



Few-Shot Adversarial Learning of Realistic Neural Talking Head Models



Few-shot Learning을 통해 매우 적은 개수의 이미지로도 사람 얼굴을 재현해 낼 수 있는 talking head model을 개발해 보자.

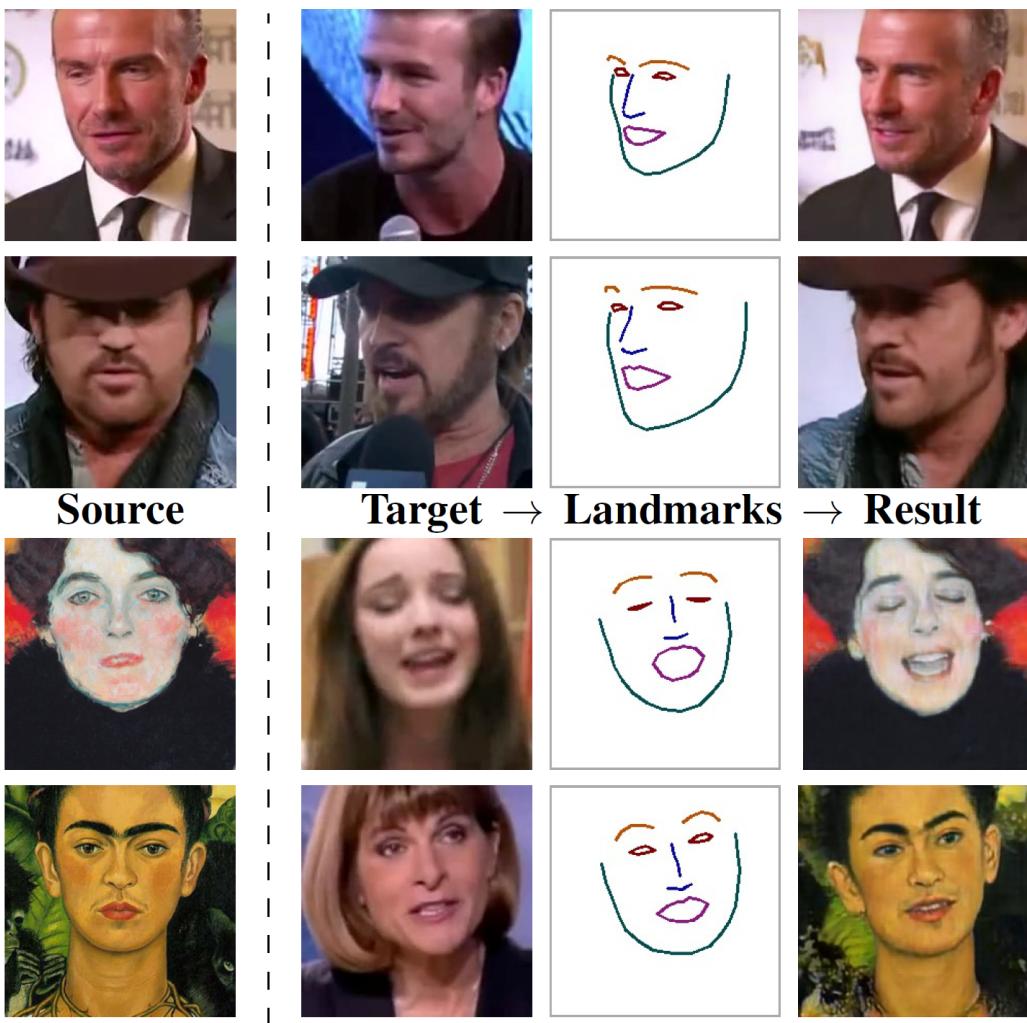
▼ One-shot Learning

- 모델이 단 하나의 예시만을 통해 새로운 클래스를 인식하거나 구별하는 능력을 가지게 하는 학습 방법
 - 하나의 데이터 포인트를 보고 새로운 클래스를 학습하고 구분할 수 있어야 함
- 예시)
 - 고양이라는 클래스를 학습한 모델에게 고양이 사진 하나만 보여주고 그 다음부터는 다른 이미지를 인식하도록 요구

