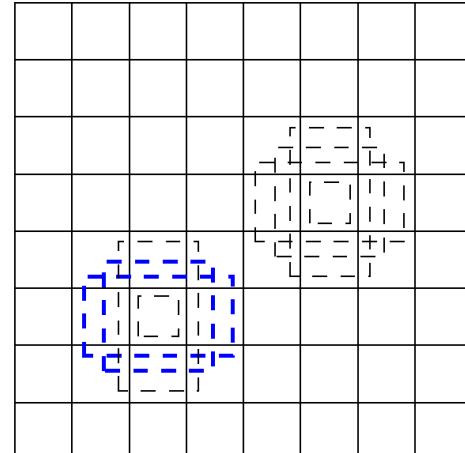


- SSD uses default boxes, which are similar to anchors in Faster R-CNN
- Default boxes located at each cell on the feature map
 - Multiple boxes corresponding to different aspect ratios
- Default boxes across feature layers for better discretization of scale

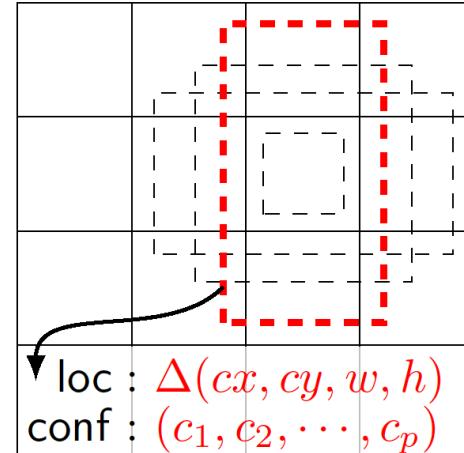
SSD: Default boxes (anchors)



(a) Image with GT boxes



(b) 8×8 feature map



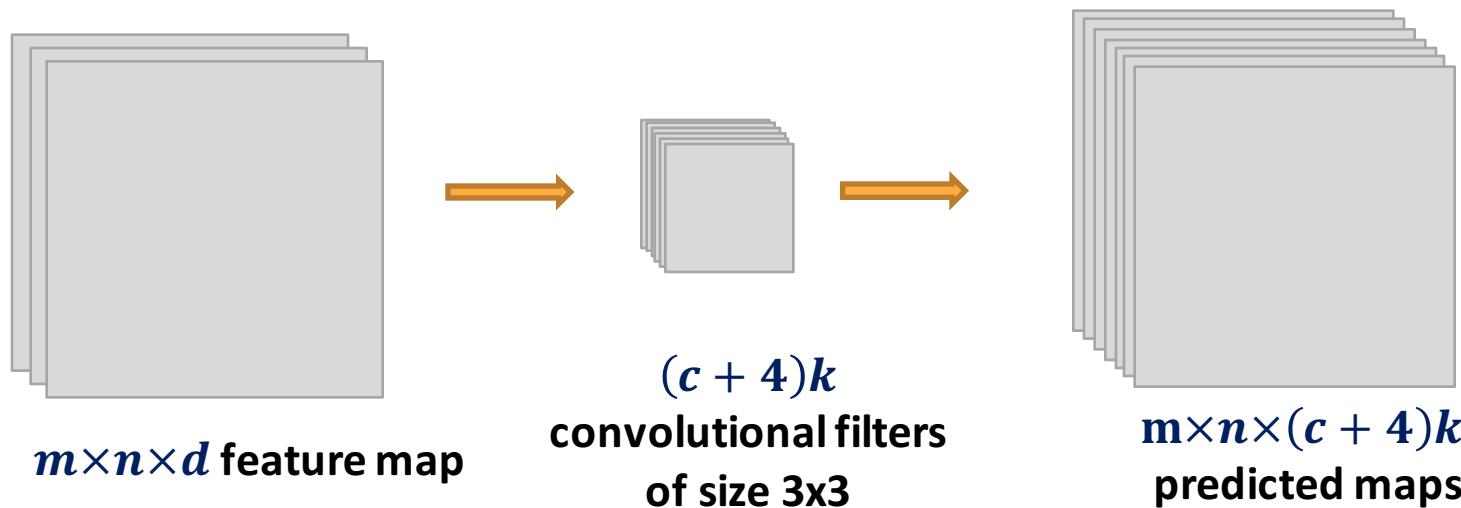
conf : (c_1, c_2, \dots, c_p)

(c) 4×4 feature map

- Anchor boxes of different feature maps have different scales
 - Various feature maps responsible for detecting objects of different sizes
- Let m be the number of feature maps used for prediction
- Let s_k denote the scale of feature map k
- Define $s_{\min} = 0.2$ and $s_{\max} = 0.9$

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m - 1}(k - 1), \quad k \in [1, m]$$

SSD: Convolution prediction at multiple scales



- On each $m \times n \times d$ feature map, two types of $3 \times 3 \times d$ convolutional filters applied
 - Goal for the filter: assign a class and regress location for each default box
 - c filters for class prediction, where c is the number of object categories
 - 4 filters for bounding box regression, for coordinates x, y, w, h
- With k default boxes at every cell, there are $(c+4)k$ filters for each feature map
- Output for $m \times n$ feature map is a $m \times n \times (c+4)k$ map
 - Represents category label and regression offsets for each default box

