Semantic Segmentation

SHALA2020.GITHUB.IO

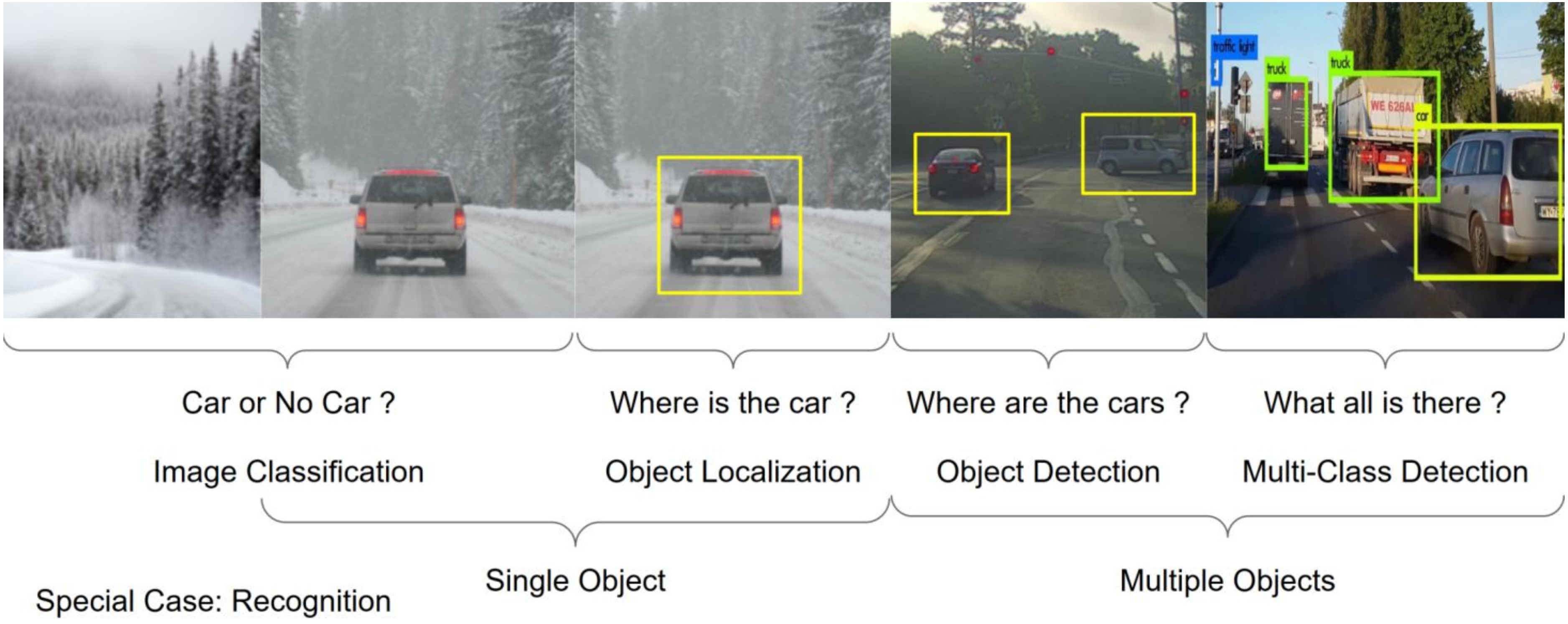
Learning Objectives

- Define the semantic segmentation task
- Describe various types of convolutions
- Design a fully convolutional architecture
- Design a UNet architecture

What can you do with images?

- Classification (Image label)
- Semantic segmentation (pixel-wise label)
- Localization (bounding box)
- Object detection (multiple bounding boxes)
- Instance segmentation (multiple segments)
- Image captioning

Last lecture covered classification (recognition)

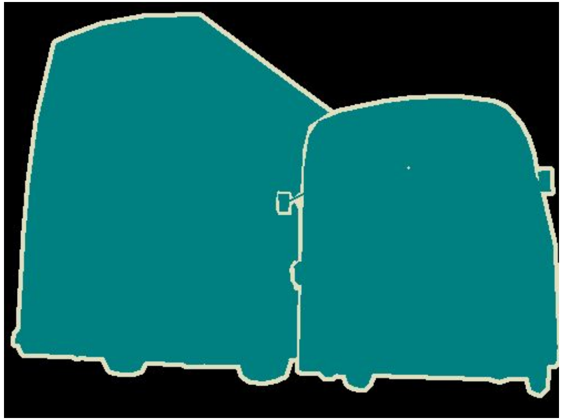


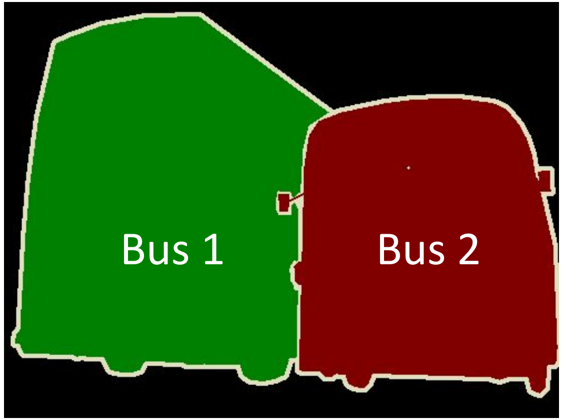
Slide courtesy Sumandeep Banerjee, SHALA2020.GITHUB.IO

Pixel labels for training images must be known to train for **semantic segmentation**

Image Source: "ICNet for Real-Time Semantic Segmentation on High-Resolution Images" Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, Jiaya Jia, ECCV'18

Difference between semantic and instance segmentation





Input Image

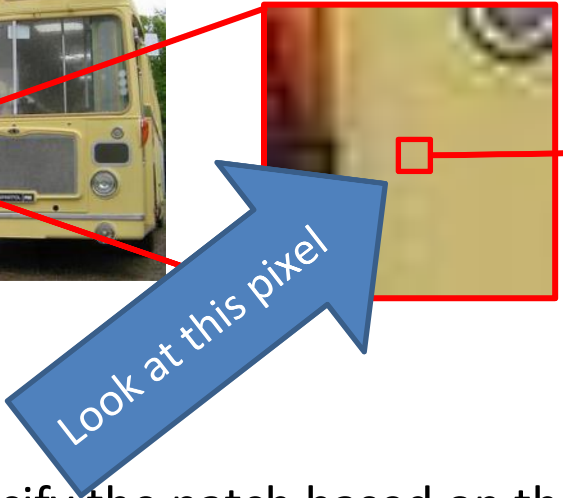
Semantic segmentation
(classify each pixel)
An Image can have multiple classes
e.g. bus, person, background

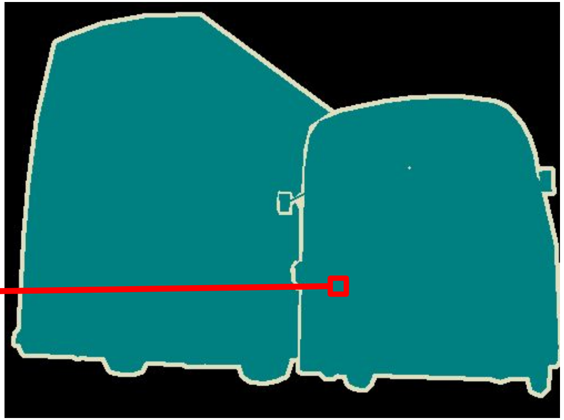
Instance segmentation
(segment each object
(class instance) separately)

Image source: http://host.robots.ox.ac.uk/pascal/VOC/voc2012/segexamples/index.html

Pixel and its spatial context

Input Image

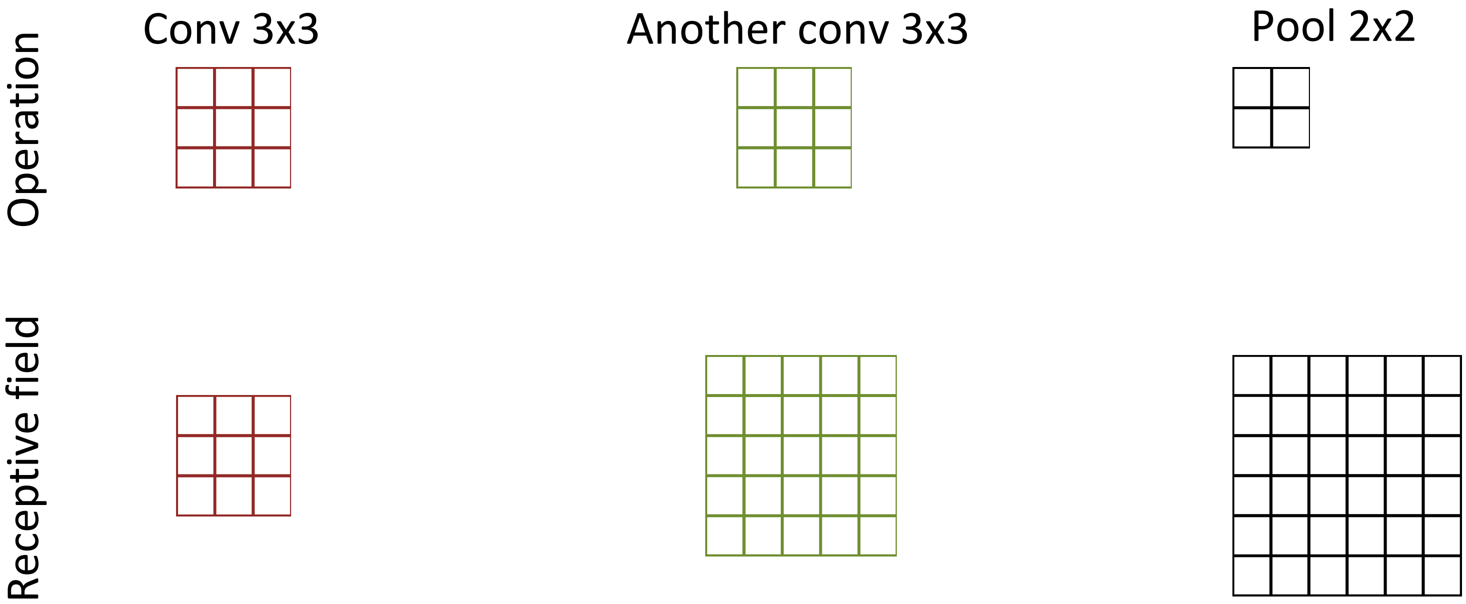




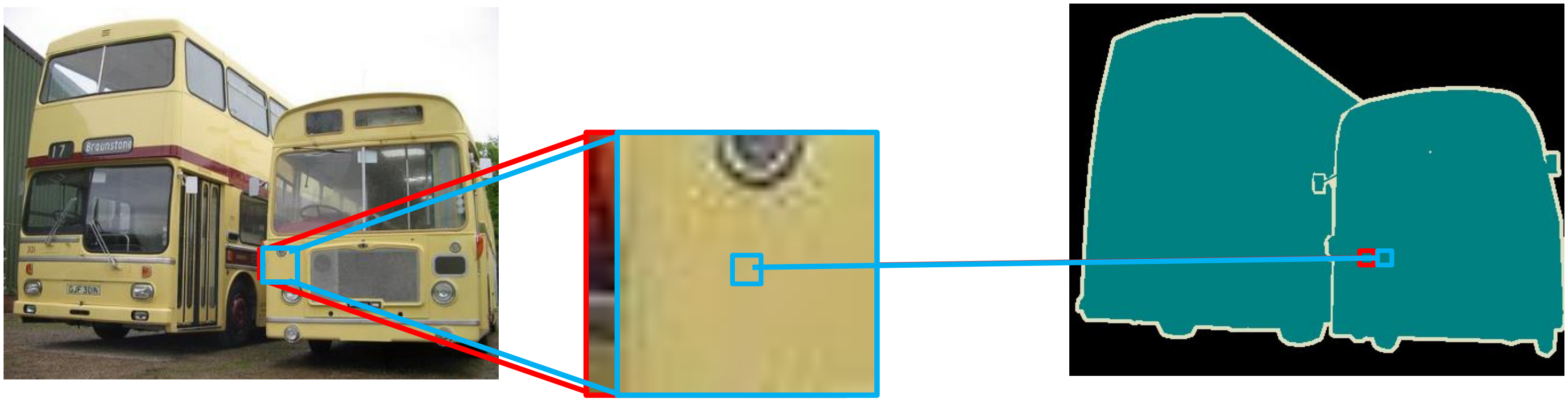
- Can we classify the patch based on the class of the central pixel?
- How big should this patch be for good accuracy?

Image source: http://host.robots.ox.ac.uk/pascal/VOC/voc2012/segexamples/index.html

Receptive fields



Naïve approach – classify patch based on the label of the central pixel, then slide one pixel over

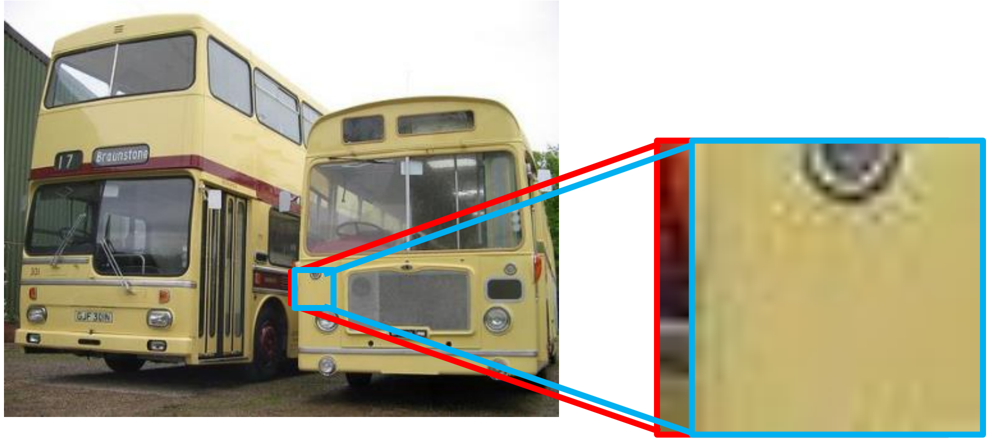


Input Image

- This is not very efficient!

