# Stylization

2018~2020 loss function 정리

201810957 염지현

# Contents

# Contents

## 1. Total loss

$$L(G, F, D_{\mathcal{D}}, D_{\mathcal{P}}):$$

$$\boxed{\min_{G,F}}\ \boxed{\max_{D_{\mathcal{D}},D_{\mathcal{P}}}} L(G, F, D_{\mathcal{D}}, D_{\mathcal{P}})$$

$$= (L_{adv}(G, D_{\mathcal{D}}) + L_{adv}(F, D_{\mathcal{P}})) + \lambda_1 L_{relaxed-cyc}(G, F)$$

$$+ \lambda_2 L_{strict-cyc}(G, F) + \lambda_3 L_{trunc}(G, F) + \lambda_4 L_{style}(G, D_{\mathcal{D}})$$

# 01 Unpaired Portrait Drawing Generation via Asymmetric Cycle Mapping

## 2. Adversarial loss

$$L_{adv}(G, D_{\mathcal{D}}) = \sum_{D \in D_{\mathcal{D}}} \mathbb{E}_{d \in S(d)}[\log D(d)] + \sum_{D \in D_{\mathcal{D}}} \mathbb{E}_{p \in S(p)}[\log(1 - D(G(p, s))]$$

$$L_{adv}(F, D_{\mathcal{P}}) = \mathbb{E}_{p \in S(p)}[\log D_{\mathcal{P}}(p)] + \mathbb{E}_{d \in S(d)}[\log(1 - D_{\mathcal{P}}(F(d))]$$

### 3. Relaxed forward cycle-consistency loss

$$L_{relaxed-cyc}(G, F) = \mathbb{E}_{p \in S(p)}[L_{lpips}(H(p), H(F(G(p, s))))]$$

### 4. Strict backward cycle-consistency loss

$$L_{strict-cyc}(G, F) = \mathbb{E}_{d \in S(d)}[||d - G(F(d), s(d))||_1]$$

## 5. Truncation loss

$$L_{trunc}(G,F) = \mathbb{E}_{p \in S(p)}[L_{lpips}(H(p), H(F(T[G(p,s)])))$$

## 6. Style loss

$$L_{cls}(G, D_{\mathcal{D}}) = \mathbb{E}_{d \in S(d)}[-\sum_{c} p(c) \log D_{cls}(c|d)] + \mathbb{E}_{p \in S(p)}[-\sum_{c} p'(c) \log D_{cls}(c|G(p,s))]$$

**Truncation loss.** The truncation loss is designed to prevent the generated drawing from hiding information in small values. It is in the same format as the relaxed cycle-consistency loss, except that the generated drawing $G(p,s)$ is first truncated to 6 bits (a general digital image stores intensity in 8 bits) to ensure encoded information is clearly visible, and then fed into $F$ to reconstruct the photo. Denote the truncation operation as $T[\cdot]$, the truncation loss is formulated as:

$$L_{trunc}(G,F) = \mathbb{E}_{p \in S(p)}[L_{lpips}(H(p), H(F(T[G(p,s)])))] \quad (6)$$

In the first period of training, the weight for the truncation loss is kept low, otherwise it would be too hard for the model to optimize. The weight gradually increases as the training progresses.

# 02 Adjustable Real-time Style Transfer

1. Total loss

$$\min_{\theta} \left( \mathcal{L}_c(T(\mathbf{c})) + \mathcal{L}_s(T(\mathbf{c})) \right)$$

2. Content loss & Style loss

$$\mathcal{L}_c(\mathbf{p}) = \sum_{l \in C} \alpha_c^l \mathcal{L}_c^l(\mathbf{p}) \text{ and } \mathcal{L}_s(\mathbf{p}) = \sum_{l \in S} \alpha_s^l \mathcal{L}_s^l(\mathbf{p})$$

$$\mathcal{L}_c^l(\mathbf{p}) = ||\phi^l(\mathbf{p}) - \phi^l(\mathbf{s})||_2^2$$

$$\mathcal{L}_s^l(\mathbf{p}) = ||G(\phi^l(\mathbf{p})) - G(\phi^l(\mathbf{s}))||_F^2$$

$$\mathbf{p} = \Psi(\mathbf{c}, \mathbf{s}, \boldsymbol{\alpha}_c, \boldsymbol{\alpha}_s)$$ → Stylized image

