

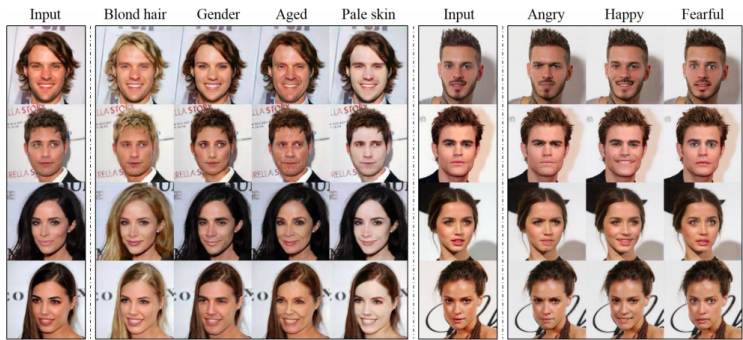
# 리뷰 - StarGAN

Created by 임 도형, last modified yesterday at 10:51 AM

## 개요

원제 : StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation

1개의 네트워크를 가지고 얼굴의 복수의 속성을 동시 변환하였다.



이를 위해 CycleGAN, InfoGAN을 사용.

복수의 다른 데이터를 사용한 것이 특이.

## 고찰

논문에서 말하는 multi domain 간 변환은 맞다. 하지만 엄밀히는 아니다.

흔히 말하는 domain 변환



그보다는 단지 얼굴의 복수 속성 변환을 한개의 네트워크로 가능할 뿐이다.

각 속성이 도메인은 아니다.

하지만 이러한 방법이 다른 방법을 고안하기 위한 직관이나 된다는 믿음 기반을 준다.

복수의 데이터 셋을 동시에 사용한 기법은 새로운 방법 보다는 트릭 정도의 수준.

논문 자체를 잘 써서, 그림을 잘 그려서 돋보인 논문.

# StarGAN 장점

- n개의 도메인간 변환을 위해 복수개의 G 필요 없이 1개의 G로 가능.
- 복수의 데이터를 동시에 사용

## 방법

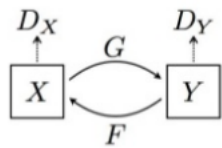
StarGAN = CycleGAN + InfoGAN

복수의 데이터를 사용하기 위해 mask사용.

## CycleGAN

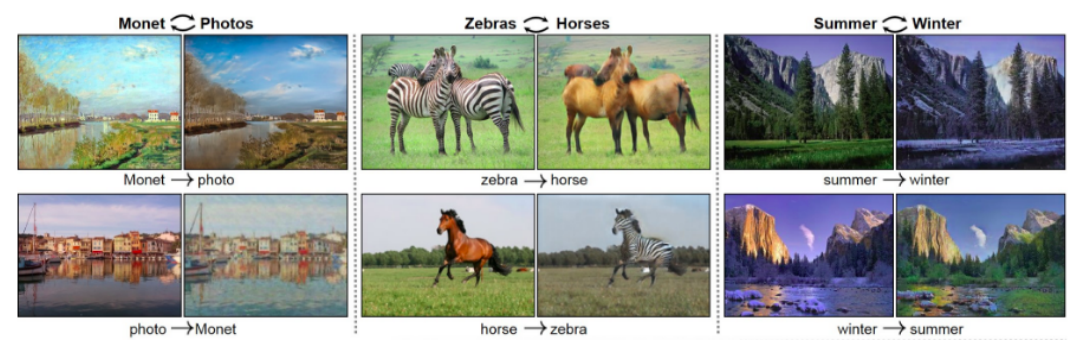
G로 생성한 결과를 다시 G의 입력으로 해서 다시 생성한다.

그리고 그 결과를 원본과 비교한다.



