CSE 465
Lecture 8

More CNN Architectures

Group Convolution

- Input and kernel are split into g groups across channel dimension
- Each group then performs the convolutions independently
- Each layer is defined using following parameters:
  - # Input channels (C_1)
  - # Output channels (C_2)
  - Kernel size ($w_1 \times h_1$)
  - Padding
  - Stride
  - Dilation rate (r)
  - # of groups (g)
- Parameter reduction??

Group vs Standard Convolution Layer

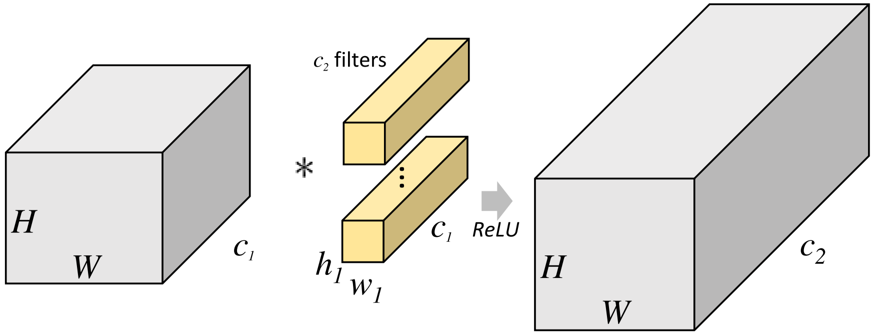


Figure: Standard convolution

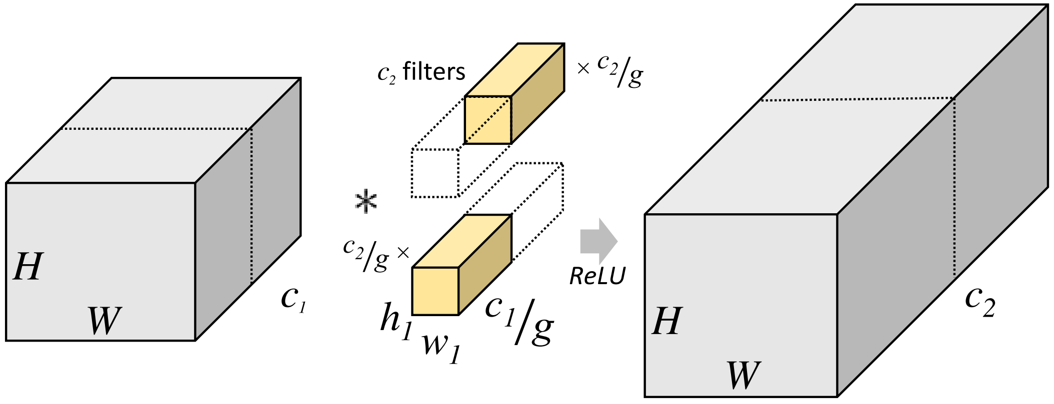


Figure: Grouped convolution

Depth-wise Convolution

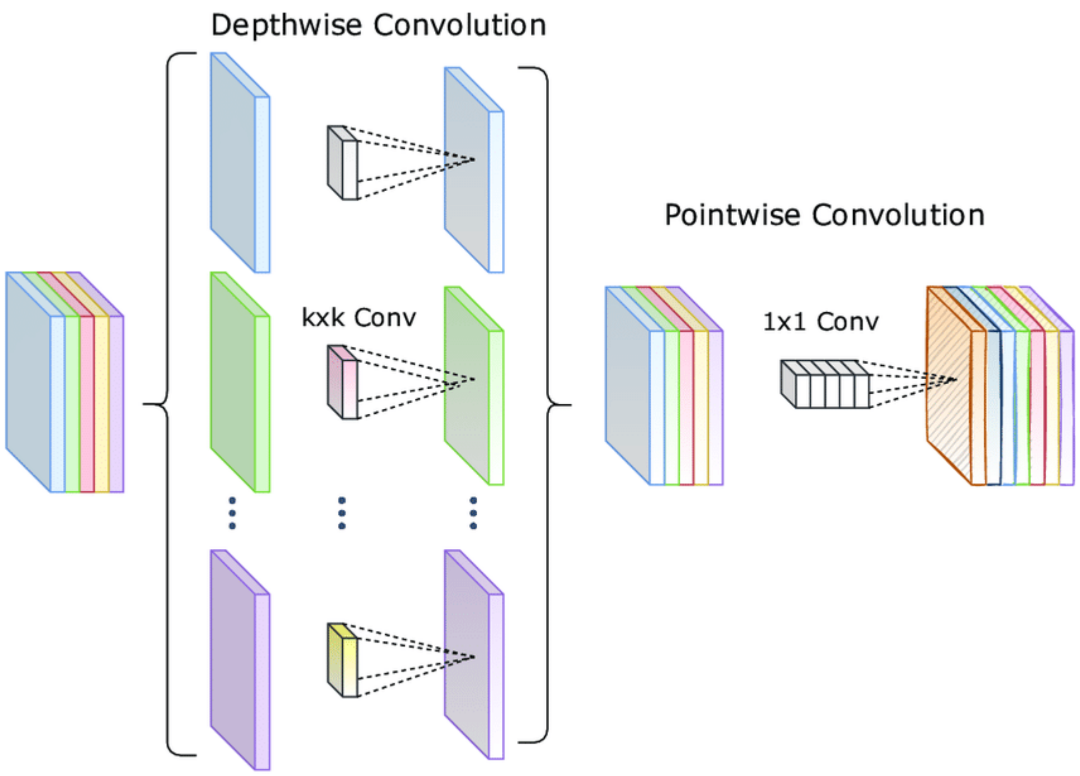
- Special case of group convolution where each channel is processed independently

  # input channels = # groups = # output channels

- Parameter reduction??

