Semantic Segmentation

Semantic Segmentation

Label every pixel!

Don't differentiate instances (cows)

Classic computer vision problem

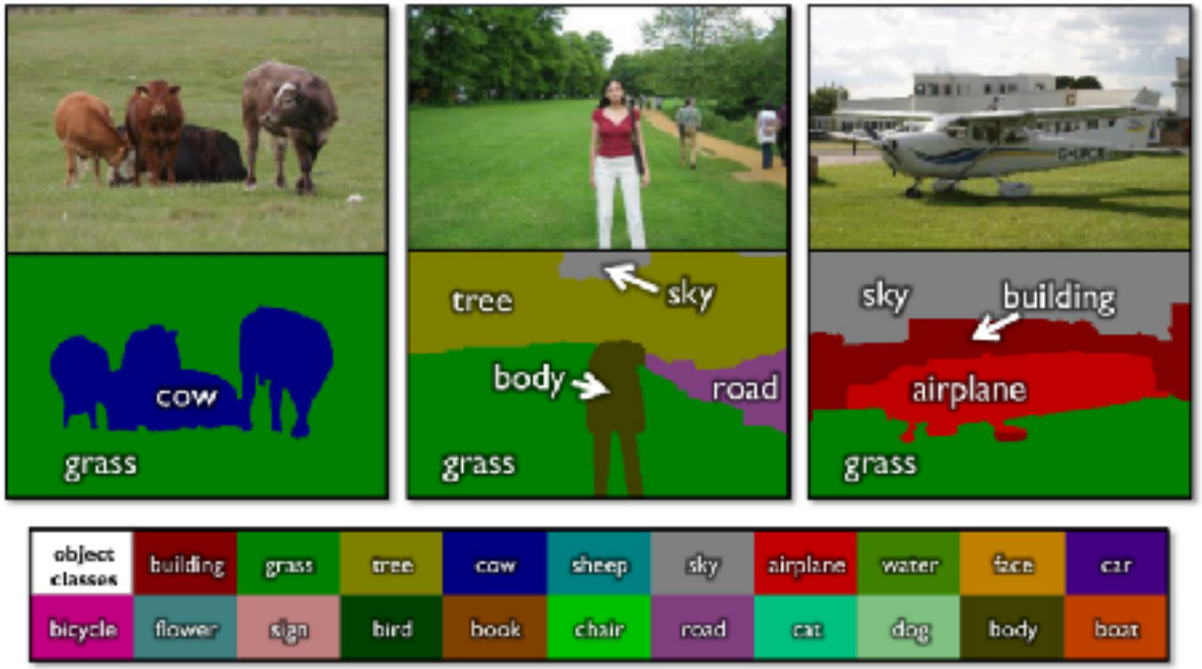


Figure credit: Shotton et al, "TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context", IJCV 2007

Semantic Segmentation

Detect instances, give category, label pixels

"simultaneous detection and segmentation" (SDS)

Lots of recent work (MS-COCO)

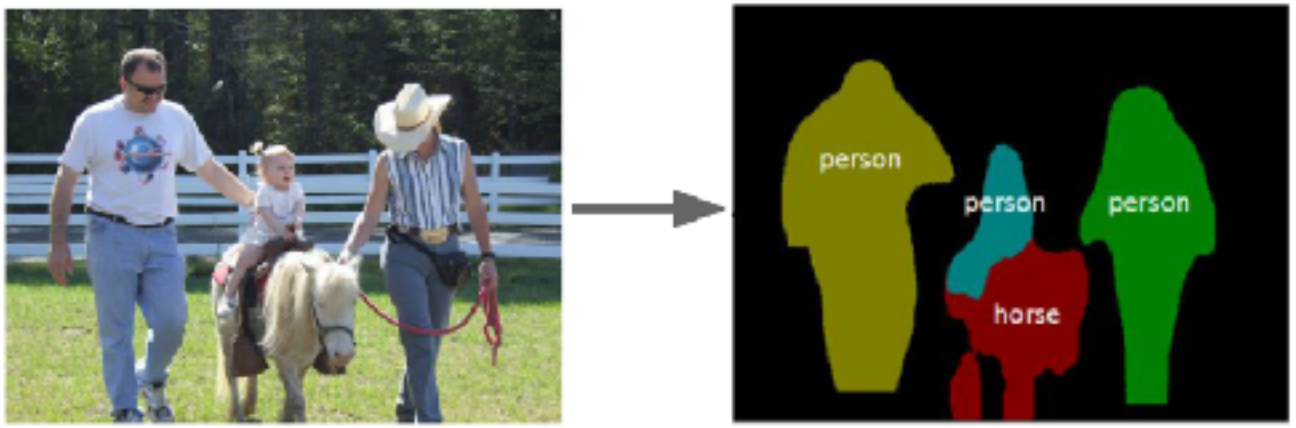
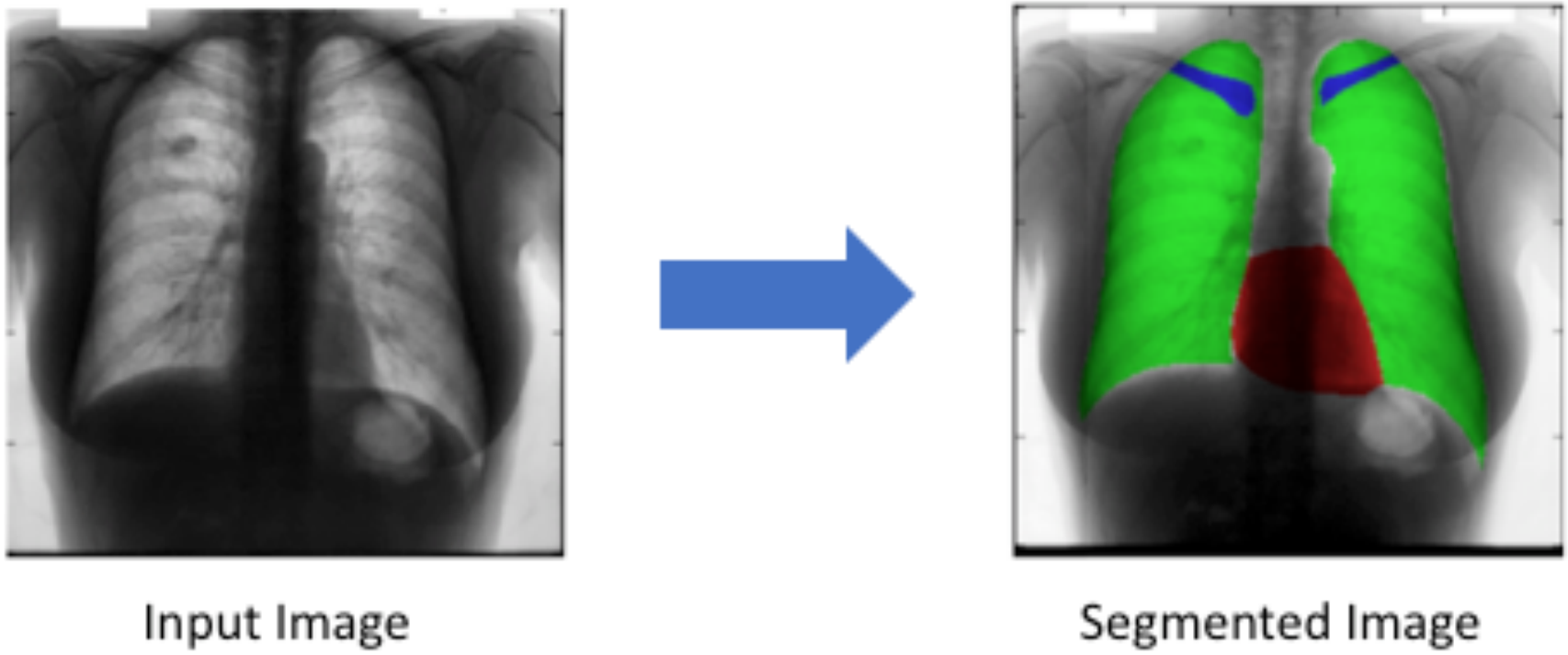


Figure credit: Dai et al, "Instance-aware Semantic Segmentation via Multi-task Network Cascades", arXiv 2015

Semantic Segmentation

A real-time segmented road scene for autonomous driving.
(https://www.youtube.com/watch?v=ATlcEDSPWXY)

Semantic Segmentation



A chest x-ray with the heart (red), lungs (green), and clavicles (blue) are segmented.

https://arxiv.org/abs/1701.08816

Goal of Semantic Segmentation

Input

segmented

1: Person
2: Purse
3: Plants/Grass
4: Sidewalk
5: Building/Structures

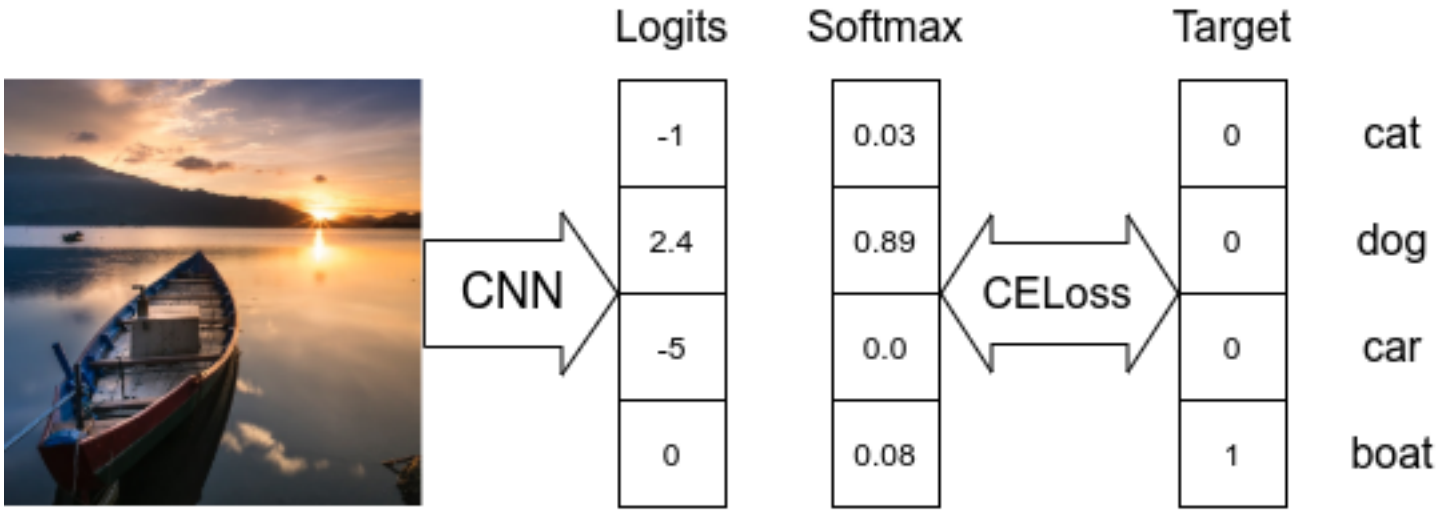
Semantic Labels

low-resolution prediction map only for clarity,
In reality label resolution = input image resolution

https://www.jeremyjordan.me/semantic-segmentation/

Recall Image Classification

- Target by one-hot encoding class labels



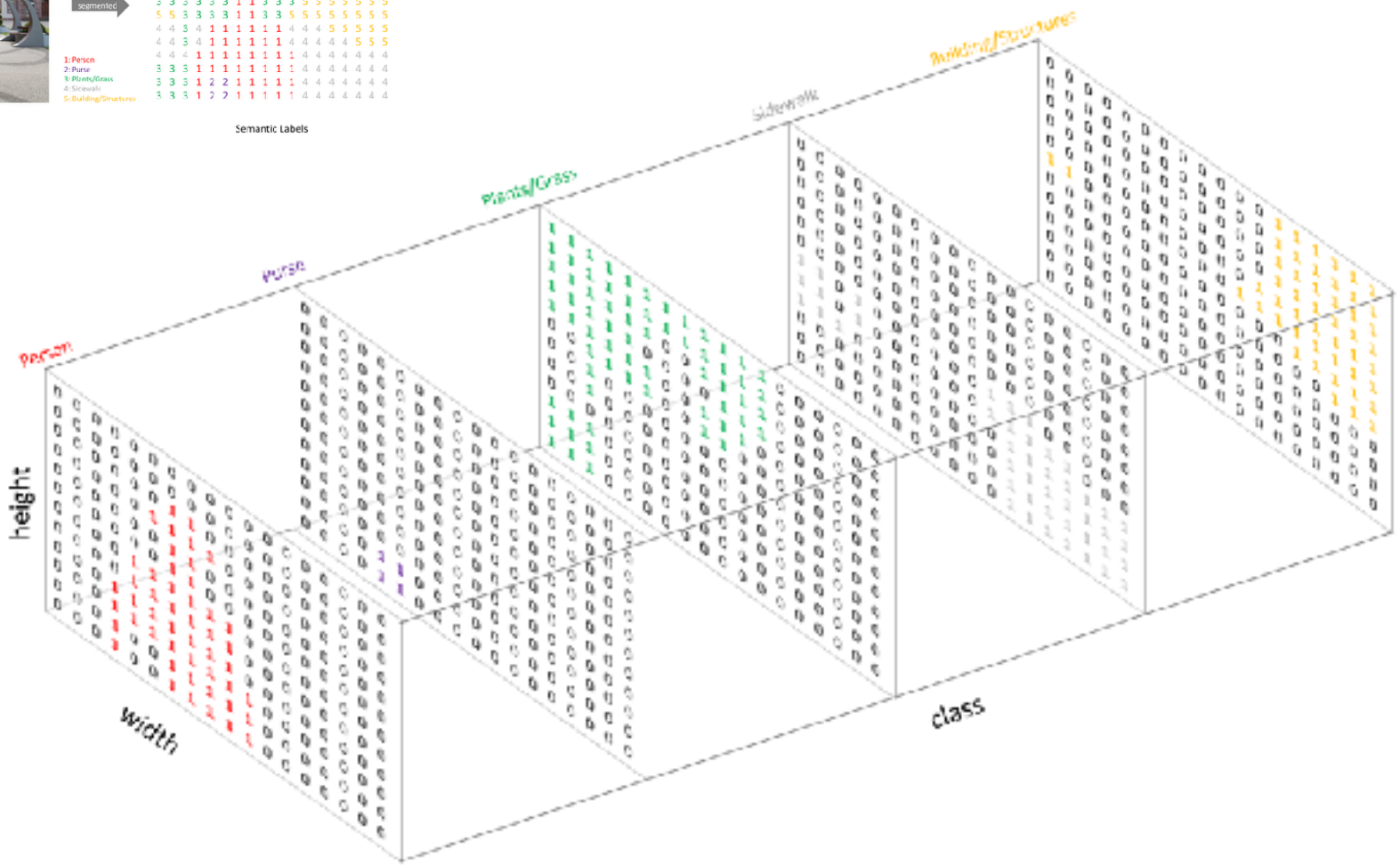
https://learnopencv.com/multi-label-image-classification-with-pytorch-image-tagging/

