Beyond classification:

Object detection, Segmentation, Human pose estimation

Gül Varol

IMAGINE team, École des Ponts ParisTech

gul.varol@enpc.fr

http://imagine.enpc.fr/~varolg/

@RecVis, 07.11.2023

With many slides from: Weidi Xie, I. Laptev, R. Girshick, K. He, N. Murray, A. Karpathy, L. Fei Fei, J. Johnson, A. Torralba, K. He, S. Yeung, J. Sivic, M. Aubry

École des Ponts ParisTech

Neural Networks

Last week: Neural networks for visual recognition

(G. Varol)

This week: Beyond classification: Object detection, Segmentation, Human pose estimation

(G. Varol)

Next week: Large-scale image and video search

(Josef. Sivic)

Recap: Neural networks for image classification

AlexNet (2012)

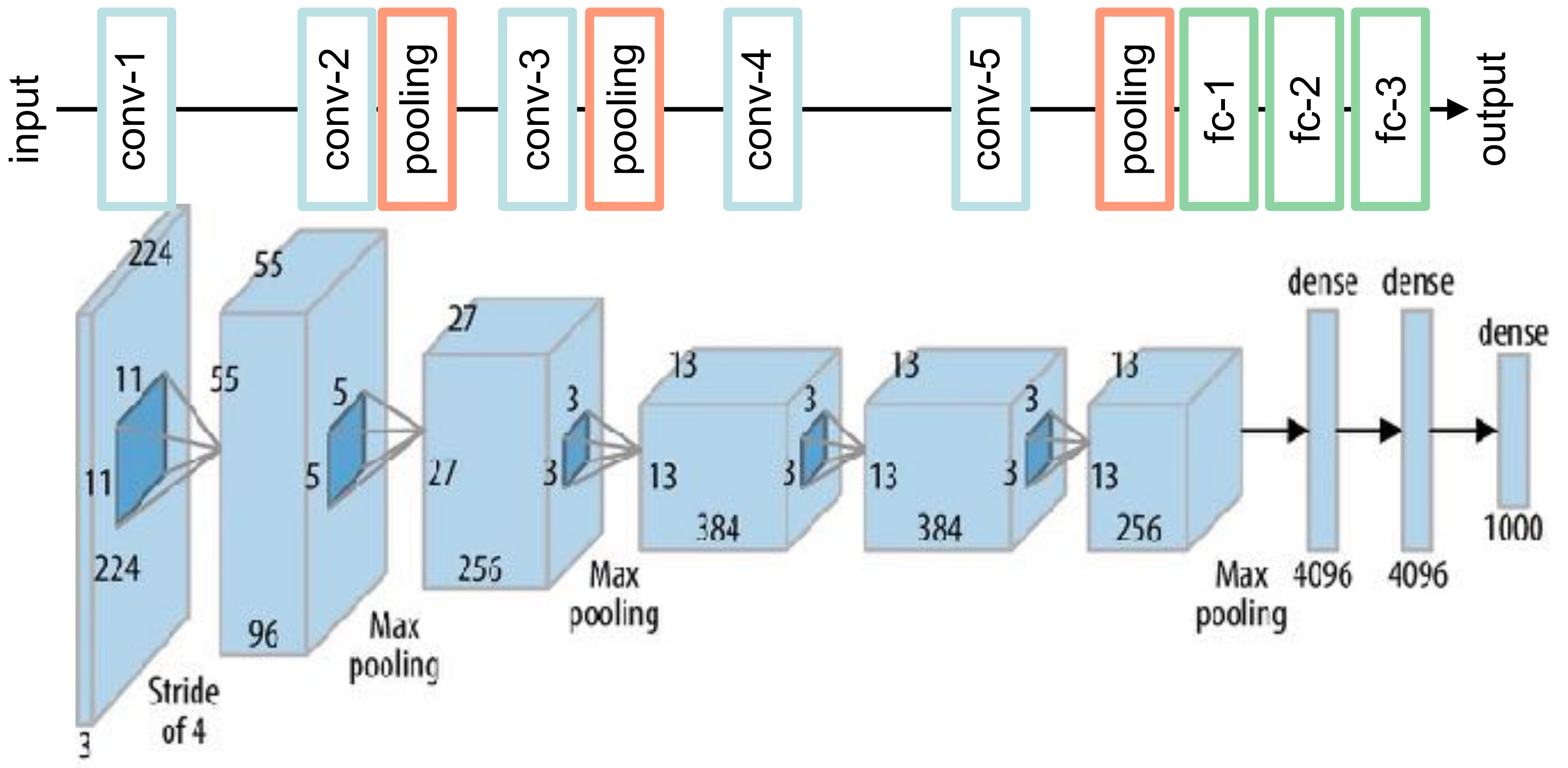


Image source: oreilly.com

VGG-16 (2015)

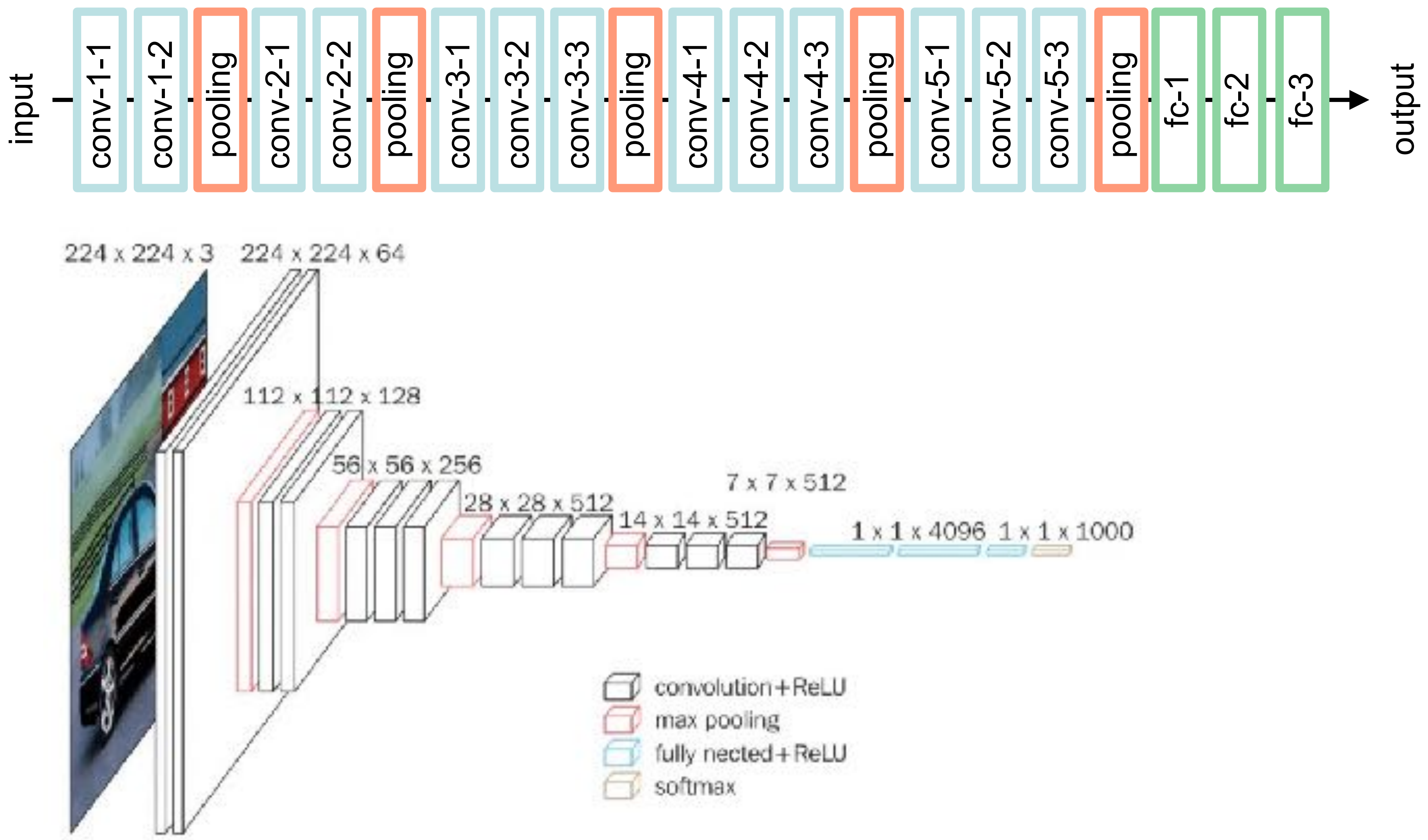
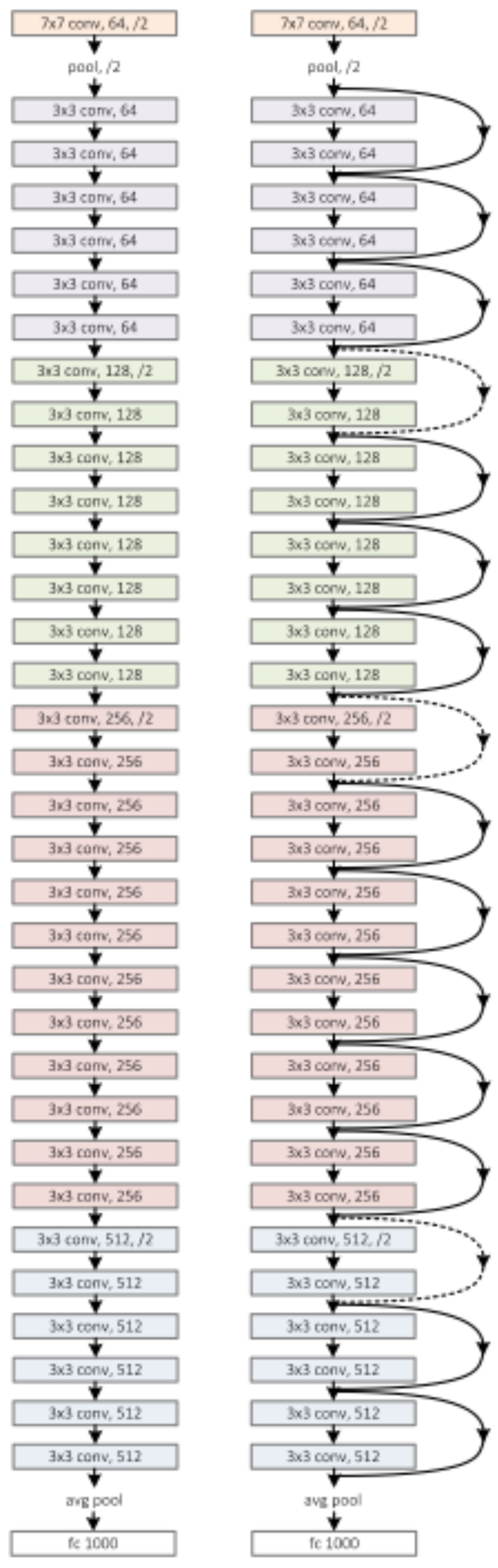


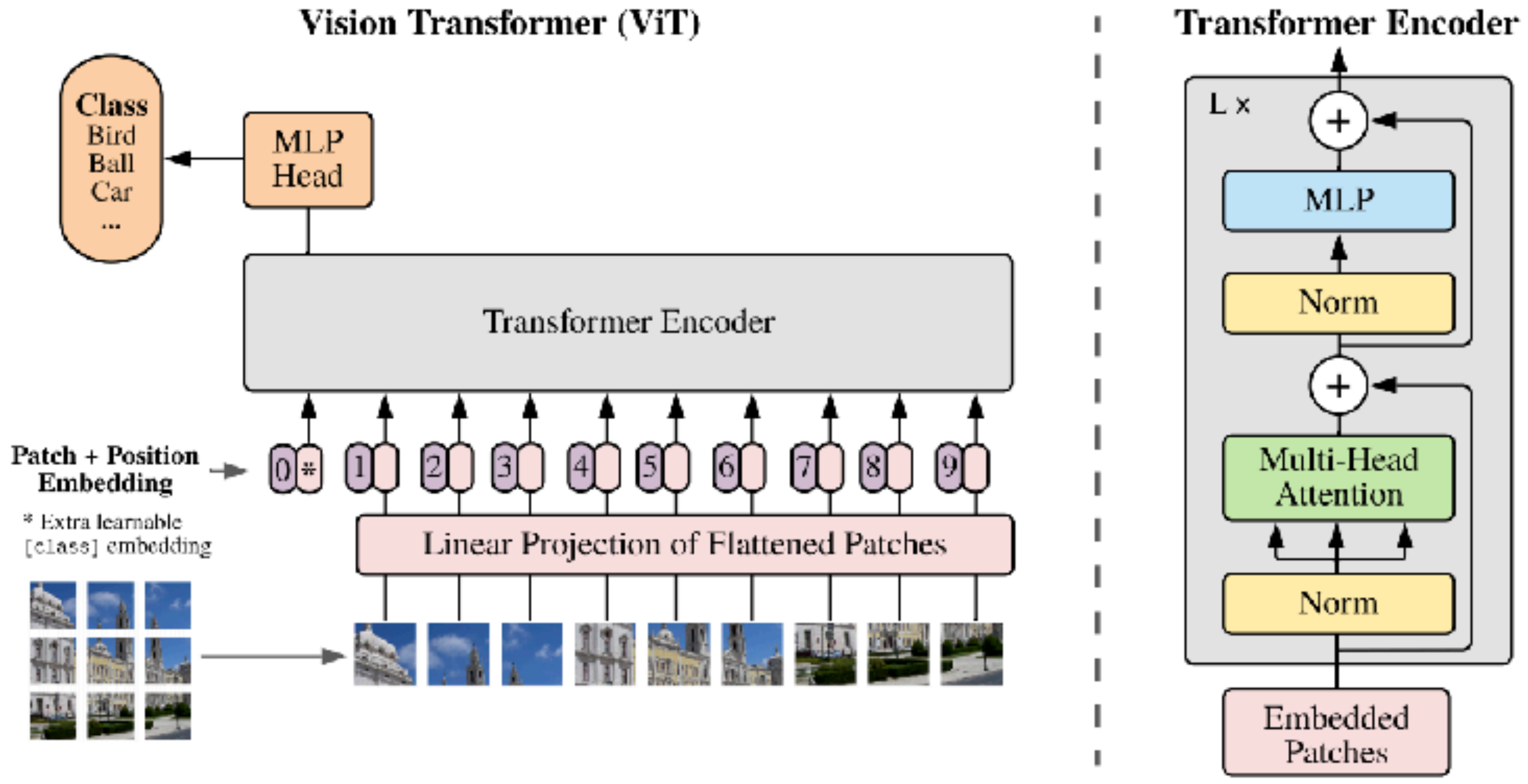
Image source: neurohive.io/en/popular-networks/vgg16/

ResNet (2016)



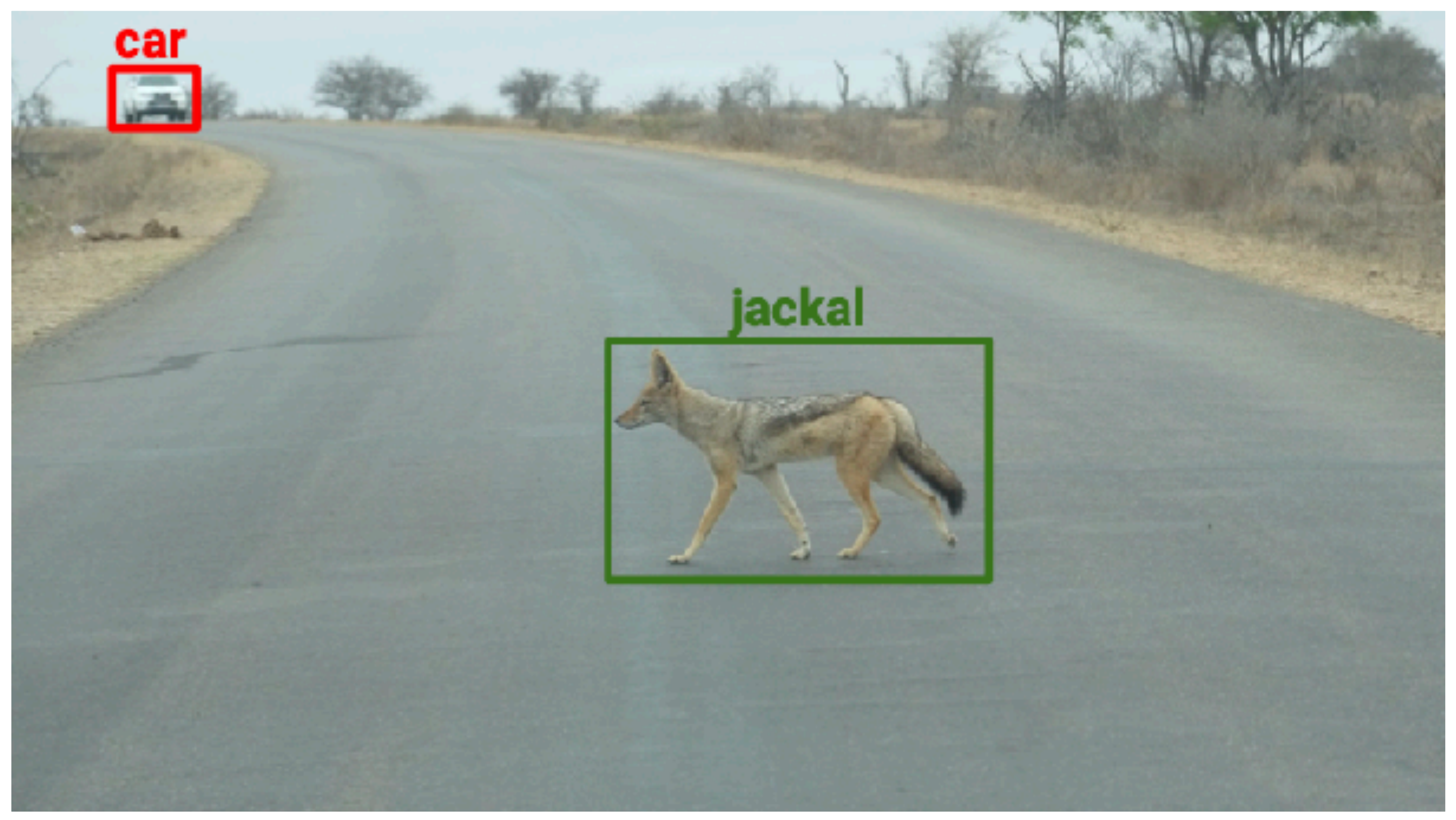
VGG-19 ResNet-34

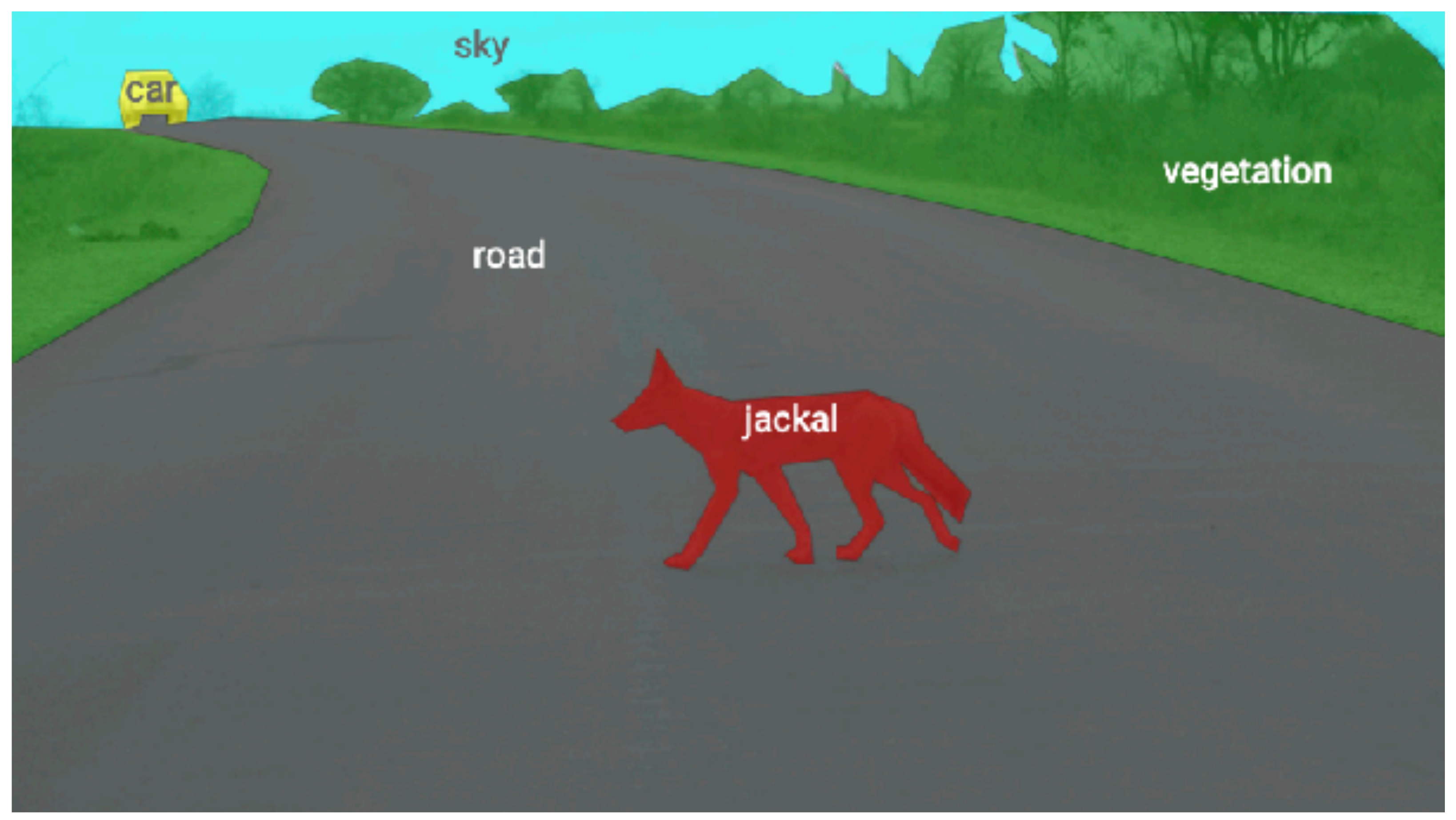
ViT (2021)



Agenda

- 0. Intro to structured outputs
- 1. Object detection (localization)
- 2. Segmentation
- 3. Human pose estimation





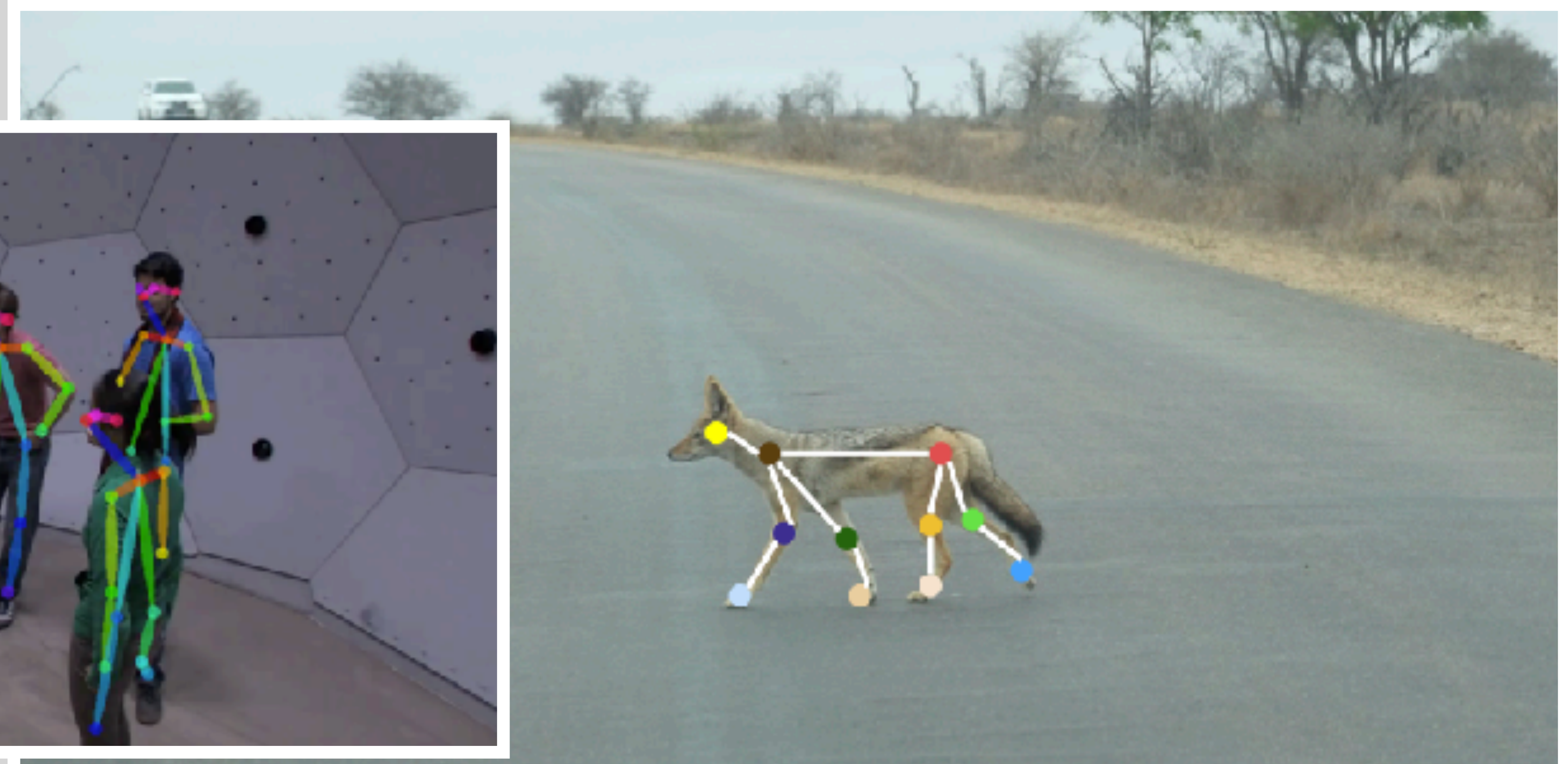
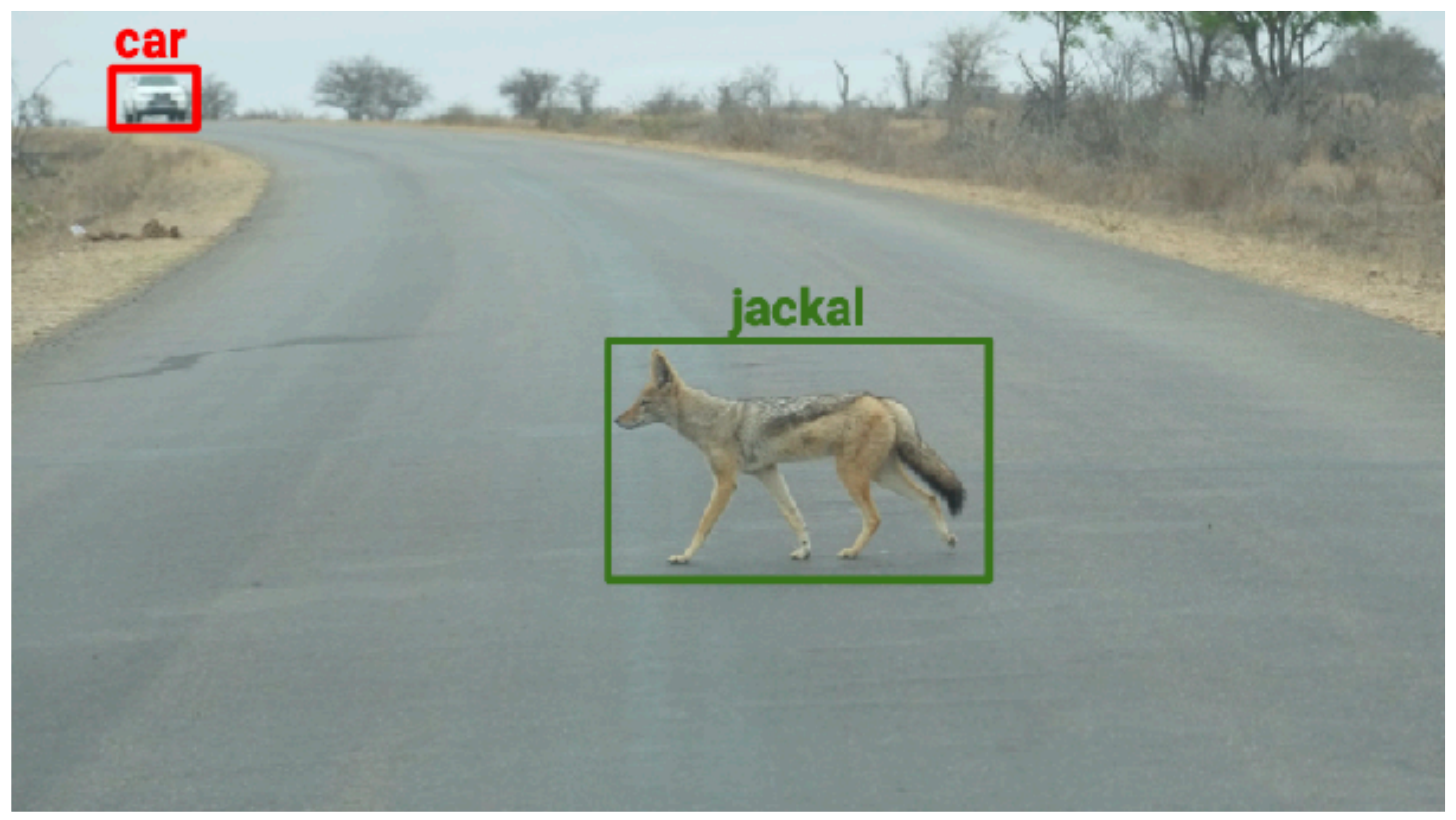
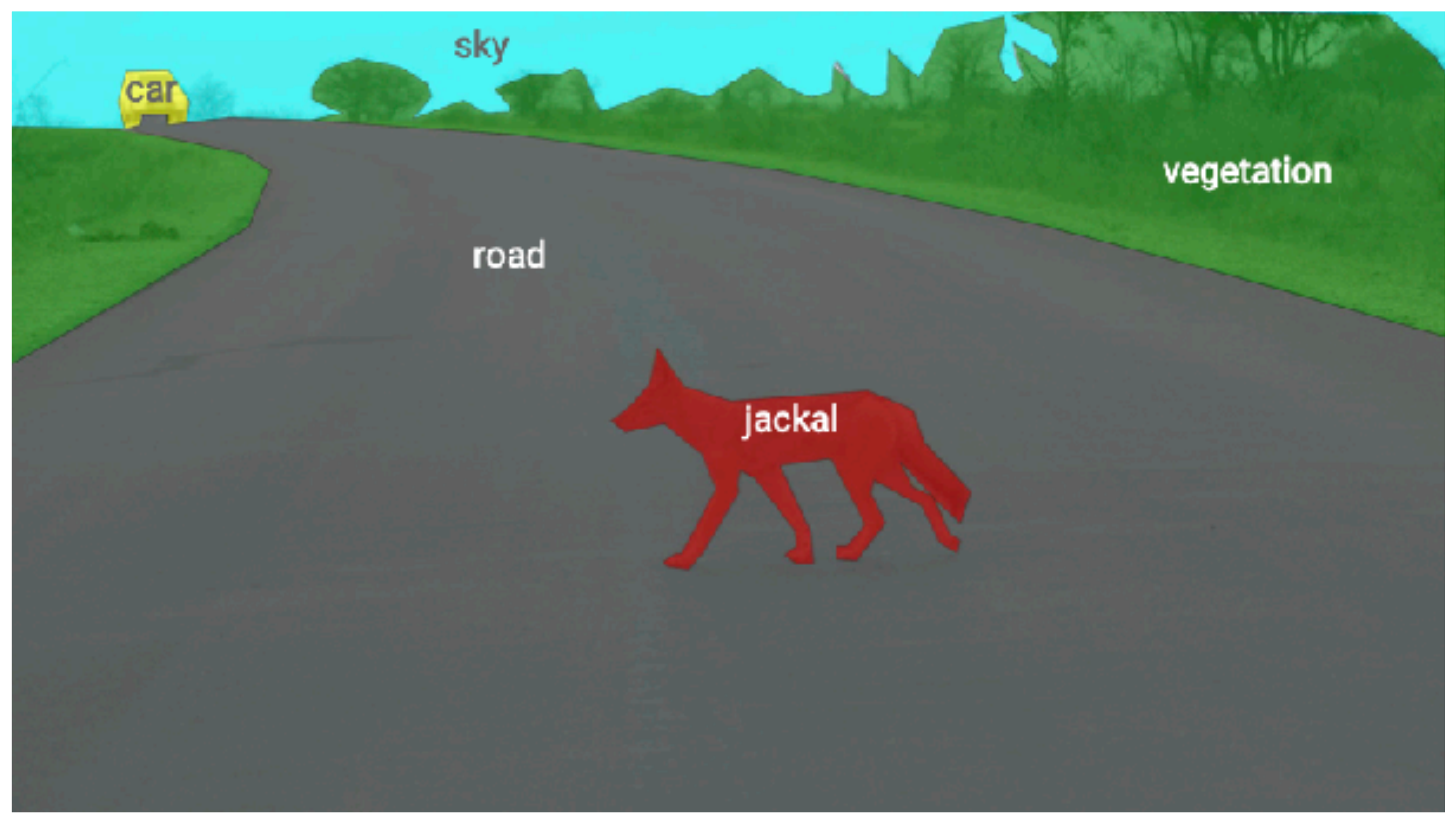


Image credits: Naila Murray

Agenda

- 0. Intro to structured outputs
- 1. Object detection (localization)
- 2. Segmentation
- 3. Human pose estimation





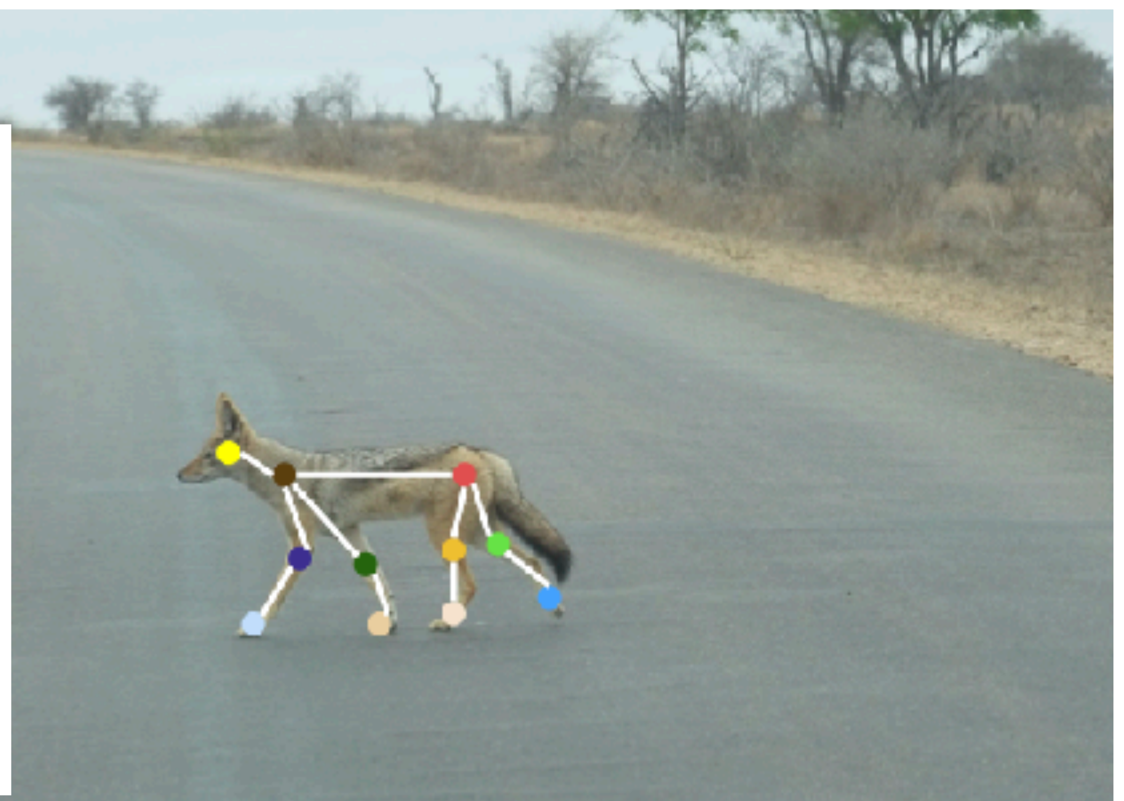
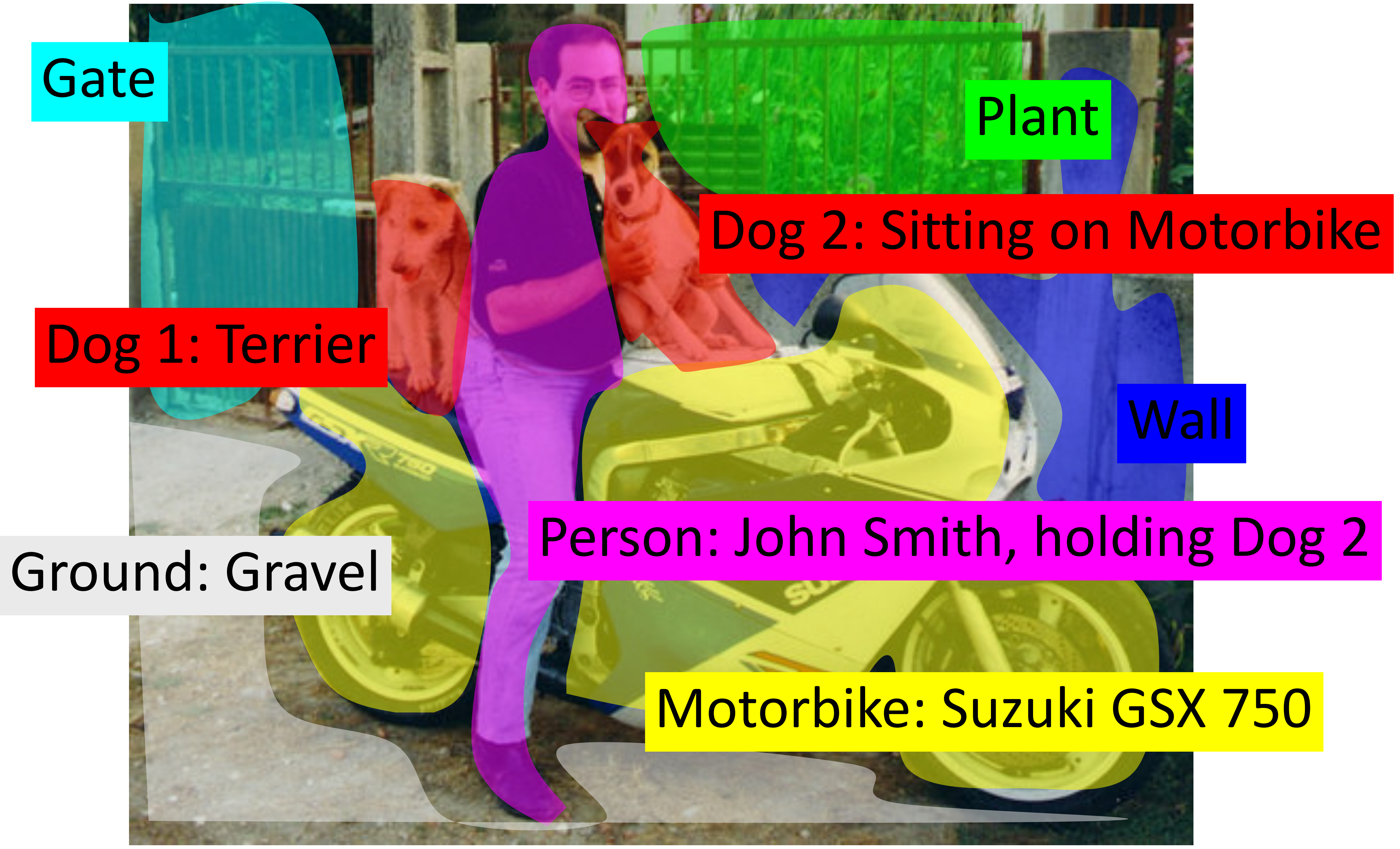


Image credits: Naila Murray

What we would like to do...

- Visual scene understanding
- **<u>What</u>** is in the image and **<u>where</u>**



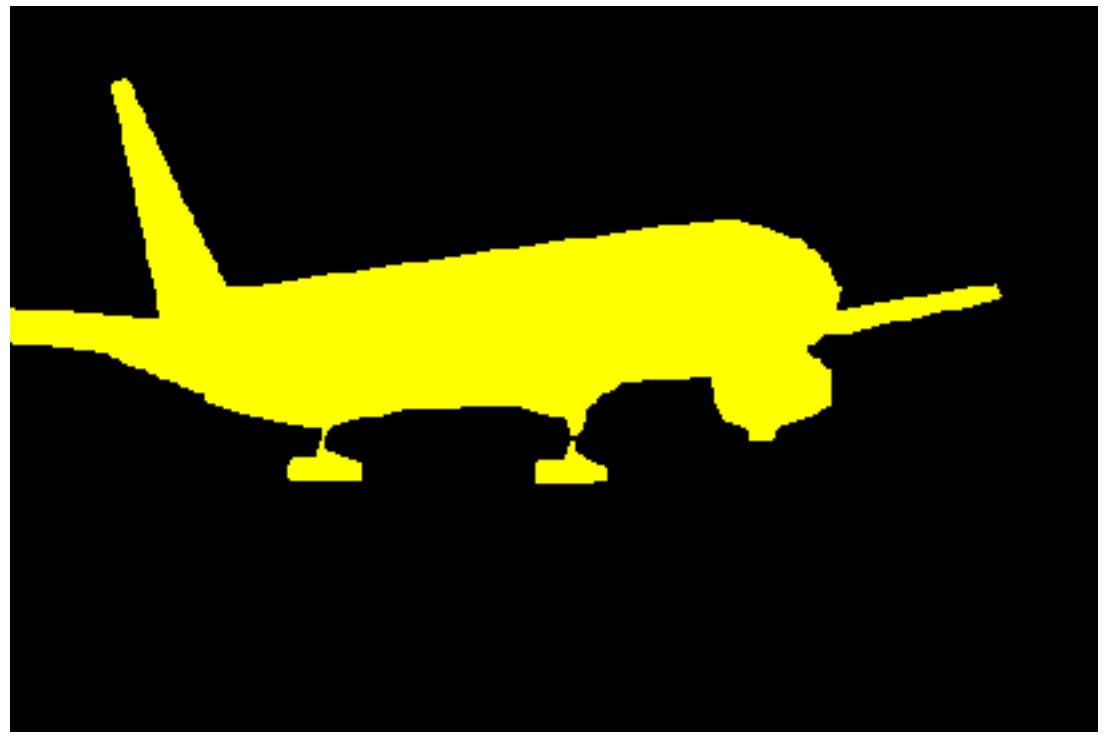
- Object categories, identities, properties, activities, relations, …

(Some) Fundamental Tasks in Computer Vision

- **Image Classification**
  – Does the image contain an aeroplane?
  (last lecture)

- **Object Class Detection/Localization**
  – Where are the aeroplanes (if any)?

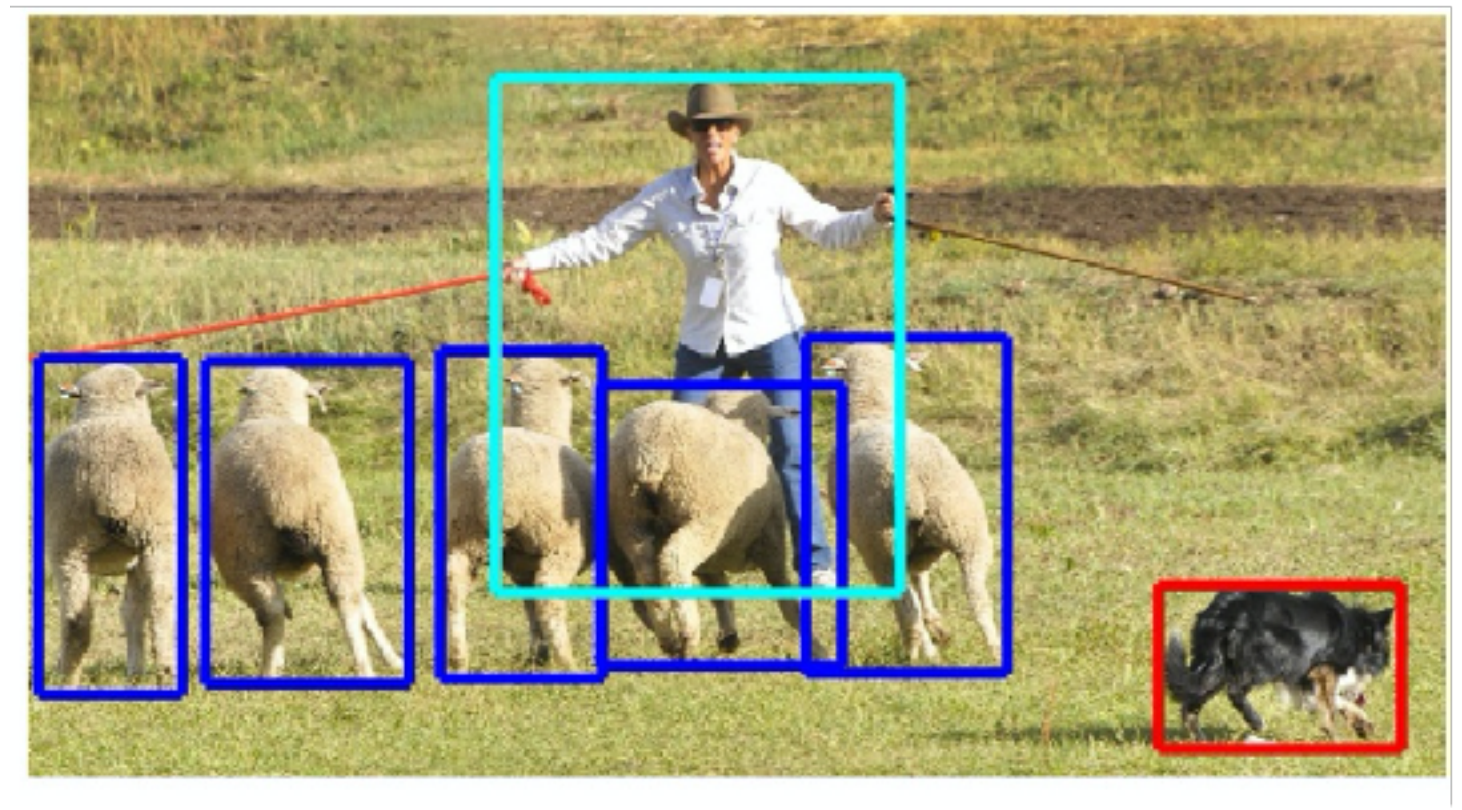
- **Object Class Segmentation**
  – Which pixels are part of an aeroplane (if any)?



(Some) Fundamental Tasks in Computer Vision

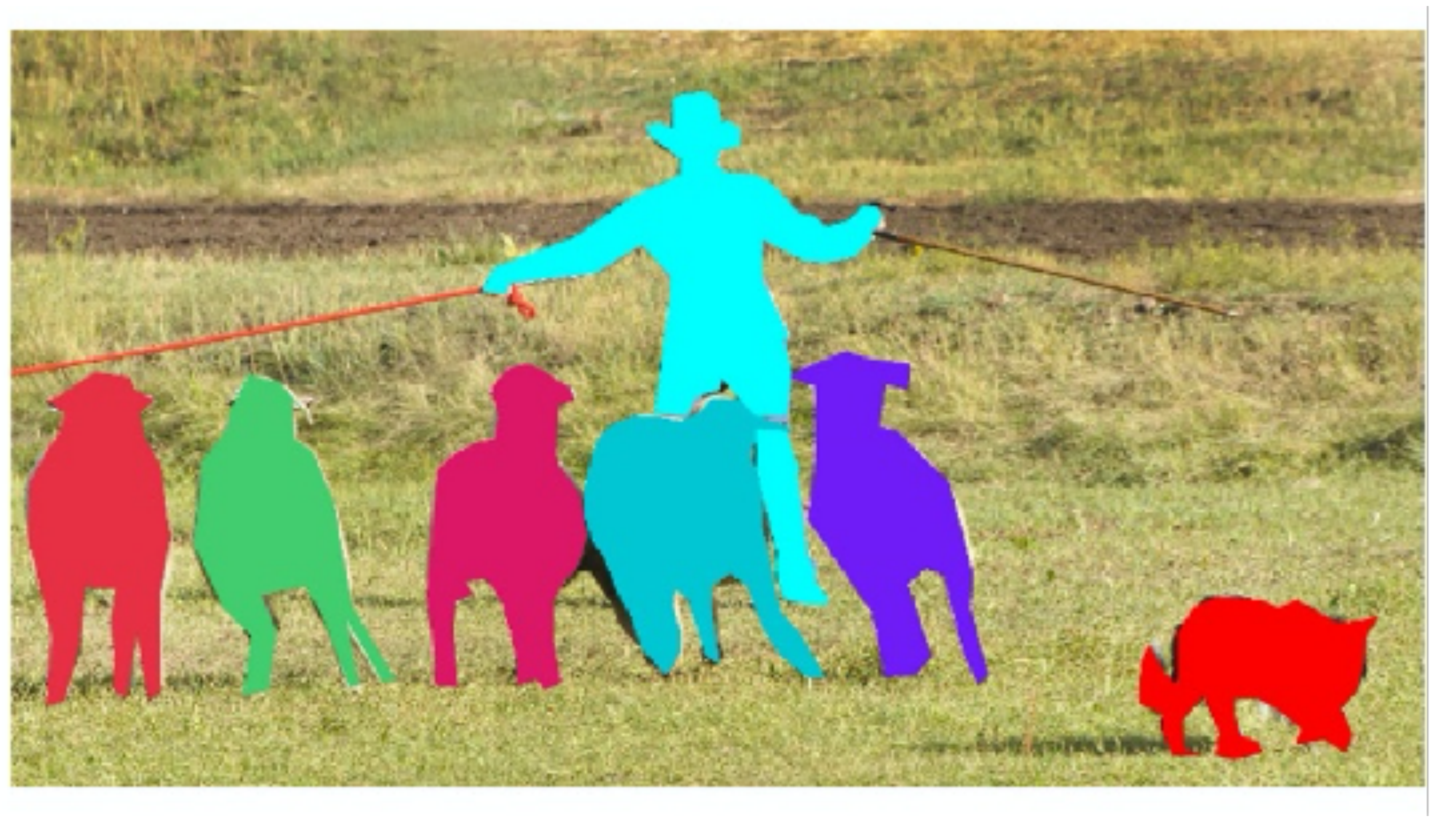
- Representing objects in the image:
  - Class labels
  - Bounding box
  - Semantic pixel-wise labels
  - Instance pixel-wise labels

Image Classification



Object Detection

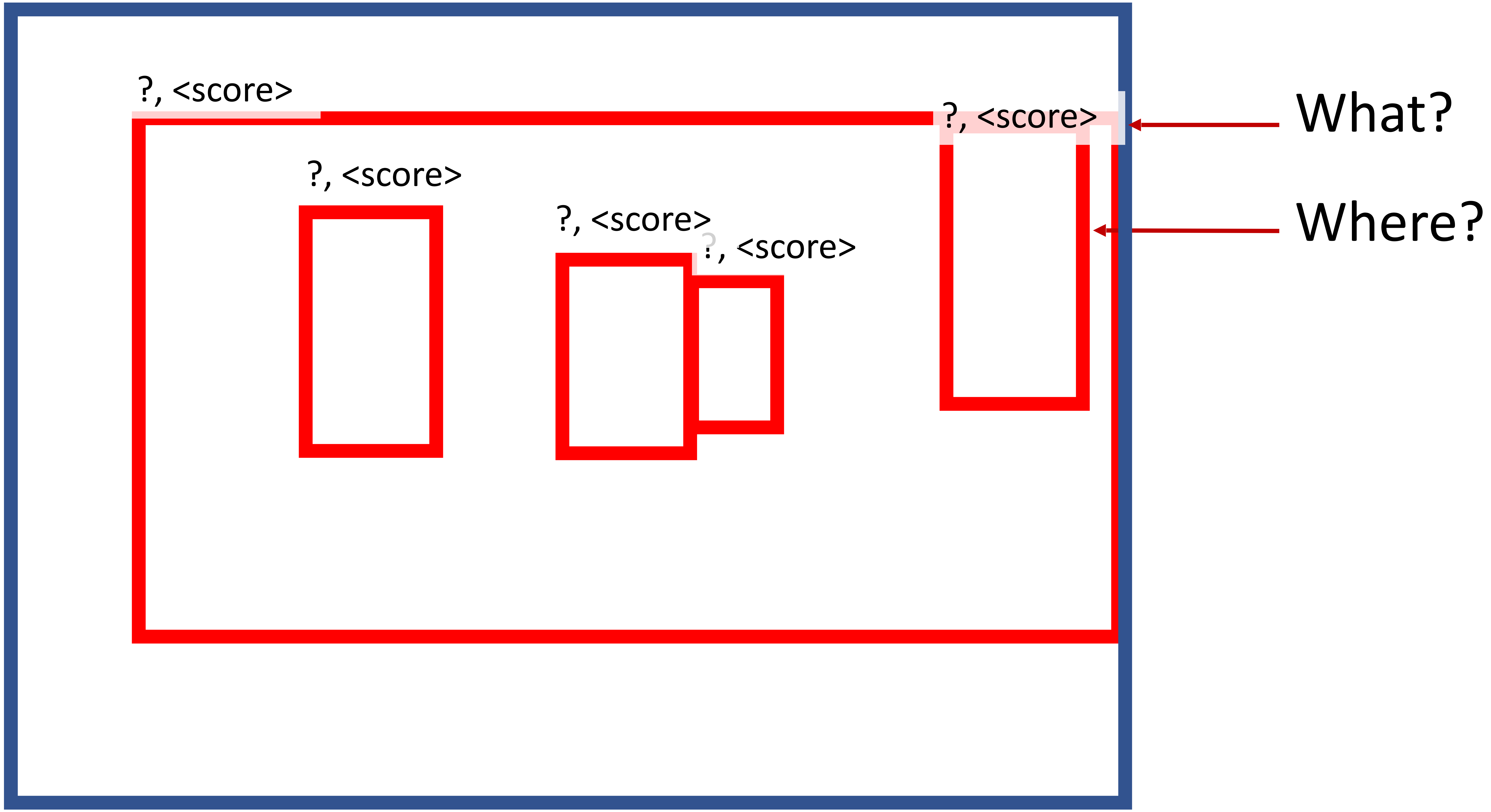
Semantic Segmentation



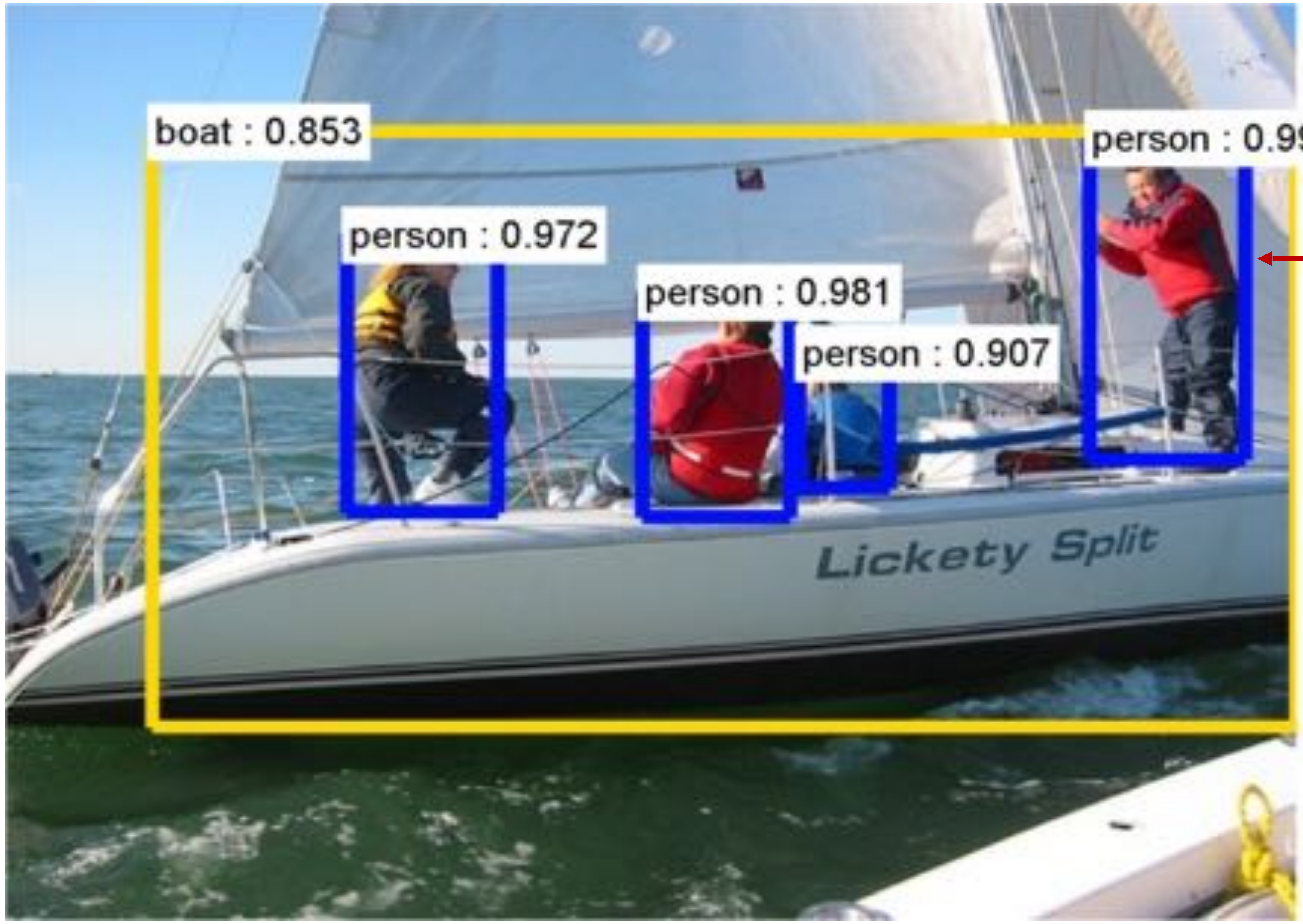
Instance Segmentation

Lin *et al.* “Microsoft COCO: Common Objects in Context”

Object Detection



Object Detection with **Bounding Boxes**

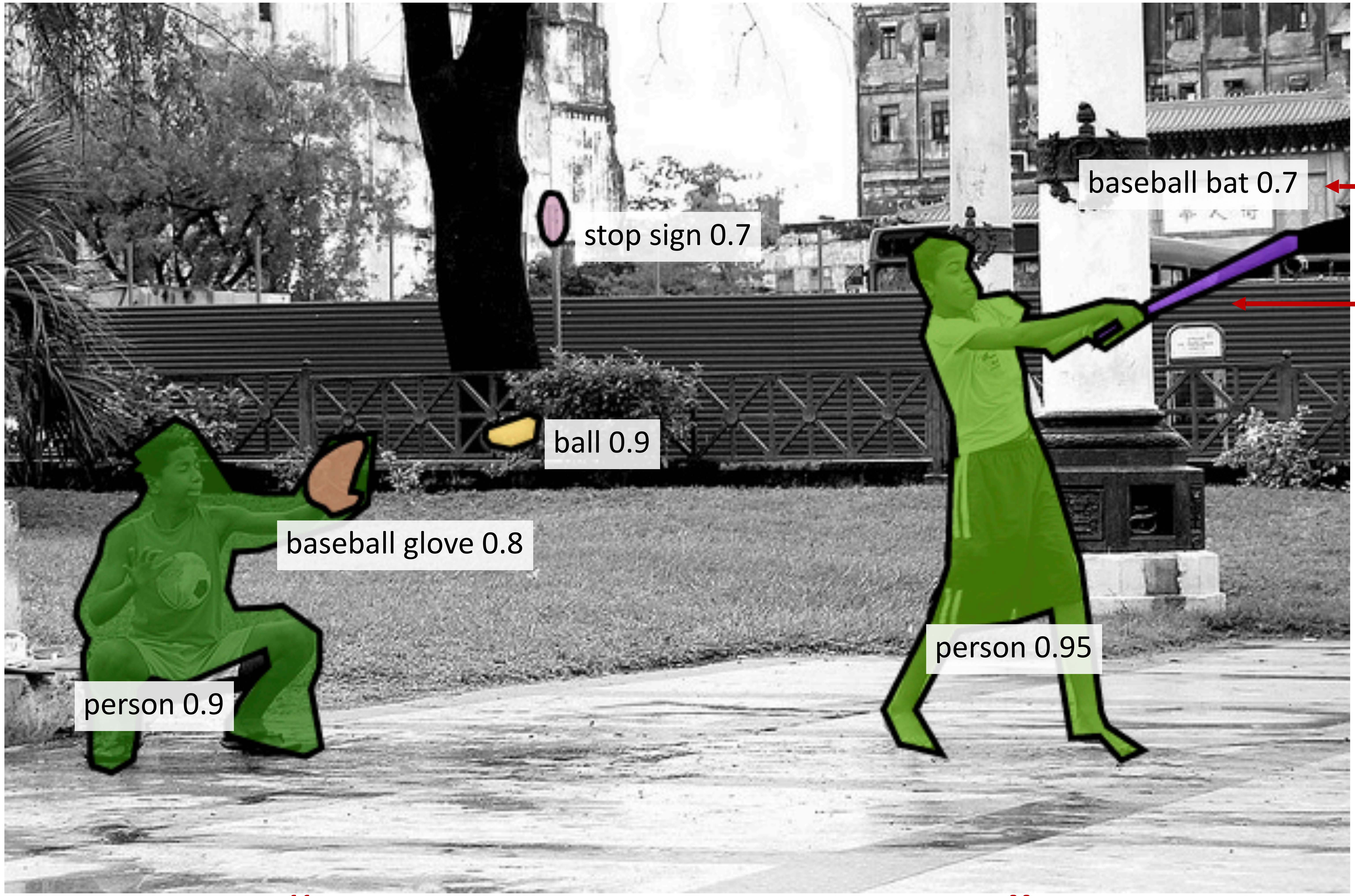


What?

Where?

"Object detection"

Object Detection with Segmentation Masks

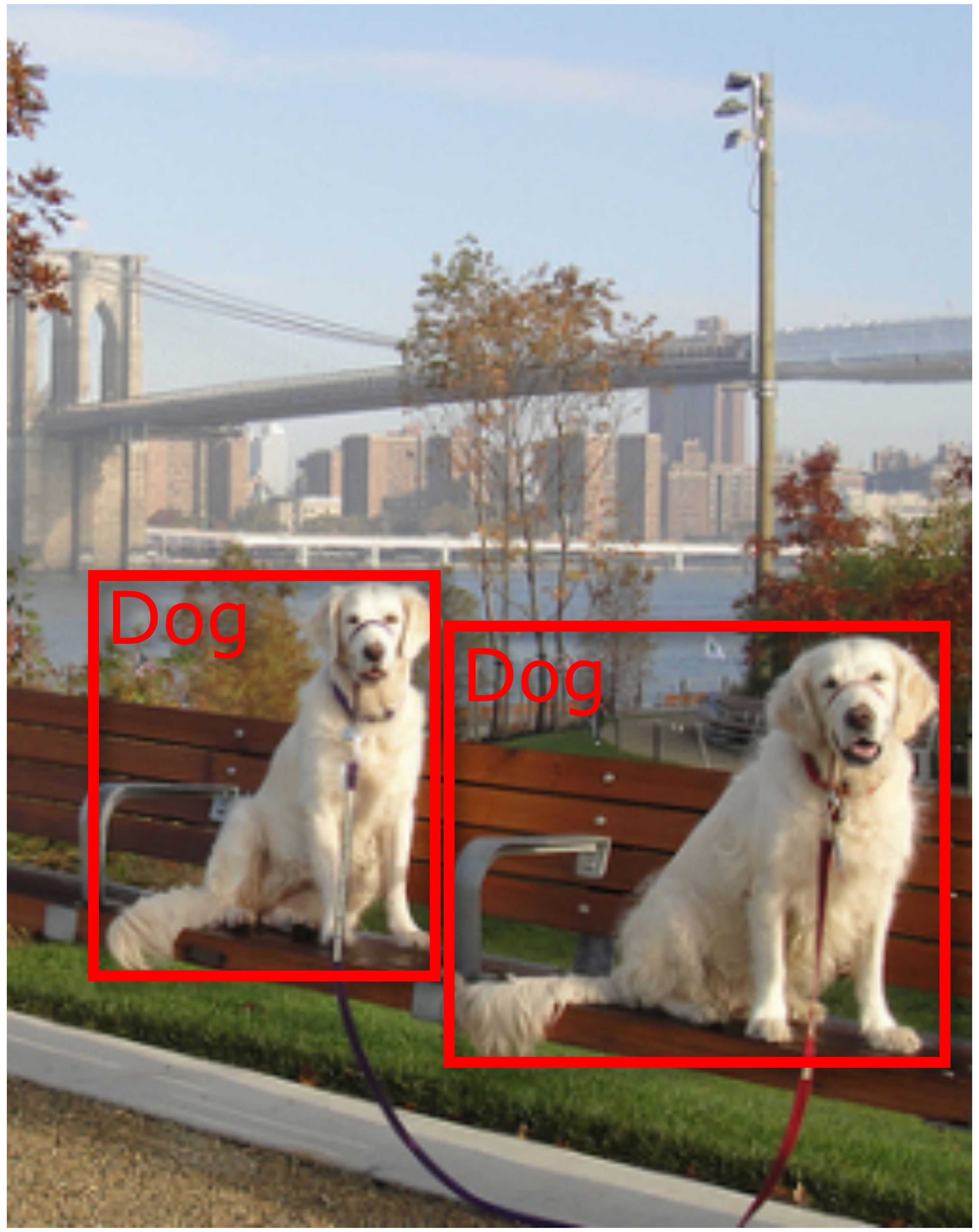


What?

Where?

