FaceShifter: Towards High Fidelity And Occlusion Aware Face Swapping

정지헌

jeongjiheon.ai@gmail.com

Background

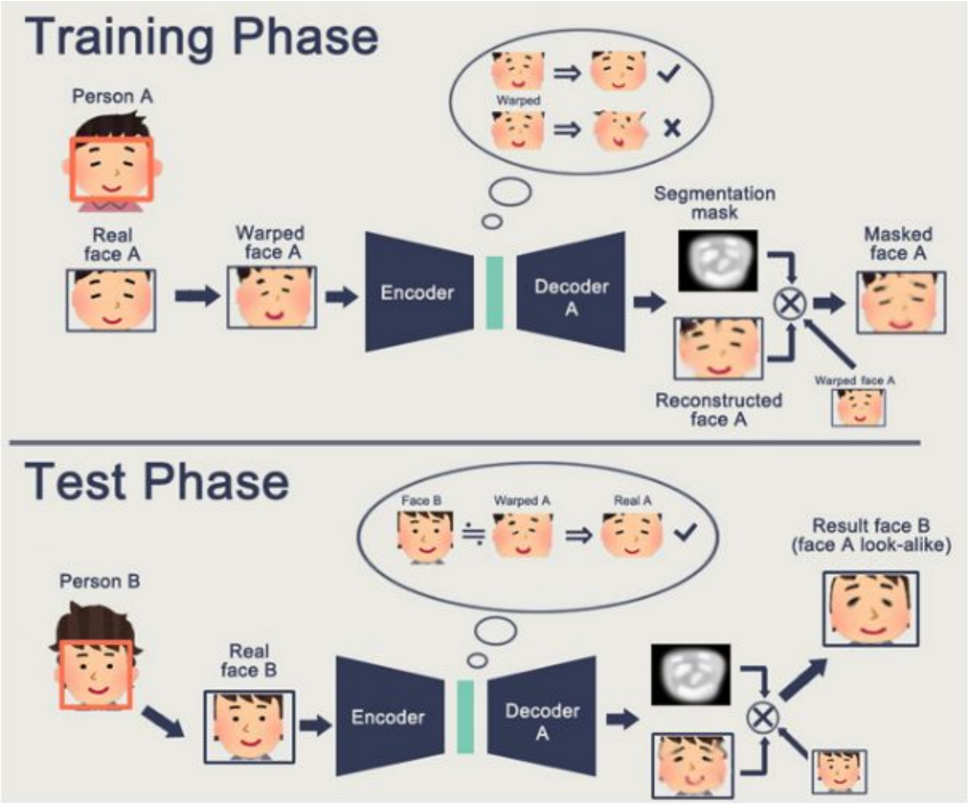
DeepFake Pipeline

1. Extraction : Face Detection (MTCNN, RetinaFace), Face Segmentation

2. Training : FaceSwapGAN, DeepFaceLab

