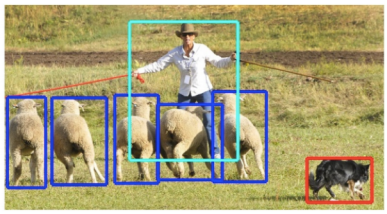
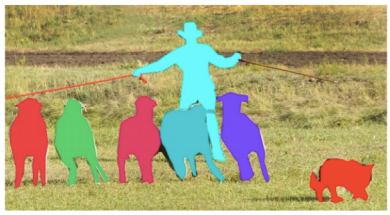
CNNs for dense image labeling

image classification



object detection

semantic segmentation



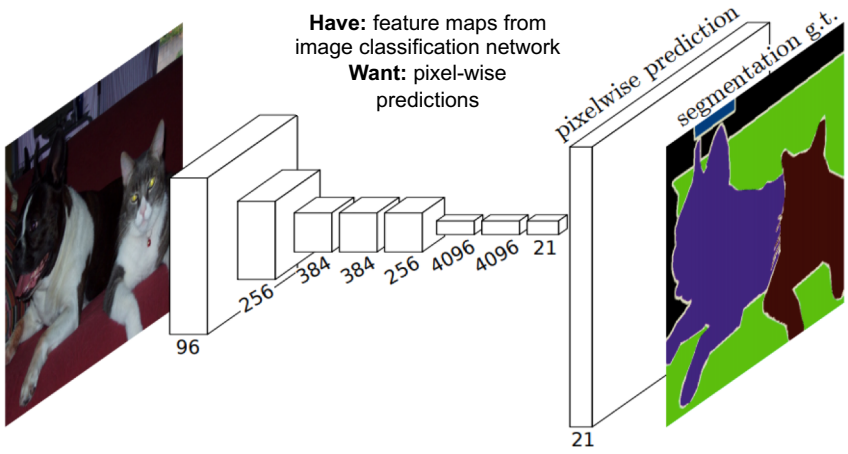
instance segmentation

Outline

- Early “hacks”
  - Hypercolumns
  - Zoom-out features
- Fully convolutional networks
  - Learned upsampling architectures
  - Dilated convolutions
- Instance segmentation
  - Mask R-CNN
- Other dense prediction problems

Early “hacks”

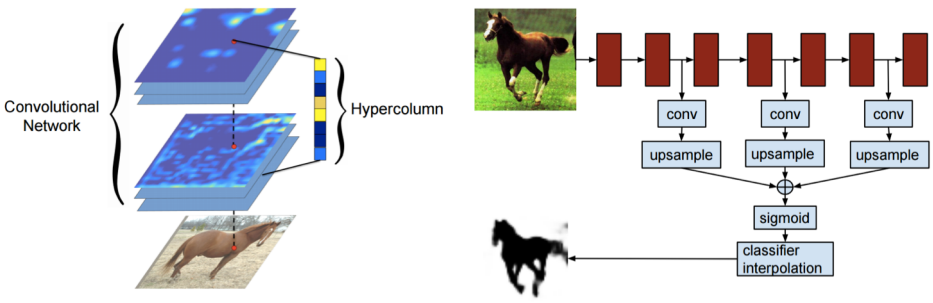
- Do dense prediction as a post-process on top of an image classification CNN



Hypercolumns

- Idea: to obtain a feature representation for an individual pixel, upsample all feature maps to original image resolution and concatenate values from feature maps “above” that pixel

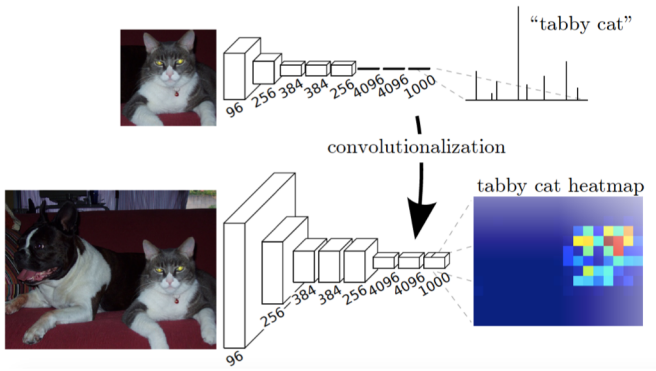
Representation as an end-to-end network:



B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik, Hypercolumns for Object Segmentation and Fine-grained Localization, CVPR 2015

