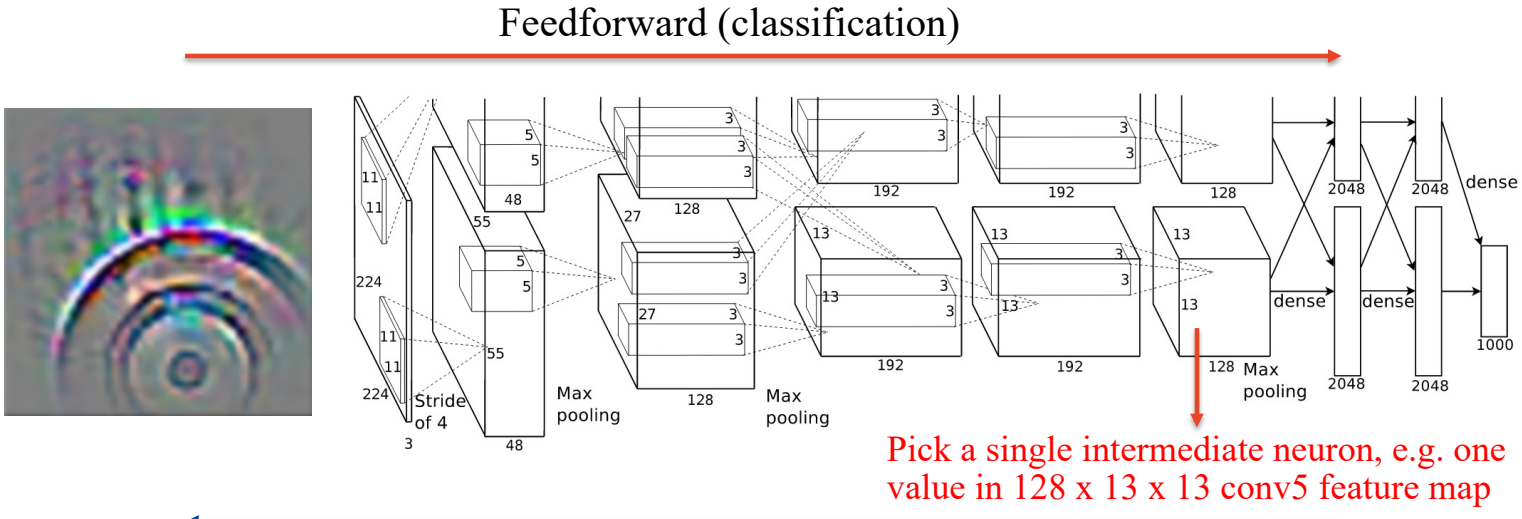
Extension

Visualize features

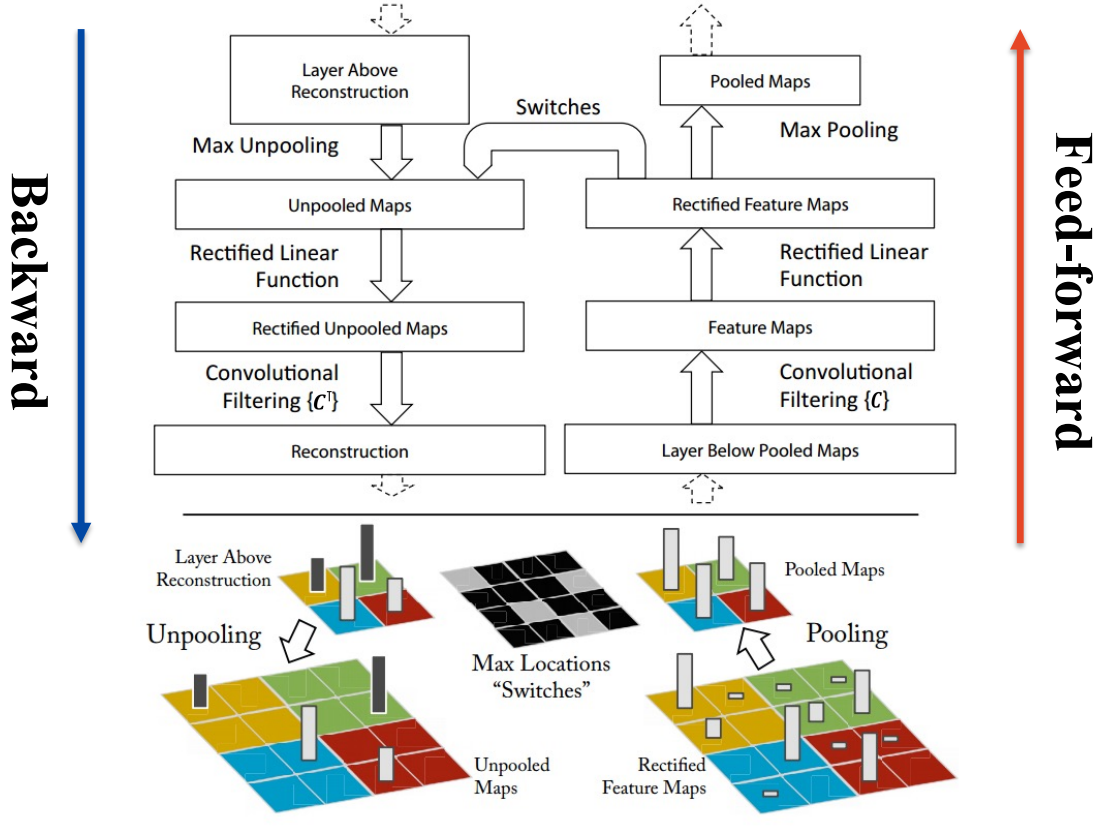
Zeiler and Fergus, ECCV 2014

- A non-parametric view of invariance, showing which patterns from the training set activate the feature map
- Using deconvolution to project the feature activations back to the input pixel space



Deconvolutional Method

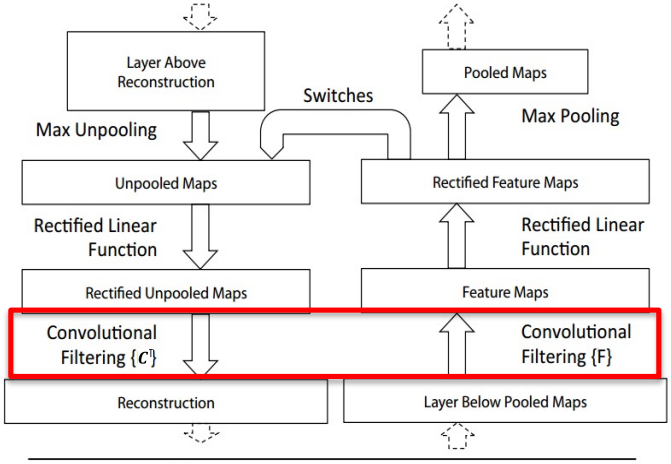
- For one convolutional layer.



