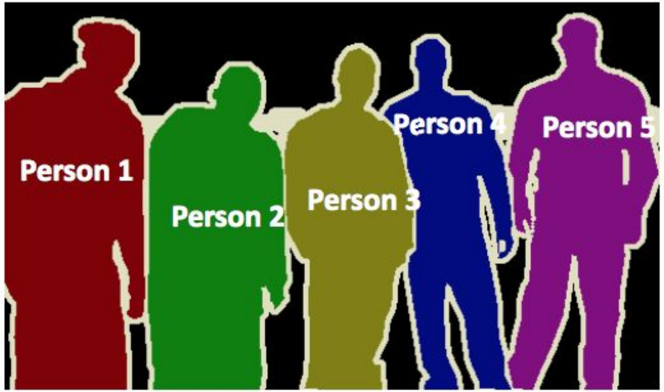
Instance Segmentation

Riley Simmons-Edler, Berthy Feng

Instance Segmentation Task

- Label each foreground pixel with object and instance
- Object detection + semantic segmentation



Instance Segmentation

Slide Credit: Kaiming He

In This Lecture...

- Microsoft COCO dataset
- Mask R-CNN (fully supervised)
- $Mask^X$ R-CNN (partially supervised)

Microsoft COCO: Common Objects in Context

Tsung-Yi Lin, Michael Maire, Serge Belongie, et al. “Microsoft COCO: Common Objects in Context.” arXiv, 2015.

Previous Datasets

- **ImageNet**: many object categories
- **PASCAL VOC**: object detection in natural images, small number of classes
- **SUN**: labeling scene types and commonly occurring objects, but not many instances per category

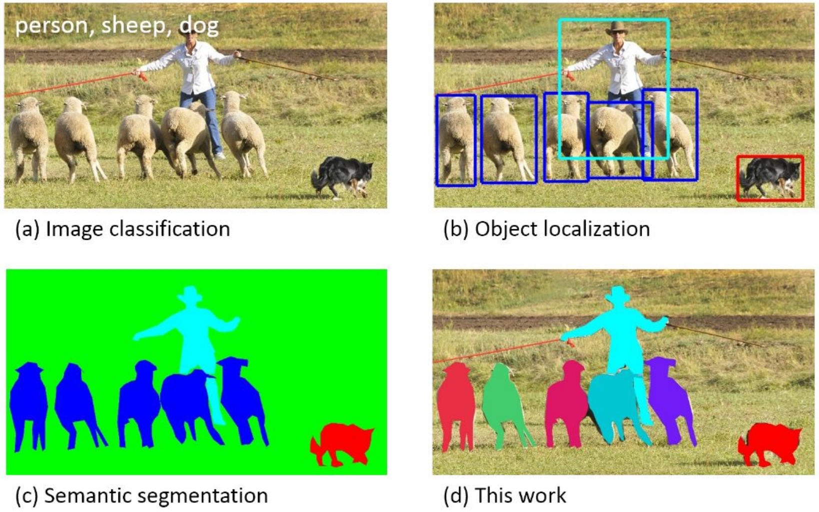


Image Credit: Tsung-Yi Lin et al.

Goal: Push research in scene understanding

1. Detecting non-iconic views
2. Contextual reasoning between objects
3. Precise 2D localization of objects

MS COCO Dataset

- 91 object classes
- 328,000 images
- 2.5 million labeled instances

Image Credit: Tsung-Yi Lin et al.

Image Collection & Annotation

Object Categories

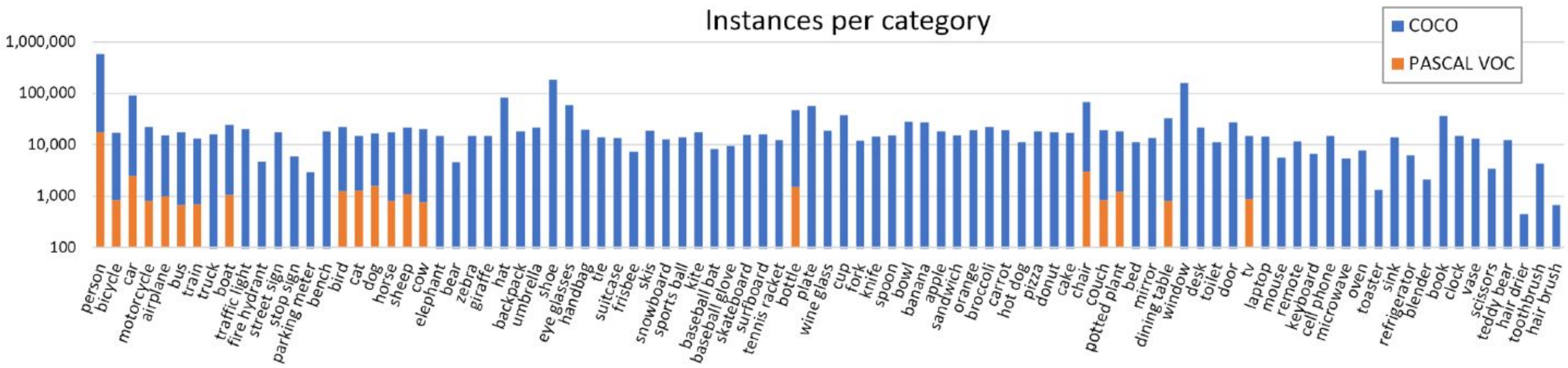


Image Credit: Tsung-Yi Lin et al.

Non-Iconic Image Collection

(a) Iconic object images (b) Iconic scene images (c) Non-iconic images

Image Credit: Tsung-Yi Lin et al.

Annotation

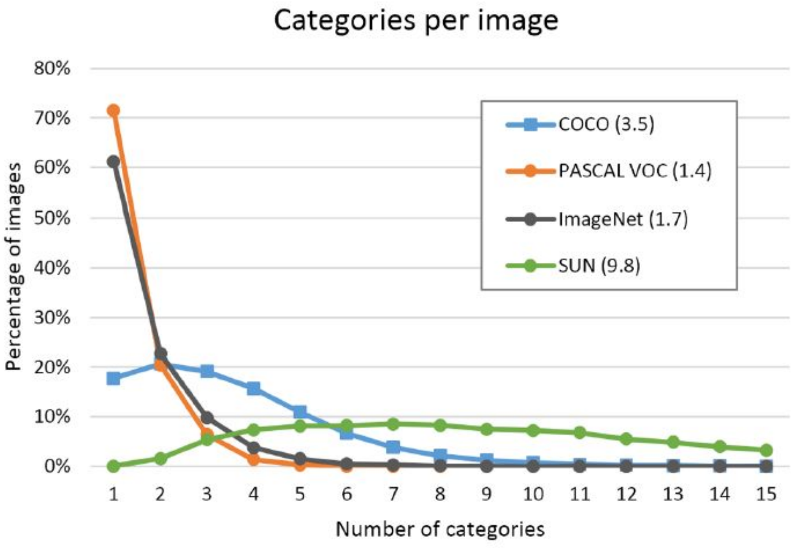
(a) Category labeling (b) Instance spotting (c) Instance segmentation

Fig. 3: Our annotation pipeline is split into 3 primary tasks: (a) labeling the categories present in the image (§4.1), (b) locating and marking all instances of the labeled categories (§4.2), and (c) segmenting each object instance (§4.3).

Image Credit: Tsung-Yi Lin et al.

