CNN

이미지 인식

슈퍼비전 우승

Convolutional neural network

- 컨볼루션: 정보를 섞는다는 개념(ex. 내적)

인풋이미지, 커널, 피쳐맵

Input image

Convolution Kernel

$$+ \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} =$$

Feature map

<채널당 1개>

CNN의 목적 : feature engineering 을 CNN이 대신 처리해주

Convolutional theorem

Continuous

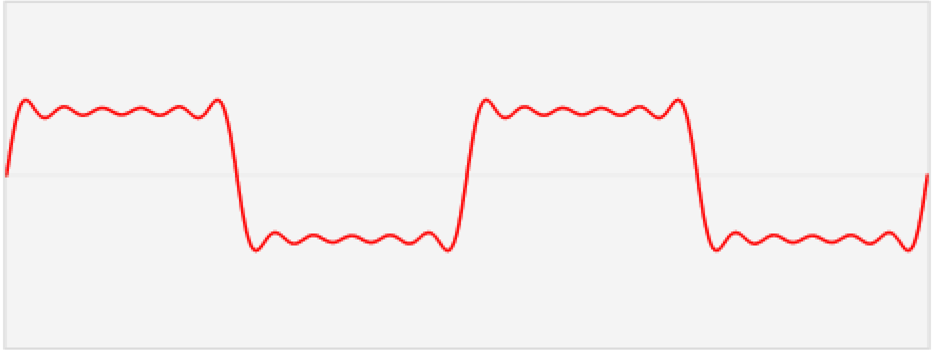
F: fourier transform

$$h(x) = f \otimes g = \int_{-\infty}^{\infty} f(x-u)g(u)\,du = \mathcal{F}^{-1}\left(\sqrt{2\pi}\mathcal{F}[f]\mathcal{F}[g]\right)$$

$$\text{feature map} = \text{input} \otimes \text{kernel} = \sum_{y=0}^{\text{columns}} \left(\sum_{x=0}^{\text{rows}} \text{input}(x-a, y-b)\,\text{kernel}(x,y) \right) = \mathcal{F}^{-1}\left(\sqrt{2\pi}\mathcal{F}[\text{input}]\mathcal{F}[\text{kernel}]\right)$$

