Face-to-Parameter Translation for Game Character Auto-Creation

Tianyang Shi[1], Yi Yuan[1], Changjie Fan[1], Zhengxia Zou[2], Zhenwei Shi[3], Yong Liu[4]

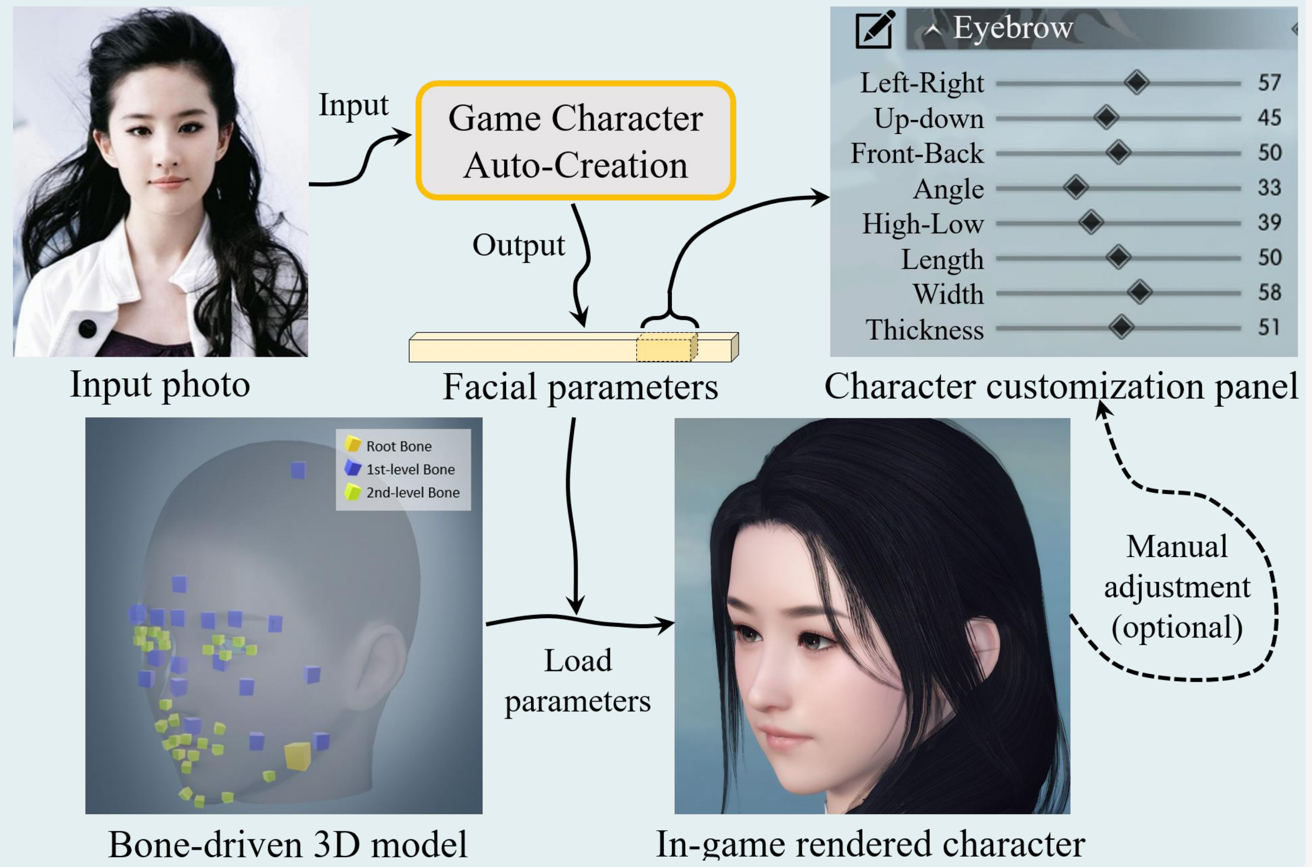
NetEase Fuxi AI Lab[1], University of Michigan[2], Beihang University[3], Zhejiang University[4]