Dataset Evaluation

Statistics



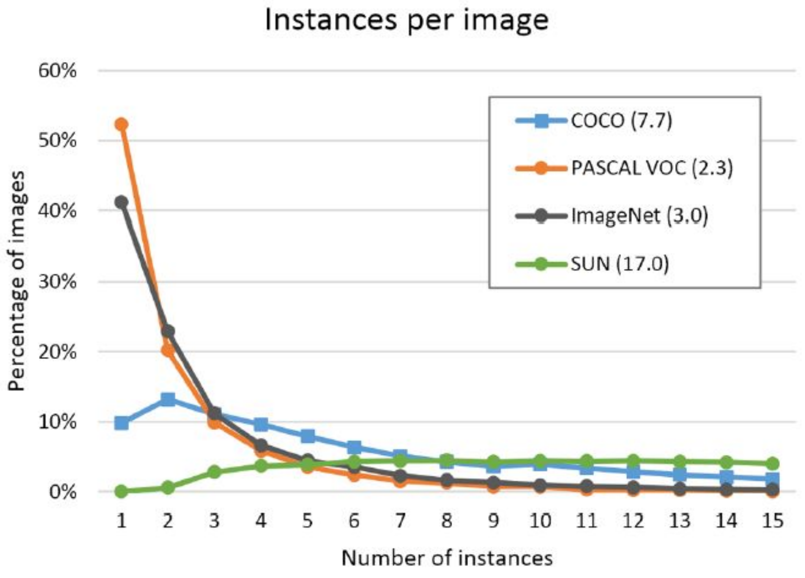
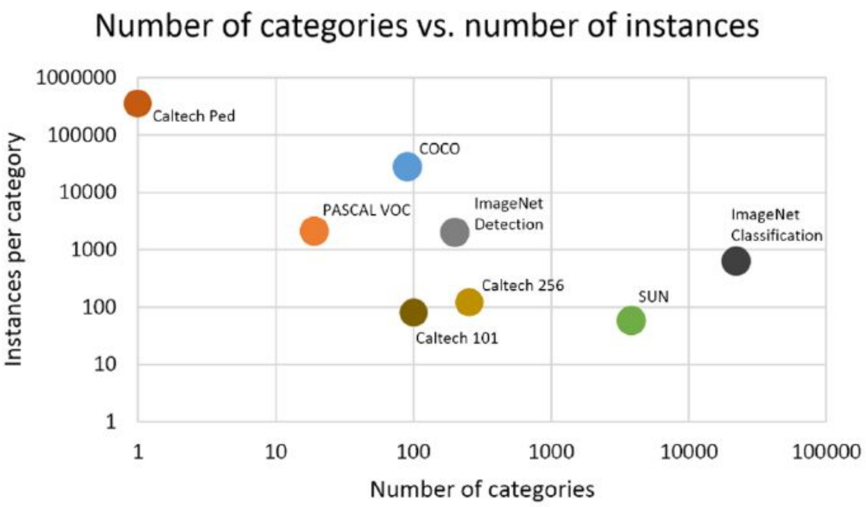


Image Credit: Tsung-Yi Lin et al.

Statistics



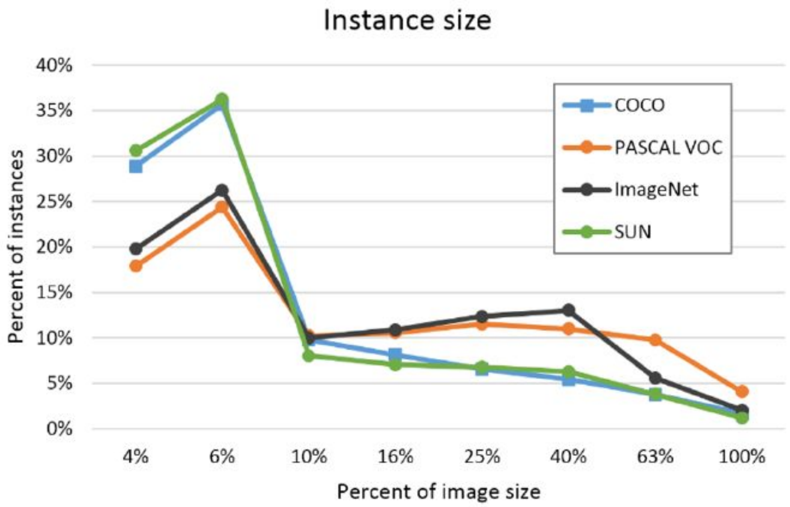


Image Credit: Tsung-Yi Lin et al.

COCO Detection Challenge

The COCO 2017 Detection Challenge is designed to push the state of the art in object detection forward. Teams are encouraged to compete in either (or both) of two object detection challenges: using bounding box output or object segmentation output. For full details of this task please see the COCO Detection Challenge page.

Image Credit: Tsung-Yi Lin et al.

COCO Keypoint Challenge

The COCO 2017 Keypoint Challenge requires localization of person keypoints in challenging, uncontrolled conditions. The keypoint challenge involves simultaneously detecting people *and* localizing their keypoints (person locations are *not* given at test time). For full details of this task please see the COCO Keypoints Challenge page.

Image Credit: Tsung-Yi Lin et al.

COCO Stuff Challenge

The COCO 2017 Stuff Segmentation Challenge is designed to push the state of the art in semantic segmentation of *stuff* classes. Whereas the COCO 2017 Detection Challenge addresses *thing* classes (person, car, elephant), this challenge focuses on *stuff* classes (grass, wall, sky). For full details of this task please see the COCO Stuff Challenge page.

Image Credit: Tsung-Yi Lin et al.

