

# FADING & GOAE 기반 실종 아동 이미지 예측 시스템

Team

CV4팀

Member

김현진 윤정수 김건우 김민규



# Contents

01 Introduction

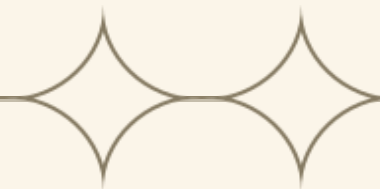
02 Stage 1 - FADING

03 Stage 2 - GOAE

04 Results

05 Limitation

06 References

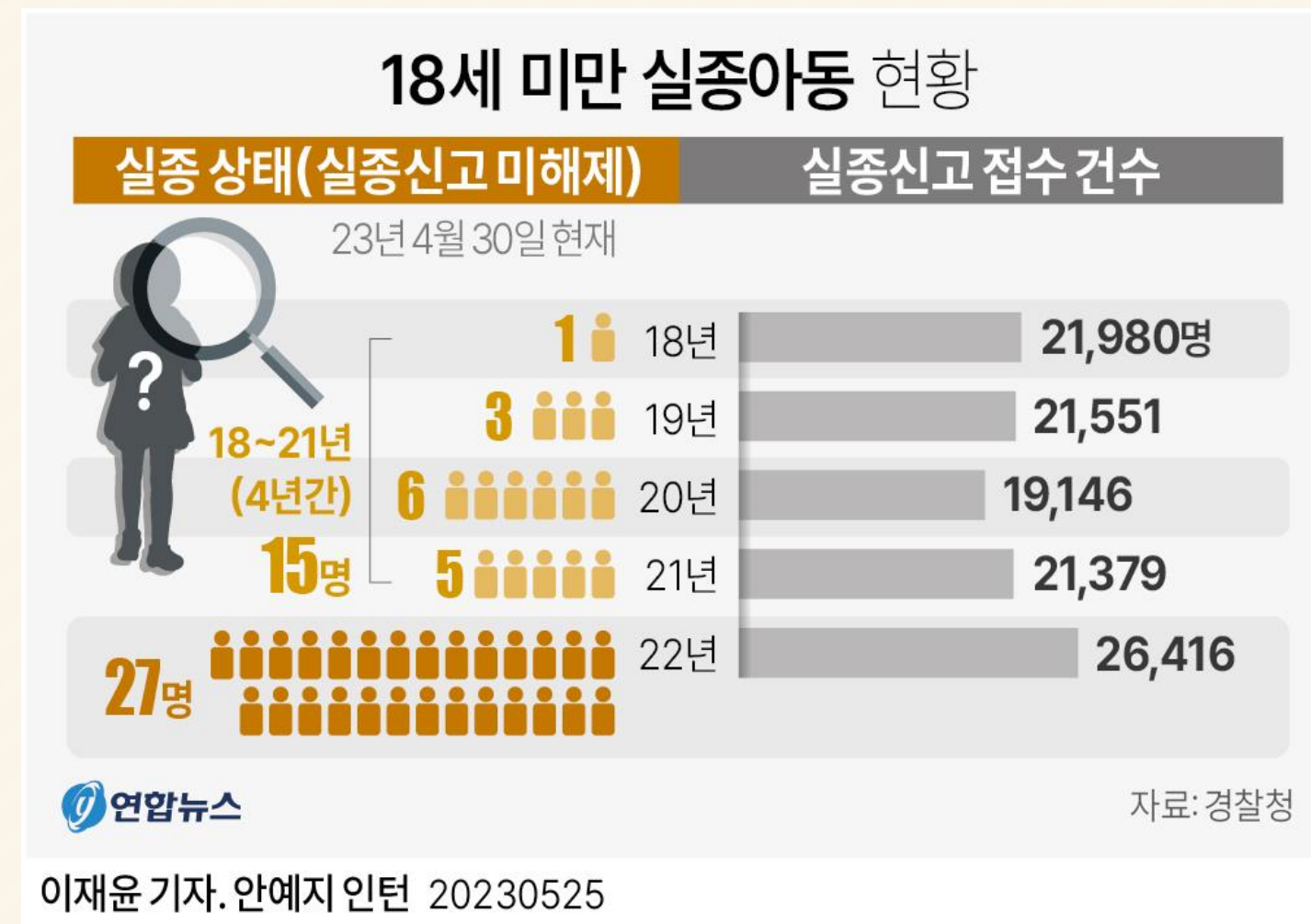




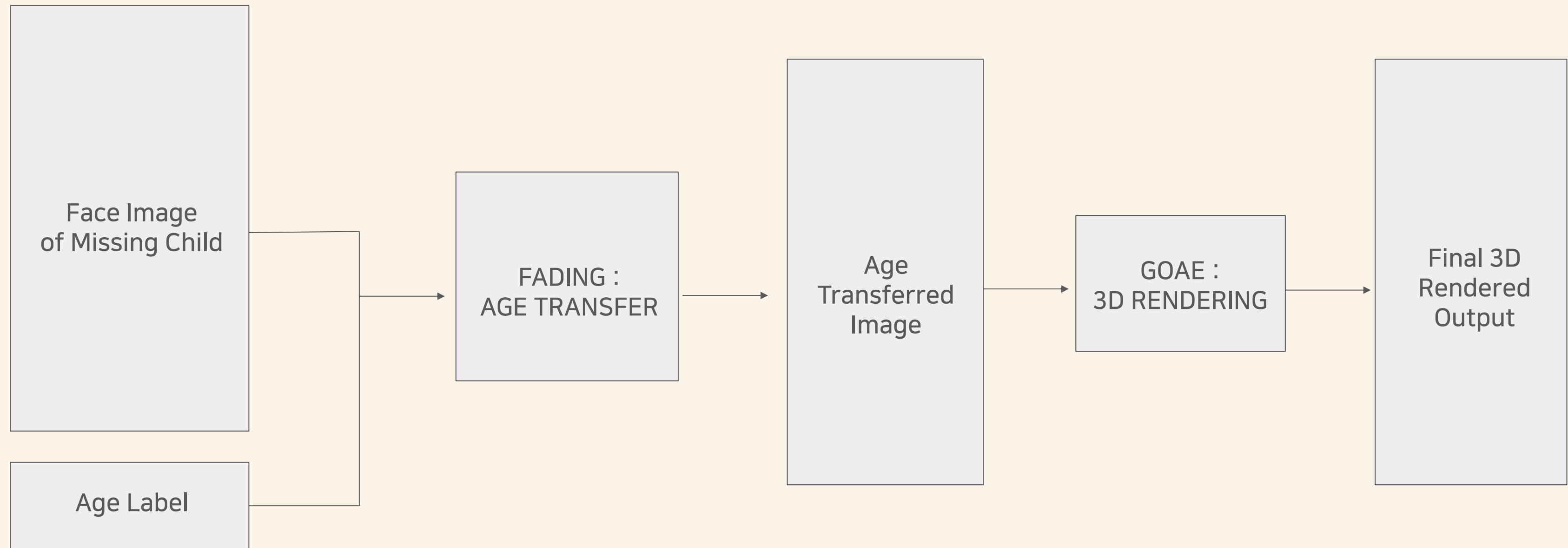
# #1 Introduction



# Introduction



# Model Pipeline



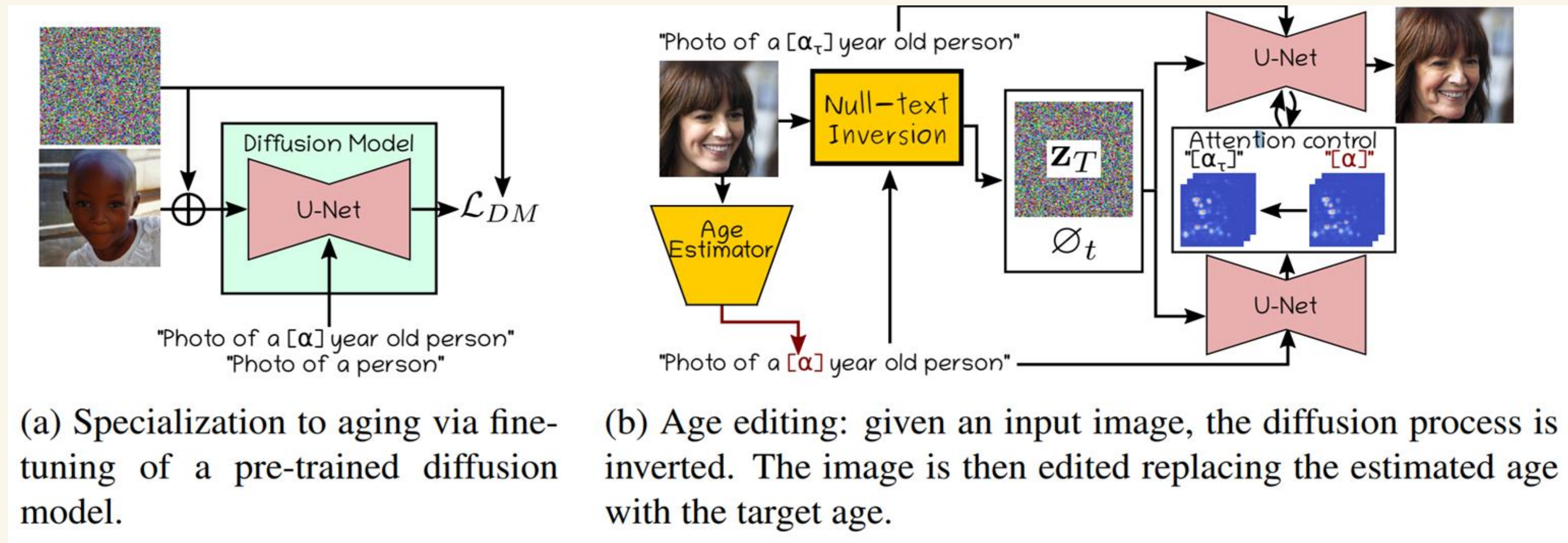


# #2 FADING



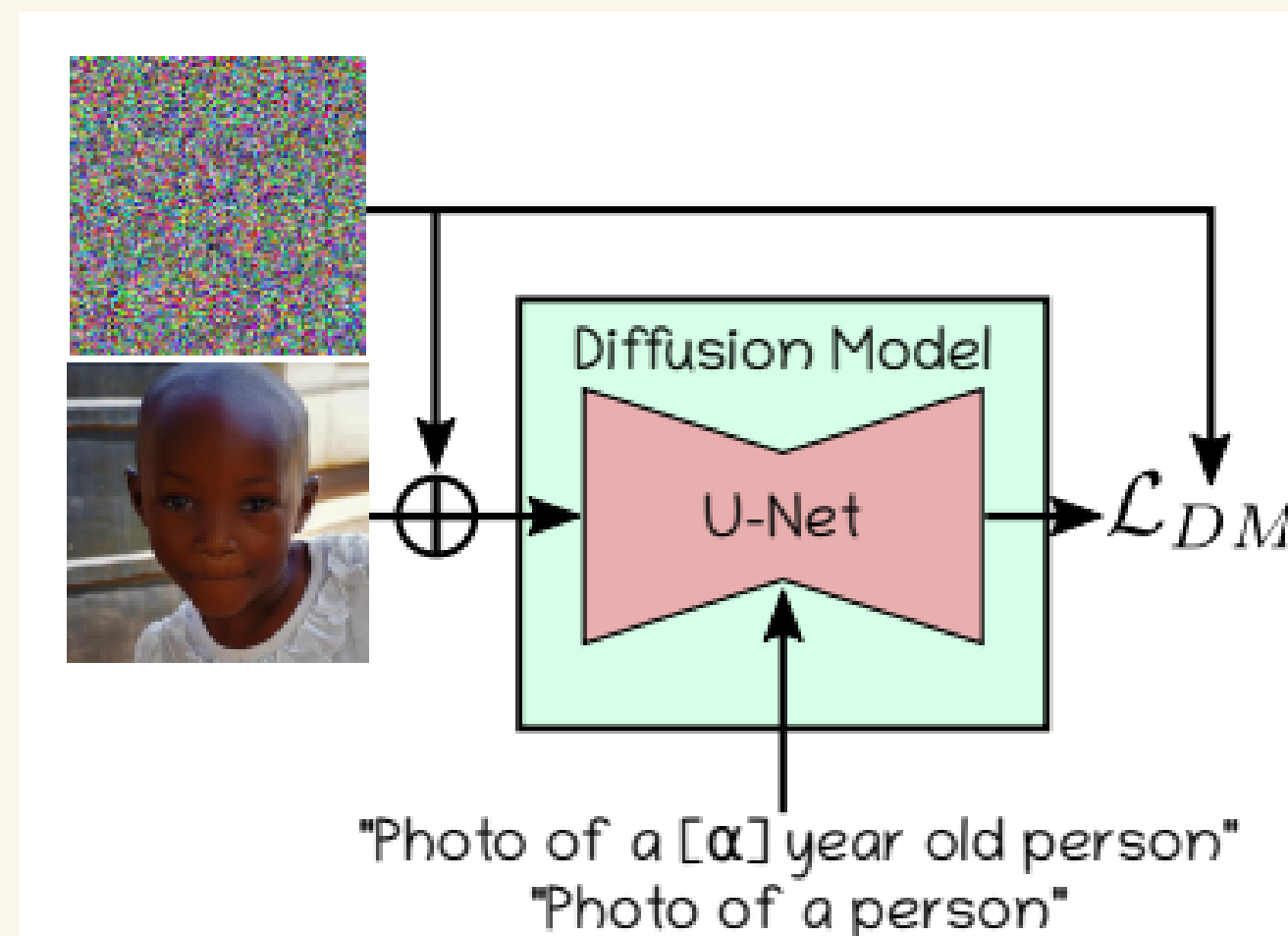
# FADING: Face Aging via Diffusion-based Editing

## Model Architecture



# FADING: Face Aging via Diffusion-based Editing

## Stage 1: Specialization



$$\mathcal{L}_{DM} = \mathbb{E}_{\mathbf{z}_0 \sim \mathcal{E}(x), \alpha, \epsilon, \epsilon', t} [\|\epsilon - \epsilon_\theta(\mathbf{z}_t, t, \mathcal{P})\|_2^2 + \|\epsilon' - \epsilon_\theta(\mathbf{z}'_t, t, \mathcal{P}_\alpha)\|_2^2]$$

인간에 대한 "Specialization"이 필요해!