Targets in Semantic Segmentation

- Target by one-hot encoding class labels

Input

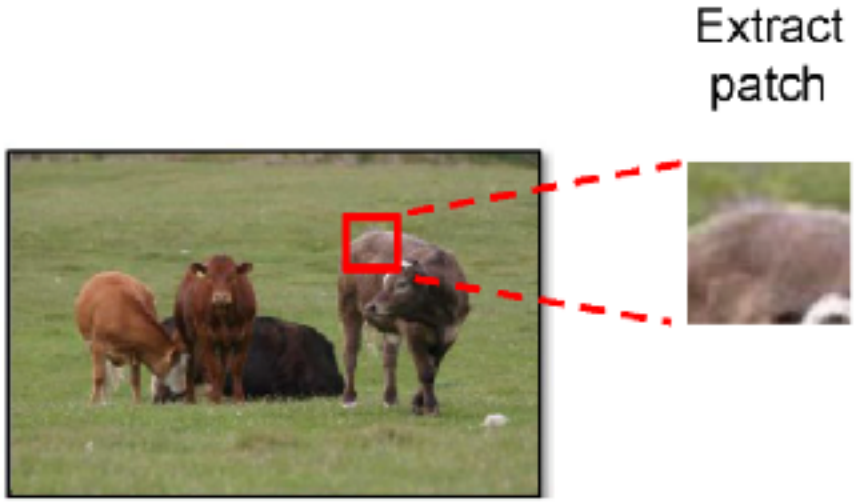
Semantic Labels



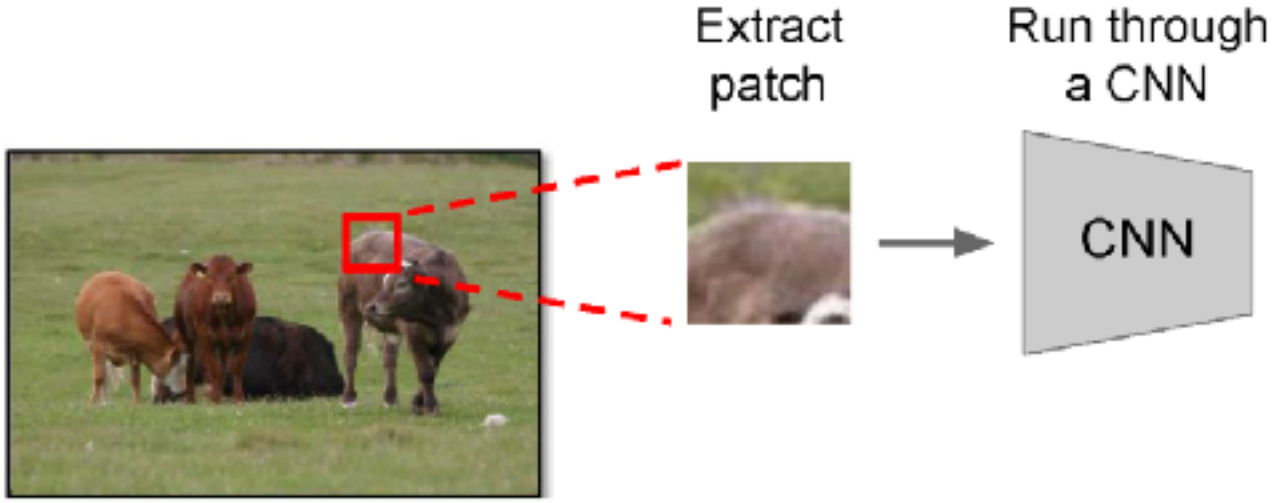
https://www.jeremyjordan.me/semantic-segmentation/

Semantic Segmentation via Classification

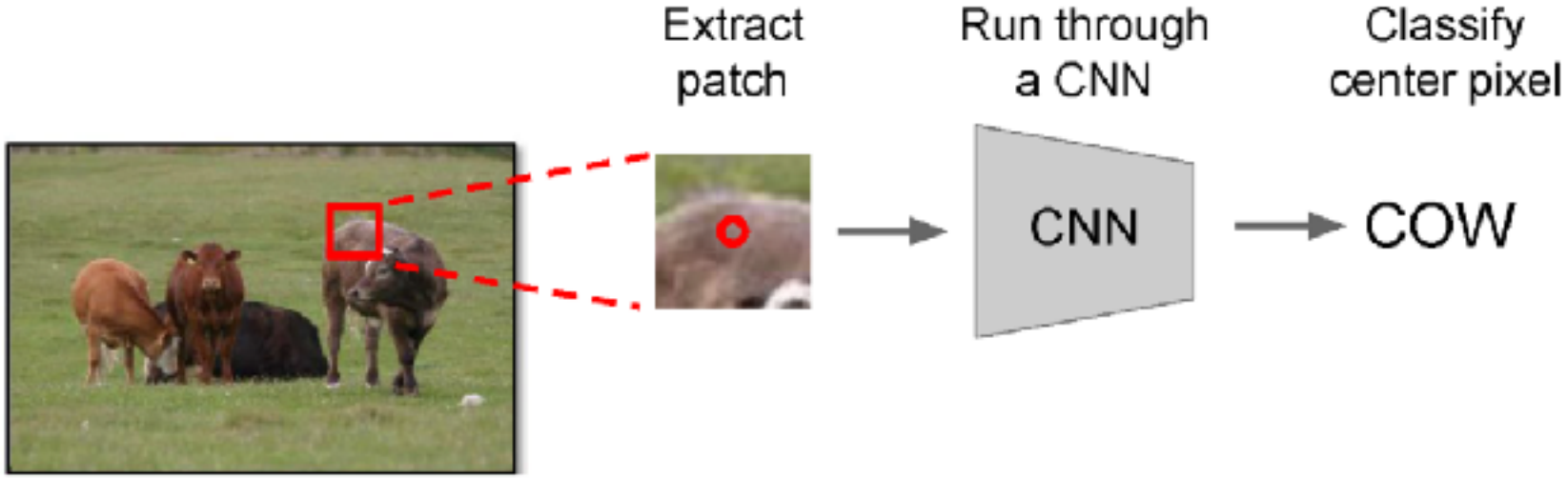
Semantic Segmentation via Classification



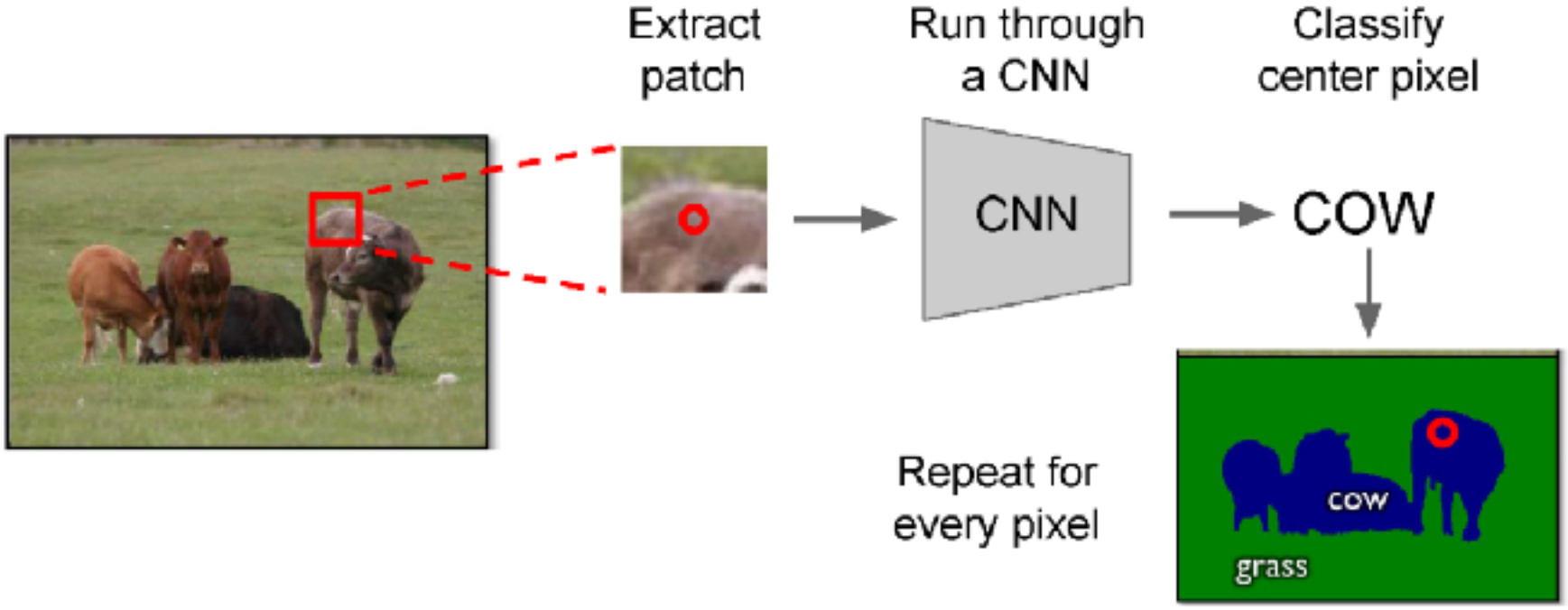
Semantic Segmentation via Classification



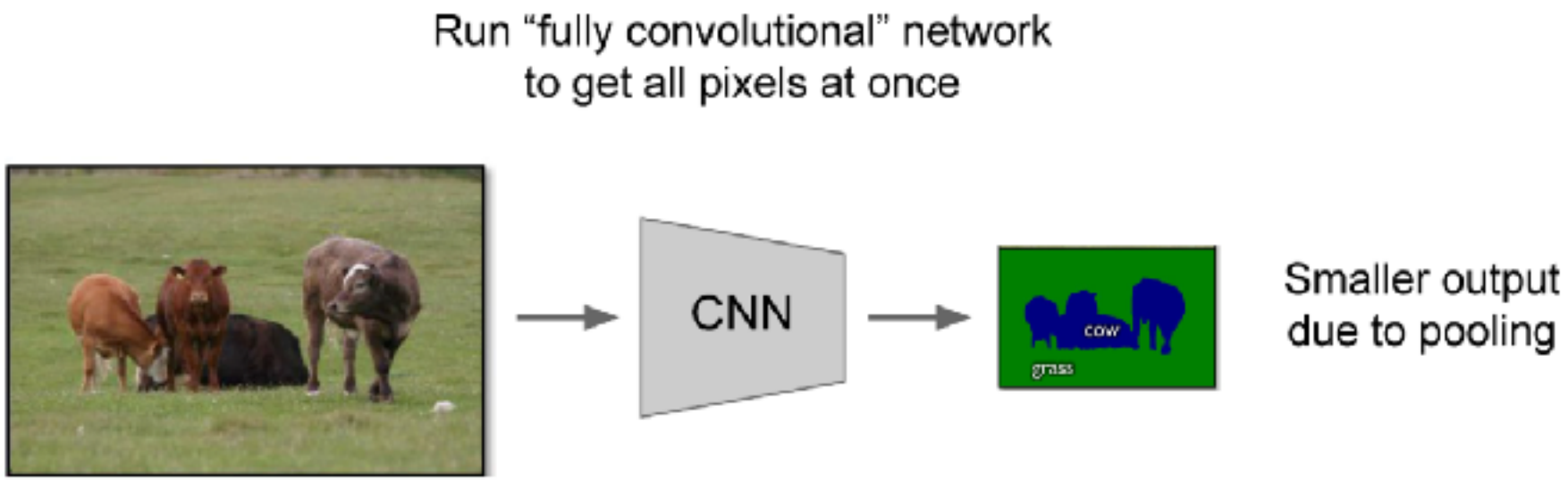
Semantic Segmentation via Classification



Semantic Segmentation via Classification

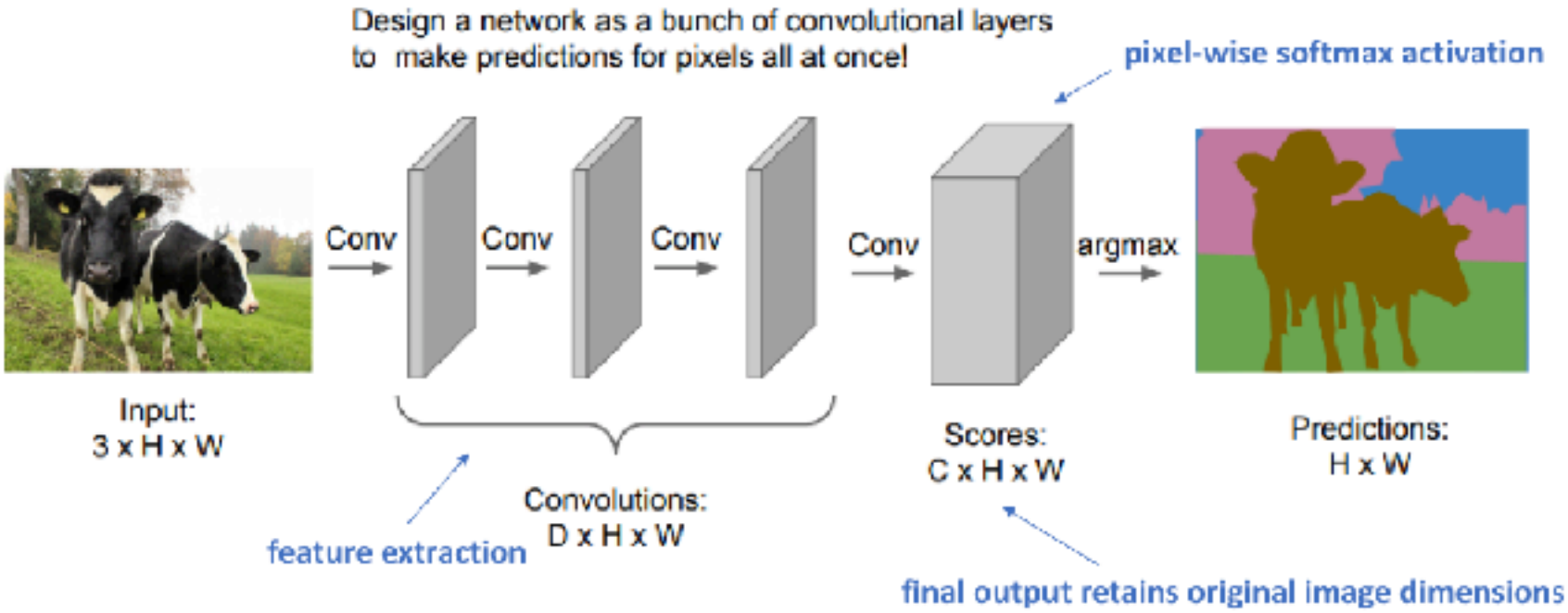


Semantic Segmentation via Classification



- computationally expensive and extremely slow to perform forward pass independently for each pixel

Semantic Segmentation: Naive Approach



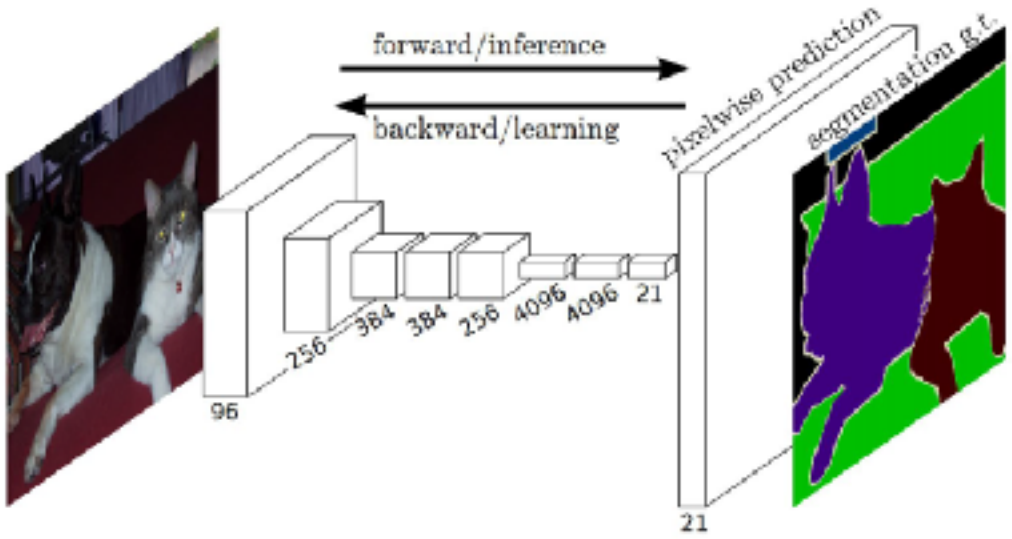
Downside: Preserving image dimensions throughout entire network will be computationally expensive.

- Computationally expensive and memory intensive to preserve the full resolution throughout the network.

https://www.jeremyjordan.me/semantic-segmentation/

Semantic Segmentation: Upsampling

- In order to maintain expressiveness, we typically need to increase the number of feature maps (channels) as we get deeper in the network.

