EE655: Computer Vision & Deep Learning

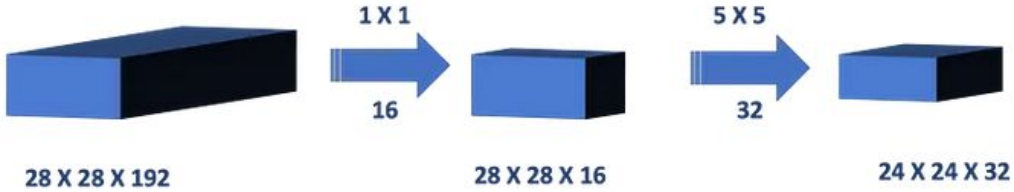
Lecture 10

Koteswar Rao Jerripothula, PhD
Department of Electrical Engineering
IIT Kanpur

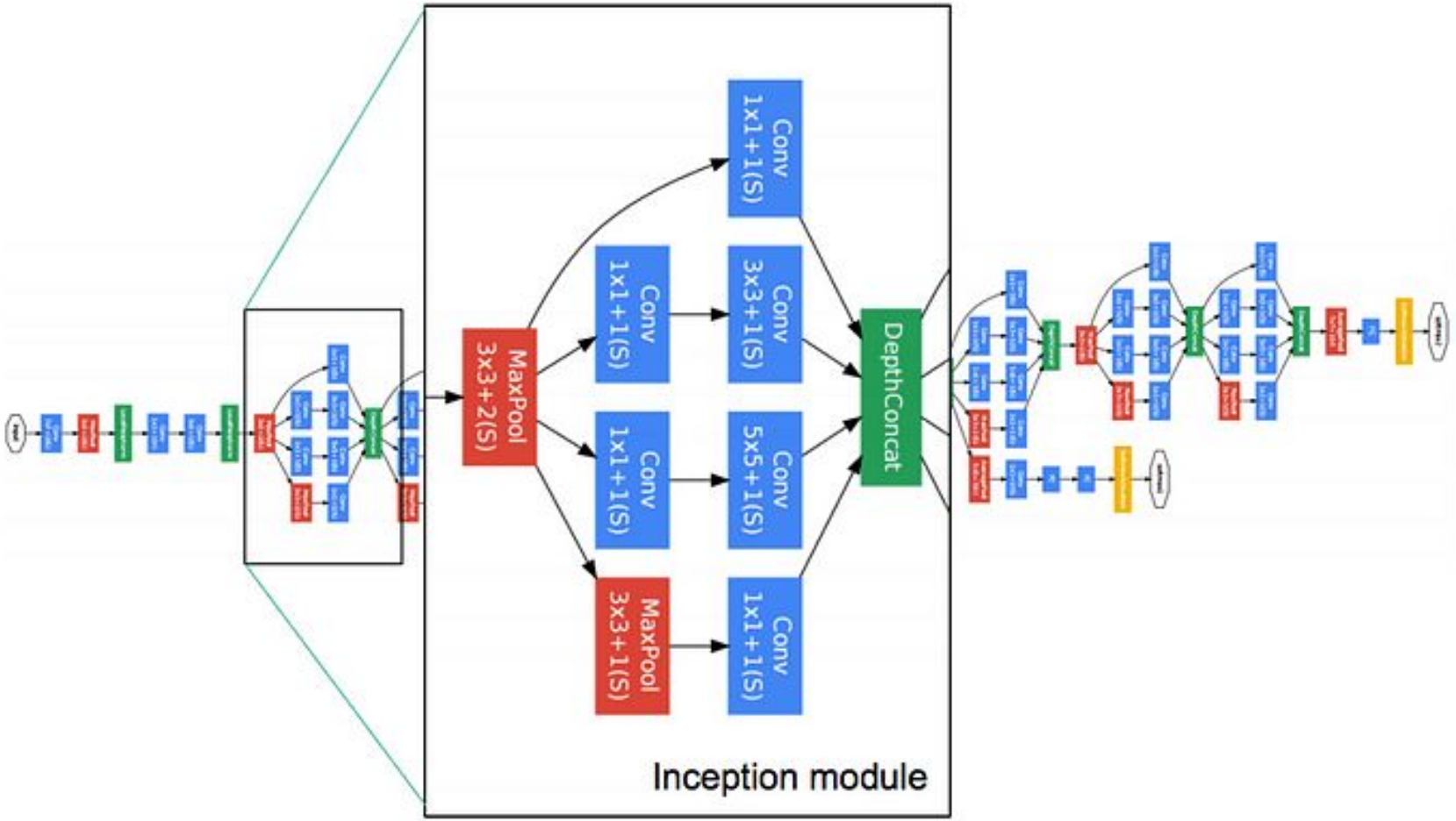
1x1 Convolution

Helps with changing channel-size as we need

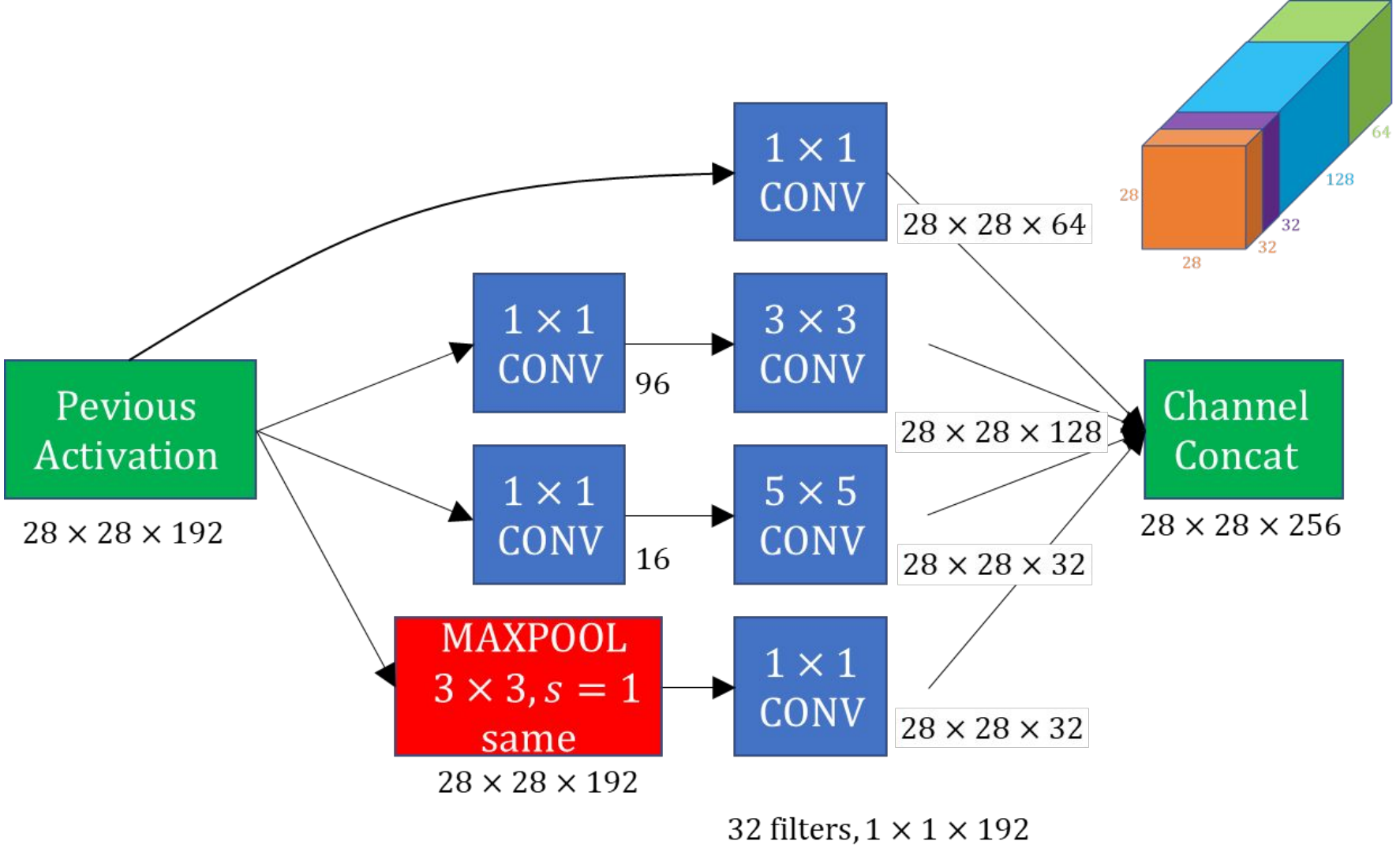
Number of Operations : (28X28X32) X (5X5X192) = 120.422 Million Ops



Number of Operations for 1 X 1 Conv Step : (28X28X16) X (1X1X192) = 2.4 Million Ops
Number of Operations for 5 X 5 Conv Step : (28X28X32) X (5X5X16) = 10 Million Ops
Total Number of Operations = 12.4 Million Ops



GoogleNet Architecture


Pevious Activation
28 × 28 × 192
1 × 1 CONV
1 × 1 CONV
96
3 × 3 CONV
1 × 1 CONV
16
5 × 5 CONV
MAXPOOL 3 × 3, s = 1 same
28 × 28 × 192
