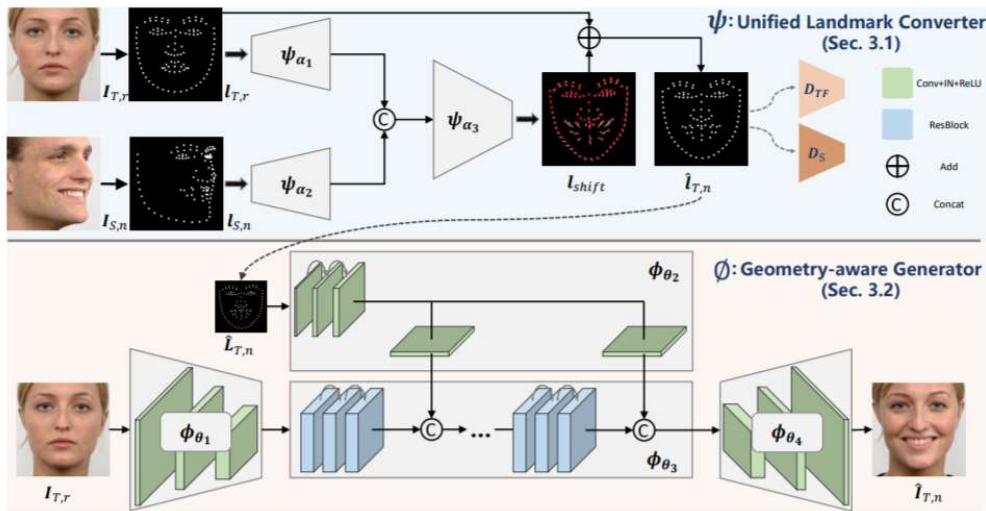




Various ways of extracting and expressing facial expression and Identity

FReeNet: Multi-identity Face Reenactment

CVPR 2020



○ 추출

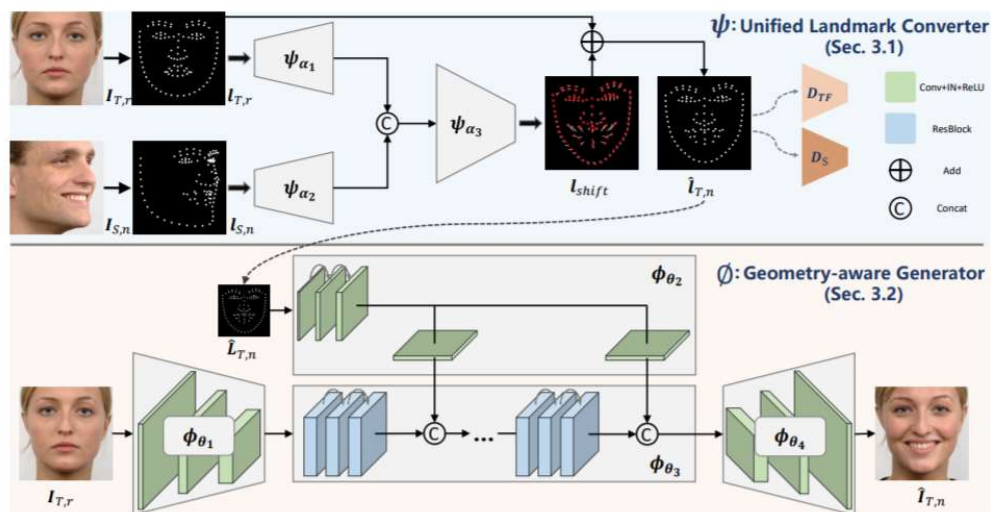
- Face landmark detector를 통해 두개의 input 이미지를 latent space로 encode 한다.
- Landmark는 벡터형식으로 출력된다
- $I_{T,r}$ 은 T identity, r표정의 이미지를 뜻하고 $I_{T,n}$ 은 T identity, n표정의 랜드마크
- 이 논문은 표정을 따로 추출하는 것이 아니라 추출한 landmark를 표정으로 사용

○ generate landmark

- ψ_{α_1} 과 ψ_{α_2} 는 landmark features를 추출하는 encoder
- ψ_{α_3} 은 shift decoder로 두 랜드마크 사이의 차이를 이용해 I_{shift} 를 생성.
- $I_{T,r}$ 에 I_{shift} 를 concat하여 $I_{T,n}$ 를 생성한다.

FReeNet: Multi-identity Face Reenactment

CVPR 2020

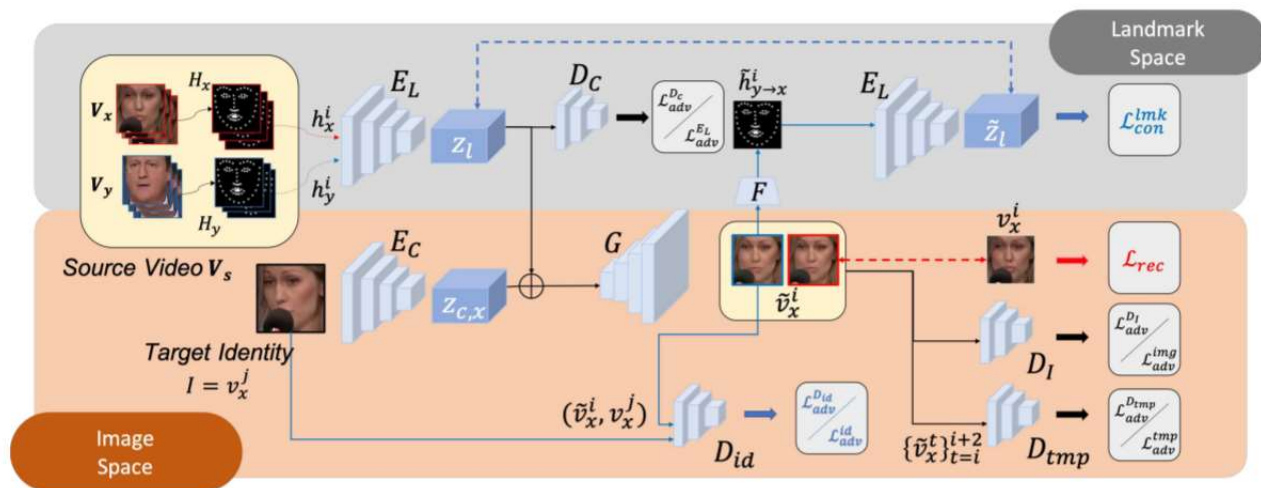


표정 추출

- landmark(vector)를 표정으로 사용
- 입력으로 V_s (T 프레임)과 해당하는 얼굴 랜드마크 H_s 를 입력으로 한다.
- 구조에 landmark를 추출하는 것이 없고 입력으로 주는 것을 보아 landmark detector를 사용하여 landmark를 추출하는 것으로 추정

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

CVPR 2020

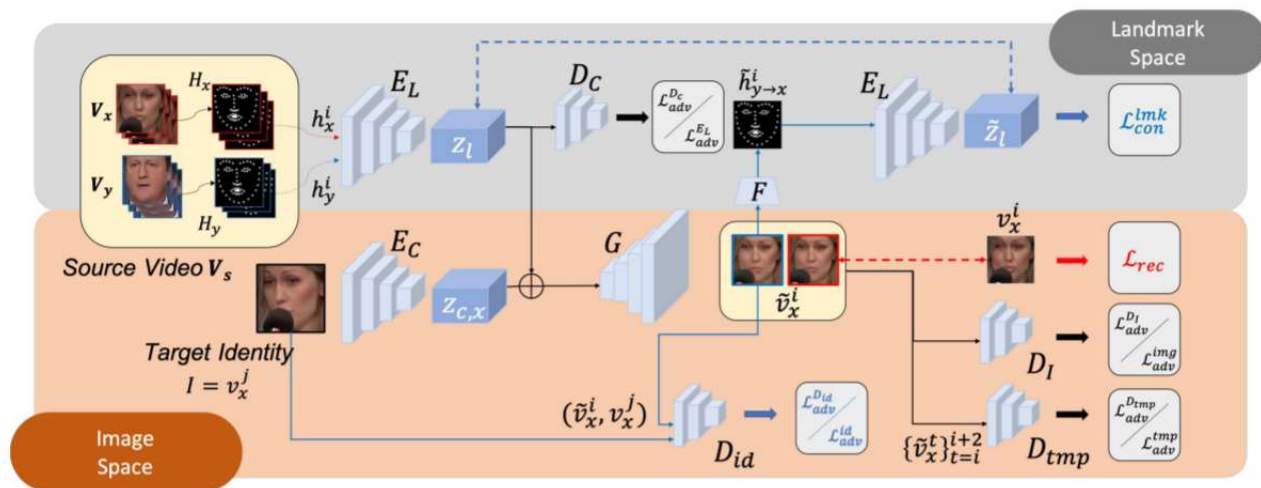


○ 표정 추출

- landmark(vector)를 사용
- 입력으로 V_s (T 프레임)과 해당하는 얼굴 랜드마크 H_s 를 입력으로 한다.
- 구조에 landmark를 추출하는 것이 없고 입력으로 주는 것을 보아 landmark detector를 사용하여 landmark를 추출하는 것으로 추정
- E_L : ID-invariant facial landmark encoder, 측정된 랜드마크를 ID-invariant motion latent code로 인코딩. 이것을 표정으로 사용
- E_C : ID-preserving content encoder, 입력과 identity latent code를 매핑

