Advanced Deep Learning Architectures
COMP 5214 & ELEC 5680

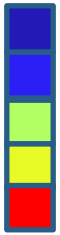
Instructor: Dr. Qifeng Chen

https://cqf.io

Credit: https://d2l.ai/

Semantic Segmentation

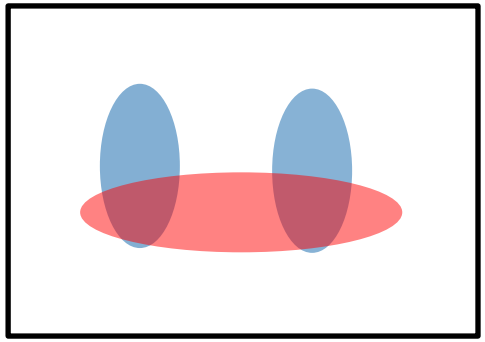
The Task



person
grass
trees
motorbike
road

Evaluation metric

- Pixel classification!
- Accuracy?
- Heavily unbalanced
- Common classes are over-emphasized
- *Intersection over Union*
- Average across classes and images
- Per-class accuracy
- Compute accuracy for every class and then average



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”

Challenges in data collection

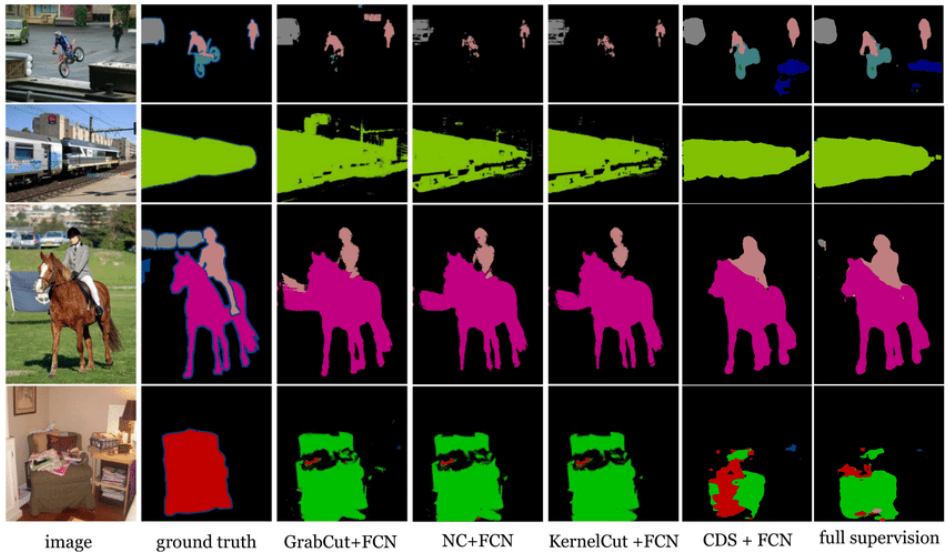
- Precise localization is hard to annotate
- Annotating every pixel leads to heavy tails
- Common solution: annotate few classes (often things), mark rest as “Other”
- Common datasets: PASCAL VOC 2012 (~1500 images, 20 categories), COCO (~100k images, 20 categories)

Dataset

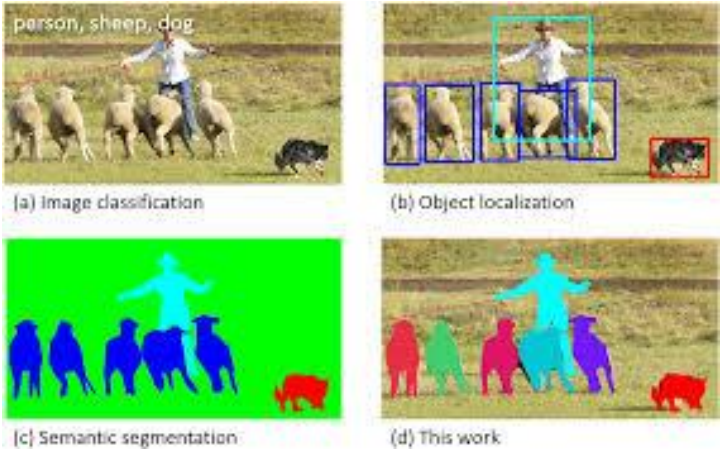
PASCAL Visual Object Classes (VOC)
Annotated images available for 5 tasks—classification, segmentation, detection, action recognition, and person layout.
21 classes of object labels.

- Pixel are labeled as background if they do not belong to any of these classes.

PASCAL Context
An extension of the PASCAL VOC 2010.
Contains more than 400 classes.
Many of the object categories of this dataset are too sparse.



Dataset



Microsoft Common Objects in Context (MS COCO)
Large-scale object detection, segmentation, and captioning dataset.
Includes images of complex everyday scenes and common objects.
Mainly for segmenting individual object instances.

Cityscapes
Focus on semantic understanding of urban street scenes.
Contains a diverse set of stereo video sequences.

Metrics for Segmentation Models

Pixel Accuracy (PA):

PA $\quad \mathrm{PA} = \dfrac{\sum_{i=0}^{K} p_{ii}}{\sum_{i=0}^{K} \sum_{j=0}^{K} p_{ij}}$

The ratio of pixels properly classified, divided by the total number of pixels.

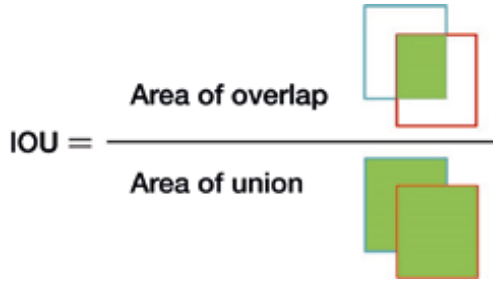
Mean Pixel Accuracy (MPA):

MPA $\quad \mathrm{MPA} = \dfrac{1}{K+1} \sum_{i=0}^{K} \dfrac{p_{ii}}{\sum_{j=0}^{K} p_{ij}}$

The ratio of correct pixels is computed in a per-class manner and then averaged over the total number of classes.

Metrics for Segmentation Models

Intersection over Union (**IoU**



IoU $\text{IoU} = J(A, B) = \frac{|A \cap B|}{|A \cup B|}$

Also called **Jaccard Index**

The most commonly used metrics in semantic segmentation. (mean-IoU/mIoU)

A denotes the ground truth and B denotes the predicted segmentation maps.

TP: True Positive
FP: False Positive
FN: False Negative

Metrics for Segmentation Models

Precision / Recall / F1 score:

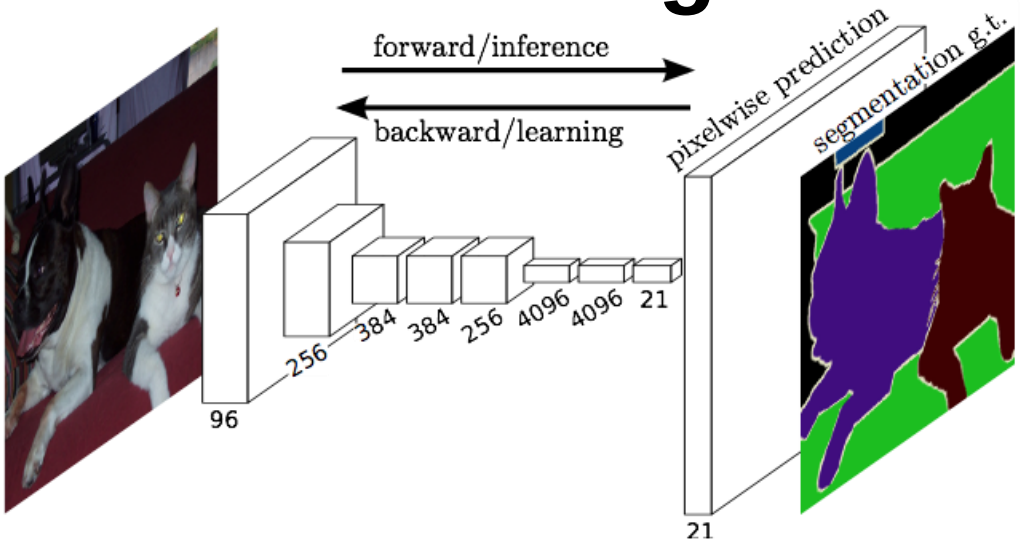
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1-score} = \frac{2\ \text{Prec}\ \text{Rec}}{\text{Prec} + \text{Rec}}$$

F1: harmonic mean of precision and

Predicted Value (predicted by the test)		Actual Value (as confirmed by experiment): positives	Actual Value (as confirmed by experiment): negatives
	positives	**TP** True Positive	**FP** False Positive
	negatives	**FN** False Negative	**TN** True Negative

Fully Convolutional Networks for Semantic Segmentation



Slide credit: Jonathan Long

Overview

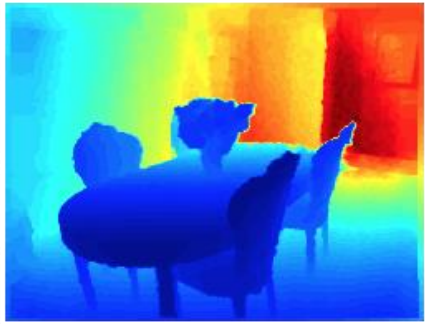
- Reinterpret standard classification convnets as “Fully convolutional” networks (FCN) for semantic segmentation
- Use AlexNet, VGG, and GoogleNet in experiments
- Novel architecture: combine information from different layers for segmentation
- State-of-the-art segmentation for PASCAL VOC 2011/2012, NYUDv2, and SIFT Flow at the time
- Inference less than one fifth of a second for a typical image

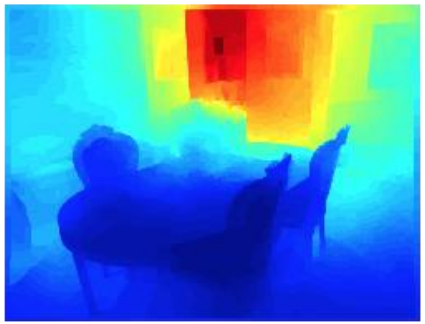
Slide credit: Jonathan Long

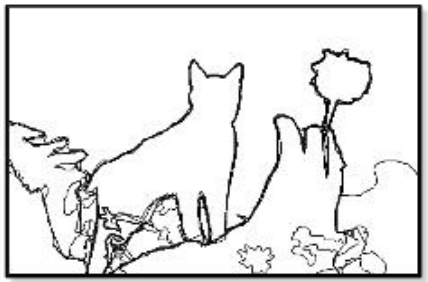
pixels in, pixels out

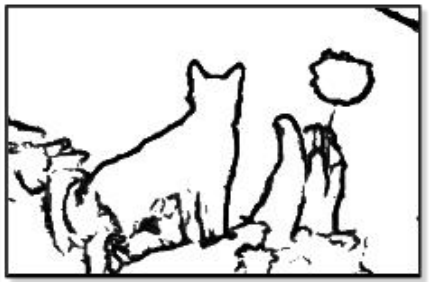
semantic segmentation

monocular depth estimation (Liu et al. 2015)





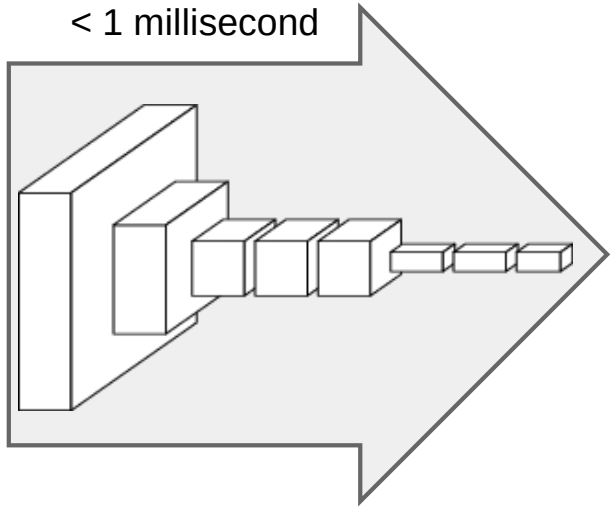


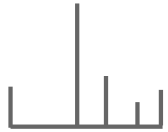


boundary prediction (Xie & Tu 2015)

Slide credit: Jonathan Long

convnets perform classification





“tabby cat”

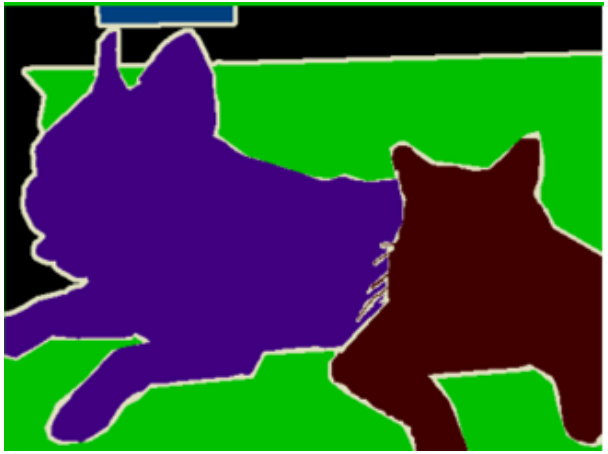
1000-dim vector

end-to-end learning

Slide credit: Jonathan Long

< 1/5 second

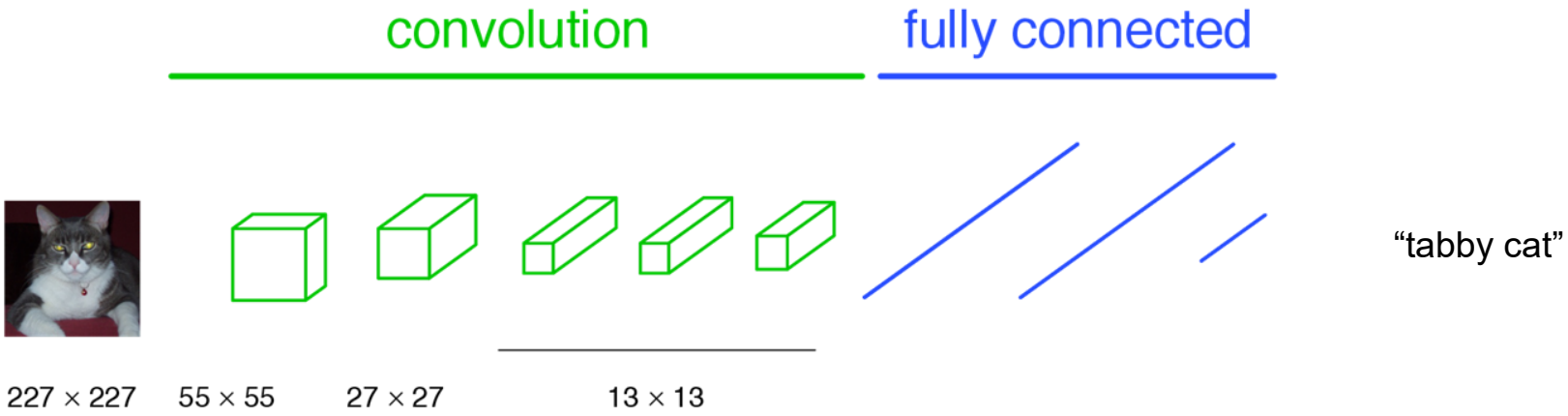
???



