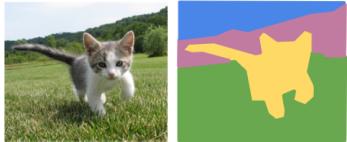
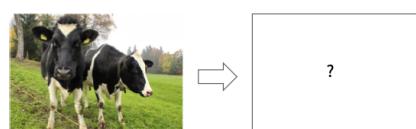




## Semantic Segmentation: The Problem



GRASS, CAT, TREE,  
SKY, ...



At test time, classify each pixel of a new image.

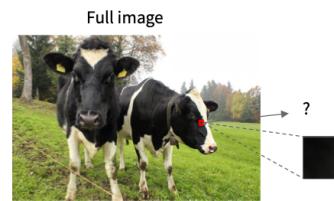
Paired training data: for each training image, each pixel is labeled with a semantic category.

Fei-Fei Li, Ehsan Adeli

Lecture 11 - 9

April 30, 2024

## Semantic Segmentation Idea: Sliding Window



Fei-Fei Li, Ehsan Adeli

Lecture 11 - 10

April 30, 2024

## Semantic Segmentation Idea: Sliding Window



Full image

Impossible to classify without context

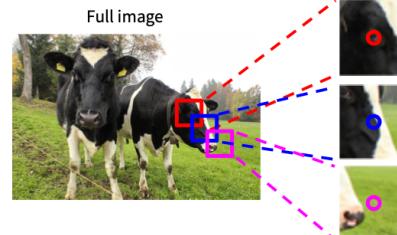
Q: how do we include context?

Fei-Fei Li, Ehsan Adeli

Lecture 11 - 11

April 30, 2024

## Semantic Segmentation Idea: Sliding Window



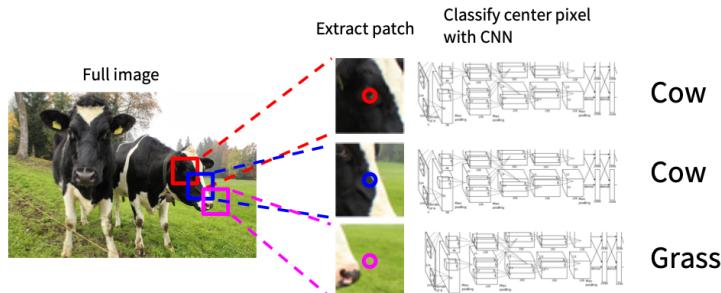
Q: how do we model this?

Fei-Fei Li, Ehsan Adeli

Lecture 11 - 12

April 30, 2024

## Semantic Segmentation Idea: Sliding Window



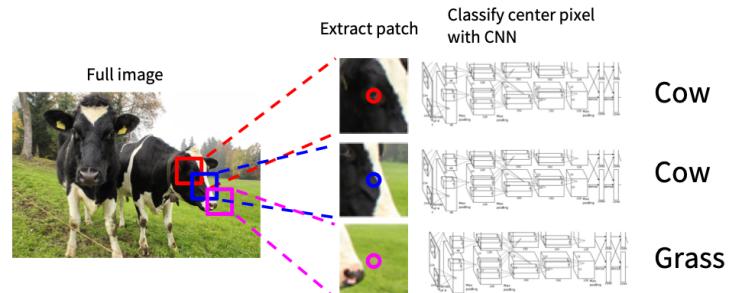
Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013  
Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

Fei-Fei Li, Ehsan Adeli

Lecture 11 - 13

April 30, 2024

## Semantic Segmentation Idea: Sliding Window



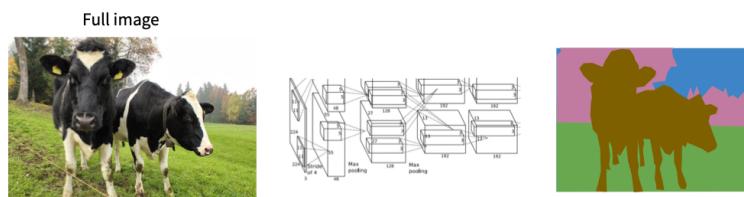
Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013  
Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

Fei-Fei Li, Ehsan Adeli

Lecture 11 - 14

April 30, 2024

## Semantic Segmentation Idea: Convolution



An intuitive idea: encode the entire image with conv net, and do semantic segmentation on top.

