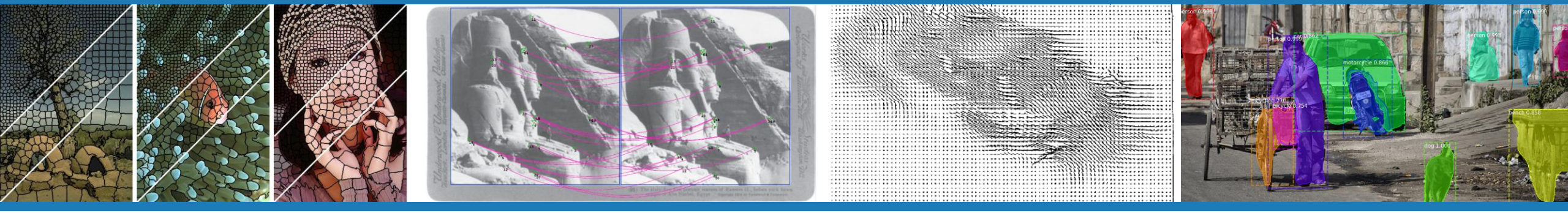
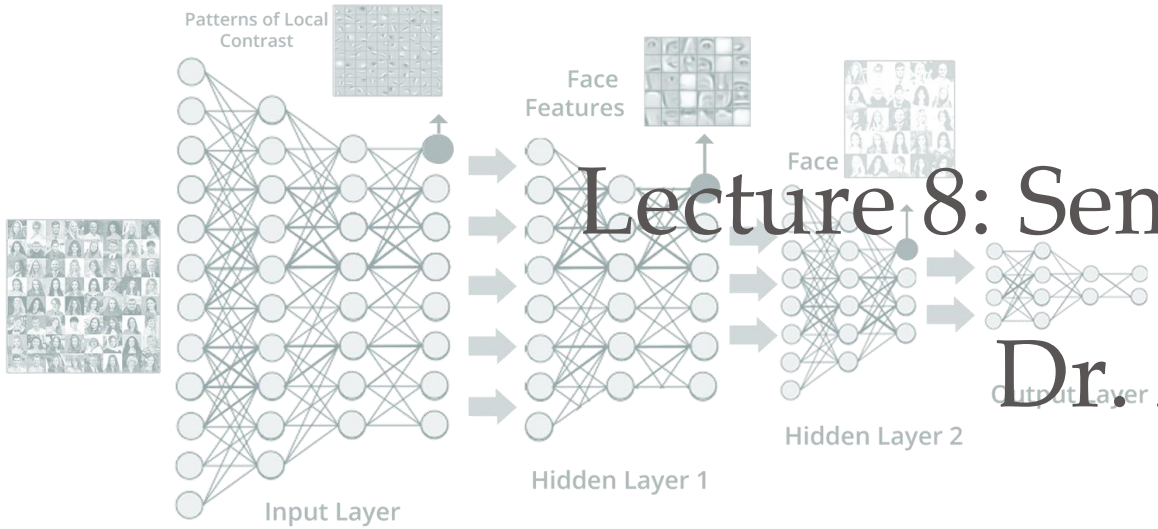
INSTITUTE FOR MACHINE LEARNING AND ANALYTICS



Computer Vision



Lecture 8: Semantic Segmentation

Dr. Xiao Zhao

EMI Elektrotechnik, Medizintechnik und Informatik

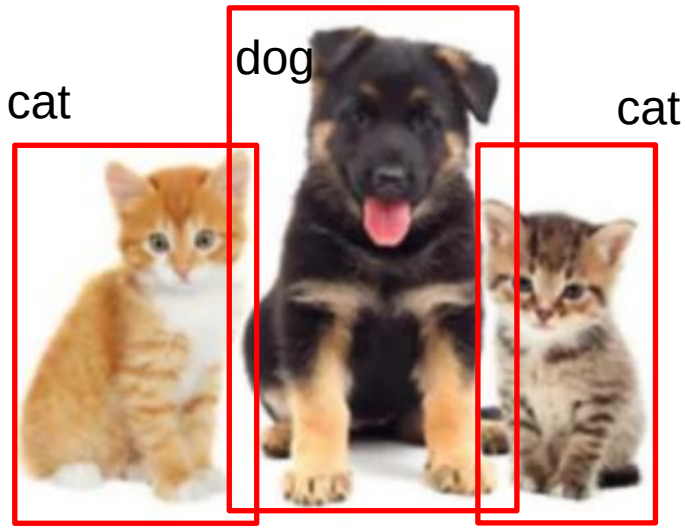
23. Mai 2025

Segmentation Tasks

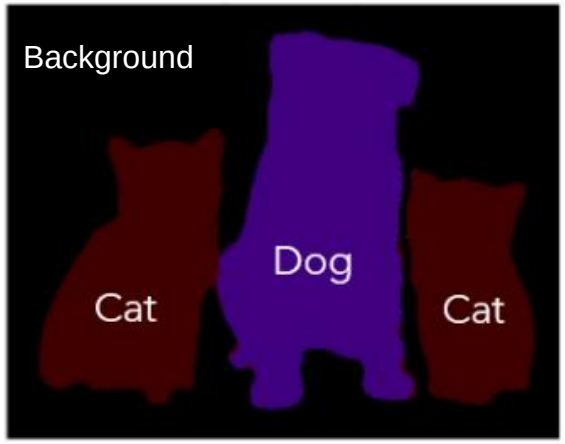
Input Image

network

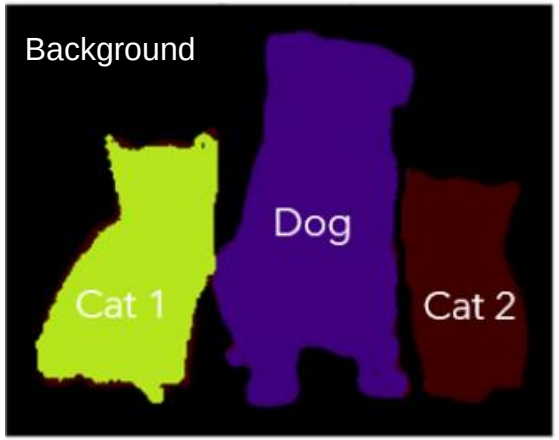
Object Detection



Semantic Segmentation



Instance Segmentation



Source: https://www.v7labs.com/blog/instance-segmentation-guide

Semantic Segmentation

- Assign each pixel in the image with a category label
- Do not differentiate instances of the same category

Source: https://d2l.ai/chapter_computer-vision/semantic-segmentation-and-dataset.html

Instance Segmentation

- Each instance of a category will be assigned by a different label.

Semantic Segmentation

Instance Segmentation

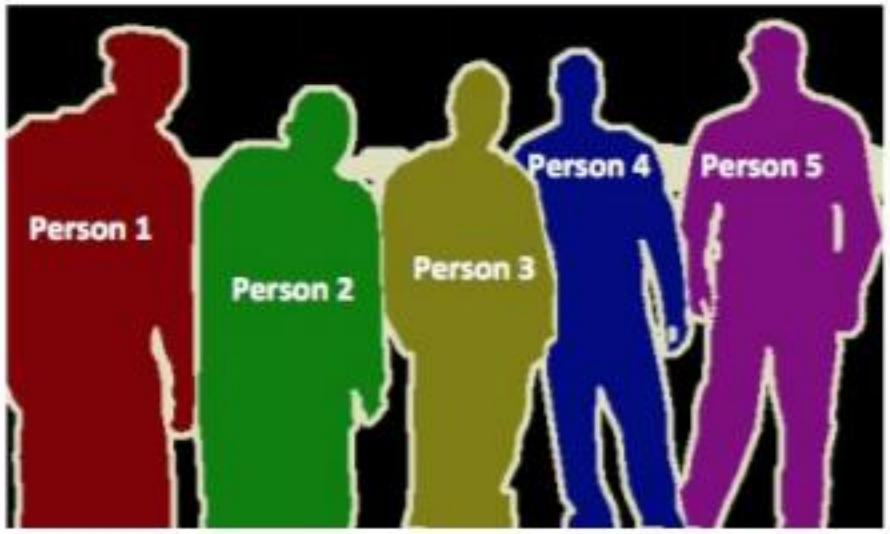
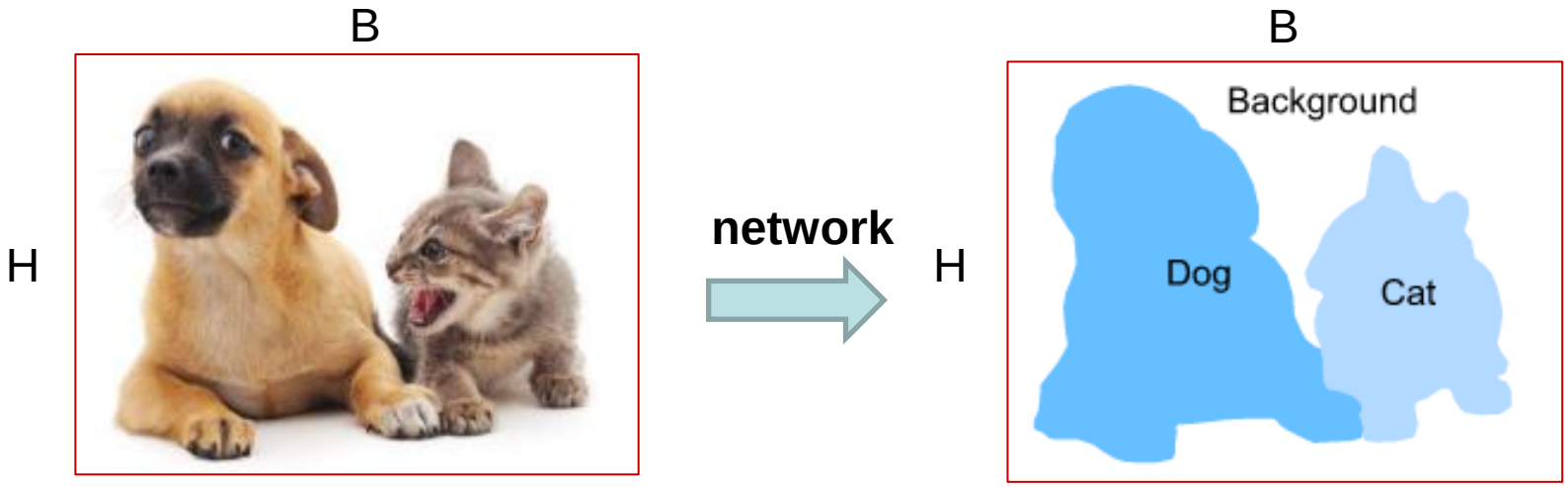
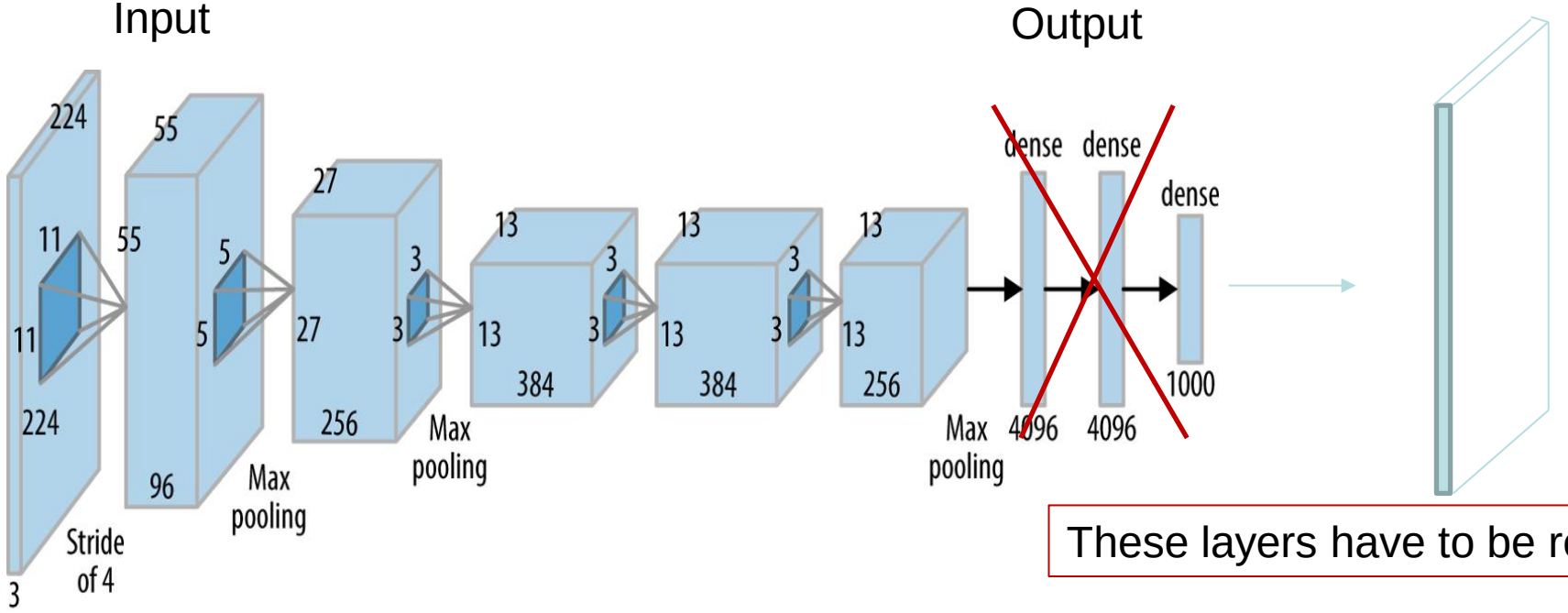


Image source: https://www.v7labs.com/blog/instance-segmentation-guide

Semantic Segmenation: Idea



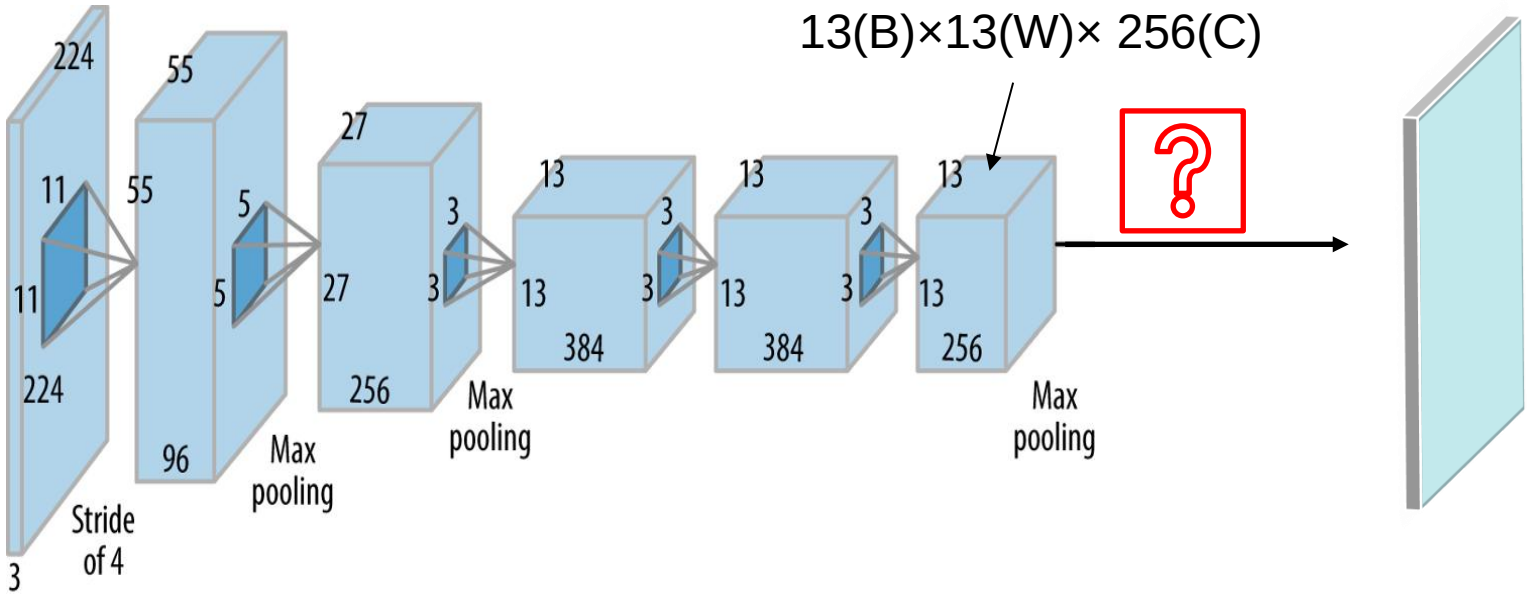


These layers have to be re-designed!