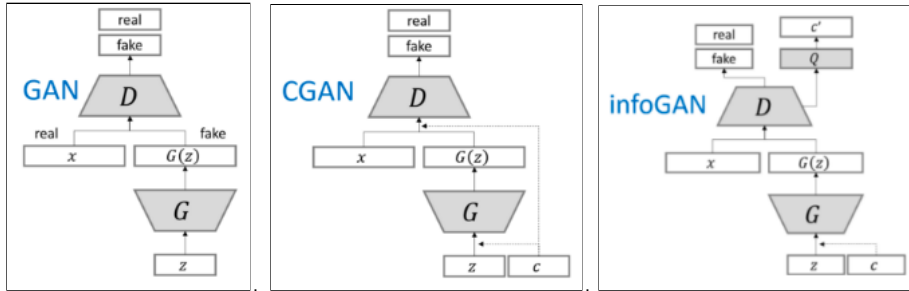
본질은 유지한 채 특성만을 변환하는 예들이 있다.

아래의 예에서 본질(horse)은 유지한 채 특성(무늬)만을 변경하였다.



본 논문에서는 본질(인물)은 유지한 채 특성(머리색갈, 나이, 성별)을 변경하기 위해 차용했다.

## InfoGAN



MNIST의 예를 든다면,  
 GAN은 어떤 번호의 이미지는 하여간에 필기체 숫자처럼 보이는 것만 G가 생성하면 된다. 10개의 번호가 골고루 나올 필요도 없다. 그냥 D만 구별 못하면 된다.  
 CGAN은 condition을 G와 D에 모두 준다. 그래서 지금 보이는 이미지가 숫자 뭐라고 D에 알려주기 때문에 G는 무조건 해당 숫자의 이미지를 생성해야 한다.  
 InfoGAN과 CGAN은 거의 유사. G에게는 똑같고, 단지 D가 그 정보를 classifier로 사용되도록 사용하는 것 외에. G에게는 적용결과가 동일하다.

### mask

CelebA 데이터를 사용할 경우 RaFD label이 없음을, 그리고 반대로  
 RaFD 데이터를 사용할 경우 CelebA label이 없음을 mask로 사용하였다.

그래서 데이터 모양은 다음과 같다.  
 (1,0,1,0,0)+(0,0,0,0,0)+(1,0) : CelebA의 경우 RaFD의 label은 전부 0. 그리고 mask는 (1,0)  
 (0,0,0,0,0)+(1,0,1,0,1)+(0,1) : RaFD의 경우 celebA의 label은 전부 0. 그리고 mask는 (0,1)

□ 그냥 class 여부를 사용하는 0/1이 아닌 -1과 같이 N/A를 의미하는 값을 주었어도 되었을 것 같다.

## 사용 데이터

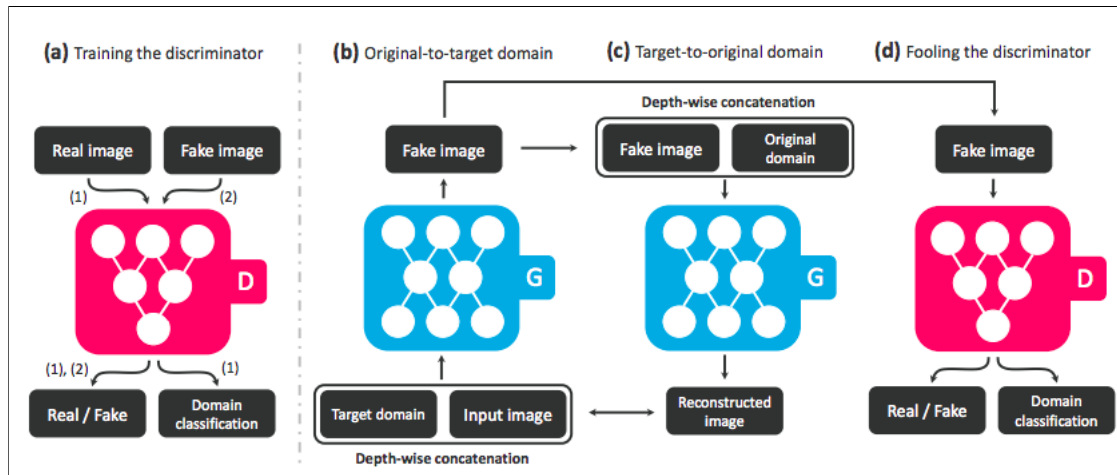
얼굴 이미지 데이터 2가지를 사용함

- CelebA
- RaFB

각 데이터가 제공하는 속성이 다름

- CelebA는 머리 색, 성별, 나이의 속성
- RaFB는 감정에 대한 것. 행복, 화남, 슬픔 등

## 방법 사례 설명



## G 입력/출력

입력 : 레이블(CeleA의, RaFB의, 도메인) + 이미지

출력 : 이미지

## D 입력/출력

입력 : 이미지

출력 : Real-Fake 여부 + CeleA 레이블 + RaFB 레이블

## CelebA 데이터의 경우

CelebA의 레이블은 있고, RaFB의 레이블은 없다.