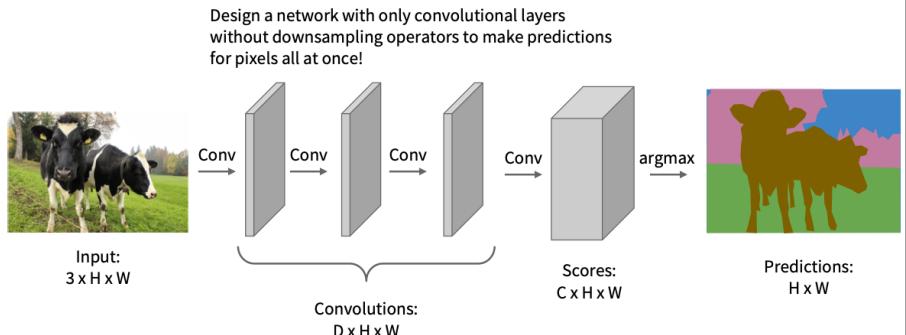
Problem: classification architectures often reduce feature spatial sizes to go deeper, but semantic segmentation requires the output size to be the same as input size.

Fei-Fei Li, Ehsan Adeli

Lecture 11 - 15

April 30, 2024

## Semantic Segmentation Idea: Fully Convolutional



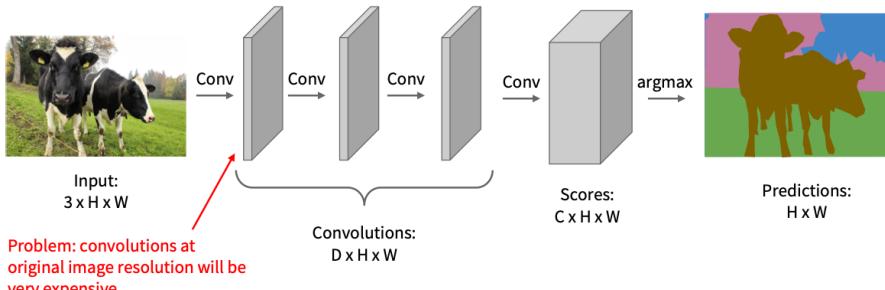
Fei-Fei Li, Ehsan Adeli

Lecture 11 - 16

April 30, 2024

## Semantic Segmentation Idea: Fully Convolutional

Design a network with only convolutional layers without downsampling operators to make predictions for pixels all at once!



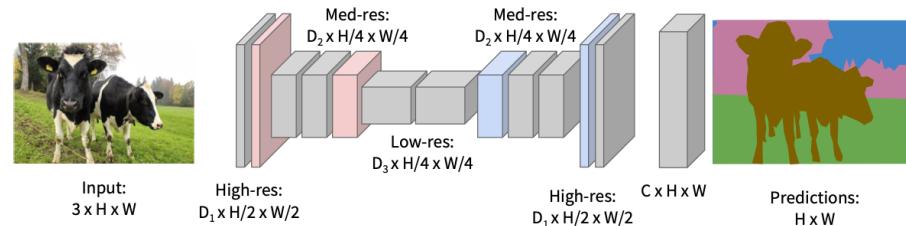
Fei-Fei Li, Ehsan Adeli

Lecture 11 - 17

April 30, 2024

## Semantic Segmentation Idea: Fully Convolutional

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



Fei-Fei Li, Ehsan Adeli

Lecture 11 - 18

April 30, 2024

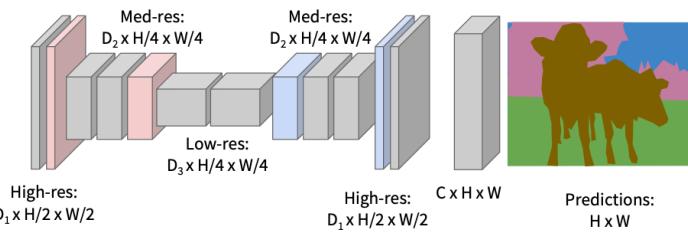
## Semantic Segmentation Idea: Fully Convolutional

Downsampling:  
Pooling, stride convolution



Input:  
 $3 \times H \times W$

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



## In-Network upsampling: “Unpooling”

Nearest Neighbor

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

Input:  $2 \times 2$

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

Output:  $4 \times 4$

“Bed of Nails”

1	2		
0	0	0	0
3	4		

Input:  $2 \times 2$

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

Output:  $4 \times 4$

Long, Shelhamer, and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015  
Noh et al., “Learning Deconvolution Network for Semantic Segmentation”, ICCV 2015

