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

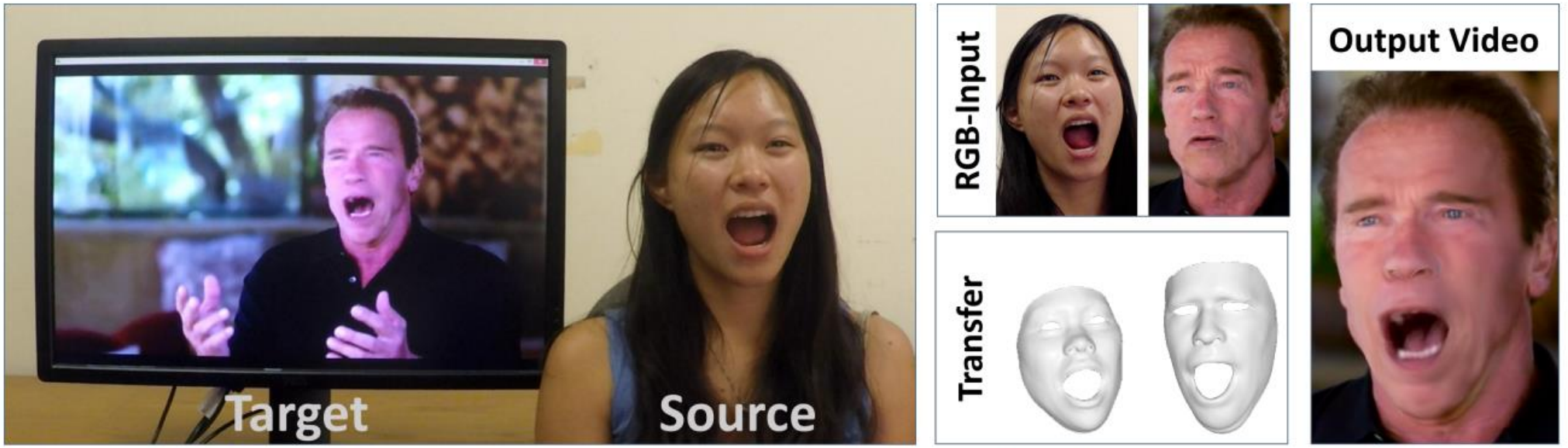
Hyperconnect
AAAI 2020

석사과정 김 진용

AI Lab. ARTIFICIAL INTELLIGENCE LAB.

Background

Face Reenactment 란?



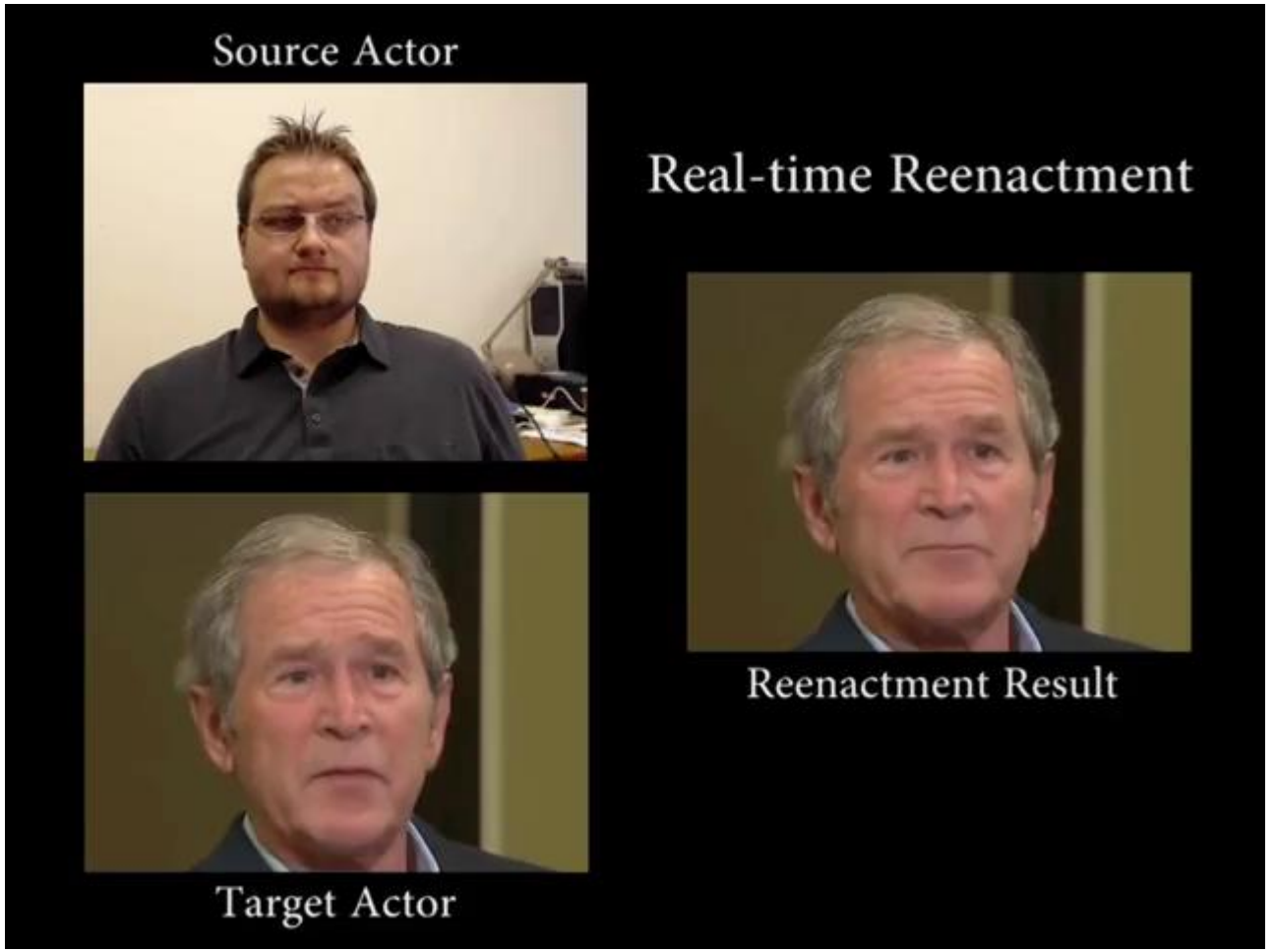
Proposed online reenactment setup: a monocular target video sequence (e.g., from Youtube) is reenacted based on the expressions of a source actor who is recorded live with a commodity webcam.

Source Video의 얼굴을 그대로 Target Video가 Identity를 잃지 않고 재연하는(Reenactment) 것

AI Lab. ARTIFICIAL INTELLIGENCE LAB.

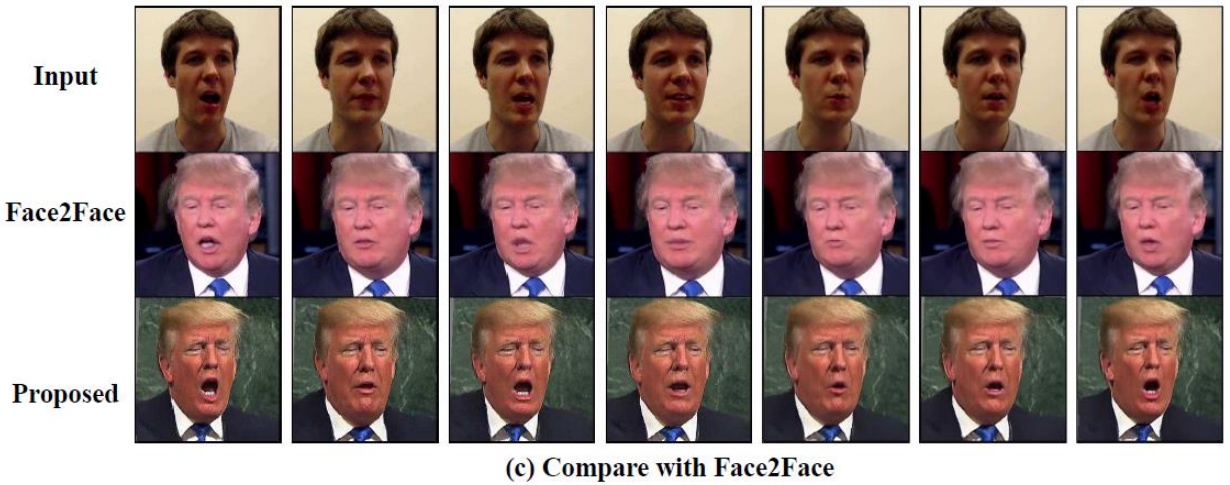
Face2Face: Real-time Face Capture and Reenactment of RGB Videos.
CVPR 2016 Thies et al

Face Reenactment 란?