# 03 Deformable Style Transfer

## 1. Total loss

$$
\begin{aligned}
L(X, \theta, I_c, I_s, P, P') = \alpha L_{\text{content}}(I_c, X) \\
&+ L_{\text{style}}(I_s, X) + L_{\text{style}}(I_s, W(X, \theta)) \\
&+ \beta L_{\text{warp}}(P, P', \theta) \\
&+ \gamma R_{\text{TV}}(f_\theta),
\end{aligned}
$$

## 2. Style loss

$$L_{\text{style}}(I_s, X) + L_{\text{style}}(I_s, W(X, \theta)).$$

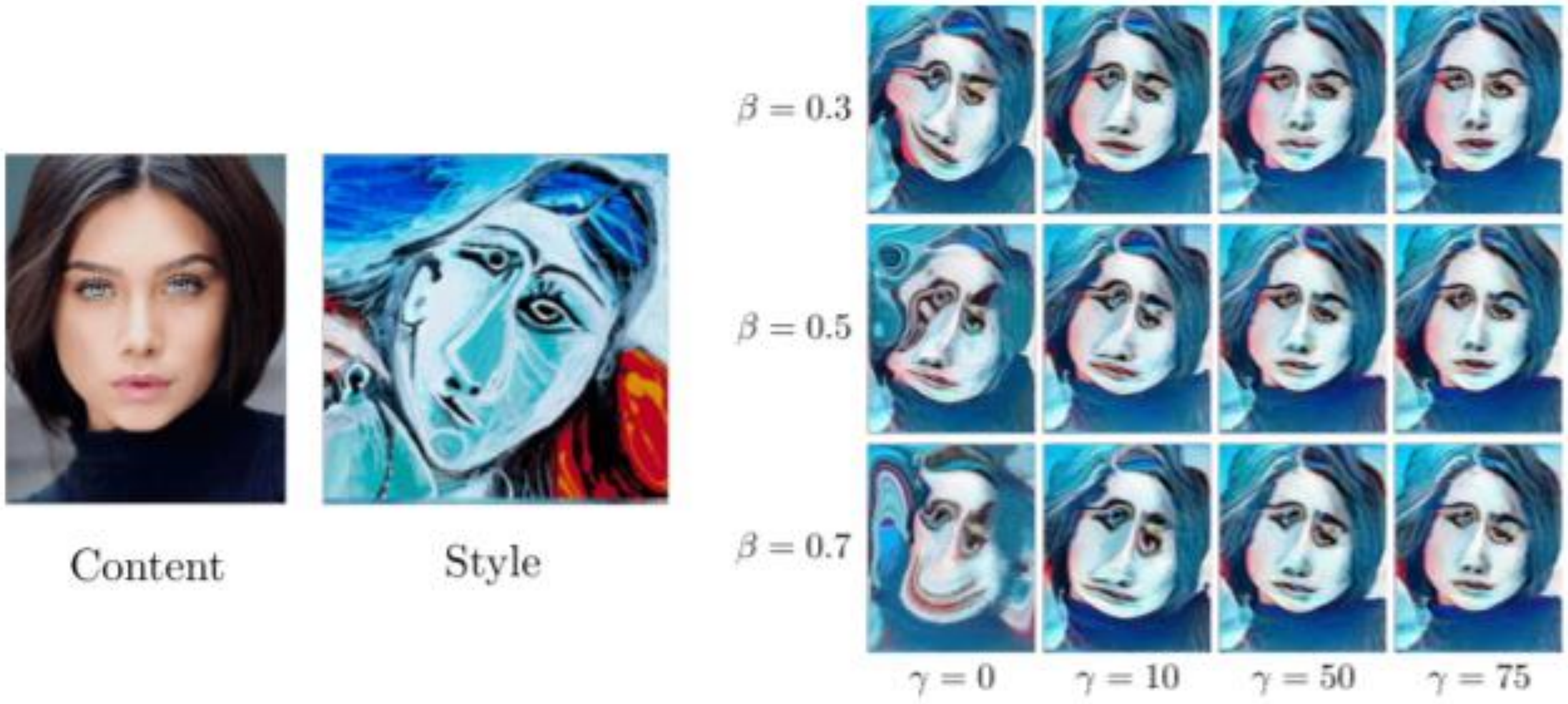
## 3. Deformation loss

$$L_{\text{warp}}(P, P', \theta) = \frac{1}{k} \sum_{i=1}^{k} \| p'_i - (p_i + \theta_i) \|_2,$$

## +) Regularization term

$$R_{\text{TV}}(f) = \frac{1}{\mathrm{W} \times \mathrm{H}} \sum_{i=1}^{\mathrm{W}} \sum_{j=1}^{\mathrm{H}} \| f_{i+1,j} - f_{i,j} \|_1 + \| f_{i,j+1} - f_{i,j} \|_1.$$

# 03 Deformable Style Transfer



# Learning to Cartoonize Using White-Box Cartoon Representations

## 1. Surface loss

$$\mathcal{L}_{surface}(G,\ D_s) = logD_s(\mathcal{F}_{dgf}(\boldsymbol{I}_c, \boldsymbol{I}_c)) + log(1 - D_s(\mathcal{F}_{dgf}(G(\boldsymbol{I}_p), G(\boldsymbol{I}_p))))$$

## 2. Structure loss

$$\mathcal{L}_{structure} = \|VGG_n(G(\boldsymbol{I}_p)) - VGG_n(\mathcal{F}_{st}(G(\boldsymbol{I}_p)))\|$$

## 3. Texture loss

$$\mathcal{L}_{texture}(G, D_t) = logD_t(\mathcal{F}_{rcs}(\boldsymbol{I}_c)) + log(1 - D_t(\mathcal{F}_{rcs}(G(\boldsymbol{I}_p))))$$