Fei-Fei Li, Ehsan Adeli

Lecture 11 - 19

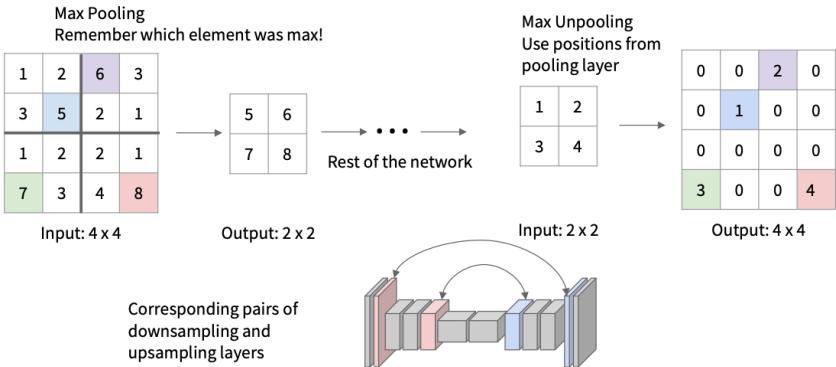
April 30, 2024

Fei-Fei Li, Ehsan Adeli

Lecture 11 - 20

April 30, 2024

## In-Network upsampling: “Max Unpooling”



Fei-Fei Li, Ehsan Adeli

Lecture 11 - 21

April 30, 2024

## Learnable Upsampling

Recall: Normal 3 x 3 convolution, stride 1 pad 1



Input: 4 x 4

Output: 4 x 4

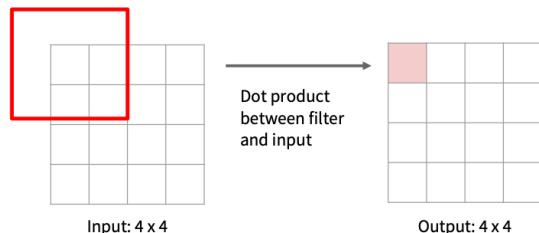
Fei-Fei Li, Ehsan Adeli

Lecture 11 - 22

April 30, 2024

## Learnable Upsampling

Recall: Normal 3 x 3 convolution, stride 1 pad 1



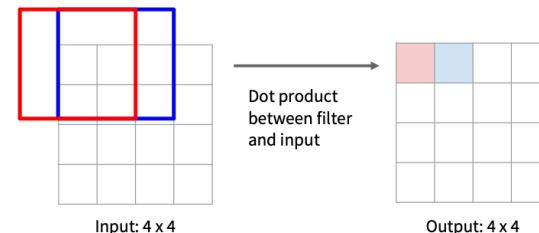
Fei-Fei Li, Ehsan Adeli

Lecture 11 - 23

April 30, 2024

## Learnable Upsampling

Recall: Normal 3 x 3 convolution, stride 1 pad 1



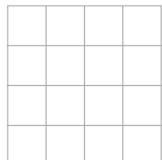
Fei-Fei Li, Ehsan Adeli

Lecture 11 - 24

April 30, 2024

## Learnable Upsampling

Recall: Normal 3 x 3 convolution, stride 2 pad 1



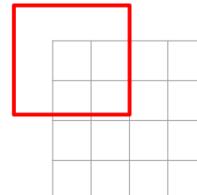
Input: 4 x 4



Output: 2 x 2

## Learnable Upsampling

Recall: Normal 3 x 3 convolution, stride 2 pad 1



Input: 4 x 4

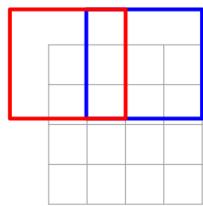
Dot product  
between filter  
and input



Output: 2 x 2

## Learnable Upsampling

Recall: Normal 3 x 3 convolution, stride 2 pad 1



Input: 4 x 4

Dot product  
between filter  
and input



Output: 2 x 2

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

Stride gives ratio between movement in input and output

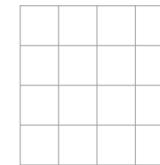
We can interpret strided convolution as “learnable downsampling”.

