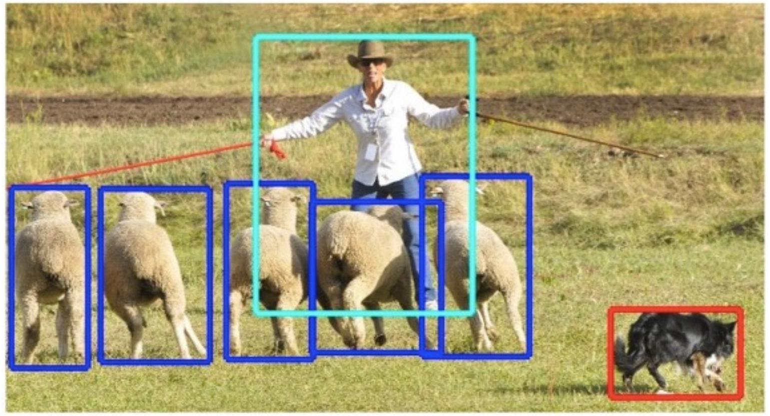
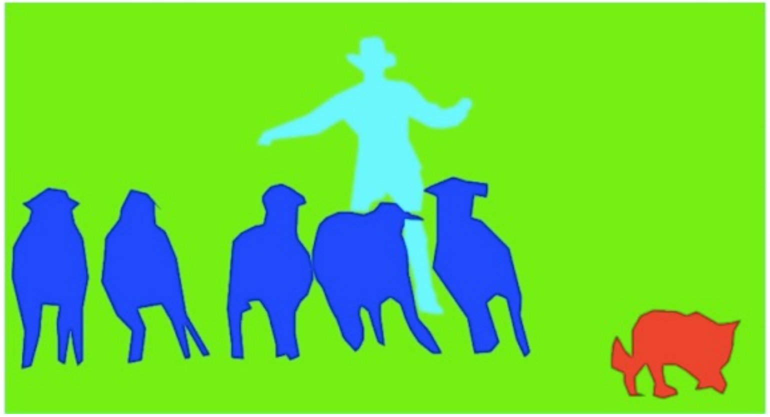
CNNs for dense image labeling

image classification



object detection



semantic segmentation

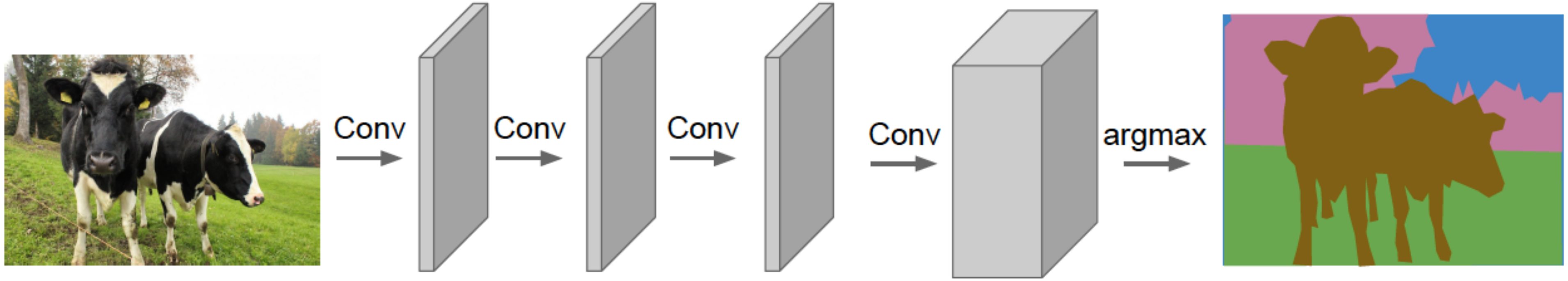
instance segmentation

Outline

- Fully convolutional networks
- Operations for dense prediction
    - Transposed convolutions, unpooling
- Architectures for dense prediction
    - DeconvNet, SegNet, U-Net
- Instance segmentation
    - Mask R-CNN
- Other dense prediction problems

Fully convolutional networks

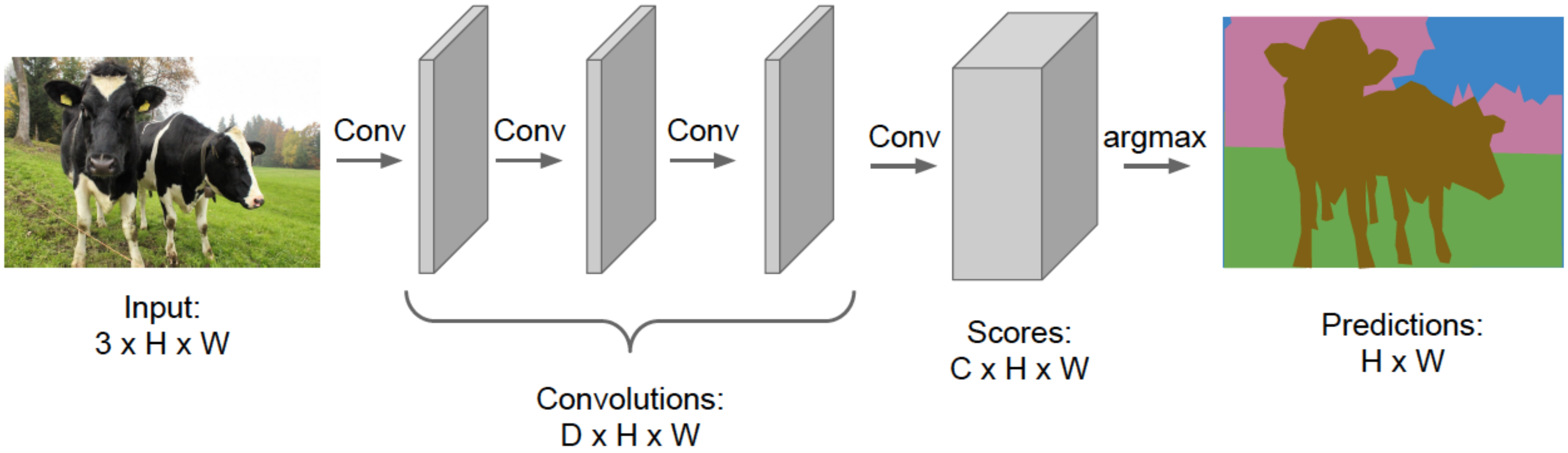
- Design a network with only convolutional layers, make predictions for all pixels at once



Source: [Stanford CS231n]

Fully convolutional networks

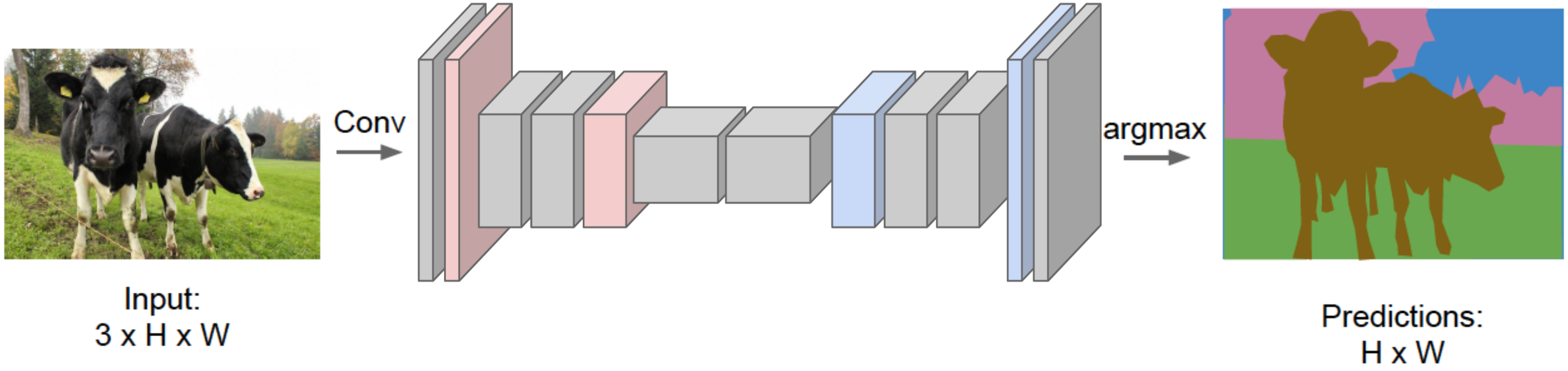
- Design a network with only convolutional layers, make predictions for all pixels at once
- Can the network operate at full image resolution?



Source: [Stanford CS231n]

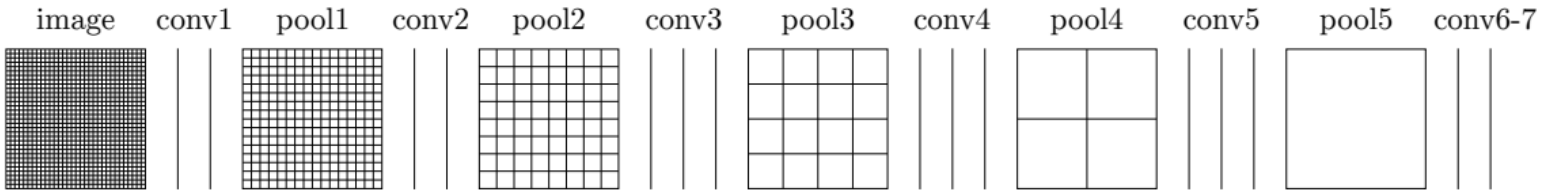
Fully convolutional networks

- Design a network with only convolutional layers, make predictions for all pixels at once
- Can the network operate at full image resolution?
- Practical solution: first downsample, then upsample



Source: [Stanford CS231n]

Fully convolutional networks (FCN)

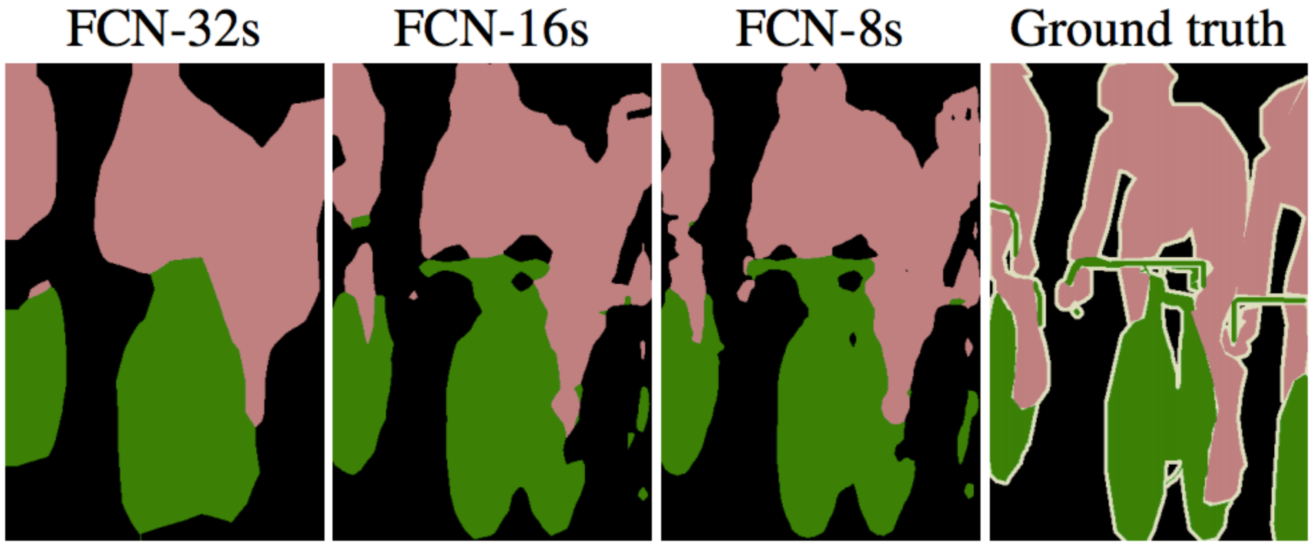


- Predictions by 1x1 conv layers, bilinear upsampling to original image resolution

- Predictions by 1x1 conv layers, learned 2x upsampling using *transposed convolutions*, fusion by summing

J. Long, E. Shelhamer, and T. Darrell, Fully Convolutional Networks for Semantic Segmentation, CVPR 2015

Fully convolutional networks (FCN)



Comparison on a subset of PASCAL 2011 validation data:

	pixel acc.	mean acc.	mean IU
FCN-32s-fixed	83.0	59.7	45.4
FCN-32s	89.1	73.3	59.4
FCN-16s	90.0	75.7	62.4
FCN-8s	**90.3**	**75.9**	**62.7**

