1.도입

기존의 문제점들.

기존의 것들은 그렇게 좋은 성능을 보여주지 못한다.
특히 selfie2anime 같은 도메인이 확실히 다른 pair 같은 경우에
그러므로 전처리과정이 데이터분포의 복잡성을 제한시킴으로써 피해졌다
거기다가 DRIT 같은 것은 모양을 보존하는 것과
모양을 바꾸는 것에 대해 원하는 결과를 얻지 못한다.

고정된 네트워크 구조와 하이퍼 파라미터때문에
이런 네트워크 구조나 하이퍼파라미터 설정은
특정 데이터셋에 맞춰 조정되어야 한다.

이 작업에서 우리는 novel(참신한) method를 추구한다.
이것은 비지도 im2im 변환에 새 attention module과 새 normalization function을
end-to-end 방식으로 incorporate(통합)한다.
이전의 attention 기반의 작업은 변환된 오브젝트에 배경을 맞춰야 했기 때문에 shape를 바꾸지 못했다.

이 모델은 translation을 도와준다.
더 중요한 영역에 집중하고 덜 중요한 영역을 덜 집중한다.
auxiliary classifier 기반의 어텐션 맵에 기반해 소스와 타겟 도메인을 무시함으로써,
shape transformation을 facilitating(촉진한다)

real image와 target domain의 fake image의 차이에
집중함으로써 generator의 attention map이
두 region을 구분하는 areas에 집중하도록 induce(유도하다) 하는것에 대해
판별자 내의 어텐션 맵이 파인튜닝을 돕는다.

어텐셔널 매커니즘과 관련해
모양과 질감의 다른 변화량을 가진 다양한 데이터셋에서
normalization function의 선택이
변환결과의 질에 엄청난 효과가 있다

batch-instance normalization(BIN)에 영향을 받아,
Adaptive Layer-Instance Normalization(AdaLIN)
적응적으로 Instance Normalization과 Layer Normalization 사이의 적절한 비율을 셀렉팅함으로써 AdaLin의 파라미터는 데이터셋으로부터 학습한다

AdaLIN function은 attention-guided모델이 유연하게
질감과 모양의 변화량을 컨트롤하도록 돕는다.

그 결과로 구조나 하이퍼파라미터의 변화 없이
모델은 holistic(전체적인) 변화나 큰 모양의 변화 없이도
이미지변환을 수행할 수 있다.

제안된 작업의 주요 기여(contribution) (대충 요약되어있는 느낌)
We propose a novel method for unsupervised imageto-image translation with a new attention module and
a new normalization function, AdaLIN.
• Our attention module helps the model to know where
to transform intensively by distinguishing between
source and target domains based on the attention map
obtained by the auxiliary classifier.
• AdaLIN function helps our attention-guided model to
flexibly control the amount of change in shape and texture without modifying the model architecture or the
hyper-parameters.

2.관련 work
2.1.GAN
우리 모델은 타겟 도메인이 소스도메인과 완전히 달라도
unpair 이어도 된다.
2.2 Img to Img
pix2pix/cycle-GAN/UNIT/MUNIT-등
기존 모델들의 한계
2.3 Class Activation Map
global average pooling을 이용
특정 클래스를 위한 CAM은 CNN이 클래스를 결정하기 위한
판별적 이미지 지역 같은 느낌.

cam 접근방법으로 두 도메인을 구별하여 제안된
discriminative image regions로 두드러지게 변화시켰다.
그러나 global average pooling 말고도 global max pooling도 쓰임.

3.U-GAT-IT
소스 도메인 X(s)로부터 타겟도메인 X(t)로의 매핑 G(s->t)을
각 도메인에서 뽑아낸 unpaired 샘플을 이용해서
학습시킨다.
프레임워크의 구조는 다음과 같다.
G(s->t)와 G(t->s), 그리고 2개의 discriminatior Ds와 Dt
우리는 어텐션 모듈을 G와 D 둘 다에 합친다.

D의 어텐션은 G가 진짜같은 이미지를 생성하는데
중요한 것에 집중할 수 있게 한다.

G의 어텐션은 다른 도메인과 구별되는 지역에
집중하도록 도와주는 역할을 한다.

3.1모델
Xs와 Xt는 각 도메인의 표본임.
G(s->t)는 Es와 Gt로 이루어짐.

auxiliary classifier *n*s(x) : Xs로부터 x가 나올 확률

Eks(x)는 인코더의 k번째 activation map
E kij s (x)는 그 맵의 (i,j)의 값.
CAM의 영향을 받아 auxiliary classifier는
global average pooling과 global max pooling을 이용해
k번째 피쳐맵의 중요한 가중치(w ks,)를 학습한다.