end-to-end learning

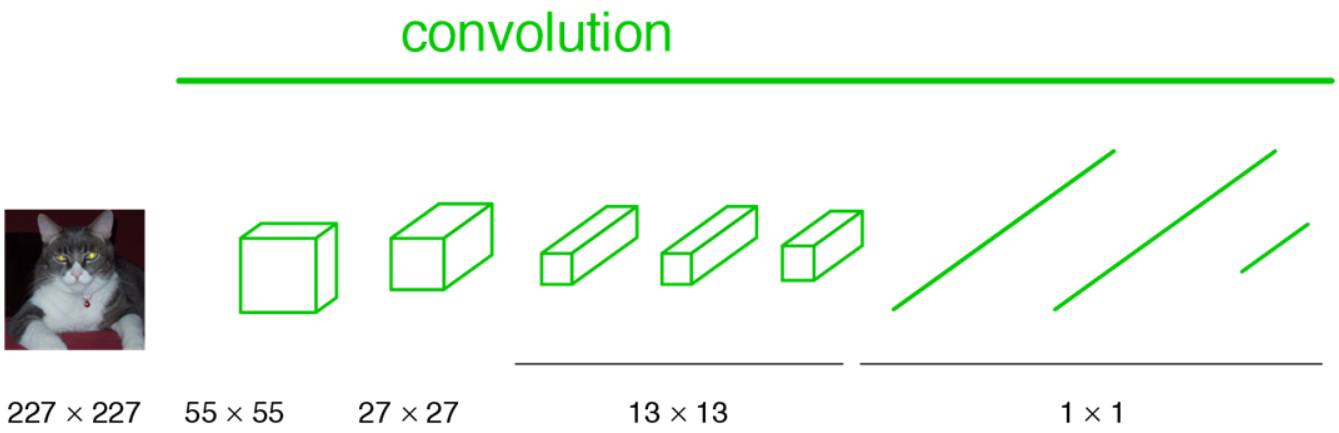
Slide credit: Jonathan Long

a classification network



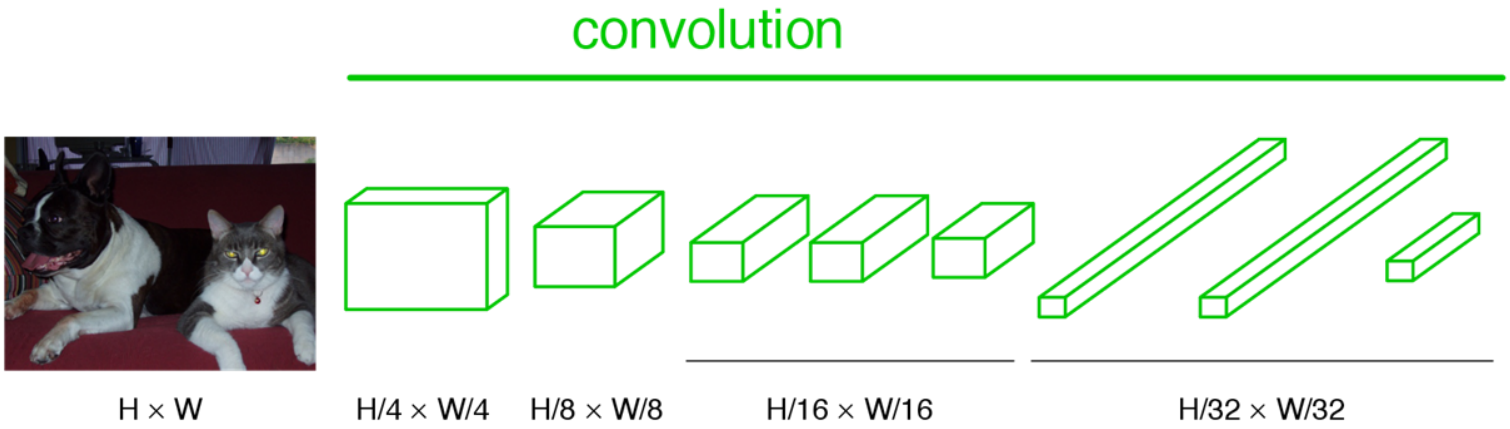
Slide credit: Jonathan Long

becoming fully convolutional



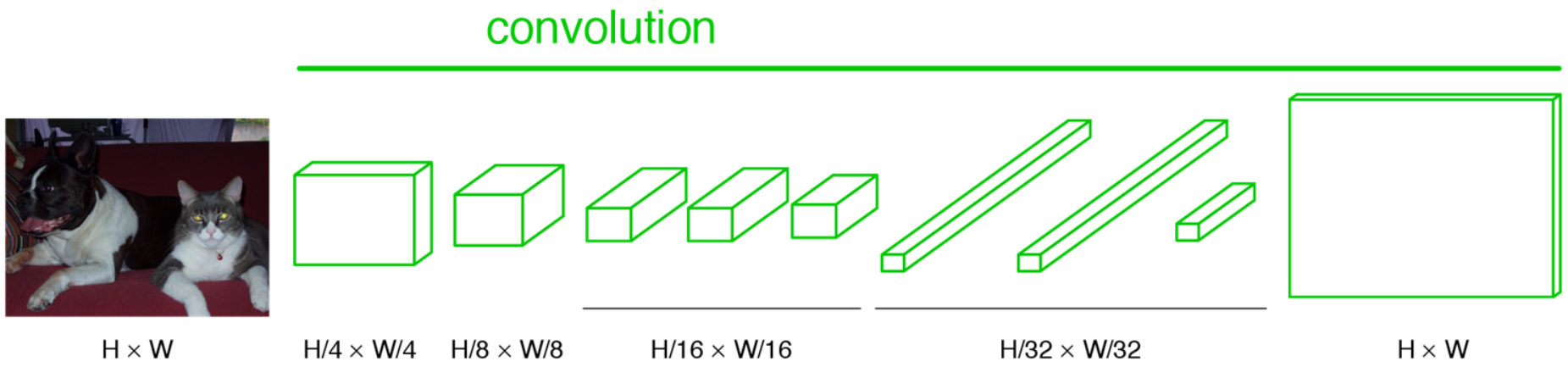
Slide credit: Jonathan Long

becoming fully convolutional



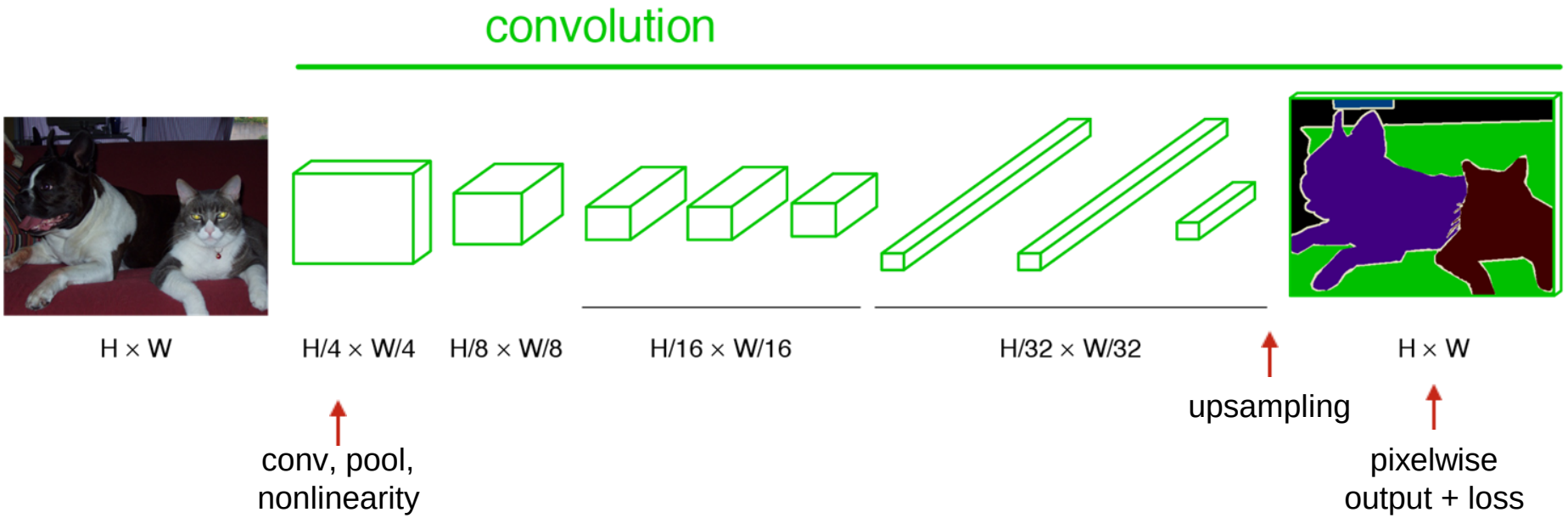
Slide credit: Jonathan Long

upsampling output



Slide credit: Jonathan Long

end-to-end, pixels-to-pixels network



Slide credit: Jonathan Long

Dense Predictions

- Shift-and-stitch: trick that yields dense predictions without interpolation
- Upsampling via deconvolution/up-convolution
- Shift-and-stitch used in preliminary experiments, but not included in final model
- Upsampling found to be more effective and efficient

Classifier to Dense FCN

- Convolutionalize proven classification architectures: AlexNet, VGG, and GoogLeNet (reimplementation)
- Remove classification layer and convert all fully connected layers to convolutions
- Append 1x1 convolution with channel dimensions and predict scores at each of the coarse output locations (21 categories + background for PASCAL)

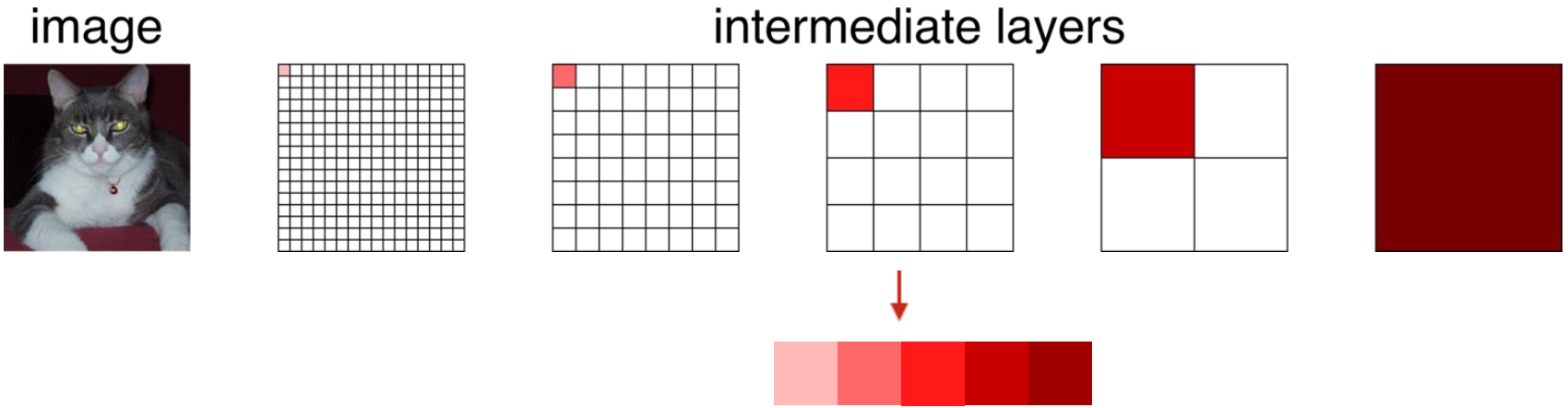
Classifier to Dense FCN

Cast ILSVRC classifiers into FCNs and compare performance on validation set of PASCAL 2011

	FCN-AlexNet	FCN-VGG16	FCN-GoogLeNet[4]
mean IU	39.8	**56.0**	42.5
forward time	50 ms	210 ms	59 ms
conv. layers	8	16	22
parameters	57M	134M	6M
rf size	355	404	907
max stride	32	32	32

spectrum of deep features

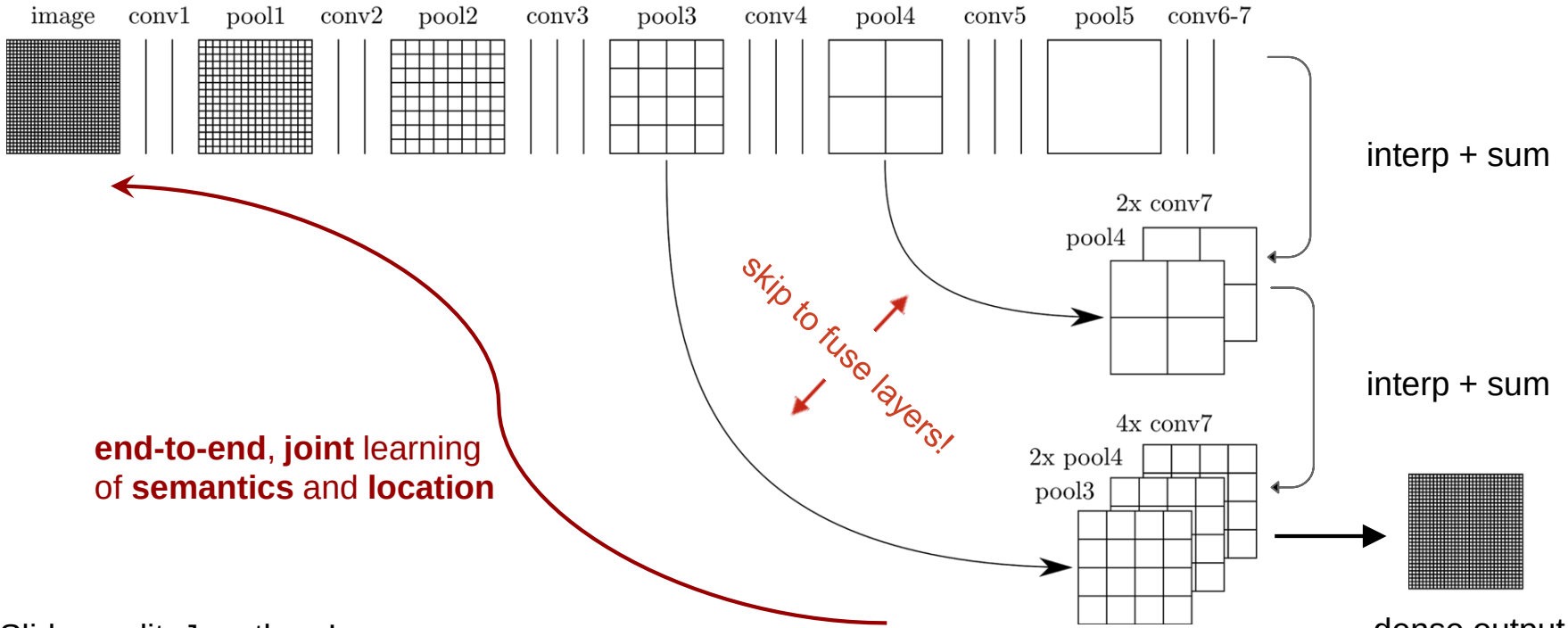
combine *where* (local, shallow) with *what* (global, deep)



Slide credit: Jonathan Long

(cf. Hariharan et al. CVPR15 “hypercolumn”)

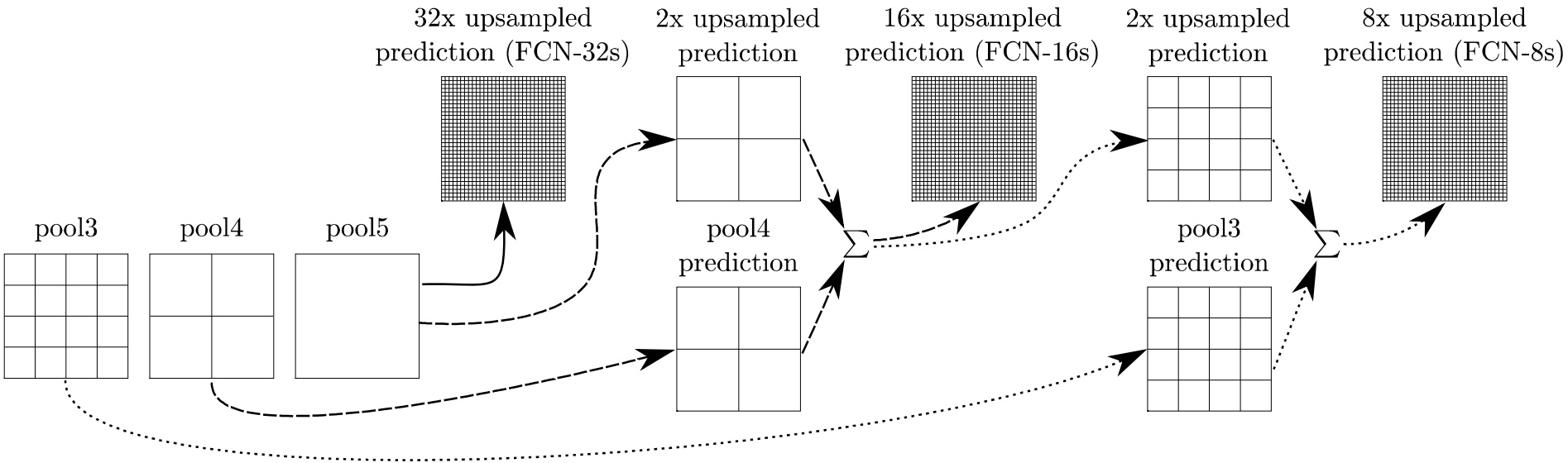
skip layers



end-to-end, **joint** learning
of **semantics** and **location**

Slide credit: Jonathan Long

skip layers


32x upsampled
prediction (FCN-32s)
2x upsampled
prediction
16x upsampled
prediction (FCN-16s)
2x upsampled
prediction
8x upsampled
prediction (FCN-8s)
pool3
pool4
pool5
pool4
prediction
pool3
prediction

