Stylized-Colorization for Line Arts

Tzu-Ting Fang[1] Duc Minh Vo[2] Akihiro Sugimoto[3] Shang-Hong Lai[1]

[1]National Tsing Hua University
Hsinchu, Taiwan

[2]SOKENDAI
Tokyo, Japan

[3]National Institute of Informatics
Tokyo, Japan

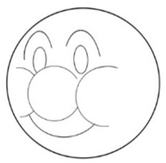
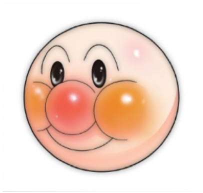
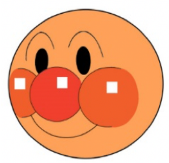
Introduction

- There are 3 major steps to finish an anime character illustration:
  1. Drawing a sketch
  2. Converting the sketch to a clean line–art
  3. Colorizing the line–art

Rough sketch Clean line-art Colored illustration

Introduction

- Inspired by the coloring practice table available for the online artist community, the colorizing step is more than filling with appropriate colors.
- There are 5 typical“coloring styles”.

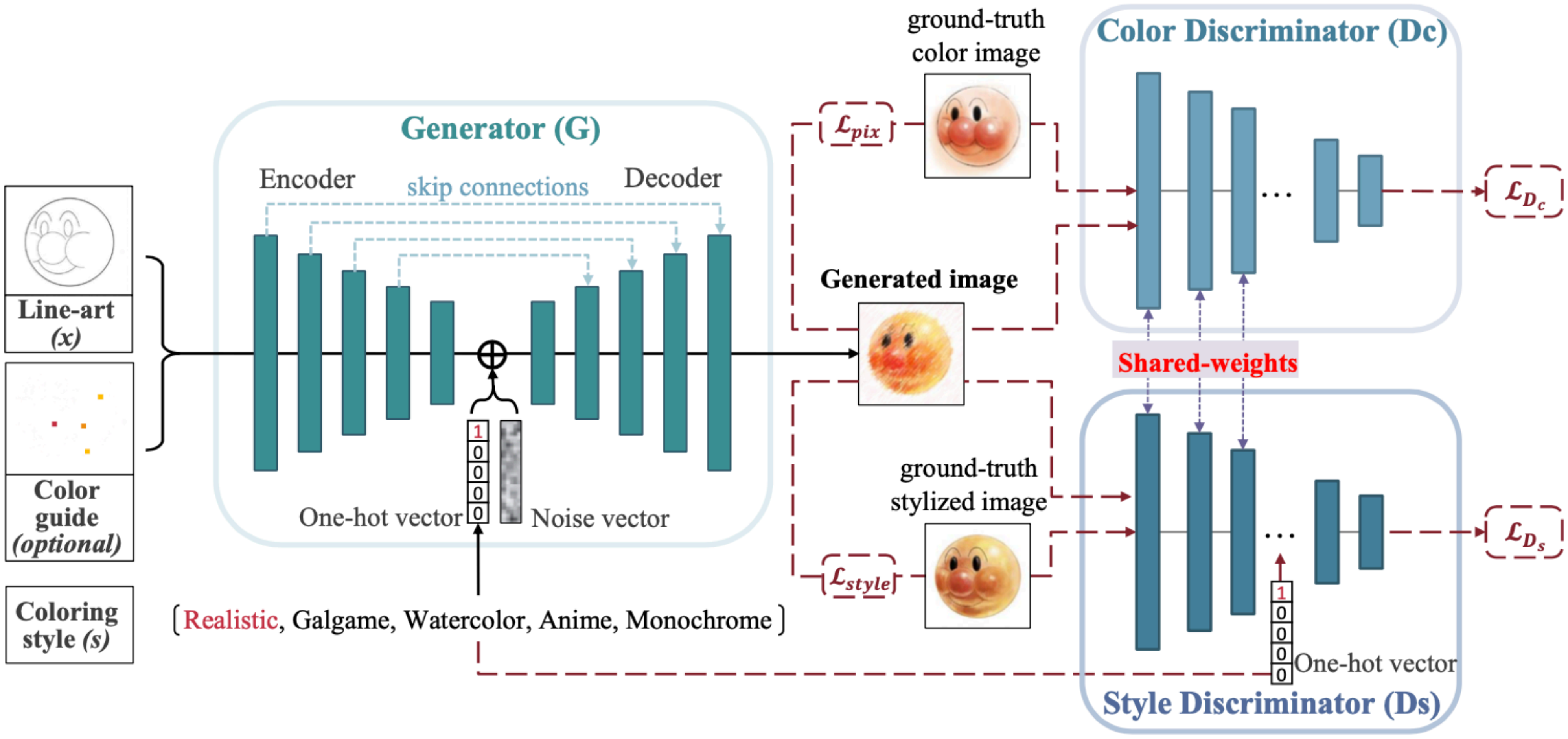
Line-art	1.Realistic	2.GalGame	3.Watercolor	4.Animate	5.Monochrome
					

