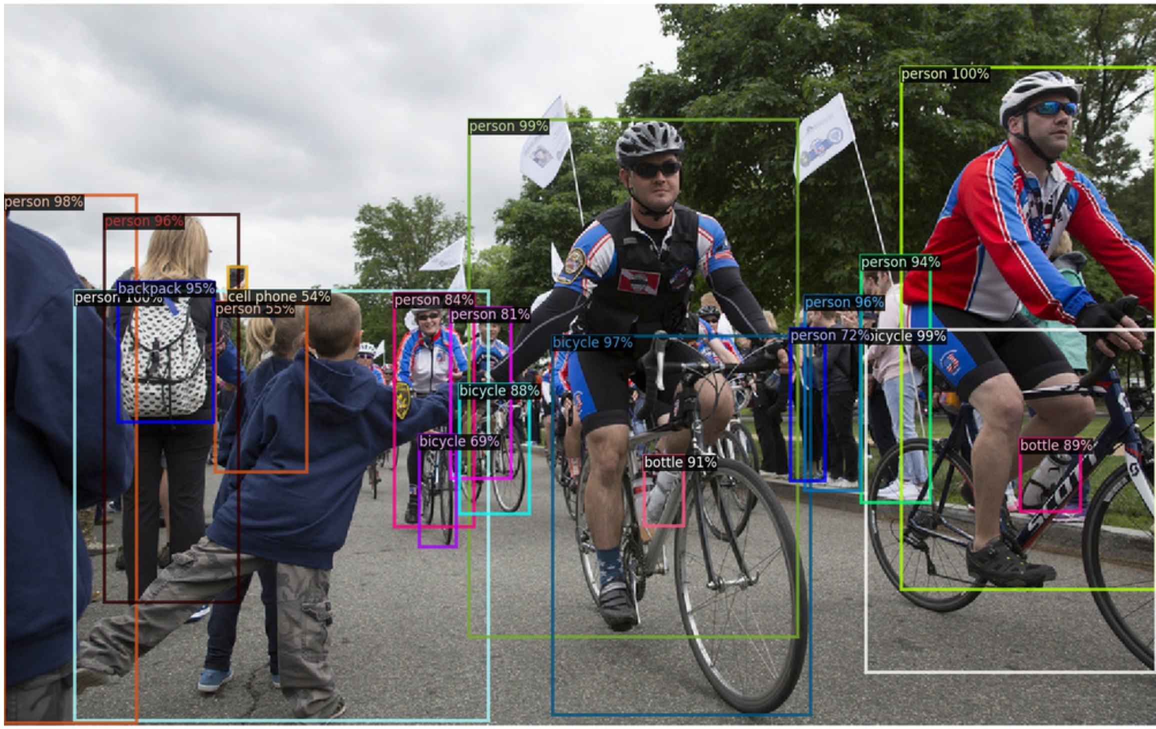
# CSE 152: Computer Vision

Hao Su

## Lecture 10: Object Recognition

# How do we represent objects

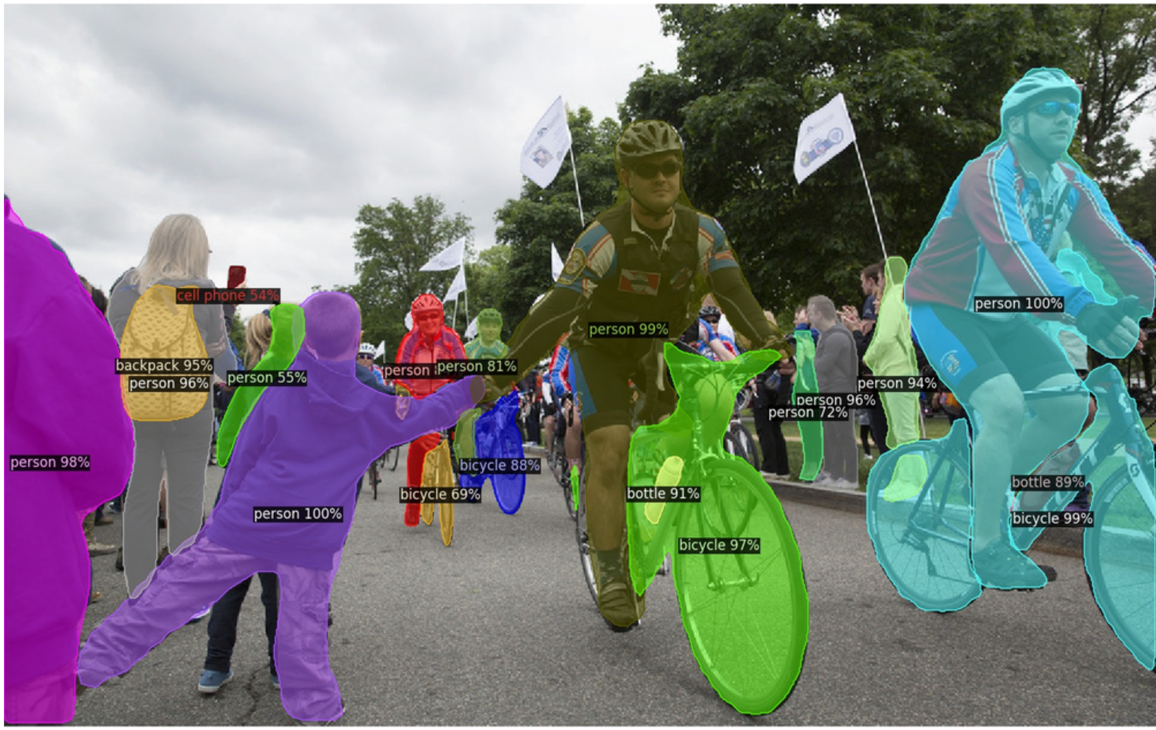
- Bounding box



Figures from https://github.com/facebookresearch/detectron2

