Object Detection

Xiaolong Wang

This Class: Object Detection

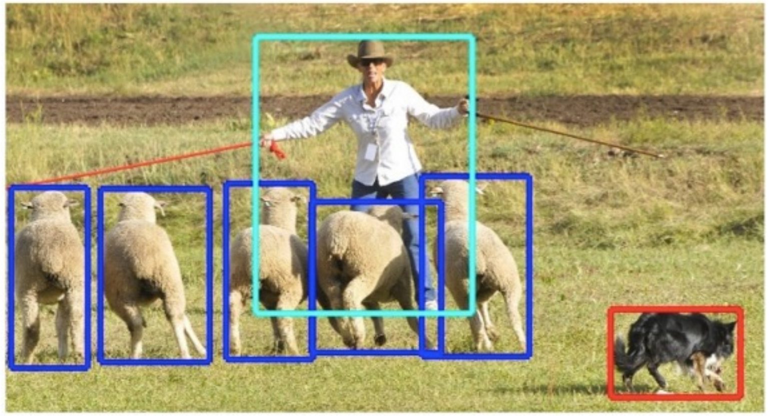
- Background and old fashion object detection
- 2-stage object detection
- FPN , Mask R-CNN and more

Slides partially from: http://cs231n.stanford.edu/ https://slazebni.cs.illinois.edu/fall20/

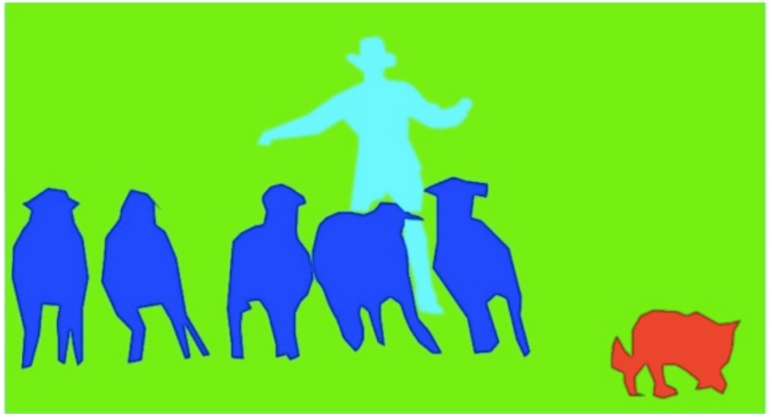
Background and old fashion object detection

The task: Object Detection

image classification



object detection

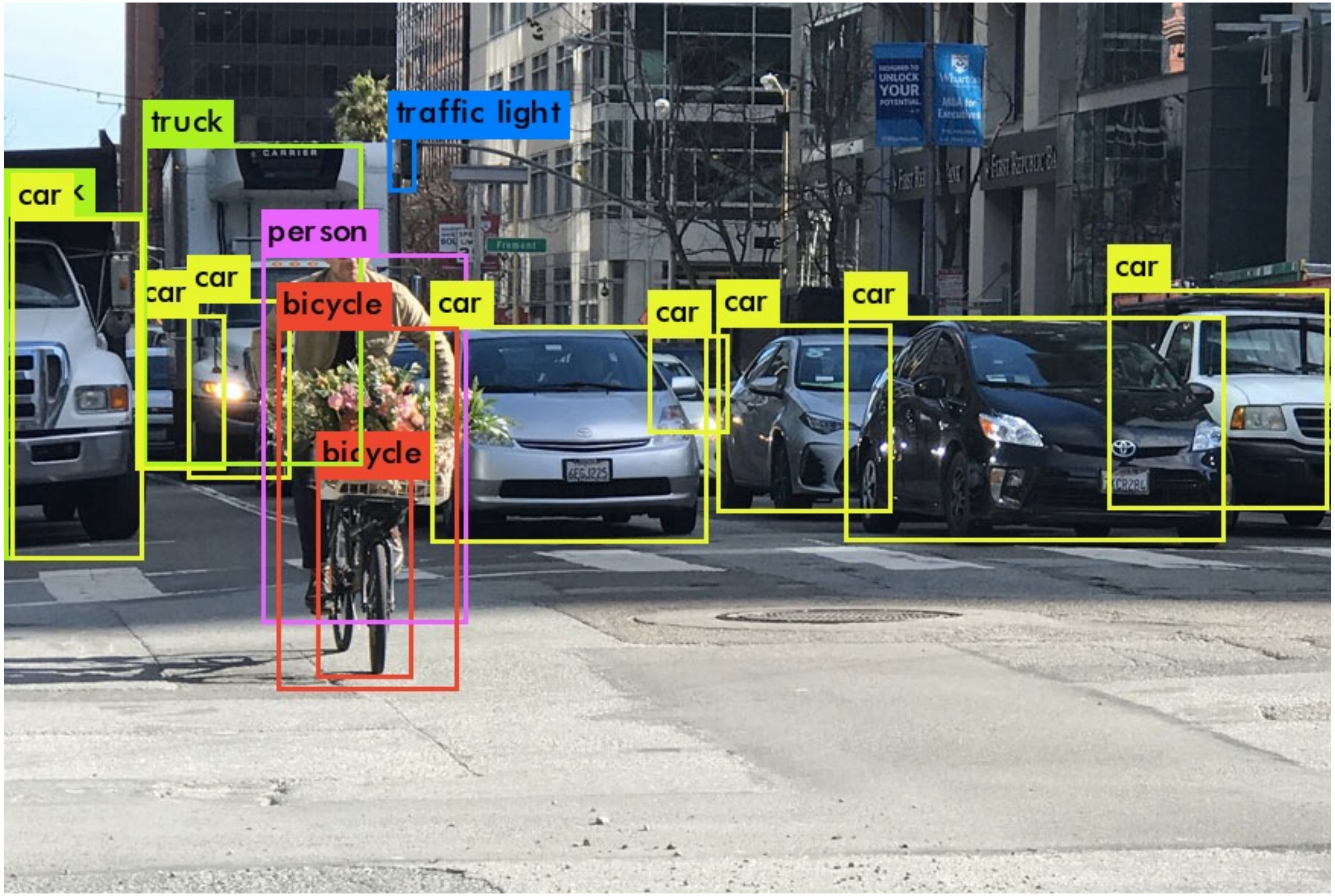


semantic segmentation

instance segmentation

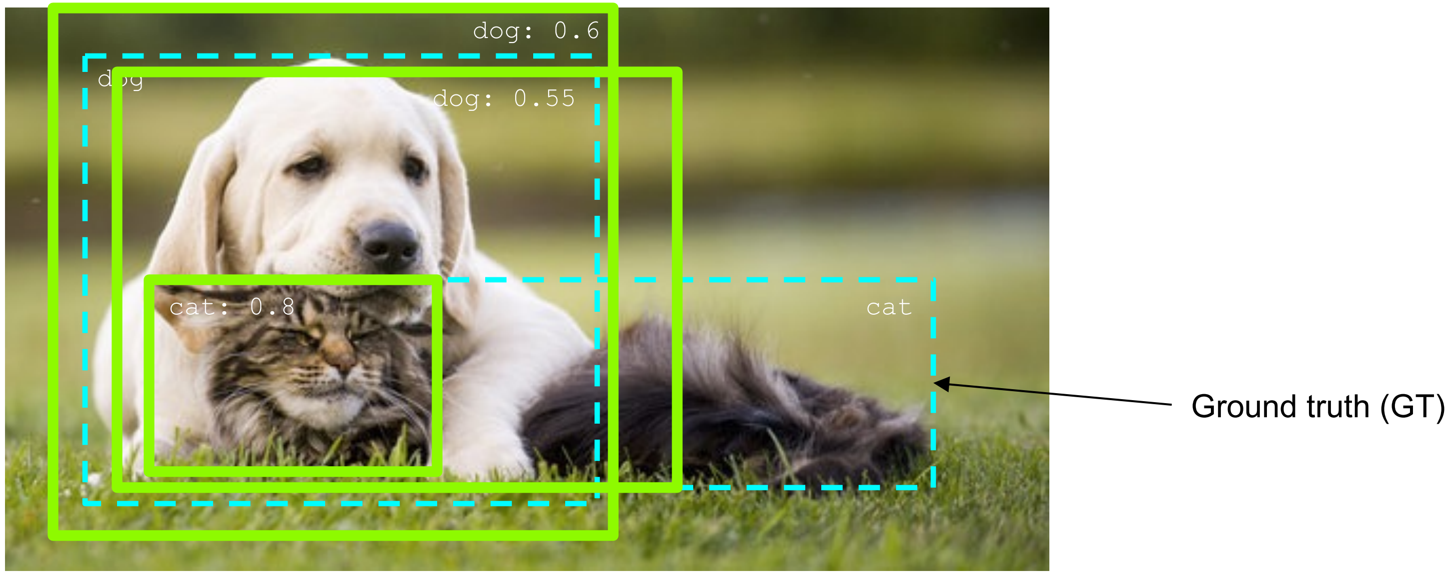
The task: Object Detection

Images may contain more than one class, multiple instances from the same class



Evaluation

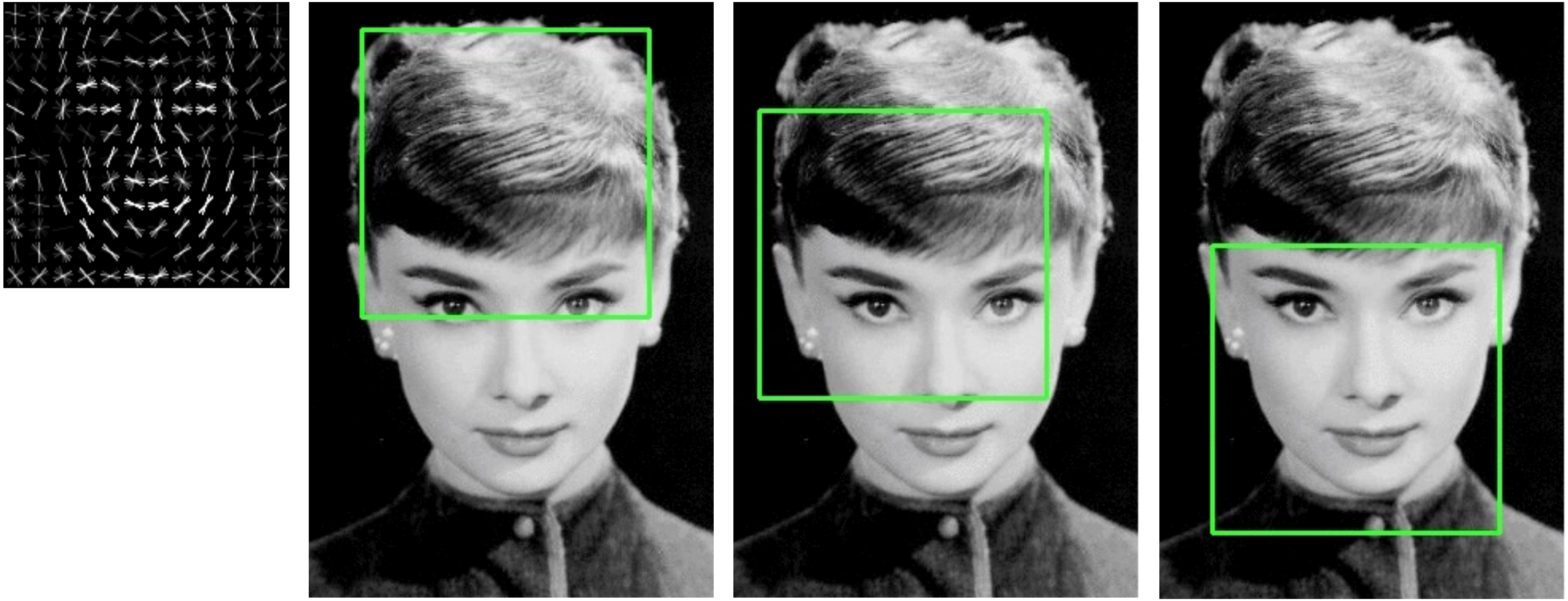
- At test time, predict bounding boxes, class labels, and confidence
- For each detection, determine whether it is a true or false positive
  - PASCAL criterion: Area(GT ∩ Det) / Area(GT ∪ Det) > 0.5
  - For multiple detections of the same ground truth box, only one is considered a true positive



