Fully Convolutional Networks

Machine Perception

Siyu Tang

02 April 2019

Advanced Interactive Technologies

ETH

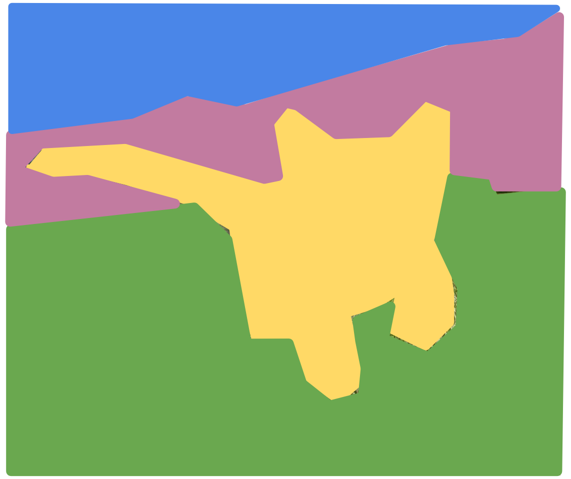
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Pixel-wise Computer Vision Tasks

Semantic Segmentation

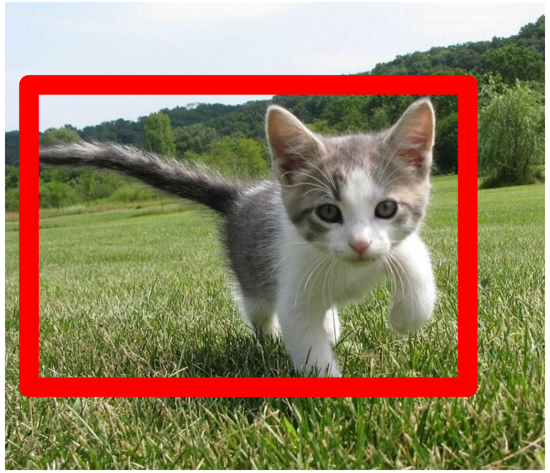
Semantic Segmentation

Semantic Segmentation



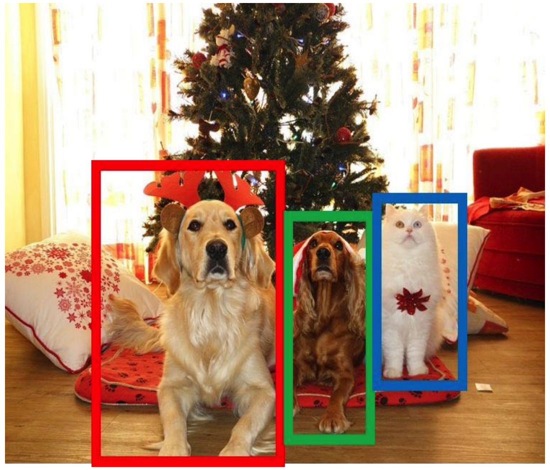
GRASS, CAT, TREE, SKY

Classification + Localization



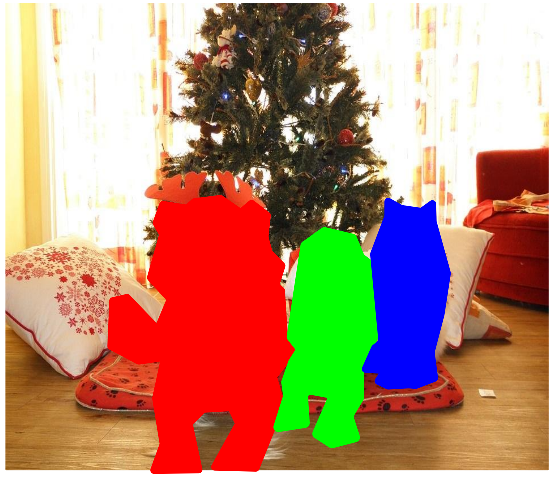
CAT

Object Detection



DOG, DOG, CAT

Instance Segmentation



DOG, DOG, CAT

No objects, just pixels

Single Object

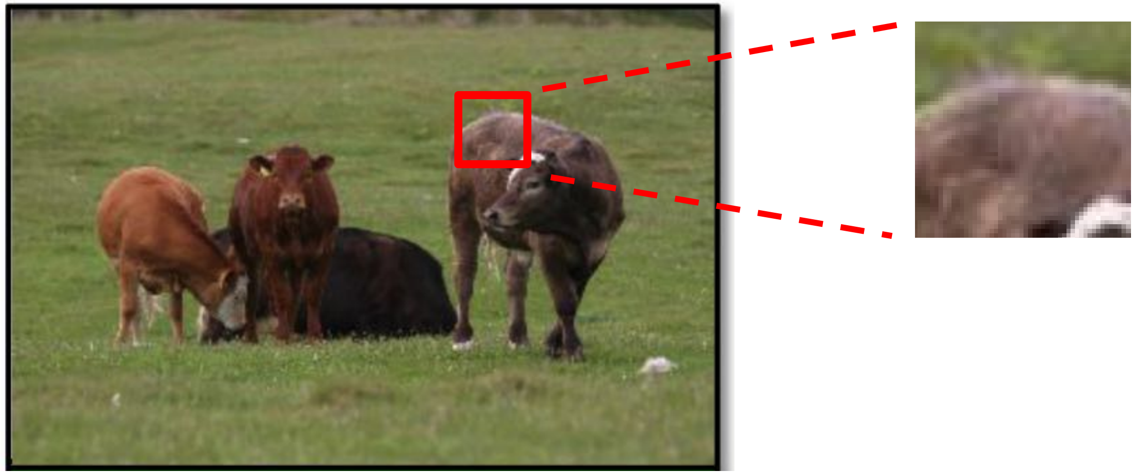
Multiple Object

Semantic Segmentation

[Slides adopted from: Li, Karpathy]

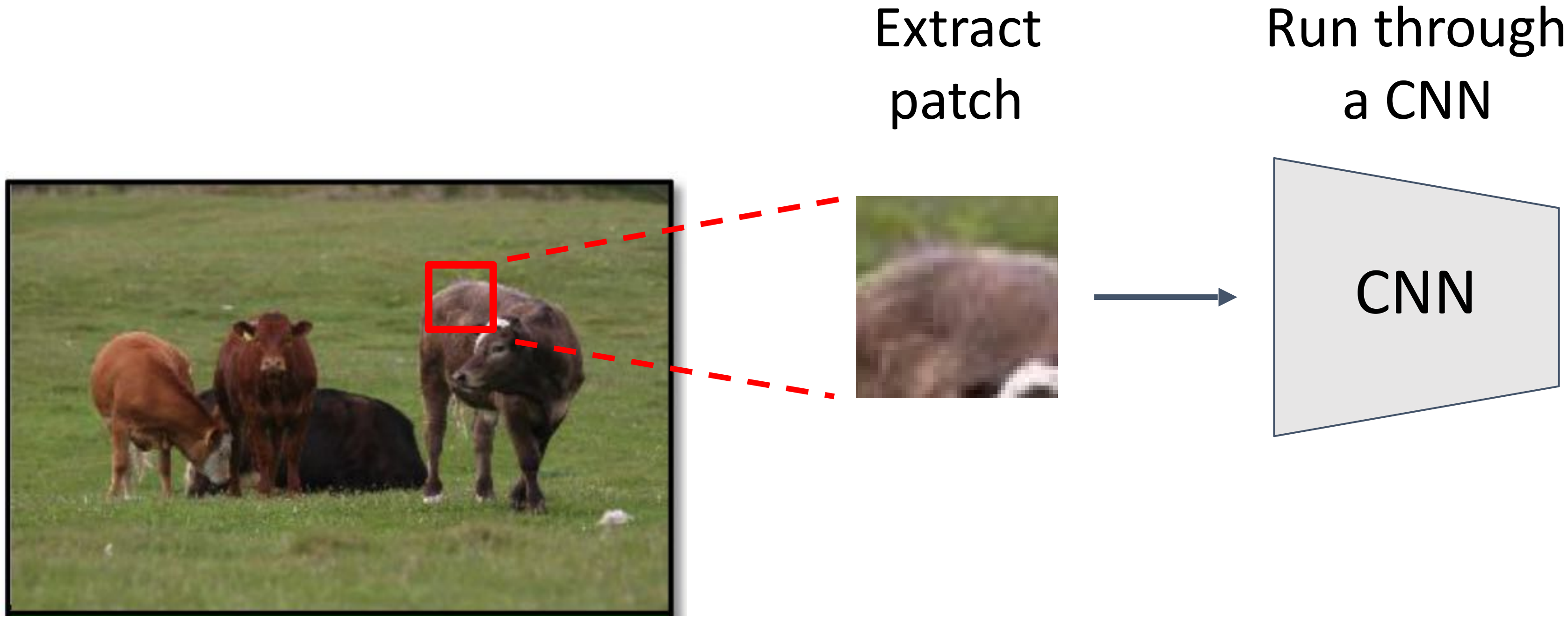
Semantic Segmentation

Extract
patch



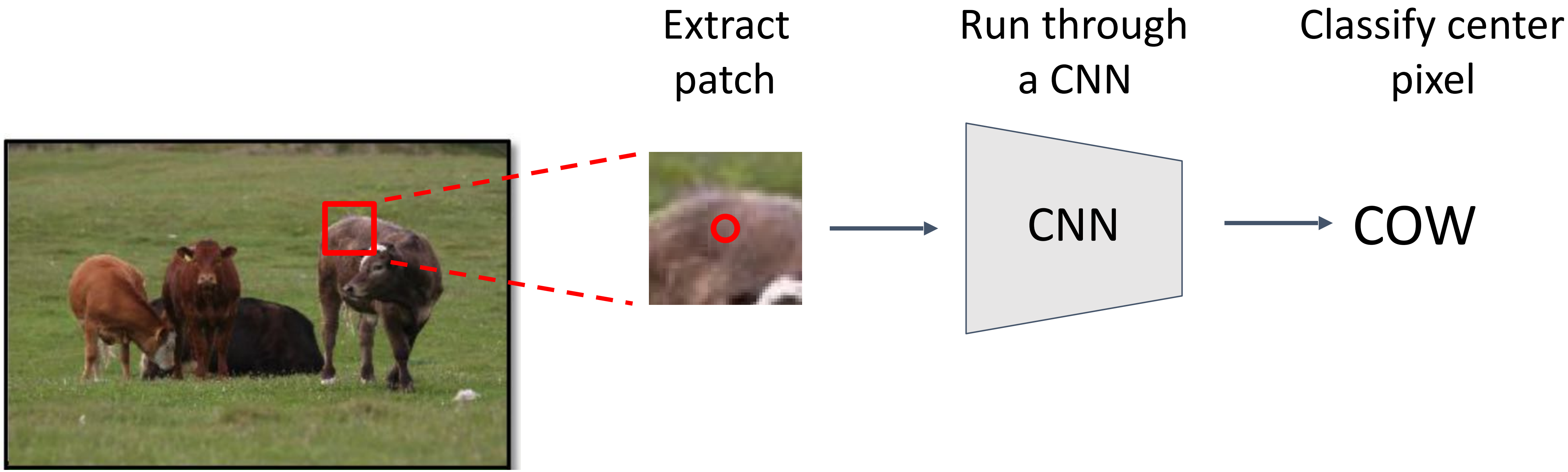
[Slides adopted from: Li, Karpathy]

