[Zhang18-ECCV] Generative Adversarial Network with Spatial Attention for Face Attribute Editing

Presenter : Ji-In Kim

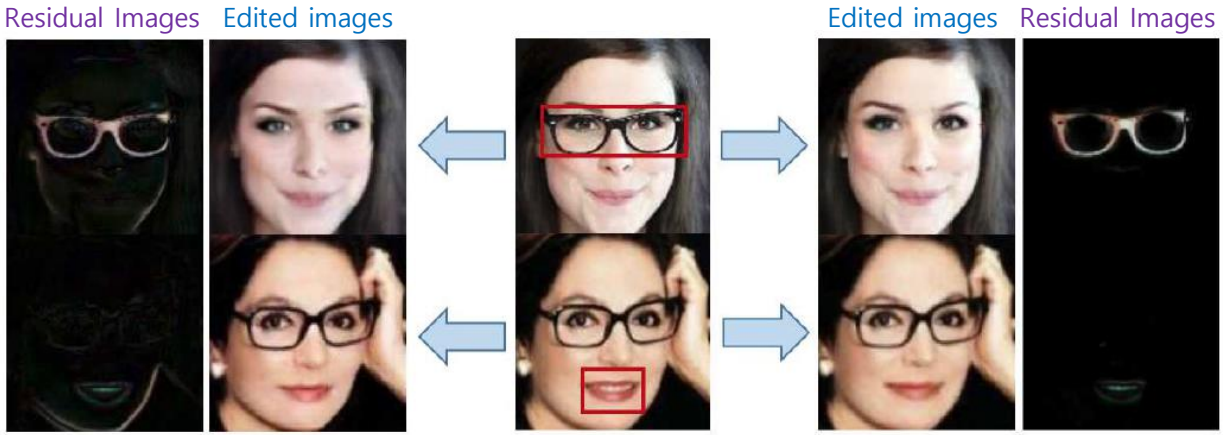
Contents

My opinion

Abstract

Abstract

- Face attribute editing 의 목표
  - 주어진 attribute 로 face image 를 수정하는 것.
- 대부분의 현존하는 방법들
  - Face attribute editing 에 GAN 을 사용한다. 그러나, 이 방법들은 attribute 와 관련 없는 부분들을 바꿔버린다.



(b) : Attribute 가 local 이지만, 이미지 전체가 바뀌었다.

(c) : 기댓값. Attribute-specific region 만 바뀌어야 하고 나머지는 바뀌면 안된다.

(Residual image = | Input face images - Edited face images |)

- Ours : GAN framework + Spatial attention mechanism
  - Attribute-specific region 만을 바꾼다!

1. Introduction

- Face attribute editing
  - Facial animation, Art, Entertainment, Face expression recognition 에 많이 사용된다. [1], [2], [3], [4]
- Face attribute editing 에서 바라는 결과
  - Attribute-specific region 만 바뀌고 나머지 region 은 바뀌지 않게 하자!

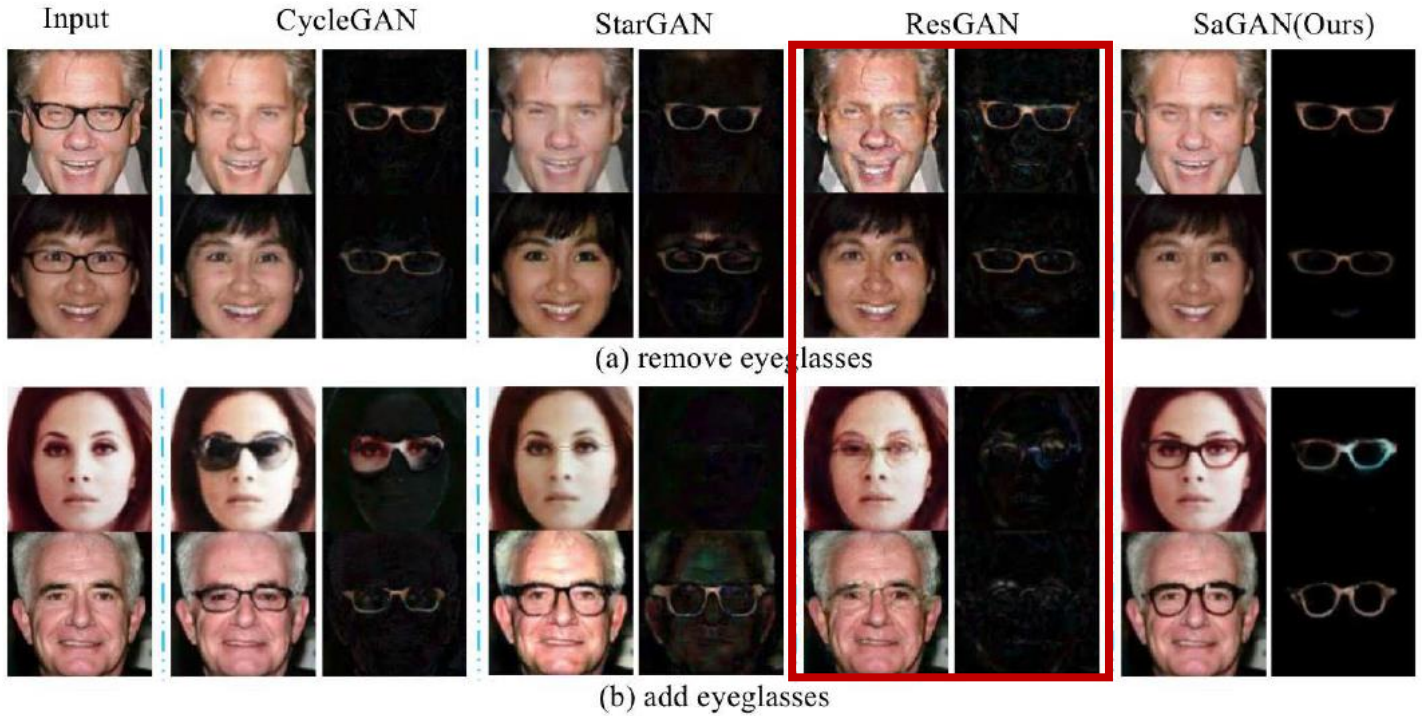
- 일찍이, Face attribute editing 을 paired training samples 를 사용하는 regression problem 으로 생각했다.
  - Zhu et al. [5]
    - Face frontalized method
  - Zhang et al. [6]
    - Removing eyeglasses
    - Paired training data 매우 의존한다.

- GAN 의 출현
  - GAN(Goodfellow et al. [7]) 은 많은 연구에서 매우 큰 성과를 보였고, Face attribute editing 역시 GAN 에게 많은 도움을 받는다.
  - GAN 접근법
    - Face attribute editing 을 unpaired image-to-image translation task 로 취급한다.
    - Original face image + given attribute → edited face image

GAN을 사용한 접근법들
Conditional GAN - Mirza, M., and Osindero, S., Conditional generative adversarial nets, Proc. of CoRR 2014, PP. 1411-1784, 2014.
IcGAN - Perarnau, G., Weijer, Joost., Raducanu, B., and Álvarez, J. M., Invertible conditional gans for image editing, Proc. of CoRR 2016, pp. 1611-06355, 2016.
CycleGAN - Zhu, J. Y., Park, T., Isola, P., and Efros, A. A., Unpaired image-to-image translation using cycle-consistent adversarial networks, Proc. of ICCV 2017, pp. 2223-2232, 2017
StarGAN - Choi, Y., Choi M. J., Kim, M., Ha, J. W., Kim, S., and Choo., J, Stargan: Unified generative adversarial networks for multi-domain image-to-image translation, Proc. of CVPR 2018, pp. 8789-8797, 2018.

- 그러나, 이 방법들은 모두 attribute-specific region 이 아닌, 이미지 전체가 바뀐다는 단점을 가진다.

- Shen et al. [17] (ResGAN)의 등장
  - Shen, W., and Liu, R., Learning residual images for face attribute manipulation, Proc. of CVPR, pp. *1225-1233, 2017*
  - Sparse residual images 를 학습
    - Residual image 의 대부분 영역을 0으로 만들어서 attribute 와 관련 없는 부분이 바뀌는 것을 피한다.
  - 단점
    - Target attributes 의 location 과 appearance 를 하나의 sparse residual image 에서 얻는다. 이것은 location 과 appearance 를 따로 modeling 하는 것보다 최적화하기가 어렵다.
    - Local attribute 일지라도 residual image 에서 나타나는 반응이 이미지 전체에 퍼져 있다.