COCO Places Challenges

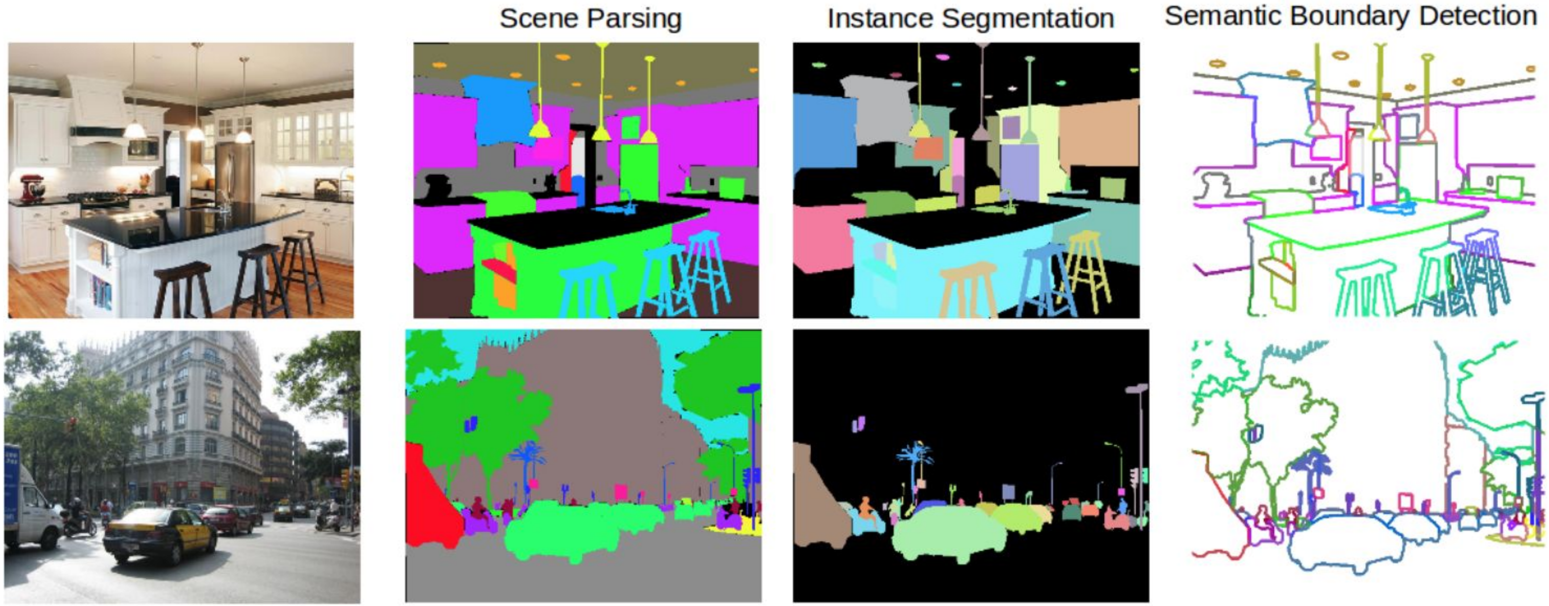


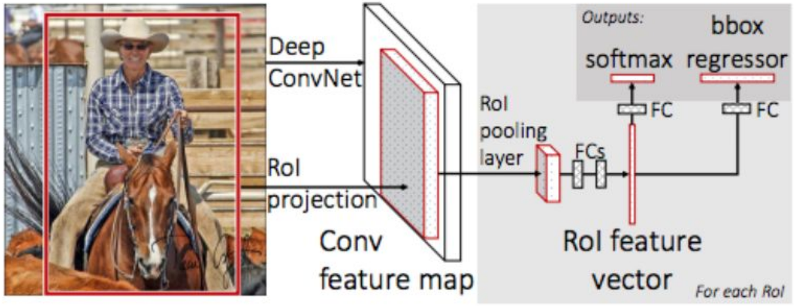
Image Credit: Tsung-Yi Lin et al.

Mask R-CNN

Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. “Mask R-CNN.” ICCV, 2017.

Faster R-CNN

Fast R-CNN



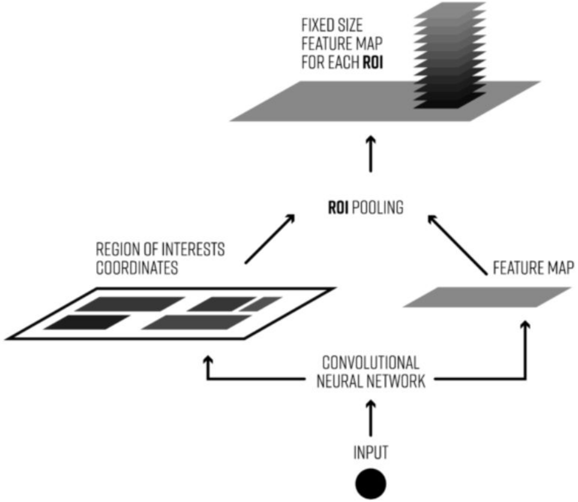


Image Credit: Shaoqing Ren et al.

Image Credit: Tomasz Grel

Insight: Region Proposal and Detection Use Same Features

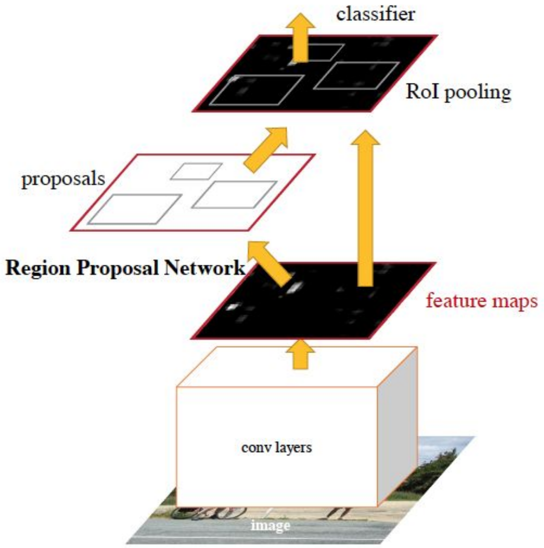
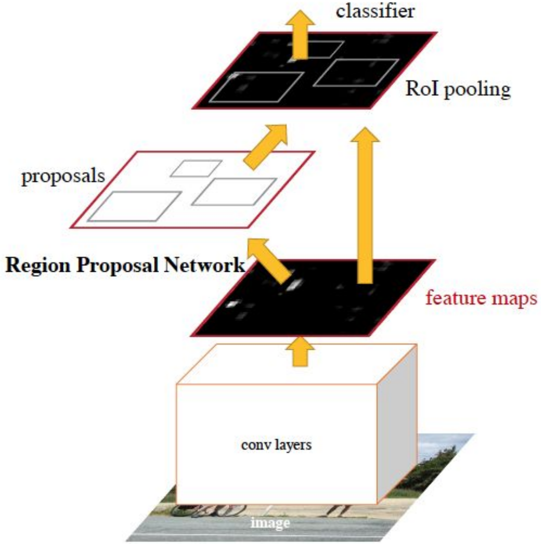


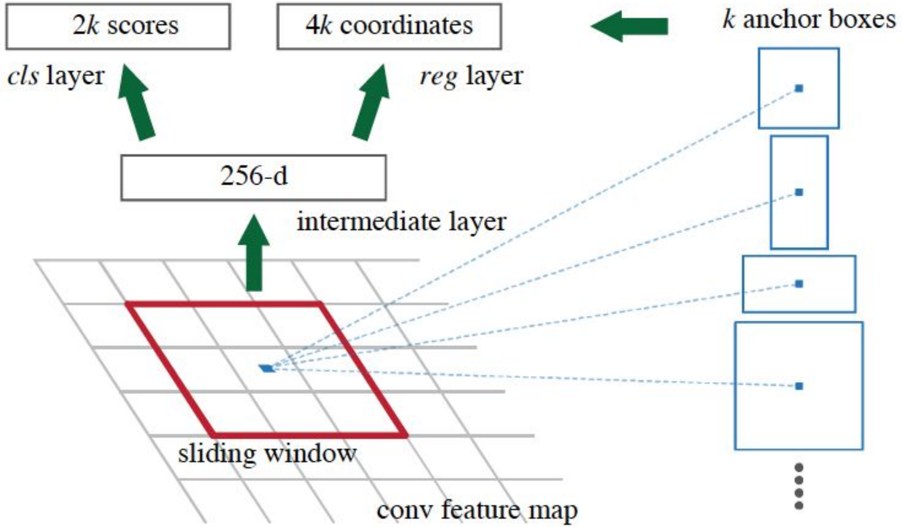
Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

Image Credit: Shaoqing Ren et al.

