창직종합설계프로젝트 1

선행논문 분석 요약문

팀명: Your cartoon is

팀원: B611155 이유진, B511064 박성춘, B511170 장형주, B511157 이현수

논문1 < Video Key-Frame Extraction using Unsupervised Clustering and Mutual Comparison >

출처 International Journal of Image Processing (IJIP) 2016

Unsupervised clustering 과 mutual comparison을 사용하여 Video에서 key-frame을 추출하기 위한 논문. Mutual comparison에는 color component와 texture를 사용함.

이 논문에서는

1. Shot Boundary Detection (shot: video segmentation)

   video stream으로부터 visual features(color, shape, texture, motion) 또는 그 feature들의 결합을 뽑아내고 frame들 사이의 similarity를 측정함. 이때 similarity에는 g(i,i+k)사용, frame의 difference 또는 discontinuity를 나타냄

   $$g(i, i+k) = \sum_{x,y}(| I_i(x,y) - I_{i+k}(x,y) |)$$

   이때 I(X,Y)는 image pixel의 (x,y)좌표의 intensity level을 나타냄. 하지만 위의 식은 intensity의 변화나 object, camera motion에 매우 민감 false 확률이 증가할 수 있음.

2. key-Frame Extraction

   가장 선호되는 key frame 추출 방법은 histogram-based method(frame의 color 분배). Color histogram의 한계는 비슷한 histogram을 가진 이미지가 서로 다른 visual appearance를 가질 수도 있다는 것.

   Shot segmentation 과정을 거치고 나면 shot의 frame들이 매우 비슷하기 때문에 이중 shot의 내용을 가장 잘 반영하는 key-frame을 뽑아냄. 크게 6가지의 분야가 있으며 (sequential comparison-based, Global comparison-based, Reference frame-based, Clustering, Curve simplification-based, object/events) 이 중 clustering을 사용함

   Supervised clustering에서 shot boundary detection으로 얻어진 shot의 N개의 frame을 m개

의 cluster로 분류하고, 두 frame 사이의 similarity는 color와 texture요소의 가중치 조합 사용함. Color feature은 HSV color space의 global level histogram 사용. Texture feature은 GLCM을 사용했으며 GLCM에서 사용된 feature들은 Contrast, Correlation, Energy, Homogeneity 임.

Histogram similarity는 color bin들의 최소값의 합을 사용하는 histogram intersection method 사용하였으며 다음과 같음

$$Simi_{hsv} = \sum_{i=1}^{C} min\left(h_x(i), h_y(i)\right)$$

Hx와 hy는 frame x와 y 각각의 HSV histogram, C는 histogram 안의 color bin 수

GLCM texture features를 이용한 frame similarity index는 다음과 같음 Euclidean distance method를 사용하며 다음과 같음

$$Simi_{glcm} = \sqrt{\sum\left(glcm_{x-}glcm_{y}\right)^2}$$

Glcmx와 glcmy는 각각 frame x,y의 GLCM texture feature

위의 두 요소 모두가 videao frame의 visual content를 표현하는데 똑같이 효과적인 것이 아니므로 가중치 필요(feature들의 중요성에 기반) 논문에서는 color histogram에 70%, GLCM texture에 30% 따라서 결합된 식은

$$Combine_{Simi} = \left(Simi_{hsv} * 0.7 + Simi_{glcm} * 0.3\right)$$

위의 식을 이용하여 clustering 진행, 이때 clustering density를 제어하는 임계 값 δ 조절. δ가 높을수록 cluster가 많아짐. 새로 들어온 값에 대한 value가 δ보다 낮으면 이는 노드가 cluster에 추가되기에 충분하지 않다는 것을 나타냄. (이때 value는 node와 cluster 중심 사이의 유사성)

<Clustering 과정>

(1) 초기화: f1을 cluster1에 넣고 f1을 cluster1의 중심으로 설정, numCluster = 1

(2) 다음 frame fi를 가져옴 이때 frame pool이 비어 있으면 (6)번으로 이동

(3) Fi와 이미 존재하는 cluster(K = 1,2,…..,numCluster) 사이의 유사성을 비교: simi(fi,cluster)