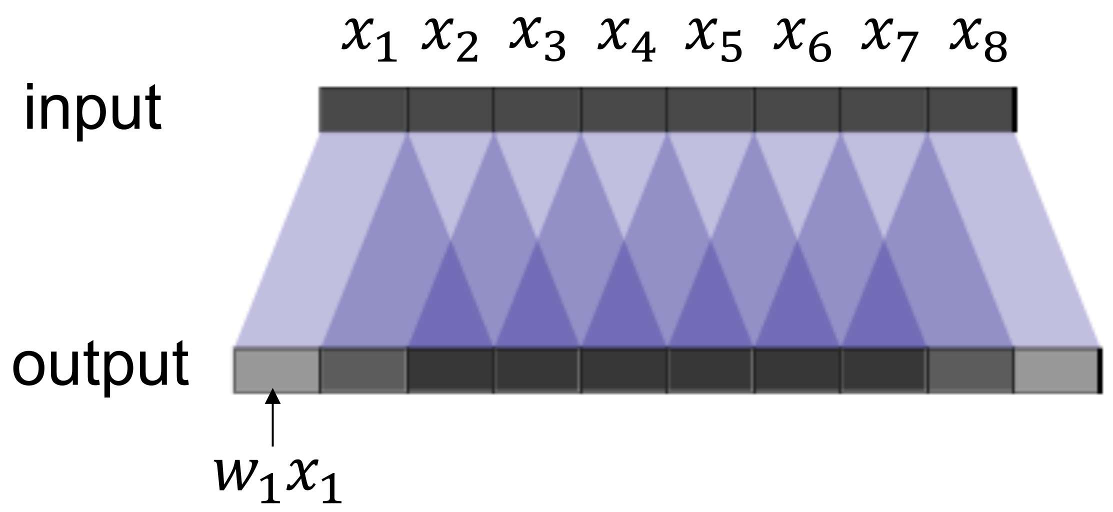
J. Long, E. Shelhamer, and T. Darrell, [Fully Convolutional Networks for Semantic Segmentation](), CVPR 2015

Outline

- Fully convolutional networks
- Operations for dense prediction
    - Transposed convolutions, unpooling

Transposed convolution

- Use the filter to “paint” in the output: place copies of the filter on the output, multiply by corresponding value in the input, sum where copies of the filter overlap
- 1D example:

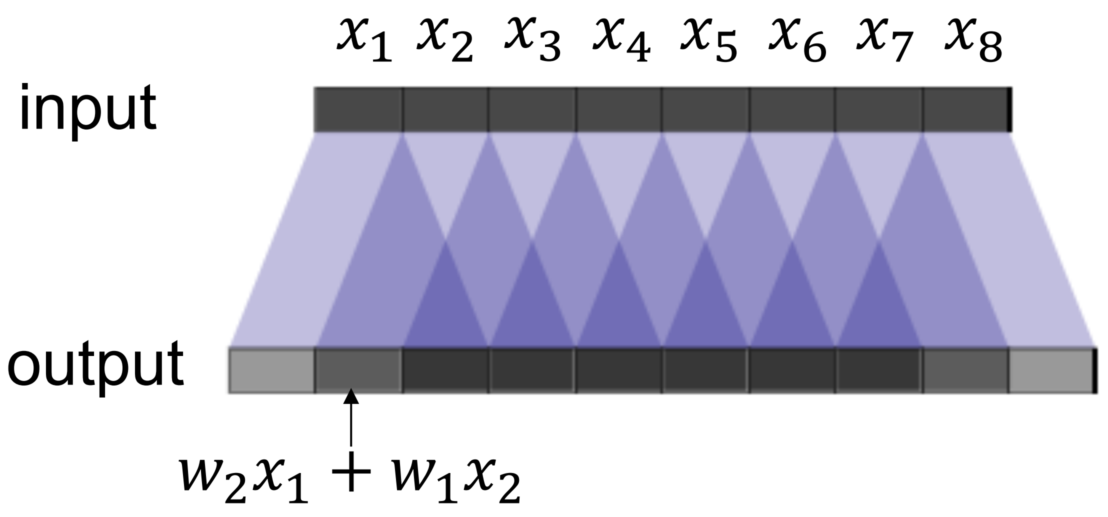


filter: $[w_1 \quad w_2 \quad w_3]$

Animation: https://distill.pub/2016/deconv-checkerboard/

Transposed convolution

- Use the filter to “paint” in the output: place copies of the filter on the output, multiply by corresponding value in the input, sum where copies of the filter overlap
- 1D example:

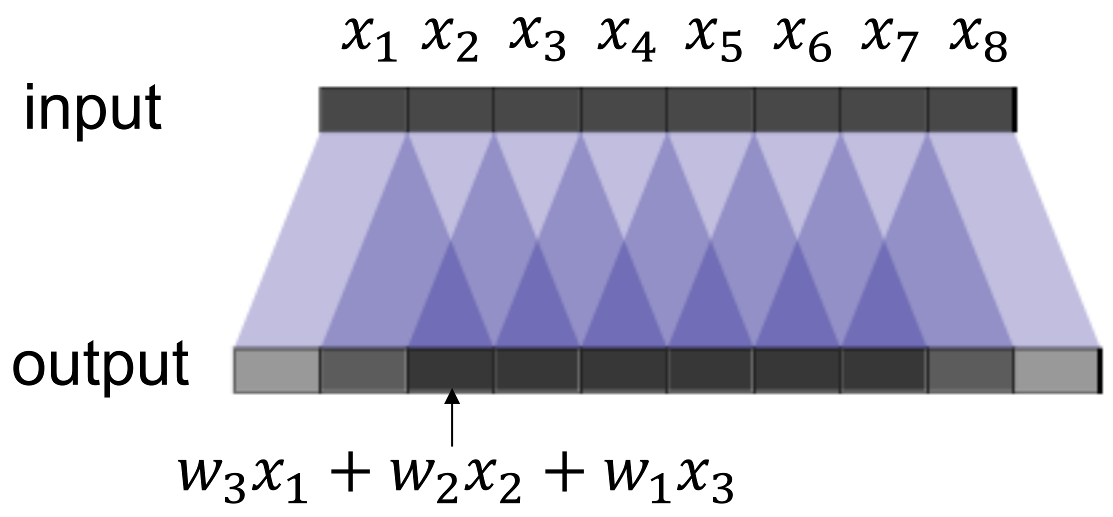


filter: $[w_1 \quad w_2 \quad w_3]$

Animation: https://distill.pub/2016/deconv-checkerboard/

Transposed convolution

- Use the filter to “paint” in the output: place copies of the filter on the output, multiply by corresponding value in the input, sum where copies of the filter overlap
- 1D example:



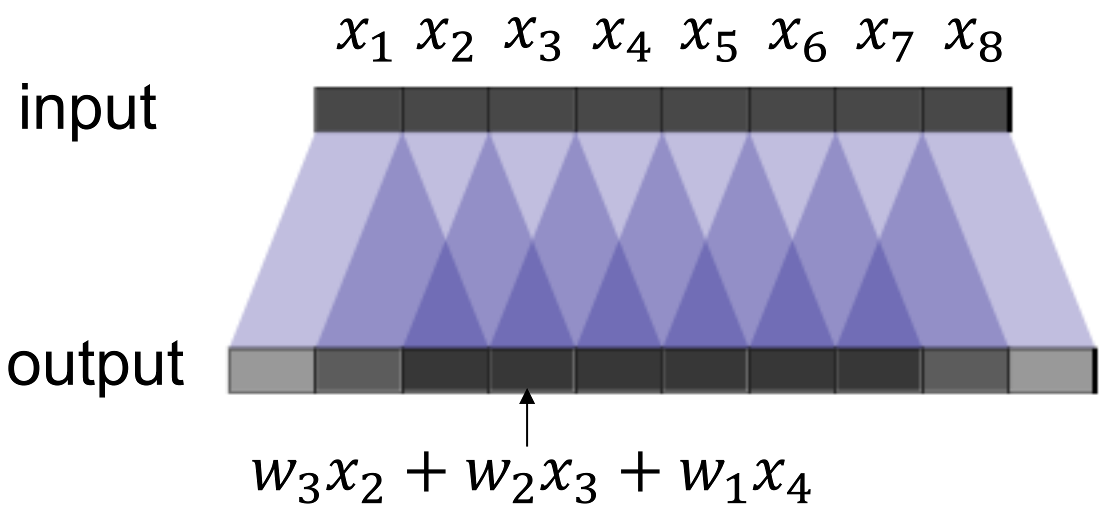
filter: $[w_1 \quad w_2 \quad w_3]$

Same as convolution with a flipped filter!

Animation: https://distill.pub/2016/deconv-checkerboard/

Transposed convolution

- Use the filter to "paint" in the output: place copies of the filter on the output, multiply by corresponding value in the input, sum where copies of the filter overlap
- 1D example:



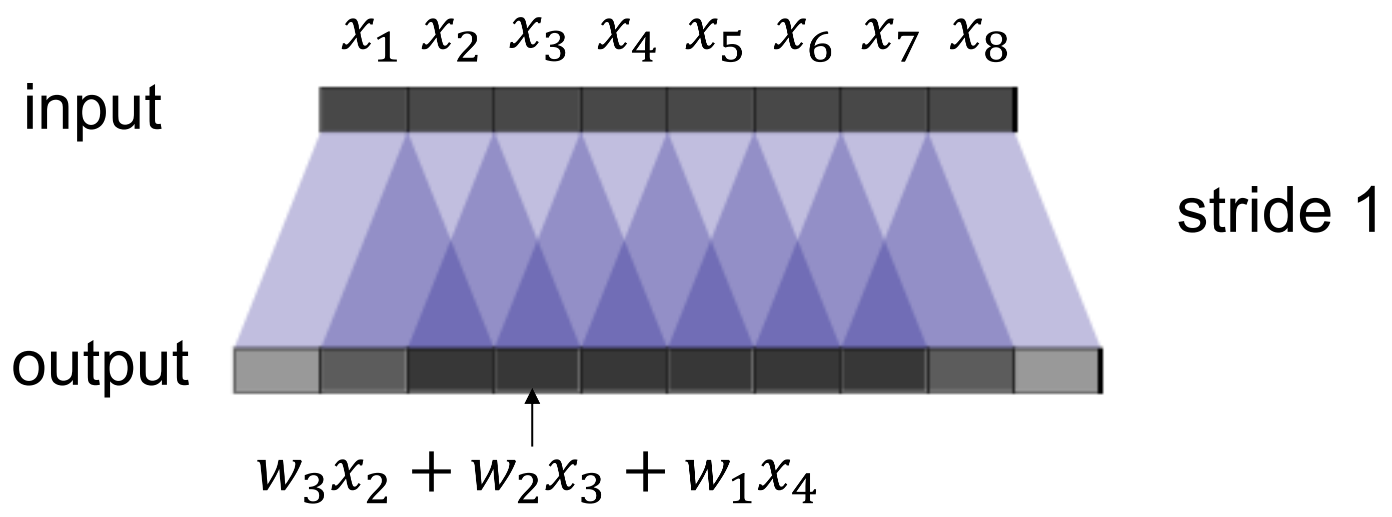
filter: $[w_1 \quad w_2 \quad w_3]$

Same as convolution with a flipped filter!

Animation: https://distill.pub/2016/deconv-checkerboard/

Upsampling by transposed convolution

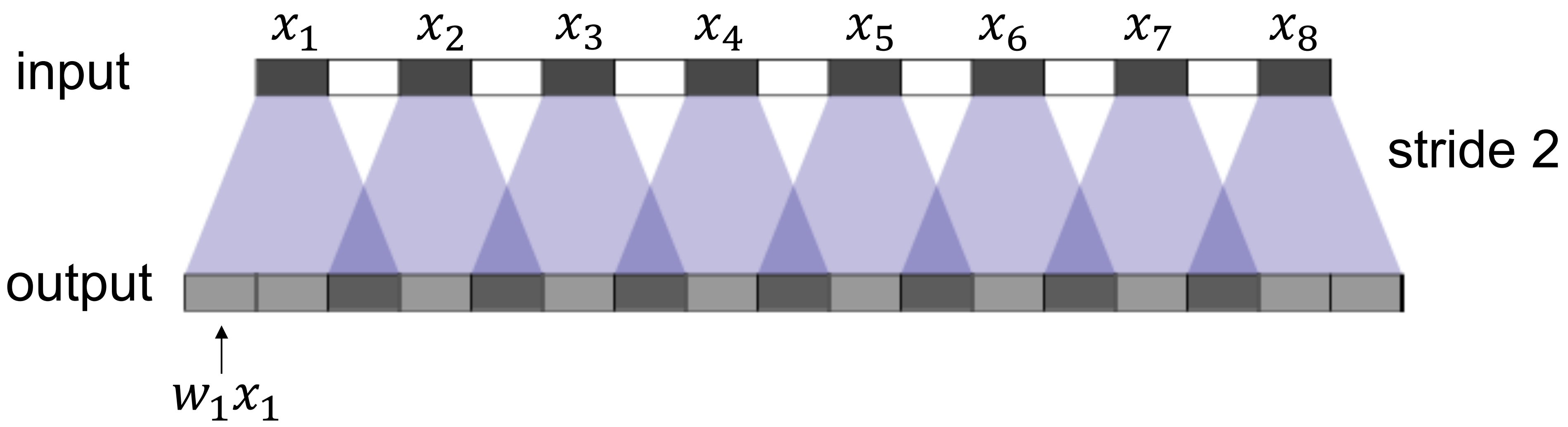
- *Backwards-strided convolution*: to increase resolution, use *output stride* > 1