FSGAN: Subject Agnostic Face Swapping and Reenactment, ICCV 2019
FaceSwapNet: Landmark Guided Many-to-Many Face Reenactment, preprint
ReenactGAN: Learning to Reenact Faces via Boundary Transfer, ECCV 2018

Introduction



(c) Compare with Face2Face

ReenactGAN: Learning to Reenact Faces via Boundary Transfer, W Wu et al

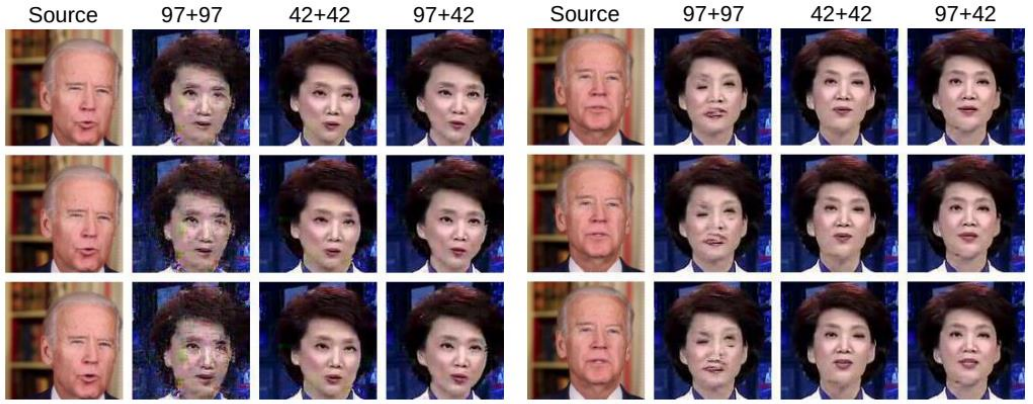
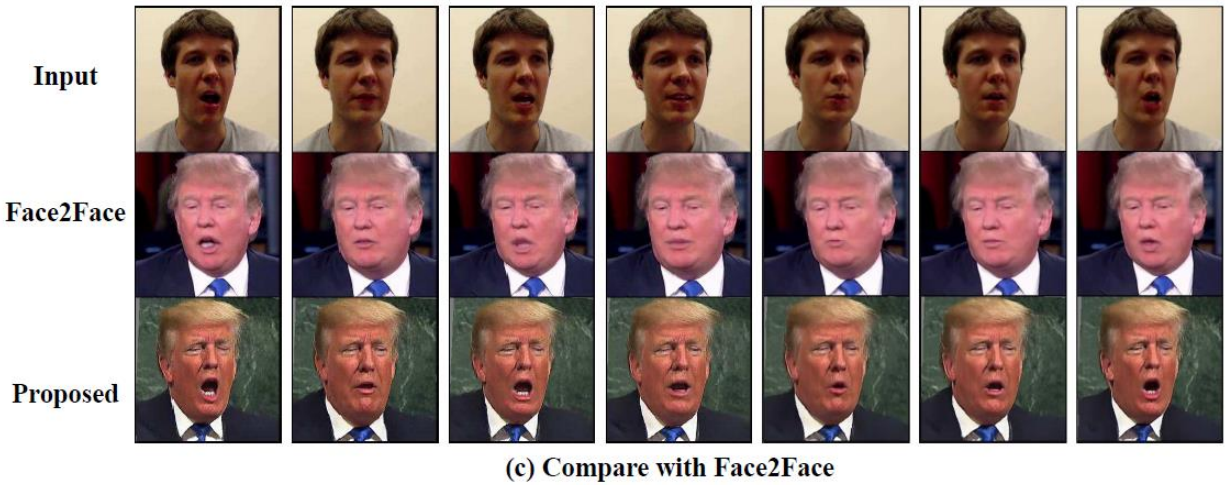


Figure 2: The first column are three frames in succession. 97 + 97 model fails to generate clear images. 42+42 model generates much better images but results in abrupt deformation. 97 + 42 model alleviates the deformation and achieves images close to real images.

Figure 3: The first column are three source frames in succession. The other columns are results of different models. 97+97 model generates distorted faces. 42 + 42 and 97 + 42 model generate similar result.

Face Transfer with Generative Adversarial Network , R Xu et al

최근 CycleGAN을 base로 Face Reenactment를 연구한 논문들이 많이 있다.



(c) Compare with Face2Face

ReenactGAN: Learning to Reenact Faces via Boundary Transfer, W Wu et al

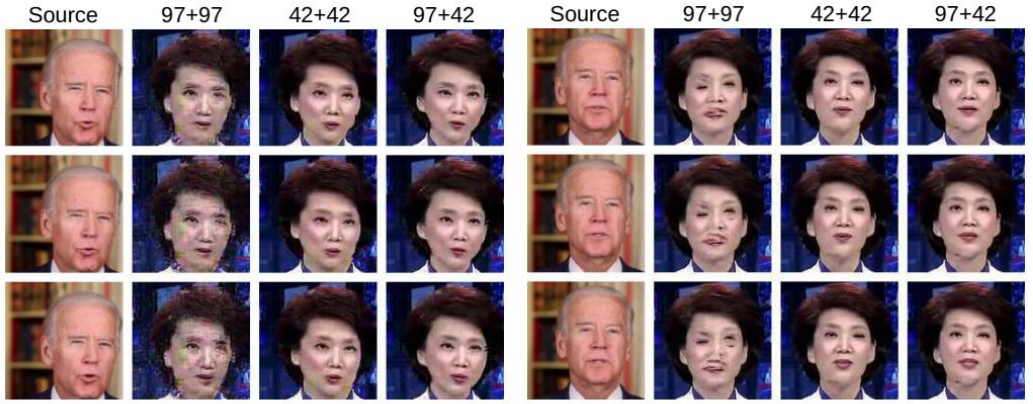


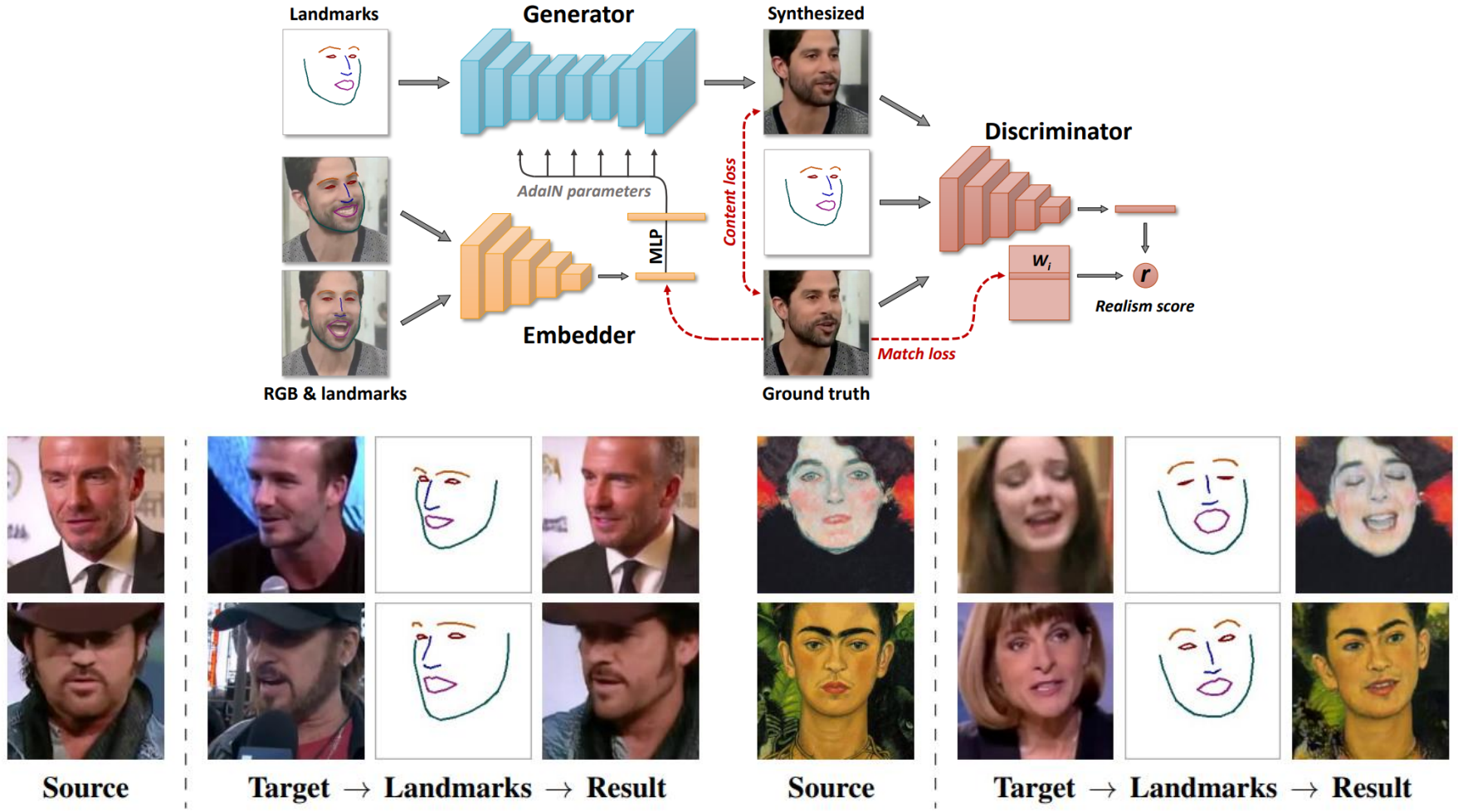
Figure 2: The first column are three frames in succession. 97 + 97 model fails to generate clear images. 42+42 model generates much better images but results in abrupt deformation. 97 + 42 model alleviates the deformation and achieves images close to real images.

Figure 3: The first column are three source frames in succession. The other columns are results of different models. 97+97 model generates distorted faces. 42 + 42 and 97 + 42 model generate similar result.

Face Transfer with Generative Adversarial Network , R Xu et al

그러나 이 연구들의 단점은
- 최소 몇분 정도의 데이터(비디오 데이터)가 필요하다
- Input에 대한 특정 Identity 밖에 인식하지 못한다.

=> 이것들은 Wild 한 환경(real world) 에서 매력적이지 못한 조건들이다.



Few-Shot Adversarial Learning of Realistic Neural Talking Head Models , Zakharov et al

이러한 단점들을 극복하기위해 Few-shot learning을 통한 Reenactment가 연구되어왔다.