Discrete

Fast Fourier Tranfrom



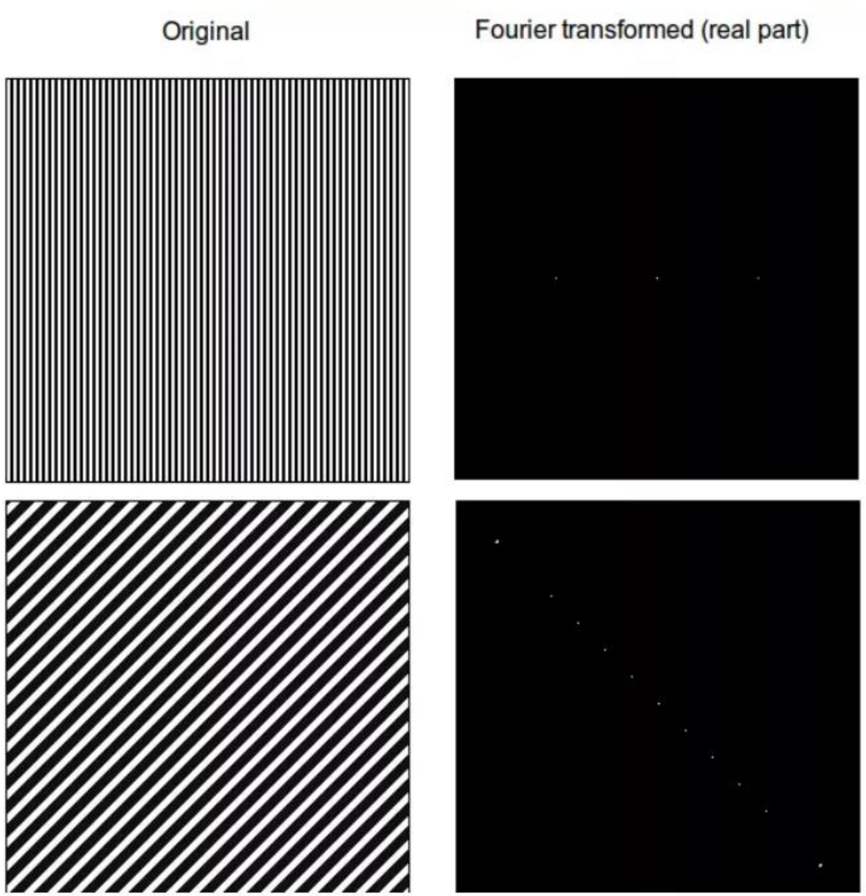
슬라이드로 동영상보기

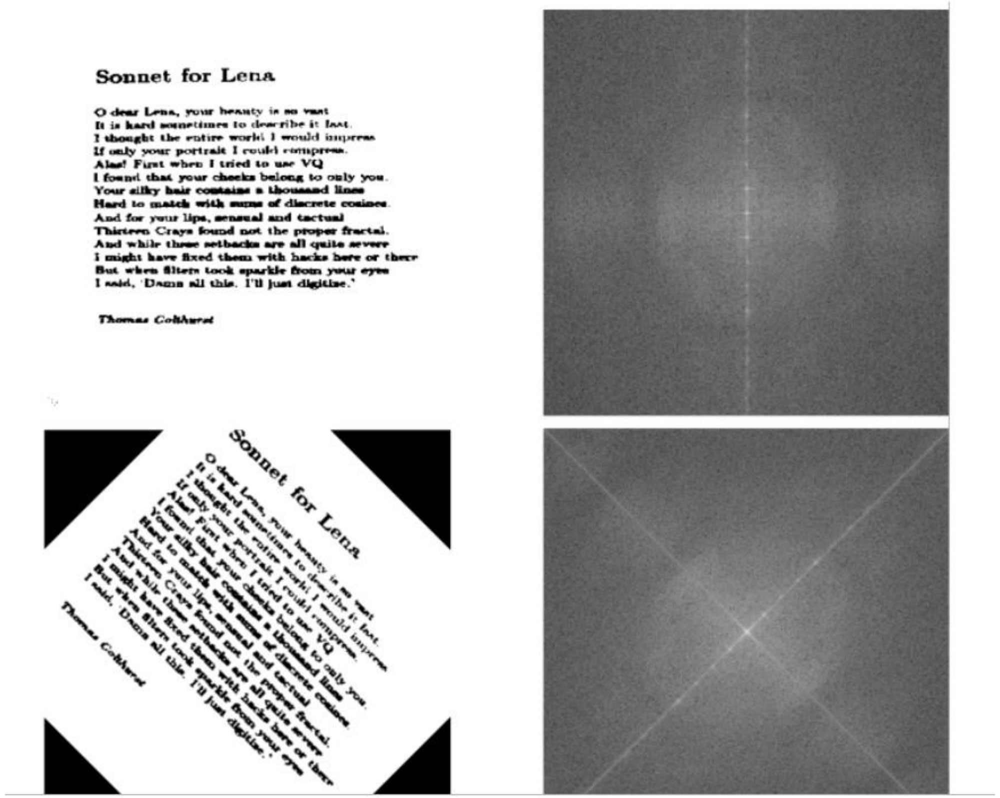
Transformation of the time domain (red) into the frequency domain (blue).Source

If the red signal is a song then the blue values might be the equalizer bars displayed by your mp3 player.

출처: http://hamait.tistory.com/536 [HAMA 블로그]

