* 객체 = 인스턴스 ex) 사진속의 사람, 사람이 입고 입는 치마, 기린,, 등 변환할 타겟을 뜻
* 비지도(unsupervised) 번역= 짝을 이루지 않은(unpaired) 번역 = 입력값과 결과값을 동시에 학습시키지 않고 입력값만 넣어 결과를 만들어 내겠다는 뜻

<요약>

기존 gan에서 이미지 안에 인스턴스가 여러 개가 있고, 인스턴스의 모양 자체가 변하는 경우 (ex)바지를 입은 사람을 치마를 입은 사람으로) 제대로 변환하지 못함

이 문제를 해결하기 위해 insta gan은

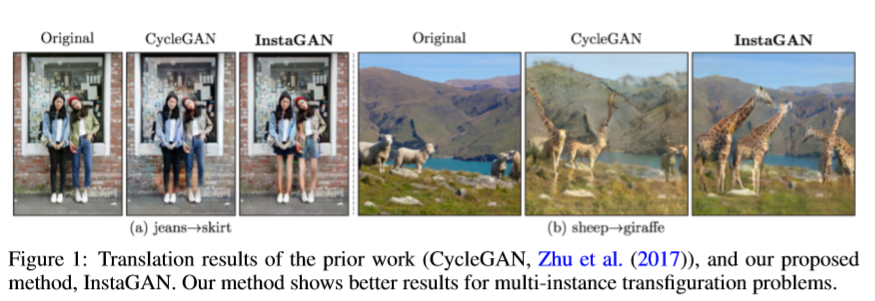
1. 각 인스턴스의 순열 불변 속성(permutation invariance property)을 유지하면서 이미지와 그에 해당하는 인스턴스 속성 집합을 변환 🡪 이를 위해 네트워크가 타겟이 되는 인스턴스의 밖에서 identity function을 학습하도록 하는 context preserving loss 를 소개
2. 제한된 GPU 메모리로 여러 인스턴스를 처리하는 sequential mini-batch inference/training technique 소개

<1. 소개>

이전 방법은 특히 번역 작업에 인스턴스 모양이 크게 변경되거나 (Zhu etal., 2017) 번역 할 이미지에 여러 대상 인스턴스 (Gokaslanetal)가 포함 된 경우 까다로운 작업에서 실패함

우리의 목표는 이미지에서 이미지로의 번역을 이러한 까다로운 작업으로 확장하는 것. 예를 들어, 패션 이미지에서 바지를 치마로 바꾸어 고객이 어느 쪽이 더 나은지 결정할 수 있도록 다음 단계까지 적용 성을 강화할 수 있음. 이를 위해, 본 발명자들은 GAN (generative adversarial network)의 프레임 워크에 다수의 타겟 객체의 인스턴스 정보를 통합하는 새로운 방법을 제안. 따라서 인스턴스 인식 GAN (instance aware gan = InstaGAN)이라고 했음.

이 작업에서는 인스턴스 정보를 위해 객체 분할 마스크(object segmentation masks)를 사용합니다. 예를 들어 색상과 같은 다른 세부 사항은 무시하고 객체의 경계를 표시하여 모양을 잘 표현할 수 있습니다. 이 정보를 사용하여, 우리의 방법은 그림 1과 같이 다중 인스턴스 변환 작업에 대한 인상적인 결과를 보여줍니다.



논문에서 밝힌 세가지 주요 기여

1. instance-augmented neural architecture

: 이미지와 해당 인스턴스 속성 세트를 모두 변환하는 신경망 아키텍처를 제안합니다. 우리 아키텍처는 입력에 의해 조절되는 임의의 수의 인스턴스 속성을 번역 할 수 있으며 인스턴스 순서에 따라 변형이 가능하도록 설계되었습니다.

1. a context preserving loss

: 네트워크가 번역에서 대상 인스턴스에 초점을 맞추고 외부에서 identity function을 배우도록 장려하는 컨텍스트 보존 손실(context preserving loss)을 제안합니다. 즉, 대상 인스턴스를 변환하면서 백그라운드 컨텍스트를 유지하는 것을 목표로 합니다.