Evaluation

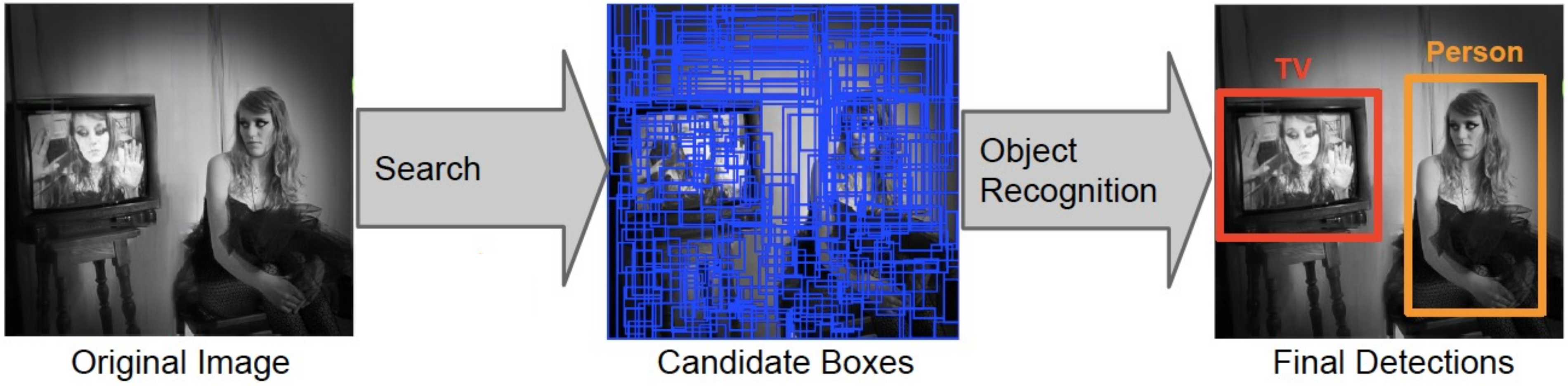
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Sliding window approach for detection



Histograms of Oriented Gradients. Dalal et al. 2005

Object proposal for object detection



- First generate a lot of region proposals (using low-level cues)
- Classification on each proposal

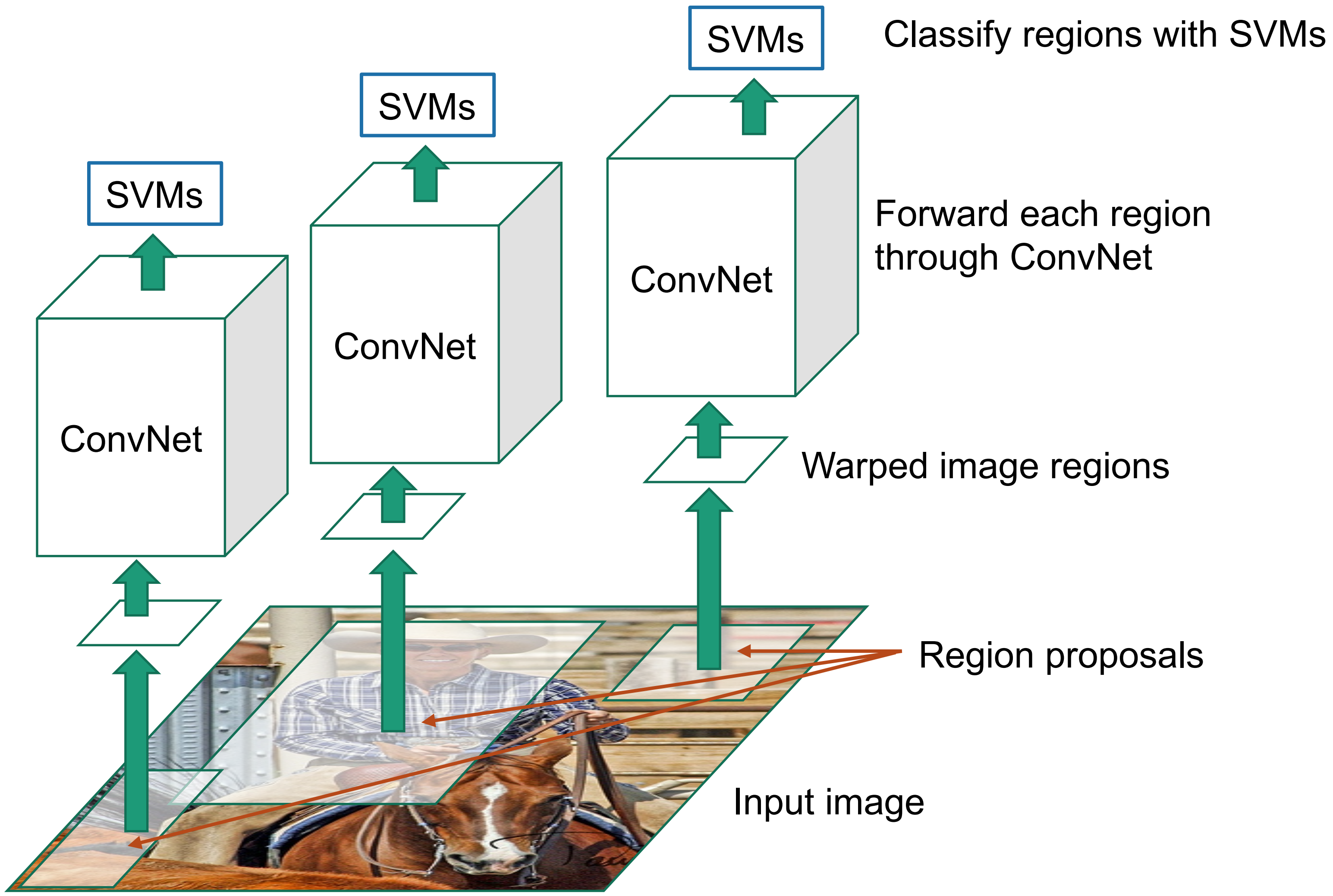
Uijlings et al., 2013

Selective search to generate object proposal for object detection

- Use hierarchical segmentation: start with small *superpixels* and merge based on diverse cues

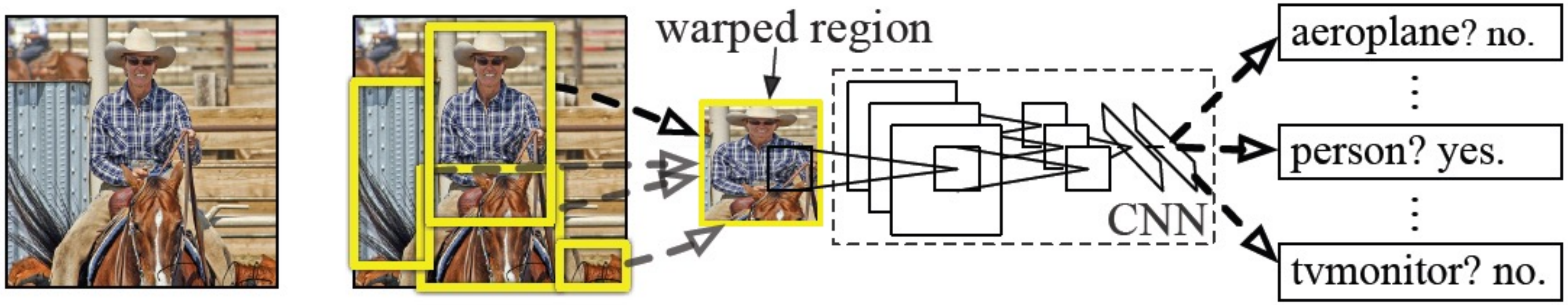
2-stage object detection

R-CNN: Region proposals + CNN



Girshick, et al, 2014

R-CNN: Region proposals + CNN

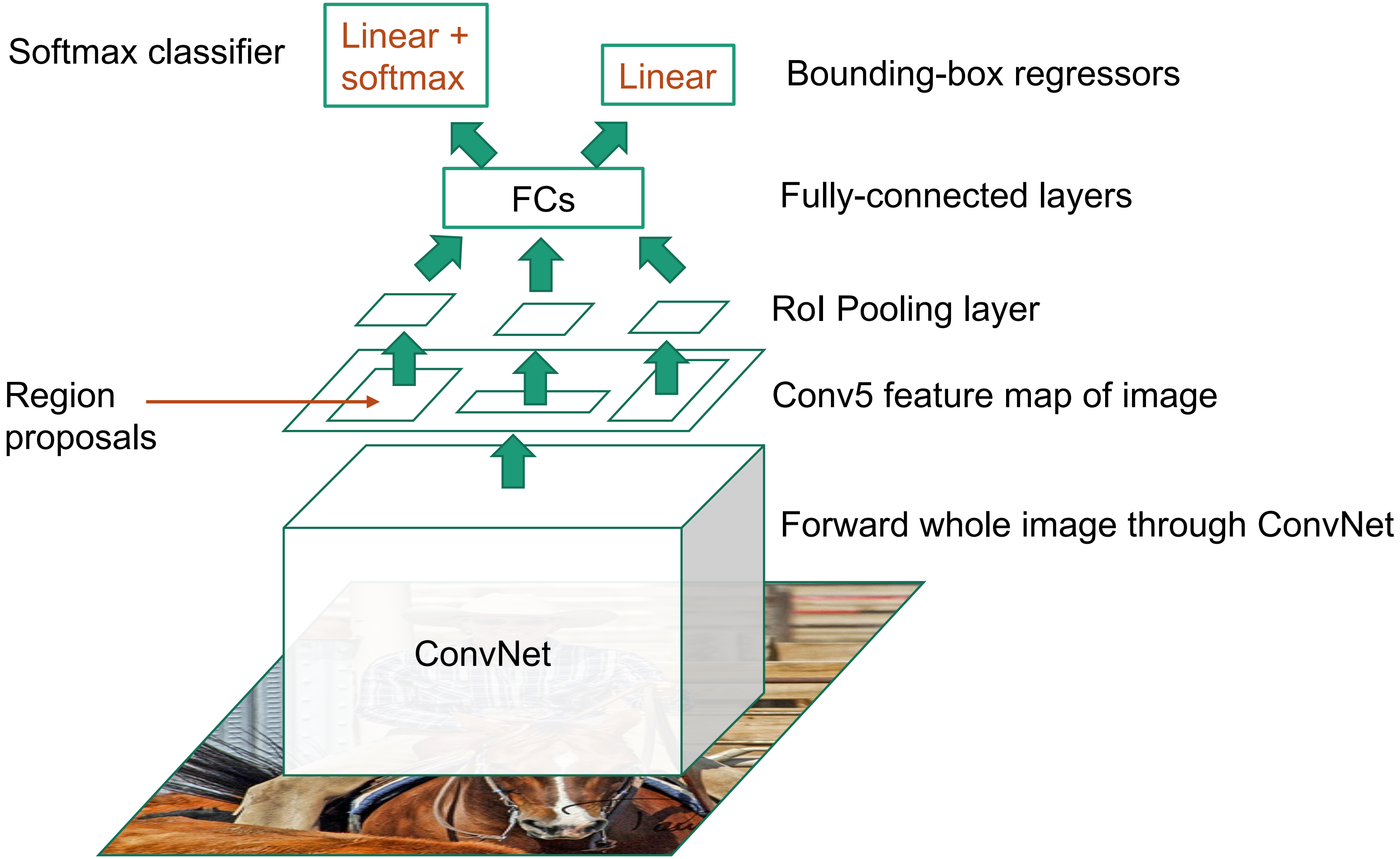


- **Regions**: ~2000 Selective Search proposals
- **Network**: AlexNet *pre-trained* on ImageNet (1000 classes), *fine-tuned* on PASCAL (21 classes)
- **Final detector**: warp proposal regions, extract fc7 network activations (4096 dimensions), classify with linear SVM
- **Bounding box regression** to refine box locations
- **Performance:** mAP of **53.7%** on PASCAL 2010 (vs. **35.1%** for Selective Search and **33.4%** for Deformable Part Models)

R-CNN: Region proposals + CNN

- Pros
  - Much more accurate than previous approaches!
  - Any deep architecture can immediately be "plugged in"
- Cons
  - Not a single end-to-end system
    - Fine-tune network with softmax classifier (log loss)
    - Train post-hoc linear SVMs (hinge loss)
  - Training was slow (84h), took up a lot of storage
    - 2000 CNN passes per image
  - Inference (detection) was slow (47s / image with VGG16)

