VGGnet

- 3*3 Convolutional filter
- Stride=2인 2*2 MaxPooling layer
- 마지막 3개의 FC layer로 classifier 구성
- Layer의 개수를 증가시킬수록 성능 증가

- 단일 7*7 Conv layer 대신 3개의 3*3 Conv layer
  -> 적은 parameter, non-linearity 증가

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 3: **ConvNet performance at a single test scale.**

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
A	256	256	29.6	10.4
A-LRN	256	256	29.7	10.5
B	256	256	28.7	9.9
C	256	256	28.1	9.4
	384	384	28.1	9.3
	[256;512]	384	27.3	8.8
D	256	256	27.0	8.8
	384	384	26.8	8.7
	[256;512]	384	25.6	8.1
E	256	256	27.3	9.0
	384	384	26.9	8.7
	[256;512]	384	**25.5**	**8.0**

ResNet

- NN이 깊어질수록 성능은 좋아지는가? - Degradation Problem
- Residual learning framework - Vanishing gradient문제 해결
- Linear projection으로 dimension 조절
- Bottleneck Design으로 연산량 감소 (VGG16 > ResNet-152)

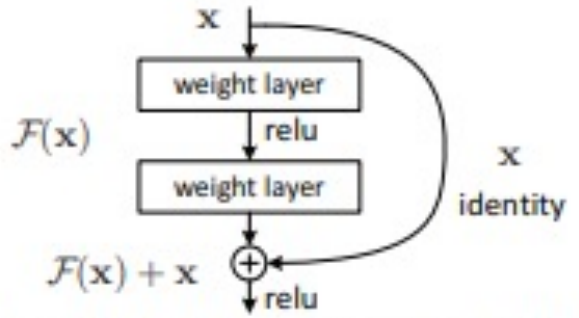
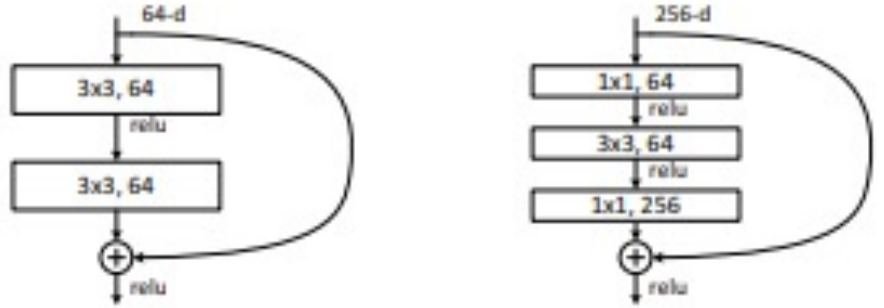


Figure 2. Residual learning: a building block.

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + W_s\mathbf{x}.$$

	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

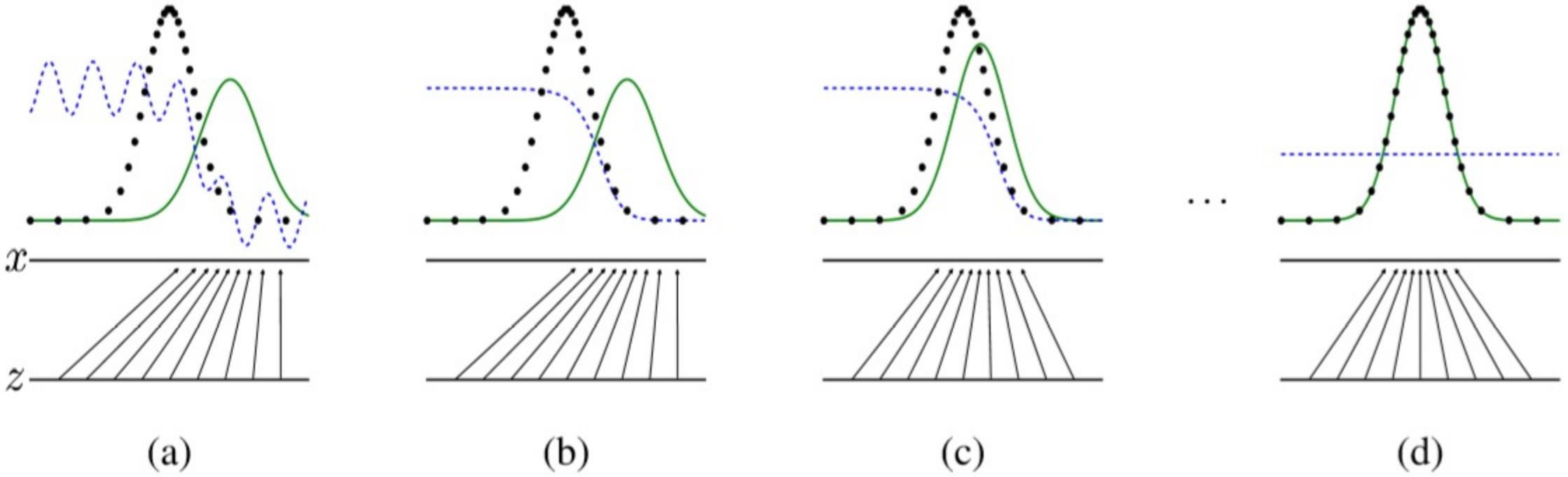
Table 2. Top-1 error (%, 10-crop testing) on ImageNet validation. Here the ResNets have no extra parameter compared to their plain counterparts. Fig. 4 shows the training procedures.