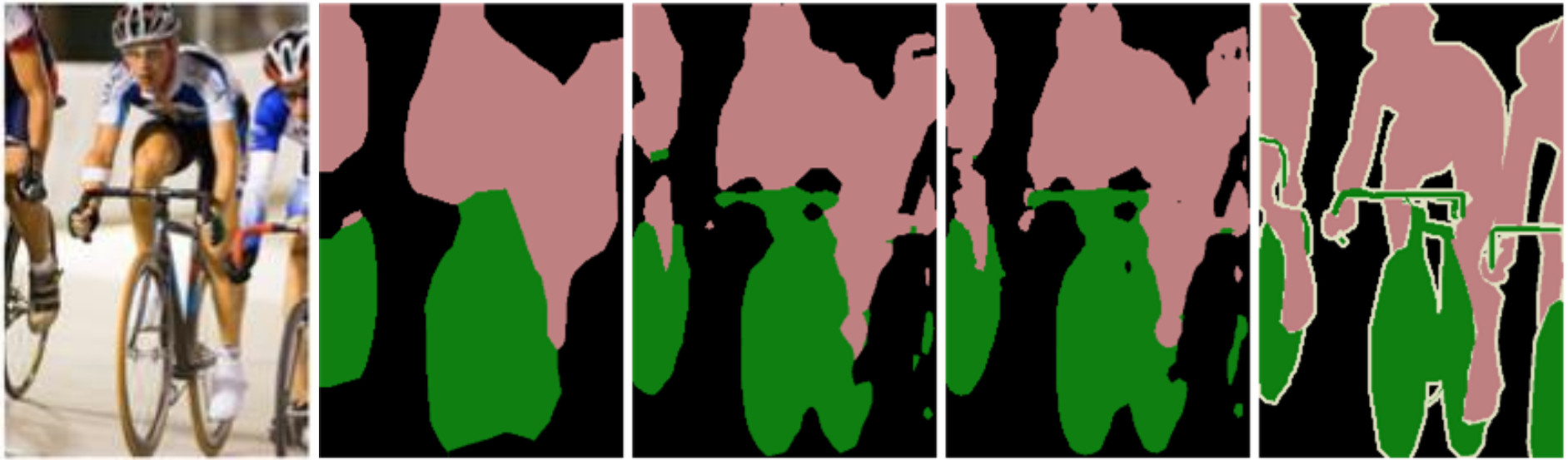
Comparison of skip FCNs

Results on subset of validation set of PASCAL VOC 2011

	pixel acc.	mean acc.	mean IU	f.w. IU
FCN-32s-fixed	83.0	59.7	45.4	72.0
FCN-32s	89.1	73.3	59.4	81.4
FCN-16s	90.0	75.7	62.4	83.0
FCN-8s	**90.3**	**75.9**	**62.7**	**83.2**

skip layer refinement

input image | stride 32 | stride 16 | stride 8 | ground truth

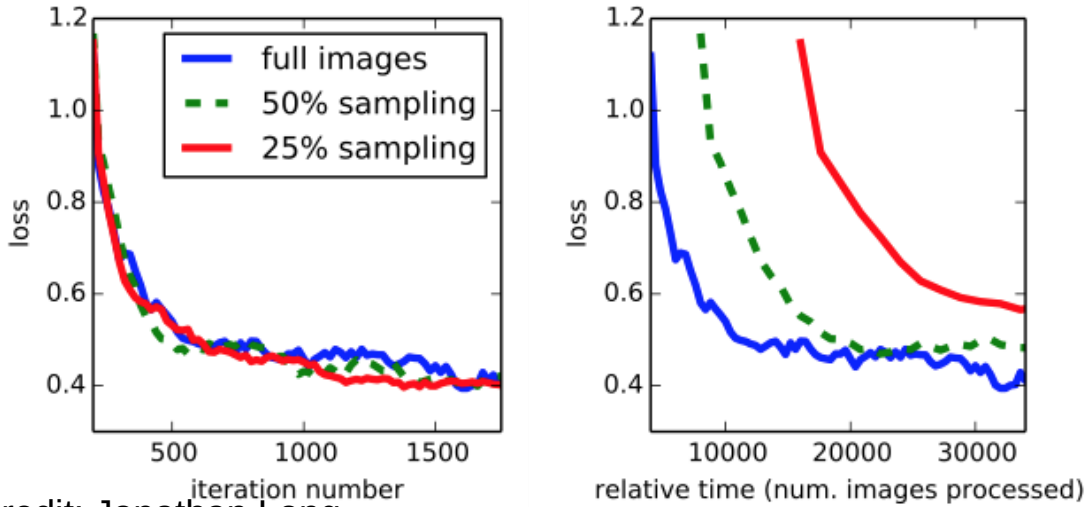


no skips | 1 skip | 2 skips

Slide credit: Jonathan Long

training + testing

- train full image at a time *without patch sampling*
- reshape network to take input of any size
- forward time is ~150ms for 500 x 500 x 21 output



Slide credit: Jonathan Long

Results – PASCAL VOC 2011/12

VOC 2011: 8498 training images (from additional labeled data

	mean IU VOC2011 test	mean IU VOC2012 test	inference time
R-CNN [12]	47.9	-	-
SDS [16]	52.6	51.6	~ 50 s
FCN-8s	**62.7**	**62.2**	~ **175 ms**

Results – NYUDv2

1449 RGB-D images with pixelwise labels → 40 categories

	pixel acc.	mean acc.	mean IU	f.w. IU
Gupta *et al*. [14]	60.3	-	28.6	47.0
FCN-32s RGB	60.0	42.2	29.2	43.9
FCN-32s RGBD	61.5	42.4	30.5	45.5
FCN-32s HHA	57.1	35.2	24.2	40.4
FCN-32s RGB-HHA	64.3	44.9	32.8	48.0
FCN-16s RGB-HHA	65.4	46.1	34.0	49.5

Results – SIFT Flow

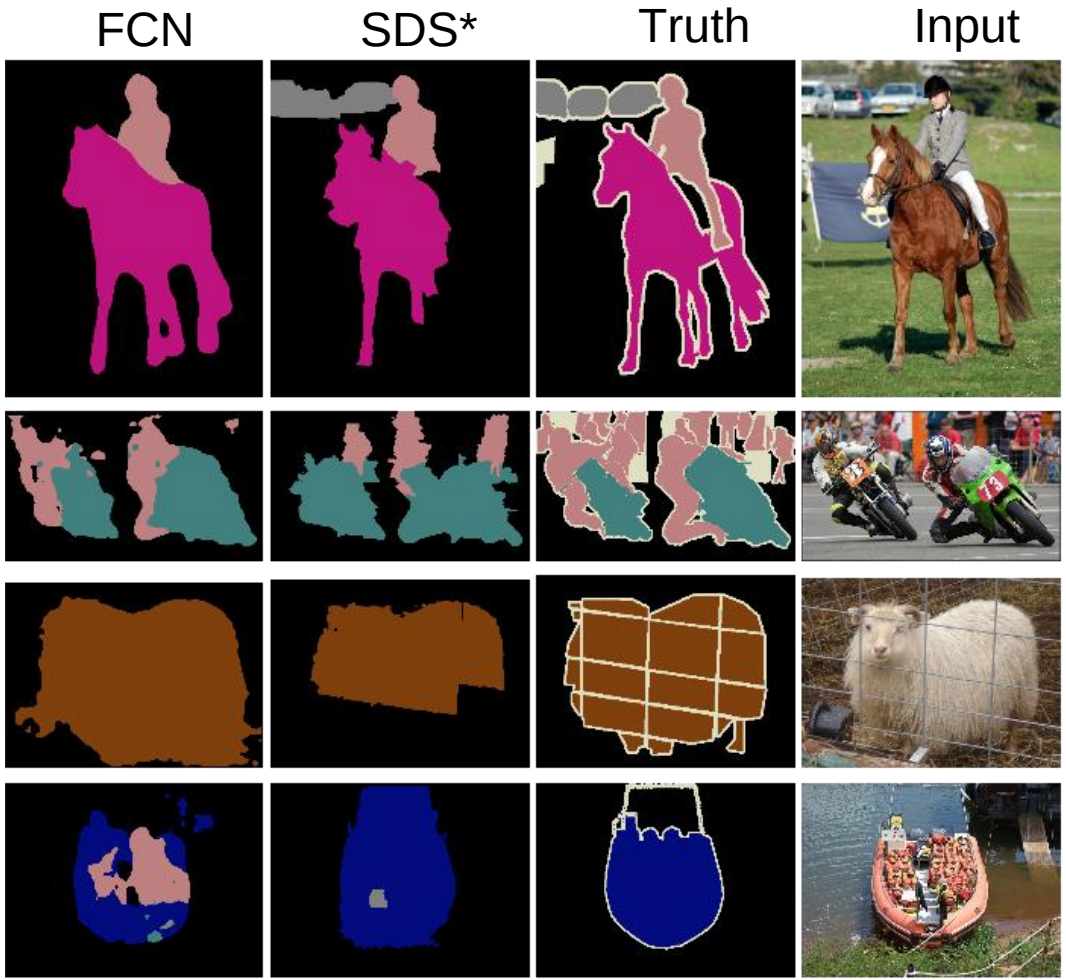
2688 images with pixel labels

→33 semantic categories, 3 geometric categories

Learn both label spaces jointly

→ learning and inference have similar performance and computation as independent models

	pixel acc.	mean acc.	mean IU	f.w. IU	geom. acc.
Liu *et al*. [23]	76.7	-	-	-	-
Tighe *et al*. [33]	-	-	-	-	90.8
Tighe *et al*. [34] 1	75.6	41.1	-	-	-
Tighe *et al*. [34] 2	78.6	39.2	-	-	-
Farabet *et al*. [8] 1	72.3	50.8	-	-	-
Farabet *et al*. [8] 2	78.5	29.6	-	-	-
Pinheiro *et al*. [28]	77.7	29.8	-	-	-
FCN-16s	85.2	51.7	39.5	76.1	94.3

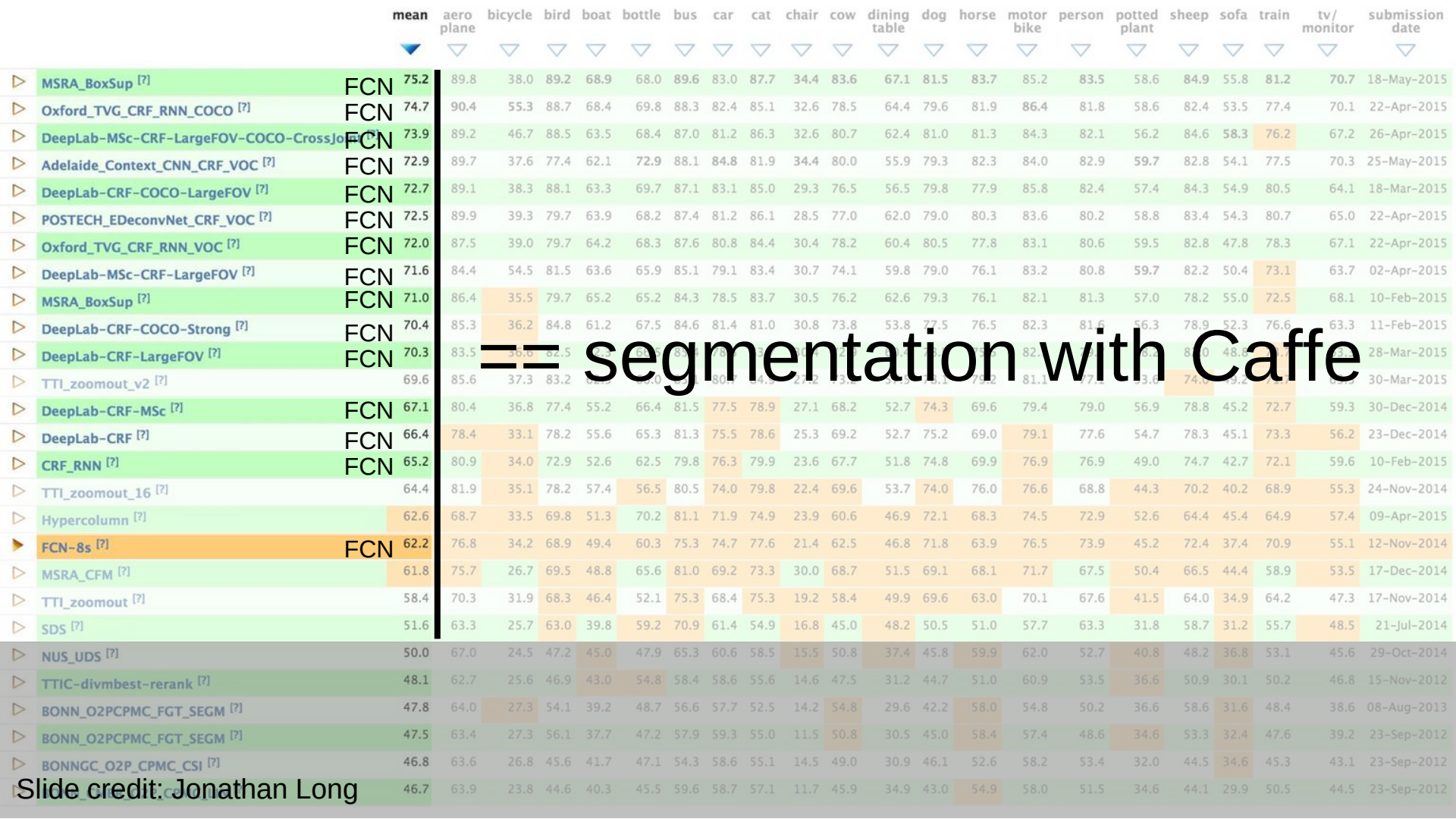


Slide credit: Jonathan Long

Relative to prior state-of-the-art SDS:

- 20% relative improvement for mean IoU
- 286× faster

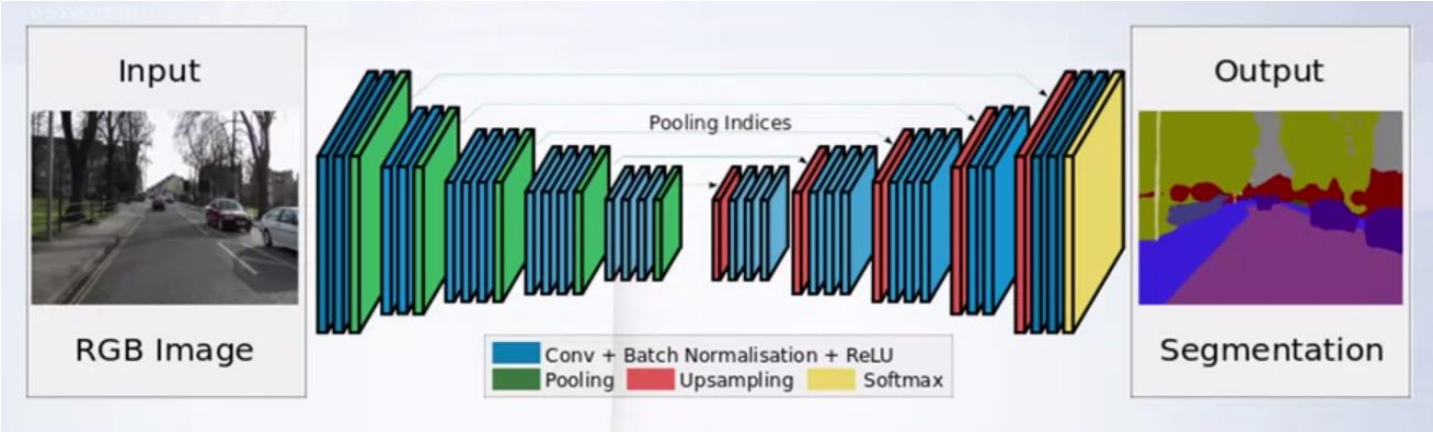
*Simultaneous Detection and Segmentation Hariharan et al. ECCV14



	mean	aero plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	dining table	dog	horse	motor bike	person	potted plant	sheep	sofa	train	tv/ monitor	submission date
MSRA_BoxSup [?] FCN	75.2	89.8	38.0	89.2	68.9	68.0	89.6	83.0	87.7	34.4	83.6	67.1	81.5	83.7	85.2	83.5	58.6	84.9	55.8	81.2	70.7	18-May-2015
Oxford_TVG_CRF_RNN_COCO [?] FCN	74.7	90.4	55.3	88.7	68.4	69.8	88.3	82.4	85.1	32.6	78.5	64.4	79.6	81.9	86.4	81.8	58.6	82.4	53.5	77.4	70.1	22-Apr-2015
DeepLab-MSc-CRF-LargeFOV-COCO-CrossJoint [?] FCN	73.9	89.2	46.7	88.5	63.5	68.4	87.0	81.2	86.3	32.6	80.7	62.4	81.0	81.3	84.3	82.1	56.2	84.6	58.3	76.2	67.2	26-Apr-2015
Adelaide_Context_CNN_CRF_VOC [?] FCN	72.9	89.7	37.6	77.4	62.1	72.9	88.1	84.8	81.9	34.4	80.0	55.9	79.3	82.3	84.0	82.9	59.7	82.8	54.1	77.5	70.3	25-May-2015
DeepLab-CRF-COCO-LargeFOV [?] FCN	72.7	89.1	38.3	88.1	63.3	69.7	87.1	83.1	85.0	29.3	76.5	56.5	79.8	77.9	85.8	82.4	57.4	84.3	54.9	80.5	64.1	18-Mar-2015
POSTECH_EDeconvNet_CRF_VOC [?] FCN	72.5	89.9	39.3	79.7	63.9	68.2	87.4	81.2	86.1	28.5	77.0	62.0	79.0	80.3	83.6	80.2	58.8	83.4	54.3	80.7	65.0	22-Apr-2015
Oxford_TVG_CRF_RNN_VOC [?] FCN	72.0	87.5	39.0	79.7	64.2	68.3	87.6	80.8	84.4	30.4	78.2	60.4	80.5	77.8	83.1	80.6	59.5	82.8	47.8	78.3	67.1	22-Apr-2015
DeepLab-MSc-CRF-LargeFOV [?] FCN	71.6	84.4	54.5	81.5	63.6	65.9	85.1	79.1	83.4	30.7	74.1	59.8	79.0	76.1	83.2	80.8	59.7	82.2	50.4	73.1	63.7	02-Apr-2015
MSRA_BoxSup [?] FCN	71.0	86.4	35.5	79.7	65.2	65.2	84.3	78.5	83.7	30.5	76.2	62.6	79.3	76.1	82.1	81.3	57.0	78.2	55.0	72.5	68.1	10-Feb-2015
DeepLab-CRF-COCO-Strong [?] FCN	70.4	85.3	36.2	84.8	61.2	67.5	84.6	81.4	81.0	30.8	73.8	53.8	77.5	76.5	82.3	81.6	56.3	78.9	52.3	76.6	63.3	11-Feb-2015
DeepLab-CRF-LargeFOV [?] FCN	70.3	83.5	36.6	82.5	62.3	66.2	82.1	79.0	83.2	30.8	73.7	59.3	78.9	77.8	82.6	80.5	59.0	80.8	48.7	73.2	63.3	28-Mar-2015
TTI_zoomout_v2 [?]	69.6	85.6	37.3	83.2	62.5	66.0	82.7	80.7	84.9	27.2	75.2	54.3	78.1	79.2	81.1	77.1	53.6	74.0	49.2	71.7	59.3	30-Mar-2015
DeepLab-CRF-MSc [?] FCN	67.1	80.4	36.8	77.4	55.2	66.4	81.5	77.5	78.9	27.1	68.2	52.7	74.3	69.6	79.4	79.0	56.9	78.8	45.2	72.7	59.3	30-Dec-2014
DeepLab-CRF [?] FCN	66.4	78.4	33.1	78.2	55.6	65.3	81.3	75.5	78.6	25.3	69.2	52.7	75.2	69.0	79.1	77.6	54.7	78.3	45.1	73.3	56.2	23-Dec-2014
CRF_RNN [?] FCN	65.2	80.9	34.0	72.9	52.6	62.5	79.8	76.3	79.9	23.6	67.7	51.8	74.8	69.9	76.9	76.9	49.0	74.7	42.7	72.1	59.6	10-Feb-2015
TTI_zoomout_16 [?]	64.4	81.9	35.1	78.2	57.4	56.5	80.5	74.0	79.8	22.4	69.6	53.7	74.0	76.0	76.6	68.8	44.3	70.2	40.2	68.9	55.3	24-Nov-2014
Hypercolumn [?]	62.6	68.7	33.5	69.8	51.3	70.2	81.1	71.9	74.9	23.9	60.6	46.9	72.1	68.3	74.5	72.9	52.6	64.4	45.4	64.9	57.4	09-Apr-2015
FCN-8s [?] FCN	62.2	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	12-Nov-2014
MSRA_CFM [?]	61.8	75.7	26.7	69.5	48.8	65.6	81.0	69.2	73.3	30.0	68.7	51.5	69.1	68.1	71.7	67.5	50.4	66.5	44.4	58.9	53.5	17-Dec-2014
TTI_zoomout [?]	58.4	70.3	31.9	68.3	46.4	52.1	75.3	68.4	75.3	19.2	58.4	49.9	69.6	63.0	70.1	67.6	41.5	64.0	34.9	64.2	47.3	17-Nov-2014
SDS [?]	51.6	63.3	25.7	63.0	39.8	59.2	70.9	61.4	54.9	16.8	45.0	48.2	50.5	51.0	57.7	63.3	31.8	58.7	31.2	55.7	48.5	21-Jul-2014
NUS_UDS [?]	50.0	67.0	24.5	47.2	45.0	47.9	65.3	60.6	58.5	15.5	50.8	37.4	45.8	59.9	62.0	52.7	40.8	48.2	36.8	53.1	45.6	29-Oct-2014
TTIC-divmbest-rerank [?]	48.1	62.7	25.6	46.9	43.0	54.8	58.4	58.6	55.6	14.6	47.5	31.2	44.7	51.0	60.9	53.5	36.6	50.9	30.1	50.2	46.8	15-Nov-2012
BONN_O2PCPMC_FGT_SEGM [?]	47.8	64.0	27.3	54.1	39.2	48.7	56.6	57.7	52.5	14.2	54.8	29.6	42.2	58.0	54.8	50.2	36.6	58.6	31.6	48.4	38.6	08-Aug-2013
BONN_O2PCPMC_FGT_SEGM [?]	47.5	63.4	27.3	56.1	37.7	47.2	57.9	59.3	55.0	11.5	50.8	30.5	45.0	58.4	57.4	48.6	34.6	53.3	32.4	47.6	39.2	23-Sep-2012
BONNGC_O2P_CPMC_CSI [?]	46.8	63.6	26.8	45.6	41.7	47.1	54.3	58.6	55.1	14.5	49.0	30.9	46.1	52.6	58.2	53.4	32.0	44.5	34.6	45.3	43.1	23-Sep-2012
[illegible]	46.7	63.9	23.8	44.6	40.3	45.5	59.6	58.7	57.1	11.7	45.9	34.9	43.0	54.9	58.0	51.5	34.6	44.1	29.9	50.5	44.5	23-Sep-2012

Slide credit: Jonathan Long

U-Net: Convolutional Network for Segmentation

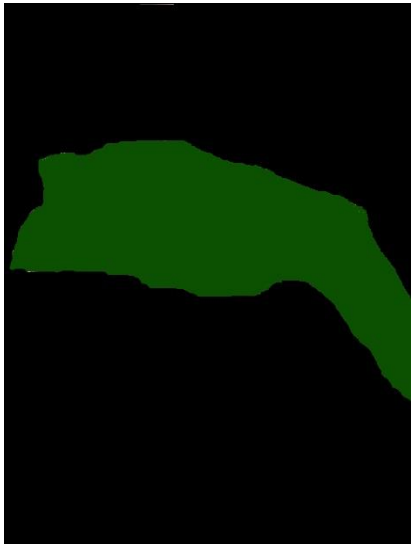


What does a U-Net do?

Input Image

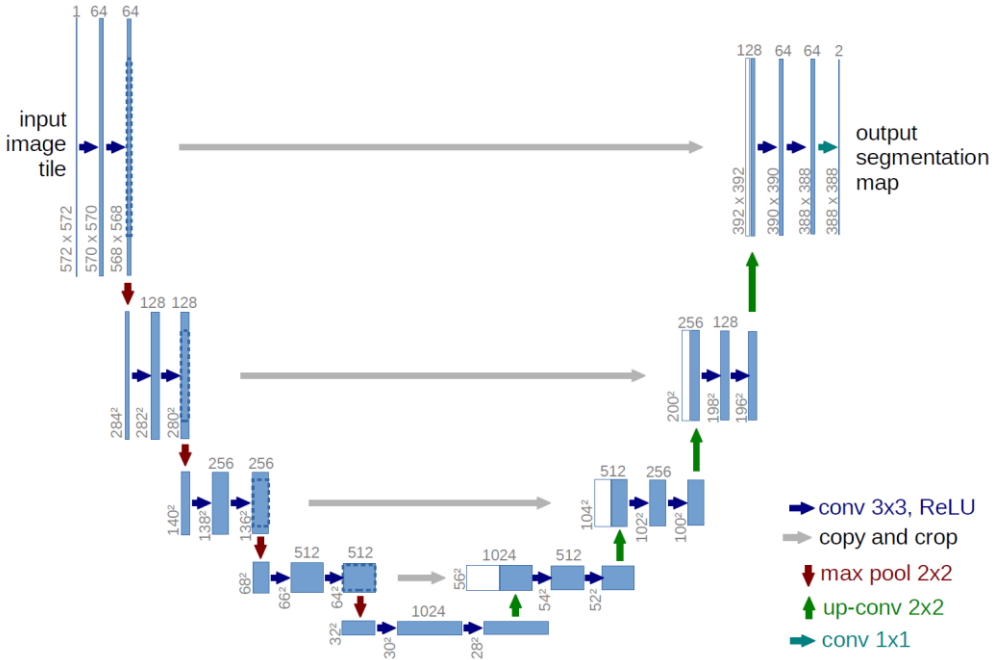
Learns Segmentation

U-Net



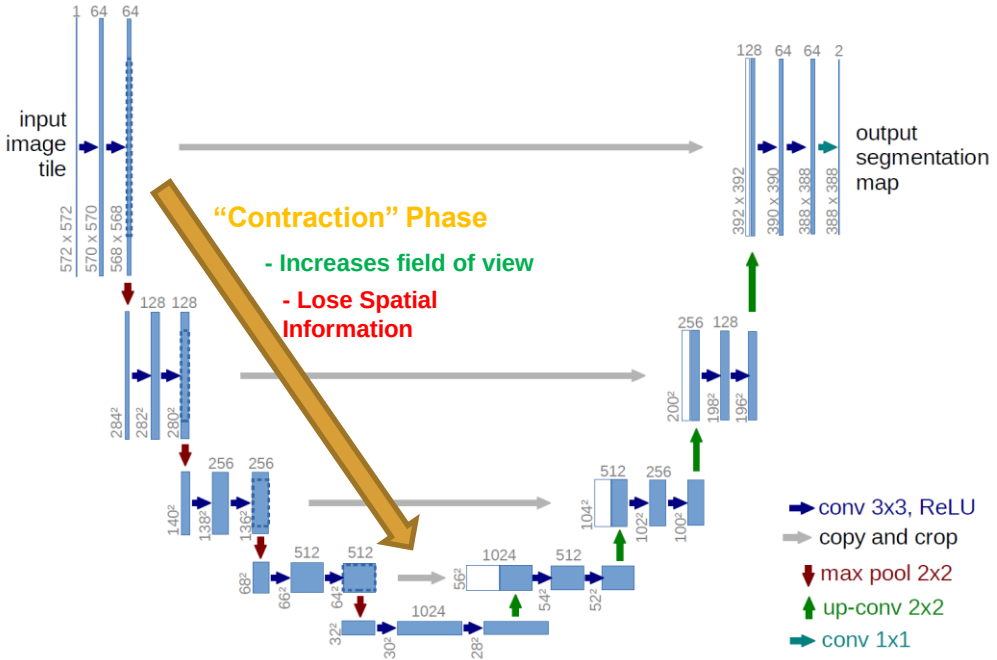
Output Segmentation Map

U-Net Architecture



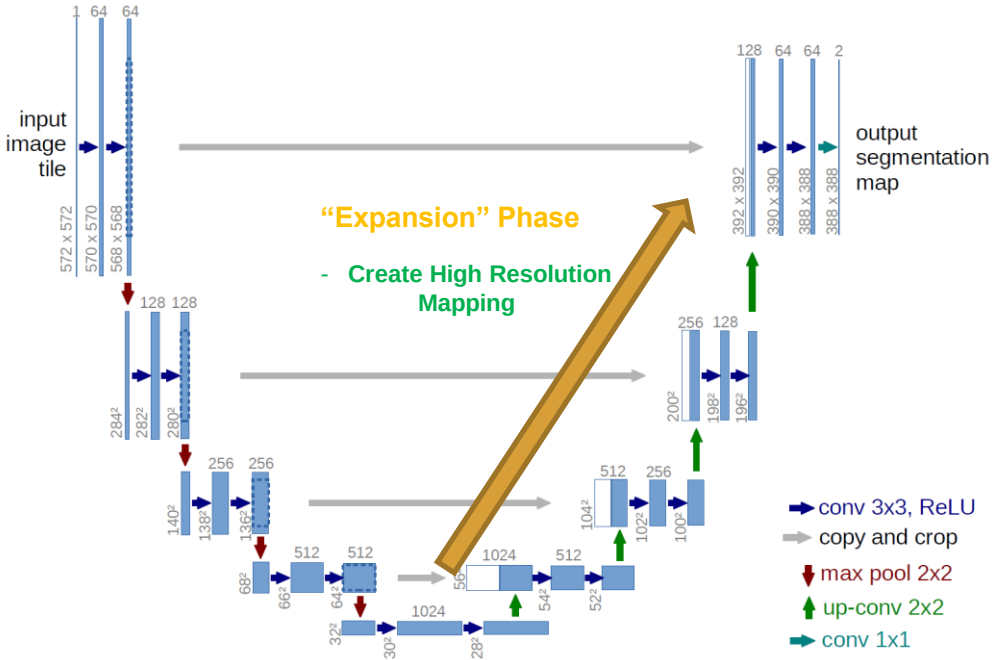
Ronneberger et al. (2015) U-net Architecture