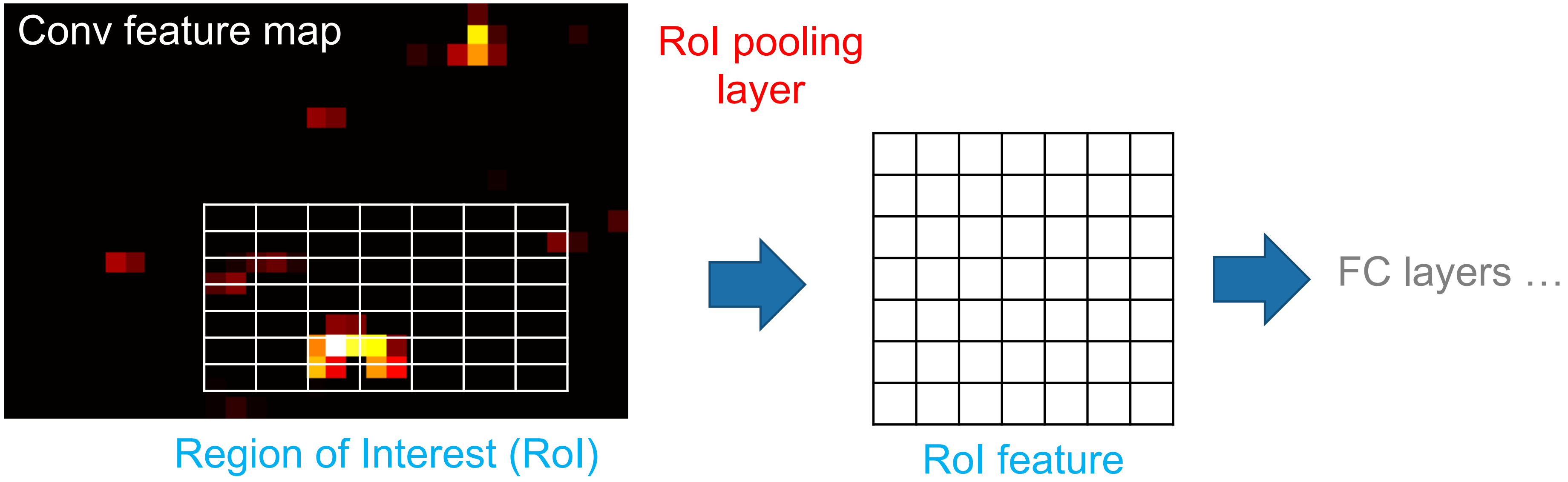
Fast R-CNN



R. Girshick, [Fast R-CNN](), ICCV 2015

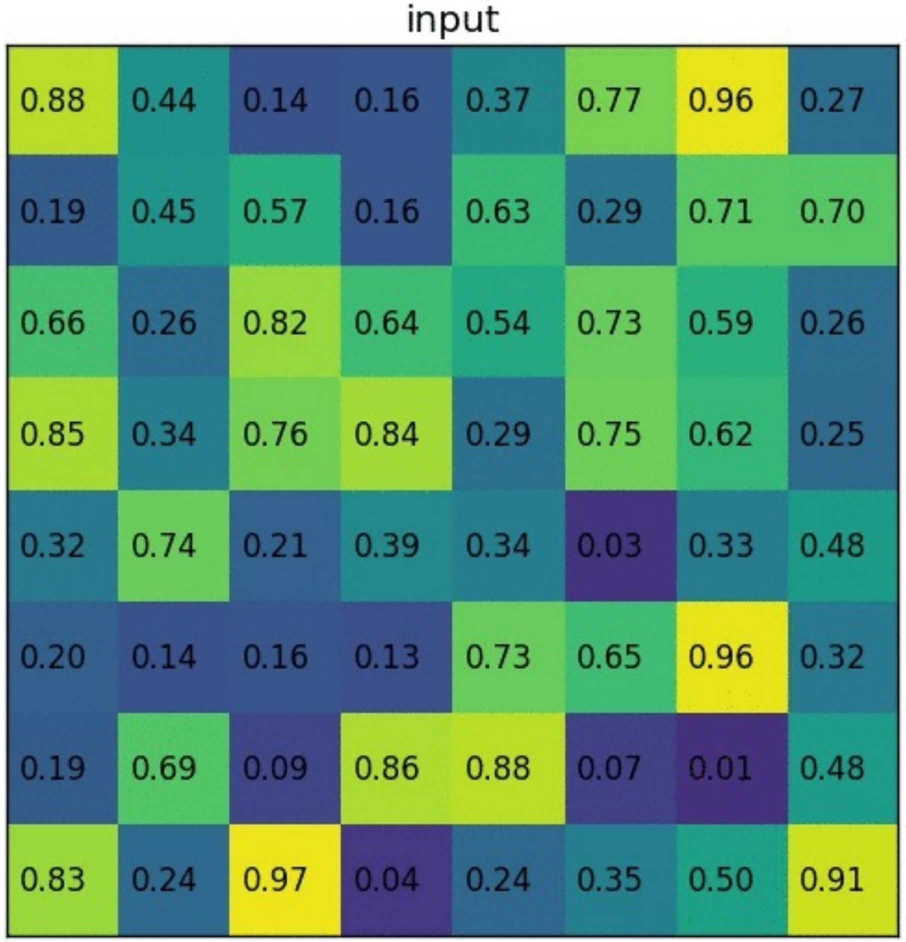
RoI pooling

"Crop and resample" a fixed-size feature representing a region of interest out of the outputs of the last conv layer



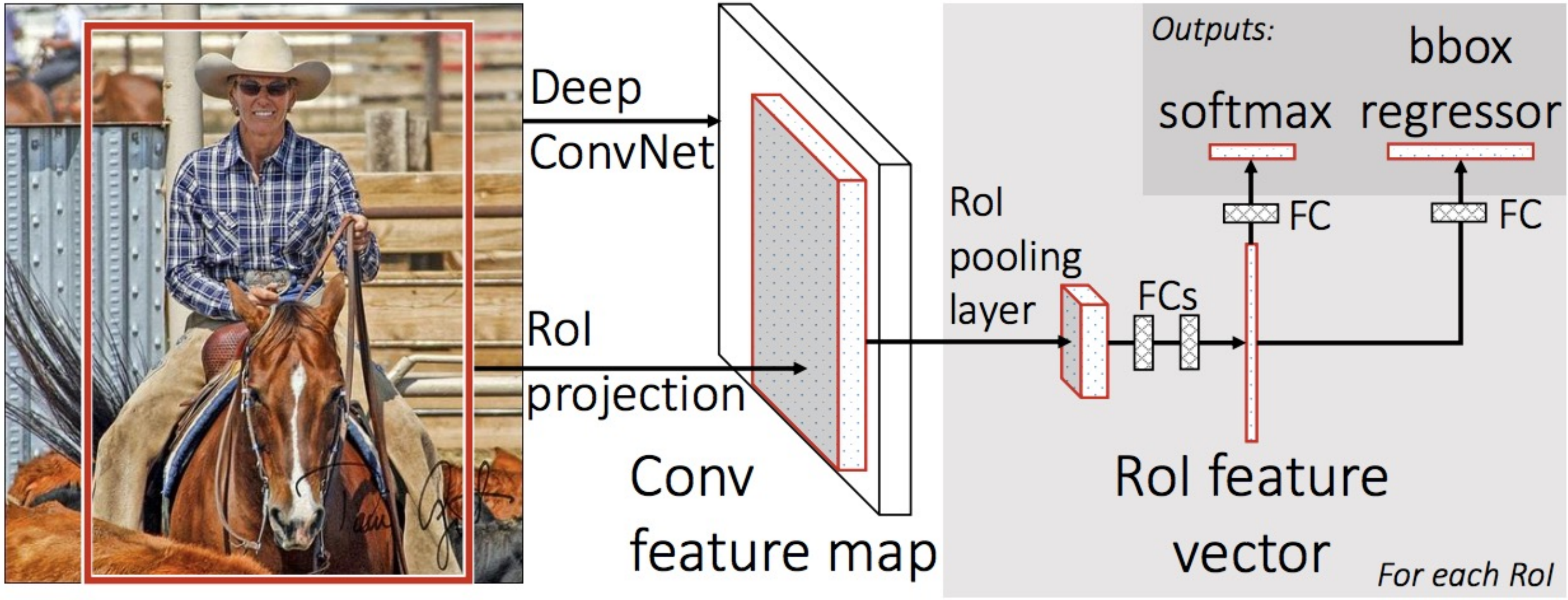
Source: R. Girshick, K. He

RoI pooling

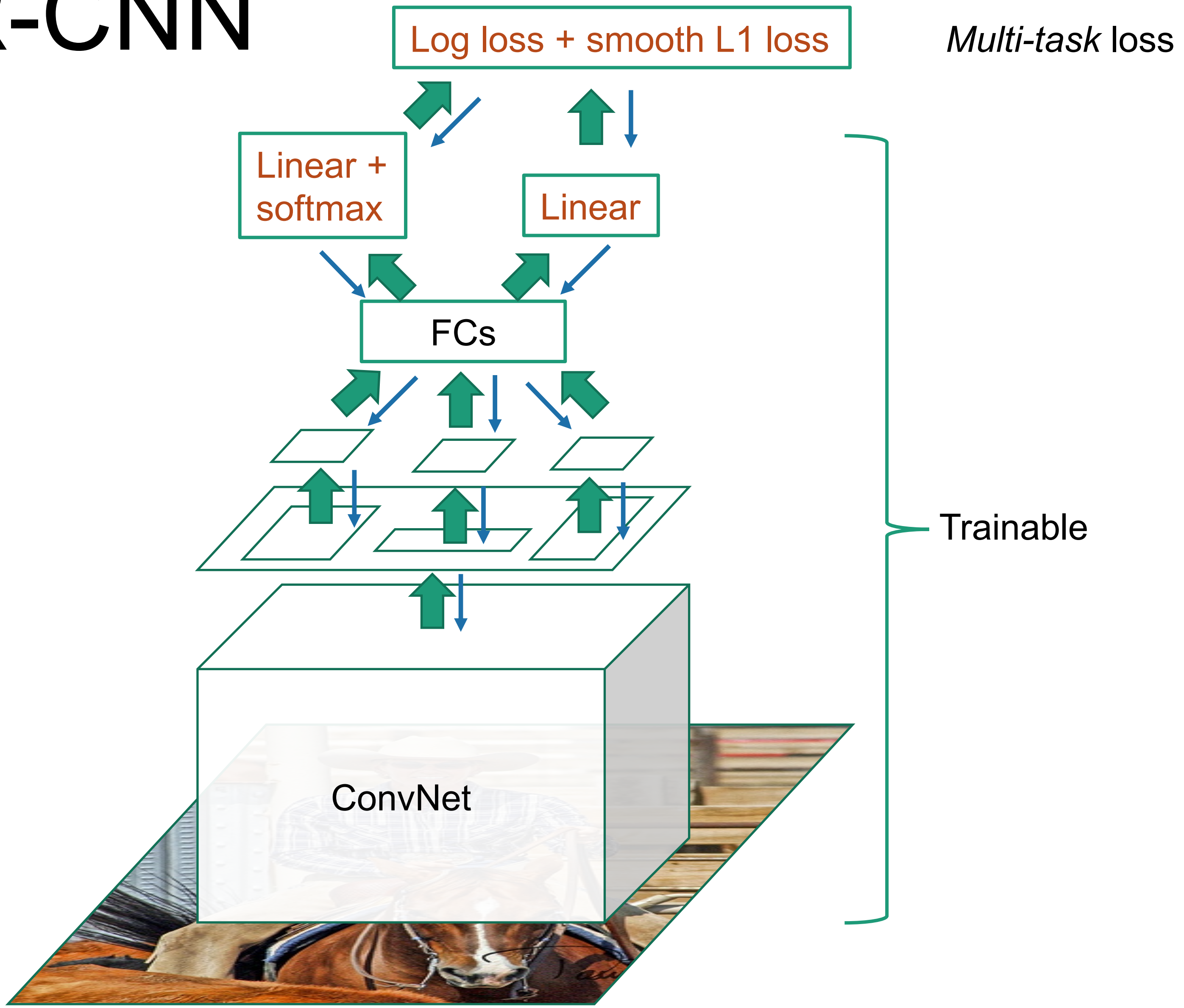

input
0.88 0.44 0.14 0.16 0.37 0.77 0.96 0.27
0.19 0.45 0.57 0.16 0.63 0.29 0.71 0.70
0.66 0.26 0.82 0.64 0.54 0.73 0.59 0.26
0.85 0.34 0.76 0.84 0.29 0.75 0.62 0.25
0.32 0.74 0.21 0.39 0.34 0.03 0.33 0.48
0.20 0.14 0.16 0.13 0.73 0.65 0.96 0.32
0.19 0.69 0.09 0.86 0.88 0.07 0.01 0.48
0.83 0.24 0.97 0.04 0.24 0.35 0.50 0.91

Fast R-CNN

For each RoI, network predicts probabilities for $C + 1$ classes (class 0 is background) and four bounding box offsets for C classes