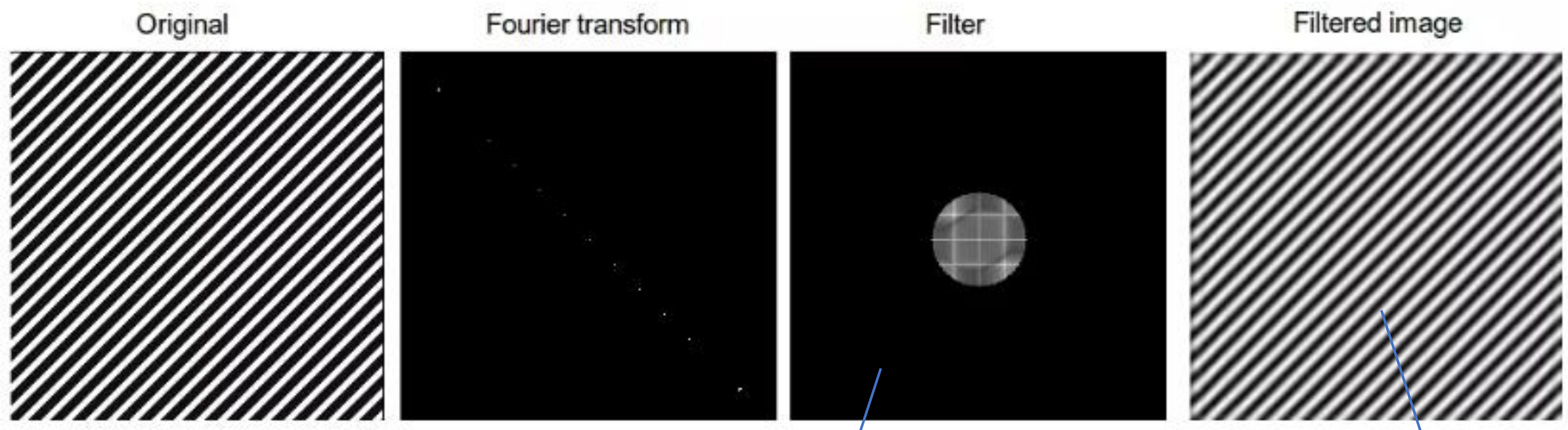
The Fourier domain for images





Frequency filtering and convolution

input Kernel,filter output



Images by Fisher & Koryllos (1998). Source

Zero padding

High frequency illiminated

Insights from fluid mechanics

- Fourier transforms not only simplify convolution, but also differentiation

- Fourier transforms are widely used in the field of fluid mechanics

Diffusion

- Convolution = diffusion process (propagator = pdf) = change color of pixel

: which direction fluid particles diffuse over time

- Intensity of pixel = Salt concentration

Interpreting the propagator

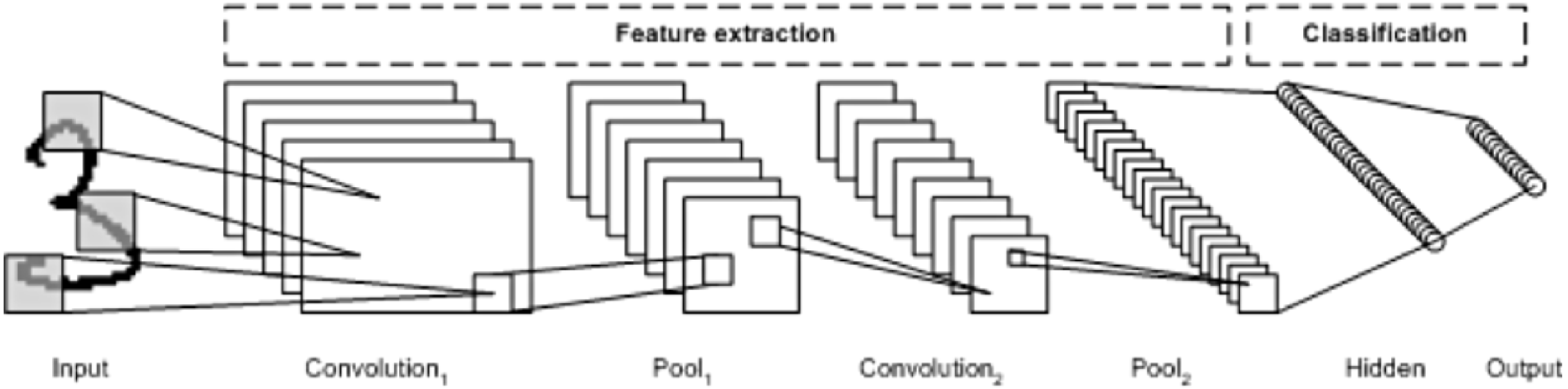
Propagator(probability density function) 이 diffusion procces를 수행함

-pdf를 곱한다는 데서 확산의 의미

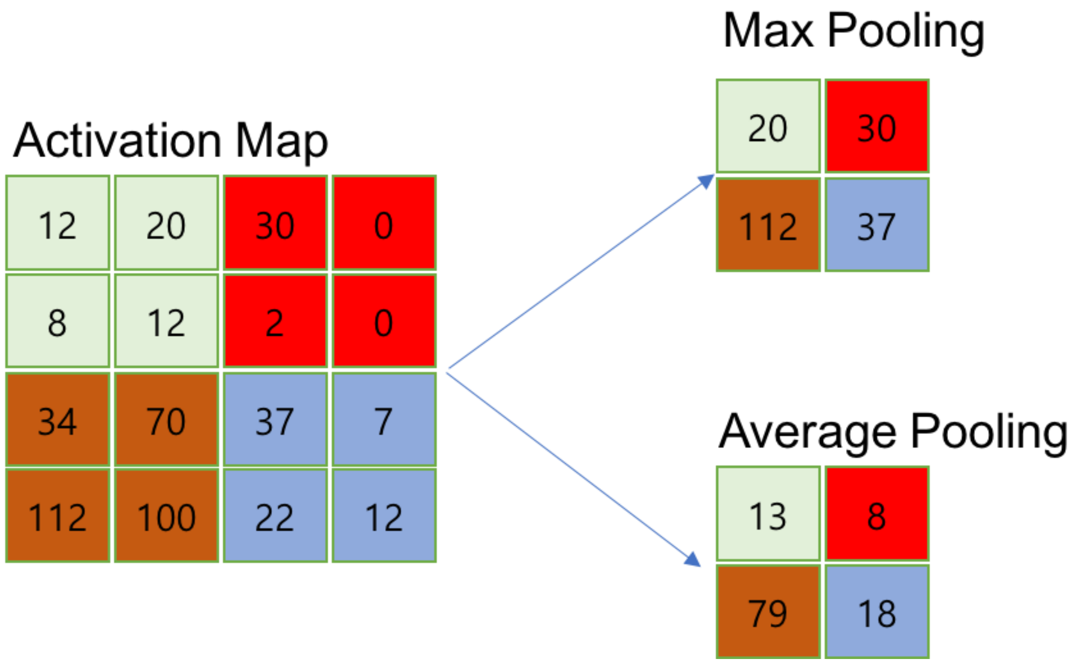
-(softmaxed)kerne을 곱한다는 데서 classification의 의미