(Zeiler and Fergus, 2014)

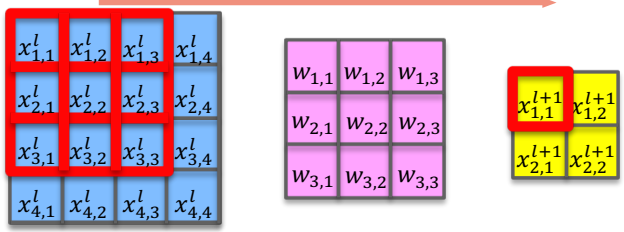
Deconvolutional Method

- Conv and Deconv
  - Use **learned filters** $\{\boldsymbol{C}\}$ to convolve forward signal, but use transposed versions $\{\boldsymbol{C}^{\mathsf{T}}\}$ for backward signal.



(Zeiler and Fergus, 2014)

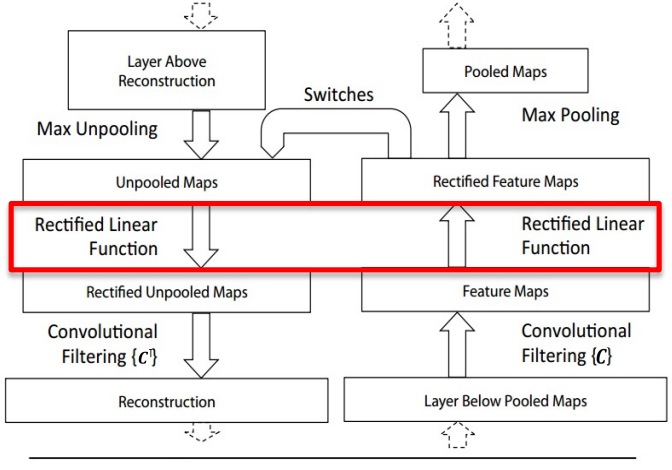
$$\boldsymbol{C}^{l-1} \times (x_{1,1}^{l-1}, \cdots, x_{4,4}^{l-1})^{\mathsf{T}} = (x_{1,1}^{l}, \cdots, x_{2,2}^{l})^{\mathsf{T}}$$



$$\boldsymbol{C}^{\mathsf{T}} \times (x_{1,1}^{l-1}, \cdots, x_{2,2}^{l-1})^{\mathsf{T}} = (x_{1,1}^{l}, \cdots, x_{4,4}^{l})^{\mathsf{T}}$$

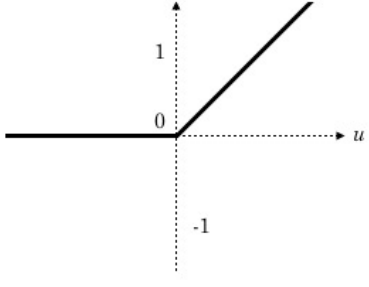
Deconvolutional Method

- Activation function
  - Both feedforward and backward processes employ ReLU as activation functions to ensure feature maps are always positive.



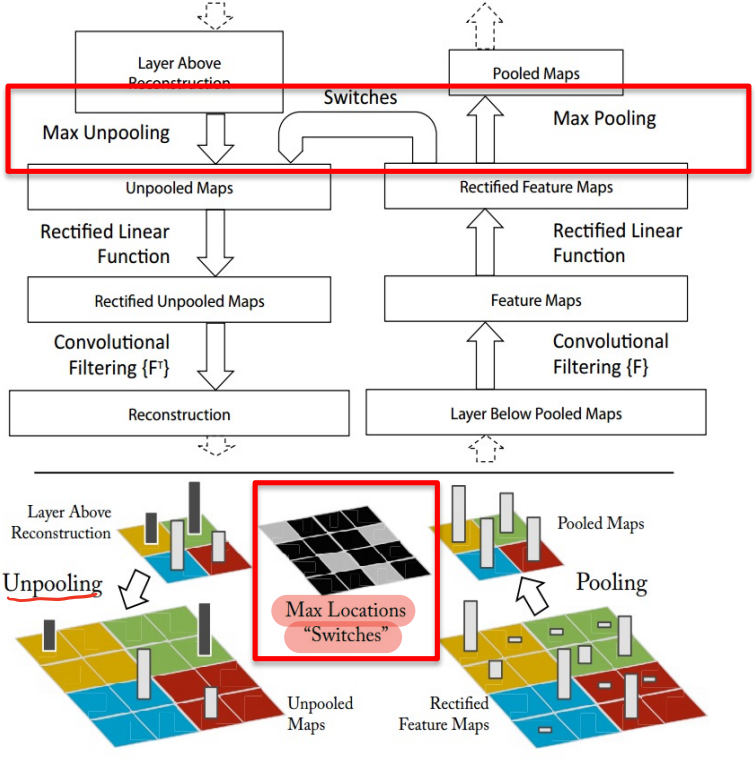
(Zeiler and Fergus, 2014)

