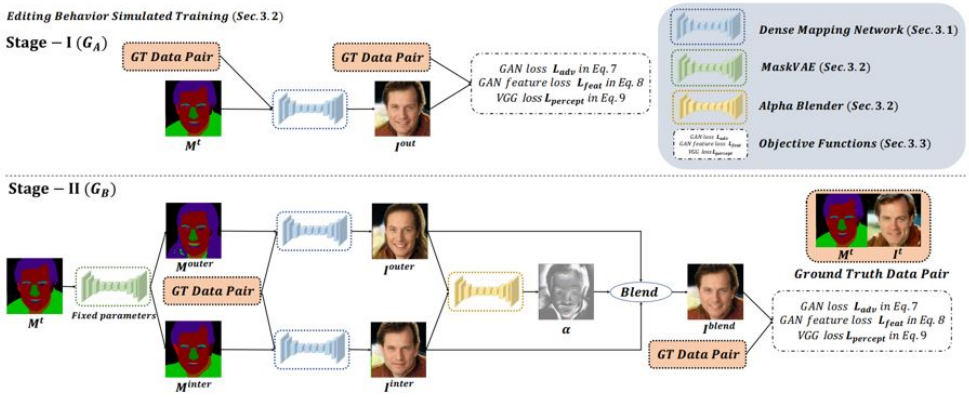
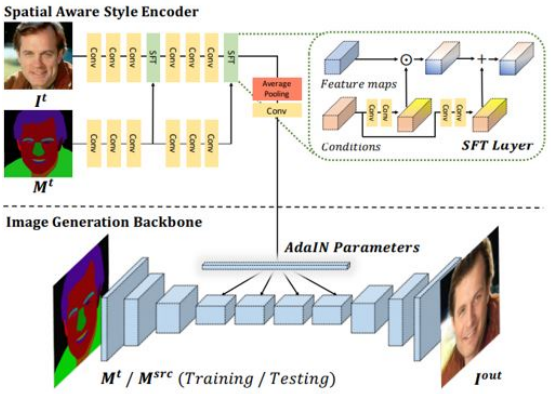
D21010 정재민 jaemin.jeong@hallym.ac.kr

본 보고서는 Semantic Image Synthesis와 Image Inpainting 관련 논문을 10편 조사한 보고서입니다.

1 MaskGAN: Towards Diverse and Interactive Facial Image Manipulation [1]



(*A*)



(*B*)

Algorithm 1 Editing Behavior Simulated Training

Initialization: Pre-trained $G_A, Enc_{\text{VAE}}, Dec_{\text{VAE}}$ models
Input: I^t, M^t, M^{ref}
Output: I^{out}, I^{blend}

1: **while** iteration not converge **do**
2: Choose one minibatch of N mask and image pairs $\left\{M_i^t, M_i^{ref}, I_i^t\right\}, i = 1, ..., N.$
3: $z^t = Enc_{\text{VAE}}(M^t)$
4: $z^{ref} = Enc_{\text{VAE}}(M^{ref})$
5: $z^{inter}, z^{outer} = z^t \pm \frac{z^{ref} - z^t}{\lambda_{inter}}$
6: $M^{inter} = Dec_{\text{VAE}}(z^{inter})$
7: $M^{outer} = Dec_{\text{VAE}}(z^{outer})$
8: Update $G_A(I^t, M^t)$ with Eq. 6
9: Update $G_B(I^t, M^t, M^{inter}, M^{outer})$ with Eq. 6
10: **end while**

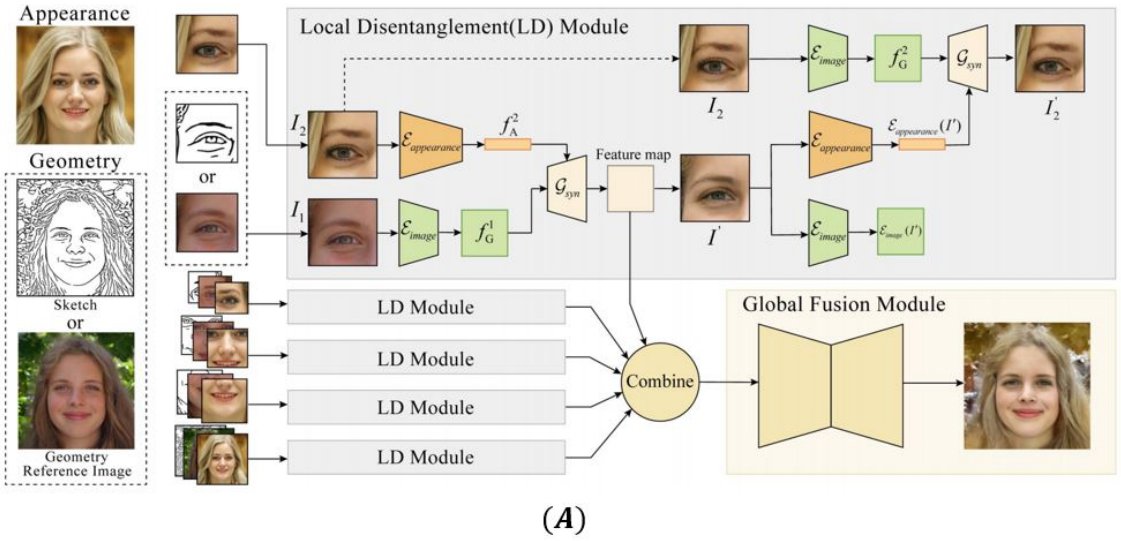
(*C*)

본 논문은 이전 방법이 미리 정의된 얼굴 속성에서 동작하거나 사용자가 대화식으로 image를 조작할 수 없다는 문제점을 지적하고 **MaskGAN**이라는 프레임워크를 제안합니다.

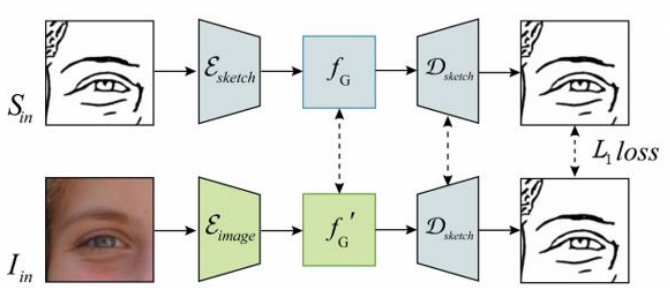
그림 (A)는 Editing Behavior Simulated Trining(EBST)의 흐름을 나타냅니다. EBST는 사용자의 편집 행동을 모델링합니다. EBST는 잘 훈련된 Dense Mapping Network(DMN), 낮은 reconstructure error를 가지는 MaskVAE, 처음부터 훈련 된 Alpha Blender 가 필요합니다. Stage-1은 target image의 semantic label mask M^t와 target image I^t를 입력받아 DMN를 통과해 I^{out}을 출력하며 target image I^t와 GAN loss, GAN feature loss, VGG loss를 구하는 학습 절차를 나타냅니다. 여기서 DMN는 그림 (B)로 표현 됩니다. DMN은 Spatial Aware Style Encoder과 Image Generation Backbone으로 이루어집니다. Spatial Aware Style Encoder을 통해 Generator Backbone에 사용할 AdalN parameters를 구하고 이를 통해 image를 생성합니다.