Semantic Segmentation



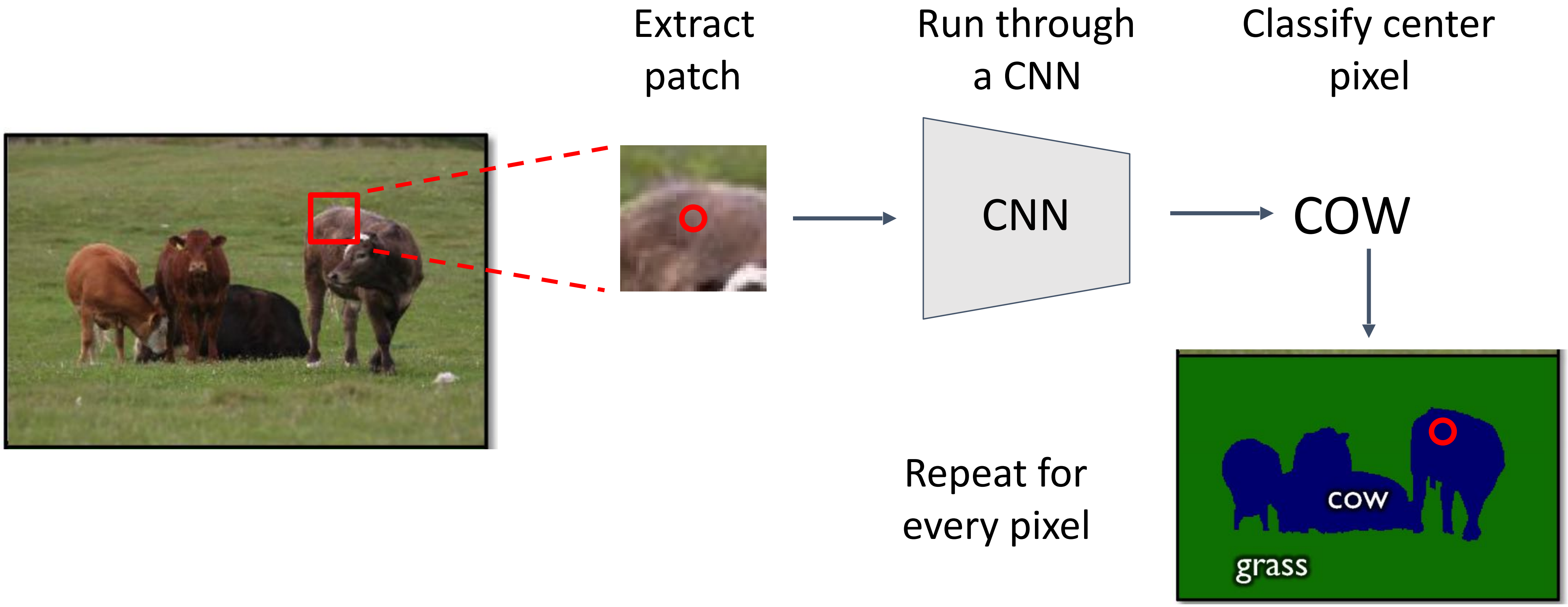
[Slides adopted from: Li, Karpathy]

Semantic Segmentation



[Slides adopted from: Li, Karpathy]

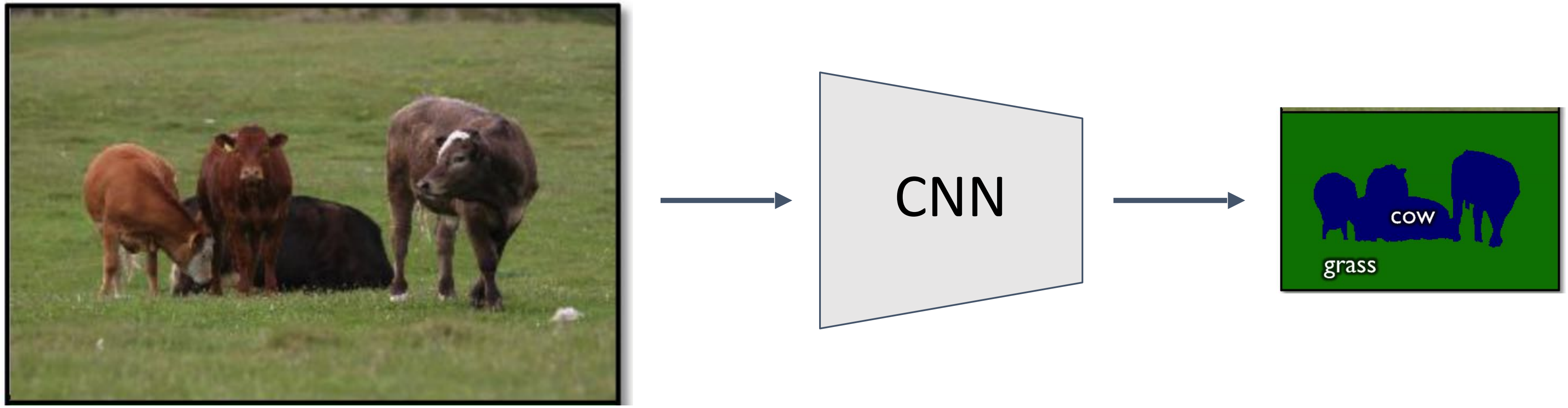
Semantic Segmentation



[Slides adopted from: Li, Karpathy]

Semantic Segmentation

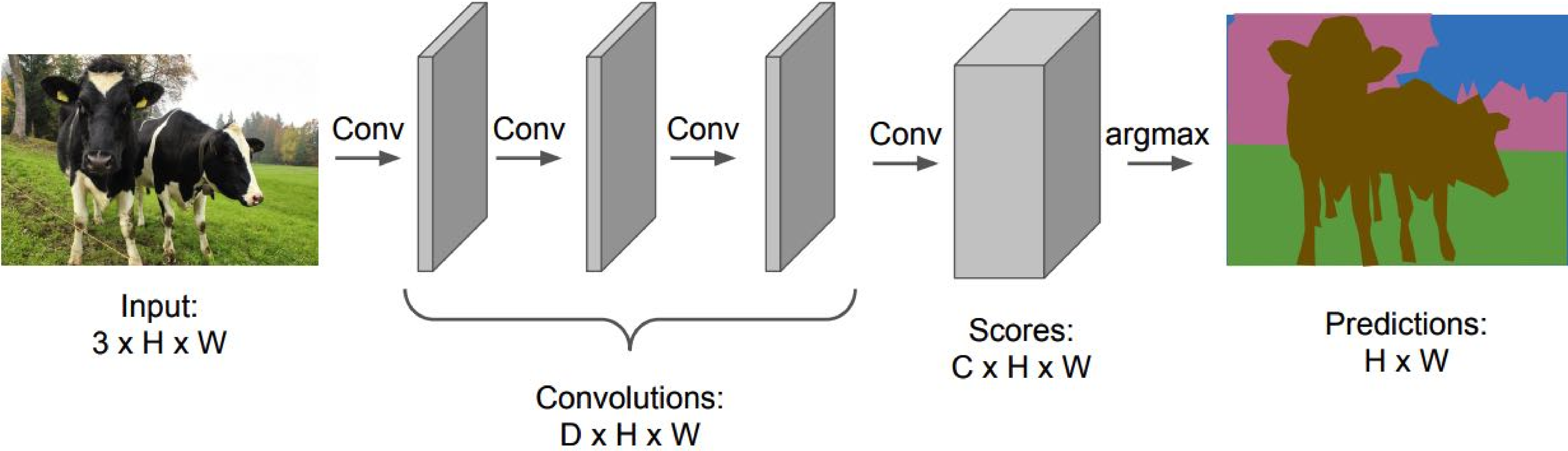
Run "fully convolutional" network
to get all pixels at once



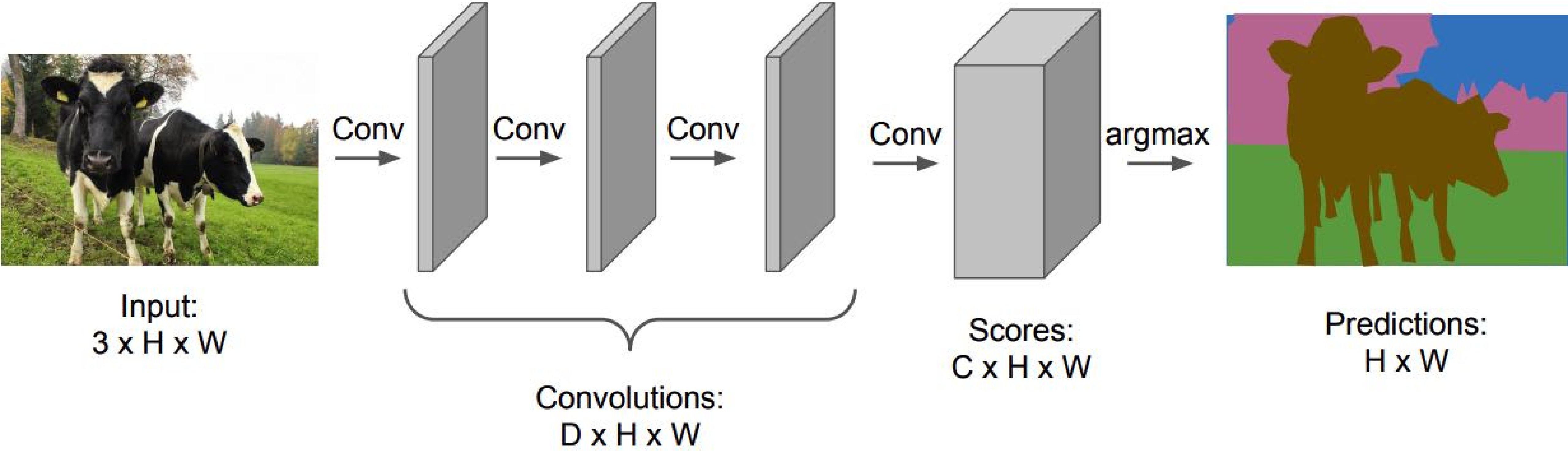
Smaller output
due to pooling

[Slides adopted from: Li, Karpathy]

Semantic Segmentation



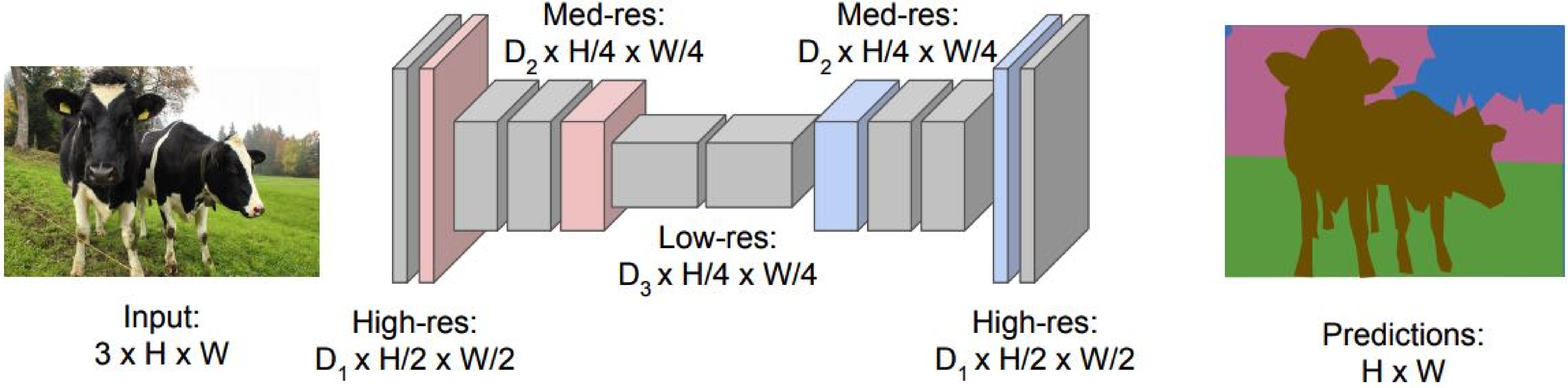
Semantic Segmentation



Problem: convolutions at original image resolution is very expensive!

Semantic Segmentation

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

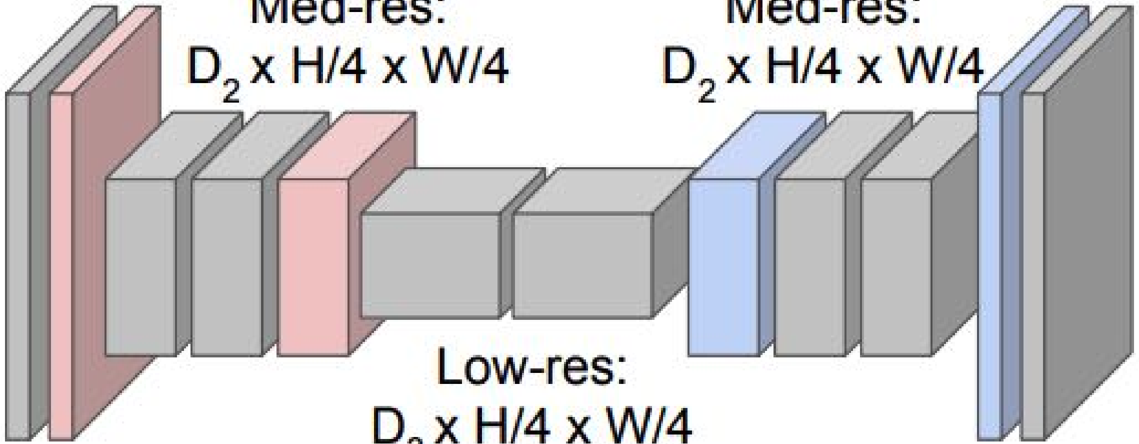


Long, Shelhamer, and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015

Semantic Segmentation

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Input:
3 x H x W



Predictions:
H x W

Downsampling:
Pooling, strided convolution

Upsampling:
???

