Lecture Notes for

Neural Networks and Machine Learning

Fully Convolutional Learning I: Introduction to Semantic Segmentation

Logistics and Agenda

- Logistics
  - Lab Grading Update
- Agenda
  - Segmentation
    - Intro to Semantic (this time)
    - Object (partially this time)
    - Instance (next time)

Types of Fully Convolutional Problems

- Semantic Segmentation
- Object Detection
- Instance Segmentation

Introduction to Semantic Segmentation

Karandeep Singh @kdpsinghlab · 10h ···

Statistician: Do you ever use statistics?

ML researcher: Nope. Never.

Statistician: What about when reading a paper?

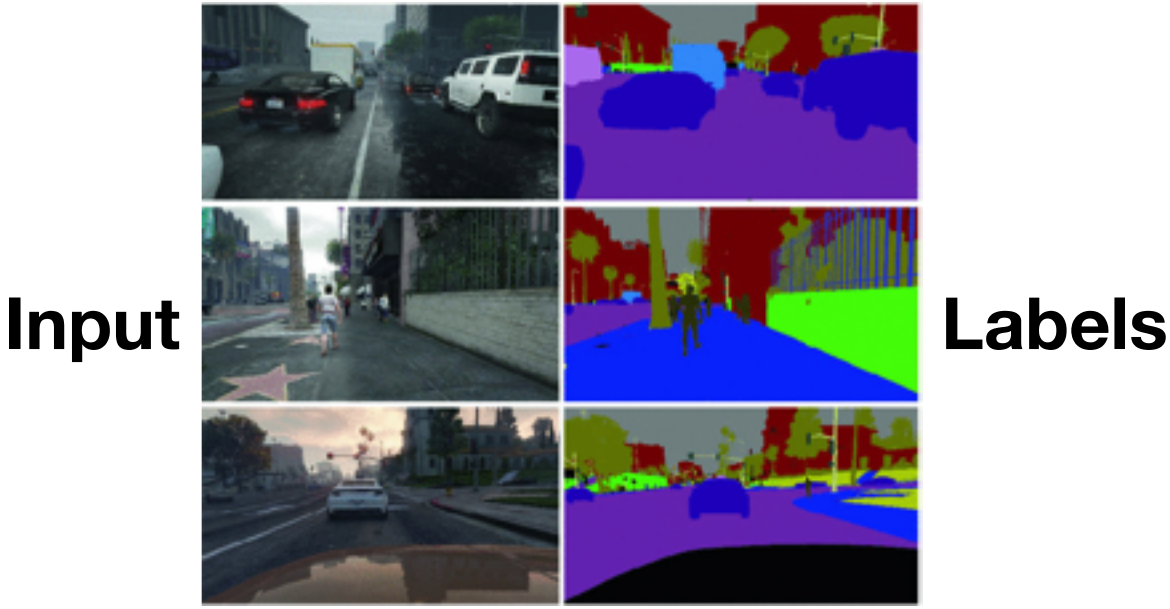
ML: Nope. Never.

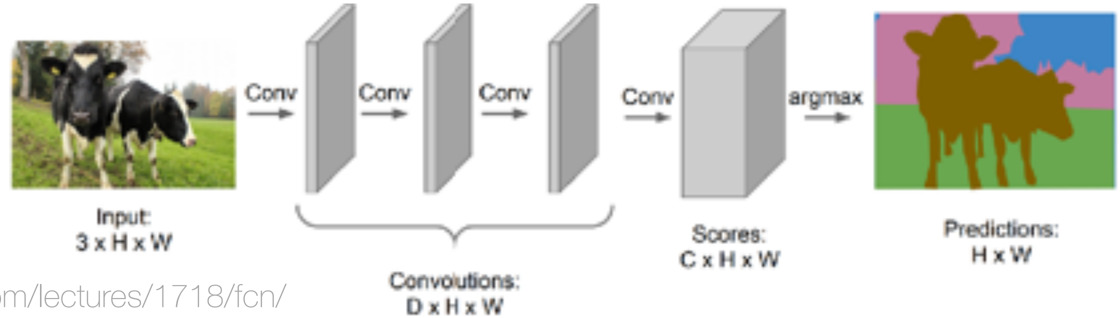
Statistician: Ok. So if you're reading an ML paper comparing lots of models, how do you know which one is the best?

ML: Bold font.

Semantic Segmentation

- Given a set of pixels, classify each pixel according to what instance it belongs

