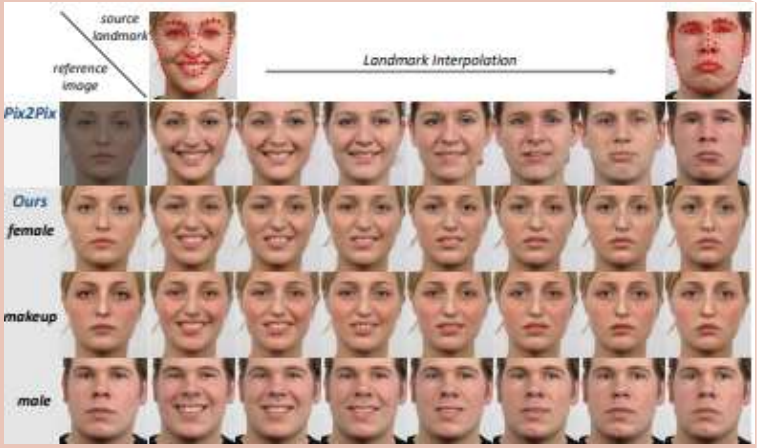
Review of papers about Reenactment

김형범

FReeNet: Multi-Identity Face Reenactment



얼굴 표정을 임의의 소스 얼굴 이미지에서 대상 얼굴로 옮기는 FReeNet이라는 multi-identity face reenactment framework.

기존 GAN 기반의 face reenactment 모델들이 성과를 보였음에도 불구하고 네트워크가 훈련된 후에야 두 개의 특정한 Identity 사이에서 얼굴을 재현할 수 있다는 것을 해결하기 위함.

축처와 대상자 사시에서 얼굴 윤곽의 차이가 존재하기 때문에 통합된 네트워크에 의한 multity identity reenactment를 위한 모델
+ 포즈, 색조, 조명에 일관성을 유지하면서 사진적, identity가 일치하는 target 얼굴을 기존 얼굴을 재현하는 얼굴

FReeNet: Multi-Identity Face Reenactment

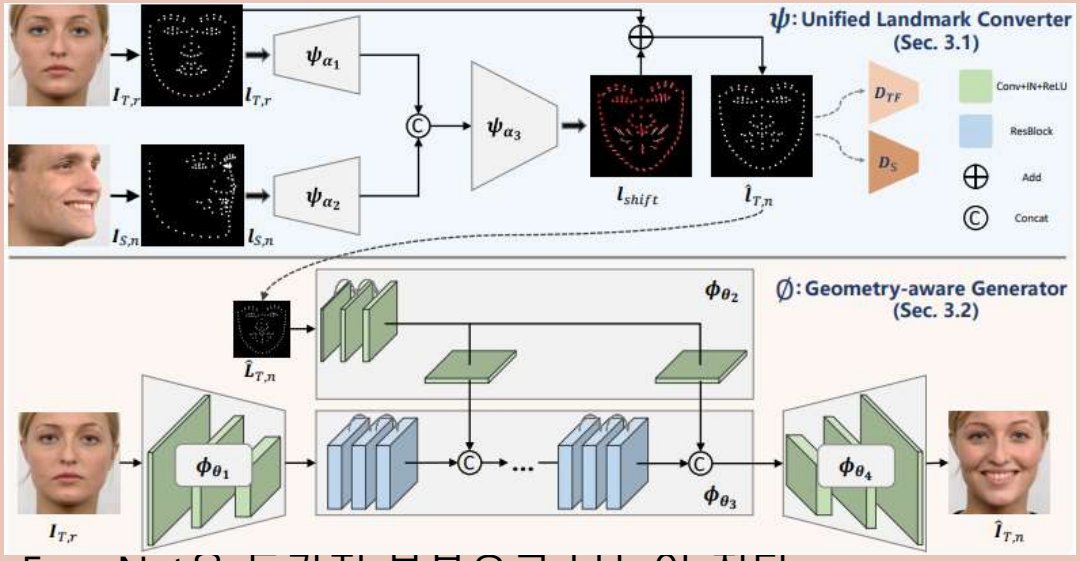
Multi-Identity : 이 모델의 차별점이라고 할 수 있음.

소스 얼굴이 정해진 사람으로부터 온 것이 아니라 임의의 사람으로 부터 온 것이고 target 얼굴이 특정하지 않다.

차별점

- 소스 identity에서 target identity로 표현을 변환하는 통일된 컨버터 -> 소스 identity와 타겟 identity가 모두 여러 사람으로 부터 나온다.
- 디커플링으로 설계되어 별개의 경로에서 appearance 및 기하정보를 추출하는 geometry-aware generator. -> 사진의 사실적 표정을 재현한다.
- 재현된 얼굴의 얼굴 디테일을 풍부하기 위한 novel triplet perceptual loss
- 실험결과 many to many face reenactment task를 고품질로 해결

FReeNet: Multi-Identity Face Reenactment

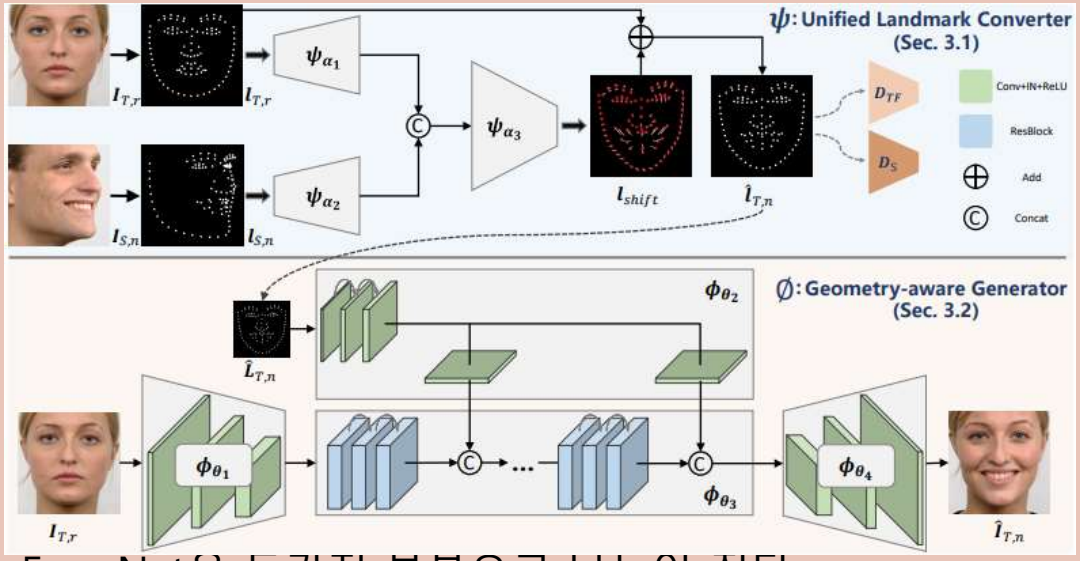


Free Net은 두가지 부분으로 나누어 진다.

Unified landmark converter(ULC)

- 인코더 디코더 구조를 채용
- 잠재된 랜드마크 공간에서 표정을 효율적으로 변환하기 위해 소스와 타겟의 identity 간의 얼굴 윤곽 차이를 좁힌다.
- Dtf는 생성된 결과가 real인지 fake인지 판단
- Ds 는 generated와 supervised간의 랜드마크 유사성을 측정

FReeNet: Multi-Identity Face Reenactment



Free Net은 두가지 부분으로 나누어 진다.

Geometry-aware Generator(GAG)

- 변환된 랜드마크를 활용하여 대상자의 참조 이미지와 함께 사실적 이미지를 재현
- triplet perceptual loss를 이용하여 GAG 모듈이 외관과 기하학적 정보를 동시에 학습하여 재현된 얼굴 이미지의 디테일을 살린다.
- Conv layer와 ResBlock으로 구성되어 있다.

FReeNet: Multi-Identity Face Reenactment

- **ULC**

During the training phase, the overall loss function $\mathcal{L}_{ULC}$ is defined as:

$$\mathcal{L}_{ULC} = \lambda_1 \mathcal{L}_{L1} + \lambda_2 \mathcal{L}_{cyc} + \lambda_3 \mathcal{L}_D, \quad (2)$$

Point – wise L1 Loss.

Point-wise L1 Loss. The first term $\mathcal{L}_{L1}$ is defined by the point level $l1$ loss function to calculate errors of the landmark coordinates:

$$\mathcal{L}_{L1} = ||\hat{\boldsymbol{l}}_{T,n} - \boldsymbol{l}_{T,n}||_1. \quad (3)$$

랜드마크 좌표의 오차를 계산하기 위함

Cycle Consistent Loss

기존 모델들의 cycle loss

$$\mathcal{L}_{D_{TF}} = \mathbb{E}_{x \sim p_{data}(x)}[\log(D_{TF}(x))] + \mathbb{E}_{z \sim p_{data}(z)}[\log(1 - D_{TF}(\boldsymbol{\psi}(z)))], \quad (5)$$

$$\mathcal{L}_{D_S} = \mathbb{E}_{x_1, x_2 \sim p_{data}(x)}[\log(D_S(x_1, x_2))] + \mathbb{E}_{z \sim p_{data}(z), x_1 \sim p_{data}(x)}[\log(1 - D_S(x_1, \boldsymbol{\psi}(z)))], \quad (6)$$

FReeNet: Multi-Identity Face Reenactment

- **ULC**

During the training phase, the overall loss function $\mathcal{L}_{ULC}$ is defined as:

$$\mathcal{L}_{ULC} = \lambda_1 \mathcal{L}_{L1} + \lambda_2 \mathcal{L}_{cyc} + \lambda_3 \mathcal{L}_D, \quad (2)$$

Adversarial Loss

$$\mathcal{L}_{D_{TF}} = \mathbb{E}_{x \sim p_{data}(x)}[\log(D_{TF}(x))] + \mathbb{E}_{z \sim p_{data}(z)}[\log(1 - D_{TF}(\psi(z)))], \quad (5)$$

$$\mathcal{L}_{D_S} = \mathbb{E}_{x_1, x_2 \sim p_{data}(x)}[\log(D_S(x_1, x_2))] + \mathbb{E}_{z \sim p_{data}(z), x_1 \sim p_{data}(x)}[\log(1 - D_S(x_1, \psi(z)))], \quad (6)$$

- 두개의 discriminator가 존재. 각각의 loss를 따로 구함
- ULC module을 하나의 generator로 취급

Dtf는 생성된 landmark가 가짜인지 실제인지 판별하는 것.

Ds는 landmark pair의 identity 윤곽 유사도를 측정하는 것.

따라서 landmark가 실제 랜드마크처럼 정교하게 생성되고 identity 사이의 얼굴 윤곽차이를 줄이기 위해 사용한다.

FReeNet: Multi-Identity Face Reenactment

- **GAG**

- Pixel-wise L1 Loss.

Pixel-wise L1 Loss. The first term $\mathcal{L}_{pix}$ calculates $l1$ errors between generated and supervised images:

$$\mathcal{L}_{pix} = \|\hat{\boldsymbol{I}}_{T,n} - \boldsymbol{I}_{T,n}\|_1. \quad (9)$$

Generated 이미지와 supervised 이미지 사이의 에러를 계산

Adversarial Loss. The second term $\mathcal{L}_{adv}$ introduces the discriminator to improve the realism of the generated images in an adversarial idea:

$$\begin{aligned}\mathcal{L}_{adv} =& \mathbb{E}_{x\sim p_{data}(x)}[\log(D(x))]+ \\ & \mathbb{E}_{k\sim p_{data}(k),l\sim p_{data}(l)}[\log(1-D(\boldsymbol{\phi}(k,l)))],\end{aligned} \quad (10)$$

where x indicates real image data space, and k and l represent input image and landmark space respectively of $\boldsymbol{\psi}$. Discriminator D is similar to the work[43].

FReeNet: Multi-Identity Face Reenactment

- **GAG**

- Adversarial Loss

Adversarial Loss. The second term $\mathcal{L}_{adv}$ introduces the discriminator to improve the realism of the generated images in an adversarial idea:

$$\mathcal{L}_{adv} = \mathbb{E}_{x \sim p_{data}(x)}[\log(D(x))] + \mathbb{E}_{k \sim p_{data}(k), l \sim p_{data}(l)}[\log(1 - D(\phi(k, l)))], \quad (10)$$

where x indicates real image data space, and k and l represent input image and landmark space respectively of ψ. Discriminator D is similar to the work[43].

구조 그림에는 나오지 않았지만 discriminator를 채용하여 adversarial loss를 구함

X는 실제 이미지 데이터를 말하고 K는 input image, l은 landmark space를 말함.

Input 이미지 k와 생성된 landmark l을 이용하여 생성된 이미지 φ(k, l)의 차이를 줄여 realistic한 reenactment를 가능하게 하기 위함.

FReeNet: Multi-Identity Face Reenactment

- Triplet Perceptual Loss

RGB이미지와 랜드마크 이미지 사이의 다른 분포로 인해 발생하는 GAG 모듈의 문제점을 개선하기 위함.

Triplet loss와 perceptual loss을 결합하여 class 간의 perceptual variation을 최대화 하고 class내의 perceptual variation을 최소화한다.

$$\mathcal{L}_{TP}(\hat{\boldsymbol{I}}_{T,n_2}, \hat{\boldsymbol{I}}_{T,n_3}, \hat{\boldsymbol{I}}_{R,n_2}) = \Big[m + D\left(\kappa(\hat{\boldsymbol{I}}_{T,n_2}), \kappa(\hat{\boldsymbol{I}}_{T,n_3})\right) - D\left(\kappa(\hat{\boldsymbol{I}}_{T,n_2}), \kappa(\hat{\boldsymbol{I}}_{R,n_2})\right) \Big]_+, \tag{11}$$

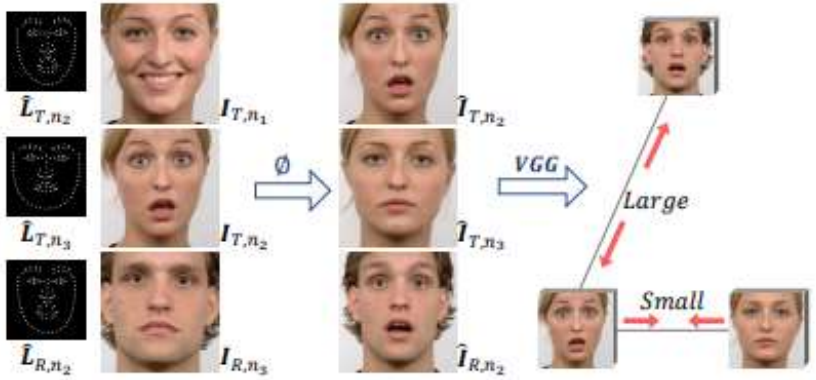


Figure 3. Schematic diagram of the triplet perceptual loss. Simultaneously maximizing inter-class perceptual variation of reenacted images ($\hat{\boldsymbol{I}}_{T,n_2}$ and $\hat{\boldsymbol{I}}_{R,n_2}$) and minimizing intra-class perceptual variation ($\hat{\boldsymbol{I}}_{T,n_2}$ and $\hat{\boldsymbol{I}}_{T,n_3}$).