$$\mathcal{F}_{rcs}(\boldsymbol{I}_{rgb}) = (1-\alpha)(\beta_1 * \boldsymbol{I}_r + \beta 2 * \boldsymbol{I}_g + \beta_3 * \boldsymbol{I}_b) + \alpha * \boldsymbol{Y}$$

## 4. Content loss

$$\mathcal{L}_{content} = \|VGG_n(G(\boldsymbol{I}_p)) - VGG_n(\boldsymbol{I}_p)\|$$

## +) regularization

$$\mathcal{L}_{tv} = \frac{1}{H * W * C}\| \bigtriangledown_x (G(\boldsymbol{I}_p)) + \bigtriangledown_y (G(\boldsymbol{I}_p))\|$$

## 5. Total loss

$$\mathcal{L}_{total} = \lambda_1 * \mathcal{L}_{surface} + \lambda_2 * \mathcal{L}_{texture} + \lambda_3 * \mathcal{L}_{structure} + \lambda_4 * \mathcal{L}_{content} + \lambda_5 * \mathcal{L}_{tv}$$

# 05 U-GAT-IT: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation

## 1. Adversarial loss

$$L_{lsgan}^{s\rightarrow t} = (\mathbb{E}_{x\sim X_t}[(D_t(x))^2] + \mathbb{E}_{x\sim X_s}[(1 - D_t(G_{s\rightarrow t}(x)))^2]).$$

## 2. Cycle loss

$$L_{cycle}^{s\rightarrow t} = \mathbb{E}_{x\sim X_s}[|x - G_{t\rightarrow s}(G_{s\rightarrow t}(x)))|_1].$$

## 3. Identity loss

$$L_{identity}^{s\rightarrow t} = \mathbb{E}_{x\sim X_t}[|x - G_{s\rightarrow t}(x)|_1].$$

# 05 U-GAT-IT: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation

## 4. CAM loss

$$L_{cam}^{s \to t} = -(\mathbb{E}_{x \sim X_s}[log(\eta_s(x))] + \mathbb{E}_{x \sim X_t}[log(1 - \eta_s(x))]),$$

$$L_{cam}^{D_t} = \mathbb{E}_{x \sim X_t}[(\eta_{D_t}(x))^2] + \mathbb{E}_{x \sim X_s}[(1 - \eta_{D_t}(G_{s \to t}(x))^2].$$