- ResGAN [17] 에서 영감을 받아, 더 정교한 face attribute editing 을 위해 GAN 에 spatial attention mechanism 을 도입했다.
- Spatial attention mechanism
  - GAN 의 더 빠르고 정확한 처리를 위해 찾고자 하는 부분을 먼저 찾고 나머지 부분은 무시한다.
  - Image classification [18], [19], [20] 과 semantic segmentation [21] 등에서 성공적으로 수행되었다.
  - Spatial-specific regions 내에서 manipulation 을 제한하기 위해 사용한다.

- Contribution

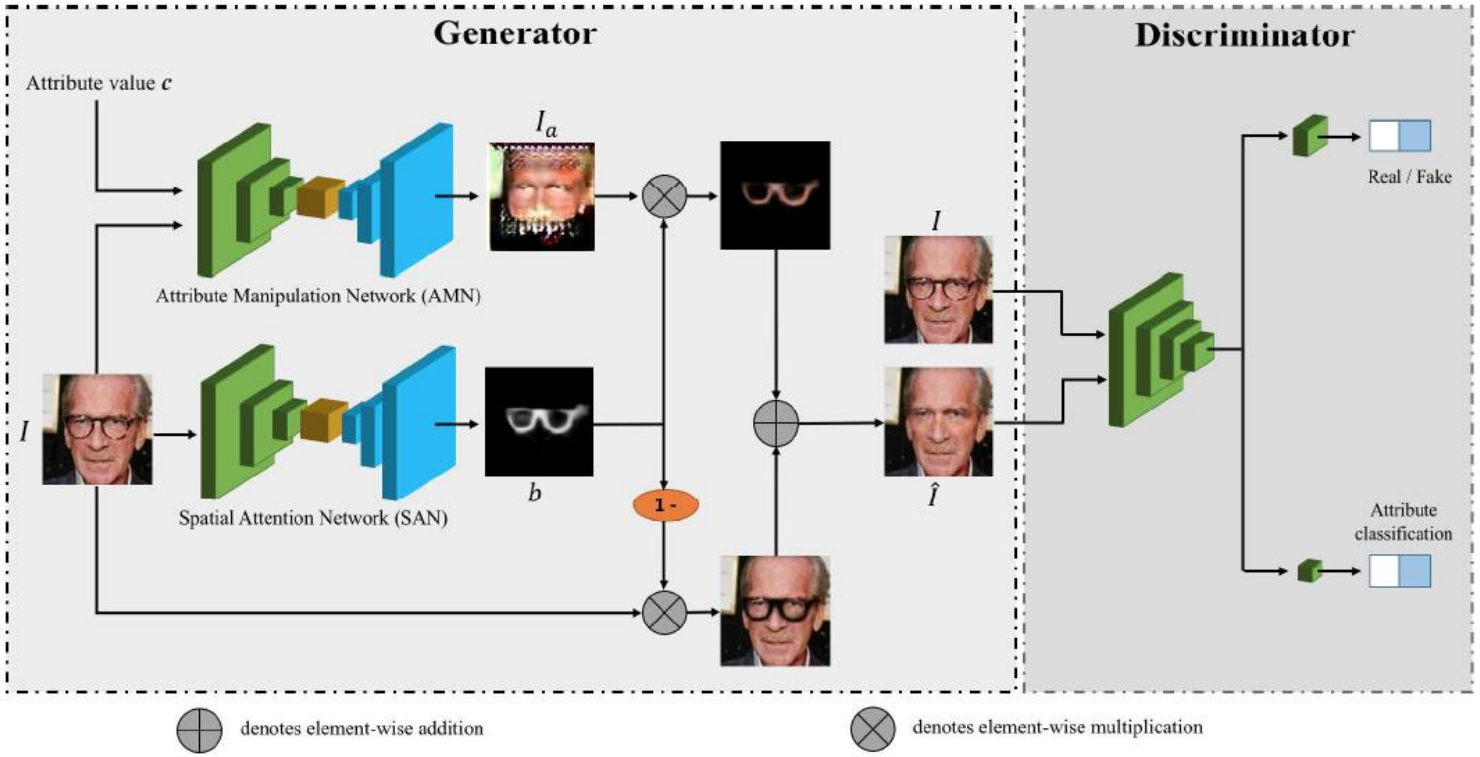
I. Gan framework 에 spatial attention 을 도입했다. (SaGAN)

II. SaGAN 은 두 개의 inverse face attribute editing 을 위한 dual generators 가 아닌 attribute 를 conditional signal 로 사용하는 단일 generator 를 가진다.

III. SaGAN 은 꽤 좋은 결과를 낸다. 특히 attribute 와 관련 없는 부분이 잘 보존된다. 또한 data augmentation 에 의한 face recognition 에 이점이 있다.

2. Generative Adversarial Network with Spatial Attention

• SaGAN 의 Overview



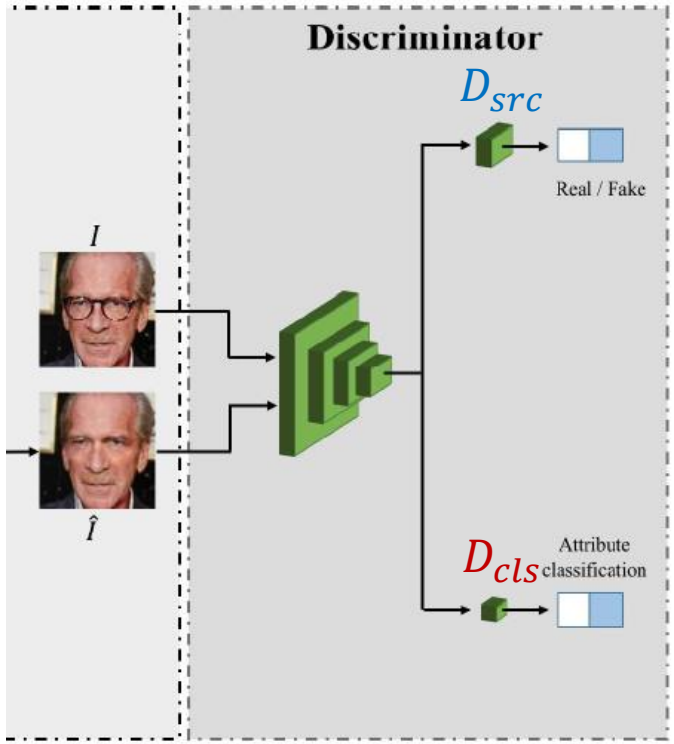
- Generator G
  - Attribute manipulation network (AMN) : 주어진 attribute 로 face image 를 수정한다.
  - Spatial attention network (SAN) : Attribute-specific region 의 위치를 알아낸다.
- Discriminator D
  - Real ones 로부터 generated images 를 구분한다.
  - Face attribute 를 분류한다.

- Face attribute editing 의 목적
  - Input image I 와 attribute value c 로 새로운 image $\hat{I}$ 를 만들자.
  - Attribute c 를 가진 $\hat{I}$
    1) 현실적이어야 한다.
    2) Attribute-specific region 을 제외한 나머지 부분은 input image 와 같아야 한다.

2.1. Discriminator

- Discriminator D 의 목적
  - Real ones 로부터 generated images 를 구분한다.
  - Generated images 와 real images 의 attribute 를 분류한다.

