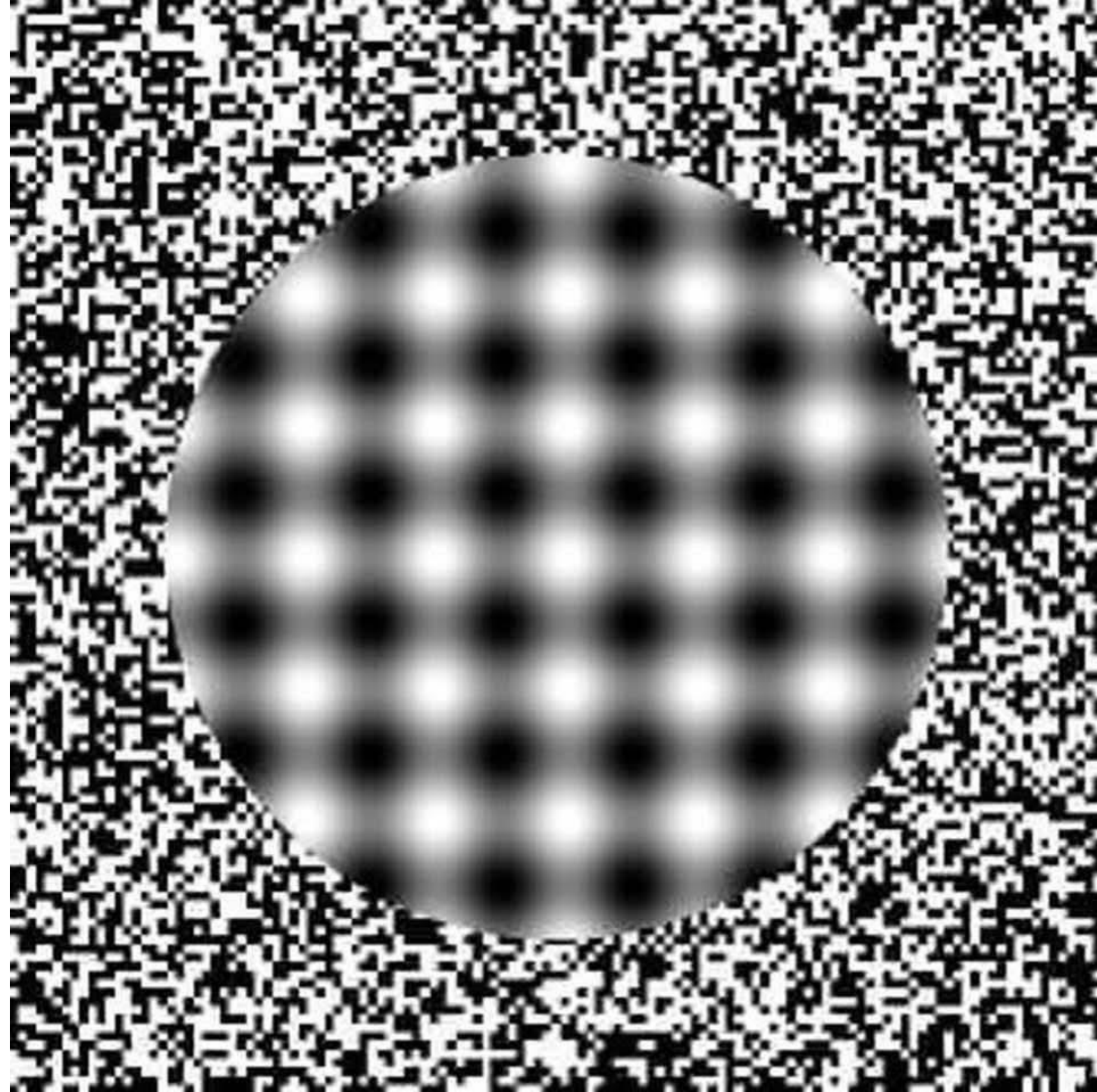
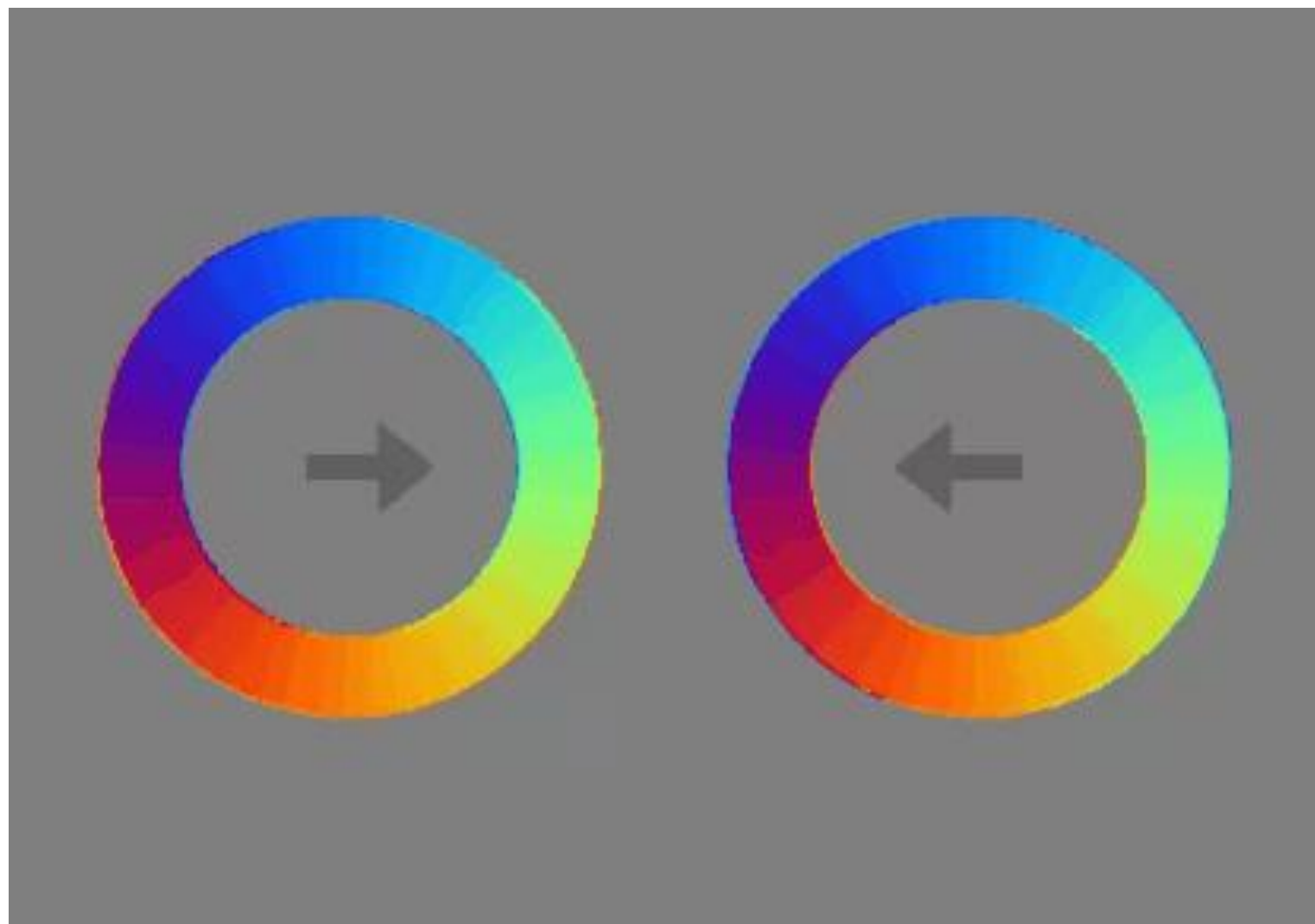
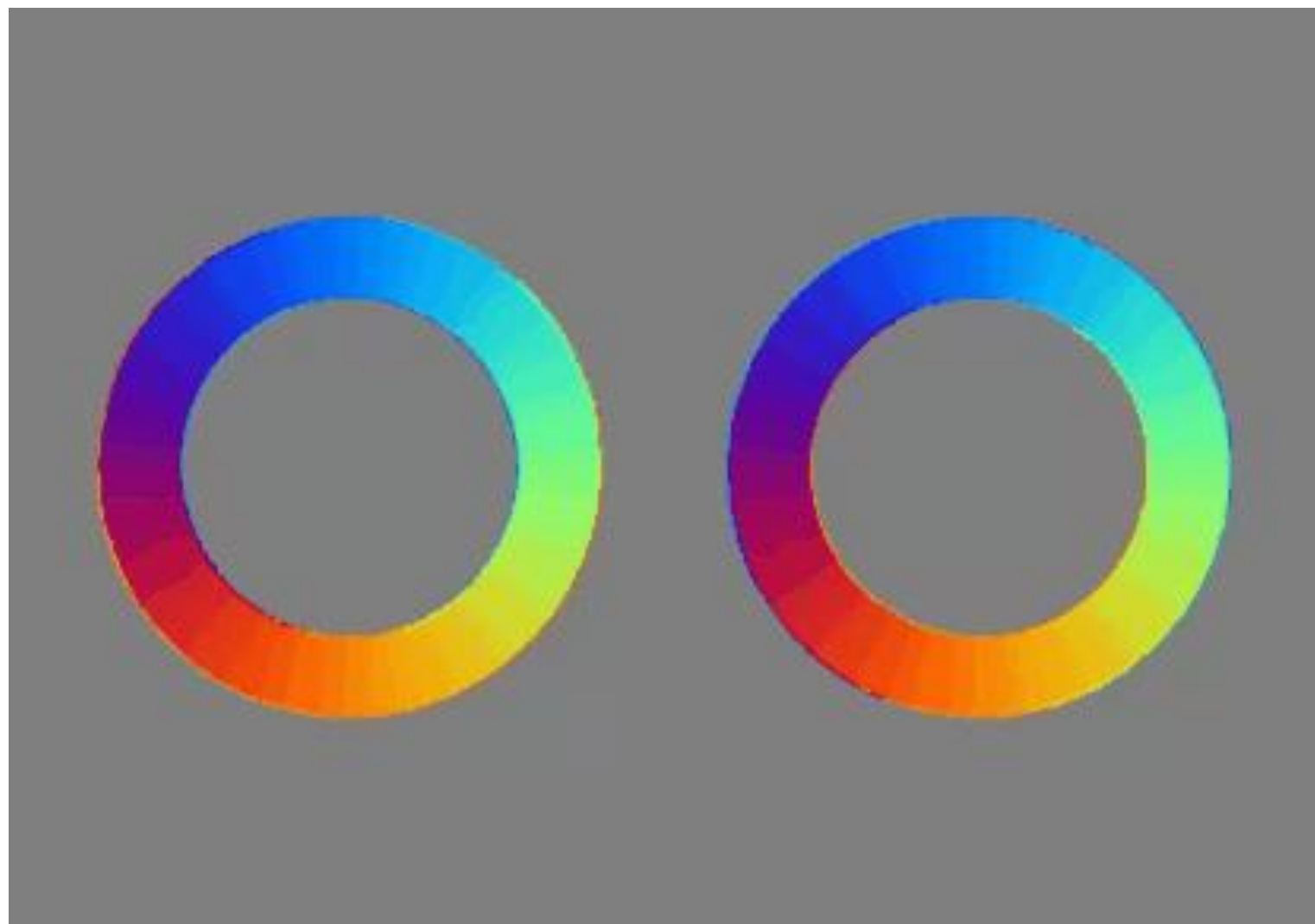










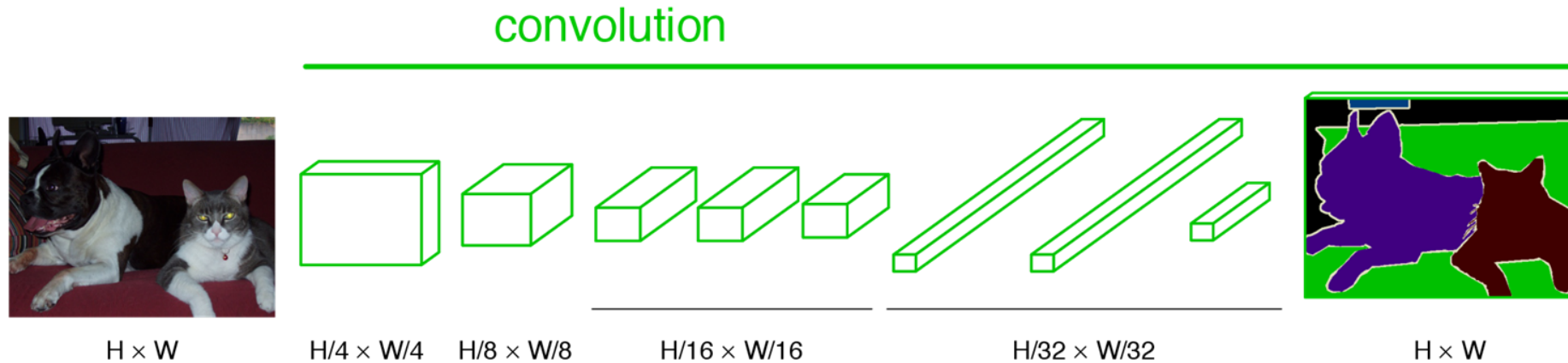
# Structured Predictions with Deep Learning

James Hays

# Outline – More complex outputs from deep networks

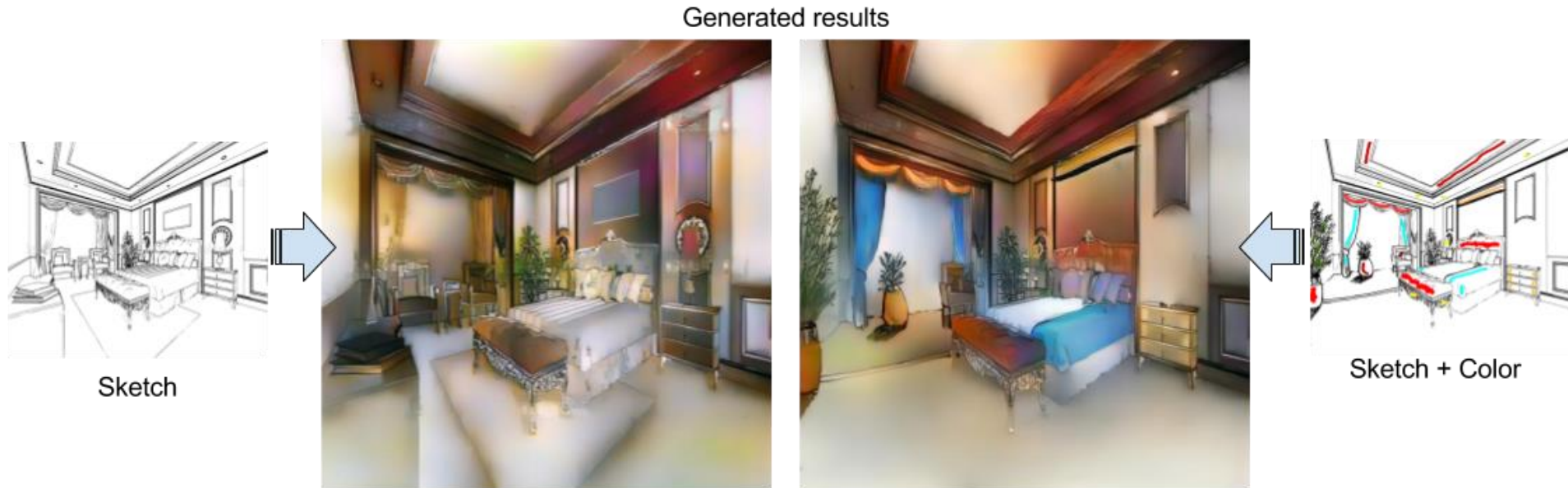
- Image Output (e.g. colorization, semantic segmentation, super-resolution, stylization, depth estimation...)
- Attributes
- Text Captions
- Semantic Keypoints
- Object Detection
  - Bounding boxes
  - Keypoint locations
  - Segmentation masks
  - 3D cuboids
  - 3D object coordinates

# end-to-end, pixels-to-pixels network



# What if we want other types of outputs?

- Easy\*: Predict any fixed dimensional output

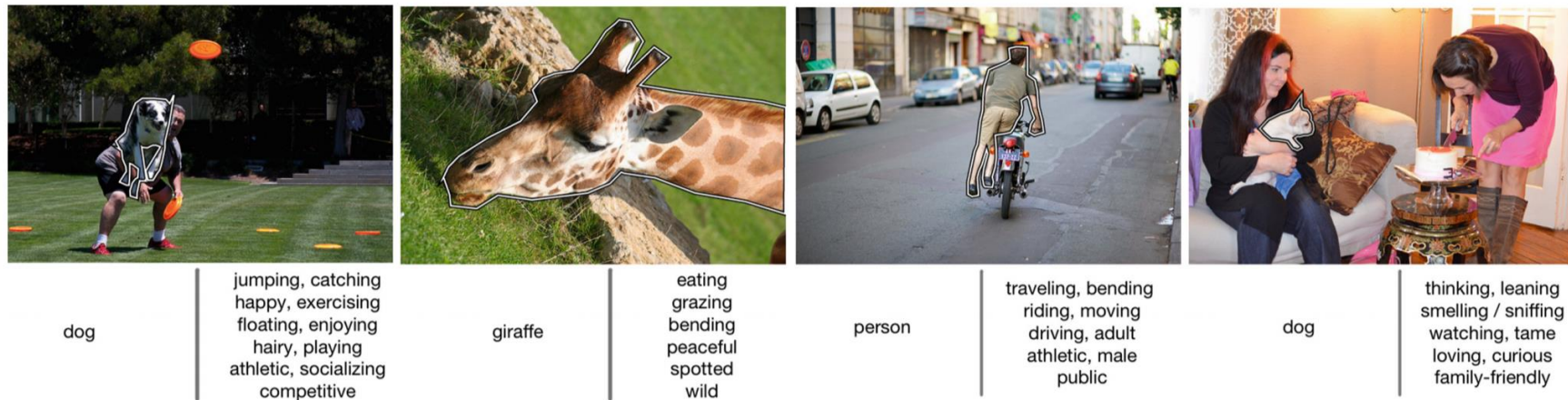


Scribbler: Controlling Deep Image Synthesis with Sketch and Color.  
Sangkloy, Lu, Chen Yu, and Hays. CVPR 2017

\*easy to design an architecture. Not necessarily easy to get working well.

# What if we want other types of outputs?

- Easy: Predict a fixed number of labels. For *classification*, there will be just one best answer, but for other labels like *attributes*, dozens could be appropriate for an image.



**Fig. 1.** *Examples from COCO Attributes.* In the figure above, images from the COCO dataset are shown with one object outlined in white. Under the image, the COCO object label is listed on the left, and the COCO Attribute labels are listed on the right.

# What if we want other types of outputs?

- Hard: Outputs with varying dimensionality or cardinality
  - A natural language image caption
  - An arbitrary number of bounding boxes (4 parameters each)
  - An arbitrary number of human skeletons (17 keypoints each)
  - An arbitrary number of instance segmentation masks (hundreds of parameters each)
- Today we will examine influential methods for keypoint prediction and object detection
  - The keypoint detection approach is “*bottom up*” and the object detection approach is “*top down*”.

# The Seminal R-CNN

## Rich feature hierarchies for accurate object detection and semantic segmentation

Tech report (v5)

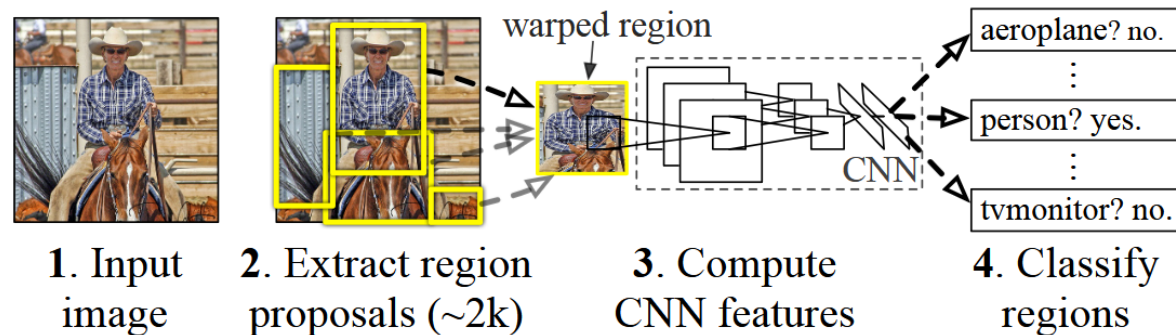
Ross Girshick   Jeff Donahue   Trevor Darrell   Jitendra Malik  
UC Berkeley

`{rbg, jdonahue, trevor, malik}@eecs.berkeley.edu`

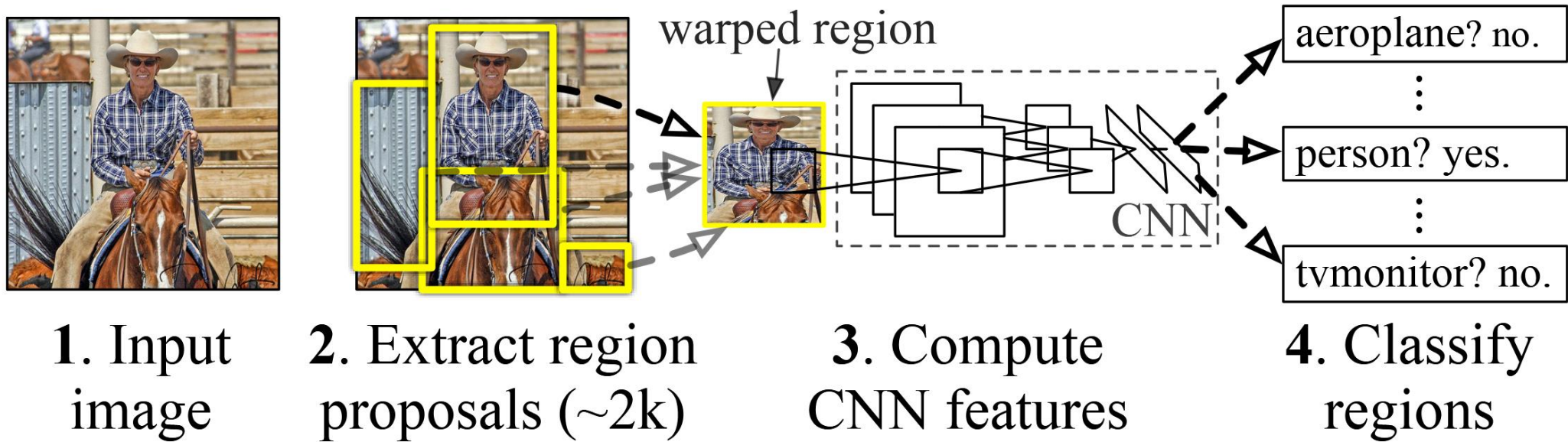
### Abstract

*Object detection performance, as measured on the canonical PASCAL VOC dataset, has plateaued in the last few years. The best-performing methods are complex ensemble systems that typically combine multiple low-level image features with high-level context. In this paper we*

### R-CNN: *Regions with CNN features*



## R-CNN: *Regions with CNN features*

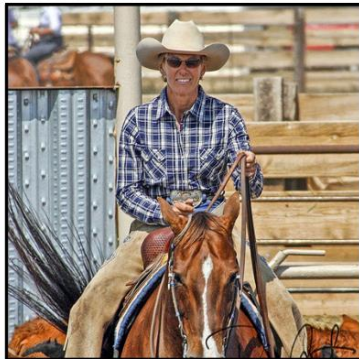


**Figure 1: Object detection system overview.** Our system (1) takes an input image, (2) extracts around 2000 bottom-up region proposals, (3) computes features for each proposal using a large convolutional neural network (CNN), and then (4) classifies each region using class-specific linear SVMs. R-CNN achieves a mean average precision (mAP) of **53.7%** on **PASCAL VOC 2010**. For

Stage 1

Stage 2

## R-CNN: *Regions with CNN features*

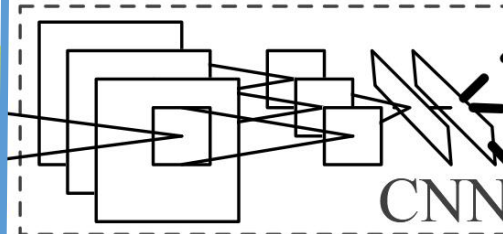


1. Input image



2. Extract region proposals (~2k)

warped region



CNN

3. Compute CNN features

aeroplane? no.