SSD: Training

- Need to match ground truth to default boxes to allow training
- Ground truth positives: match ground truth to each default box with IoU > 0.5
- Ground truth negatives: all other default boxes

SSD: Training

- Need to match ground truth to default boxes to allow training
- Ground truth positives: match ground truth to each default box with IoU > 0.5
- Ground truth negatives: all other default boxes
- Loss: Combination of classification confidence and localization accuracy
- Classification loss is softmax for multi-class prediction

$$L_{conf}(x, c) = - \sum_{i \in Pos}^N x_{ij}^p \log(\hat{c}_i^p) - \sum_{i \in Neg} \log(\hat{c}_i^0) \quad \text{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$$

$x_{ij}^p = 1$ if default box i matches ground truth box j of class p , otherwise 0

- Smooth L1 loss for localization

$$L_{loc}(x, l, g) = \sum_{i \in Pos}^N \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k \text{smooth}_{L1}(l_i^m - \hat{g}_j^m)$$

Offsets Predicted location Ground truth for box j

SSD: Training

- Need to match ground truth to default boxes to allow training
- Ground truth positives: match ground truth to each default box with IoU > 0.5
- Ground truth negatives: all other default boxes
- Loss: Combination of classification confidence and localization accuracy
- Classification loss is softmax for multi-class prediction

$$L_{conf}(x, c) = - \sum_{i \in Pos}^{N} x_{ij}^p \log(\hat{c}_i^p) - \sum_{i \in Neg} \log(\hat{c}_i^0) \quad \text{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$$

$x_{ij}^p = 1$ if default box i matches ground truth box j of class p , otherwise 0

- Smooth L1 loss for localization

$$L_{loc}(x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k \text{smooth}_{L1}(l_i^m - \hat{g}_j^m)$$

Offsets Predicted location Ground truth for box j

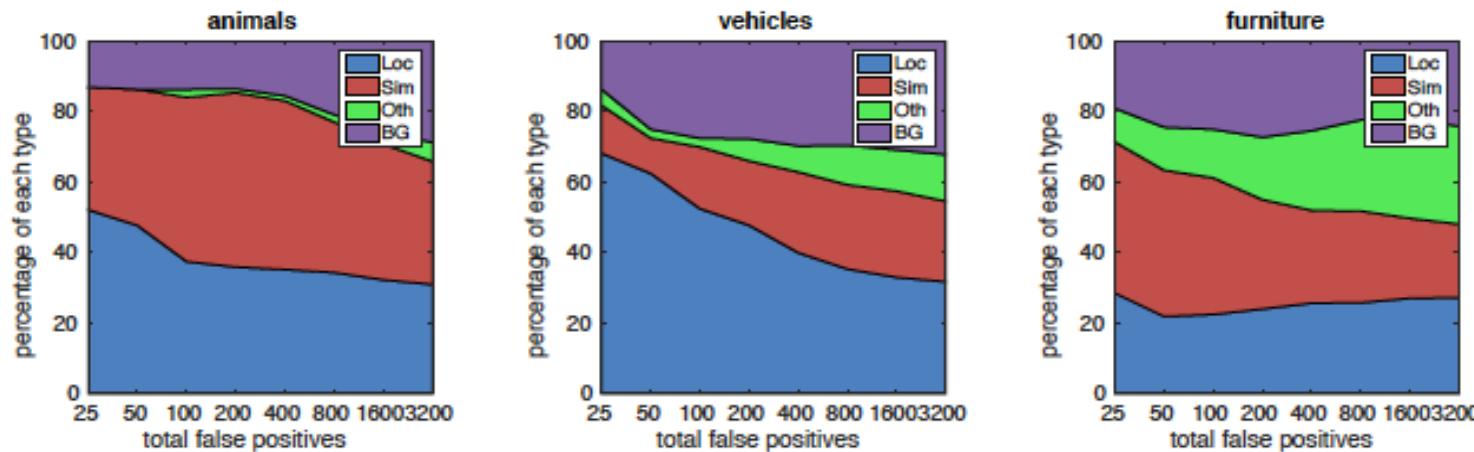
- Number of negatives far exceeds positives
 - Hard negative mining: choose default boxes with highest classification loss

SSD: Analysis

- Helps to use multiple layers for prediction

Prediction source layers from:							mAP		# Boxes
	conv4_3	conv7	conv8_2	conv9_2	conv10_2	conv11_2	use boundary boxes?		
Yes	No								
✓	✓	✓	✓	✓	✓	✓	74.3	63.4	8732
✓	✓	✓	✓	✓	✓		74.6	63.1	8764
✓	✓	✓	✓	✓			73.8	68.4	8942
✓	✓	✓	✓				70.7	69.2	9864
✓	✓						64.2	64.4	9025
							62.4	64.0	8664

- In general, most errors in SSD are due to poorer localization
 - Explicit pooling of region proposals in Faster R-CNN prevents this
 - Possibly also cause for more confusion between similar categories in SSD



- SSD also tends to do better for larger objects compared to smaller ones

Object Detection: Accuracy and Speed

