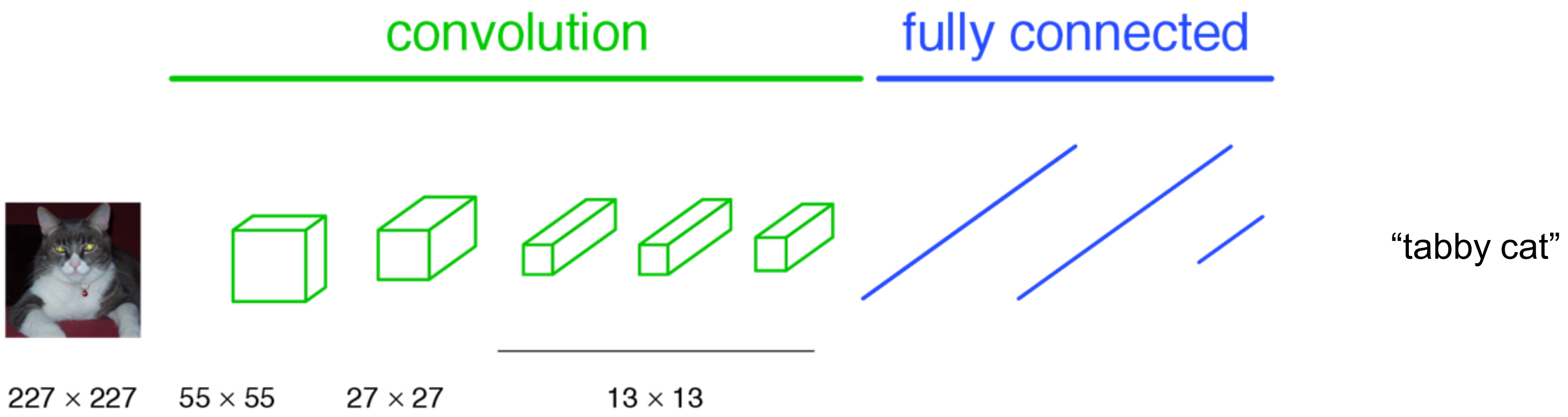
Structured Predictions with Deep Learning

James Hays

Outline – More complex outputs from deep networks

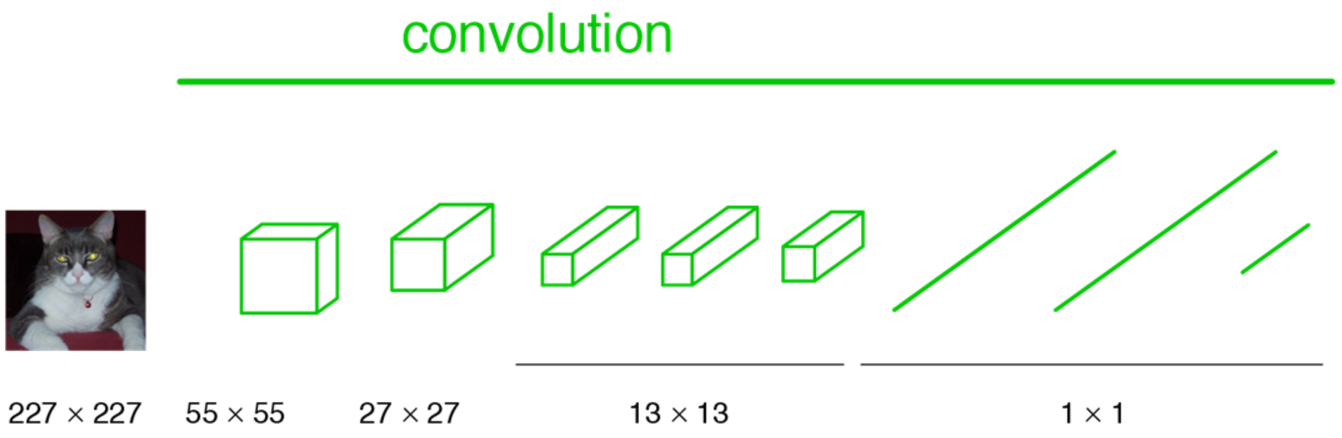
- Image Output (e.g. colorization, semantic segmentation, super-resolution, stylization, depth estimation…)
- Attributes
- Text Captions
- Semantic Keypoints
- Object Detection

a classification network



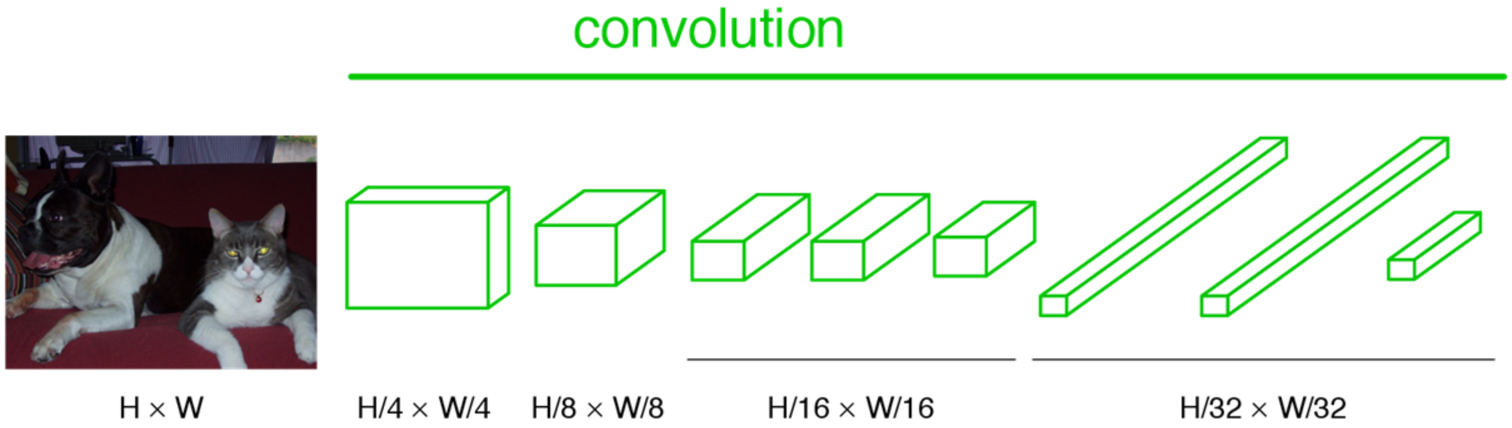
Fully Convolutional Networks for Semantic Segmentation.
Jon Long, Evan Shelhamer, Trevor Darrell. CVPR 2015

becoming fully convolutional

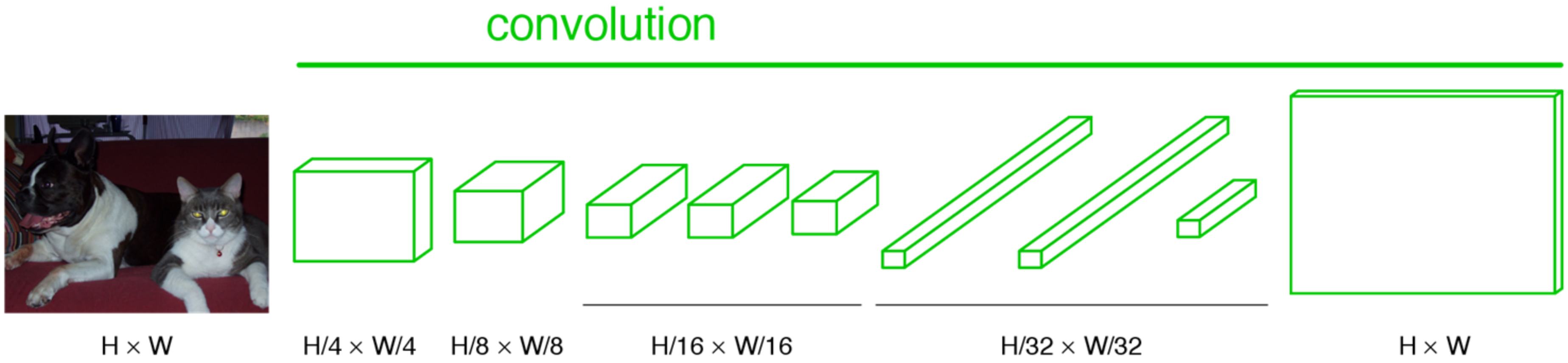


Note: “Fully Convolutional” and “Fully Connected” aren’t the same thing.
They’re almost opposites, in fact.

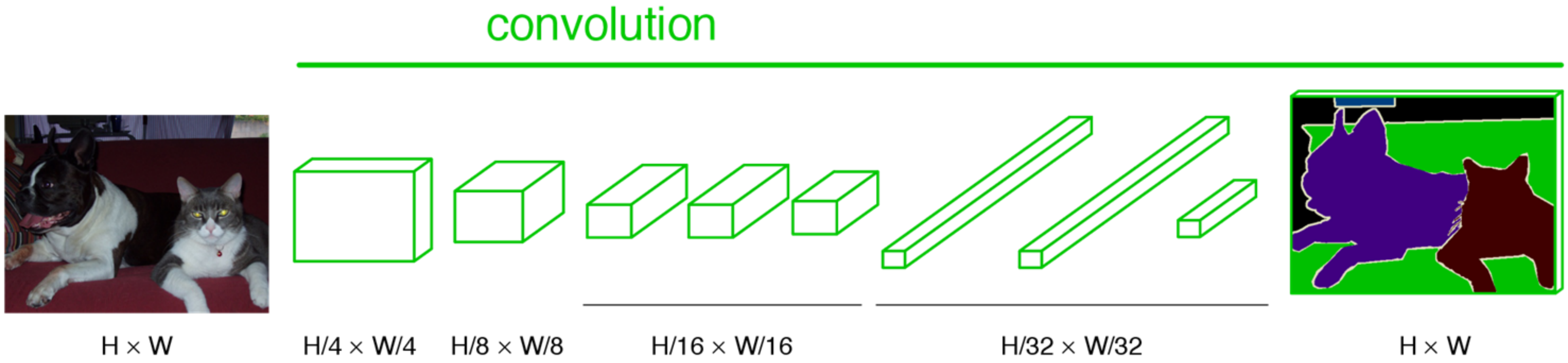
becoming fully convolutional



upsampling output

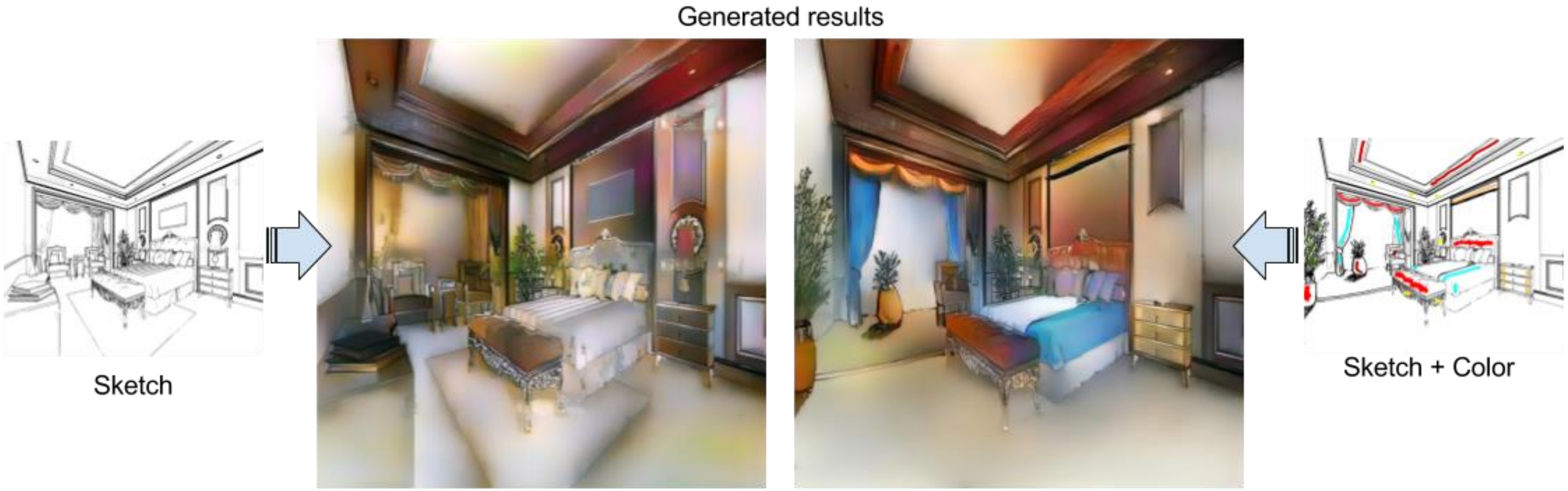


end-to-end, pixels-to-pixels network



What if we want other types of outputs?

- Easy*: Predict any fixed dimensional output, whether a feature (embedding networks) or an image.



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.

What if we want other types of outputs?

- Easy: Predict any number of labels (with classification, there will be just one best answer, but for other labels like attributes dozens could be appropriate for an image)

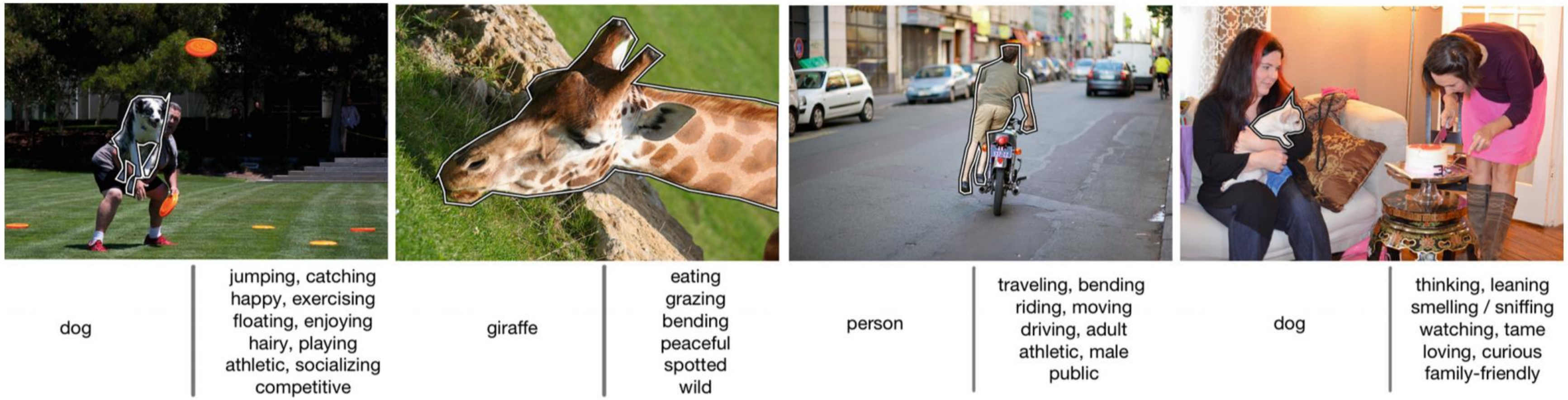


Fig. 1. *Examples from COCO Attributes.* In the figure above, images from the COCO dataset are shown with one object outlined in white. Under the image, the COCO object label is listed on the left, and the COCO Attribute labels are listed on the right. The COCO Attributes labels give a rich and detailed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

What if we want other types of outputs?

- Hard: Outputs with varying dimensionality or cardinality
    - A natural language image caption
    - An arbitrary number of human keypoints (17 points each)
    - An arbitrary number of bounding boxes (4 parameters each)
- Today we will examine influential methods for keypoint prediction and object detection

Realtime Multi-Person Pose Estimation using Part Affinity Fields

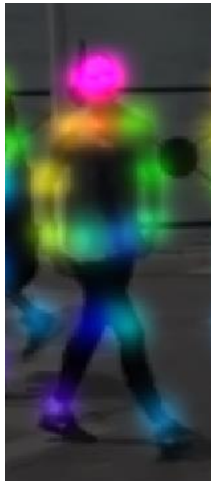
Zhe Cao, Tomas Simon, Shih-En Wei, Yaser Sheikh
Carnegie Mellon University

Human Pose Estimation

Human Pose Estimation

Single-Person Pose Estimation

Single-Person Pose Estimation

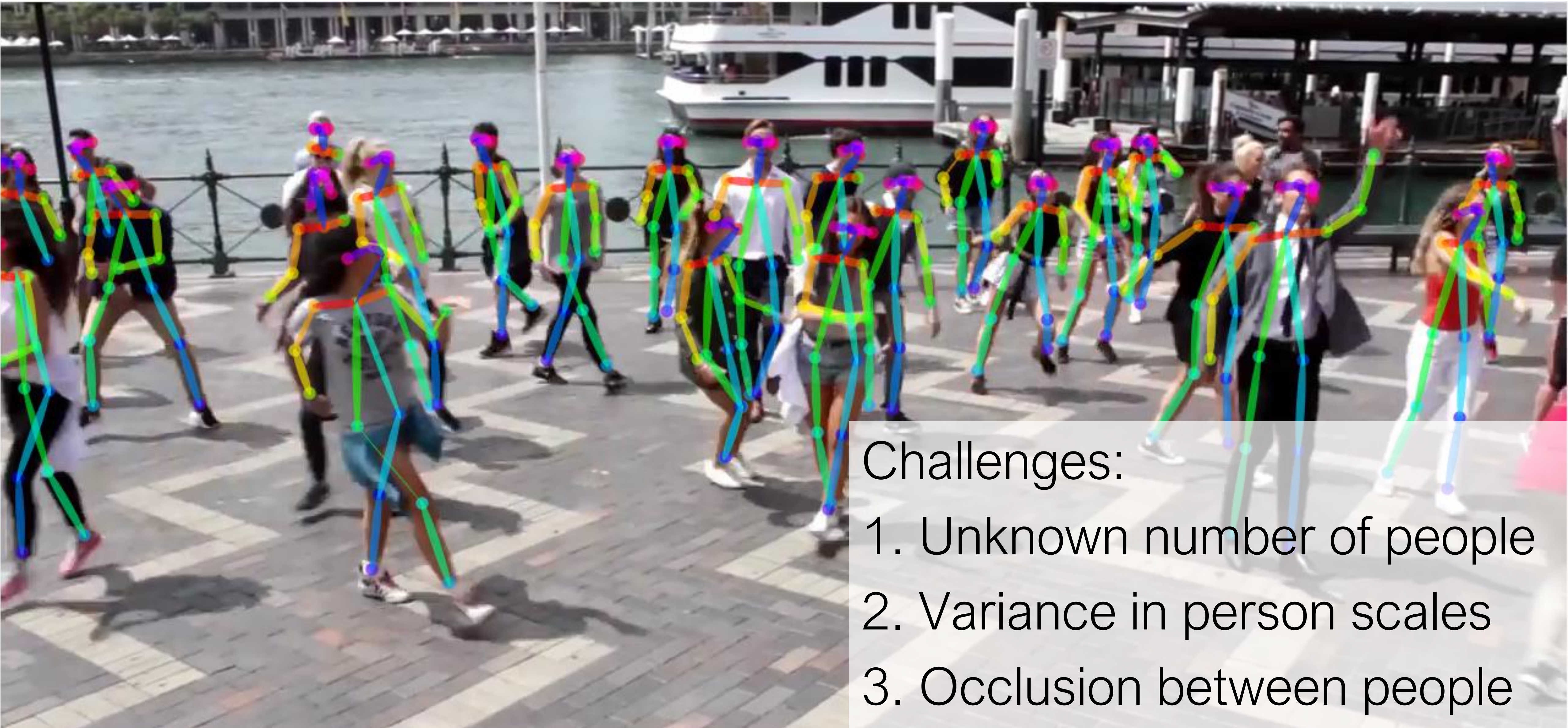


Multi-Person Pose Estimation

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

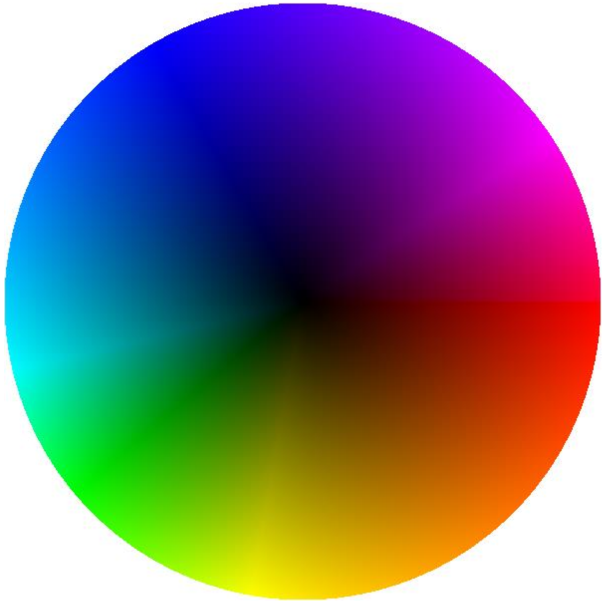
For 30 people and each with 17 joints, there are in total **1.3 x 10^5** pair-wise connection cost, NP-hard optimization

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

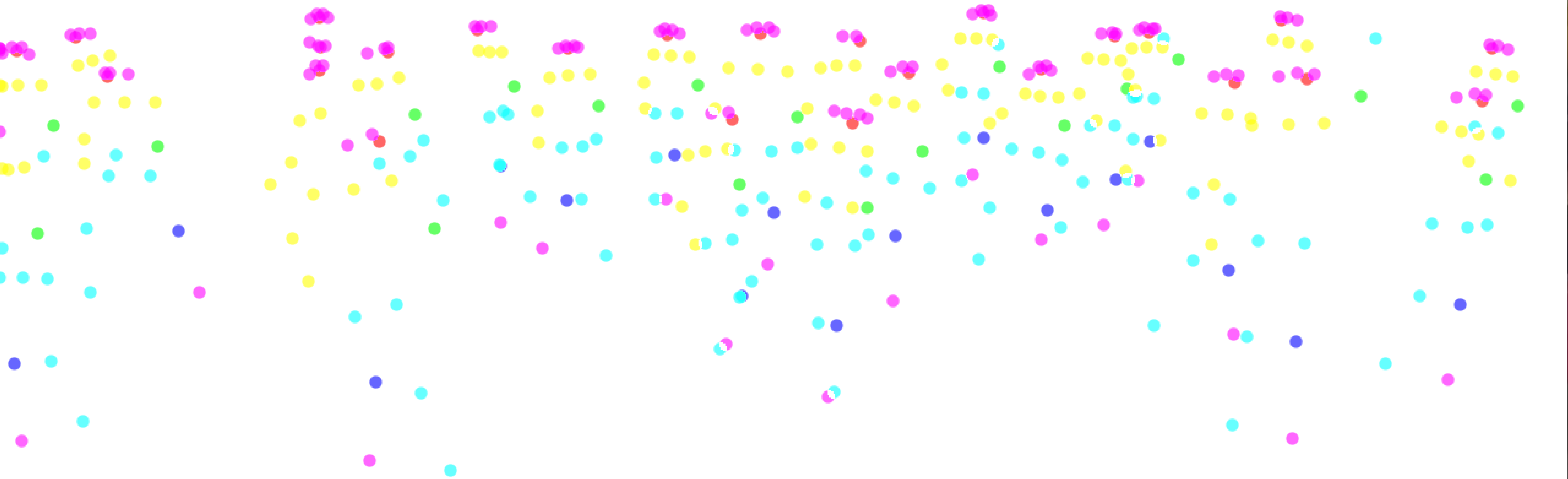


Part Affinity Field between right elbow and wrist

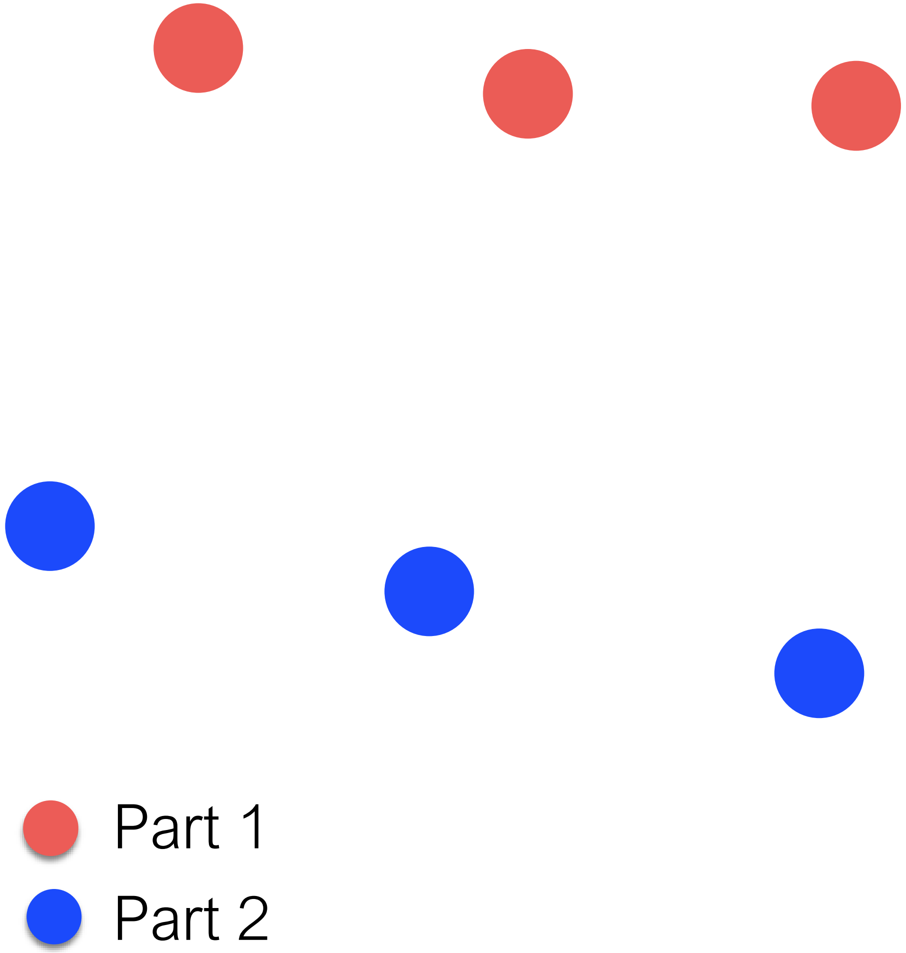
Novelty: Part Affinity Fields for Parts Association