## 5. Total loss

$$\min_{G_{s \to t}, G_{t \to s}, \eta_s, \eta_t} \max_{D_s, D_t, \eta_{D_s}, \eta_{D_t}} \lambda_1 L_{lsgan} + \lambda_2 L_{cycle} + \lambda_3 L_{identity} + \lambda_4 L_{cam},$$

# 06 Neural Style Transfer via Meta Networks

## 1. Total loss

$$\min_{\theta} \sum_{I_c \in \mathcal{D}_c} \sum_{I_s \in \mathcal{D}_s} \mathbf{PLoss}(I_x | I_c, I_s),$$

$$I_x = \mathcal{N}(I_c; w_\theta)$$

$$w_\theta = Meta\mathcal{N}(I_s; \theta)$$

$$\lambda_c ||\mathbf{CP}(\boxed{I}) - \mathbf{CP}(I_c)||_2^2 + \lambda_s ||\mathbf{SP}(I) - \mathbf{SP}(I_s)||_2^2$$

I : transferred image

# 07 Arbitrary Style Transfer With Deep Feature Reshuffle

1. Total loss

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{cont} + (1-\alpha)\mathcal{L}_{shuf}.$$

2. Content loss

$$\mathcal{L}_{cont} = ||F_o - F_c||_{\mathrm{F}}^2,$$

3. Reshuffle loss

$$\mathcal{L}_{shuf} = \sum\nolimits_p ||\Psi_p(F_o) - \Psi_{\mathrm{NNC}(p)}(F_s)||_{\mathrm{F}}^2,$$

$$\mathrm{NNC}(p) = \mathop{\arg\max}_{p'=1,2,\ldots,\Theta} \Big(\frac{\Psi_p(F_o)\cdot\Psi_{p'}(F_s)}{||\Psi_p(F_o)||_{\mathrm{F}}^2\cdot||\Psi_{p'}(F_s)||_{\mathrm{F}}^2} - \lambda\frac{\Gamma(\Psi_{p'}(F_s))}{R\times R}\Big),$$

# Avatar-Net: Multi-Scale Zero-Shot Style Transfer by Feature Decoration

**1. Total loss**

$$\ell_{\text{total}} = \|\mathbf{x} - \tilde{\mathbf{x}}\|_2^2 + \lambda_1 \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \|\boldsymbol{\Psi}_{\text{VGG}}^i(\mathbf{x}) - \boldsymbol{\Psi}_{\text{VGG}}^i(\tilde{\mathbf{x}})\|_2^2 + \lambda_2 \ell_{\text{TV}}(\mathbf{x}),$$

# 09 Stroke Controllable Fast Style Transfer with Adaptive Receptive Fields

**1. Total loss**

$$\mathcal{L}_{\mathcal{B}_{s_k}} = \alpha\mathcal{L}_c + \beta_k\mathcal{L}_{\mathcal{T}_k} + \gamma\mathcal{L}_{tv},$$

**2. Semantic loss**

$$\mathcal{L}_c = \sum\nolimits_{l\in\{l_c\}} \|\mathcal{F}^l(I_c) - \mathcal{F}^l(I_o)\|^2,$$