### Pre-trained Diffusion Model :

- 인간 얼굴에 최적화되어있지 않음
- 나이와 관련된 조건부 프롬프트("30대 남성")는 나이에 대한 개념을 잘 캡처할 수 있지만, "32세 남성"과 같은 특정 숫자를 포함한 텍스트는 정확하게 반영되지 않는 경우가 많음.

나이  $\alpha$ 에 대응하는 얼굴 이미지  $x$ 에 대해

→ "photo of a  $[\alpha]$  year old person"이라는 프롬프트와 함께 fine-tuning

$\alpha$ : 이미지 속 인물의 나이

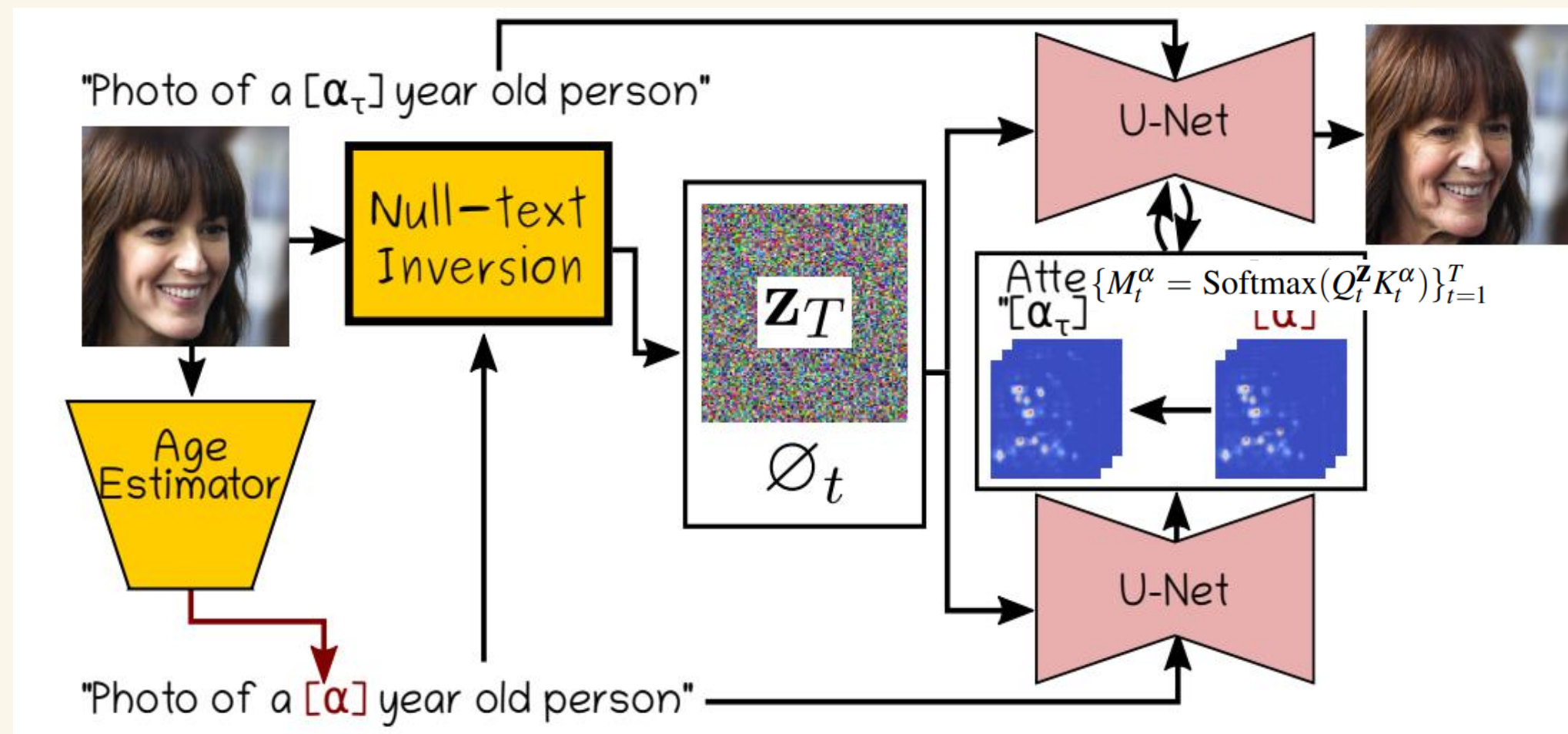
→ "photo of a person"이라는 나이와 무관한 추가 프롬프트를 사용하여 매 반복마다 fine-tuning을 수행하는 double-prompt scheme

⇒ 나이 정보와 나이와 관련 없는 정보(예: 정체성, 배경 등) disentangle 가능



# FADING: Face Aging via Diffusion-based Editing

## Stage 2: Age Editing



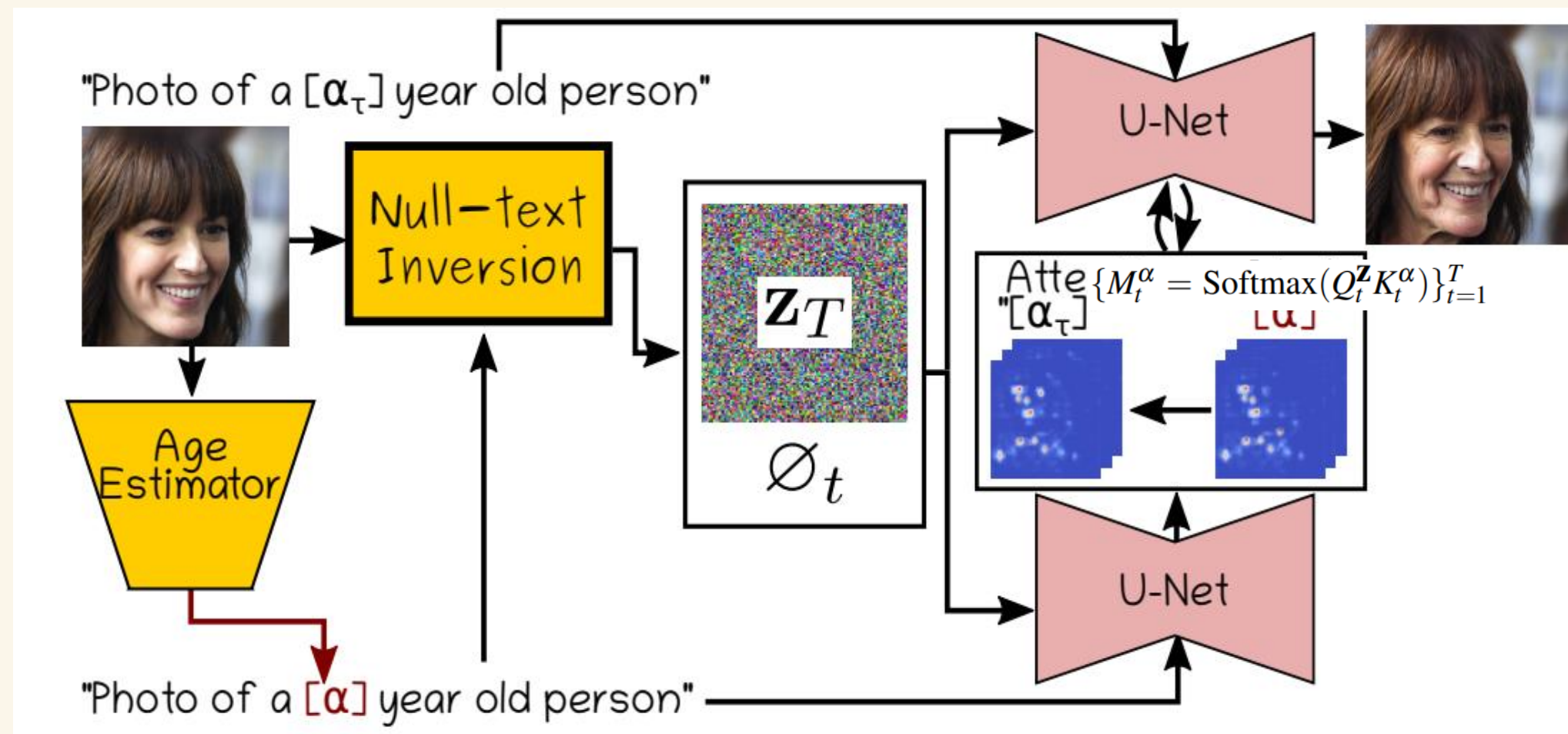
### 01. Null text inversion

- 실제 이미지 편집을 위한 input image와 text prompt inverting을 위한 과정
- Inversion을 통해 얻은 잠재 벡터  $z_T$ 와 optimized unconditional embeddings으로 주어진 prompt에 맞게 이미지 재구성

$$\min_{\mathcal{O}_t} \|z_{t-1}^{inv} - z_{t-1}(\bar{z}_t, t, \mathcal{P}_{inv}; \mathcal{O}_t)\|_2^2$$