1 × 1 CONV
32 filters, 1 × 1 × 192
28 × 28 × 64
28 × 28 × 128
28 × 28 × 32
28 × 28 × 32
Channel Concat
28 × 28 × 256
64
128
32
28
32
28

Semantic Segmentation

To label *each pixel* of an image with a corresponding ***class*** of what is being represented

Input

segmented

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

3	3	3	3	3	3	3	3	3	3	3	3	5	5	5	5	5	5
3	3	3	3	3	3	3	3	3	3	3	3	5	5	5	5	5	5
3	3	3	3	3	3	1	1	3	3	3	3	5	5	5	5	5	5
3	3	3	3	3	1	1	1	1	3	3	3	5	5	5	5	5	5
3	3	3	3	3	3	1	1	3	3	3	5	5	5	5	5	5	5
5	5	3	3	3	3	1	1	3	3	5	5	5	5	5	5	5	5
4	4	3	4	1	1	1	1	1	1	4	4	4	5	5	5	5	5
4	4	3	4	1	1	1	1	1	1	4	4	4	4	4	5	5	5
4	4	4	1	1	1	1	1	1	1	1	4	4	4	4	4	4	4
3	3	3	1	1	1	1	1	1	1	1	4	4	4	4	4	4	4
3	3	3	1	2	2	1	1	1	1	1	4	4	4	4	4	4	4
3	3	3	1	2	2	1	1	1	1	1	4	4	4	4	4	4	4