Semantic Segmenation: Upsampling

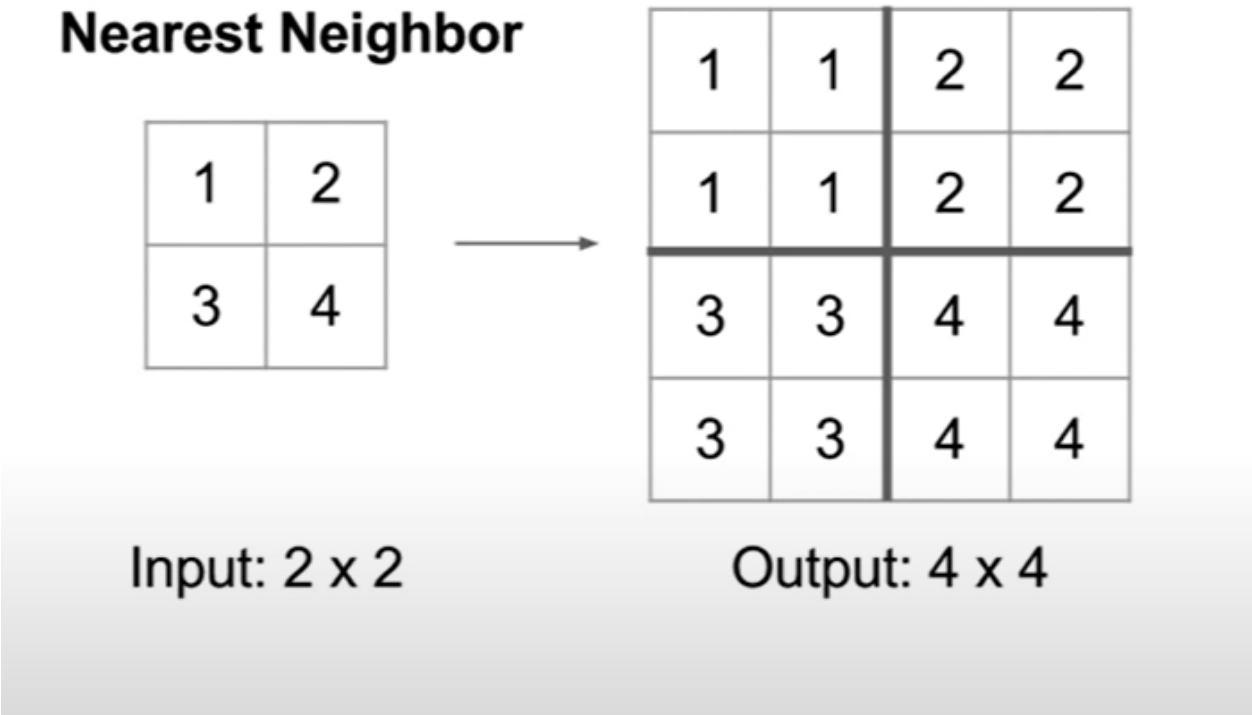
Input: 224(B)×224(W) × 3(C)

Output: 224(B)×224(W)×N



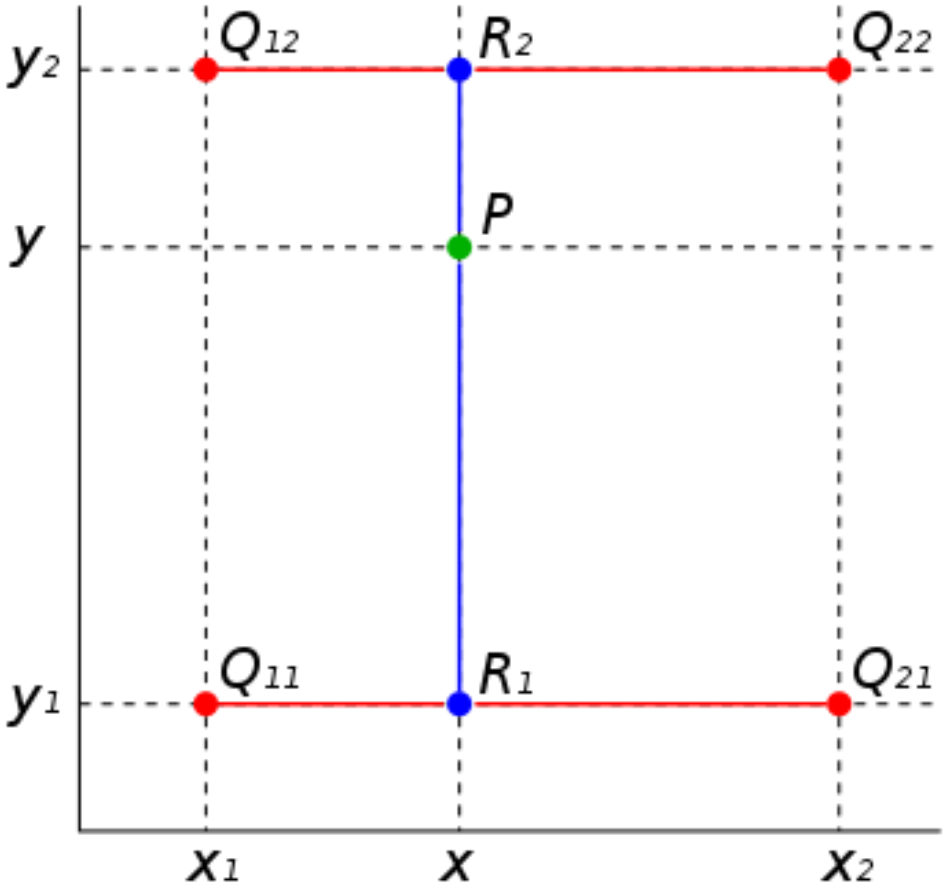
We need layers, which can upsample from 13×13 to 224×224.

Upsampling by „Unpooling“



- No trainable parameters

Upsampling by Bilinear Interpolation

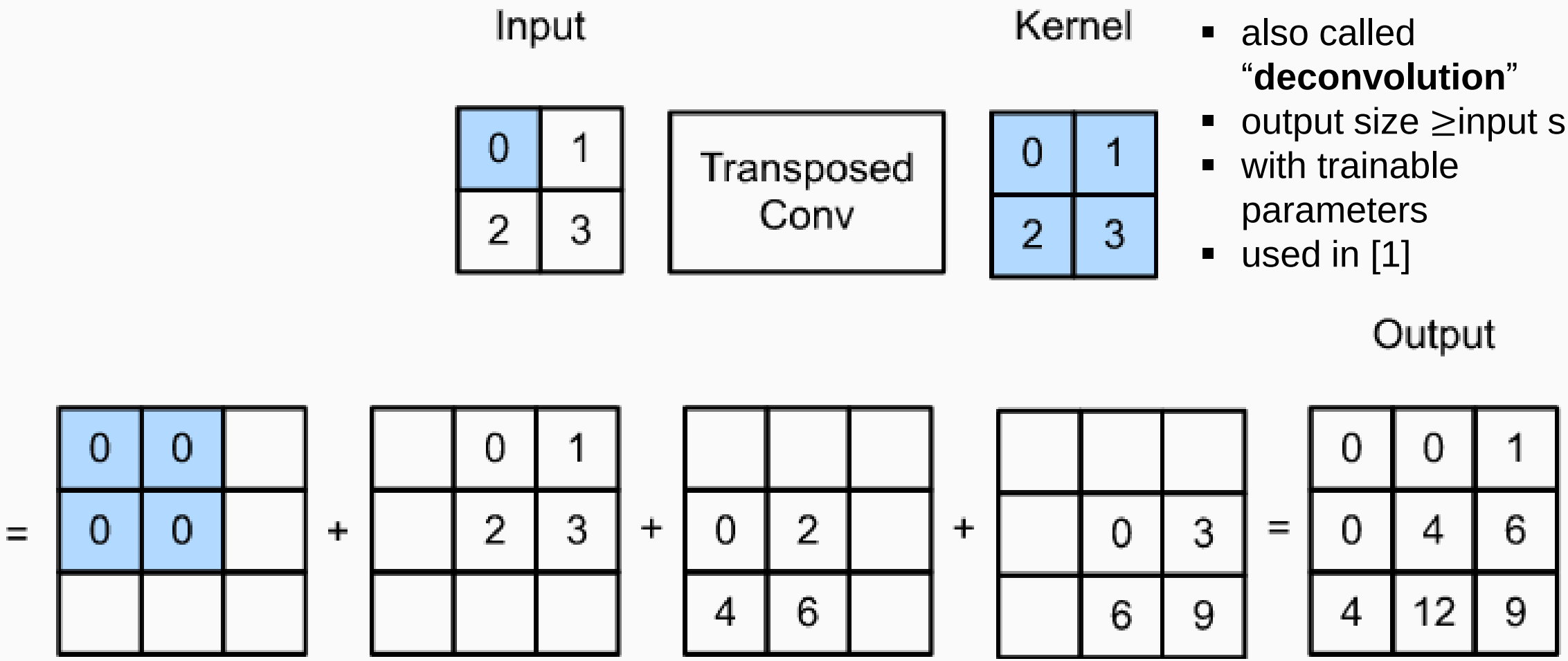


- Bilinear Interpolation: Given values of points Q_{11}, Q_{12}, Q_{21}, Q_{22}, what is the value of point P?
- No trainable parameters
- First tried in [1]

Image: https://en.wikipedia.org/wiki/Bilinear_interpolation

[1] Long, et al., Fully Convolutional Networks for Semantic Segmentation, CVPR, 2015

Upsampling by Transpose Convolution

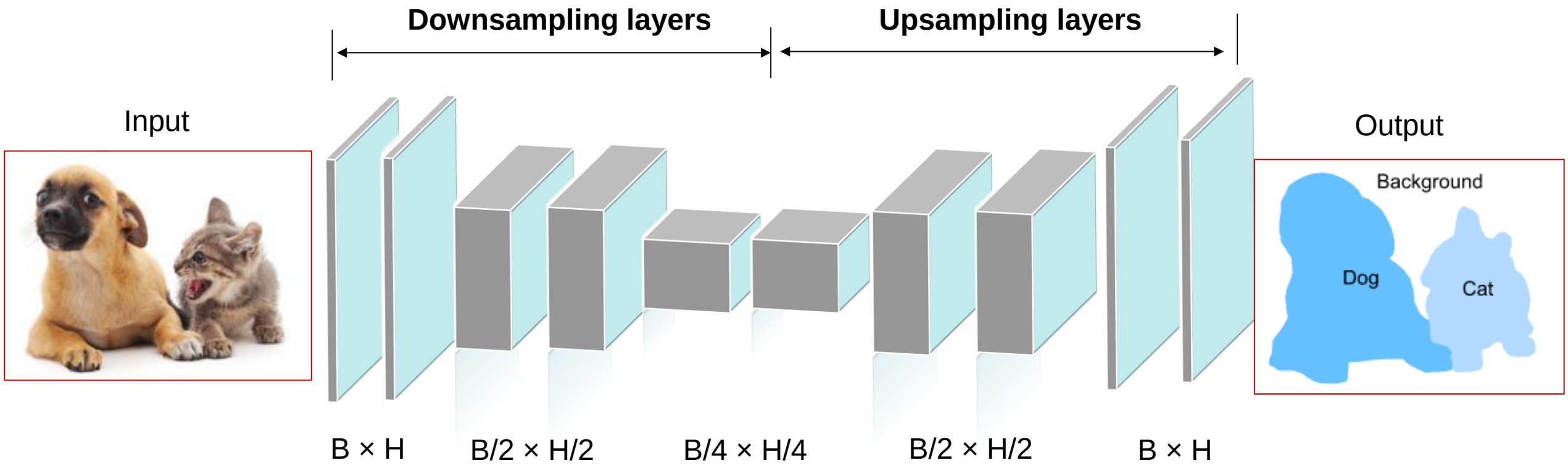


- also called "**deconvolution**"
- output size ≥input size
- with trainable parameters
- used in [1]

Image: https://d2l.ai/chapter_computer-vision/transposed-conv.html

[1] Noh, et al., Learning Deconvolution Network for Semantic Segmentation, ICCV, 2015

Semantic Segmenation: Fully Convolutional Network (FCN)



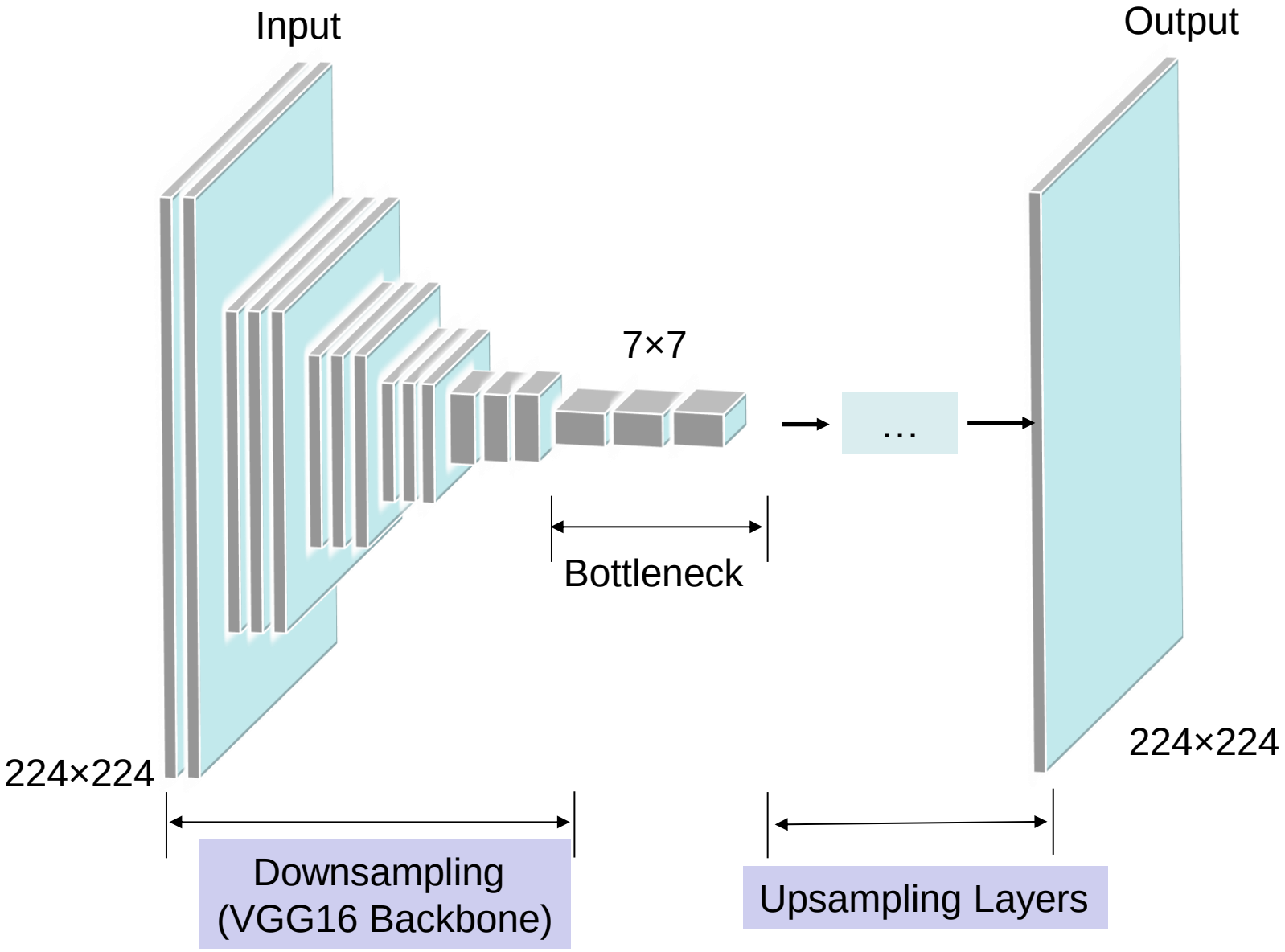
A breakthrough Method!

