Aprendizado Profundo (Deep Learning)

Semantic Segmentation

Dario Oliveira (dario.oliveira@fgv.br)

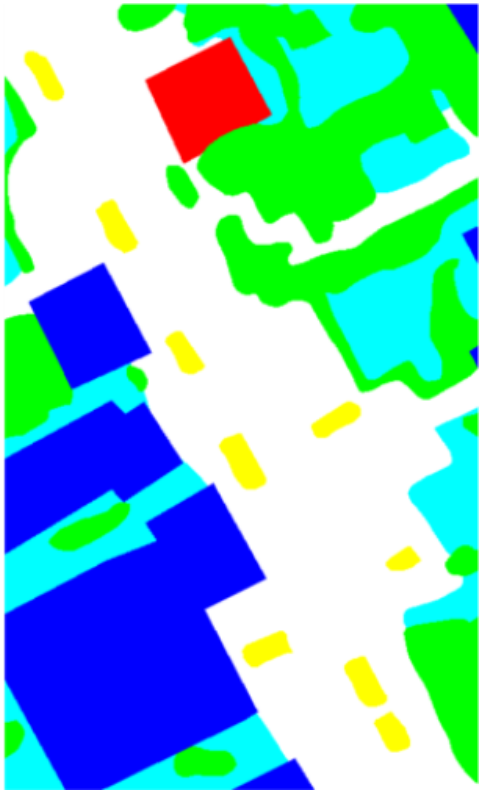
Overview

- Introduction
- Metrics
- Fully Convolutional Networks
  - SegNet (pool indices)
  - U-Net (skip connections)
  - ResUNet
  - DeepLabv1 & DeepLabv2 (Atrous conv. & ASPP)
  - PARSENet & PSPNet (Image Pooling – Parse Pooling)
  - DeepLabv3 & DeepLabv3+
- Loss Functions
- Practical Remark

Main Application Groups

Image Classification

Semantic Segmentation



Detection/ Localization

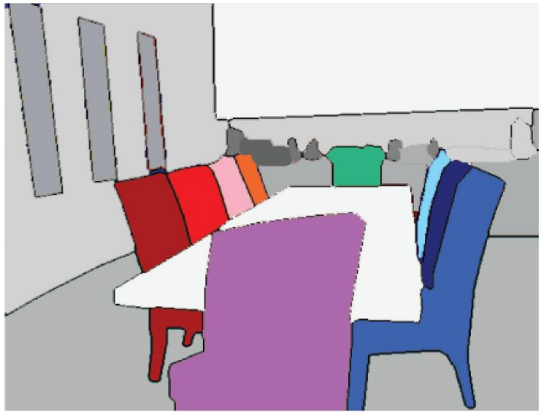
Semantic vs Instance Segmentation

Semantic Segmentation: identifies the object category of each pixel → labels are class aware.

Instance Segmentation: identifies each object instance of each pixel → labels are instance-aware.

Input Image

Semantic Segmentation

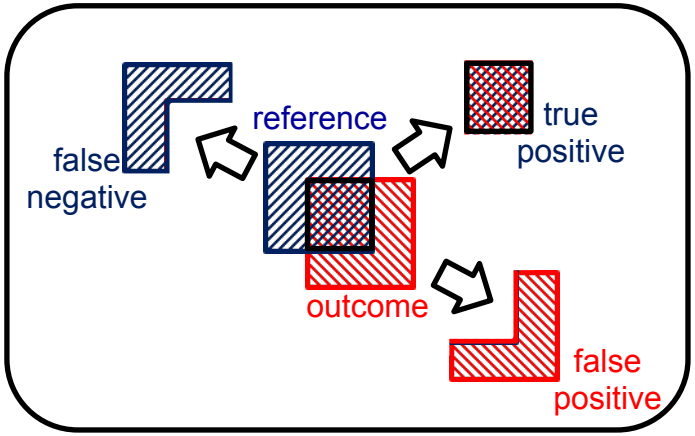


Semantic Instance Segmentation

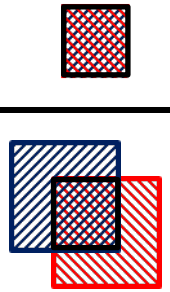
Overview

- Introduction
- Metrics
- Fully Convolutional Networks
  - SegNet (pool indices)
  - U-Net (skip connections)
  - ResUNet
  - DeepLabv1 & DeepLabv2 (Atrous conv. & ASPP)
  - PARSENet & PSPNet (Image Pooling – Parse Pooling)
  - DeepLabv3 & DeepLabv3+
- Loss Functions
- Practical Remark

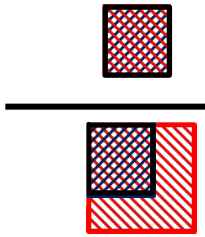
Metrics for SS



$$\textbf{Intersection over Union } (IoU) =$$



$$\textbf{Precision} = \frac{true\ positive}{true\ positive + false\ positive} =$$

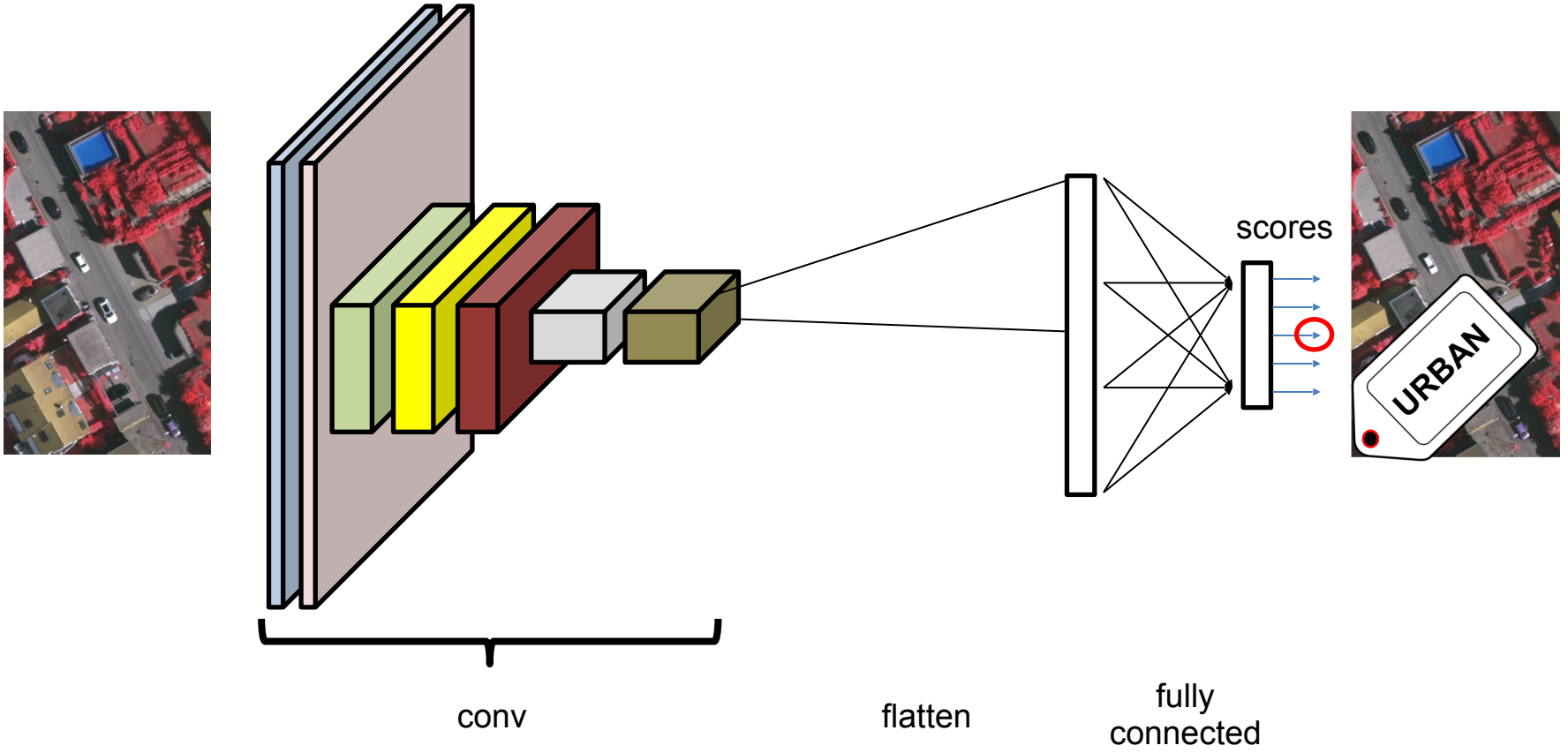


$$\textbf{Average Precision } (AP) = precision\ over\ all\ classes$$

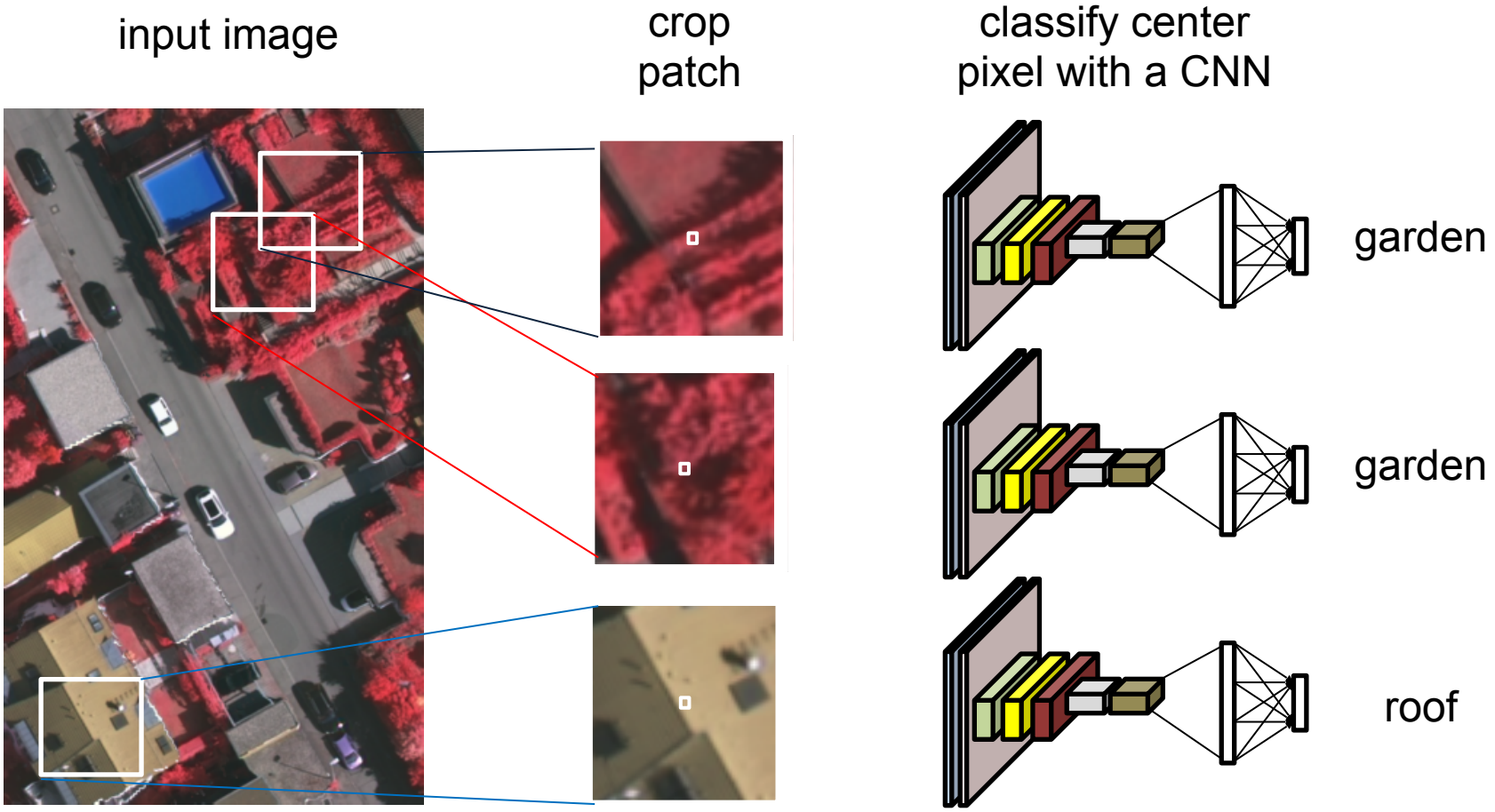
Overview

- Introduction
- Metrics
- Fully Convolutional Networks
  - SegNet (pool indices)
  - U-Net (skip connections)
  - ResUNet
  - DeepLabv1 & DeepLabv2 (Atrous conv. & ASPP)
  - PARSENet & PSPNet (Image Pooling – Parse Pooling)
  - DeepLabv3 & DeepLabv3+
- Loss Functions
- Practical Remark

Typical ConvNet for Image Classification



1st idea for SS: sliding window



Redundant operations due to overlap. Inefficient!

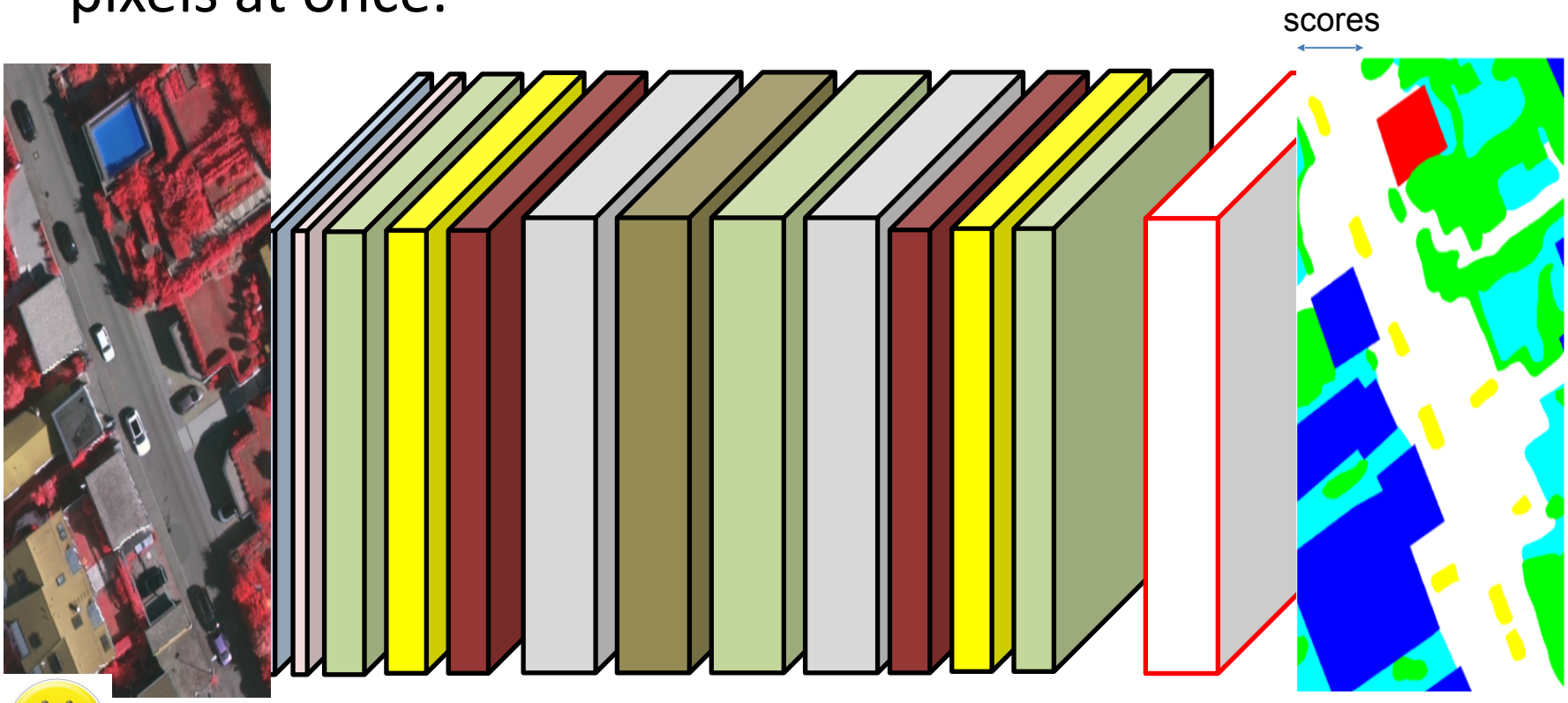
Spatial accuracy

Example:

- The CNN scores high for “car” and “road“ for both patches on the right.
- In consequence, CNN patch classification oversmooths objects boundaries.
- FCN, on the contrary, learns class specific structures contained in each patch.

2nd idea for SS: Fully Convolutional (a)

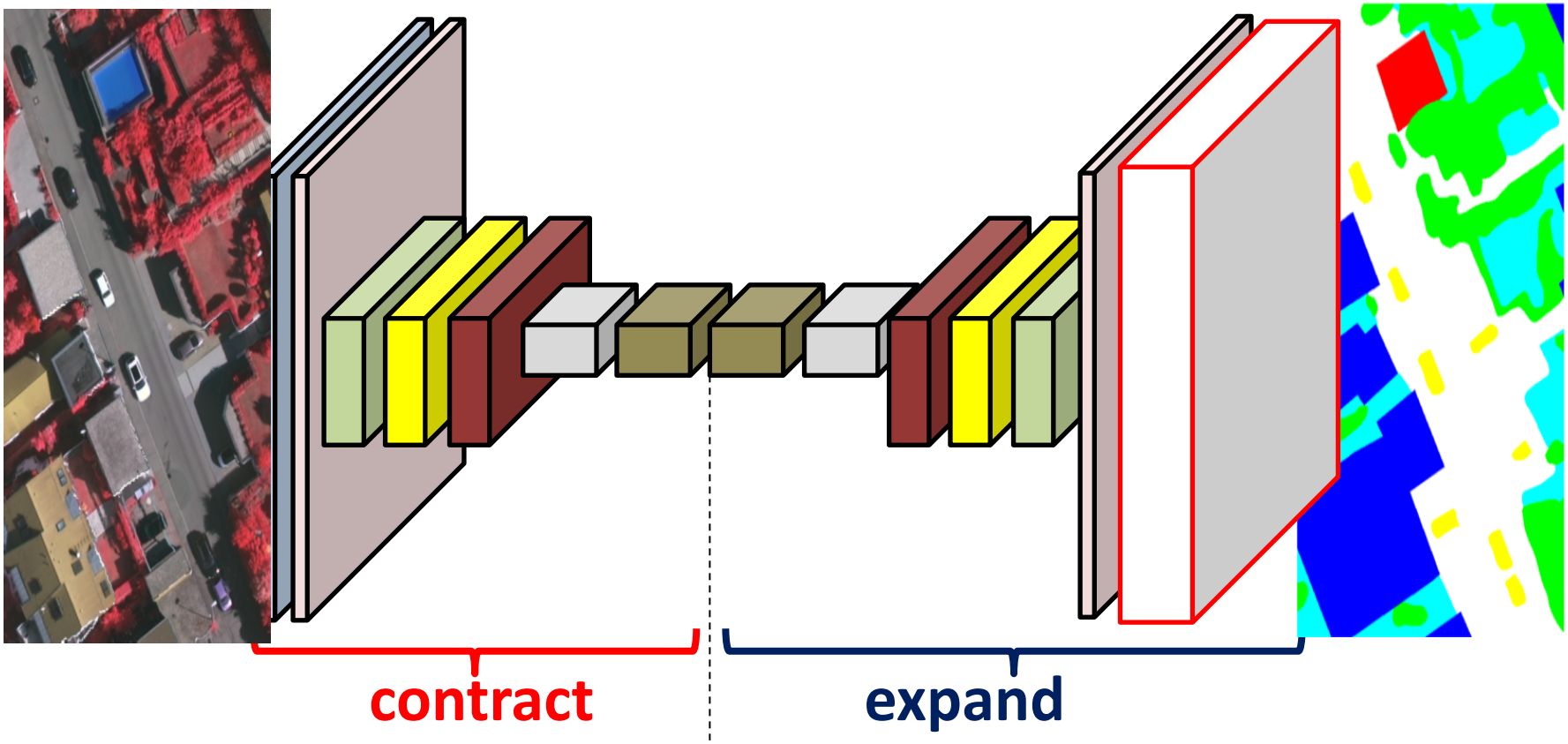
A bunch of layers at input resolution to classify all pixels at once.