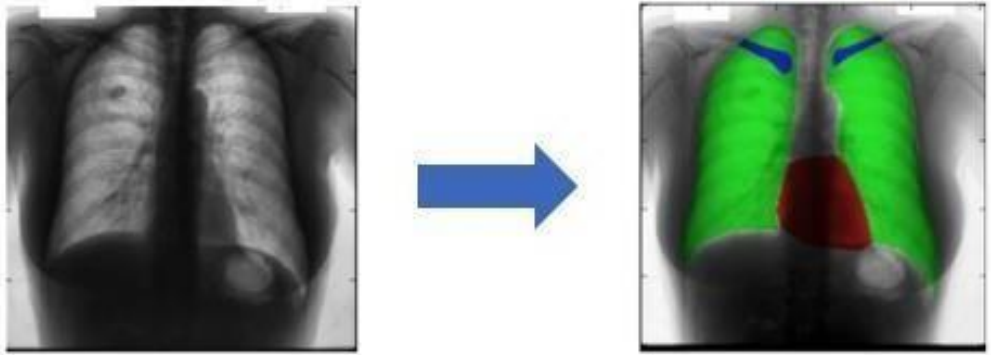
Semantic Labels

predict

Person
Bicycle
Background

Applications: Autonomous Vehicles

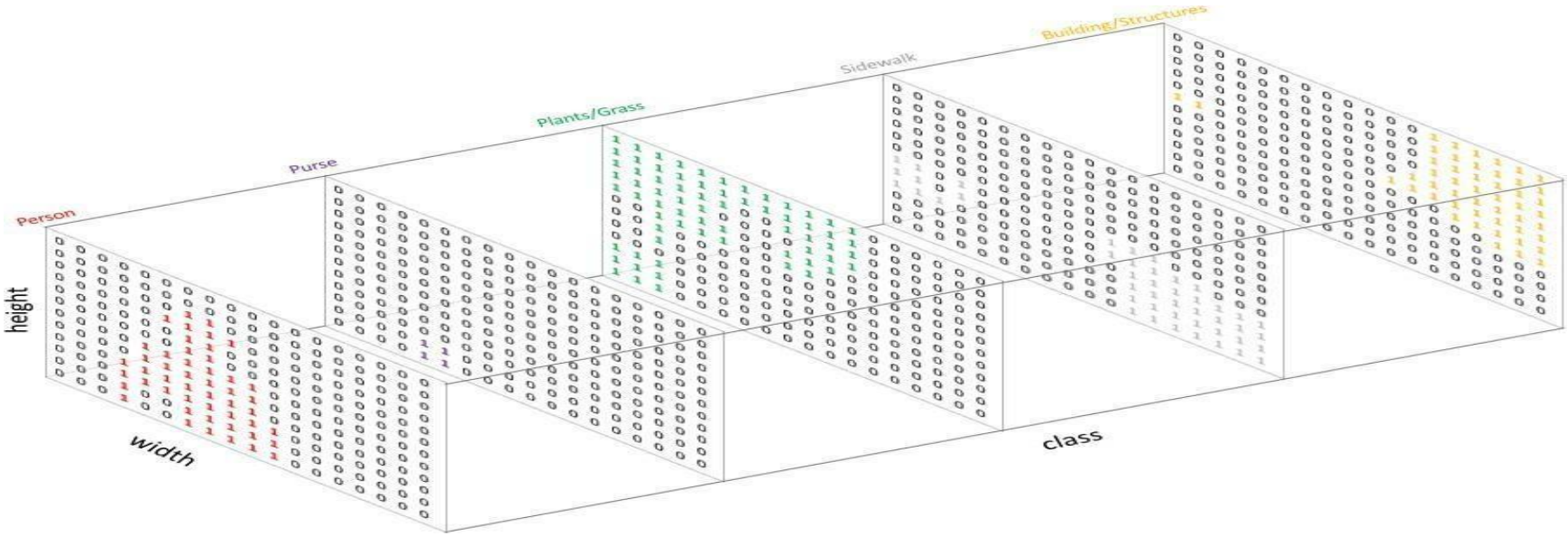
Applications: Medical Diagnosis

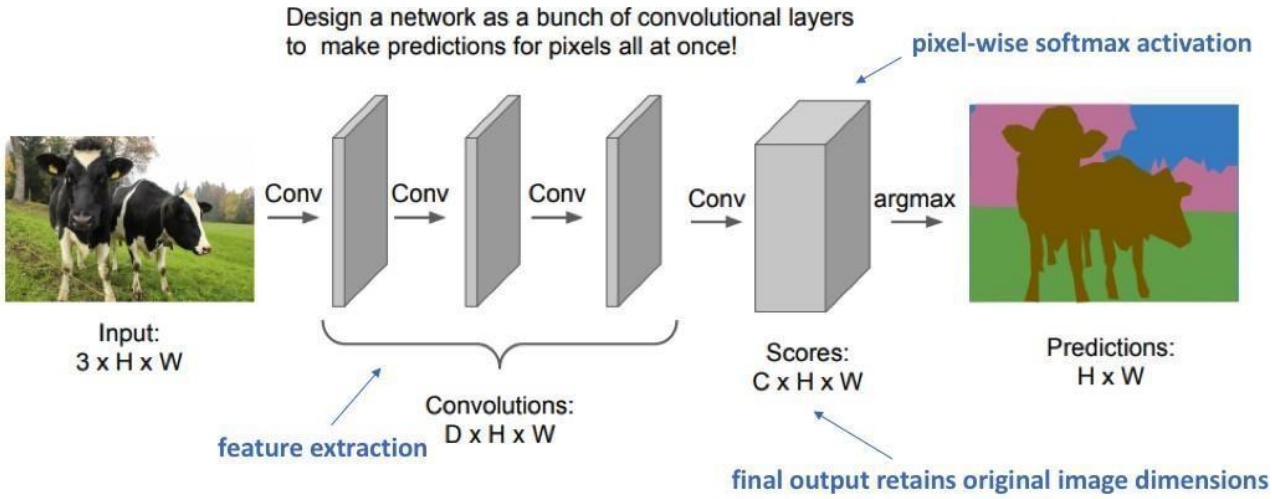


Input Image