⋮

person? yes.

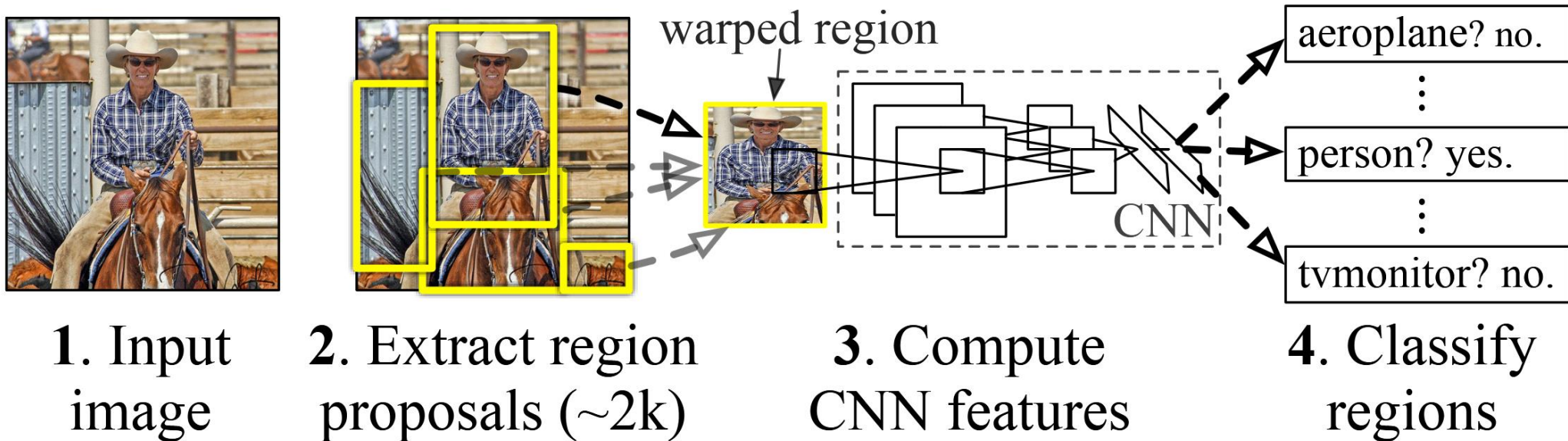
⋮

tvmonitor? no.

4. Classify regions

Learning happens only  
in the dashed box

## R-CNN: *Regions with CNN features*



Some critical stages are omitted from this figure

But also here in many architectures

Non-maximum suppression (NMS) needs to happen here

## R-CNN: Regions with CNN features

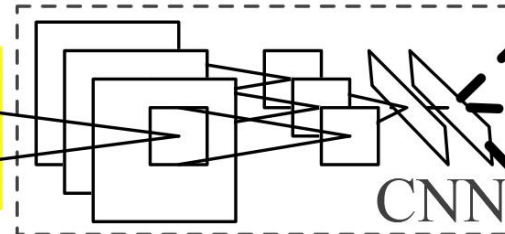


1. Input image



2. Extract region proposals (~2k)

warped region



CNN

3. Compute CNN features

aeroplane? no.

⋮

person? yes.

⋮

tvmonitor? no.

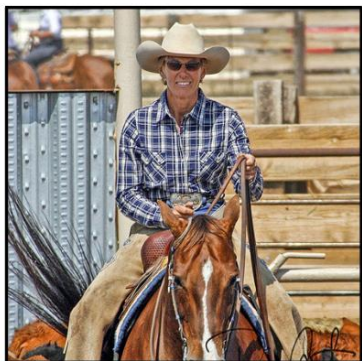
4. Classify regions

Stage 1

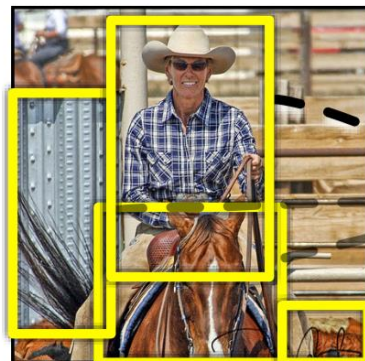
Stage 2

Stage 3

## R-CNN: *Regions with CNN features*

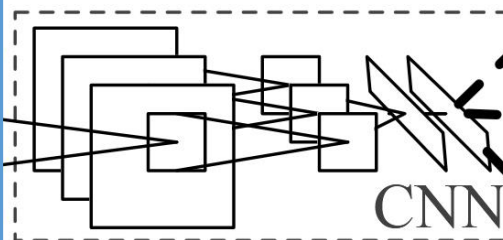


1. Input image

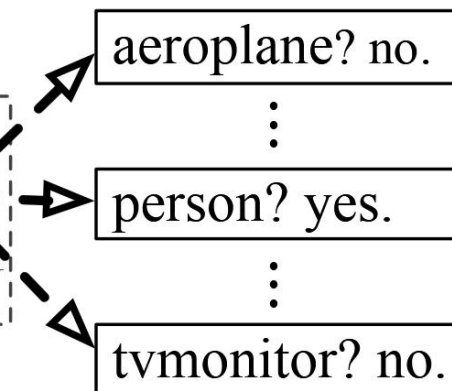


2. Extract region proposals (~2k)

warped region



3. Compute CNN features



4. Classify regions

NMS

# SSD: Single Shot MultiBox Detector

Wei Liu(1), **Dragomir Anguelov(2)**, Dumitru Erhan(3), Christian Szegedy(3),  
Scott Reed(4), Cheng-Yang Fu(1), Alexander C. Berg(1)

UNC Chapel Hill(1), **Zoox Inc.(2)**, Google Inc.(3),  
University of Michigan(4)

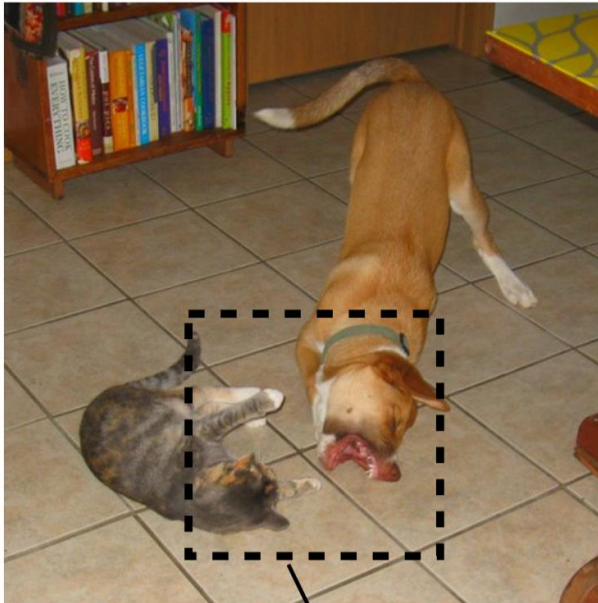


THE UNIVERSITY  
of NORTH CAROLINA  
at CHAPEL HILL

2015

# Bounding Box Prediction

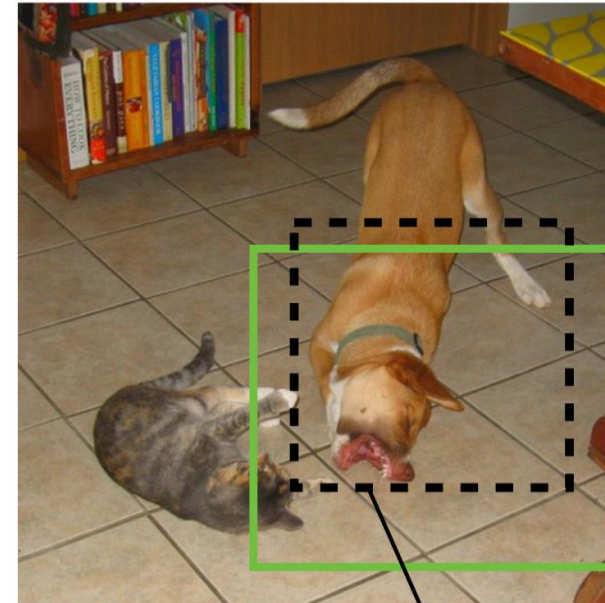
Classical sliding windows



Is it a cat? **No**

Discretize the box space **densely**

SSD and other deep approaches



dog: 0.4 cat: 0.2

Discretize the box space more **coarsely**  
**Refine** the coordinates of each box

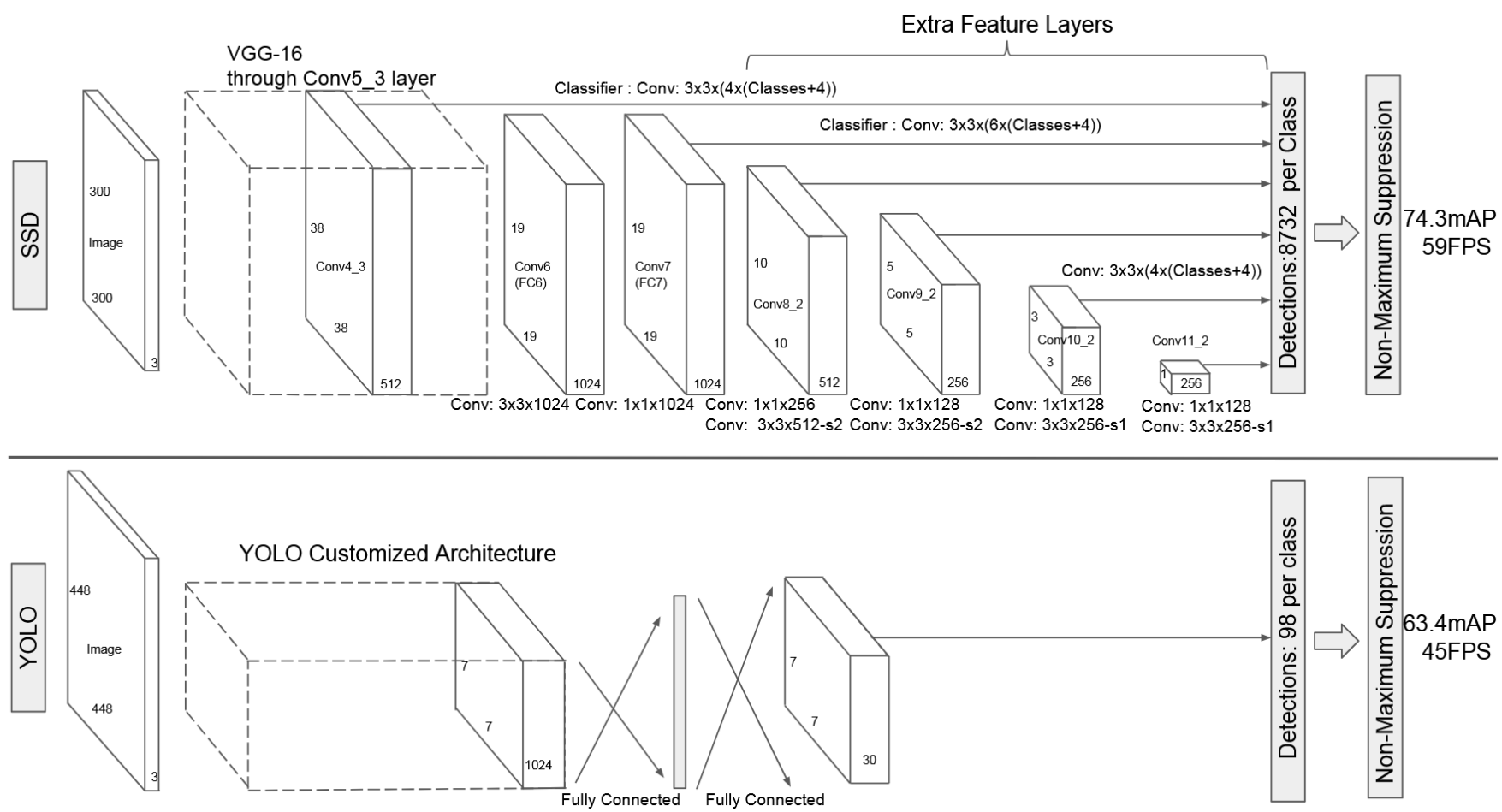
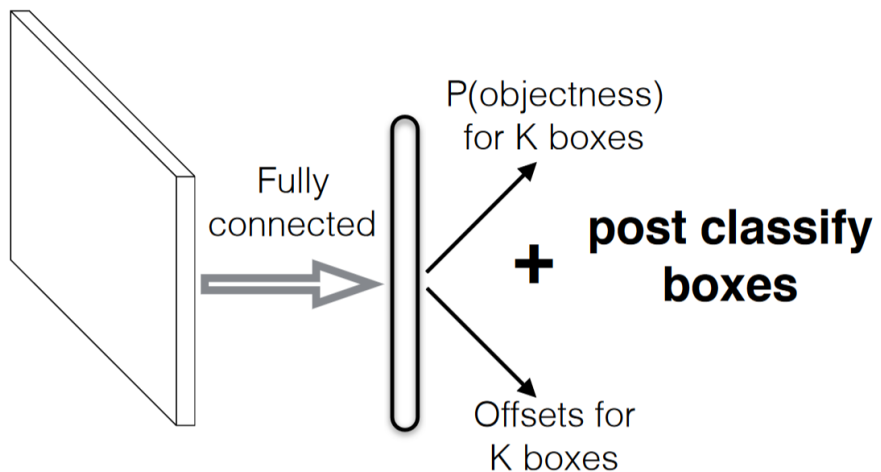


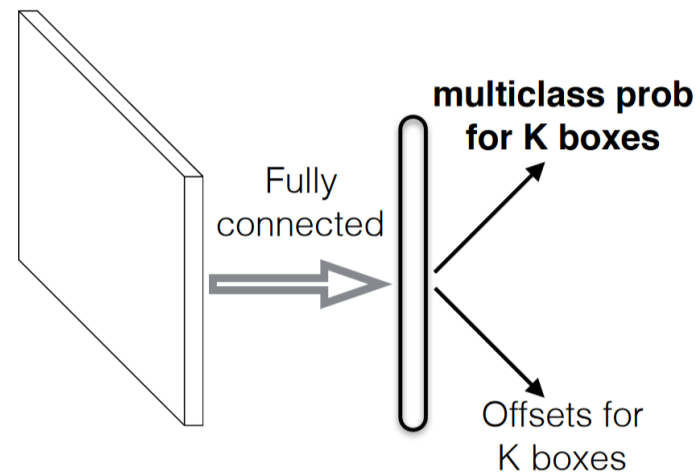
Fig. 2: A comparison between two single shot detection models: SSD and YOLO [5]. Our SSD model adds several feature layers to the end of a base network, which predict the offsets to default boxes of different scales and aspect ratios and their associated confidences. SSD with a  $300 \times 300$  input size significantly outperforms its  $448 \times 448$  YOLO counterpart in accuracy on VOC2007 test while also improving the speed.

# Related Work

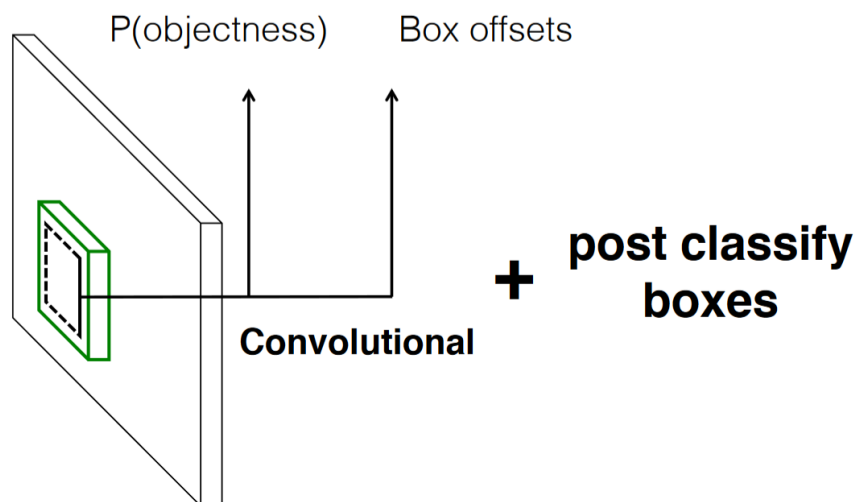
**MultiBox** [Erhan et al. CVPR14]



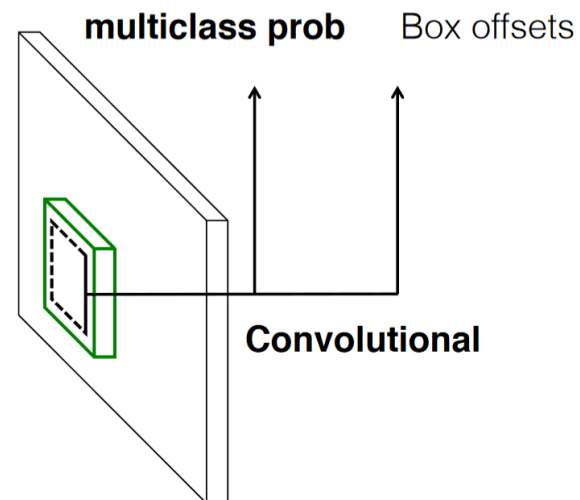
**YOLO** [Redmon et al. CVPR16]

