Deep Learning

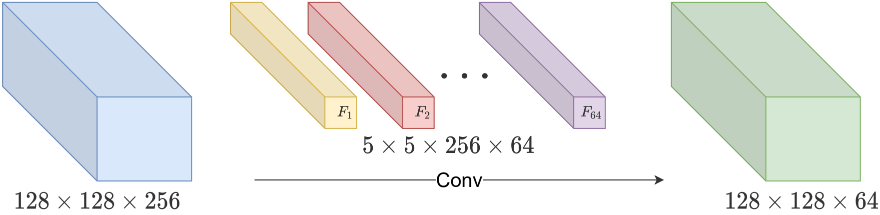
Variations of Convolutional Neural Networks

Syed Irtaza Muzaffar

CNN Variations

- There are *lots* of variations of the basic CNN idea.
  - Fully convolutional networks. No pooling and no fully connected layer.
  - 1×1 convolutions to reduce computations.
  - Inception modules to combine multiple filter sizes.
  - Residual blocks to avoid vanishing gradients.
  - Depthwise separable convolutions to reduce parameters and computations.
  - Lightweight and fast models (SqueezeNet, MobileNet, ...) for edge computing.
  - Fast search over hyperparameters (EfficientNet).
- A whole course can be dedicated to CNNs alone.

Cost of Convolution Layer

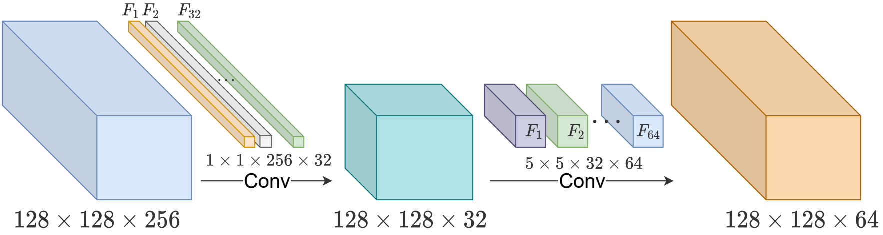


$$\text{Cost} = \#\ \text{multiplications} = \underbrace{(128 \times 128 \times 64)}_{\text{Output neurons}} \times \underbrace{(5 \times 5 \times 256)}_{\text{Cost per neuron}}$$

$$= 6710886400$$

$$= 6.7 \text{ billion}$$

1×1 convolution



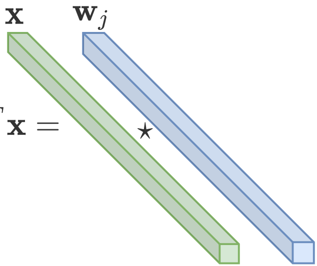
$$
\begin{aligned}
\text{Cost} &= \left(\underbrace{(128 \times 128 \times 32)}_{\text{Output neurons}} \times \underbrace{(1 \times 1 \times 256)}_{\text{Cost per neuron}} \right) + \left(\underbrace{(128 \times 128 \times 64)}_{\text{Output neurons}} \times \underbrace{(5 \times 5 \times 32)}_{\text{Cost per neuron}} \right) \\
&= 134217728 + 838860800 \\
&= 973078528 = 0.97 \text{ billion}
\end{aligned}
$$

Almost 7 times reduction in number of multiplications to produce output volume of the same size.

1×1 convolution

- A 1×1 convolution is just a linear combination of the input channels.
- The fully connected layer of a traditional MLP can also be represented via 1×1 convolutions.

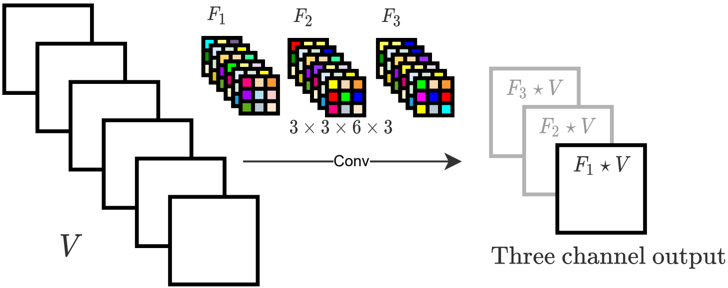
$\mathbf{w}_j^T \mathbf{x} =$



Depthwise Separable Convolution

What happens in standard convolution?

Consider the case of standard convolution using 3 filters.