Long, Shelhamer, and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015

In-Network upsampling: Unpooling

Nearest Neighbor

1	2
3	4

Input: 2 x 2

1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

Output: 4 x 4

“Bed of Nails”

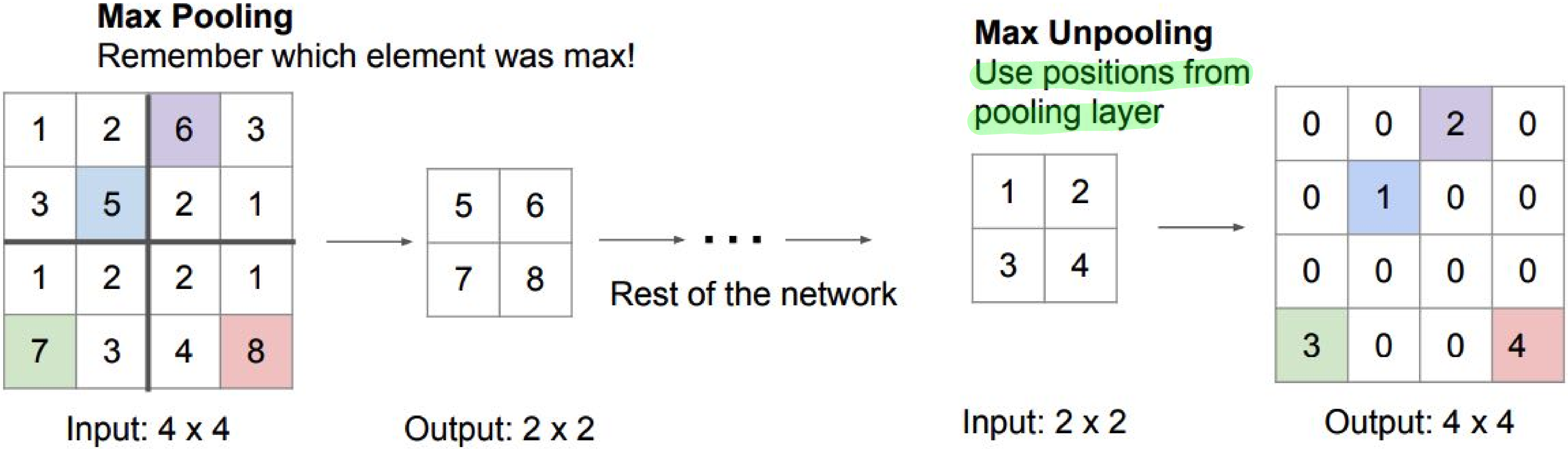
1	2
3	4

Input: 2 x 2

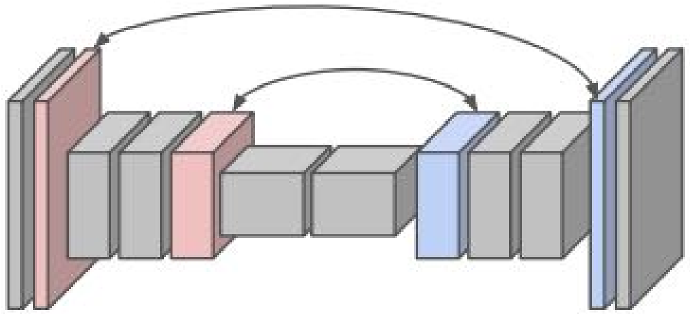
1	0	2	0
0	0	0	0
3	0	4	0
0	0	0	0

Output: 4 x 4

In-Network upsampling: Max Unpooling

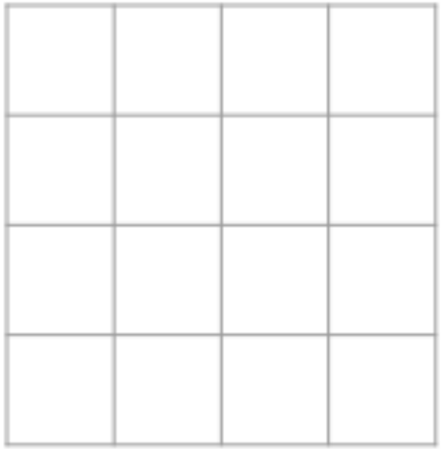


Corresponding pairs of downsampling and upsampling layers

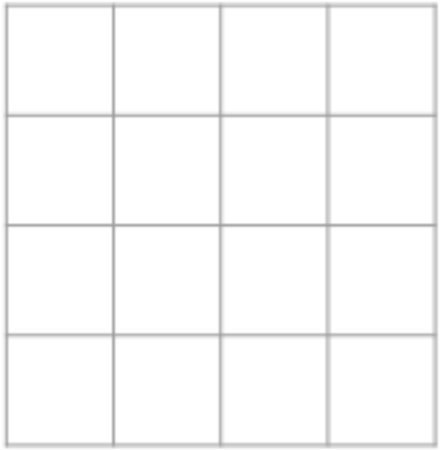


Learnable Upsampling

Normal 3X3 convolution, stride 1 and pad 1



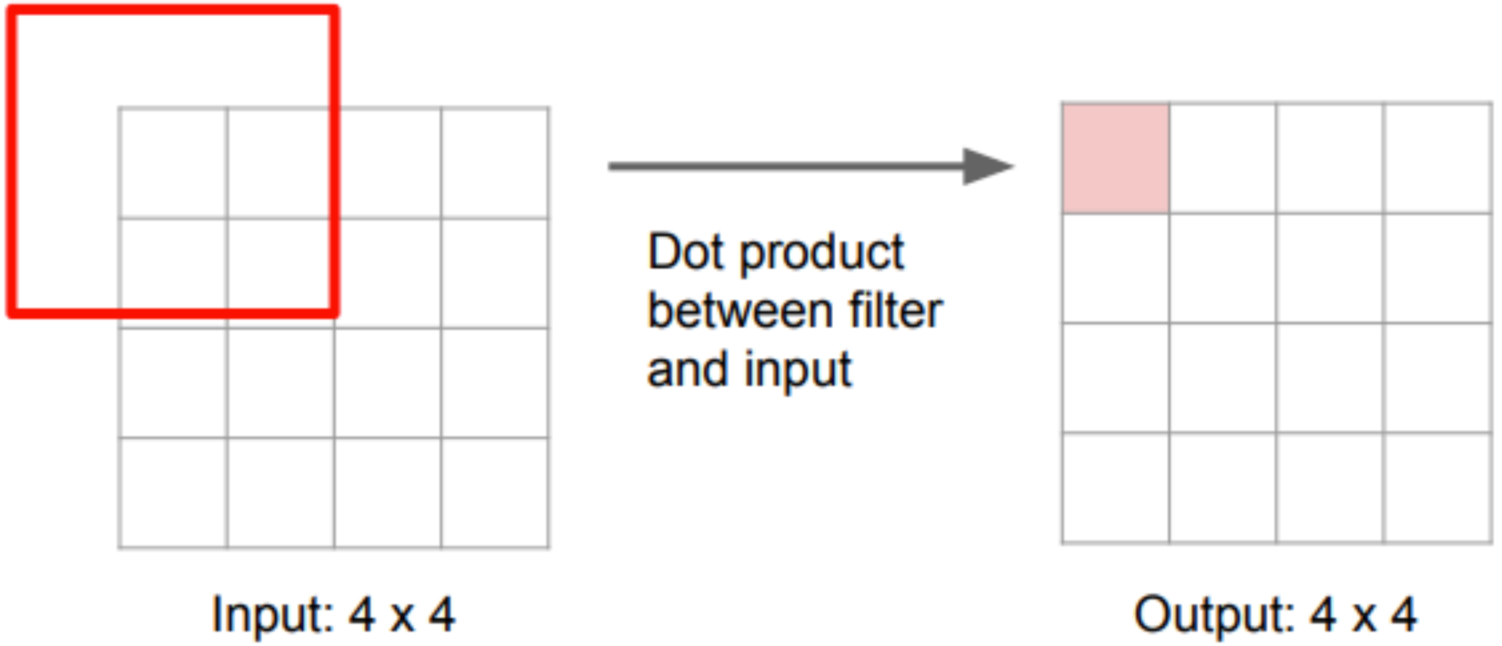
Input: 4 x 4



Output: 4 x 4

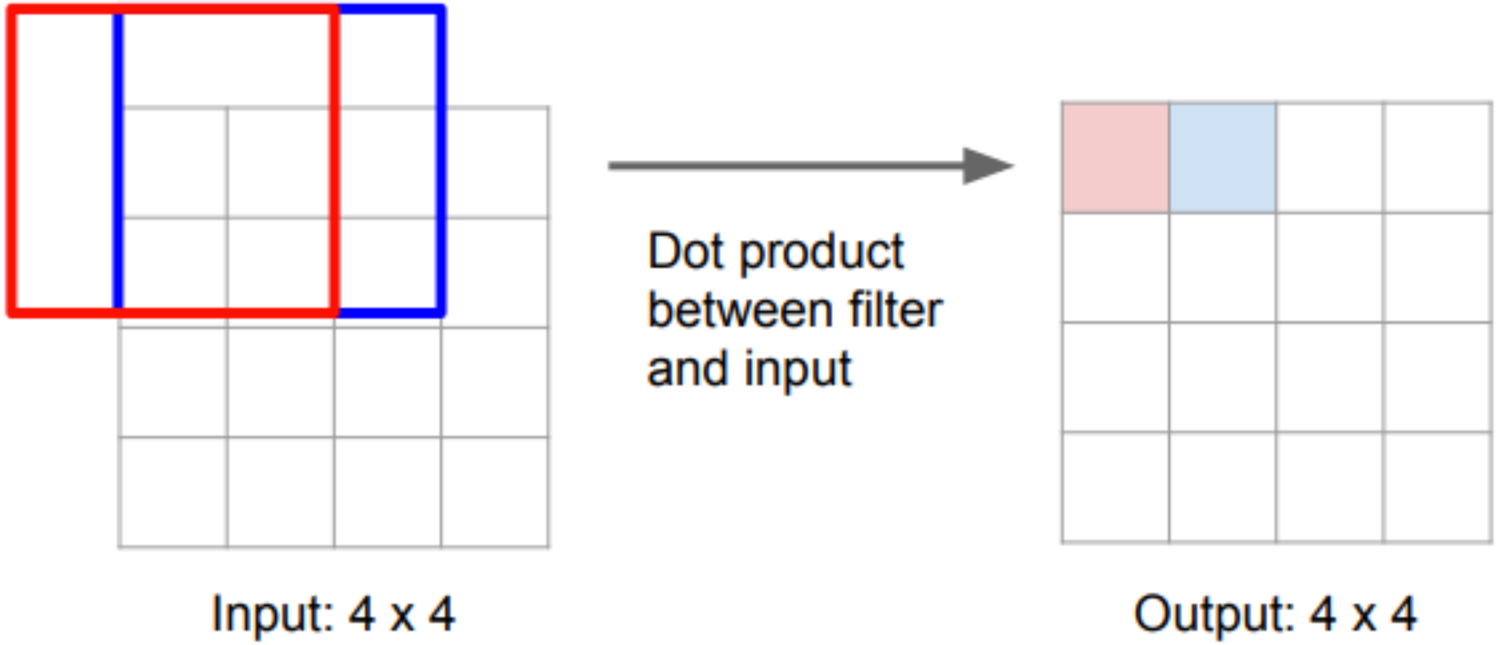
Learnable Upsampling

Normal 3X3 convolution, stride 1 and pad 1



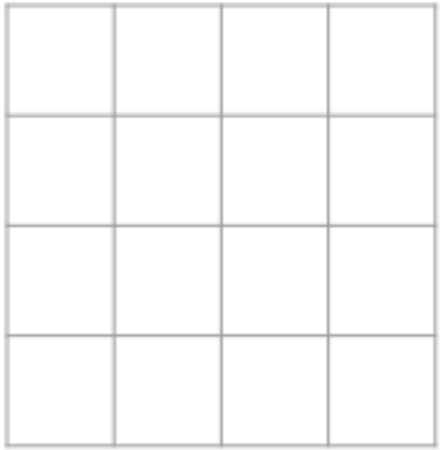
Learnable Upsampling

Normal 3X3 convolution, stride 1 and pad 1

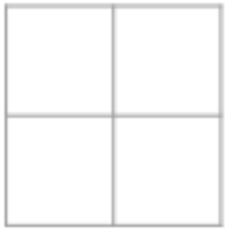


Learnable Upsampling

Normal 3X3 convolution, stride 2 and pad 1



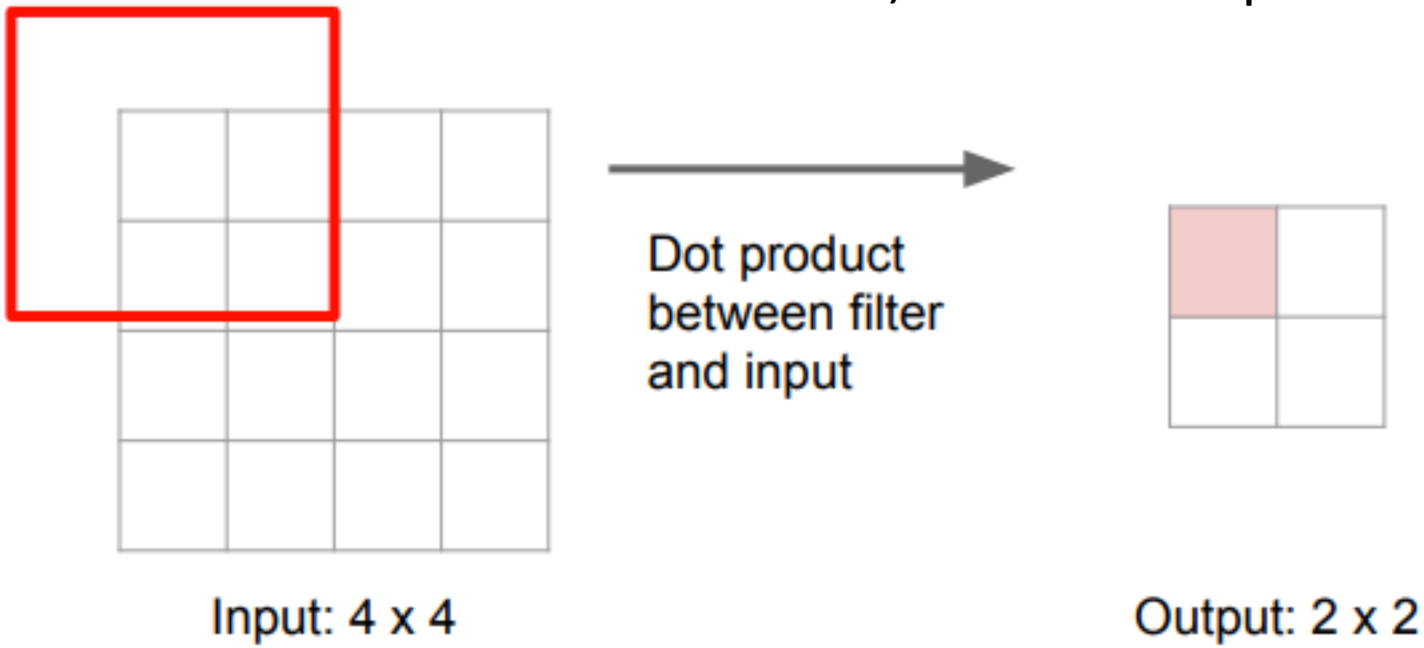
Input: 4 x 4



Output: 2 x 2

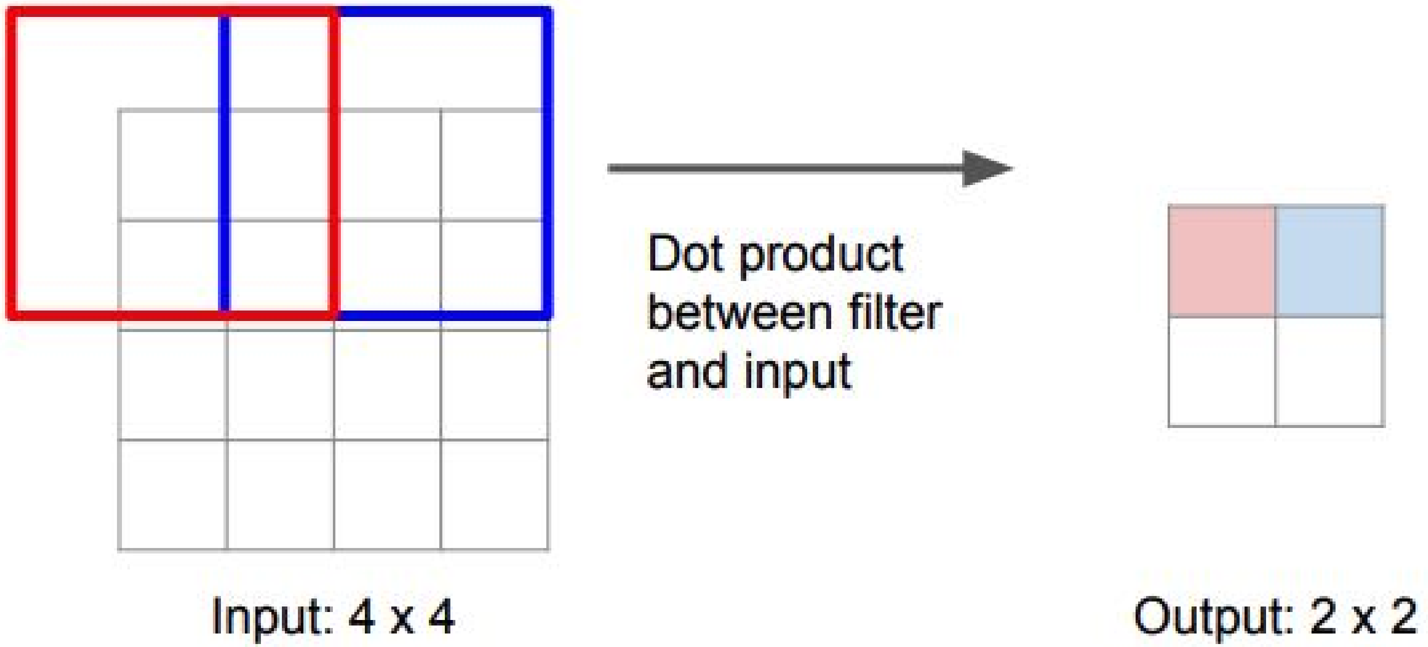