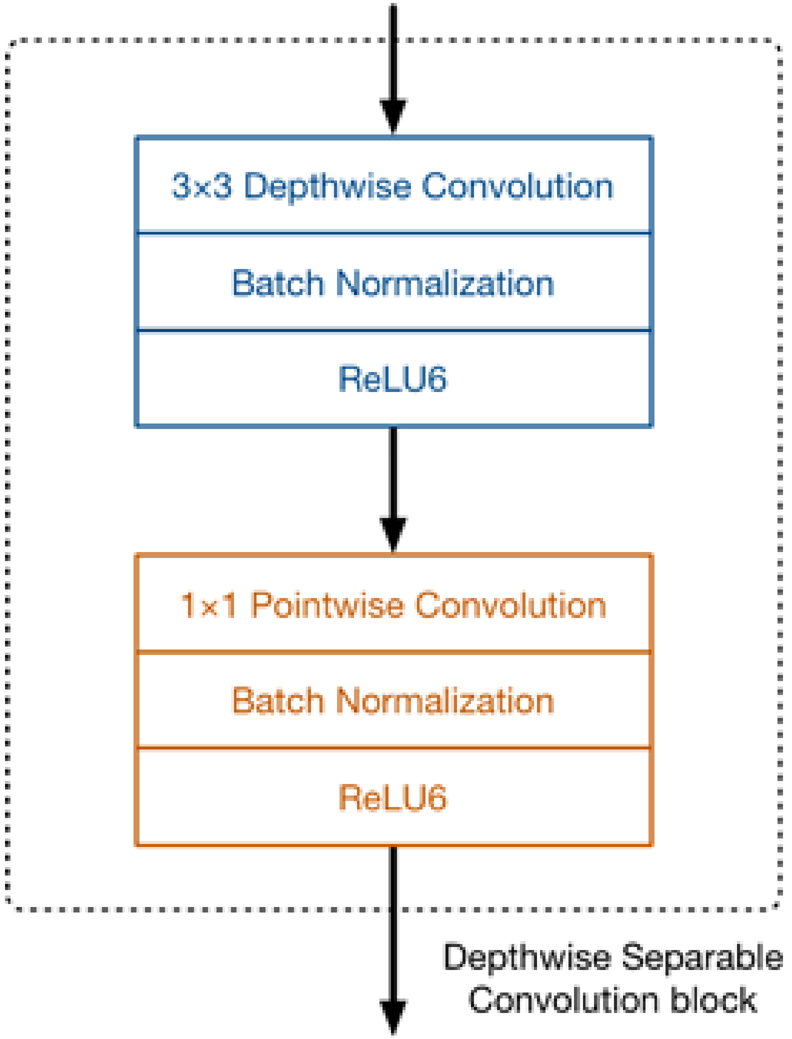
MobileNet

- Replaces expensive convolution layers by a cheaper **depthwise separable** convolution, a 3×3 depthwise convolution layer followed by a 1×1 pointwise convolution layer
- This requires a lot fewer learned parameters than a regular convolution but approximately does the same thing
-



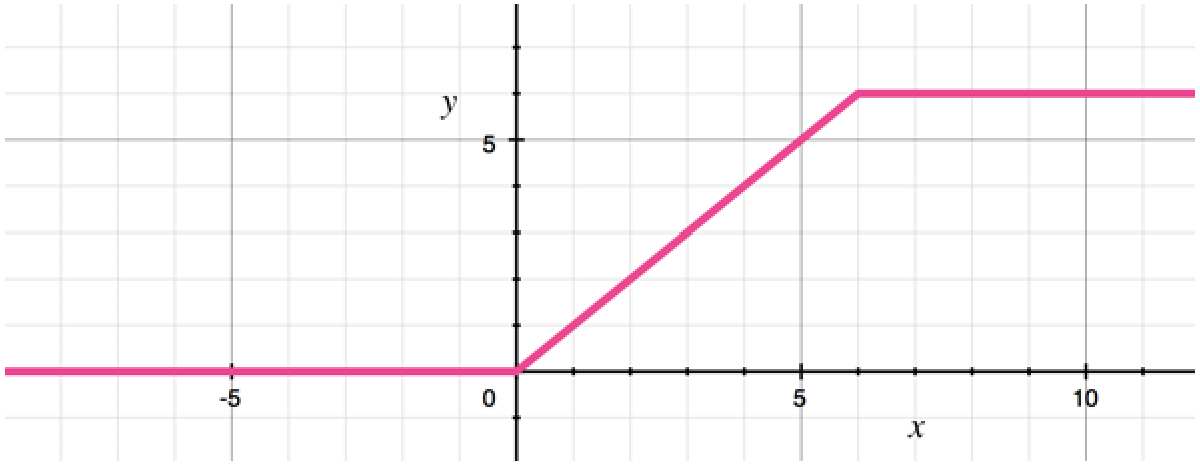
MobileNet

- There are no pooling layers in between these depthwise separable blocks
- Some of the depthwise layers have a stride of 2 to reduce the spatial dimensions of the data
  - In that case, the corresponding pointwise layer also doubles the number of output channels. If the input image is 224×224×3 then the output of the network is a 7×7×1024 feature map.



MobileNet

- The convolution layers are followed by batch normalization.
- The activation function used by MobileNet is ReLU6.
- This is like the well-known ReLU but it prevents activations from becoming too big:



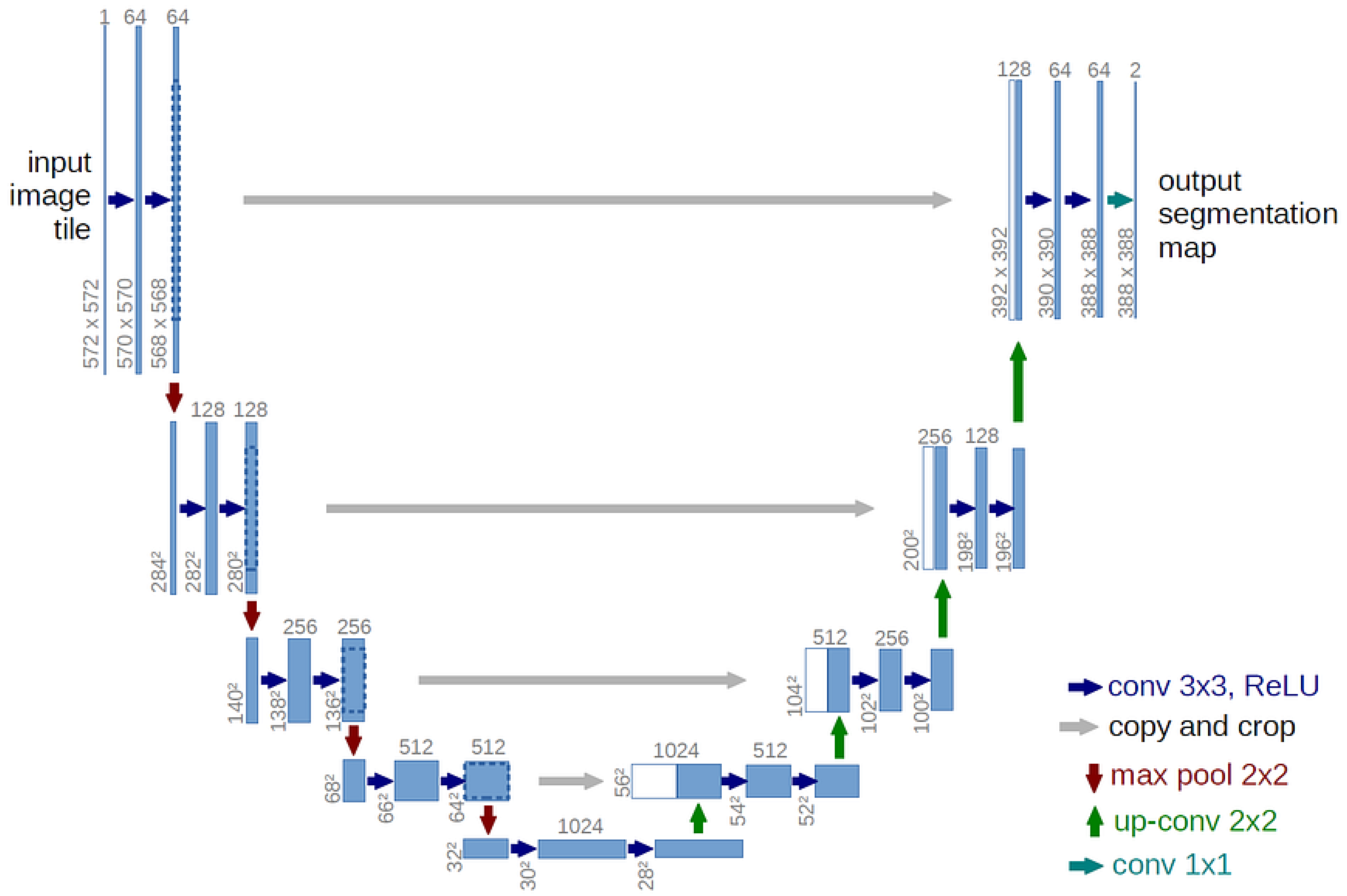
MobileNet

- MobileNet can be configured using hyperparameters
- The most important hyperparameter is the depth multiplier
  - That is how many channels are in each layer
- Using a depth multiplier of 0.5 will halve the number of channels used in each layer
  - Reducing the number of computations by a factor of 4
- MobileNet is roughly nine times faster if used with depthwise separable convolution

UNET – CNN for semantic segmentation

- UNET is one of the most popular architecture for medical image segmentation
- It is designed to learn from few training samples
- It uses fully convolutional scheme – no fully connected neurons are used
- UNET is a U-shaped encoder-decoder network architecture
- It consists of four encoder blocks and four decoder blocks that are connected via a bridge

UNET – CNN for semantic segmentation



UNET – CNN for semantic segmentation

- The encoder network (contracting path) half the spatial dimensions and double the number of features (feature channels) at each encoder block
- The decoder network doubles the spatial dimensions and half the number of feature channels.