# How do we represent objects

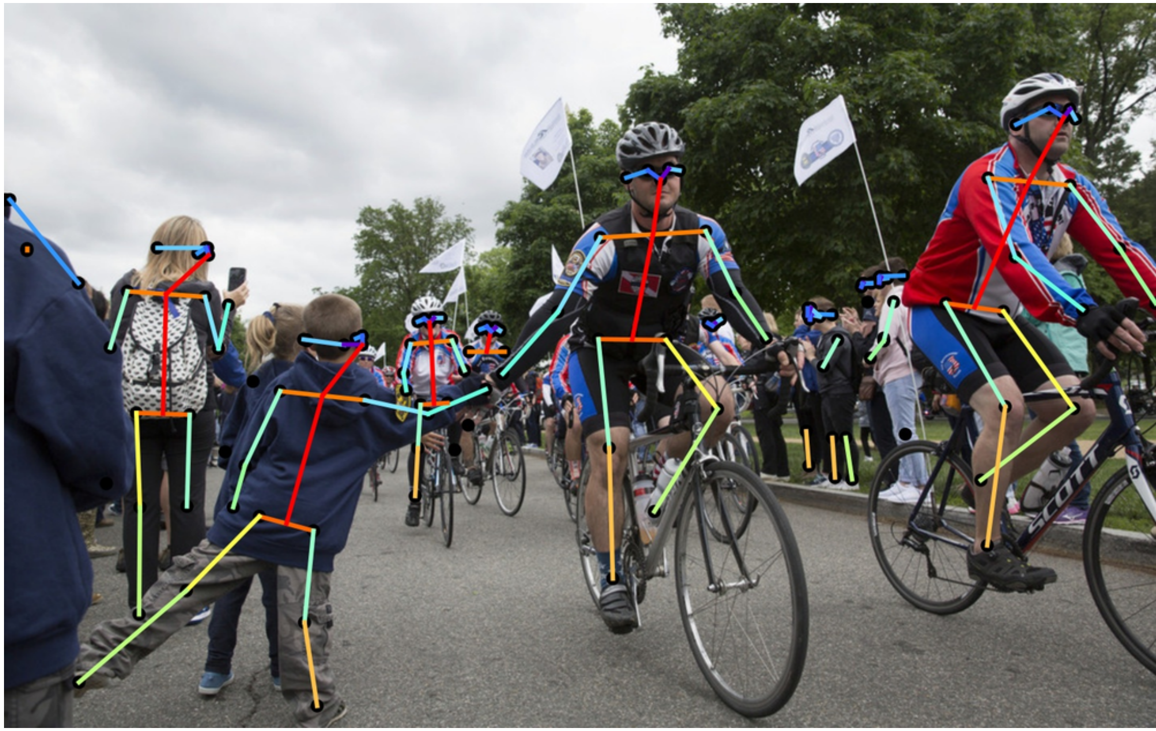
- Bounding box
- Instance mask



Figures from https://github.com/facebookresearch/detectron2

# How do we represent objects

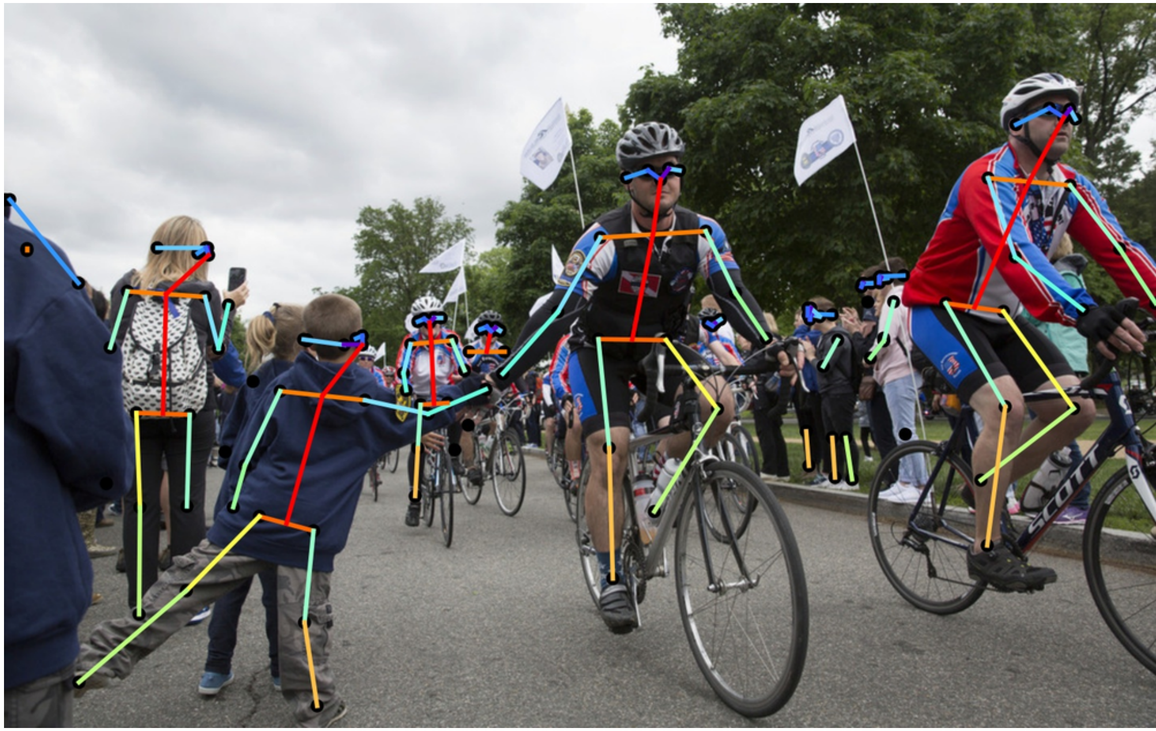
- Bounding box
- Instance mask
- Keypoint