Fully convolutional networks

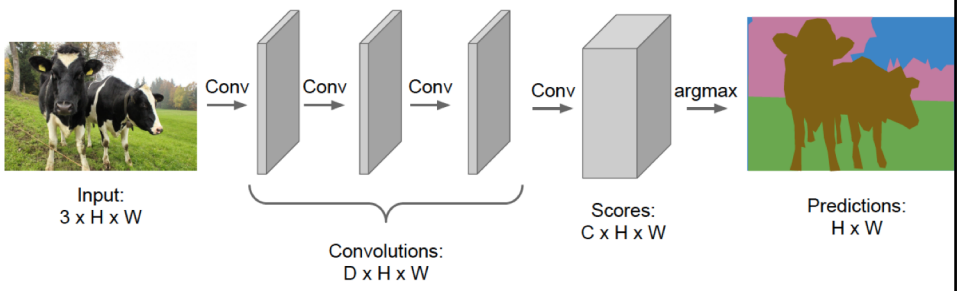
- Design a network with only convolutional layers, make predictions for all pixels at once



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

Fully convolutional networks

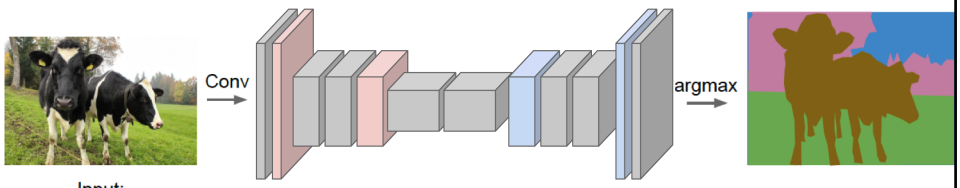
- Design a network with only convolutional layers, make predictions for all pixels at once
- Ideally, we want convolutions at full image resolution, but implementing that naively is too expensive



Source: Stanford CS231n

Fully convolutional networks

- Design a network with only convolutional layers, make predictions for all pixels at once
- Ideally, we want convolutions at full image resolution, but implementing that naively is too expensive
  - Solution: first downsample, then upsample



Input:
3 x H x W

Predictions:
H x W

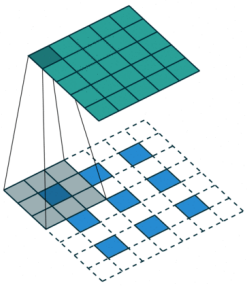
Source: Stanford CS231n

Upsampling in a deep network

- *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

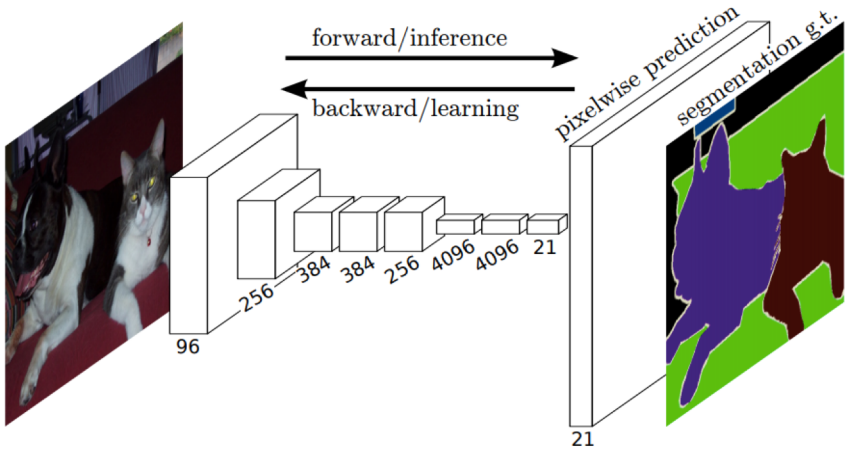


Q: What 3x3 filter would correspond to bilinear upsampling?

