Computer Vision

DATA.ML.300, 5 study credits

Esa Rahtu
Unit of Computing Sciences, Tampere University

Object category detection

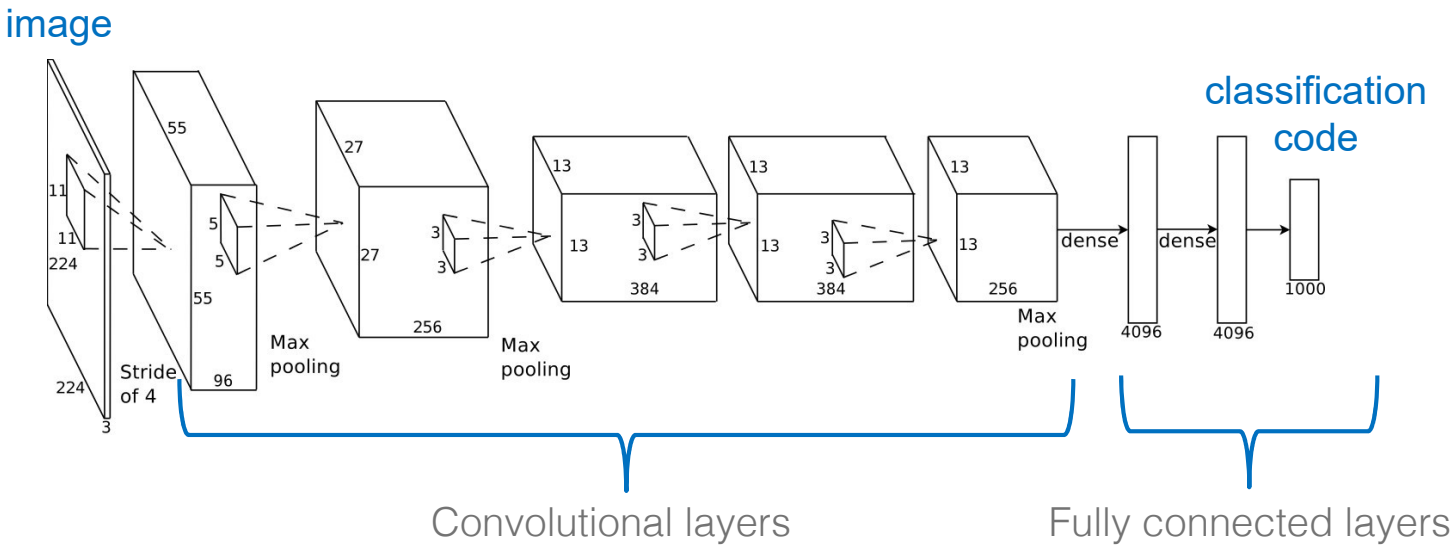
Outline for this week

- Lecture 1: Principles of sliding window detectors
  - Training a sliding window detector
  - Speeding up inferences
- **Lecture 2: Deep networks for object category detection**
  - Two-stage and one-stage networks
  - State-of-the-art

Background

Reminder: Classification CNNs

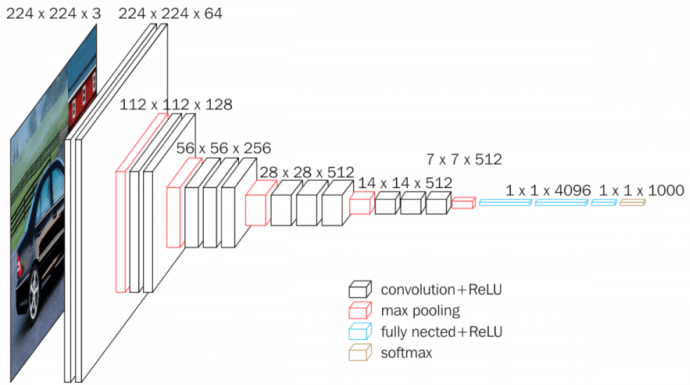
AlexNet (Krizhevsky et al. 2012)

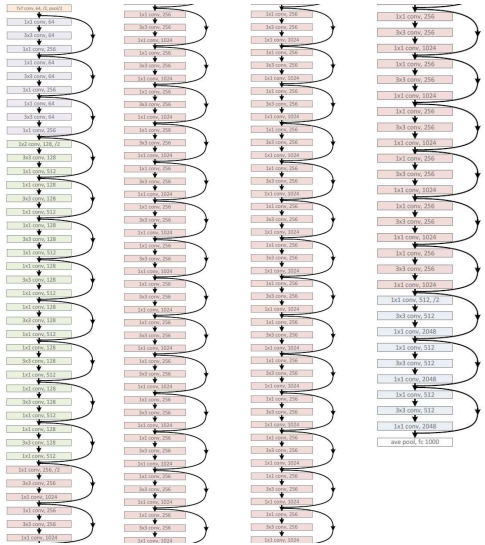


ImageNet classification with deep convolutional neural networks, Krizhevsky et al. NIPS 2012

60 Million parameters

Reminder: Classification CNNs

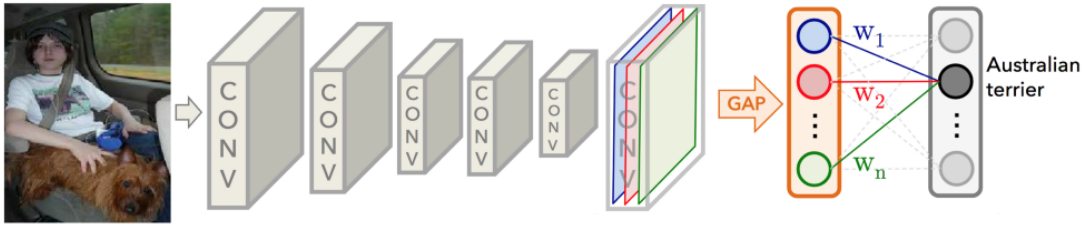




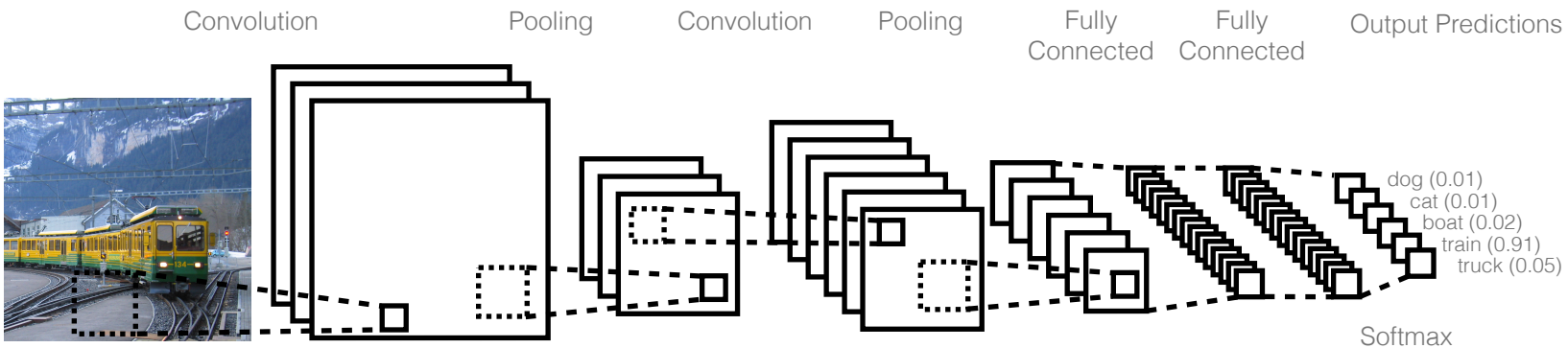
Very deep convolutional networks for large-scale image recognition, Simonyan et al. arXiv 2014
Deep residual learning for image recognition, He et al. CVPR 2016

CNNs for detection - intuition I

- Modern classification architectures, such as ResNet or Inception, use convolutional layers throughout
  - No fully connected layers
  - Less parameters
  - Feature vector by spatial pooling

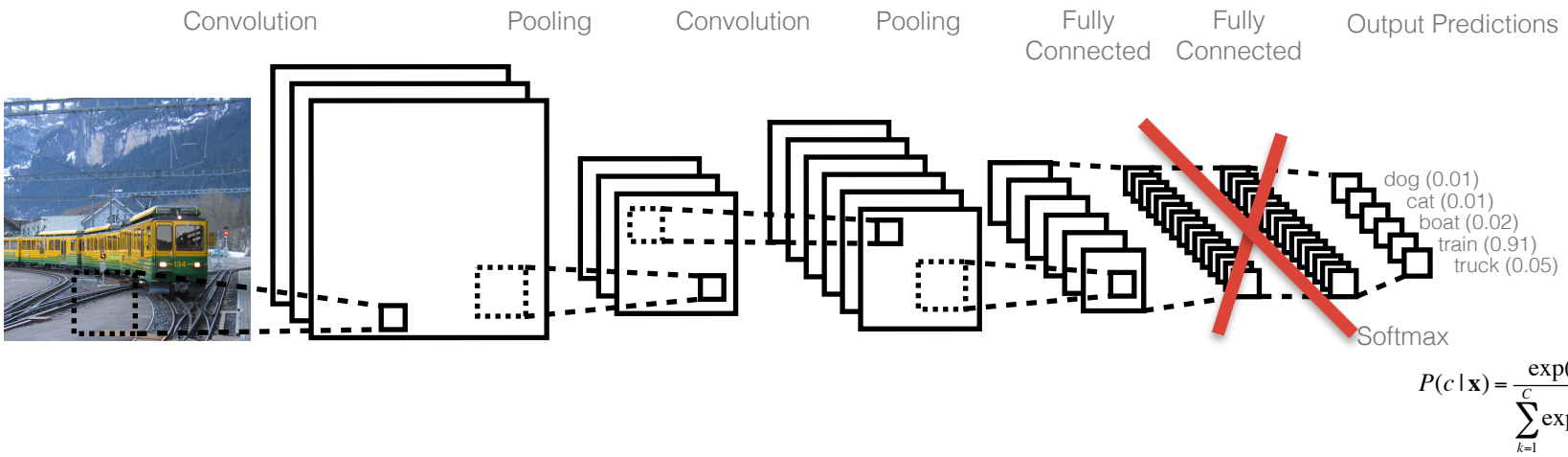


CNNs for detection - intuition I

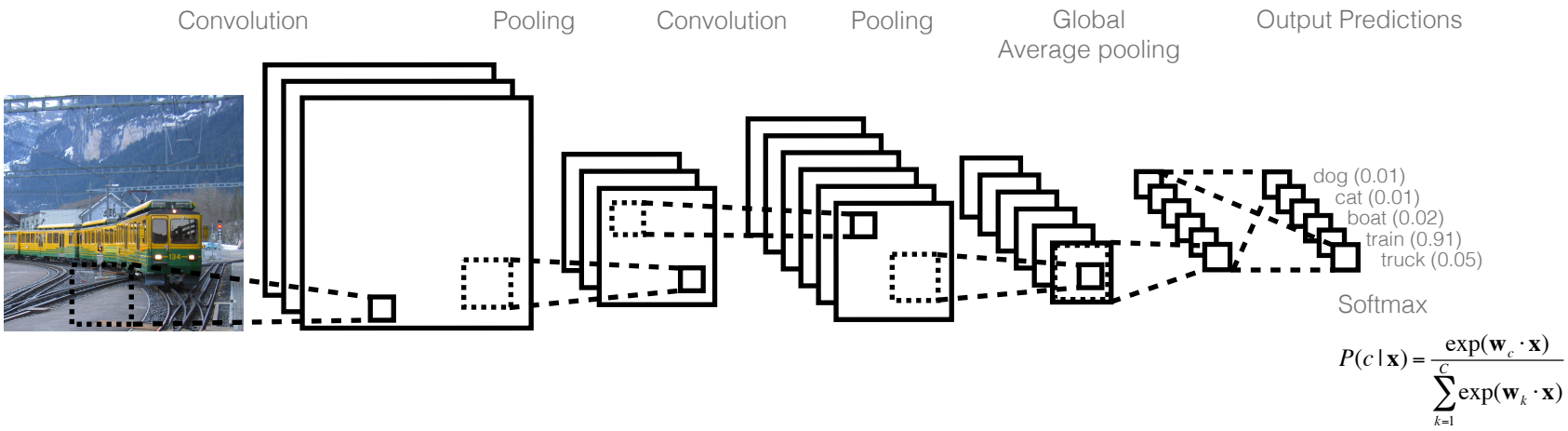


$$P(c \mid \mathbf{x}) = \frac{\exp(\mathbf{w}_c \cdot \mathbf{x})}{\sum_{k=1}^{C} \exp(\mathbf{w}_k \cdot \mathbf{x})}$$

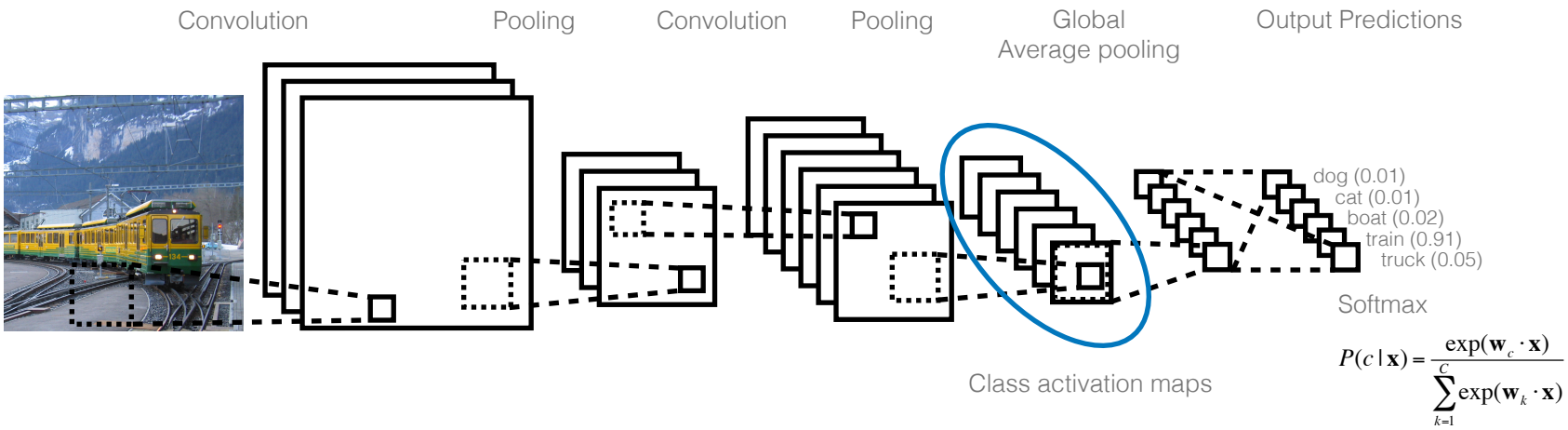
CNNs for detection - intuition I



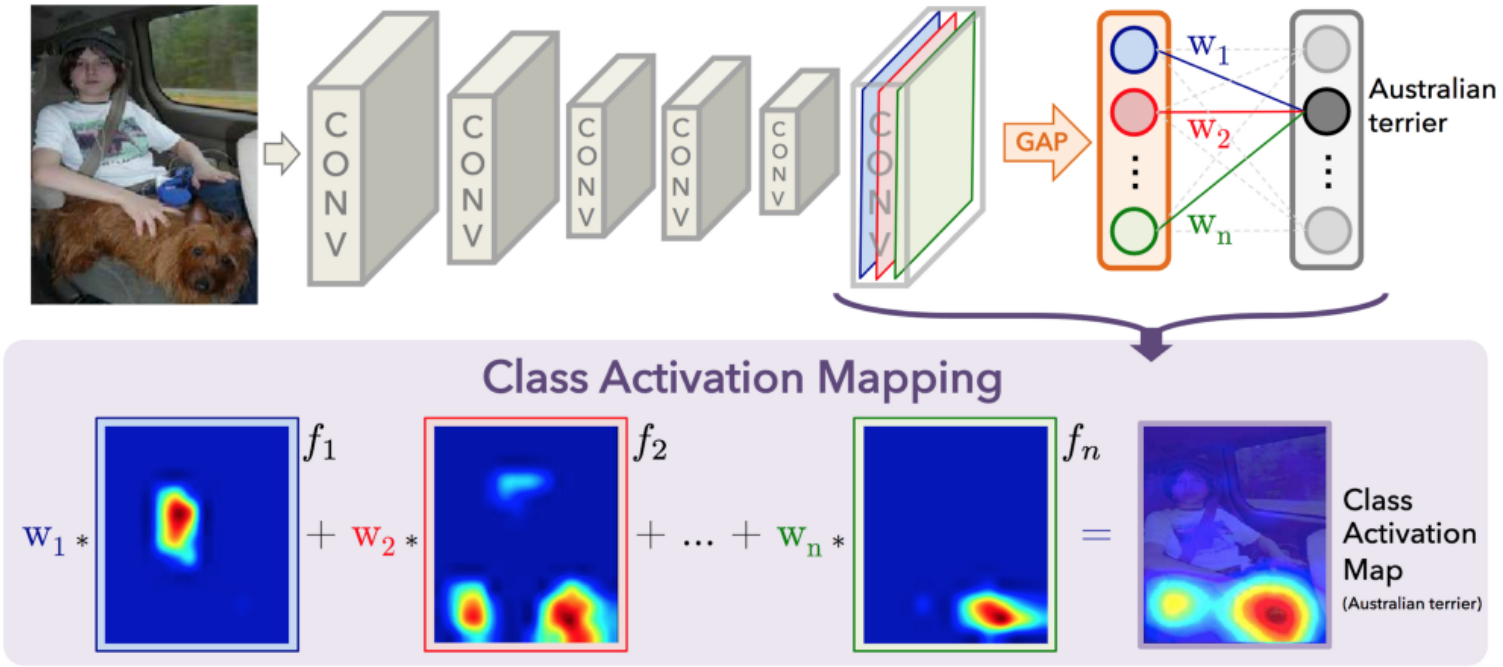
CNNs for detection - intuition I



CNNs for detection - intuition I

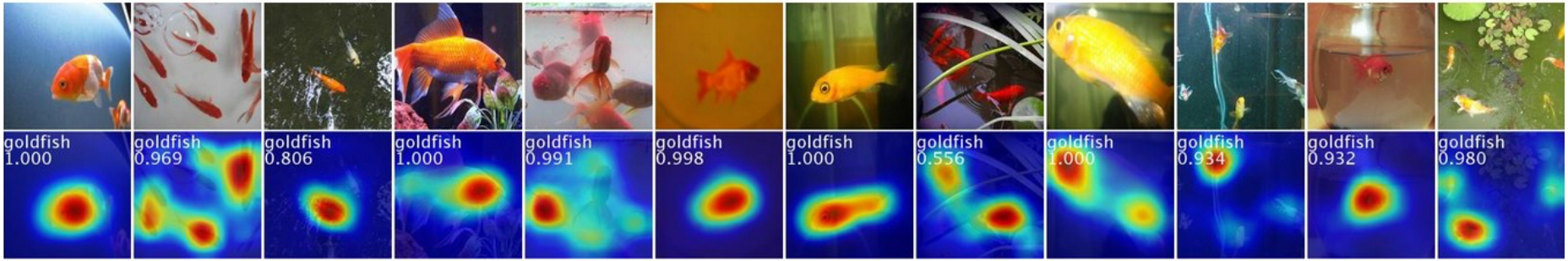


CNNs for detection - intuition II



Is object localisation for free?-weakly-supervised learning with convolutional neural networks, Oquab et al. CVPR 2015
Learning deep features for discriminative localisation, Zhou et al. CVPR 2016

CNNs for detection - intuition II



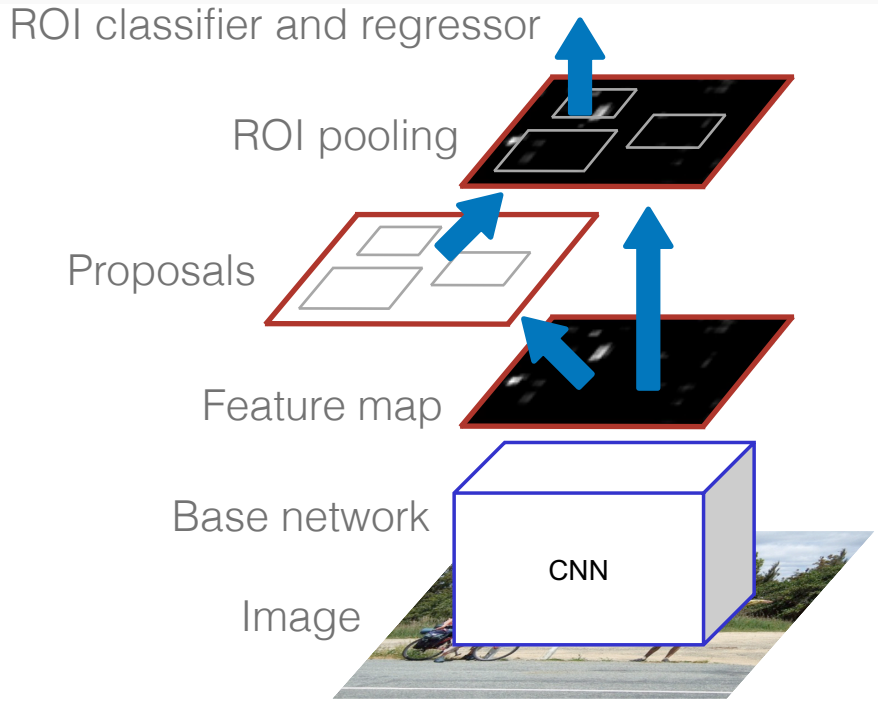
Object detection networks

Classical object detectors

- Two stage procedure:
  1. Propose class agnostic regions in the image (sliding window or proposals)
  2. Classify regions into object classes or background
- Can this be captured in a deep network?