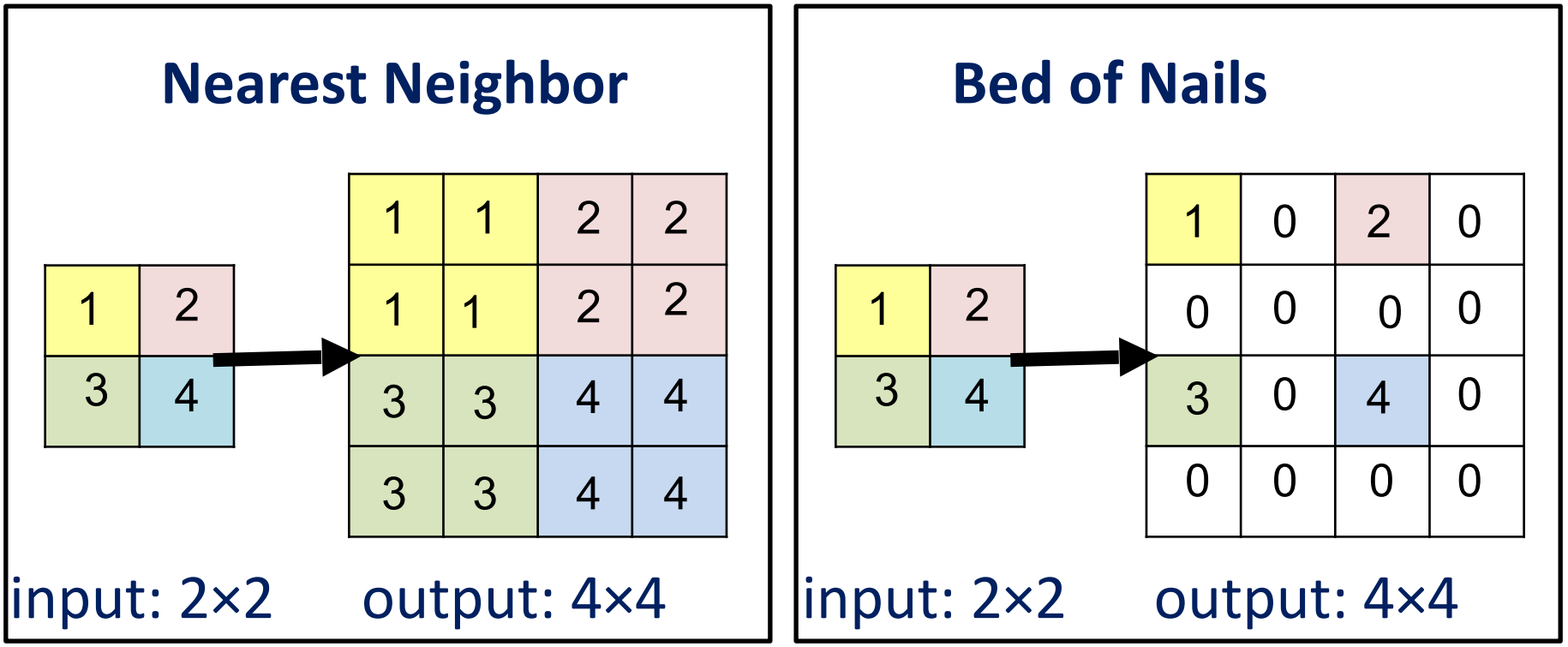
Convolution at original resolution expensive!

3rd idea for SS: Fully Convolutional (b)

A bunch of layers with **downsampling** and **upsampling** inside the network.



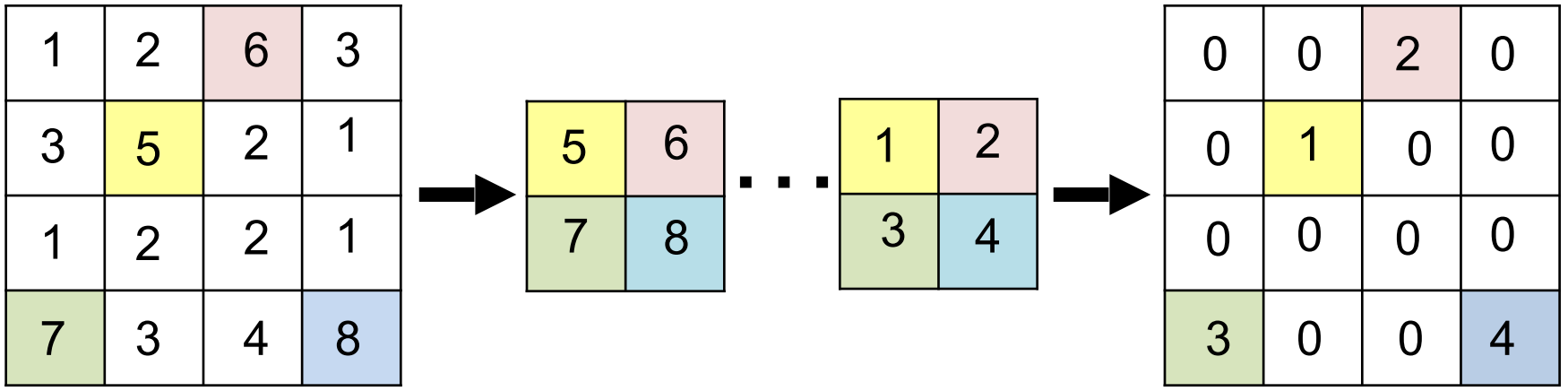
Upsampling: Unpooling



Upsampling: Max Unpooling

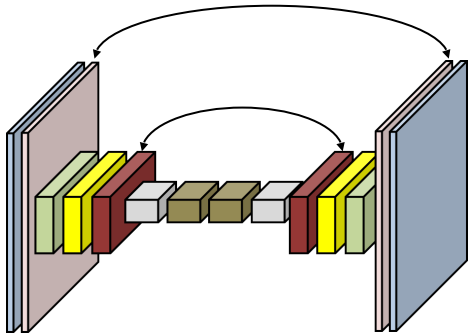
Max Pooling
memorizes where the max came from

Max Unpooling
puts at the location where the max came from



input: 4×4 output: 2×2 input: 2×2 output: 4×4

corresponding pairs of down- and upsampling layers

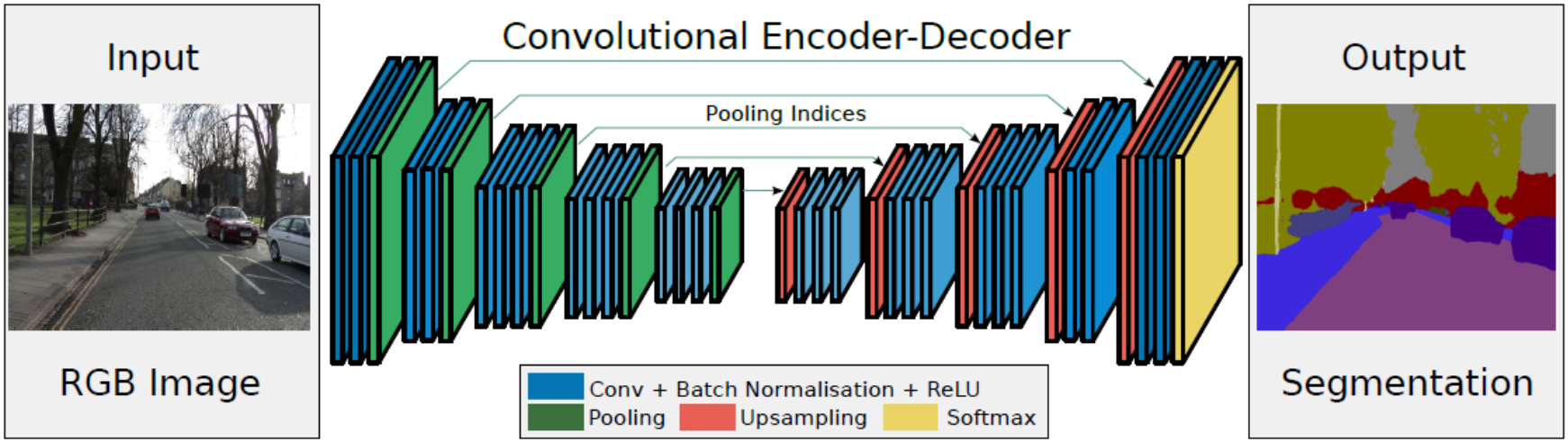


Overview

- Introduction
- Metrics
- Fully Convolutional Networks
  - SegNet (pool indices)
  - U-Net (skip connections)
  - ResUNet
  - DeepLabv1 & DeepLabv2 (Atrous conv. & ASPP)
  - PARSENet & PSPNet (Image Pooling – Parse Pooling)
  - DeepLabv3 & DeepLabv3+
- Loss Functions
- Practical Remark

SegNet

SegNet uses Max Pooling indices to upsample:

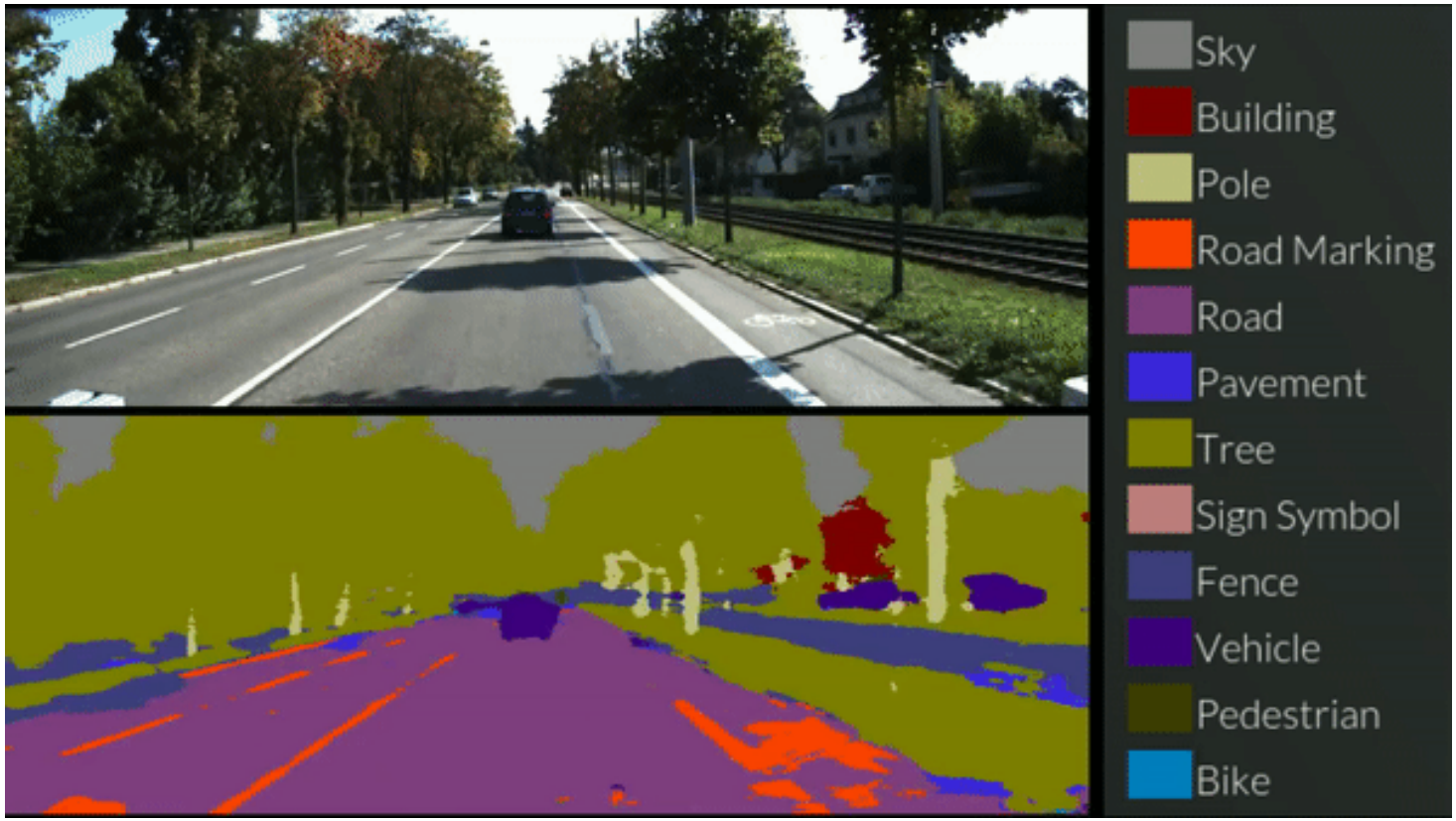


Vijay Badrinarayanan, Alex Kendall and Roberto Cipolla "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation." PAMI, 2017. (arxiv .pdf) (poster)

Picture from Sik-Ho Tsang available at https://towardsdatascience.com/review-segnet-semantic-segmentation-e66f2e30fb96:.

SegNet

SegNet demo: Road Scene Segmentation (1)



demo by authors at : https://towardsdatascience.com/review-segnet-semantic-segmentation-e66f2e30fb96

SegNet

SegNet demo: Road Scene Segmentation (2)

Another demo by authors at Youtube: https://www.youtube.com/watch?v=CxanE_W46ts

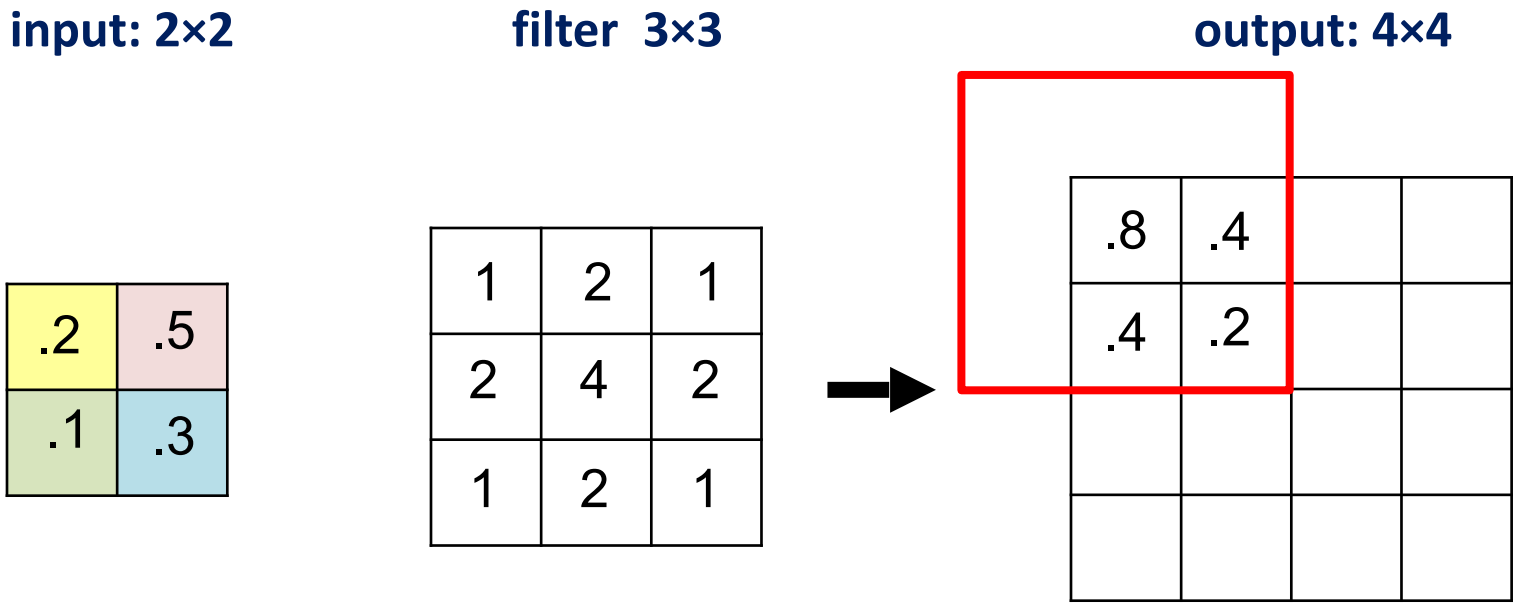
SegNet

SegNet demo: Random image

Clik on the image to test SegNet on a random image

Learnable Upsampling: Transpose Convolution

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



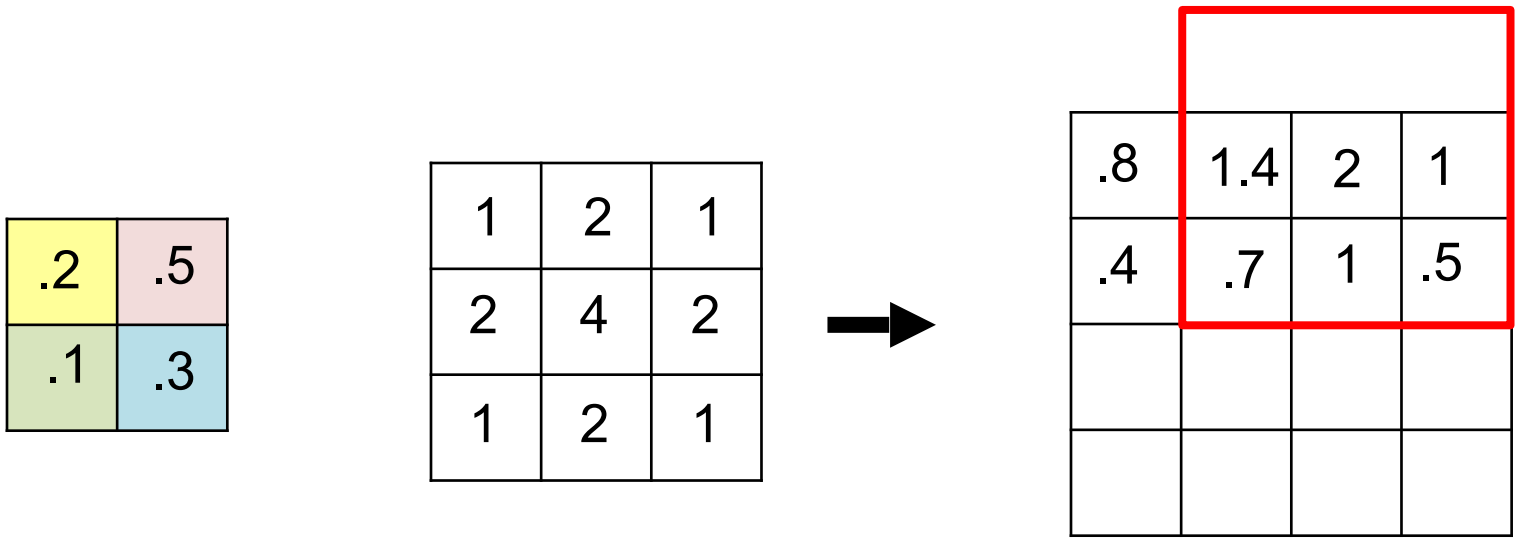
Filter moves 2 pixels in the output for every pixel in the input.

Stride gives ratio between movement in output and input.