GAN

- 경쟁하는 과정을 통해 Generative model을 추정하는 새로운 framework
- 2개의 모델 학습(Generative model, Discriminative model)
- Generative model, G : training data의 분포를 묘사하여 D가 구별하지 못하도록 함
- Discriminative model, D: Sample data가 G에서 나온 데이터가 아니라 실제 training data에서 나왔을 확률을 계산
- G는 D가 실수할 확률을 최대화 하려고 한다 -> Minimax two-player game
- D는 V(D,G)를 최대화 시키려 하고 G는 V(D,G)를 최소화 시키려고 한다 -> 균형을 잘 이뤄야함

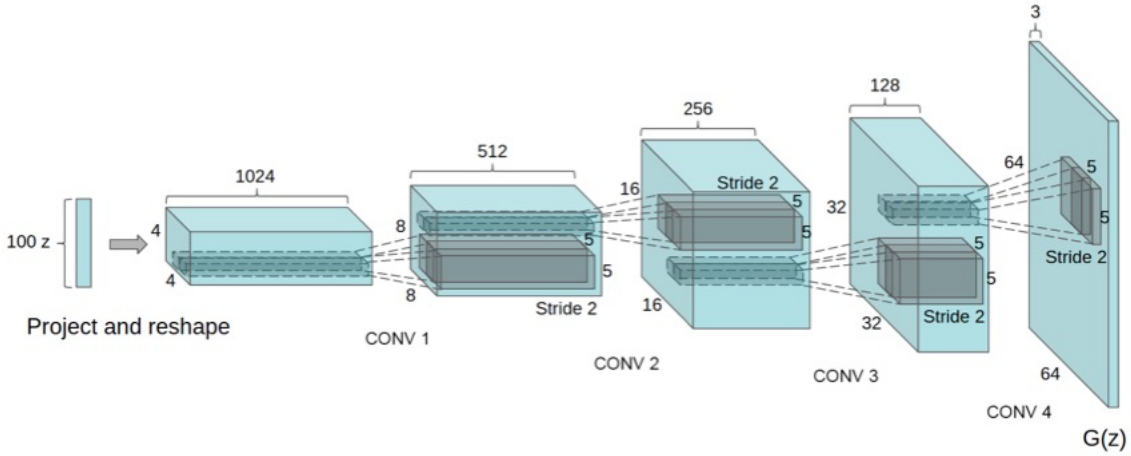
$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$



