StarGAN v2: Diverse Image Synthesis for Multiple Domains

Yunjey Choi*, Youngjung Uh*, Jaejun Yoo*, Jung-Woo Ha

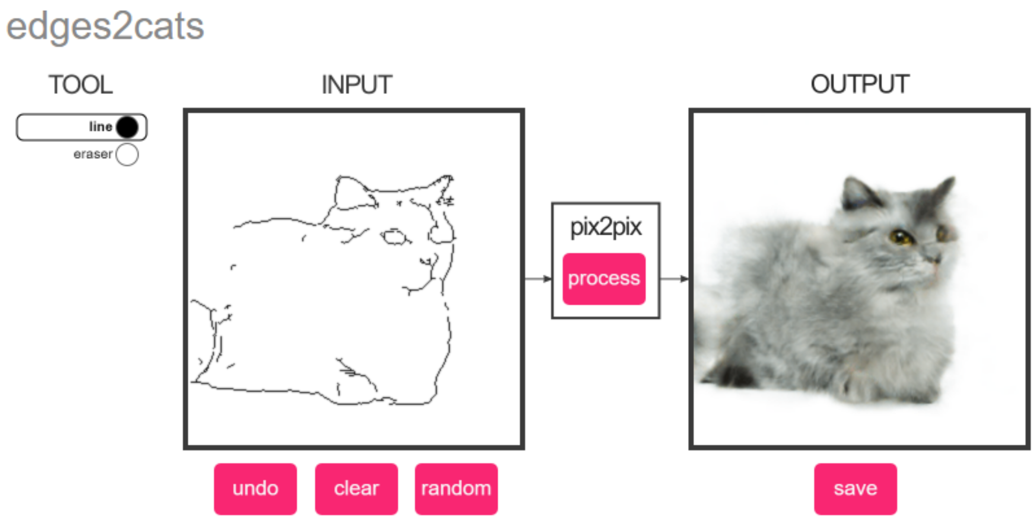
Clova AI Research, NAVER Corp, EPFL

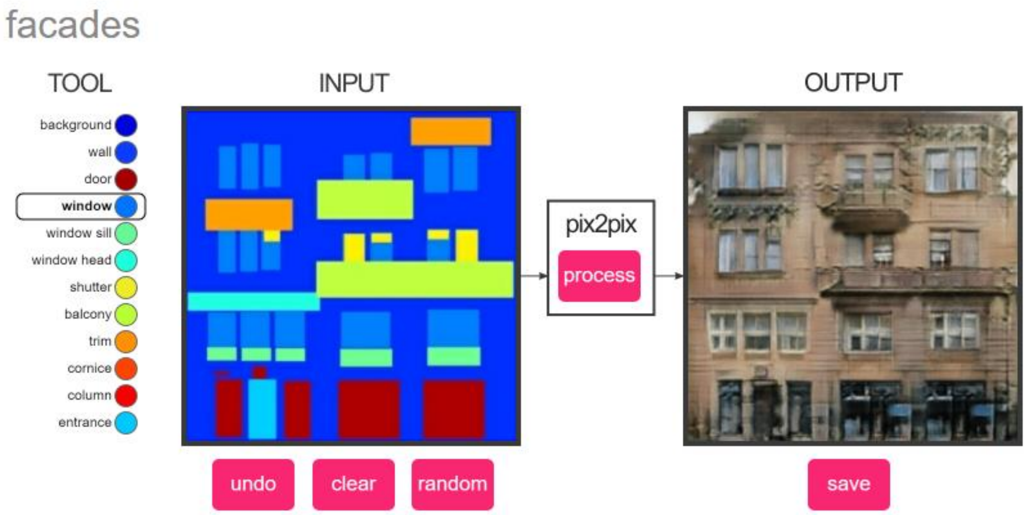
Gio Paik
giopaik@naver.com

✱ indicates equal contribution

Image to Image Translation

- Translate Input Image to another Input Image.
- A good Image to Image translation model has following properties:
  - Diversity of generated images
  - Scalability over multiple domains





Captured Images from https://affinelayer.com/pixsrv/

What does domain mean?

