1. Convolutions

In the case of a CNN, the convolution is performed on the input data with the use of a filter or kernel (these terms are used interchangeably) to then produce a feature map. We execute a convolution by sliding the filter over the input. At every location, a matrix multiplication is performed and sums the result onto the feature map, as shown in Figure 1.

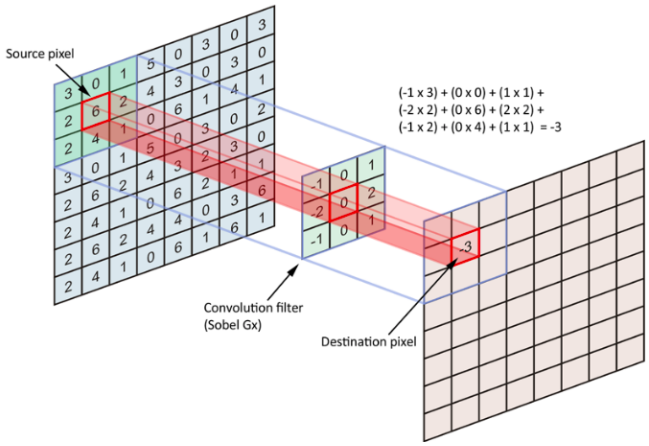

Figure 1. The filter slides over the input and performs its output on the new layer.

In reality convolutions are often performed in 3D. Each image is namely represented as a 3D matrix with a dimension for width, height, and depth. Depth is a dimension because of the colours channels used in an image (RGB).

- An image matrix (volume) of dimension $(h \times w \times d)$
- A filter of dimension $(f_h \times f_w \times d)$

Stride is the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on.
Sometimes filter does not fit perfectly fit the input image. We have two options:

- Pad the picture with zeros (**zero-padding**) so that it fits;
- Drop the part of the image where the filter did not fit. This is called valid padding which keeps only valid part of the image.

1.1. Dilated convolutions.

Dilated convolutions introduce another parameter, denoted as r, called the dilation rate. Dilations introduce "holes" in a convolutional kernel. The "holes" basically define a spacing between the values of the kernel. So, while the number of weights in the kernel is unchanged, the weights are no longer applied to spatially adjacent samples. Dilating a kernel by a factor of r introduces a kind of striding of r.
Let's see 3 sequential convolution in Figure 2. Layers (denoted by a,b,c) that are illustrated in the image with normal convolution, $r = 2$ dilation factor, and $r = 4$ dilation factor. In Figure 2 (a) we have a normal 3x3 convolution with receptive field 3x3. In Figure 2 (b) we have a 2-dilated 3x3 convolution that is applied in the output of layer (a) which is a normal convolution. As a result, each element in the 2 coupled layers now has a receptive field of 7×7. If we studied 2-dilated convolution alone the receptive field would be simply 5x5 with the same number of parameters. In Figure 2 (c) by applying a 4-dilated convolution, each element in the third sequential convolution layer now has a receptive field of 15×15. As a result, the receptive field grows exponentially while the number of parameters grows linearly.

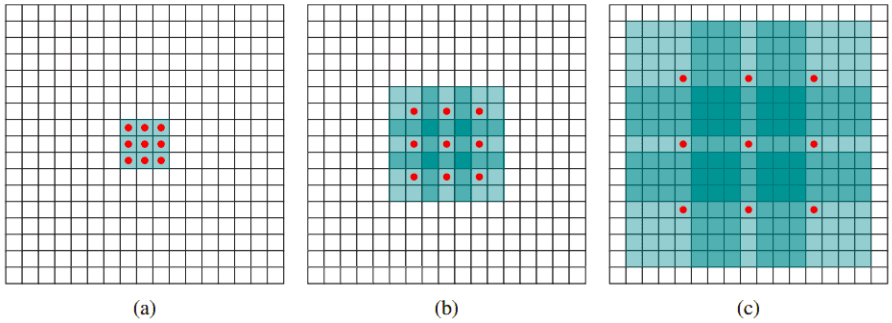

FIGURE 2. (a)normal convolution, (b) r=2 dilation factor, and (c) r=4 dilation factor.

2. **Pooling**

The Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model.
There are two popular types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.

3. **Receptive Field**

Receptive Field (RF) is defined as the size of the region in the input that produces the feature[3]. Basically, it is a measure of association of an output feature (of any layer) to the input region (patch).
A convolutional unit only depends on a local region (patch) of the input. It is trivial to talk about RF on fully connected layers since each unit has access to all the input region.
Why do we care about the receptive field of a convolutional network? For instance, if we want to predict the boundaries of an object (i.e. a car, an organ like the heart, a tumor) it is important that we provide the model access to all the relevant parts of the input object that we want to segment. In Figure 3, you can see two receptive fields: the green and the orange one. Which one would you like to have in your architecture? Obviously the orange one is better for our inference. It is thus important to design a convolutional model so that we ensure that its RF covers the entire relevant input image region.

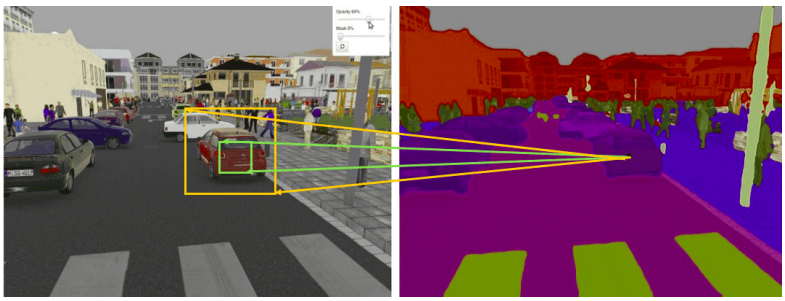

FIGURE 3. The green and orange rectangles are two different receptive fields. Which one would you prefer?

There are a plethora of ways and tricks to increase the RF, that can be summarized as follows:

- Add more convolutional layers (make the network deeper)

- Add pooling layers or higher stride convolutions (sub-sampling)
- Use dilated convolutions
- Depth-wise convolutions (see MobileNet [1])

References

[1] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.