## Learnable Upsampling: Transposed Convolution

3 x 3 transposed convolution, stride 2 pad 1



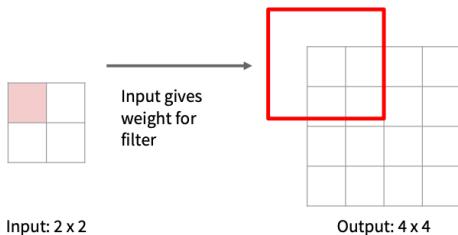
Input: 2 x 2



Output: 4 x 4

## Learnable Upsampling: Transposed Convolution

3 x 3 transposed convolution, stride 2 pad 1



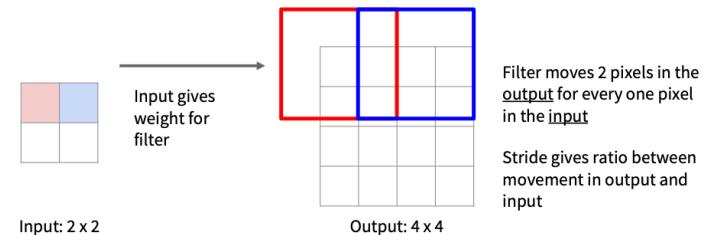
Fei-Fei Li, Ehsan Adeli

Lecture 11 - 29

April 30, 2024

## Learnable Upsampling: Transposed Convolution

3 x 3 transposed convolution, stride 2 pad 1



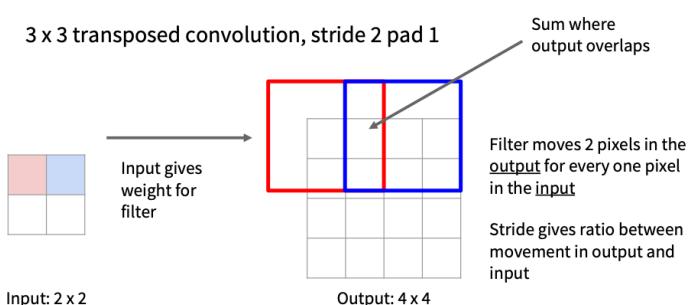
Fei-Fei Li, Ehsan Adeli

Lecture 11 - 30

April 30, 2024

## Learnable Upsampling: Transposed Convolution

3 x 3 transposed convolution, stride 2 pad 1



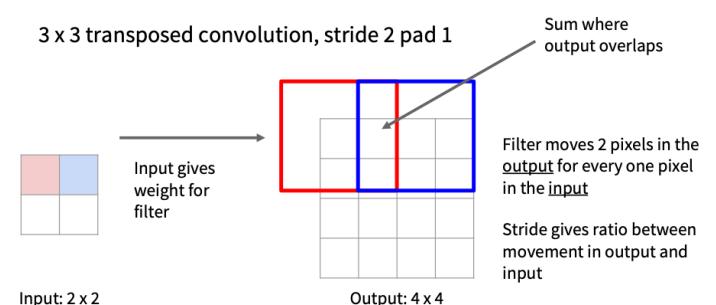
Fei-Fei Li, Ehsan Adeli

Lecture 11 - 31

April 30, 2024