Faster R-CNN = RPN + Fast R-CNN

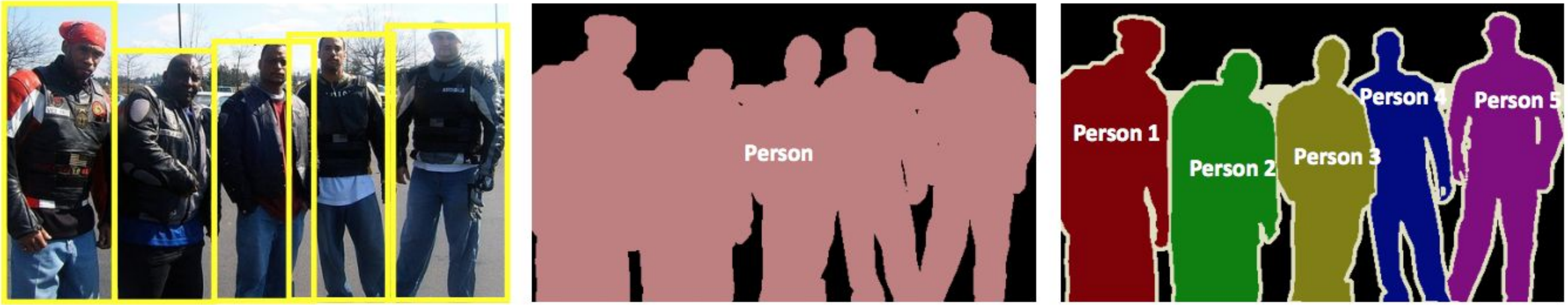


RPN = Fully Convolutional Network



Extending to Instance Segmentation

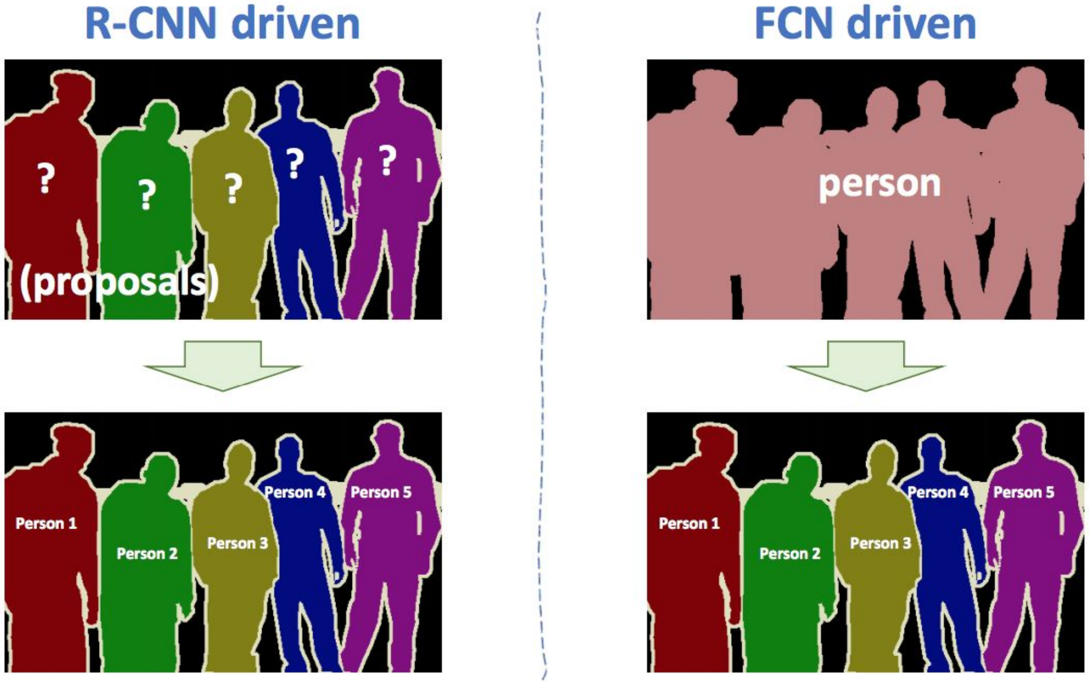
Visual Perception Problems



Object Detection	Semantic Segmentation	Instance Segmentation
✓	✓	?