FReeNet: Multi-Identity Face Reenactment

코드 있음 https://github.com/zhangzjn/FReeNet

Dataset : RaFD, multi-PIE

Learning Identity-Invariant Motion Representations for Cross-ID Face Reenactment

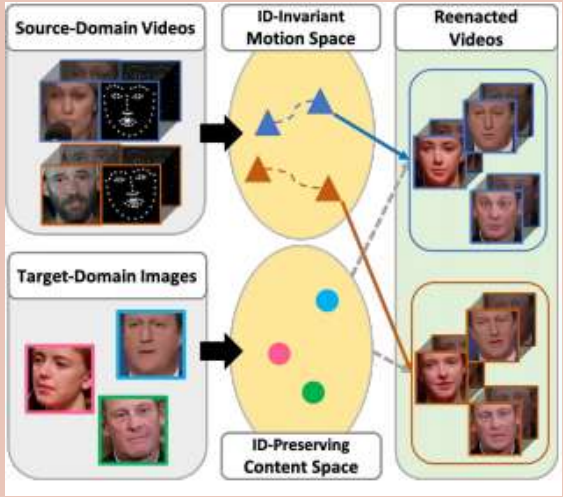
기존의 face reenactment 모델들은 통일된 모델에서는 여러 identity를 다룰 수 없다.

Supervised 하고 unsupervised를 모두 견고하게 모델을 훈련시키기 위함.

Face reenactment를 위한 기존 접근 방식은 미리 정의된 parametric 3D 모델을 정의하여 얼굴을 나타내는데 인간의 머리 움직임은 충분히 반영되지 못한다.

→ GAN 모델이 대안으로 나왔지만 두 얼굴 identity 사이의 일대일 매핑만 지원

통일된 모델과 함께 여러 identity에 걸쳐 얼굴 reenactment 즉 many to many 매핑을 위해 얼굴 랜드마크에서 ID-invariant 표정을 학습하는 CrossID-GAN



Learning Identity-Invariant Motion Representations for Cross-ID Face Reenactment

Identity-invariant : identity를 유지하며 face reenactment를 한다는 의미

Cross-ID : 통일한 모델로 여러 identity를 다룰 수 있다는 의미. 따라서 many-to-many face reenactment가 가능하다.

Learning Identity-Invariant Motion Representations for Cross-ID Face Reenactment

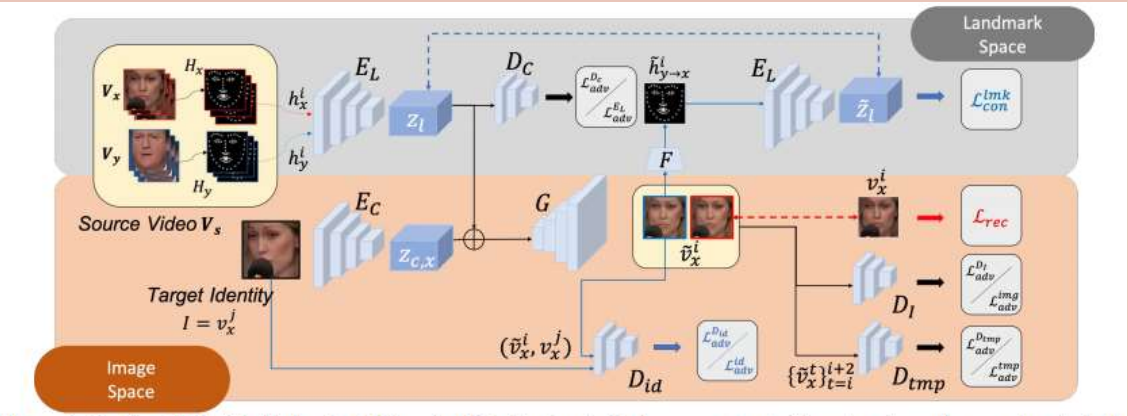


Figure 2. Our CrossID-GAN with learning ID-invariant facial landmarks for face reenactment. Since the observed source-domain landmarks can be from the same or distinct identity as that in the target domain during training, our model can be trained in supervised or unsupervised settings. (Note that arrows in red/blue indicate supervised/unsupervised training processes.)

Source domain video인 Vs(T프레임) 과 해당하는 얼굴 랜드마크 Hs를 입력으로 주고 x identity의 이미지 I를 준다. 결과로는 Vs처럼 표정을 짓는 Vx 비디오를 만들어낸다.

ID-invariant facil landmark encoder(EL) : 측정된 랜드마크를 ID-invariant motion latent code로 인코딩하기위함.

ID-preserving content encoder(EC) : 입력과 identity latent code를 매핑

Conditional generator G : 이미지를 생성

Learning Identity-Invariant Motion Representations for Cross-ID Face Reenactment

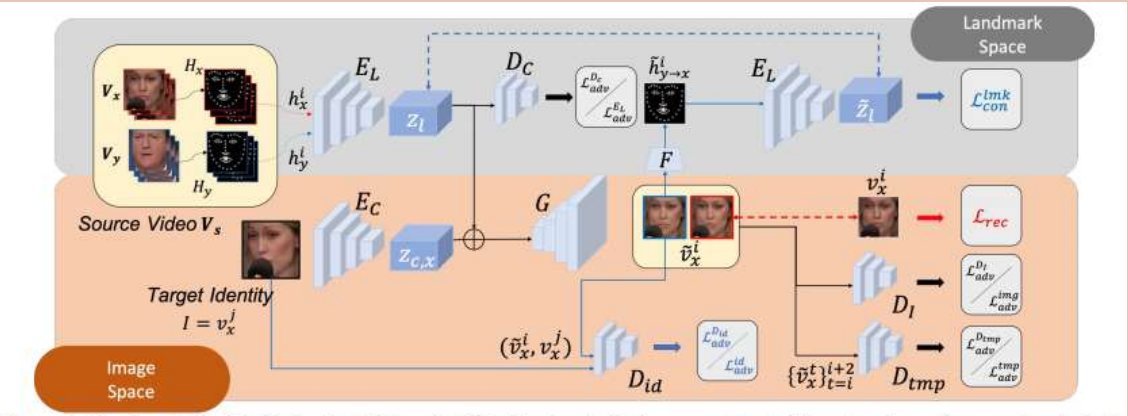


Figure 2. Our CrossID-GAN with learning ID-invariant facial landmarks for face reenactment. Since the observed source-domain landmarks can be from the same or distinct identity as that in the target domain during training, our model can be trained in supervised or unsupervised settings. (Note that arrows in red/blue indicate supervised/unsupervised training processes.)

supervised learning시에 사용하는 Discriminator

D_I는 image discriminator로 생성된 이미지와 실제 이미지의 차이를 구하는 일반적인 discriminator

Learning Identity-Invariant Motion Representations for Cross-ID Face Reenactment

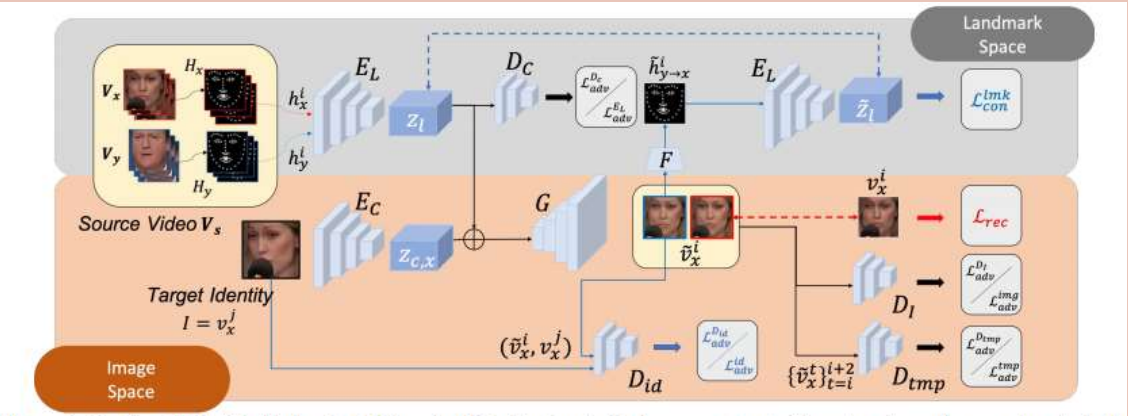


Figure 2. Our CrossID-GAN with learning ID-invariant facial landmarks for face reenactment. Since the observed source-domain landmarks can be from the same or distinct identity as that in the target domain during training, our model can be trained in supervised or unsupervised settings. (Note that arrows in red/blue indicate supervised/unsupervised training processes.)

unsupervised learning시에 사용하는 Discriminator

Dc는 content/id classifier이다

Content와 identity를 분리해주어 생성되는 이미지의 identity invariance를 유지하기 위함이다.

Dtmp는 video discriminator이다

3d convolutional layer로 구성되고 source-domain video의 sequence와 모델의 결과물로 나온 fake의 sequence의 차이를 측정

생성된 video의 품질을 높여주기 위함이다.

Learning Identity-Invariant Motion Representations for Cross-ID Face Reenactment

Supervised learning일 때

$$\mathcal{L}_{rec} = ||\tilde{v}_x^i - v_x^i||^2 + \sum_f |\Phi_{VGG}^f(\tilde{v}_x^i) - \Phi_{VGG}^f(v_x^i)|.$$

Reconstruction Loss로써 L2-norm loss와 perceptual loss에 의해 계산된다.

v^i_x는 identity가 x인 이미지이고 v^~i_x는 identity가 x인 상태로 reenactment를 진행한 이미지이다.

Perceptual loss는 생성된 이미지가 identity가 같은 원래의 이미지와 더 비슷하게 만든다.

Visual feature을 뽑기 위해 VGG-19를 적용한다.

$$\mathcal{L}_{adv}^{D_I} = \mathbb{E}[\log(D_I(\tilde{v}_x^i))] - \mathbb{E}[\log(1 - D_I(v_x^i)],$$
$$\mathcal{L}_{adv}^{img} = -\mathbb{E}[\log(D_I(\tilde{v}_x^i))].$$

Synthesized image와 real image를 구분하기 위한 image discriminator Di를 사용

Vix는 real image Vi-x는 생성된 이미지.

Learning Identity-Invariant Motion Representations for Cross-ID Face Reenactment

Unsupervised learning일 때

$$\mathcal{L}_{adv}^{D_C} = \mathbb{E}[\log P(l_r = r_s | E_L(h_s^i))]$$
$$\mathcal{L}_{adv}^{E_L} = -\mathbb{E}[\log P(l_r = r_s | E_L(h_s^i))],$$

Content/ID classifier인 DC와 back propagation을 통해 EL과 공동으로 훈련하여 identity의 invariance를 보장한다. E_L은 landmark latent code를 추출하는 encoder

P는 도메인 Ir의 확률분포이고 rs는 identity representation의 one-hot vector이다.

이 loss function을 사용해여 EI은 h_s^i(landmark latent code)로부터 identity를 인코딩하도록 강제

$$\mathcal{L}_{con}^{lmk} = |z_l^i - \tilde{z}_l^i|.$$

입력 이미지의 identity latent vector와 생성된 이미지의 latent vector의 차이를 최소화 하여 identity-invariant를 위한 consistency loss.

Learning Identity-Invariant Motion Representations for Cross-ID Face Reenactment

Unsupervised learning일 때

$$\mathcal{L}_{adv}^{tmp} = -\mathbb{E}[\log(D_{tmp}(\{\tilde{v}_x^t\}_{t=i}^{i+2}))],$$

$$\mathcal{L}_{adv}^{D_{tmp}} = \mathbb{E}[\log(D_{tmp}(\{\tilde{v}_x^t\}_{t=i}^{i+2}))] - \mathbb{E}[\log(1 - D_{tmp}(\{v_x^t\}_{t=n}^{n+2}))].$$

Source-domain video의 sequence와 모델의 결과물로 나온 fake의 sequence의 차이를 측정

v^~t_x는 생성된 fake, v^t는 souce-domain video의 sequence.

Learning Identity-Invariant Motion Representations for Cross-ID Face Reenactment

Dataset : 300VW

코드 없음

Neural Head Reenactment with Latent Pose Descriptors

기존 neural one-shot head reenactment를 개선하기 위함

심층 네트워크를 이용하여 대상 사람의 단일 이미지로 부터 head reenactment가 가능하게 해준다.