## Learnable Upsampling: Transposed Convolution

3 x 3 transposed convolution, stride 2 pad 1

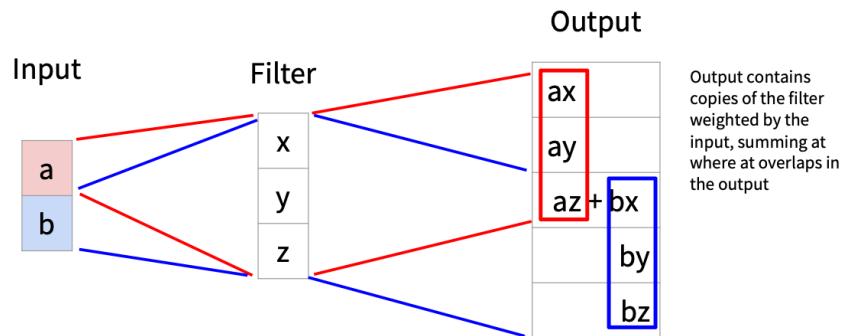


Fei-Fei Li, Ehsan Adeli

Lecture 11 - 32

April 30, 2024

## Learnable Upsampling: 1D Example

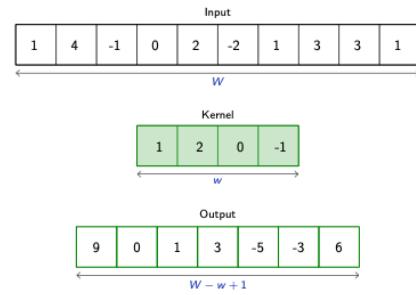


Fei-Fei Li, Ehsan Adeli

Lecture 11 - 33

April 30, 2024

## Convolution layer

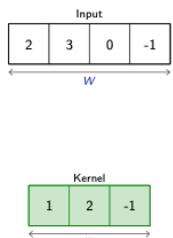


François Fleuret

Deep learning / 7.1. Transposed convolutions

5 / 14

## Transposed convolution layer

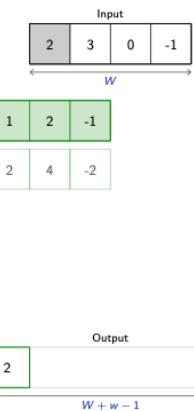


François Fleuret

Deep learning / 7.1. Transposed convolutions

6 / 14

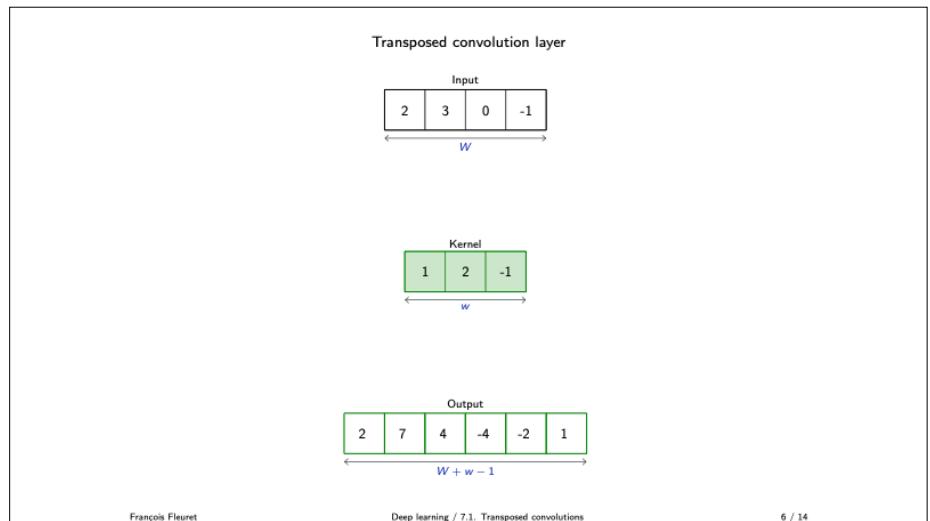
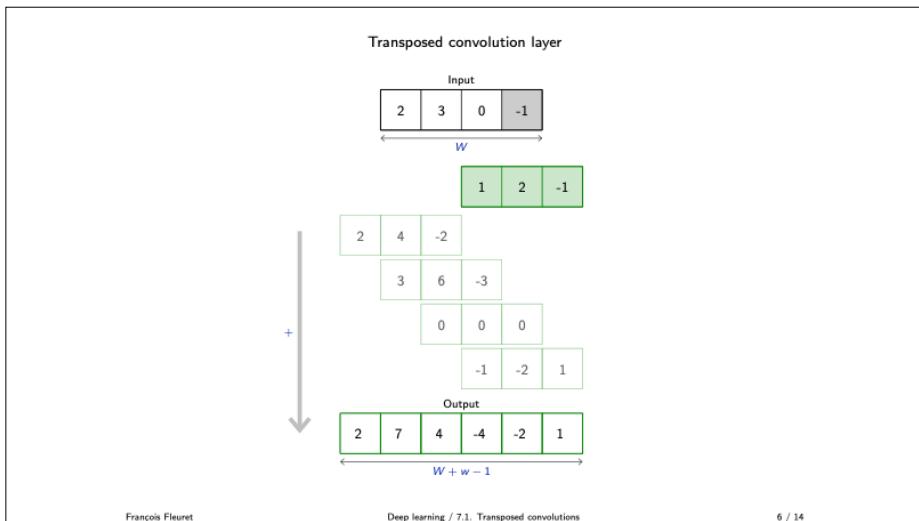
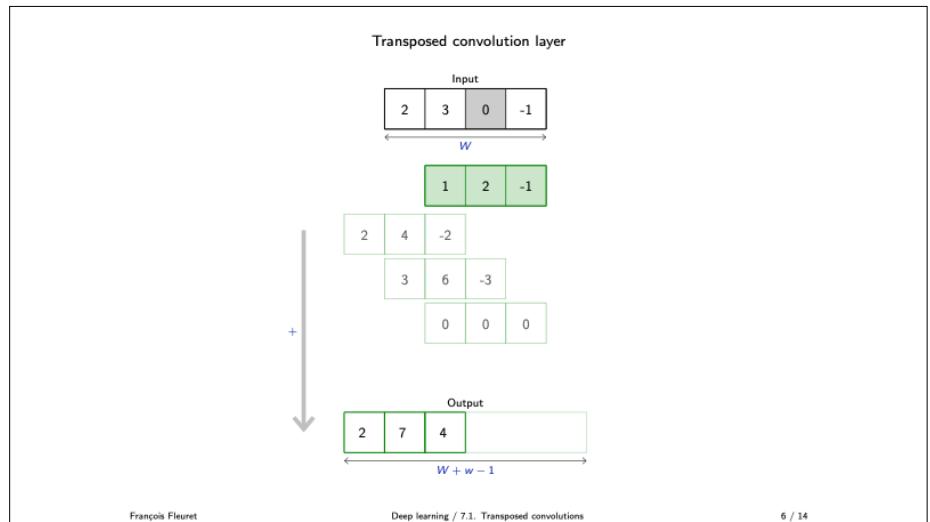
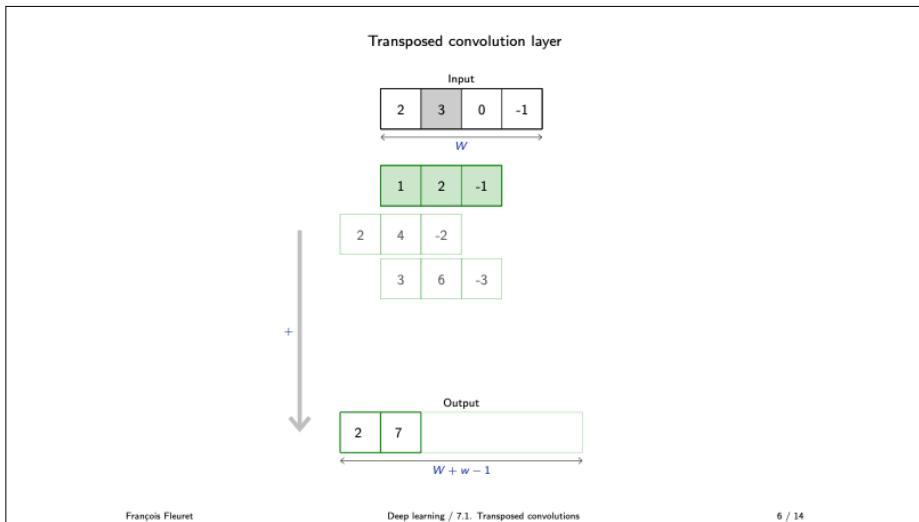
## Transposed convolution layer

