



다양한 Convolutional 방법론

✨ 기존 Conv2d는 여러가지 문제점이 존재하여, 문제들을 해결하기 위해 연구자들을 다양한 Convolution기법들을 개발하였습니다.

기존 연구의 문제

- **Expensive Cost** [연산비용 상당히 큼]
- **Deat Channels** [의미없는 Channels (BottleNeck에서 주로 발생)]
- **Low Corrleation between Channels** [채널별 낮은 상관관계]

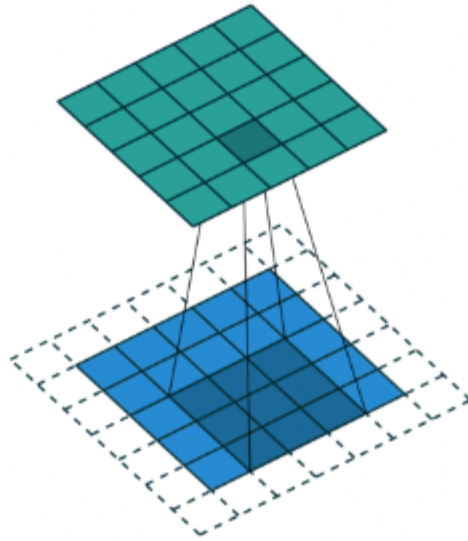
ComputerVision에서는 충분한 **Contextual Information**을 확보하기 위해 상대적으로 넓은 **Receptive Field**를 고려해야 합니다.

따라서, DL에서는 **Computational**을 **경량화**하면서, **Information Loss**가 발생하지 않도록 다양한 **Convolution mechanism**이 등장하였습니다.

Contents

- Conv2d
- Dilated Conv2d [atrous Deconvolution]
- Transposed Conv2d
- Separable Conv2d
- Depthwise Conv2d
- Pointwise Conv2d
- Grouped Conv2d
- Deformable Conv2d

1. Conv2d (Convolution)

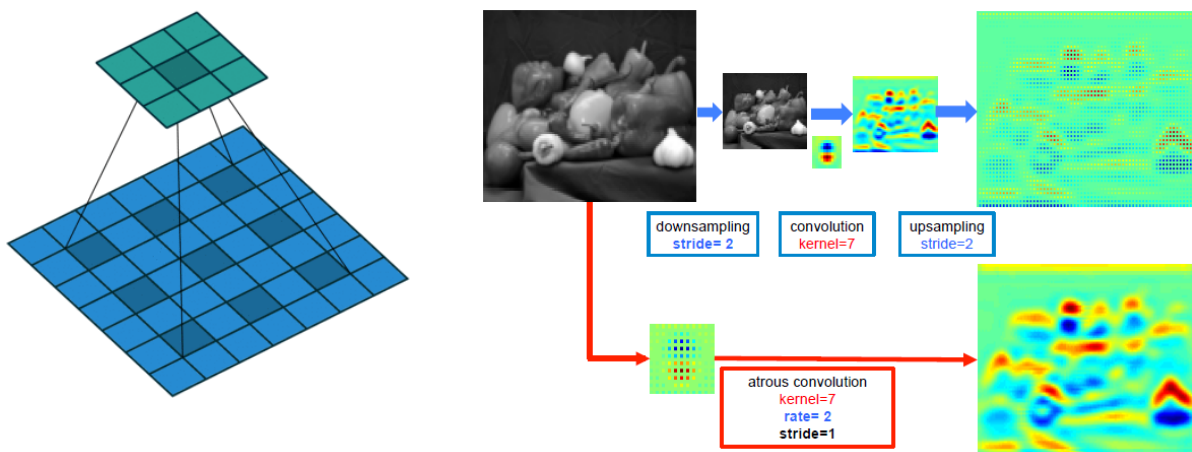


2D convolution using a kernel size of 3, stride of 1 and padding

가장 기본적인 형태의 Convolution입니다.

- **Kernel Size** : Conv2d의 View를 결정합니다. 보통 2차원의 3x3 Pixel을 사용합니다.
- **Stride** : Image를 scan할 때, Kernel의 Step size를 결정합니다. default는 1이지만, 보통 Max Pooling과 비슷하게 이미지를 Down Sampling하기 위해, Stride를 2로 사용한다.
- **Padding** : Padding은 Sample border를 어떻게 조절할지를 결정한다.
- **Input & Output Channels** : Conv2d layer는 **Input channels** → **Output channels**를 계산합니다.