# FADING: Face Aging via Diffusion-based Editing

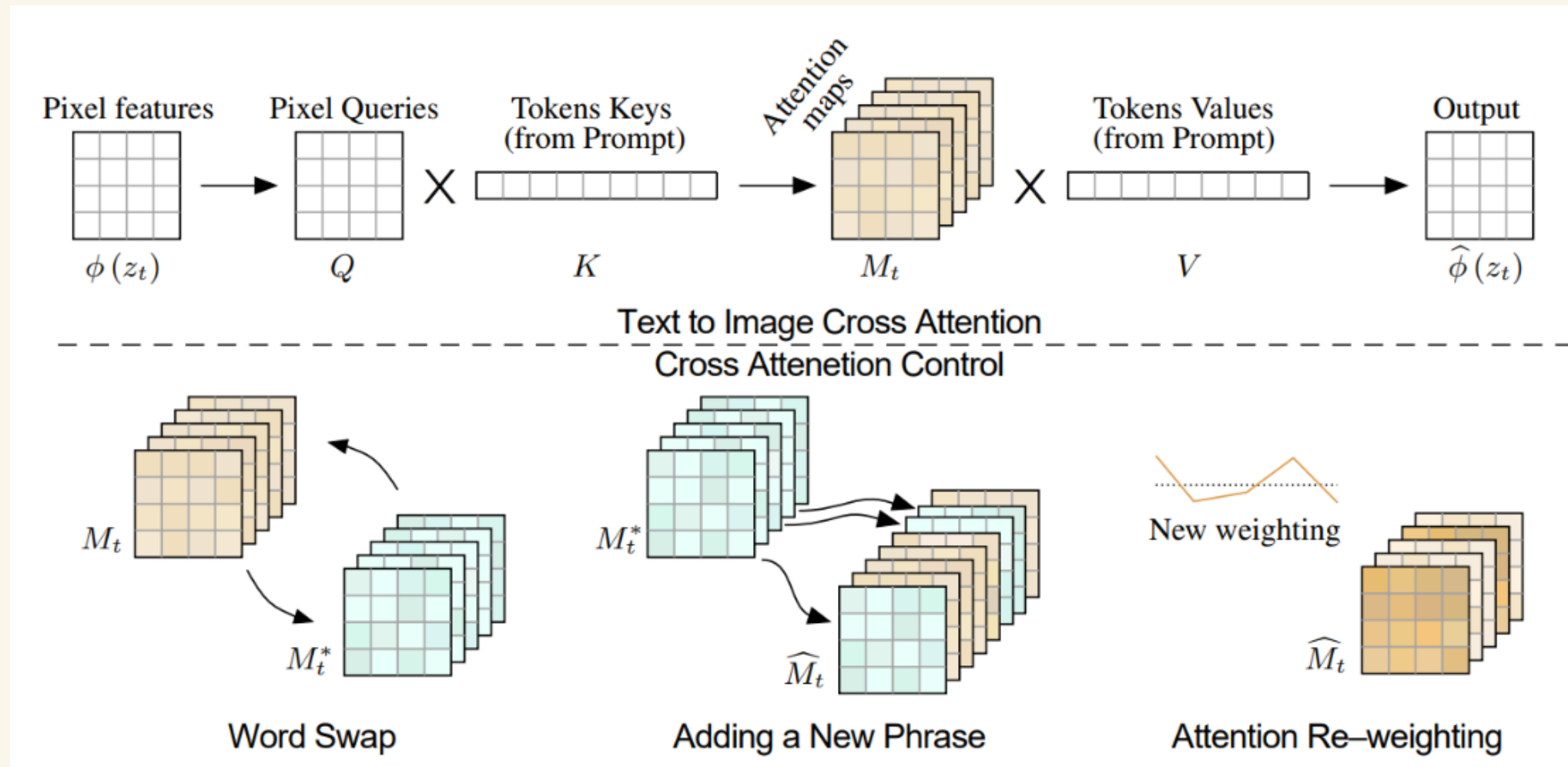
## Stage 2: Age Editing



$$\min_{\varnothing_t} \|z_{t-1}^{inv} - z_{t-1}(\bar{z}_t, t, \mathcal{P}_{inv}; \varnothing_t)\|_2^2$$

# FADING: Face Aging via Diffusion-based Editing

## Stage 2: Age Editing



## 02. 타겟 연령대로의 이미지 변환

→ Diffusion 과정에서  $z_T$ 로부터 계산된 query와 initial prompt로부터 얻은 key를 활용해 cross attention map 계산

$$\{M_t^\alpha = \text{Softmax}(Q_t^Z K_t^\alpha)\}_{t=1}^T$$

→ Attention map은 이미지 속 사람의 연령과 관련한 픽셀을 나타내는 토큰  $\alpha$ 와 대응하게 됨

→ Prompt의 토큰을 변환하고자 하는 타겟 연령대로 바꿈

→ Attention map은 유지한 채로 attention value 계산, original spatial structure는 보존하면서 타겟 연령에 맞춰 이미지 변환

\*\* 새로운 프롬프트에서 나이와 관련한 단어만 갱신되기 때문에 age-related token과 관련이 큰 픽셀만 greatest attention을 받게 됨

# FADING: Face Aging via Diffusion-based Editing

Pre-trained Model



Age Label: 58

Predicted Age :

10

36

45

65

90





# FADING: Face Aging via Diffusion-based Editing

Pre-trained Model



Age Label: 58

Predicted Age :

10

36

45

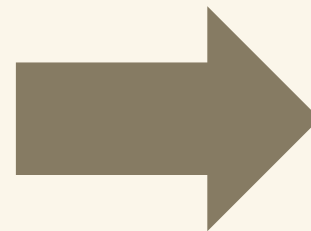
65

90



# FADING: Face Aging via Diffusion-based Editing

Pre-trained Model

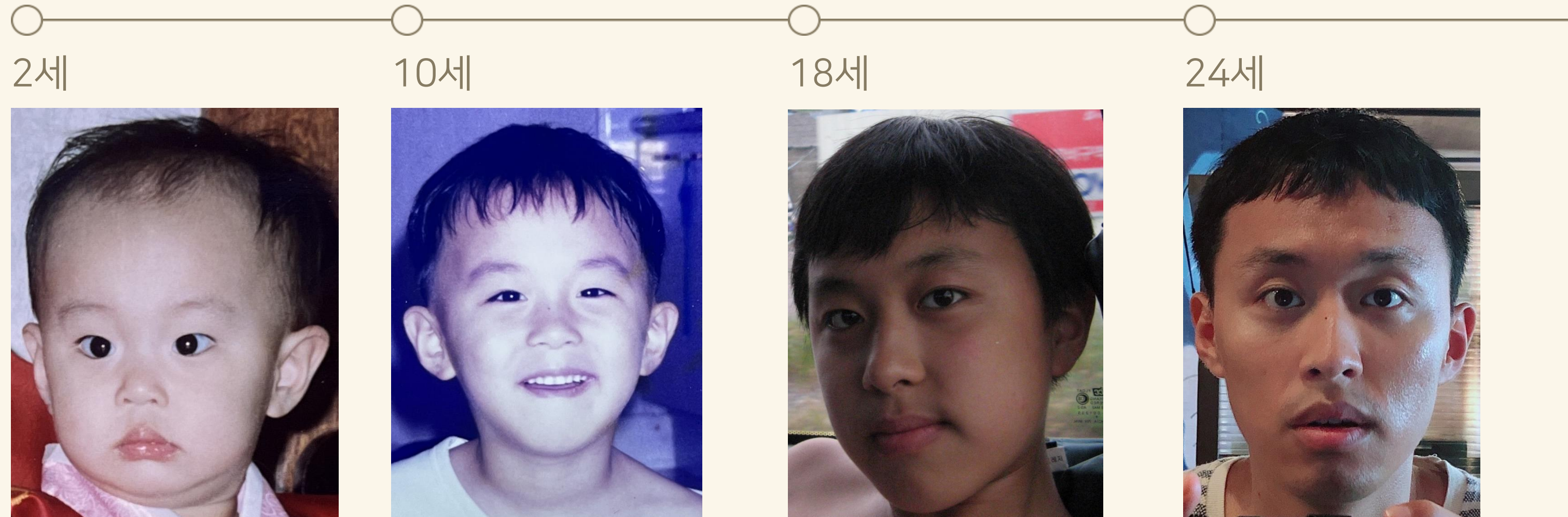


한국인 얼굴 이미지 데이터로 Fine Tuning 수행



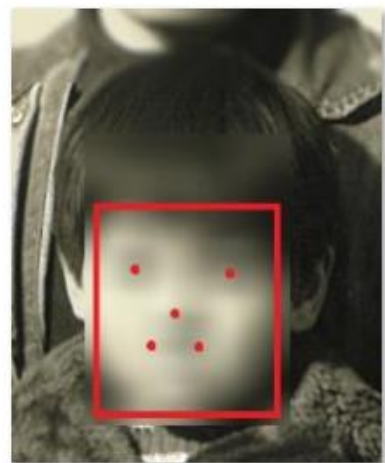
# FADING: Face Aging via Diffusion-based Editing

Fine Tuning Dataset: 안면 인식 에이징 (Aging) 이미지 데이터 (출처: AI Hub)