Stage-2에서는 먼저 MaskVAE의 인코더 부분에 target mask M^t와 reference mask(random selection) M^{ref}를 통과시켜 $z^t \pm \frac{z^{ref} - z^t}{\lambda_{inter}}$ 로 z^{inter}, z^{outer}를 계산 한 뒤 MaskVAE의 디코더 부분에 넣고 M^{inter}과 M^{outer}를 만듭니다 ($\lambda = 2.5$). 핵심 아이디어는 학습 된 MaskVAE에서 manifold를 순회하여 두개의 local perturbed input masks를 생성 결과 입니다(사람의 mask 조작을 모방). 각각의 mask로 DMN을 거쳐 I^{inter}과 I^{outer}를 만들고 둘을 Alpha Blender를 사용해 manipulation consistency를 유지하기 위한 $\alpha, I^{inter}, I^{outer}$를 blend 한 $I^{blend} = \alpha \times I^{inter} + (1 - \alpha) \times I^{outer}$를 만듭니다. 그리고 I_t와 GAN loss, GAN feature loss, VGG loss를 구하여 학습합니다. 여기서 Alpha Blender는 원본 image와 일관성을 유지하면서 두 image를 target image쪽으로 혼합하는 방법을 학습합니다. 그림 (C)는 전체 학습 절차를 나타냅니다. 훈련시에는 각각의 stage에서 M_t를 사용하여 훈련되지만 추론시에는 사용자가 조작한 mask M_{src}를 사용합니다.

2 DeepFaceEditing: Deep Face Generation and Editing with Disentangled Geometry and Appearance Control [2]



(A)



(B)

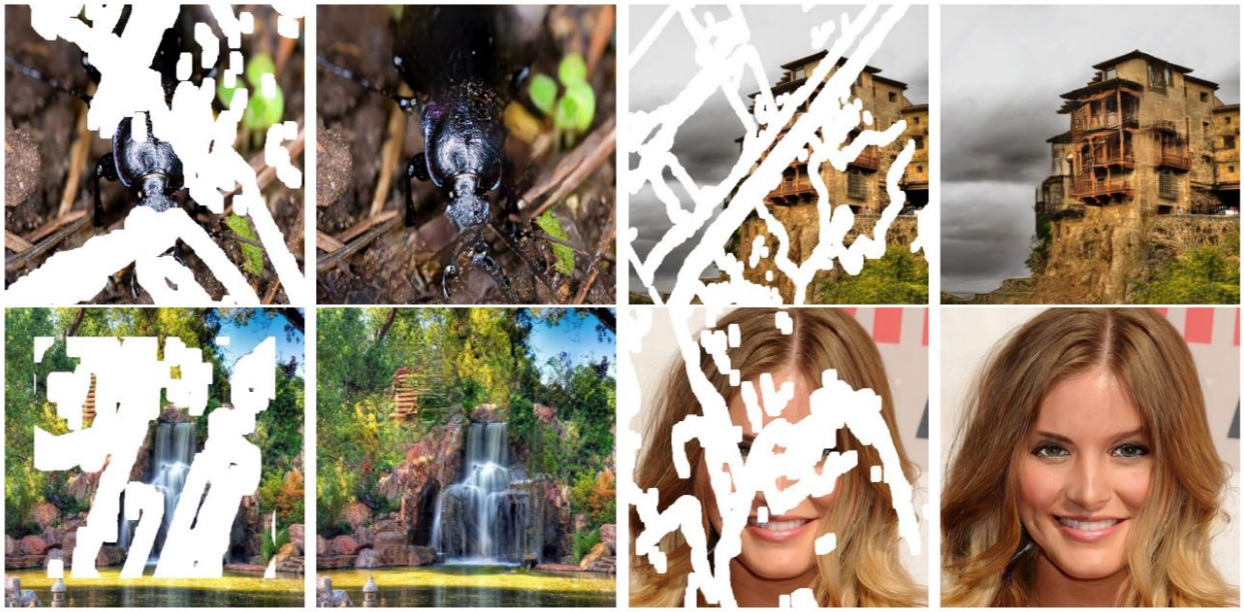
본 논문은 sketch-based 모델로 기존에 존재하는 disentangling 방법이 face editing에 최적화 되어있지 않으며 주름과 같은 얼굴의 세밀한 부분을 제어하지 못한다는 한계를 해결하기위해 structured disentanglement 프레임워크인 DeepFaceEditing를 제안합니다. 고품질 얼굴 image를 생성하기위해 local-global 프레임워크를 사용하고 5개의 구성요소(왼쪽 눈, 오른쪽 눈, 코, 입, 배경)로 분해하여 개별 네트워크 모듈로 처리합니다. DeepFaceEditing은 각 구성 요소의 geometry와 appearance로 분리하는 local disentanglement 모듈과 각 구성 요소의 기능을 통합하여 고품질의 결과를 생성하는 global fusion 모듈로 구성됩니다.

Local Disentanglement(LD) 모듈은 그림 (A)에서 보여줍니다. LD 모듈은 Geometry Encoder와 Appearance Encoder로 구성됩니다. sketch domain은 이미 윤곽선을 묘사하기 때문에 geometry 정보를 단순한 auto encoder 구조로 추출할 수 있습니다. 하지만 가장 큰 문제는 실제 image에서 geometry 정보를 추출하는 것입니다. 직관적인 접근법은 사전 훈련된 image-to-sketch 변환 모델을 사용하여 실제 image를 sketch domain으로 보내는 것이지만 LD는 sketch와 실제 image로 부터 geometry 정보를 추출하는 통합 된 방법을 보여줍니다. 먼저 그림 (B)의 top에서 sketch S_{in}의 중간 특징을 추출하기 위해서 sketch encoder E_{sketch}와 sketch decoder D_{sketch}를 학습시킵니다. f_G는 인코딩 된 스케치 image이며 실제 image I_{in}를 인코딩한 값이 f_G와 같기를 원하면서 real image encoder E_{image}를 학습시킵니다($f_G^{'} = E_{image}(I_{in})$). f_G와 $f_G^{'}$를 동일한 분포를 가지도록 하기 위해서 sketch decoder에 $f_G^{'}$를 입력하고 f_G와 $f_G^{'}$, $D_{sketch}(f_G)$ 와 $D_{sketch}(f_G^{'})$의 L1-loss로 학습을 진행합니다.

$E_{appearance}$를 사용하여 appearance feature를 추출합니다. global average pooling을 사용하여 공간 정보를 제거하고 apppearance feature를 추출합니다. Image Synthesis Generator G_{syn}은 4개의 residual blocks과 4개의 up-sampling layers로 구성됩니다. 입력 image의 크기와 같은 64-channels의 feature map이 생성되며 단일 convolution layer를 통해 input geometry와 appearnce feature를 가지는 image가 생성됩니다. 그림 (A)에서 image $I^{'}$를 생성한 후에 나오는 모듈들은 cycle loss를 구하기 위한 작업에 요구되는 모듈입니다.

Global Fusion Module은 encoder, residual blocks and a decoder로 구성되며 LD 모듈에 의해서 인코딩 된 local image feature를 통합하여 최종 image를 생성합니다. 생성 된 local image를 그대로 결합하는 방법이 있지만 이렇게하면 경계선에 artifacts를 생성하게 됩니다. 그래서 이 논문에서는 image 생성 네트워크에 입력하기 전에 LD 모듈의 중간 feature map(64-channels)을 결합한 뒤(결합에 대한 언급이 없지만 여기서 말하는 결합은 concat이나 addition이 아니라 해당 위치(눈, 코, 입)에 붙이는 작업으로 보임) 입력하여 최종 결과를 생성합니다.