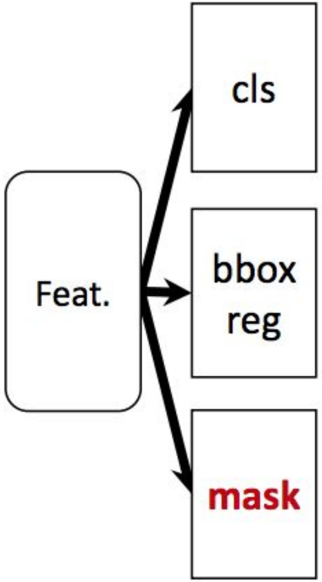
Slide Credit: Kaiming He

Instance Segmentation Methods

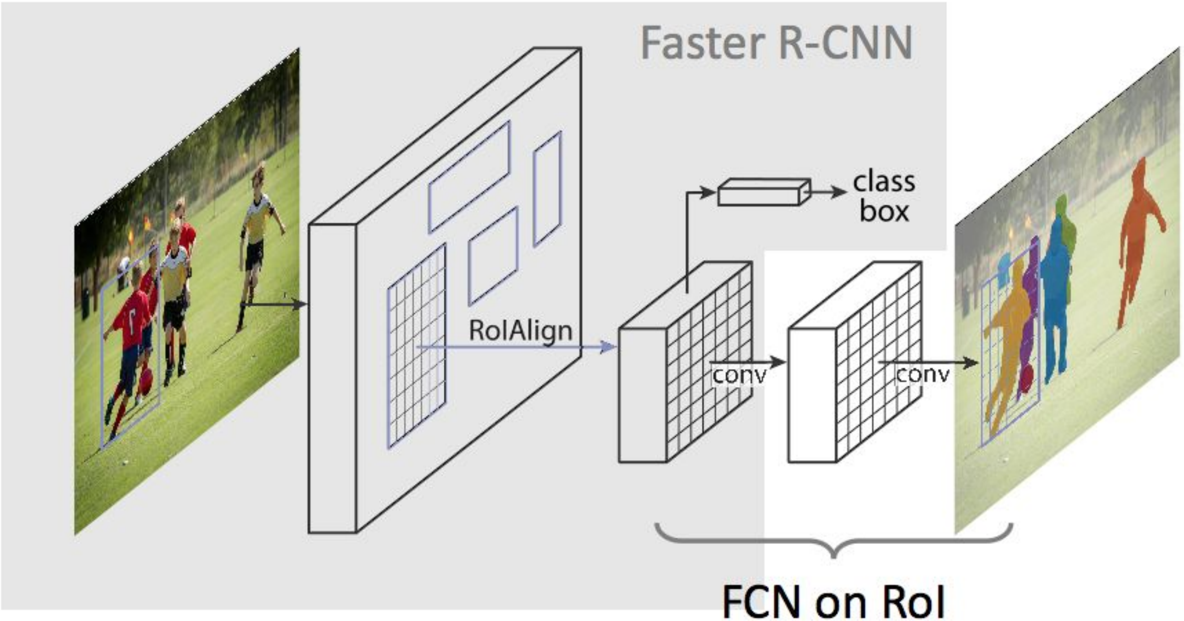


Slide Credit: Kaiming He

Insight: Mask Prediction in Parallel



Mask R-CNN



FCN on RoI

Slide Credit: Kaiming He

RoIPool

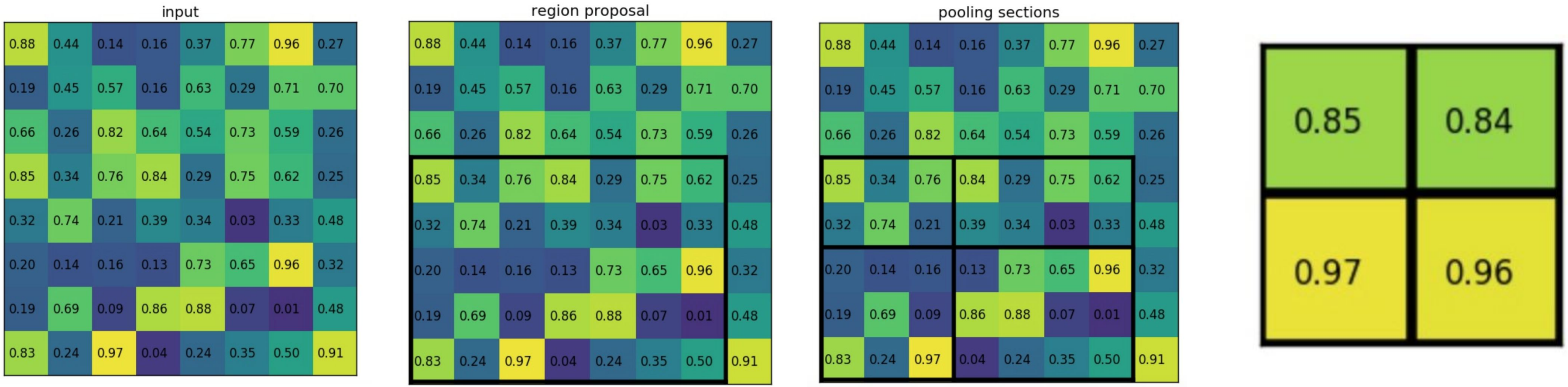
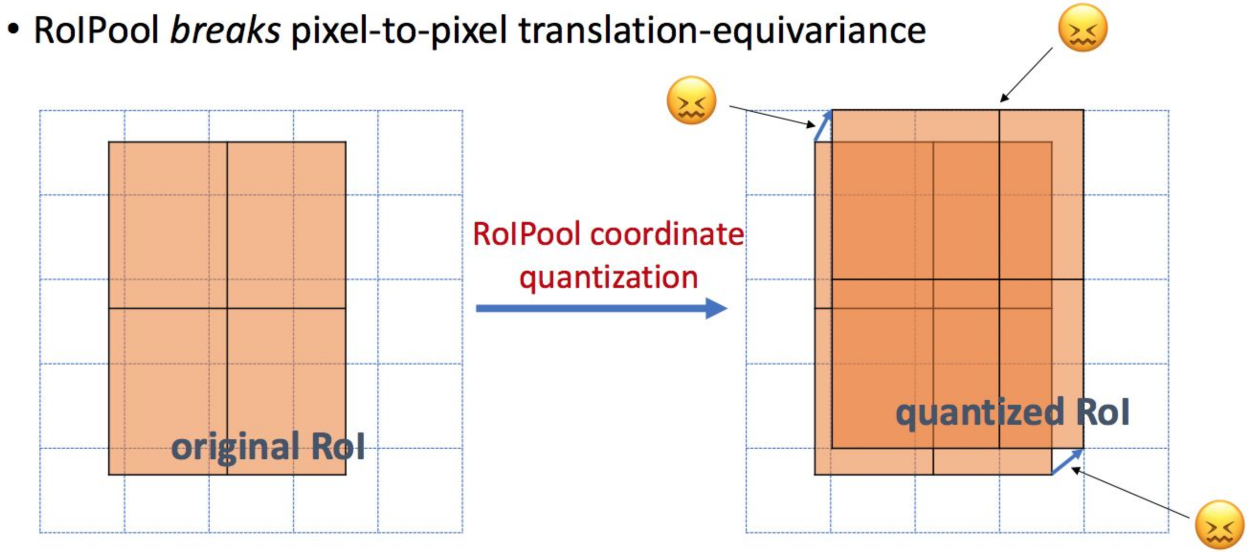


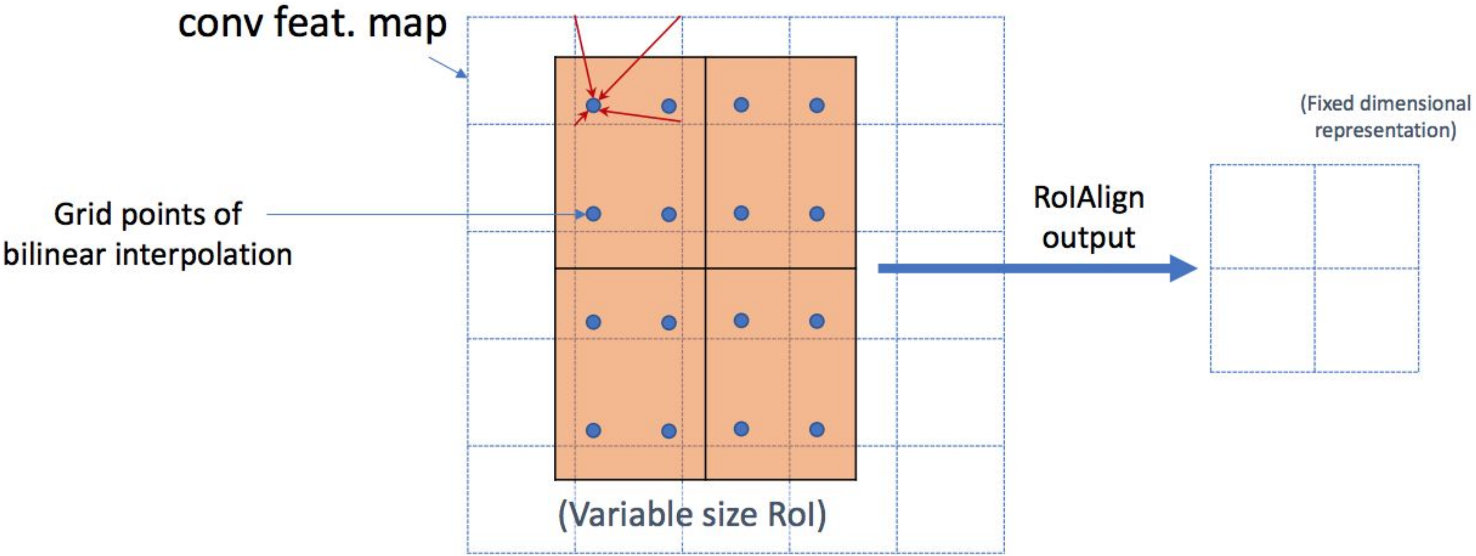
Image Credit: Tomasz Grel

RoIPool



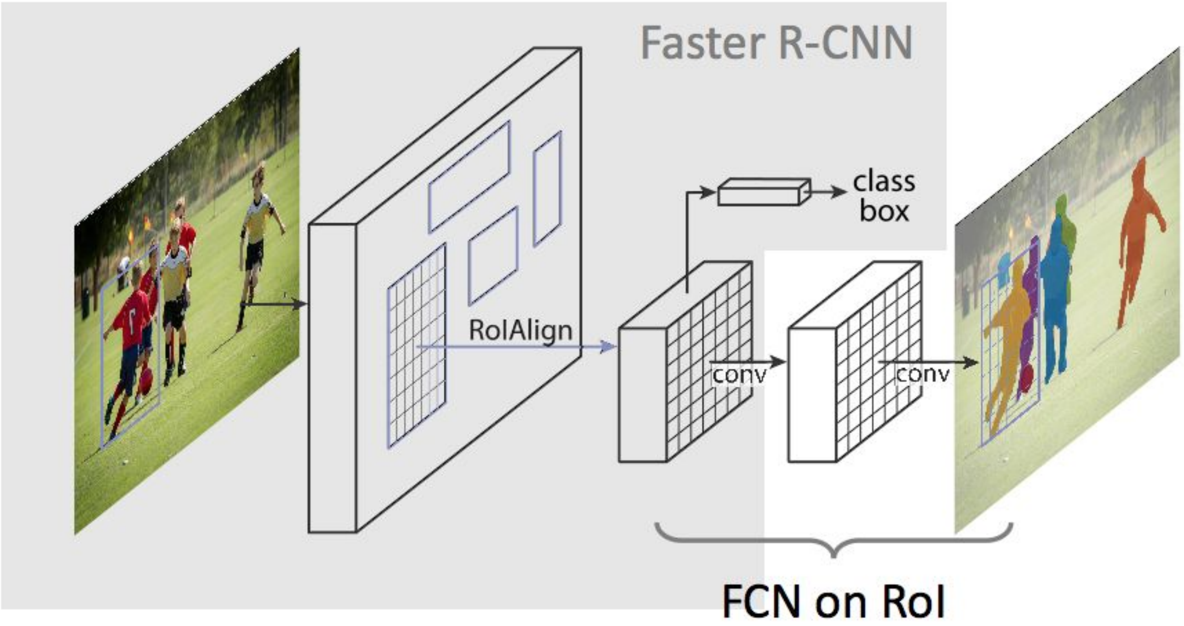
Slide Credit: Kaiming He

RoIAlign



Slide Credit: Kaiming He

Mask R-CNN



Mask R-CNN Results

Examples

- Mask AP = 35.7

Figure 5. More results of **Mask R-CNN** on COCO test images, using ResNet-101-FPN and running at 5 fps, with 35.7 mask AP (Table 1).