# FADING: Face Aging via Diffusion-based Editing

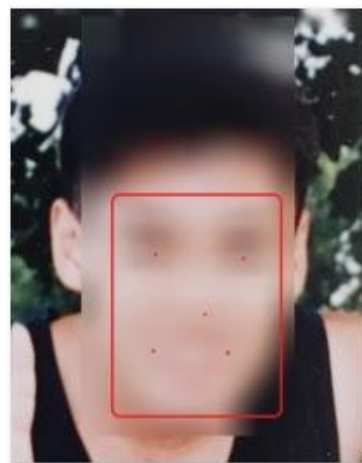
Fine Tuning Dataset: 안면 인식 에이징 (Aging) 이미지 데이터 (출처: AI Hub)



6세 사진



24세 사진



27세 사진



39세 사진



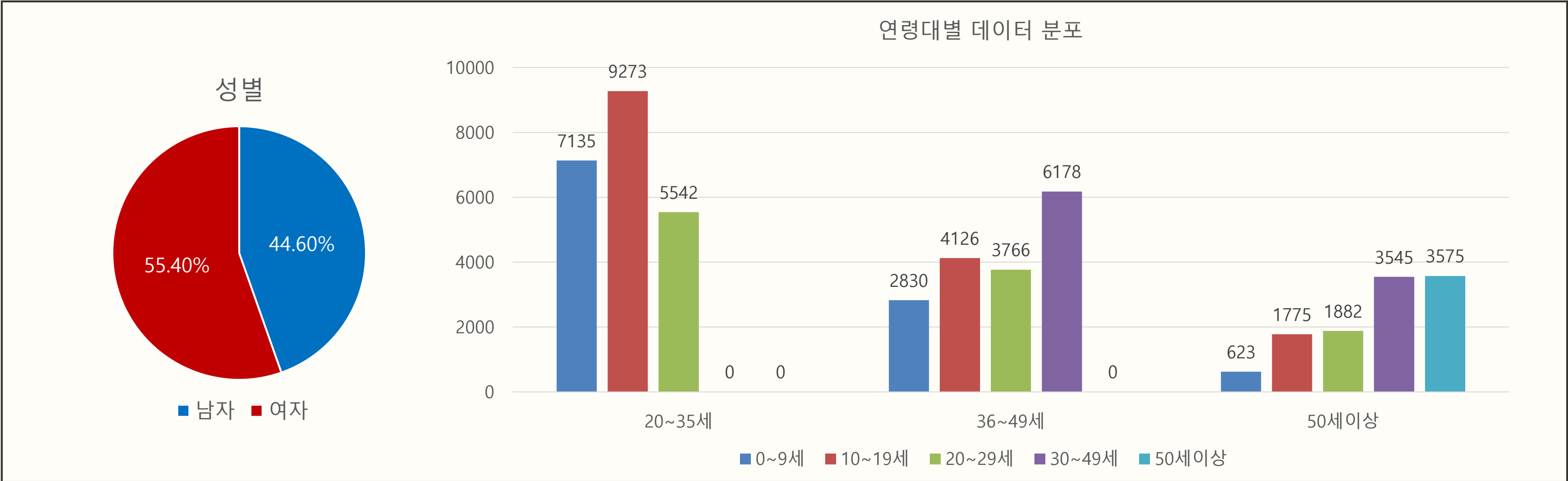
50세 사진

- 이름: 000
- 출생 연도: 1994년
- 현재 나이: 29세
- 과거 나이: 24세
- 성별: 남성
- 박스 좌표:  $x = 78.87, y = 355.88, w = 612.76, h = 670.00$
- 얼굴 랜드마크: coordinate 1 = (381.68, 664.49), 2, 3, 4, 5



# FADING: Face Aging via Diffusion-based Editing

Fine Tuning Dataset: 안면 인식 에이징 (Aging) 이미지 데이터 (출처: AI Hub)



# FADING: Face Aging via Diffusion-based Editing

## Data Preprocessing: Face Alignment & Normalization

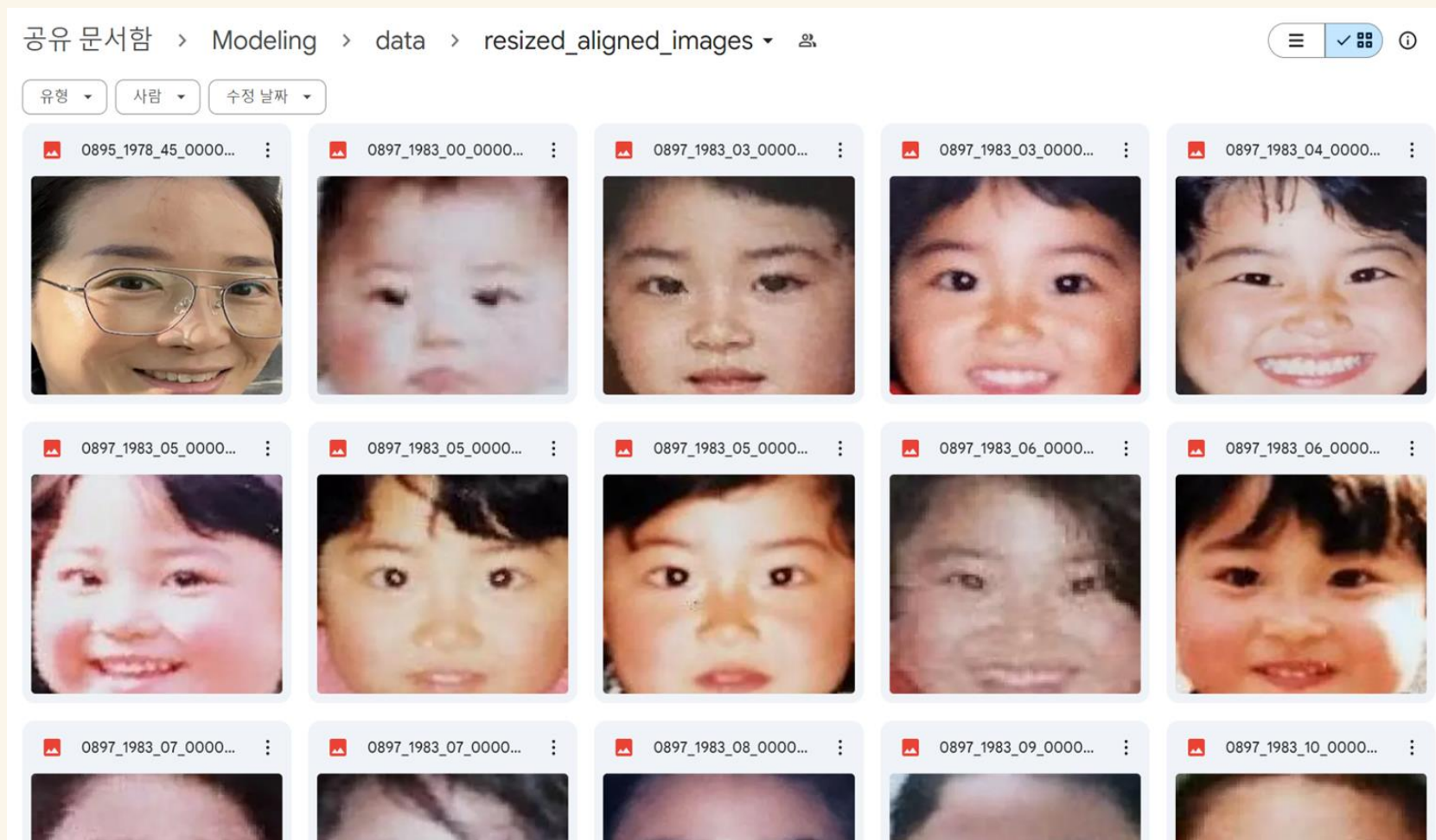
```
import matplotlib.pyplot as plt
faces = RetinaFace.extract_faces(img_path = "img.jpg", align = True)
for face in faces:
    plt.imshow(face)
    plt.show()
```



CNN 기반 Face Detection model  
**RetinaFace** 활용

# FADING: Face Aging via Diffusion-based Editing

## Data Preprocessing: Face Alignment & Normalization



학습에 총 2,043개 Face Image 활용  
→ Fine Tuning 목적으로 4GB의 이미지 사용

# FADING: Face Aging via Diffusion-based Editing

## Data Preprocessing: image Dataset & Age info npy

### Train a new model

```
accelerate launch specialize_general.py \
--instance_data_dir 'specialization_data/training_images' \
--instance_age_path 'specialization_data/training_ages.npy' \
--output_dir <PATH_TO_SAVE_MODEL> \
--max_train_steps 150
```

Training images should be saved at `specialization_data/training_images`. The training set is described through `training_ages.npy` that contains the age of the training images.

```
array([[ '00007.jpg', '1'],
       [ '00004.jpg', '35'],
       ...,
       [ '00009.jpg', '35']], dtype='<U21')

```

```
root_folder = "label" # Change this to your desired path

data_list = []

for subdir, dirs, files in os.walk(root_folder):
    for file in files:
        if file.endswith('.json'):
            file_path = os.path.join(subdir, file)

            with open(file_path, 'r', encoding='utf-8') as json_file:
                json_data = json.load(json_file)

                file_name = json_data.get('filename', '')
                file_format = json_data.get('format', '')
                age_past = str(json_data.get('age_past', ''))

                full_file_name = f"{file_name}.jpg"

                data_list.append([full_file_name, age_past])

result_array = np.array(data_list)

print(result_array)

[ ['0895_1978_04_00000044_F.jpg' '4']
  ['0895_1978_04_00000045_F.jpg' '4']
  ['0895_1978_06_00000046_F.jpg' '6']
  ...
  ['1004_1999_24_00000043_d.jpg' '24']
  ['1004_1999_24_00000044_d.jpg' '24']
  ['1004_1999_24_00000045_d.jpg' '24']]

```



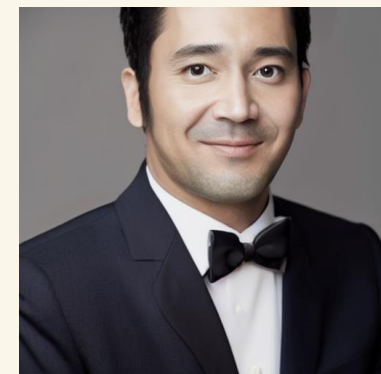
# FADING: Face Aging via Diffusion-based Editing

