

[Shen17-CVPR] Learning Residual Images for Face Attribute Manipulation

Presenter : Ji-In Kim

Contents

1. Introduction
2. Related work
3. Learning the Residual Images
4. Datasets
5. Implementation Details
6. Experiments
7. Discussion
8. Conclusion

1. Introduction

- 과거의 Face image processing

빨강 : 딥러닝 전 / 파랑 : 딥러닝 후

- Age analysis : [22], [26]
- Emotion detection : [1], [5]
- Attribute classification : [4], [20], [15], [18]



Attributes 를 추론하는 것에 집중

- Ours

- Face attribute manipulation

Introduction



(a) *Glasses*: remove and add the glasses



(b) *Mouth_open*: close and open the mouth

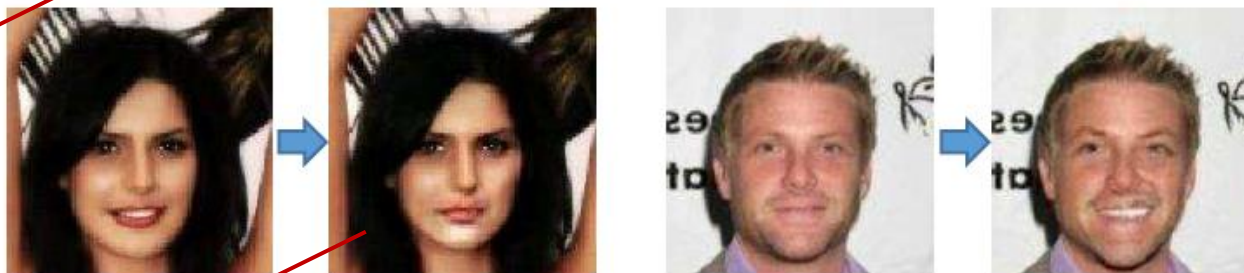


(c) *No_beard*: add and remove the beard



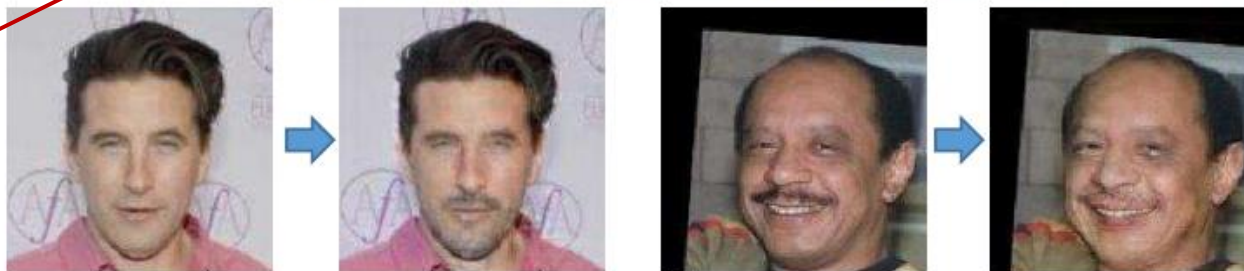
(a) *Glasses*: remove and add the glasses

안경이 깔끔하게
제거되지 않았다.



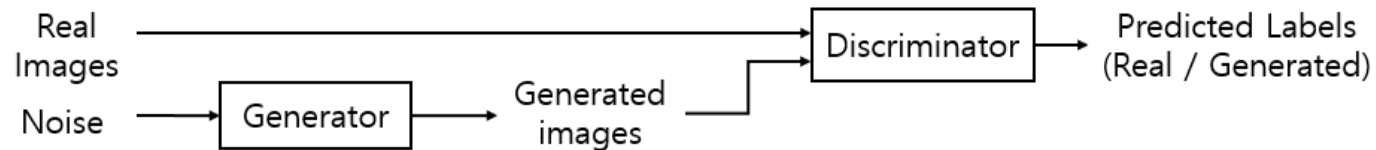
(b) *Mouth_open*: close and open the mouth

턱 밑에 하얀색의
noise 가 존재한다.

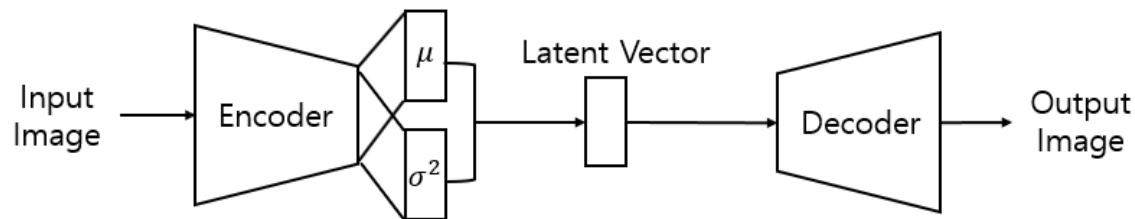


(c) *No_beard*: add and remove the beard

- Image 생성에는 GANs 와 VAEs 를 사용한다.
- GAN models
 - GAN 으로 만들어진 image 는 현실적이다.
 - 그러나, GAN models 는 image generation 에 random noise 를 사용하기 때문에 images 를 encode 할 수 없다.



- VAE models
 - VAE models 는 image 를 latent representation 으로 encode 할 수 있다.
 - 그러나, Encoder-Decoder pipeline 을 통과한 image 는 reconstruction quality 가 좋지 않다.



- GAN 과 VAE 를 사용해서 Face attribute manipulation 을 할 때, 손실된 details 는 identity 와 관련된 것일 수 있다. 이것은 곧 undesired changes 를 만든다.
- 따라서, GAN models 와 VAE models 를 Face attribute manipulation 에 직접적으로 적용 하는 것은 어렵다.

- GAN 은 noise 를 다양하게 주어 얼굴을 만든다. 그래서 원하는 특정한 얼굴이 나오는 것이 아니라 새로운 얼굴이 만들어진다.
- 이 논문은 attribute 를 바꾸는 것으로 GAN 처럼 엉뚱한 사람이 튀어나오면 안된다.
- 이 논문에서 원하는 것은 "1) Identity 유지하기 2) Attribute 를 control 하기 " 이다.
- 그러므로, GAN 과 VAE 을 직접적으로 사용하는 것은 이 논문이 원하는 것과 거리가 멀다고 할 수 있다.

- 한가지 대안은, Face manipulation 을 transformation process 로 보는 것이다.
- Transformation process
 - Original images 가 input 으로 들어가면 explicit embedding 없이 transformed images 가 출력된다. feed-forward CNN 을 사용한다.

- 이 논문에서, Face attribute manipulation 을 위해 residual image learning 을 사용한 방법을 제안한다.
- ➔ GAN 의 generative power 와 feed-forward network(CNN) 의 efficiency 를 합치는 방법이다.

- Our Contribution

1. Residual images 를 학습했다.

- Residual image : original input image 와 desired manipulated image 의 차이
- 불필요하고 관련 없는 details 를 가진 entire face 가 아닌, 특정 attribute 에 초점을 맞춘다.

2. Dual learning scheme 를 만들었다.

- Dual learning process 는 high quality images 를 만드는 데에 좋다.