3 Image Inpainting for Irregular Holes Using Partial Convolutions [3]



본 논문은 규칙적이지 않은 mask를 채우기 위해 standard convolutional network는 성능이 좋지 않음을 파악하여 masked image를 위한 Partial Convolutions을 제안한 논문입니다. partial convolution operation과 mask update function을 통합하여 Partial Convolutional Layer라고 부릅니다. $x^{'}$는 출력 feature 입니다.

$$x^{'} = \begin{cases} W^T(X \odot M)\frac{sum(1)}{sum(M)} + b & sum(M) > 0 \\ 0 & otherwise \end{cases}$$

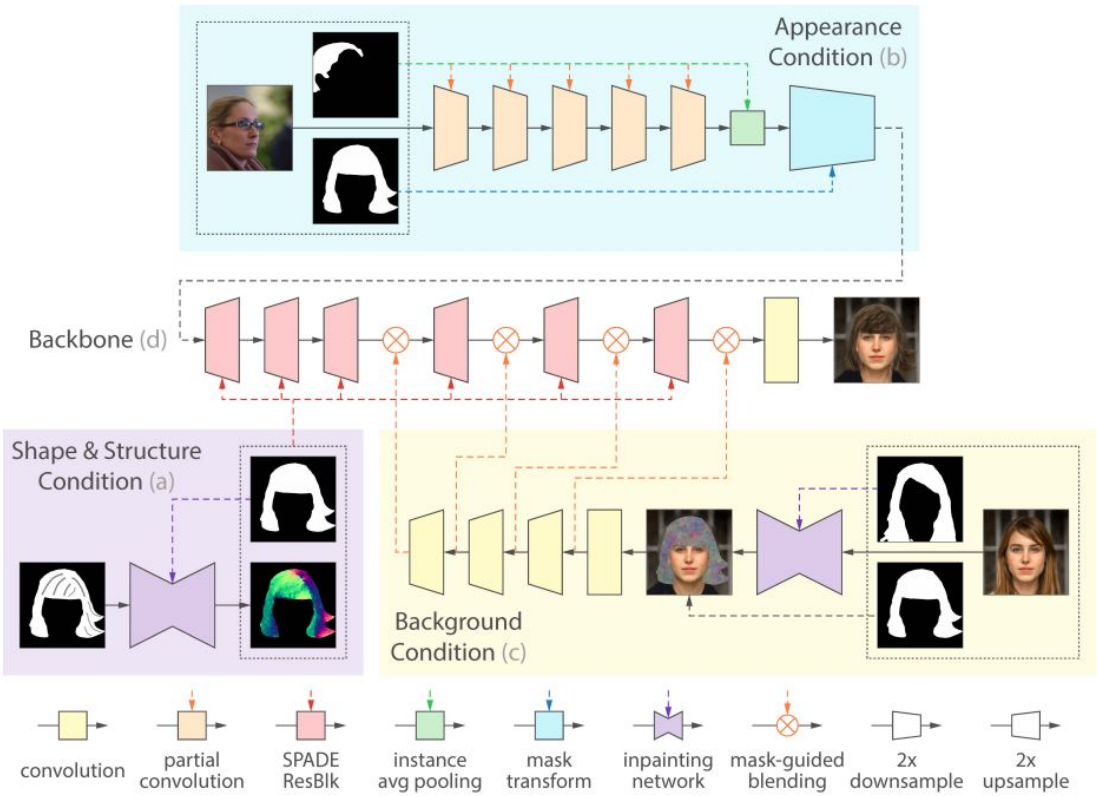
W는 convolution filter weights를 나타내고 b는 bias, X는 현재 window에 feature values를 나타냅니다. M은 그에 해당하는 binary mask 입니다 (가려진 pixel이 1). scaling factor $\frac{sum(1)}{sum(M)}$는 현재 window 내에서 가려지지 않은 pixels에 곱하여 normalization 역할을 합니다. $m^{'}$은 출력 mask 입니다.

$$m^{'} = \begin{cases} 1 & sum(M) > 0 \\ 0 & otherwise \end{cases}$$

X와 M이 partial convolution layer에 입력되면 $x^{'}$와 $m^{'}$가 출력 됩니다. M이 $m^{'}$로 변환될때 $m^{'}$이 0 이상의 값을 가진다면 해당 mask pixel은 1로 되며 다음에 convolution layer의 연산에 포함되게 됩니다. 즉, Mask가 점점 채워지며 조금씩 복원 됩니다.

모델은 그림 설명이 없지만 U-net 기반이며 3-channels mask와 image로 구성되고 U-net의 모든 convolution layer를 partial convolution layer로 대체합니다.

4 MichiGAN: Multi-Input-Conditioned Hair Image Generation for Portrait Editing [4]



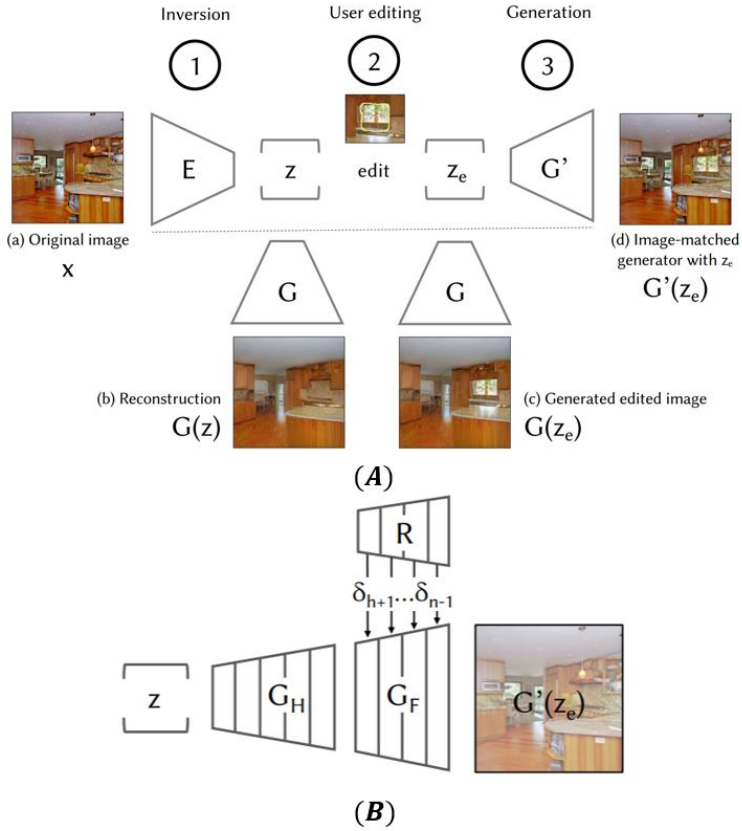
본 논문은 기하학적인 형상과 모양의 복잡성으로 인해 여전히 도전적인 과제인 머리카락 조작을 위한 MichiGAN을 제안합니다. 머리카락의 조작을 위해 shape, structure, appearance, background 4가지 속성을 명시적으로 분리합니다. 입력 image I가 주어지면 MichiGAN은 배경을 변경하지 않고 머리카락을 편집하는 것을 목표로 합니다. 출력 I_{out}을 구하기 위해 입력 image I, 머리카락 mask M, dense orientation map O, 머리카락 모양 참조 image I_{ref}를 모델에 입력합니다($I_{out} = G(M, O, I_{ref}, I)$).

