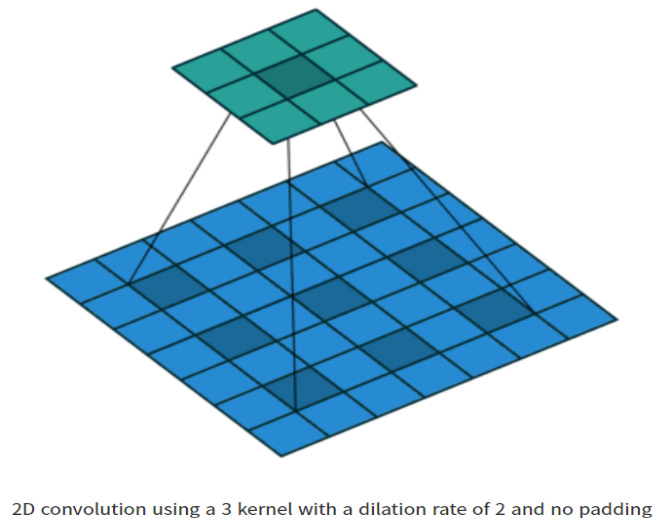
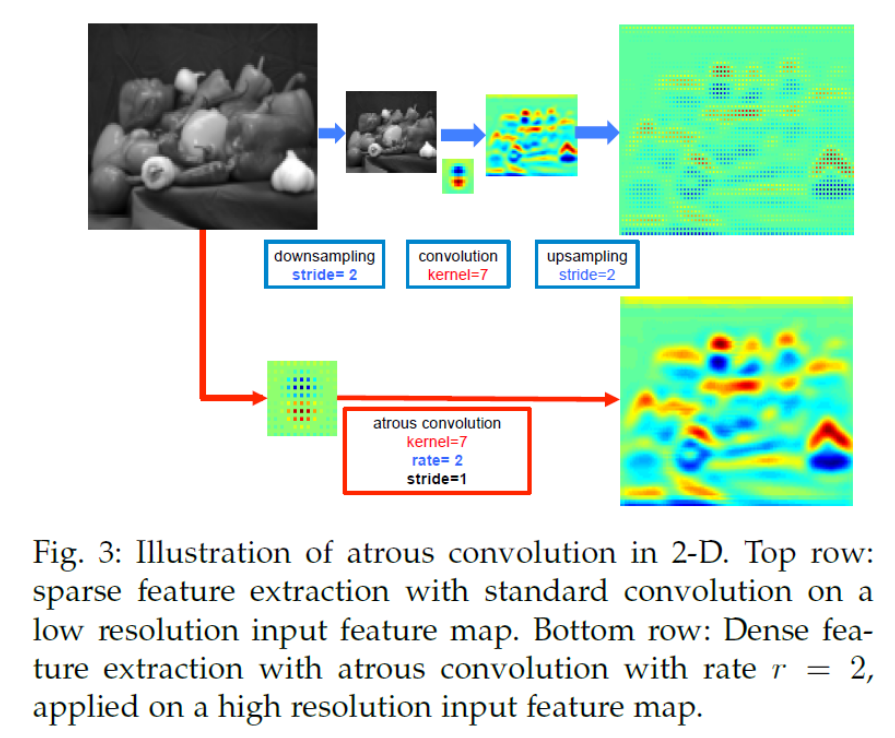
1. Dilation Convolution



-기존의 Convolution layer에 dilation rate라는 parameter를 추가한 형태이다. 이때, dilation rate는 기존의 kernel이 수용하는 픽셀 사이의 간격을 조정한다.

-위의 예시는 dilation rate가 2인 3x3 커널로 기존의(dilated rate를 사용하지 않는) 5x5 kernel과 동일한 view를 가지게 된다.

-이는 기존에 비해 상대적으로 작은 kernel을 가지고 더 넓은 receptive field를 커버할 수 있는 효과가 생긴다. 기존의 CNN에서는 receptive field를 확장하기 위해 pooling을 통해 input layer의 크기를 줄인 후 convolution을 했던 것과는 대조를 이룬다. 아래 그림을 통해 이를 이해할 수 있다. 아래의 예시에선 dilation을 통해 큰 receptive field를 유지하여 정보의 손실을 최소화 하고 해상도 높은 output을 얻는 것을 볼 수 있다.

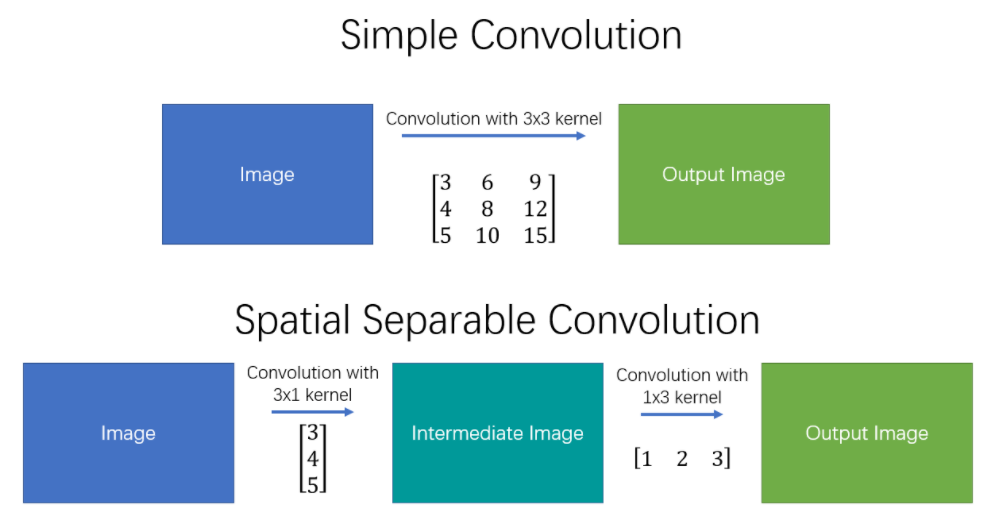


2. Separable Convolution

-기존의 convolution의 kernel을 두개의 작은 kernel로 나눈 후, 그 두개의 kernel을 가지고 원래의 큰 kernel로 작업했을 때와 동일한 효과를 거둘수 있게 한다.