2. Dilated Conv2d (atrous Convolution)



Dilated Conv2d는 기존 **Conv2d Filter**가 adapt하는 Pixel사이에 간격을 둔 형태를 의미합니다.

Input Pixel 수는 동일하지만, 넓은 범위에 대한 입력을 수용할 수 있습니다.

Dilated Conv2d은 특히 **real-time Segmentation**분야에서 주로 사용됩니다. 넓은View가 필요하고, 여러 **Conv2d**나 **large Kernel**을 사용할 여유가 없을 때 주로 사용합니다. 즉, 적은 계산 비용으로 **Receptive Field**를 늘리는 방법이라고 할 수 있습니다.

Dilated Conv2d는 **Filter** 내부에 **Zero Padding**을 추가하여, 수용 영역(**Receptive Field**)를 늘립니다.

Receptive Field는 **Filter**가 한번 보는 영역으로 **Image**의 **Feature**를 파악하고, 추출하기 위해서는 넓은 **Receptive Field**를 사용하는 것이 좋습니다.

Dimension은 손실이 적고, 대부분의 **Weight**가 0이기에 연산의 효율이 좋습니다. 공간적 특징을 유지하는 **Segmentation**에서 주로 사용됩니다.

☀ **Segmentation**문제에서는 **Dilated Conv2d**를 통해 이득을 얻는 경우를 볼 수 있습니다.

Dilated Conv2d의 그림을 보면, **Receptive Field**를 크게 가져가면서, 정보 손실(**Information Loss**)를 최소화하면서 해상도는 큰 **Output**을 얻을 수 있습니다.

3. Transposed Conv2d

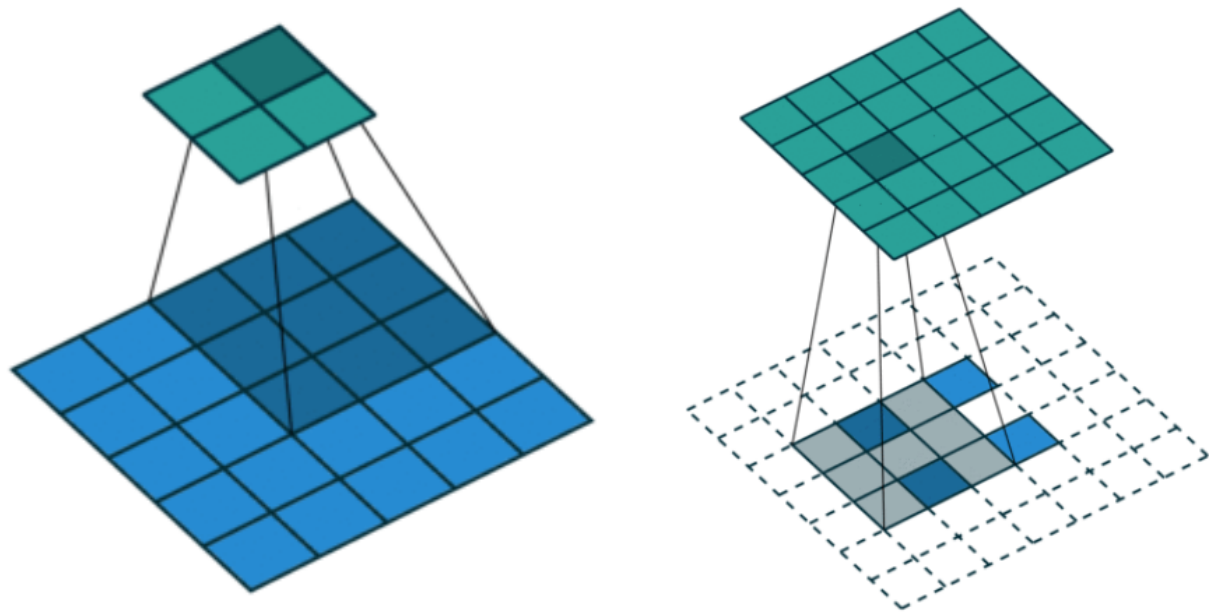
✨ **Deconvolution or fractionally strided Conv2d**

Transposed Conv2d보다는 **Deconvolution**이라는 이름으로 많이 사용됩니다. 하지만, 실제로는 **Deconvolution**이 아닙니다.