[1] Long, et al., Fully Convolutional Networks for Semantic Segmentation, CVPR, 2015
[2] Noh, et al., Learning Deconvolution Network for Semantic Segmentation, ICCV, 2015

Fully Convolutional Network (FCN): Comparison

	Classification Network, e.g. AlexNet, ResNet	**FCN**
Input	Image B*H*C	Image B*H*C
Network	Convolution + fully connected Layers	**Only convolutional Layers**
Output	1D vecter (1*1*N)	2D vecter (B*H*N)
Prediction Type	Sparse prediction	**Dense prediction**

Fully Convolutional Network (FCN): Feed-forward



Problems?

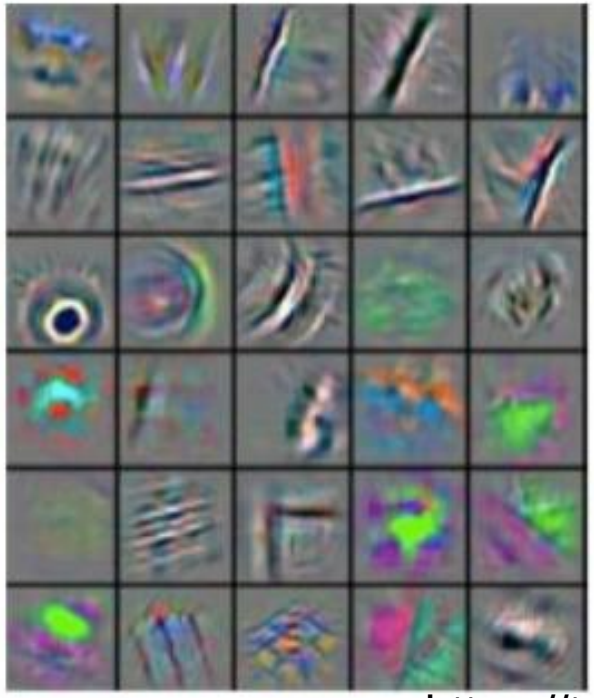
- Information has to go through bottleneck layers
- High-resolution output must be resolved from low-resolution bottleneck layers
- Coarse and blur output

Recall: Features at different layers for classification network

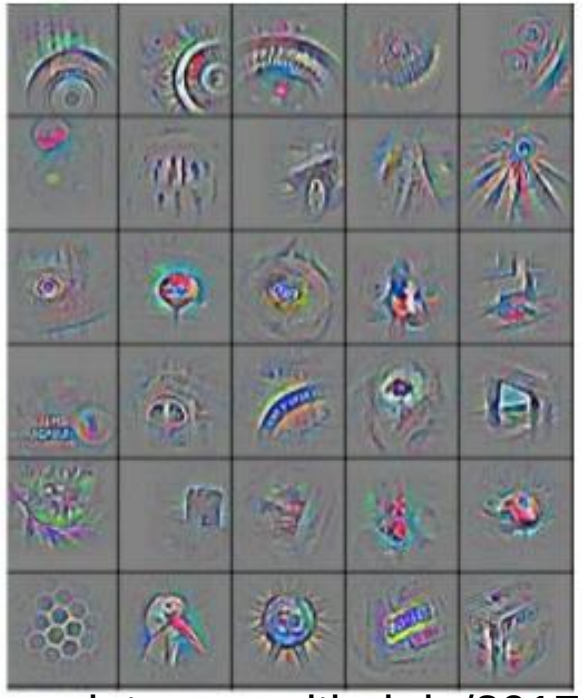
Low-level convolutions → Mid-level convolutions → High-level convolutions → classifier

low-level features

mid-level features



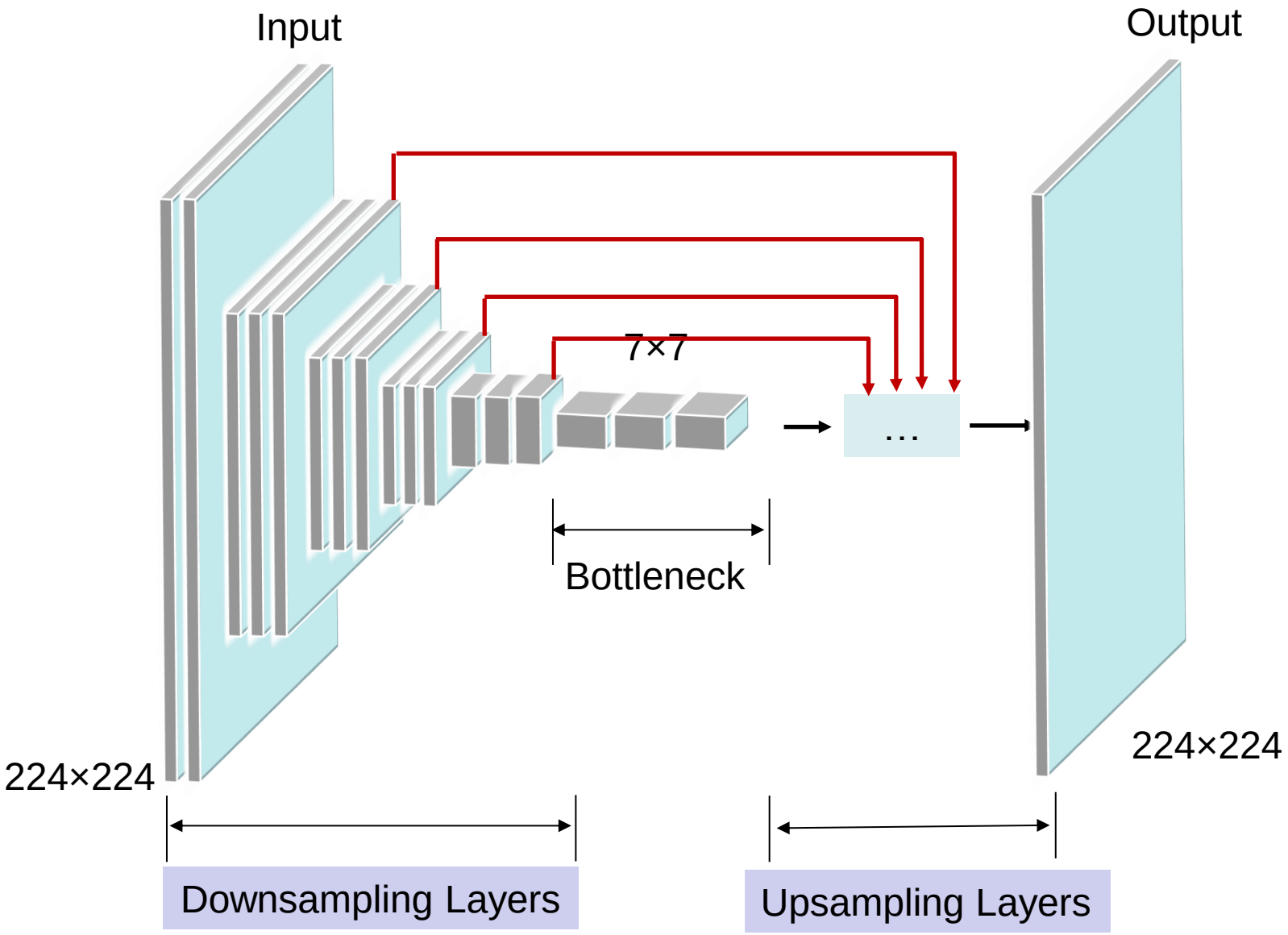
high-level features



- Layers at different levels correspond to features with different scales

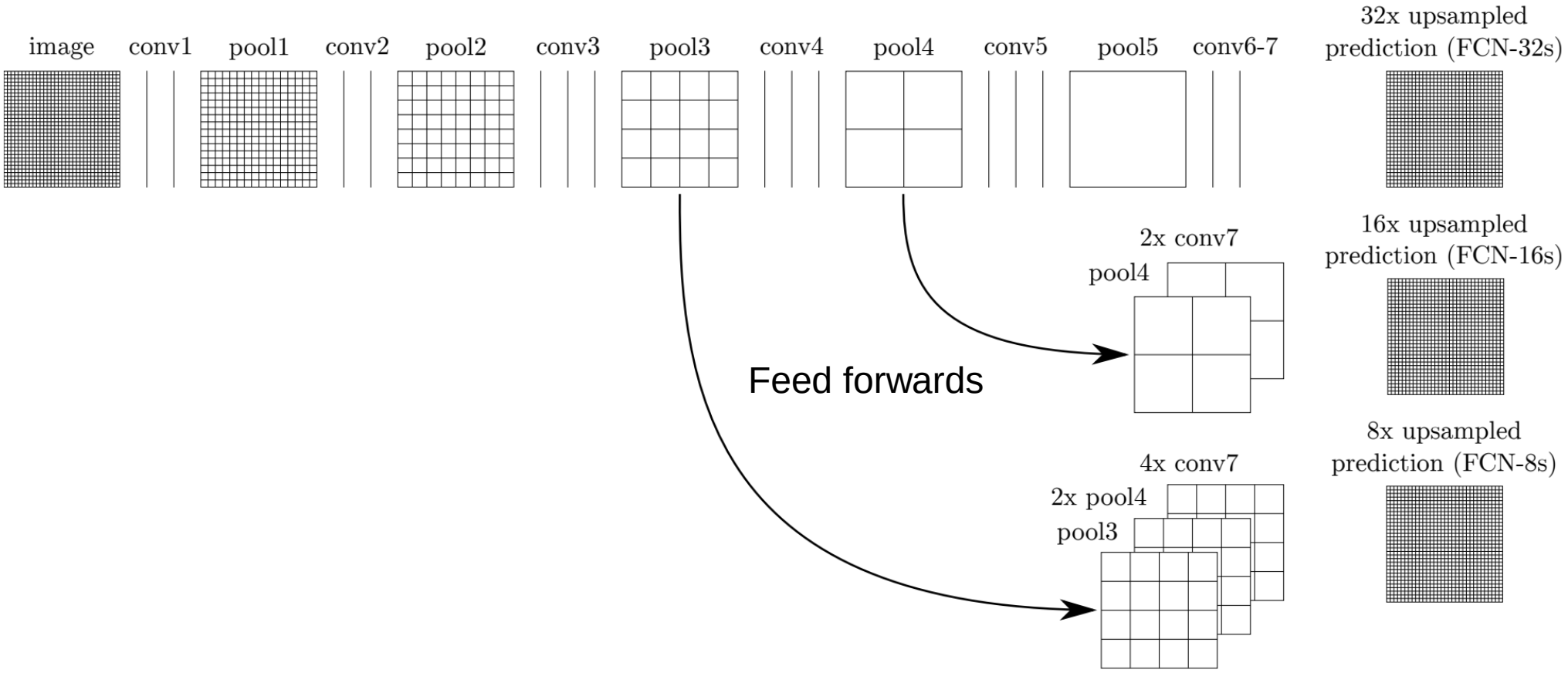
https://twopointseven.github.io/2017-10-29/cnn/

Fully Convolutional Network (FCN): Feed forward



- High-resolution output needs small-scale features
- Idea: Feed features from previous layers to upsampling layers

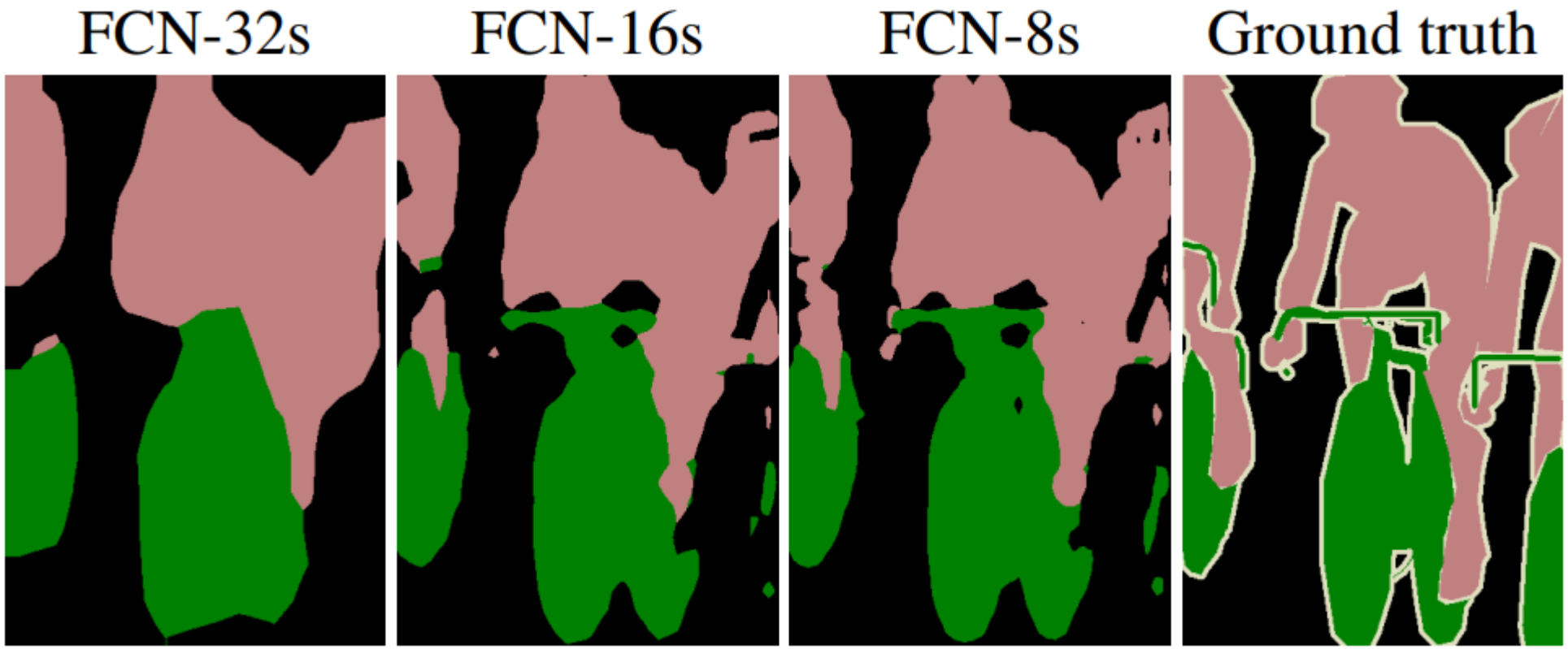
FCN: Proposed Structure in [1]



- This net learns to combine coarse, high layer information with fine, low layer information[1]
- Up-poolings are applied to different layers

[1] Long, et al., Fully Convolutional Networks for Semantic Segmentation, CVPR, 2015

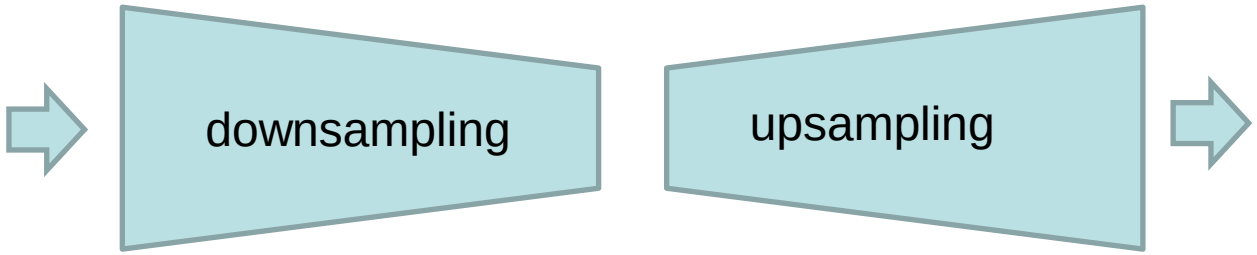
FCN: Results



- Refining FCN by fusing information from layers with different strides improves segmentaion detail.