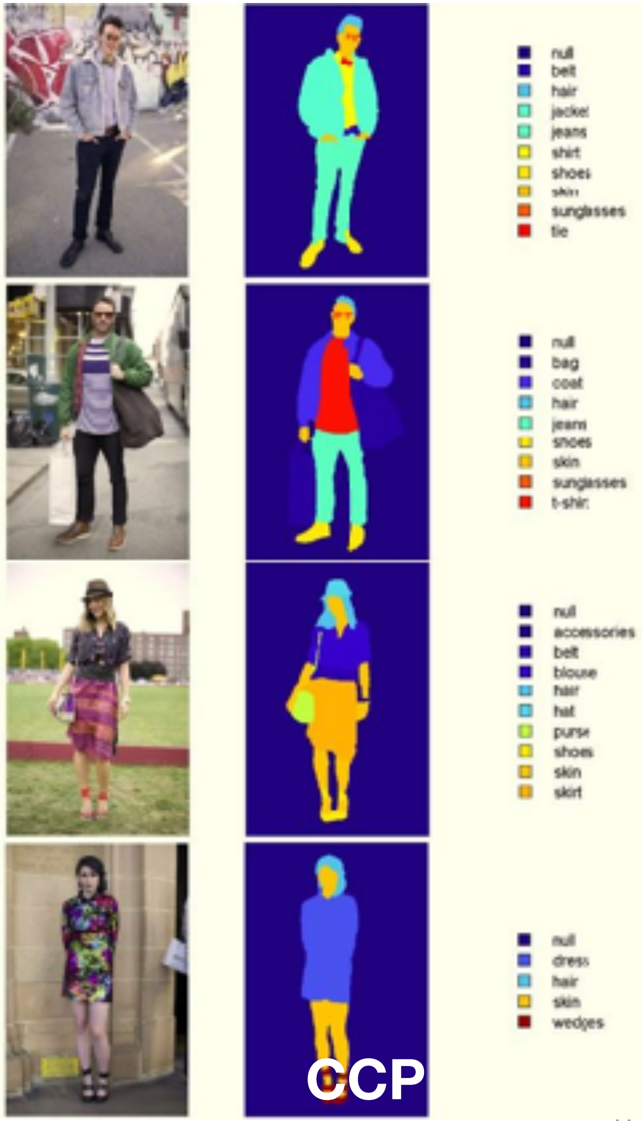




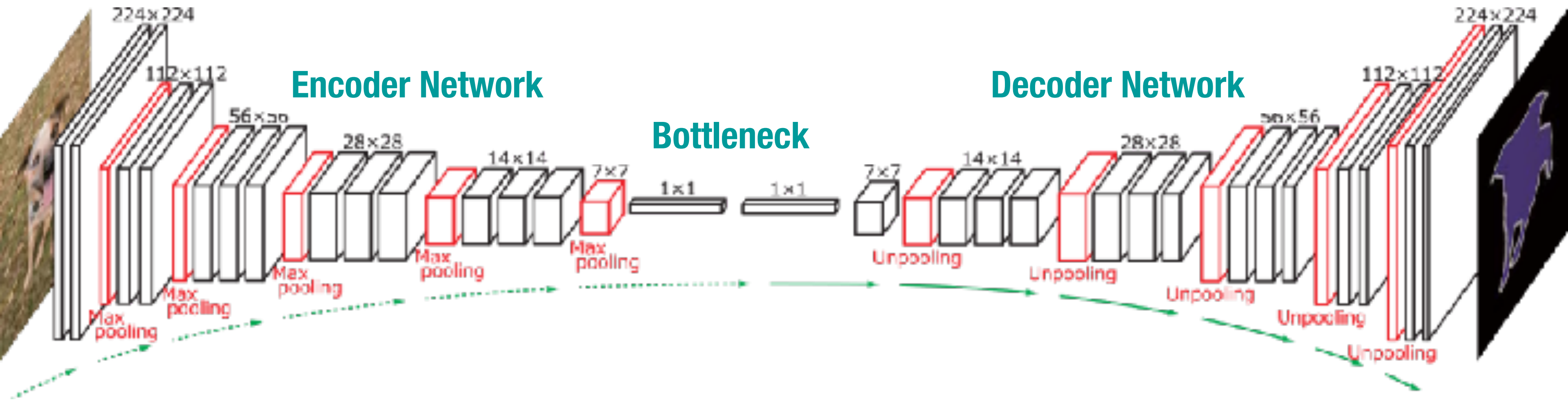
https://tjmachinelearning.com/lectures/1718/fcn/

Popular Semantic Segmentation Datasets

COCO http://cocodataset.org/



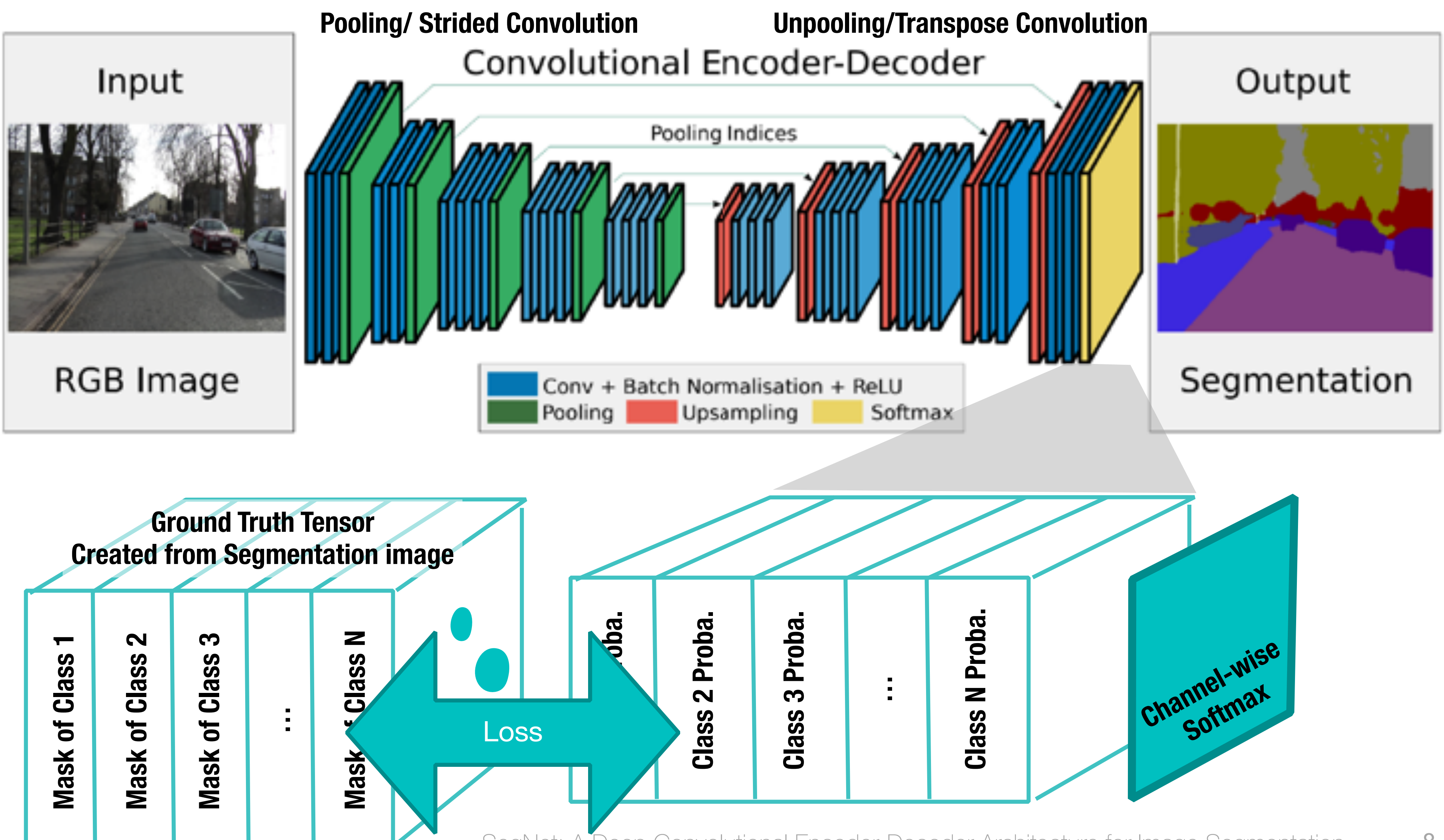
Early Training Methods (Pre 2018)



- Init Encoder with traditional CNN (like VGG or DarkNet)
- Freeze encoder and train decoder with segmented image maps
- Unfreeze encoder and fine tune
  - Repeat tuning as needed

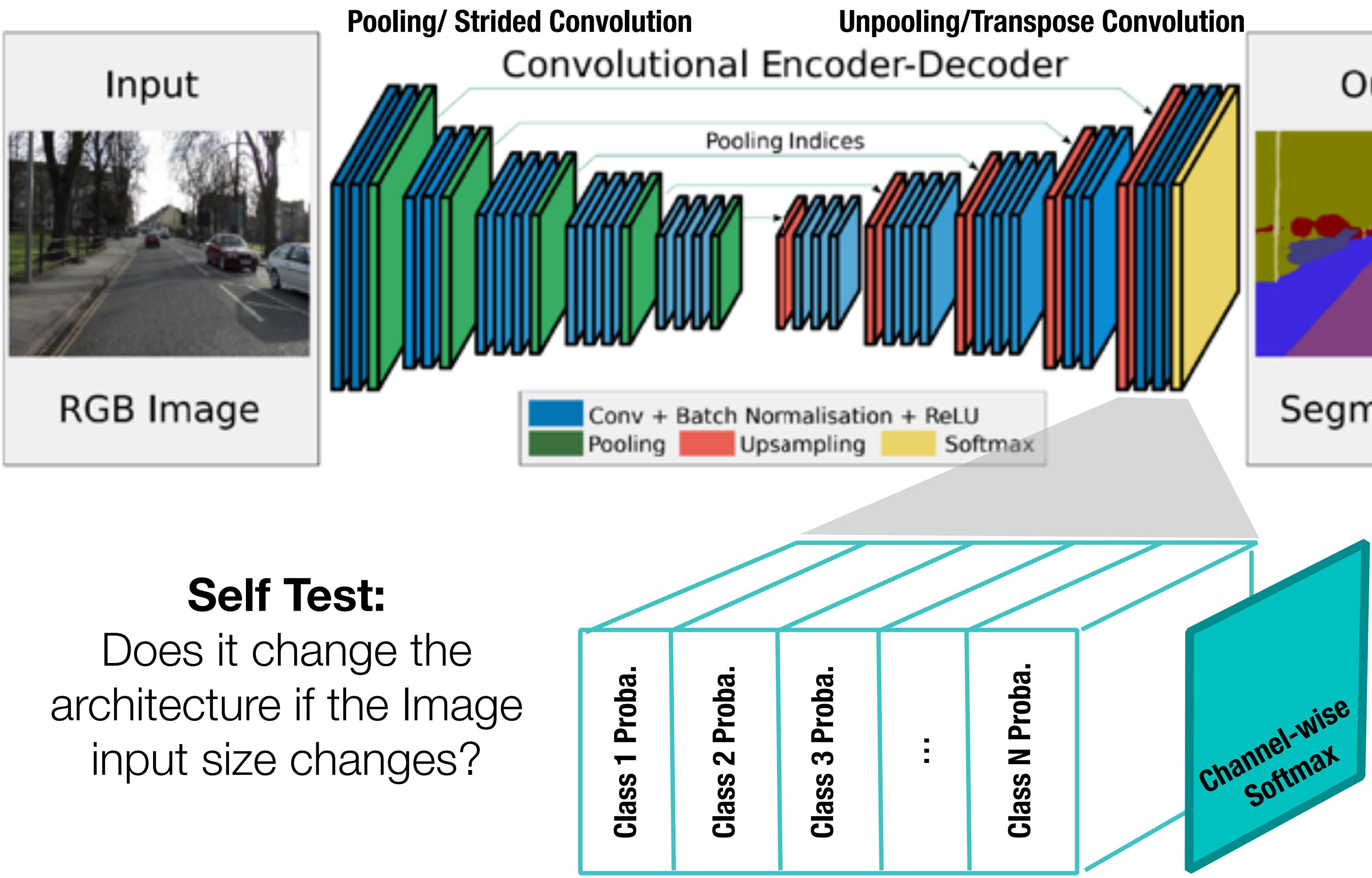
https://tjmachinelearning.com/lectures/1718/fcn/