Animation: https://distill.pub/2016/deconv-checkerboard/

Upsampling by transposed convolution

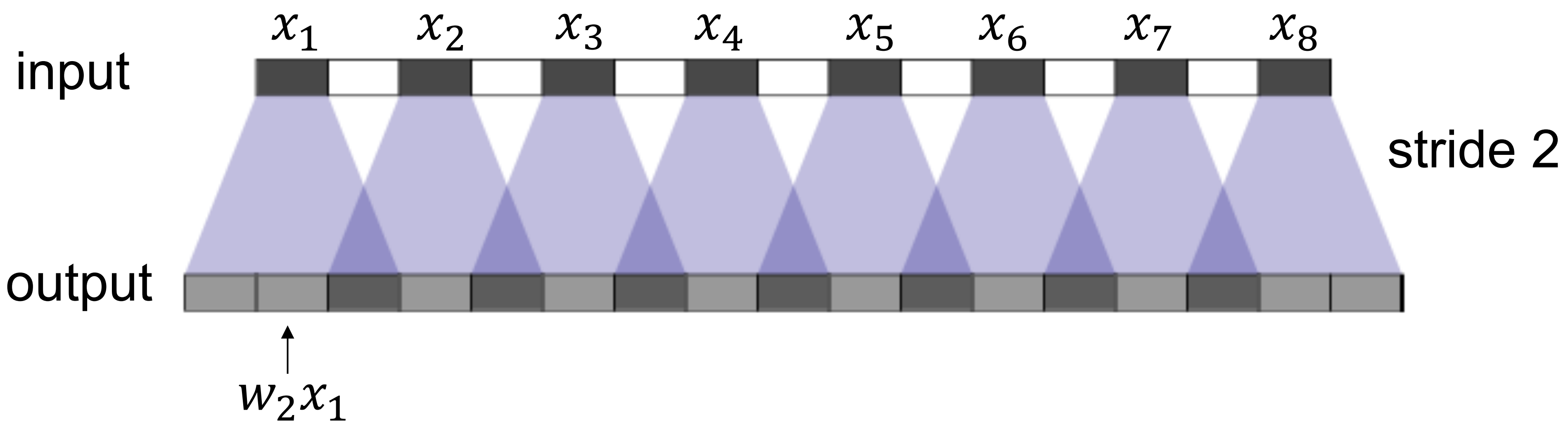
- *Backwards-strided convolution*: to increase resolution, use *output stride* > 1



Animation: https://distill.pub/2016/deconv-checkerboard/

Upsampling by transposed convolution

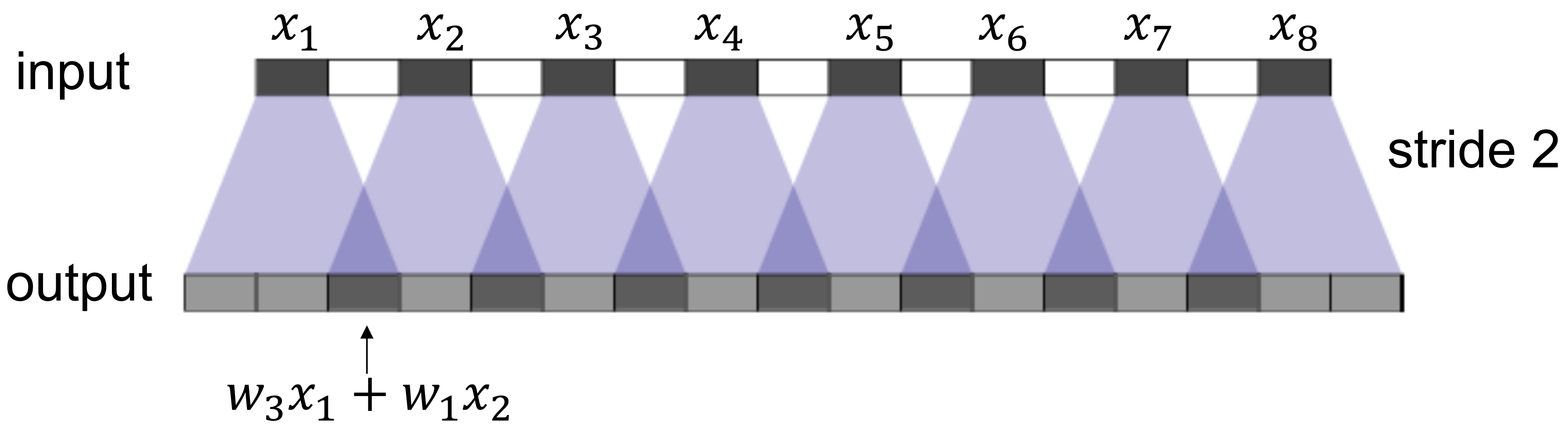
- *Backwards-strided convolution*: to increase resolution, use *output stride* > 1



Animation: https://distill.pub/2016/deconv-checkerboard/

Upsampling by transposed convolution

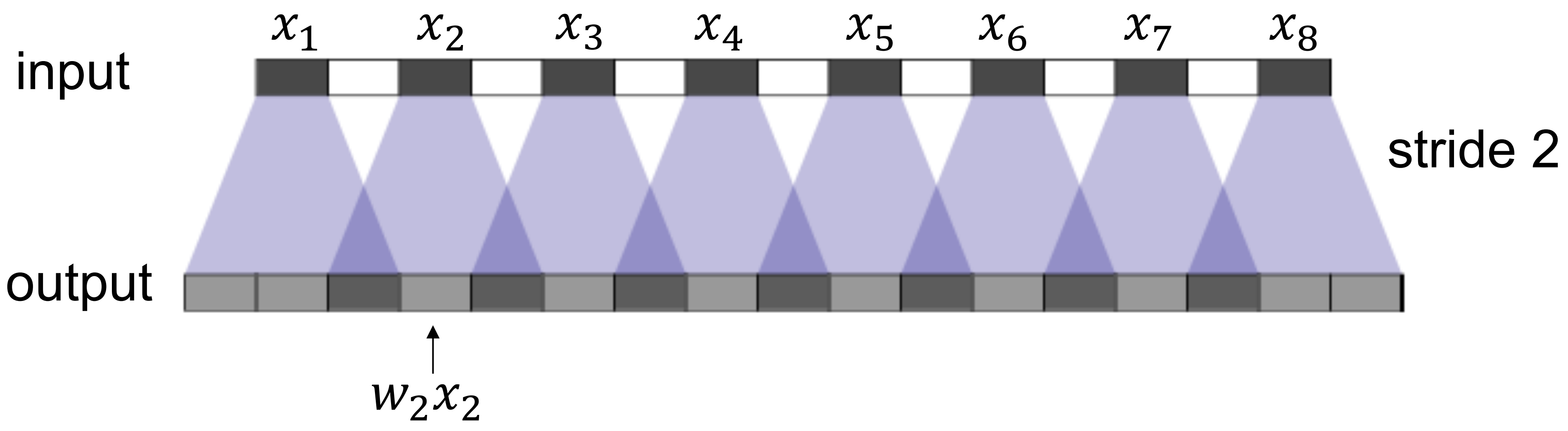
- *Backwards-strided convolution*: to increase resolution, use *output stride* > 1



Animation: https://distill.pub/2016/deconv-checkerboard/

Upsampling by transposed convolution

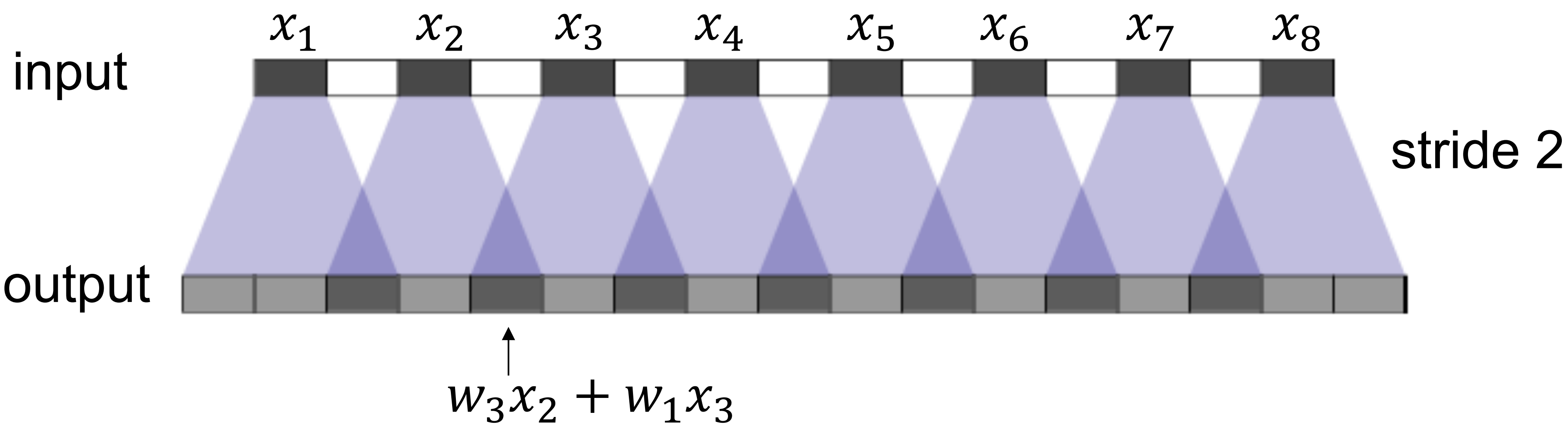
- *Backwards-strided convolution*: to increase resolution, use *output stride* > 1



Animation: https://distill.pub/2016/deconv-checkerboard/

Upsampling by transposed convolution

- *Backwards-strided convolution*: to increase resolution, use *output stride* > 1

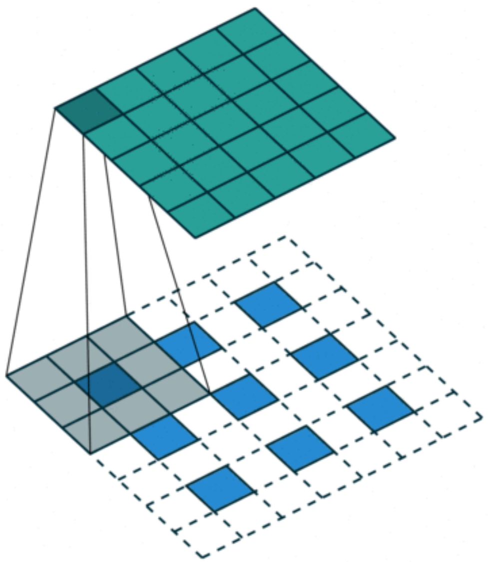


Animation: https://distill.pub/2016/deconv-checkerboard/

Upsampling by transposed convolution

- *Backwards-strided convolution*: to increase resolution, use *output stride* > 1
  - For stride 2, dilate the input by inserting rows and columns of zeros between adjacent entries, convolve with flipped filter
  - Sometimes called convolution with *fractional input stride* 1/2

output



input

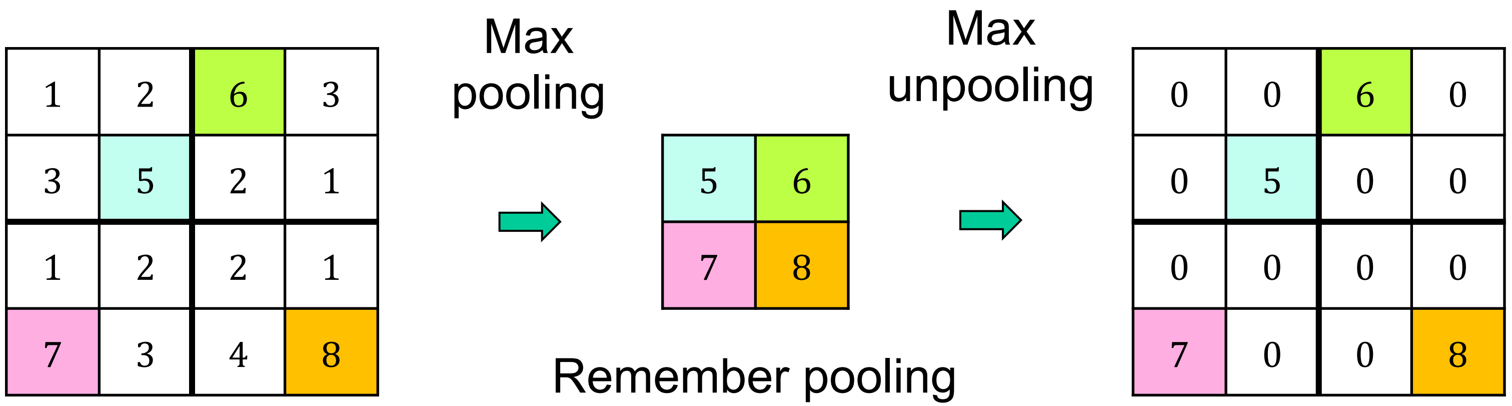
Q: What 3x3 filter would correspond to bilinear upsampling?