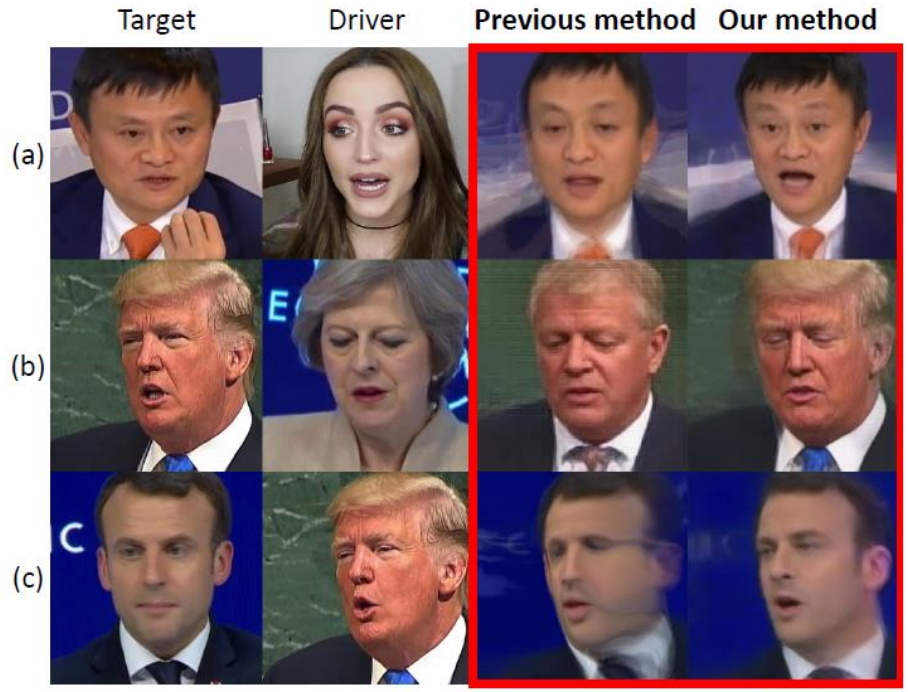
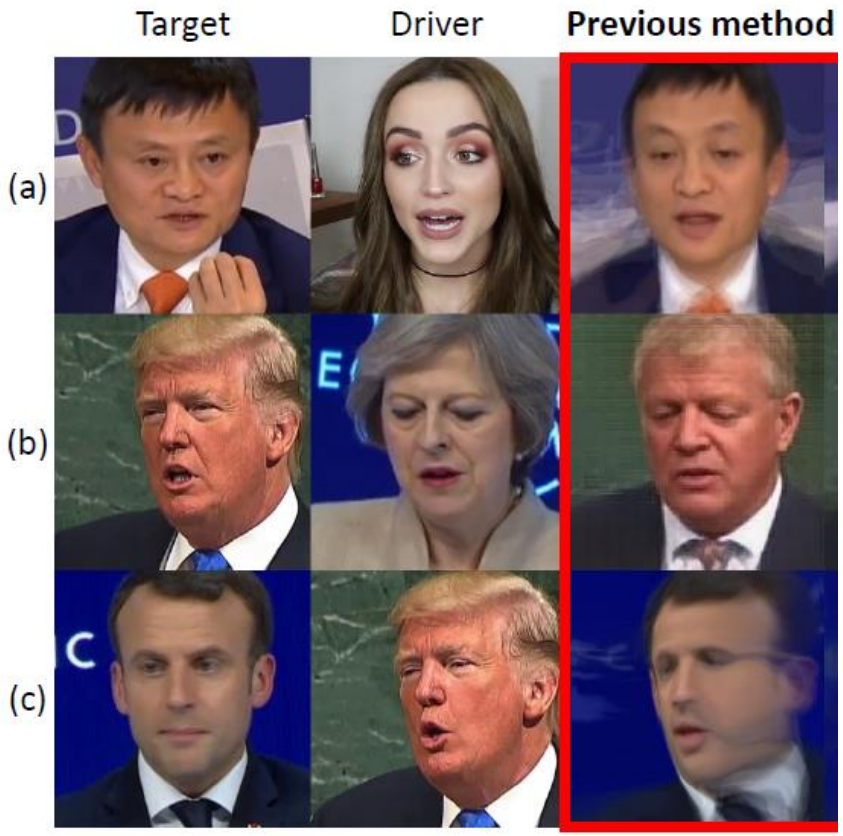


Figure 1: Examples of identity preservation failures and improved results generated by the proposed method. Each row shows (a) driver shape interference, (b) losing details of target identity, and (c) failure of warping at large poses.

Few-shot의 SoTA에서 또한 문제점을 발견했는데, 이를 **Identity preservation problem**으로 정의한다
이는 말 그대로 Target의 Identity를 소실한다는 것.



3가지의 Failure를 발견할 수 있는데,

(a) Identity mismatch를 무시하게되면 Driver가 Target의 합성을 방해한다.

(b) Identity 정보를 보존하기 위한 compressed vector representation(e.g AdaIN)의 허가량(용량)이 충분하지 않으면 detail을 잃게된다.

(c) 큰 포즈(표정)를 처리할때 결함 발생

-> 이를 해결하기 위한 **MarioNETte** model 제안

MarioNETte

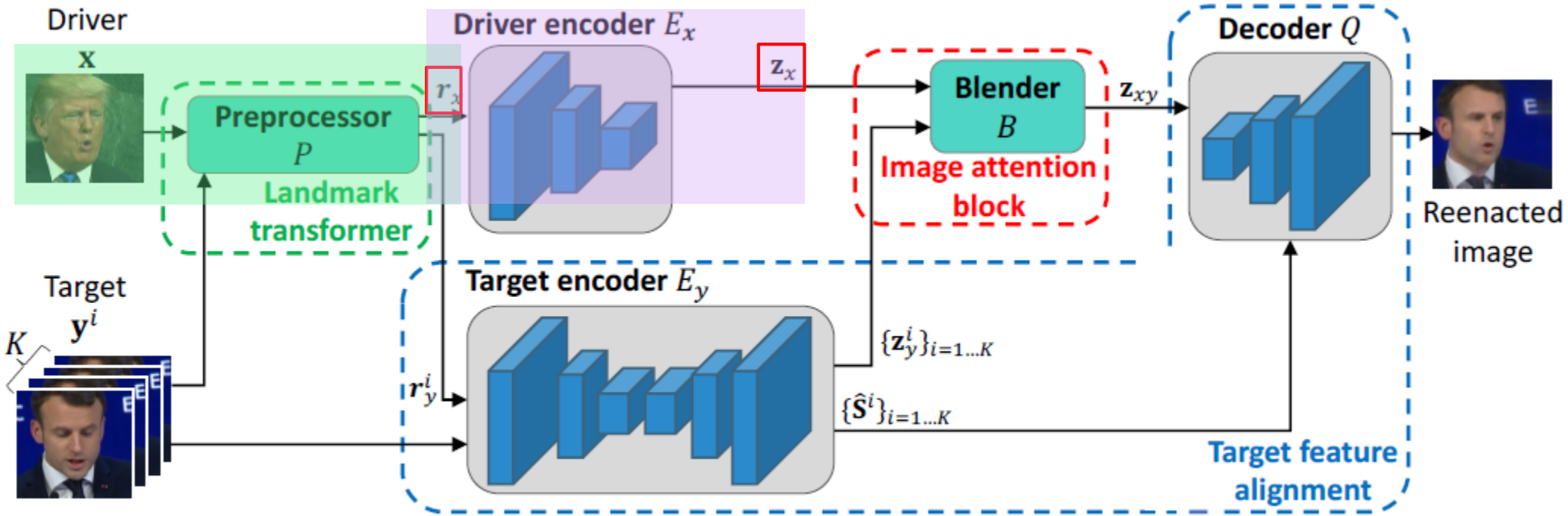


Figure 2: The overall architecture of MarioNETte.

$$\mathbf{r}_x = P(\mathbf{x})$$

Input Image로 Landmark 생성 및 transform

$$E_x(\mathbf{r}_x)$$

r_x를 입력받아 Driver Feature map z_x 생성

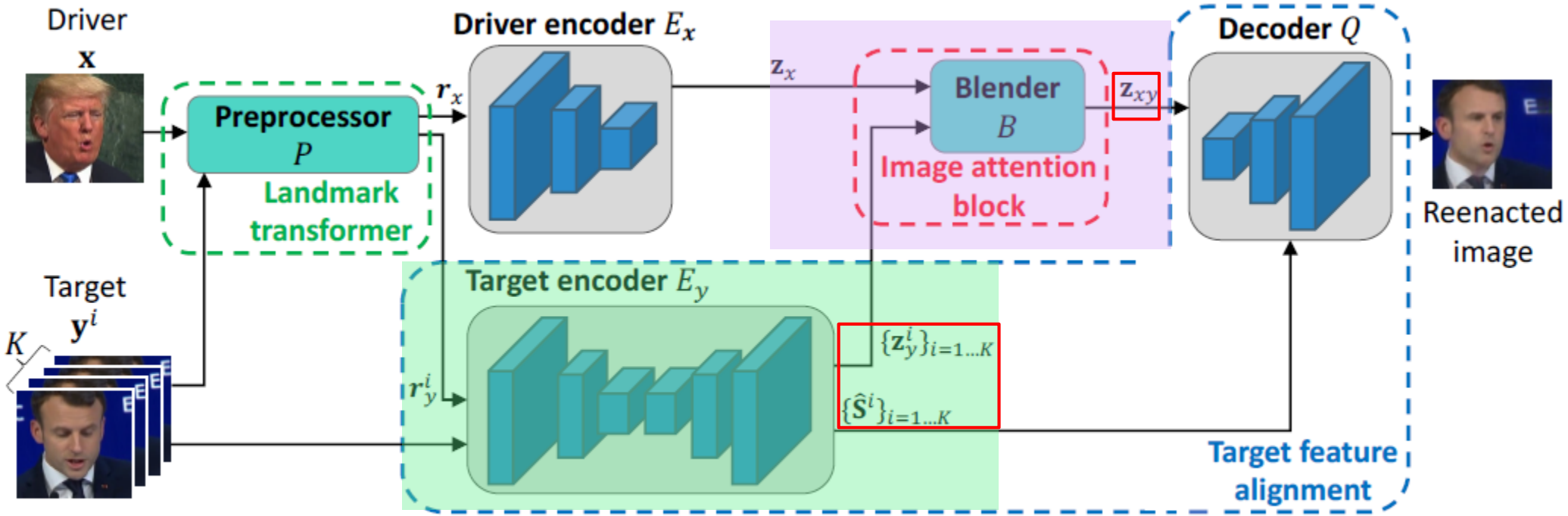


Figure 2: The overall architecture of MarioNETte.