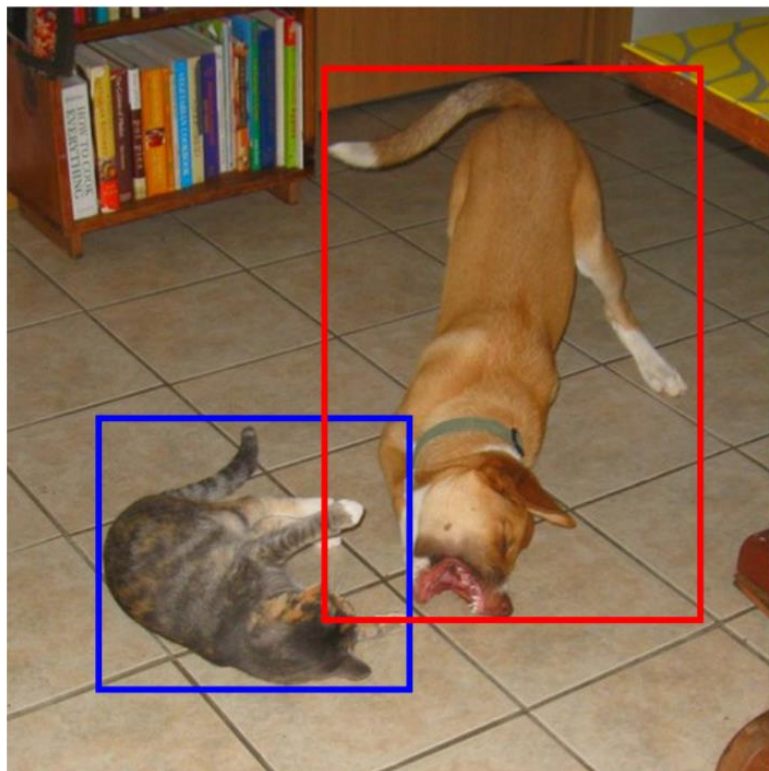


**Faster R-CNN** [Ren et al. NIPS15]

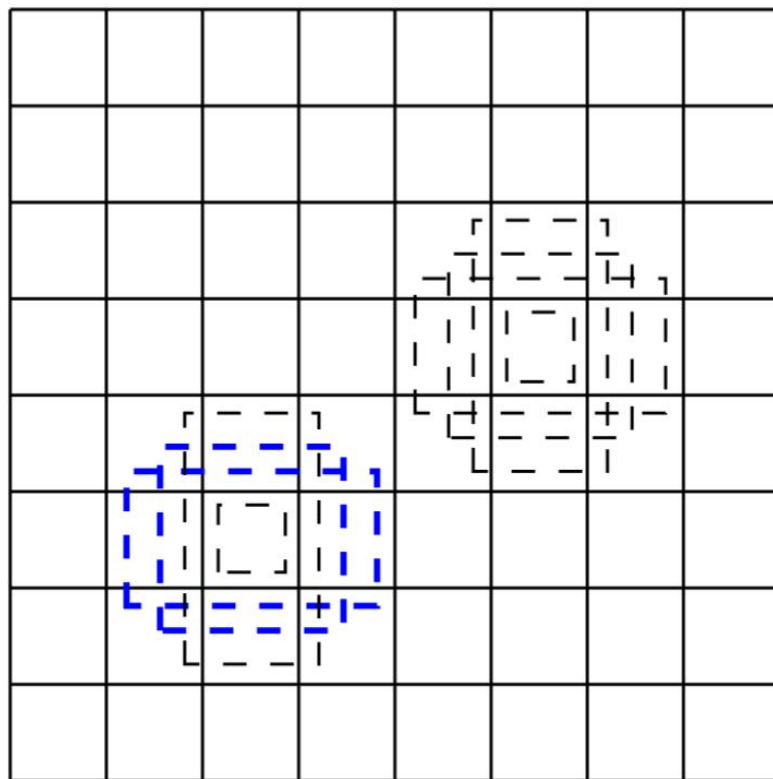


**SSD**

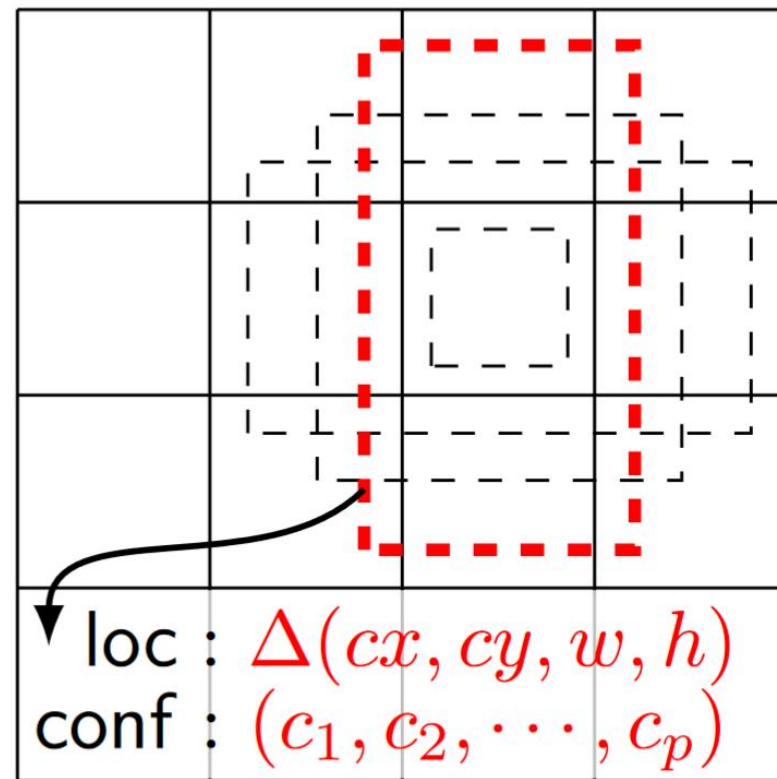




(a) Image with GT boxes



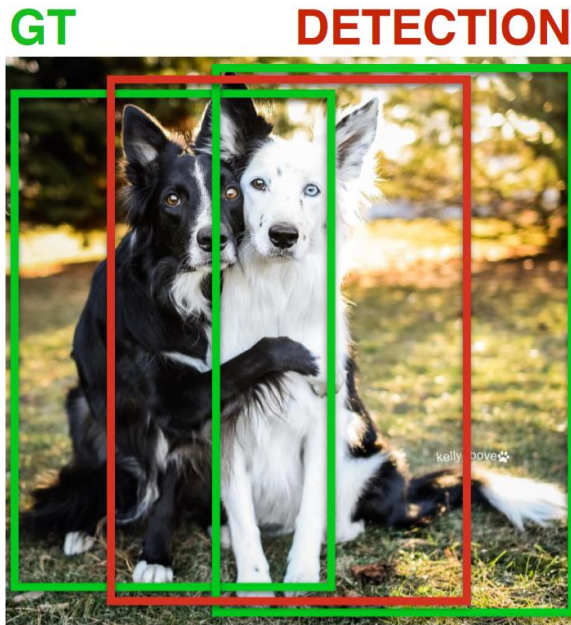
(b)  $8 \times 8$  feature map



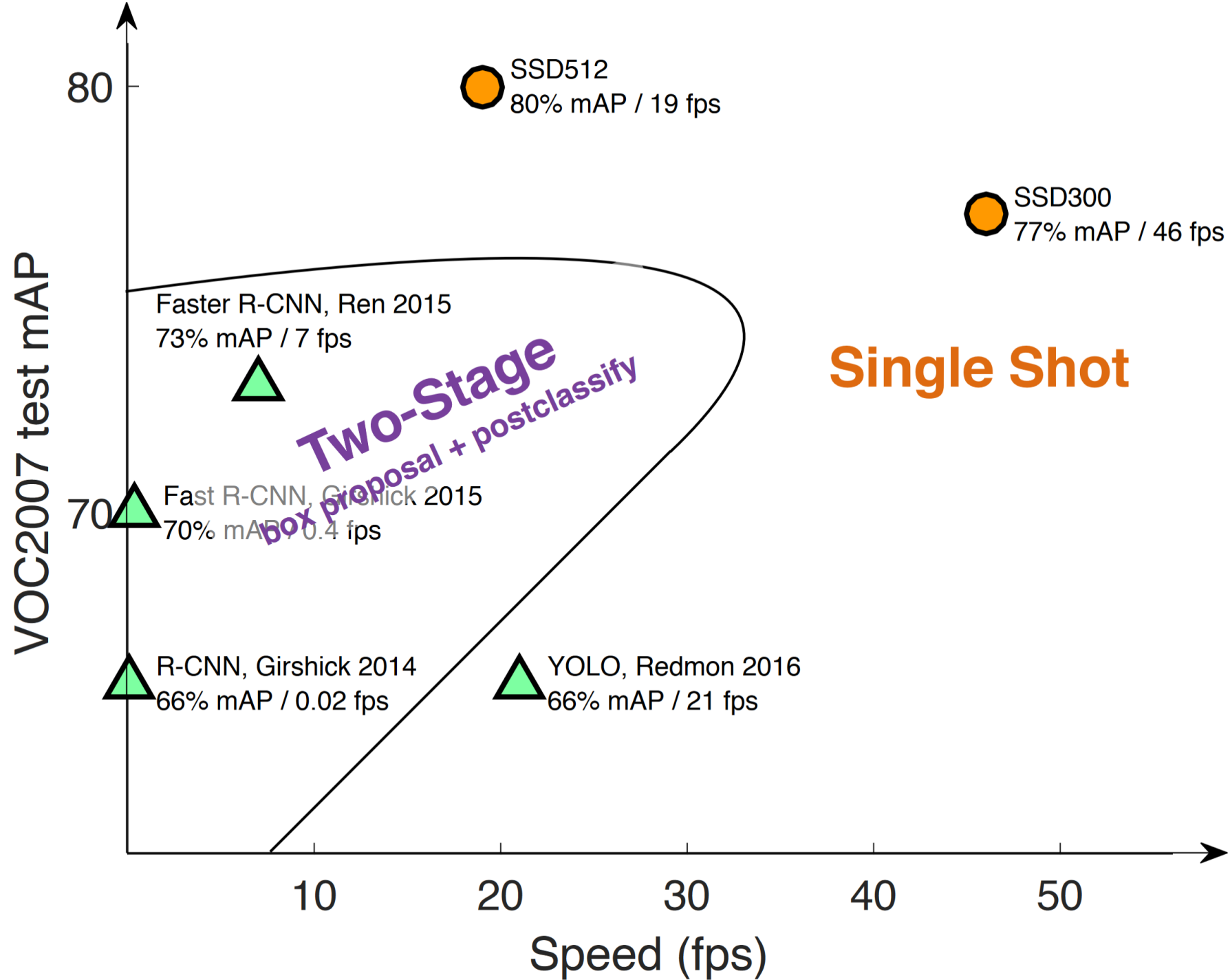
(c)  $4 \times 4$  feature map

# Why So Many Default Boxes?

|                 | Faster R-CNN | YOLO    | SSD300  | SSD512  |
|-----------------|--------------|---------|---------|---------|
| # Default Boxes | 6000         | 98      | 8732    | 24564   |
| Resolution      | 1000x600     | 448x448 | 300x300 | 512x512 |



- SmoothL1 or L2 loss for box shape averages among likely hypotheses
- Need to have enough default boxes (discrete bins) to do accurate regression in each
- General principle for regressing complex continuous outputs with deep nets



# Realtime Multi-Person Pose Estimation using Part Affinity Fields

Zhe Cao, Tomas Simon, Shih-En Wei, Yaser Sheikh  
Carnegie Mellon University

CVPR 2017



# Human Pose Estimation



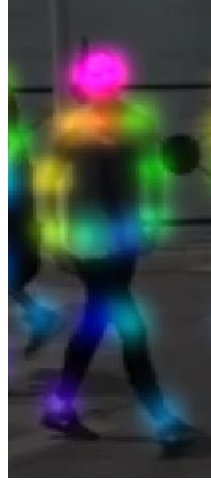
# Human Pose Estimation



# Single-Person Pose Estimation



# Single-Person Pose Estimation



# Multi-Person Pose Estimation



Color encodes the body part type

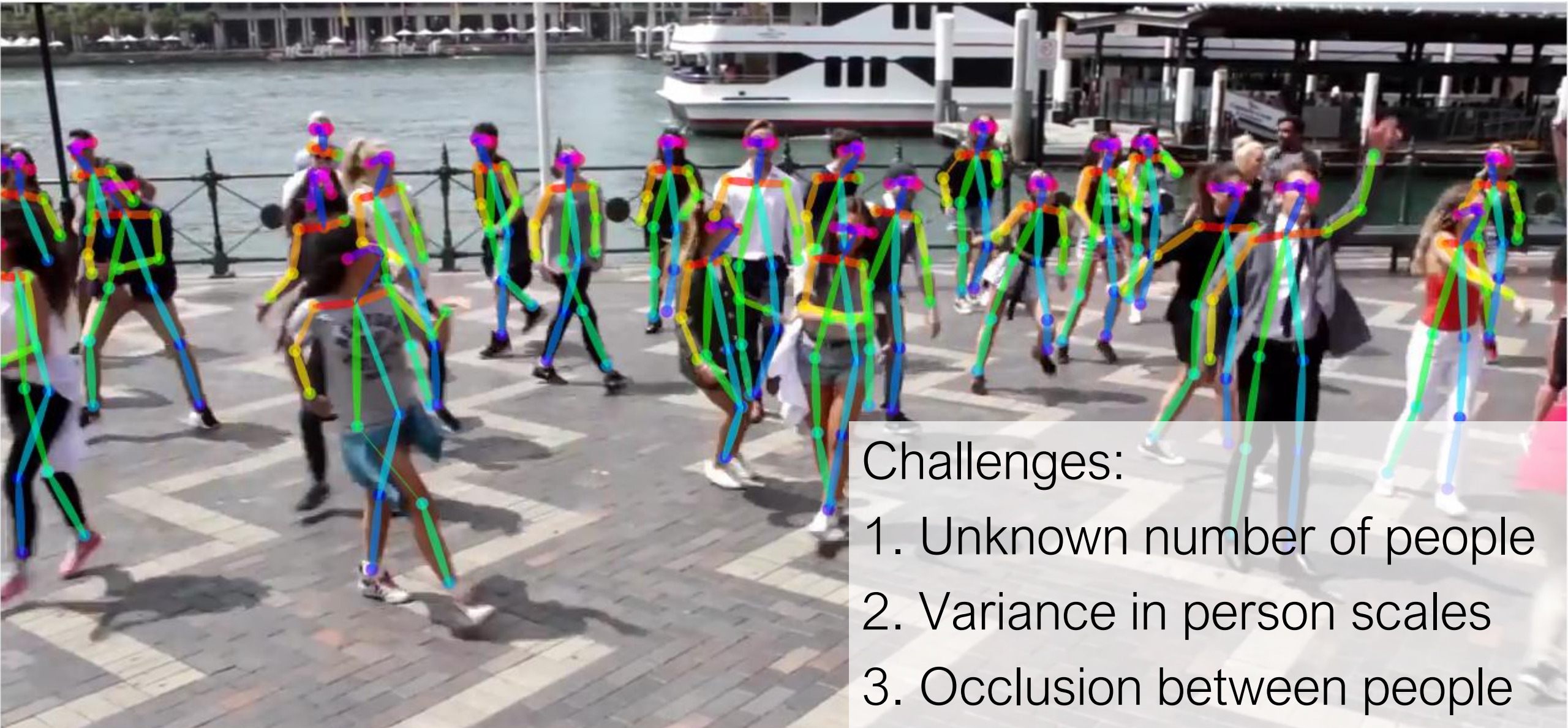
# Multi-Person Pose Estimation



# Major Challenge: Part-to-Person Association



# Major Challenge: Part-to-Person Association



Challenges:

1. Unknown number of people
2. Variance in person scales
3. Occlusion between people

# Major Challenge: Part-to-Person Association



For 30 people and each with 17 joints, there are in total **1.3 x 10<sup>5</sup>** pair-wise connection cost, NP-hard optimization

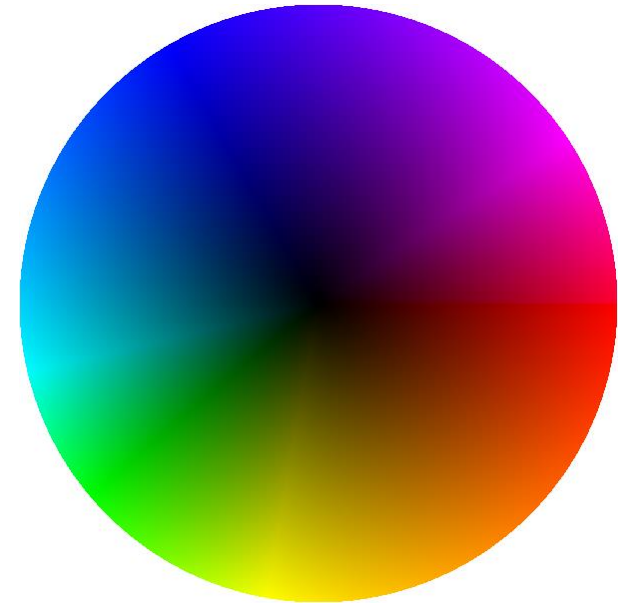
# Unexpected Conclusion



Bottom-up

An **efficient** representation is **discriminative** enough that a greedy parse is sufficient to produce high-quality results

# Novelty: Part Affinity Fields for Parts Association



Part Affinity Field between right elbow and wrist

# Novelty: Part Affinity Fields for Parts Association

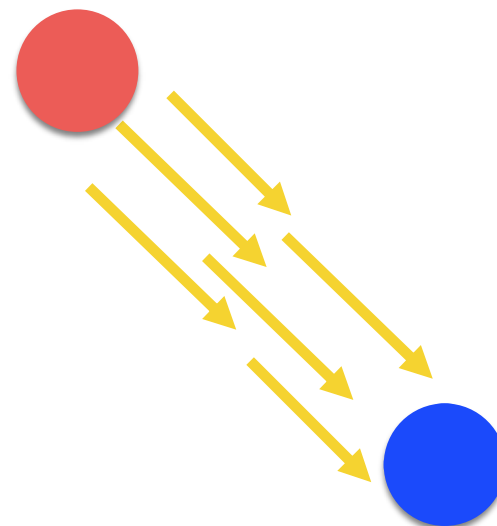
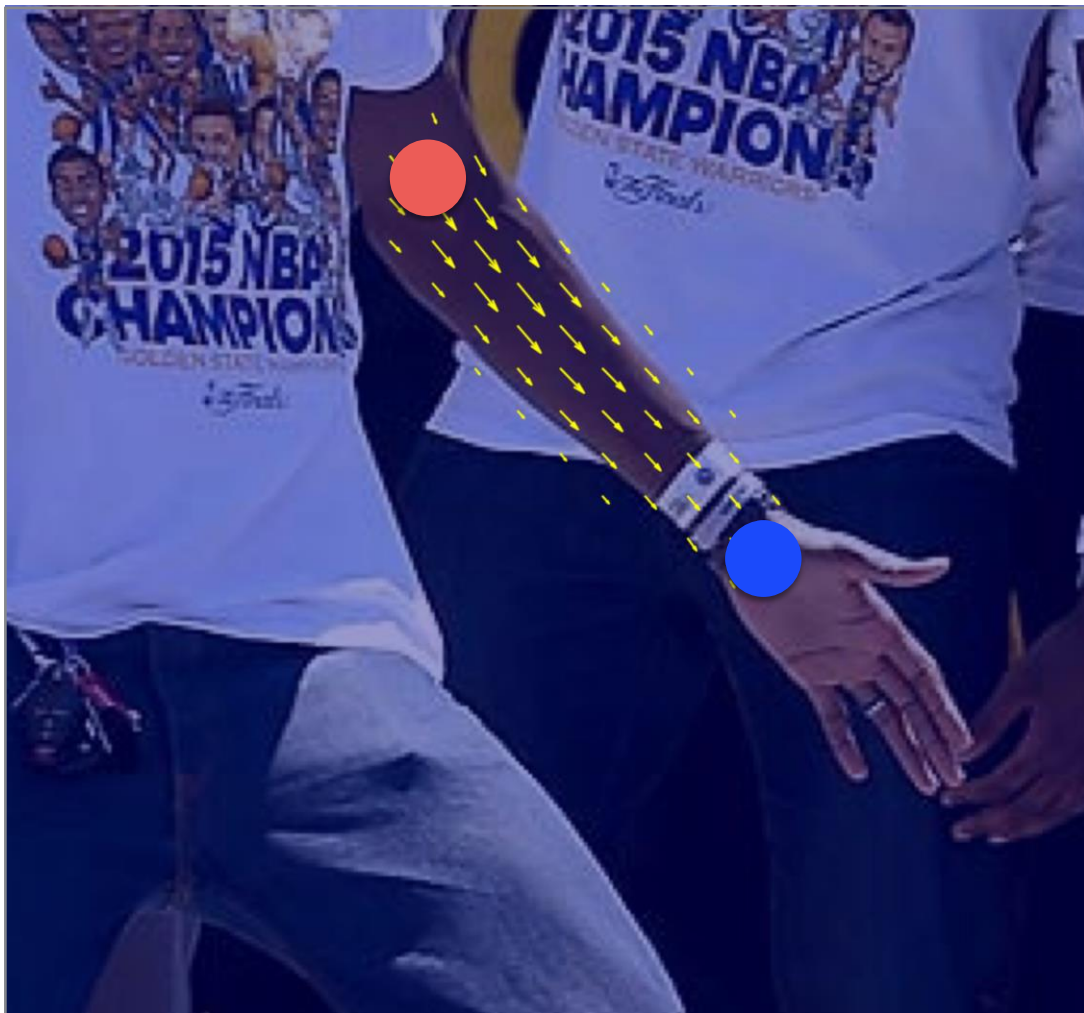


Part Affinity Field between right elbow and wrist

# Novelty: Part Affinity Fields for Parts Association

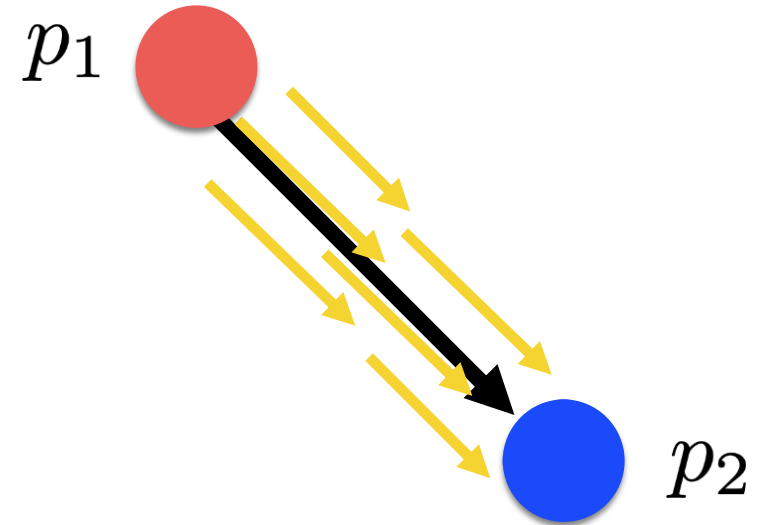


# Part Affinity Fields for Part-to-Part Association



- ➡ Direction vector in the PAFs
- Part 1
- Part 2

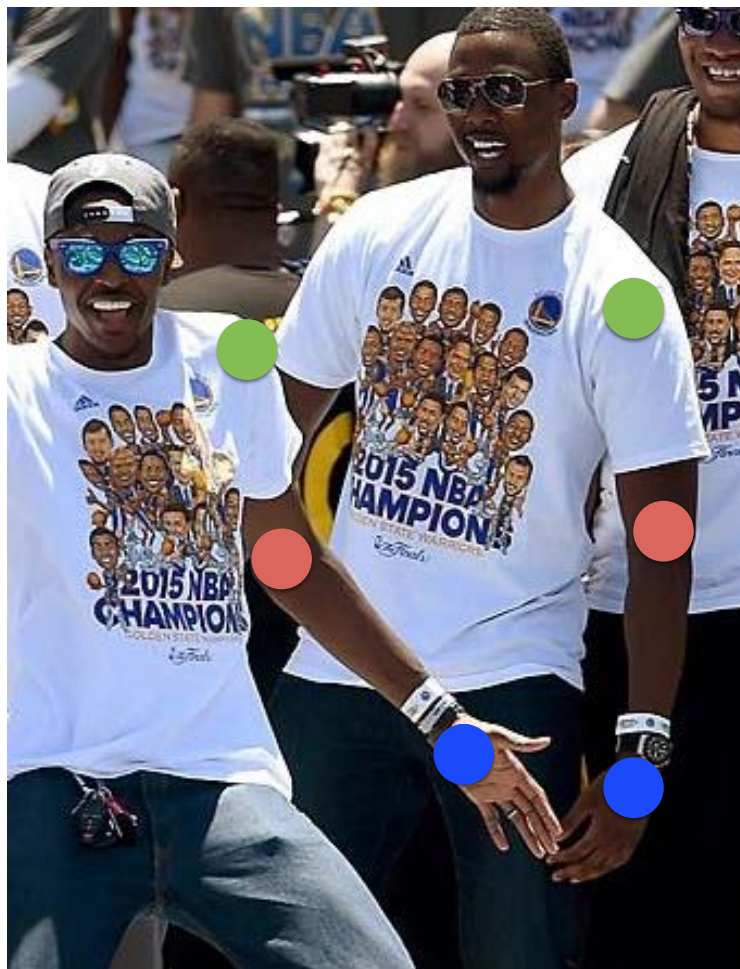
# Part Affinity Fields for Part-to-Part Association



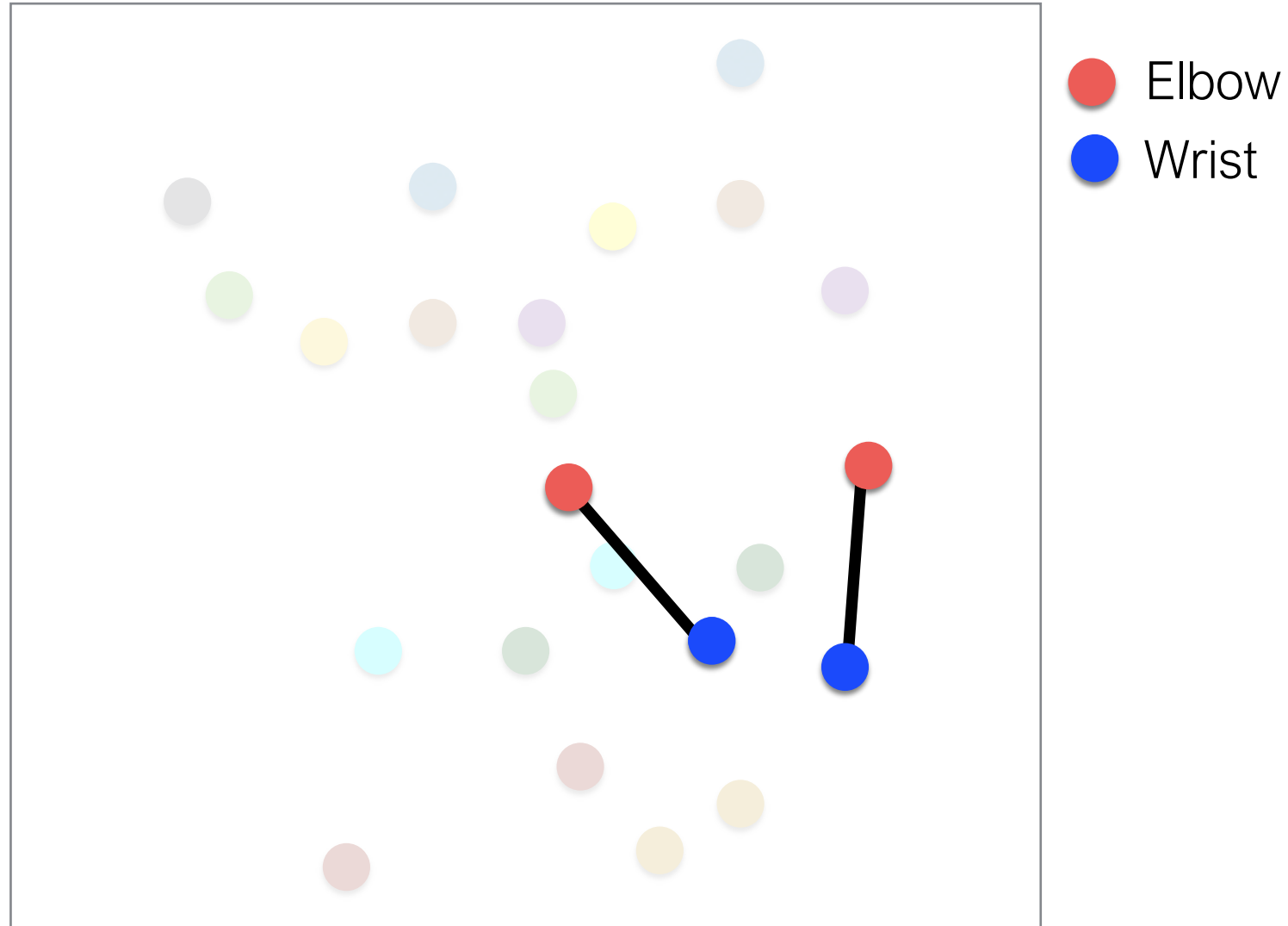
Affinity score between  $p_1$  and  $p_2$   
=  $\text{sum}(\vec{\mathbf{v}} \cdot p_1 \vec{p}_2)$