Part Affinity Field between right elbow and wrist

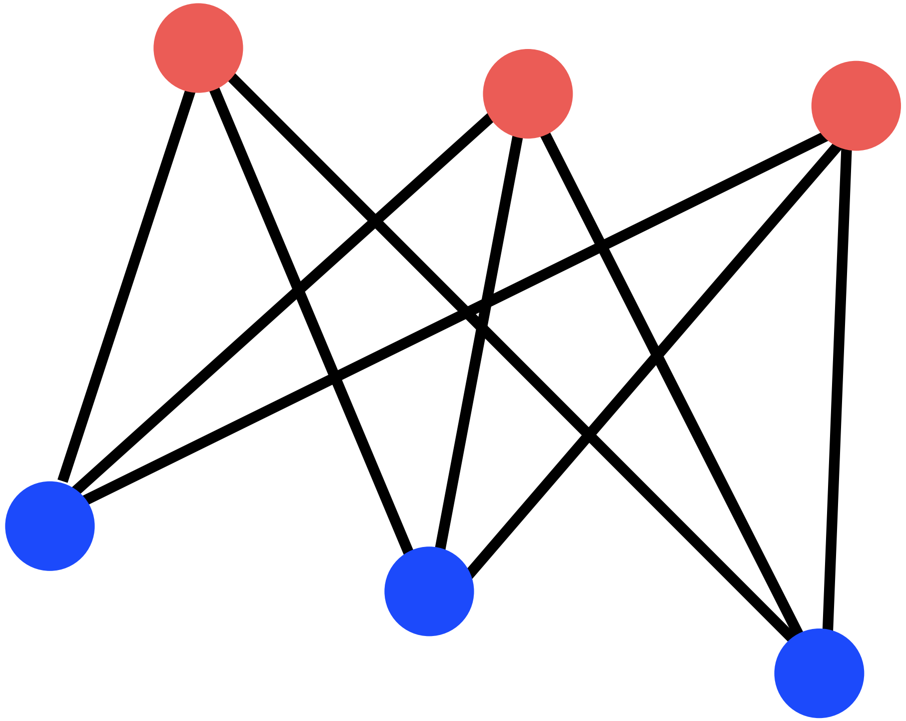
Novelty: Part Affinity Fields for Parts Association



Part-to-Part Association

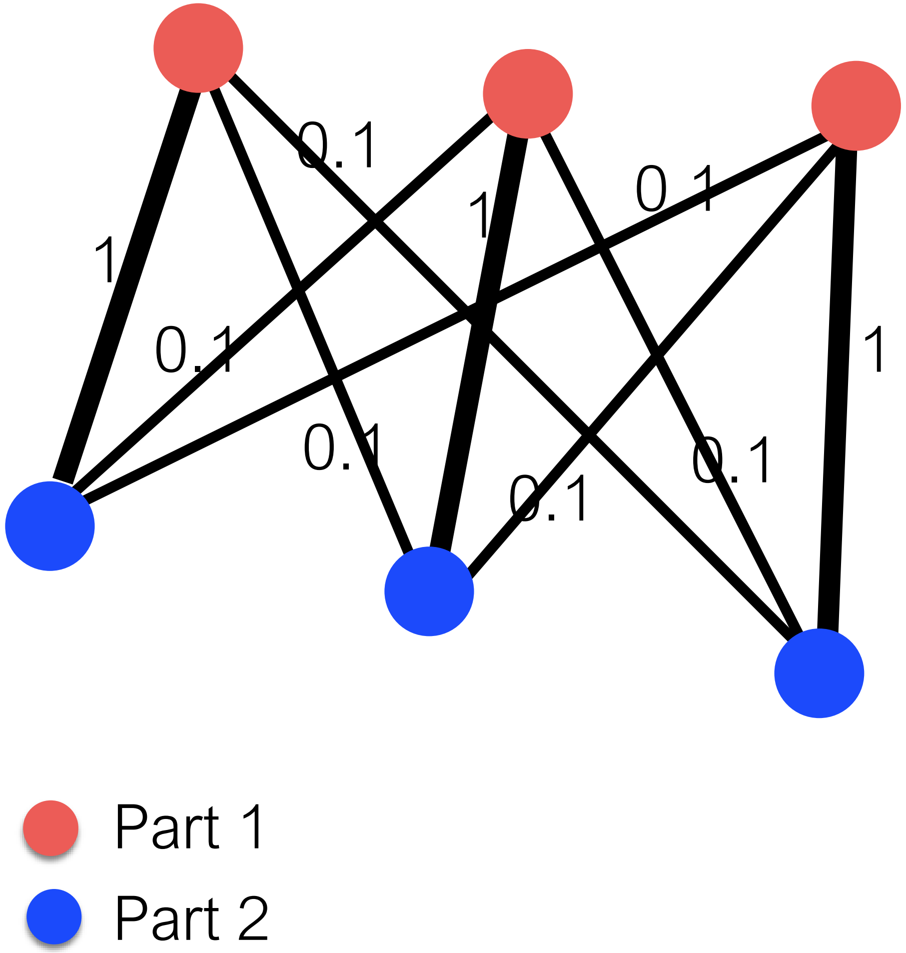


Part-to-Part Association

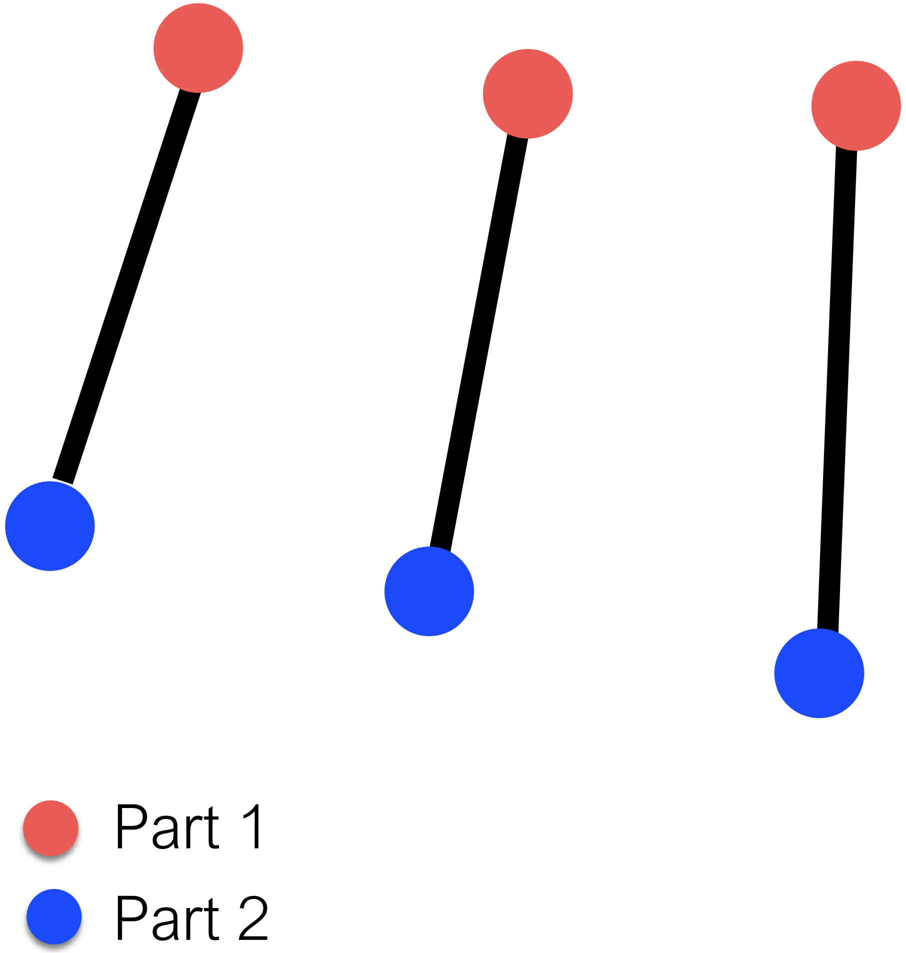


Part 1
Part 2

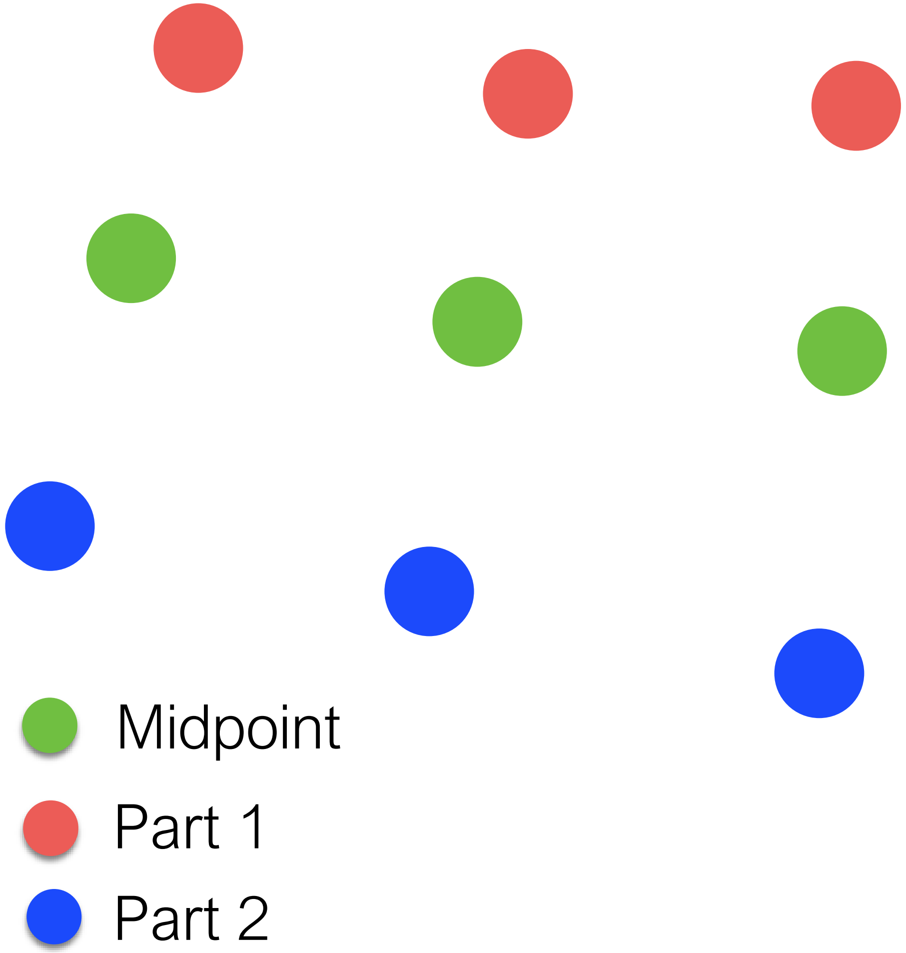
Part-to-Part Association



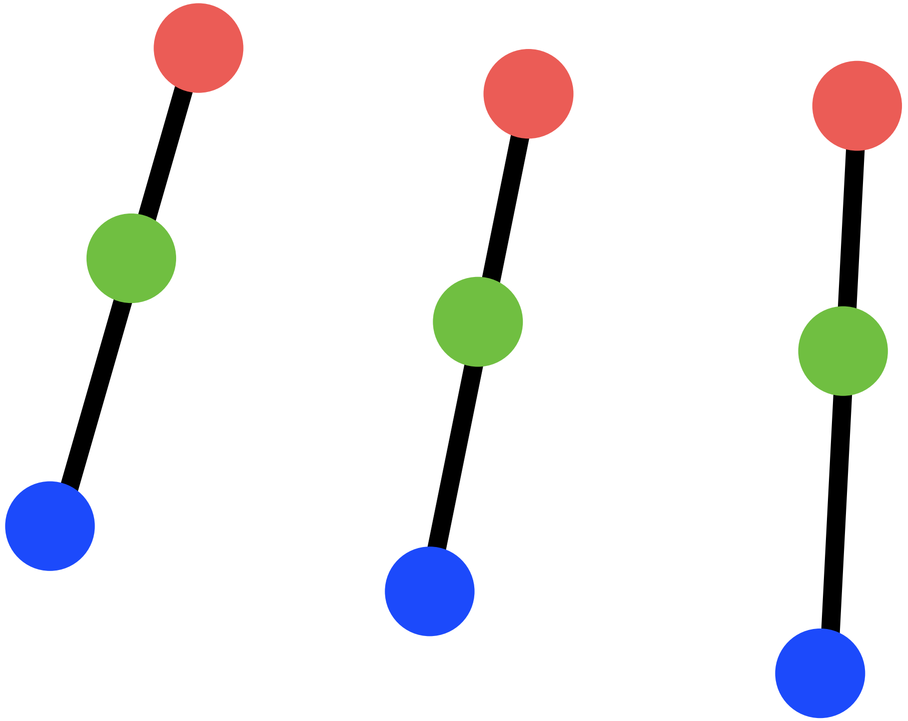
Part-to-Part Association



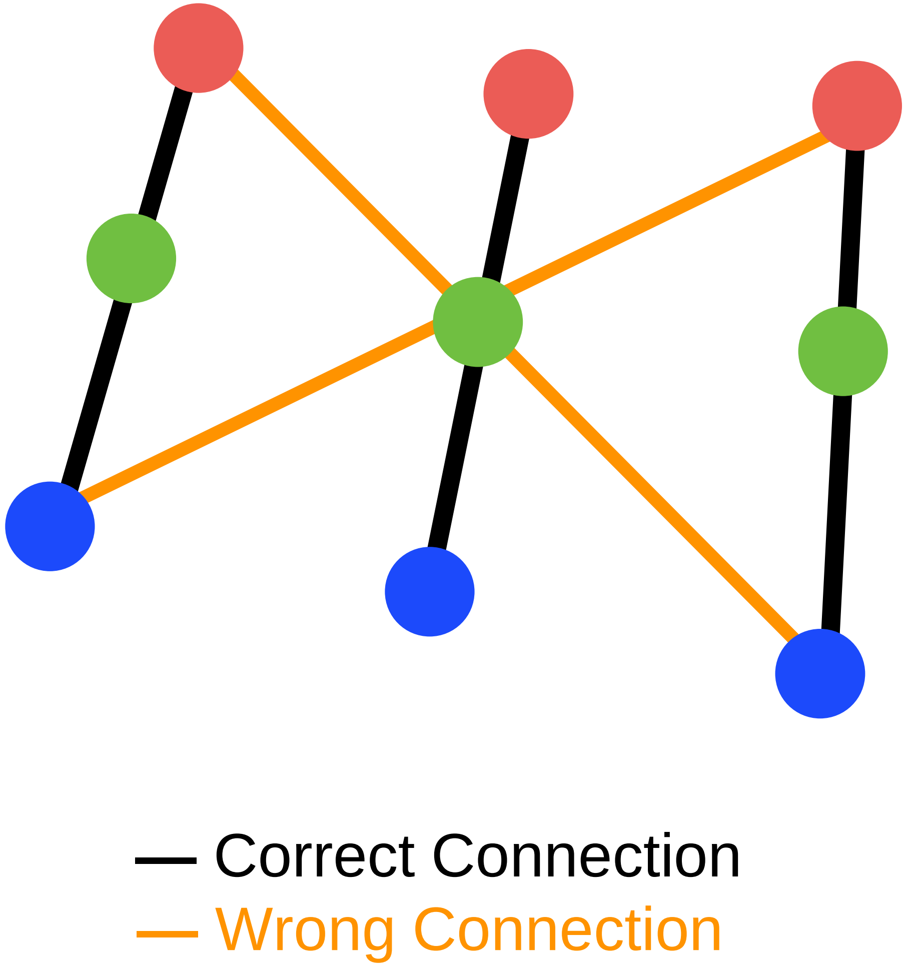
Midpoint Representation for Part-to-Part Association



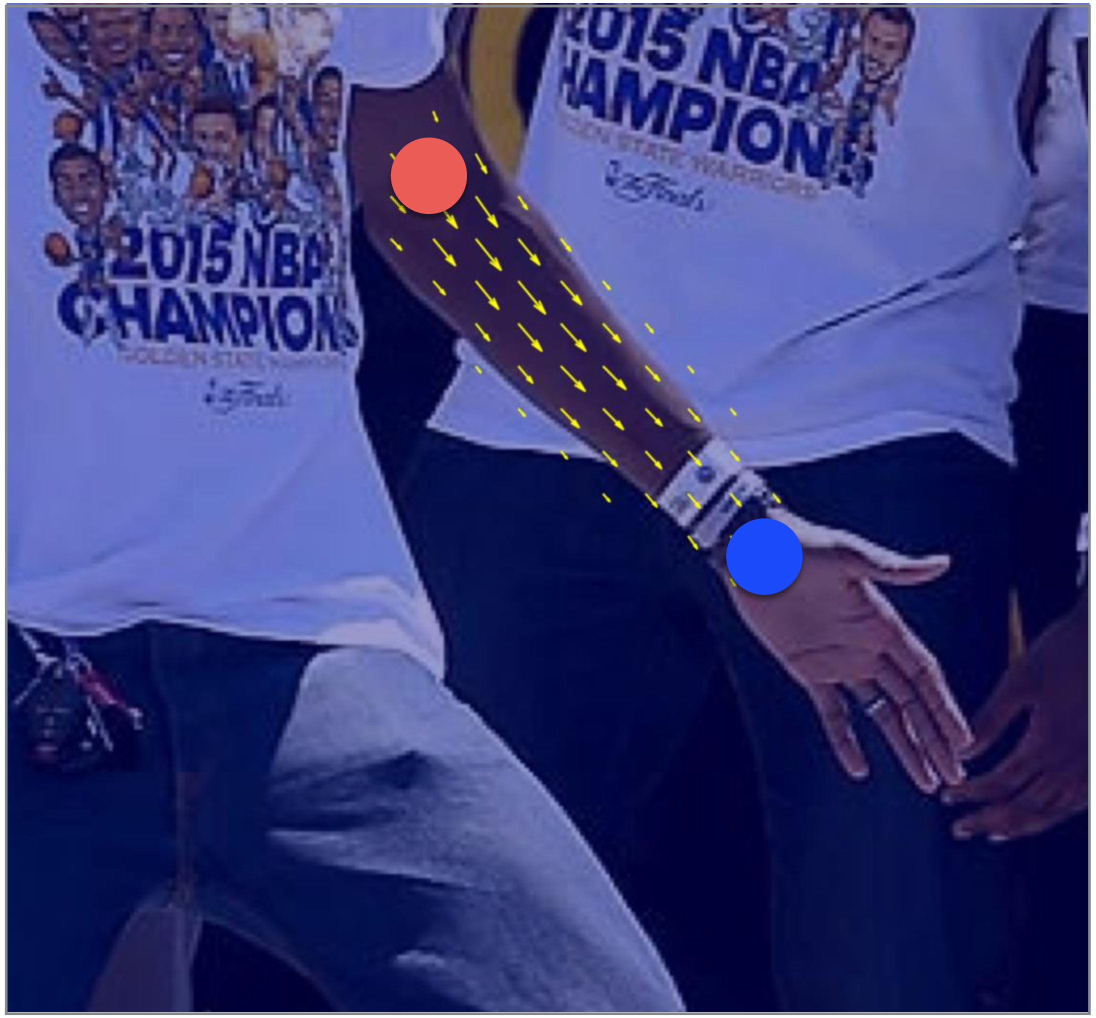
Spatial Ambiguity of the Midpoint Representation