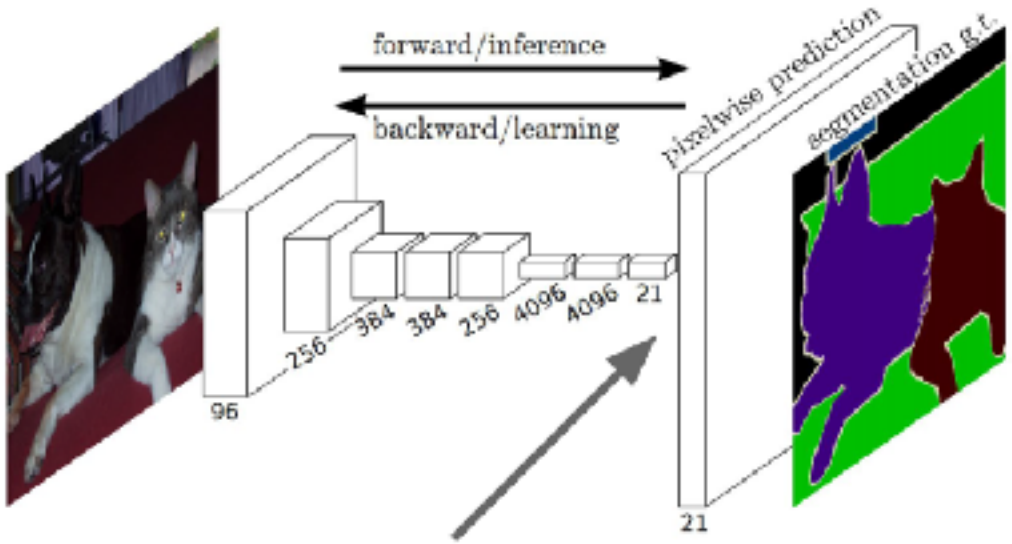


Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015

https://arxiv.org/pdf/1411.4038.pdf

Semantic Segmentation

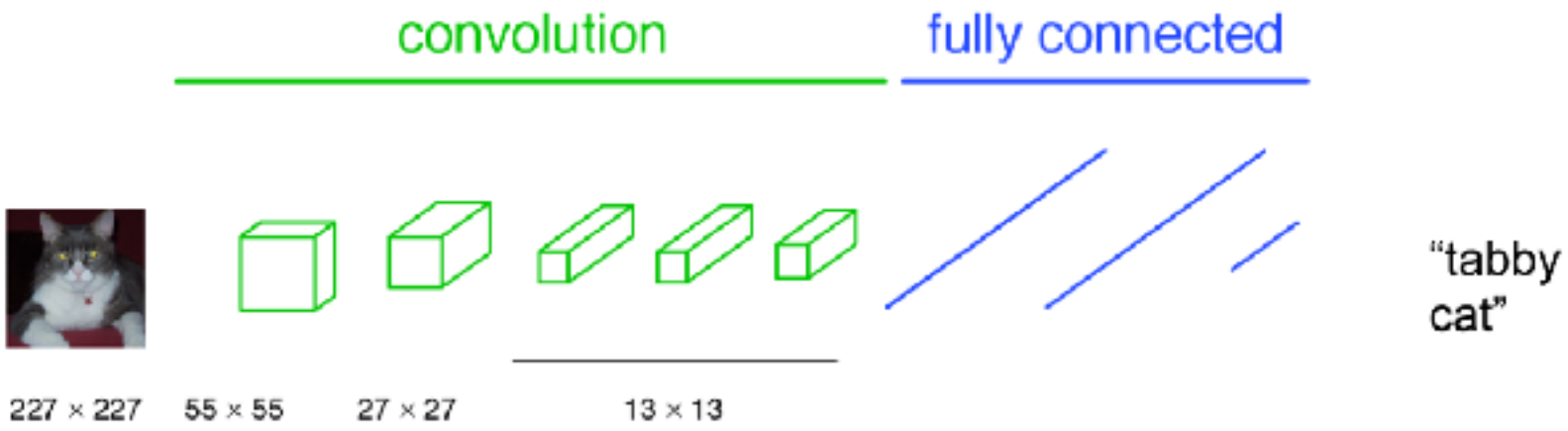
- In order to maintain expressiveness, we typically need to increase the number of feature maps (channels) as we get deeper in the network.