Image Credit: Kaiming He et al.

Comparisons

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MNC [10]	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [26] +OHEM	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [26] +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**

Image Credit: Kaiming He et al.

Comparisons

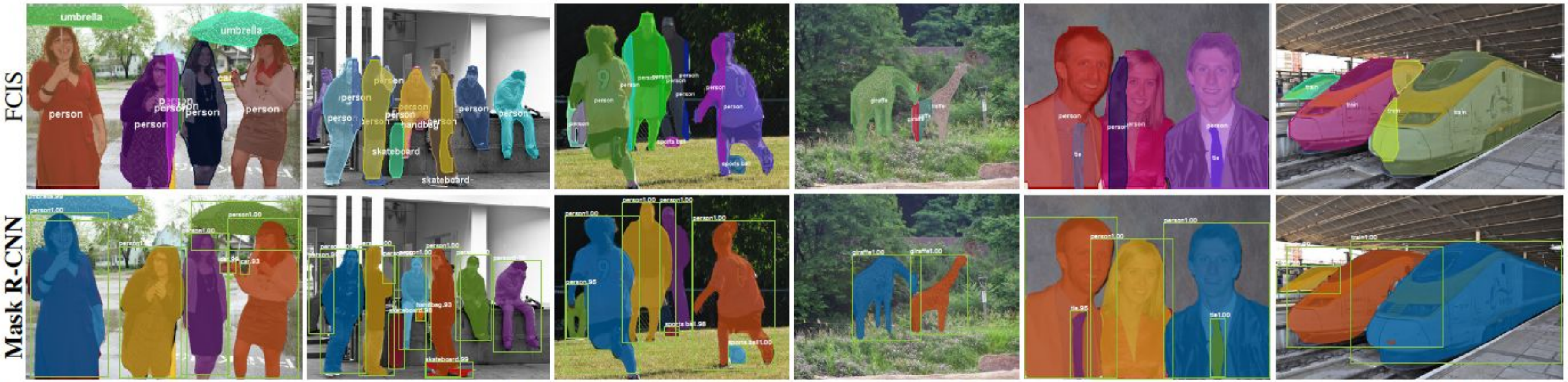


Figure 6. FCIS+++ [26] (top) *vs*. Mask R-CNN (bottom, ResNet-101-FPN). FCIS exhibits systematic artifacts on overlapping objects.

Image Credit: Kaiming He et al.

Application: Human Pose Estimation

Figure 7. Keypoint detection results on COCO test using Mask R-CNN (ResNet-50-FPN), with person segmentation masks predicted from the same model. This model has a keypoint AP of 63.1 and runs at 5 fps.

Image Credit: Kaiming He et al.

Mask R-CNN Recap

- Add parallel mask prediction head to Faster-RCNN
- RoIAlign allows for precise localization
- Mask R-CNN improves on AP of previous state-of-the-art, can be applied in human pose estimation

Learning to Segment Every Thing

Ronghang Hu, Piotr Dollar, Kaiming He, Trevor Darrell, and Ross Girshick. “Learning to Segment Every Thing.” arXiv, 2017.

Partially Supervised Model

Motivation for a Partially Supervised Model

A = set of object categories with complete mask annotations

B = set of object categories with only bounding boxes (no segmentation annotations)

How can we know C = A U B?

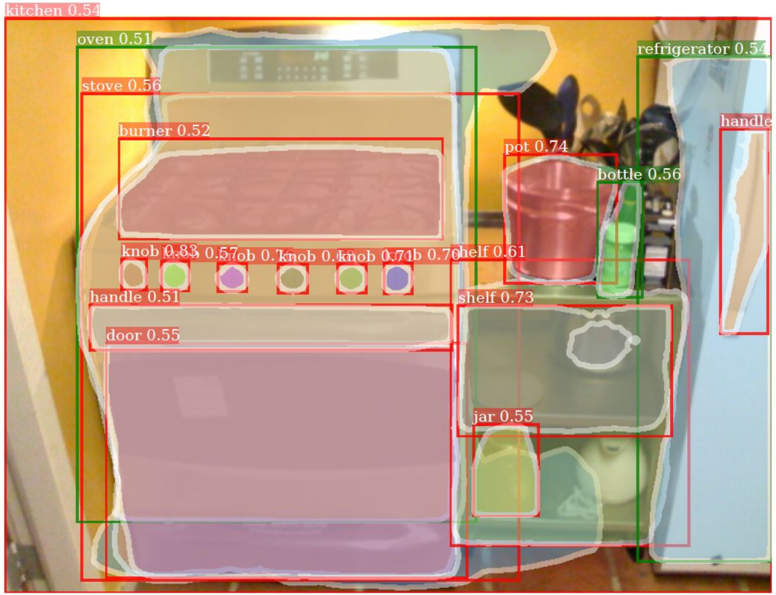


Image Credit: Ronghang Hu et al.