Blue : D | Green : G | Black : Training data

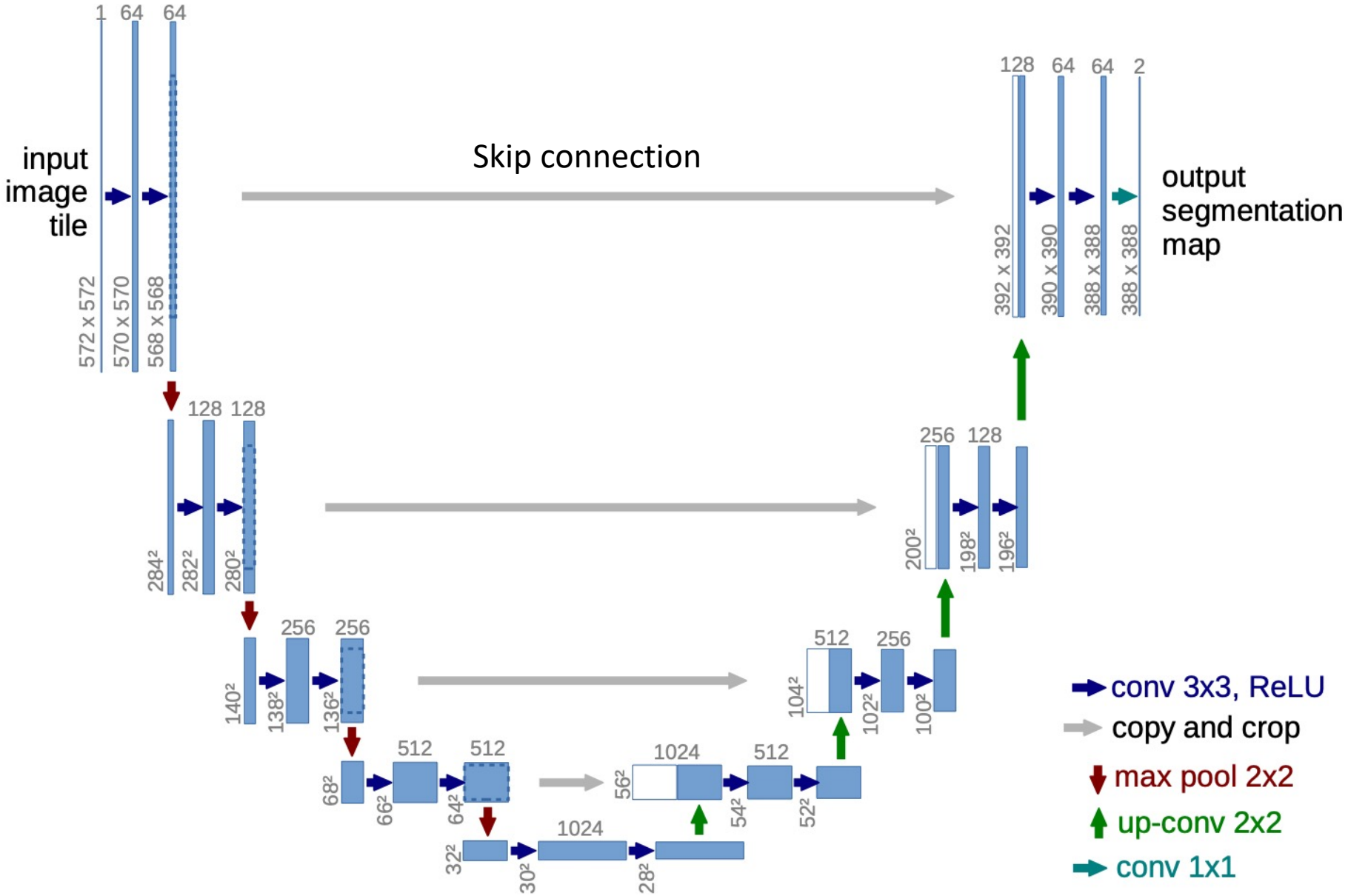
DCGAN(Deep Convolutional GAN)

- 기존 GAN의 불안정성을 크게 개선
- 기존 GAN : FC layer -> DCGAN : Convolution layer
- D : strided convolutions, LeakyReLU
- G : fractional-strided convolutions, ReLU, Tanh(output layer)



U-Net

- Encoder(Contracting path)-Decoder(Expanding path) 구조
- Contracting path : Convolutional layer
- Expanding path : Up-sampling(Transposed Convolution), Skip connection
- 다양한 학습 방법 사용
- Patch 선택 시 Overlap 비율이 적어 속도 상승
- Patch size에 따른 context 추출과 Localization간의 trade-off 개선


input image tile
1 64 64
572 x 572
570 x 570
568 x 568
Skip connection
128 64 64 2
392 x 392
390 x 390
388 x 388
388 x 388
output segmentation map
128 128
284²
282²
280²
256 128
200²
198²
196²
256 256
140²
138²
136²
512 256
104²
102²
100²
512 512
68²
66²
64²
1024 512
56²
54²
52²
1024
32²
30²
28²
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

U-Net

- Overlap-tile strategy : 이미지 크기가 큰 경우 이미지를 잘라야 하고 Input보다 Output이 이미지 크기가 더 작으므로 겹치는 부분이 존재하도록 이미지를 잘라야 함
- Mirroring Extrapolate : Padding을 이미지 경계에 대해 좌우 반전한 Mirror 이미지로 사용.
- Weighted Loss : 사전에 Ground-Truth에 대한 Weight map을 구해 학습에 반영함.
- Data Augmentation : Elastic deformation(Affine transform + probabilistic spin)

출처 : https://url.kr/76jl8x