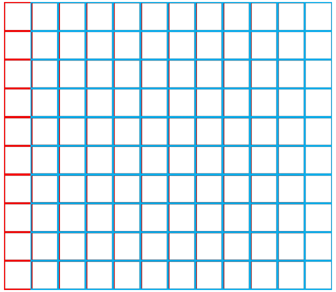
Image source: http://host.robots.ox.ac.uk/pascal/VOC/voc2012/segexamples/index.html

Naïve approach – classify patch based on the label of the central pixel, then slide one pixel over



Input Image

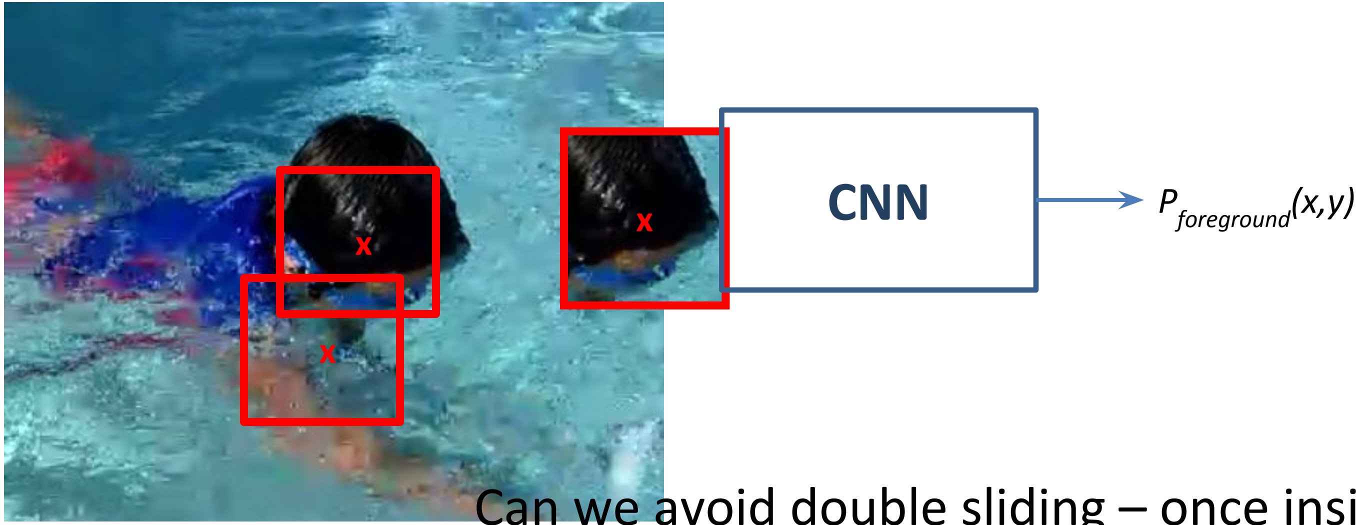
w_{11}	w_{12}	w_{13}
w_{21}	w_{22}	w_{23}
w_{31}	w_{32}	w_{33}



Operations are being repeated without change in inputs or weights

Image source: http://host.robots.ox.ac.uk/pascal/VOC/voc2012/segexamples/index.html

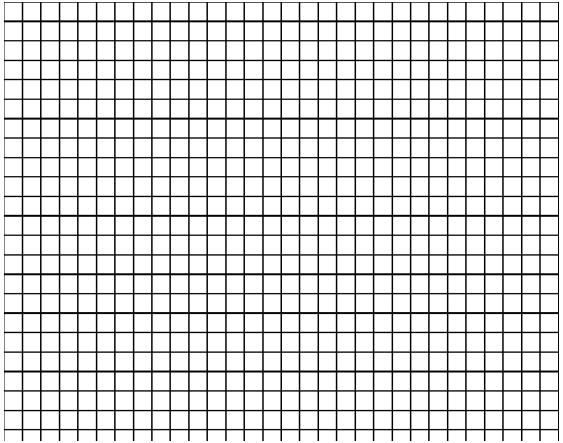
For segmentation, a pixel class can be predicted using some spatial context



Can we avoid double sliding – once inside the conv layers, and once of the CNN itself?

Fully convolutional network resolves the problem of double convolution

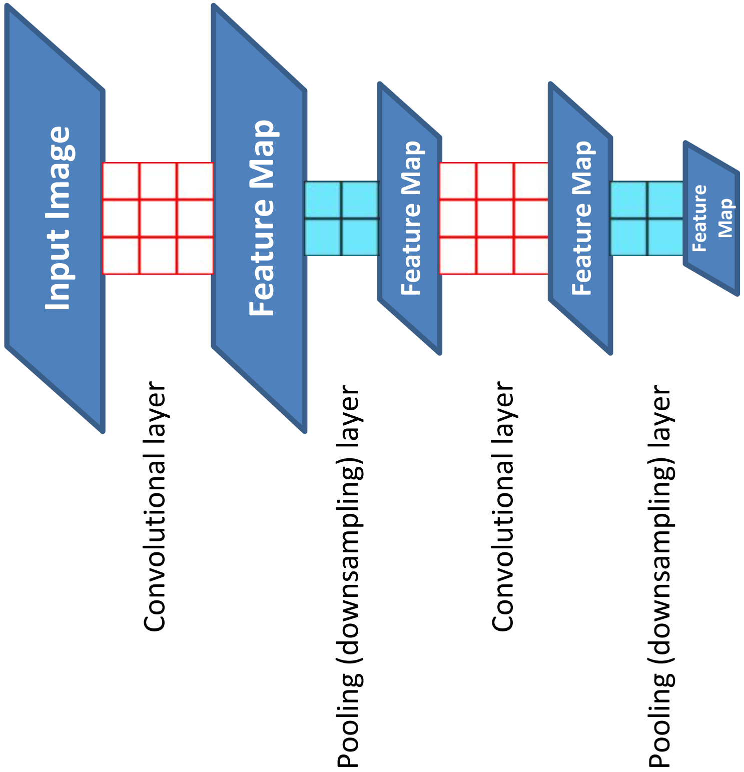
w_{11}	w_{12}	w_{13}
w_{21}	w_{22}	w_{23}
w_{31}	w_{32}	w_{33}



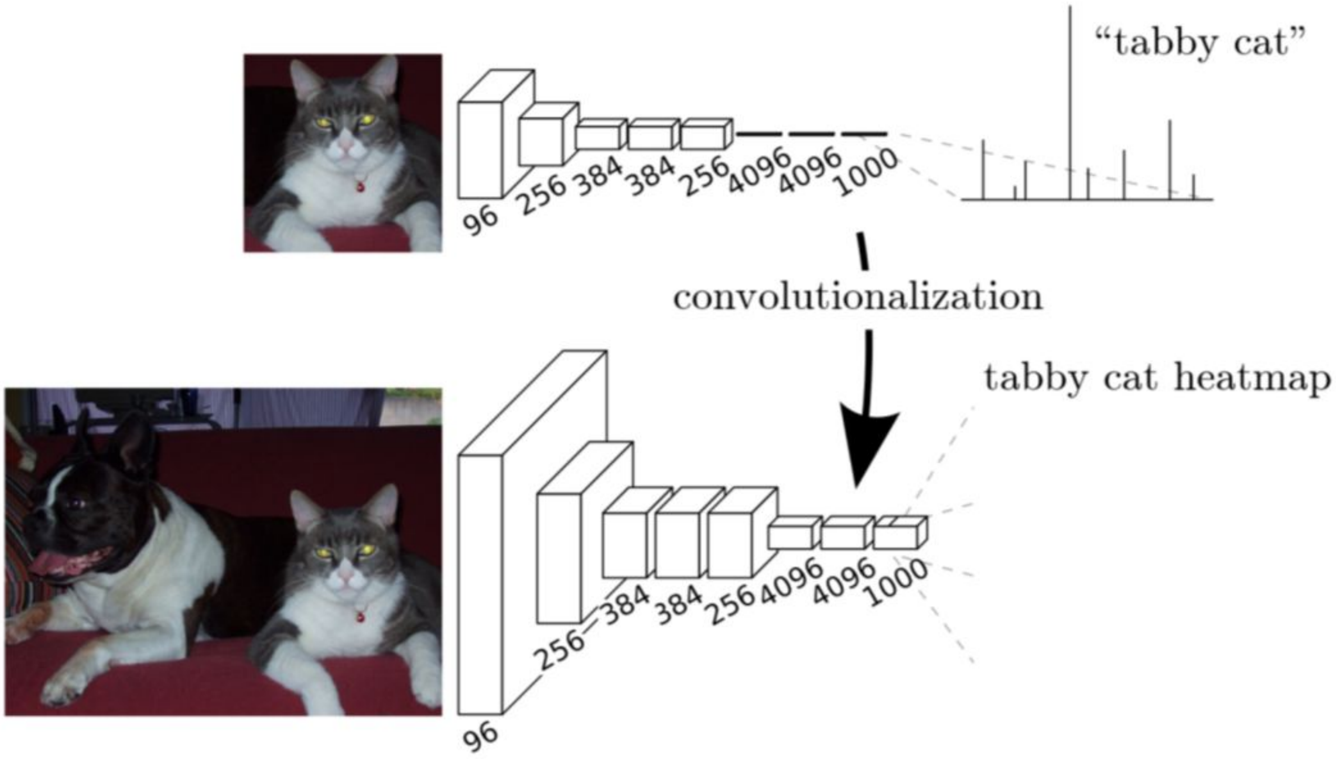
Input Image

Image source: http://host.robots.ox.ac.uk/pascal/VOC/voc2012/segexamples/index.html

1. Use only convolutional and pooling layer to compute feature-maps and “heat-maps”

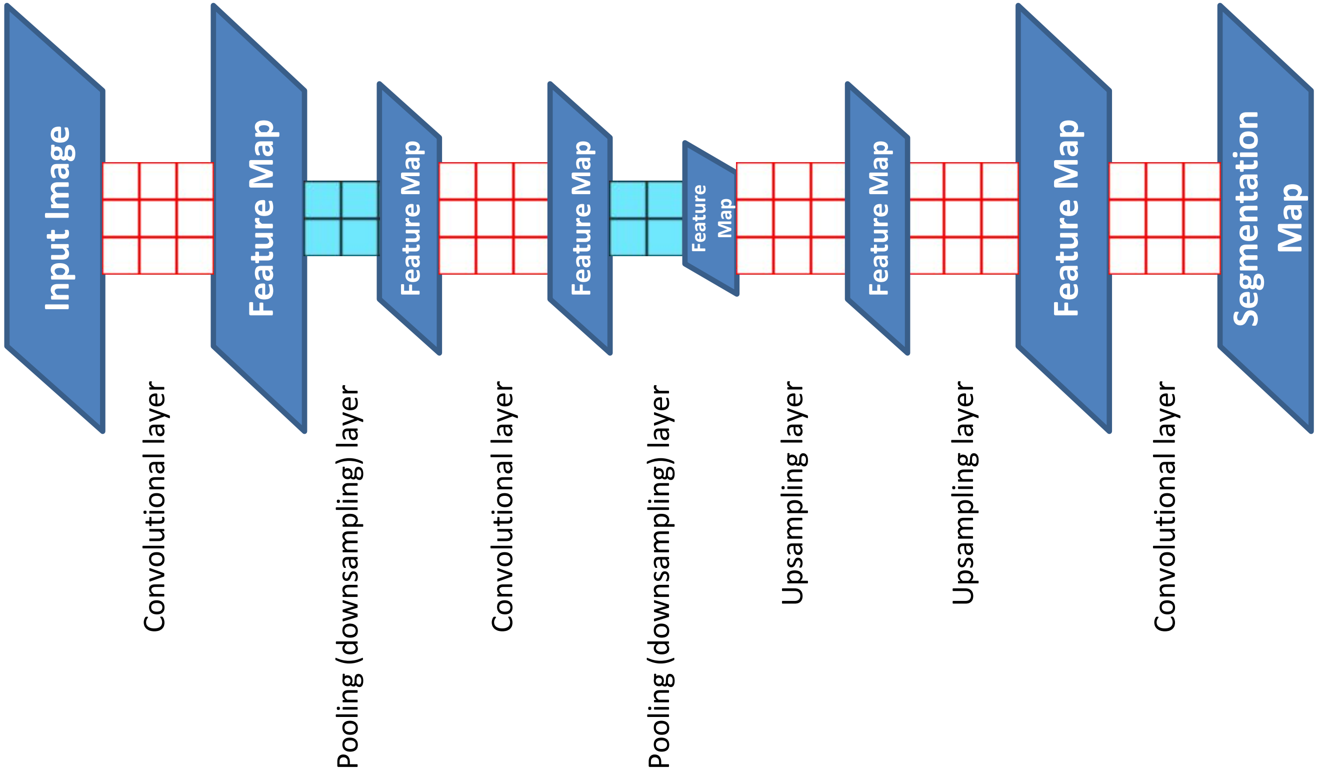


1 (contd.) "Convolutionalize" fully-connected layer to get heatmaps (of smaller size)

