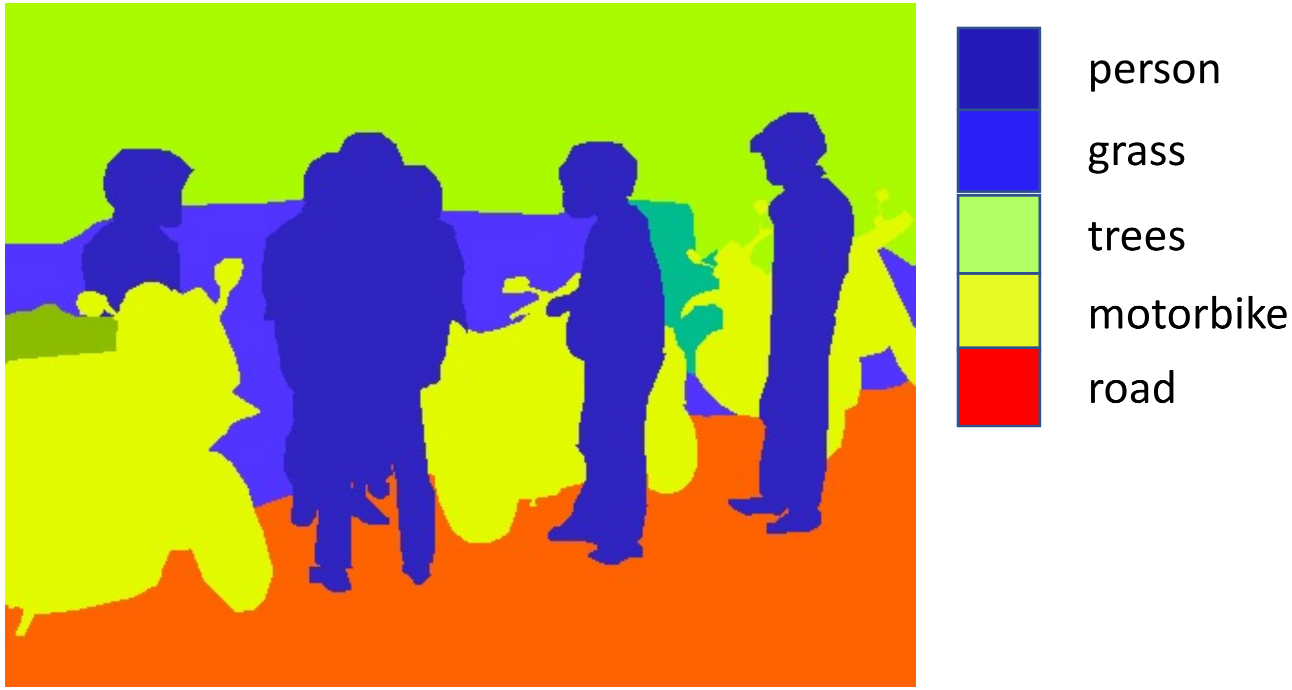
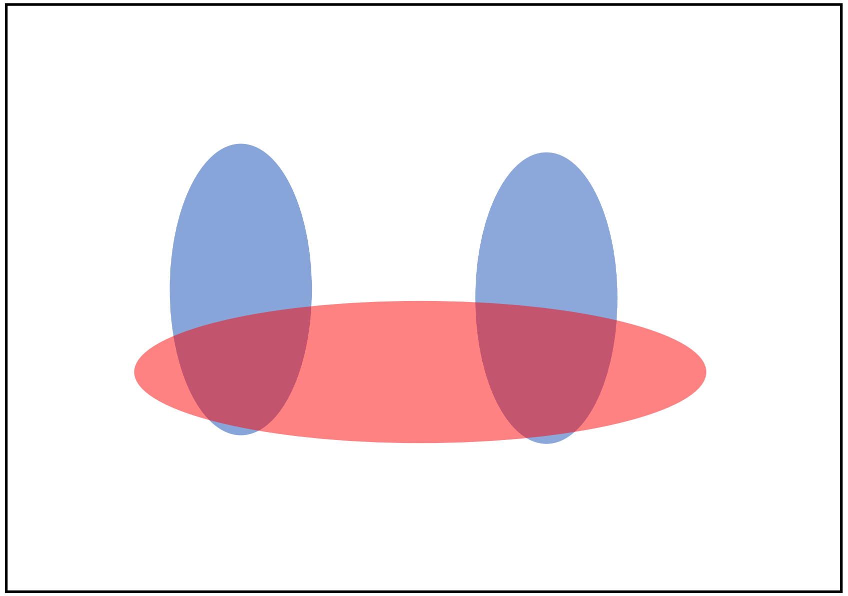
(Semantic / Instance / Panoptic) Segmentation

Semantic Segmentation



Evaluation metric

- Pixel classification!
- Accuracy?
  - Heavily unbalanced
- *Intersection over Union*
  - Average across classes and images
- Per-class accuracy
  - Average across classes and images



Things vs Stuff

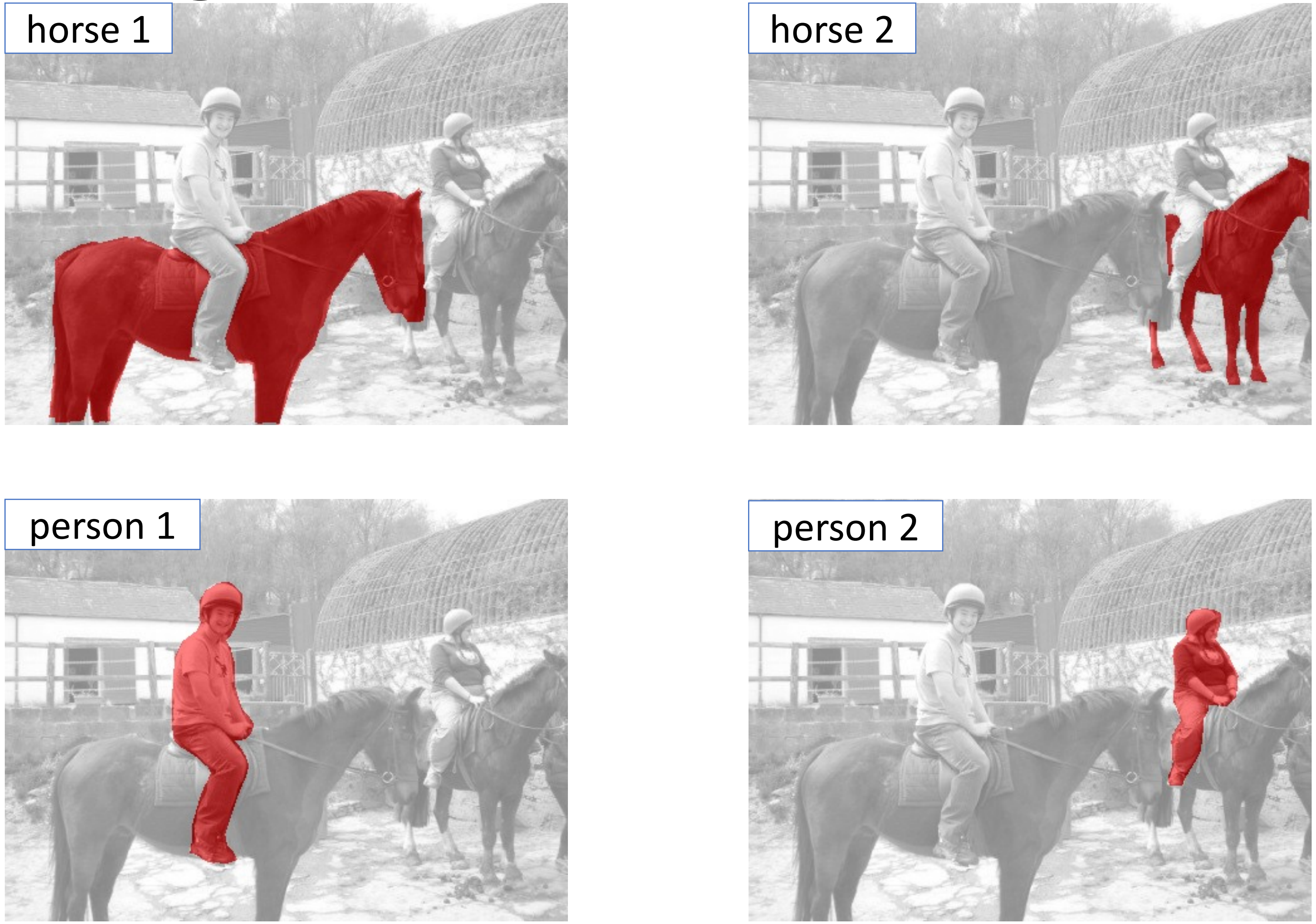
THINGS

- Person, cat, horse, etc
- Constrained shape
- Individual instances with separate identity
- May need to look at objects

STUFF

- Road, grass, sky etc
- Amorphous, no shape
- No notion of instances
- Can be done at pixel level
- “texture”

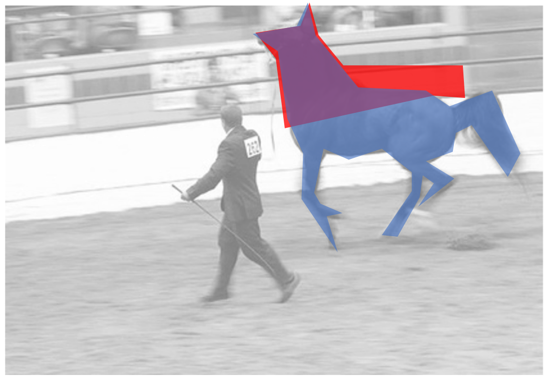
Instance Segmentation



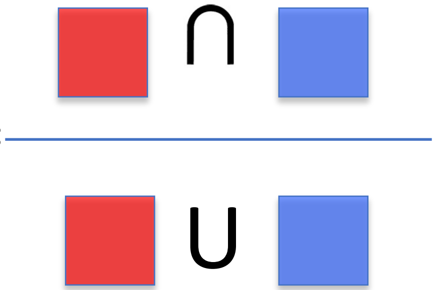
Instance Segmentation

Evaluation Protocol

- Sort predicted instances by confidence
- Match prediction to closest annotation based on *segment overlap*
  - If segment overlap > threshold, correct