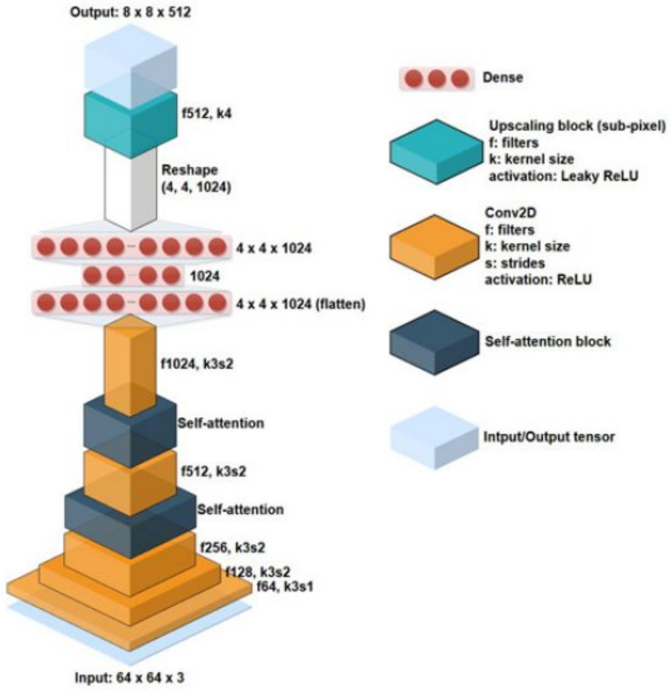
3. Conversion

FaceSwapGAN - pipeline

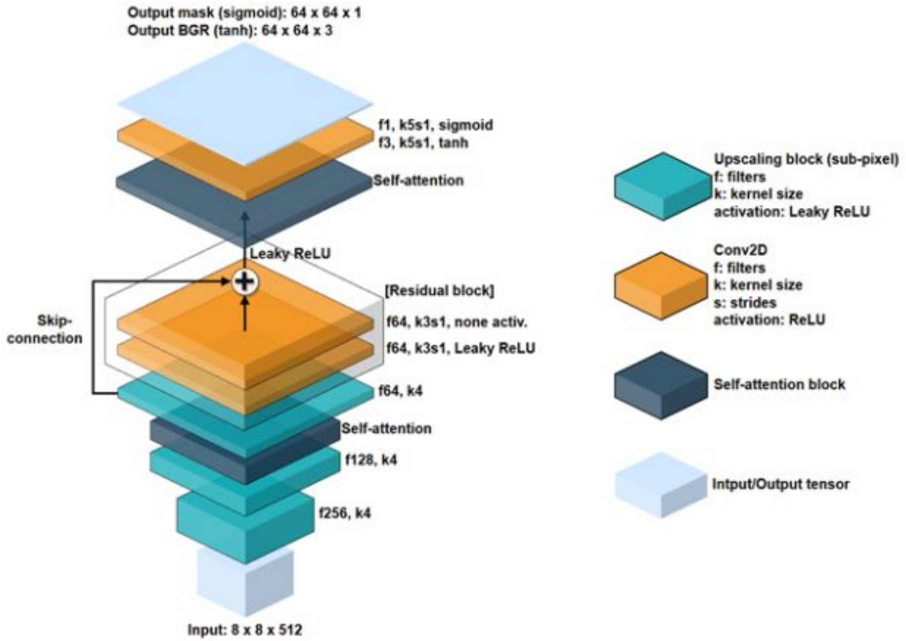


FaceSwapGAN - Encoder, Decoder

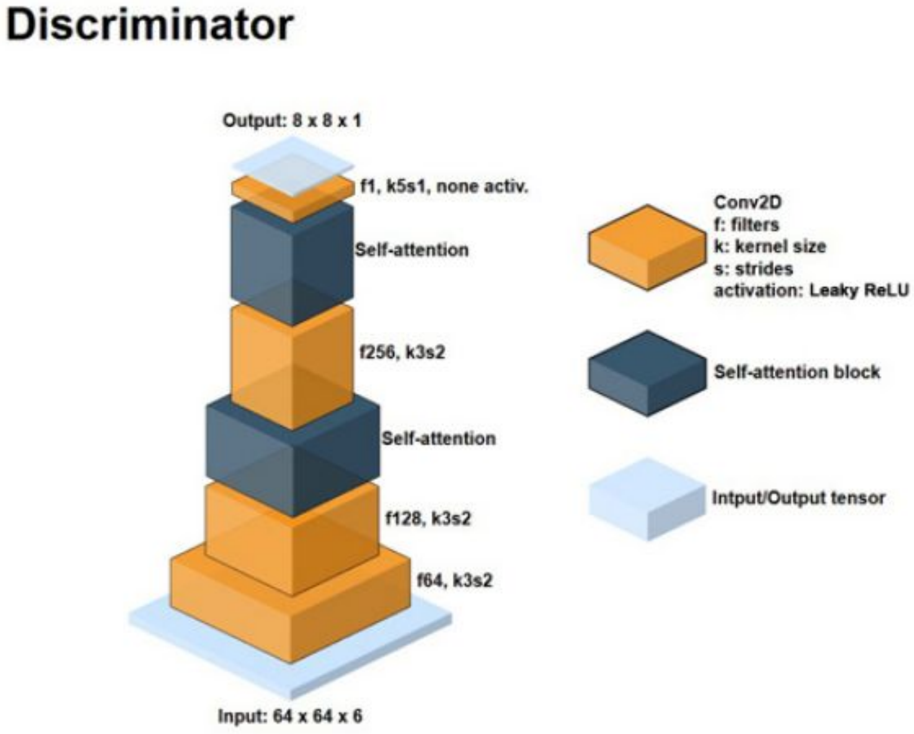
Encoder



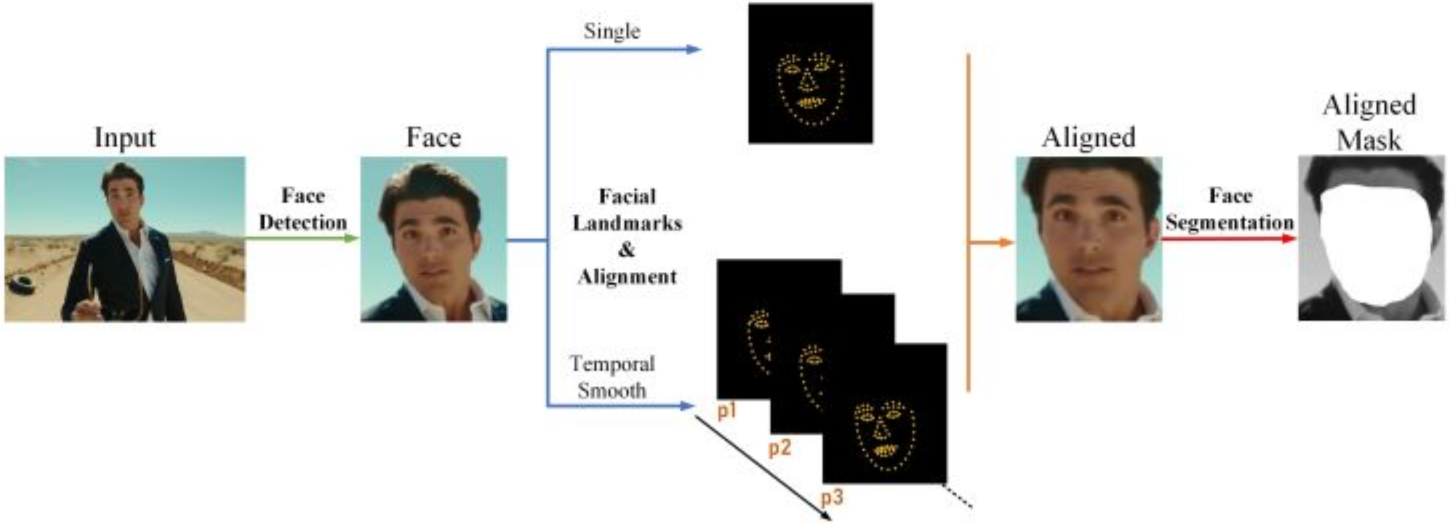
Decoder



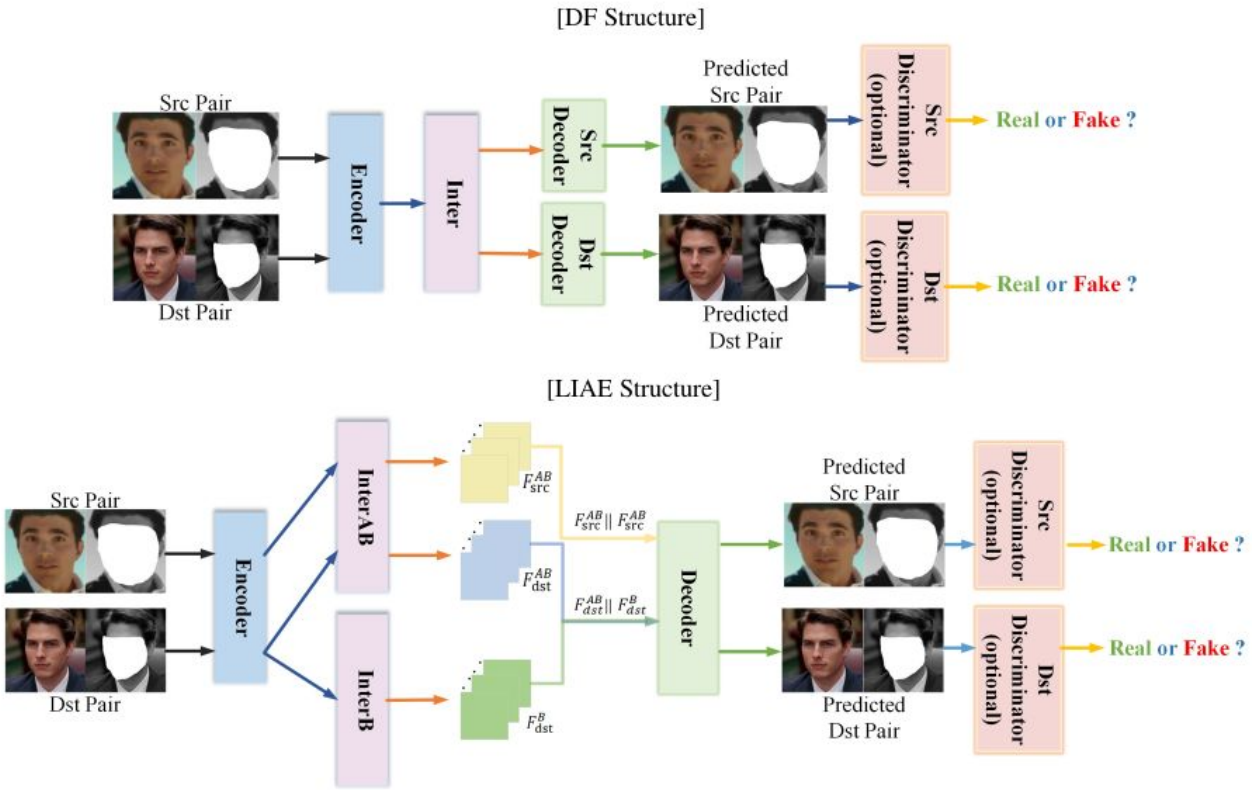
FaceSwapGAN - Discriminator