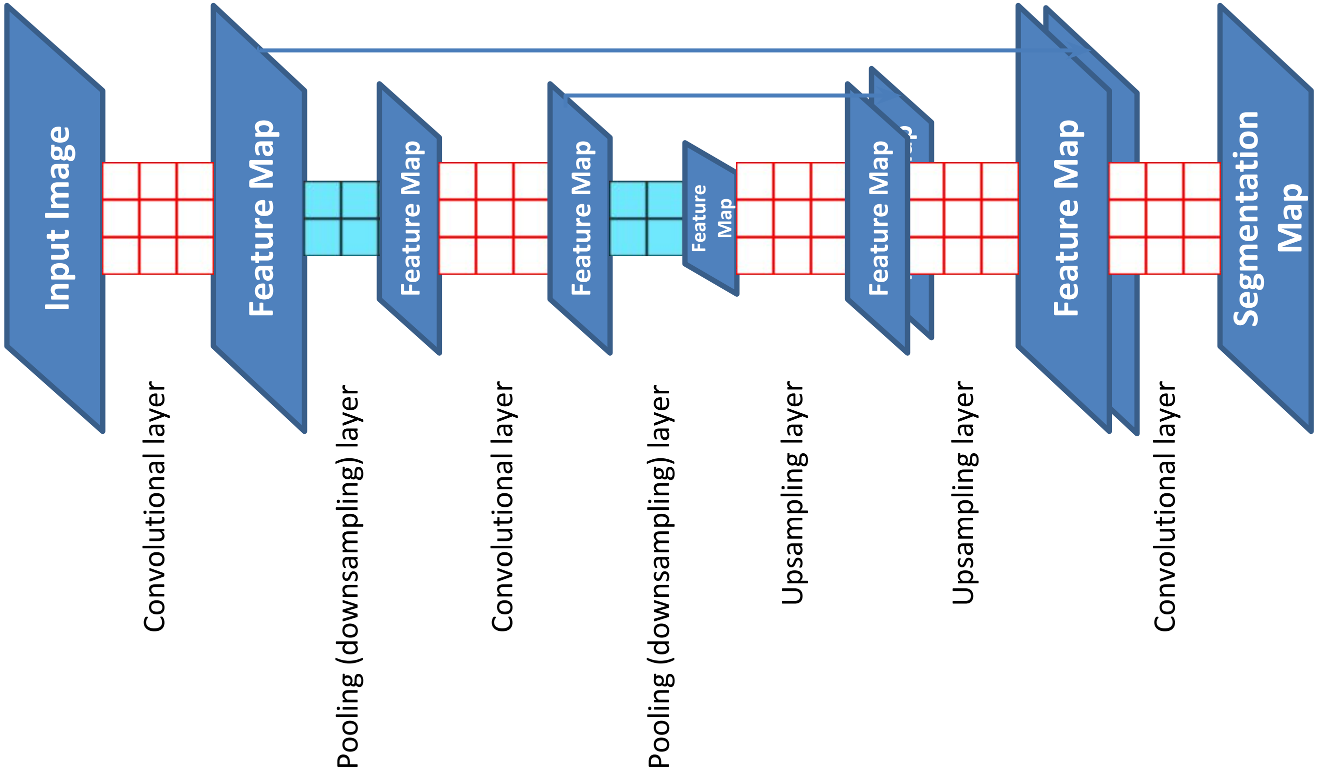


Source: "Fully Convolutional Networks for Semantic Segmentation" by Jonathan Long, Evan Shelhamer, Trevor Darrell

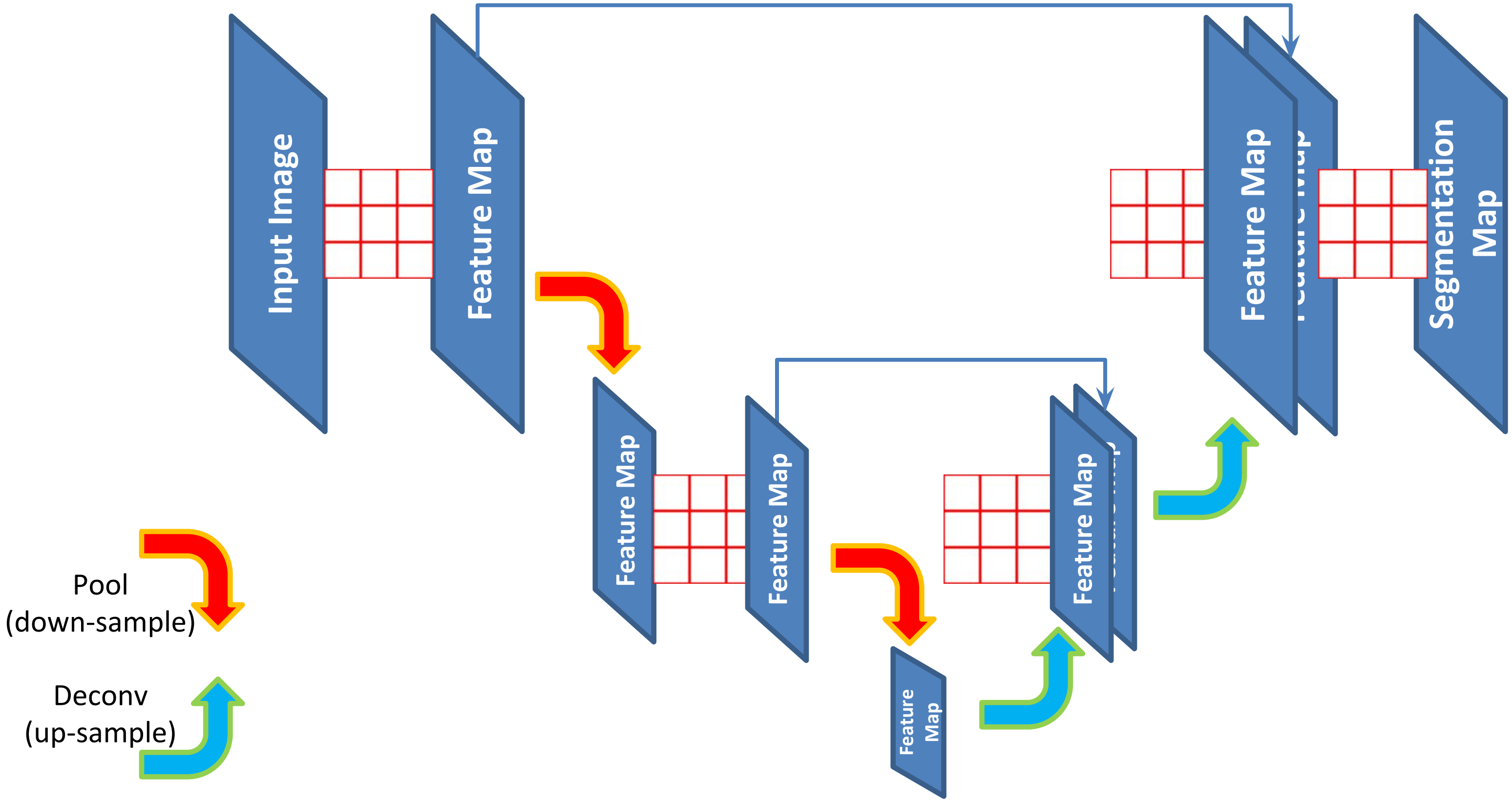
2. Up-sample (deconv) to increase size



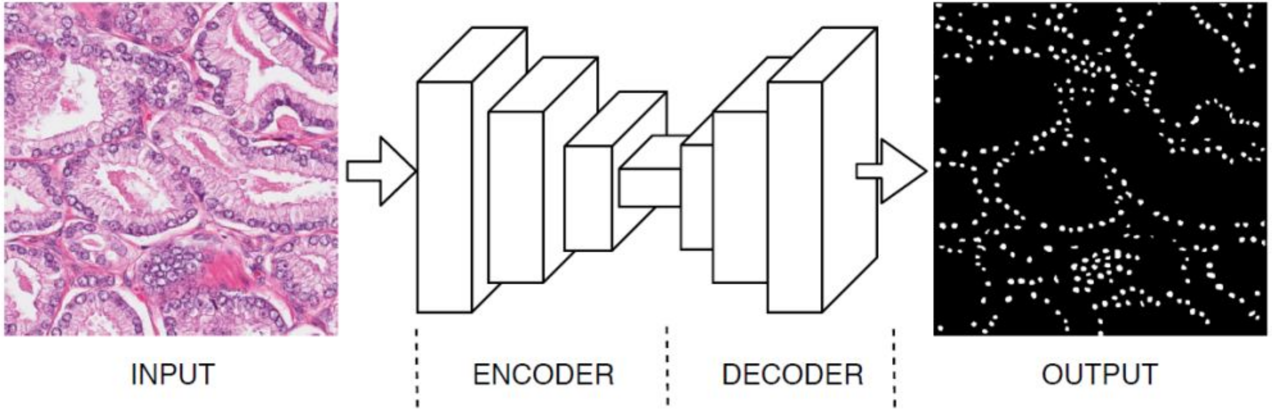
3. Use multi-scale feature maps using "skip connections"



A different way to draw the same network



The general idea is to have an encoder and a decoder



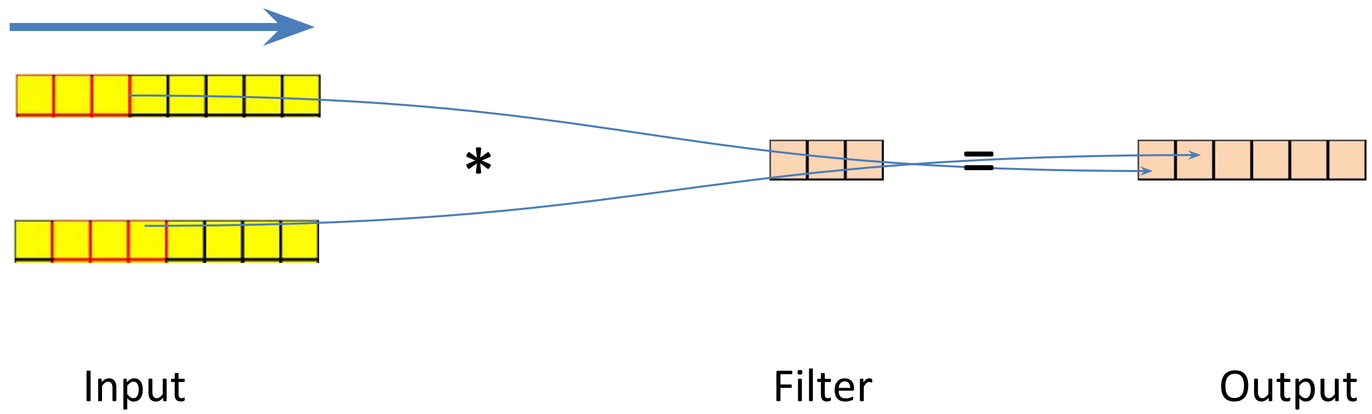
Encoder collects spatial context from a large receptive field

Decoder refines the features from coarse to fine scale

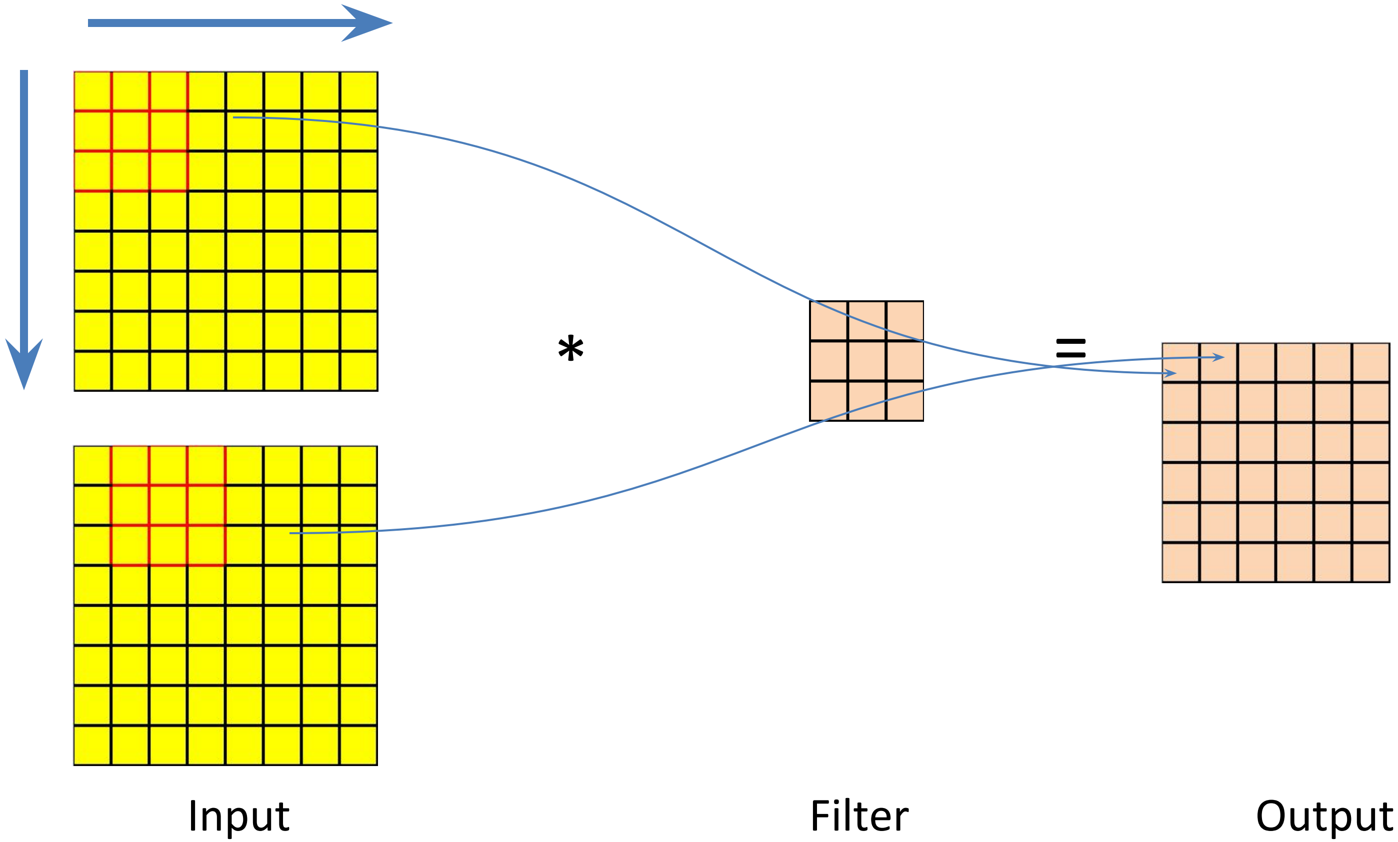
Different types of convolution

- NxN: valid, same
- Stride > 1
- 1x1
- Nx1 and 1xN
- Atrous / dilated
- Upconv / upsampling / deconv / transposed convolution