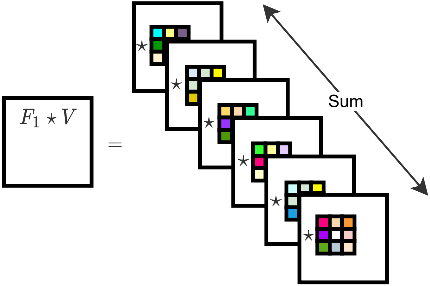


Number of weights to produce 3 channel output $= 3 \times 3 \times 6 \times 3 = 162$.

Depthwise Separable Convolution

What happens in standard convolution?

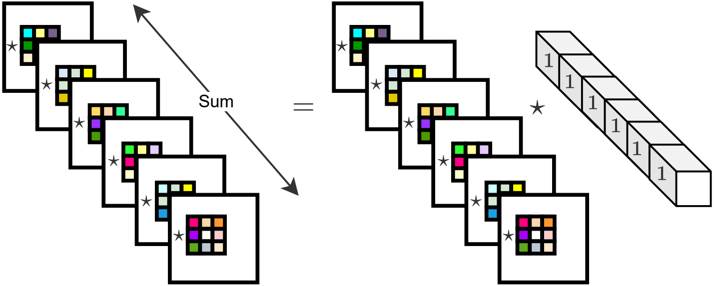
The first output channel is produced by 6 channel-wise convolutions that are then added together.



Depthwise Separable Convolution

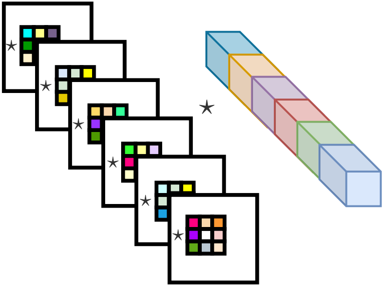
What happens in standard convolution?

Summation of per-channel results corresponds to 1×1 convolution with a volume of 1s.



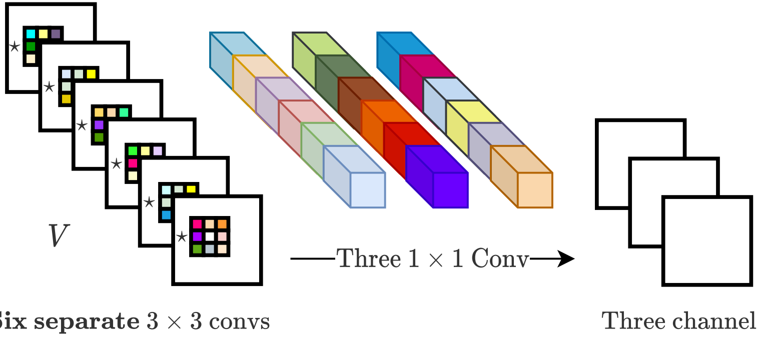
Depthwise Separable Convolution

Replace sum by a linear combination. This is called a *depthwise separable convolution*.



Depthwise Separable Convolution

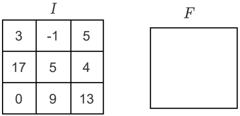
Multiple linear combinations lead to multiple output channels.



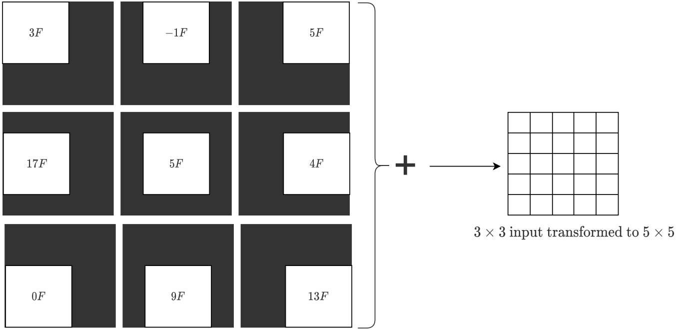
Number of weights to produce 3 channel output $= (3 \times 3 \times 6) + (6 \times 3) = 72$.

Expensive convolution (excluding the summation) is performed only once.
Multiple channels are produced via cheap 1×1 convolution.