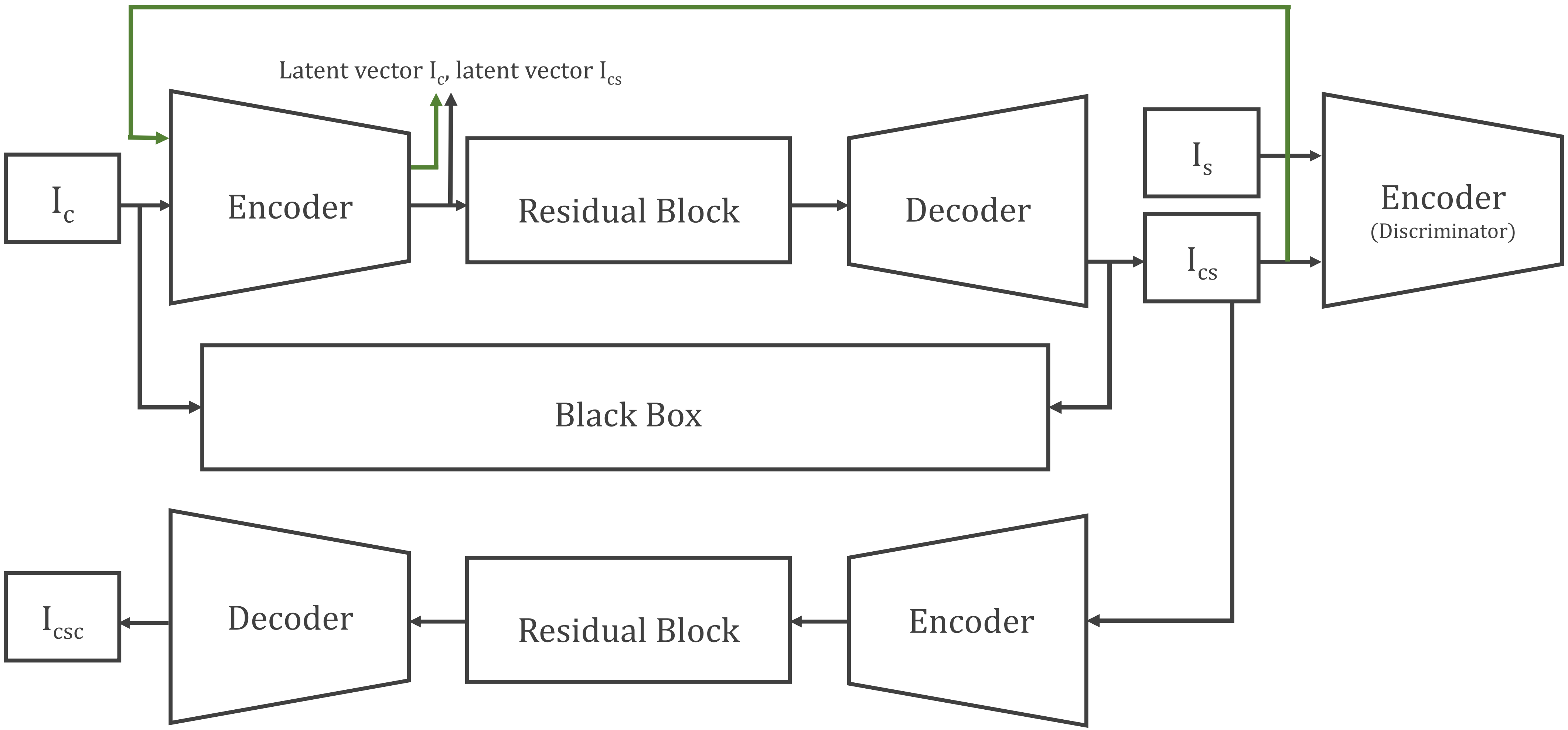
**3. Stroke loss**

$$\mathcal{L}_{\mathcal{T}_k} = \sum\nolimits_{l\in\{l_s\}} \|\mathcal{G}(\mathcal{F}^l(\mathcal{R}(I_s, \mathcal{T}_k))') - \mathcal{G}(\mathcal{F}^l(I_o^{\mathcal{B}_{s_k}})')\|^2,$$

$$\mathcal{G}(\mathcal{F}^l(I_s)') = [\mathcal{F}^l(I_s)'][\mathcal{F}^l(I_s)']^T$$

$$\mathcal{F}^l(I)' \in \mathbb{R}^{C\times(H\times W)}$$

# 10 A Style-Aware Content Loss for Real-time HD Style Transfer



# A Style-Aware Content Loss for Real-time HD Style Transfer

1. Optimization problem

$$\mathcal{L}(E,G,D) = \mathcal{L}_c(E,G) + \mathcal{L}_t(E,G) + \lambda\mathcal{L}_D(E,G,D)$$