$$ReLU(x) = \max(0, x)$$



Deconvolutional Method

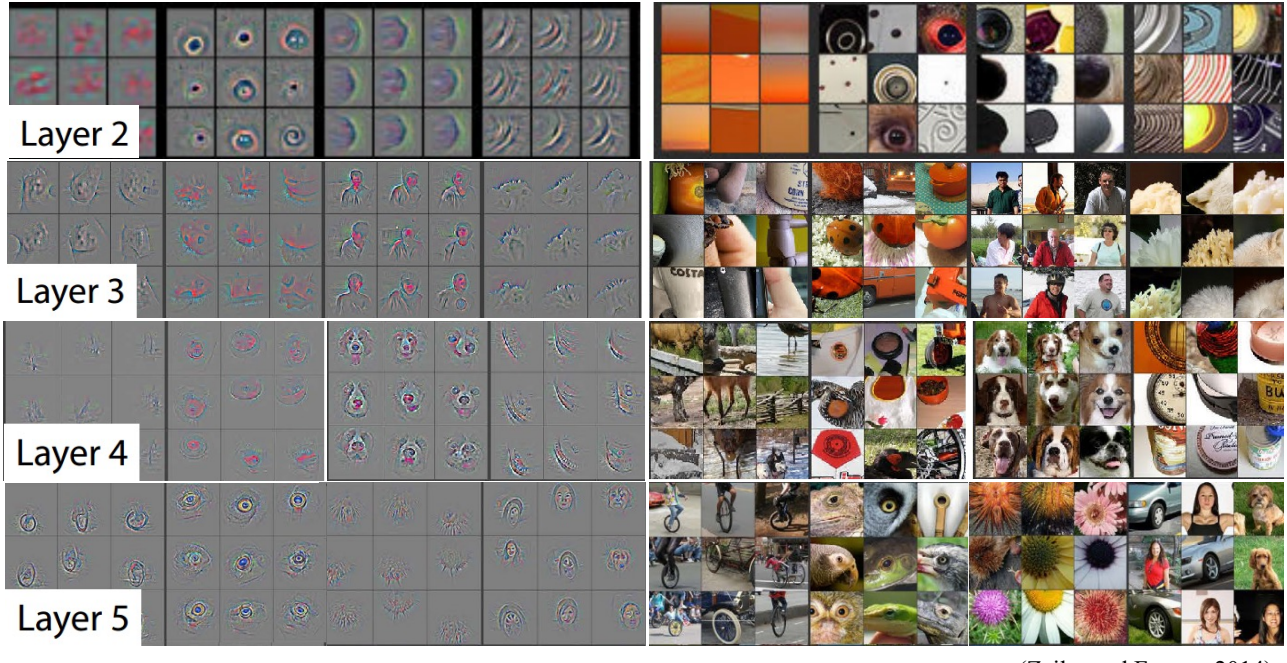
- Pooling and Unpooling
  - Despite the non-invertibility of max pooling, an approximate inverse could be obtained by recording the locations of the maxima within each pooling region in a set of **switch variables**.



(Zeiler and Fergus, 2014)

Visualize features

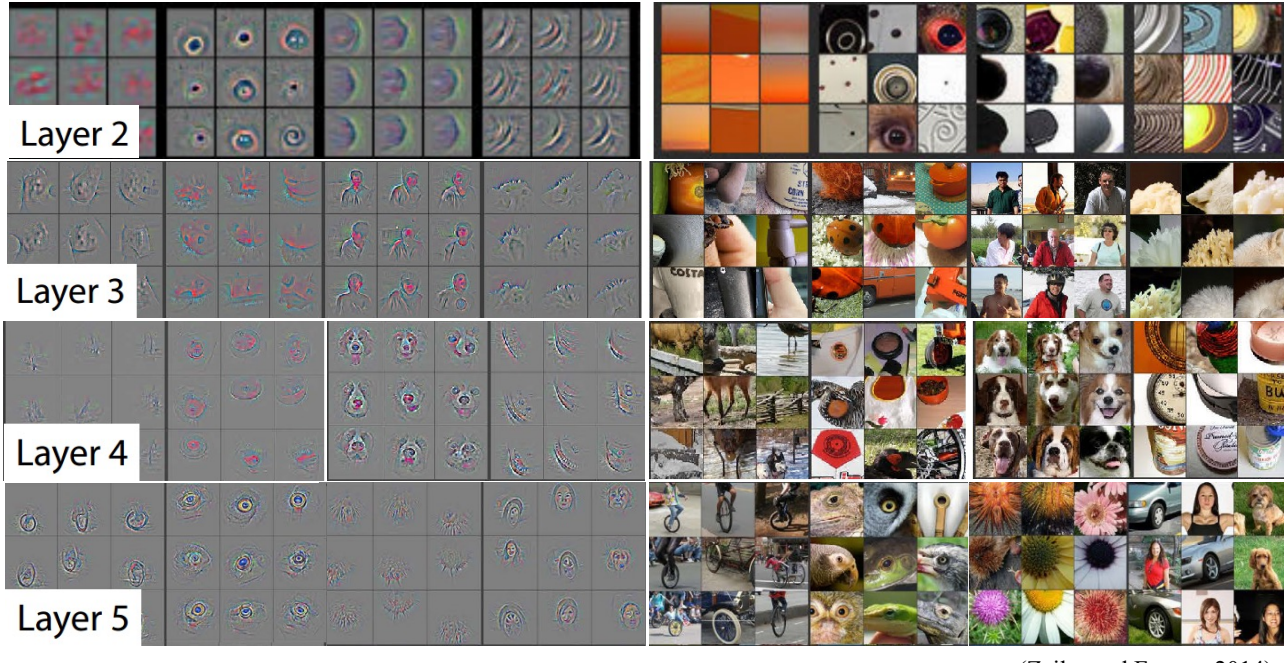
- Instead of showing the single strongest activation for a given feature map, the article shows the top 9 activations layer 2-5.
- Alongside these visualizations are the corresponding image patches.



(Zeiler and Fergus, 2014)

Visualize features

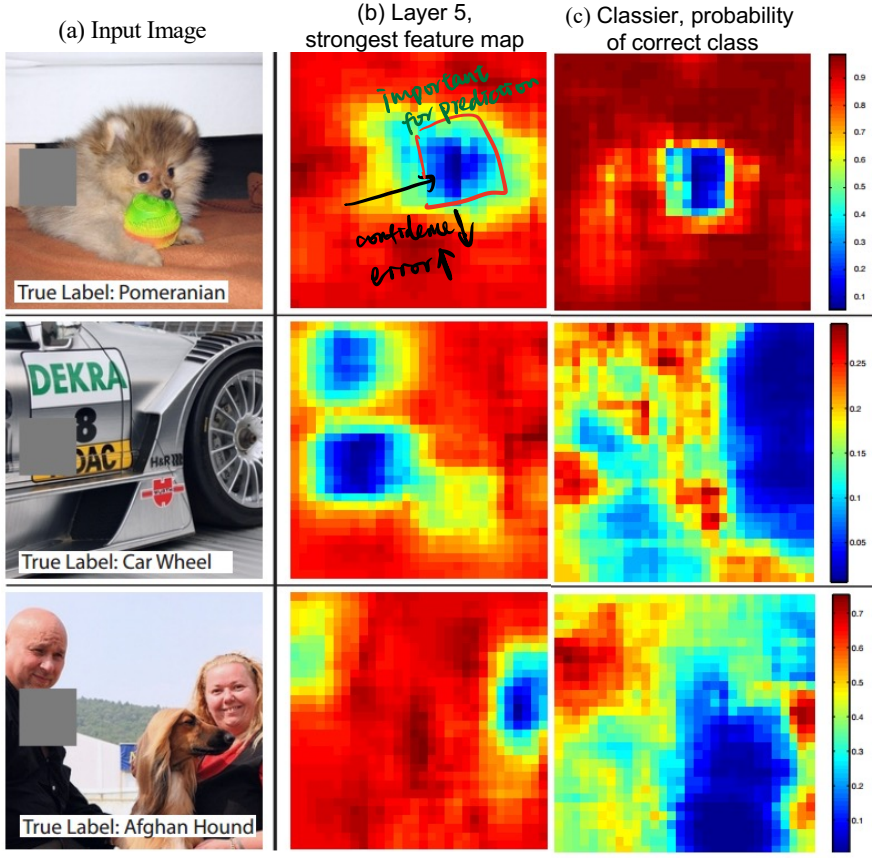
- Layer 2 responds to corners and other edge/color conjunctions.
- Layer 3 has more complex invariances, capturing similar textures (e.g. mesh patterns).
- Layer 4 shows significant variation, but is more class-specific.
- Layer 5 shows entire objects with significant pose variation.