$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$
$\frac{1}{2}$	1	$\frac{1}{2}$
$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$

V. Dumoulin and F. Visin, [A guide to convolution arithmetic for deep learning](), arXiv 2018

Upsampling by unpooling

- Alternative to transposed convolution: max unpooling



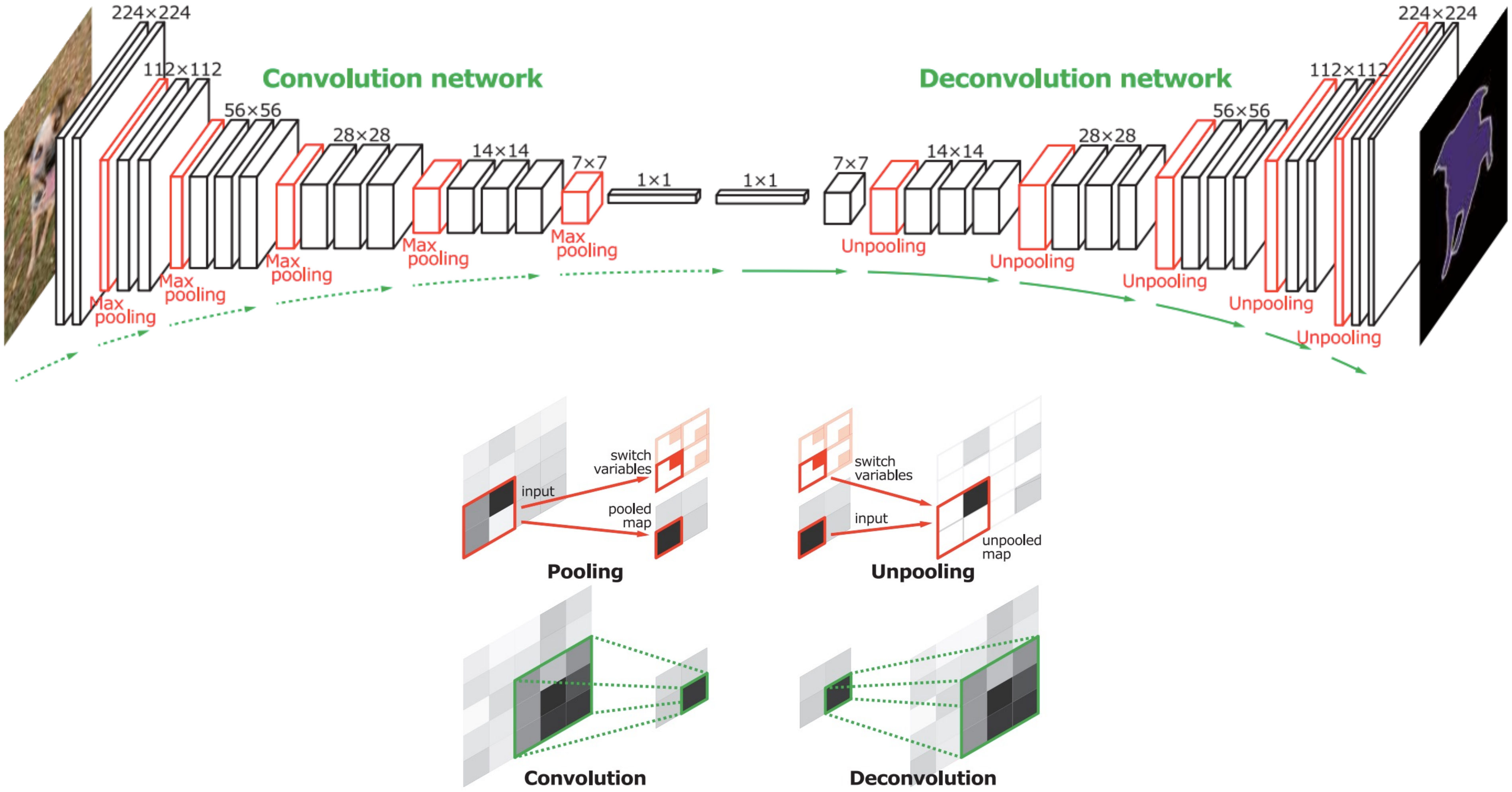
Remember pooling indices (which element was max)

Output is sparse, so unpooling is typically followed by a transposed convolution layer

Dense prediction: Outline

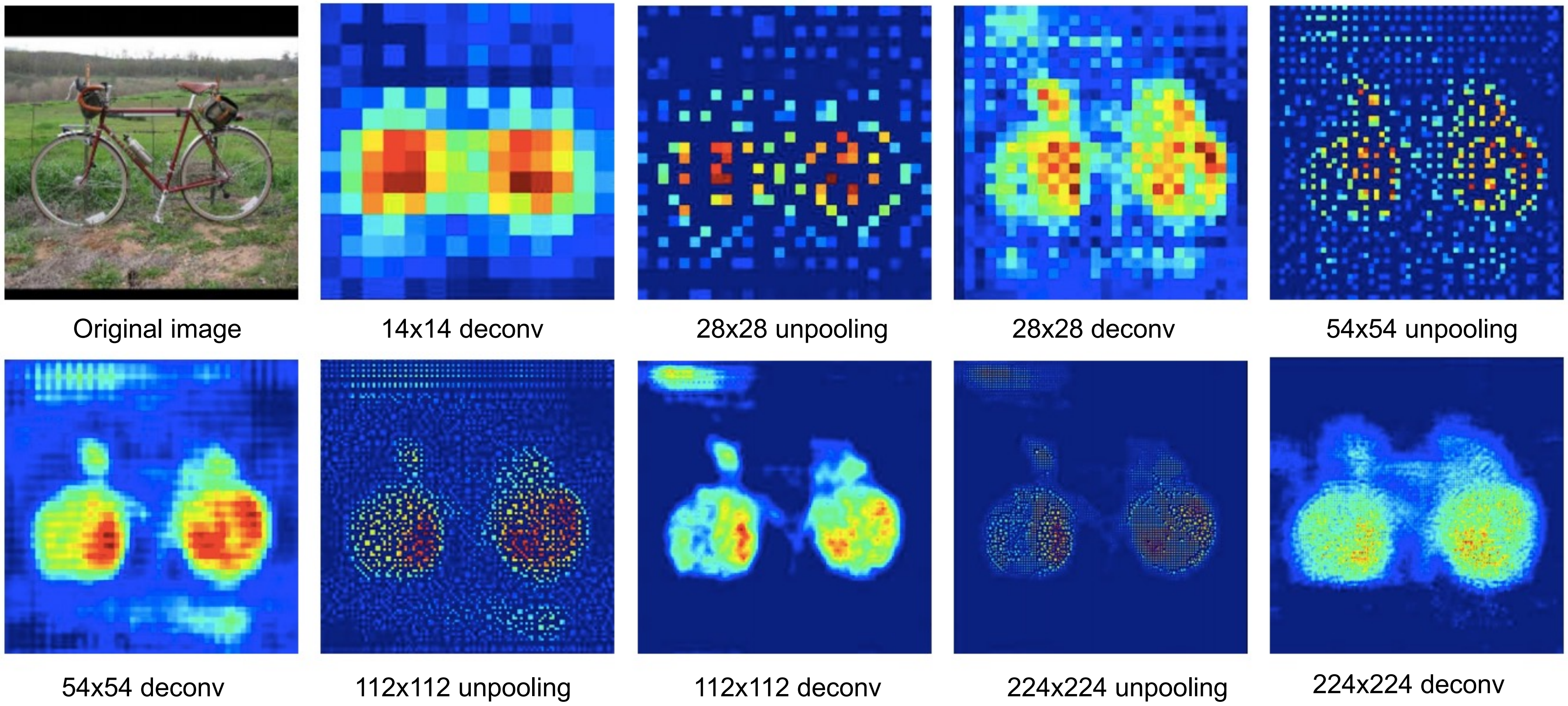
- Fully convolutional networks
- Operations for dense prediction
    - Transposed convolutions, unpooling
- Architectures for dense prediction
    - DeconvNet, SegNet, U-Net
- Instance segmentation
    - Mask R-CNN
- Other dense prediction problems

DeconvNet



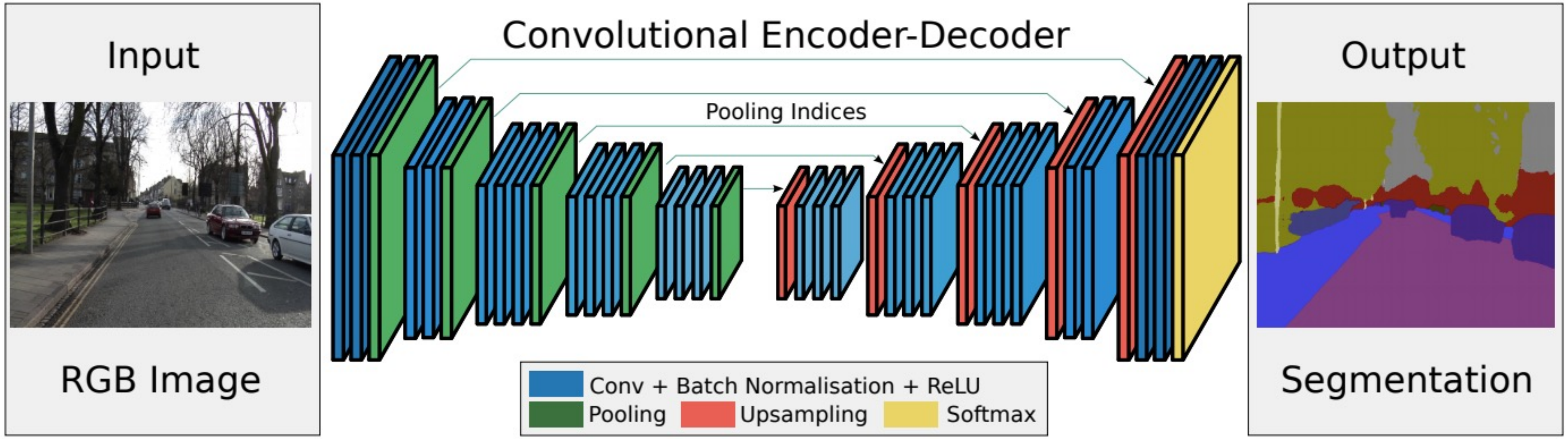
H. Noh, S. Hong, and B. Han, [Learning Deconvolution Network for Semantic Segmentation](), ICCV 2015

DeconvNet



H. Noh, S. Hong, and B. Han, [Learning Deconvolution Network for Semantic Segmentation](), ICCV 2015

Similar architecture: SegNet

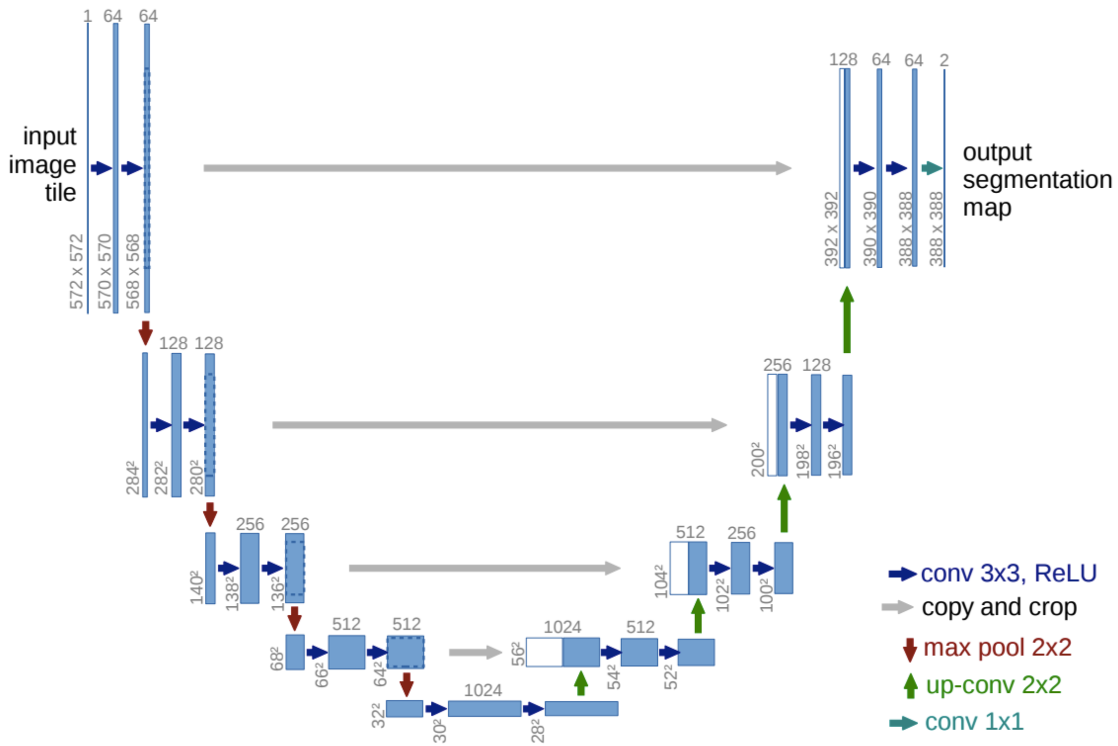


Drop the FC layers,
get better results

V. Badrinarayanan, A. Kendall and R. Cipolla, [SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation](), PAMI 2017

U-Net

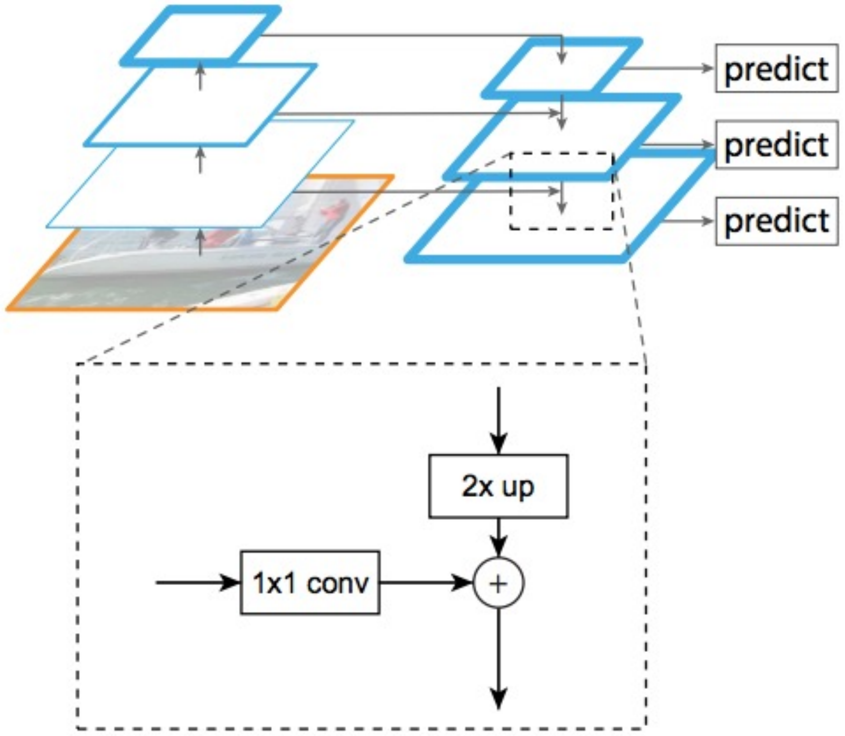
- Like FCN, fuse upsampled higher-level feature maps with higher-res, lower-level feature maps
- Unlike FCN, fuse by concatenation, predict at the end