왜냐하면 실제의 **Decon2d**는 **Conv2d**의 과정을 되돌리는 작업입니다. 실제 진짜의 **Deconv2d**과정은 아래 그림과 같은데, 일반 **2D Conv2d** 그림에서 **5x5**의 입력 이미지가 있을 때,

Stride 2, Padding X, Kernel은 **3x3** 적용되었습니다. 이러한 결과 **2x2 image**가 생성되었습니다.

이 과정은, 실제 **Deconvolution** 작업은 역 수한 연산을 위해 입력값의 각 픽셀로부터 9개의 값을 뽑아야 하며, 그 후에 **Stride**가 2인 출력 이미지를 지나게 되지만, 실제로는 그렇지 않습니다.



Transposed Conv2d는 Deconvolutional layer와 동일한 공간 해상도를 생성한다는 점에서 유사합니다. 하지만, 실질적인 Computational 방식은 다릅니다.

Transposed Conv2d 연산 과정은 다음과 같이, 압축된 정보를 또 다른 과정을 통해 Up-Sampling하여 복원하는 것이다.

☀ CNN을 사용한 Encoder-Decoder구조에서 AutoEncoder가 있습니다. Encoder에서 Pooling등을 하여, 데이터를 압축하였다면,
그 데이터를 다시 복원하기위해 Up-Sampling해야하는데 이때 주로 사용합니다.

단순히 이미지의 해상도를 높이는 Super Resolution에서도 많이 사용됩니다. 이미지의 크기를 복원하는 방법은 Transposed Conv2d만 존재하는 것이 아닙니다.
“Bilinear Upsampling”을 사용하기도 합니다.

4. Separable Conv2d

Separable Convolution에는 Kernel작업을 여러 단계로 나눌 수 있습니다.

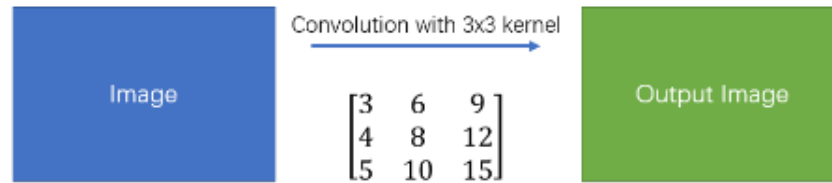
Conv2d를 $y = \text{Conv}(x, k)$ 로 표현할 때, X는 Input Image, Y는 Output Image, K는 Kernel이다.

그리고, $K = k1 \cdot \text{dot}(k2)$ 로 계산한다고 가정할 때, K와 2D Conv2d를 수행하는 대신 K1과 K2로 1D Convolution 하는 것과 동일한 결과를 가져오기 때문에, Separable Conv2d라고 할 수 있습니다.

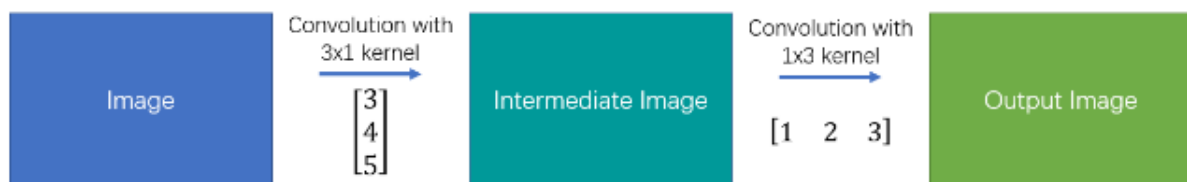
즉, Separable Convolution은 단순히 Kernel을 두 개의 작은 Kernel로 나눈 뒤 원래 Kernel로 작업한 결과물과 동일한 효과를 얻기 위하여, 3번의 곱셈으로 2번 Conv2d를 수행하게 됩니다.

곱셈 연산이 적을수록 복잡성이 줄어들고, 네트워크를 더 빠르게 실행가능합니다.