Learnable Upsampling: Transpose Convolution

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

input: 2×2 **filter 3×3** **output: 4×4**

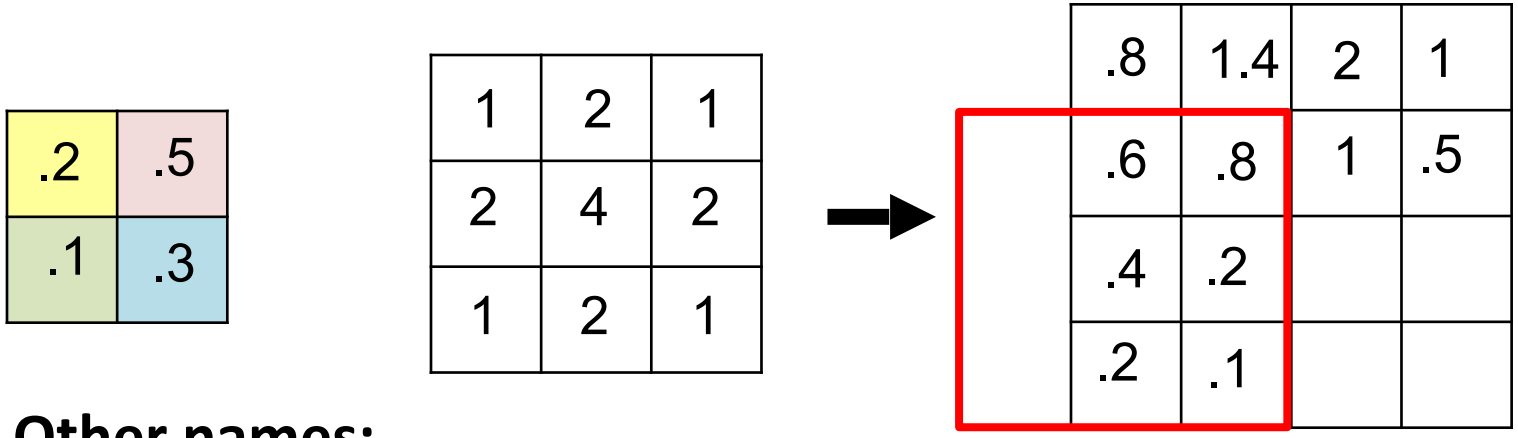


Output contains copies of the filter weighted by the input, summing up at overlaps in the output.

Learnable Upsampling: Transpose Convolution

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

input: 2×2 **filter 3×3** **output: 4×4**



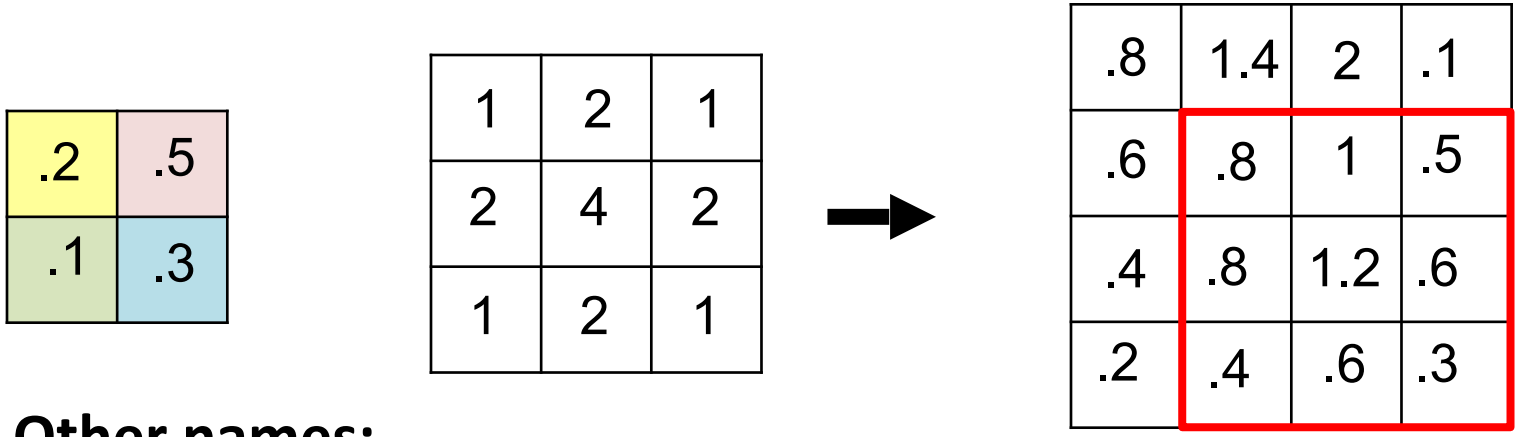
Other names:

- deconvolution,
- upconvolution,
- fractionally strided convolutionl,
- backward strided convolution.

Learnable Upsampling: Transpose Convolution

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

input: 2×2 **filter 3×3** **output: 4×4**



Other names:

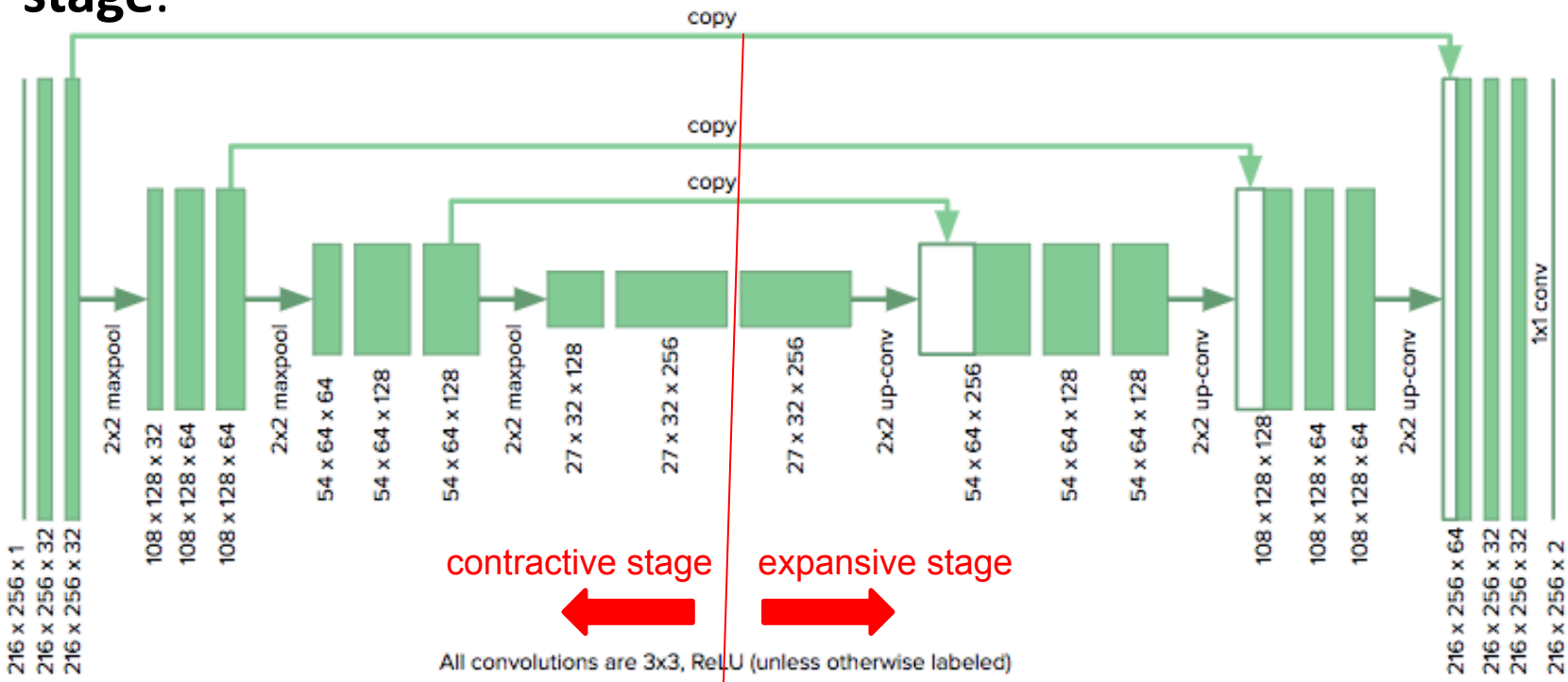
- deconvolution,
- upconvolution,
- fractionally strided convolution,
- backward strided convolution.

Overview

- Introduction
- Metrics
- Fully Convolutional Networks
  - SegNet (pool indices)
  - U-Net (skip connections)
  - ResUNet
  - DeepLabv1 & DeepLabv2 (Atrous conv. & ASPP)
  - PARSENet & PSPNet (Image Pooling – Parse Pooling)
  - DeepLabv3 & DeepLabv3+
- Loss Functions
- Practical Remark

FCN Example: U-net

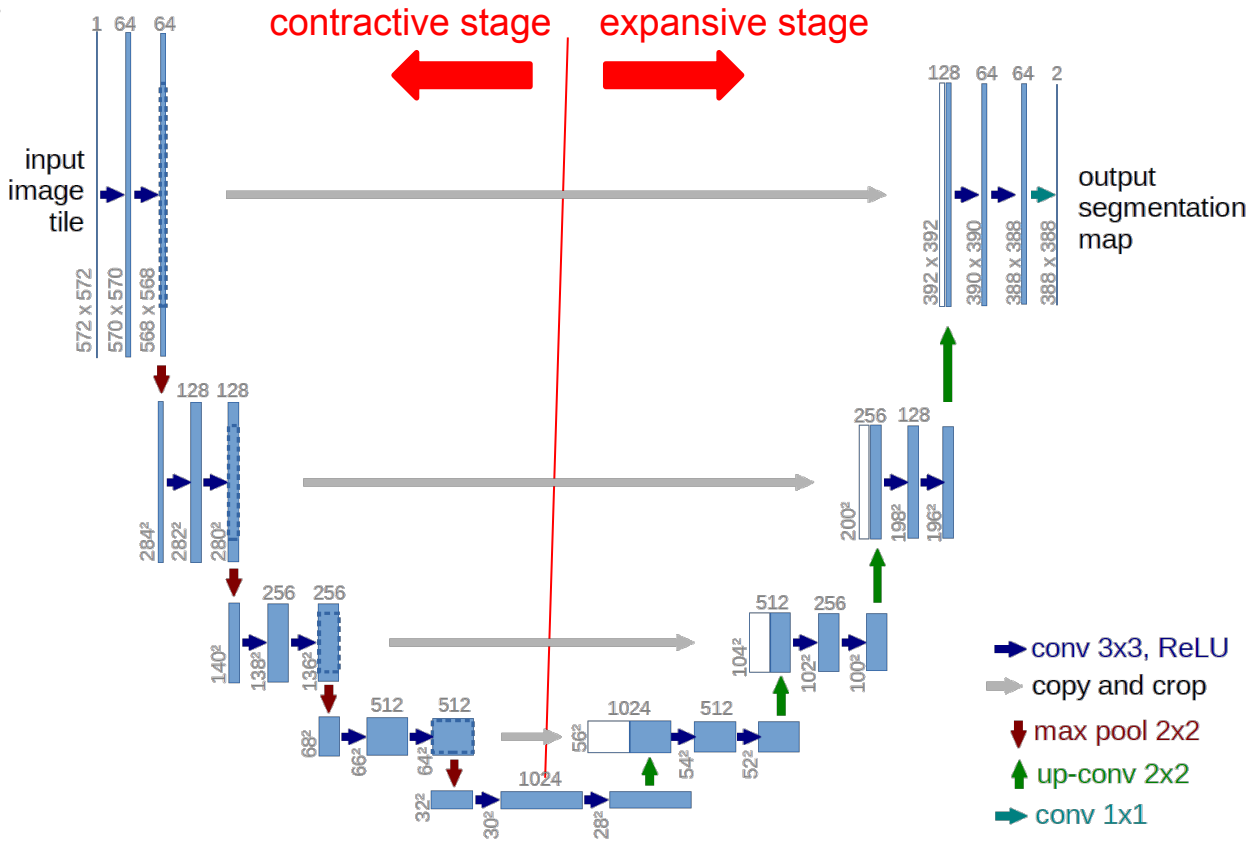
To improve spatial accuracy, **output of corresponding layer** in **contractive stage** is **appended to the inputs of the expansive stage**.



Picture from: Ronneberger O., Fischer P., Brox T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham.

FCN Example: U-net

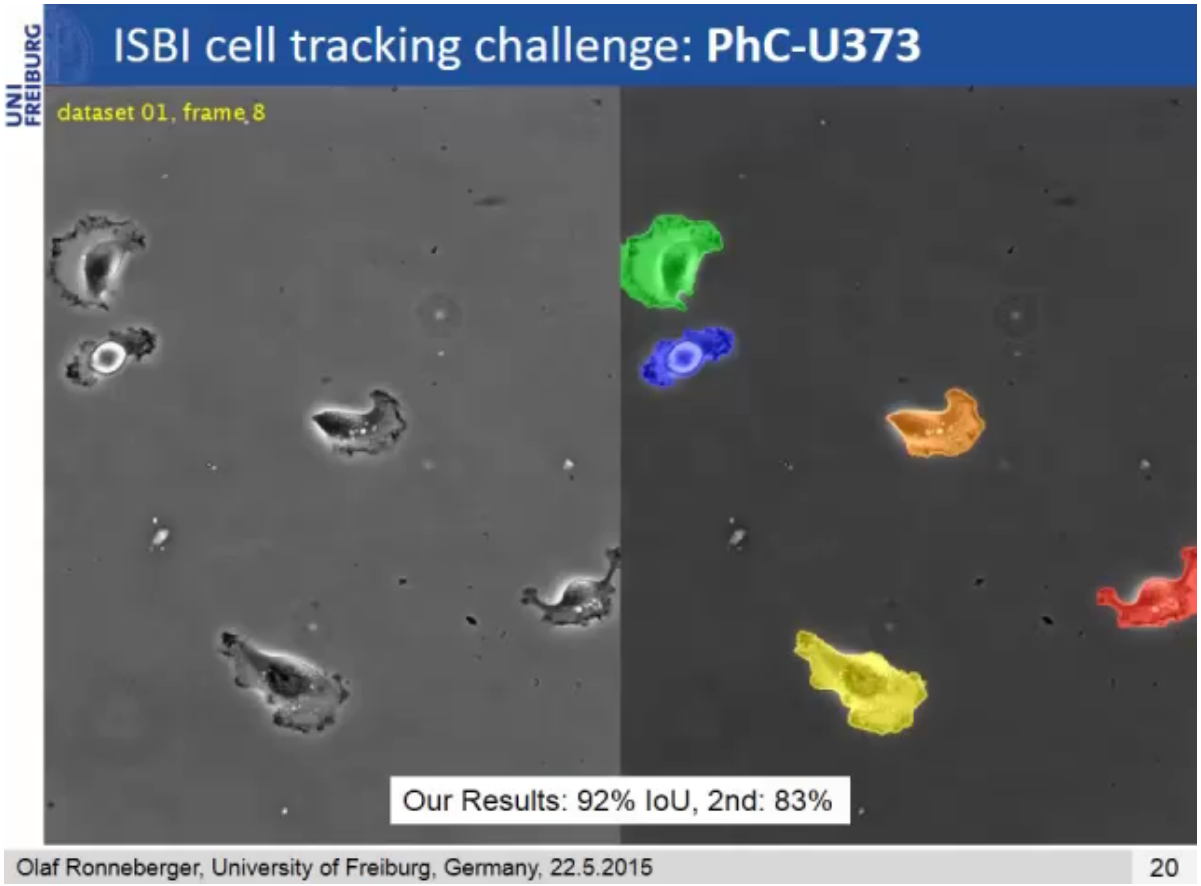
To improve spatial accuracy, **output of corresponding layer** in **contractive stage** is **appended to the inputs of the expansive stage**.



Picture from Ronneberger, O., et al., (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". arXiv:1505.04597:.

U-Net

U-Net demo: cell tracking



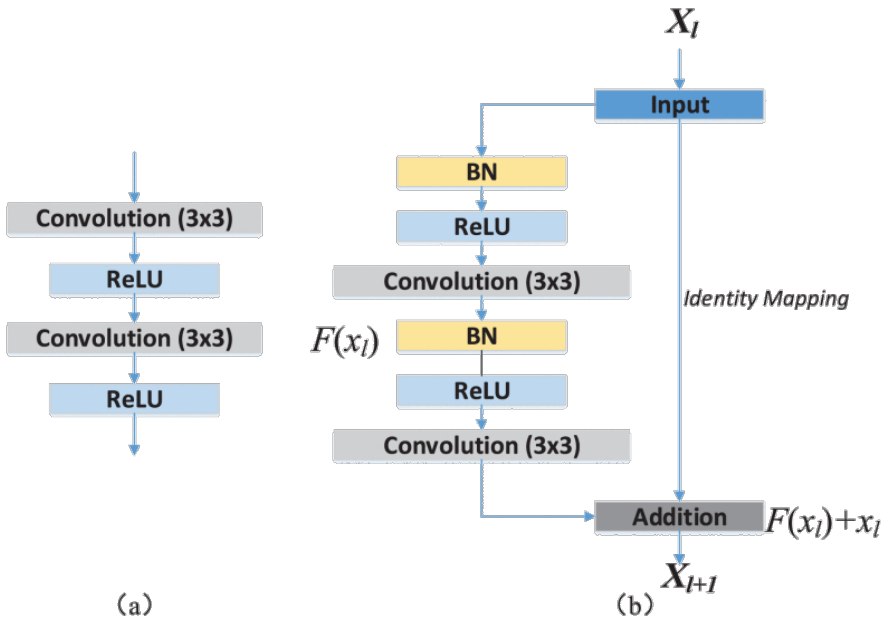
Full demo available at https://www.youtube.com/watch?v=81AvQQnpG4Q

Overview

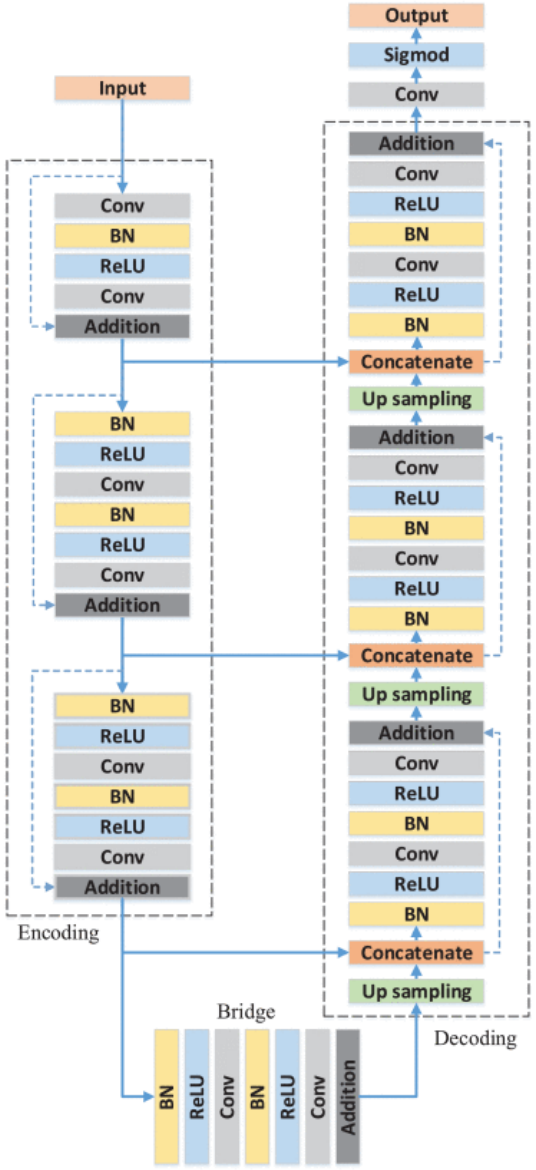
- Introduction
- Metrics
- Fully Convolutional Networks
  - SegNet (pool indices)
  - U-Net (skip connections)
  - ResUNet
  - DeepLabv1 & DeepLabv2 (Atrous conv. & ASPP)
  - PARSENet & PSPNet (Image Pooling – Parse Pooling)
  - DeepLabv3 & DeepLabv3+
- Loss Functions
- Practical Remark

ResUNet

Replaces UNet neural units by residual blocks to ease training



Building blocks of neural networks. (a) Plain neural unit used in U-Net. (b) Residual unit with identity mapping used in the proposed ResUnet.



ResUNet Architecture

Picture from Zhang et al. (2018) "Road Extraction by Deep Residual U-Net"

ResUNet (sample results)

Example results on the test set of Massachusetts roads data set. (a) Input image. (b) Ground truth. (c) Saito et al. [5]. (d) U-Net [24]. (e) Proposed ResUnet. Zoomed-in view to see more details.

Source"Zhang et al. (2018) Road Extraction by Deep Residual U-Net"

Overview

- Introduction
- Metrics
- Fully Convolutional Networks
  - SegNet (pool indices)
  - U-Net (skip connections)
  - ResUNet
  - DeepLabv1 & DeepLabv2 (Atrous conv. & ASPP)
  - PARSENet & PSPNet (Image Pooling – Parse Pooling)
  - DeepLabv3 & DeepLabv3+
- Loss Functions
- Practical Remark

Importance of Context/Scale

What is in the picture?