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

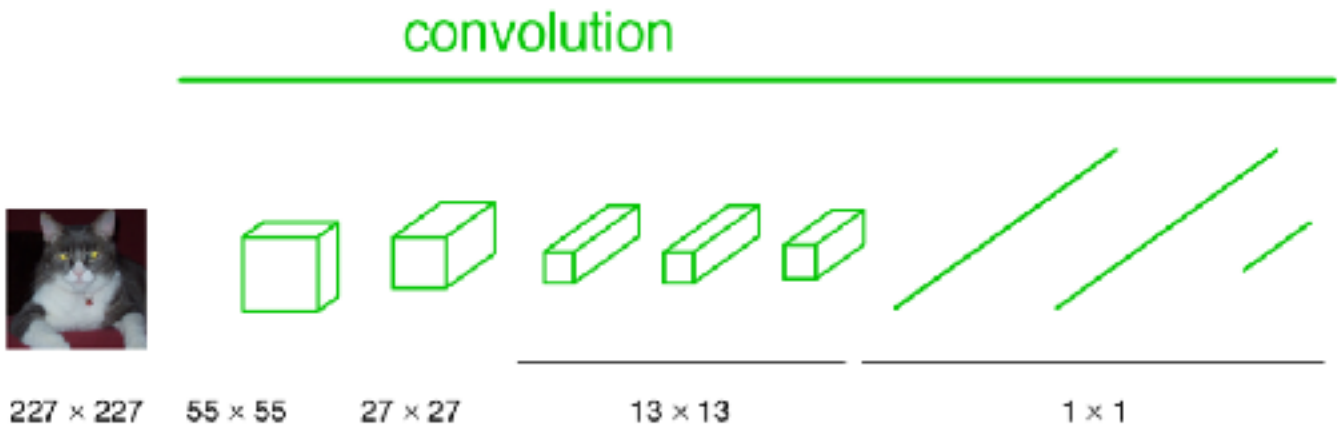
https://arxiv.org/pdf/1411.4038.pdf

a classification network



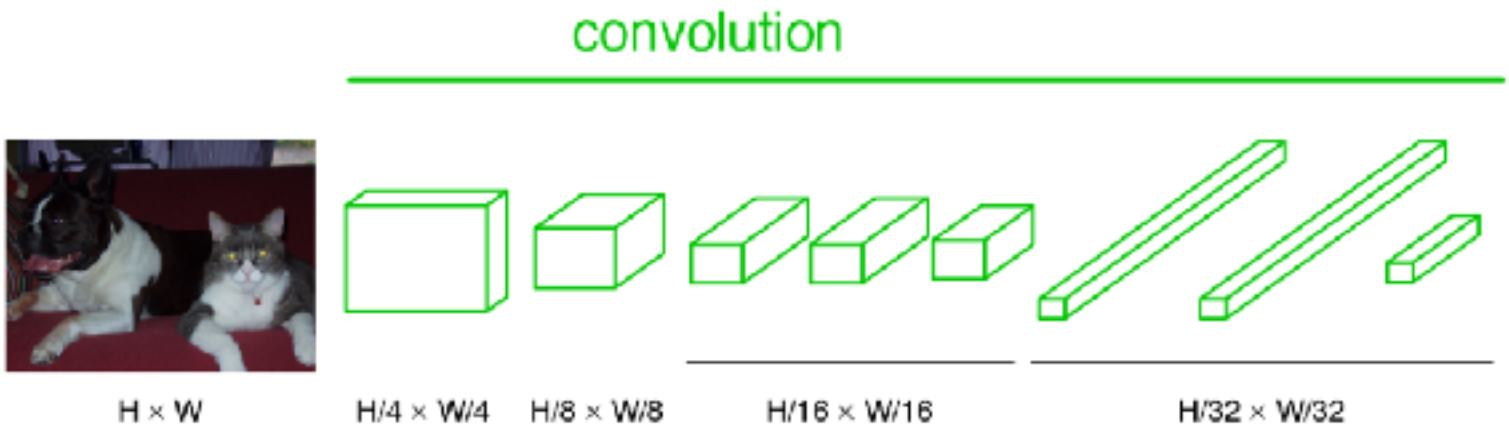
Slide credit: Jonathan Long

becoming fully convolutional



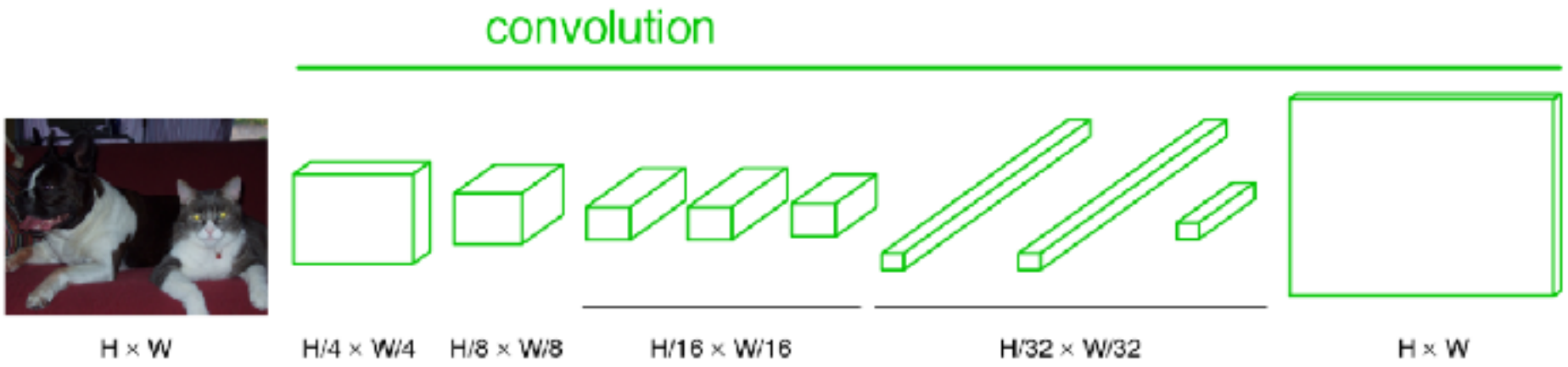
Slide credit: Jonathan Long

becoming fully convolutional



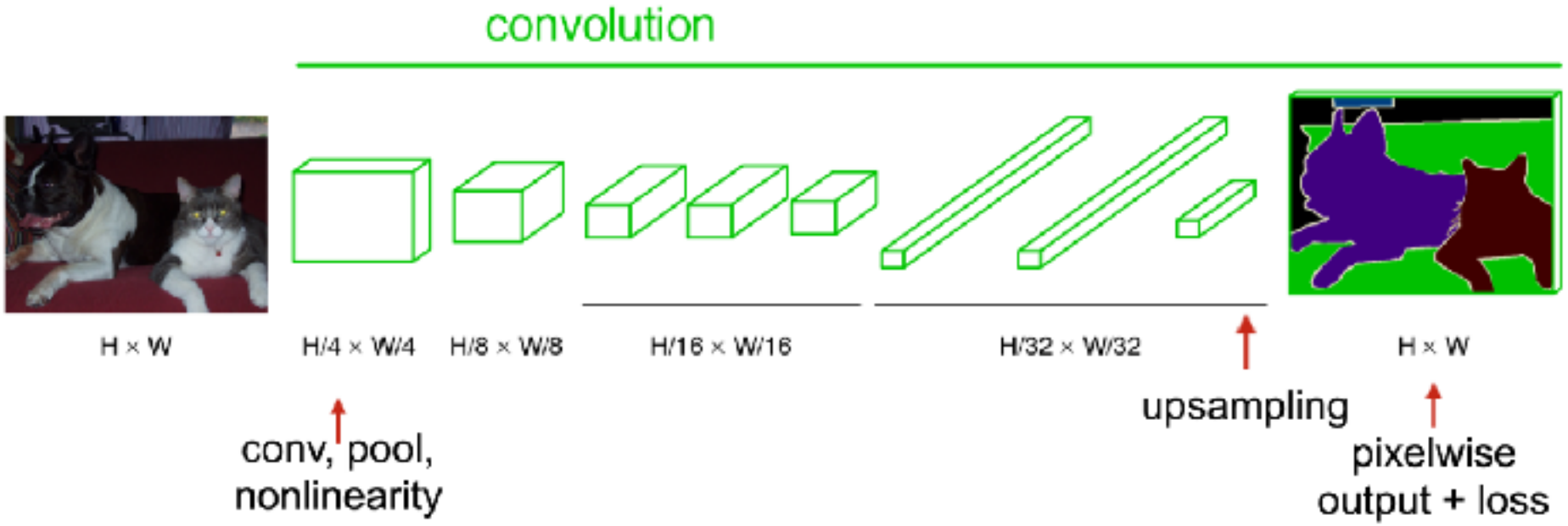
Slide credit: Jonathan Long

upsampling output



Slide credit: Jonathan Long

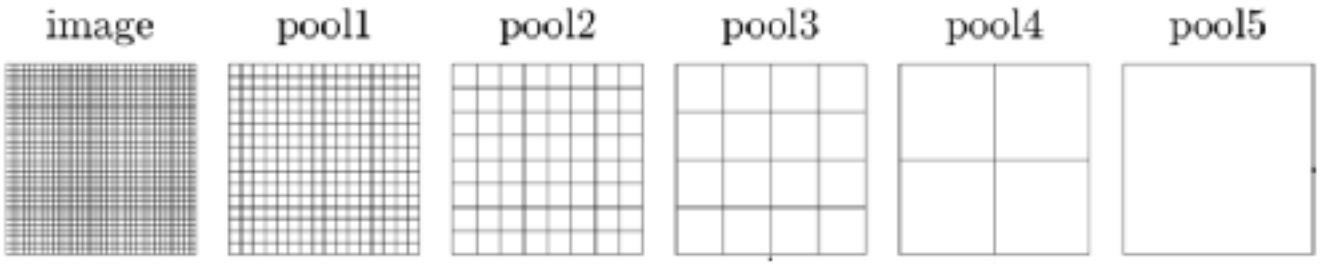
end-to-end, pixels-to-pixels network



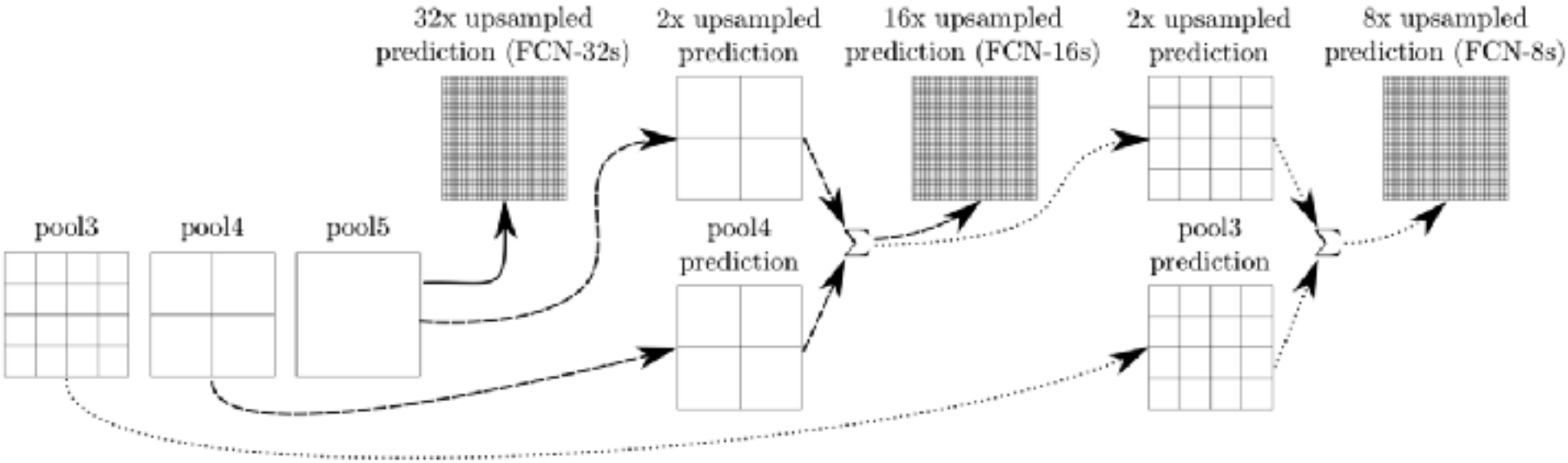
Slide credit: Jonathan Long

skip layers

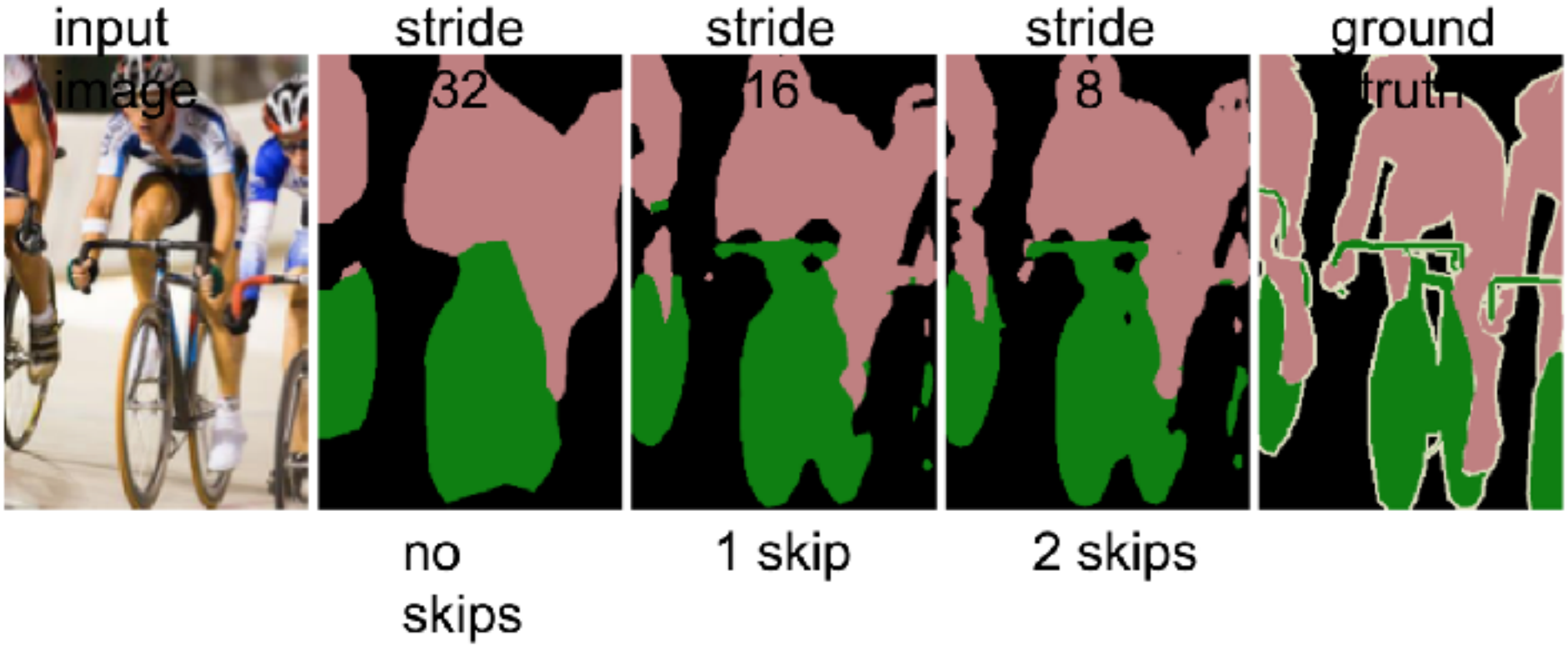
- Encoder/Downsampling



- Learnable Upsampling

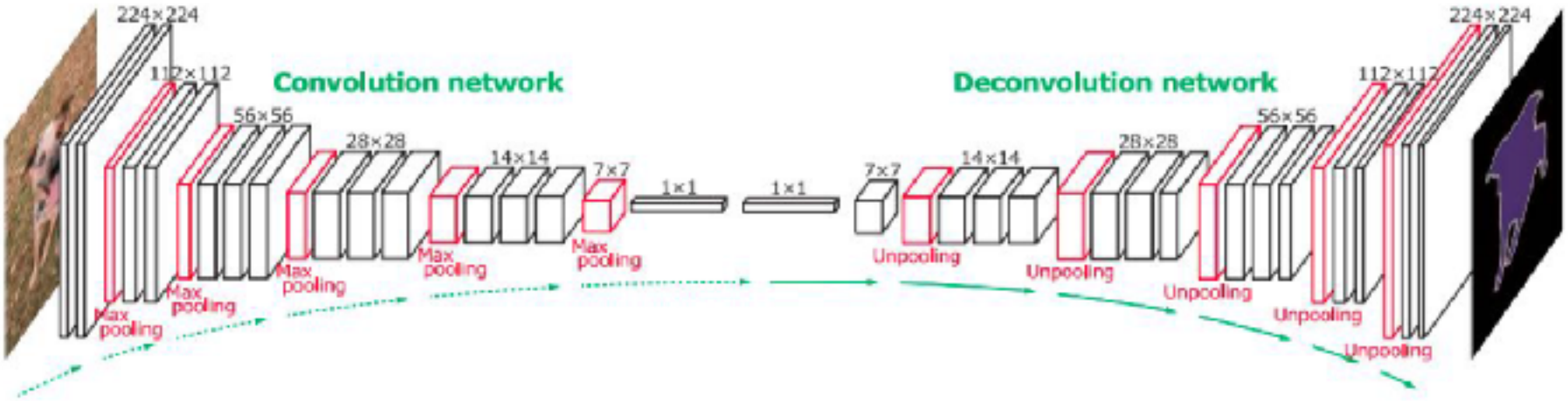


skip layer refinement



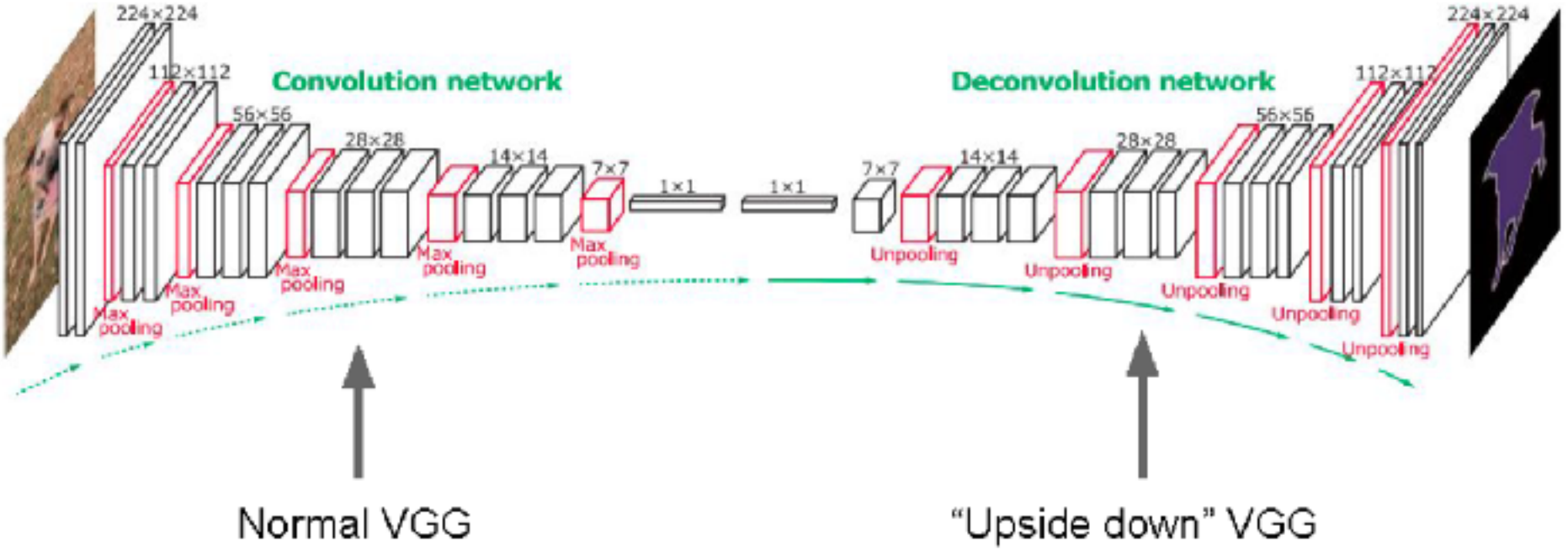
- Slide credit: Jonathan Long

Semantic Segmentation: Encoder/Decoder



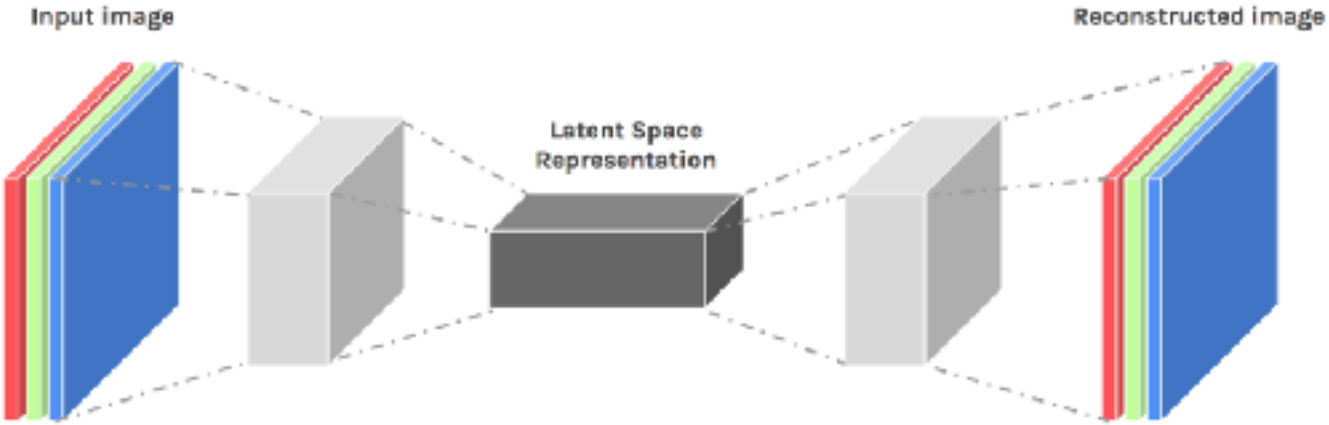
] Deconvolution Network for Semantic Segmentation", ICCV 2015

Semantic Segmentation: Encoder/Decoder



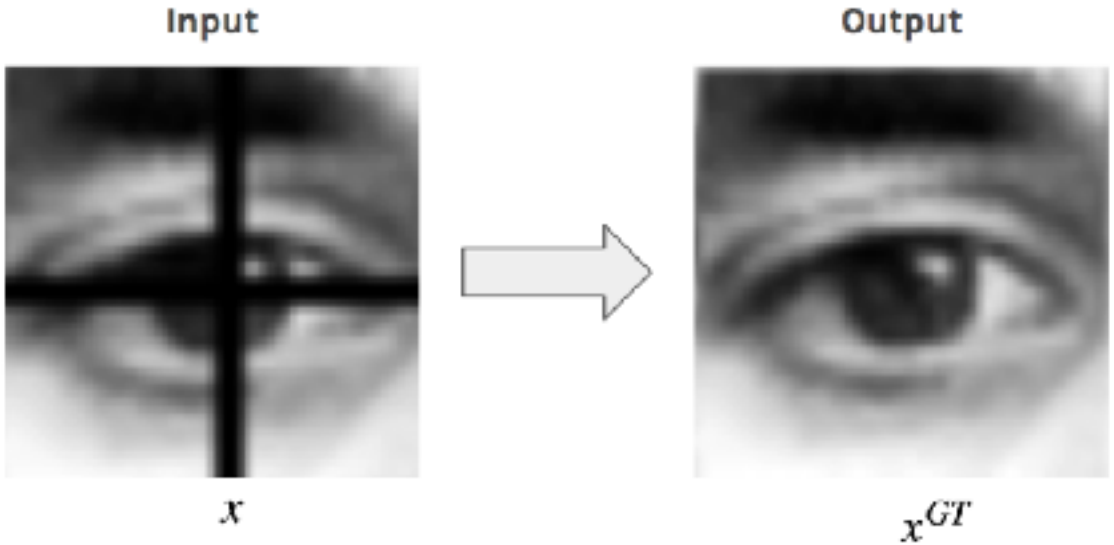
Recap: Convolutional Autoencoders

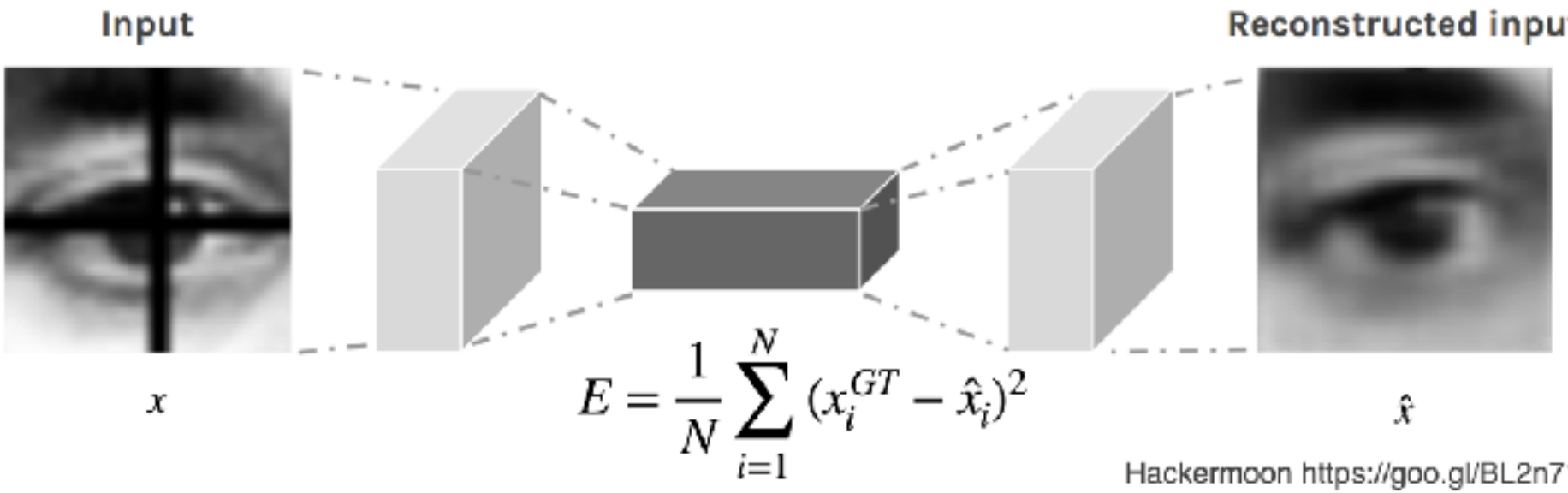
- Replace *fully connected layers by convolutional layers*. This helps the network extract **visual features** from the images, and therefore obtain a much more accurate latent space representation. The reconstruction process uses ***upsampling*** and convolutions.



Hackermoon https://goo.gl/BL2n71

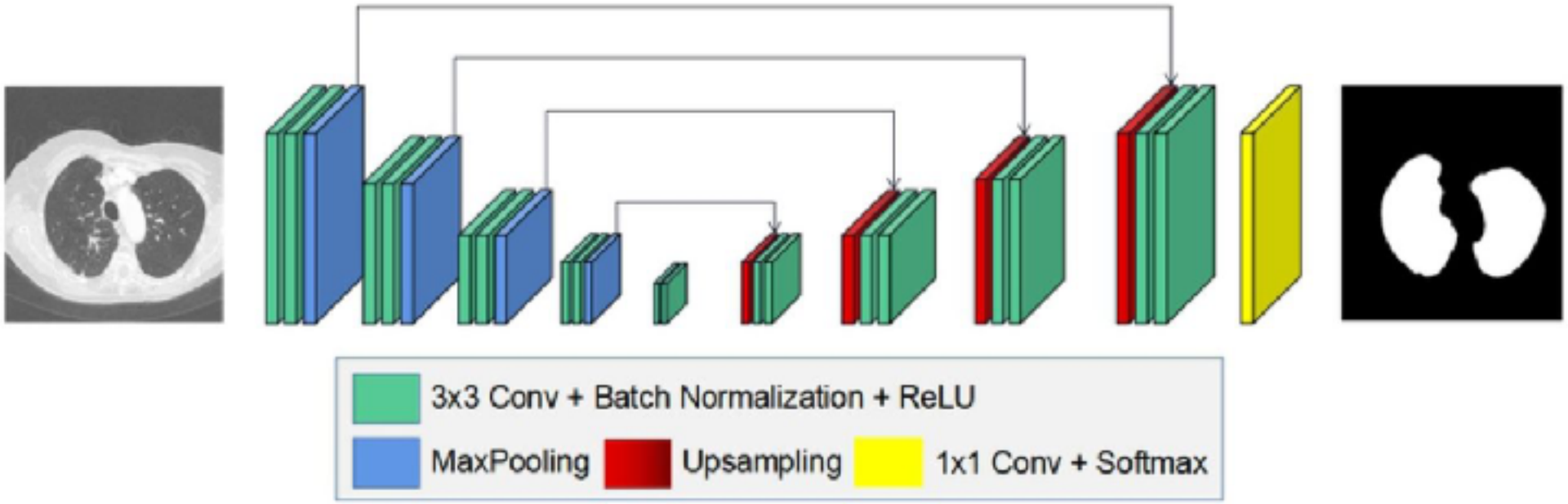
Example 1: Ultra-basic image reconstruction