3. Landmark detection accuracy gain 을 metric 으로 사용했다.

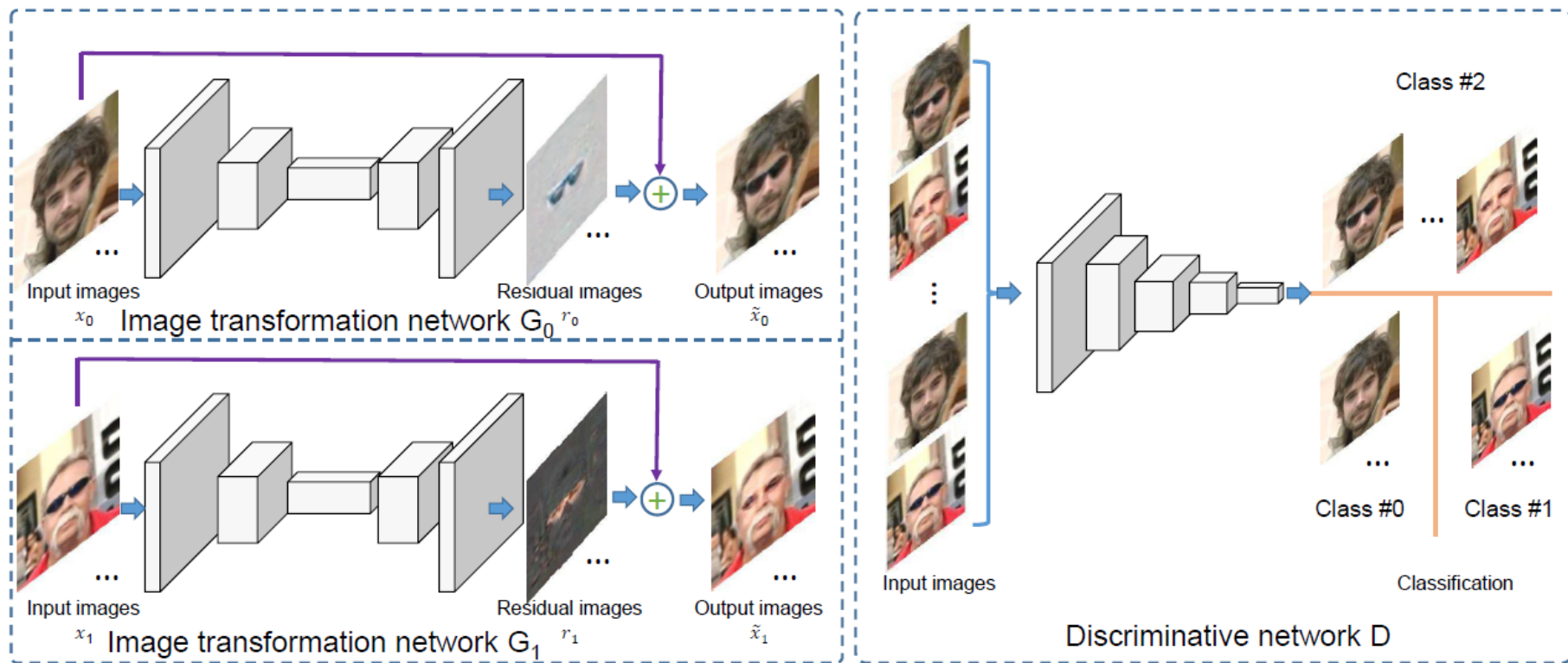
- Removal glasses 가 효과적이라는 것을 양적으로 보여준다.

2. Related Work

- Image generation 을 위한 많은 기술들
 - [23], [2], [17], [8], [3], [14]
- Li *et al.* [19]
 - M. Li, W. Zuo, and D. Zhang. Deep identity-aware transfer of facial attributes. *arXiv preprint arXiv:1610.05586*, 2016.
 - Face attributes 의 identity 를 인식하기 위해 deep convolutional network model 을 제안했다.
- Our work 와 [19]의 차이
 1. 우리의 방법은 residual images 를 사용해서 manipulated images 를 만든다.
 2. 우리의 방법은 동작하는 동안 하나의 discriminator 를 공유함으로써 한 개의 single architecture 안에서 두 개의 inverse manipulations 를 만든다. [19] 는 각 manipulation 을 독립적으로 취급한다.
 3. 우리의 방법은 [19]에서는 필수적인 작업인 post-processing 을 필요로 하지 않는다.

3. Learning the Residual Image

Learning the Residual Image



- G_0 : Primal manipulation
- G_1 : Dual manipulation
- D : 3개의 카테고리로 reference images 와 generated images 를 분류한다.

3.1. Generative Adversarial Networks

- GAN
 - Generative model G
 - Training data 를 복구하기 원한다.
 - D 를 속인다.
 - Discriminative model D
 - Real image 인지 Fake image 인지 구별한다.

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)} [\log D(\mathbf{x})] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))]. \quad (1)$$

- GAN
 - Realistic images 의 data distribution 을 효과적으로 학습할 수 있다.
 - Desired attribute 로 images 를 생성할 수 있다.

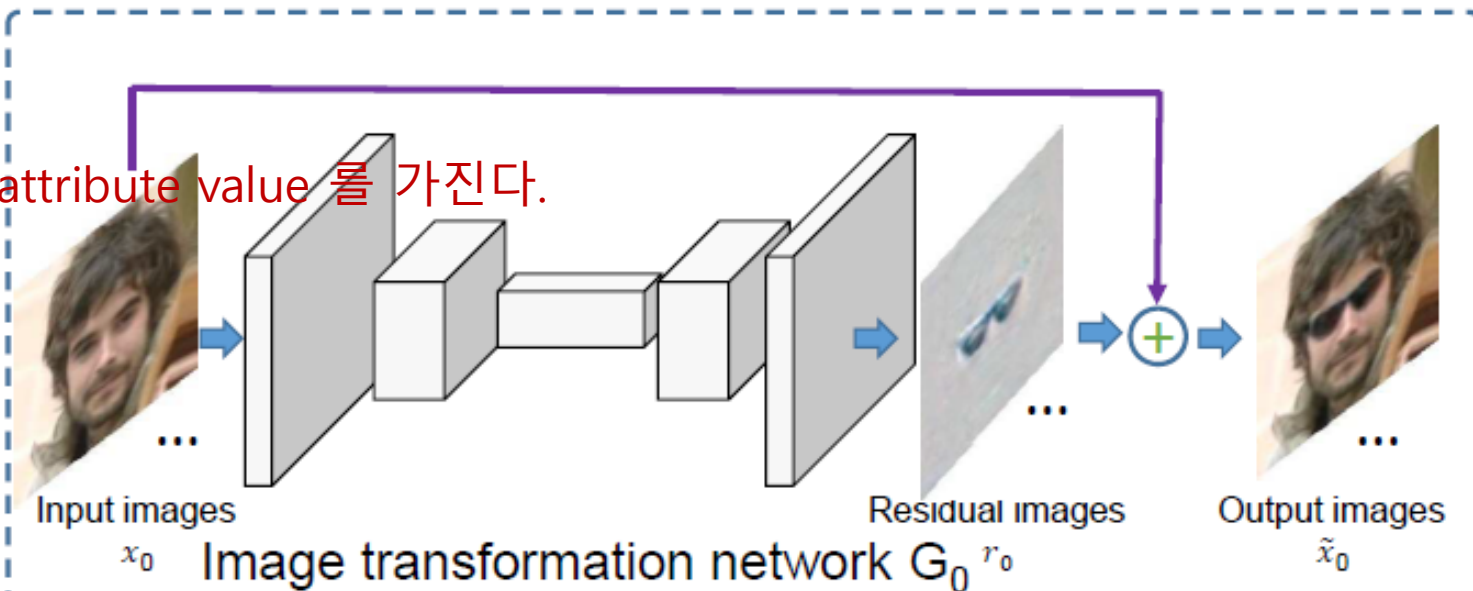
3.2. Image Transformation Networks

- Motivation
 - Attribute-specific face area 에서는 manipulation 이 되고, 나머지 부분은 바뀌지 않았으면 한다.
 - ex) Face image 에서 안경을 제거할 때
 - 안경이 있는 부분만 skin 또는 eyes 로 바뀌고 mouth, nose, hair 와 같은 다른 부분들은 바뀌지 않는다.
- 그러므로, 우리는 attribute-specific area 를 대상으로 한 residual image 를 학습하는 manipulation 을 만든다.