FCN: Summery

- Dense prediction
- Only convolution (no fully connected layers)
- Network contains a down-sampling phase and an upsampling phase
  - Down-sampling: max-pooling, stride
  - Upsampling: interpolation (no deconvolution)

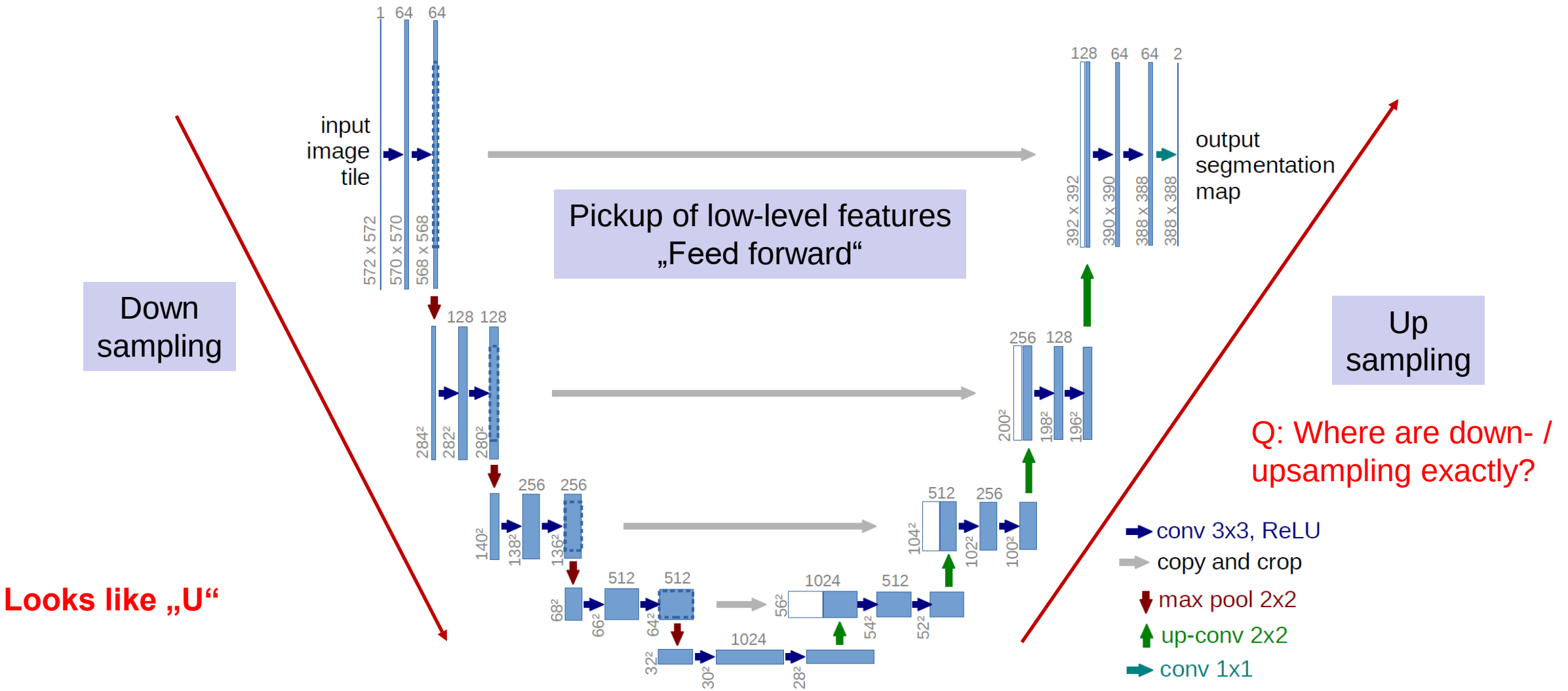


- Lower-level Features may be feed forward.

Content

- Semantic Segmentation
  - Fully convolutional network (FCN)
  - U-Net
  - DeepLab

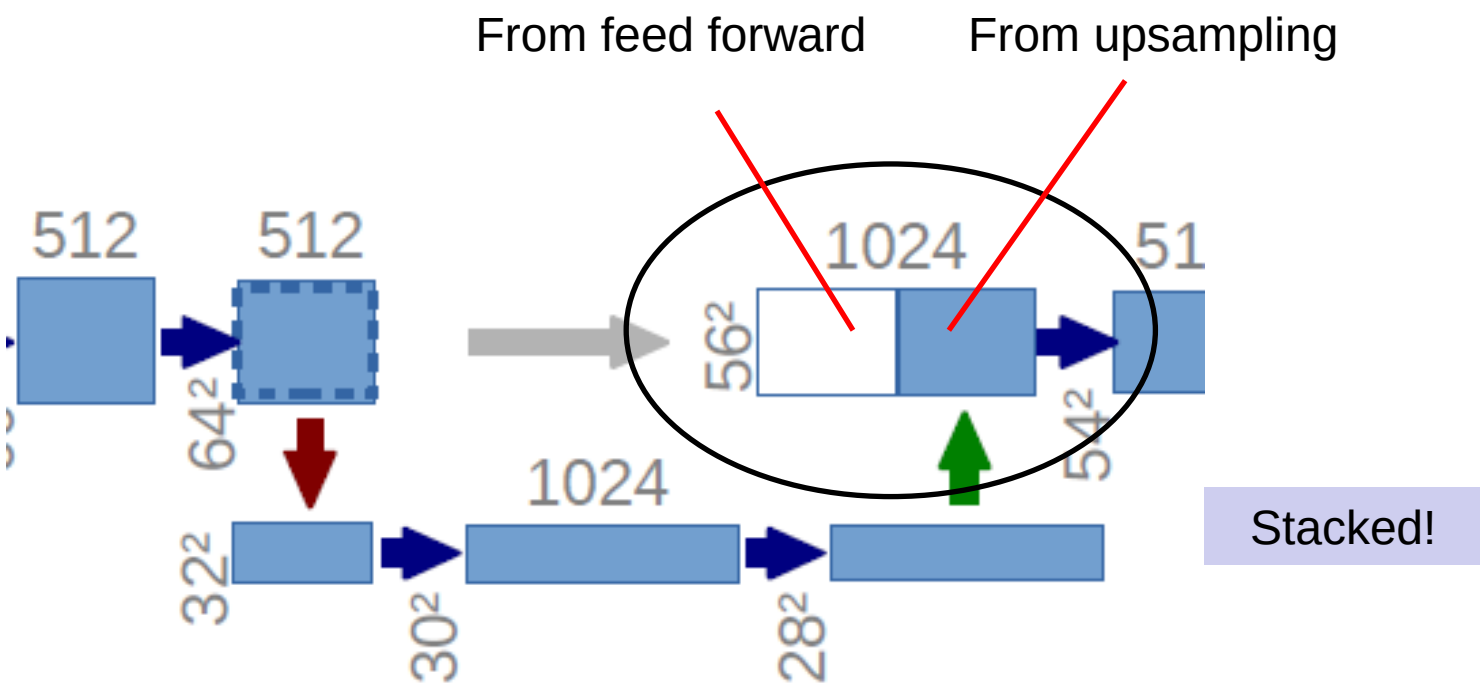
Semantic Segmenation: U-Net [1]



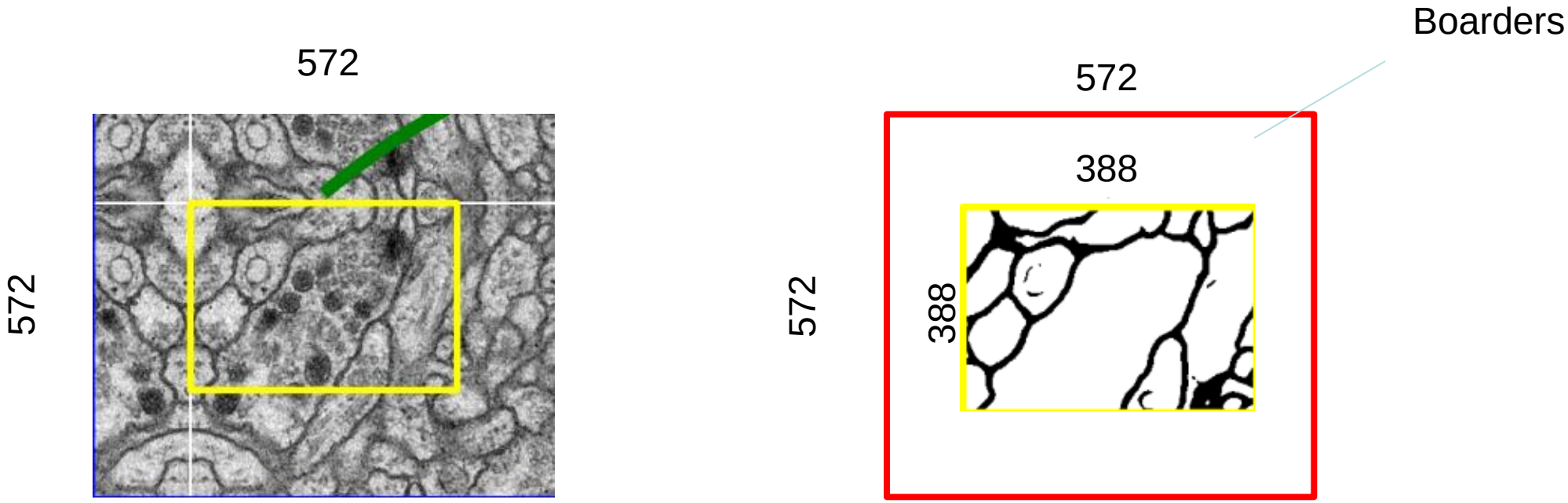
[1] Ronneberger, et al., U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI, 2015

U-Net: feature maps are stacked

- Lower level features are feed forward and stacked.



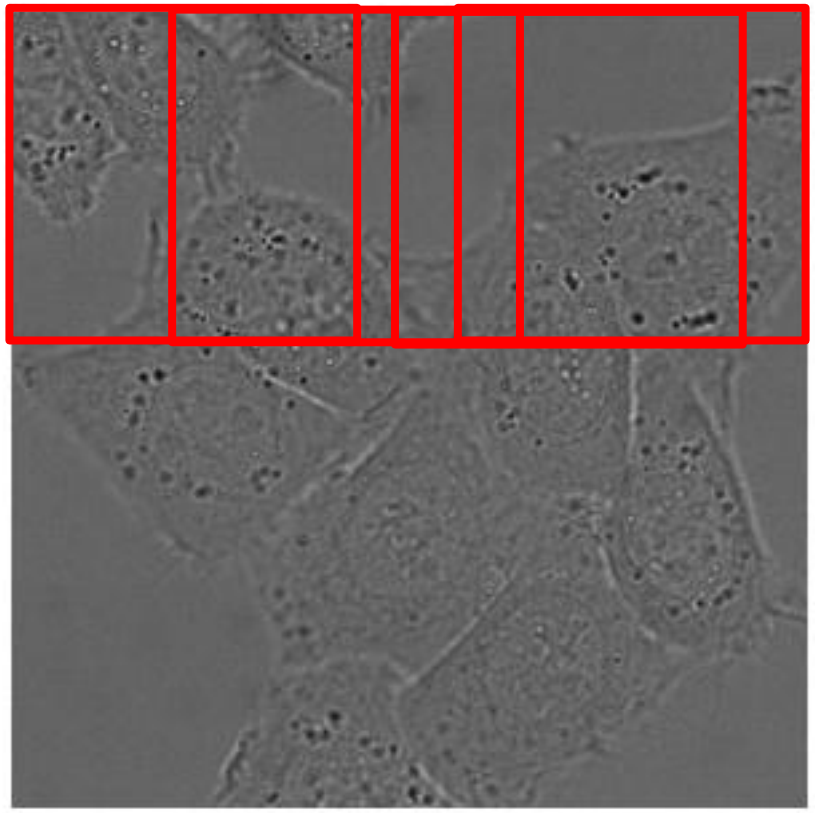
U-Net: Output size < Input size (no padding in conv. layers)



- No padding are used in conv. layers
  - Feature size is reduced after each conv. layer
  - Final feature map is smaller than the input image
- **Borders of original image can not be segmented**

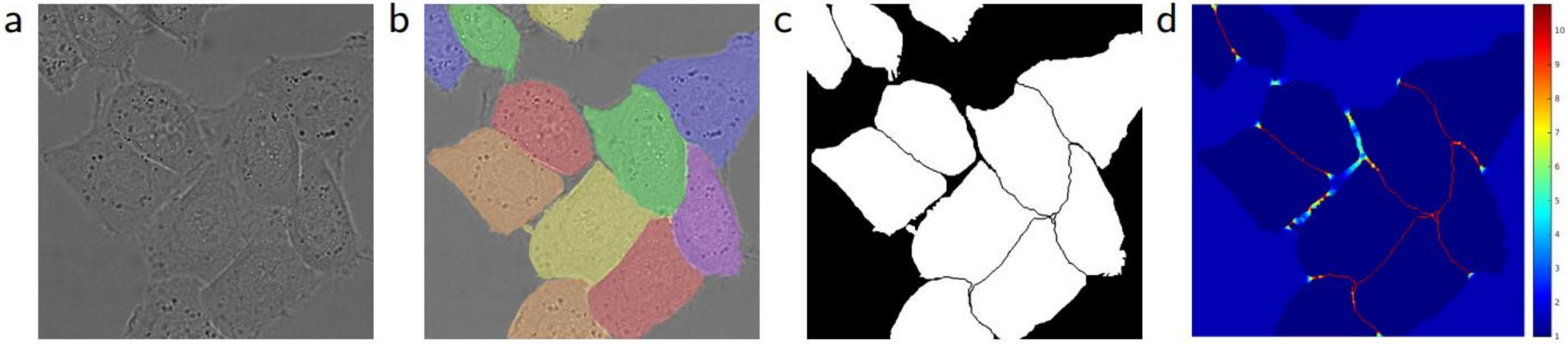
Q: What is the advantage?