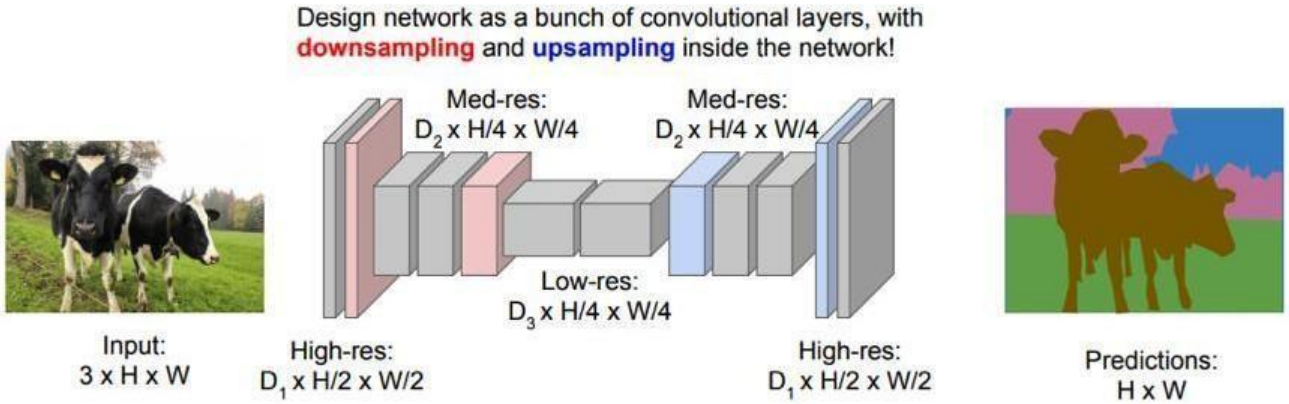
Segmented Image

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



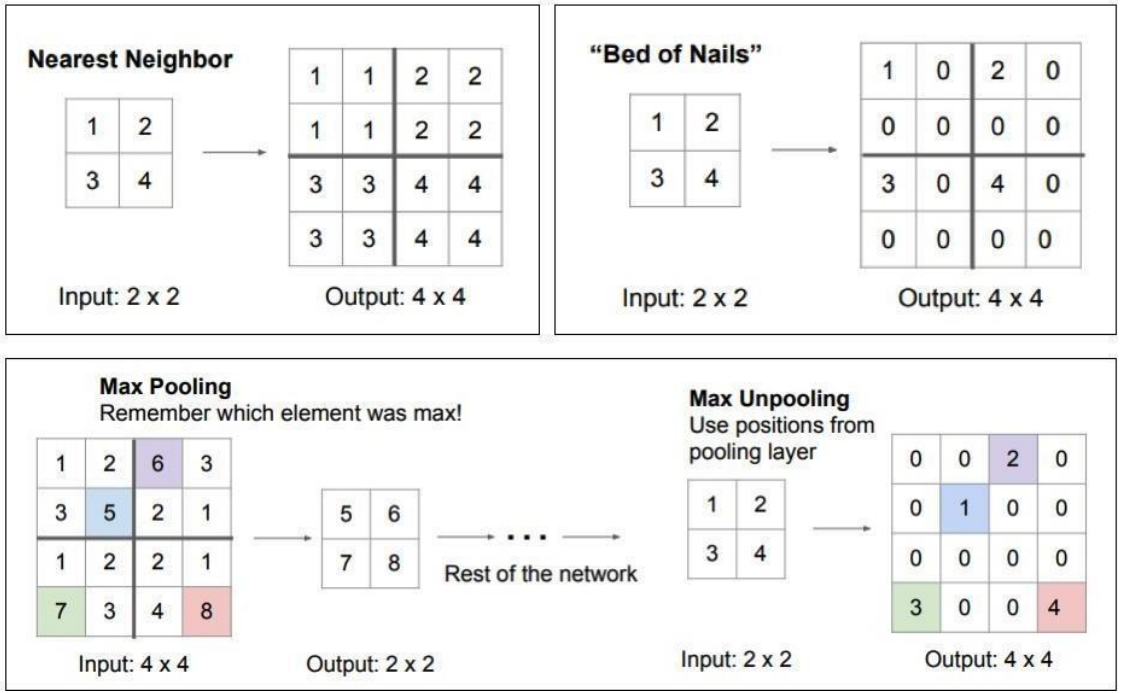


Downside: Preserving image dimensions throughout entire network will be computationally expensive.