$$E_y(\mathbf{y}, \mathbf{r}_y)$$

Target Encoder에 target image y와 landmark r_y를 넣어 스타일 정보를 추출하여 워핑된 target feature map S'(landmark)
target feature map z_y(Style) 를 생성해낸다.

$$B(\mathbf{z}_x, \{\mathbf{z}_y^i\}_{i=1...K})$$

Driver의 style과 target의 style을 입력으로 받음

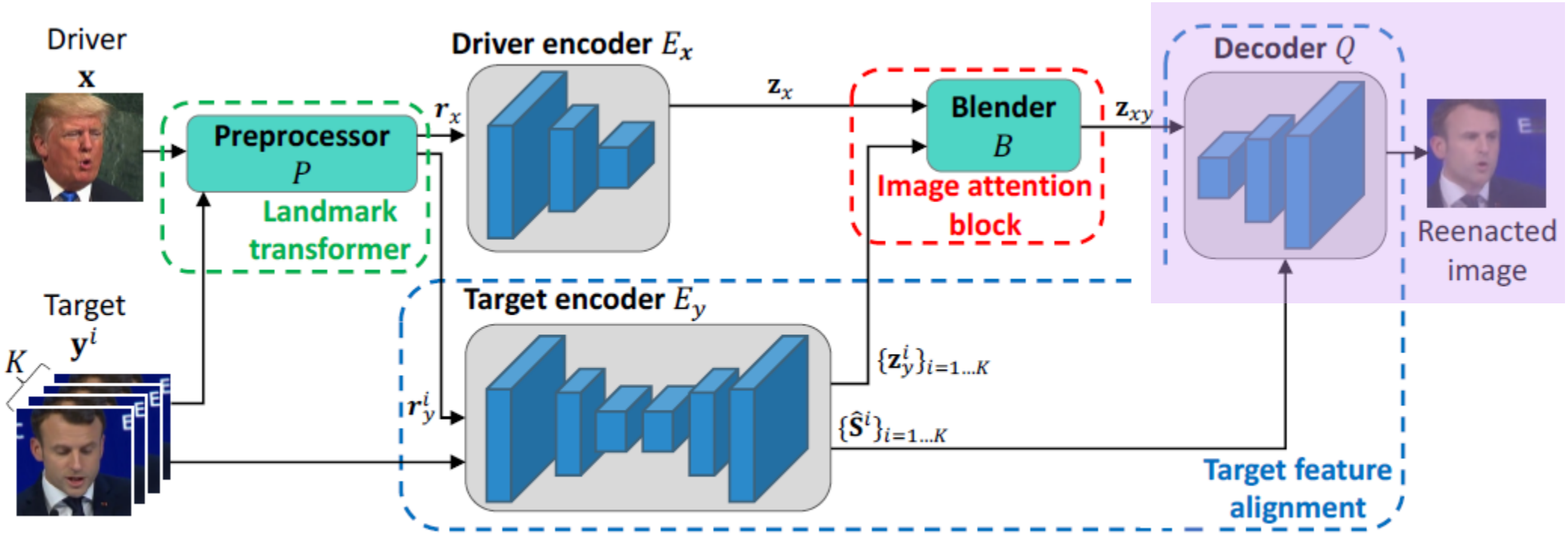


Figure 2: The overall architecture of MarioNETte.

$$Q(\mathbf{z}_{xy}, \{\mathbf{S}^i\}_{i=1...K})$$

Warp되어 비틀어진 S'과 z_xy를 입력으로 받아 reenacted image 생성
비틀어진-> alignment한

Main Technique : Image attention block

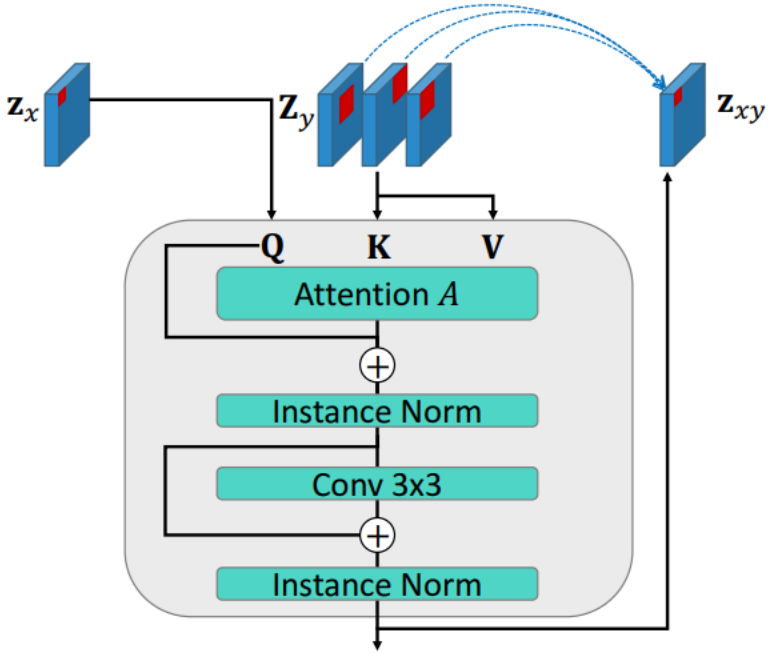


Figure 3: Architecture of the image attention block. Red boxes conceptually visualize how each position of $\mathbf{z}_x$ and $\mathbf{Z}_y$ are associated. Our attention can attend different position of each target feature maps with different importance.

- Driver feature map을 Attention query로 (요청),
  Target feature map을 Attention memory로 사용한다.(저장)

- Attention Mechanism으로 각 feature의 특정 position에 집중

Main Technique : Image attention block

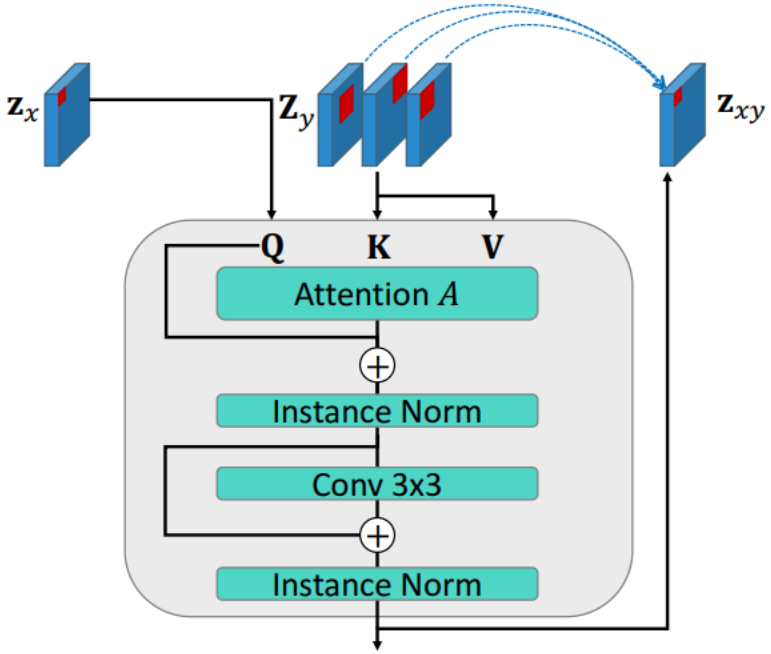


Figure 3: Architecture of the image attention block. Red boxes conceptually visualize how each position of $\mathbf{z}_x$ and $\mathbf{Z}_y$ are associated. Our attention can attend different position of each target feature maps with different importance.

- 입력으로 들어가는 z_x와 z_yi들은 다음과 같다

$$\mathbf{z}_x \in \mathbb{R}^{h_x \times w_x \times c_x}$$

$$\mathbf{Z}_y = [\mathbf{z}_y^1, \ldots, \mathbf{z}_y^K] \in \mathbb{R}^{K \times h_y \times w_y \times c_y}$$

Main Technique : Image attention block

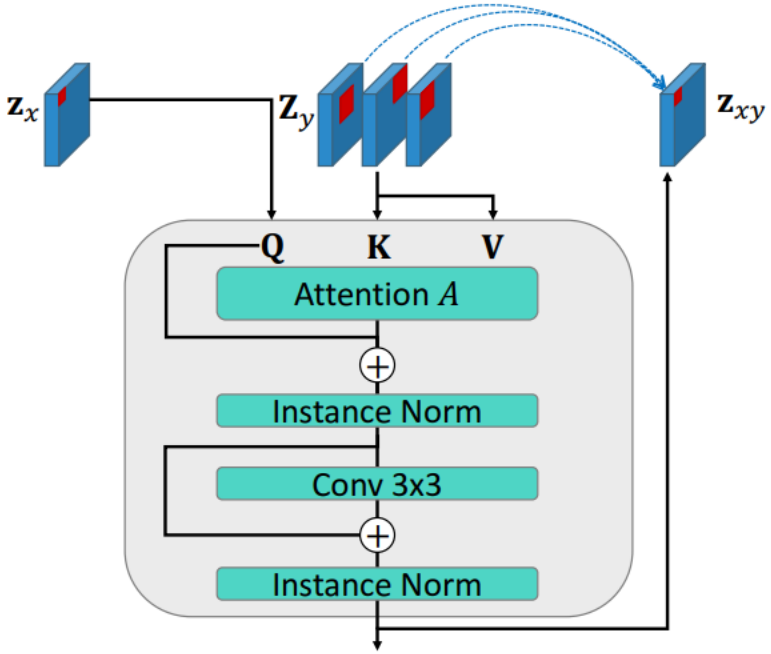


Figure 3: Architecture of the image attention block. Red boxes conceptually visualize how each position of $\mathbf{z}_x$ and $\mathbf{Z}_y$ are associated. Our attention can attend different position of each target feature maps with different importance.