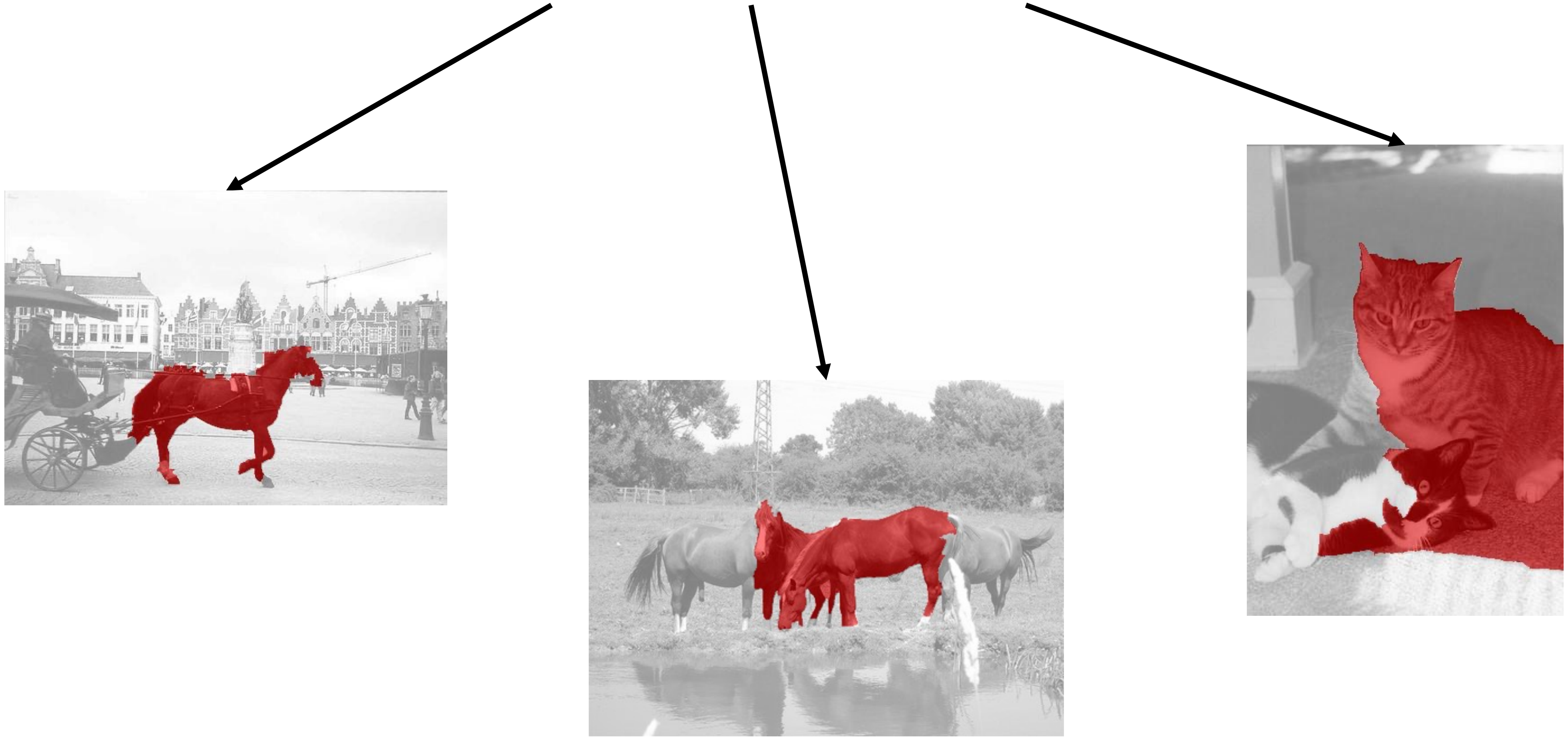
segment overlap =



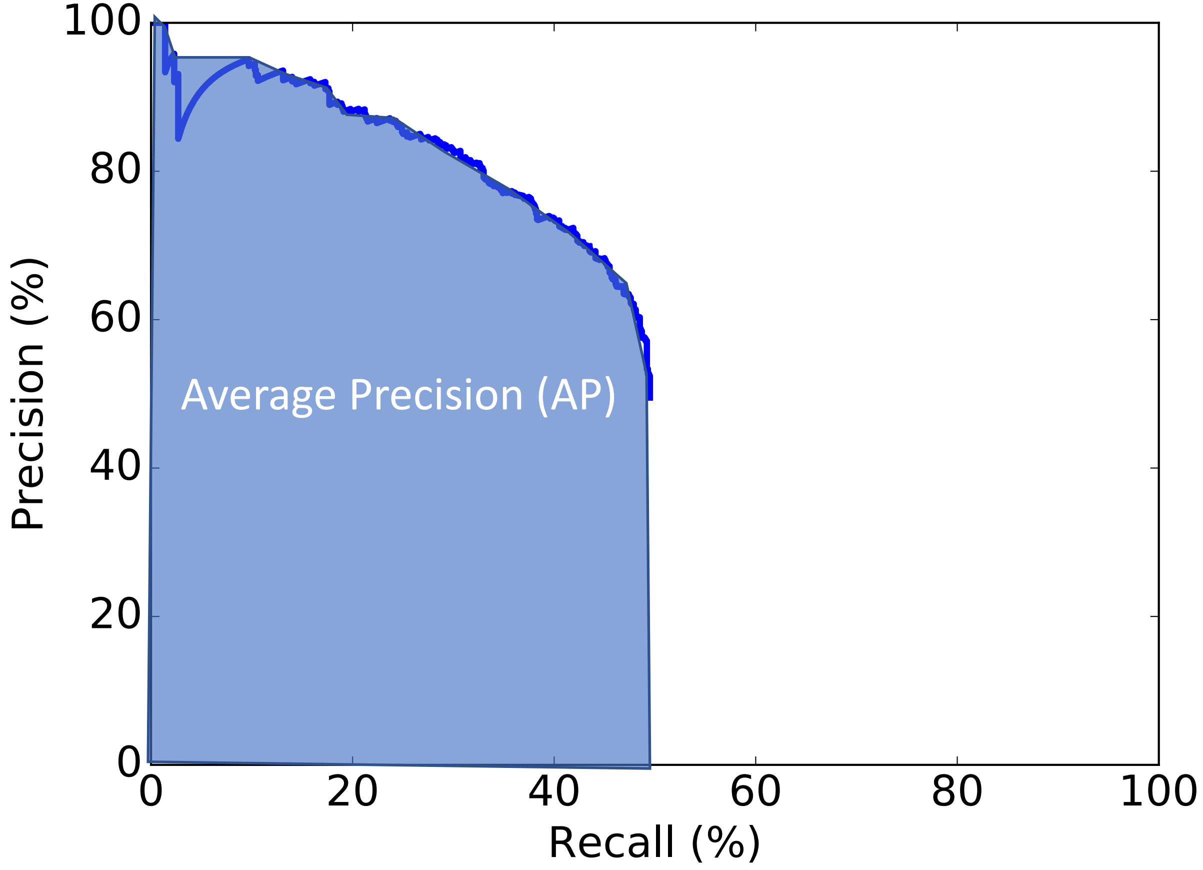
Evaluation Protocol

Labels = [✓ ✓ ✗ ✓ ✗ ✗ ✓ ....]

Scores = [0.90 0.87 0.82 0.78 0.70 0.69 0.60 ....]

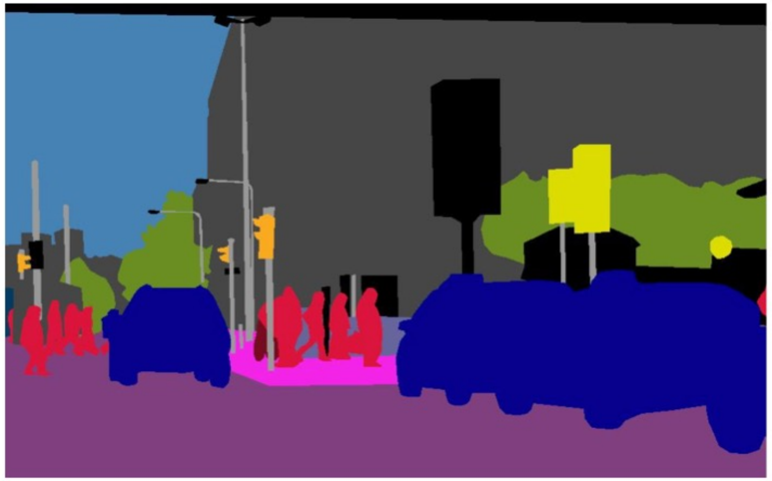


Evaluation protocol

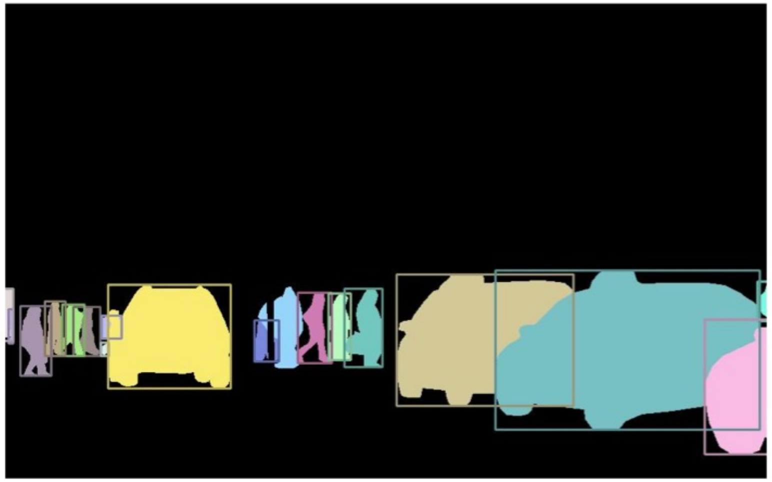


Panoptic Segmentation

(a) image



(b) semantic segmentation



(c) instance segmentation

(d) panoptic segmentation

Panoptic segmentation evaluation metric

$$\mathrm{PQ} = \underbrace{\frac{\sum_{(p,g)\in TP} \mathrm{IoU}(p,g)}{|TP|}}_{\text{segmentation quality (SQ)}} \times \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\text{recognition quality (RQ)}}$$

The COCO Challenge

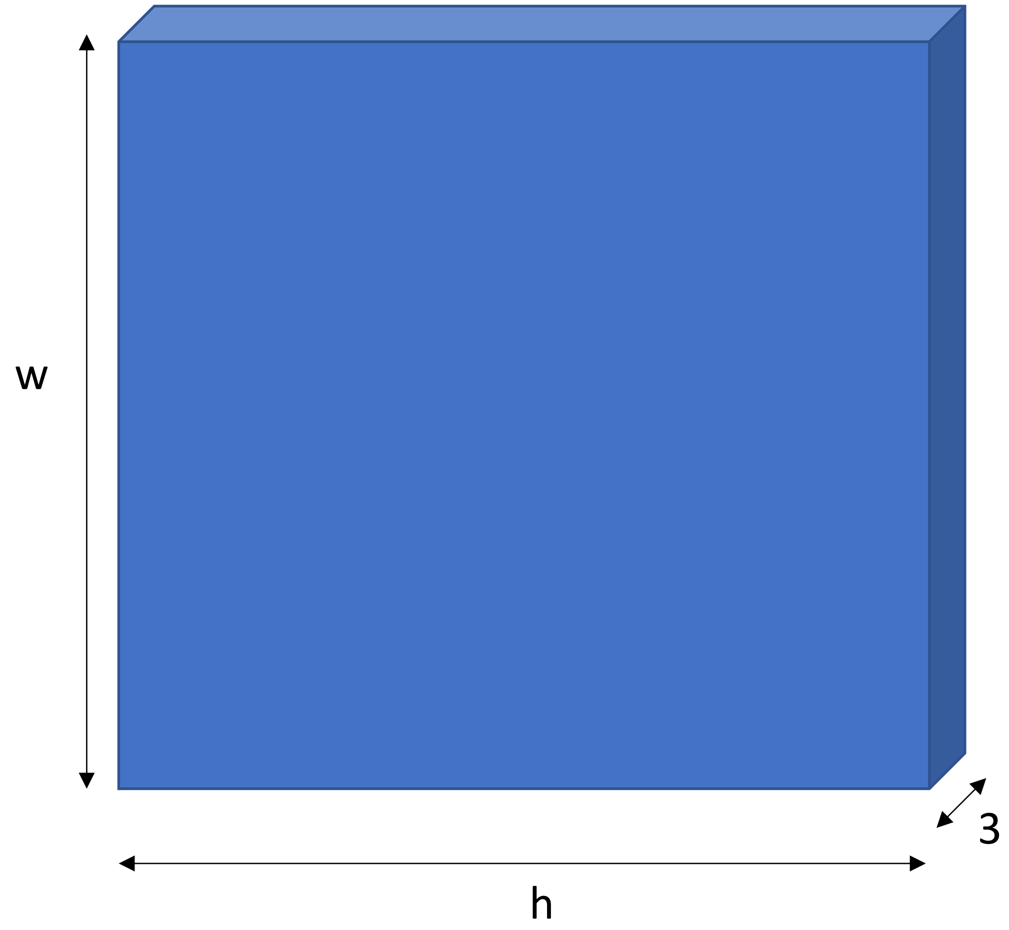
mscoco.org

T. Y. Lin et al. **Microsoft COCO: Common Objects in Context**. In ECCV, 2014

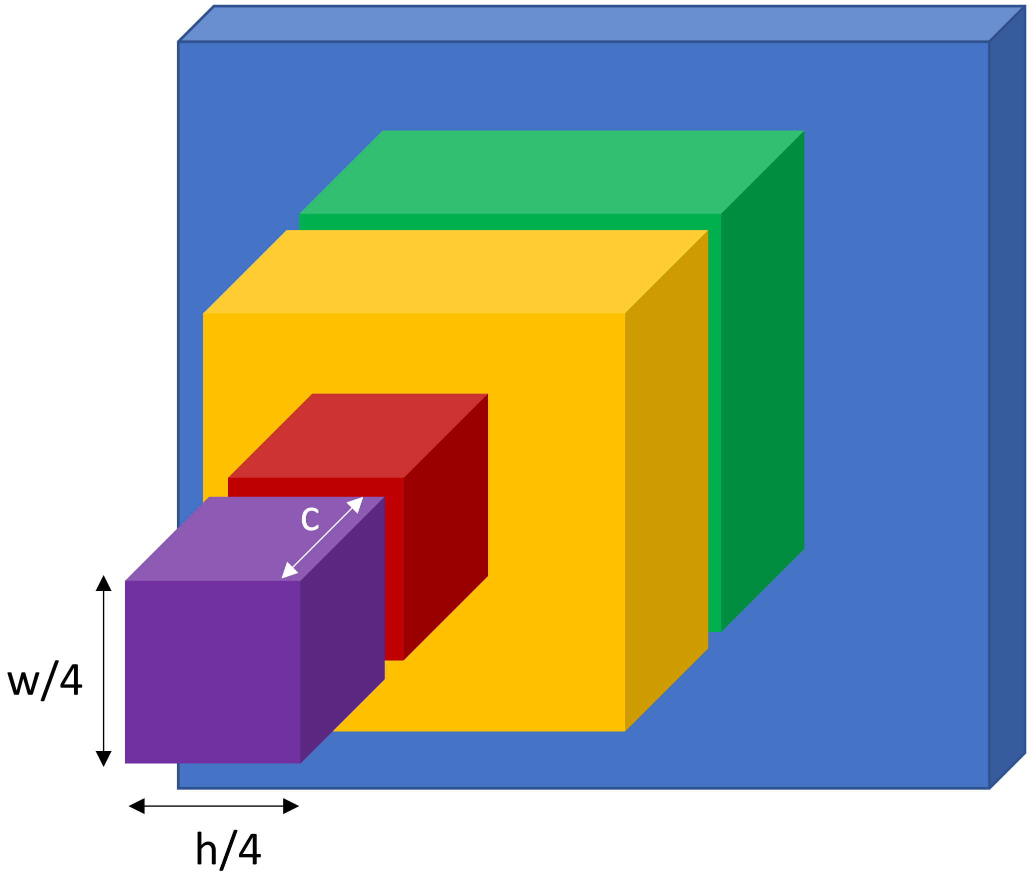
Challenges in data collection

- Precise localization is hard to annotate
- Annotating every pixel leads to heavy tails
- Common solution: annotate few classes (often things), mark rest as “Other”
- Common datasets: PASCAL VOC 2012 (~1500 images, 20 categories), COCO (~100k images, 20 categories)