U-Net Architecture



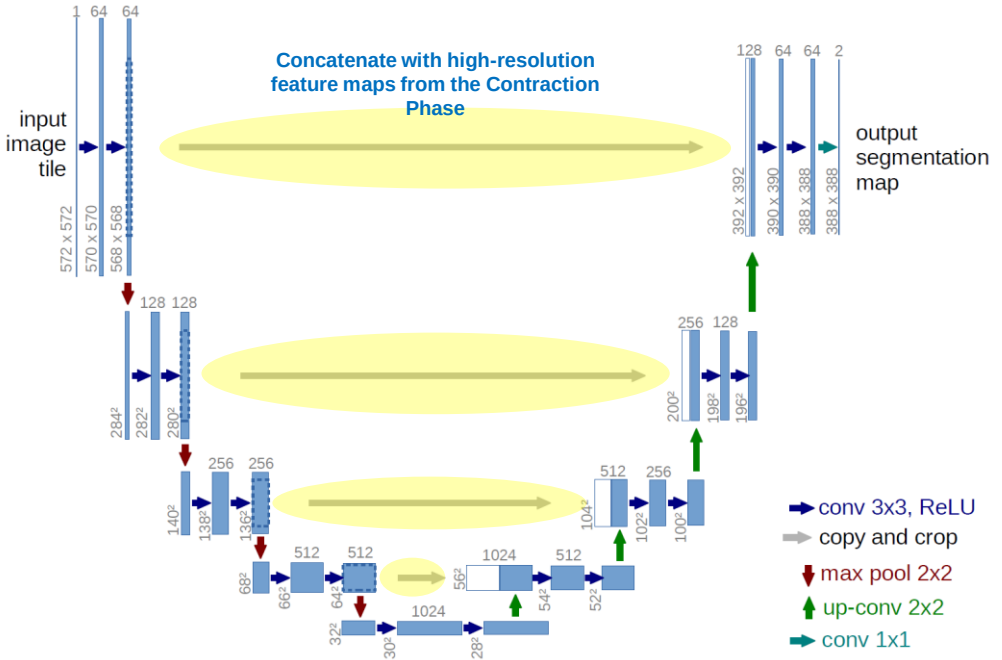
Ronneberger et al. (2015) U-net Architecture

U-Net Architecture



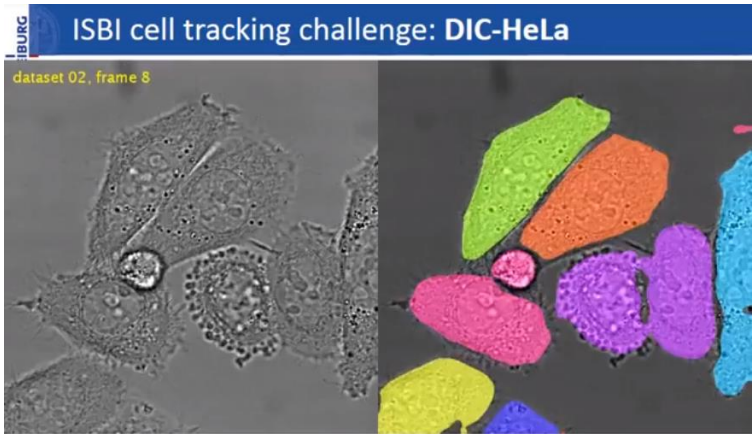
Ronneberger et al. (2015) U-net Architecture

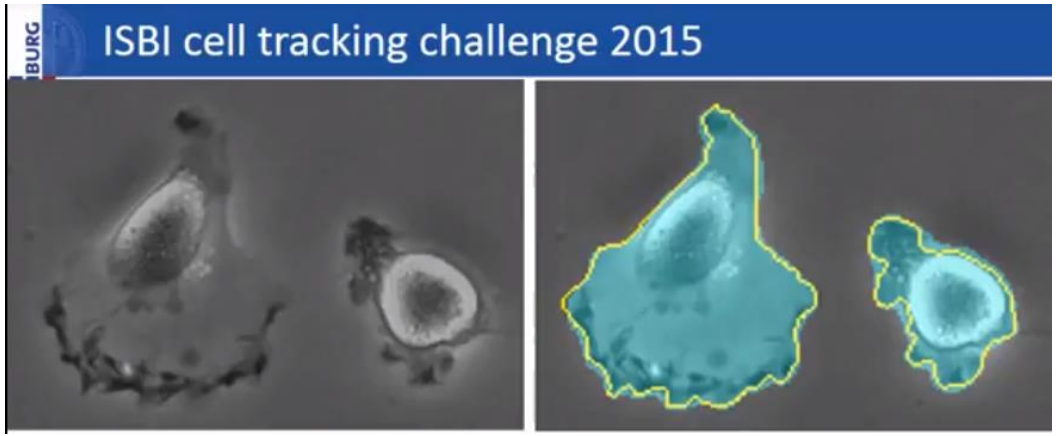
U-Net Architecture



Ronneberger et al. (2015) U-net Architecture

Results





Ronneberger et al. (2015) ISBI cell tracking challenge

Multi-scale Context Aggregation by Dilated Convolutions

Slide Credit: SHEN, Lifeng

Motivation

- Previous **modern image classification networks** integrate multi-scale contextual information via successive **pooling** layers that *reduce resolution* until a global prediction is obtained.

- In contrast, dense prediction calls for multiscale contextual reasoning in combination with full-resolution output.

Conflicting demands:
Multi-scale reasoning and **Full-resolution dense prediction**

Dilated Convolution

The idea of Dilated Convolution is come from the wavelet decomposition. It is also called “atrous convolution” and “hole algorithm”.

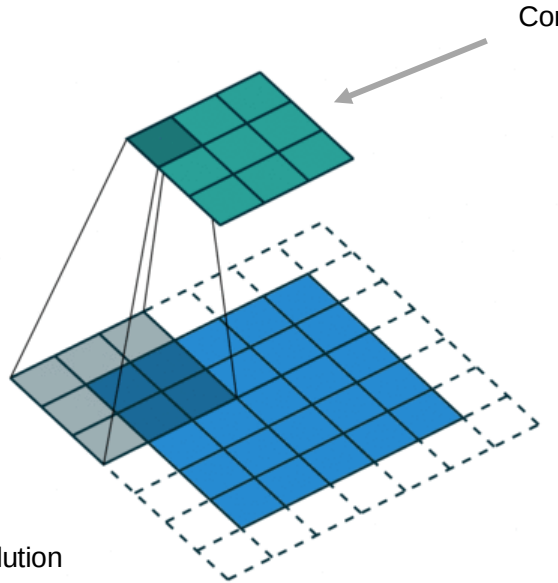
Aims:

- to increase image resolution, enabling dense feature extraction in CNNs.
- to enlarge the reception field of convolutional kernels.

The main idea of dilated convolution:

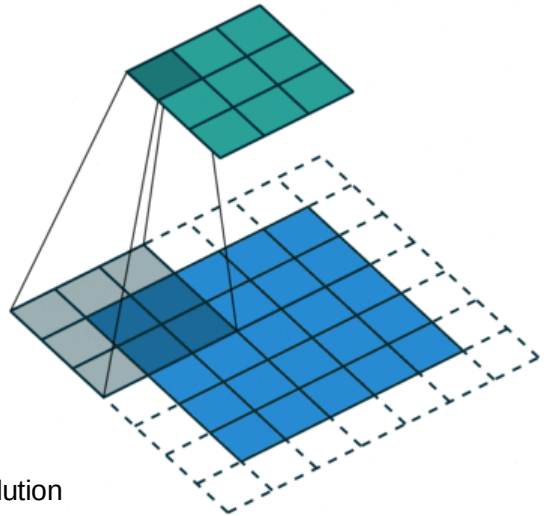
- insert “holes”(zeros) between pixels in convolutional kernels

Dilated Convolution

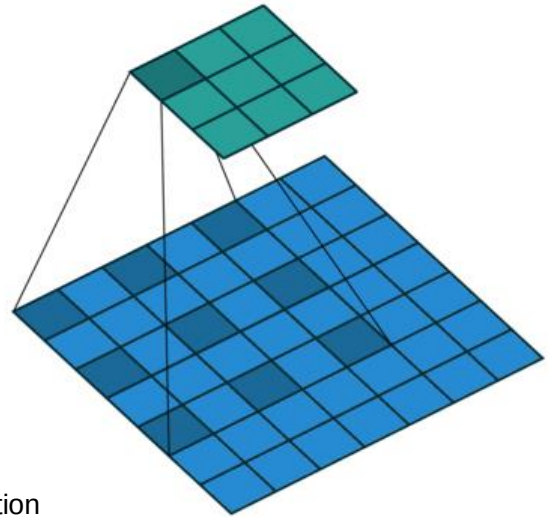


Standard Convolution

Dilated Convolution



Standard Convolution

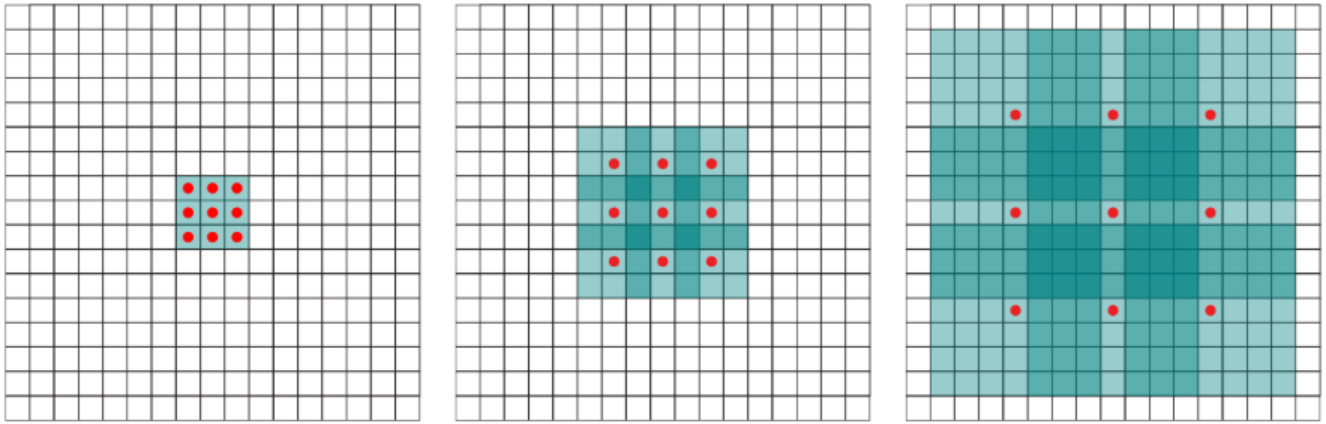


Dilated Convolution

- We can see that **the receptive field is larger** compared with the standard one.

Dilated Convolution

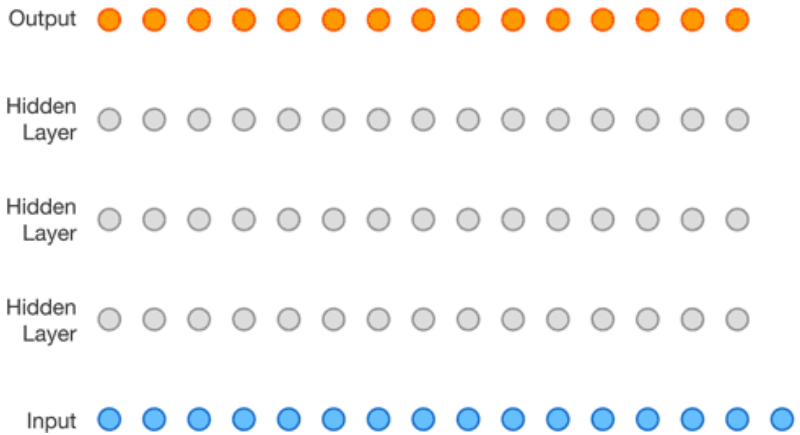
Multi-scale Context Aggregation



(a) (b) (c)

Figure 1: Systematic dilation supports exponential expansion of the receptive field without loss of resolution or coverage. (a) F_1 is produced from F_0 by a 1-dilated convolution; each element in F_1 has a receptive field of 3×3. (b) F_2 is produced from F_1 by a 2-dilated convolution; each element in F_2 has a receptive field of 7×7. (c) F_3 is produced from F_2 by a 4-dilated convolution; each element in F_3 has a receptive field of 15×15. The number of parameters associated with each layer is identical. The receptive field grows exponentially while the number of parameters grows linearly.

Dilated Convolution



Increasing reception field using dilated convolution

Wavenet: A generative model for raw audio

A Oord, S Dieleman, H Zen, K Simonyan… - arXiv preprint arXiv …, 2016 - arxiv.org

This paper introduces WaveNet, a deep neural network for generating raw audio waveforms. The model is fully probabilistic and autoregressive, with the predictive distribution for each audio sample conditioned on all previous ones; nonetheless we show that it can be …

☆ 99 被引用次数：1235 相关文章 所有 13 个版本 »

Systematic dilation supports exponential expansion of the receptive field without loss of resolution or coverage.

The number of parameters associated with each layer is identical. The receptive field grows exponentially while the number of parameters grows linearly.

Dilated Convolution

Note that the frontend module that provides the input to the context network in our experiments produces feature maps at 64x64 resolution.

Layer	1	2	3	4	5	6	7	8
Convolution	3×3	3×3	3×3	3×3	3×3	3×3	3×3	1×1
Dilation	1	1	2	4	8	16	1	1
Truncation	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Receptive field	3×3	5×5	9×9	17×17	33×33	65×65	67×67	67×67

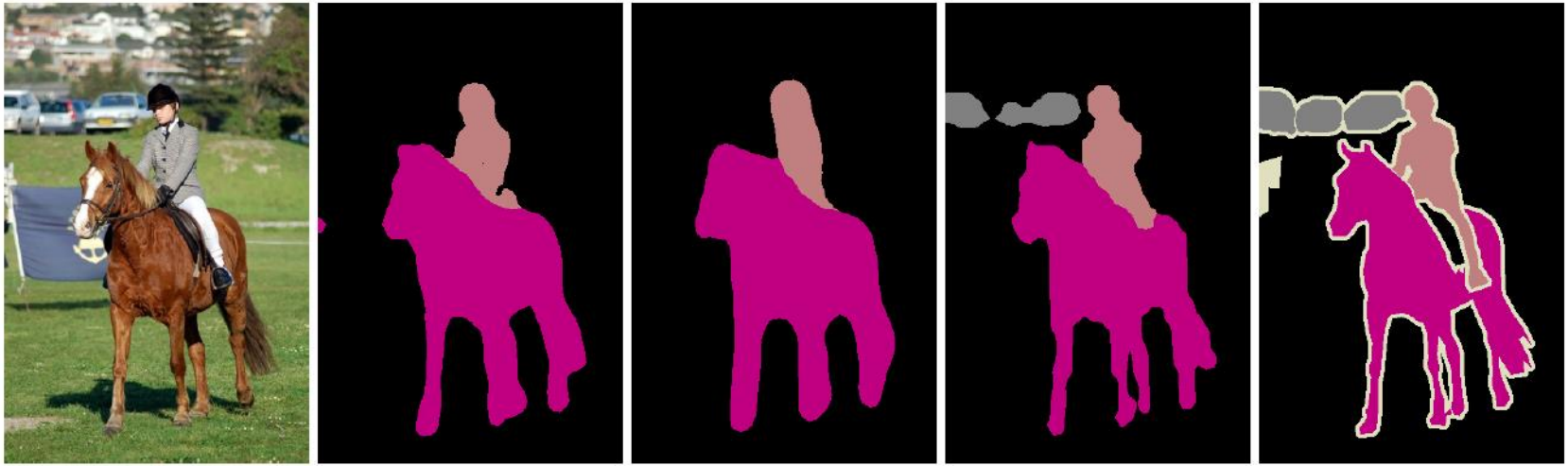
Table 1: Context network architecture (w/o pooling)

We implemented and trained a front-end prediction module that takes a color image as input and **produces C = 21 feature maps as output**. The front-end module follows the work of Long et al. (2015) and Chen et al. (2015a), but was implemented separately. We adapted the VGG-16 network for dense prediction and **removed the last two pooling and striding layers**. Specifically, each of these pooling and striding layers was removed and **convolutions in all subsequent layers were dilated by a factor of 2 for each pooling layer that was ablated**. Thus convolutions in the final layers, which follow both ablated pooling layers, are dilated by a factor of 4. This enables initialization with the parameters of the original classification network, but produces higher-resolution output.