2.1. Discriminator



- Classifiers (D_{src} , D_{cls})
  - Softmax function + CNN
  - 두 개의 networks 는 처음 몇 개의 convolutional layers 를 공유한다. 뒤이어, 다른 분류를 위해 서로 다른 fully-connected layers 가 온다.
  - Input image : real images 또는 generated images
  - $D_{src}(I)$
    - Real/Fake classifier
    - Output : Image 가 real one 인지에 대한 확률
  - $D_{cls}(c|I)$
    - Attribute classifier
    - Output : Image I 가 attribute c 를 가질 확률
    - $c \in \{0, 1\}$

- **Loss function for Discriminator**

1. Real/Fake classifier
   - Standard cross-entropy loss

$$\mathcal{L}_{src}^{D} = \mathbb{E}_{I}[\log D_{src}(I)] + \mathbb{E}_{\hat{I}}[\log(1 - D_{src}(\hat{I}))], \quad (1)$$

I : Real image
$\hat{I}$: Generated image

2. Attribute classifier
   - Standard cross-entropy loss

$$\mathcal{L}_{cls}^{D} = \mathbb{E}_{I,c^g}[-\log D_{cls}(c^g|I)], \quad (2)$$

c^g : Real image I 의 ground truth attribute label

- Overall objective function

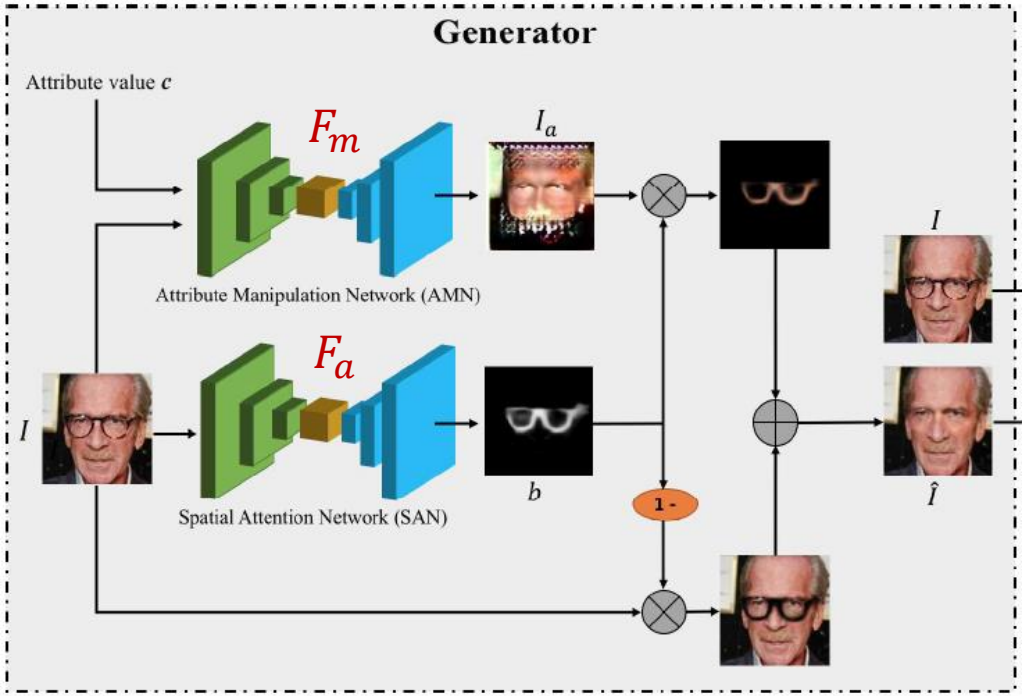
$$\min_{D_{src},D_{cls}} \mathcal{L}_D = \mathcal{L}_{src}^{D} + \mathcal{L}_{cls}^{D}. \quad (3)$$

2.2. Generator

- Generator G : Input face image I + attribute c → edited face image $\hat{I}$

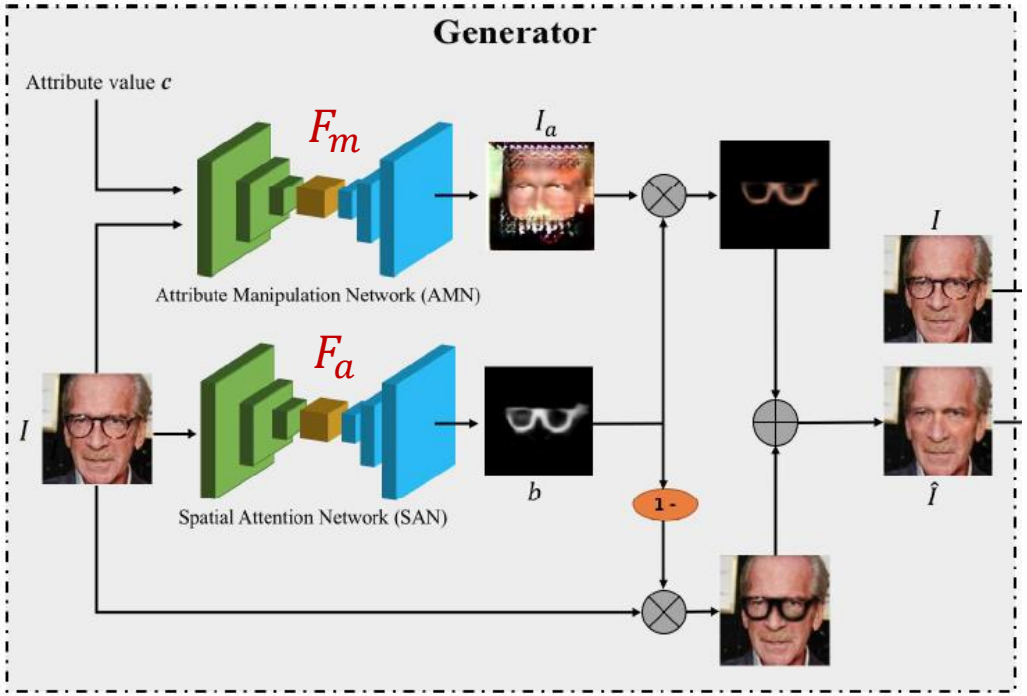
$$\hat{I} = G(I, c), \tag{4}$$

- G 는 두 개의 modules 를 가진다.
  - Attribute manipulated network (AMN) : F_m
    - "어떻게 manipulate 할까?"
  - Spatial attention network (SAN) : F_a
    - "어디를 manipulate 할까?"



- Attribute manipulated network (AMN) $$I_a = F_m(I, c). \quad (5)$$
- Spatial attention network (SAN) $$b = F_a(I), \quad (6)$$
  - Face image I 를 input 으로 가지고 spatial attention mask b 를 예측한다.
  - b : AMN 의 alternation 을 제한하는데 사용된다.
  - Optimization 을 거치면, b에서 values 는 0과 1 사이의 연속적인 값이 된다.
    - Non-zero attention values → Attribute-specific region
    - Zero attention values → Attribute-irrelevant region

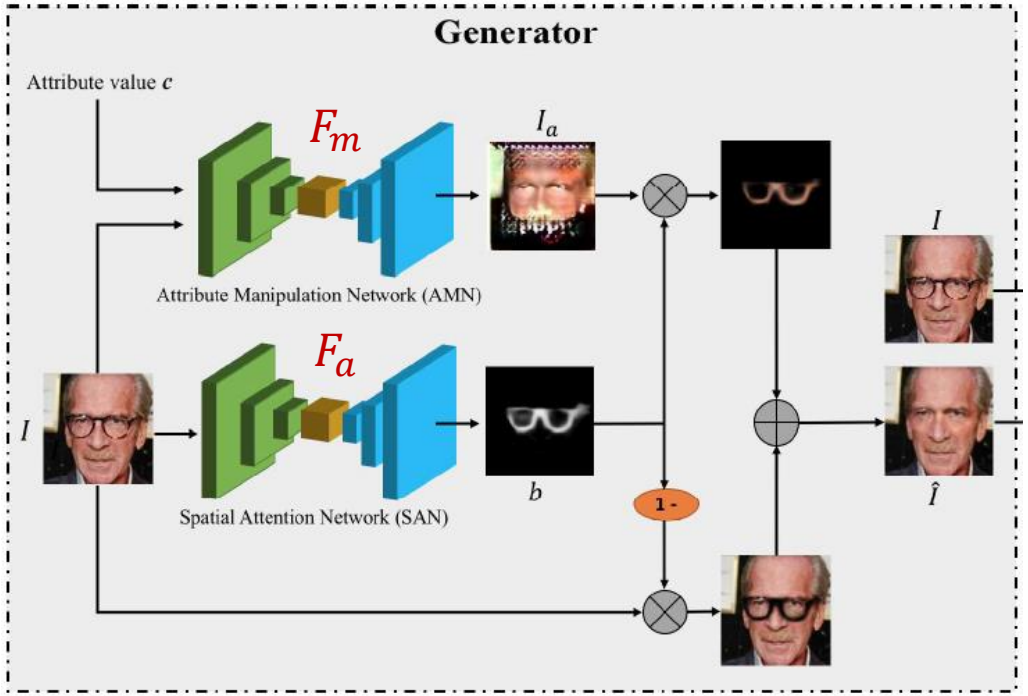
I_a에 대한 설명은 자세히 나와있지 않다.
F_m 은 어떤 이미지를 학습하기 원하는 것일까?