Learnable Upsampling

Normal 3X3 convolution, stride 2 and pad 1



Learnable Upsampling

Normal 3X3 convolution, stride 2 and pad 1



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

Stride gives ratio between movement in input and output

Transposed Convolutions

Transposed Convolutions are used to upsample the input feature map to a desired output feature map using some learnable parameters.
The basic operation that goes in a transposed convolution is explained below:

1. Consider a 2x2 encoded feature map which needs to be upsampled to 3x3 feature map.

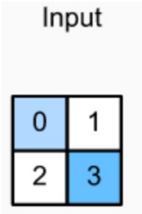


Figure 7. Input Feature Map

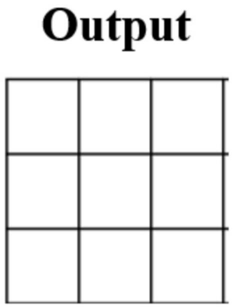


Figure 8. Output Feature Map

2. We take a kernel of size 2x2 with unit stride and zero padding.

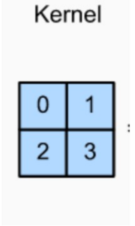


Figure 9. Kernel of size 2x2

3. Now we take the upper left element of the input feature map and multiply it with every element of the kernel as shown in figure 10.

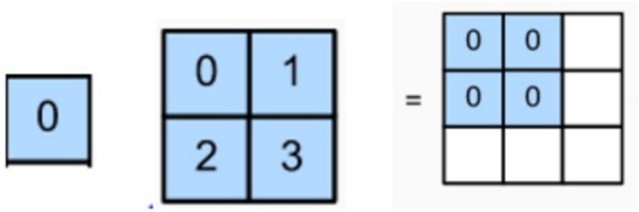
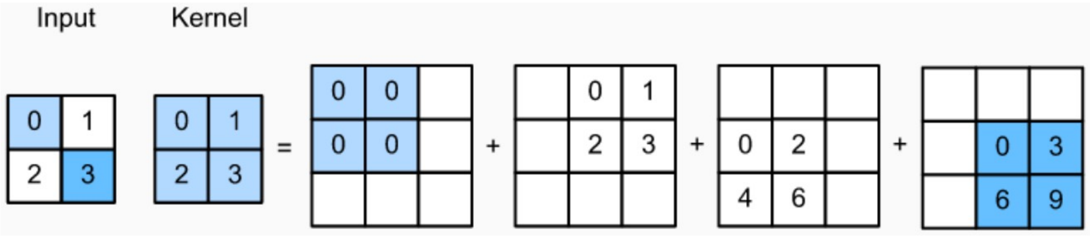
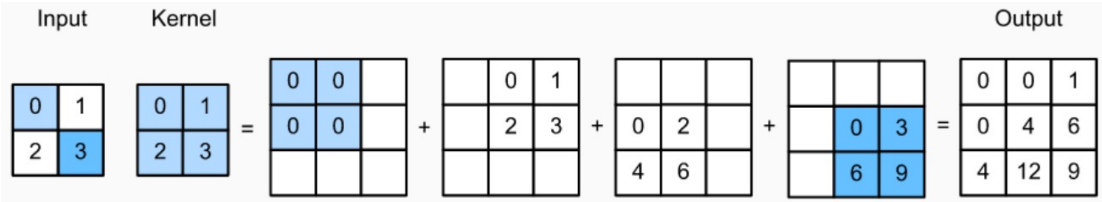


Figure 10.