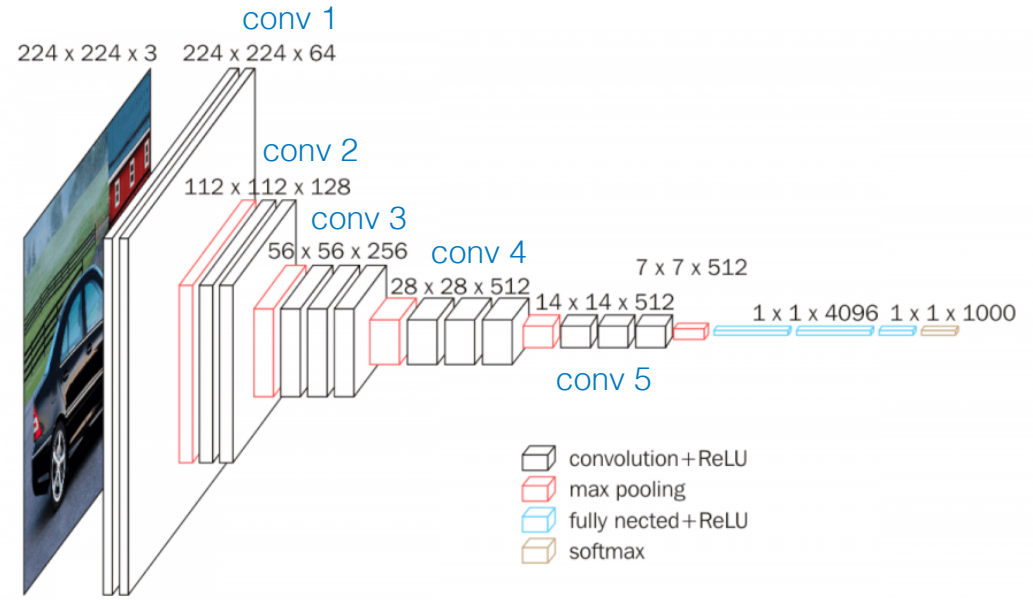
Faster R-CNN

- Two stage system:
  - Region proposal network (RPN)
  - Classification/regression network
- Base network VGG16



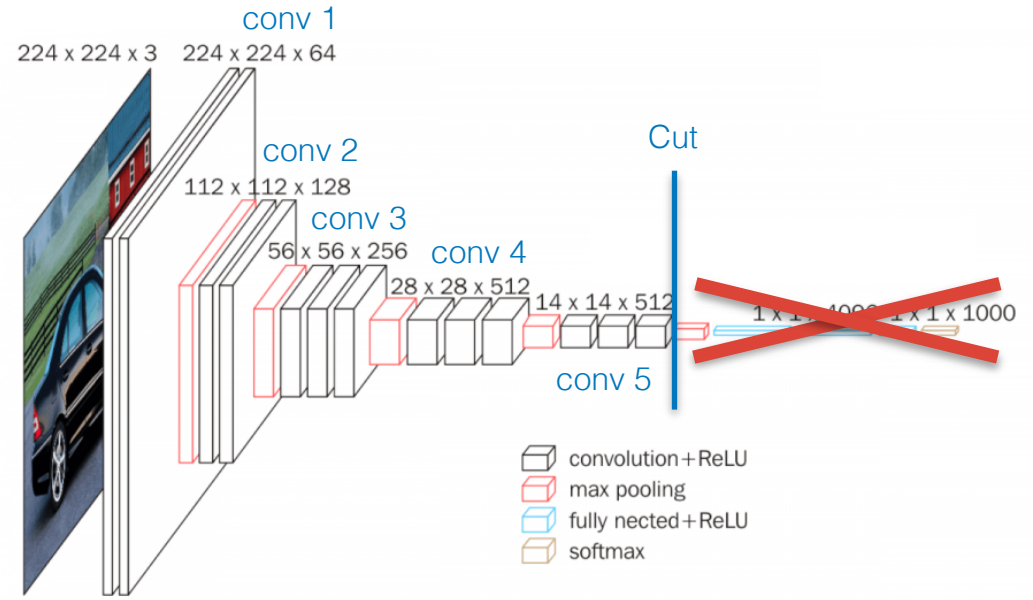
Faster R-CNN: Towards real-time object detection with region proposals, Ren et al. NIPS 2015

Reminder VGG-16



Very deep convolutional networks for large-scale image recognition, Simonyan et al. arXiv 2014

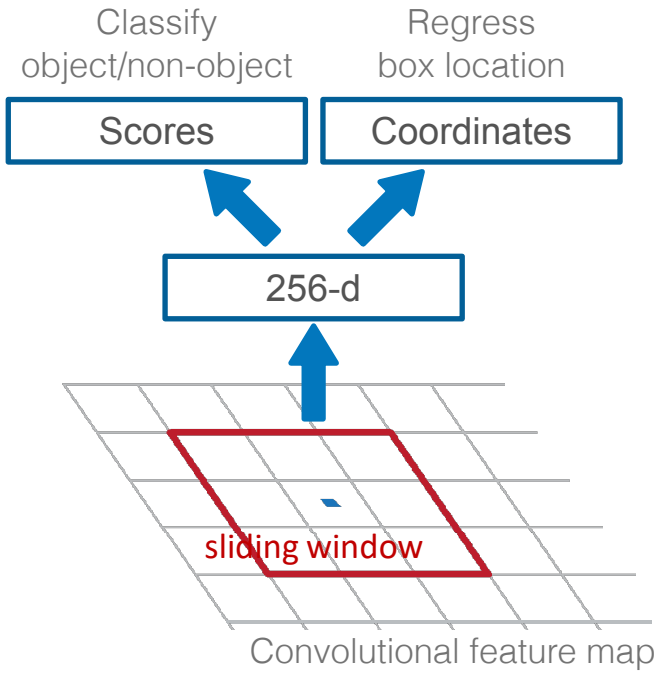
Reminder VGG-16



Very deep convolutional networks for large-scale image recognition, Simonyan et al. arXiv 2014

Region proposal network (RPN)

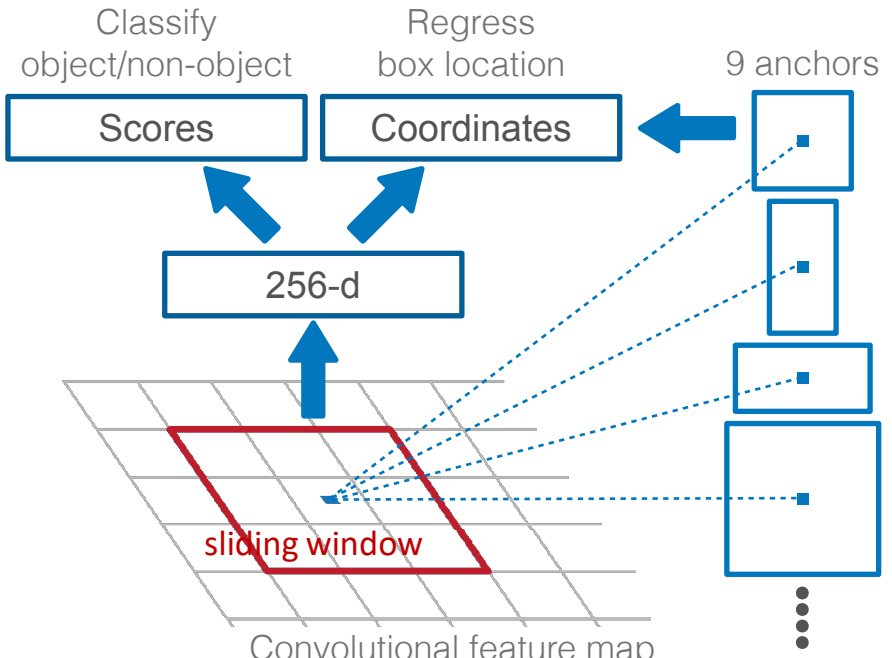
- Slide a small window on feature map
- Window position provides localisation **with reference to the image**
- Box regression provides finer localisation **with reference to window**



Faster R-CNN: Towards real-time object detection with region proposals, Ren et al. NIPS 2015

“Anchors”: predefined candidate regions

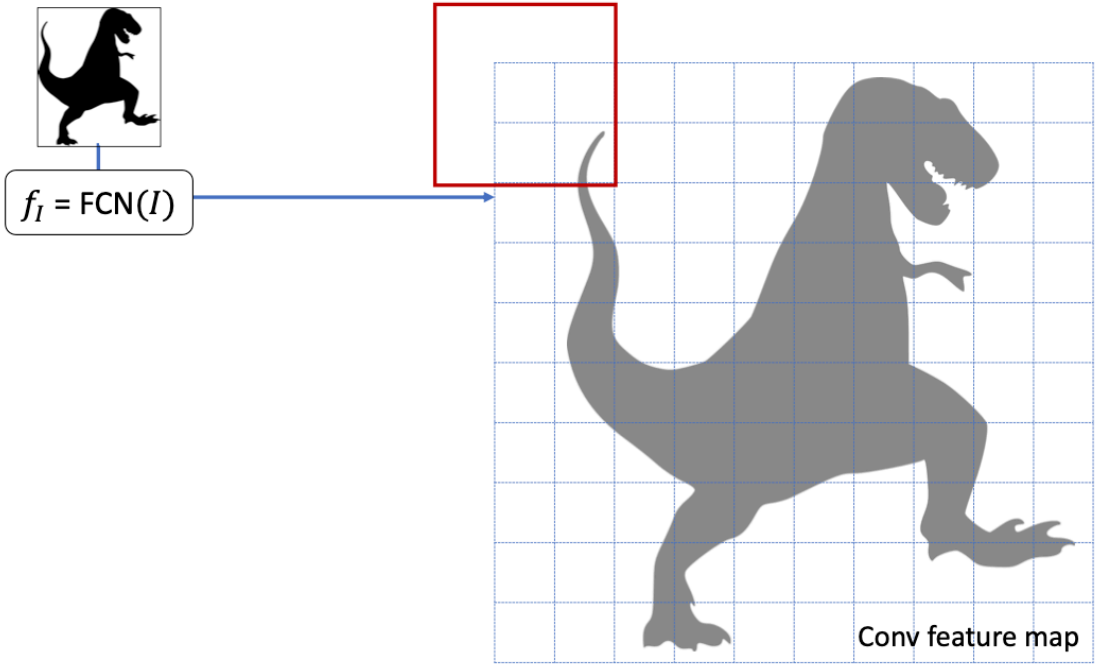
- Multi-scale/size anchors are used at each position: 3 scales x 3 aspect ratios yields 9 anchors
- Each anchor has its own prediction function
- **Single-scale** features, multi-scale predictions



Faster R-CNN: Towards real-time object detection with region proposals, Ren et al. NIPS 2015

Region proposal network (RPN)

RPN: Region Proposal Network

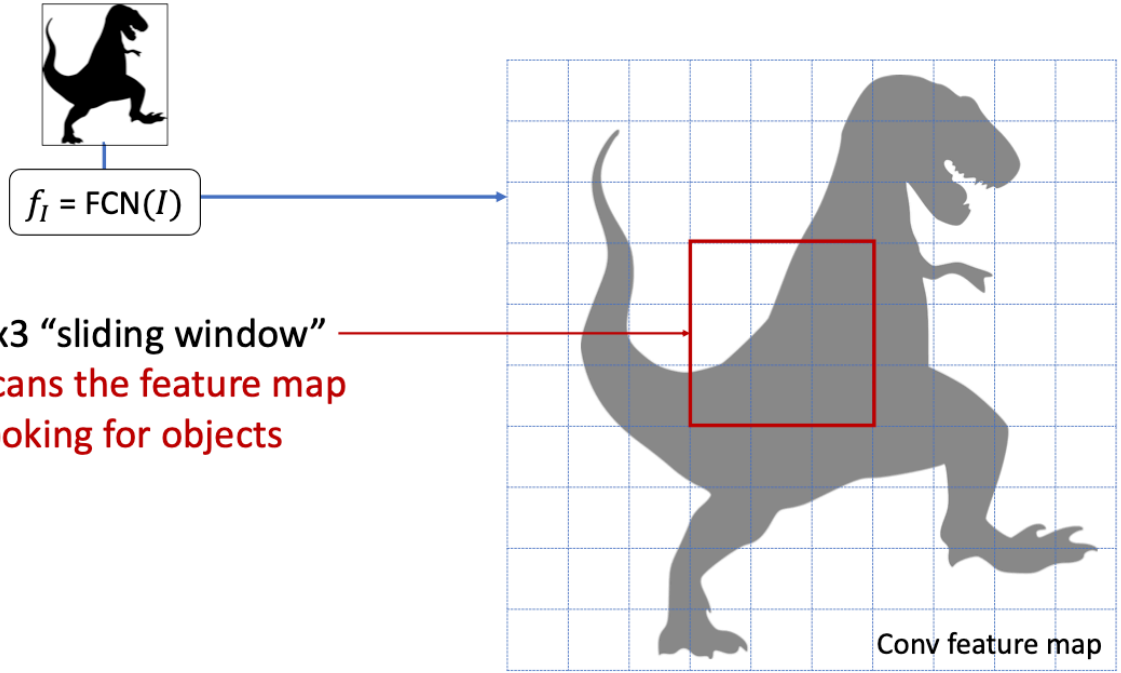


Feature Map : 16 x 16 x 256

slide credit: Ross Girshick

Region proposal network (RPN)

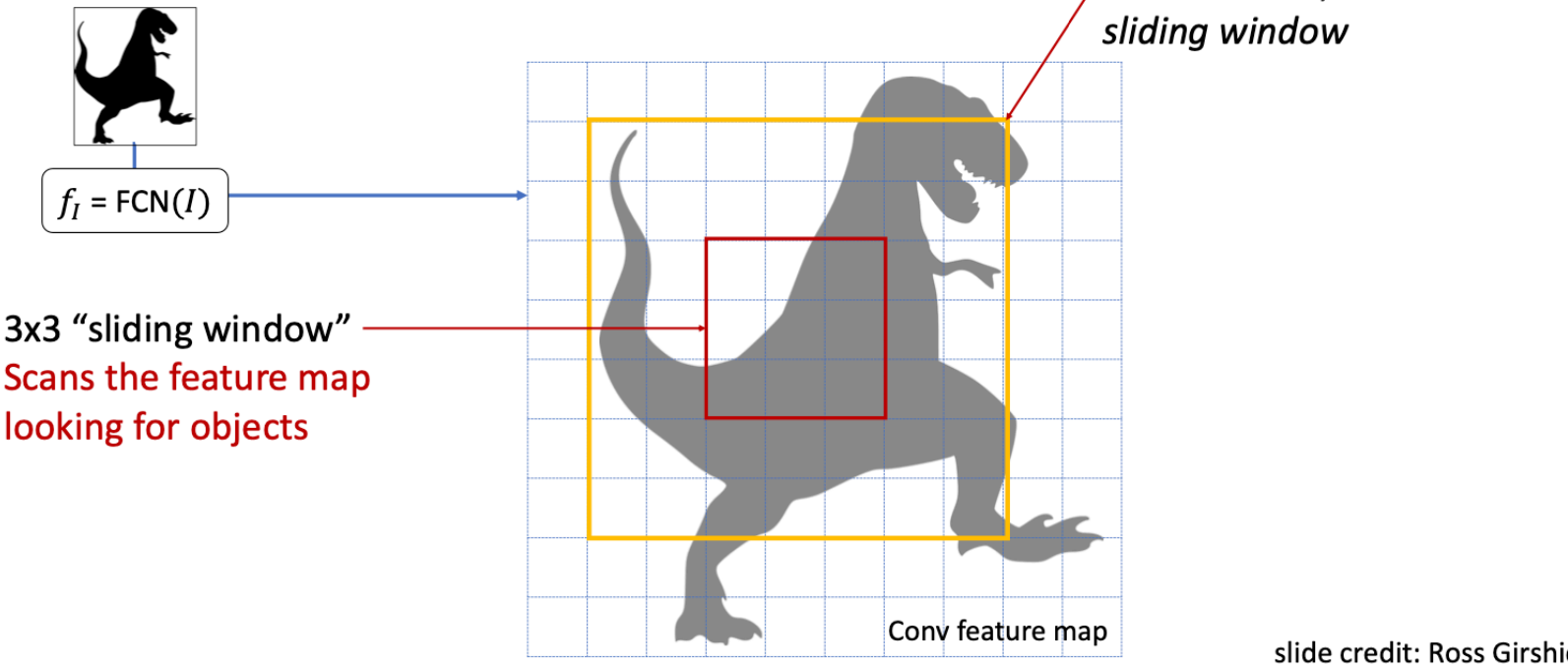
RPN: Region Proposal Network



slide credit: Ross Girshick

Region proposal network (RPN)

RPN: Anchor Box

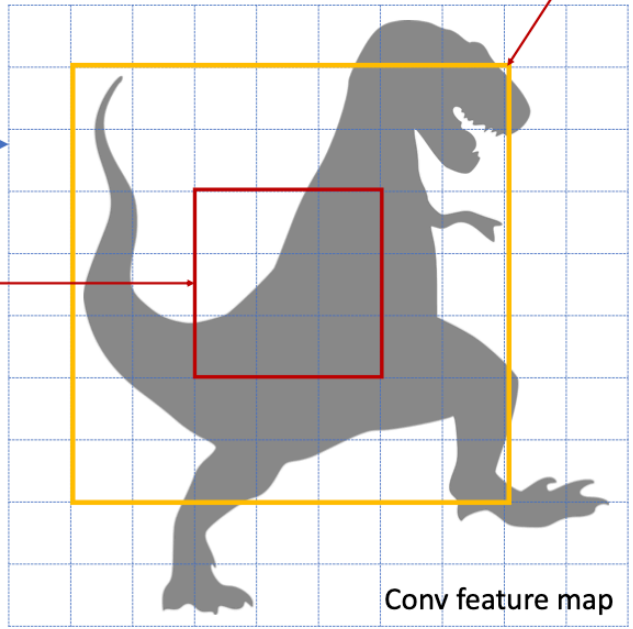


slide credit: Ross Girshick

Region proposal network (RPN)

RPN: Anchor Box

f_I = FCN(I)



Anchor box: predictions are w.r.t. this box, *not the 3x3 sliding window*

3x3 "sliding window"

- Objectness classifier [0, 1]
- Box regressor predicting (dx, dy, dh, dw)

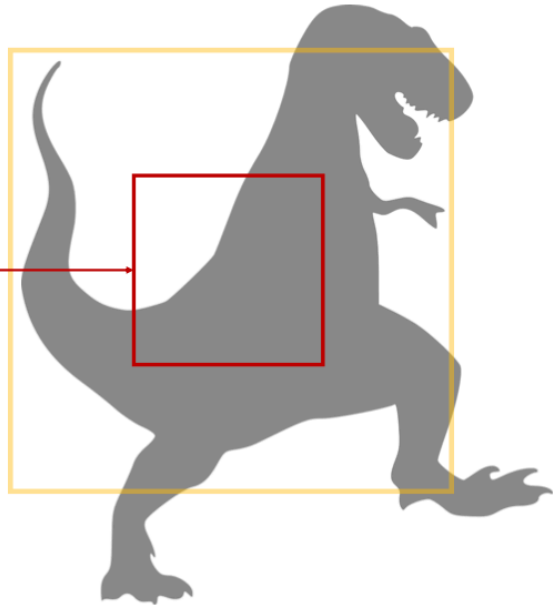
slide credit: Ross Girshick

Region proposal network (RPN)

RPN: Prediction (on object)

Objectness score

P(object) = 0.94



3x3 “sliding window”

- Objectness classifier [0, 1]
- Box regressor predicting (dx, dy, dh, dw)

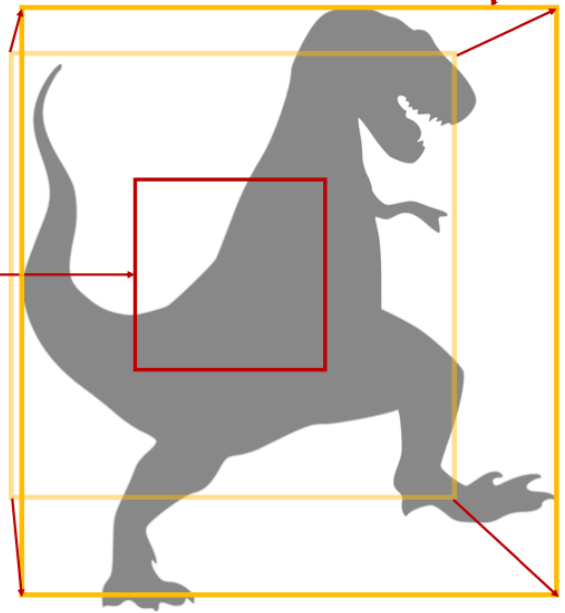
slide credit: Ross Girshick

Region proposal network (RPN)

RPN: Prediction (on object)

Anchor box: transformed by box regressor

P(object) = 0.94



3x3 “sliding window”

- Objectness classifier [0, 1]
- Box regressor predicting (dx, dy, dh, dw)