- 최종 edited face image $\hat{I}$
  - Attention mask 에 의해 guide 를 받는 attribute-specific regions 는 target attribute 를 향해 manipulate 되고, 나머지 regions 는 그대로 유지된다.

$$\hat{I} = G(I, c) = I_a \cdot b + I \cdot (1 - b), \tag{7}$$

spatial-specific region 은 I_a 와 곱하고,
나머지 부분은 I 와 곱하기



여기에는 pixel loss, perceptual loss 는 없다. loss 도 많아질 수록 overhead 이니까, 꼭 필요한 loss 들만 사용한 것일까?

- **Loss function for Generator**

1. Adversarial loss $\mathcal{L}_{src}^{G} = \mathbb{E}_{\hat{I}}[-\log D_{src}(\hat{I})]. \quad (8)$
   - Edited face image $\hat{I}$를 photo-realistic 하게 만들기 위해 대부분의 GAN-based methods 를 따라 만들었다.
2. Attribute classification loss $\mathcal{L}_{cls}^{G} = \mathbb{E}_{\hat{I}}[-\log D_{cls}(c|\hat{I})]. \quad (9)$
   - $\hat{I}$ 이 target attribute c 를 가진 것을 정확히 예측하기 위해 Attribute classifier 가 $\hat{I}$의 attribute 를 잘 예측할 수 있도록 설계했다.
3. Reconstruction loss $\mathcal{L}_{rec}^{G} = \lambda_1 \mathbb{E}_{I,c,c^g}[(\|I - G(G(I,c),c^g)\|_1] + \lambda_2 \mathbb{E}_{I,c^g}[(\|I - G(I,c^g)\|_1], \quad (10)$
   - Attribute-irrelevant region 이 바뀌지 않게 하기 위해 CycleGAN[12]와 StarGAN[16]과 유사하게 만들었다.
     - c^g : original attribute of input image I
     - λ_1, λ_2 : two balance parameters
     - First term : Dual reconstruction loss → $G(G(I,c),c^g)$ 는 original image I 와 같도록 기대된다.
     - Second term : Identity reconstruction loss → input image I 가 c^g에 의해 edit 될 때 수정되지 않도록 한다.
     - L1 norm → more clear reconstruction 을 위해 채택되었다.

- Overall objective function

F_m 과 F_a 는 같은 loss 를 사용한다.

$$\min_{F_m,F_a} \mathcal{L}_G = \mathcal{L}_{adv}^{G} + \mathcal{L}_{cls}^{G} + \mathcal{L}_{rec}^{G}. \quad (11)$$

- **CycleGAN[12]** vs **SaGAN**
  - 공통점
    - Adversarial loss, Dual reconstruction loss, Identity reconstruction loss 를 사용한다.
  - 차이점
    - CycleGAN
      - Whole image 에서 동작한다.
      - Cycle 구조를 사용해서 counter editing 을 한다.
    - SaGAN
      - Attribute-specific region 에만 집중한다.
      - 다른 conditional signal 을 가진 하나의 generator 로 counter editing 을 한다.

하나의 generator 를 사용하는 것은
모델이 가벼워져서 좋은 것일까?

- **StarGAN[16]** vs **SaGAN**
  - 차이점
    - StarGAN
      - Whole image 에서 동작한다.
      - 하나의 모델로 다수의 attributes 를 edit 할 수 있다. (StarGAN 의 장점)
    - SaGAN
      - Attribute-specific region 에만 집중한다.
      - 오직 하나의 attribute 만 edit 한다.

하나의 attribute 만 수정할 수 있는 것은 이 모델의 단점이 될 수 있다.

- **ResGAN[17]** vs **SaGAN**
  - 공통점
    - Attribute-specific region 만을 수정하고 나머지 region 은 바꾸지 않는 것을 목표로 한다.
  - 차이점
    - ResGAN
      - Sparse residual image 를 사용해 attribute-specific region 을 조정한다. Sparse residual image 는 sparsity constraint 를 통해 attribute-specific region 을 결정한다. Sparsity 정도는 control parameter 에 의존하고 attribute 자체에 의존하지는 않는다.
      - 두 개의 generators 를 사용해서 counter editing 을 한다.
    - SaGAN
      - Spatial attention network 로부터 예측된 attention mask 를 통해 attribute-specific region 을 결정한다. 이것은 attribute 에 맞추기 때문에, simple sparsity constraint 를 사용하는 것보다 정확하다.
      - 다른 conditional signal 을 가진 하나의 generator 로 counter editing 을 한다.

SaGAN 에서 attention mask 대신 residual image 를 사용하면 과연 성능이 어떻게 나올까?

3. Implementation Details

- **Optimization**
  - Adversarial real/fake classification 을 더 안정적으로 최적화하기 위해, 모든 실험에서 Eq(1) 과 Eq(8) 의 objectives 는 WGAN-GP[22] 를 사용해서 최적화된다.

$$\mathcal{L}_{src}^{D} = \mathbb{E}_{I}[\log D_{src}(I)] + \mathbb{E}_{\hat{I}}[\log(1 - D_{src}(\hat{I}))], \tag{1}$$

$$\mathcal{L}_{src}^{G} = \mathbb{E}_{\hat{I}}[-\log D_{src}(\hat{I})]. \tag{8}$$

⇩

$$\mathcal{L}_{src}^{D} = -\mathbb{E}_{I}[D_{src}(I)] + \mathbb{E}_{\hat{I}}[D_{src}(\hat{I})] + \lambda_{gp}\mathbb{E}_{\tilde{I}}[(\|\nabla_{\tilde{I}} D_{src}(\tilde{I})\|_2 - 1)^2], \tag{12}$$

$$\mathcal{L}_{src}^{G} = -\mathbb{E}_{\hat{I}}[D_{src}(\hat{I})], \tag{13}$$

- $\tilde{I}$ 는 edited images $\hat{I}$ 와 real image I 사이에서 직선을 따라 균일하게 sampled 된 결과이다.
- λ_{gp} 는 gradient penalty 의 계수이다. $\lambda_{gp} = 10$

• Network Architecture

• Generator

• AMN 과 SAN 은 input 과 output 에서 약간의 차이만 제외하고 같은 network 구조를 공유한다.

차이점

	Attribute Manipulate Network (AMN)	**Spatial Attention Network (SAN)**
Input	Four-channel tensor (Input image + attribute value)	Input image
Output	Three-channel RGB image	Single channel attention mask image
Activation function	Input image 가 [-1, 1]로 정규화 되었기 때문에 Tanh 를 사용	Attention 이 [0, 1] 이기 때문에 sigmoid 를 사용

Generator G의 구조

Layer	Attribute Manipulation Network (AMN)	Spatial Attention Network (SAN)
L1	Conv(I4,O32,K7,P3,S1),IN,ReLU	Conv(I3,O32,K7,P3,S1),IN,ReLU
L2	Conv(I32,O64,K4,P1,S2),IN,ReLU	Conv(I32,O64,K4,P1,S2),IN,ReLU
L3	Conv(I64,O128,K4,P1,S2),IN,ReLU	Conv(I64,O128,K4,P1,S2),IN,ReLU
L4	Conv(I128,O256,K4,P1,S2),IN,ReLU	Conv(I128,O256,K4,P1,S2),IN,ReLU
L5	Residual Block(I256,O256,K3,P1,S1)	Residual Block(I256,O256,K3,P1,S1)
L6	Residual Block(I256,O256,K3,P1,S1)	Residual Block(I256,O256,K3,P1,S1)
L7	Residual Block(I256,O256,K3,P1,S1)	Residual Block(I256,O256,K3,P1,S1)
L8	Residual Block(I256,O256,K3,P1,S1)	Residual Block(I256,O256,K3,P1,S1)
L9	Deconv(I256,O128,K4,P1,S2),IN,ReLU	Deconv(I256,O128,K4,P1,S2),IN,ReLU
L10	Deconv(I128,O64,K4,P1,S2),IN,ReLU	Deconv(I128,O64,K4,P1,S2),IN,ReLU
L11	Deconv(I64,O32,K4,P1,S2),IN,ReLU	Deconv(I64,O32,K4,P1,S2),IN,ReLU
L12	Conv(I32,O3,K7,P3,S1),Tanh	Conv(I32,O1,K7,P3,S1),Sigmoid