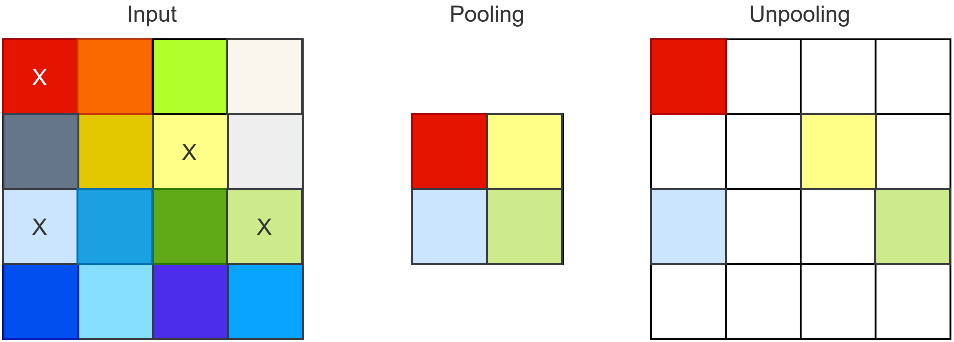
Transposed Convolution



A *transposed convolution* superimposes copies of the filter F scaled by the values in input I. Can be used to increase size.



Unpooling



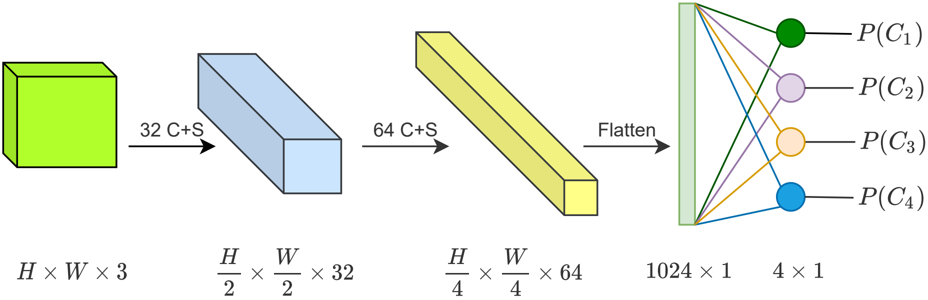
4×4 2×2 4×4

Reverses the size reduction effect of subsampling.

Fully Convolutional Networks (FCN)

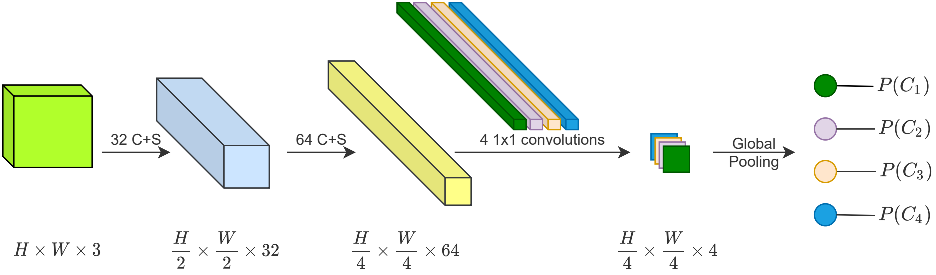
- An architecture for semantic segmentation.
- Only locally connected layers: convolution, pooling and upsampling.
- No fully connected layers (fewer parameters, faster training).
- Input image can be of any size.

The problem with fully connected layers



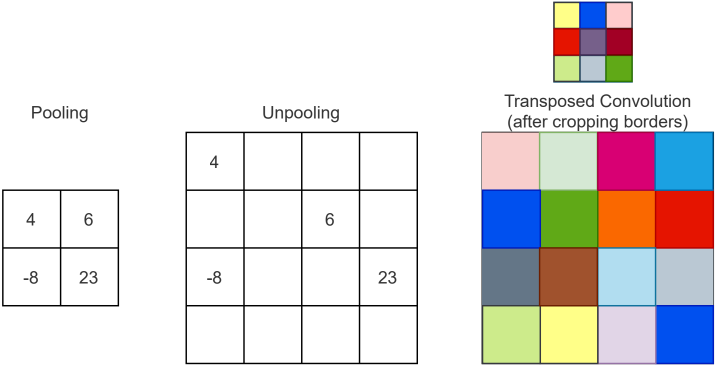
- K-class classification of an input image requires K softmax neurons at the output.
- 1024 neurons in fully connected layer imply that $H \times W$ must equal 256.
- So this can work with images of a certain size.