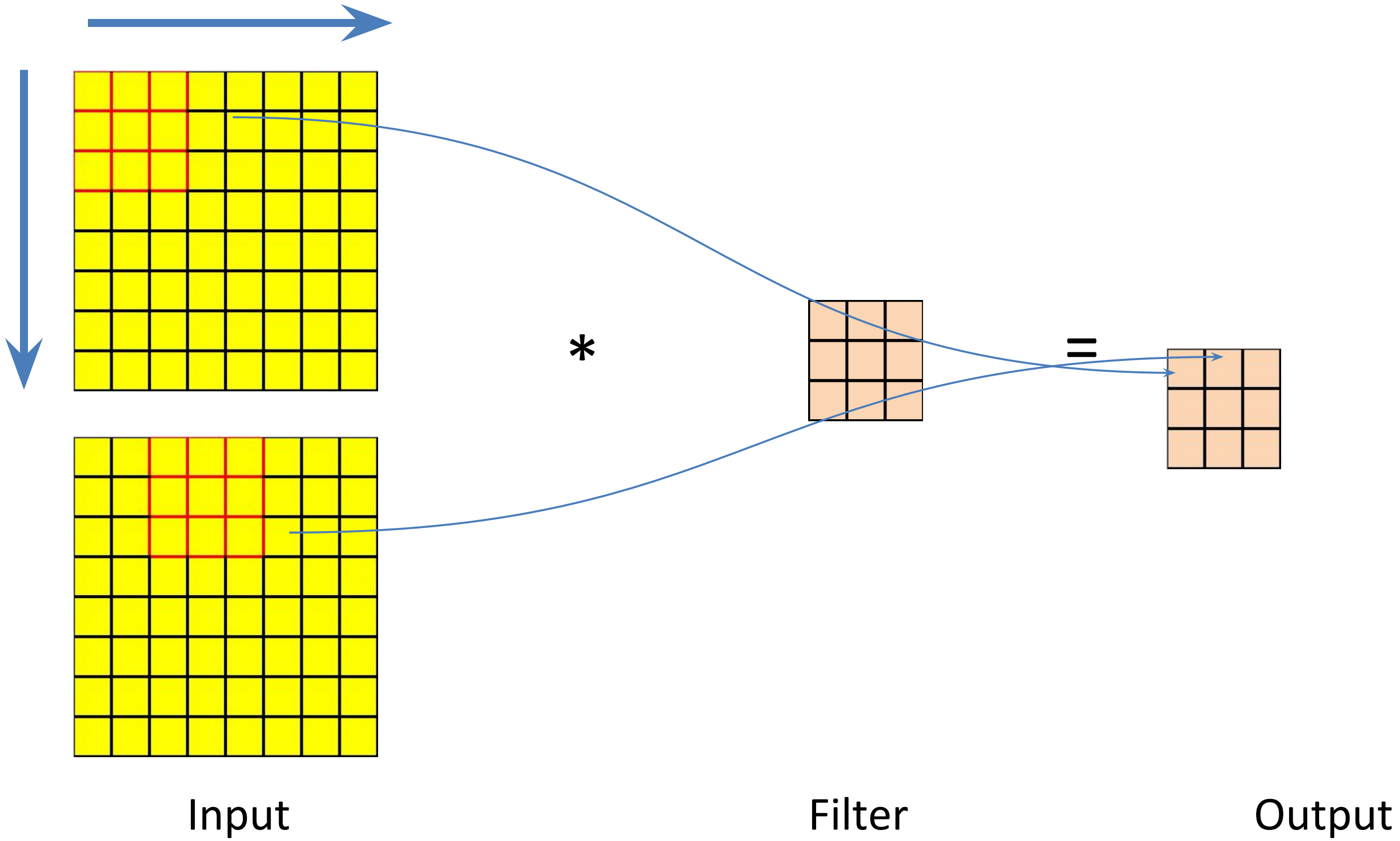
1-D convolution on 1 channel with stride 1



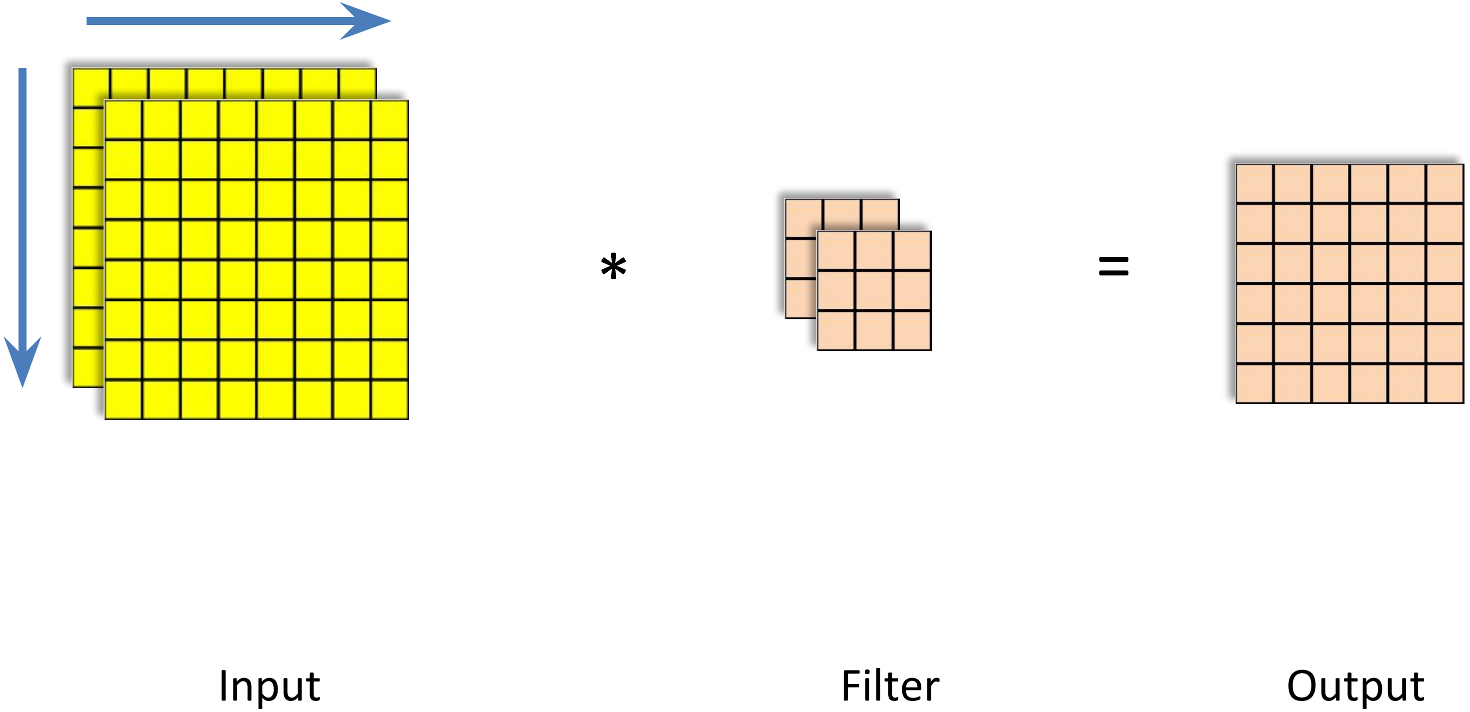
2-D convolution on 1 channel with stride 1



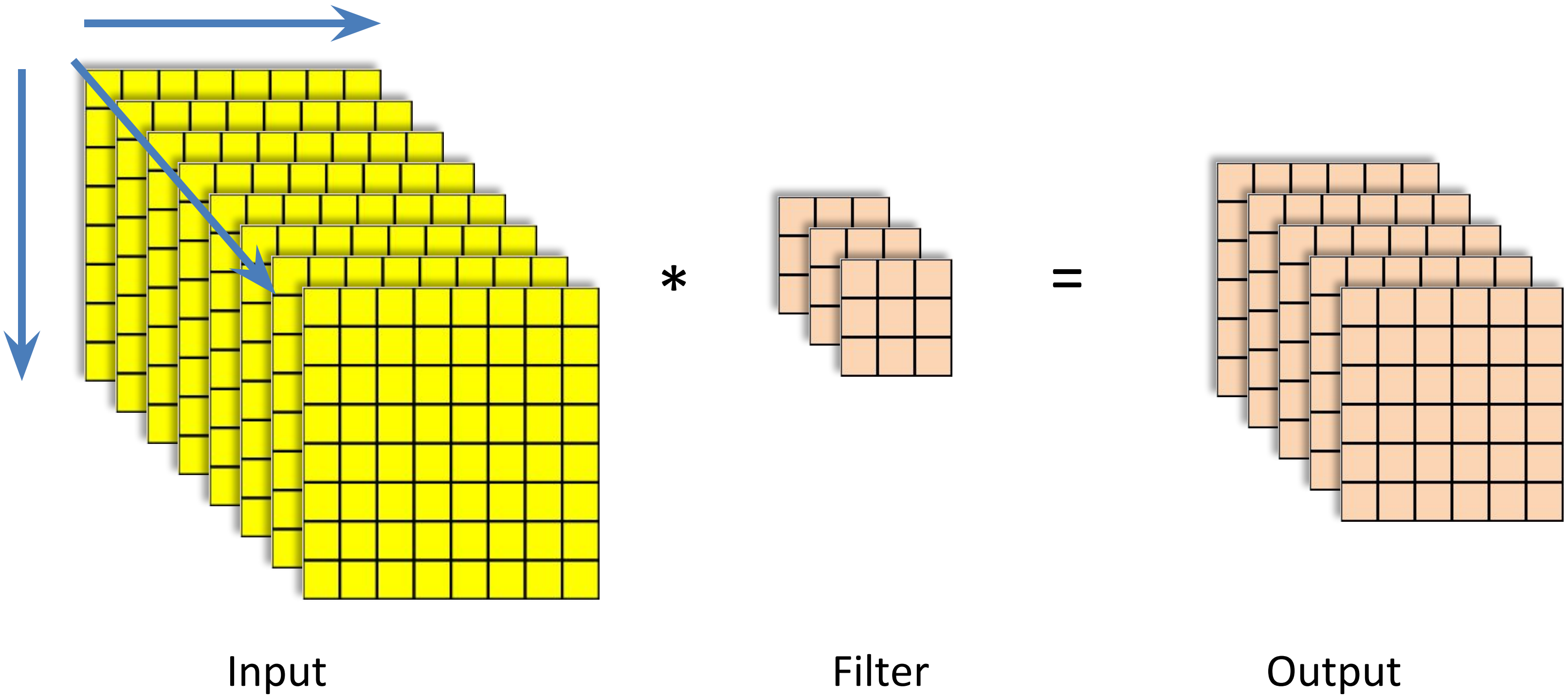
2-D convolution on 1 channel with stride 2