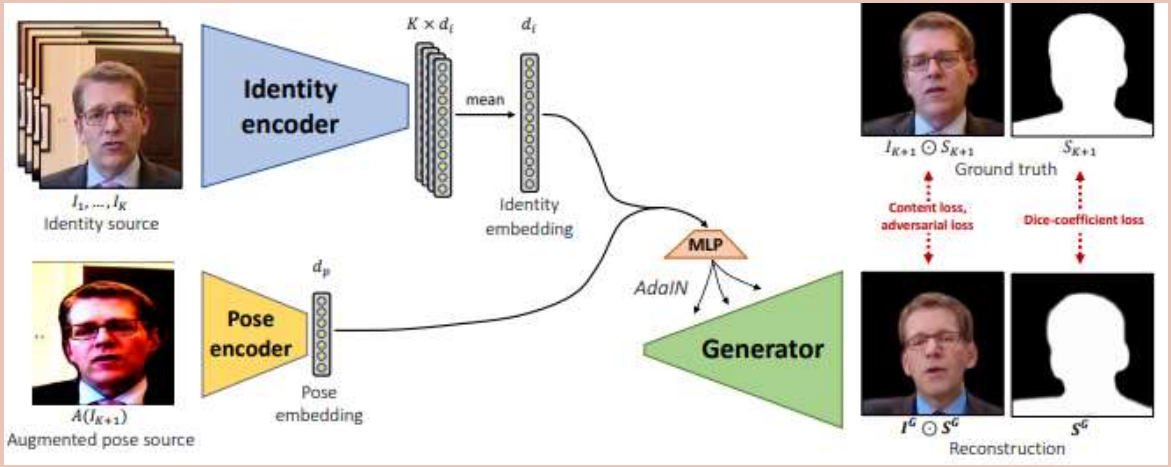
Figure 1: **Being a Mona Lisa.** Our system can generate realistic reenactments of arbitrary talking heads (such as Mona Lisa) using arbitrary people as pose drivers (top row). Despite learning in an unsupervised setting, the method can successfully decompose pose and identity, so that the identity of the reenacted person is preserved.

Neural Head Reenactment with Latent Pose Descriptors

Latent Pose : latent pose representation에 의해 작동하며 foreground segmentation을 RGB 영상과 함께 예측할 수 있다.

Foreground segmentation을 예측할 수 있는 능력으로 이전의 reenactment 기술들을 개선한다.
이러한 예측은 다양한 시나리오의 input에서 reenactment를 잘 수행할 수 있게 해준다.
큰 비디오 데이터 세트와 짝을 이룬 동일한 비디오에서 다수의 프레임을 샘플링하는 두 decriptor를 통해 reenactment 작업을 효과적으로 할 수 있다.

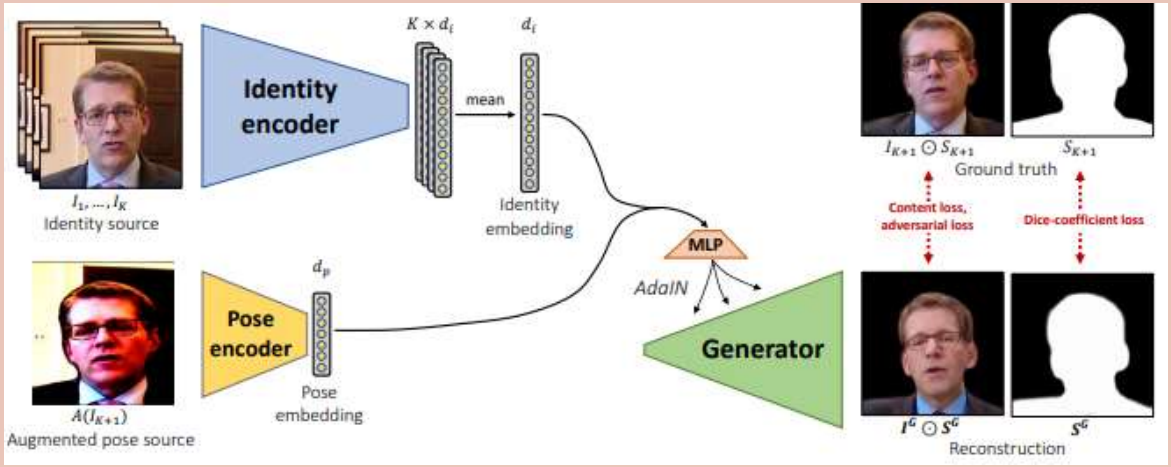
Neural Head Reenactment with Latent Pose Descriptors



기존의 Zakharov의 reenactment 모델을 확장하고 개선했다.

1. Segmentation을 예측하는 능력이 추가되었다.
2. Keypoints(landmark)로부터 학습하는 기존모델과 다르게 latent pose vector로 부터 학습된다.

Neural Head Reenactment with Latent Pose Descriptors



두개의 인코더.

Identity encoder는 비디오의 프레임에 적용되고, 포즈 encoder는 holdout(마지막) frame에 적용

획득한 embedding은 generator로 전달되며 이 embedding의 목표는 마지막 frame을 재구성 하는 것.

ID encoder를 통해 모든 포즈의 독립적인 정보를 추출하고 pose encoder을 통해 취하고 싶은 포즈의 latent pose vector을 추출한다.

Neural Head Reenactment with Latent Pose Descriptors

Segmentation을 위한 dice coefficient loss(구글링)

$$D = \frac{2\sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2}$$

Segmentation시에 많이 사용하는 loss

Pi는 prediction의 pixel values, gi는 ground의 pixel values (여기서는 전경과 배경의 segmentation을 위한 것으로 값으로 0,1 사용)

최대화 하는 것이 목적

Neural Head Reenactment with Latent Pose Descriptors

VGGFace model의 content loss 사용(reference 논문에서 가져옴)

$$\mathcal{L}(\phi,\psi,\mathbf{P},\theta,\mathbf{W},\mathbf{w}_0,b) = \mathcal{L}_{\text{CNT}}(\phi,\psi,\mathbf{P}) + \mathcal{L}_{\text{ADV}}(\phi,\psi,\mathbf{P},\theta,\mathbf{W},\mathbf{w}_0,b) + \mathcal{L}_{\text{MCH}}(\phi,\mathbf{W}).$$

Lcnt는 ground truth 이미지와 reconstruction 이미지의 거리를 계산

Lcnt에 대한 수식은 안 나와있음

Neural Head Reenactment with Latent Pose Descriptors

→ 이 loss function은 다른 모델과의 성능 비교를 위한 것

Adverarial training과 cycle-consistency를 위한 loss.

$$\mathcal{I}_T = \frac{1}{30 \cdot 29 \cdot 32} \sum_{k=1}^{30} \sum_{\substack{i=1 \\ i \neq k}}^{30} \sum_{j=1}^{32} \left[1 - \text{csim}\left(R\left(T_k(I_j^i)\right), r_k\right)\right]$$

원래의 이미지의 사람과 reconstruct된 이미지의 사람이 얼마나 같은 사람으로 추정되는지를 위한 identity loss L_t

$$\mathcal{P}_T = \frac{1}{30 \cdot 32} \sum_{k=1}^{30} \sum_{j=33}^{64} d_{\text{landmarks}}\left(L\left(I_j^k\right), L\left(T_k(I_j^k)\right)\right).$$

Pose reconstruction error는 자세와 얼굴 표정을 얼마나 잘 재현하는지 측정하기 위함.

이는 face landmark 관점으로 정의된다.

T : 모델, I : input, R : identity vector을 출력하는 network

Csim : cosine similarity

학습시에는 오직 image reconstruction loss만 사용한다고 한다.

Neural Head Reenactment with Latent Pose Descriptors

Dataset : VoxCeleb2 dataset

Code : https://github.com/jkvt2/Latent-Pose-Descriptors

MarioNETte: Few-shot Face Reenactment Preserving Identity of Unseen Targets

기존의 기술로는 target identity와 driver identity가 일치하지 않을 경우 face reenactment는 결과의 품질에 심각한 저하가 생긴다. 모델이 대상의 디테일을 잃어버리기 때문

위 문제를 해결하기 위한 image attention block, target feature alignment, landmark transformation으로 구성된 모델

target얼굴에 driver 표정을 넣자

MarioNETte: Few-shot Face Reenactment Preserving Identity of Unseen Targets

Few shot : 제안된 아키텍쳐는 관련된 feature를 warping하고 attending함으로써 적은 shot으로부터 보이지 않는 identity를 고품질로 재현
Few shot 환경에서 작동하도록 linear base에 대한 coefficient를 regress 하기 위한 신경망을 도입하고 각각의 module을 제작
Preserving identity of Unseen target :

랜드마크 transformer는 랜드마크 disentanglement를 통해 expression geometry를 분리함으로써 identity 보존 문제를 해결한다.
Unsupervised한 방법으로 identity mismatch를 조정함으로써 identity를 잘 보존해준다

Driving face의 특징과 target face의 특징이 크게 다른 상황에서도 target의 identity를 보존해준다. 모델이 target feature map의 관련 위치에 attend할 수 있는 attention block을 이용한다. Feature level의 warping operation이 포함된 target feature alignment를 함께 사용하여 face reenactment의 품질을 높일 수 있다.

MarioNETte: Few-shot Face Reenactment Preserving Identity of Unseen Targets

Paired target과 서로 다른 identity의 driver 이미지를 얻을 수 없으므로 명시적인 주석 없이 target과 같은 비디오에서 추출한 driver 이미지를 사용하여 훈련

MarioNETte: Few-shot Face Reenactment Preserving Identity of Unseen Targets

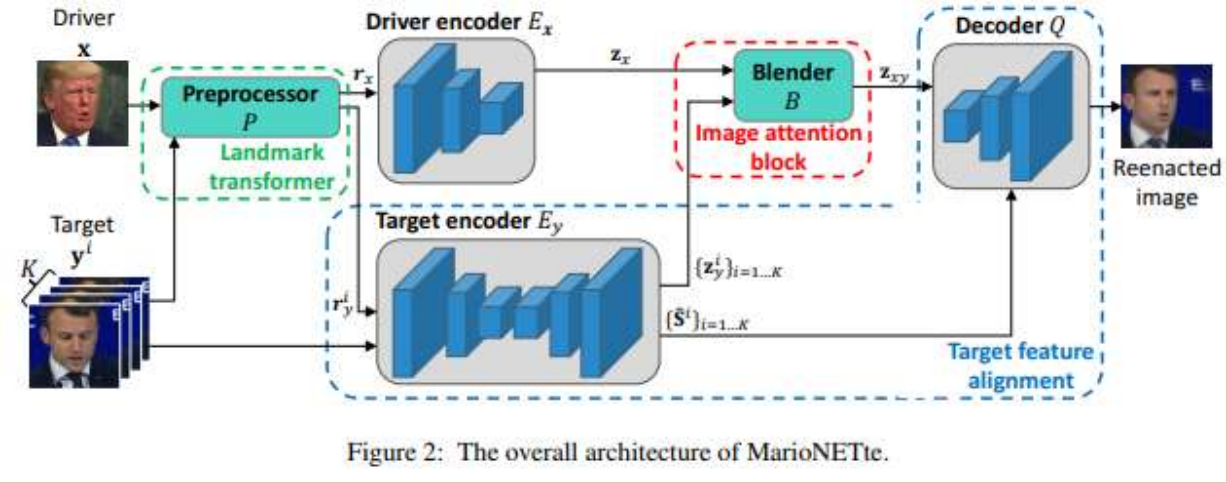


Figure 2: The overall architecture of MarioNETte.

Proprocessos P는 3d landmark detector를 사용하여 facial keypoint를 추출하고 landmark 이미지로 render한다. 위에서 제안한 transformer는 이곳에 포함된다.

Driver encoder Ex(rx)는 driver input으로부터 포즈와 표정 정보를 추출하고 driver feature map zx를 생성한다.

Target encoder Ey(y,ry)는 U-Net 구조를 채용하여 target input으로부터 스타일 정보를 추출하고 target feature map zy와 warped target feature map S^를 생성한다.

MarioNETte: Few-shot Face Reenactment Preserving Identity of Unseen Targets

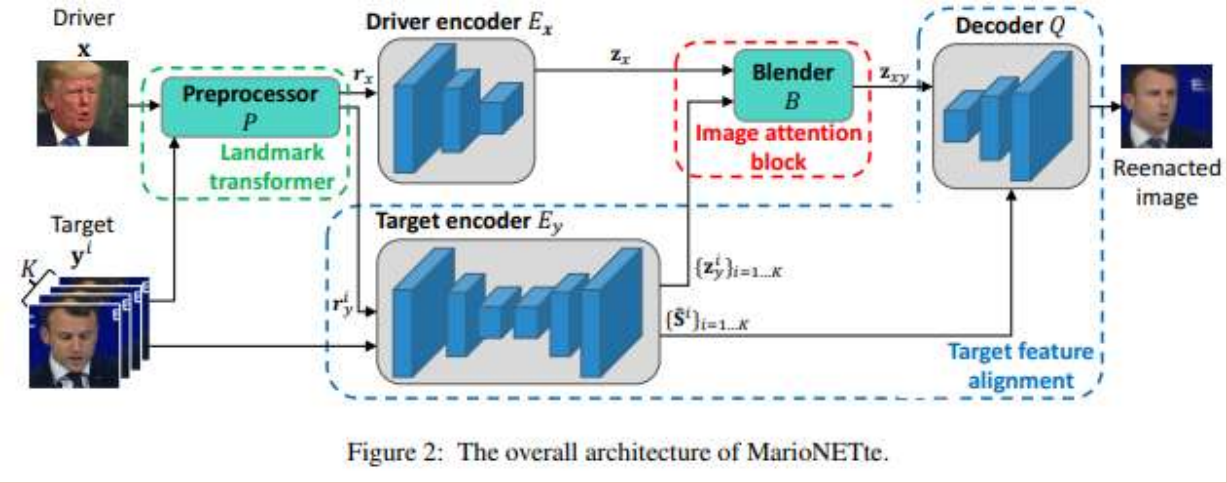


Figure 2: The overall architecture of MarioNETte.

Blender B(zx, {z i y}i=1...K) 는 driver feature map zx와 target feature maps Zy = [z 1 y , . . . , z K y] 를 받아 mixed feature map zxy를 생성한다. 제안된 attention block은 blender의 basic building block이다.

Decoder Q(zxy, {Sˆi}i=1...K)는 warped target feature map S^와 mixed feature map zxy를 사용하여 reenacted image를 생성한다. Decoder는 target feature alignment를 사용하여 reenacted 이미지의 품질을 향상시킨다.