Putting it all together



SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation

Putting it all together



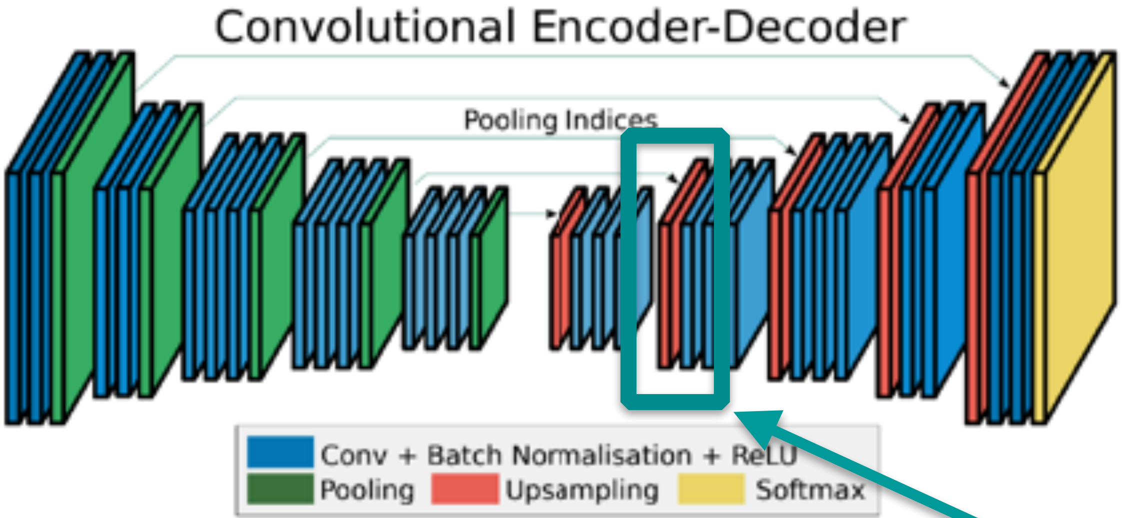
Self Test:

Does it change the architecture if the Image input size changes?

SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation

Upsampling Layers

Decoder Network



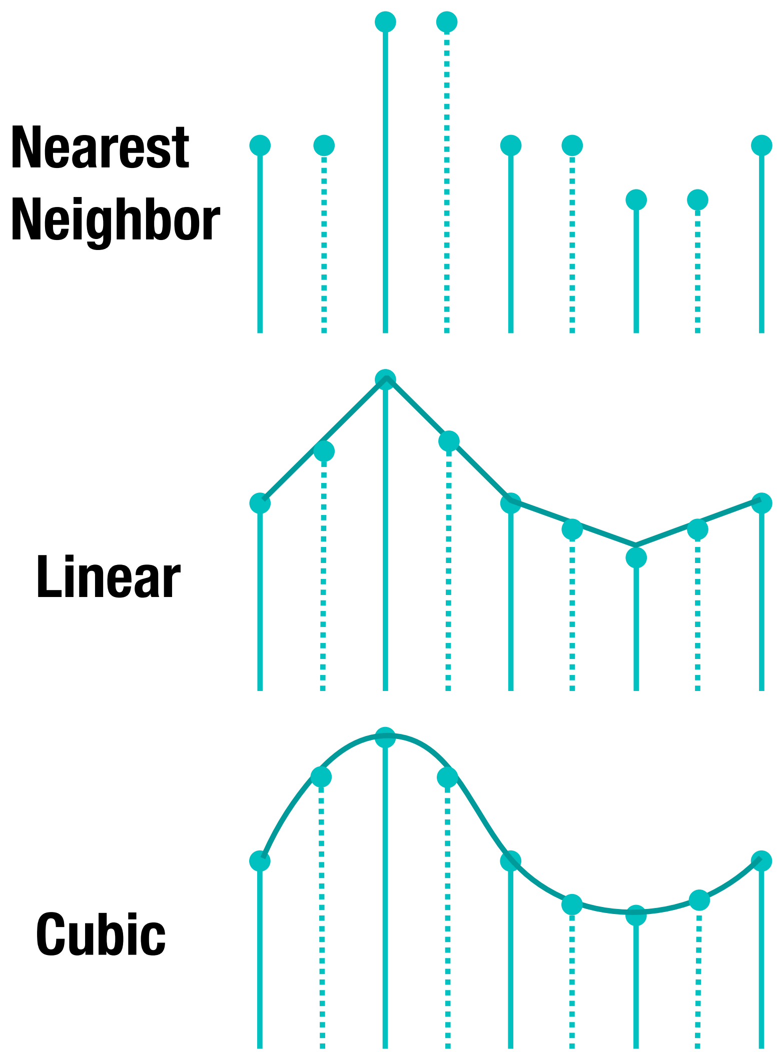
Some researcher started calling this **deconvolution**.

If you use that term in this class, **you fail**.