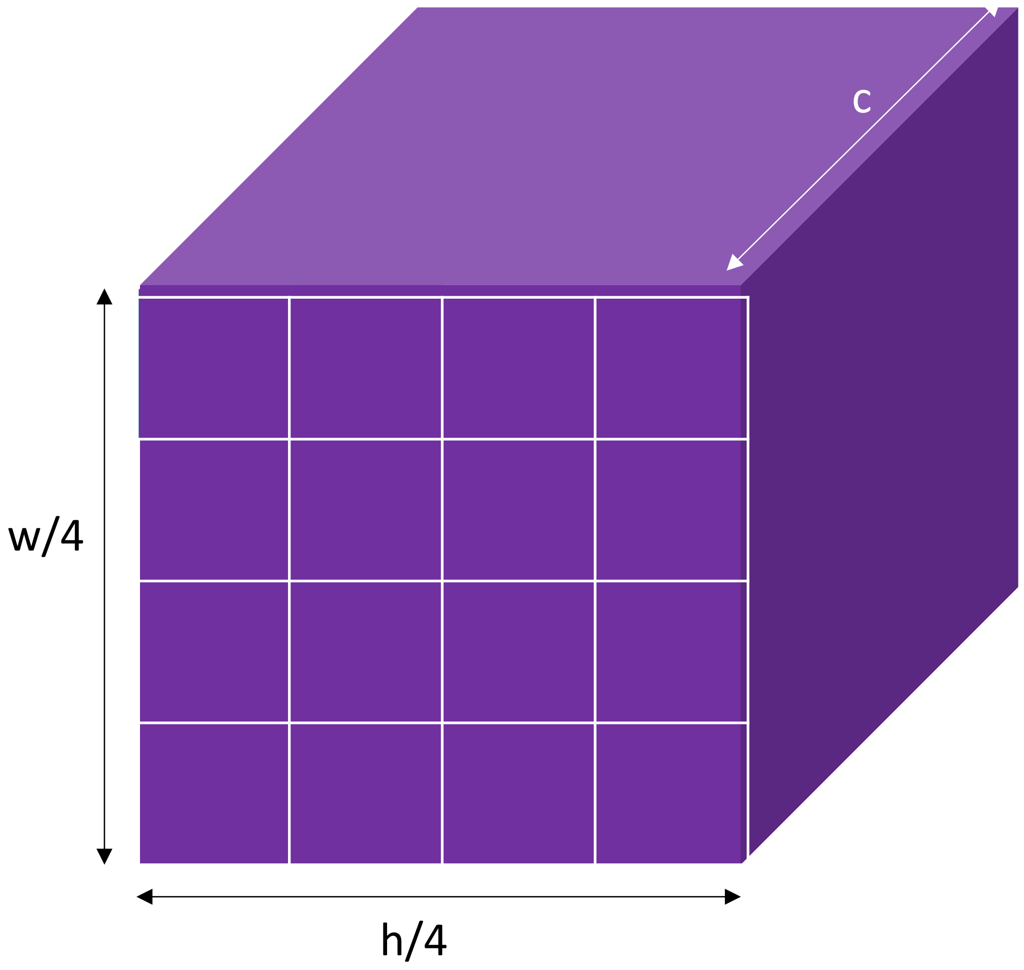
Semantic segmentation using convolutional networks



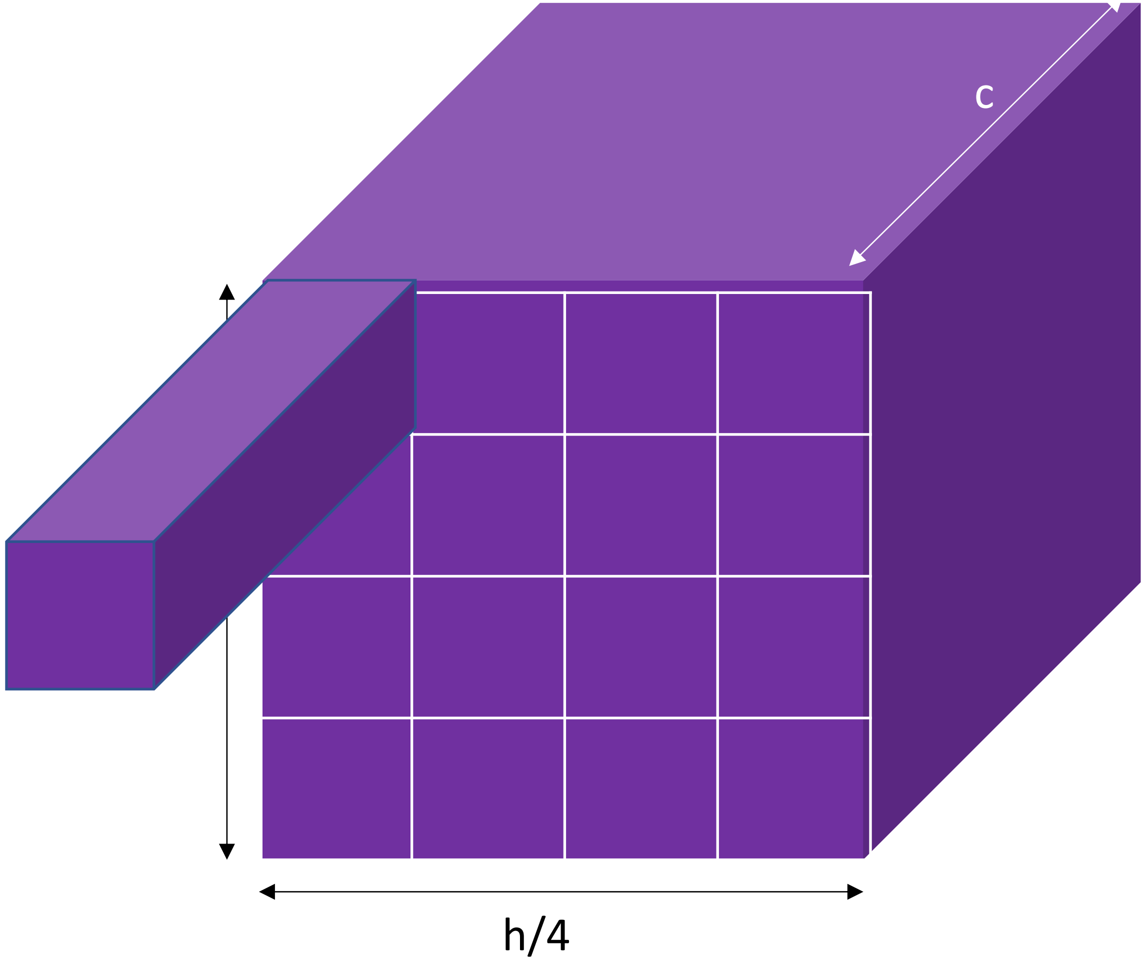
Semantic segmentation using convolutional networks



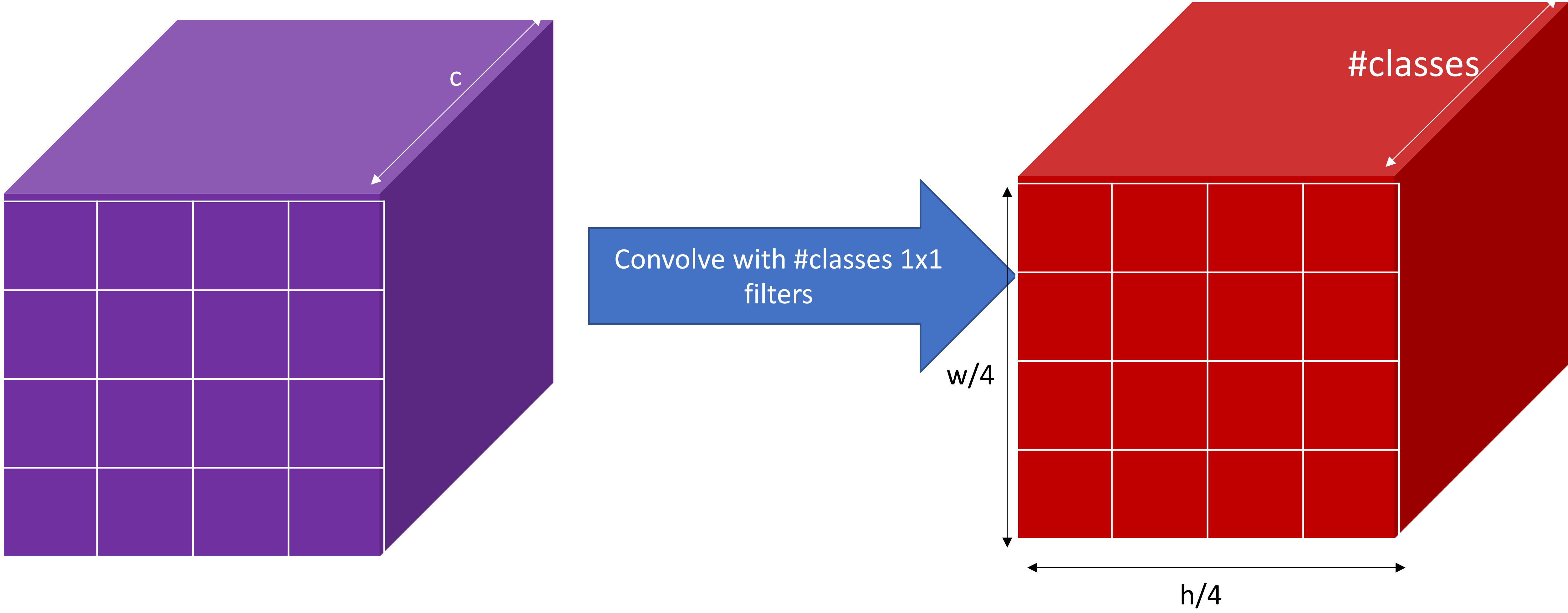
Semantic segmentation using convolutional networks



Semantic segmentation using convolutional networks



Semantic segmentation using convolutional networks



Semantic segmentation using convolutional networks

- Pass image through convolution and subsampling layers
- Final convolution with #classes outputs
- Get scores for *subsampled* image
- Upsample back to original size