Propagator = pdf = softmaxed normalization of kernel

$$Z = \mathrm{softmax}\left[\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}\right] = \begin{pmatrix} 0.0001 & 0.0001 & 0.0001 \\ 0.0001 & 0.9992 & 0.0001 \\ 0.0001 & 0.0001 & 0.0001 \end{pmatrix}$$

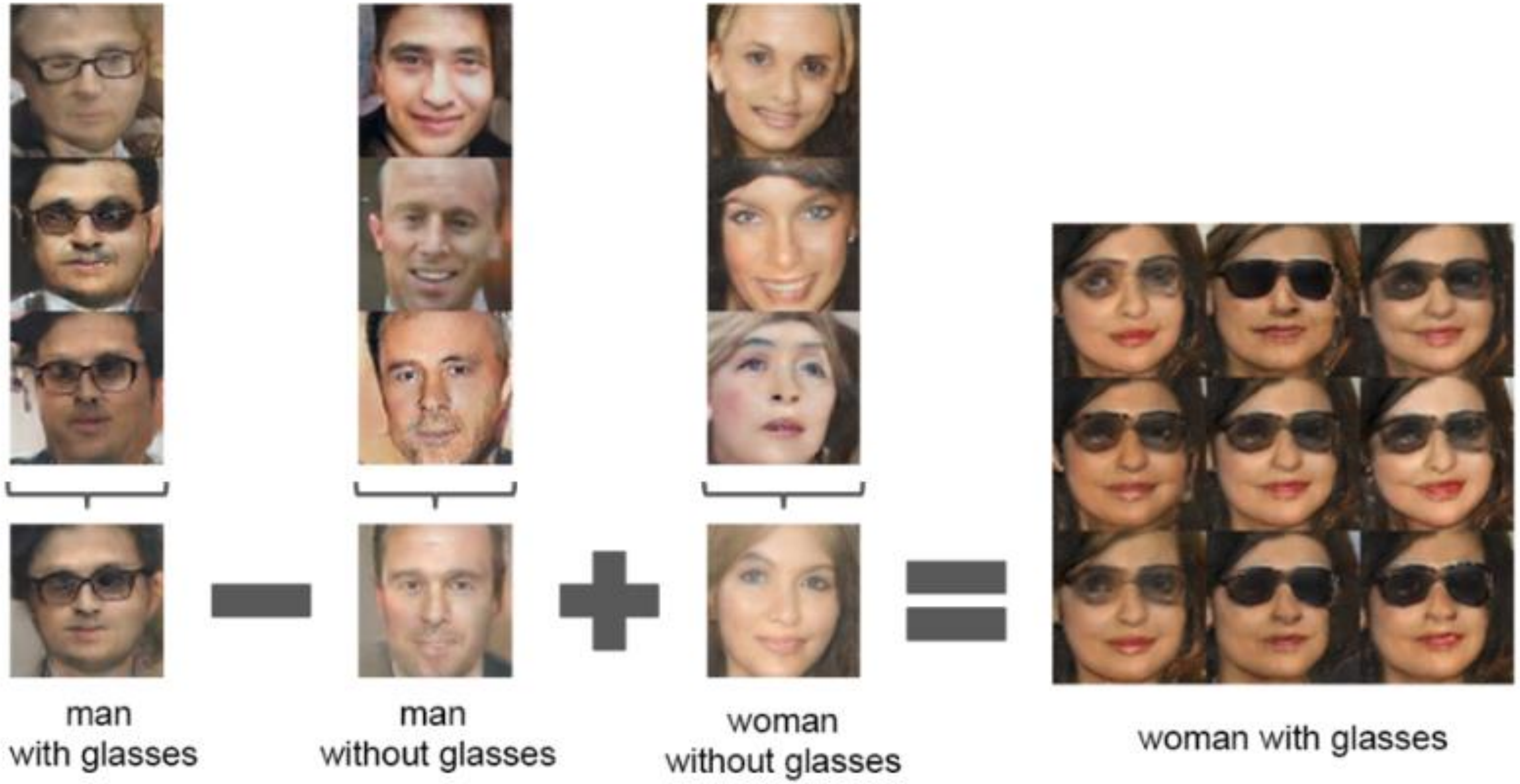

Feature extraction
Classification
Input
Convolution$_1$
Pool$_1$
Convolution$_2$
Pool$_2$
Hidden
Output