O. Ronneberger, P. Fischer, T. Brox, [U-Net: Convolutional Networks for Biomedical Image Segmentation](), MICCAI 2015

Recall: 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)



T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, [Feature pyramid networks for object detection](), CVPR 2017

Summary of dense prediction architectures

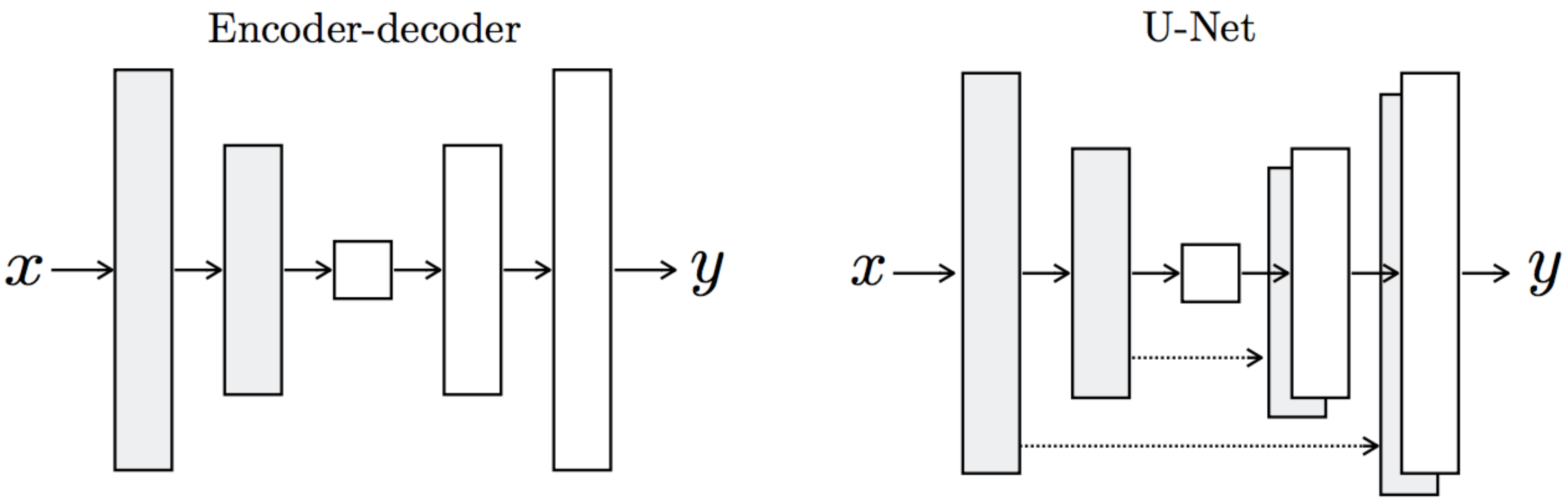
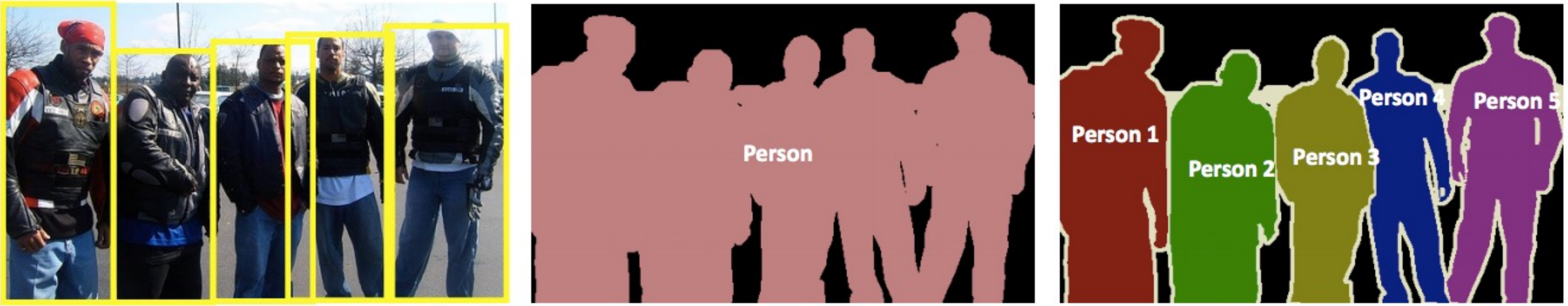


Figure source

Outline

- Fully convolutional networks
- Operations for dense prediction
  - Transposed convolutions, unpooling
- Architectures for dense prediction
  - DeconvNet, SegNet, U-Net
- Instance segmentation
  - Mask R-CNN

Instance segmentation



Object Detection ✓

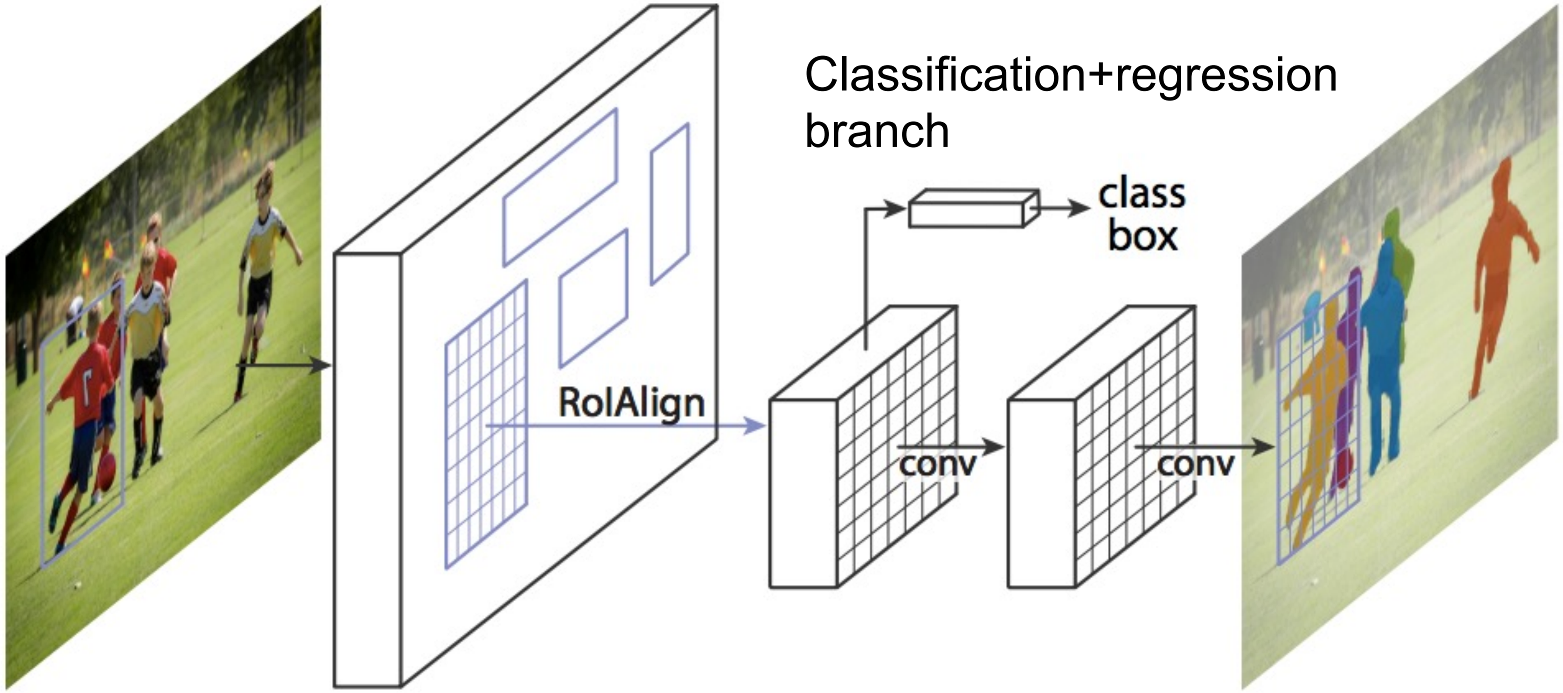
Semantic Segmentation ✓

Instance Segmentation ?

Source: [Kaiming He]

Mask R-CNN

- Mask R-CNN = Faster R-CNN + FCN on RoIs



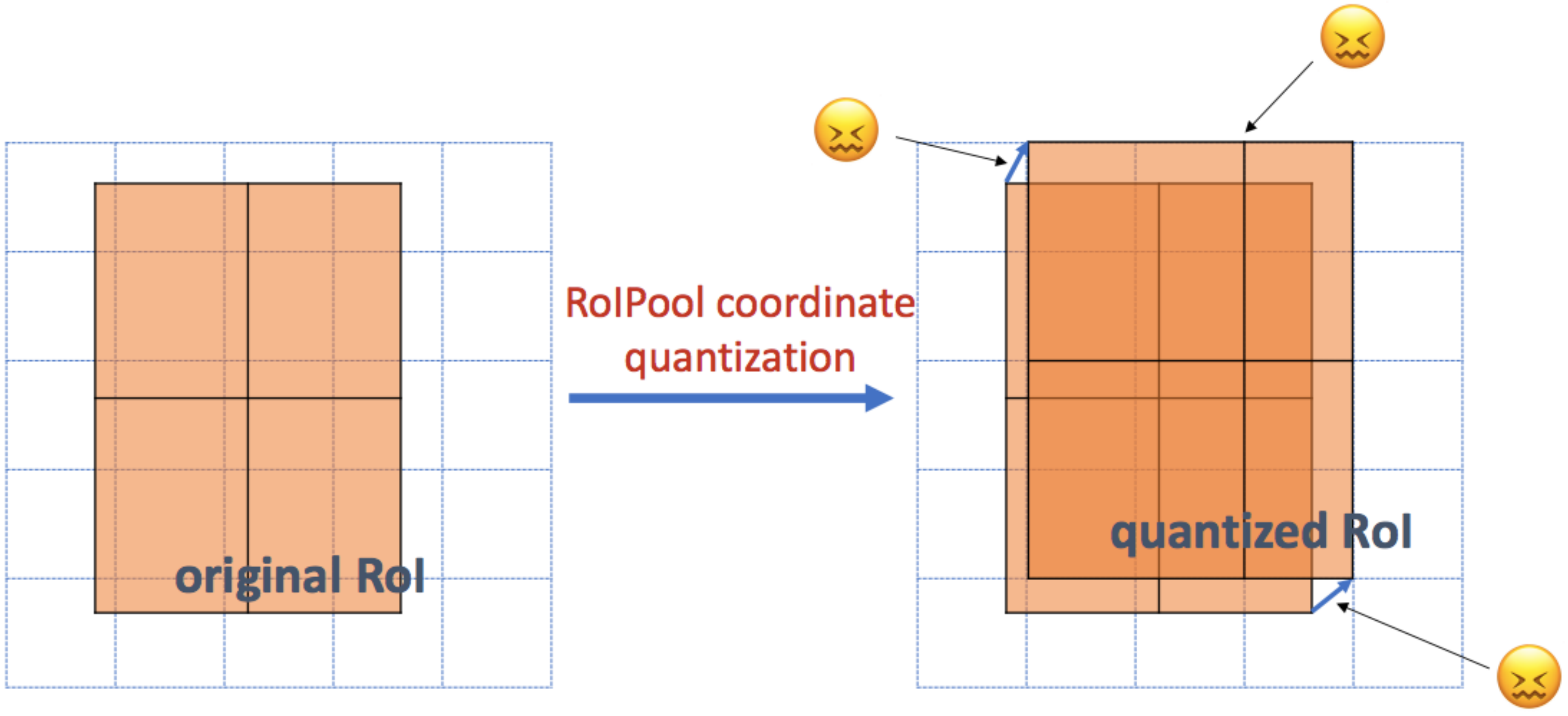
Classification+regression branch

Mask branch: separately predict segmentation for each possible class

K. He, G. Gkioxari, P. Dollar, and R. Girshick, [Mask R-CNN](), ICCV 2017 (Best Paper Award)

RoIAlign vs. RoIPool

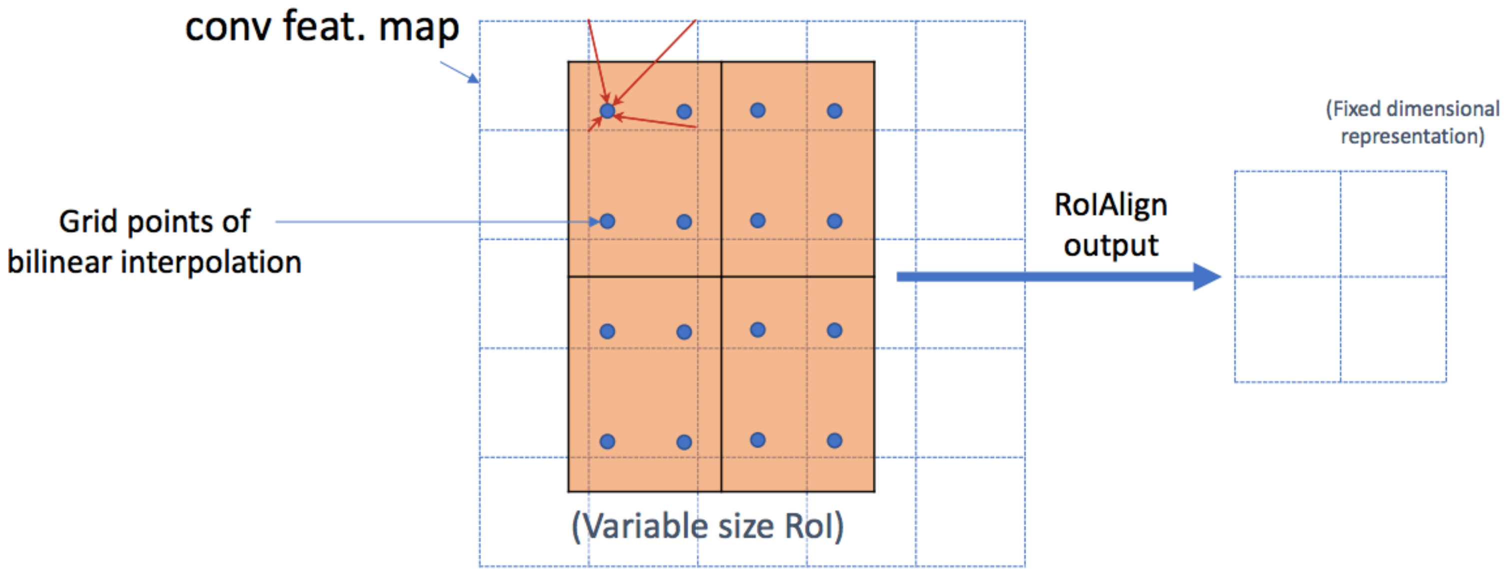
- RoIPool: nearest neighbor quantization