Transfer learning for semantic segmentation

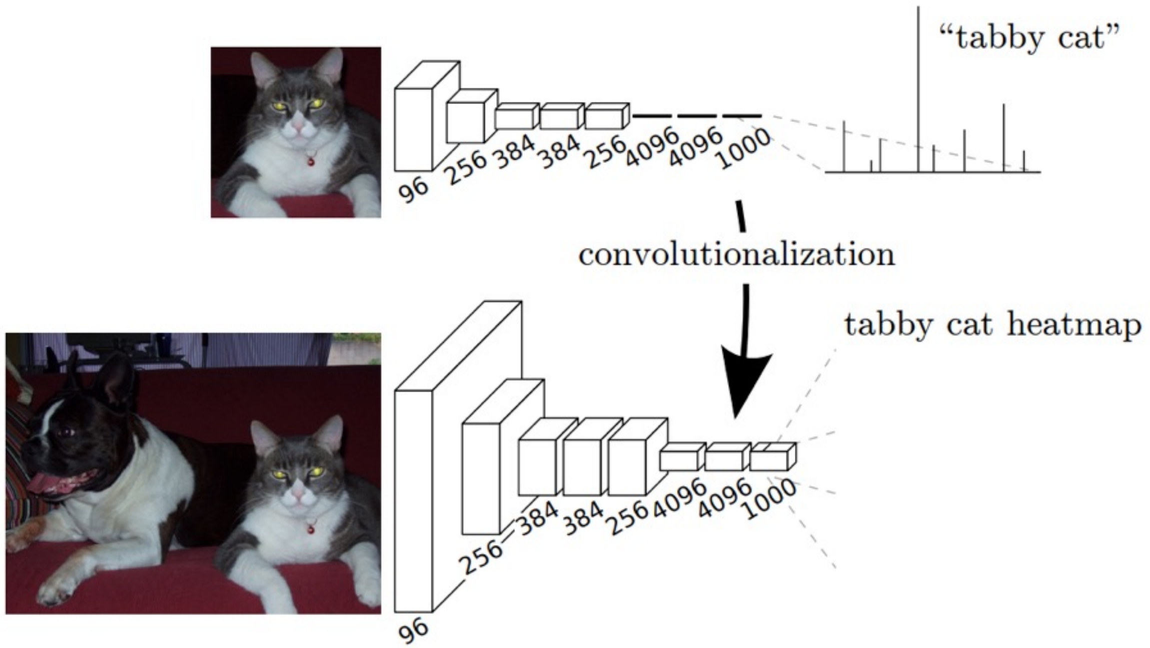


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

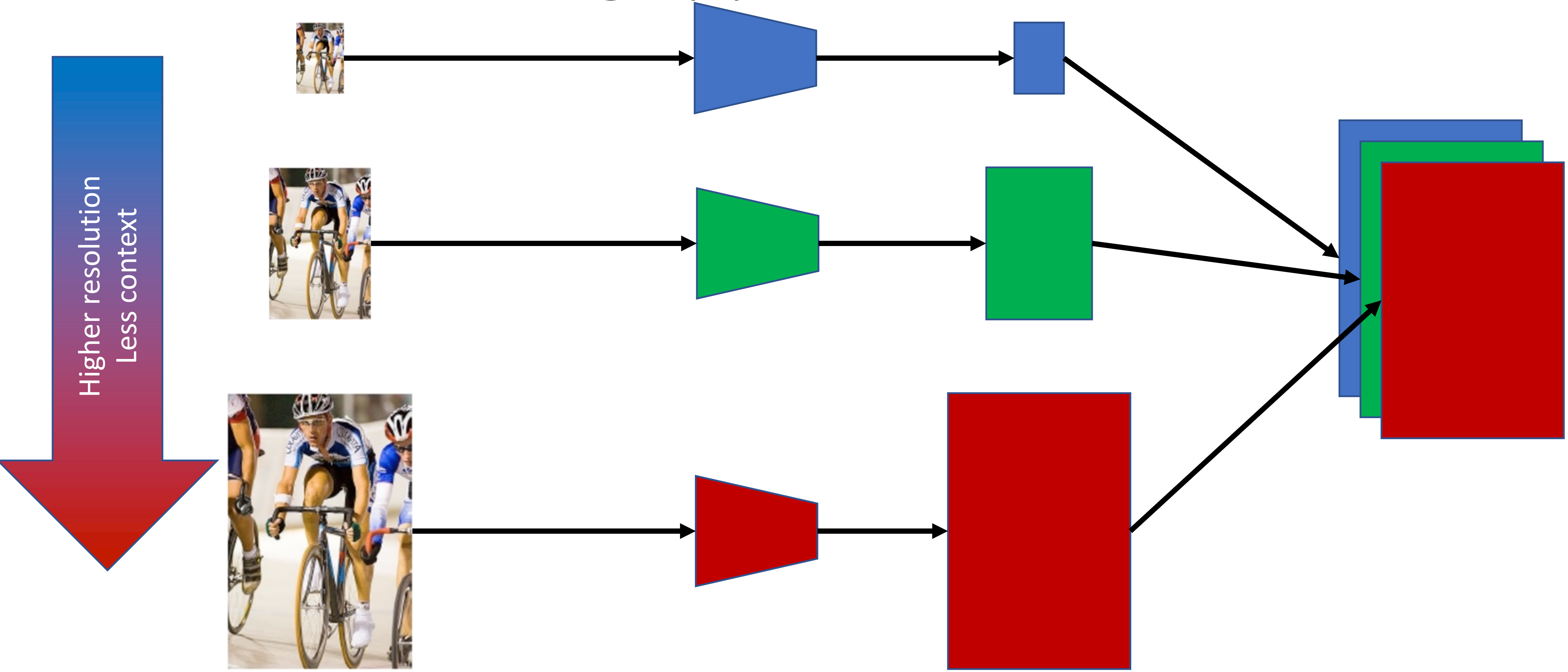
Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

Semantic segmentation using convolutional networks

The resolution issue

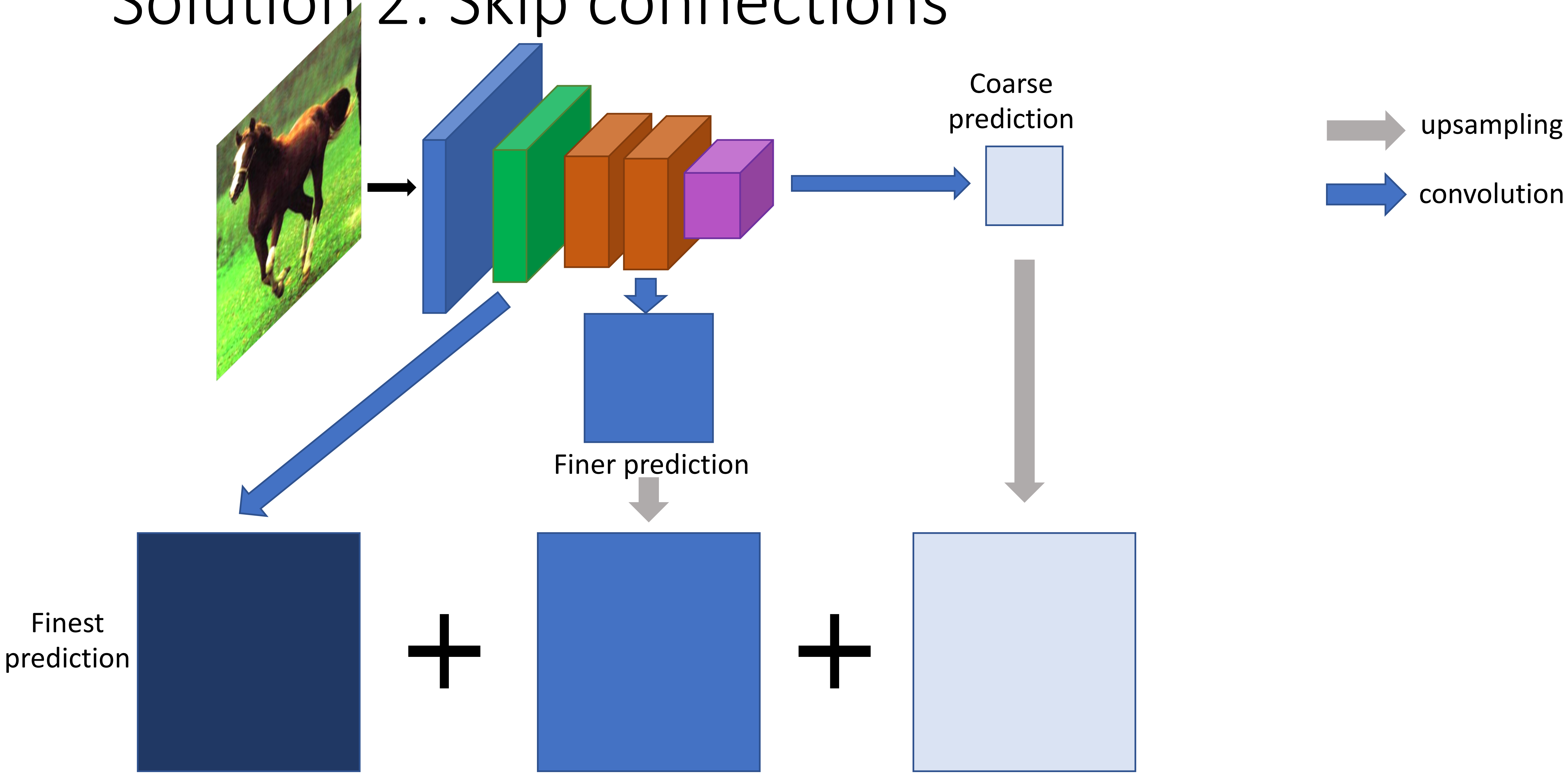
- Problem: Need fine details!
- Shallower network / earlier layers?
  - Not very semantic!
- Remove subsampling?
  - Looks at only a small window!

Solution 1: Image pyramids



Learning Hierarchical Features for Scene Labeling. Clement Farabet, Camille Couprie, Laurent Najman, Yann LeCun. In *TPAMI,* 2013.

Solution 2: Skip connections

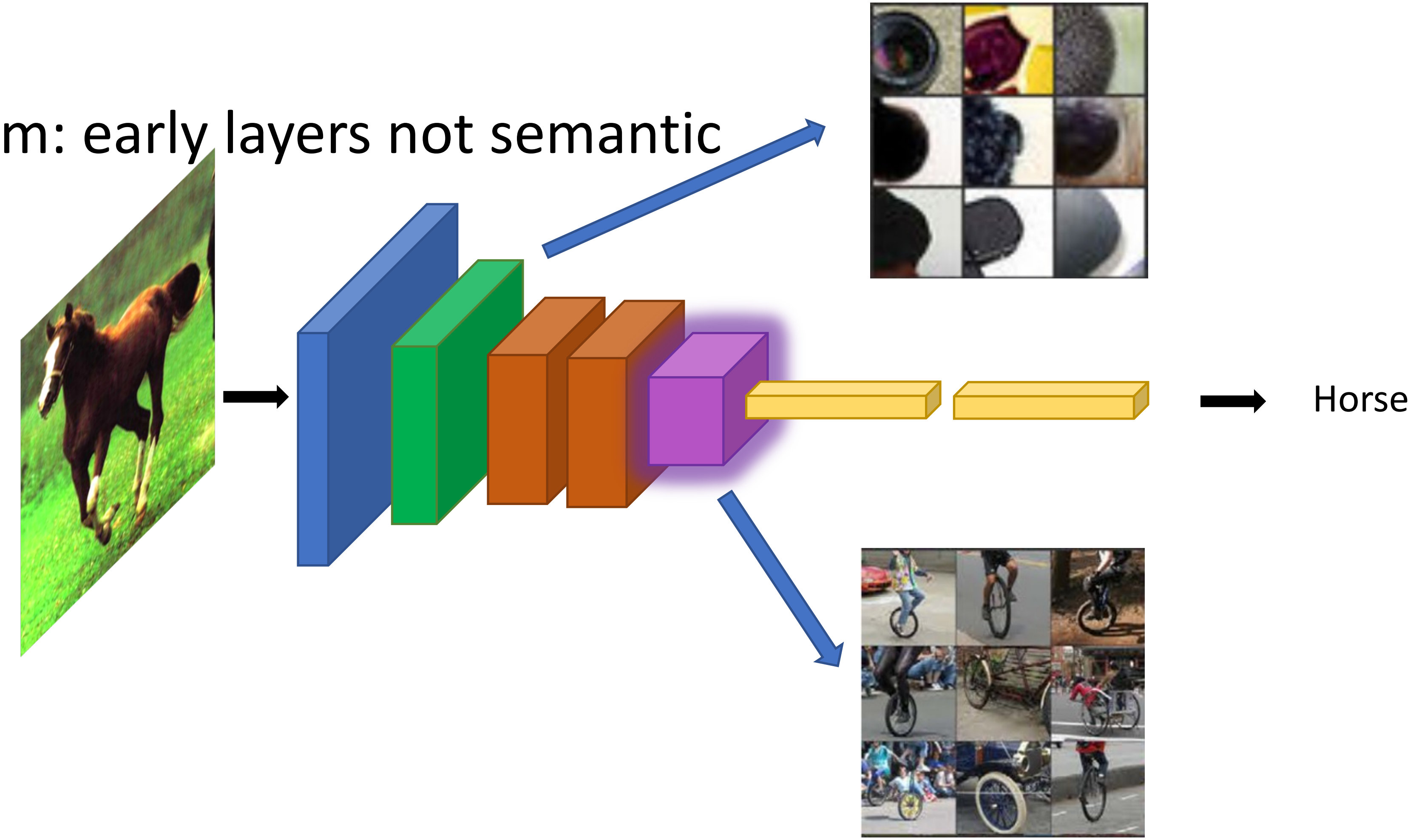


Skip connections

Fully convolutional networks for semantic segmentation. Evan Shelhamer, Jon Long, Trevor Darrell. In *CVPR* 2015