DeepFaceLab - pipeline



DeepFaceLab - pipeline



◦||◦ Concatenate two vector

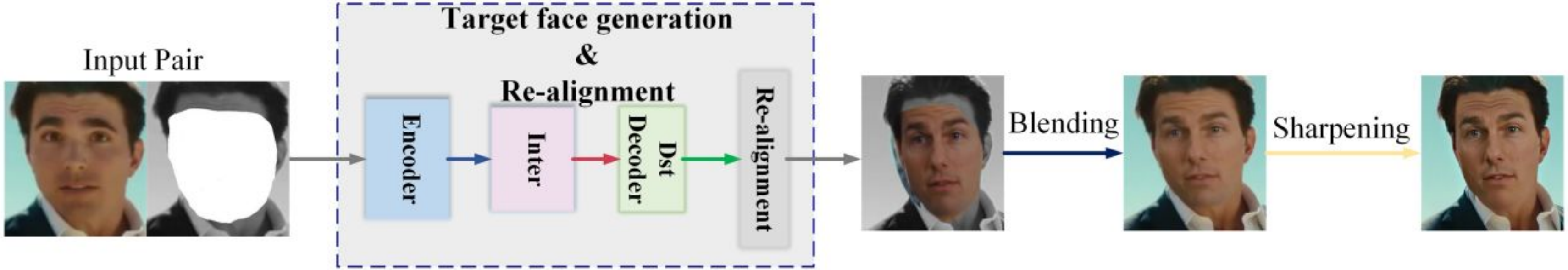
DeepFaceLab - Training

Training Loss

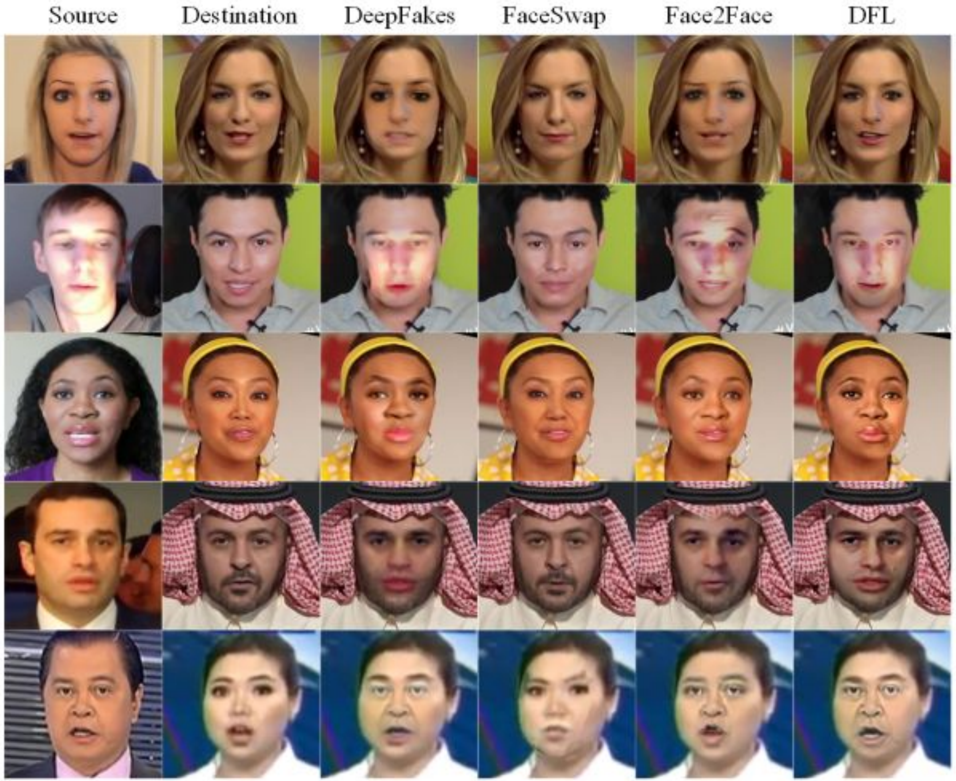
Loss = DSSIM + MSE
MSE : provides better clarity
DSSIM : generalize human faces faster