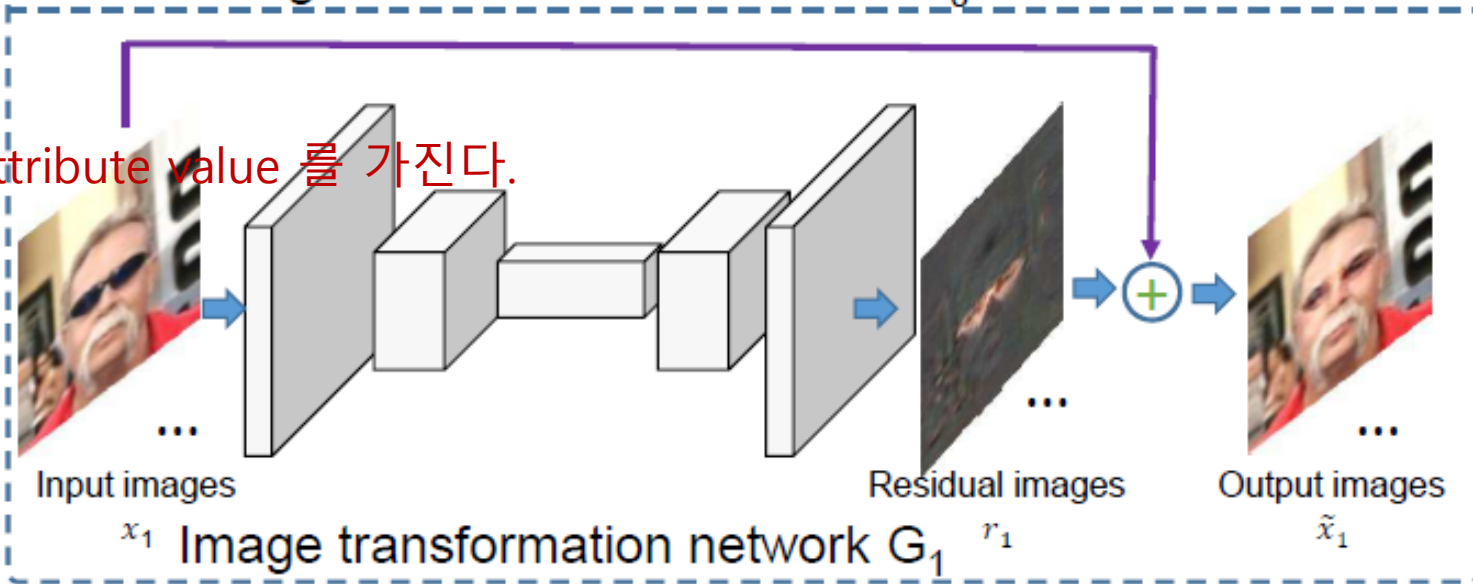
Learning the Residual Image

Image transformation networks G_0 와 G_1 은 manipulation 과 dual operation 에 사용된다.

Negative attribute value 를 가진다.



Positive attribute value 를 가진다.

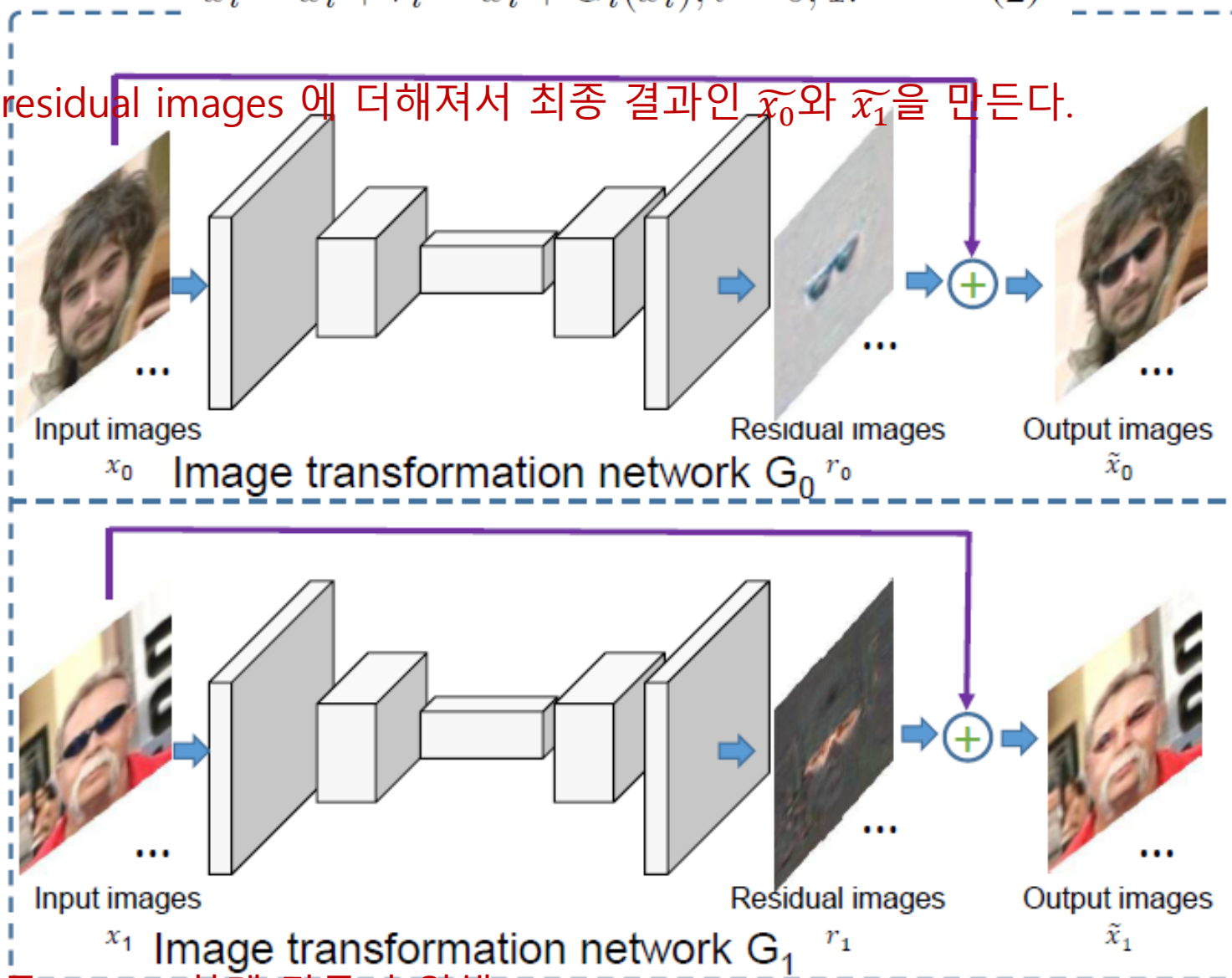


G_0 와 G_1 은 manipulation transformations 으로 residual images r_0 와 r_1 을 만든다.

Learning the Residual Image

$$\tilde{x}_i = x_i + r_i = x_i + G_i(x_i), i = 0, 1. \quad (2)$$

Input image 가 residual images 에 더해져서 최종 결과인 \tilde{x}_0 와 \tilde{x}_1 을 만든다.



Residual image 를 sparse 하게 만들기 위해
L-1 norm regularization 을 적용한다.

$$\ell_{pix}(r_i) = ||r_i||_1, i = 0, 1.$$

(3)

- Residual image 를 sparse 하게 만들기 위해 L1 norm regularization 을 적용했다. 이렇게 함으로써 Residual image 에서 관심 있는 곳은 값을 갖도록, 관심 없는 곳은 무시하도록 한다.
- Regularization 을 하는 이유는 알겠는데, 왜 L1 이 좋은지는 모르겠다. 다른 norm 을 써도 되지 않을까.

model.py

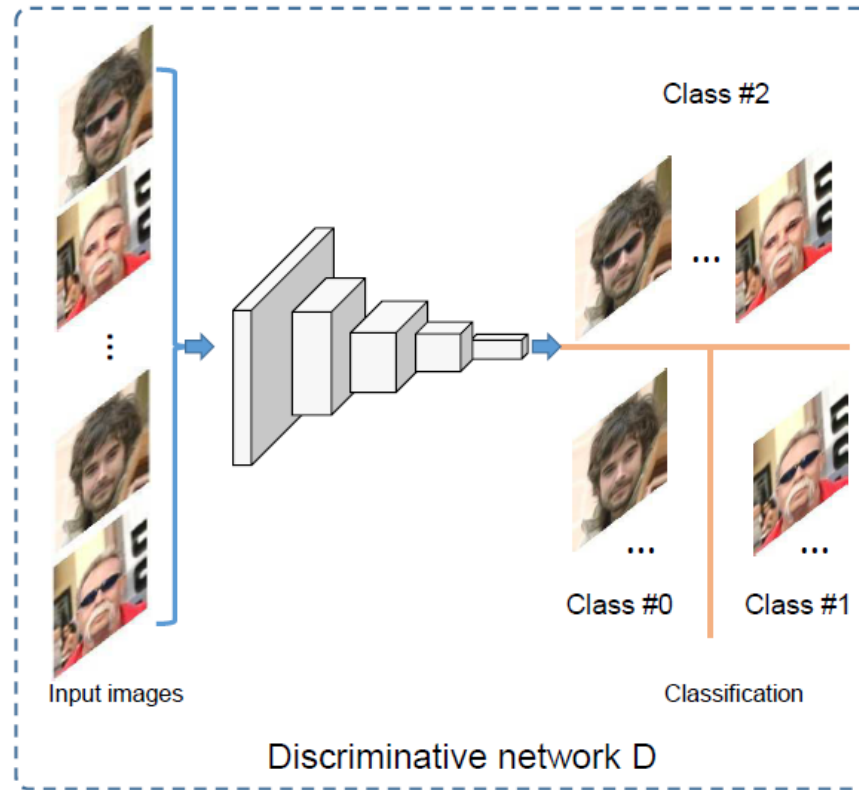
```
# Get instances to the network-classes
g_0 = Generator(seed, 'g_0')
g_1 = Generator(seed, 'g_1')
d = Discriminator(seed)

# Residual images/outputs from generators
r0 = g_0(self.X_neg)
r1 = g_1(self.X_pos)

# The altered image
x_theta_0 = r0 + self.X_neg
x_theta_1 = r1 + self.X_pos
```

3.3. The Discriminative Network

- Real images
 - x_0 : label 0
 - x_1 : label 1
- Transformed images
 - $\widetilde{x}_0, \widetilde{x}_1$: label 2



$$\ell_{cls}(t, p) = -\log(p_t), t = 0, 1, 2,$$

(4)

t : image 의 label

p_t : t-th label 의 softmax probability

model.py

```
# Discriminator output  
phi_fake_0, p_fake_0 = d(x_theta_0) # without glasses -> with glasses  
phi_fake_1, p_fake_1 = d(x_theta_1, True) # with glasses -> without glasses  
phi_real_0, p_real_0 = d(self.X_neg, True) # without glasses  
phi_real_1, p_real_1 = d(self.X_pos, True) # with glasses
```


- Perceptual loss

- 여러 이미지들 사이에서 content 차이를 측정하기 위해 사용된다.