# Part Association for Full-body Pose

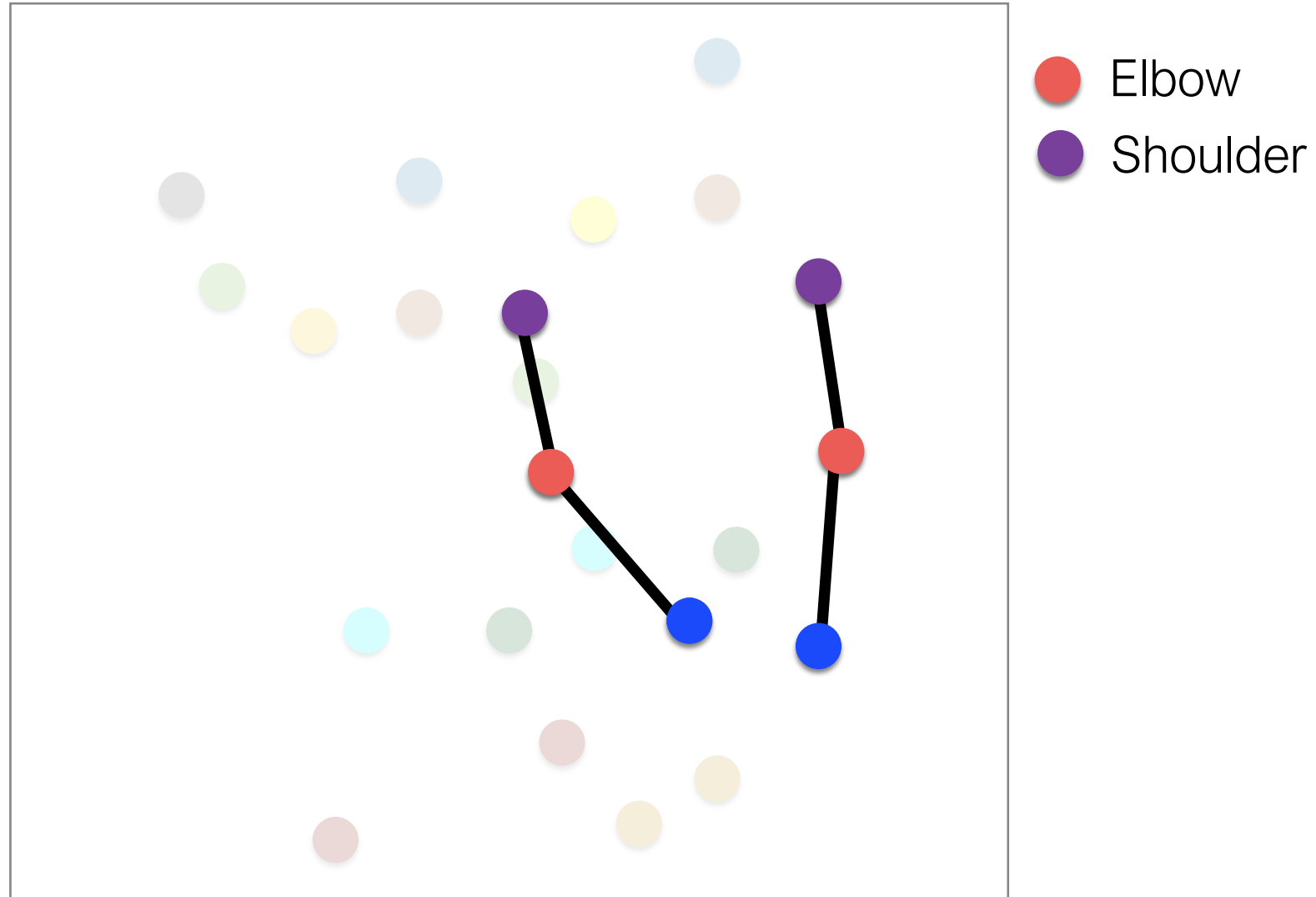
- Elbow
- Wrist
- Shoulder



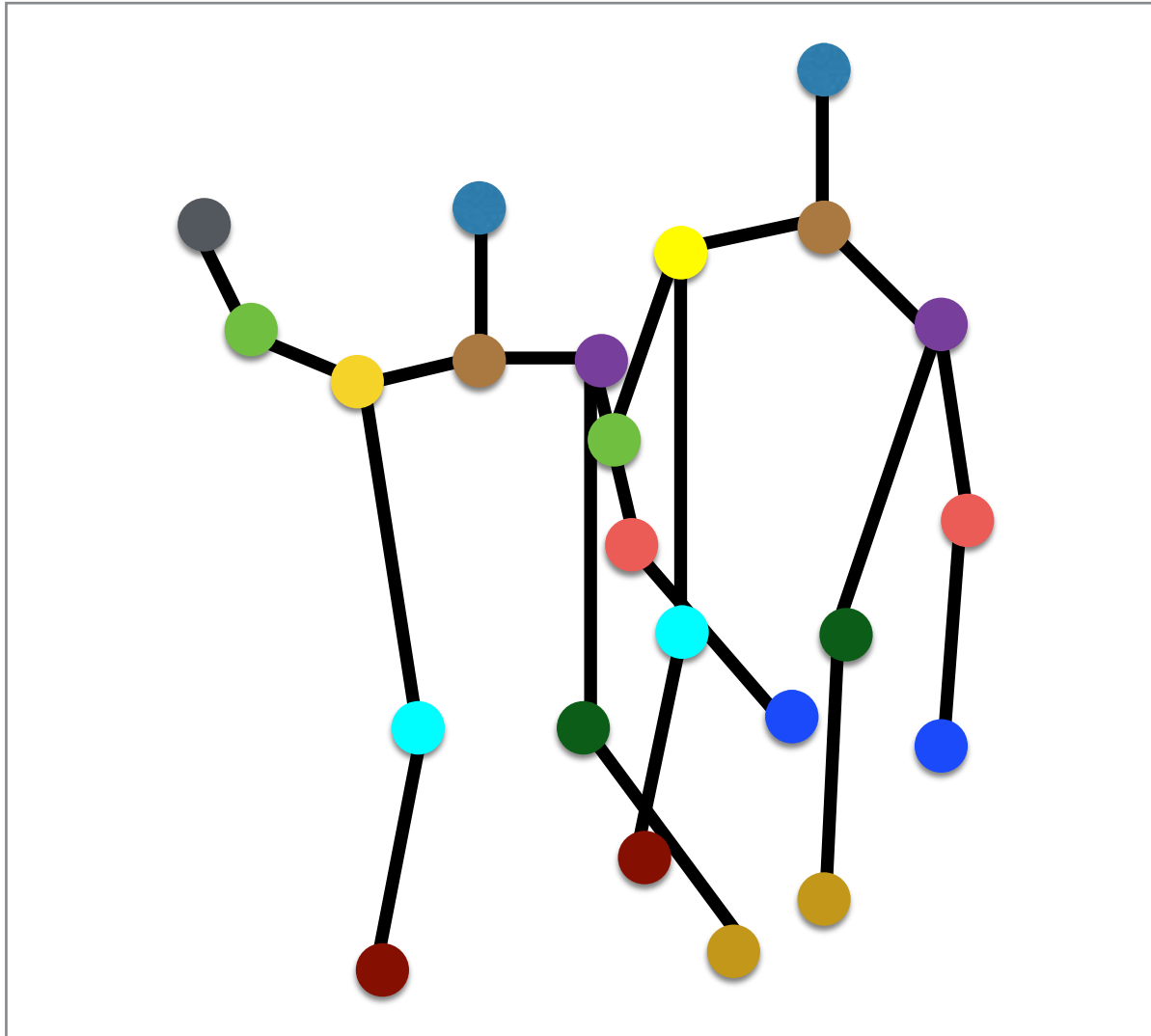
# Greedy Algorithm for Body Parts Association



# Greedy Algorithm for Body Parts Association



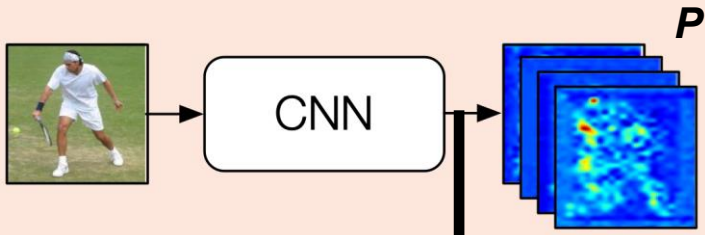
# Greedy Algorithm for Body Parts Association





# Jointly Learning Parts Detection and Parts Association

Stage 1

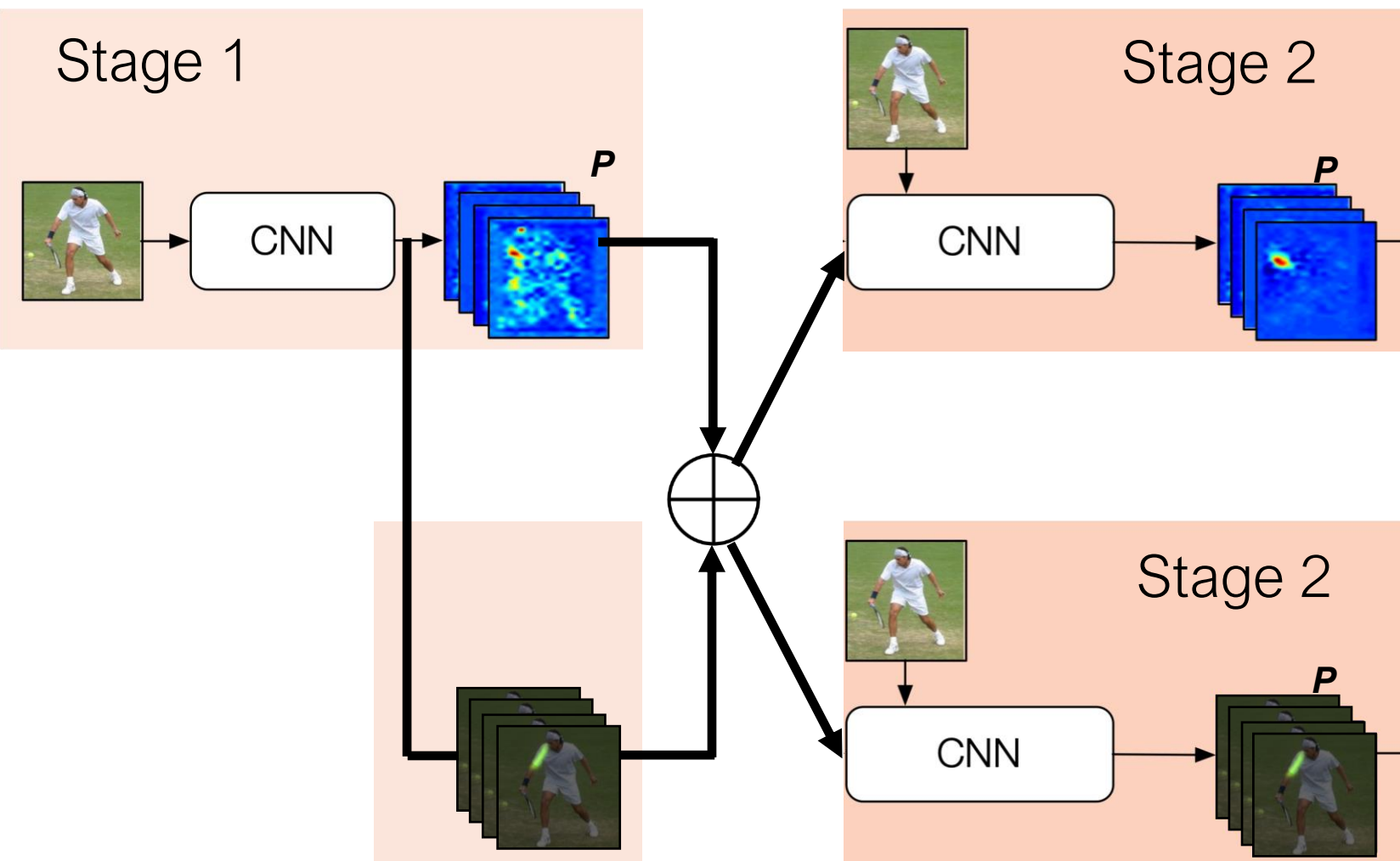


**1st** branch  
part heatmaps

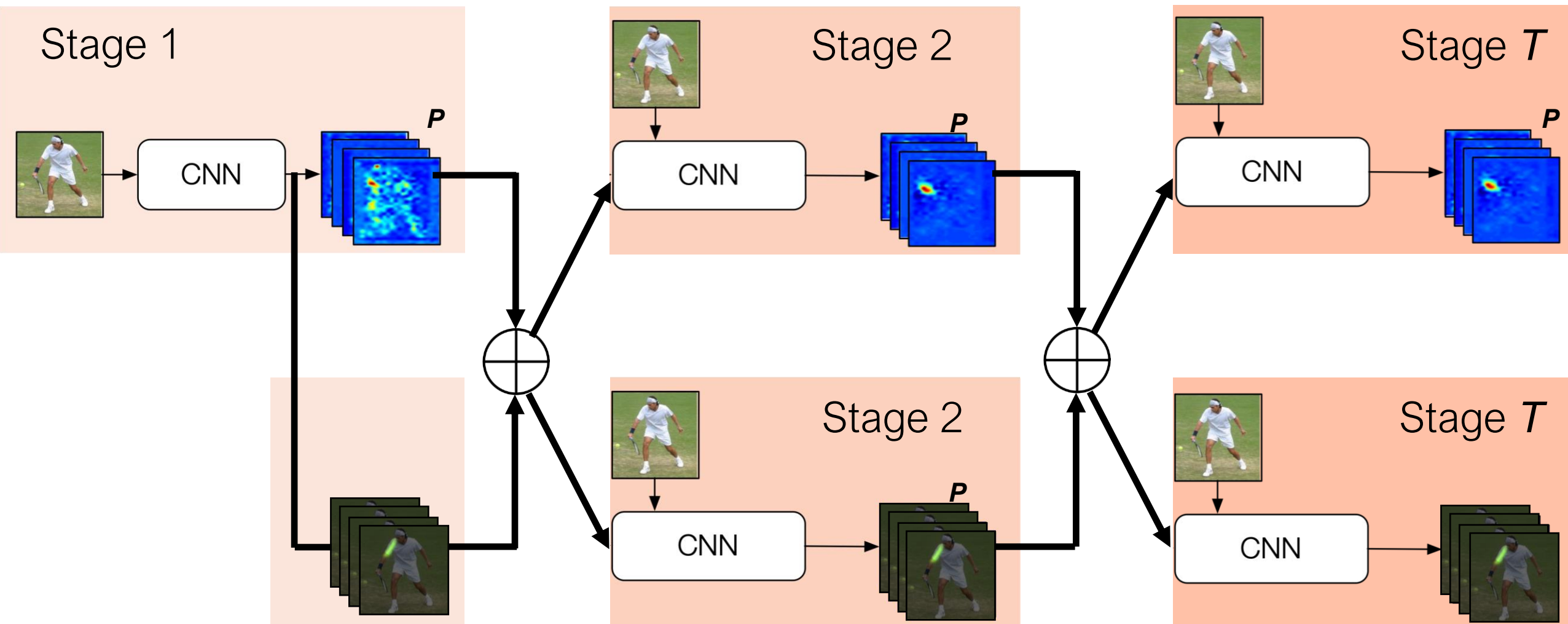


**2nd** branch  
part affinity fields

# Jointly Learning Parts Detection and Parts Association



# Jointly Learning Parts Detection and Parts Association





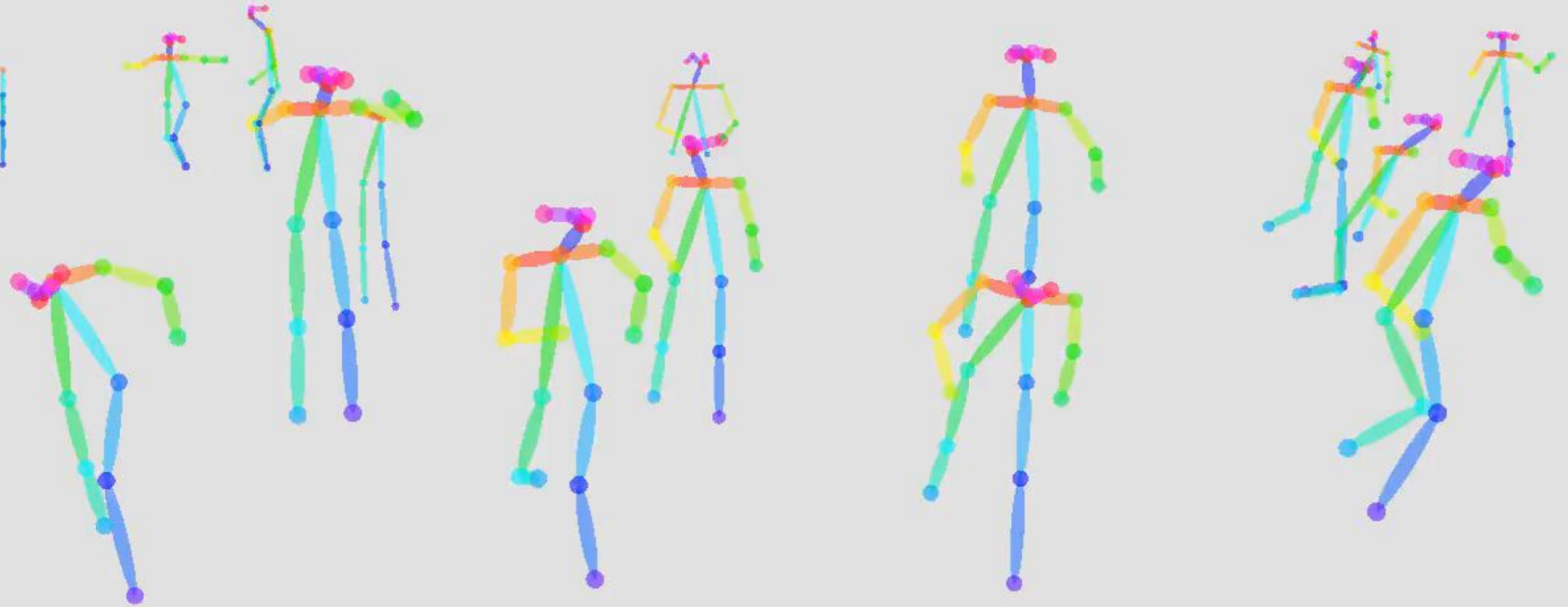
Bkg



PAFs

10.4 fps

## Frame by frame detection (no tracking)





# Mask R-CNN

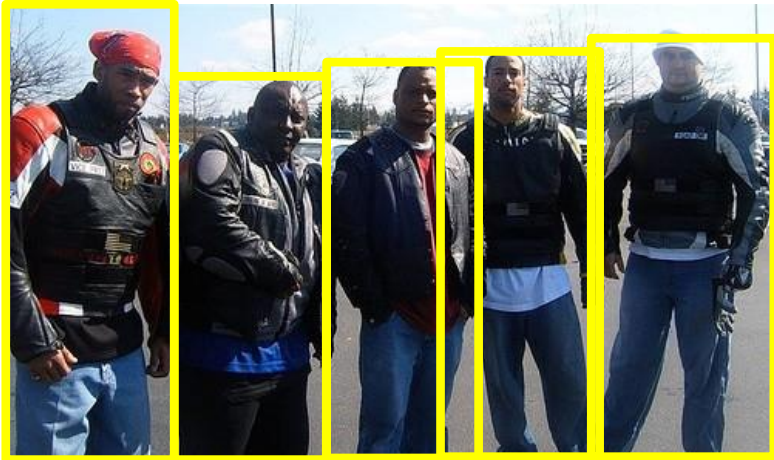
ICCV 2017

Kaiming He,

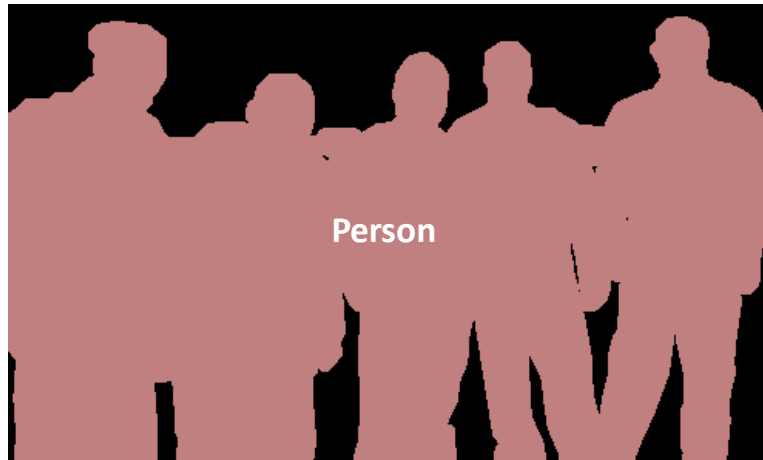
Georgia Gkioxari, Piotr Dollár, and Ross Girshick

Facebook AI Research (FAIR)

