

Supplementary Material

Thank you for reading the supplementary material, in which we introduce more details of experiments and SPSE. Moreover, we provide qualitative comparisons and synthesized results.

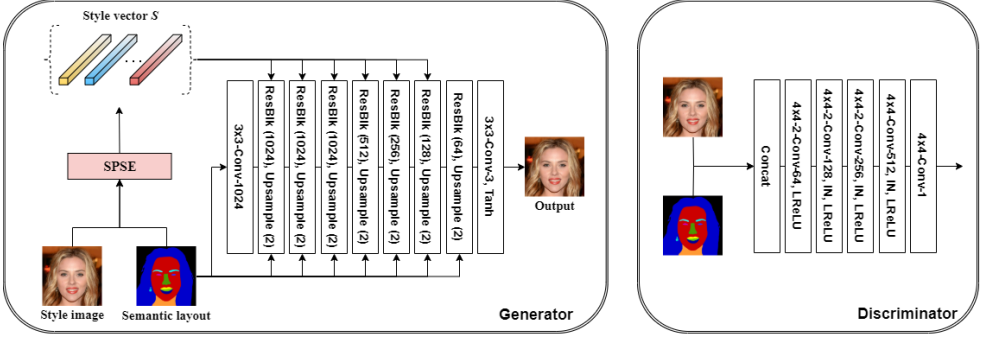


Figure 11: Whole framework of SuperStyleNet. (left) Structure of the generator. Details of ResBlk are described in Fig. 4. (right) Structure of the discriminator.

A. Additional Implementation Details

Generator: Adopting the SEAN generator [40], the architecture of our generator is comprised of a series of the residual blocks with nearest neighbor upsampling as shown in Fig. 11. We use $8 \times$ downsampled semantic layouts as input of the generator. The style vectors are generated by SPSE, and they are fed into each residual block excluding the last one. In contrast, the semantic layouts and noise are fed into all residual blocks.

Discriminator: Following the previous works [32, 45, 50], we employ two multi-scale discriminators, which consist of convolution layers with spectral normalization [49], instance normalization (IN) [40], and Leaky ReLU (LReLU) as illustrated in Fig. 11.

Learning objectives: To train the proposed network, the learning objectives are utilized as follows:

(1) *Perceptual loss:* We employ the VGG network [55] to calculate the perceptual loss [47]. Given a style image R and its corresponding masks M , let ϕ_i be feature maps of the i -th layer of the VGG network and N be the number of its layers. Then, the shape of the feature maps is $C_i \times H_i \times W_i$, and the perceptual loss is described as:

$$L_{\text{percept}} = \frac{1}{N} \sum_{i=1}^N \frac{1}{C_i H_i W_i} \sum_{c=1}^{C_i} \sum_{h=1}^{H_i} \sum_{w=1}^{W_i} \|\phi_i(R)_{c,h,w} - \phi_i(\mathbf{G}(SC, M))_{c,h,w}\|_1, \quad (5)$$

where \mathbf{G} is the generator, and SC is extracted style codes from the style image.

(2) *Feature matching loss:* Inspired by Pix2PixHD [45], we apply the feature matching loss to our network. Let D_i^k be feature maps of the i -th layer of multi-scale discriminators D^1 and D^2 , and M_k be the number of layers in these discriminator, then the feature matching loss is formulated as:

$$L_{FM} = \frac{1}{M_k} \sum_{i=1}^{M_k} \frac{1}{C_i H_i W_i} \sum_{c=1}^{C_i} \sum_{h=1}^{H_i} \sum_{w=1}^{W_i} \left\| D_i^k(R, M)_{c,h,w} - D_i^k(\mathbf{G}(SC, M), M)_{c,h,w} \right\|_1. \quad (6)$$

(3) *Adversarial loss*: We adopt the hinge loss term [29, 48] for adversarial learning. Then, the learning objective is formulated as:

$$\begin{aligned} L_{GAN} &= -\mathbb{E}[\min(0, -1 + D_k(R, M))] - \mathbb{E}[\min(0, -1 - D_k(\mathbf{G}(SC, M), M))], \\ L_{ad} &= -\mathbb{E}[D_k(\mathbf{G}(SC, M), M)], \end{aligned} \quad (7)$$

where L_{GAN} and L_{ad} are for the discriminator and the generator, respectively.

Overall, the final loss function for SuperStyleNet is described as:

$$L_{total} = \alpha L_{percept} + \sum_{k=1,2} (\beta L_{FM} + L_{ad}), \quad (8)$$

where α and β are the weight parameters of individual loss terms, and both are set to 10, respectively.

