

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
maxpool					
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	conv3-256	conv3-256 conv1-256	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	conv3-512	conv3-512 conv1-512	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	conv3-512	conv3-512 conv1-512	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 gradient 문제 해결
- Linear projection으로 dimension 조절
- Bottleneck Design으로 연산량 감소 (VGG16 > ResNet-152)

	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.

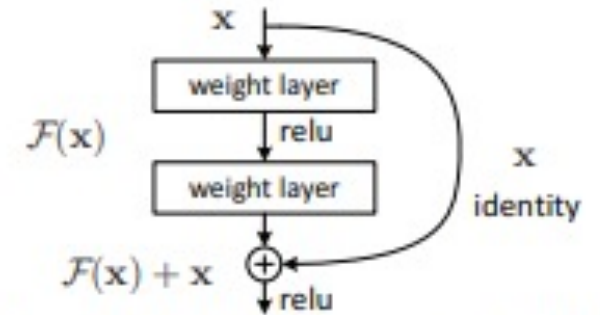


Figure 2. Residual learning: a building block.

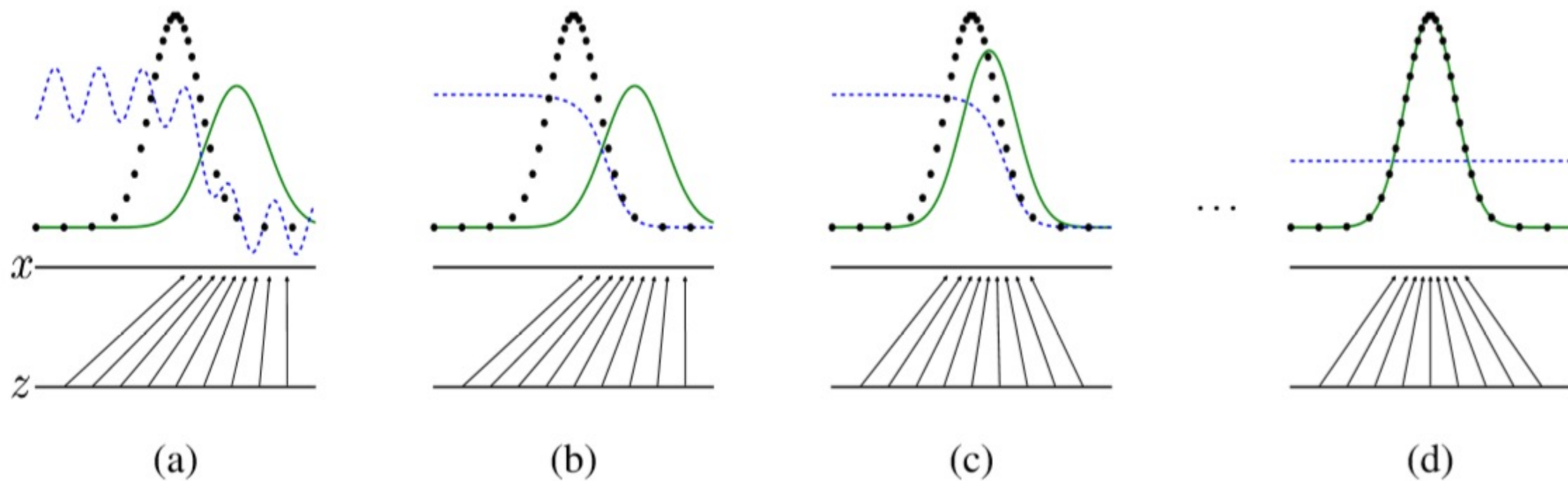