Importance of Context/Scale

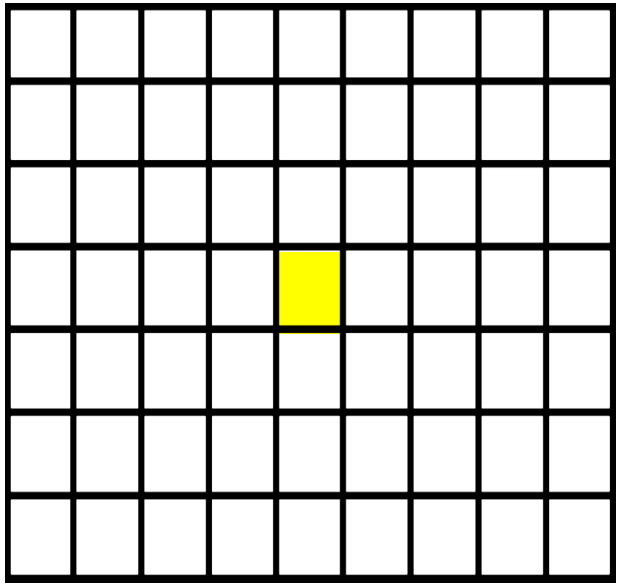
What is in the picture?

Source: http://bestpaintingsforsale.com/painting/lincoln_in_dali_vision-4020.html

Importance of Context/Scale

Larger filters capture larger context…

but involve more parameters/computation.

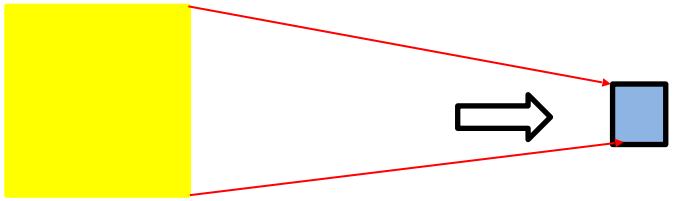


Importance of Scale

Larger filters capture larger context…

but involve more parameters/computation

3×3 filter



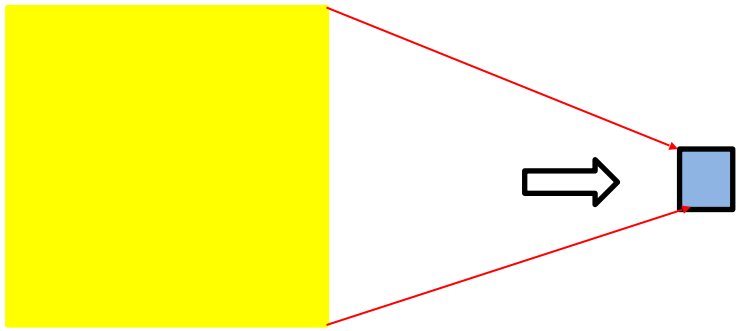
Receptive field = 3×3

Importance of Scale

Larger filters capture larger context…

but involve more parameters/computation

5×5 filter

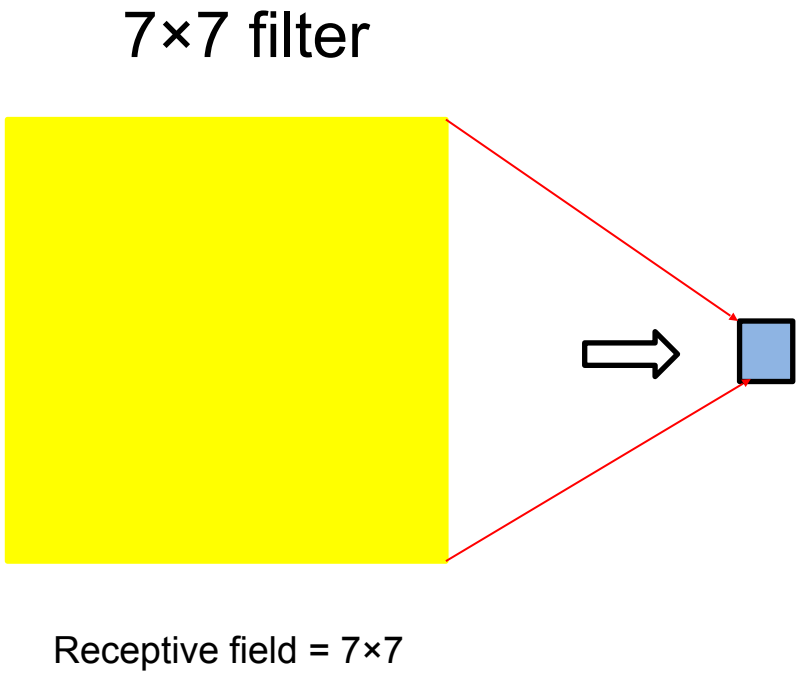


Receptive field = 5×5

Importance of Scale

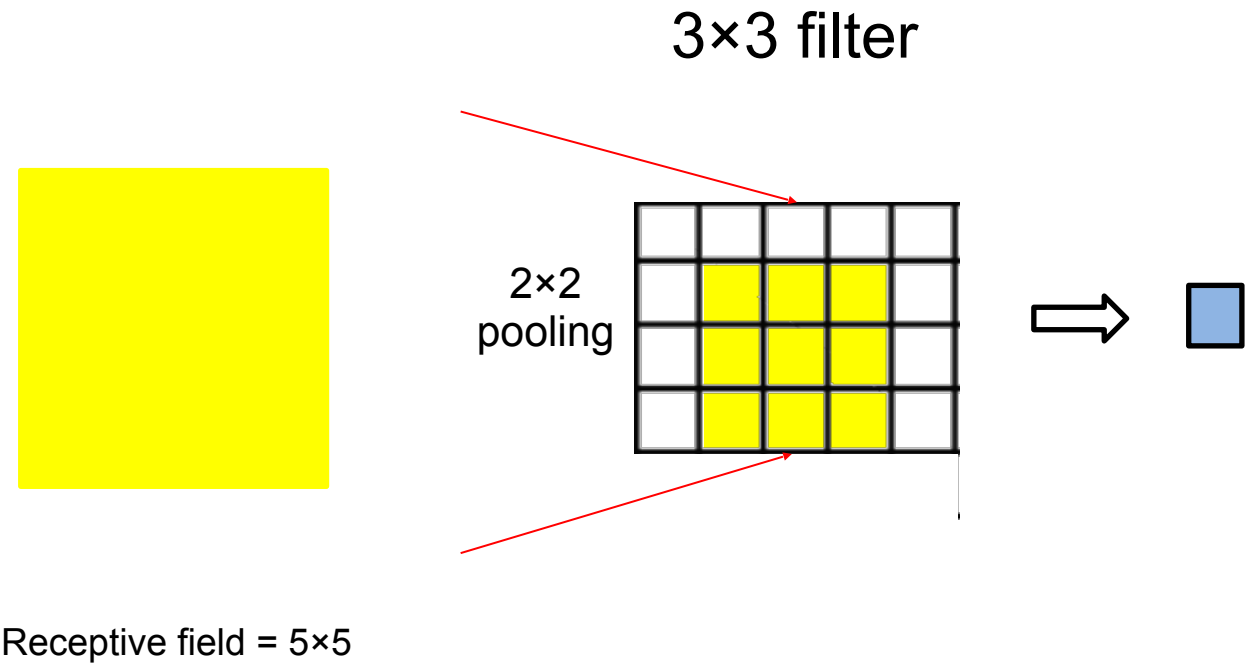
Larger filters capture larger context…

but involve more parameters/computation



Importance of Scale

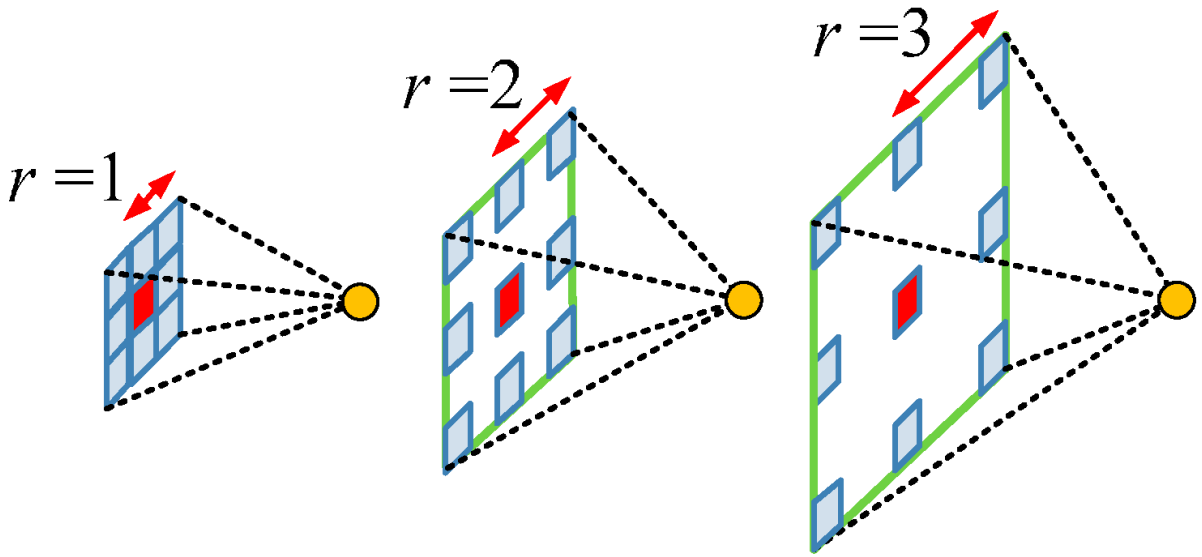
Pooling helps increasing the receptive field without implying more parameters (weights) to be learned.



However, it looses spatial information (resolution)!

Atrous Convolution

Equivalent to convolving the input $\boldsymbol{x}$ with upsampled filter produced by inserting r-1 zeros between consecutive filter values along each dimension.



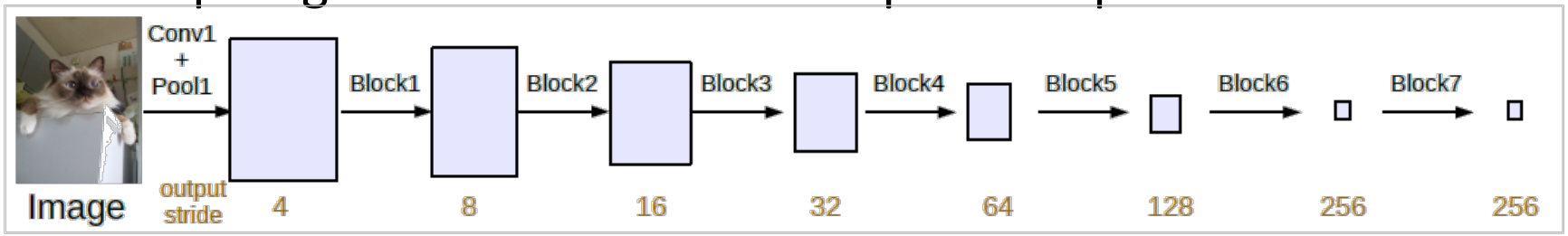
$$\boldsymbol{y}(\boldsymbol{i}) = \sum_{\boldsymbol{k}} \boldsymbol{x}\left(\boldsymbol{i} + r\boldsymbol{k}\right)\boldsymbol{w}\left(\boldsymbol{k}\right)$$

Atrous Convolution

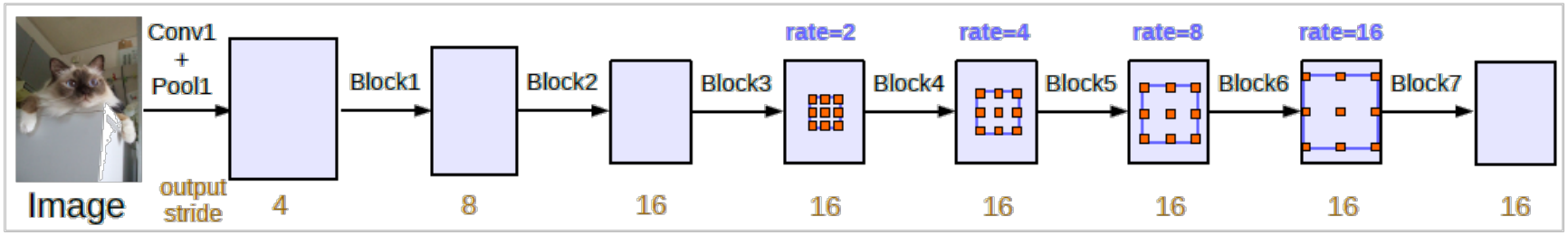
Without atrous conv: stride and pooling reduce feature map and loose location/spatial information.

With atrous conv: keep stride constant with larger receptive field without increasing the number of parameters.

It keeps high resolution feature maps in deep blocks.



(a) Going deeper without atrous convolution.

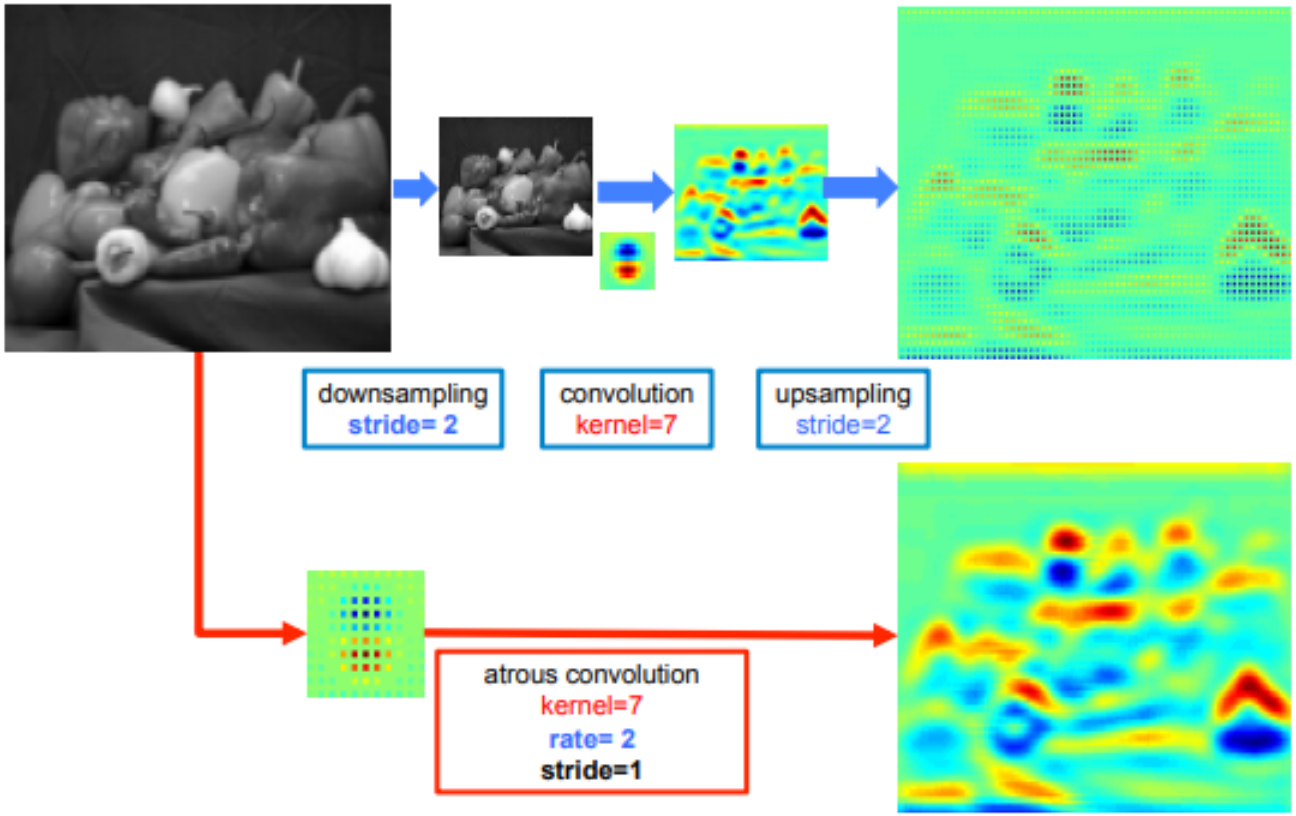


(b) Going deeper with atrous convolution. Atrous convolution with $rate > 1$ is applied after block3 when $output\_stride = 16$.

Source Chen, et al., 2017 available at https://arxiv.org/pdf/1606.00915.pdf

Atrous Convolution

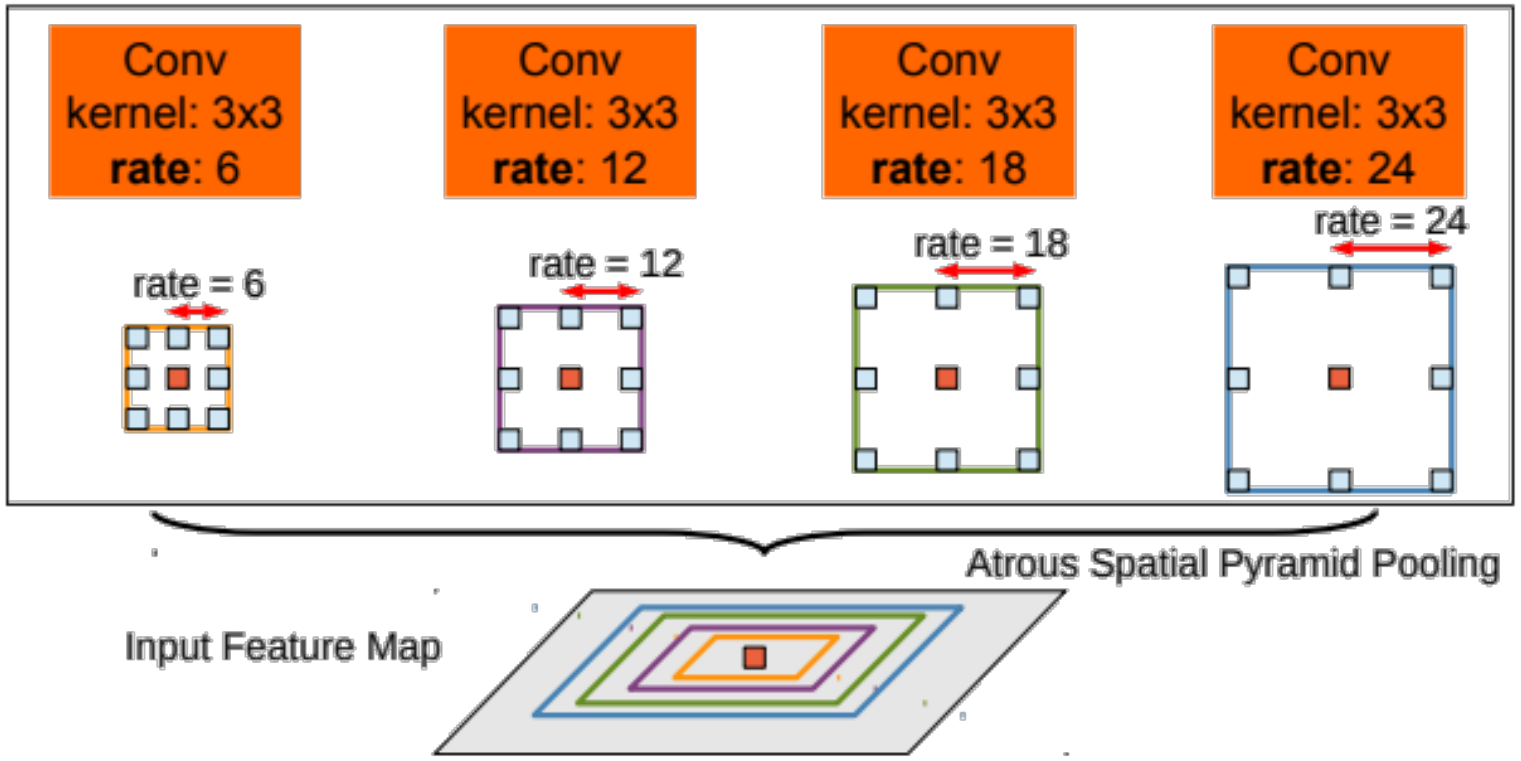
Comparison with and without atrous conv.