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

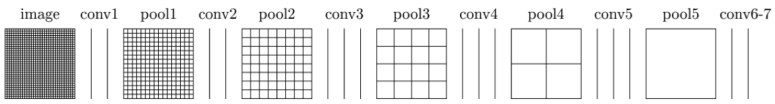
V. Dumoulin and F. Visin, A guide to convolution arithmetic for deep learning, arXiv 2018

Fully convolutional networks (FCN)

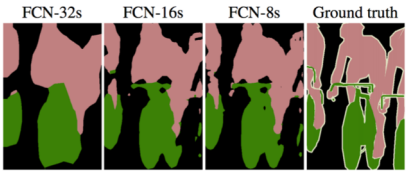


J. Long, E. Shelhamer, and T. Darrell, Fully Convolutional Networks for Semantic Segmentation, CVPR 2015

Fully convolutional networks (FCN)



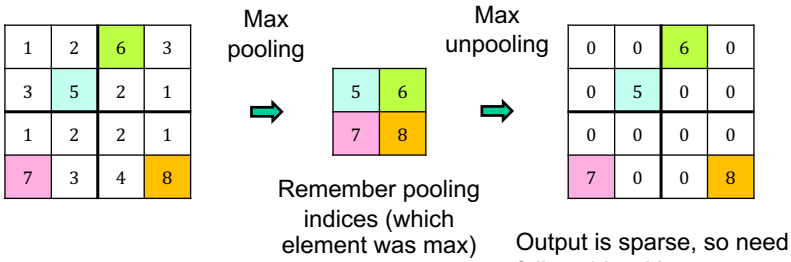
- Predictions by 1x1 conv layers, bilinear upsampling
- Predictions by 1x1 conv layers, learned 2x upsampling, fusion by summing



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

Upsampling in a deep network

- Alternative to transposed convolution: max unpooling

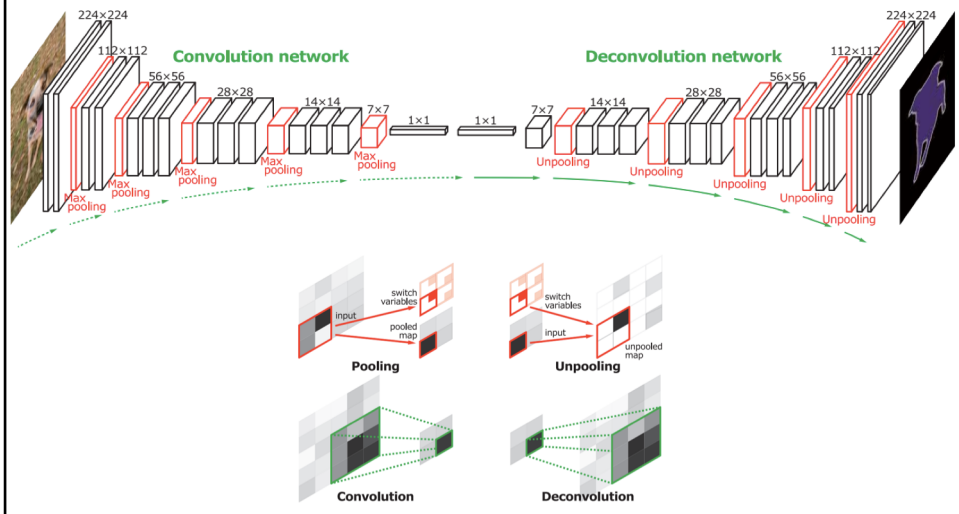


Remember pooling indices (which element was max)

Output is sparse, so need to follow this with a transposed convolution layer

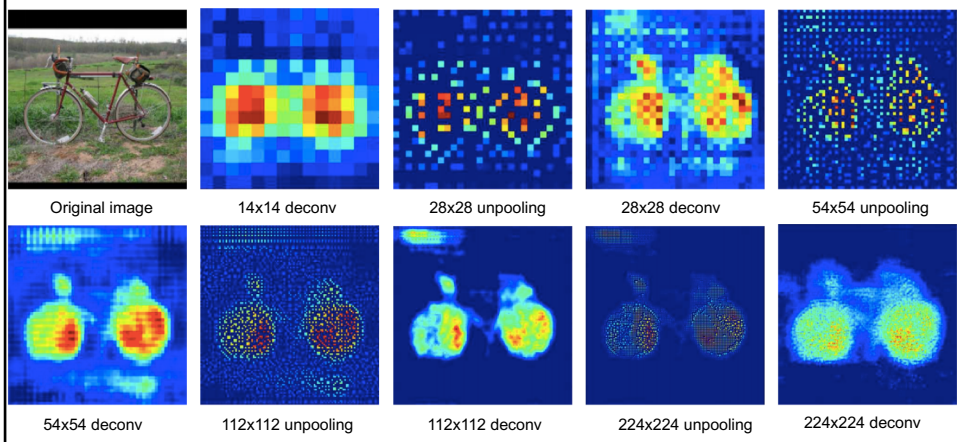
(sometimes called deconvolution instead of transposed convolution, but this is not accurate)