Summary

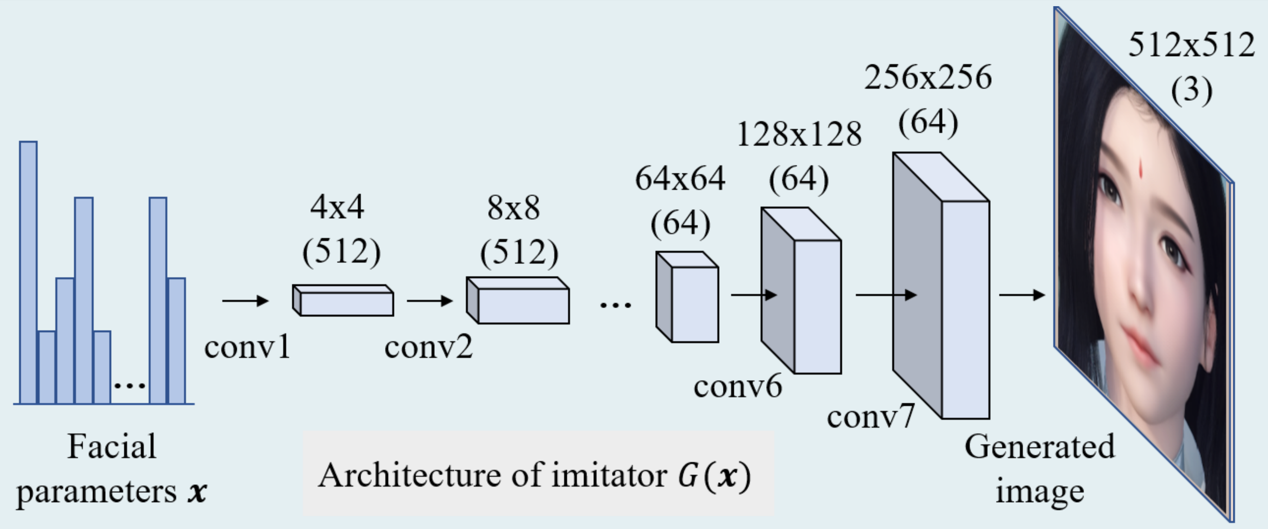
- **The character customization system** is an important component in RPGs, where players are allowed to edit the profiles of their in-game characters according to their own preferences.
- A player has to **spend several hours** manually **adjusting hundreds of parameters** to create a character with desired facial appearance.
- We aim to make the character creation **automatic** via a single photo.

Contributions:

1. We propose an end-to-end approach for **game character auto-creation.**
2. We introduce an **imitator** by constructing a deep generative network to imitate the behavior of a game engine and make it **differentiable**.
3. Discriminative loss and facial content loss are specifically designed for the cross-domain **facial similarity measurement**.