U-Net: apply on large images



- What if input image is bigger than 572*572?
  - Apply U-Net repeatedly to its patches.
  - Assemble the results
  - No need to rescale the input image

U-Net: Result [6]



a. Original image

b. Overlay with ground truth segmentation

c. Generated segmentation mask

d. Map with a pixel-wise loss weight to force the network to learn the border pixels

[6] Ronneberger, et al., U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI, 2015

Content

- Introduction
- Semantic Segmentation
  - Fully convolutional network (FCN)
  - U-Net
  - DeepLab

DeepLab: Outline

- New Technologies:
  - **Dilated Convolution**:
    - Also called „Atrous convolution“
  - **Conditional random field (CRF)**
  - **Atrous Spatial Pyramid Pooling (ASPP)**
- DeepLab v1 [1]: Dilated Convolution + CRF
- DeepLab v2 [2]: Dilated Convolution + CRF + ASPP

[1] Chen, et al., Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs, ICLR, 2015
[2] Chen, et al., DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, TPAMI, 2017

DeepLab v1: Outline

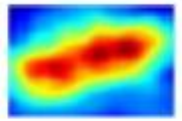
- „Dilated convolution“
- Fully convolutional

Input

Deep Convolutional Neural Network

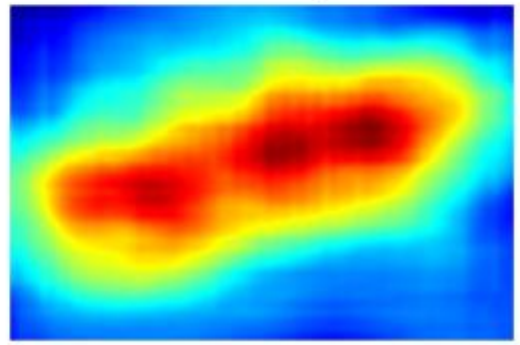
$\times \frac{1}{8}$

Aeroplane Coarse Score map

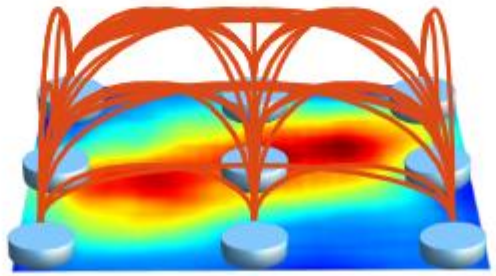


$\times 8$

Bi-linear Interpolation



Fully Connected CRF



Post-Processing

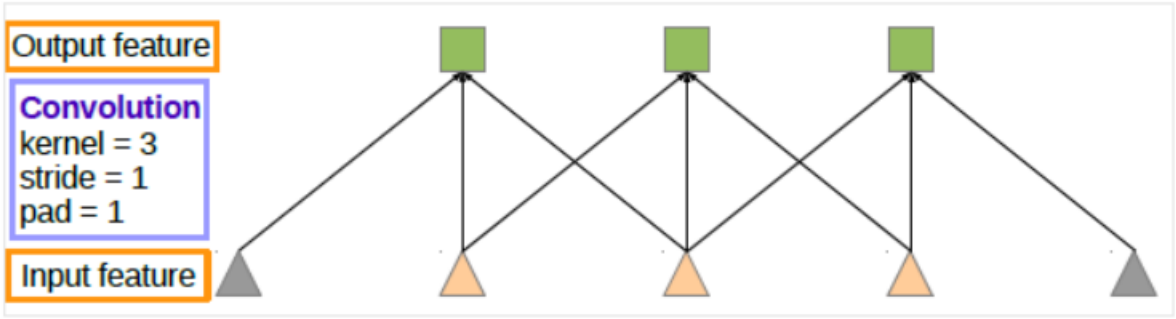
Final Output

No deconvolution applied!

Dilated convolution

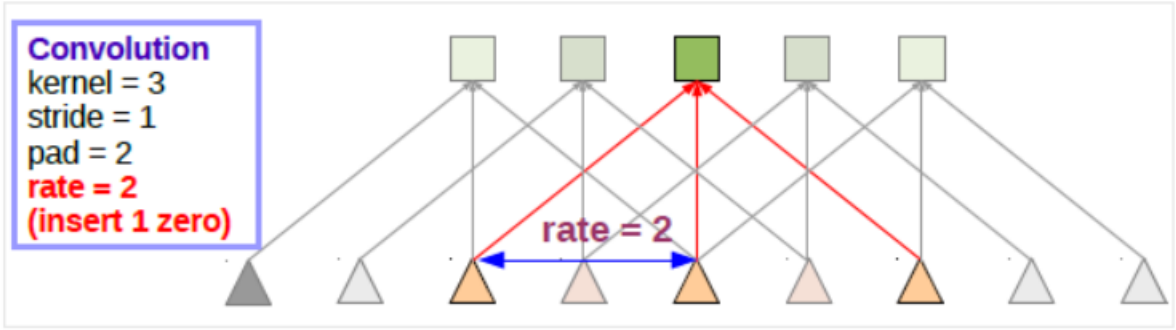
Dilated convolution (1D)

„Standard“ convolution:



(a) Sparse feature extraction

Dilated convolution:



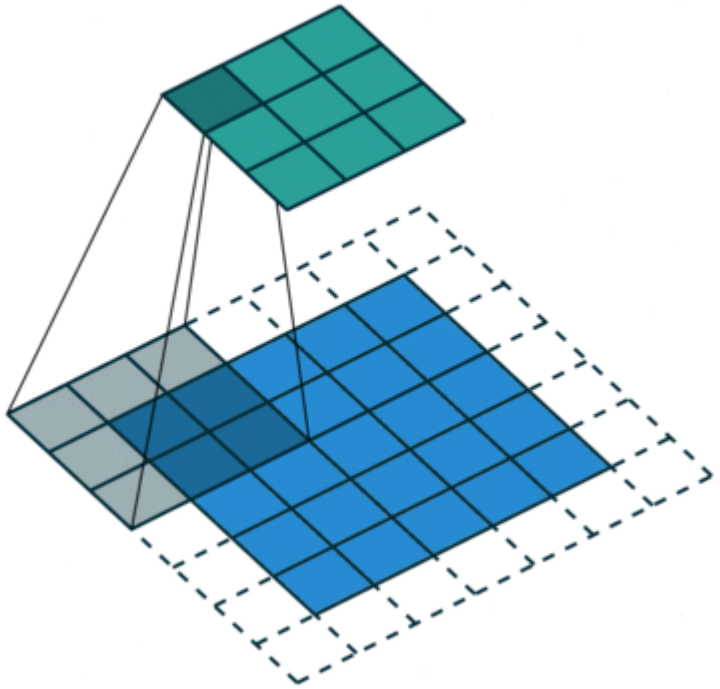
(b) Dense feature extraction

[7] Chen, et al., Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs, ICLR, 2015

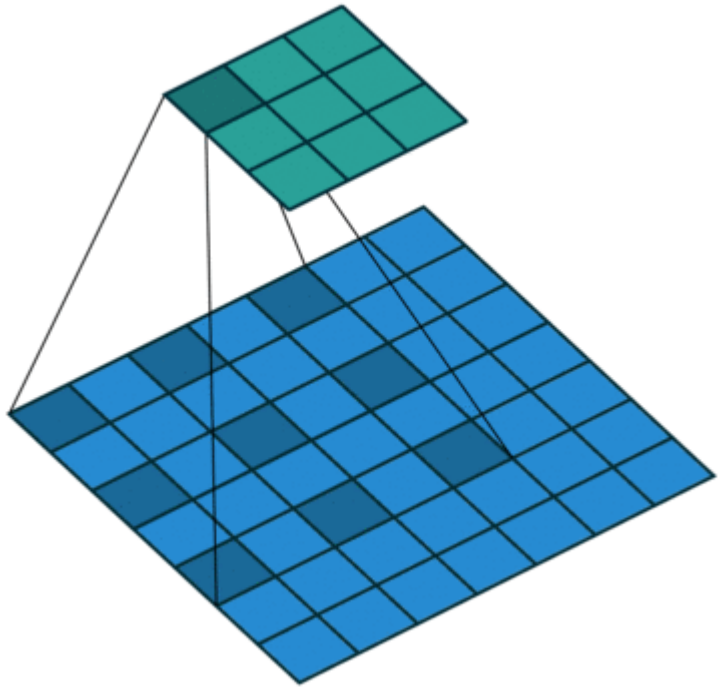
Dilated convolution (2D)

Standard convolution

Kernel: 3
Stride: 1
Pad: 1
(Rate: 1)



Dilated convolution



Kernel: 3
Stride: 1
Pad: 0
Rate: 2

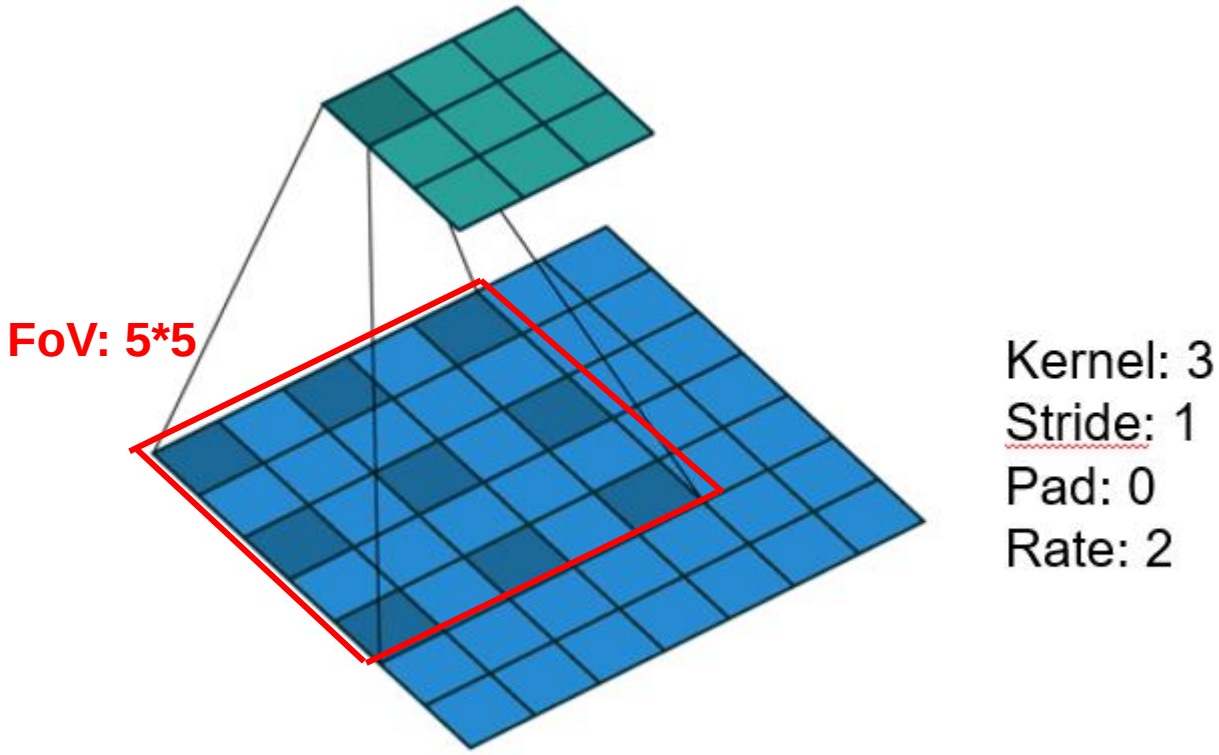
- Also called „Atrous convolution“
- Atrous (in French): with holes

➢ Convolution with holes

Image: https://towardsdatascience.com/review-dilated-convolution-semantic-segmentation-9d5a5bd768f5

Why dilated convolution?

- Compared to standard convolution, dilated convolution **increases FoV without increasing the number of parameters**, namely kernel size
- FoV ↑ -> larger objects can be recognized



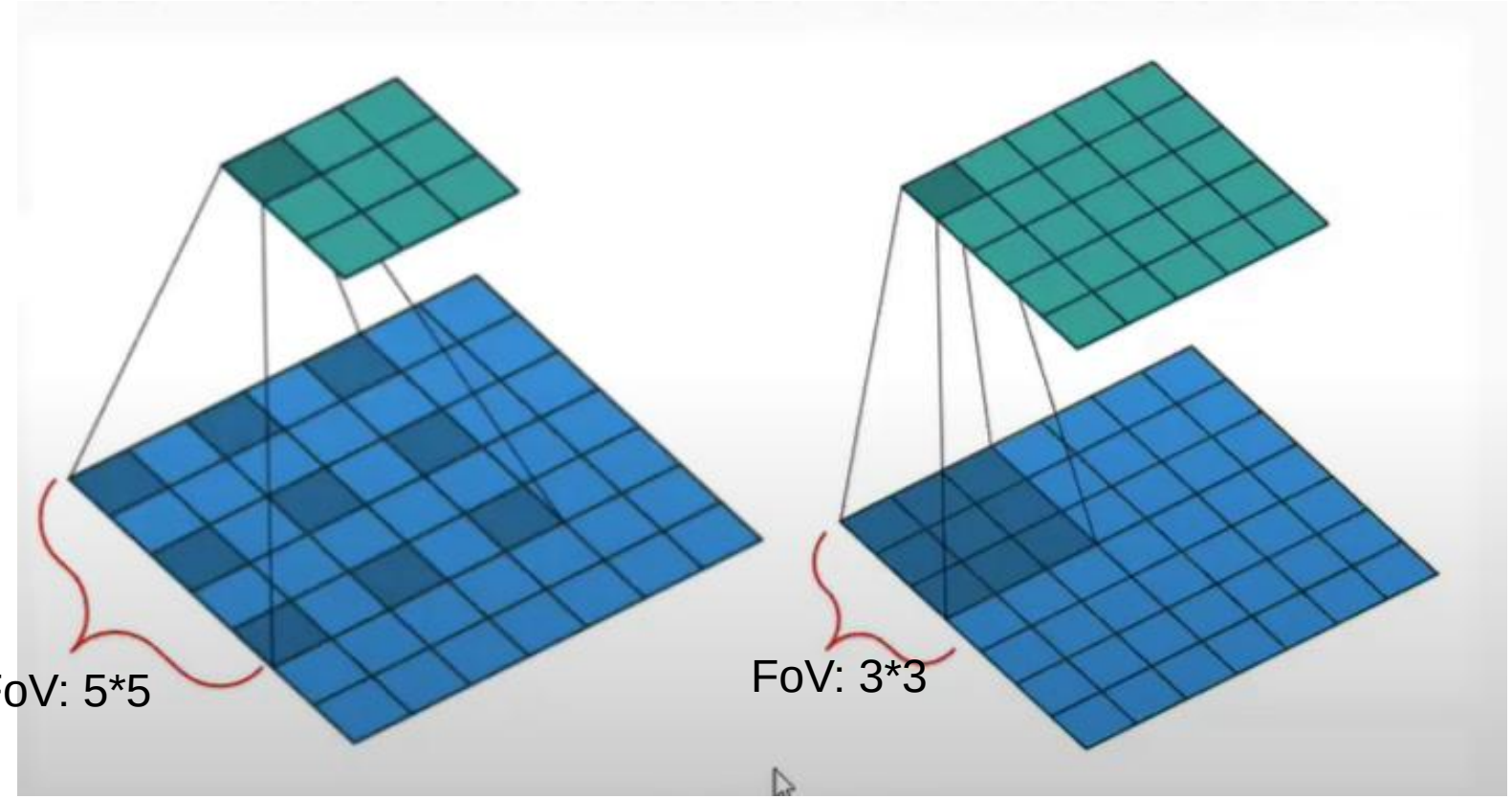
How to increase FoV?