K. He, G. Gkioxari, P. Dollar, and R. Girshick, [Mask R-CNN](), ICCV 2017 (Best Paper Award)

RoIAlign vs. RoIPool

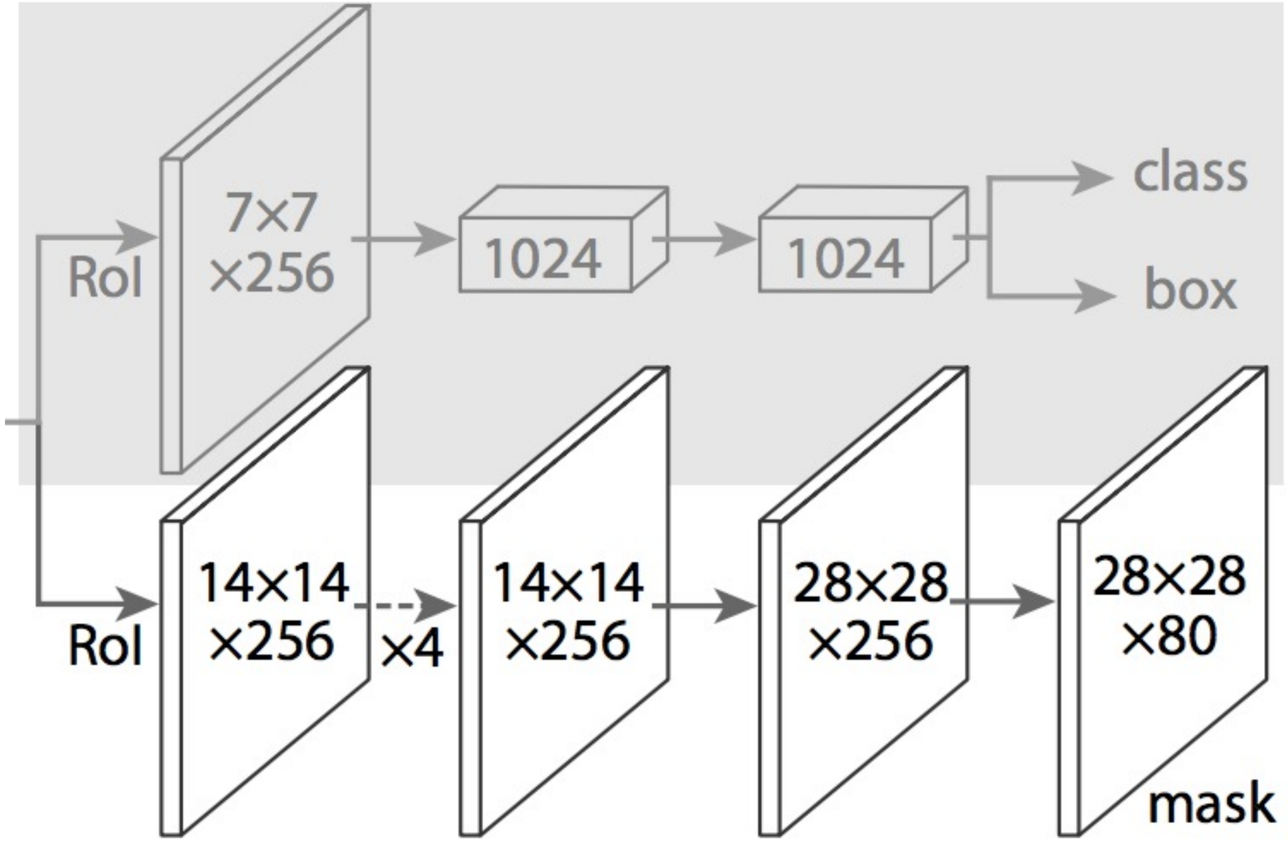
- RoIPool: nearest neighbor quantization
- RoIAlign: bilinear interpolation



K. He, G. Gkioxari, P. Dollar, and R. Girshick, [Mask R-CNN](), ICCV 2017 (Best Paper Award)

Mask R-CNN

- From RoIAlign features, predict class label, bounding box, and segmentation mask



Classification/regression head from an established object detector (e.g., FPN)

Separately predict binary mask for each class with per-pixel sigmoids, use average binary cross-entropy loss

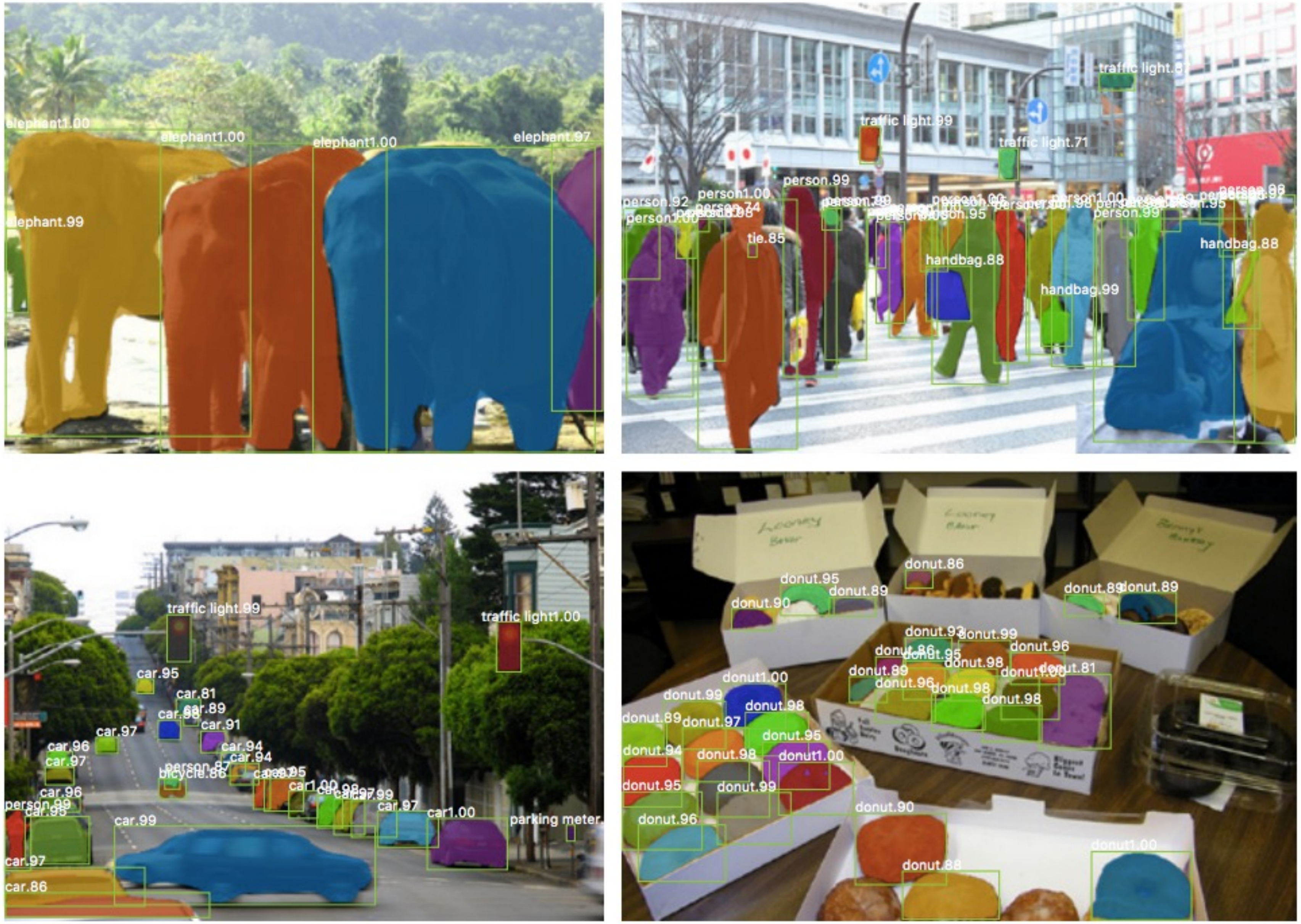
K. He, G. Gkioxari, P. Dollar, and R. Girshick, [Mask R-CNN](), ICCV 2017 (Best Paper Award)

Mask R-CNN

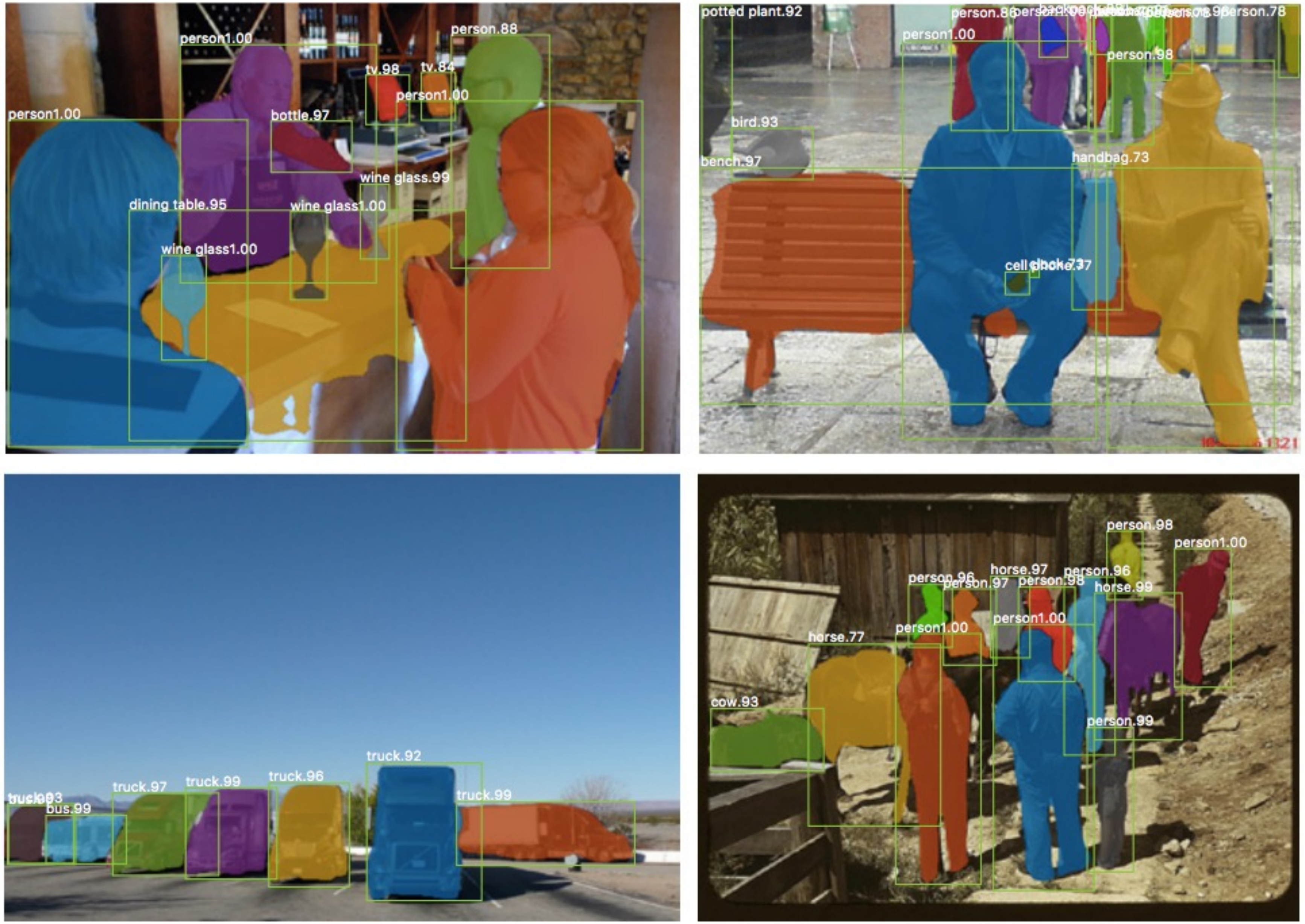
Validation image with box detection shown in red