[10][6][17]

- Input face image 와 유사한 content 를 가진 transformed image 를 만들기 위해 사용한다.

$$\ell_{per}(x, \tilde{x}) = ||\phi(x) - \phi(\tilde{x})||_1. \quad (5)$$

- $\phi(x)$: Discriminator 의 세 번째 layer 의 activation

model.py

```
# perceptual loss
self.l_per0 = tf.reduce_mean(tf.reduce_sum(tf.abs(phi_real_0 - phi_fake_0), axis=[1, 2, 3]))
self.l_per1 = tf.reduce_mean(tf.reduce_sum(tf.abs(phi_real_1 - phi_fake_1), axis=[1, 2, 3]))
```

phi_fake_0 : $\phi(\widetilde{x}_0)$ phi_fake_1 : $\phi(\widetilde{x}_1)$

phi_real_0 : $\phi(x_0)$ phi_real_1 : $\phi(x_1)$

- GAN loss
 - Discriminative network D 가 주어지면, image transformation networks G_0 와 G_1 에 대한 GAN loss 는 다음과 같다.

$$\ell_{GAN} = \begin{cases} -\log(D(G_i(x_i))) & i = 0, \\ -\log(1 - D(G_i(x_i))) & i = 1. \end{cases} \quad (6)$$

model.py

```
# gan loss
gan0_indices = ([1] * self.batchsize)
gan0_one_hot = tf.one_hot(gan0_indices, depth = 3)
self.l_gan0 = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits_v2(labels=gan0_one_hot, logits=p_fake_0))

gan1_indices = ([2] * self.batchsize)
gan1_one_hot = tf.one_hot(gan1_indices, depth = 3)
self.l_gan1 = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits_v2(labels=gan1_one_hot, logits=p_fake_1))
```

3.4. Dual Learning

- Machine translation 에서의 dual task
 - Source language \rightarrow Target language (Primal task)
 - Target language \rightarrow Source language (Dual task)
- Dual learning 의 방법은 two-player communication game 과 같다.
 1. First player 가 A 언어에서 B 언어로 메시지를 번역하고 second player 에게 보낸다.
 2. Second player 가 B 언어로 번역한 것이 자연스러운지 확인하고 first player 에게 알린다.
 3. Second player 는 A 언어로 메시지를 번역해서 다시 first player 에게 보낸다.
 4. First player 는 original message 와 일치하는지 확인하고 second player 에게 알린다.

- Dual learning process

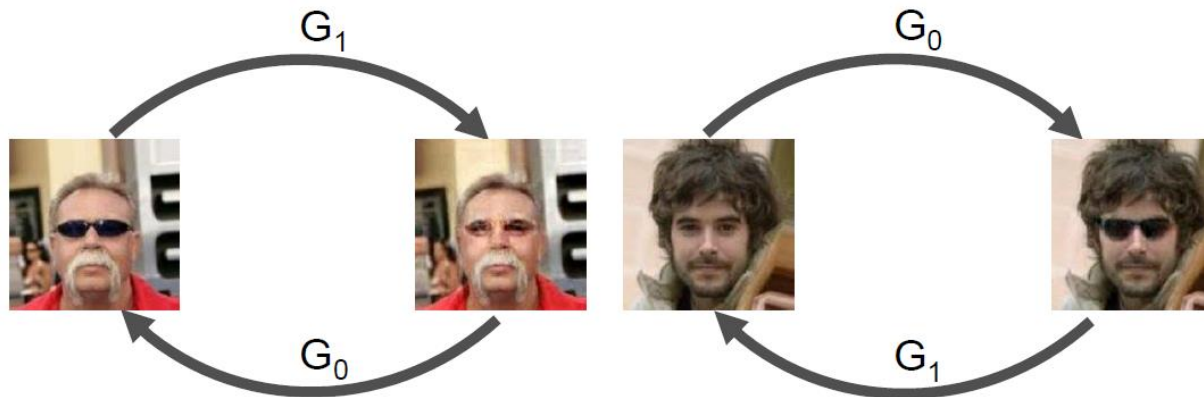


Figure 3: The dual learning process in this work.

- Negative attribute value 를 가진 image x_0 을 G_0 으로 보낸다.
- $\widetilde{x}_0 = G(x_0)$ 은 G_1 으로 들어간다.
- $\widehat{x}_0 = G_1(\widetilde{x}_0) = G_1(G_0(x_0))$
- \widehat{x}_0 은 x_0 와 같은 attribute value 를 찾는 것으로 기대된다.
- Transformation networks 를 위한 loss function $\ell_{dual}(\tilde{x}_i) = \begin{cases} -\log(1 - D(G_{1-i}(\tilde{x}_i))) & i = 0, \\ -\log(D(G_{1-i}(\tilde{x}_i))) & i = 1. \end{cases} \quad (7)$

model.py

```
# Dual output  
phi_dual_x0_theta, dual_x0_theta = d(g_1(x_theta_0, True) + x_theta_0, True)  
phi_dual_x1_theta, dual_x1_theta = d(g_0(x_theta_1, True) + x_theta_1, True)
```

model.py

```
#dual loss  
dual0_indices = ([2] * self.batchsize)  
dual0_one_hot = tf.one_hot(dual0_indices, depth = 3)  
self.l_dual0 = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits_v2(labels=dual0_one_hot, logits=dual_x0_theta))  
  
dual1_indices = ([1] * self.batchsize)  
dual1_one_hot = tf.one_hot(dual1_indices, depth = 3)  
self.l_dual1 = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits_v2(labels=dual1_one_hot, logits=dual_x1_theta))
```

3.5. Loss Function

- G_0/G_1 에 대한 loss function

$$\ell_G = \ell_{GAN} + \ell_{dual} + \alpha \ell_{pix} + \beta \ell_{per}, \quad (8)$$

- D 에 대한 loss function

$$\ell_D = \ell_{cls}. \quad (9)$$

model.py

```
# Loss_cls
#Discriminator loss
cls_indices = ([0]* (2 * self.batchsize)) + ([1] * self.batchsize) + ([2] * self.batchsize)
cls_one_hot = tf.one_hot(cls_indices, depth = 3)
cls_input = tf.concat([p_fake_0, p_fake_1, p_real_1, p_real_0], 0)
self.l_cls = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits_v2(labels=cls_one_hot, logits=cls_input))
```

model.py

```
# Loss_pix
self.l_pix0 = tf.reduce_mean(tf.norm(tf.reshape(r0, (self.batchsize, -1)), ord = 1, axis = 1))
self.l_pix1 = tf.reduce_mean(tf.norm(tf.reshape(r1, (self.batchsize, -1)), ord = 1, axis = 1))
```

4. Datasets

- CelebA
 - 200K개 이상의 유명인 이미지
 - 40개의 binary attributes
 - Attributes 들 중에서 6개를 골랐다.
 - *glass, mouth_open, smile, no_beard, young*
 - *male* 은 proposed method 를 평가하기 위해 사용했다.
 - CelebA dataset 이미지에서 중앙 부분을 128x128 크기로 crop 했다.
 - Dataset 에 매우 많은 image 들이 있지만, attribute labels 가 매우 편향되었다. 그러므로, test 에서 는 balance dataset 을 만들어서 사용한다.
- LFW
 - LFW dataset 은 generalization 을 testing 할 때만 사용한다.
- CelebA 에서 transformation networks 를 training 하기 위한 ground truth manipulated images 는 없다. (Unsupervised learning)