Figures from https://github.com/facebookresearch/detectron2

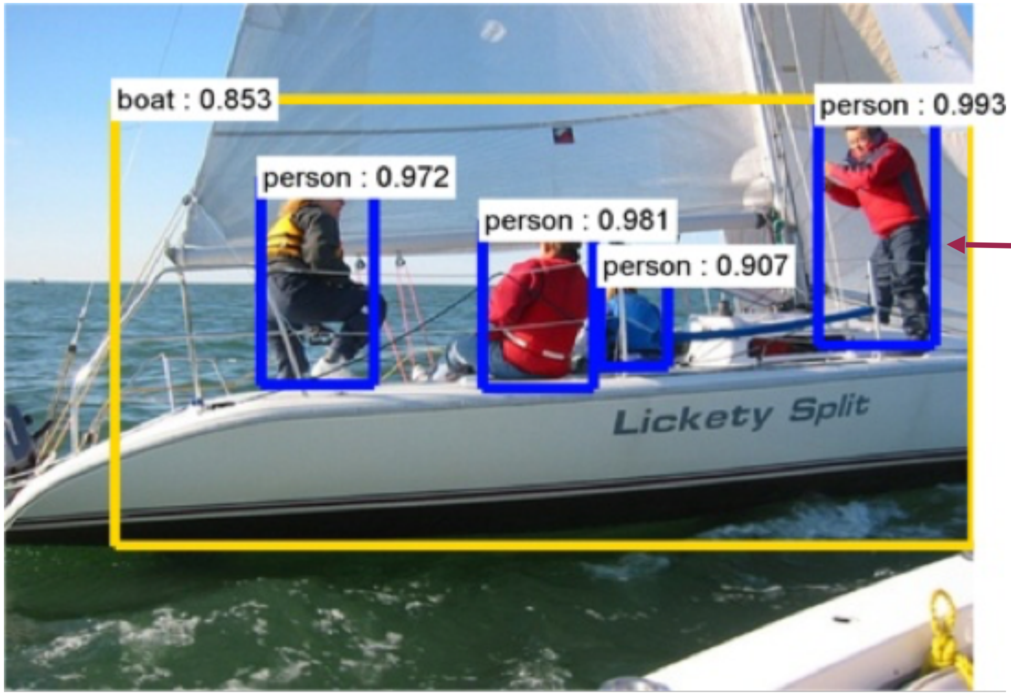
# How do we represent objects

- **Bounding box**
- Instance mask
- Keypoint



Figures from https://github.com/facebookresearch/detectron2

# Object Detection with Bounding Boxes



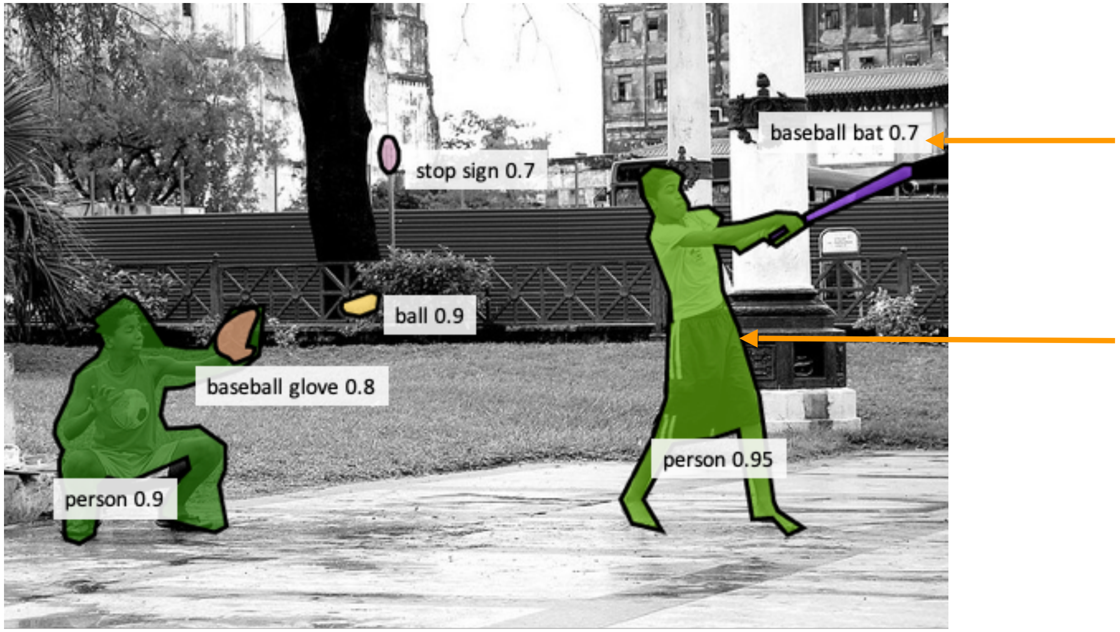
What? - Recognition/ Classification

Where? - Localization/ Regression

“Object detection”