I : input channel 의 수
O : output channel 의 수
K : kernel size
P : padding size
S : strike size

- **Network Architecture**
  - Discriminator

Discriminator D의 구조

Layer	Discriminator	
L1	Conv(I3,O32,K4,P1,S2),Leaky ReLU	
L2	Conv(I32,O64,K4,P1,S2),Leaky ReLU	
L3	Conv(I64,O128,K4,P1,S2),Leaky ReLU	
L4	Conv(I128,O256,K4,P1,S2),Leaky ReLU	
L5	Conv(I256,O512,K4,P1,S2),Leaky ReLU	
L6	Conv(I512,O1024,K4,P1,S2),Leaky ReLU	
L7	*src*: CONV(I2014,O1,K3,P1,S1)	*cls*: CONV(I1024,O1,K2,P0,S1),Sigmoid

I : input channel 의 수
O : output channel 의 수
K : kernel size
P : padding size
S : strike size
IN : instance normalization

- **Training Settings**
  - 모든 models 의 parameters 는 mean=0 이고 standard deviation=0.02 인 normal distribution 에 따라 randomly 하게 초기화
  - Optimizer : $\beta_1 = 0.5$, $\beta_2 = 0.999$인 Adam
  - Learning rate : $lr = 0.0002$
  - Generator 의 reconstruction loss 에서 $\lambda_1 = 20$, $\lambda_2 = 100$
  - batch size = 16
  - Generator 는 1번 update, discriminator 는 3번 update

4. Experiments

4.1. Datasets

- CelebA
  - 10,177 명의 유명인사들이 있는 202,599 개의 얼굴 이미지
  - 각 face image 는 40 binary attributes 로 표기
  - 공식적으로 crop 된 128x128 이미지를 사용
  - Training 에는 8,177 people, Testing 에는 2,000 people
  - Training data : 191,649 images, testing data : 10,950 images
- LFW
  - SaGAN 의 일반화를 testing 하기 위해 사용
- 4개의 attributes 를 대표로 사용했다. *eyeglasses, mouth_slightly_open, smiling, no_beard*

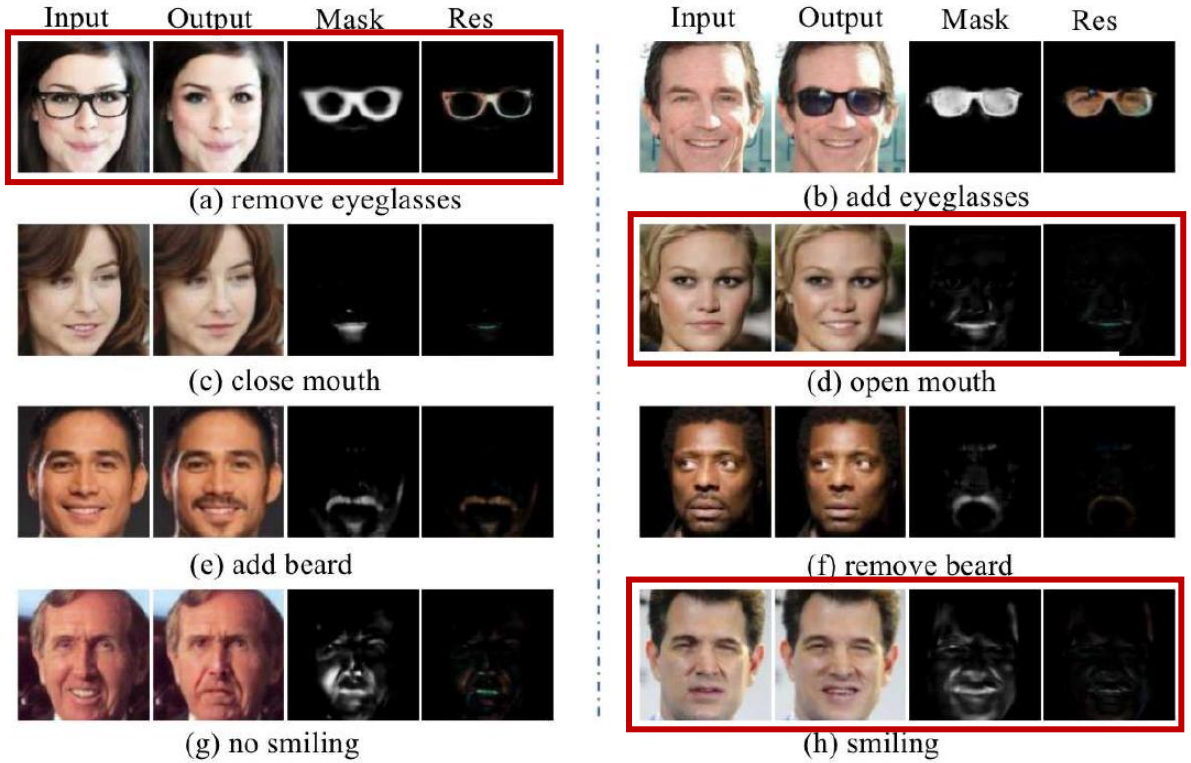
4개를 제외한 다른 attribute 를 사용하면 결과가 어떻게 나올까?

4.2. Visual Comparison on face attribute editing

- **Investigation of SAN**
  - Spatial attention network (SAN)은 attribute-specific region 의 위치를 알아내는 것을 목표로 한다.
  - SAN 이 어떻게 face attribute editing 에 기여하는지 알기 위해 spatial attention masks 를 시각화 한다.

Experiments

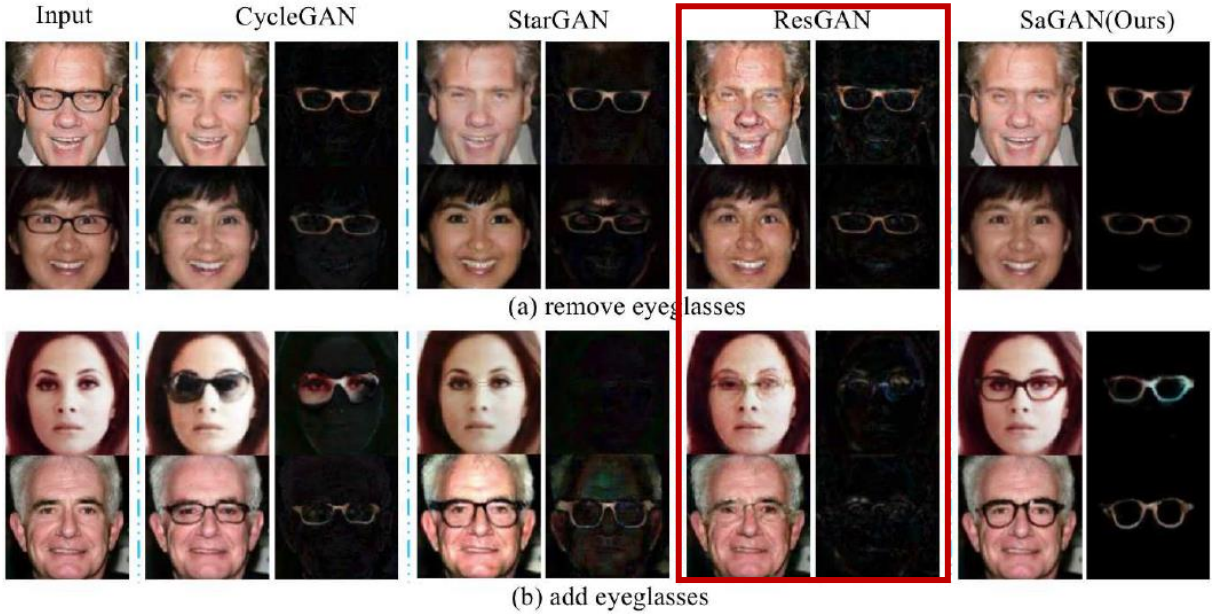
- CelebA dataset 사용
- Mask : Spatial attention mask
- Res : residual images