"Instance segmentation"

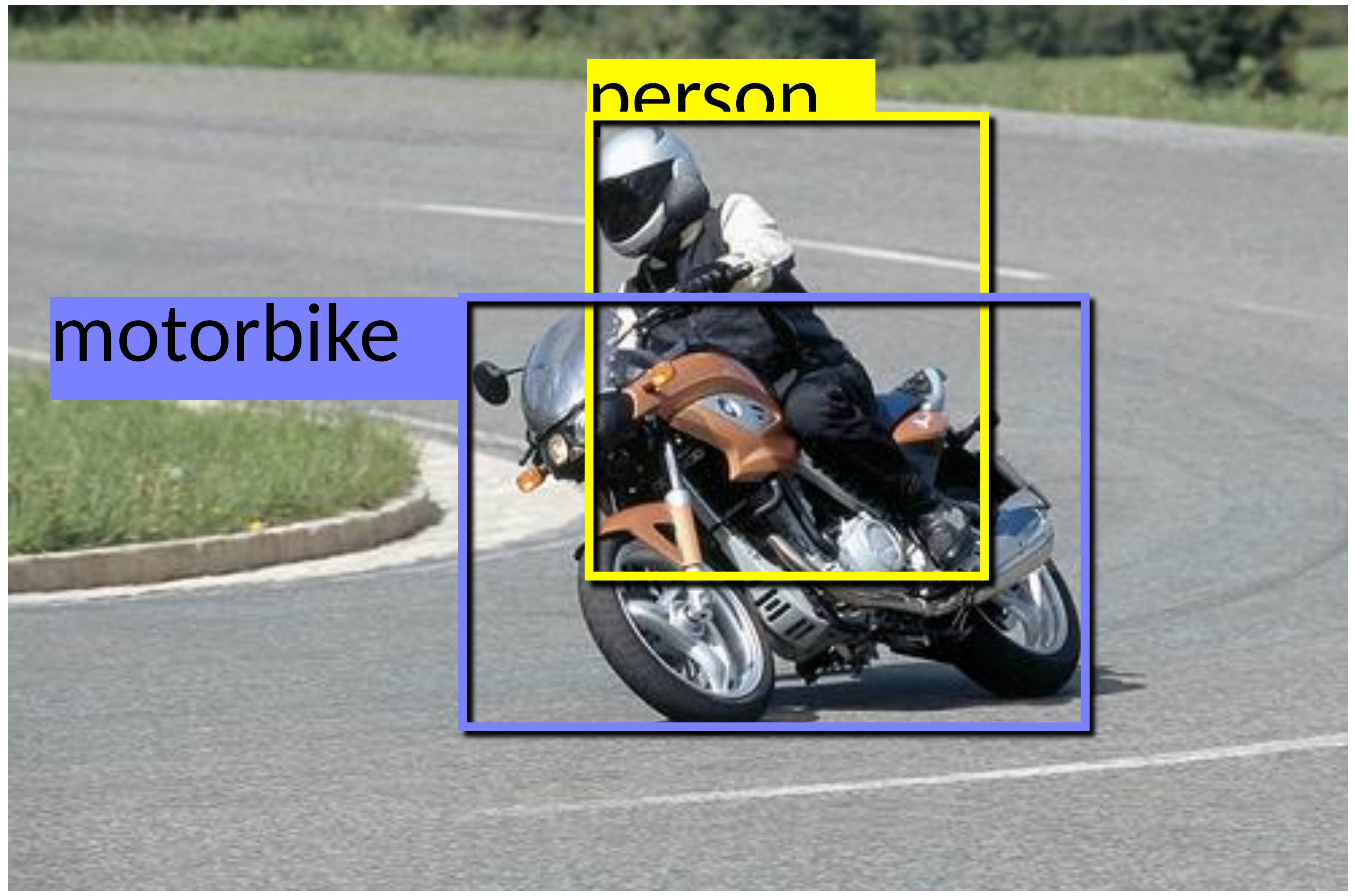
Classification vs. Detection



Problem formulation

{ airplane, bird, motorbike, person, sofa }

Input

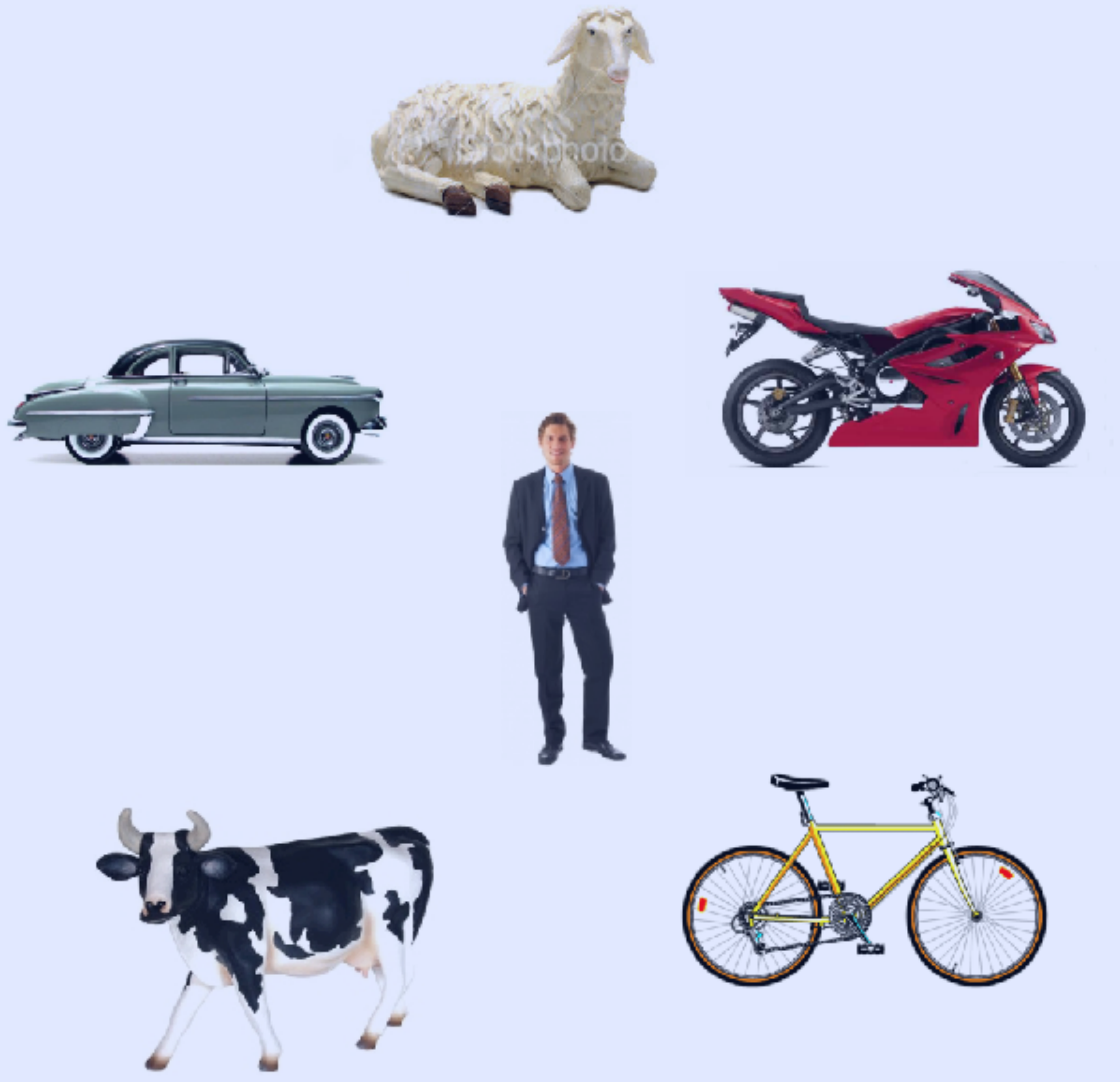


Desired output

Things vs. Stuff

Ted Adelson, Forsyth et al. 1996.

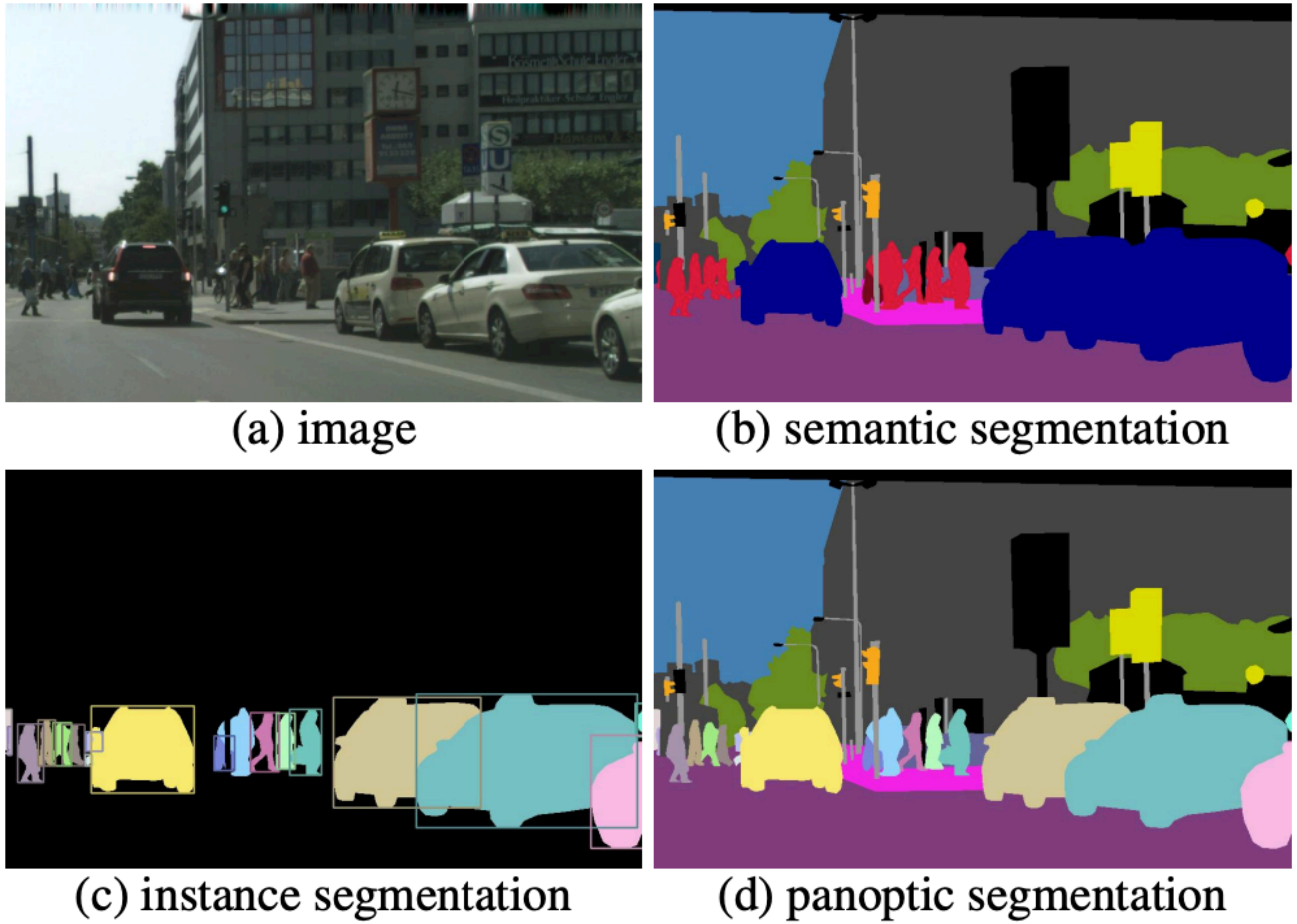
Thing (n): An object with a specific size and shape.



Stuff (n): Material defined by a homogeneous or repetitive pattern of fine-scale properties, but has no specific or distinctive spatial extent or shape.

Slide: Geremy Heitz

Panoptic segmentation

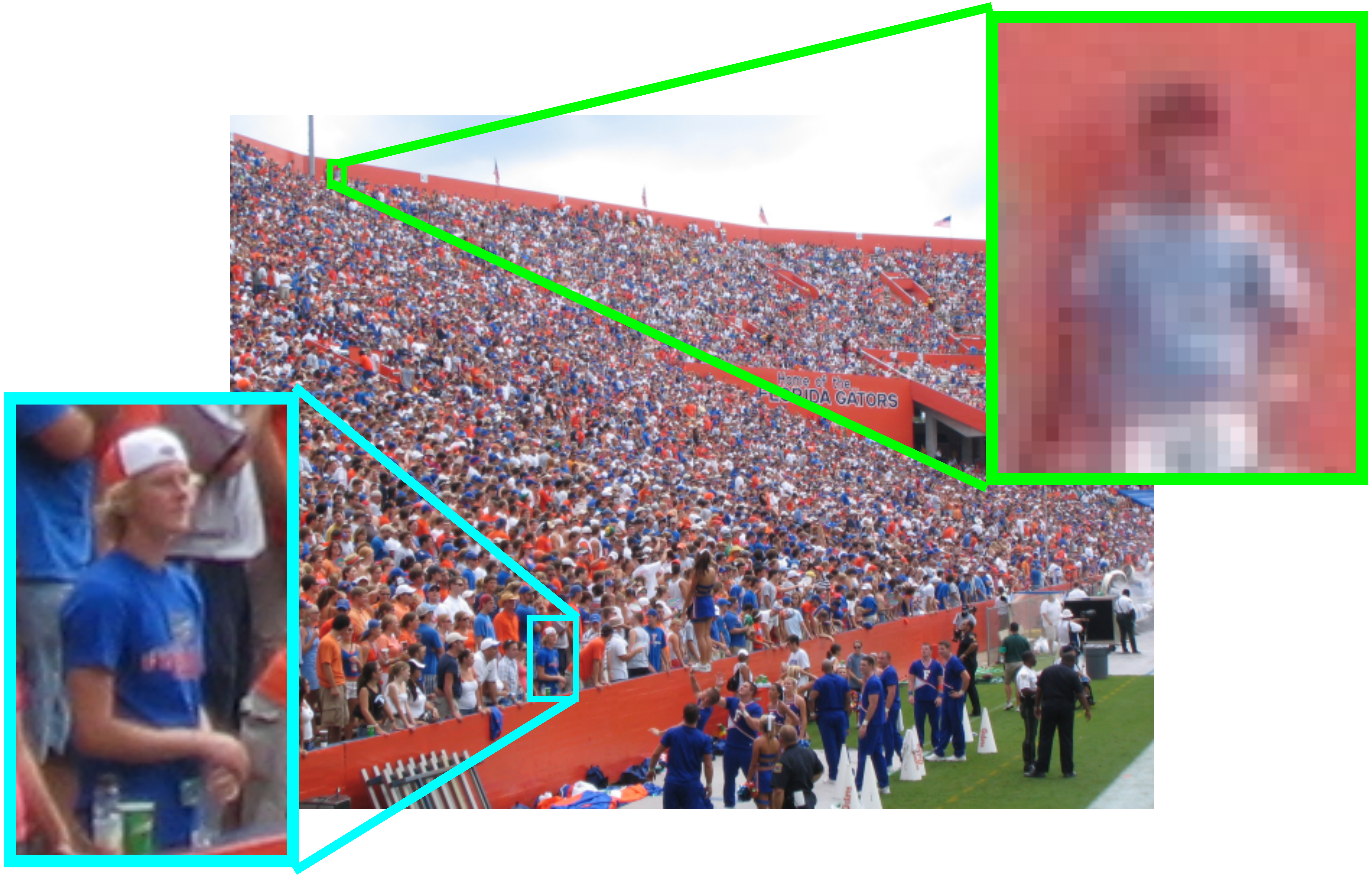


things – countable objects such as **people, animals, tools**

stuff – amorphous regions of similar texture or material such as **grass, sky, road**

Panoptic segmentation
[Kirillov, He, Girshick, Rother, Dollar CVPR 2019]

Challenges: Scale



Challenges: Occlusion and truncation

Challenges: Background Clutter

Challenges: Intra-class variation







Challenges: How to evaluate object detection?

Images may contain many objects and classes

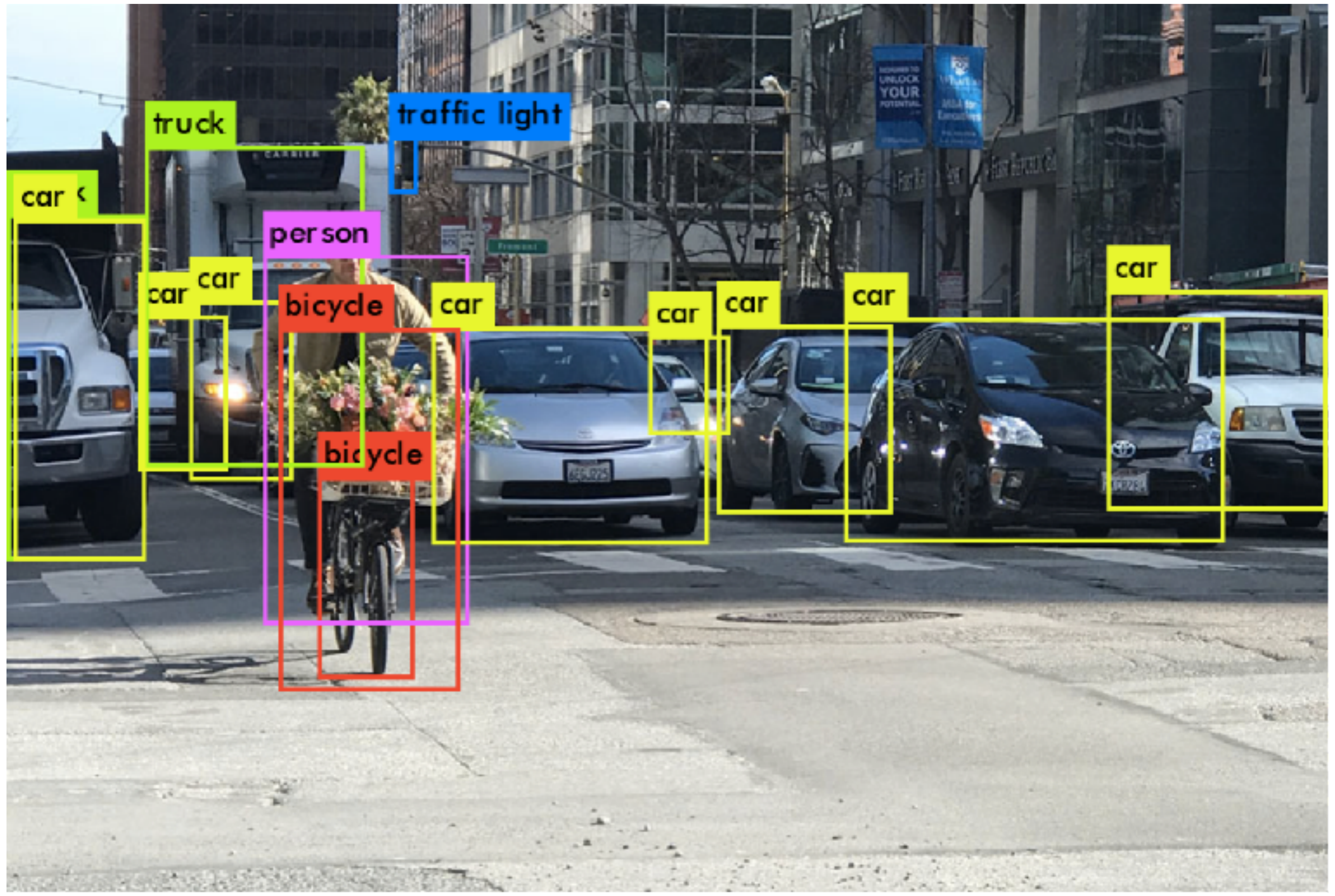
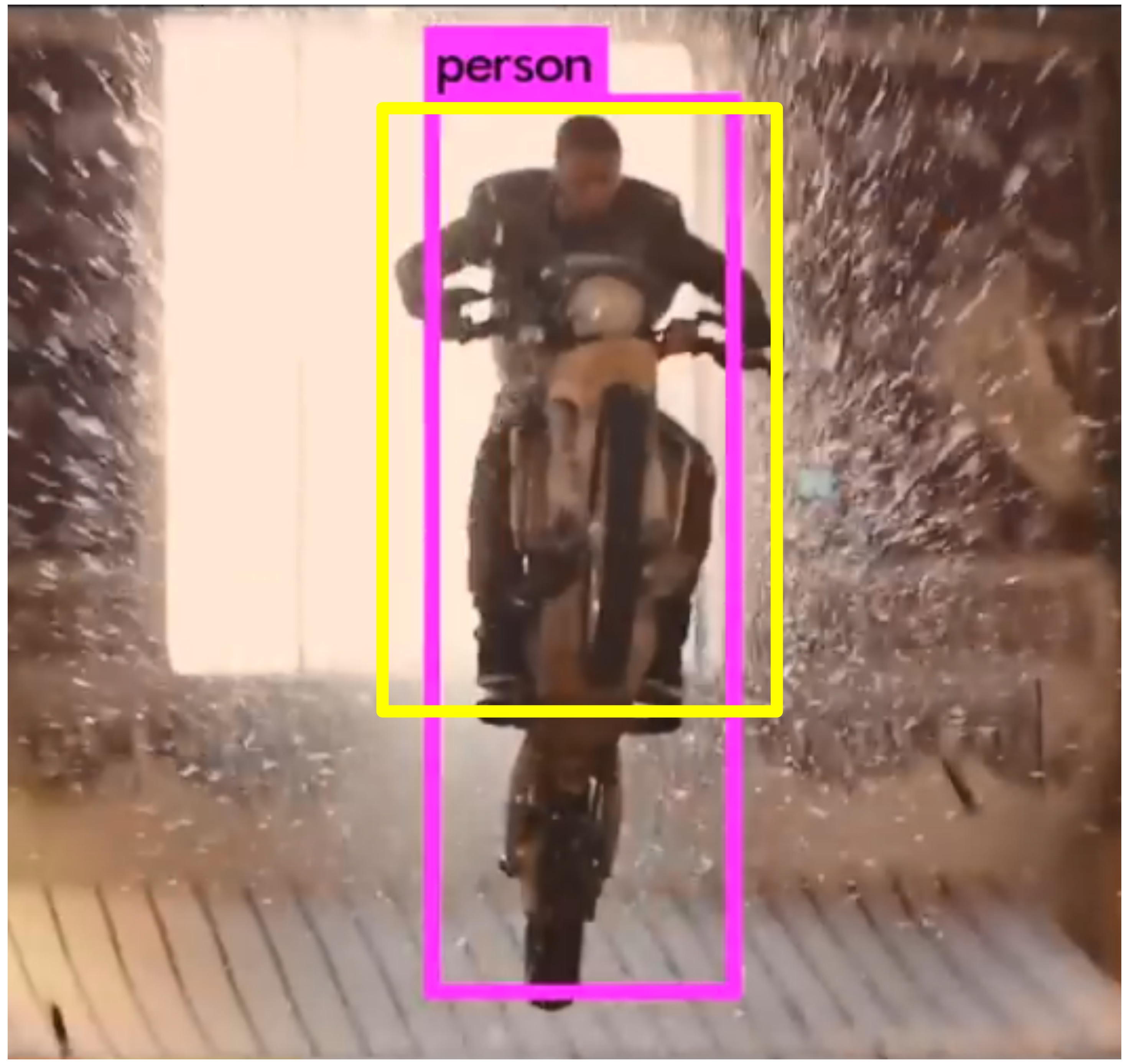


Image source

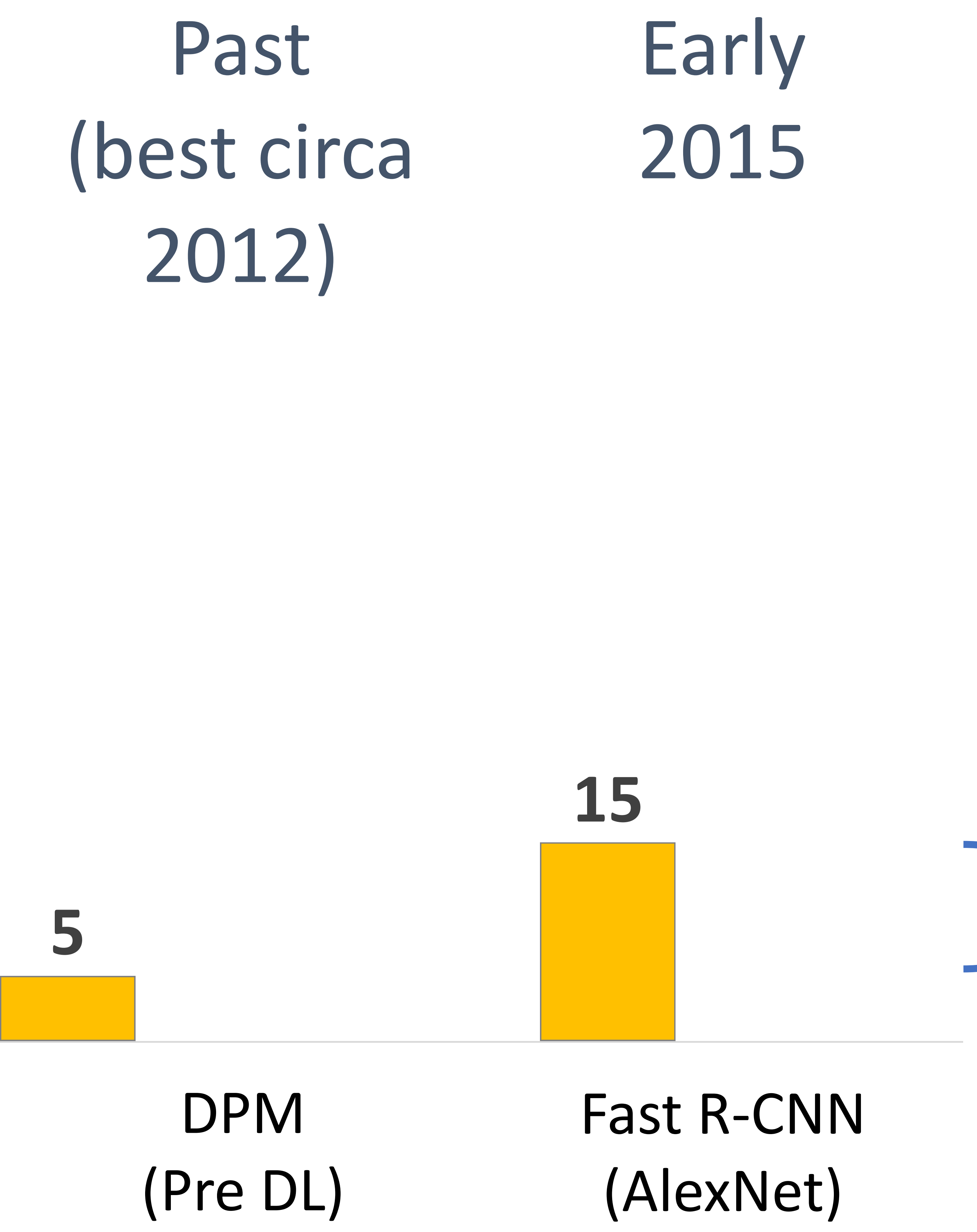
Localization results may not be precise



Ground truth

Detector output

COCO Object Detection Average Precision (%)



Movement to
Deep Learning methods:
3x improvement in AP

COCO Object Detection Average Precision (%)

Steady Progress on Boxes and Masks

- ➢ **R-CNN** [Girshick et al. 2014]
- ➢ SPP-net [He et al. 2014]
- ➢ **Fast R-CNN** [Girshick. 2015]
- ➢ **Faster R-CNN** [Ren et al. 2015]
- ➢ R-FCN [Dai et al. 2016]
- ➢ Feature Pyramid Networks + Faster R-CNN [Lin et al. 2017]
- ➢ Mask R-CNN [He et al. 2017]
- ➢ Training with Large Minibatches (MegDet) [Peng, Xiao, Li, et al. 2017]
- ➢ Cascade R-CNN [Cai & Vasconcelos 2018]
- ➢ …

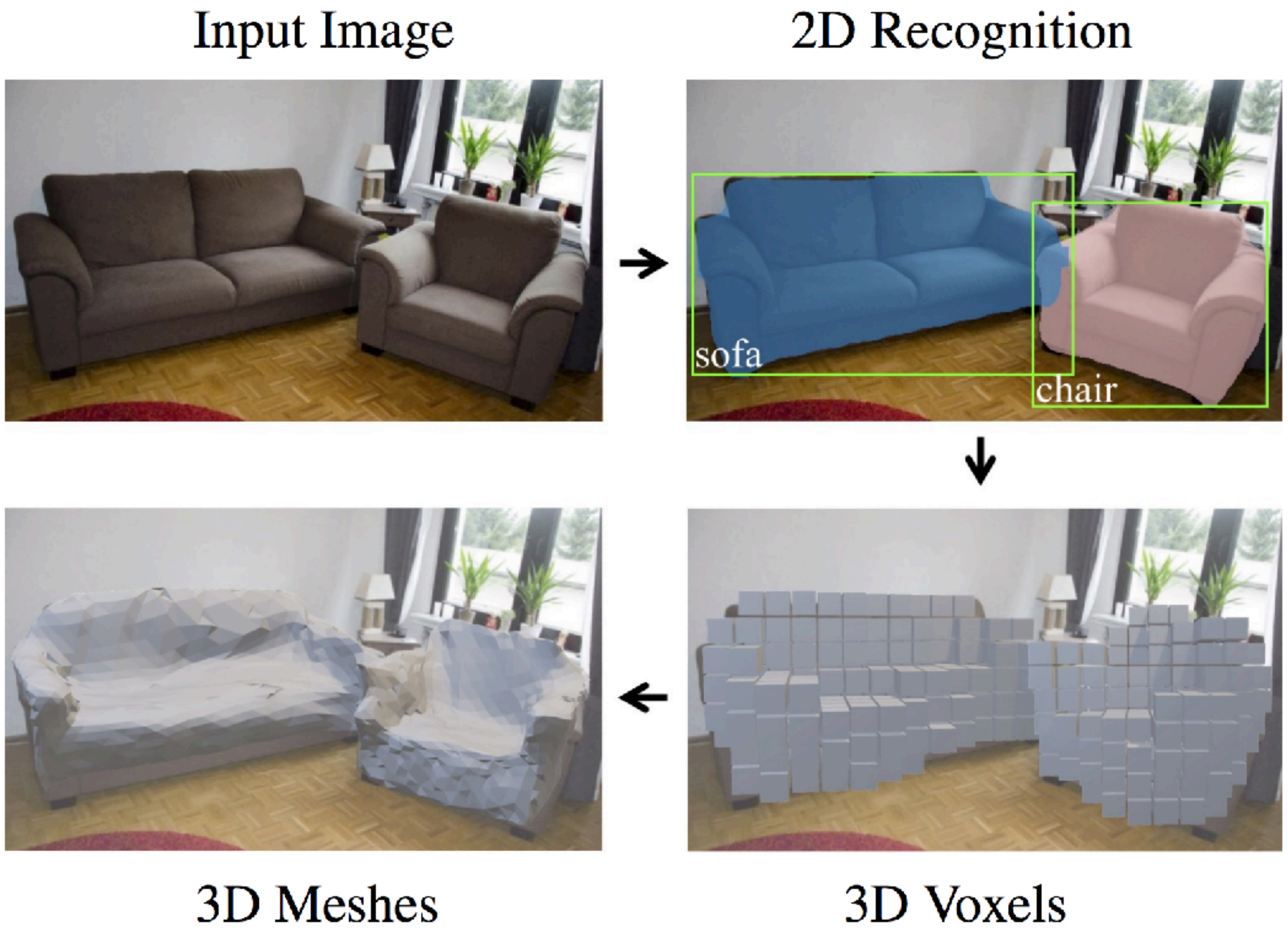
Beyond Boxes and Masks: Human Keypoints

COCO Keypoint Detection Task
[COCO team @ cocodataset.org 2016 - present]

Beyond Boxes and Masks: Human Surfaces

DensePose: Dense Human Pose Estimation In The Wild
[Güler, Neverova, Kokkinos CVPR 2018]

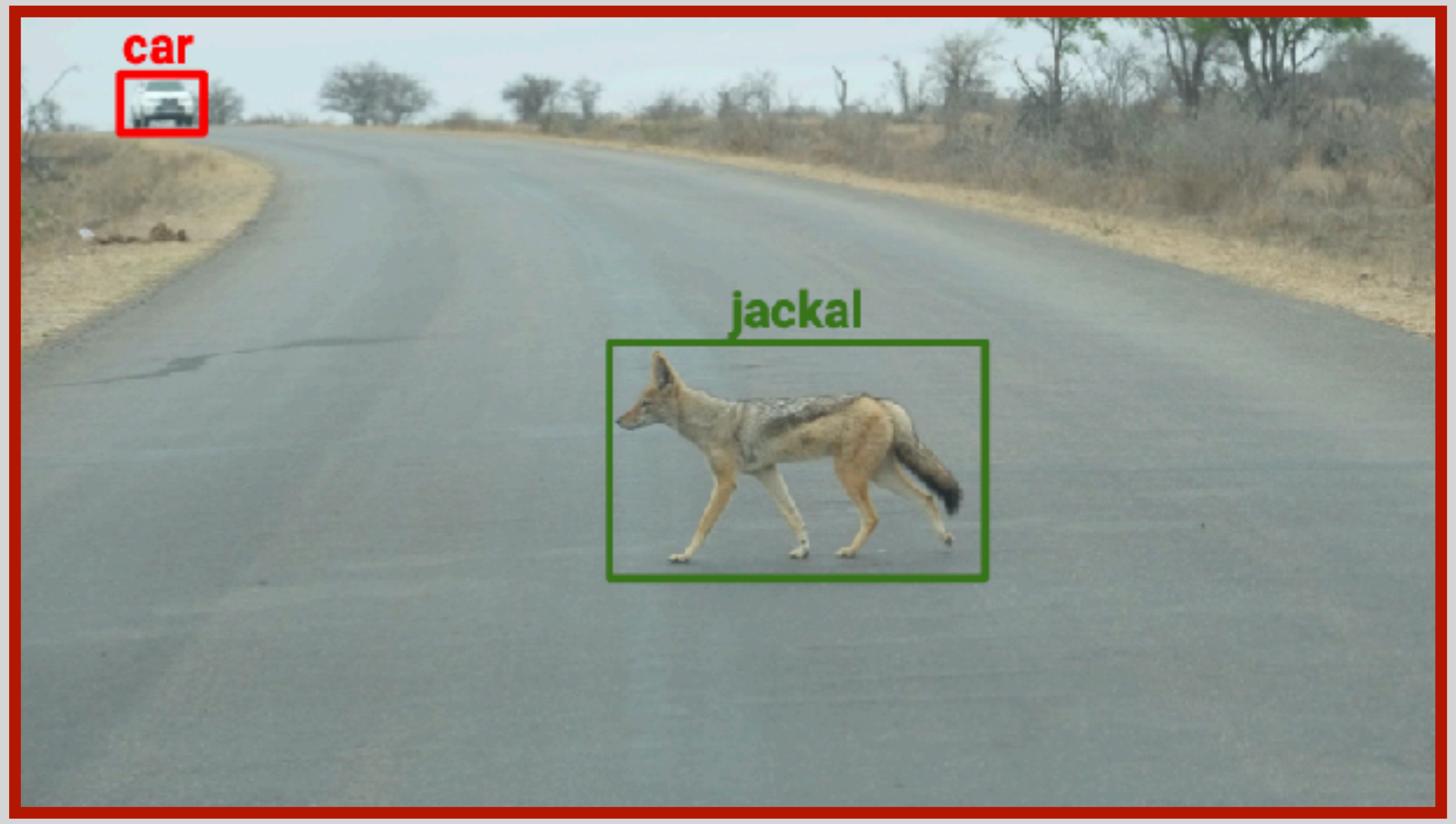
Beyond Boxes and Masks: 3D Shape



Mesh R-CNN
[Gkioxari, Malik, Johnson ICCV 2019]

Agenda

- 0. Intro to structured outputs
- 1. Object detection (localization)
- 2. Segmentation
- 3. Human pose estimation



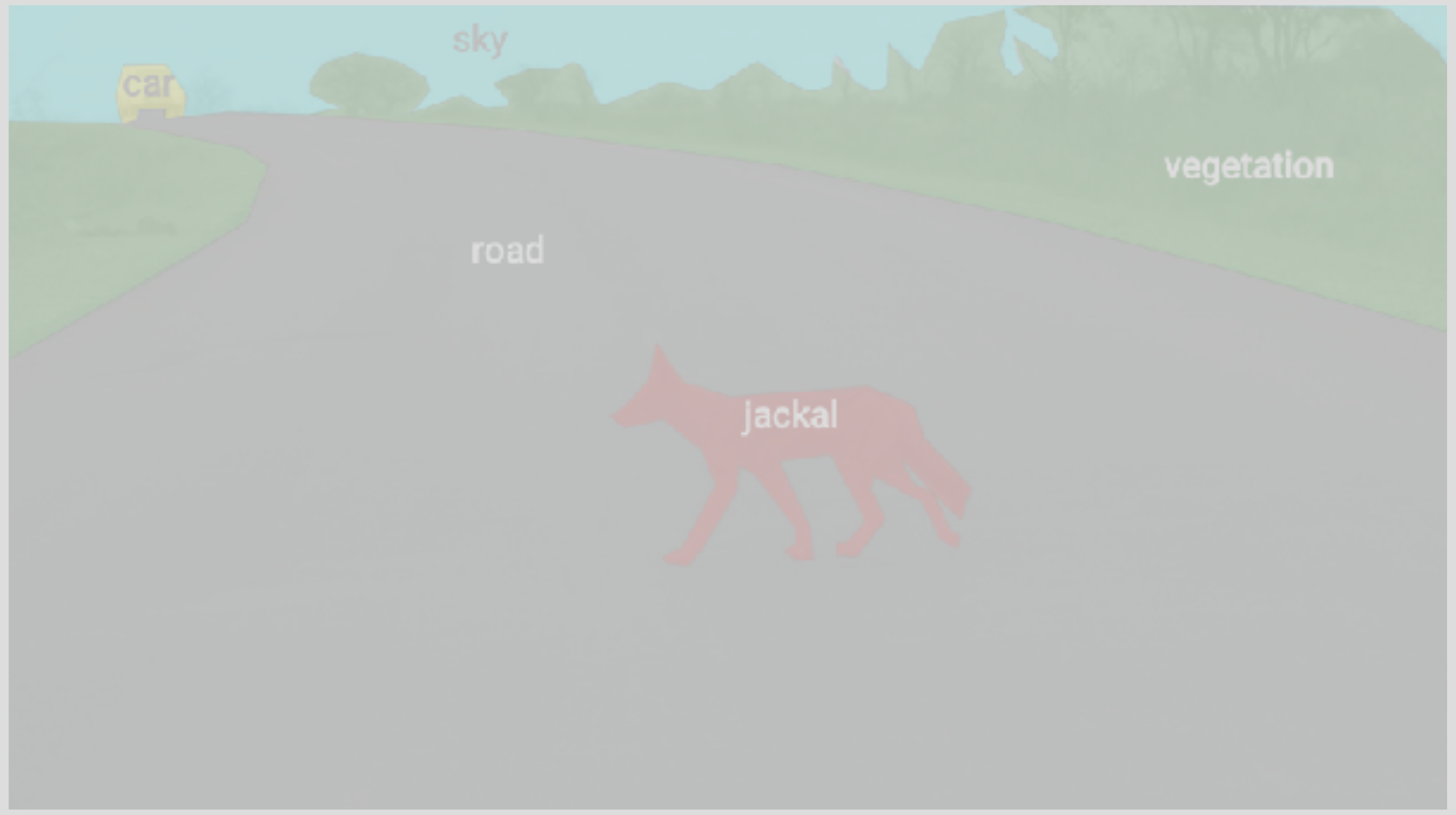
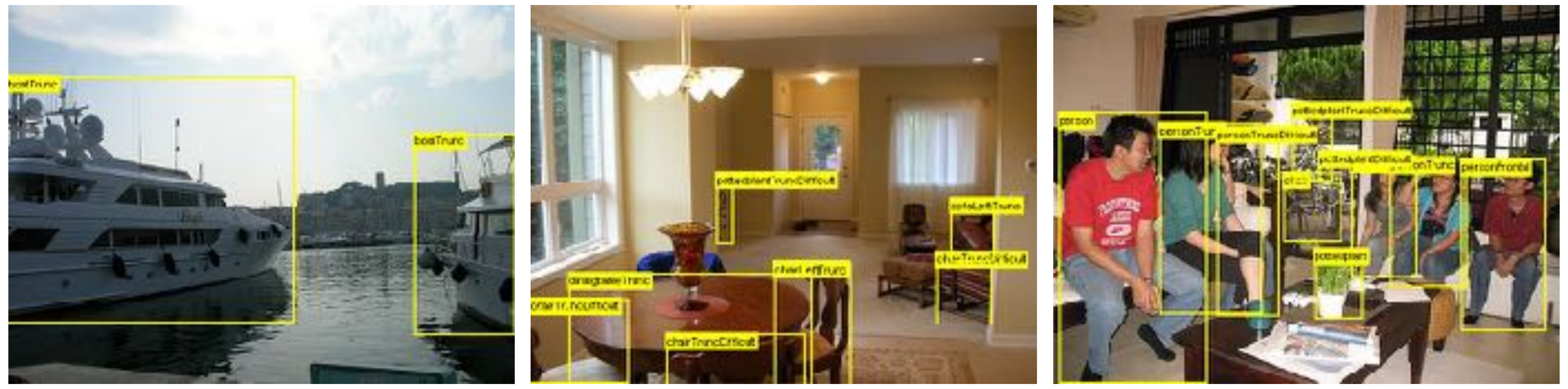


Image credits: Naila Murray

Object detection datasets (benchmarks)

Datasets	Categories	Images	Bounding Boxes
PASCAL-VOC	20	11K	27K
COCO	80 (91 stuff)	328K	2500K
LVIS	1200	164K	2.2M



PASCAL-VOC [2005-2008]

COCO [2014-2015]

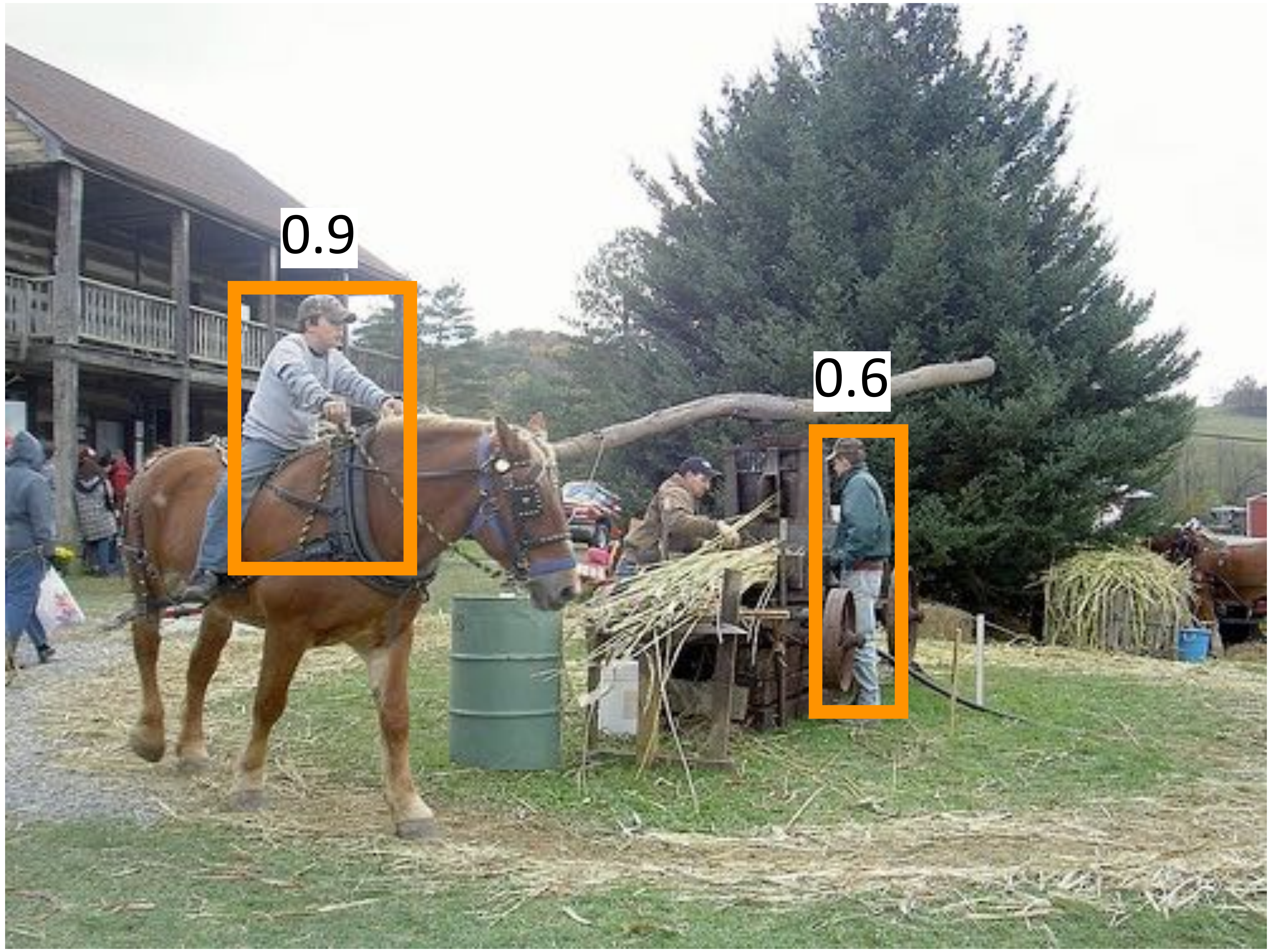
Evaluating a detector

Test image (previously unseen)

First detection ...

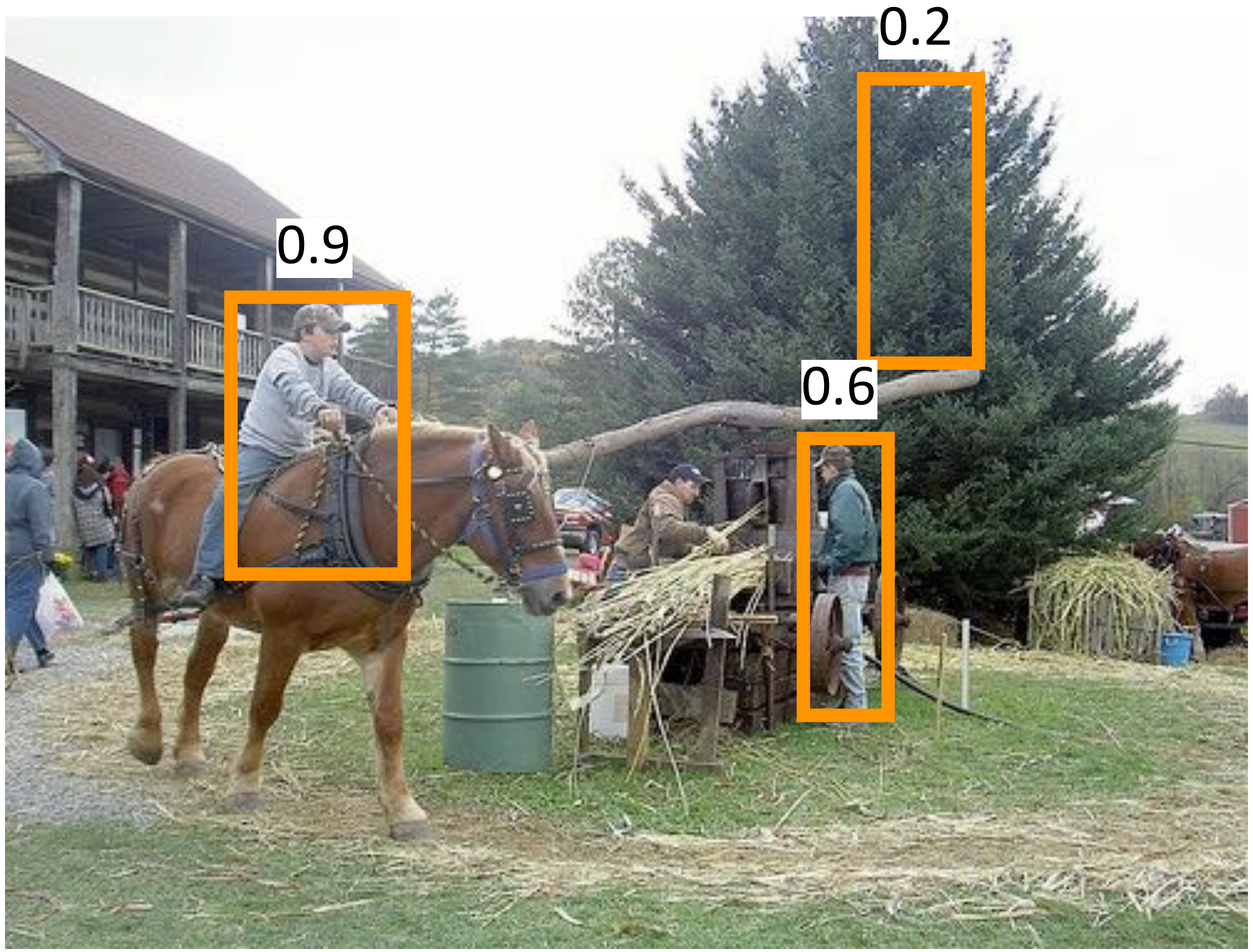
‘person’ detector predictions

Second detection ...



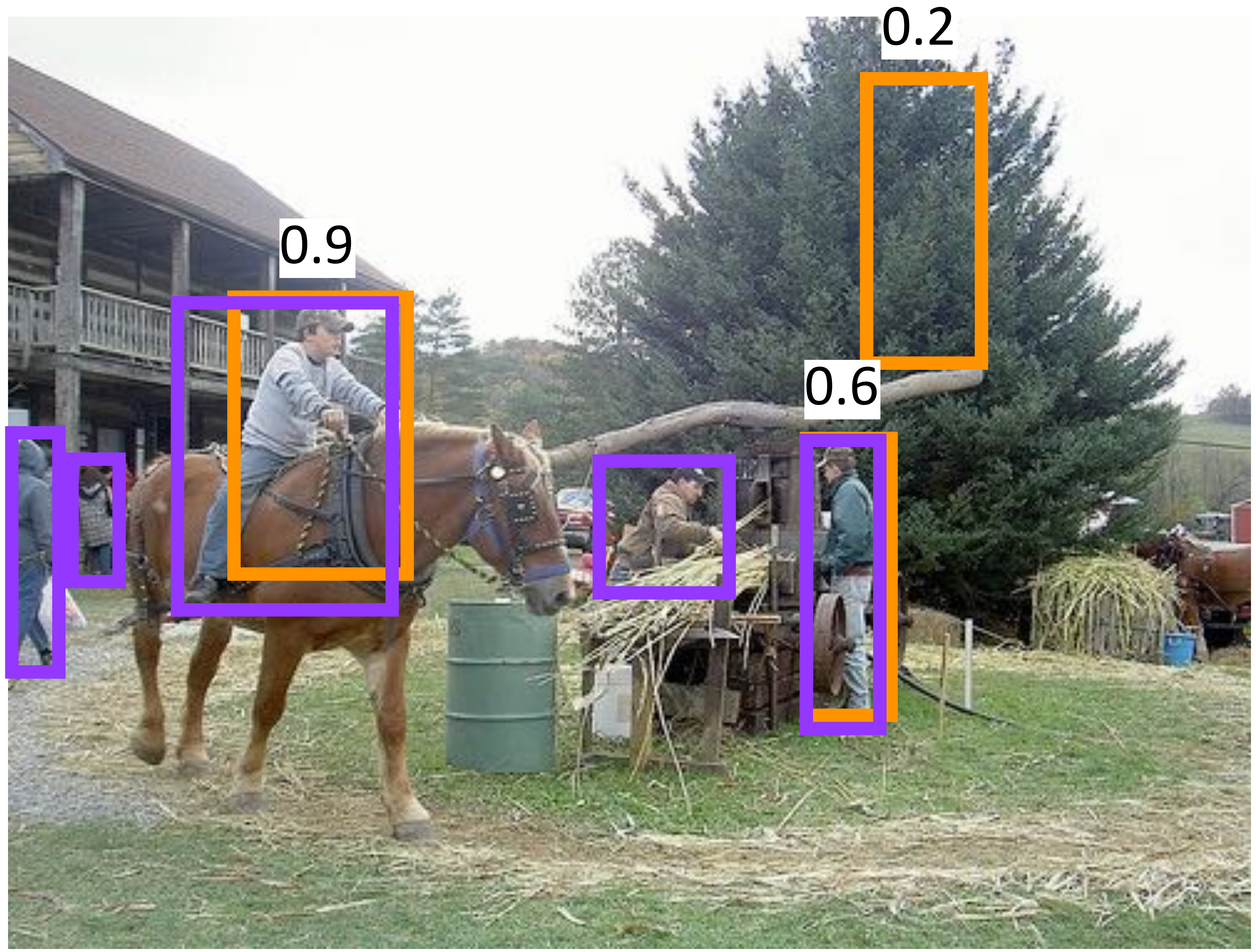
'person' detector predictions

Third detection ...



'person' detector predictions

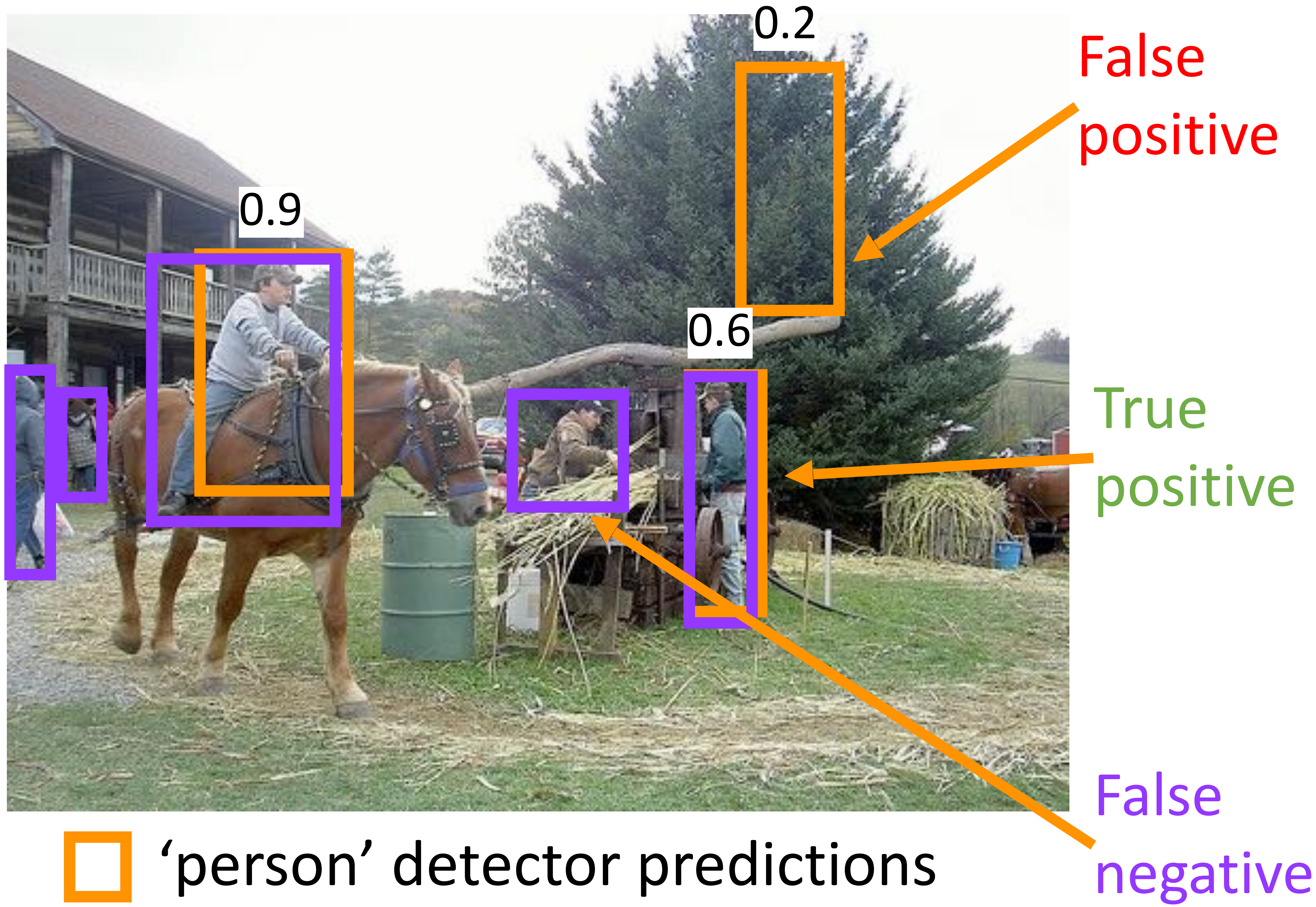
Compare to ground truth



‘person’ detector predictions

ground truth ‘person’ boxes

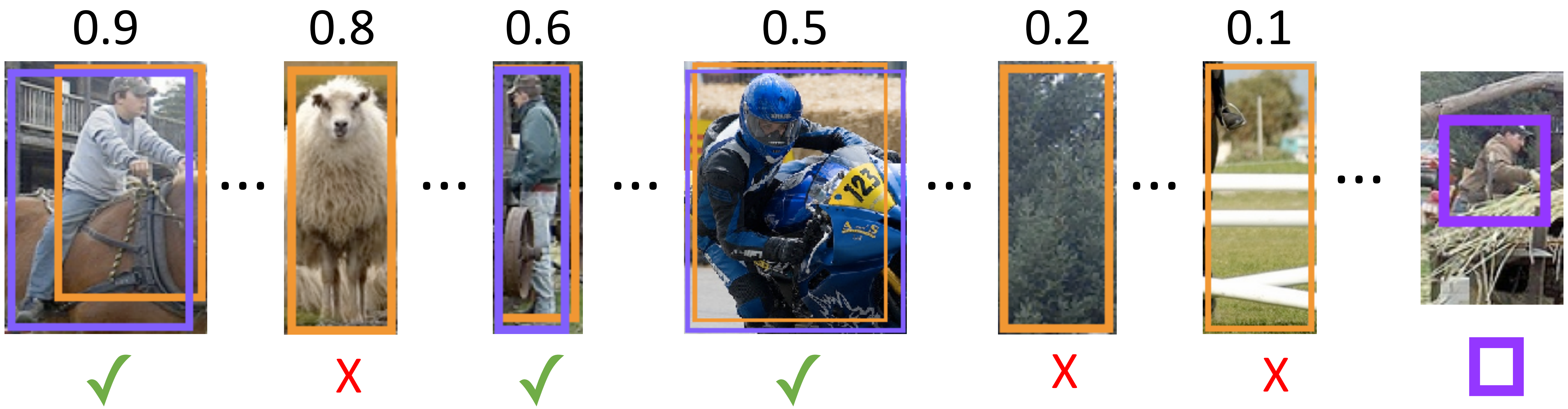
Compare to ground truth



'person' detector predictions

ground truth 'person' boxes

Sort by confidence

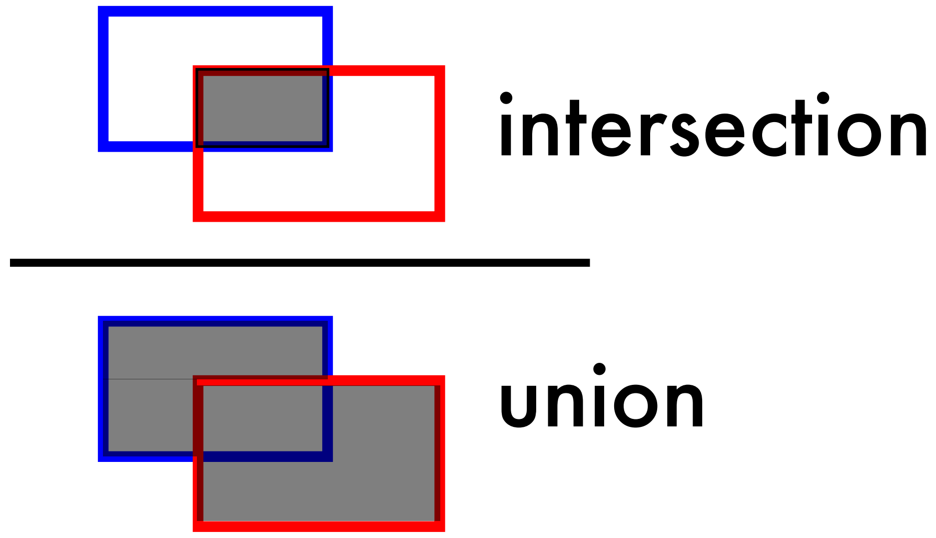


true positive
(high overlap)
i.e., IOU > threshold

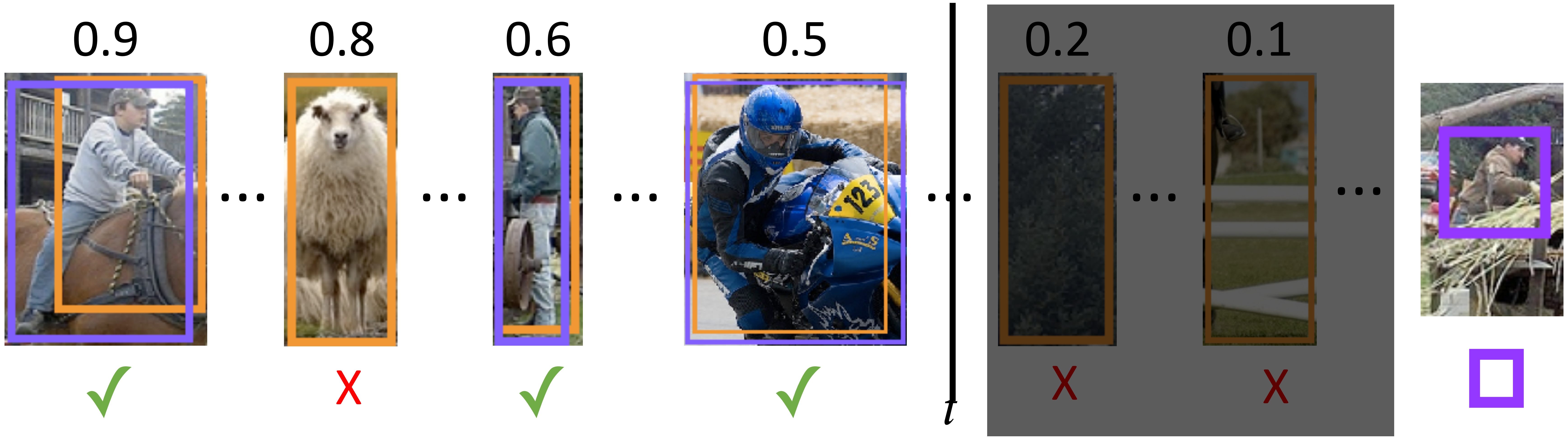
false positive
(no overlap,
low overlap,
or duplicate)

false negative
(missing
detection)

IOU (intersection over union):



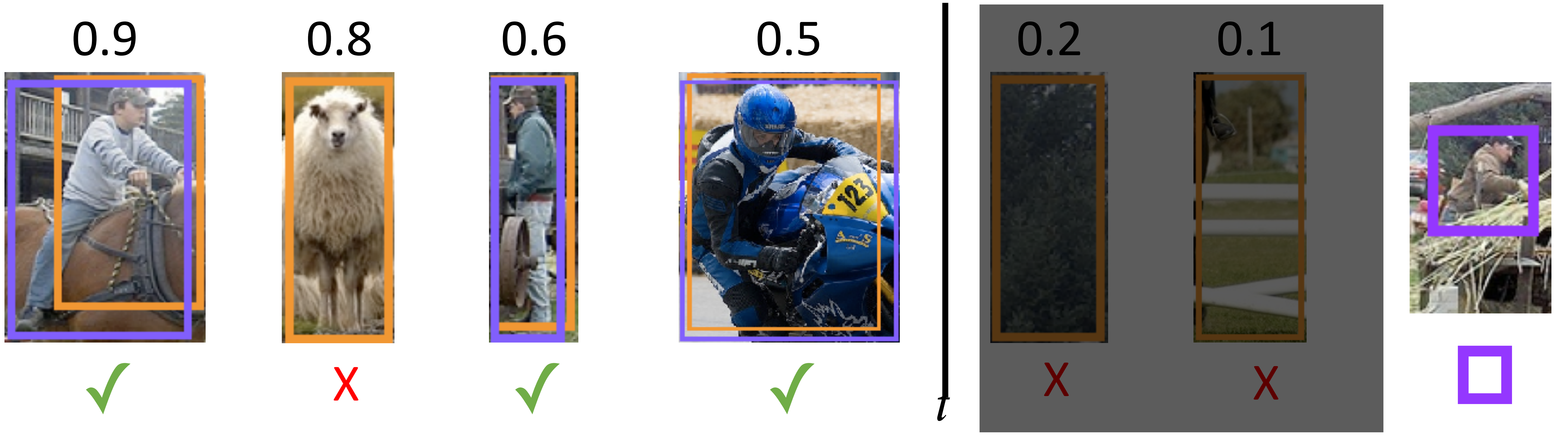
Sort by confidence



$$precision@t = \frac{\#true\ positives@t}{\#true\ positives@t + \#false\ positives@t} \qquad \frac{\checkmark}{\checkmark + X}$$

$$recall@t = \frac{\#true\ positives@t}{\#ground\ truth\ objects} \qquad \frac{\checkmark}{\square}$$

Sort by confidence



$t = 0.5$

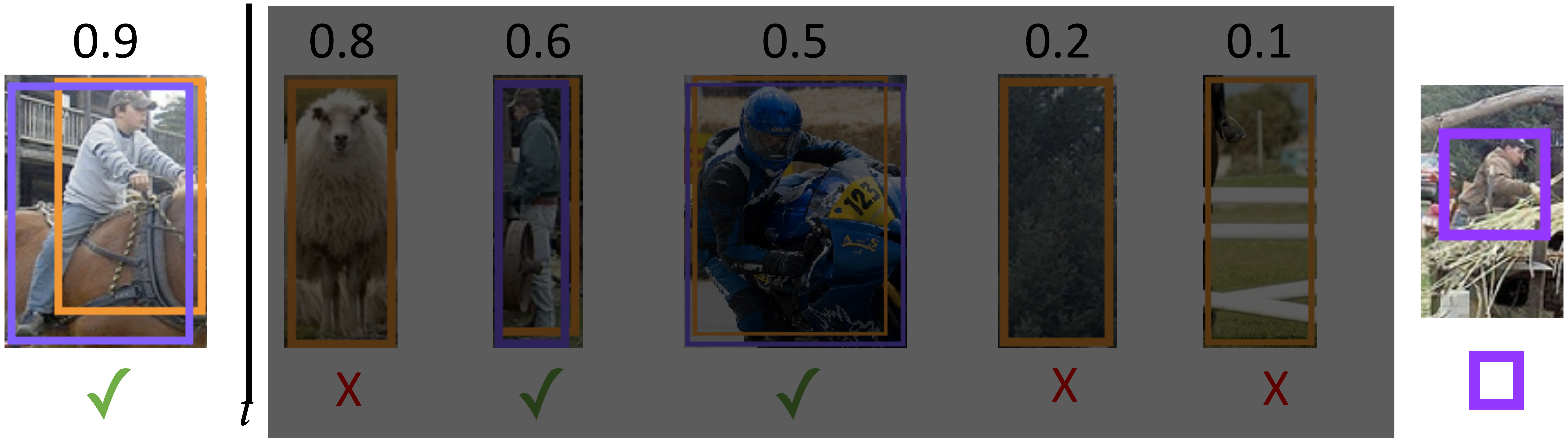
$$precision@t = \frac{\#true\ positives@t}{\#true\ positives@t + \#false\ positives@t}$$

$$\frac{✓}{✓ + X} = 75\%$$

$$recall@t = \frac{\#true\ positives@t}{\#ground\ truth\ objects}$$

$$\frac{✓}{□} = 75\%$$

Sort by confidence

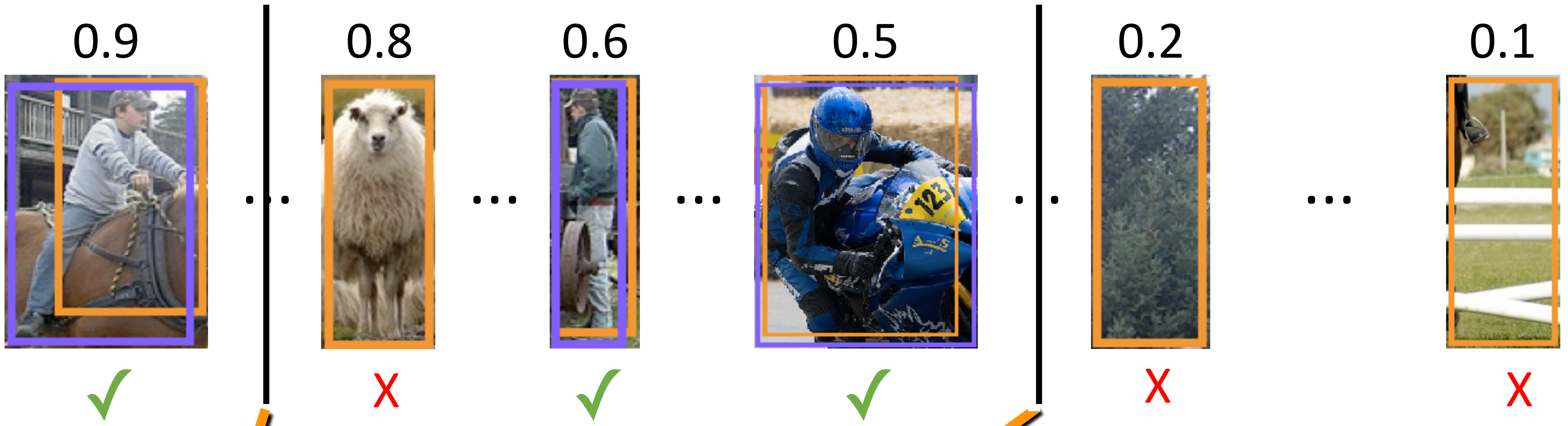


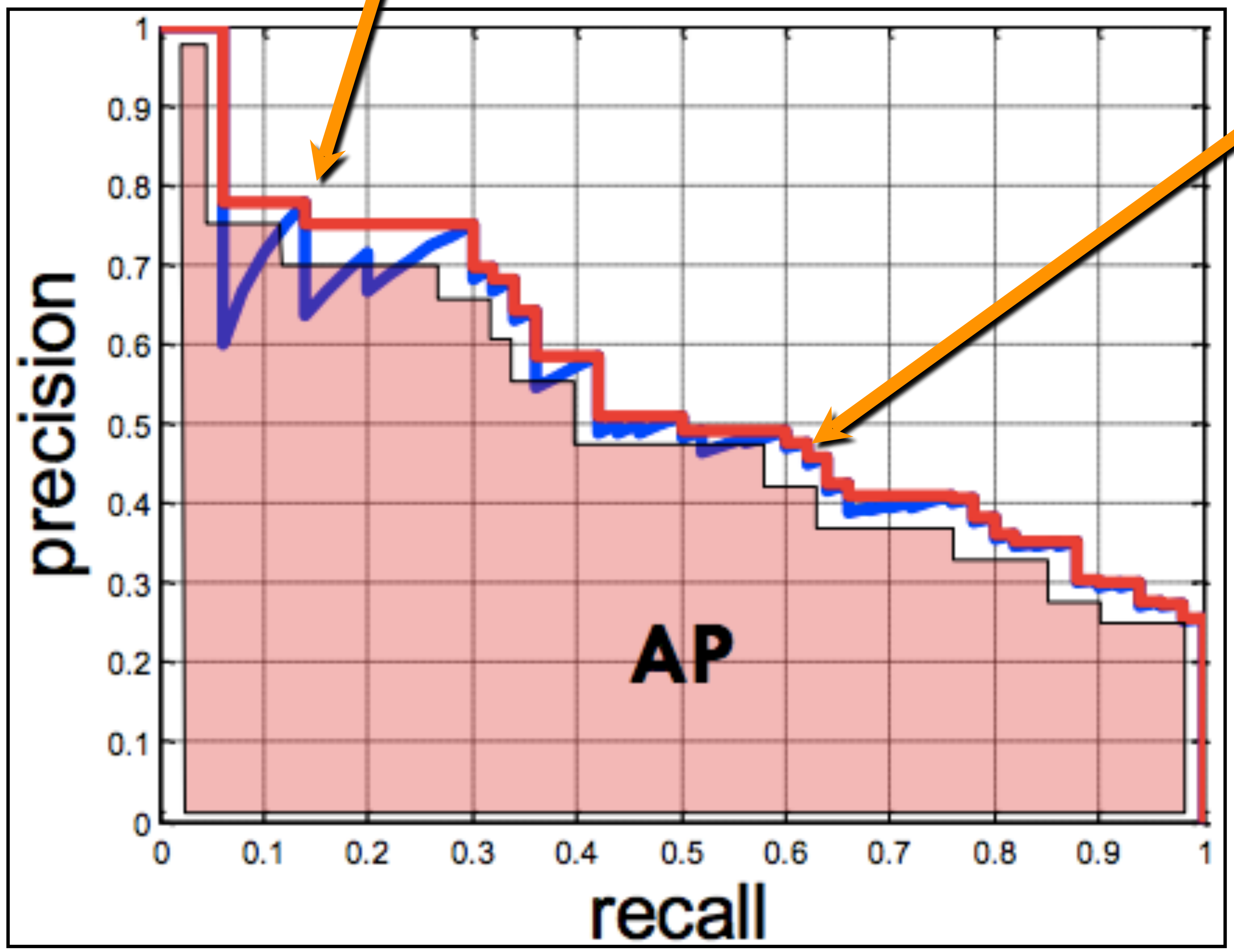
$t = 0.9$

$$precision@t = \frac{\#true\ positives@t}{\#true\ positives@t + \#false\ positives@t} \qquad \frac{\checkmark}{\checkmark + \text{X}} = 100\%$$

$$recall@t = \frac{\#true\ positives@t}{\#ground\ truth\ objects} \qquad \frac{\checkmark}{\square} = 25\%$$