그래서 입력이 다음과 같이 구성됨

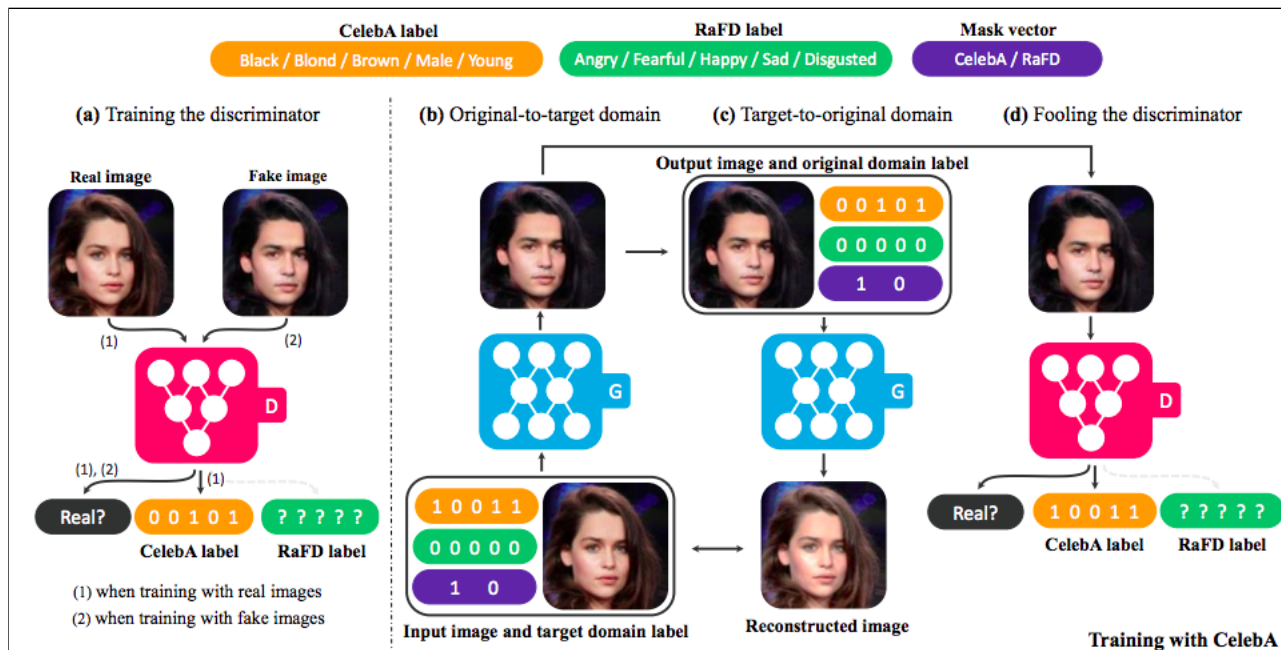
CelebA Label + RaFB Label + mask

= ( is black, is blond, is brown, is male, is young ) + ( is angry, is fearfu, is happy, is sad, is disgusted ) + ( is CelebA, is RaFB )

= ( 0, 0, 1, 0, 1 ) + ( x, x, x, x, x ) + ( 1, 0 )



사용된 label의 값은 0은 생성될 이미지의 것이다.



## RaFD 데이터의 경우

CeleA의 레이블은 없고, RaFB의 레이블은 있다.