(4) Maxsimi를 계산하여 어떤 cluster가 가장 가까운지 결정함

이때 Maxsimi 는 $max_{k=0}^{numCluster} Simi(f_i,\sigma_k).$

Maxsimi가 임계 값 δ보다 작으면 충분히 가까운 클러스터가 없음: (5)번으로 이동

그렇지 않으면 Maxsimi를 가진 cluster에 fi를 넣고 (6)번으로 이동

(5) numCluster = numCluster + 1, 새로운 cluster에 fi를 넣음

(6) cluster의 centroid를 조정, 이후 (2)번으로 이동

centroid 조정 = $c_{\sigma_k} = D/(D+1)c'_{\sigma_k} + 1=(D+1)\vec{f_i}$

가장 왼쪽의 c0K는 새로운 centroid, 중간의 c0K'은 이전의 centroid, D는 cluster 안의 frame의 수

Cluster가 끝나면 key-frame을 선택해야함. 이 논문에서는 cluster 내부의 frame 수가 min_clust_size = shot의 전체 frame수의 10% 보다 크면 충분히 큰 cluster 라고 판단하며 해당 cluster의 key-frame만 선택. Cluster 내부의 key-frame은 cluster의 중심에 가장 가까운 frame을 선택. Min_clust_size를 줄이면 cluster의 수가 증가하고 over-segmentation 생길 수 있다. 반대의 경우 under-segmentation 생길 수 있음.

논문 결과 : over-segmentation의 결과로 추정되는 중복 프레임이 발견됨. Min_clust_size를 줄여 보았지만 under-segmentation이 되는 한계 발견. 따라서 이를 mutual comparison을 사용하여 해결, key-frame을 다른 key-frame들과 비교하여 similarity가 특정한 임계 값보다 높으면 중복으로 판단 제거함.

그 결과 중복된 frame을 잘 제거함. 활동이 많은 video에서 더 많은 key-frame이 추출되고 활동이 적은 video 에서는 적은 key-frame이 나옴.

장점: easy to implement and fast to compute 무조건 첫 장면이 나오거나 하는 문제 없음, real time에 적용될 수 있음

논문2 < INSTAGAN: INSTANCE-AWARE IMAGE-TO-IMAGE TRANSLATION >

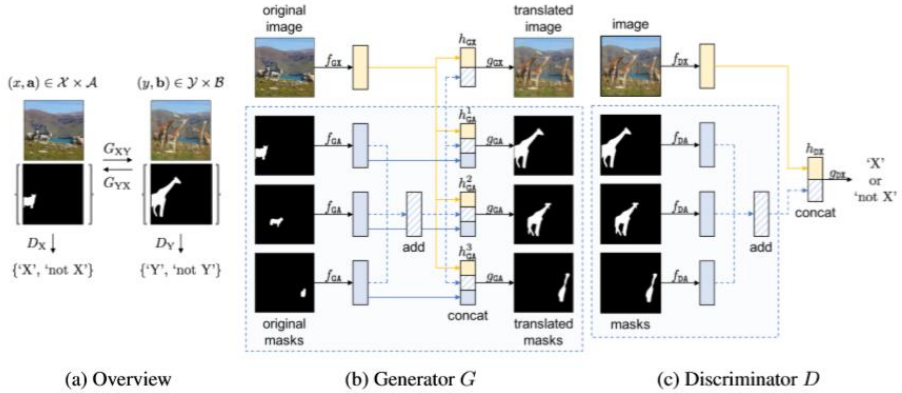
출처 Published as a conference paper at ICLR 2019

Image-to-image translation 에서 사진 속 인스턴스의 모양이 크게 변경되거나 사진속에 인스턴스가 여러 개 있을 경우 이미지를 제대로 변환하지 못하는 문제를 해결하기 위한 논문.

이 논문에서 사용하는 모델 (이하 insta gan)은

1. instance-augmented neural architecture

   이미지와 이미지 속 인스턴스의 속성 세트를 모두 변환하는 신경망 아키텍처 제안



(a) Overview (b) Generator G (c) Discriminator D

Figure 2: (a) Overview of InstaGAN, where generators G_{XY}, G_{YX} and discriminator D_X, D_Y follows the architectures in (b) and (c), respectively. Each network is designed to encode both an image and set of instance masks. G is permutation equivariant, and D is permutation invariant to the set order. To achieve properties, we sum features of all set elements for invariance, and then concatenate it with the identity mapping for equivariance.

2. a context preserving loss

   이미지 변환 시 변환하는 인스턴스에 초점을 맞추어 대상 인스턴스만 변환하고 백그라운드 컨텍스트(예 : 배경 또는 인스턴스의 방향과 같은 도메인 독립적 특성) 는 유지하기 위해 외부에서 identity function을 배우도록 하는 컨텍스트 보존 손실함수(context preserving loss) 정의.