Simple Convolution



Spatial Separable Convolution



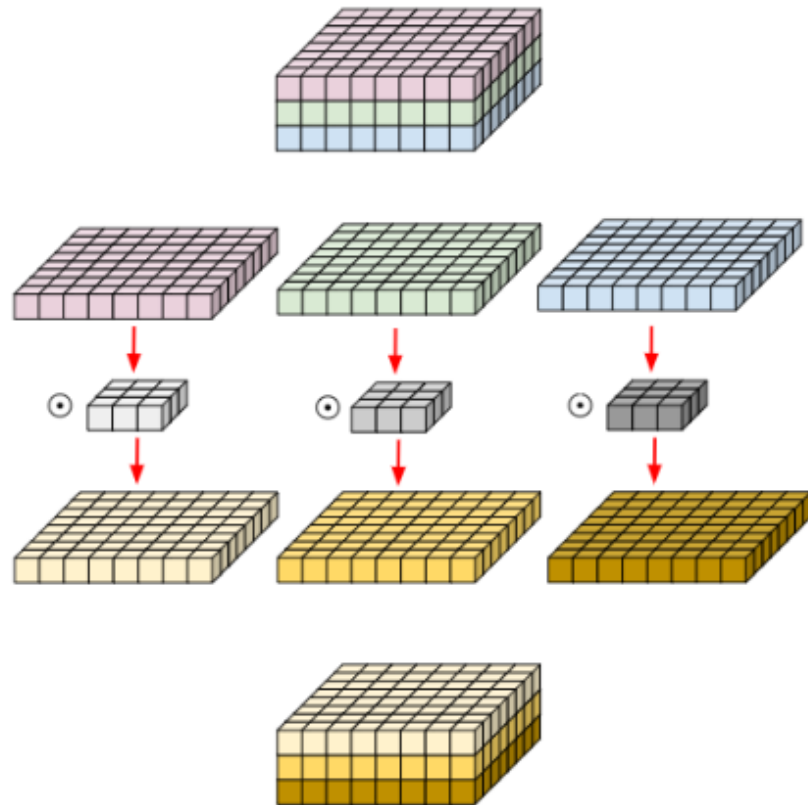
5. Depthwise Conv2d

일반적인 Conv2d Filter는 입력 영상의 모든 **Channels의 영향**을 받게 되므로, 완벽히 특정 채널만의 **Spatial Feature**를 추출하는 것이 **불가능**합니다.

즉, 다른 Channels의 정보가 관여하는 것이 불가피 하기 때문입니다.

하지만, **Depthwise Convolution**은 각 단일 채널에 대해서만 수행되는 Filter들을 사용하였습니다.

MobileNet구조에서는 **Depthwise Convolution**을 택하여 연산량을 기하 급수적으로 줄여 실시간에서 동작 가능하도록 하였습니다.



Depthwise Convolution

위 그림과 같이, 3x8x8 matrix를 Depthwise Convolution하기 위해, 3x3x3 커널을 사용한 예시입니다.

이 때, 해당 Kernel은 각 Channel별로 분리하여, 3x3의 2차원 Kernel을 8x8크기의 분리된 Matrix에 붙여 각 Convolution을 진행하고 다시 합쳐지는 구조로 되어있습니다.

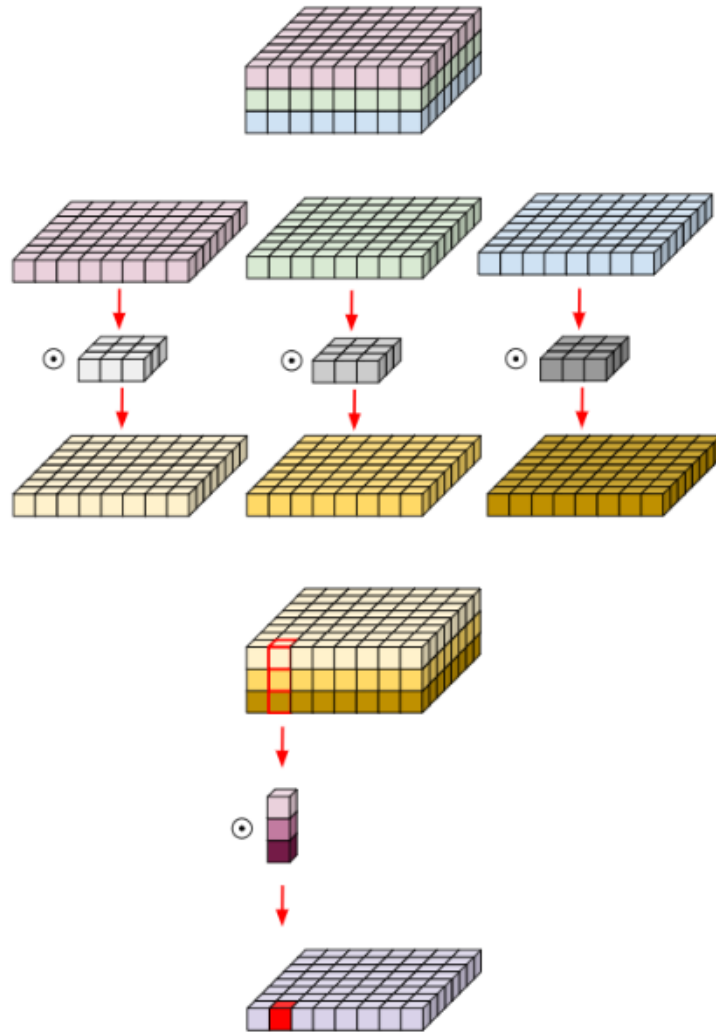
즉, 채널방향의 Convolution은 진행하지 않고, 공간 방향의 Convolution만을 진행하였습니다.