UNET – Encoder Block

- The encoder network acts as the feature extractor and learns an abstract representation of the input image through a sequence of the encoder blocks
- Each encoder block consists of two 3x3 convolutions followed by a ReLU (Rectified Linear Unit) activation function
- Next follows a 2x2 max-pooling, where the spatial dimensions (height and width) of the feature maps are reduced by half
- This reduces the computational cost by decreasing the number of trainable parameters

UNET – Skip Connection

- Skip connections provide additional information that helps the decoder to generate better semantic features
- They also act as a shortcut connection that helps the indirect flow of gradients to the earlier layers without any degradation
- Skip connection helps in better flow of gradient while backpropagation, which in turn helps the network to learn better representation

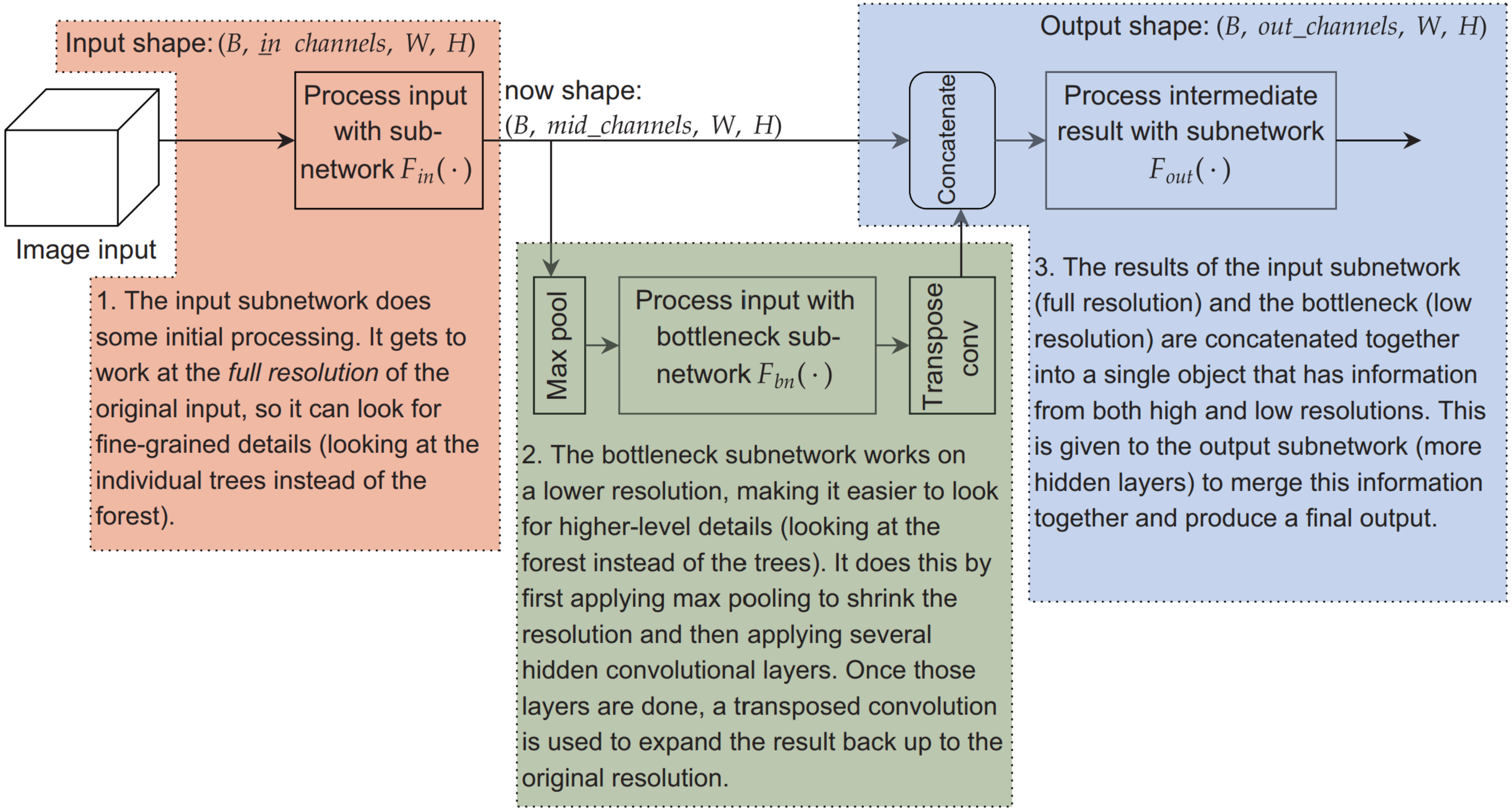
UNET – bridge

- The bridge connects the encoder and the decoder network and completes the flow of information
- It consists of two 3x3 convolutions, where a ReLU activation function follows each convolution

UNET – Decoder

- The decoder network is used to take the abstract representation and generate a semantic segmentation mask
- The decoder block starts with a 2x2 transpose convolution
  - Next, it is concatenated with the corresponding skip connection feature map from the encoder block
- After that, two 3x3 convolutions are used, where a ReLU activation function follows each convolution
-