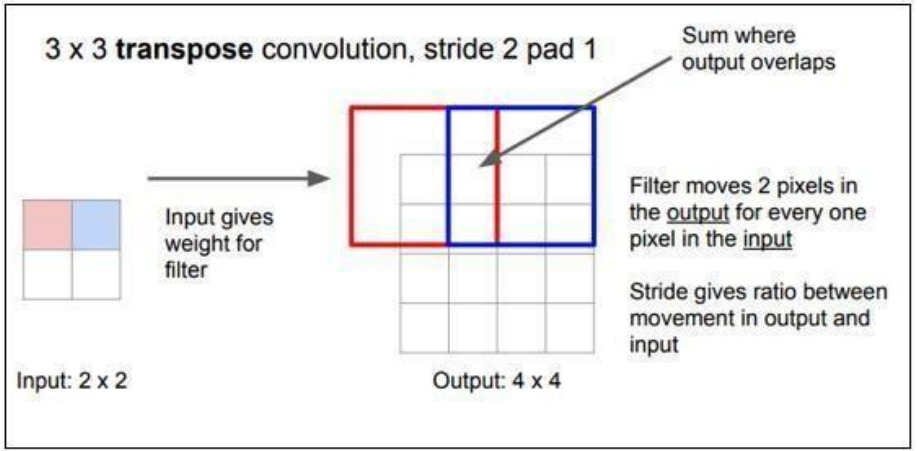
Solution: Make network deep and *work at a lower spatial resolution* for many of the layers.

Upsampling



It's followed by a convolutional layer

Transpose convolutions are by far the most popular approach as they allow for us to develop a learned upsampling



Fully Convolutional Networks for Semantic Segmentation, CVPR'15

Stride 3, Crop 1

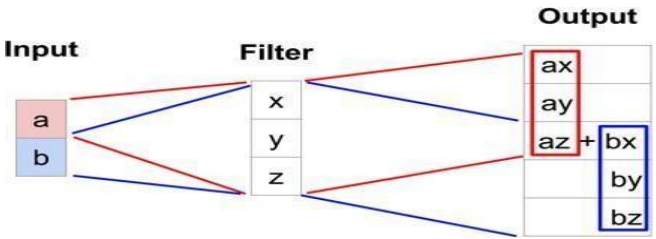
2	3
-2	1

Input

2	1	4
-1	-2	-3
-2	1	5

Conv Mask

-4	-6	-3	-6
2	10	-6	3
-2	-8	2	1
4	6	-1	-2



Stride 2, Crop 0

Steps

- Multiply the transposed convolution filter with every number in the input matrix one-by-one. The multiplication will yield a matrix of same size as the filter. Now we need to arrange these matrices based on the pre-defined stride
- Keep building the output while taking steps as per the pre-defined strides. If there is an overlap at any location, add the values.
- Neglect the boundary values of the output based on the crop parameter.

Input

2	3
-2	1

Filter

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

Stride=3,3
Crop=1

Output

-4	-6	-3	-6
2	8	-6	3
-2	-8	2	1
4	6	-1	-2

4	2	8
-2	-4	-6
-4	2	8

6	3	12
-3	-6	-9
-6	3	12

-4	-2	-8
2	4	6
4	-2	-8

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

Matrices

Input

2	3
-2	1

Filter

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

Stride=2,2

Pad=0

Please solve it

4	2	8
-2	-4	-6
-4	2	8

6	3	12
-3	-6	-9
-6	3	12

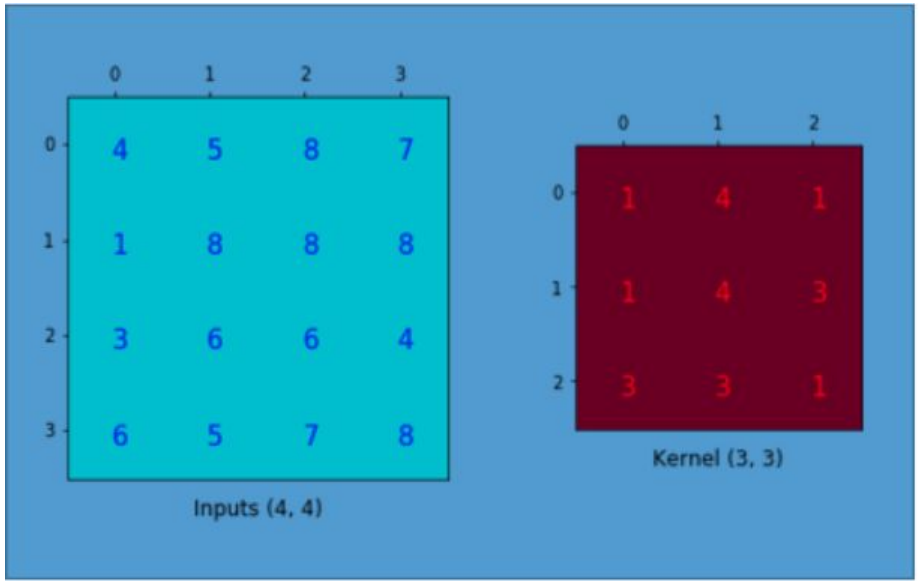
-4	-2	-8
2	4	6
4	-2	-8

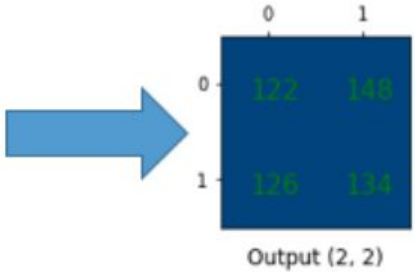
2	1	4
-1	-2	-3
-2	1	4

Why it’s called transposed convolution?