This is upsampling and then convolution, but **now the interpolation filters are learned**!!

https://tjmachinelearning.com/lectures/1718/fcn/

Integer Upsampling via Interpolation



All are equivalent to inserting zeros and applying convolutional filter

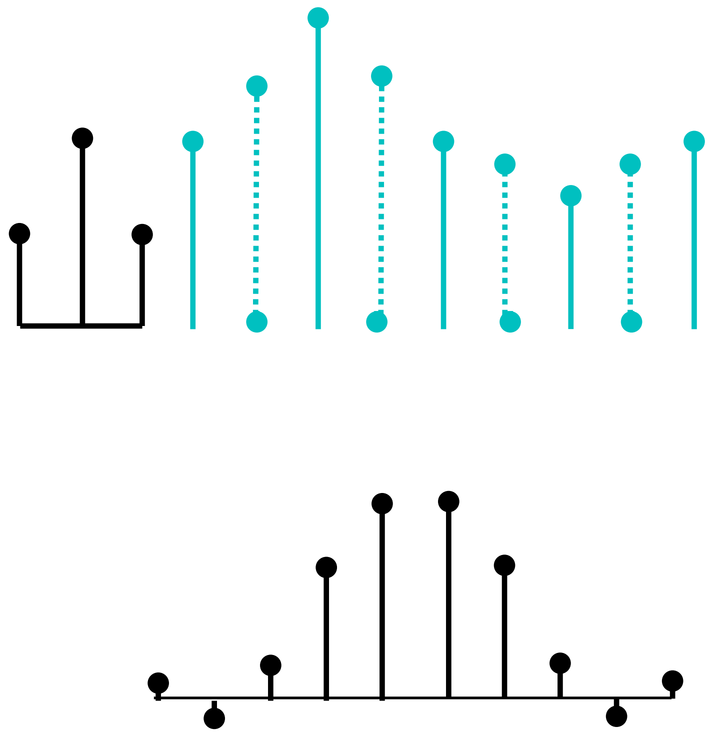


Image Upsampling, Integer Factor

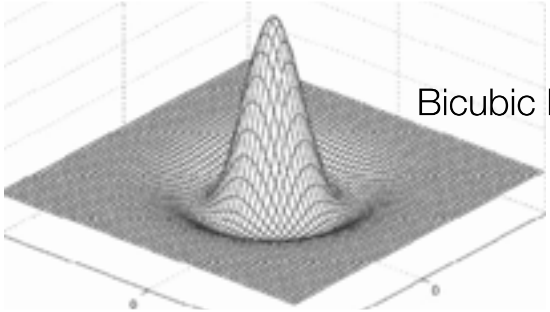
- Insert Zeros
- Convolve

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

1		2		3		4	
5		6		7		8	
9		10		11		12	
13		14		15		16	

0.25	0.5	0.25
0.5	1	0.5
0.25	0.5	0.25

Bilinear Filtering



Bicubic Filter

Image Upsampling, Integer Factor

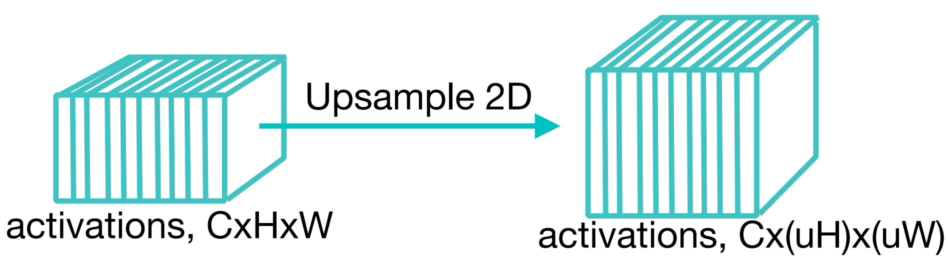
Nearest Neighbor

`UpSampling2D()`

Bilinear

`UpSampling2D(interpolation='bilinear')`

Bicubic



Many Types of Upsampling, with varying computational cost:
area, bicubic, gaussian, lanczos3, lanczos5, mitchellcubic

https://www.cs.toronto.edu/~guerzhoy/320/lec/upsampling.pdf

What about transpose convolution?

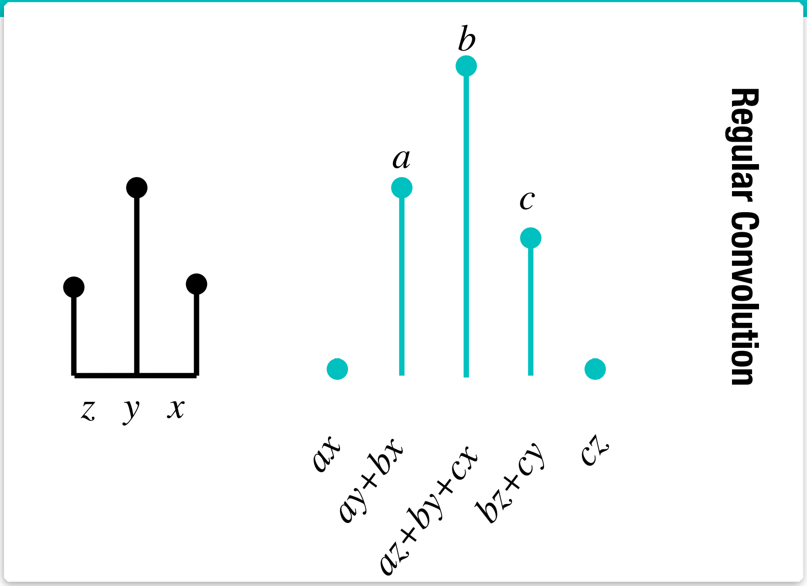
Convolution as Matrix Multiplication

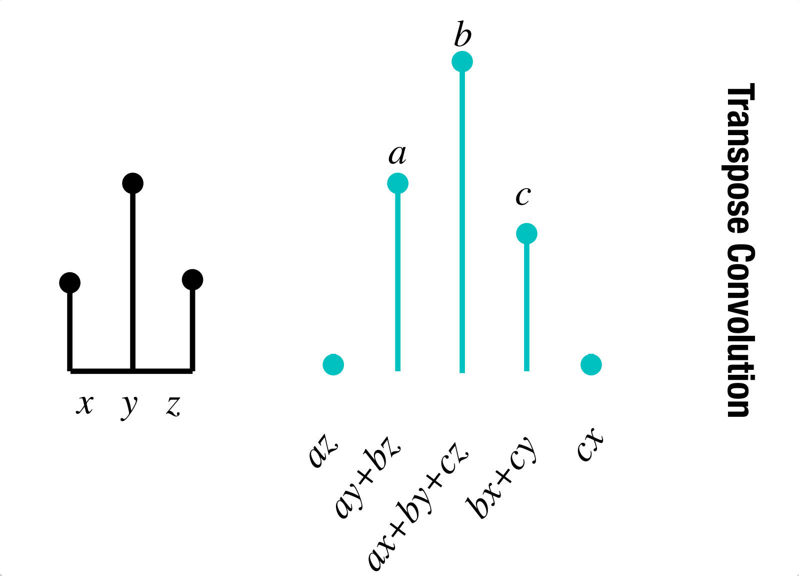
$$\begin{bmatrix} y & x & 0 & 0 & 0 \\ z & y & x & 0 & 0 \\ 0 & z & y & x & 0 \\ 0 & 0 & z & y & x \\ 0 & 0 & 0 & z & y \end{bmatrix} \times \begin{bmatrix} 0 \\ a \\ b \\ c \\ 0 \end{bmatrix} = \begin{bmatrix} ax \\ ay+bx \\ az+by+cx \\ bz+cy \\ cz \end{bmatrix}$$

Transpose

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

like convolving with “reversed coefficients”