DeconvNet



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

DeconvNet

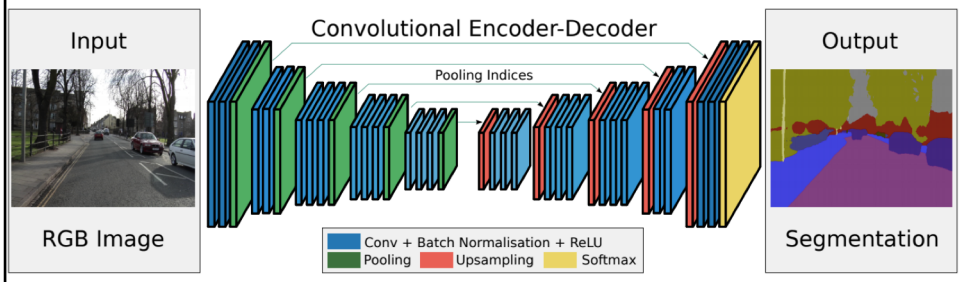


H. Noh, S. Hong, and B. Han, Learning Deconvolution Network for Semantic Segmentation, ICCV 2015

DeconvNet results

PASCAL VOC 2012	mIoU
Hypercolumns	59.2
ZoomOut	64.4
FCN-8	62.2
DeconvNet	69.6
Ensemble of DeconvNet and FCN	71.7

Similar architecture: SegNet

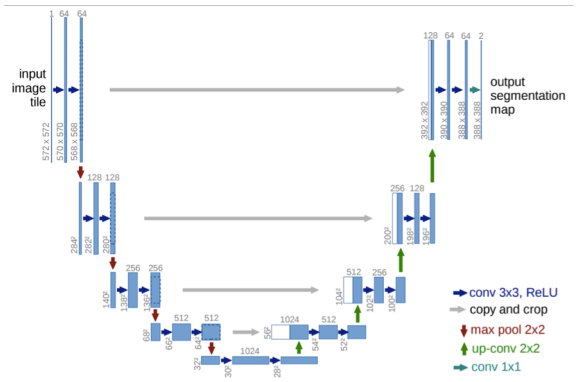


Drop the FC layers,
get better results

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

U-Net

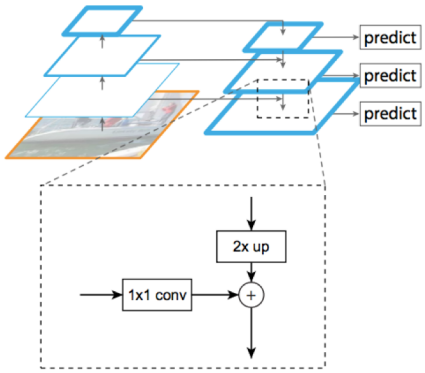
- Like FCN, fuse upsampled higher-level feature maps with higher-res, lower-level feature maps
- Unlike FCN, fuse by concatenation, predict at the end



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 upsampling architectures

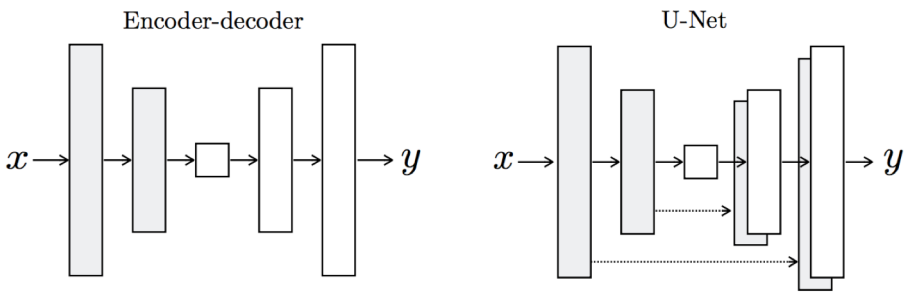


Figure source

Dilated convolutions

- Idea: instead of reducing spatial resolution of feature maps, use a large sparse filter
  - Also known as *à trous* convolution