4. Similarly, we do it for all the remaining elements of the input feature map as depicted in figure 11.

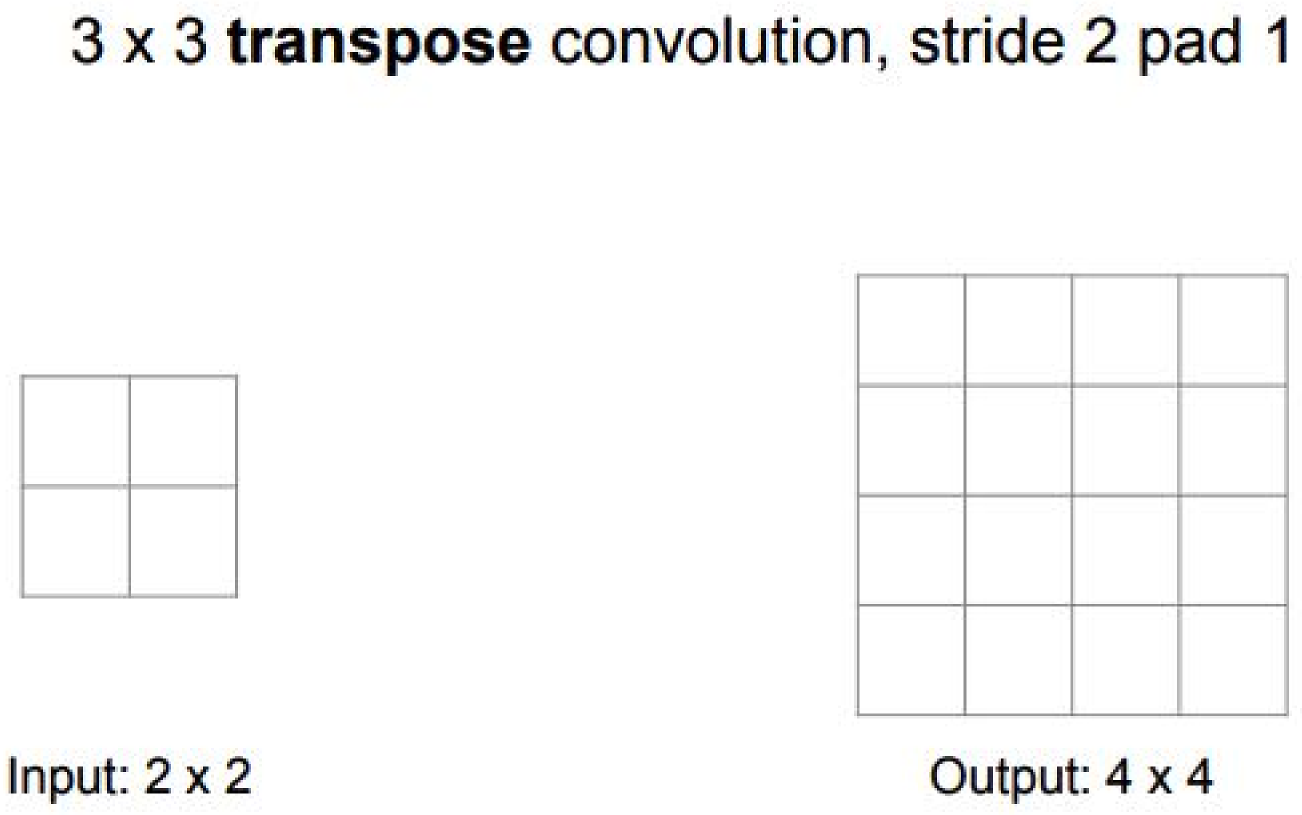


5. As you can see, some of the elements of the resulting upsampled feature maps are over-lapping. To solve this issue, we simply add the elements of the over-lapping positions.



Transposed Convolution Demystified
https://towardsdatascience.com/transposed-convolution-demystified-84ca81b4baba

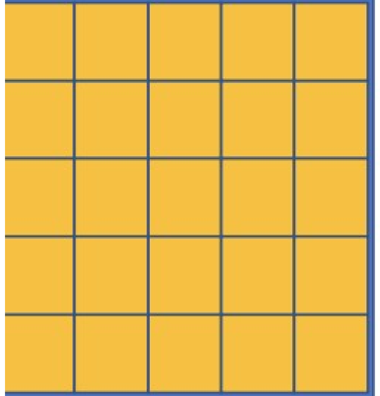
Learnable Upsampling



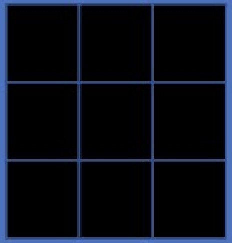
推荐阅读：
一文搞懂transposed convolution
https://www.cnblogs.com/shine-lee/p/11559825.html

Computer Vision

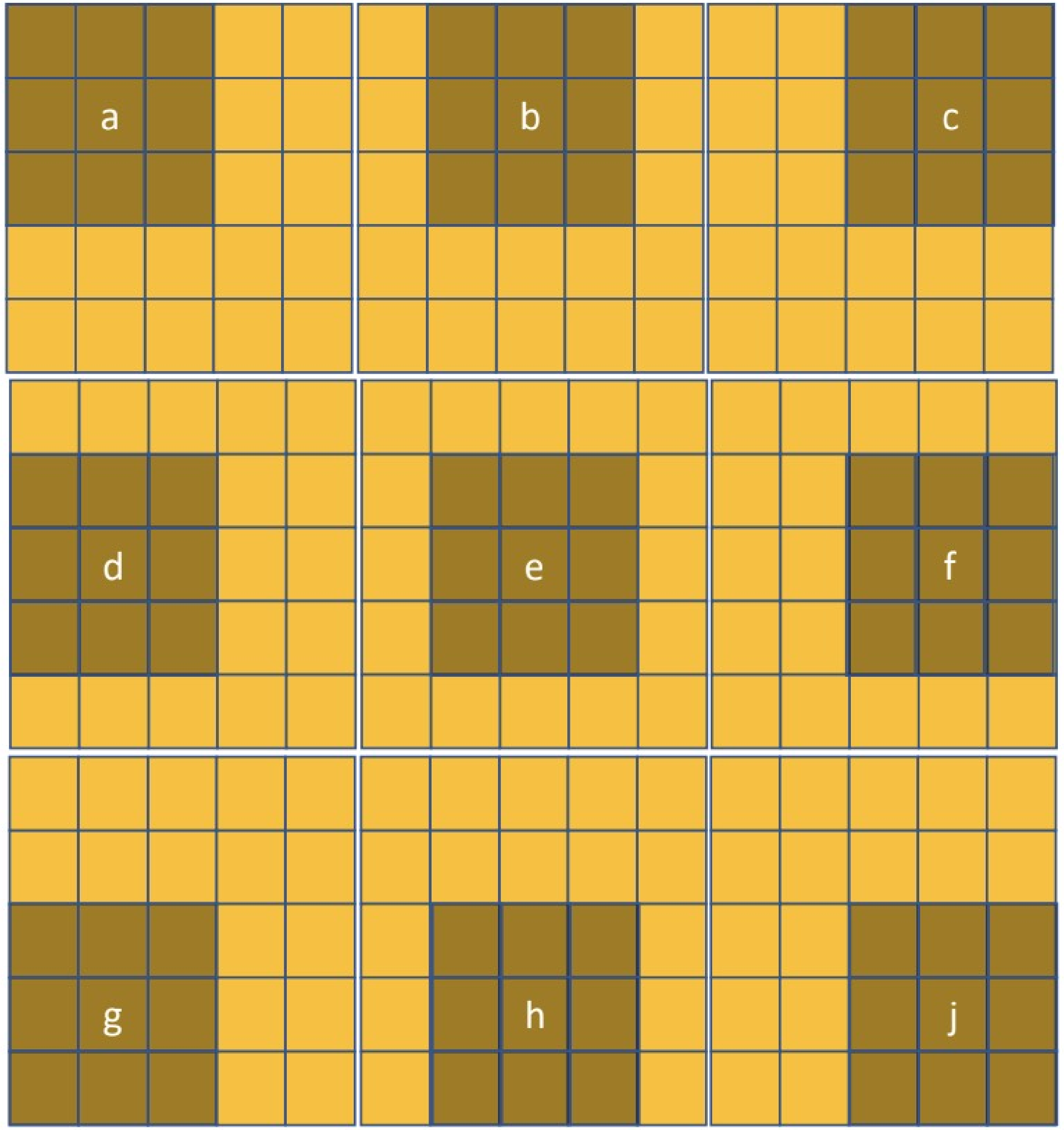
Detection



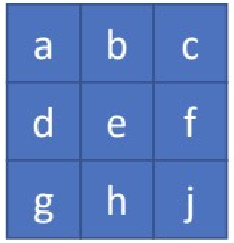
Image



kernel



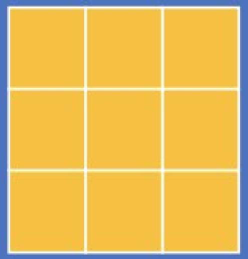
Convolution



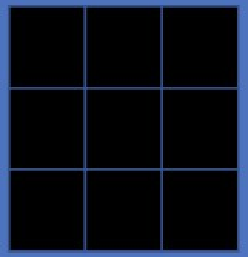
Computer Vision

Detection

Segmentation

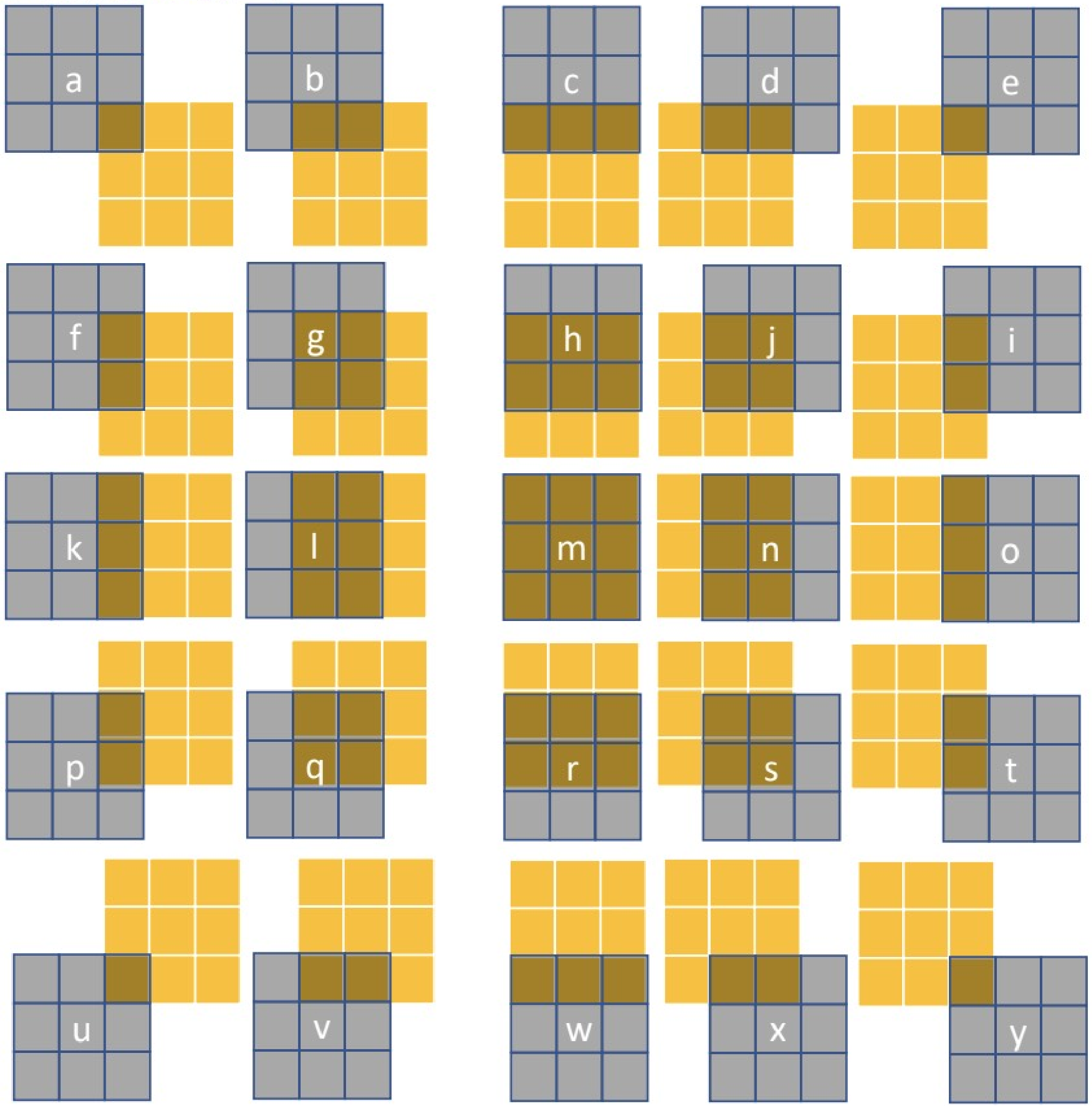


Image



kernel

这里可能是padding了一下



Transposed Convolution

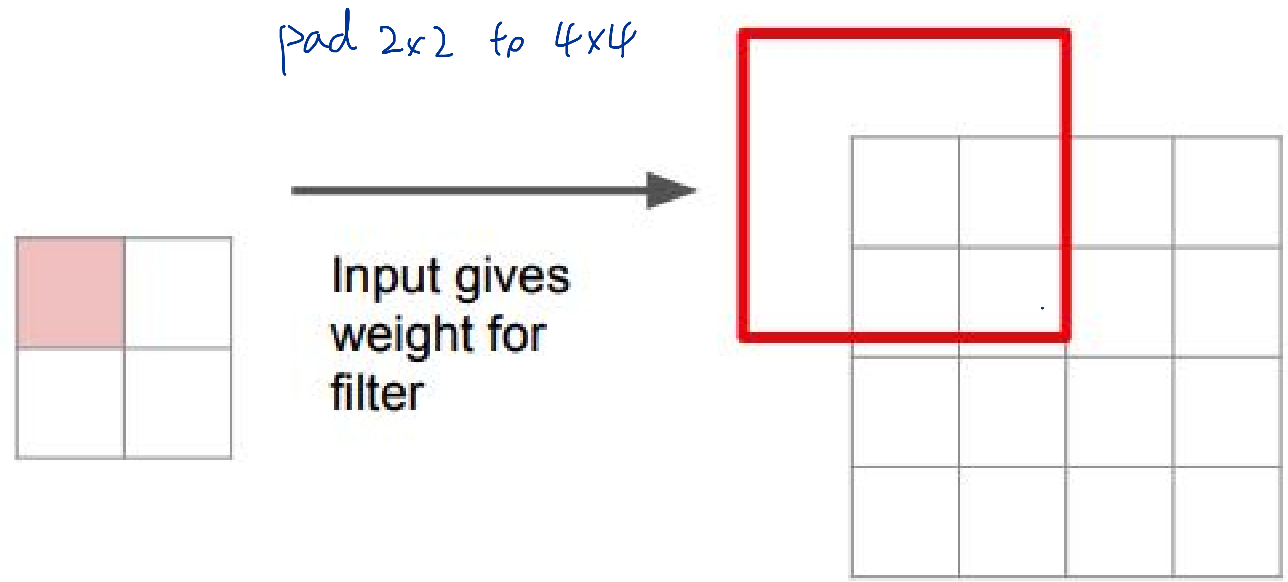
Sometimes referred to as deconvolution but that is not correct terminology.