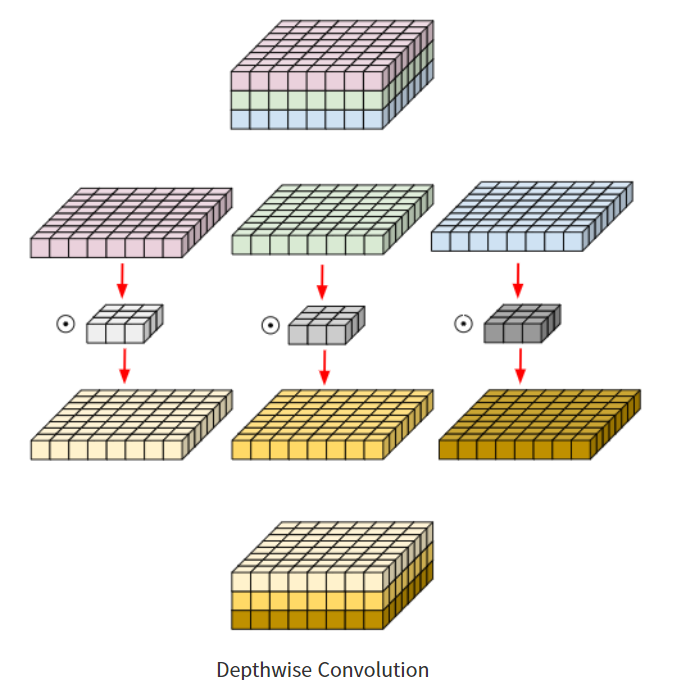
-각각의 곱셈 연산이 줄어들기에 계산의 복잡도가 줄어들고, 더 빠르게 convolution을 할 수 있게 되는 장점이 있다.

-아래 사진에서 보듯, 기존의 convolution에서는 3x3 커널을 활용한 2d convolution이 이뤄지는 반면, separable convolution은 두번의 1d convolution 연산을 통해 동일한 output을 산출시킨다.



3. Depthwise Convolution

-기존의 kernel은 input layer의 모든 channel에 대해 영향을 받게 되므로, 특정 channel만의 특징을 특정하기가 어려운 단점이 존재함. 하지만 depthwise convolution은 각 단일 channel에 대해서만 kernel을 입힌다고 생각하면 된다.

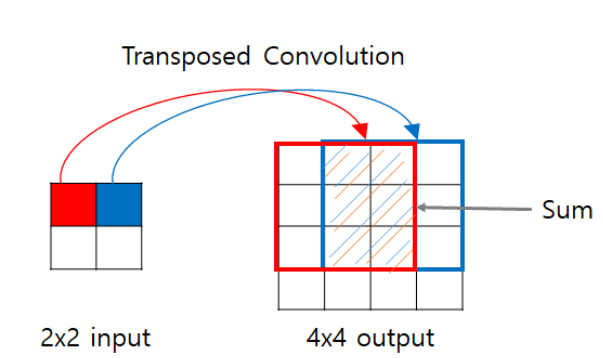


-위의 예시에서 8x8x3의 input layer를 8x8의 matrix로 분리하고, 각 분리된 matrix에 3x3의 2차원 kernel를 적용해 convolution을 적용한 뒤 3개의 ouput layer들을 하나로 합쳐 최종 output을 산출하는 것이 depthwise convolution의 작동 원리이다.

-따라서 각 kernel은 하나의 채널에만 대응되는 parameter를 가지게 되고, 그렇기 때문에 input channel과 output channel의 수가 동일해진다.

4. Transposed Convolution

-기존의 convolution 연산은 output의 크기가 input보다 작아지는 특성이 있는데, 이와는 반대로 output의 크기가 input의 크기보다 더 커지게 하는 방법을 upsampling이라 하며 그 예시로 transposed convolution이 있다.



-기존의 convolution의 input과 output을 바꿔서 생각해보자. Input의 빨간색 원소를 3x3의 kernel에 곱하여 output에 대응하는 자리에 넣고, input의 파란색 원소에도 이와 같은 연산을 적용하자. 이때 Output에 겹치는 구간의 값은 모두 더해준다. 나머지 input 원소에 대해서도 이 연산을 진행한다.

-이는 마치 convolution을 역으로 진행하는 것과 같아 보이지만, 엄연히 deconvolution과는 차이가 있다. 수학적으로 deconvolution은 정확하게 convolution의 역연산이 된다. 즉, 기존의 convolution layer를 거쳐서 얻은 feature map을 deconvolution할 시 원래의 image를 얻을 수 있다. Deconvolution은 기존의 convolution 연산에서 활용된 kernel과 output을 정확히 알고 있어야 하지만, transposed convolution에서 활용되는 kernel은 정확히 convolution layer와 대응되는 것이 아니다. 자세히는 transposed convolution은 학습을 통해서 kernel을 찾아간다는 차이가 있다고 한다.

Reference

-https://eehoeskrap.tistory.com/431

-https://realblack0.github.io/2020/05/11/transpose-convolution.html