1. a sequential mini-batch inference/training technique

: 전체 세트를 한 번에 수행하는 대신 인스턴스 속성의 미니 배치를 순차적으로 변환하는 방법을 제안합니다. 제한된 GPU 메모리를 사용하여 많은 수의 인스턴스 속성을 처리 할 수 ​​있으므로 많은 인스턴스가 있는 이미지의 네트워크를 보다 일반화 할 수 있습니다. 또한, 여러 중간 샘플을 생성하여 훈련 중 데이터 확대 기능을 수행하기 때문에 몇 가지 경우로 이미지의 번역 품질을 향상시킵니다

이 논문에서 제안하는 방법은 이전에 제안된 방법들과 다르게 배경을 유지하면서 변환하고 싶은 인스턴스만 효율적으로 변환할 수 있다고 자랑함.

Unlike the previous results in a simple setting, our focus is on the harmony of instances naturally rendered with the background. On the other hand, CycleGAN (Zhu et al., 2017) show some results for multi-instance cases, but report only a limited performance for transﬁguration tasks.

<2. INSTAGAN: INSTANCE-AWARE IMAGE-TO-IMAGE TRANSLATION >

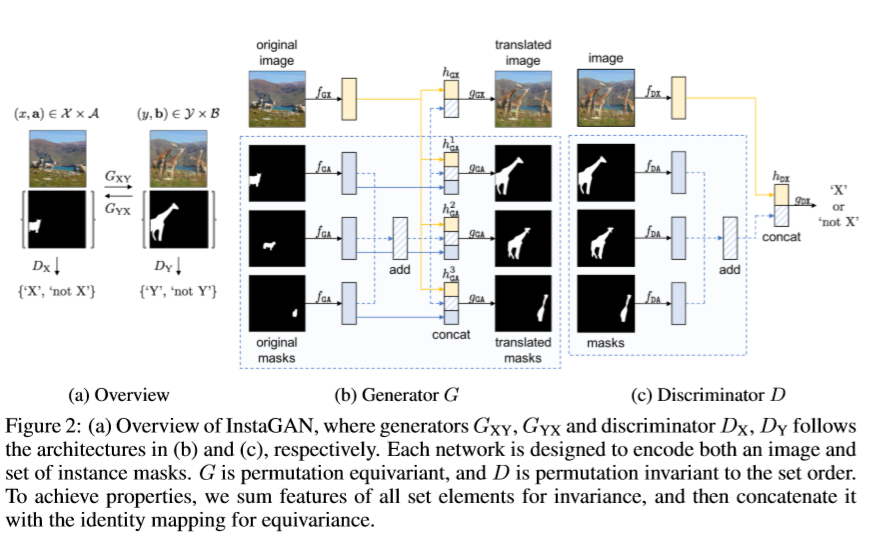
두개의 이미지 도메인 X , Y 가 주어졌을 때, 이미지에서 이미지로 변환하는 문제는 Gxy : X->Y , Gyx : Y->X 와 같은 서로 다른 이미지 도메인 간의 매핑을 배우는 것. (조건부 p (y | x) 또는 / 및 p (x | y)를 추정하는 조건부 생성 모델링 작업으로 공식화 할수 있음) 우리가 다루는 비지도 번역의 목표는 두 이미지 도메인의 pdata (x) 및 pdata (y) , 즉 원본 데이터의 marginal distribution에서 짝을 이루지 않은 샘플만 사용하여 이러한 매핑을 복구하는 것입니다

우리의 접근 방식의 주된 독창적 인 아이디어는 추가 인스턴스 정보를 통합하는 것입니다. 즉, 인스턴스 속성 A 세트의 공간을 원본 이미지 공간 X로 확장하여 이미지 간 이미지 변환을 향상시키는 것. A에 속한 인스턴스 속성 a는 N개의 타겟 인스턴의 모든 개별 속성으로 구성됨 : a = {ai}Ni=1

이 작업에서는 instance segmentation mask만 사용하지만 속성에 유용한 모든 유형의 인스턴스 정보를 통합할 수 있습니다. 우리의 접근법은 속성이 증대된 공간(attribute-augmented spaces) X × A와 Y × B 사이의 공동 매핑 학습으로 설명될 수 있다. 이로 인해 이미지에서 서로 다른 인스턴스가 분리되어 생성기가 정확하고 상세한 변환을 수행할 수 있습니다.