Slides modified  from Ross Girshick tutorial at CVPR 2019

# Object Detection with Segmentation Masks



What? - Recognition

Where? - Segmentation

"Instance segmentation"

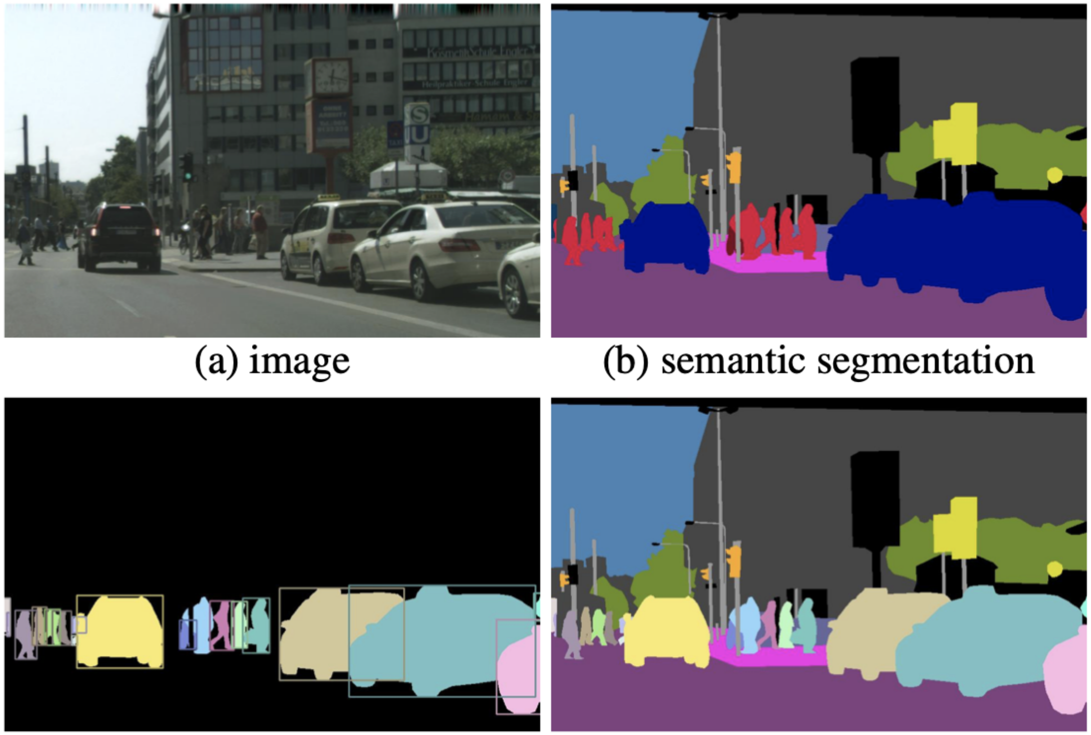
Slides modified from Ross Girshick tutorial at CVPR 2019

# Semantic Segmentation

Predict a pixel-wise class label

Stuff: walls, buildings, sky, road

Things: human, cars, bikes



(a) image

(b) semantic segmentation

(c) instance segmentation

(d) panoptic segmentation

Figures from *Panoptic Segmentation*, CVPR 2019

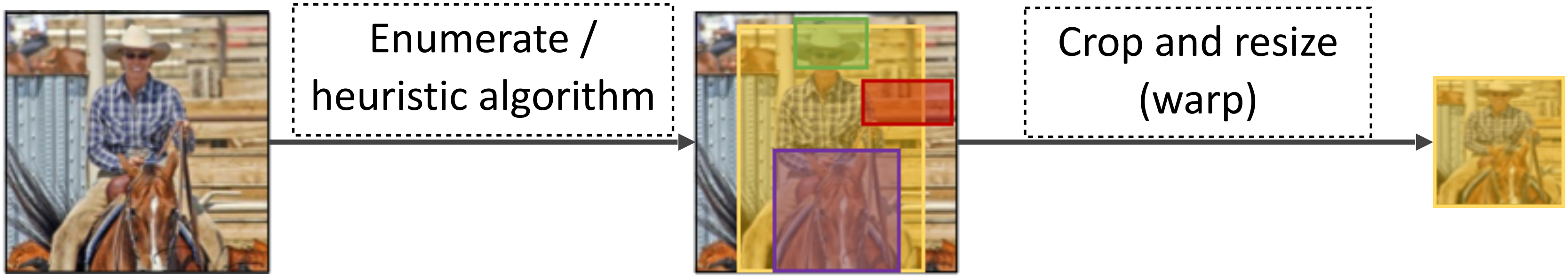
# Datasets

Microsoft
COCO

**Visual Object Classes Challenge 2012 (VOC2012)**

# Object Detection

# Object Detection → Object Classification



Enumerate / heuristic algorithm

Crop and resize (warp)