Skip connections

- Problem: early layers not semantic



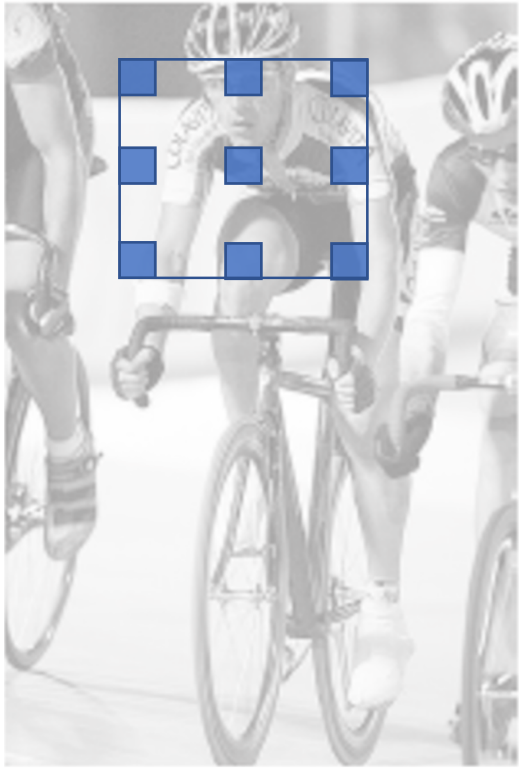
Visualizations from : M. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. In *ECCV* 2014.

Solution 3: Dilation

- Need subsampling to allow convolutional layers to capture large regions with small filters
  - Can we do this without subsampling?

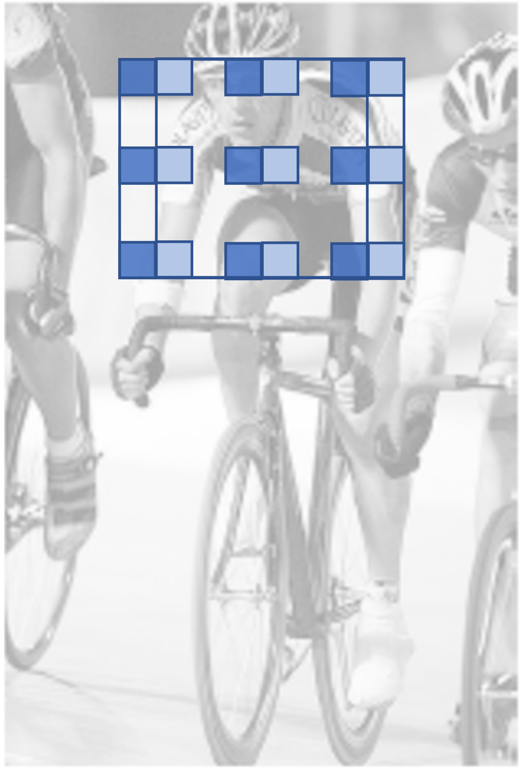
Solution 3: Dilation

- Need subsampling to allow convolutional layers to capture large regions with small filters
  - Can we do this without subsampling?



Solution 3: Dilation

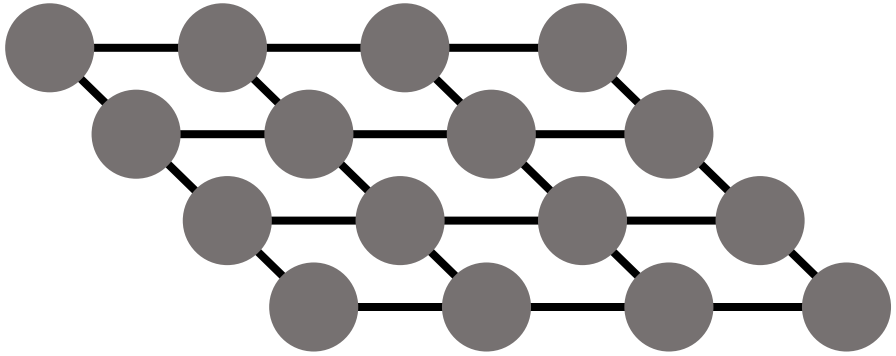
- Need subsampling to allow convolutional layers to capture large regions with small filters
  - Can we do this without subsampling?



Solution 3: Dilation

- Instead of subsampling by factor of 2: dilate by factor of 2
- Dilation can be seen as:
  - Using a much larger filter, but with most entries set to 0
  - Taking a small filter and “exploding”/ “dilating” it
- Not panacea: without subsampling, feature maps are much larger: memory issues

Solution 4: Conditional random fields



$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} e^{-E(\mathbf{y},\mathbf{x})}$$

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$$

$$= \arg\min_{\mathbf{y}} E(\mathbf{y},\mathbf{x})$$

$$E(\mathbf{y},\mathbf{x}) = \sum_i E_{data}(y_i,\mathbf{x}) + \sum_{i,j \in \mathcal{N}} E_{smooth}(y_i, y_j, \mathbf{x})$$

Solution 4: Conditional Random Fields

- Idea: take convolutional network prediction and sharpen using classic techniques
- *Conditional Random Field*

$$\mathbf{y}^* = \arg\min_{\mathbf{y}} \sum_i E_{data}(y_i, \mathbf{x}) + \sum_{i,j \in \mathcal{N}} E_{smooth}(y_i, y_j, \mathbf{x})$$

$$E_{smooth}(y_i, y_j, \mathbf{x}) = \underbrace{\mu(y_i, y_j)}_{\text{Label compatibility}} \underbrace{w_{ij}(\mathbf{x})}_{\text{Pixel similarity}}$$

Inference in CRFs

- Problem: combinatorial optimization
- Variational methods: Approximate complex distribution p(y) with simple distribution q(y)
- Mean-field approximation: q(y) is independent distribution for each pixel:

$$q(\mathbf{y}) = \prod_i q_i(y_i)$$

- If N pixels and K classes, basically N K-dimensional vectors

Mean field inference

- If we can find best q, solution is highest probability output for each pixel
- Try to match p with q by minimizing *Kulback-Leibler Divergence*

$$KL(q||p) = \sum_{\mathbf{y}} q(\mathbf{y}) \log p(\mathbf{y}) - \sum_{\mathbf{y}} q(\mathbf{y}) \log q(\mathbf{y})$$

- Iterative process: in each iteration, do coordinate ascent on one $q(y_i)$

Mean field inference

- Coordinate descent on $q_i(y_i)$
- At each step, keep other pixels fixed and update one
- Each step (approximately):
  - Take current $q_j(y_j)$ on all $j \neq i$
  - Use this to compute $p(y_i|y_{-i})$ where $y_{-i} = \{y_j : j \neq i\}$
  - Set q_i to this

$$q_i \propto \mathbb{E}_{q_{-i}}[\log p(y_i|y_{-i})]$$

Fully Connected CRFs

- Typically, only adjacent pixels connected
  - Fewer connections => Easier to optimize
- Dense connectivity: every pixel connected to everything else
- Intractable to optimize except if pairwise potential takes specific form

$$E_{smooth}(y_i, y_j, \mathbf{x}) = \mu(y_i, y_j) w_{ij}(\mathbf{x})$$

$$w_{ij}(\mathbf{x}) = \sum_m w_m e^{-\|\mathbf{f}_m(i) - \mathbf{f}_m(j)\|^2}$$

Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. Philipp Krahenbuhl, Vladlen Koltun. In *NIPS,* 2011.

Gaussian edge potentials

$$w_{ij}(\mathbf{x}) = \sum_m w_m e^{-\|\mathbf{f}_m(i) - \mathbf{f}_m(j)\|^2}$$

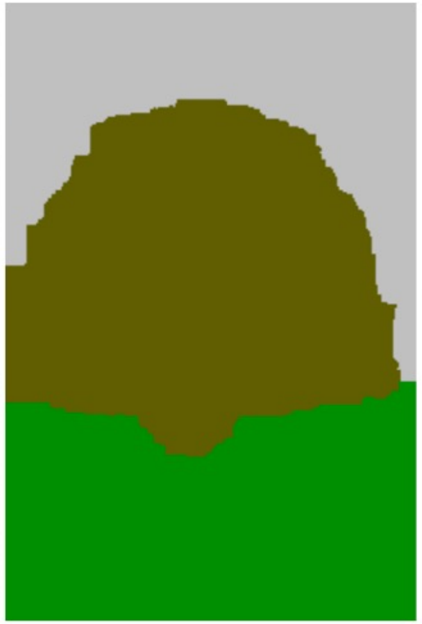
- What should **f** be?
- simple answer: color, position

Mean field inference for Dense-CRF

$$q_i(y_i = l) \propto \exp[\underbrace{-\psi_u(y_i)}_{\text{Unary}} - \sum_{l'} \underbrace{\mu(l,l')}_{\text{Label compatibility transform}} \underbrace{\sum_m w_m \sum_{j \neq i} e^{-\|\mathbf{f}_m(i)-\mathbf{f}_m(j)\|^2} q_j(y_j = l')}_{\text{Message passing}}]$$

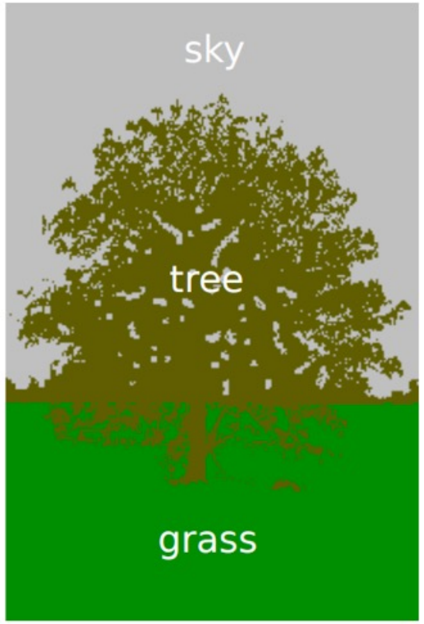
$$\mathbf{q}_i \propto \exp[-\boldsymbol{\psi}_u^{(i)} - \boldsymbol{\mu} \sum_j \mathbf{m}_{j \to i}]$$