- It can be observed that the styles give different impression to the same character.
- **We aim at colorizing line-arts in different coloring styles.**

Problem statement

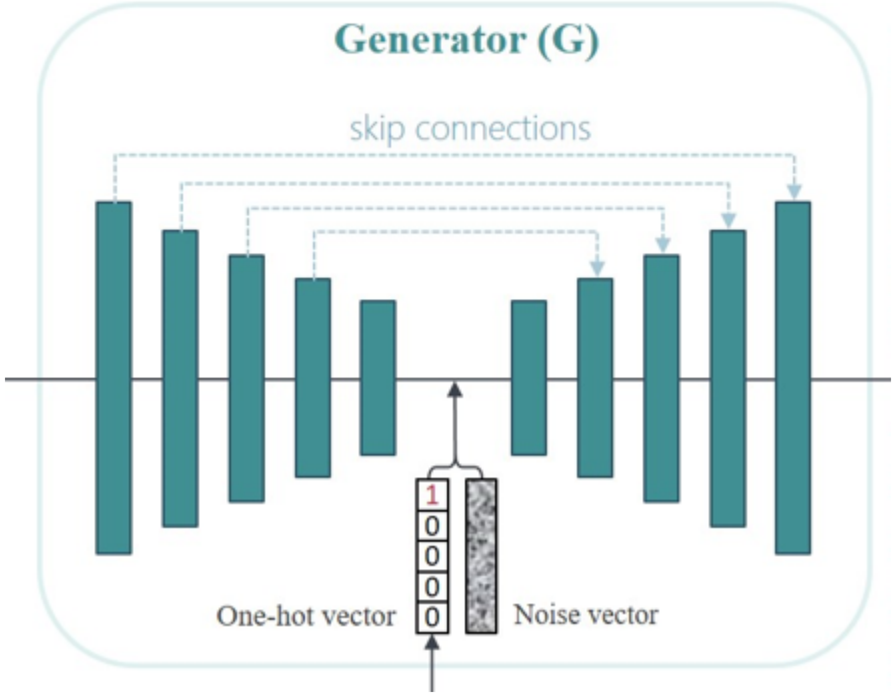
- Our task is challenging because it has 2 aspects: (1) colorization and (2) style transfer.
- Different from style transfer
  - Methods for style transfer usually bring distortion of object contours to reflect painting style. Our styles more take into account lightness, shading, and saturation of colors.
  - Existing style transfer methods can not handle multi–domain of styles.
  - We directly specify the styles, unlike style transfer where styles are given as images.
- We propose a GAN–based end–to–end model to address a novel stylized–colorization problem, which can be stated as a multi–domain image translation task.