(Zeiler and Fergus, 2014)

Occlusion Experiments

- Occlude images at different locations and visualize feature activations and classifier confidence



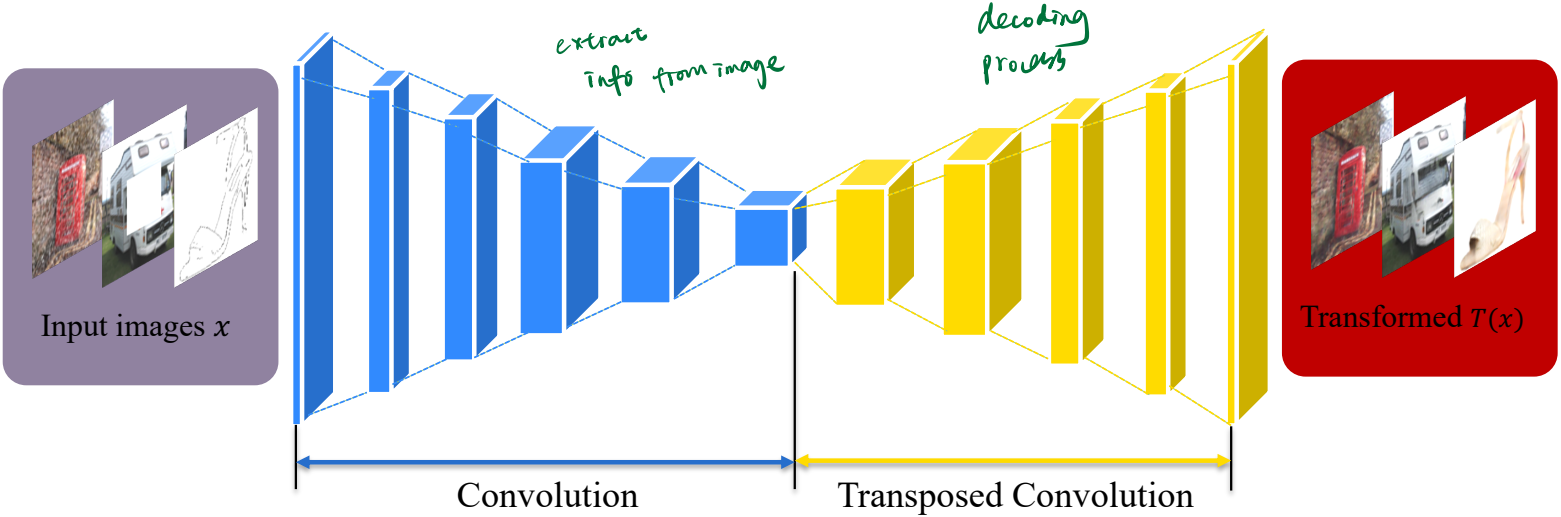
classification error ↑

(Zeiler and Fergus, 2014)

Variations of CNNs

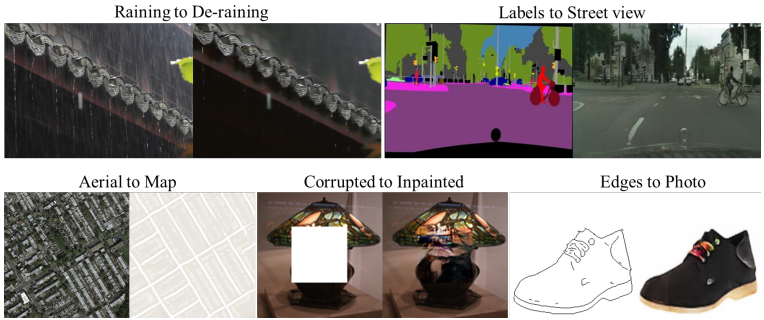
Transposed Convolution

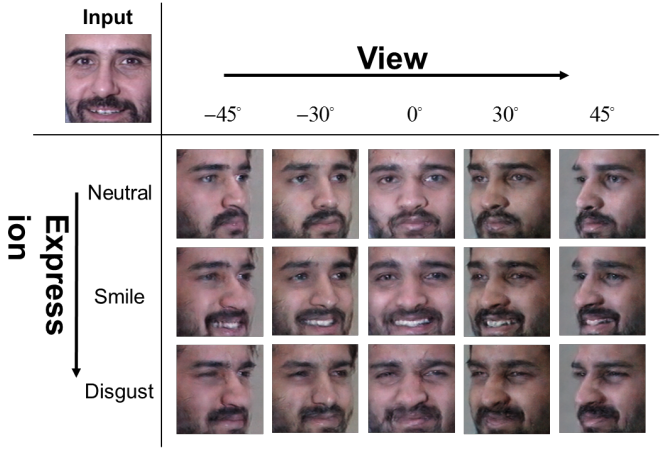
- Transposed Convolution
  - Also named “Deconvolution”, “up-convolution” …
  - Usually, the output is an image
  - Different from the above ‘Deconvolutional method’ for visualizing CNNs learned features

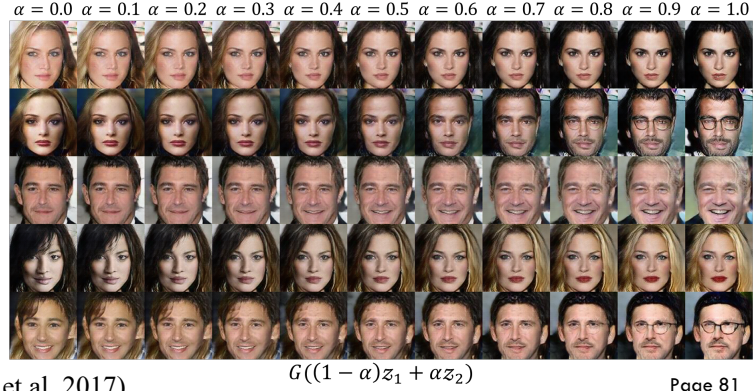


Transposed Convolution

- Applications
  - Generative model, encoder-decoder, image editing, image transformation …