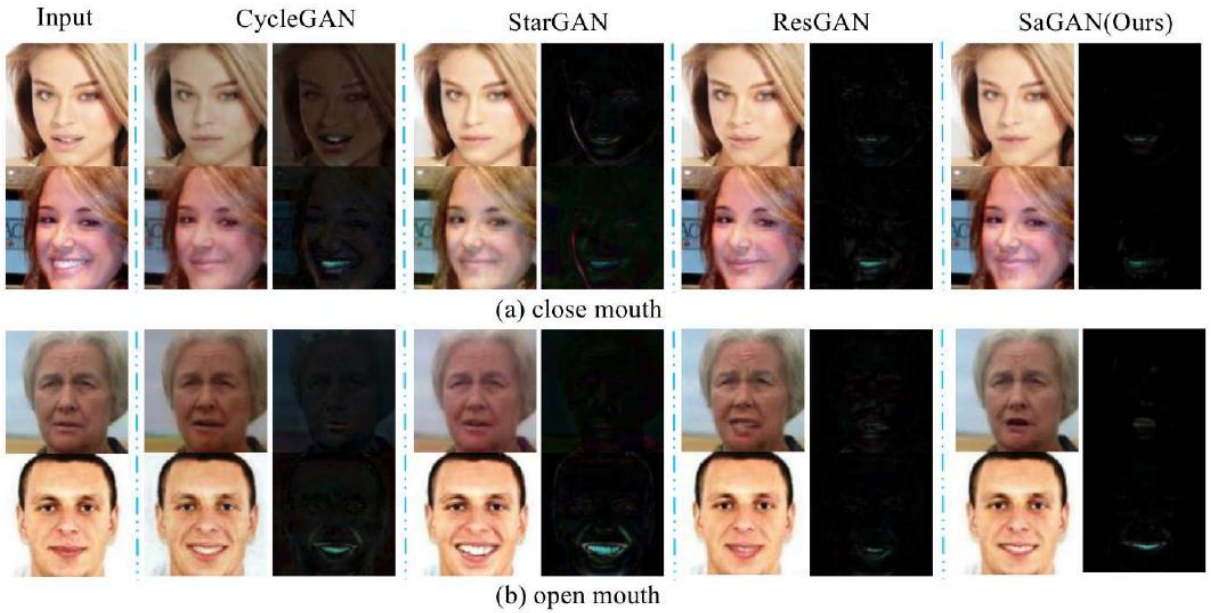
- Spatial attention masks 는 주로 attribute-specific regions 에만 집중하는 것을 볼 수 있다.
- *eyeglasses* 와 같은 local attribute 에서, Spatial attention masks 는 오직 eyes 주변에서만 반응한다.
- *mouth open, smiling* 과 같은 global face 의 이동이 개입된 attribute 는, spatial attention 이 얼굴의 더 큰 부분에서 반응한다. 이것은 Spatial attention network 가 attribute 에 따라 attribute-specific regions 를 결정한다는 것을 보여준다.

Visual results on CelebA

eyeglasses

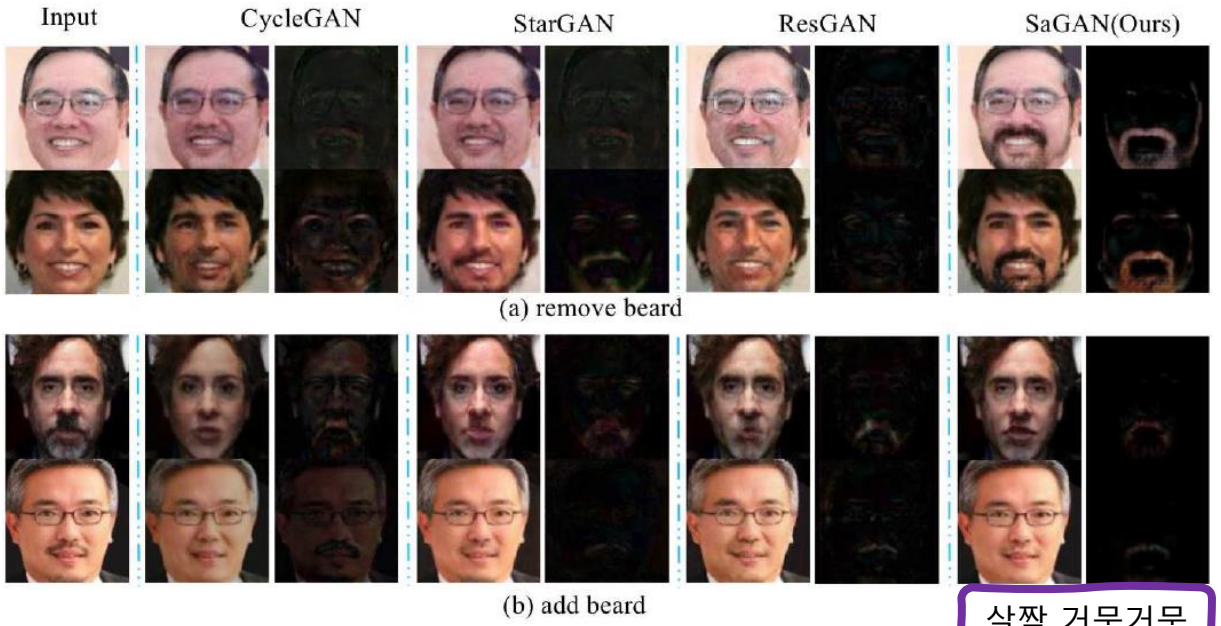


mouth_slightly_open



- CycleGAN, StarGAN, ResGAN 과 비교해서, SaGAN 은 대부분의 attribute 와 관계없는 영역이 바뀌지 않고 보존된다.
- ResGAN 에서 나온 attribute-specific regions 에서(특히 eyeglasses) artifacts 가 존재하는 것을 볼 수 있다.
- SaGAN 은 attribute-specific region 에서 양호한 조정을 하고, attribute 와 관계없는 나머지 부분은 바꾸지 않고 잘 보존한다.
- 그 이유는 SaGAN 의 generator 가 spatial attention module SAN 을 가지기 때문이다.

no_beard



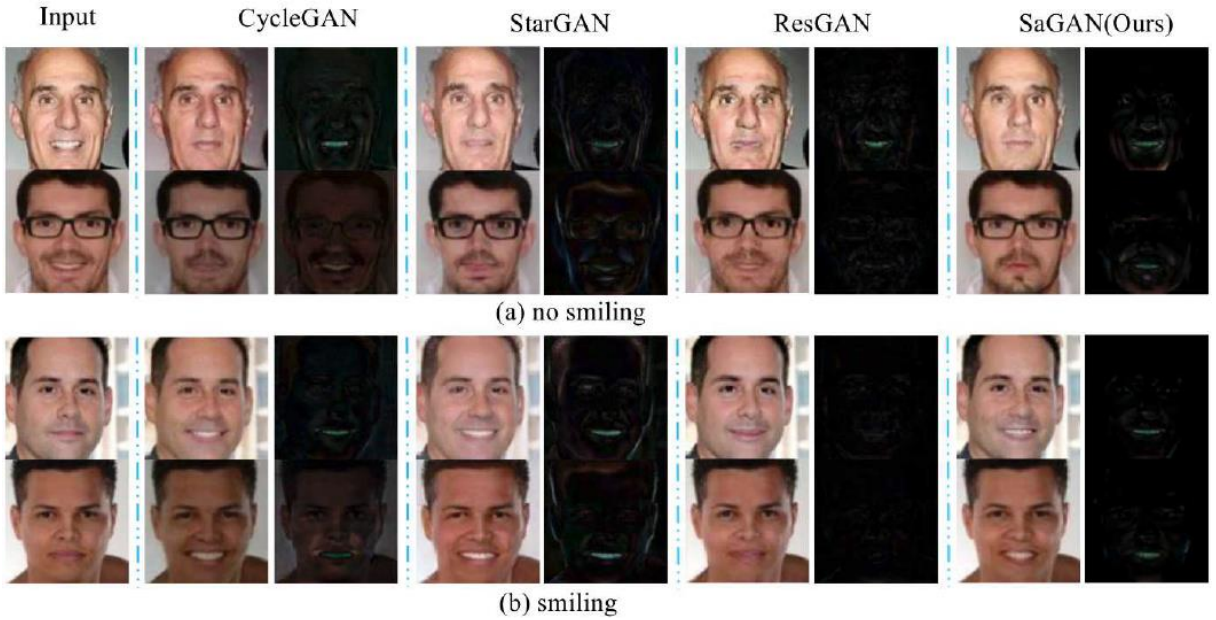
눈썹도 같이 남성스럽게 변했다.