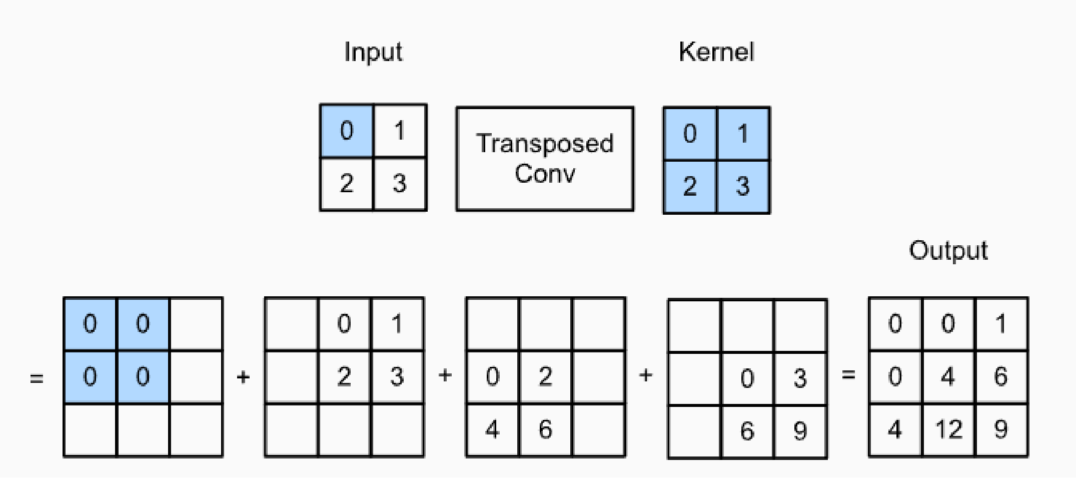
UNET in Summary

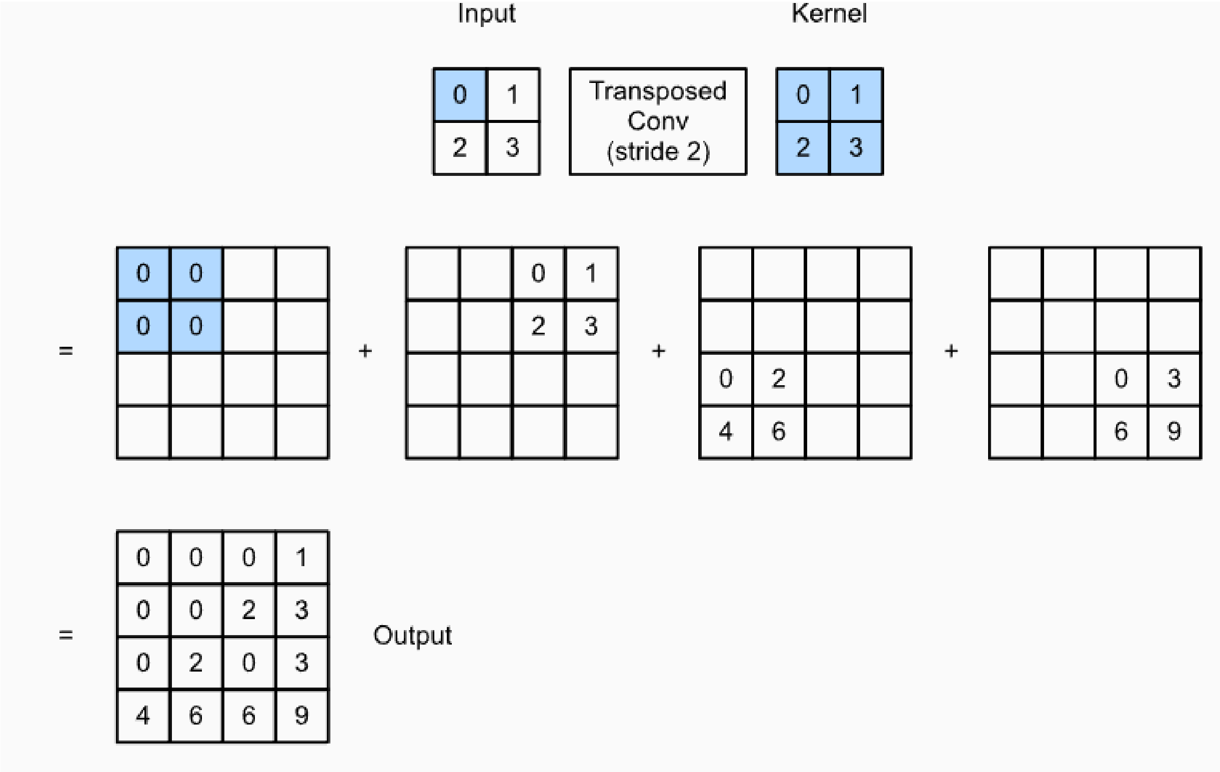


Transpose Convolution

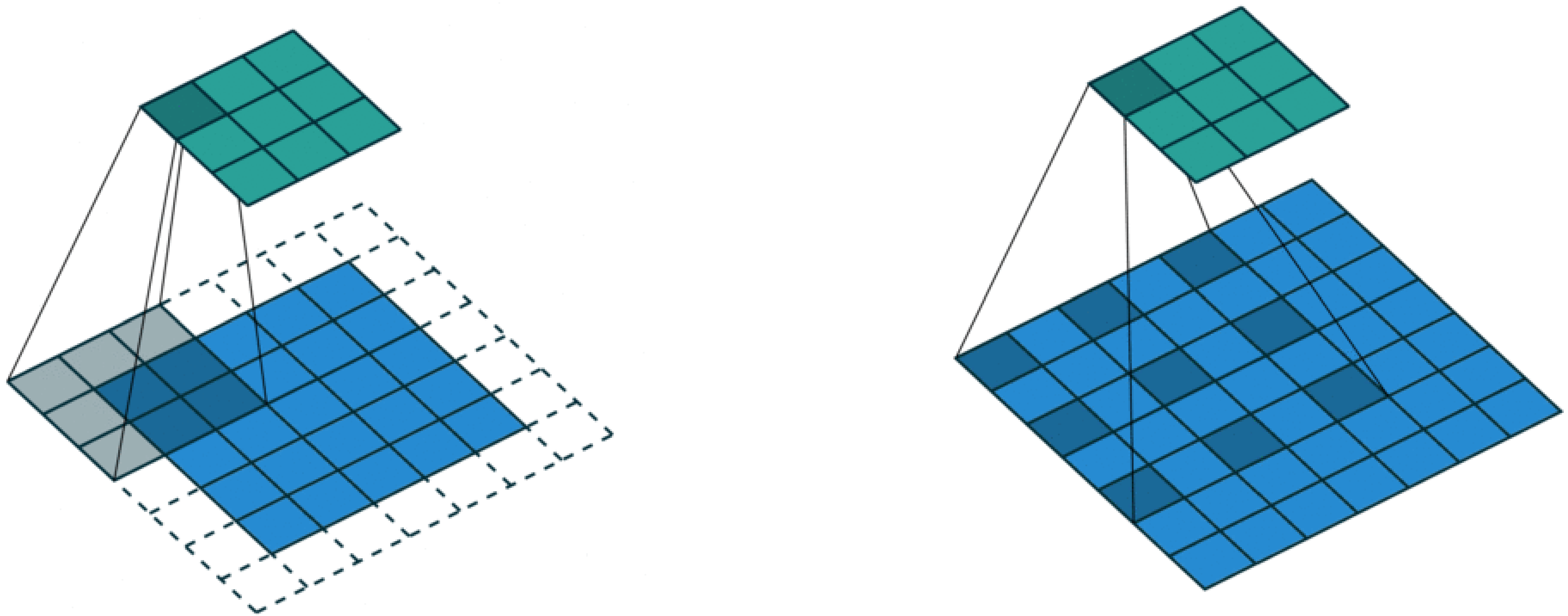
- A transposed convolution (can be thought as) the reverse of convolution
  - Get back to the same spatial resolution as the original image
- A transposed convolutional layer carries out a regular convolution but reverts its spatial transformation