   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

   (1) GAN loss (domain loss) : Goodfellow가 제안한 것으로, 생성기 G와 핀별기 D를 교대로 훈련함. 우수한 성능의 LSGAN방식 사용

$$\mathcal{L}_{\text{LSGAN}} = (D_X(x, a) - 1)^2 + D_X(G_{YX}(y, b))^2 + (D_Y(y, b) - 1)^2 + D_Y(G_{XY}(x, a))^2.$$

(2) Cycle-consistency loss = L_{cyc} , identity mapping loss = L_{idt} : 이미지 변환시 원래 컨텍스트를 잘 유지하도록 해줌

$$\mathcal{L}_{\text{cyc}} = \|G_{\text{YX}}(G_{\text{XY}}(x, \boldsymbol{a})) - (x, \boldsymbol{a})\|_1 + \|G_{\text{XY}}(G_{\text{YX}}(y, \boldsymbol{b})) - (y, \boldsymbol{b})\|_1,$$
$$\mathcal{L}_{\text{idt}} = \|G_{\text{XY}}(y, \boldsymbol{b}) - (y, \boldsymbol{b})\|_1 + \|G_{\text{YX}}(x, \boldsymbol{a}) - (x, \boldsymbol{a})\|_1.$$

$$\mathcal{L}_{\text{ctx}} = \|w(\boldsymbol{a}, \boldsymbol{b}') \odot (x - y')\|_1] + \|w(\boldsymbol{b}, \boldsymbol{a}') \odot (y - x')\|_1$$

(3) Context preserving loss = L_{ctx} : 인스턴스 외의 것(ex. 배경)은 그대로 두고 인스턴스만 변환하도록 함

따라서 imsta gan 의 최종 손실함수는

$$\mathcal{L}_{\text{InstaGAN}} = \underbrace{\mathcal{L}_{\text{LSGAN}}}_{\text{GAN (domain) loss}} + \underbrace{\lambda_{\text{cyc}}\mathcal{L}_{\text{cyc}} + \lambda_{\text{idt}}\mathcal{L}_{\text{idt}} + \lambda_{\text{ctx}}\mathcal{L}_{\text{ctx}}}_{\text{content loss}},$$

3. a sequential mini-batch inference/training technique

제한된 GPU 메모리를 사용하여 많은 수의 인스턴스 속성을 처리할 수 있도록 전체 세트를 한번에 수행하는 대신 대상 인스턴스 속성의 미니 배치를 순차적으로 변환, 그 과정에서 여러 중간샘플을 생성하여 훈련 중 데이터를 확대함으로써 이미지의 품질 향상