Results

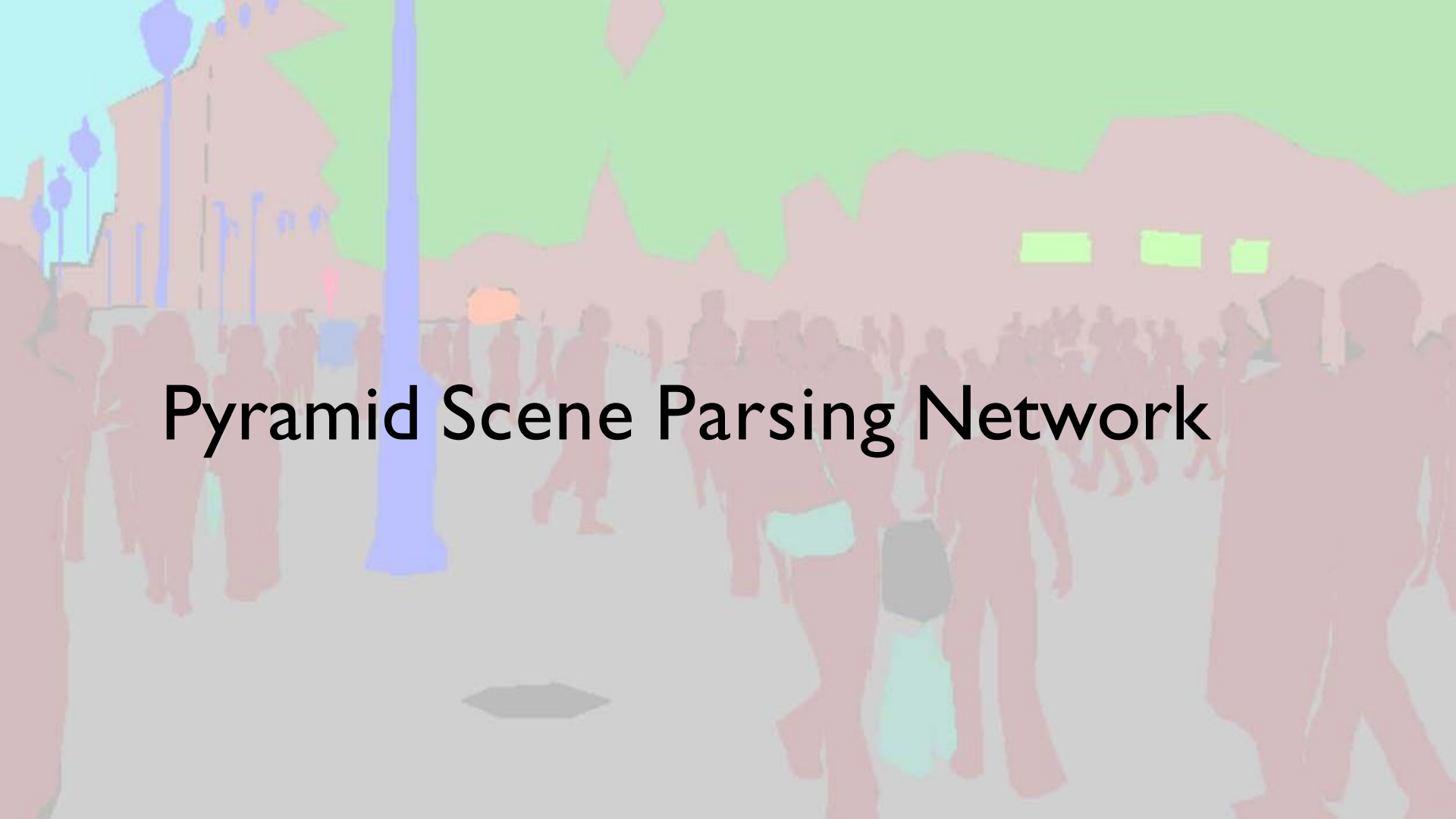
	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean IoU
FCN-8s	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
DeepLab	72	31	71.2	53.7	60.5	77	71.9	73.1	25.2	62.6	49.1	68.7	63.3	73.9	73.6	50.8	72.3	42.1	67.9	52.6	62.1
DeepLab-Msc	74.9	34.1	72.6	52.9	61.0	77.9	73.0	73.7	26.4	62.2	49.3	68.4	64.1	74.0	75.0	51.7	72.7	42.5	67.2	55.7	62.9
Our front end	**82.2**	**37.4**	**72.7**	**57.1**	**62.7**	**82.8**	**77.8**	**78.9**	**28**	**70**	**51.6**	**73.1**	**72.8**	**81.5**	**79.1**	**56.6**	**77.1**	**49.9**	**75.3**	**60.9**	**67.6**

Table 2: Our front-end prediction module is simpler and more accurate than prior models. This table reports accuracy on the VOC-2012 test set.



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

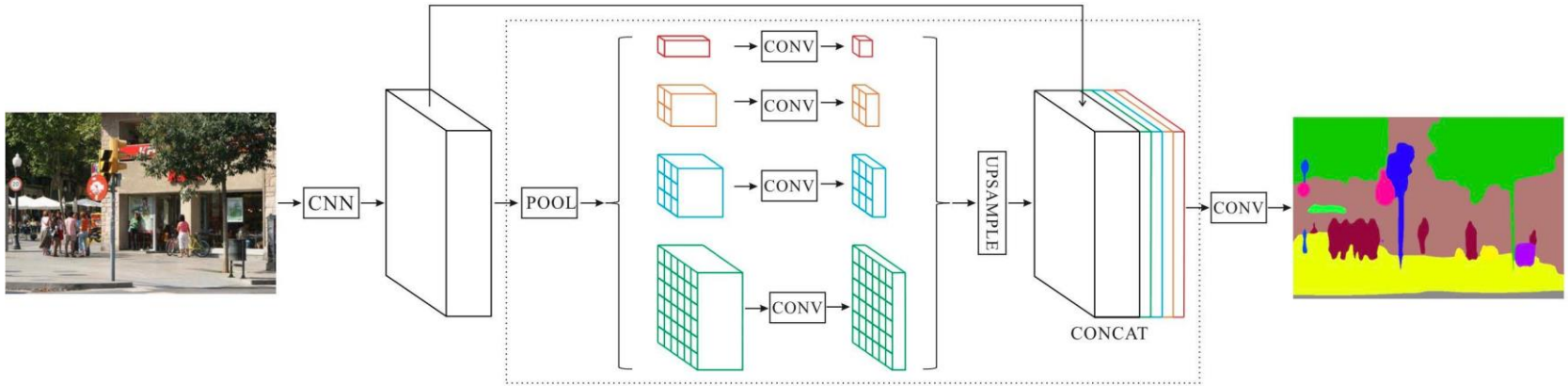
Deep Lab: Chen, Liang-Chieh, et al. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In ICLR, 20



Pyramid Scene Parsing Network

Overview

- Introduce Pyramid Pooling Module for better context grasp with sub-region awareness



Why did I choose this paper?

- Presented in CVPR 2017
- 1st place in ImageNet Scene Parsing Challenge 2016 (ADE20K)
- **was** 1st place in Cityscapes leaderboard

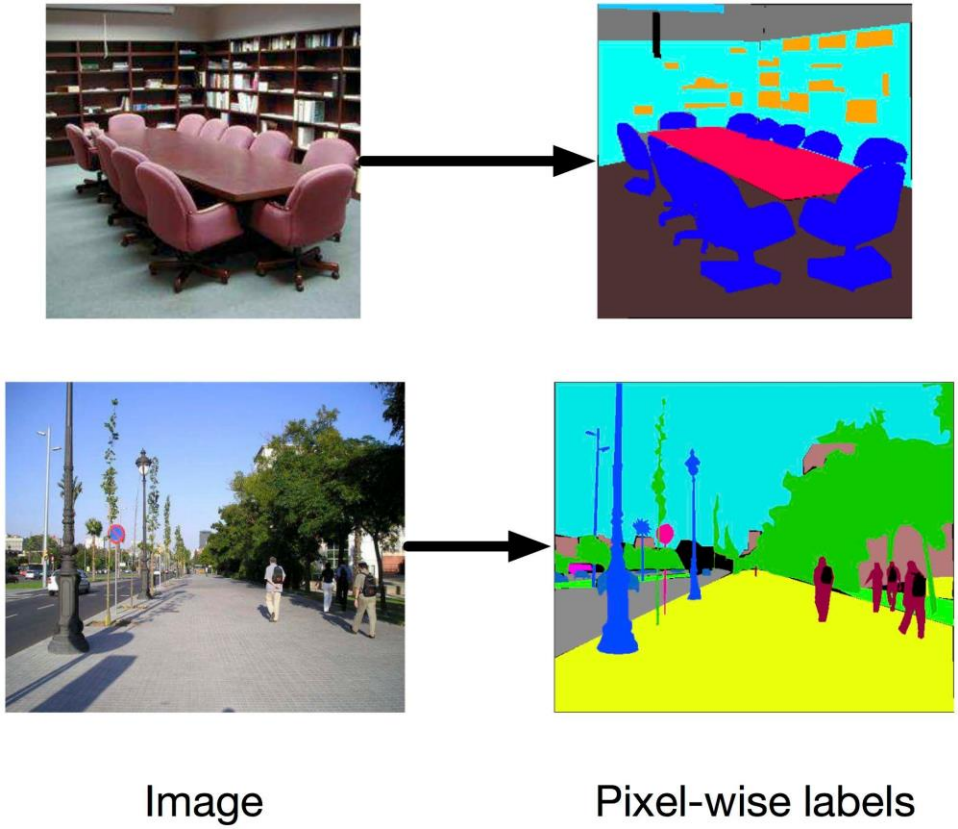
	name	fine	coarse	16-bit	depth	video	sub	IoU class	iIoU class	IoU category	iIoU category	Runtime [s]	code
	motovis	yes	yes	no	no	no	no	81.3	57.7	91.5	80.7	n/a	no
	PSPNet	yes	yes	no	no	no	no	81.2	59.6	91.2	79.2	n/a	yes
	ResNet-38	yes	yes	no	no	no	no	80.6	57.8	91.0	79.1	n/a	yes
	NetWarp	yes	yes	no	no	yes	no	80.5	59.5	91.0	79.8	n/a	no
	TuSimple_Coarse	yes	yes	no	no	no	no	80.1	56.9	90.7	77.8	n/a	yes
	SegModel	yes	yes	no	no	no	no	79.2	56.4	90.4	77.0	n/a	no

Agenda

1. Common building blocks in semantic segmentation
2. Major Issue
3. Prior Work
4. Pyramid Pooling Module
5. Experiment results

Semantic Segmentation

- Predict pixel-wise labels from natural images
- Each pixel in an image belongs to an object class
- So it's **not instance-aware** 😩



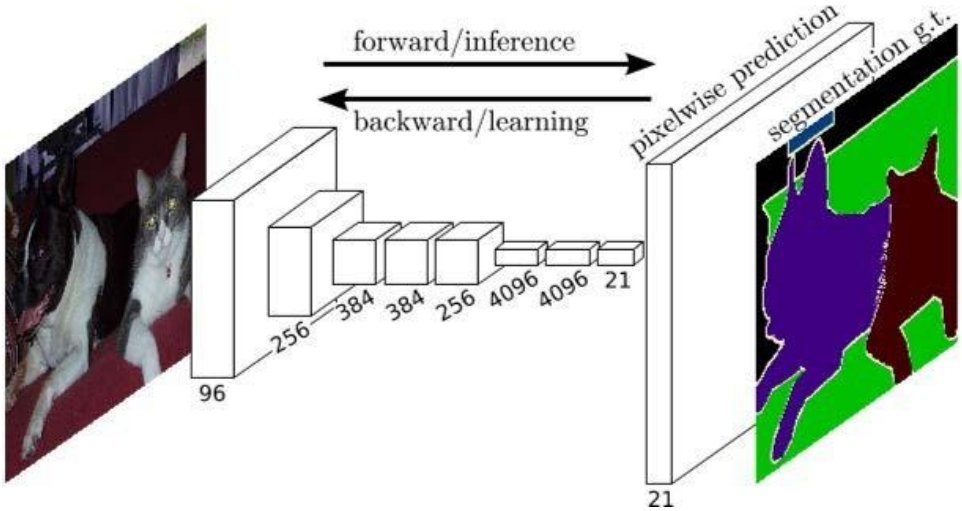
Image

Pixel-wise labels

Common Building Blocks (1)

Fully convolutional network (FCN)[1]

- A deep convolutional neural network which doesn't include any fully-connected layers
- Almost all recent methods are based on FCN
- Typically pre-trained with ImageNet under classification problem setting

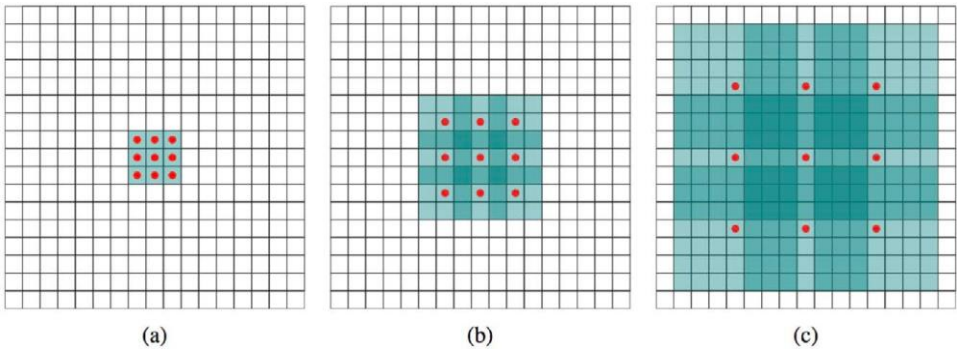


[1] "Semantic Contours from Inverse Detectors", ICCV 2011, http://home.bharathh.info/pubs/codes/SBD/download.html

Common Building Blocks (2)

Dilated convolution[1]

- Widen receptive field without reducing feature map resolution
- Important for leveraging global context prior efficiently

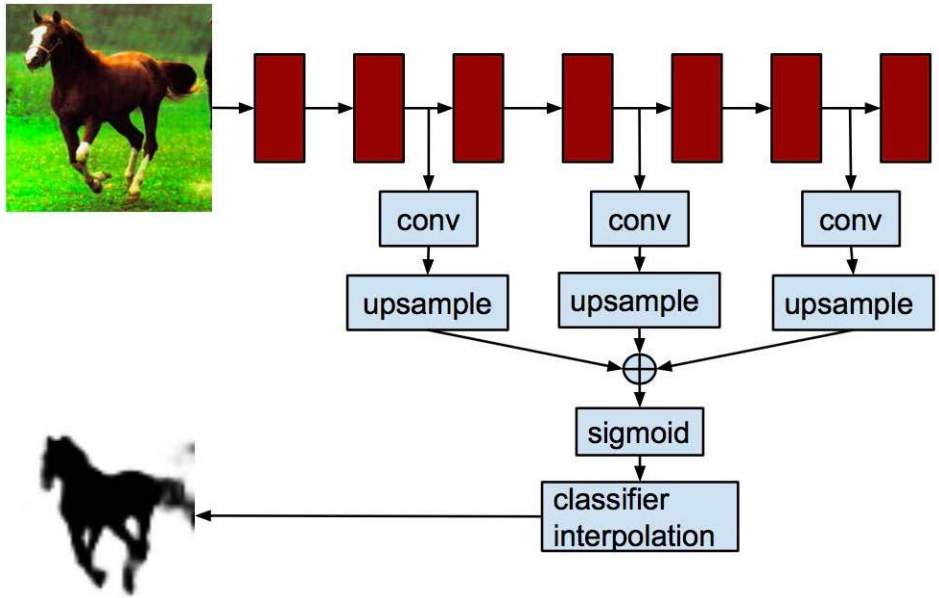


[1] "Semantic Contours from Inverse Detectors", ICCV 2011, http://home.bharathh.info/pubs/codes/SBD/download.html