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

CVPR 2020



○ 표정 추출

- D_C 는 content/id classifier로 E_L 로 뽑아낸 latent code가 ID-invariant 한지 판단.
- D_I 는 Image discriminator로 생성된 이미지와 실제 이미지의 차이를 구하는 일반적인 discriminator
- D_{tmp} 는 video discriminator로 source-domain video의 sequence와 결과물로 나온 fake sequence의 차이를 측정
- D_{id} 는 target image의 identity latent code에 source video의 id-invariant motion latent code를 concat하여 생성한 code를 generate하여 생성한 이미지와 target이미지의 id 차이를 구하는 역할

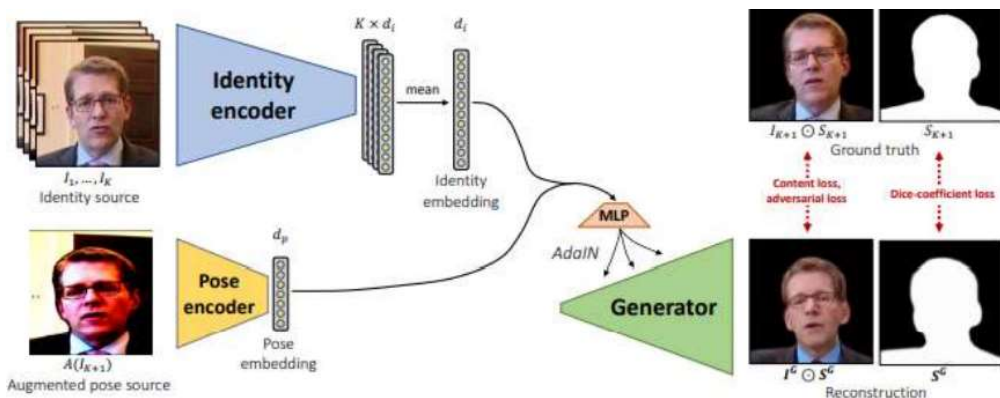
Learning Identity-Invariant Motion Representations for Cross-ID Face Reenactment

CVPR 2020



Neural Head Reenactment with Latent Pose Descriptors

CVPR 2020



○ 표정 추출 및 표현

- 본 논문에서는 표정을 pose에 포함하여 pose encoder를 통해 pose와 expression을 추출
- pose와 expression은 pose embedding vector로 추출 및 표현
- id는 identity encoder를 통해 pose-independent한 정보를 담고있는 identity embedding 추출
- 입력 으로는 여러 프레임이 들어오므로 여러 identity embedding 이 나오는데 이를 평균을 내서 사용한다.
- pose encoder는 MobileNetV2 구조, identity encoder는 ResNeXt-50 구조

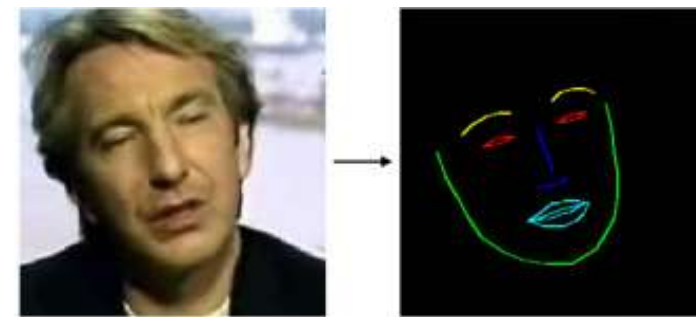
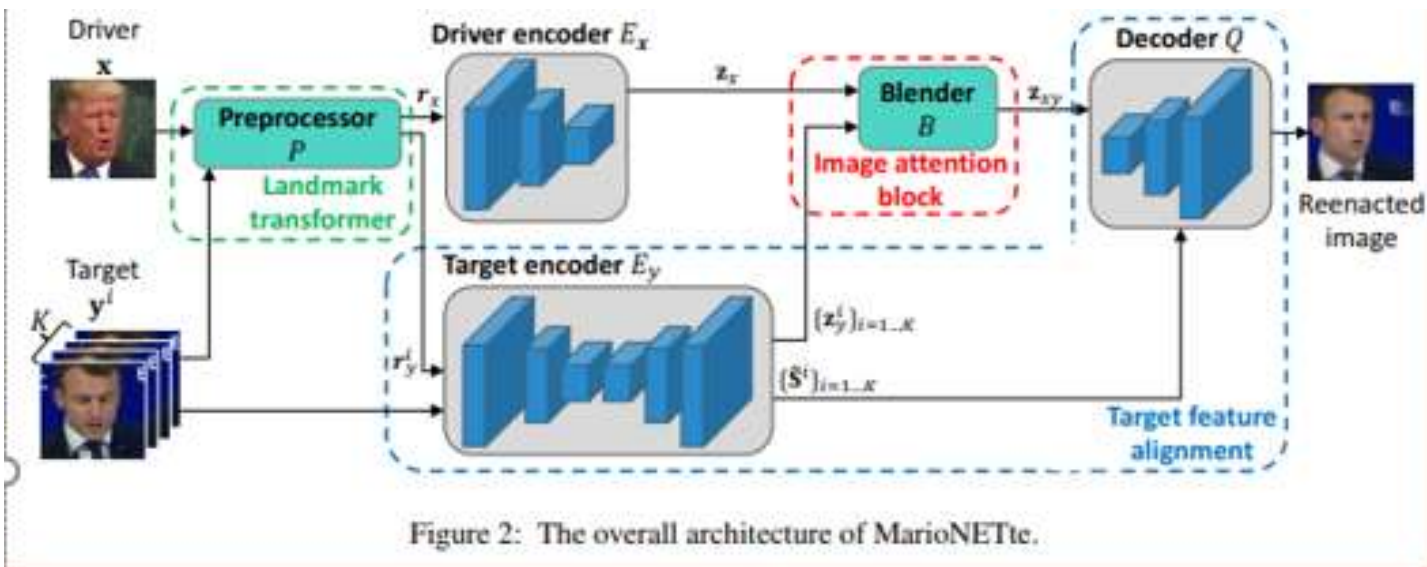
Neural Head Reenactment with Latent Pose Descriptors

CVPR 2020



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

AAAI 2020



○ 표정 추출 및 표현

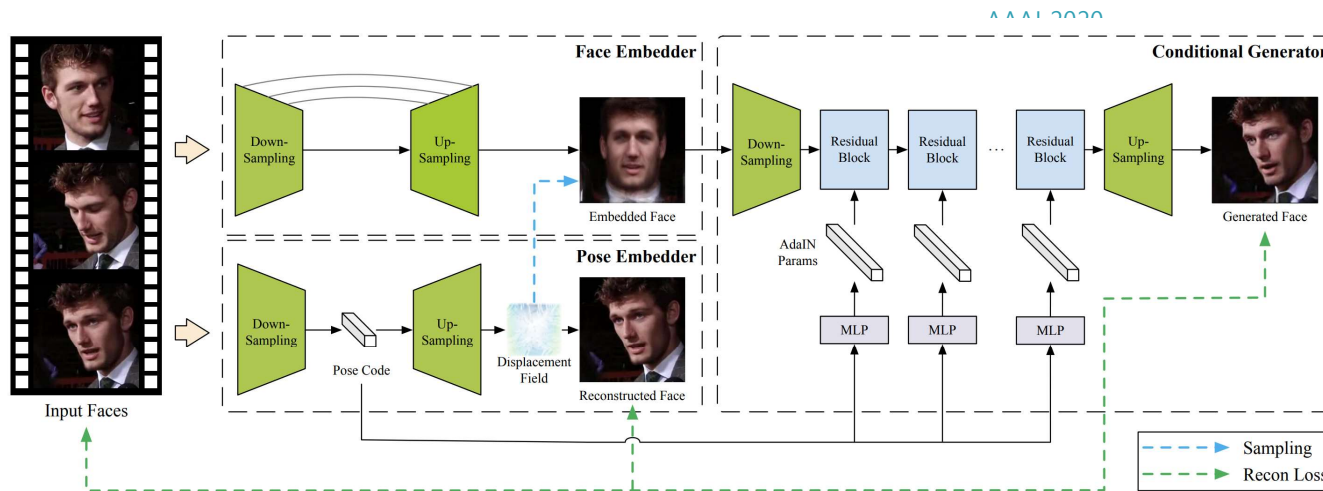
- 표정을 facial landmark(key point)로 나타냄
- 2D landmark를 사용하지 않고 3D landmark를 사용
- Preprocessor P에서 3D landmark detector를 사용하여 facial key point를 추출해낸다.
- 추출한 3d facial key point를 2d landmark image로 render한다.
- landmark이미지를 이용하여 pose, expression information을 feature z_x 로 뽑아낸다.