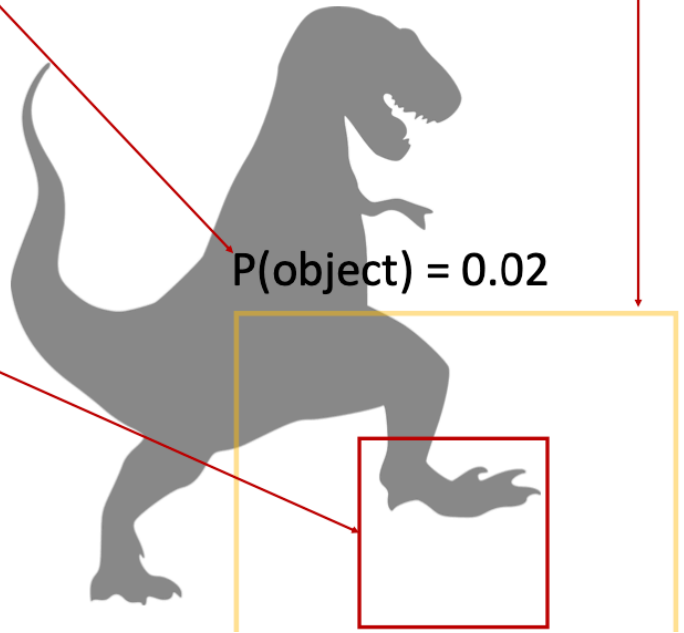
slide credit: Ross Girshick

Region proposal network (RPN)

RPN: Prediction (off object)

Objectness score

Anchor box: transformed by box regressor



3x3 “sliding window”

- Objectness classifier
- Box regressor predicting (dx, dy, dh, dw)

slide credit: Ross Girshick

Region proposal network (RPN)

RPN: Multiple Anchors

f_I = FCN(I)

Anchor boxes: *A* anchors per location with different scales and aspect ratios

Feature Map : 16 x 16 x 256

Output : 16 x 16 x A x (1+4)

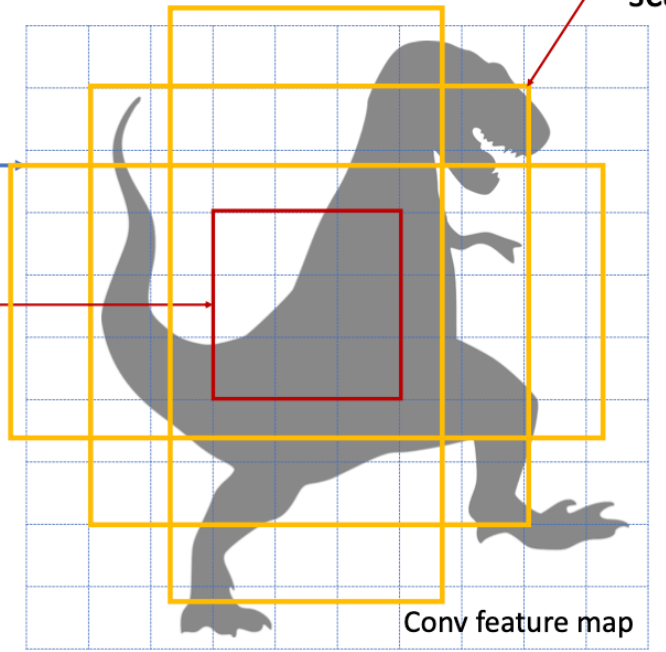
#anchors

Objectness (1)

Box regression (4)

3x3 "sliding window"

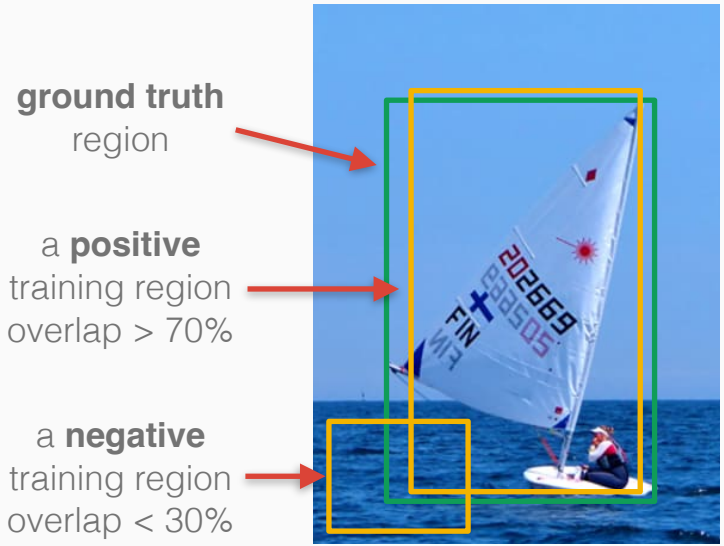
- *A* objectness classifiers
- *A* box regressors



slide credit: Ross Girshick

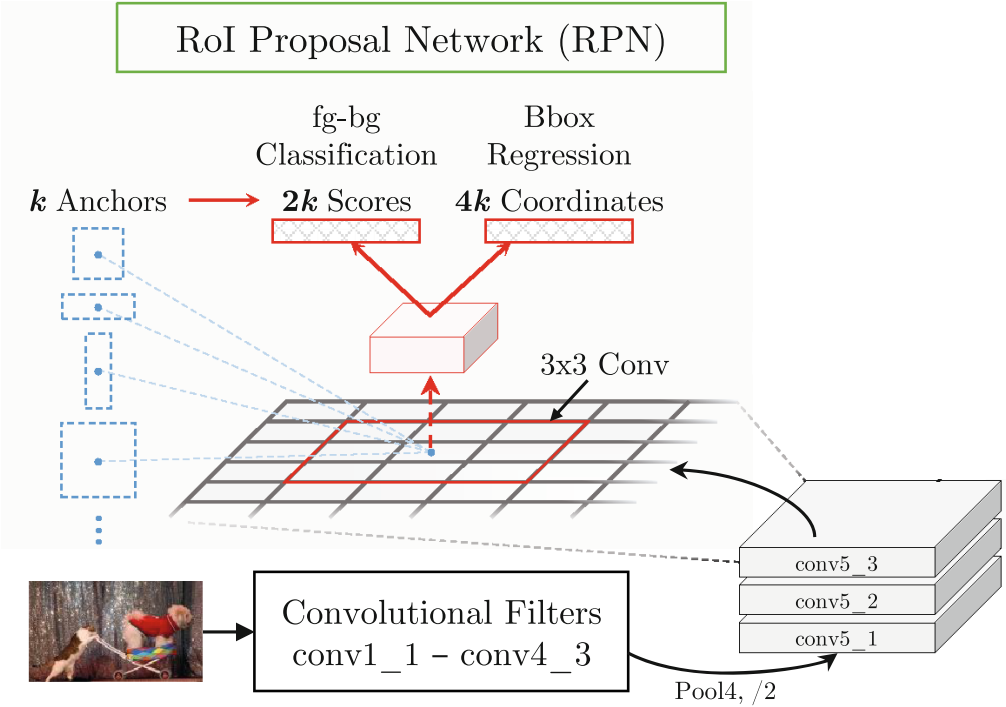
Training data: positive and negative boxes

- Label training boxes based on overlap with ground truth box
- Pre-train VGG16 CNN on ImageNet classification task



Faster R-CNN: Towards real-time object detection with region proposals, Ren et al. NIPS 2015

Faster R-CNN



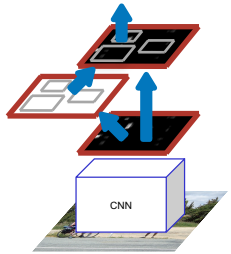


Figure from: Contextual priming and feedback for Faster R-CNN, Srivastava et al., ECCV 2016

Faster R-CNN

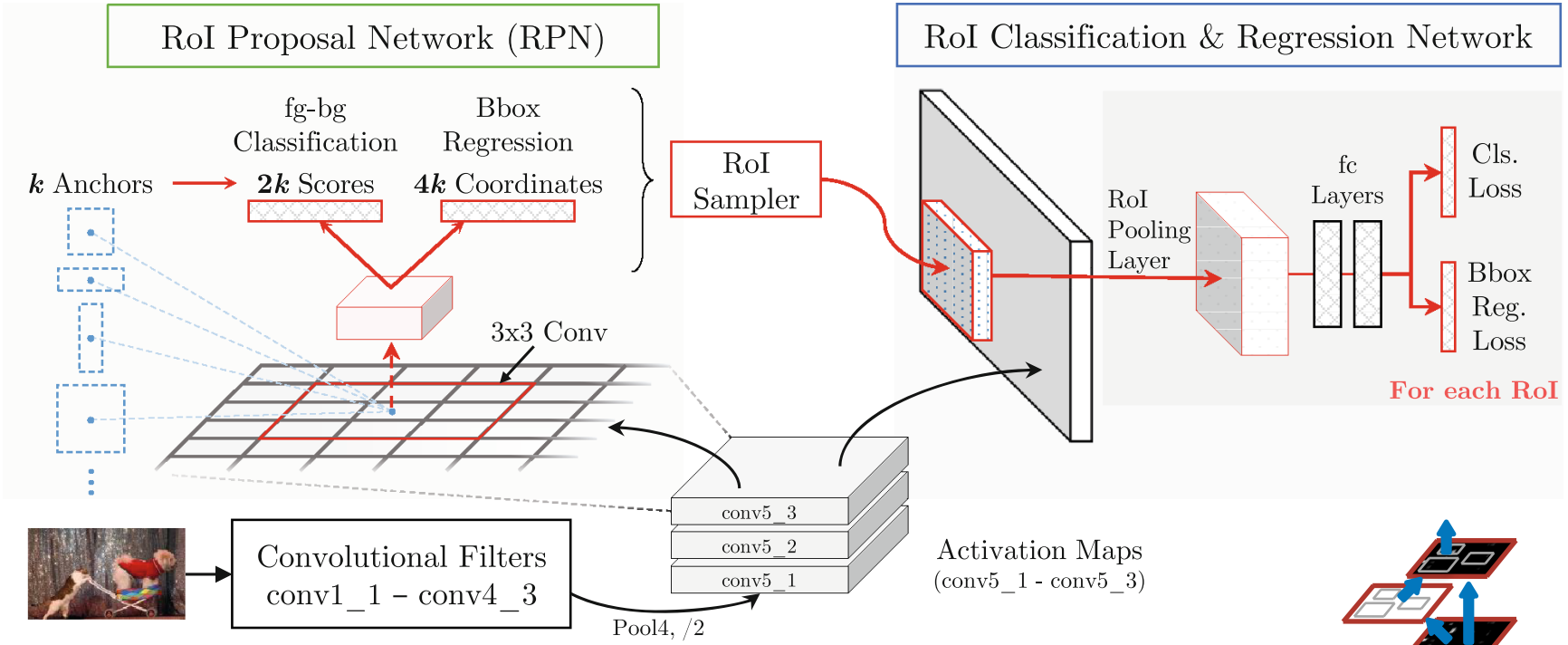


Figure from: Contextual priming and feedback for Faster R-CNN, Srivastava et al., ECCV 2016

Faster R-CNN

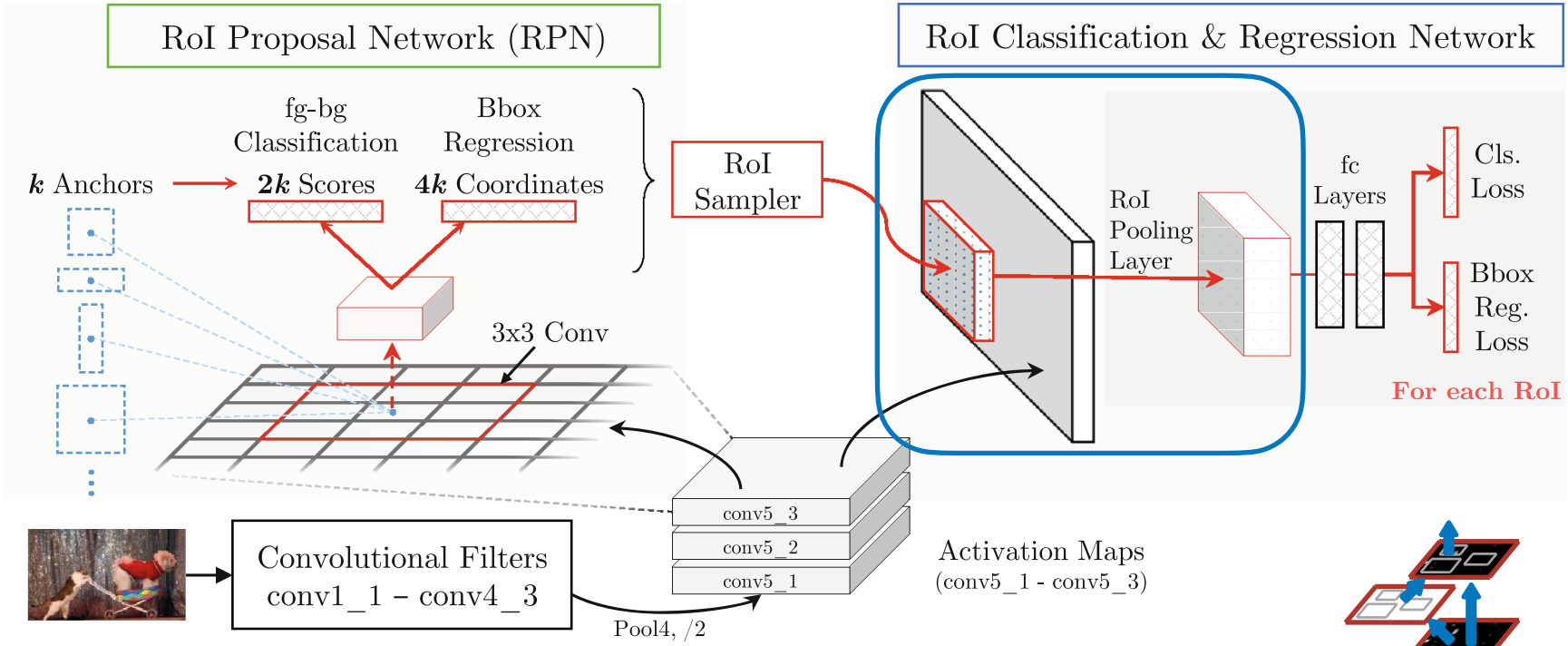
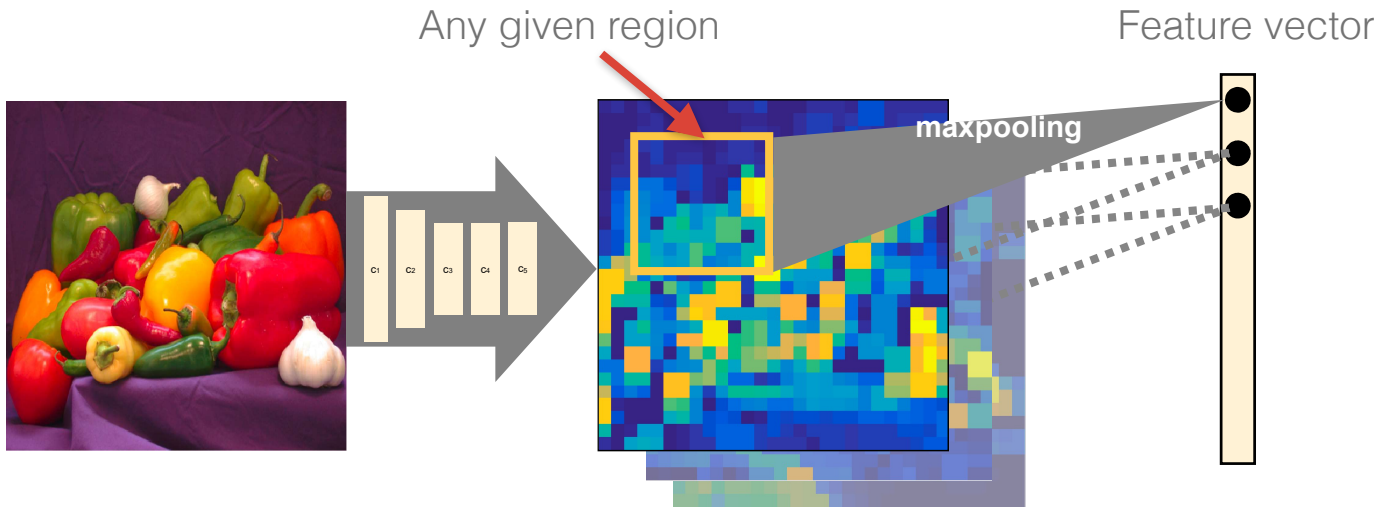


Figure from: Contextual priming and feedback for Faster R-CNN, Srivastava et al., ECCV 2016

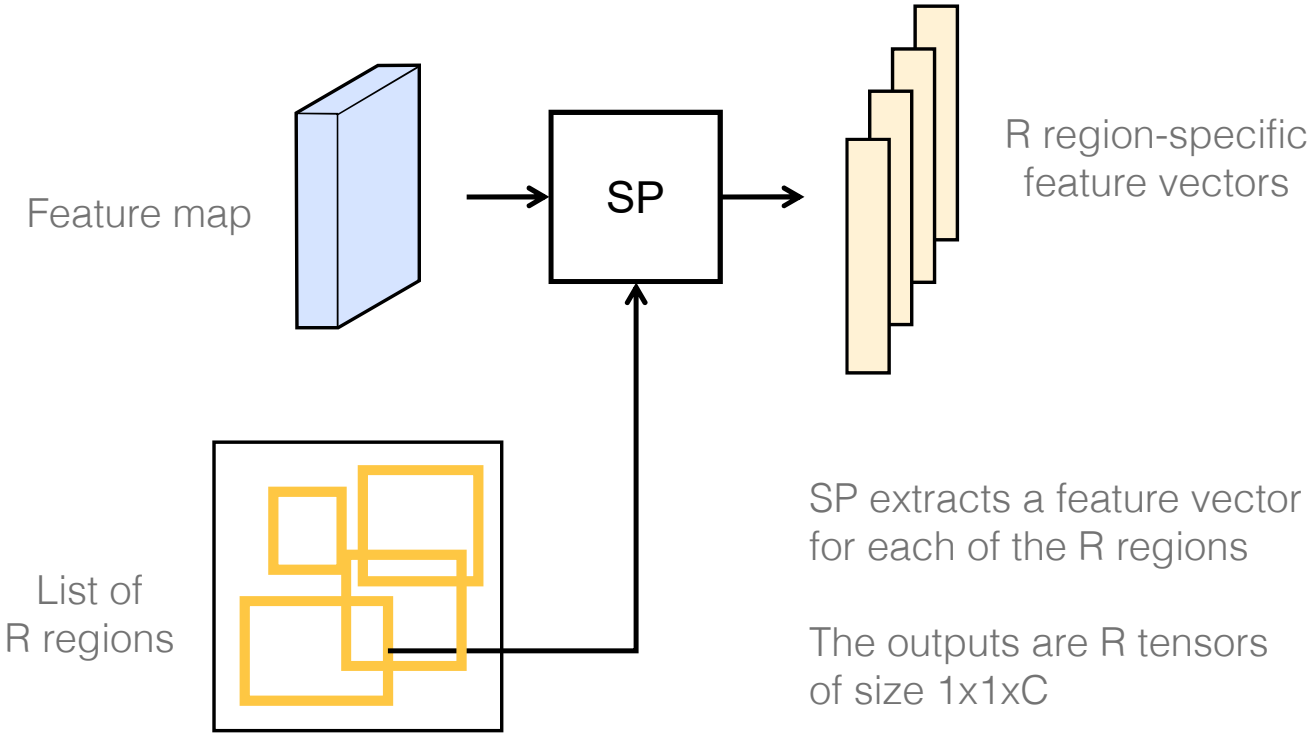
The Spatial Poolin (SP) layer

- Performs max-pooling for the feature responses in a given region
- Can be used to extract many region-specific feature vectors using same convolutional feature output



Spatial pyramid pooling (SPP) in deep convolutional networks for visual recognition, He et al. ECCV 2014

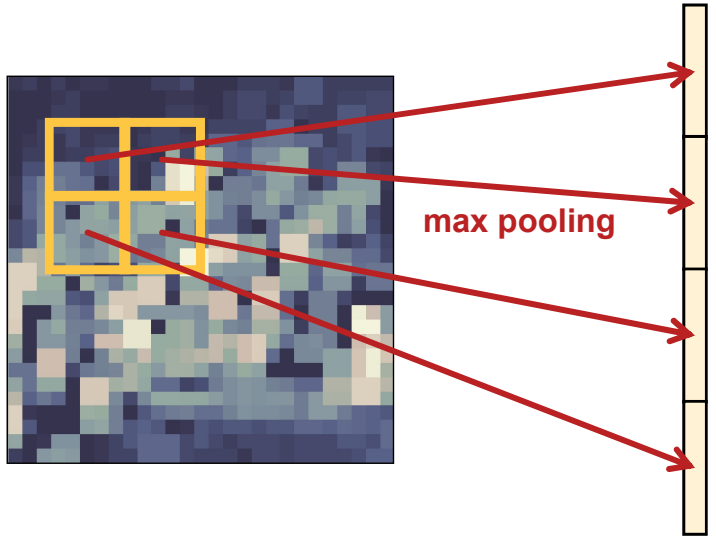
The Spatial Poolin (SP) layer as a building block