Fully Connected CRFs



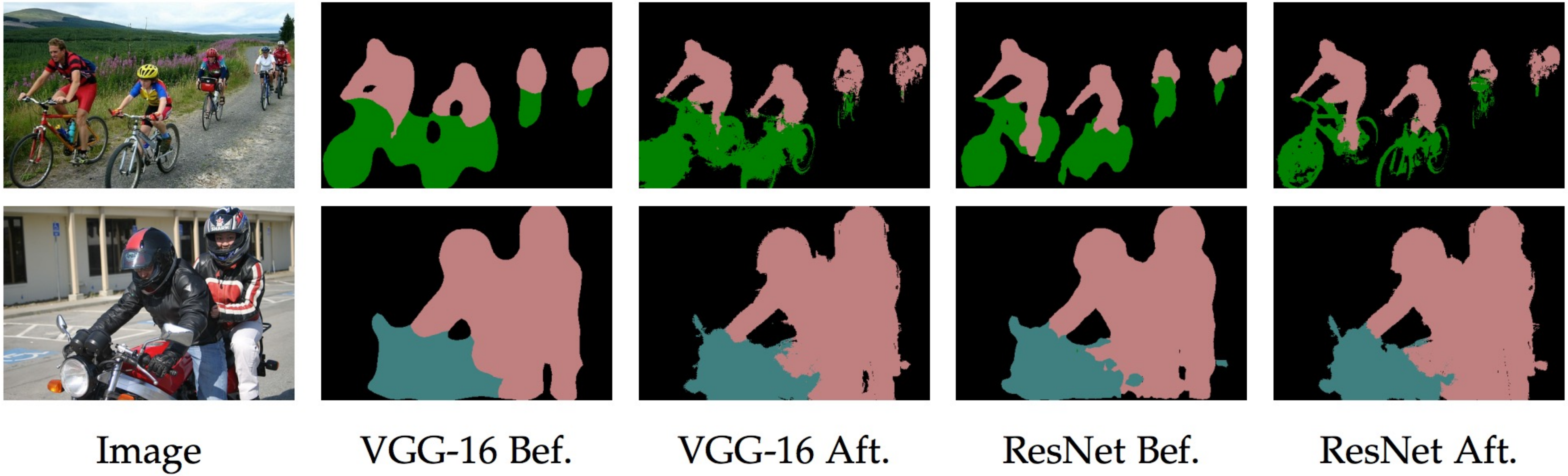
Grid CRF

Fully connected CRF



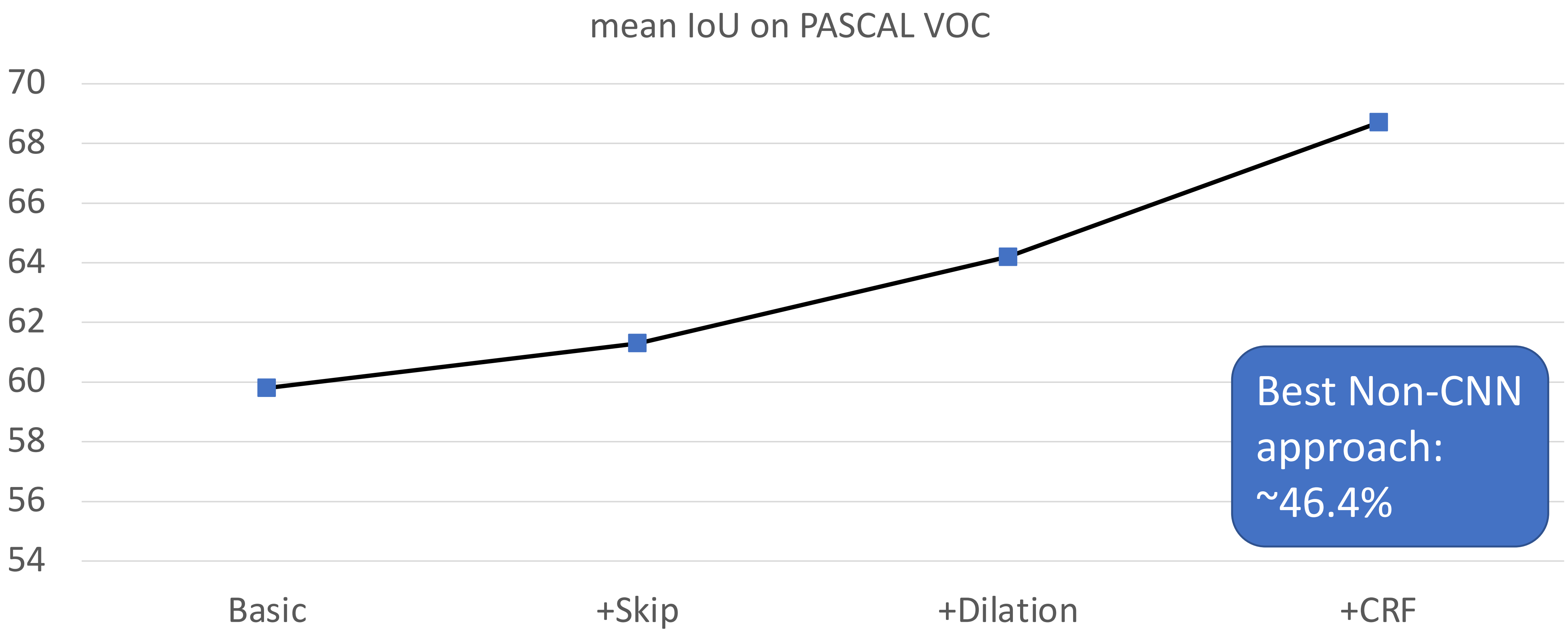
Ground truth

Fully connected CRFs



Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan Yuille. In *ICLR,* 2015.

Putting it all together



Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan Yuille. In *ICLR,* 2015.

Other additions

Method	mean IoU (%)
VGG16 + Skip + Dilation	65.8
ResNet101	68.7
ResNet101 + Pyramid	71.3
ResNet101 + Pyramid + COCO	74.9

DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan Yuille. Arxiv 2016.

Alternative: coarse-to-fine prediction

- Inspiration 1: we are making independent predictions from each layer
- Inspiration 2: CRF-like approaches require iterated inference
- Inspiration 3: Coarse-to-fine refinement works because: coarse scales capture large scale structure coarsely, fine scales capture fine-scale structures

U-Net

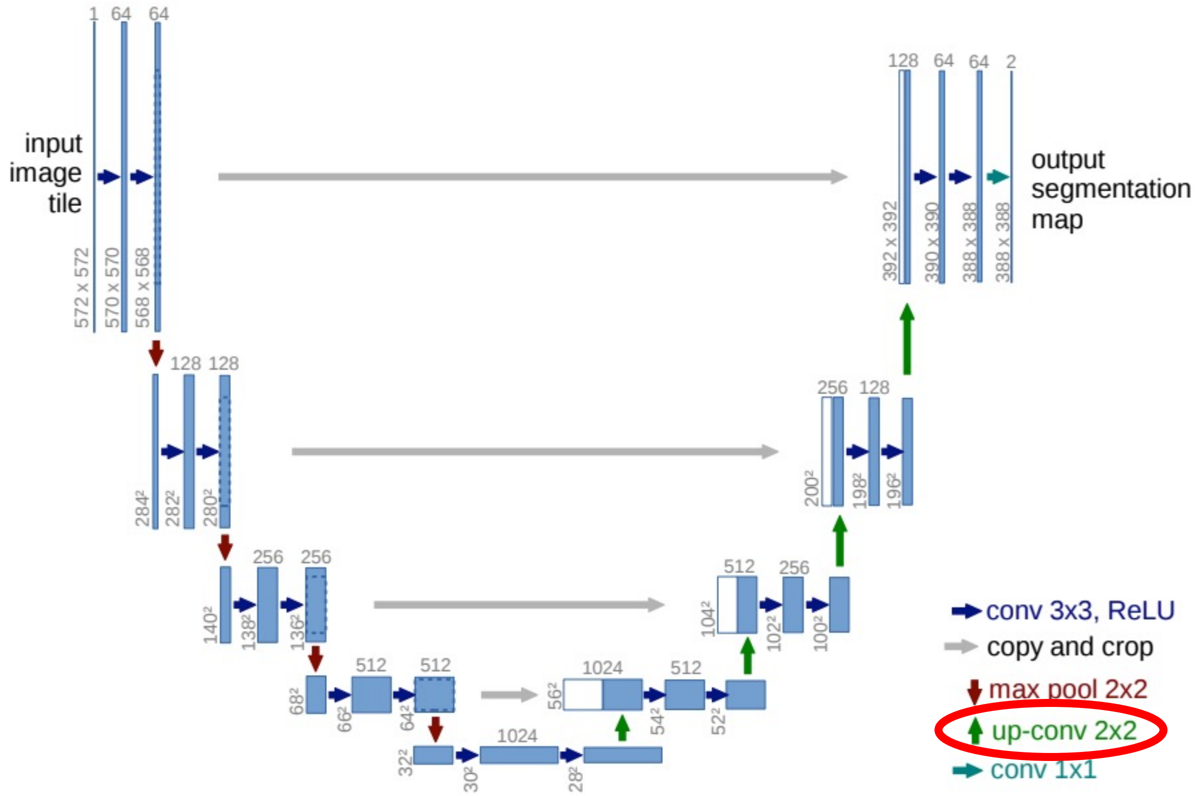


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.

Learned upsampling

- Bilinear upsampling

$$y[i] = \begin{cases} x[i/2] & \text{if } i \text{ is even} \\ \frac{(x[(i+1)/2]+x[(i-1)/2])}{2} & \text{ow} \end{cases}$$

- Assume fractional indices in x are 0
- Assume w[-1] = w[1] = 0.5, w[0] = 1
- Then

$$y[i] = \sum_{k=-1}^{1} w[k]x[(i-k)/2]$$

Learned upsampling

$$y[i] = \sum_{k=-1}^{1} w[k]x[(i-k)/2]$$

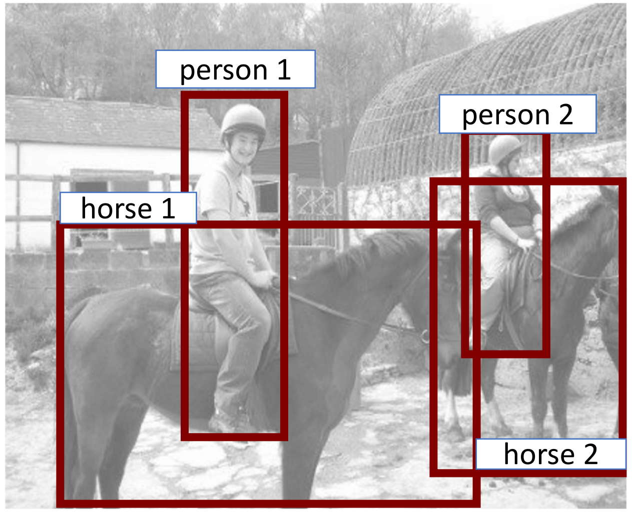
- Looks remarkably like convolution
- But output size is twice input size
- Filter w can be learnt!
- “Up-convolution”, “Transposed convolution”, ~~“Deconvolution”~~