- 이 입력값을 이용해 다음과같이 3개의 연산을 통해 Attention 진행

$$
\begin{aligned}
\mathbf{Q} &= \mathbf{z}_x \mathbf{W}_q + \mathbf{P}_x \mathbf{W}_{qp} && \in \mathbb{R}^{h_x \times w_x \times c_a} \\
\mathbf{K} &= \mathbf{Z}_y \mathbf{W}_k + \mathbf{P}_y \mathbf{W}_{kp} && \in \mathbb{R}^{K \times h_y \times w_y \times c_a} \\
\mathbf{V} &= \mathbf{Z}_y \mathbf{W}_v && \in \mathbb{R}^{K \times h_y \times w_y \times c_x}
\end{aligned} \tag{1}
$$

$$
A(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{f(\mathbf{Q}) f(\mathbf{K})^T}{\sqrt{c_a}}\right) f(\mathbf{V}), \tag{2}
$$

Main Technique : Image attention block

$$
\begin{aligned}
\mathbf{Q} &= \mathbf{z}_x \mathbf{W}_q + \mathbf{P}_x \mathbf{W}_{qp} && \in \mathbb{R}^{h_x \times w_x \times c_a} \\
\mathbf{K} &= \mathbf{Z}_y \mathbf{W}_k + \mathbf{P}_y \mathbf{W}_{kp} && \in \mathbb{R}^{K \times h_y \times w_y \times c_a} \\
\mathbf{V} &= \mathbf{Z}_y \mathbf{W}_v && \in \mathbb{R}^{K \times h_y \times w_y \times c_x}
\end{aligned} \tag{1}
$$

$$
A(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{f(\mathbf{Q}) f(\mathbf{K})^T}{\sqrt{c_a}} \right) f(\mathbf{V}), \tag{2}
$$

$$
f \;:\; \mathbb{R}^{d_1 \times \ldots \times d_k \times c} \;\rightarrow\; \mathbb{R}^{(d_1 \times \ldots \times d_k) \times c}
$$

- 입력값을 One-demension으로 바꾸기 위한 flattening function
- (-1, c) 값으로 flatten

AI Lab. ARTIFICIAL INTELLIGENCE LAB.

Main Technique : Image attention block

$$
\begin{aligned}
\mathbf{Q} &= \mathbf{z}_x \mathbf{W}_q + \mathbf{P}_x \mathbf{W}_{qp} && \in \mathbb{R}^{h_x \times w_x \times c_a} \\
\mathbf{K} &= \mathbf{Z}_y \mathbf{W}_k + \mathbf{P}_y \mathbf{W}_{kp} && \in \mathbb{R}^{K \times h_y \times w_y \times c_a} \qquad (1) \\
\mathbf{V} &= \mathbf{Z}_y \mathbf{W}_v && \in \mathbb{R}^{K \times h_y \times w_y \times c_x}
\end{aligned}
$$

$$
A(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{f(\mathbf{Q}) f(\mathbf{K})^T}{\sqrt{c_a}} \right) f(\mathbf{V}), \qquad (2)
$$

- W는 마지막 차원에서 특정 수의 채널에 매핑하기 위한 linear projection matrices

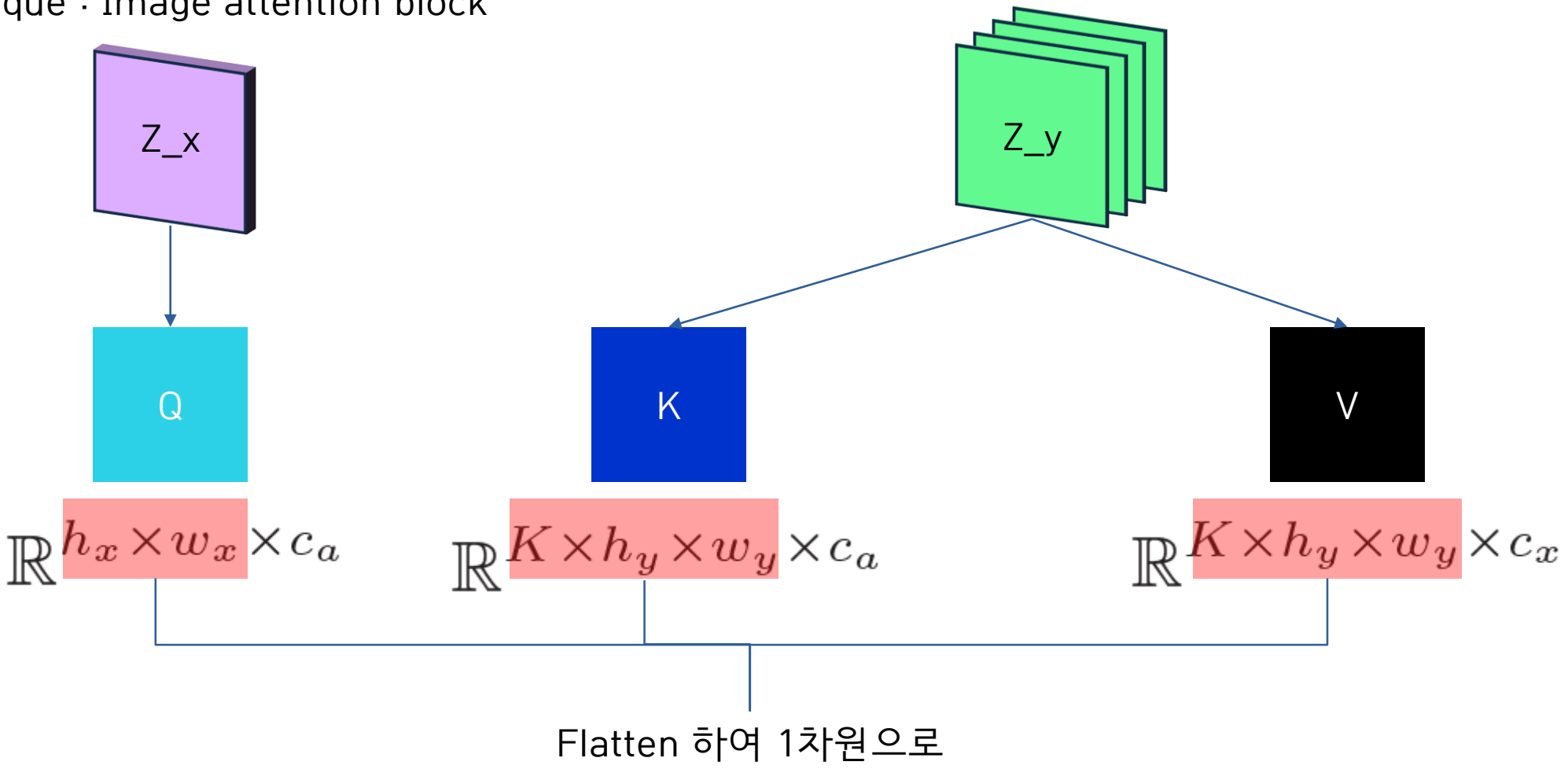
Main Technique : Image attention block

$$
\begin{aligned}
\mathbf{Q} &= \mathbf{z}_x \mathbf{W}_q + \mathbf{P}_x \mathbf{W}_{qp} && \in \mathbb{R}^{h_x \times w_x \times c_a} \\
\mathbf{K} &= \mathbf{Z}_y \mathbf{W}_k + \mathbf{P}_y \mathbf{W}_{kp} && \in \mathbb{R}^{K \times h_y \times w_y \times c_a} \\
\mathbf{V} &= \mathbf{Z}_y \mathbf{W}_v && \in \mathbb{R}^{K \times h_y \times w_y \times c_x}
\end{aligned} \tag{1}
$$

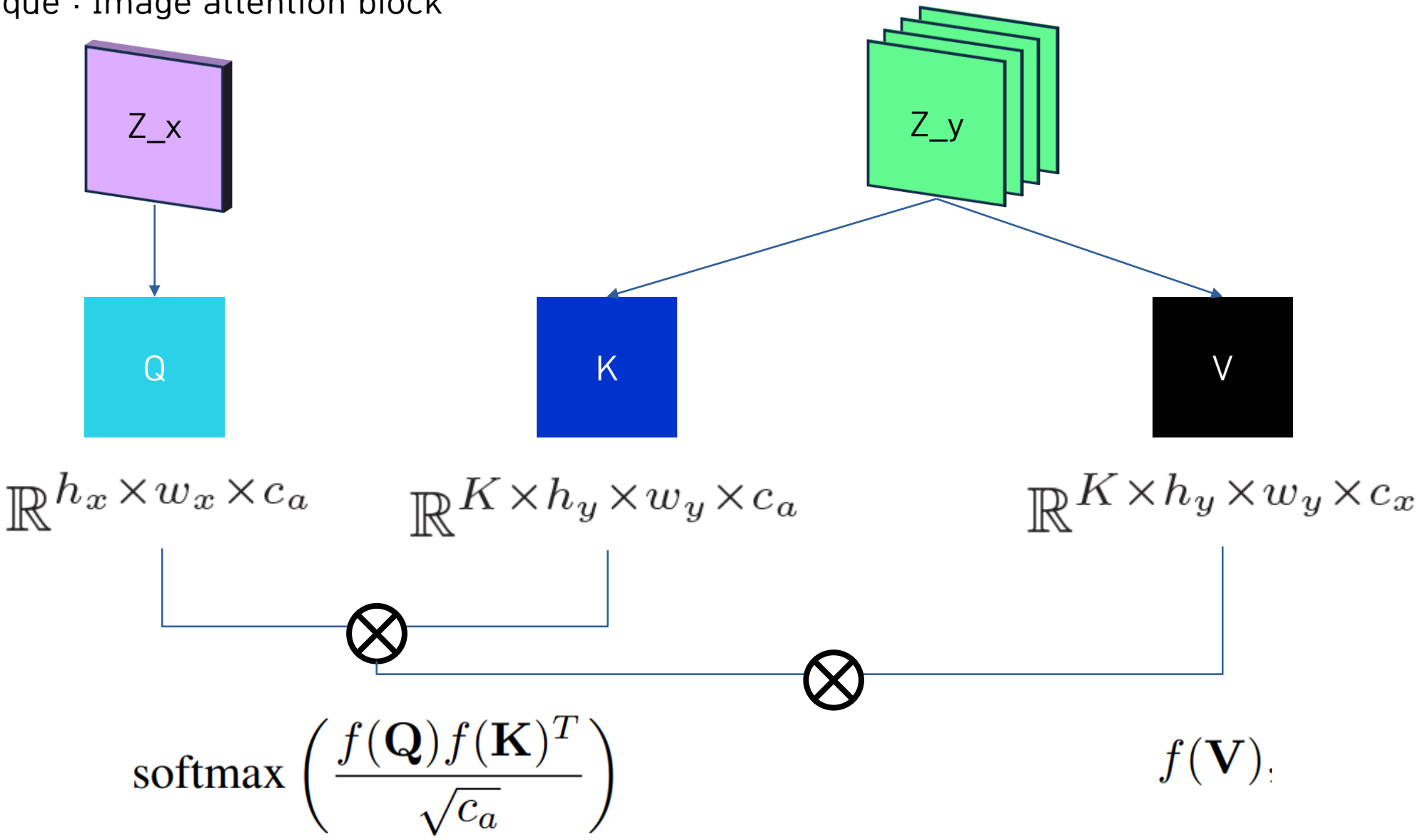
$$
A(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{f(\mathbf{Q}) f(\mathbf{K})^T}{\sqrt{c_a}} \right) f(\mathbf{V}), \tag{2}
$$