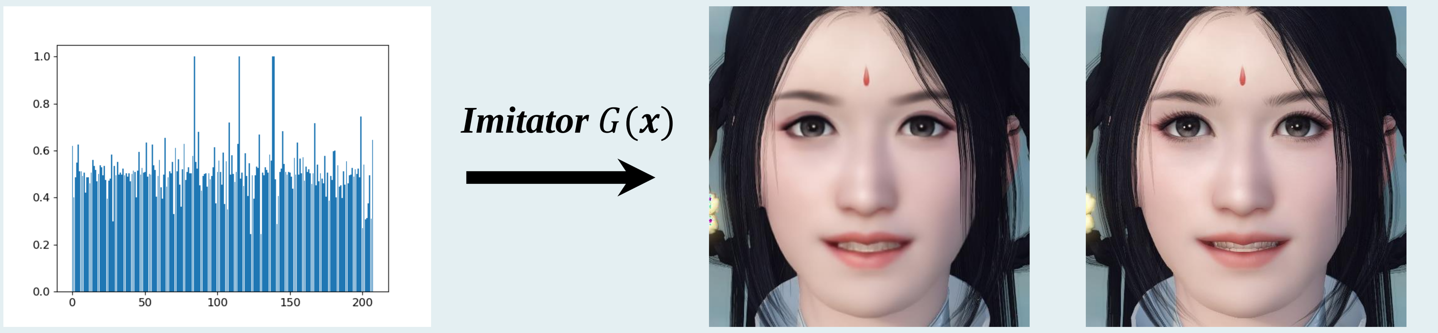


Imitator



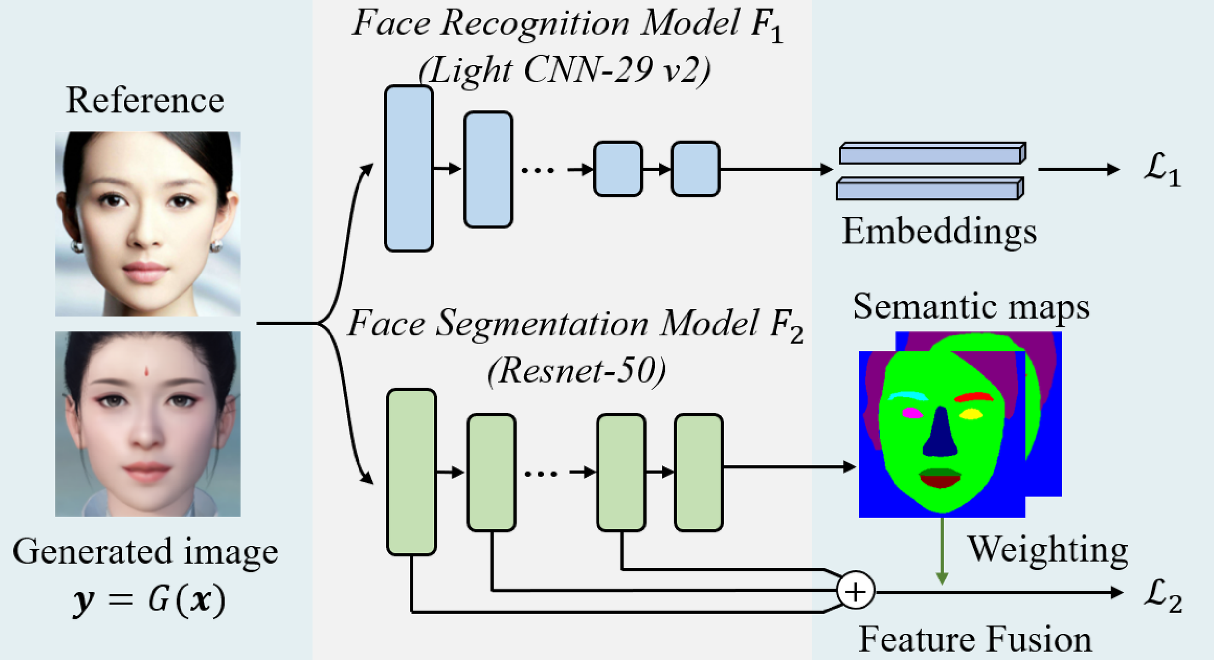
- We train a CNN as our imitator to fit the **input-output relationship** of a game engine.
- Similar to the configuration of DC-GAN [1], our imitator $G(x)$ consists of **eight transposed convolution layers**.
- To build the training dataset, we **randomly generate** 20,000 individual faces with their corresponding facial parameters by using the engine of the game "Justice".
- We frame the learning and prediction of the imitator as a standard deep learning based **regression problem**:

$$\mathcal{L}_G(x) = E_{x\sim u(x)}\{\|G(x) - \text{Engine}(x)\|_1\}$$



Input facial parameters x — Generated face image — Ground truth (in-game)