5. Implementation Details

- $G_0/G_1, D$ 구조의 세부 사항

Image transformation networks G_0/G_1	Discriminative network D
Input 128×128 color images	Input 128×128 color images
5×5 conv. 64 leaky RELU. stride 1. batchnorm	4×4 conv. 64 leaky RELU. stride 2. batchnorm
4×4 conv. 128 leaky RELU. stride 2. batchnorm	4×4 conv. 128 leaky RELU. stride 2. batchnorm
4×4 conv. 256 leaky RELU. stride 2. batchnorm	4×4 conv. 256 leaky RELU. stride 2. batchnorm
3×3 conv. 128 leaky RELU. stride 1. upsampling. batchnorm	4×4 conv. 512 leaky RELU. stride 2. batchnorm
3×3 conv. 64 leaky RELU. stride 1. upsampling. batchnorm	4×4 conv. 1024 leaky RELU. stride 2. batchnorm
4×4 conv. 3	4×4 conv. 1

Table 1: The network architectures of the image transformation networks G_0/G_1 and the discriminative network D

- Hyperparameters
 - $\beta = 0.1\alpha$
 - Local face attribute(*glasses, no_beard, mouth_open, smile*)
 - $\alpha = 5e - 4$
 - Global face attribute(*male, young*)
 - $\alpha = 1e - 6$
- Weights
 - Zero-centered Normal distribution 으로 초기화
 - Standard deviation=0.02
- Optimization
 - Adam optimizer 사용
- Learning rate
 - $2e - 4$ (both G and D)
- G_0 과 G_1 은 같은 때에 훈련된다.

model.py

```
alpha = 5e-5
beta = 0.1 * alpha

self.l_g0 = self.l_gan0 + self.l_dual0 + alpha * self.l_pix0 + beta * self.l_per0
self.l_g1 = self.l_gan1 + self.l_dual1 + alpha * self.l_pix1 + beta * self.l_per1

self.train_g0 = tf.train.AdamOptimizer(2e-4).minimize(self.l_g0, var_list=tf.get_collection(tf.GraphKeys.GLOBAL_VARIABLES,
"gan0"))
self.train_g1 = tf.train.AdamOptimizer(2e-4).minimize(self.l_g1, var_list=tf.get_collection(tf.GraphKeys.GLOBAL_VARIABLES,
"gan1"))
self.train_d = tf.train.AdamOptimizer(2e-4).minimize(self.l_cls, var_list=tf.get_collection(tf.GraphKeys.GLOBAL_VARIABLES,
"d_"))
```

6. Experiments

6.1. Local and Global Attribute Manipulation

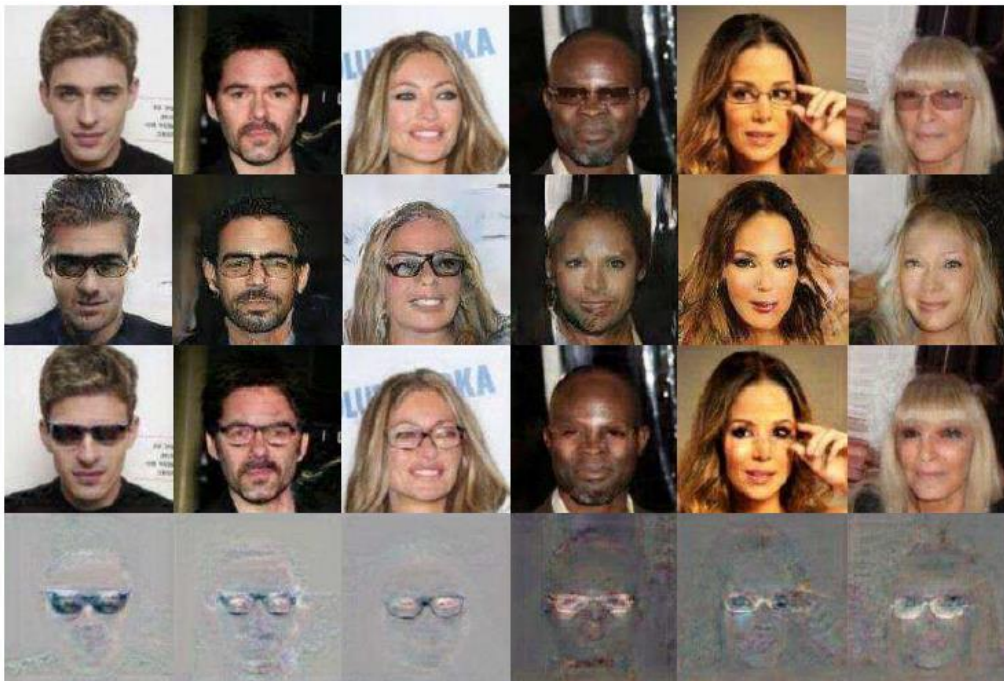
- Local attributes
 - *glasses, mouth-open, smile, no_beard*
- Global attributes
 - *male, young*

• Ours **vs** VAE-GAN[17] (CelebA)

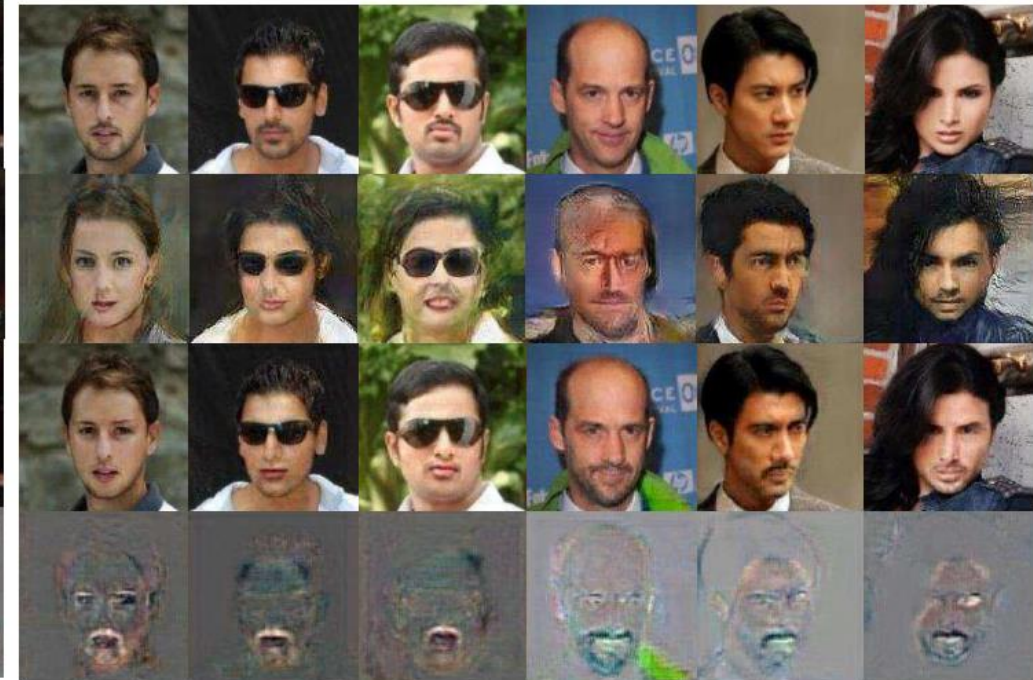
- First row : Original face images
- Second row : VAE-GAN model
- Third row : Ours
- Last row : Ours 로 학습된 Residual images
- 각 attribute manipulation 에서 Primal/Dual task 의 결과는 first/last three columns 에 있다.

Primal

Dual



(a) *Glasses*



(b) *No_beard*

- Ours **vs** VAE-GAN[17] (CelebA)

- First row : Original face images
- Second row : VAE-GAN model
- Third row : Ours
- Last row : Ours 로 학습된 Residual images
- 각 attribute manipulation 에서 Primal/Dual task 의 결과는 first/last three columns 에 있다.



(c) *Mouth-open*

(d) *Smile*

- Ours **vs** VAE-GAN[17] (CelebA)

- First row : Original face images
- Second row : VAE-GAN model
- Third row : Ours
- Last row : Ours 로 학습된 Residual images
- 각 attribute manipulation 에서 Primal/Dual task 의 결과는 first/last three columns 에 있다.



(e) *Male*

(f) *Young*

- Ours **vs** VAE-GAN[17] (LFW)

- First row : input images
- Second row : manipulated images
- 각 attribute manipulation 에서 Primal/Dual task 의 결과는 first/last three columns 에 있다.