저자는 머리카락 mask 영역은 실제 머리카락 모양의 근사치라는 문제점을 지적하고 사실적인 모양과 자연스러운 blending 효과를 위해 fuzzy mask conditioning을 사용하여 머리카락 mask boundary가 flexibility 가지게합니다. 이를 위해 학습 중에 머리카락 mask 영역을 랜덤한 폭으로 확장하거나 침식시켜 정확하게 boundary를 분리합니다. 그리고 머리카락 구조는 보통 강한 anisotropy(이방성)과 homogeneity(균질성)을 가지기에 항상 머리카락이 영역 내에 균일하게 분포되어있습니다. 따라서 각 pixel에 대해 2D orientation angle을 정의하는 머리카락 구조를 표현하기 위해 dense orientation map을 사용합니다. confident pixel-wise orientation maps의 상태를 추정하기 위해서 oriented filter kernels$\{K_\theta\}$(32 Gabor filters)를 사용합니다. 입력 image I의 pixel position p의 orientation label map $o^{'}$와 신뢰도 w는 $o^{'}_p = argmax_\theta |(K_\theta \otimes I)_p|$, $w_p = max_\theta |(K_\theta \otimes I)_p|$ 으로 정의합니다. continuous rotation 표현은 신경망에서 중요합니다. orientation label map은 $[0, \pi)$의 discrete 각도 값이 포함되며 실제로 continuous 하지 않습니다. 그래서 이를 2-channels continuous orientation map $O^{'}$로 변환합니다. $O^{'}_p = [cos(2 \cdot o^{'}_p), sin(2 \cdot o^{'}_p)]$ 그리고 역전파에서는 $o^{'}_p = arctan(O^{'}_p)/2$로 나타냅니다.

그림의 (a)는 연산 후 $O^{'}$와 M이 backbone 내부의 SPADE [5]의 중간 입력으로 들어가는 것을 보여줍니다. 그림의 (b)는 머리의 appearance를 고려한 부분입니다. reference image에서 스타일을 추출하여 입력 image의 머리 모양에 전송합니다. 마지막 partial convolution [3] 의 출력 $A^{'}_{ref}$은 mask transform을 거쳐 backone에 입력됩니다. mask transform은 $A = \frac{\sum(A^{'}_{ref} * M_{ref})}{\sum M_{ref}} \odot M$ 으로 정의됩니다. $*$는 element-wise 곱이며 $\odot$은 duplication 연산자입니다. mask transform을 통해 머리카락 영역에만 초점을 맞추며 appearnce 특징을 균일하게 하여 결과 품질을 높여줍니다. 그림의 (c)는 마지막으로 고려해야할 배경에 대한 정보를 추출하는 역할을 합니다. 저자는 머리카락을 제외한 영역은 고정되길 원합니다. background encoder에 입력되는 $I_{back} = N * M^{'} + \mathcal{I}(I * M_{in}, M_{in} - M_{in} \cap M) * (1 - M^{'})$은 원본 배경 영역입니다. $\mathcal{I}$에서 나온 image의 mask 영역을 랜덤 노이즈 N으로 채우고 background image inpainter [3] $\mathcal{I}(I, M)$와 합치게 됩니다. 그리고 I_{back}이 background condition module에 입력됩니다. background condition module i 번째 layer 출력 F^b_i와 backbone module의 뒤에서 부터 $i+1$ 번째 layer 출력 F^g_i을 mask-guided 방법으로 $F_i = F^g_i * M + F^b_i * (1 - M)$를 만들어 이를 통해 배경 정보를 backbone에 전달합니다. 위에 언급 된 방법으로 appearance, shape, structure, background 4가지 속성을 통해 최종 결과를 출력합니다.

5 Semantic Photo Manipulation with a Generative Image Prior [6]

NOTE. 그림이 약간 헷갈림 $G(z_e)$를 $x_e = G(z_e)$ 이런식으로 바꾸어야 할 듯



본 논문은 natural photograph의 highlevel 속성을 조작할 때 image를 정확히 재현하기 어렵다는 점, 조작 후 새롭게 합성 된 pixel이 원본 image에 잘 맞지 않다는 점과 같은 문제를 해결하기 위한 방법을 제안합니다. 그림 (A)는 image editing 파이프라인을 보여줍니다. natural photograph가 주어지면 먼저 image generator를 사용하여 re-render 합니다. 그 후에 사용자는 특정 object를 추가, 제거, 변경하는 대화형 인터페이스를 사용하여 image를 조작합니다. 이러한 편집에 따라 latent representation을 업데이트 하고 수정 된 representation이 주어지면 최종 결과를 rendering 합니다.

먼저 $z = E(x)$를 통해 latent vector z를 계산합니다. E는 인코더 [7] 입니다. 그리고 semantic 조작 $z_e = edit(z)$를 적용하여 latent vector z를 수정합니다. 그리고 수정된 z_e로 이미지를 생성합니다. 하지만 그림 (A)의 (b)에서 보이듯이 z는 정확한 결과를 생성하지 못하기 때문에 z_e로 생성 된 이미지 x_e 에는 많은 속성이 손실됩니다. 그래서 원본 이미지 x의 편집되지 않은 부분의 영역을 잘 생성하게 하는 $G^{'}$를 학습합니다. 최종 결과는 (d)에서 볼 수 있으며 원본 정보를 손실하지 않고 이미지 $x_e^{'}$ 를 생성할 수 있음을 보여줍니다.

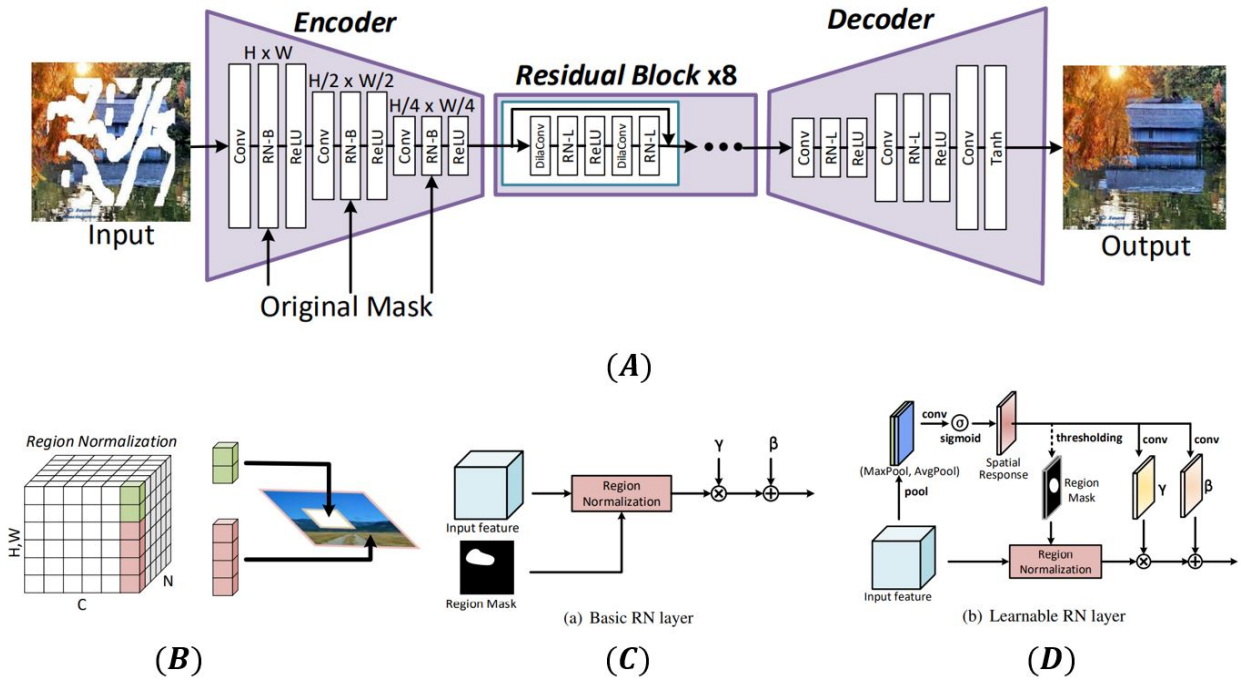
G는 progressive GANs [8] 입니다. 그림 (B)에서 G_H는 high-level layers를 나타내며 G_F는 fine-grained layers를 나타냅니다. 생성기의 초반 layers는 객체의 존재와 layout과 같은 high-level 정보를 나타내며 후반 layers는 객체의 color, edge와 같은 low-level 정보를 나타냅니다. G의 semantic structure를 그대로 두고 학습하기를 원하기 때문에 G_H는 고정하고 G_F만 학습하게 됩니다. 하지만 그대로 학습시키게 되면 과적합이 발생하여 특정한 artifact가 생성됩니다. 그래서 작은 네트워크 R을 만들고 R의 출력에 약간의 perturbations δ를 곱하여 과적합을 방지시킵니다.