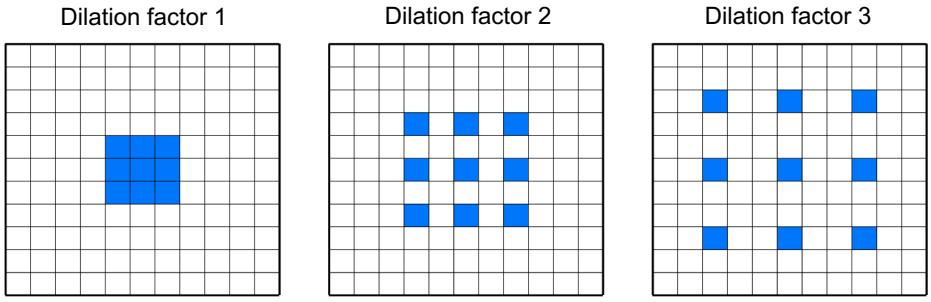
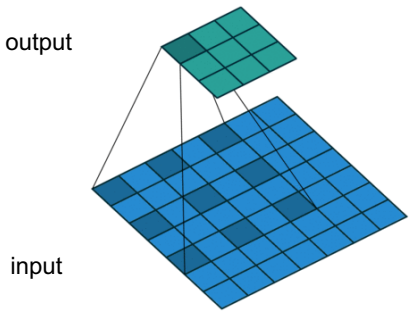


Image source

Dilated convolutions

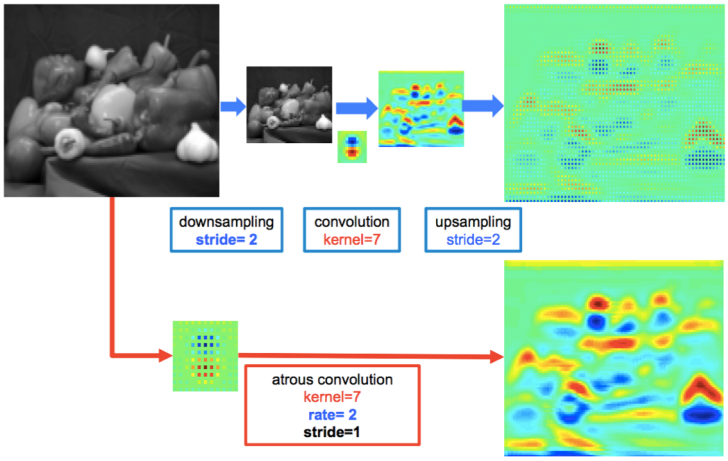
- Idea: instead of reducing spatial resolution of feature maps, use a large sparse filter



Like 2x downsampling followed by 3x3 convolution followed by 2x upsampling

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

Dilated convolutions



L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. Yuille, DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, PAMI 2017

Dilated convolutions

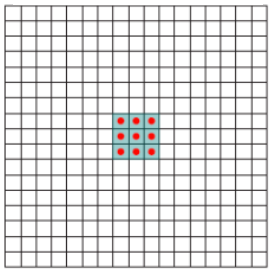
- Use in FCN to remove downsampling:
  change stride of max pooling layer from 2 to 1,
  dilate subsequent convolutions by factor of 2
  (possibly without re-training any parameters)

L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. Yuille, DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, PAMI 2017

Dilated convolutions

- Can increase receptive field size exponentially with a linear growth in the number of parameters

Feature map 1 (F1) produced from F0 by 1-dilated convolution



Receptive field: 3x3

Receptive field: 7x7

Receptive field: 15x15

F. Yu and V. Koltun, Multi-scale context aggregation by dilated convolutions, ICLR 2016

Dilated convolutions: Evaluation

(a) Image (b) FCN-8s (c) DeepLab (d) Our front end

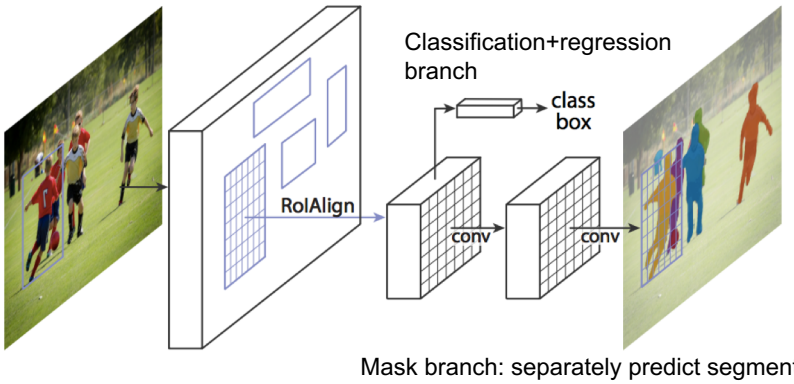
Instance segmentation

Object Detection ✓ | Semantic Segmentation ✓ | **Instance Segmentation** ?