$$y = F(x, \{W_i\}) + W_s x.$$



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

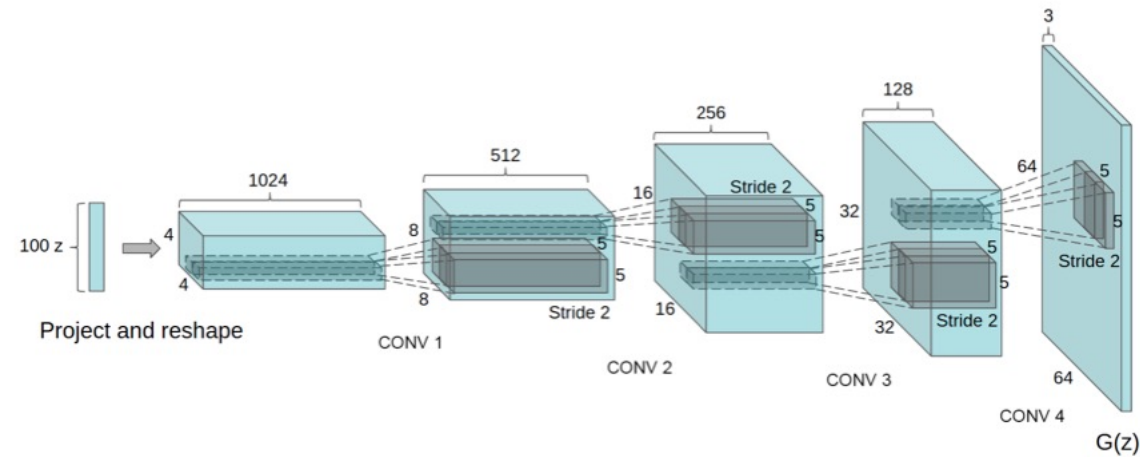
$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))]$$



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

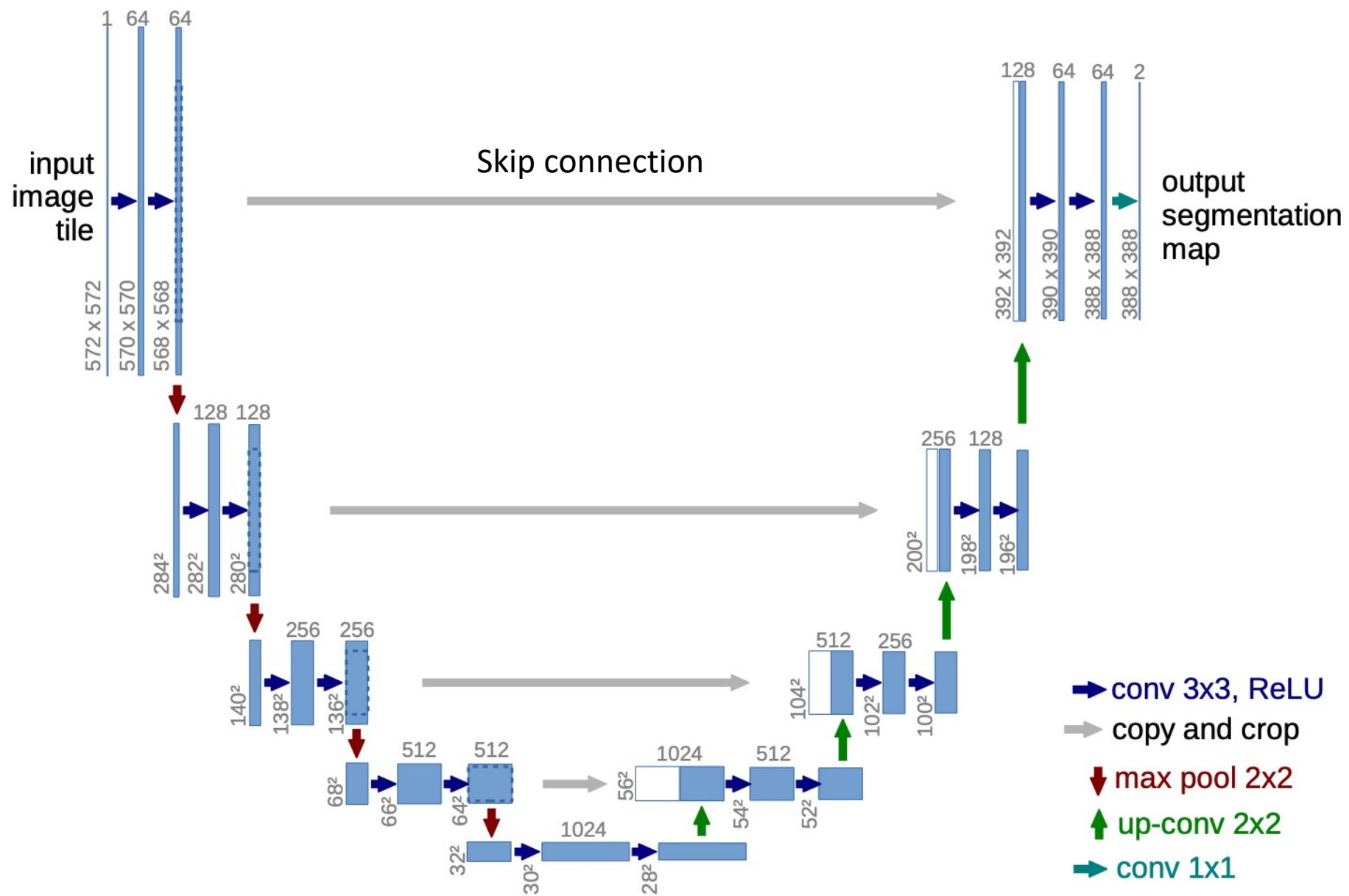
DCGAN(Deep Convolutional GAN)