Input: an image

Proposals/Candidates

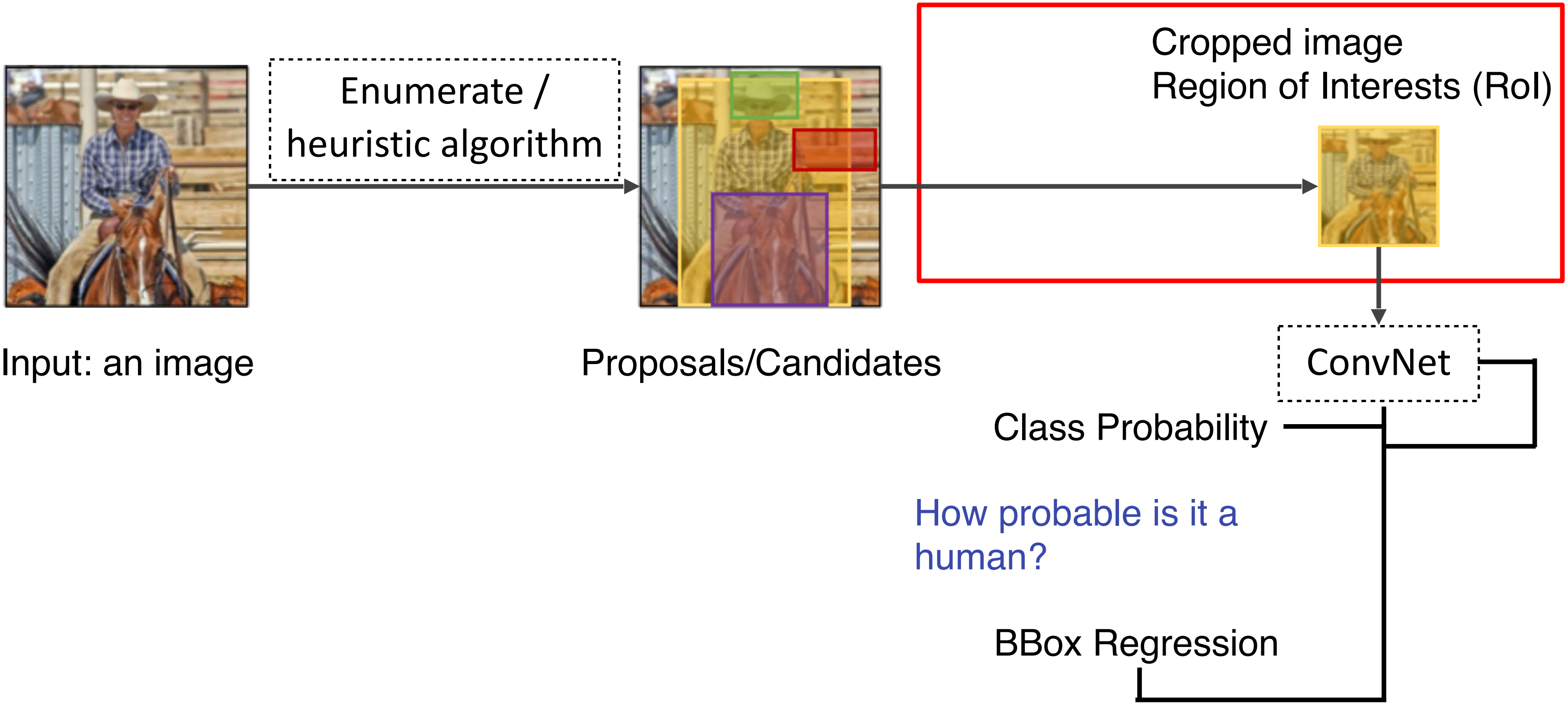
Cropped image

We’ve already reduced object detection to object classification!

Slides modified  from Ross Girshick tutorial at CVPR 2019

# R-CNN (Regional ConvNet)

Computationally expensive



Enumerate / heuristic algorithm

Cropped image
Region of Interests (RoI)