MarioNETte: Few-shot Face Reenactment Preserving Identity of Unseen Targets

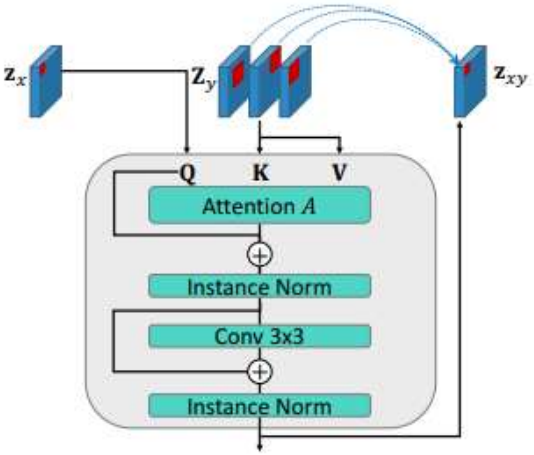


Image attention block

Attention block은 transformer의 encoder–decoder attention에서 영감을 얻어

Driver feature map은 attention query의 역할을 하고 target feature map은 attention memory로 작용한다.

어텐션의 기본 아이디어는 디코더에서 출력을 하는 매 시점마다 인코더에서 전체 입력을 참고한다는 뜻. 단 전부 다 동일한 비율로 참고하는 것이 아니라 해당 시점에서 연관이 있는 부분을 좀 더 집중하여 본다는 것.

MarioNETte: Few-shot Face Reenactment Preserving Identity of Unseen Targets

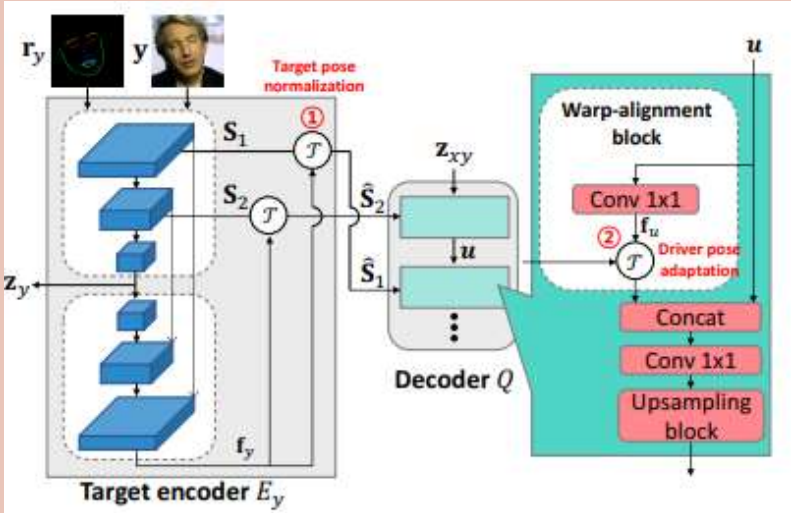


Figure 4: Architecture of target feature alignment.

Target feature alignment

세부적인 target identity의 detail을 보존해주기 위함.

Target pose normalization으로 pose normalized target feature maps을 생성하며 driver pose adaptation은 표준화된 feature map을 driver의 포즈에 맞춰 align한다. 이 과정은 모델이 다른 identity의 구조적인 차이를 더 잘 처리할 수 있게 해준다.

MarioNETte: Few-shot Face Reenactment Preserving Identity of Unseen Targets

$$\begin{aligned} \hat{\mathbf{x}} = &\quad G(\mathbf{r}_x; \{\mathbf{y}^i\}, \{\mathbf{r}_y^i\}) \\ \mathcal{L}_D = &\quad \max(0, 1 - D(\mathbf{x}, \mathbf{r}_x, c)) \quad + \\ &\quad \max(0, 1 + D(\hat{\mathbf{x}}, \mathbf{r}_x, c)). \end{aligned}$$

Hinge loss GAN loss를 사용한다.

Discriminator d를 optimize하기 위함.

Discriminator은 identity c의 실제 이미지와 generator에 의해 생성된 c의 합성 영상을 구별하는 것을 목표

$$\mathcal{L}_G = \mathcal{L}_{GAN} + \lambda_P \mathcal{L}_P + \lambda_{PF} \mathcal{L}_{PF} + \lambda_{FM} \mathcal{L}_{FM}.$$

Perceptual loss와 GAN loss를 합친 종합 loss

Perceptual loss는 ground truth loss와 generated image차이를 구한 것이다.

해결하고자 하는 문제
키워드 및 차별점
모델 구조
Loss function
Dataset, code

MarioNETte: Few-shot Face Reenactment Preserving Identity of Unseen Targets

Dataset : VoxCeleb1, CelebV

소스코드 없음

Realistic Face Reenactment via Self-Supervised Disentangling of Identity and Pose

Manual annotation의 수요를 완화하기 위해 라벨이 부착되지 않은 비디오가 대량으로 주어지면 자연스럽게 얼굴을 재현하는 방법을 학습하는 DAE-GAN

Face reenactment는 통제되지 않는 조건에서 얼굴의 외형은 identity, pose, 표정, reflection 등 몇 가지 결합된 요인에 의해 결정되는데 이는 얼굴 간에 특정한 속성을 전달하기 어렵게 만든다.

이를 해결하기 위한 auto encoder와 GAN을 결합한 모델

Realistic Face Reenactment via Self-Supervised Disentangling of Identity and Pose

Self-supervised : 많은 양의 라벨이 부착되지 않은 동영상을 보며 자연스럽게 talking face를 reenact 하는 self-supervised 체계 훈련 단계에서는 동일한 비디오 시퀀스의 프레임으로만 모델을 최적화한다.

Network로 데이터 자체를 학습하여 pretraining 시키고 downstream task로 transfer learning하는 접근방법 -> 즉 비디오 하나가 하나의 domain이 된다!

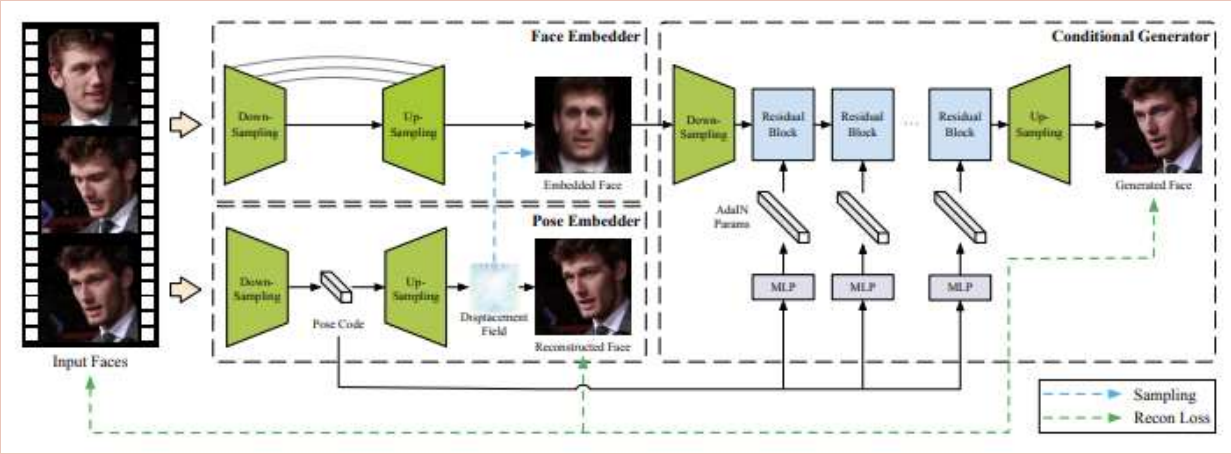
Disentangling of identity and pose : reenactment의 품질을 높이기 위해 identity와 pose를 분리한다. 이는 이미지 검색에 응용할 수 있는 가능성을 나타낸다.

Realistic Face Reenactment via Self-Supervised Disentangling of Identity and Pose

Training 단계에서 모델은 하나의 identity에 대해 최적화 한다. (self-supervised) 그 후 test 단계에서는 face embedder과 pose embedder는 서로 다른 identity input을 받을 수 있다.

→ 따라서 cross-id, multi-id reenactment가 가능하다고 생각합니다!

Realistic Face Reenactment via Self-Supervised Disentangling of Identity and Pose

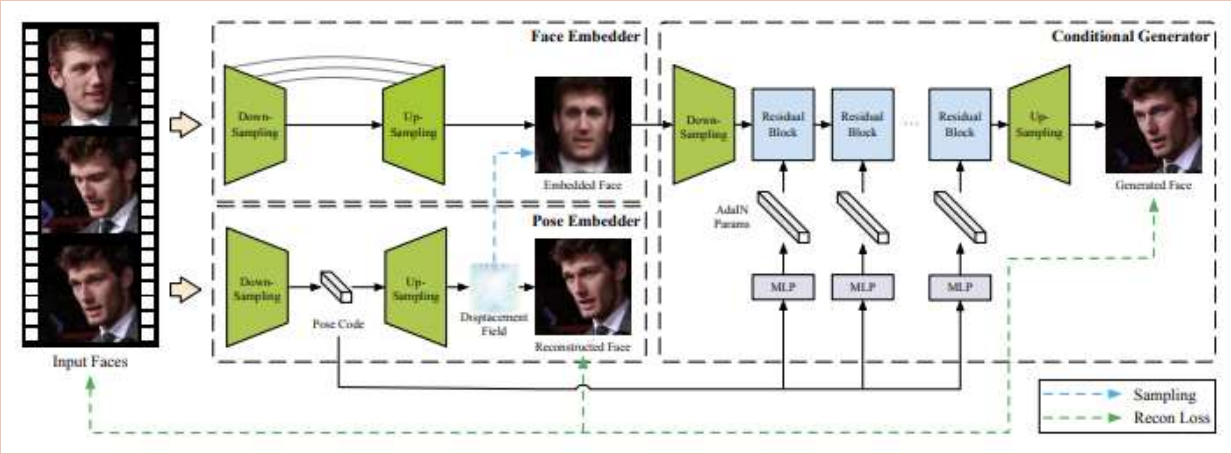


전체 아키텍처에는 identity를 분리하고 feature를 나타내는 두개의 임베더, 한쌍의 generator-discriminator가 재현 얼굴을 합성한다. 입력은 비디오

Face embedder는 하나의 비디오에서 여러 프레임을 가져와서 embedded 페이스에 매핑. Embedded face에는 각 프레임의 포증와 표정에 불변하는 identity가 포함.

Poes embedder는 embedded face와 다른 포즈의 프레임들을 input으로 받는다. Embedded face와 다른 프레임을 reconstruct하도록 설계

Realistic Face Reenactment via Self-Supervised Disentangling of Identity and Pose



Conditional generator은 추출된 포즈 벡터와 embedded face를 이용하여 pose-alike image를 생성한다.

Discriminator은 합성된 영상과 해당 영상 프레임을 입력받아 그들 사이의 세부적인 차이를 구별한다.
현실적 이미지를 생성하기 위해 generator와 adversarial하게 학습한다.
구조 그림에는 나와있지 않지만 마지막 generated영상과 해당 영상 프레임을 입력받아 차이를 구별합니다.

Realistic Face Reenactment via Self-Supervised Disentangling of Identity and Pose

$$\mathcal{L}_{\text{REC}}\left(x_i^t, \hat{x}_i^t\right) = \left\|x_i^t - \hat{x}_i^t\right\|_1.$$

Recostruction loss : real image와 reconstruction 이미지의 차이를 측정.

$$\min_D \max_G \mathcal{L}_{\text{GAN}}(D,G) + \lambda_{\text{R}}\mathcal{L}_{\text{R}}(G) + \lambda_{\text{FM}}\mathcal{L}_{\text{FM}}(G).$$

Generator과 discriminator의 Adversarial한 학습을 위한 loss

Realistic Face Reenactment via Self-Supervised Disentangling of Identity and Pose

Dataset : VoxCeleb1, RaFD

코드 없음

Neural Voice Puppetry: Audio-driven Facial Reenactment

ECCV2020

오디오로 구동되는 얼굴 비디오 reenactment를 위한 Neural Voice pupperty

입력으로 source person의 audiop sequence를 소스 입력의 오디오와 동기화 되는 리얼한 출력 비디오를 생성한다.

Visual digital assitant 의 시나리오에서 사용할 수 있는 photo-realistic facial animation방식의 Neural Voice Puppetry를 도입하여 누락된 visual channel을 제공하는 것.

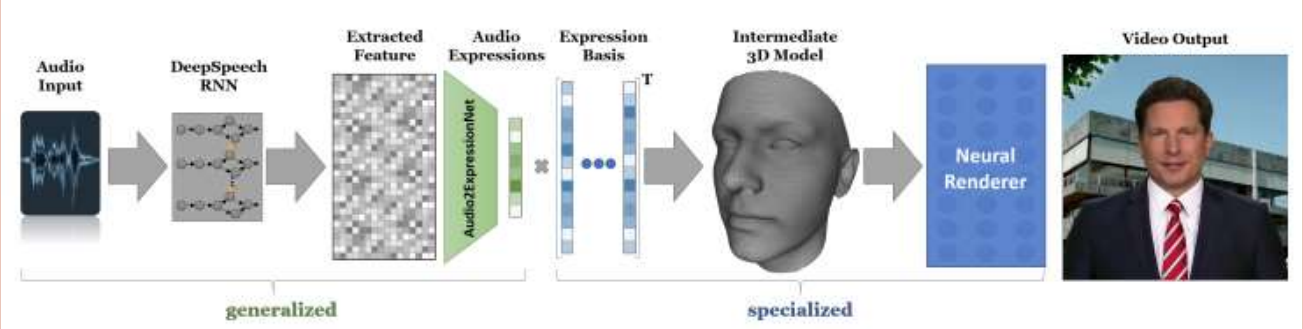
Neural Voice Puppetry: Audio-driven Facial Reenactment

Neural Voice Puppetry : 이 논문의 핵심 기술. 입력 오디오에 맞는 입술동작을 추정하고 대상의 표정을 설득력있게 연출하는 것.
단일 대상 비디오나 manual한 사용자 입력의 방대한 양의 비디오 영상을 필요로 하지 않는 오디오-비디오 변환 도구.