- P_x,P_y 는 좌표를 인코딩하는 Sinusoidal 포지셔널 인코딩.
- 위 말한대로 Attention의 position

Main Technique : Image attention block



Main Technique : Image attention block



각 h_x W_x 별 h_y w_y에 대한 Attention 값 = Attention

Main Technique : Target Feature Alignment

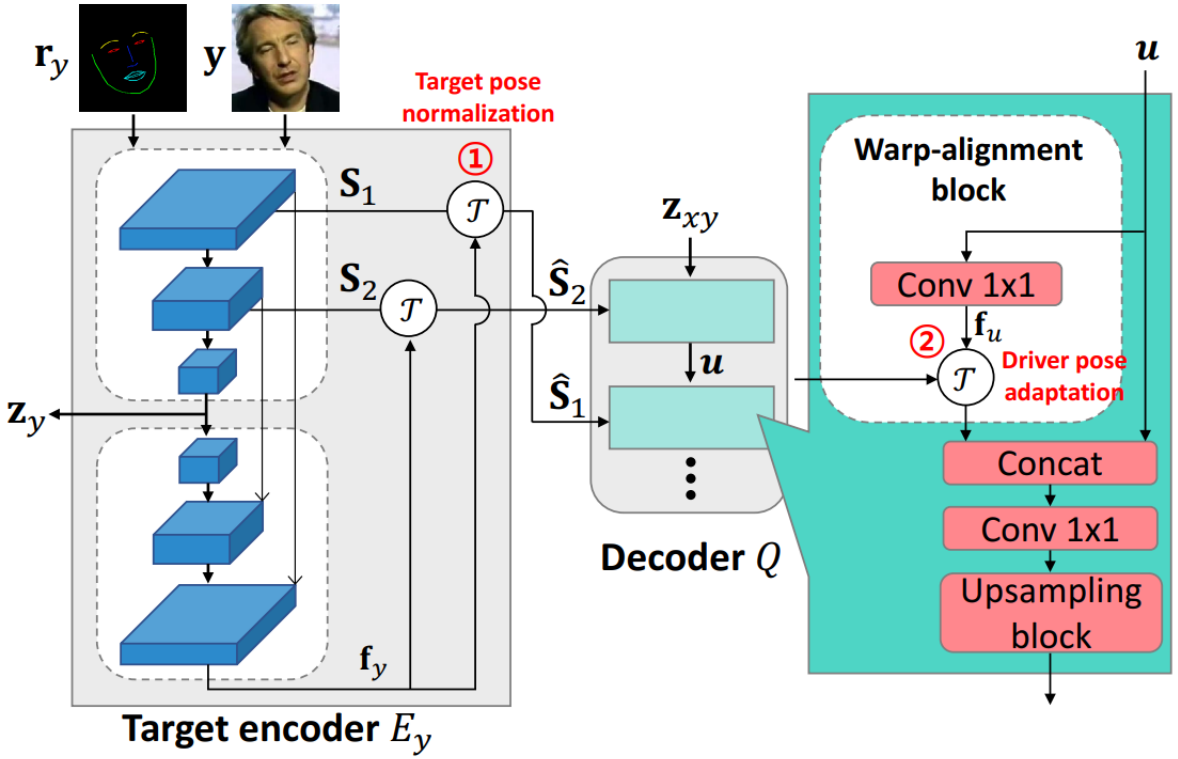
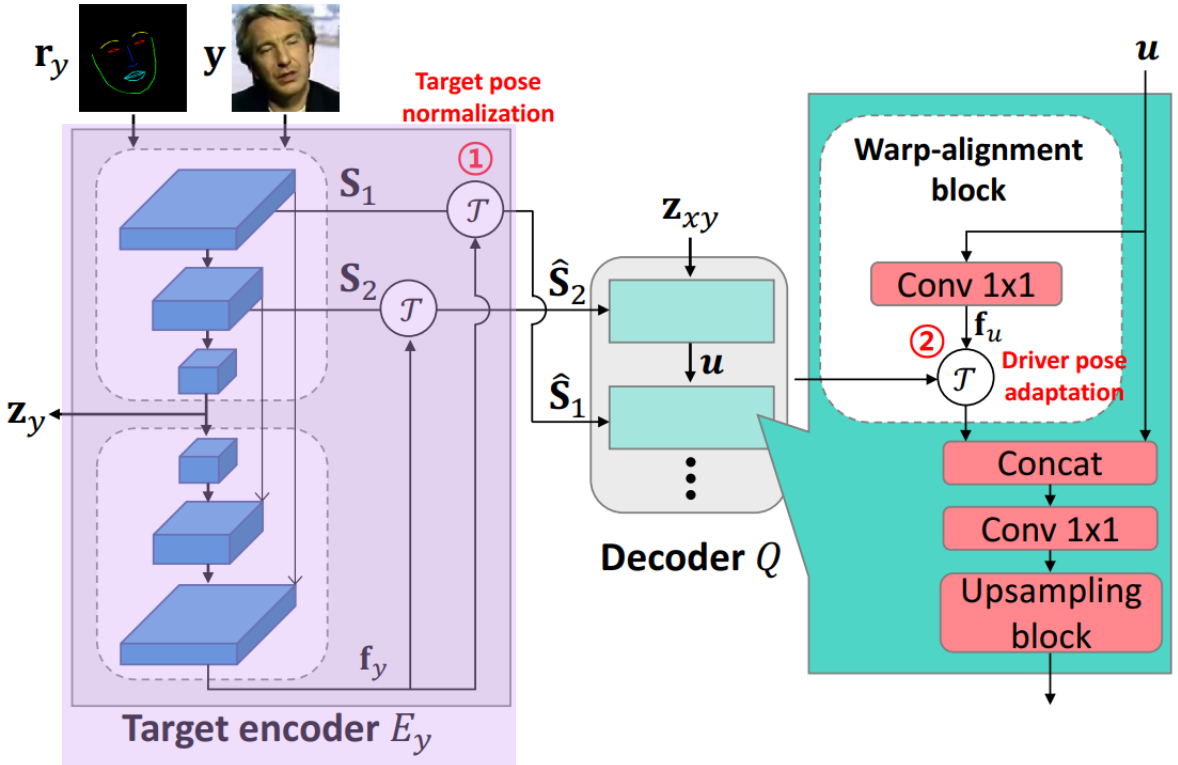


Figure 4: Architecture of target feature alignment.

- 이 논문은 target feature map을 2단계에 걸쳐 워핑하는 target feature alignment를 제시한다.

Main Technique : Target Feature Alignment



$$\hat{\mathbf{S}} = \{\mathcal{T}(\mathbf{S}_1; \mathbf{f}_y), \ldots, \mathcal{T}(\mathbf{S}_{n_y}; \mathbf{f}_y)\}$$

Figure 4: Architecture of target feature alignment.

- 입력값을 U-net구조를 이용해 Skip connection 후 output f_y에 대해 Tau(S_ny,f_y)를 하여 normalizaition

- 이렇게 되면 S'은 pose-agnostic하게 됨

Main Technique : Target Feature Alignment

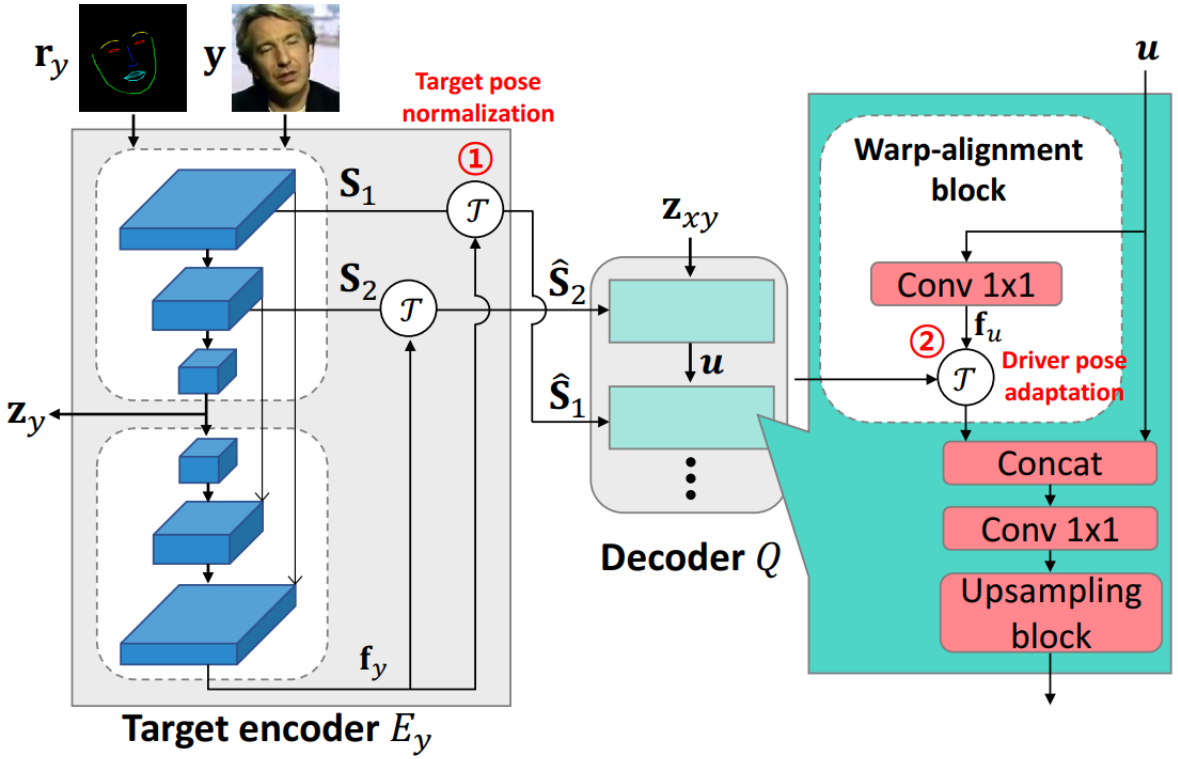


Figure 4: Architecture of target feature alignment.