Input: an image

Proposals/Candidates

ConvNet

Class Probability

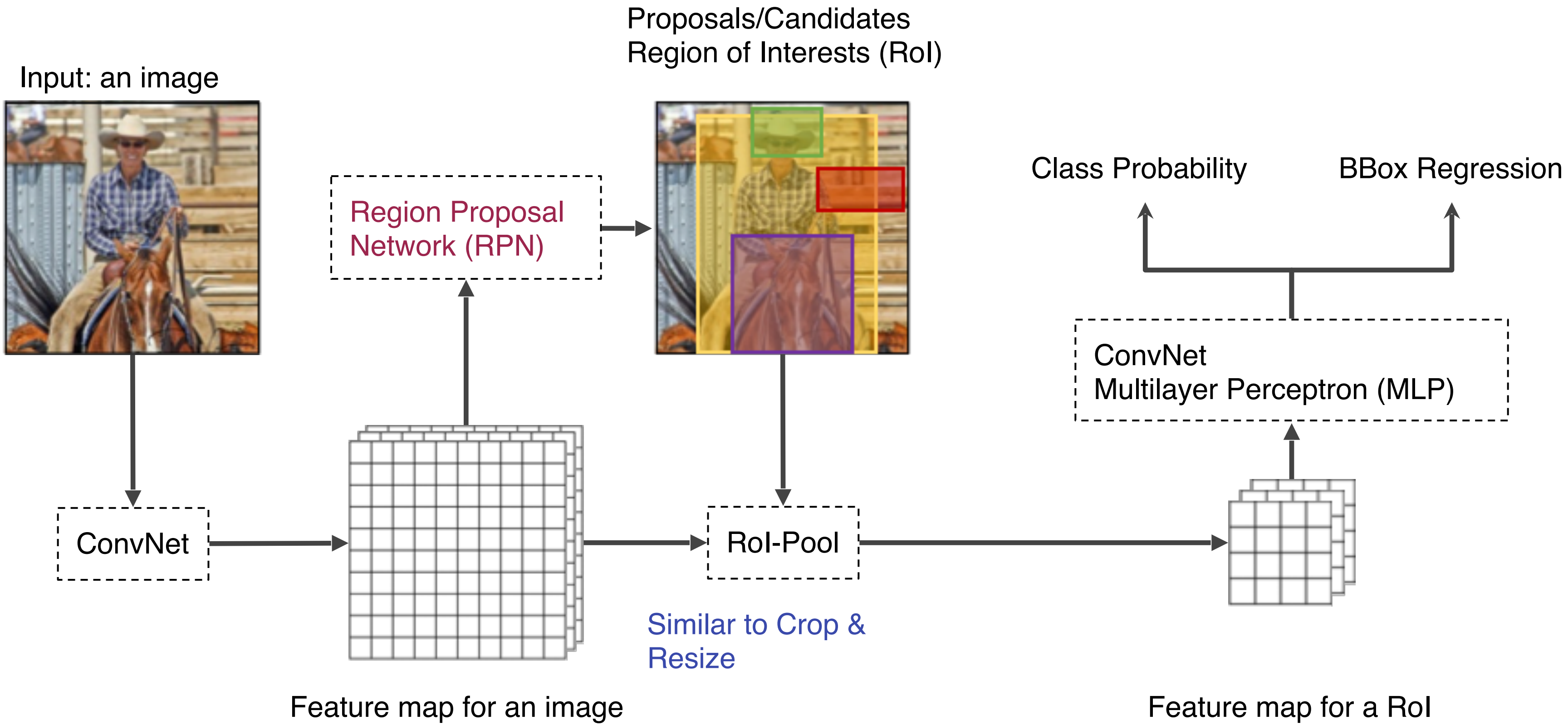
How probable is it a human?

BBox Regression

How can we modify this bounding box?