Average Precision for a (class, IOU threshold) pair





Average Precision (AP)

0% is worst

100% is best

mean AP over classes (mAP)

AP(class, threshold): area under PR curve

Average Precision for a class

$$AP(class) = \frac{1}{\#threshods} \sum_{iou \in threshold} AP(class, iou)$$

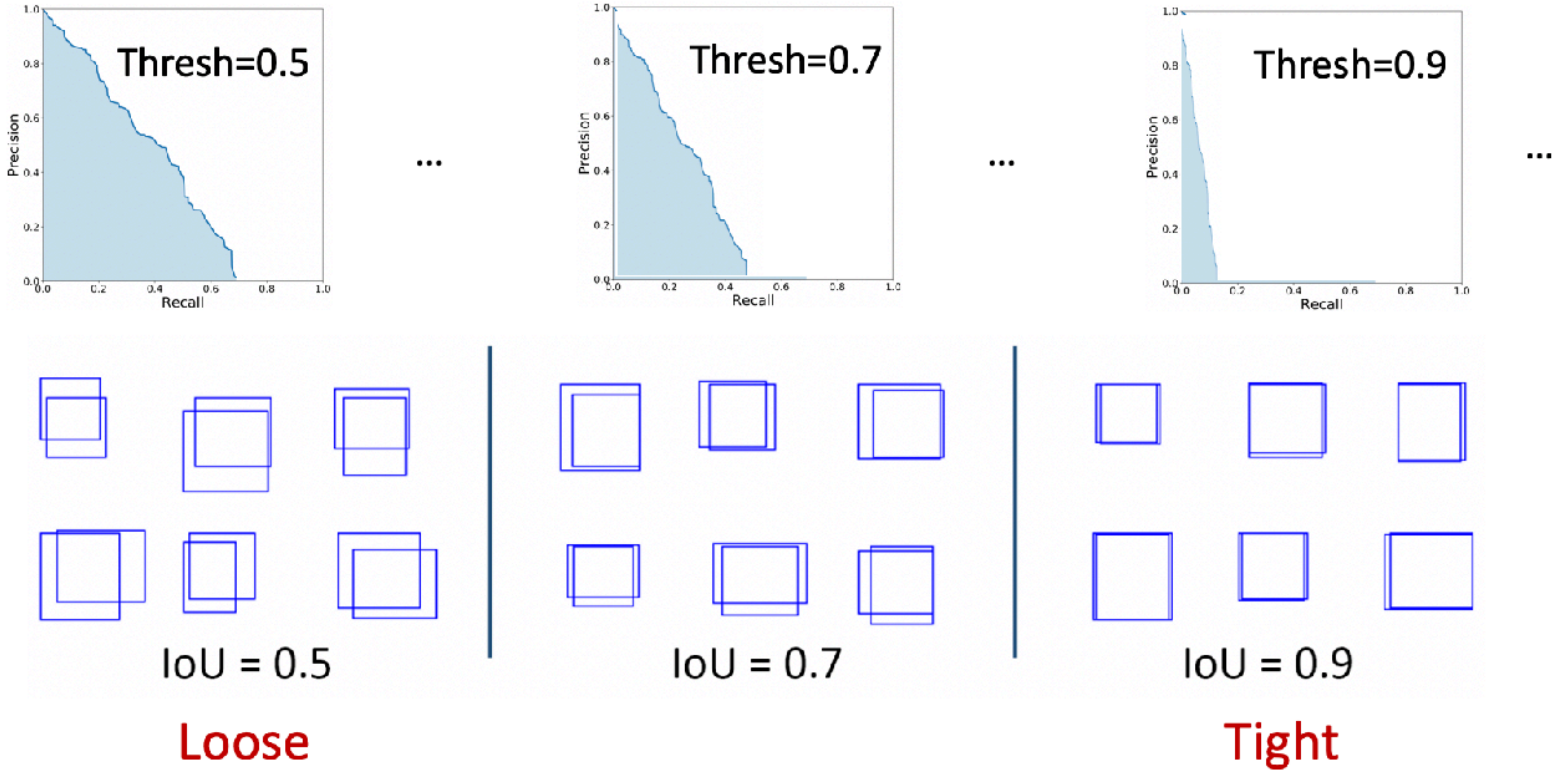


Figure credits: Dollár and Zitnick

Slides modified from Ross Girshick tutorial at CVPR 2019

Overall **Average Precision** (%)

$$AP = \frac{1}{\#classes} \sum_{class \in classes} AP(class)$$

"AP" is really an average, average, average precision.

classes iou thresholds precision @ different recall levels

Average Precision (AP):

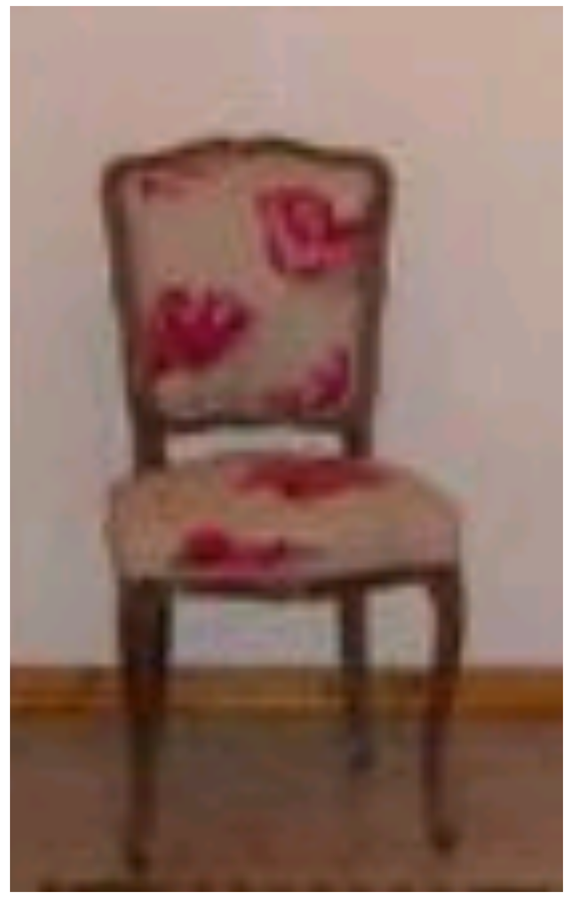
AP	% AP at IoU=.50:.05:.95 **(primary challenge metric)**
$AP^{IoU=.50}$	% AP at IoU=.50 (PASCAL VOC metric)
$AP^{IoU=.75}$	% AP at IoU=.75 (strict metric)

AP Across Scales:

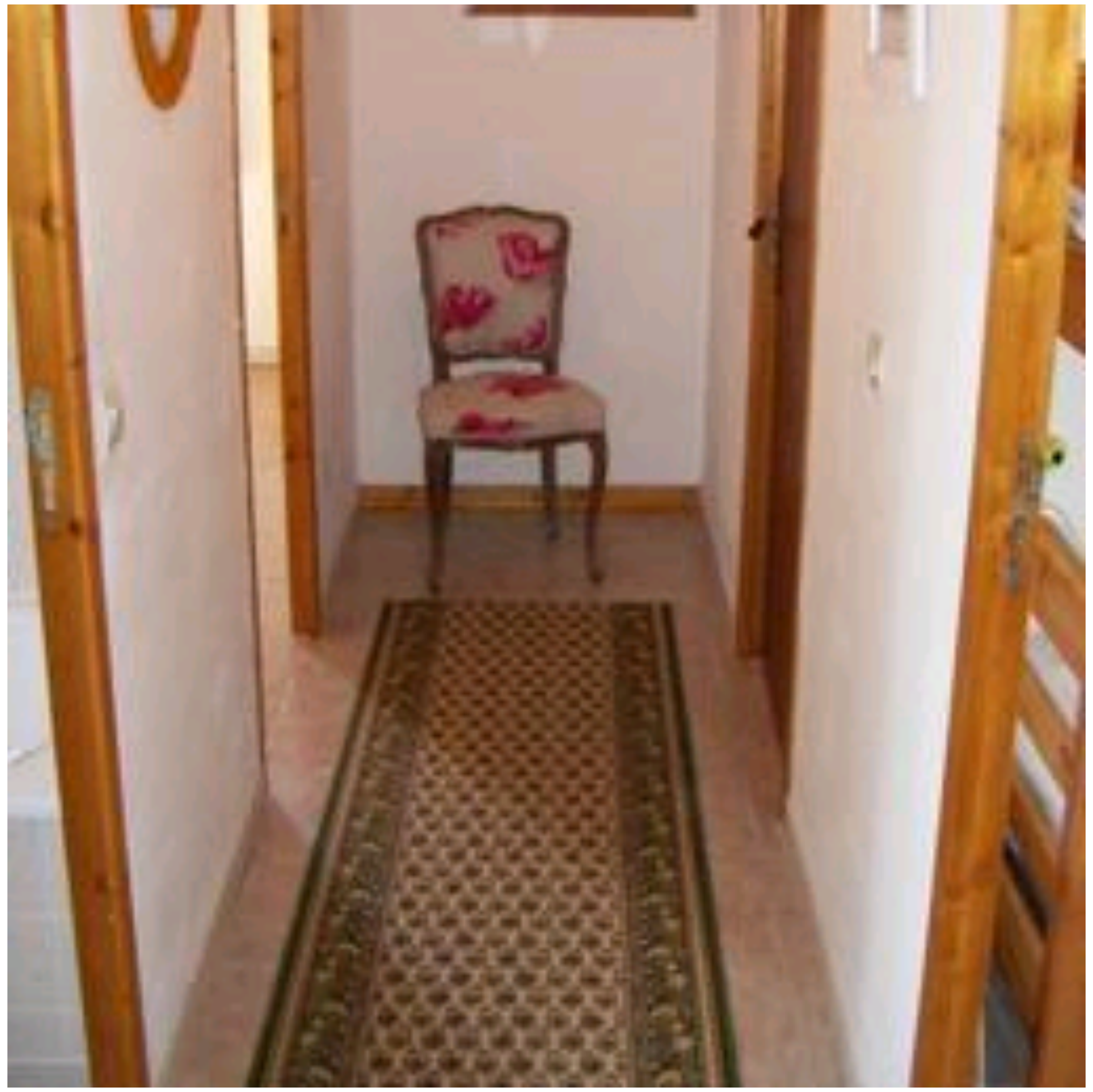
AP^{small}	% AP for small objects: area < 32^2
AP^{medium}	% AP for medium objects: 32^2 < area < 96^2
AP^{large}	% AP for large objects: area > 96^2

Object detection: naive attempt

This is a chair



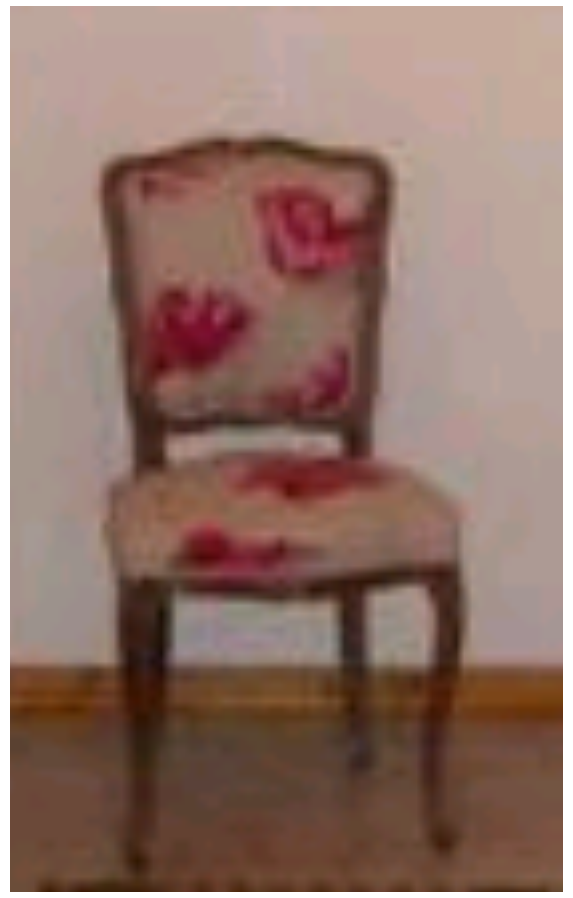
Find the chair in this image



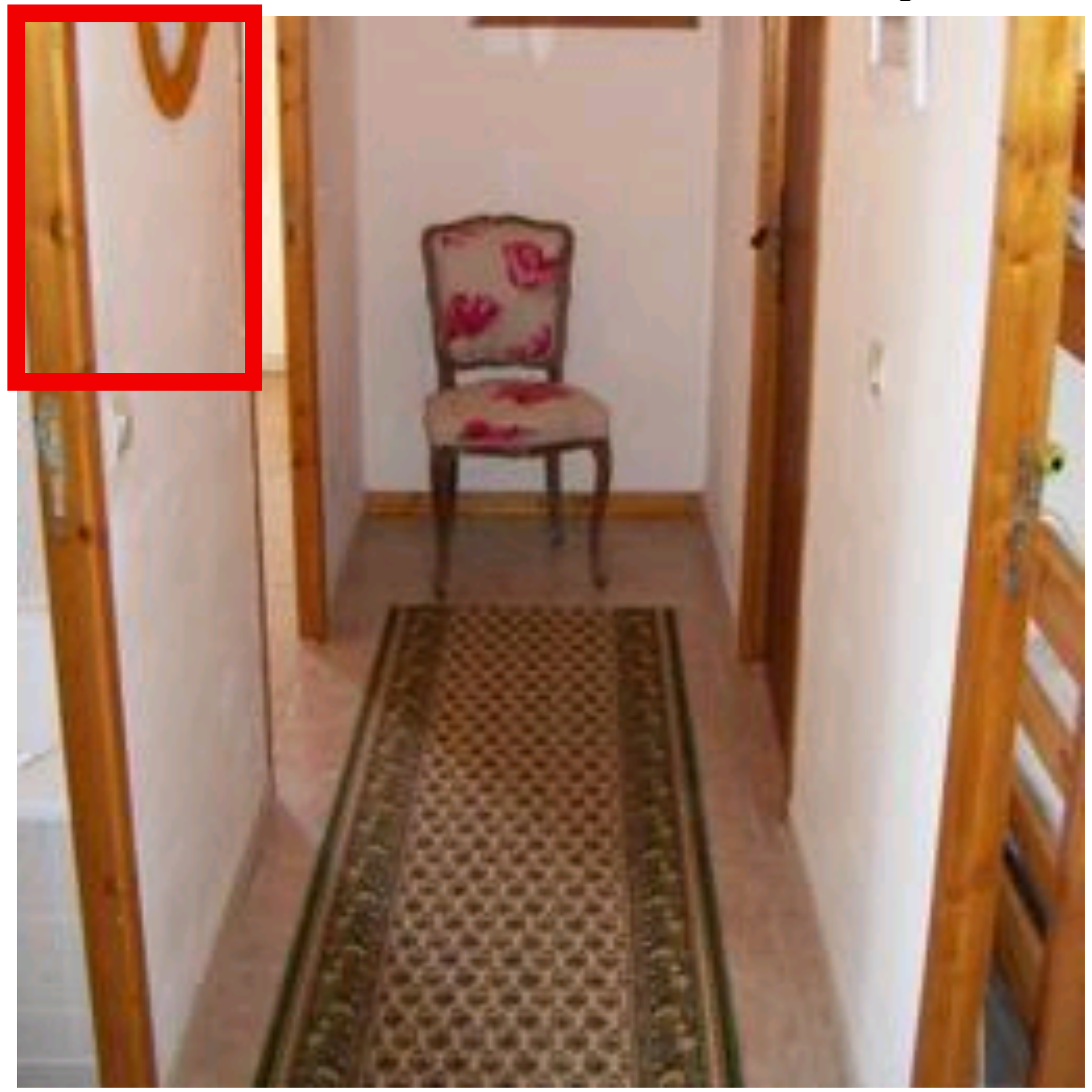
Slide credit: A. Torralba

Object detection: naive attempt

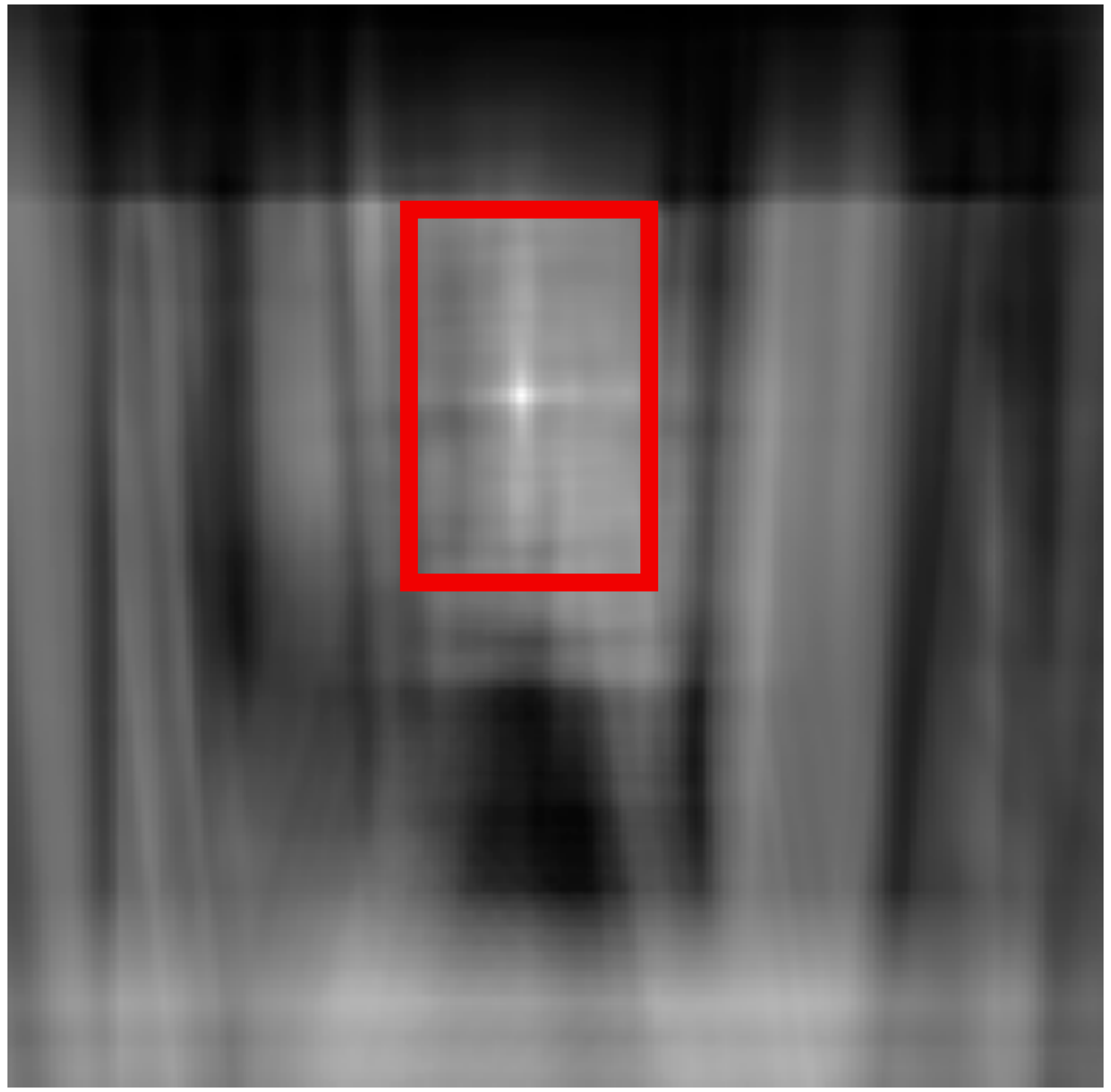
This is a chair



Find the chair in this image

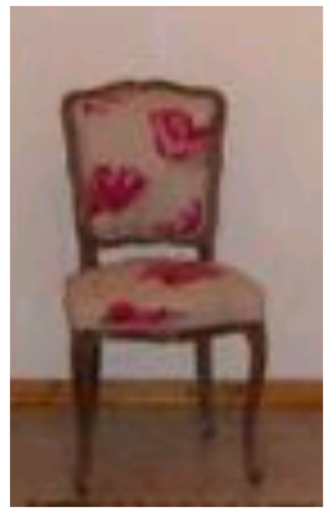


Output of normalized correlation

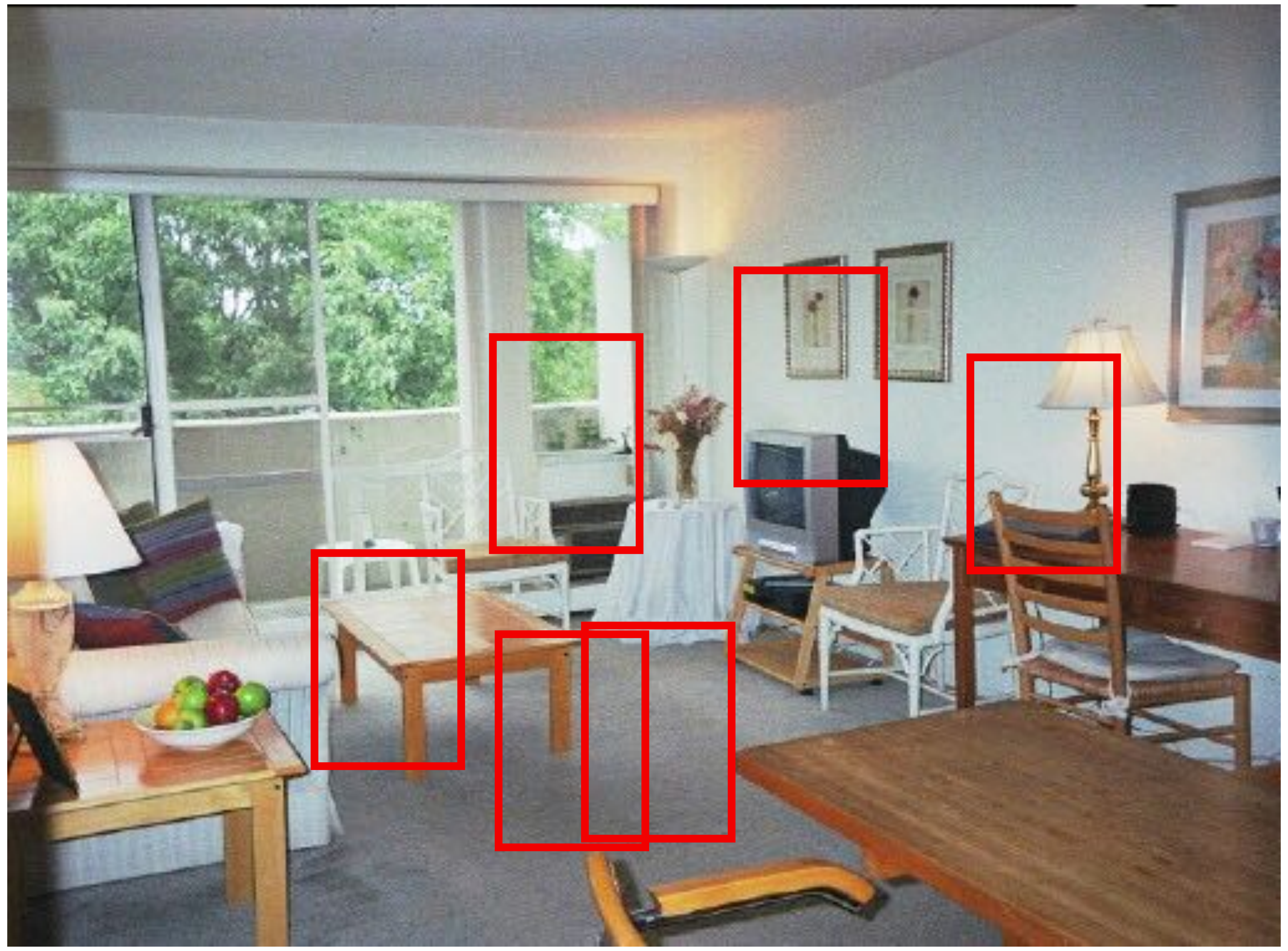


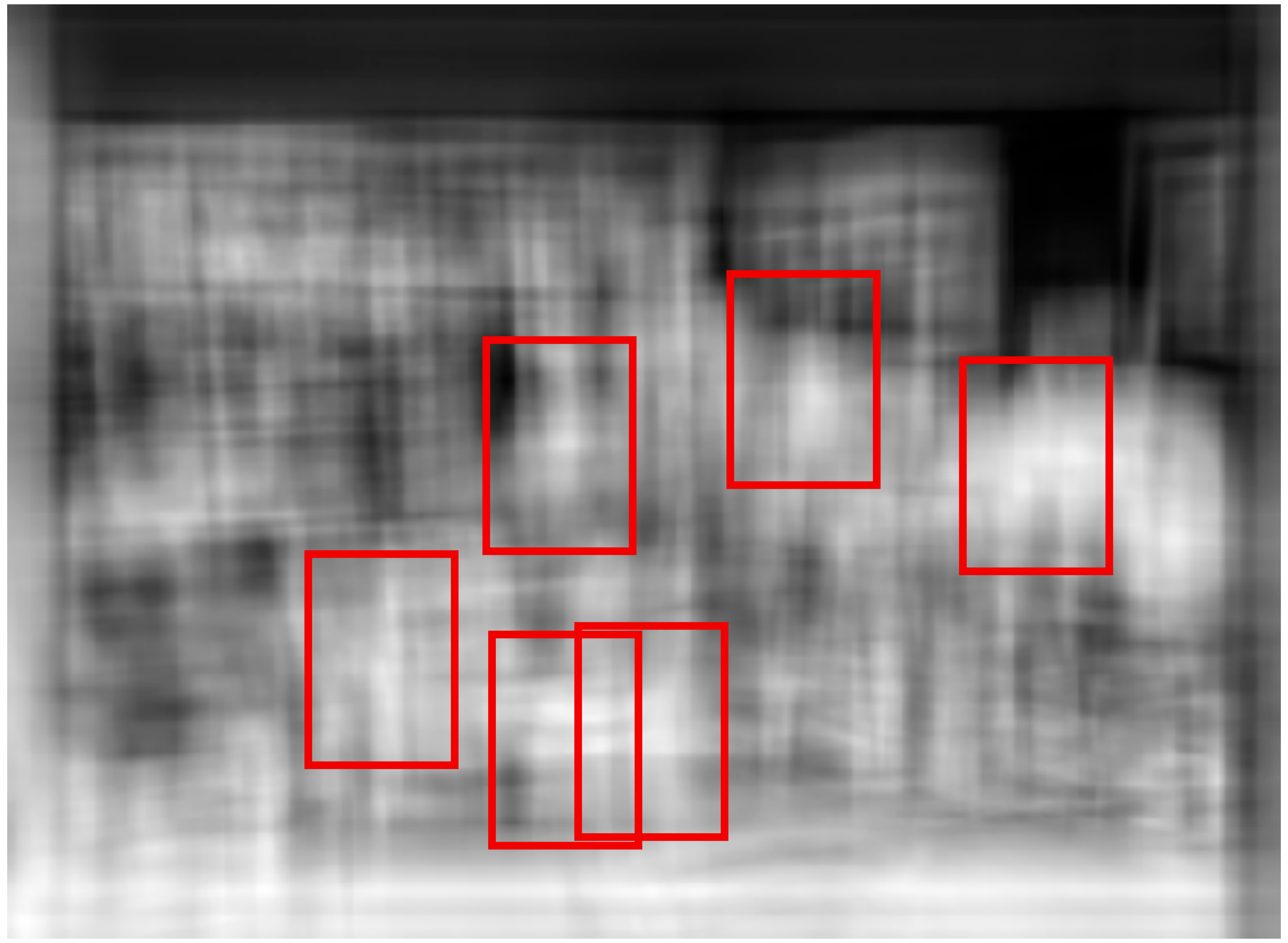
Slide credit: A. Torralba

Object detection: naive attempt



Find the chair in this image



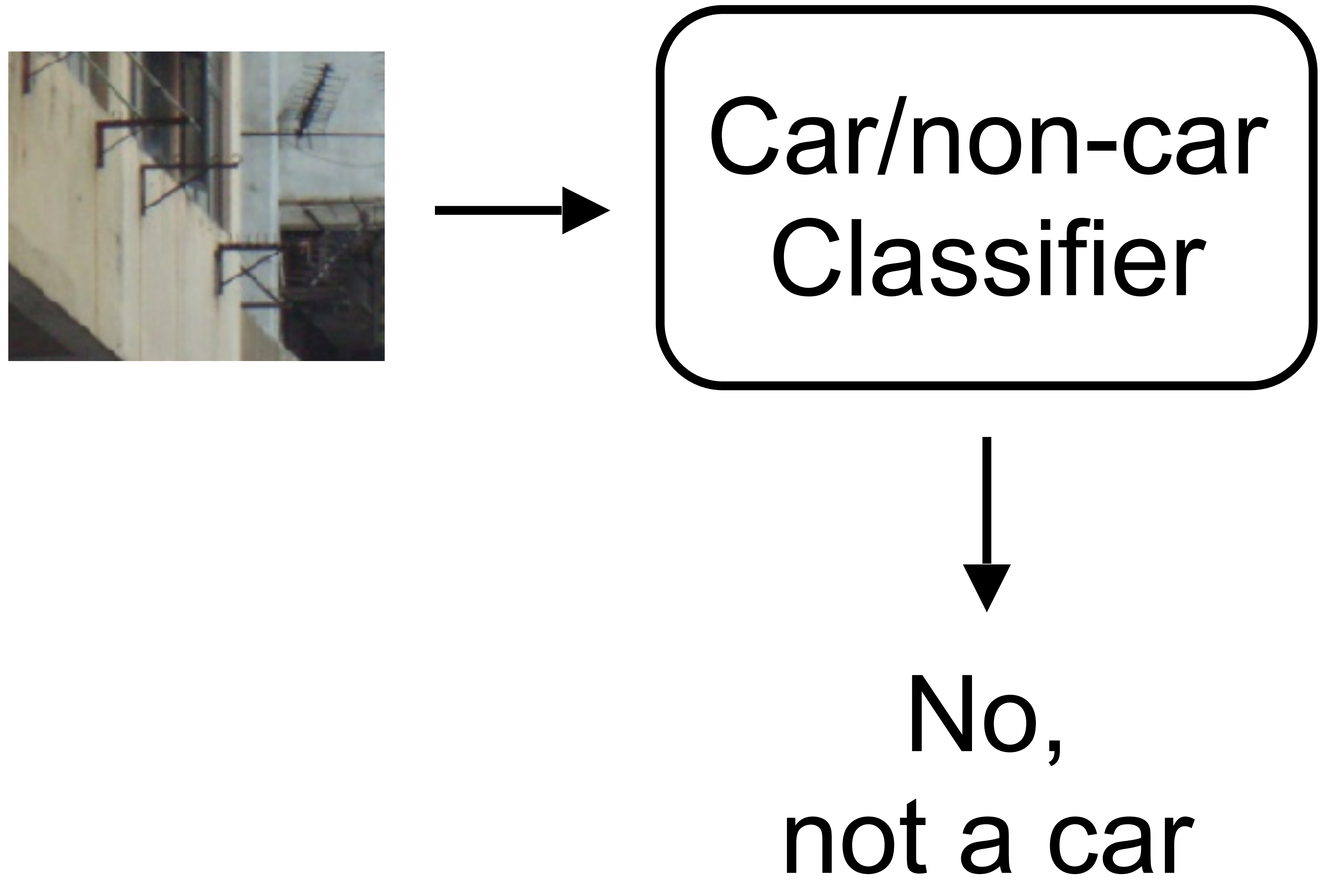


Pretty much garbage
Simple template matching is not going to make it

Slide credit: A. Torralba

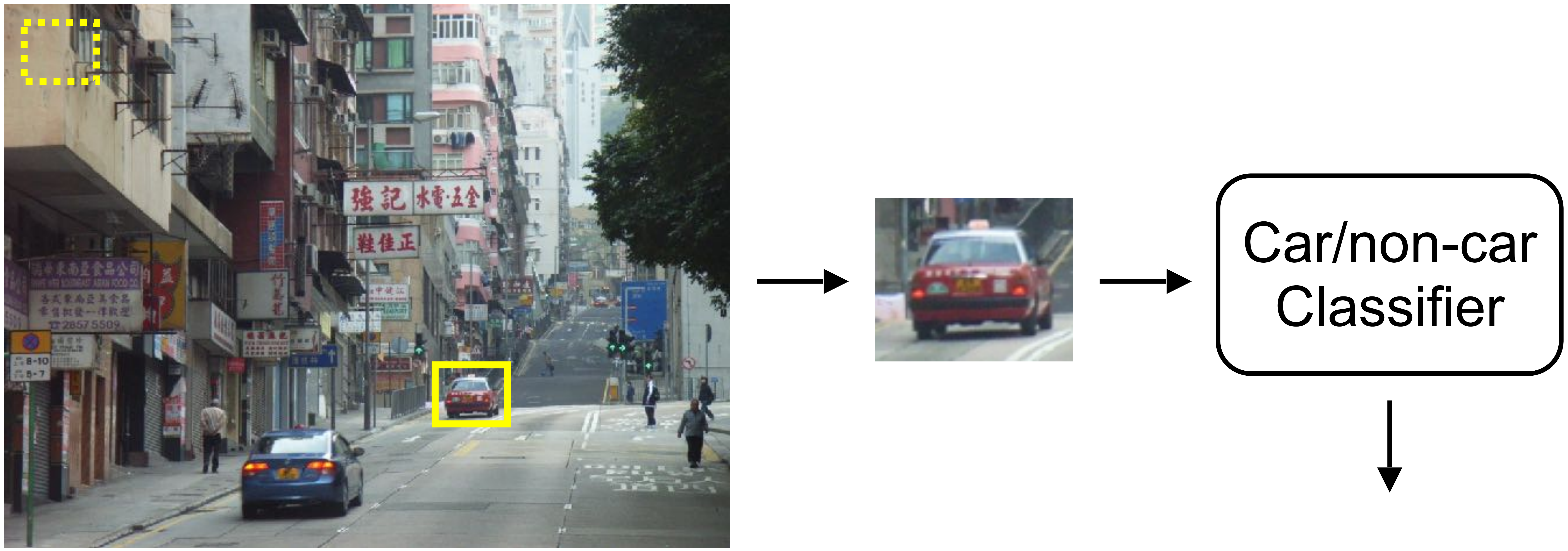
Detection by Classification

- Basic component: binary classifier



Detection by Classification

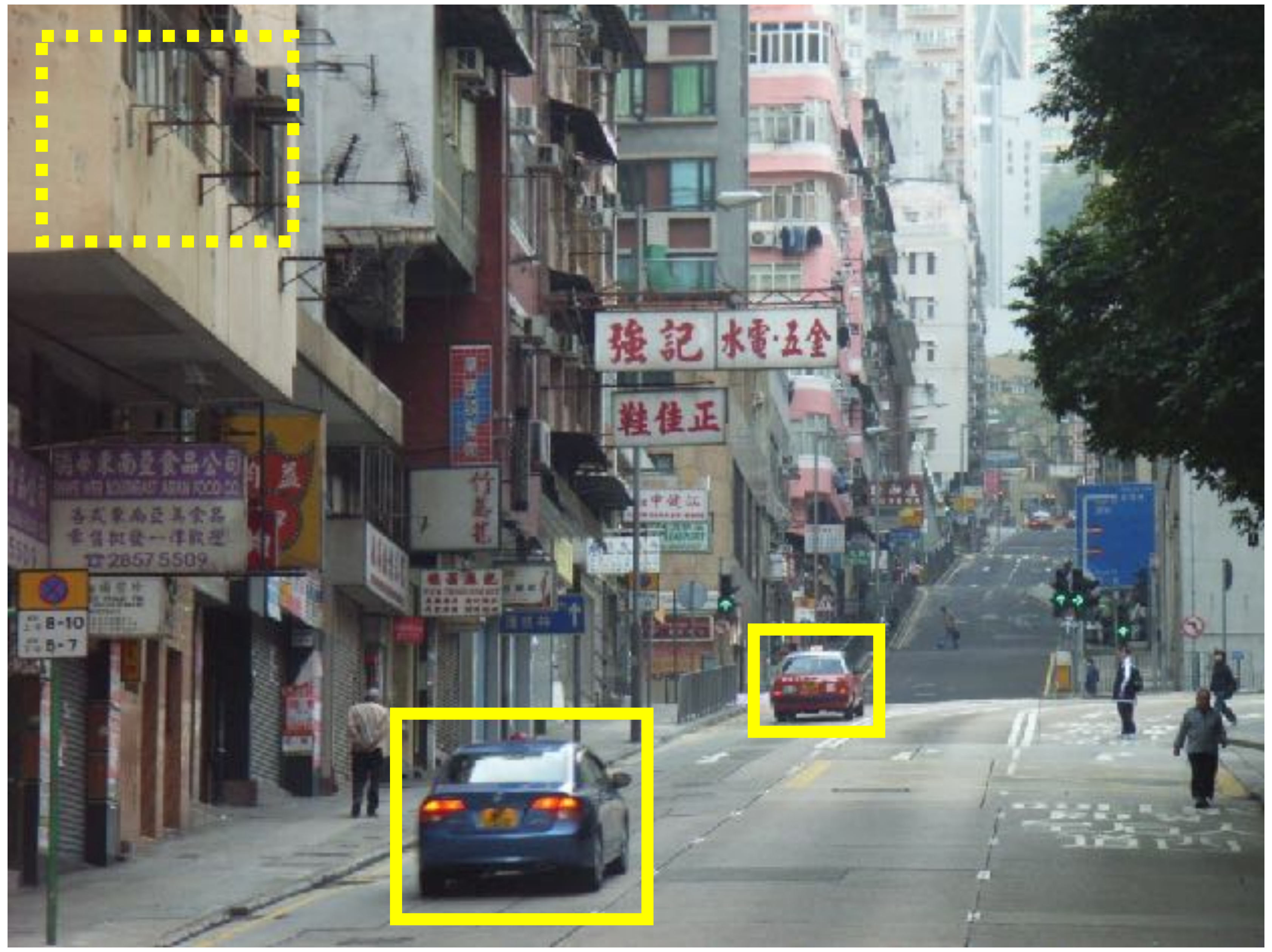
- Detect objects in clutter by <u>**search**</u>



- **Sliding window**: exhaustive search over position and scale

Detection by Classification

- Detect objects in clutter by **search**

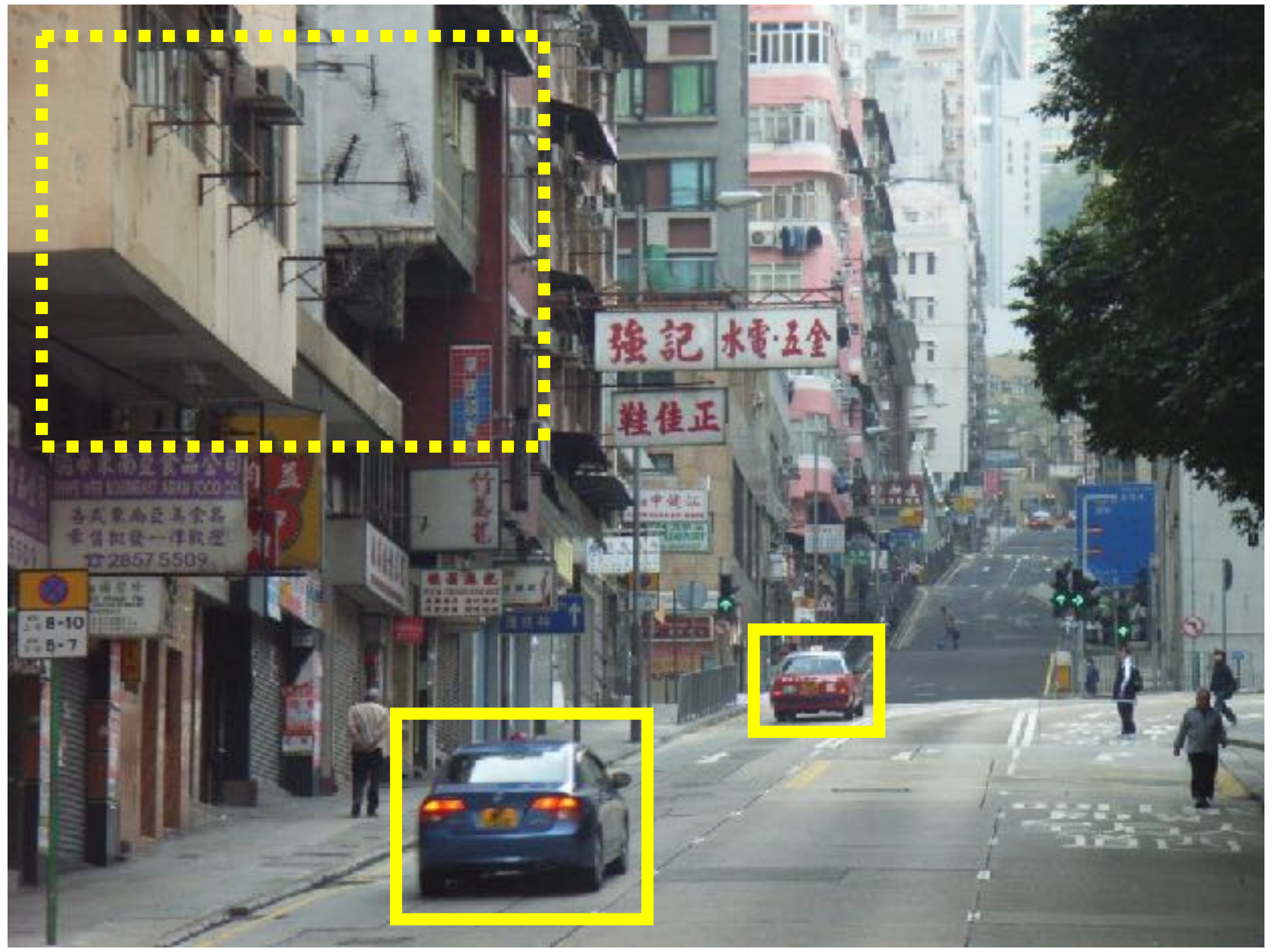


Car/non-car Classifier

- **Sliding window**: exhaustive search over position and scale

Detection by Classification

- Detect objects in clutter by <u>**search**</u>

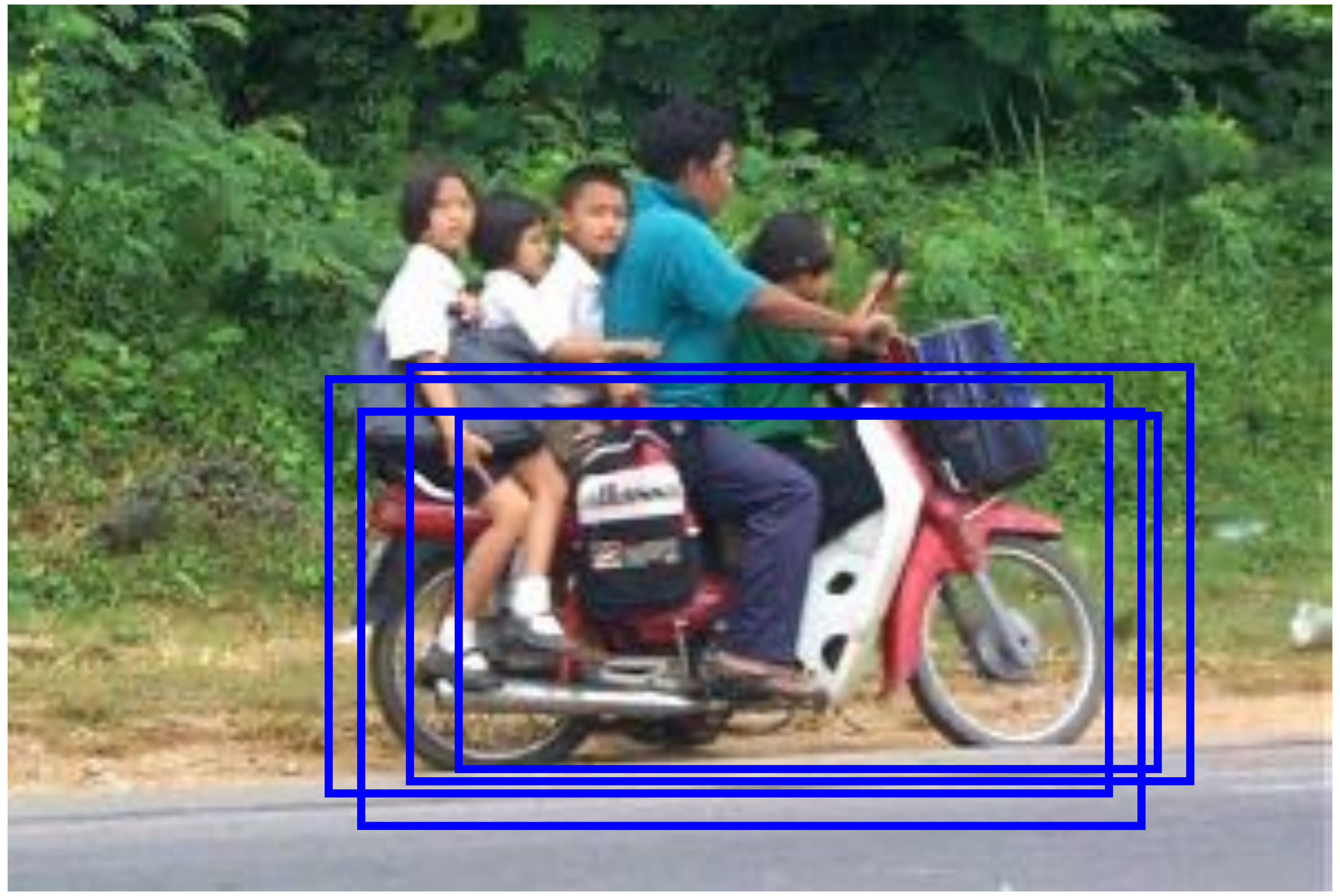


Car/non-car Classifier

- **Sliding window**: exhaustive search over position and scale
(can use same size window over a spatial pyramid of images)

Test: Non-maximum suppression (NMS)

- Scanning-window detectors typically result in multiple responses for the same object

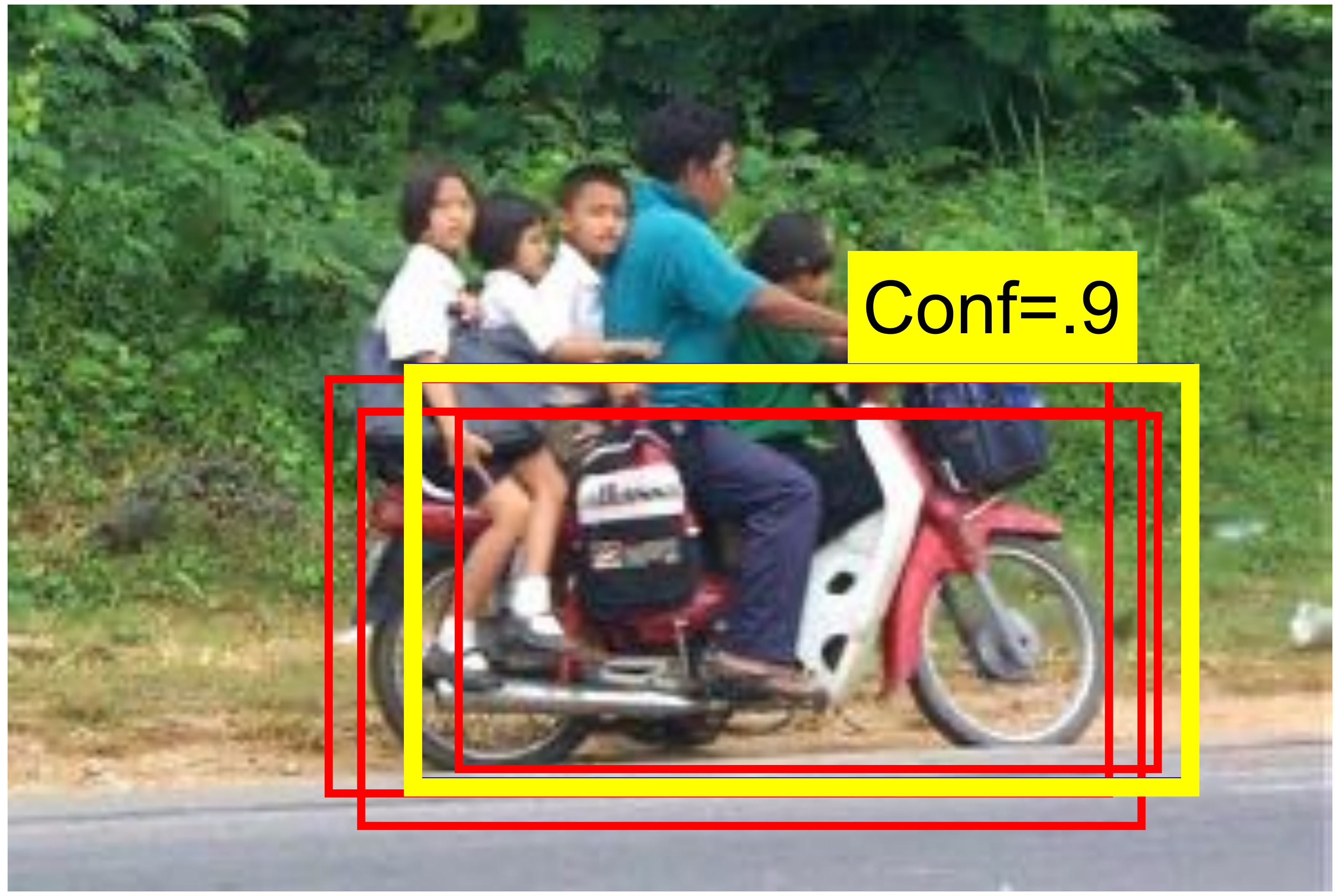


- To remove multiple responses, a simple greedy procedure called “Non-maximum suppression” is applied:

NMS:
1. Sort all detections by detector confidence
2. Choose most confident detection d_i; remove all d_j s.t. $overlap(d_i,d_j)>T$
3. Repeat Step 2. until convergence

Test: Non-maximum suppression (NMS)

- Scanning-window detectors typically result in multiple responses for the same object

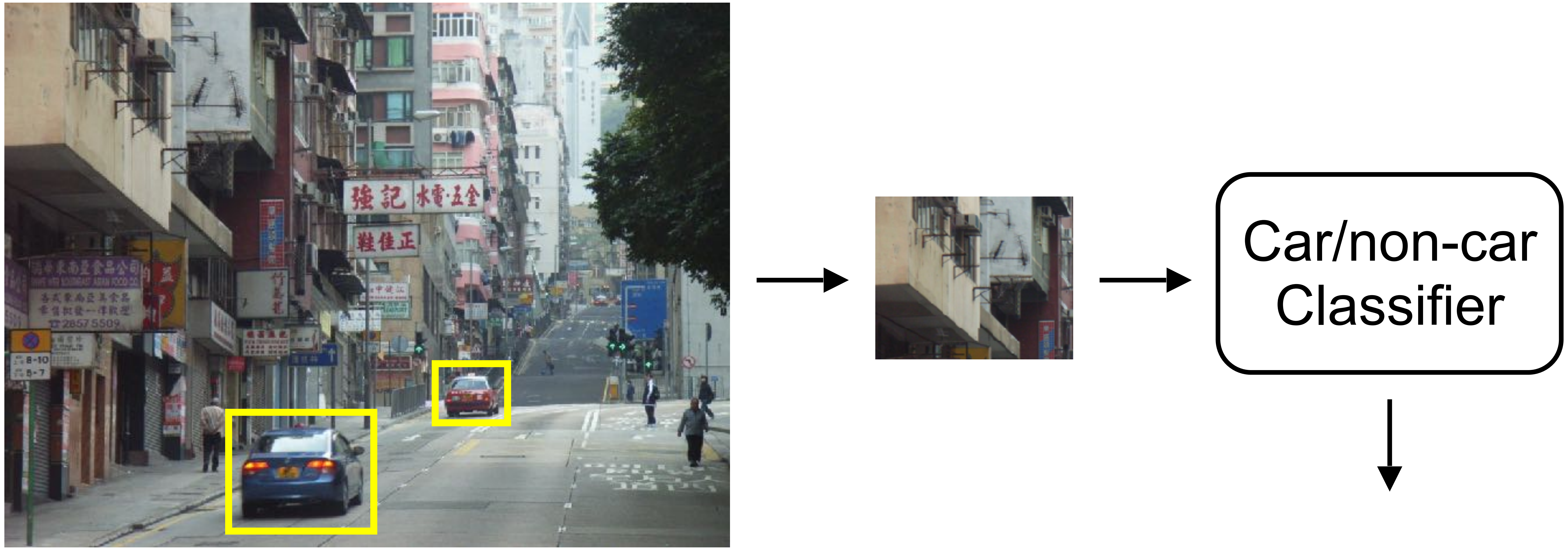


- To remove multiple responses, a simple greedy procedure called “Non-maximum suppression” is applied:

NMS:
1. Sort all detections by detector confidence
2. Choose most confident detection d_i; remove all d_j s.t. $overlap(d_i,d_j)>T$
3. Repeat Step 2. until convergence

Detection by Classification

- Detect objects in clutter by **search**

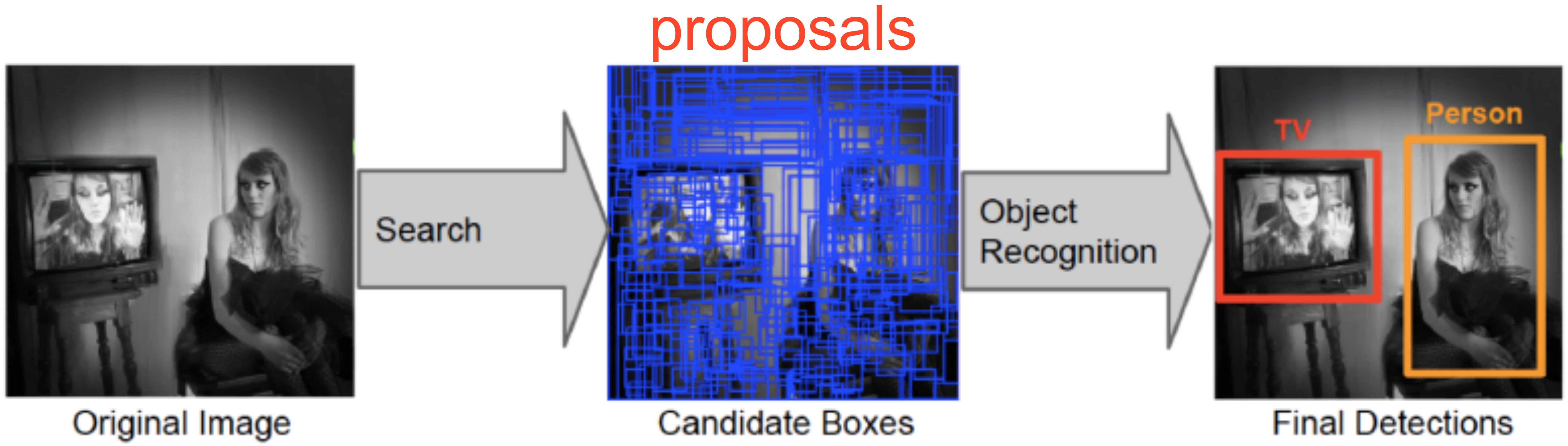


Problem: too many windows to run a classifier

- **Sliding window**: exhaustive search over position and scale (can use same size window over a spatial pyramid of images)

Object proposals

Generate and evaluate a few hundred region proposals.



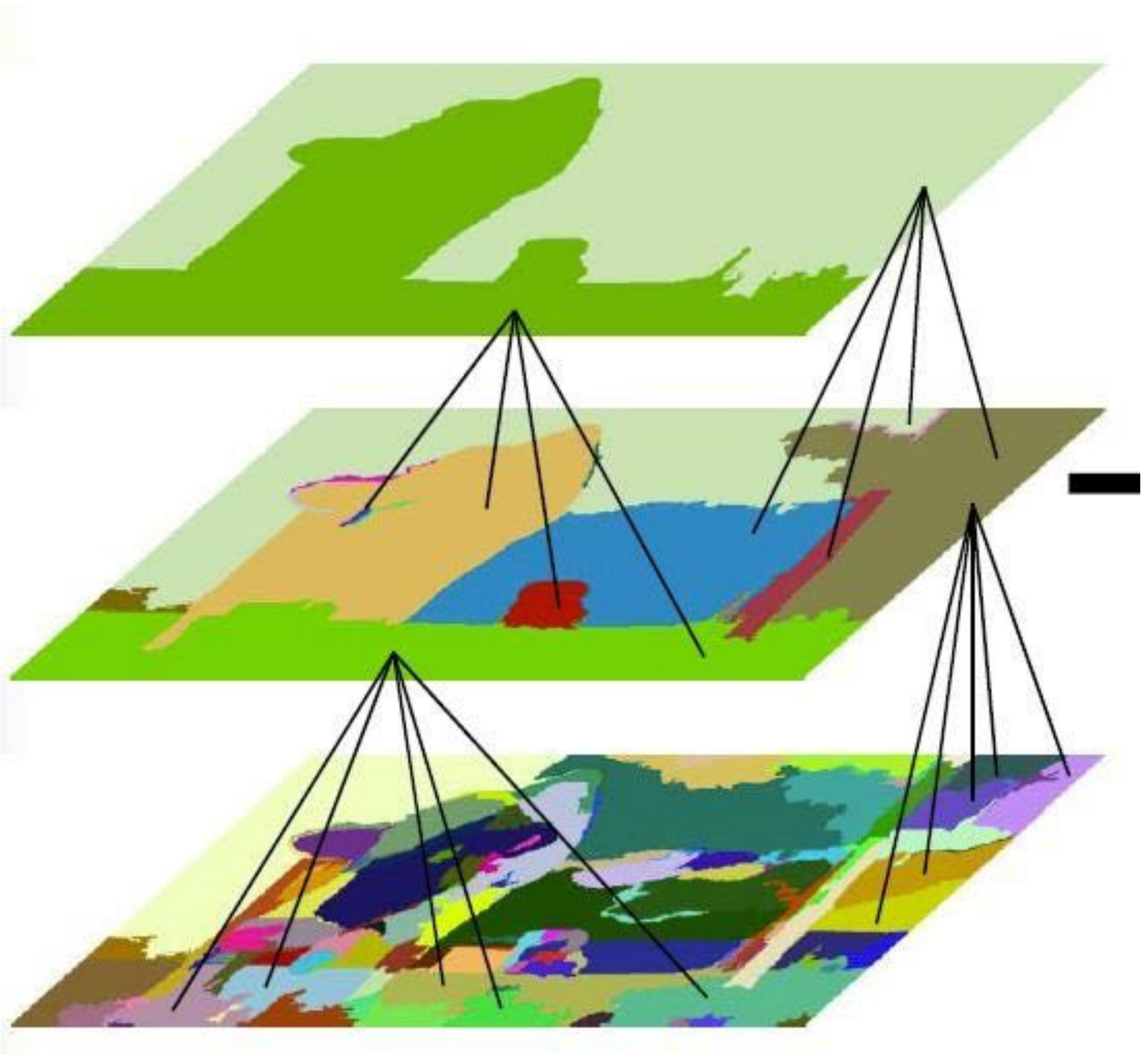
- Proposal mechanism can:
  ➡ take advantage of low-level perceptual organization cues,
  ➡ be category-specific or category-independent, handcrafted or trained.

- Classifier can be slower but more powerful.

Slide credit: Lana Lazebnk

Region proposals: Selective search

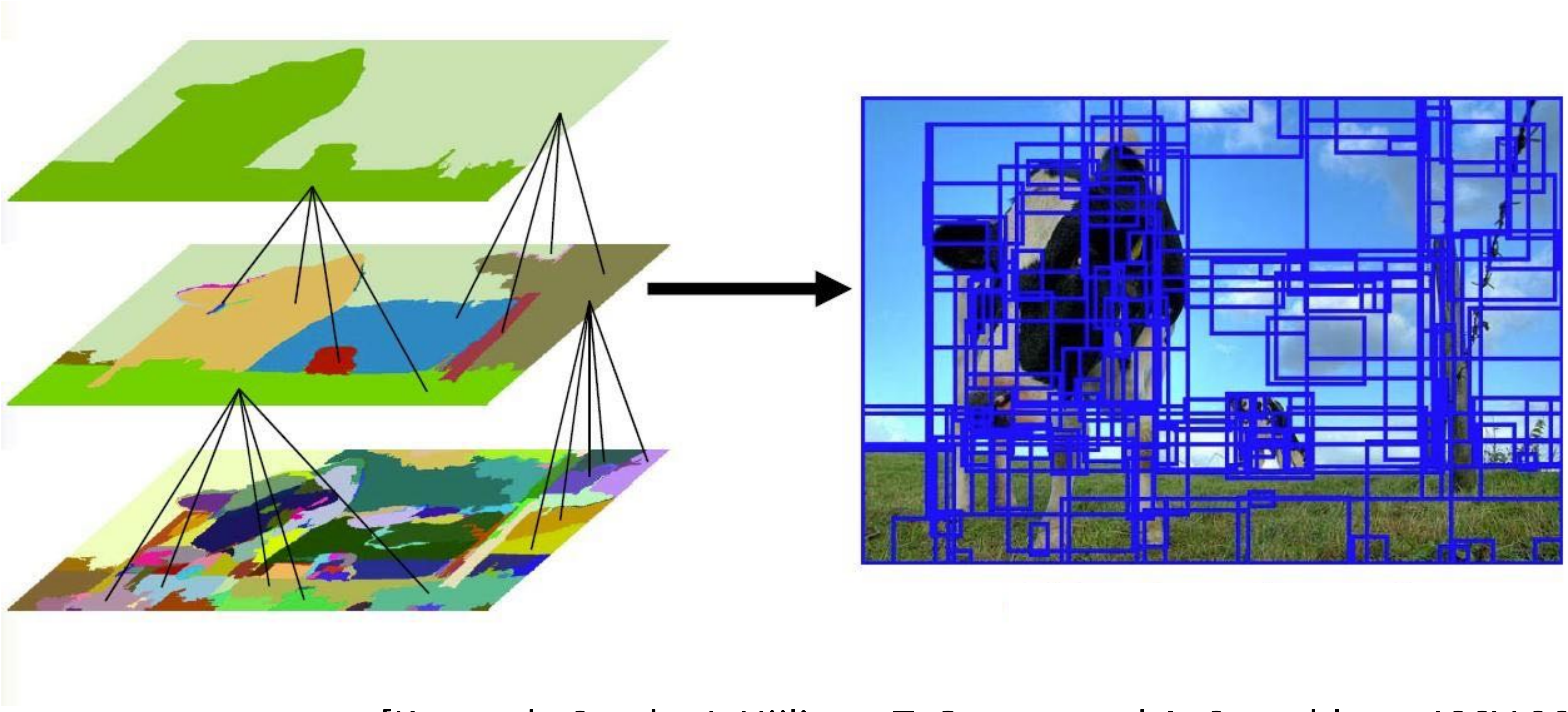
1. Merge two most similar regions based on similarity.
2. Update similarities between the new region and its neighbors.
3. Go back to step 1. until the whole image is a single region.



[K. van de Sande, J. Uijlings, T. Gevers, and A. Smeulders, ICCV 2011]

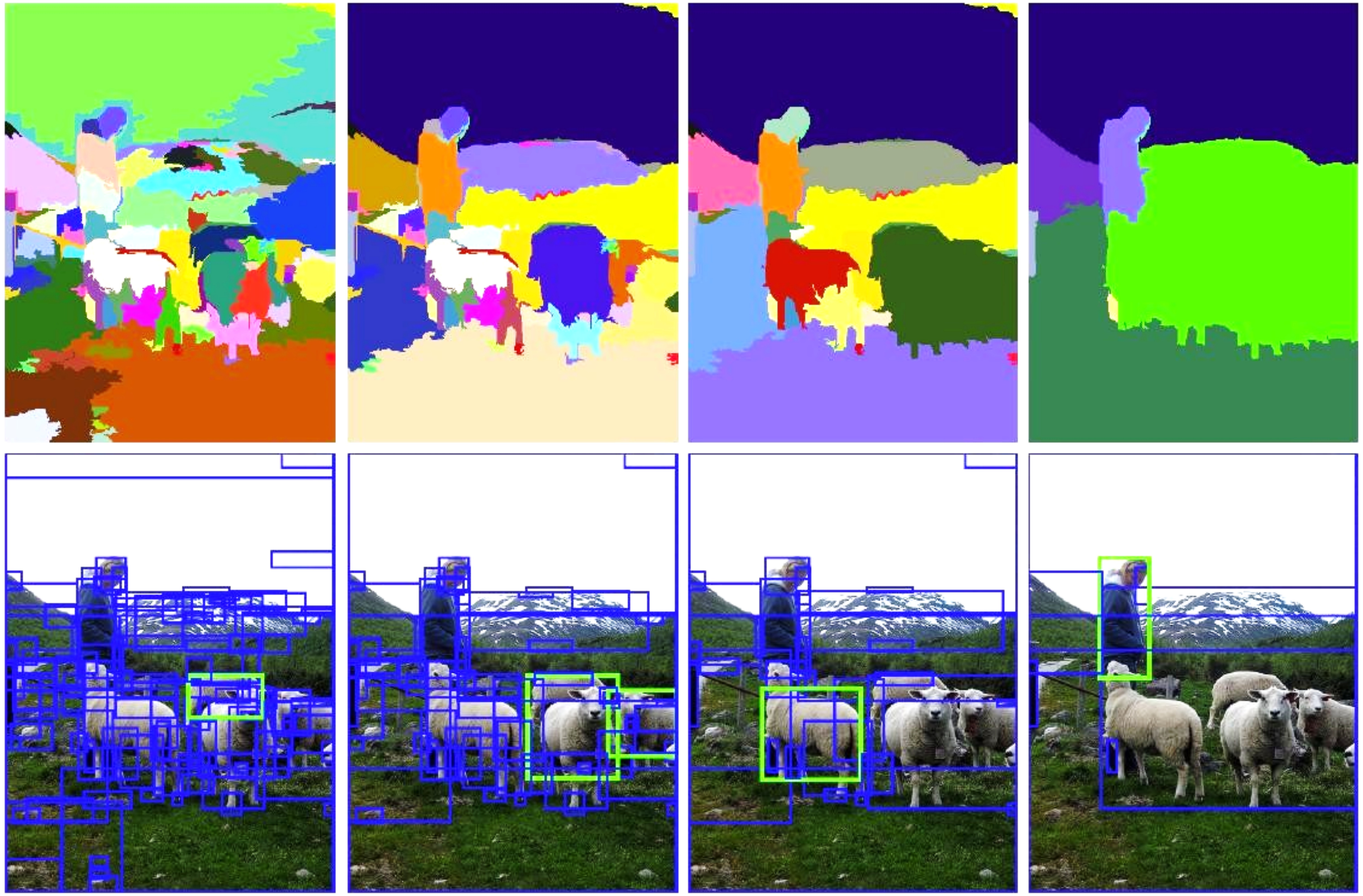
Region proposals: Selective search

Take bounding boxes of all generated regions and treat them as possible object locations.