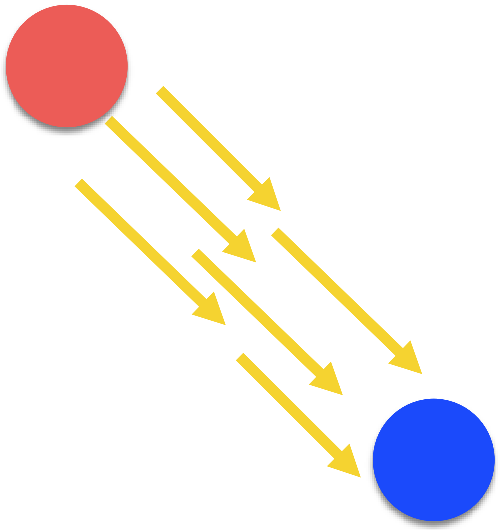


Spatial Ambiguity of the Midpoint Representation



Part Affinity Fields for Part-to-Part Association



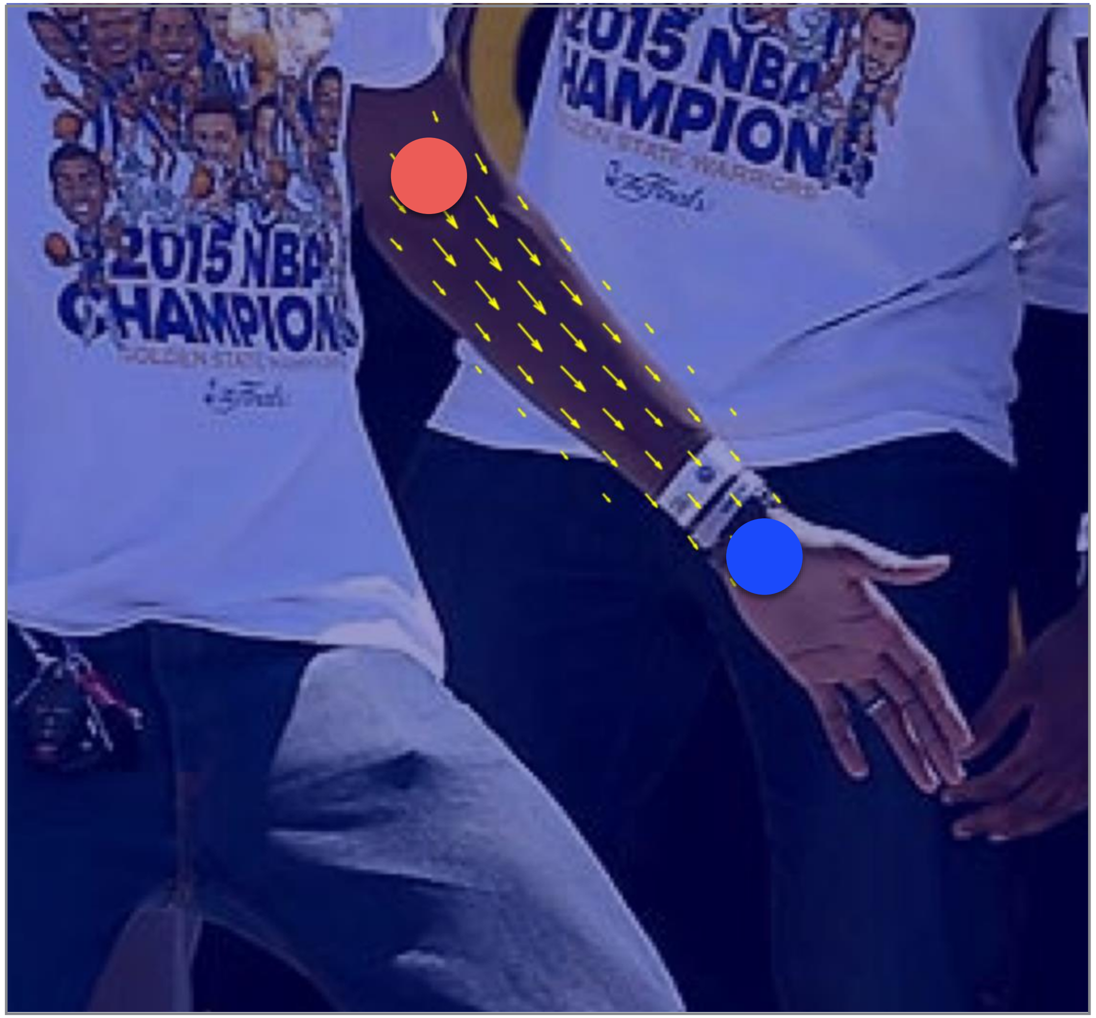


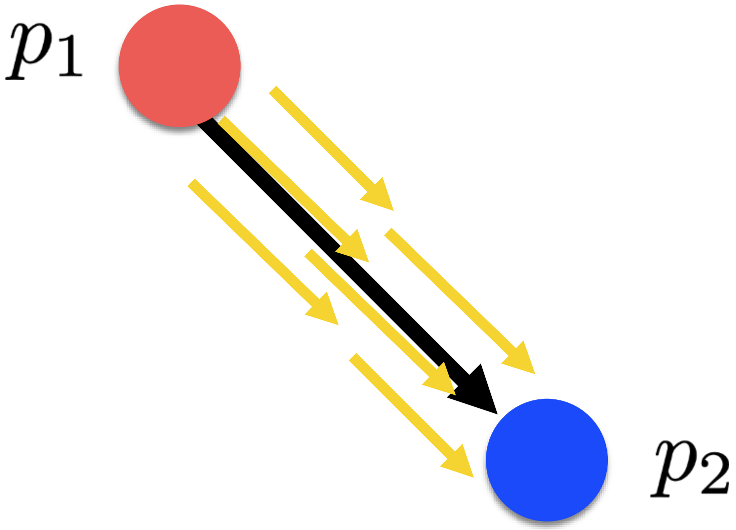
Direction vector in the PAFs

Part 1

Part 2

Part Affinity Fields for Part-to-Part Association

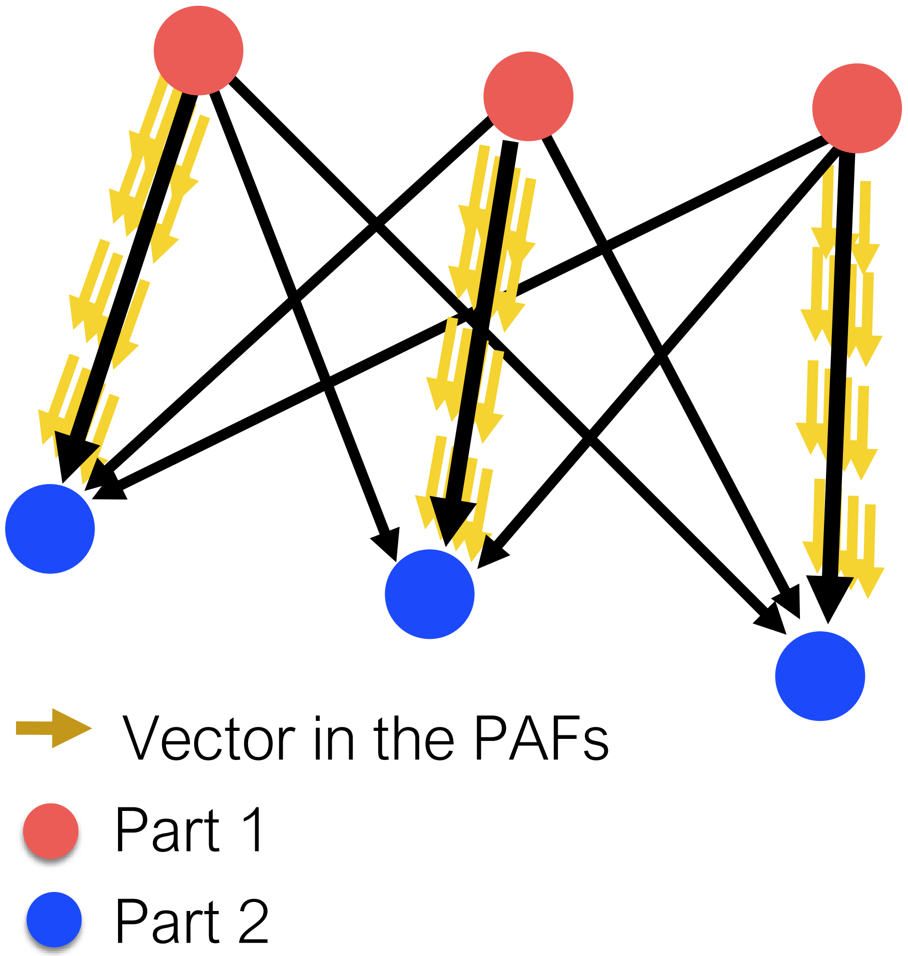




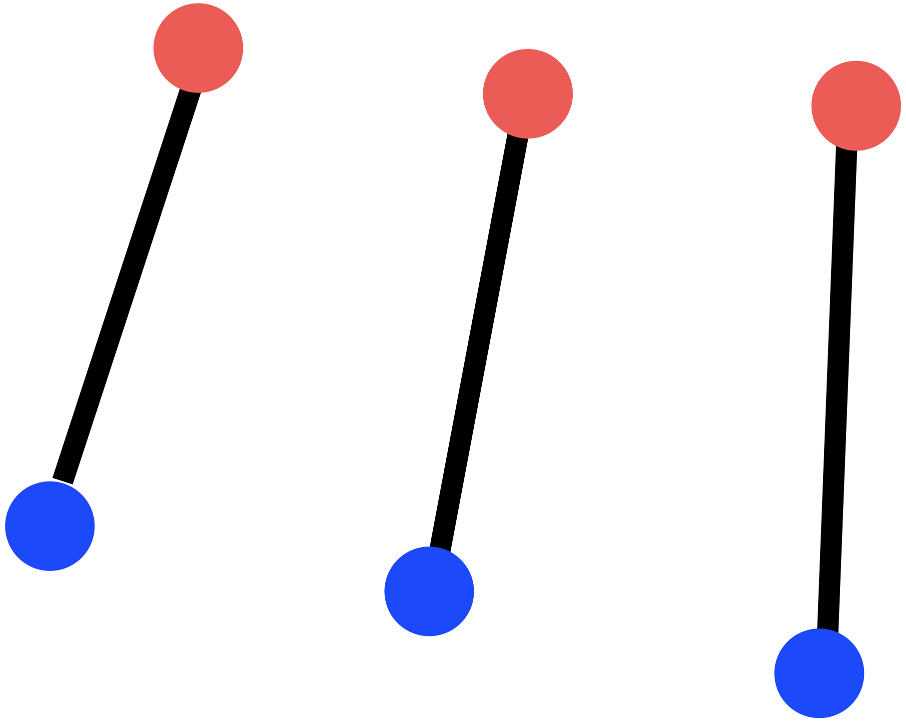
Affinity score between p_1 and p_2

$$= \text{sum}(\vec{\mathbf{v}} \cdot \overrightarrow{p_1 p_2})$$

Part Affinity Fields for Part-to-Part Association

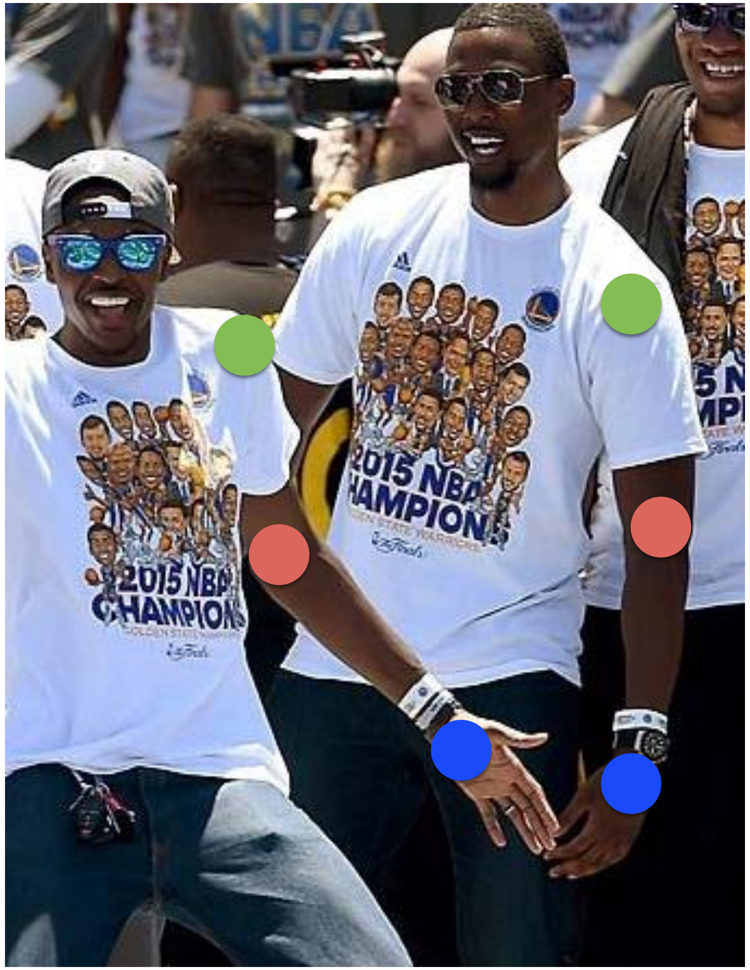


Part Affinity Fields for Part-to-Part Association



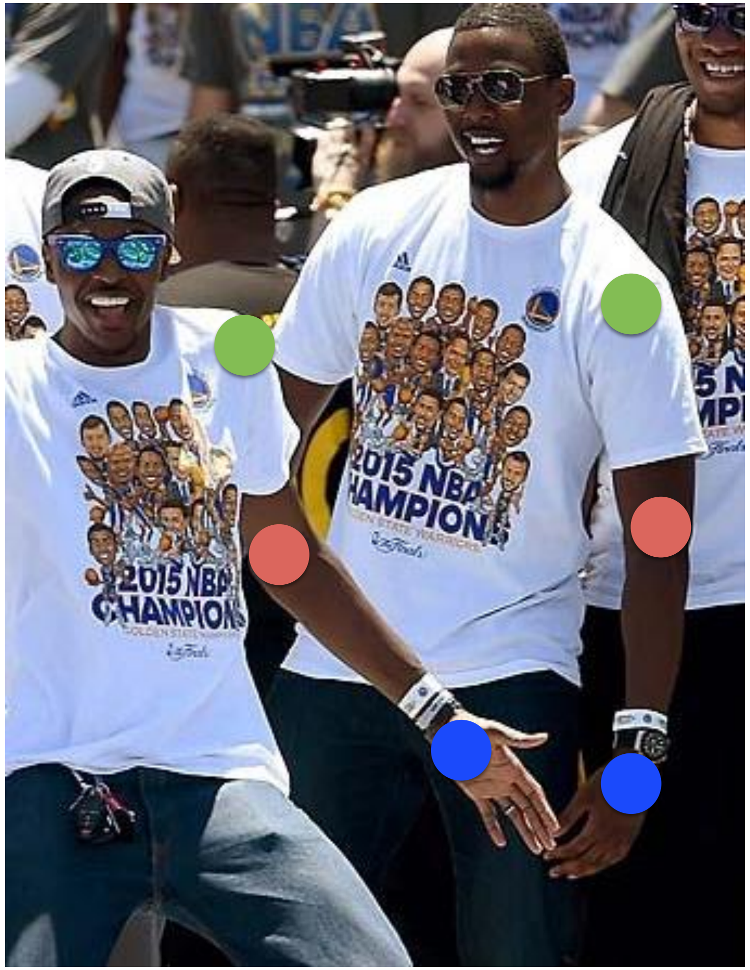
Part Association for Full-body Pose

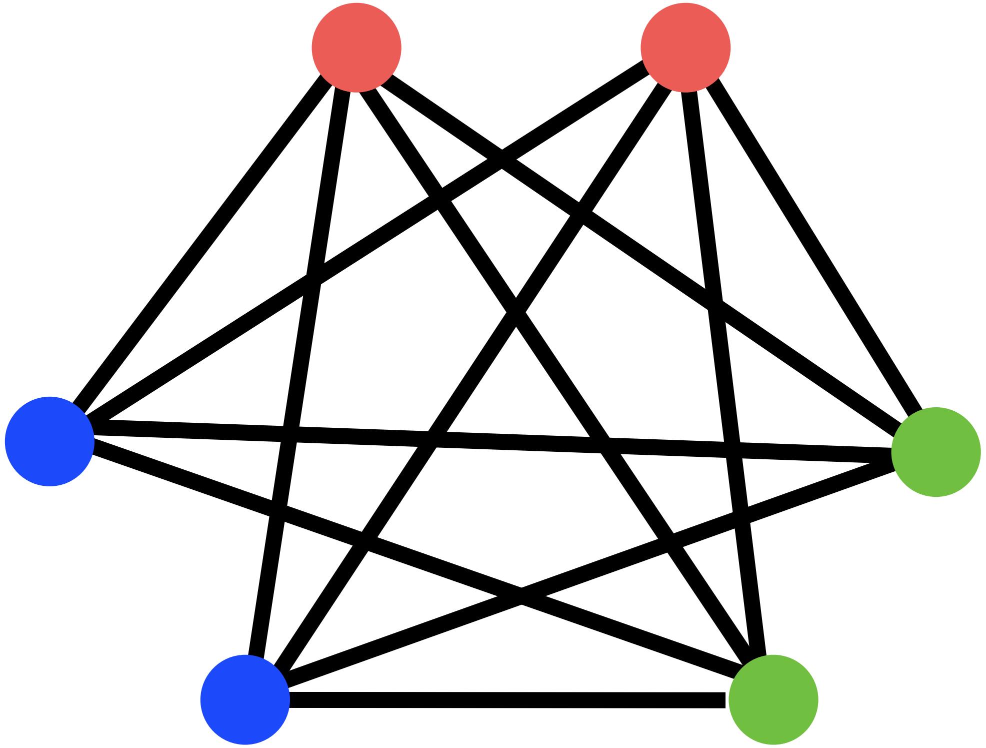
Elbow
Wrist
Shoulder



Part Association for Full-body Pose

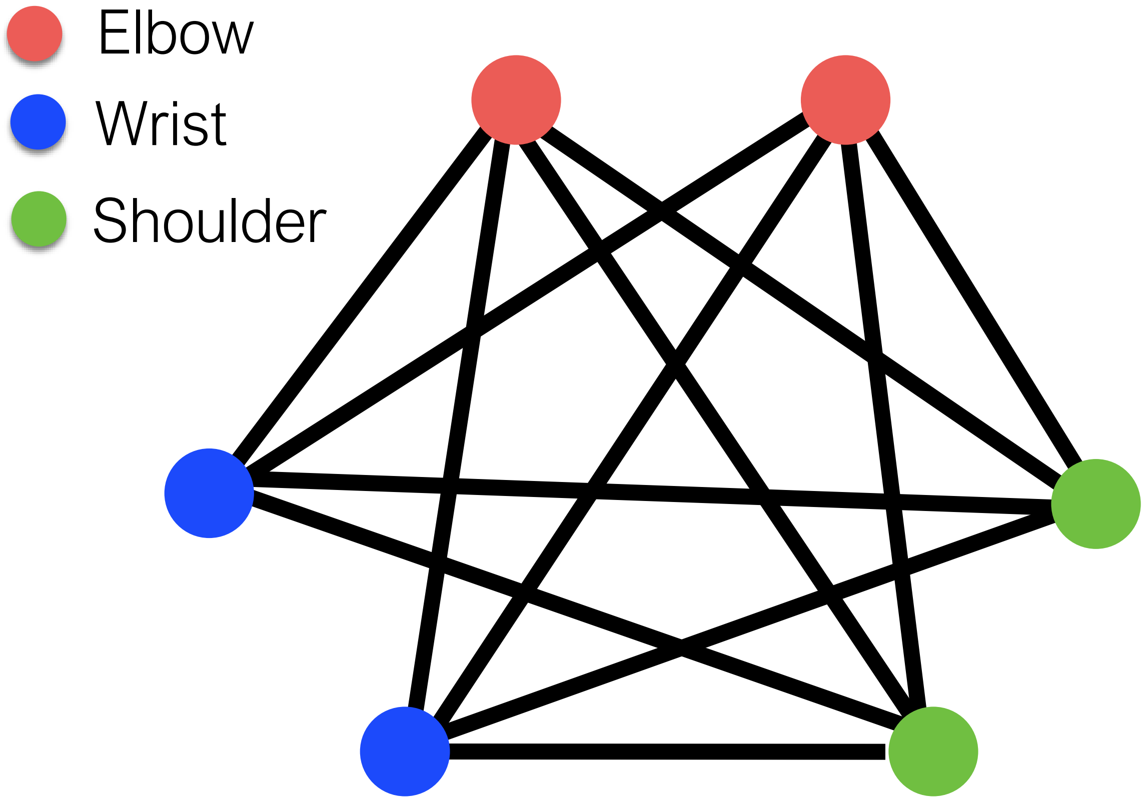
Elbow
Wrist
Shoulder





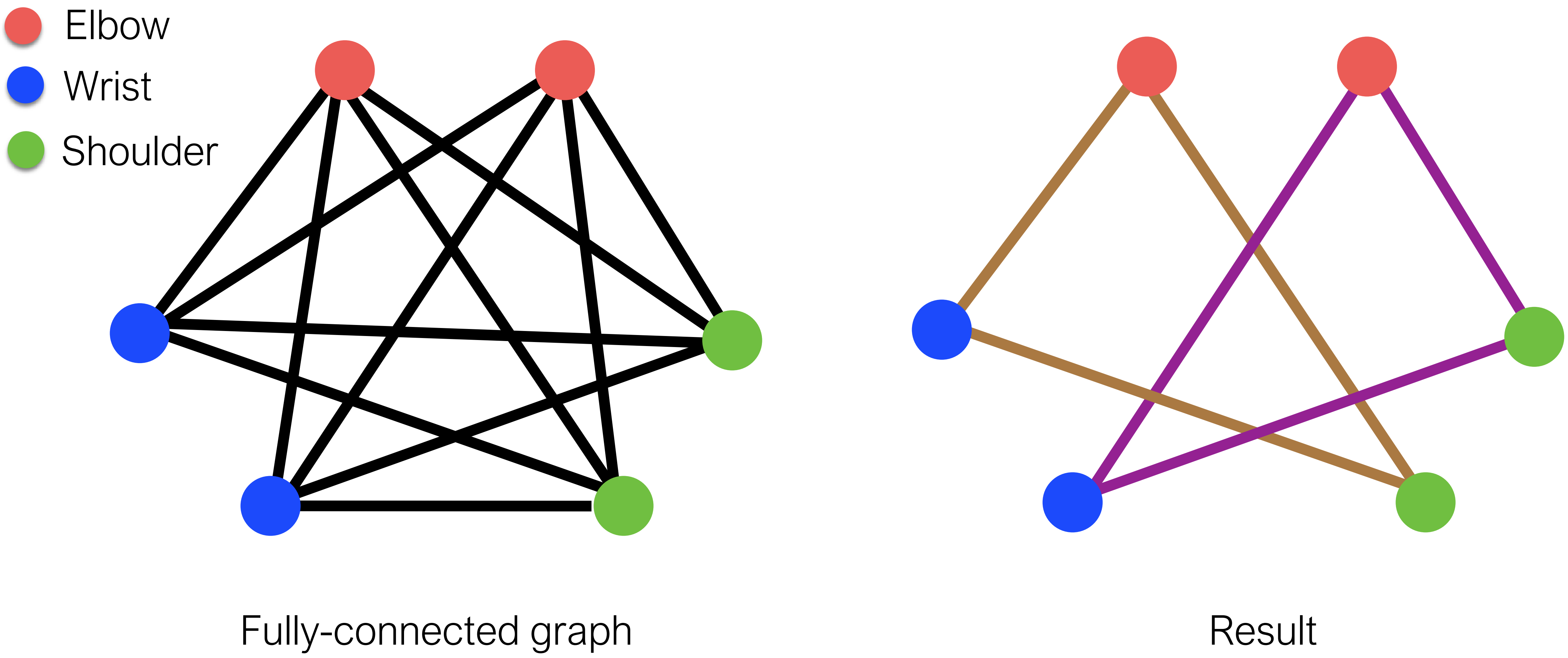