중요 가중치를 추출함으로써
우리는 계산할 수 있다
특정 어텐션 피쳐맵 as(x)=ws * Es(x) (1<=k<=n)
의 도메인셋을 계산 가능하다.
n은 encoded feature maps의 개수
G(s->t)는 Gt(as(x))와 같아진다.

affine transformation parameters in normalization layers와
combine normalization functions에 영향을 받아서
파라미터가 attention map의 FC layer에 의해
동적으로 계산되는 AdaLin으로 구성된
residual block을 갖췄다.

$$\hat{a_I} = \frac{a-\mu_I}{\sqrt{\sigma_I^2+\epsilon}}, \hat{a_L} = \frac{a-\mu_L}{\sqrt{\sigma_L^2+\epsilon}}$$

$$AdaLin(\alpha,\gamma,\beta) = \gamma \bullet (\rho \bullet \hat{a_I} + (1-\rho) \bullet \hat{a_L}) + \beta,$$
$$\rho \leftarrow clip_{[0,1]}(\rho - \tau \triangle \rho),$$

뮤i와 뮤L 그리고 시그마i와 시그마L은 각각 channel-wise, layer-wise 평균 그리고 표준 편차이다.(I가 channel-wise, L가 layer wise인 듯)

그리고 감마와 베타는 FC layer로부터의 파라미터고
tau는 러닝레이트, triangle rho는 옵티마이저가 결정하는 업데이트 벡터를 가리킨다.

rho의 값은 0과 1사이로 제한된다. 파라미터 업데이트 스텝에 경계를 부과하는 것.

제네레이터는 instance normalization이 중요한 작업에서 이러한 값들을 조정해서 rho의 값이 1에 가까워진다. 그리고 LN이 중요할 때에는 rho가 0으로 가까워진다.
rho의 값은 디코더의 res block에서 1로 시작하고 디코더의 up-sampling block에서 0으로 시작한다.

콘텐트 피쳐를 스타일 피쳐로 변환하는 최적의 방법은 화이트닝이랑 컬러링 트랜스폼. 그러나 행렬과 역행렬의 공분산 계산 때문에 계산적 비용이 너무 높다
그렇지만 AdaIN은 WCT보다 훨씬 빠르고,피쳐 채널간의 상관관계가 없다고 가정할 때 WCT에 비해 차선책이다.
그러므로 변환된 피쳐는 약간씩 콘텐트의 패턴으로 구성되어 있다.

반면에 LN은 채널간 상관관계가 없다고 가정하지 않는다.
그러나 가끔씩 LN은 content structure of original domain well
왜냐하면 그것은 피쳐맵의 global statistics만 고려하기 때문이다.

이것을 극복하려면 우리가 제안한 AdaLIN과 LN을 선택적으로 합친다?
선택적으로 content information을 지키거나 바꾸거나 해서. 이러면 넓은 범위의 img2img 변환 문제를 해결할 수 있다

3.1.2 Discriminator

$x \in \{X_t, G_{s\rightarrow t}(X_s)\}$는 각각 타겟 도메인과 translated 된 source domain을 의미
D_t는 인코더 E_{D_t},분류기 C_{D_t}+auxilary classifier η_{D_t}로 이루어짐
η_{D_t}와 D_t는 x가 어느 도메인에서 오든 학습을 시작한다.
표본 x가 주어지면 D_t(x)는 imporntance weight WDt와 encoded feature map EDt(x)에서 어텐션 피쳐맵 $a_{D_t}(x) = w_{D_t} * E_{D_t}(x)$을 추출한다.
EDt는 auxilary classifier에 의해 학습된다.

3.1.3 Loss function
4개의 loss function
GAN꺼 말고 우리는 the Least Squares GAN 목적함수를 쓴다 안정적 학습을위해

Adversarial loss
적대적 loss는 생성된 이미지를 타겟 이미지 분포에 맞추기 위해 쓴다
$L_{gan}^{s\rightarrow t} = E_{x\sim X_t}[(D_t(x))^2] + E_{x\sim X_s}[(1-D_t(G_{s\rightarrow t}(x)))^2]$

cycle loss
mode collapse 문제를 완화하기 위해 우리는 cycle consistency constraint to the generator.cycle 일관성 제약 조건을 제네레이터에 적용.
Xs의 이미지가 주어지면 Xs의 이미지의 Xt로의 그리고 Xt에서 Xs로의 연속적 변환이 이루어진 후에, 이미지는 반드시 성공적으로 원래의 도메인으로 변환되어야만 한다.
$L_{cycle}^{s\to t} = E_{x\sim X_s}[|G_{t\to s}(G_{s\to t}(x))|_1]$

Identity Loss
색 분포를 확실히 하고 아웃풋을 비슷하게 하려면
identity 일관성 제약조건을 제네레이터에 적용했다
변환된 이미지 Xt가 주어지면 Gst를 통해 변환된 후에 이미지는 그대로여야 한다.
$L_{identity}^{s\to t} = E_{x\sim X_t}[|x - G_{s\to t}(x)|_1]$

CAM loss
s와 Dt의 보조분류기에서 정보를 추출해냄으로써 주어진 이미지 x
Gst와 Dt는 알아야 한다 그들이 어디가 나아져야 하는지 또는 무엇이 현재 상태에서 두 도메인간의 차이를 만드는지.
$L_{CAM}^{s\to t} = -(E_{x\sim X_s}[\log(\eta_s(x))] + E_{x\sim X_t}[\log(1-\eta_{D_t}(G_{s\to t}(x)))^2]$
$L_{CAM}^{D_t} = E_{x\sim X_t}[(\eta_{D_t}(x))^2] + E_{x\sim X_t}[\log(1-\eta_{D_t}(G_{s\to t}(x)))^2]$