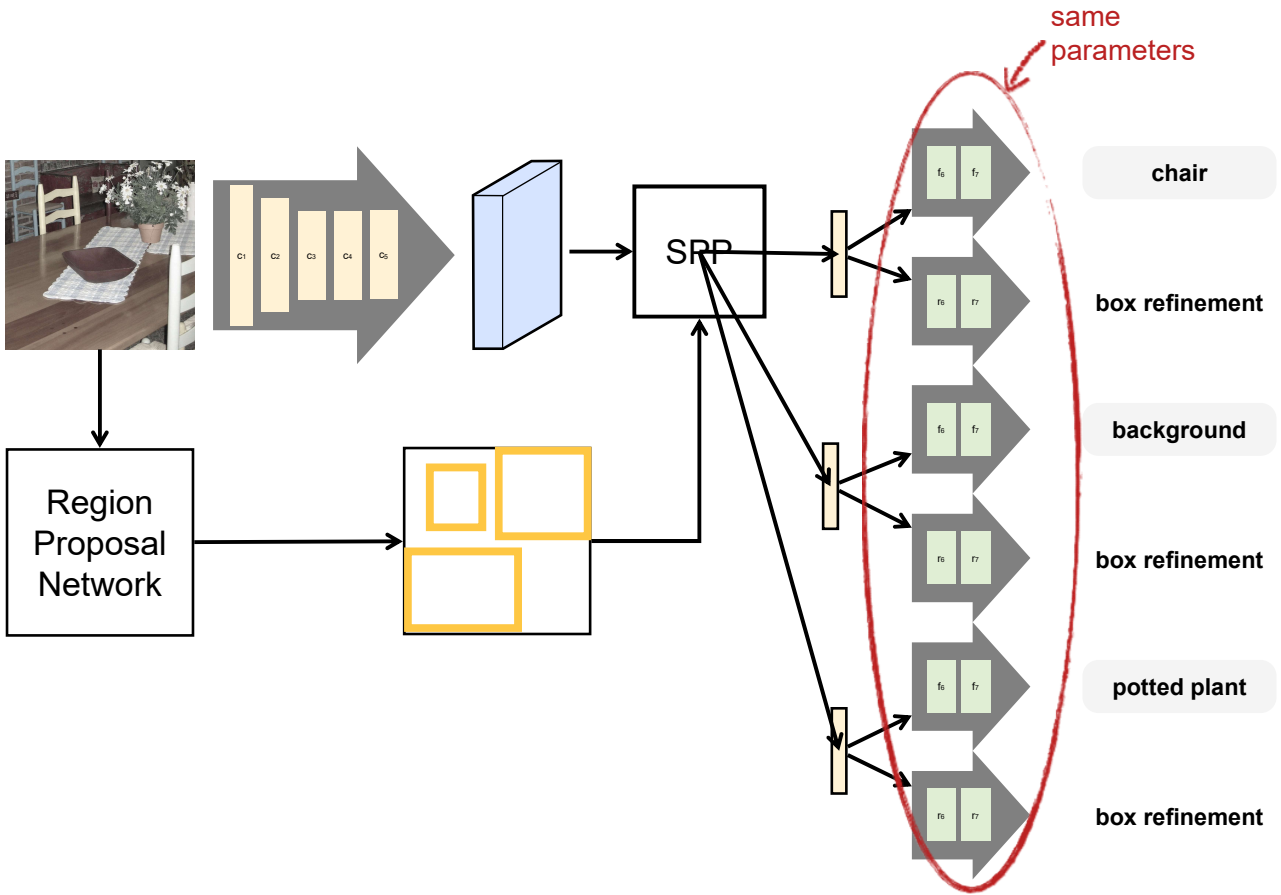
Spatial pyramid pooling (SPP) in deep convolutional networks for visual recognition, He et al. ECCV 2014

The Spatial Pyramid Pooling (SPP) layer

- Similar to SP, but pools features in tiles of a grid-like subdivision of the region (SP with multiple subdivisions)
- Feature vector **captures the spatial layout** of the original region
- Converts the region to a **fixed size vector**

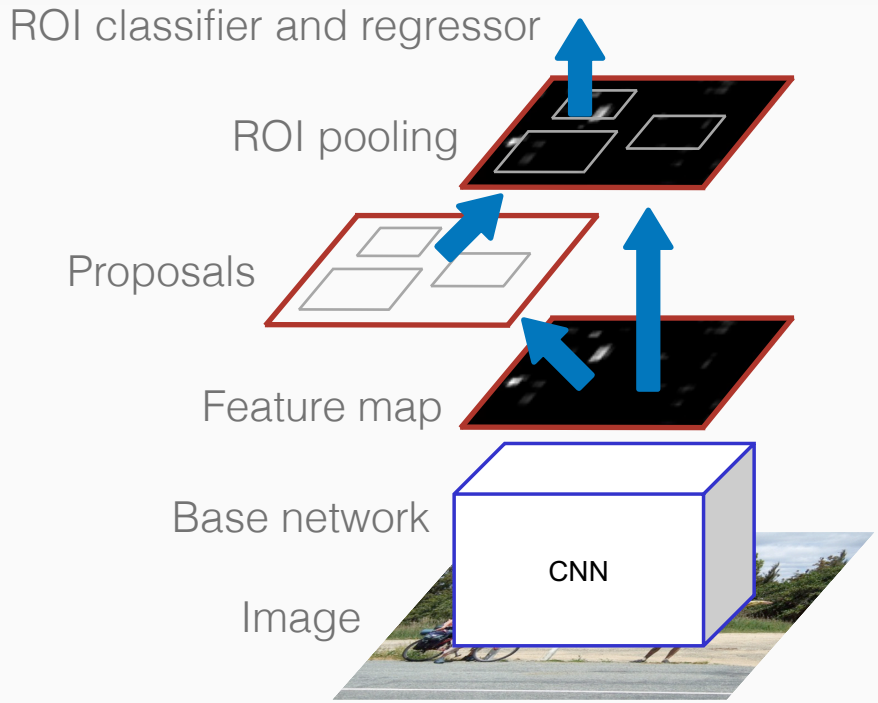


The Spatial Pyramid Pooling (SPP) layer



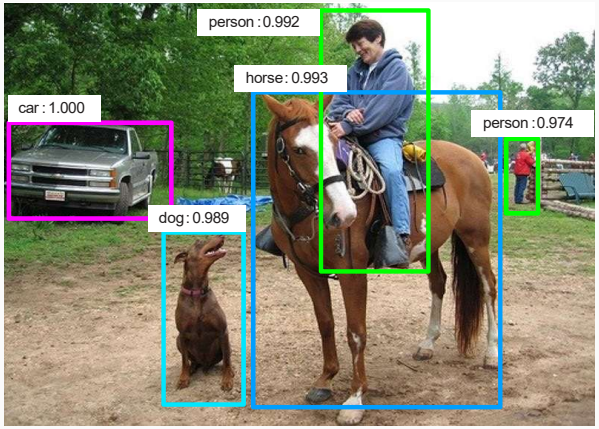
Faster R-CNN

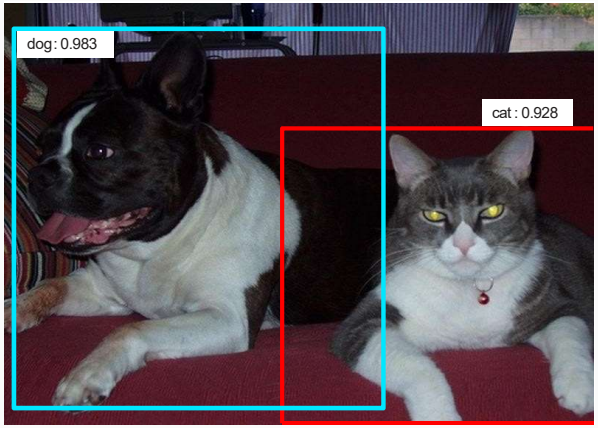
- Same CNN conv5 features used for:
  - The region proposal network
  - Classifying/regressing the regions
- Thus CNN runs only once on image
- Trained end-to-end
- Base network VGG 16

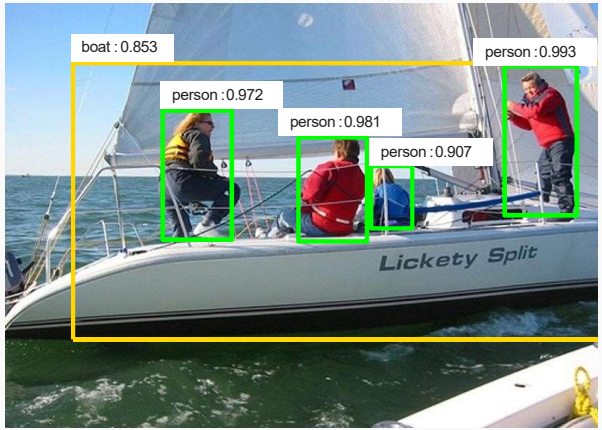


Faster R-CNN: Towards real-time object detection with region proposals, Ren et al. NIPS 2015

Example detections

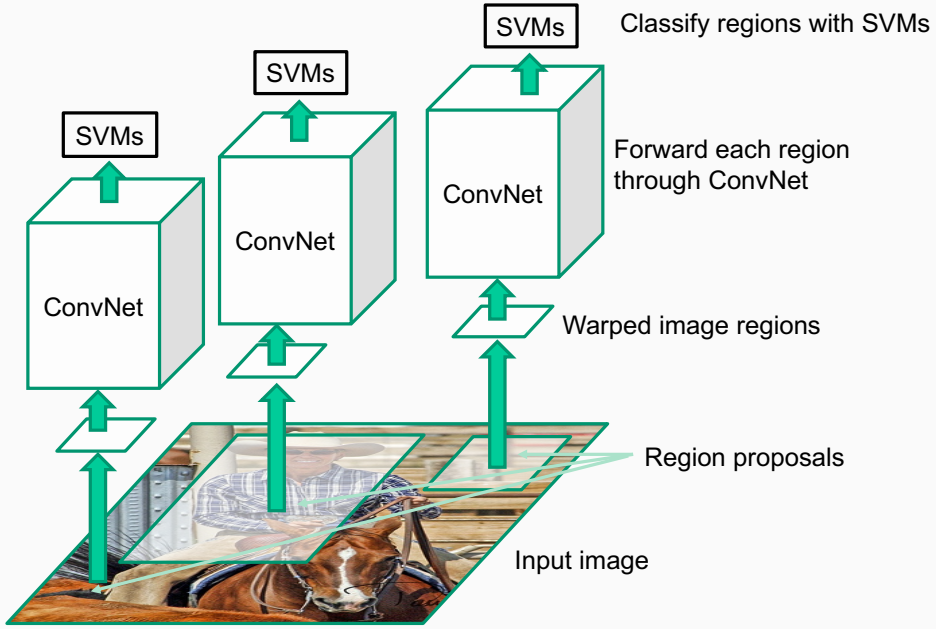






Why “Faster R-CNN”?

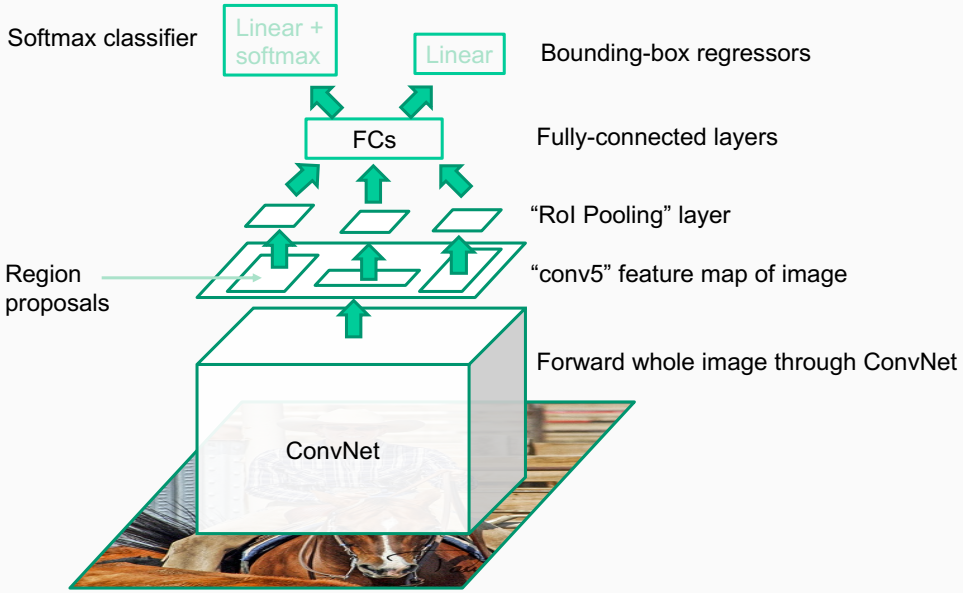
- First: R-CNN
- Inference time approx.
  50s per image



Rich feature hierarchies for accurate object detection and semantic segmentation, Girshick et al., CVPR 2014

Why “Faster R-CNN”?

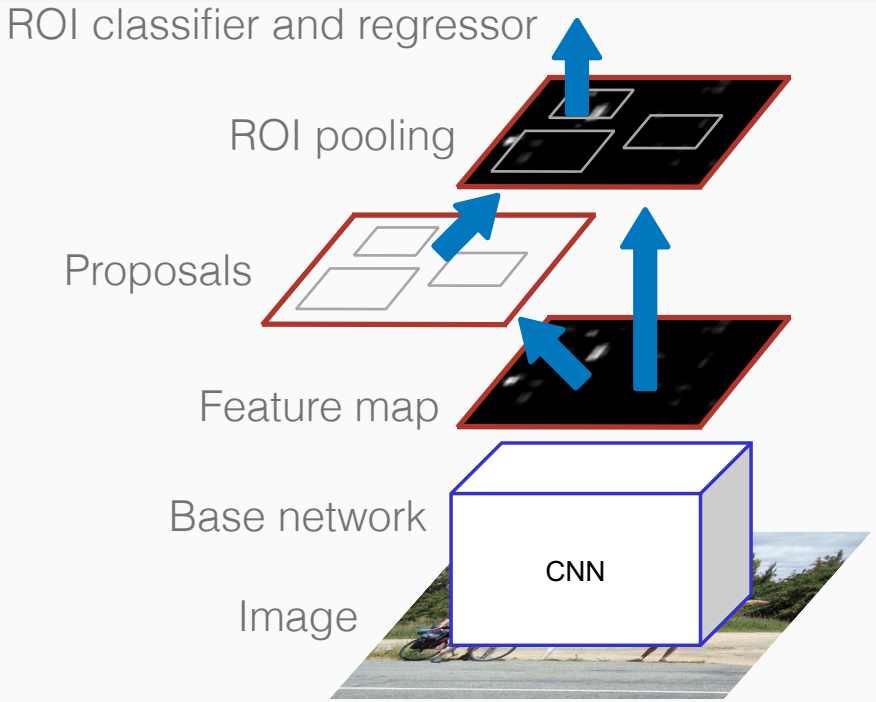
- Second: Fast R-CNN
- Inference time approx. 2s per image



Fast R-CNN, Girshick., ICCV 2015

Why “Faster R-CNN”?

- Third: Faster R-CNN
- Inference time approx. 198ms per image



Faster R-CNN: Towards real-time object detection with region proposals, Ren et al. NIPS 2015

Evaluating object detectors

MS Common Objects in Context (COCO)

What is COCO?

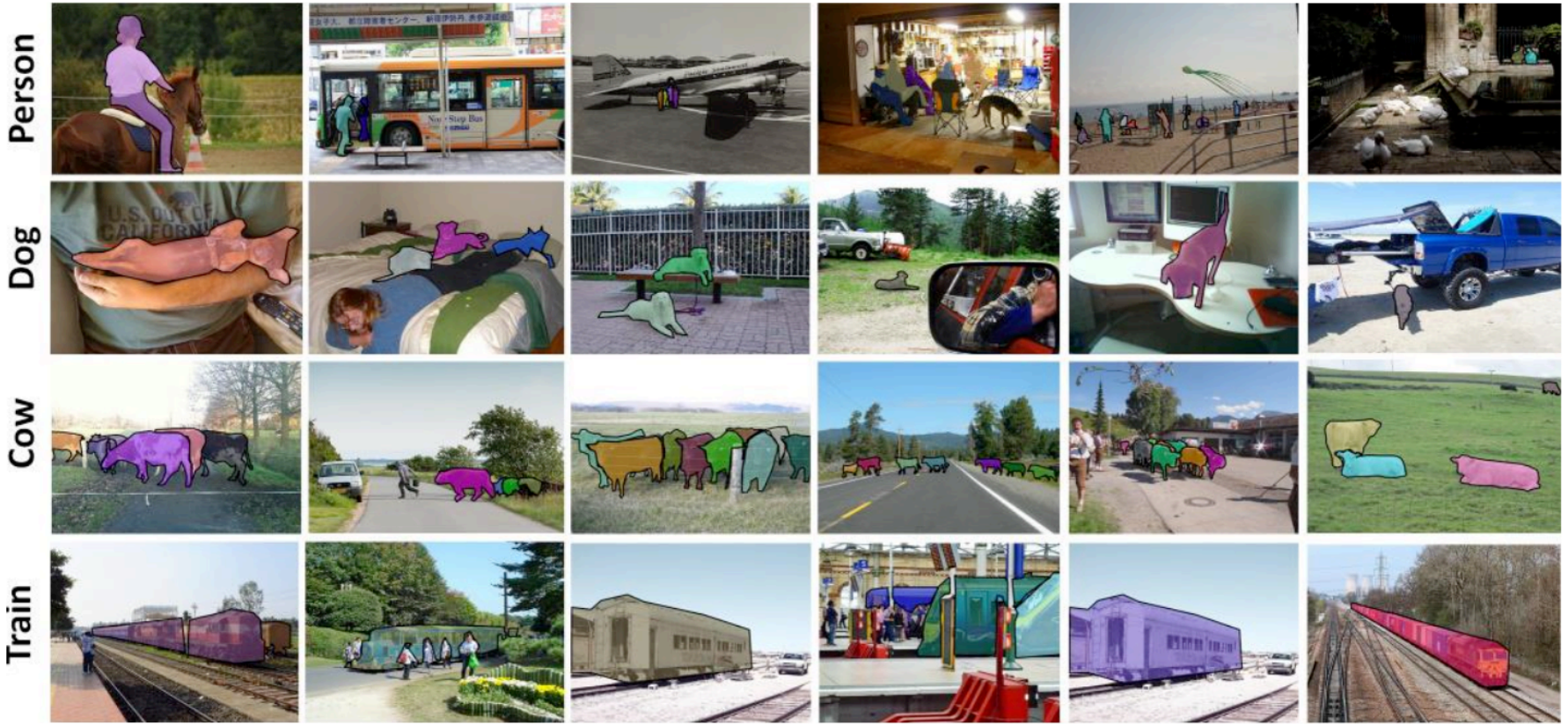
COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- Object segmentation
- Recognition in context
- Superpixel stuff segmentation
- 330K images (>200K labeled)
- 1.5 million object instances
- 80 object categories
- 91 stuff categories
- 5 captions per image
- 250,000 people with keypoints

COCO
Common Objects in Context

http://mscoco.org/home/

MS COCO: Example images



Bounding boxes can created around each segmentation instance.

Figure: A. Zisserman

MS COCO: Example images

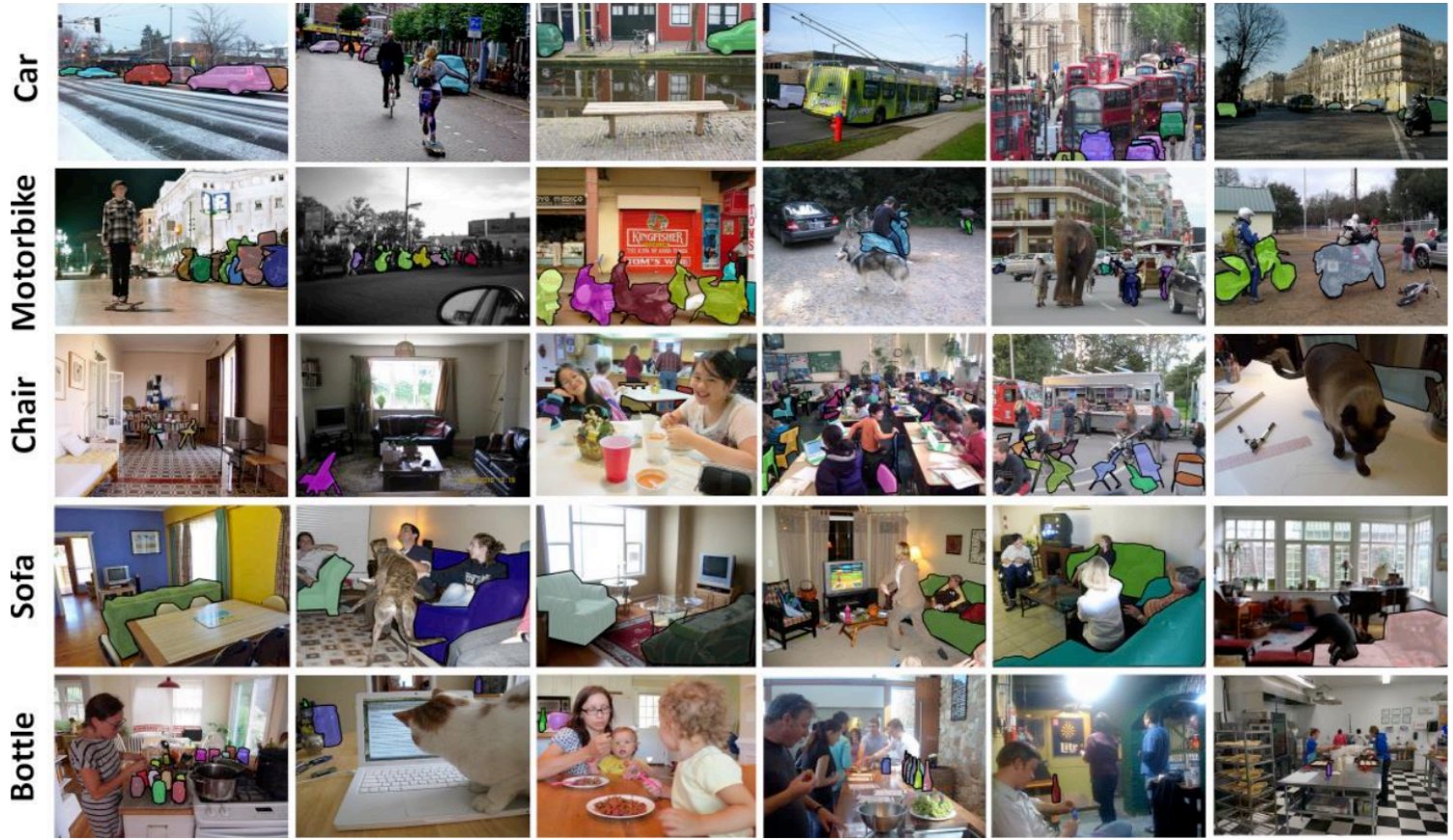
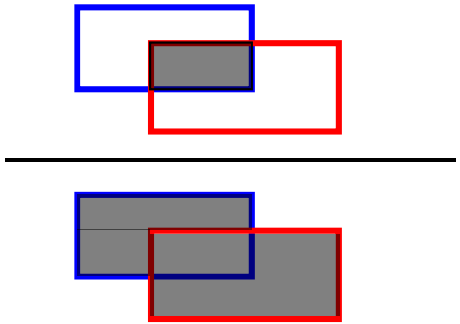


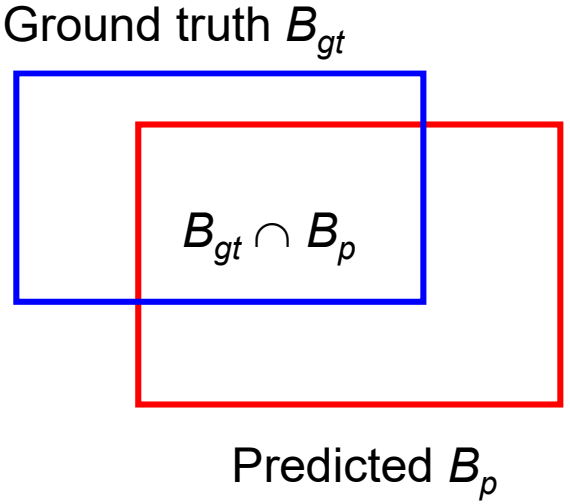
Figure: A. Zisserman

Recall: Intersection over union measure

- Area of overlap (AO) measure
- Correct detection if intersection over union larger than threshold



> threshold (50 %)



$$AO(B_{gt}, B_p) = \frac{|B_{gt} \bigcap B_p|}{|B_{gt} \bigcup B_p|}$$

COCO detection evaluation metrics

```
Average Precision (AP):
  AP                  % AP at IoU=.50:.05:.95 (determines challenge winner)
  AP^IoU=.50          % AP at IoU=.50 (PASCAL VOC metric)
  AP^IoU=.75          % AP at IoU=.75 (strict metric)
```

- Challenges Score:

  AP is averaged over multiple IoU values between 0.5 and 0.95

  More comprehensive metric than the traditional AP at a fixed IoU value (e.g. 0.5 in PASCAL).