Fully-connected graph

Part Association for Full-body Pose

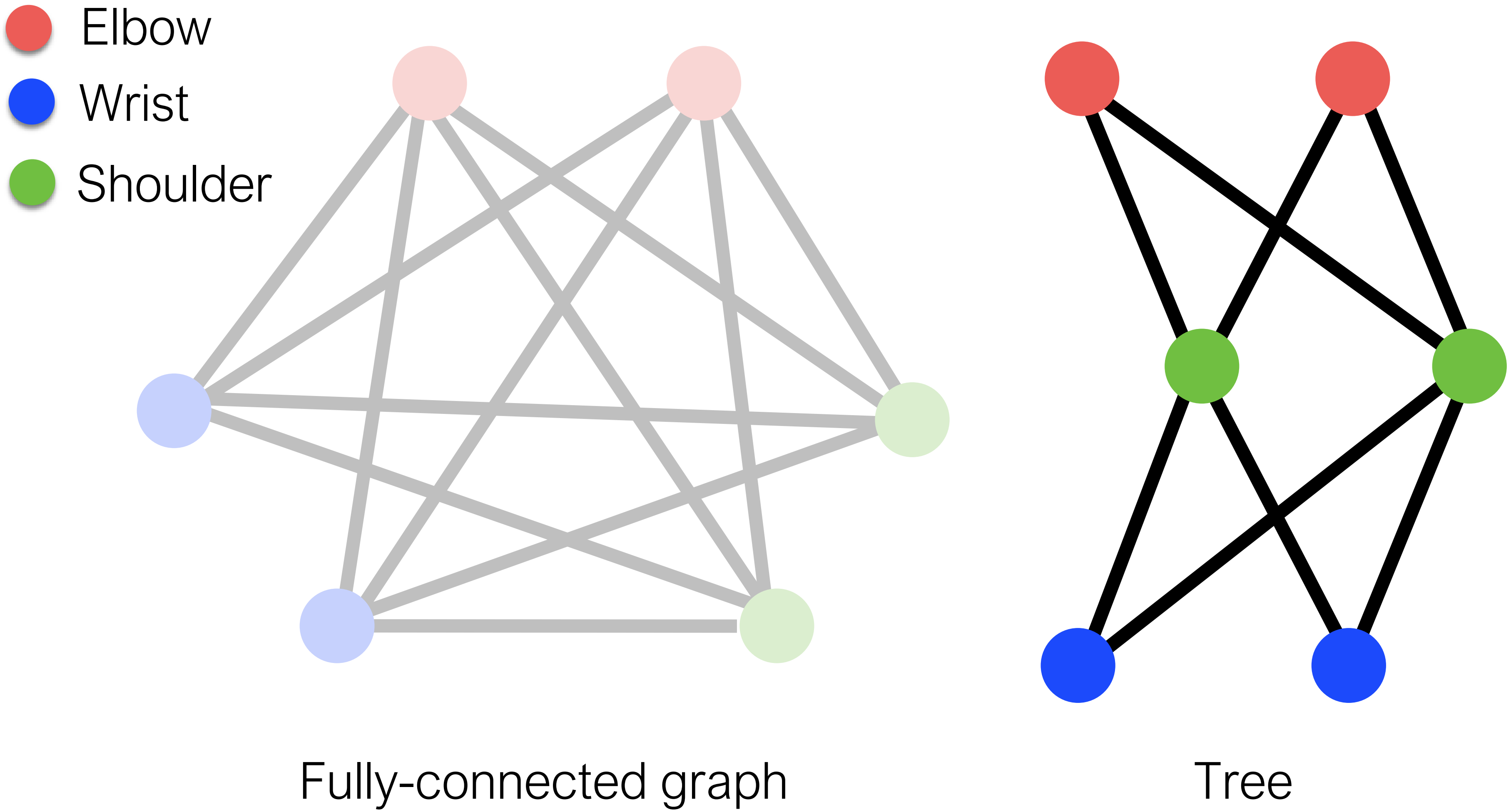


Fully-connected graph

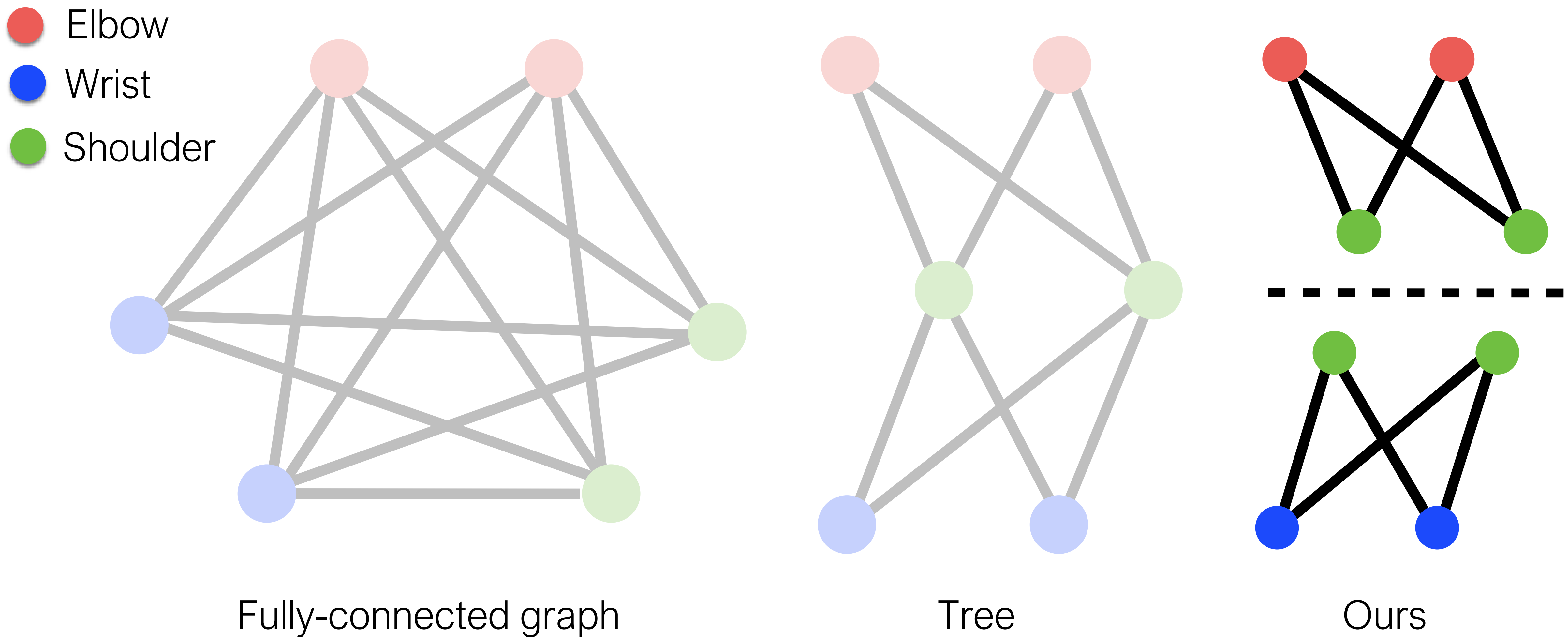
Part Association for Full-body Pose



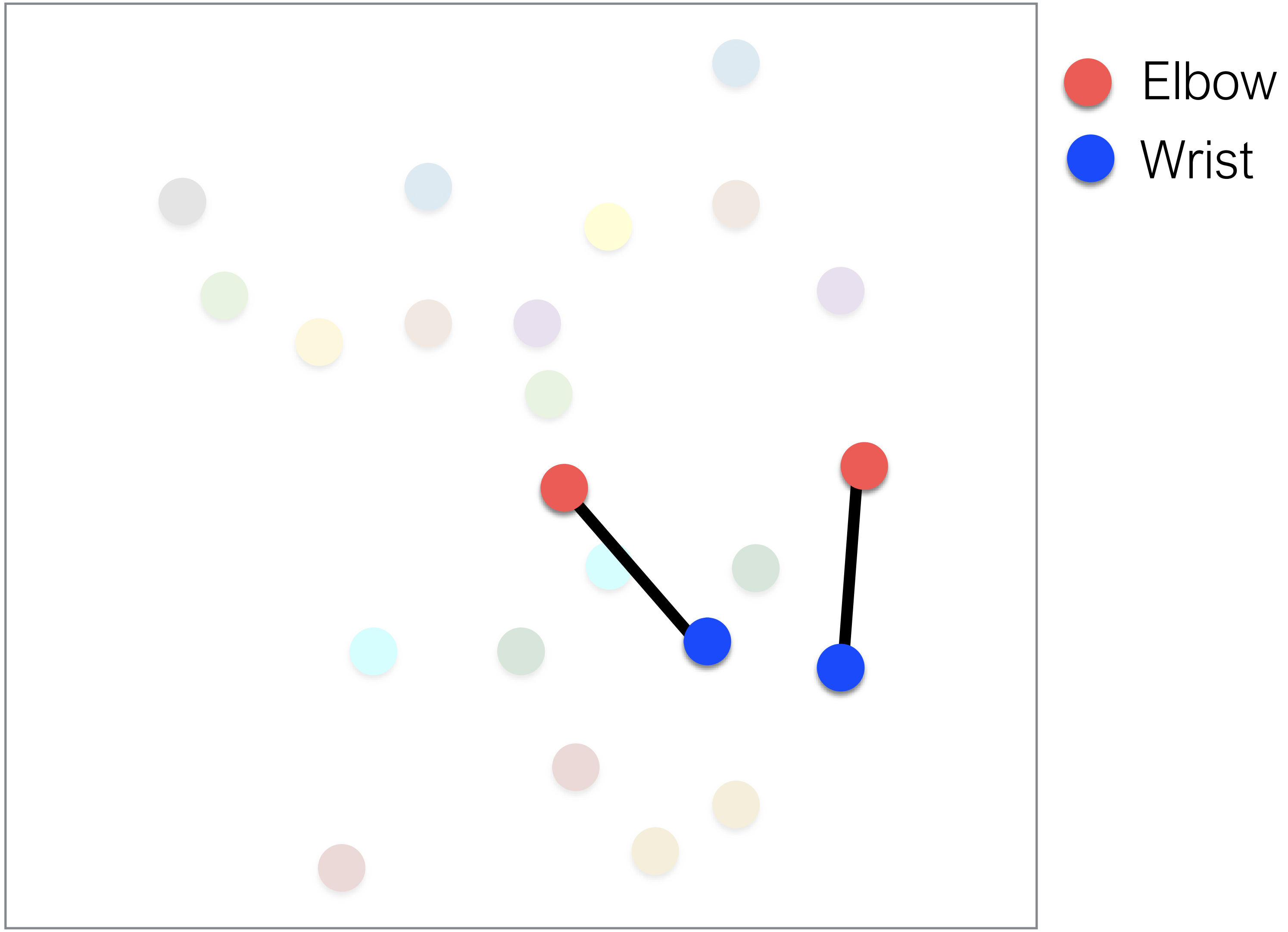
Part Association for Full-body Pose



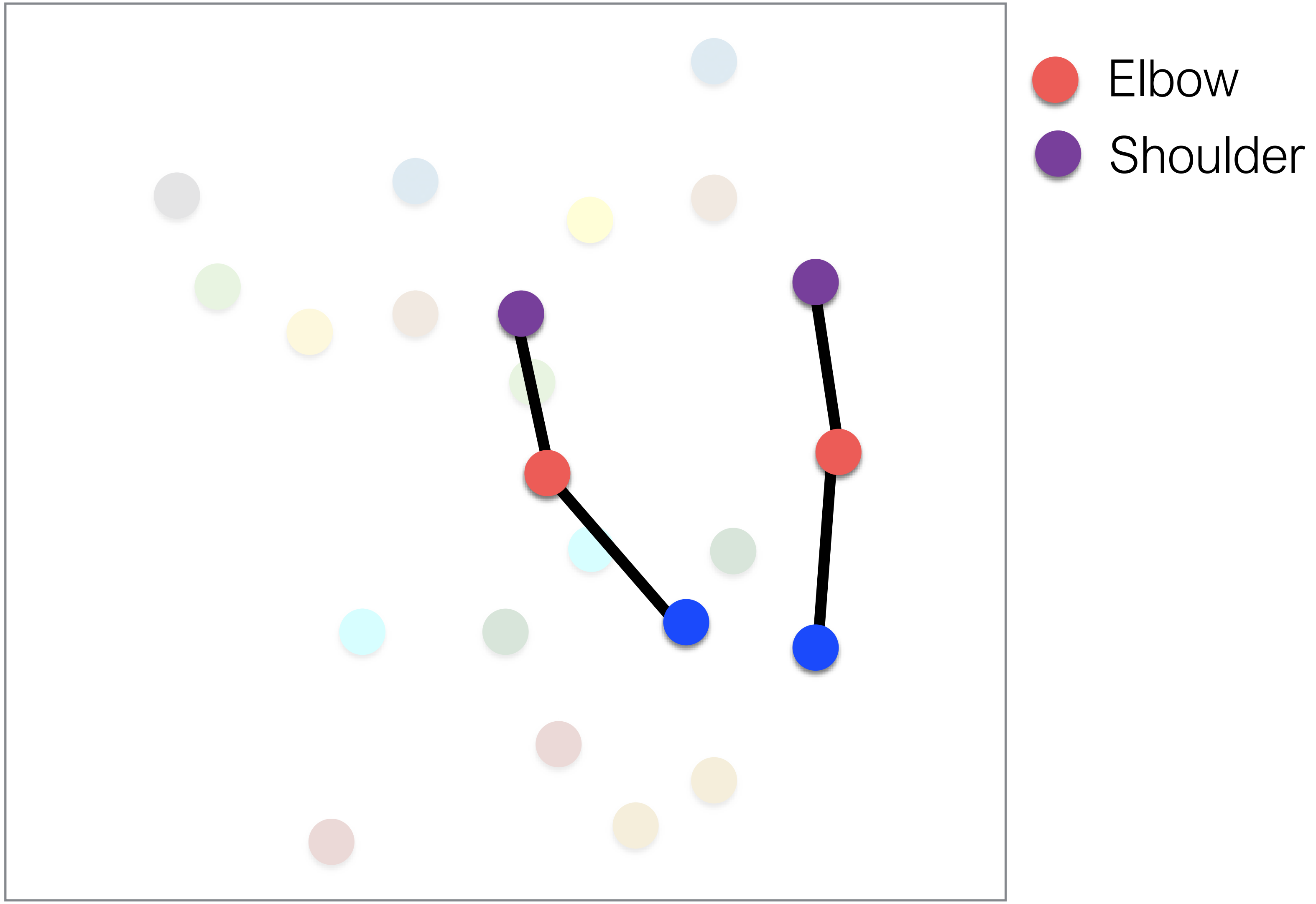
Part Association for Full-body Pose



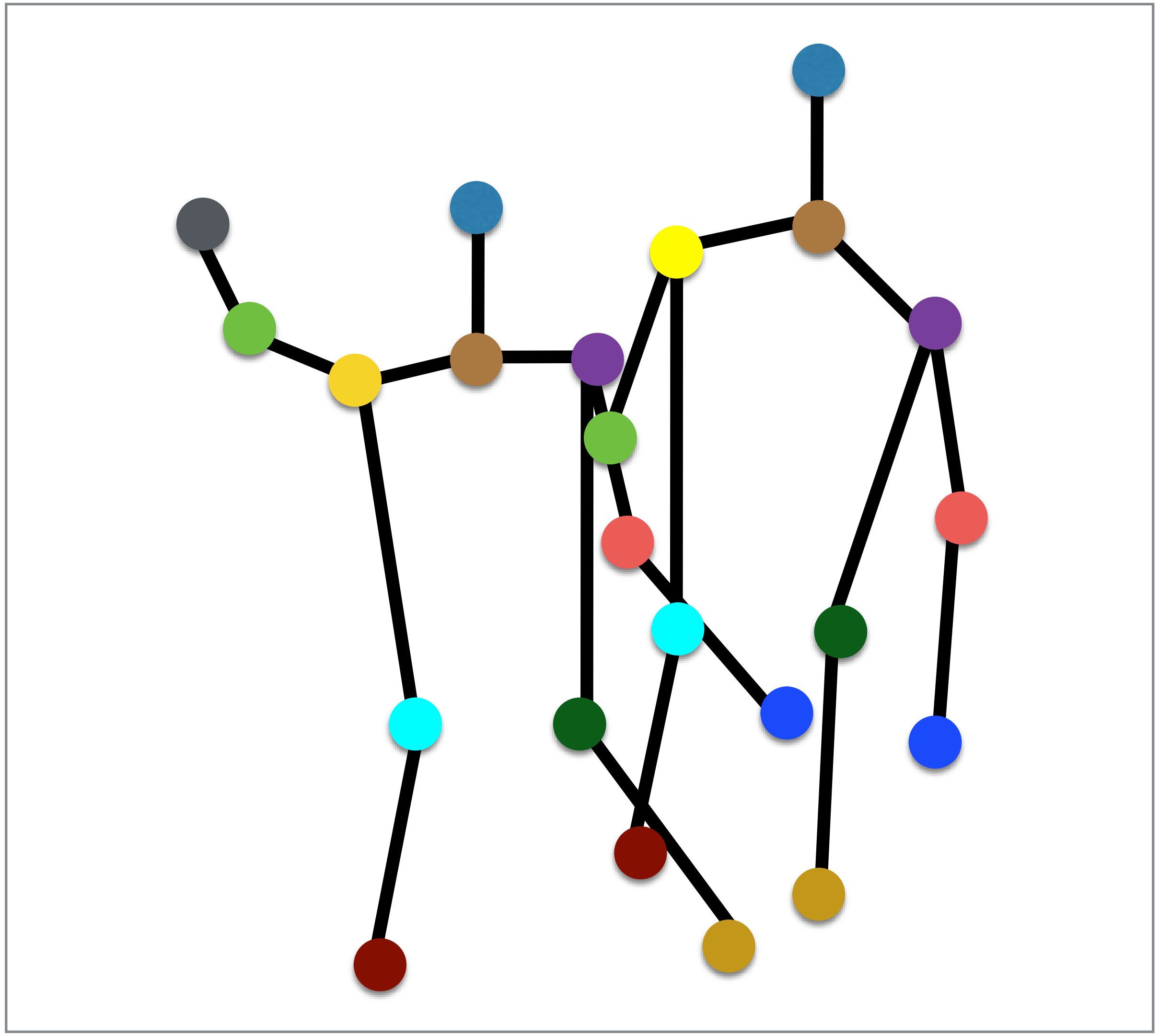
Greedy Algorithm for Body Parts Association



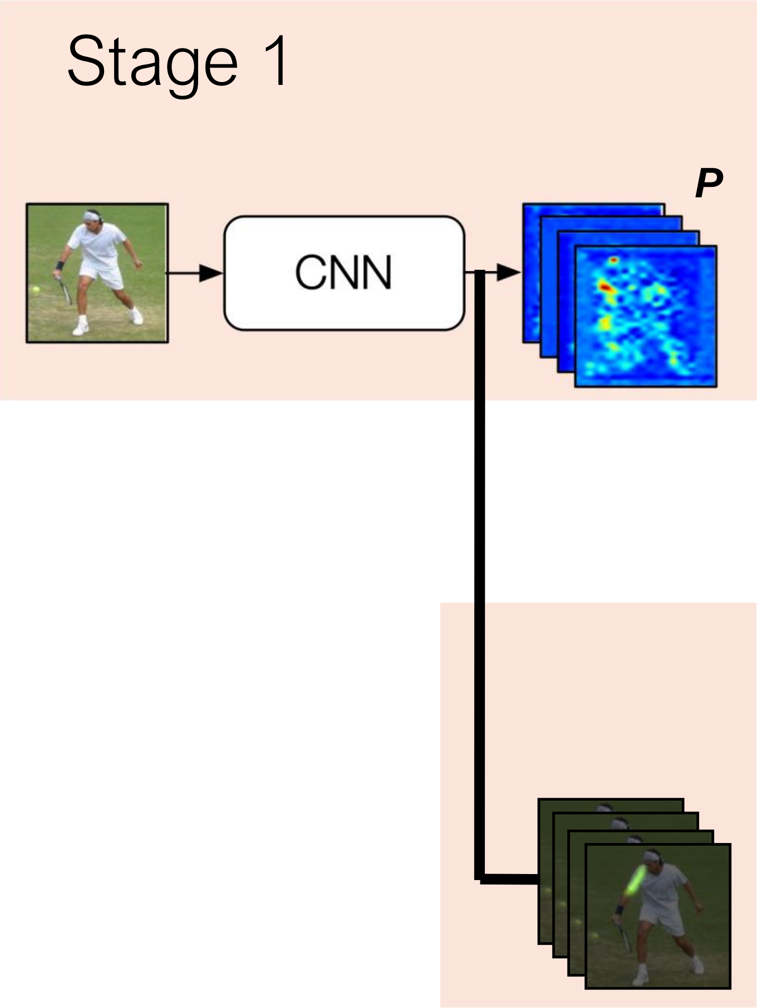
Greedy Algorithm for Body Parts Association



Greedy Algorithm for Body Parts Association



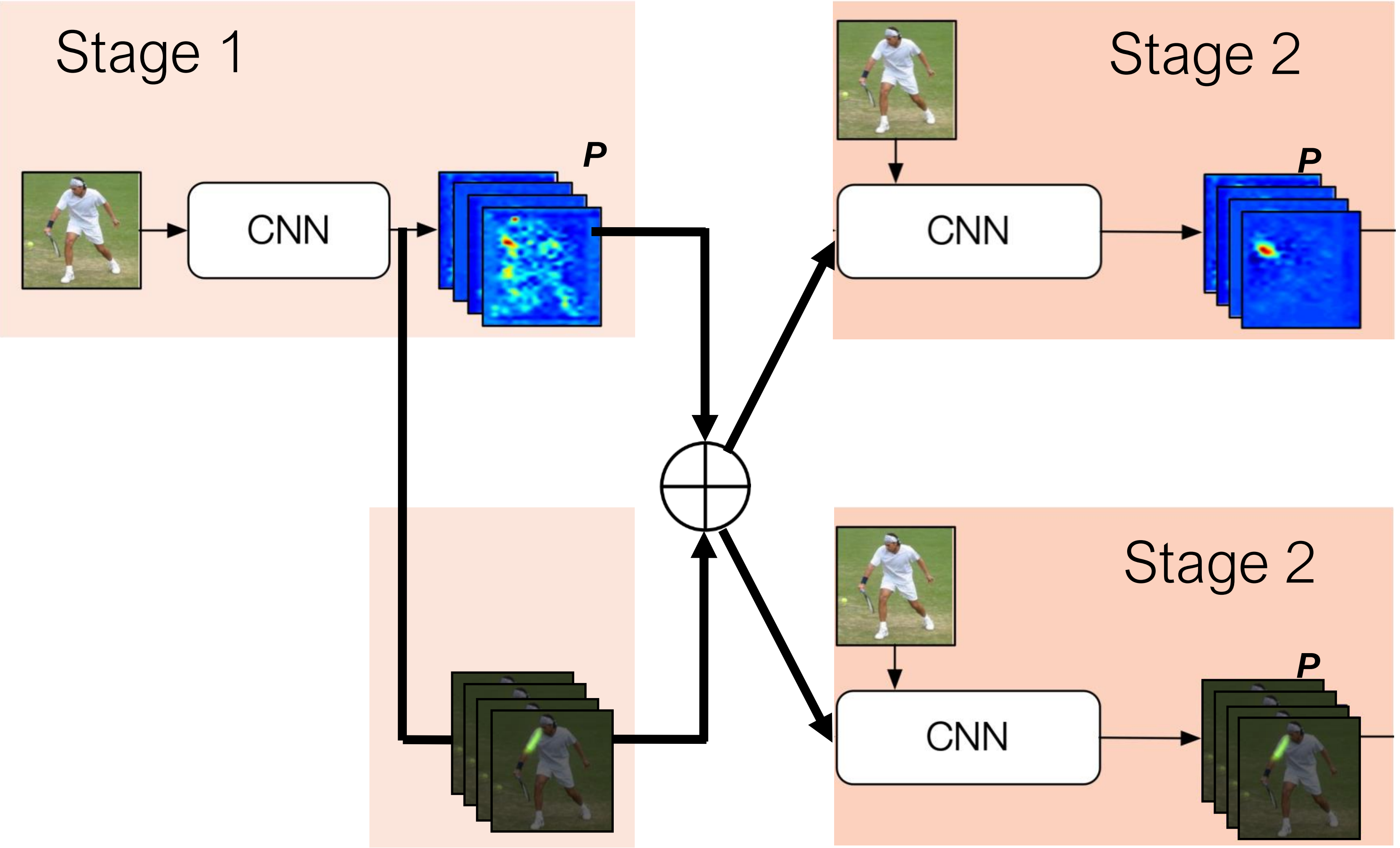
Jointly Learning Parts Detection and Parts Association



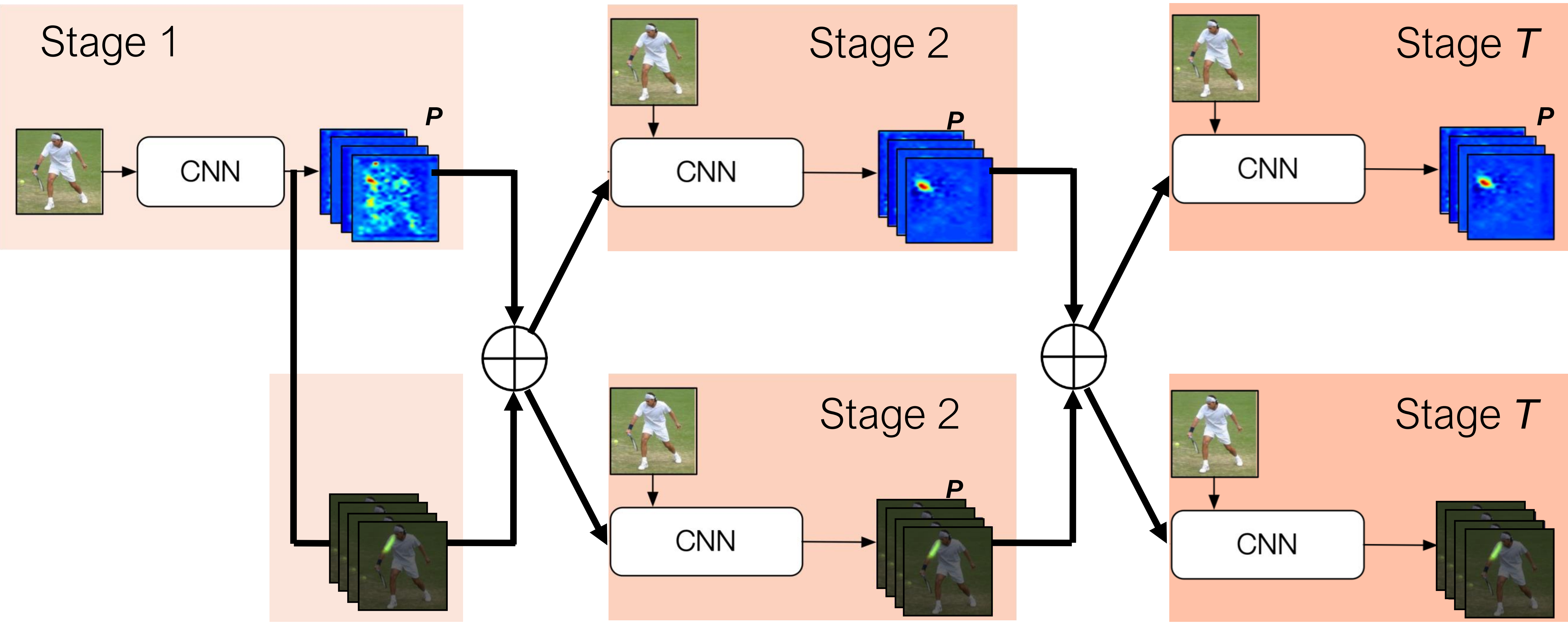
1st branch
part heatmaps

2nd branch
part affinity fields

Jointly Learning Parts Detection and Parts Association



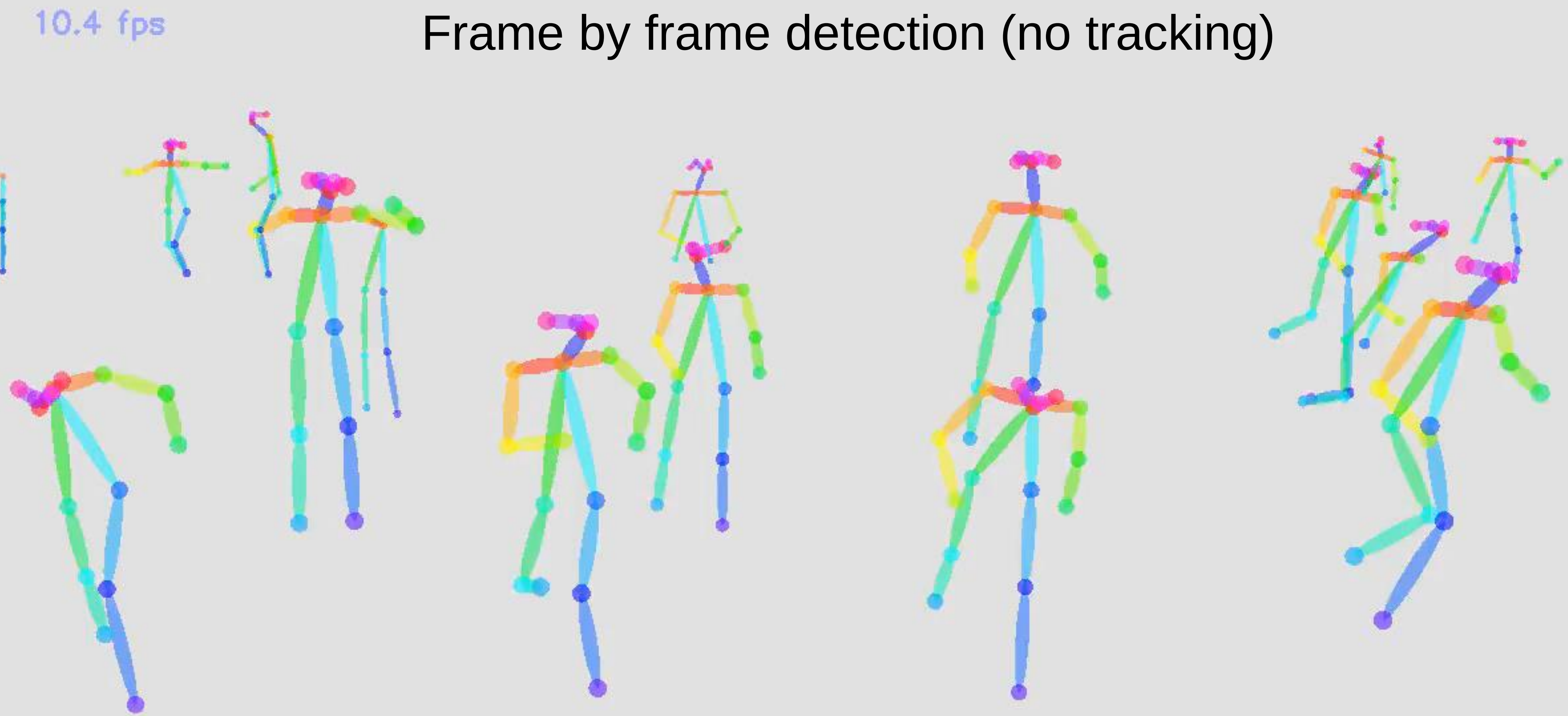
Jointly Learning Parts Detection and Parts Association