Slides modified  from Ross Girshick tutorial at CVPR 2019

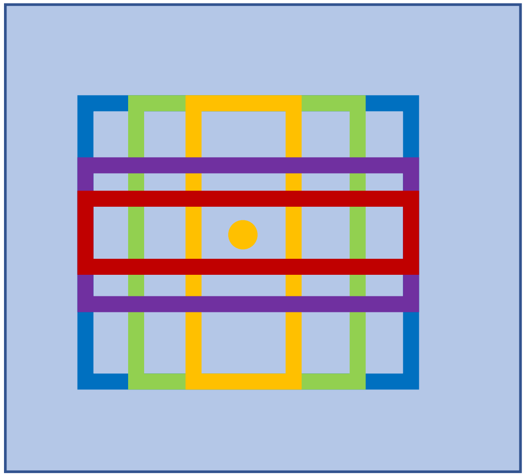
# Faster R-CNN



Slides modified  from Ross Girshick tutorial at CVPR 2019

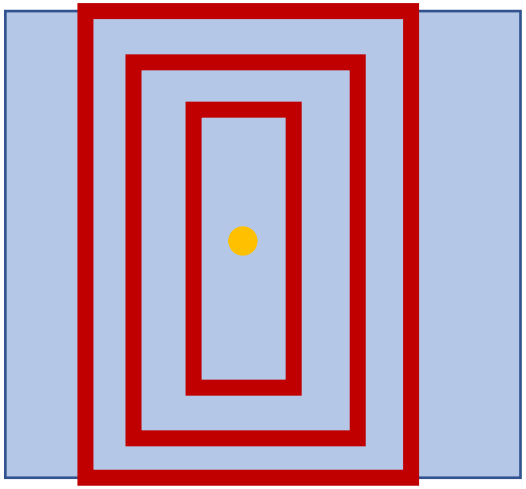
# Faster R-CNN

- At each location, consider boxes of many different sizes and aspect ratios



# Faster R-CNN

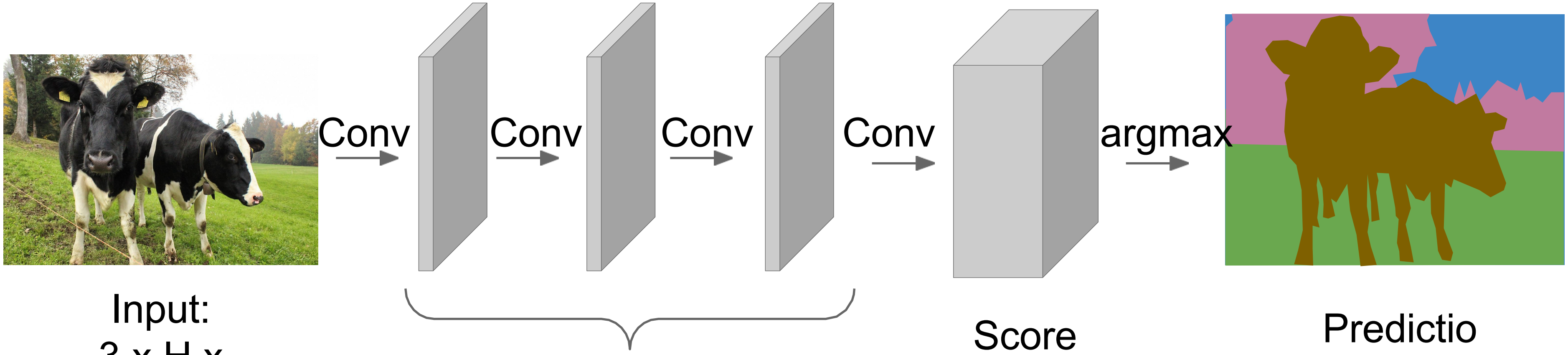
- At each location, consider boxes of many different sizes and aspect ratios



# Object Segmentation

# Semantic Segmentation Idea: Fully Convolutional

Design a network as a bunch of convolutional layers to make predictions for pixels all at once!



Conv

Conv

Conv

Conv

argmax

Input:
3 x H x
W

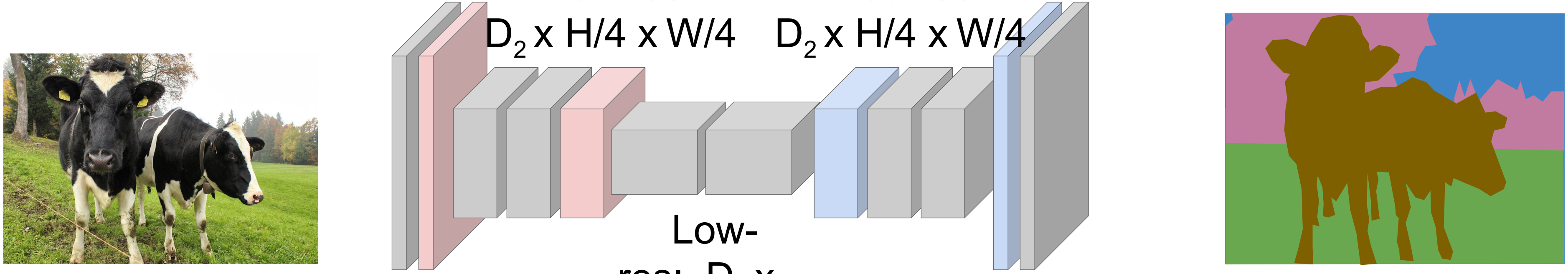
Convolutio
ns: D x
H x W

Score
s: C x
H x W

Predictio
ns: H
x W

# Semantic Segmentation Idea: Fully Convolutional

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!



Med-res: $D_2$ x H/4 x W/4

Med-res: $D_2$ x H/4 x W/4

Input: 3 x H x W

High-res: $D_1$ x H/2 x W/2

Low-res: $D_3$ x H/4 x W/4

High-res: $D_1$ x H/2 x W/2

Predictions: H x W

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

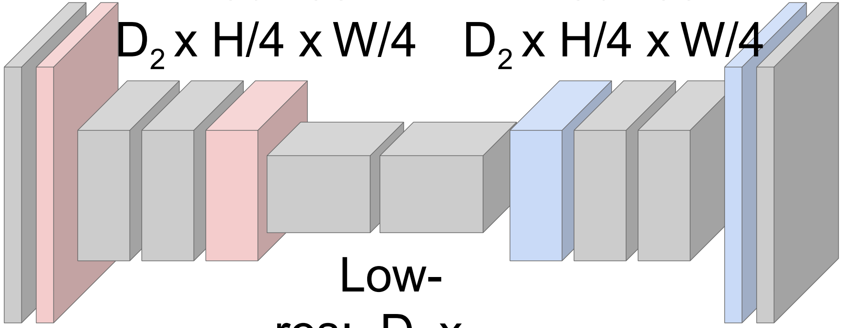
# Semantic Segmentation Idea: Fully Convolutional