Transfer Learning

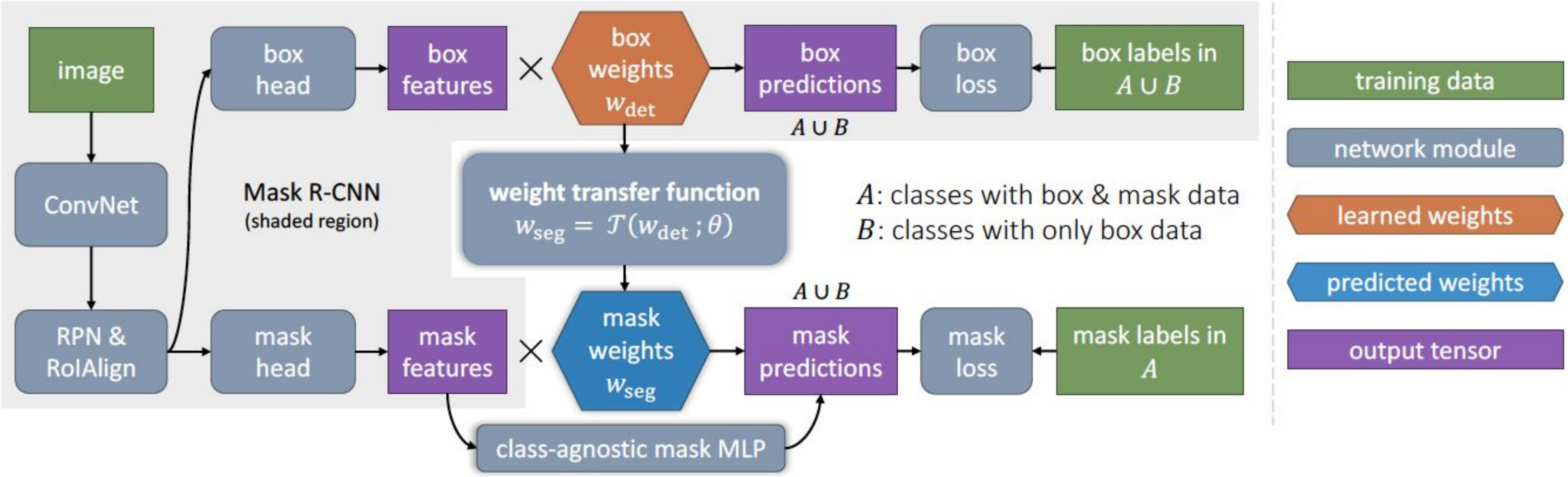
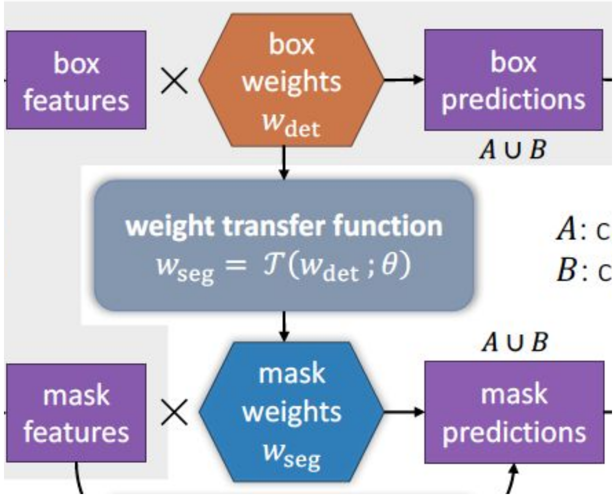


Image Credit: Ronghang Hu et al.

Weight Transfer Function



$$w^c_{\text{seg}} = \mathcal{T}(w^c_{\text{det}}; \theta)$$

$$w^c_{\text{det}} = w^c_{\text{box}}$$

$$w^c_{\text{det}} = w^c_{\text{cls}}$$

$$w^c_{\text{det}} = [w^c_{\text{cls}}, w^c_{\text{box}}]$$

Image Credit: Ronghang Hu et al.

Training

- Train bounding box head using standard box detection losses on all classes in A U B
- Train mask head, weight transfer function using mask loss on classes in A

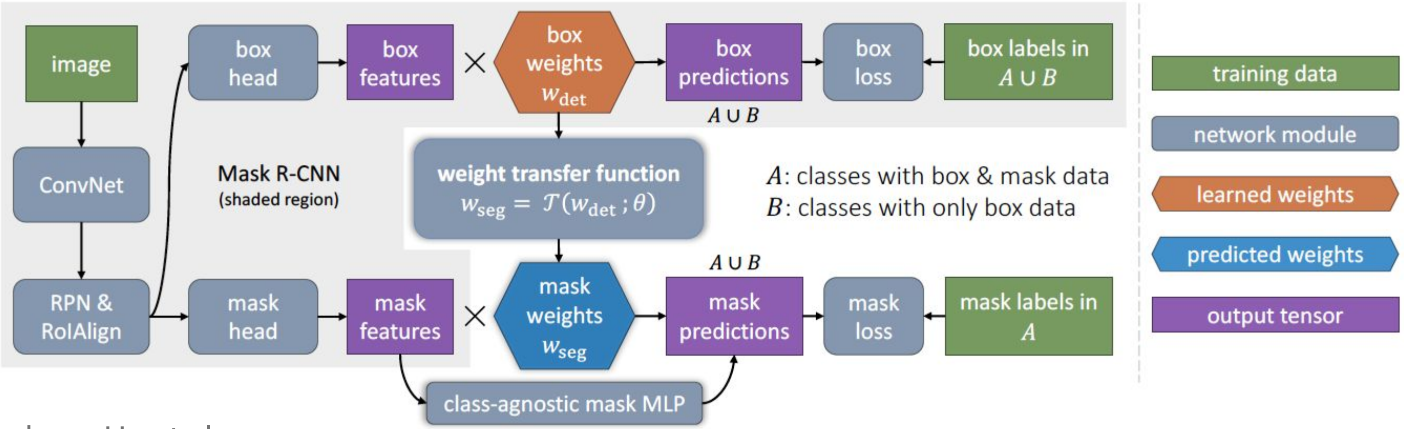


Image Credit: Ronghang Hu et al.

Stage-Wise Training

1. Detection training
2. Segmentation training

- Train detection once and then fine-tune weight transfer function
- Inferior performance

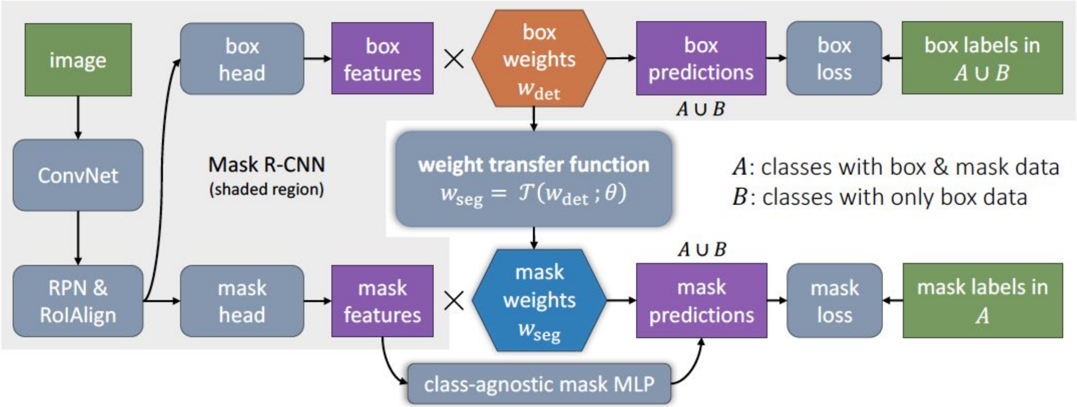
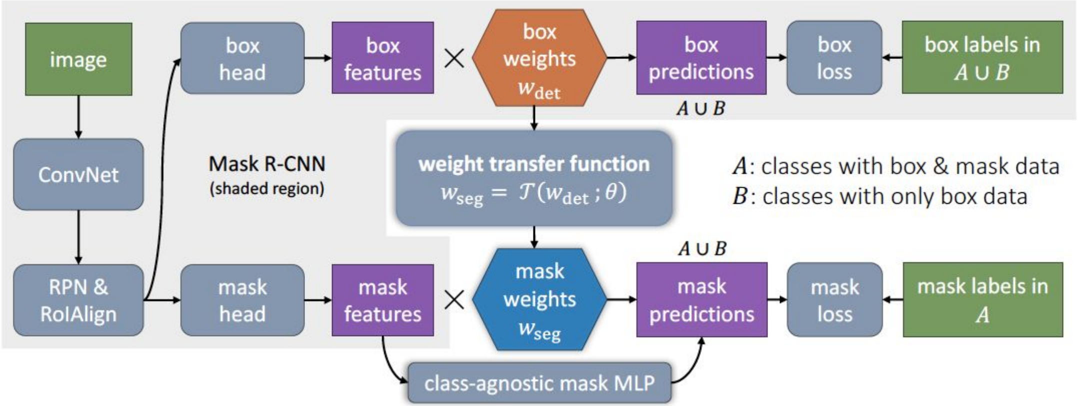


Image Credit: Ronghang Hu et al.