Instance segmentation

Till now

horse, person

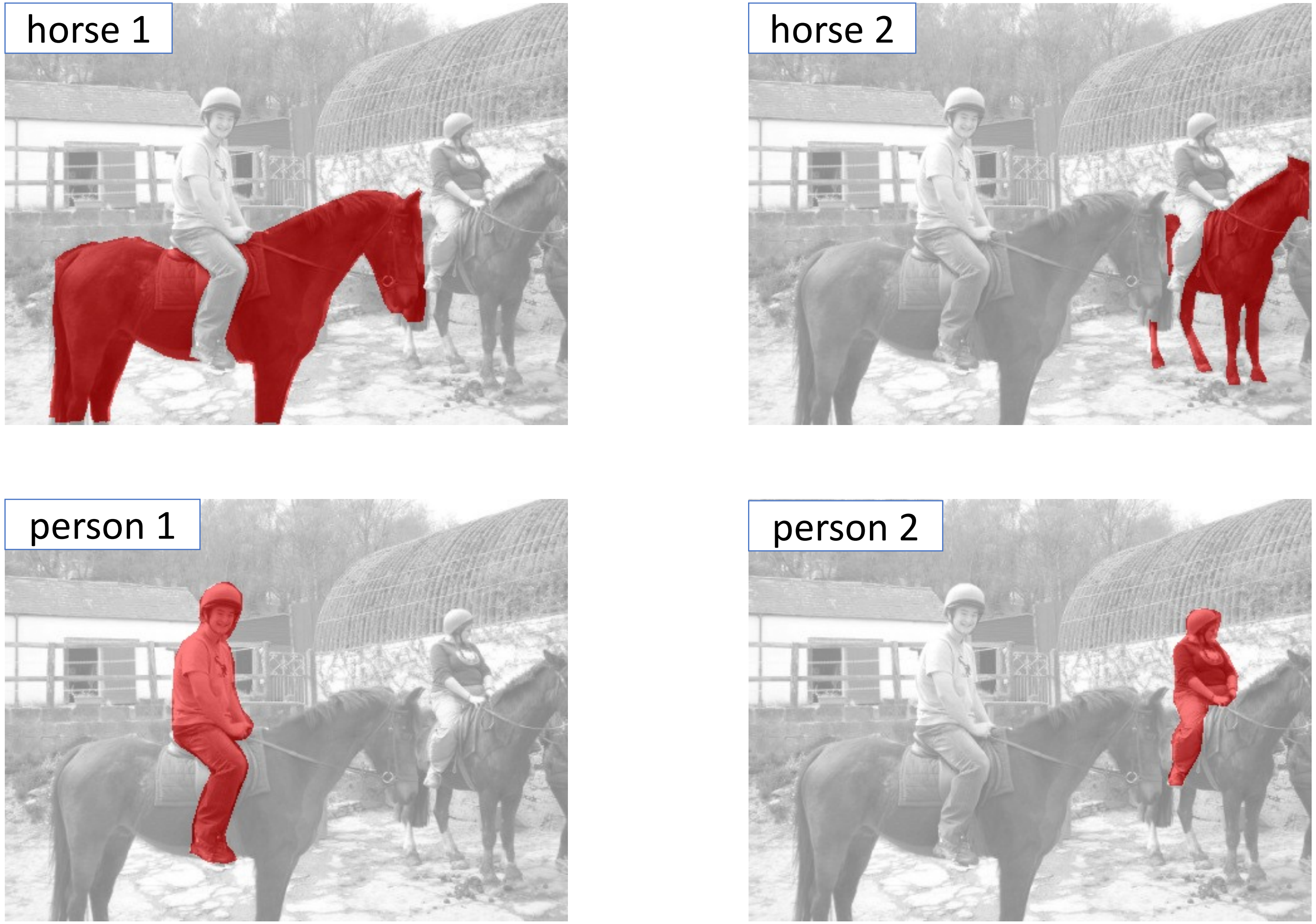
Image Classification



Object Detection

Semantic Segmentation

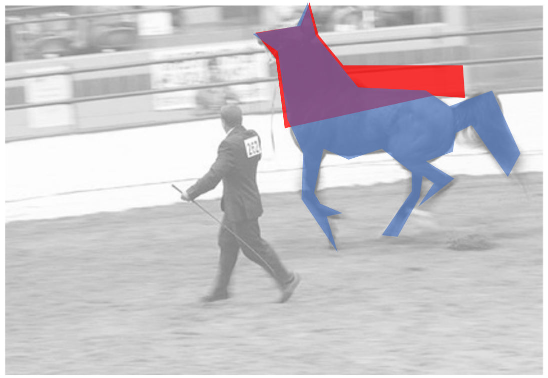
Fine-grained Localization



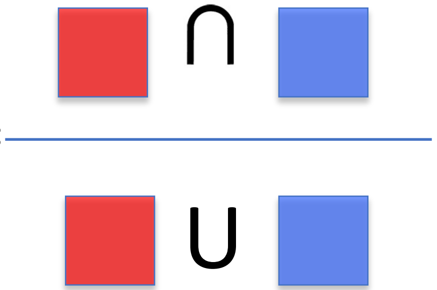
Instance Segmentation

Evaluation Protocol

- Sort predicted instances by confidence
- Match prediction to closest annotation based on *segment overlap*
  - If segment overlap > threshold, correct



segment overlap =

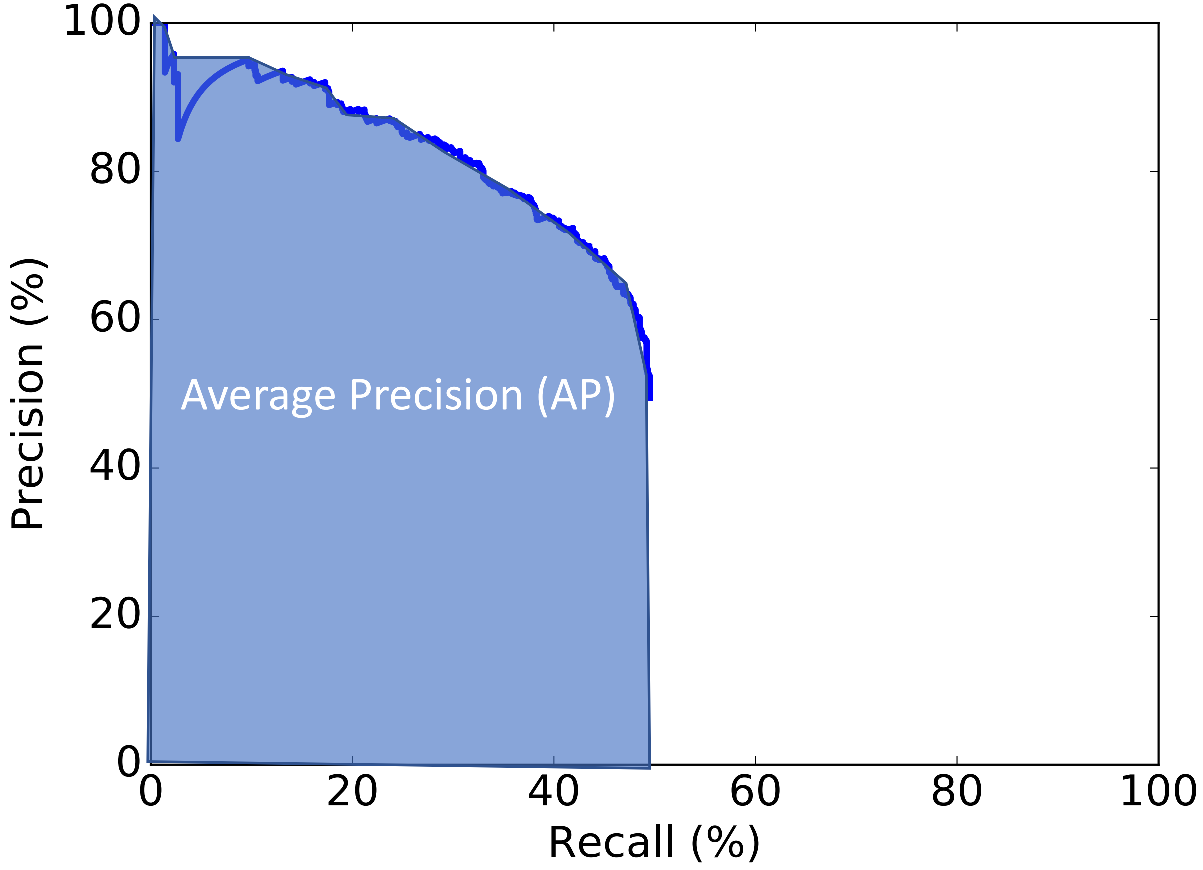


Evaluation Protocol

Labels = [✓ ✓ ✗ ✓ ✗ ✗ ✓ ....]

Scores = [0.90 0.87 0.82 0.78 0.70 0.69 0.60 ....]

Evaluation protocol



The COCO Challenge

mscoco.org

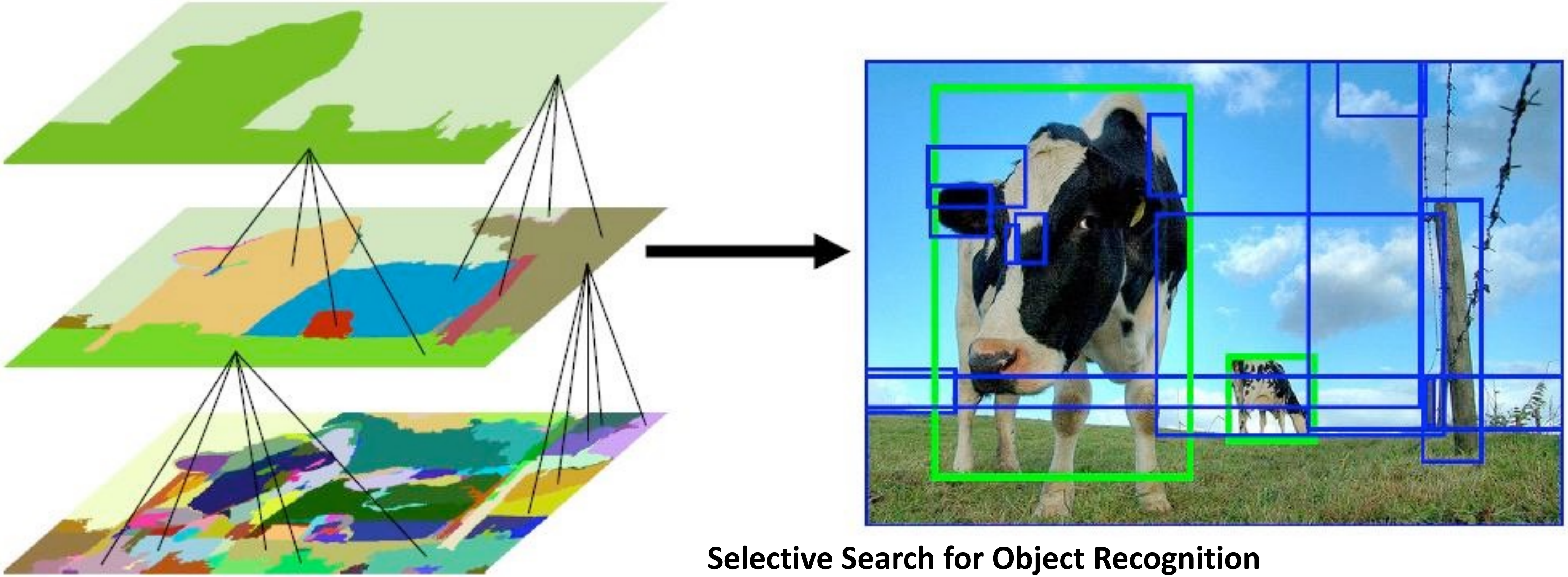
T. Y. Lin et al. **Microsoft COCO: Common Objects in Context**. In ECCV, 2014

Two strategies

- Segment then classify
  - Use bottom-up techniques to come up with *segment* proposals
  - Classify segment proposals with convnets
  - Segmentation is category agnostic
  - Modification: use convnets to produce segmentation proposals
- Detect then segment
  - Use standard object detection to produce boxes
  - Segment boxes
  - Segmentation is *category specific*

Box proposals

- Use segmentation to produce ~5K candidates

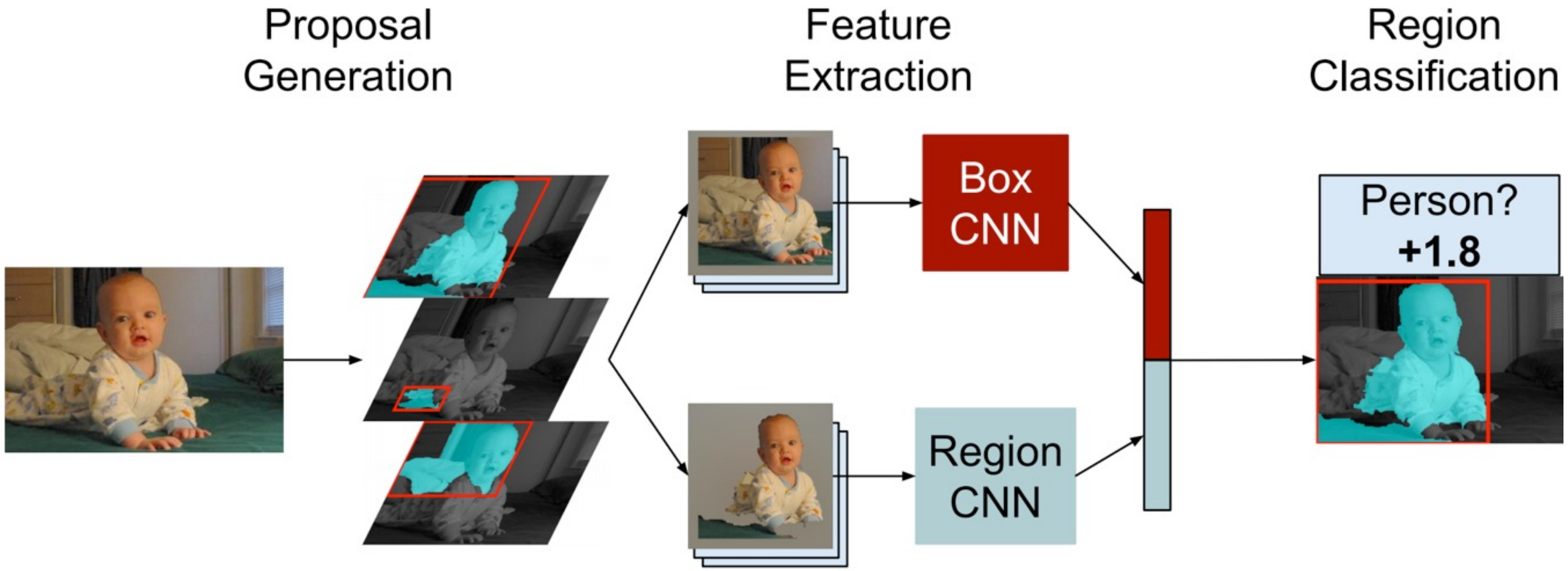


Selective Search for Object Recognition
J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, A. W. M. Smeulders
In International Journal of Computer Vision 2013.

Segment proposals

Multi-scale combinatorial grouping. Pablo Arbelaez, Jordi Pont-Tuset, Jonathan Barron, Ferran Marques, Jitendra Malik. In *CVPR,* 2014.