Source Chen, et al., 2017 available at https://arxiv.org/pdf/1606.00915.pdf

Atrous Spatial Pyramid Pooling (ASPP)

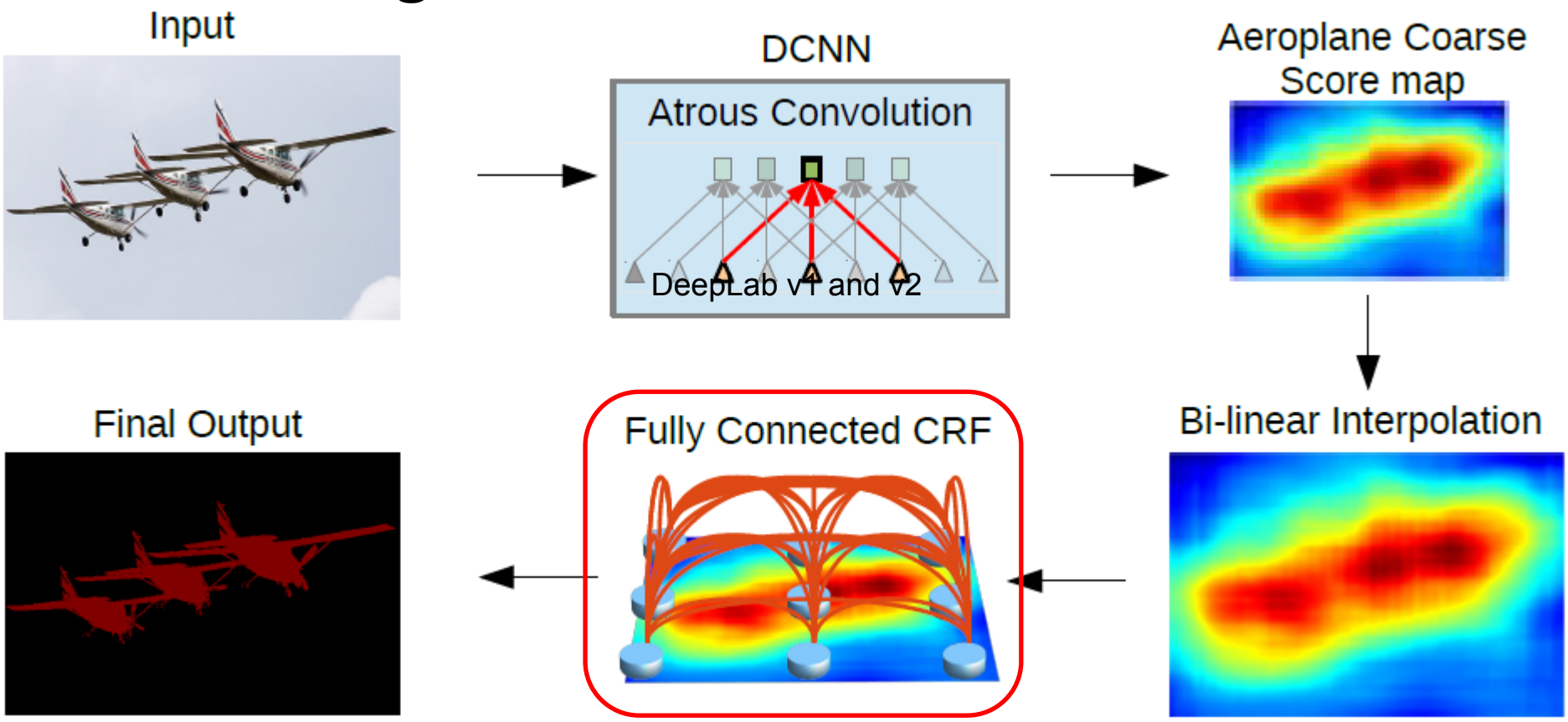
- Also introduced in **DeepLab v2**
- Parallel rather than cascade atrous conv captures context at multiple scales.



Source Chen, et al., 2017 available at https://arxiv.org/pdf/1606.00915.pdf

DeepLab v1 and v2

Processing chain



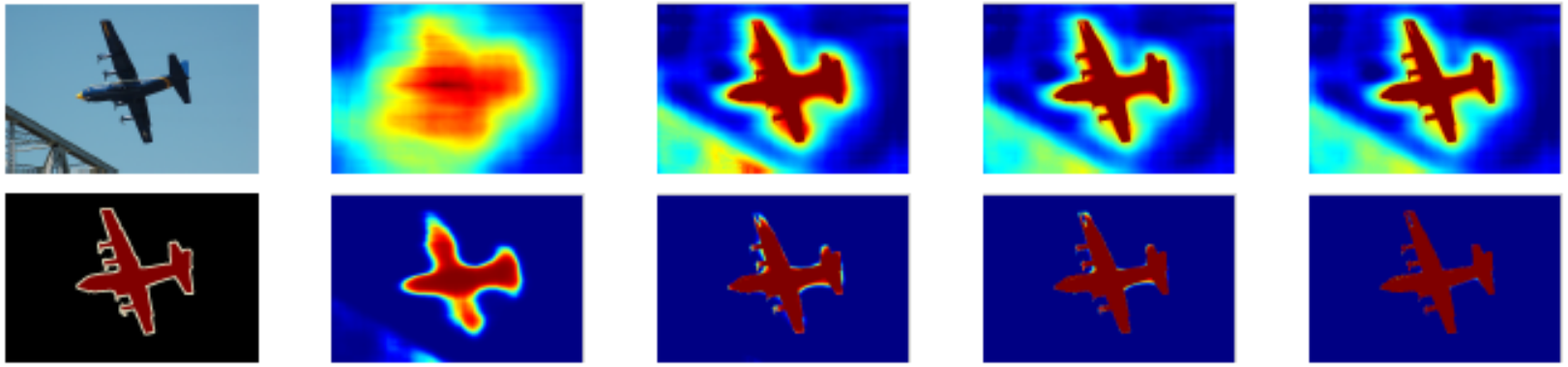
Chen et al., 2015, DeepLabv1 available at https://arxiv.org/pdf/1412.7062.pdf

Chen et al., 2018, DeepLabv2 available at https://arxiv.org/pdf/1606.00915.pdf

DeepLab v1 and v2

Fully Connected Conditional Random Field

- a post processing step, (10 iterations)



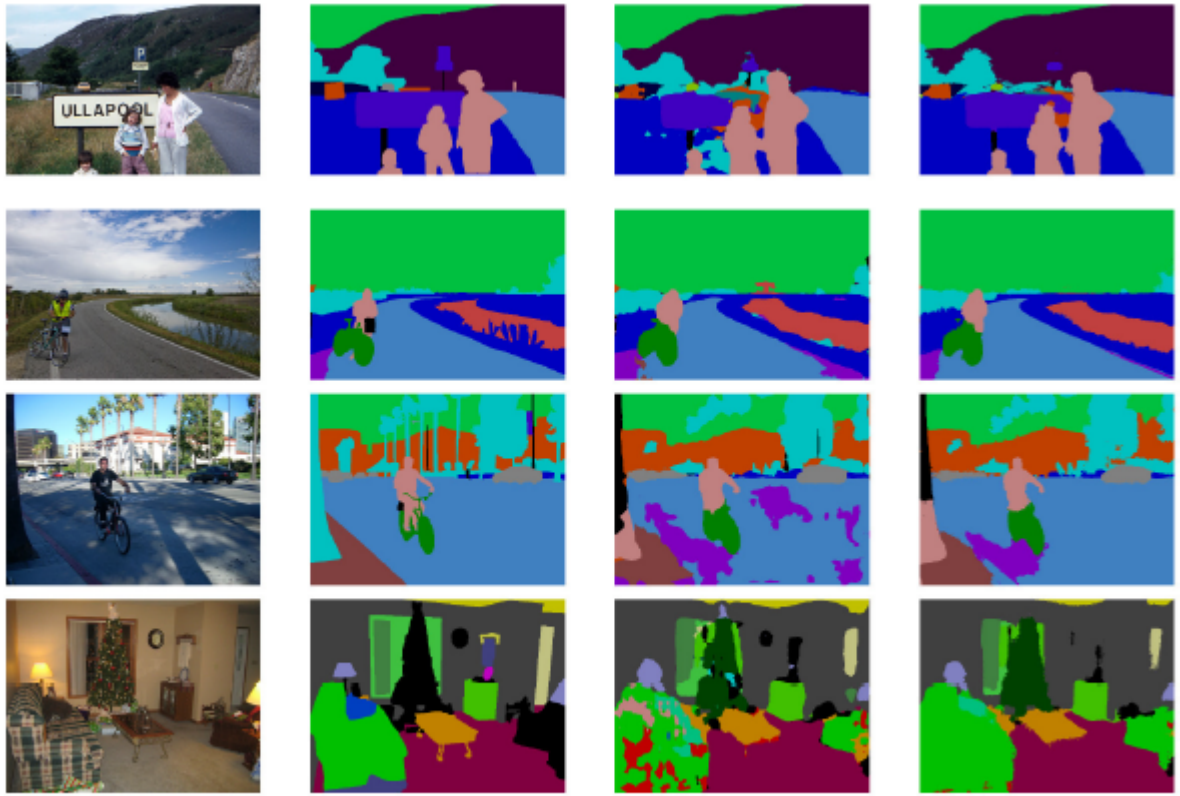
- exact solution is intractable, good approximations exist.
- **not an end-to-end** learning framework.
- not used in **DeepLabv3** and **DeepLabv3+ .**

Chen et al., 2015, DeepLabv1 available at https://arxiv.org/pdf/1412.7062.pdf

Chen et al., 2018, DeepLabv2 available at https://arxiv.org/pdf/1606.00915.pdf

DeepLab v1 and v2

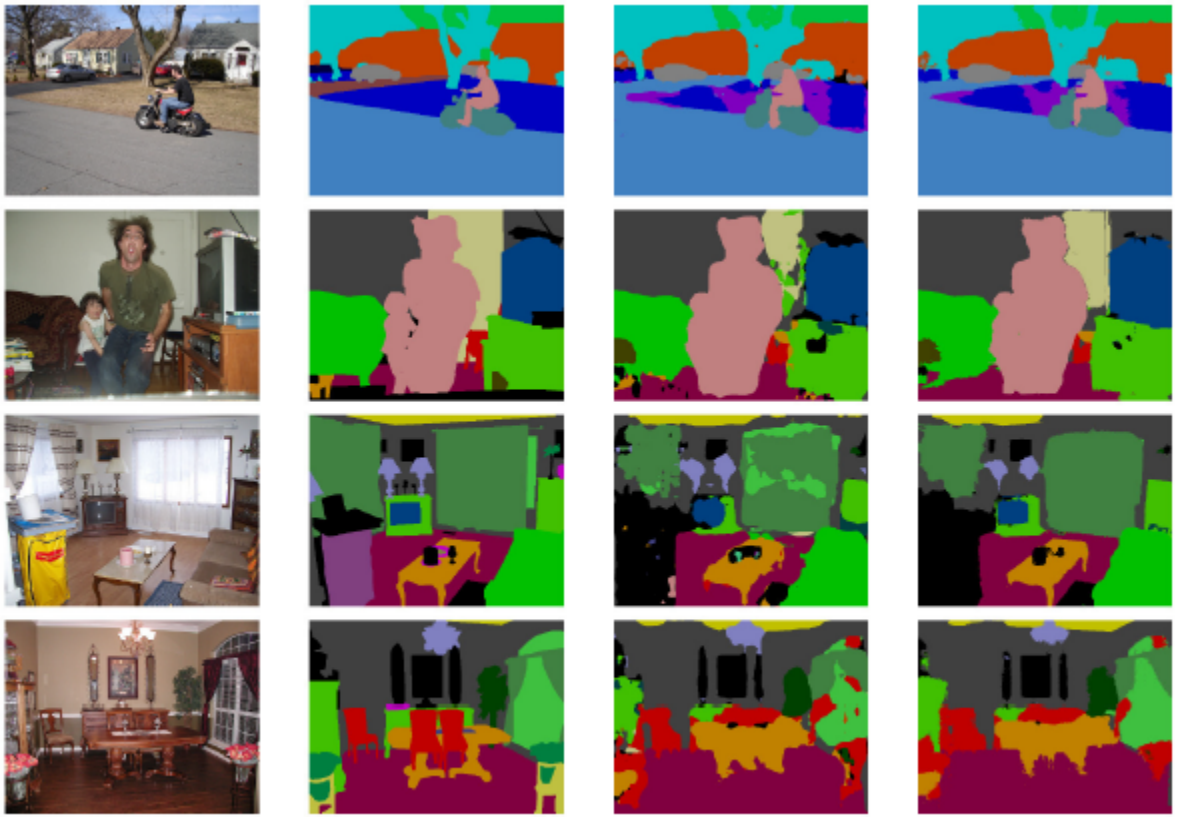
Qualitative results



Source https://towardsdatascience.com/review-deeplabv1-deeplabv2-atrous-convolution-semantic-segmentation-b51c5fbde92d

DeepLab v1 and v2

Qualitative results



(a) Image (b) G.T. (c) Before CRF (d) After CRF

Source https://towardsdatascience.com/review-deeplabv1-deeplabv2-atrous-convolution-semantic-segmentation-b51c5fbde92d

Overview

- Introduction
- Metrics
- Fully Convolutional Networks
  - SegNet (pool indices)
  - U-Net (skip connections)
  - ResUNet
  - DeepLabv1 & DeepLabv2 (Atrous conv. & ASPP)
  - PARSENet & PSPNet (Image Pooling – Parse Pooling)
  - DeepLabv3 & DeepLabv3+
- Loss Functions
- Practical Remark

Global Context

What is her pregnancy month?

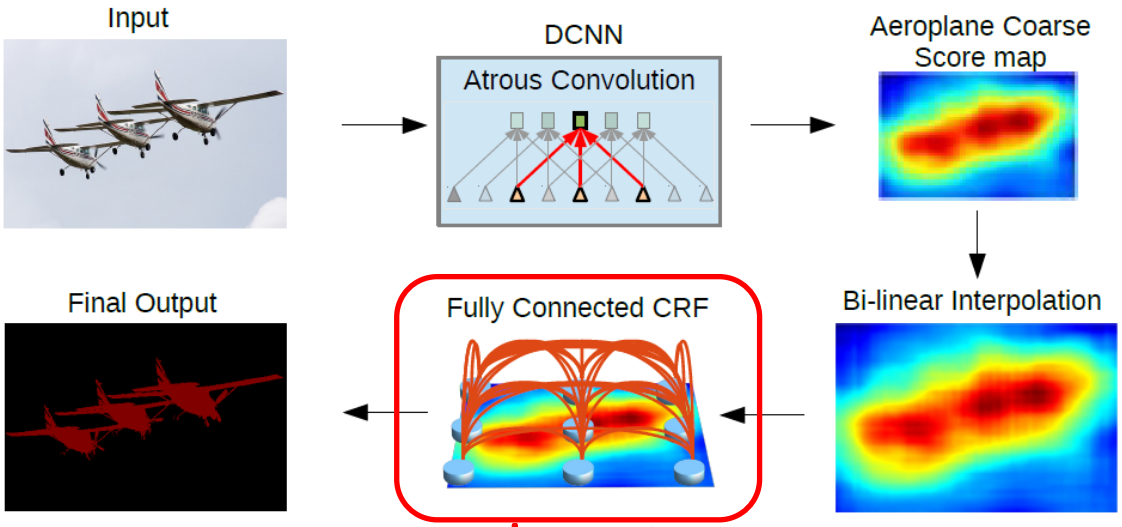
Global Context

Color of his shirt?

Global Context

"global context helps clarifying local confusion".

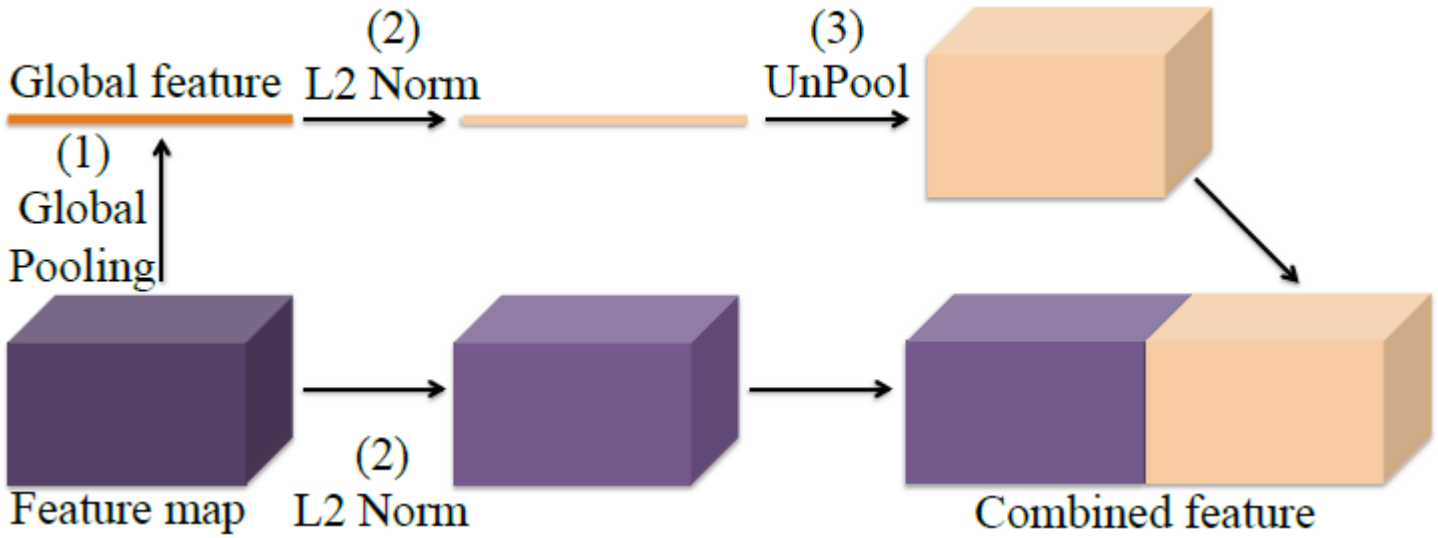
Recall DeepLab v2:



aims at alleviating this problem

Image Pooling or Image Level Feature

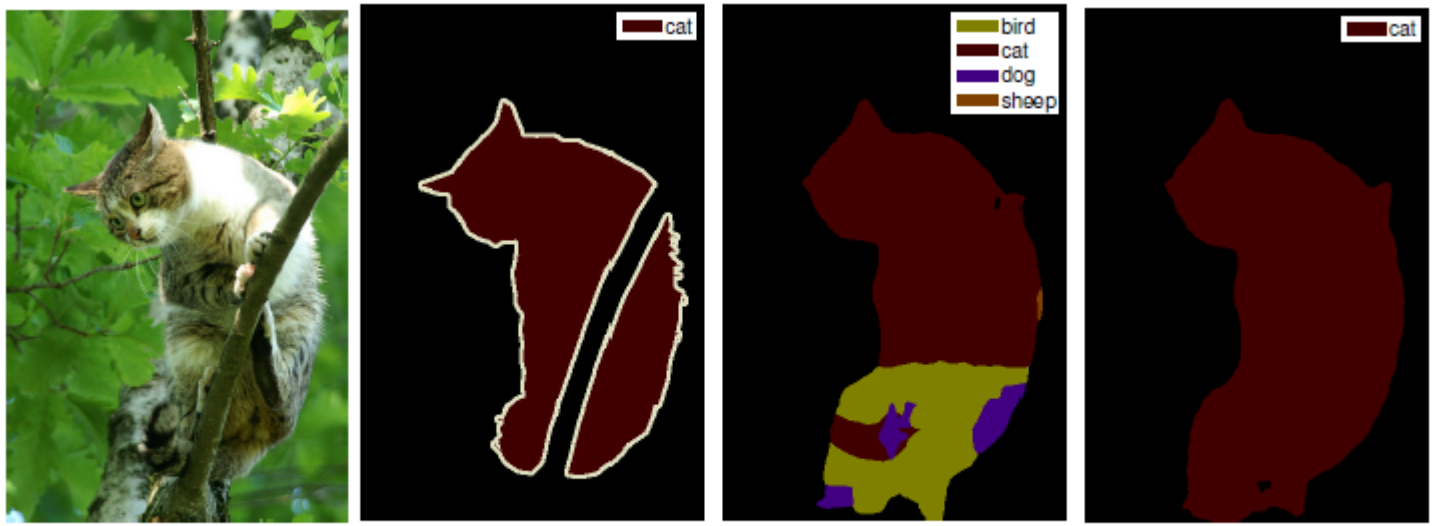
Introduced in *ParseNet: Looking Wider to See Better*