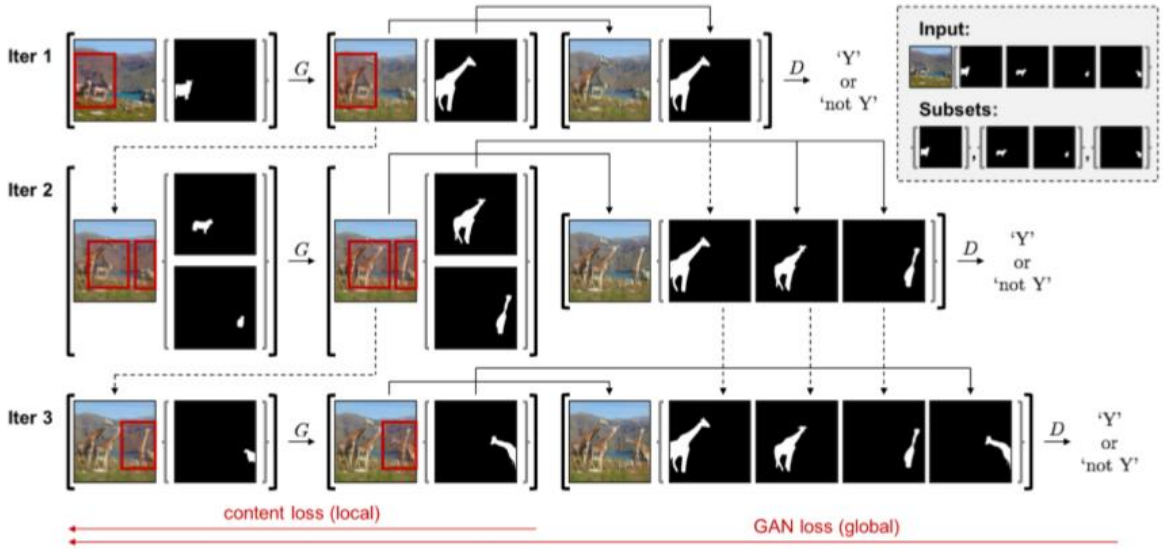


Figure 3: Overview of the sequential mini-batch training with instance subsets (mini-batches) of size 1,2, and 1, as shown in the top right side. The content loss is applied to the intermediate samples of current mini-batch, and GAN loss is applied to the samples of aggregated mini-batches. We detach every iteration in training, in that the real line indicates the backpropagated paths and dashed lines indicates the detached paths. See text for details.

Insta gan의 최종 training loss는

$$\mathcal{L}_{\texttt{InstaGAN-SM}} = \sum_{m=1}^{M} \mathcal{L}_{\texttt{LSGAN}}((x, \boldsymbol{a}), (y'_m, \boldsymbol{b}'_{1:m})) + \mathcal{L}_{\texttt{content}}((x_m, \boldsymbol{a}_m), (y'_m, \boldsymbol{b}'_m))$$

where $\mathcal{L}_{\texttt{content}} = \lambda_{\texttt{cyc}}\mathcal{L}_{\texttt{cyc}} + \lambda_{\texttt{idt}}\mathcal{L}_{\texttt{idt}} + \lambda_{\texttt{ctx}}\mathcal{L}_{\texttt{ctx}}$.

모든 m번째 훈련을 분리하기 때문에 training instance의 수에 상관없이 고정된 크기의 GPU 메모리만 필요함

논문 결과 : 기존의 cycle gan을 이용한 이미지 변환 시 인스턴스의 모양을 바꾸기는 힘들고, 바꾼다고 해도 원래 배경이 유지가 되지 않으며 정확도가 떨어지는 데에 반해 insta gan은 대상 인스턴스의 합리적인 형태를 생성하고 컨텍스트 보존 손실함수를 통해 인스턴스에 초점을 맞춤으로써 원래의 컨텍스트를 유지함.

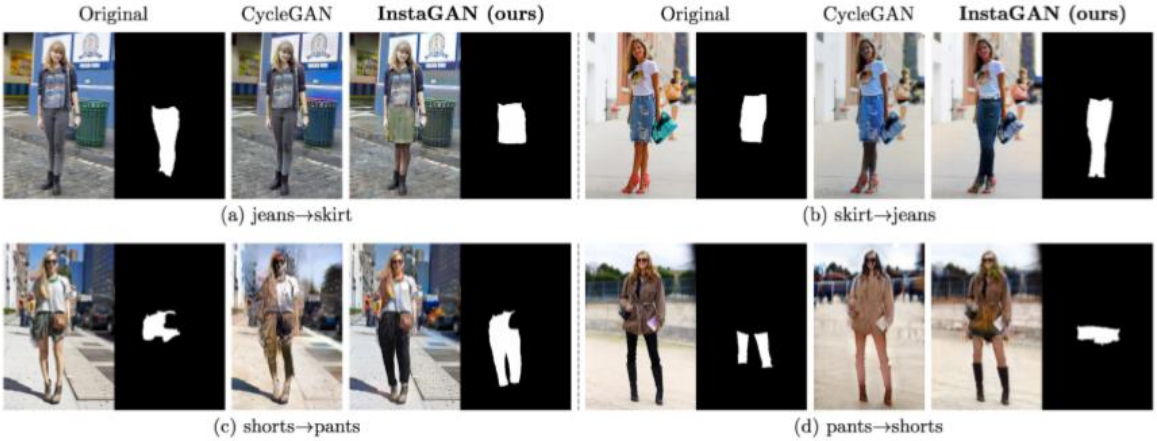


Figure 4: Translation results on clothing co-parsing (CCP) (Yang et al., 2014) dataset.

Figure 5: Translation results on multi-human parsing (MHP) (Zhao et al., 2018) dataset.

논문3 < CartoonGAN: Generative Adversarial Networks for Photo Cartoonization >

출처 The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018

실제 현실 이미지를 만화 스타일로 변환하는 것을 목표로 하며, 이미지를 양식화(Stylize)하는데 널리 사용되는 학습 방법인 GAN에 기초함.