Pooling



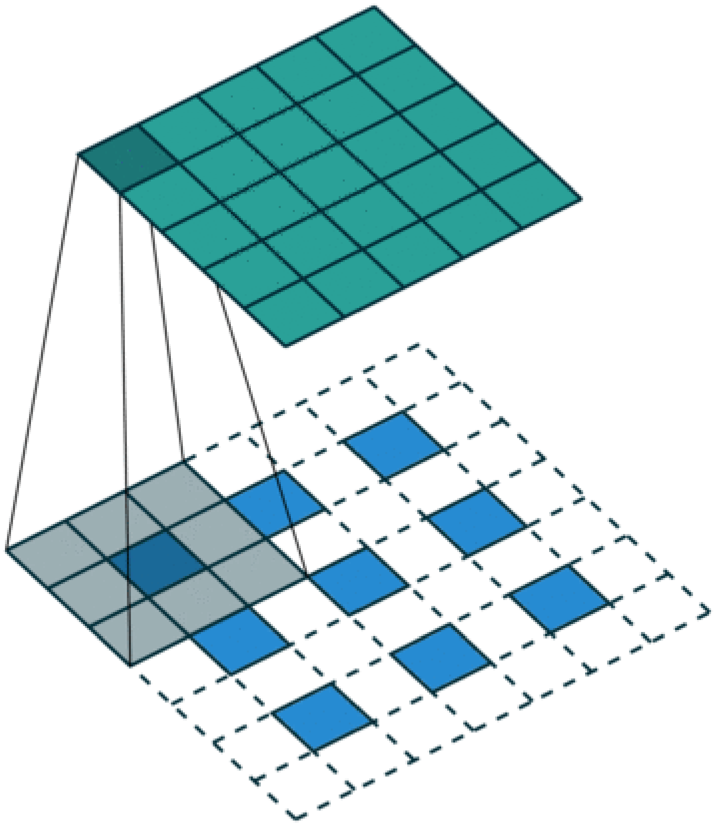
Activation Map
12 20 30 0
8 12 2 0
34 70 37 7
112 100 22 12
Max Pooling
20 30
112 37
Average Pooling
13 8
79 18

DCGAN



<input average>

fractionally-strided convolution



슬라이드 동영상

3x3 input을 upsampling 한후 convolution
-> 5x5 output

=Full convolution = upconvolution

Architecture guidelines for stable Deep Convolutional GANs

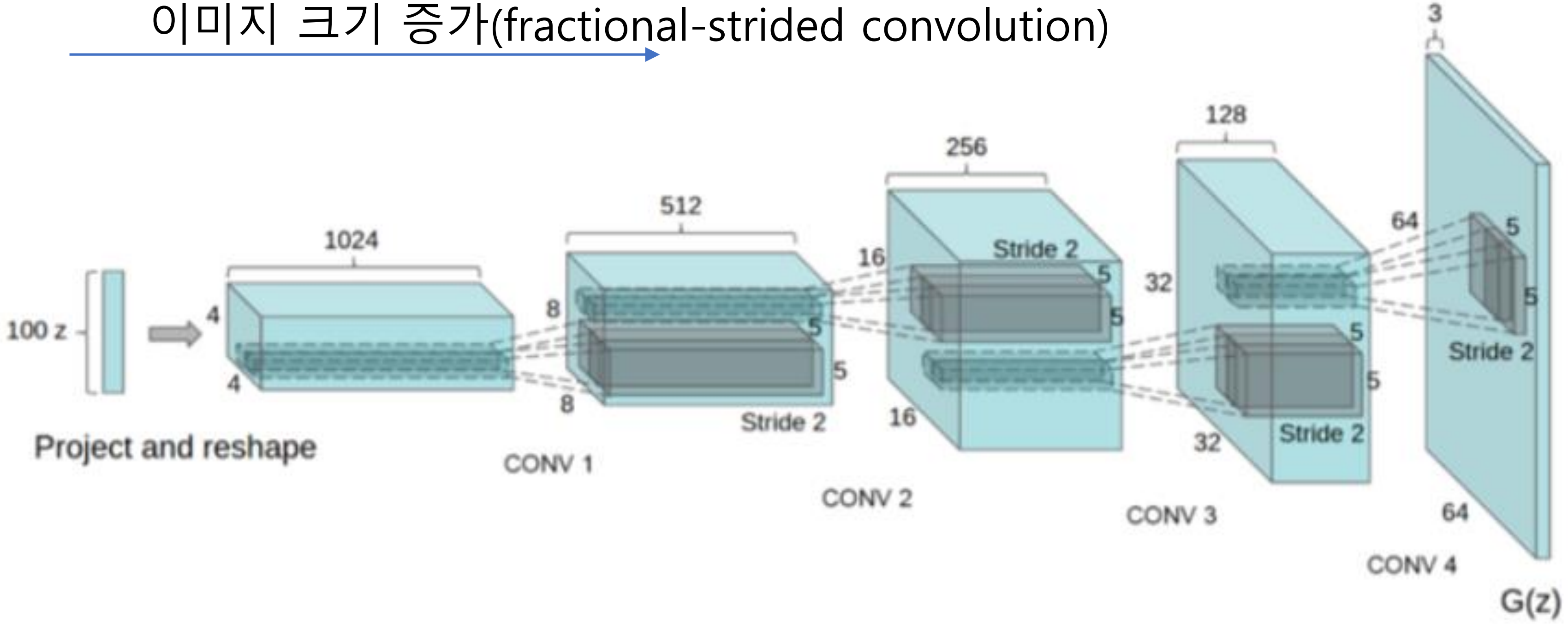
- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