K. He, G. Gkioxari, P. Dollar, and R. Girshick, [Mask R-CNN](), ICCV 2017 (Best Paper Award)

Example results



Example results



Instance segmentation results on COCO

	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**

AP at different IoU thresholds

AP for different size instances

K. He, G. Gkioxari, P. Dollar, and R. Girshick, Mask R-CNN, ICCV 2017 (Best Paper Award)

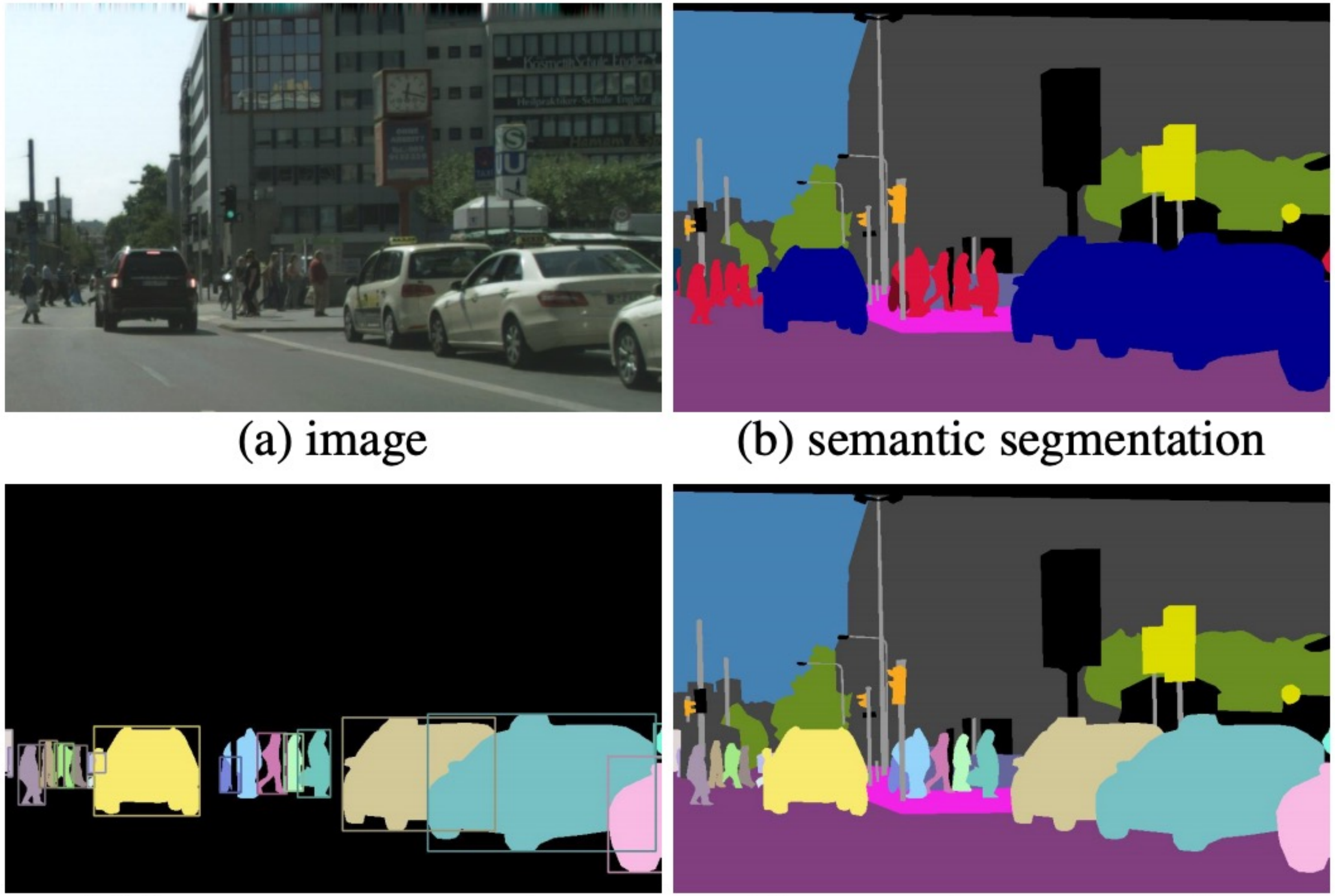
Keypoint prediction

- Given K keypoints, train model to predict K $m \times m$ *one-hot* maps with cross-entropy losses over m^2 outputs

Outline

- Fully convolutional networks
- Operations for dense prediction
  - Transposed convolutions, unpooling
- Architectures for dense prediction
  - DeconvNet, SegNet, U-Net
- Instance segmentation
  - Mask R-CNN
- Other dense prediction problems

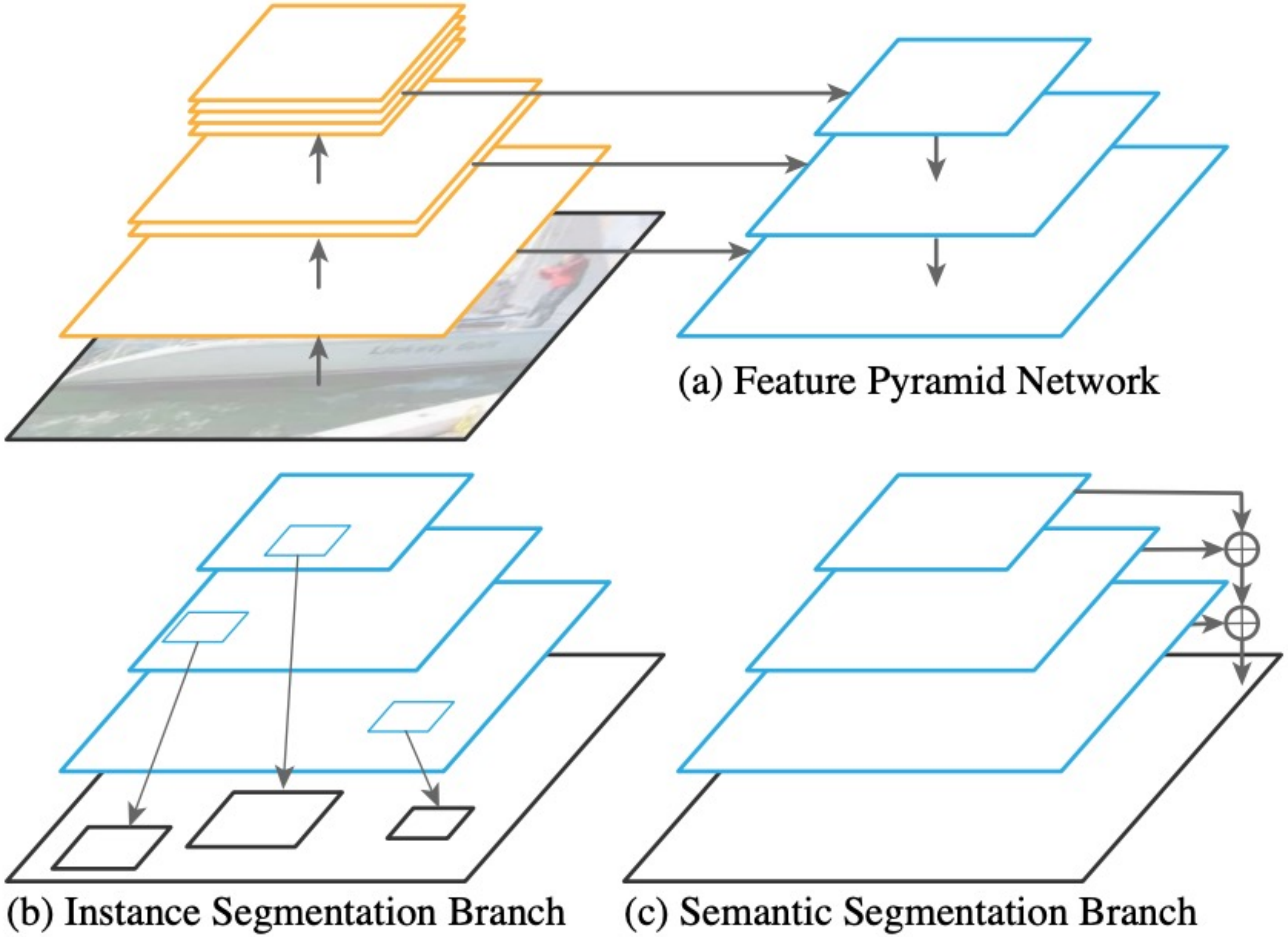
Recently proposed task: Panoptic segmentation



(a) image
(b) semantic segmentation
(c) instance segmentation
(d) panoptic segmentation

A. Kirillov et al. [Panoptic segmentation](). CVPR 2019

Panoptic feature pyramid networks



A. Kirillov et al. [Panoptic feature pyramid networks](). CVPR 2019

Panoptic feature pyramid networks

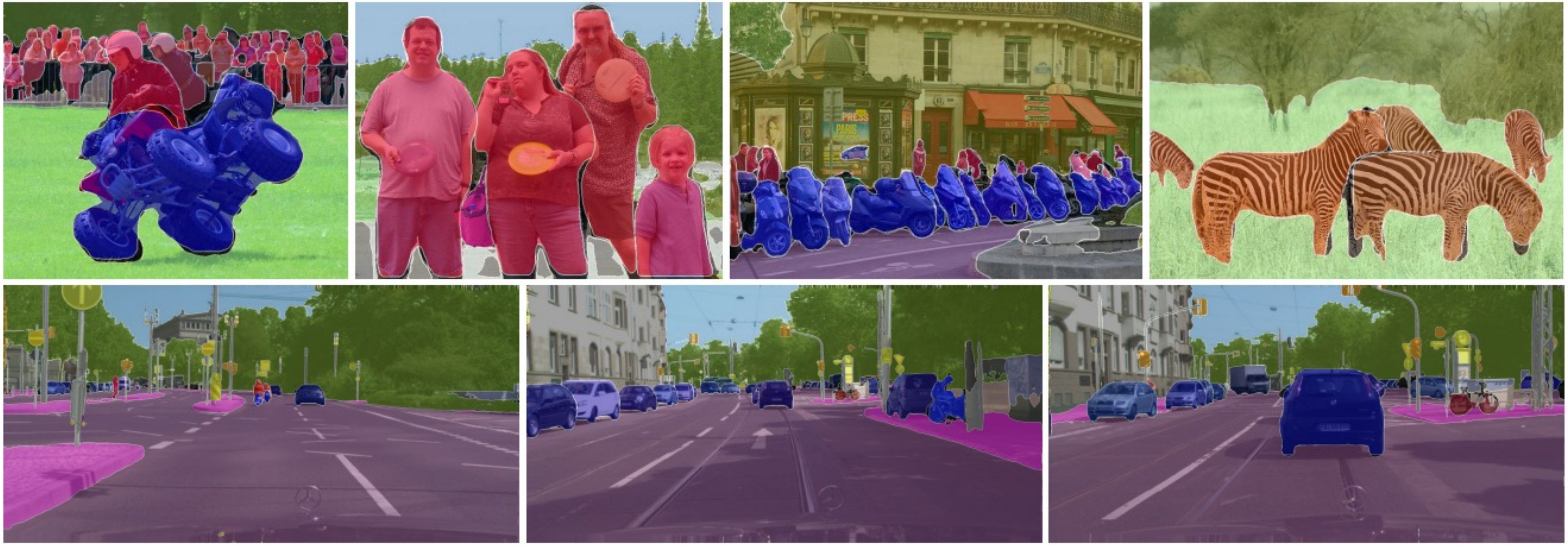
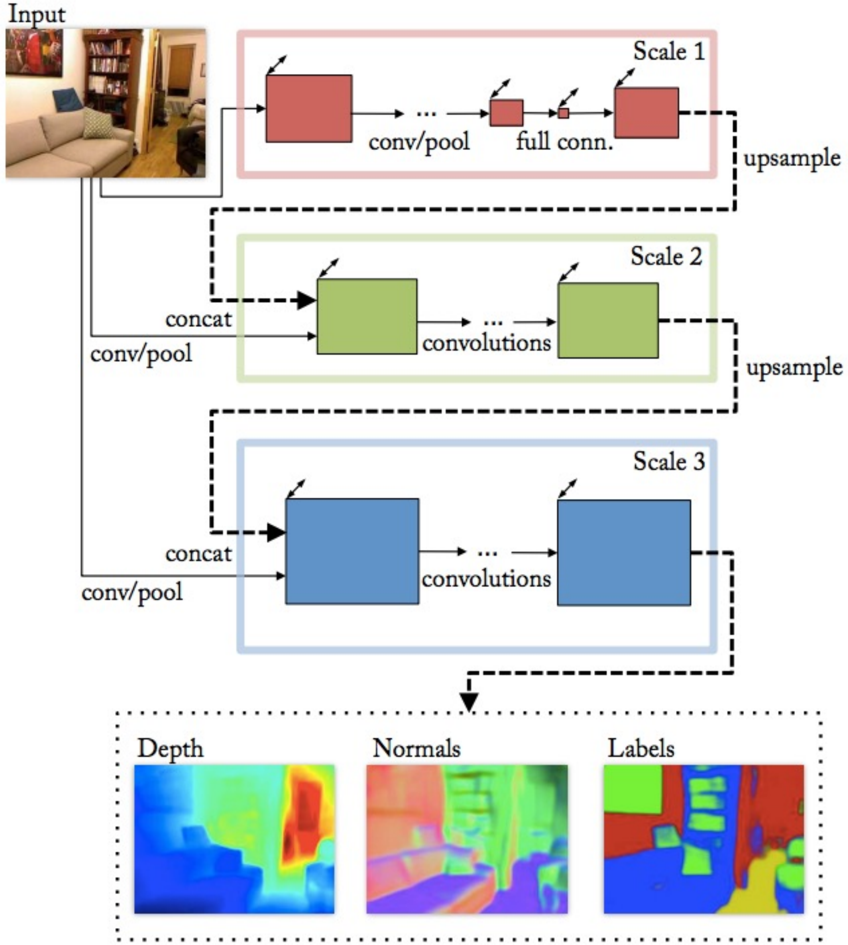