만화 스타일은 높은 수준의 간단함과 추상적인 특징을 가지고, 깔끔한(clear) 가장자리와 부드러운 질감(texture)을 가지고 있는데, 현재 널리 사용되는 texture-descriptor-loss function으로는 이를 처리하기 힘들기 때문에 만족스러운 결과를 도출하지 못함.

따라서 CartoonGAN에서는 기존 GAN이 사용하는 loss function에 기초를 둔 두가지 loss function을 도입함.

1. Edge-promoting adversarial Loss

   만화 이미지에서 윤곽선은 매우 중요한 특성을 나타내는 특징이지만, 이미지 전체에서 이 윤곽선의 비율이 매우 적기 때문에, 기존의 adversarial Loss로는 이를 정확하게 학습시키는데 어려움이 있음. 이를 보완하기 위해, 학습을 위한 만화 이미지 $S_{data}(c)$와 $S_{data}(c)$에서 윤곽선을 제거한 $S_{data}(e)$를 이용하여 새로운 adversarial Loss를 사용.

   새로운 adversarial Loss, Edge-promoting adversarial Loss를 다음과 같이 정의

$$\begin{aligned}\mathcal{L}_{adv}(G, D) = &\mathbb{E}_{c_i \sim S_{data}(c)}[\log D(c_i)] \\ &+ \mathbb{E}_{e_j \sim S_{data}(e)}[\log(1 - D(e_j))] \\ &+ \mathbb{E}_{p_k \sim S_{data}(p)}[\log(1 - D(G(p_k)))].\end{aligned}$$

   기존 adversarial Loss에 두번째 항이 추가된 것으로, 윤곽선이 제거된 만화 이미지 e_j가 Discriminator D에 의해 거짓으로 판별되게 하여, 윤곽선에 대한 가중치를 올리도록 함.

2. Content Loss

   실제 현실 이미지에서 만화 스타일로 변화하는 과정에서 매우 중요한 목표 중 하나는 입력 사진에서 의미를 가진 내용을 보존하여 결과를 내는 것임. 이를 위해 CartoonGAN에서는 사전에 학습된 VGG network의 고수준 특징 맵(High-Level feature map)을 적용하였으며, 적용 결과 좋은 보존 결과를 얻을 수 있었음.

Content Loss를 다음과 같이 정의.

$$\mathcal{L}_{con}(G, D) = \mathbb{E}_{p_i \sim S_{data}(p)}[||VGG_l(G(p_i)) - VGG_l(p_i)||_1]$$

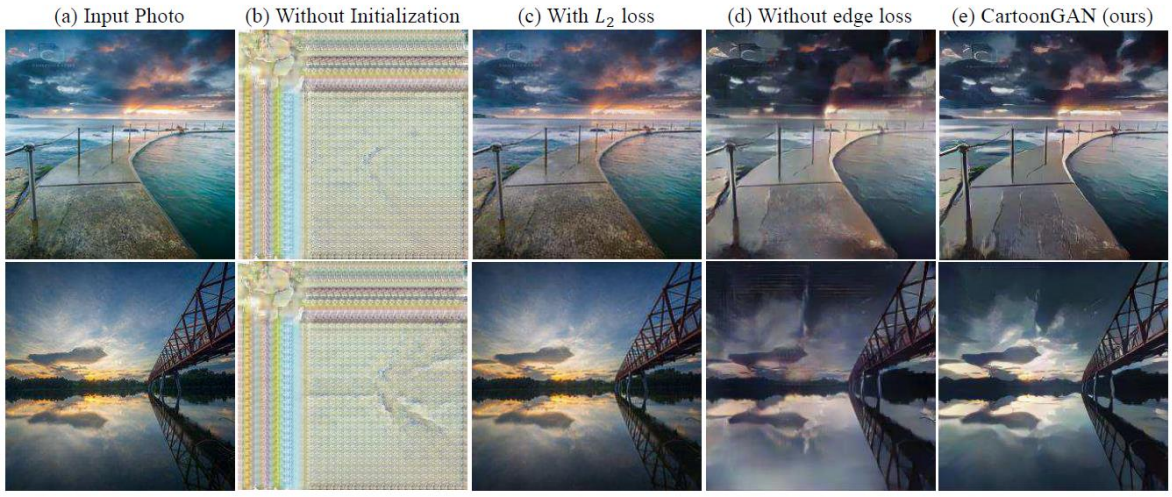
여기서 l은 특정 VGG 레이어의 특징 맵(feature map)을 나타냄.