$$\{\hat{\mathbf{S}}^i\}_{i=1\ldots K}$$

$$\hat{\mathbf{S}}_j = \sum_i \hat{\mathbf{S}}_j^i / K$$

- Adaption 하기 위해 기존 input u(point-wise(z_xy))와 S' 을 input으로 받음

- 그리고 few-shot setting에서 해상도가 호환되는 서로 다른 target 이미지들로부터 얻은 feature map 을 average 시킴 (이해안감)

Main Technique : Target Feature Alignment

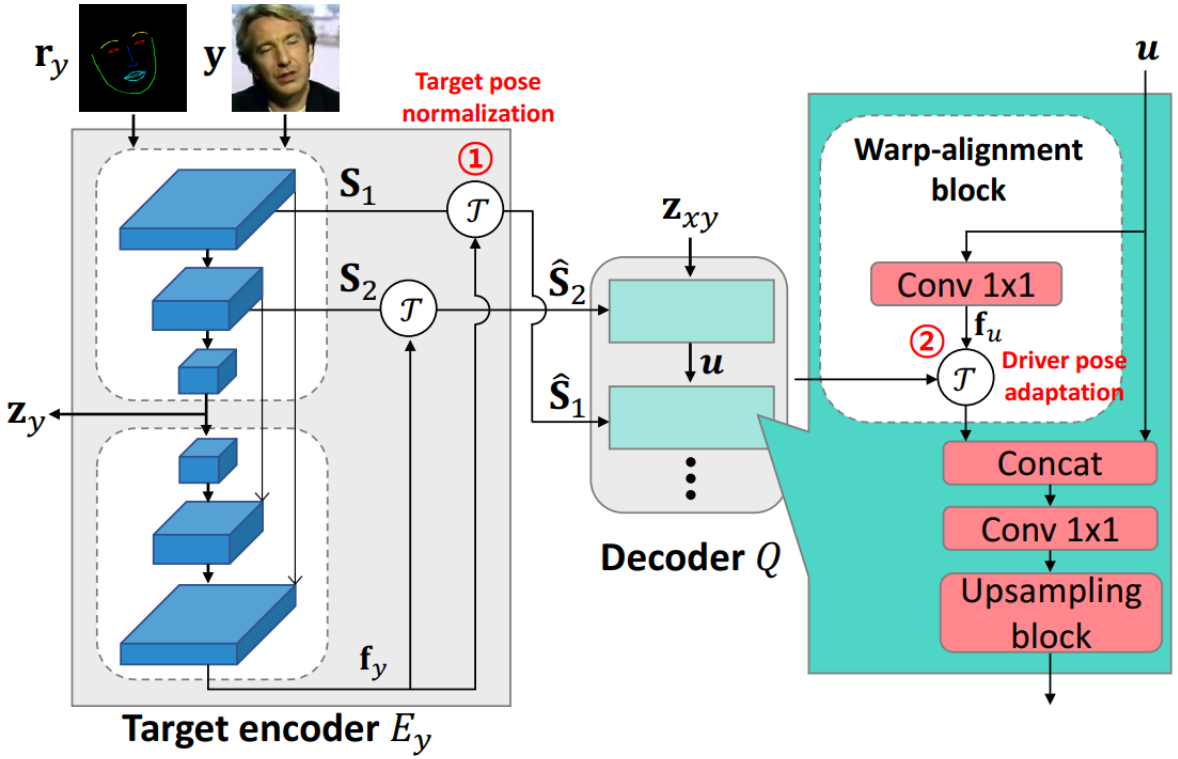


Figure 4: Architecture of target feature alignment.

$$\mathcal{T}(\hat{\mathbf{S}}_j; \mathbf{f}_u)$$

- Alignment -> Up-sampling 수순

결국 Alignment를 시키기위해 Average와 Adaption을 시키는 과정

이에 대한 세세한 테크닉들은 기존 방법

Main Technique : Target Feature Alignment

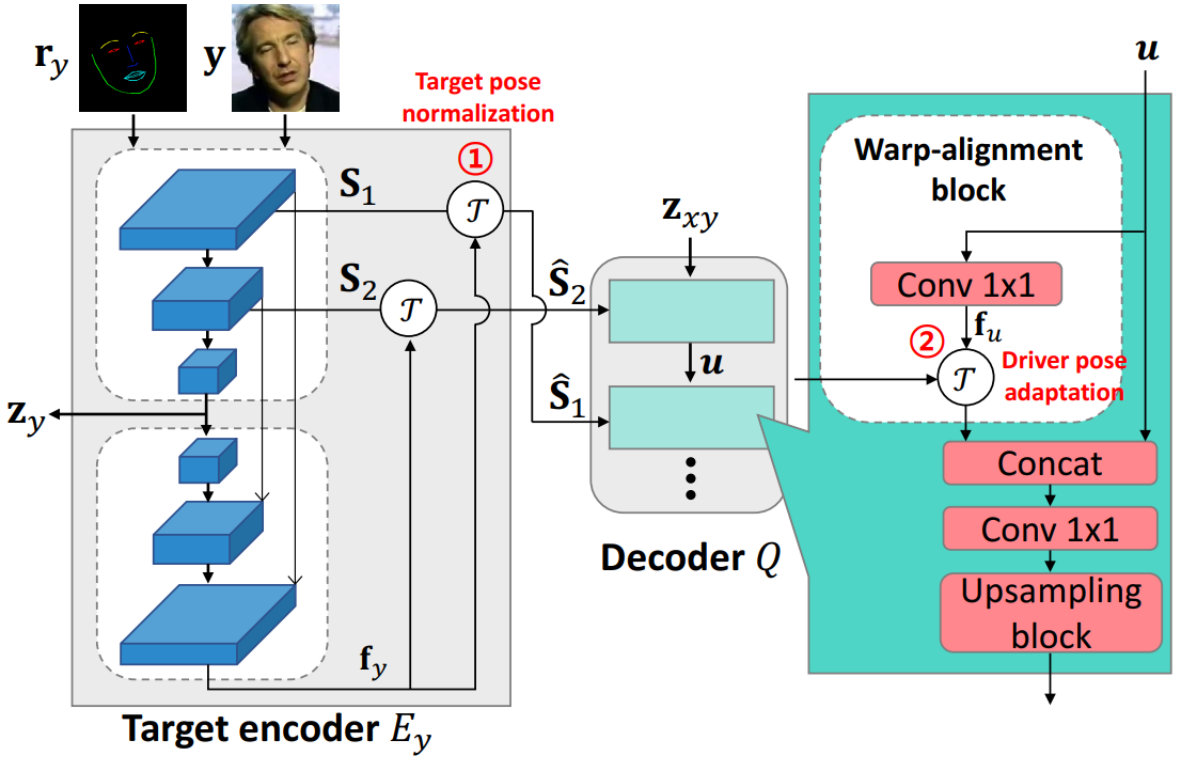


Figure 4: Architecture of target feature alignment.

$$\mathcal{T}(\hat{\mathbf{S}}_j; \mathbf{f}_u)$$

- Alignment -> Up-sampling 수순

결국 Alignment를 시키기위해 Average와 Adaption을 시키는 과정

이에 대한 세세한 테크닉들은 기존 방법

Main Technique : Landmark Transformer

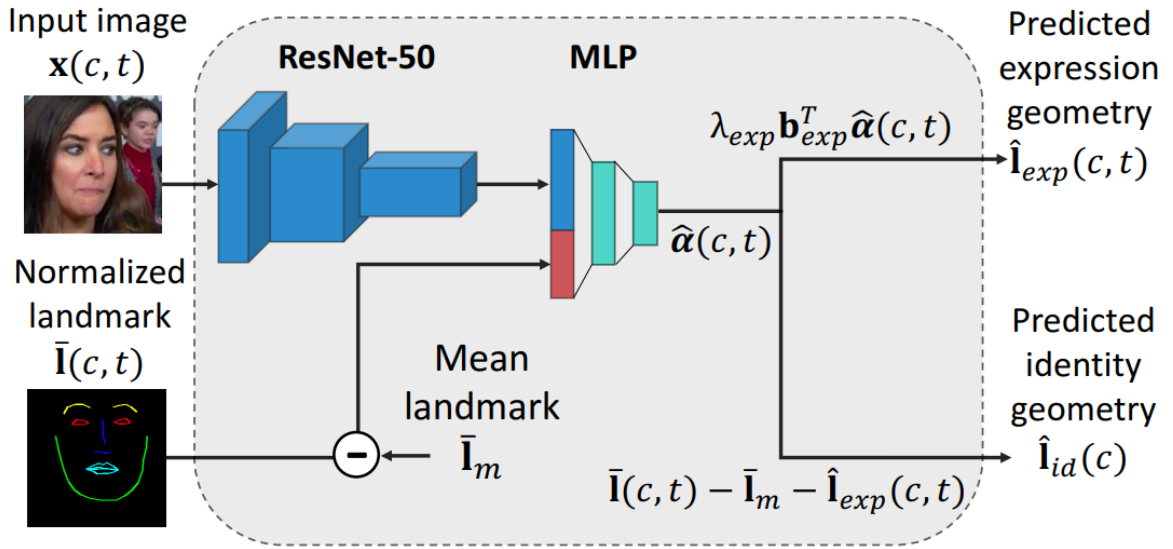


Figure 5: Architecture of landmark disentangler. Note that $\bar{\mathbf{I}}(c,t)$ is a set of landmark points but visualized as an image in the figure.