- Let’s see an example

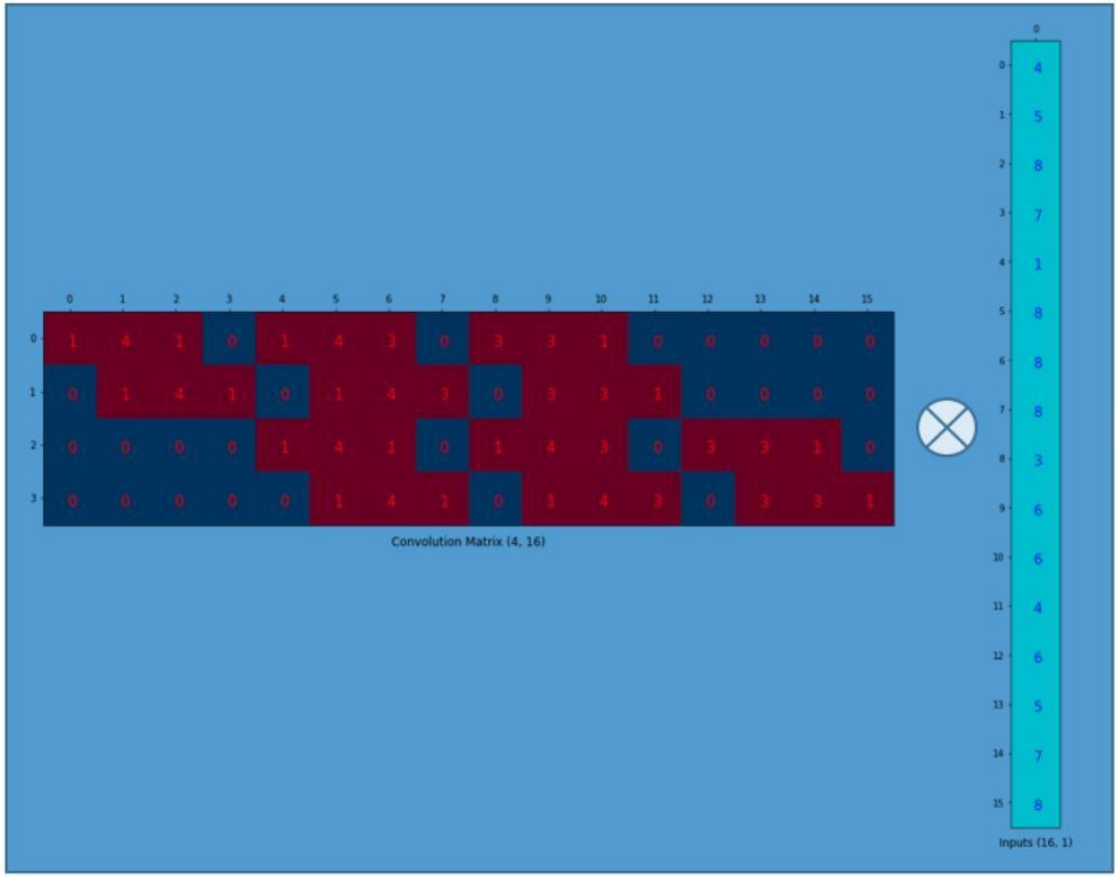
A convolution

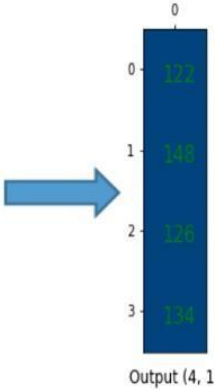




122	148
126	134

We can achieve the same by matrix multiplication as well.