Predicted Age :  
10      36      45      65      90

*Pre-trained...*

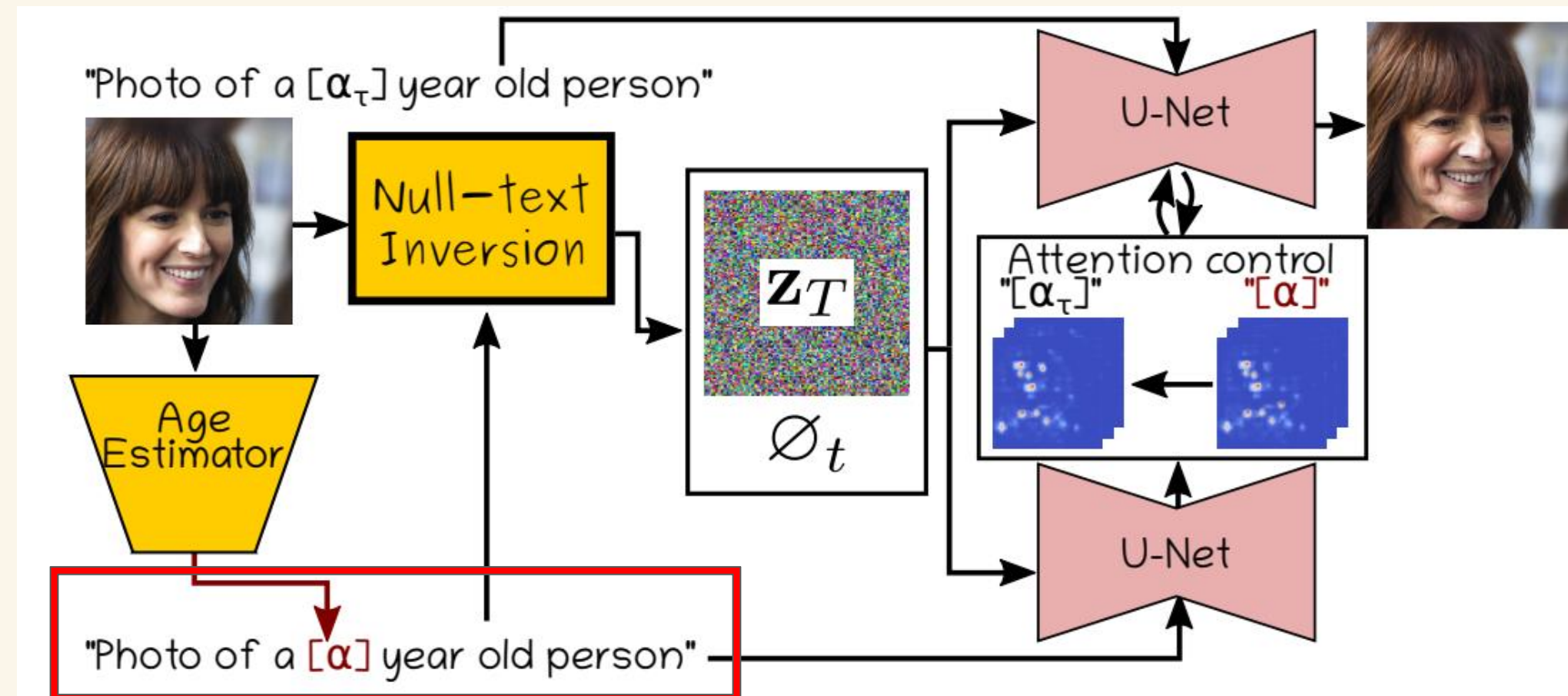


*Fine-tuned...*



# FADING: Face Aging via Diffusion-based Editing

→ *changing prompt!*





# FADING: Face Aging via Diffusion-based Editing

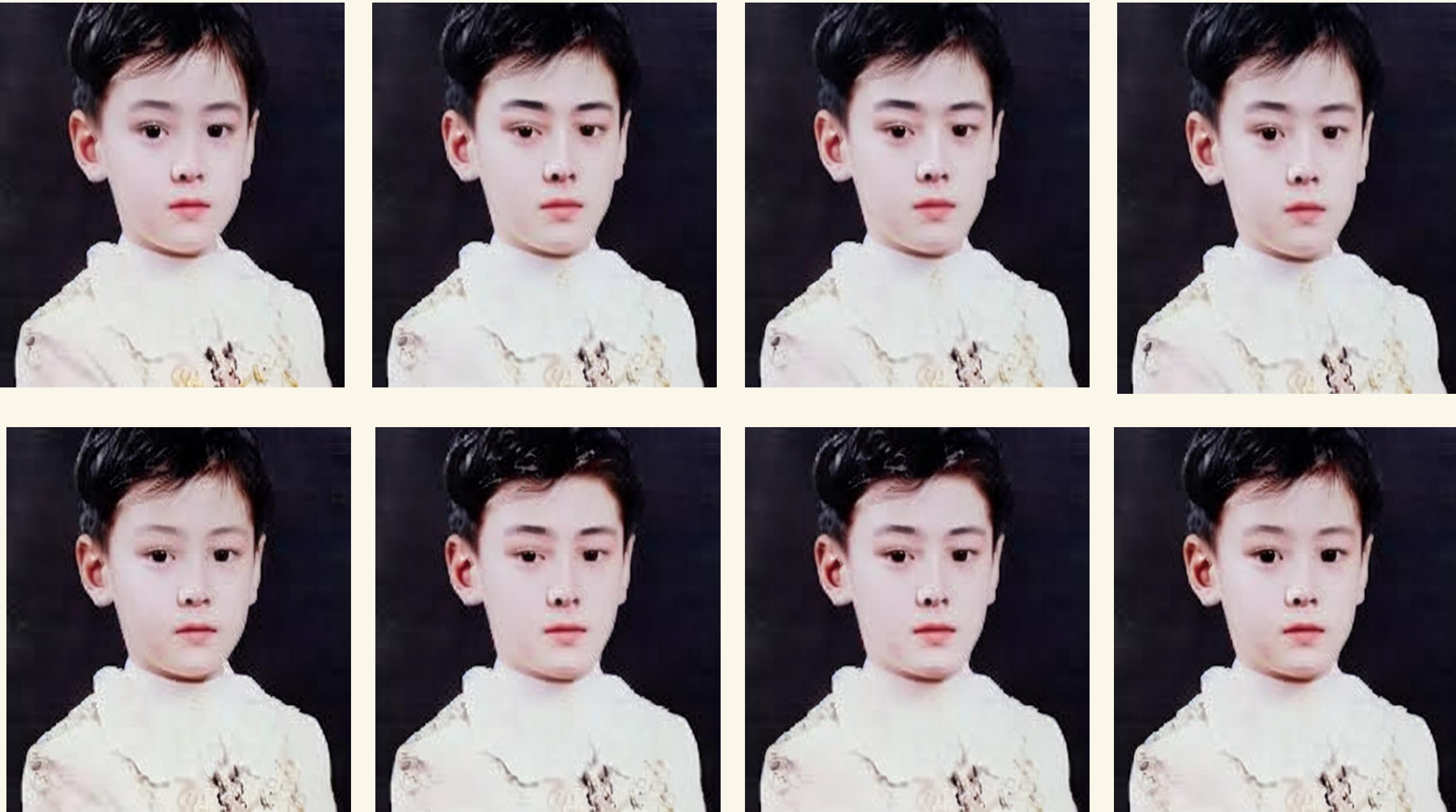
Predicted Age : 12                      16                      22                      36

Original  
&  
Asian



Age Label: 7

Young  
looking





Predicted Age : 12

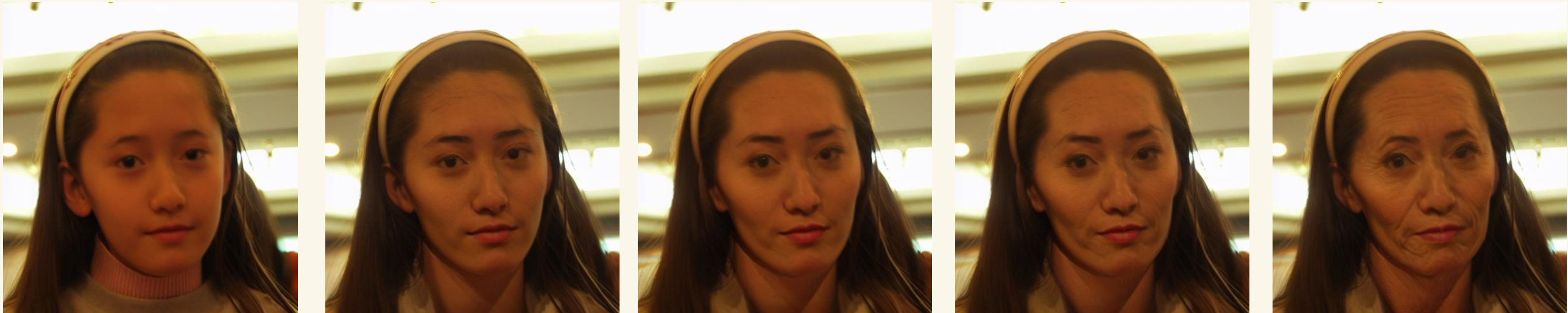
16

22

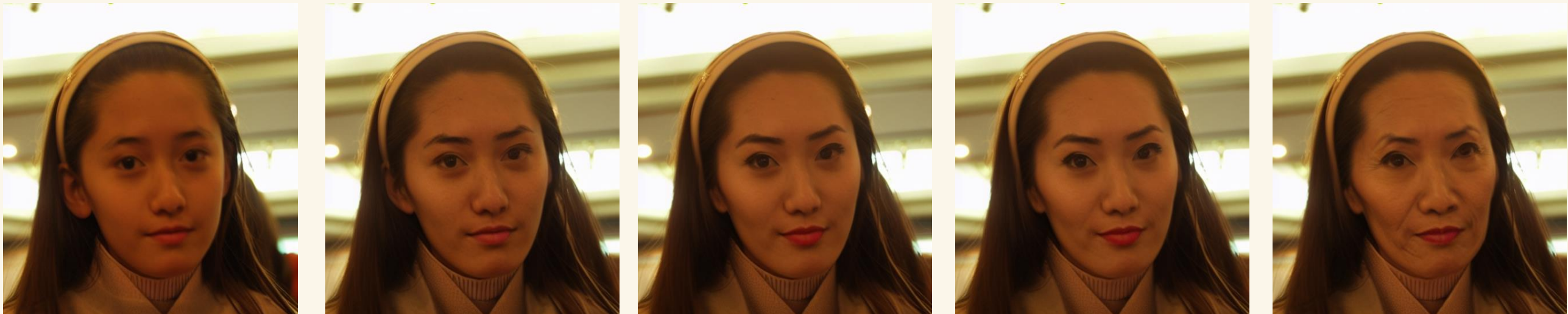
36

40

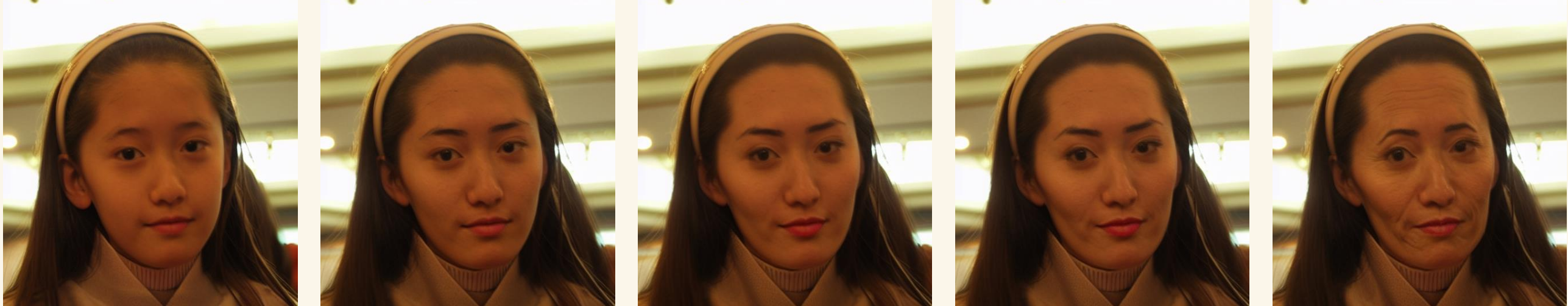
Original



Asian



Young looking



Age Label: 9





Predicted Age :