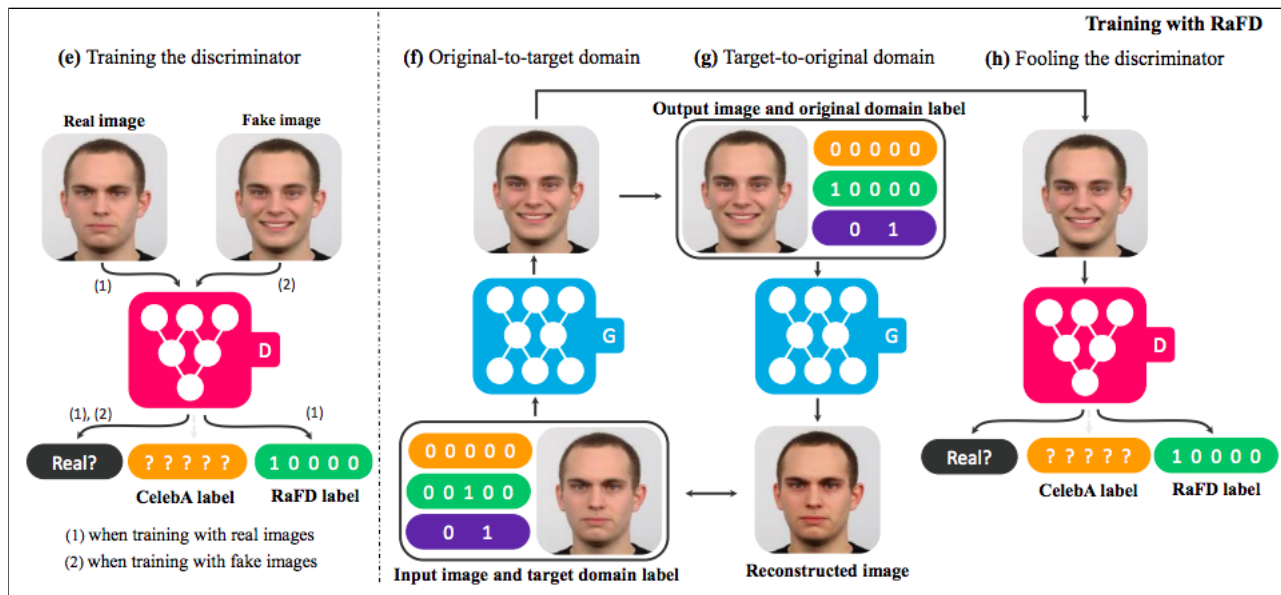
그래서 입력이 다음과 같이 구성됨

CelebA Label + RaFB Label + mask

= ( is black, is blond, is brown, is male, is young ) + ( is angry, is fearfu, is happy, is sad, is disgusted ) + ( is CelebA, is RaFB )

= ( x, x, x, x, x ) + ( 0, 0, 1, 0, 0 ) + ( 0, 1 )





## 코스트

### D

일반적인 GAN과 달리 Loss-cls 항목이 하나 더 있다.

$$\mathcal{L}_D = -\mathcal{L}_{adv} + \lambda_{cls} \mathcal{L}_{cls}^r$$

Loss-adv는 일반적인 GAN 자체의 코스트 이고,

$$\mathcal{L}_{adv} = \mathbb{E}_x [\log D_{src}(x)] + \mathbb{E}_{x,c} [\log (1 - D_{src}(G(x, c)))]$$

Loss-cls는 InfoGAN의 분류(classification) 코스트

$$\mathcal{L}_{cls}^r = \mathbb{E}_{x,c'} [-\log D_{cls}(c'|x)]$$

where  $D(x)$  : probability distribution over  $x$

### G

일반적인 GAN의 항목외에 2개가 더 있다.

$$\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{cls} \mathcal{L}_{cls}^f + \lambda_{rec} \mathcal{L}_{rec}$$

Loss-clс는 InfoGAN의 분류(classification) 코스트 : target label로 속성이 변하도록 한다.

$$\mathcal{L}_{cls}^f = \mathbb{E}_{x,c}[-\log D_{cls}(c|G(x,c))]$$

Loss-rec는 CycleGAN의 원복(reconstruction) 코스트 : 속성만 변하고 본질은 유지하게 한다.

$$\mathcal{L}_{rec} = \mathbb{E}_{x,c,c'}[||x - G(G(x,c),c')||_1]$$

Network Architecture

별 다를 것 없이 일반적이다.