Source: Kaiming He

Mask R-CNN

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

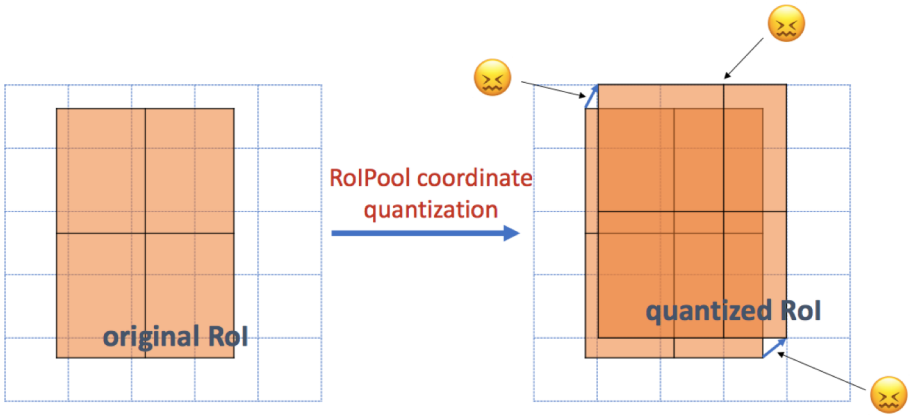


Mask branch: separately predict segmentation for each possible class

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

RoIAlign vs. RoIPool

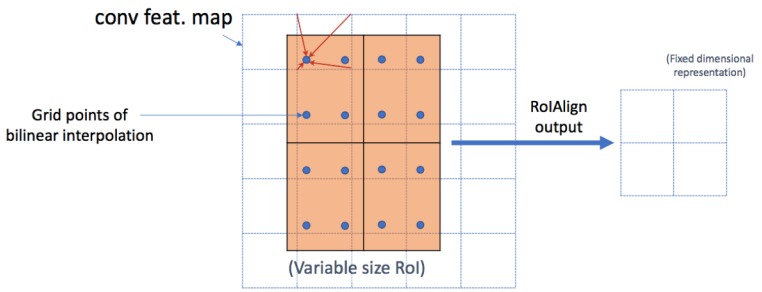
- RoIPool: nearest neighbor quantization



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

RoIAlign vs. RoIPool

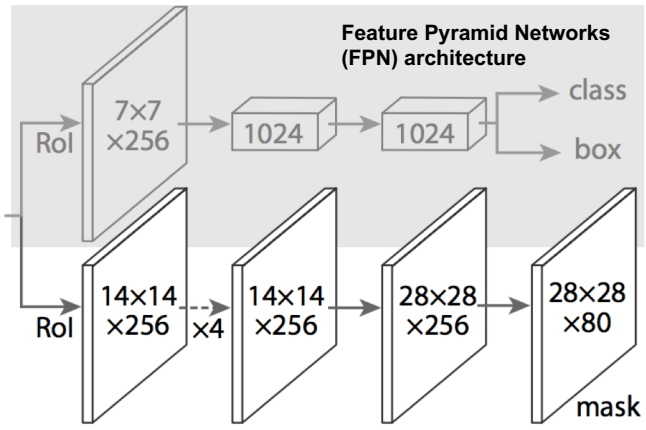
- RoIPool: nearest neighbor quantization
- RoIAlign: bilinear interpolation