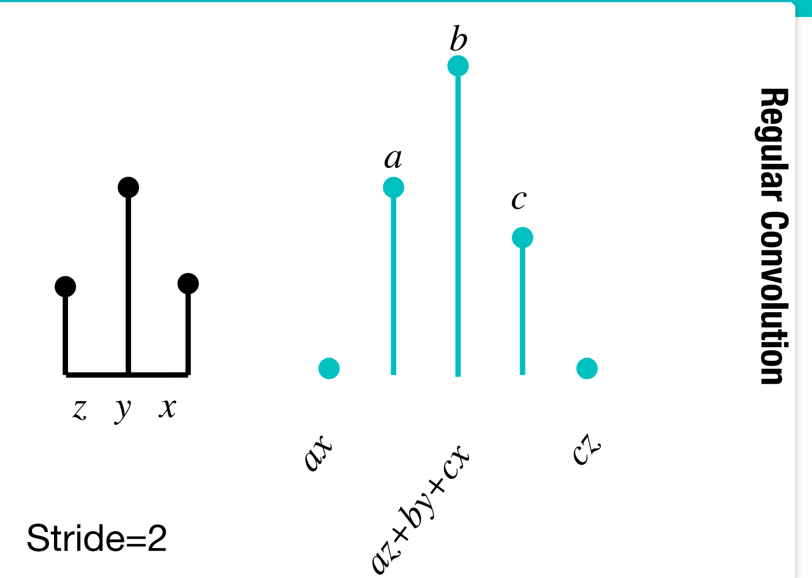
Transpose Convolution: Strides

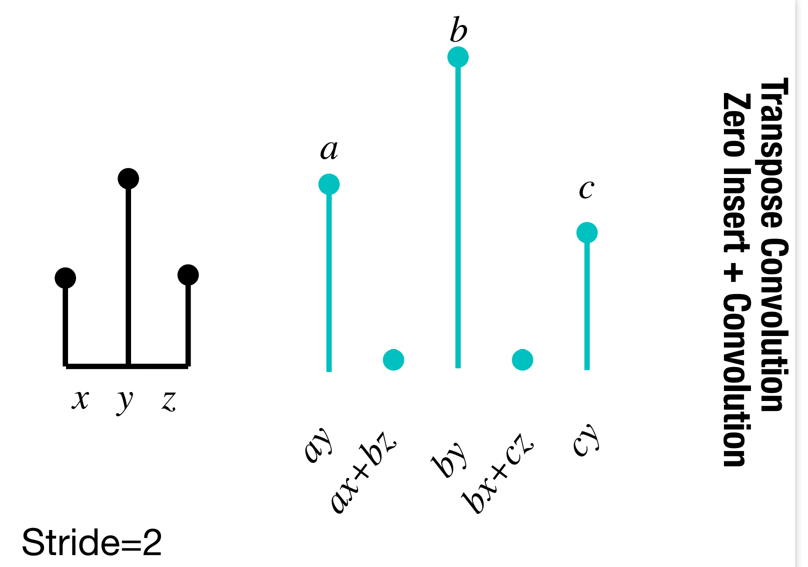
Strided Convolution as Matrix Multiplication

$$\begin{bmatrix} y & x & 0 & 0 & 0 \\ 0 & z & y & x & 0 \\ 0 & 0 & 0 & z & y \end{bmatrix} \times \begin{bmatrix} 0 \\ a \\ b \\ c \\ 0 \end{bmatrix} = \begin{bmatrix} ax \\ az+by+cx \\ cz \end{bmatrix}$$

Transpose

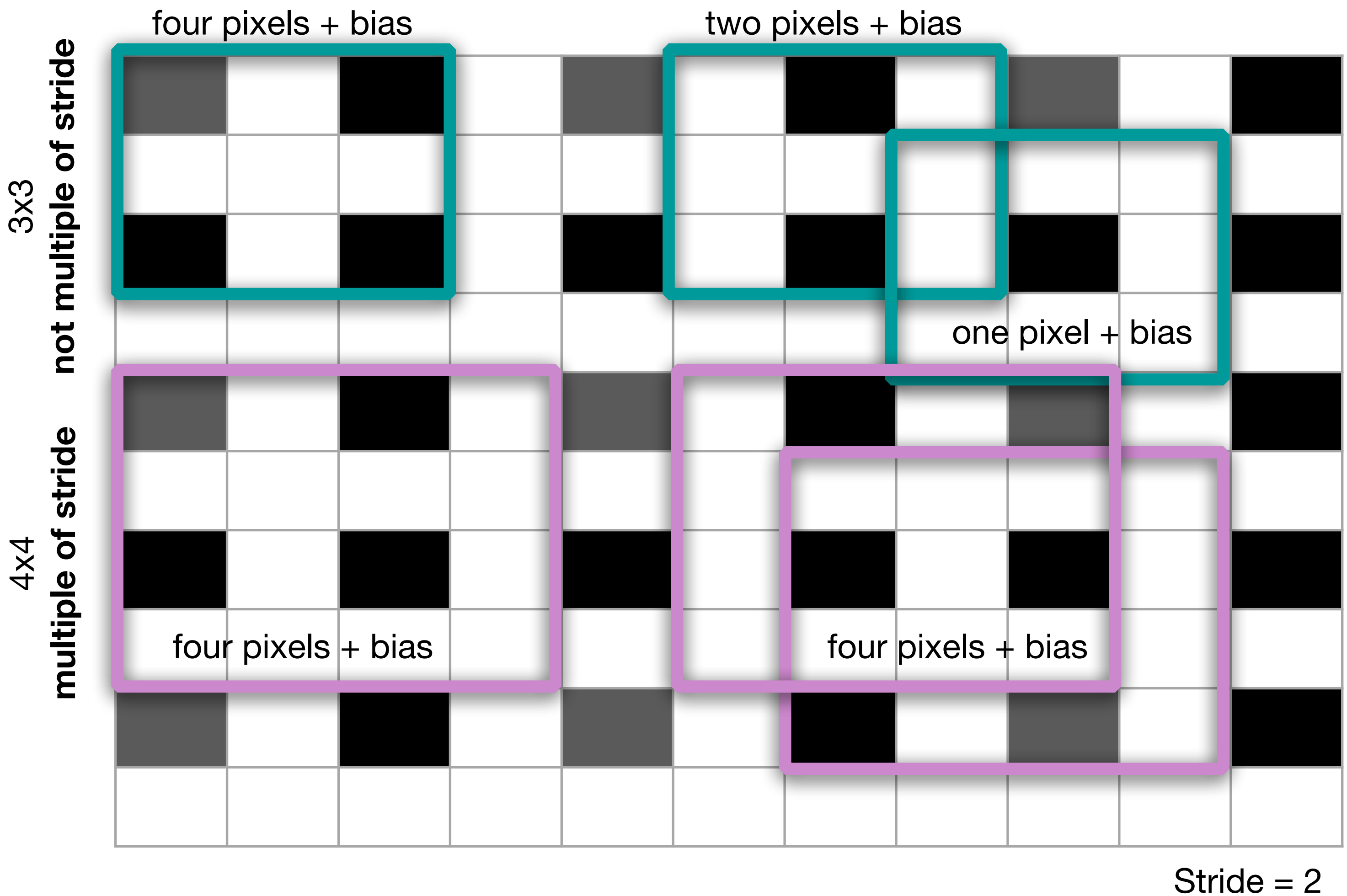
$$\begin{bmatrix} y & 0 & 0 \\ x & z & 0 \\ 0 & y & 0 \\ 0 & x & z \\ 0 & 0 & y \end{bmatrix} \times \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} ay \\ ax+bz \\ by \\ bx+cz \\ cy \end{bmatrix}$$