Credit: A. Zisserman

COCO detection evaluation metrics

AP Across Scales:

AP^{small} % AP for small objects: area < 32^2

AP^{medium} % AP for medium objects: 32^2 < area < 96^2

AP^{large} % AP for large objects: area > 96^2

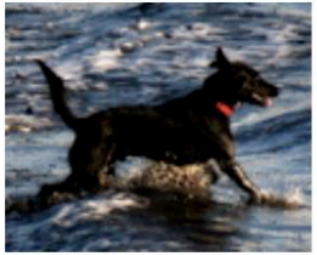
- Other scores: Size AP

  AP is averaged over instance size:

  - Small (A < 32 x 32)
  - Medium (32 x 32 < A < 96x96)
  - Large (A > 96x96)

A < 32x32

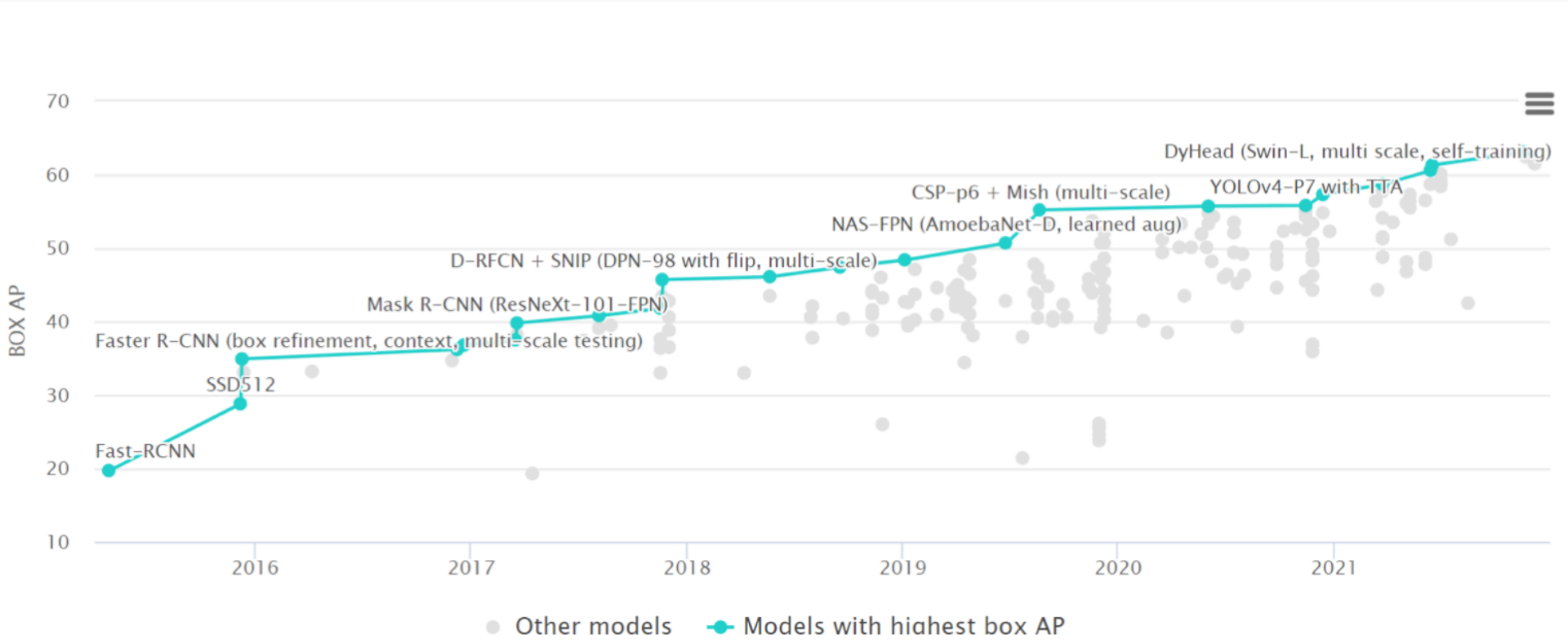
32x32 < A < 96x96



A > 96x96

Credit: A. Zisserman

Object Detection on COCO test-dev (Jan 2022)



https://paperswithcode.com/sota/object-detection-on-coco

Object Detection on COCO test-dev

Methods	Architecture	Year	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Inference Speed (fps)
Two-Stage Detectors:									
R-CNN (only test on Pascal VOC) [1]	VGG-16	2014	–	–	–	–	–	–	0.02
SPPNet (only test on Pascal VOC) [2]	VGG-16	2015	–	–	–	–	–	–	–
Fast R-CNN [3]	VGG-16	2015	19.7	35.9	–	–	–	–	0.5
Faster R-CNN [4]	VGG-16	2015	21.9	42.7	–	–	–	–	5
Fast RCNN + OHEM [5]	VGG-16	2016	22.6	42.5	22.2	5.0	23.7	37.9	5
Faster-RCNN+++ [6]	ResNet-101	2016	34.9	55.7	37.4	15.6	38.7	50.9	2.4
R-FCN [7]	ResNet-101	2016	29.9	51.9	–	10.8	32.8	45.0	6
Faster-RCNN w Feature Pyramid Net [8]	ResNet-101	2016	36.2	59.1	–	18.2	39.0	48.2	7
Deformable R-FCN [9]	ResNet-101	2017	34.5	55.0	–	14.0	37.7	50.3	5
Mask-RCNN [10]	ResNeXt-101	2017	39.8	62.3	43.4	22.1	43.2	51.2	5
Mask-RCNN-GroupNorm [11]	ResNet-101	2018	42.3	62.8	46.2	–	–	–	5
SNIPER [12]	ResNet-101	2018	46.1	67.0	51.6	29.6	48.9	58.1	–
Faster-RCNN with DCN-v2 [13]	ResNet-101	2019	44.0	66.3	48.8	24.4	48.1	59.6	–
Faster-RCNN-TridentNet [14]	ResNet-101	2019	42.7	63.6	46.5	23.9	46.6	56.6	2.4
Faster-RCNN-TridentNet-Deformable [14]	ResNet-101	2019	48.4	69.7	53.5	31.8	51.3	60.3	0.7
DetectoRS [15]	ResNeXt-101	2020	55.7	74.2	61.1	37.7	58.4	68.1	–
Swin-L (HTC++) [16]	Swin Transformer	2021	**58.7**	–	–	–	–	–	–
SwinV2-G (HTC++) [17]	Swin Transformer V2	2021	**62.5**	–	–	–	–	–	–
One-Stage Detectors:									
SSD [18]	VGG-16	2016	28.8	48.5	30.3	10.9	31.8	43.5	19
YOLOv2 [19]	DarkNet-19	2017	21.6	44.0	19.2	5.0	22.4	35.5	40
YOLOv3 [20]	DarkNet-53	2017	33.0	57.9	34.4	18.3	35.4	41.9	20
RetinaNet [21]	ResNet-101	2017	39.1	59.1	42.3	21.8	42.7	50.2	5.4
CornerNet [22]	Hourglass-104	2018	42.1	57.8	45.3	20.8	44.8	56.7	4.1
ExtremeNet [23]	Hourglass-104	2019	42.1	61.1	45.9	24.1	45.5	52.8	3.1
CenterNet-1 [24]	Hourglass-104	2019	45.1	63.9	49.3	26.6	47.1	57.7	7.8
EfficientDet-D2(768, single-scale) [25]	EfficientNet	2020	43.0	62.3	46.2	–	–	–	41.6
EfficientDet-D4(1024, single-scale) [25]	EfficientNet	2020	49.4	69.0	53.4	–	–	–	13.5
EfficientDet-D7(1536, single-scale) [25]	EfficientNet	2020	52.2	71.4	56.3	–	–	–	3.8
DETR [26]	R50+Transformer	2020	42.0	62.4	44.2	20.5	45.8	61.1	28

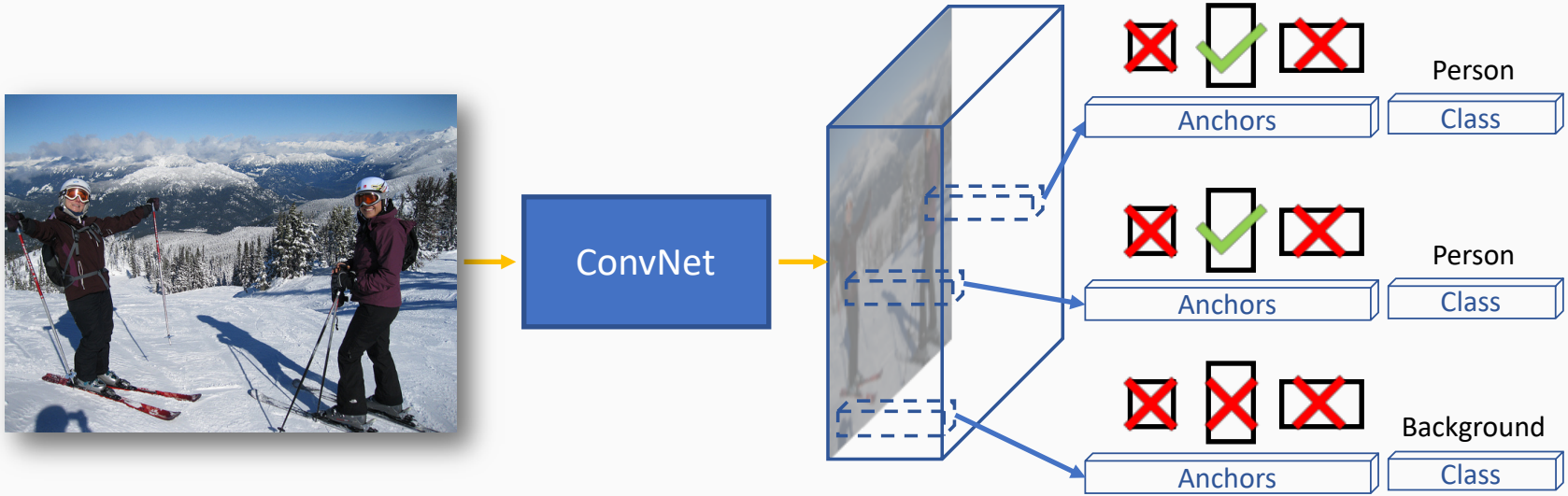
Credit: Weidi Xi

Single stage detectors

Two strands of detection architectures

- Detectors using region proposal networks (RPN)
  - Two stages: 1) RPN, followed by 2) features from regions for classification and regression of box
  - Possibly slow due to two steps
  - Examples: Faster RCNN, R-FCN
- Detector using unified framework (no explicit RPN)
  - Regions are build into the architecture (convolutional layers) -> possibly fast
  - Examples: Anchor based, e.g. YOLO, SSD, RetinaNet, EfficientDet
    Point based, e.g. CornerNet, CenterNet, FCOS

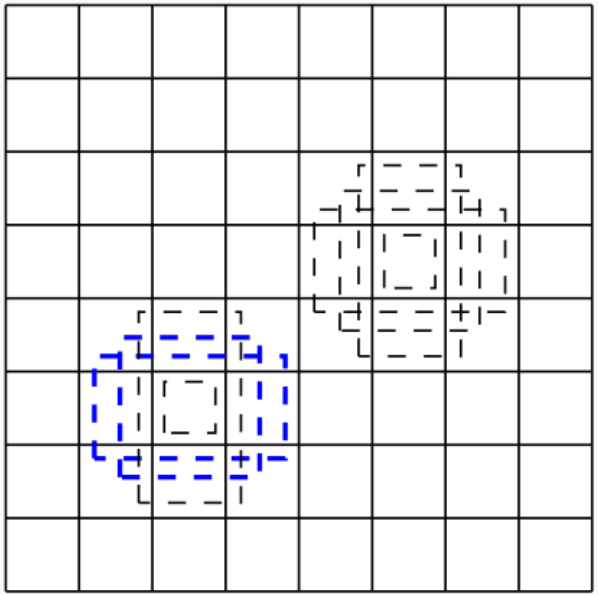
One-stage detectors



Redmond et al. CVPR 2017, Shen et al. ICCV 2017, Liu et al. ECCV 2016,
Fu et al. arXiv 2017, Lin et al. ICCV 2017, Zhang et al. CVPR 2018

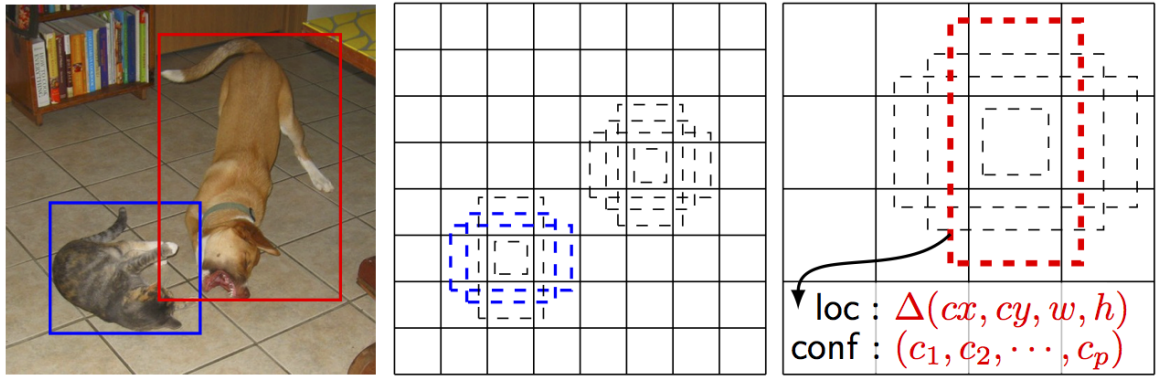
Single Shot MultiBox Detector (SSD)

- Fully convolutional detector (no RPN)
- Pre-defines regions:
  - Predict categories and box offsets
  - Multiple aspect ratios per cell
  - Similar to Faster R-CNN anchor boxes

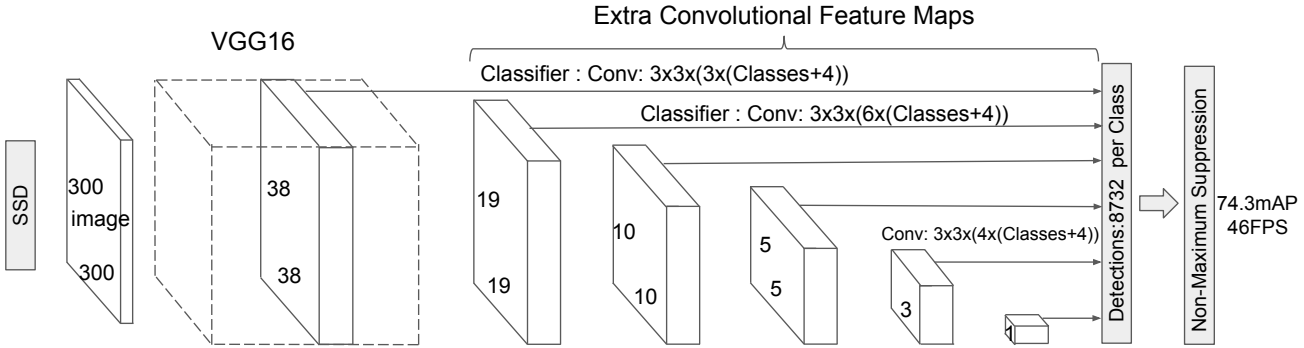


SSD: Single Shot MultiBox Detector, Liu et al., ECCV 2016

Single Shot MultiBox Detector (SSD)