[K. van de Sande, J. Uijlings, T. Gevers, and A. Smeulders, ICCV 2011]

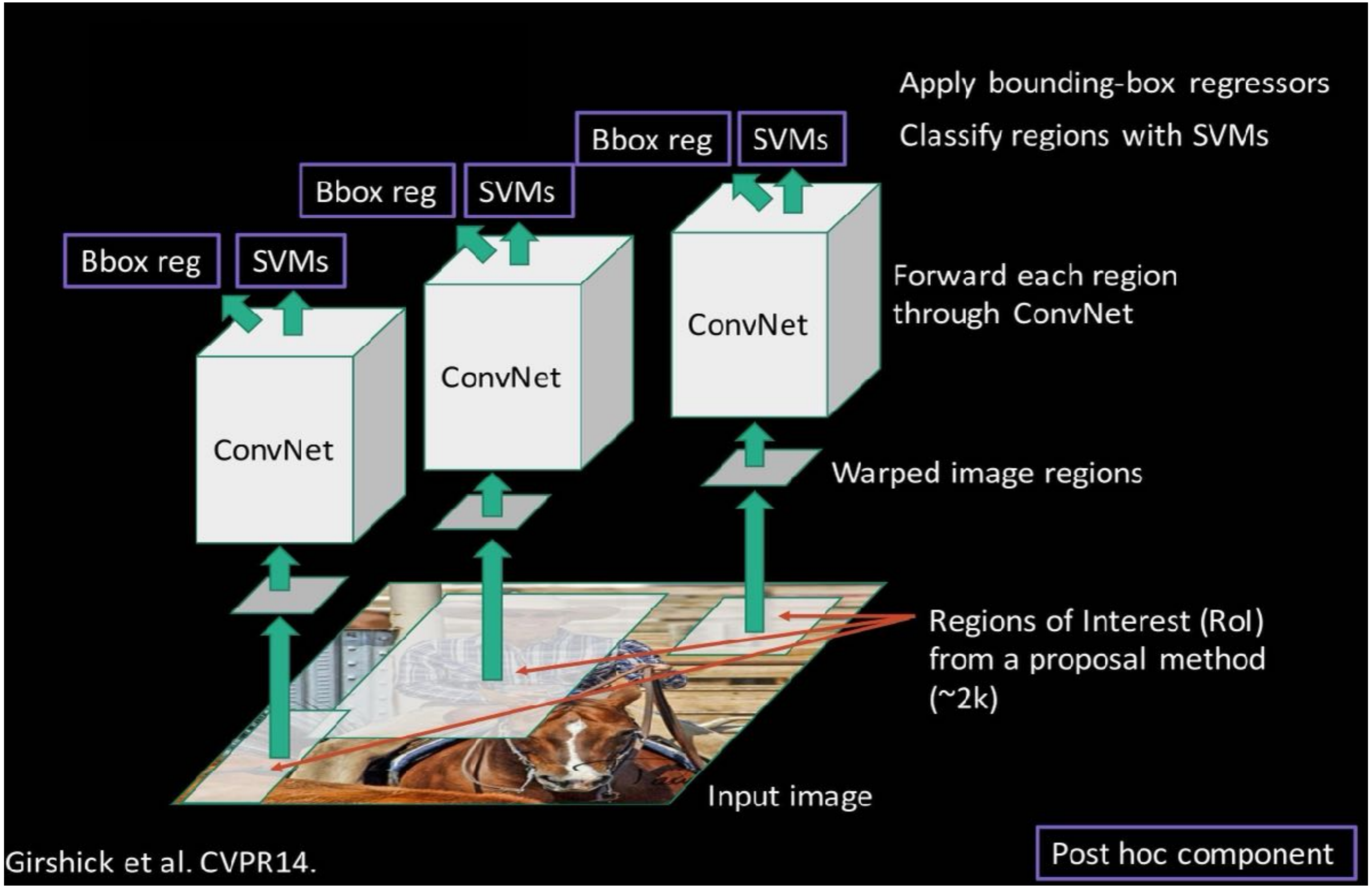
Region proposals: Selective search



[K. van de Sande, J. Uijlings, T. Gevers, and A. Smeulders, ICCV 2011]

Object detection: CNN-based methods

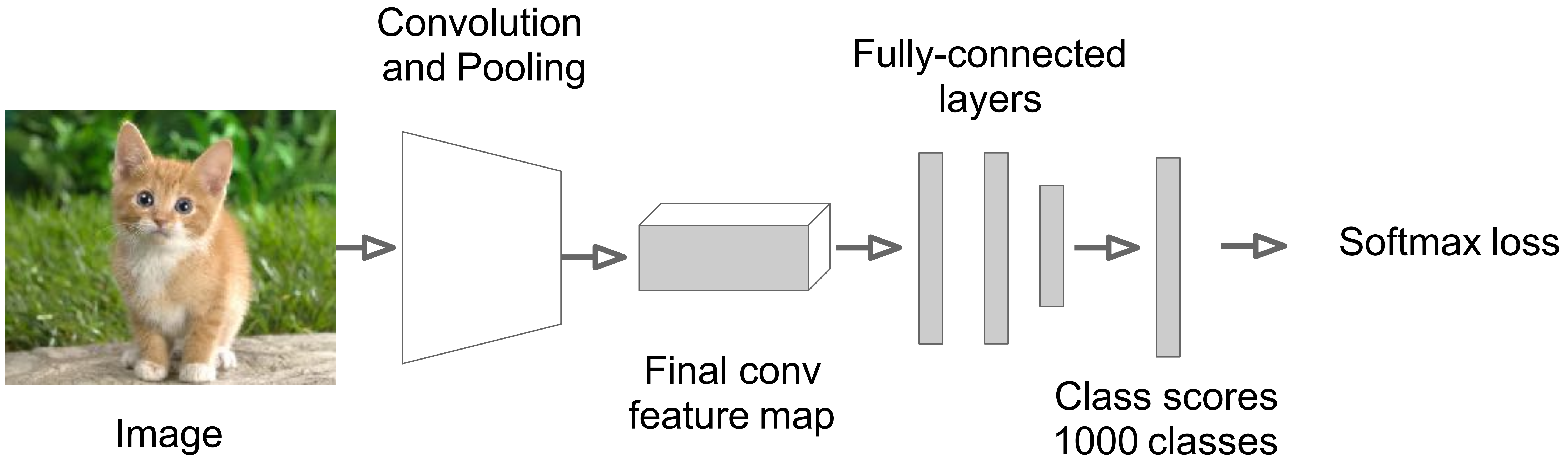
R-CNN: Region-based CNN



Slide credit: Ross Girschick [Girschick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014]

R-CNN Training

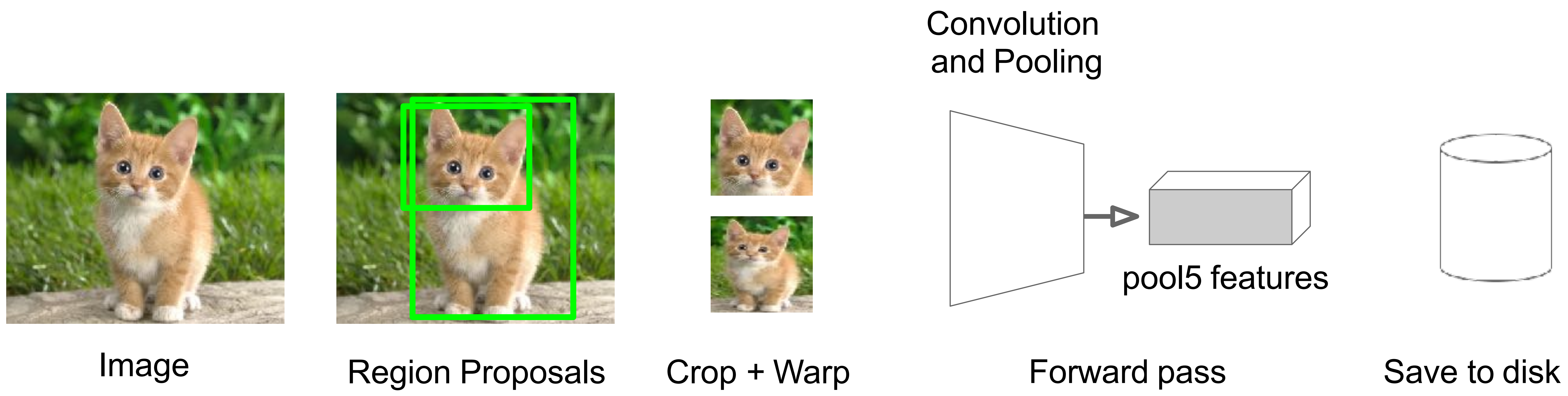
Step 1: Train (or download) a classification model for ImageNet (AlexNet)



R-CNN Training

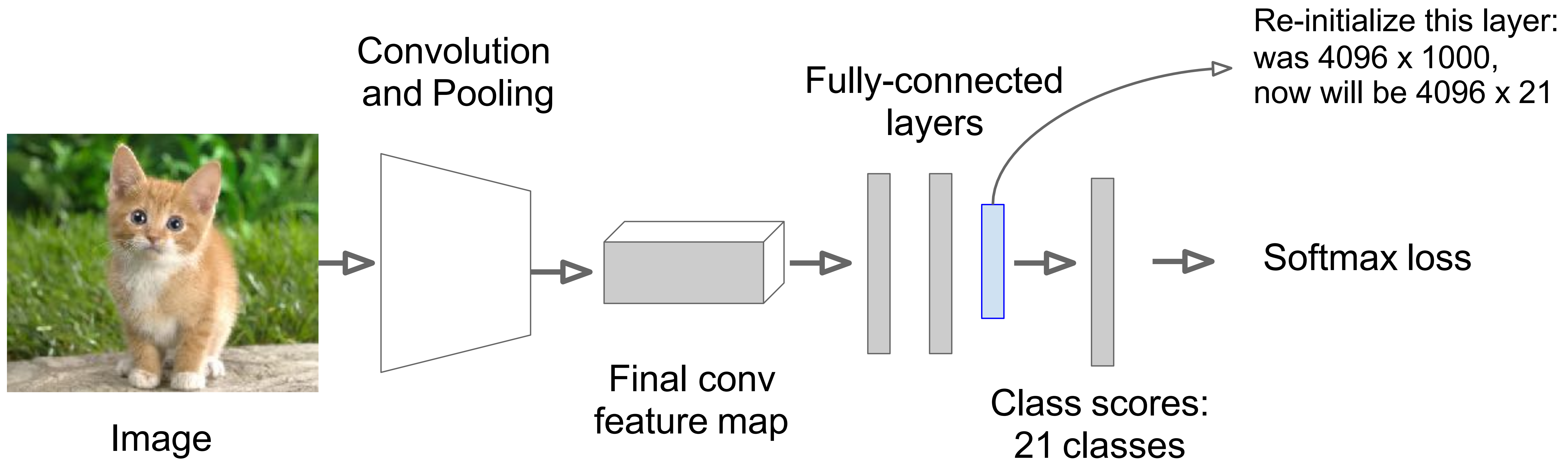
Step 2: Extract features

- Extract region proposals for all images
- For each region: warp to CNN input size, run forward through CNN, save pool5 features to disk
- Have a big hard drive: features are ~200GB for PASCAL dataset!



R-CNN Training

Step 3: Fine-tune model for detection
- Instead of 1000 ImageNet classes, want 20 object classes + background
- Throw away final fully-connected layer, reinitialize from scratch
- Keep training model using positive / negative regions from detection images

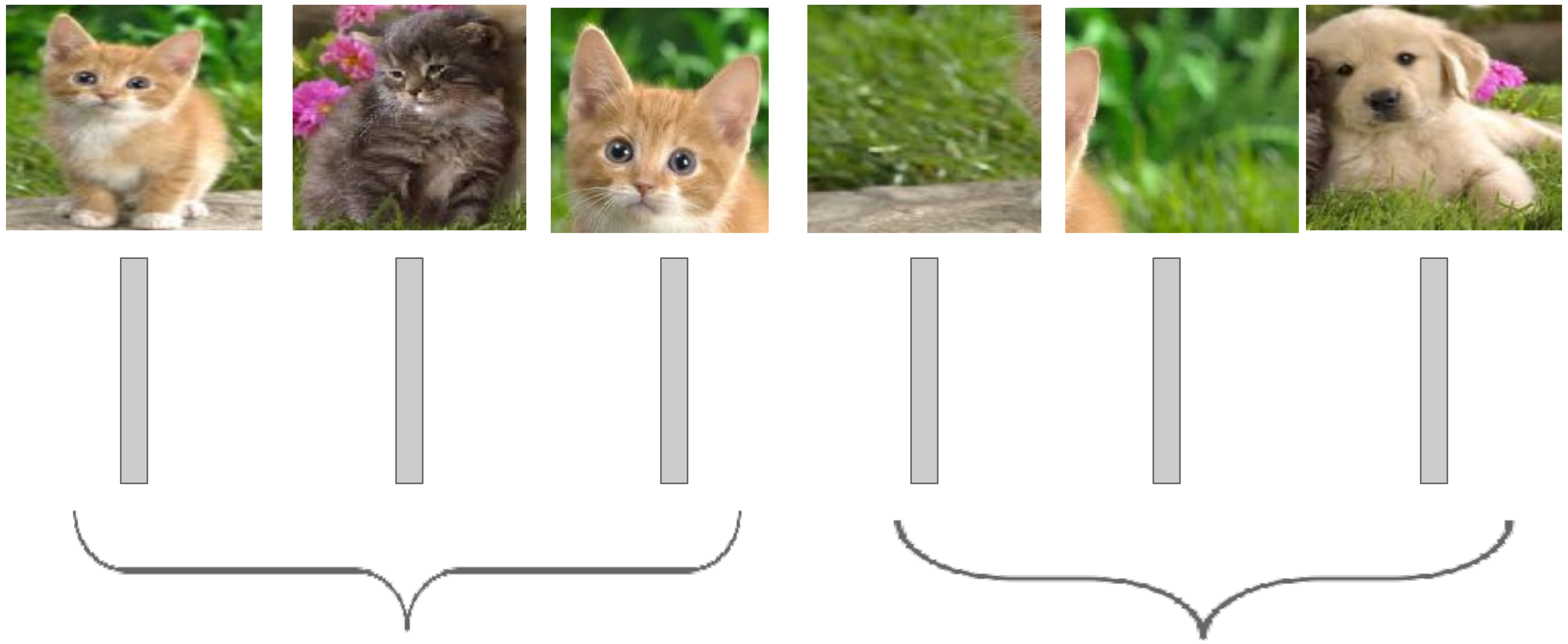


R-CNN Training

Step 4: Train **one binary SVM per class** to classify region features

Training image regions

Cached region features



+ Positive samples for cat SVM

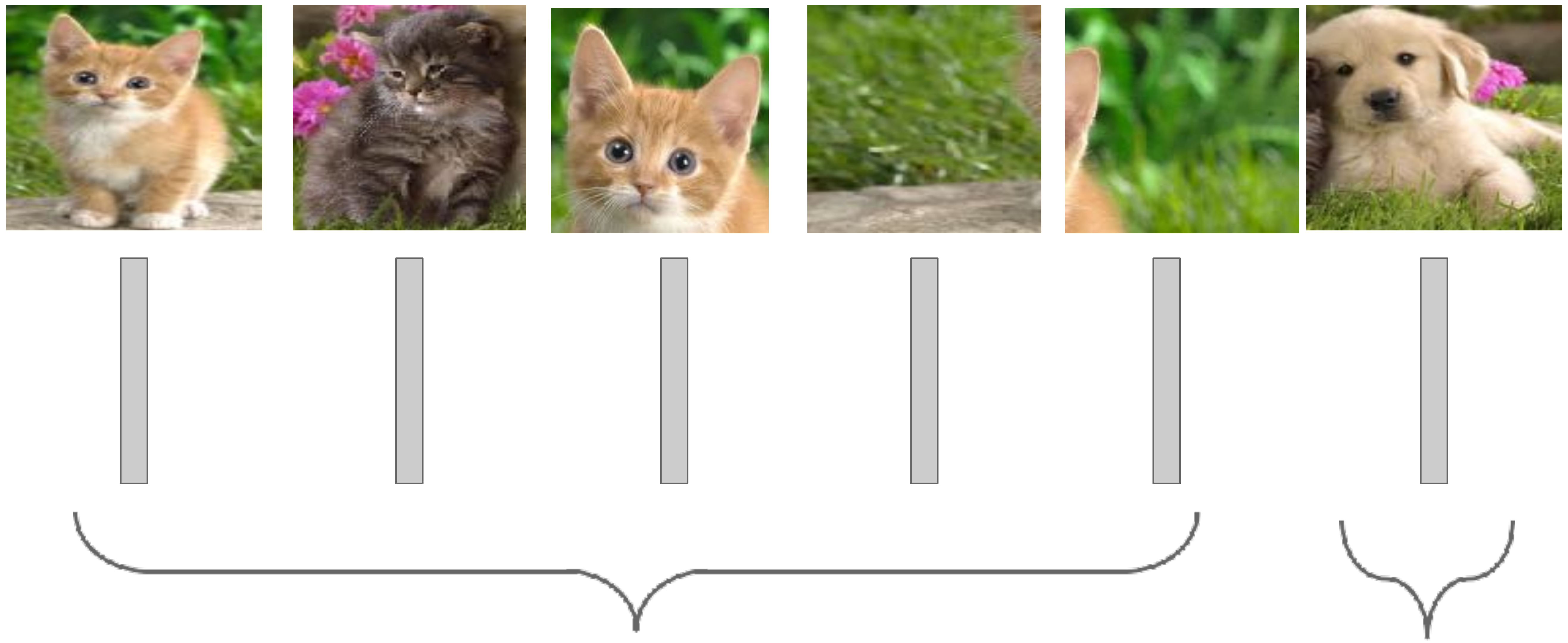
- Negative samples for cat SVM

R-CNN Training

Step 4: Train **one binary SVM per class** to classify region features

Training image regions

Cached region features



- Negative samples for dog SVM

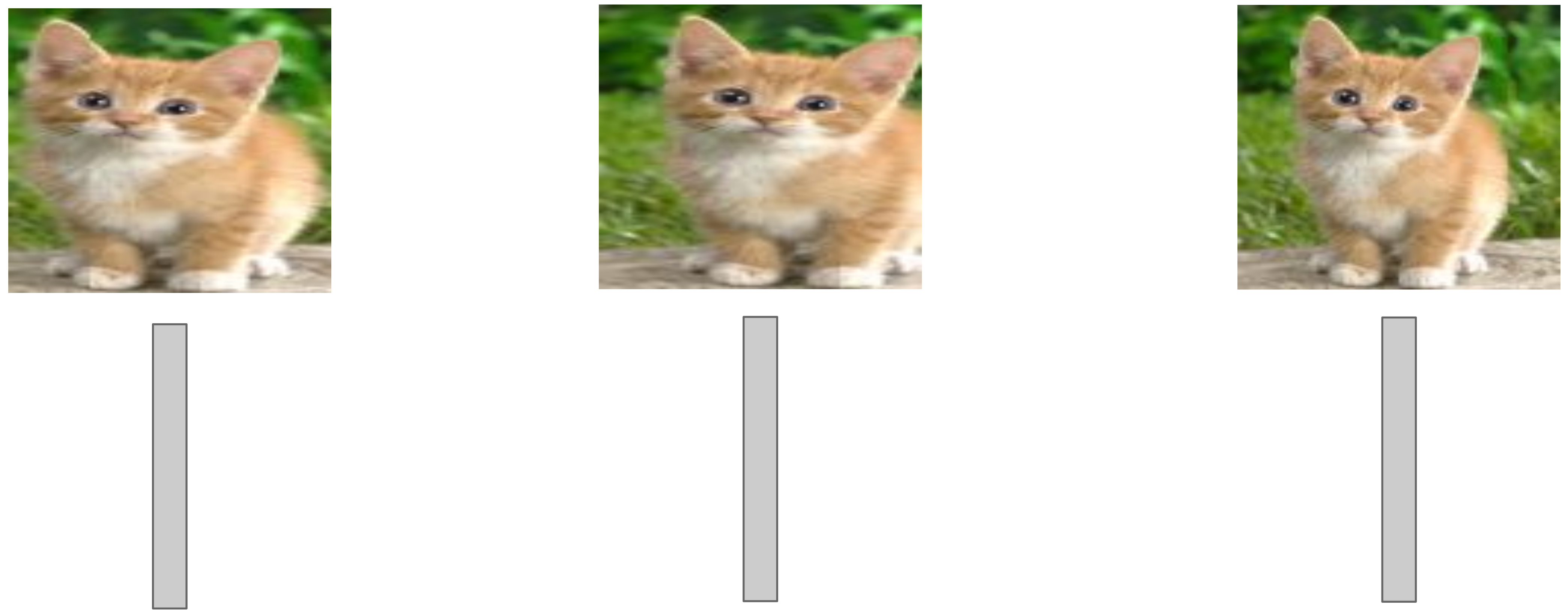
+ Positive samples for dog SVM

R-CNN Training

Step 5 (bbox regression): For each class, train a linear regression model to map from cached features to offsets to GT boxes to make up for "slightly wrong" proposals

Training image regions

Cached region features



Regression targets
(dx, dy, dw, dh)
Normalized coordinates

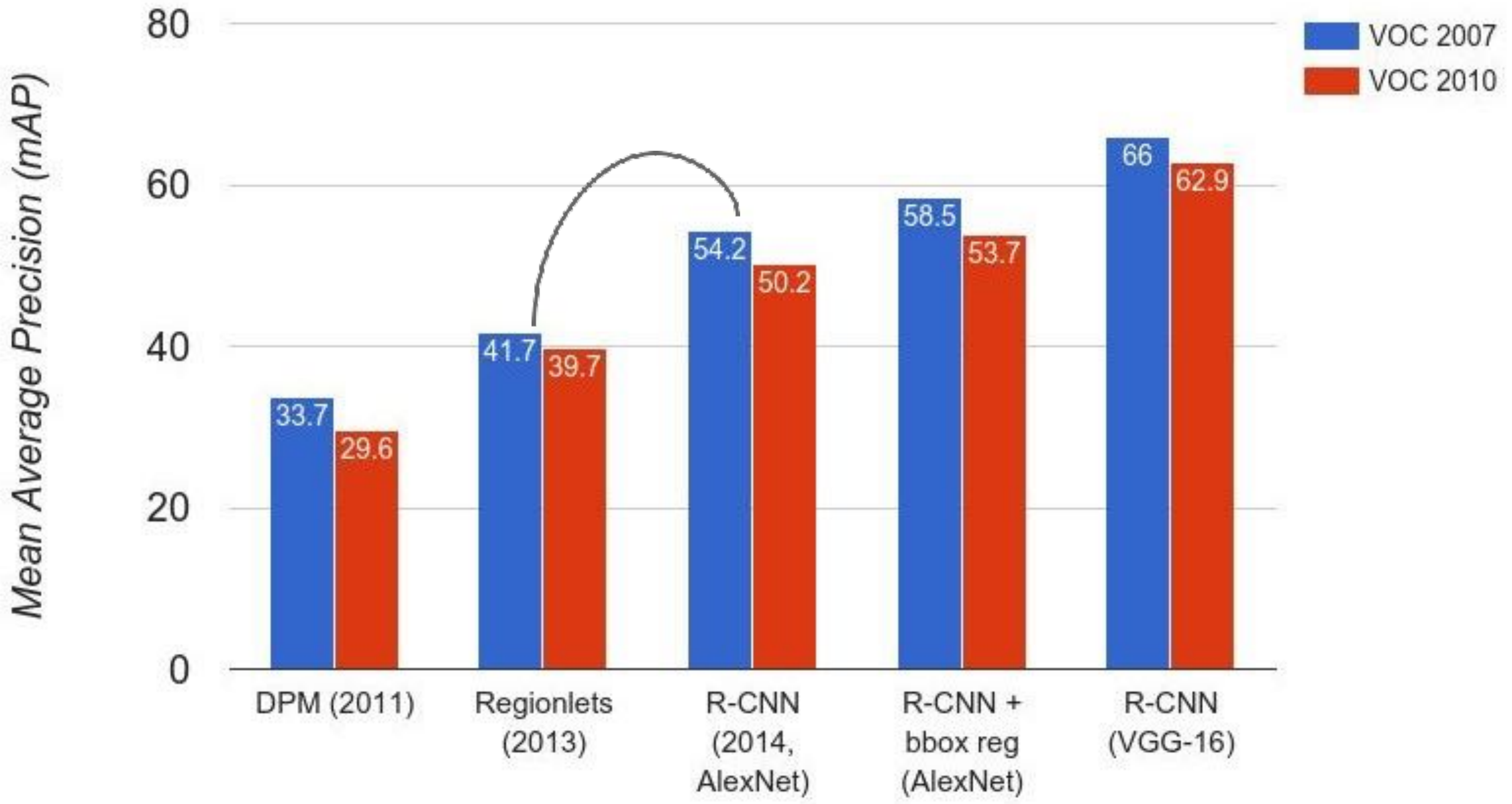
(0, 0, 0, 0)
Proposal is good

(.25, 0, 0, 0)
Proposal too far to left

(0, 0, -0.125, 0)
Proposal too wide

R-CNN Results

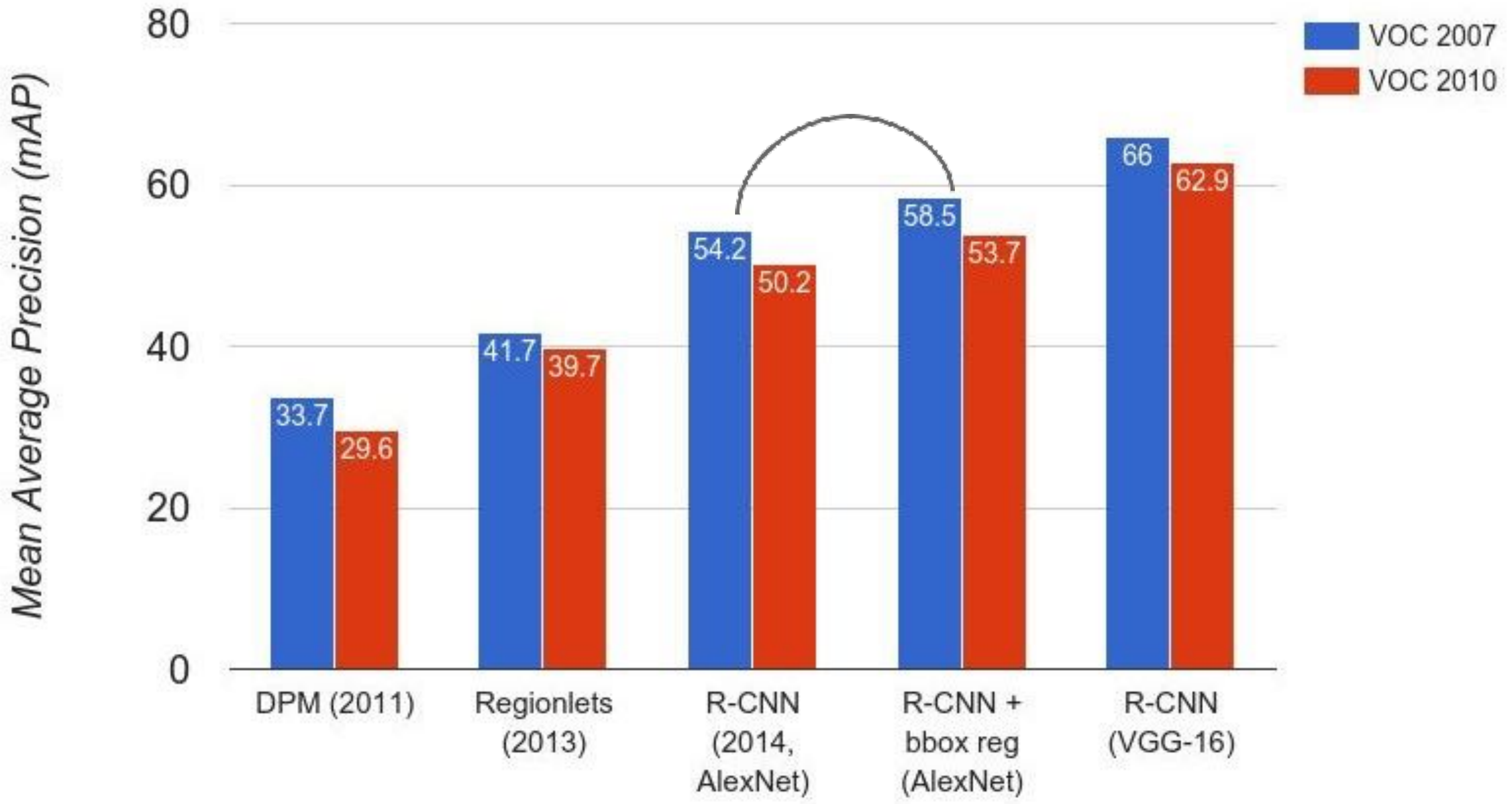
Big improvement compared
to pre-CNN methods



Wang et al, "Regionlets for Generic Object Detection", ICCV 2013

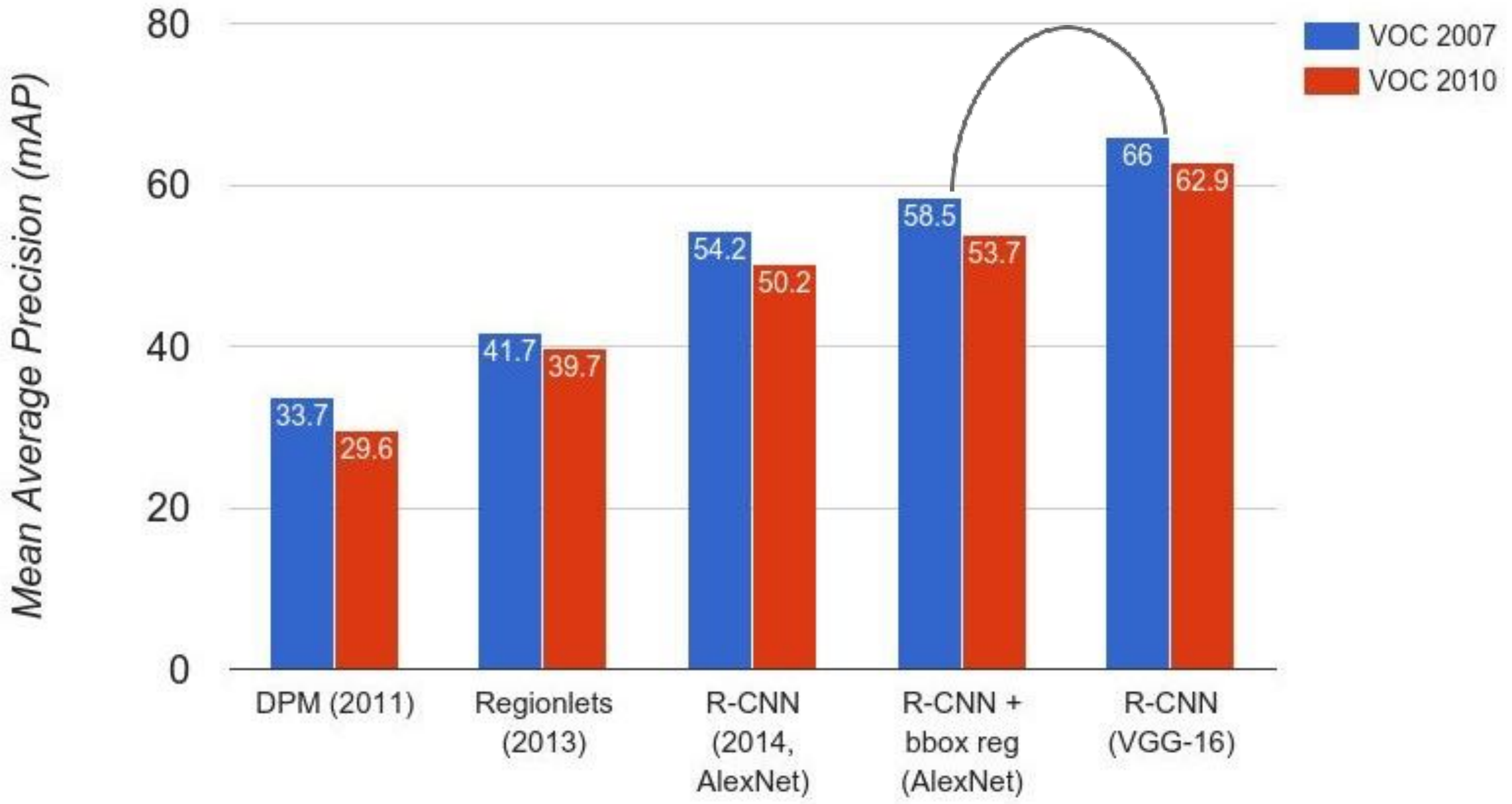
R-CNN Results

Bounding box regression
helps a bit



R-CNN Results

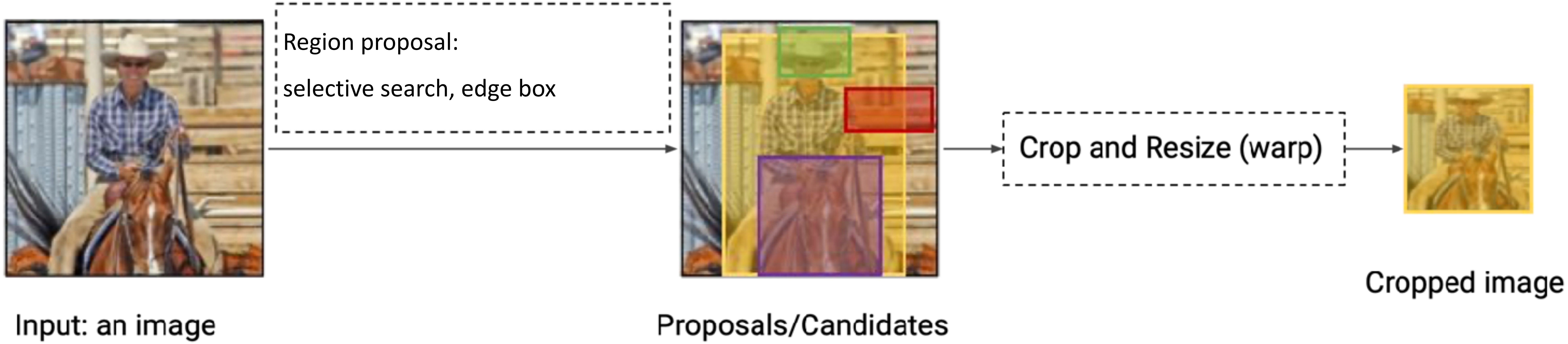
Features from a deeper network help a lot



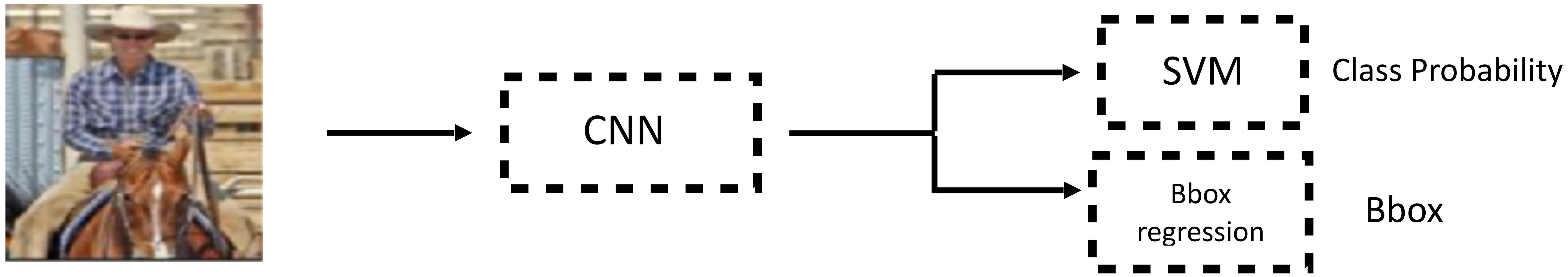
R-CNN [CVPR 2014] Summary

Two-stage detector

- Propose large number of regions potentially with objects



- Classify each proposed region



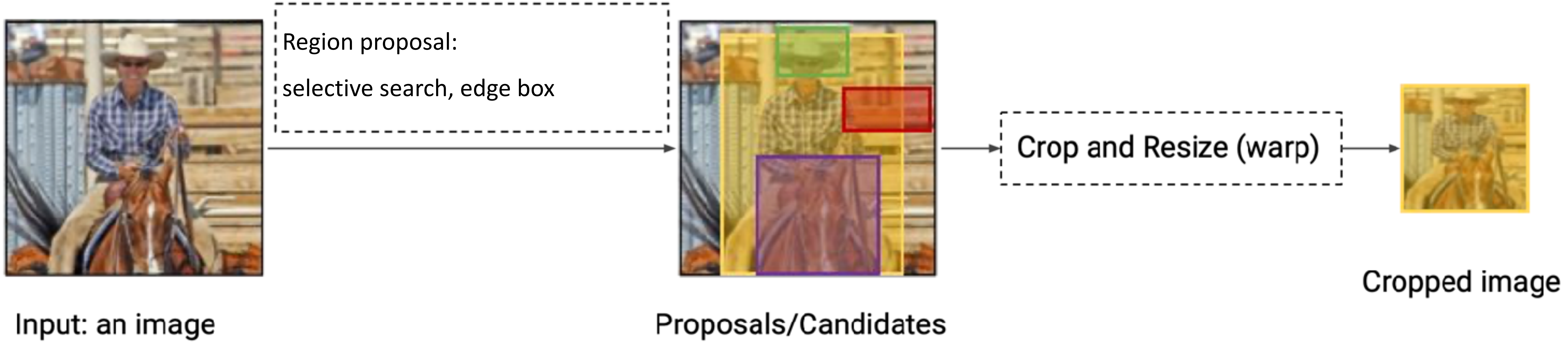
R. Girshick, J. Donahue, T. Darrell, and J. Malik, [Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation](), CVPR 2014.
e.g. region proposal: J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, Selective Search for Object Recognition, IJCV 2013

Slides modified from Ross Girshick tutorial at CVPR 2019

R-CNN [CVPR 2014] Limitations

Two-stage detector

- Propose large number of regions potentially with objects



- Classify each proposed region

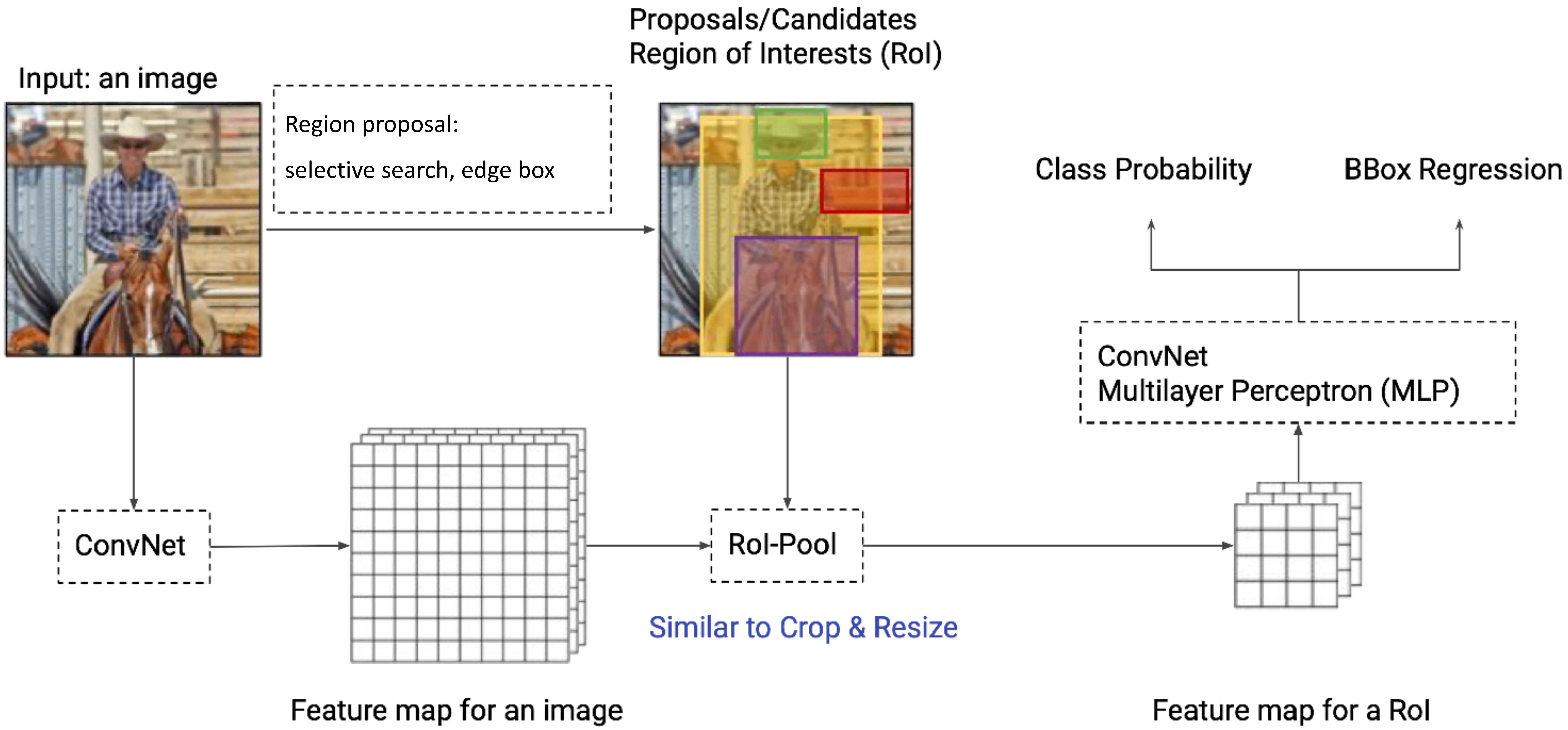
1. **Slow at test-time:** need to run full forward pass of CNN for each region proposal
2. **Not end-to-end:** SVMs and regressors are post-hoc, CNN features not updated in response to SVMs and regressors
3. **Complex** multistage training pipeline

R. Girshick, J. Donahue, T. Darrell, and J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, CVPR 2014.
e.g. region proposal: J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, Selective Search for Object Recognition, IJCV 2013

Slides modified from Ross Girshick tutorial at CVPR 2019

Fast R-CNN [ICCV 2015]

- Small accuracy improvement
- Timing excluding region proposal
  - ~10x faster for training
  - ~100x faster for testing (< 1 sec / image)

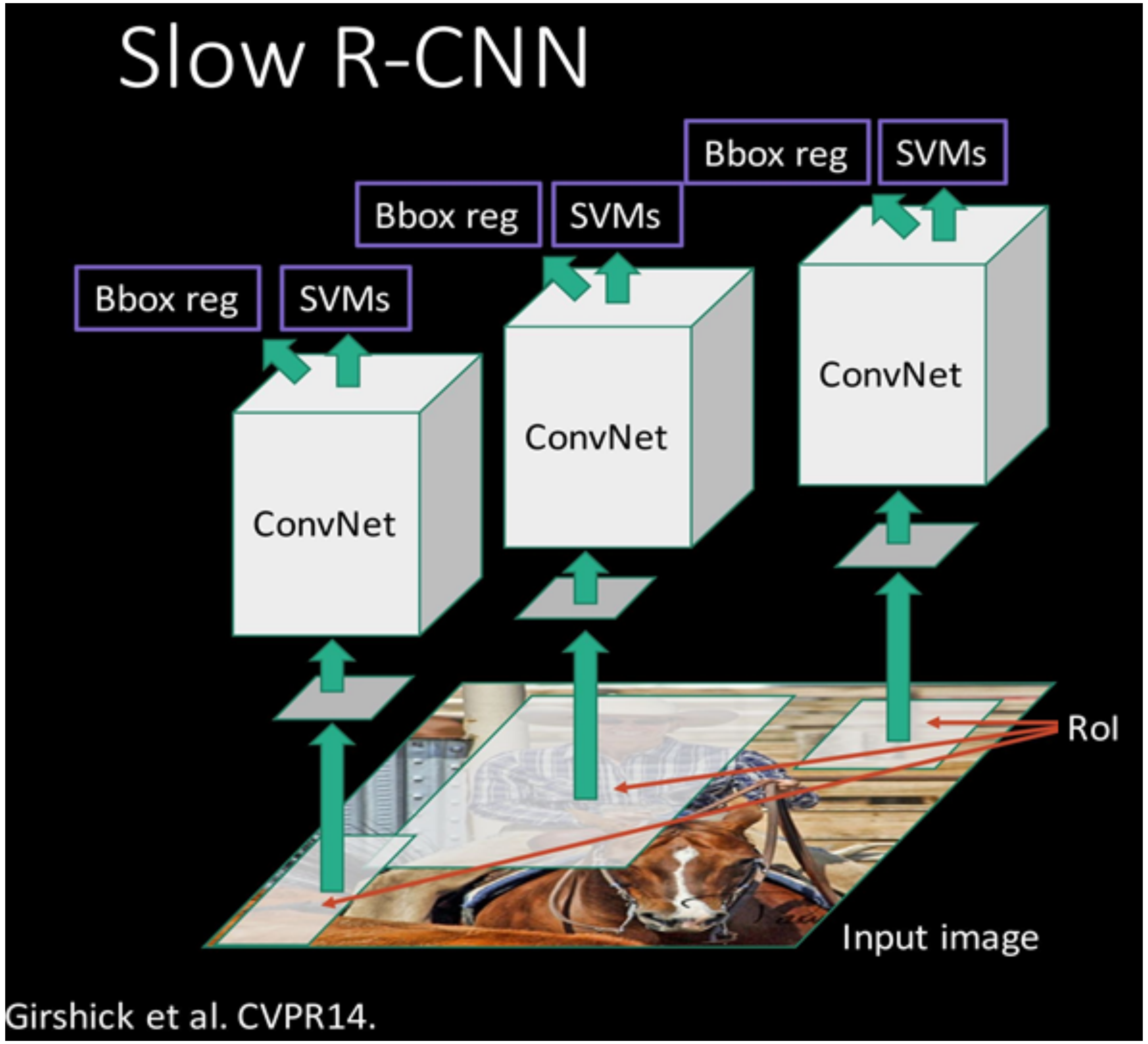


R. Girshick, Fast R-CNN, ICCV 2015

Slides modified from Ross Girshick tutorial at CVPR 2019

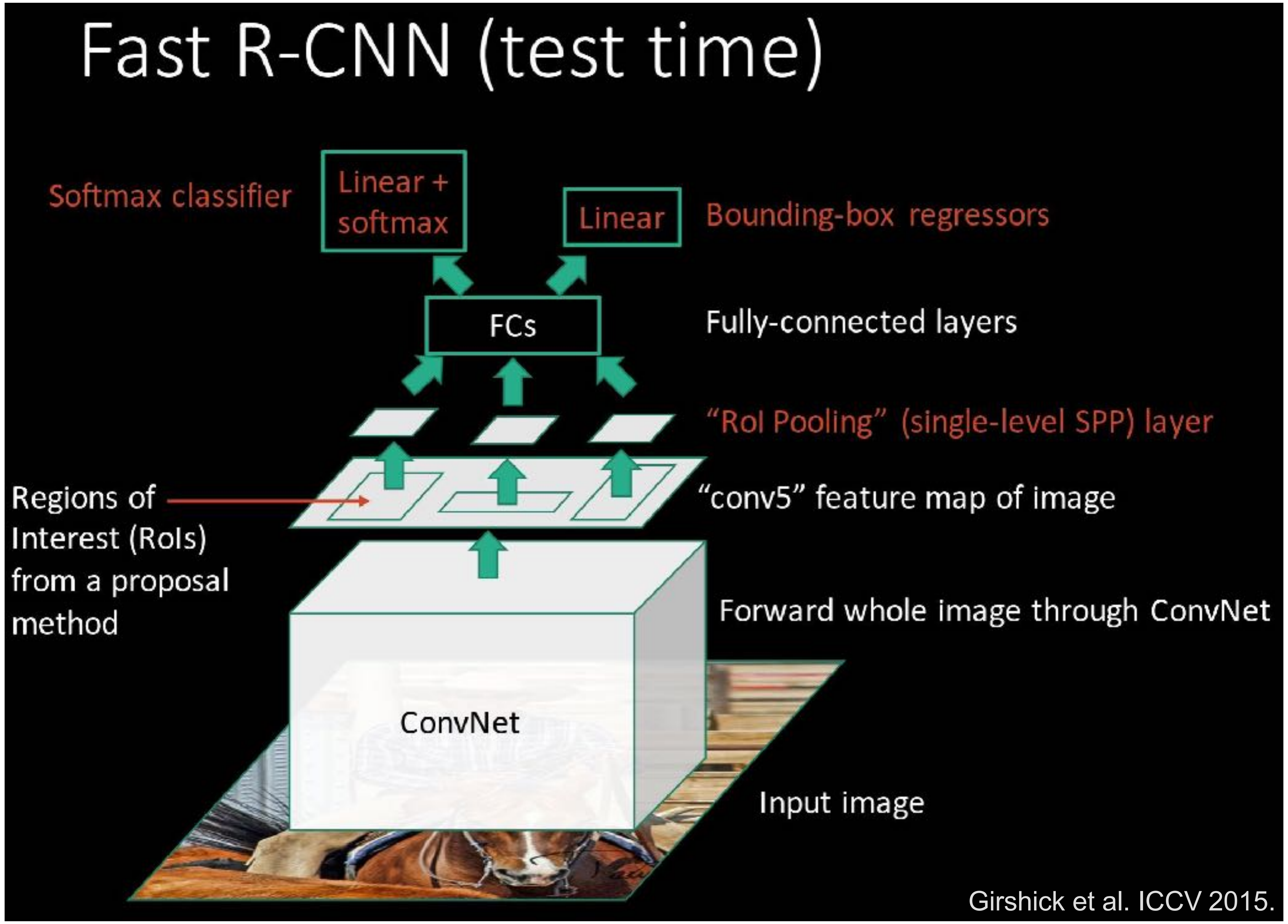
R-CNN Problems

Problem #1: **Slow** at test-time due to **independent forward passes** of the CNN



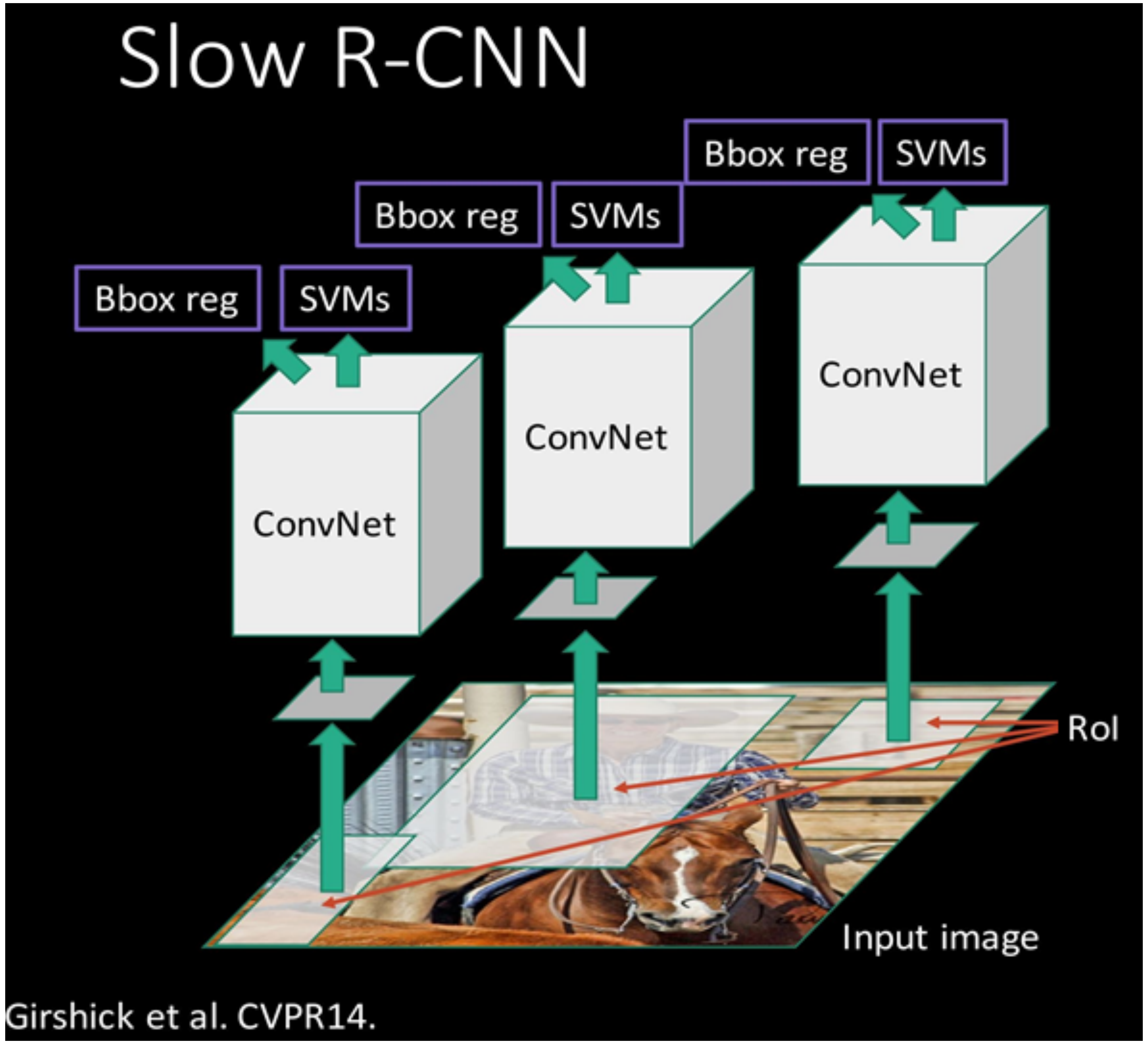
Fast R-CNN Solutions

Solution: **Share computation of convolutional layers** between proposals for an image



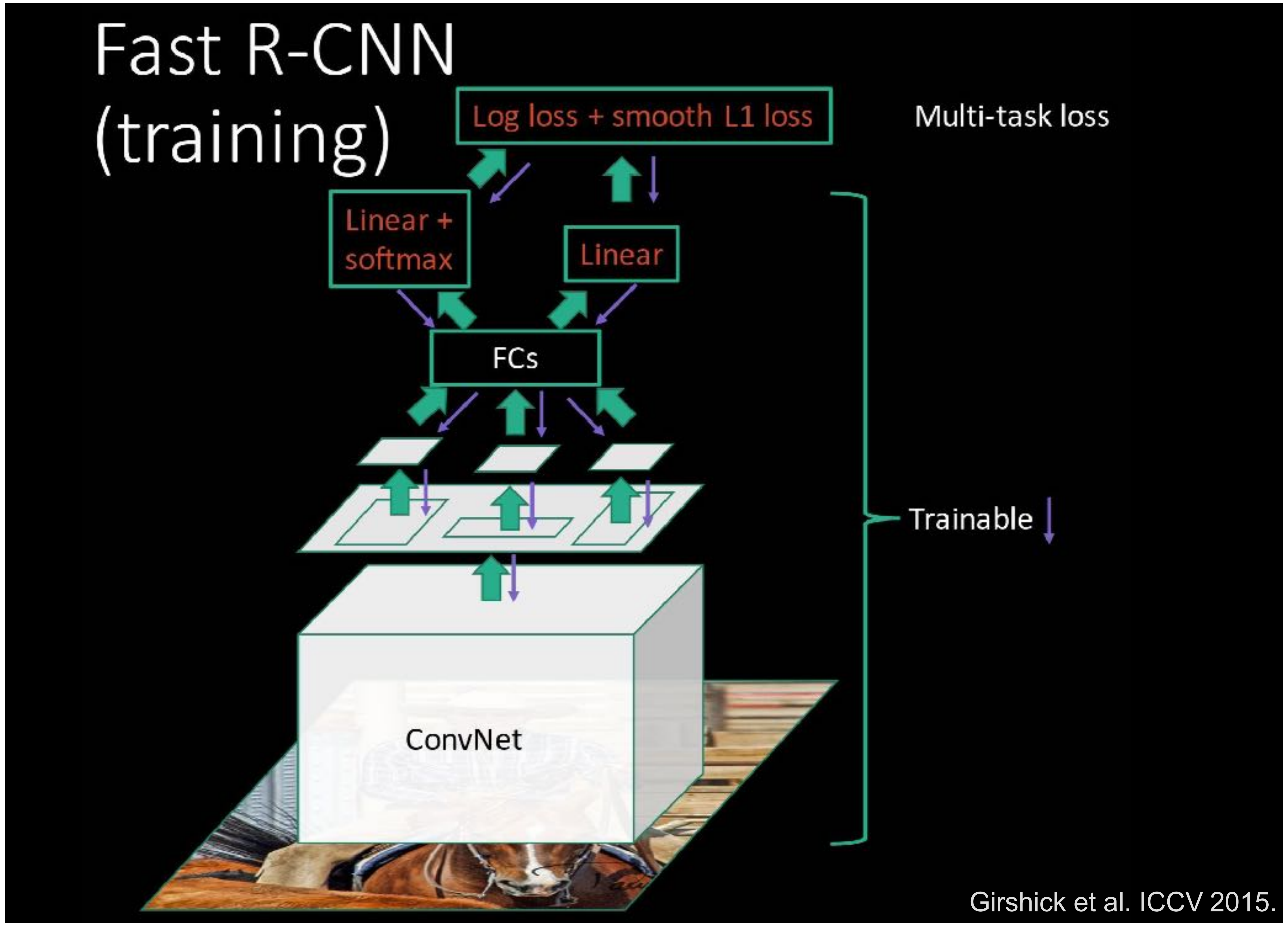
R-CNN Problems

Problem #2: Post-hoc training: **CNN not updated** in response to final classifiers and regressors.
Problem #3: **Complex** training pipeline.



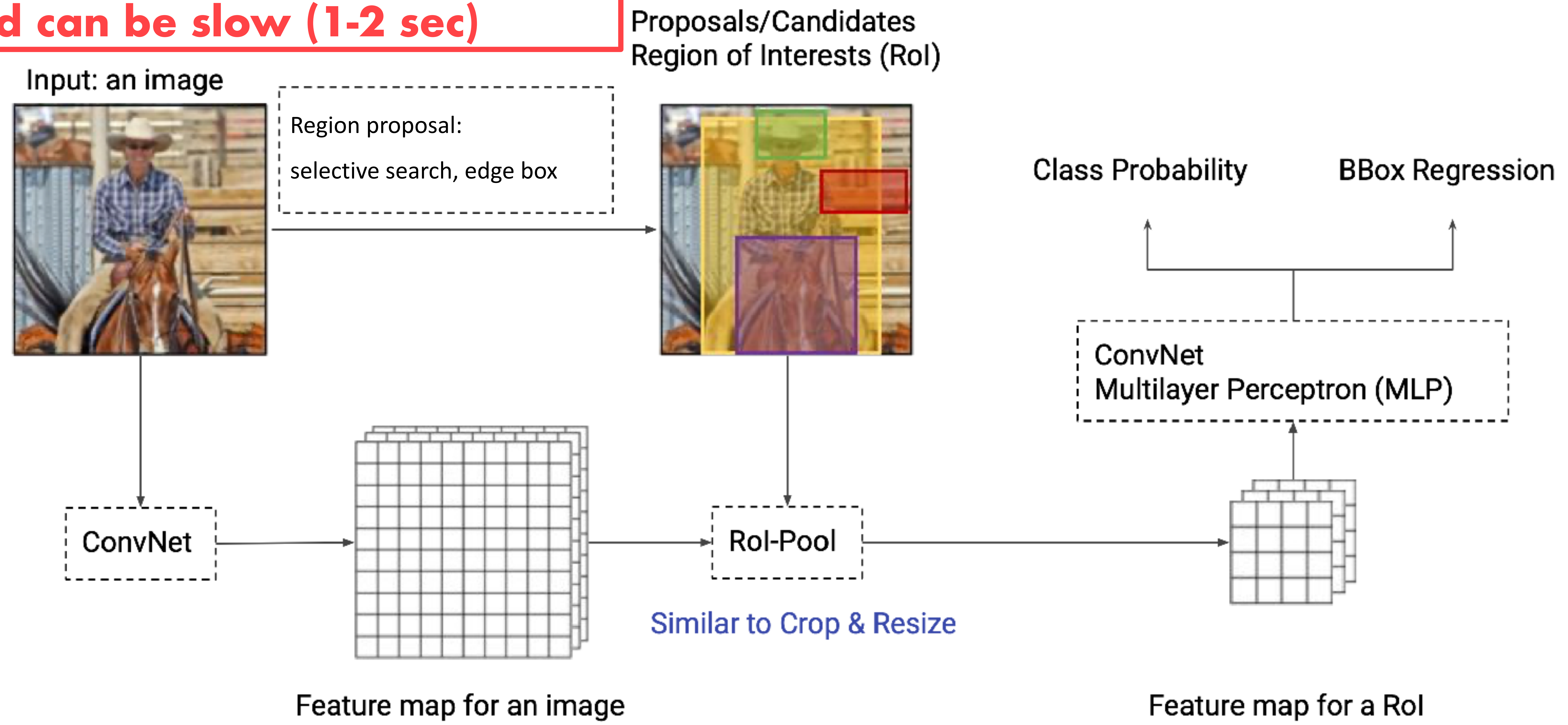
Fast R-CNN Solutions

Solution: **Just train the whole system end-to-end all at once!**



Fast R-CNN [ICCV 2015]

Region proposal is still independent and can be slow (1-2 sec)

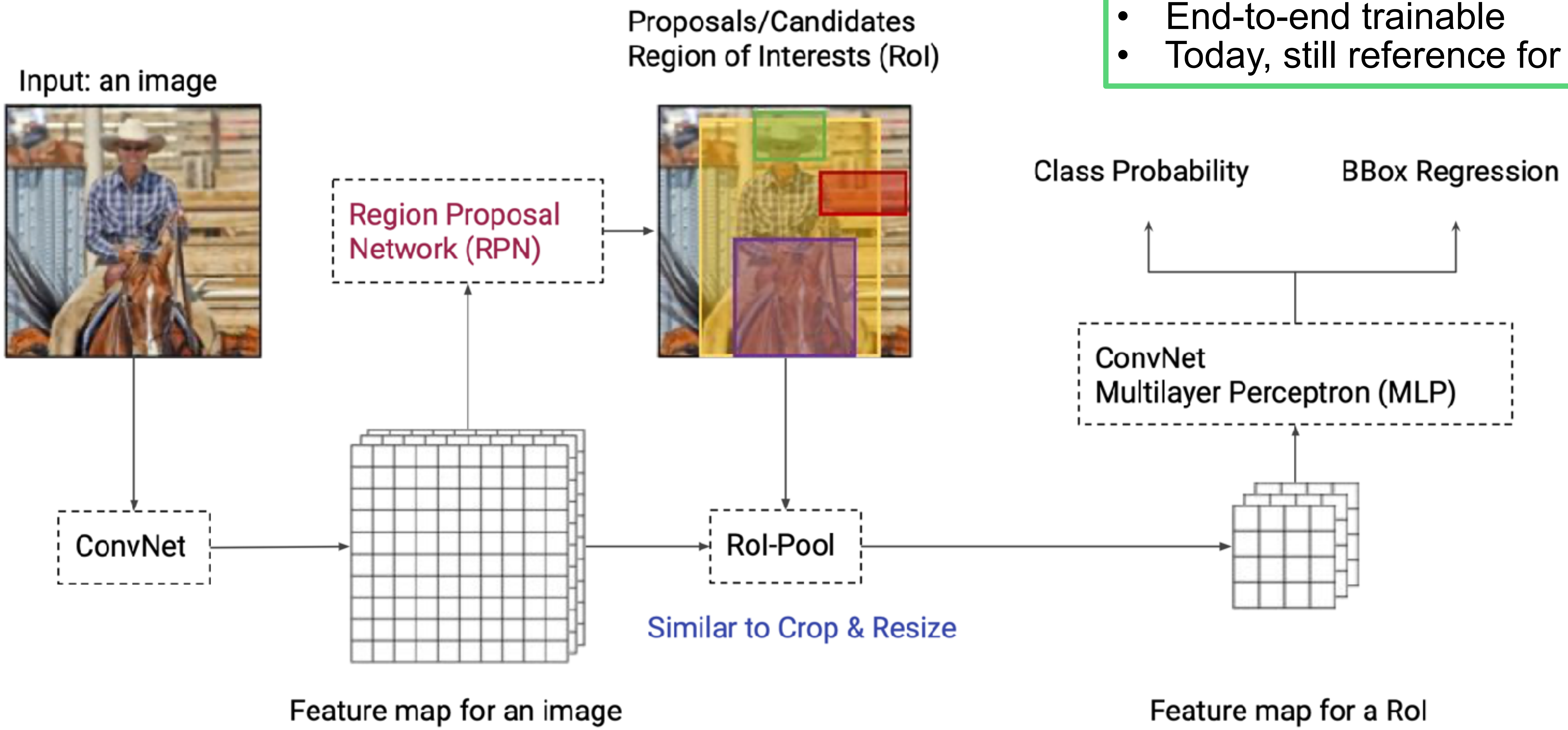


R. Girshick, Fast R-CNN, ICCV 2015

Slides modified from Ross Girshick tutorial at CVPR 2019

Faster R-CNN [NeurIPS 2015]

- Clear boost in performance
- ~ 0.2 sec / image
- End-to-end trainable
- Today, still reference for detection

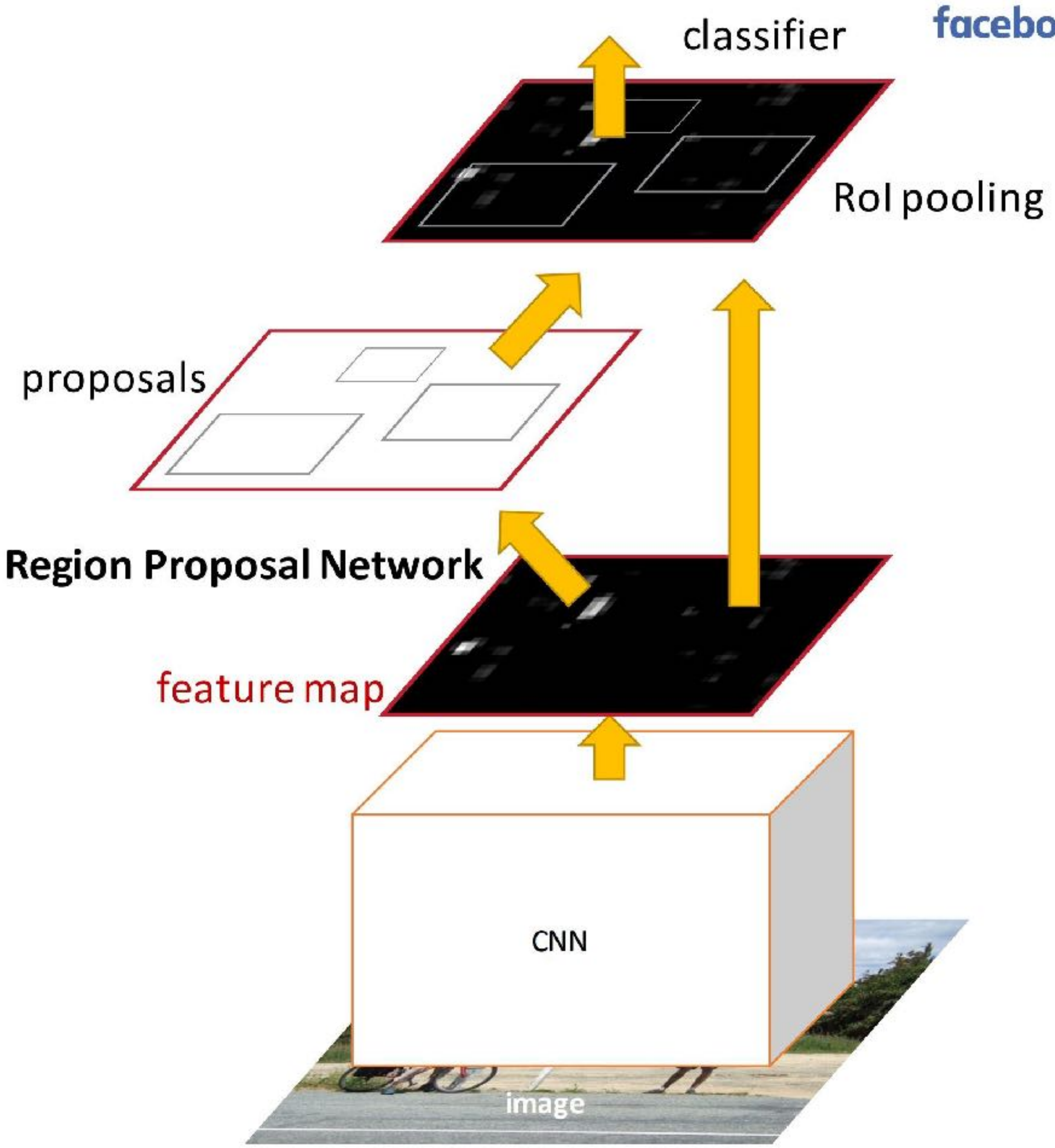


S. Ren, K. He, R. Girshick, and J. Sun, [Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks](), NeurIPS 2015

Slides modified from Ross Girshick tutorial at CVPR 2019

Faster R-CNN

- Insert a **Region Proposal Network (RPN)** after the 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 bbox regressor just like Fast R-CNN.