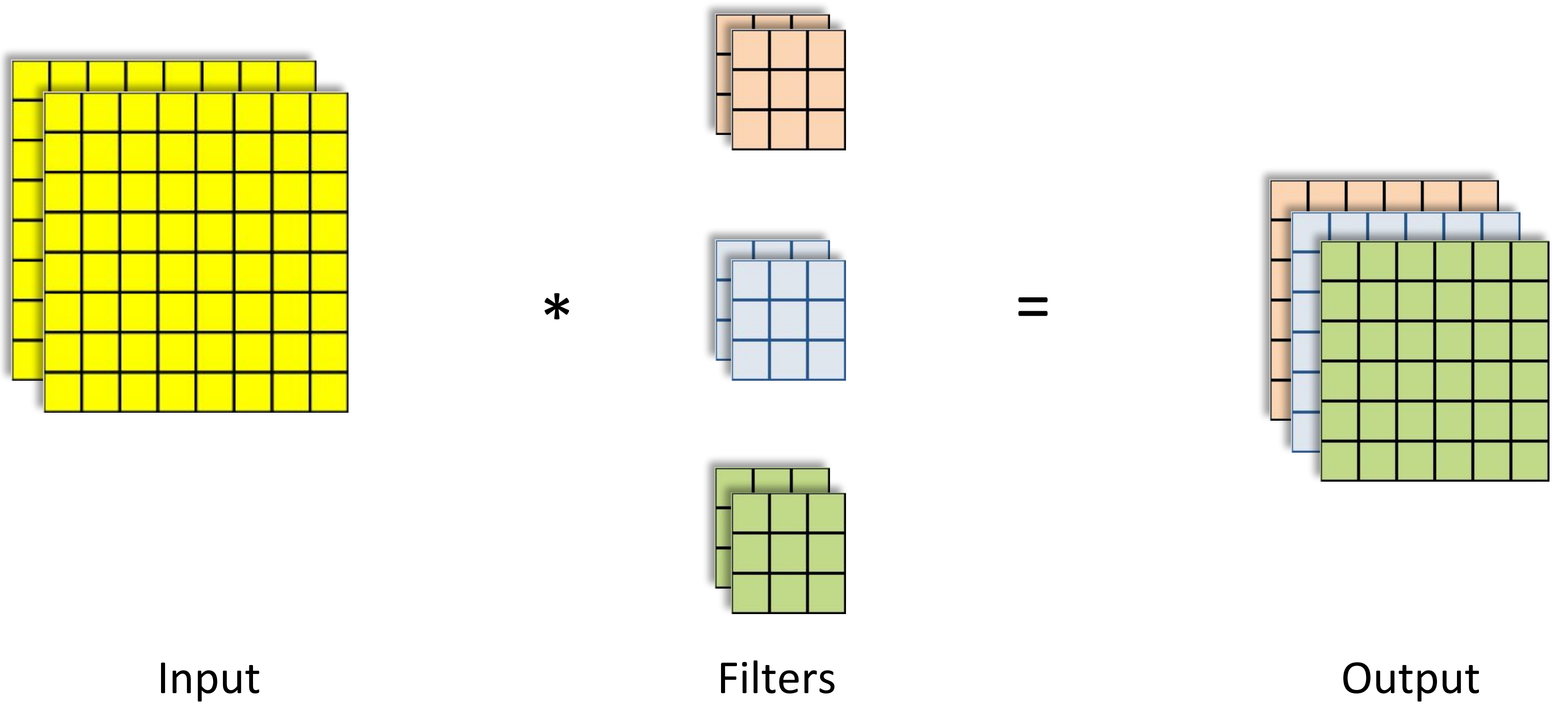
2-d convolution on 2 channels



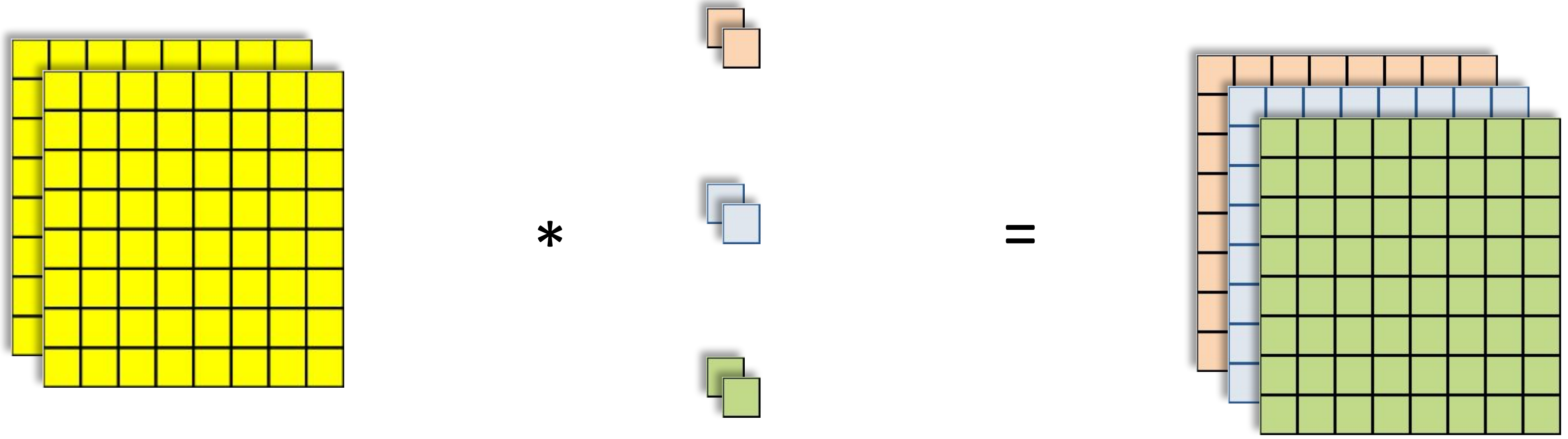
3-d convolution on 1 channel



2-d convolution on 2 channels with 3 filters

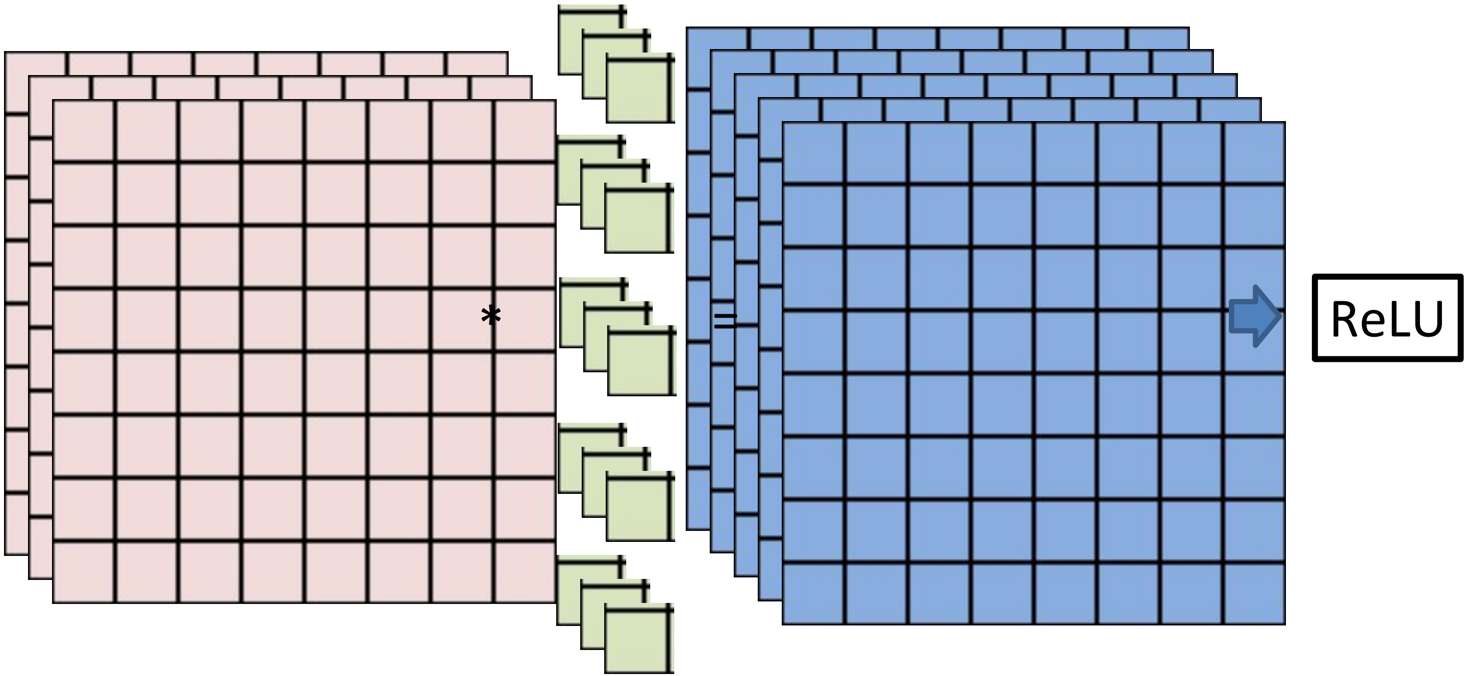


2-d convolution on 2 channels with 3 1x1 filters



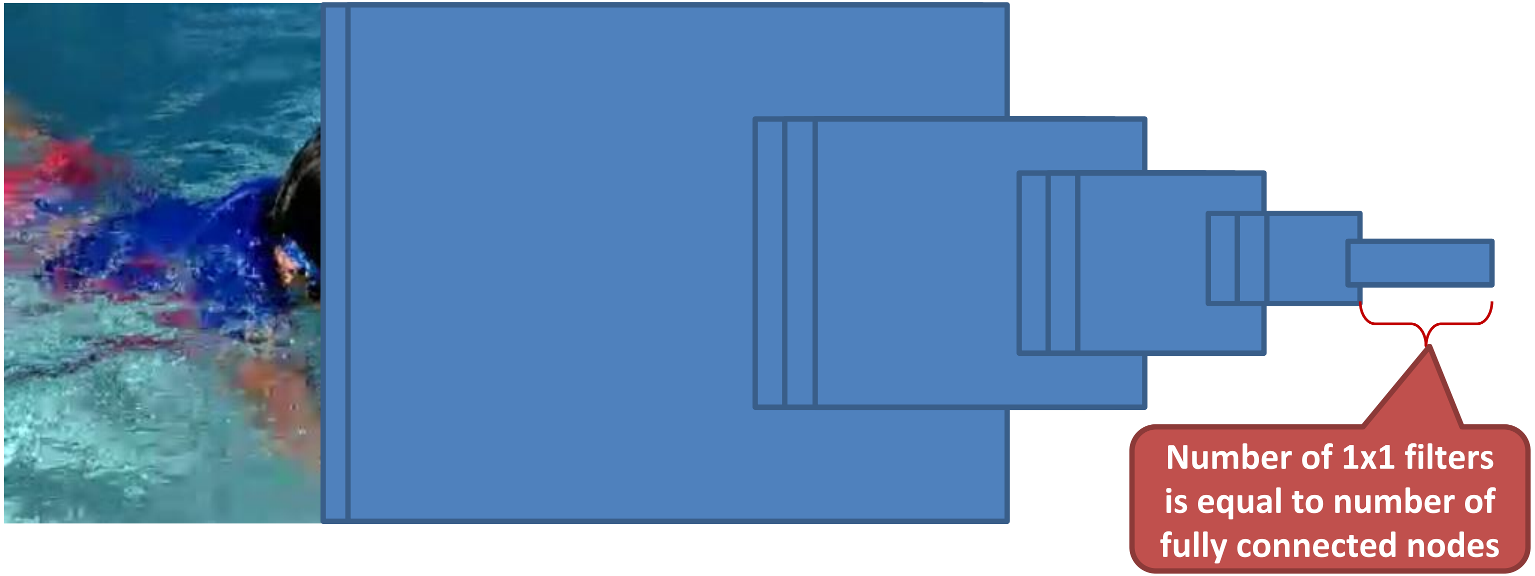
Input Filters Output

1x1 convolutions can also be used to change the number of feature maps

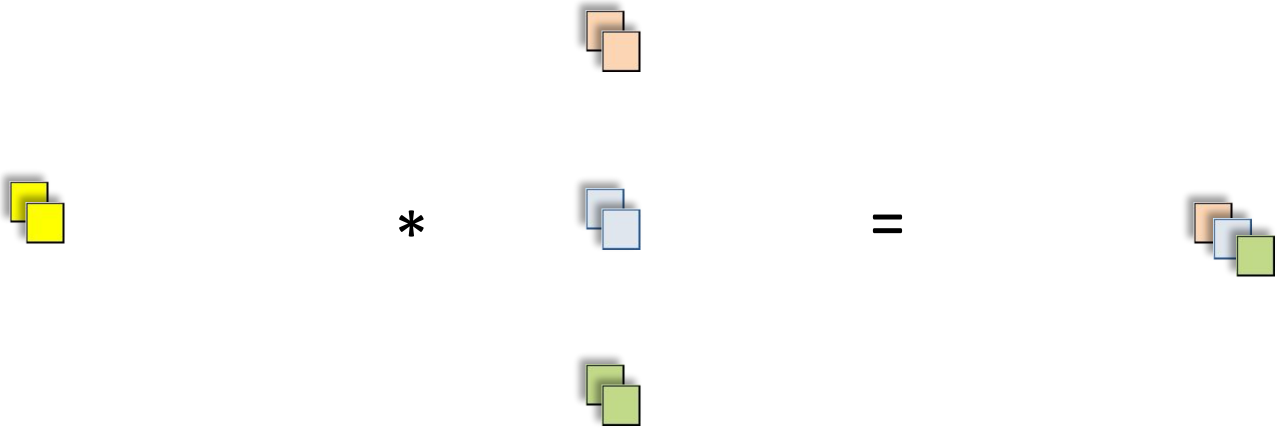


Using 1x1 convolutions is equivalent to having a fully connected layer

- This way, a fully convolutional network can be constructed from a regular CNN such as VGG11