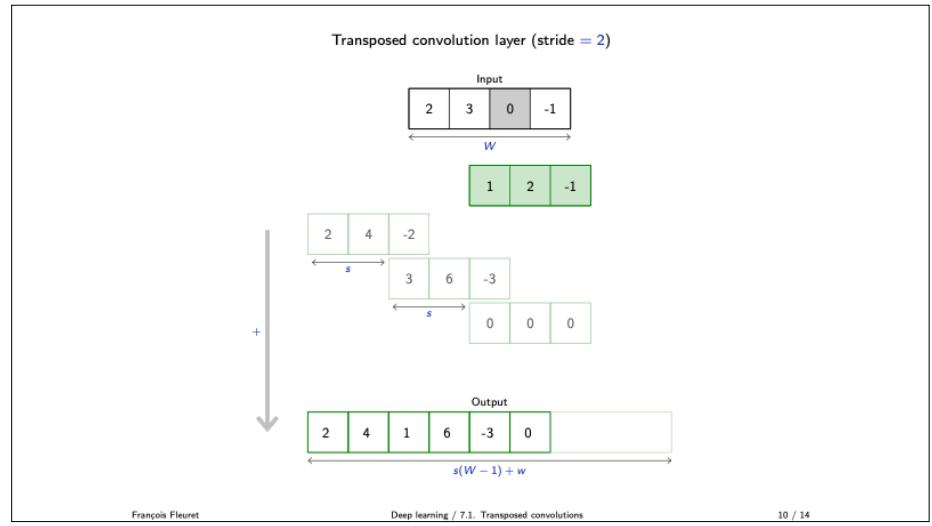
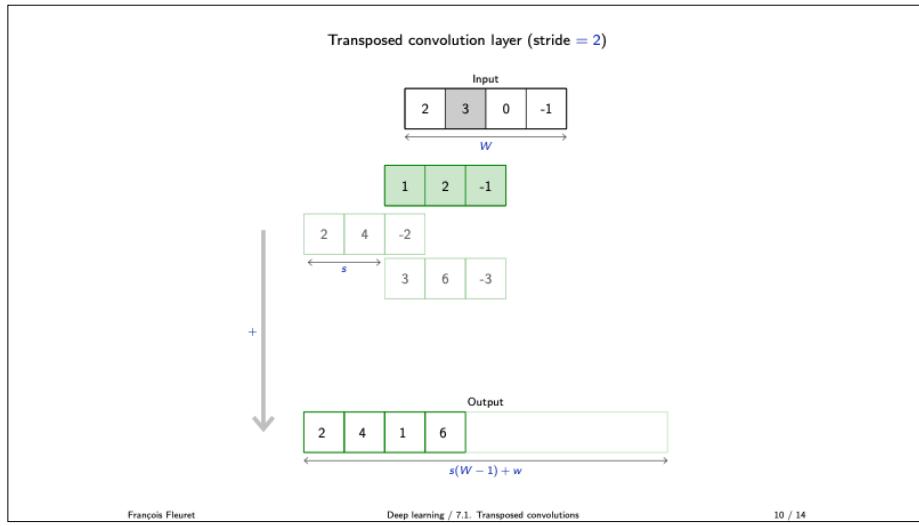
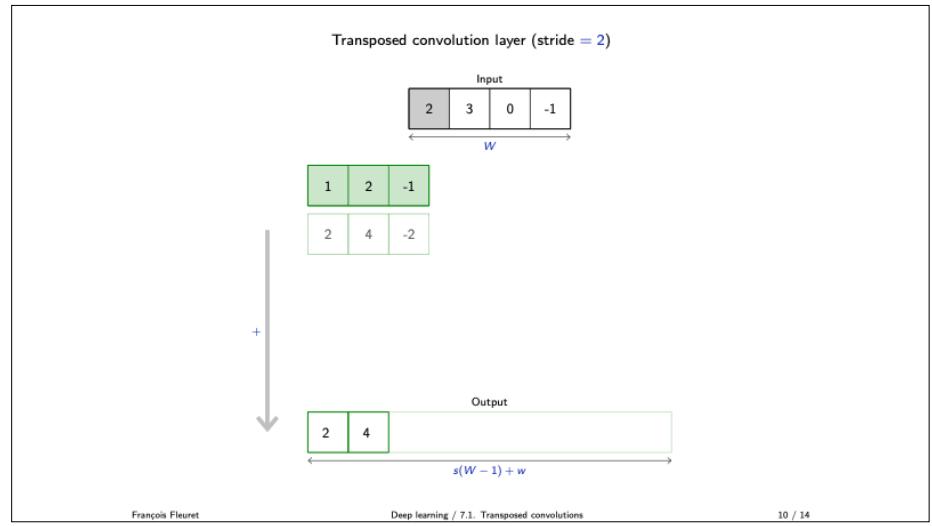
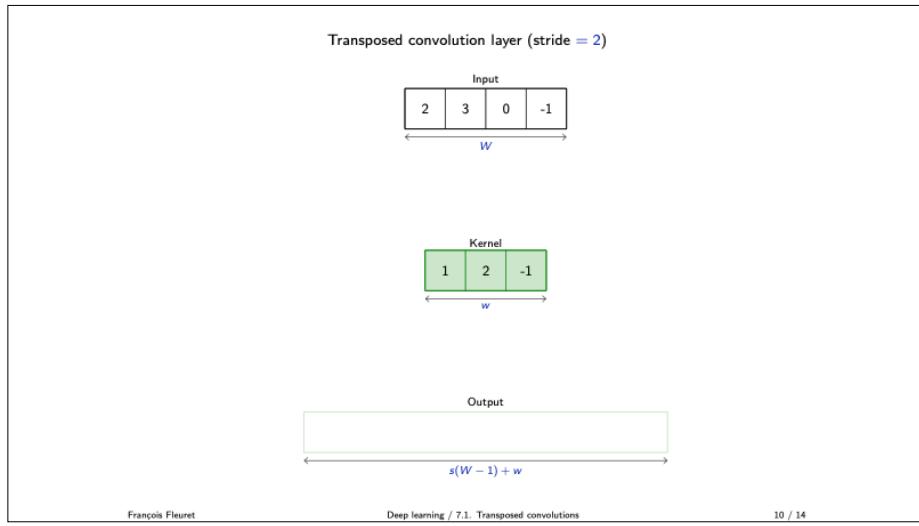