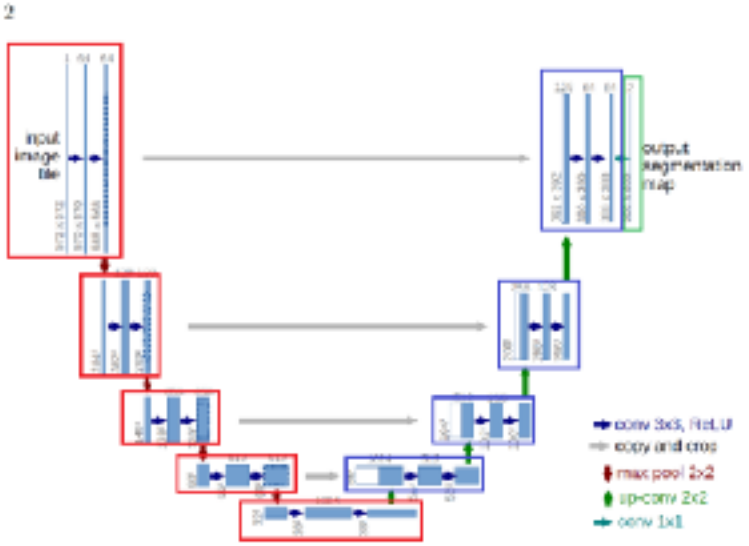




Hackermoon https://goo.gl/BL2n71

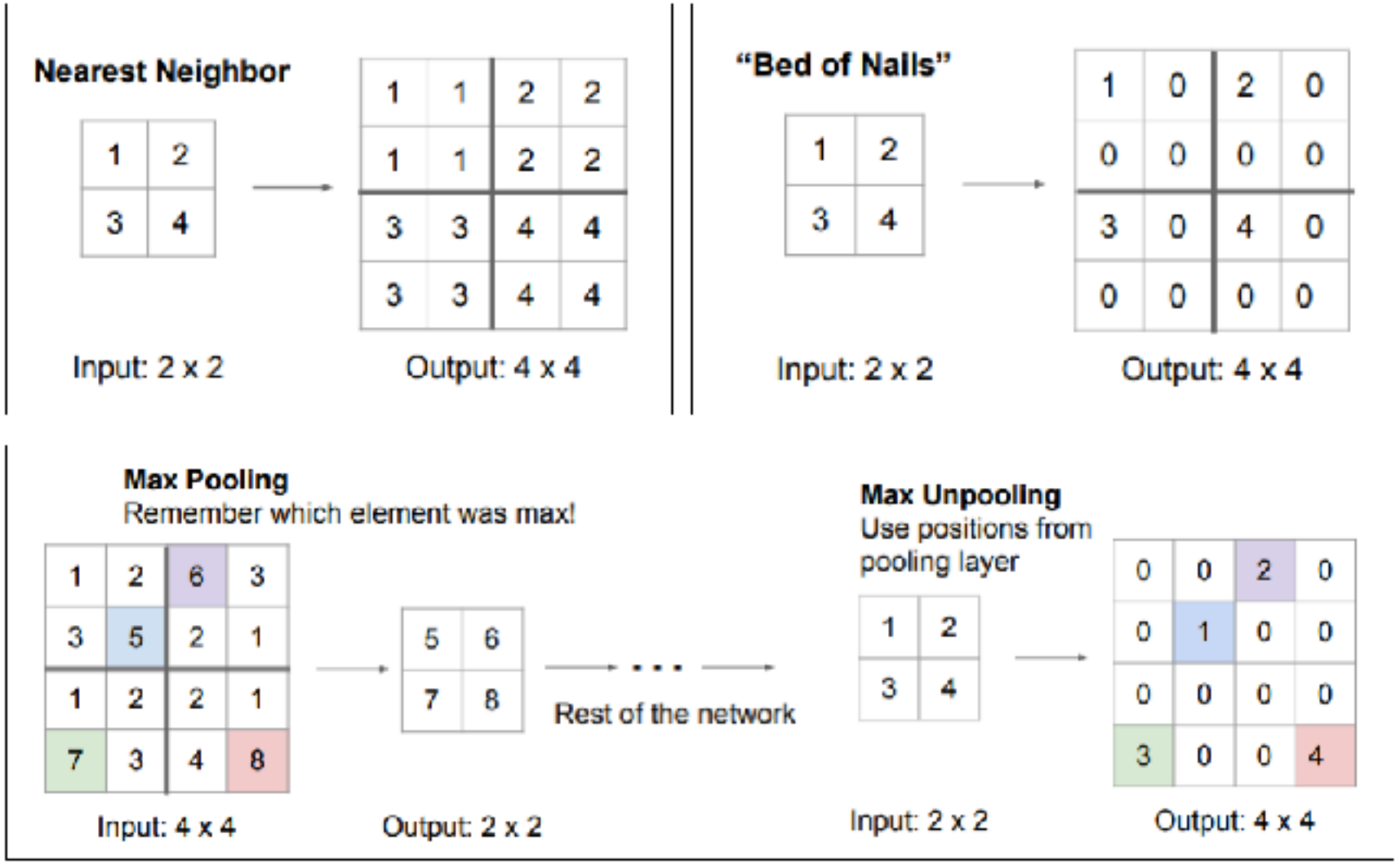
U-Net:
Convolutional Networks for Biomedical Image Segmentation





Methods for upsampling

Methods for upsampling

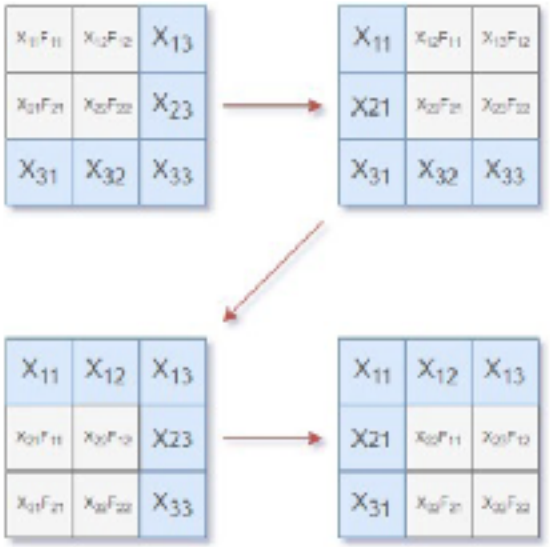


https://www.jeremyjordan.me/semantic-segmentation/

Transposed Convolution

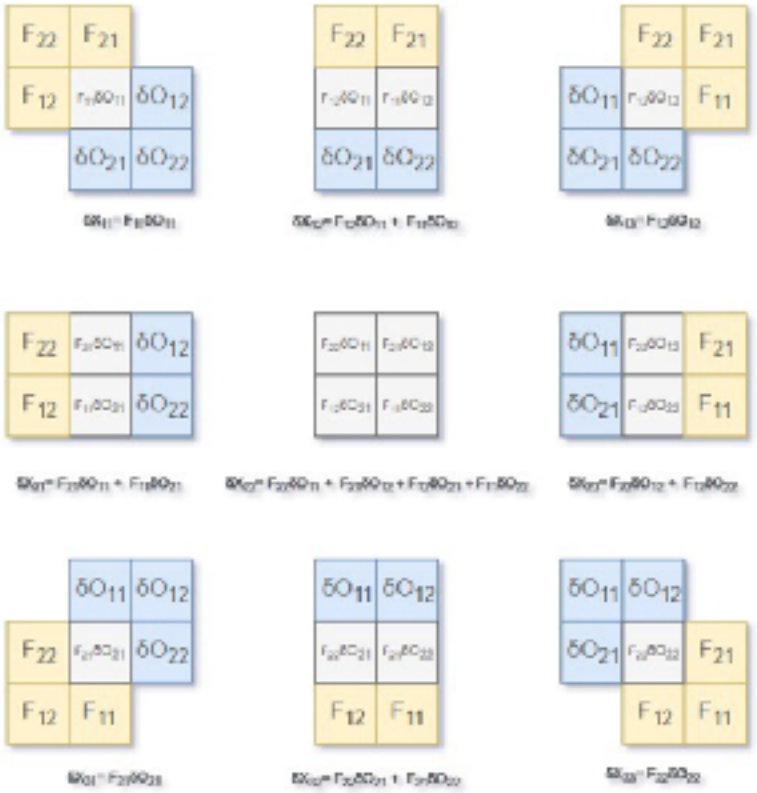
- Convolution

$$\begin{bmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{bmatrix} = \text{Convolution}\left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}, \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \right)$$



- Full Convolution

$$\begin{bmatrix} \partial E/\partial X_{11} & \partial E/\partial X_{12} & \partial E/\partial X_{13} \\ \partial E/\partial X_{21} & \partial E/\partial X_{22} & \partial E/\partial X_{23} \\ \partial E/\partial X_{31} & \partial E/\partial X_{32} & \partial E/\partial X_{33} \end{bmatrix} = \text{Full\_Convolution}\left(\begin{bmatrix} \partial E/\partial O_{11} & \partial E/\partial O_{12} \\ \partial E/\partial O_{21} & \partial E/\partial O_{22} \end{bmatrix}, \begin{bmatrix} F_{22} & F_{21} \\ F_{12} & F_{11} \end{bmatrix} \right)$$



Summary

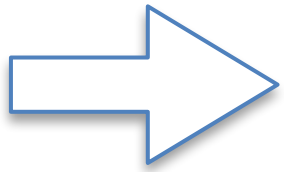
- The transposed convolution operation forms the **same connectivity** as the normal convolution but in the **backward direction**.

- We can use it to conduct **up-sampling**. Moreover, the **weights in the transposed convolution are learnable**. So we do not need a predefined interpolation method.

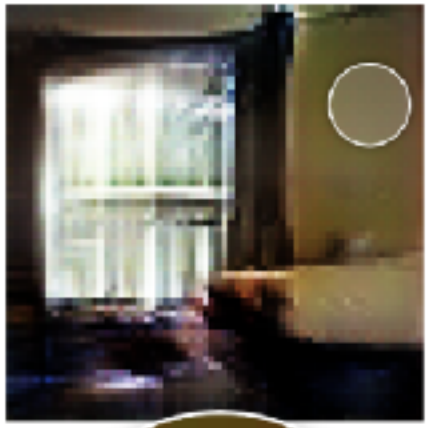
- Even though it is called the transposed convolution, it does not mean that we take some existing convolution matrix and use the transposed version.

Transposed Convolution and Checkerboard Artifacts

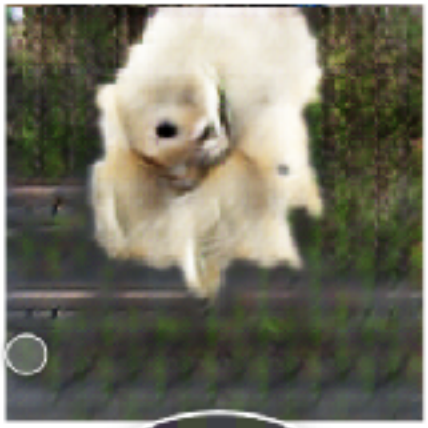
Supplementary Slides



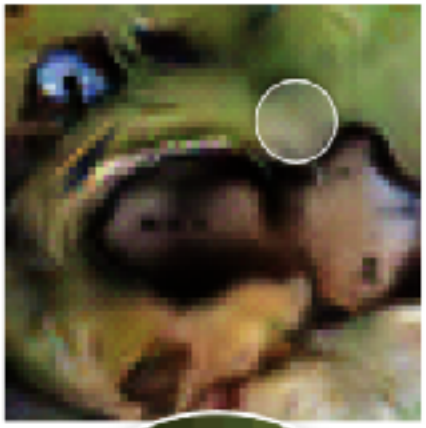
Checkerboard Artifacts



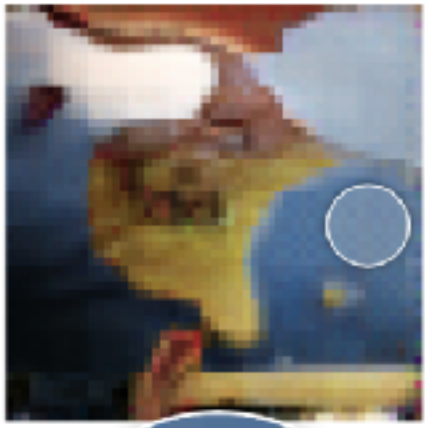
Radford, et al., 2015 [1]



Salimans et al., 2016 [2]



Donahue, et al., 2016 [3]

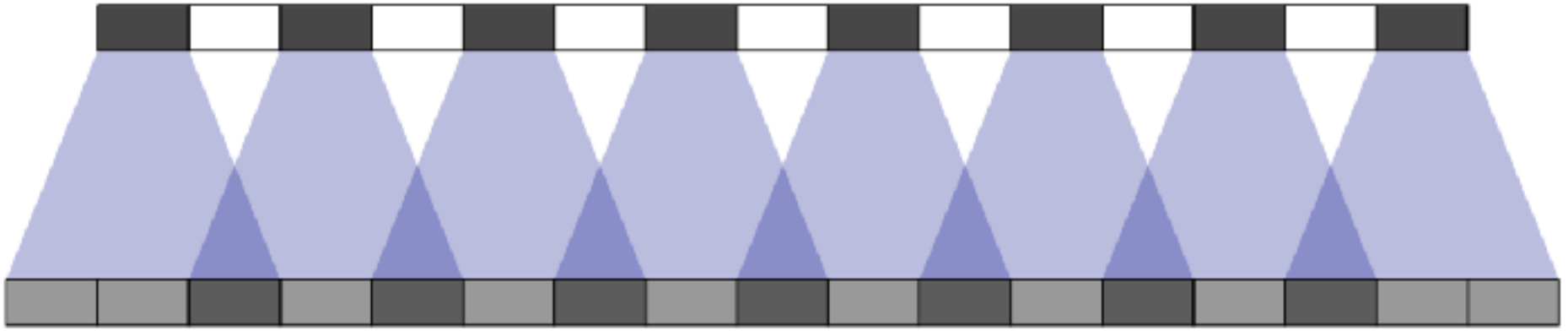


Dumoulin, et al., 2016 [4]

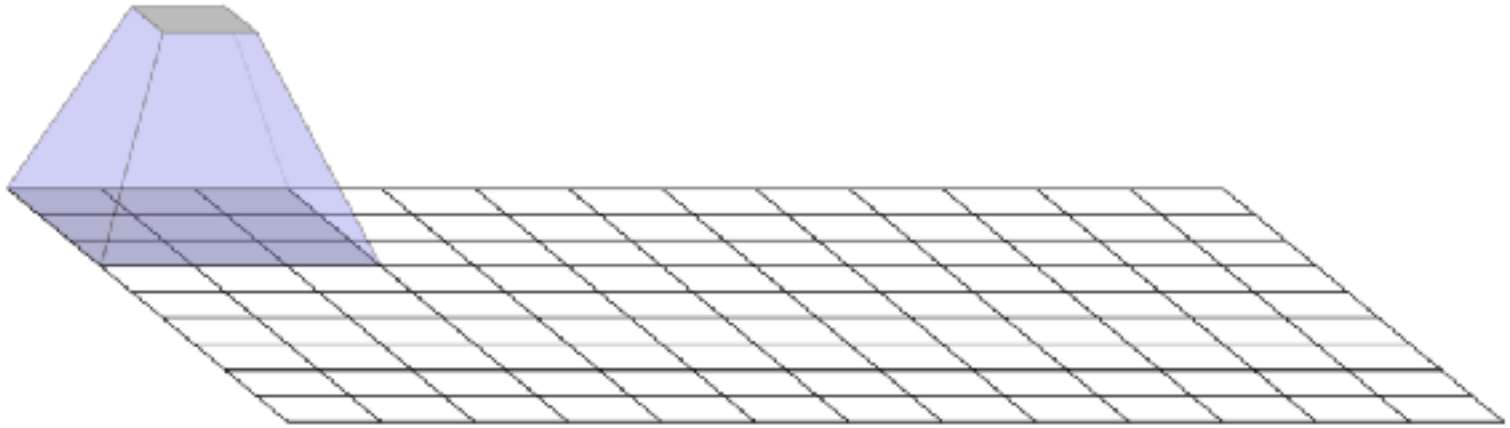
https://distill.pub/2016/deconv-checkerboard/