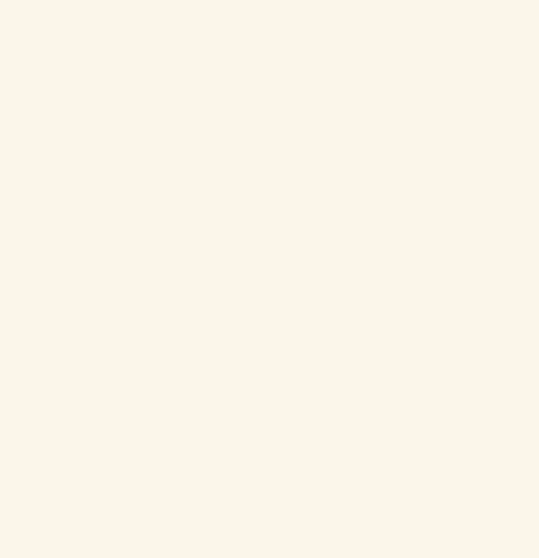
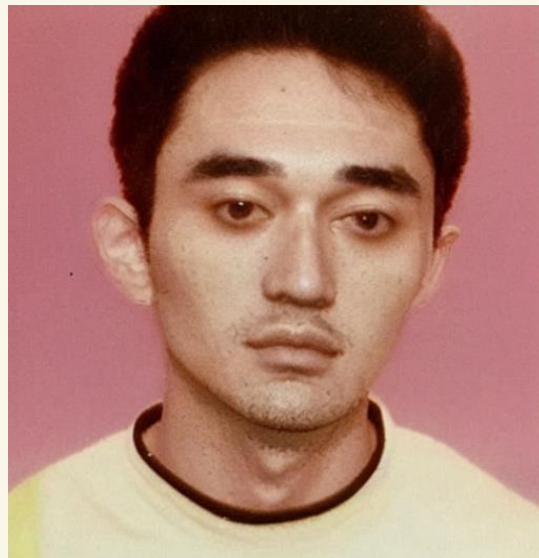
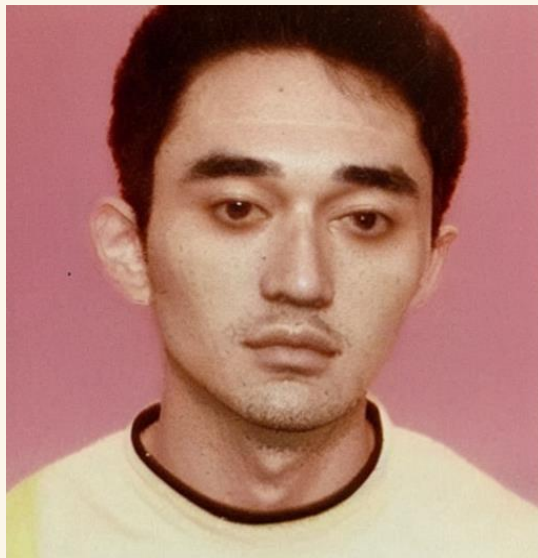
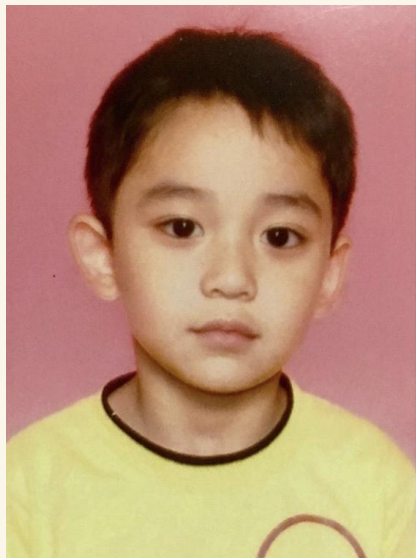
12

16

22

36

40





# #3 GOAE



# GOAE: Geometry and Occlusion-Aware Encoding

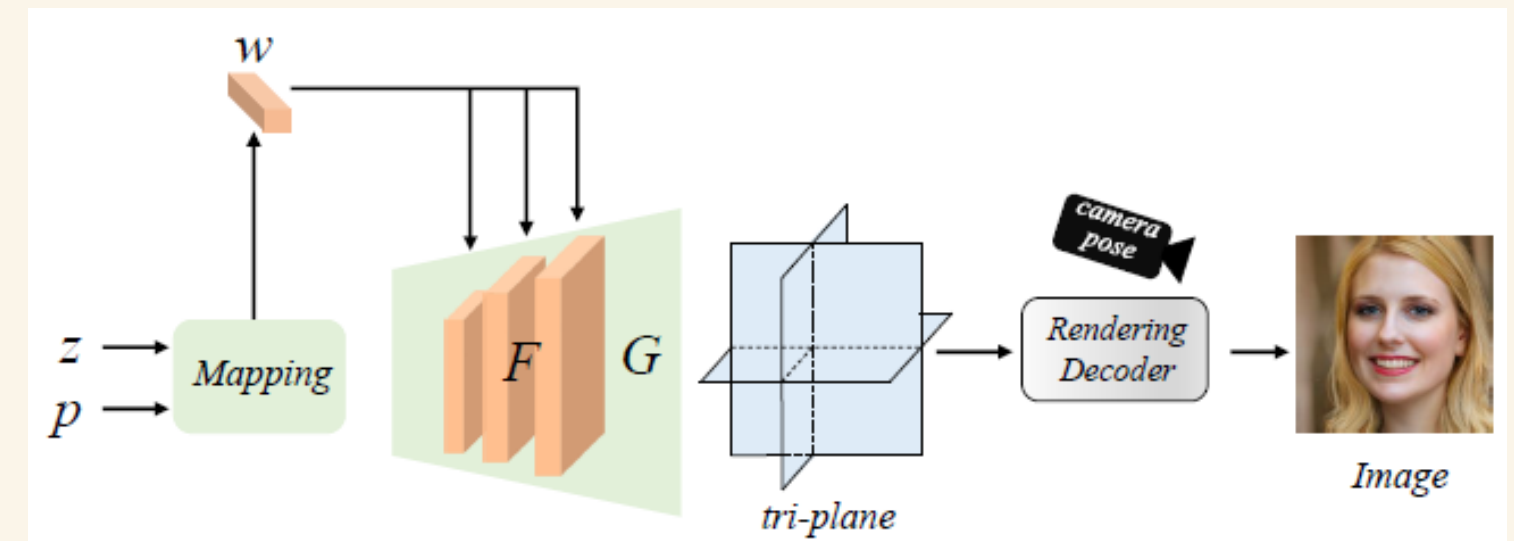
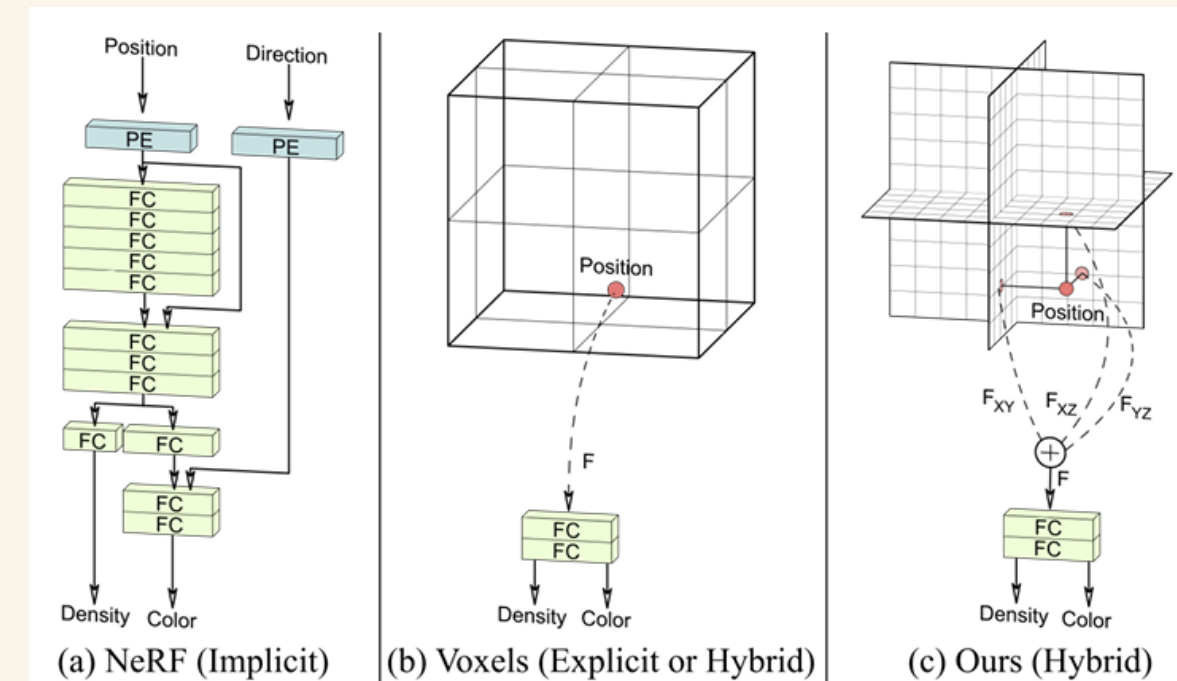
## EG3D

### Tri-plane 3D representation

→ FC layer를 활용한 성능 향상 중심의 Implicit 방법과  
공간 좌표를 활용한 효율성 중심의 Explicit 방법의 결합

### 3D-GAN Framework

→ 저해상도 이미지에서 랜더링 후 고해상도로 upsampling  
사진 촬영 각도 (pose)와 저/고해상도 이미지를 동시에  
활용해 이미지 진위 여부를 구분하는 이중 식별 형태