마지막으로 $edit$을 어떻게 했는지를 설명합니다. 추가, 제거, 수정이 있는데 먼저 추가와 제거를 하기위해 원하는 클래스 c를 선택합니다. 그리고 사용자가 선택 된 영역 $U \in \mathbb{R}^{8\times8\times512}$를 통해 channel mask $a_c = (i_c \otimes U) \in \mathbb{R}^{8\times8\times512}$를 구성합니다. $\otimes, \odot$은 각각 broadcasted outer product, elementwise Hadamard product를 나타냅니다.

$$z_e = (1 - \alpha_c) \odot z + \alpha_c \odot (sp_c)$$

z는 feature vector p_c와 blending 되며 $p_c \in \mathbb{R}^{8\times8\times512}$는 모든 이미지에 대한 객체의 클래스의 평균 activation을 나타냅니다. s를 0보다 크게 설정하면 추가가 되며 0으로 설정하면 제거가됩니다. 수정의 경우 위 식에서 p_c의 i번째 구성요소를 클래스 c에 관련 channel에 대한 reference image의 latent vector z의 i번째 채널에서 positive activations의 평균을 취하는 것인데 이때 다른요소는 0로 합니다.

6 Region Normalization for Image Inpainting [9]

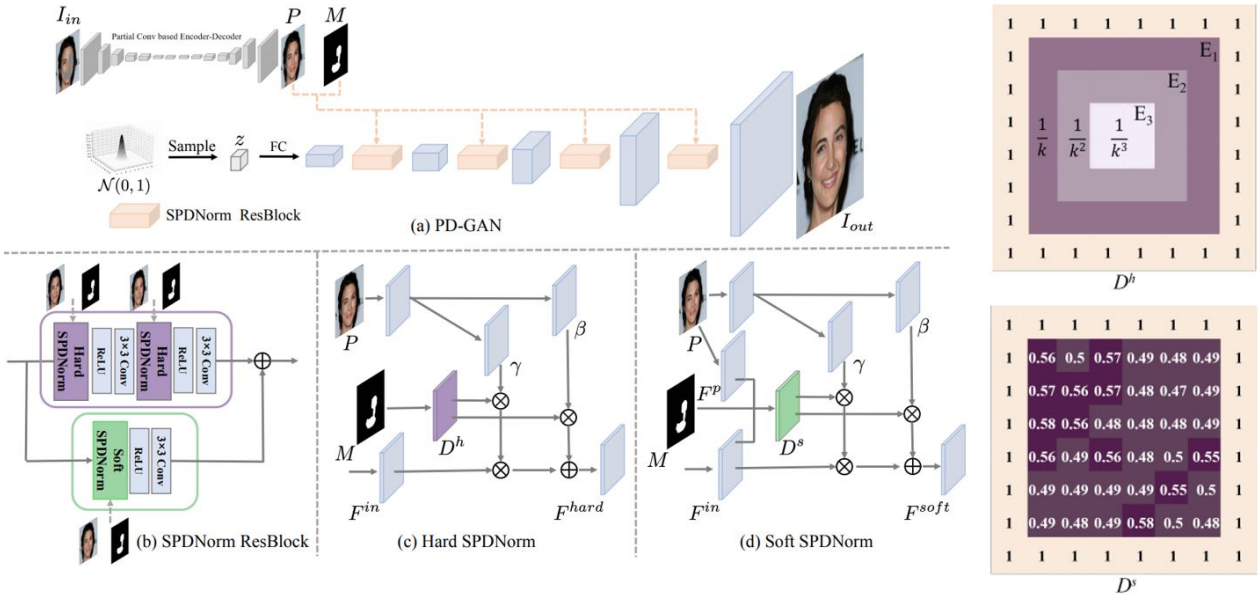


본 논문은 기존 Image Inpainting 방법이 image의 손상된 영역이 정규화에 미치는 영향을 고려하지 않고 Feature Normalization를 적용하는 문제점에 대해 지적하고 Region Normalization이라는 정규화 방법을 제안합니다. Region Normalization을 통해 Inpainting Mask를 기반으로 단순하게 손상된 영역과 손상되지 않은 영역 Normalization을 각각 계산하여 평균과 분산의 shift 문제를 해결합니다. shift 문제는 Region Mask가 255값인 경우 평균과 분산이 각각 255, 0이 되는데 이를 통해 Normalization 되면 손상되지 않은 영역의 픽셀 값이 shift 되는 현상입니다.

그림에서 (A)는 전체적인 모델 구조를 나타냅니다. RN-B는 그림 (C)의 Basic RN layer이며 RN-L은 그림 (D)의 Learnable RN layer입니다. 그리고 그림 (B)는 각각의 layer에 포함되어 있는 Region Normalization을 나타냅니다. RN-B는 Region Mask를 직접 입력으로 하여 학습 파라미터 γ, β만 학습시킨다면 RN-L은 Region Mask를 스스로 감지하는 방법을 학습합니다. RN-L은 global affine transformation 통해 손상된 영역과 손상되지 않은 영역 결합을 부드럽게 하기위해 사용됩니다.

결론적으로, 제안 된 Region Normalization을 통하여 평균과 분산의 shift 문제를 해결하고 Image Inpainting 성능을 향상시킬 수 있다는 것을 보여줍니다.

7 PD-GAN: Probabilistic Diverse GAN for Image Inpainting [10]



본 논문은 probabilistic diverse GAN(PD-GAN)을 제안한 논문입니다. PD-GAN은 random noise를 기반으로 이미지를 생성하는 Vanilla GAN을 기반으로 합니다. hole을 채우는 동안 hole의 boundary의 pixels가 더 결정적이어야하고 hole의 중심으로 갈수록 다양성이 있어야한다고 주장합니다. 이를 위해서 Spatially Probabilistic Diversity Normalization(SPDNorm)를 제안합니다. SPDNorm은 hole의 경계쪽으로 갈 수록 이미지와 더 유사한 pixels을 생성하며 중심쪽으로 갈수록 더 다양한 pixels를 생성합니다.
대략적인 예측을 얻기위해서 미리 학습된 Partial Convolutional encoder-decoder [3]을 사용합니다. 사전 지식을 제공하기 위해서 대략적인 예측과 mask image를 SPDNorm Residual Blocks으로 입력합니다. SPDNorm Residual Blocks는 Hard SPDNorm과 Soft SPDNorm로 구성됩니다.
hard probabilistic diversity map D^h는 학습하는 process 없이 inpainting mask M에 의해서 결정됩니다. soft probability diversity map D^s는 학습 process를 포함하며 입력 feature map과 대략적인 예측으로 얻어진 adaptive map 입니다.
그림에서 $(c),(d)$에서 $P \in \mathbb{R}^{H\times W\times 3}$는 사전 정보(대략적인 예측), $F_{in} \in \mathbb{R}^{H\times W\times C}$ 입니다.