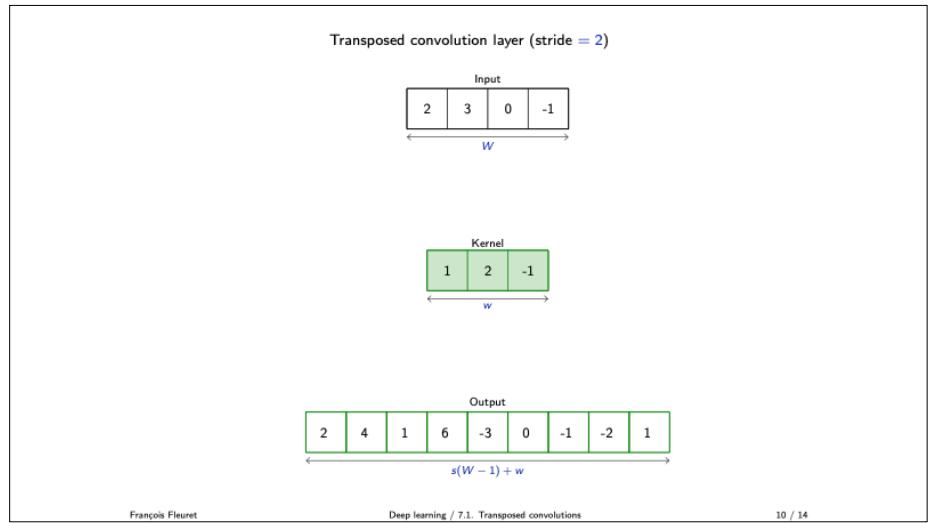
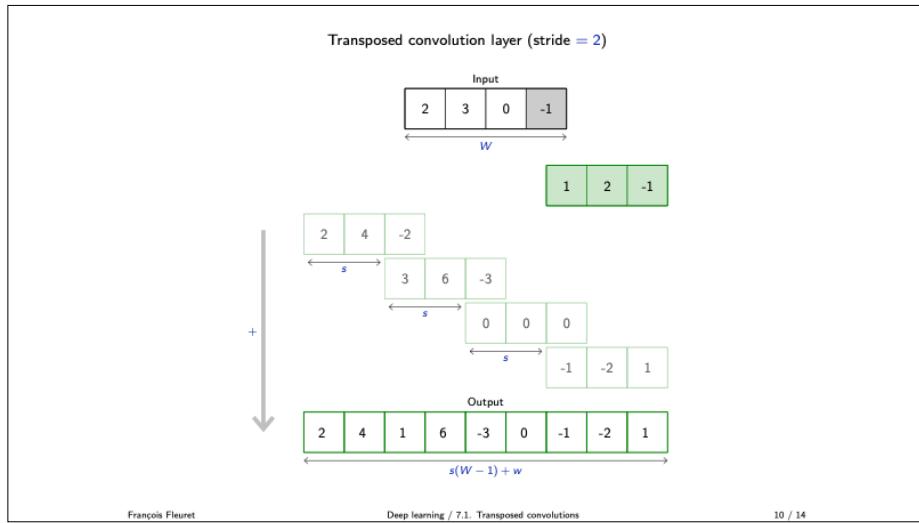
François Fleuret