Bkg

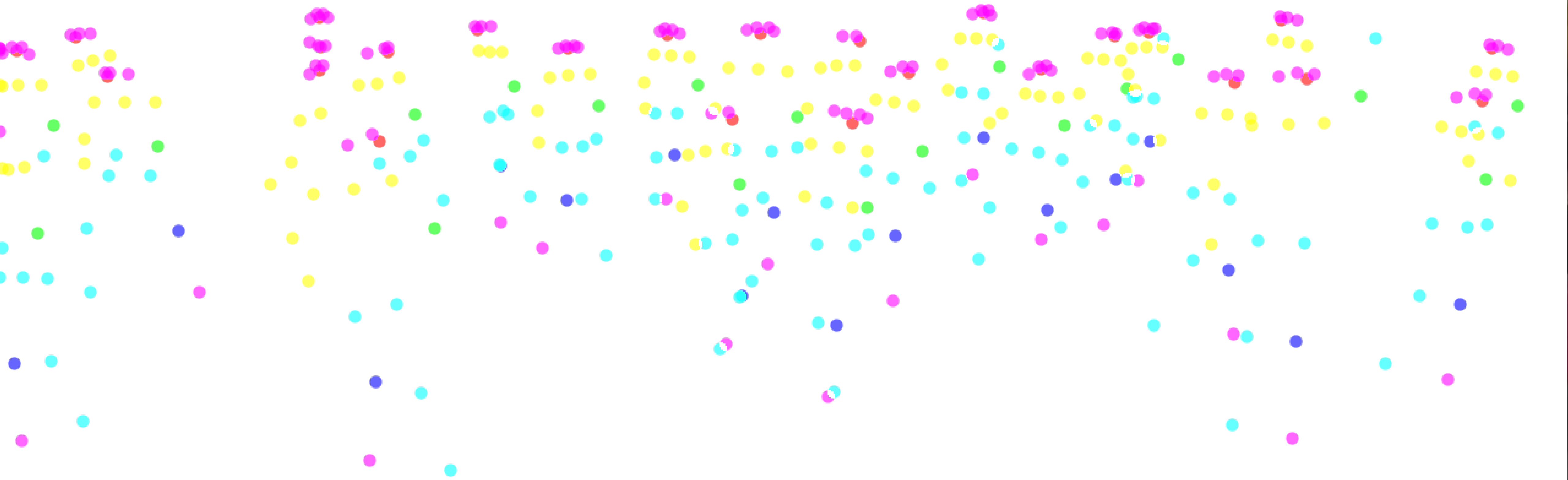

PAFs

Frame by frame detection (no tracking)



Source: https://www.youtube.com/watch?v=2DiQUX11YaY

Major Contribution: Part Affinity Fields for Parts Association



PAFs: an **efficient** representation is **discriminative** enough that a greedy parse is sufficient to produce high-quality results in realtime.

Intermission

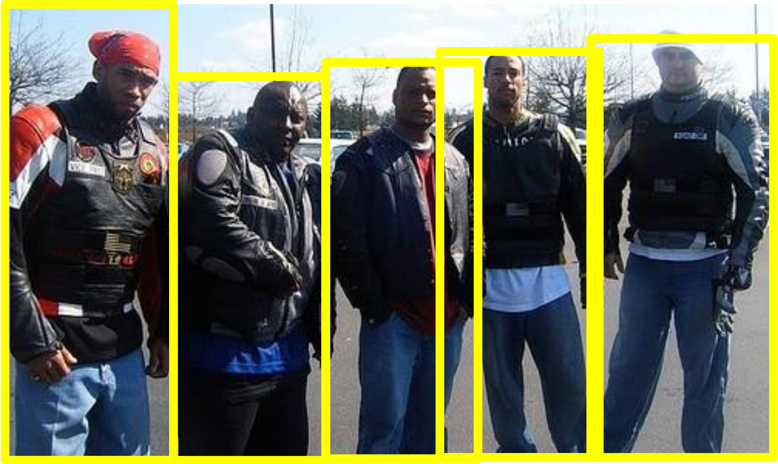
Mask R-CNN

ICCV 2017

Kaiming He

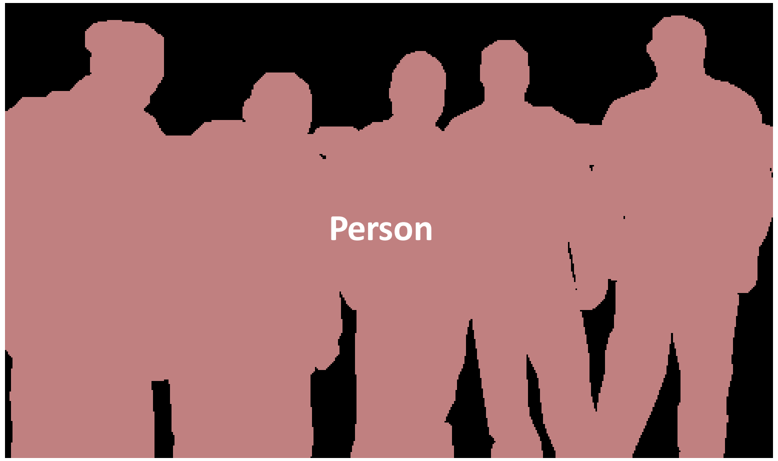
Georgia Gkioxari, Piotr Dollár, and Ross Girshick Facebook AI Research (FAIR)

Visual Perception Problems



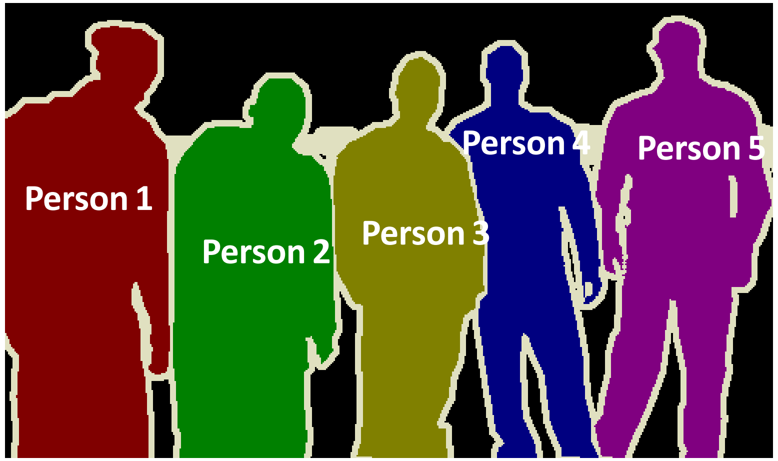
Object Detection

✓



Semantic Segmentation

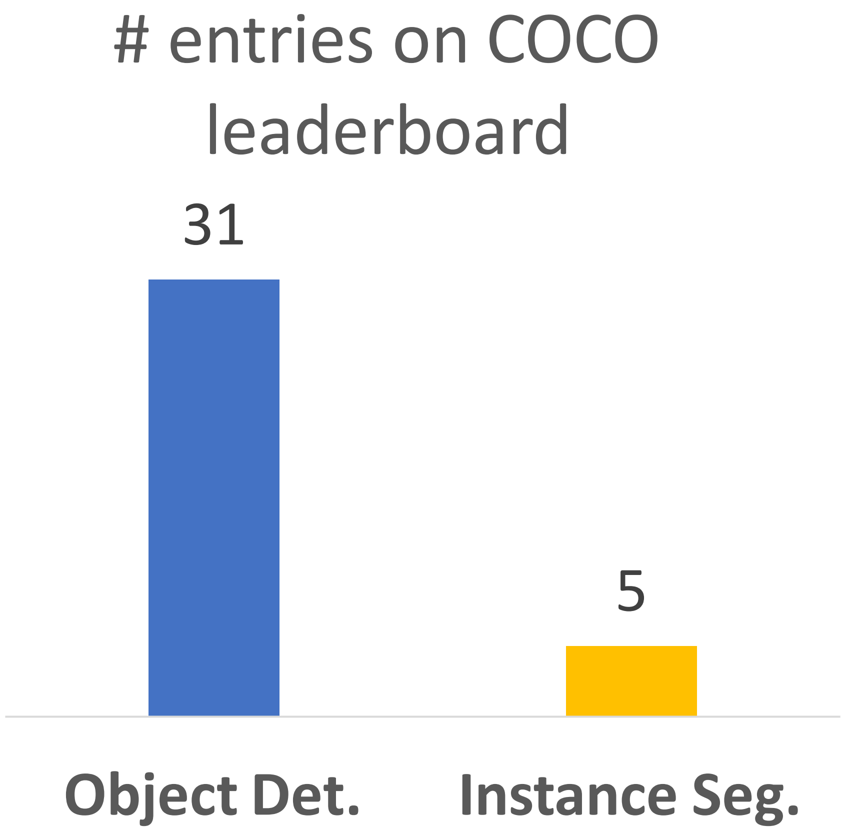
✓

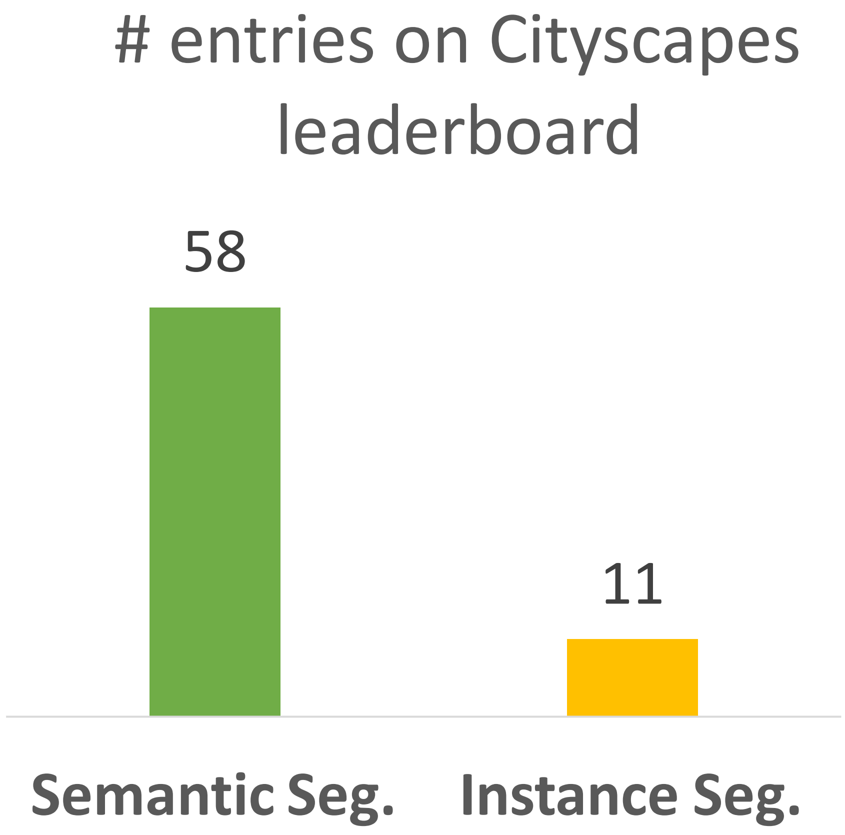


Instance Segmentation

?

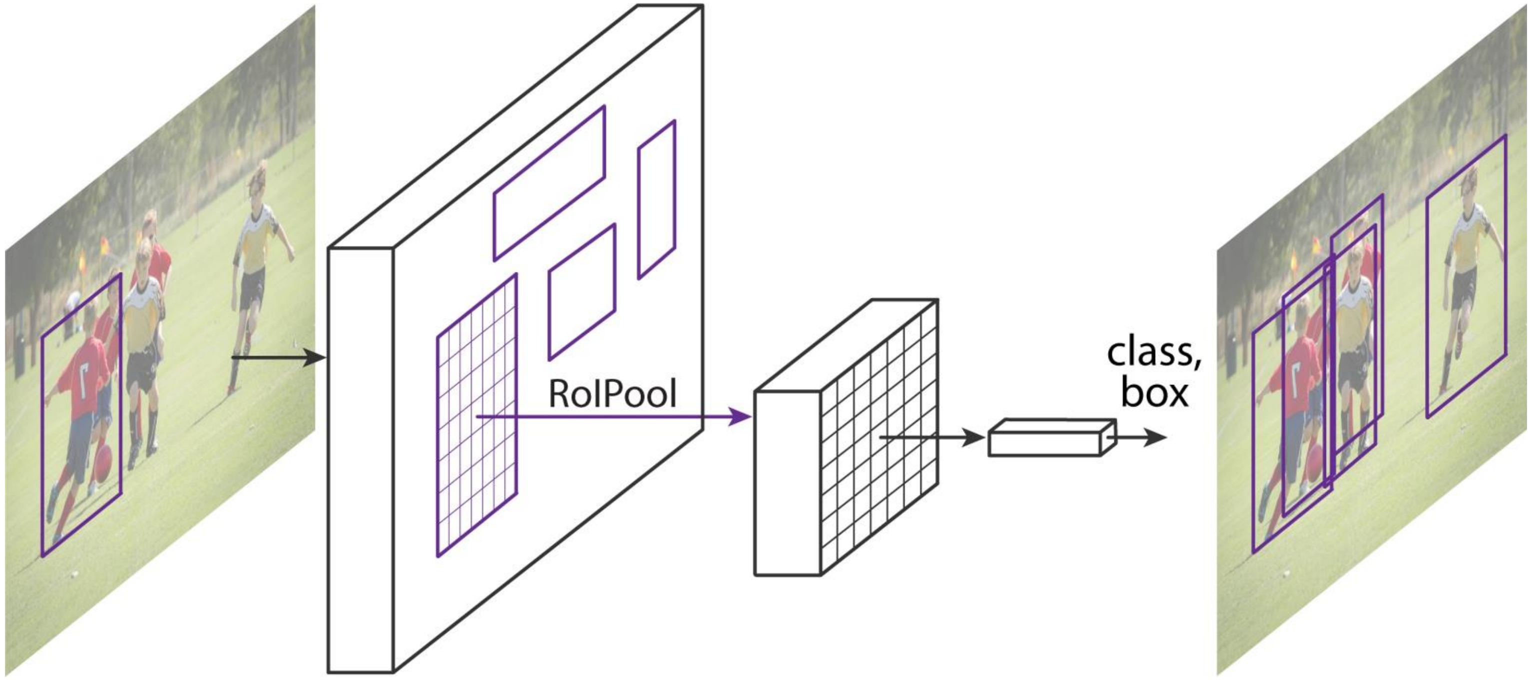
A Challenging Problem...





Object Detection

- Fast/Faster R-CNN
  - ✓Good speed
  - ✓Good accuracy
  - ✓Intuitive
  - ✓Easy to use



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.

Semantic Segmentation

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

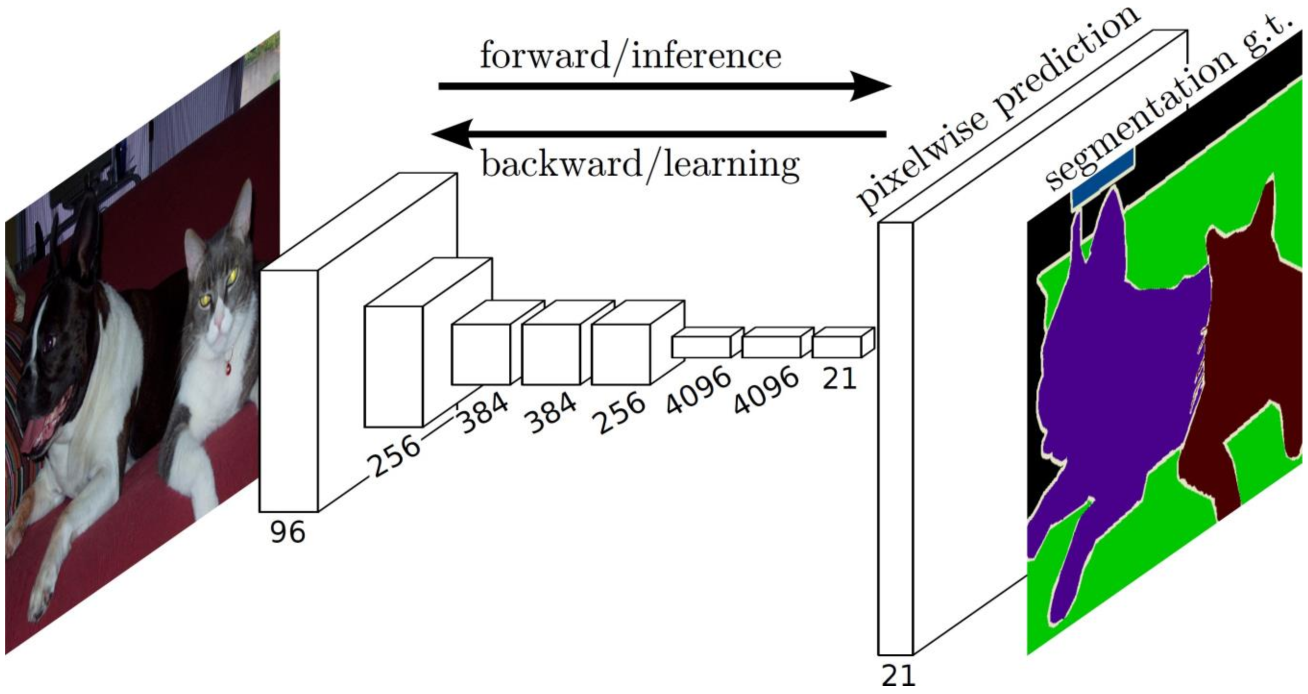
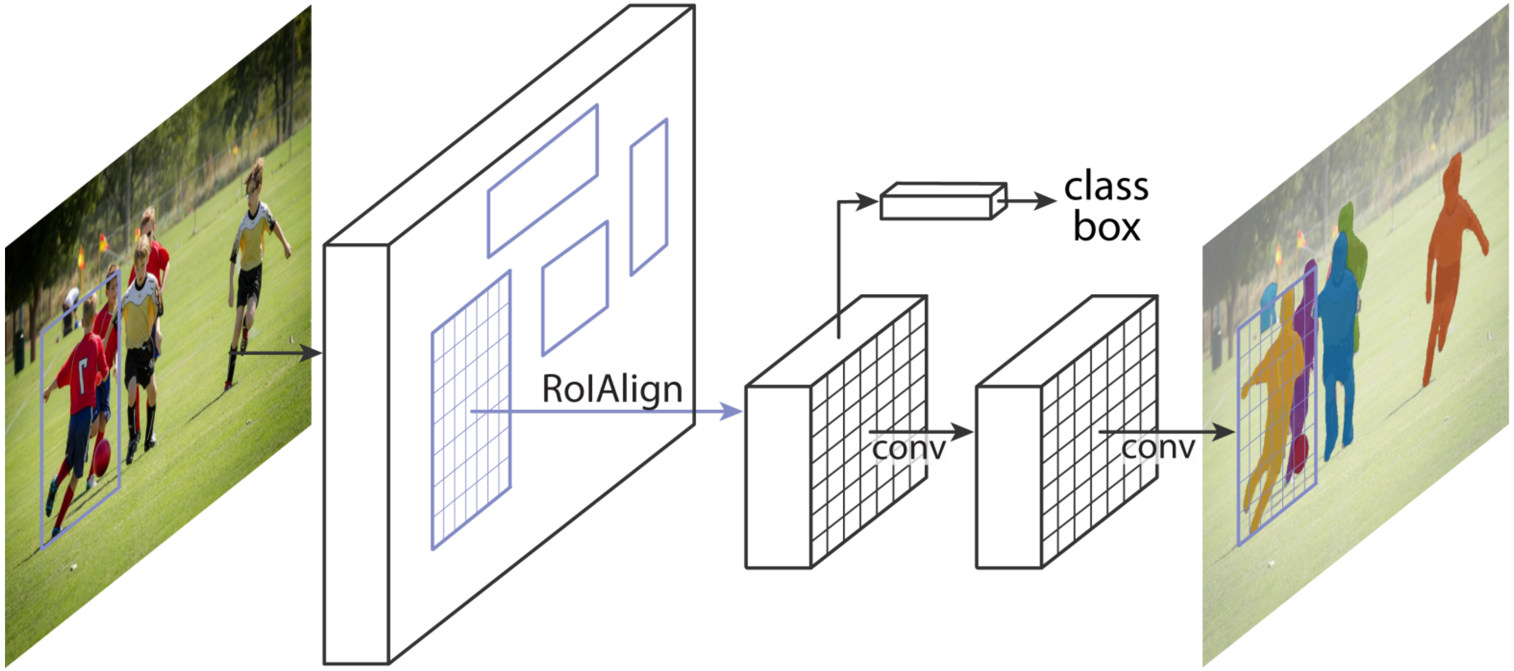


Figure credit: Long et al

Jonathan Long, Evan Shelhamer, & Trevor Darrell. “Fully Convolutional Networks for Semantic Segmentation”. CVPR 2015.