有时称为反卷积，但这不是正确的术语

a	b	c	d	e
f	g	h	i	j
k	l	m	n	o
p	q	r	s	t
u	v	w	x	y

Learnable Upsampling

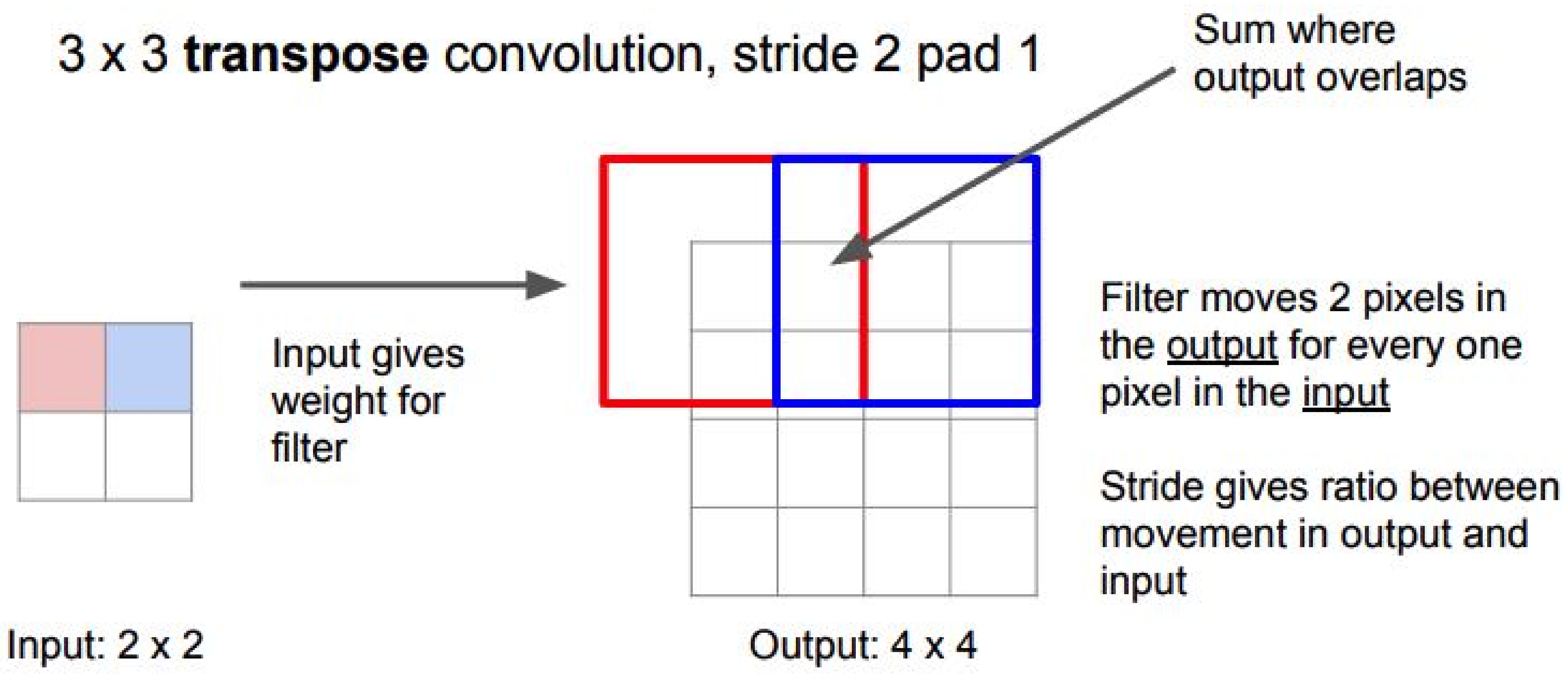
3 x 3 **transpose** convolution, stride 2 pad 1



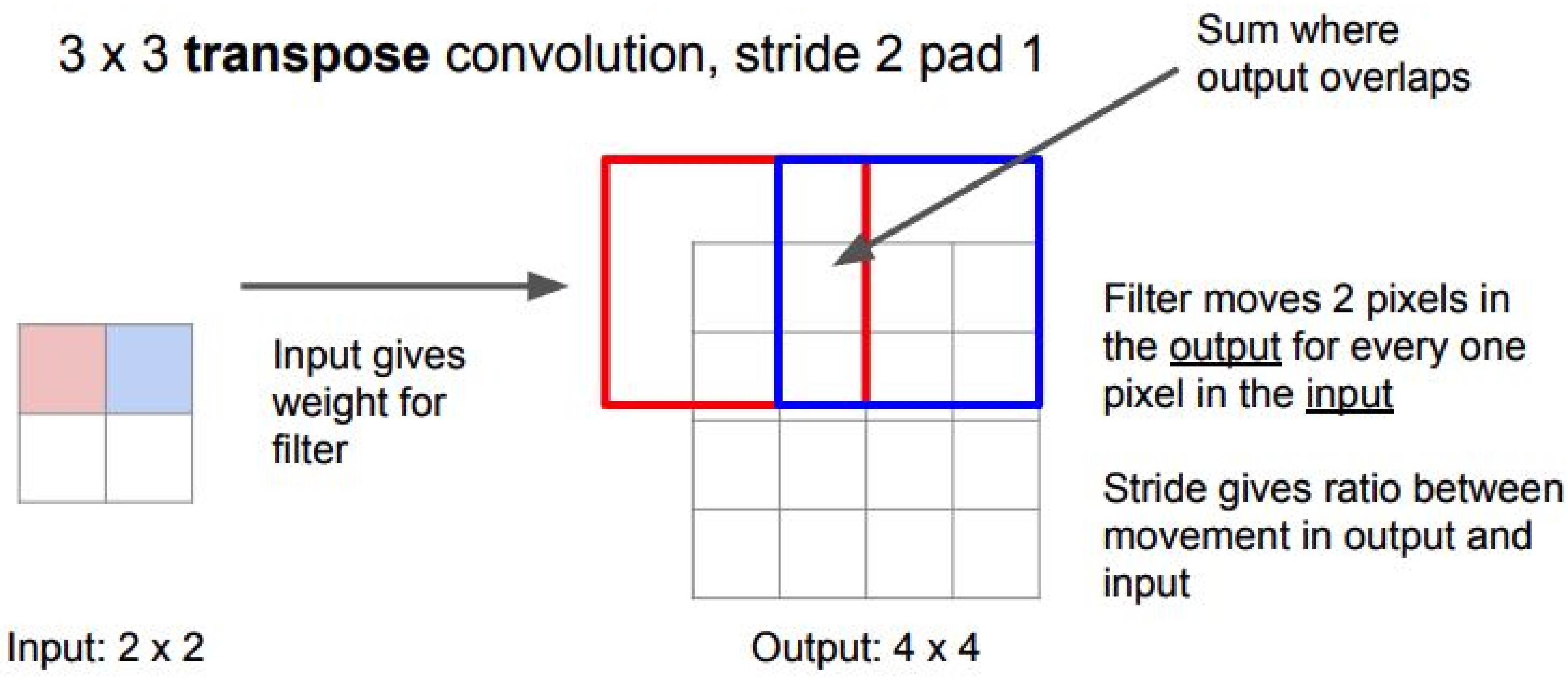
Input: 2 x 2

Output: 4 x 4

Learnable Upsampling



Learnable Upsampling

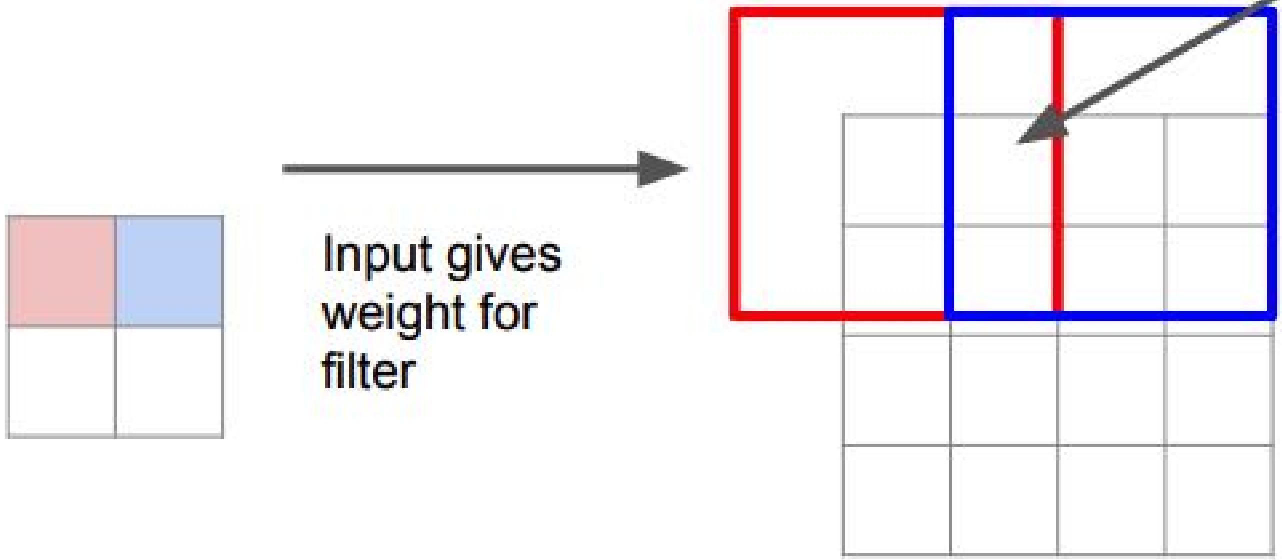


Learnable Upsampling

Other names:
-Deconvolution (bad)
-Upconvolution
-Fractionally strided convolution
-Backward strided convolution