Part	Input → Output Shape	Layer Information
Down-sampling	$(h, w, 3 + n_c) \rightarrow (h, w, 64)$	CONV-(N64, K7x7, S1, P3), IN, ReLU
	$(h, w, 64) \rightarrow (\frac{h}{2}, \frac{w}{2}, 128)$	CONV-(N128, K4x4, S2, P1), IN, ReLU
	$(\frac{h}{2}, \frac{w}{2}, 128) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	CONV-(N256, K4x4, S2, P1), IN, ReLU
Bottleneck	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
Up-sampling	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{2}, \frac{w}{2}, 128)$	DECONV-(N128, K4x4, S2, P1), IN, ReLU
	$(\frac{h}{2}, \frac{w}{2}, 128) \rightarrow (h, w, 64)$	DECONV-(N64, K4x4, S2, P1), IN, ReLU
	$(h, w, 64) \rightarrow (h, w, 3)$	CONV-(N3, K7x7, S1, P3), Tanh

Table 4. Generator network architecture

Layer	Input → Output Shape	Layer Information
Input Layer	$(h, w, 3) \rightarrow (\frac{h}{2}, \frac{w}{2}, 64)$	CONV-(N64, K4x4, S2, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{2}, \frac{w}{2}, 64) \rightarrow (\frac{h}{4}, \frac{w}{4}, 128)$	CONV-(N128, K4x4, S2, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{4}, \frac{w}{4}, 128) \rightarrow (\frac{h}{8}, \frac{w}{8}, 256)$	CONV-(N256, K4x4, S2, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{8}, \frac{w}{8}, 256) \rightarrow (\frac{h}{16}, \frac{w}{16}, 512)$	CONV-(N512, K4x4, S2, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{16}, \frac{w}{16}, 512) \rightarrow (\frac{h}{32}, \frac{w}{32}, 1024)$	CONV-(N1024, K4x4, S2, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{32}, \frac{w}{32}, 1024) \rightarrow (\frac{h}{64}, \frac{w}{64}, 2048)$	CONV-(N2048, K4x4, S2, P1), Leaky ReLU
Output Layer ( $D_{src}$ )	$(\frac{h}{64}, \frac{w}{64}, 2048) \rightarrow (\frac{h}{64}, \frac{w}{64}, 1)$	CONV-(N1, K3x3, S1, P1)
Output Layer ( $D_{cls}$ )	$(\frac{h}{64}, \frac{w}{64}, 2048) \rightarrow (1, 1, n_d)$	CONV-(N( $n_d$ ), K $\frac{h}{64}$ x $\frac{w}{64}$ , S1, P0)

Table 5. Discriminator network architecture



## 실험 결과



Figure 9. Single and multiple attribute transfer on CelebA (Input, Black hair, Blond hair, Brown hair, Gender, Aged, Hair color + Gender, Hair color + Aged, Gender + Aged, Hair color + Gender + Aged).

맨 우측이 3개의 속성(머리 색깔, 성별, 나이들)을 전부 변환한 예.



# Reference

- paper : <https://arxiv.org/pdf/1711.09020.pdf>
- DIAT : Deep Identity-Aware Transfer of Facital Attribute : <https://arxiv.org/pdf/1610.05586.pdf>
- CycleGAN implementation : <https://github.com/junyanz/CycleGAN>
- infoGAN : <https://arxiv.org/abs/1606.03657>
- 다양한 GAN 설명 : <http://nooverfit.com/wp/%E7%8B%AC%E5%AE%B6%EF%BD%9Cgan%E5%A4%A7%E7%9B%98%E7%82%B9%EF%BC%8C%E8%81%8A%E8%81%8A%E8%BF%99%E4%BA%9B%E5%B9%B4%E7%9A%84%E7%94%9F%E6%88%90%E5%AF%B9%lsgan-wgan-cgan-info/>

No labels

Powered by a free **Atlassian Confluence Open Source Project License** granted to Flamingo. Evaluate Confluence today.  
This Confluence installation runs a Free Gliffy License - Evaluate the Gliffy Confluence Plugin for your Wiki!  
This Confluence installation runs a Free Gliffy License - Evaluate the Gliffy Confluence Plugin for your Wiki!