Checkerboard Artifacts - 1D Example

- Stride 2
- Size 3

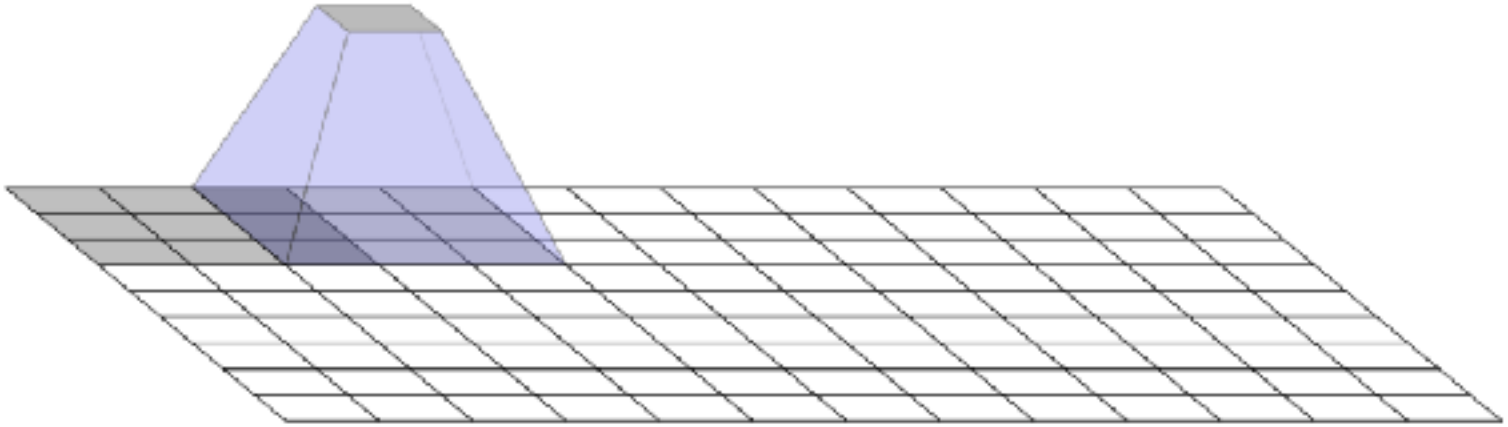


Transposed Convolution & Overlap



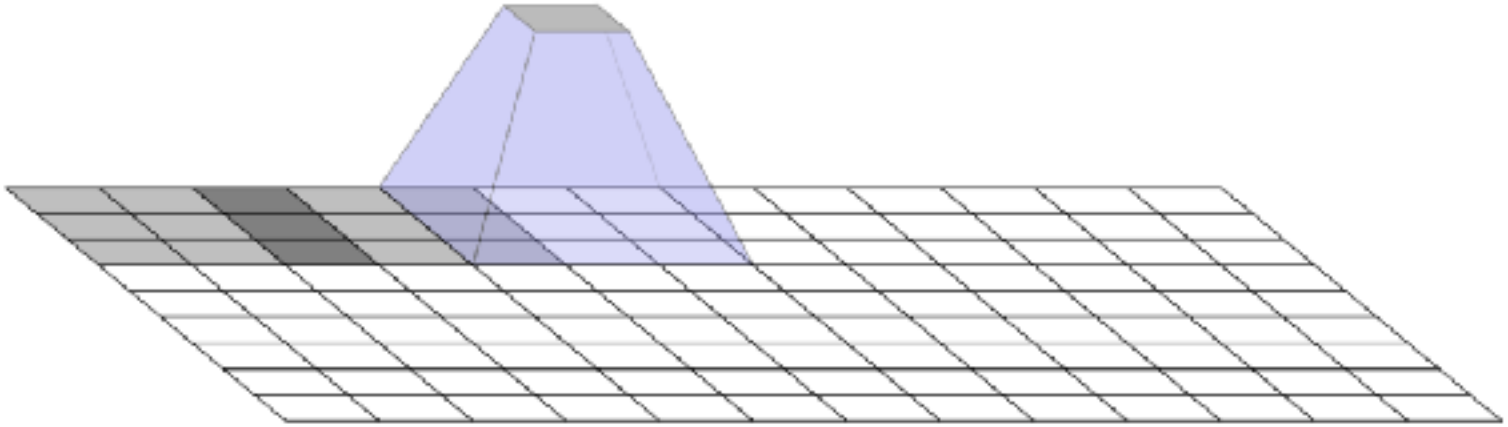
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



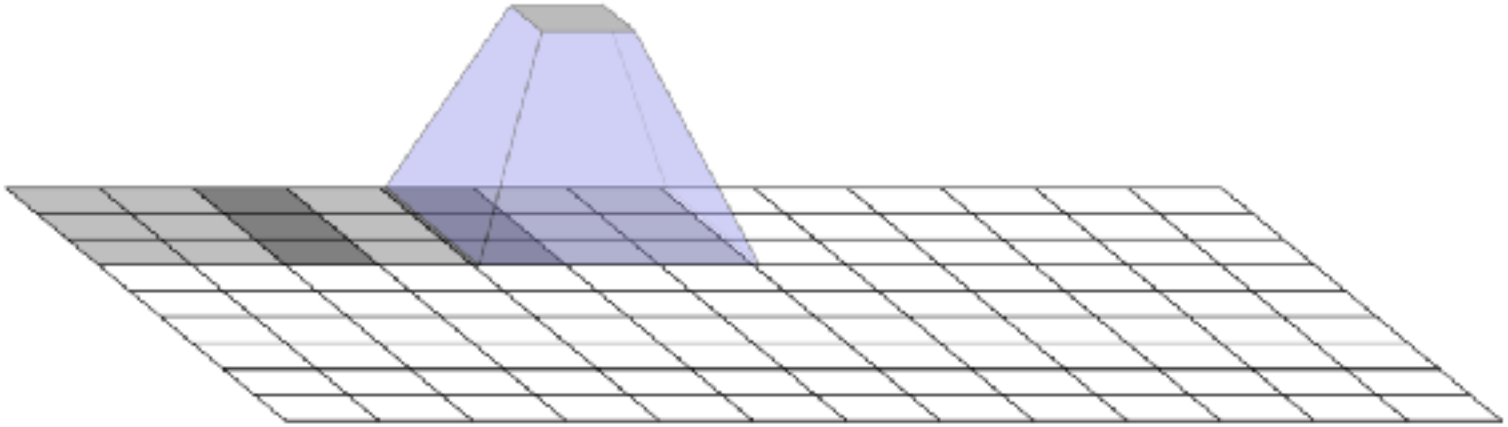
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



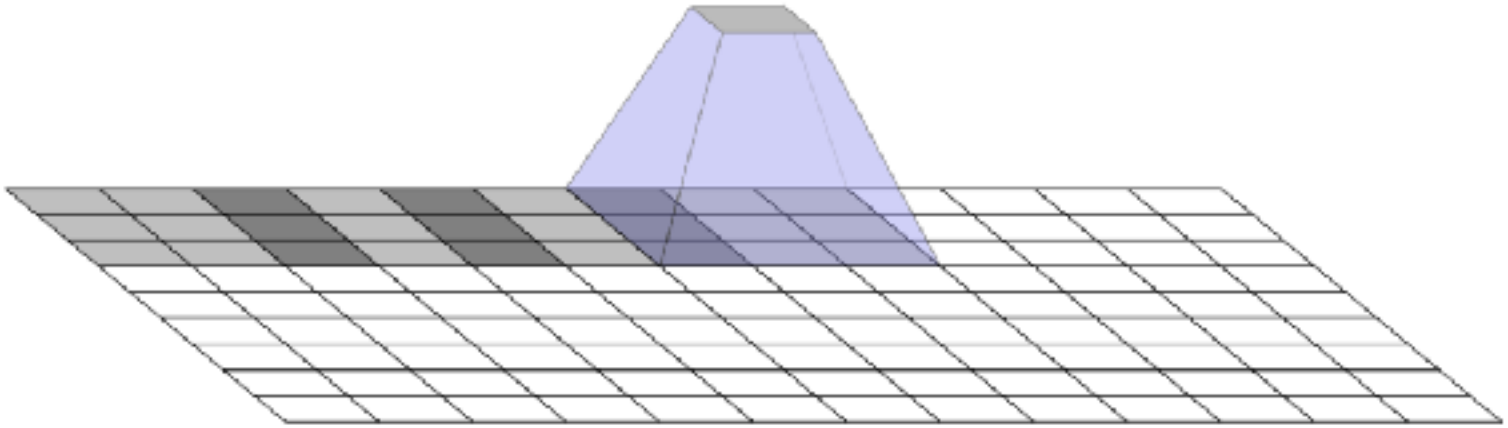
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



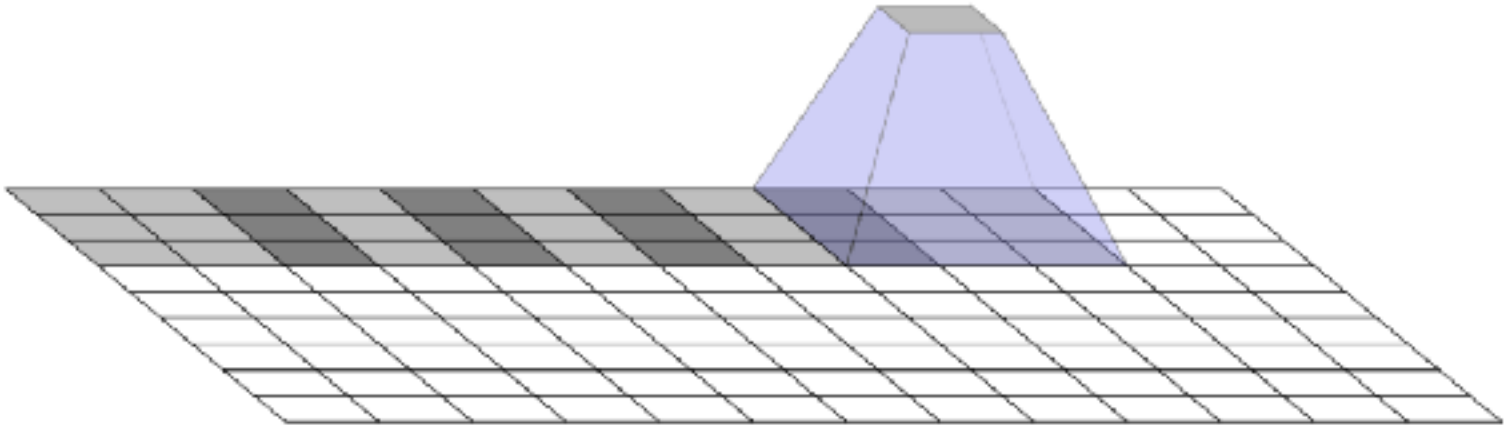
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



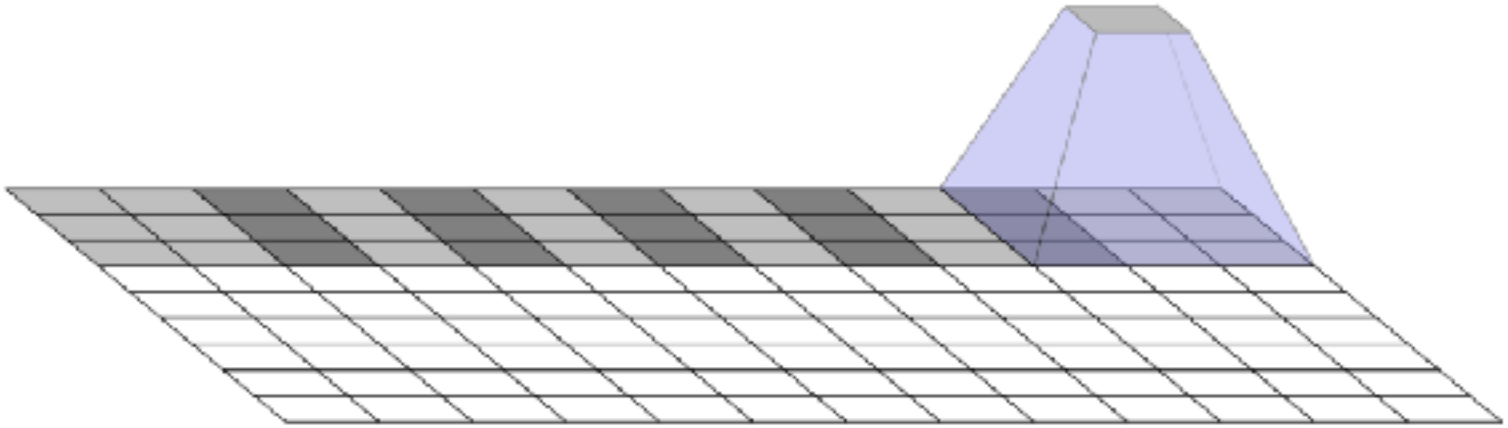
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



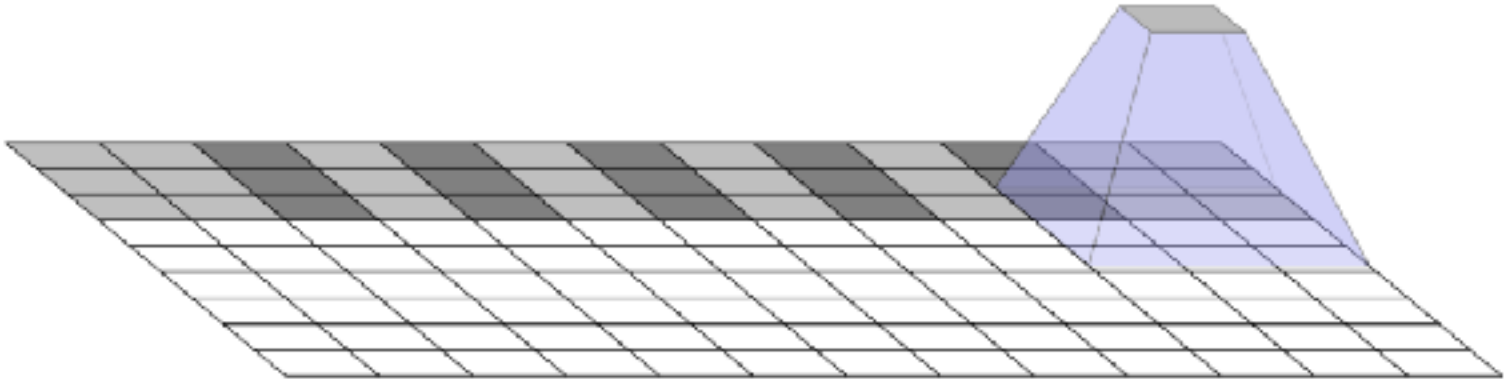
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



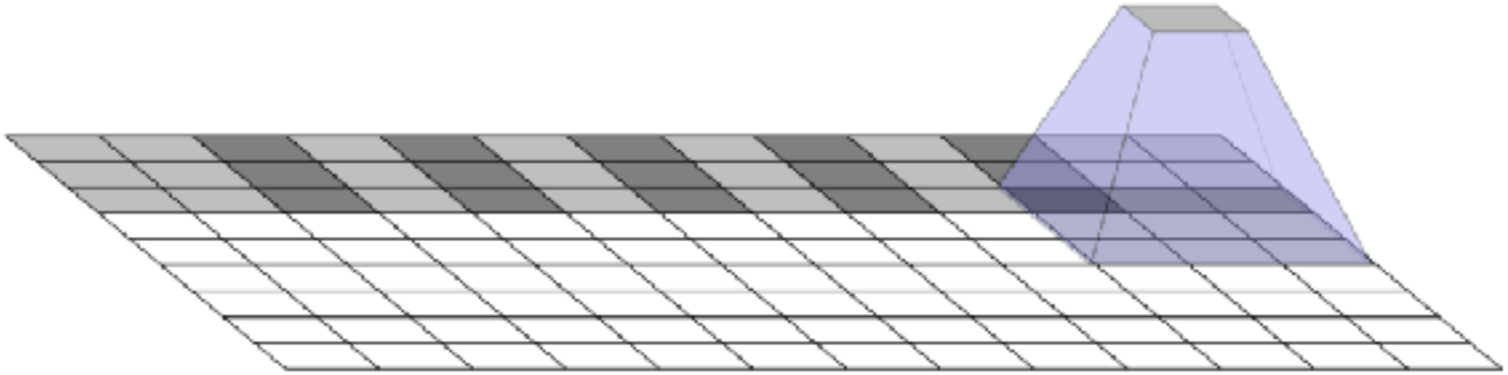
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



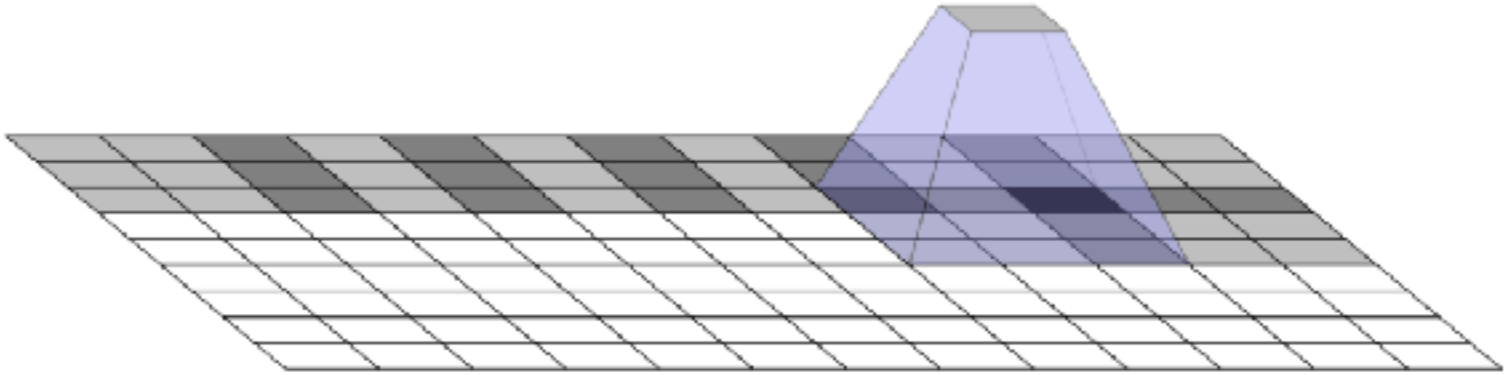
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