Convolution after zero insertion

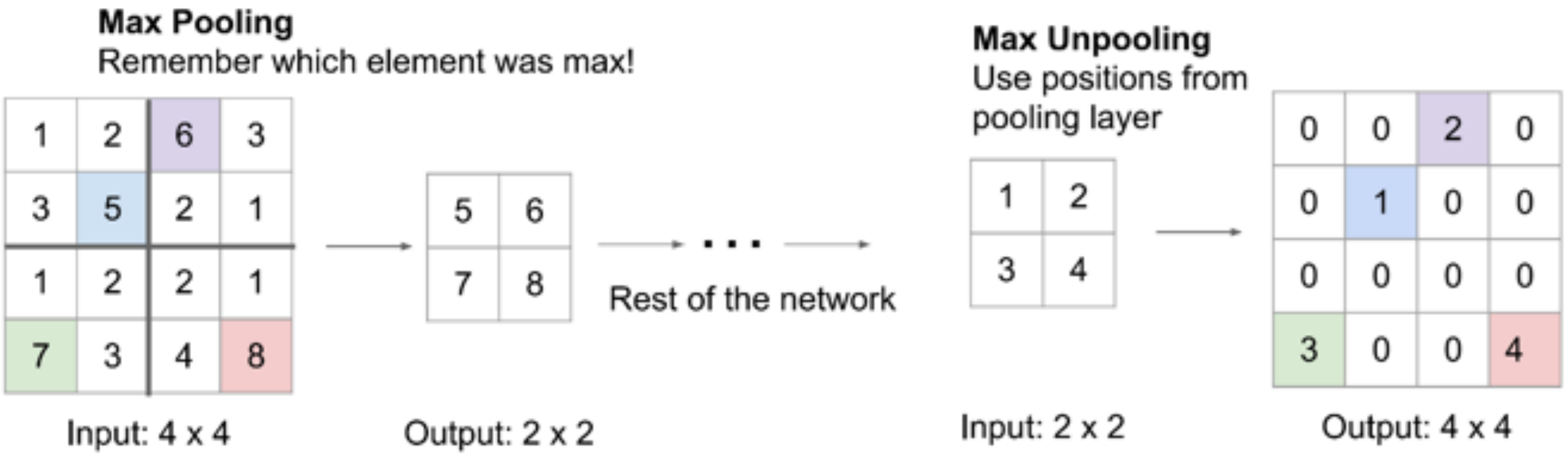
- Kernel size should be a symmetric multiple of the stride



Bias needs to account for both when different numbers of pixels overlap with the kernel

Multiple of stride ensures that same number of active pixels overlap the kernel.

Unpooling: a different method of zero insertion



- Unpooling: insert values to upsample where you pooled
- Why does this make sense? The upsampling happens much later in the network…
- And it increases computational overhead and memory to track indices…
- Not very advantageous…

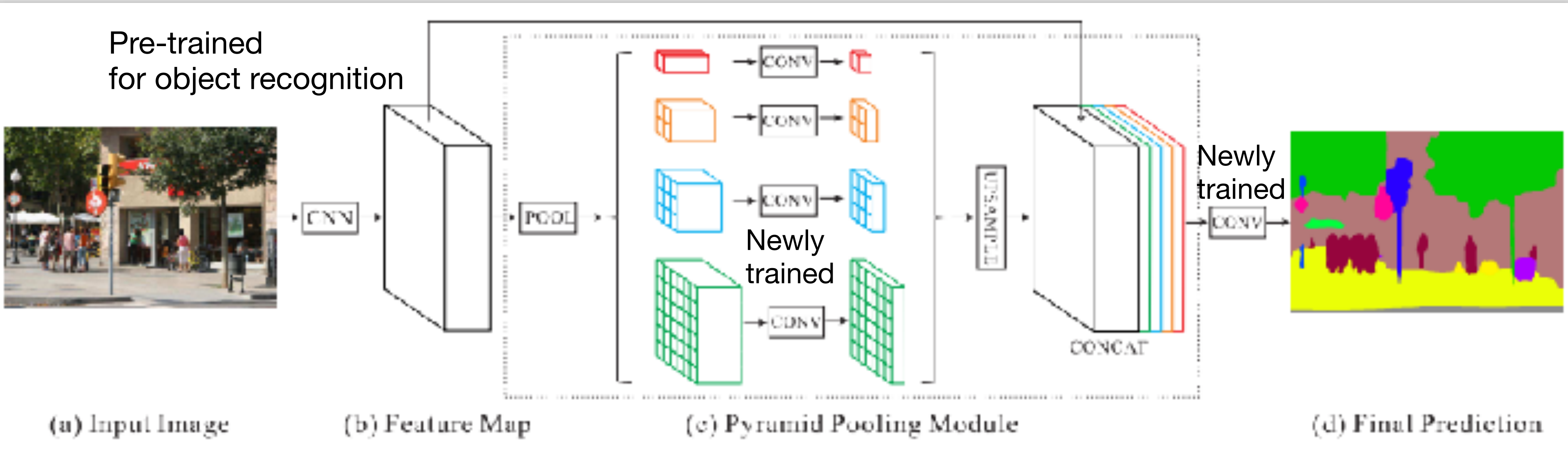
https://tjmachinelearning.com/lectures/1718/fcn/

Back up Slides for Semantic Segmentation

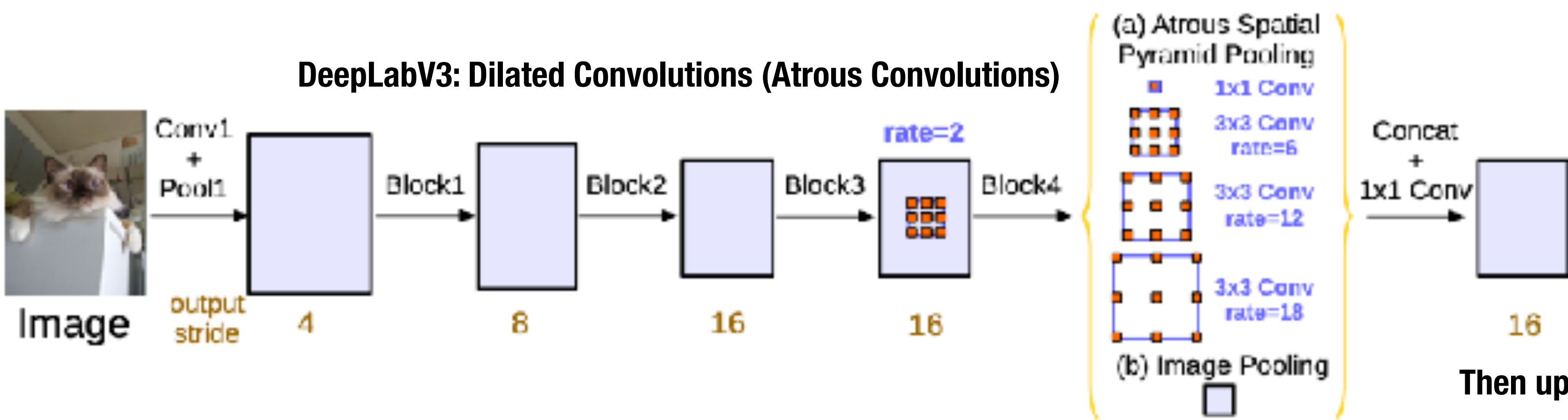
Some Examples

REFERENCE SLIDE

Pyramid Scene Parsing Network (PSPNet)



DeepLabV3: Dilated Convolutions (Atrous Convolutions)