Training Loss (optional)

Adversarial Loss + Perceptual Loss



DeepFaceLab - Training



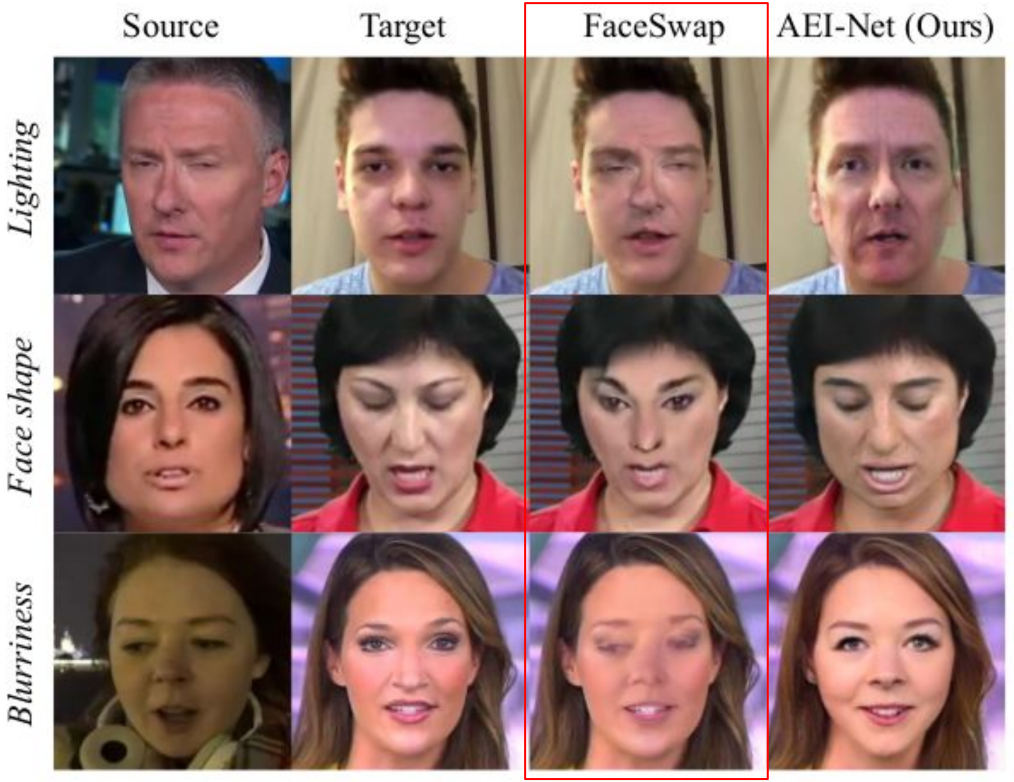
(a) The comparison of DFL and representative open-source face-swapping projects.

Previous Problem 1

How to extract identity and attribute?

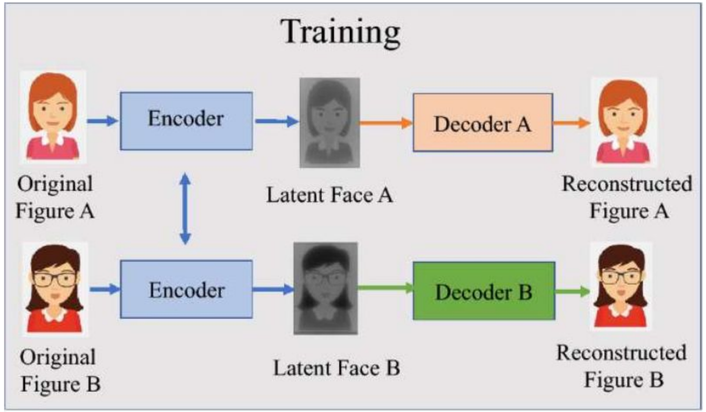
- Replacement-based work
  - 간단하게 안쪽 얼굴 영역의 pixel만 대체
- 3D-baed works
  - 3D 얼굴 구조의 accuracy & robust가 만족스럽지 않음.
- GAN-based works
  - Resalistic + High-Fidelity 는 여전히 challenge

Previous Problem 1

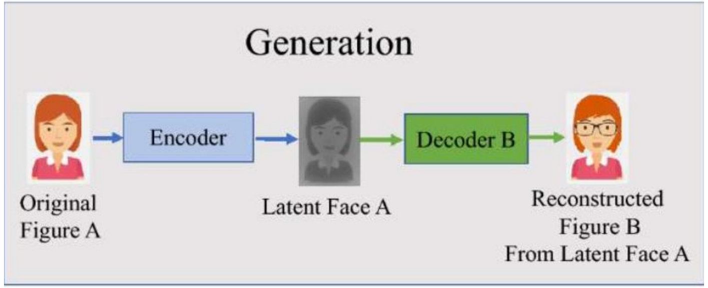


Mask를 이용해서 blend -> **Artifact**

Previous Problem 2



(a) Training Phase



(b) Generation Phase

인물마다의 Encoder / Decoder를 새로 학습시켜야함

Objective

1. **Mask, Landmark 와 같은 Align Extractor가 필요함**

   -> 특별한 Extractor 없이 End-to-End로 학습하기를 원함

2. **기존의 방식들은 Artifact가 생김 (Occlusion, blur, 등등)**

   -> Artiact 없이 생성되기를 원함.

3. **인물이 바뀔 때 마다 새로 학습해야함.**

   -> 한번에 학습하기를 원함.