(a) Glasses

(b) No_beard



(c) Mouth_open

(d) Smile



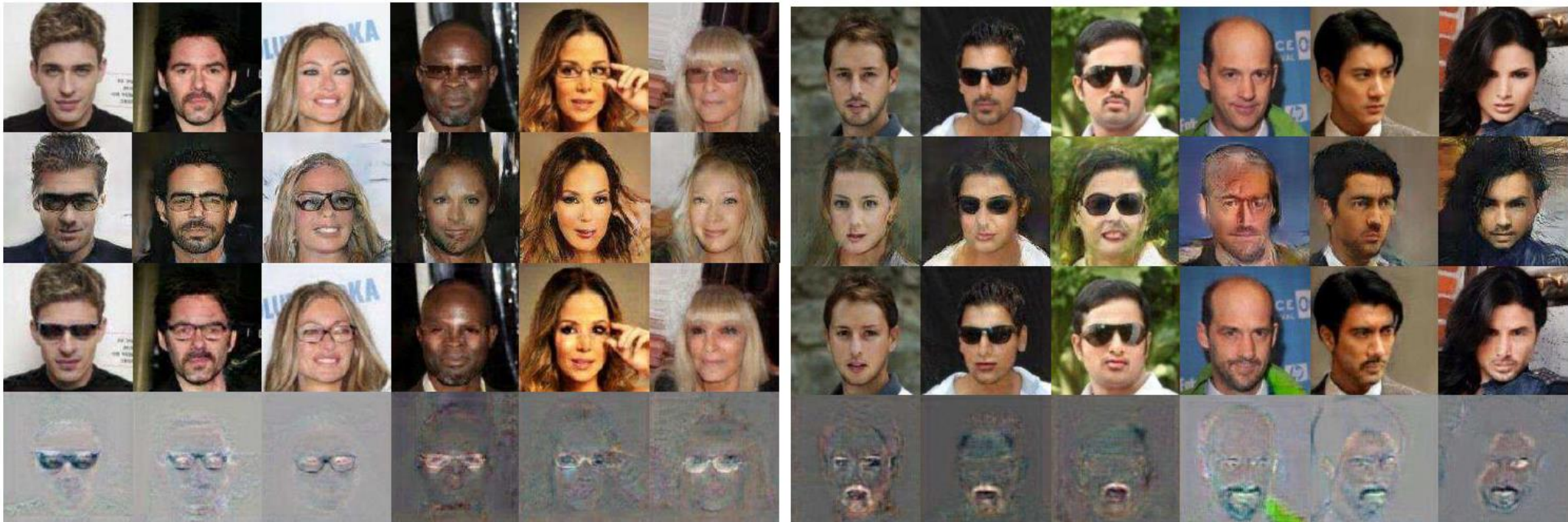
(e) Male

(f) Young

Experiments

VAE-GAN	Ours
많은 details 를 바꾼다.(hair style, skin color, background objects)	대부분의 details 을 보존한다.

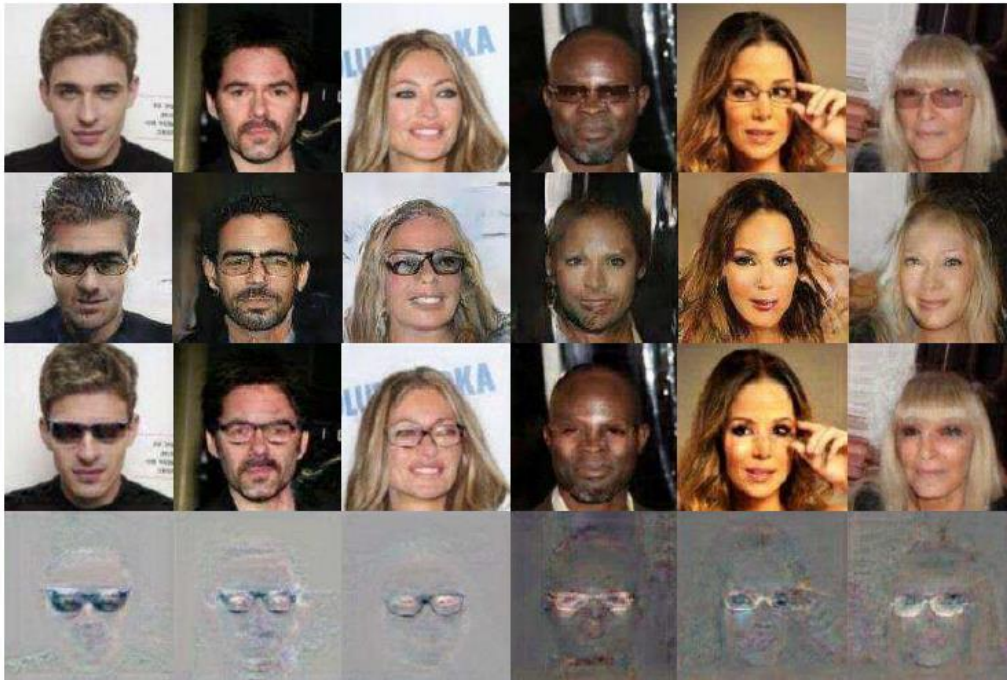
- First row (Original image) **vs** Third row (Transformed image)
- Manipulated counterparts 에서 Target attributes 에 해당하는 부분은 제외하고 original images 의 details 가 대부분 남아있는 것을 볼 수 있다.



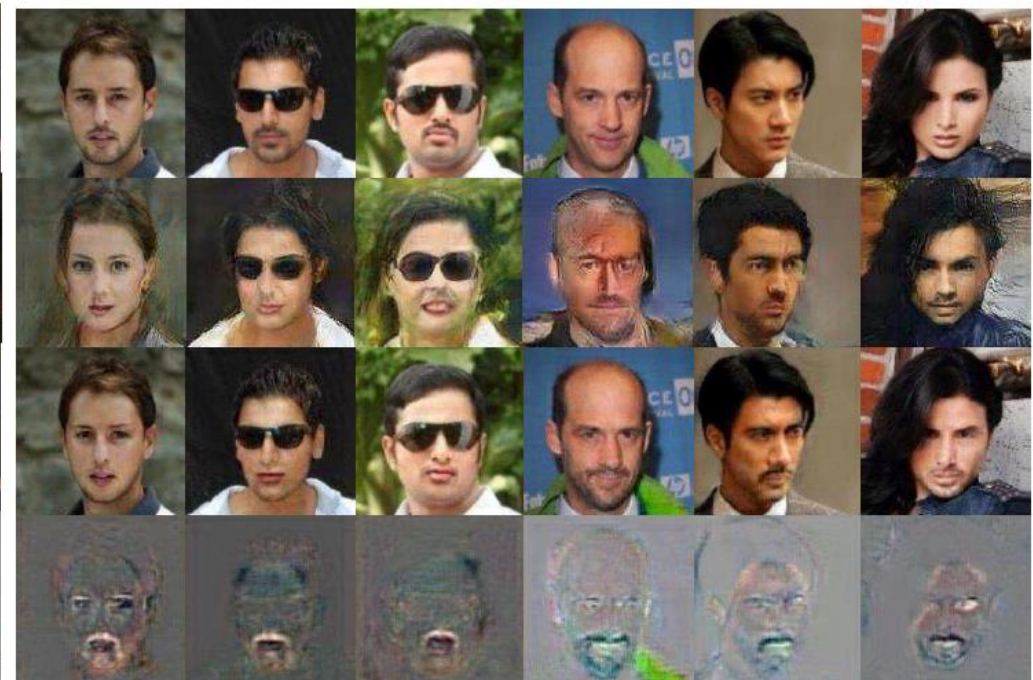
(a) *Glasses*

(b) *No_beard*

- Last row (residual images)
- Local attribute manipulation 에서, Residual images 에서는 주로 local areas 강하게 반응한다.
- Example
 - Sun glasses 추가
 - residual image 에서 black sun glasses 부분이 가장 강한 반응을 보인다.
 - Glasses 제거
 - residual image 가 eyes 를 enhance 하도록 하고 original face image 에서 보여지는 glasses 의 hint 가 되는 부분은 제거한다.



(a) Glasses



(b) No_beard

- Local face attribute manipulations 는 간단하고 알아차리기 쉽다.
- *mouth_open* 과 *smile* 모두 턱의 "이동"을 발생시킨다.
- Image transformation networks 는 *mouth_open* manipulation 에서 턱을 아래로 내리고 *mouth_close* manipulation 에서 턱을 위로 올린다.