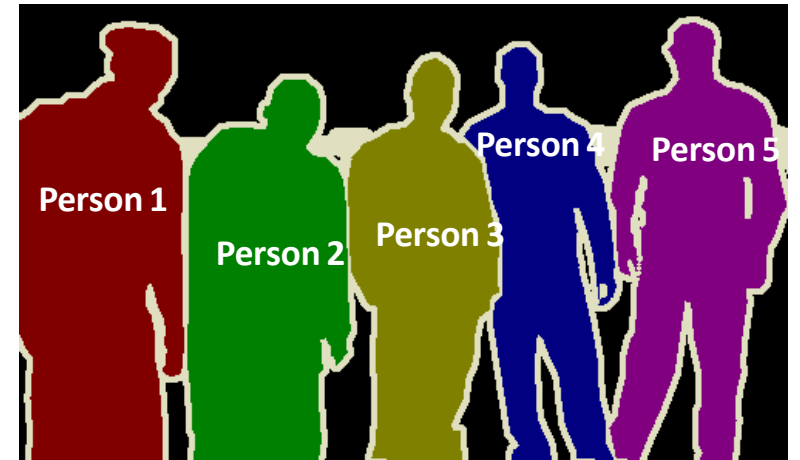
# Visual Perception Problems



Object Detection



Semantic Segmentation



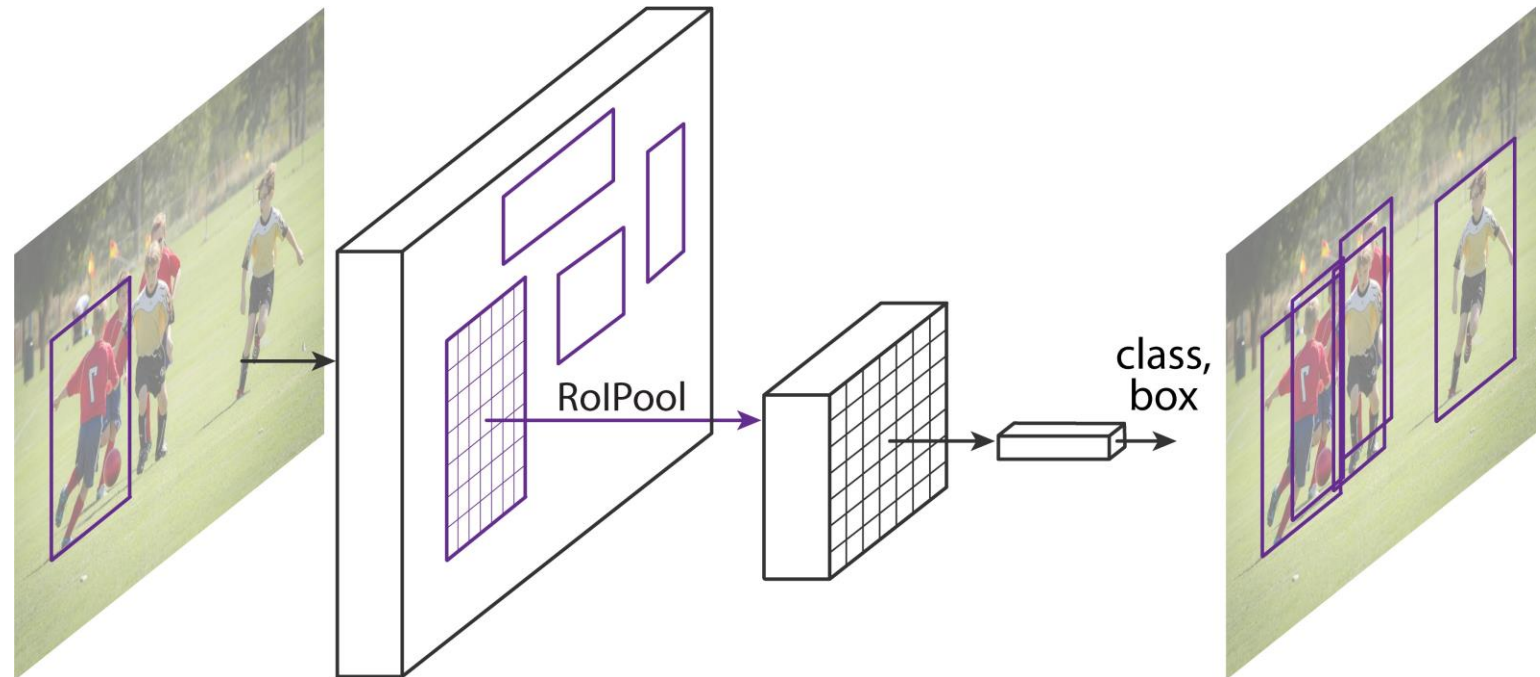
Instance Segmentation



# Object Detection

- Fast/Faster R-CNN

- ✓ Good speed
- ✓ Good accuracy
- ✓ Intuitive
- ✓ Easy to use



# Semantic Segmentation

- Fully Convolutional Net (FCN)
  - ✓ Good speed
  - ✓ Good accuracy
  - ✓ Intuitive
  - ✓ Easy to use

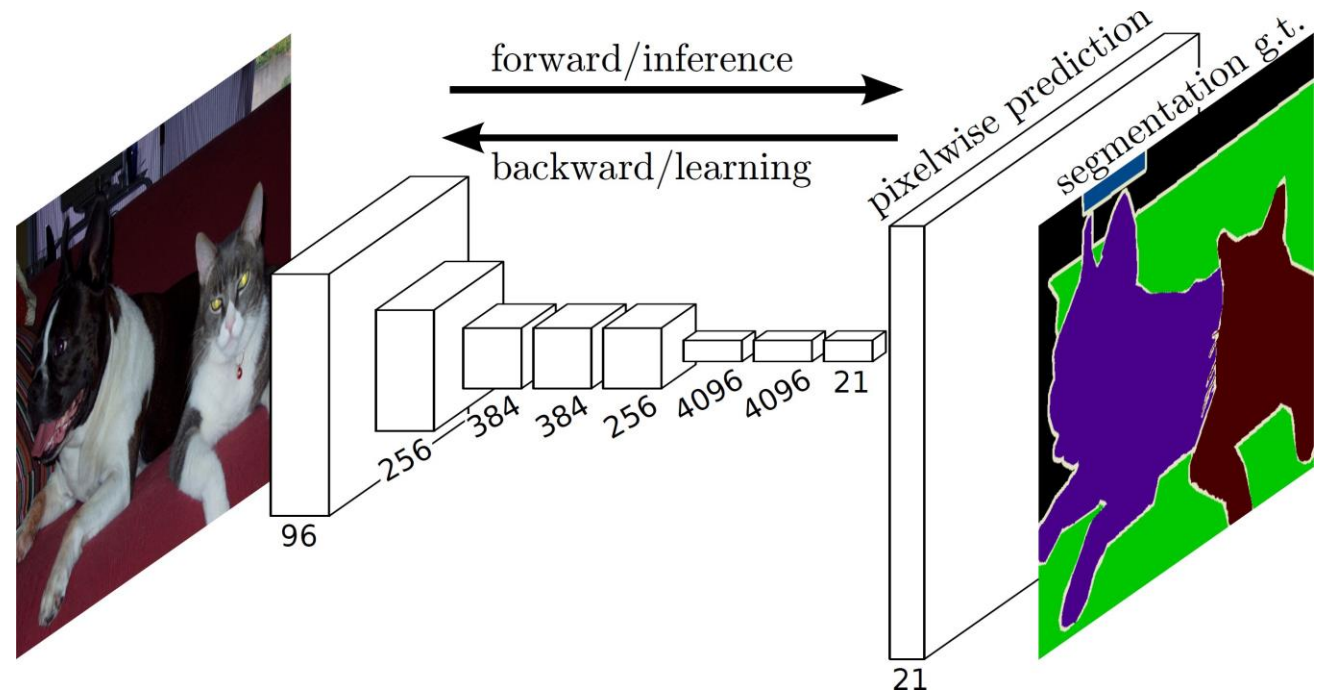
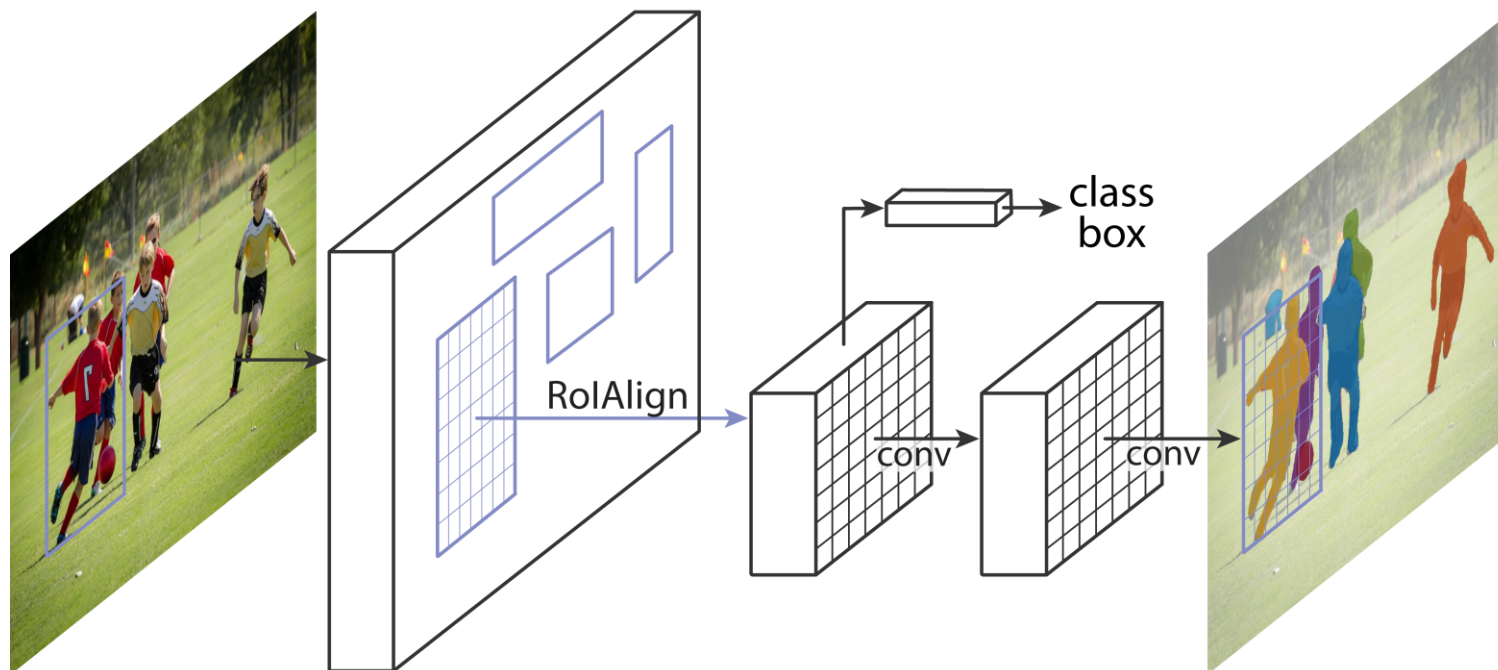


Figure credit: Long et al

# Instance Segmentation

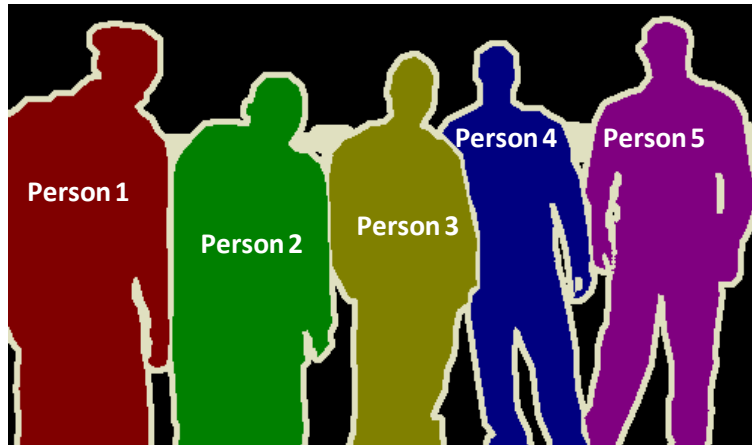
- **Goals** of Mask R-CNN

- ✓ Good speed
- ✓ Good accuracy
- ✓ Intuitive
- ✓ Easy to use

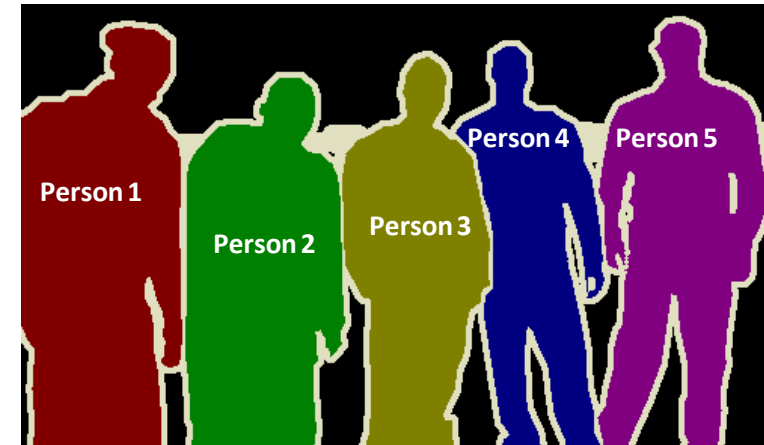
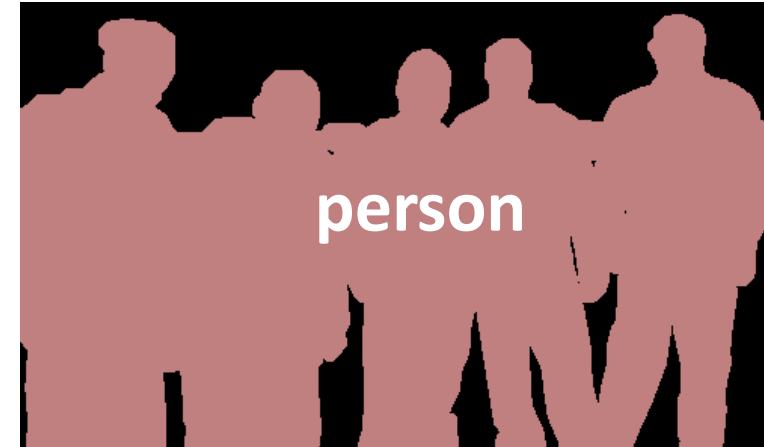


# Instance Segmentation Methods

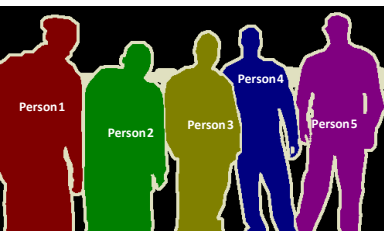
R-CNN driven



FCN driven

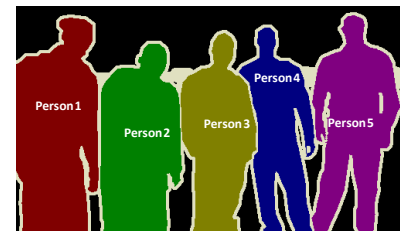
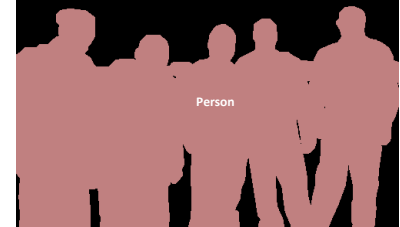


# Instance Segmentation Methods



**RCNN-driven**

- SDS [Hariharan et al, ECCV'14]
- HyperCol [Hariharan et al, CVPR'15]
- CFM [Dai et al, CVPR'15]
- MNC [Dai et al, CVPR'16]

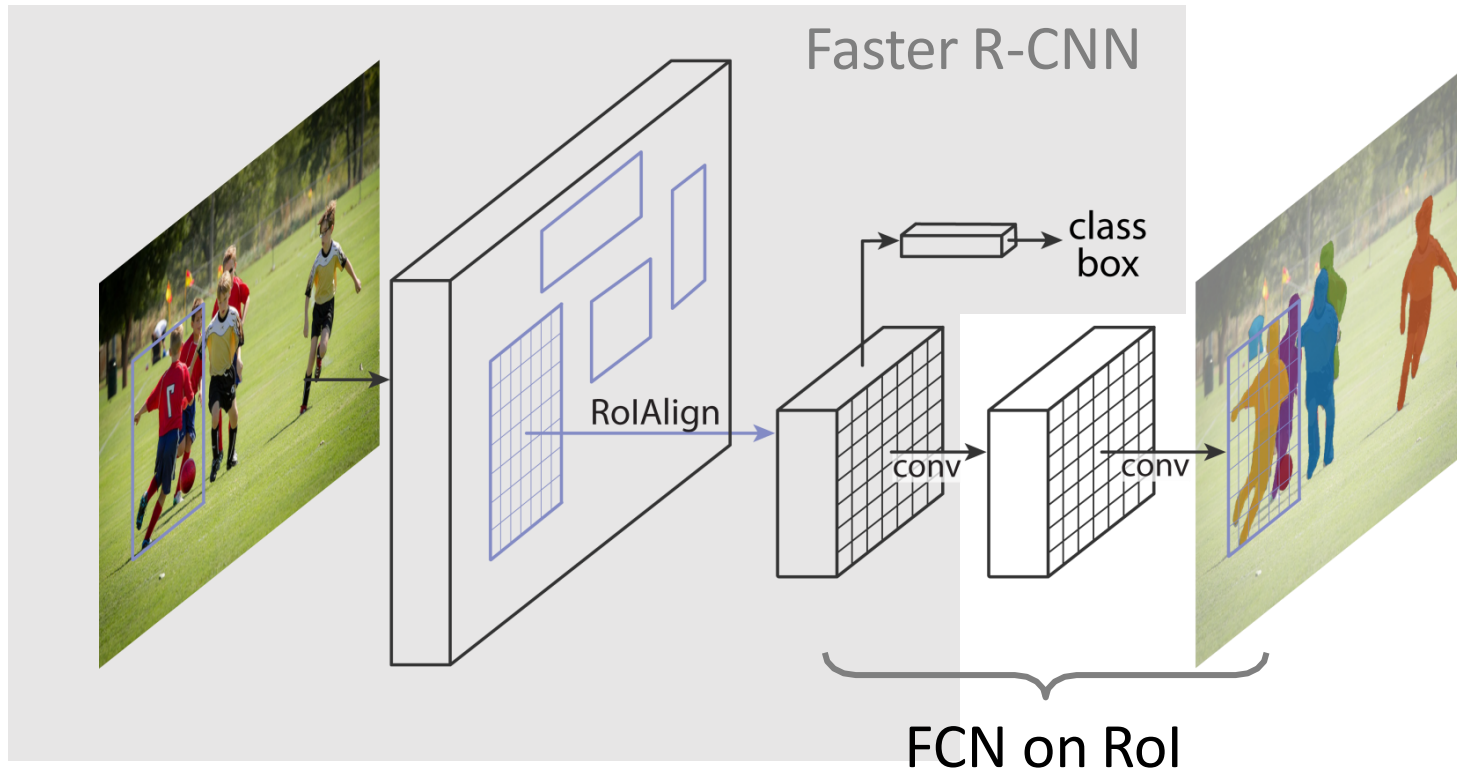


**FCN-driven**

- PFN [Liang et al, arXiv'15]
  - InstanceCut [Kirillov et al, CVPR'17]
  - Watershed [Bai & Urtasun, CVPR'17]
- 
- FCIS [Li et al, CVPR'17]
  - DIN [Arnab & Torr, CVPR'17]