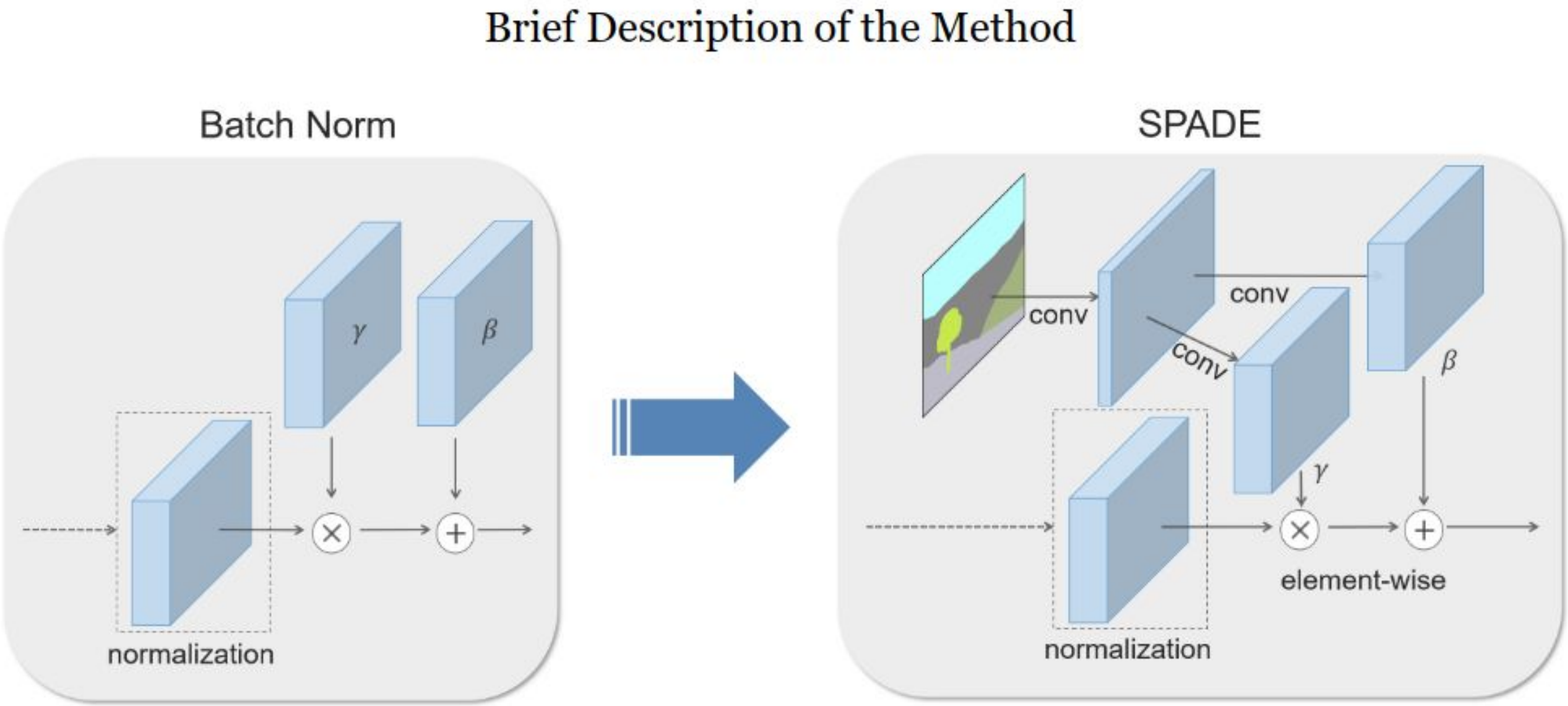
Background - AdaIN

$$\mathrm{AdaIN}(x, y) = \sigma(y)\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \mu(y)$$

Affine Parameter에 따라 특정 Transfer가 가능,

arbitrary affine parameters를 통해서 임의로 style transfer해보자.

Background - SPADE



Method

1. **Adaptive Embedding Integration Network (AEI-Net)**
   - 이미지 합성

2. **Heuristic Error Acknowledging Refinement Network (HEAR-Net)**
   - 이미지 정제

Adaptive Embedding Integration Network (AEI-Net)

1. **Identity Encoder**

   - Pretrained SOTA face recognition model as identity encoder (ArcFace)

2. **Multi-level Attributes Encoder**

   - Face attributes, such as pose, expression require more spatial information than identity Encoder

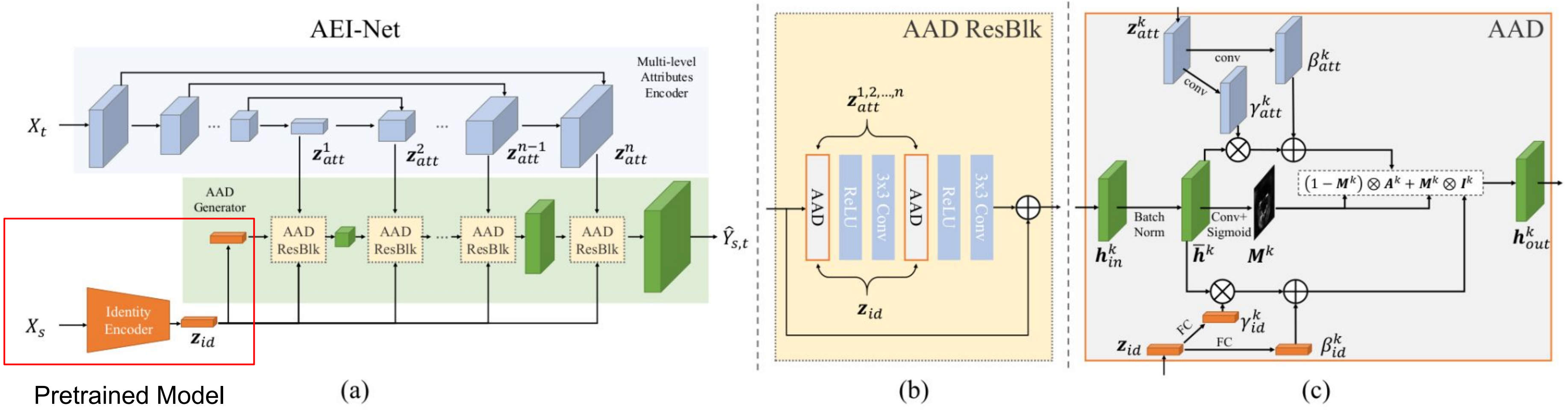
3. **Adaptive Attentional Denormalization Generator**