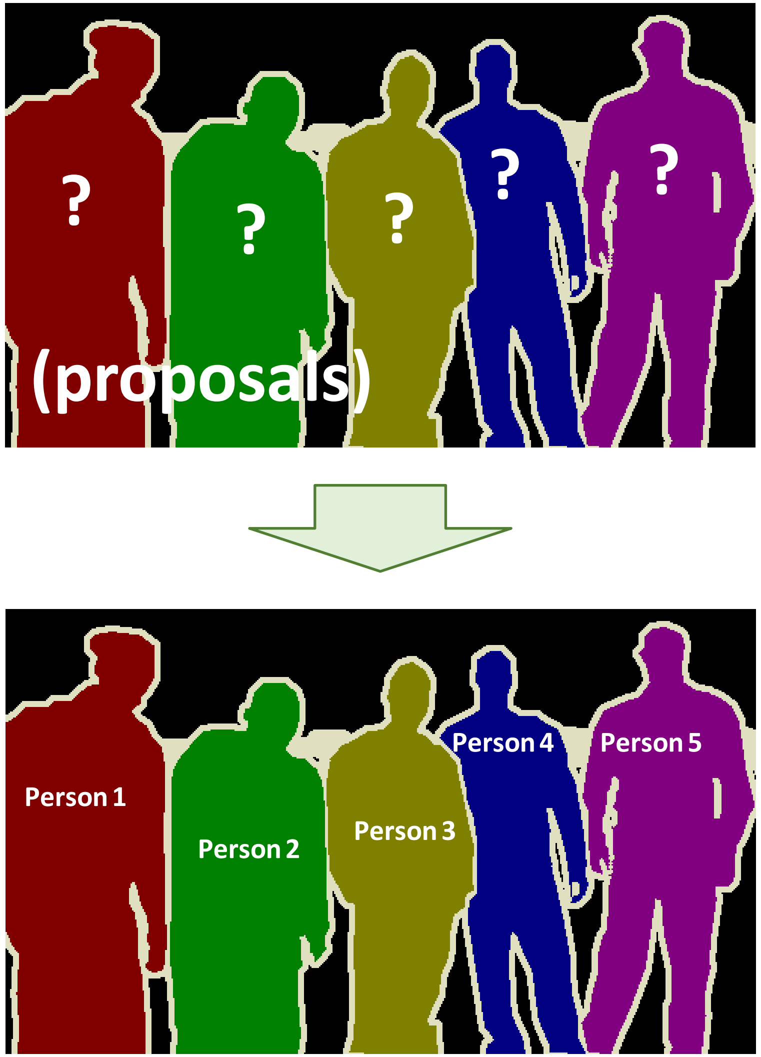
Instance Segmentation

- **Goals** of Mask R-CNN
  - ✓Good speed
  - ✓Good accuracy
  - ✓Intuitive
  - ✓Easy to use

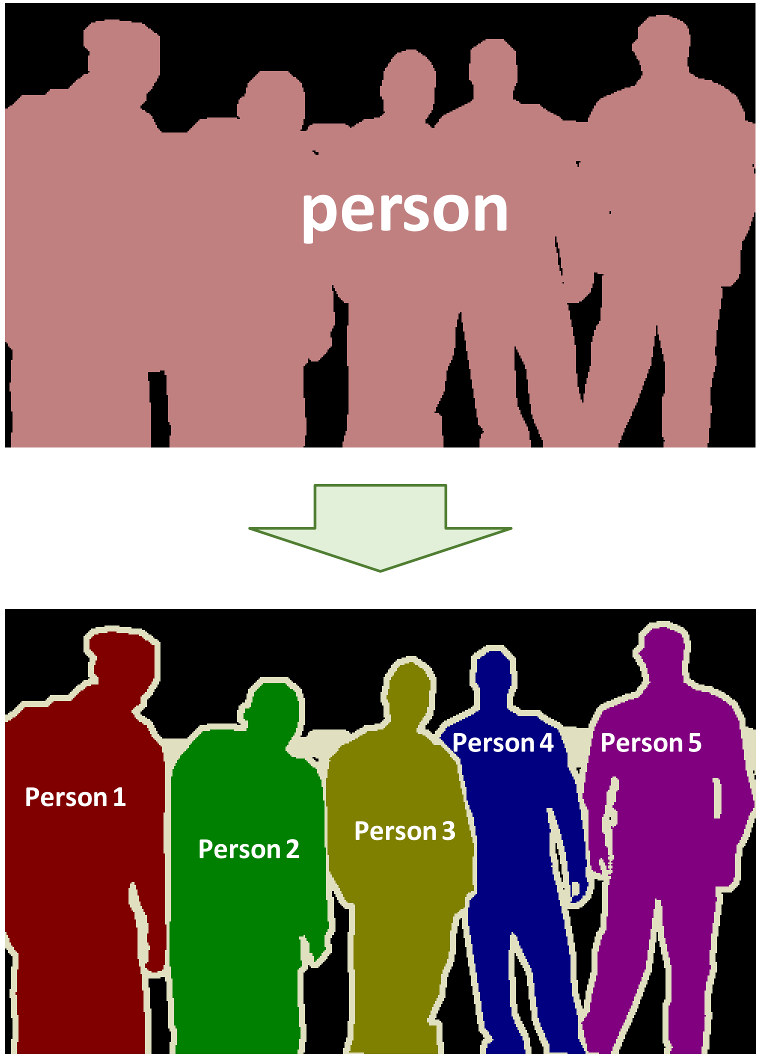


Instance Segmentation Methods

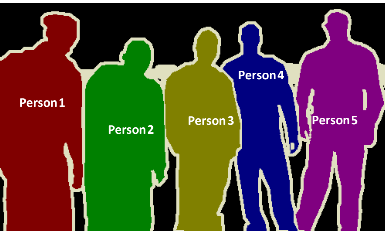
R-CNN driven



FCN driven

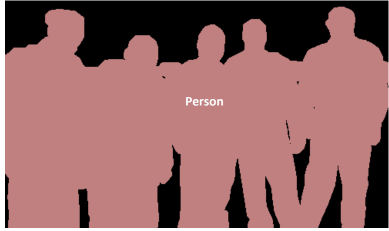


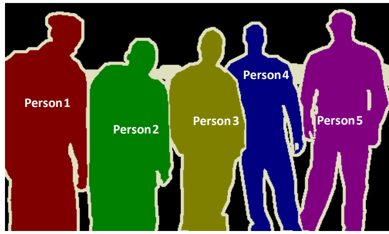
Instance Segmentation Methods



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]





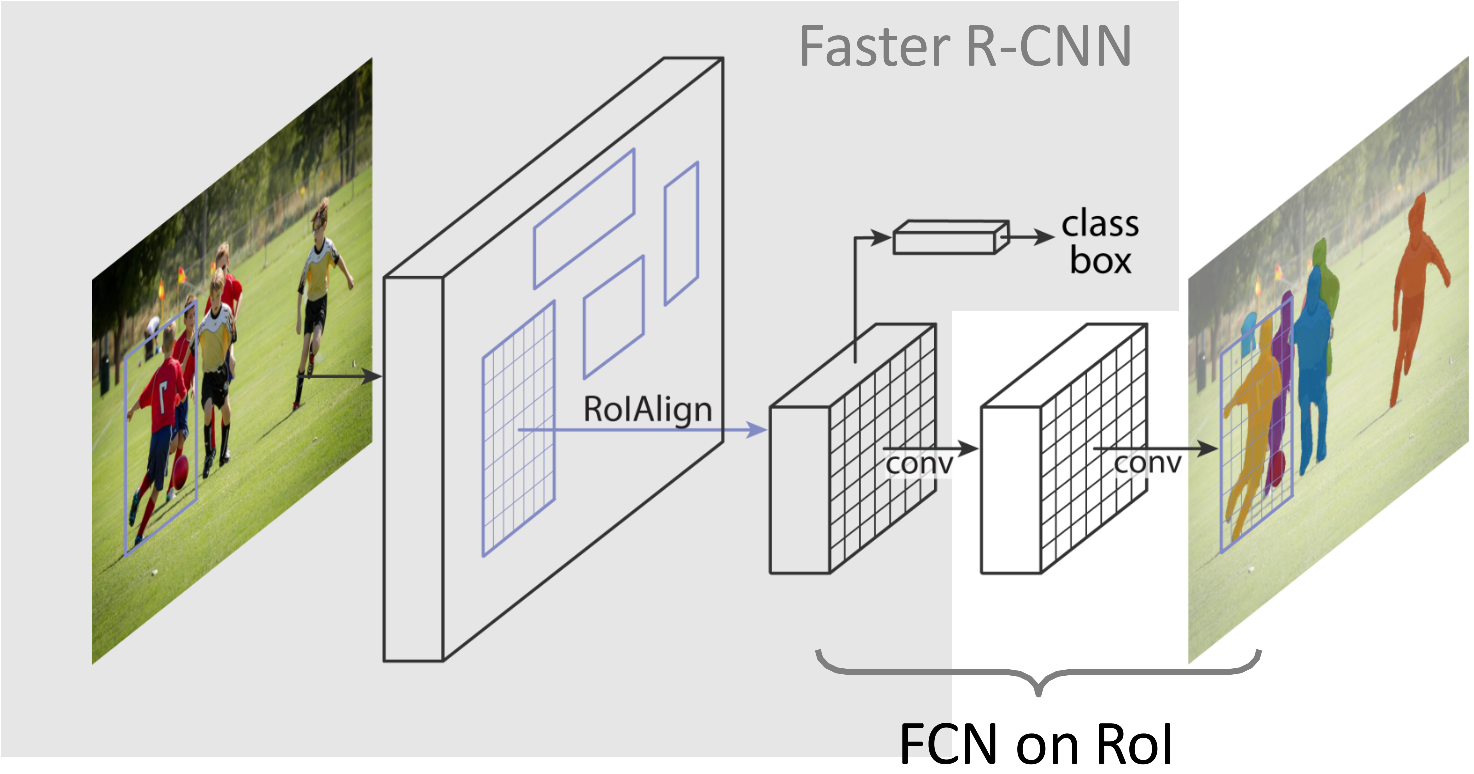
FCN-driven

- 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]

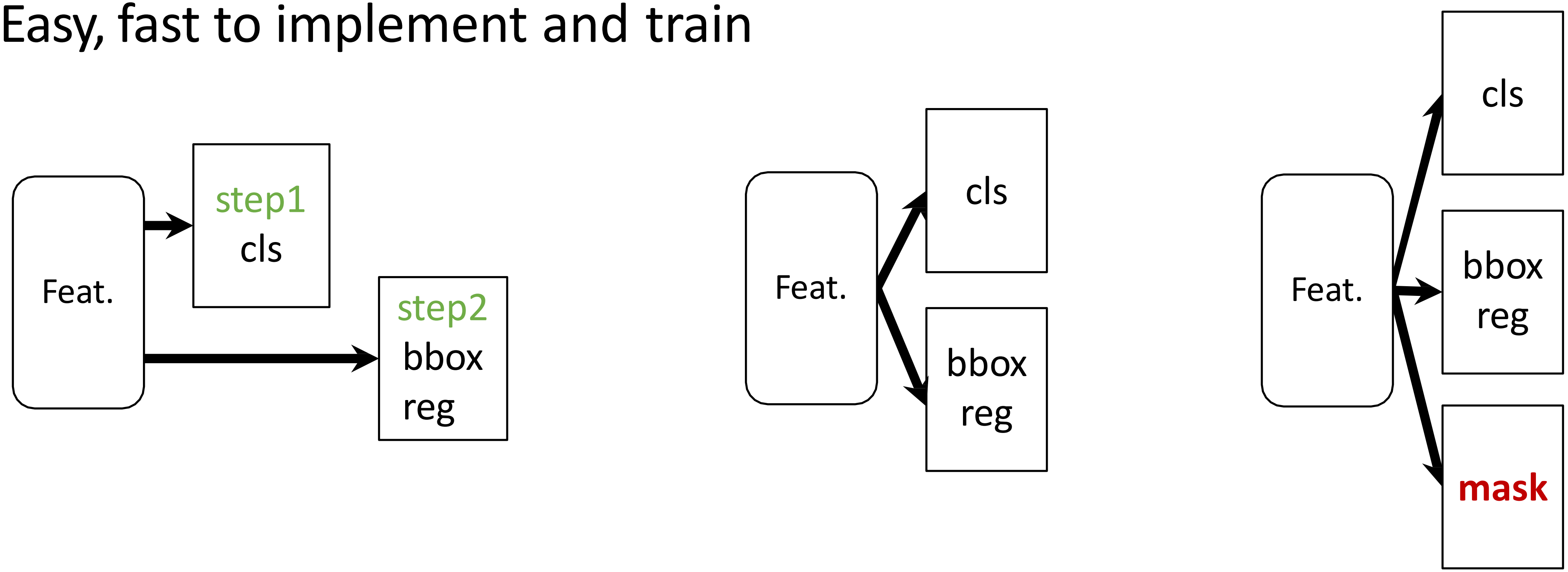
Mask R-CNN

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



Parallel Heads

- Easy, fast to implement and train



(slow) R-CNN

Fast/er R-CNN

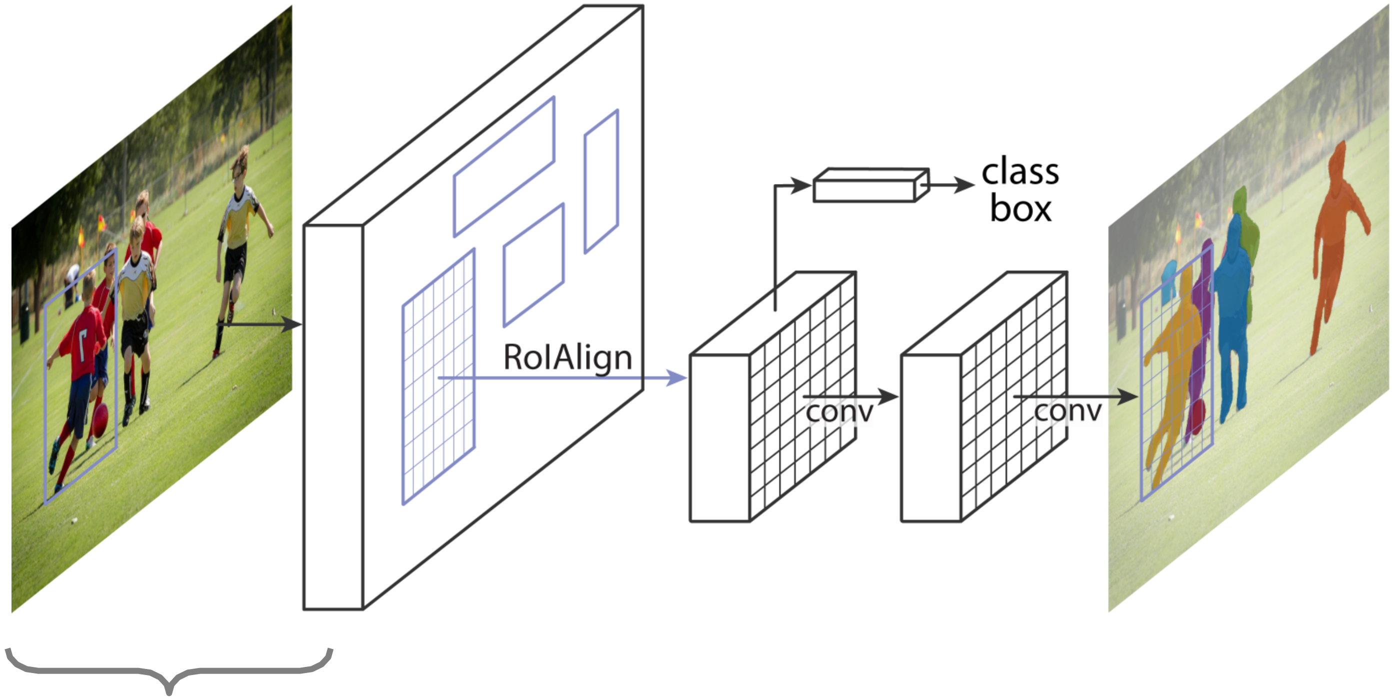
Mask R-CNN

Invariance vs. Equivariance

- **Equivariance**: changes in input lead to corresponding changes in output

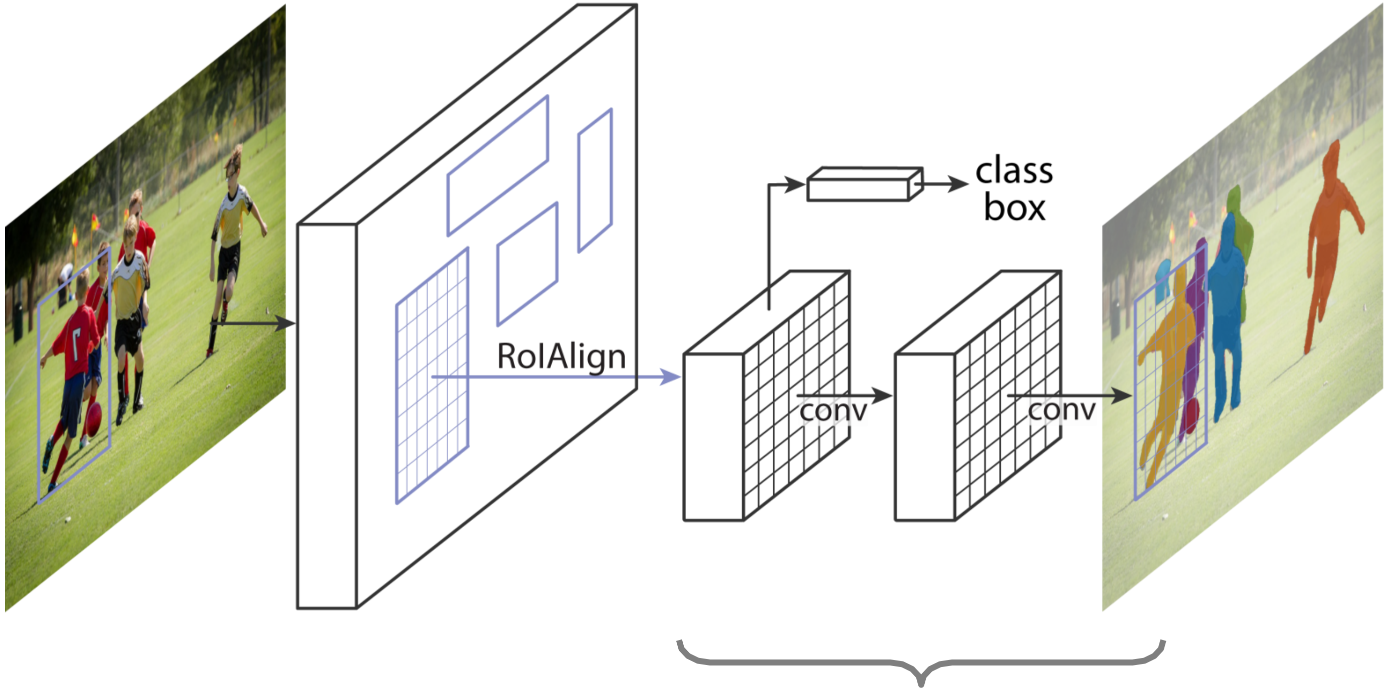
- *Classification* desires *invariant* representations: output a label
- *Instance Seg.* desires *equivariant* representations:
  - Translated object => translated mask
  - Scaled object => scaled mask
  - *Big and small* objects are equally important (due to AP metric)
    - unlike semantic seg. (counting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

Equivariance in Mask R-CNN



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

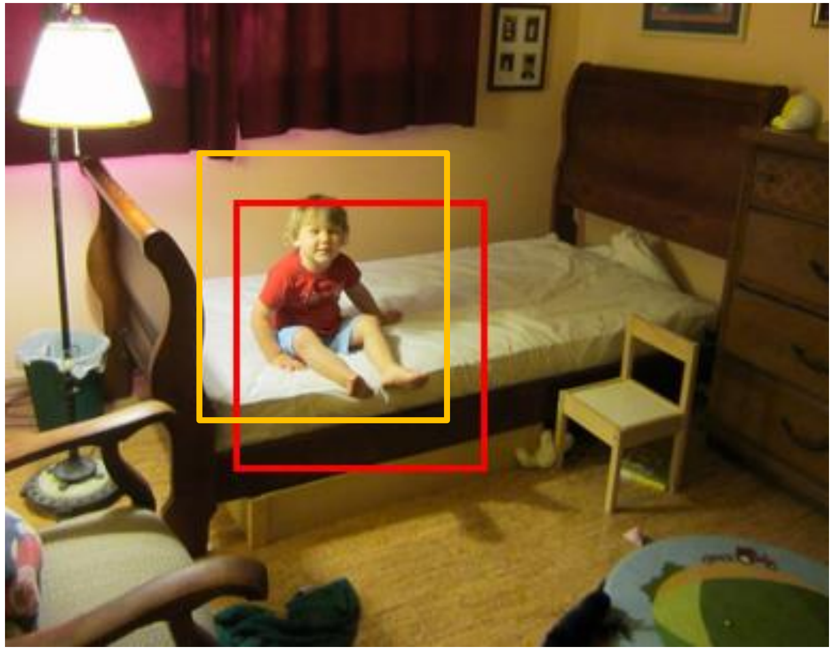
Equivariance in Mask R-CNN



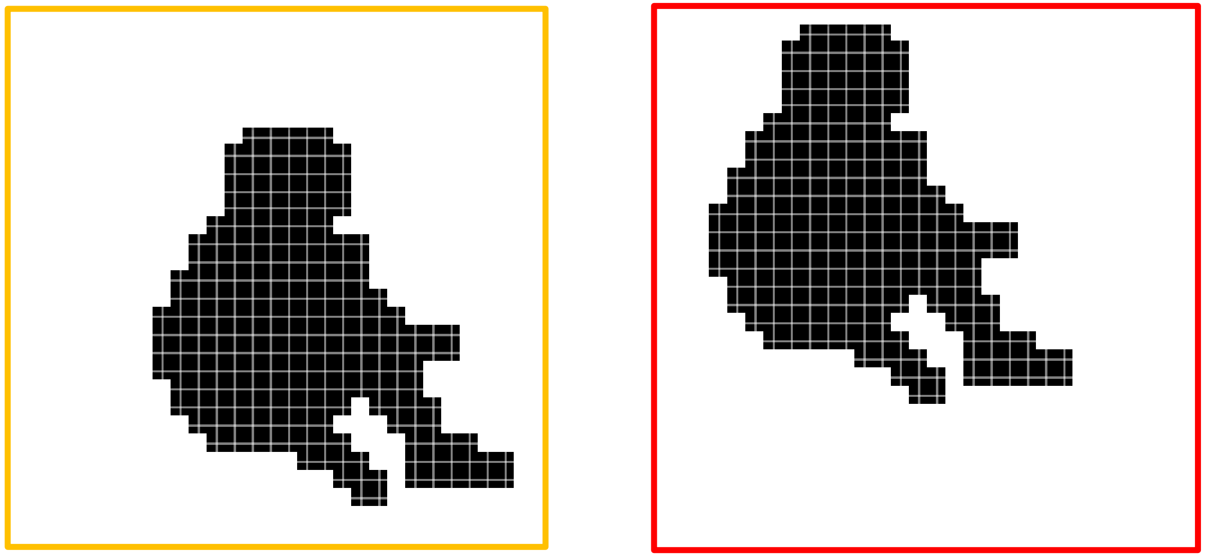
2. Fully-Conv on RoI:

equivariant to translation within RoI

Fully-Conv on RoI



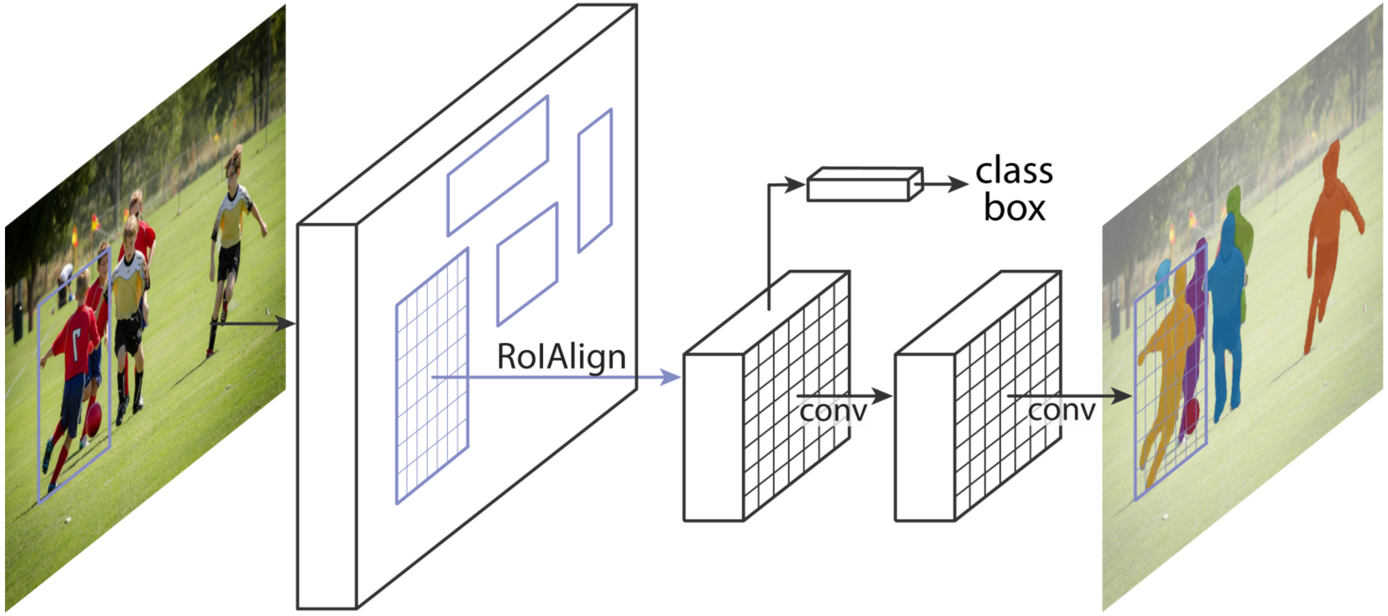
target masks on RoIs



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

- Equivariant to small translation of RoIs
- More robust to RoI's localization imperfection

Equivariance in Mask R-CNN



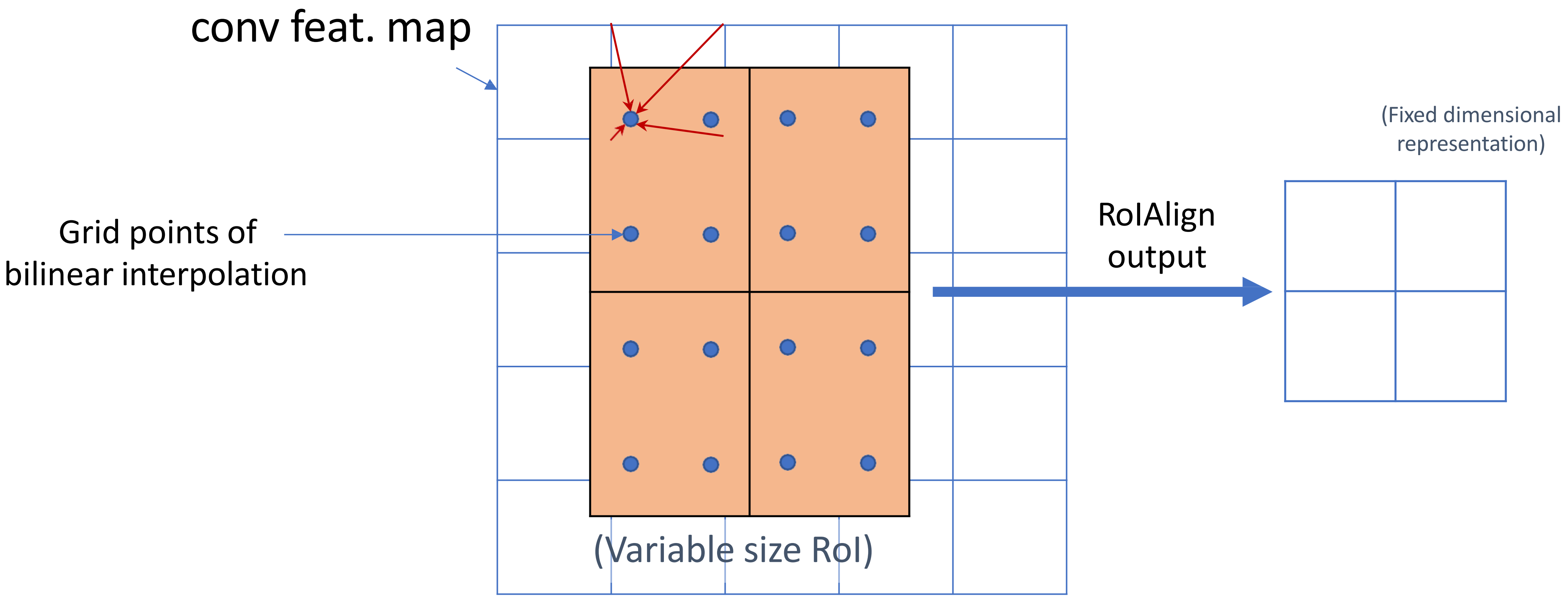
3. RoIAlign:

3a. maintain translation-equivariance before/after RoI

RoIAlign

FAQs: how to sample grid points within a cell?