Audio-driven : 대상 사람의 오디오를 입력을 받아 입술동작을 추정하여 표정을 연출

Audio2ExpressionNet이라는 네트워크가 오디오를 사람 특유의 말하기 스타일을 나타낼 수 있는 3D blendshape basis로 매핑

Neural Voice Puppetry: Audio-driven Facial Reenactment

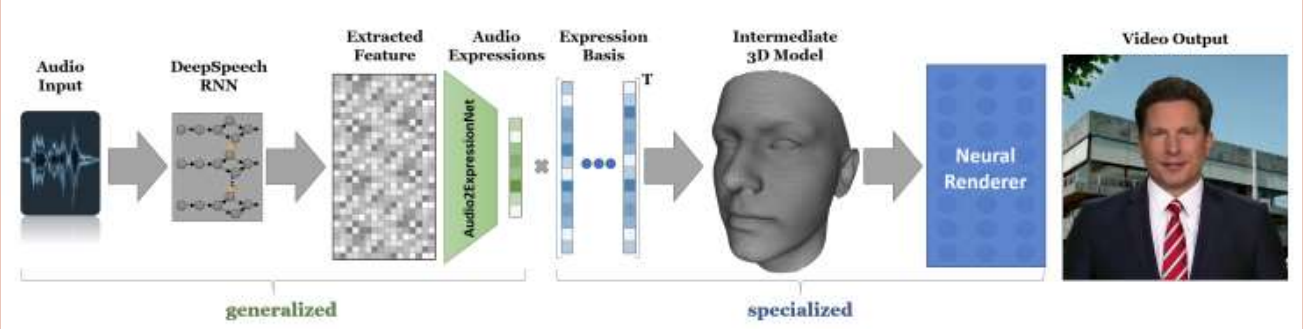


Neural Vocie Puppetry는 2부분으로 나누어진다.

Generalized network : latent expression vector을 예측하여 audio-expression space를 확장한다. 이 audio expression space는 모든 사람이 공유하는 것이며 재현이 가능하다. 즉 예측된 motion을 한 사람에게서 다른 사람에게 전달하는 것.

Audio expresison을 3d face model의 혼합형 계수로 해석하는데 이는 사람마다 다르며 얼굴 표정과 외모 등 대상자의 특성이 들어가 있다. 이를 specialized network로 전달.

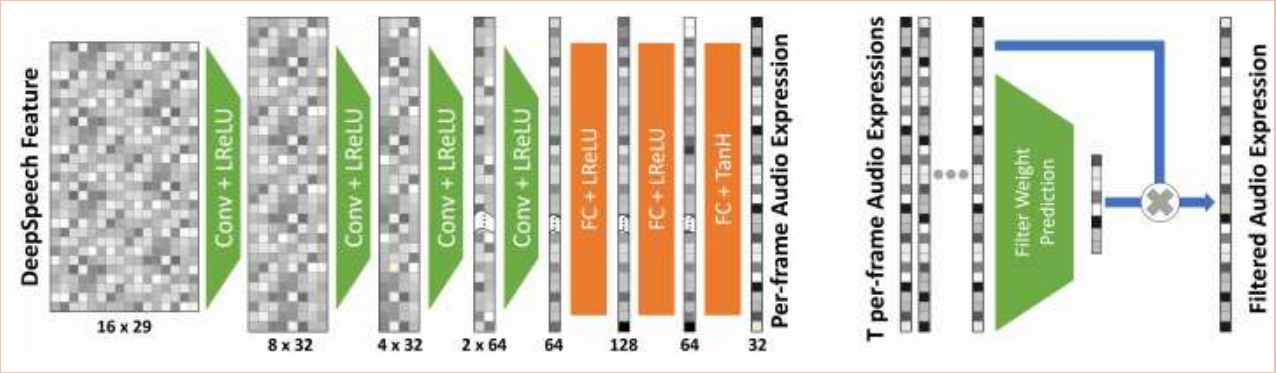
Neural Voice Puppetry: Audio-driven Facial Reenactment



Specialized network : 해석된 audio expressions를 이용해 얼굴의 motion과 외모등 대상자의 특성을 capture

이를 이용해 3d 모델링을 하여 neural renderer을 통해 video output을만들어 낸다.

Neural Voice Puppetry: Audio-driven Facial Reenactment



Audio2ExpressionNet

DeepSpeech feature을 입력으로 받아 프레임 별 audio-expression을 추정

부드러운 audio-expression을 얻기 위해 시간 차원에 따른 content-aware filtering을 사용

Convolutional layer와 FC layer을 결합하여 사용

Neural Voice Puppetry: Audio-driven Facial Reenactment

$$L_{expr} = RMS(v_t - v_t^*) + \lambda \cdot L_{temp}$$

얼굴 예측을 위한 loss로 vertex 기반 loss로 face의 입 부분에 더 높은(x10) 가중치가 있다.

RMS(root mean squared)거리의 관점에서 계산

Vt는 시간 t에 측정된 face vertices이고 vt*는 추정된 face vertices

$$L_{rendering} = \ell_1(I, I^*) + \ell_1(\hat{I}, I^*) + VGG(I, I^*)$$

Neural Face rendering 단계에서 사용하는 Loss.

정확한 에러를 측정하기 위한 L1 loss와VGG style loss 기반이다.

I는 최종 synthtic 이미지를 말하고 I*은 기존 실제 이미지, I^은 첫 네트워크 이후에 나온 중간 결과 이미지이다.

Neural Voice Puppetry: Audio-driven Facial Reenactment

Dataset : Mozilla’s CommonVoice dataset. 자체 dataset

소스코드 : https://github.com/JustusThies/NeuralVoicePuppetry

Face2Face: Real-time Face Capture and Reenactment of RGB Videos

Cvpr 2016 (justus Thies)

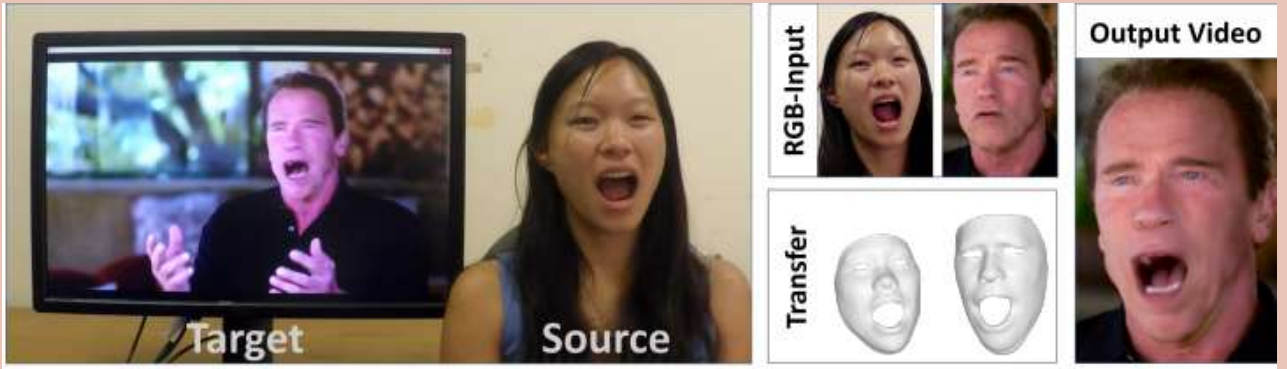
기존방식으로는 실시간 face reenactment가 불가능

웹캠으로 라이브로 캡쳐된 비디오 스트림으로부터 실시간 face reenactment를 위한 접근법

소스 사람의 얼굴표정을 살리고 조작된 출력 영상을 포토 리얼리티 방식으로 다시 렌더링 하는 것이 목표

기존 방식과는 달리 실시간 비디오 스트림으로부터 RGB센서로 얼굴을 캡쳐하고 온라인으로 타겟에 표정을 전달하는 모델

Deep learning이 아니다?



Face2Face: Real-time Face Capture and Reenactment of RGB Videos

Real-time : 실시간으로 face reenactment를 해줌

실시간으로 얼굴 표정을 재현하는 것이 주된 차별점
오프라인으로 재현하는 기존 방식과는 달리 RGB센서에 의해 포착된 소스 얼굴표정을 대상 사람에게 온라인으로 전달하는 것.

Face2Face: Real-time Face Capture and Reenactment of RGB Videos

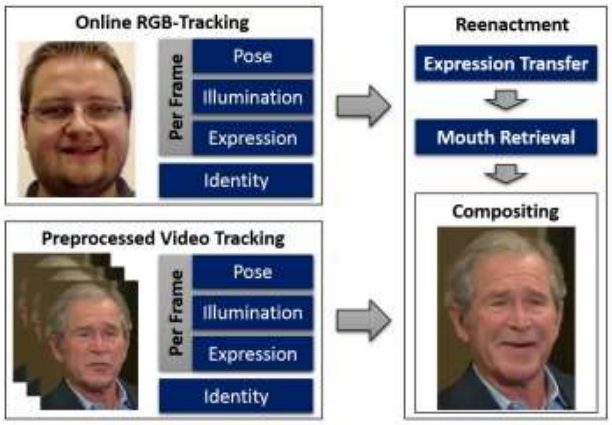


Figure 1: Method overview.

Multi-linear PCA model을 사용한다. 처음의 두 차원은 face identity와 skin reflectance 그리고 나머지 한 차원은 얼굴 표정을 조절한다.

new global non-rigid model을 이용하여 target과 source에서 pose, illumination, expression, identity를 pca model의 형태로 추출한다.

추출한 결과를 이용해 source얼굴의 표정을 옮겨주고 mouth retrieval을 하여 이미지를 합쳐 합성해준다.

Dense analysis-by-synthesis approach : 합성을 하고 렌더링을 한다.

Face2Face: Real-time Face Capture and Reenactment of RGB Videos

$$E(\mathcal{P}) = \underbrace{w_{col}E_{col}(\mathcal{P}) + w_{lan}E_{lan}(\mathcal{P})}_{data} + \underbrace{w_{reg}E_{reg}(\mathcal{P})}_{prior}$$

Photo-Consistency In order to quantify how well the input data is explained by a synthesized image, we measure the photo-metric alignment error on pixel level:

$$E_{\text{col}}(\mathcal{P}) = \frac{1}{|\mathcal{V}|}\sum_{\boldsymbol{p}\in\mathcal{V}} \|C_{\mathcal{S}}(\boldsymbol{p}) - C_{\mathcal{I}}(\boldsymbol{p})\|_2 \ , \qquad (4)$$

Cs가 렌더링한 이미지를 뜻함

P는 이미지로부터 추출된 parameter들을 말함(pose, illumination, identity, expression)

렌더링한 이미지와 원본 이미지의 차이를 구함

Face2Face: Real-time Face Capture and Reenactment of RGB Videos

Feature Alignment: In addition, we enforce feature similarity between a set of salient facial feature point pairs detected in the RGB stream:

$$E_{\text{lan}}(\boldsymbol{\mathcal{P}}) = \frac{1}{|\mathcal{F}|} \sum_{\boldsymbol{f}_j \in \mathcal{F}} w_{\text{conf},j} \left\| \boldsymbol{f}_j - \Pi(\Phi(\boldsymbol{v}_j) \right\|_2^2 .$$

Π(Φ(vj) : **V가 3d모델 space에 있는 vertex 인데 이것을 image space로 매핑 이것을 image space에 위치하는 feature point의 위치와 비교한다.**

Face2Face: Real-time Face Capture and Reenactment of RGB Videos

웹캠으로 라이브로 캡쳐된 비디오를 사용

데모버전? 소스코드 : https://github.com/datitran/face2face-demo

Deep Video Portraits

ToG 2018 (justus Thies)

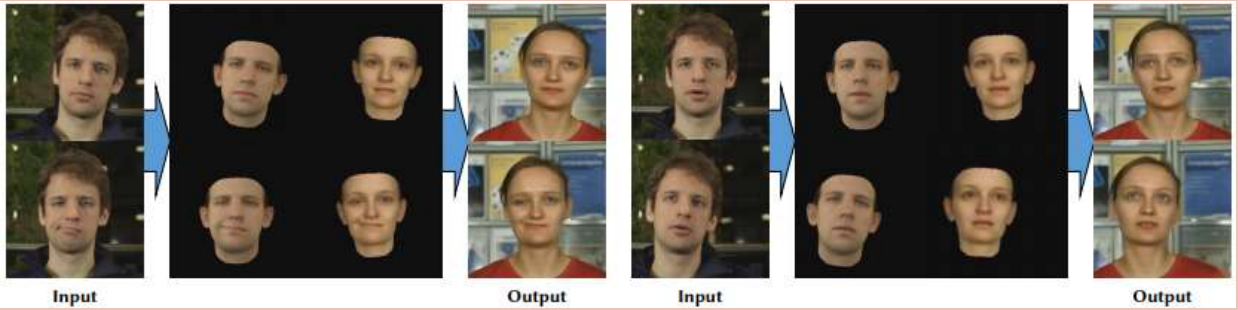
대상의 얼굴 특징을 photorealism 수준으로 전달해 대상을 완벽하게 재현

입력 비디오만 사용하여 portrait 비디오의 photorealism 재현이 가능한 접근법

머리회전, 머리 위치, 얼굴 표정, 시선, 눈 깜박임을 전송

소스 비디오에서 재구성된 헤드 애니메이션 파라미터로 합성한다.

Conditional generative adversarial network(cGAN)을 사용(deep learning이 사용된 부분)



Deep Video Portraits

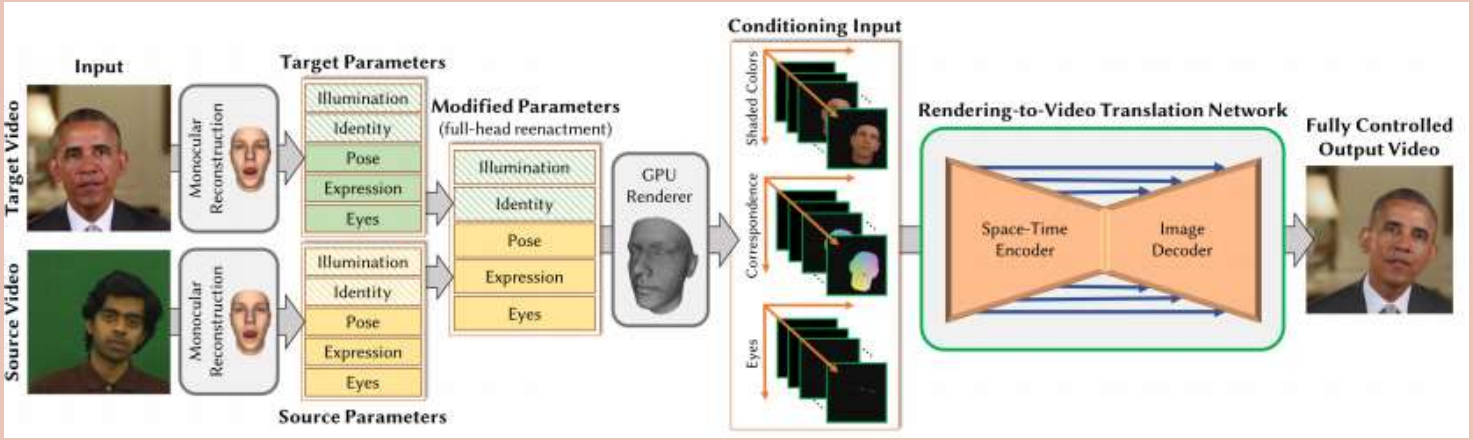
영상 속 대상의 표정만 수정하는 기존의 방식과는 달리 머리 자세와 얼굴 표정, 안구 동작을 높은 품질으로 대상을 완벽하게 재현할 수 있다.