Common Building Blocks (3)

Multi-scale feature ensemble

- Higher-layer feature contains more semantic meaning and less location information
- Combining multi-scale features can improve the performance[1]

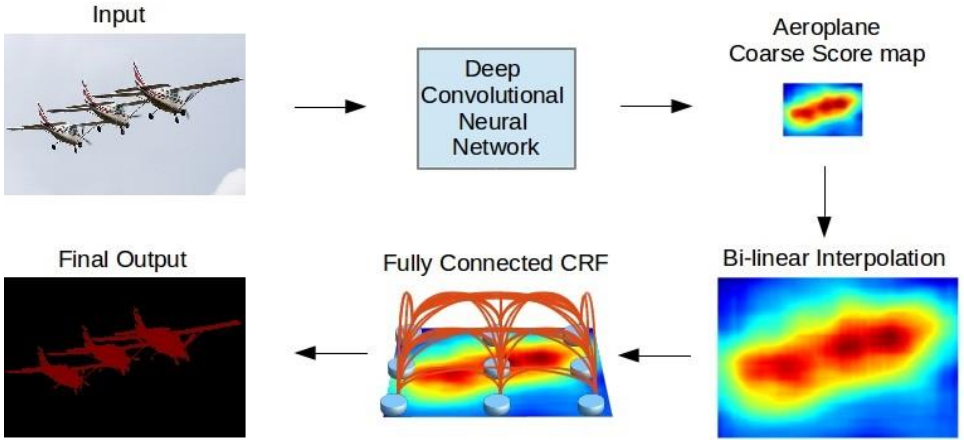


[1] "Semantic Contours from Inverse Detectors", ICCV 2011, http://home.bharathh.info/pubs/codes/SBD/download.html

Common Building Blocks (4)

Conditional random field (CRF)

- Post-processing to refine the segmentation result (DeepLab[1])
- Some following methods refined network via end-to-end modeling (DPN[2], CRF as RNN[3], Detections and Superpixels[4])



[1] "Semantic Contours from Inverse Detectors", ICCV 2011, http://home.bharathh.info/pubs/codes/SBD/download.html

[2] "Semantic image segmentation via deep parsing network", ICCV 2015

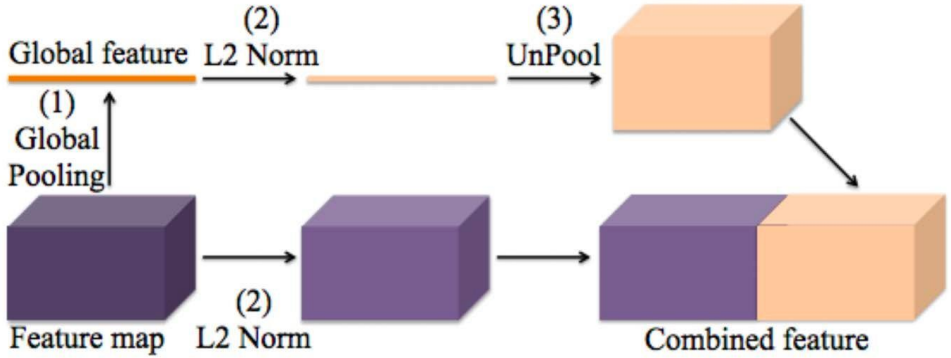
[3] "Conditional random fields as recurrent neural networks", ICCV 2015

[4] "Higher order conditional random fields in deep neural networks", ECCV 2016

Common Building Blocks (5)

Global average pooling (GAP)

- ParsenNet[1] proved that global average pooling with FCN can improve semantic segmentation results
- But the global descriptors used in the paper are not representative enough for some challenging datasets like ADE20K



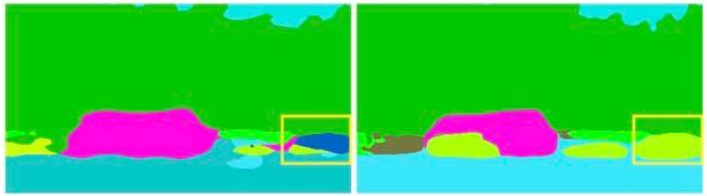
[1] "Semantic Contours from Inverse Detectors", ICCV 2011, http://home.bharathh.info/pubs/codes/SBD/download.html

Major Issue (1)

Mismatched relationship

- Co-occurrent visual patterns imply some contexts
  - e.g., an airplane is likely to fly in sky while not over a road
- Lack of the ability to collect contextual information increases the chance of misclassification
- In the right figure, FCN predicts the boat in the yellow box as a "car" based on its appearance

(a) Image (b) Ground Truth



(c) FCN (d) PSPNet

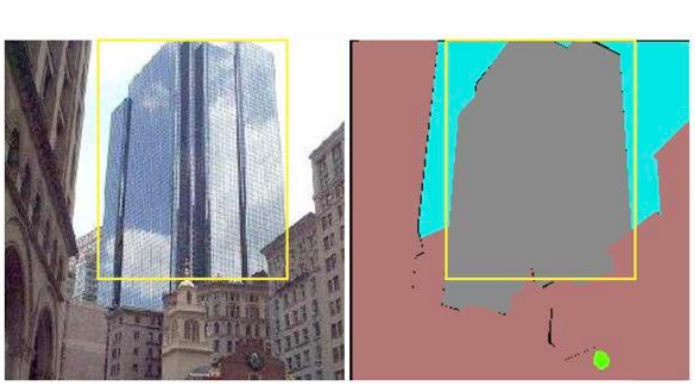
(e) ColorMap

Major Issue (2)

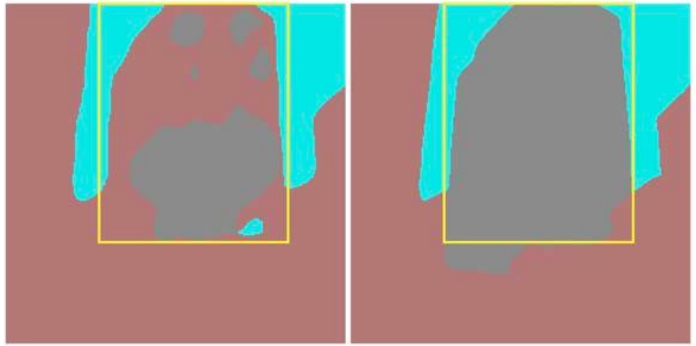
Confusing Classes

- There are confusing classes in major datasets: field and earth; mountain and hill; wall, house, building and skyscraper, etc.
- **The expert human annotator still makes 17.6% pixel error for ADE20K**[1]
- FCN predicts the object in the box as part of skyscraper and part of building but **the whole object should be either skyscraper or building, not both**
- Utilizing the relationship between classes is important

[1] "Semantic Contours from Inverse Detectors", ICCV 2011, http://home.bharathh.info/pubs/codes/SBD/download.html



(a) Image (b) Ground Truth



(c) FCN (d) PSPNet

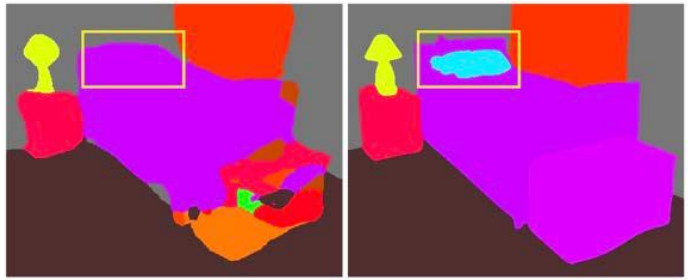
(e) ColorMap

Major Issue (3)

Inconspicuous Classes

- Small objects like streetlight and signboard are inconspicuous and hard to find while they may be important
- Big objects may appear in discontinuous, but FCN couldn't label the pillow which has similar appearance with the sheet correctly
- To improve performance for small or very big objects, **sub-regions should be paid more attention**

(a) Image (b) Ground Truth



(c) FCN (d) PSPNet

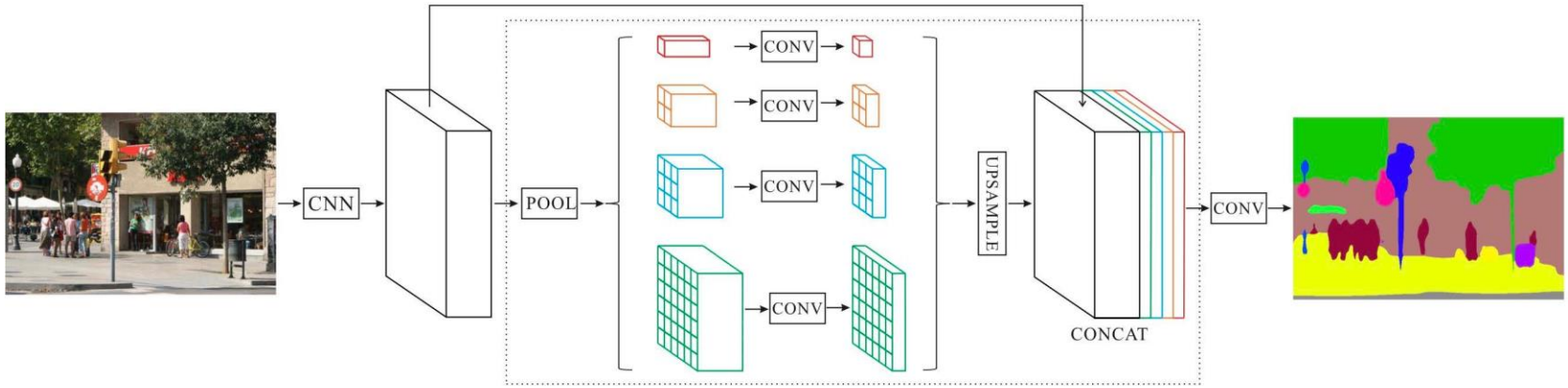
(e) ColorMap

Summary of Issues

- Use co-occurrent visual patterns as context
- Consider relationship between classes
- Sub-regions should be paid more attention

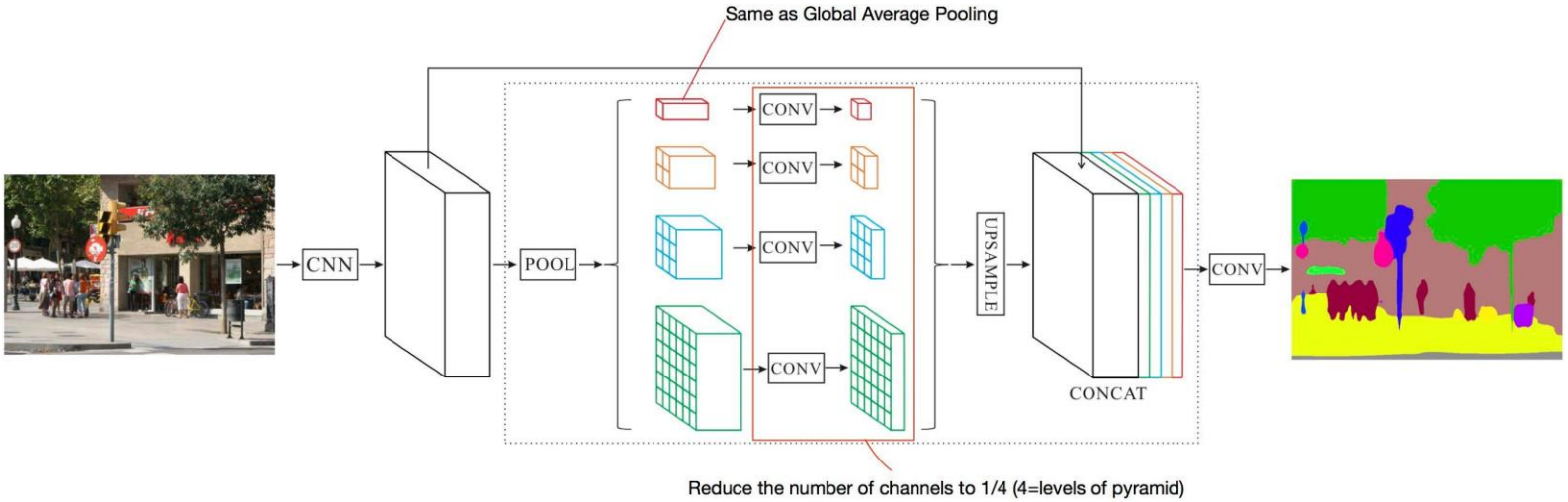
Pyramid Pooling Module

- **A hierarchical global prior**, containing information with **different scales** and **varying among different sub-regions**
- Pyramid Pooling Module for global scene prior constructed on the top of the final-layer-feature-map



Pyramid Pooling Module

- Use 1x1 conv to reduce the number of channels
- Then upsample (bilinear) them to the same size and concatenate all



ImageNet Scene Parsing Challenge 2016

- Dataset: ADE20K
  - 150 classes and 1,038 image-level labels
  - 20,000/2,000/3,000 pixel-level labels for train/val/test

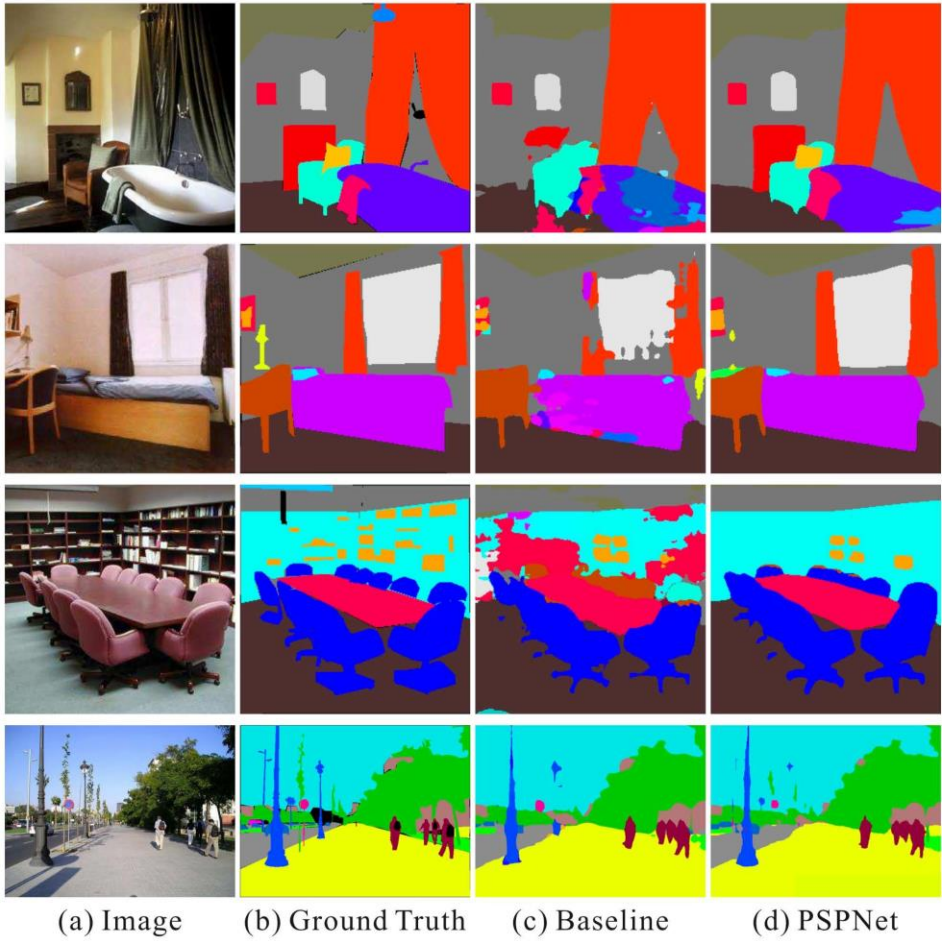


Figure 6. Visual improvements on ADE20K, PSPNet produces more accurate and detailed results.