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

AAAI 2020



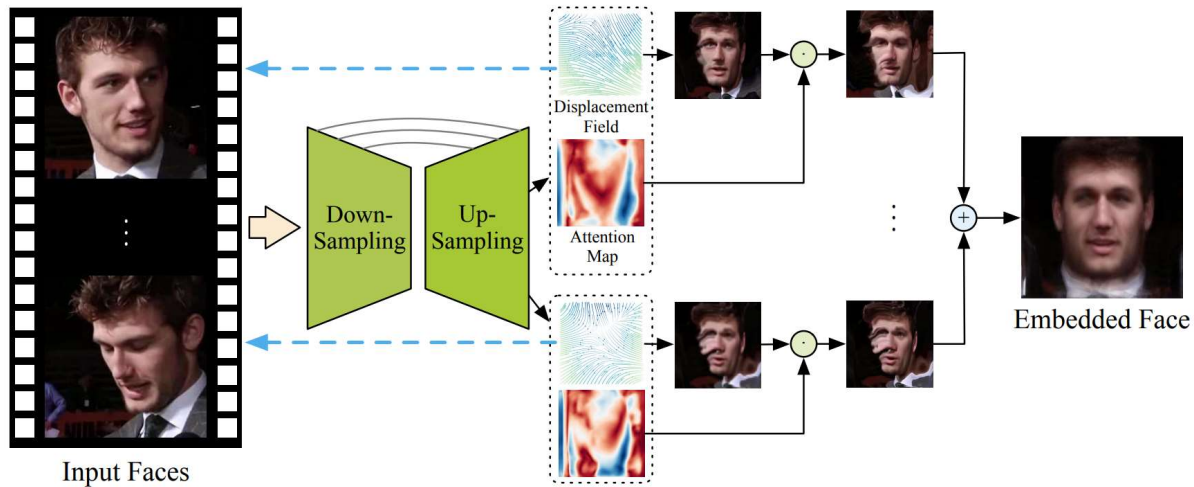
Realistic Face Reenactment via Self-Supervised Disentangling of Identity and Pose



○ 표정 추출 및 표현

- Face embedder, Pose embedder을 통해 identity와 pose feature를 분리
- Face embedder은 입력 이미지를 받아 embedded face를 생성하는데 각 frame의 pose와 expression에 invariant한 Id feature만 포함되어 있다.
- Pose embedder은 embedded face와 다른포즈를 입력으로 받는다
- 각각의 프레임의 pose-related information을 추출한다.
- displacement field는 학습시에 예측한 bilinear sampling

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



○ face embbader

- displacement field T(bilinear sampler)와 attention map을 예측한다
- 각각의 frame에 displacement field를 적용하여 이미지를 생성한 후 attention map을 이용해 다시 이미지를 생성한다
- 각 프레임에서 생성된 이미지를 concat하여 embedded face를 생성한다.
- embedded face는 identity라고 생각하면 된다.

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

CVPR 2016

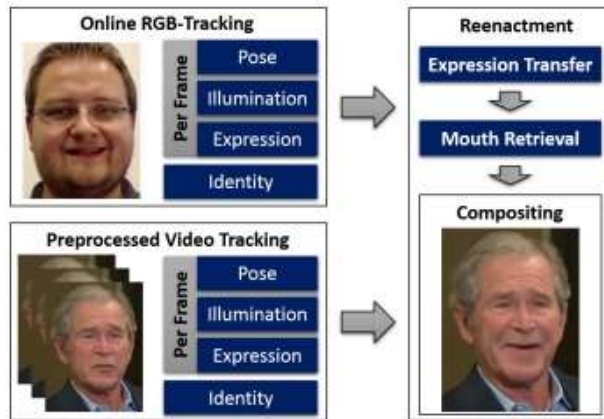


Figure 1: Method overview.

○ 표정 추출 및 표현

- Multi-linear PCA model을 사용한다.
- Multi-linear PCA model은 주성분 분석을 하여 주성분만 뽑아내어 표현하는 것
- Multi-linear PCA model의 처음 두차원은 face identity와 skin reflectance 를 표현한다.
- 나머지 한 차원은 얼굴 표정을 표현한다.
- 이렇게 추출한 PCA 모델을 사용하여 id와 표정을 별도로 조절할 수 있다.

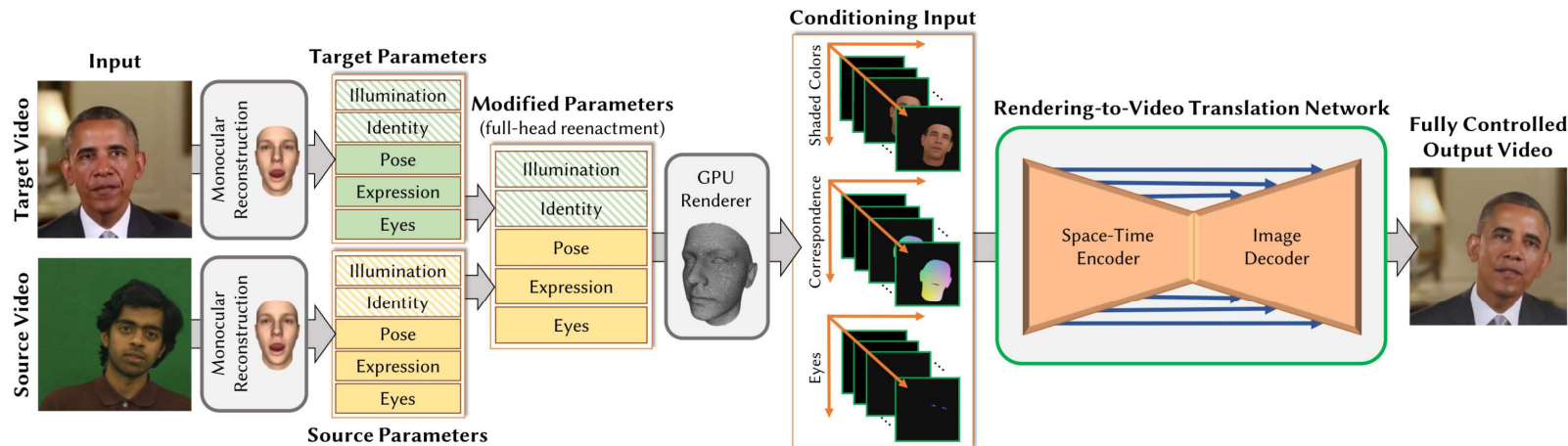
Face2Face: Real-time Face Capture and Reenactment of RGB Videos

CVPR 2016



Deep Video Portrait

ToG 2018

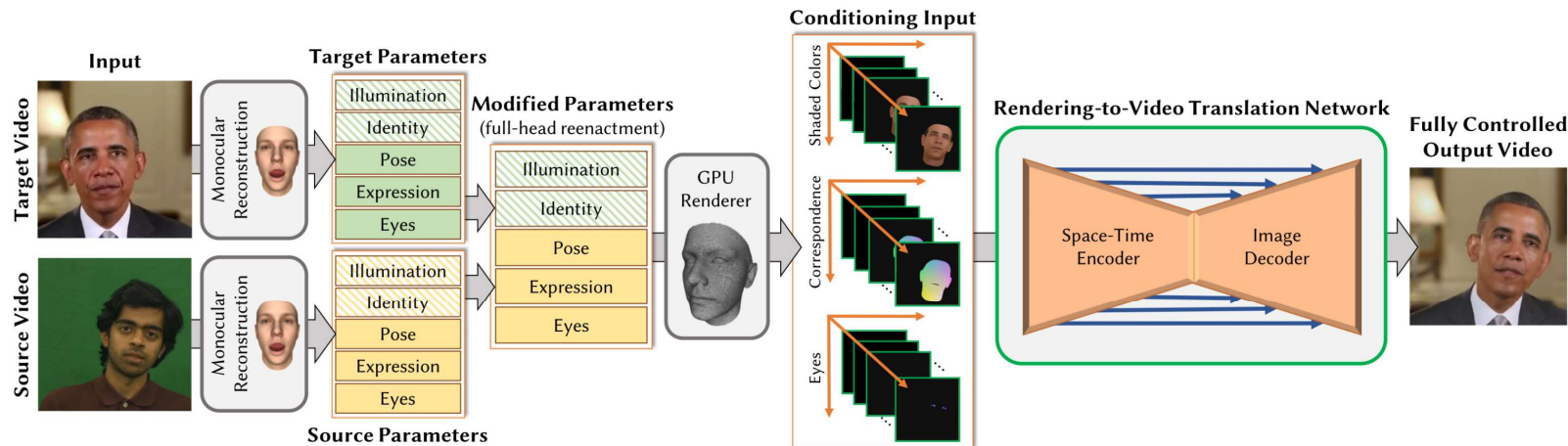


○ 표정 추출 및 표현

- monocular face reconstruction을 통해 소스 비디오와 타겟 비디오의 low dimensional parametric representation을 구한다.
- low dimensional parametric parameter에는 illumination, identity, Pose, Expression, Eyes로 구성
- parameters는 앞에 봤던 face2face논문의 multi-linear-PCA-moel을 사용해 추출
- Target parameters와 source parameters들을 이용하여 modified parameters를 만든다.