Facial Similarity Measurement



- Two kinds of loss functions are designed as measurements in terms of both **global facial appearance** and **local details.**
- We use a SOTA **face recognition model** "Light CNN-29 v2" [2] to extract the facial embeddings and then construct the discriminative loss based on their cosine distance:

$$\mathcal{L}_1(\boldsymbol{x}, \boldsymbol{y}_r) = 1 - \cos\left(F_1(G(\boldsymbol{x})), F_1(\boldsymbol{y}_r)\right)$$

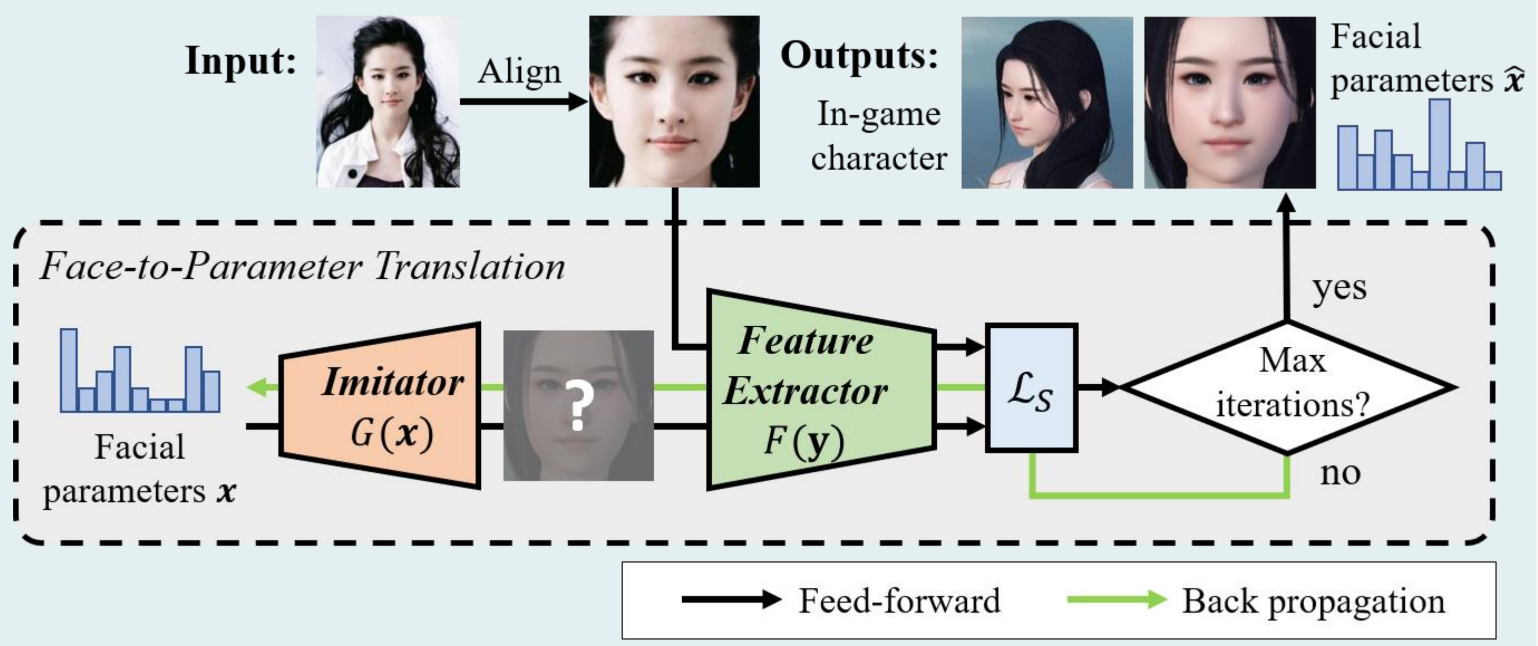
- We build a **face segmentation model** based on Resnet-50 [3] to extract facial features, of which weighted pixel-wise error is used to define the content loss:

$$\mathcal{L}_2(\boldsymbol{x}, \boldsymbol{y}_r) = \left\|\omega(\boldsymbol{x})F_2(G(\boldsymbol{x})) - \omega(\boldsymbol{y}_r)F_2(\boldsymbol{y}_r)\right\|_1$$