▼ Few-shot Learning

- 모델이 매우 제한된 수의 예시만을 통해 새로운 작업이나 클래스를 학습하는 능력을 가지게 하는 학습 방법
- 보통은 소수의 예시 (예: 1~5개)를 통해 모델을 학습시키고 이를 기반으로 새로운 작업을 수행하거나 클래스를 인식하도록 함
- 실제 세계에서 데이터를 수집하고 레이블을 지정하는 것이 비용이 많이 드는 경우에 유용

1. Introduction



- 목표) Target의 Landmarks를 이용해 Source Style의 Result를 만드는 것
- 사람 사진의 Source는 8장, 예술 작품들은 1장만 가지고 학습시킴

- 해당 연구는 개인화된 사실적인 토킹 헤드 모델을 만드는 작업을 다룸
 - 해당 모델은 어떤 사람의 얼굴 랜드마크를 기반으로 사실적인 머리 이미지를 생성하여 해당 인물의 말과 표정을 동영상으로 합성할 수 있음
 - 이러한 능력은 원격 회의, 게임 등의 분야에서의 실용적인 응용으로 이어질 수 있음
- 현재 이러한 모델을 만드는 것에는 여러 어려움이 존재
 - 이는 인간 머리의 복잡성과 인간 시각 시스템의 민감성 때문
 - 이를 극복하기 위해 몇 가지 방법(warping-based synthesis, GAN을 활용한 direct synthesis 등)이 제안되었지만, 대부분의 방법들은 많은 양의 데이터와 수많은 계산이 요구됨
- 해당 연구에서는 소수의 사진과 제한된 훈련 시간으로 토킹 헤드 모델을 생성하는 시스템을 제안함

- 다양한 외관의 사람들이 나타나는 대량의 talking head video로 meta-learning 진행
 - face landmark로 사람의 이미지를 만드는 방법을 학습하므로 메타학습임
 - few-shot learning ability를 얻음
 - 구체적으로, 비디오에서 가져온 동일한 인물의 사진 여러장과 face landmark를 이용해 face landmark와 동일하면서 현실적인 이미지를 만드는 작업을 학습
- 소량의 새로운 사람 사진을 이용해 adversarial learning을 수행하여 적은 학습으로도 입력한 사람의 특징을 가진 현실적인 이미지를 생성함
 - landmark로 이미지를 만드는 법을 배웠으므로, 내가 원하는 인물 사진을 만들도록 fine-tuning을 수행함

▼ meta learning

- 기계 학습의 한 분야로, 모델이 새로운 작업이나 환경에 대해 학습할 수 있는 능력을 개발하는 데 중점을 두는 학습 방법
- 주어진 작업에 대한 단일 학습 세트를 가지고 여러 다른 작업에 적용할 수 있도록 모델을 훈련시키는 것을 목표로 함
- 일반적으로 두 가지 레벨의 학습을 포함
 1. 내부 레벨(inner level) 학습
 - 여러 작업에서 모델이 얼마나 잘 수행되는지를 개선하는 데 사용
 2. 외부 레벨(outer level) 학습
 - 다양한 작업에 적응할 수 있는 모델의 초기화 상태를 개선
- 소수의 데이터로부터 효과적인 학습을 할 수 있도록 하는 few-shot learning 문제의 해결에 활용됨
 - 모델이 새로운 작업에 대해 더 빠르게 학습할 수 있도록 함
 - 적은 양의 데이터로도 높은 일반화 성능을 달성할 수 있도록 함

▼ 적대적 학습(adversarial learning)

- 기계 학습의 한 방법으로, 두 개의 모델이 서로 경쟁하는 방식으로 학습을 진행
 - 생성자(generator) 모델 vs 판별자(discriminator) 모델
 - 생성자: 입력 데이터를 생성하여 진짜 데이터와 구별하기 어렵게 만들려고 노력
 - 판별자: 생성된 데이터와 진짜 데이터를 구별하려고 노력

- 서로 경쟁하면서 점차적으로 더 현실적인 데이터를 생성하는 생성자를 훈련시키는 방식
- 주로 생성 모델을 학습하는 데 활용됨
 - 사진, 음악, 텍스트 등과 같은 다양한 종류의 데이터를 생성하는 데 적용될 수 있음
 - 특히 데이터가 제한적이거나 쉽게 얻을 수 없는 경우에 유용한 방식

2. Related Work

- 해당 연구에서는 인간 얼굴의 외관을 통계적으로 모델링하는 작업과 최근의 이미지 생성 모델링에서의 진전을 기반으로하여 개인화된 토킹 헤드 모델을 만들기 위한 시스템을 제안
- ⇒ 적대적 훈련과 조건부 판별자를 활용