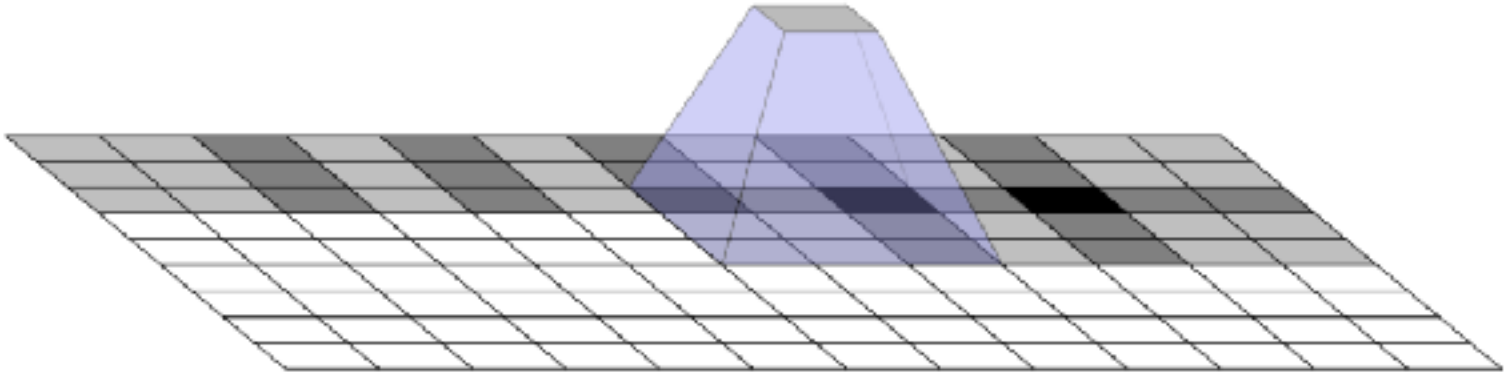
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



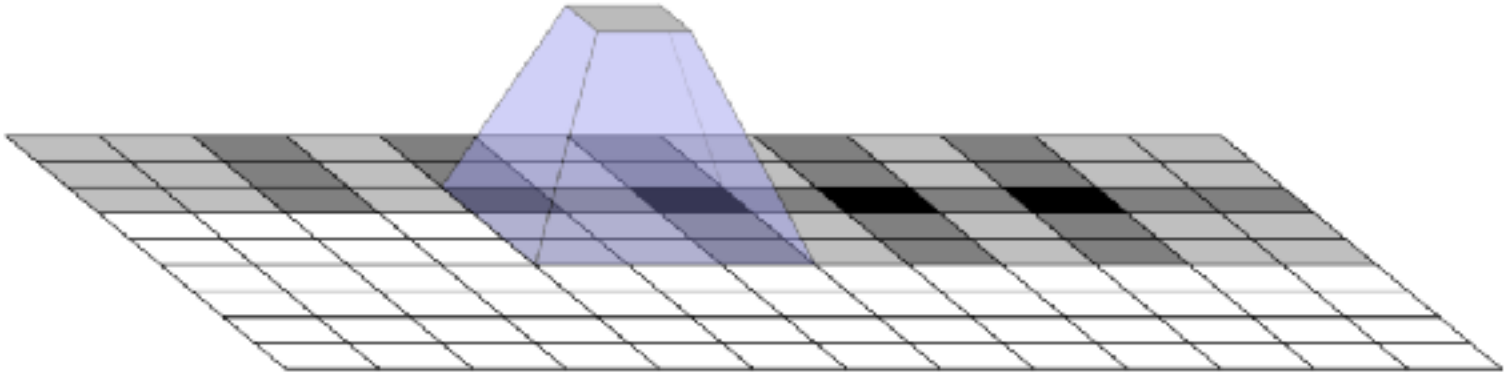
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



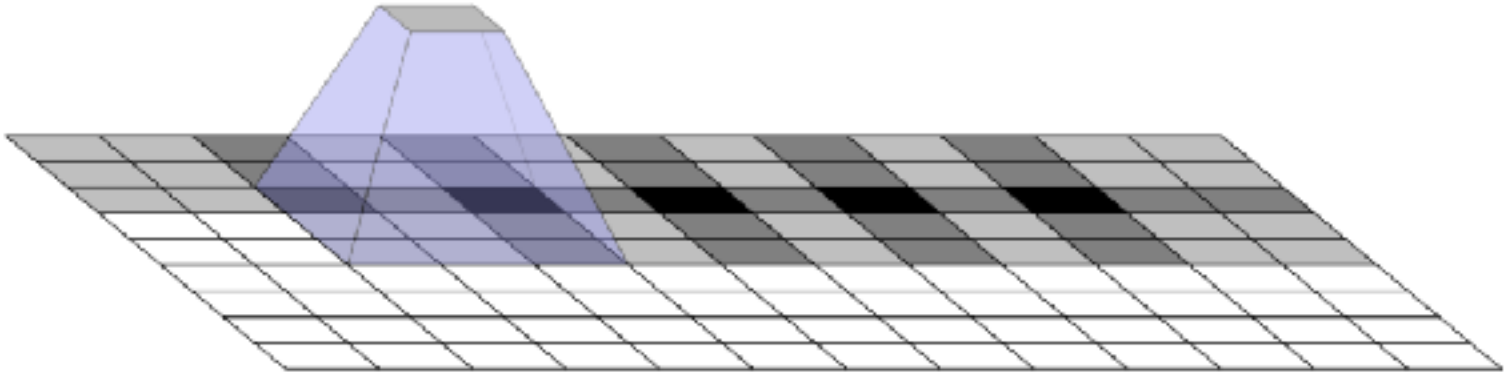
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



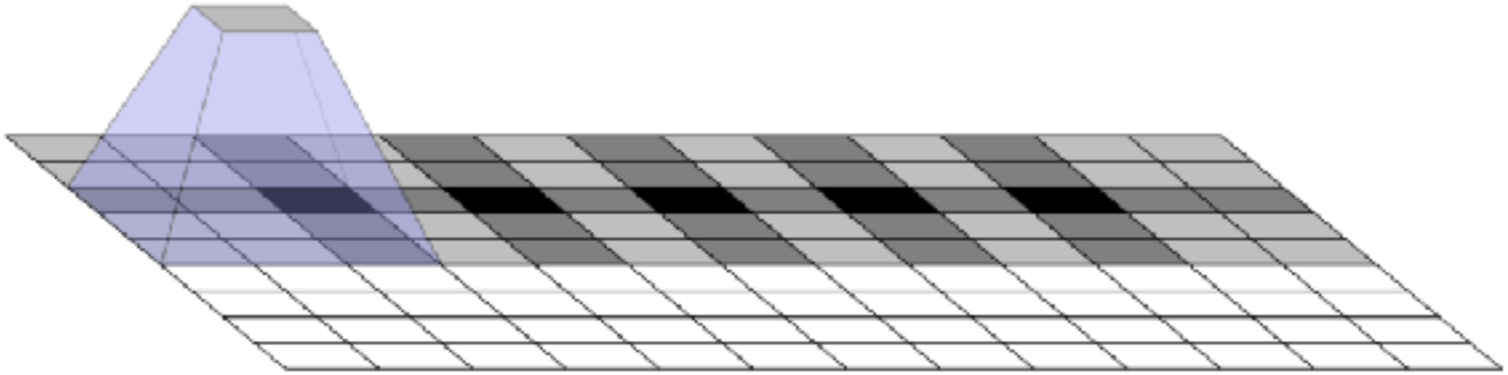
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



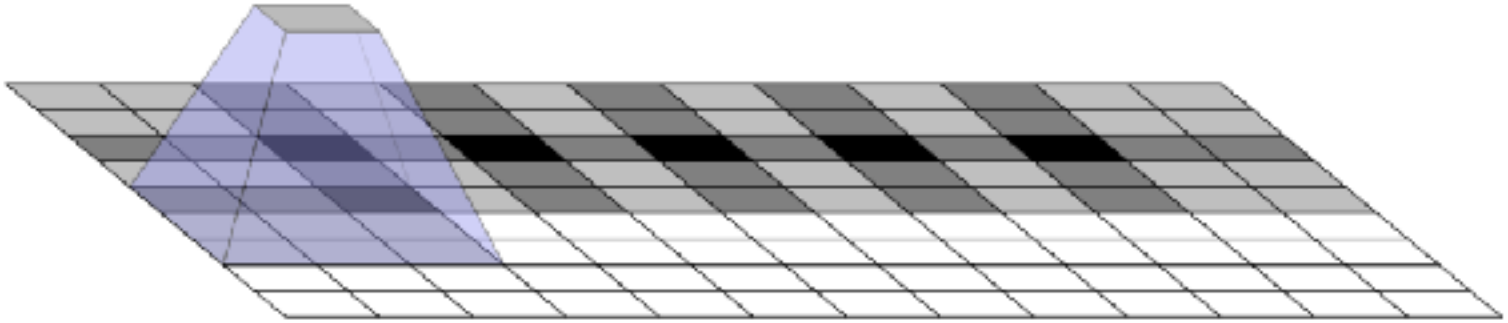
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



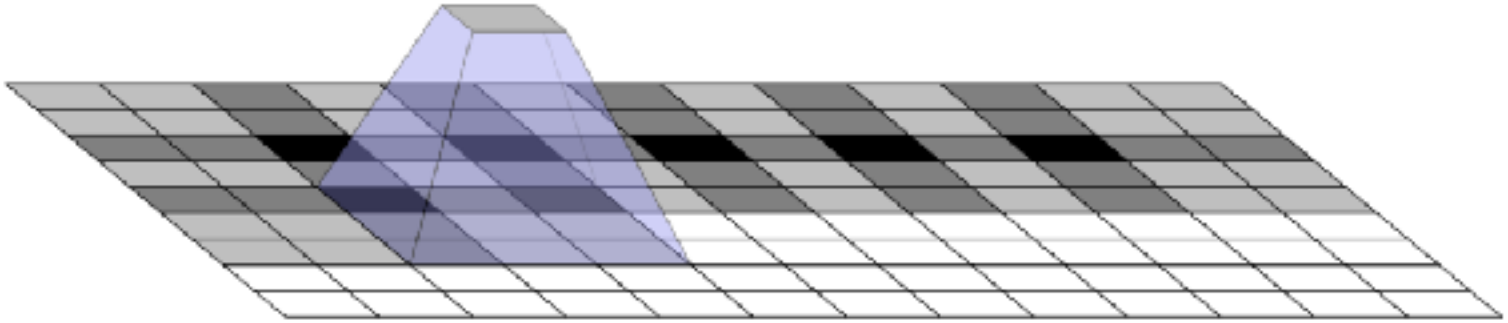
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



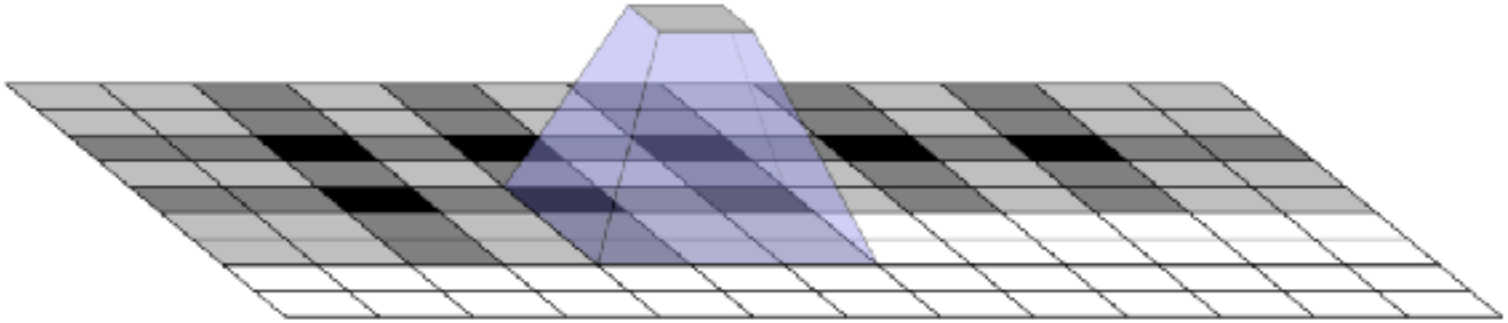
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



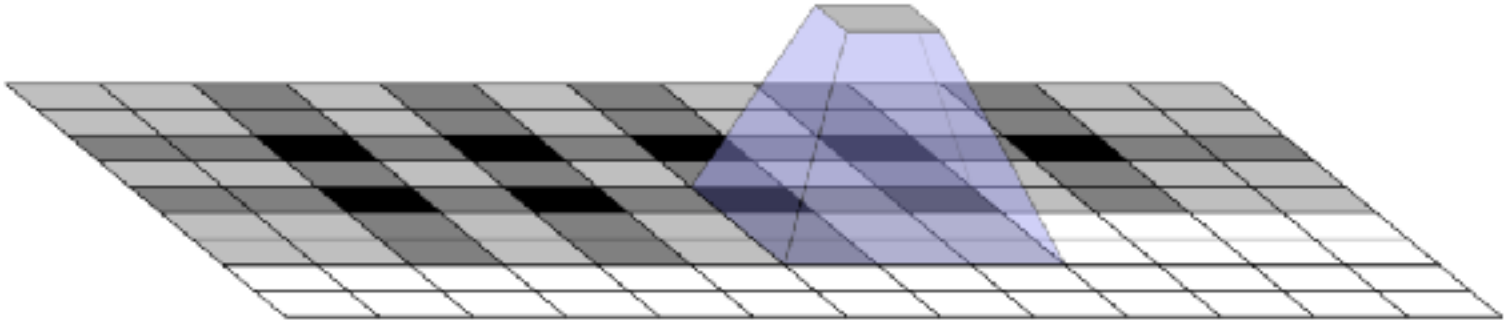
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



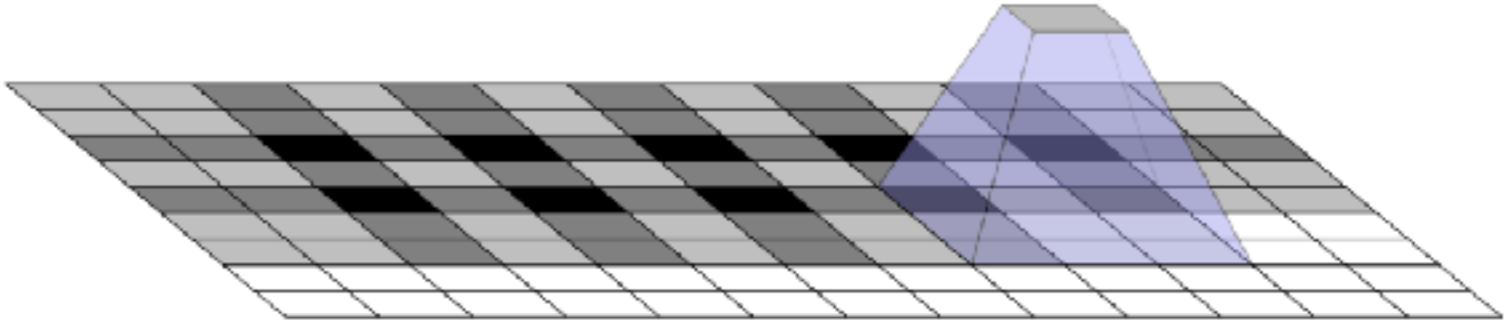
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



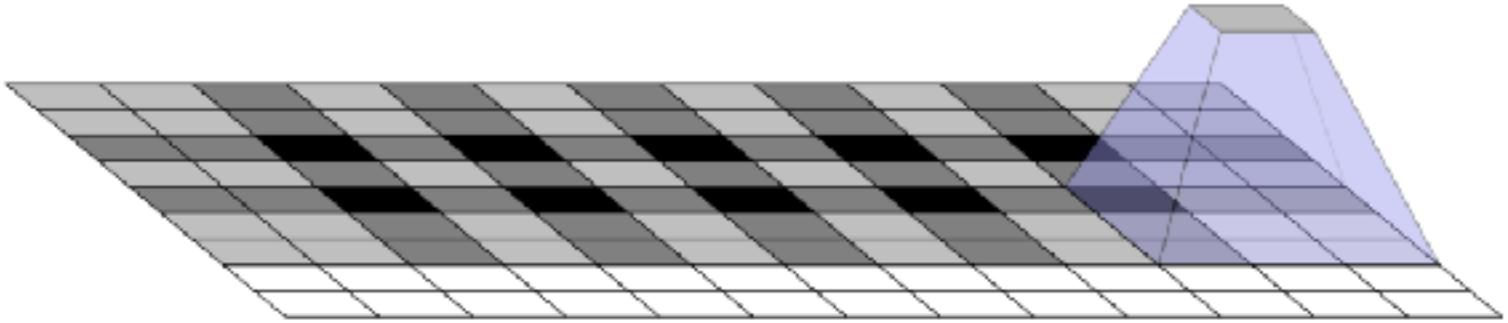
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