- Overall loss function:

$$\mathcal{L}_s(\boldsymbol{x}, \boldsymbol{y}_r) = \alpha\mathcal{L}_1 + \mathcal{L}_2$$

Face-to-Parameter Translation



- We use the **gradient descent** method to solve the following optimization problem:

$$\min_{\boldsymbol{x}} \mathcal{L}_s(\boldsymbol{x}, \boldsymbol{y}_r)$$
$$\text{s.t. } x_i \in [0,1]$$

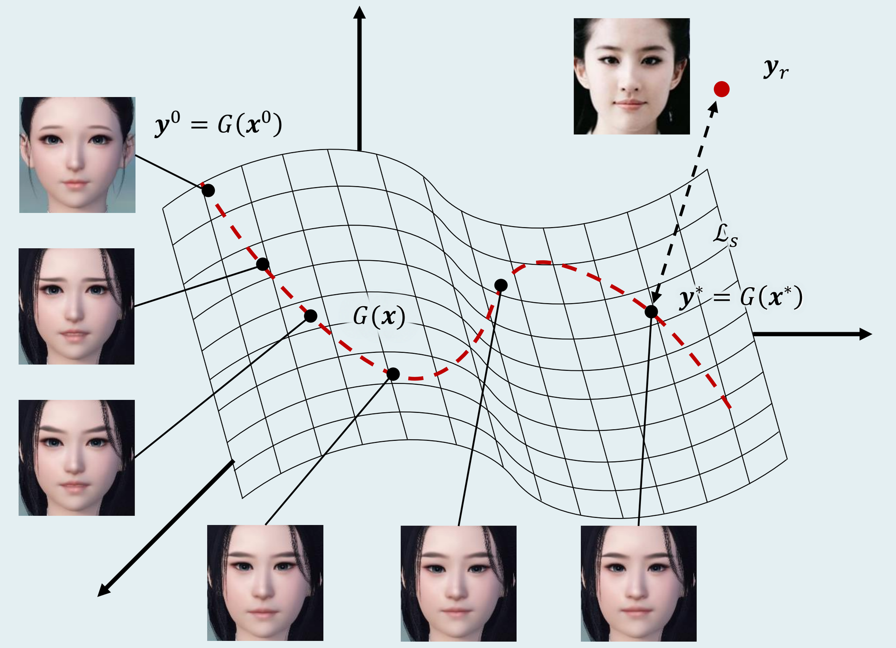
where $\boldsymbol{x}$ represents the facial parameters to be optimized and $\boldsymbol{y}_r$ represents an input reference facial photo.

- A complete optimization process of our method is summarized as follows:
  - **Stage I.** Train the imitator G, the face recognition network F_1 and the face segmentation network F_2.
  - **Stage II.** Fix G, F_1 and F_2, initialize and update facial parameters $\boldsymbol{x}$, until reach the max-number of iterations:

    $\boldsymbol{x} \leftarrow \boldsymbol{x} - \mu\frac{\partial\mathcal{L}_s}{\partial\boldsymbol{x}}$ (μ: learning rate)

    Project x_i to [0, 1]: $x_i \leftarrow \max(0, \min(x_i, 1))$

* The parameter generation can be considered as a **searching process** on the manifold of the imitator:



Game Character Auto-Creation

Input photo — Aligned input — Created character — Generated in-game character — Generated in-game character — Generated facial parameters

Auto-creation with artworks

Input sketch image — Generated character — Input Caricature — Generated character

Contact Us

fuxi.163.com/en/

References

[1] Alec Radford, *et al.*, "Unsupervised representation learning with deep convolutional generative adversarial networks", arXiv:1511.06434, 2015.

[2] Xiang Wu, *et al.*, "A light cnn for deep face representation with noisy labels", IEEE Transactions on Information Forensics and Security, 13(11):2884—2896, 2018.

[3] Kaiming He, *et al.*, "Deep residual learning for image recognition", in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.