Slide credit: Ross Girschick

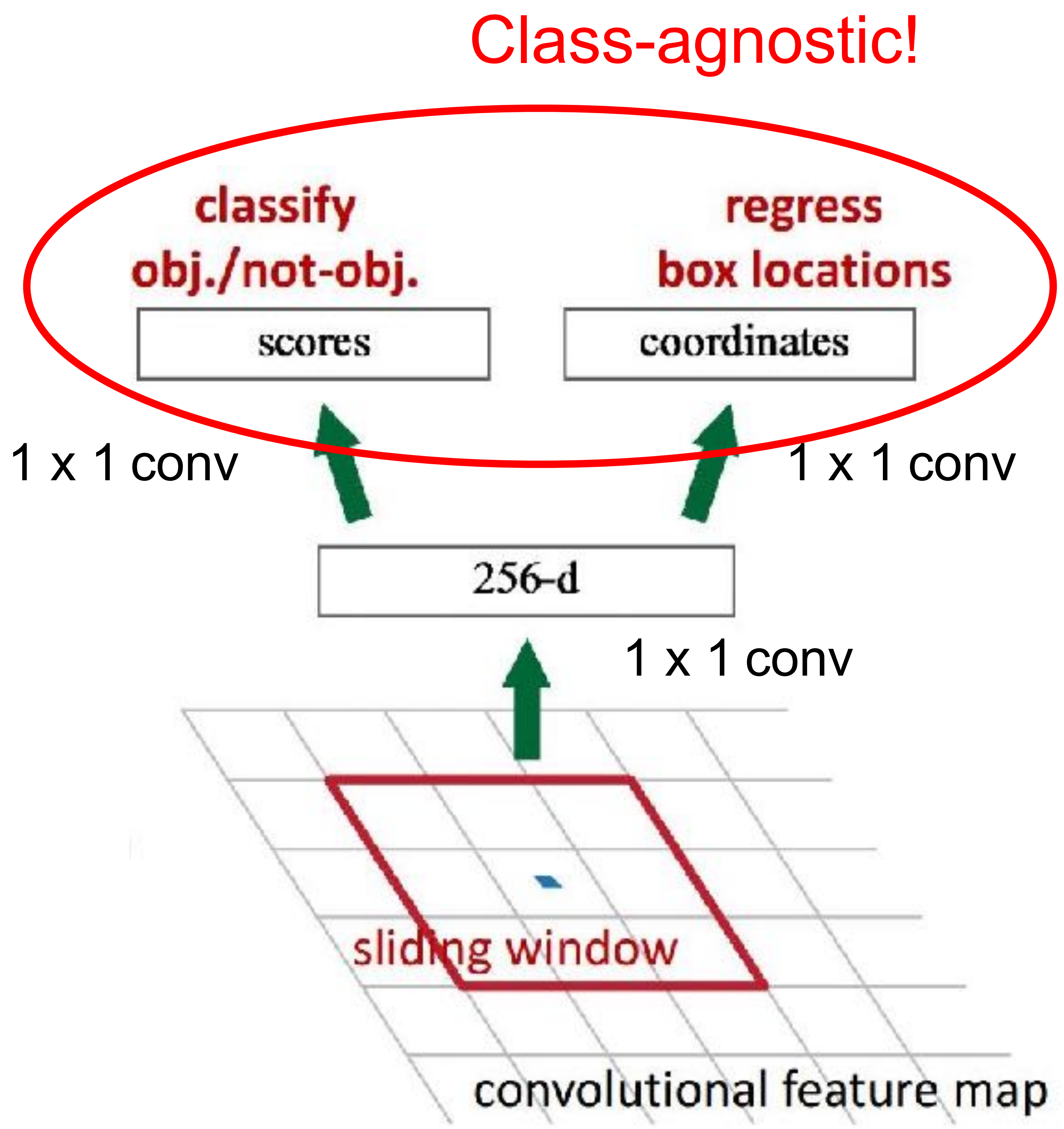
RPN: Region Proposal Network

Slide a small window on the feature map

Build a small network for:
- classifying object or not-object, and
- regressing bbox locations

Position of the sliding window provides localization information with reference to the image

Box regression provides finer localization information with reference to this sliding window



Slide credit: Kaiming He

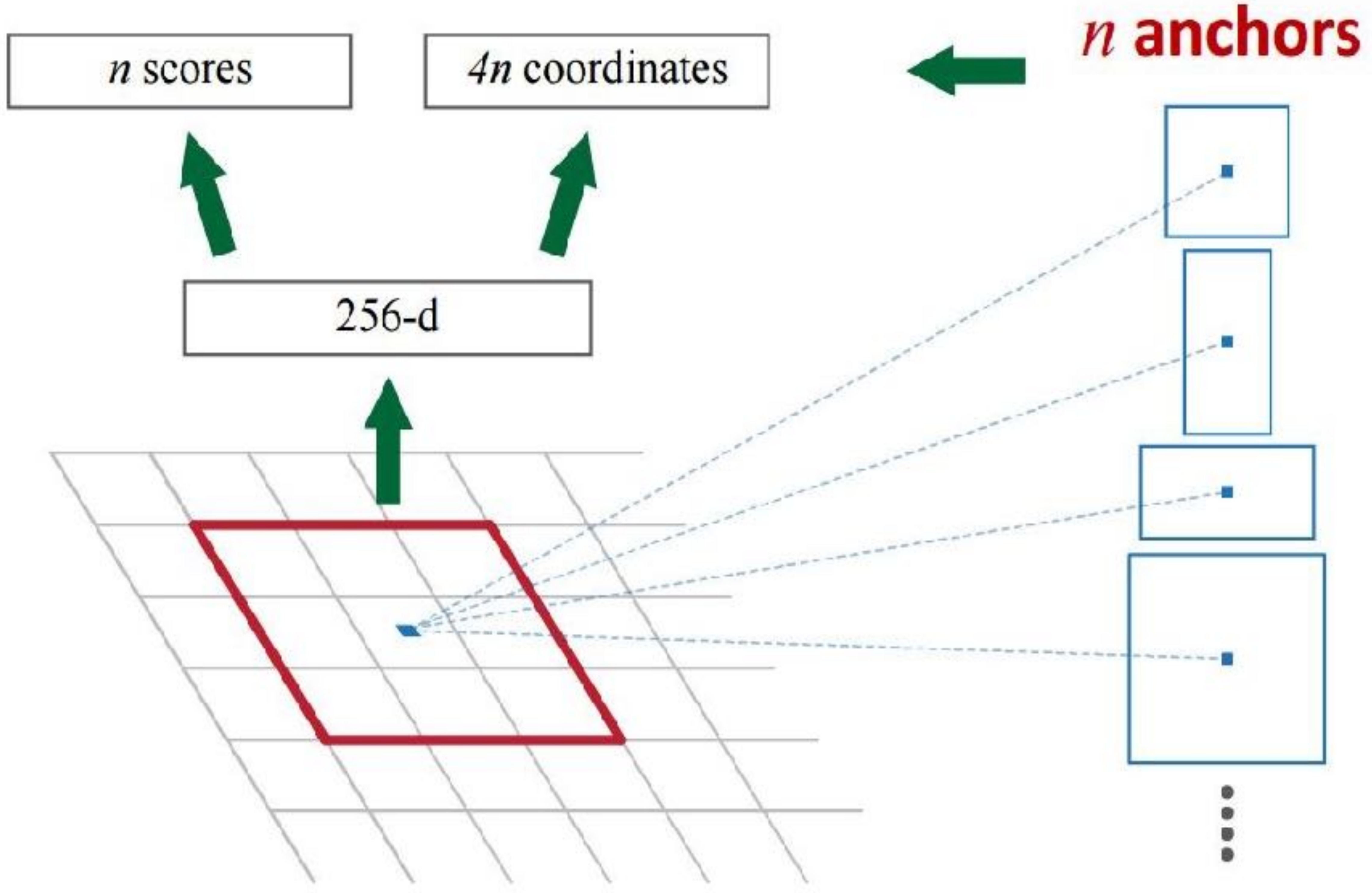
RPN: Region Proposal Network

Use **N anchor boxes** at each location.

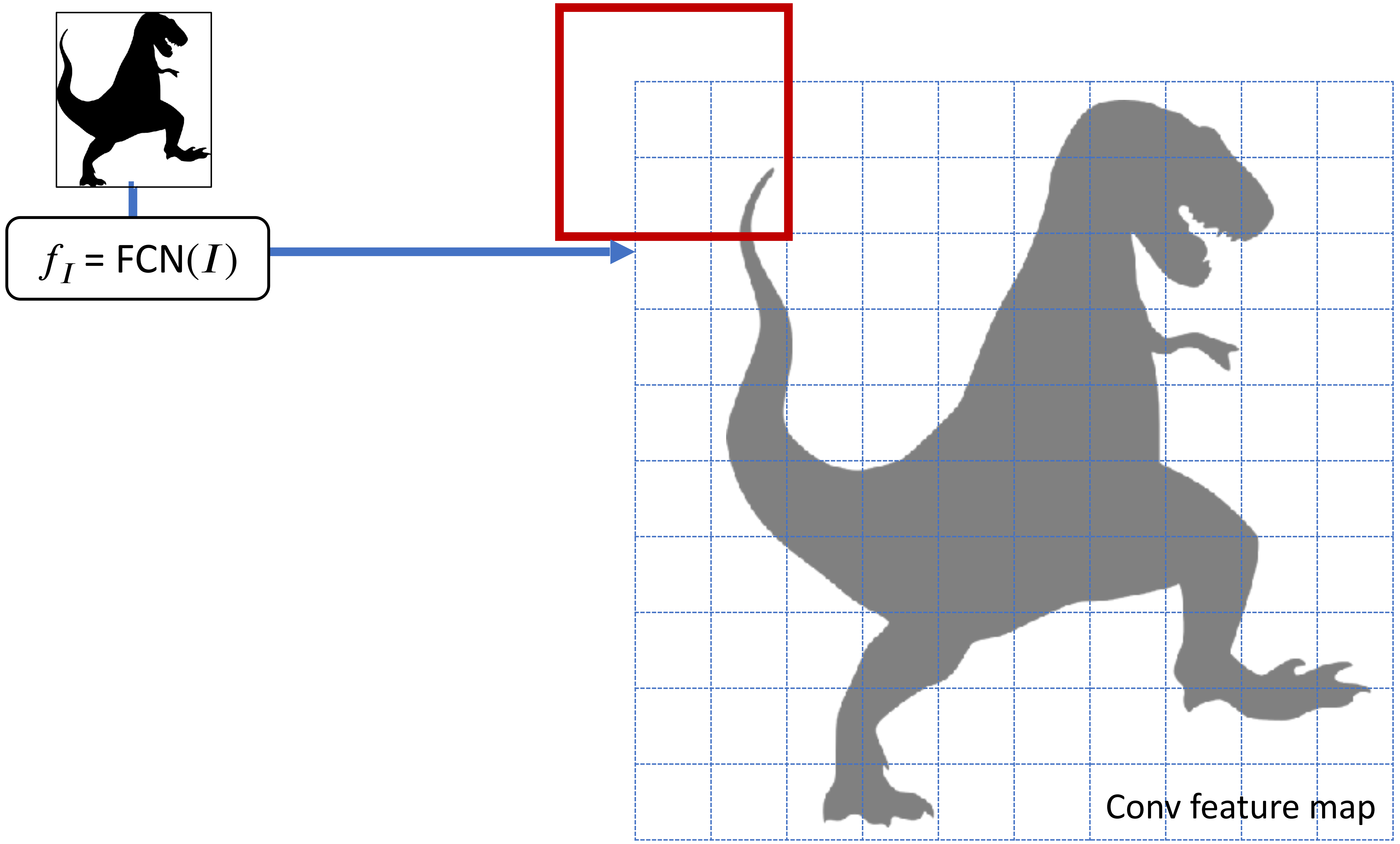
Anchors are **translation invariant**: use the same ones at every location.

Regression gives offsets from anchor boxes.

Classification gives the probability that each (regressed) anchor shows an object.



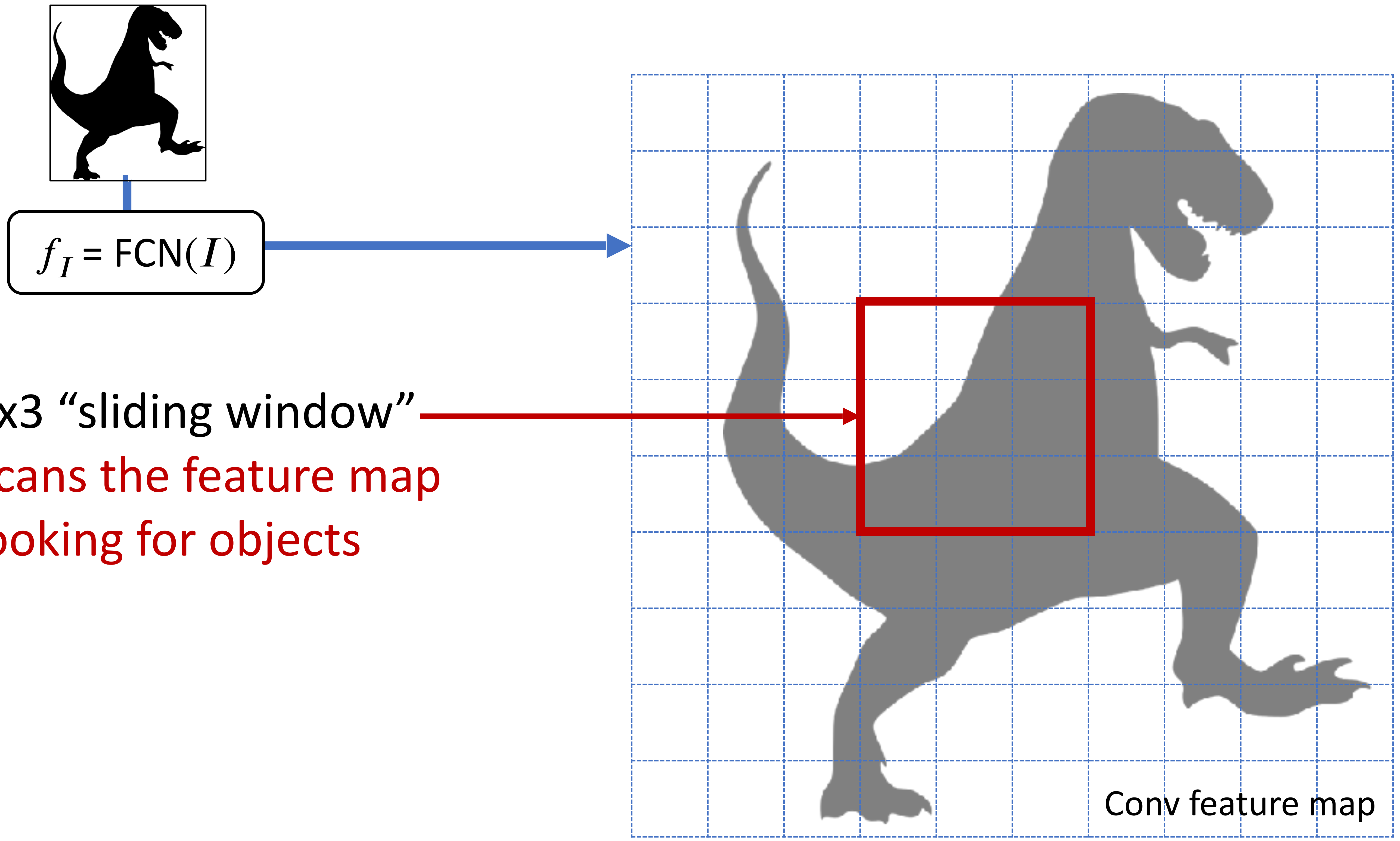
RPN: Region Proposal Network



Feature Map : 16 x 16 x 256

Slide credit: Ross Girschick

RPN: Region Proposal Network



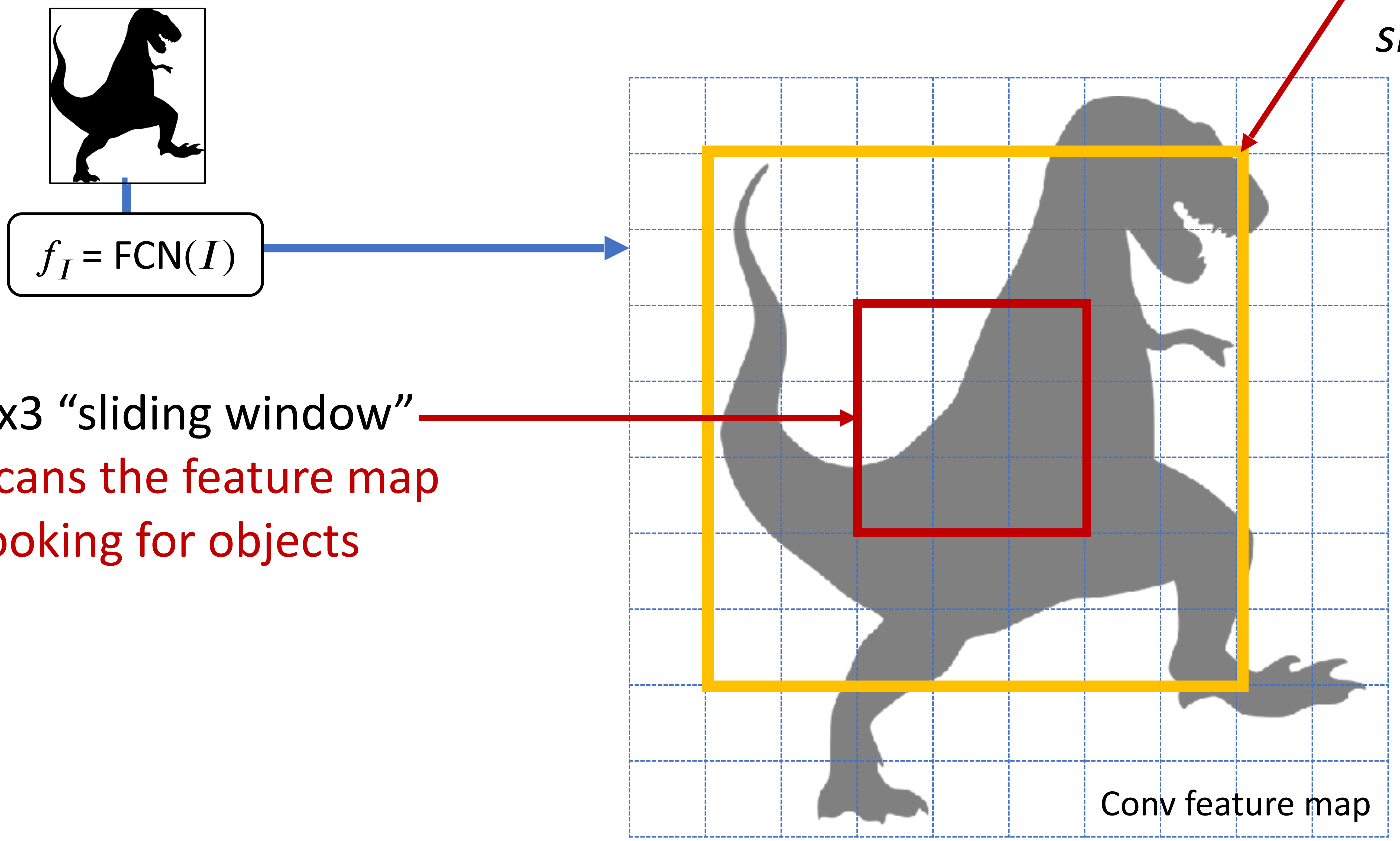
3x3 "sliding window"
Scans the feature map
looking for objects

Feature Map : 16 x 16 x 256

Slide credit: Ross Girschick

RPN: Anchor Box

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

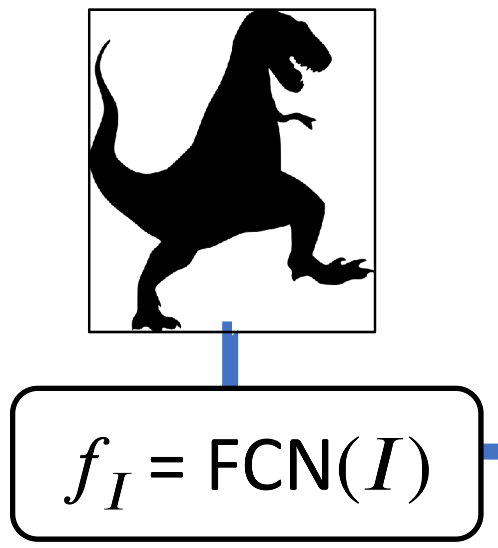


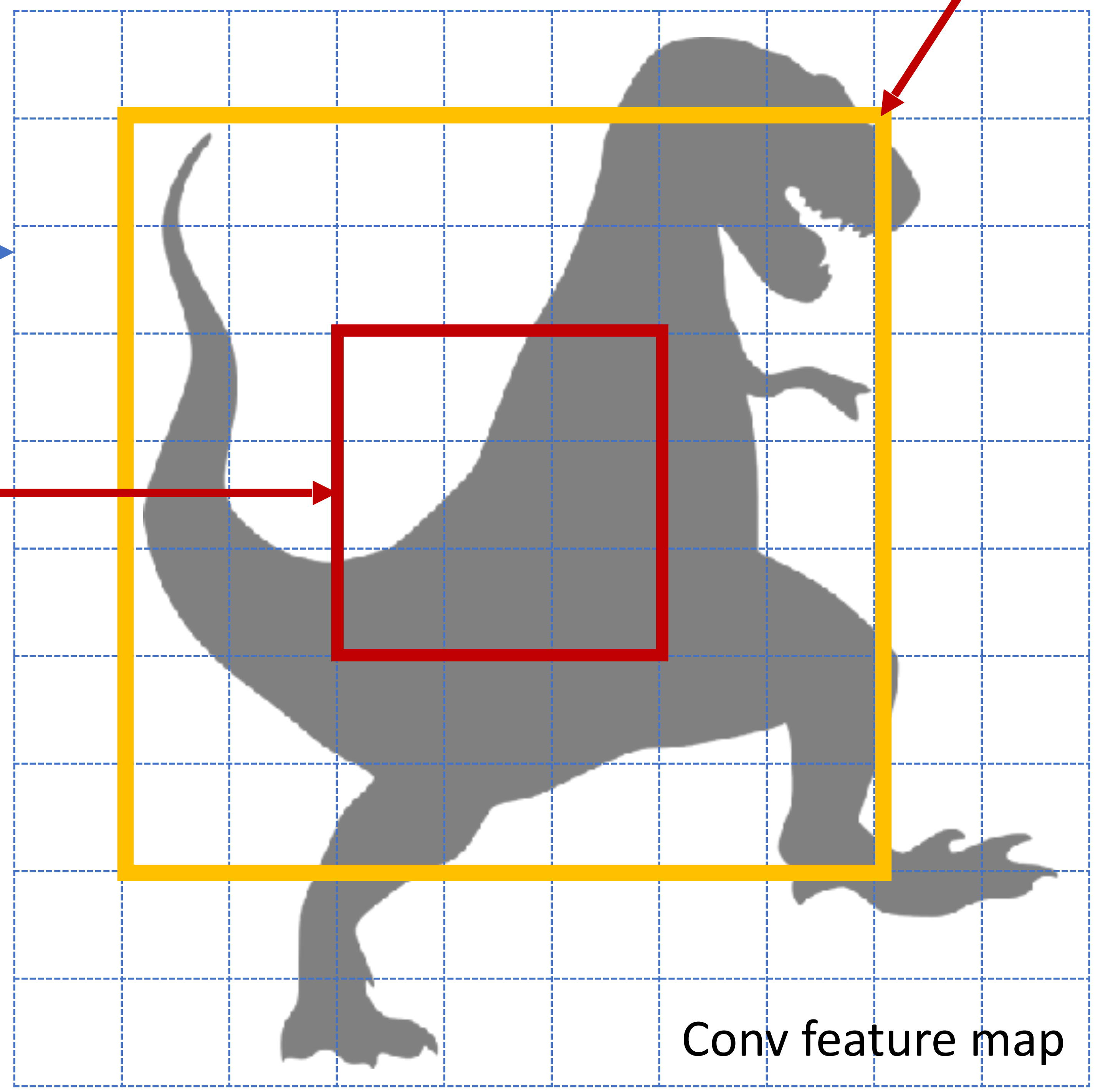
3x3 "sliding window"
Scans the feature map looking for objects

Feature Map : 16 x 16 x 256

Slide credit: Ross Girschick

RPN: Anchor Box





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)

Feature Map : 16 x 16 x 256

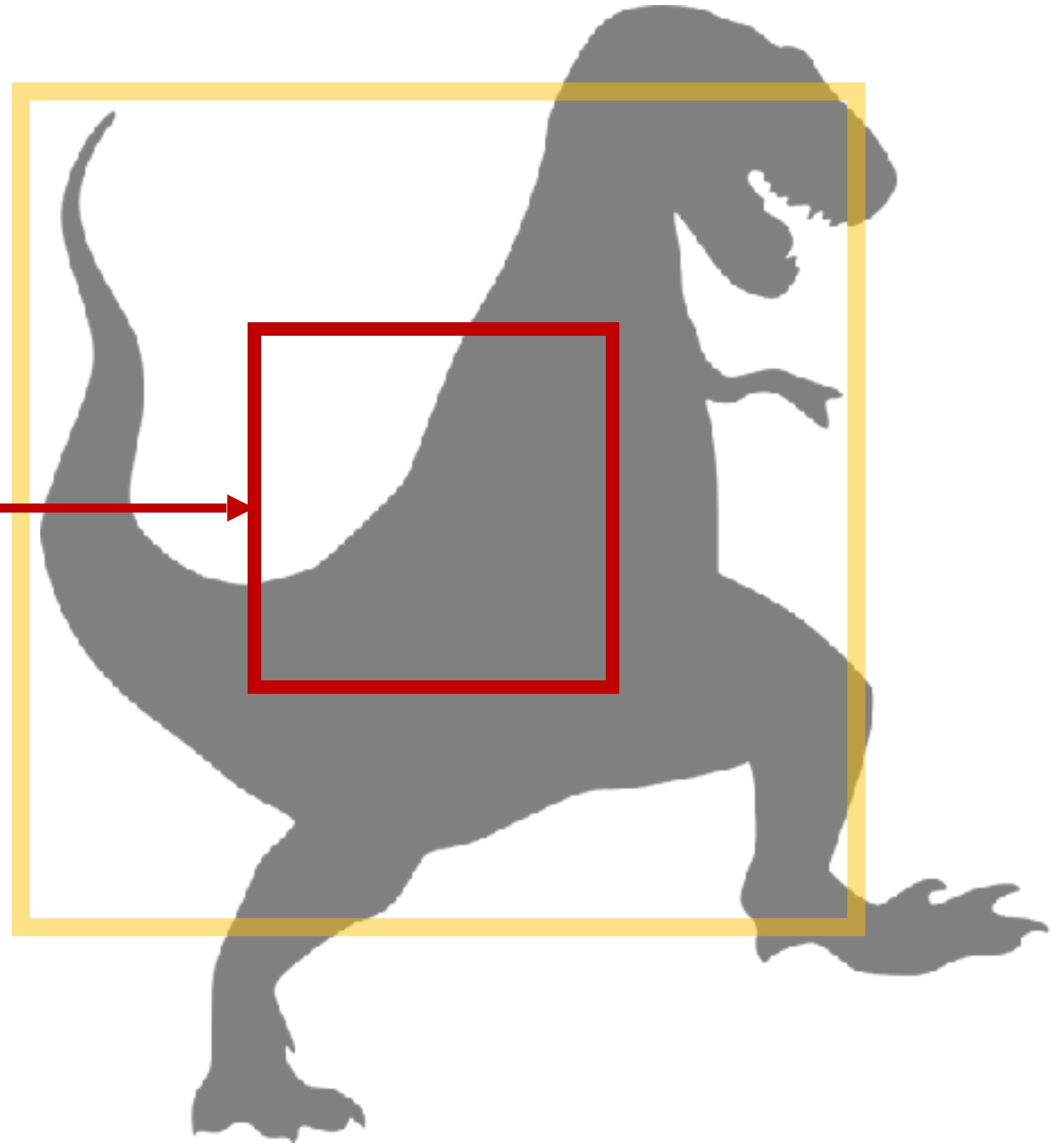
Slide credit: Ross Girschick

RPN: Prediction (on object)

the probability that each (regressed) anchor shows an object

Objectness score

P(object) = 0.94

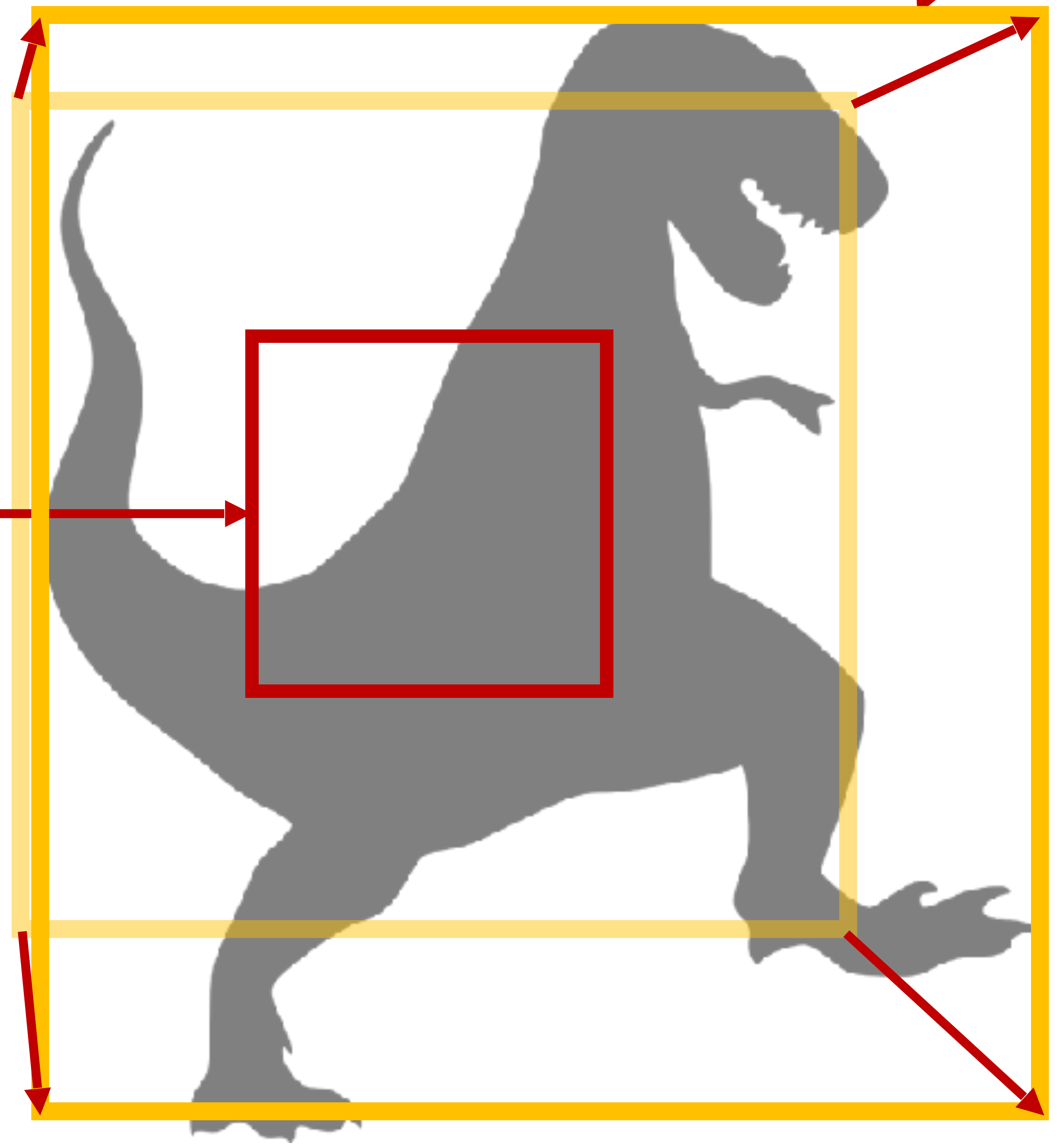


3x3 "sliding window"
- ➢ Objectness classifier [0, 1]
- ➢ Box regressor predicting (dx, dy, dh, dw)

Slide credit: Ross Girschick

RPN: Prediction (on object)

Anchor box: transformed by box regressor



3x3 “sliding window”

➢ Objectness classifier [0, 1]

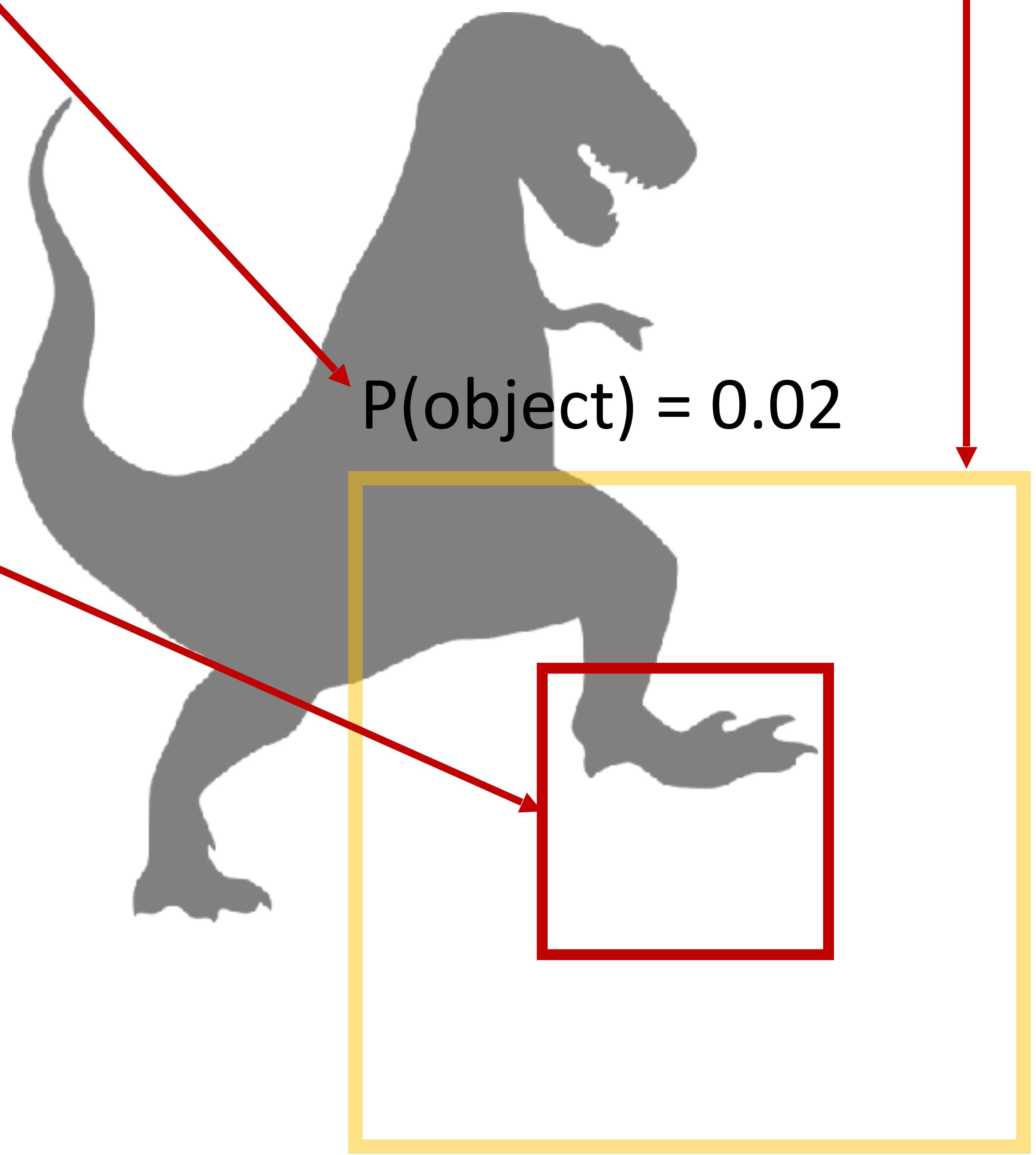
➢ Box regressor predicting (dx, dy, dh, dw)

Slide credit: Ross Girschick

RPN: Prediction (off object)

Anchor box: transformed by box regressor

Objectness score



3x3 "sliding window"

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

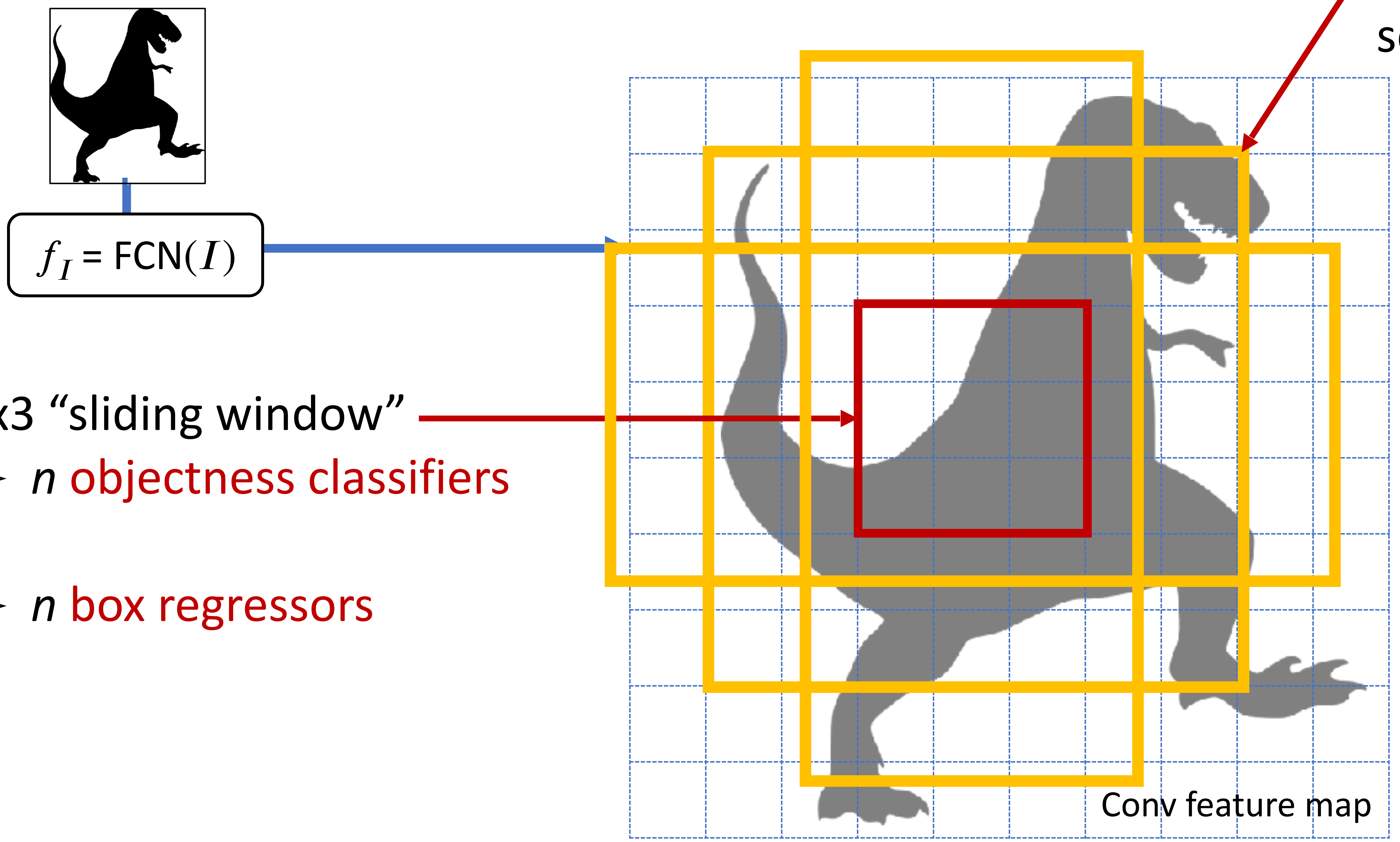
Slide credit: Ross Girschick

RPN: Multiple Anchors

Y (ground truth) : [1 1 1 1 1 1 0 0 0 0 1 0 0 0 0]

M (mask) : [1 dx dy dh dw 0 - - - - 0 - - - -]

$$Loss = \sum_i M_i \,.\, \mathscr{L}(\hat{Y}_i, Y_i)$$



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

3x3 "sliding window"

- *n* objectness classifiers
- *n* box regressors

Feature Map : 16 x 16 x 256

$\hat{Y}$ (output): 16 x 16 x *n* x (1+4)

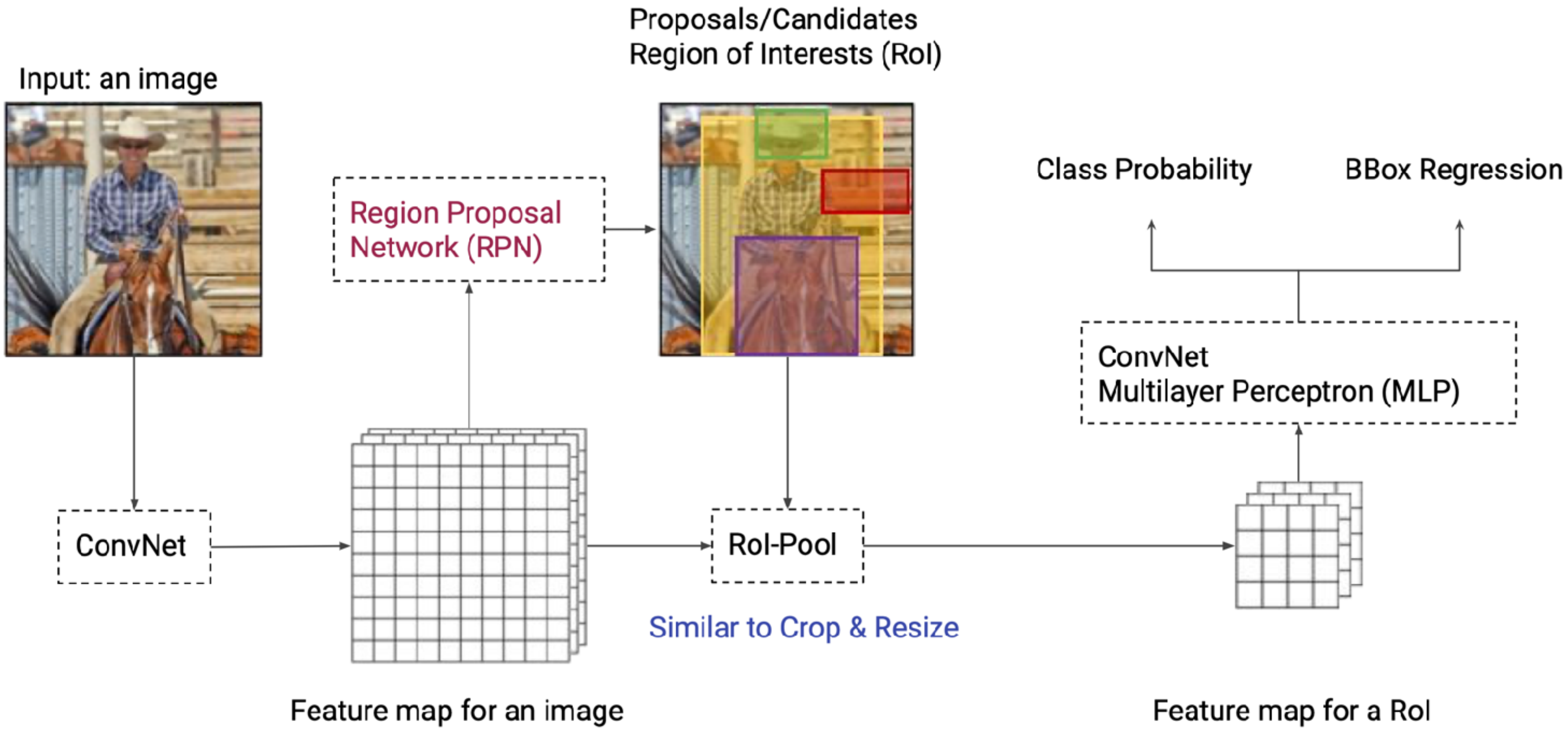
box regression

objectness

#anchors

Faster R-CNN

- Still two-stage



S. Ren, K. He, R. Girshick, and J. Sun, [Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks](), NeurIPS 2015

Slides modified from Ross Girshick tutorial at CVPR 2019

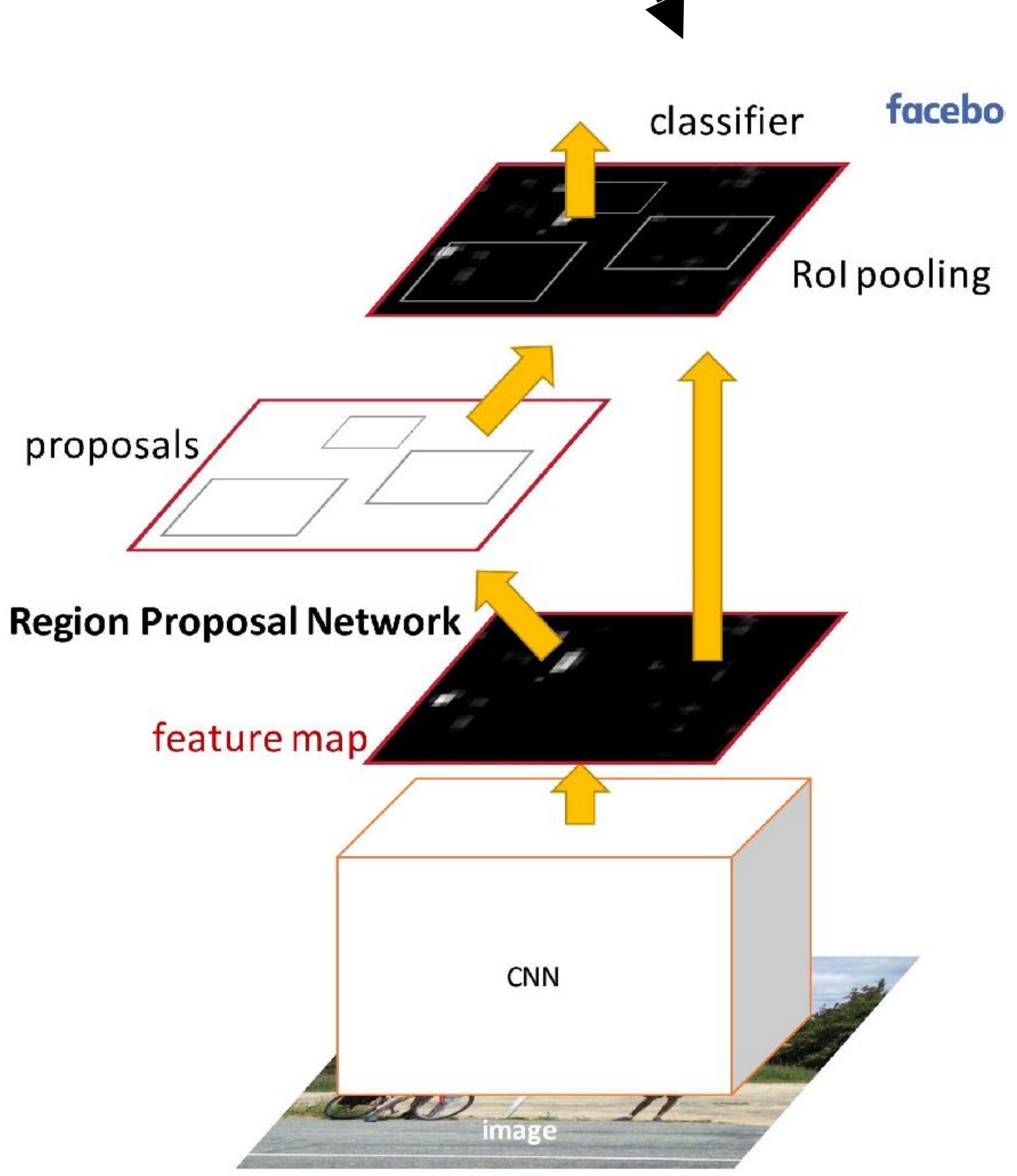
Object detection: 2-stage vs 1-stage

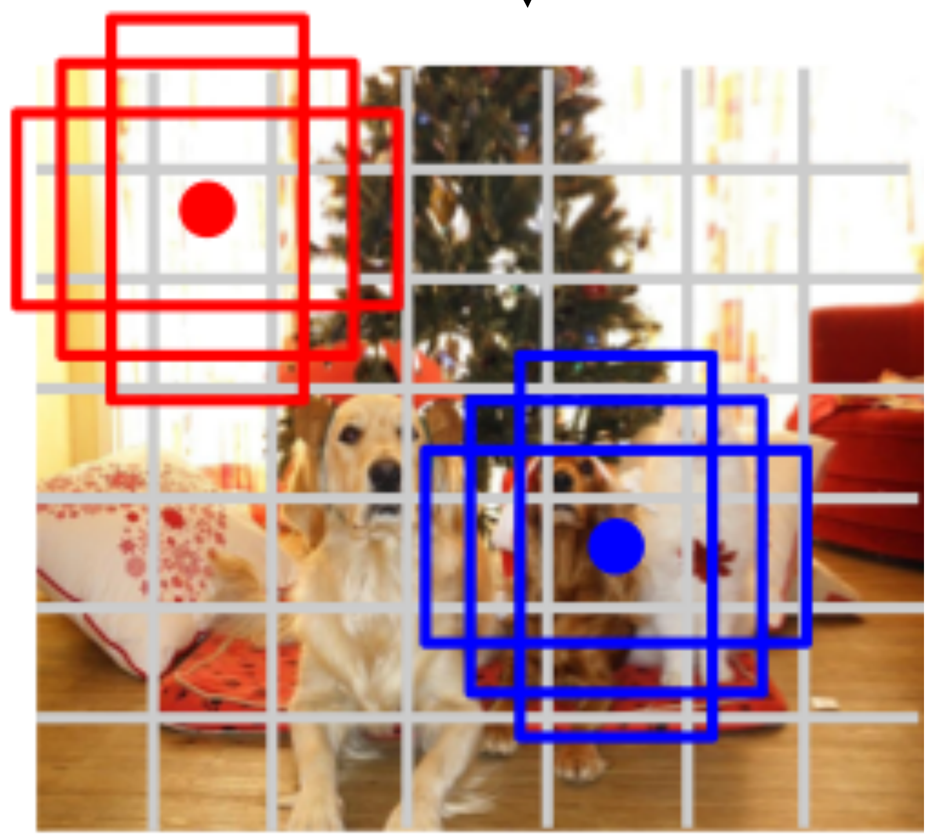
CNN Detectors

2-stage detectors

1-stage detectors

Anchor-free detectors





- No object proposals
- Use **anchors**
- Faster but less accurate

- Use **points**

Fast(er) RCNN, Mask-RCNN, SNIPPER, PANet, TridentNet

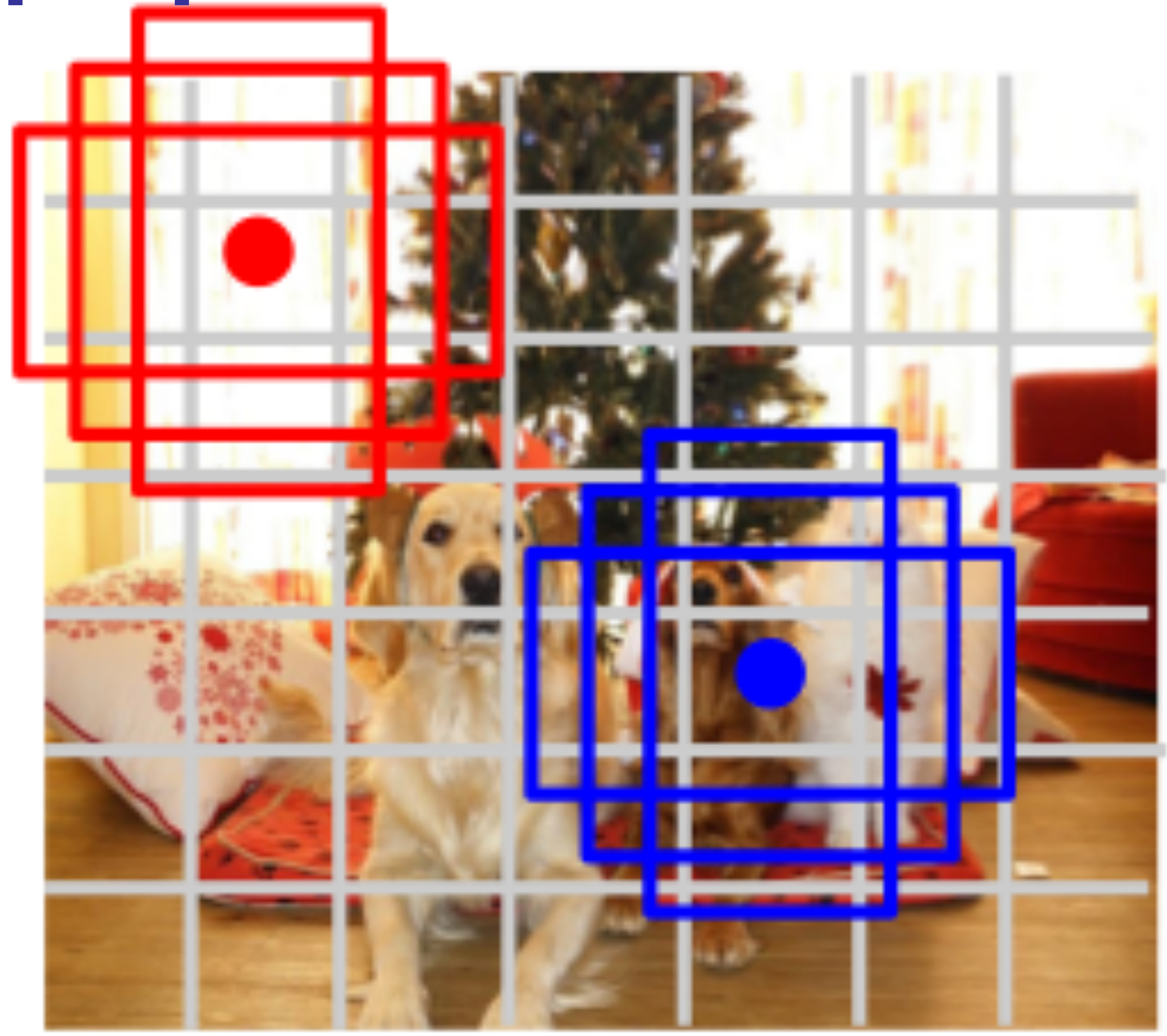
YOLO, SSD, **RetinaNet**, EfficientDet

CornerNet**, CenterNet**, FCOS, ExtremeNet

1-stage object detection: YOLO/SSD

Detection without proposals

Input image
3 x H x W



Divide image into grid
7 x 7

Image a set of **base boxes** centered at each grid cell
Here B = 3

Redmon et al, "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016
Liu et al, "SSD: Single-Shot MultiBox Detector", ECCV 2016

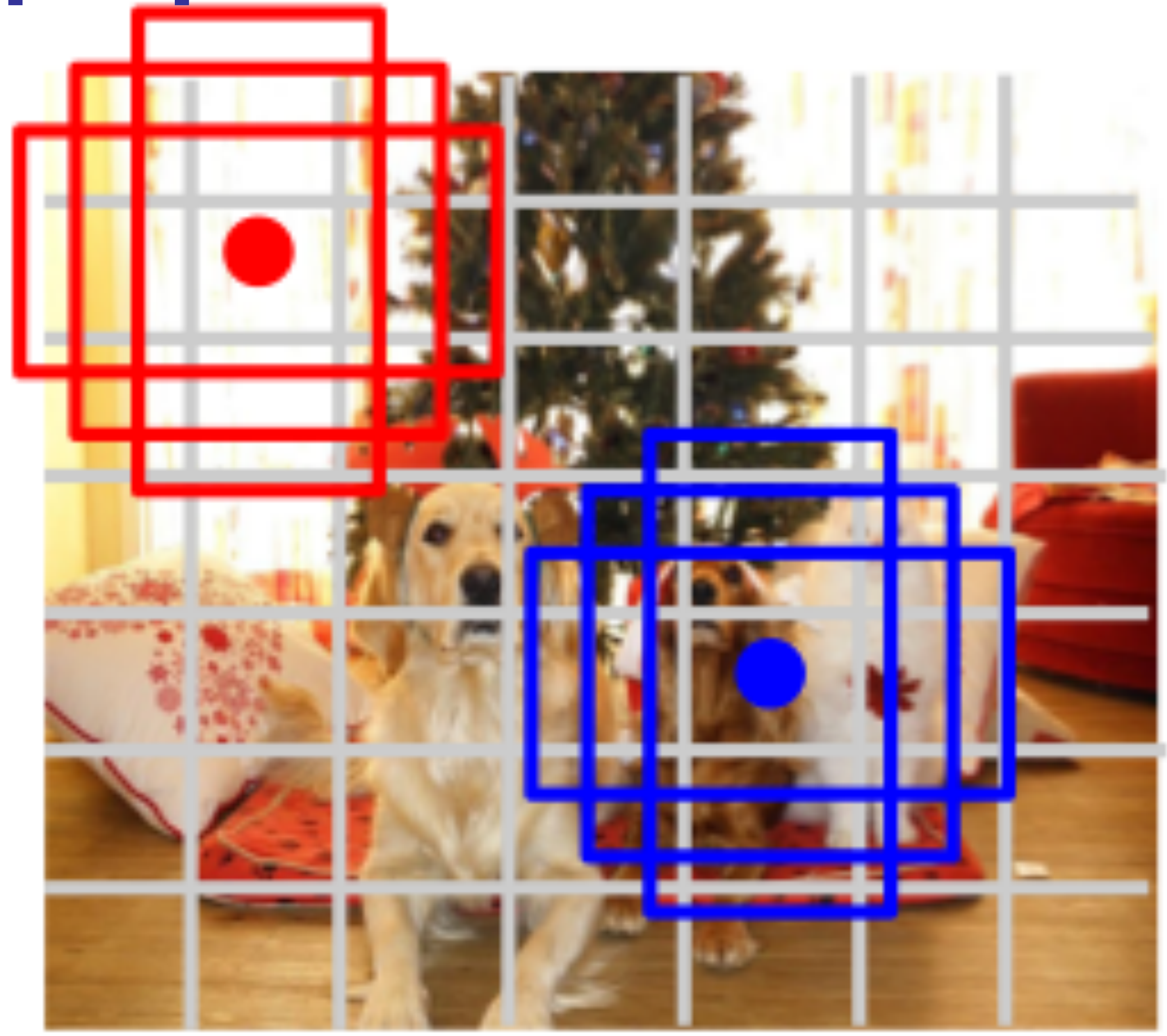
Slide credit: L. Fei Fei, J. Johnson, S. Yeung, http://cs231n.stanford.edu/

1-stage object detection: YOLO/SSD

Detection without proposals

Within each grid cell:
- Regress from each of the B base boxes to a final box with 5 numbers: (dx, dy, dh, dw, confidence)
- Predict scores for each of C classes (including background as a class)

Output:
7 x 7 x (5 * B + C)



Divide image into grid
7 x 7

Image a set of **base boxes** centered at each grid cell
Here B = 3

From input image to scores with a single network. Faster but not as accurate as RCNN.
See also: Lin et al., Focal loss for dense object detection, ICCV 2017.

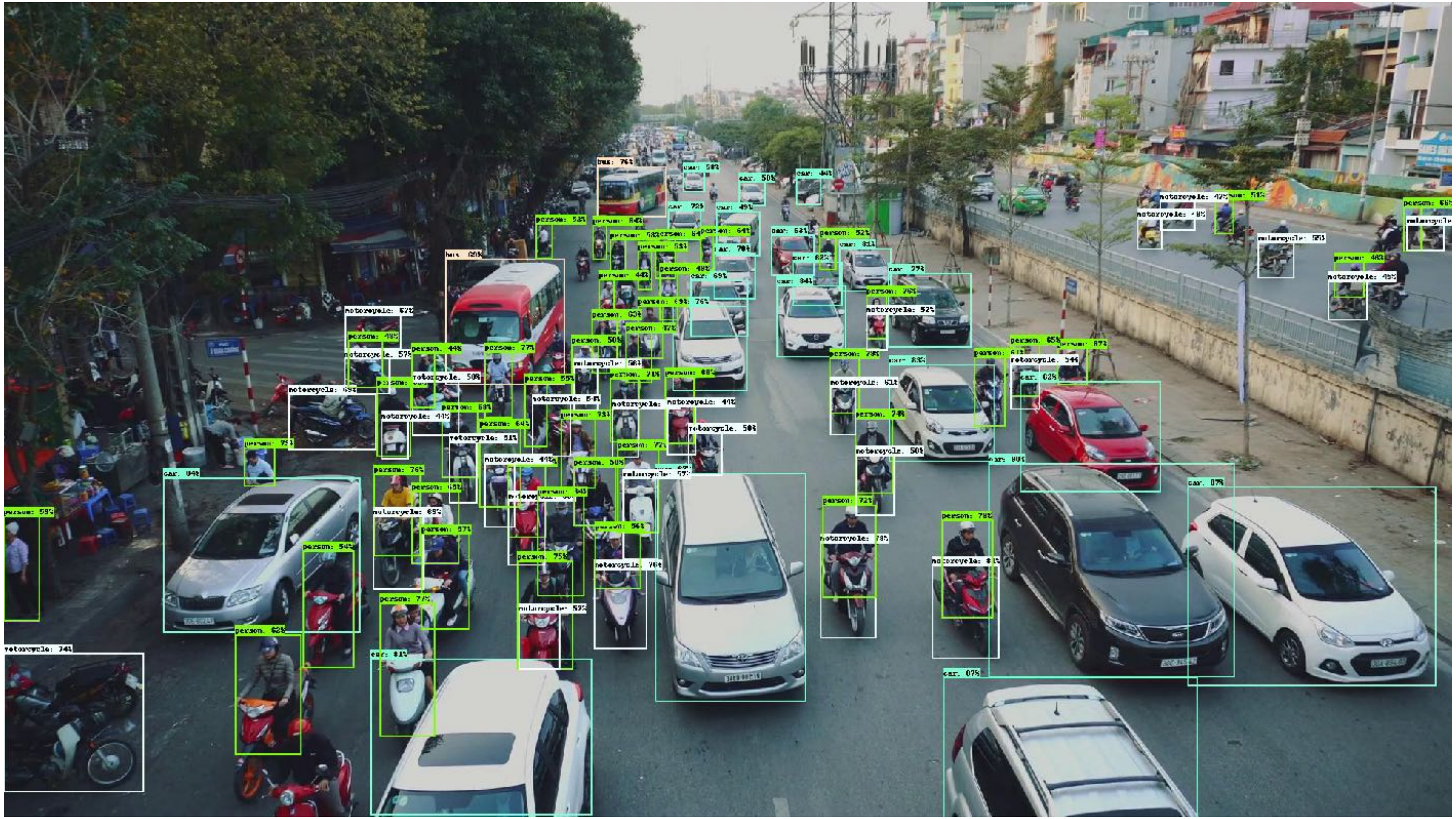
Slide credit: L. Fei Fei, J. Johnson, S. Yeung, http://cs231n.stanford.edu/

Yolo v2 Demo video

[Redmon et al., CVPR'17]

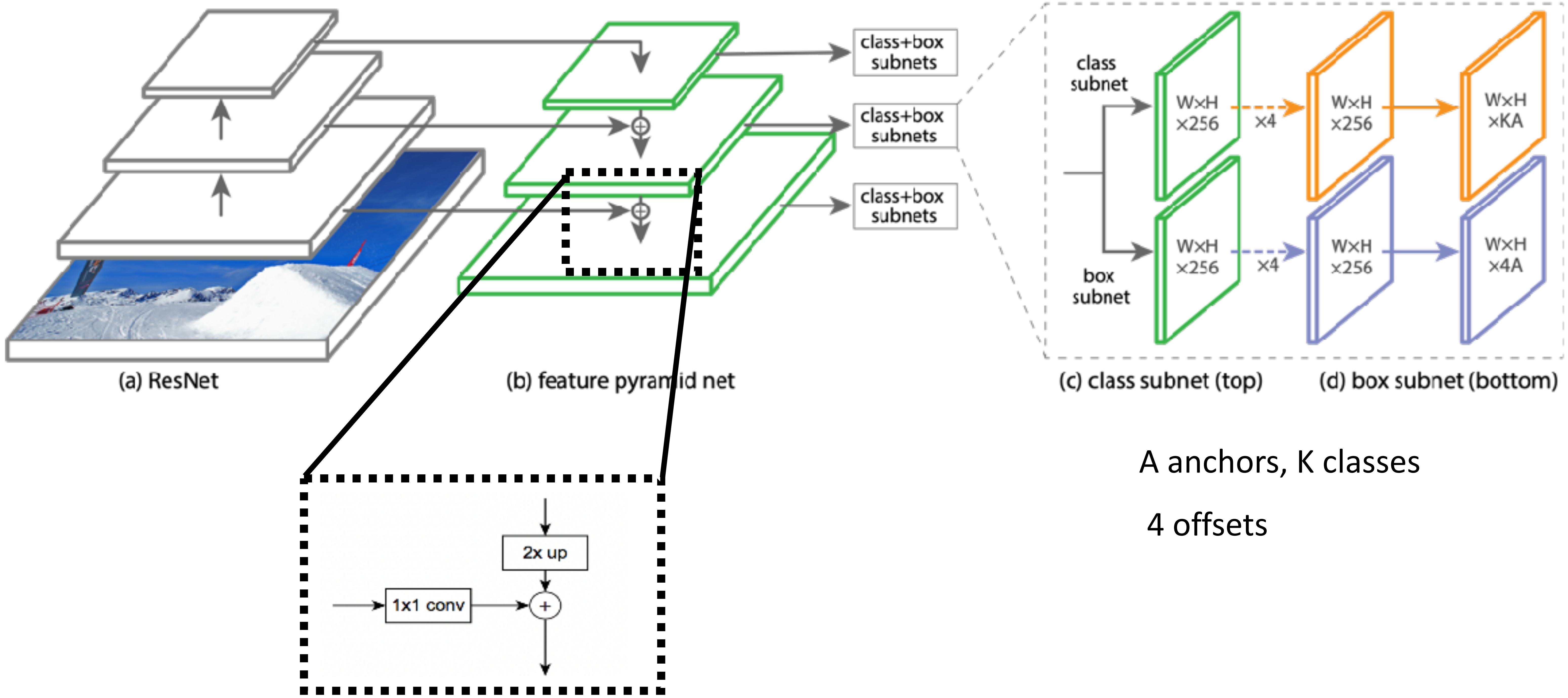
Scale in object detection

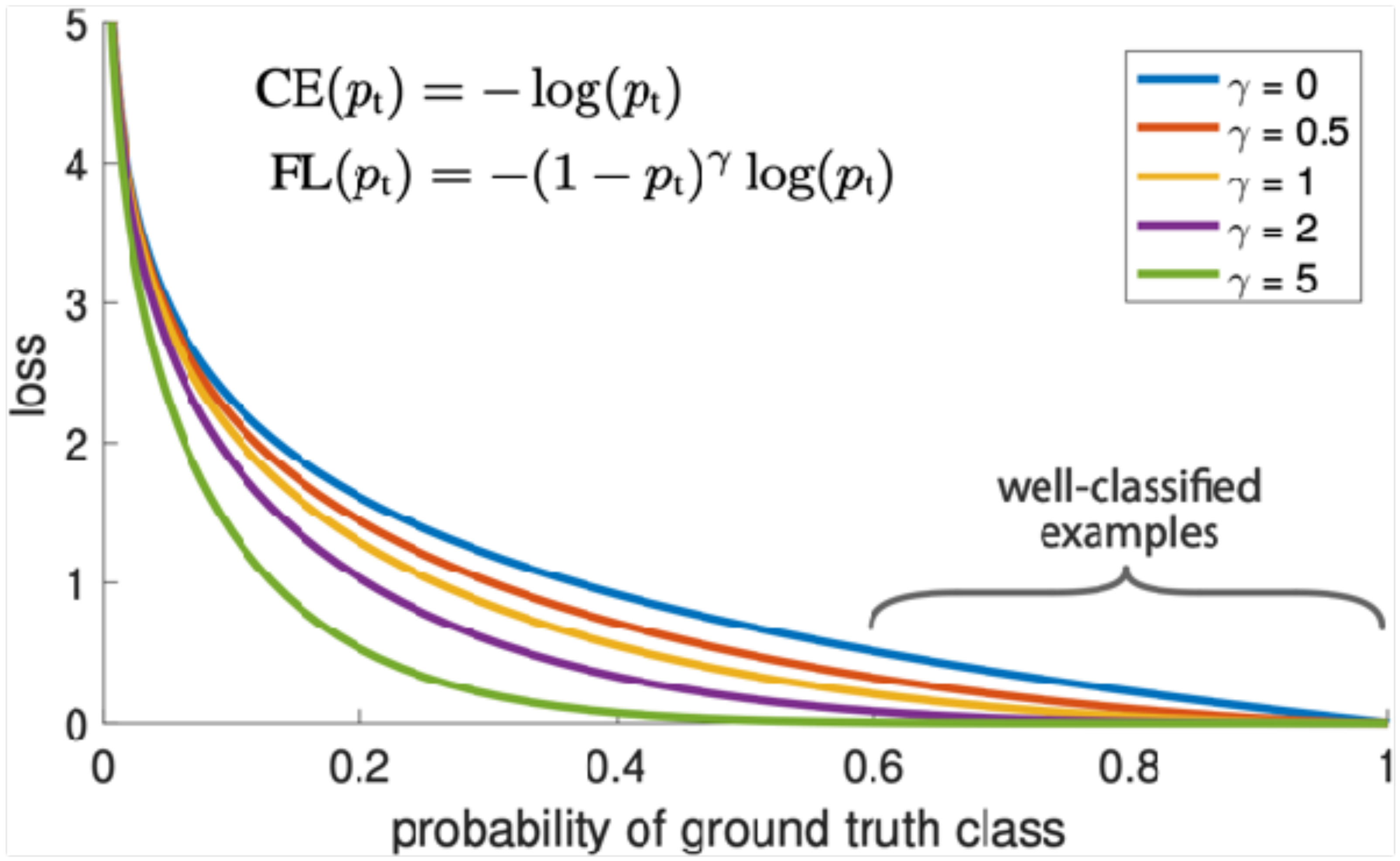
<u>Problem with YOLO:</u> Single cell can corresponds to multiple objects, even with multiple anchors, still **too coarse.**



1-stage object detection: RetinaNet

- Pre-define anchor boxes on **multiple scales**, e.g., Feature Pyramid Networks (FPNs).
- 6 anchors per location, 100 - 200k anchor boxes to classify per image (dense detection).
- Focal loss for soft-version hard sample mining.