Face modeling

- talking head modeling과 깊은 연관이 있으나, talking head가 더 복잡함

Adaptive Instance Normalization(AdaIN)

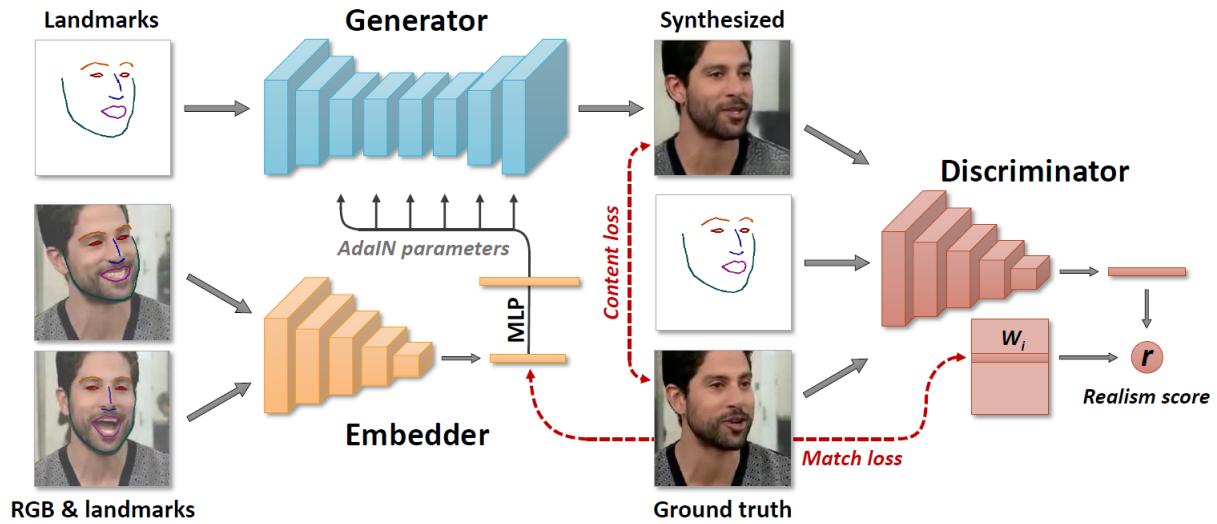
- 주로 조건 생성 모델에 이용되는 기법
 - 논문에서는 메타학습을 위해 사용되었음

Model-Agnostic Meta-Learner(MAML)

- 이미지 분류를 위한 분류기의 초기 상태를 얻기 위해 메타학습 활용
 - 학습에서 보지 못한 몇몇 데이터만 가지고 빠르게 학습시키는 방법

3. Methods

3-1. Architecture and notation



- 메타학습 단계
 - Embedder에 동일 비디오의 여러 사진들을 넣고, 결과를 평균내 embedding vector를 만듦
 - AdaIN으로 Generator의 파라미터를 조정
- Generator
 - Embedder에서 사용하지 않은 이미지의 Landmarks를 가지고 Synthesized를 만들고 Ground truth와 비교
- Discriminator
 - Synthesized의 Realism score는 낮게, Ground truth는 높게 만들도록 학습
- 다음의 3개의 network를 훈련시킴



공통사항

- $x_i(t)$: i 번째 비디오의 t 번째 프레임
- $y_i(t)$: $x_i(t)$ 의 landmark image

1. Embedder

$$E(\mathbf{x}_i(s), \mathbf{y}_i(s); \phi) = \hat{\mathbf{e}}_i(s)$$

- $\hat{\mathbf{e}}_i(s)$: N-dimensional embedding vector
 - ϕ : 네트워크 파라미터

- $\hat{\mathbf{e}}_i(s)$ 가 특정 프레임 s 에서의 자세/표정 등과 무관한 그 사람의 특징 같은 person-specific한 정보가 되도록 함

2. Generator

$$G(\mathbf{y}_i(t), \hat{\mathbf{e}}_i; \psi, \mathbf{P}) = \hat{\mathbf{x}}_i(t)$$

- $\hat{x}_i(t) : y_i(t)$ 를 통해 만들어 낸 합성 이미지
- Generator의 파라미터는 person-generic ψ 와 person-specific ψ 로 이루어짐
 - ψ : 직접 학습됨
 - ψ : $P\hat{e}_i$ 로 구현