이 두가지 Loss를 종합한 최종 Loss는

$$\mathcal{L}(G, D) = \mathcal{L}_{adv}(G, D) + \omega \mathcal{L}_{con}(G, D),$$

여기서 ω는 두 Loss에 대한 가중치 변수(Weight)로 ω가 커질수록 더 많은 content information이 보존됨. 본 CartoonGAN 논문에서는 가중치 변수 ω=10으로 두고 진행함.

추가적으로, CartoonGAN에서는 기존 GAN 모델이 매우 비선형(nonlinear)이기 때문에, 무작위로 초기화(Initialization)할 경우 원하는 방향으로 최적화되지 않을 가능성이 큼. 이를 방지하기 위해, CartoonGAN은 위의 semantic content Loss를 이용하여 생성자 G(Generator)를 사전에 학습. 생성자 G는 입력 이미지의 content를 보존하면서 만화 스타일로 재구성(Reconstruct)하도록 학습. 이 초기화 단계의 중요성은 아래 실험 결과를 통해 살펴볼 수 있음.

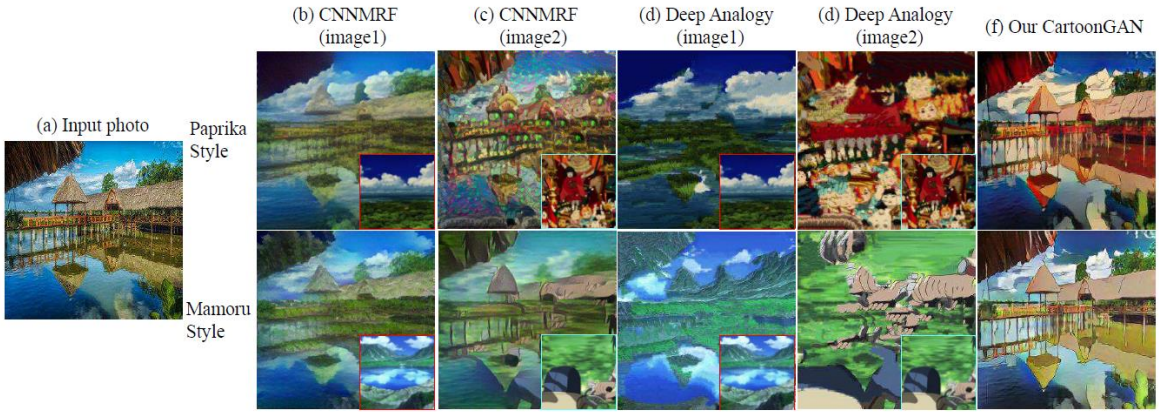


(b)에서 보이는 것처럼, 이미지의 몇몇 중요 특징들은 보이지만, 변환된 스타일이 예상과 매우 다른 것을 알 수 있음.

(c)는 L1 정규화를 사용하지 않고, L2 정규화를 사용한 결과로 L1 정규화와 L2 정규화에 관한 내용은 다음을 참고하면 좋음.

(https://developers.google.com/machine-learning/crash-course/regularization-for-

sparsity/l1-regularization)



위 그림은 CartoonGAN과 CNNMRF, Deep Analogy를 비교한 그림.

마무리로, 현재의 CartoonGAN의 loss function은 특정한 자연 환경을 만화 스타일로 변환하는 것만 다루지만, 유사한 아이디어가 다른 이미지를 합성하는 작업 등의 유용할 것이라 생각함. 또한, 학습 과정에 여러 연속적인 제약을 추가하여 영상으로 확대할 계획.

논문4 < LARGE SCALE GAN TRAINING FOR HIGH FIDELITY NATURAL IMAGE SYNTHESIS >

출처 Published as a conference paper at ICLR 2019

최근 이미지 모델링의 발전에도 불구하고 ImageNet과 같은 복잡한 데이터 세트에서 고해상도의 다양한 샘플은 생성하는 것은 어려움. 이를 위해 이 논문에서는 최대 규모로 GAN을 훈련시키고 그 규모에서 불안정성을 연구함.

이 논문에서 사용하는 모델(이하 BigGAN)의 평가 척도는 클수록 좋은 IS(Inseption Score)와 작을 수록 좋은 FID(Fr´echet Inception Distance) 를 사용.