1.Max-Pooling layer를 없애고, strided convolution이나 fractional-strided convolution 을 사용하여 feature map 조절.
2. Batch normalization 적용
3. Fully connected hidden layer 제거
4. Generator의 출력단의 활성함수로 Tanh를 사용, 나머지 layer는 ReLU사용
5. Discriminator의 활성함수로 LeakyReLU를 사용.

참고. 일반적인 strided convolution에서 stride값을 1 이상으로 하면 이미지 크기를 줄이지만 , 1이하의 분수로 하면 (fractional-strided convolution) 이미지 크기 증가.

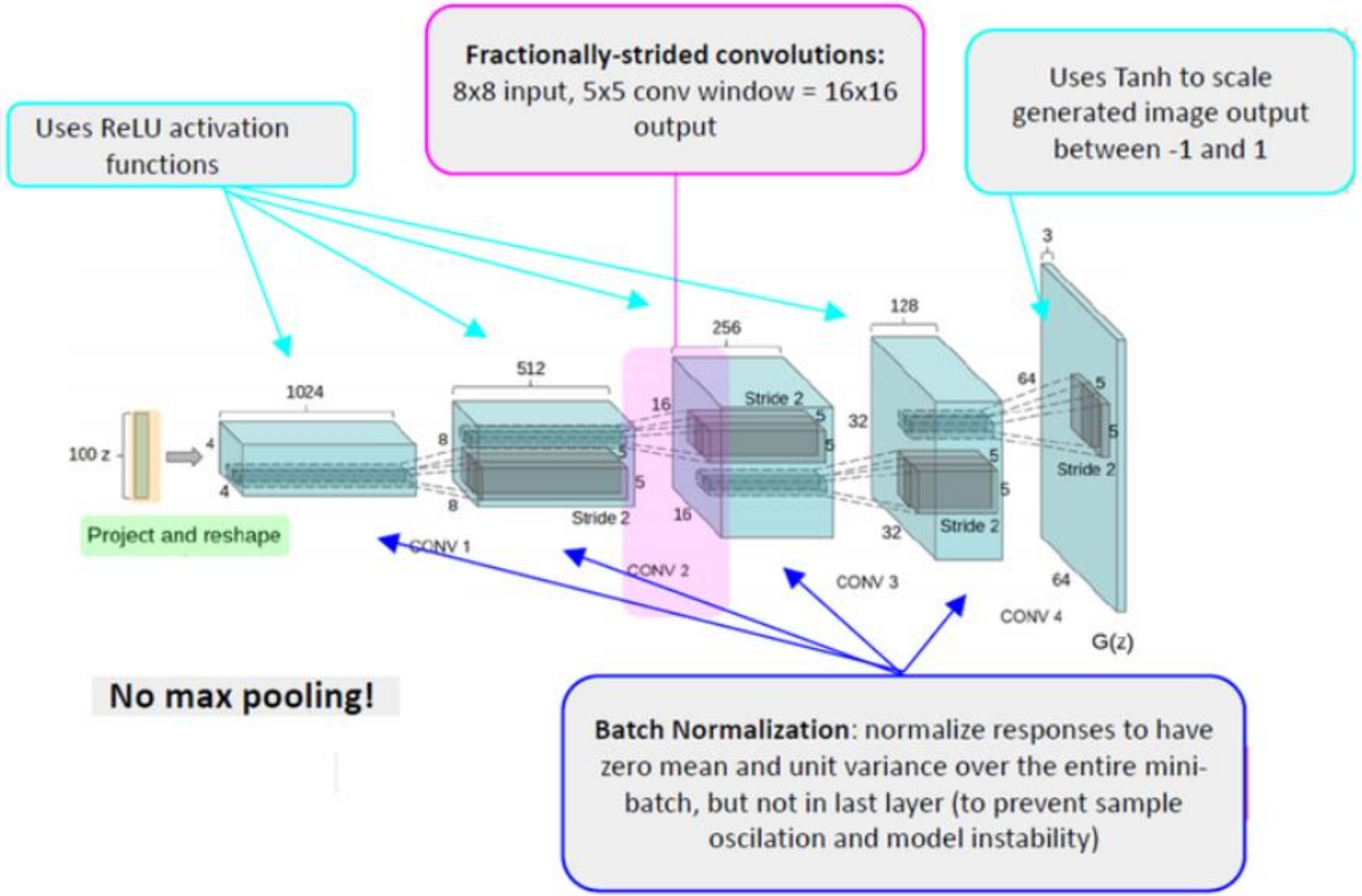
Generator

이미지 크기 증가(fractional-strided convolution)