- 기존 landmark를 뽑아서 진행했던(Labeled) 연구는 input image의 얼굴정보와 일치하지않아 뭉게지는 현상이 발생함
- 그에따라 landmark를 transform하는 연구가 진행되어옴

Main Technique : Landmark Transformer

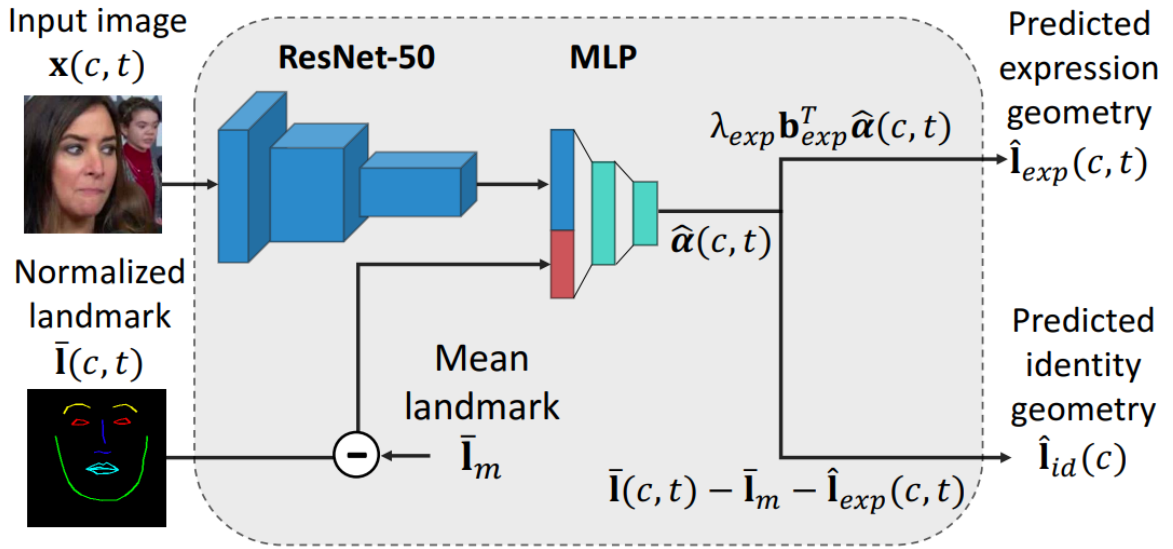


Figure 5: Architecture of landmark disentangler. Note that $\bar{\mathbf{I}}(c,t)$ is a set of landmark points but visualized as an image in the figure.

- Given video의 다른 identity와 input이미지의 identity를 맞추는 작업

Main Technique : Landmark Transformer

Landmark decomposition

$$\bar{\mathbf{l}}(c,t) = \bar{\mathbf{l}}_m + \bar{\mathbf{l}}_{id}(c) + \bar{\mathbf{l}}_{exp}(c,t)$$

- 여러 비디오 장면들이 주어졌을 때,

  c번째 비디오의 t번째 frame 을 위와같이 쓸 수 있다.

- 이를 normalized landmarks로 위와 같이 풀어쓸 수 있음. (Blanz and Vetter 1999 et al - 안읽어봄)

Main Technique : Landmark Transformer

Landmark decomposition

$$\bar{\mathbf{l}}(c,t) = \bar{\mathbf{l}}_m + \bar{\mathbf{l}}_{id}(c) + \bar{\mathbf{l}}_{exp}(c,t)$$

- 이 때,

l_m 은 평균적인 얼굴의 landmark geometry (모든? landmark들의 평균)
l_id(c)는 identity c(Video)의 landmark geometry

$$\bar{\mathbf{l}}_{exp}(c,t) = \bar{\mathbf{l}}(c,t) - \bar{\mathbf{l}}_m - \bar{\mathbf{l}}_{id}(c)$$

- l_expression(c,t)는 위와같이 풀어쓸 수 있다.

Main Technique : Landmark Transformer

Landmark decomposition

$$\bar{\mathbf{l}}(c_x \rightarrow c_y, t_x) = \bar{\mathbf{l}}_m + \bar{\mathbf{l}}_{id}(c_y) + \bar{\mathbf{l}}_{exp}(c_x, t_x)$$

- 본 논문에서 적용되는 수식은 위와 같다.

- Identity는 Target의 것이고, expression은 driver의 것

Few-shot이기에 이 식 외에도 더 필요하다

Main Technique : Landmark Transformer

Landmark disentanglement

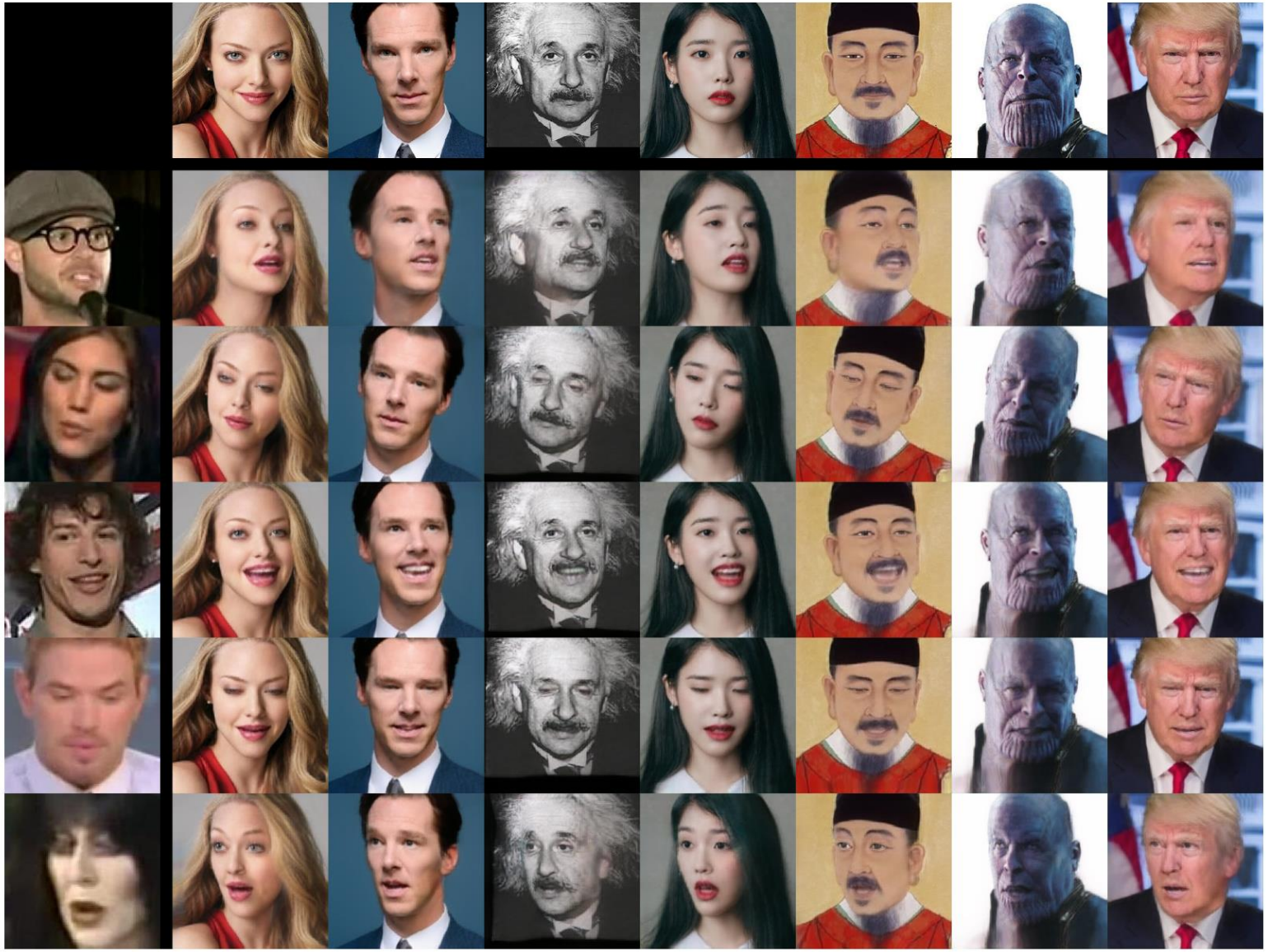
$$\bar{\mathbf{l}}_{exp}(c,t) = \sum_{k=1}^{n_{exp}} \alpha_k(c,t)\mathbf{b}_{exp,k} = \mathbf{b}_{exp}^T \boldsymbol{\alpha}(c,t)$$

- identity와 expression을 분리(disentanglement)하기 위해 neural net을 이용하여 계수를 예측

b_exp,k : 기저(basis) ,
alpha(c,t) : 계수(coefficient)

- 즉, image x(c,t)와 landmark l(c,t)가 주어지면 alpha(c,t)를 예측함. (expression disentanglement)

MarioNETte-LT



너무 길어서 실험부분 뺌.

뒷부분은 본 ppt 만든이 블로그 확인
https://realdr4g0n.github.io/

실험결과는 hyperconnect project page 확인
https://hyperconnect.github.io/MarioNETte/