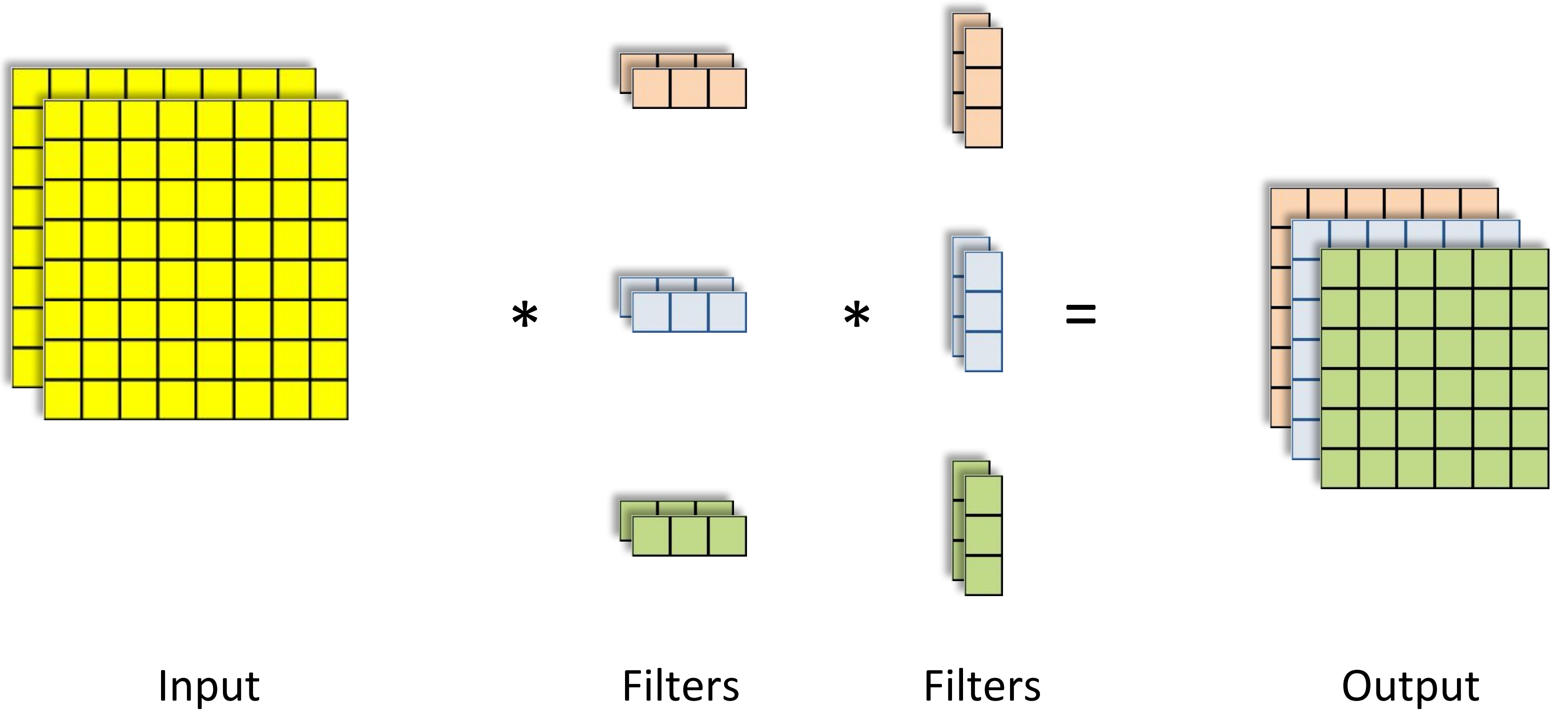


2-d convolution on 2 channels with 3 1x1 filters

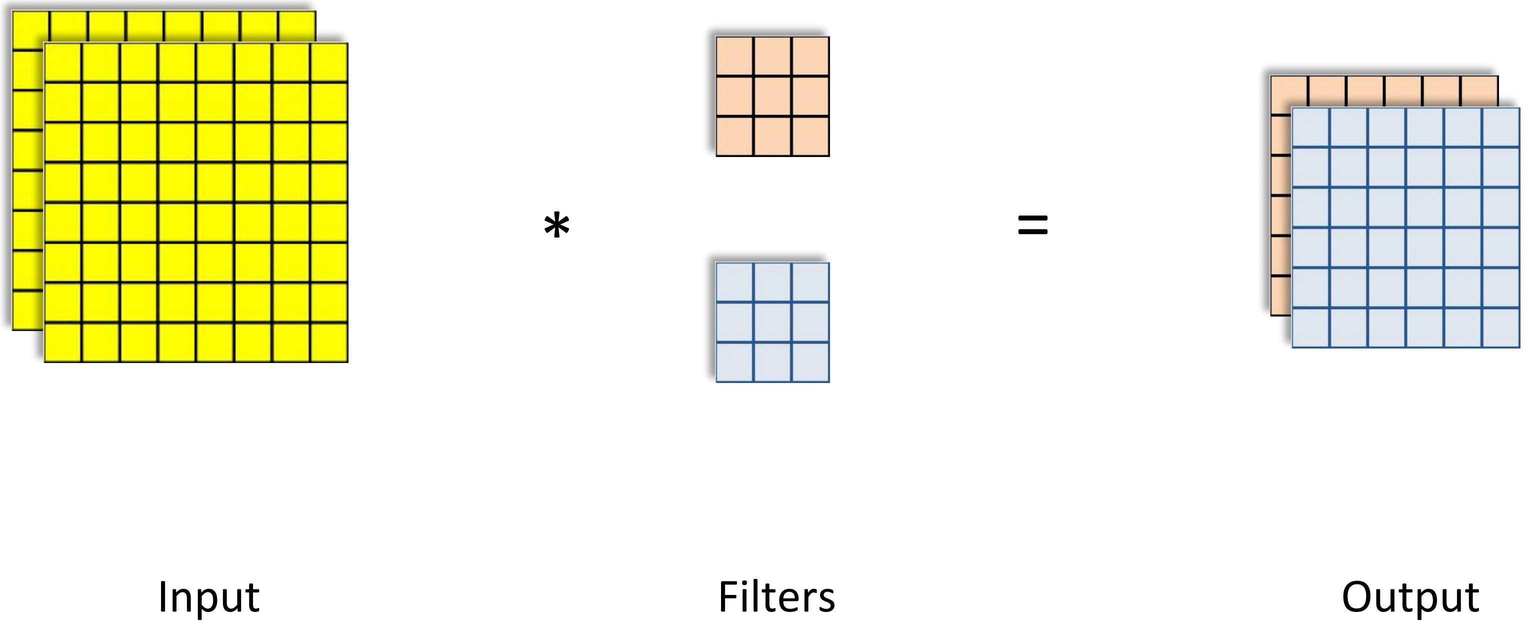


Input Filters Output

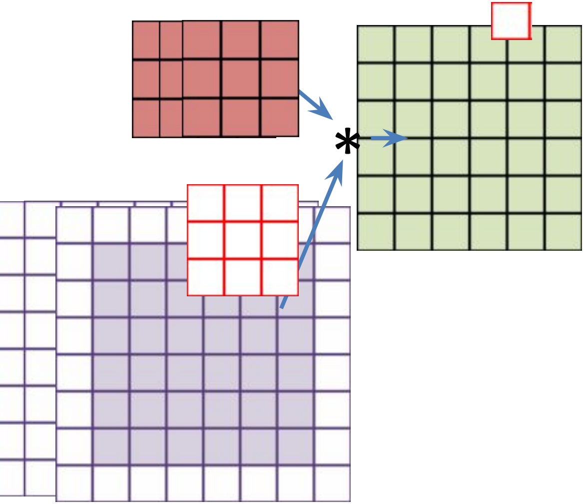
2-d convolution on 2 channels with 1xN and Nx1 filters

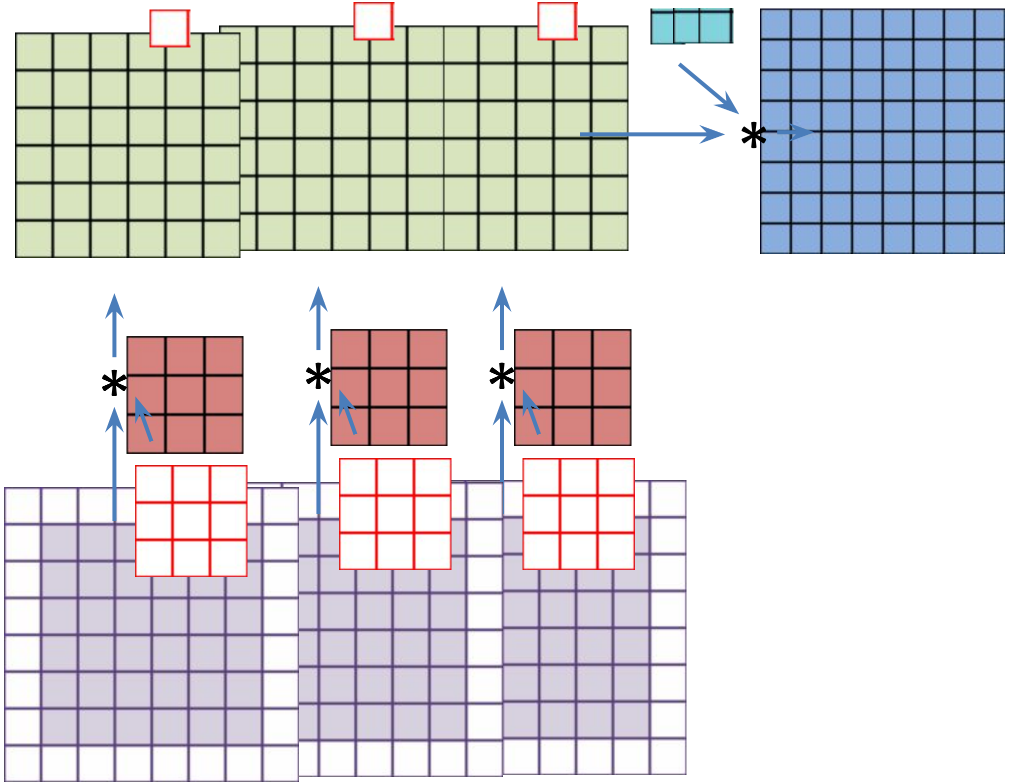


Depth-wise separable convolution



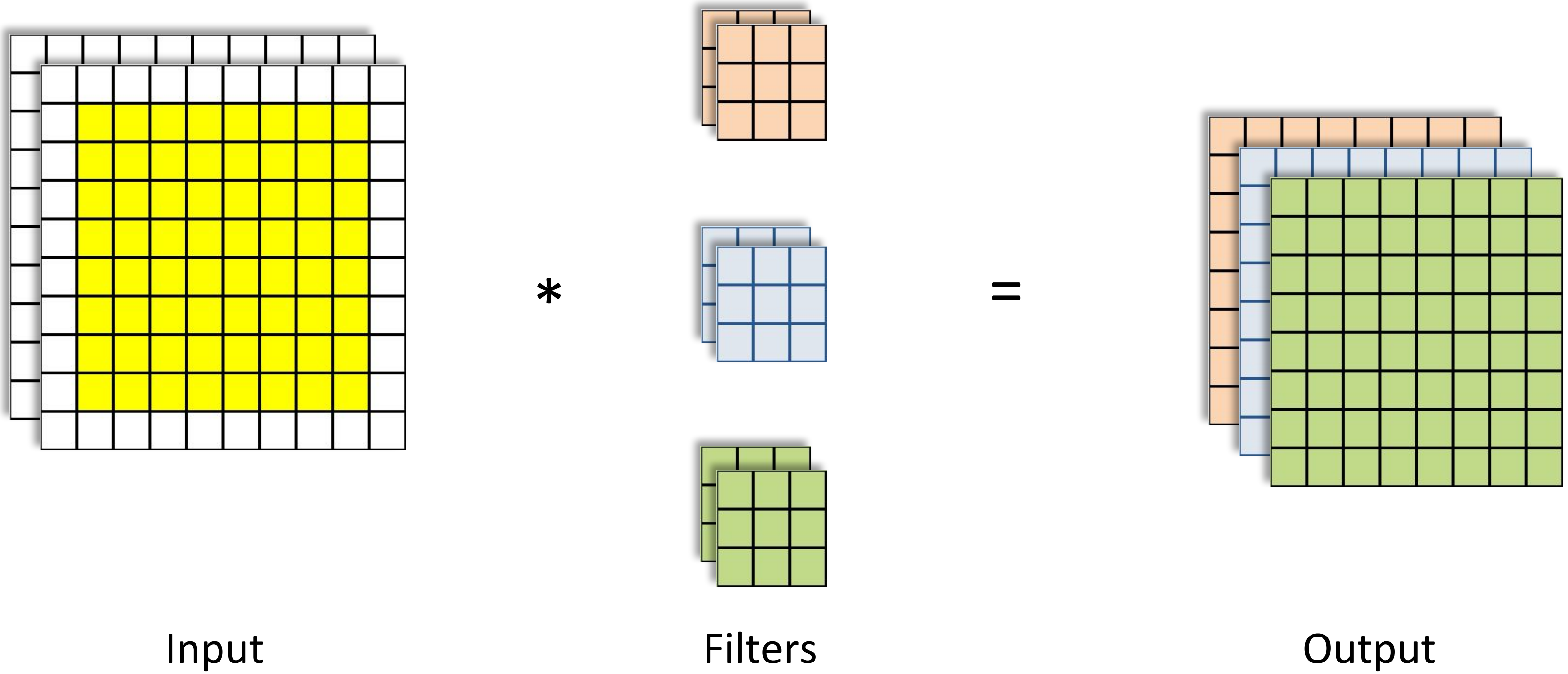
MobileNet filters each feature map separately



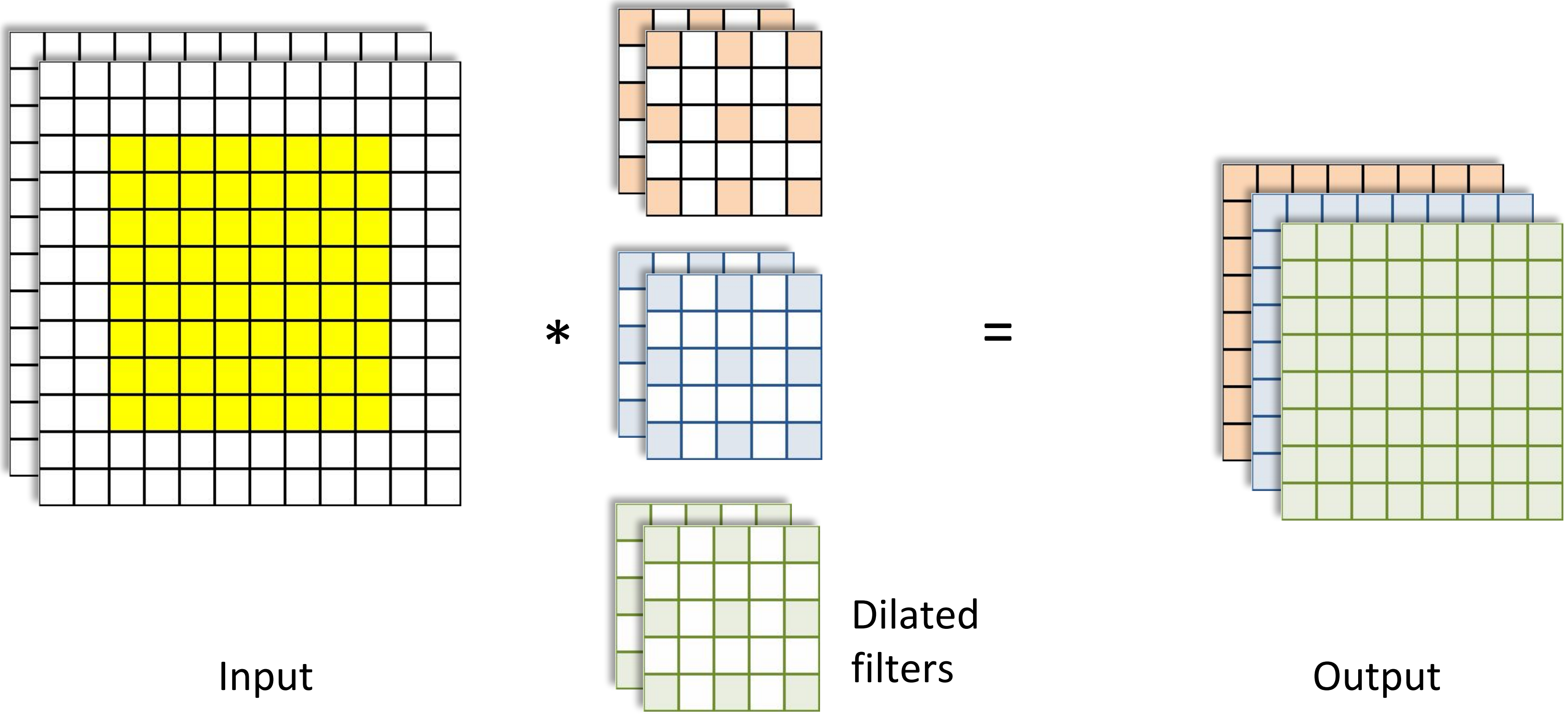


“MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications” by Andrew G. Howard Menglong Zhu Bo Chen Dmitry Kalenichenko Weijun Wang Tobias Weyand Marco Andreetto Hartwig Adam, Google , 2017