Source: Liu et al., 2016, *ParseNet: Looking Wider to See Better*

Image Pooling or Image Level Feature

Introduced in *ParseNet: Looking Wider to See Better*

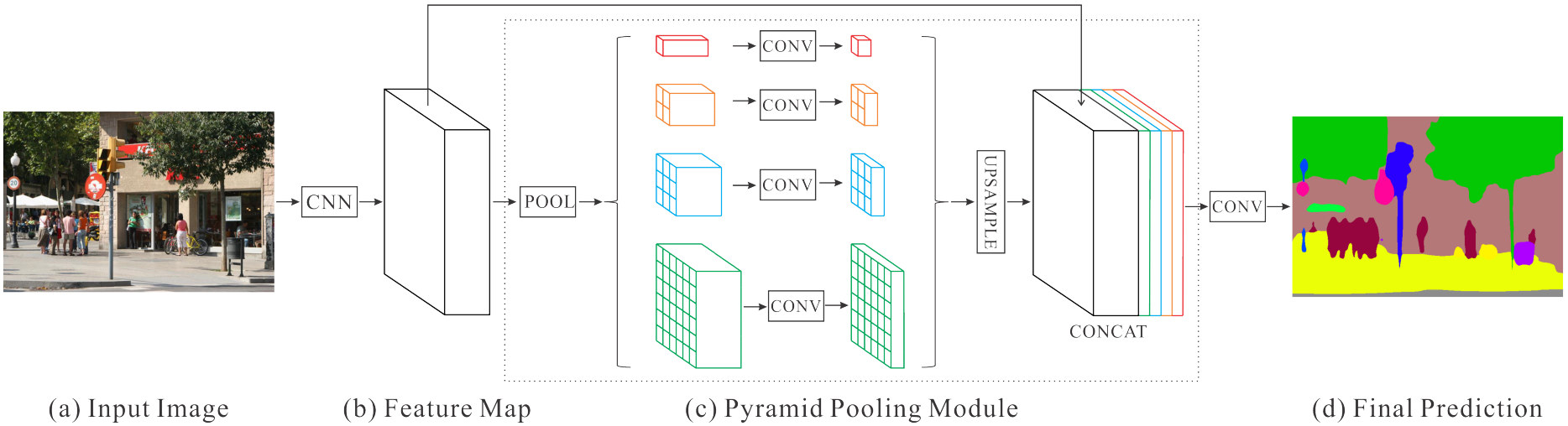


(a) Image (b) Ground truth (c) FCN output (d) ParseNet output

Source: Liu et al., 2016, *ParseNet: Looking Wider to See Better*

Pyramid Scene Pooling

Introduced in [Pyramid Scene Parsing Network]

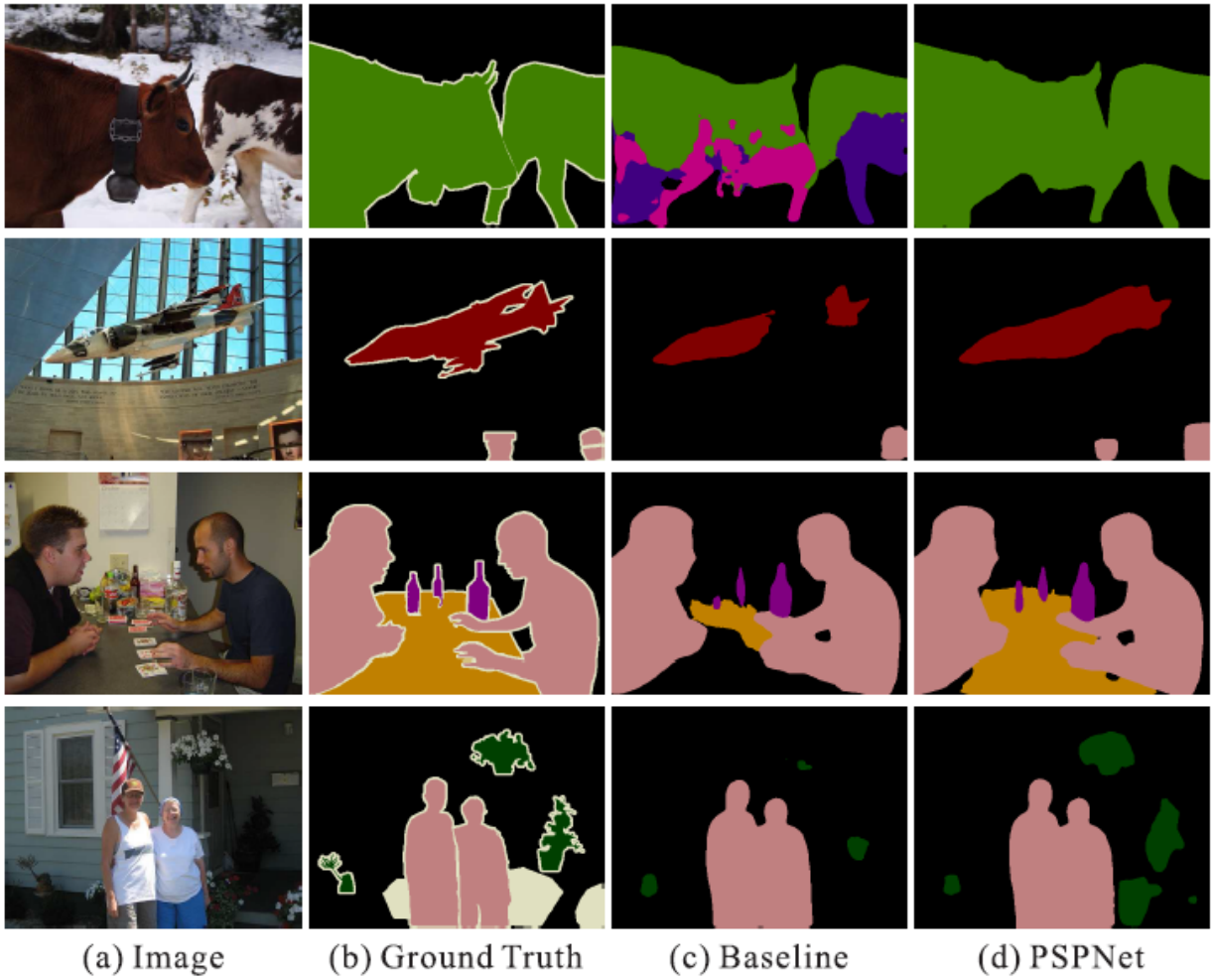


Source: [Zhao et al., 2017 Pyramid Scene Parsing Network CVPR (2017)]

Pyramid Scene Pooling

Qualitative results

PASCAL VOC 2012



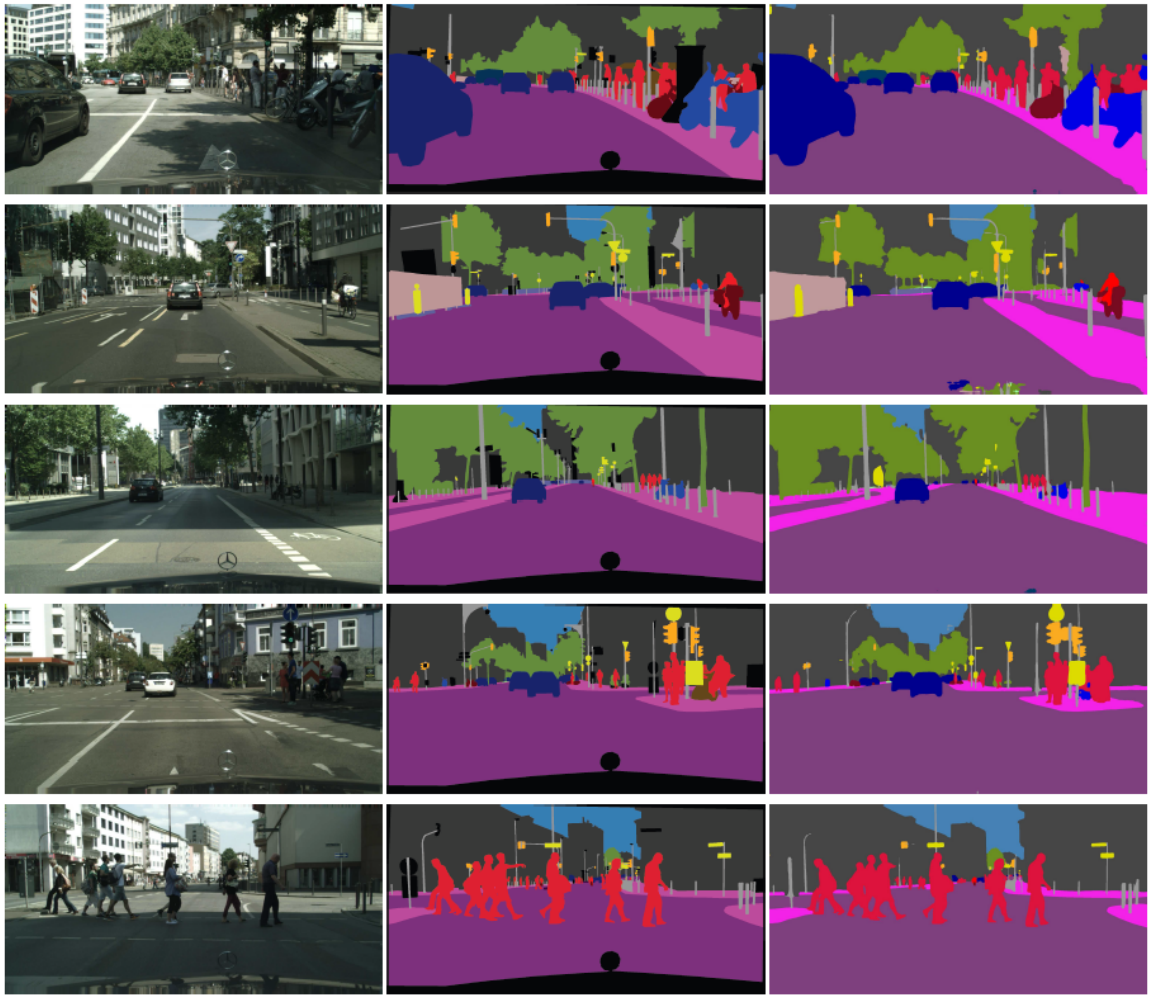
Baseline: ResNet50-based FCN with dilated network

Source: [Zhao et al., 2017 Pyramid Scene Parsing Network CVPR (2017)]

Pyramid Scene Pooling

Qualitative results

Cityscapes



(a) Image (b) Ground Truth (c) PSPNet

Baseline: ResNet50-based FCN with dilated network

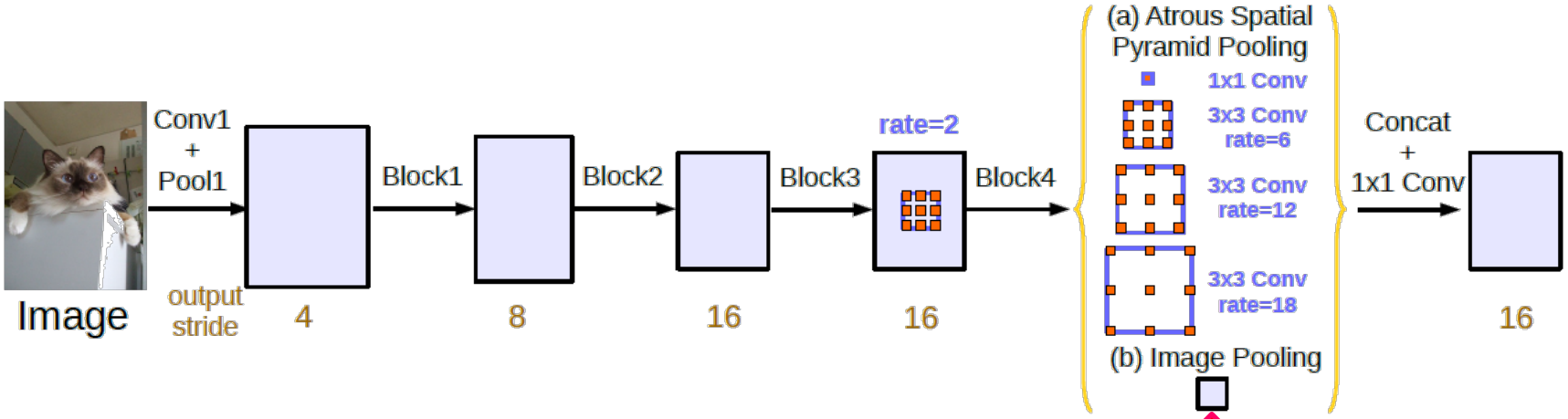
Source: [Zhao et al., 2017 Pyramid Scene Parsing Network CVPR (2017)]

Overview

- Introduction
- Metrics
- Fully Convolutional Networks
  - SegNet (pool indices)
  - U-Net (skip connections)
  - ResUNet
  - DeepLabv1 & DeepLabv2 (Atrous conv. & ASPP)
  - PARSENet & PSPNet (Image Pooling – Parse Pooling)
  - DeepLabv3 & DeepLabv3+
- Loss Functions
- Practical Remark

DeepLab v3

- **one 1×1 convolution and three 3×3 convolutions with rates = (6, 12, 18)** when output stride = 16.



- Also, **image pooling**, or **image-level feature.**
- All with **256 filters** and **BN**.
- Concatenated results pass through a 1x1 conv before final logits.
- No **CRF.**

Chen et.al., 2017 DeepLabv3 available at https://arxiv.org/pdf/1706.05587.pdf

DeepLab v3

Demo DeepLabv3: cityscapes

Available in Youtube at https://www.youtube.com/watch?v=ATIcEDSPWXY

DeepLabv3+

Inherits from DeepLabv2:

- Atrous convolution
- Atrous Spatial Pyramid Pooling (ASPP)

Main novelties :

- Brings back the encoder-decoder structure
- depth wise separable convolution to both ASPP module and decoder module

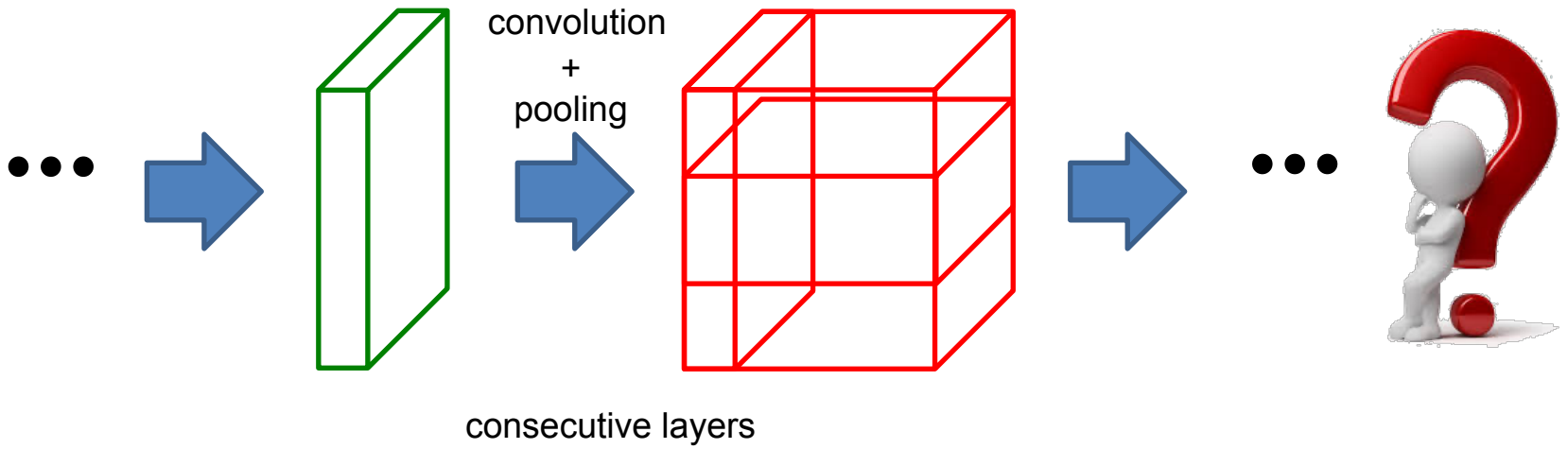
Chen et.al., 2018 DeepLabv3+ available at https://arxiv.org/pdf/1802.02611.pdf

A Dilemma

More filters → more capacity → more memory.

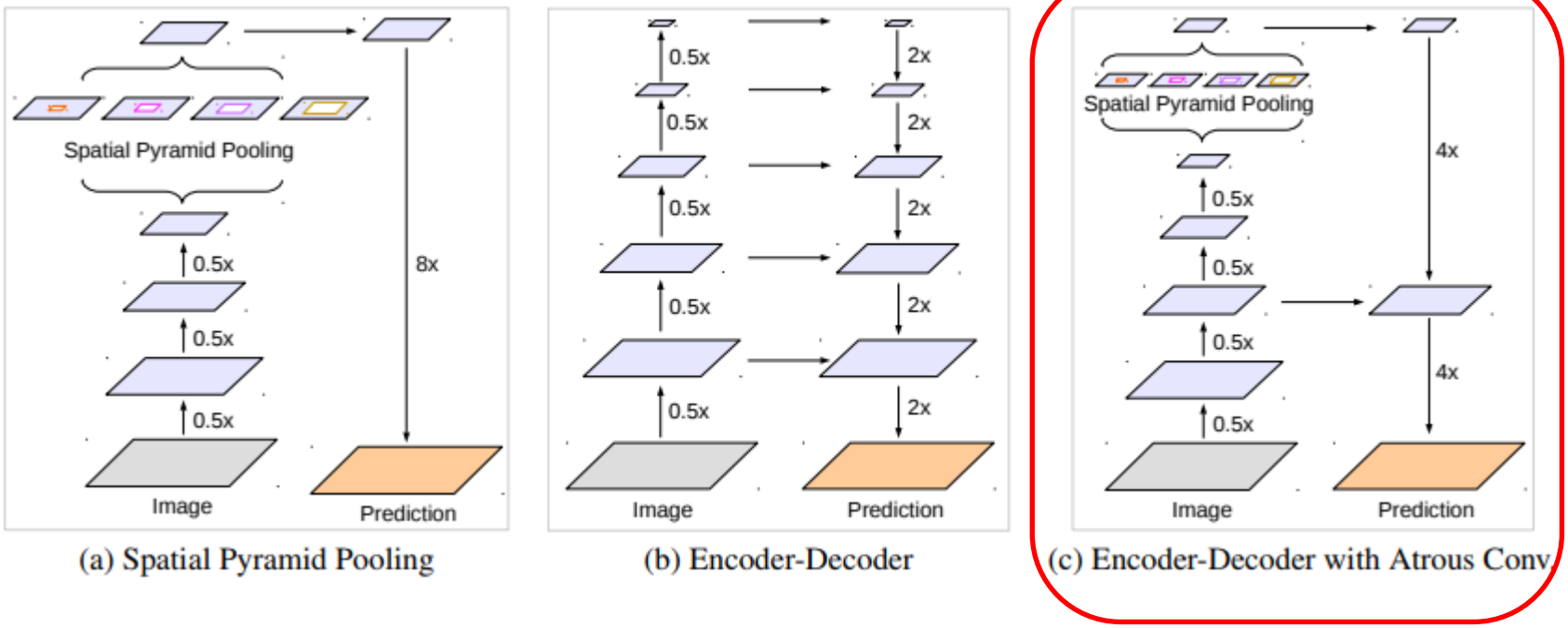
For more capacity with same memory:

reduce resolution → loose spatial information.