(c) *Mouth-open*

(d) *Smile*

Experiments

- Global attributes manipulations (male, young) 은 가장 어려운 task 이다.
- Networks 는 wrinkles, hair color, beard 등과 같은 미묘한 차이를 학습해야 한다.
- (e) Male
 - Residual image 에서는 male 과 female 의 주요한 차이 : beard, the color of lips and eyes
 - Residual images 에서 이 두가지의 manipulations 에 대해 강하게 반응하는 곳은 local attribute 와는 달리 이미지 전체에 흩어져 있다.
- (f) Young
 - Primal task (young \rightarrow old) : 주름을 만든다.
 - Dual task (old \rightarrow young) : hair color 를 어둡게 만든다.



(e) Male

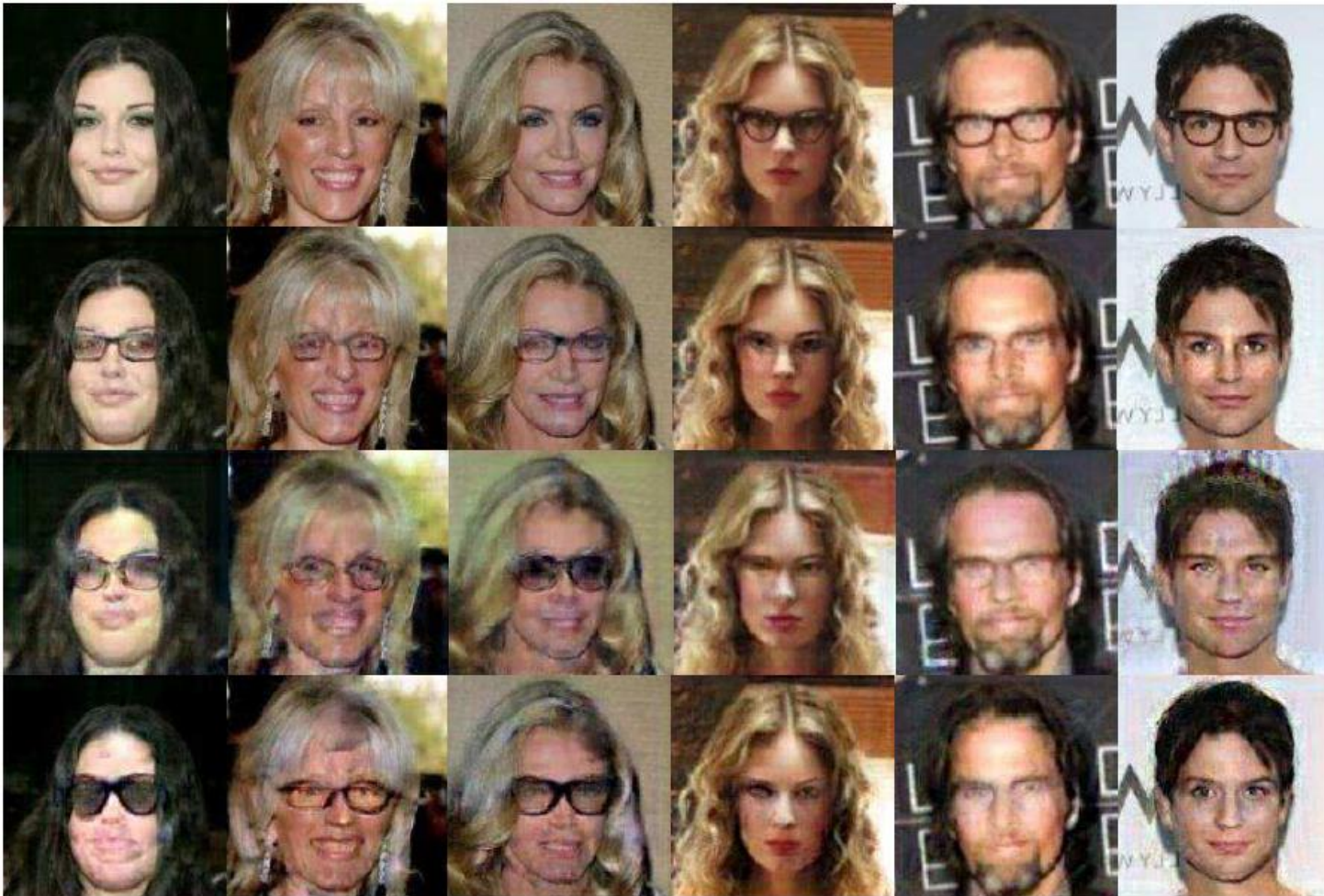
(f) Young

6.2. Ablation Study

- 우리 model 은 두 개의 중심이 되는 components 로 구성된다.
 - Residual image learning
 - Dual learning
- 이번 section 에서, 이것들의 효과를 확인할 것이다.

Experiments

- Validation of residual image learning and dual learning in *glasses* manipulation
 - First row: Original input images
 - Second row: Ours
 - Third row: Ours \ominus Residual image learning
 - Last row: Ours \ominus Dual learning



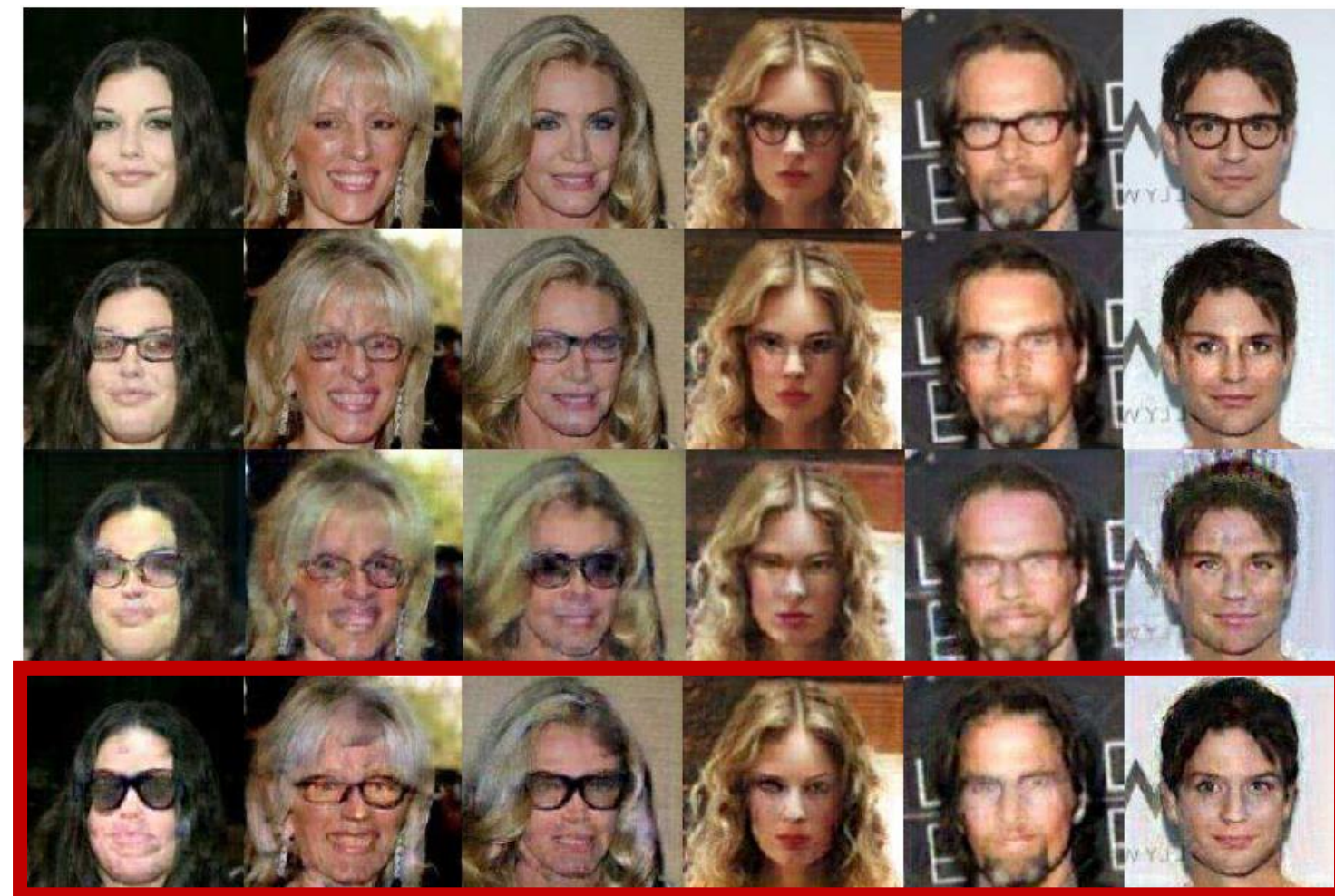
Experiments

- Validation of residual image learning and dual learning in *glasses* manipulation
 - First row: Original input images
 - Second row: Ours
 - Third row: Ours \ominus Residual image learning
 - Last row: Ours \ominus Dual learning



- Residual image learning 이 없는 경우
 - Noise 가 나타나고, 수염도 잘못 붙여졌다.
 - model 이 더 낮은 quality 의 image 를 만든다

- Validation of residual image learning and dual learning in *glasses* manipulation
 - First row: Original input images
 - Second row: Ours
 - Third row: Ours \ominus Residual image learning
 - Last row: Ours \ominus Dual learning



- Dual learning 이 없는 경우
 - Hair color 에 변화가 일어났다.
-
- Dual learning 의 효과
 - 1) 두 개의 generators 에서 만들어진 images 는 training sample 의 수를 증가시킨다.
 - 2) Dual learning 의 단계 동안에, $G_1(G_0(x_0))$ 과 $G_0(G_1(x_1))$ 에 대한 ground truth images 가 보여지고, 그것은 두 개의 generator 들이 training 하는 것을 쉽게 한다.