Ablation Study for Pyramid Pooling Module

- **Average pooling** works better than max pooling in all settings
- **Pooling with pyramid parsing** outperforms that using global pooling
- With dimension reduction (DR; **reducing the number of channels after pyramid pooling**), the performance is further enhanced

Method	Mean IoU(%)	Pixel Acc.(%)
ResNet50-Baseline	37.23	78.01
ResNet50+B1+MAX	39.94	79.46
ResNet50+B1+AVE	40.07	79.52
ResNet50+B1236+MAX	40.18	79.45
ResNet50+B1236+AVE	41.07	79.97
ResNet50+B1236+MAX+DR	40.87	79.61
ResNet50+B1236+AVE+DR	**41.68**	**80.04**

Table 1. Investigation of PSPNet with different settings. Baseline is ResNet50-based FCN with dilated network. 'B1' and 'B1236' denote pooled feature maps of bin sizes $\{1 \times 1\}$ and $\{1 \times 1, 2 \times 2, 3 \times 3, 6 \times 6\}$ respectively. 'MAX' and 'AVE' represent max pooling and average pooling operations individually. 'DR' means that dimension reduction is taken after pooling. The results are tested on the validation set with the single-scale input.

Ablation Study for Auxiliary Loss

- Set the auxiliary loss weight α between 0 and 1 and compared the final results
- $\alpha = 0.4$ **yields the best performance**

Loss Weight α	Mean IoU(%)	Pixel Acc.(%)
ResNet50 (without AL)	35.82	77.07
ResNet50 (with $\alpha = 0.3$)	37.01	77.87
ResNet50 (with $\alpha = 0.4$)	**37.23**	**78.01**
ResNet50 (with $\alpha = 0.6$)	37.09	77.84
ResNet50 (with $\alpha = 0.9$)	36.99	77.87

Table 2. Setting an appropriate loss weight α in the auxiliary branch is important. 'AL' denotes the auxiliary loss. Baseline is ResNet50-based FCN with dilated network. Empirically, $\alpha = 0.4$ yields the best performance. The results are tested on the validation set with the single-scale input.

Ablation Study for the ResNet Part

Deeper is better

Method	Mean IoU(%)	Pixel Acc.(%)
PSPNet(50)	41.68	80.04
PSPNet(101)	41.96	80.64
PSPNet(152)	42.62	80.80
PSPNet(269)	**43.81**	**80.88**
PSPNet(50)+MS	42.78	80.76
PSPNet(101)+MS	43.29	81.39
PSPNet(152)+MS	43.51	81.38
PSPNet(269)+MS	**44.94**	**81.69**

Table 3. Deeper pre-trained model get higher performance. Number in the brackets refers to the depth of ResNet and 'MS' denotes multi-scale testing.

More Detailed Performance Analysis

Additional processing	Improvement (% in mIoU)
Data augmentation (DA)	+1.54
Auxiliary loss (AL)	+1.41
Pyramid pooling module (PSP)	+4.45
Use deeper ResNet (50 to 269)	+2.13
Multi-scale testing (MS)	+1.13

- For multi-scale testing, they create prediction at 6 different scales (0.5, 0.75, 1, 1.25, 1.5, and 1.75) and take average of them.

Method	Mean IoU(%)	Pixel Acc.(%)
FCN [26]	29.39	71.32
SegNet [2]	21.64	71.00
DilatedNet [40]	32.31	73.55
CascadeNet [43]	34.90	74.52
ResNet50-Baseline	34.28	76.35
ResNet50+DA	35.82	77.07
ResNet50+DA+AL	37.23	78.01
ResNet50+DA+AL+PSP	**41.68**	**80.04**
ResNet269+DA+AL+PSP	43.81	80.88
ResNet269+DA+AL+PSP+MS	**44.94**	**81.69**

Table 4. Detailed analysis of our proposed PSPNet with comparison with others. Our results are obtained on the validation set with the single-scale input except for the last row. Results of FCN, SegNet and DilatedNet are reported in [43]. 'DA' refers to data augmentation we performed, 'AL' denotes the auxiliary loss we added and 'PSP' represents the proposed PSPNet. 'MS' means that multi-scale testing is used.

Results on PASCAL VOC 2012

- Extended with Semantic Boundaries Dataset (SBD) [1], they used
  - 10582, 1449, and 1456 images for train/val/test
- **Mismatched relationship:** For "aeroplane" and "sky" in the second and third rows, PSPNet finds missing parts.
- **Confusing classes:** For "cows" in row one, our baseline model treats it as "horse" and "dog" while PSPNet corrects these errors
- **Conspicuous objects:** For "person", "bottle" and "plant" in following rows, PSPNet performs well on these small-size-object classes in the images compared to the baseline model

Figure 7. Visual improvements on PASCAL VOC 2012 data. PSP-Net produces more accurate and detailed results.

[1] "Semantic Contours from Inverse Detectors", ICCV 2011, http://home.bharathh.info/pubs/codes/SBD/download.html

Results on PASCAL VOC 2012

- Comparing PSPNet with previous best-performing methods on the testing set based on two settings, i.e., with or without pre-training on MS-COCO dataset

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU
FCN [26]	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
Zoom-out [28]	85.6	37.3	83.2	62.5	66.0	85.1	80.7	84.9	27.2	73.2	57.5	78.1	79.2	81.1	77.1	53.6	74.0	49.2	71.7	63.3	69.6
DeepLab [3]	84.4	54.5	81.5	63.6	65.9	85.1	79.1	83.4	30.7	74.1	59.8	79.0	76.1	83.2	80.8	59.7	82.2	50.4	73.1	63.7	71.6
CRF-RNN [41]	87.5	39.0	79.7	64.2	68.3	87.6	80.8	84.4	30.4	78.2	60.4	80.5	77.8	83.1	80.6	59.5	82.8	47.8	78.3	67.1	72.0
DeconvNet [30]	89.9	39.3	79.7	63.9	68.2	87.4	81.2	86.1	28.5	77.0	62.0	79.0	80.3	83.6	80.2	58.8	83.4	54.3	80.7	65.0	72.5
GCRF [36]	85.2	43.9	83.3	65.2	68.3	89.0	82.7	85.3	31.1	79.5	63.3	80.5	79.3	85.5	81.0	60.5	85.5	52.0	77.3	65.1	73.2
DPN [25]	87.7	59.4	78.4	64.9	70.3	89.3	83.5	86.1	31.7	79.9	62.6	81.9	80.0	83.5	82.3	60.5	83.2	53.4	77.9	65.0	74.1
Piecewise [20]	90.6	37.6	80.0	67.8	74.4	92.0	85.2	86.2	39.1	81.2	58.9	83.8	83.9	84.3	84.8	62.1	83.2	58.2	80.8	72.3	75.3
PSPNet	**91.8**	**71.9**	**94.7**	**71.2**	**75.8**	**95.2**	**89.9**	**95.9**	**39.3**	**90.7**	**71.7**	**90.5**	**94.5**	**88.8**	**89.6**	**72.8**	**89.6**	**64.0**	**85.1**	**76.3**	**82.6**
CRF-RNN† [41]	90.4	55.3	88.7	68.4	69.8	88.3	82.4	85.1	32.6	78.5	64.4	79.6	81.9	86.4	81.8	58.6	82.4	53.5	77.4	70.1	74.7
BoxSup† [7]	89.8	38.0	89.2	68.9	68.0	89.6	83.0	87.7	34.4	83.6	67.1	81.5	83.7	85.2	83.5	58.6	84.9	55.8	81.2	70.7	75.2
Dilation8† [40]	91.7	39.6	87.8	63.1	71.8	89.7	82.9	89.8	37.2	84.0	63.0	83.3	89.0	83.8	85.1	56.8	87.6	56.0	80.2	64.7	75.3
DPN† [25]	89.0	61.6	87.7	66.8	74.7	91.2	84.3	87.6	36.5	86.3	66.1	84.4	87.8	85.6	85.4	63.6	87.3	61.3	79.4	66.4	77.5
Piecewise† [20]	94.1	40.7	84.1	67.8	75.9	93.4	84.3	88.4	42.5	86.4	64.7	85.4	89.0	85.8	86.0	67.5	90.2	63.8	80.9	73.0	78.0
FCRNs† [38]	91.9	48.1	93.4	69.3	75.5	94.2	87.5	92.8	36.7	86.9	65.2	89.1	90.2	86.5	87.2	64.6	90.1	59.7	85.5	72.7	79.1
LRR† [9]	92.4	45.1	94.6	65.2	75.8	**95.1**	89.1	92.3	39.0	85.7	70.4	88.6	89.4	88.6	86.6	65.8	86.2	57.4	85.7	77.3	79.3
DeepLab† [4]	92.6	60.4	91.6	63.4	76.3	95.0	88.4	92.6	32.7	88.5	67.6	89.6	92.1	87.0	87.4	63.3	88.3	60.0	86.8	74.5	79.7
PSPNet†	**95.8**	**72.7**	**95.0**	**78.9**	**84.4**	94.7	**92.0**	**95.7**	**43.1**	**91.0**	**80.3**	**91.3**	**96.3**	**92.3**	**90.1**	**71.5**	**94.4**	**66.9**	**88.8**	**82.0**	**85.4**

Table 6. Per-class results on PASCAL VOC 2012 testing set. Methods pre-trained on MS-COCO are marked with '†'.

Results on Cityscapes

- Cityscapes dataset consits of 2975, 500, and 1525 train/val/tests images (19 classes)
- 20000 coarsely annotated images are available (in the table below, ‡ means it's used)

Method	IoU cla.	iIoU cla.	IoU cat.	iIoU cat.
CRF-RNN [41]	62.5	34.4	82.7	66.0
FCN [26]	65.3	41.7	85.7	70.1
SiCNN [16]	66.3	44.9	85.0	71.2
DPN [25]	66.8	39.1	86.0	69.1
Dilation10 [40]	67.1	42.0	86.5	71.1
LRR [9]	69.7	48.0	88.2	74.7
DeepLab [4]	70.4	42.6	86.4	67.7
Piecewise [20]	71.6	51.7	87.3	74.1
PSPNet	**78.4**	**56.7**	**90.6**	**78.6**
LRR‡ [9]	71.8	47.9	88.4	73.9
PSPNet‡	**80.2**	**58.1**	**90.6**	**78.2**

Table 7. Results on Cityscapes testing set. Methods trained using both fine and coarse data are marked with '‡'.

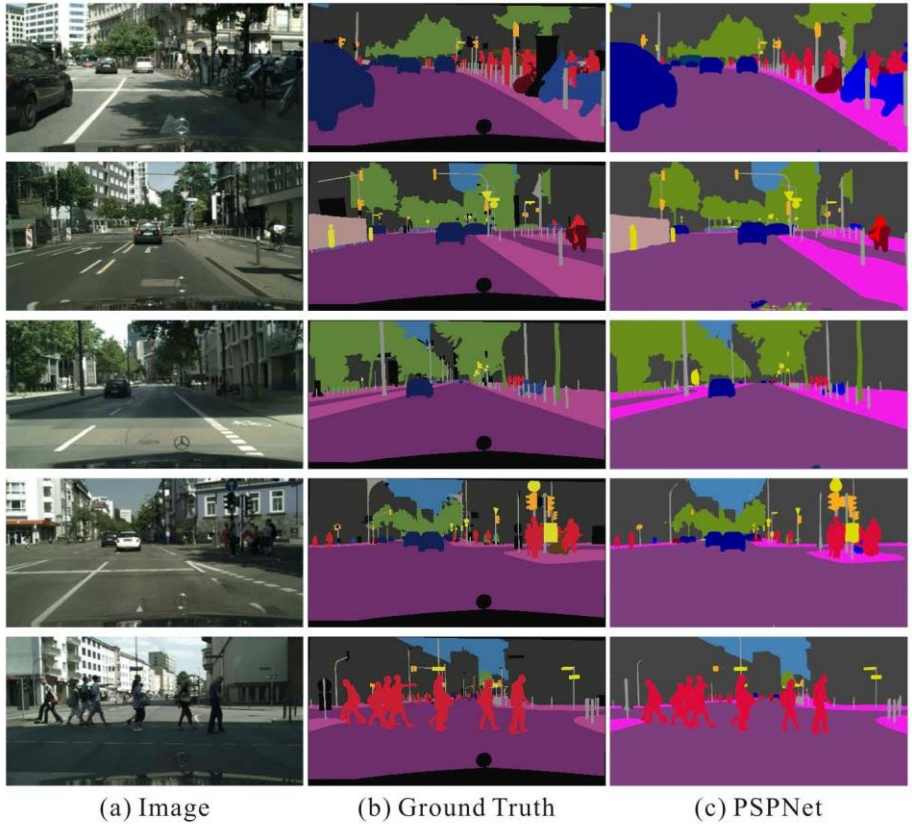


Figure 8. Examples of PSPNet results on Cityscapes dataset.

SETR: Segmentation with Transformers

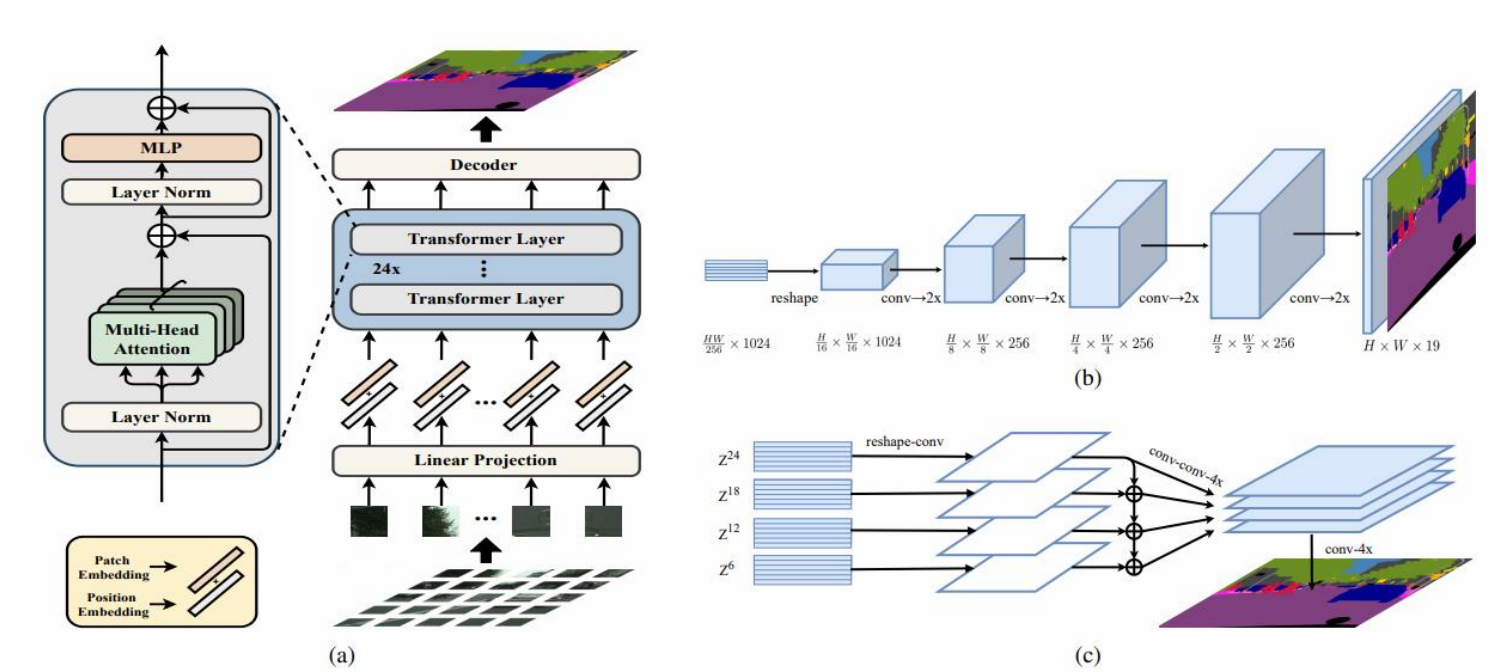


Figure 1. **Schematic illustration of the proposed *SEgmentation TRansformer* (SETR)** (a). We first split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. To perform pixel-wise segmentation, we introduce different decoder designs: (b) progressive upsampling (resulting in a variant called SETR-*PUP*); and (c) multi-level feature aggregation (a variant called SETR-*MLA*).

The End