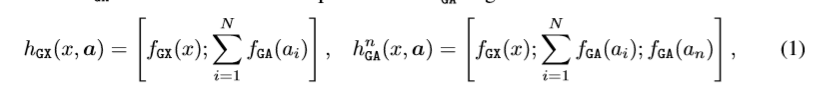
<2.1 INSTAGAN ARCHITECTURE>

Insta gan 구축하기 위해 cycle gan접근법 사용, 입력 이미지만 넣어도 됨

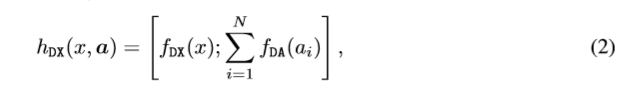
그림 2는 모델의 전체 아키텍처

우리는 두개의 generator를 학습함 : Gxy : X × Y 🡪 Y × B 와 Gyx = Y × B 🡪 X × A Gxy는 원래 데이터 (x,a)를 타겟 도메인 데이터 (y’, b’)으로 변환함 (Gyx 도 반대로 변환)

적대적 판별기(Adversarial discrinator) : Dx : X×A 🡪 {‘X’, ‘not X’} and Dy : Y×B→{‘Y’, ‘not Y’}, : Dx 는 데이터(원래의 (x,a) 또는 변환된 (x’, a’))가 타겟 도메인 X×A가 맞는지 아닌지 결정함 (Dy 도 반대로 똑같이)

생성기 G는 x와 a를 모두 인코딩하여 y0과 b0으로 변환합니다. 특히, 세트 a에서의 인스턴스 속성의 순서는 변환된 이미지 y0에 영향을 미치지 않아야 하고, 세트 a에서의 각각의 인스턴스 속성은 b0에서의 대응하는 속성으로 변환되어야 한다. 다시 말해, y0은 a의 인스턴스에 대해 순열-불변이고, b0는 그것들에 대한 순열-변형이다.  
이러한 속성은 피쳐 인코딩에 적절한 연산자를 도입하여 구현할 수 있음. 이미지 피쳐 추출기(image feature extractor) fGX 및 속성 피쳐 추출기(attribute feature extractor) fGA를 사용하여 이미지 및 속성에서 개별 피쳐를 추출합니다. fGA를 사용하여 개별적으로 추출 된 속성 피쳐는 summation :N i=1 fGA(ai)를 통해 순열-불변량 세트 피쳐(permutation-invariant set feature)로 집계됩니다. 그림 2b에 설명 된대로 일부 이미지 및 속성 피쳐를 set feature와 연결하여 이미지 및 속성 생성기에 공급합니다. 공식적으로, 생성기 G에서의 이미지 표현 hGX 및 n 번째 속성 표현 hnGA 는 다음과 같이 공식화 될 수있다 :

hnGA 인코딩에서 모든 속성 피쳐는 contextual feature로 진행한다 ? . 마지막으로 hGX 는 이미지 생성기 gGX 로 들어가고, hnGA ( n = 1,…,N)는 속성생성기 gGX로 들어간다

한편 판별기 D는 x 와 a 모두를 인코딩하고 그 쌍이 도메인에서 왔는지 여부를 판단. 이때 집합 a의 인스턴스 속성 순서는 출력에 영향을 미치지 않아야 함. 이와 유사하게 인스턴스에 순열이 없는 판별기 D의 표현은 다음과 같이 공식화됩니다.

이것은 적대적 판별기 gDX에 공급된다.

우리는 네트워크가 x와 a의 관계를 배울 수 있기 때문에 각 신경 구성 요소에 대한 이미지 x와 인스턴스 속성 a의 공동 인코딩(joinit encoding)이 중요하다는 것을 강조합니다. 예를 들어, x와 a에 대해 두 개의 개별 인코딩 및 판별기가 사용되는 경우, 서로 일치하지 않는 이미지 및 인스턴스 마스크를 생성하도록 생성기가 잘못 판단할 수 있음. 조인트 인코딩 및 판별기를 사용하여 생성기는 분할 마스크와 일치하는 영역에 올바른 인스턴스 이미지를 생성할 수 있음.