(a)와 (b)과 바뀐 것 같다.

살짝 거뭇거뭇

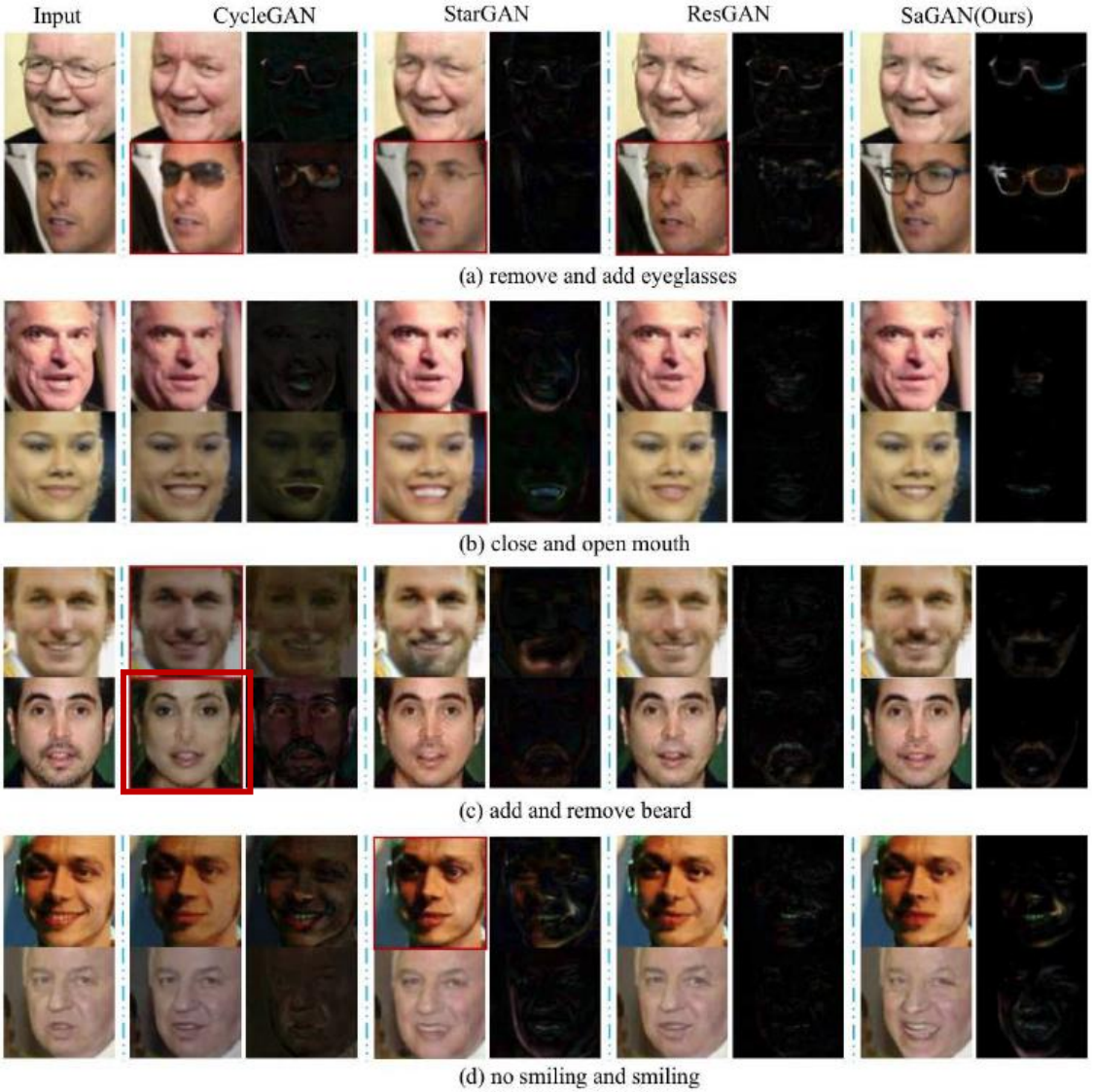
- *no_beard* 는 gender 와 관련이 있기 때문에 모든 방법들은 부득이하게 input face 의 성별을 바꾼다.
- 그래도 SaGAN 은 images 를 적당히 수정한다. e.g. 볼과 턱 너머 대부분의 영역을 보존한다.

smile



- 여기서도 역시, SaGAN 은 더 나은 visual quality 를 보여준다.

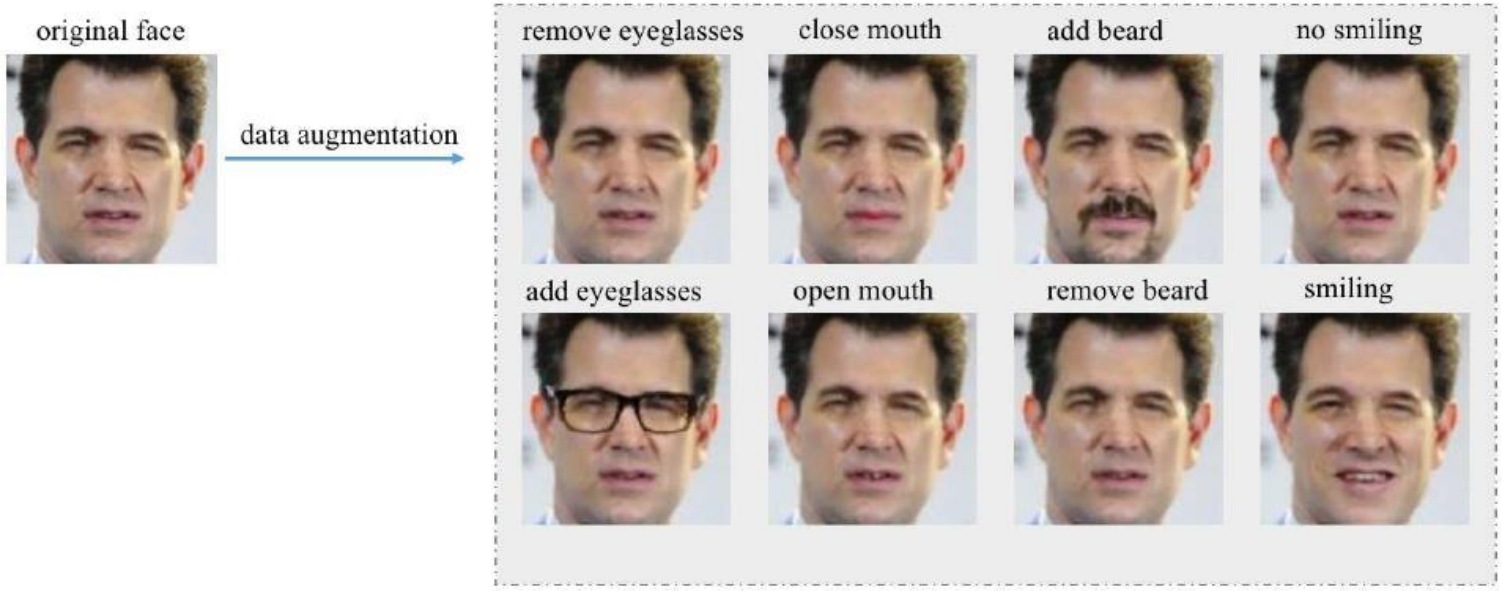
• Visual results on LFW



- SaGAN 의 일반화 성능을 조사하기 위해 CelebA 에서 훈련된 model 은 LFW 에서 평가되었다.
- CycleGAN, StarGAN, ResGAN 의 방법들은 LFW 에서 결과가 좋지 않은 것을 볼 수 있다.
- CycleGAN 은 beard 를 제거했을 때 male 에서 female 로 바뀐다.
- 놀랍게도, SaGAN 은 CelebA 에서처럼 좋은 결과를 낸다.