Deep Video Portrait

ToG 2018



○ 표정 추출 및 표현

- Modified parameter을 이용, GPU Renderer를 사용하여 재현 영상을 렌더링 한다.
- 렌더링 과정에서 CGAN구조를 이용한다.
- Shaded colors, correspondence, Eyes 의 condition하에서 CGAN을 학습한다.

Deep Video Portrait

ToG 2018

Source



Target



Source

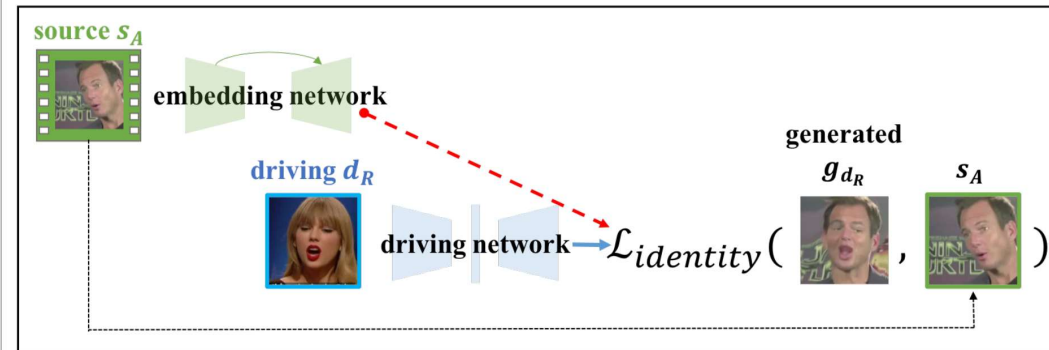
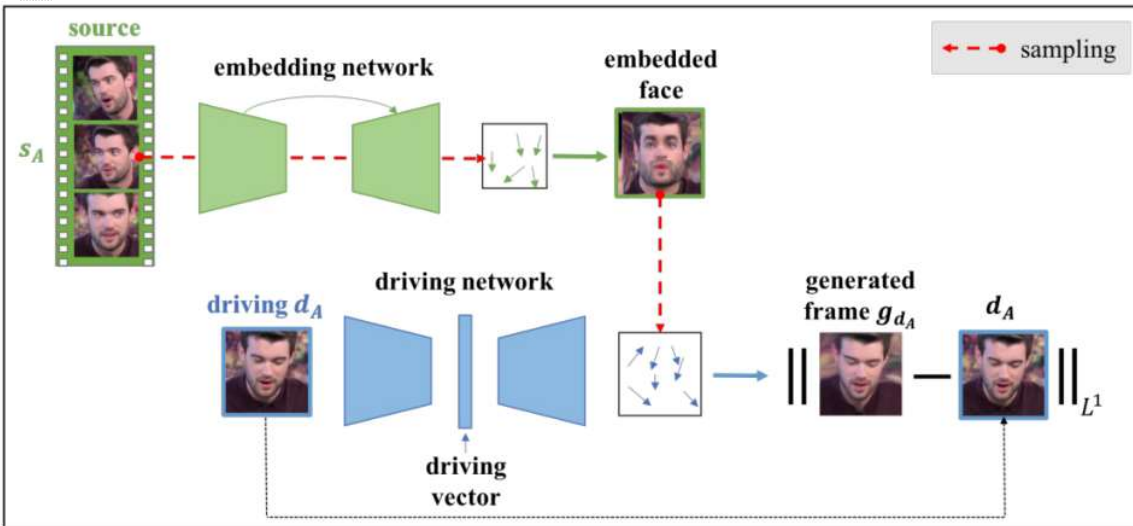


Target



X2Face: A network for controlling face generation using images, audio, and pose codes

ECCV 2018

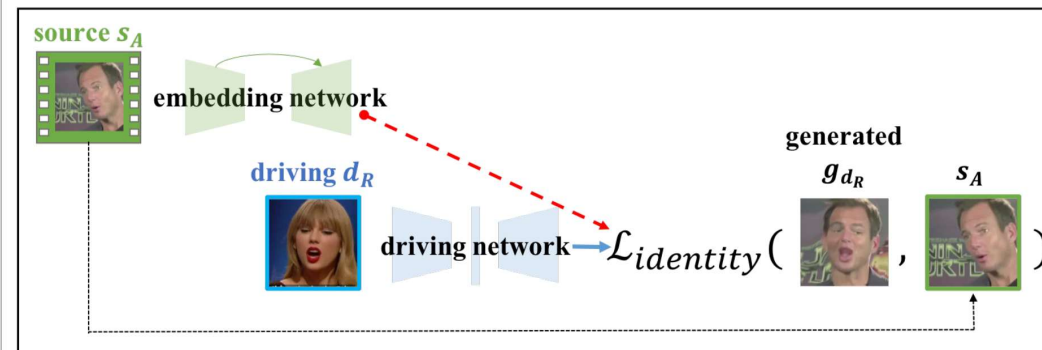
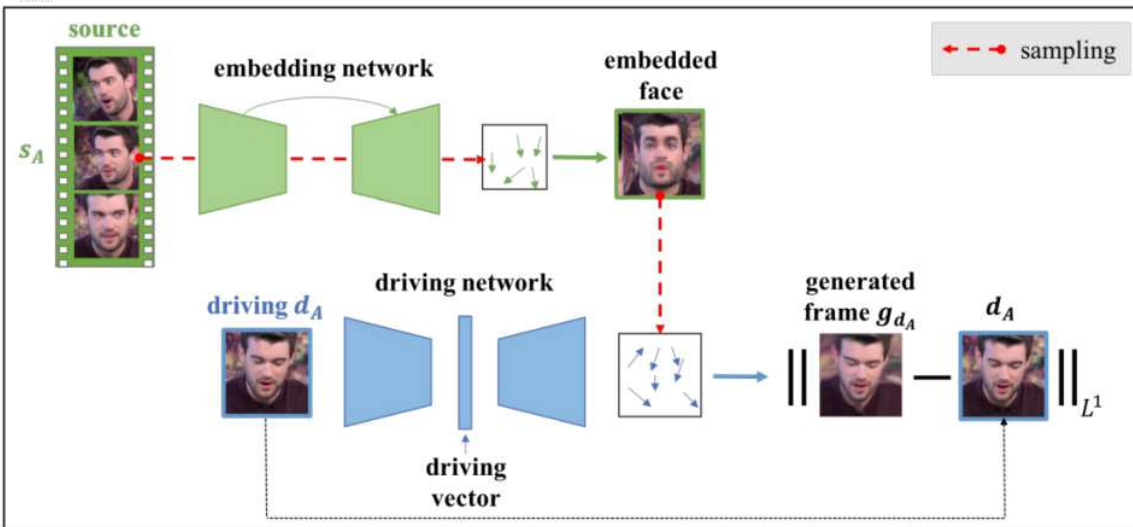


○ 표정 추출 및 표현

- bilinear sampler를 학습하여 소스프레임에서 embedded face로 매핑
- embedding network는 U-net구조와 pix2pix구조에 기반했으며 출력은 2-channel image
- 각각의 픽셀의 흐름인 $\delta x, \delta y$ 을 인코딩한 2-channel image
- driving network는 driving frame을 입력으로 받아 bilinear sampler을 학습
- embedded face에 driving network에서 추출한 bilinear sampler를 적용해 generated frame을 생성

X2Face: A network for controlling face generation using images, audio, and pose codes

ECCV 2018



○ 표정 추출 및 표현

- driving network에서는 driving vector를 embedding하는데 driving vector에는 pose, expression, zoom, other factors of variation이 encode되어 있다.
- 추출된 driving vector는 pose code로 사용가능
- image를 사용한 train에서는 pose code는 추출만 할 뿐 train에 영향을 주는 것은 아니다
- 따라서 본 논문에서 표정으로 사용한 것은 bilinear sampler이다.