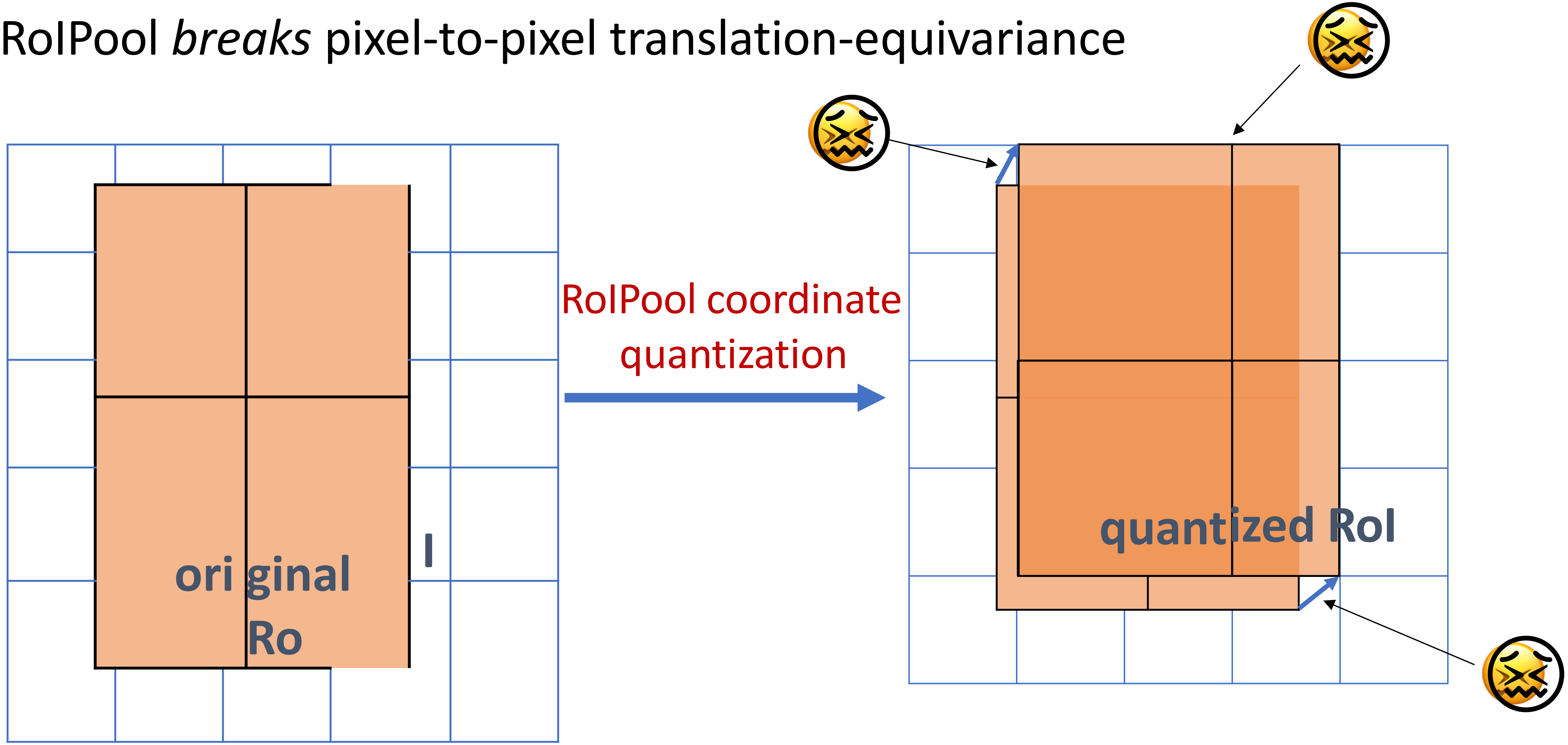
- 4 regular points in 2x2 sub-cells
- other implementation could work



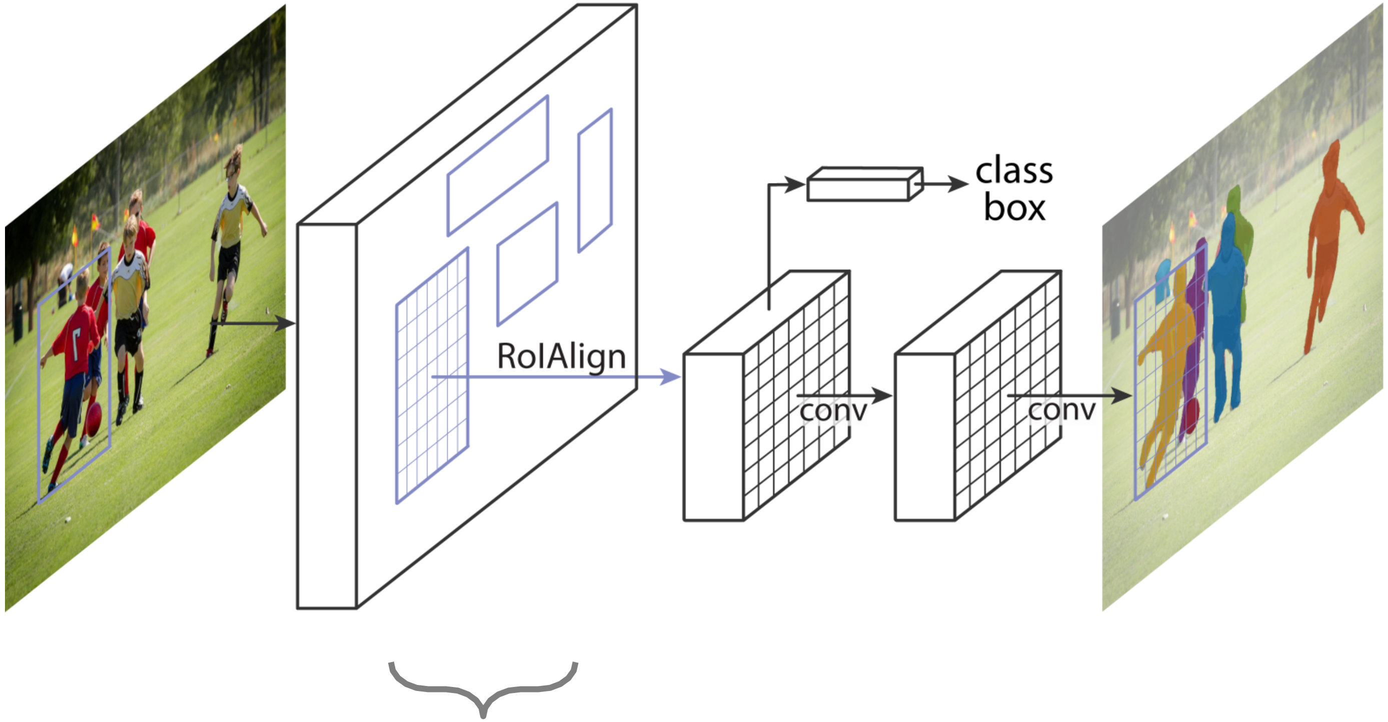
see also “What is wrong with convolutional neural nets?”, Geoffrey Hinton, 2017

RoIAlign vs. RoIPool

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



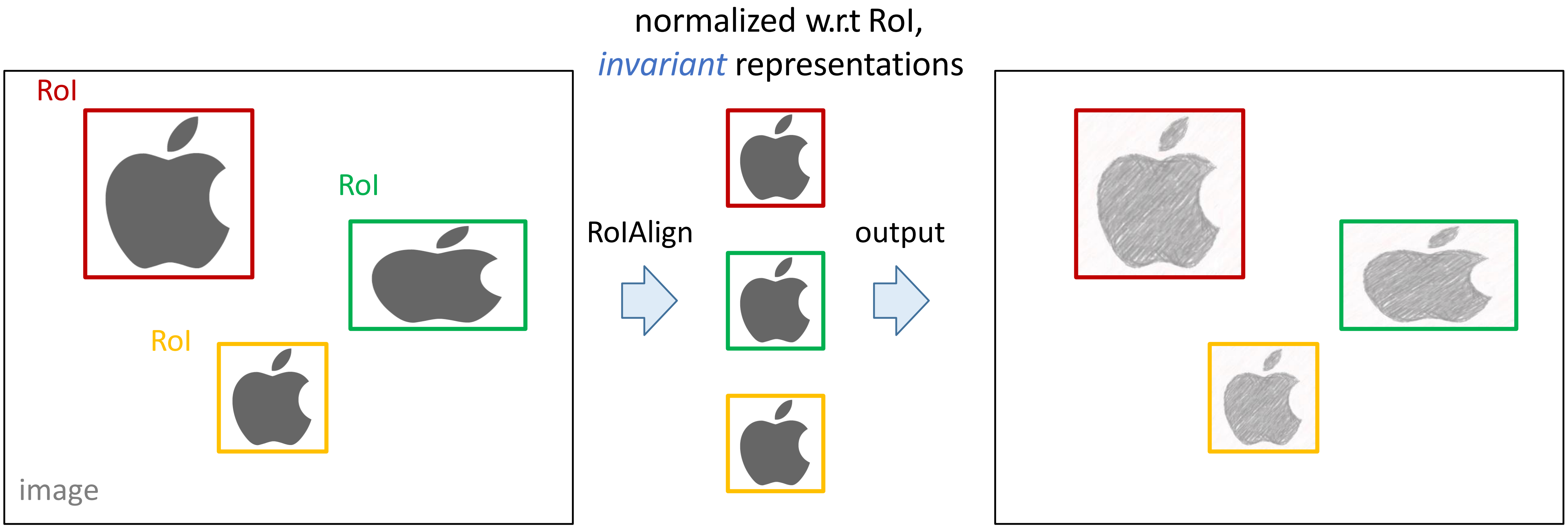
Equivariance in Mask R-CNN



3. RoIAlign:

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

RoIAlign: Scale-Equivariance

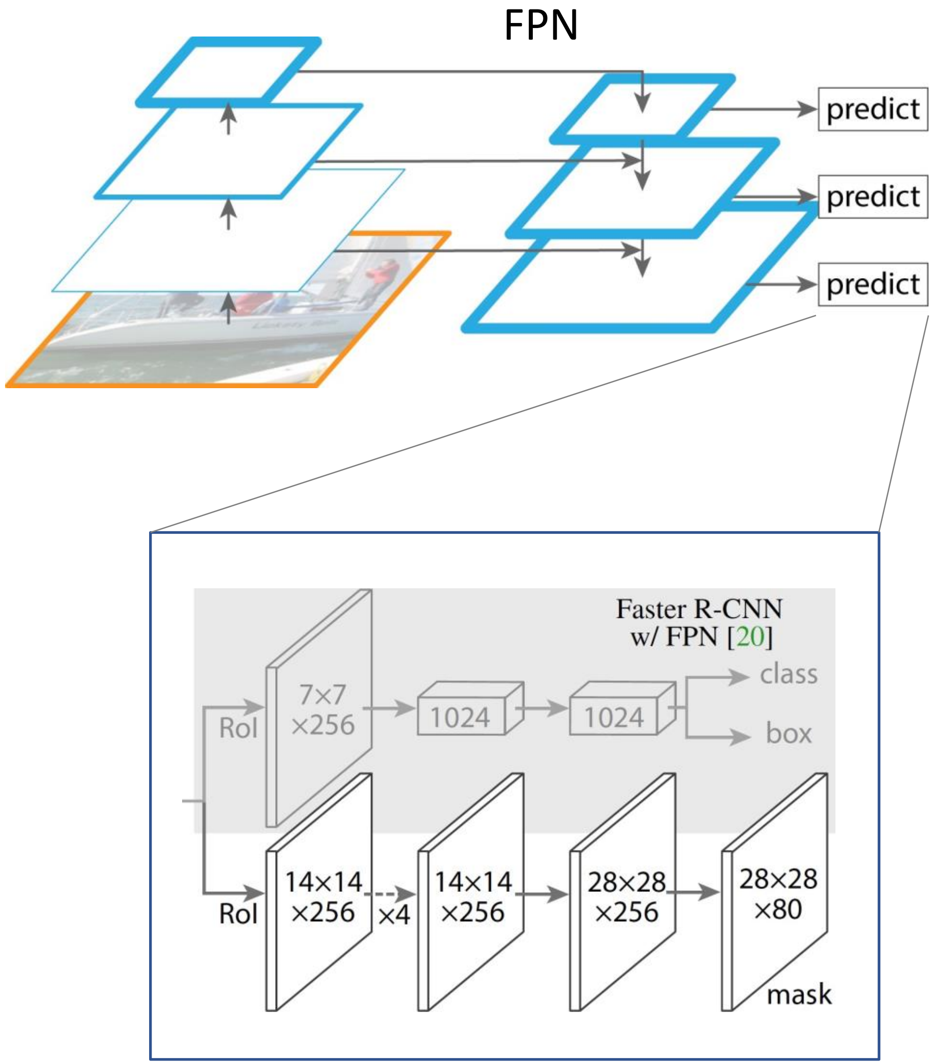


- RoIAlign creates *scale-invariant* representations
- RoIAlign + "output pasted back" provides *scale-equivariance*

More about Scale-Equivariance: FPN

- RoIAlign is scale-invariant if on raw pixels:
  - = (slow) R-CNN: crops and warps RoIs
- RoIAlign is scale-invariant if on scale-invariant feature maps

- Feature Pyramid Network (FPN) [Lin et al. CVPR'17] creates approx. scale-invariant features



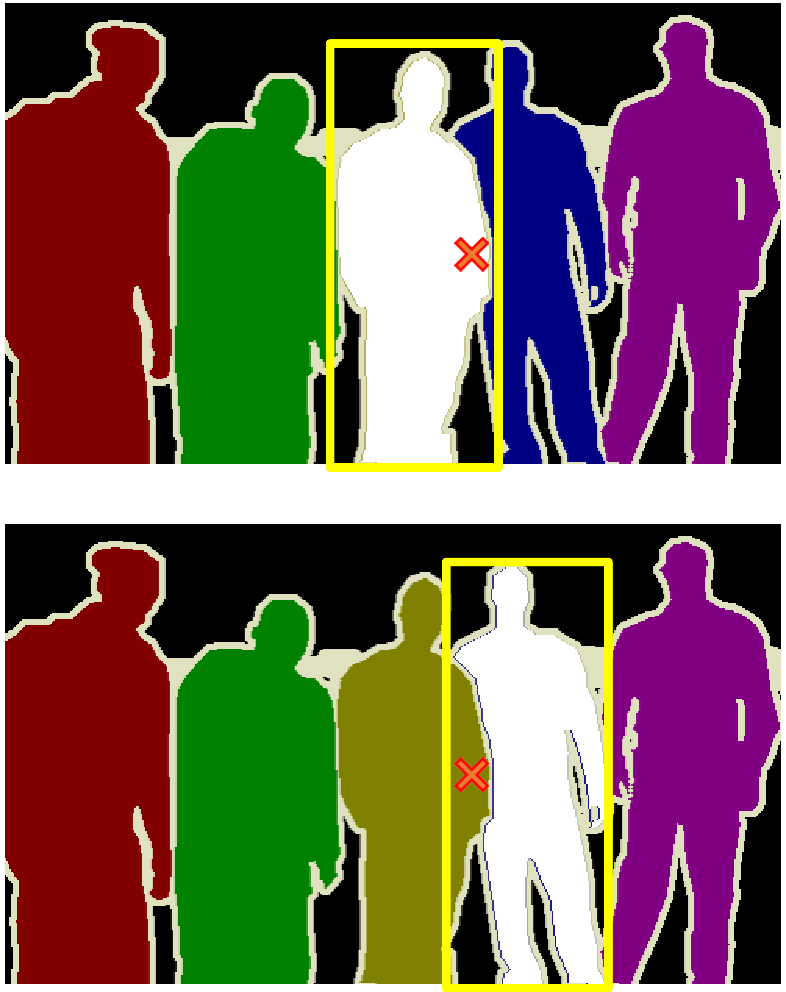
Equivariance in Mask R-CNN: Summary

- Translation-equivariant
  - FCN features
  - FCN mask head
  - RoIAlign (pixel-to-pixel behavior)

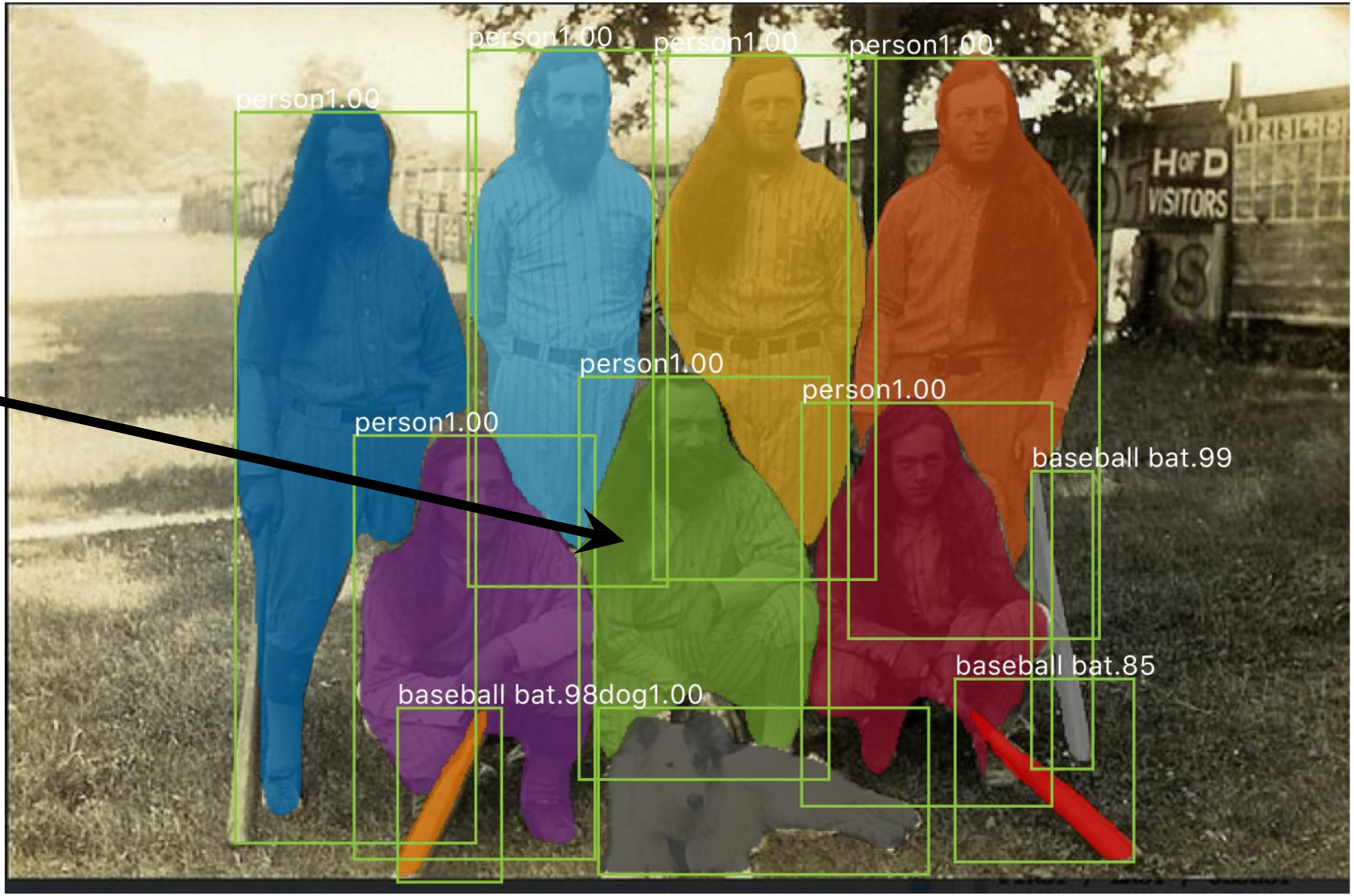
- Scale-equivariant (and aspect-ratio-equivariant)
  - RoIAlign (warping and normalization behavior) + paste-back
  - FPN features

Instance Seg: When we don't want equivariance?