3. Discriminator

$$D(\mathbf{x}_i(t), \mathbf{y}_i(t), i; \theta, \mathbf{W}, \mathbf{w}_0, b) = r$$

- 출력으로 N -dimensional vector를 만드는 ConvNet $V(x_i(t), y_i(t); \theta)$ 와 이를 Projection하는 부분으로 구성
- 입력된 이미지 $x_i(t)$ 가 i 번째 비디오에 실제로 있는지, $y_i(t)$ 와 일치하는지를 나타내는 **realism score** r 를 출력
 - W : i 번째 비디오와 관련, person-specific한 정보와 관련
 - w_0, b : person-generic한 정보와 관련

3-2. Meta-Learning stage

- 3가지 네트워크를 모두 학습시킴
 - k-shot learning 방법 활용(여기서는 $k = 8$)
 - 동일한 i 번째 비디오에서 k 개의 frame(s_1, s_2, \dots, s_K)를 가져와 활용
 - 이때 \hat{e}_i 는 하나여야 함
- Embedder와 Generator는 아래 loss function을 최적화 시키는 방법으로 학습을 진행함

$$\begin{aligned} \hat{\mathbf{e}}_i(s) &= \frac{1}{N} \sum^K k = 1 E(\mathbf{x}_i(s_k), \mathbf{y}_i(s_k); \phi) \\ \mathcal{L}(\phi, \psi, \mathbf{P}, \theta, \mathbf{W}, \mathbf{w}_0, b) &= \mathcal{L}_{\text{CNT}}(\phi, \psi, \mathbf{P}) + \\ &\quad \mathcal{L}_{\text{ADV}}(\phi, \psi, \mathbf{P}, \theta, \mathbf{W}, \mathbf{w}_0, b) + \mathcal{L}_{\text{MCH}}(\phi, \mathbf{W}) \end{aligned}$$

- ϕ : Embedder parameter

- ψ, P : Generator parameter
 - θ, W, w_0, b : Discrimminator parameter
-
- L_{CNT} : ground truth($x_i(t)$)와 $\hat{x}_i(t)$ 를 유사하게 만들기 위해 사용하는 perceptual loss
 - VGG19와 VGGFace를 이용해 두 개를 구한 후 가중합
 - L_{ADV} : Discriminator의 출력 realism score r 를 최대화 하고, 학습 안정성을 위해 Discriminator를 이용한 perceptual loss로 구성

$$\mathcal{L}_{ADV}(\phi, \psi, \mathbf{P}, \theta, \mathbf{W}, \mathbf{w}_0, b) = -D(\hat{\mathbf{x}}_i(t), \mathbf{y}_i(t), i; \theta, \mathbf{W}, \mathbf{w}_0, b) + \mathcal{L}_{FM}$$
 - Realism score $r (= D)$
$$D(\hat{\mathbf{x}}_i(t), \mathbf{y}_i(t), i; \theta, \mathbf{W}, \mathbf{w}_0, b) = V(\hat{\mathbf{x}}_i(t), \mathbf{y}_i(t); \theta)^T (\mathbf{W}_i + \mathbf{w}_0) + b$$
 - V : Discriminator의 일부인 ConvNet
 - W 의 i 번째 열은 i 번째 비디오(or style)과 관련
 - w_0, b : landmark 및 \hat{x}_i 의 일반적인 특성과 관련
 - L_{FM} : L_{CNT} 와 유사하지만, 사용된 네트워크가 Discriminator의 V 임

⇒ 앞의 방법으로 Embedder의 ϕ 와 Generator의 ψ 를 학습시키고, Discriminator의 θ, W, w_0, b 를 학습

$$\begin{aligned} \mathcal{L}_{DSC}(\phi, \psi, \mathbf{P}, \theta, \mathbf{W}, \mathbf{w}_0, b) &= \\ \max(0, 1 + D(\hat{\mathbf{x}}_i(t), \mathbf{y}_i(t), i; \phi, \psi, \theta, \mathbf{W}, \mathbf{w}_0, b)) &+ \\ \max(0, 1 - D(\mathbf{x}_i(t), \mathbf{y}_i(t), i; \theta, \mathbf{W}, \mathbf{w}_0, b)) \end{aligned}$$