Fast R-CNN

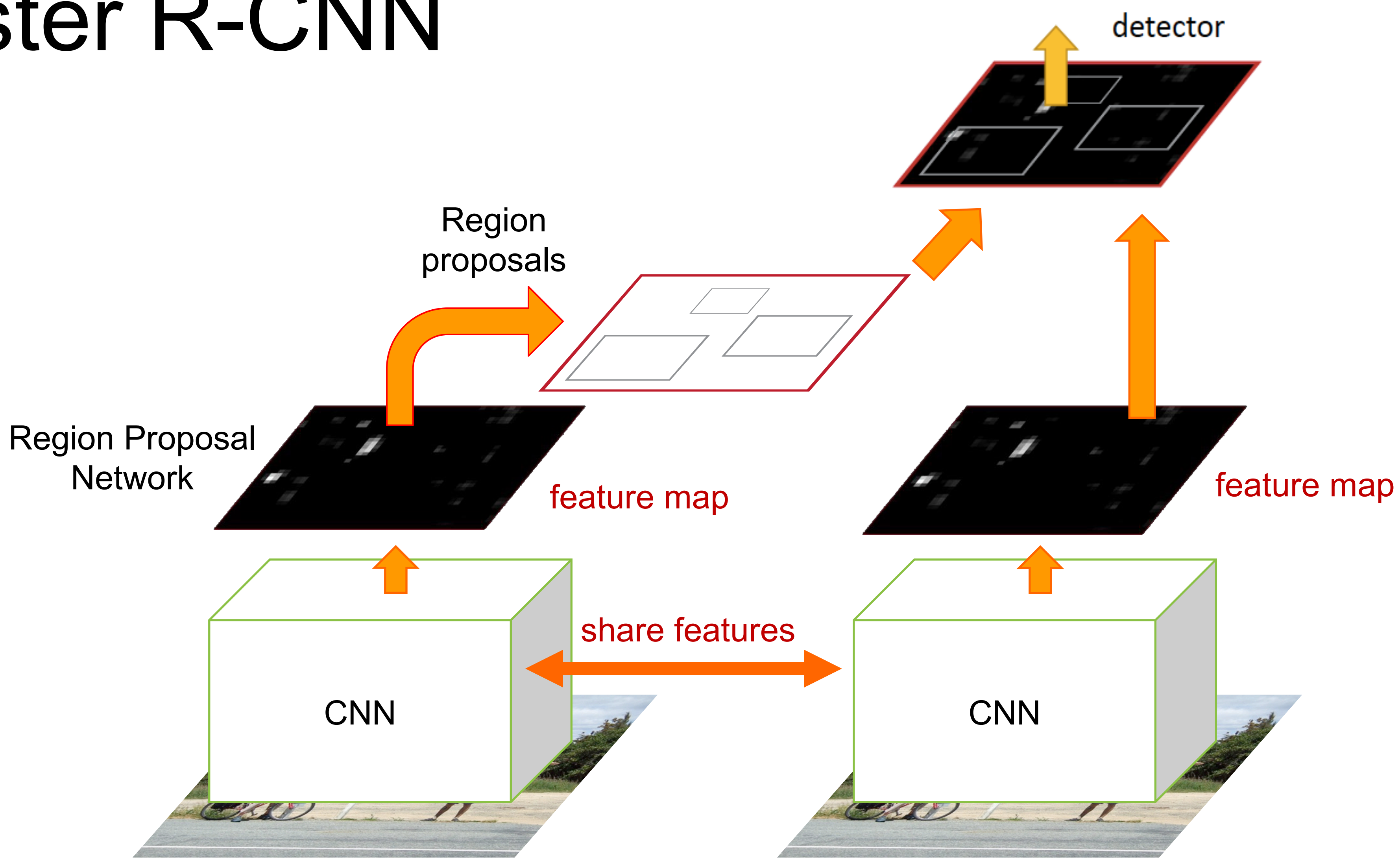


R. Girshick, [Fast R-CNN](), ICCV 2015

Fast R-CNN results with VGG16

	Fast R-CNN	**R-CNN**
Train time (h)	**9.5**	84
- Speedup	**8.8x**	
Test time / image	**0.32s**	47.0s
- Test speedup	**146x**	
mAP	**66.9%**	66.0%

Faster R-CNN

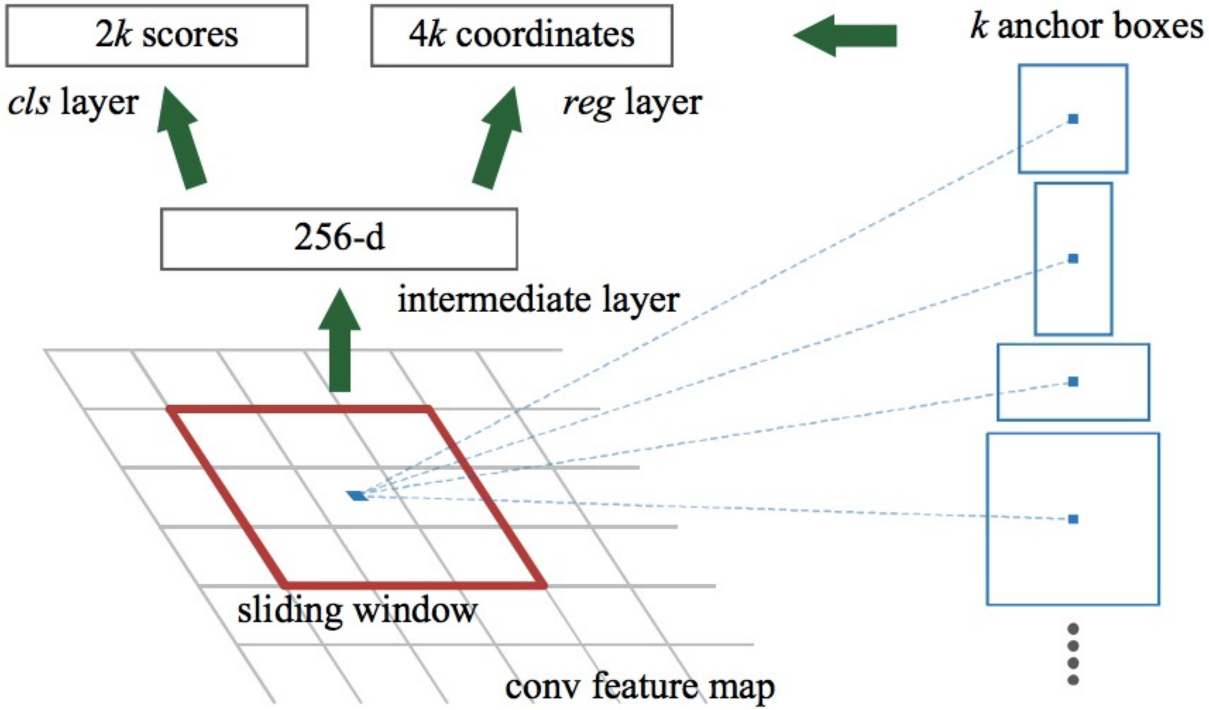


Ren, et al., NIPS 2015

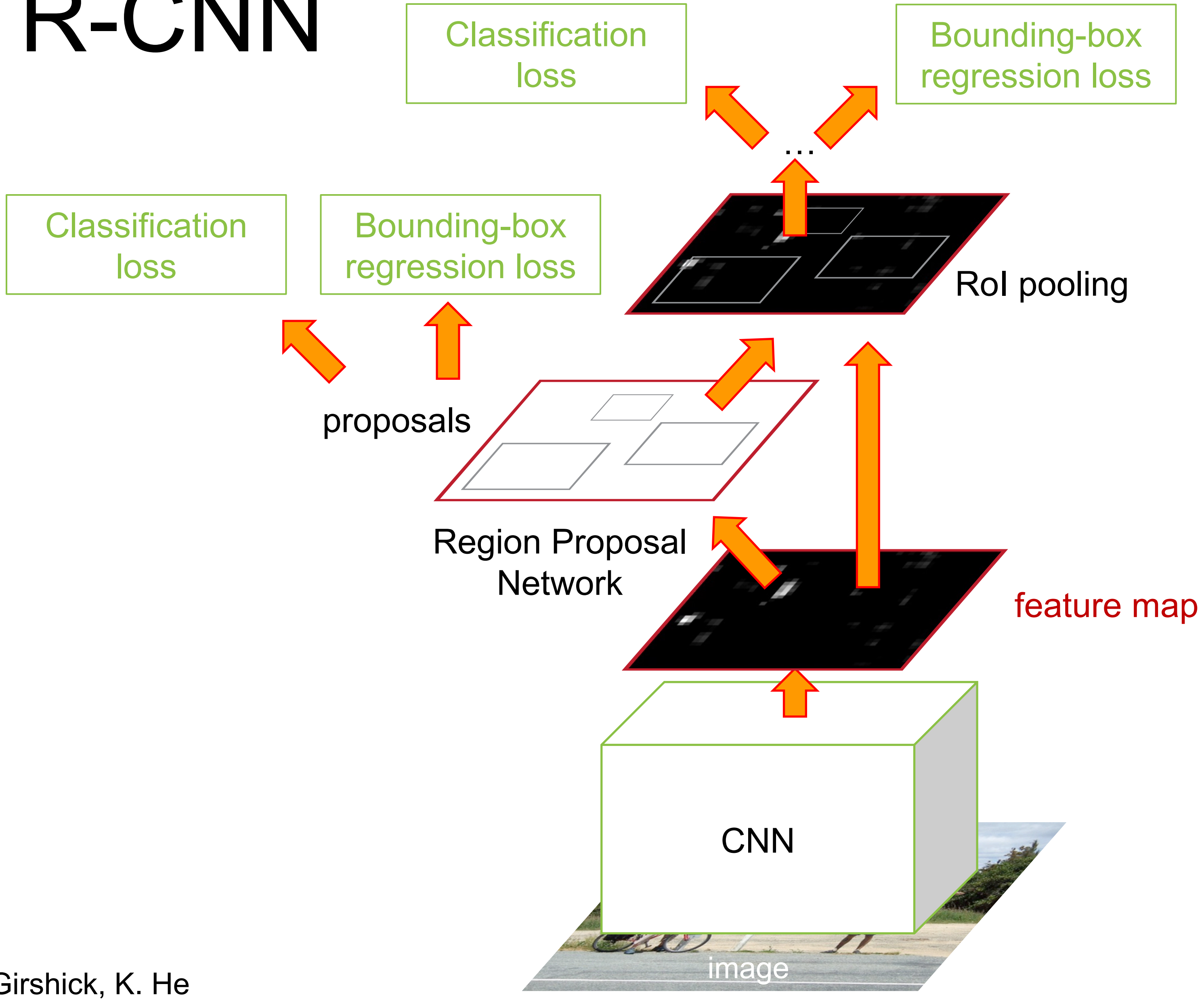
Region proposal network (RPN)

Slide a small window (3x3) over the conv5 layer

- Predict object/no object
- Regress bounding box coordinates with reference to *anchors* (3 scales x 3 aspect ratios)



Faster R-CNN



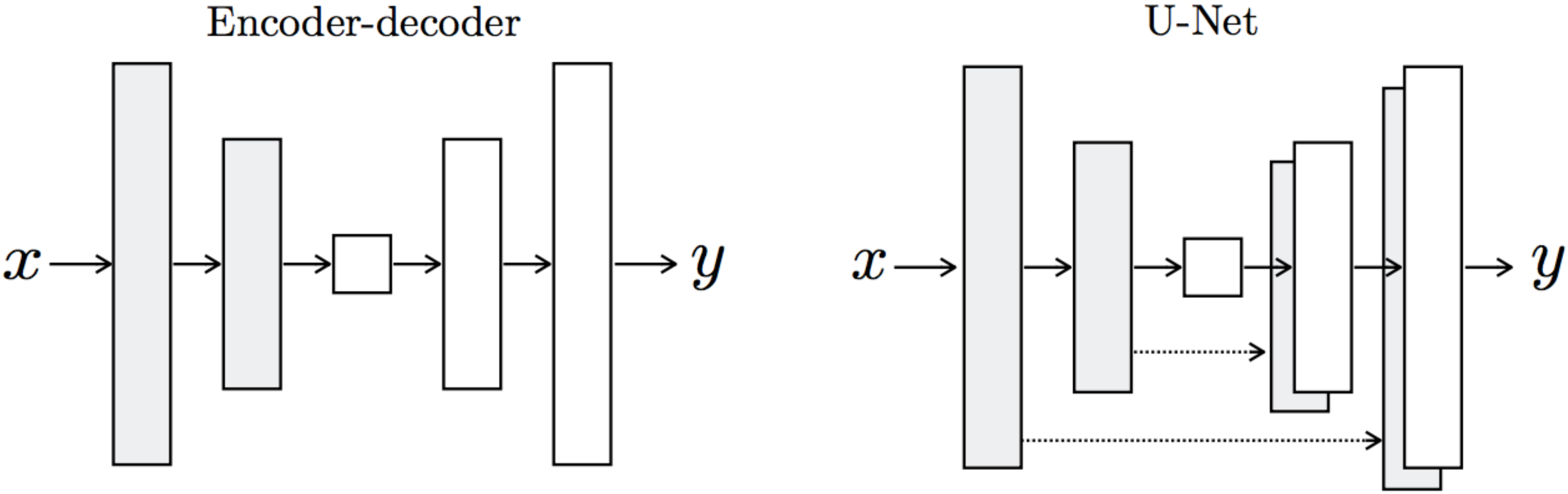
Source: R. Girshick, K. He

Faster R-CNN results

system	time	07 data	07+12 data
R-CNN	~50s	66.0	-
Fast R-CNN	~2s	66.9	70.0
Faster R-CNN	198ms	**69.9**	**73.2**