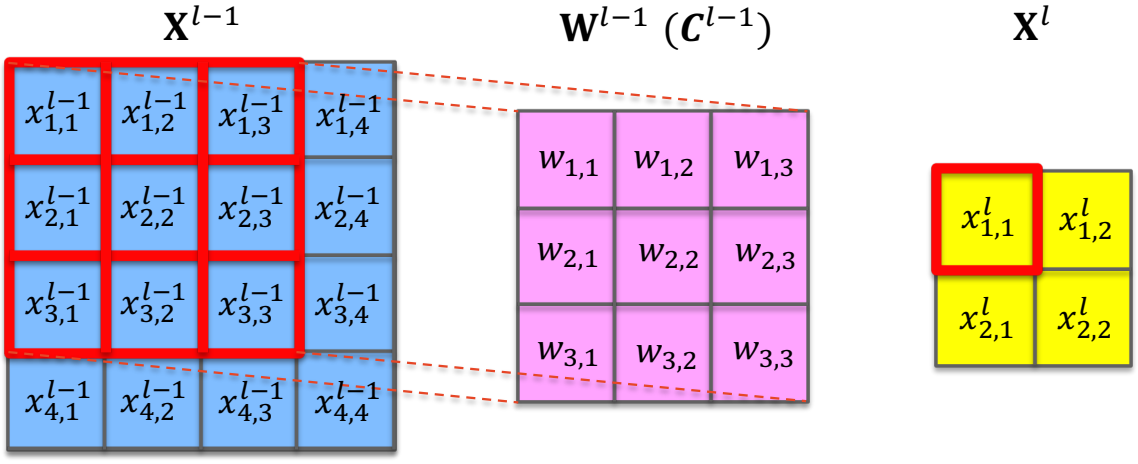




(Chaoyue et al, 2017)

Transposed Convolution

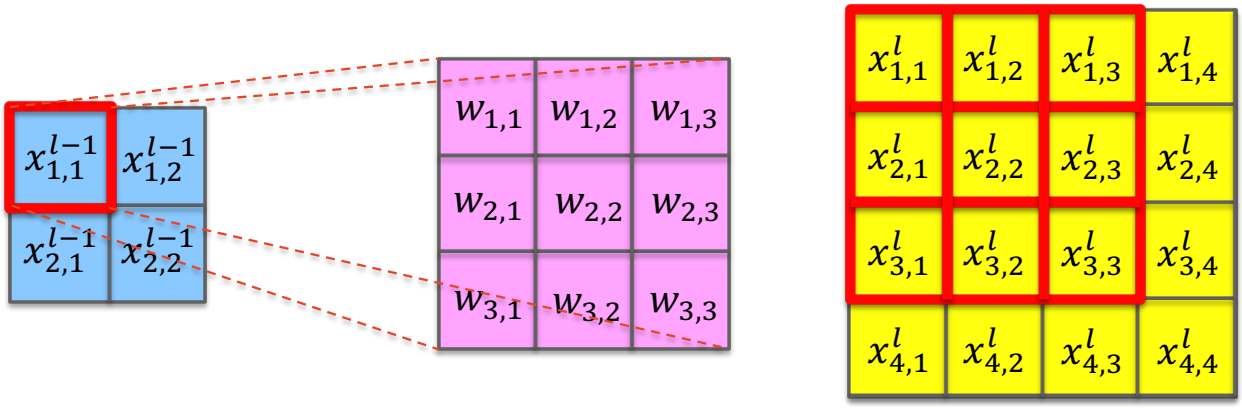
- Compare to aforementioned convolution operation, can we go the other way around, i.e., map from a 4-dim space to a 16-dim space, while keeping the connectivity pattern of the original convolution.



Convolution operation

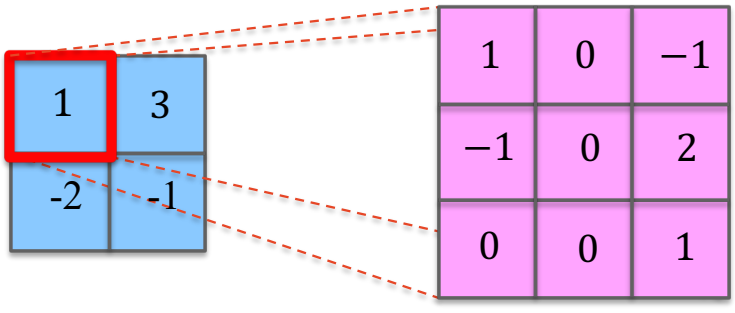
Transposed Convolution

- Compare to aforementioned convolution operation, can we go the other way around, i.e., map from a 4-dim space to a 16-dim space, while keeping the connectivity pattern of the original convolution.