   - Identity & Attributes embedding을 조합

Adaptive Embedding Integration Network (AEI-Net)

1. Identity Encoder

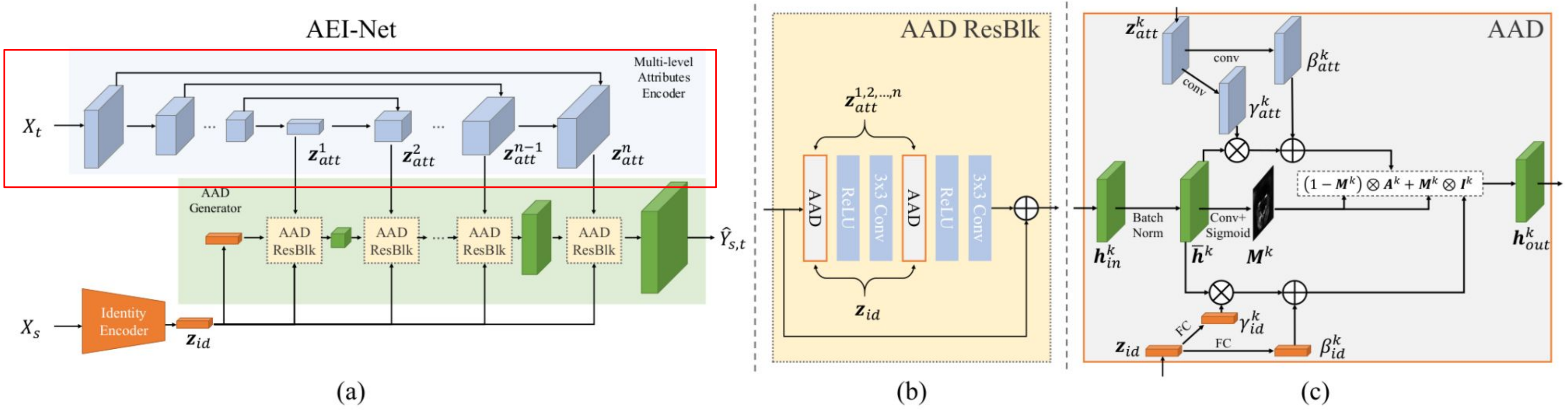
- Source Image Embedding
- 대량의 2D face data로 학습



Adaptive Embedding Integration Network (AEI-Net)

2. Multi-level Attributes Encoder

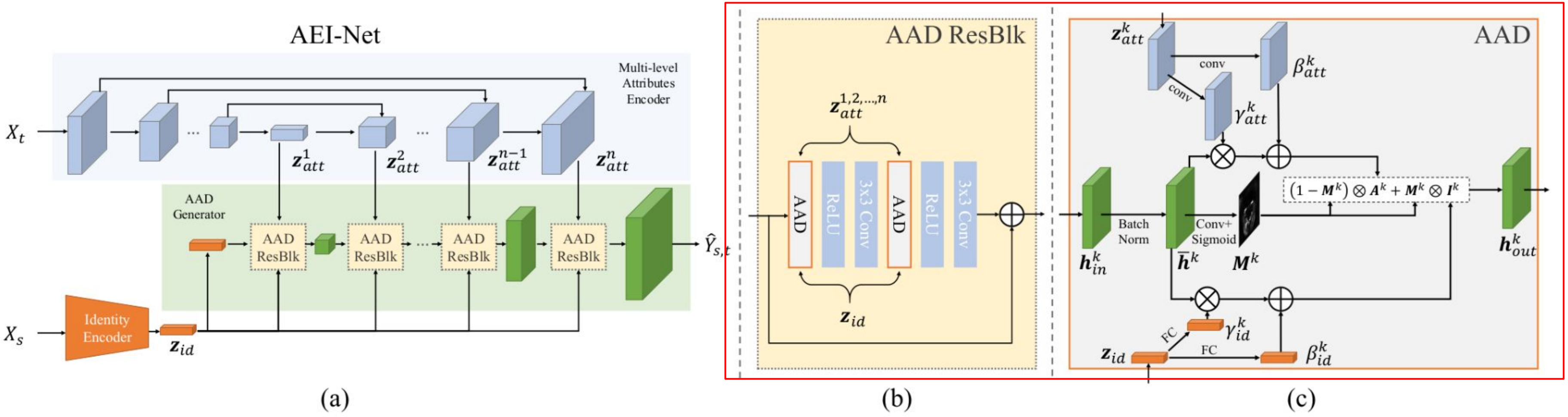
- 압축하여 Single Vector로 나타내는 Limitation
- Attributes의 디테일을 유지하기 위해 multi-level feature map을 적용
- U-Net Decoder에서 생성된 feature map들을 attributes embedding이라고 정의



Adaptive Embedding Integration Network (AEI-Net)

3. Adaptive Attentional Denormalization Generator (AAD)

- Identity & Attributes embedding을 조합해서 Image 생성
- Previous research에서는 단순히 concatenation하여 사용 -> blurry한 결과
- 그래서 SPADE와 AdaIN에서 착안하여 AAD Layer를 이용



Adaptive Embedding Integration Network (AEI-Net)