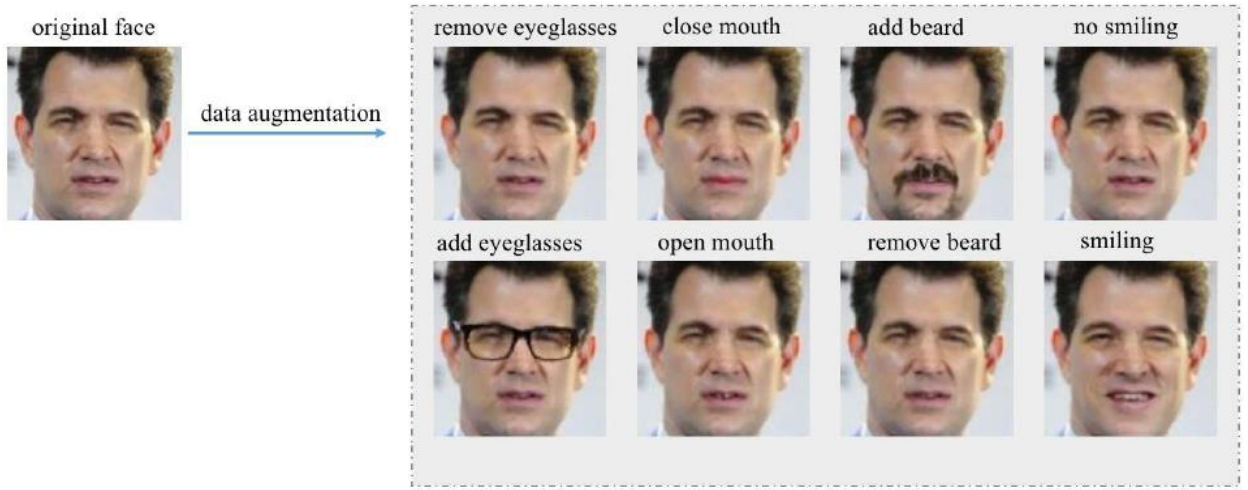
4.3. Comparison on face recognition

- 좋은 시각적 editing 결과를 볼 때, 자연스러운 생각은 data augmentation 의 결과로 나온 이미지들(with SaGAN)로 수행한 face recognition 의 성능을 확인해보는 것이다.
- 이것을 조사하기 위해, 우리는 각 training sample 에 대해서 attribute 를 수정해서 augment 했다.



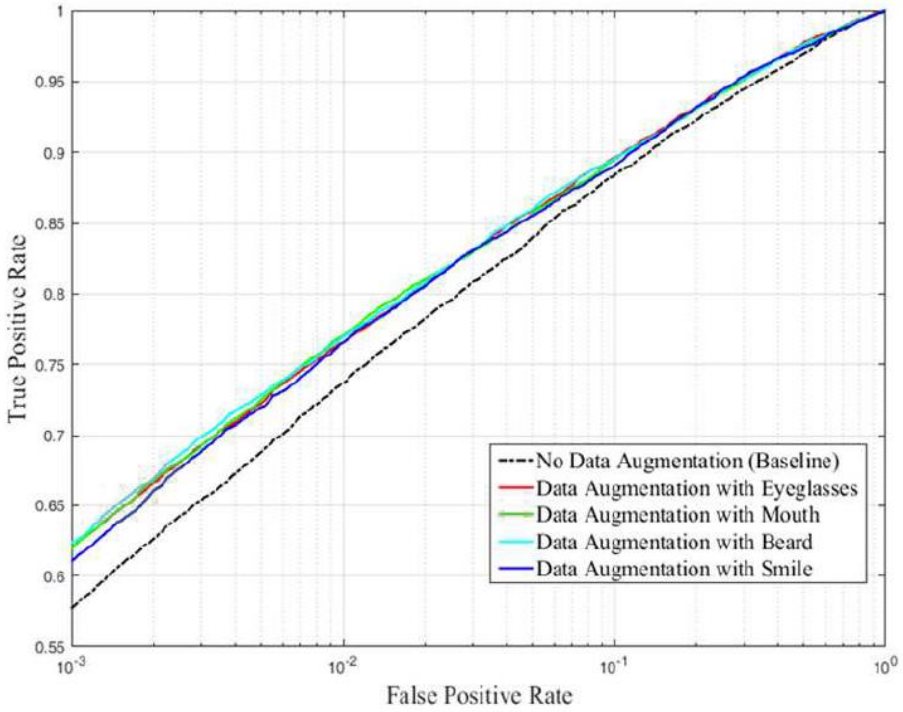
An Example of Augmentation

Experiments



- 실제로, own attribute 로 edit 된 face image 는 original image 와 거의 같다. Original attribute 로 Augmenting 을 한 이유는 단지 이미지의 attribute 를 따로 분류할 필요가 없는 간단함 때문이다.
- 사용한 Face recognition model : ResNet-18[17]
- Test dataset : CelebA 의 test sets, LFW
  - CelebA 에서 하나의 face image 가 randomly 하게 target 으로 선택되고 나머지는 query 로 선택된다.
  - LFW 에서 standard protocol 이 사용된다.
- 두 개의 datasets 에서, 성능은 ROC curves 로 나타낸다.

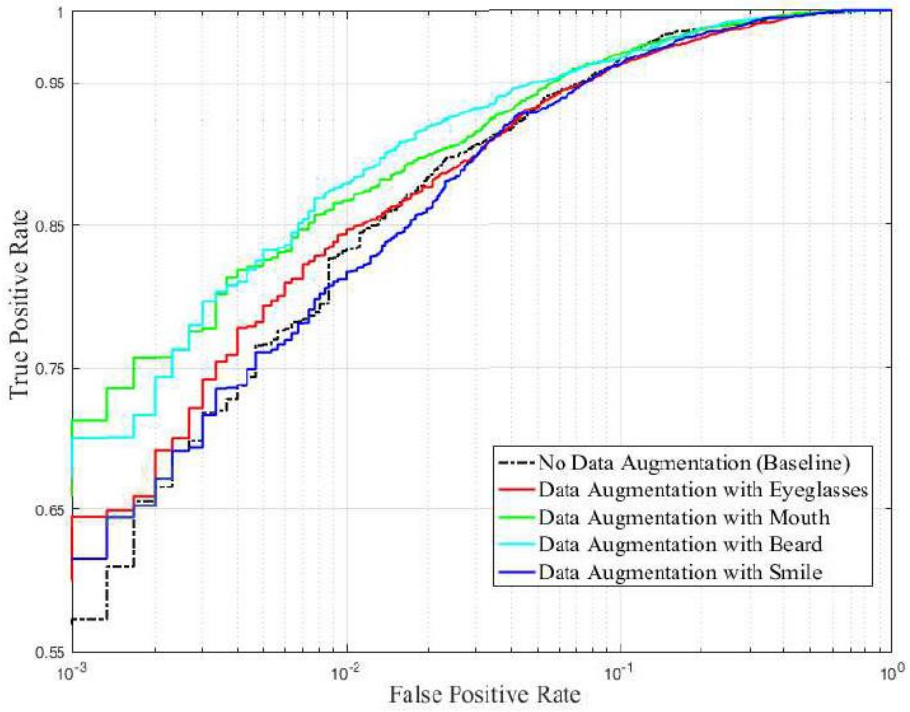
- Data augmentation 을 하지 않은 baseline model 보다 Data augmentation 을 한 model 이 더 나은 수행을 한다.
- SaGAN 으로 만들어진 augmentation 결과는 robust model 을 만들고 training data 의 variation 을 높인다.



Face verification on CelebA

- *Smile* 을 제외한 모든 attributes 로 data augmentation 한 model 은 baseline model 보다 더 낫다. 이것은 SaGAN 이 face verification 에 이점이 있다는 것을 설명한다.
- *Smile* 로 augmentation 한 결과의 성능이 좋지 않은 이유는 test data 에서 smile faces 가 적고 model 을 *Smile* 에 편향되게 만들어 성능 저하를 이끈다.

Data set 에서 attribute 의 분포는 얼굴 인식 성능을 결정하는 중요한 기준이다.



Face verification on LFW

5. Conclusions and Future Works

- **Conclusions**
  - GAN framework 에 spatial attention mechanism 을 도입하고, 더 정확한 face attribute editing 을 위해 SaGAN 을 만들었다.
  - Spatial attention mechanism 은 오직 attribute-specific regions 에서만 manipulate 되고 나머지 attribute 와 관련 없는 부분은 바뀌지 않는 것을 보장한다.
  - CelebA 와 LFW 에서의 실험은, SaGAN 이 현존하는 face attribute editing methods 보다 더 나은 성능을 낸 다는 것을 설명한다.
  - SaGAN 은 또한 data augmentation 을 통해 face recognition 에 이득을 준다.
- **Future works**
  - General image editing tasks 에 SaGAN 을 적용해 볼 것이다.

Thank you for listening