detection mAP on PASCAL VOC 2007, with VGG-16 pre-trained on ImageNet

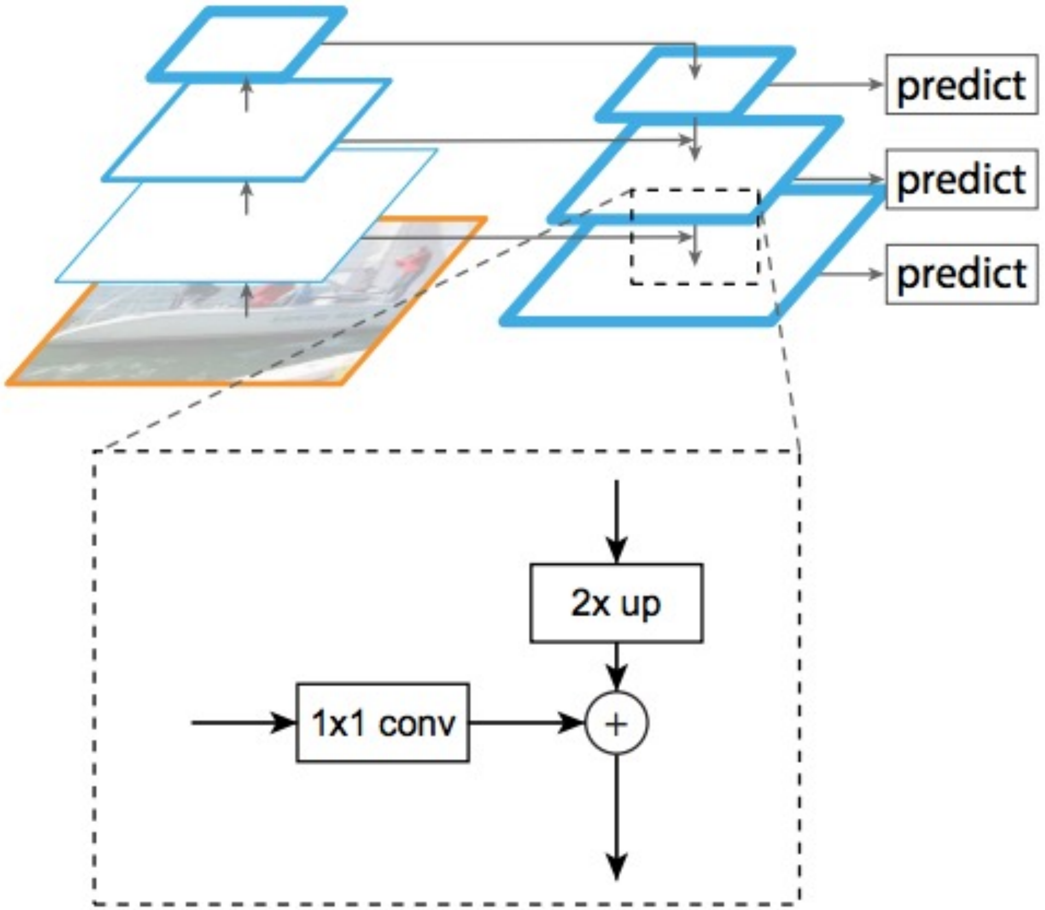
Feature pyramid networks



https://phillipi.github.io/pix2pix/

Feature pyramid networks

- Improve predictive power of lower-level feature maps by adding contextual information from higher-level feature maps
- Predict different sizes of bounding boxes from different levels of the pyramid (but share parameters of predictors)

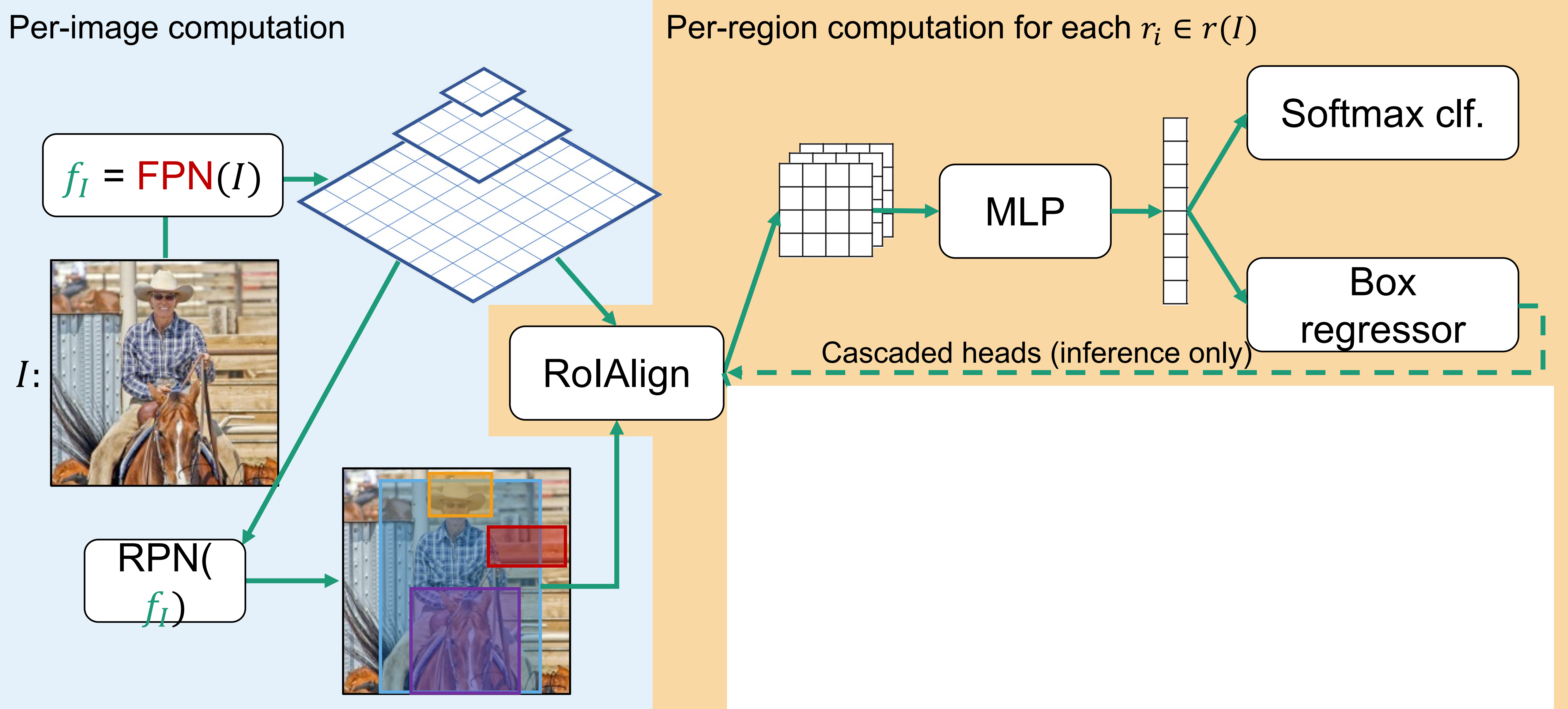


Lin et al., CVPR 2017

Mask R-CNN

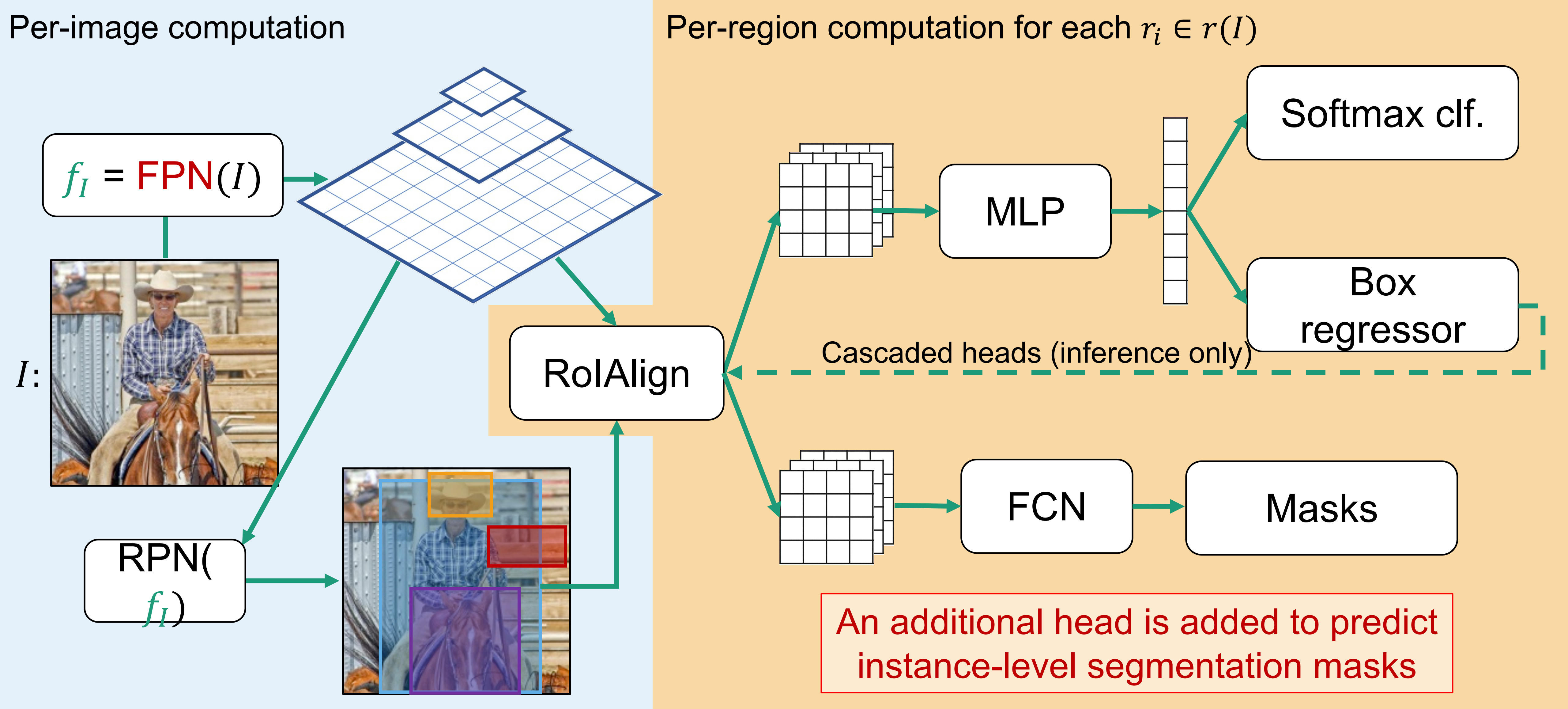
He et al. Mask R-CNN. ICCV 2017.

Mask R-CNN

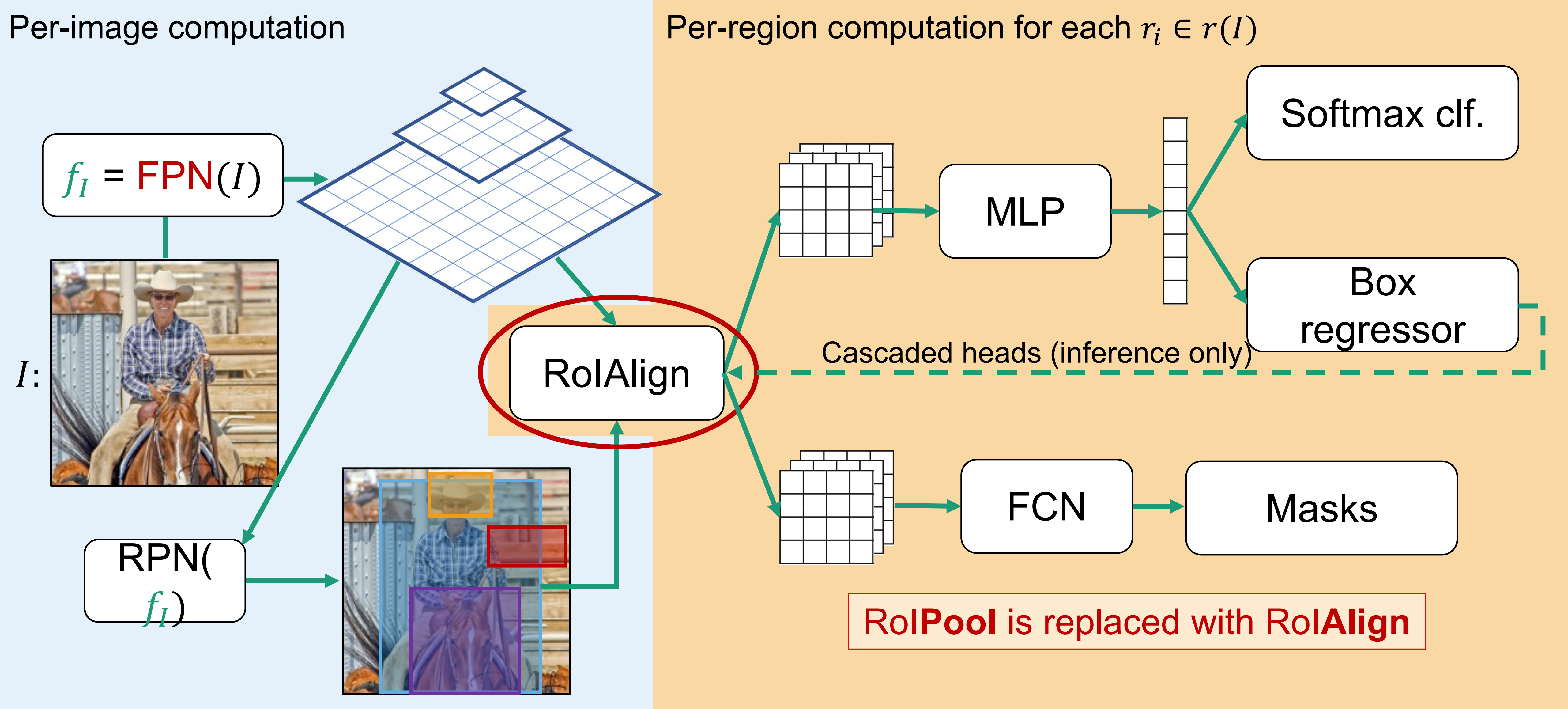


He et al. Mask R-CNN. ICCV 2017.