X2Face: A network for controlling face generation unsing images, audio, and pose codes

ECCV 2018

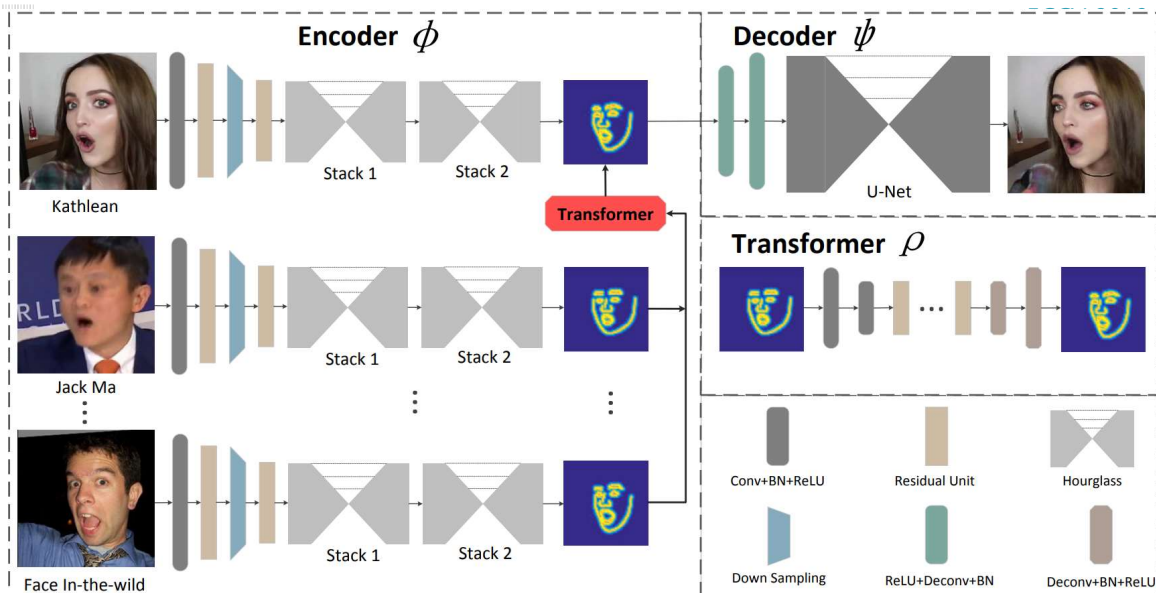


(a)



(b)

ReencactGAN: Learning to Reenact Faces via Boundary Transfer



○ 표정 추출 및 표현

- encoder로 input face image를 boundary latent space로 매핑

- 인코더는 $\phi : X \rightarrow B$

- 디코더는 $\psi^T : B \rightarrow X$

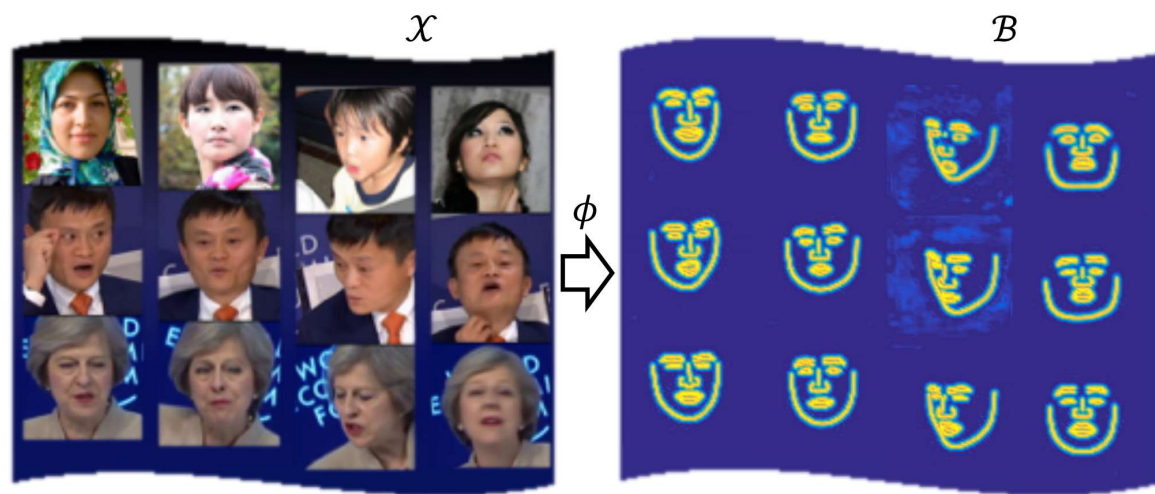
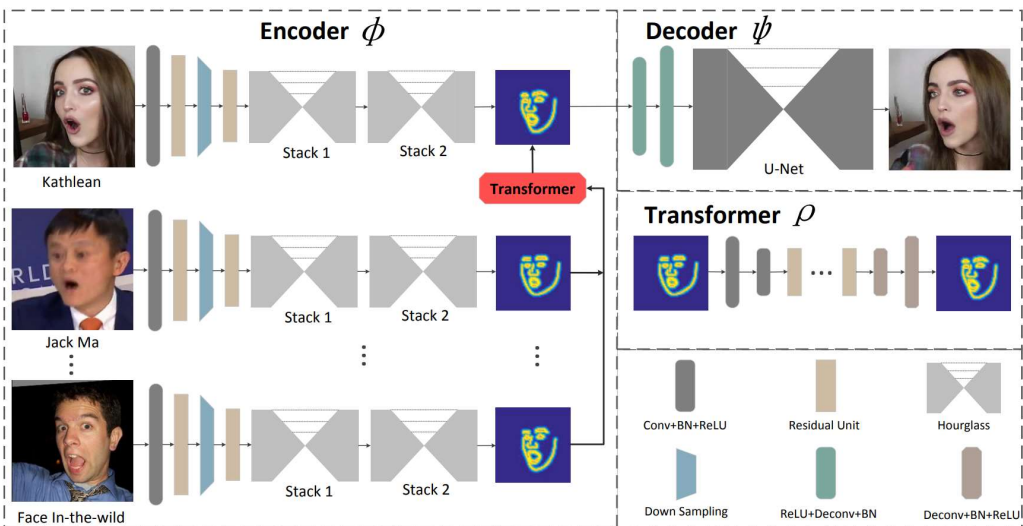
- X 는 face at pixel space를 말한다. B 는 boundary space를 말한다.

- Encoder는 X 의 얼굴 x 를 latent space에 있는 b 로 매핑하는 역할을 한다.

- Decoder는 latent boundary b 를 특정 얼굴의 face $t \in T \subset X$ 로 변환

ReencactGAN: Learning to Reenact Faces via Boundary Transfer

ECCV 2018



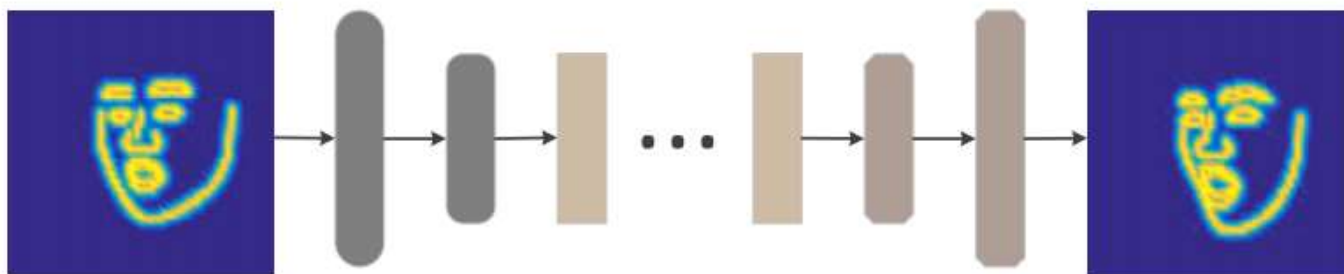
Boundary Latent space

- 얼굴 표정에는 sensitive하나 identity에는 less sensitive하게 만들었다.
- 얼굴의 구조적인 정보를 풍부하게 포함하고 있다.
- 특정 얼굴 부분의 윤곽을 나타내는 K boundary heatmap의 stack으로 표현
- k boundary heatmap으로 표정과 윤곽을 표현

ReencactGAN: Learning to Reenact Faces via Boundary Transfer

ECCV 2018

Transformer ρ



○ Transformer

- Decode를 다른 사람의 heatmap에 적용하면 target과 source의 얼굴모양 사이에 구조적 차이가 큰 경우 심각한 아티팩트가 발생할 수 있다.
- 이를 위해 target-specific transformer ρ_T 을 이용하여 해결한다.
- ρ_T 는 $\varphi(X)$ to $\varphi(T)$
- Source heat map을 target heatmap의 얼굴 윤곽으로 맞추자.