- 만들어진 이미지 $\hat{x}_i(t)$ r 은 작게(첫 번째 max, 실제 이미지 $x_i(t)$ 의 r 은 크게(두 번째 max) 만들도록 학습시킴

3-3. Few-shot learning by fine-tuning

- 메타학습이 끝나면 메타학습에서 사용하지 않은 데이터와 landmark를 이용해 특정 스타일(사람)의 사진을 생성함

- 이때, 메타 학습과 유사하게 T 개의 이미지로 스타일을 학습함
- 해당 이미지들 역시 landmark가 있어야 함
- T 의 값이 K 와 같을 필요는 없음
- 메타 학습에서 이용하지 않은 새로운 스타일(사람)의 이미지를 원함
 - i는 필요로 하지 x
 - Embedder의 출력은 다음과 같음

$$\hat{\mathbf{e}}_{NEW} = \frac{1}{N} \sum_{t=1}^T E(\mathbf{x}(t), \mathbf{y}(t); \phi)$$
- 더 나은 이미지를 위한 fine-tuning을 진행하였음
 - Generator: $G(\mathbf{y}(t), \hat{\mathbf{e}}_{NEW}; \psi, \mathbf{P}) \rightarrow G'(\mathbf{y}(t); \psi, \psi')$
 - $\hat{e}_i(t)$ 와 P 를 이용해 계산해서 사용한 person-specific 파라미터 ψ' 도 ψ 와 함께 학습하고, $\psi' = Pe_{NEW}$ 로 파라미터의 초기값으로 설정
 - Discriminator

$$D(\mathbf{x}_i(t), \mathbf{y}_i(t), i; \theta, \mathbf{W}, \mathbf{w}_0, b)$$

$$\rightarrow D'(\mathbf{x}(t), \mathbf{y}(t); \theta, \mathbf{w}', b) = V(\hat{\mathbf{x}}(t), \mathbf{y}(t); \theta)^T \mathbf{w}' + b$$
 - 메타학습에서 사용한 데이터를 사용 x → i 가 없음
 - W_i 도 사용 불가
 - 대신, L_{MCH} 를 이용해 \hat{e}_i 와 W_i 간의 유사도를 증가시키도록 학습시킴
 - $\mathbf{w}' = \mathbf{w}_0 + \hat{\mathbf{e}}_{NEW}$ 로 초기화

⇒ 각 모델의 파라미터를 새롭게 초기화

- 메타학습과 유사하게 Generator부터 학습시킴

$$\mathcal{L}'(\psi, \psi', \theta, \mathbf{w}', b) = \mathcal{L}'_{CNT}(\psi, \psi') + \mathcal{L}'_{ADV}(\psi, \psi', \theta, \mathbf{w}', b)$$

- 이후 Discriminator를 학습시킴

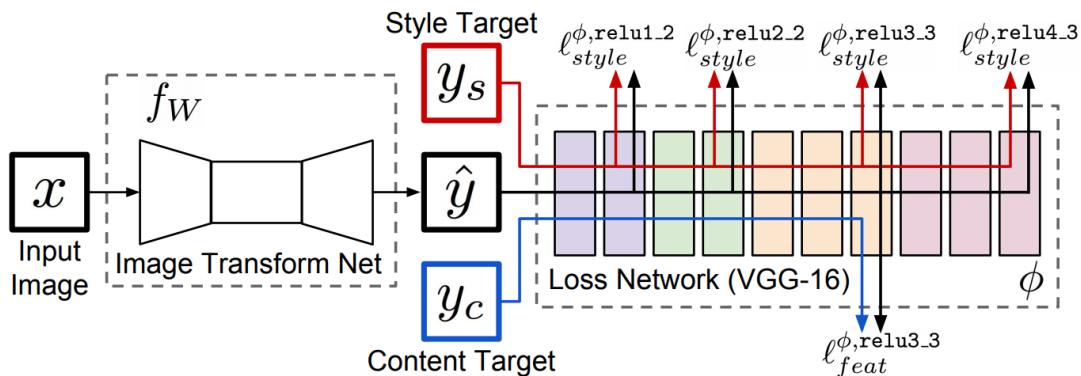
$$\begin{aligned} \mathcal{L}'_{DSC}(\psi, \psi', \theta, \mathbf{w}', b) = & \\ \max(0, 1 + D(\hat{\mathbf{x}}(t), \mathbf{y}(t); \psi, \psi', \theta, \mathbf{w}', b)) + & \\ \max(0, 1 - D(\mathbf{x}(t), \mathbf{y}(t); \theta, \mathbf{w}', b)) & \end{aligned}$$