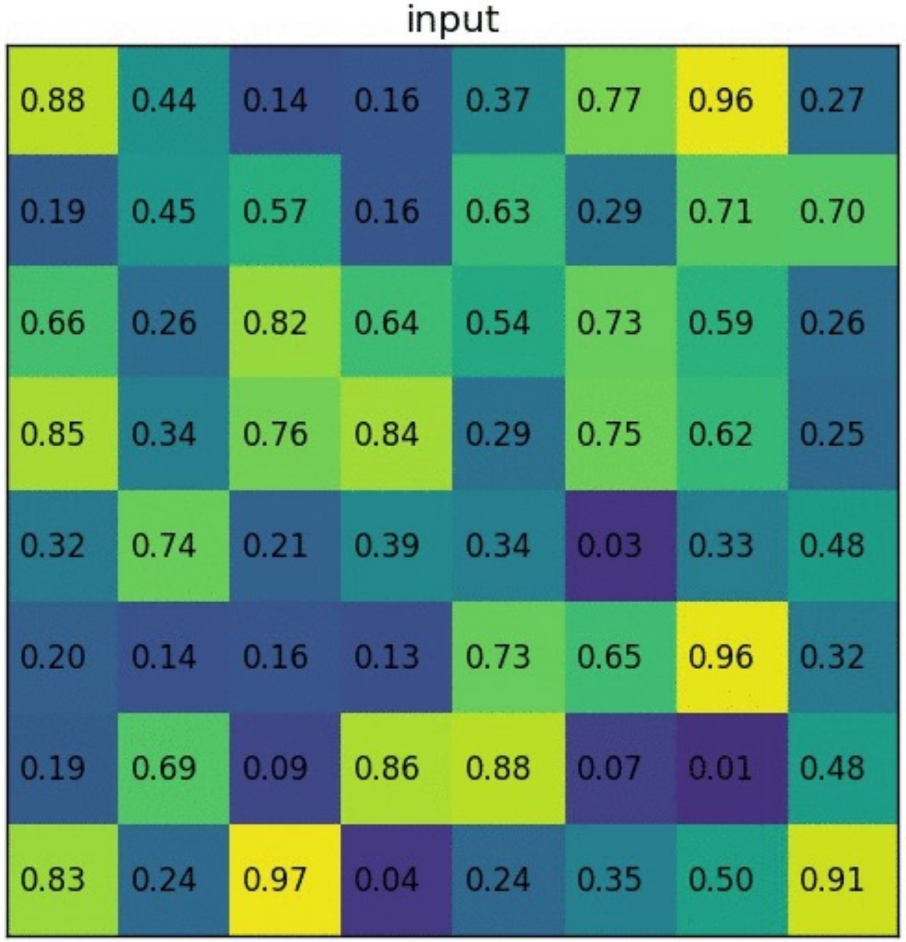
Mask R-CNN



Mask R-CNN

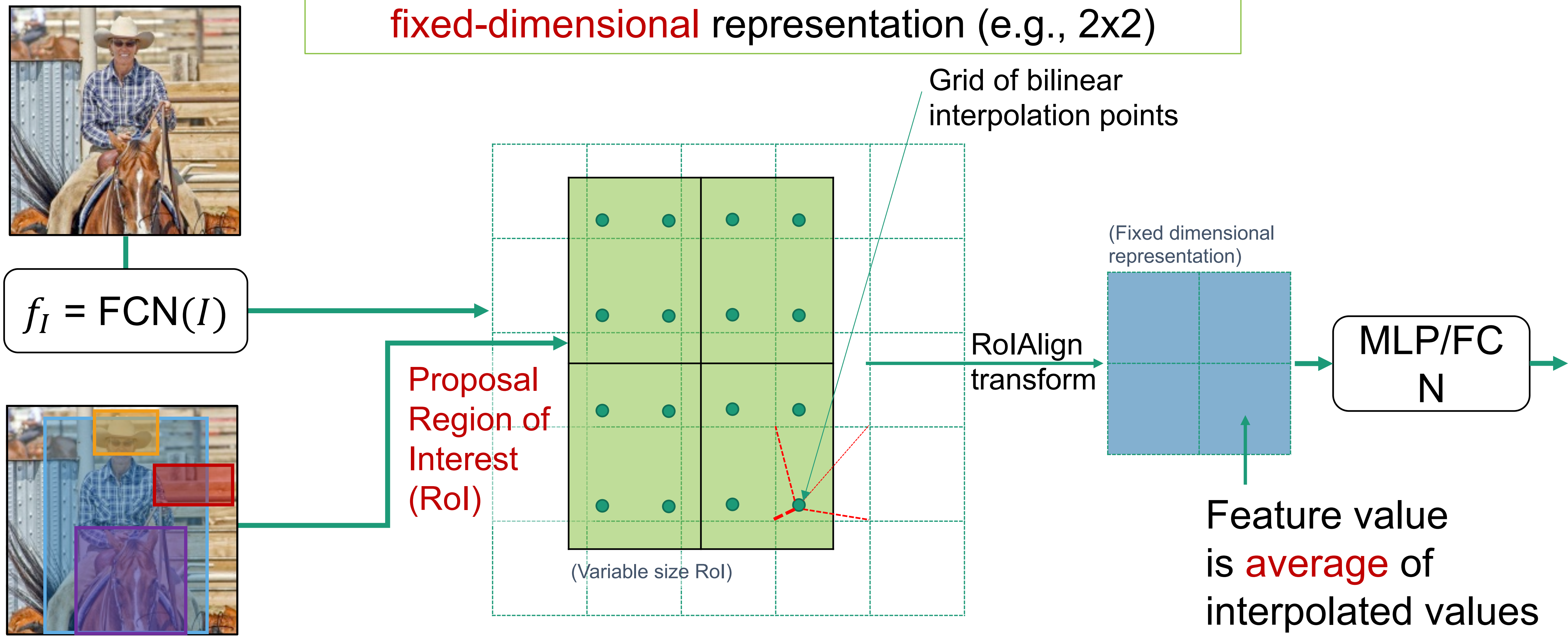

Per-image computation
Per-region computation for each $r_i \in r(I)$
f_I = FPN(I)
I:
RPN(f_I)
RoIAlign
MLP
Softmax clf.
Box regressor
Cascaded heads (inference only)
FCN
Masks
RoI**Pool** is replaced with RoI**Align**

RoI pooling

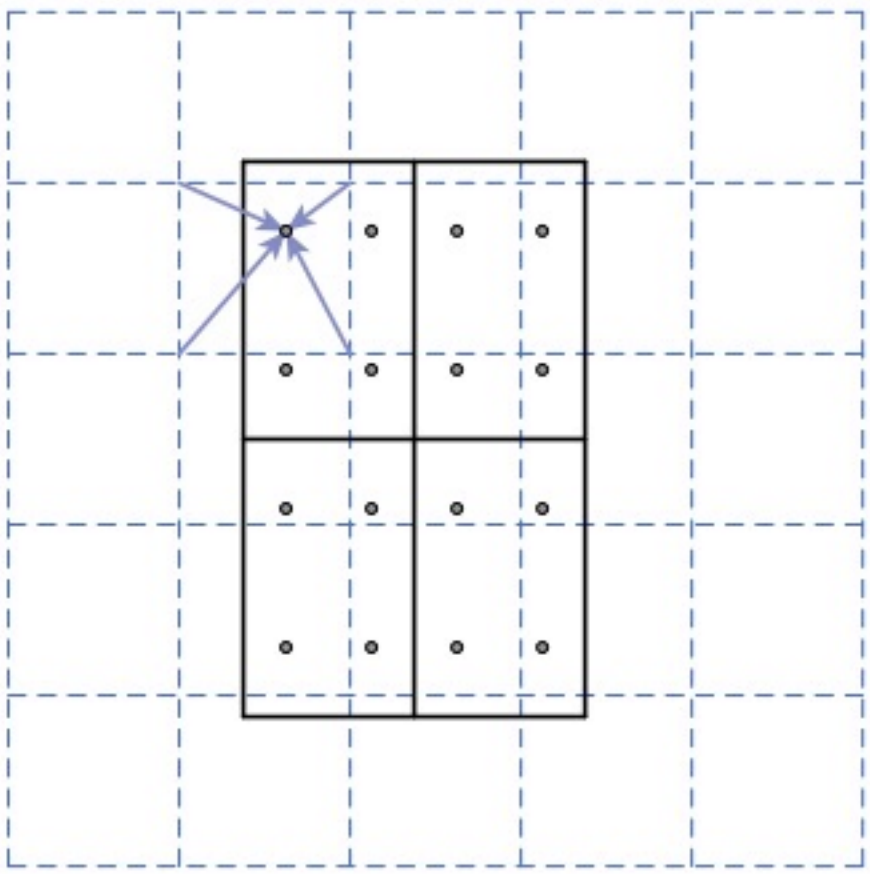

input
0.88 0.44 0.14 0.16 0.37 0.77 0.96 0.27
0.19 0.45 0.57 0.16 0.63 0.29 0.71 0.70
0.66 0.26 0.82 0.64 0.54 0.73 0.59 0.26
0.85 0.34 0.76 0.84 0.29 0.75 0.62 0.25
0.32 0.74 0.21 0.39 0.34 0.03 0.33 0.48
0.20 0.14 0.16 0.13 0.73 0.65 0.96 0.32
0.19 0.69 0.09 0.86 0.88 0.07 0.01 0.48
0.83 0.24 0.97 0.04 0.24 0.35 0.50 0.91

RoI pooling → RoIAlign

Transform arbitrary size proposal into a fixed-dimensional representation (e.g., 2x2)