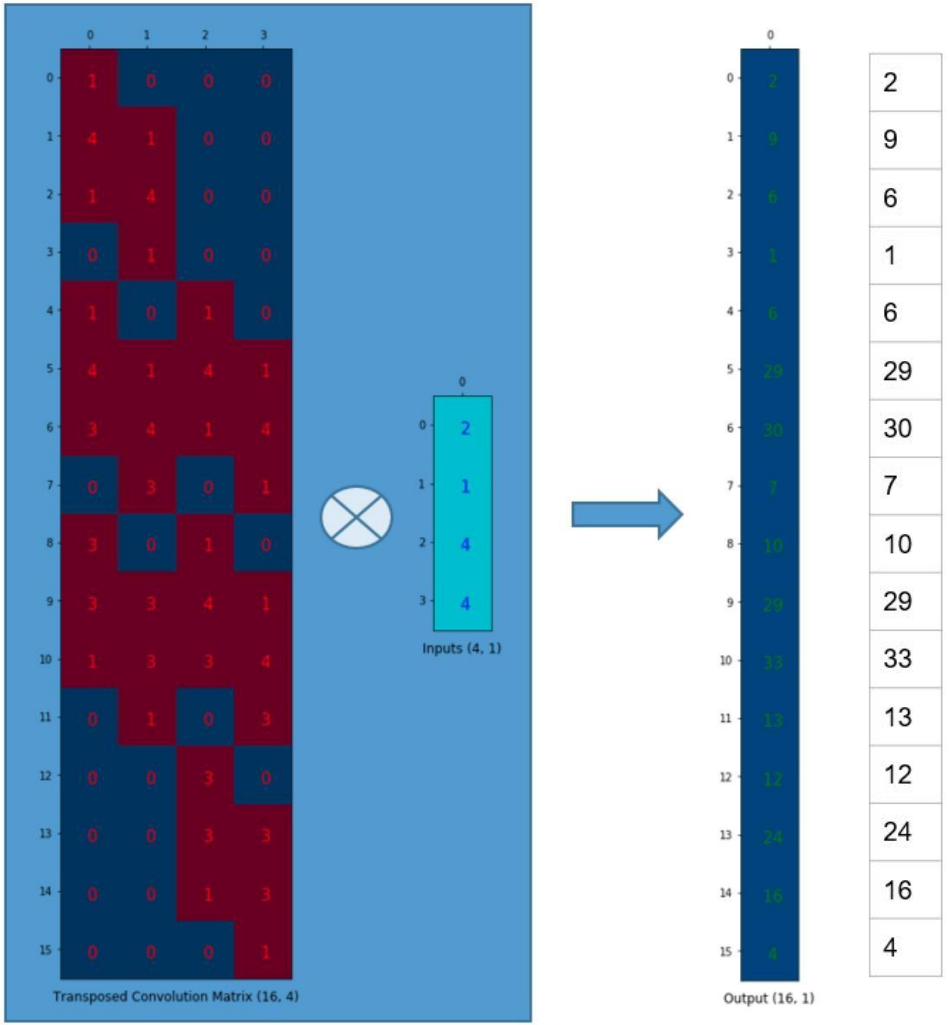


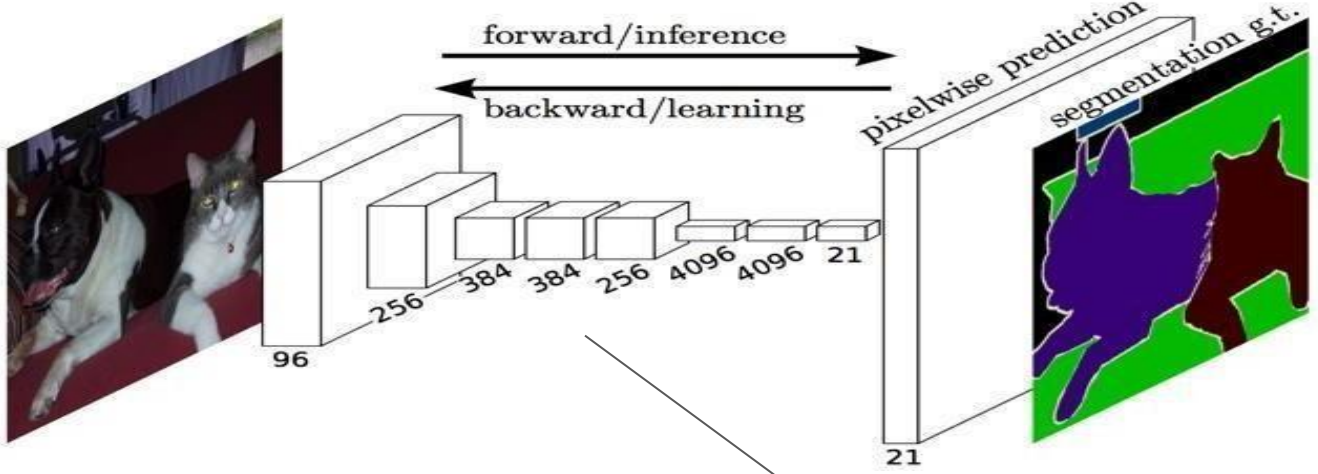


122
148
126
134

- Let’s transpose the filter now

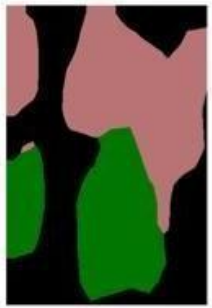
Since we need to transpose the filter matrix to obtain our result, it’s called transposed convolution