B. Additional Details of SPSE

Algorithm: In this paper, we propose superpixel based parameter-free style encoding (SPSE) to encapsulate the input image in the style codes as described in Algorithm 1. To be specific, we convert the RGB input image $I \in \mathbb{R}^{n \times 3}$ to a five-dimensional color and position space ($LabXY$) $\hat{I} \in \mathbb{R}^{n \times 5}$, where n is the number of pixels. After that, we initialize k superpixel centers $S_l^0 \in \mathbb{R}^{k \times 5}$ in a given semantic mask M_l using an uniform distribution [49], and extract pixels p_l from a converted input \hat{I} in the given semantic mask M_l . Then, we compute the association map $Q \in \{0, \dots, k-1\}^{n \times 1}$ between the pixels and nearest superpixel centers by computing the distance during iteration t :

$$Q_{p_l}^t = \arg \min_{i \in \{0, \dots, k-1\}} D(\hat{I}_{p_l}, S_i^{t-1}), \quad (9)$$

where D is the Euclidean distance. To update new superpixel centers in each iteration, we take average pixel features of the converted input \hat{I} inside each superpixel cluster i :

$$S_{i_l}^t = \frac{1}{Z_{i_l}^t} \sum_{p_l | Q_{p_l}^t = i_l} \hat{I}_{p_l}, \quad (10)$$

where $Z_{i_l}^t$ is the number of pixels in the superpixel cluster i_l . After proceeding all iterations, we obtain the final association maps Q' and superpixel centers S' in the given semantic label. To convert the superpixel centers to the style codes $SC \in \mathbb{R}^{k \times 3}$, we repeat Eq. 10 with the original input I :

$$SC_{i_l} = \frac{1}{Z_{i_l}} \sum_{p_l | Q_{p_l} = i_l} I_{p_l}. \quad (11)$$

Finally, the style codes are reshaped to $3k$, and interpolated with the desired length N of the style ones.

C. Additional Results

To validate the effectiveness of SuperStyleNet, We provide more qualitative comparison of semantic image synthesis on CelebAMask-HQ, Cityscapes, and CMP Facades in Fig. 12 and Fig. 13. Furthermore, we combine multiple style images on given segmentation masks to generate style mixed images as shown in Fig. 14.

Algorithm 1 Superpixel based Parameter-free Style Encoding (SPSE)**Input:** Image I , Semantic labels L .**Output :** Style codes SC per semantic mask.

- 1: Convert input I to the $LabXY$ space \hat{I} .
- 2: **for** each semantic label $l \in L$ **do**
- 3: Initialize superpixel centers S_l^0 .
- 4: Extract pixels p_l from \hat{I} in a semantic mask M_l .
- 5: **for** each iteration t **do**
- 6: Compute association between each pixel p_l and the nearest superpixel i_l , $Q_{p_l}^t = \arg \min_{i \in \{0, \dots, k-1\}} D(\hat{I}_{p_l}, S_i^{t-1})$.
- 7: Compute new superpixel centers, $S_{i_l}^t = \frac{1}{Z_{i_l}^t} \sum_{p_l | Q_{p_l}^t = i_l} \hat{I}_{p_l}$.
- 8: **end for**
- 9: Convert the final superpixel centers to the style code, $SC_{i_l} = \frac{1}{Z_{i_l}} \sum_{p_l | Q_{p_l} = i_l} I_{p_l}$.
- 10: **end for**