- Add more layers
- Increase kernel size
- E.g. Dilated convolution

Dilated convolution: Increase „Field-of-View“

Dilated convolution

Standard convolution



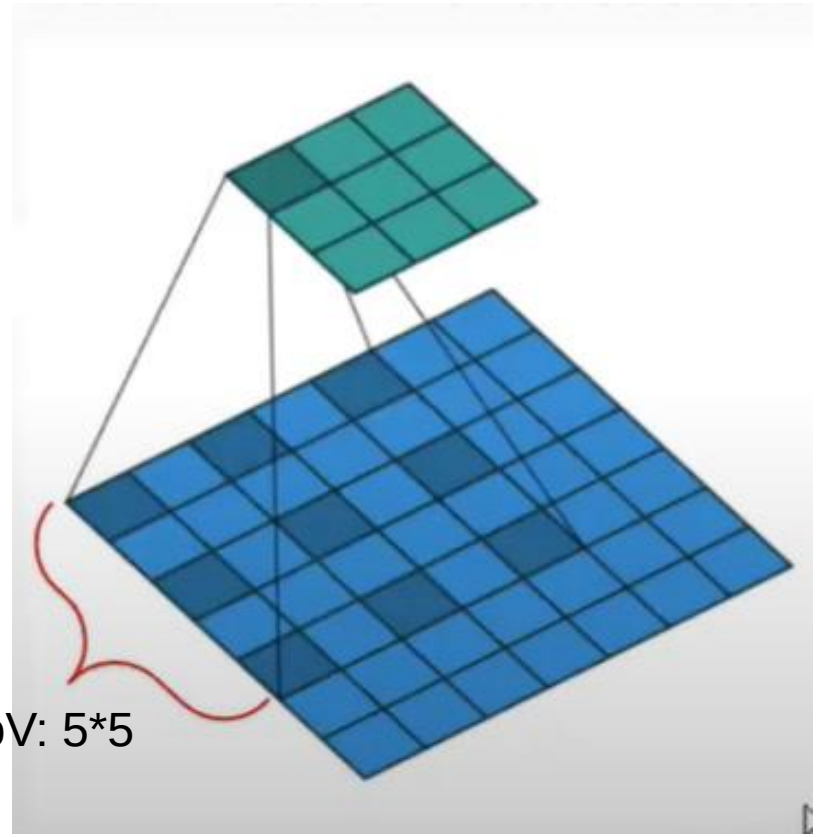
Kernel: 3*3
Stride: 1
Pad: 0
Rate: 2

Kernel: 3*3
Stride: 1
Pad: 0

- Dilated convolution increases “Field-of-View”
- Stacking multiple dilated convolution layers leads to enlarged FoV for multiple times

Dilated convolution: Increase „Field-of-View“

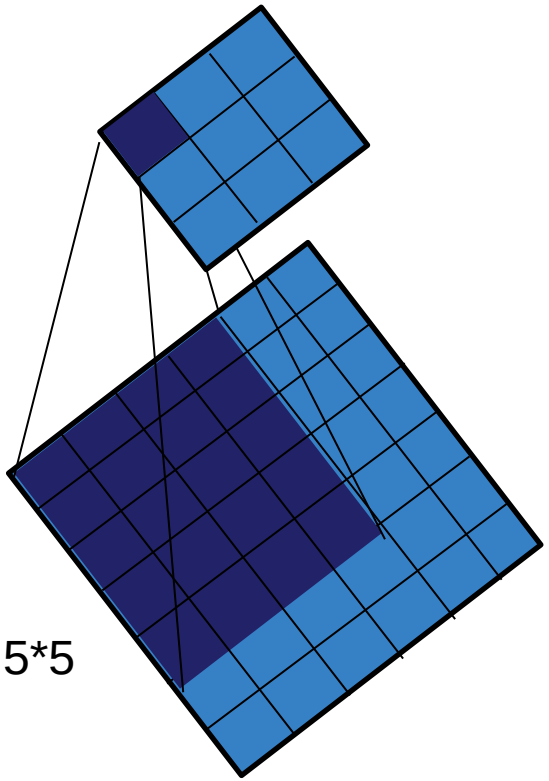
Dilated convolution



FoV: 5*5

Kernel: 3*3
Stride: 1
Pad: 0
Rate: 2

Standard convolution



FoV: 5*5

Kernel: 5*5
Stride: 1
Pad: 0

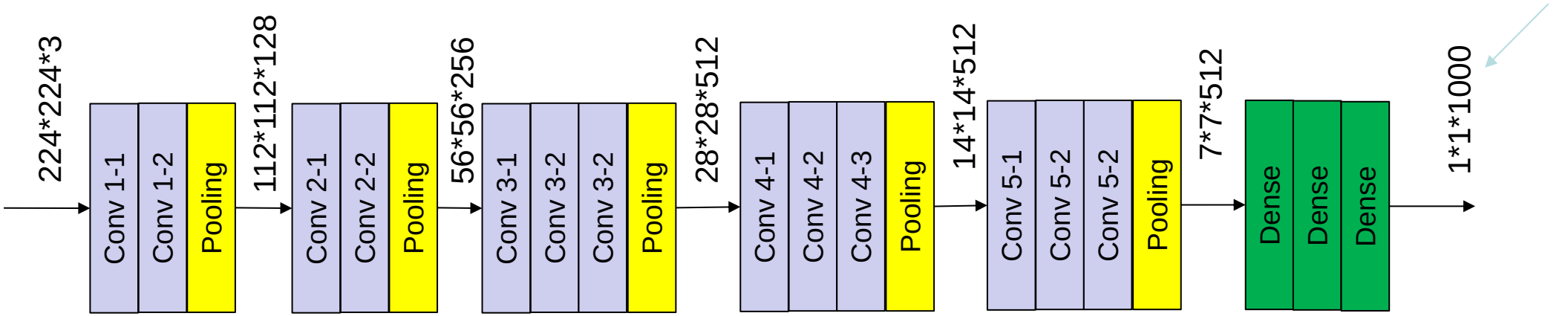
- To obtain the same FoV, dilated convolution needs smaller kernel size

➢ Less parameters needed

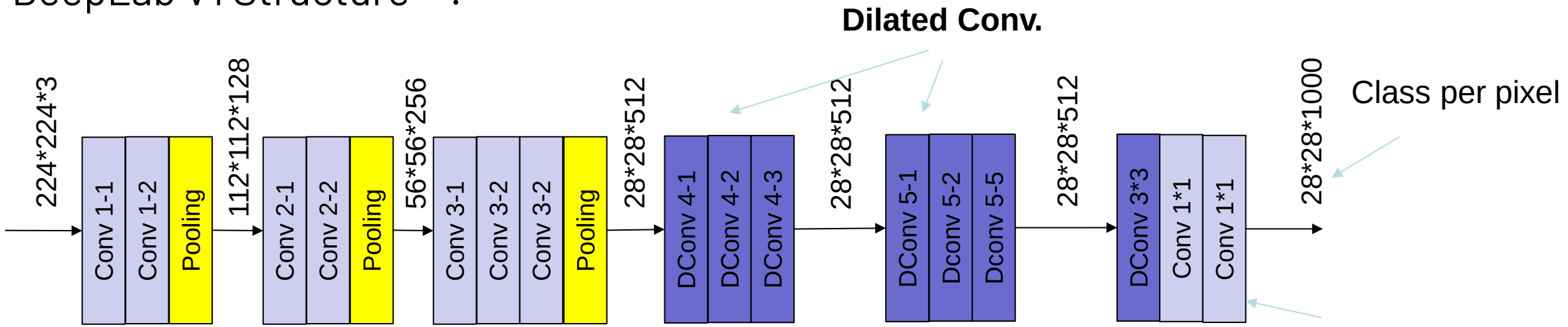
➢ Less training time

DeepLab V1: Network Structure

- Start from a VGG-16 network:



- DeepLab V1 Structure[*]:



* Exact structure is not shown in orig. paper. This is the interpreted results from the text of the orig. paper.

Conditional Random Fields (CRF)

DeepLab V1: Outline

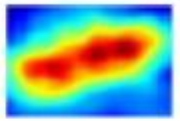
- „Dilated convolution“
- Fully convolutional

Input

Deep Convolutional Neural Network

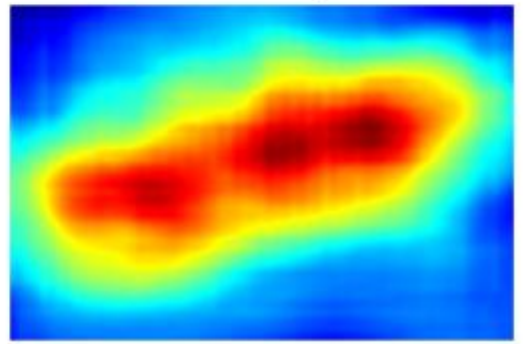
$\times \frac{1}{8}$

Aeroplane Coarse Score map

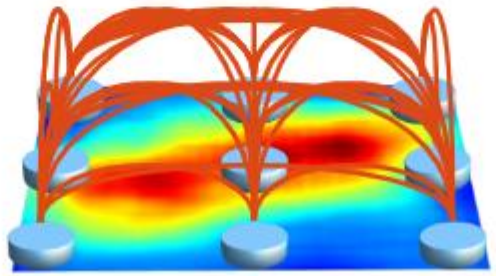


$\times 8$

Bi-linear Interpolation



Fully Connected CRF



Post-Processing

Final Output

No deconvolution is applied this step!

Fully-Connected Conditional Random Fields (CRF)

- Motivation:
  - **a post-processing step**
  - Smoothing for better local consistency (e.g. avoid coarse boundary)
  - Avoid sharp changes in the image

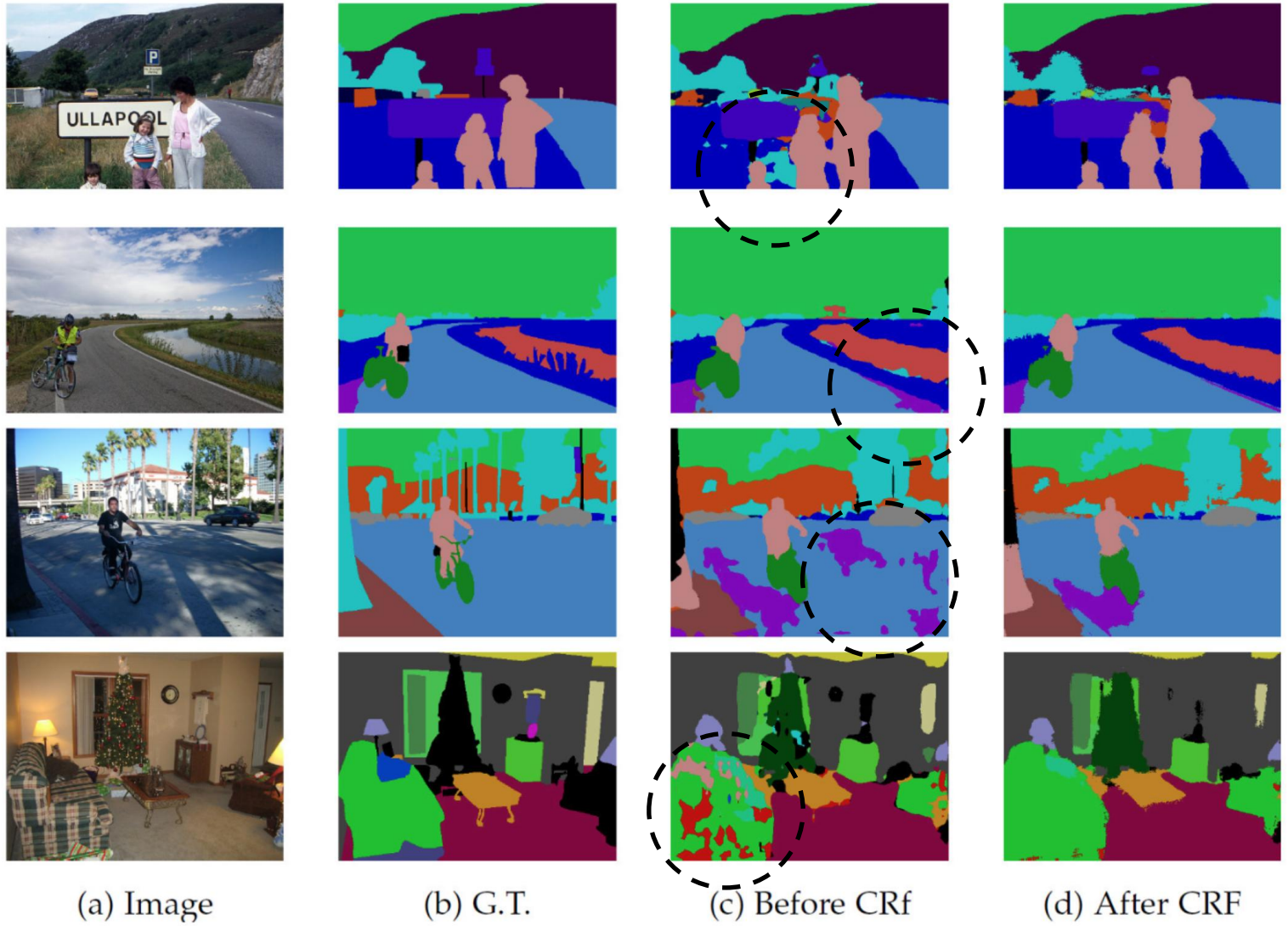
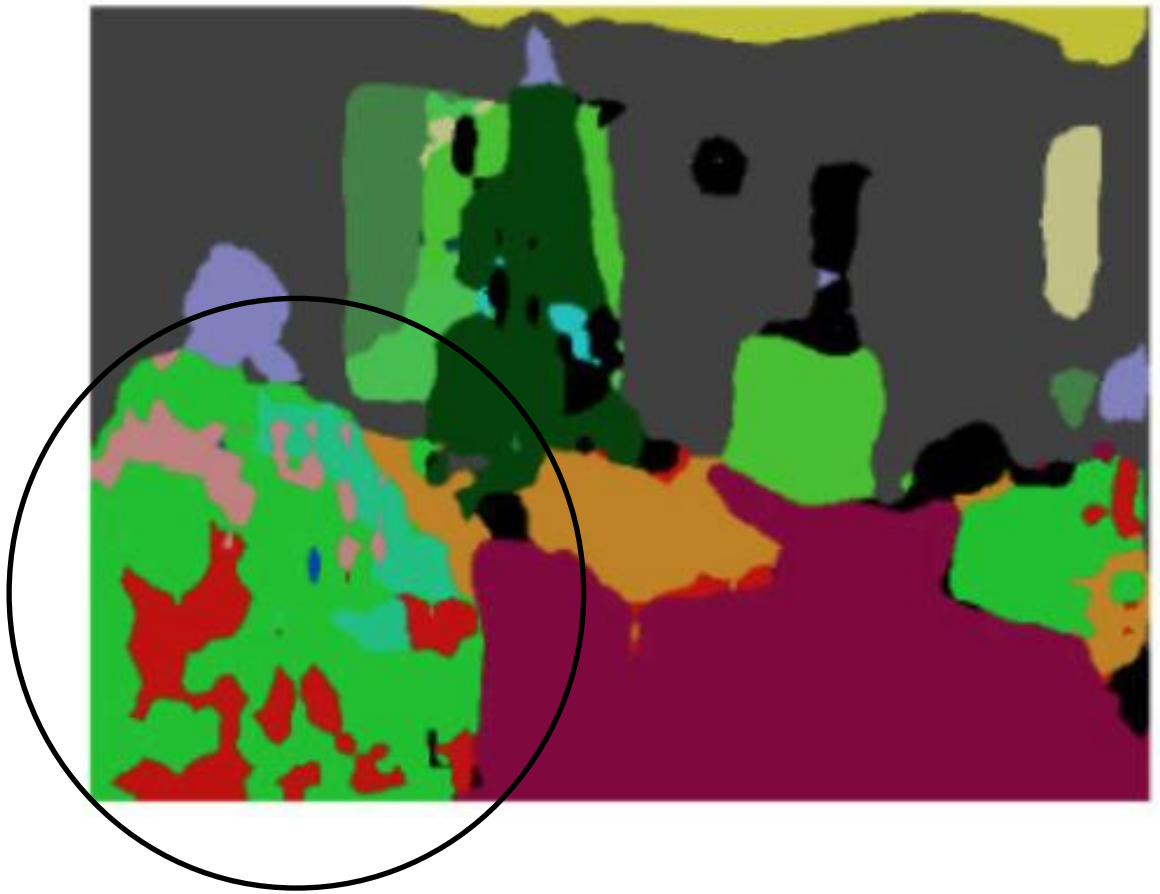


Image from [8]

CRF: Motivation

- Output image of segmentation contains **coarse boundary** or “**island**".
- Why? Each pixel are predicted „**independently**“. Their relationship are not modeled explicitly. In other words, it is assumed that the predictions of different pixels are independent.
- However, for certain cases, it is more likely that a black pixel is surrounded by black or grey pixels. It is less likely that a black pixel is surrounded by white pixels.
- If such correlations among pixels is strong, the standard NN tends to underperform.