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

**Downsampling**: Pooling, strided convolution

**Upsampling**: ???

Input: 3 x H x W



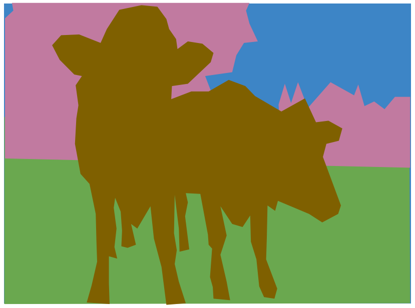
High-res: $D_1$ x H/2 x W/2

Med-res: $D_2$ x H/4 x W/4

Low-res: $D_3$ x H/4 x W/4

Med-res: $D_2$ x H/4 x W/4

High-res: $D_1$ x H/2 x W/2

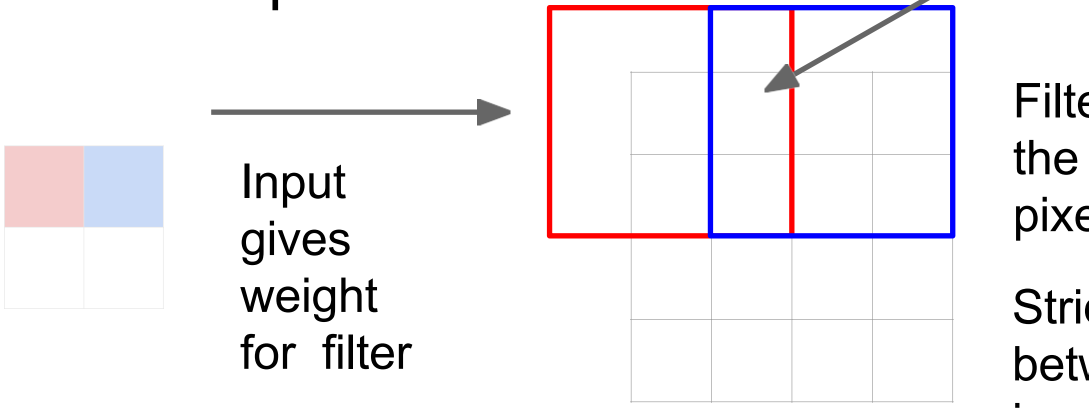


Predictions: H x W

Long, Shelhamer, and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015
Noh et al, “Learning Deconvolution Network for Semantic Segmentation”, ICCV 2015

# Learnable Upsampling: Transpose Convolution

3 x 3 **transpose** convolution,
stride 2 pad 1



Input gives weight for filter

Input: 2 x 2

Output: 4 x 4

Sum where output overlaps

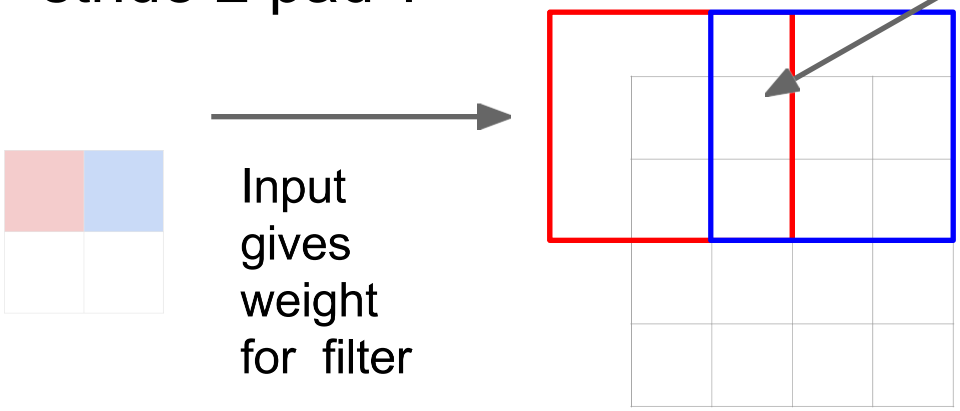
Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input

# Learnable Upsampling: Transpose Convolution

3 x 3 **transpose** convolution, stride 2 pad 1

Sum where output overlaps



Input gives weight for filter

Input: 2 x 2

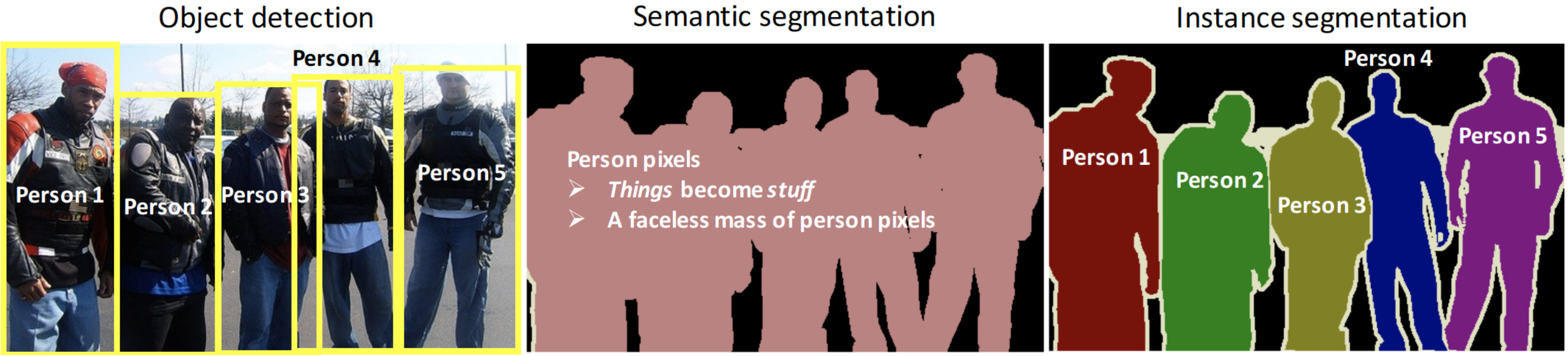
Output: 4 x 4

Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input

**Other names:**
-Deconvolution (bad)
-Upconvolution
-Fractionally strided convolution
-Backward strided convolution

# Semantic vs. Instance Segmentation



Object detection

Semantic segmentation

Instance segmentation

Person pixels

- *Things* become *stuff*
- A faceless mass of person pixels

Slides modified from Ross Girshick tutorial at CVPR 2019

# Mask R-CNN

- First do object detection using the Faster R-CNN arch, and then do semantic segmentation inside the cropped region

- Share features of the first few layers for detection and segmentation