Transposed Convolution

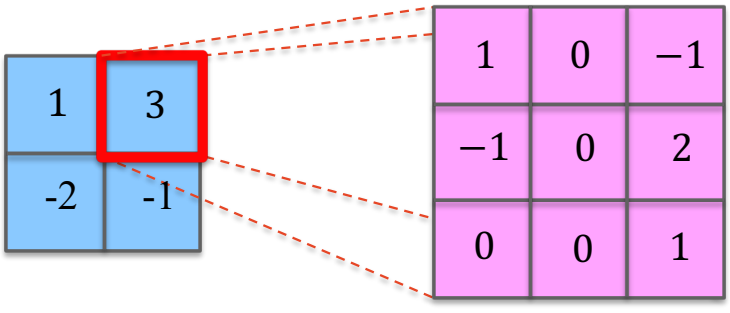
Transposed Convolution



1	0	−1	
−1	0	2	
0	0	1	

1	0	−1	
−1	0	2	
0	0	1	

Transposed Convolution

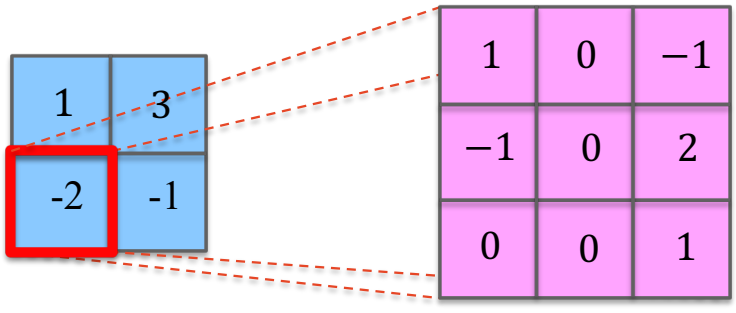


1	0	−1	−3
−1	−3	2	6
0	0	1	3

1	0	−1	
−1	0	2	
0	0	1	

	3	0	−3
	−3	0	6
	0	0	3

Transposed Convolution



1	3	−1	−3
−1	−3	4	6
0	0	−1	3
0	0	−1	

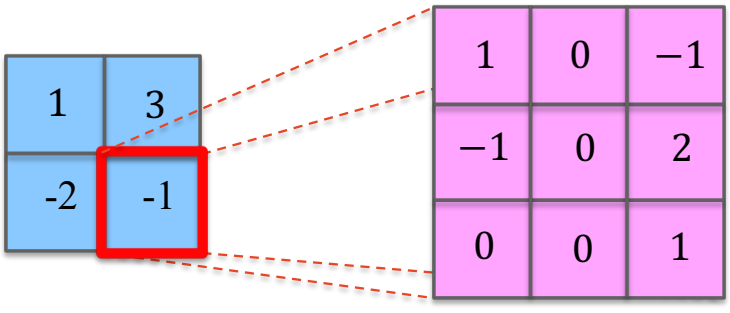
1	0	−1	
−1	0	2	
0	0	1	

	3	0	−3
	−3	0	6
	0	0	3

−2	0	2	
2	0	−4	
0	0	−2	

Transposed Convolution

overlap → use Σ



1	3	−1	−3
−3	−4	4	6
2	0	−3	3
0	0	−1	2

1	0	−1	
−1	0	2	
0	0	1	

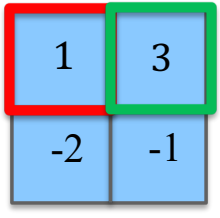
	3	0	−3
	−3	0	6
	0	0	3

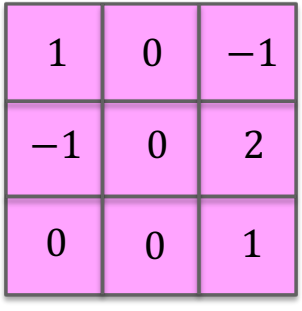
−2	0	2	
2	0	−4	
0	0	−2	

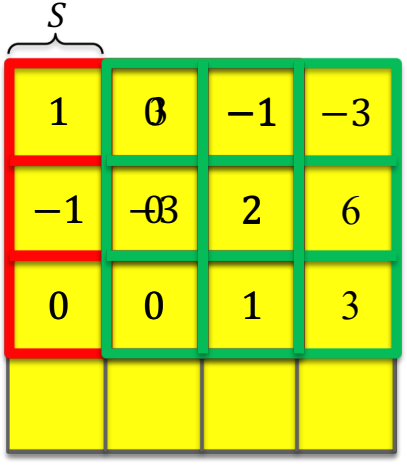
	−1	0	1
	1	0	−2
	0	0	−1

Transposed Convolution

- Stride



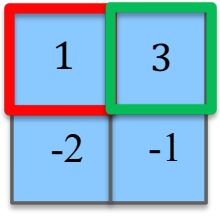


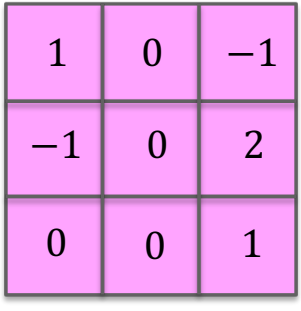


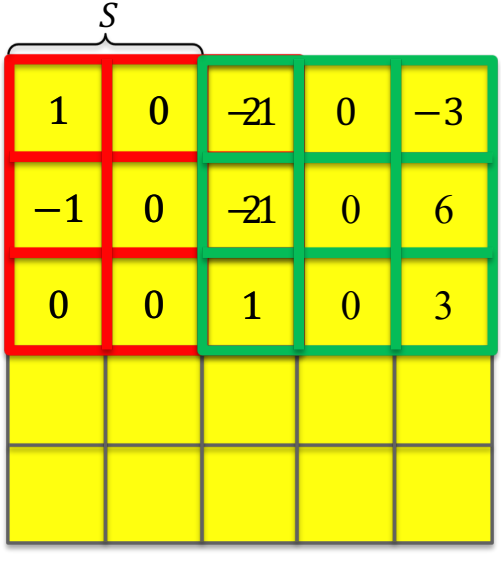
Stride = 1

Transposed Convolution

- Stride



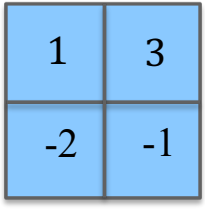


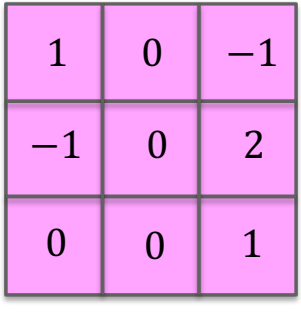


Stride = 2

Transposed Convolution

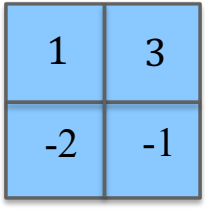
- Cropping