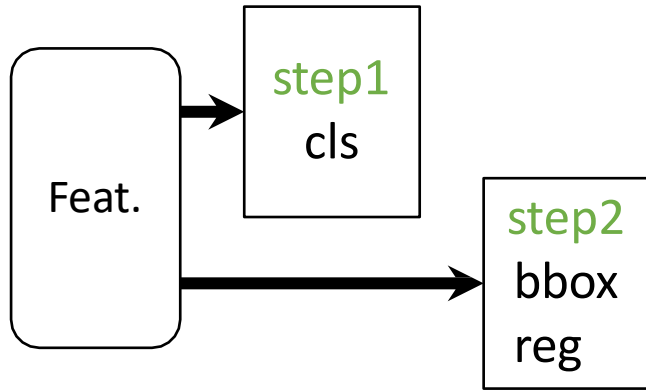
# Mask R-CNN

- Mask R-CNN = **Faster R-CNN** with **FCN** on Rols

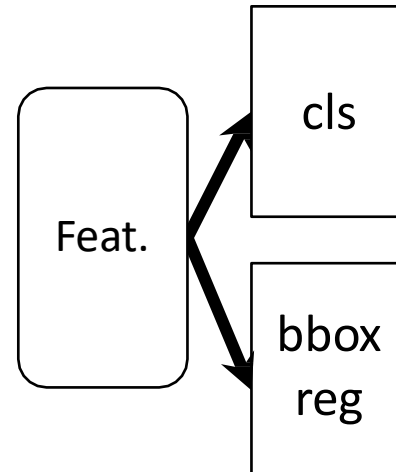


# Parallel Heads

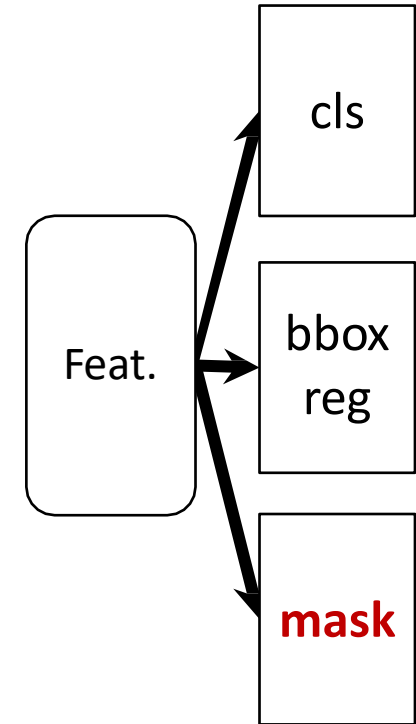
- Easy, fast to implement and train



(slow) R-CNN



Fast/er R-CNN

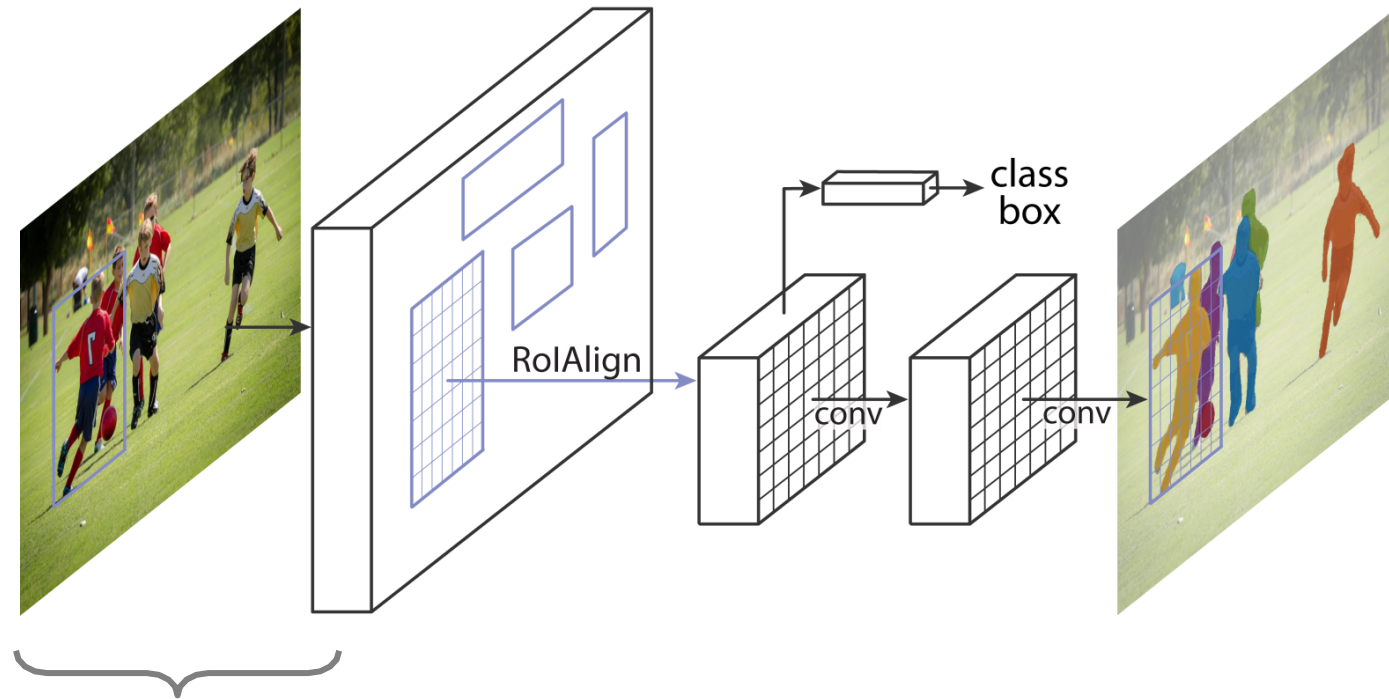


Mask R-CNN

# Invariance vs. Equivariance

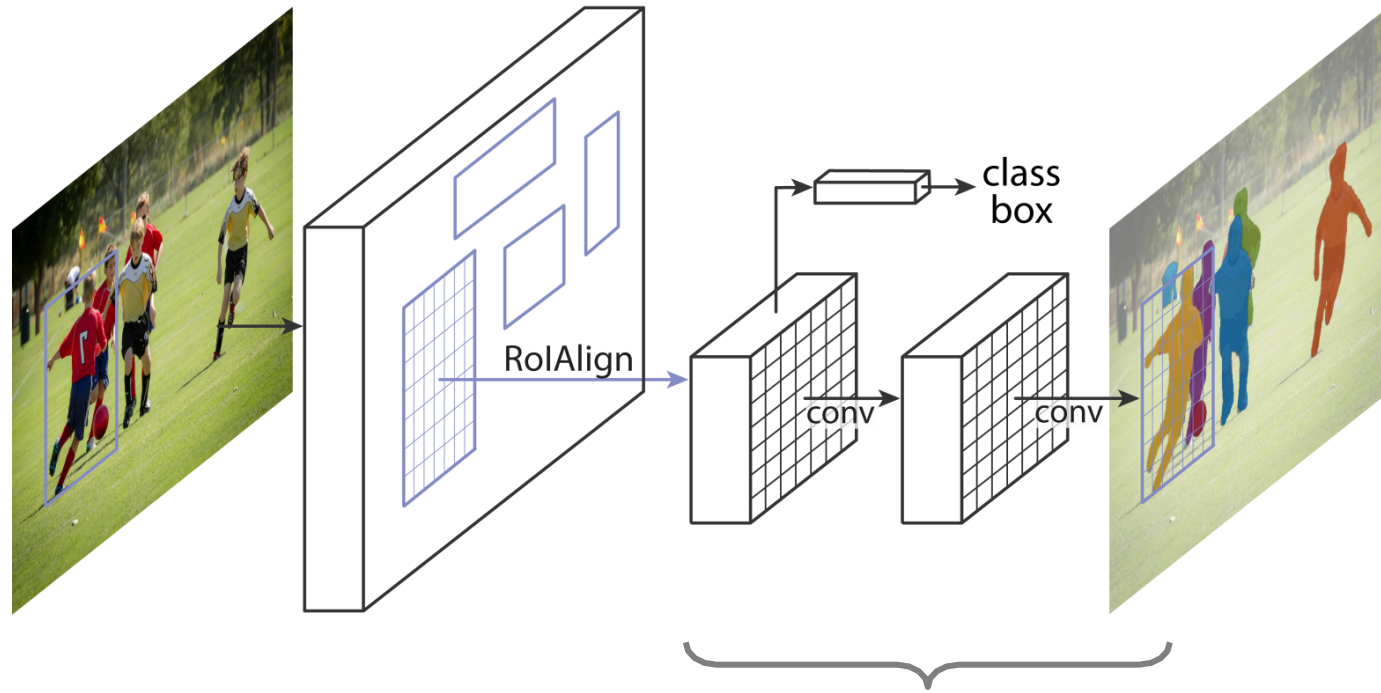
- **Equivariance**: changes in input lead to corresponding changes in output
- *Classification* desires *invariant* representations: output a label
- *Instance Seg.* desires *equivariant* representations:
  - Translated object => translated mask
  - Scaled object => scaled mask
  - *Big and small* objects are equally important (due to AP metric)
    - unlike semantic seg. (counting pixels)

# Equivariance in Mask R-CNN



1. Fully-Conv Features:  
equivariant to global (image) translation

# Equivariance in Mask R-CNN

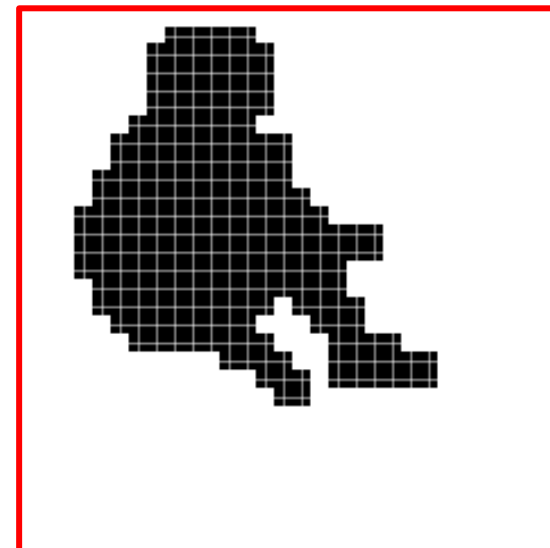
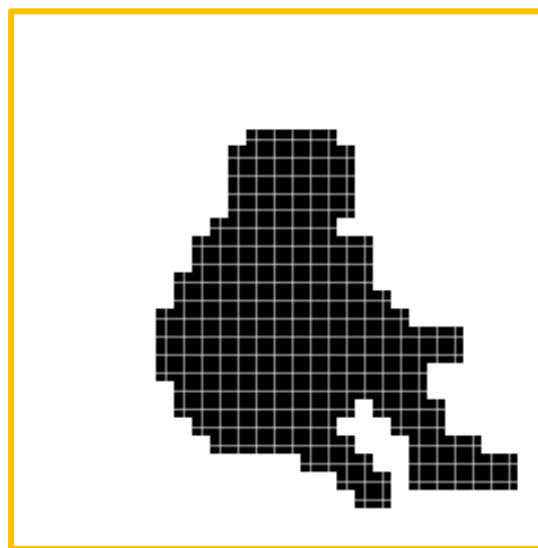


2. Fully-Conv on RoI:  
equivariant to translation within RoI

# Fully-Conv on Rol



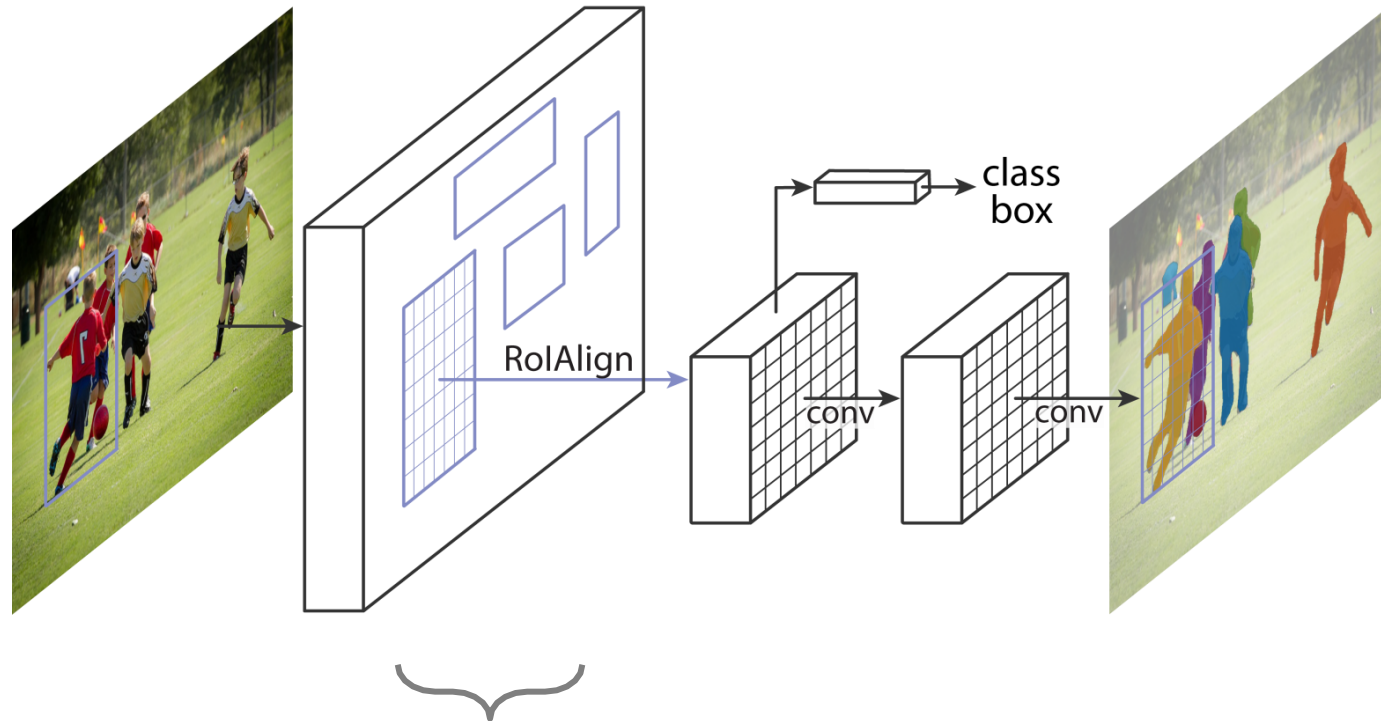
target masks on Rols



Translation of object in Rol => Same translation of mask in Rol

- Equivariant to small translation of Rols
- More robust to Rol's localization imperfection

# Equivariance in Mask R-CNN



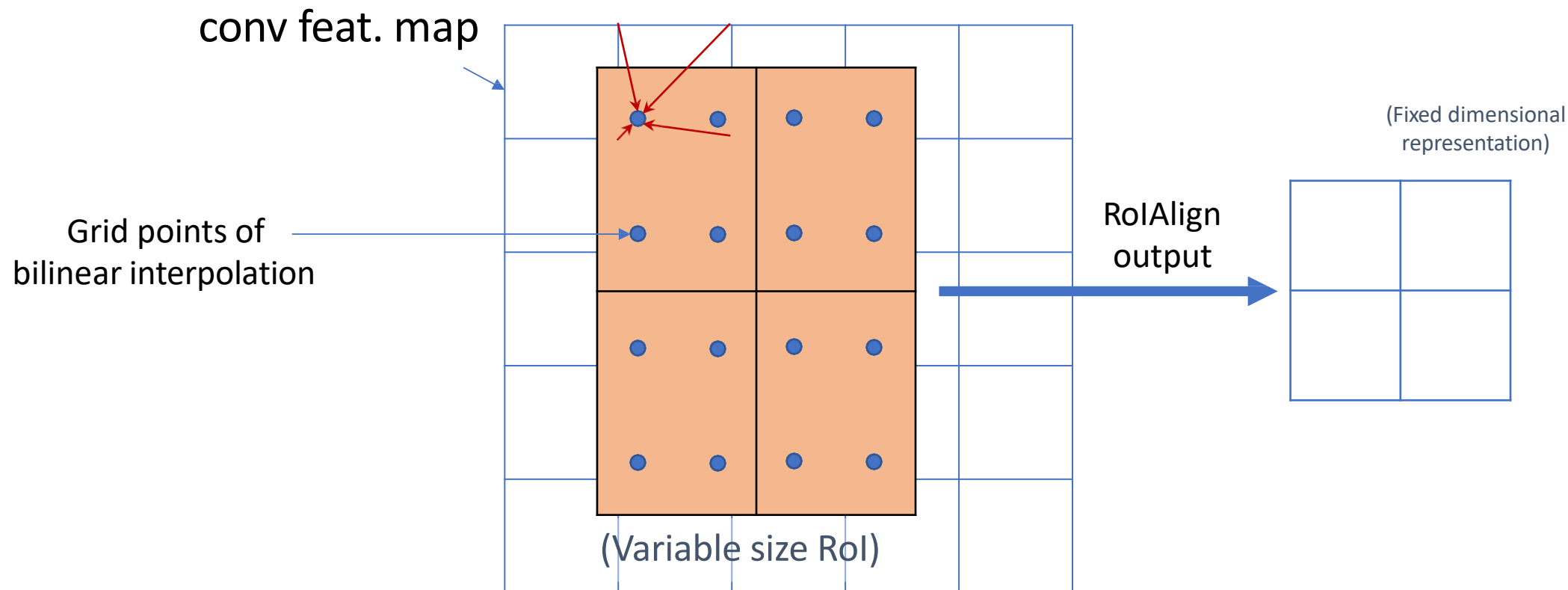
3. RoIAlign:

3a. maintain translation-equivariance before/after RoI

# RoIAlign

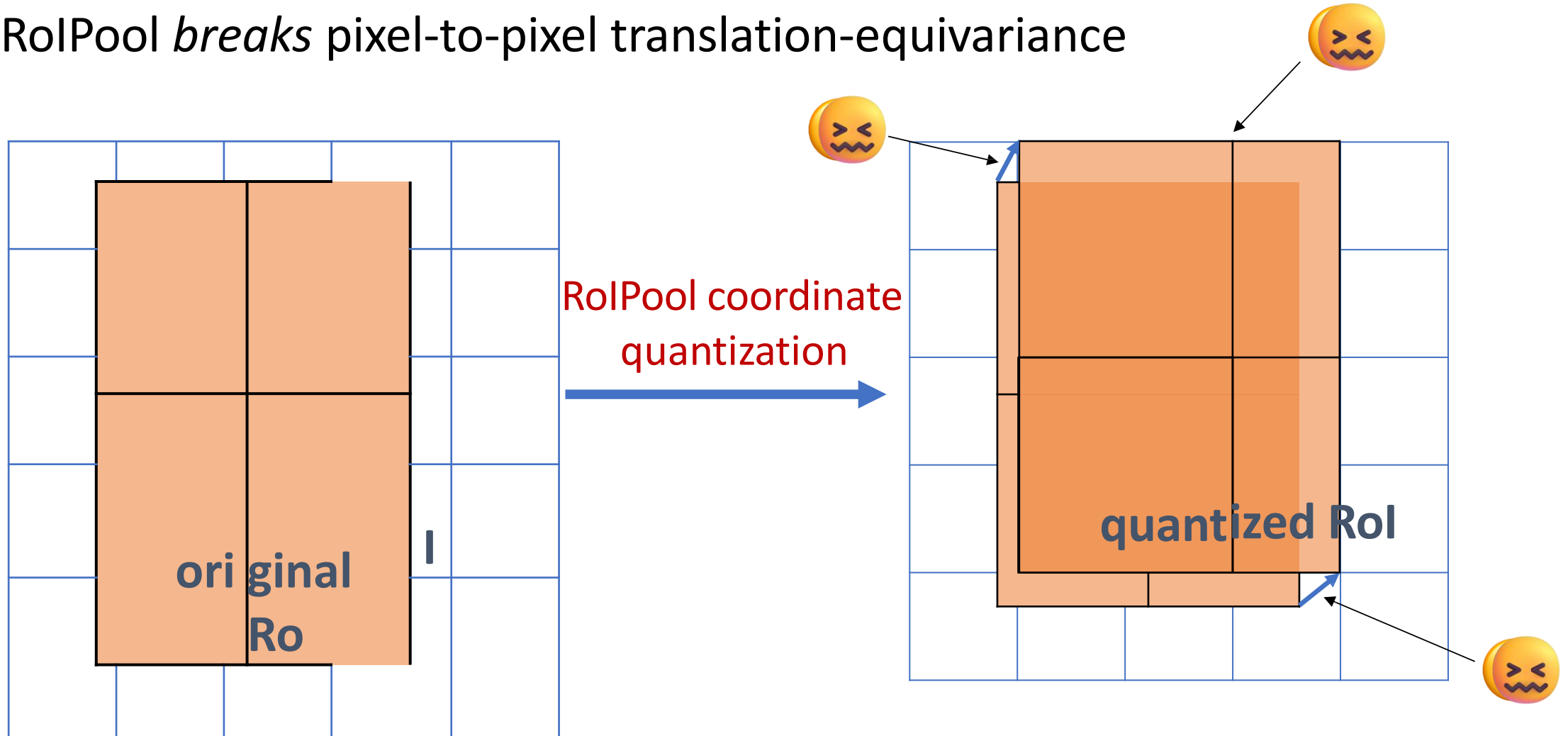
FAQs: how to sample grid points within a cell?

- 4 regular points in 2x2 sub-cells
- other implementation could work

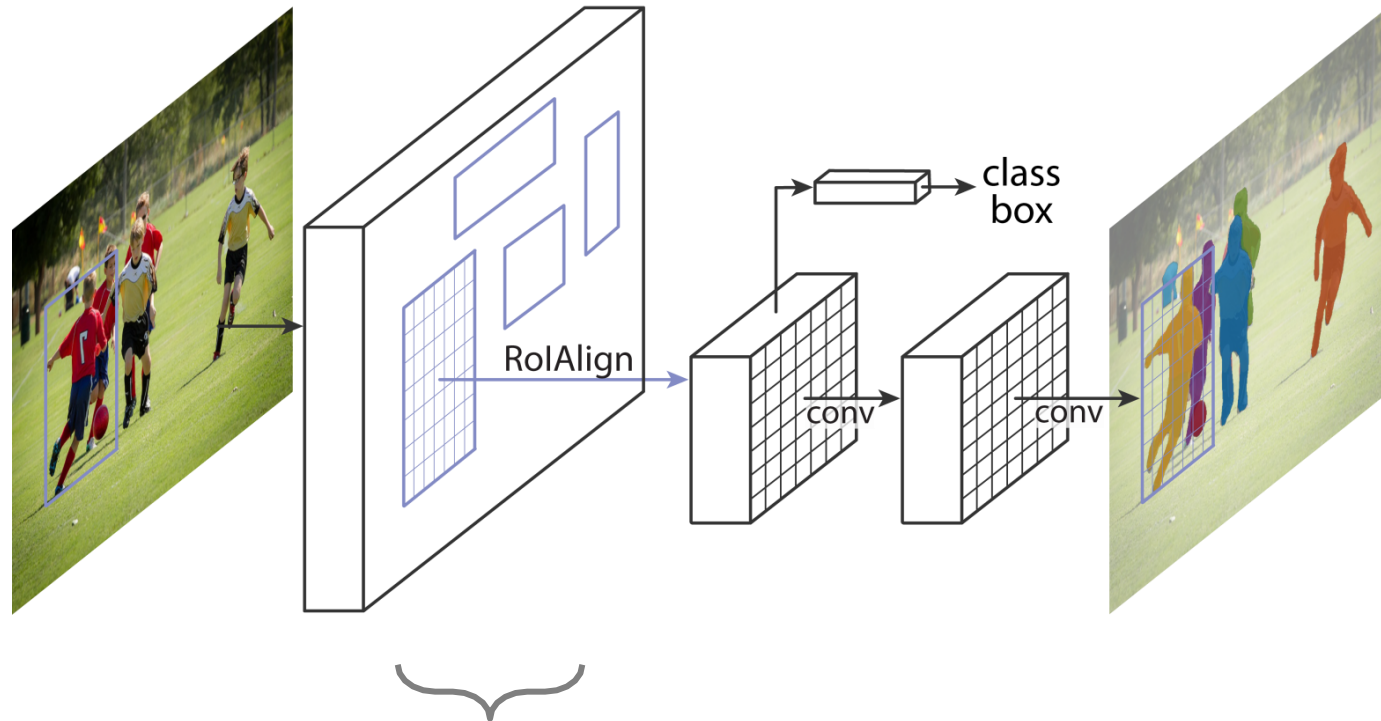


# RoIAlign vs. RoIPool

- RoIPool *breaks* pixel-to-pixel translation-equivariance



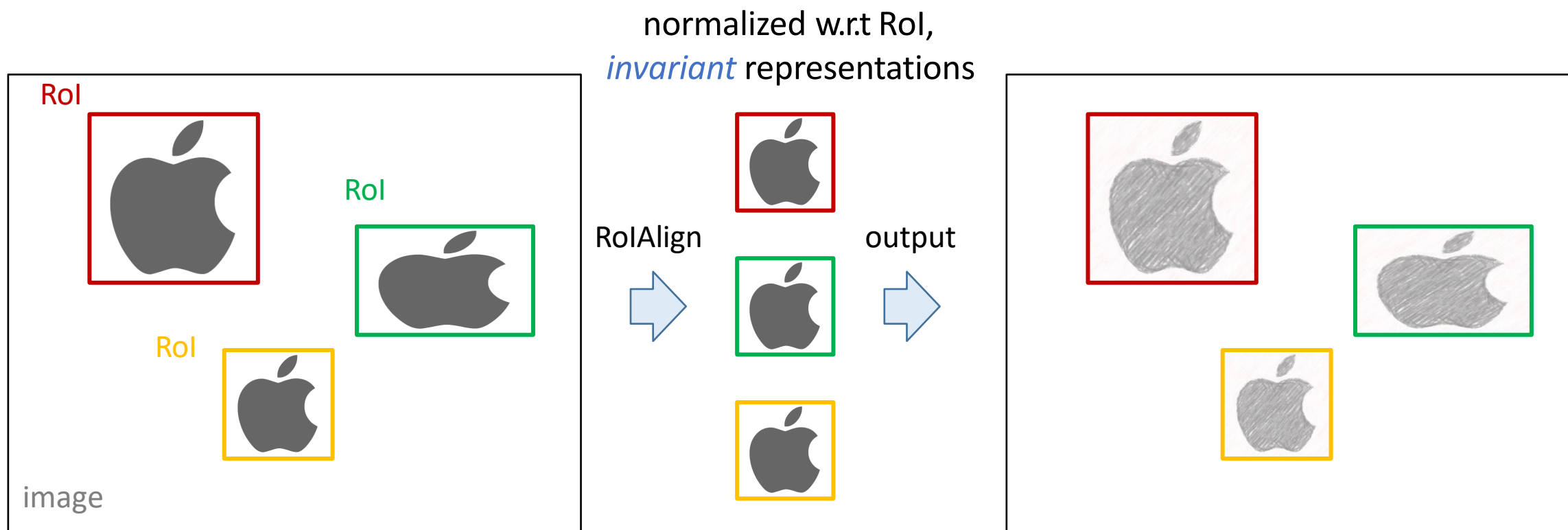
# Equivariance in Mask R-CNN



3. RoIAlign:

3b. Scale-equivariant (and aspect-ratio-equivariant)