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

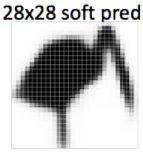


K. He, G. Gkioxari, P. Dollar, and R. Girshick, Mask R-CNN, ICCV 2017 (Best Paper Award)

Mask R-CNN

Validation image with box detection shown in red

28x28 soft prediction



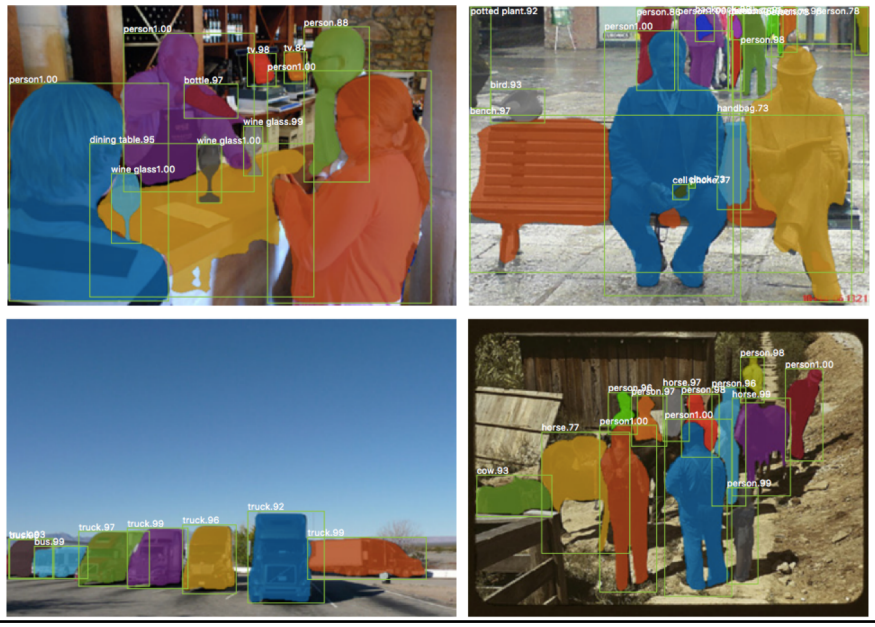
Resized Soft prediction

Final mask

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)

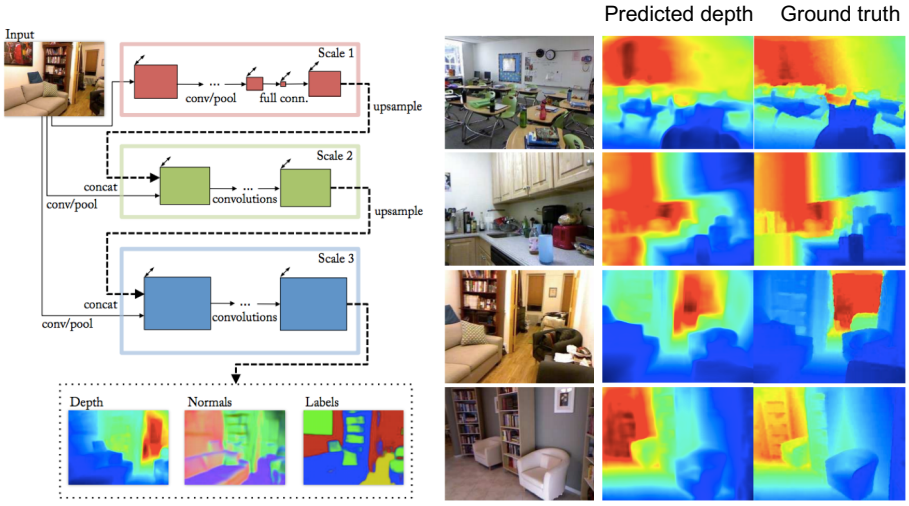
Keypoint prediction

- Given K keypoints, train model to predict K m x m *one-hot* maps

Other dense prediction tasks

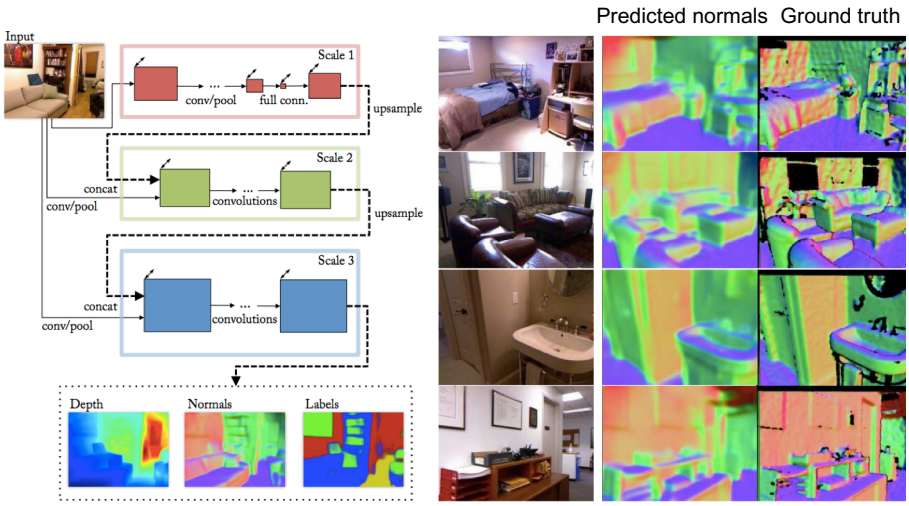
- Depth estimation
- Surface normal estimation
- Colorization
- ….