소스와 타겟 파라미터를 자유롭게 결합하여 머리카락, 몸체, 배경들을 명시적으로 모델링 하지 않고도 다양한 영상 재작 가능

머리 자세, 얼굴 표정, 눈 움직임의 완전한 통제 하에 기존 배경 앞에서 전체 사실적 비디오를 생성하는 독특한 접근방식

Monocular face reconstruction : State of art dense face reconstruction을 이용한다. 입력이 들어오면 optimize를 통해 얼굴을 재현할 수 있는 parameter들을 찾는다. 이를 이용해 3d face(mesh)를 만든다.

Deep Video Portraits



먼저 소스 비디오와 타겟 비디오의 low dimensional parametric representation을 monocular face reconstruction을 이용하여 구한다.

머리 자세, 표정 눈 시선은 parameter 공간에서 옮겨 modified parameter를 만든다. Face reenactment가 목표이므로 배경은 수정하지 않는다.

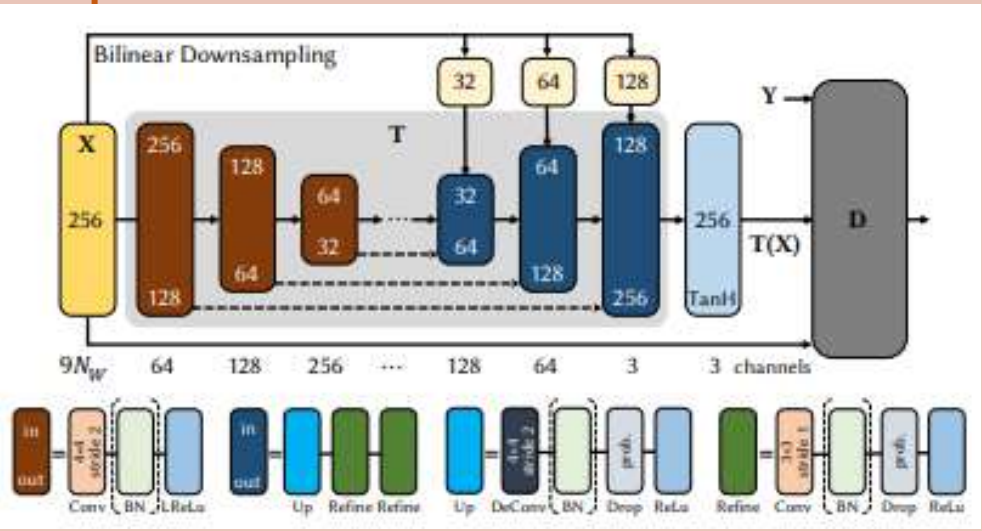
마지막으로 modified parameter을 이용해 auto encoder 구조를 통해 재현 영상을 렌더링한다.

cGAN으로 학습된 Rendering-to-video translation network를 통해 영상 재현

GPU Renderer로 parameter을 이용해 만든 3d face로 얼굴을 재현하는 것이 기존의 기술

이 논문에서는 이 사이에 cGAN을 사용해 품질을 높였다는 것이 차별점

Deep Video Portraits



Rendering-to-video translation network(Auto encoder)의 구조이다.

256*256 입력 해상도를 위한 rendering-to-video 변환 네트워크이다.

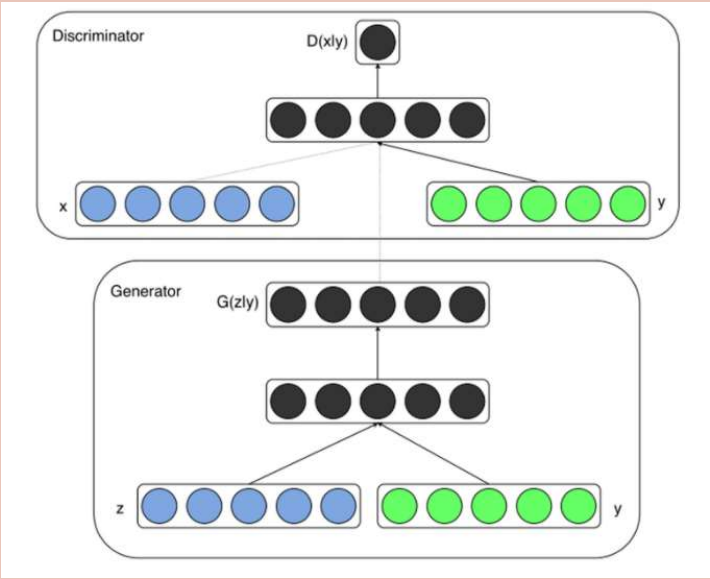
인코더는 출력 채널이 있는 8개의 다운샘플링 모듈로 구성, 디코더는 출력채널이 있는 8개의 업샘플링 모듈로 구성

이 모듈을 훈련할 때 cGAN 이용

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))]$$

Deep Video Portraits

cGAN



GAN에 특정 condition을 나타내는 정보 y를 가해준 형태

이때 condition으로 앞에서 구한 color rendering, correspondence image, eye gaze image를 사용하는 것

Deep Video Portraits

$$T^* = \arg\min_T \max_D E_{cGAN}(T, D) + \lambda E_{\ell_1}(T).$$

Objective function.

최적의 결과를 위해 adversarial한 방법으로 훈련을 한다.

E_cGAN은 adversarial loss(일반적인 Gan loss와 동일)이고 E_l1은 reproduction loss이다.

T는 generator라고 생각하면 됨.

$$E_{\ell_1}(T) = \mathbb{E}_{X,Y}\left[\, \|Y - T(X)\|_1 \,\right].$$

합성된 영상 T(x)와 실제 이미지 Y와의 차이를 측정, 최소화 하기 위함.

Deep Video Portraits

작은 dataset만 사용했다.

저자의 코드는 없다

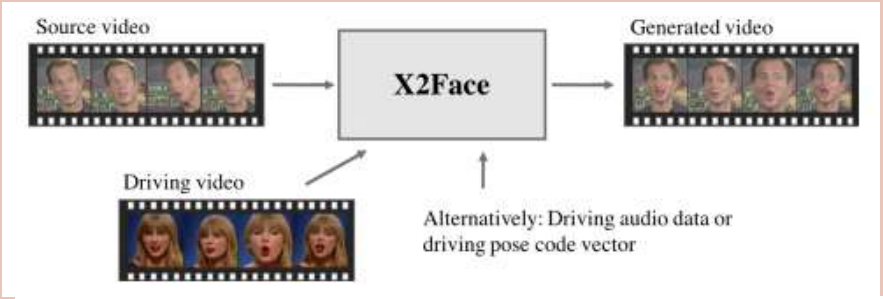
Rebuilding된 코드 : https://github.com/SunMars/DeepVideoPortraits

X2face: A network for controlling face generation by using images, audio, and pose codes

ECCV2018

다른 얼굴이나 양식(오디오 등등)을 사용하여 주어진 얼굴의 자세와 표정을 제어하는 모델

소스 프레임의 identity를 유지하며 타겟의 표정으로 reenactment



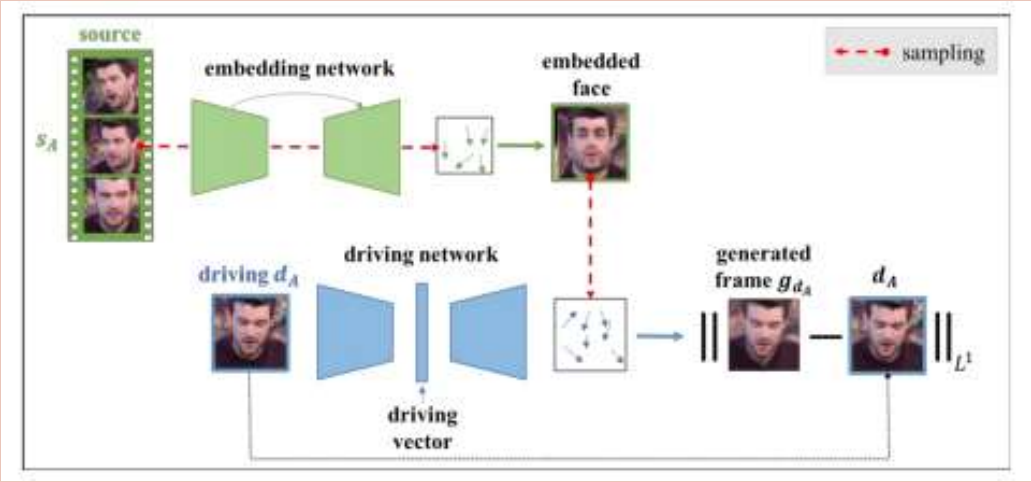
X2face: A network for controlling face generation by using images, audio, and pose codes

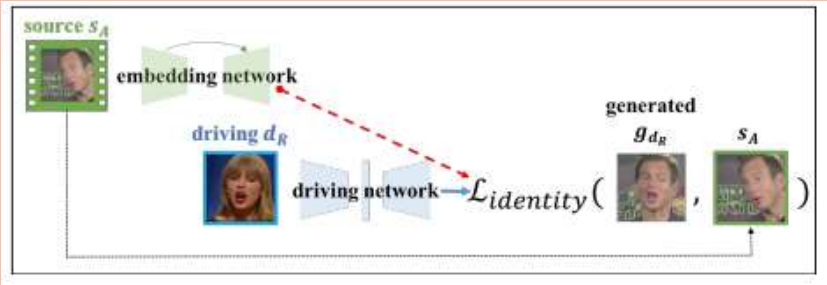
X2face : 소스페이스를 제어하는 네트워크. 소스 프레임의 identity를 유지하며 driving의 페이스와 표정으로 생성된 프레임으로 reenactment

네트워크의 추가 훈련없이 오디오나 포즈 코드와 같은 다른 양식에 의해서도 generation할 수 있다.

많은 비디오 데이터를 사용하여 네트워크가 완전한 self-supervised 훈련 시키는 method를 제안

X2face: A network for controlling face generation by using images, audio, and pose codes





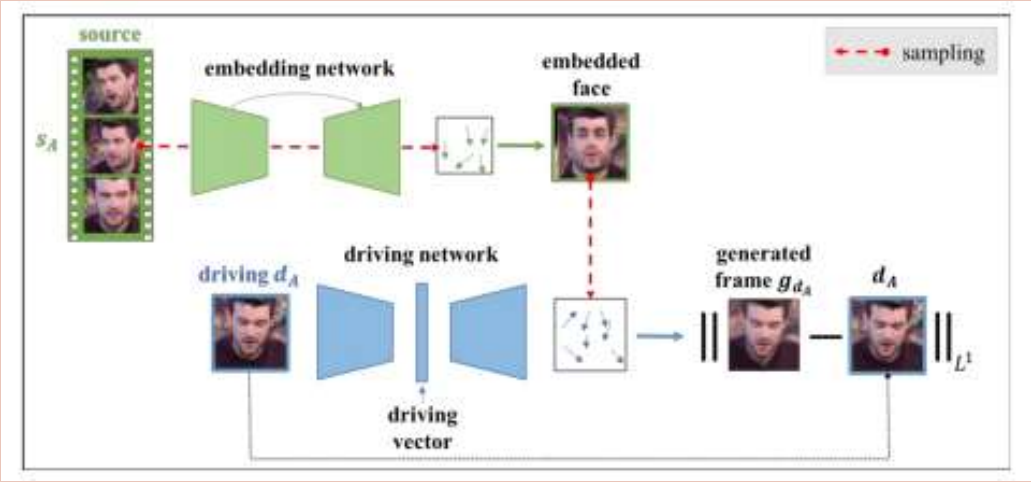
비디오의 여러 프레임을 입력을 받아 하나의 프레임은 소스 프레임으로, 다른 프레임은 driving frame으로 지정된다.

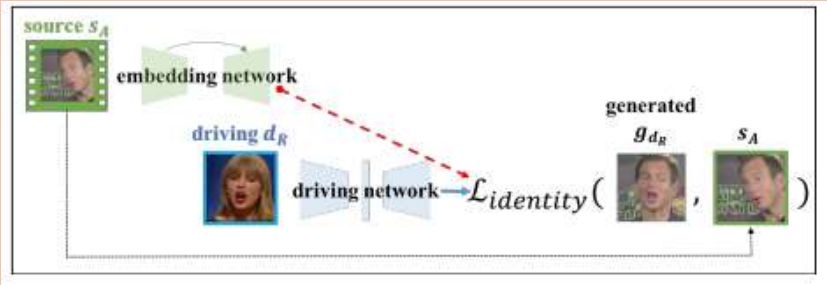
Source frame은 embedding 네트워크의 입력. Embedded face에 매핑된다.

Driving frame은 얼굴에서 생성된 프레임에 픽셀을 매핑하는 driving network에 입력

생성된 프레임은 소스 프레임의 identity와 driving 프레임의 pose, 표정을 가진다.