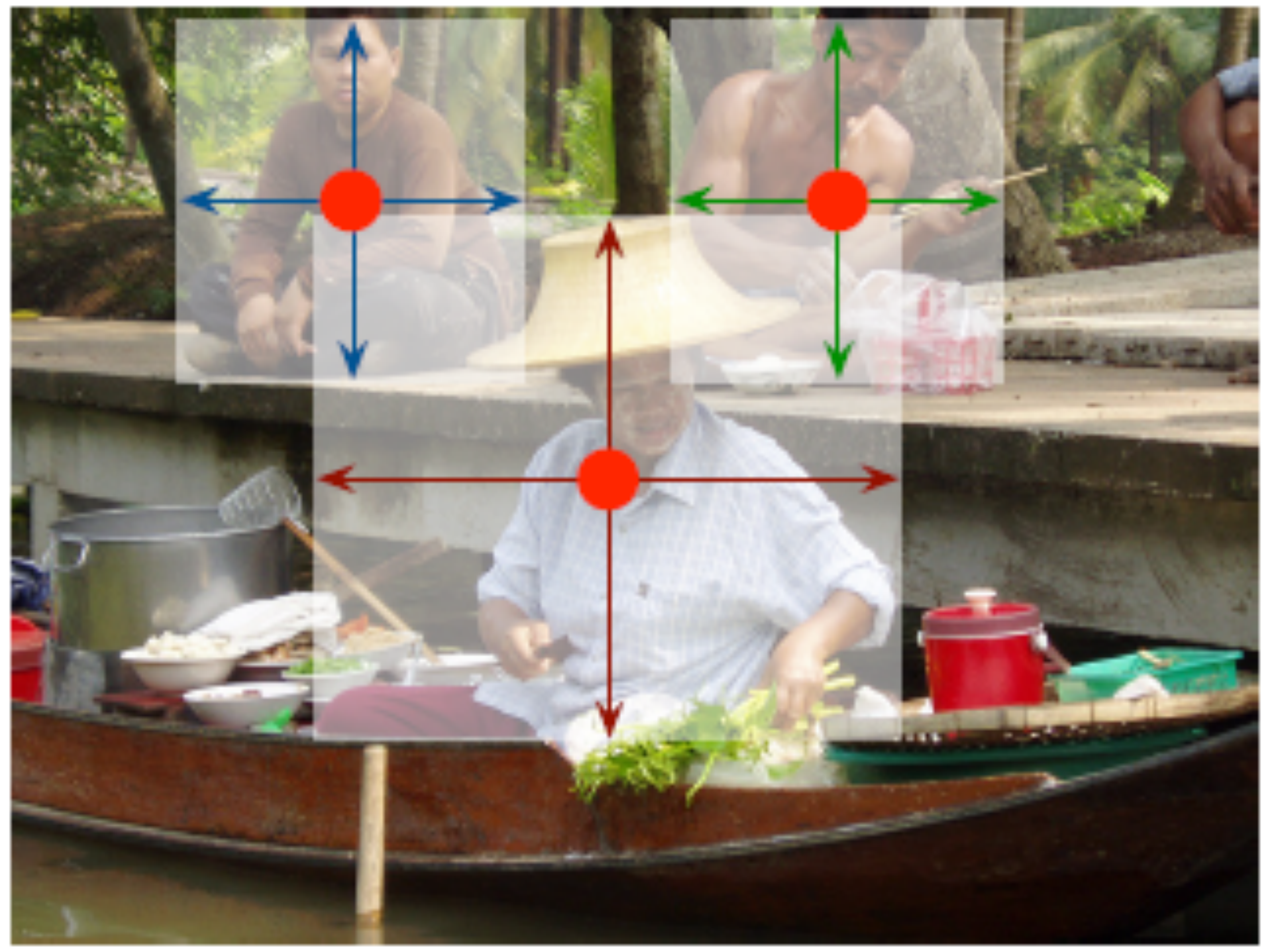




A anchors, K classes

4 offsets

1-stage object detection: CenterNet (anchor-free)



- Represent objects by a single point + (width, height)
- Regress other parameters such as
  - Bounding box
  - 3D box
  - human pose
  - ...

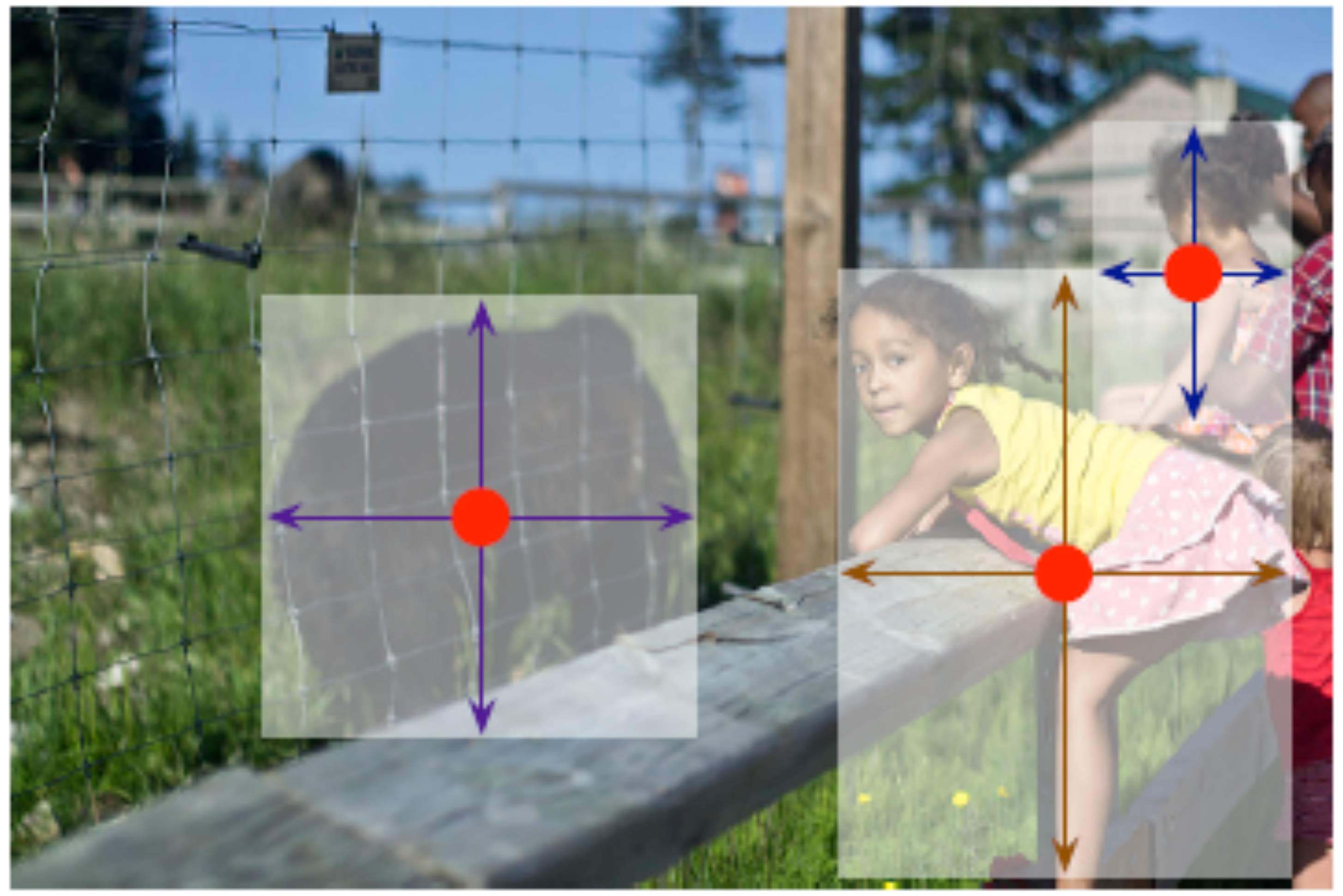
Output : 64 x 64 x 1 x (C+2+2)

#keypoints

offsets

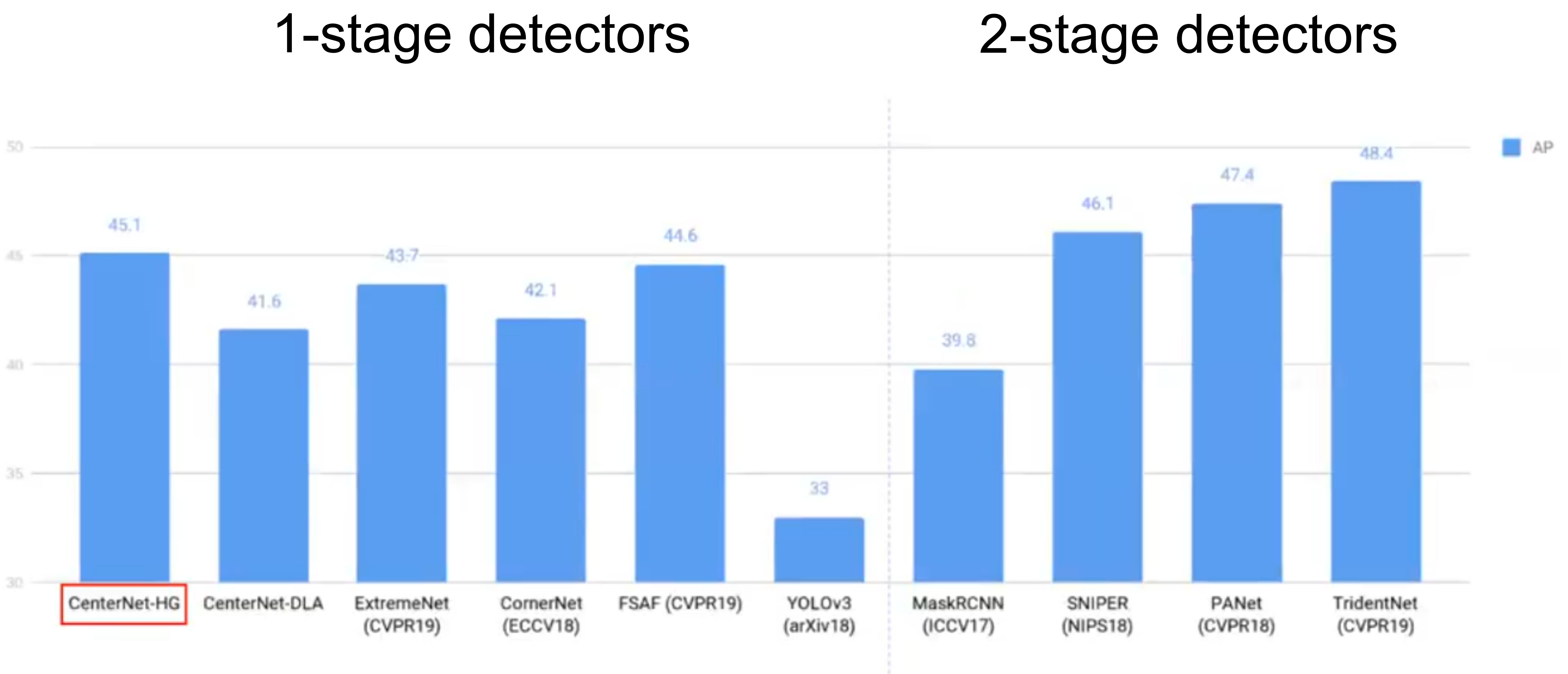
w,h

object category



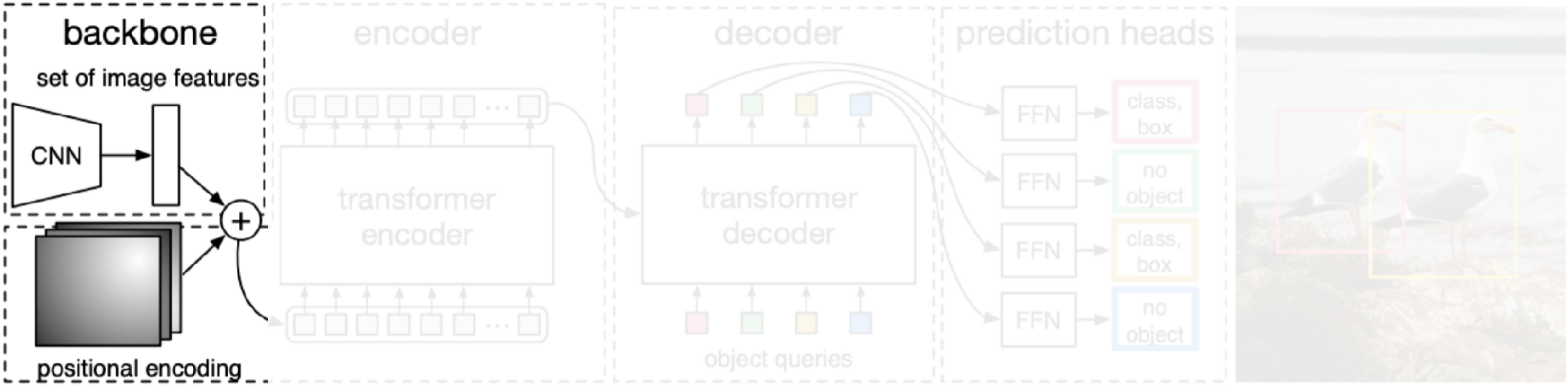
[Objects as Points, X. Zhou, D. Wang and P. Krahenbuhl, 2019]

State-of-the-art comparison: MS COCO



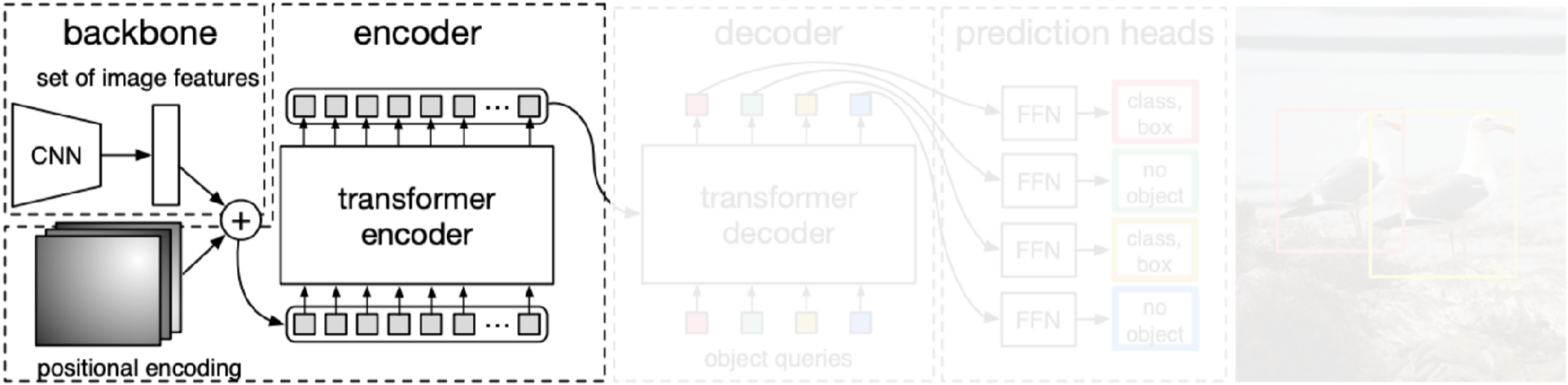
Object detection: Transformer-based methods

DETR: Object detection with transformers



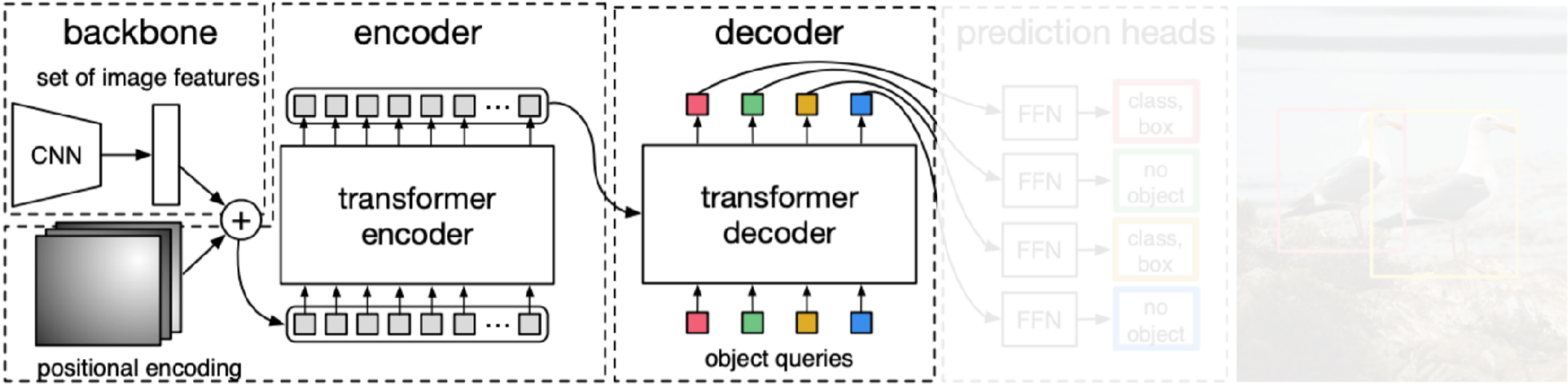
N. Carion et al., End-to-end object detection with transformers, ECCV 2020

DETR: Object detection with transformers



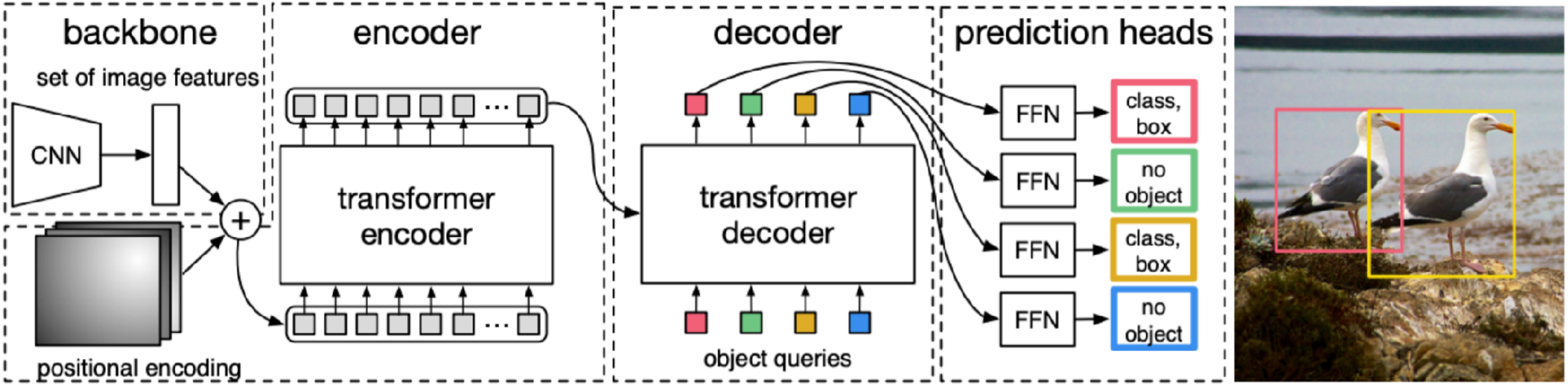
N. Carion et al., End-to-end object detection with transformers, ECCV 2020

DETR: Object detection with transformers



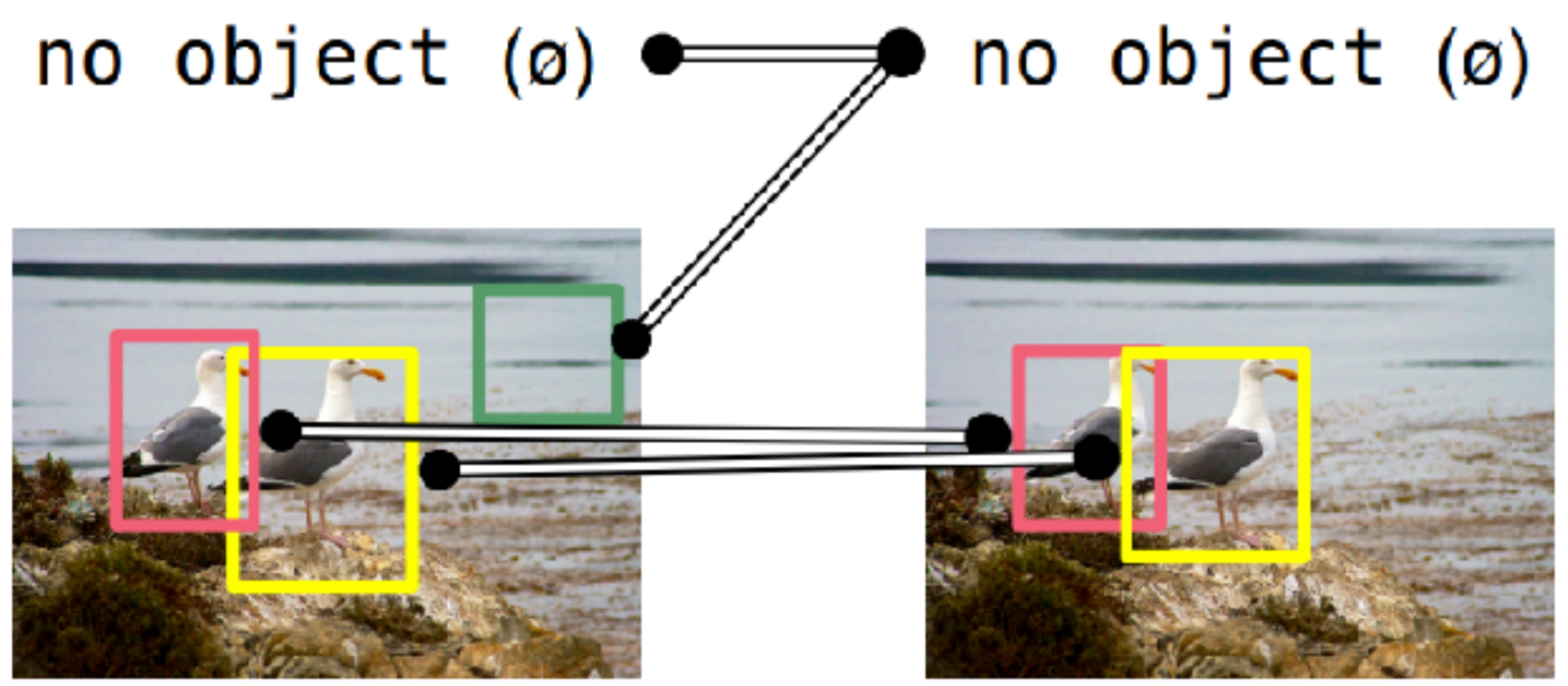
N. Carion et al., End-to-end object detection with transformers, ECCV 2020

DETR: Object detection with transformers

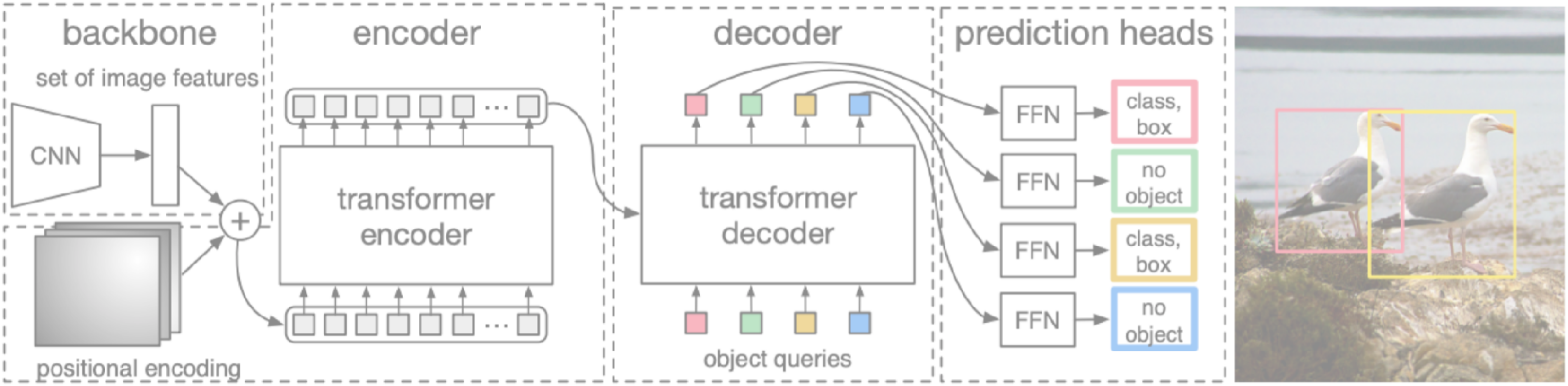


N. Carion et al., End-to-end object detection with transformers, ECCV 2020

DETR: Training



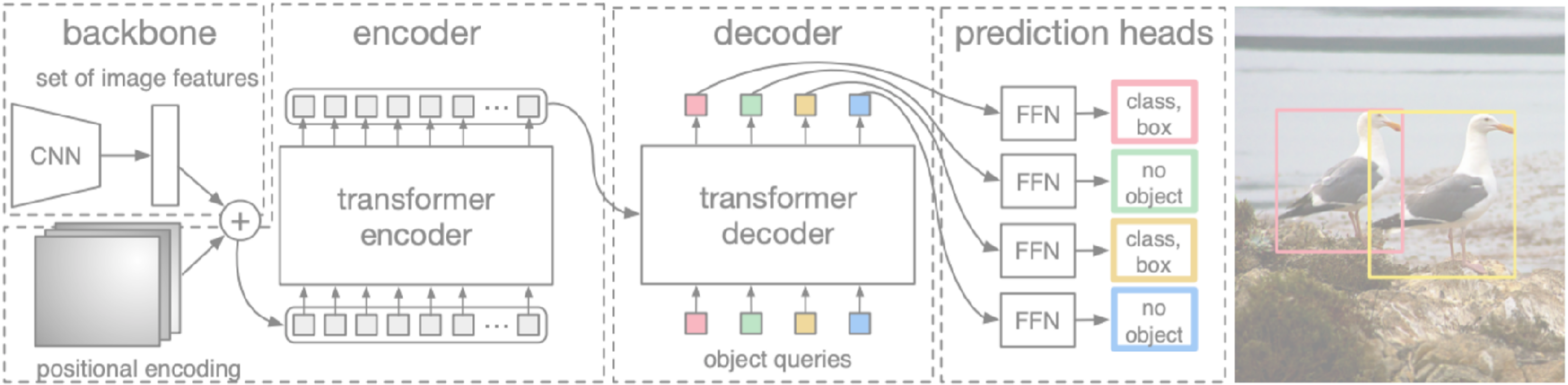
- Match each box proposal to ground truth
- Use Hungarian algorithm to find permutation minimizing matching loss



N. Carion et al., End-to-end object detection with transformers, ECCV 2020

DETR: Results COCO Val

Model	Epochs	mAP	mAP (small)	mAP (medium)	mAP (large)
Faster RCNN-FPN	109	42.0	26.6	45.4	53.4
DETR	500	42.0	20.5	45.8	61.1



N. Carion et al., End-to-end object detection with transformers, ECCV 2020

Agenda

- 0. Intro to structured outputs
- 1. Object detection (localization)
- 2. Segmentation
- 3. Human pose estimation

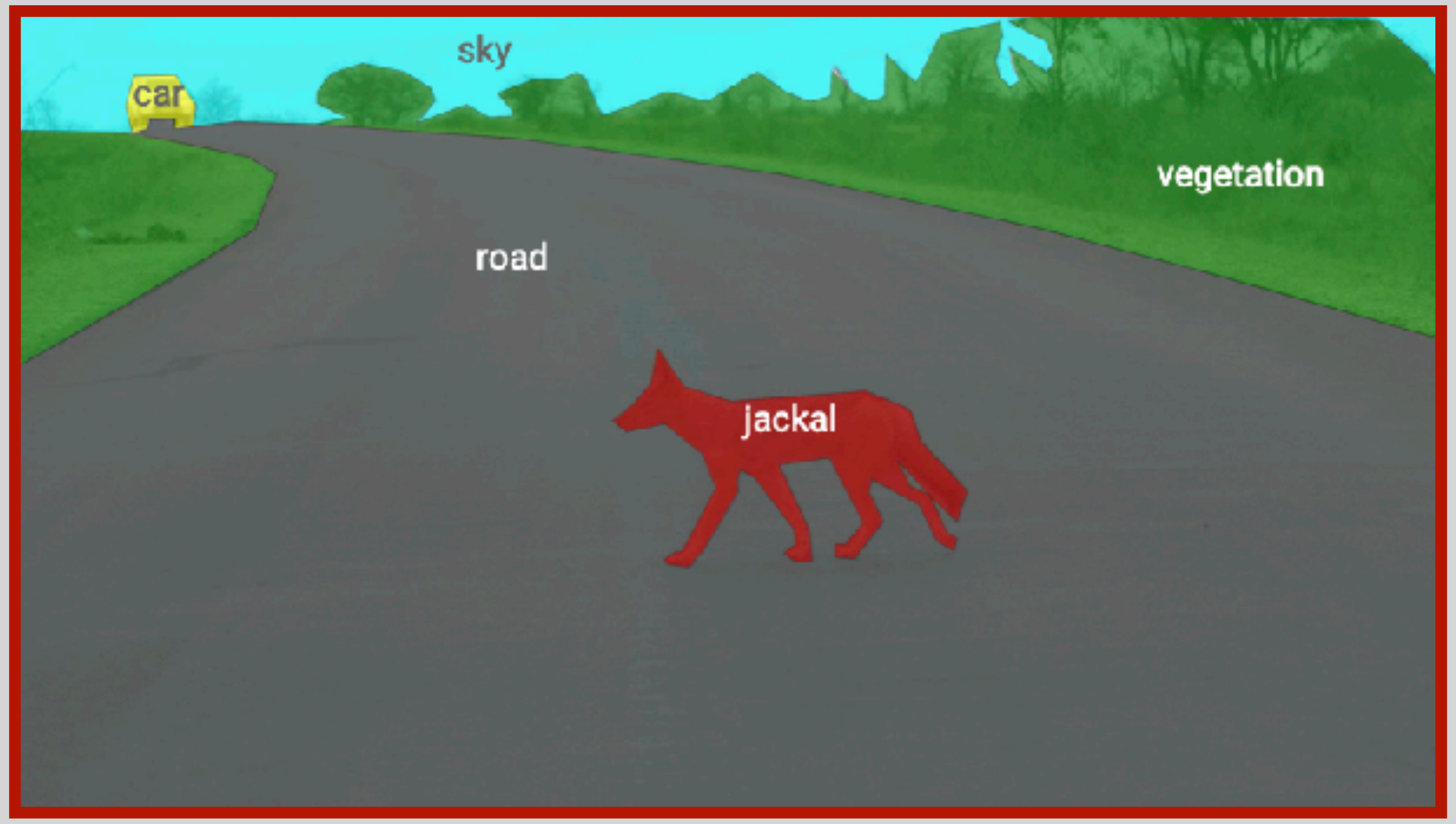
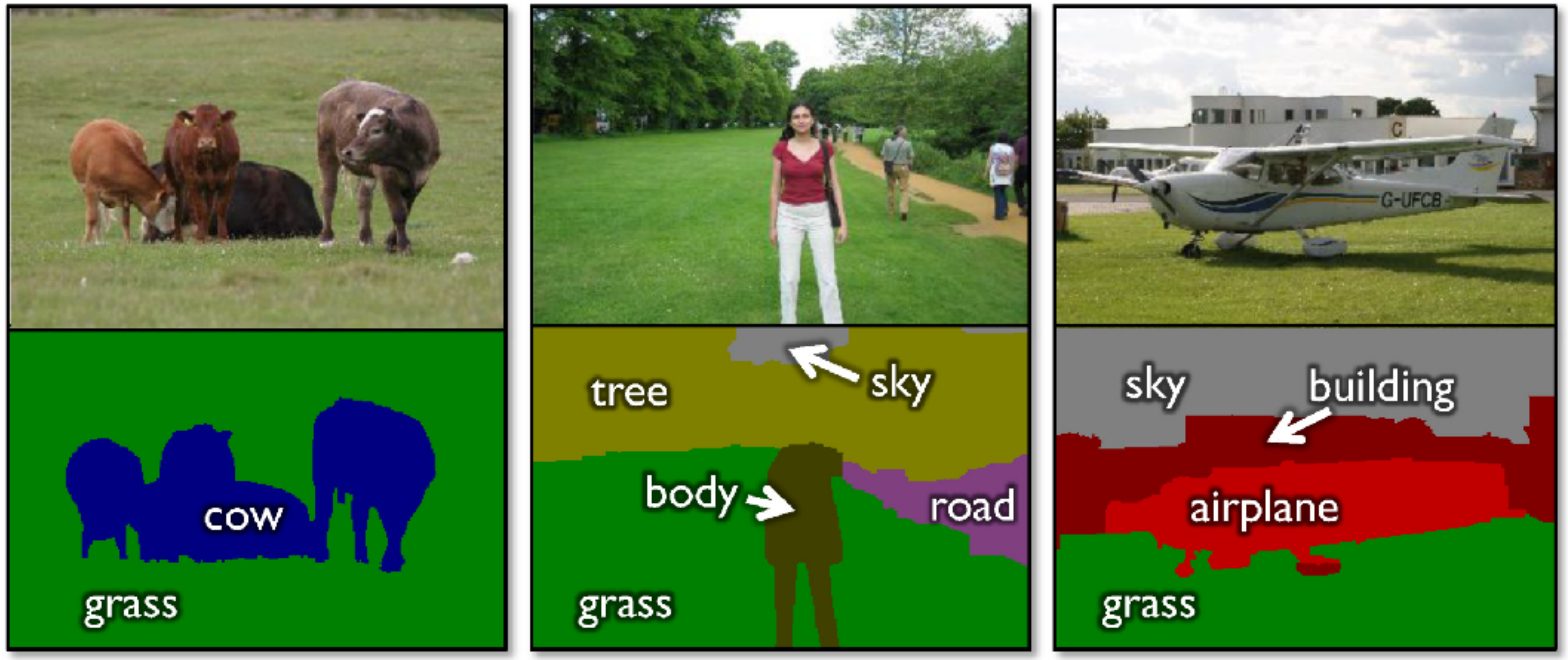


Image credits: Naila Murray

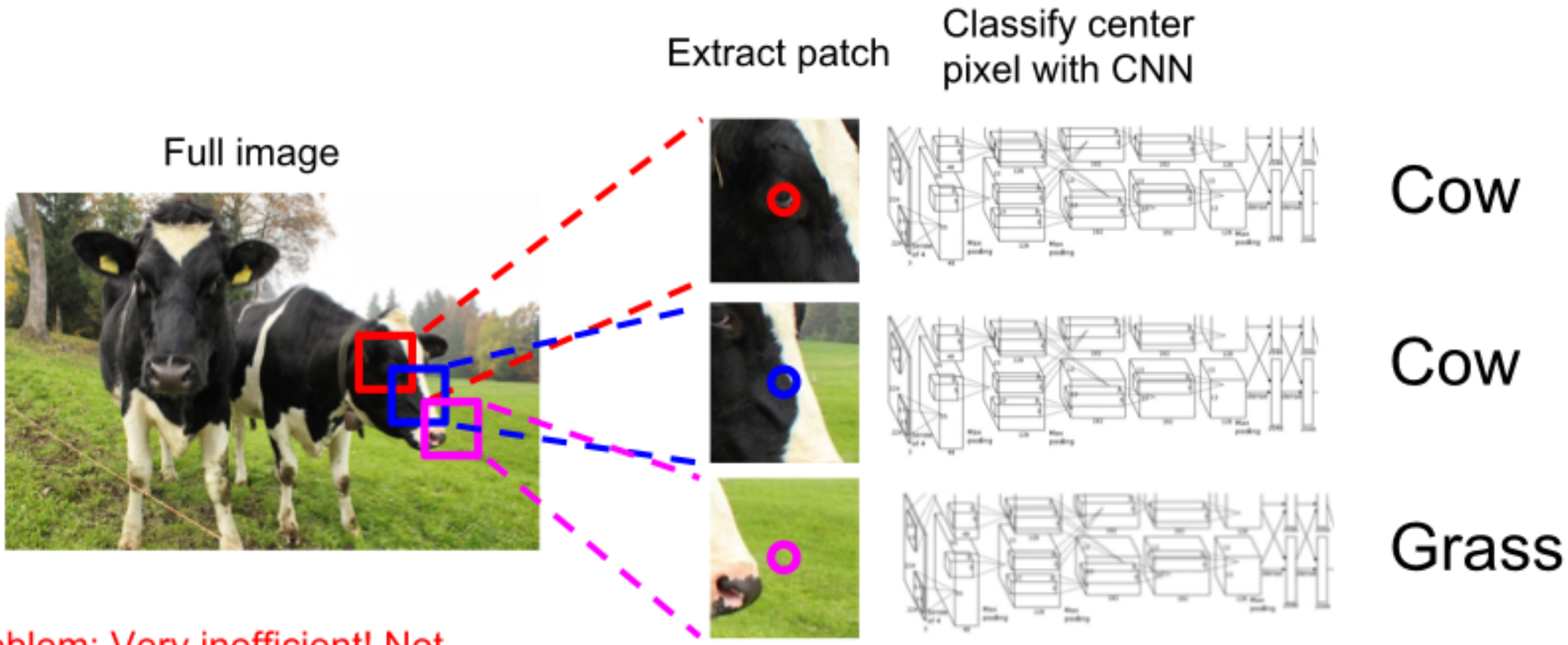
Semantic segmentation

- Label each pixel in the image with a category label
- Don't differentiate instances, only care about pixels



object classes	building	grass	tree	cow	sheep	sky	airplane	water	face	car
bicycle	flower	sign	bird	book	chair	road	cat	dog	body	boat

Semantic segmentation: sliding window

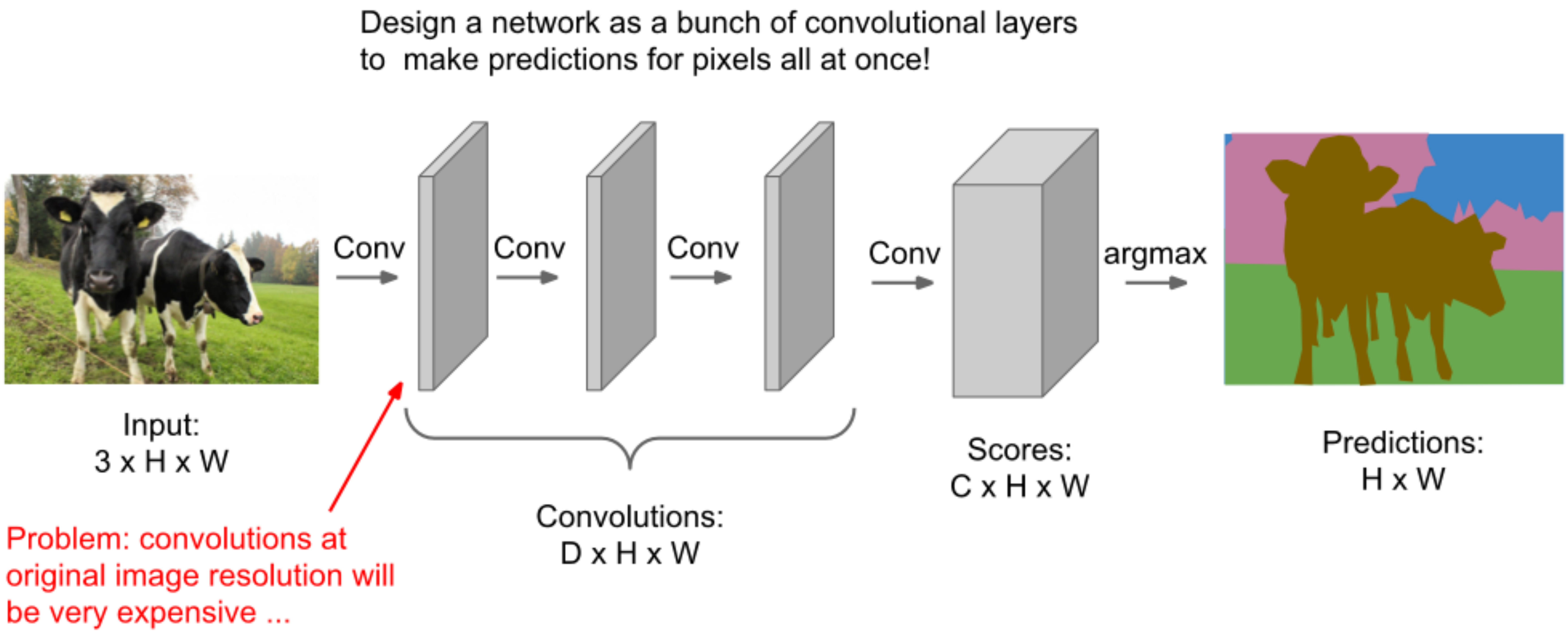


Problem: Very inefficient! Not reusing shared features between overlapping patches

Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013
Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

Slide credit: L. Fei Fei, J. Johnson, S. Yeung, http://cs231n.stanford.edu/

Semantic segmentation: fully convolutional



Slide credit: L. Fei Fei, J. Johnson, S. Yeung, http://cs231n.stanford.edu/

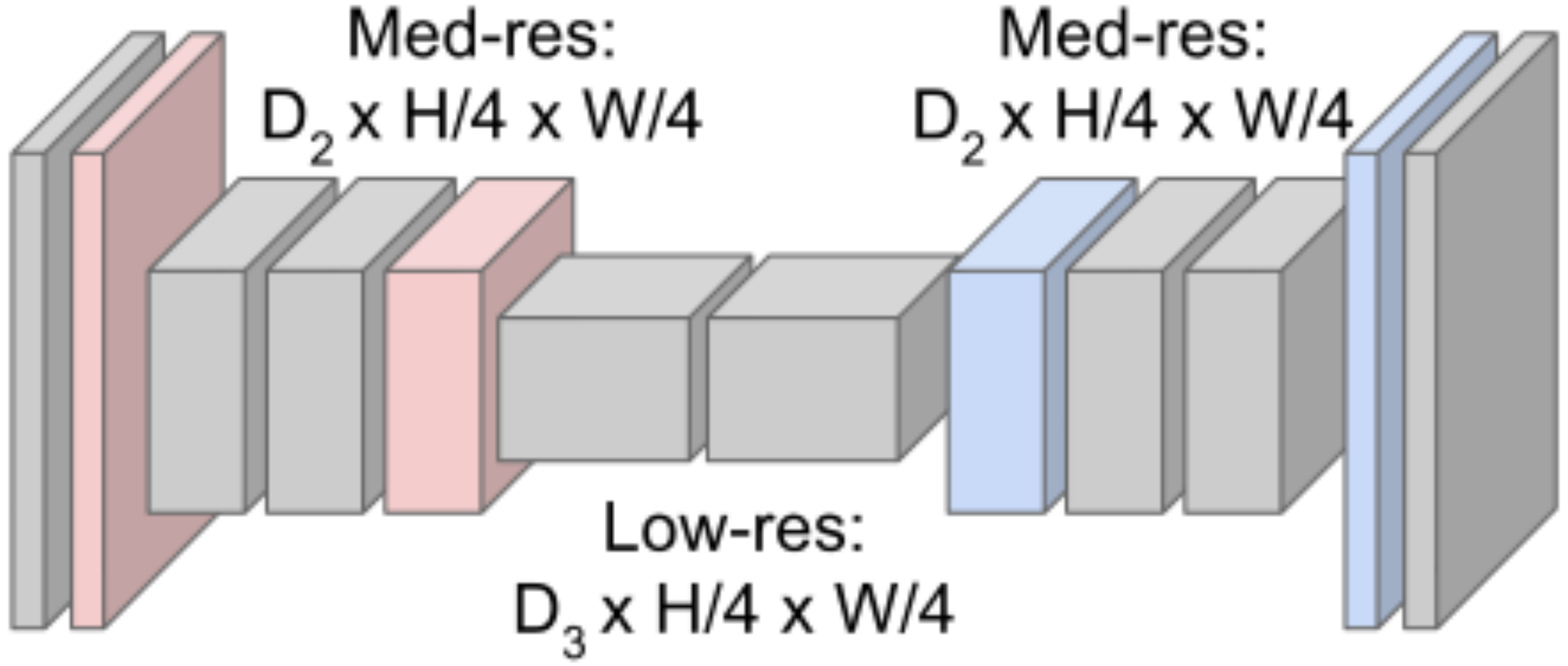
Semantic segmentation: fully convolutional

Downsampling:
Pooling, strided convolution

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Upsampling:
???

Input:
3 x H x W

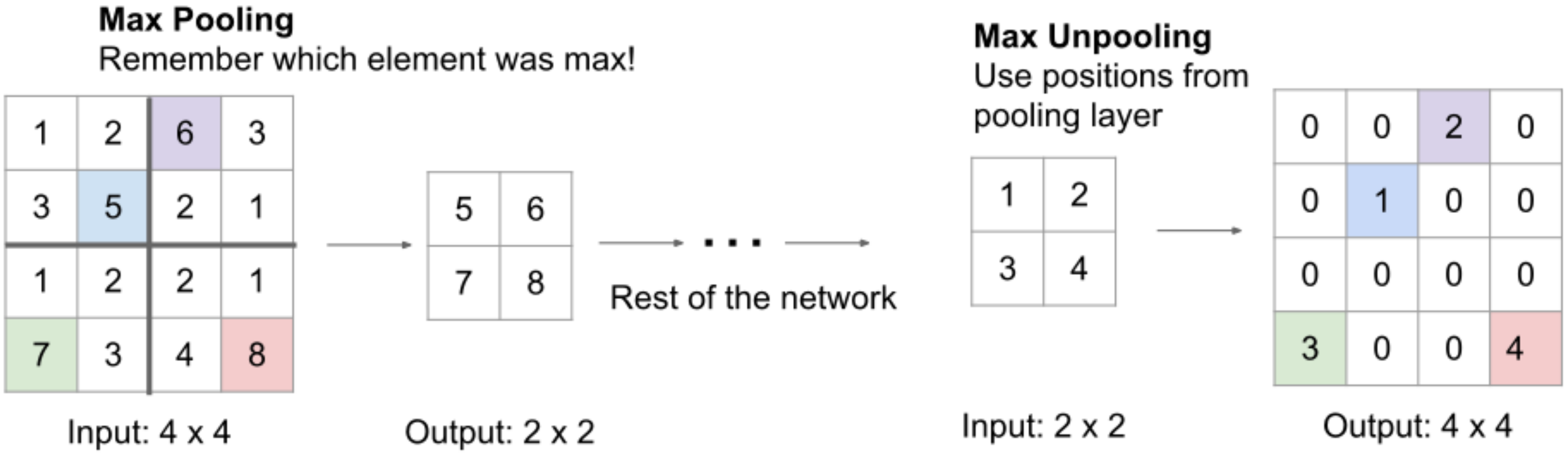


Predictions:
H x W

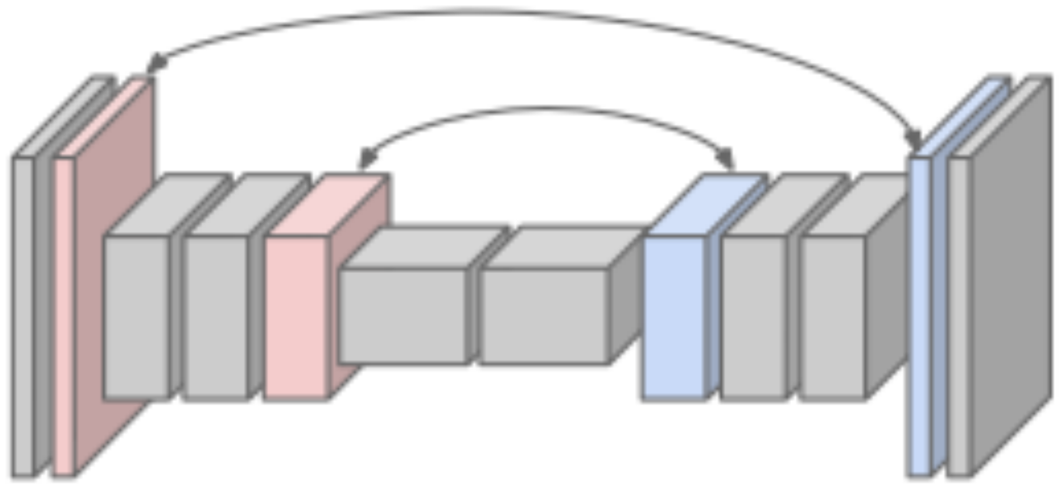
Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al. "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Slide credit: L. Fei Fei, J. Johnson, S. Yeung, http://cs231n.stanford.edu/

In-network upsampling: “Unpooling”



Corresponding pairs of downsampling and upsampling layers



Slide credit: L. Fei Fei, J. Johnson, S. Yeung, http://cs231n.stanford.edu/

Learnable upsampling

Other names: Deconvolution (bad); Upconvolution; Fractionally strided convolution; Backward strided convolution; Transpose convolution

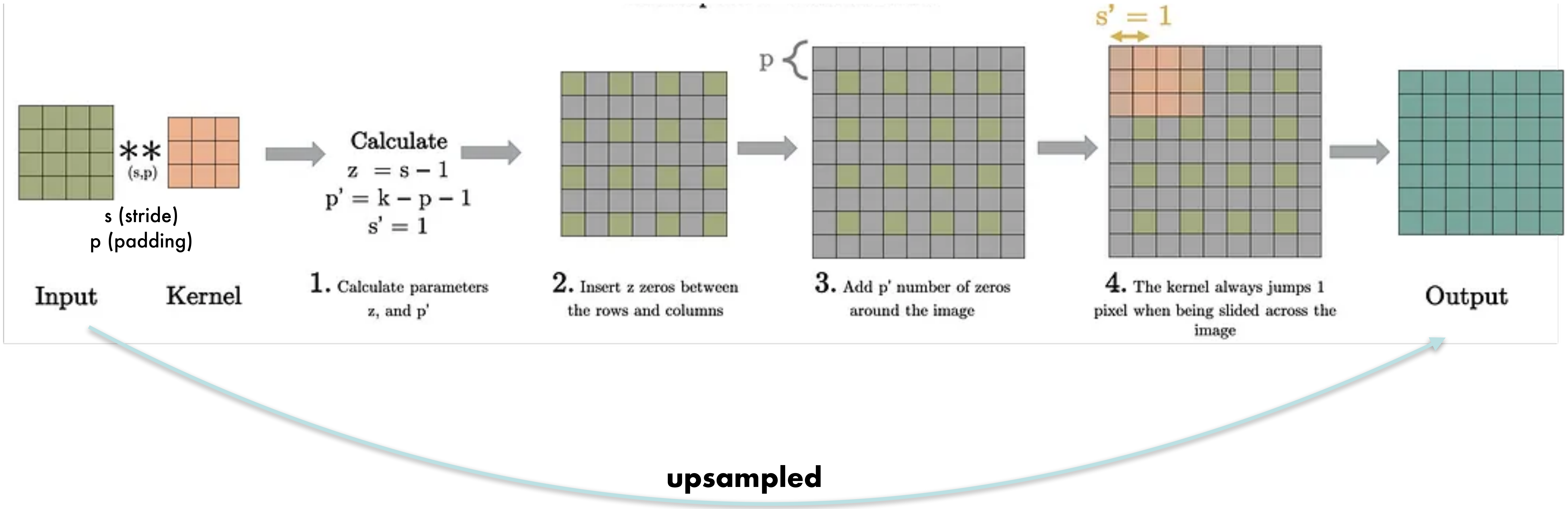
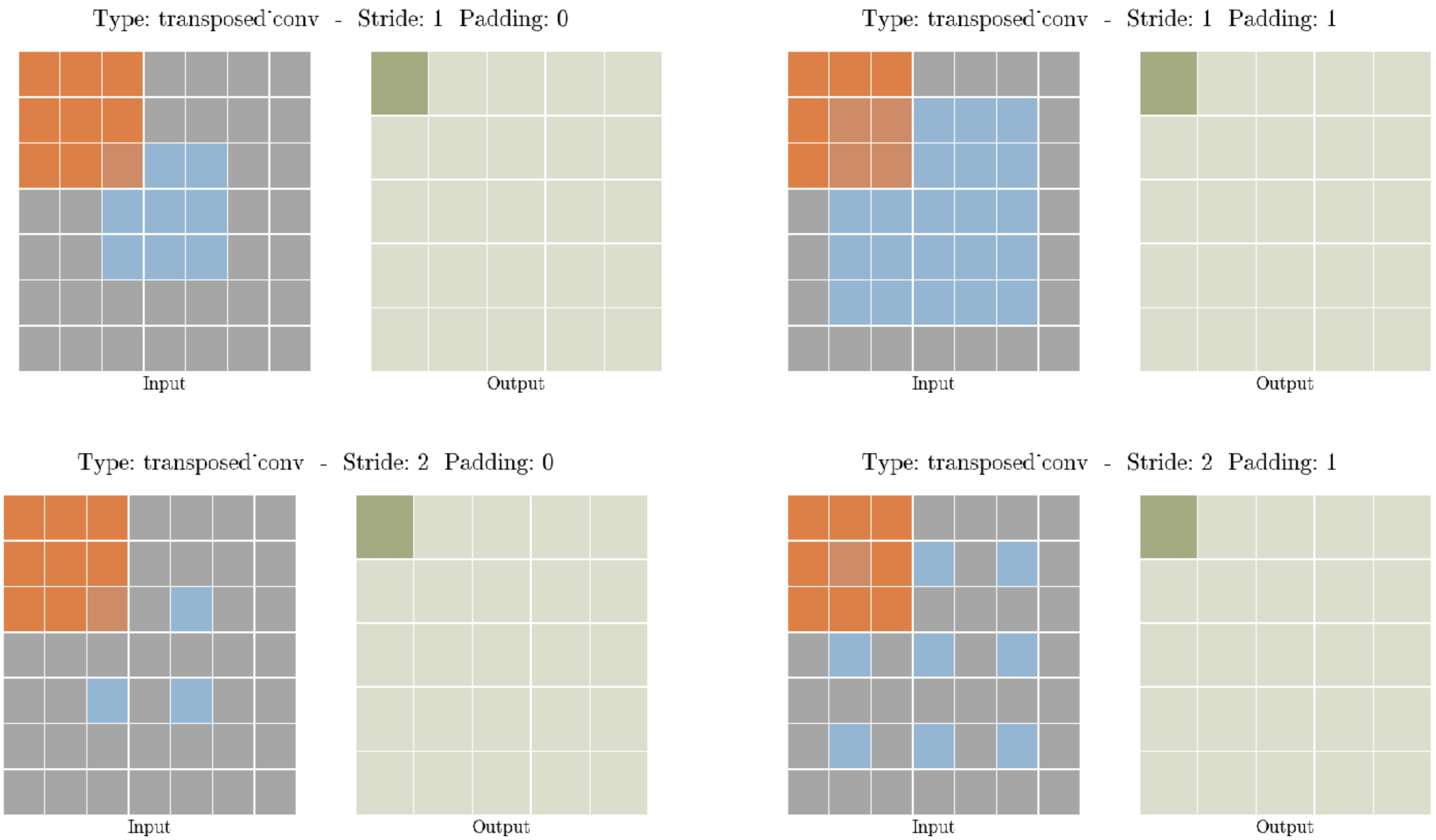


Image source

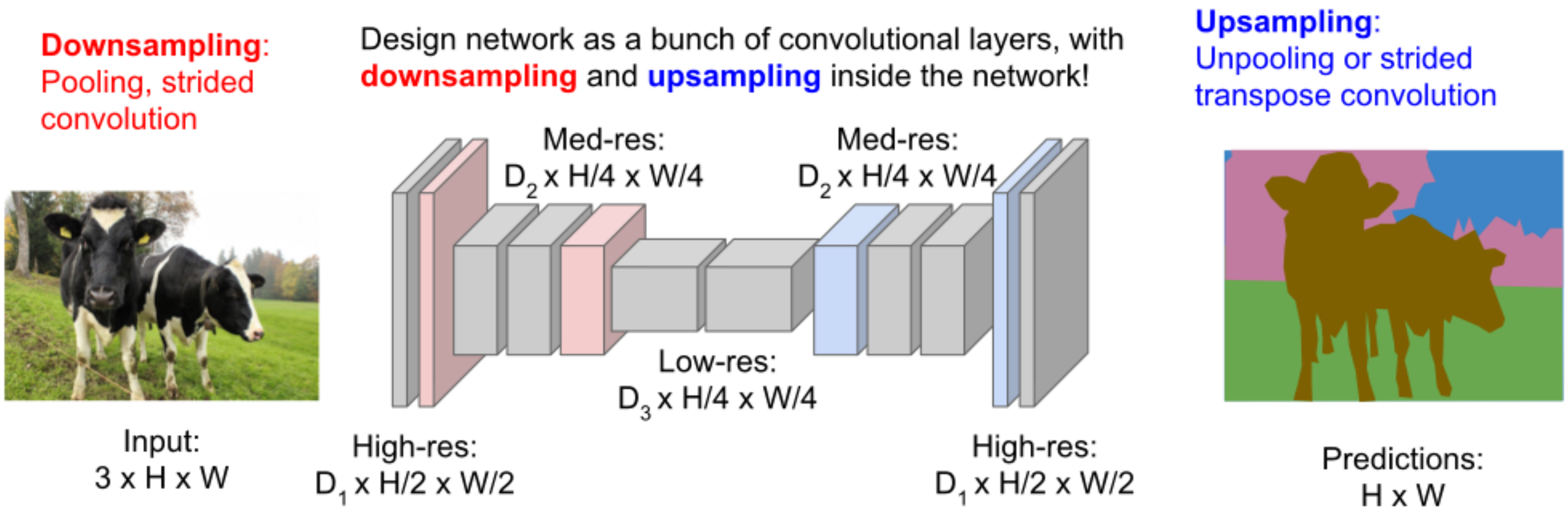
Learnable upsampling

Other names: Deconvolution (bad); Upconvolution; Fractionally strided convolution; Backward strided convolution; Transpose convolution



[Image source](#)

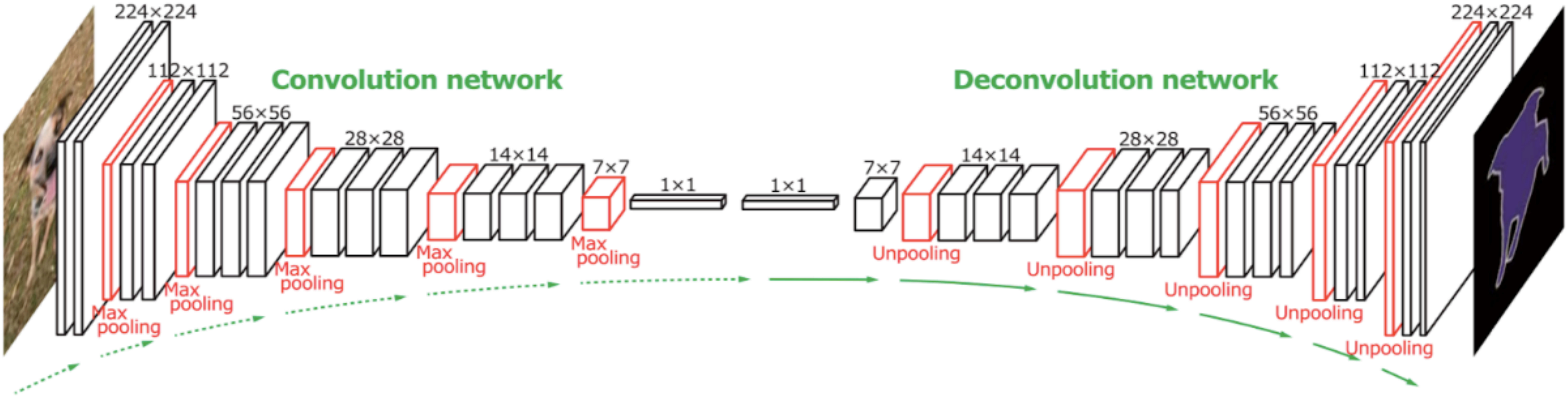
Semantic segmentation: fully convolutional



Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al. "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

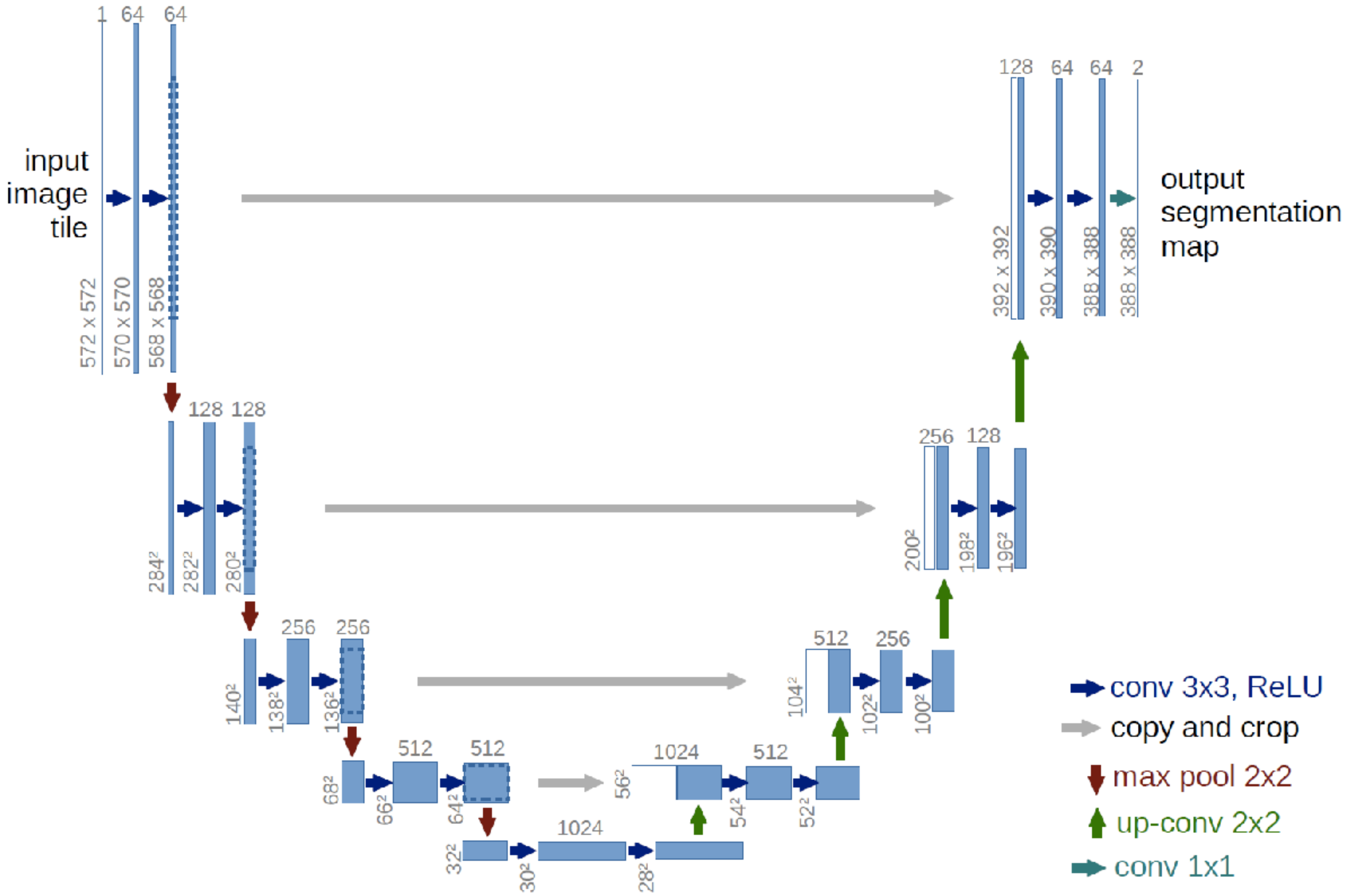
Slide credit: L. Fei Fei, J. Johnson, S. Yeung, http://cs231n.stanford.edu/

Semantic segmentation: Auto-encoder

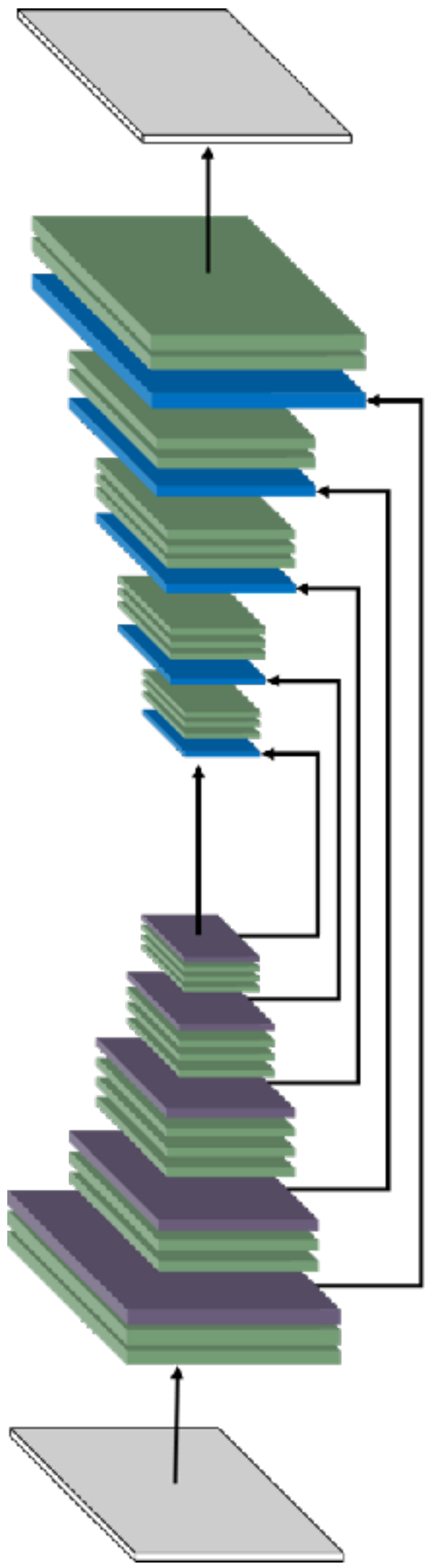


Why is this a bad idea for segmentation?

Semantic segmentation: U-Net or “Hourglass”



Ronneberger et al., MICCAI 2015.

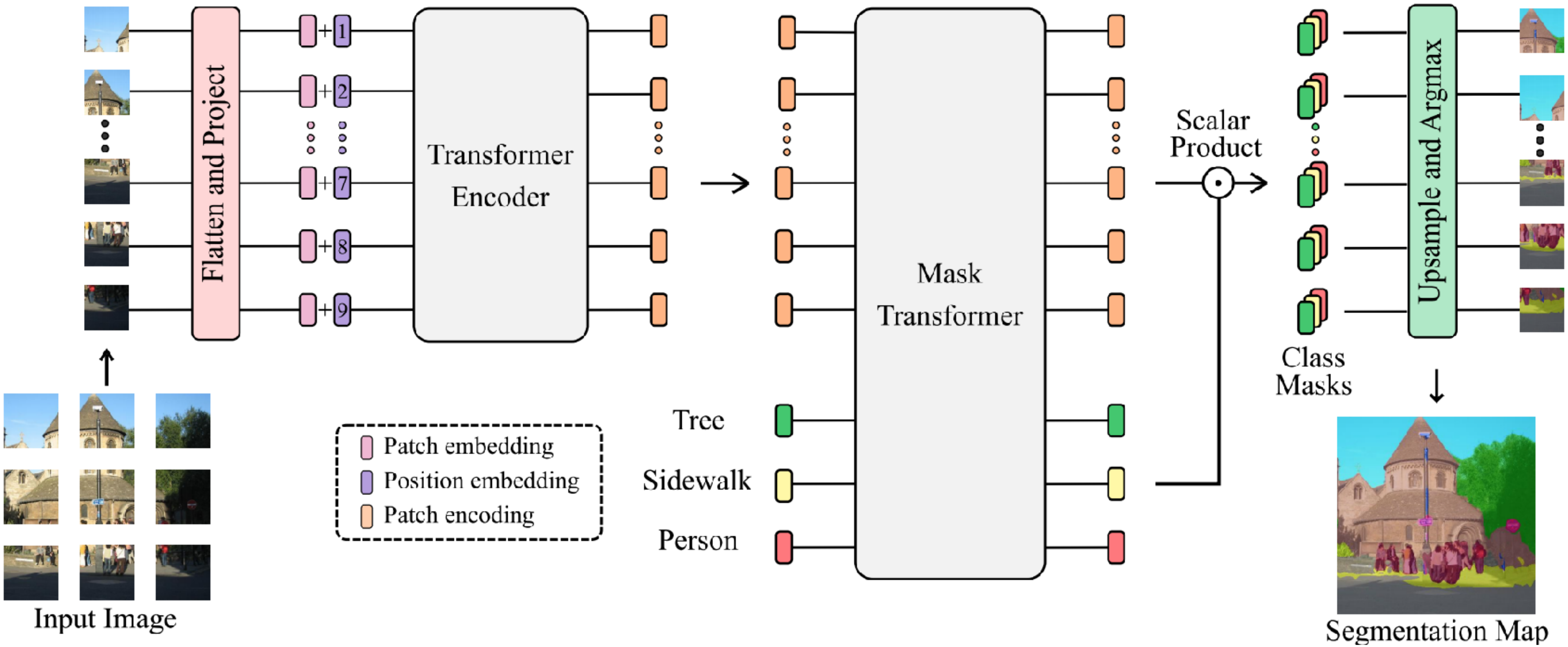


Newell et al., ECCV 2016.