Overview of proposed method



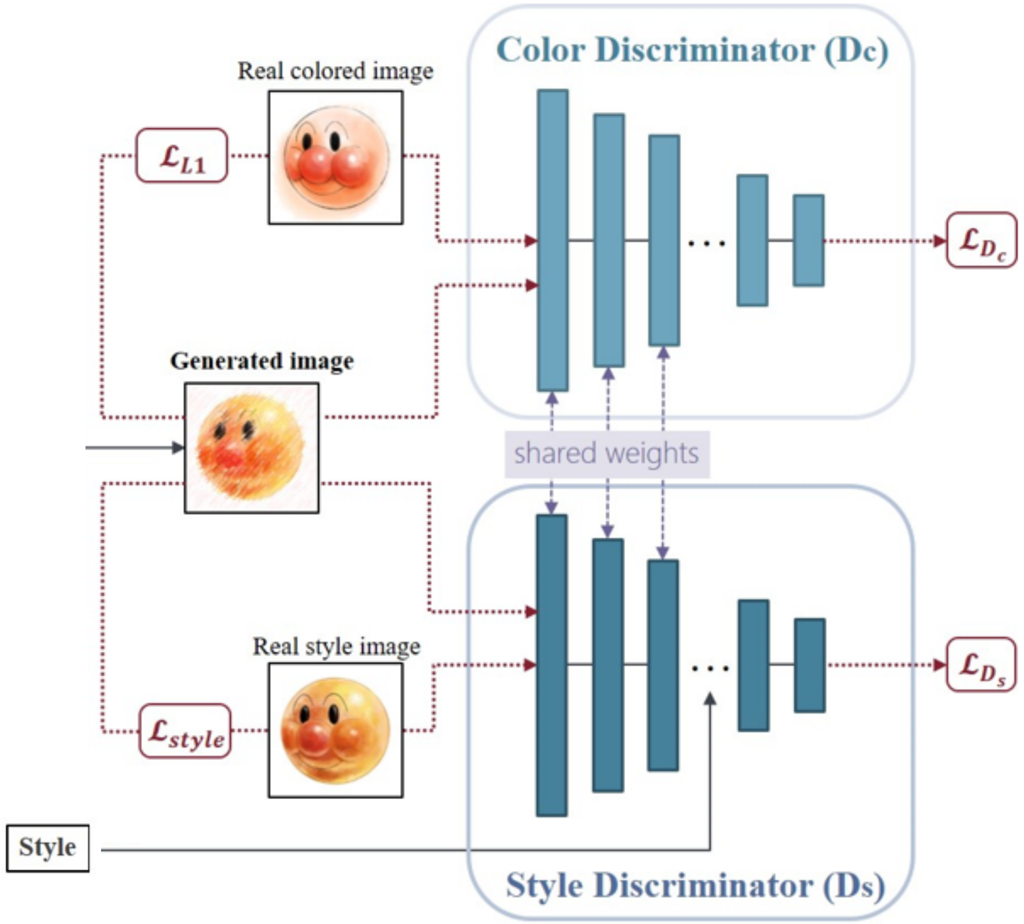
Network architecture

- Generator (G)
  - U–Net architecture.
  - Skip connections are added between mirrored layers of the encoder and decoder.



Network architecture

- **Color discriminator** (Dc)
  - Adopting PatchGAN
  - Dc receives either a ground–truth colored image or a generated image.
  - Distinguish if each NxN patch is real or fake.
- **Style discriminator** (Ds)
  - We condition the coloring style following the cGAN architecture.
  - Ds receives either a ground–truth stylized image or a generated image.



Network architecture

- We do not use any batch normalization layer in our network in order to keep the flexibility of colors.

- Stabilize the training and avoid collapsing our model
  - We add **spectral normalization** in both of Dc and Ds.
  - Adopt shared–weights at the first three layers between two discriminators.

Objective function

- Adversarial loss
  - The hinge loss: It helps the adversarial learning to be strong and stable.
  - Minimize the loss function for **Ds** and **Dc**:

$$\begin{aligned}\mathcal{L}_{D_c} =& \mathbb{E}_{y_c \sim p_{\text{data}}}\left[\max(0, 1 - D_c(y_c))\right] \\ &+ \mathbb{E}_{(x,s)\sim p_{\text{data}}}\left[\max(0, 1 + D_c(G(x,s))\right],\end{aligned} \tag{1}$$

$$\begin{aligned}\mathcal{L}_{D_s} =& \mathbb{E}_{(y_s,s) \sim p_{\text{data}}}\left[\max(0, 1 - D_s(y_s, s))\right] \\ &+ \mathbb{E}_{(x,s)\sim p_{\text{data}}}\left[\max(0, 1 + D_s(G(x,s), s)\right],\end{aligned} \tag{2}$$

x: input line-art
s: the coloring style
y_c: ground-truth colored image
y_s: ground-truth stylized image

  - Minimize the loss function for **G**:

$$\begin{aligned}\mathcal{L}_G =& -\mathbb{E}_{(x,s)\sim p_{\text{data}}}\left[D_c(G(x,s)\right] \\ &- \mathbb{E}_{(x,s)\sim p_{\text{data}}}\left[D_s(G(x,s), s))\right].\end{aligned} \tag{3}$$