- Domain means a set of images that can be grouped as visually distinctive category.
- We call this style.
- Male/Female, Big/Small eyes, Long/Short hair etc…

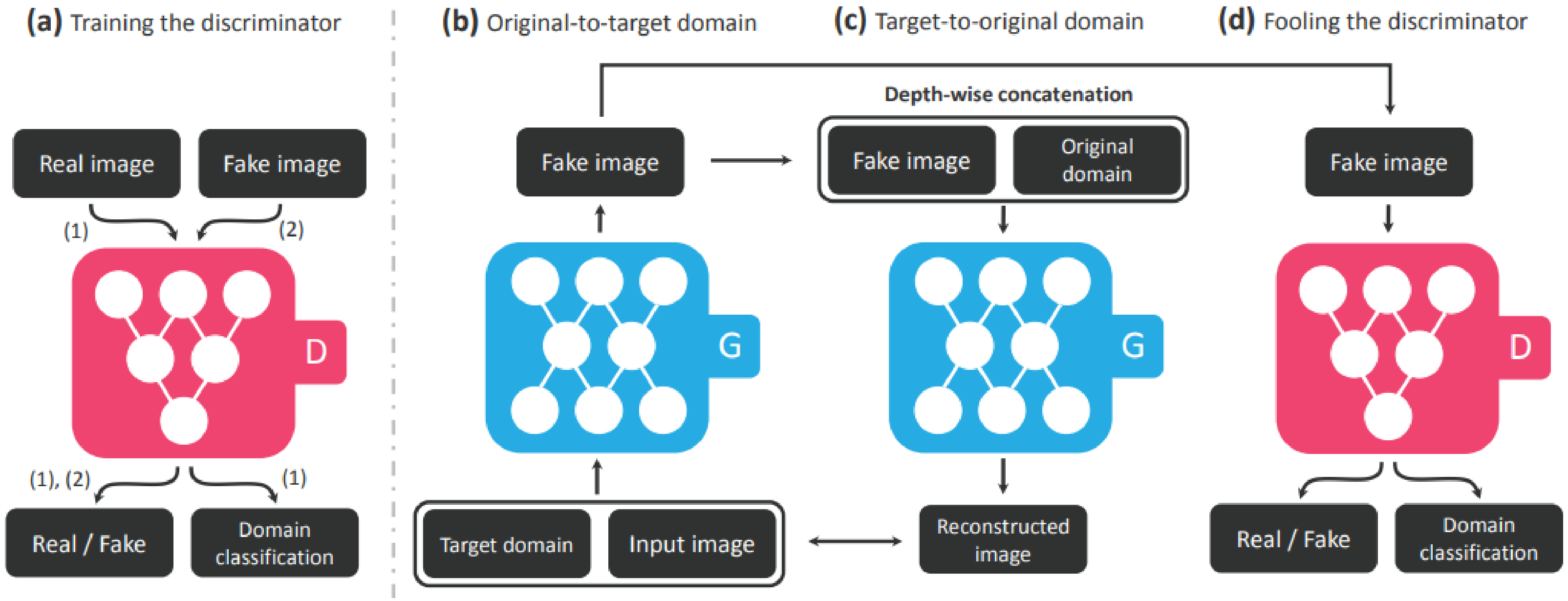
Limitation of previous methods

- Previous image to image translation used the following method:
  - Injecting Latent Vector to the generator.
  - This method is not scalable. Hence, It’s not practical.
  - If we have K domains, we need to train $K(K-1)$ generator models.
  - Each generator only consider a mapping between two domains.

Figure from Yunjey Choi et al, StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation, CVPR 2018

StarGAN

- StarGAN used single generator for all available domains.



- Still learns a deterministic mapping per each domain.

Yunjey Choi et al, StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation, CVPR 2018

StarGAN v2

- **StarGAN v2** is a single framework for Image Synthesis with
  - Diversity of generated images
  - Scalability over multiple domains
  - Superior visual quality.