End-to-End Joint Training

- Jointly train detection head and mask head end-to-end
- Want detection weights to stay constant between A and B



$$w_{seg}^{c} = \mathcal{T}(\texttt{stop\_grad}(w_{det}^{c}); \theta)$$

Image Credit: Ronghang Hu et al.

End-to-End Training Better

method	training	stop grad on w_{det}	voc → non-voc AP on B	voc → non-voc AP on A	non-voc → voc AP on B	non-voc → voc AP on A
class-agnostic	sw	n/a	14.2	34.4	21.5	30.7
transfer	sw	n/a	20.2	35.2	26.0	31.2
class-agnostic	e2e	n/a	19.2	36.8	23.9	32.5
transfer	e2e		20.2	37.7	24.8	33.2
transfer	e2e	✓	**22.2**	37.6	**27.6**	33.1

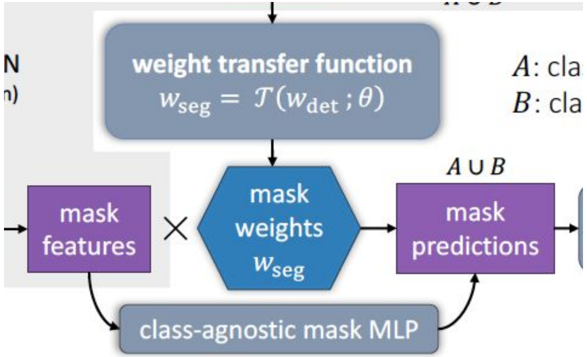
(d) **Ablation on the training strategy.** We try both stage-wise ('sw') and end-to-end ('e2e') training (see §3.2), and whether to stop gradient from $\mathcal{T}$ to w_{det}. End-to-end training improves the results and it is crucial to stop gradient on w_{det}.

Image Credit: Ronghang Hu et al.

Mask Prediction

Baseline: Class-agonistic FCN mask prediction

Extension: FCN+MLP mask head



method	voc → non-voc AP on B	voc → non-voc AP on A	non-voc → voc AP on B	non-voc → voc AP on A
class-agnostic	14.2	34.4	21.5	30.7
class-agnostic+MLP	17.1	35.1	22.8	31.3
transfer	20.2	35.2	26.0	31.2
transfer+MLP	**21.3**	35.4	**26.6**	31.4

(c) **Impact of the MLP mask branch.** Adding the class-agnostic MLP mask branch (see §3.4) improves the performance of classes in set B for both the class-agnostic baseline and our weight transfer approach.

Image Credit: Ronghang Hu et al.

Results

Examples

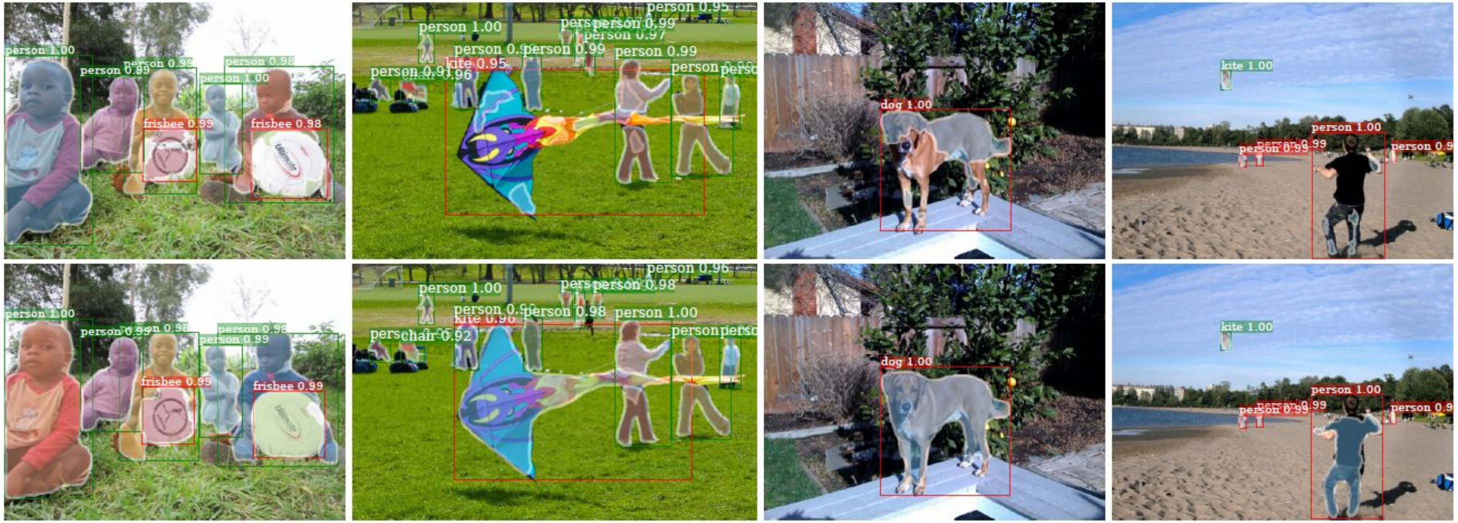


Figure 4. **Mask predictions from the class-agnostic baseline (top row) *vs.* our Mask^X R-CNN approach (bottom row).** Green boxes are classes in set A while the red boxes are classes in set B. The left 2 columns are $A = \{\text{voc}\}$ and the right 2 columns are $A = \{\text{non-voc}\}$.

Image Credit: Ronghang Hu et al.

Comparisons

backbone	method	voc → non-voc: test on B = {non-voc} AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	non-voc → voc: test on B = {voc} AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
R-50-FPN	class-agnostic (baseline)	19.2	36.4	18.4	11.5	23.3	24.4	23.9	42.9	23.5	11.6	24.3	33.7
	Mask^X R-CNN (ours)	**23.7**	**43.1**	**23.5**	**12.4**	**27.6**	**32.9**	**28.9**	**52.2**	**28.6**	**12.1**	**29.0**	**40.6**
	fully supervised (oracle)	33.0	53.7	35.0	15.1	37.0	49.9	37.5	63.1	38.9	15.1	36.0	53.1
R-101-FPN	class-agnostic (baseline)	18.5	34.8	18.1	11.3	23.4	21.7	24.7	43.5	24.9	11.4	25.7	35.1
	Mask^X R-CNN (ours)	**23.8**	**42.9**	**23.5**	**12.7**	**28.1**	**33.5**	**29.5**	**52.4**	**29.7**	**13.4**	**30.2**	**41.0**
	fully supervised (oracle)	34.4	55.2	36.3	15.5	39.0	52.6	39.1	64.5	41.4	16.3	38.1	55.1

Table 2. **End-to-end training of Mask^X R-CNN.** As in Table 1, we use 'cls+box, 2-layer, LeakyReLU' implementation of $\mathcal{T}$ and add the MLP mask branch ('transfer+MLP'), and follow the same evaluation protocol. We also report AP_{50} and AP_{75} (average precision evaluated at 0.5 and 0.75 IoU threshold respectively), and AP over small (AP_S), medium (AP_M), and large (AP_L) objects. Our method significantly outperforms the baseline on those classes in set B without mask training data for both ResNet-50-FPN and ResNet-101-FPN backbones.

Image Credit: Ronghang Hu et al.

Segmenting Everything

Figure 5. **Example mask predictions from our Mask^X R-CNN on 3000 classes in Visual Genome.** The green boxes are the 80 classes that overlap with COCO (set A with mask training data) while the red boxes are the remaining 2920 classes not in COCO (set B without mask training data). It can be seen that our model generates reasonable mask predictions on many classes in set B. See §5 for details.

Image Credit: Ronghang Hu et al.