→ 해당 과정을 fine-tuning이 완료될 때까지 반복

3-4. Implementation details

- **Generator**

- 다음의 Perceptual loss의 구조를 가져옴



- down/upsampling을 residual block으로 대체함

- BN: AdaIN이나 IN으로 대체

- **Embedder**

- Generator에 사용된 것과 같지만, normalization layer는 다른 residual downsampling block으로 구성하였음
- 벡터 출력을 얻기 위해 마지막에 해상도에 관해 global sum pooling와 ReLU를 추가하였음

- **Discriminator**

- Embedder와 동일하지만 마지막 pooling전에 4×4 의 residual block을 추가하였음

-
- 모든 모델에 공통적으로 **spectral normalization**을 적용하였음
 - Generator의 upsampling part와 모든 네트워크의 downsampling part에 self-attention block을 사용하였음
 - Spectral normalization
 - 해당 논문에서 처음 제시된 방식
 - 간단한 구현과 적은 계산량으로 효율적인 하이퍼 파라미터 튜닝을 가능하게 해주는 weight normalization 방법의 일종
 - 학습 동안 Generator와 Embedder를 학습할 때마다 Discriminator는 두 번씩 학습시켰음

4. Experiments

Datasets

- VoxCeleb1(256p, 1fps 비디오)와 VoxCeleb2(224p, 25fps 비디오)를 사용
 - VoxCeleb1: baseline 비교와 ablation studies에 활용
 - VoxCeleb2: 최고 성능을 보이는 데 활용

Metrics

Frechet-inception distance(FID)

- 실제 데이터와 생성된 데이터에서 얻은 feature의 평균과 공분산을 비교하여 구함
- 이때 Inception 네트워크를 활용 \Rightarrow inception distance라 함
 - ▼ Inception Network
 - 구글이 개발한 딥러닝 아키텍처 중 하나로, 이미지 분류 및 인식 작업을 위해 사용됨
 - 기존의 일련의 합성곱(Convolution) 및 풀링(Pooling) 레이어를 쌓은 구조 대신, 인셉션 모듈이라는 특별한 형태의 모듈을 사용하여 네트워크의 깊이를 증가 시킴
 - 여러 크기의 커널을 동시에 적용하는 것으로 구성됨
 - \rightarrow 네트워크가 서로 다른 크기의 특징을 동시에 추출할 수 있도록 도와줌
 - 1×1 컨볼루션 레이어를 사용하여 차원 감소를 수행
 - \rightarrow 네트워크가 더 적은 파라미터를 가지고도 효과적으로 특징을 추출할 수 있도록 도와줌

Structured Similarity(SSIM)

- ground truth와의 low-level similarity를 측정

Cosine Similarity(CSIM)

- SOTA face recognition 네트워크와 본 모델의 embedding vector간 CSIM

USER

- 사람응답자에게 원본 두 장, 생성된 이미지 한장을 보여주었을때 생성된 이미지를 선택한 비율을 산출

Methods

X2Face

- Warping-based method의 baseline
- ptr-trained model 활용
- 이미지를 생성할때 direct method와 달리 landmark보다 많은 정보를 포함한 입력(ground truth를 계산하여 얻고, 불공정한 이점이 된다)을 사용하였음

Pix2pixHD

- Direct synthesis method의 baseline
- 논문의 저자가 제안한 방법으로 모델을 구성하고 해당 논문의 모델과 동일한 데이터로 처음부터 학습시킴
- fine-tuning 시 X2Face에 비해 40 epoch이 추가로 요구되었음

Comparison results

Method (T)	FID↓	SSIM↑	CSIM↑	USER↓
VoxCeleb1				
X2Face (1)	45.8	0.68	0.16	0.82
Pix2pixHD (1)	42.7	0.56	0.09	0.82
Ours (1)	43.0	0.67	0.15	0.62
X2Face (8)	51.5	0.73	0.17	0.83
Pix2pixHD (8)	35.1	0.64	0.12	0.79
Ours (8)	38.0	0.71	0.17	0.62
X2Face (32)	56.5	0.75	0.18	0.85
Pix2pixHD (32)	24.0	0.70	0.16	0.71
Ours (32)	29.5	0.74	0.19	0.61