섹션 3에서 볼 수 있듯이, 우리의 접근법은 원래 레이아웃을 고려하여 출력 인스턴스를 분 할 수 있습니다. fGX, fGA, fDX, fDA, gGX, gGA 및 gDX와 같은 임의의 유형의 신경망이 상기 언급된 서브 네트워크 아키텍처에 사용될 수 있음에 유의한다. 우리는 부록 A의 실험에 사용된 세부 아키텍처를 설명합니다.

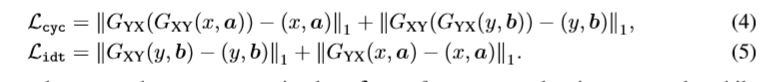
<2.2 TRAINING LOSS >

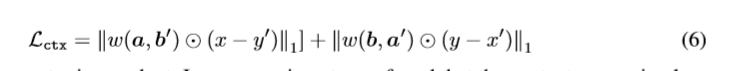
이미지 간 변환 모델의 목표: 원래 컨텍스트를 유지하면서 (예 : 배경 또는 인스턴스의 방향과 같은 도메인 독립적 특성) 도메인을 번역하는 것. 🡪 두가지 loss 모두 고려

도메인 손실(domain loss): 생성된 출력이 대상 도메인의 스타일을 따르도록 함 컨텐츠 손실(content loss): 출력을 원래 내용으로 유지

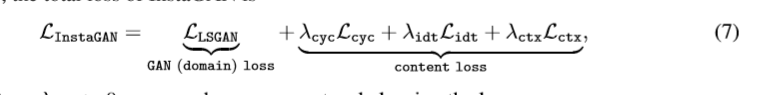
기본모델인 cycle gan에 따라 domain loss에는 GAN loss를 사용하고,content loss에는 cycle-consistency loss와 identity mapping loss 둘다 사용, // 새로운 content loss인 context preserving loss도 제안함. Loss function 을 싱글 트레이닝 샘플 (x,a) ∈X×A and (y,b) ∈Y×B 의 함수로 나타냄

GAN 손실은 원래 Goodfellow가 generative modeling을 위해 제안한것으로, 생성기 G와 판별기 D를 교대로 훈련함. 여기서 D는 G에 의해 생성 된 가짜 / 생성 / 번역 된 데이터 중 실제 데이터인지를 결정합니다. 우수한 성능인 LSGAN방식을 사용

cycle-consistency loss인 Lcyc와 identity mapping loss인 Lidt 는 샘플이 각 두번, 한번씩 변환된 후에도 원본 내용을 잘 유지하도록 해줌

마지막으로 새로 제안된 context preserving loss 인 Lctx는 인스턴스 외의 것(ex) 배경)은 그대로 두고 인스턴스만 변환하도록 함. pixel-wise weighted l1-loss 인데, 배경은 가중치가 1이고 인스턴스는 0. 그런데 두 도메인의 배경은 transfiguration-type 변환이후 달라지므로 , 원래 이미지와 변환된 이미지 둘 다에서 배경에 있는 픽셀에만 0이 아닌 가중치를 적용함. 즉, 원래 샘플 (x, a), (y, b) 및 변환 된 샘플 (y0, b0), (x0, a0)의 경우 가중치 w (a, b0), w (b, a0) 이진 표현 인스턴스 마스크의 요소 별 최소값에서 1을 뺀 값입니다.

동그라미 두개는 element-wise product임

따라서 insta gan의 최종 loss는

Where lλcyc,λidt,λctx > 0 are some hyper-parameters balancing the losses.