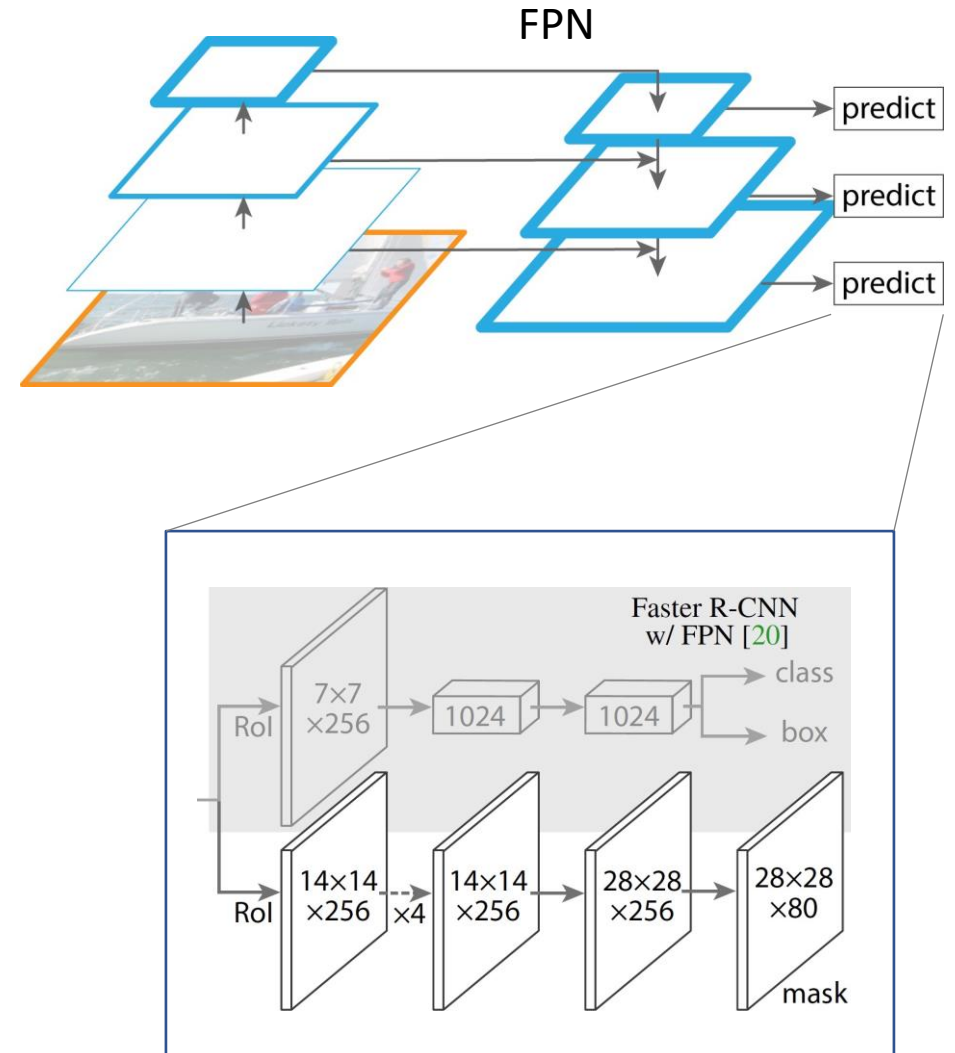
# RoIAlign: Scale-Equivariance



- RoIAlign creates *scale-invariant* representations
- RoIAlign + “output pasted back” provides *scale-equivariance*

# More about Scale-Equivariance: FPN

- RoIAlign is scale-invariant if **on raw pixels**:
  - = (slow) R-CNN: crops and warps Rols
- RoIAlign is scale-invariant if on **scale-invariant feature maps**
- Feature Pyramid Network (FPN) [Lin et al. CVPR'17] creates approx. scale-invariant features

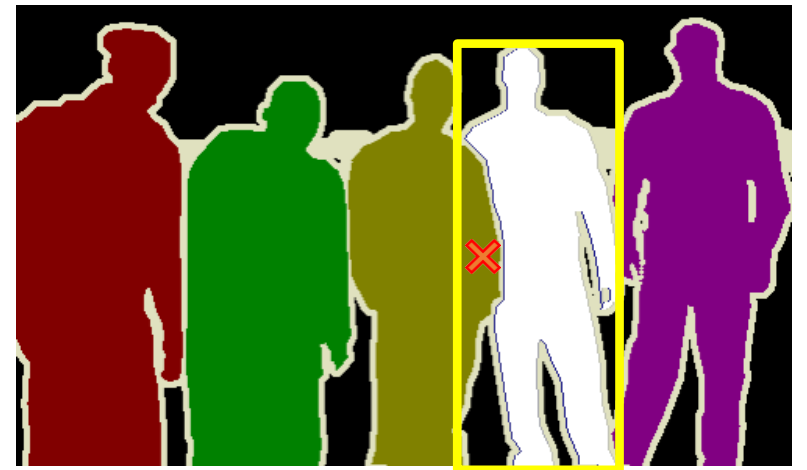


# Equivariance in Mask R-CNN: Summary

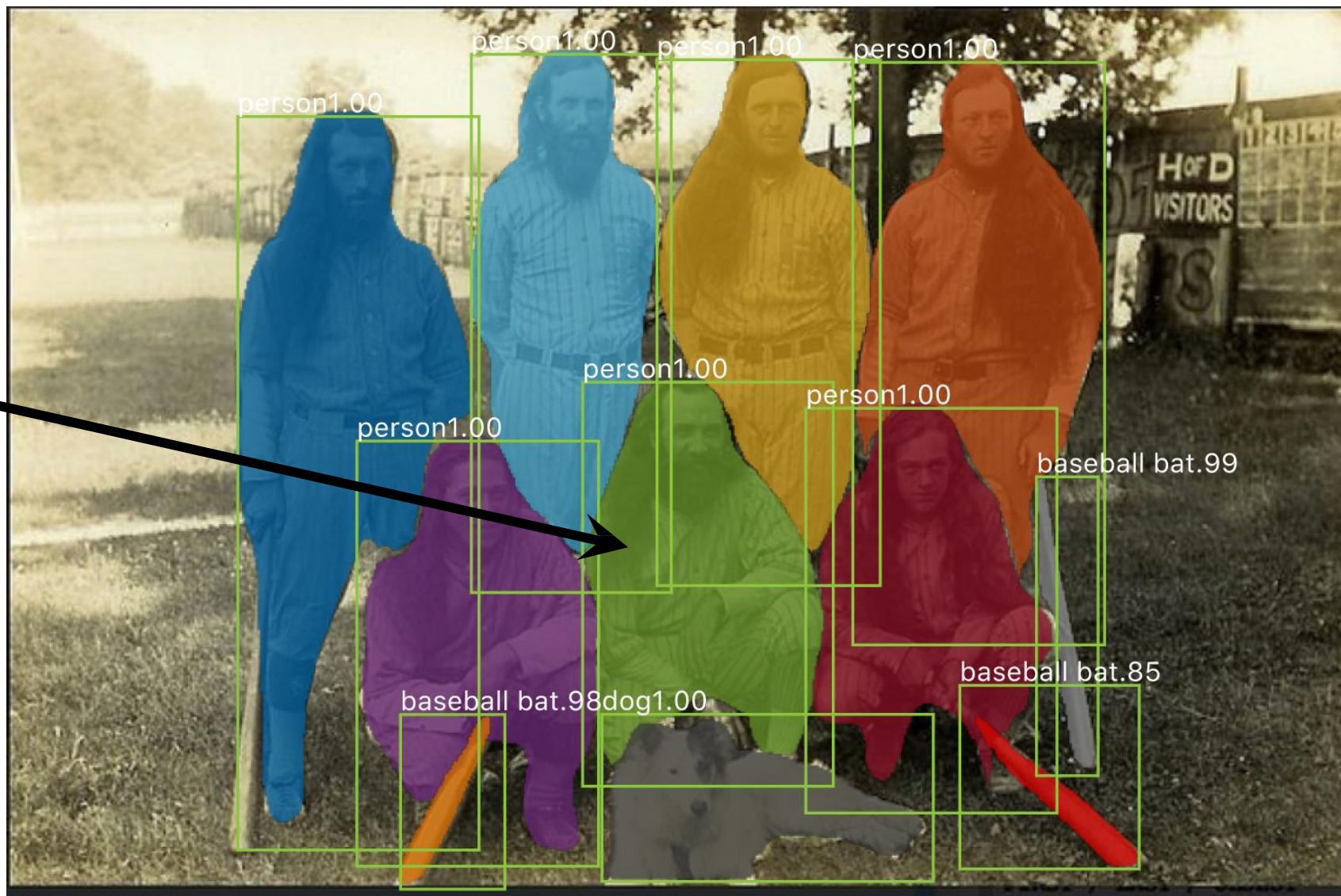
- Translation-equivariant
  - FCN features
  - FCN mask head
  - RoIAlign (pixel-to-pixel behavior)
- Scale-equivariant (and aspect-ratio-equivariant)
  - RoIAlign (warping and normalization behavior) + paste-back
  - FPN features

# Instance Seg: When we don't want equivariance?

- A pixel  $x$  could have a different label w.r.t. different Rols
  - zero-padding in Rol boundary breaks equivariance
  - outside objects are suppressed
  - only **equivariant to small changes** of Rols (which is desired)



object  
surrounded by  
same-category  
objects



Mask R-CNN results on COCO

# Result Analysis

# Instance Segmentation Results on COCO

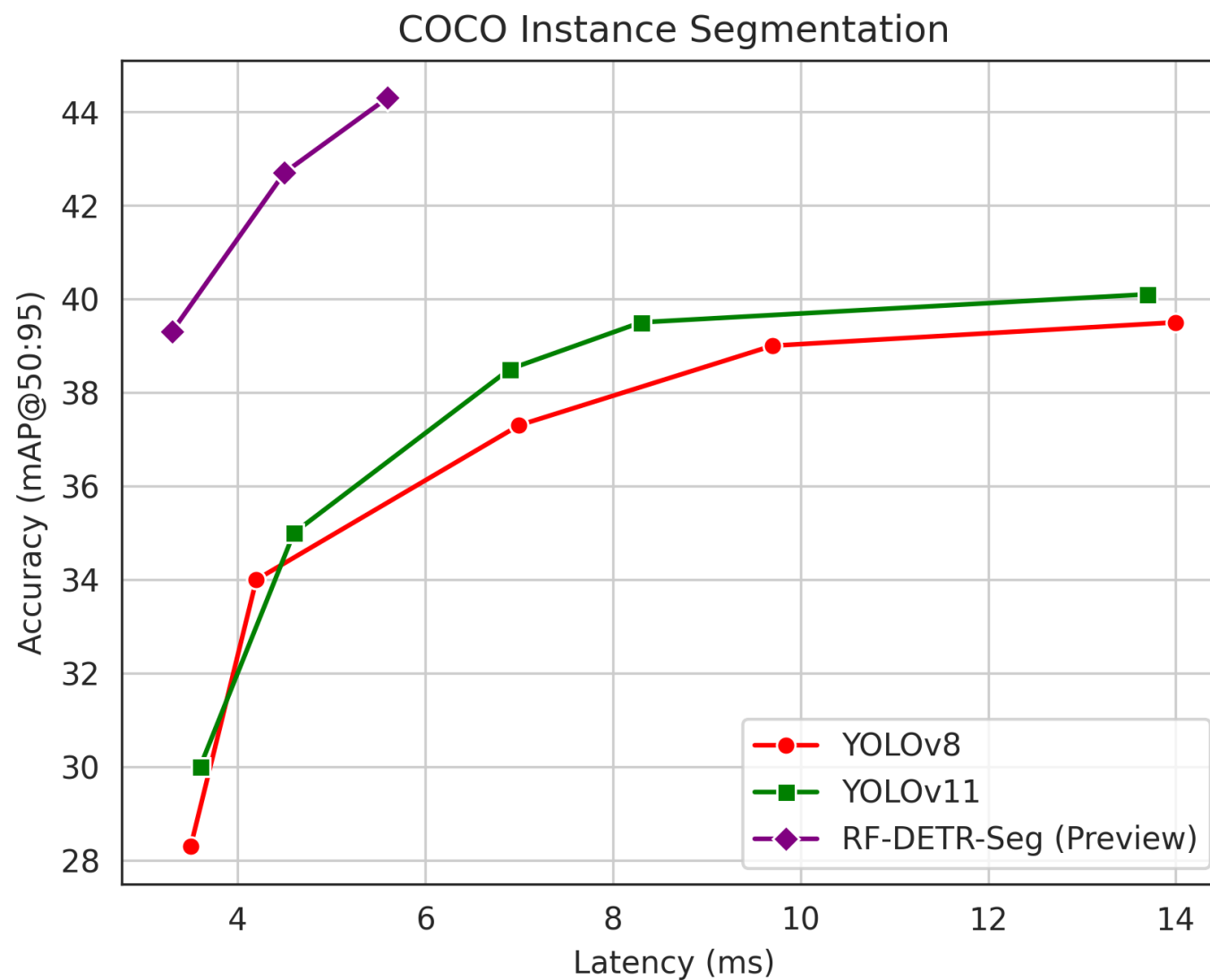
|                    | backbone              | AP          | AP <sub>50</sub> | AP <sub>75</sub> | AP <sub>S</sub> | AP <sub>M</sub> | AP <sub>L</sub> |
|--------------------|-----------------------|-------------|------------------|------------------|-----------------|-----------------|-----------------|
| MNC [7]            | ResNet-101-C4         | 24.6        | 44.3             | 24.8             | 4.7             | 25.9            | 43.6            |
| FCIS [20] +OHEM    | ResNet-101-C5-dilated | 29.2        | 49.5             | -                | 7.1             | 31.3            | 50.0            |
| FCIS+++ [20] +OHEM | ResNet-101-C5-dilated | 33.6        | 54.5             | -                | -               | -               | -               |
| Mask R-CNN         | ResNet-101-C4         | 33.1        | 54.9             | 34.8             | 12.1            | 35.6            | 51.1            |
| Mask R-CNN         | ResNet-101-FPN        | 35.7        | 58.0             | 37.8             | 15.5            | 38.1            | 52.4            |
| Mask R-CNN         | ResNeXt-101-FPN       | <b>37.1</b> | <b>60.0</b>      | <b>39.4</b>      | <b>16.9</b>     | <b>39.9</b>     | <b>53.5</b>     |

- **2 AP better** than SOTA w/ R101, without bells and whistles
- **200ms / img**

# Instance Segmentation Results on COCO

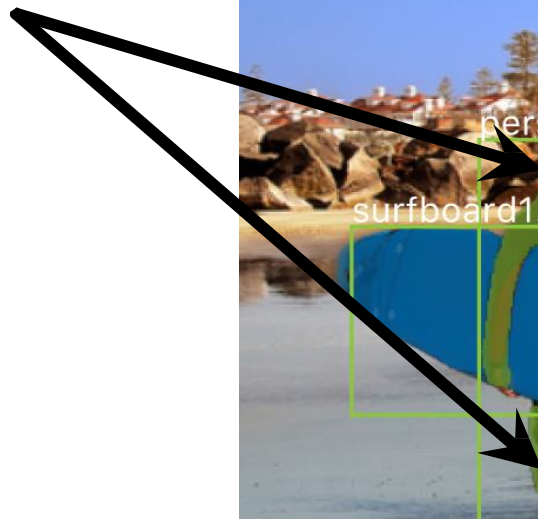
|                    | backbone              | AP          | AP <sub>50</sub> | AP <sub>75</sub> | AP <sub>S</sub> | AP <sub>M</sub> | AP <sub>L</sub> |
|--------------------|-----------------------|-------------|------------------|------------------|-----------------|-----------------|-----------------|
| MNC [7]            | ResNet-101-C4         | 24.6        | 44.3             | 24.8             | 4.7             | 25.9            | 43.6            |
| FCIS [20] +OHEM    | ResNet-101-C5-dilated | 29.2        | 49.5             | -                | 7.1             | 31.3            | 50.0            |
| FCIS+++ [20] +OHEM | ResNet-101-C5-dilated | 33.6        | 54.5             | -                | -               | -               | -               |
| <b>Mask R-CNN</b>  | ResNet-101-C4         | 33.1        | 54.9             | 34.8             | 12.1            | 35.6            | 51.1            |
| <b>Mask R-CNN</b>  | ResNet-101-FPN        | 35.7        | 58.0             | 37.8             | 15.5            | 38.1            | 52.4            |
| <b>Mask R-CNN</b>  | ResNeXt-101-FPN       | <b>37.1</b> | <b>60.0</b>      | <b>39.4</b>      | <b>16.9</b>     | <b>39.9</b>     | <b>53.5</b>     |

- benefit from better features (ResNeXt [Xie et al. CVPR'17])