- T: fine-tuning에서 모델이 본 이미지의 수

- 결과
 - X2Face는 학습과정에서 L2 loss를 사용하므로 SSIM이 높음
 - Pix2pixHD는 perceptual loss만을 이용해 학습하므로 FID는 낮지만 identity perservation을 고려하지 않아 CSIM이 낮음
- CSIM은 최종 이미지의 품질과 관련이 높지만, 이미지의 현실성이나 blurry등은 고려하지 못한다는 단점 존재
 - 사실상 USER를 제외하면 Uncanny Valley를 고려하지 않으므로 의미 없을 수 있음



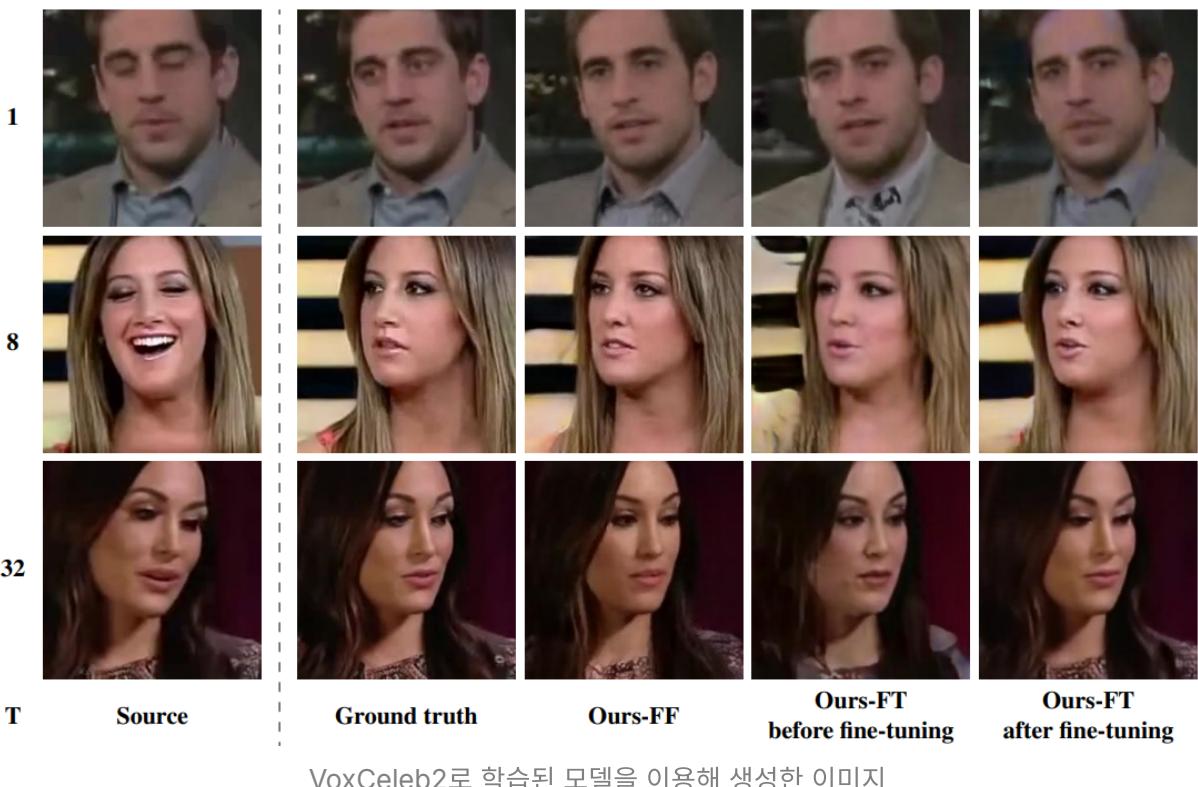
VoxCeleb1로 학습된 모델을 이용해 생성한 이미지

Large-scale results

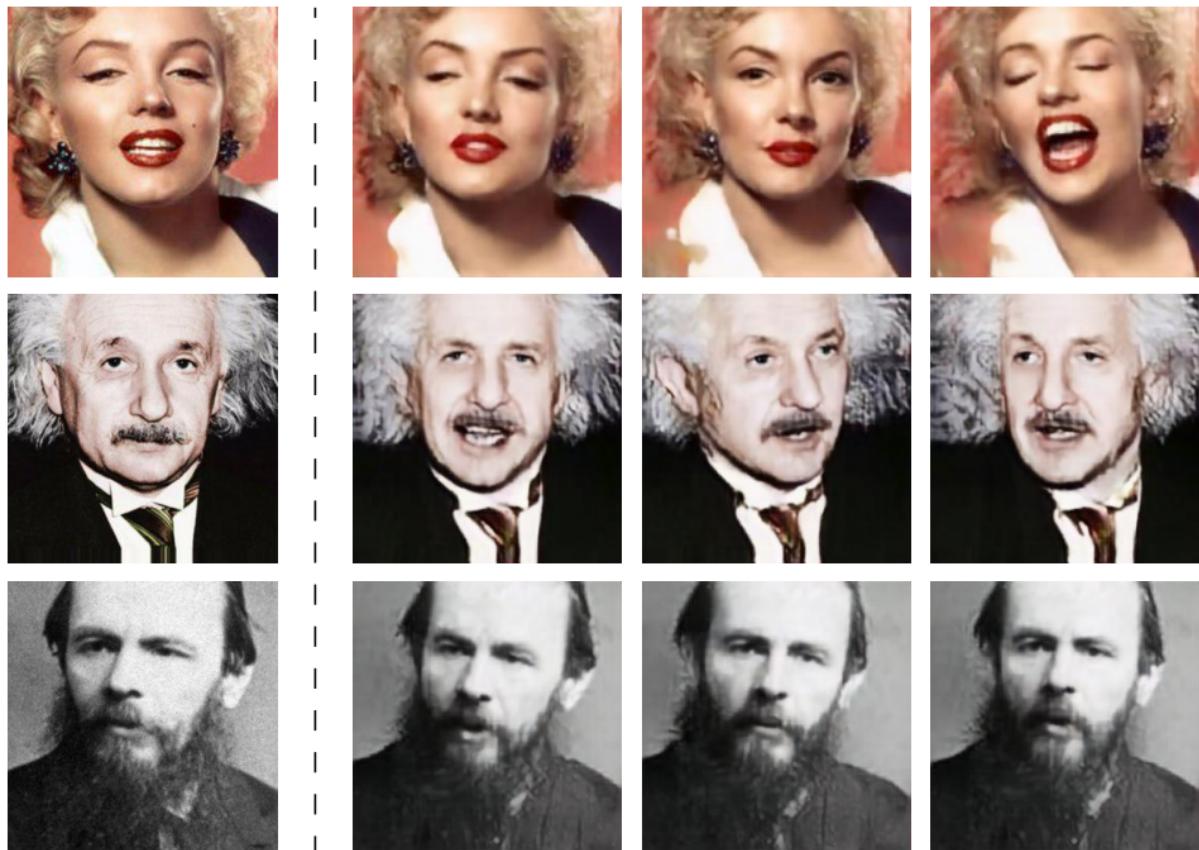
Method (T)	FID↓	SSIM↑	CSIM↑	USER↓
VoxCeleb2				
Ours-FF (1)	46.1	0.61	0.42	0.43
Ours-FT (1)	48.5	0.64	0.35	0.46
Ours-FF (8)	42.2	0.64	0.47	0.40
Ours-FT (8)	42.2	0.68	0.42	0.39
Ours-FF (32)	40.4	0.65	0.48	0.38
Ours-FT (32)	30.6	0.72	0.45	0.33

- FF: L_{MCH} 없이 메타학습을 진행하고, fine-tuning 없이 사용한 모델

- FT: 논문에서 언급한 방식대로 학습한 모델
- 학습에 사용된 이미지의 수가 적을수록 fine-tuning이 없어 바로 사용 가능한 FF의 성능이 좋지만, 많아지면 FT가 더 좋아지는 것을 확인할 수 있음
- 32장의 이미지로 학습한 FT는 USER의 최고점 0.33을 달성하였음



Puppeteering results



Source

Generated images

One-shot Model을 이용해 생성한 이미지들

5. Conclusion

- 메타학습과 GAN을 이용해 성능이 좋은 구조를 제안
- Landmark들은 어느 방향을 쳐다보는지를 표현하지 않아 시선처리 등에 한계가 존재
- 스타일의 사람과 다른 사람의 landmark를 사용하면 성능이 악화되므로, landmark adaption이 필요