X2face: A network for controlling face generation by using images, audio, and pose codes





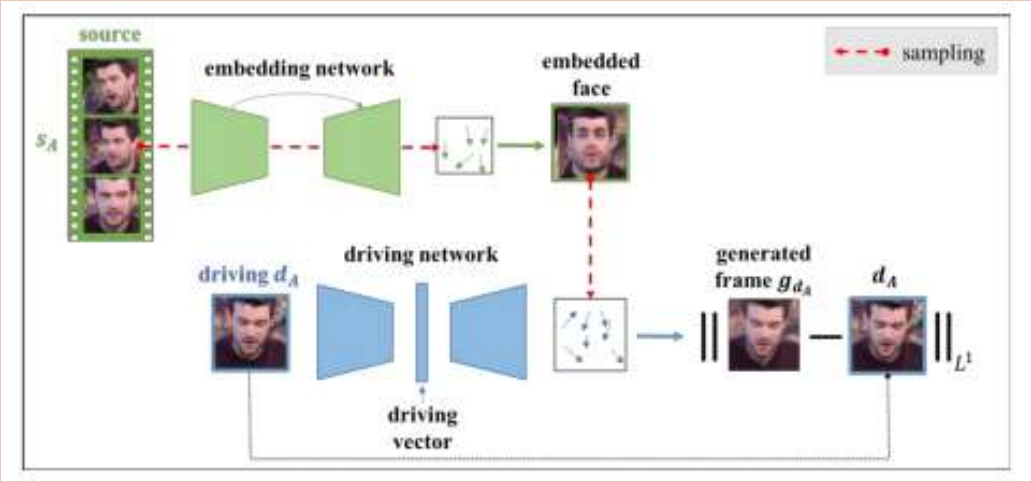
Embedding network는 bilinear sampler를 학습하여 소스 프레임에서 embedded face로 매핑하는 방법을 결정

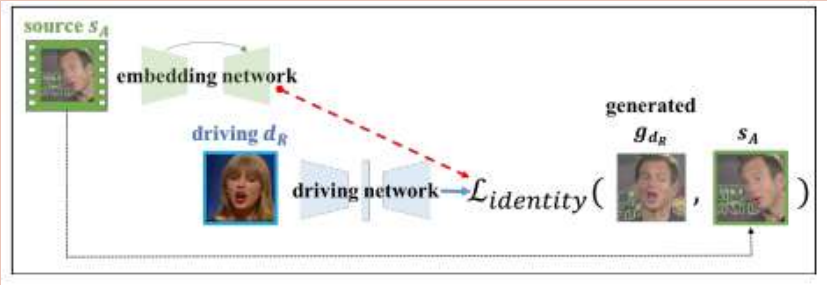
Driving network는 driving frame을 입력으로 받아 embedded face에서 픽셀을 변형하여 generated frame을 만들기 위해 bilinear sampler를 학습한다. 인코더-디코더 구조

임베딩 네트워크가 명시적으로 소스 프레임의 정면화를 강요하는 것은 아니다.

하지만 소스 프레임의 포즈와 표정을 알지 못한 채 sampling을 하기 때문에 포즈와 표현이 다른 driving과 같은 표정을 할 수 있는 공통된 표정을 지을 수 있는 embedded face가 필요하기 때문에 정면화를 한다.

X2face: A network for controlling face generation by using images, audio, and pose codes





훈련 시에는 프레임이 동일한 비디오에서 나온 것이므로 generated and driving frame이 일치해야 한다. (self-supervised)

Test 시에는 source and driving face의 identity가 다를 수 있다.

X2face: A network for controlling face generation by using images, audio, and pose codes

Controlling the image generation with other modalities

$$v_{emb}^{driving} = v_{emb}^{source} + v_{emb}^{\Delta pose} = v_{emb}^{source} + f_{p \to v}(p_{driving} - p_{source}).$$

Driving frame으로 generation을 제어하는 대신 pose code를 사용하여 source face의 head pose를 제어할 수 있다.

$$v_{emb}^{driving} = v_{emb}^{source} + f_{a \to v}(a_{driving}) - f_{a \to v}(a_{source}) + f_{p \to v}(p_{audio} - p_{source}),$$

비디오의 오디오 데이터를 이용하여 driving vector를 수정하여 포즈와 유사한 방식으로 source face를 drive할 수 있다.

X2face: A network for controlling face generation by using images, audio, and pose codes

Source frame과 generated frame의 identity 차이를 측정하여 동일한 identity를 갖게 만드는 L1 identity loss. 수식은 나오지 않음.

X2face: A network for controlling face generation by using images, audio, and pose codes

Dataset : VGG-Face

소스코드 : https://github.com/oawiles/X2Face

ReenactGAN: Learning to Reenact Faces via Boundary Transfer

ECCV 2018

임의의 사람의 비디오 입력에서 대상의 동영상으로 움직임과 표정을 전송할 수 있는 ReenactGAN

실시간으로 reenactment를 고품질로 하기 위함.

기존의 GAN 모델의 구조가 오로지 feed forward로 이루어져 있지 않아 생기는 문제점을 개선

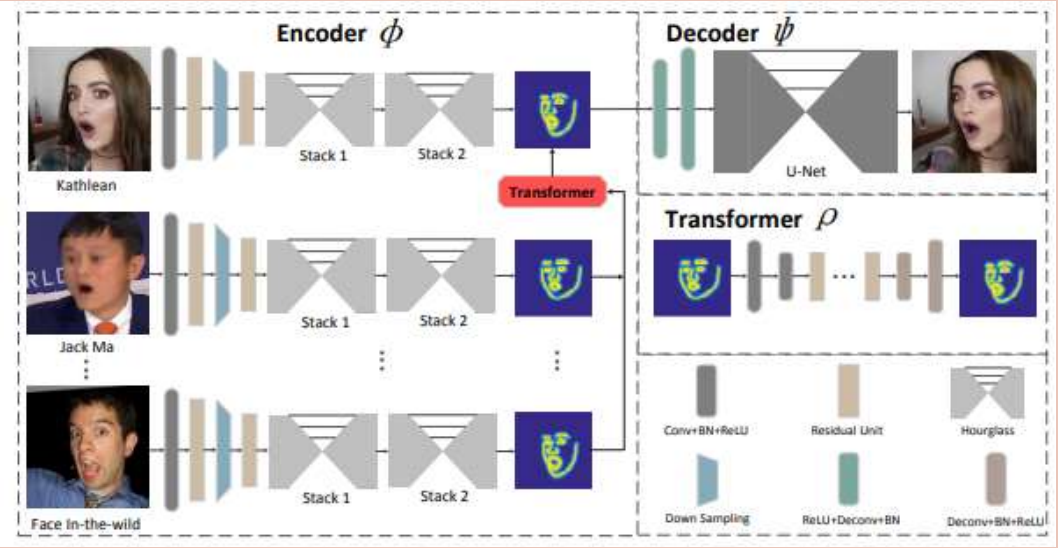
ReenactGAN: Learning to Reenact Faces via Boundary Transfer

Boundary Transfer : 구조적인 아티팩트가 발생할 수 있는 픽셀 공간에서 전송을 수행하지 않고 source face를 boundary latent space로 매핑한다.

효과적이고 신뢰할 수 있는 boundary transfer 덕분에 photo-realistic face reenactment가 가능하다.

전체 reenactment 과정이 모두 feed-forward 과정이기 때문에 reenactment가 실시간으로 진행될 수 있다.

ReenactGAN: Learning to Reenact Faces via Boundary Transfer



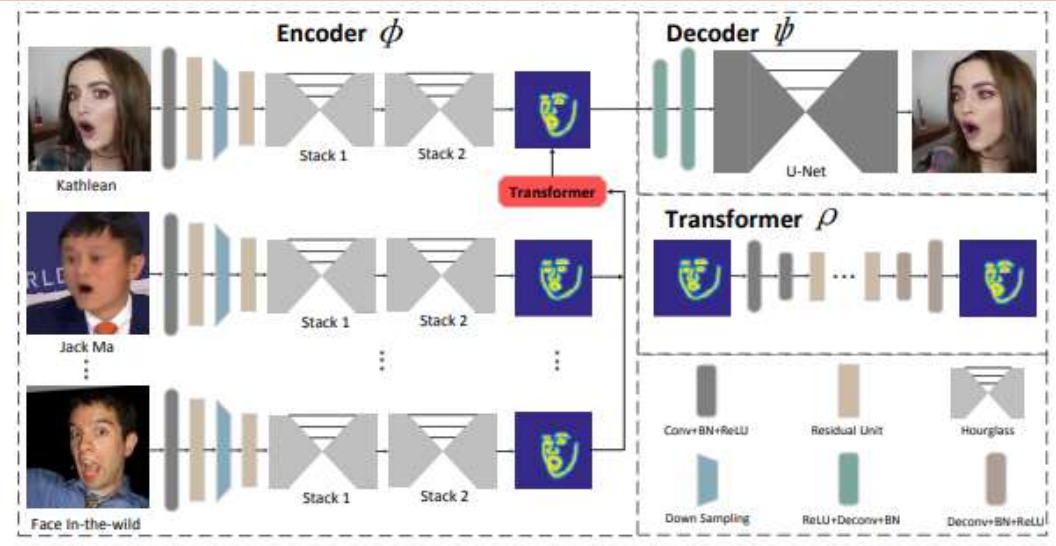
모두 feed forward인 3개의 요소로 구성된다

Boundary encoder : input face image를 latent space로 매핑 시킨다.

Target specific decoder : latent boundary를 입력으로 받아 원래의 사람의 이미지로 decode한다.

Encoder와 decoder는 최신 Pix2Pix 접근법을 사용했다.

ReenactGAN: Learning to Reenact Faces via Boundary Transfer



인코더는 φ : X → B

디코더는 ψT : B → X

X는 face at pixel space를 말한다. B는 boundary space를 말한다.

Encoder는 X의 얼굴 x를 latent space에 있는 b로 매핑하는 역할을 한다.

Decoder는 latent boundary b를 특정 얼굴의 face t ∈ T ⊂ X 로 변환

Boundary Latent space

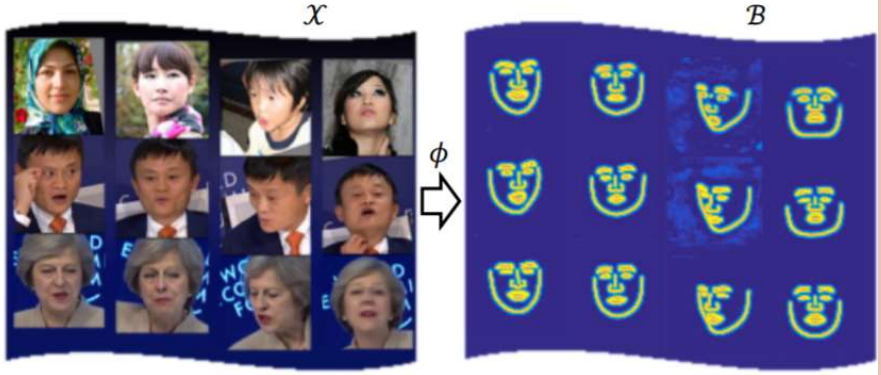
Pixel space 대신 변환이 진행되는 space

얼굴 표정에는 sensitive하나 identity에는 less sensitive하게 만들었다.

예를 들어 다른 identity와 같은 표정의 두 얼굴은 boundary space 상에는 가까이에 위치해야 한다.

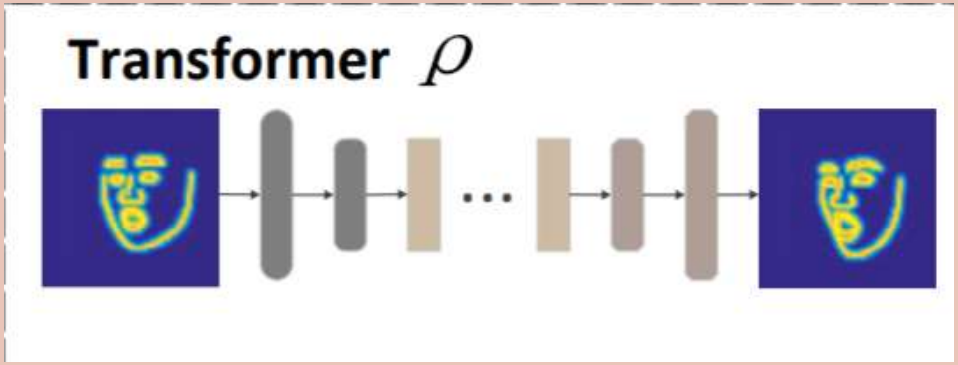
Appearance의 decode 과정을 지원하기 위해 풍부한 구조적인 정보를 포함하여야 한다.

이것을 이용하여 특정 얼굴 부분의 윤곽을 나타내는 K boundary heatmap의 stack으로 표현



ReenactGAN: Learning to Reenact Faces via Boundary Transfer

Boundary Transformer



Decode를 다른 사람의 heatmap에 적용하면 target과 source의 얼굴모양 사이에 구조적 차이가 큰 경우 심각한 아티팩트가 발생할 수 있다.

이를 위해 target-specific transformer ρT을 이용하여 해결한다.

ρT는 φ(X) to φ(T)

Source heat map을 target heatmap의 얼굴 윤곽으로 맞추자.

ReenactGAN: Learning to Reenact Faces via Boundary Transfer

$$L(\psi \cdot \phi, \theta) = L_{\text{GAN}}(\psi \cdot \phi, \theta) + L_{\ell_1}(\psi \cdot \phi) + L_{\text{feat}}(\psi \cdot \phi).$$

Combined loss for encoder and decoder.

L_GAN은 일반적인 adversarial loss로 discriminator 가 실사 이미지와 재구성된 이미지를 구별할 수 있게 해준다.
L_l1은 L1 reconstruction loss이다
L_feat은 feature간의 L2 distance를 계산한다.
이 세가지 loss function의 조합은 선명하고 사실적인 출력을 생성하기 위한 face reenactment에 널리 사용된다.

ReenactGAN: Learning to Reenact Faces via Boundary Transfer

$$L(\{\rho_\cdot\}, \{C_\cdot\}) = L_{\text{cycle}} + L_{\text{GAN}} + L_{\text{shape}},$$

Loss for transformers

$$L_{\text{cycle}} = \mathrm{E}_{i \neq j} \left[\| \rho_i \cdot \rho_j(b_i) - b_i \| \right].$$

Cycle consistency를 위한 loss이다. P는 각각의 transformer이다.