Back to the encoder-decoder

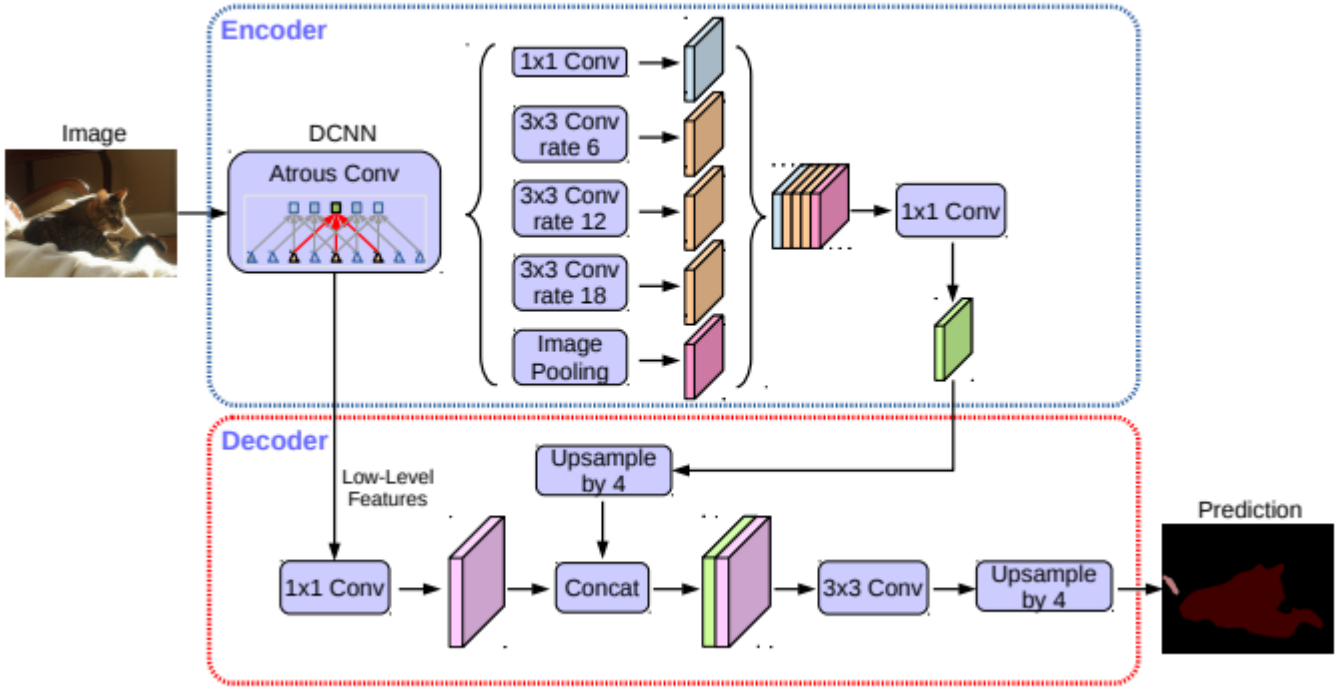
Combines atrous convolution and skip connections to **recover object boundaries information** lost due to pooling and striding, at a **lower computational cost**.



Chen et.al., 2018 DeepLabv3+ available at https://arxiv.org/pdf/1802.02611.pdf

Encoder-Decoder structure

- Encoder encodes multi-scale contextual information applying atrous convolutions.
- Decoder refines segmentation on object boundaries.



Chen et.al., 2018 DeepLabv3+ available at https://arxiv.org/pdf/1802.02611.pdf

Depthwise Separable Convolutions

"… a powerful operation to reduce the computation cost and number of parameters while maintaining similar (or slightly better) performance."

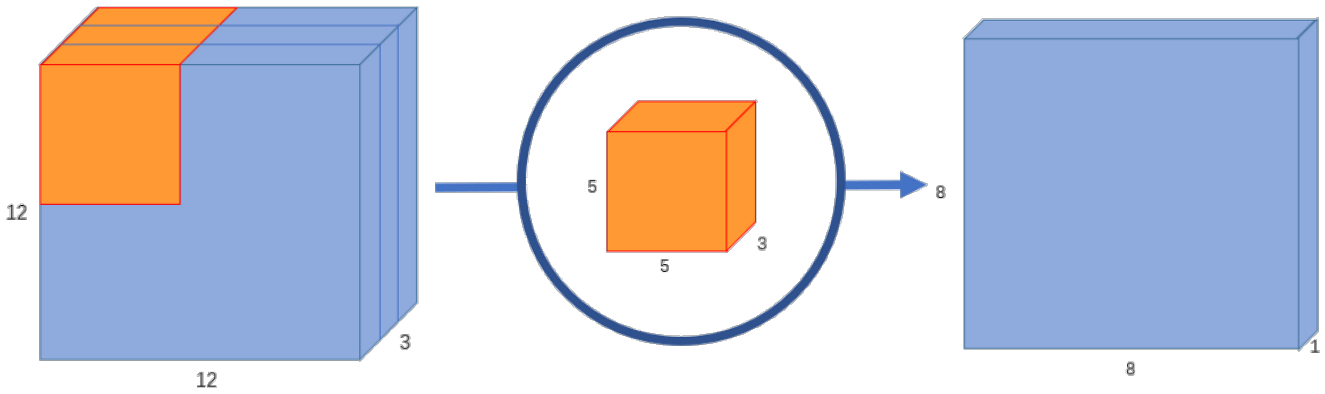
Convolution is broken down into two steps:

- Depth wise convolution: a spatial convolution independently for each input channel
- Pointwise convolution: combine the output from the depth wise convolution.

Chen et.al., 2018 DeepLabv3+ available at https://arxiv.org/pdf/1802.02611.pdf

Normal vs Separable Convolution

Normal Convolution



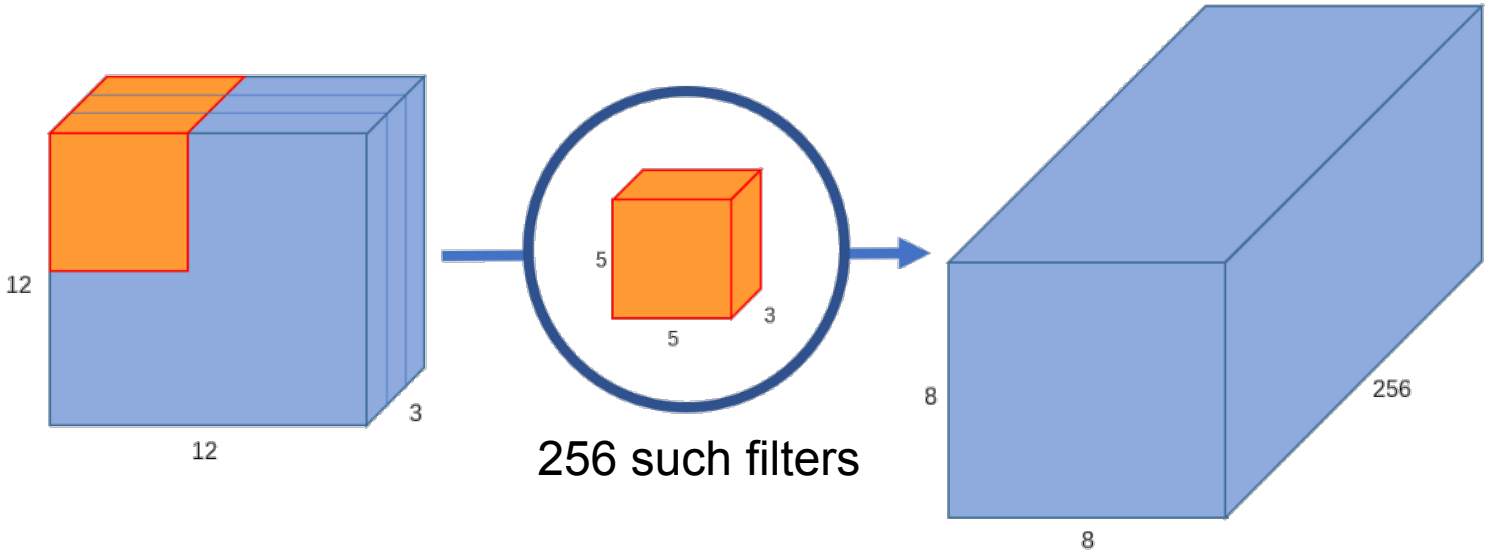
Number of parameters: (5x5x3) = 75

Number of multiplications: 75 x (8x8) = 4,800

Source: https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728

Normal vs Separable Convolution

Normal Convolution - with multiple filters



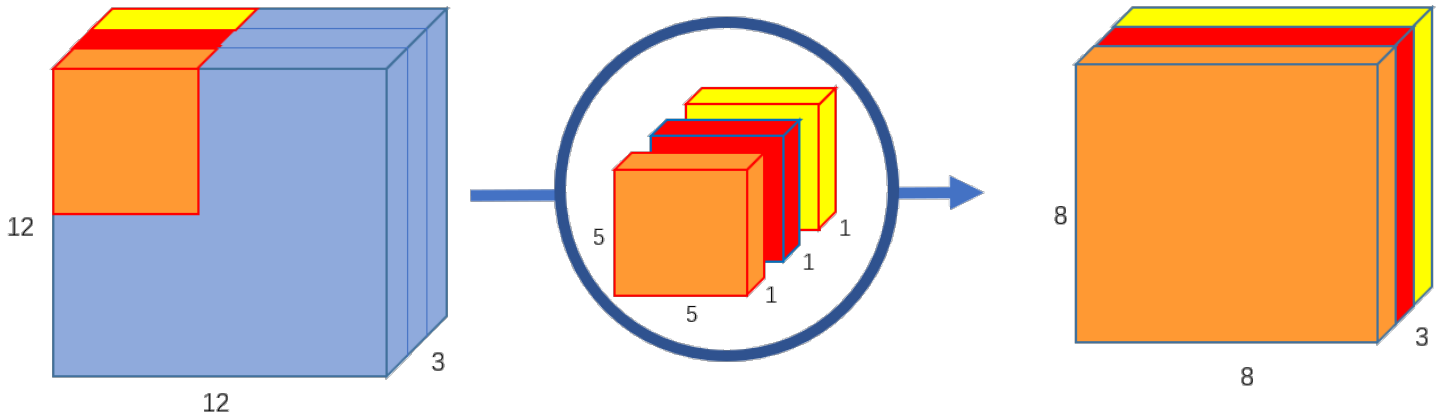
Number of parameters: (5x5x3) x 256 = 19,200

Number of multiplications: 75 x (8x8) x 256 = 1,245,952

Source: https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728

Normal vs Separable Convolution

Separable convolution: first, depth wise



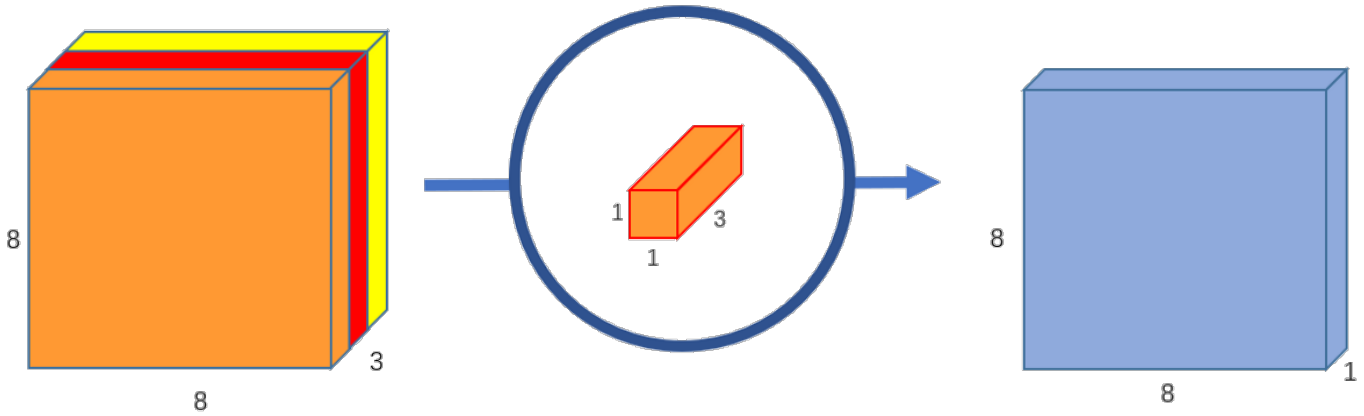
Number of parameters: (5x5x1) x 3 = 75

Number of multiplications: 75 x (8x8) = 4,800

Source: https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728

Normal vs Separable Convolution

Separable convolution: then, point wise



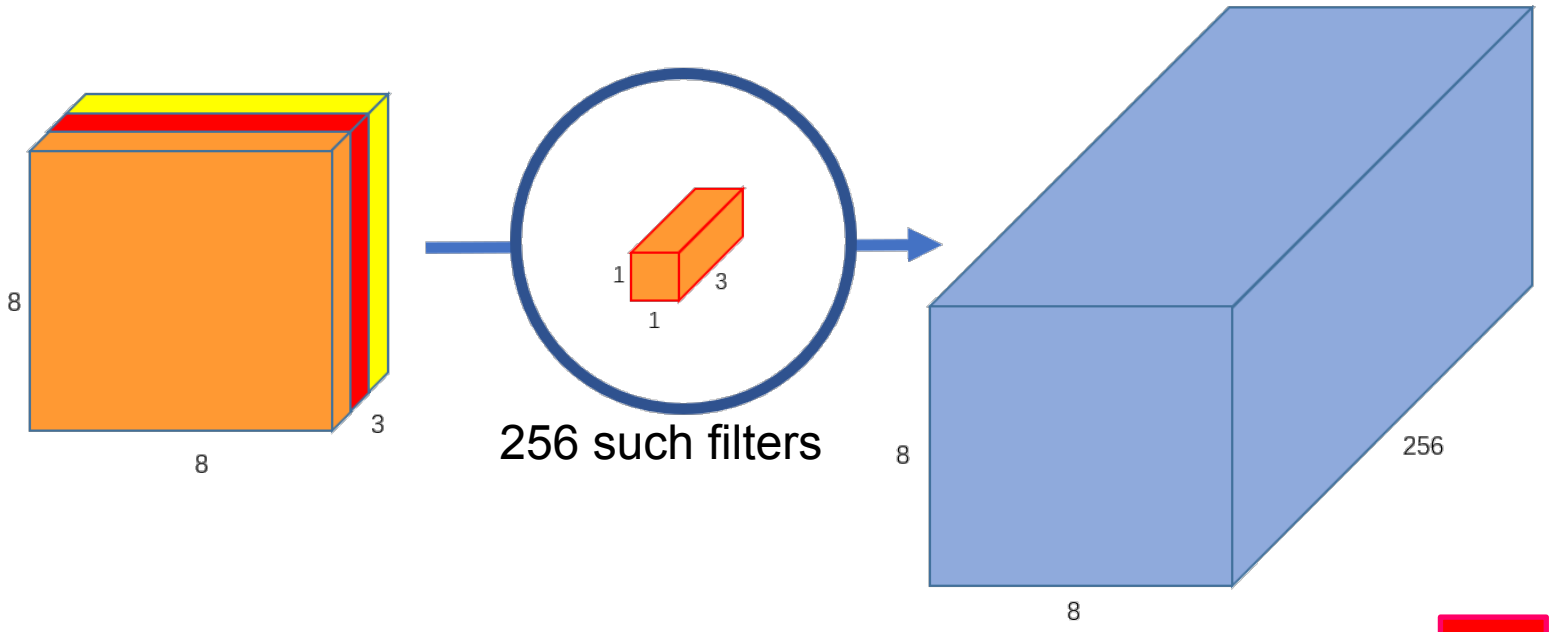
Number of parameters: 75 + (1x1x3) = 78

Number of multiplications: 4,800 + (1x1x3) x (8x8) = 74,992

Source: https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728

Normal vs Separable Convolution

Separable convolution: then, point wise, with multiple filters



Number of parameters: 75 + (1x1x3) x 256 = 843

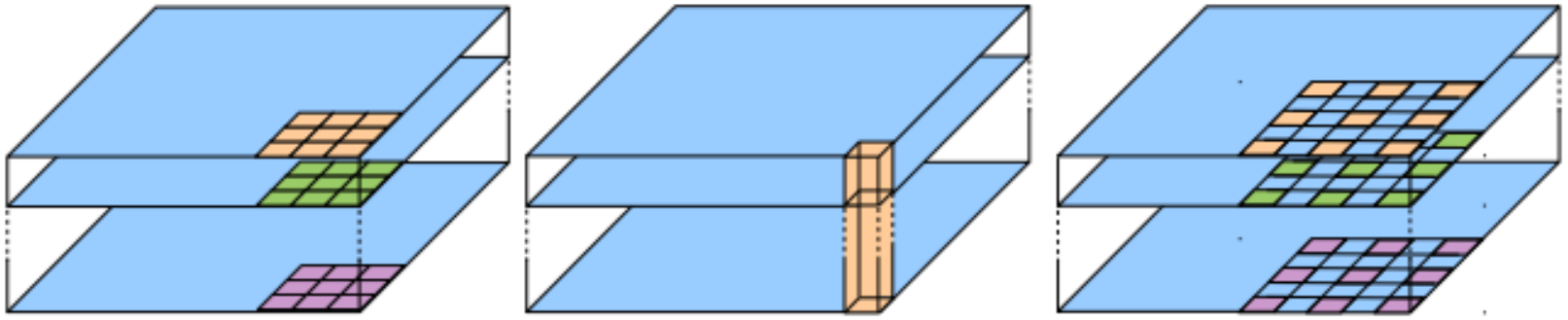
Number of multiplications: 4,800 + (1x1x3) x (8x8) x 256 = 53,952

4%

Source: https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728

Atrous Separable Convolution

Combines depthwise and pointwise convolution to create Atrous Separable Convolution.



The concept has been used in so called MobileNets* conceived for mobile and embedded applications.

*Howard, et al., 2017 MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, available at https://arxiv.org/pdf/1704.04861.pdf

Picture source: https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728

DeepLabV3+

DeepLabV3+ demo: detection of weed in farm field

From Youtube : https://www.youtube.com/watch?v=FbY1zvMBt-s

Overview

- Introduction
- Metrics
- Fully Convolutional Networks
  - SegNet (pool indices)
  - U-Net (skip connections)
  - ResUNet
  - DeepLabv1 & DeepLabv2 (Atrous conv. & ASPP)
  - PARSENet & PSPNet (Image Pooling – Parse Pooling)
  - DeepLabv3 & DeepLabv3+
- Loss Functions
- Practical Remark

FCN – class unbalance

When the instances' sizes are very different among the classes we have unbalanced training sets.

Sample replication is not enough.

Cross Entropy

$$CE = -\frac{1}{N}\sum_{i}\log\left(p_i\right)$$

probability of the true class at site i (y_i)

number of samples being classified

where

- CE is the cross-entropy

Problem:

- Majority classes dominate the computation of *CE*
- Minority classes tend to vanish in the inference.

Balanced Cross Entropy

The **loss function** is weighted to compensate for unbalanced training data.

$$BCE = -\frac{1}{N}\sum_{i} \omega_{y_i} \log\left(p_i\right)$$

probability of the true class at site i (y_i)

precomputed weights

where

- BCE is the balanced cross-entropy
- ω_{y_i} is the weight for the true class (y_i) ; larger for less abundant classes
- In binary classification ω_{y_i} tunes the tradeoff between precision and recall.

Balanced Cross Entropy

The **loss function** is weighted to compensate for unbalanced training data.

$$BCE = -\frac{1}{N}\left[\sum_{y_i=pos} \omega_{pos}\log(p_i) \quad + \sum_{y_i=neg} \omega_{neg}\log(p_i)\right]$$

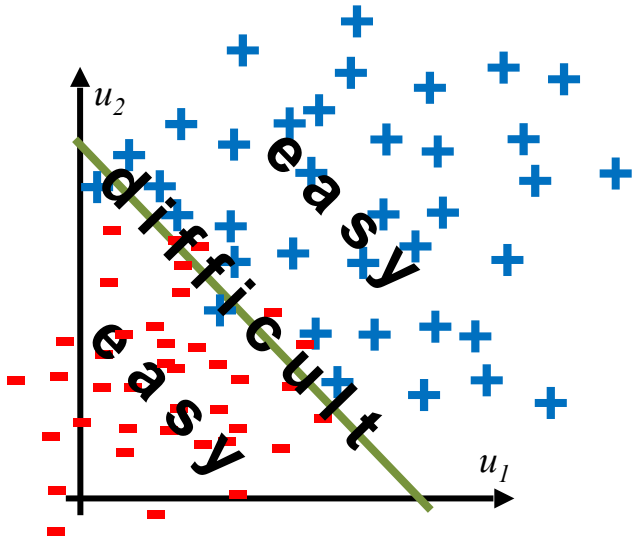
where

- BCE is the balanced cross-entropy
- ω_{y_i} is the weight for the true class (y_i) ; larger for less abundant classes
- In binary classification ω_{y_i} ($\omega_{pos}/\omega_{neg}$) tunes the tradeoff between precision and recall.