- 그러므로, Residual image learning 과 Dual learning 을 합치는 것이 더 나은 manipulation 결과를 낼 것이다.

6.3. Visual Feature Decorrelation

- Classifier 를 end-to-end 방식으로 훈련하면 dataset 이 매우 편향되었을 때 classifier 는 target visual features 를 정확히 알아차리지 못한다.
- CelebA dataset 의 분석을 통해, 우리는 *male* 과 *no_beard* 가 매우 큰 상관관계를 가지는 것을 알아냈다. (피어슨 상관계수 = -0.5222)
- 이번 section 에서, 제안된 방법이 상관관계에 얽매이지 않는 것을 설명한다.

Visual feature decorrelation in the manipulation of attribute *no_beard*

First row : original input images

Second row : VAE-GAN 의 reconstructed images

Third row : VAE-GAN 의 manipulated images

Last row : Ours' manipulated images



- VAE-GAN reconstruction results 는 VAE-GAN model 이 original images 를 잘 학습하는 것을 보여준다.
- VAE-GAN 의 manipulated images 에서 hair length 는 original image 보다 상당히 짧다. 이것은 대부분의 남성 얼굴이 short hair 를 가지기 때문이다. (*male* 과 *no_beard* 의 상관계수는 매우 크다.)
- 그러나, 제안된 방법은 성공적으로 이 두개의 features 의 상관관계를 떨어뜨렸다. Transformed images 에서 hair 는 original images 의 hair 와 거의 동일하다.
- 이 흥미로운 결과는 method 가 attribute-specific area 에 집중하게 도와주는 residual image learning 과 dual learning 덕분이다.

6.4. Landmark Detection with Glasses Removal

- Face landmark detection accuracy 에서 얻은 성능으로 glasses removal 의 효과를 양적으로 나타낸다.
- Landmark detection algorithm
 - Dlib[12] 에 의해 실행되는 ERT 방식[11]
 - Detection model 을 300-W dataset 에서 훈련시켰다.
- 세가지 test sets 을 사용한다.
 - 1) $D1$: wearing glasses 이미지
 - 2) $D0$: without glasses 이미지
 - 3) $D1_m$: $D1$ 과 동일한 이미지에서 Ours 를 사용해 안경을 제거한 이미지(without glasses 이미지)
- $D0$ 와 $D1$ 은 *glasses* manipulation 실험에 있는 같은 test sets 이다.

• Landmark Detection 결과

	without glasses	wearing glasses	without glasses
Landmark	$D0$	$D1$	$D1_m$
Eye landmarks	0.02341	0.03570	0.03048
Rest landmarks	0.04424	0.04605	0.04608

- $D1, D0, D1_m$ 에서 landmark detection algorithm 의 Average normalized distance error
- Eye landmarks: 좌우 눈의 landmarks
- Rest landmarks: 코, 그리고 입의 좌우 귀통이의 landmarks

• Landmark Detection 결과

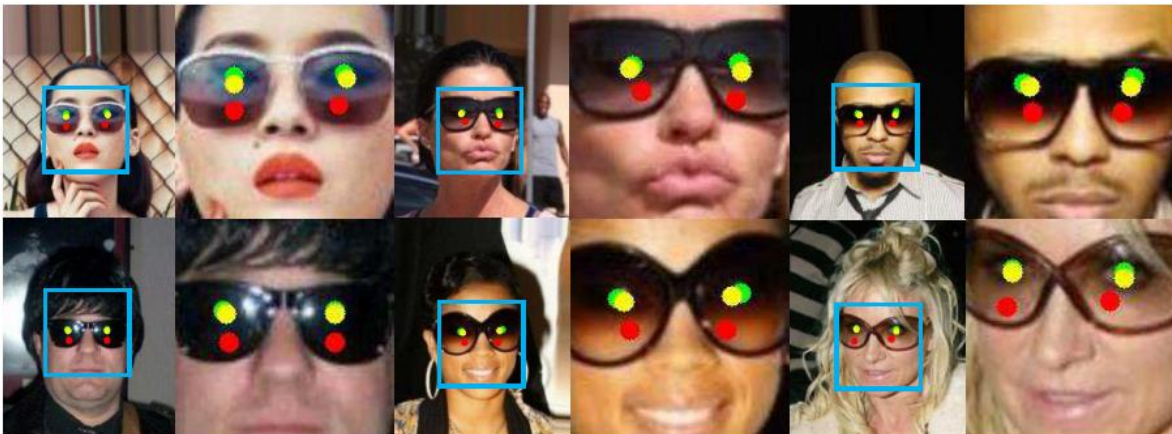
	without glasses	wearing glasses	without glasses
Landmark	$D0$	$D1$	$D1_m$
Eye landmarks	0.02341	0.03570	0.03048
Rest landmarks	0.04424	0.04605	0.04608

- $D1, D0, D1_m$ 에서 landmark detection algorithm 의 Average normalized distance error
- Eye landmarks: 좌우 눈의 landmarks
- Rest landmarks: 코, 그리고 입의 좌우 귀통이의 landmarks
- $D0$ 과 $D1$ 에서 identities 가 달라도, wearing glasses 가 landmark detection 에 영향을 미치는 것을 찾을 수 있다.
- 첫 번째 열과 두 번째 열의 eye landmark detections 의 결과를 비교해보며 detection error 가 증가한 것을 찾는다.
- 그러나, $D1_m$ 에서의 error 는 $D1$ 에서의 error 보다 매우 작은 것을 볼 수 있다.
- Tab 2 의 두 번째 행에서의 결과와 비교하면, $D1$ 과 $D1_m$ 에서의 error 가 거의 같은 것을 관찰할 수 있고 그것은 face 의 나머지 부분이 거의 변하지 않았다는 것을 나타낸다.

• Landmark Detection 결과

	without glasses	wearing glasses	without glasses
Landmark	$D0$	$D1$	$D1_m$
Eye landmarks	0.02341	0.03570	0.03048
Rest landmarks	0.04424	0.04605	0.04608

- $D1, D0, D1_m$ 에서 landmark detection algorithm 의 Average normalized distance error
- Eye landmarks: 좌우 눈의 landmarks
- Rest landmarks: 코, 그리고 입의 좌우 귀통이의 landmarks



- Glasses removal 에서 landmark detection 성능 측정
- Green points: $D0$
- Red points: $D1$
- Yellow points: $D1_m$

7. Discussion

- Hyperparameter α 와 β 를 설정했다.
 - Global attribute manipulation: $\alpha = 5e - 6$
 - Local attribute manipulation: $\alpha = 5e - 4$
- 하나의 discriminator 만을 사용한 것은 computation 효율을 향상시킨다.
- Test set 에서 identity overlaps 에 의해 발생한 약간의 information leak 이 있을 수 있다.

- Large poses 때문에 glasses removal 에서 실패한 경우
 - First row: Original input images
 - Second row: Manipulated images
 - Third row: Residual images



8. Conclusion

1. Face attribute manipulation 을 다루기 위해 GAN based method 를 제안한다.
2. Residual image learning 과 Dual learning 을 채택했다. 이 전략들은 image transformation networks 가 attribute-specific area 에 집중하도록 하고 서로를 학습하도록 한다.
3. 실험은 제안된 방법이 성공적으로 face images 를 manipulate 한 것을 보여주고 attribute 와 관련 없는 areas 에서의 대부분의 details 는 변하지 않게 한다.