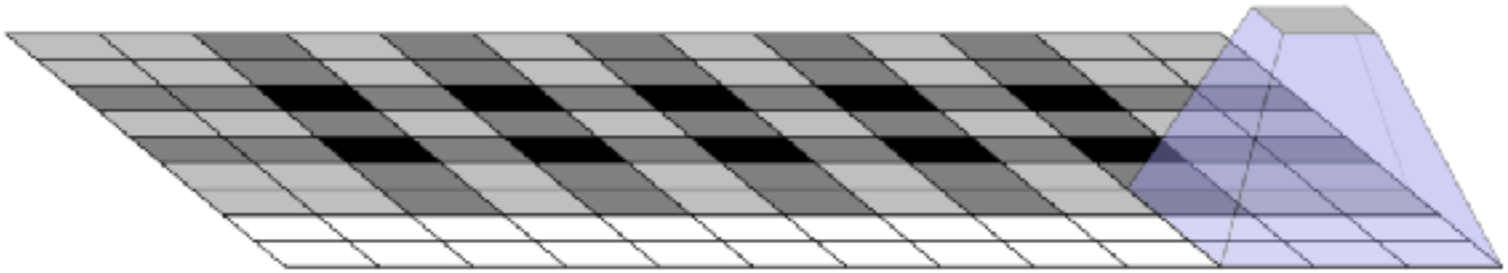
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



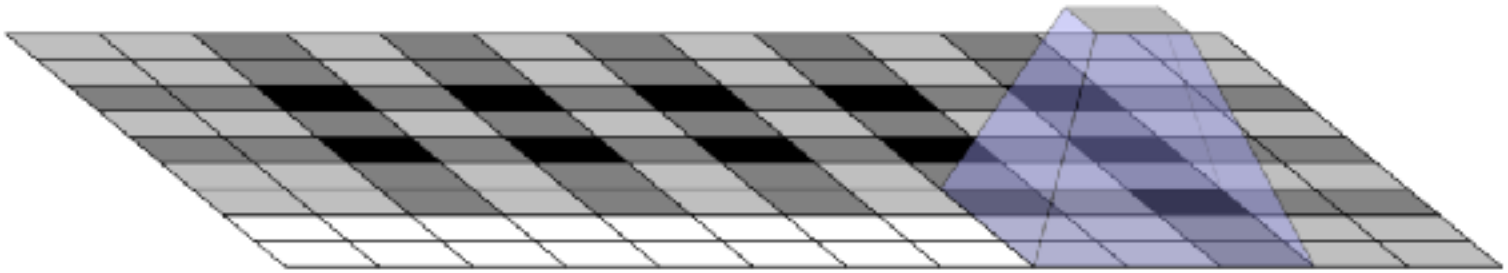
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



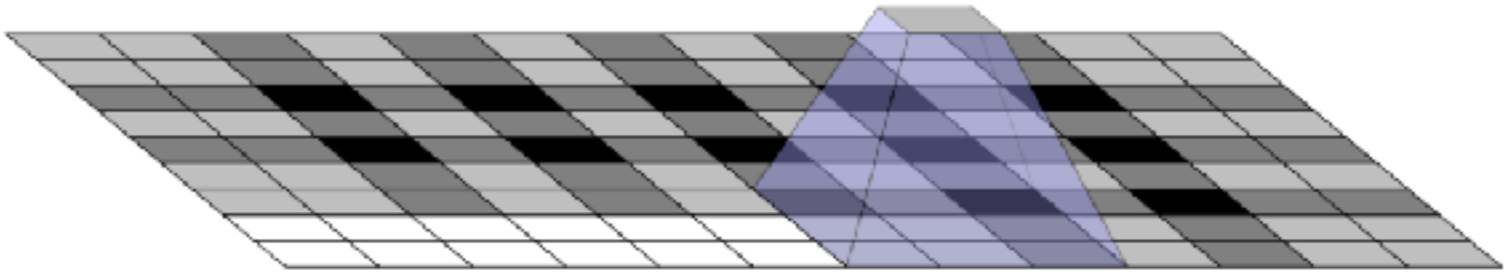
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



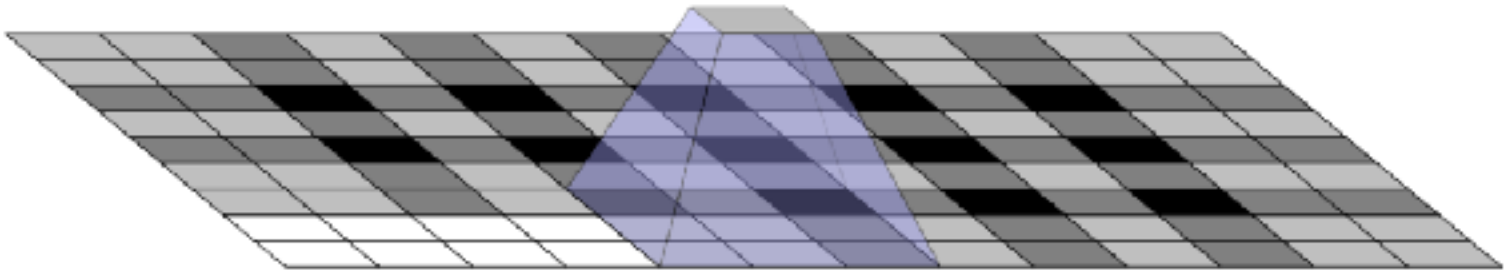
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



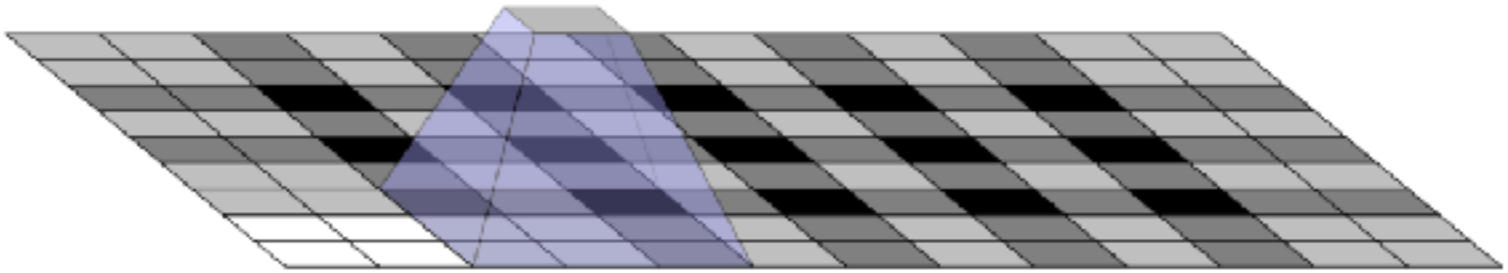
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



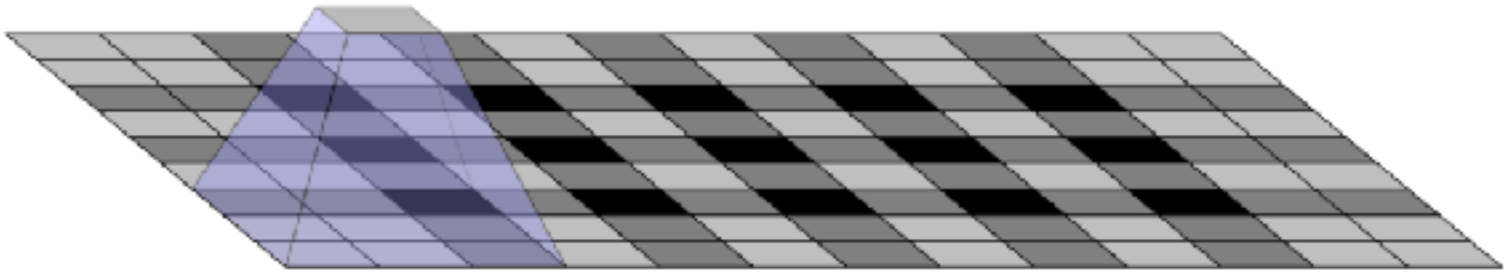
https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap



https://distill.pub/2016/deconv-checkerboard/

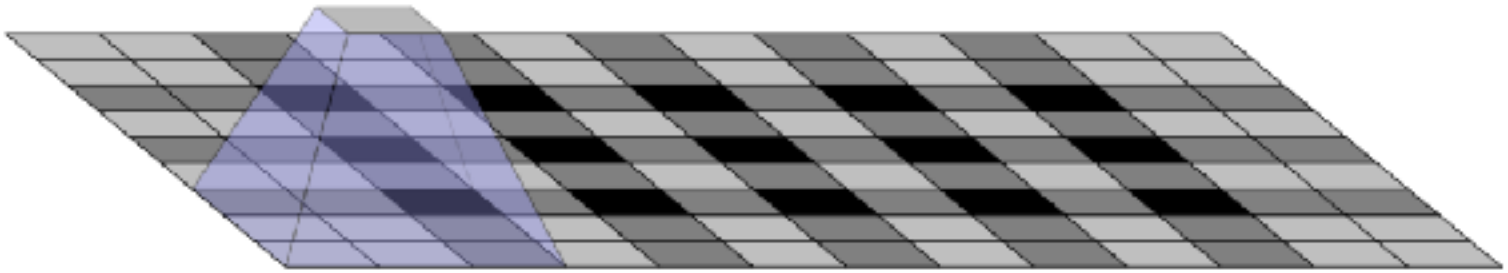
Transposed Convolution & Overlap



https://distill.pub/2016/deconv-checkerboard/

Transposed Convolution & Overlap

- In particular, deconvolution has uneven overlap when the **kernel size** (the output window size) is **not divisible** by the **stride** (the spacing between points on the top).

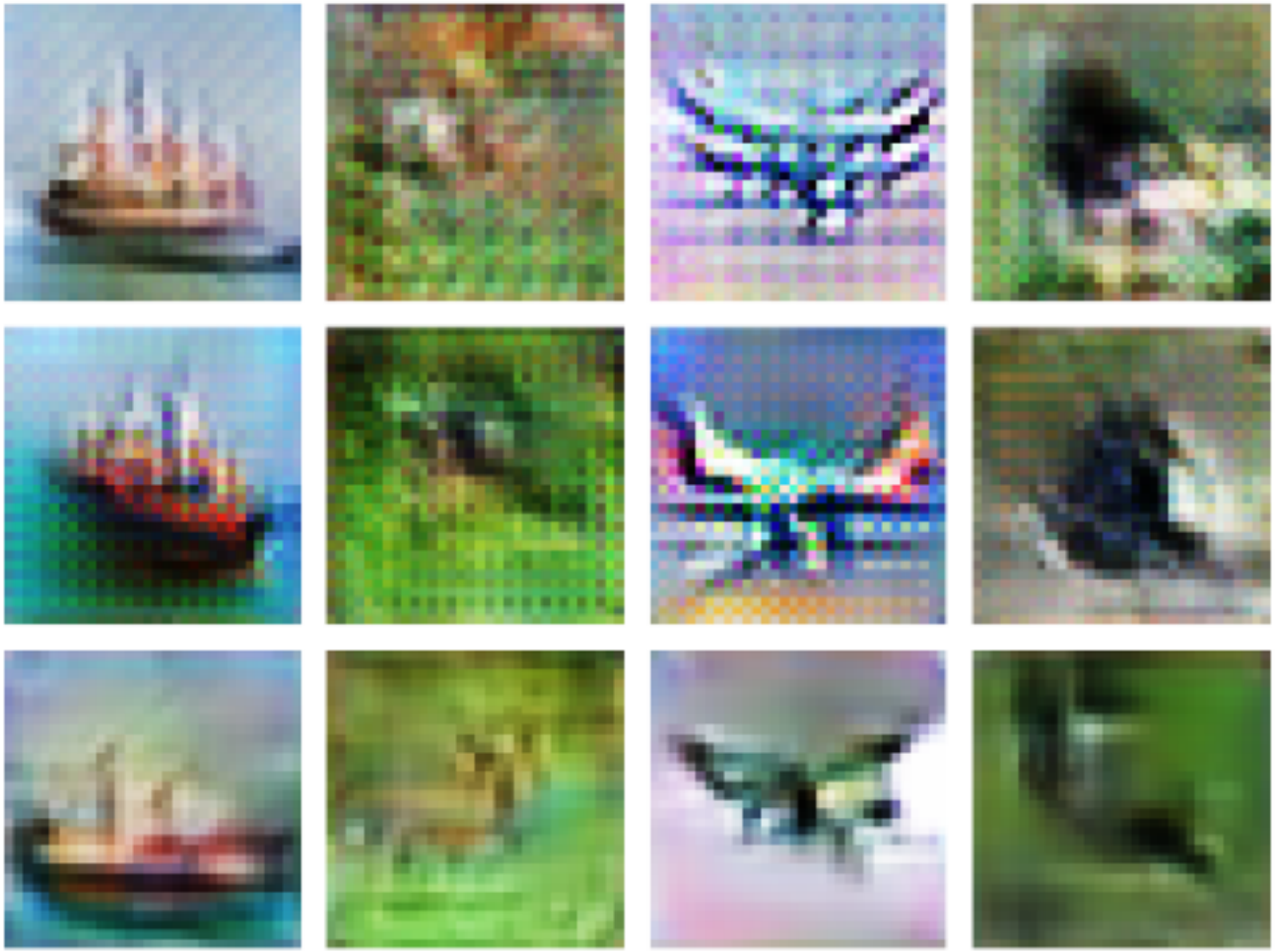


https://distill.pub/2016/deconv-checkerboard/

Resize Convolution

- Idea
  - Resize the image (using nearest-neighbor interpolation or bilinear interpolation) and then do a convolutional layer
- NN-Resize Convolution
- Bilinear-Resize Convolution

Resize Convolution

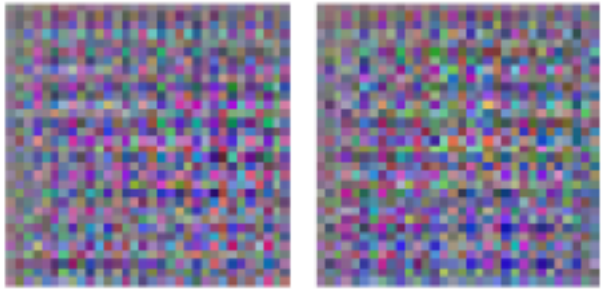


Deconv in last two layers.
Other layers use resize-convolution.
Artifacts of frequency 2 and 4.

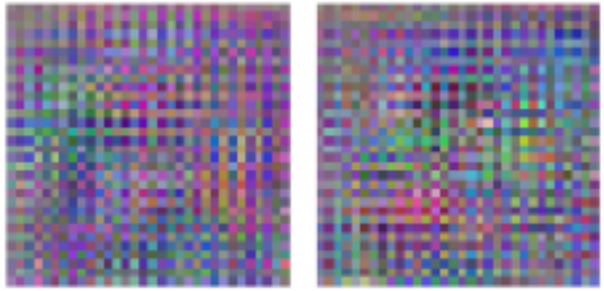
Deconv only in last layer.
Other layers use resize-convolution.
Artifacts of frequency 2.

All layers use resize-convolution.
No artifacts.

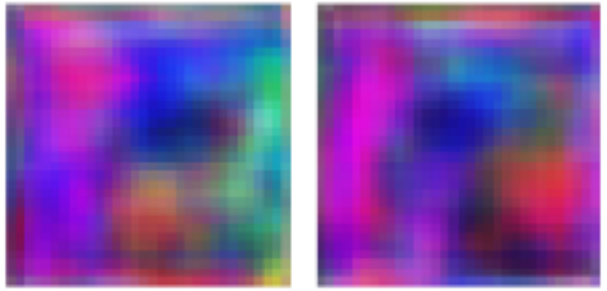
Resize Convolution



Deconvolution in last two layers.
Artifacts prior to any training.



Deconvolution only in last layer.
Artifacts prior to any training.



All layers use resize-convolution.
No artifacts before or after training.

Resize Convolution

Using deconvolution.
Heavy checkerboard artifacts.

Using resize-convolution.
No checkerboard artifacts.

Artifacts in Gradients - Resize Convolution

DeepDream only applying the neural network to a fixed position.
Severe artifacts.

DeepDream applying the network to a different position each step.
Reduced artifacts.