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

Sum where output overlaps



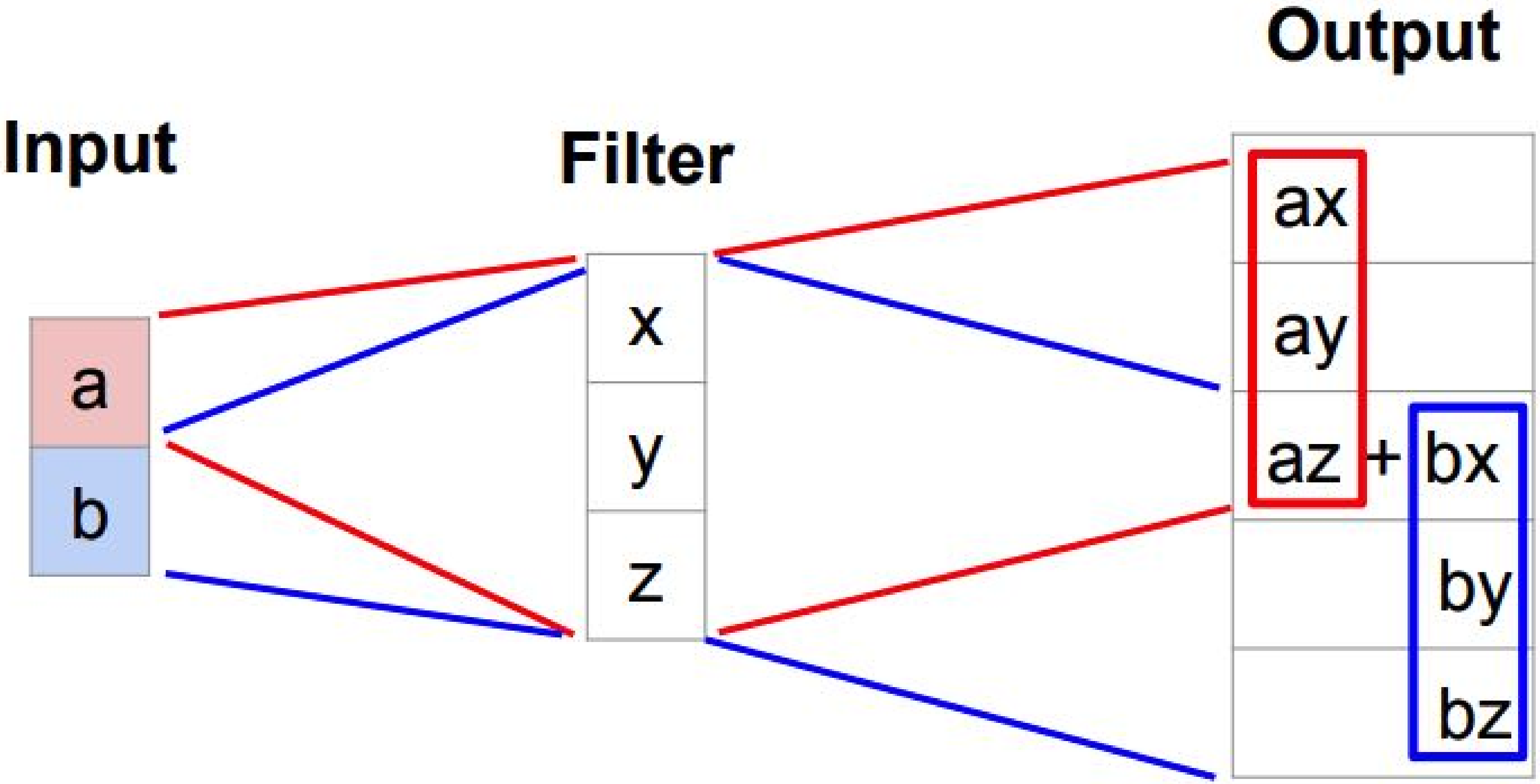
Input: 2 x 2

Output: 4 x 4

Filter moves 2 pixels in the <u>output</u> for every one pixel in the <u>input</u>

Stride gives ratio between movement in output and input

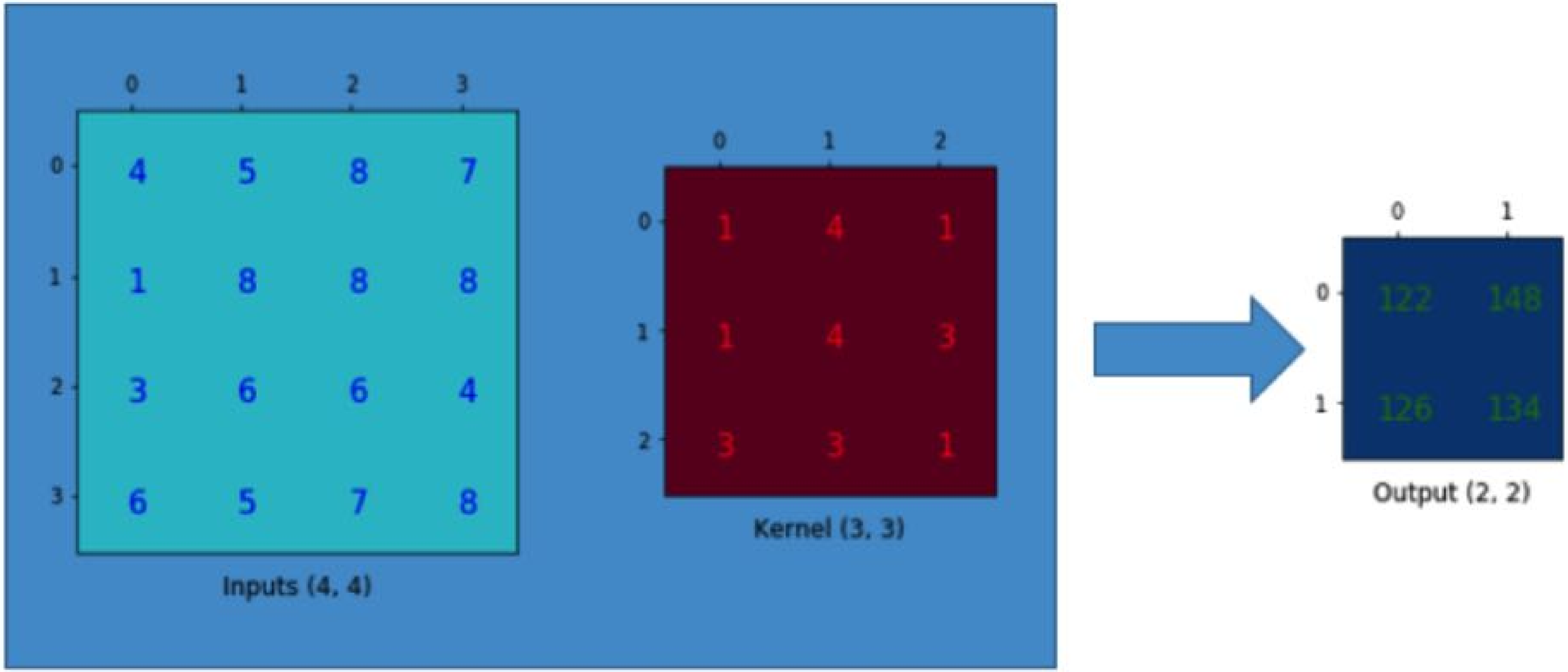
Learnable Upsampling: 1D Example



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

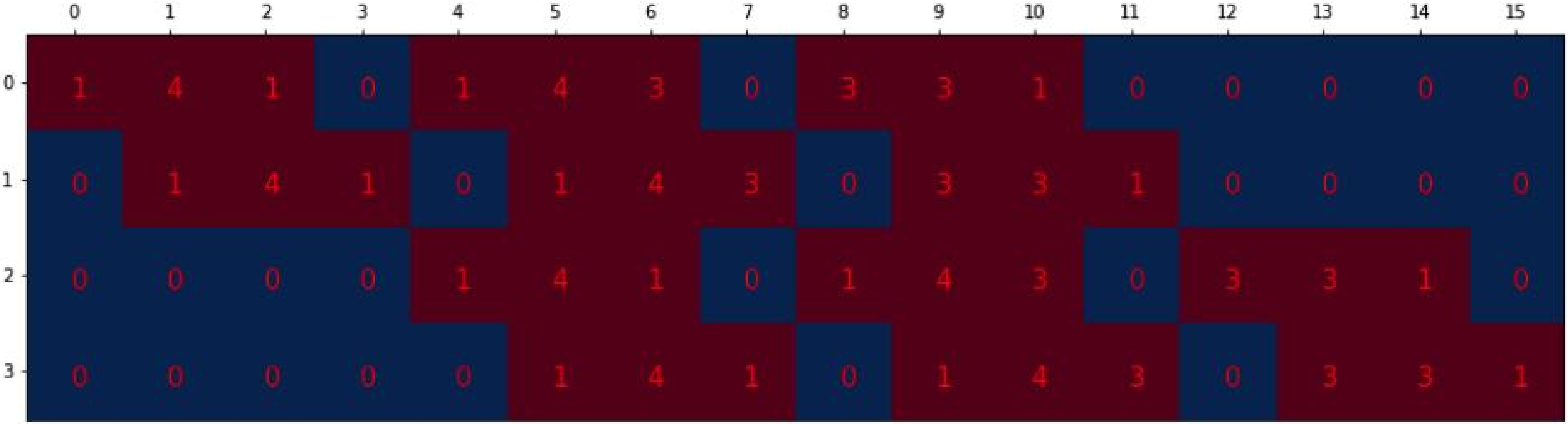
Need to crop one pixel from output to make output exactly 2x input