Fig: Nushaine Ferdinand

Semantic segmentation: Segmenter

Transformer architecture for image segmentation



R. Strudel et al., Segmenter: Transformer for Semantic Segmentation, ICCV 2021

(Object) Instance segmentation

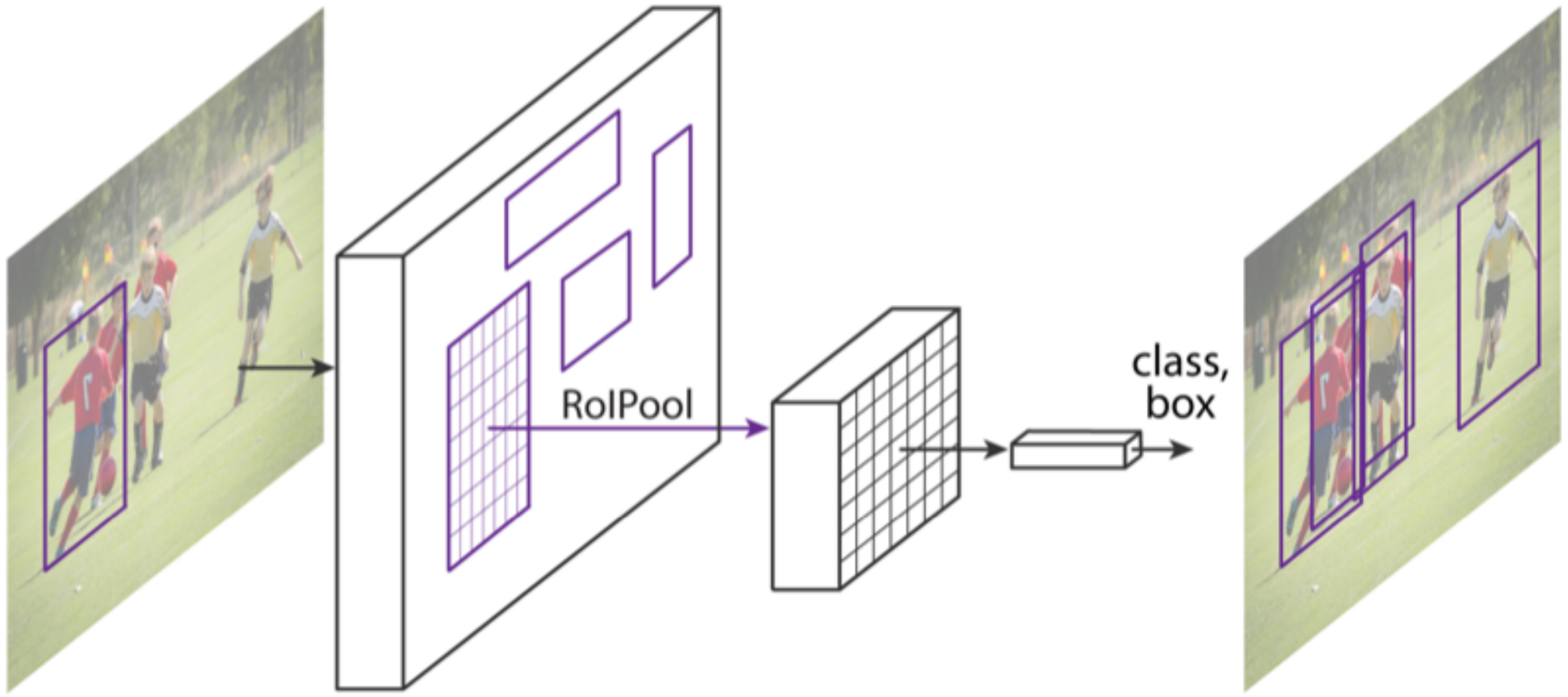
- Differentiate instances
- Object detection + segmentation

Object Detection | Semantic Segmentation | **Instance Segmentation**

Slide credit: K. He, instancetutorial.github.io

Remember:

Object detection using Fast(er) R-CNN

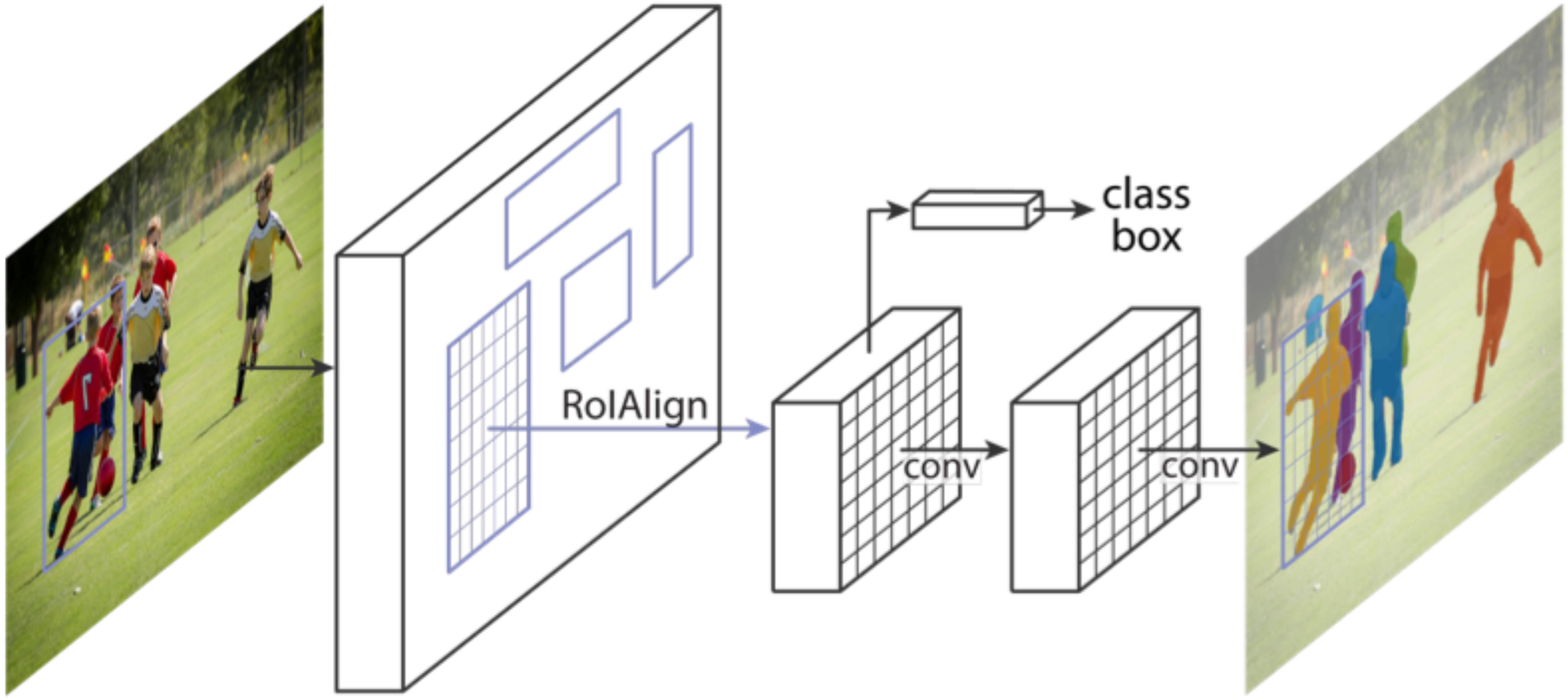


Ross Girshick. “Fast R-CNN”. ICCV 2015. Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. NIPS 2015.

Slide credit: K. He, instancetutorial.github.io

Mask R-CNN

Object detection *and* segmentation

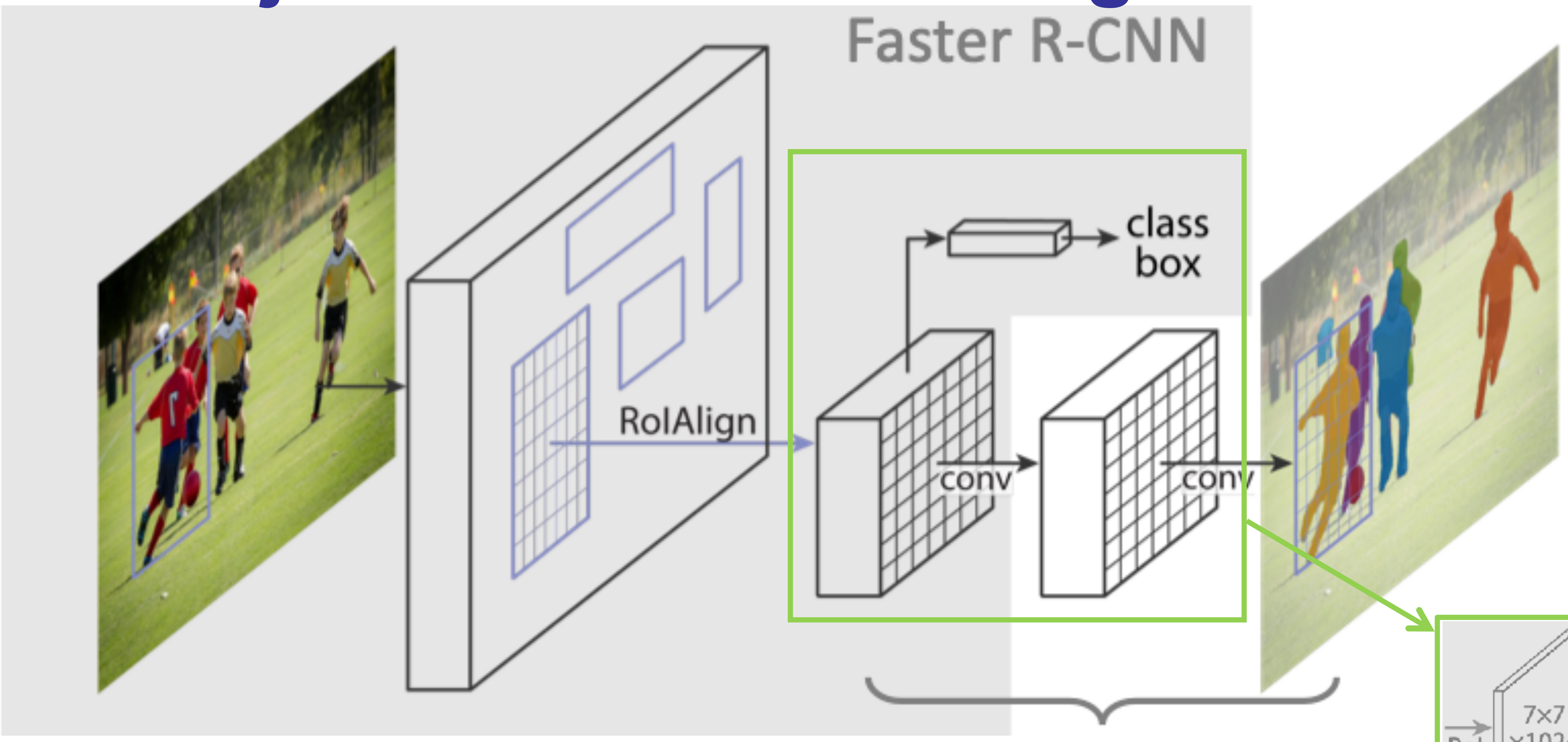


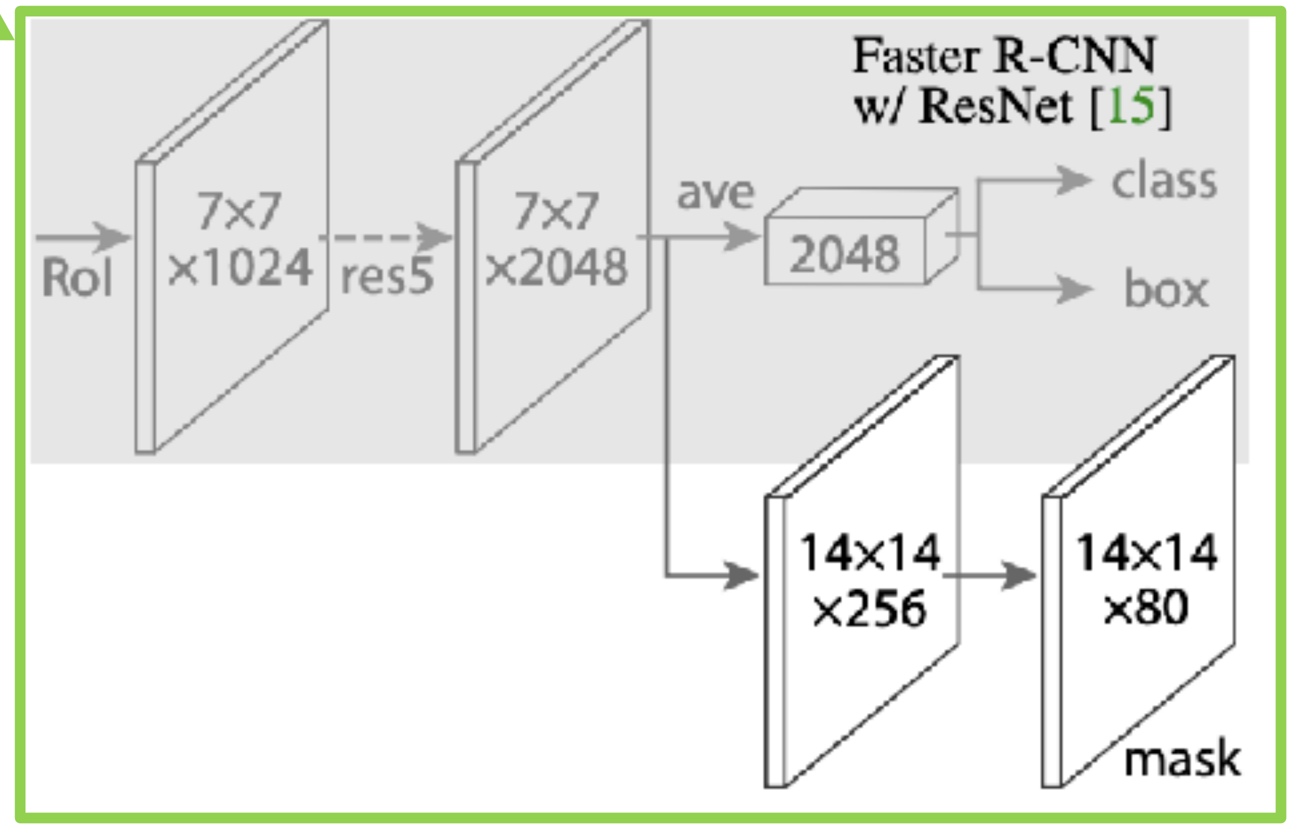
[He, K., Gkioxari, G., Dollár, P., & Girshick, R. Mask R-CNN ICCV 2017]

Slide credit: K. He, instancetutorial.github.io

Mask R-CNN

Object detection *and* segmentation





1. Object detector using Faster RCNN +
2. Fully convolutional network (FCN) on region of interest (RoI)

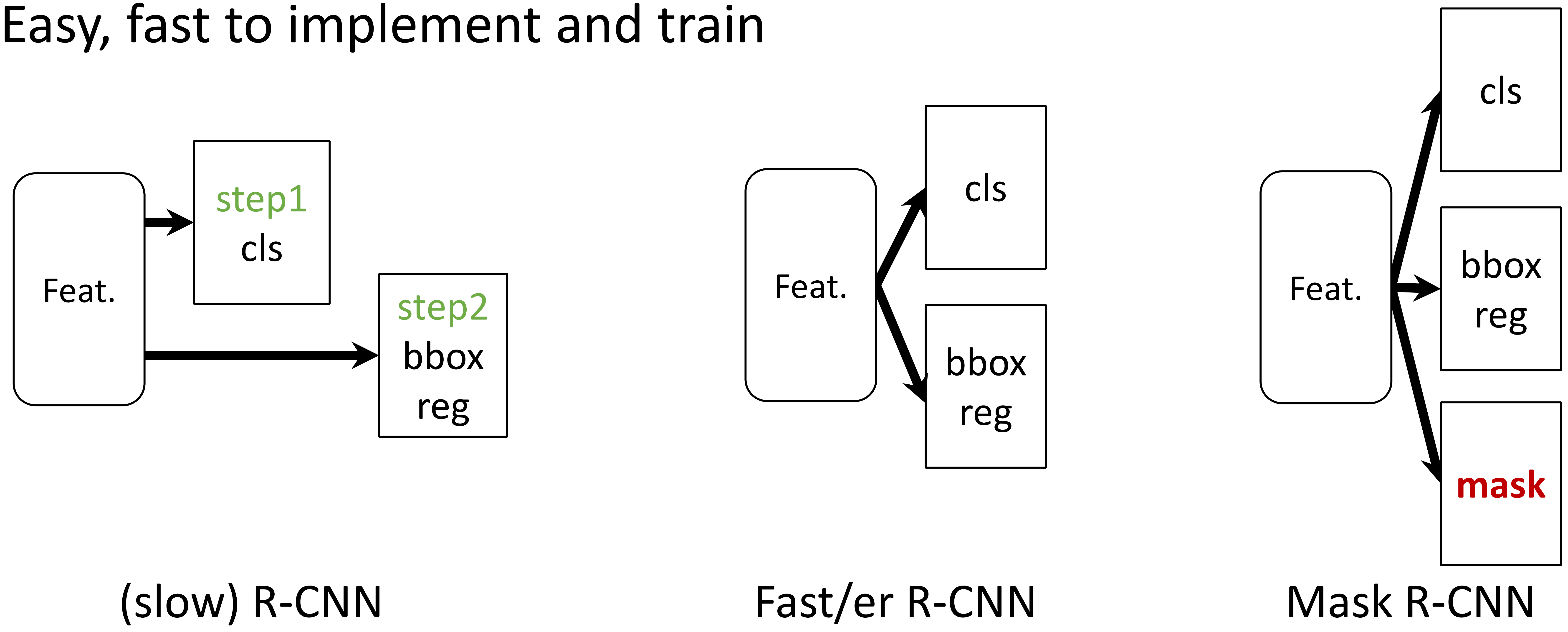
[He, K., Gkioxari, G., Dollár, P., & Girshick, R. Mask R-CNN ICCV 2017]

Slide credit: K. He, instancetutorial.github.io

Mask R-CNN

Combining loss functions

- Easy, fast to implement and train



[He, K., Gkioxari, G., Dollár, P., & Girshick, R. Mask R-CNN ICCV 2017]

Slide credit: K. He, instancetutorial.github.io

Mask R-CNN

Example results

object surrounded by same-category objects

Mask R-CNN results on COCO

[He, K., Gkioxari, G., Dollár, P., & Girshick, R. Mask R-CNN ICCV 2017]

Slide credit: K. He, instancetutorial.github.io

Mask R-CNN

Example results

disconnected
object

Mask R-CNN results on COCO

[He, K., Gkioxari, G., Dollár, P., & Girshick, R. Mask R-CNN ICCV 2017]

Slide credit: K. He, instancetutorial.github.io

Mask R-CNN

Example results

small objects

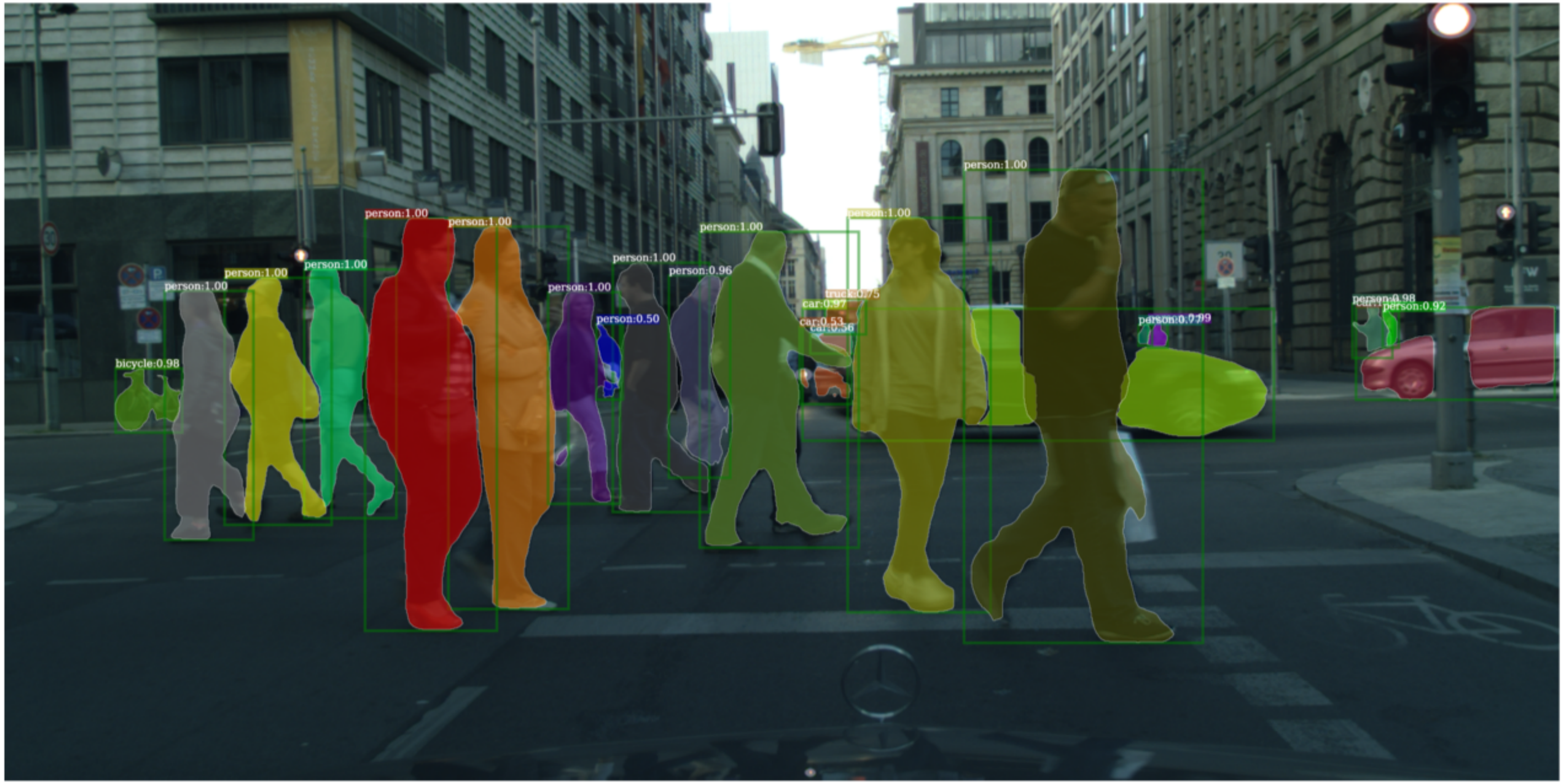
Mask R-CNN results on COCO

[He, K., Gkioxari, G., Dollár, P., & Girshick, R. Mask R-CNN ICCV 2017]

Slide credit: K. He, instancetutorial.github.io

Mask R-CNN

Example results



Mask R-CNN results on CityScapes

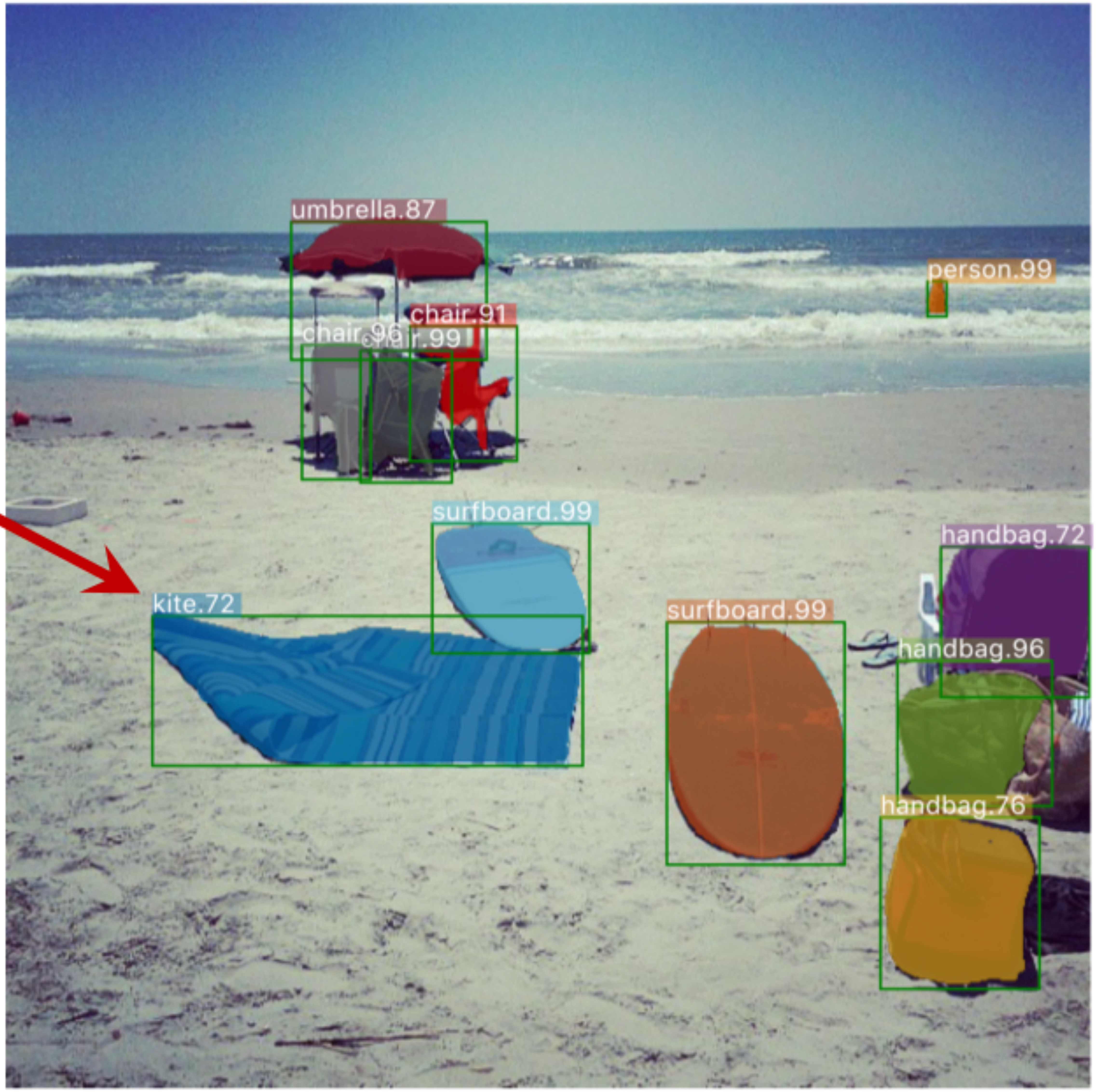
[He, K., Gkioxari, G., Dollár, P., & Girshick, R. Mask R-CNN ICCV 2017]

Slide credit: K. He, instancetutorial.github.io

Mask R-CNN

Example failures: recognition

not a kite



Mask R-CNN results on COCO

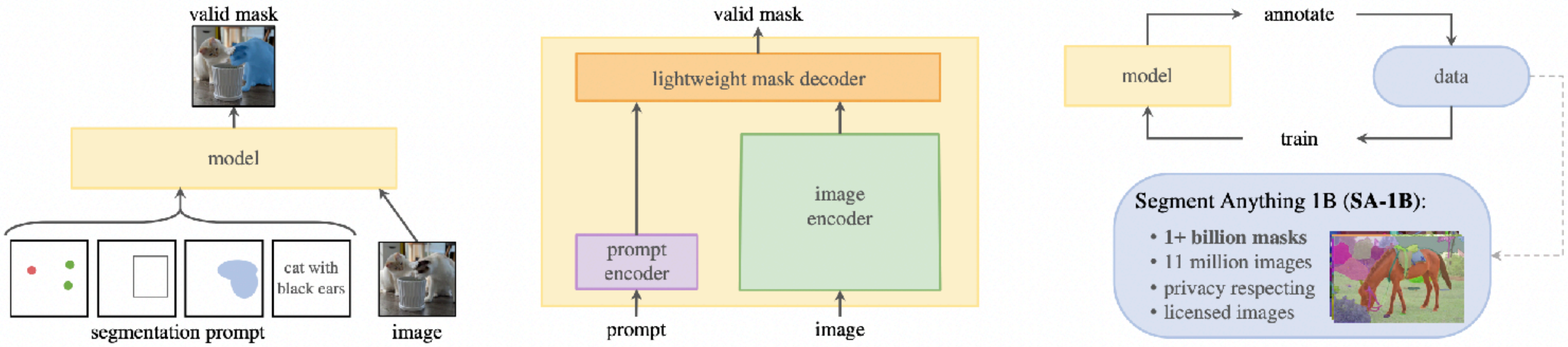
[He, K., Gkioxari, G., Dollár, P., & Girshick, R. Mask R-CNN ICCV 2017]

Slide credit: K. He, instancetutorial.github.io

Promptable segmentation

Segment Anything

Alexander Kirillov[1,2,4] Eric Mintun[2] Nikhila Ravi[1,2] Hanzi Mao[2] Chloe Rolland[3] Laura Gustafson[3]
Tete Xiao[3] Spencer Whitehead Alexander C. Berg Wan-Yen Lo Piotr Dollár[4] Ross Girshick[4]
[1]project lead [2]joint first author [3]equal contribution [4]directional lead
Meta AI Research, FAIR

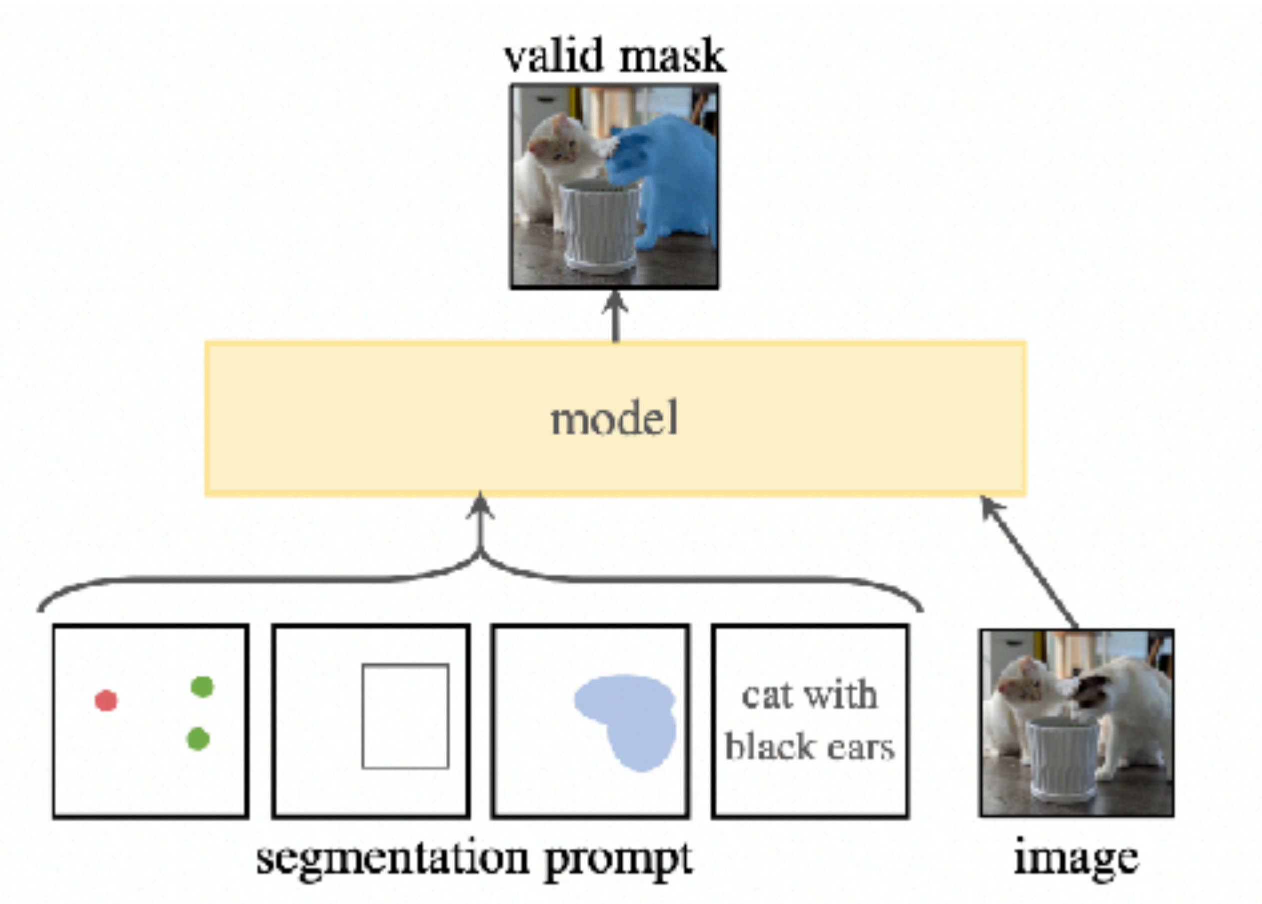


(a) **Task:** promptable segmentation (b) **Model**: Segment Anything Model (**SAM**) (c) **Data**: data engine (top) & dataset (bottom)

Figure 1: We aim to build a foundation model for segmentation by introducing three interconnected components: a promptable segmentation *task*, a segmentation *model* (SAM) that powers data annotation and enables zero-shot transfer to a range of tasks via prompt engineering, and a *data* engine for collecting SA-1B, our dataset of over 1 billion masks.

[Kirillov et al., Segment Anything, ICCV 2023 (Best Paper Honorable Mention)]

Segment Anything Model (SAM) https://segment-anything.com/demo



(a) **Task:** promptable segmentation

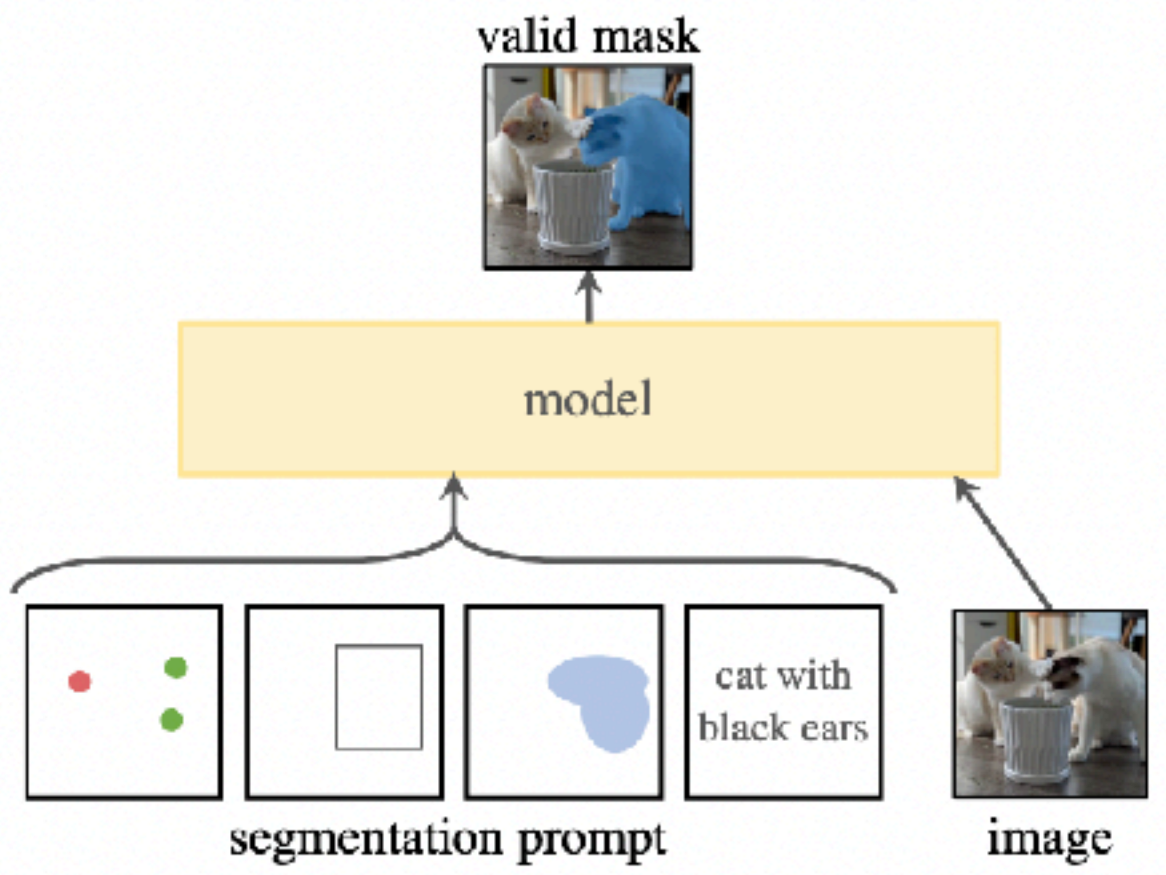

Prompting with a point


Prompting with a dense grid of points

[Kirillov et al., Segment Anything, ICCV 2023 (Best Paper Honorable Mention)]

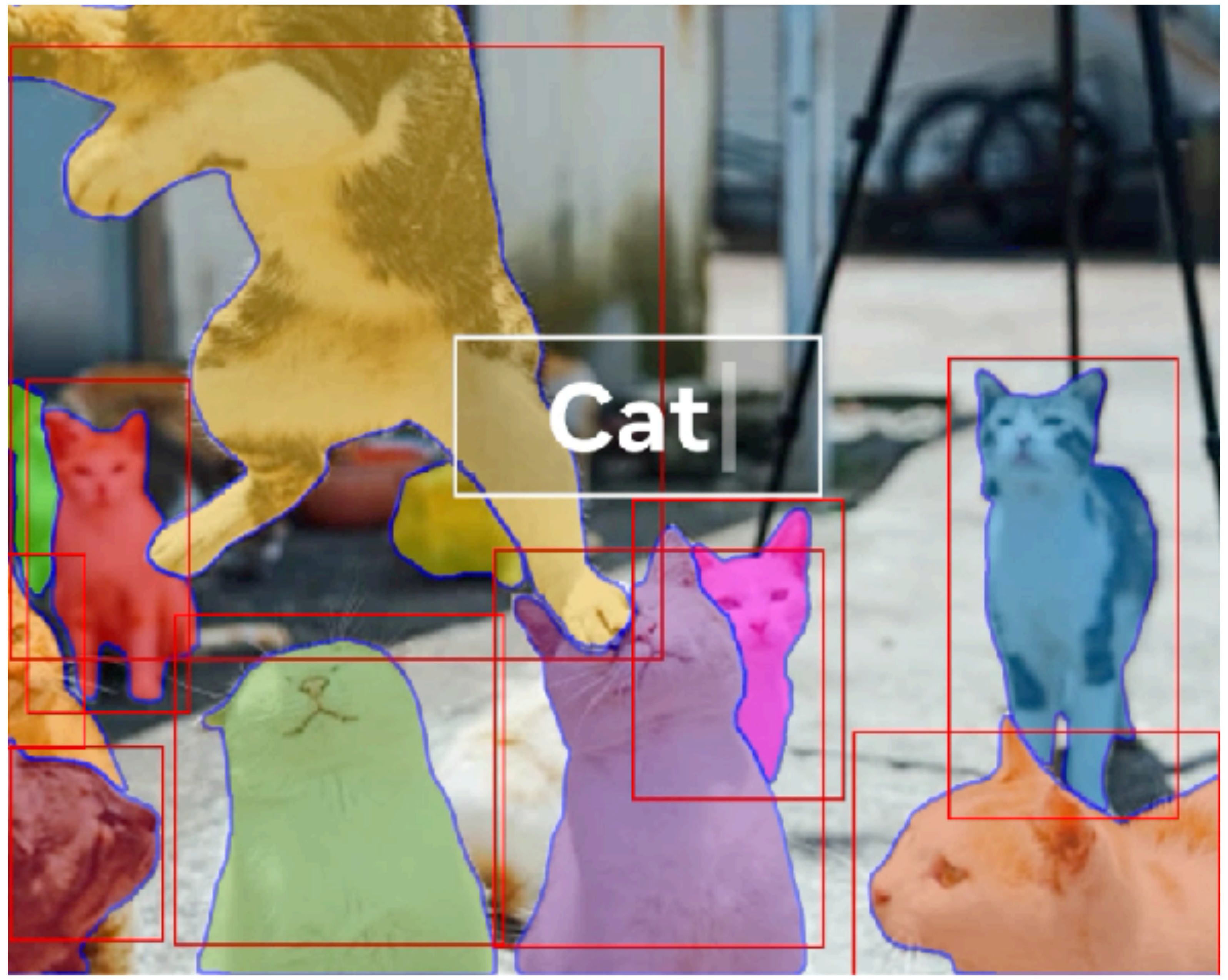
Segment Anything Model (SAM)

Not **semantic segmentation** (no category)



(a) **Task:** promptable segmentation

Could be used for **instance segmentation** by integrating an object detector



Prompting with detected boxes

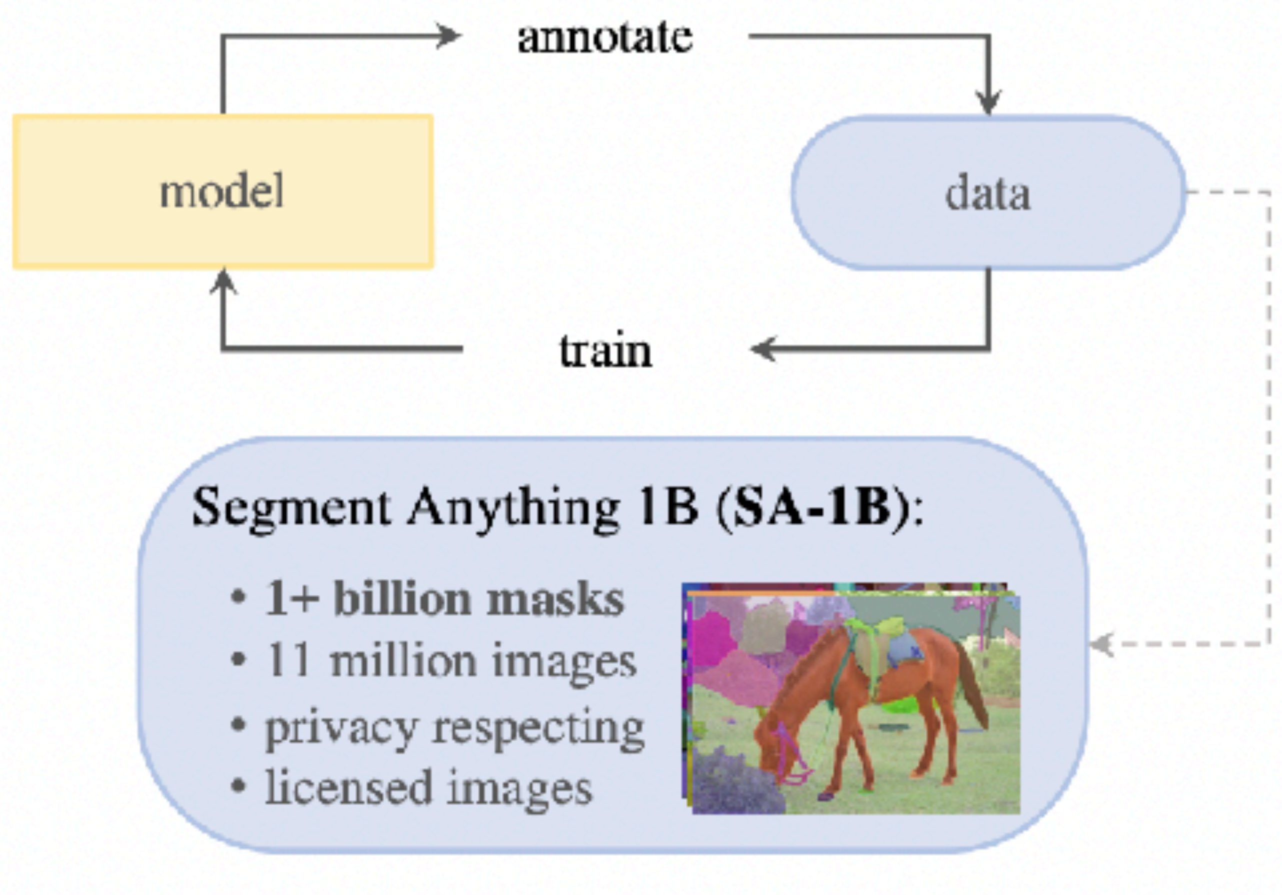
[Kirillov et al., Segment Anything, ICCV 2023 (Best Paper Honorable Mention)]

Segment Anything Dataset (SA-1B)

- 11M images
- 1B+ masks (99.1% of masks fully automatic)
- Collected through interactive interface

3-stage annotation:

- Assisted-manual stage (+30sec/image to annotate, reduced to 14sec after 6 x retraining, 4.3M masks from 12K images)
- Semi-automatic stage (bbox for less prominent objects, up to 34sec. 5 x retraining, 5.9M masks in 180K images)
- Fully-automatic stage.



(c) **Data**: data engine (top) & dataset (bottom)

[Kirillov et al., Segment Anything, ICCV 2023 (Best Paper Honorable Mention)]

Segment Anything

- Spatial distribution of object centers
- Common photographer bias
- Greater coverage of image corners in SA-1B

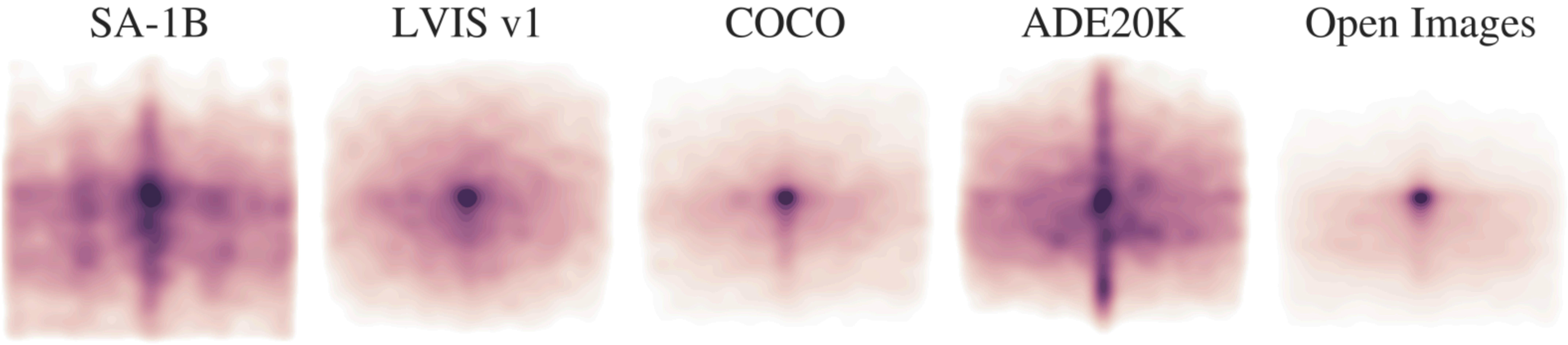
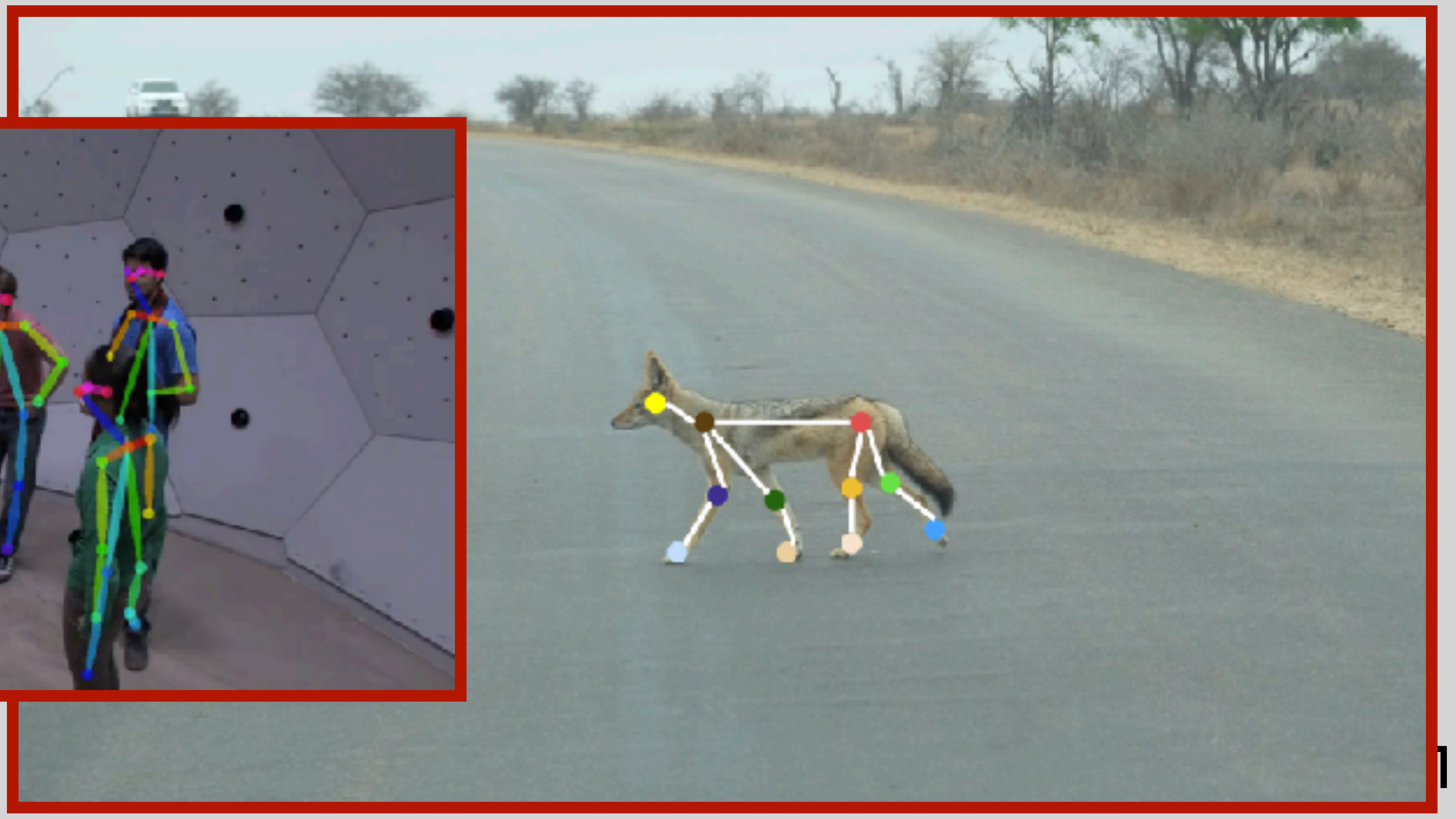


Image-size normalized mask center distributions

[Kirillov et al., Segment Anything, ICCV 2023 (Best Paper Honorable Mention)]

Agenda

- 0. Intro to structured outputs
- 1. Object detection (localization)
- 2. Segmentation
- 3. Human pose estimation

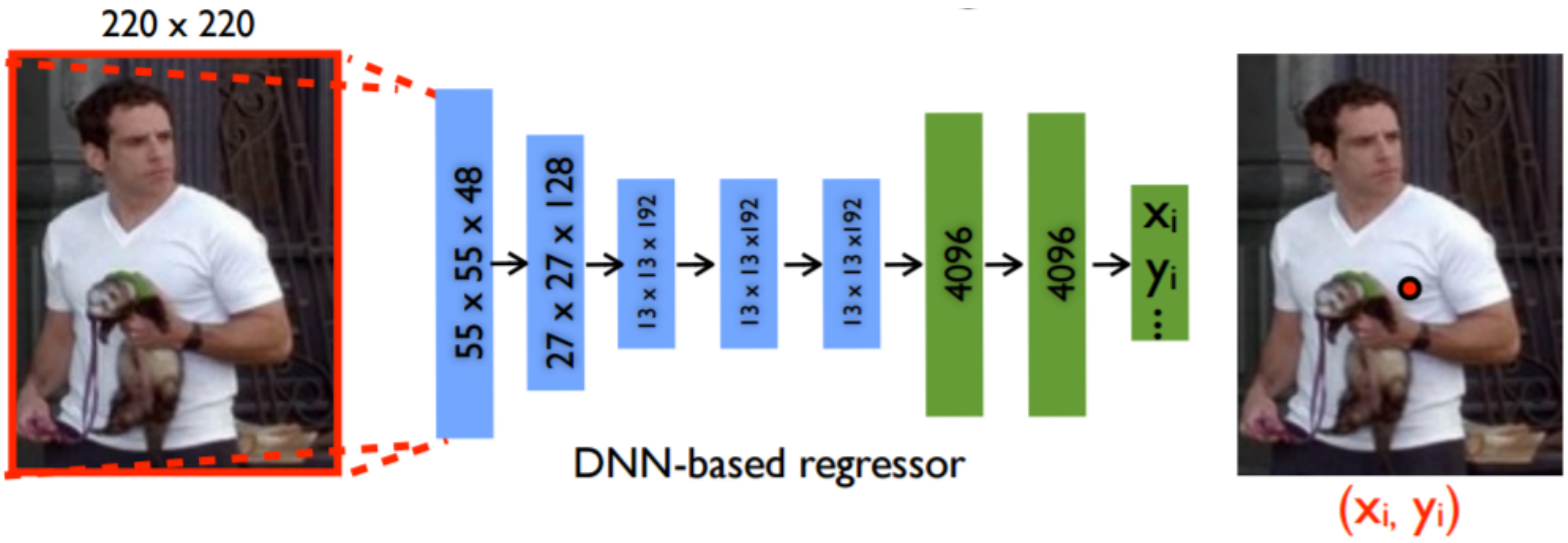


2D Human pose estimation

[Cao, Simon, Wei and Sheikh CVPR 2017]

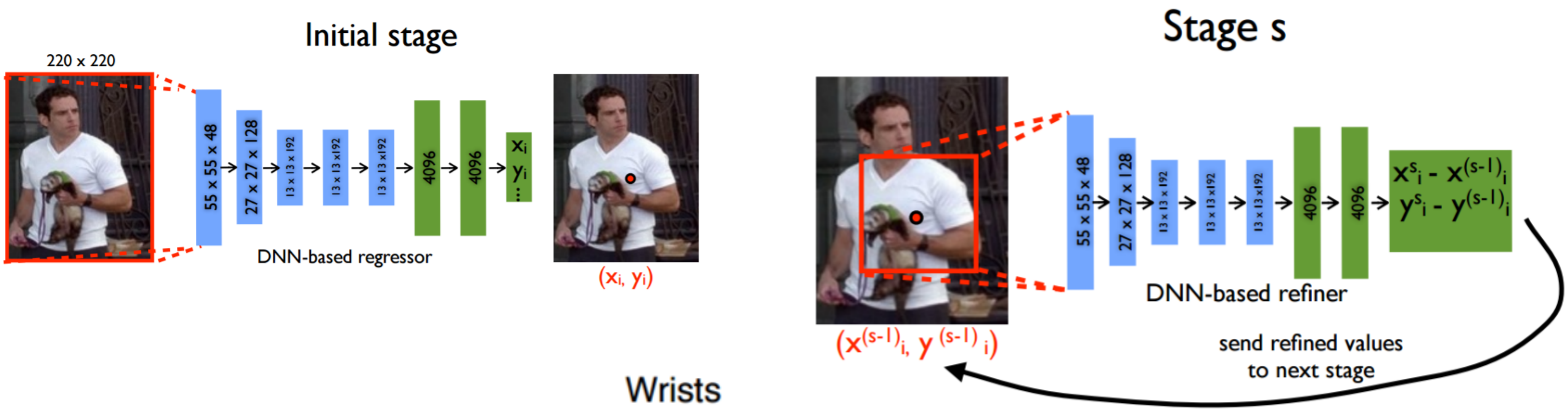
DeepPose: Human Pose Estimation via Deep Neural Networks

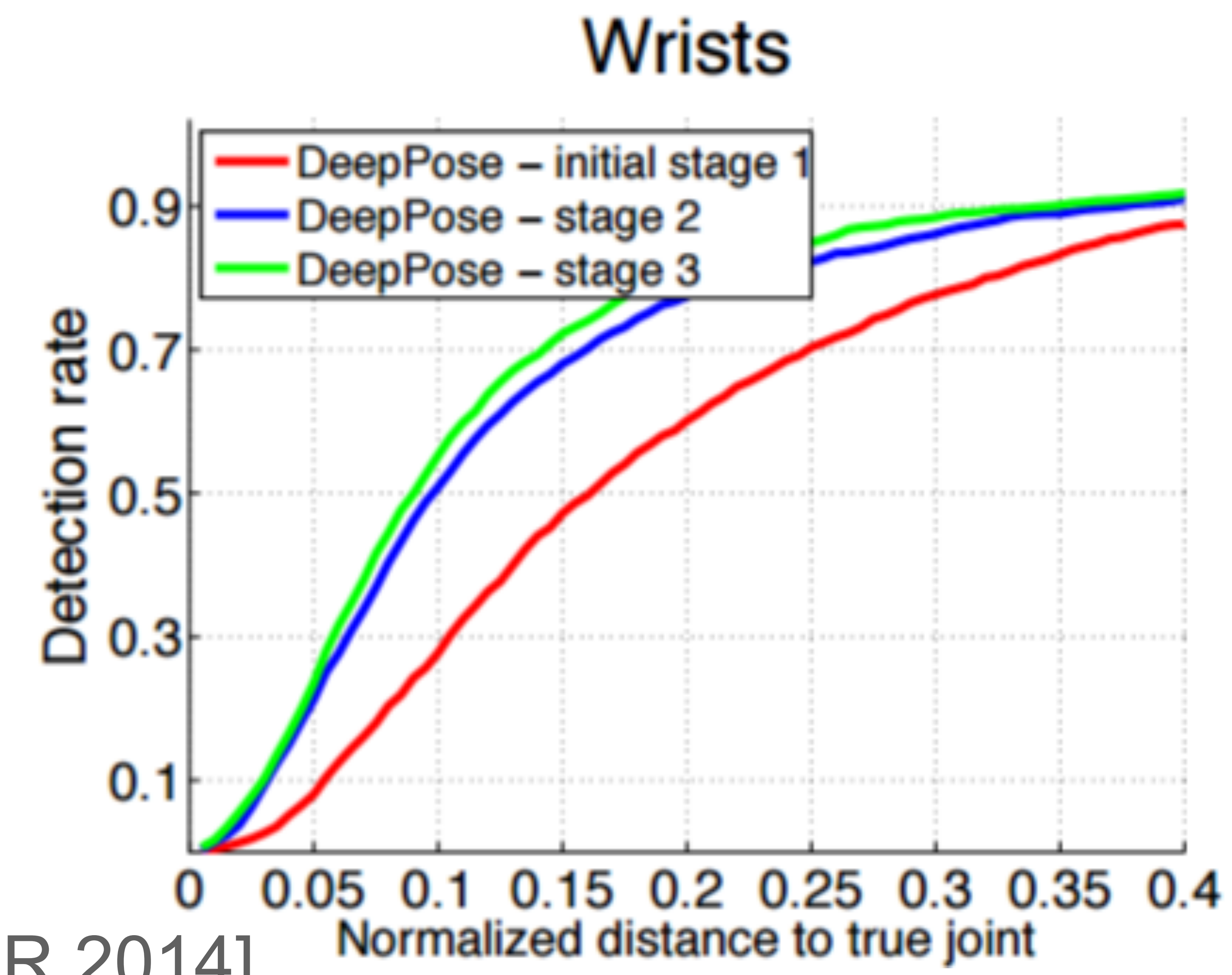
Trains CNN to **regress locations (x_i, y_i)** for each **joint i**



[Toshev and Szegedy, CVPR 2014]

DeepPose: Human Pose Estimation via Deep Neural Networks





- **Cascade regressor:**
  Stage s improves output of the previous stage s-1 using higher resolution sub-image
- **3 stages in practice**

[Toshev and Szegedy, CVPR 2014]

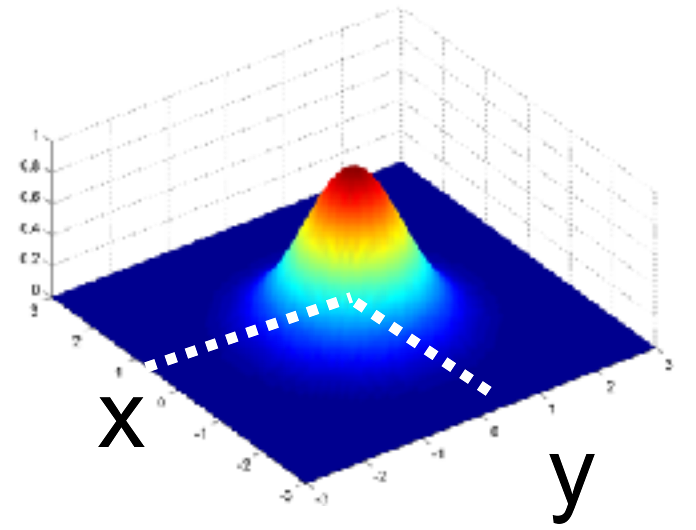
DeepPose: Human Pose Estimation via Deep Neural Networks

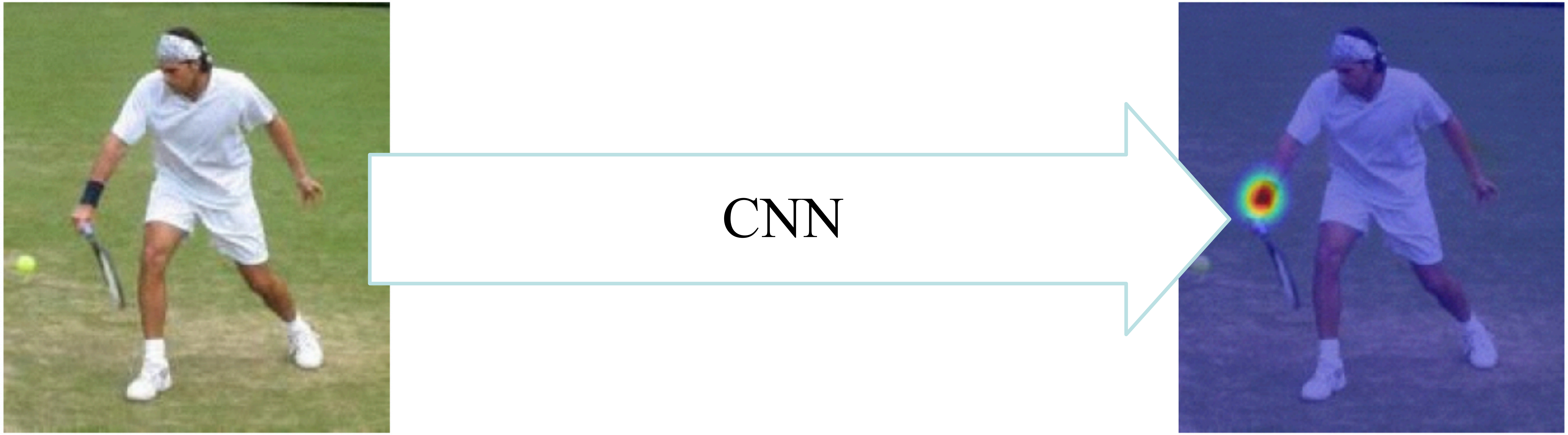
[Toshev and Szegedy, CVPR 2014]

Convolutional Pose Machines

- Regression to joint “heatmaps”: 2D gaussians around joint coordinates
- Heatmaps enable to handle spatial ambiguity

x, y





Input Image

Heatmap
for right elbow

[Wei, Ramakrishna, Kanade and Sheikh, CVPR 2016]

Convolutional Pose Machines

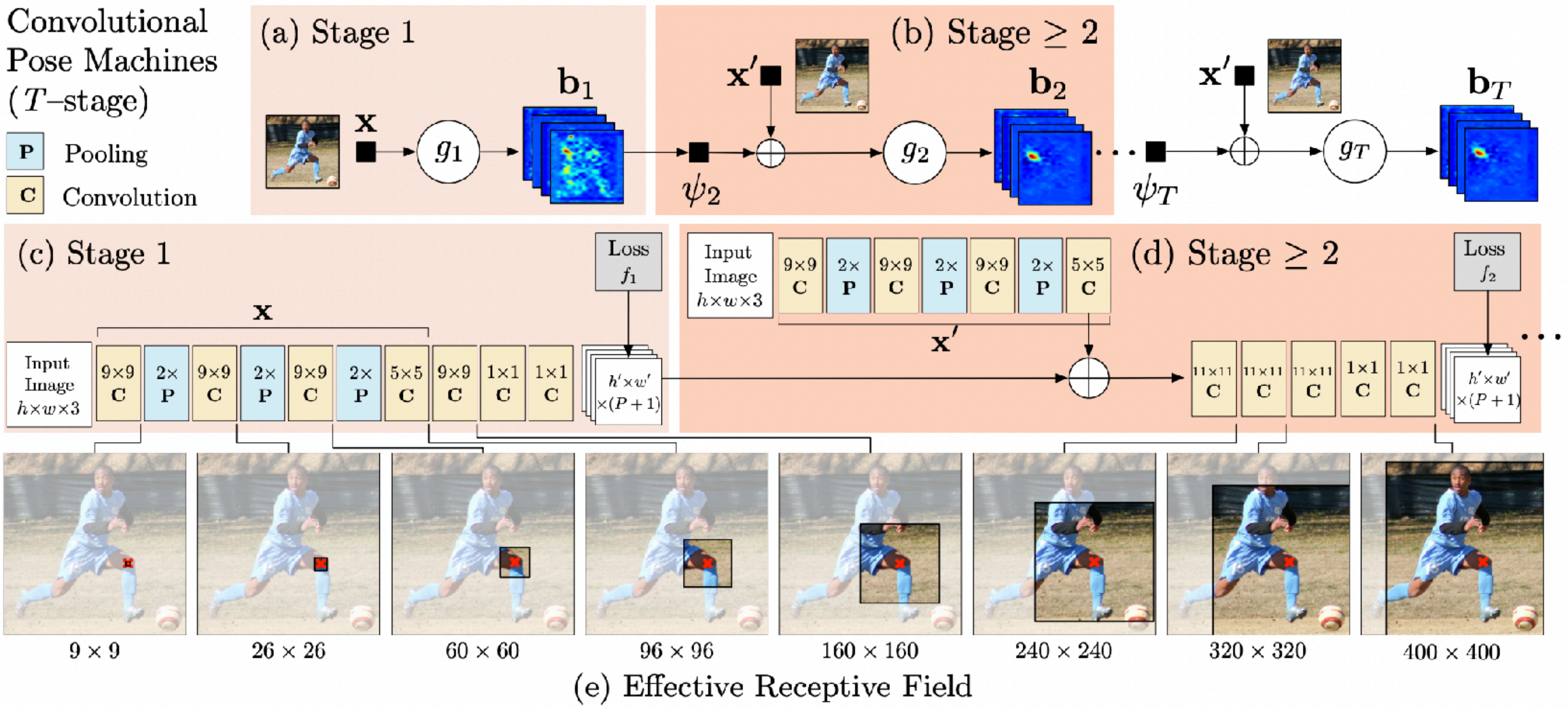
- Regression to joint "heatmaps": 2D gaussians around joint coordinates
- Heatmaps enable to handle spatial ambiguity
- Multi-stage refinement

Input Image | (a) Stage 1 | (b) Stage 2 | (c) Stage 3

[Wei, Ramakrishna, Kanade and Sheikh, CVPR 2016]