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



object surrounded by same-category objects



Mask R-CNN results on COCO

Result Analysis

Ablation: RoIPool vs. RoIAlign

baseline: ResNet-50-Conv5 backbone, **stride=32**

	mask AP			box AP		
	AP	AP_{50}	AP_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}
RoIPool	23.6	46.5	21.6	28.2	52.7	26.9
RoIAlign	**30.9**	**51.8**	**32.1**	**34.0**	**55.3**	**36.4**
	+7.3	*+ 5.3*	*+10.5*	*+5.8*	*+2.6*	*+9.5*

- huge gain at high IoU,
  in case of big stride (32)

Ablation: RoIPool vs. RoIAlign

baseline: ResNet-50-Conv5 backbone, **stride=32**

	mask AP			box AP		
	AP	AP_{50}	AP_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}
RoIPool	23.6	46.5	21.6	28.2	52.7	26.9
RoIAlign	**30.9**	**51.8**	**32.1**	**34.0**	**55.3**	**36.4**
	+7.3	*+ 5.3*	*+10.5*	*+5.8*	*+2.6*	*+9.5*

- nice box AP without dilation/upsampling

Instance Segmentation Results on COCO

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
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	**37.1**	**60.0**	**39.4**	**16.9**	**39.9**	**53.5**

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

Instance Segmentation Results on COCO

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
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	**37.1**	**60.0**	**39.4**	**16.9**	**39.9**	**53.5**

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

Object Detection Results on COCO

	backbone	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{bb}_{S}	AP^{bb}_{M}	AP^{bb}_{L}
Faster R-CNN+++ [15]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [22]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [17]	Inception-ResNet-v2 [32]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [31]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
Faster R-CNN, RoIAlign	ResNet-101-FPN	37.3	59.6	40.3	19.8	40.2	48.8
Mask R-CNN	ResNet-101-FPN	38.2	60.3	41.7	20.1	41.1	50.2
Mask R-CNN	ResNeXt-101-FPN	**39.8**	**62.3**	**43.4**	**22.1**	**43.2**	51.2

bbox detection improved by:

- RoIAlign

Object Detection Results on COCO

	backbone	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{bb}_{S}	AP^{bb}_{M}	AP^{bb}_{L}
Faster R-CNN+++ [15]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [22]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [17]	Inception-ResNet-v2 [32]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [31]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
Faster R-CNN, RoIAlign	ResNet-101-FPN	37.3	59.6	40.3	19.8	40.2	48.8
Mask R-CNN	ResNet-101-FPN	38.2	60.3	41.7	20.1	41.1	50.2
Mask R-CNN	ResNeXt-101-FPN	**39.8**	**62.3**	**43.4**	**22.1**	**43.2**	51.2

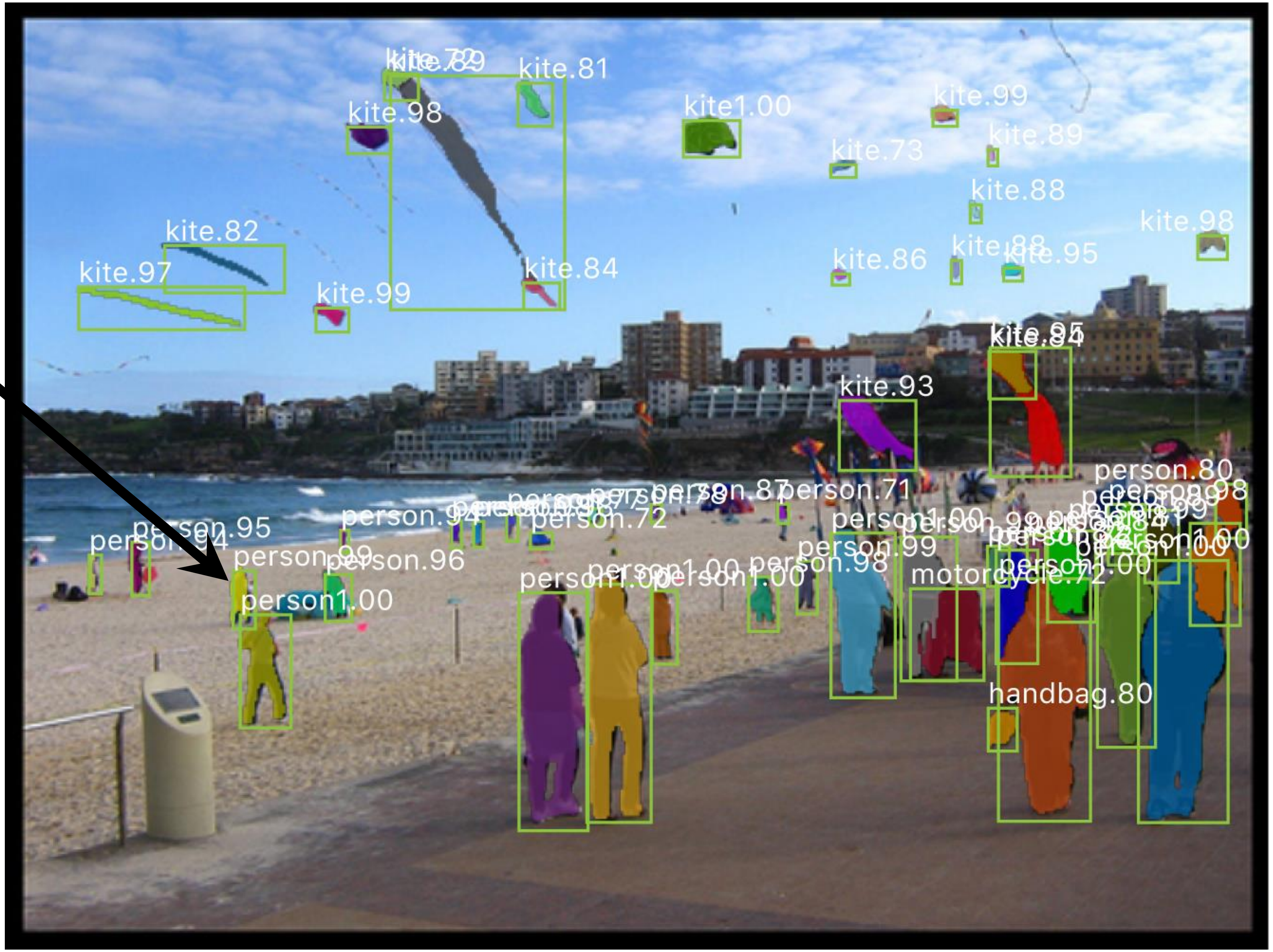
bbox detection improved by:

- RoIAlign
- Multi-task training w/ mask

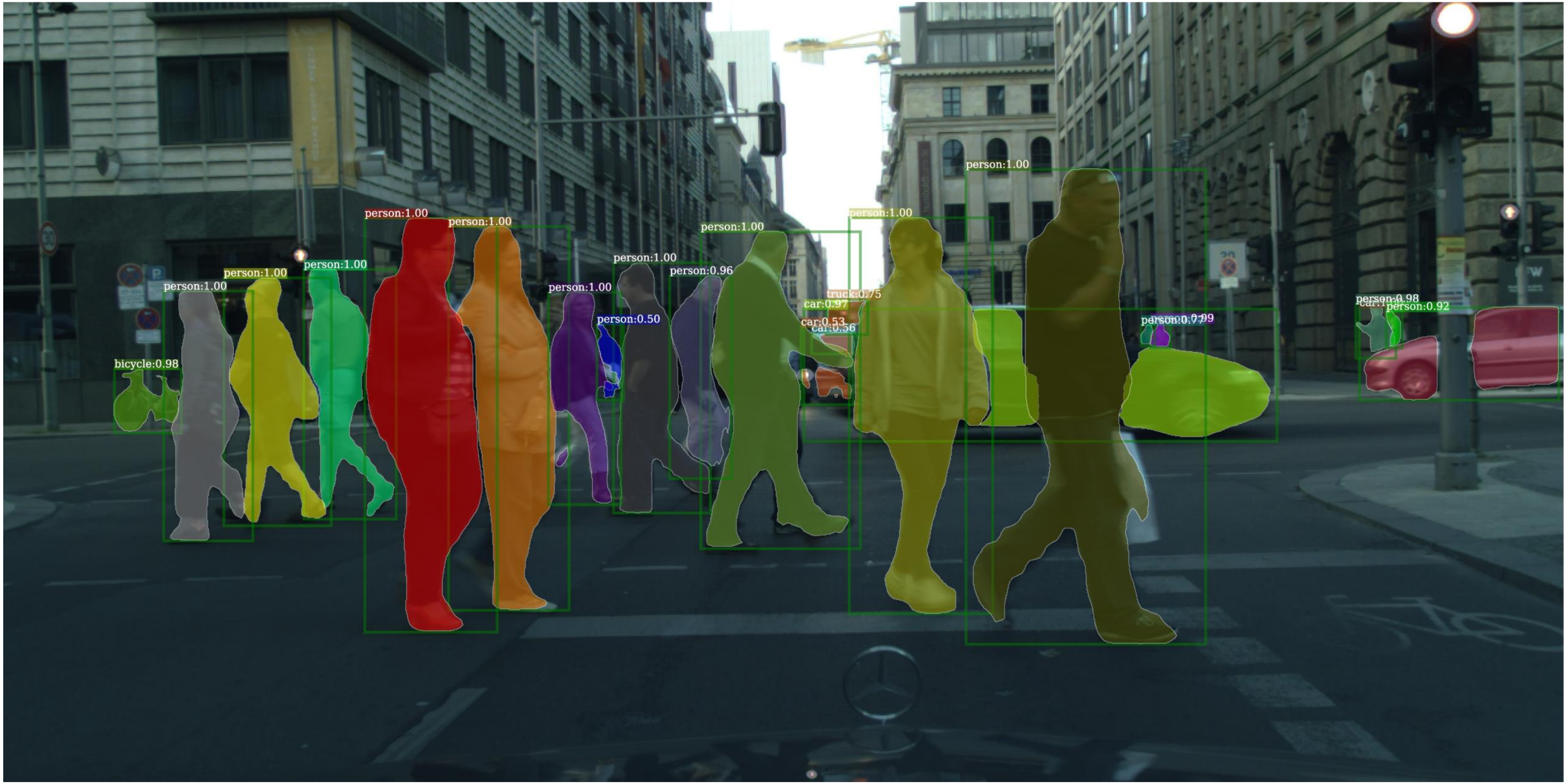
disconnected
object

Mask R-CNN results on COCO

small
objects



Mask R-CNN results on COCO

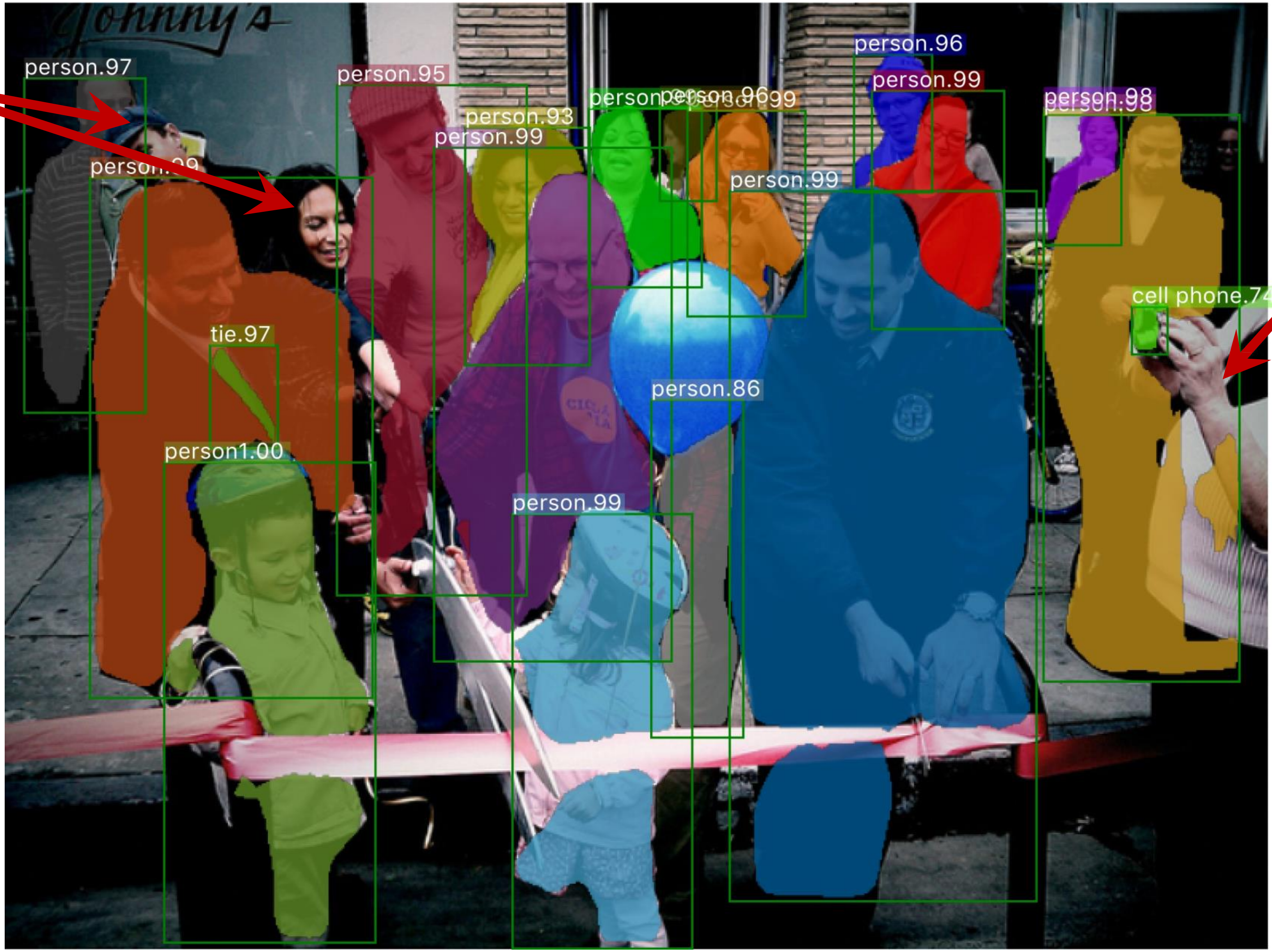


Mask R-CNN results on CityScapes

Failure case: detection/segmentation

missing

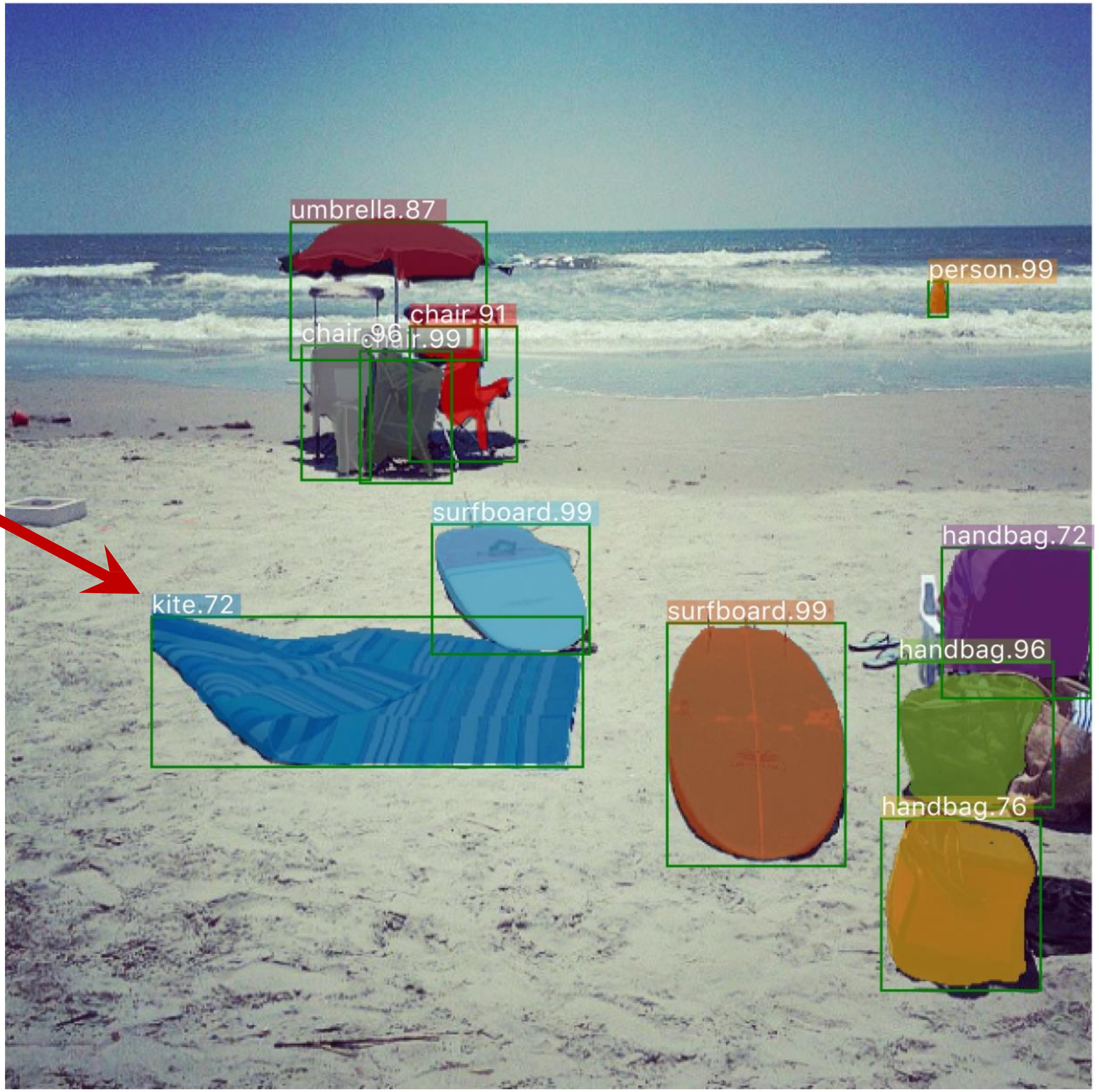
missing, false mask



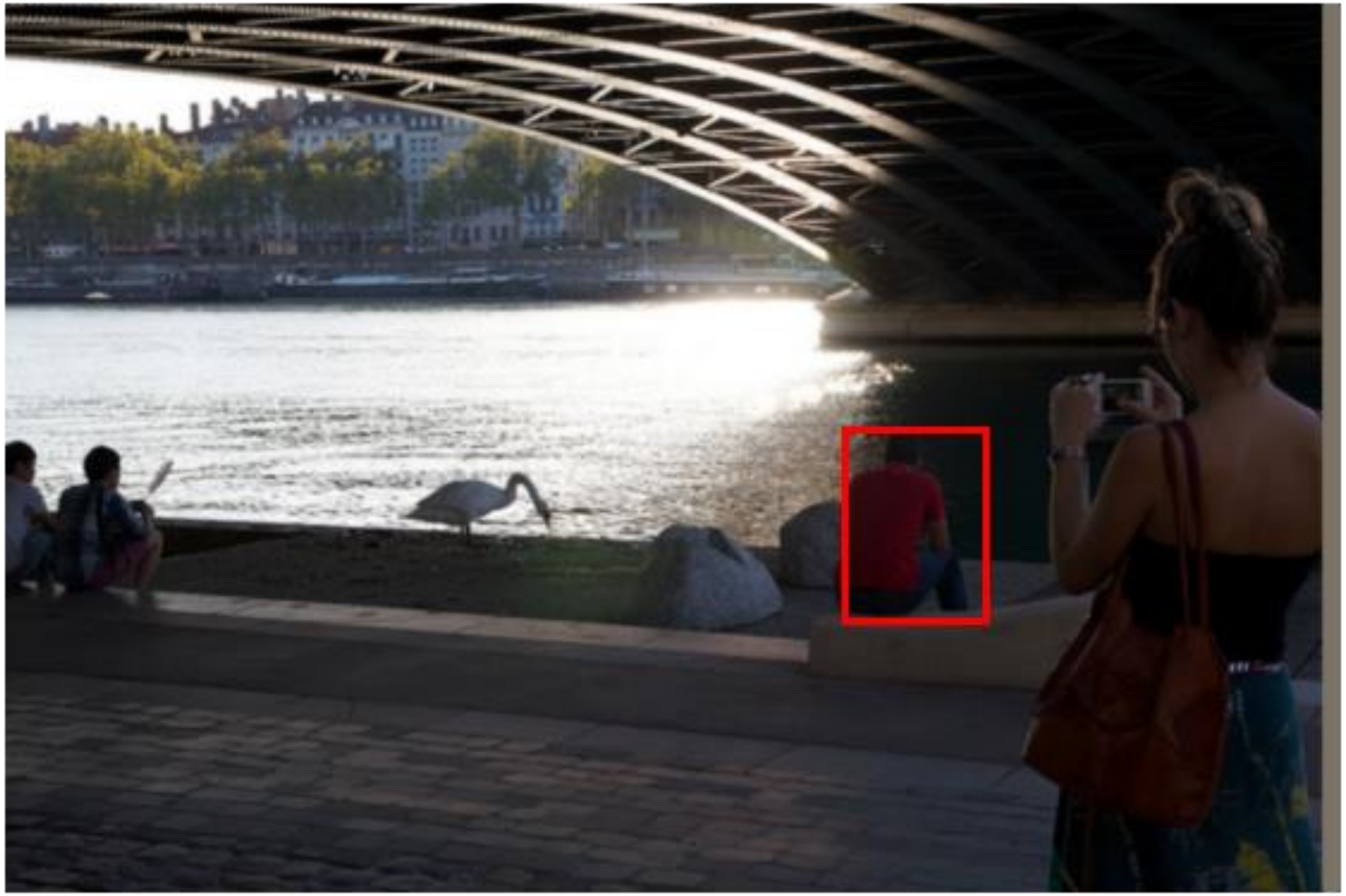
Mask R-CNN results on COCO

Failure case: recognition

not a kite



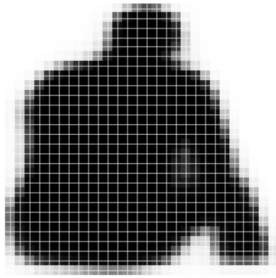
Mask R-CNN results on COCO



Validation image with box detection shown in red

28x28 soft prediction from Mask R-CNN

(enlarged)



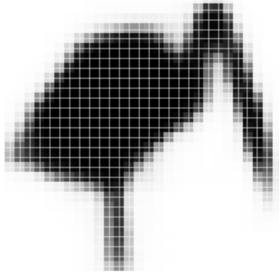
Soft prediction resampled to image coordinates

(bilinear and bicubic interpolation work equally well)

Final prediction (threshold at 0.5)

Validation image with box detection shown in red

28x28 soft prediction



Resized Soft prediction

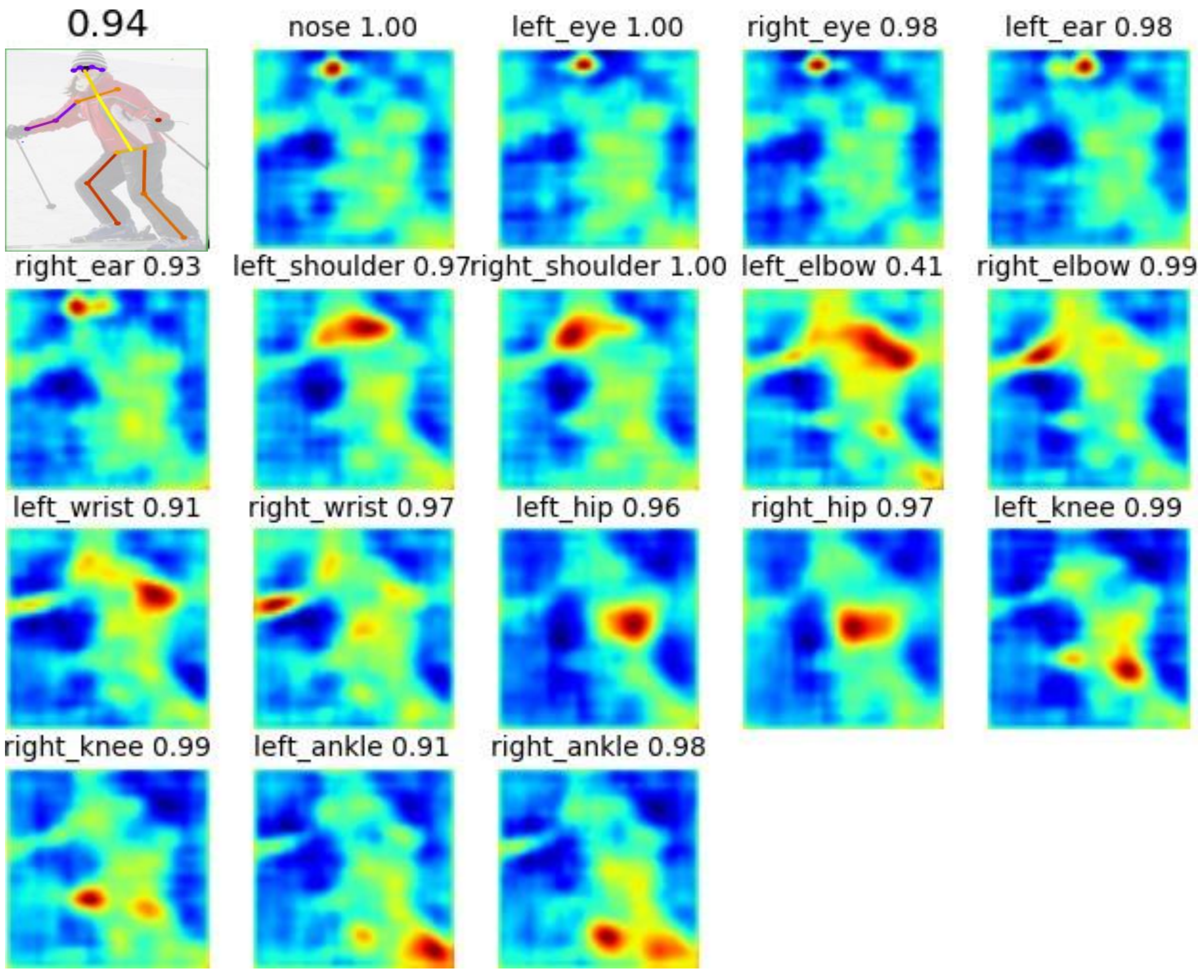
Final mask

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 56x56

- Desire the same equivariances
  - translation, scale, aspect ratio

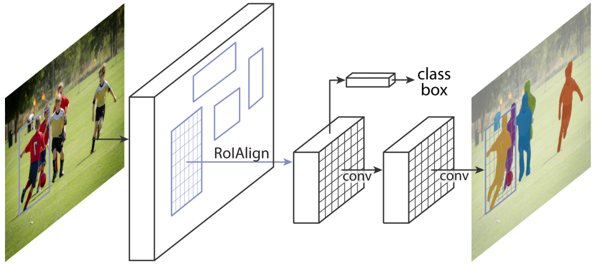



www.tfrancis.co.uk

Conclusion

Mask R-CNN

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



Code will be 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
- Semantic Keypoints
- Object Detection