# GOAE: Geometry and Occlusion-Aware Encoding

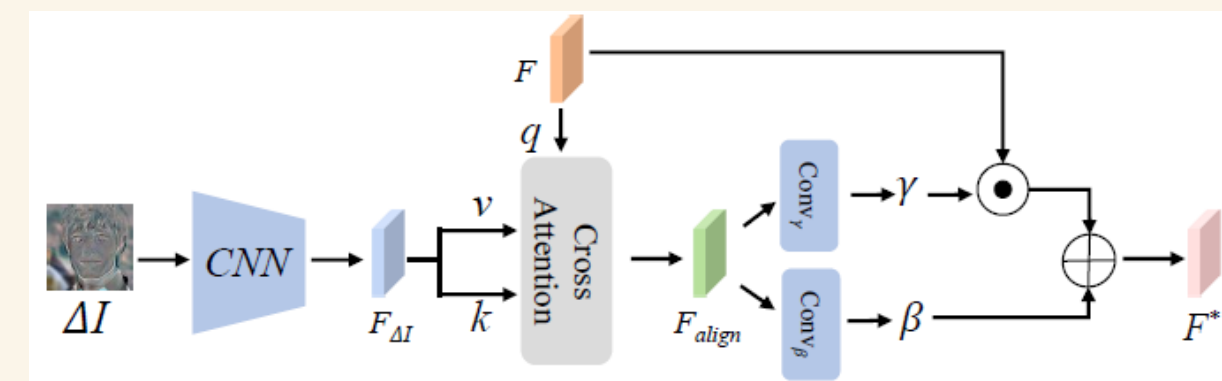
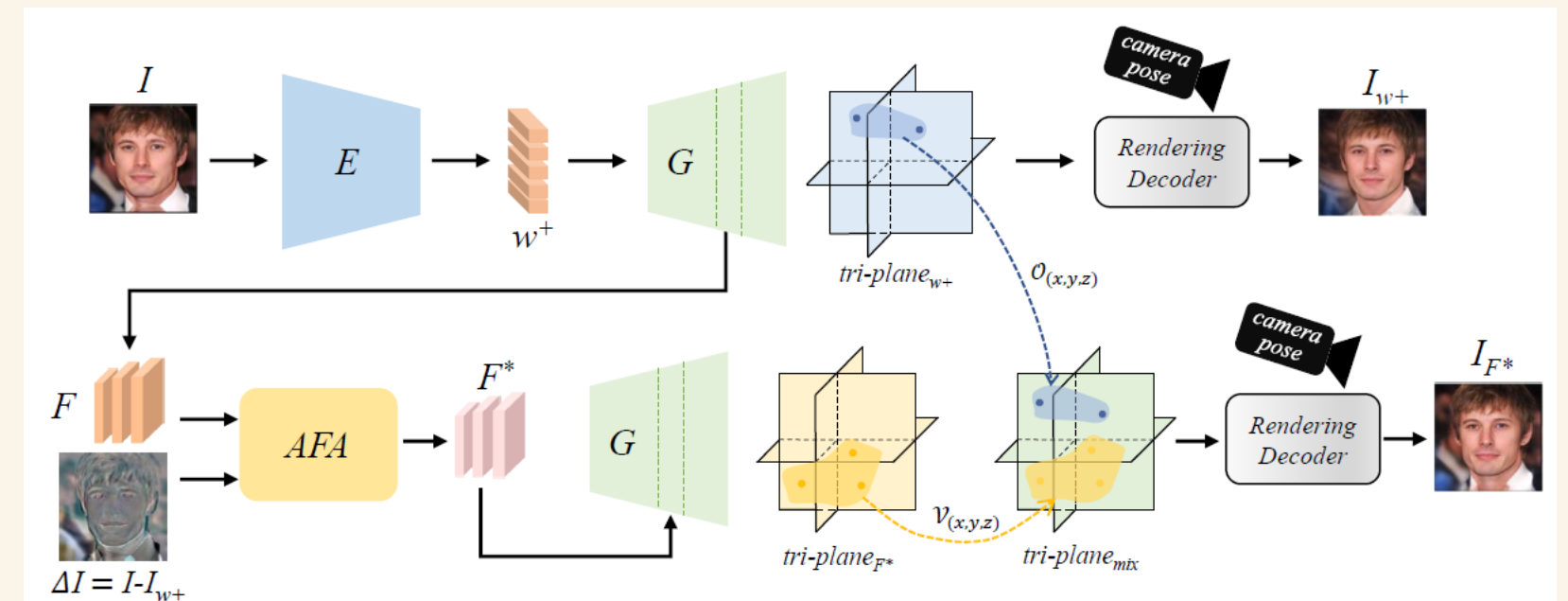
## GOAE

### Geometry / Occlusion Aware Learning

- Swin-Transformer 기반 Encoder로 face/pose 1차 학습 + 실제 image와 복원된 이미지 사이의 차이 기반 정보 보완 = Geometry(얼굴형), Occlusion(교정) 기반 학습

### Adaptive Feature Alignment (AFA)

- 생성 이미지와 실제 이미지 간 Occlusion을 학습하는 과정에서 차이에 해당하는 부분을 새로운 Canonical Latent Space에 투영해 학습시켜 robustness를 높이는 방식





# GOAE: Geometry and Occlusion-Aware Encoding

Data Preprocessing: Camera parameters

일부 이미지에 대해 눈코입이 하나로 겹쳐지지 않는 현상 발생

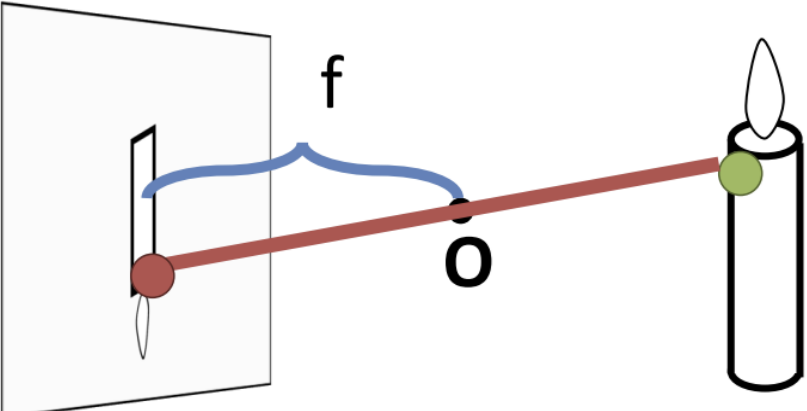


# GOAE: Geometry and Occlusion-Aware Encoding

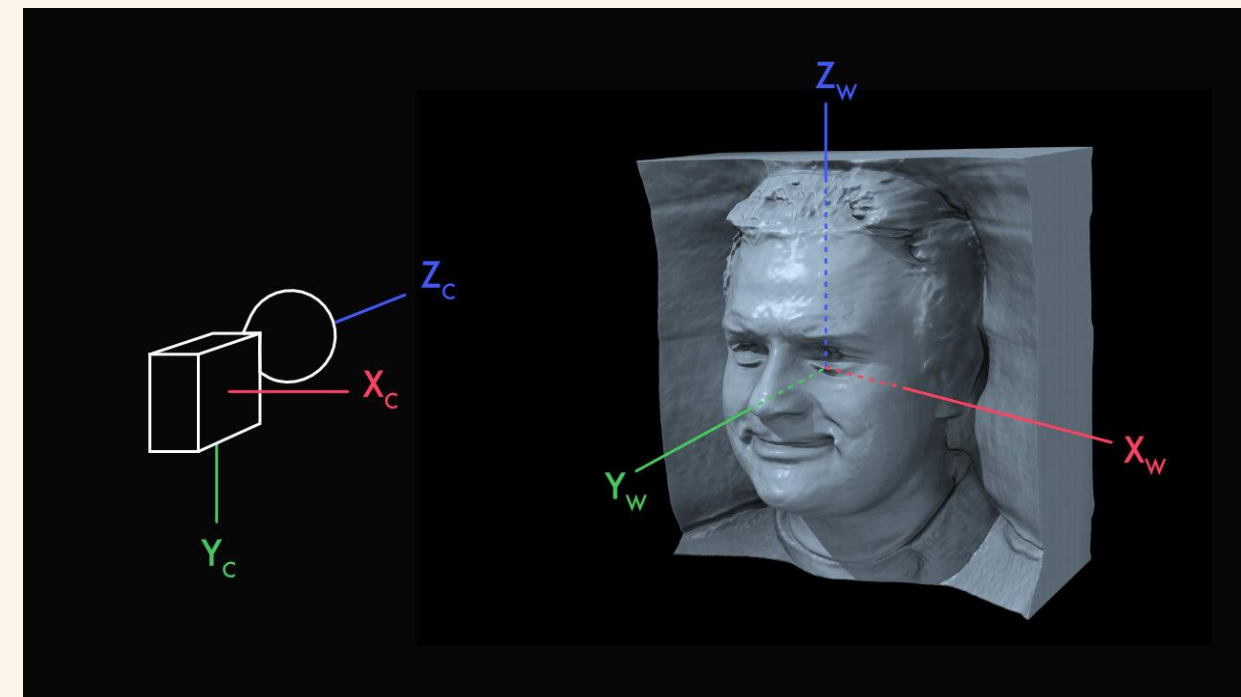
## Data Preprocessing: Camera parameters

이미지별로 존재하는 고유한 Camera Variable 값을 구해야 한다.

**Camera Variable** : 16개의 Extrinsic (위치) 변수 + 9개의 Intrinsic (스케일링) 변수



$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} fx \\ fy \\ z \end{bmatrix} \equiv \begin{bmatrix} fx/z \\ fy/z \\ 1 \end{bmatrix}$$



# GOAE: Geometry and Occlusion-Aware Encoding

## Data Preprocessing: Camera parameters

이미지별로 존재하는 고유한 Camera Variable 값을 구해야 한다.

SPI (CVPR 2023) preprocessing 함수 활용 해당 label 값 추출



```
model [FaceReconModel] was created
loading the model from checkpoints/model_name/epoch_20.pth
0%|          | 0/5 [00:00<?, ?it/s][ 0.99485397  0.02855997 -0.09721069  0.24687837  0.01311145 -0.98767161
-0.15598994  0.41645519 -0.10046732  0.15391265 -0.9829635  2.65624098
0.          0.          0.          1.          4.2647  0.
0.5         0.          4.2647  0.5         0.          0.
1.          ] SPI/test/dataset/c/1
```