Convolutional Pose Machines

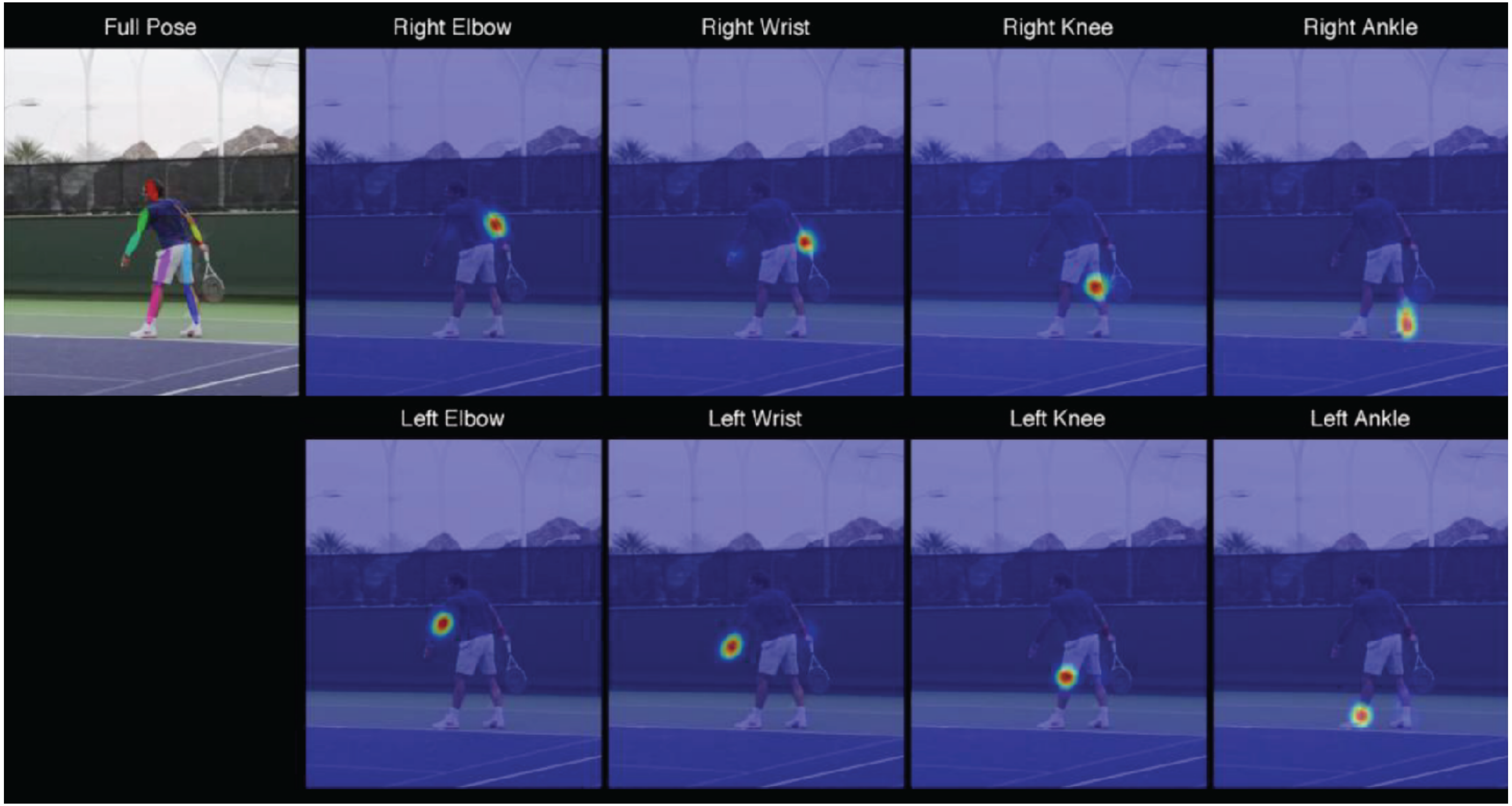
- Intermediate supervision at every stage; Increasing context



[Wei, Ramakrishna, Kanade and Sheikh, CVPR 2016]

Convolutional Pose Machines

Qualitative results



[Wei, Ramakrishna, Kanade and Sheikh, CVPR 2016]

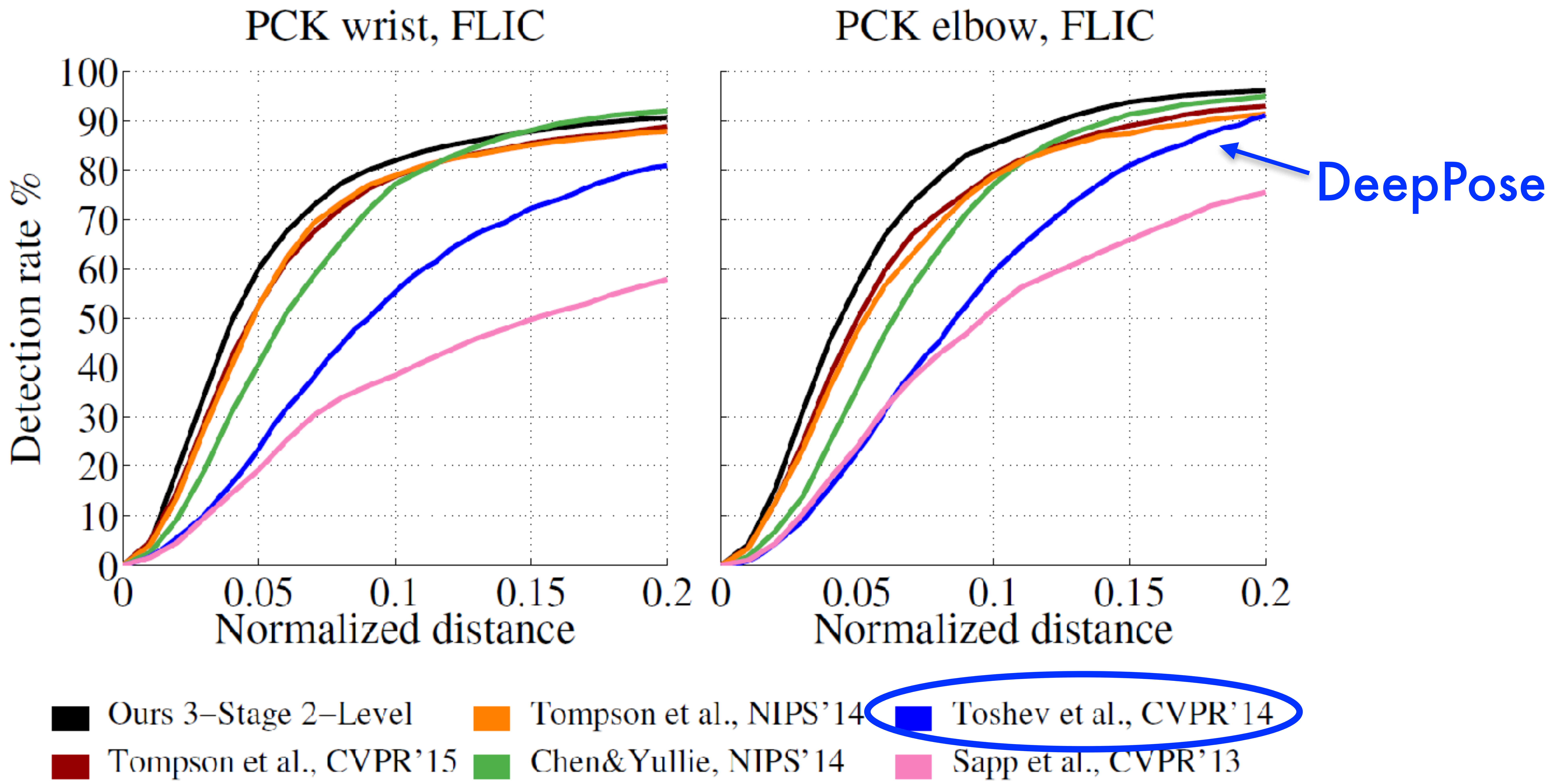
Convolutional Pose Machines

Qualitative results

[Wei, Ramakrishna, Kanade and Sheikh, CVPR 2016]

Convolutional Pose Machines

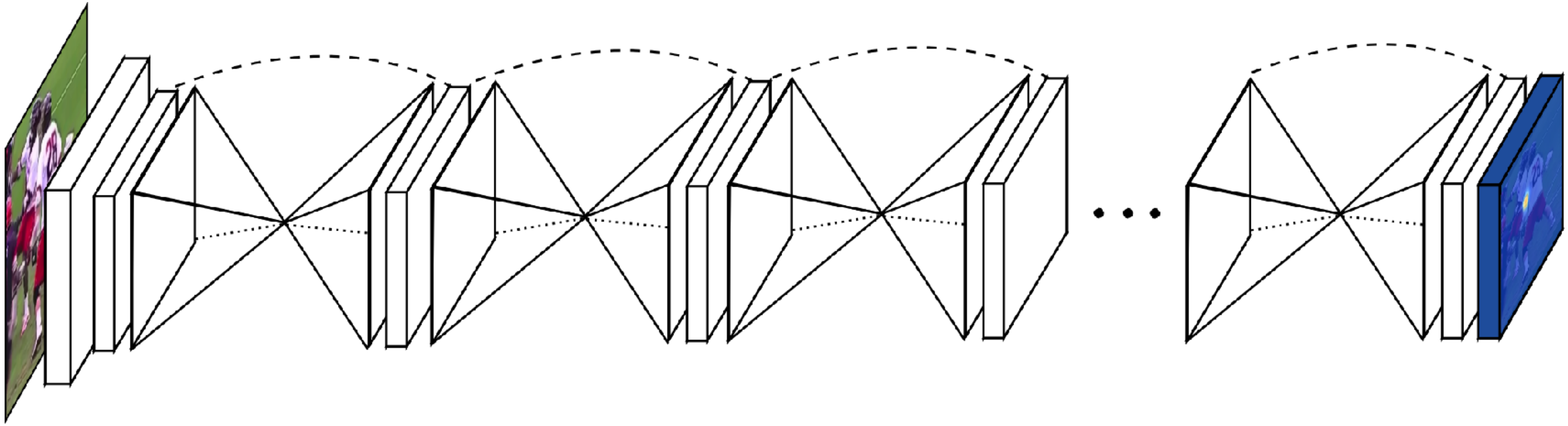
Quantitative comparison



[Wei, Ramakrishna, Kanade and Sheikh, CVPR 2016]

Stacked Hourglass Networks

Remember U-Net

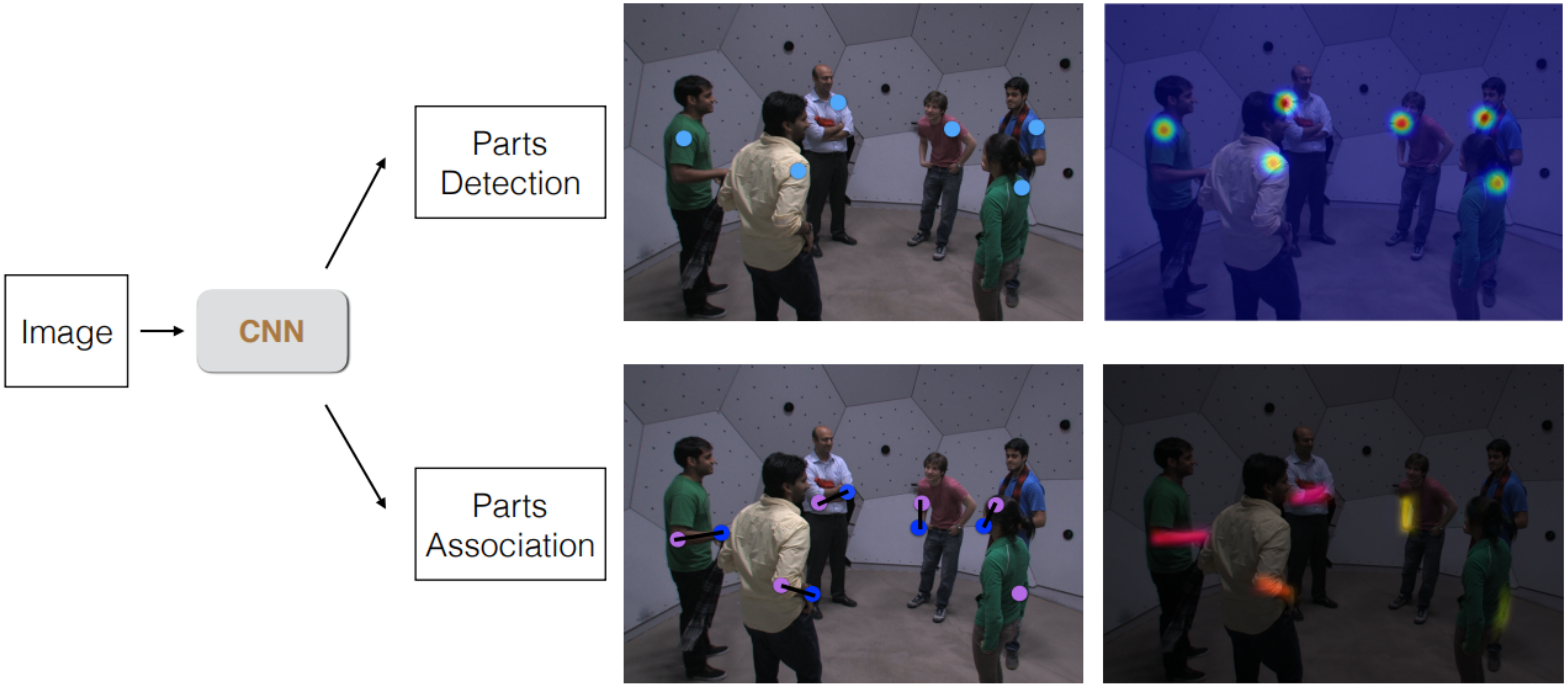


- Also heatmap regression
- Also multi-stage refinement - but full context (receptive field = entire image)

[Newell, Yang, Deng, ECCV 2016]

OpenPose: Multi-person pose estimation

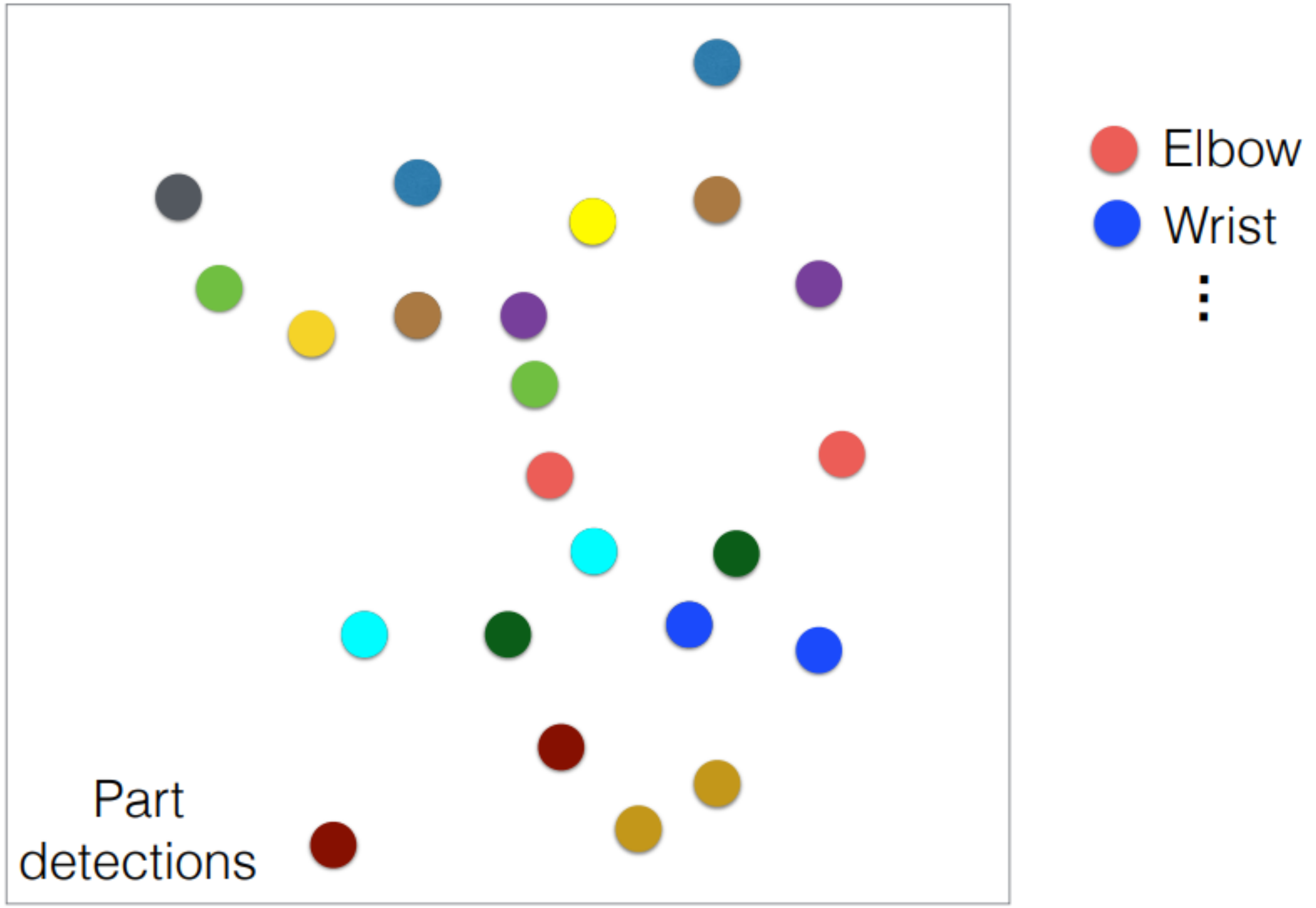
Novelty: Jointly Learning Parts Detection and Parts Association



[Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. Sheikh, OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, TPAMI 2019]
[Z. Cao, T. Simon, S. Wei, and Y. Sheikh, Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, CVPR 2017]

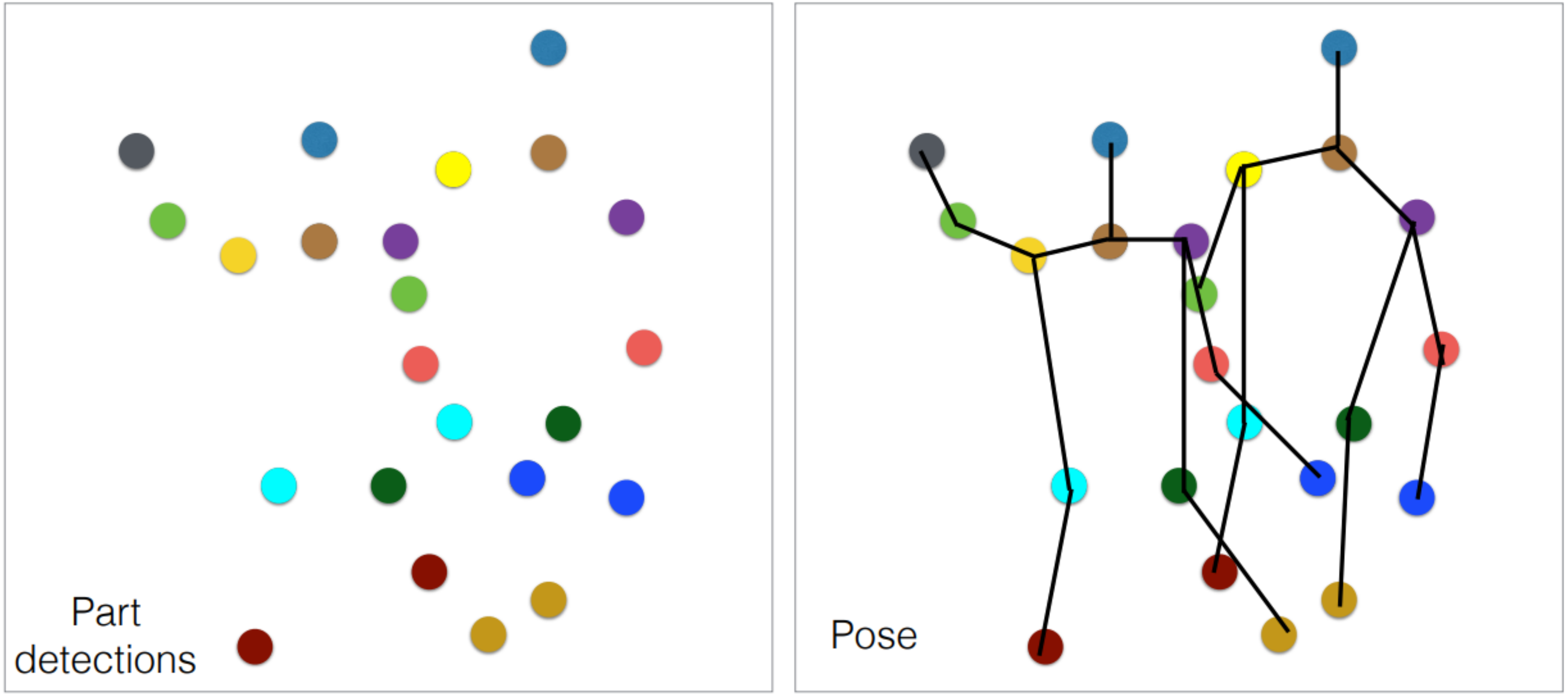
OpenPose

Part-Person Association for Multi-Person Pose Estimation

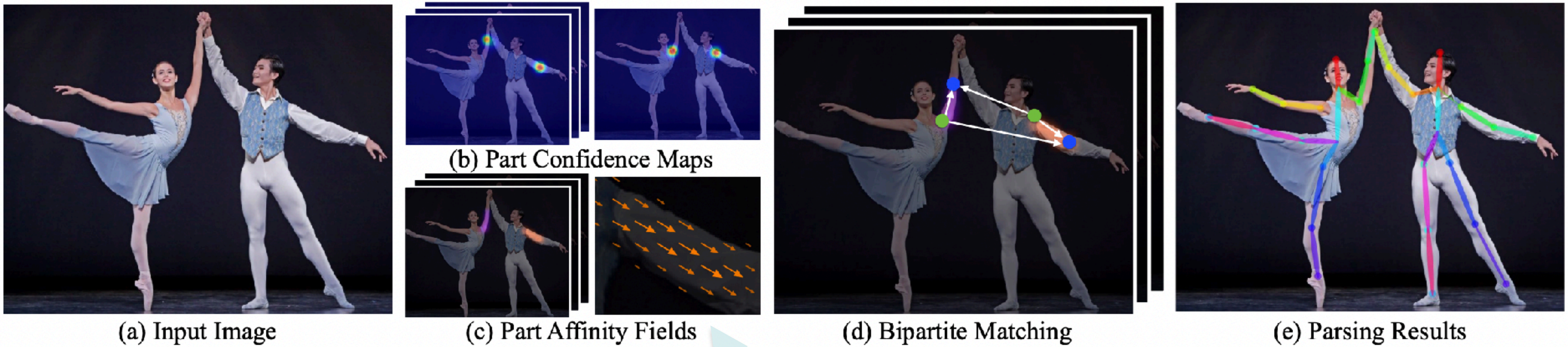


OpenPose

Part-Person Association for Multi-Person Pose Estimation



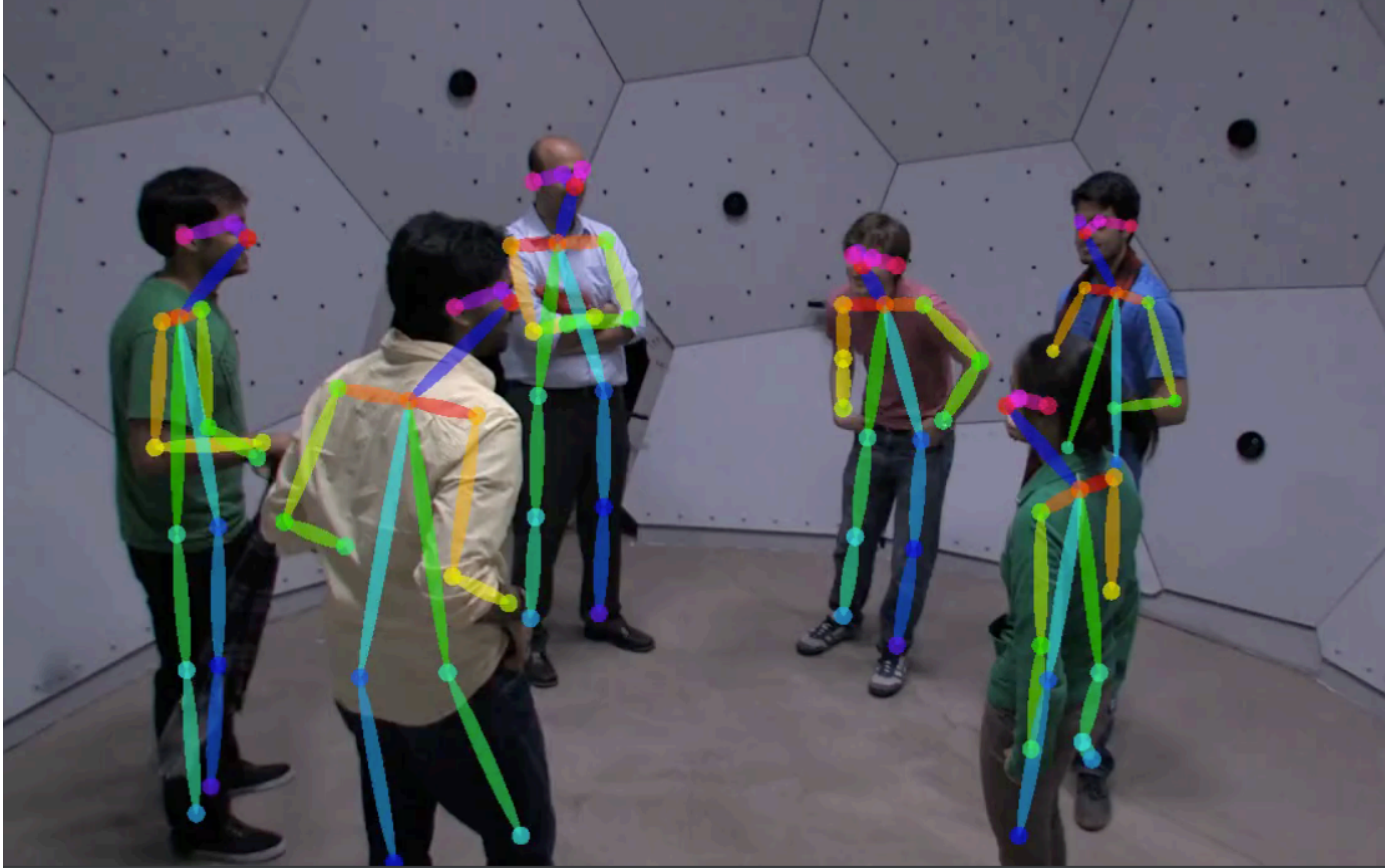
OpenPose



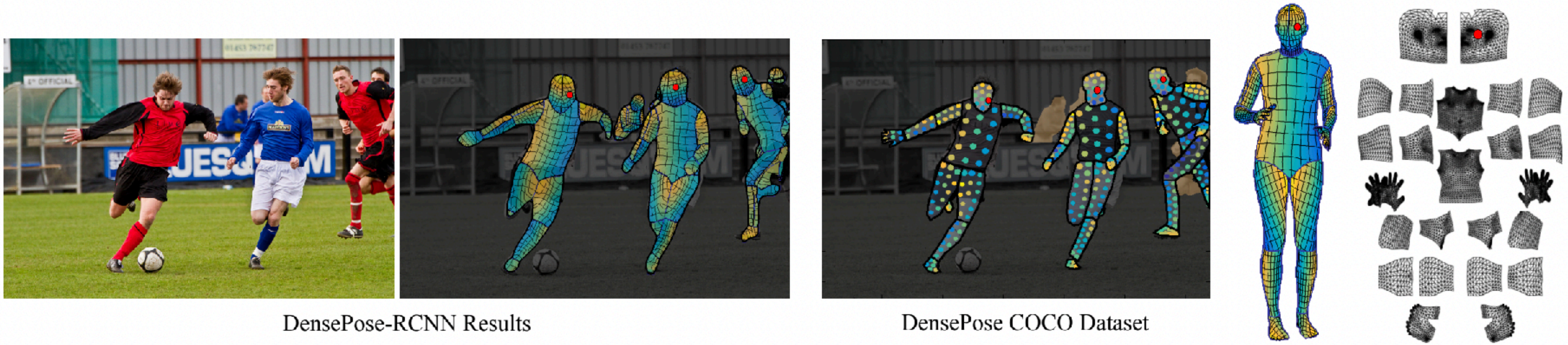
(a) Input Image (c) Part Affinity Fields (d) Bipartite Matching (e) Parsing Results

Figure 2. Overall pipeline. Our method takes the entire image as the input for a two-branch CNN to jointly predict confidence maps for body part detection, shown in (b), and part affinity fields for parts association, shown in (c). The parsing step performs a set of bipartite matchings to associate body parts candidates (d). We finally assemble them into full body poses for all people in the image (e).

Key Idea: Encode the Part Affinity Score on the Image Plane
=> Part Affinity Fields encode direction and position



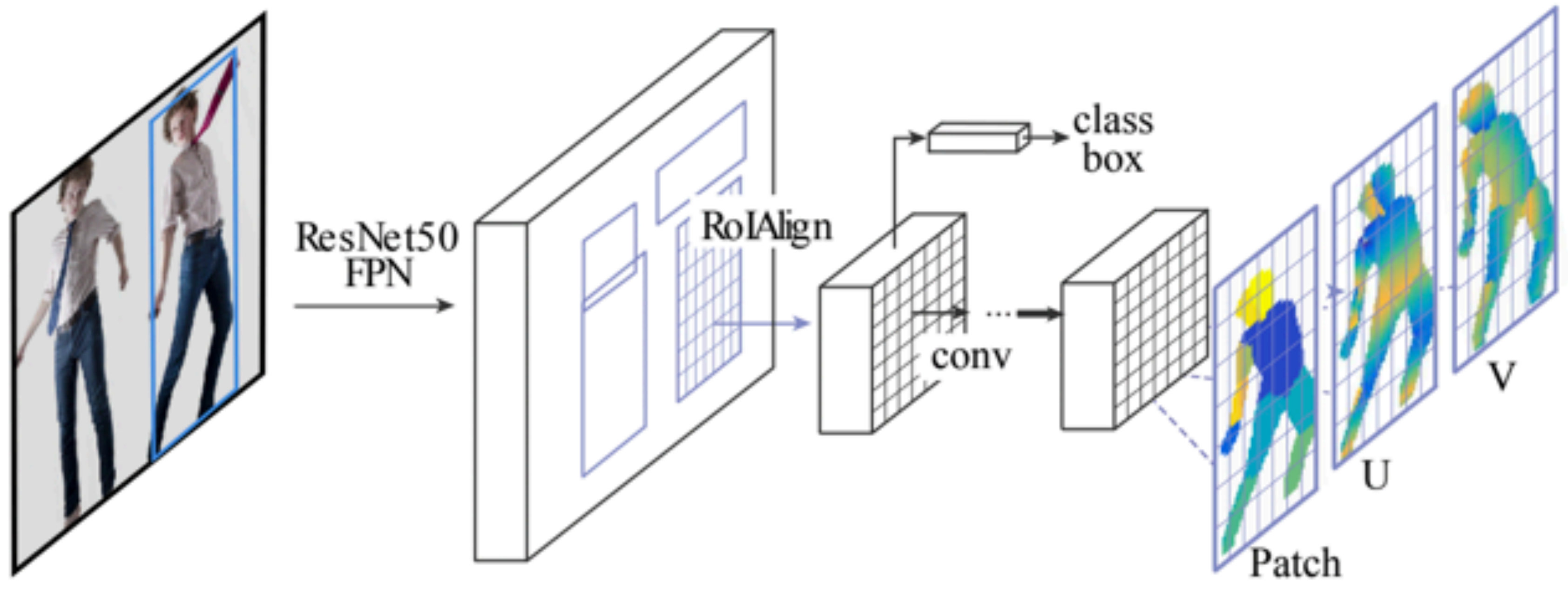
DensePose: Dense Human Pose Estimation



DensePose-RCNN Results

DensePose COCO Dataset

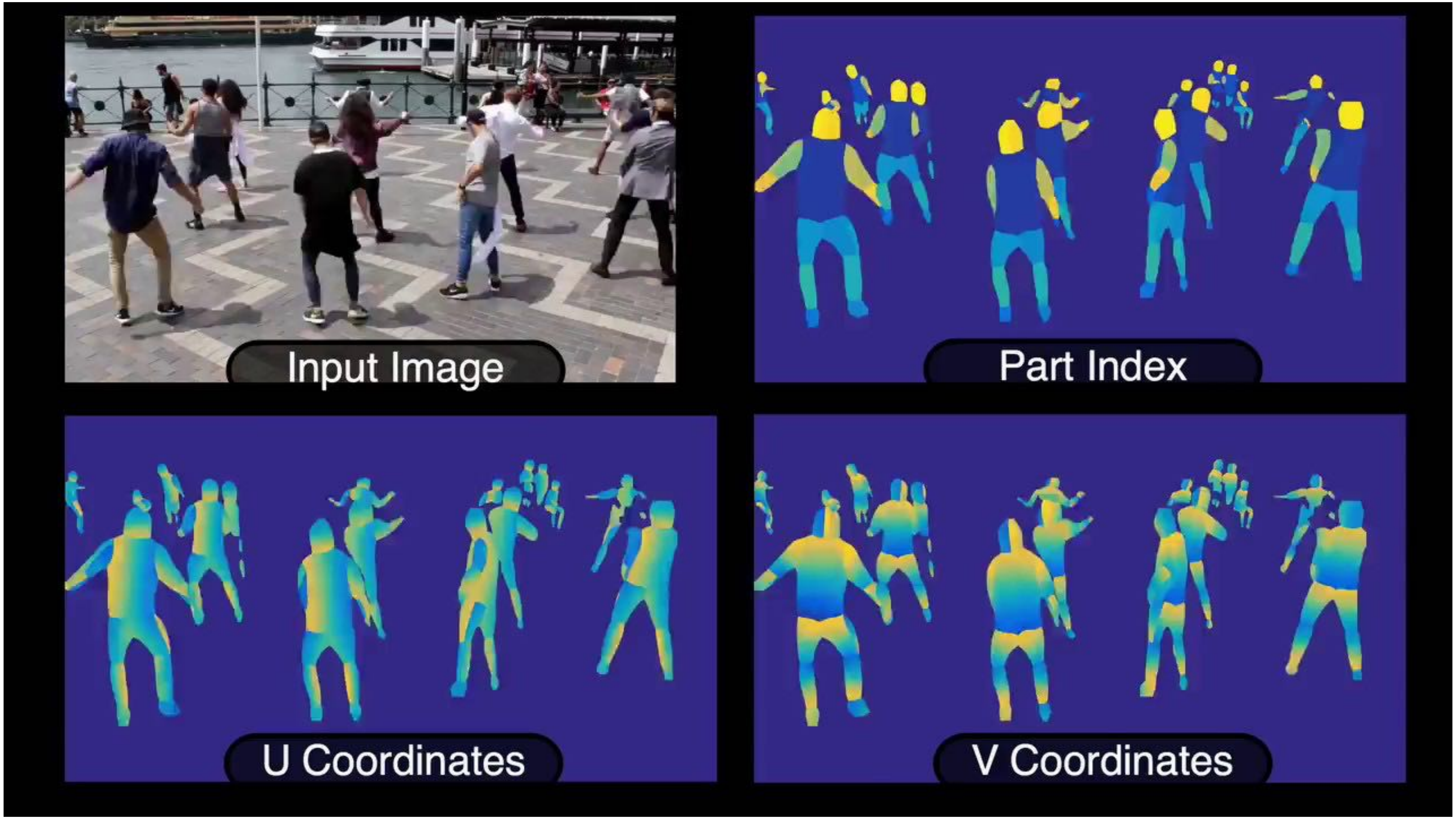
Dense pose estimation aims at mapping all human pixels of an RGB image to the 3D surface of the human body.



regresses to continuous surface coordinates

Guler et al. DensePose, CVPR 2018

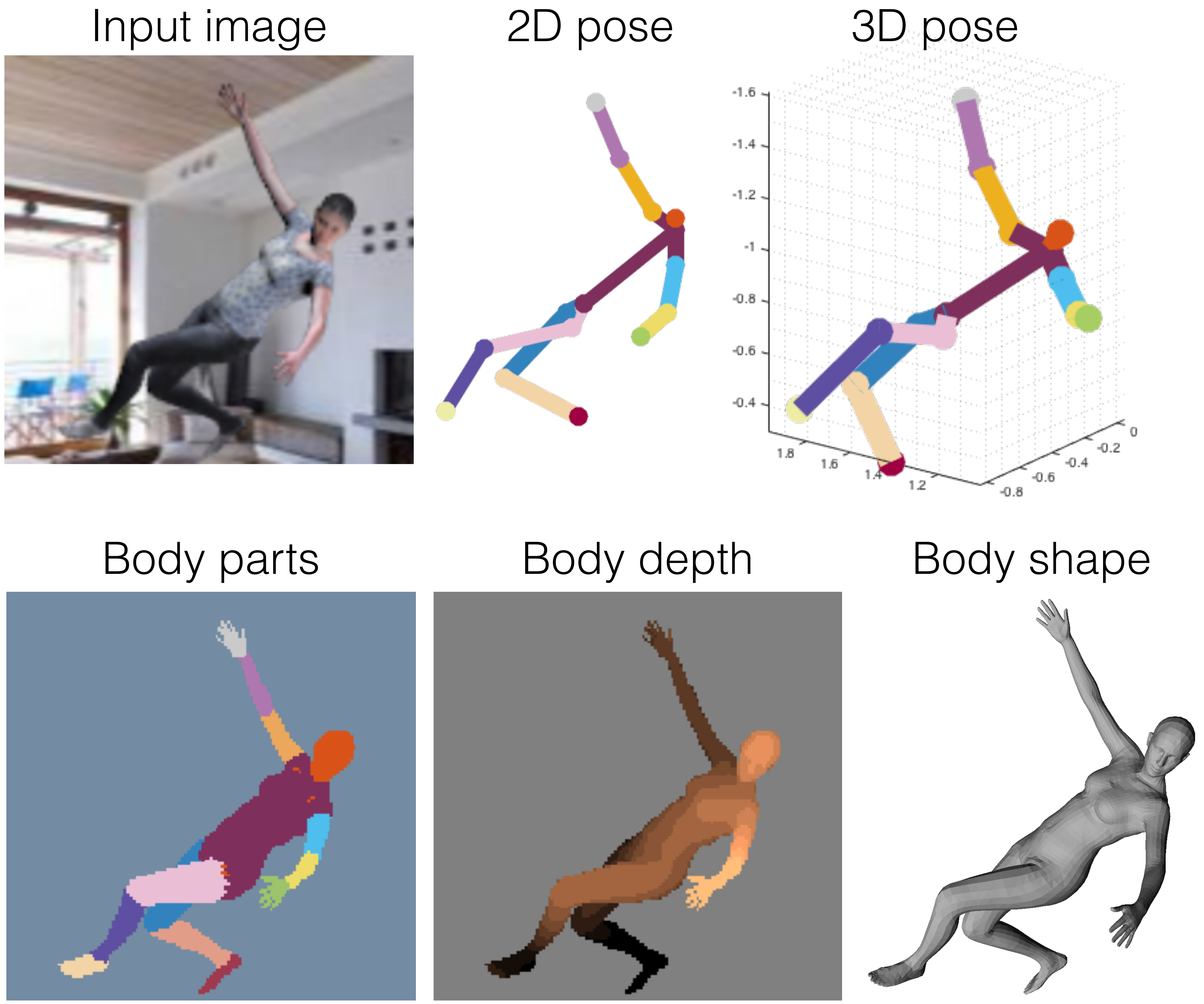
DensePose



Guler et al. DensePose, CVPR 2018

Human pose estimation beyond 2D keypoints

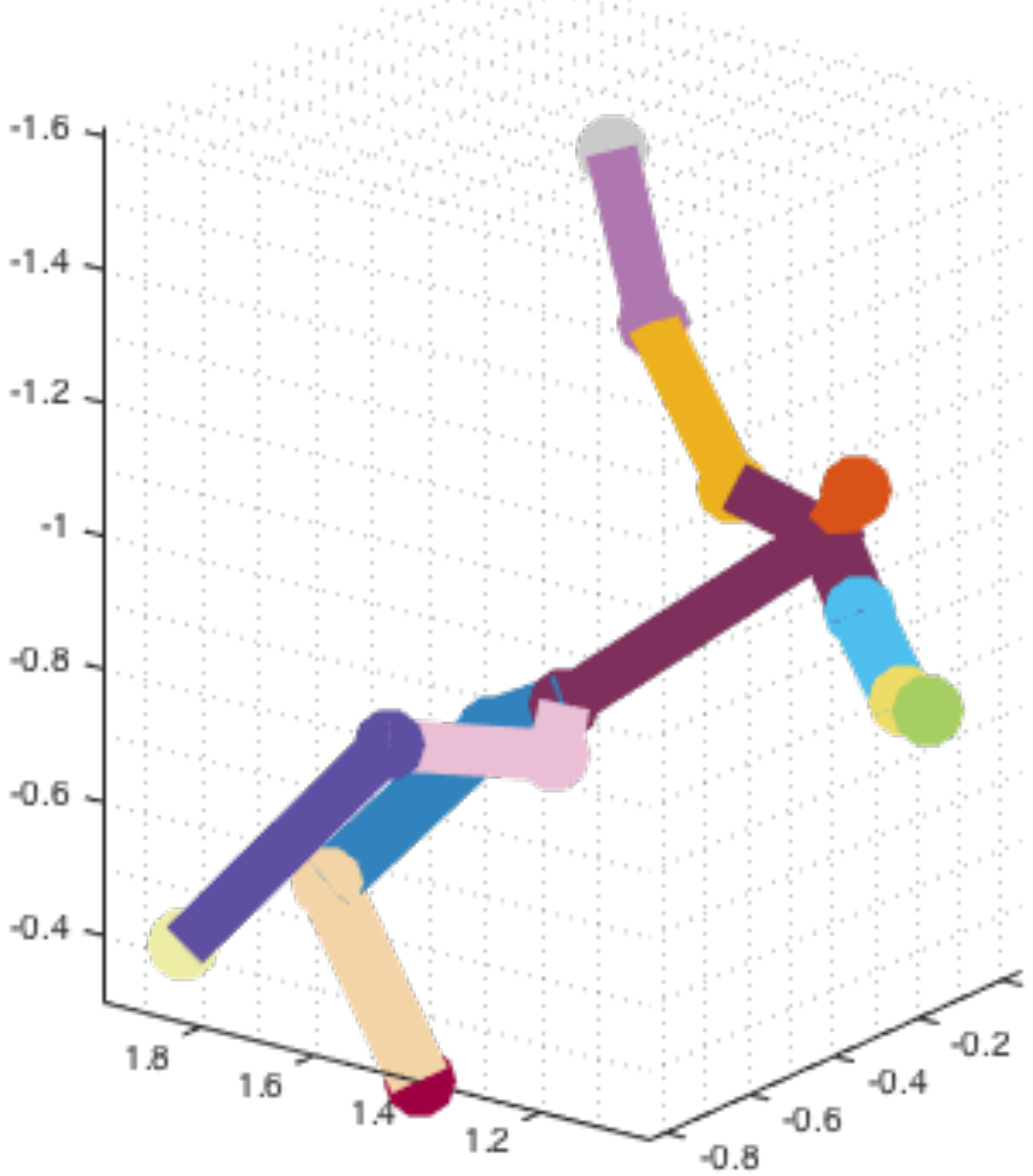
Human body analysis



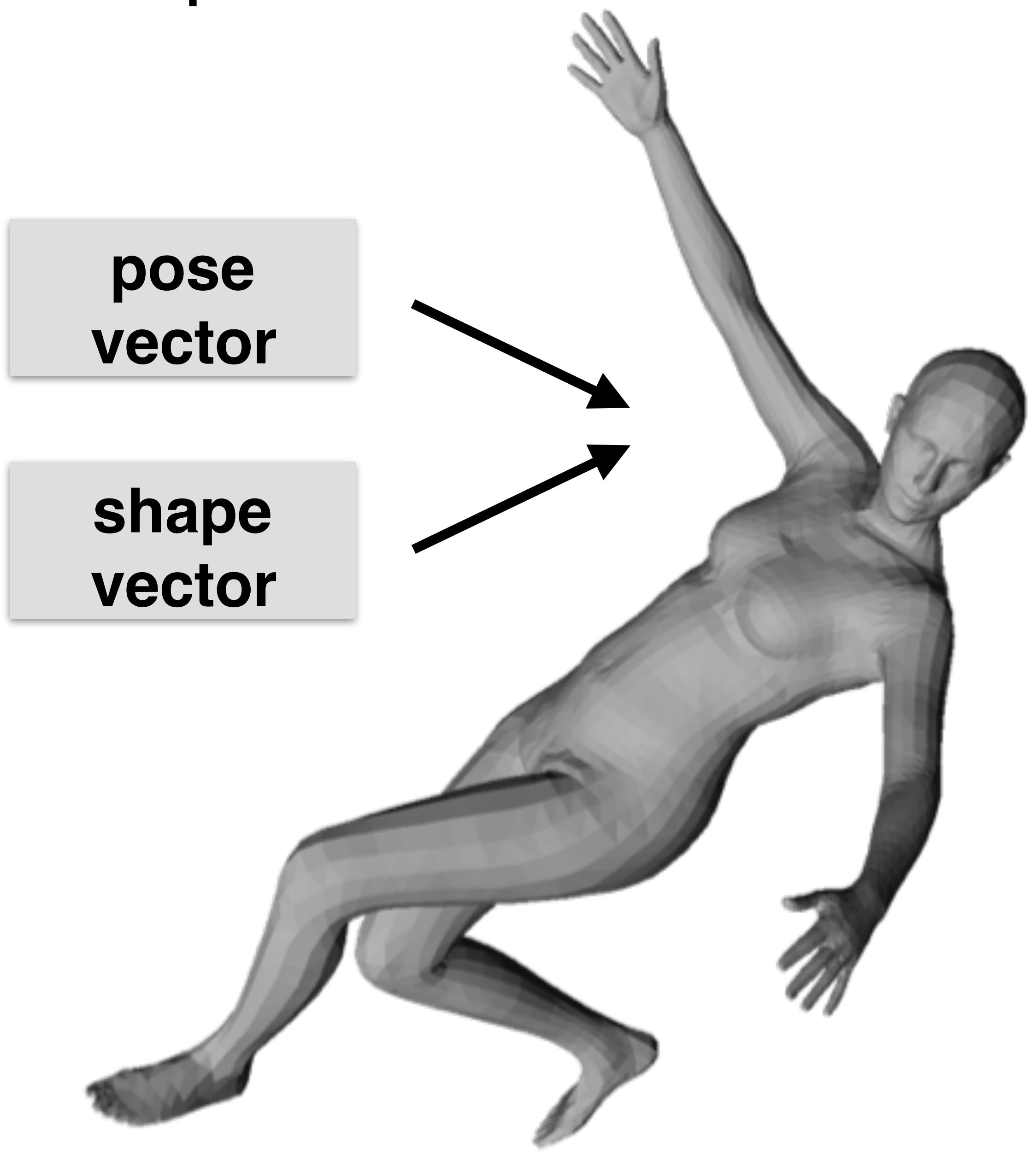
Challenges

How to model the body shape?

(a) Skeleton representation

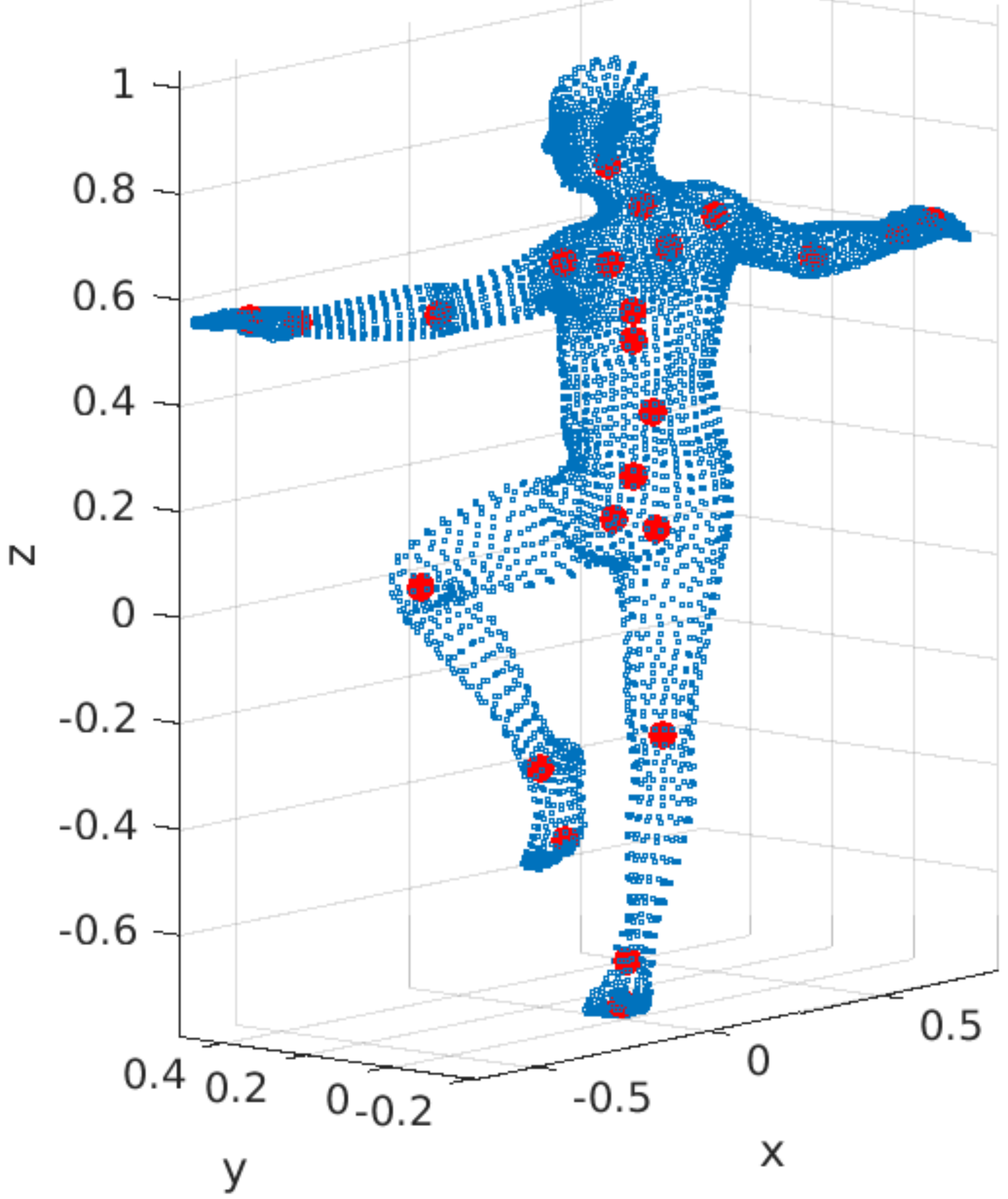


(b) Parametric representation

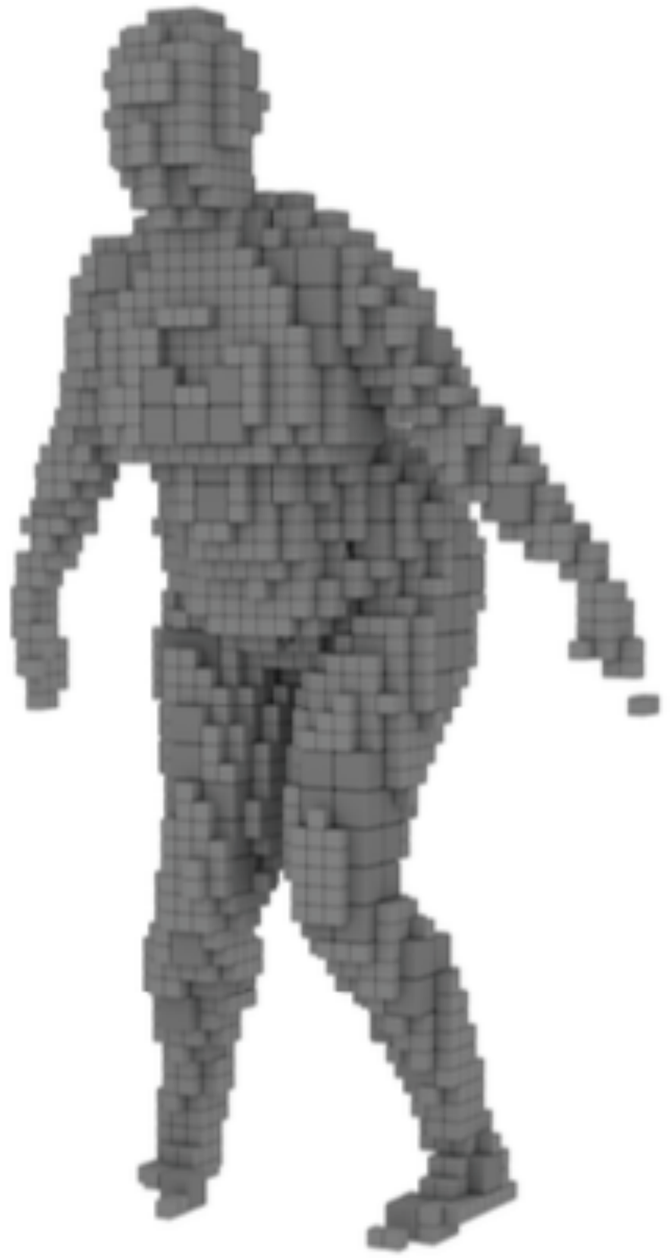


Loper et al. [1]

(c) Point cloud representation



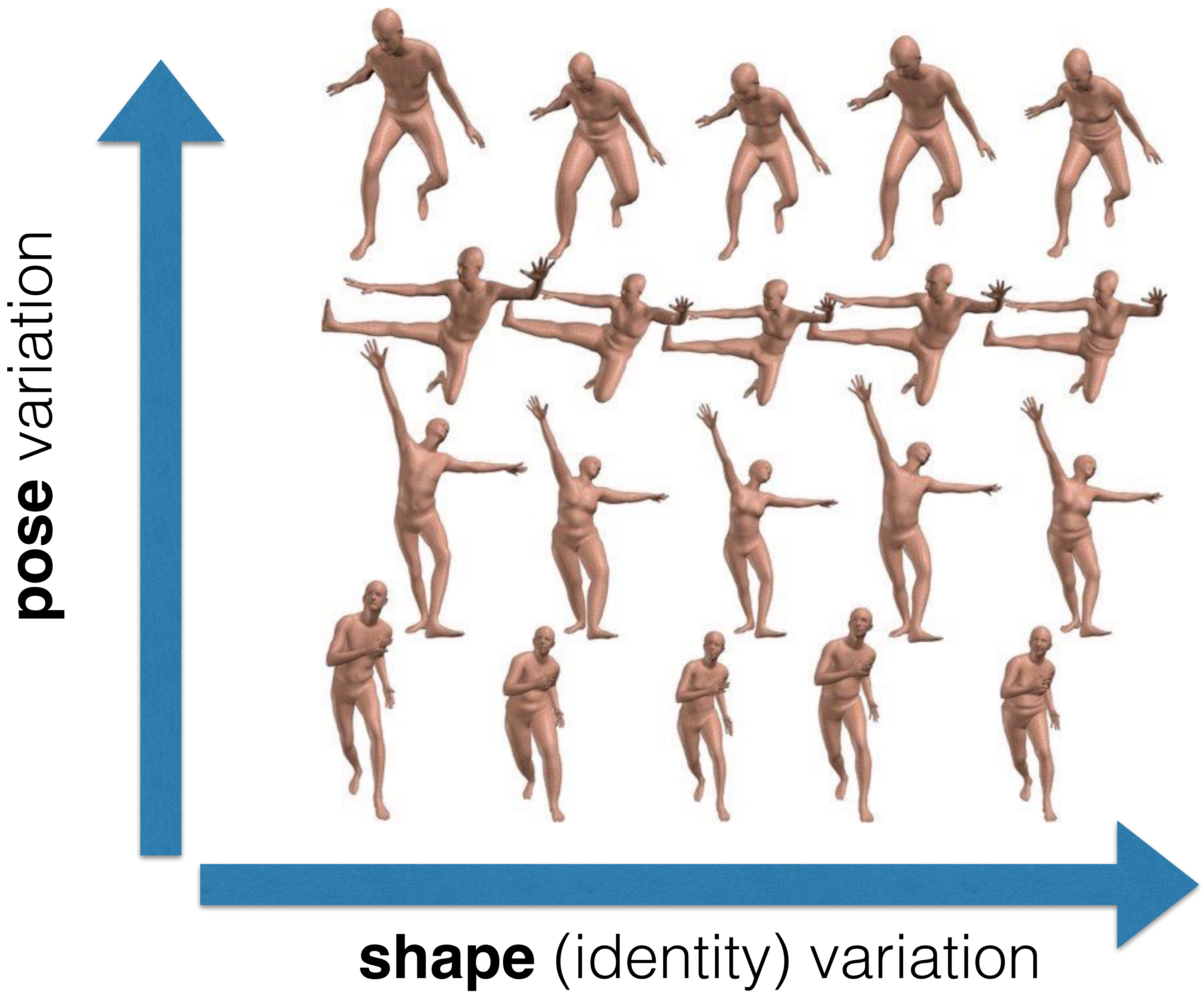
(d) Voxel representation



Tatarchenko et al. [2]

[1] Loper et al. SMPL: A Skinned Multi-Person Linear Model, SIGGRAPH Asia 2015
[2] Tatarchenko et al. Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs, ICCV 2017

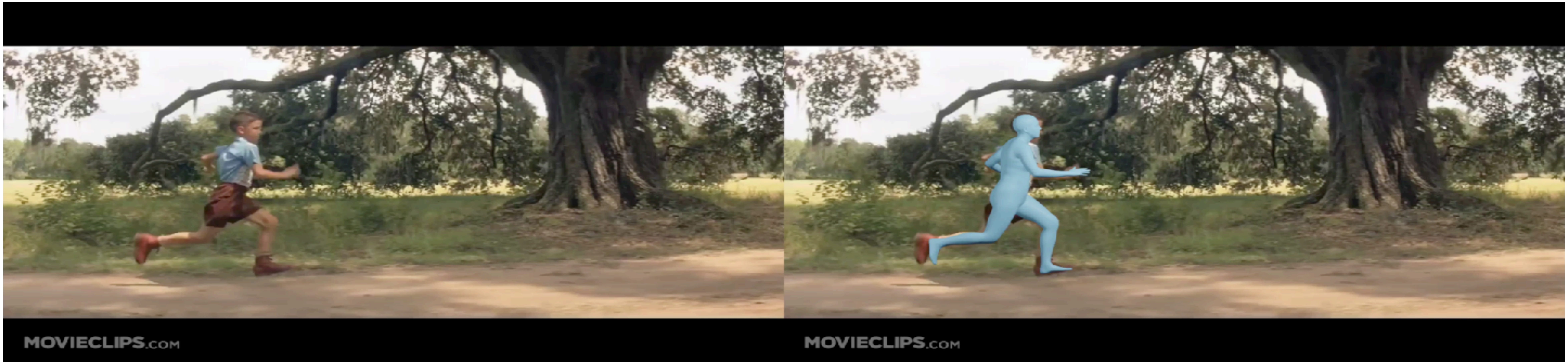
SMPL parametric body model: surface & joints



[Loper et al. 2015]

Human pose estimation beyond 2D keypoints

- A rich literature also on 3D human pose & motion estimation



VIBE [Kocabas et al. CVPR 2020]

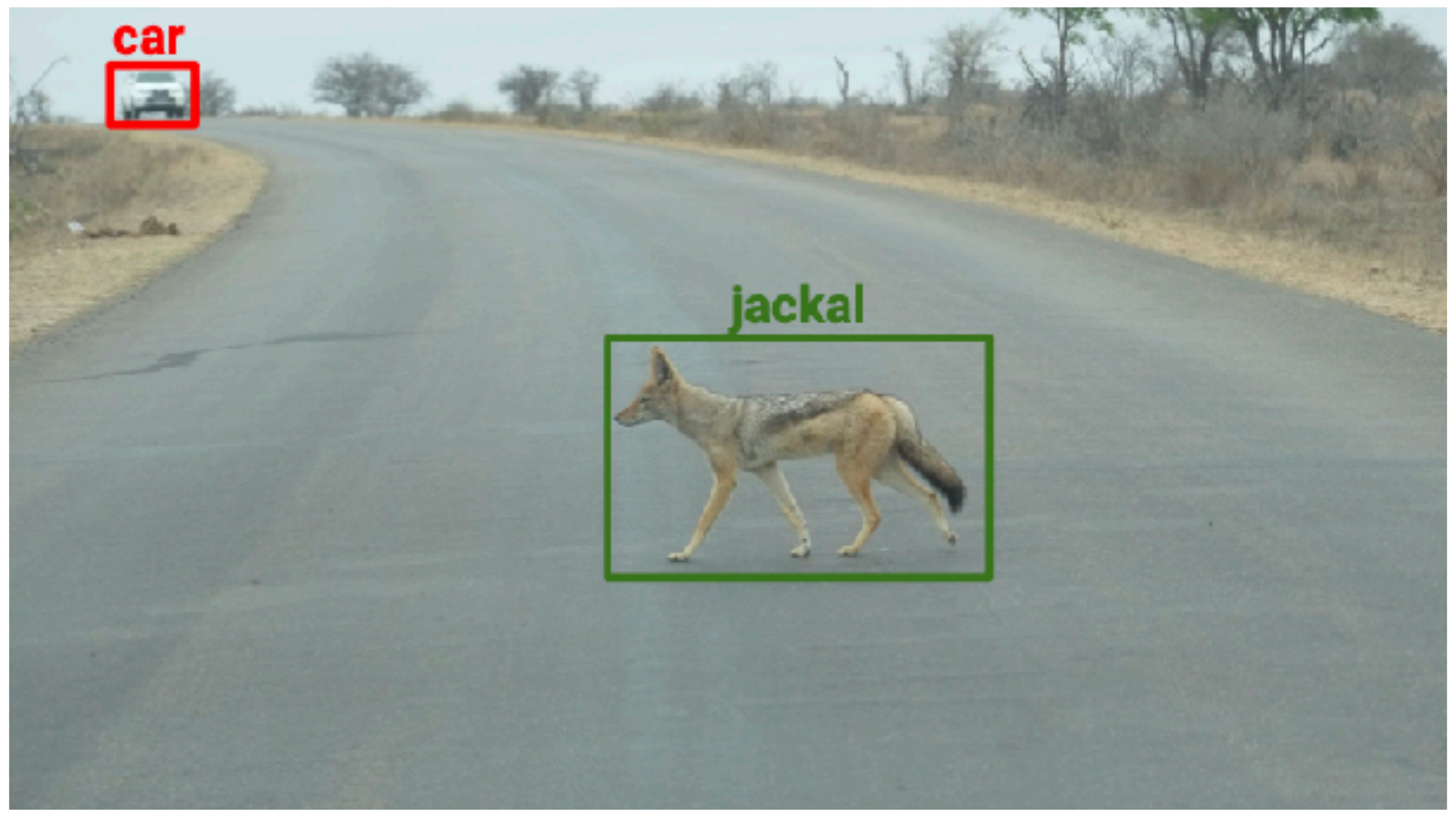
Human pose estimation beyond 2D keypoints

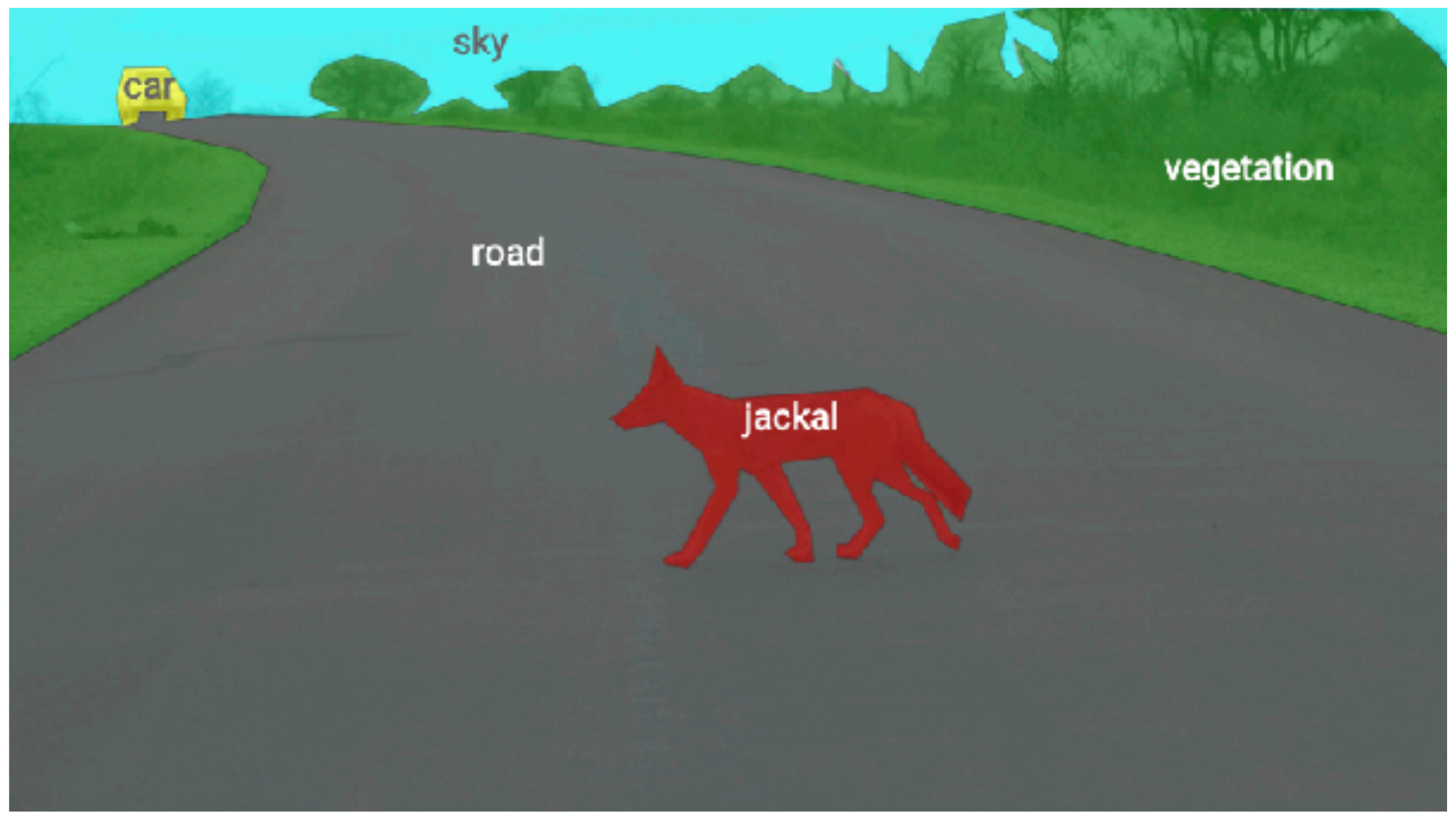
- A rich literature also on 3D human pose & motion estimation

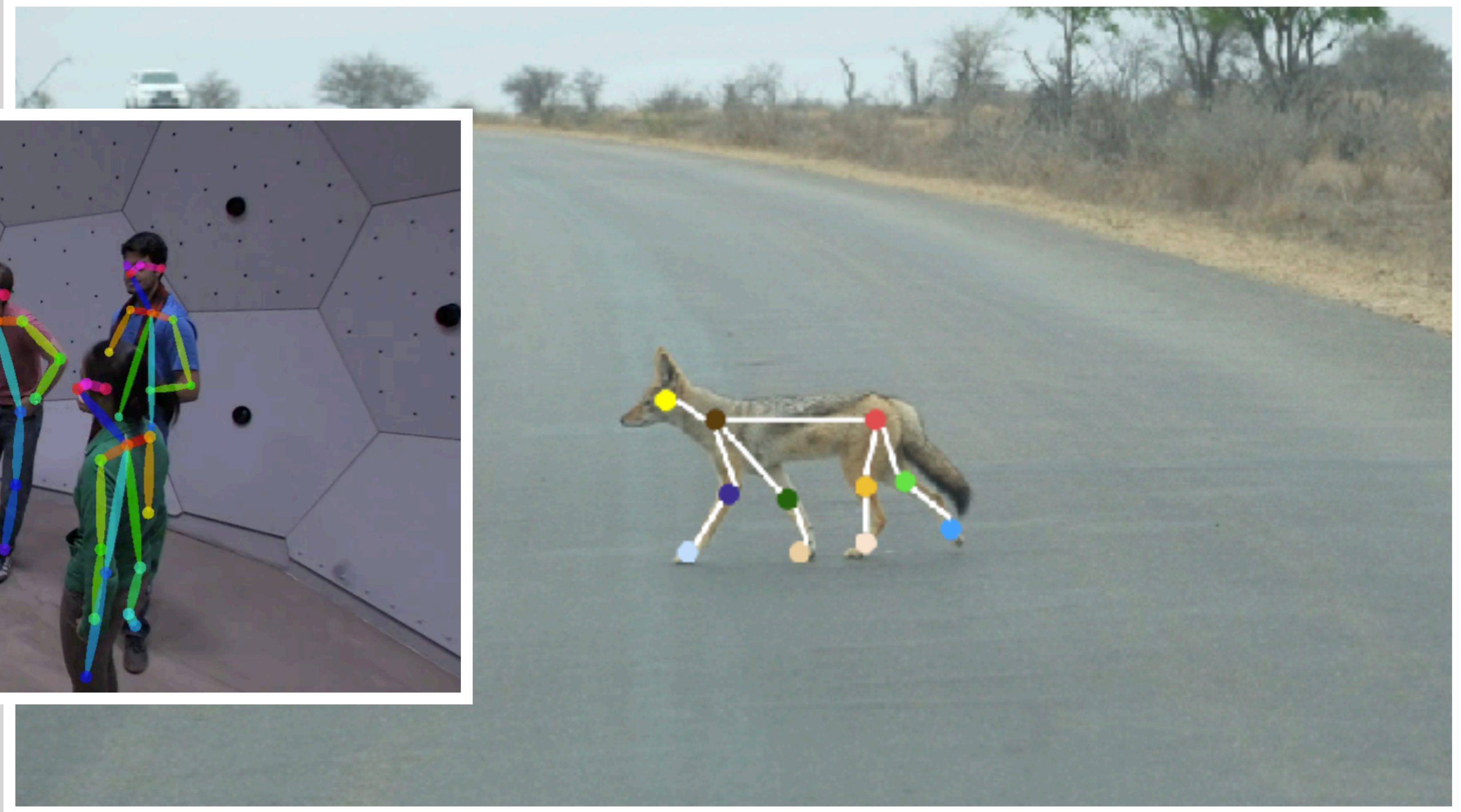
4D Humans [Goel et al. ICCV 2023]

Agenda

- 0. Intro to structured outputs
- 1. Object detection (localization)
- 2. Segmentation
- 3. Human pose estimation







Feedback welcome throughout the course (anonymous)
Can fill the form multiple times