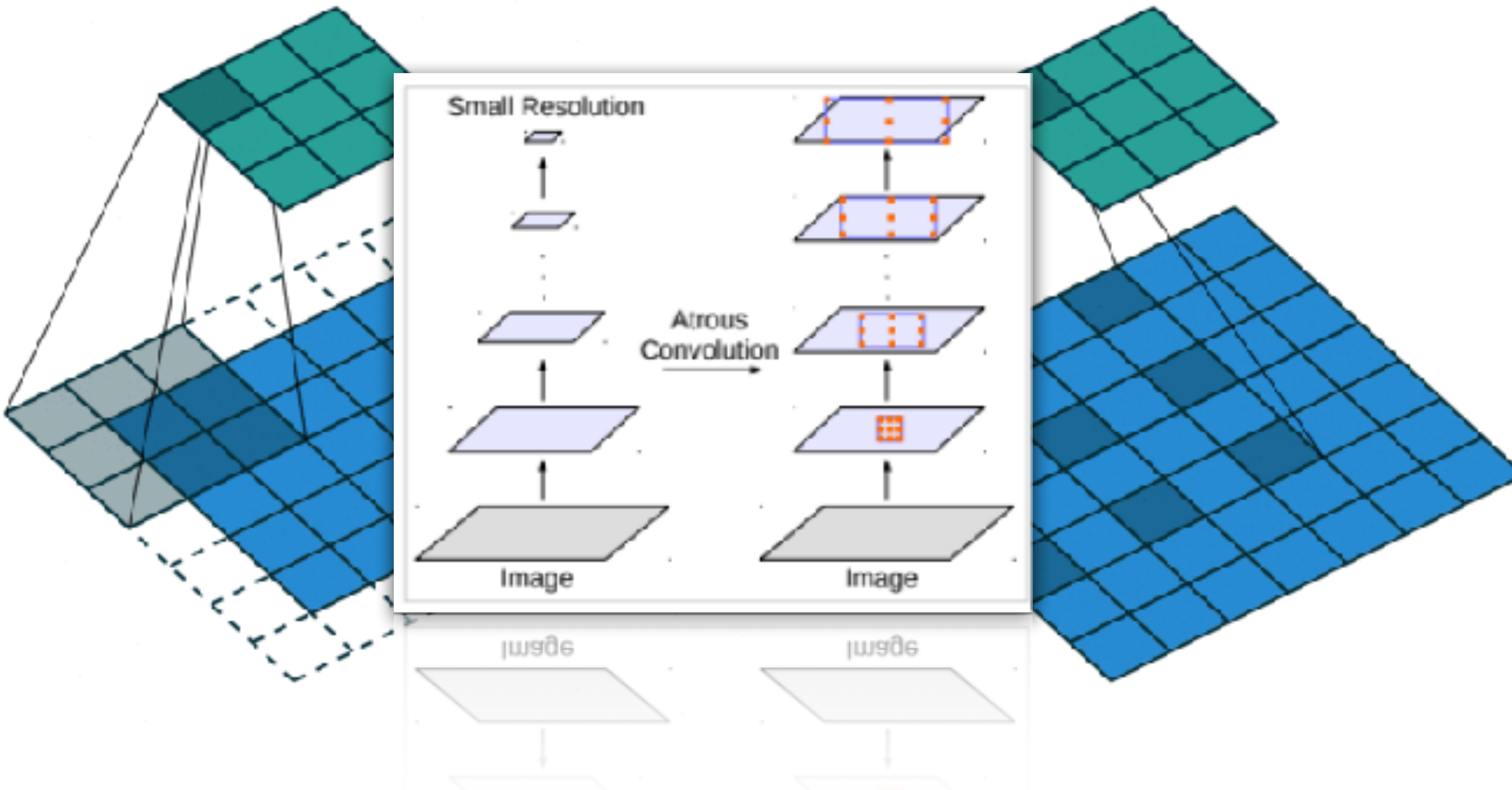


Then upscaling→

https://towardsdatascience.com/semantic-segmentation-with-deep-learning-a-guide-and-code-e52fc8958823

Dilated Convolution (Atrous)

REFERENCE SLIDE

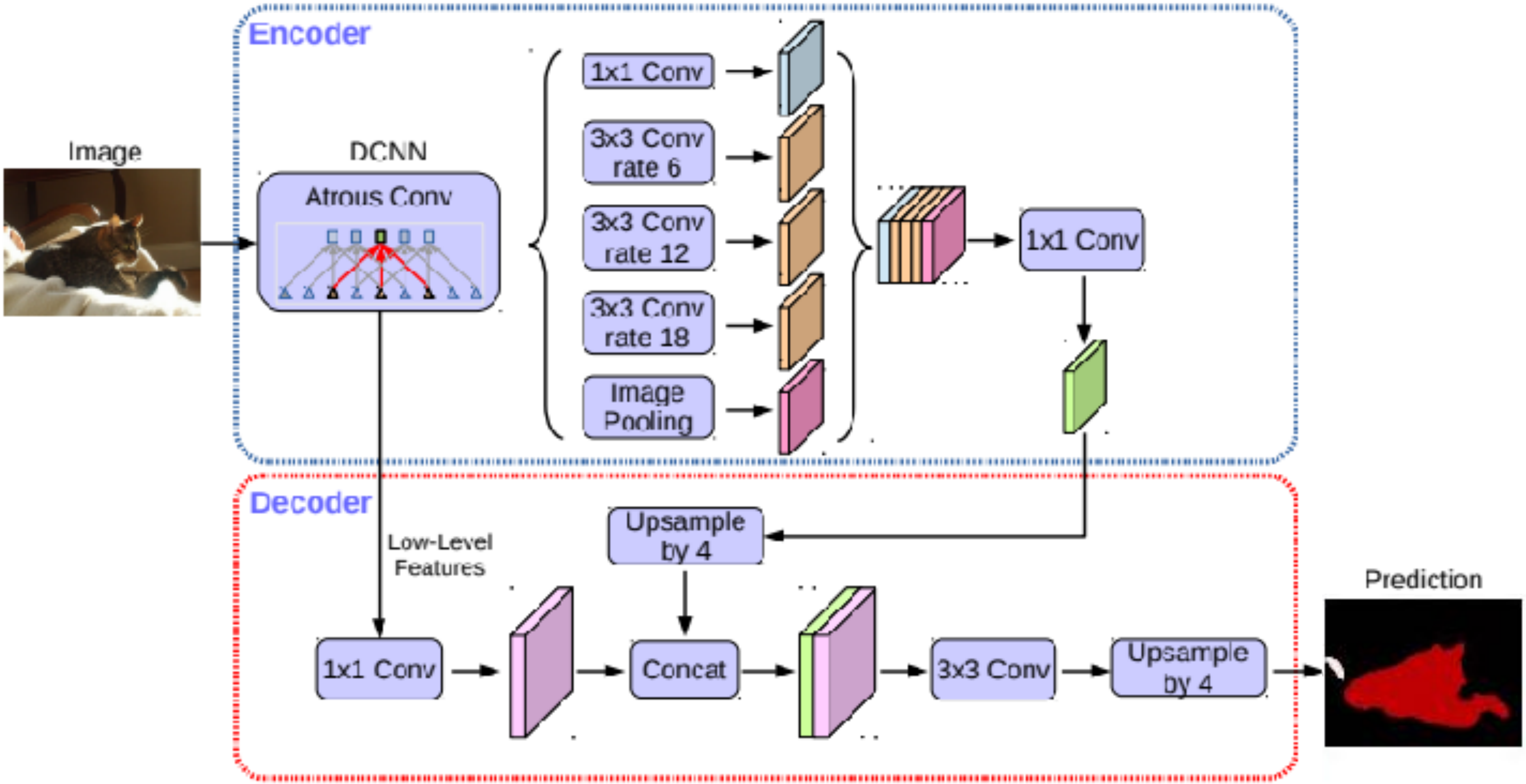


Outputs of convolution are the same size, except for edge effects!
But have advantage of processing at a different scale.