3. Adaptive Attentional Denormalization Generator (AAD)

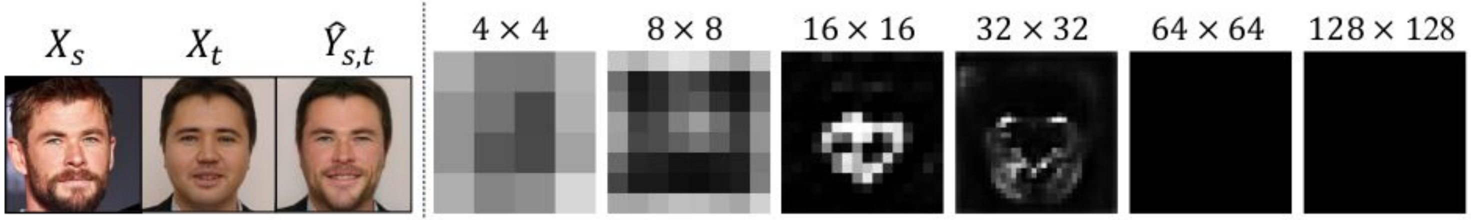
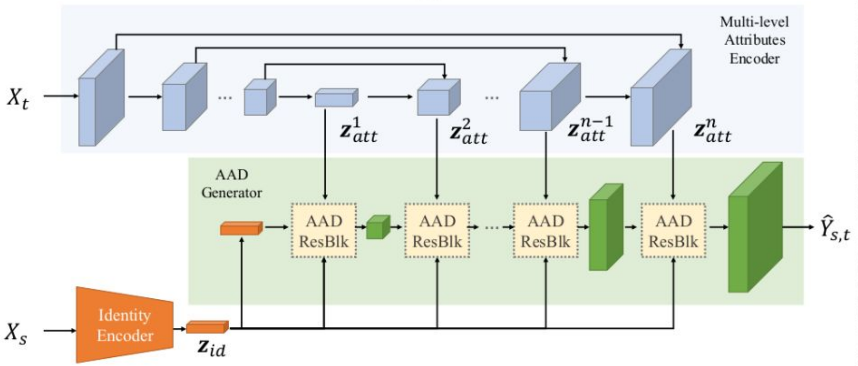


Figure 8: Visualizing attentional masks M^k of AAD layers on different feature levels. These visualizations reflect that identity embeddings are mostly effective in low and middle feature levels.

Adaptive Embedding Integration Network (AEI-Net)

Training

AEI-Net



$$\mathcal{L}_{id} = 1 - cos(z_{id}(\hat{Y}_{s,t}), z_{id}(X_s)),$$

$$\mathcal{L}_{att} = \frac{1}{2}\sum_{k=1}^{n}\left\|z_{att}^k(\hat{Y}_{s,t}) - z_{att}^k(X_t)\right\|_2^2.$$

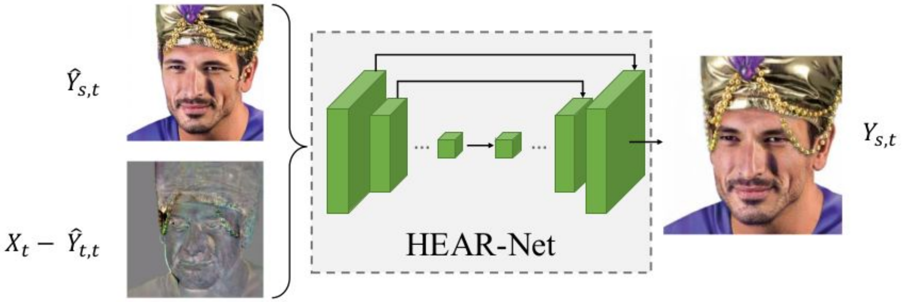
$$\mathcal{L}_{rec} = \begin{cases} \frac{1}{2}\left\|\hat{Y}_{s,t} - X_t\right\|_2^2 & if\ X_t = X_s \\ 0 & otherwise \end{cases}.$$

$$\mathcal{L}_{\text{AEI-Net}} = \mathcal{L}_{adv} + \lambda_{att}\mathcal{L}_{att} + \lambda_{id}\mathcal{L}_{id} + \lambda_{rec}\mathcal{L}_{rec},$$

Adaptive Embedding Integration Network (AEI-Net)

Result

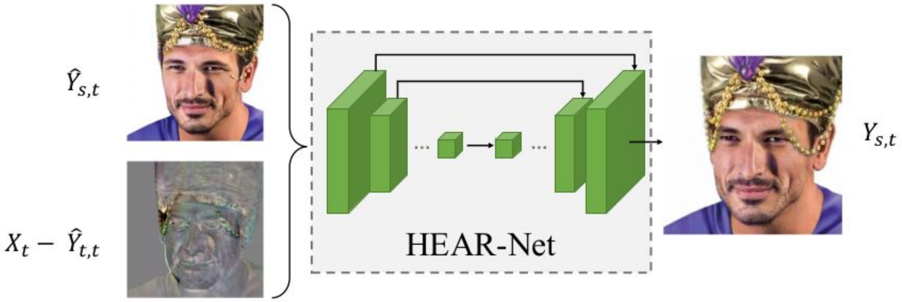
Heuristic Error Acknowledging Refinement Network



$$\hat{Y}_{t,t} = \texttt{AEI-Net}(X_t, X_t).$$

$$Y_{s,t} = \texttt{HEAR-Net}(\hat{Y}_{s,t}, \Delta Y_t).$$

Heuristic Error Acknowledging Refinement Network



$$\mathcal{L}'_{id} = 1 - cos(\boldsymbol{z}_{id}(Y_{s,t}), \boldsymbol{z}_{id}(X_s)).$$