즉, 각 Kernel들은 하나의 Channel에 대해서만 parameter를 가집니다. 그래서 입력 및 출력 채널의 수가 동일한 것이며 각 채널 고유의 Spatial 정보만을 사용하여 필터를 학습하게 됩니다. 결과적으로 입력 채널 수 만큼 그룹을 나눈 Grouped Convolution과 같아집니다.

6. Depthwise Separable Conv2d

위의, Depthwise Convolution과 다르게, 채널의 출력 값이 하나로 합쳐지는 특징을 가지고 있습니다.

이는, Spatial Feature와 Channel-wise Feature를 모두 고려하여 네트워크를 경량화하는 기법입니다.



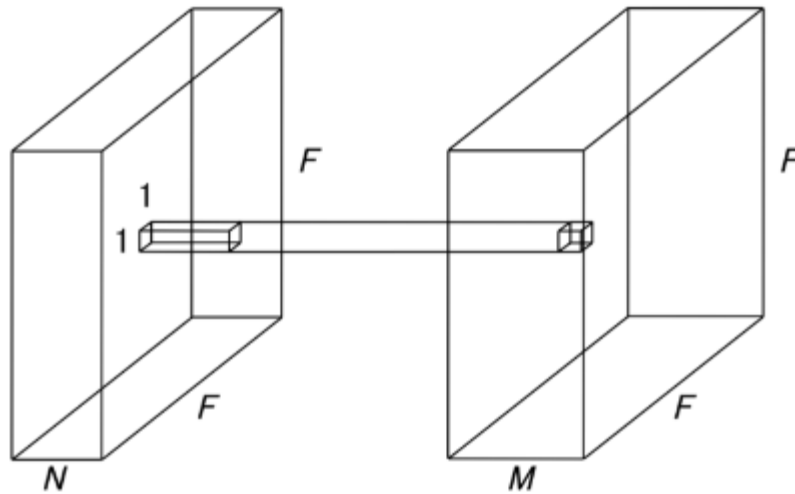
기존 Depthwise Convolution을 진행한 결과물에 각 채널을 1개의 채널로 압축 할 수 있는 추가적인 Convolution을 진행하여 결과물이 굉장히 간략하게 나옵니다.

7. Pointwise Conv2d

앞서 소개한, **Depthwise separable Conv2d**에서 Depthwise Conv2d 적용 후, 적용된 기법입니다.

Pointwise Conv2d는 공간방향의 Conv2d는 진행되지 않고, Channel방향의 Conv2d를 진행합니다.

즉, **Channel Reduction** 시 주로 사용합니다.



즉, $1 \times 1 \times C$ 크기의 Kernel을 사용하며, 입력의 특징을 1개의 채널로 압축하는 효과를 가지고 있습니다.

결국 하나의 Filter는 Channel별로 Coefficient를 가지는 Linear Combination을 표현한다. 이는 채널 단위의 Linear Combination을 통한 채널 수의 변화가 가능하다는 것을 의미합니다.

다채널 입력 영상을 더 적은 채널의 영상으로의 Embedding하는 것으로 이해할 수 있습니다.

출력 채널 수를 줄임으로써, 다음 레이어의 계산량과 파라미터를 줄일 수 있는 것입니다.