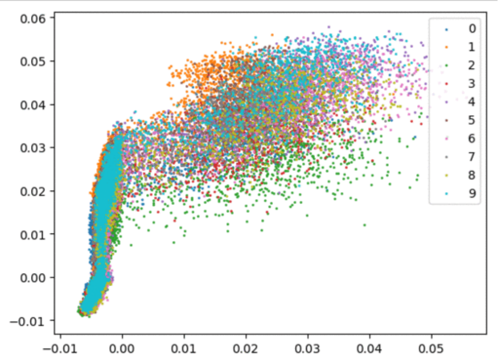
Objective function

- Per–pixel loss
  - L1 distance between the **generated image** and **ground truth colored illustration**.
  - Per–pixel loss enforces the generated image to be similar to the ground truth which is helpful in learning the color and keeping the image structure.

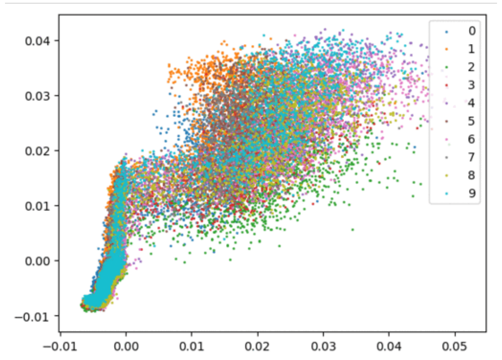
$$\mathcal{L}_{\text{pix}} = \mathbb{E}_{(y_c, x, s) \sim p_{\text{data}}} \left\| y_c - G(x, s) \right\|_1 \quad (4)$$

Objective function

- Style feature loss
  - We introduce the style feature loss to learn the styles of images.
  - First we train a style classifier **C** by employing the **center loss** [9].
    - Penalizing the distances between the style features and their corresponding center.
    - Reduce the intra-class differences.



Softmax only



Softmax + center loss

Objective function

- Style feature loss
  - After the training C, we obtained the feature representation c_s for coloring style s .
  - We employ the pre-trained classifier C to define our style feature loss:

$$\mathcal{L}_{\text{style}} = \mathbb{E}_{(x,s)\sim p_{\text{data}}}\left[\|c_s - C(G(x,s))\|_2^2 \right], \qquad (5)$$