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



U-Net

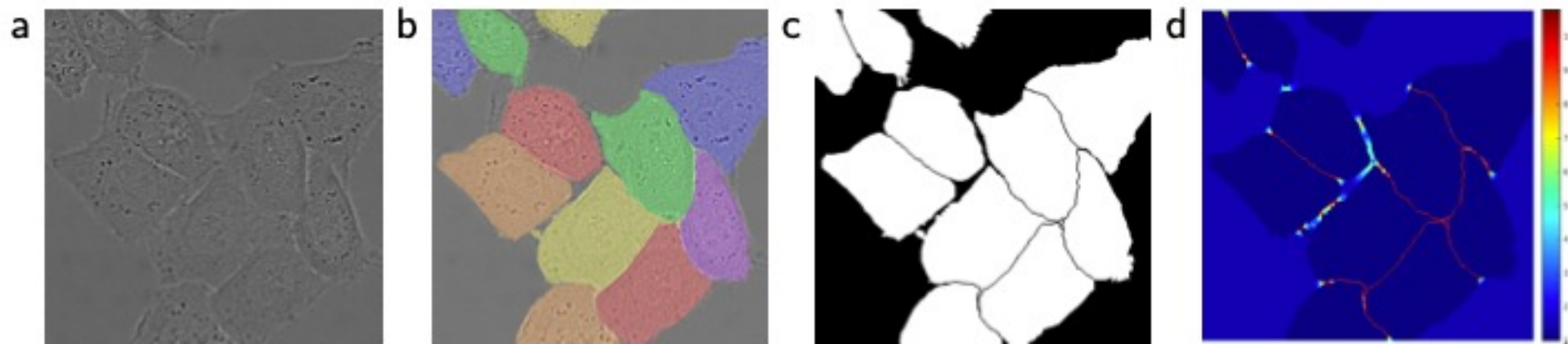
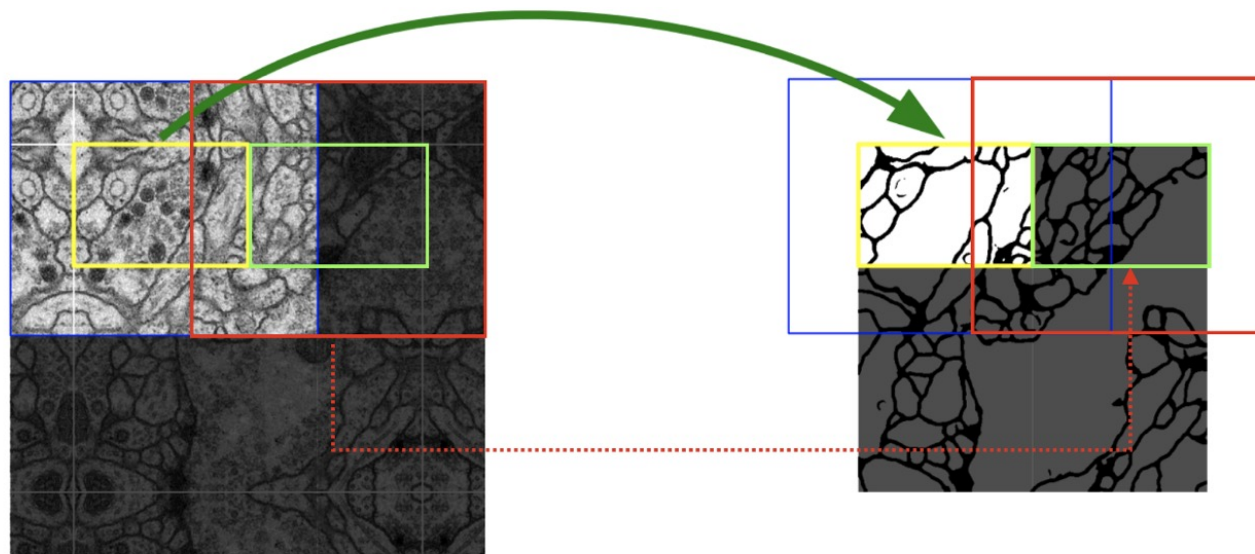
- Encoder(Contracting path)-Decoder(Expanding path) 구조
- Contracting path : Convolutional layer
- Expanding path : Up-sampling(Transposed Convolution), Skip connection
- 다양한 학습 방법 사용
- Patch 선택 시 Overlap 비율이 적어 속도 상승
- Patch size에 따른 context 추출과 Localization간의 trade-off 개선



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>



StructureFlow : image inpainting via structure-aware appearance flow

- Structure reconstruction, Texture generation
- Bilinear sampling 대신 Gaussian sampling을 사용해 sampling 작업의 receptive field 확장
- 올바른 영역이 sampling되었는지 확인하기 위해 sampling correctness loss function 도입
- Edge-preserved smooth image를 사용한 structure reconstructor 제안
- Appearance flow