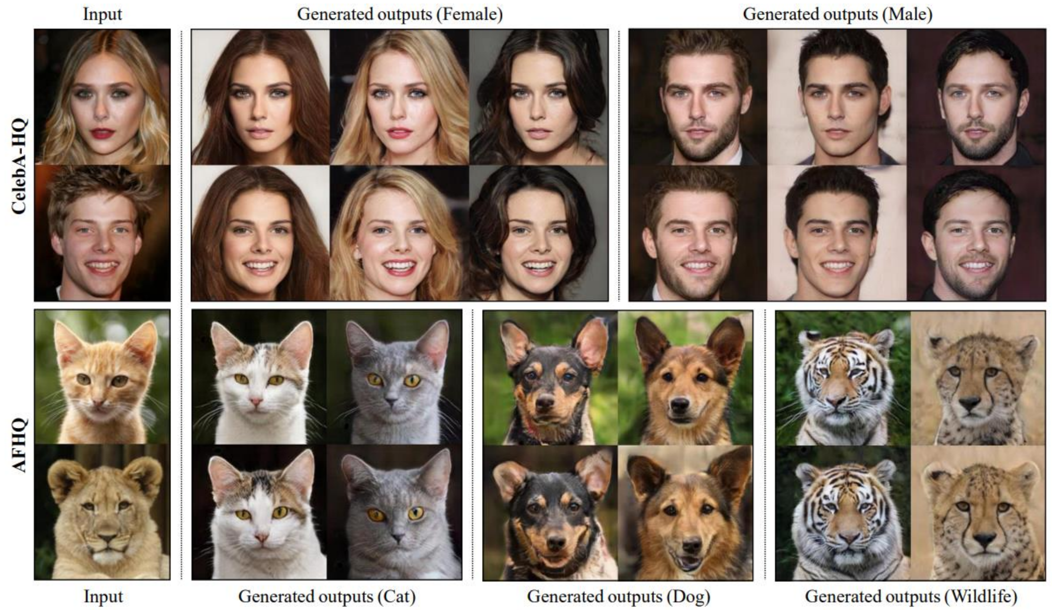
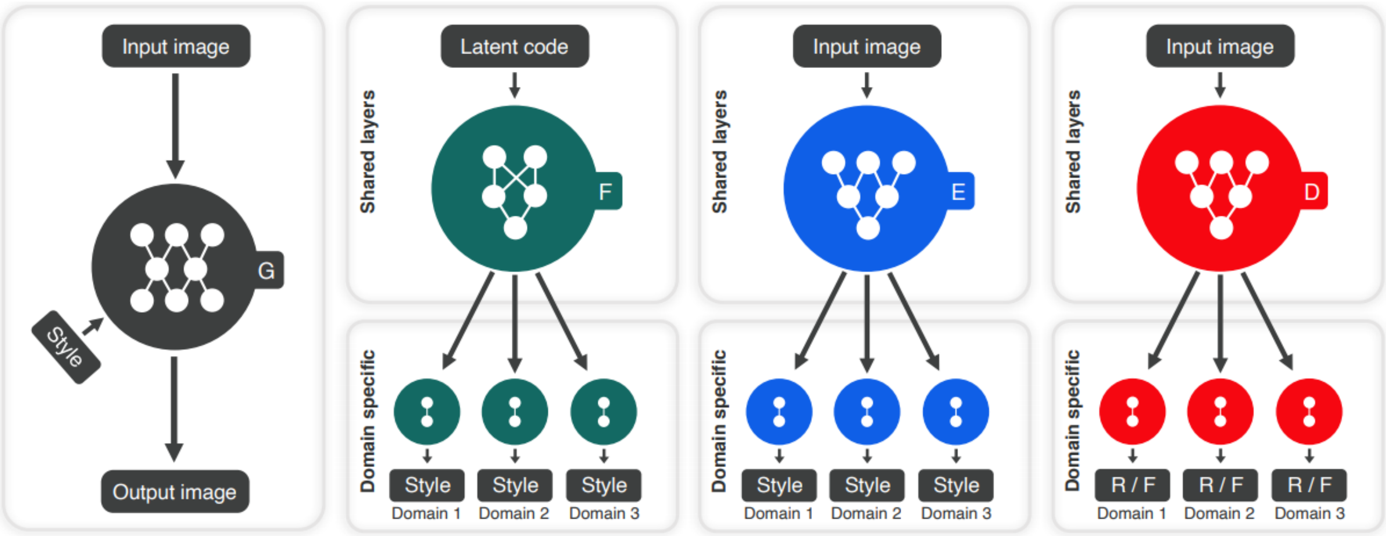


Figure 1. Diverse image synthesis results on the CelebA-HQ dataset and the newly collected animal faces (AFHQ) dataset. The first column shows input images while the remaining columns are images synthesized by StarGAN v2.