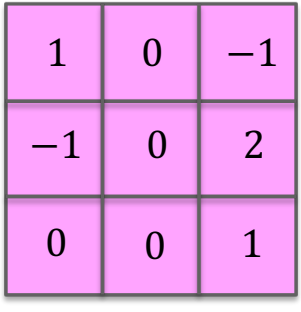


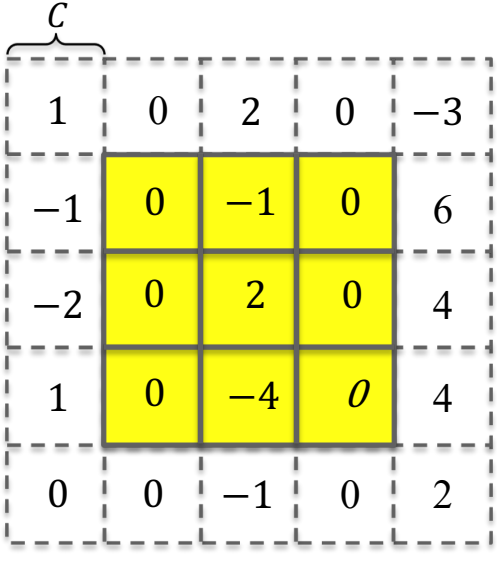


Transposed Convolution

- Cropping





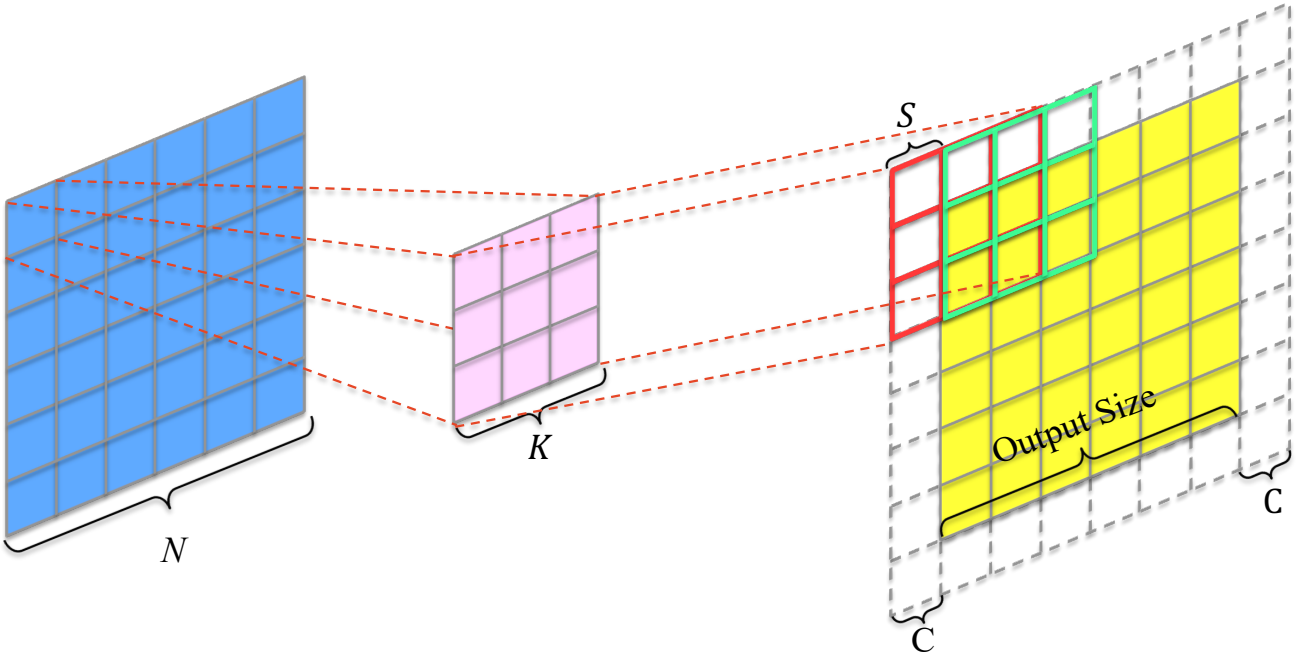


Crop = 1 ↔ padding

Transposed Convolution

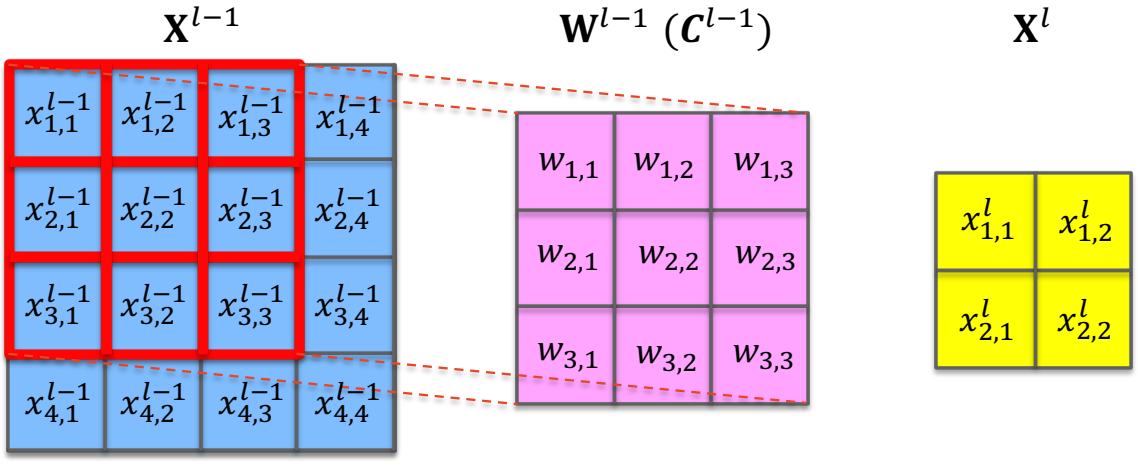
- Output Size

$$\text{Output Size} = (N-1)S + K - 2C$$



Convolution as a matrix operation

- Recall the matrix operation of original convolution



$$\boldsymbol{C}^{l-1} \times (x_{1,1}^{l-1}, \cdots, x_{4,4}^{l-1})^{\mathsf{T}} = (x_{1,1}^{l}, \cdots, x_{2,2}^{l})^{\mathsf{T}}$$

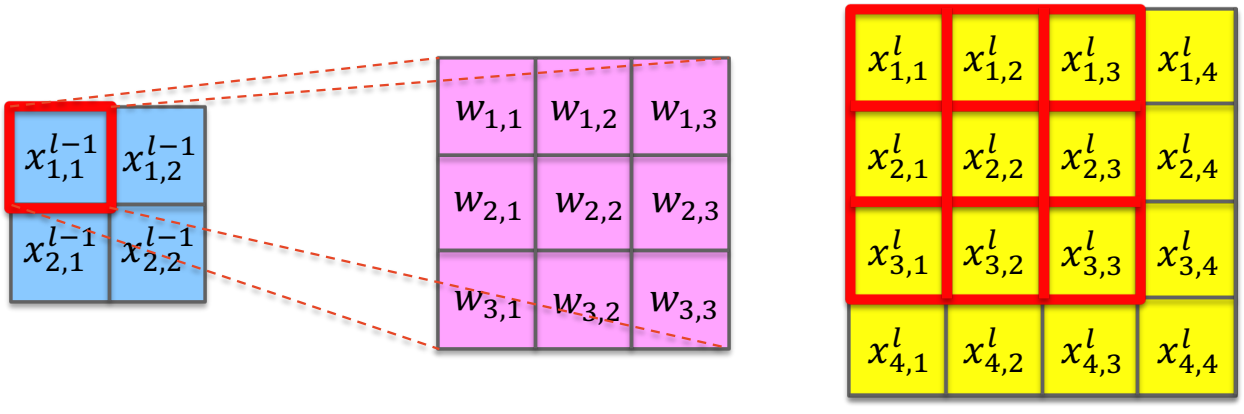
$$\boldsymbol{C}^{l-1} = \begin{pmatrix} w_{1,1} & w_{1,2} & w_{1,3} & 0 & w_{2,1} & w_{2,2} & w_{2,3} & 0 & w_{3,1} & w_{3,2} & w_{3,3} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{1,1} & w_{1,2} & w_{1,3} & 0 & w_{2,1} & w_{2,2} & w_{2,3} & 0 & w_{3,1} & w_{3,2} & w_{3,3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{1,1} & w_{1,2} & w_{1,3} & 0 & w_{2,1} & w_{2,2} & w_{2,3} & 0 & w_{3,1} & w_{3,2} & w_{3,3} & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{1,1} & w_{1,2} & w_{1,3} & 0 & w_{2,1} & w_{2,2} & w_{2,3} & 0 & w_{3,1} & w_{3,2} & w_{3,3} \end{pmatrix}$$