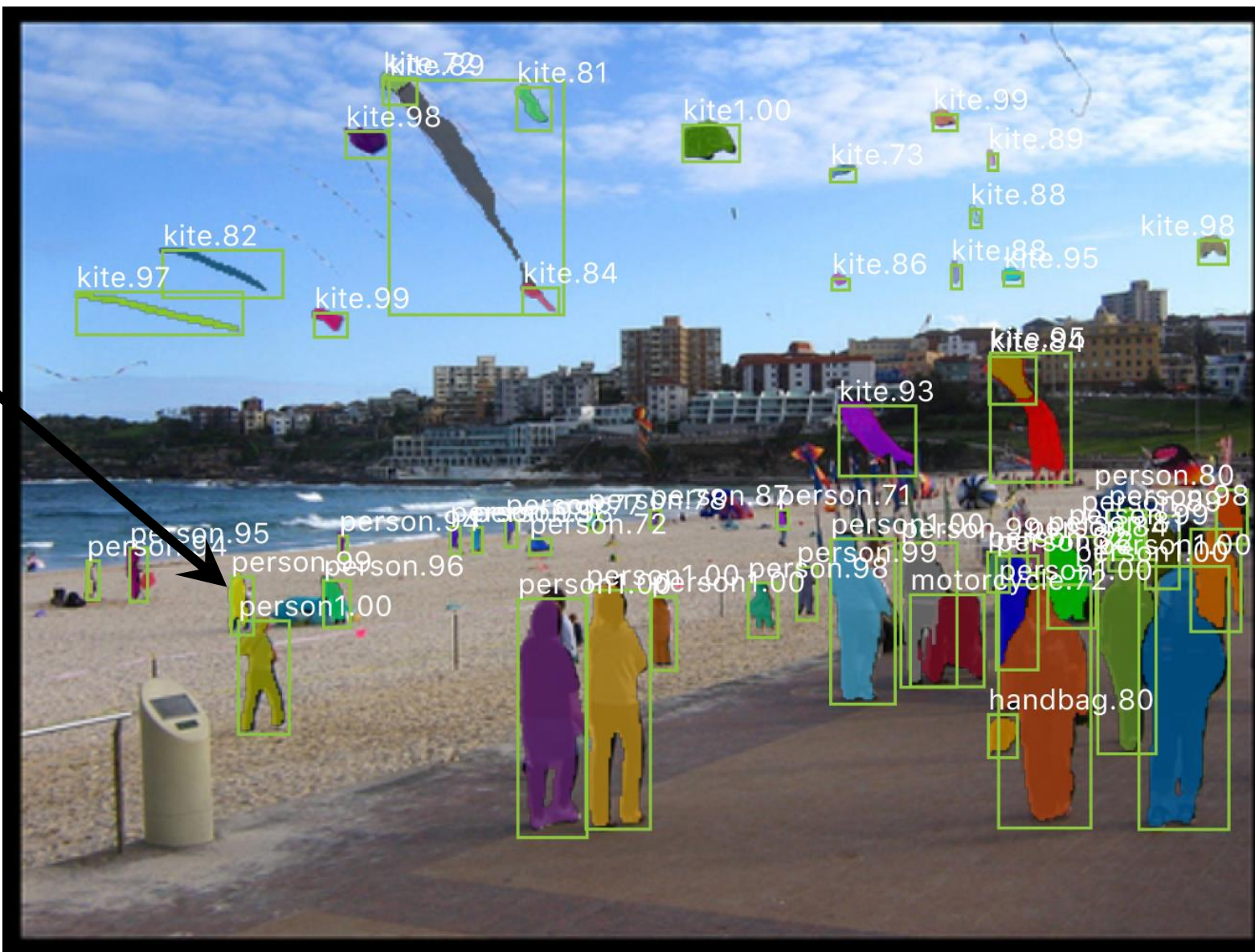


State of the art as of 2025

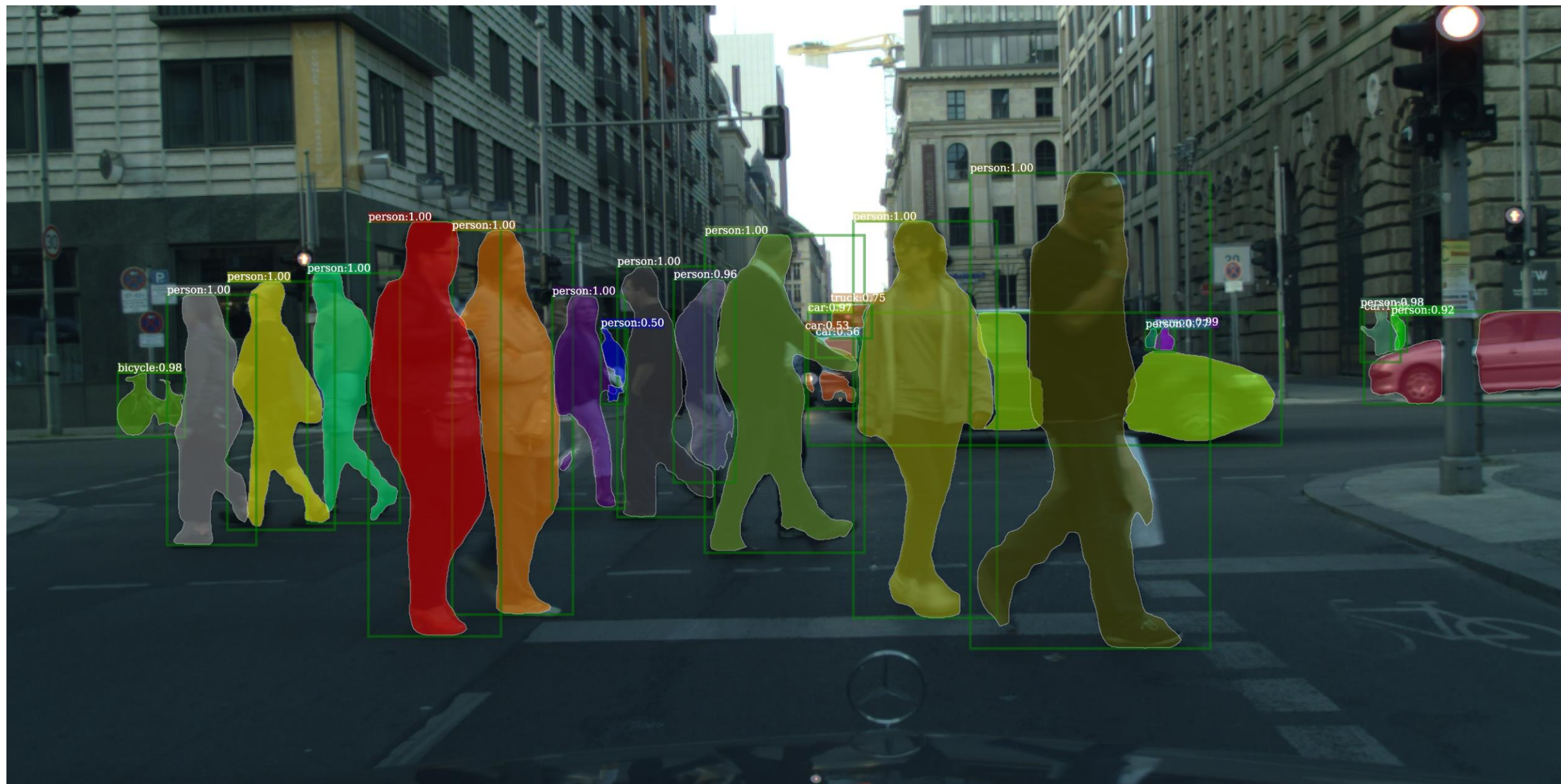
disconnected  
object



Mask R-CNN results on COCO



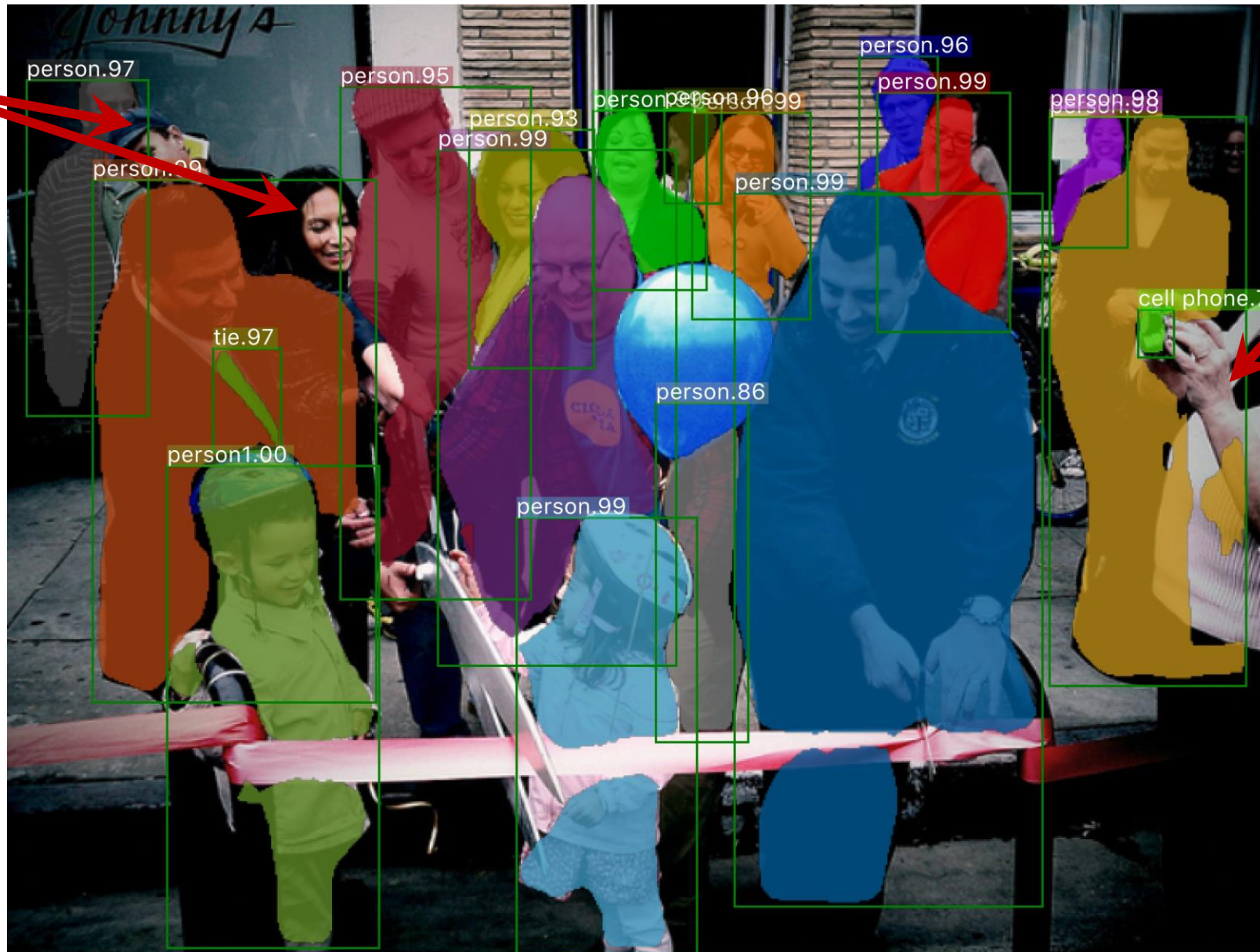
## Mask R-CNN results on COCO



Mask R-CNN results on CityScapes

# Failure case: detection/segmentation

missing

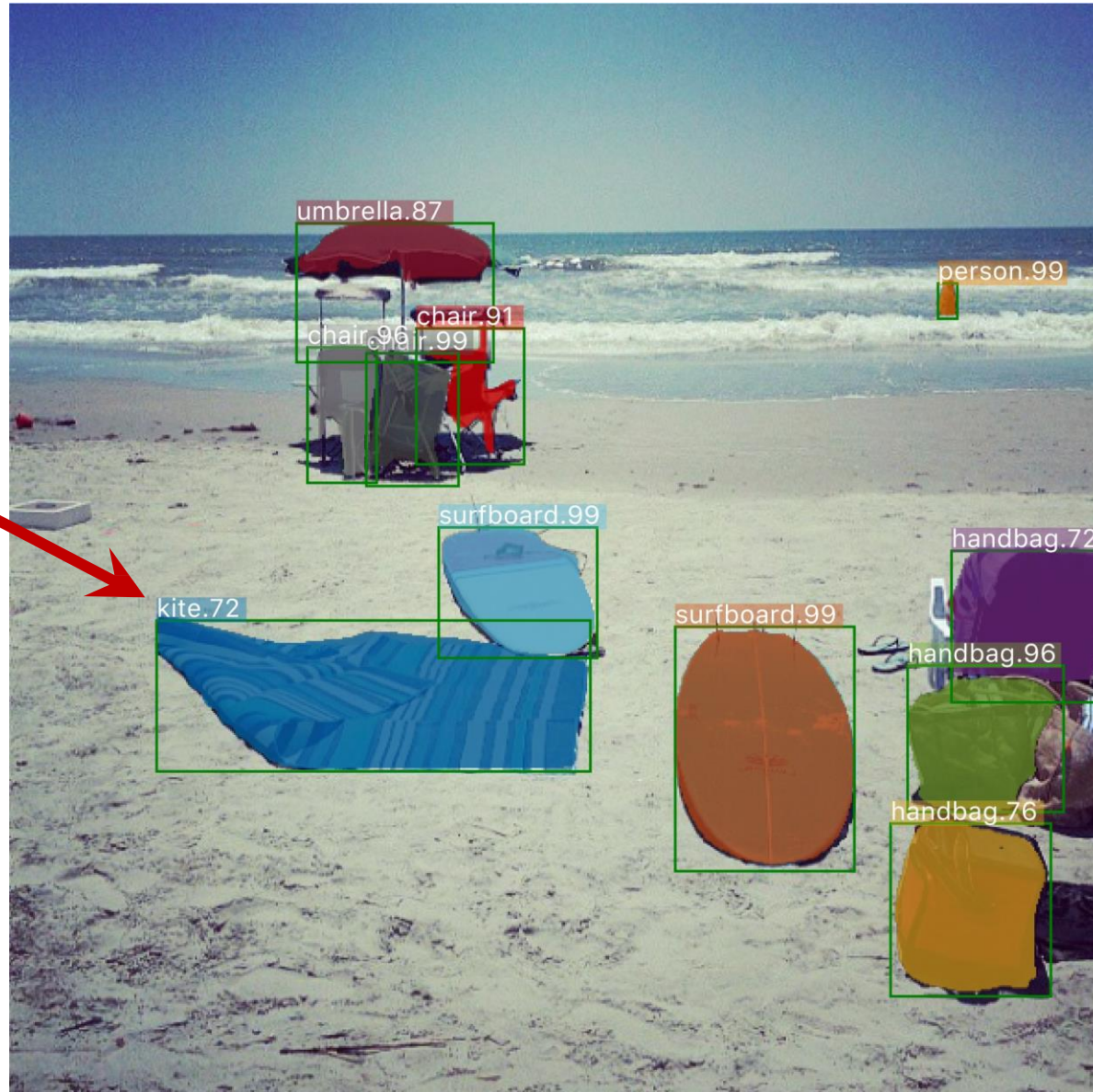
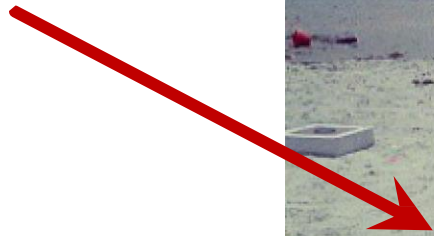


missing,  
false mask

Mask R-CNN results on COCO

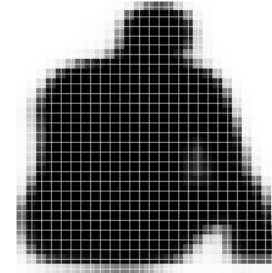
# Failure case: recognition

not a kite



Mask R-CNN results on COCO

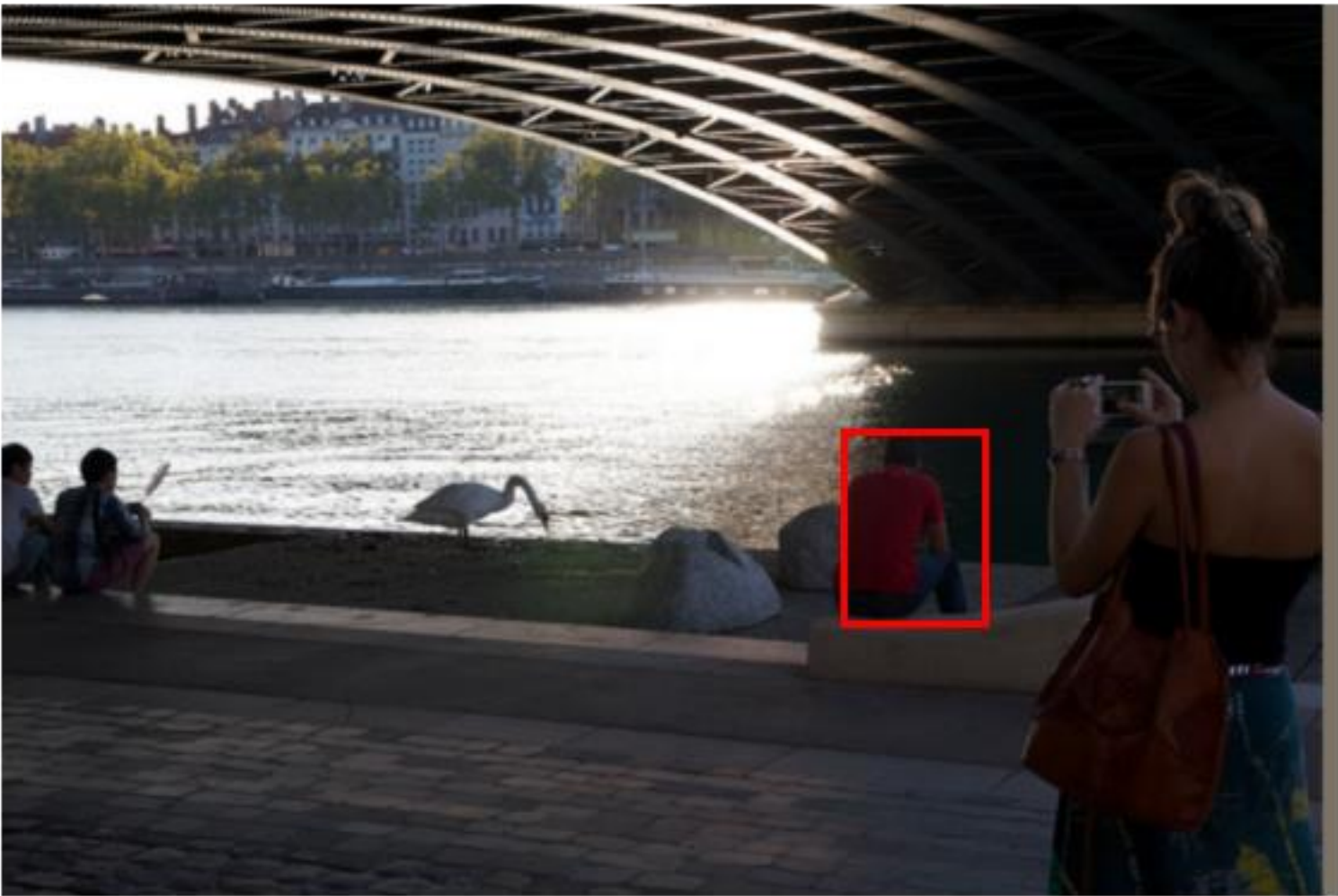
28x28 soft prediction from Mask R-CNN  
(enlarged)



Soft prediction **resampled to image coordinates**  
(bilinear and bicubic interpolation work equally well)



Final prediction (threshold at 0.5)

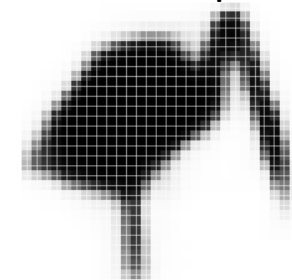


Validation image with box detection shown in red



Validation image with box detection shown in red

28x28 soft prediction



Resized Soft prediction

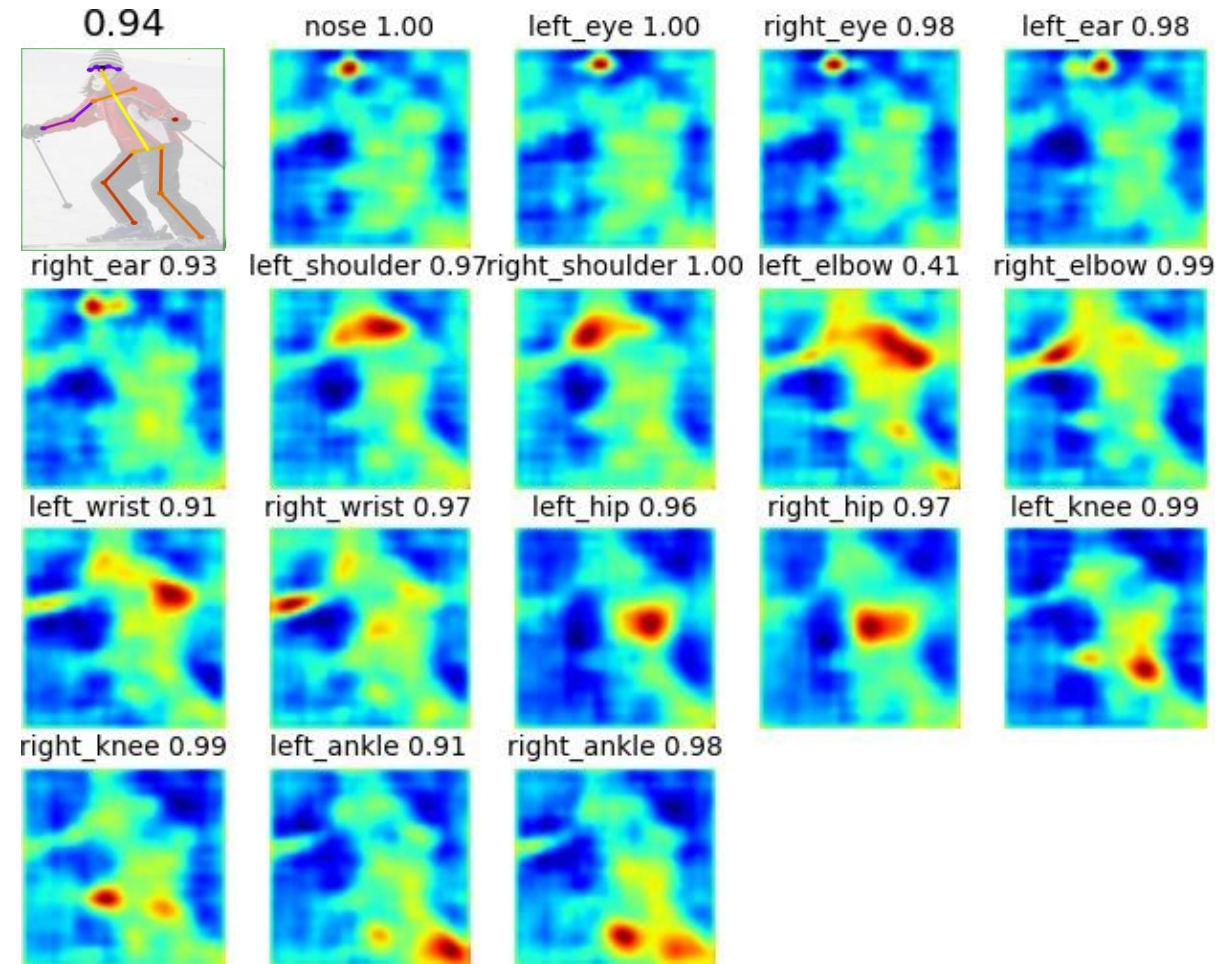


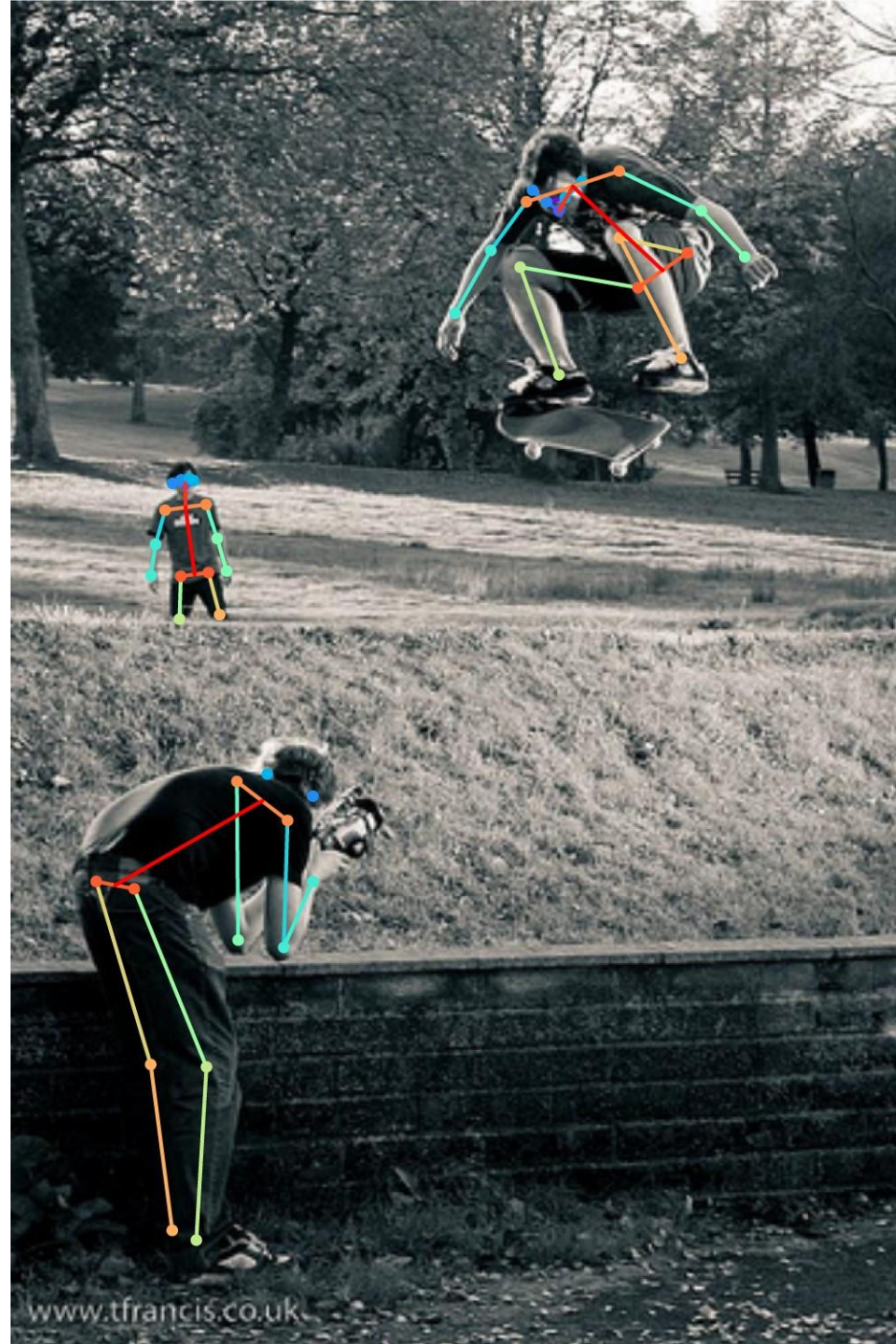
Final mask



# Mask R-CNN: for Human Keypoint Detection

- 1 keypoint = 1-hot “mask”
- Human pose = 17 masks
- Softmax over **spatial locations**
  - e.g.  $56^2$ -way softmax on  $56 \times 56$
- Desire the same equivariances
  - translation, scale, aspect ratio

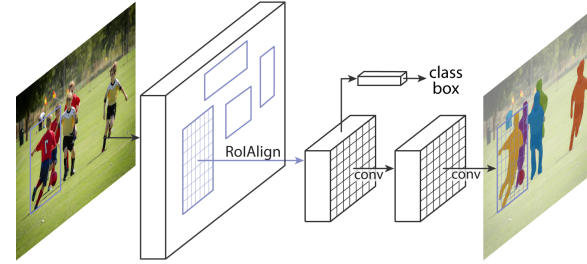




# Conclusion

## Mask R-CNN

- ✓ Good speed
- ✓ Good accuracy
- ✓ Intuitive
- ✓ Easy to use
- ✓ Equivariance matters



Code open-sourced as Facebook AI  
Research's **Detectron** platform

# Summary – More complex outputs from deep networks

- Image Output (e.g. colorization, semantic segmentation, super-resolution, stylization, depth estimation...)
- Attributes
- Text Captions
- Bottom up: Semantic Keypoints
- Top down: Object Detection
  - “single shot” vs “two stage”