Figure from paper

StarGAN v2

- Let $\mathcal{X}$ and $\mathcal{Y}$ be the set of images and possible domains.
- Goal is to train a single generator G that can generate diverse images of each domain y that corresponds to the image x.

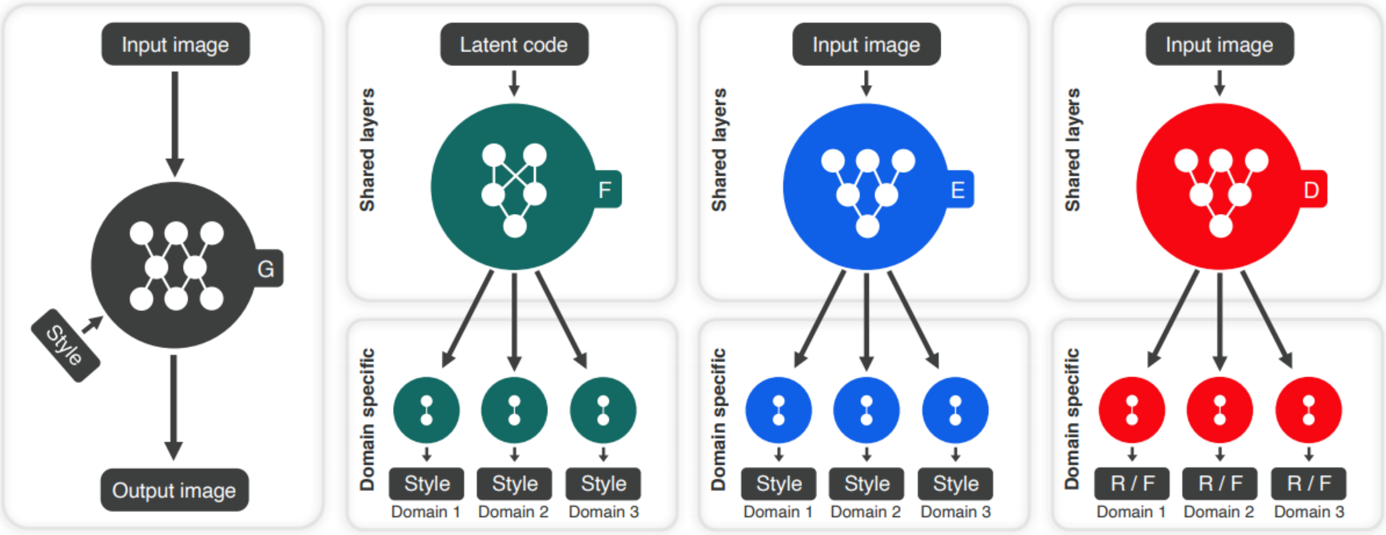


(a) Generator (b) Mapping network (c) Style encoder (d) Discriminator

Figure from paper

Generator G

- Generator G translates an input image x into an output image $G(x, s)$ reflecting a domain-specific style code s.
- We use adaptive instance normalization (AdaIN) to inject s into G.

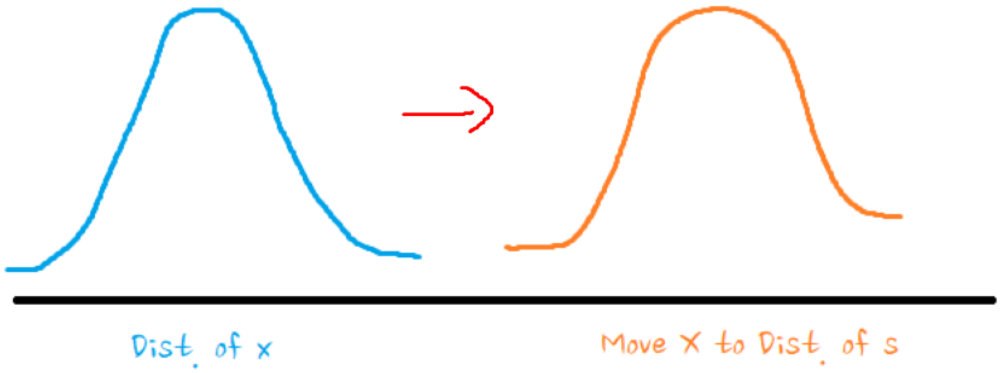


(a) Generator (b) Mapping network (c) Style encoder (d) Discriminator

Adaptive Instance Normalization

- Normalize input content x and inverse normalize in terms of s.