Transposed Convolution as a matrix operation

- Given the same matrix C, transposed convolution can be represented as:

$$\boldsymbol{C}^{\intercal} \times (x_{1,1}^{l-1}, \cdots, x_{2,2}^{l-1})^{\intercal} = (x_{1,1}^{l}, \cdots, x_{4,4}^{l})^{\intercal}$$

Forward pass of transposed convolution

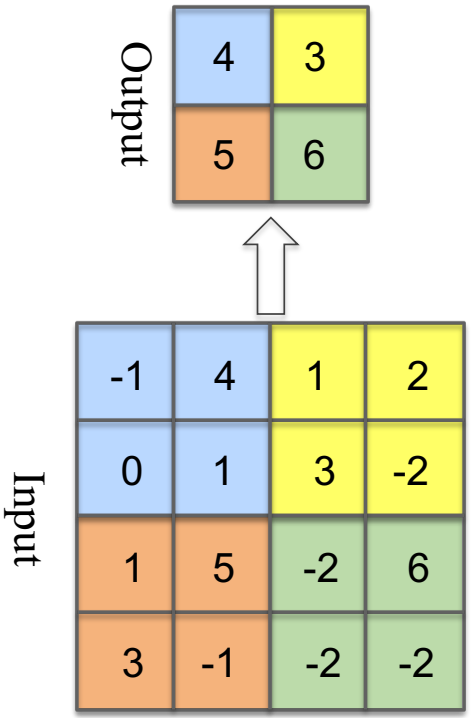


$$C = \begin{pmatrix} w_{1,1} & w_{1,2} & w_{1,3} & 0 & w_{2,1} & w_{2,2} & w_{2,3} & 0 & w_{3,1} & w_{3,2} & w_{3,3} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{1,1} & w_{1,2} & w_{1,3} & 0 & w_{2,1} & w_{2,2} & w_{2,3} & 0 & w_{3,1} & w_{3,2} & w_{3,3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{1,1} & w_{1,2} & w_{1,3} & 0 & w_{2,1} & w_{2,2} & w_{2,3} & 0 & w_{3,1} & w_{3,2} & w_{3,3} & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{1,1} & w_{1,2} & w_{1,3} & 0 & w_{2,1} & w_{2,2} & w_{2,3} & 0 & w_{3,1} & w_{3,2} & w_{3,3} \end{pmatrix}$$

Max pooling & Up-pooling

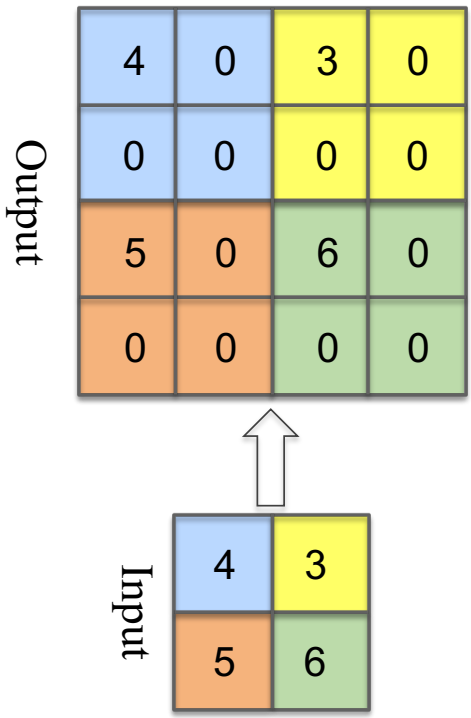
Max pooling

- Pooling ops: max(·)



Up-pooling

- Fill numbers to pre −defined position, others fill zeros.



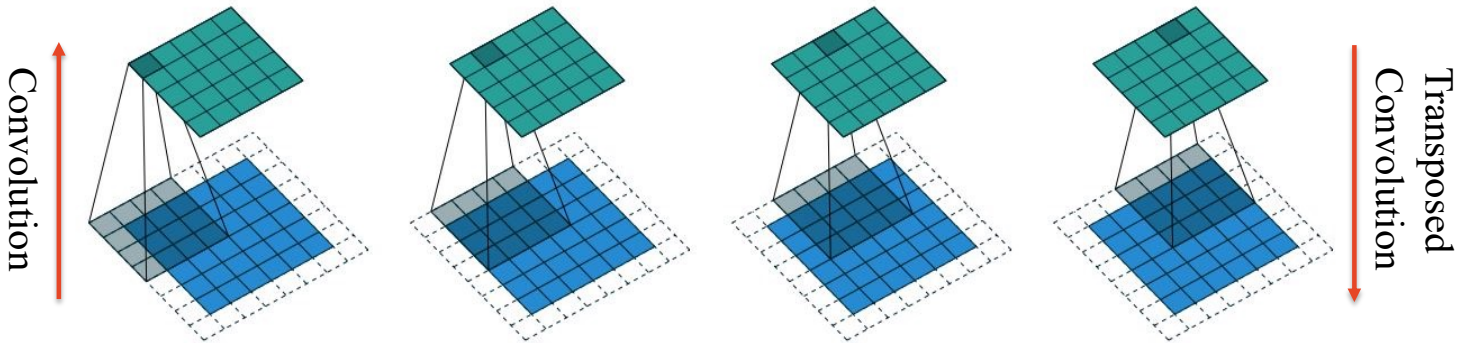
Transposed Convolution

Convolution

- From image to features
- Kernel: $n \times m \rightarrow 1$
- Padding
- $\cdots \rightarrow$ Filtering $\rightarrow$ Activate function $\rightarrow$ Pooling $\rightarrow \cdots$

Transposed convolution

- From features to image
- Kernel: $1 \rightarrow n \times m$
- Cropping
- $\cdots \rightarrow$ Filtering $\rightarrow$ Activate function $\rightarrow$ Up−pooling $\rightarrow \cdots$



https://github.com/vdumoulin/conv_arithmetic

Thank you!