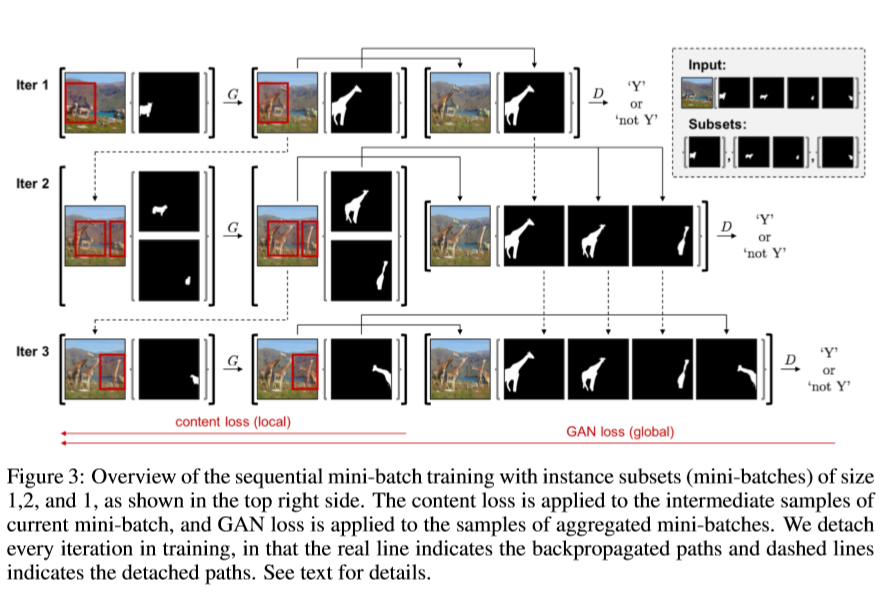
<2.3 SEQUENTIAL MINI-BATCH TRANSLATION>

필요한 GPU 메모리는 인스턴스 수에 따라 선형 적으로 증가. 실험에서 머신은 훈련 중에 적은 수의 인스턴스 속성 (예 : 2) 만 전달할 수 있었으므로 학습 된 모델은 일반화가 열악하여 인스턴스 수가 많은 이미지로 전달되었습니다. 이 문제를 해결하기 위해 GPU 메모리를 늘리지 않고 임의의 수의 인스턴스를 학습 할 수있는 새로운 추론 / 훈련 기술을 제안합니다. 우선 하위 계층을 순차적으로 번역하는 순차 추론 기법(the sequential inference scheme)을 설명하고 그에 해당하는 미니 배치 훈련 기술을 설명.

주어진 입력 (x, a) 에서 인스턴스 마스크 a를 mini-batch a1, …, am으로 나눔(a =Si ai and ai∩aj = ∅ for i != j. ) m = 1,,,M 그다음 m번쨰 반복에서 이미지 마스크 쌍 (xm, am)을 변환. 여기서 xm은 이전 반복에서 변환 된 이미지 y’m-1이고 x1 = x. 이 sequential schema에서, 각각의 반복에서 생성기 (G)는 중간 반복 이미지 (y’m)를 출력하는데, 이는 현재의 반복까지 모든 미니 배치 변환 및 누적된 인스턴스 마스크 (b’m)의 미니 배치를 축적한다:

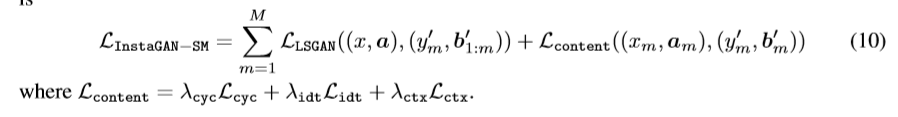
변환된 이미지를 인스턴스 마스크의 미니 배치와 정렬하기 위해 변환된 모든 미니 배치를 집계하고 변환된 샘플을 생성:

이 순차추론기법의 최종 output은 

두번쨰로 이에 상응하는 순차훈련 알고리즘(sequential training algorithm) = 그림3

(오른쪽 상단에 표시된대로 크기가 1, 2 및 1 인 인스턴스 하위 세트 (미니 배치)를 사용한 순차적 인 미니 배치 교육 개요. 콘텐츠 손실은 현재 미니 배치의 중간 샘플에 적용되고 GAN 손실은 집계 된 미니 배치 샘플에 적용됩니다. 실제 선은 역 전파 경로를 나타내고 점선은 분리 된 경로를 나타 내기 때문에 훈련에서 모든 반복을 분리합니다)