L_GAN은 vanila GAN loss이다.

$$L_{\text{shape}} = \mathrm{E}_{b \in \mathcal{B}, i \in 1,\ldots,N} \left[R(b) - R \cdot \rho_i(b) \right]$$

Shape constrain loss

변형된 boundary가 Source를 더 잘 따르도록 장려하는 loss.

이 loss는 transformer의 input과 output 사이에 정의된다.

ReenactGAN: Learning to Reenact Faces via Boundary Transfer

Dataset : Celebrity Video Dataset, Boundary Estimation Dataset.

소스 코드 : https://github.com/wywu/ReenactGAN

Ganimation: Anatomically-aware facial animation from a single image

ECCV2018

기존의 GAN을 사용한 모델의 데이터 내용에 의해 결정되는 개별적인 수의 표정만 생성할 수 있다는 문제점을 해결하기 위함.

그냥 웃음 이런 표정 말고 표정의 강도도 설정을 해보자

Ganimation: Anatomically-aware facial animation from a single image

Anatomically-aware : 해부학적 얼굴 움직임을 연속적인 AU annotation으로 사용

데이터 집합의 내용에 의해 결정되는 개별적인 수의 표정만 생성하는 기존의 모델과 달리 인간의 표정을 정의하는 해부학적 얼굴 움직임을 연속적인 manifold로 설명하는 AU annotation을 기반으로 한 GAN 모델. 얼굴표정은 분류할 수 없는 얼굴 근육의 결합이다.

Supervision이 필요없다. 해부학적 얼굴 움직임을 연속적인 AU annotation으로 사용하기 때문이다.

Ganimation: Anatomically-aware facial animation from a single image

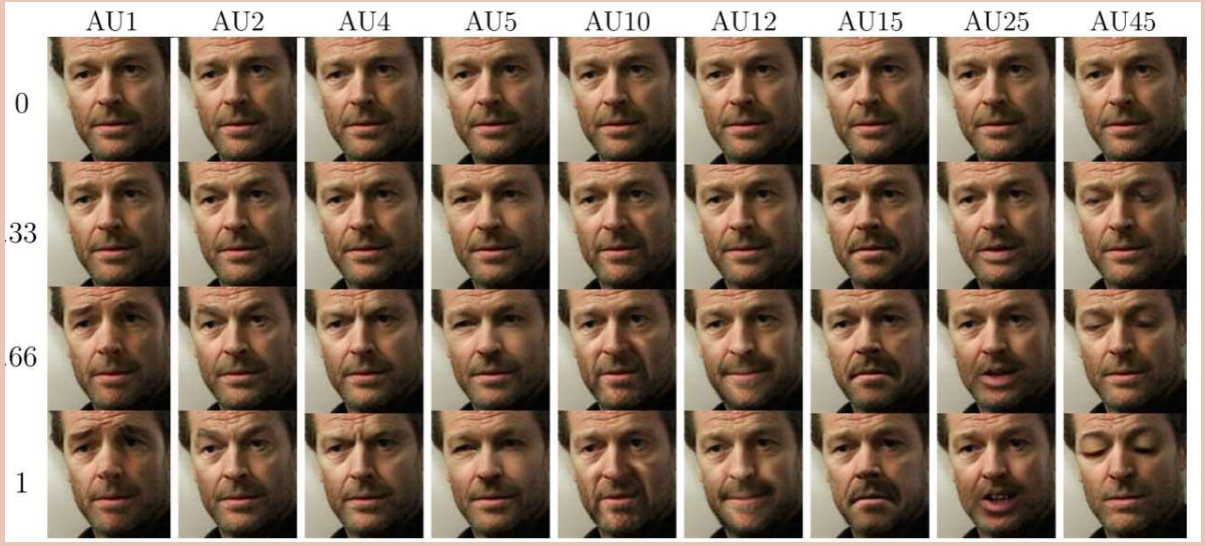
모든 표정은 N개의 action unit으로 인코딩 된다.

$y_r=(y1,y2,...,yN)^T$

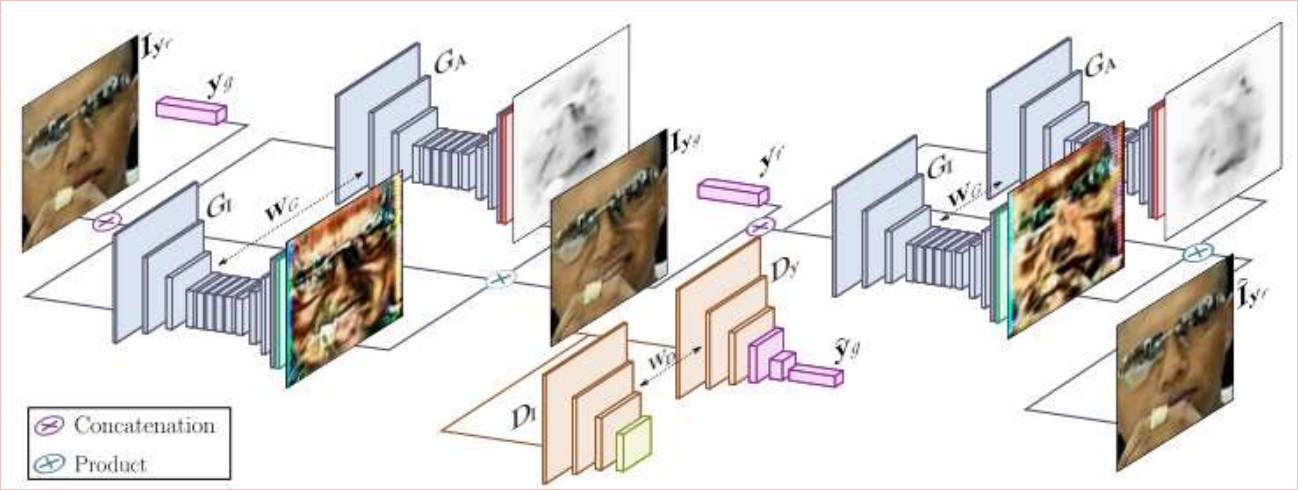
각각의 action unit 값들은 정규화하여 0~1사이의 숫자로 표현된다

이러한 연속적인 action unit의 값으로 표정을 임베딩하기 때문에 표정의 세기를 결정할 수 있다.

예) 웃음 : (0.7,0.5,...0.1) -> 강한웃음 : (0.9,0.7,...,0.2)



Ganimation: Anatomically-aware facial animation from a single image



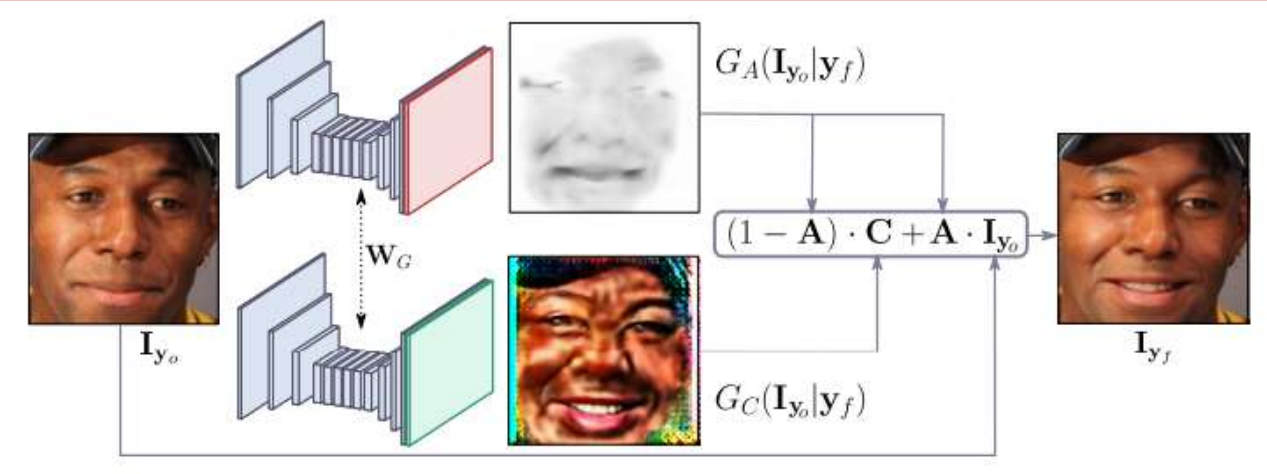
두개의 main block으로 구성되어 있다.

Generator G : attention 및 color mask를 regress 한다.

Generator의 요점은 새로운 표정을 만드는 것에 관여하는 부분만을 변경하고 나머지는 변경하지 않는 법

Photorealism 과 표정 조건화 이행 하에서 생성된 이미지를 평가하는 discriminator D

Ganimation: Anatomically-aware facial animation from a single image



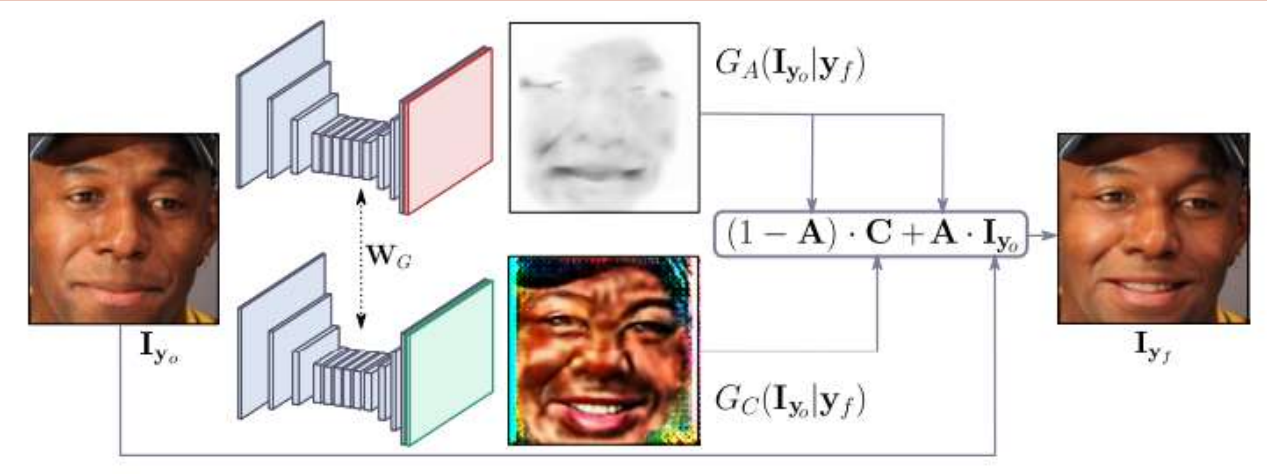
Attention-based generator

Generator가 새로운 표정을 합성하는 역할을 하는 이미지 영역에만 초점을 맞추고 머리,안경,모자 같은 나머지 요소들은 손을 대지 않는다. 이것을 attention이라고 한다.(표정에 집중한다.)

입력 이미지와 대상표정이 주어지면 generator는 전체 이미지에서 attention mask A와 RGB color transformation C를 regress 한다.

Attention mask는 원본이미지의 픽셀 당 최종 렌더링 이미지에 어떤 역할을 미치는지 지정하는 픽셀 당 강도를 정의

Ganimation: Anatomically-aware facial animation from a single image



Iyf = (1 − A) · C + A · Iyo

C는 입력 이미지에 G_c를 적용하여 얻은 color mask

A는 입력 이미지에 G_A를 적용하여 얻은 Attention mask

Ganimation: Anatomically-aware facial animation from a single image

WGAN-GP에서 제안한 image adversarial loss.

$$\mathbb{E}_{\mathbf{I}_{\mathbf{y}_o}\sim\mathbb{P}_o}[D_{\mathrm{I}}(G(\mathbf{I}_{\mathbf{y}_o}|\mathbf{y}_f))] - \mathbb{E}_{\mathbf{I}_{\mathbf{y}_o}\sim\mathbb{P}_o}[D_{\mathrm{I}}(\mathbf{I}_{\mathbf{y}_o})] + \lambda_{\mathrm{gp}}\mathbb{E}_{\widetilde{I}\sim\mathbb{P}_{\widetilde{I}}}\left[(\|\nabla_{\widetilde{I}}D_{\mathrm{I}}(\widetilde{I})\|_2 - 1)^2\right]$$

Attention loss

$$\lambda_{\mathrm{TV}}\mathbb{E}_{\mathbf{I}_{\mathbf{y}_o}\sim\mathbb{P}_o}\left[\sum_{i,j}^{H,W}[(\mathbf{A}_{i+1,j} - \mathbf{A}_{i,j})^2 + (\mathbf{A}_{i,j+1} - \mathbf{A}_{i,j})^2]\right] + \mathbb{E}_{\mathbf{I}_{\mathbf{y}_o}\sim\mathbb{P}_o}[\|\mathbf{A}\|_2]$$

Conitional expression loss

$$\mathbb{E}_{\mathbf{I}_{\mathbf{y}_o}\sim\mathbb{P}_o}\left[\|D_{\mathrm{y}}(G(\mathbf{I}_{\mathbf{y}_o}|\mathbf{y}_f))] - \mathbf{y}_f\|_2^2\right] + \mathbb{E}_{\mathbf{I}_{\mathbf{y}_o}\sim\mathbb{P}_o}\left[\|D_{\mathrm{y}}(\mathbf{I}_{\mathbf{y}_o}) - \mathbf{y}_o\|_2^2\right]$$

Identity Loss

$$\mathcal{L}_{\mathrm{idt}}(G, \mathbf{I}_{\mathbf{y}_o}, \mathbf{y}_o, \mathbf{y}_f) = \mathbb{E}_{\mathbf{I}_{\mathbf{y}_o}\sim\mathbb{P}_o}[\|G(G(\mathbf{I}_{\mathbf{y}_o}|\mathbf{y}_f)|\mathbf{y}_o) - \mathbf{I}_{\mathbf{y}_o}\|_1]$$

Ganimation: Anatomically-aware facial animation from a single image

Dataset : RaFD,EmotioNet dataset

소스코드 : https://github.com/albertpumarola/GANimation