Convolution as Matrix Operation



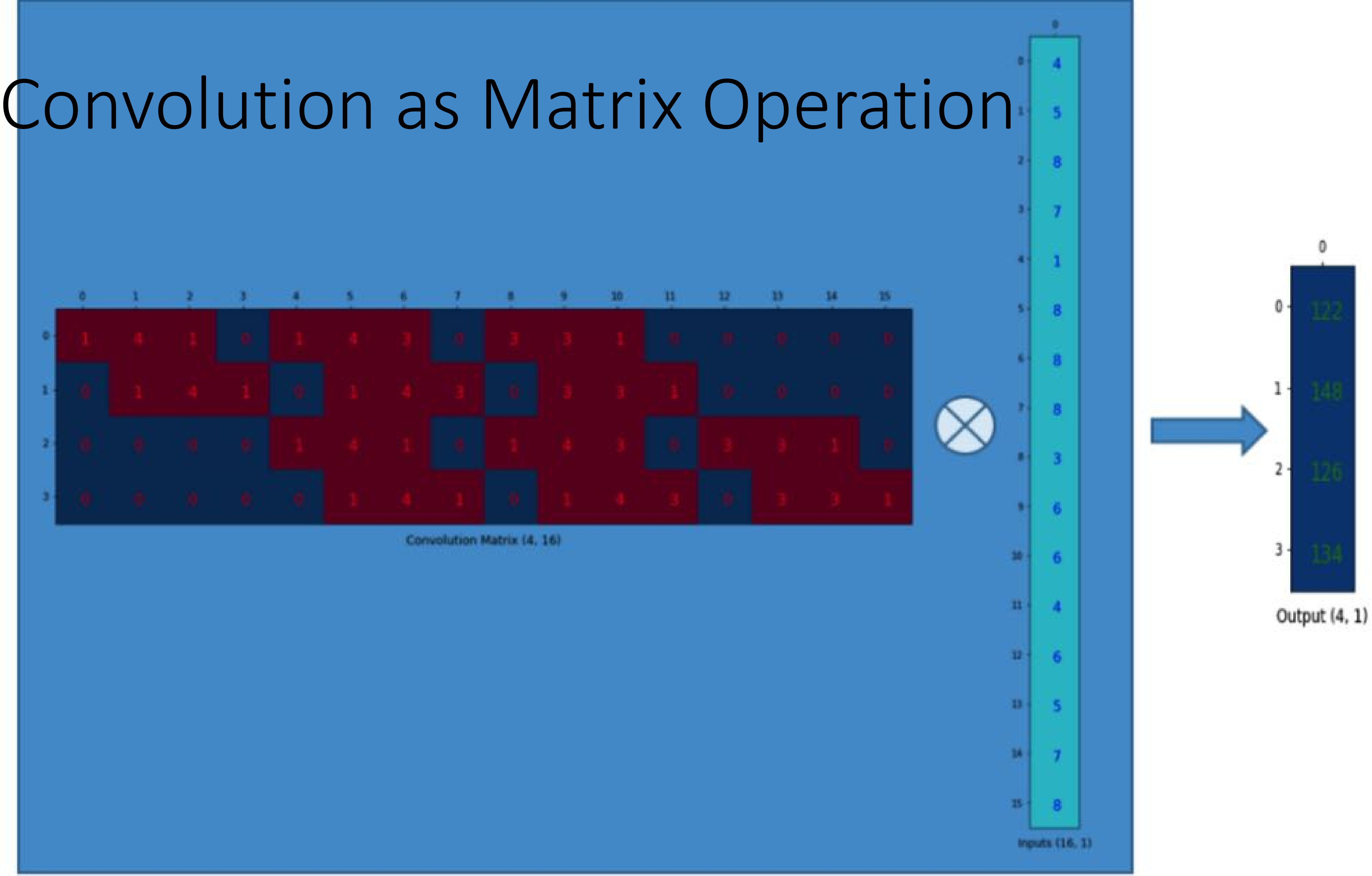
Convolution Operation

Convolution as Matrix Operation

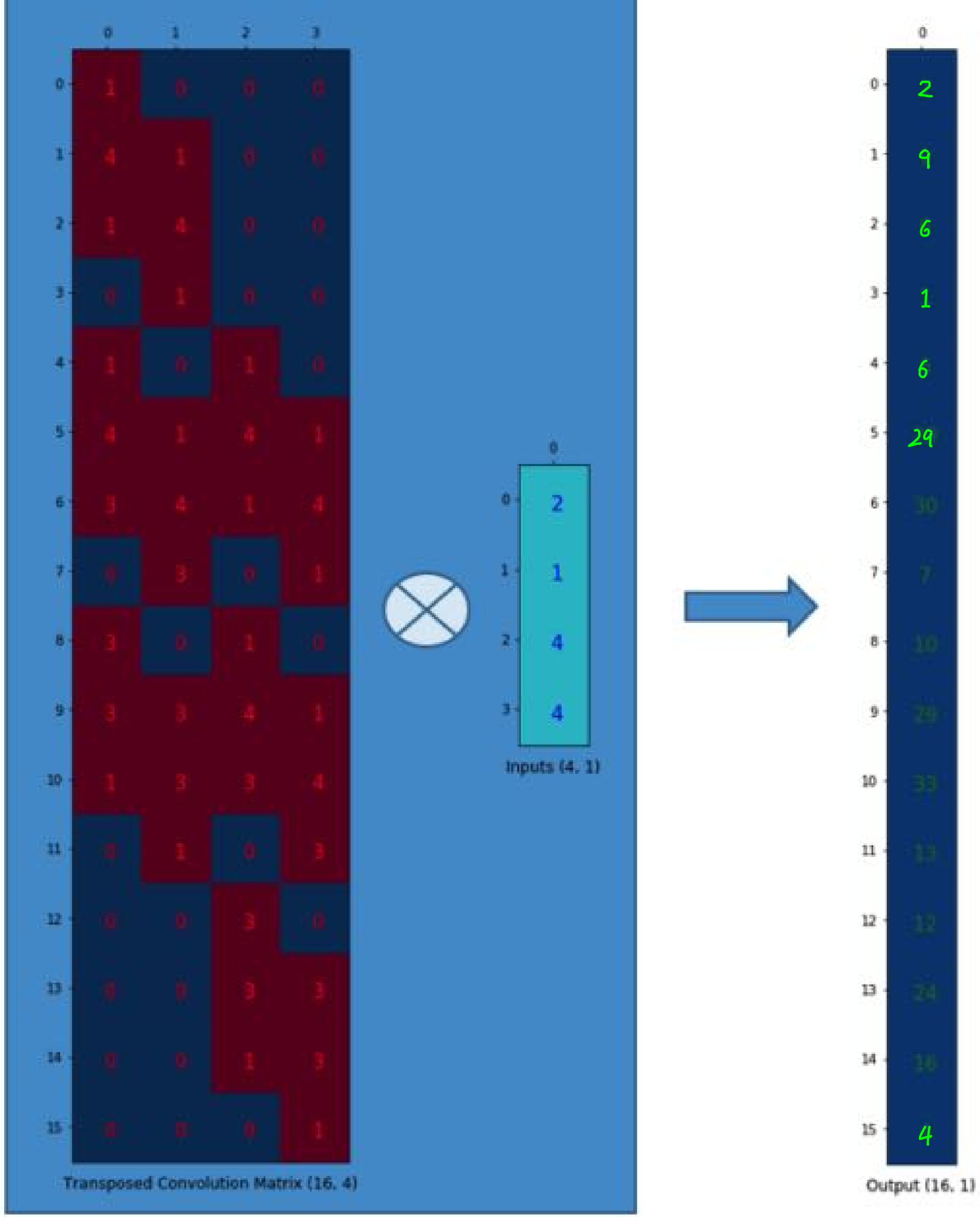


Convolution Matrix (4, 16)

Convolution as Matrix Operation

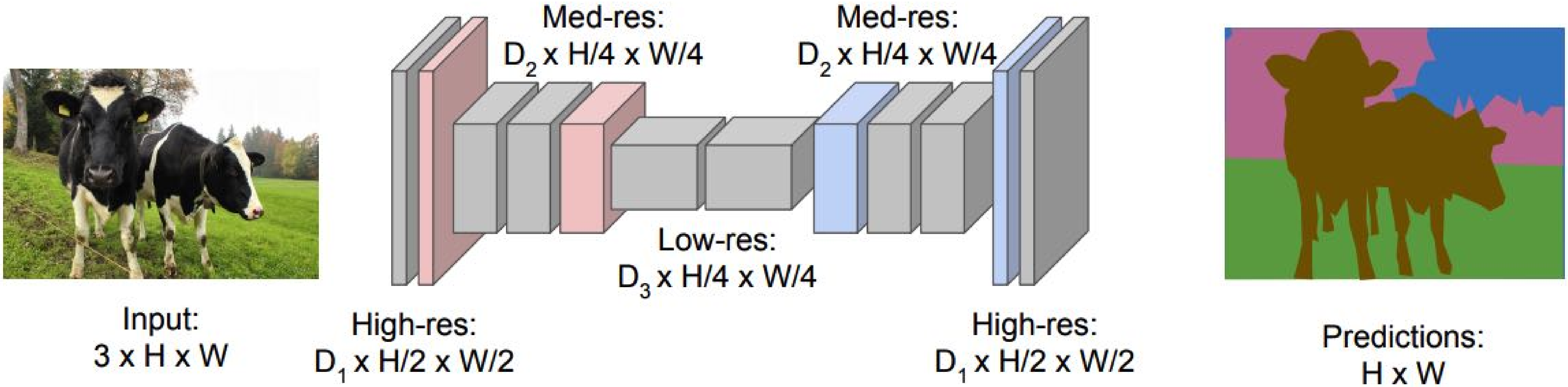


Transpose Convolution



Semantic Segmentation

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!



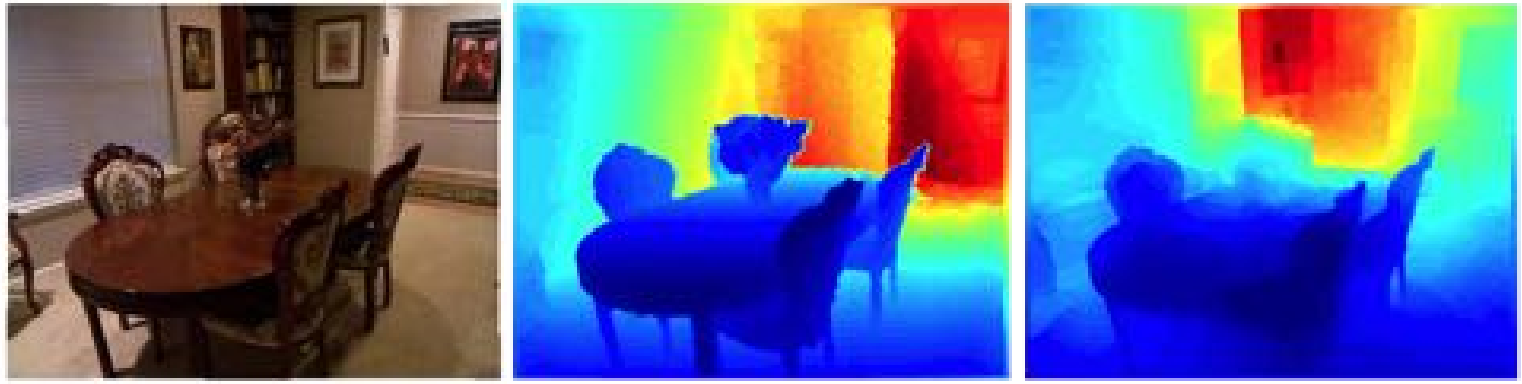
Downsampling:
Pooling, strided convolution

Upsampling:
Unpooling or strided transpose convolution

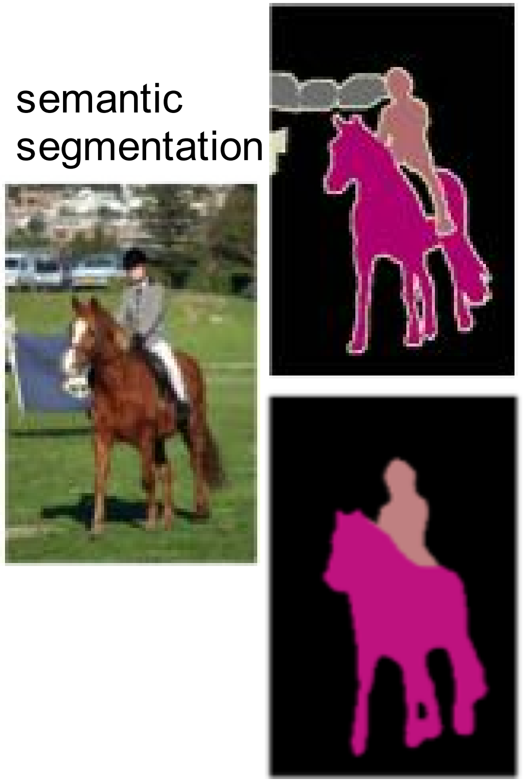
Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015

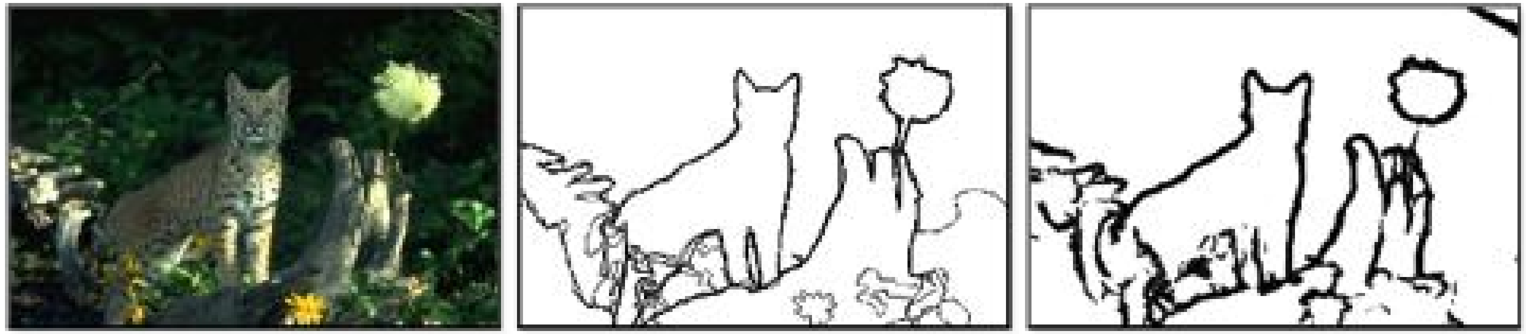
Beyond Semantic Segmentation

monocular depth estimation (Liu et al. 2015)



semantic
segmentation



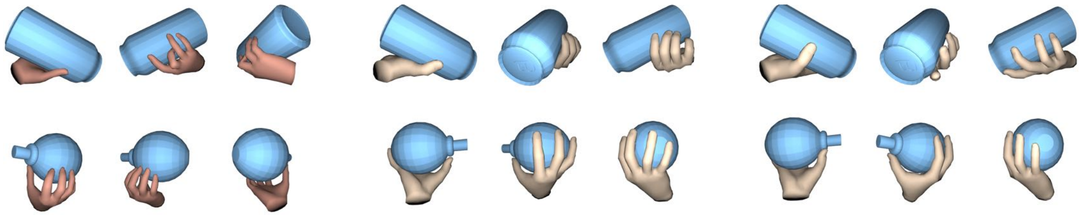


boundary prediction (Xie & Tu 2015)

Slide credit: Jonathan Long

Computer Vision and Learning Group (VLG)

Human Scene Interaction



Grasping generation

Generative human modelling

Human pose estimation