$$F^{hard}_{x,y,c} = D^h_{x,y}(\gamma_{x,y,c}(P)\frac{F^{in}_{x,y,c}-\mu_c}{\sqrt{\sigma_c^2+\epsilon}} + \beta_{x,y,c}(P)) \tag{1}$$

$$F^{soft}_{x,y,c} = D^s_{x,y}(\gamma_{x,y,c}(P)\frac{F^{in}_{x,y,c}-\mu_c}{\sqrt{\sigma_c^2+\epsilon}} + \beta_{x,y,c}(P)) \tag{2}$$

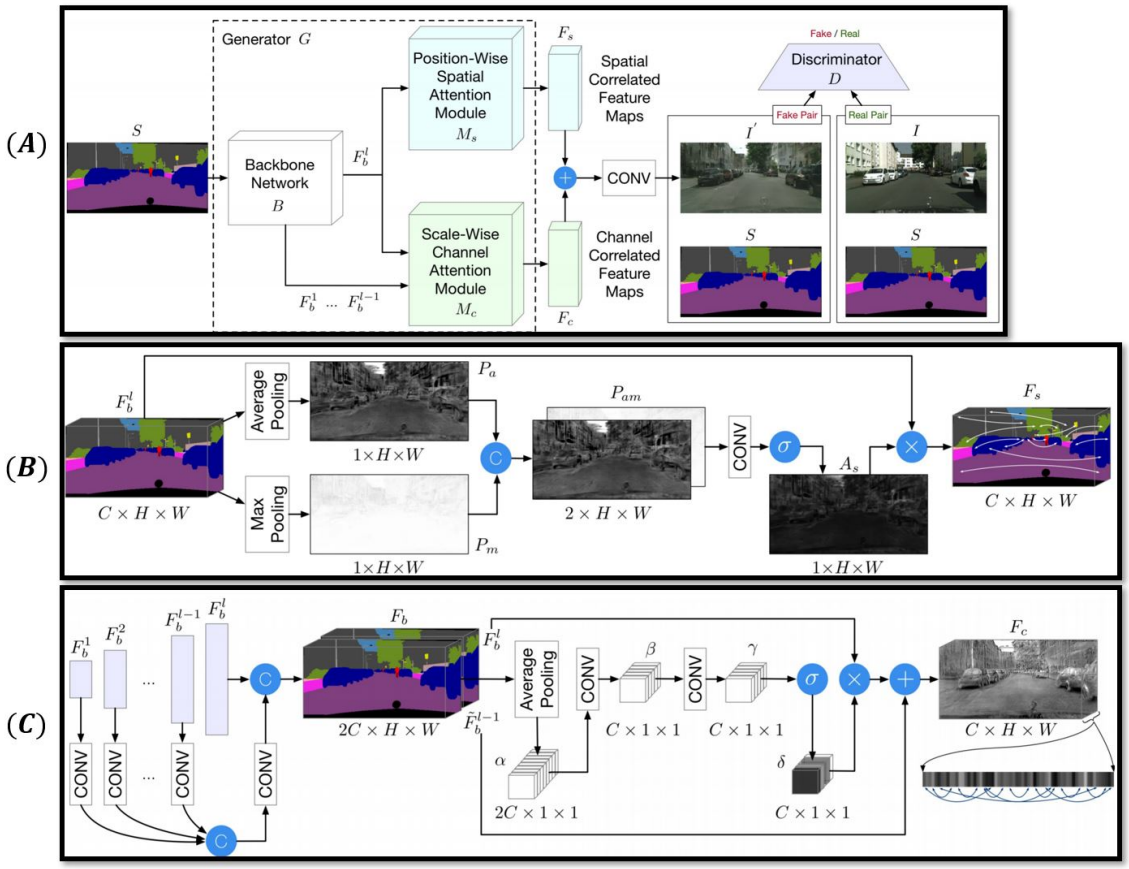
$$\mu_c = \frac{1}{H\times W}\sum_{x=1}^{H}\sum_{y=1}^{W}F^{in}_{x,y,c} \tag{3}$$

$$\sigma_c = \sqrt{\frac{1}{H\times W}\sum_{x=1}^{H}\sum_{y=1}^{W}(F^{in}_{x,y,c}-\mu_c)^2} \tag{4}$$

$$\tag{5}$$

이 논문에서 중요한 것은 D^s와 D^h 입니다. 내부는 3×3 kernel에서 시작하여 n 만큼의 반복 확장을 통해 kernel을 만듭니다. 각 영역은 $\frac{1}{k^i}$값을 가지게 됩니다. hole이 아닌 부분은 1로 처리합니다. D^h는 고정된 값으로 중앙으로 갈수록 기하급수적으로 작아지는 kernel입니다. 본 논문에서 $(k=4)$를 사용합니다. D^s는 학습으로 생성되는 kernel이며 $D^s = \sigma(Conv([F^p, F^{in}]))\cdot(1-M)+M$ 식으로 정의됩니다. F^p는 P가 convolution layer를 통과하여 나온 사전정보입니다. 배경에 포함되는 pixels는 1이되며 D^s는 σ를 통해 P에서 정보를 빌릴 확률을 적응적으로 변경합니다.