Focal Loss

Motivation: What samples are most determinant of the decision boundary?

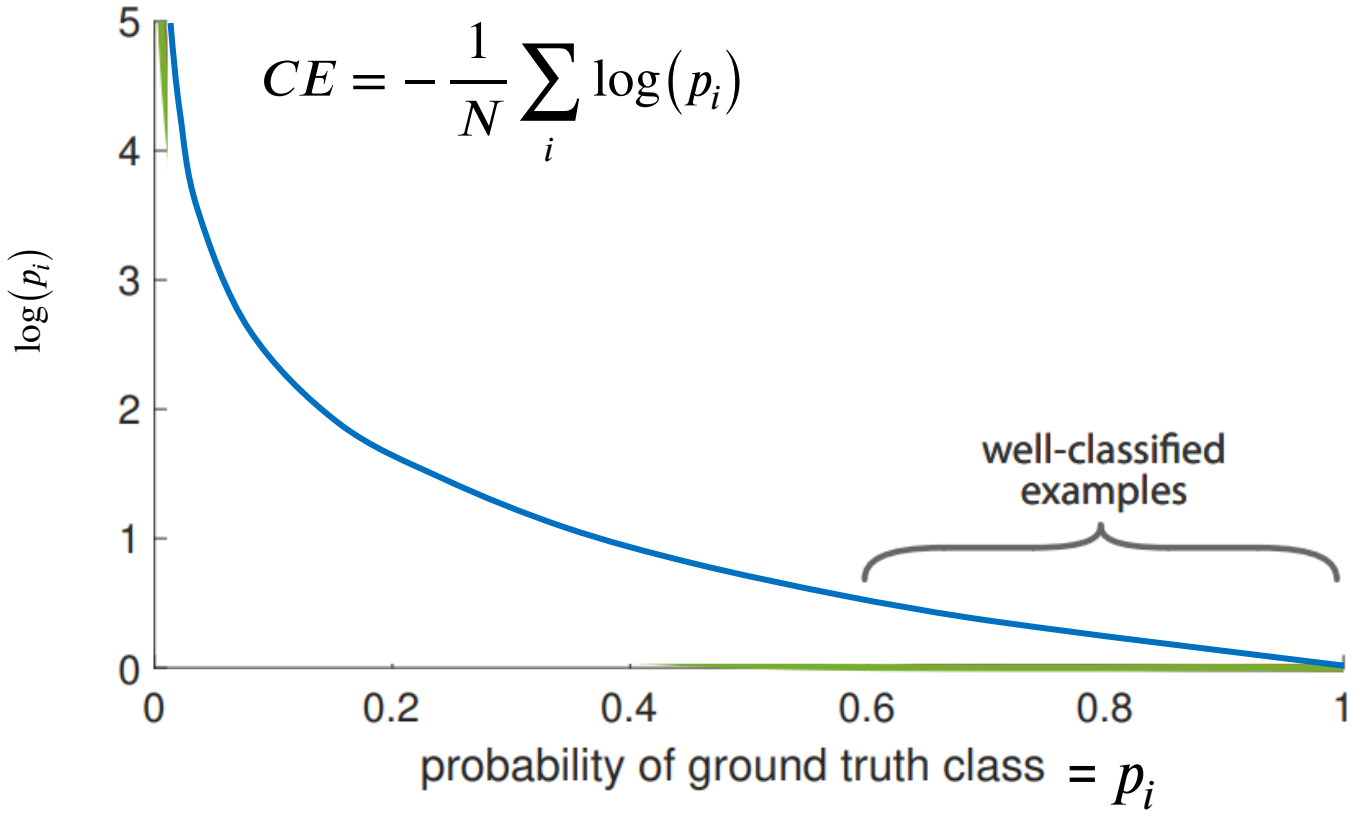
- The ones close to the boundary,
- The ones that are difficult to classify,
- The ones with low probability of belonging to the true class.



- However, easy to classify samples still add to the loss, especially if they are from majority classes.

Focal Loss

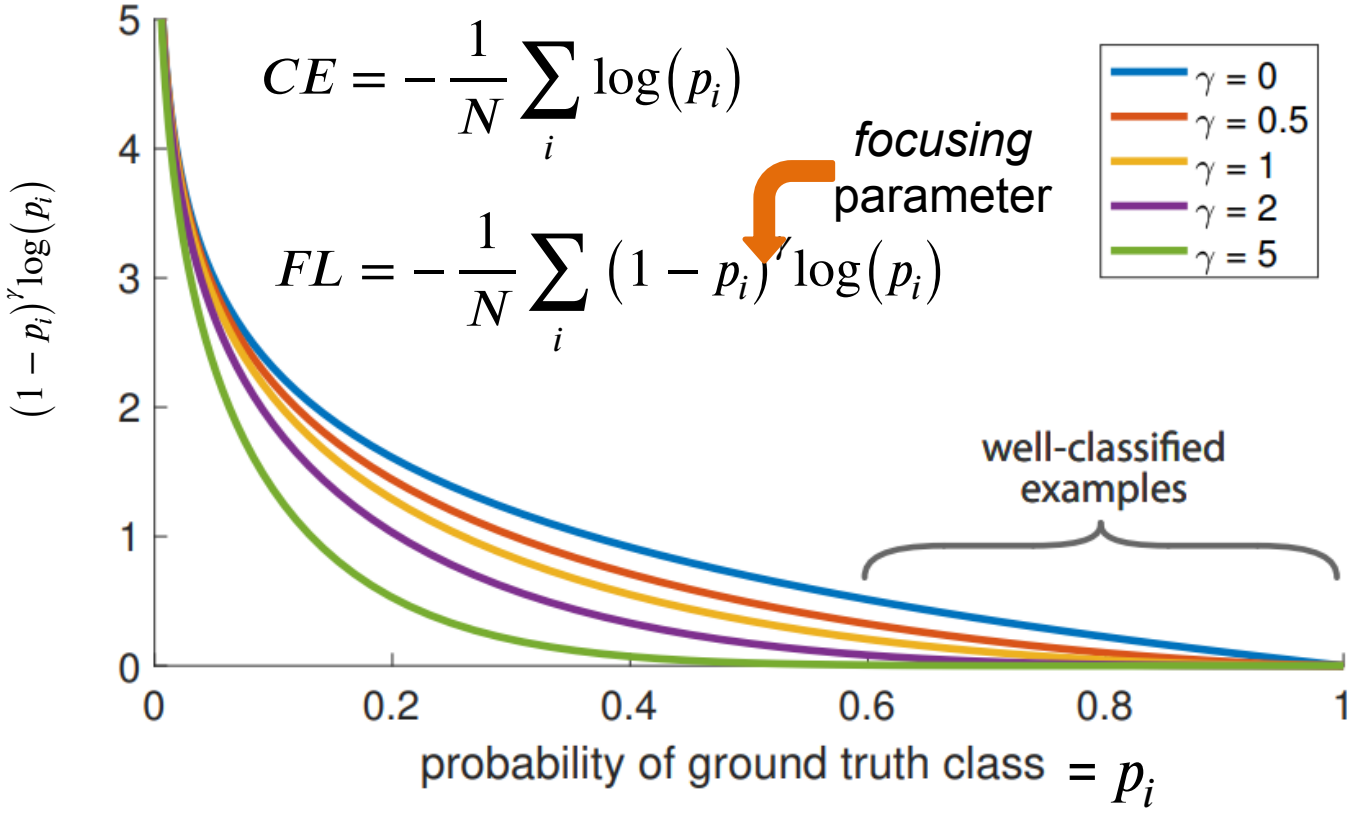
Solution: to down weight the loss assigned to well, easy to classify samples.



Source : Lin et all. (2018) "Focal Loss for Dense Object Detection"

Focal Loss

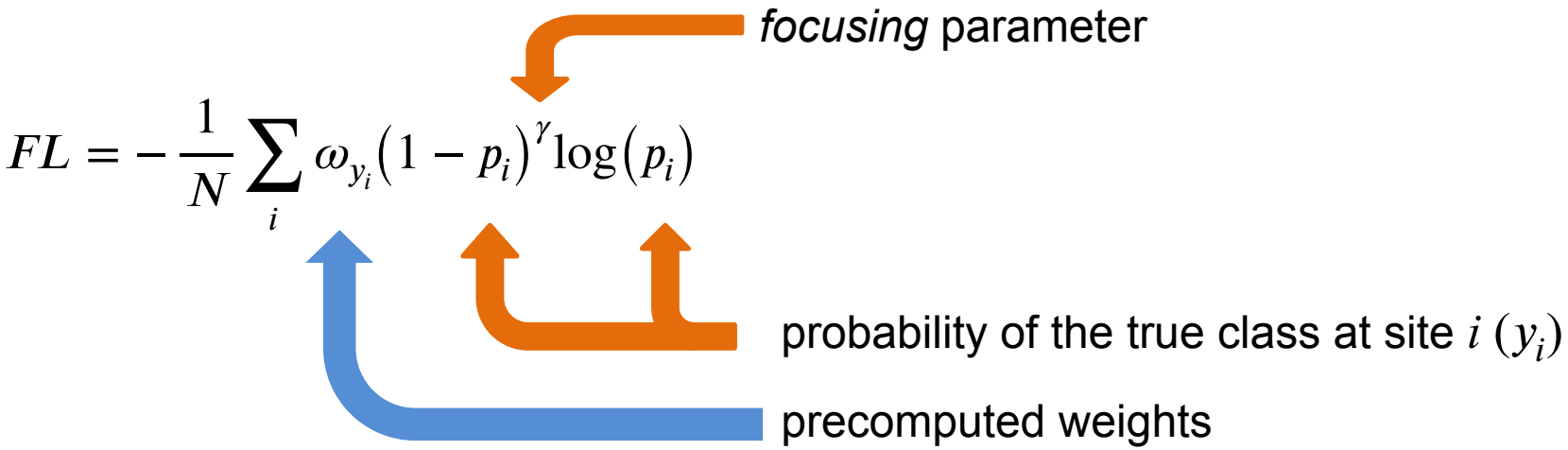
Solution: down weights the loss assigned to well, easy to classify samples.



Source : Lin et all. (2018) "Focal Loss for Dense Object Detection"

Balanced Focal Loss

Improved accuracies may be achieved by combining the two previous approaches:



Problem: one additional parameter to estimate.

Source : Lin et all. (2018) "Focal Loss for Dense Object Detection"

Loss Functions for Regression

When output is non-categorical (real values), the L_2 and L_1 norms are a common choice:

$$L_2 = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right|^2 \qquad L_1 = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right|$$

where

- y_i is the value computed by the model
- $\hat{y}_i$ is the reference value
- N is the number of samples.

More on Loss-functions

A survey of loss functions for semantic segmentation

Shruti Jadon
IEEE Member
shrutijadon@ieee.org

***Abstract*—Image Segmentation has been an active field of research as it has a wide range of applications, ranging from automated disease detection to self driving cars. In the past 5 years, various papers came up with different objective loss functions used in different cases such as biased data, sparse segmentation, etc. In this paper, we have summarized some of the well-known loss functions widely used for Image Segmentation and listed out the cases where their usage can help in fast and better convergence of a model. Furthermore, we have also introduced a new log-cosh dice loss function and compared its performance on NBFS skull-segmentation open source data-set with widely used loss functions. We also showcased that certain loss functions perform well across all data-sets and can be taken as a good baseline choice in unknown data distribution scenarios.**

***Index Terms*—Computer Vision, Image Segmentation, Medical Image, Loss Function, Optimization, Healthcare, Skull Stripping, Deep Learning**

I. Introduction

Deep learning has revolutionized various industries ranging from software to manufacturing. Medical community has also benefited from deep learning. There have been multiple innovations in disease classification, example, tumor segmen-

have proposed a new log-cosh dice loss function for semantic segmentation. To showcase its efficiency, we compared the performance of all loss functions on NBFS Skull-stripping dataset [1] and shared the outcomes in form of Dice Co-efficient, Sensitivity, and Specificity. The code implementation is available at GitHub: https://github.com/shruti-jadon/Semantic-Segmentation-Loss-Functions.

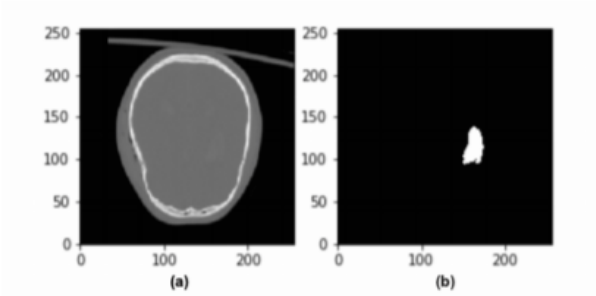


Fig. 1. Sample Brain Lesion Segmentation CT Scan [2]. In this segmentation mask you can see, that number of pixels of white area(targeted lesion) is less than number of black pixels.

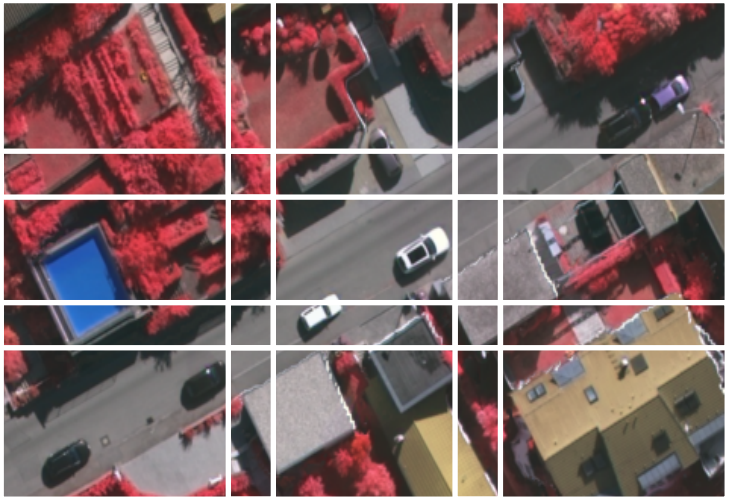
Jorgan, S., 2020, A survey of loss functions for semantic segmentation

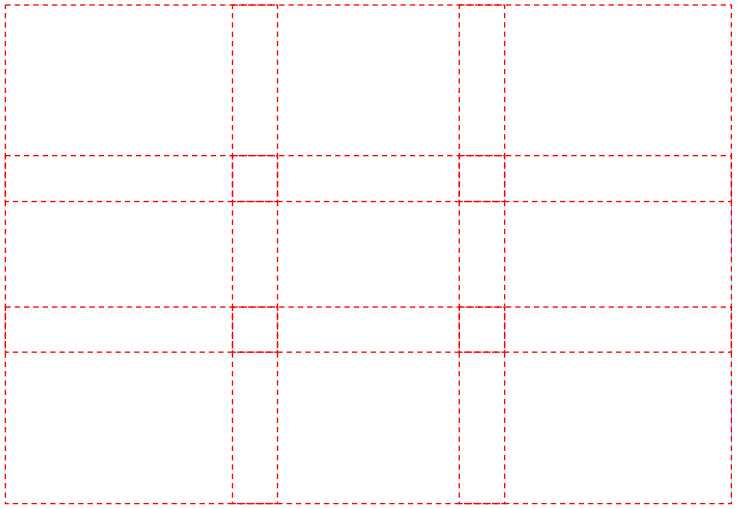
Overview

- Introduction
- Metrics
- Fully Convolutional Networks
  - SegNet (pool indices)
  - U-Net (skip connections)
  - ResUNet
  - DeepLabv1 & DeepLabv2 (Atrous conv. & ASPP)
  - PARSENet & PSPNet (Image Pooling – Parse Pooling)
  - DeepLabv3 & DeepLabv3+
- Loss Functions
- Practical Remark

Patch-wise training and inference

In practice large images are processed in tiles





- Final result is the mosaic of patches' results.
- Less spatial context at patches' borders.
- Use overlapping patches and
  - Consider the inner part,
  - Average the results.

Spatial accuracy

Example:

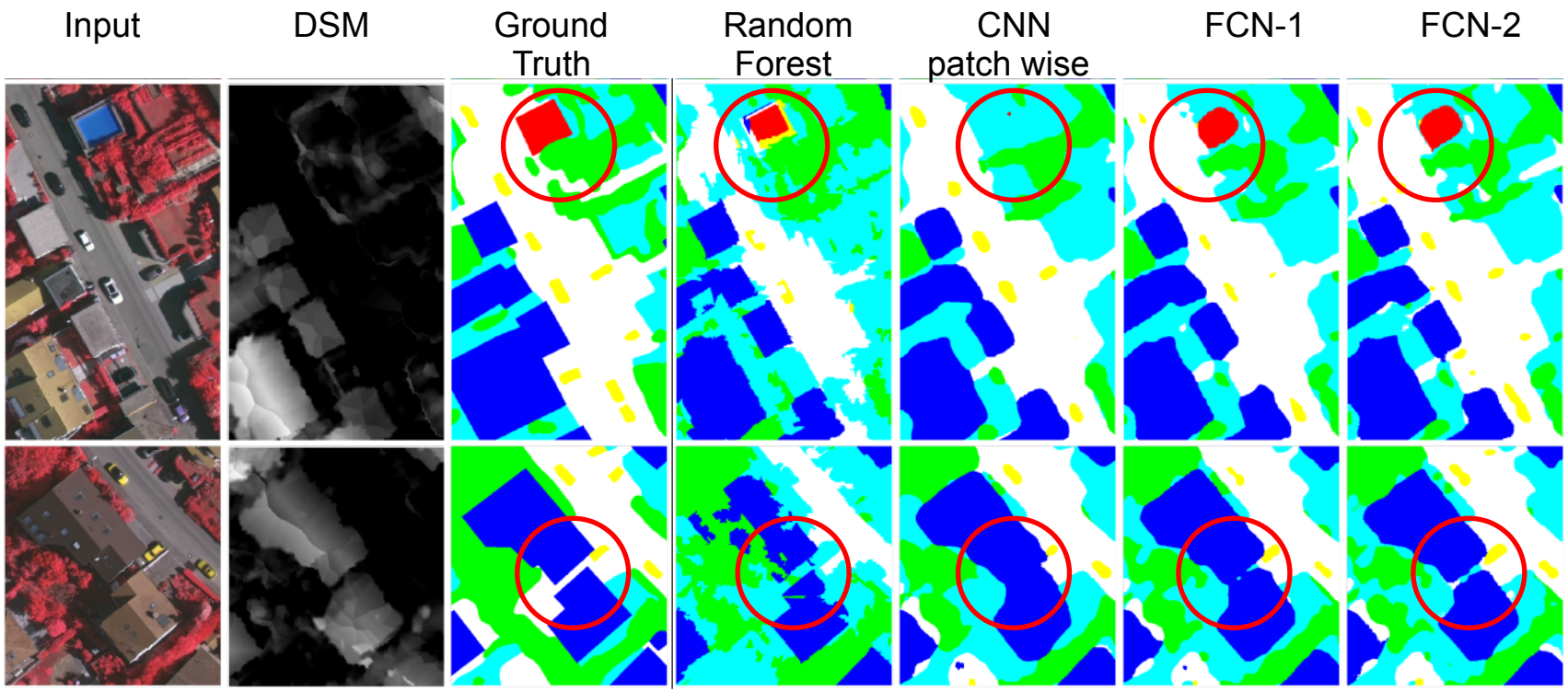
- The CNN scores high for “car” and “road“ for both patches on the right.

- In consequence, CNN patch classification over-smooths objects boundaries.

- FCN, on the contrary, learns class specific **structures** contained in each patch.

CNN patch wise vs. FCN

FCN



From: Volpi, M. and Tuia, D., 2017. Dense Semantic Labeling of Subdecimeter Resolution Images With Convolutional Neural Networks, IEEE Transactions on Geoscience and Remote Sensing , Vol. 55(2), pp. 881-893.

Next Lecture

Thursday
Lab + 2nd Programming Assignment

Semantic Segmentation

See you next class!