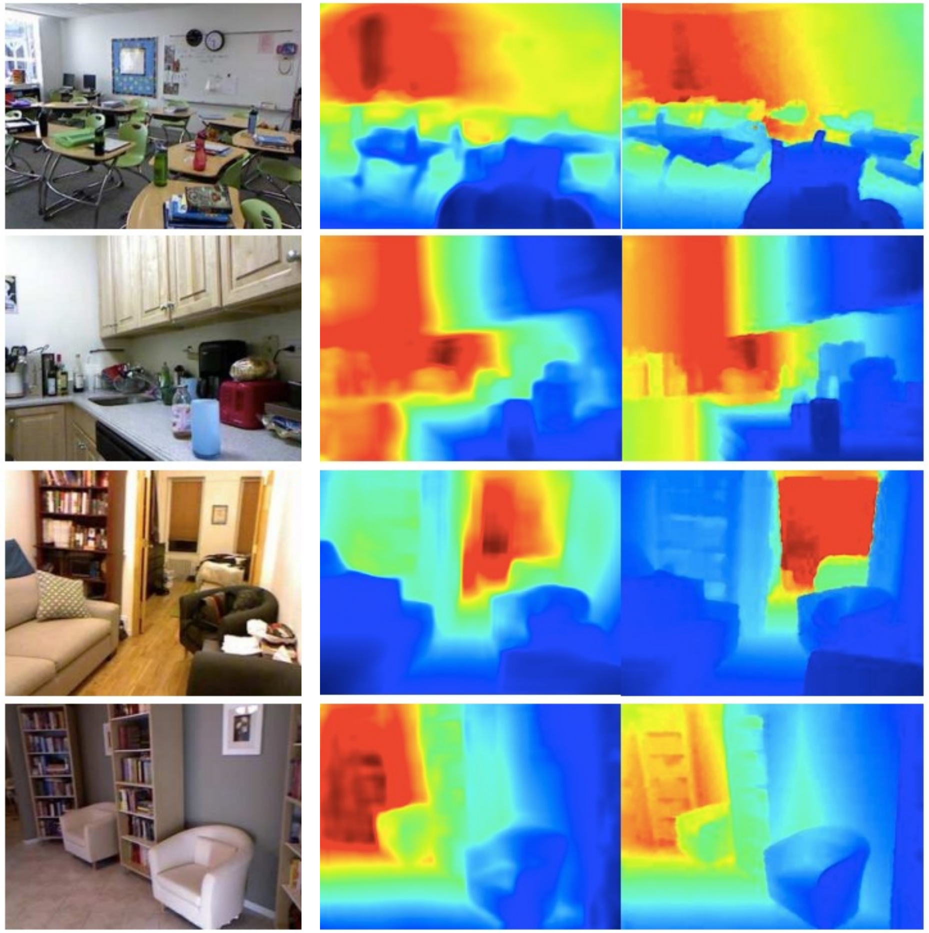
Figure 2: Panoptic FPN results on COCO (top) and Cityscapes (bottom) using a single ResNet-101-FPN network.

A. Kirillov et al. [Panoptic feature pyramid networks](). CVPR 2019

Depth and normal estimation

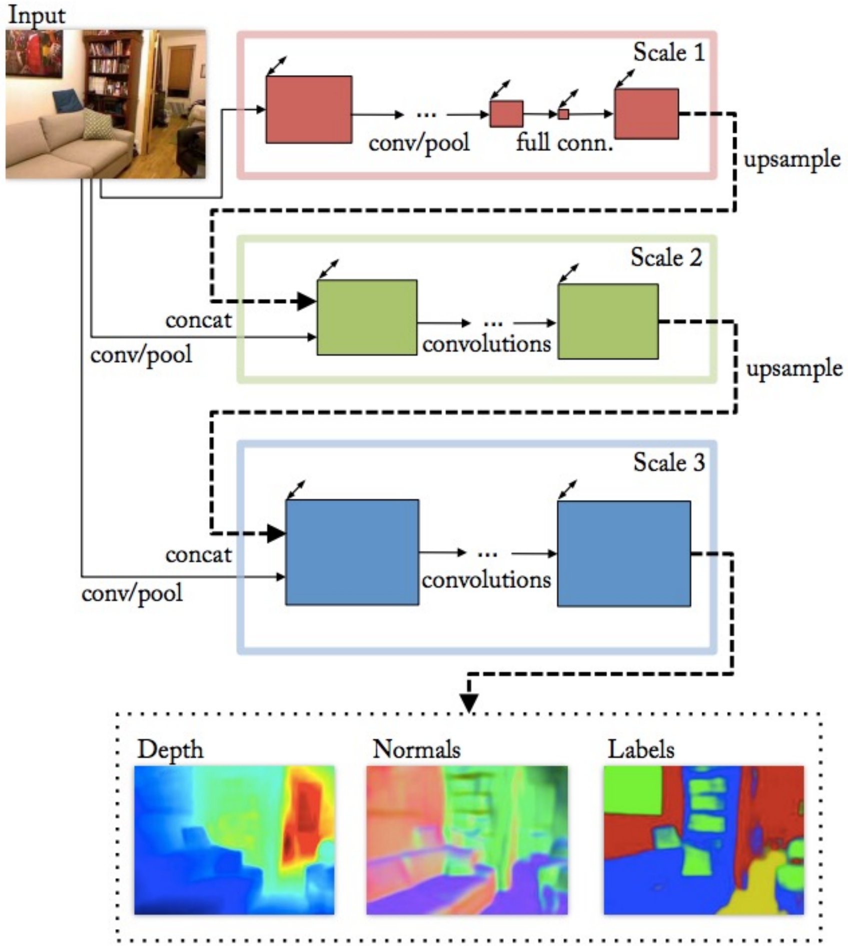


Predicted depth Ground truth

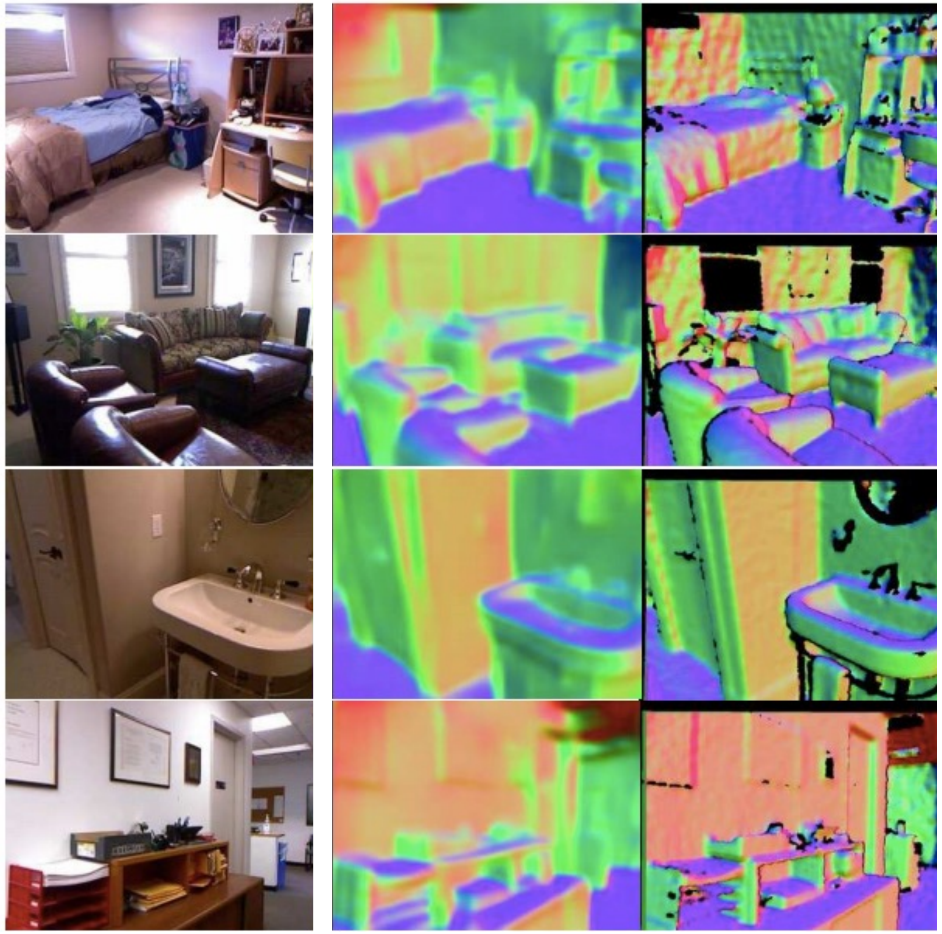


D. Eigen and R. Fergus, [Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture](), ICCV 2015

Depth and normal estimation

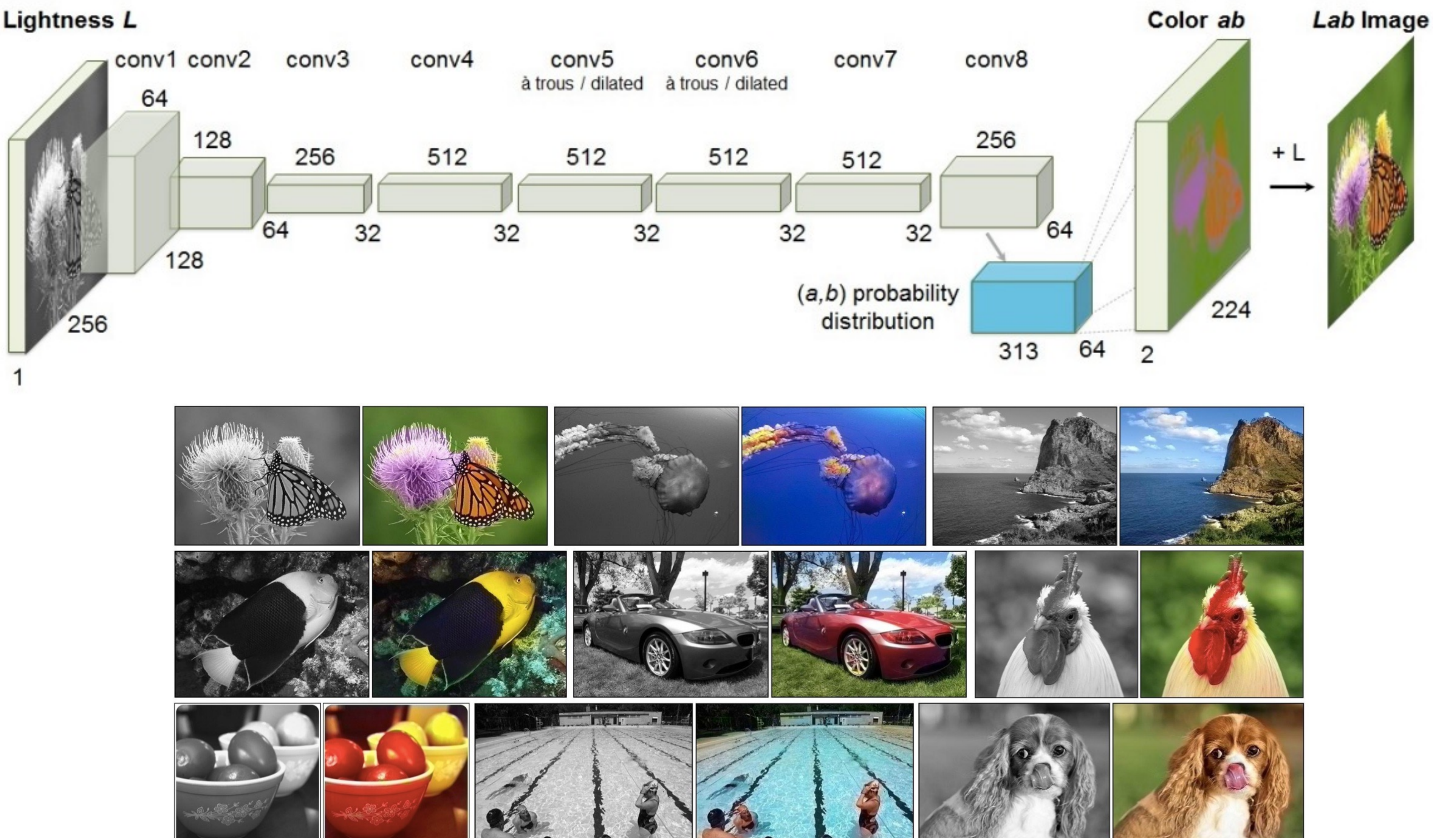


Predicted normals Ground truth



D. Eigen and R. Fergus, [Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture](), ICCV 2015

Colorization



R. Zhang, P. Isola, and A. Efros, [Colorful Image Colorization](), ECCV 2016