Depth and normal estimation



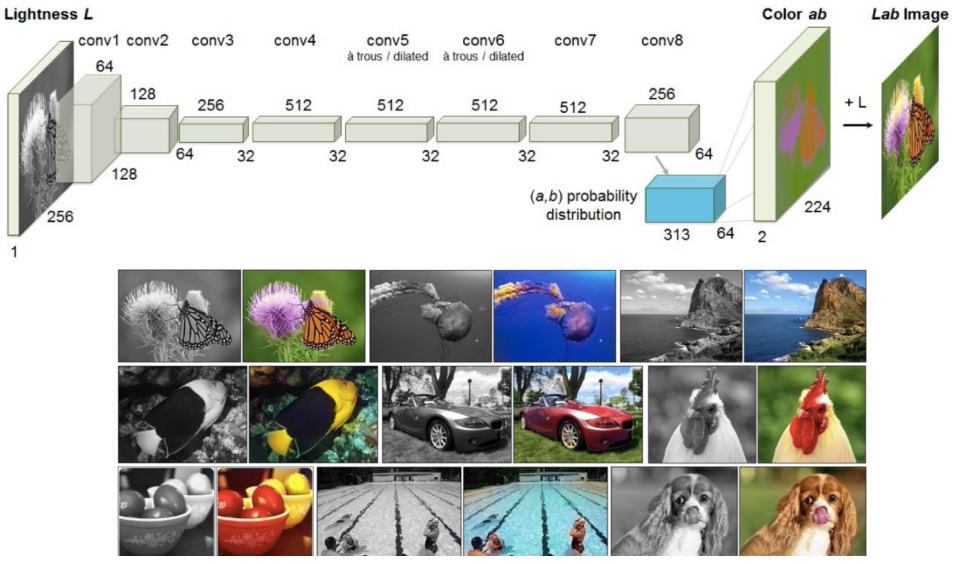
D. Eigen and R. Fergus, Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture, ICCV 2015

Depth and normal estimation



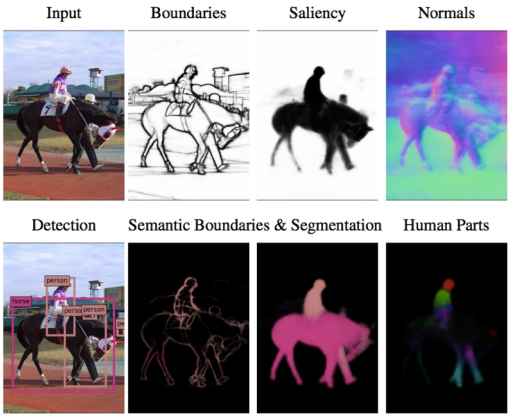
D. Eigen and R. Fergus, Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture, ICCV 2015

Colorization



R. Zhang, P. Isola, and A. Efros, Colorful Image Colorization, ECCV 2016

Estimation of everything at the same time



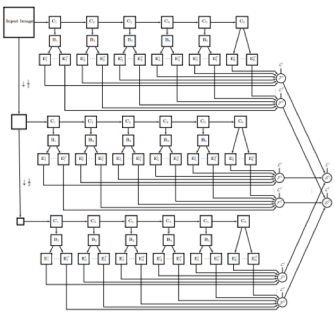


Figure 2: UberNet architecture: an image pyramid is formed by successive down-sampling operations, and each image is processed by a CNN with tied weights; the responses of the network at consecutive layers ($\mathbf{C}_i$) are processed with Batch Normalization ($\mathbf{B}_i$) and then fed to task-specific skip layers ($\mathbf{E}_i^t$); these are combined across network layers ($\mathcal{F}^t$) and resolutions (S^t) and trained using task-specific loss functions ($\mathcal{L}^t$), while the whole architecture is jointly trained end-to-end. For simplicity we omit the interpolation and detection layers mentioned in the text.

I. Kokkinos, [UberNet: Training a Universal Convolutional Neural Network for Low-, Mid-, and High-Level Vision using Diverse Datasets and Limited Memory](), ICCV 2017