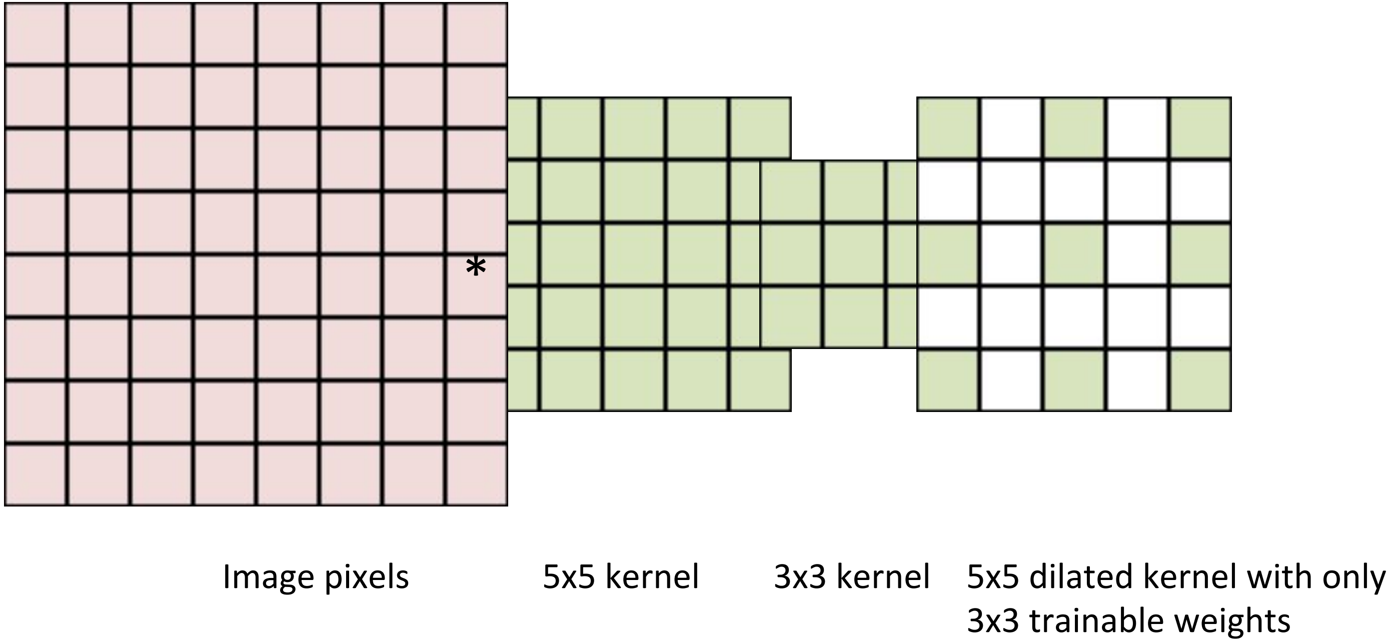
2-d convolution on 2 channels with 3 filters and zero-padding



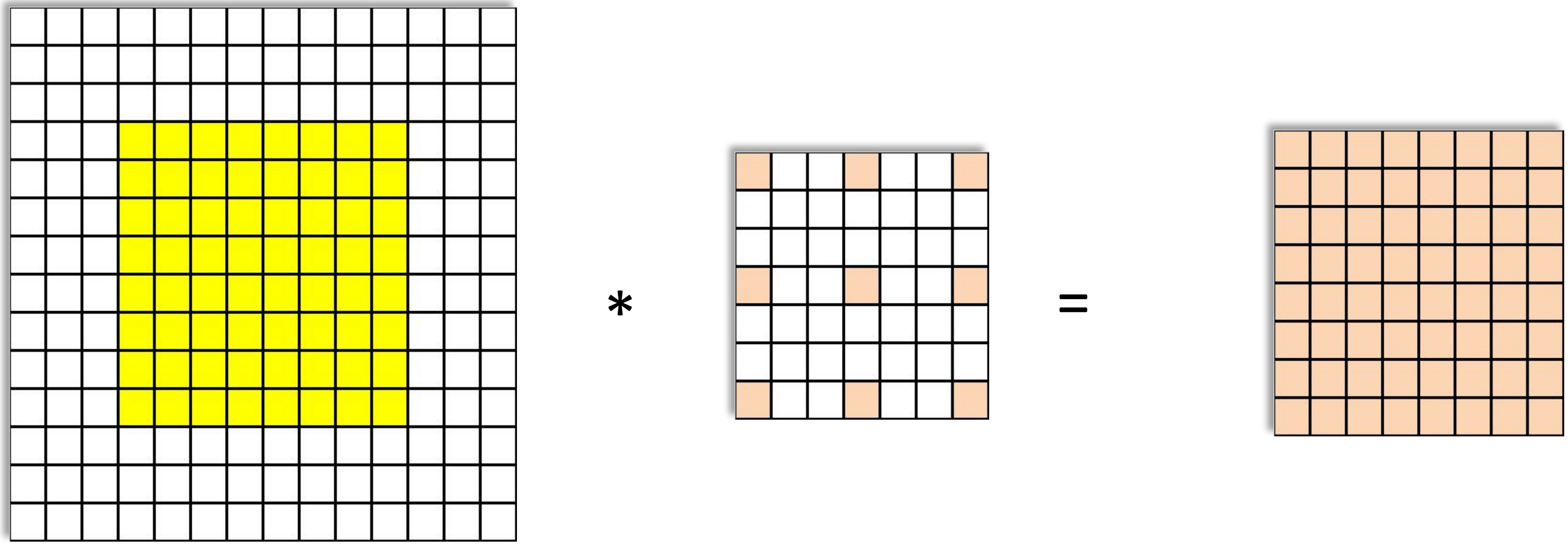
2-d dilated (atrous) convolution



Atrous (dilated) convolutions can increase the receptive field without increasing the number of weights



2-d dilated (atrous) convolution



Input　　Dilated filter　　Output

One way to up-sample is to carry forward max pooling indices

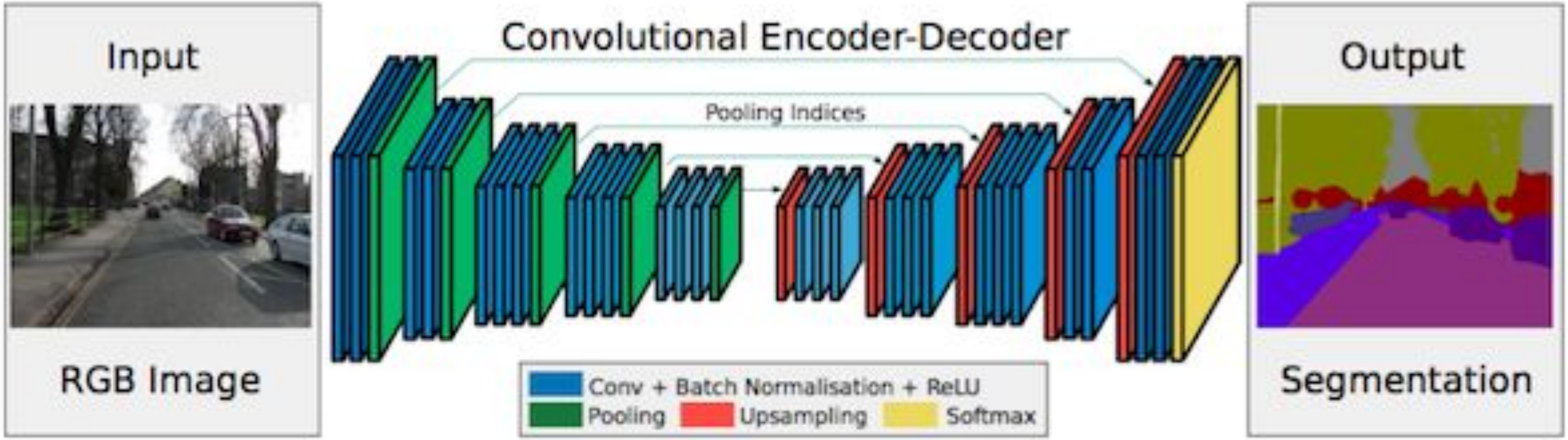
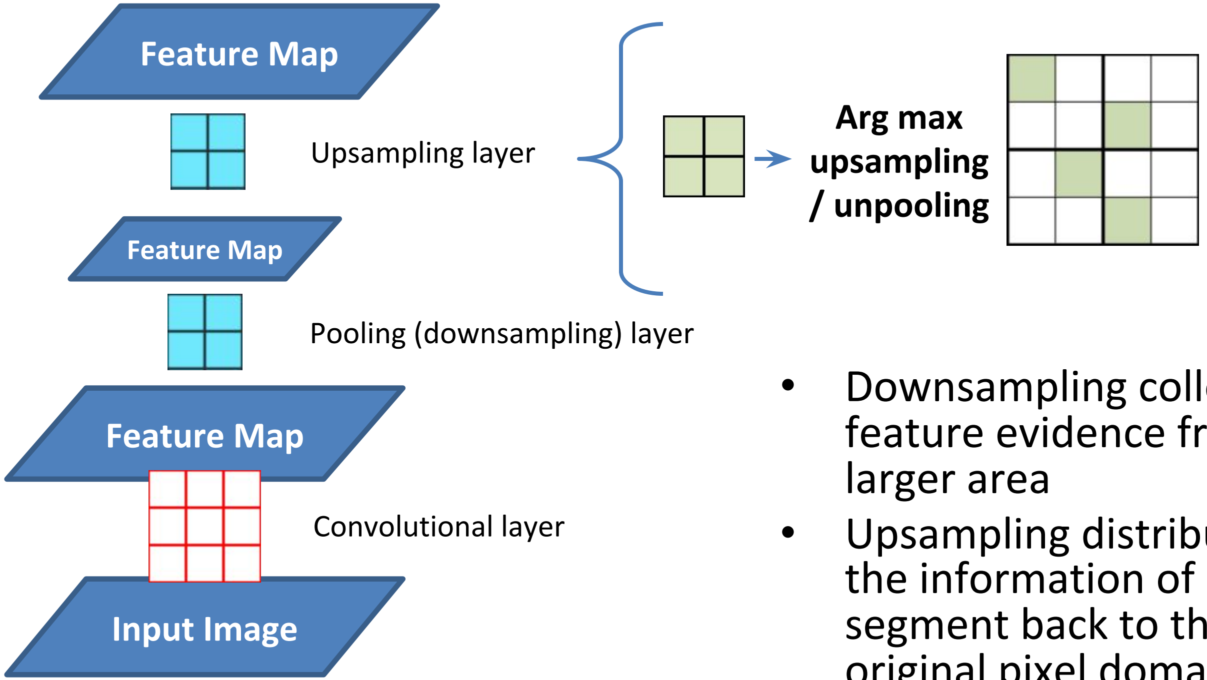


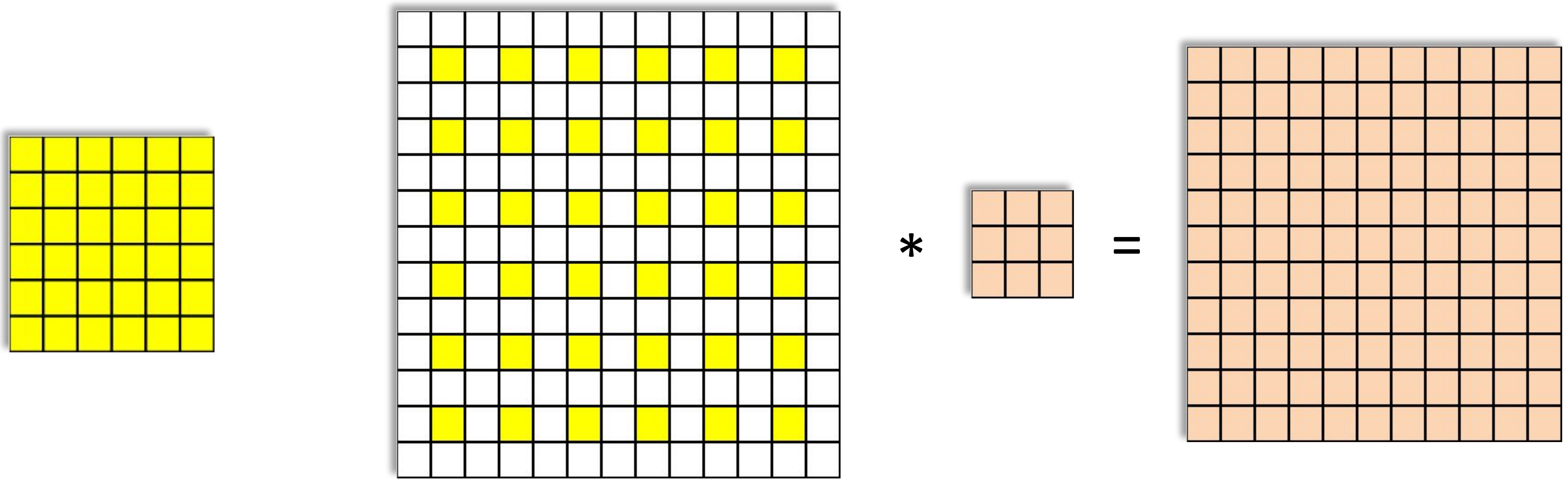
Image source: https://mc.ai/semantic-segmentation-and-transposed-convolution/