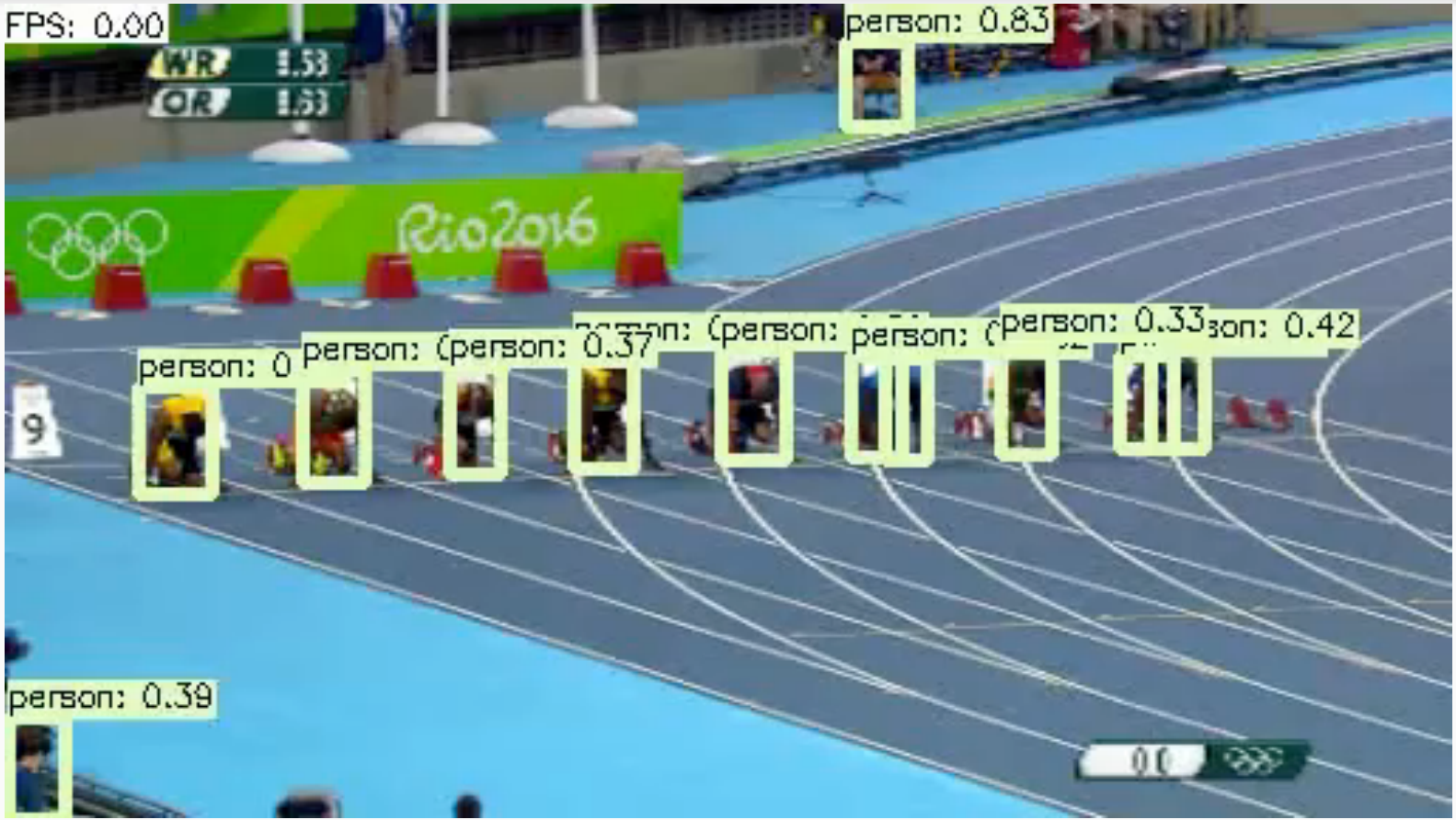


SSD runs at 59 fps
cf. 7 fps for Faster R-CNN



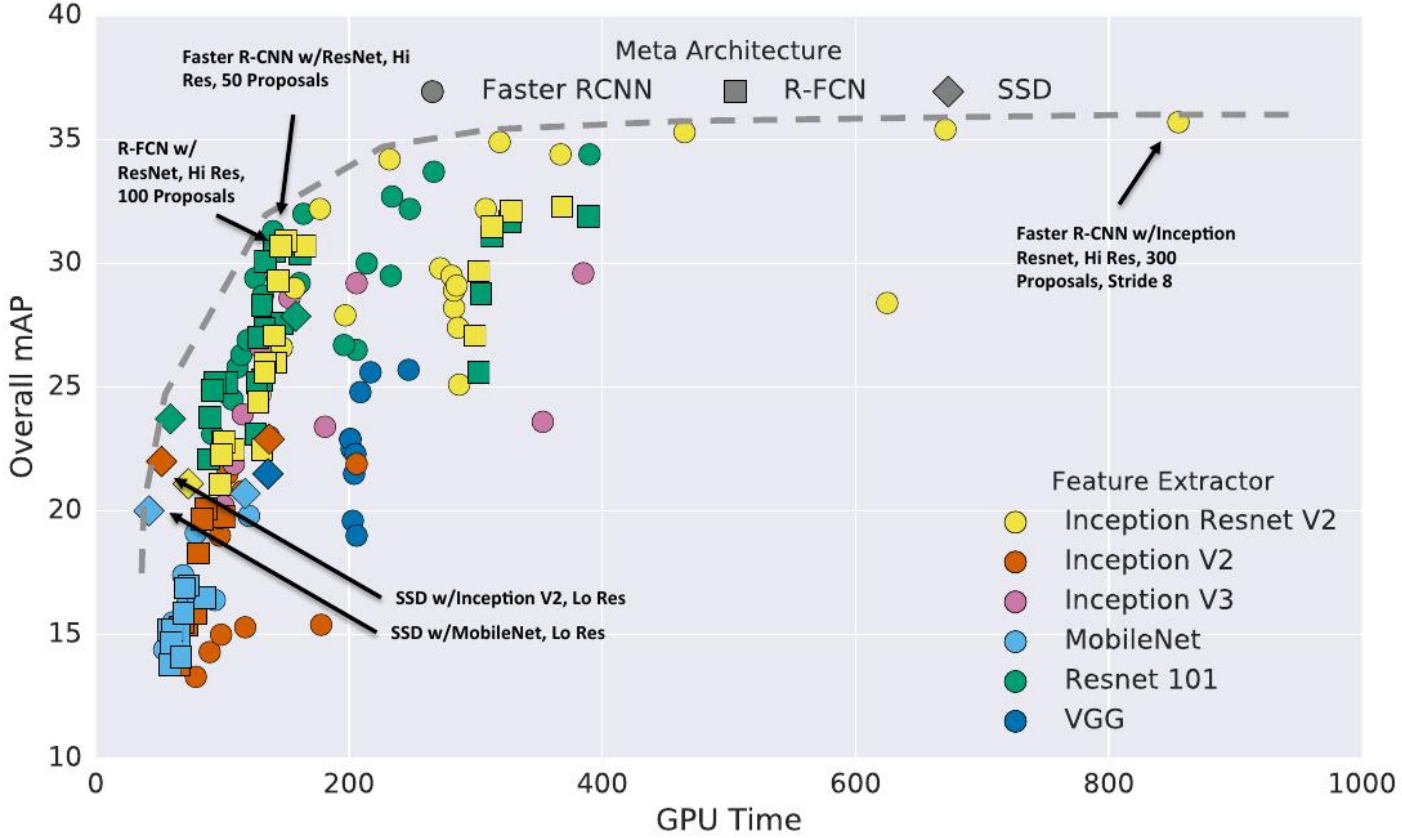
SSD: Single Shot MultiBox Detector, Liu et al., ECCV 2016

Single Shot MultiBox Detector - video example



SSD: Single Shot MultiBox Detector, Liu et al., ECCV 2016

Accuracy vs speed (COCO)



Speed/accuracy trade-offs for modern convolutional object detectors, Huang et al. CVPR 2017

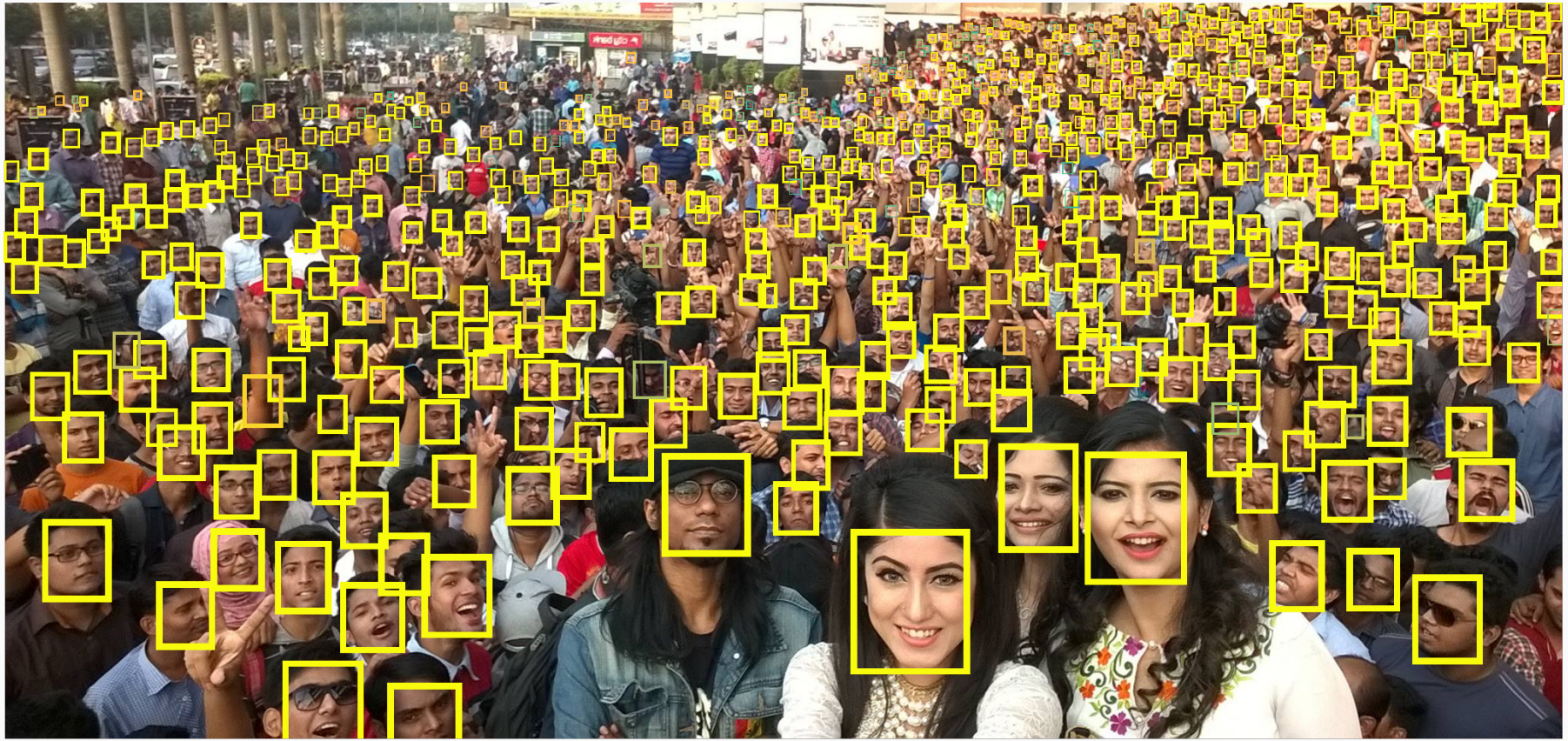
Significant improvements

Recent key developments

- Feature Pyramid Networks (FPNs)
- Anchorless networks
- Optimised architecture combinations
- Copy & Paste augmentation training
- Transformer architectures

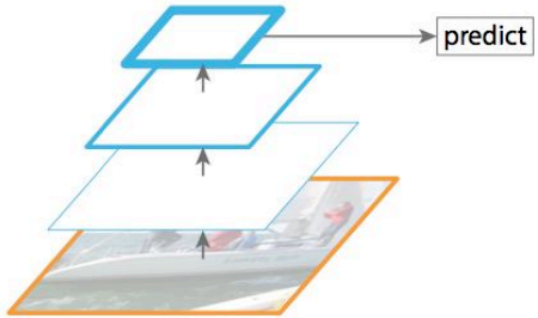
Many of these ideas can be used to improve other tasks such as segmentation and tracking

Scale in object detection



Finding tiny faces, Hu et al. CVPR 2017

Scale in object detection



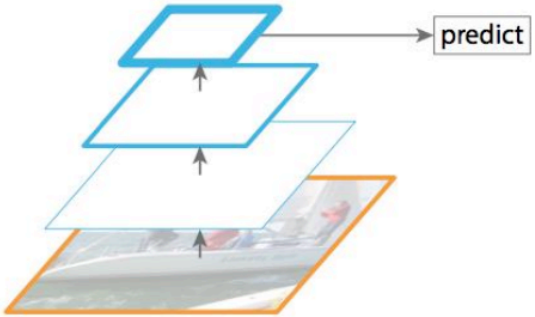
Single feature map

Only compute regions at highest feature map as in Faster R-CNN

Problems of resolution for small objects

Credit: A Zisserman

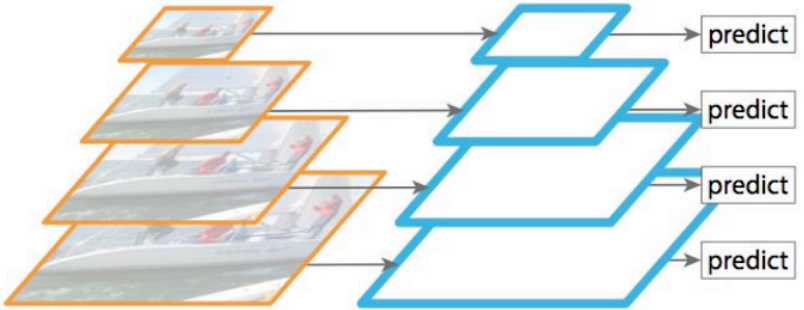
Scale in object detection



Single feature map

Only compute regions at highest feature map as in Faster R-CNN

Problems of resolution for small objects



Featurized image pyramid

Network for each image scale as in HOG

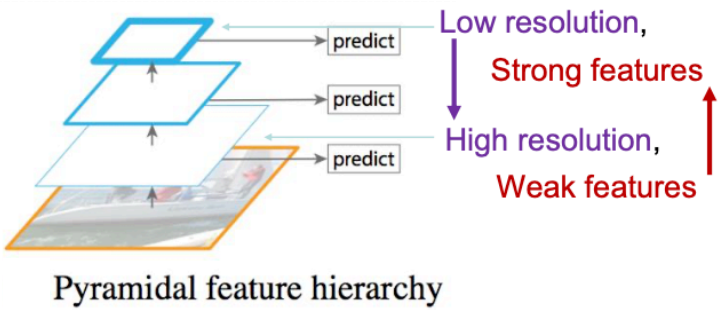
Reuse classifier network at each scale

Too slow

Credit: A Zisserman

Scale in object detection

Feature Pyramid Networks (FPNs)



Pyramidal feature hierarchy

Use the internal feature pyramids

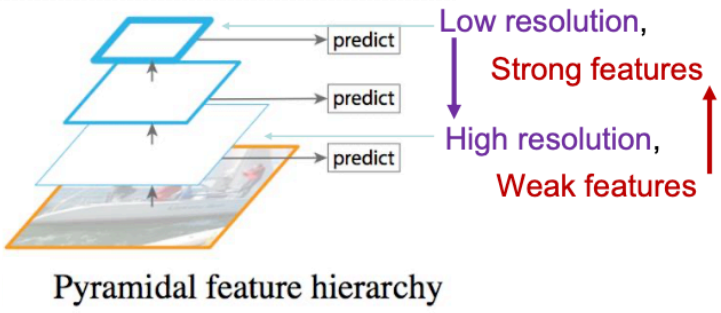
Weak features for higher resolution

Credit: A Zisserman

Feature Pyramid Networks for Object Detection, T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, CVPR, 2017

Scale in object detection

Feature Pyramid Networks (FPNs)



Pyramidal feature hierarchy

Feature Pyramid Network

Use the internal feature pyramids
Weak features for higher resolution

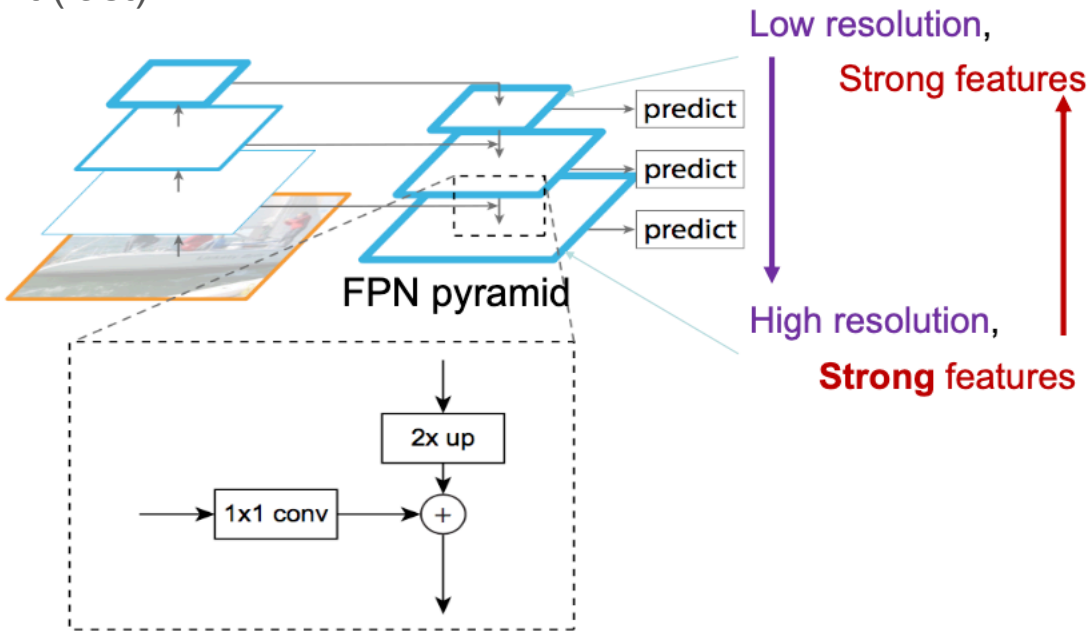
Top down enrichment for higher resolution features

Credit: A Zisserman

Feature Pyramid Networks for Object Detection, T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, CVPR, 2017

Scale in object detection

Feature Pyramid Networks (FPNs)
Light weight top-down refinement (fast)

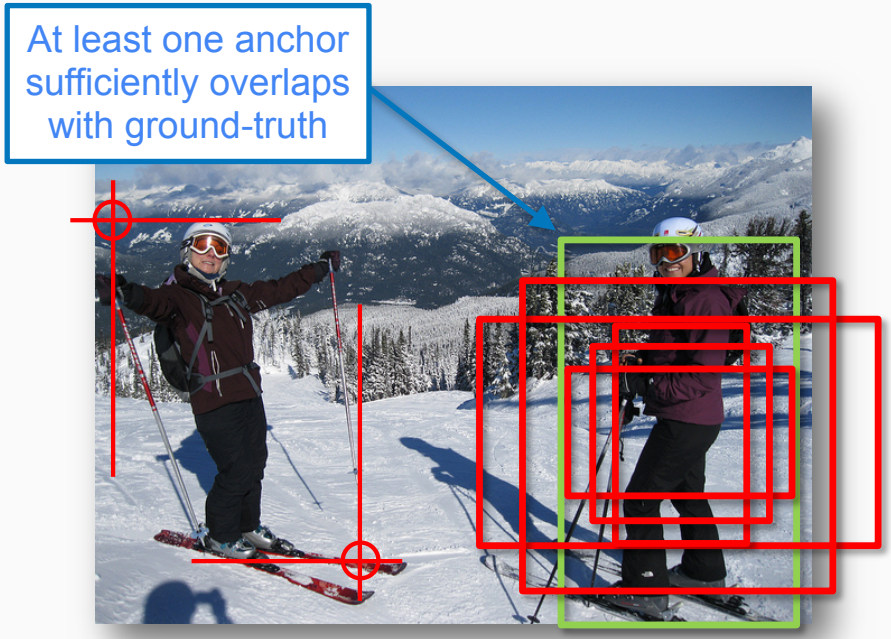


Credit: A Zisserman

Feature Pyramid Networks for Object Detection, T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, CVPR, 2017

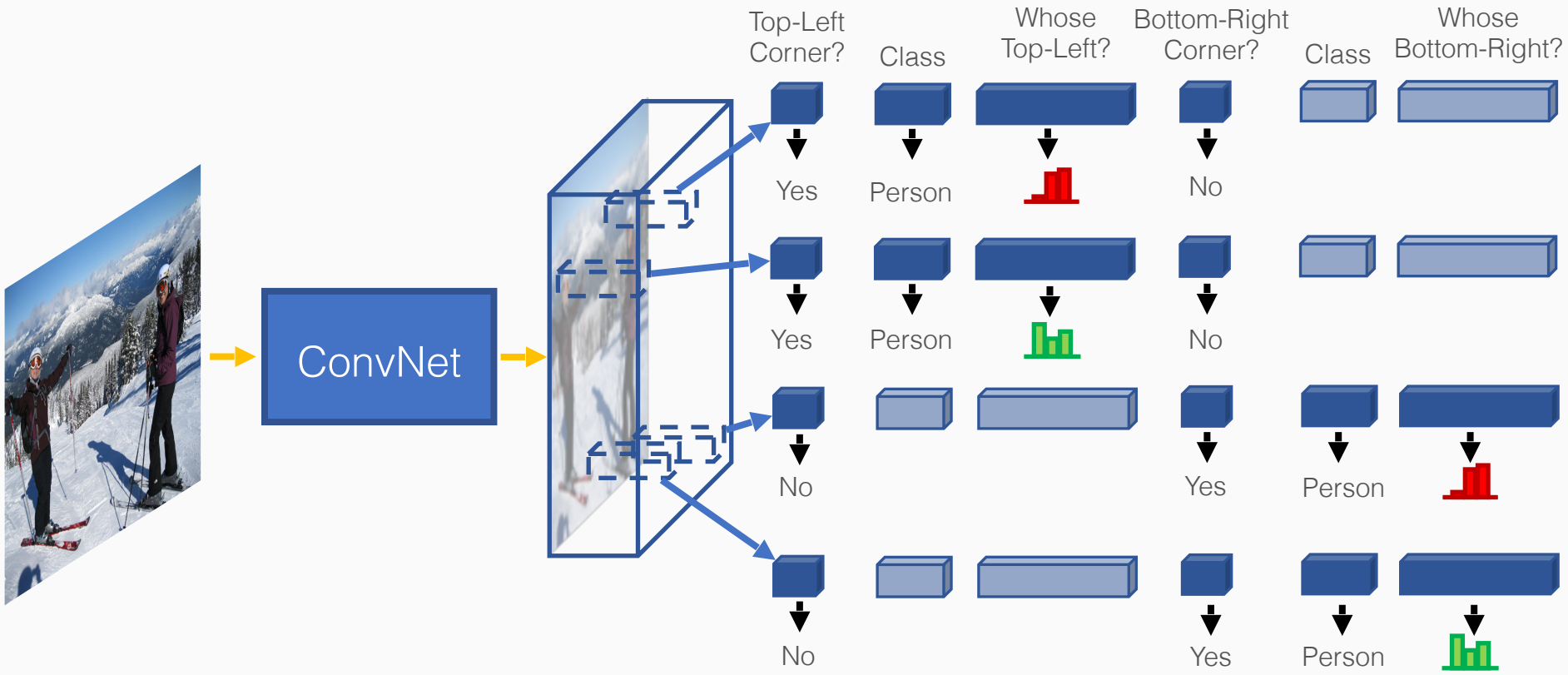
CornerNet

- Drawbacks of anchor boxes
  - Need a large number of anchors
    -> Tiny fraction are positive examples
    -> Slower training [Lin et al. ICCV 2017]
  - Extra hyperparameters -
    sizes, aspect ratios