Segmentation without CRF



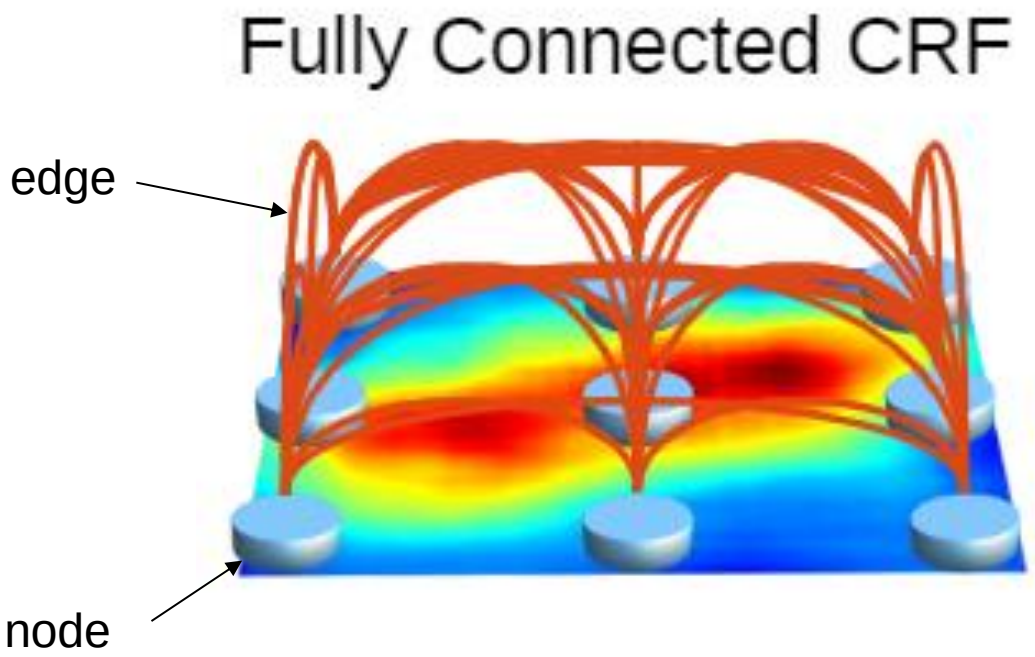
We can use a model to model their dependences!

CRF for Semantic Segmentation in 2D

- Semantic Segmentation, e.g. U-Net:
  - Dense prediction: each pixel is classified
  - Classes for each pixel are "independently" predicted.
- Motivation:
  - Use a model to model the dependence of each pixels.
  - Which model? Fully-connected conditional random field.

Label classes as fully-connected CRF

- An image of label classes, e. g. the output classes of U-Net, $y^{i,j}|x^{i,j}$, i=1, ...,M, j=1, ..., N, can be modeled by a fully-connected CRF



- Assume that this image only has 3*3 pixels, namely 9 predicted classes
- In this case, the CRF has 9 nodes and C_9^2 edges

Fully-connected CRF for Semantic Segmenation

- i$\in \{1, \dots, MN\}$: index of pixels
- M, N: width, heigh of image, K: total number of classes
- P(y_i=1), P(y_i =2), …, P(y_i =K): Possilibty distribution (network's output) at pixel i
- $\boldsymbol{x}=(x_1, \dots, x_{MN})$:
  - „Job" of CRF : class assignment for each pixel by CRF
  - $x_i \in \{1, \dots, K\}$
- P($y_i = x_i$): possibility that pixel i belongs to class x_i
- **Unary Potential**: $\boldsymbol{\theta_i(x_i) := -\log P(y_i = x_i)}$

„penalty for disregarding the network's prediction"

Chen, et al., DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, TPAMI, 2017

Pairwise potential

- $\theta_{i,j}\,(x_i, x_j)$: pairwise potential (encourage smooth annotations)

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j)\left[w_1 \exp\left(-\frac{||p_i - p_j||^2}{2\sigma_\alpha^2} - \frac{||I_i - I_j||^2}{2\sigma_\beta^2}\right) + w_2 \exp\left(-\frac{||p_i - p_j||^2}{2\sigma_\gamma^2}\right)\right]$$

  - $\mu(x_i, x_j) = 1$, if $x_i \neq x_j$. Otherwise, $\mu(x_i, x_j) = 0$
  - p_i: pixel position
  - I_i: RGB/intensity values
  - $w_1, w_2, \sigma_\alpha, \sigma_\beta, \sigma_\gamma$: hyper parameters

Pairwise potential

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j)\left[w_1 \exp\left(-\frac{||p_i - p_j||^2}{2\sigma_\alpha^2} - \frac{||I_i - I_j||^2}{2\sigma_\beta^2}\right) + w_2 \exp\left(-\frac{||p_i - p_j||^2}{2\sigma_\gamma^2}\right)\right]$$

- Only for different labels, $x_i \neq x_j$, the positional and intensity penalties contribute to E(x)
- If the positions of two pixels (p_i and p_j) get closer, $\theta_{i,j}$ gets larger
- If the color intensity of two pixels (I_i and I_j) get similar, $\theta_{i,j}$ gets larger
- In this way, by $\min_x \theta_{i,j}(x_i, x_j)$: we enforce the predicted classes $x = (x_1, \ldots, x_{MN})$ to be selected such that: If they locate closer and have the similar color/intensity, they should tend to have the same class assignment.

Energy function

- Fully-connected CRF minimizes the following energy function:

$$E(\boldsymbol{x}) = \sum_i \theta_i(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j) \qquad \Rightarrow \min_{x \epsilon R^{MN}} E(x)$$

- Solving this optimization is challenging. But it can be approximately solved in an efficient way (not covered in this lecture).

CRF: Beispiel

Bild

255	255	255
255	0	255
255	255	255

$P(y_i = 1)$

0.8	0.8	0.8
0.9	0.1	0.8
0.9	0.9	0.9

Classifiction result (no CRF)

1	1	1
1	0	1
1	1	1

$y_i = 1: object$
$y_i = 0: background$

- Now, for pairwise potential $\theta_{i,j}$,
  - set all parameters $w_1, w_2, \sigma_\alpha, \sigma_\beta, \sigma_\gamma$ as 1
  - use Chebyshev distance, i.e. for $v, w \in \mathbb{R}^m$, $||v - w|| := \max_{i=1,\dots,m} |v_i - w_i|$

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j)\left[w_1 \exp\left(-\frac{||p_i - p_j||^2}{2\sigma_\alpha^2} - \frac{||I_i - I_j||^2}{2\sigma_\beta^2}\right) + w_2 \exp\left(-\frac{||p_i - p_j||^2}{2\sigma_\gamma^2}\right)\right]$$

- For CRF, let us compare two classfication candidates x_A and x_B

x_A

1	1	1
1	0	1
1	1	1

same as before

$$\sum_{i=1}^{9} \theta_i = 4 * (-\log 0.8) + 4 * (-log 0.9) - \log(0.9) = 0.62$$

$$\sum_{i,j} \theta_{i,j} = 8 * \left[\exp\left(-\frac{1}{2} - \frac{255^2}{2}\right) + \exp\left(-\frac{1}{2}\right)\right] = 4.85$$

Why 8?

x_B

1	1	1
1	1	1
1	1	1

$$\sum_{i=1}^{9} \theta_i = 4 * (-\log 0.8) + 4 * (-log 0.9) - \log(0.1) = 1.57$$

$$\sum_{i,j} \theta_{i,j} = 0$$

Why 0?

Therefore:

- $E(x_A) > E(x_B)$
- CRF prefer x_B, compared with x_A
- In this case, CRF fill the “hole” in the normal classification result (no CRF)
- Note: Here we only compared two classification candidates x_A, x_B. Actually, $\min_x E(x)$ compares all possible candidates x and selects the best.

Normal classification result (no CRF)

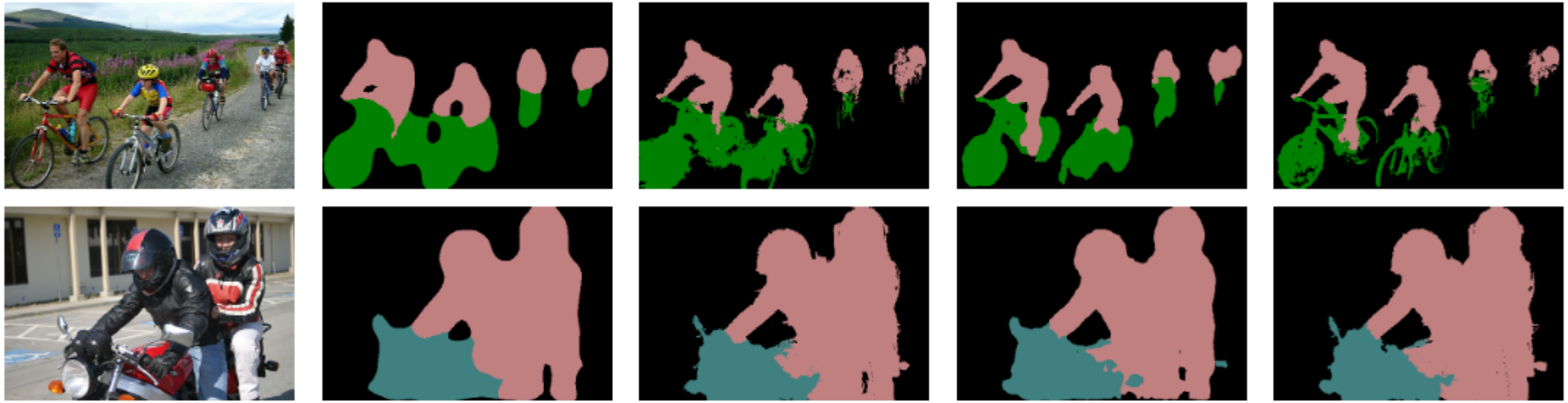
1	1	1
1	0	1
1	1	1

Classification result with CRF

1	1	1
1	1	1
1	1	1

if we only compare $x_A \; and \; x_B$

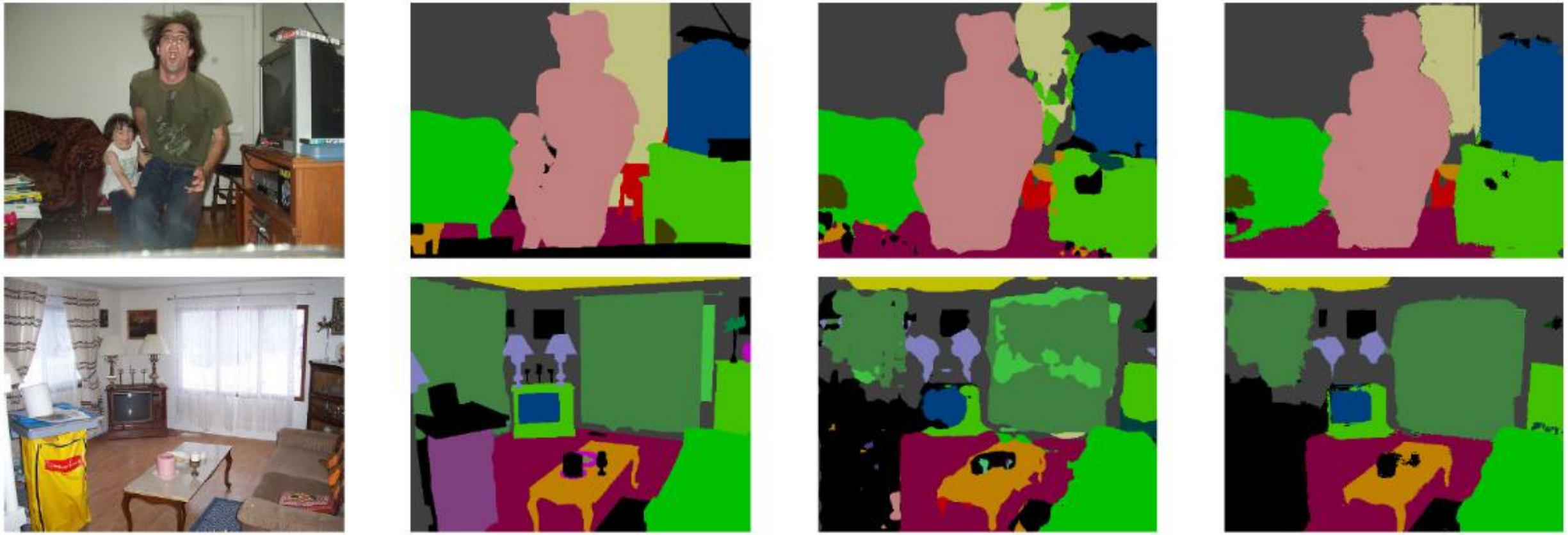
CRF: Examples



[8] Chen, et al., DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, TPAMI, 2017

CRF: Examples

Input | GT | Prediction (no CRF) | Prediction (with CRF)



[8] Chen, et al., DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, TPAMI, 2017

CRF: Performance

Method	before CRF	after CRF
LargeFOV	65.76	69.84
ASPP-S	66.98	69.73
ASPP-L	68.96	71.57

TABLE 3: Effect of ASPP on PASCAL VOC 2012 *val* set performance (mean IOU) for VGG-16 based DeepLab model. **LargeFOV**: single branch, $r = 12$. **ASPP-S**: four branches, $r = \{2, 4, 8, 12\}$. **ASPP-L**: four branches, $r = \{6, 12, 18, 24\}$.