To produce a segmentation map downsampling is followed by upsampling



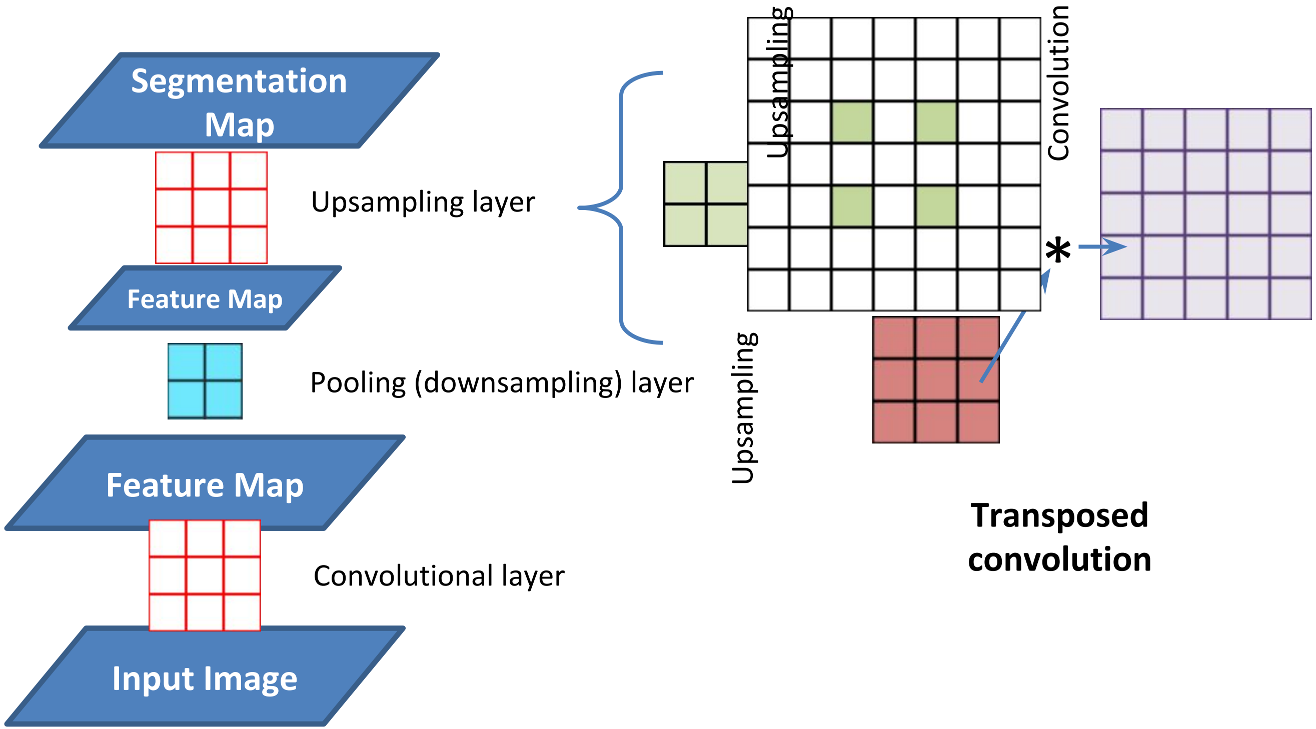
- Downsampling collects feature evidence from a larger area
- Upsampling distributes the information of a segment back to the original pixel domain

Learnable up-convolution



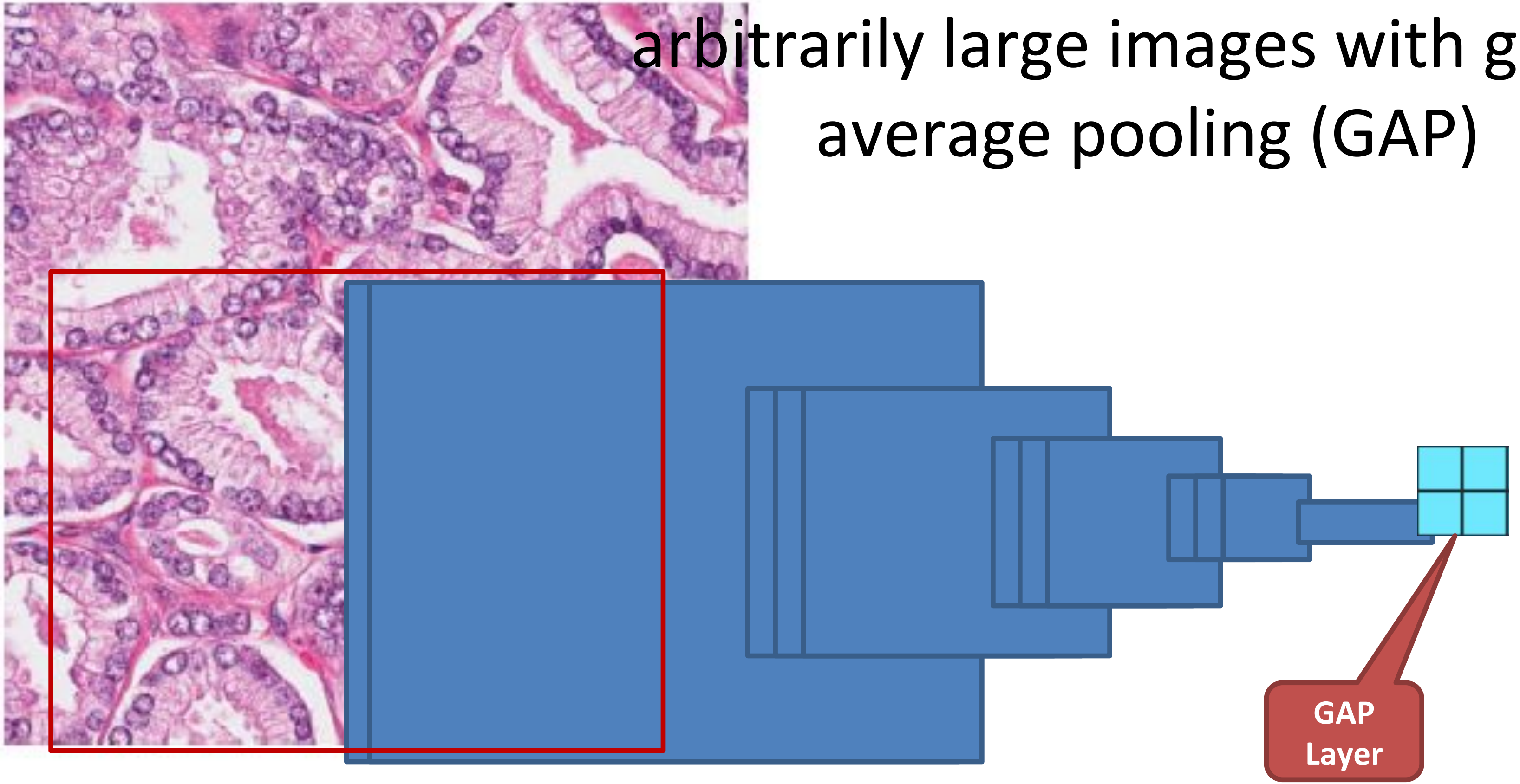
Input Dilated Input Filter Output

Upsampling can static or it can also be learned



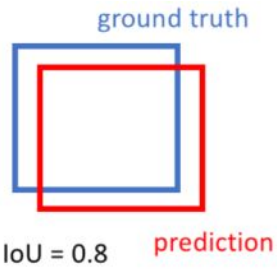
Transposed convolution

A standard architecture can process arbitrarily large images with global average pooling (GAP)



Loss functions for semantic segmentation

IoU is a concept
that is not
used as a loss



$$\frac{2|X \cap Y|}{|X| + |Y|} = \frac{2TP}{2TP + FP + FN}$$

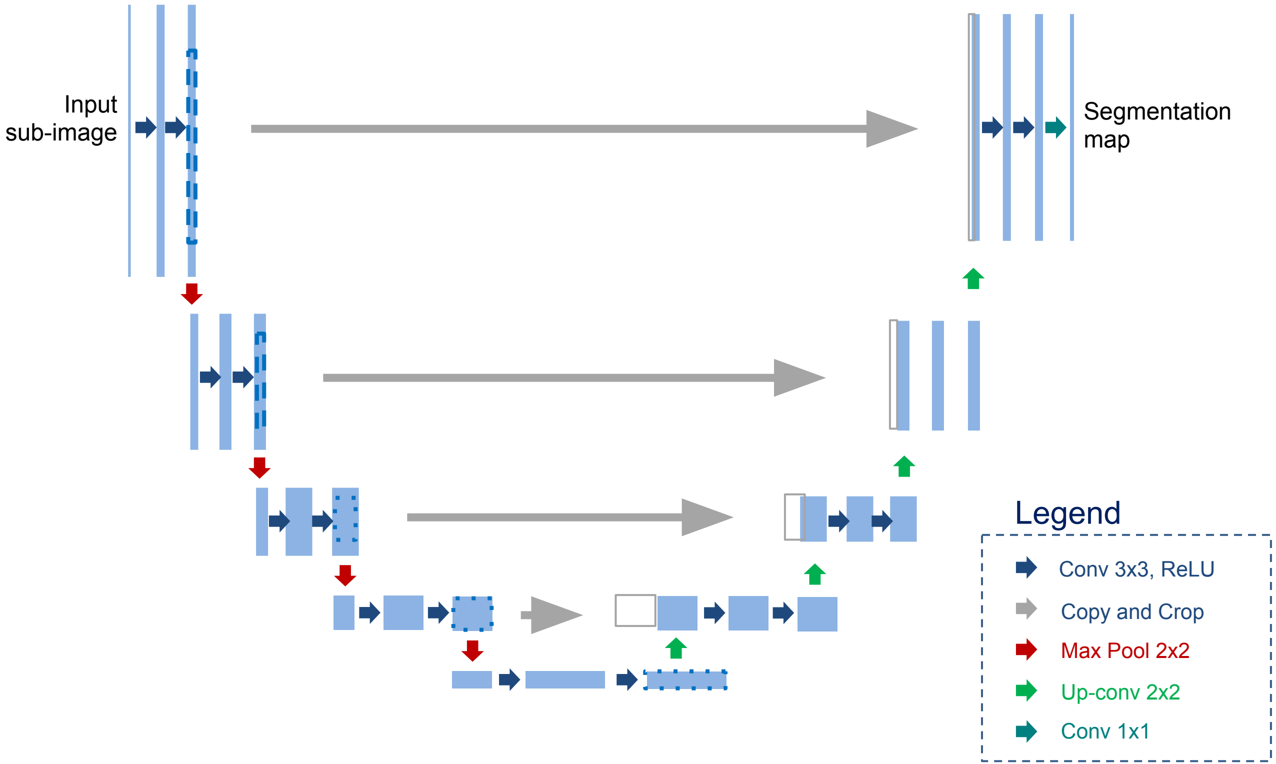
$$-\sum_{i}^{c}\sum_{j}^{N} y_i^j \log \hat{y}_i^j$$

Pixel-wise cross entropy loss

$$1 - \frac{1}{c}\sum_{i=0}^{c} \frac{\sum_j^N 2y_i^j \hat{y}_i^j + \epsilon}{\sum_j^N y_i^j + \sum_j^N \hat{y}_i^j + \epsilon}$$

DICE loss

U-Net is based on the ideas described in the previous slides



Source: "U-Net: Convolutional Networks for Biomedical Image Segmentation" Olaf Ronneberger, Philipp Fischer, Thomas Brox, 2015

Example: ICNet

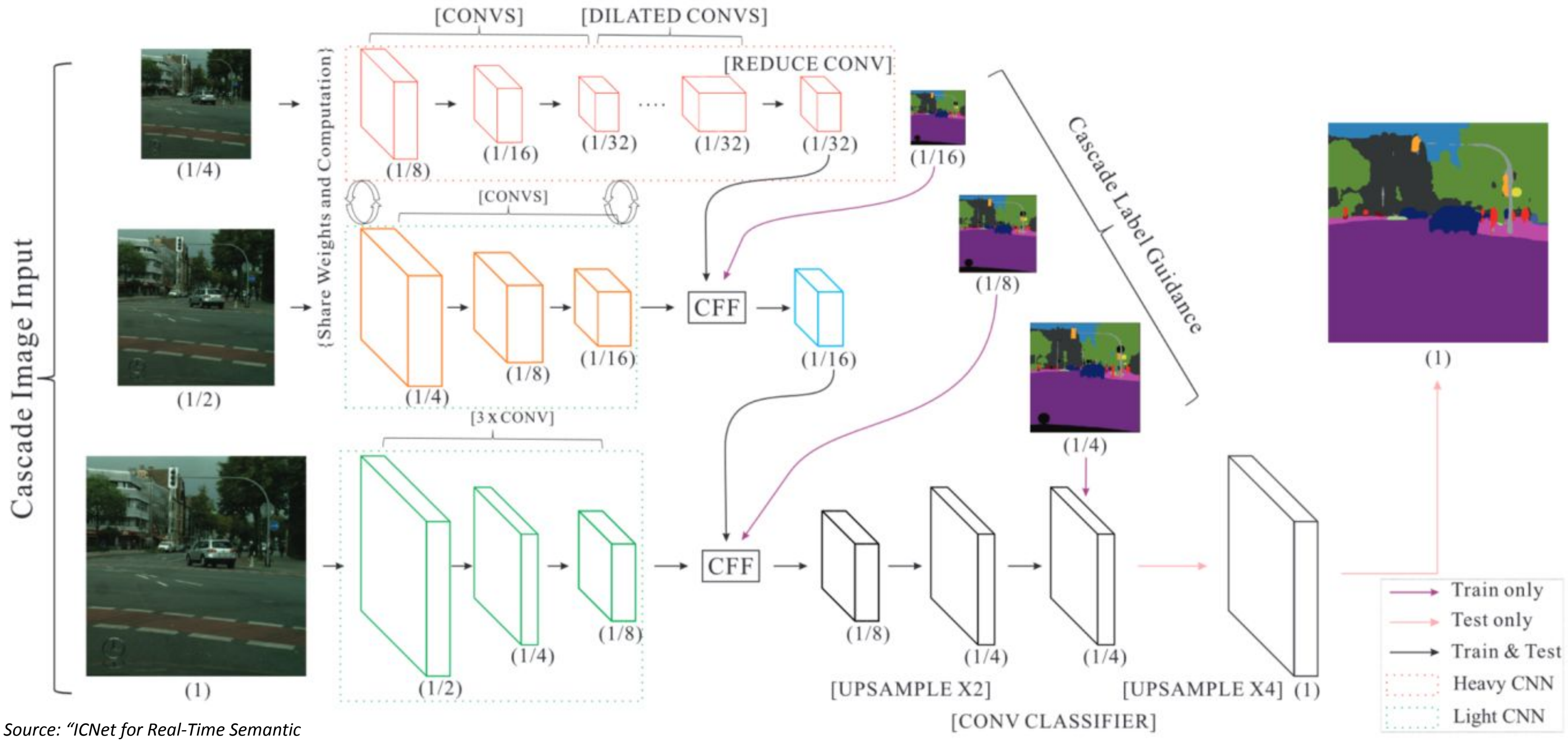


Image Source: "ICNet for Real-Time Semantic Segmentation on High-Resolution Images" Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, Jiaya Jia, ECCV'18

Sample results of ICNet

void	road	sidewalk	building	wall
fence	pole	traffic light	traffic sign	vegetation
terrain	sky	person	rider	car
truck	bus	train	motorcycle	bicycle

Image Source: "ICNet for Real-Time Semantic Segmentation on High-Resolution Images" Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, Jiaya Jia, ECCV'18