BigGAN은

1. Scailing up GANS

   스케일링을 이용하여 FID와 IS값을 향상시킴. 한 generator G 단계마다 두번의 discriminator D 단계를 수행하고, 기존보다 많은 매개변수와 배치크기를 늘려 모델을 훈련시킴.

   배치크기를 8배 늘려 IS가 46% 향상되었고, Layer의 채널 수를 50%늘려 매개변수의 수가 두배로 늘어나 IS가 추가로 21% 향상됨

   공유 임베딩 방식으로 계산 및 메모리 비용을 줄이고 훈련속도를 37% 향상시켰고, skip-z를 이용하여 성능을 약 4% 향상시키고 훈련 속도 또한 18% 향상됨.

Batch	Ch.	Param (M)	Shared	Skip-z	Ortho.	Itr $\times 10^3$	FID	IS
256	64	81.5	SA-GAN Baseline			1000	18.65	52.52
512	64	81.5	✗	✗	✗	1000	15.30	58.77(±1.18)
1024	64	81.5	✗	✗	✗	1000	14.88	63.03(±1.42)
2048	64	81.5	✗	✗	✗	732	12.39	76.85(±3.83)
2048	96	173.5	✗	✗	✗	295(±18)	9.54(±0.62)	92.98(±4.27)
2048	96	160.6	✓	✗	✗	185(±11)	9.18(±0.13)	94.94(±1.32)
2048	96	158.3	✓	✓	✗	152(±7)	8.73(±0.45)	98.76(±2.84)
2048	96	158.3	✓	✓	✓	165(±13)	8.51(±0.32)	99.31(±2.10)
2048	64	71.3	✓	✓	✓	371(±7)	10.48(±0.10)	86.90(±0.61)

Table 1: Fréchet Inception Distance (FID, lower is better) and Inception Score (IS, higher is better) for ablations of our proposed modifications. *Batch* is batch size, *Param* is total number of parameters, *Ch.* is the channel multiplier representing the number of units in each layer, *Shared* is using shared embeddings, *Skip-z* is using skip connections from the latent to multiple layers, *Ortho.* is Orthogonal Regularization, and *Itr* indicates if the setting is stable to 10^6 iterations, or it collapses at the given iteration. Other than rows 1-4, results are computed across 8 random initializations.

2. Trading off variety and fidelity with the TRUNCATION TRICK

z ~ N (0, I)로 훈련 된 모델을 절단하고 잘린 법선에서 z를 샘플링하면 (범위를 벗어난 값이 해당 범위 내에 들어가도록 재 샘플링 됨) IS와 FID가 즉시 향상됨. 이것을 Truncation trick(절단 트릭)이라고 부름.

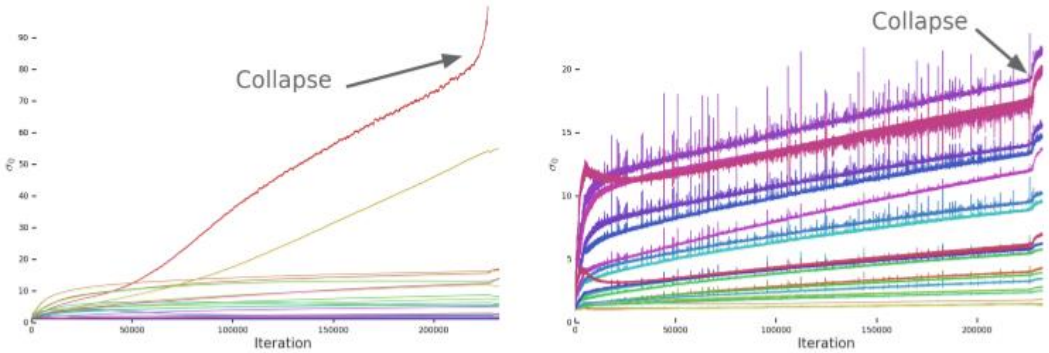
선택된 임계 값보다 큰 크기로 값을 resampling하여 z 벡터를 잘라 내면 전체 샘플 다양성은 감소하지만 개별 샘플 품질이 향상되고, 이것이 정확도와 다양성의 trade-off임.

BigGAN 은 모델이 Truncation trick에 잘 반응하게 하기위해 orthorgonal normalization (직교 정규화)를 사용하여 trade-off를 제어함.