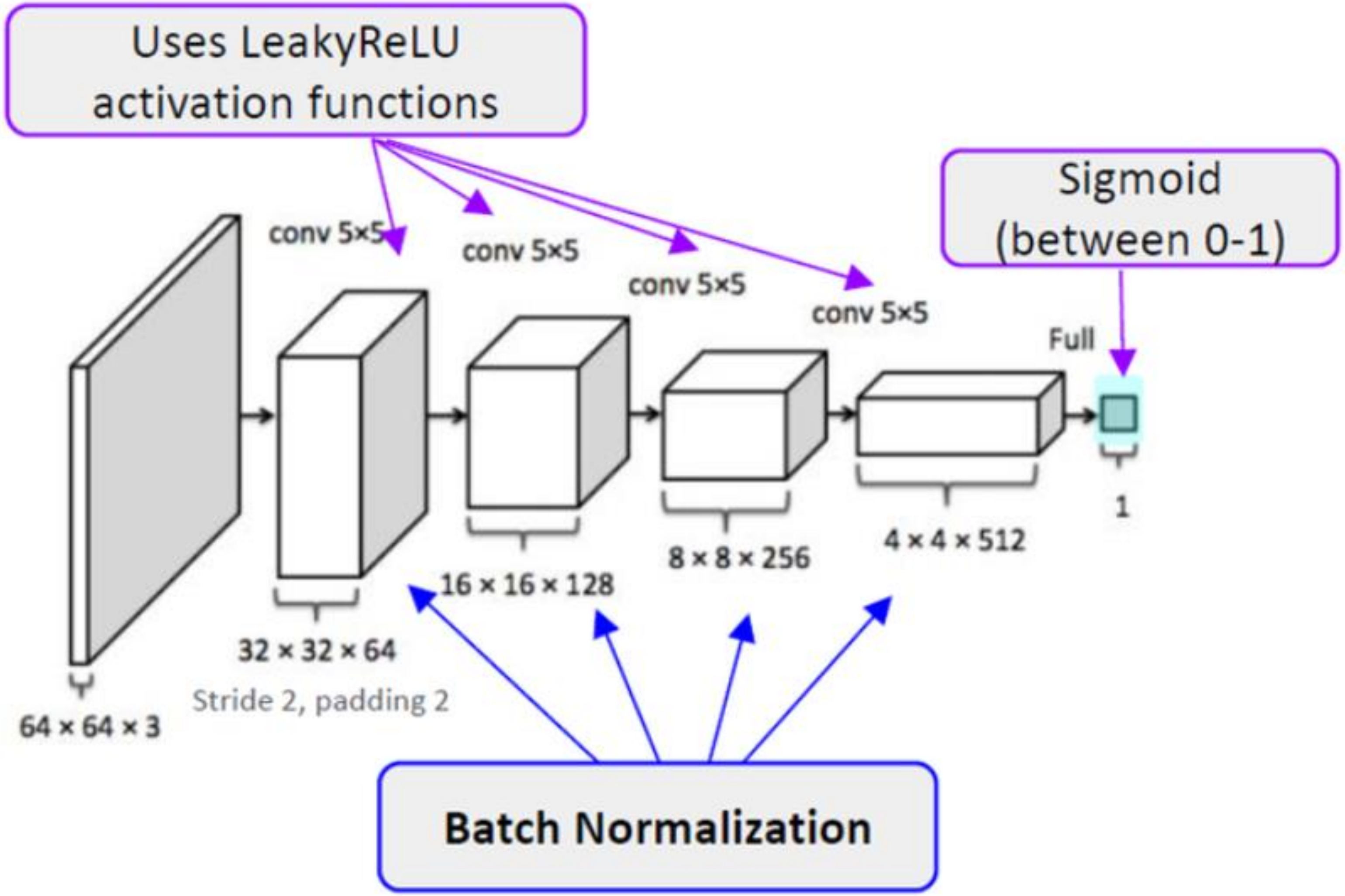


노이즈 -> Discriminator에서 사용할 컬러 이미지

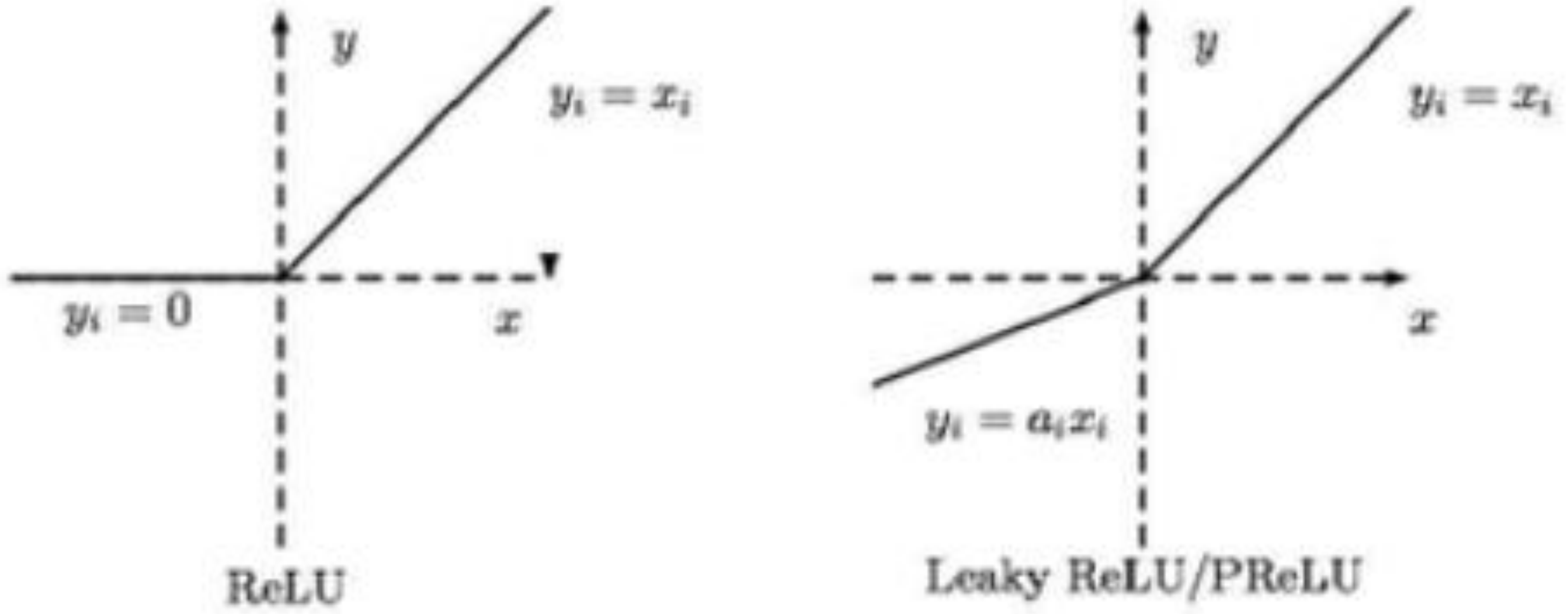
Generator


Fractionally-strided convolutions:
8x8 input, 5x5 conv window = 16x16 output
Uses ReLU activation functions
Uses Tanh to scale generated image output between -1 and 1
100 z
Project and reshape
1024
4
4
512
8
8
Stride 2
CONV 1
256
16
16
5
5
Stride 2
CONV 2
128
32
32
5
5
Stride 2
CONV 3
3
64
64
5
5
Stride 2
CONV 4
G(z)
No max pooling!
Batch Normalization: normalize responses to have zero mean and unit variance over the entire mini-batch, but not in last layer (to prevent sample oscillation and model instability)

Discriminater



Leaky ReLU



값이 음수가 나오면 0 이되어 더 이상 학습 되지 않는 문제를 해결