입력 및 출력의 함수일 뿐이므로 현재 미니 배치 am의 중간 샘플 (y’m, b’m)에 내용 손실 (4-6)을 적용합니다. 대조적으로, 인스턴스 마스크의 부분 서브 세트만을 사용할 때 네트워크가 이미지 및 마스크를 정렬하지 못하기 때문에 집계된 미니 배치 샘플 (y’m, b’1:m)에 GAN 손실 (3)을 적용합니다. . 우리는 완전한 마스크 마스크만으로 실제 / 원본 샘플 {x}을 사용했습니다.

따라서 Insta Gan의 training loss의 식은:

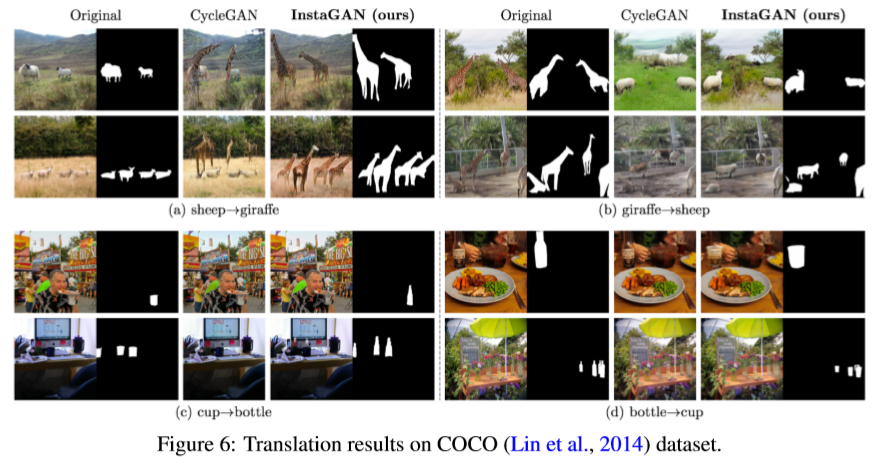
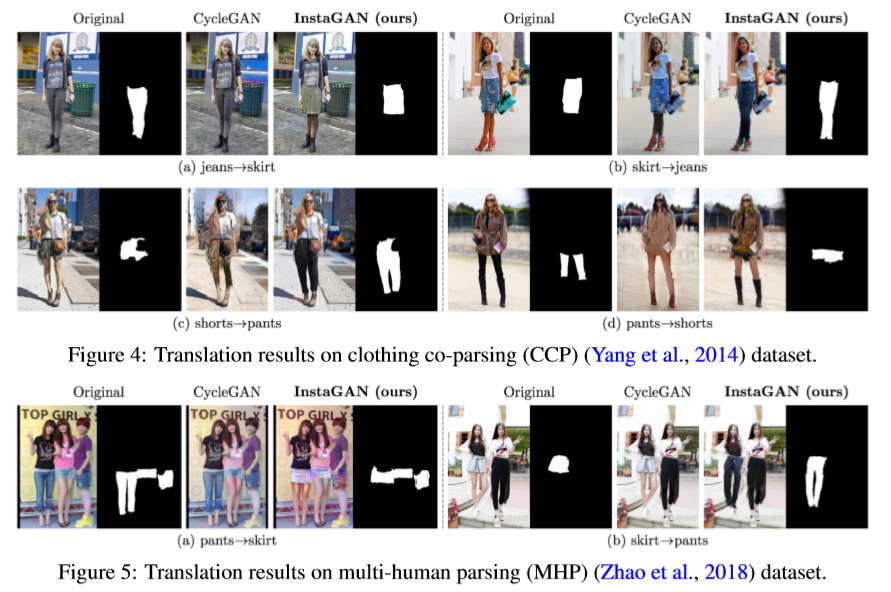
우리는 모든 m번째 훈련을 분리하기 떄문에 training instance의 수에 상관없이 고정된 크기의 GPU 메모리만 필요함.