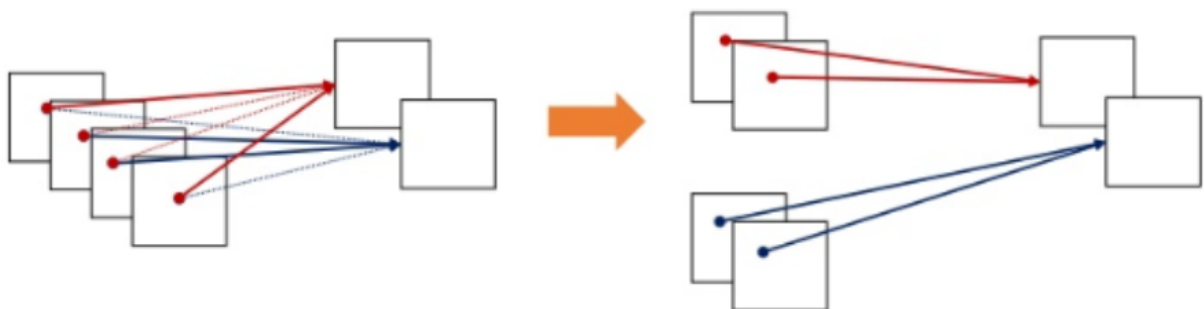
채널에 대한 Linear Combination을 수행 시

불필요한 채널들이 낮은 Coefficient를 가지며 연산결과에서 희석될 수 있습니다.

✨ Pointwise Conv2d는 Inception / Xception / SqueezeNet / MobileNet에 적용되어 경량화 및 성능 개선이 가능함을 실험적으로 증명되었습니다.

8. Grouped Conv2d

Grouped Conv2d는 입력 값의 Channel들을 여러 개의 group으로 나누어 독립적으로 Conv2d 연산을 수행하는 방식입니다. (아이디어와 구현 방법이 간단하며 병렬 처리에 유리하다는 장점)



Grouped Conv2d 기법은 기존 2D Conv2d보다 낮은 파라미터 수와 연산량을 갖게 되며, 각 그룹에 높게 Correlation을 가지는 채널이 학습될 수 있다는 장점이 있습니다.

★ 독립적인 학습이 가능하다는 것이 장점입니다.

하지만, 그룹의 수는 Hyper Parameter라는 단점이 있기에 너무 많은 그룹으로 분할하는 것은 성능 하락이 될 수 있습니다.

9. Deformable Conv2d

명칭 그대로, **변형이 가능한 CNN이라는 의미**를 가지고 있습니다.

기존 CNN에서 사용하는 여러 가지 연산은 일정한 패턴을 가정하기에 복잡한 transformation에 유연하게 대처하기 어렵다는 지적에서 나타났습니다.

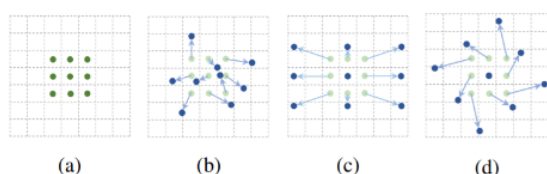


Figure 1: Illustration of the sampling locations in 3×3 standard and deformable convolutions. (a) regular sampling grid (green points) of standard convolution. (b) deformed sampling locations (dark blue points) with augmented offsets (light blue arrows) in deformable convolution. (c)(d) are special cases of (b), showing that the deformable convolution generalizes various transformations for scale, (anisotropic) aspect ratio and rotation.

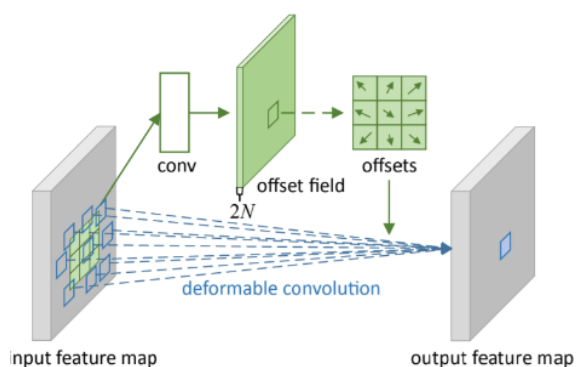


Figure 2: Illustration of 3×3 deformable convolution.

Deformable Conv2d는 Conv2d에서 사용하는 Sampling Grid에 2Doffset을 추가한 것입니다.

[Deep Learning] 딥러닝에서 사용되는 다양한 Convolution 기법들

또한, 영상 내의 객체에 대한 정확한 판단을 위해서는 Contextual Information 이 중요하다. 가령, 객체 주변의 배경은 어떠한 환경인지, 객체 주변의 다른 객체들은 어떤 종류인지 등. Object Detection 이나 Object

👉 <https://eehodeskrap.tistory.com/431>