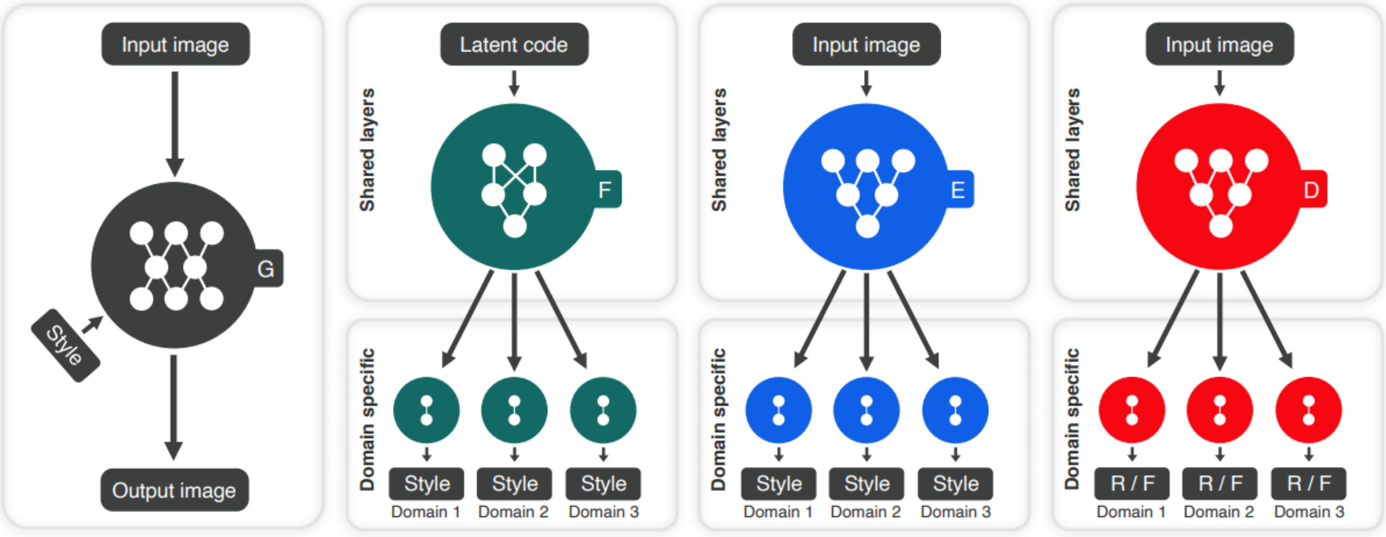
$$AdaIN(x, s) = \sigma(s)\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \mu(s)$$



X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. ICCV 2017

Mapping Network F

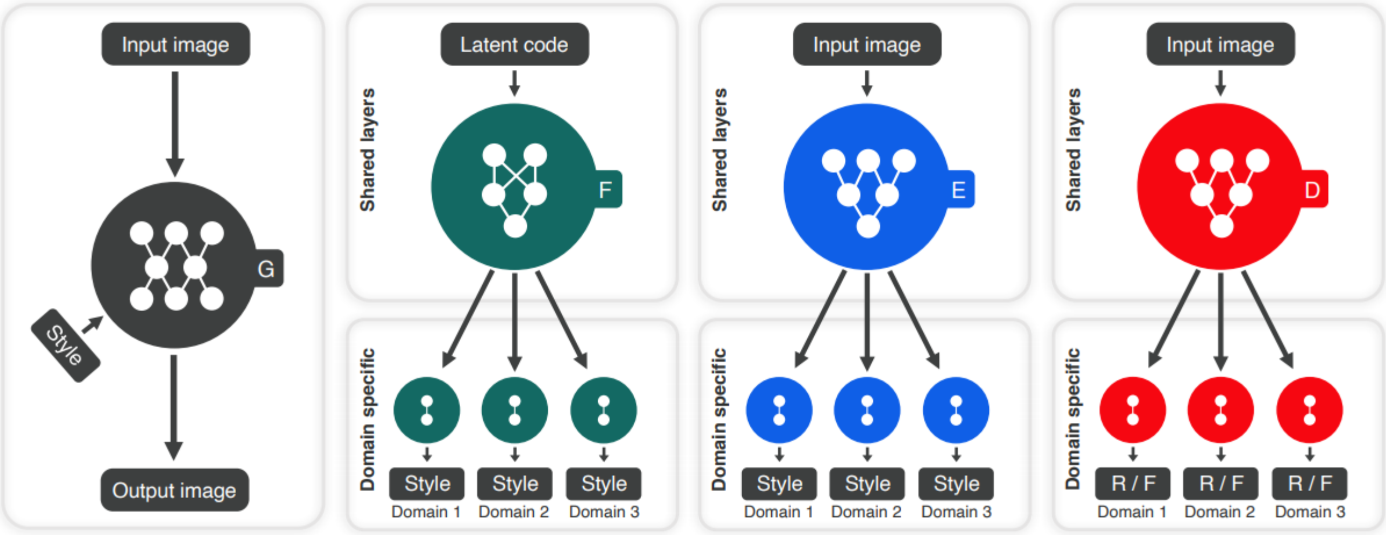
- Given a random latent code z, F generates a style code $s = F_y(z)$.
- F is a MLP with multiple output branches to provide style codes for all available domains.