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

DeepLabV3+

REFERENCE SLIDE



https://github.com/tensorflow/models/tree/master/research/deeplab

https://towardsdatascience.com/semantic-segmentation-with-deep-learning-a-guide-and-code-e52fc8958823

Gated-SCNN (Gate Shape CNN)

REFERENCE SLIDE

1. Shape stream employs Traditional Image Processing for edge detection (**image gradients**)

2. Uses activations to "gate" the image gradient. $\sigma(A) \odot I_{grad}$

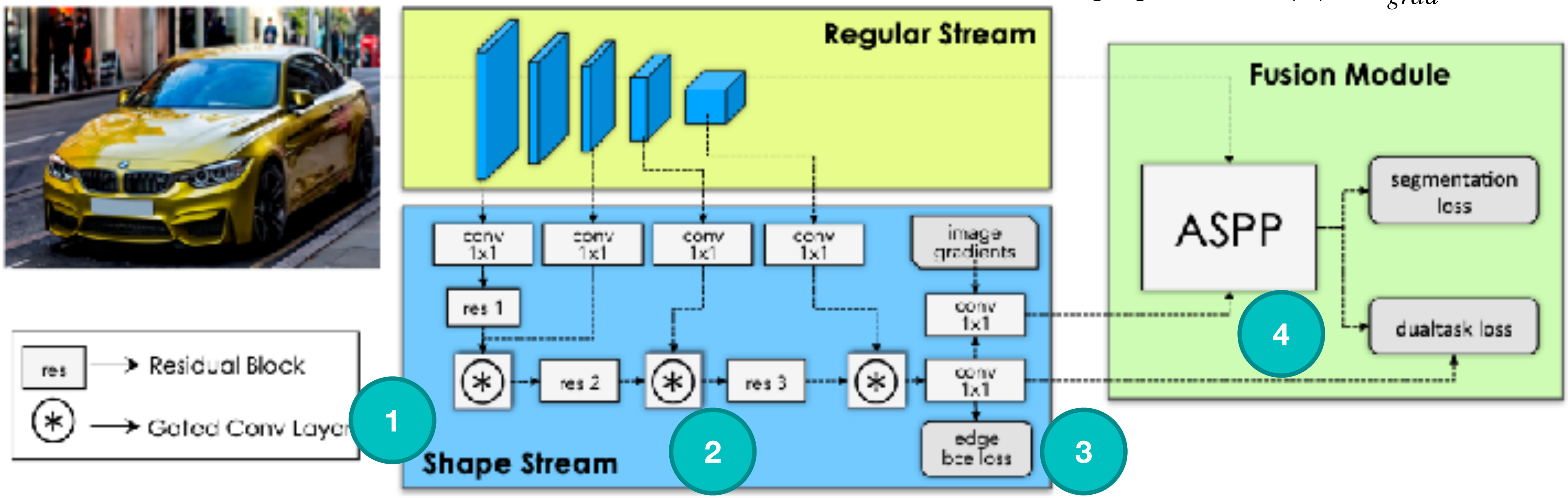


Figure 2: **GSCNN architecture.** Our architecture constitutes of two main streams. The regular stream and the shape stream. The regular stream can be any backbone architecture. The shape stream focuses on shape processing through a set of residual blocks, Gated Convolutional Layers (GCL) and supervision. A fusion module later combines information from the two streams in a multi-scale fashion using an Atrous Spatial Pyramid Pooling module (ASPP). High quality boundaries on the segmentation masks are ensured through a Dual Task Regularizer .

3. Also uses Labeled Boundaries in BCE Edge Loss Function

4. Merges segmentation with edges for finer masks. Concatenate + atrous convolution

https://heartbeat.fritz.ai/a-2019-guide-to-semantic-segmentation-ca8242f5a7fc

Performance

REFERENCE SLIDE

Figure 3: Illustration of the crops used for the distance-based evaluation.

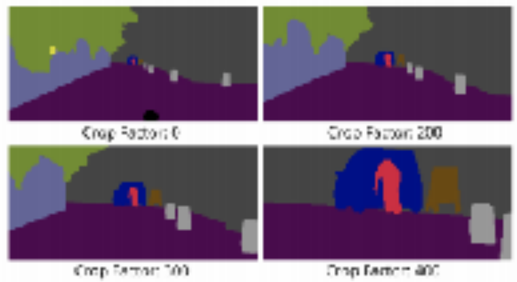


Figure 4: Predictions at diff. crop factors.

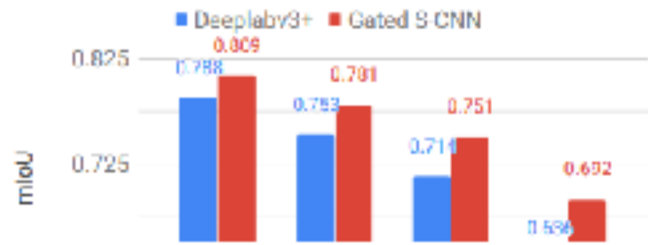
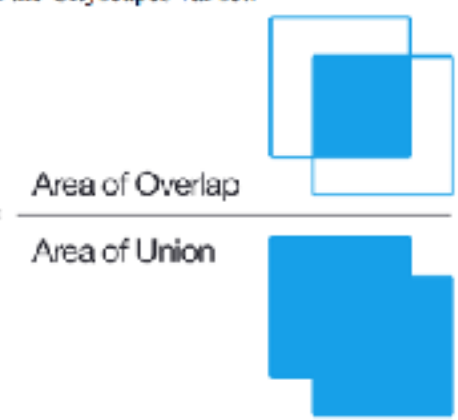


Figure 5: **Distance-based evaluation**: Comparison of mIoU at different crop factors.

Method	road	s.walk	build.	wall	fence	pole	t-light	t-sign	veg	terrain	sky	person	rider	car	truck	bus	train	motor	bike	mean
LRR [18]	97.7	79.9	90.7	44.4	48.6	58.6	68.2	72.0	92.5	69.3	94.7	81.6	60.0	94.0	43.6	56.8	47.2	54.8	69.7	69.7
DeepLabV2 [9]	97.9	81.3	90.3	48.8	47.4	49.6	57.9	67.3	91.9	69.4	94.2	79.8	59.8	93.7	56.5	67.5	57.5	57.7	68.8	70.4
Piecewise [32]	98.0	82.6	90.6	44.0	50.7	51.1	65.0	71.7	92.0	72.0	94.1	81.5	61.1	94.3	61.1	65.1	53.8	61.6	70.6	71.6
PSP-Net [58]	98.2	85.8	92.8	57.5	65.9	62.6	71.8	80.7	92.4	64.5	94.8	82.1	61.5	95.1	78.6	88.3	77.9	68.1	78.0	78.8
DeepLabV3+ [11]	98.2	84.9	92.7	**57.3**	62.1	65.2	68.6	78.9	92.7	63.5	95.3	82.3	62.8	95.4	**85.3**	89.1	80.9	64.6	77.3	78.8
Ours (GSCNN)	**98.3**	**86.3**	**93.3**	55.8	64.0	**70.8**	**75.9**	**83.1**	**93.0**	**65.1**	**95.2**	**85.3**	**67.9**	**96.0**	80.8	**91.2**	**83.3**	**69.6**	**80.4**	**80.8**

Table 1: Comparison in terms of IoU vs state-of-the-art baselines on the Cityscapes val set.

mIoU == mean Intersection over Union =



Lecture Notes for

Neural Networks and Machine Learning

FCN Learning

Next Time:
Fully Convolutional Objects

Reading: None