하지만 GAN 모델은 크기가 커지면 불안정해지고 결국 모델이 붕괴되는 상황이 발생하는데 이 논문에서는 그 이유를 Generator G 와 Discriminator D 에서 찾음.

(1) Characterizing instability: The Generator

GAN 의 안정성을 조사하기 위해 G 에서 훈련 중 붕괴될 수 있는 지표를 검색하니 각 가중치 행렬의 상위 3 개 값 σ0, σ1, σ2 가 나옴. 보통 G 의 첫번째 레이어가 불안정함.



(a) **G** (b) **D**

Figure 3: A typical plot of the first singular value σ_0 in the layers of **G** (a) and **D** (b) before Spectral Normalization. Most layers in **G** have well-behaved spectra, but without constraints a small subset grow throughout training and explode at collapse. **D**'s spectra are noisier but otherwise better-behaved. Colors from red to violet indicate increasing depth.

이 레이어가 붕괴의 원인인지를 파악하기 위해 G 에 추가 conditioning 을 적용.

각 가중치의 상위 특이 값 σ0 을 고정값인 σreg 또는 두 번째 특이 값 r · sg (σ1)의 일부 비율 r 로 직접 정규화 했더니 σ1 이 증가함. 이때 σ0 을 clamping(특정 범위 안으로 제한)하기 위해 부분 특이값 분해를 사용.

$$W = W - \max(0, \sigma_0 - \sigma_{clamp}) v_0 u_0^\top,$$

Spectral Normalization 의 유무에 관계없이 이러한 기술은 σ0 또는 σ1 의 점진적인 증가 및 폭발을 방지하는 효과가 있지만, conditioning 은 훈련 붕괴를 방지하지 못하므로 D 를 조사함.

(2) Characterizing instability: The Discriminator

G와 마찬가지로 D의 가중치 스펙트럼을 분석한 다음 추가 제약 조건을 적용하여 훈련을 안정화시킴. 그림 3 (b)는 D에 대한 전형적인 σ0 플롯을 보여줌. G와는 달리, 스펙트럼은 노이즈가 있고 σ0/σ1은 잘 작동하며 특이값은 훈련 기간 동안 커지지만 폭발하지 않고 붕괴될 때만 증가.

D 의 스펙트럼이 급등하면 주기적으로 매우 큰 기울기가 수신될 수 있지만 Frobenius 규범이 부드럽다는 것을 알 수 있음.

이 noise 가 적대적 훈련 과정을 통한 최적화의 결과.

이 스펙트럼 노이즈가 불안정성과 인과 관계가 있는 경우, 자연스러운 카운터는 gradient panelty 를 사용하여 D 의 Jacobian 의 변경을 명시 적으로 규칙 화하는 것. 따라서 우리는 Mescheder 등의 R1 제로 중심 gradient penalty 를 탐구함.

$$R_1 := \frac{\gamma}{2}\mathbb{E}_{p_{\mathcal{D}}(x)}\left[\|\nabla D(x)\|_F^2\right].$$

γ 강도가 10 이면 훈련이 안정되고 G 와 D 에서 스펙트럼의 부드러움과 경계가 향상되지만 성능이 크게 저하되어 IS 가 45 % 감소함. 페널티 강도가 1 (급격한 붕괴가 발생하지 않는 가장 낮은 강도)로 감소하더라도 IS 는 20 % 감소함.

논문 결과 : 안정성은 G 또는 D에서만이 아니라 적대적 훈련 과정을 통한 상호 작용에서 비롯된 것. 열악한 conditioning의 증상은 불안정성을 추적하고 식별하는 데 사용될 수 있지만, 트레이닝에 필요한 합리적인 conditioning을 보장하여 최종 트레이닝 붕괴를 방지하기에 충분. D를 강력하게 제한하여 안정성을 강화할 수 있지만 그렇게 하면 성능이 크게 저하됨. 현재의 기술을 사용하면 이 conditioning을 완화하고 훈련 후반 단계에서 붕괴가 발생하여 더 나은 최종 성능을 달성할 수 있으며, 이 때 모델은 충분한 결과를 얻도록 훈련됨.

Generative Adversarial Networks가 정확도 및 생성된 샘플의 다양성 측면에서 scale up을 통해 여러 범주의 자연 이미지를 모델링하도록 훈련되었음을 입증

대규모 GAN의 훈련 행동에 대한 분석을 제시하고, 무게의 단일 값으로 안정성을 특성화하고 안정성과 성능 사이의 상호 작용에 대해 논의.