Full objective

Gst, Gts, eta s, eta t 최소화
Ds, Dt, eta Ds, eta Dt 최대화
$\lambda_1 L_{gan} + \lambda_2 L_{cycle} + \lambda_3 L_{identity} + \lambda_4 L_{cam}$

$\lambda_1 = 1, \lambda_2 = 10, \lambda 3 = 10, \lambda_4 = 1000$
$L_{gan} = L_{gan}^{s\to t} + L_{gan}^{t\to s}$
다른 loss들도 비슷한 방식으로 더해진다

4.Implementation
4.1 Network architecture
G의 encoder = 2개의 conv layer(stride=2) for down-sampling, 4 res block.
G의 decoder = 4 res block, 2 up-sampling conv layer(stride=1)

encoder에 instance normalization을,
decoder에 AdaLIN을 각각 사용함

일반적으로 LN은 분류문제에서 배치 정규화가 더 not better하다.
보조분류기가 G의 encoder에 연결될 때부터 보조분류기의 정확도를 증가시키기 위해 우리는 AdaLIN 대신에 instance normalization(1 크기의 미니배치 사이즈의 배치정규화)을 사용함.
Spectral normalization은 D에 사용된다.
local(70x70) 또는 global(286x 286) image patches가 real인지 fake인지 판단하는 Discriminatior network에 PatchGAN의 다른 2개의 scales(척도)를 사용한다.

activation function은 G는 ReLU, D에는 leacky ReLU(0.2)를 사용한다.

4.2 Training
모든 모델은 Adam($\beta_1 = 0.5, \beta_2 = 0.999$)를 이용해 학습된다.
data augmentation
-0.5의 확률로 flip horizontally
-resize 286x286

-random crop 256x256

batch size는 모든 실험에 1개로 맞췄다.(위의 instance normalization 때문인 것으로 추정)

learning rate = 0.0001 (50만 iterations까지)
그리고 그 이후부터 선형적으로 100만 iterations까지 감소한다.
가중치 감소율은 0.00001
가중치 초기화는 zero centered normal distribution (표준편차 0.02)

5.experiments
5.1 Baseline model
5.1.1. CycleGAN
5.1.2. UNIT
5.1.3. MUNIT
5.1.4. DRIT

5.2.Dataset
All images are resized to 256x256 training.
(후략)
5.3 Experiment results
Attention module와 AdaLIN 모델의 효과를 분석할 것이다.
그러고 나서 다른 이전 모델에서 보인 성능과 U-GAT-IT의 대한 성능을 비교할 것
변환된 이미지의 visual quality의 평가를 위해 user study를 수행했다?
user들은 5개의 다른 방법 중에 제일 나은 결과를 선택하도록 하는 식으로

5.3.1 CAM analysis

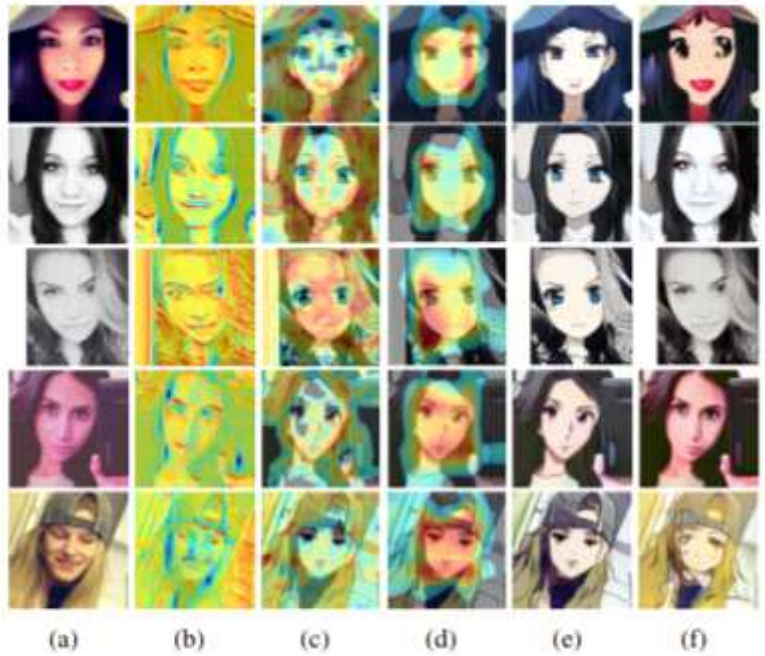

Figure 2. Visualization of the attention maps and their effects shown in the ablation experiments: (a) Source images, (b) Attention map of the generator, (c-d) Local and global attention maps of the discriminator, respectively. (e) Our results with CAM, (f) Results without CAM.

G와 D에 사용된 어텐션 모듈의 benefit을 확인하는 ablation study를 수행함