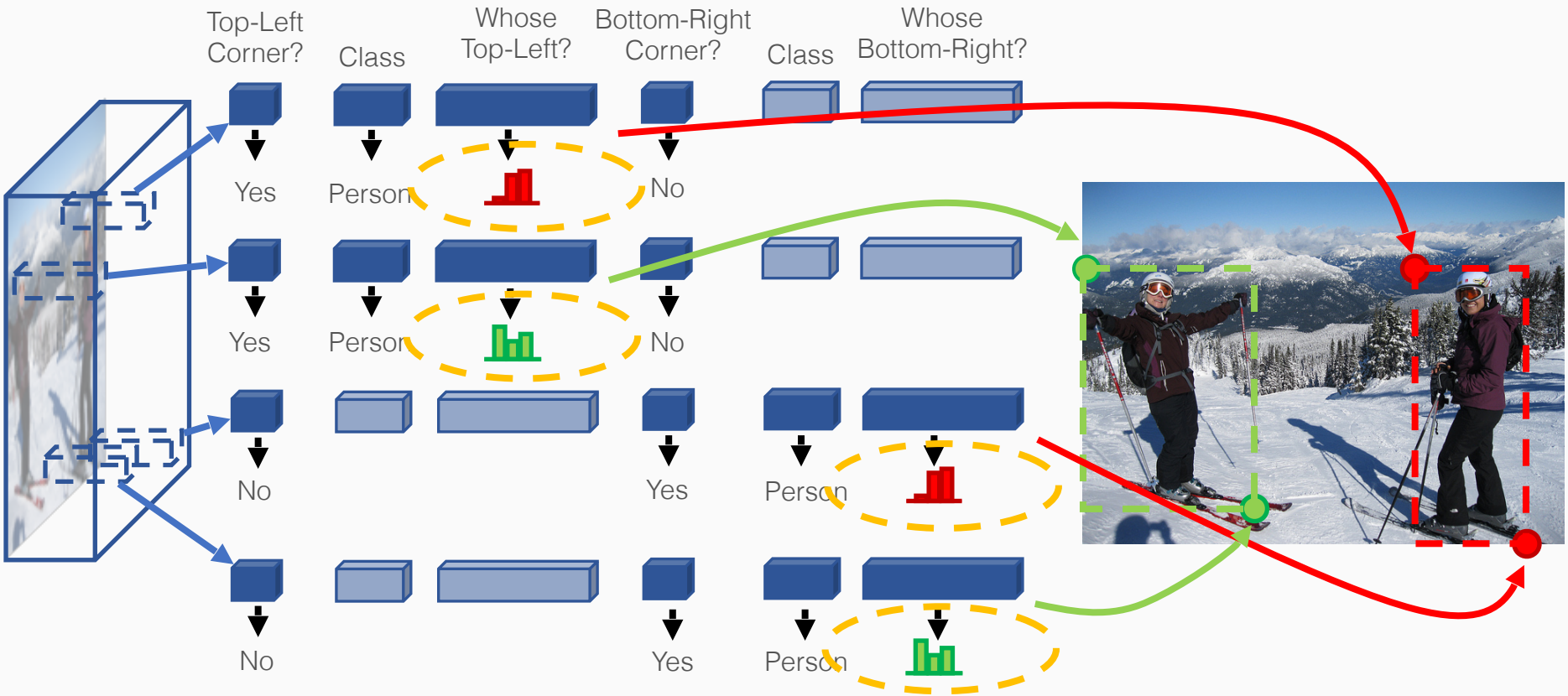
CornerNet: Detecting objects as paired key points, Law et al., ECCV 2018

CornerNet - Architecture



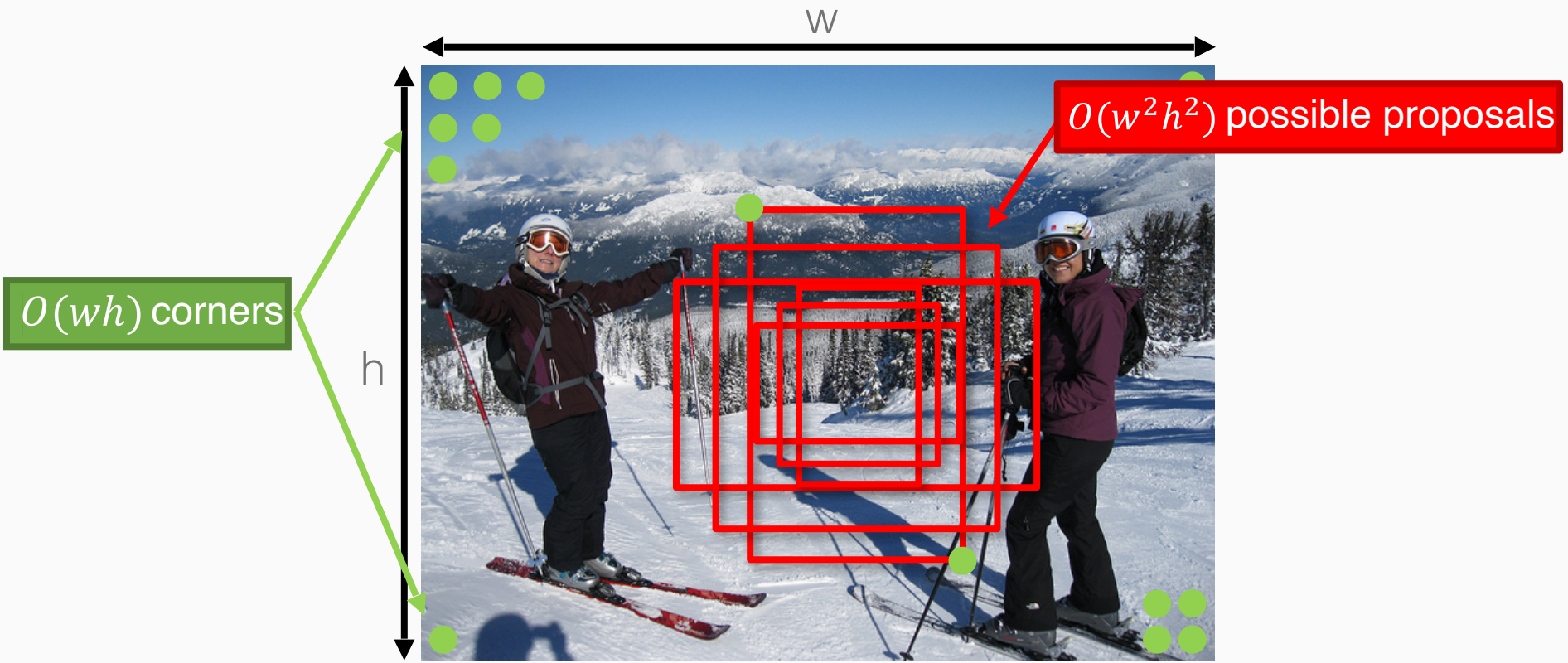
CornerNet: Detecting objects as paired key points, Law et al., ECCV 2018

CornerNet - Architecture



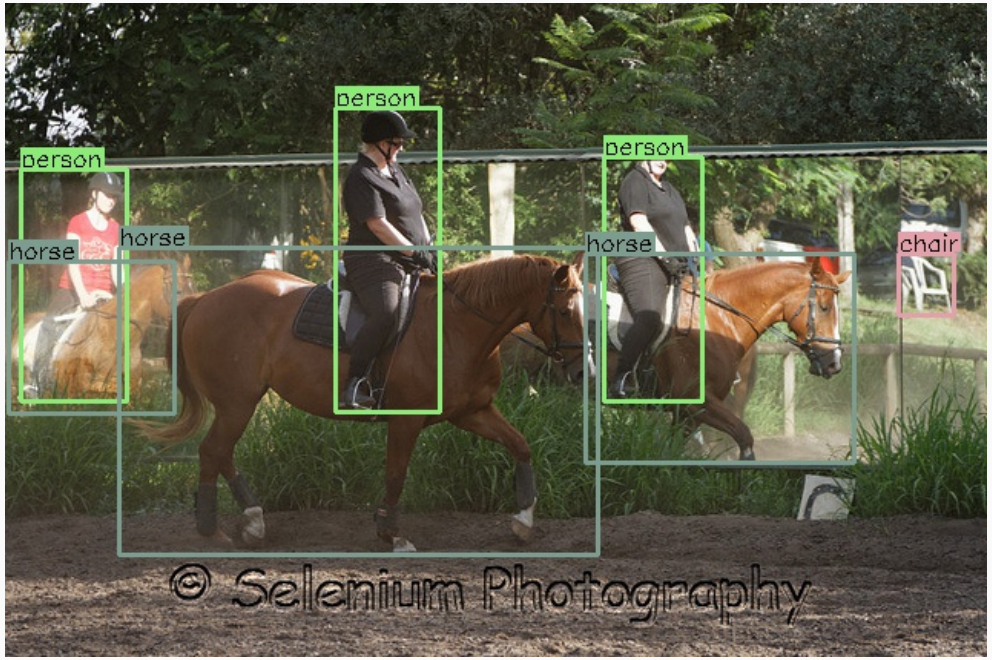
CornerNet: Detecting objects as paired key points, Law et al., ECCV 2018

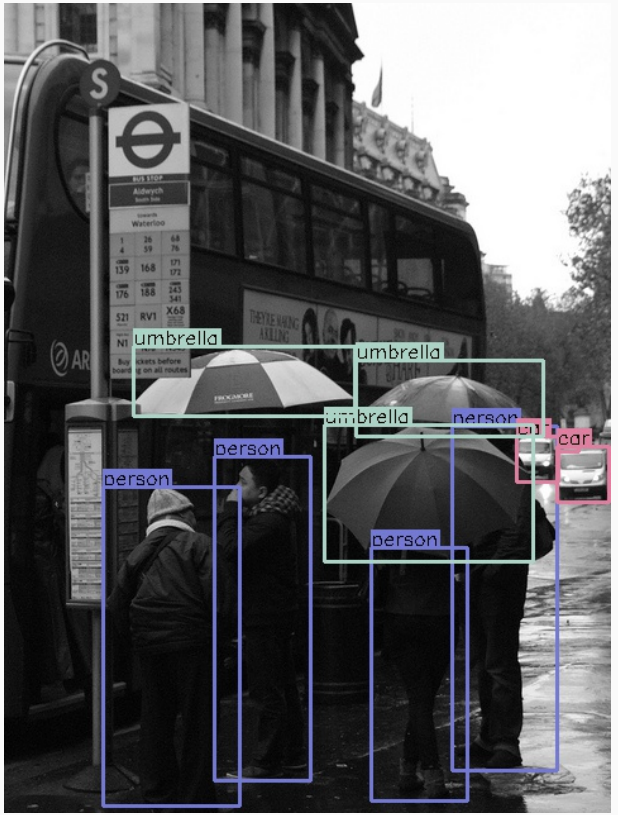
CornerNet - Advantages of detecting corners



Represent $O(w^2h^2)$ possible proposals using only $O(wh)$ corners

CornerNet - Examples





CornerNet: Detecting objects as paired key points, Law et al., ECCV 2018

CenterNet

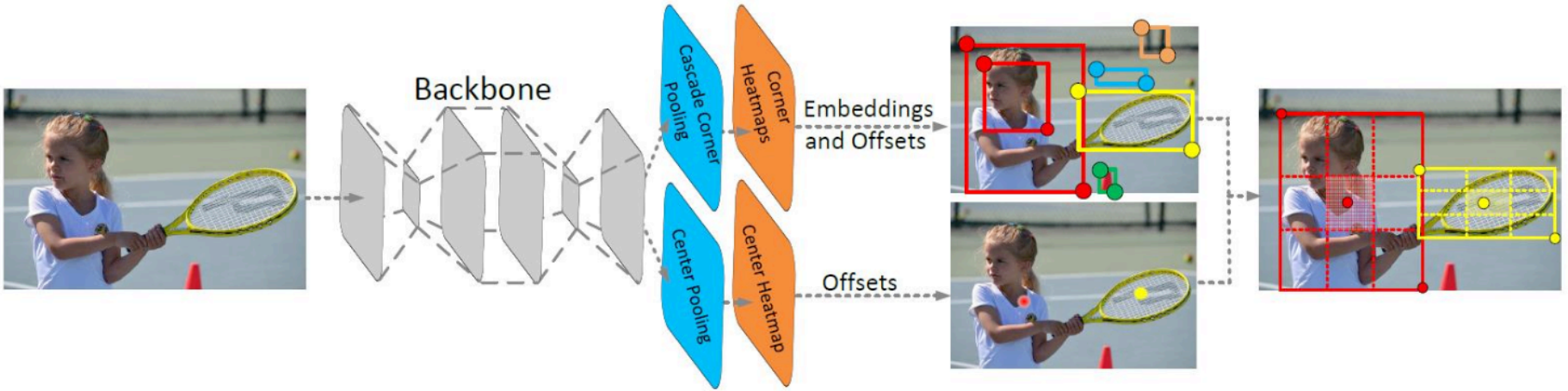


Figure 2: Architecture of CenterNet. A convolutional backbone network applies cascade corner pooling and center pooling to output two corner heatmaps and a center keypoint heatmap, respectively. Similar to CornerNet, a pair of detected corners and the similar embeddings are used to detect a potential bounding box. Then the detected center keypoints are used to determine the final bounding boxes.

CenterNet: Keypoint triplets for object detection, Duan et al, ICCV 2019

CenterNet-2

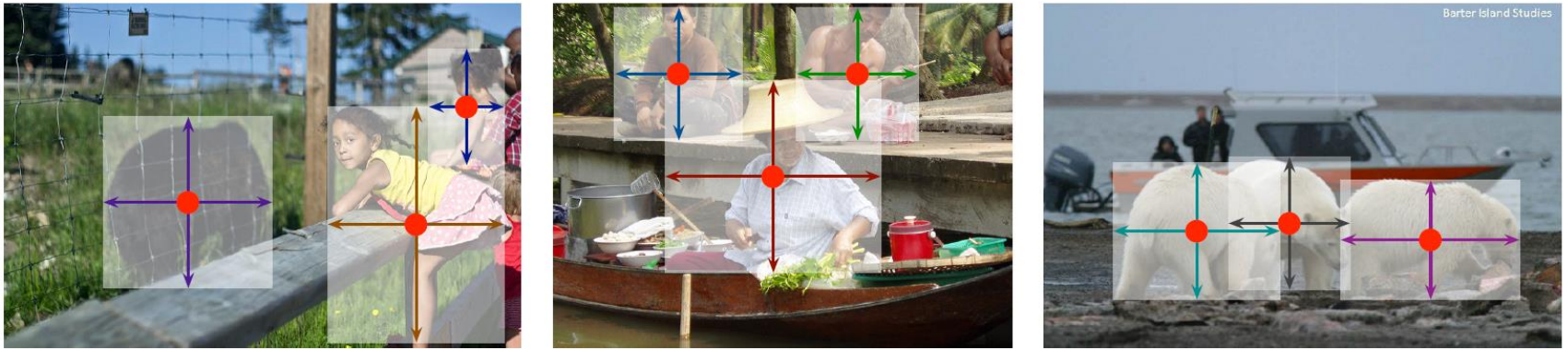
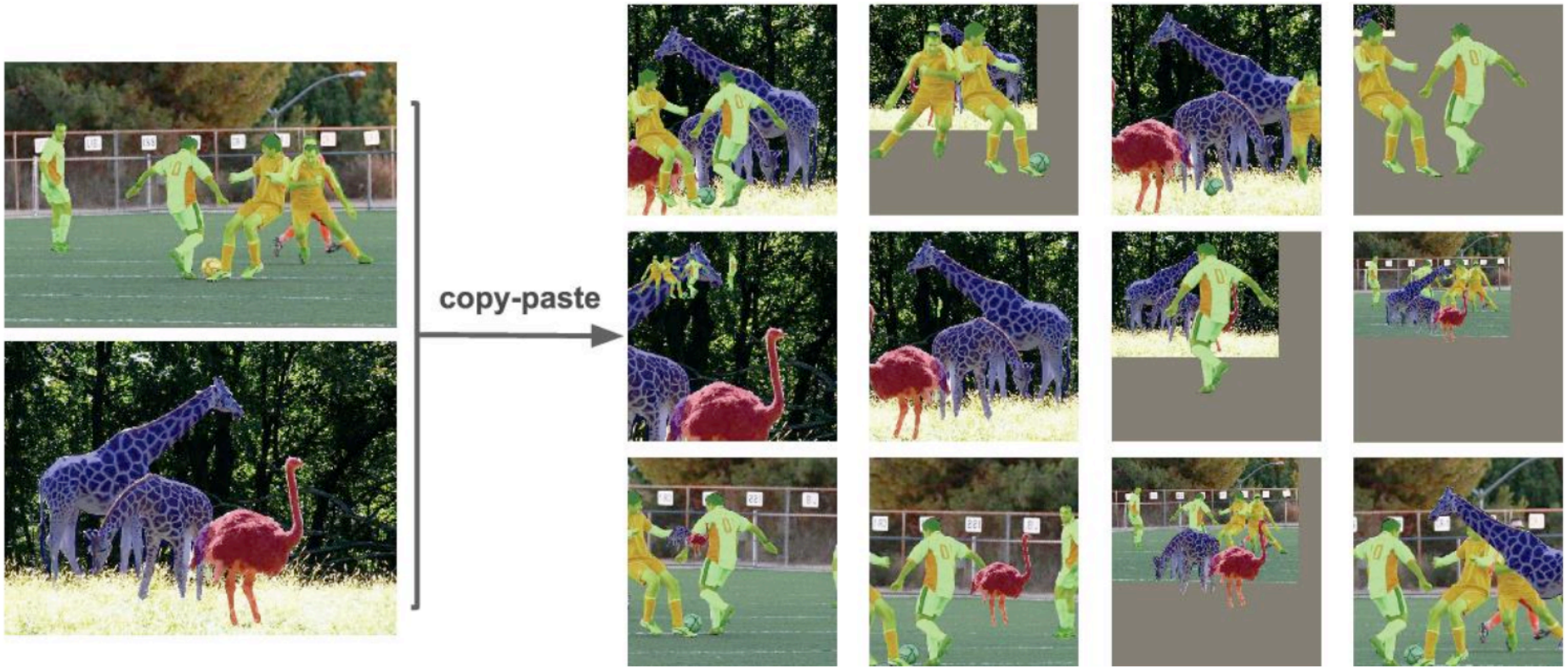


Figure 2: We model an object as the center point of its bounding box. The bounding box size and other object properties a inferred from the keypoint feature at the center. Best viewed in color.

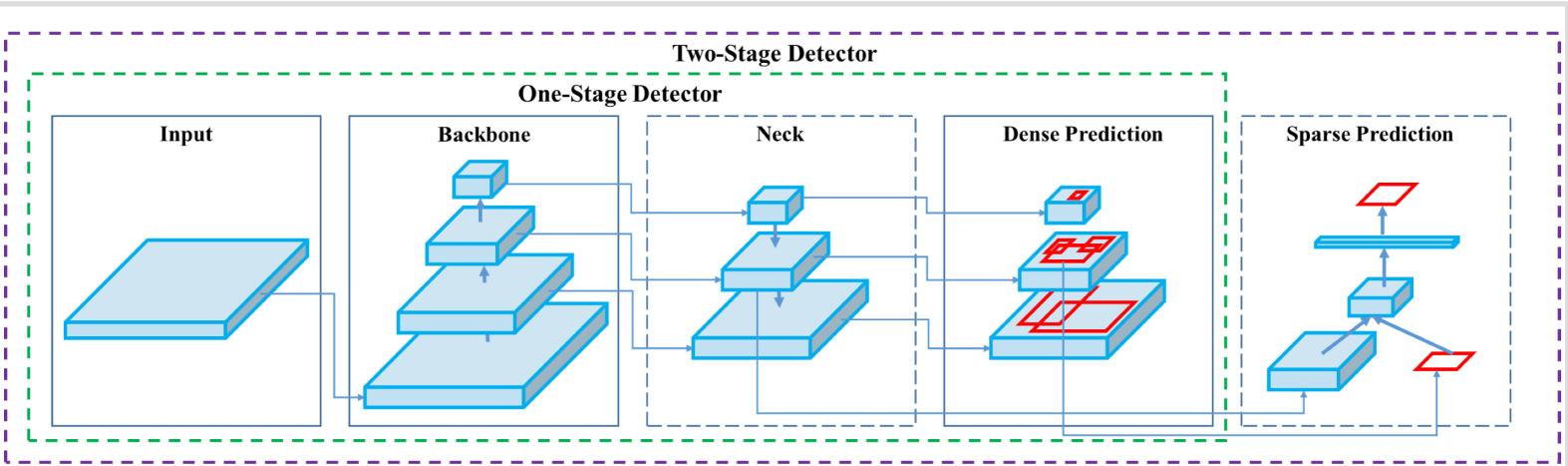
Objects as points, Xingyi Zhou, Dequan Wang, and Philipp Krahenbuhl. https://arxiv.org/pdf/1904.07850.pdf, 2019

Copy-Paste and Large-Scale Jittering Data Augmentation



Golnaz et al. Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation, CVPR 2021

YOLO v4



Input: { Image, Patches, Image Pyramid, … }

Backbone: { VGG16 [68], ResNet-50 [26], ResNeXt-101 [86], Darknet53 [63], … }

Neck: { FPN [44], PANet [49], Bi-FPN [77], … }

Head:

Dense Prediction: { RPN [64], YOLO [61, 62, 63], SSD [50], RetinaNet [45], FCOS [78], … }