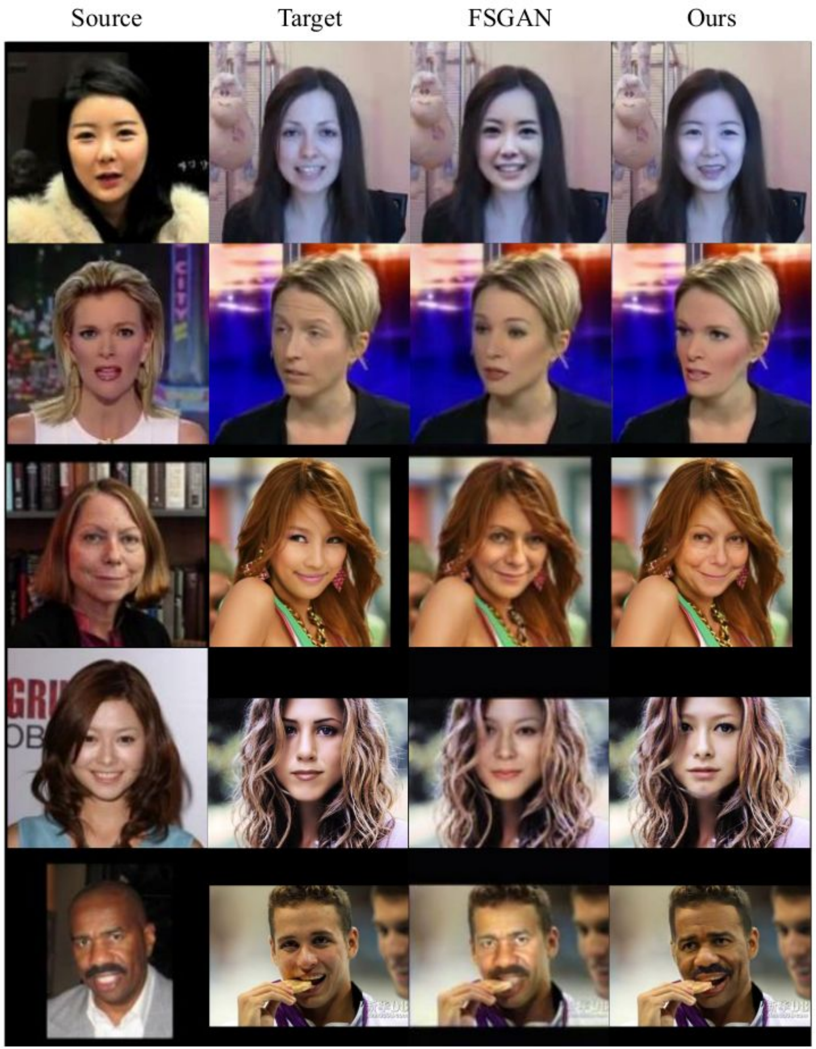
$$\mathcal{L}'_{chg} = \left|\hat{Y}_{s,t} - Y_{s,t}\right|.$$

$$\mathcal{L}'_{rec} = \begin{cases} \frac{1}{2}\left\|Y_{s,t} - X_t\right\|_2^2 & \mathit{if}\ X_t = X_s \\ 0 & \mathit{otherwise} \end{cases}$$

Training Strategy

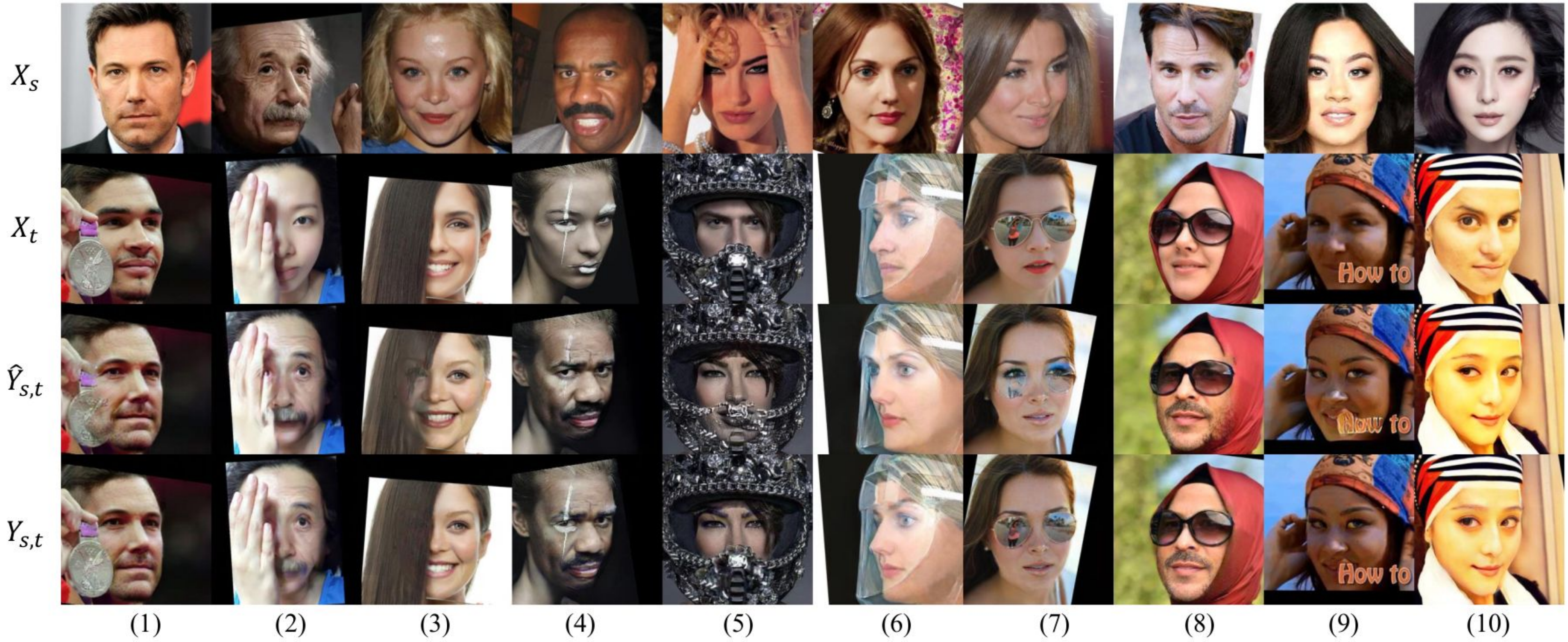
Figure 13: Augmentation with synthetic occlusions.

Result



Result

Result



Result

Figure 11: Our face swapping results on wild face images under various challenging conditions. All results are generated using a single well-trained two-stage model.

Result

method	id.	attr.	realism
DeepFakes [1]	13.7	6.8	6.1
FaceSwap [2]	12.1	23.7	6.8
Nirkin *et al*. [31]	21.3	7.4	4.2
Ours	**52.9**	**62.1**	**82.9**

Table 2: User study results. We show the averaged selection percentages of each method.

1. The one having the most similar identity with the source face
2. The one sharing the most similar head pose, face expression and scene lighting with the target image
3. The most realistic one