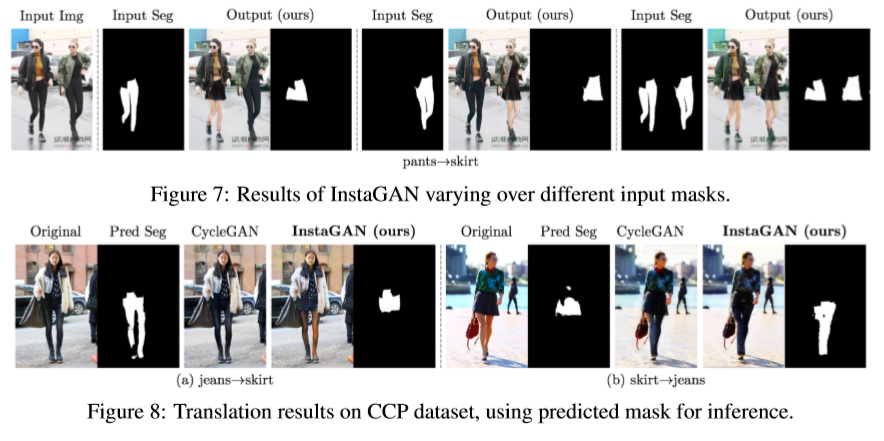
< 3 EXPERIMENTAL RESULTS >

< 3.1 IMAGE-TO-IMAGE TRANSLATION RESULTS >

Cyclegan 과의 비교. 공정한 비교를 위해 InstaGAN은 이미지와 마스크에 각각 두 개의 네트워크를 사용하므로 CycleGAN의 매개 변수 수를 두 배로 늘 렸습니다. 우리는 다양한 데이터 세트에서 두 가지 클래스를 샘플링하여 두 도메인으로 사용합니다. 번역합니다. 시각화에서는 압축을 위해 모든 인스턴스 마스크를 하나로 병합합니다. 실험에 대한 자세한 설정은 부록 B를 참조하십시오. 3 개의 데이터 세트에 대한 변환 결과가 각각 그림 4, 5 및 6에 나와 있습니다. CycleGAN은 대부분 실패하지만, 우리의 방법은 대상 인스턴스의 합리적인 형태를 생성하고 컨텍스트 보존 손실을 통해 인스턴스에 초점을 맞춤으로써 원래 컨텍스트를 유지합니다. 예를 들어, 그림 6의 양 ↔ 기린에 대한 결과를 참조하십시오. CycleGAN은 종종 양과 유사한 인스턴스를 생성하지만 원래 배경을 잃습니다. InstaGAN은 더 좋은 양 또는 기린을 생성 할뿐만 아니라 원래 인스턴스의 레이아웃을 유지합니다. 즉 양과 기린의 방향 (왼쪽, 오른쪽, 정면)은 번역 후 일관됩니다. 더 많은 실험 결과는 부록 E에 나와 있습니다. 코드 및 결과는 https://github.com/sangwoomo/instagan에서 확인할 수 있습니다.