Fully Convolutional Networks



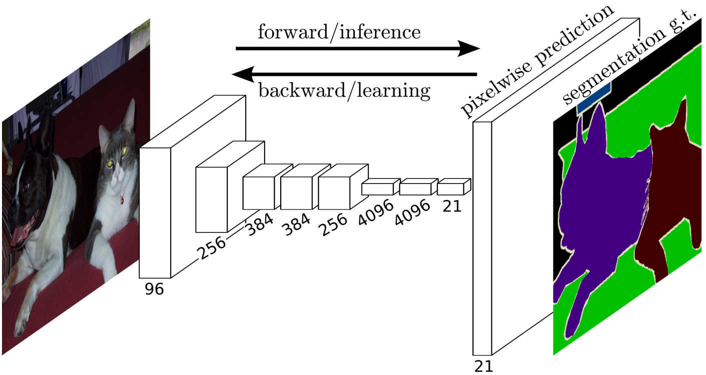
- K 1×1 convolutions corresponding to K classes.
- Followed by global pooling in each of the K channels.
- Followed by softmax.
- *Can work with images of any size.*

Image Generation via CNN



- Subsampled 2×2 result unpooled to a sparse 4×4 result that is then filled in via transposed convolution.
- Repeatedly upsample to obtain output of the same size as input.
- To generate images, use identity function at output.
- To generate pixel labels, use sigmoid or softmax.

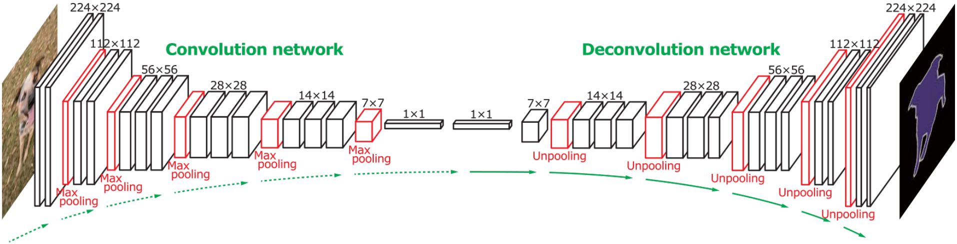
FCN for Semantic Segmentation[1]



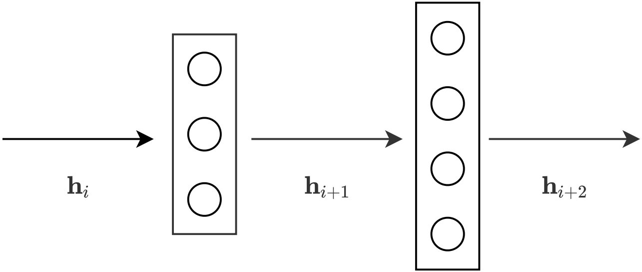
Each output pixel belongs to one of 21 classes.

[1]Segment image regions corresponding to different objects and find class of each object as well.

DeconvNet for Semantic Segmentation

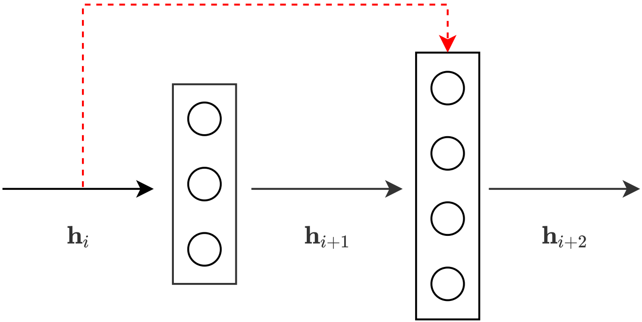


Residual Block



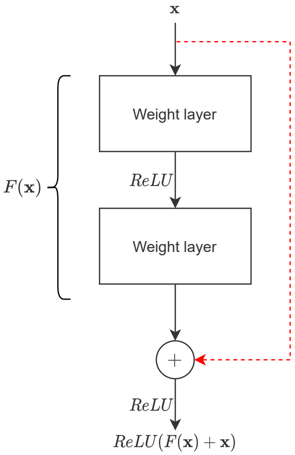
Standard propagation through two layers.

Residual Block



Skip connection between two layers.

Residual Block



If $F(\mathbf{x})$ approaches zero for any reason (e.g. due to weight regularization), the original input $\mathbf{x}$ can still be carried through.

Summary

- Vanilla CNNs have been extended in many ways.
- 1×1 convolutions reduce computations and allow the construction of FCNs.
- Depth-wise separable convolutions reduce parameters and computations.
- Unpooling and transposed convolutions generate upsampled results to output images instead of vectors.
- Residual blocks avoid vanishing gradients and make the learning task easier.
- FCNs have no fully connected layers. They allow inputs of any size.