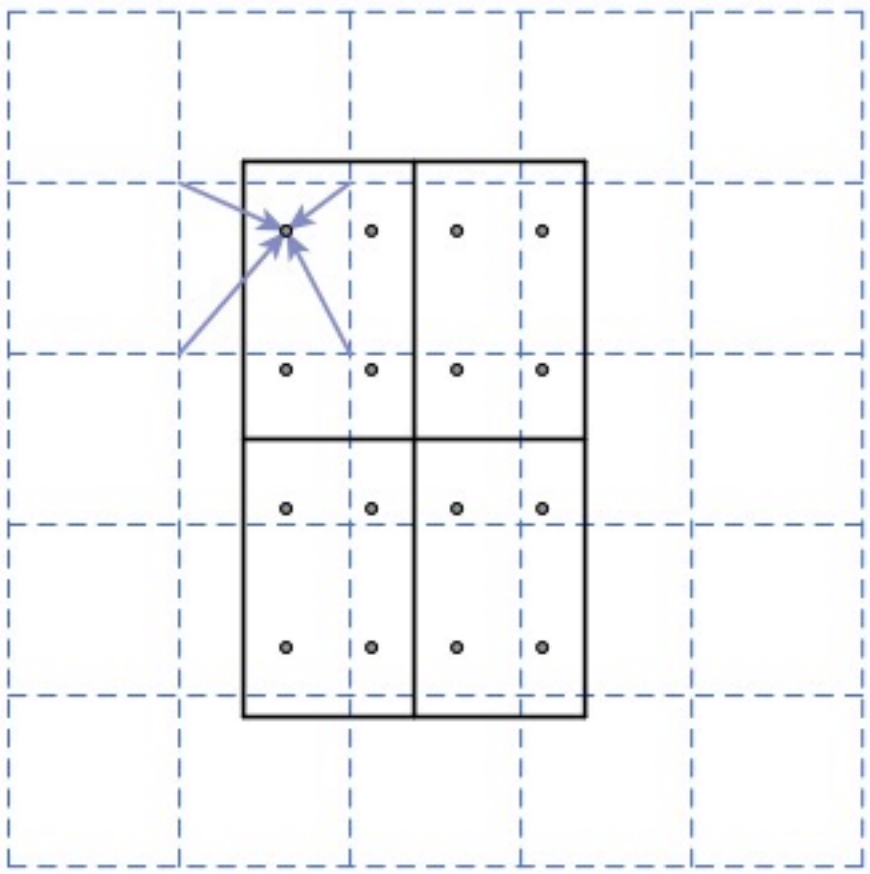


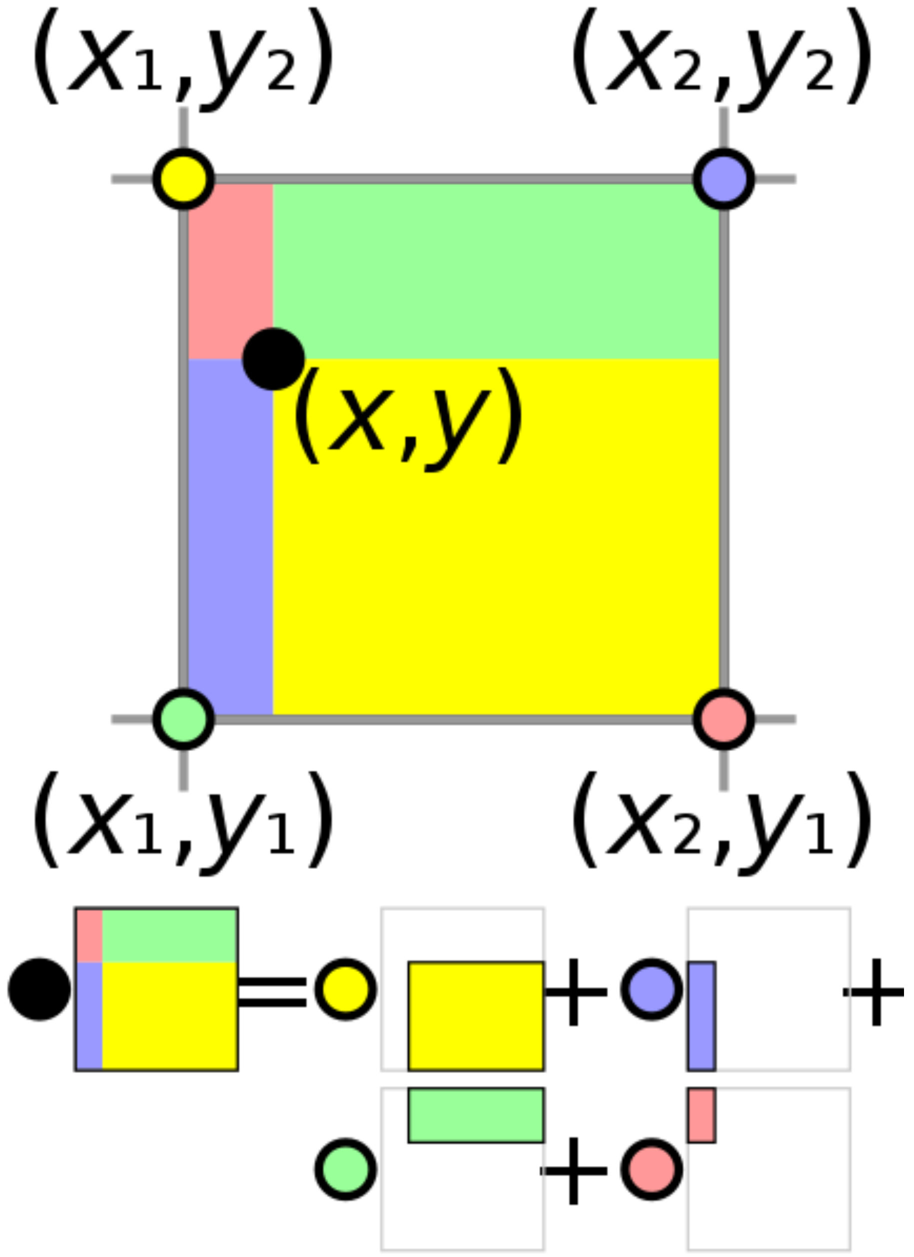
RoIAlign



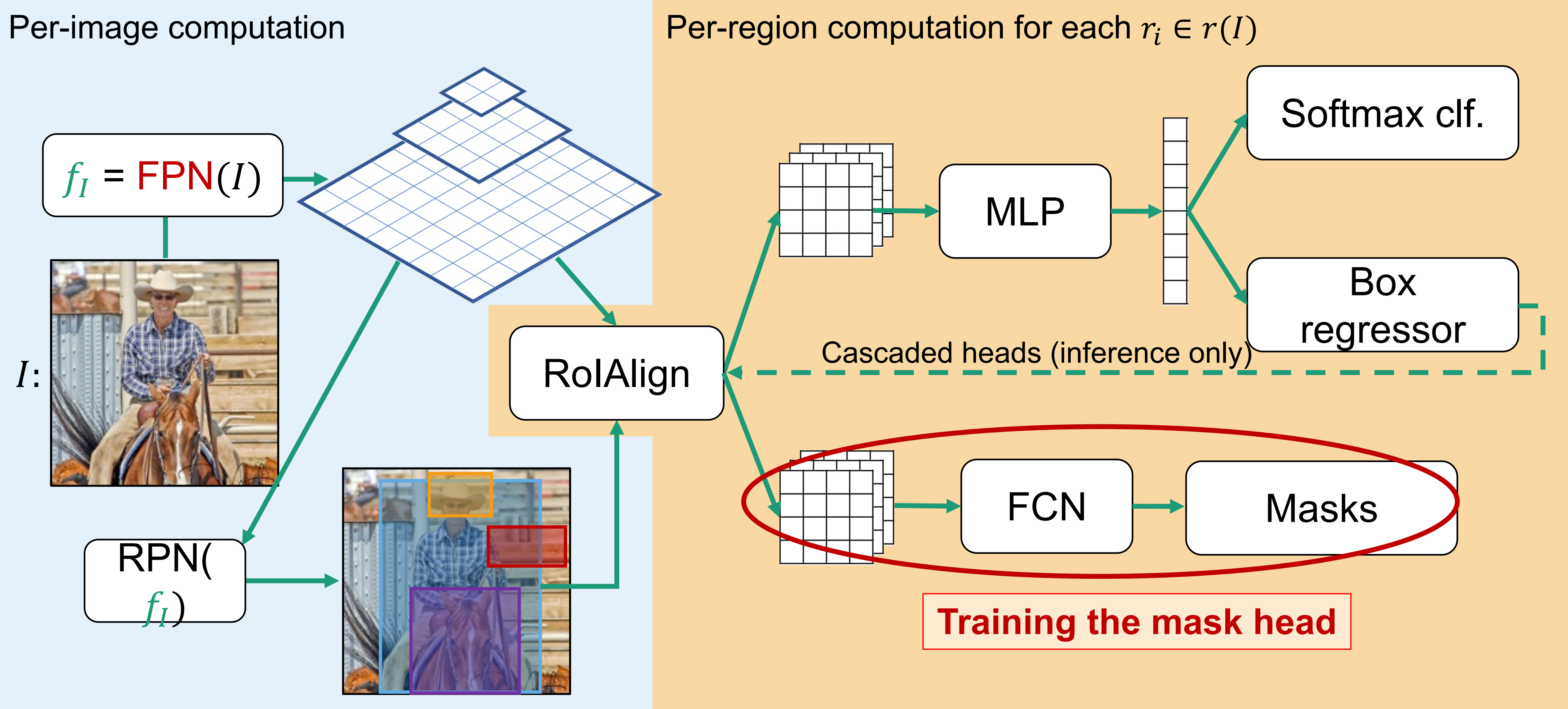
- Bilinear interpolation for each sampled location
- Use max pooling / avg pooling for each roi bin

RoIAlign





Mask R-CNN

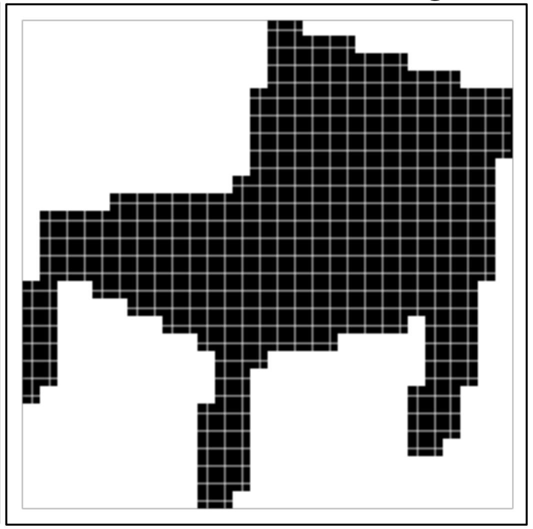


Per-image computation
Per-region computation for each $r_i \in r(I)$
f_I = FPN(I)
I:
RPN(f_I)
RoIAlign
MLP
Softmax clf.
Box regressor
Cascaded heads (inference only)
FCN
Masks
Training the mask head