* Cycle gan의 한계 : 인스턴스 모양 바꾸기 힘듬, 원래 배경 유지 불가

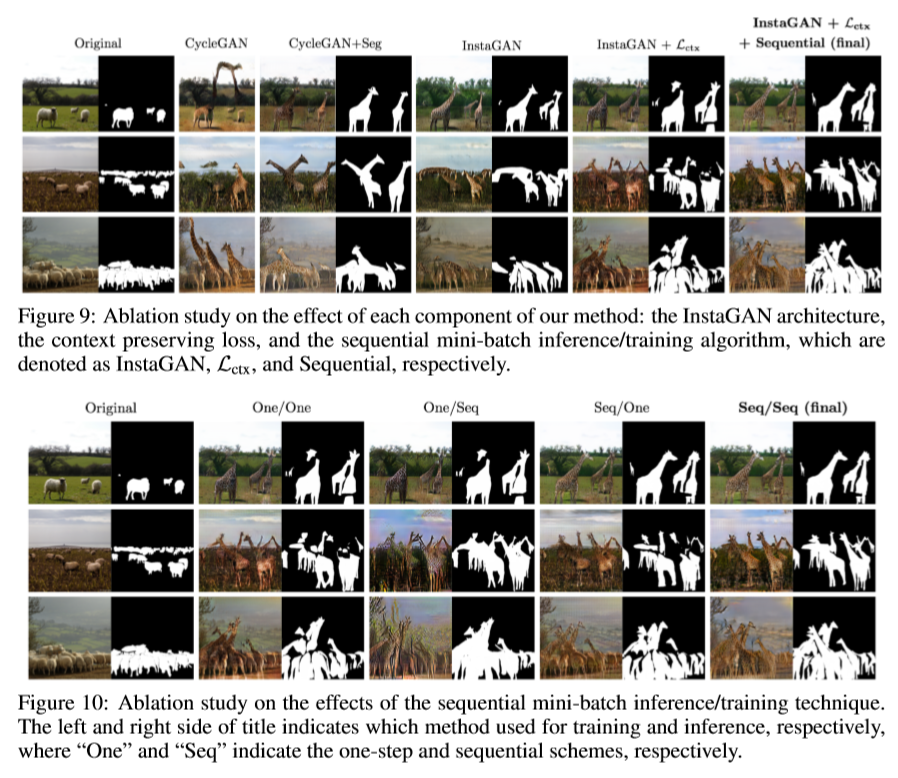
 우리의 방법은 그림 7과 같이 입력을 조정하여 변환 할 인스턴스를 제어 할 수 있습니다. 이러한 제어는 CycleGAN에서는 불가능합니다 부록 D의 표 1 및 표 2는 각각 CCP 및 COCO 데이터 세트의 분류 점수를 보여줍니다. 당사의 방법은 모든 분류 실험에서 CycleGAN을 능가합니다. 예를 들어, 바지 → 반바지 작업의 정확도는 23.2 %이며 CycleGAN은 8.5 %에 불과합니다. 



<3.2 ABLATION STUDY >

우리의 방법은 InstaGANarchitecture, 컨텍스트 보존 Lctx 및 순차적 미니 배치 추론 / 훈련 기술로 구성됩니다.

1. 인스턴스 마스크 세트를 변환하는 아키텍처와 달리 CycleGAN + Seg는 모든 마스크의 통합을 한 번에 변환합니다. 이로 인해 CycleGAN + Seg는 일부 인스턴스를 변환하지 못하고 종종 병합합니다. 다른 한편으로, 우리의 아키텍처는 모든 인스턴스를 유지하고 더 잘 풀어줍니다. = 우리게 다중 인스턴스 더 잘 변환한다
2. Context preserving loss의 영향을 연구합니다 : 배경을 더 잘 보존 할뿐만 아니라 (행 2), 매핑을 정규화 할 때 번역 결과를 향상시킵니다 (행 3). = 우리거 배경 보존된다
3. 순차 번역의 효과를 연구합니다. 일반화 성능 (행 2,3)을 향상시킬뿐만 아니라 데이터 확대 (1 행)를 통해 소수의 인스턴스에서 번역 결과를 향상
4. 그림 10은“Seq”로 표시된 순차 번역이“One”으로 표시된 1 단계 접근 방식에 비해 추론 및 훈련에 얼마나 효과적인지보고합니다. 1 단계 교육의 경우 머신에 적합한 최대 수이므로 2 개의 인스턴스 만 고려합니다. 다른 한편으로, 순차 훈련의 경우, 우리는 2 개의 인스턴스, 즉 인스턴스의 이미지를 순차적으로 훈련한다. 1 단계 추론의 경우 전체 세트를 한 번에 번역하고 순차적 추론의 경우 각 반복마다 두 개의 인스턴스를 순차적으로 변환합니다. 순차 알고리즘은 훈련과 추론에 모두 효과적이라는 것을 발견했다. 많은 인스턴스에 대한 1 단계 추론이 제한된 GPU 메모리로 인해 훈련에 표시되지 않기 때문에 여러 인스턴스에 대한 일반화 성능이 나쁩니다.



<4 CONCLUSION>

다중 인스턴스 변환 잘된다