# GOAE: Geometry and Occlusion-Aware Encoding

Data Preprocessing: Inference Results





# GOAE: Geometry and Occlusion-Aware Encoding

Data Preprocessing: Inference Results

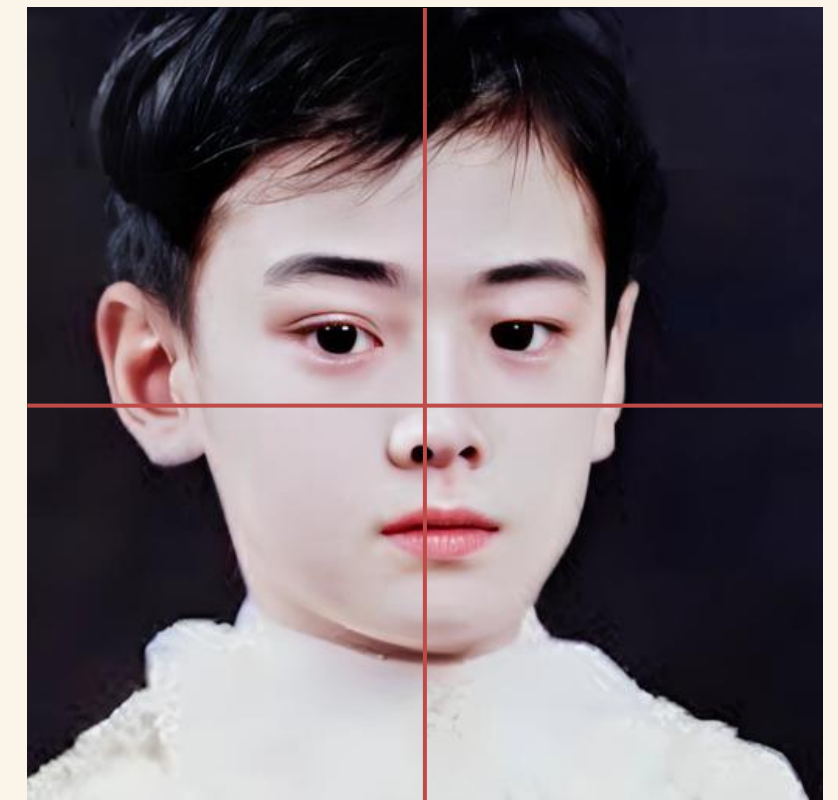
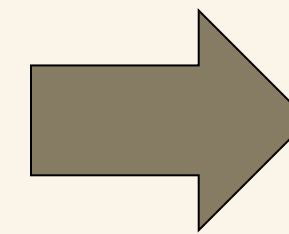


# GOAE: Geometry and Occlusion-Aware Encoding

## Data Preprocessing: FADING Outputs

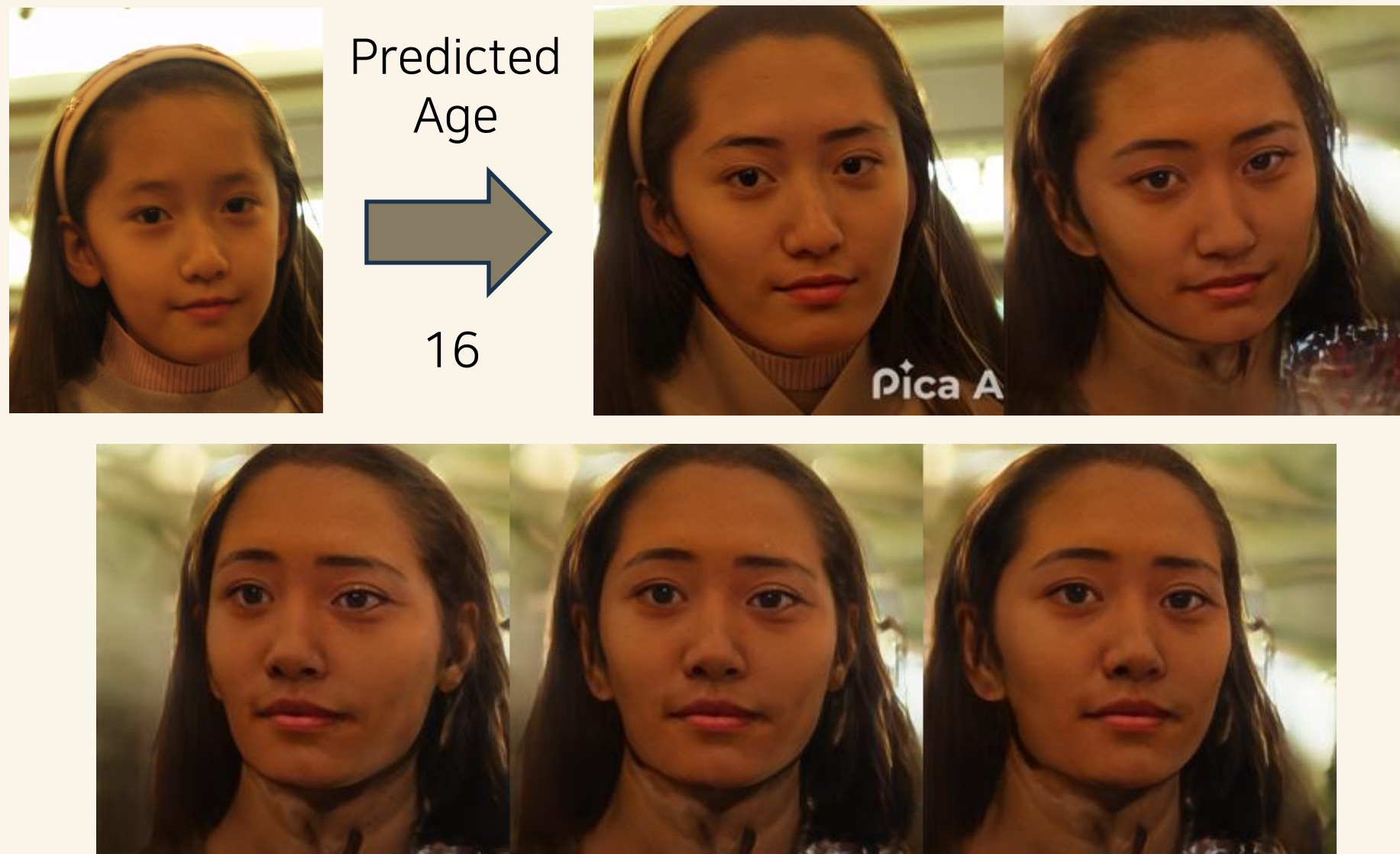
FAIDING을 통해 생성된 이미지: 부자연스럽게 뭉친 부분 + 화질 열화 → 랜더링 성능 저하 야기

Sol: Upscaling + cropping 활용한 화질 / 얼굴 중심 위치 조정



# GOAE: Geometry and Occlusion-Aware Encoding

Data Preprocessing: Inference Results for FADING





# #4 Results



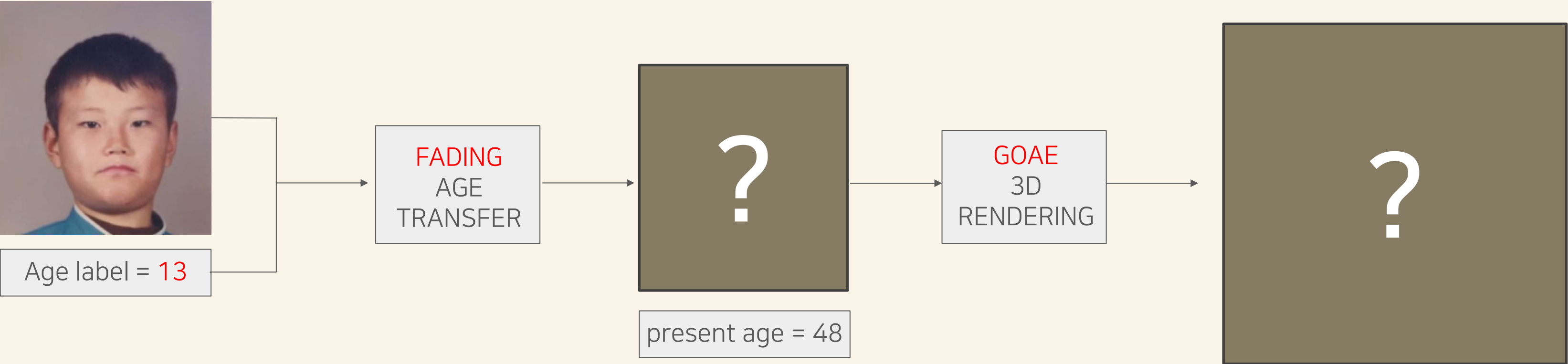
# Results

사회 > 사회일반

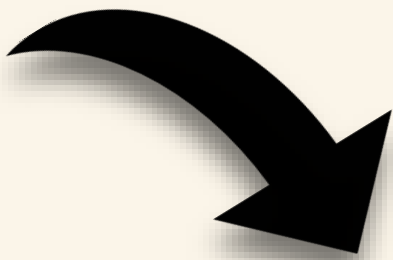
[잃어버린 가족찾기] 자전거 찾으러 나가 33년째 돌아오지 않는 아들

파이낸셜뉴스 입력 2017.12.25 19:10 수정 2017.12.25 19:10

가+ 가- ㄹ



# Results



Age label = 13

FADING  
AGE  
TRANSFER



present age = 48

GOAE  
3D  
RENDERING



# #5 Limitation



# Limitation

01

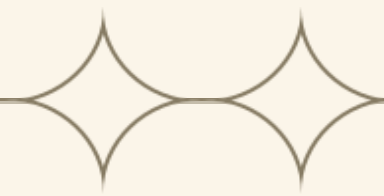
FADING - FADING 모델의 성능이 인풋 이미지의 화질, 배경의 물체 유무 등에 크게 영향을 받음  
⇒ 모델을 실제 실종아동 현재 얼굴 예측 문제에 일반적으로 적용하기는 어려움이 있음. 해당 부분 개선 필요

02

FADING - 정제된 동양인 얼굴 이미지 데이터셋 X, 동양인에 최적화된 모델 사용 불가능  
⇒ 고성능의 image resolution 모델을 사용해 custom dataset 생성 가능

03

FADING - EVALUATION METHODS 활용 X  
⇒ 추후 Pretrained age estimator를 사용해서 생성한 이미지가 해당 나이 feature를 잘 담고 있는지 평가 가능



# Limitation

04

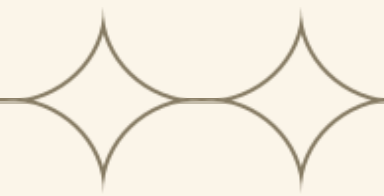
GOAE - 데이터 전처리 내 자동화 과정 부족

⇒ 목적에 맞는 Image Cropping / Upscaling / Camera Variable Labeling이 효율적으로 이뤄지도록 개선 가능

05

GOAE - 카메라 변수 추정과 재생성 과정에서의 성능 악화

⇒ 다양한 각도의 사진을 활용하는 모델이나 이전 단계에서 생성된 이미지 자체를 개선하는 방식으로 보완 가능





# **#6** References



# References

- [1] Chen, X., & Lathuilière, S. (2023, September 20). Face aging via diffusion-based editing. arXiv.org. <https://arxiv.org/abs/2309.11321>
- [2] Junseok-Rh. (2024, August 28). [평범한 학부생이 하는 논문 리뷰] Null-text Inversion for Editing Real Images using Guided Diffusion Models (CVPR 2023). 평범한 필기장. <https://juniboy97.tistory.com/81>
- [3] Deng, J., Guo, J., Zhou, Y., Yu, J., Kotsia, I., & Zafeiriou, S. (2019, May 2). *RetinaFace: Single-stage dense face localisation in the wild*. arXiv.org. <https://arxiv.org/abs/1905.00641>
- [4] Yuan, Z., Zhu, Y., Li, Y., Liu, H., & Yuan, C. (2023, March 22). *Make Encoder Great Again in 3D GAN Inversion through Geometry and Occlusion-Aware Encoding*. arXiv.org. <https://arxiv.org/abs/2303.12326>



# Thank You!