x: input line-art

s: the coloring style

c_s: feature representation of style s

$C(G(\cdot))$: the style feature of generated image

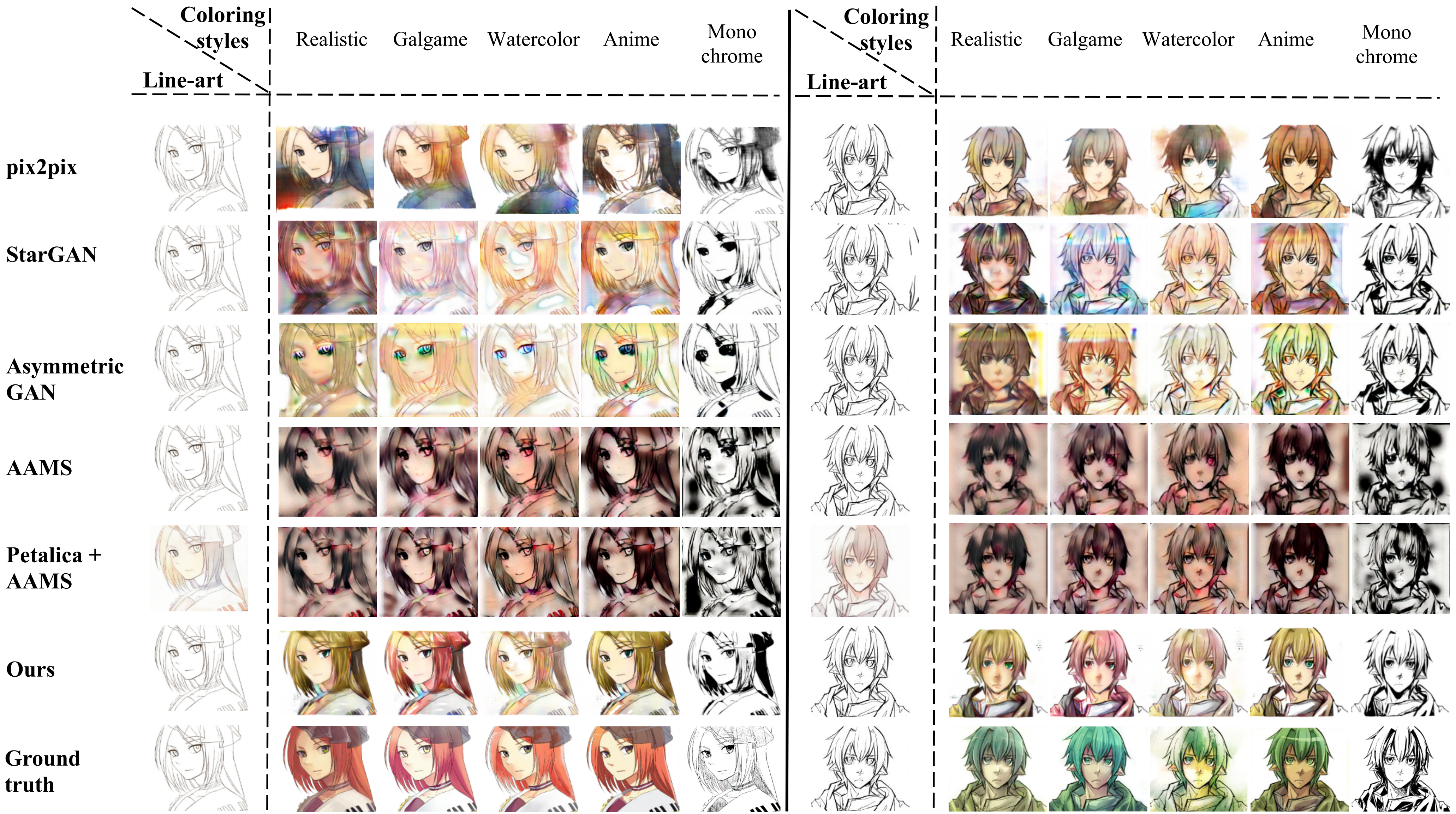
Objective function

- In summary, our full objective function is:

$$\mathcal{L} = \mathcal{L}_G + \lambda_{\text{pix}}\mathcal{L}_{\text{pix}} + \lambda_{\text{style}}\mathcal{L}_{\text{style}}, \qquad (6)$$

where λ_{pix} and λ_{style} are the hyper-parameters.

Visual comparison



Quantitative comparison

- We adopted three metrics:
  - PSNR (larger is better), SSIM (larger is better), and FID (smaller is better) for evaluation.

- R, G, W, A, M stand for realistic, galgame, watercolor, anime, monochrome, respectively.

Metric	PSNR					SSIM					FID			
Coloring style	R	G	W	A	M	R	G	W	A	M	R	W	A	M
pix2pix [10]	8.378	11.028	11.488	10.535	7.357	0.389	0.516	0.501	0.506	0.407	146.62	123.91	129.03	93.87
StarGAN [12]	4.978	5.062	6.288	5.112	3.649	0.145	0.158	0.222	0.184	0.165	165.83	161.03	176.79	106.89
AsymmetricGAN [13]	5.180	5.095	6.047	4.927	3.486	0.165	0.151	0.226	0.201	0.157	173.20	163.75	197.24	114.30
AAMS [23]	8.995	9.379	10.867	8.873	6.866	0.363	0.436	0.414	0.362	0.297	167.27	151.42	187.63	189.27
Petalica+AAMS	9.266	9.504	10.764	9.454	6.688	0.377	0.459	0.442	0.412	0.302	151.86	126.66	160.61	141.69
Ours	**10.753**	**11.511**	**12.957**	**11.165**	**7.799**	**0.455**	**0.573**	**0.563**	**0.548**	**0.445**	**118.88**	**117.38**	**125.49**	**88.34**

Conclusion

- We present a GAN-based end-to-end model to address a novel problem of stylized-colorization for line-arts.
- Our two discriminators work for judging in colorization and the coloring styles separately, and stylized-colorization results are produced by the generator.
- Our experiments demonstrate that our model successfully coloring line-arts with different styles, which is difficult to achieve with just combining existing colorization and style transfer methods.