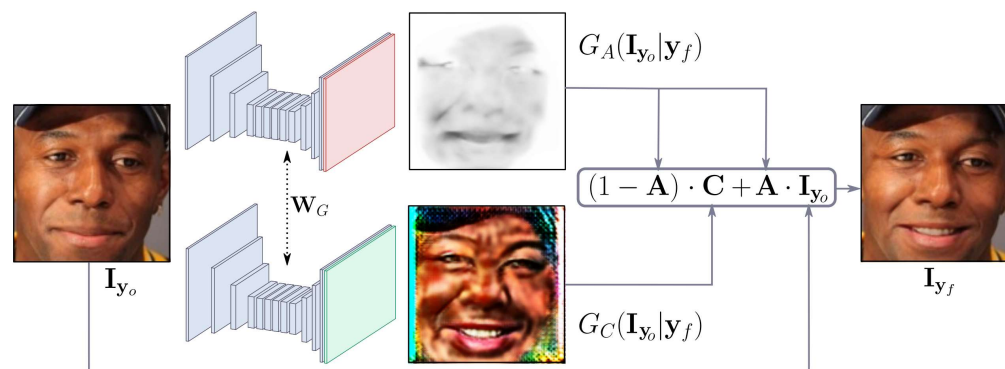
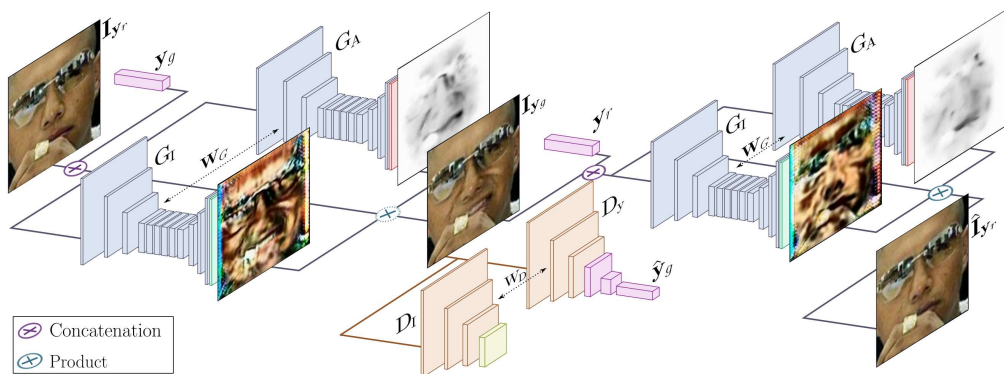
ReencactGAN: Learning to Reenact Faces via Boundary Transfer

ECCV 2018



Ganimation: Anatomically-aware facial animation from a single image

ECCV 2018

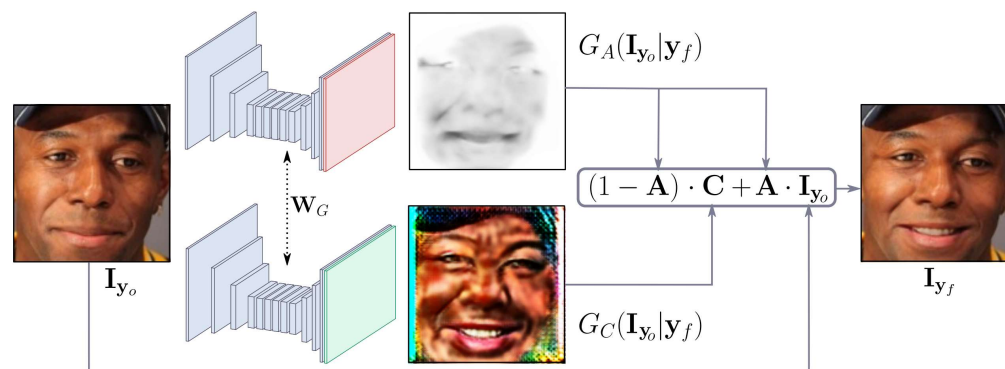
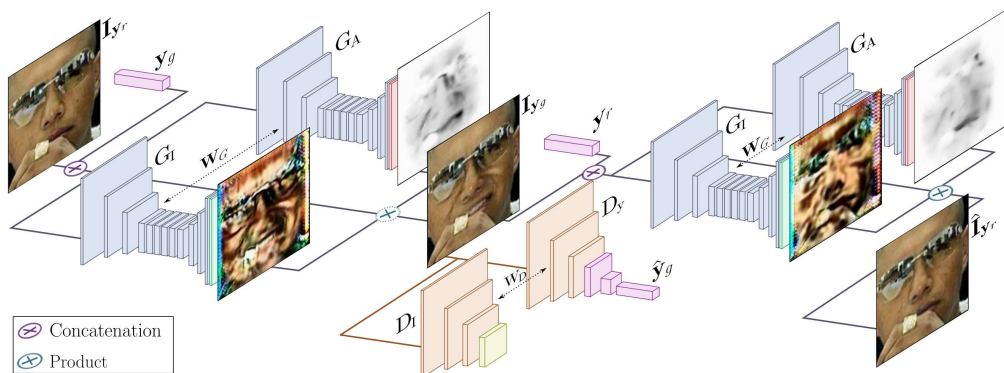


표정 추출 및 표현

- 각각의 표정은 N개의 action unit으로 인코딩, vector로 나타낸다.
- vector의 각 성분은 해당하는 AU의 값을 나타낸다.
- 값이 바뀔때마다 해당하는 AU의 모습이 바뀐다.
- 연속적인 action unit의 값으로 표정을 임베딩하기 때문에 표정의 세기를 결정할 수 있다.

Ganimation: Anatomically-aware facial animation from a single image

ECCV 2018



표정 추출 및 표현

- attention mask 및 color mask를 regress한다.
- attention mask는 새로운 표정을 합성하는 역할을 하는 이미지 영역에만 초점을 맞춤
- 이러한 attention mask에 표정정보가 포함되어 있다.
- A는 입력 이미지에 G_A 를 적용하여 얻은 Attention mask
- attention mask는 원본 이미지의 픽셀 당 최종 렌더링 이미지에 어떤 역할을 미치는지 지정하는 픽셀 당 강도를 정의 -> 따라서 표정에 따라 픽셀 당 역할이 다르므로 다르게 표현된다!

Ganimation: Anatomically-aware facial animation from a single image

ECCV 2018

AU1

AU2

AU4

AU5

AU10

AU12

AU15

AU25

AU45

0



.33



.66



1



Expression

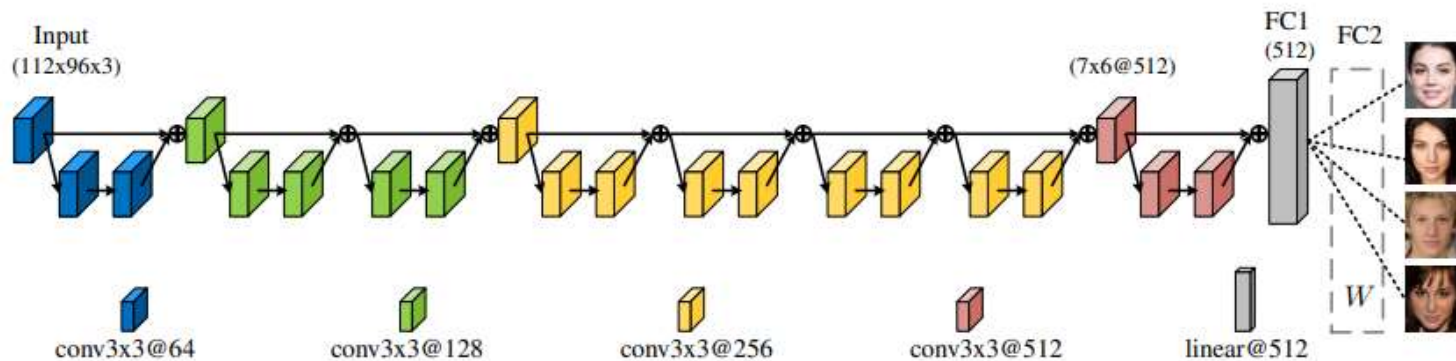
- landmark로 expression을 추출 및 표현
 - landmark detector을 이용해 landmark(vector)를 추출한다.
 - 3d landmark를 추출한 후 2d landmark 이미지로 변환하여 사용하는 것도 있음
- embedder를 사용하여 expression을 추출
 - embedder를 사용하여 이미지의 특성을 id와 expression을 분리하여 추출
 - embedder를 사용해 vector에 id, expression, pose등 정보들을 포함하는 vector를 추출하여 사용
- 직접 만든 latent space나 attention mask로 매핑
 - 얼굴 이미지의 특성을 추출하기 위해 latent space로 매핑
 - attention mask는 최종 generation에 각 픽셀이 얼마나 영향을 미치는지 표현한 것.
- multi-linear PCA model을 사용
 - pca model을 사용하여 주성분을 추출
 - 각 차원에는 id, expression pose, eye gaze 등 정보를 담고 있다.
- bilinear sampler를 사용
 - bilinear sampler는 각 픽셀의 흐름을 나타내는 것으로 표정을 나타낼 수 있다.

Identity

- Reenactment에는 id와 expression을 분리하는 작업이 중요하게 작용
- PCA model, Embedding, encode를 통해 각 얼굴의 특성을 vector형식으로 추출
- expression과 id가 vector의 다른 차원에 있어 따로 조절할 수 있다.