[8] Chen, et al., DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, TPAMI, 2017

Fully-connected CRF in DeepLab: Summary

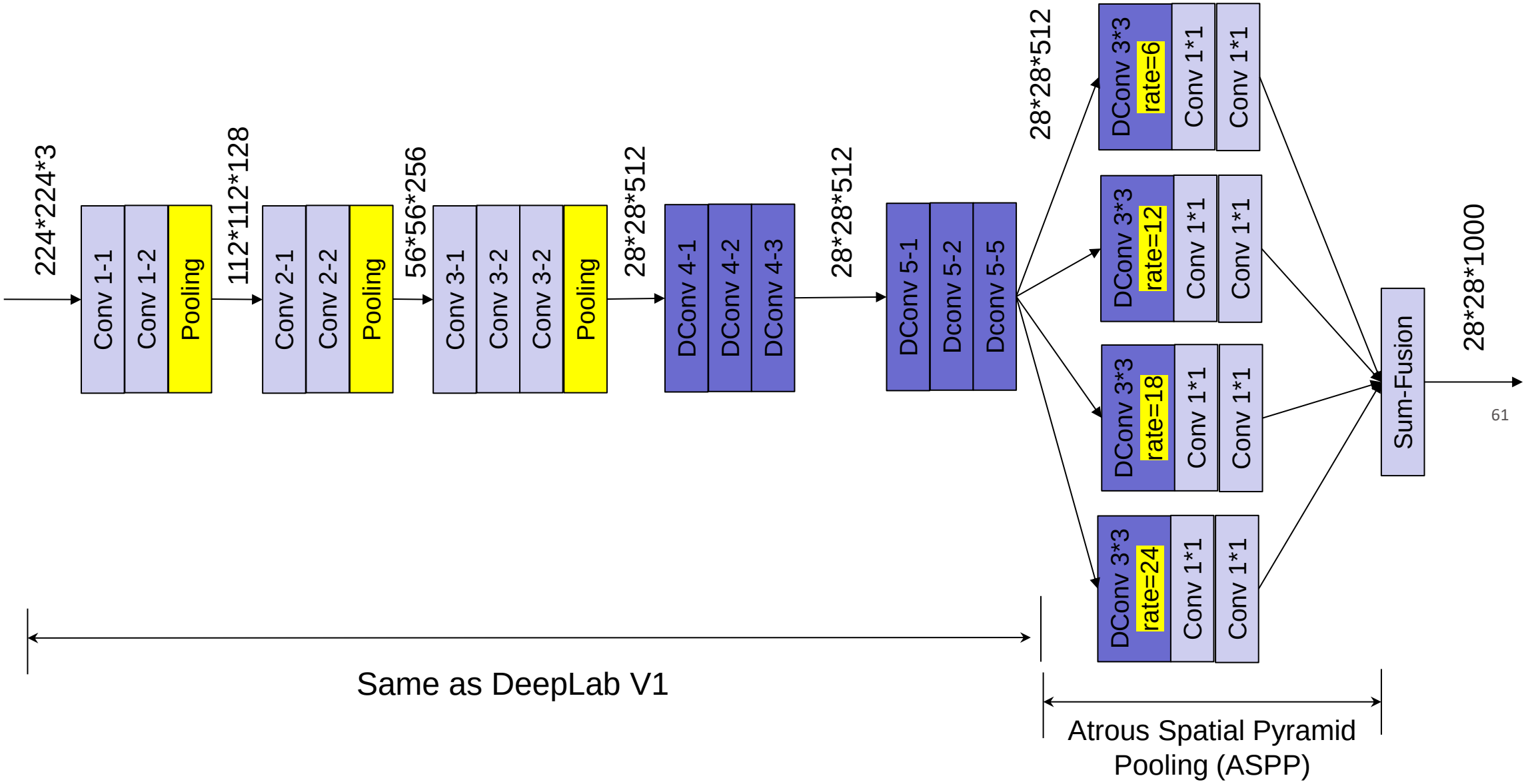
- CRF is a **post-processing step** in DeepLab (no training process)
- CRF favor that similarly-colored neighbored positions have the same class predictions
- CRF penalize that similarly-colored neighbored positions have different class predictions
- CRF minimize an energy function E(x)
- Solving this optimization is challenging. But it can be efficiently solved approximated
- Tuning parameters σ_α, σ_β, σ_γ in CRF may be a difficult task in practice

DeepLab: Outline

- New Technologies:
  - **Dilated Convolution**:
    - Also called „Atrous convolution“
  - **Conditional random field (CRF)**
  - **Atrous Spatial Pyramid Pooling (ASPP)**
- DeepLab v1 [7]: Dilated Convolution + CRF
- DeepLab v2 [8]: Dilated Convolution + CRF + ASPP

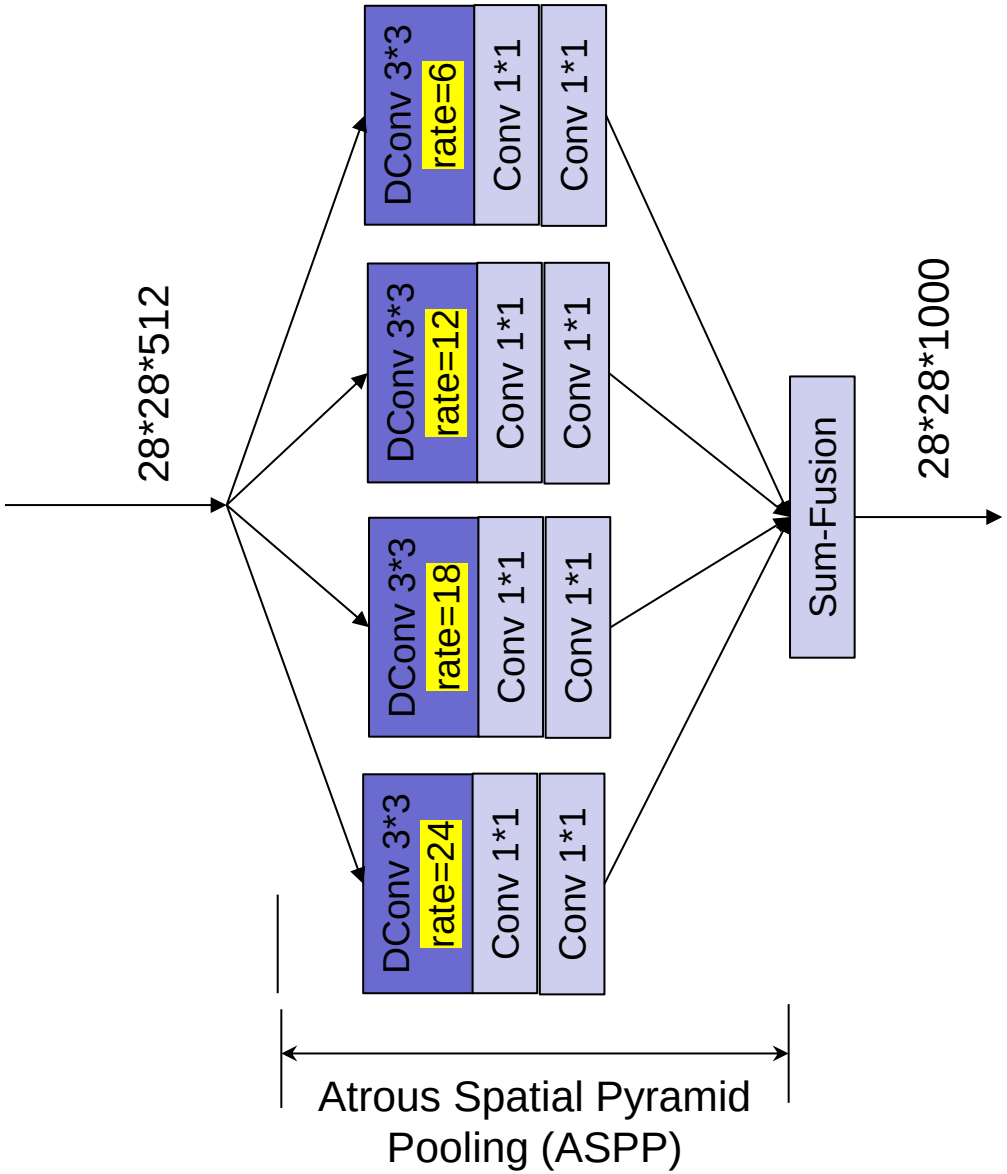
[7] Chen, et al., Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs, ICLR, 2015
[8] Chen, et al., DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, TPAMI, 2017

DeepLab V2: Network Structure



61

DeepLab V2: ASPP



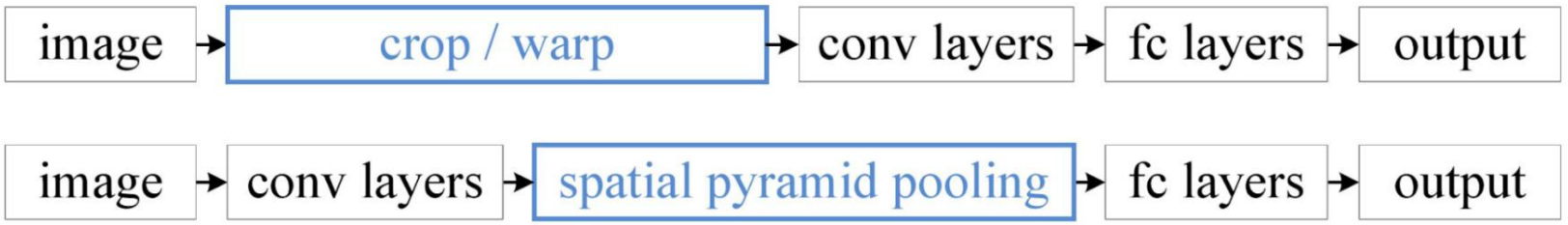
- 4 different dilated convolutions
  - Different Field-of-View
- Idea: Multiscale processing
  - treat object's scale by 4 parallel routes
- Similar to the idea of GoogleNet
- Not exactly the same idea as Spatial Pyramid Pooling (SPP)[*]
  - Because SPP has only convolutions

[*] To my personal understanding

Spatial Pyramid Pooling (SPP)[11] (Not belong to DeepLab)

crop

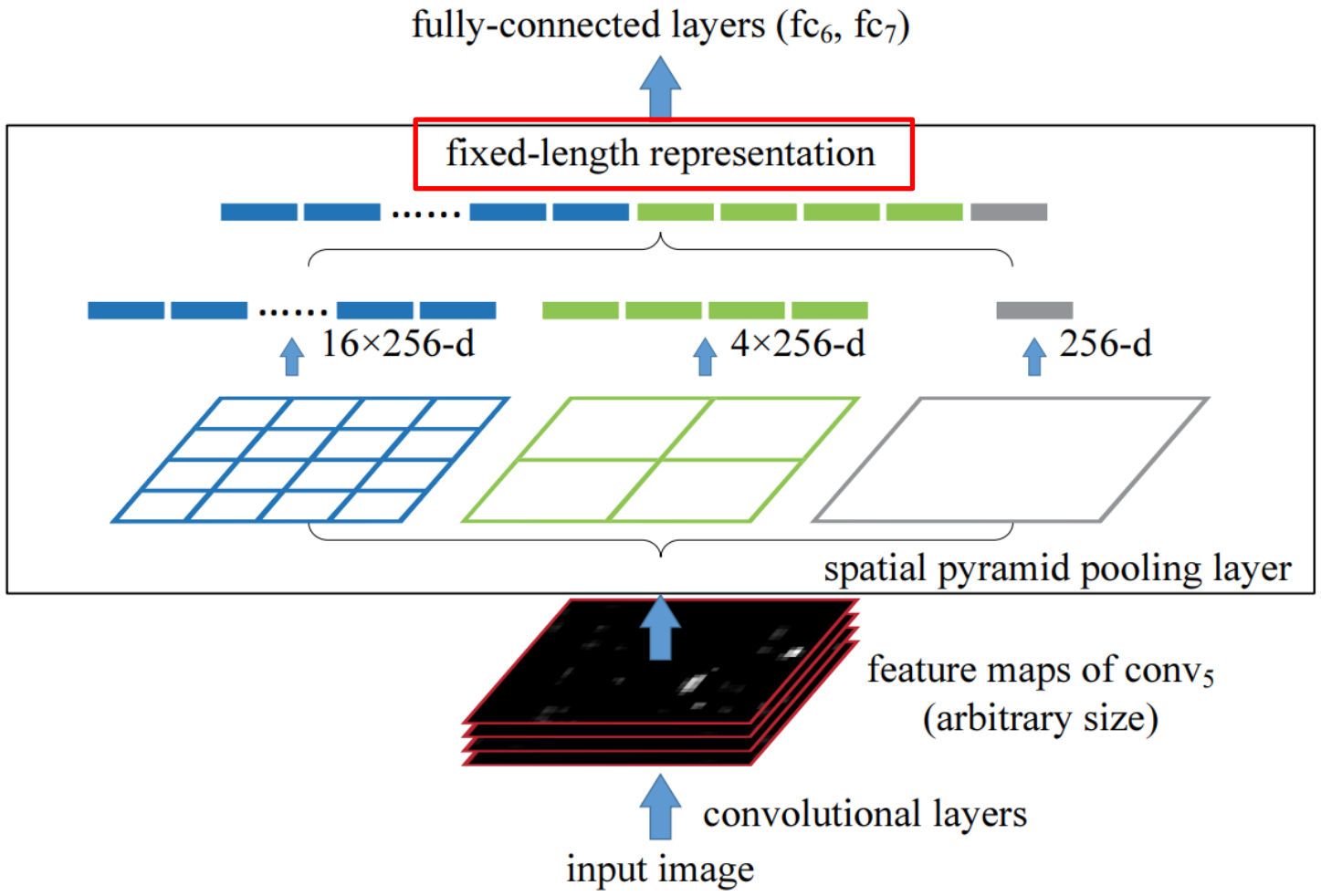
warp



- Standard network (e.g. VGG16) needs a fixed image size
- If image is too big or differently scaled, crop/warp has to be applied
- SPP can be applied to accept image input of arbitrary size

[11] He, et al., Spatial pyramid pooling in deep convolutional networks for visual recognition, TPAMI, 2015

Spatial Pyramid Pooling (SPP)



Input: Image with **arbitrary** size and height-width ratio

SPP Layer:
- multiple max pooling layers
- strides are **proportional** to the input size
- **Same** output size

Output: Fixed-length vector

DeepLab V2: Performance [8]

techniques

DeepLab with different backbone

Fully convolution network

best

Method	MSC	COCO	Aug	LargeFOV	ASPP	CRF	mIOU
VGG-16							
DeepLab [38]				✓			37.6
DeepLab [38]				✓		✓	39.6
ResNet-101							
DeepLab							39.6
DeepLab	✓		✓				41.4
DeepLab	✓	✓	✓				42.9
DeepLab	✓	✓	✓	✓			43.5
DeepLab	✓	✓	✓		✓		44.7
DeepLab	✓	✓	✓		✓	✓	45.7
O_2P [45]							18.1
CFM [51]							34.4
FCN-8s [14]							37.8
CRF-RNN [59]							39.3
ParseNet [86]							40.4
BoxSup [60]							40.5
HO_CRF [91]							41.3
Context [40]							43.3
VeryDeep [93]							44.5

TABLE 6: Comparison with other state-of-art methods on PASCAL-Context dataset.

DeepLab V1 & V2: Summary

- New Technologies:
  - Dilated Convolution to increase FoV
  - ASPP for multiscale processing
  - CRF for post-processing
- Bilinear Interpolation for upsampling
  - No deconvolutions
- DeepLab v3 [9], v3+ [10]: not covered in this lecture

[9] Chen, et al., Rethinking Atrous Convolution for Semantic Image Segmentation, arXiv, 2017 (DeepLab v3)

[10] Chen, et al., Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation, ECCV, 2018 (DeepLab v3+)

DeepLab V3 & V3+

- **DeepLabv1** [7]: We use *atrous convolution* to explicitly control the resolution at which feature responses are computed within Deep Convolutional Neural Networks.
- **DeepLabv2** [8]: We use *atrous spatial pyramid pooling* (ASPP) to robustly segment objects at multiple scales with filters at multiple sampling rates and effective fields-of-views.
- **DeepLabv3** [9]: We augment the ASPP module with *image-level feature* to capture longer range information. We also include *batch normalization* parameters to facilitate the training. In particular, we applying atrous convolution to extract output features at different output strides during training and evaluation, which efficiently enables training BN at output stride = 16 and attains a high performance at output stride = 8 during evaluation.
- **DeepLabv3+** [10]: We extend DeepLabv3 to include a simple yet effective decoder module to refine the segmentation results especially along object boundaries. Furthermore, in this encoder-decoder structure one can arbitrarily control the resolution of extracted encoder features by atrous convolution to trade-off precision and runtime.

Summary

- Where can you learn more?
  - Listed references
  - Search in Google

Reference

[1] Krizhevsky, et al., ImageNet Classification with Deep Convolutional Neural Networks, NIPS, 2012

[2] He, et al., Deep Residual Learning for Image Recognition, CVPR, 2015

[3] Szegedy, et al., Going Deeper with Convolutions, CVPR, 2015

[4] Long, et al., Fully Convolutional Networks for Semantic Segmentation, CVPR, 2015

[5] Noh, et al., Learning Deconvolution Network for Semantic Segmentation, ICCV, 2015

[6] Ronneberger, et al., U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI, 2015

[7] Chen, et al., Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs, ICLR, 2015 (DeepLab v1)

[8] Chen, et al., DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, TPAMI, 2017 (DeepLab v2)

[9] Chen, et al., Rethinking Atrous Convolution for Semantic Image Segmentation, arXiv, 2017 (DeepLab v3)

[10] Chen, et al., Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation, ECCV, 2018 (DeepLab v3+)

Reference

[11] He, et al., Spatial pyramid pooling in deep convolutional networks for visual recognition, TPAMI, 2015

[12] Huang, et al., Densely connected convolutional networks, arXiv, 2018