Transpose Convolution

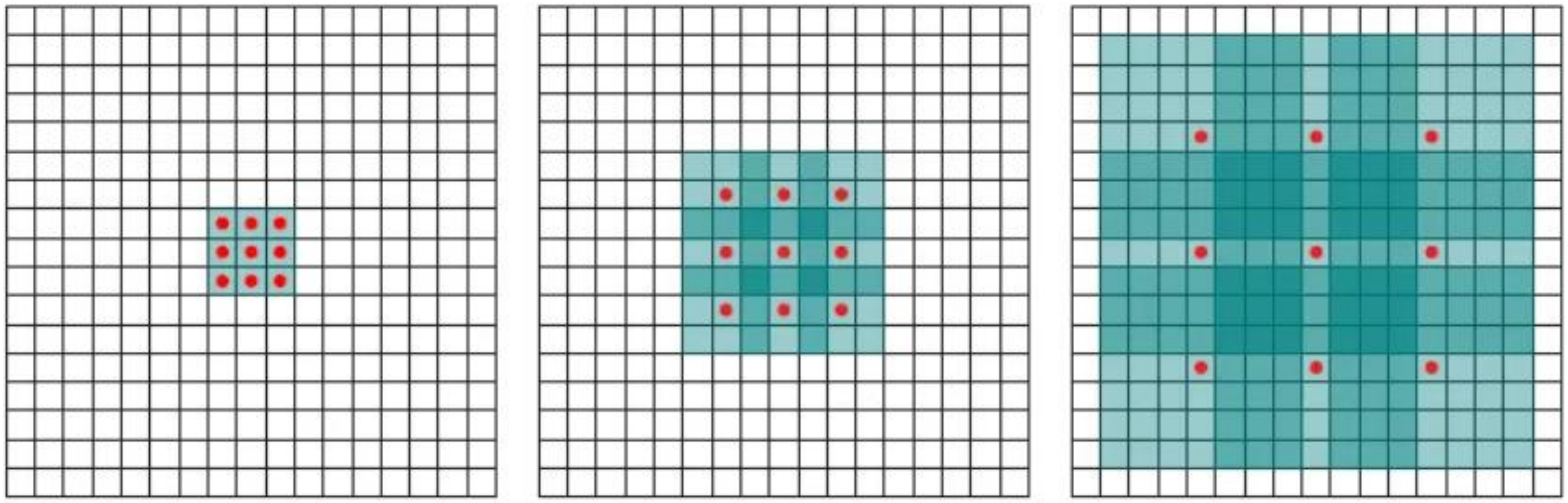




Dilated Convolution



Dilated Convolution



l=1 (left), l=2 (Middle), l=4 (Right)