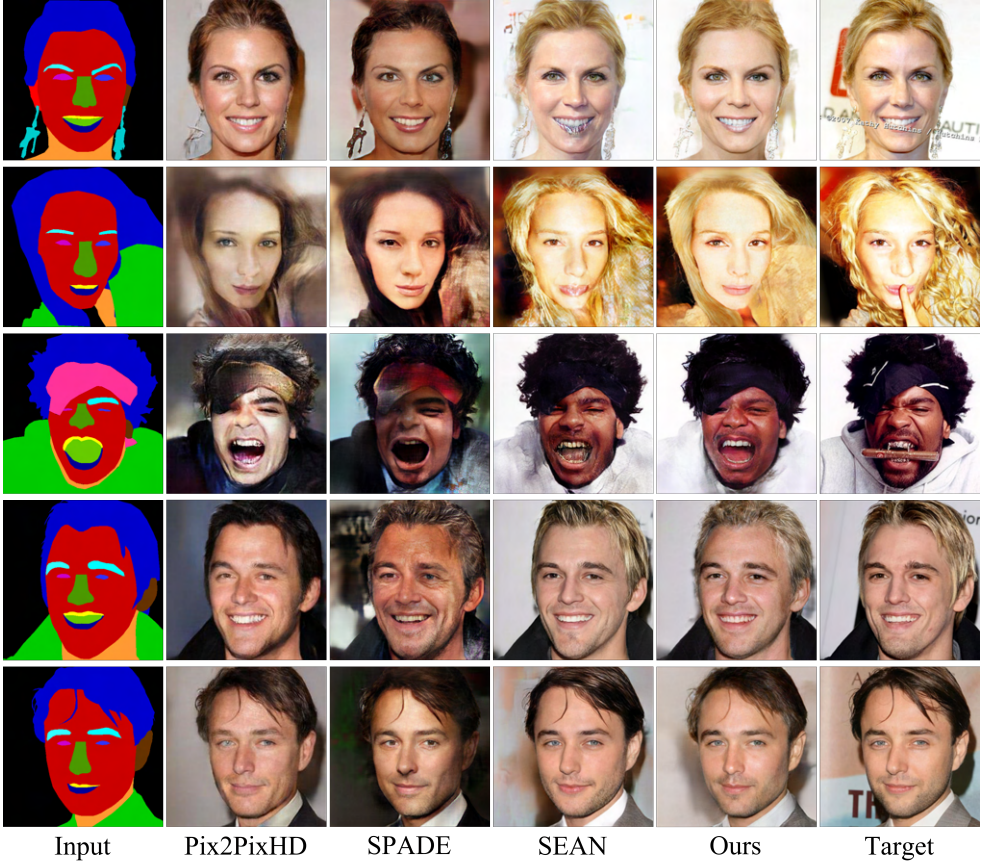


Figure 12: Qualitative comparison of semantic image synthesis on CelebAMask-HQ.

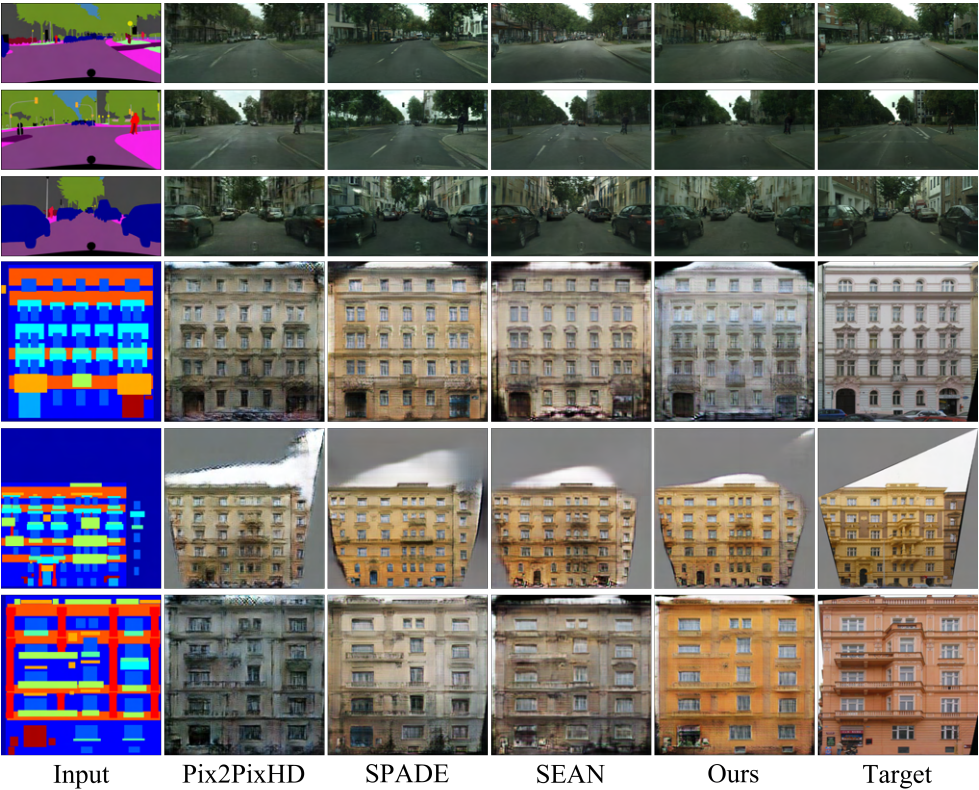


Figure 13: Qualitative comparison of semantic image synthesis on Cityscapes and CMP Facades.



Figure 14: Style mixing with multiple style images on given segmentation masks.