RegularFace: Deep Face Recognition via Exclusive Regularization

CVPR 2019



○ Recognition

- class간 분리성, 차별성의 다른 측면에 맞춘 'exclusive regularization'
- 가까운 이웃 사이의 각도에 페널티를 줌으로써 identity 사이의 거리를 명시적으로 유지
- 이를 위한 loss function들을 사용
- Resnet20 구조를 사용하여 얼굴 identity에 대한 feature map을 구함.
- 구한 feature map을 이용하여 recognition

RegularFace: Deep Face Recognition via Exclusive Regularization

CVPR 2019

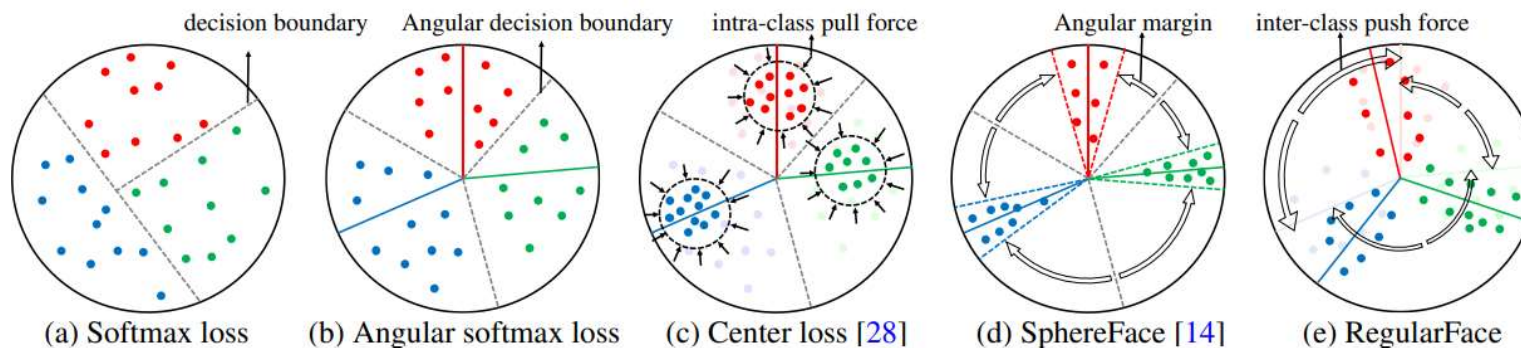


Figure 1. Illustration of face embeddings trained under various loss functions, points in color indicate different identities. (a) Softmax loss learns separable decision boundaries. (b) Angular softmax loss learns angularly separable decision boundaries. (c) *Center loss* [28] ‘pulls’ embeddings of the same identity towards their center, in order to obtain compact and discriminative representations. (d) *SphereFace* [14] (A-Softmax loss) proposes the ‘angular margin’ to clamp representations within a narrow angle. (e) Our proposed *RegularFace* introduces ‘inter-class push force’ that explicitly ‘pushes’ representations of different identities far way.

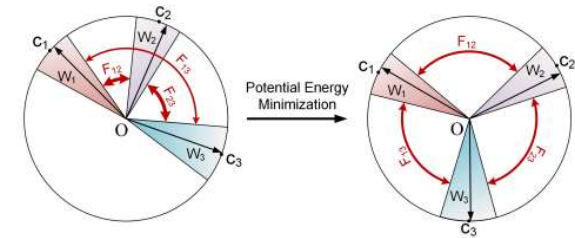
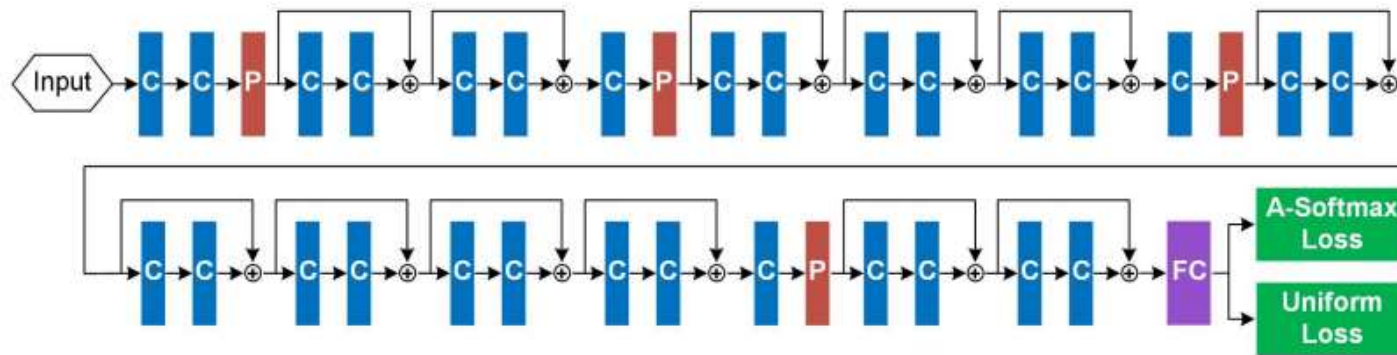
○ loss function

- softmax loss는 decision boundary를 결정
- Angular softmax loss는 중앙으로 부터의 각도 경계를 결정
- center loss는 identity가 같은 것끼리 중심으로 모아준다.(intra-class pull force)
- A-softmax loss는 각 클래스간 거리(각도)를 벌려준다.

UniformFace: Learning Deep Equidistributed Representation for Face Recognition

CVPR 2019

C: The convolution layer
P: The max-pooling layer
FC: The fully connected layer

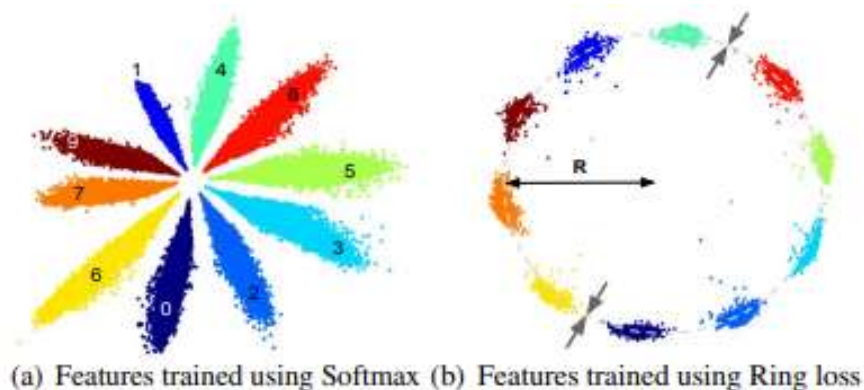


○ Recognition

- 얼굴 이미지를 CNN구조를 통해 fc layer의 feature map을 얻음
- 이 feature map을 이용해 A-softmax Loss와 Uniform Loss를 구함
- A-softmax에 의해 학습된 feature는 identity 사이의 angular distribution을 학습하는데 이때 유클리드 거리 대신 코사인 거리를 학습 -> inter class간 거리를 좁힘
- Uniform loss를 사용하여 각 클래스의 센터를 등분할 한다. -> 각기 다른 클래스의 거리가 균일하게 분포되게 해준다

Ring loss: Convex Feature Normalization for face Recognition

CVPR 2018



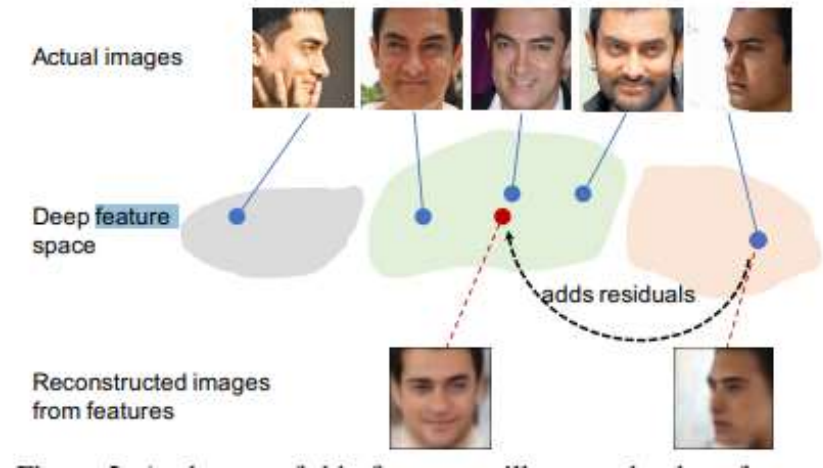
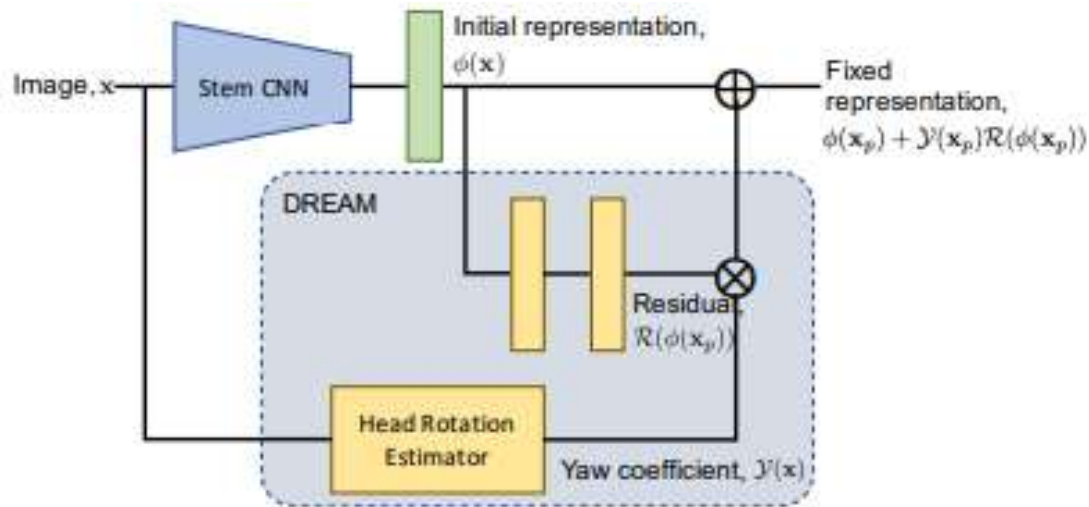
$$L_R = \frac{\lambda}{2m} \sum_{i=1}^m (\|\mathcal{F}(\mathbf{x}_i)\|_2 - R)^2$$