R-CNN for instance segmentation



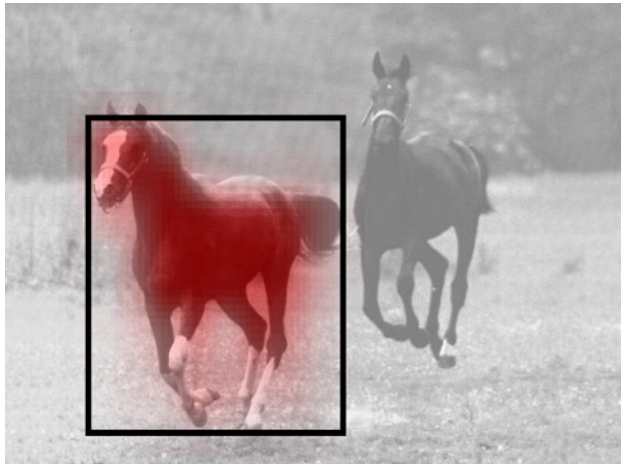
Simultaneous Detection and Segmentation Bharath Hariharan , Pablo Arbelaez , Ross Girshick, Jitendra Malik. In *ECCV* 2014

Two strategies

- Segment then classify
  - Use bottom-up techniques to come up with *segment* proposals
  - Classify segment proposals with convnets
  - Segmentation is category agnostic
  - Modification: use convnets to produce segmentation proposals
- Detect then segment
  - Use standard object detection to produce boxes
  - Segment boxes
  - Segmentation is *category specific*

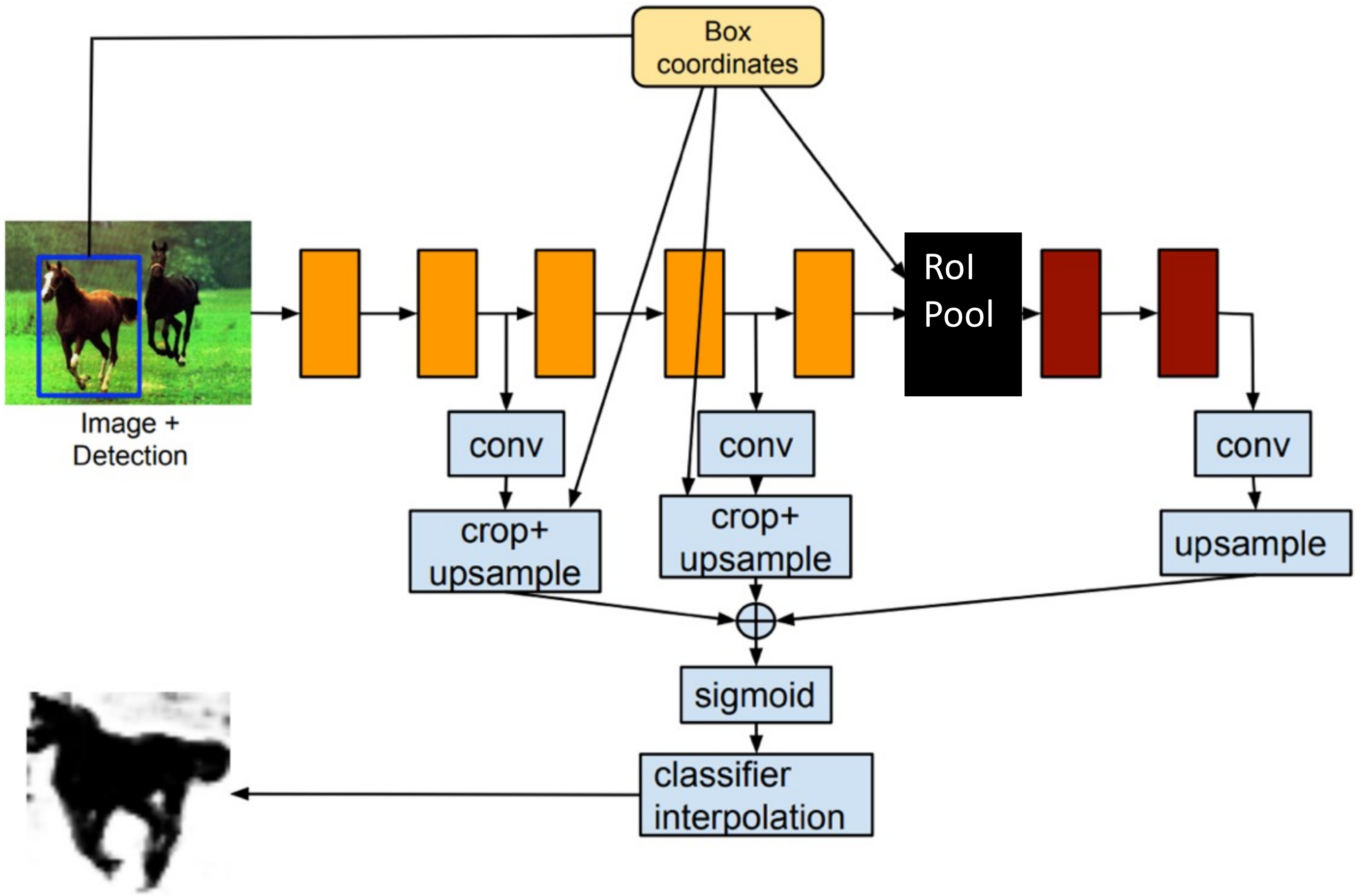
Detect then segment

- How should we segment a detected object?
- We have already computed features using ROI Pooling
- Idea: use features to predict mask!
  - Can either use a simple linear layer
  - Or can use convolution
  - Issue: can be very coarse



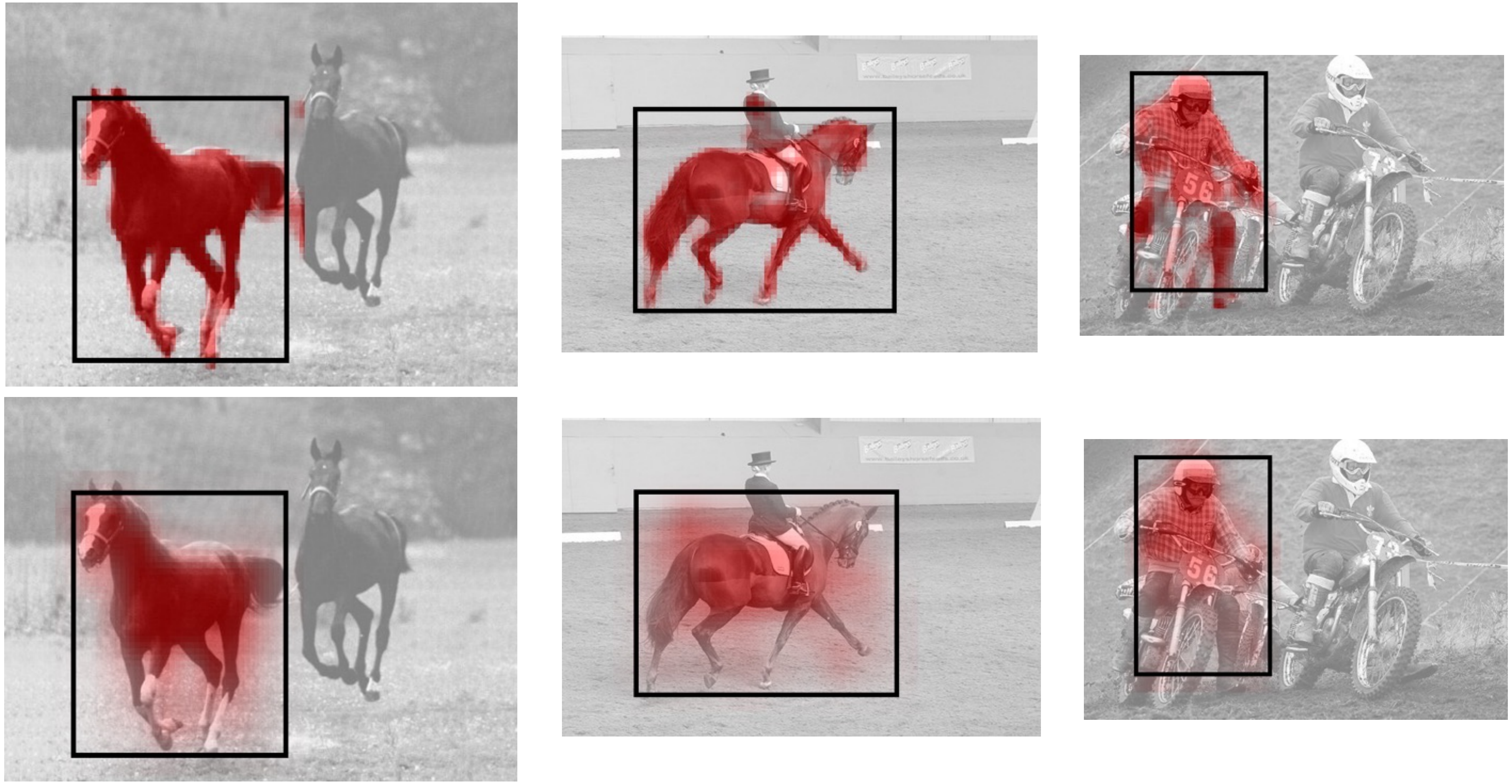
Skip connections with RoI pooling

- Finer-grained segmentation: tap into earlier layers



Hariharan, Bharath, et al. "Object instance segmentation and fine-grained localization using hypercolumns." *IEEE Transactions on Pattern Analysis & Machine Intelligence* 4 (2017): 627-639.

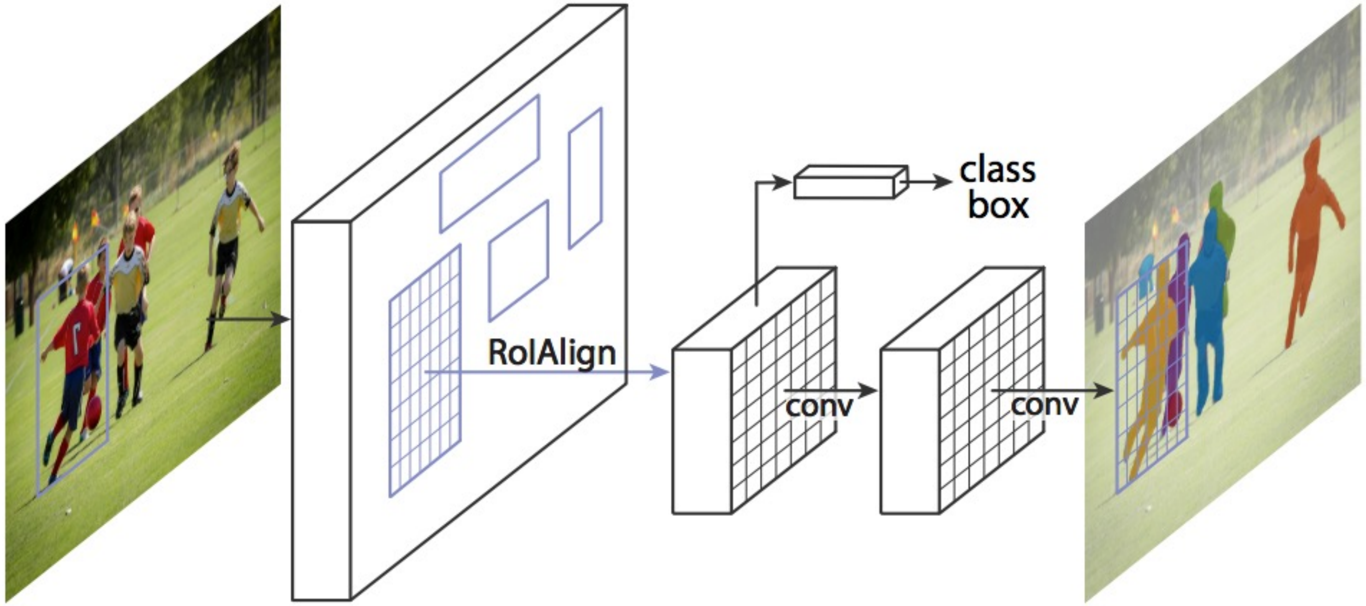
Skip connections for finer-grained details



Hypercolumns for Object Segmentation and Fine-grained Localization. Bharath Hariharan, Pablo Arbeláez, Ross Girshick, Jitendra Malik. CVPR, 2015

Mask R-CNN

- With deeper networks and ROI Align, skip connections not needed (?)



He, Kaiming, et al. "Mask r-cnn." *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017.

Final results - what works?

- First detect, then segment
- Big problem for instance segmentation is object detection
- Mask R-CNN (Faster R-CNN + convolution on RoI Pooled feature to get masks) is good starting point