Ground truth target

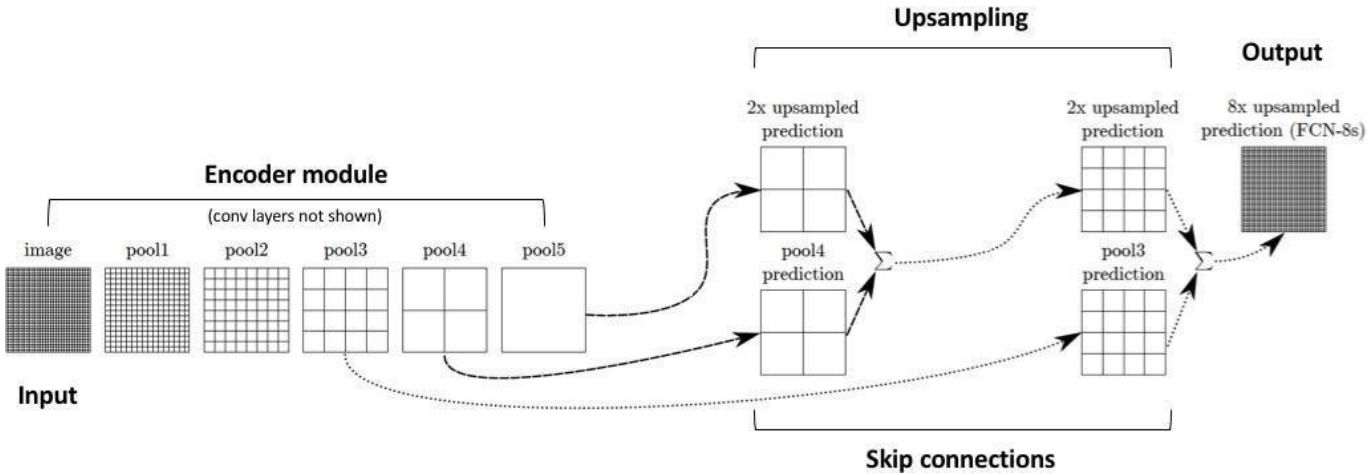
Predicted segmentation



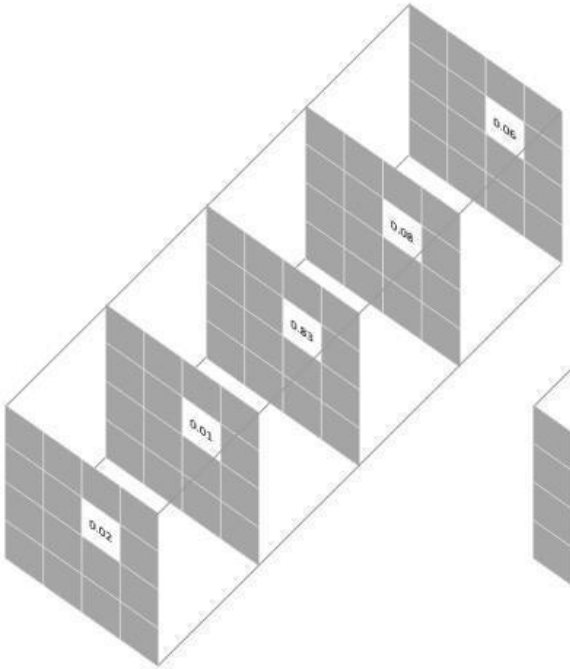
Pre-trained Network

Fully Convolutional Networks for Semantic Segmentation, CVPR'15

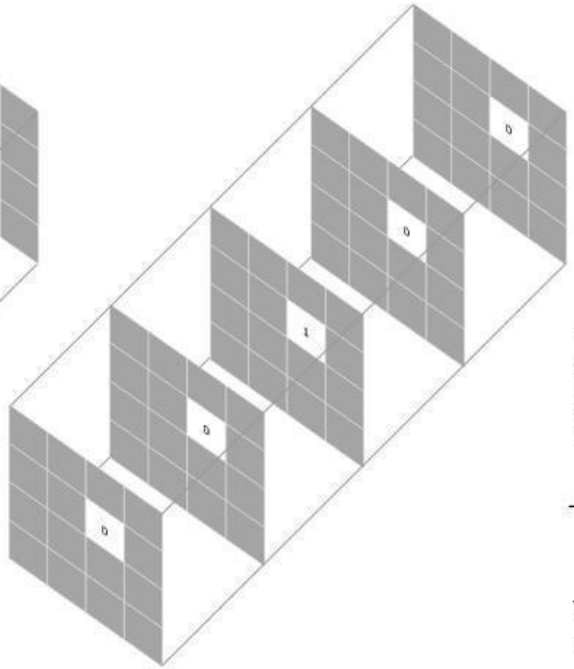
Skip Connections to obtain fine-grained segmentations (reconstruct accurate shapes)



Fully Convolutional Networks for Semantic Segmentation, CVPR'15



Prediction for a selected pixel



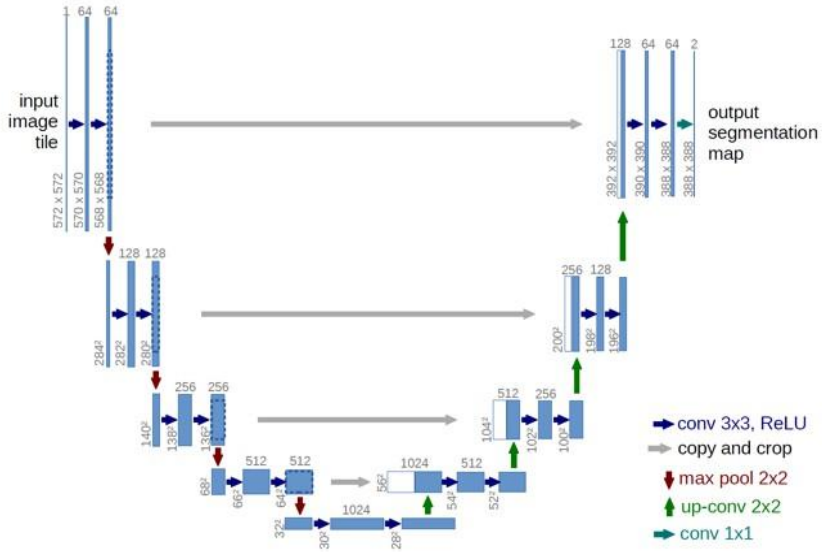
Target for the corresponding pixel

Pixel-wise loss is calculated as the log loss, summed over all possible classes

$$-\sum_{classes} y_{true} \log\left(y_{pred}\right)$$

This scoring is repeated over all **pixels** and averaged

U-Net



Introduces the following:

1) Symmetric structure of the network
2) Concatenation of activation in encoder with activations of decoder
3) Convolution operations in decoder as well

U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI'15

For evaluation

- Round off the output probabilities.
- For every class, compute the jaccard similarity score.
- Compute Average jaccard similarity score for a given image.

Intersection over Union Evaluation Metric (Jaccard Similarity)

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

1	1	0	1
1	1	1	1
0	1	1	1
0	0	0	0

0	0	0	0
0	1	1	0
0	1	0	0
0	0	1	0

0	0	0	0
0	1	1	0
0	1	0	0
0	0	0	0

1	1	0	1
1	1	1	1
0	1	1	1
0	0	1	0

3/11=0.27