○ Ring loss for face recognition

- softmax의 기능을 증대하도록 설계된 Ring loss
- Res64 구조를 통과한 fc5 layer의 feature map을 이용해 loss를 계산
- $\mathcal{F}(\mathbf{x}_i)$ 가 input이미지를 넣었을 때 fc5 layer feature map이다.
- R 은 normalization의 기준이 되는 기준점으로써 이 점을 기준으로 원형으로 각 identity가 위치되게 된다.

Pose-Robust Face Recognition via Deep Residual Equivariant Mapping

CVPR 2018

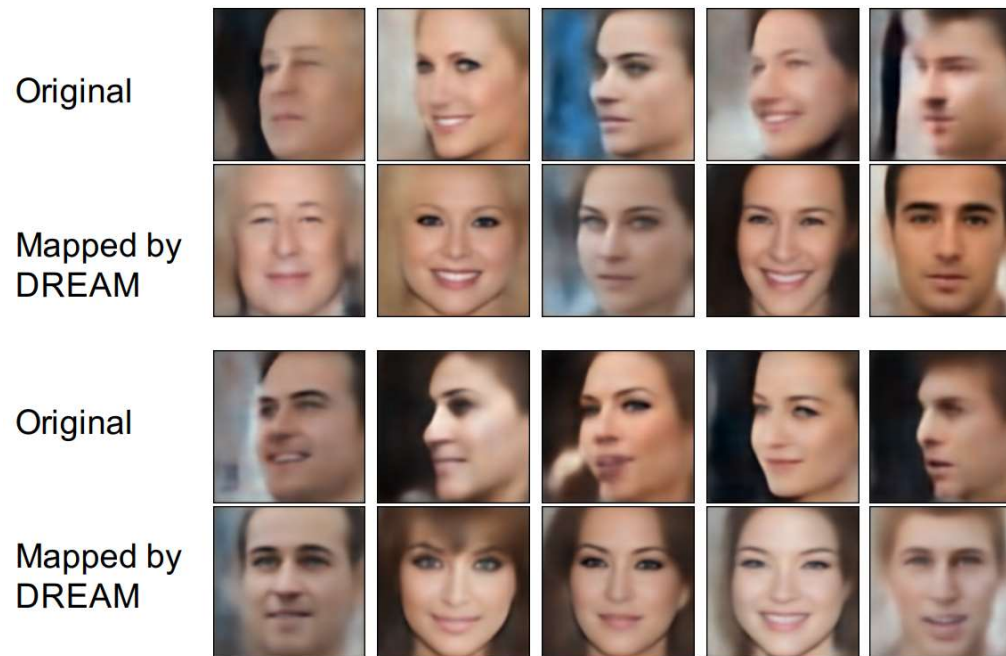


○ recognition

- 얼굴 정면과 종단면 사이에 고유 매핑이 존재한다고 가정
- 정면과 종단면 표정의 불일치에 의한 identity 불일치를 등변형 매핑에 의해 연결한다.
- stem CNN를 통과한 feature map을 사용
- 이미지를 dream block에 전달해 머리의 rotation을 측정하고 그에 해당하는 잔차(residual)를 feature map에 더해준다.
- 결과적으로 head pose에 invariant한 recognition이 가능하다.

Pose-Robust Face Recognition via Deep Residual Equivariant Mapping

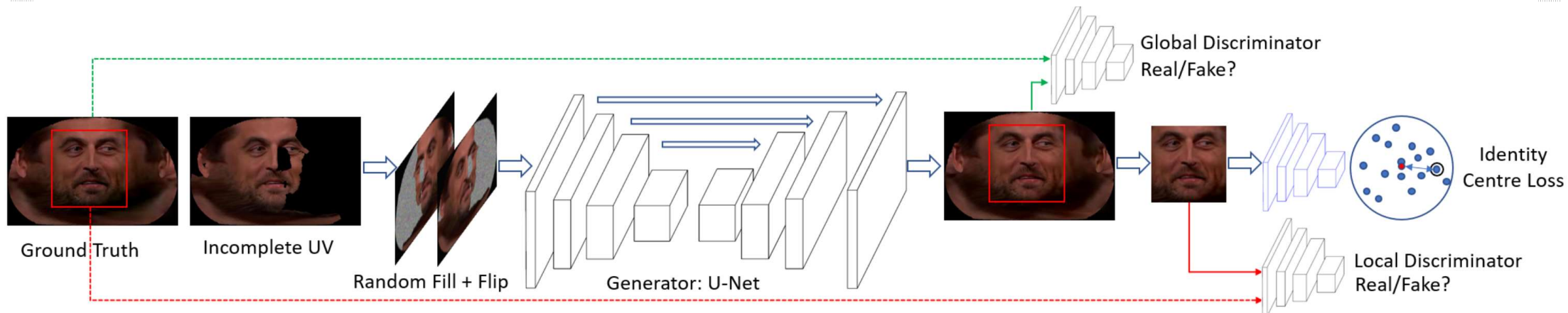
CVPR 2018



- 앞에서 보았던 embedded face를 추출하는 것과 같아 보인다.

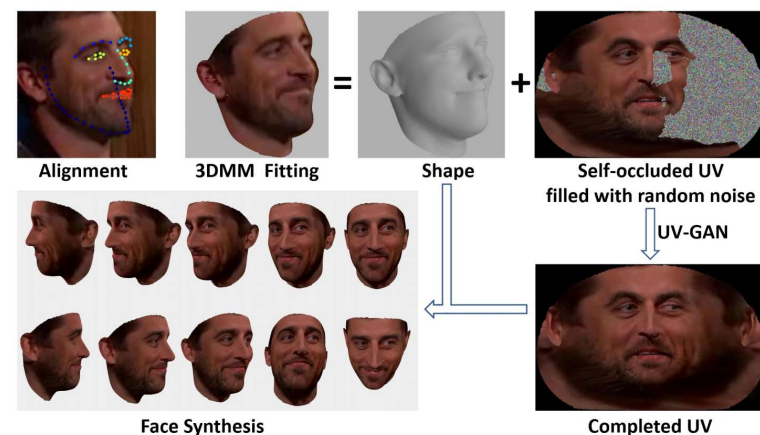
UV-GAN: Adversarial Facial UV Map Completion for Pose-invariant Face Recognition

CVPR 2018



○ Pose-invariant Face Recognition

- facial UV map을 생성해 recognition을 한다.
- UV란 3d 이미지를 2d로 펼쳐준 것
- 이미지를 3dMM에 fitting하고 이를 이용해 self-occluded UV를 만든다
- 만든 UV를 UV-GAN을 통해 completed UV로 만든다
- completed UV를 local Discriminator로 main face region을 찾는다.
- main face region을 ResNet-27구조를 적용해 얻은 feature map을 identity로 사용, recognition한다.



Identity in papers of face recognition

- CNN 구조 마지막 FC layer의 feature map을 identity로 사용
- feature map을 이용하고, 다른 요인에 영향을 덜 받도록 loss function를 고안하여 사용
- pose invariant한 recognition을 위해 head pose를 추정하여 그에 맞는 residue를 feature map에 더해주는 방법도 있다.
- 이미지를 받아 3dmm을 적용 한 후 UV map을 완성 시키고 UV map에서 local discriminator를 이용해 main face region을 찾는 방법도 있다.
- 몇개의 recognition 논문을 더 보았으나 전부 fc layer의 feature map을 이용하여 recognition을 진행했으며 어떤 loss, 구조, 방법을 쓰는 지에 따라 성능이 달라졌다.
- inter class는 좁게, intra class는 멀리 하지만 각 클래스간 거리는 최대한 균등하게 맞추려고 하였다.