2. Total loss

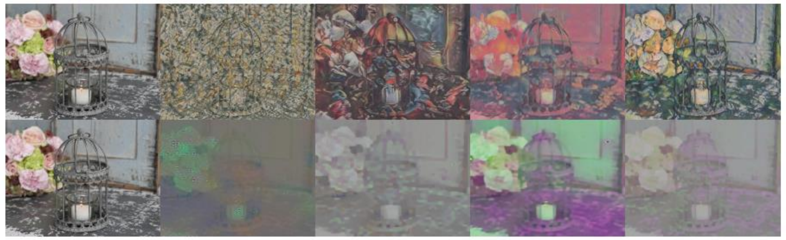
$$E,G = \arg\min_{E,G}\max_{D}\mathcal{L}(E,G,D).$$

### 3. Adversarial loss

$$\mathcal{L}_D(E,G,D) = \mathop{\mathbb{E}}_{y\sim p_Y(y)}[\log D(y)] + \mathop{\mathbb{E}}_{x\sim p_X(x)}[\log(1 - D(G(E(x))))]$$

### 4. style-aware content loss

$$\mathcal{L}_c(E,G) = \mathop{\mathbb{E}}_{x\sim p_X(x)}\left[\frac{1}{d}\|E(x) - E(G(E(x)))\|_2^2\right]$$

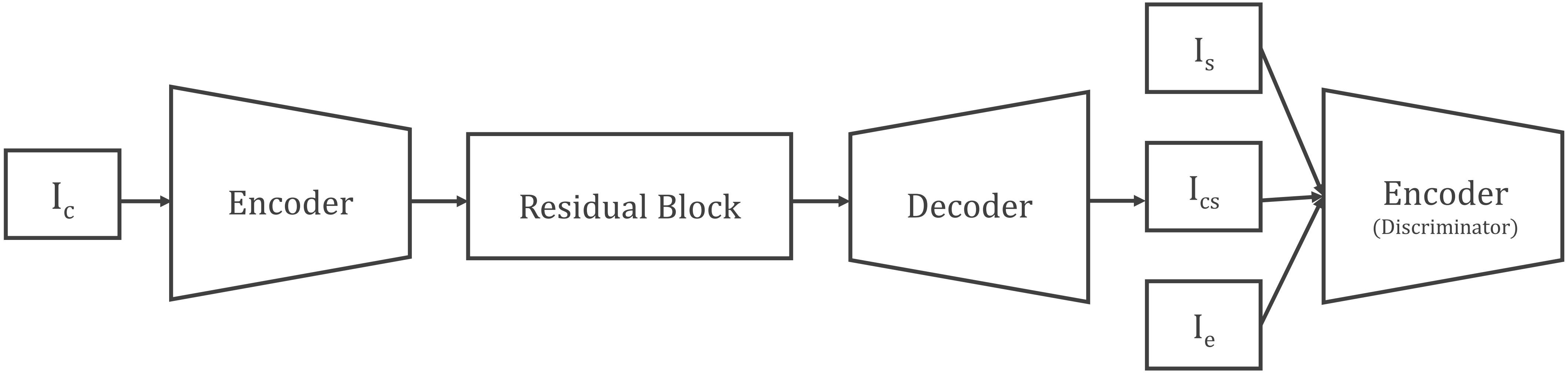



### 5. Transformed image loss

$$\mathcal{L}_{\mathbf{T}}(E,G) = \mathop{\mathbb{E}}_{x\sim p_X(x)}\left[\frac{1}{CHW}||\mathbf{T}(x) - \mathbf{T}(G(E(x))||_2^2\right]$$


**Fig. 10.** Different variations of our method for Gauguin stylization. See Sect. 3.3 for details. (a) Content image; (b) full model ($\mathcal{L}_c$, $\mathcal{L}_{rgb}$ and $\mathcal{L}_D$); (c) $\mathcal{L}_{rgb}$ and $\mathcal{L}_D$; (d) without transformer block; (e) only $\mathcal{L}_D$; (f) trained with all of Gauguin's artworks as style images. Please zoom in to compare.