8 Dual Attention GANs for Semantic Image Synthesis [11]



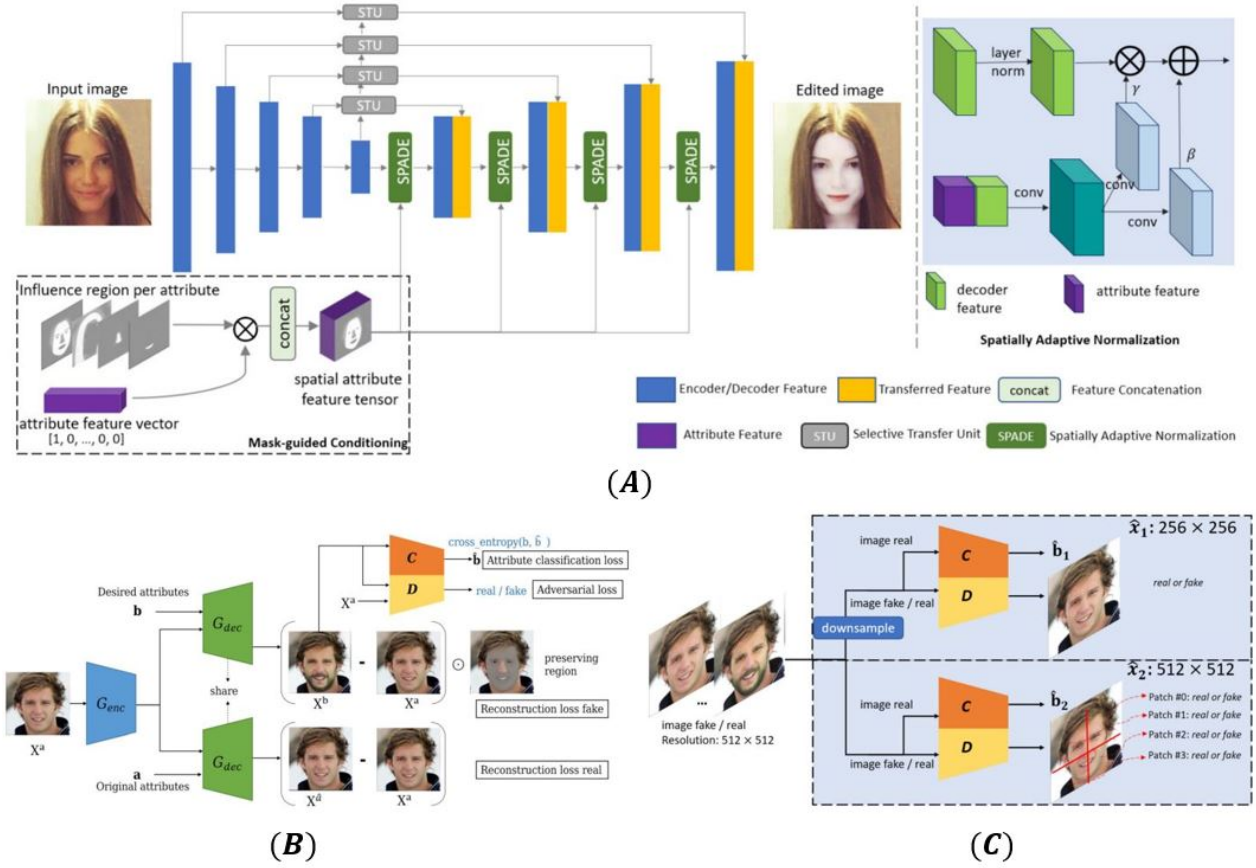
본 논문은 semantic image sysnthesis task에 중점을 둡니다. 기존 방법에 semantic 정보를 보존하고 공간과 채널 차원에서 구조적인 상관관계를 무시하는 효과적인 semantic constraints이 없기 때문에 불만족스러운 결과가 나오게 된다고 언급합니다. 그리고 이러한 제약사항을 해결하기 위해서 기존 방법의 network architecture를 수정하지 않고 입력 레이아웃의 세부 정보를 사용하여 사실적이고 의미적으로 일관된 이미지를 합성하는 새로운 Dual Attention GAN(DAGAN)과 공간 및 채널 차원에서 semantic structure를 포착하기 위해 Spatial Attention Moudle(SAM)과 scale-wise Channel Attention Module(CAM)을 제안합니다. 그림 $(A), (B), (C)$는 각각 DAGAN, SAM, CAM에 대한 전체적인 구조를 보여줍니다.

SAM은 point-wise 방식이라고 설명됩니다. 보통 max pooling, average pooling은 feature map을 기반으로 처리하지만 이 논문에서는 동일한 의미의 label을 가진 영역간의 상관관계를 모델링해야하기 때문에 channel을 기반으로 처리합니다. 각 pooling에서 추출된 특징 맵을 concat하고 P_{am}을 만든 뒤 sigmoid를 거쳐 A_s를 만듭니다. 입력 특징맵 F_b^l과 곱하여 최종적으로 결과 F_s를 만들어냅니다.

CAM은 scale-wise 방식입니다. backbone network에 각 layer의 특징 맵을 동일한 크기로 만든 뒤 전부 concat을 하여 $\tilde{F}_b^{l-1}$을 만듭니다. 그리고 $\tilde{F}_b^{l-1}$를 통해 average pooling을 한뒤 α를 생성하고 β와 γ를 순서대로 생성합니다. 그리고 γ를 sigmoid하여 δ를 만들고 이는 각 channel의 중요도를 나타냅니다. backbone의 마지막 layer의 출력 특징맵 F_b^l을 δ와 곱하고 $\tilde{F}_b^{l-1}$와 더하여 최종적으로 결과 F_c를 만들어냅니다.

CAM과 SAM에서 나온 출력을 더하고 convolution layer를 거쳐 최종 생성 이미지를 만들어 냅니다.

9 MagGAN: High-Resolution Face Attribute Editing with Mask-Guided Generative Adversarial Network [12]



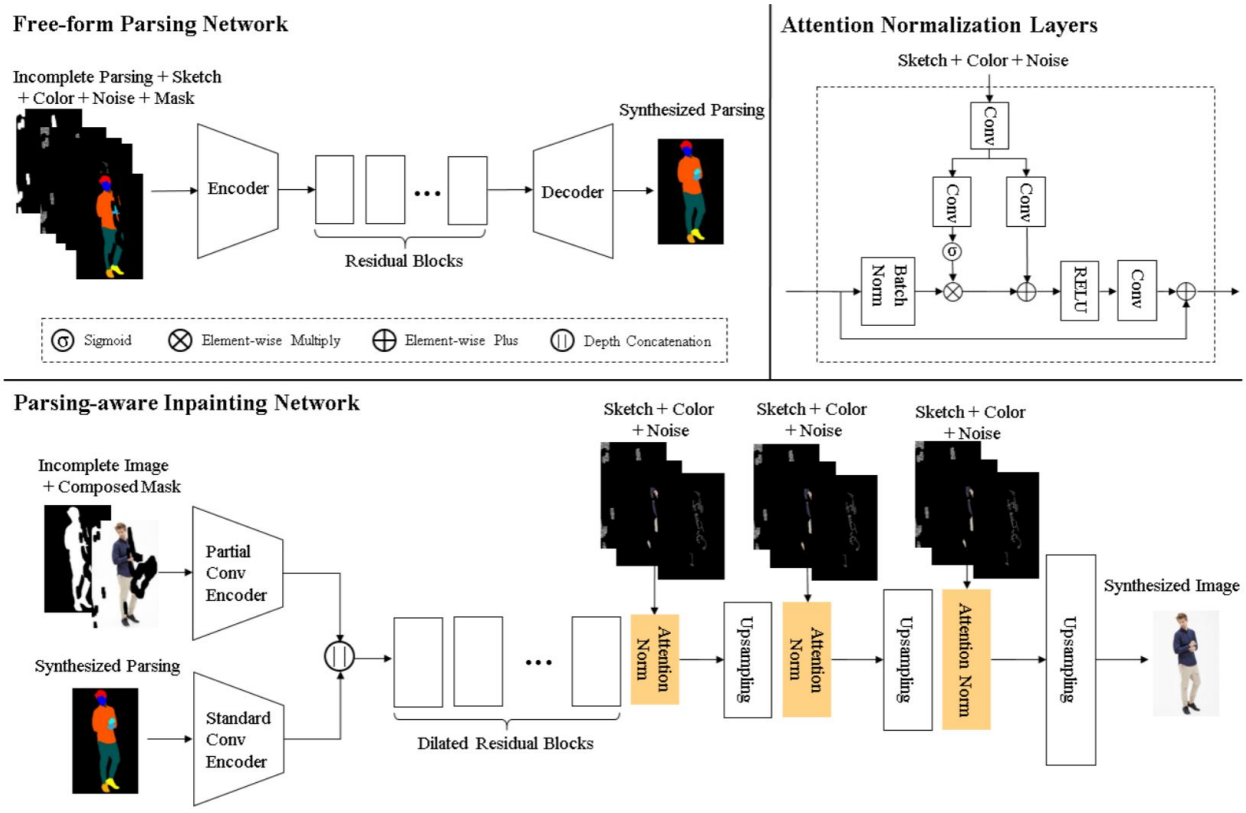
본 논문에서는 고해상도 face attribute editing을 위한 Mask-guided Generate Adversarial Network(MagGAN)을 제안합니다. mask-guided reconstruction loss를 사용하여 속성과 관련없는 영역 (대머리를 만드는데 모자가 포함되어 있는 경우)을 보존하며 편집하는 방법을 학습합니다.