Example Mask Training Targets

Image with training proposal

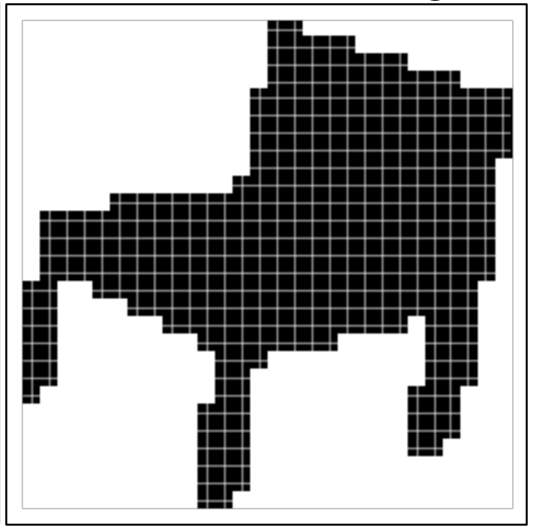
28x28 mask target

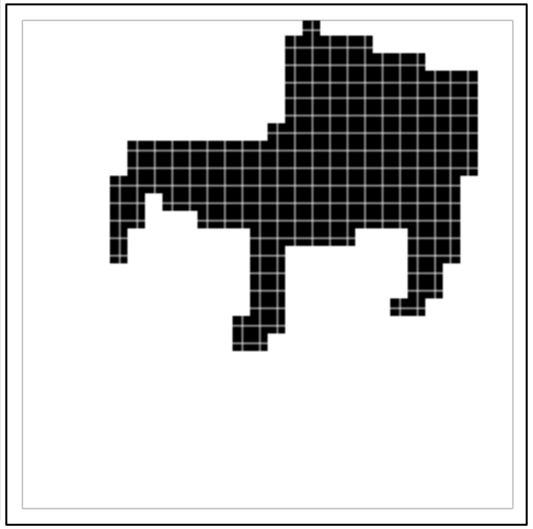


Example Mask Training Targets

Image with training proposal

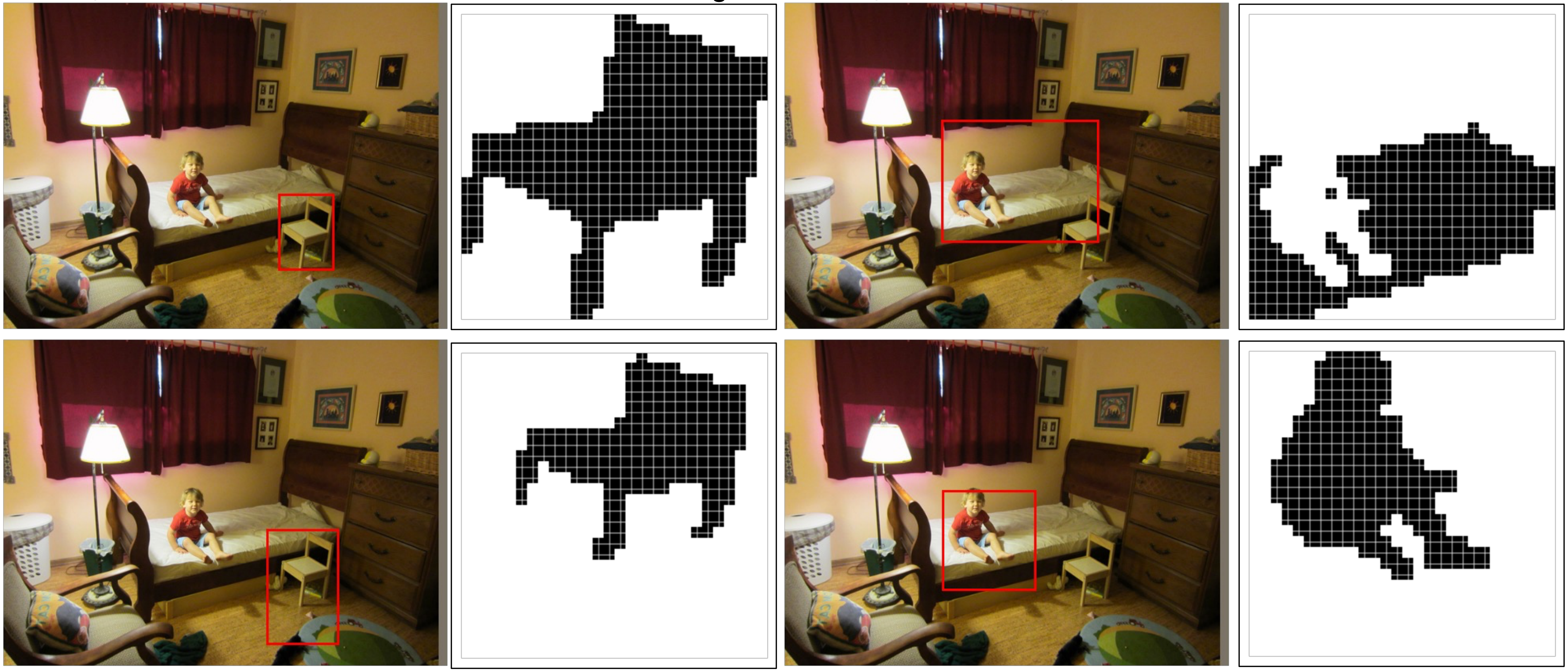
28x28 mask target



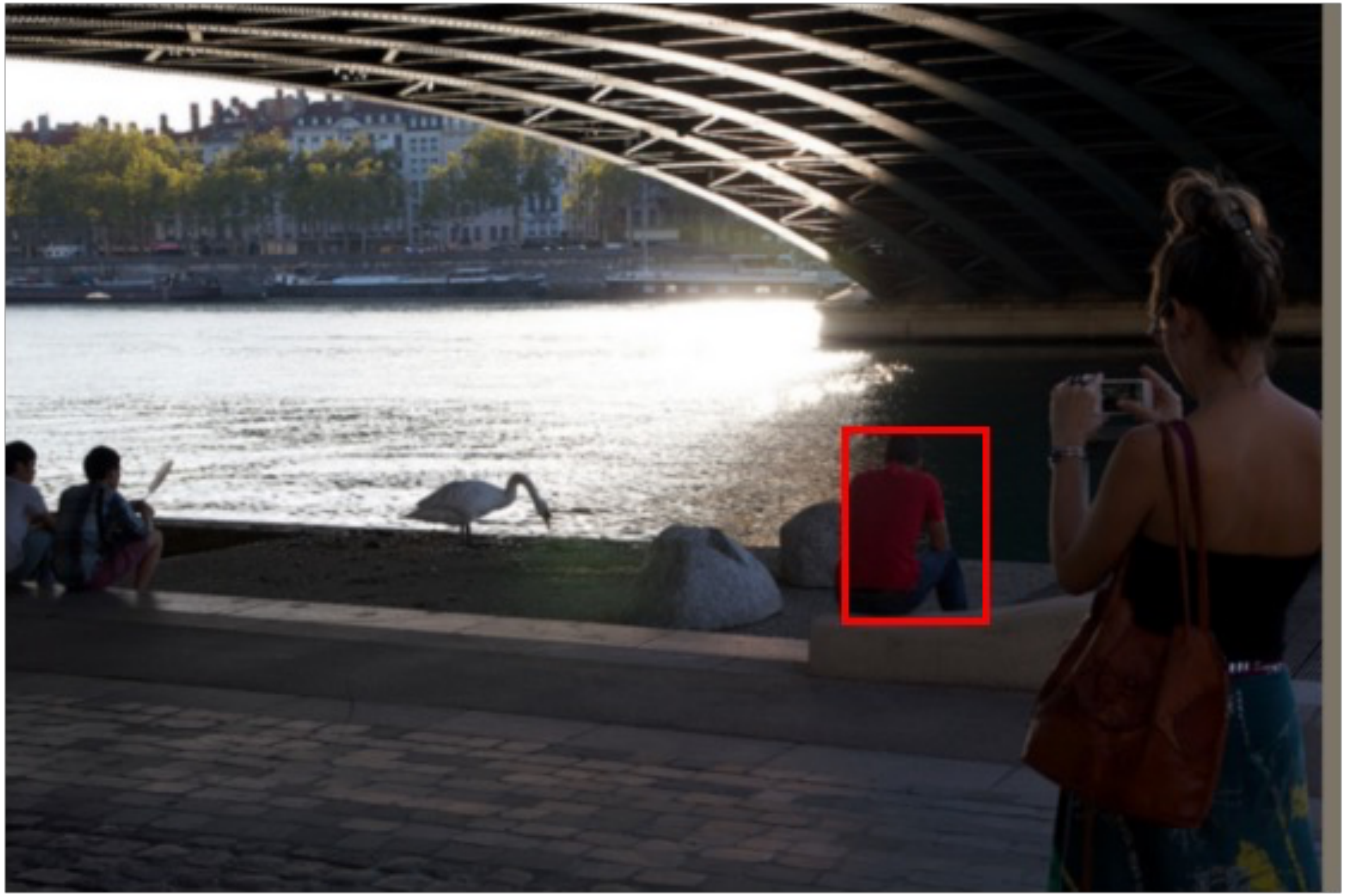


Example Mask Training Targets

Image with training proposal | 28x28 mask target | Image with training proposal | 28x28 mask target

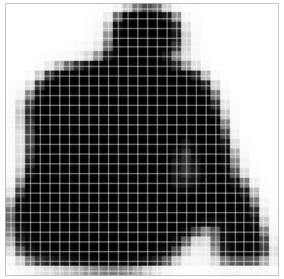


Binary Cross Entropy Loss on each pixel

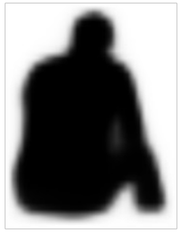


Validation image with box detection shown in red

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



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

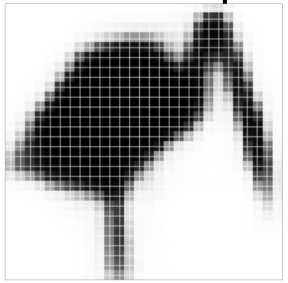


Final prediction (threshold at 0.5)

Binary Cross Entropy Loss on each pixel


Validation image with box detection shown in red

28x28 soft prediction

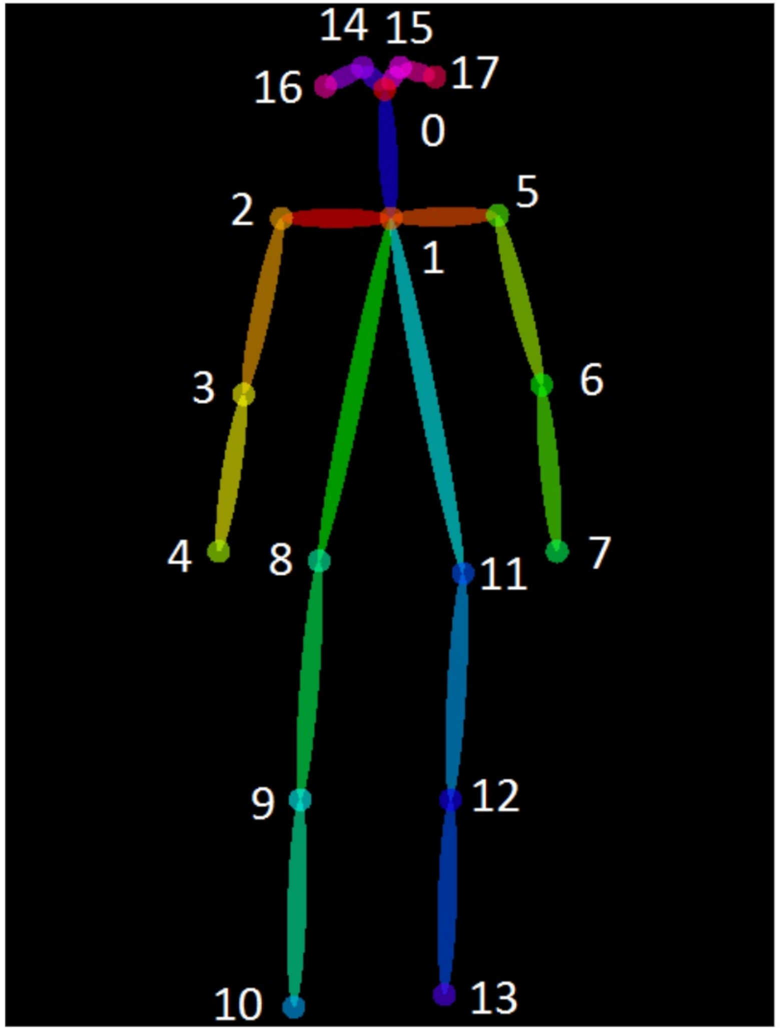


Resized Soft prediction

Final mask


person1.00
person1.00
person1.00
person1.00
person.91
person.98
person.74
surfboard1.00
surfboard1.00
surfboard.98
surfboard1.00
surfboard1.00

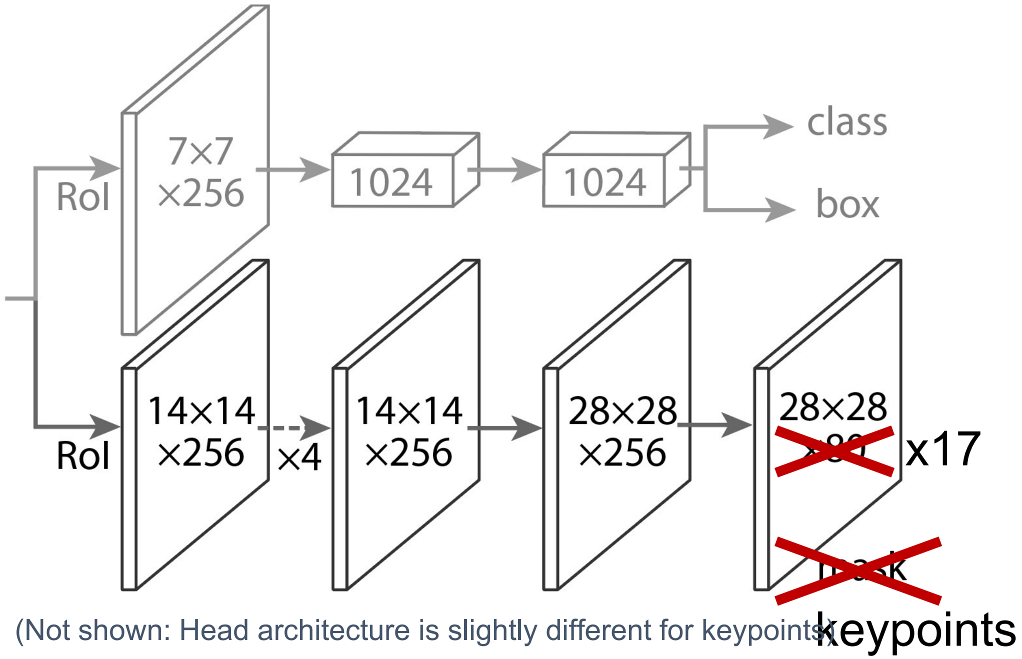
Human Pose Estimation



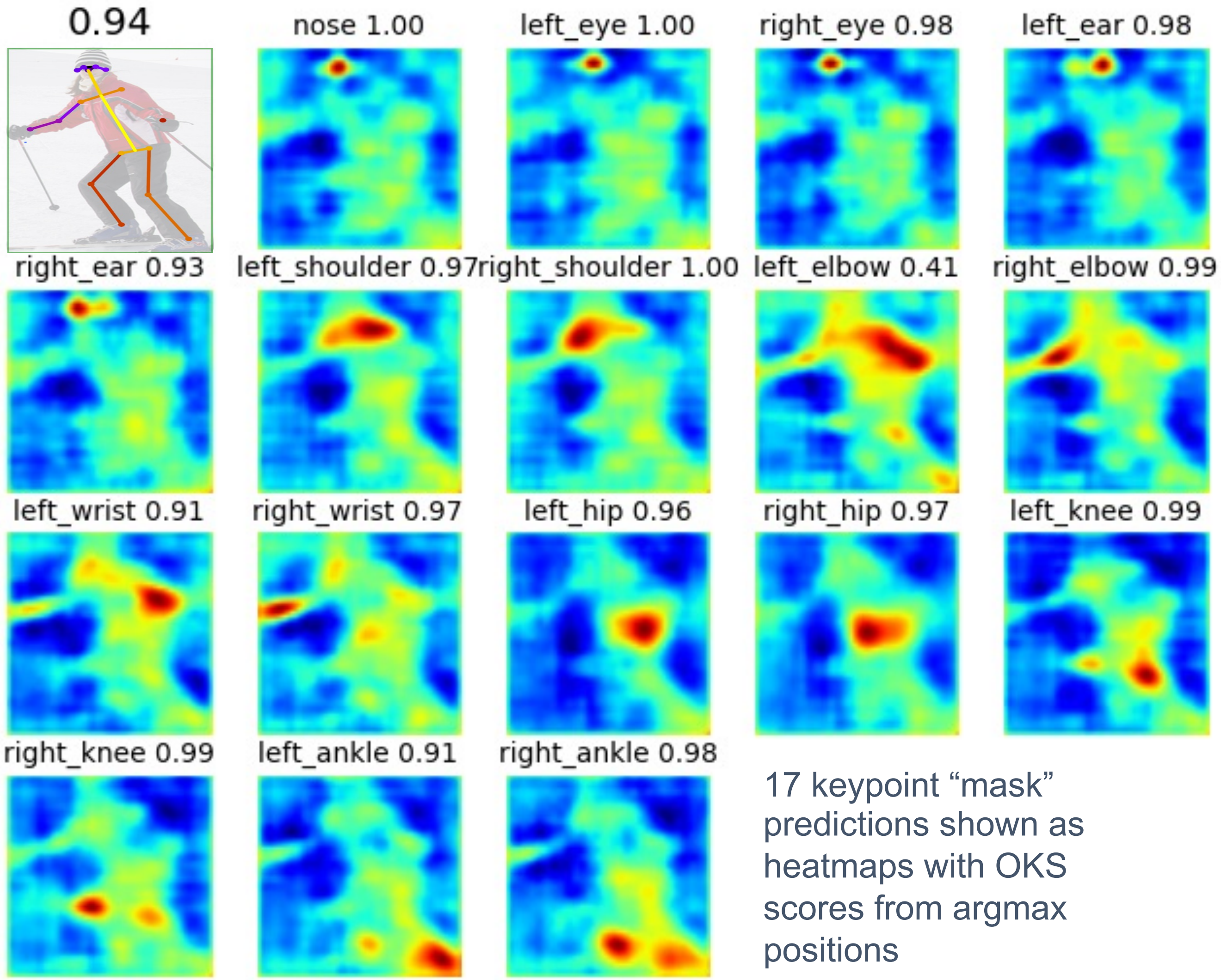
Human Pose Estimation

Human Pose GT generation

Pose Head



(Not shown: Head architecture is slightly different for keypoints)



17 keypoint “mask” predictions shown as heatmaps with OKS scores from argmax positions

- ➢ Add keypoint head (28x28x17)
- ➢ Predict one “mask” for each keypoint
- ➢ Softmax over spatial locations (encodes one keypoint per mask “prior”)

Pose Head