Deep learning / 7.1. Transposed convolutions

6 / 14







## Convolution as Matrix Multiplication (1D Example)

We can express convolution in terms of a matrix multiplication

$$\vec{x} * \vec{a} = X\vec{a}$$

$$\begin{bmatrix} x & y & z & 0 & 0 & 0 \\ 0 & 0 & x & y & z & 0 \end{bmatrix} \begin{bmatrix} 0 \\ a \\ b \\ c \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} ay + bz \\ bx + cy + dz \end{bmatrix}$$

Example: 1D conv, kernel size=3, stride=2, padding=1

## Convolution as Matrix Multiplication (1D Example)

We can express convolution in terms of a matrix multiplication

$$\vec{x} * \vec{a} = X\vec{a}$$

$$\begin{bmatrix} x & y & z & 0 & 0 & 0 \\ 0 & 0 & x & y & z & 0 \end{bmatrix} \begin{bmatrix} 0 \\ a \\ b \\ c \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} ay + bz \\ bx + cy + dz \end{bmatrix}$$

Example: 1D conv, kernel size=3, stride=2, padding=1

Transposed convolution multiplies by the transpose of the same matrix:

$$\vec{x} *^T \vec{a} = X^T \vec{a}$$

$$\begin{bmatrix} x & 0 \\ y & 0 \\ z & x \\ 0 & y \\ 0 & z \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} ax \\ ay \\ az + bx \\ by \\ bz \\ 0 \end{bmatrix}$$

Example: 1D transposed conv, kernel size=3, stride=2, padding=0

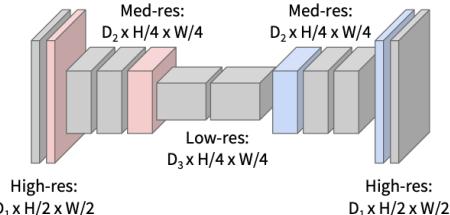
## Semantic Segmentation Idea: Fully Convolutional

Downsampling:  
Pooling, strided convolution



Input:  
 $3 \times H \times W$

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



Upsampling:  
Unpooling or strided transposed convolution



Predictions:  
 $H \times W$

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015  
Noh et al., "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Fei-Fei Li, Ehsan Adeli

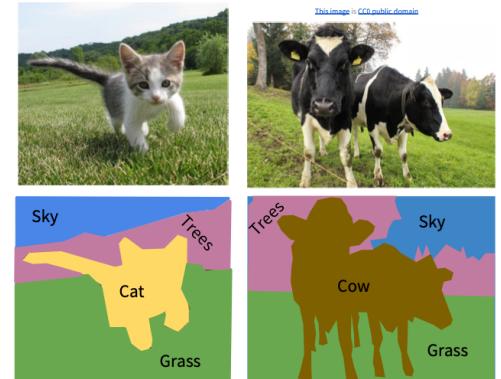
Lecture 11 - 36

April 30, 2024

## Semantic Segmentation

Label each pixel in the image with a category label

Don't differentiate instances, only care about pixels



Fei-Fei Li, Ehsan Adeli

Lecture 11 - 38

April 30, 2024