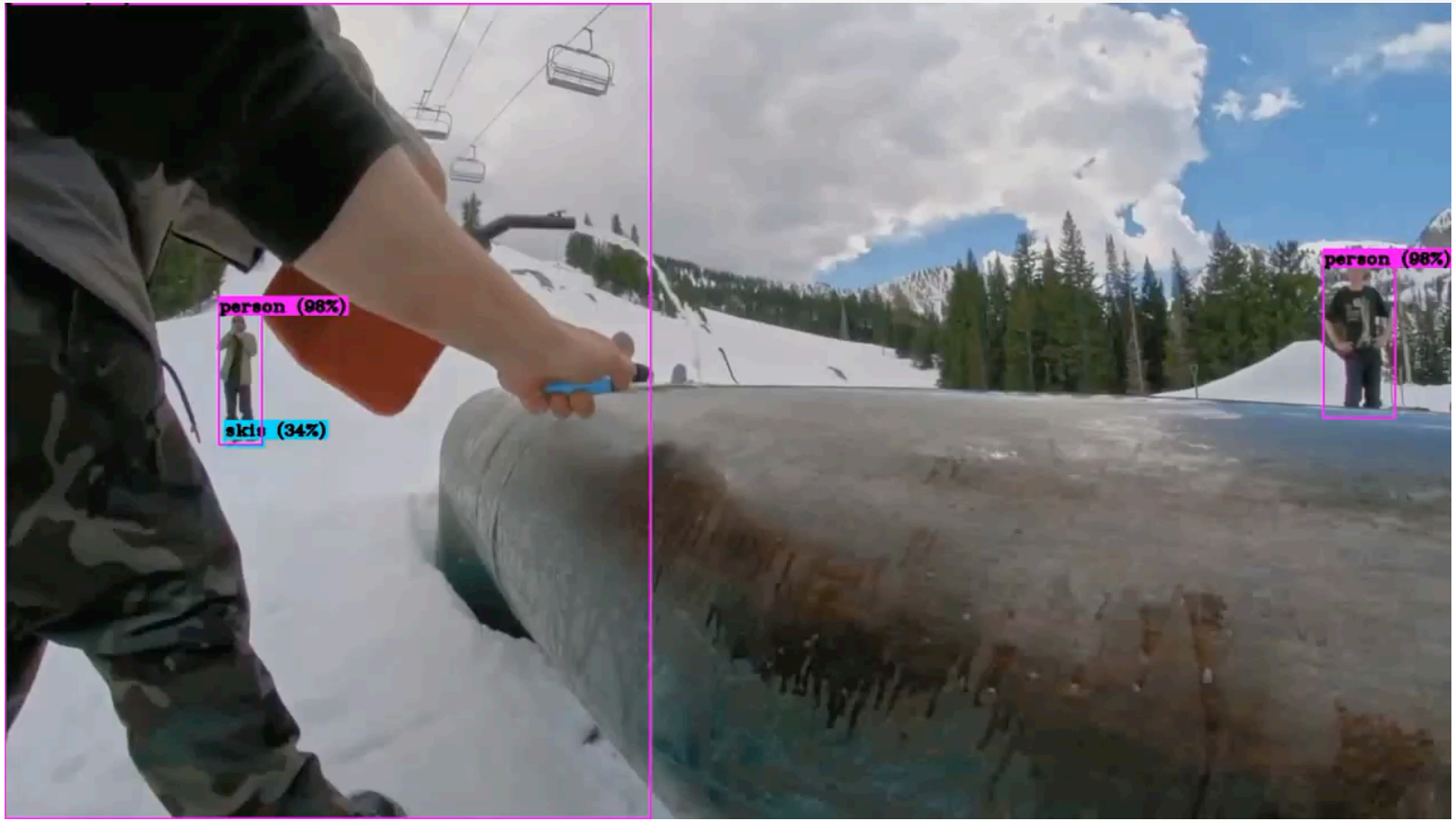
Sparse Prediction: { Faster R-CNN [64], R-FCN [9], … }

YOLOv4 consists of:

- Backbone: CSPDarknet53 [81]
- Neck: SPP [25], PAN [49]
- Head: YOLOv3 [63]

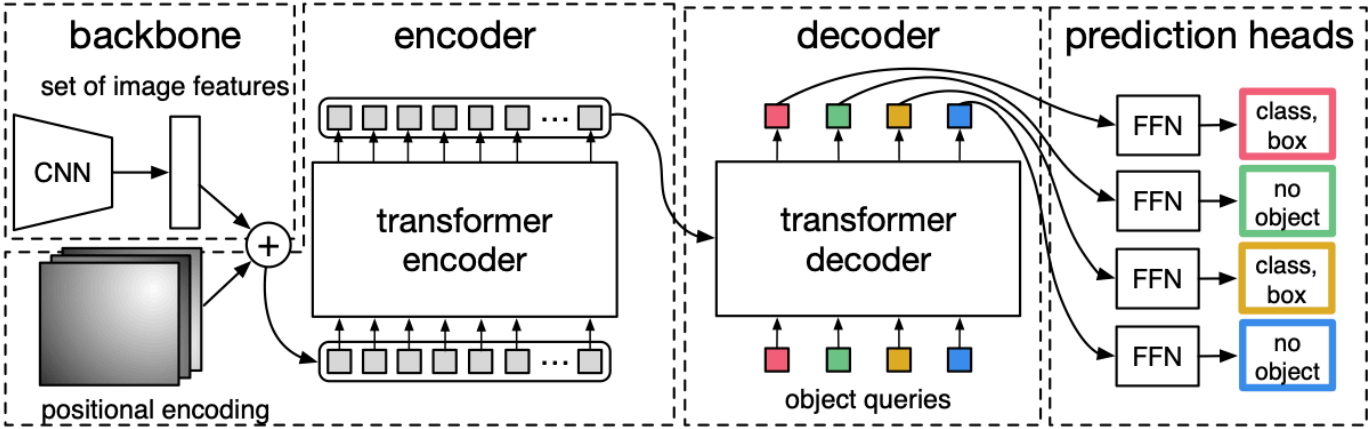
YOLOv4: Optimal speed and accuracy of object detection, Bochkovskiy et al., arXiv 2020

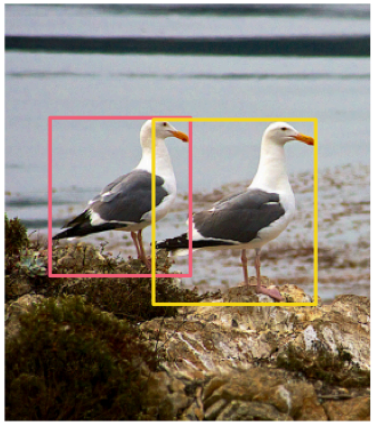
YOLO v4



https://www.youtube.com/watch?v=1_SiUOYUoOI&feature=emb_logo

DETR: End to end object detection using transformer

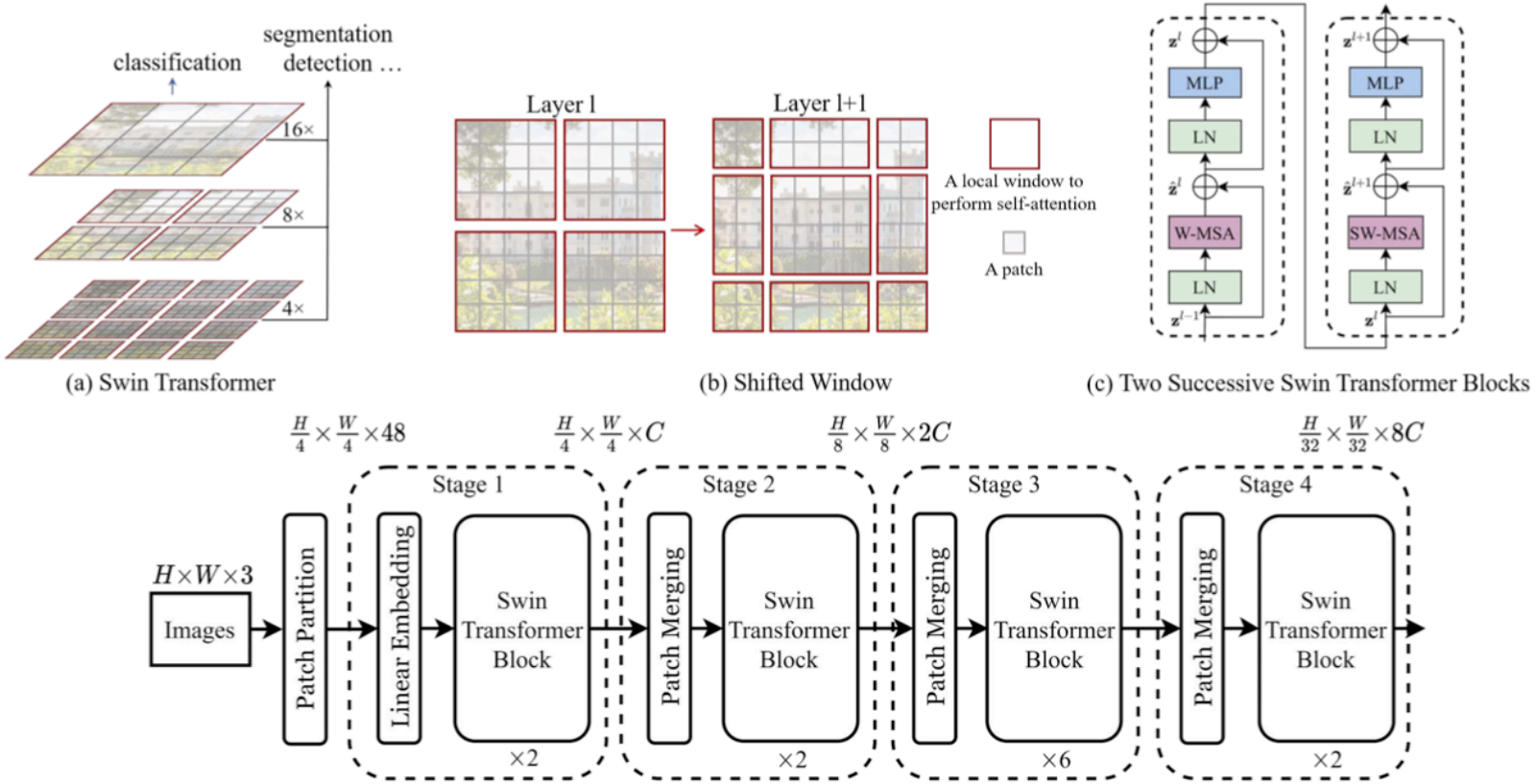




- Uses transformer architecture
- Learnable anchors (object queries)

End-to-End Object Detection with Transformers, Carion et al., ECCV 2020

Swin transformer: Hierarchical Vision Transformer using Shifted Windows



Ze Liu et al. Swin transformer: Hierarchical Vision Transformer using Shifted Windows, ICCV 2021

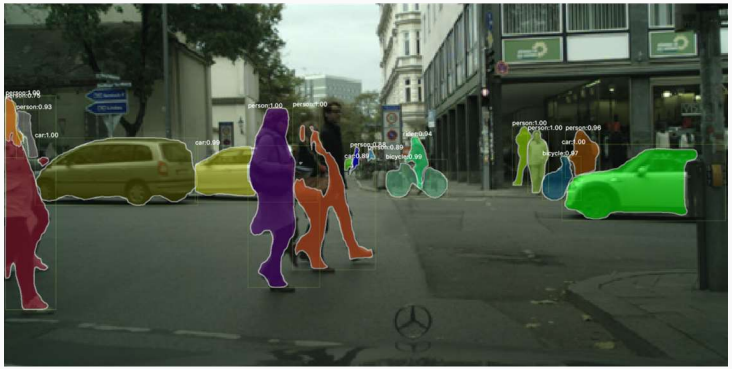
Object Detection on COCO test-dev

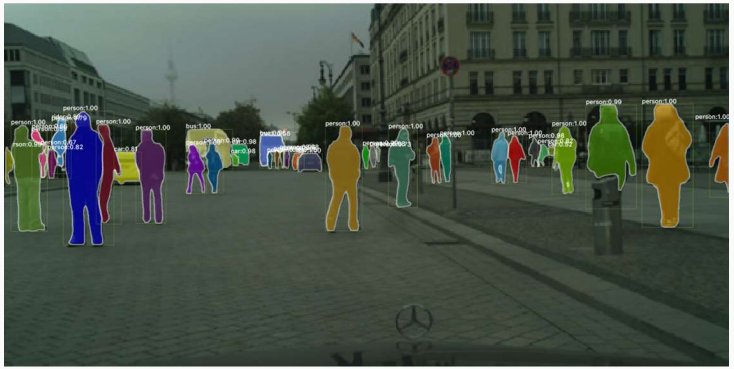
Methods	Architecture	Year	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Inference Speed (fps)
Two-Stage Detectors:									
R-CNN (only test on Pascal VOC) [1]	VGG-16	2014	–	–	–	–	–	–	0.02
SPPNet (only test on Pascal VOC) [2]	VGG-16	2015	–	–	–	–	–	–	–
Fast R-CNN [3]	VGG-16	2015	19.7	35.9	–	–	–	–	0.5
Faster R-CNN [4]	VGG-16	2015	21.9	42.7	–	–	–	–	5
Fast RCNN + OHEM [5]	VGG-16	2016	22.6	42.5	22.2	5.0	23.7	37.9	5
Faster-RCNN+++ [6]	ResNet-101	2016	34.9	55.7	37.4	15.6	38.7	50.9	2.4
R-FCN [7]	ResNet-101	2016	29.9	51.9	–	10.8	32.8	45.0	6
Faster-RCNN w Feature Pyramid Net [8]	ResNet-101	2016	36.2	59.1	–	18.2	39.0	48.2	7
Deformable R-FCN [9]	ResNet-101	2017	34.5	55.0	–	14.0	37.7	50.3	5
Mask-RCNN [10]	ResNeXt-101	2017	39.8	62.3	43.4	22.1	43.2	51.2	5
Mask-RCNN-GroupNorm [11]	ResNet-101	2018	42.3	62.8	46.2	–	–	–	5
SNIPER [12]	ResNet-101	2018	46.1	67.0	51.6	29.6	48.9	58.1	–
Faster-RCNN with DCN-v2 [13]	ResNet-101	2019	44.0	66.3	48.8	24.4	48.1	59.6	–
Faster-RCNN-TridentNet [14]	ResNet-101	2019	42.7	63.6	46.5	23.9	46.6	56.6	2.4
Faster-RCNN-TridentNet-Deformable [14]	ResNet-101	2019	48.4	69.7	53.5	31.8	51.3	60.3	0.7
DetectoRS [15]	ResNeXt-101	2020	55.7	74.2	61.1	37.7	58.4	68.1	–
Swin-L (HTC++) [16]	Swin Transformer	2021	**58.7**	–	–	–	–	–	–
SwinV2-G (HTC++) [17]	Swin Transformer V2	2021	**62.5**	–	–	–	–	–	–
One-Stage Detectors:									
SSD [18]	VGG-16	2016	28.8	48.5	30.3	10.9	31.8	43.5	19
YOLOv2 [19]	DarkNet-19	2017	21.6	44.0	19.2	5.0	22.4	35.5	40
YOLOv3 [20]	DarkNet-53	2017	33.0	57.9	34.4	18.3	35.4	41.9	20
RetinaNet [21]	ResNet-101	2017	39.1	59.1	42.3	21.8	42.7	50.2	5.4
CornerNet [22]	Hourglass-104	2018	42.1	57.8	45.3	20.8	44.8	56.7	4.1
ExtremeNet [23]	Hourglass-104	2019	42.1	61.1	45.9	24.1	45.5	52.8	3.1
CenterNet-1 [24]	Hourglass-104	2019	45.1	63.9	49.3	26.6	47.1	57.7	7.8
EfficientDet-D2(768, single-scale) [25]	EfficientNet	2020	43.0	62.3	46.2	–	–	–	41.6
EfficientDet-D4(1024, single-scale) [25]	EfficientNet	2020	49.4	69.0	53.4	–	–	–	13.5
EfficientDet-D7(1536, single-scale) [25]	EfficientNet	2020	52.2	71.4	56.3	–	–	–	3.8
DETR [26]	R50+Transformer	2020	42.0	62.4	44.2	20.5	45.8	61.1	28

Credit: Weidi Xi

Instance segmentation

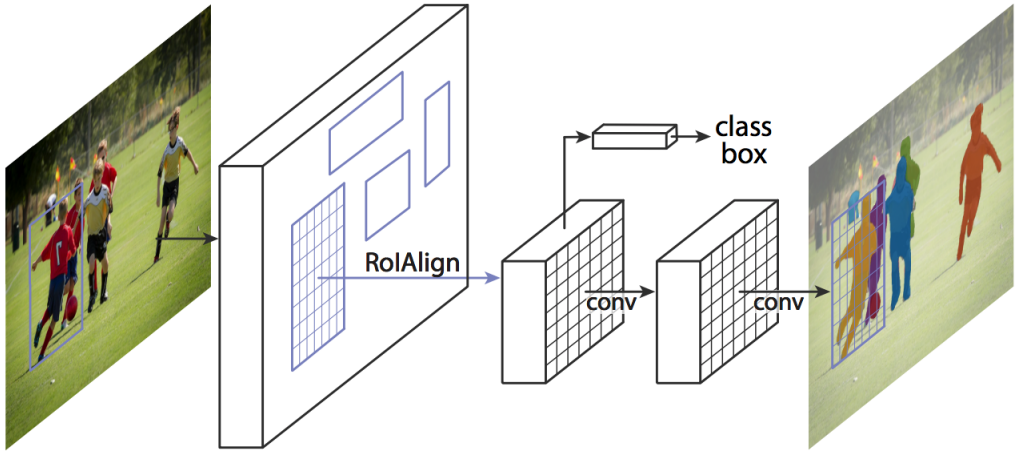
- Given an image produce instance-level segmentation
  - Which class does each pixel belong to?
  - Which instance does each pixel belong to?





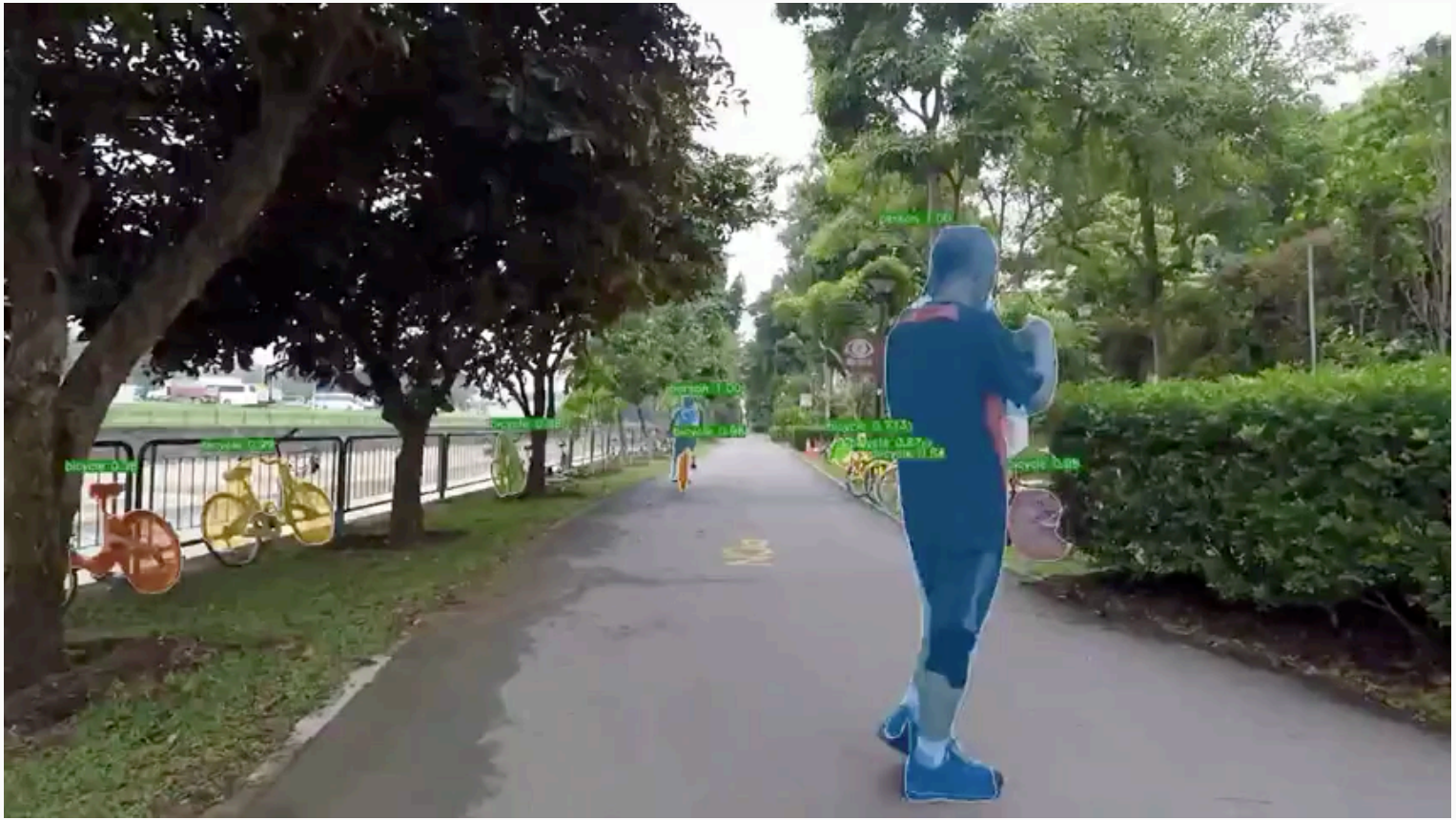
Mask R-CNN

- Extend Faster R-CNN to predict mask as well as a box



Mask R-CNN, He et al., CVPR 2017

Mask R-CNN - video example



Mask R-CNN, He et al., CVPR 2017

https://www.youtube.com/watch?v=UWtac4cFERM

Challenges

- Detection has **not** saturated – AP performance has only reached 63% on COCO
- New benchmarks:
  - LVIS (Large Vocabulary Instance Segmentation): 1200 categories, 164K images, 2.2M instance segmentations
  - Open Image-v6: 600 categories, 15,851K bounding boxes.
- Few shot object category detection
- Open set object category detection
- Weak/self-supervision

Credit: A Zisserman

That’s it folks!