# CartoonGAN: Generative Adversarial Networks for Photo Cartoonization



# CartoonGAN: Generative Adversarial Networks for Photo Cartoonization

**1. Optimization problem**

$$(G^*, D^*) = \arg\min_G \max_D \mathcal{L}(G, D)$$

**2. Total loss**

$$\mathcal{L}(G, D) = \mathcal{L}_{adv}(G, D) + \omega \mathcal{L}_{con}(G, D),$$

**3. Adversarial loss**

$$\begin{aligned}\mathcal{L}_{adv}(G, D) = \ & \mathbb{E}_{c_i \sim S_{data}(c)}[\log D(c_i)] \\ & + \mathbb{E}_{e_j \sim S_{data}(e)}[\log(1 - D(e_j))] \\ & + \mathbb{E}_{p_k \sim S_{data}(p)}[\log(1 - D(G(p_k)))].\end{aligned}$$

usually very small in the whole image. Therefore, an output image without clearly reproduced edges but with correct shading is likely to confuse the discriminator trained with a standard loss.

**4. Content loss**

$$\mathcal{L}_{con}(G, D) = \mathbb{E}_{p_i \sim S_{data}(p)}[||VGG_l(G(p_i)) - VGG_l(p_i)||_1]$$

# Style Transfer by Relaxed Optimal Transport and Self-Similarity

Stylization에 대한 content preserving의 상대적 중요성

**1. Total loss**

$$L(X, I_C, I_S) = \frac{\boxed{\alpha}\ell_C + \ell_m + \ell_r + \frac{1}{\alpha}\ell_p}{2 + \alpha + \frac{1}{\alpha}}$$

**2. Content loss**

$$\mathcal{L}_{content}(X, C) = \frac{1}{n^2}\sum_{i,j}\left|\frac{D_{ij}^{X}}{\sum_i D_{ij}^{X}} - \frac{D_{ij}^{I_C}}{\sum_i D_{ij}^{I_C}}\right|$$

$$C_{ij} = \begin{cases} \beta * D_{cos}(A_i, B_j), \text{ if } i \in X_{tk}, j \in S_{sk} \\ \infty, \text{ if } \exists k \text{ s.t. } i \in X_{tk}, j \notin S_{sk} \\ D_{cos}(A_i, B_j) \text{ otherwise,} \end{cases}$$

## Style Transfer by Relaxed Optimal Transport and Self-Similarity

### 3. REMD loss

$$\mathrm{EMD}(A,B) = \min_{T \geq 0} \sum_{ij} T_{ij} C_{ij}$$

$$s.t. \sum_j T_{ij} = 1/m$$

$$\sum_i T_{ij} = 1/n$$

$$\downarrow$$

$$\ell_r = REMD(A,B) = \max(R_A(A,B), R_B(A,B))$$

$$R_A(A,B) = \min_{T \geq 0} \sum_{ij} T_{ij} C_{ij} \quad s.t. \quad \sum_j T_{ij} = 1/m$$

$$R_B(A,B) = \min_{T \geq 0} \sum_{ij} T_{ij} C_{ij} \quad s.t. \quad \sum_i T_{ij} = 1/n$$

$$\downarrow$$

$$\ell_r = \max\left(\frac{1}{n}\sum_i \min_j C_{ij}, \frac{1}{m}\sum_j \min_i C_{ij}\right)$$

$$C_{ij} = D_{\cos}(A_i, B_j) = 1 - \frac{A_i \cdot B_j}{\|A_i\|\|B_j\|}$$

# Style Transfer by Relaxed Optimal Transport and Self-Similarity

**4. Moment matching loss**

$$\ell_m = \frac{1}{d}\|\mu_A - \mu_B\|_1 + \frac{1}{d^2}\|\Sigma_A - \Sigma_B\|_1$$

**5. Color matching loss**

$$\ell_p$$

- 생성된 image와 style image 사이의 pixel colors distance를 relaxed EMD 로 사용하여 측정
- 팔레트의 이동과 content 보존은 서로 상충 → 따라서 $1/\alpha$ 가중

# Thank You