Related Work

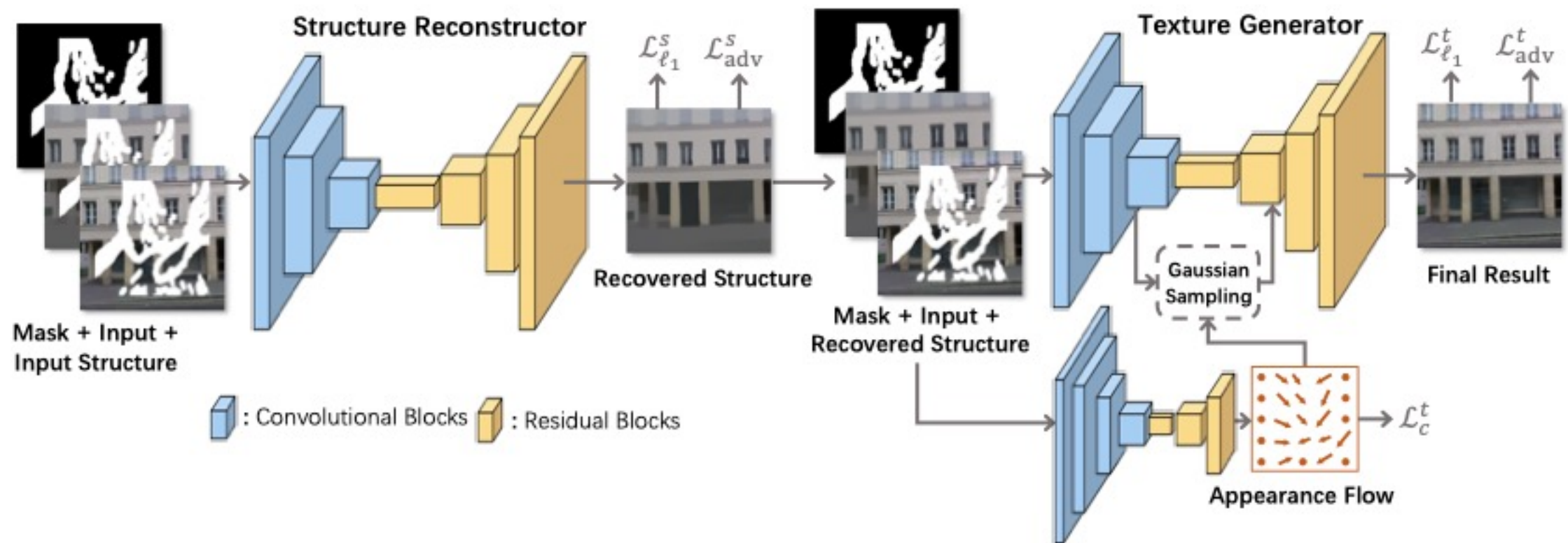
- 기존 image inpainting은 diffusion-based, patch-based방식이 존재
- Diffusion-based : 인접 영역 정보를 hole로 전파하여 texture 생성 -> 작은 hole만 처리 가능
- Patch-based : 원본 image의 손상되지 않은 영역에서 유사한 image patch를 검색, 복사하여 대상 영역을 채움 -> 상대적으로 큰 missing holed에 대해 사실적인 질감 생성
- 적합한 image patch를 찾기 위해 similarity를 계산할 때 더 많은 Visual information과 Visual artifact의 절감을 위해 bidirectional similartiy를 제안. Patchmatch

Related Work

- Patch-based method는 hole이 있는 부분이 없는 부분과 유사한 내용을 가진다고 가정
- -> 반복적 구조를 가지는 image에서는 잘 작동하지만 독특한 구조를 가진 image에서는 합리적 결과 X
- Context encoder : Encoder-Decode 구조를 사용해 feature를 추출, output 재구성
- CNN은 Long-term correlation에 적합하지 않음 -> contextual attention
- EdgeConnect : Edge map을 복구한 후 세부적으로 채움. Edge map의 제한된 representation ability
-> Object의 경계에서 잘못된 정보 생성 가능

Related Work

- Optical flow를 정확하게 추정하기 위해 Deep NN을 사용 가능하나 충분한 ground truth optical flow field를 필요로 함.
- Unsupervised learning을 사용하면 large motion을 포착하기 어려움 -> 저자는 이것이 bilinear sampling의 limited receptive field 때문이라고 생각해 gaussian sampling 사용
- Appearance flow : source, target correlation을 계산해 2D coordinate vector(appearance flow field)를 예측 -> image inpainting에 대해 적용. Source -> Hole에 대한 appearance flow 적용



Structure reconstructor

- Predict missing structure -> global structure image
- Generative adversarial framework
- Reconstruction loss : 예측한 structure와 ground-truth structure간의 L1-distance
- Adversarial loss : Generative adversarial framework

Texture generator

- Draw details according to reconstructed structure -> output image
- Yield vivid texture
- Appearance flow in Unsupervised manner
- Gaussian sampling
- Sampling correctness loss (calculate by pre-trained VGG19)