그림 (A)는 MagGAN의 전체적인 encoder-decoder 구조를 나타냅니다. STGAN [13]에 Selective Transfer Units와 adaptive layer normalization [5]를 사용합니다. $\hat{x} = G(x, att_{diff})$로 encoder-decoder 구조를 표현할 수 있으며 $add_{diff} = att_t - att_s$로 표현됩니다. $att_s(\text{or} att_t) \in \mathbb{R}^C$는 속성을 나타냅니다. 속성과 관련없는 영역을 보존하기 위해서 semantic masks를 사용하며 semantic masks는 미리 학습 된 BiseNet [14]을 사용하여 추출됩니다. 기존의 방법과 다른 부분은 mask-guided conditioning인데 변경하기를 원하는 속성의 마스크를 사용하여 SPADE의 입력에 넣어주는 방법입니다. 이를 통해 편집하지 않아야 할 영역을 보존하려합니다.

그림 (B)는 MagGAN의 loss function의 디자인을 나타냅니다. 원본 이미지를 encoding하고 원본 속성, 원하는 속성을 각각 decoder에 입력하여 이미지를 생성합니다. reconstruction loss real는 실제 이미지와 원본 속성과 함께 생성 된 이미지의 L1-loss 입니다. reconstruction loss fake는 실제 이미지와 원하는 속성과 함께 생성 된 이미지의 차이와 influence regions에 따라서 학습하게 됩니다. influence regions는 저자가 직접 정의한 $M_i^+, M_i^- \in [0,1]^{H \times W}$를 사용하는데 만약 "대머리"속성을 가지는 경우 M_i^+에 마킹이 되면 M_i^-는 "배경, 피부, 귀, 귀걸이"속성을 마킹하게됩니다. 즉, 대머리를 만들때 관련이 있는 배경, 피부, 귀, 귀걸이를 제외한 부분에는 영향이 없도록 학습하게 됩니다. attribute classification loss와 adversarial loss는 그림 C에서 설명이 됩니다.

그림 (C)은 discriminators를 나타냅니다. 단일 "shllow"discriminator는 남성/여성 같은 전역적인 개념을 학습할 수 없고 단일 "deep"discriminator는 adversarial 학습을 매우 불안정하게 만들어 이미지의 품질을 저하시킵니다. 본 논문에서는 다단계 패치 기반 "shallow"discriminator를 제안합니다. 위에 coarsest-level discriminator는 전체 down sampling 된 이미지를 보고 이미지 생성에 전반적인 일관성을 담당합니다. 아래에 finer-level discriminator는 고해상도 이미지의 패치를 보고 실제인지 거짓인지 판별합니다.

10 Fashion Editing with Adversarial Parsing Learning [15]



본 논문에서는 free-form sketch, sparse color strokes을 통해 fashion image 조작을 하기위한 Fashion Editing Generative Adversarial Network(FE-GAN)을 제안합니다. FE-GAN은 color와 sketch 조작으로 human parsing generation을 제어하는 방법을 학습하는 free-form parsing network, 세부적인 texture를 rendering하는 parsing-aware inpainting network 두개의 모듈로 구성됩니다. 그리고 이미지의 품질을 향상시키기 위해서 새로운 attention normalization layer가 적용됩니다.

그림에서 free-form parsing network는 incomplete parsing map으로 complete parsing map을 만들어내는 역할을 합니다. incomplete image를 복원하는 것보다는 쉬운 문제입니다. parsing map은 각 부분에서 세부적인 texture를 위한 지침이 될수 있습니다. free-form parsing network에는 사용자의 sketch 정보, incomplete parsing map, color, noise, mask를 입력으로 합니다. U-net 기반의 모델을 사용하여 합성된 parsing map을 출력합니다.

parsing-aware inpainting network는 먼저 [3]을 기반으로한 partial conv encoder를 사용합니다. 입력으로 incomplete image와 composed mask가 들어가며 $M^{'} = (1 - M) \odot M_{foreground}$에서 $M^{'}, M, M_{foreground}$는 각각 composed mask, original mask, foreground mask를 의미합니다. $\odot$은 element-wise multiply를 나타냅니다. 그 다음 free-form parsing network에서 만들어진 합성된 parsing map은 semantic feature를 추출하기 위한 stadard conv encoder에 입력이 됩니다. 그리고 각 출력을 channel-wise concat하여 dilated residual blocks을 거쳐 attention normalization과 upsampling을 반복하여 최종 출력을 얻습니다. Attention Normalization Layer(ANL)은 그림에서 직관적으로 그려져있습니다. ANL은 batch normalization을 하기전 활성화 맵에서 중요한 정보를 추출하는 attention map을 학습한다고 이해하면 됩니다. sketch, color, noise를 통해 α, β를 구합니다. α는 convolution layer 후에 sigmoid를 통해 계산되며 이는 attention map을 나타냅니다. β는 bias를 나타냅니다. 그리고 batch normalization을 통과한 활성화 맵과 α를 곱하고 β를 더하는 연산을 하여 출력됩니다. 학습되는 α, β는 공간적으로 변하기 때문에 wash away되는 현상을 방지합니다. ANL은 다양한 스케일로 연산되며 미세한 semantic 정보를 추출하기 때문에 보다 정확하게 편집이 가능합니다.

Literatur

[1] Cheng-Han Lee u. a. „MaskGAN: Towards Diverse and Interactive Facial Image Manipulation“. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Juni 2020.

[2] Shu-Yu Chen u. a. „DeepFaceEditing: Deep Face Generation and Editing with Disentangled Geometry and Appearance Control“. In: *arXiv preprint arXiv:2105.08935* (2021).

[3] Guilin Liu u. a. „Image inpainting for irregular holes using partial convolutions“. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, S. 85–100.

[4] Zhentao Tan u. a. „MichiGAN: multi-input-conditioned hair image generation for portrait editing“. In: *arXiv preprint arXiv:2010.16417* (2020).

[5] Taesung Park u. a. „Semantic image synthesis with spatially-adaptive normalization“. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, S. 2337–2346.

[6] David Bau u. a. „Semantic photo manipulation with a generative image prior“. In: *arXiv preprint arXiv:2005.07727* (2020).

[7] Jun-Yan Zhu u. a. „Generative visual manipulation on the natural image manifold“. In: *European conference on computer vision*. Springer. 2016, S. 597–613.

[8] Tero Karras u. a. „Progressive growing of gans for improved quality, stability, and variation“. In: *arXiv preprint arXiv:1710.10196* (2017).

[9] Tao Yu u. a. „Region normalization for image inpainting“. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Bd. 34. 07. 2020, S. 12733–12740.

[10] Hongyu Liu u. a. „PD-GAN: Probabilistic Diverse GAN for Image Inpainting“. In: *arXiv preprint arXiv:2105.02201* (2021).

[11] Hao Tang, Song Bai und Nicu Sebe. „Dual Attention GANs for Semantic Image Synthesis“. In: *Proceedings of the 28th ACM International Conference on Multimedia*. 2020, S. 1994–2002.

[12] Yi Wei u. a. „MagGAN: High-Resolution Face Attribute Editing with Mask-Guided Generative Adversarial Network“. In: *Proceedings of the Asian Conference on Computer Vision*. 2020.

[13] Ming Liu u. a. „STGAN: A unified selective transfer network for arbitrary image attribute editing“. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, S. 3673–3682.

[14] Changqian Yu u. a. „Bisenet: Bilateral segmentation network for real-time semantic segmentation“. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, S. 325–341.

[15] Haoye Dong u. a. „Fashion editing with adversarial parsing learning“. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, S. 8120–8128.