(a) Generator (b) Mapping network (c) Style encoder (d) Discriminator

Style Encoder E

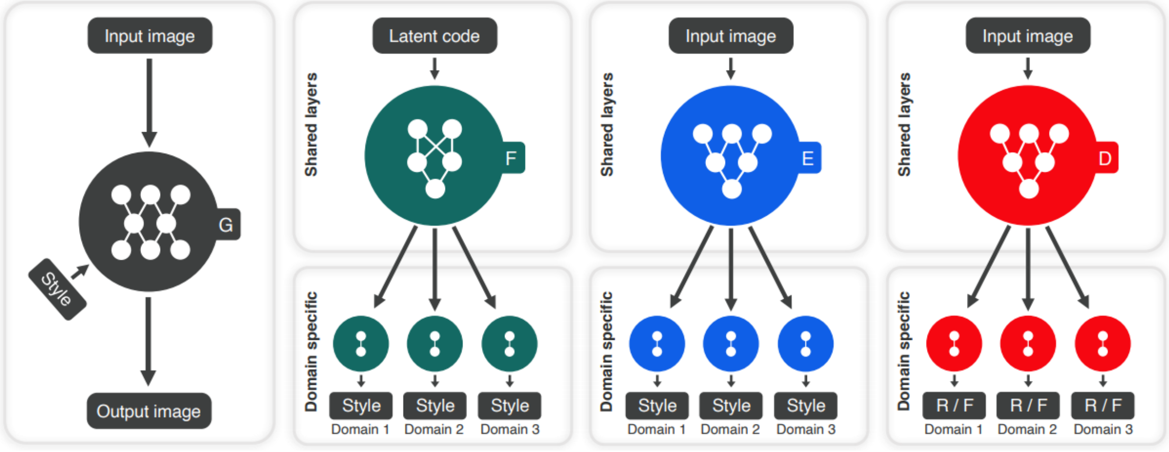
- Similar to F, E takes input image x and it's corresponding domain y and extracts the style code $s = E_y(x)$ of x.



(a) Generator (b) Mapping network (c) Style encoder (d) Discriminator

Discriminator D

- D is a multi task discriminator, which consists of multiple output branches.
- Each branch D_y learns a binary classification determining whether an image x is a real image of its domain y or a fake image $G(x, s)$produced by G.



(a) Generator (b) Mapping network (c) Style encoder (d) Discriminator

Training Objectives

- Given an image $x \in \mathcal{X}$ and its original domain $y \in \mathcal{Y}$, we train our framework using the following objectives.
  - Adversarial Objective
  - Style Reconstruction
  - Style Diversification
  - Preserving Source Characteristics
- Let's look into it!

Adversarial Objective

- By sampling $z \in \mathcal{Z}\ and\ y \in \mathcal{Y}$ randomly, we can generate a target style code $\tilde{s} = F_{\tilde{y}}(z)$.
- Generator G takes an image x and $\tilde{s}$ as inputs and learns to generate an output image $G(x, \tilde{s})$via an adversarial loss

$$\mathcal{L}_{adv} = \mathbb{E}_{x,y}\left[\log D_y(x)\right] + \mathbb{E}_{x,\tilde{y},z}[\log(1 - D_{\tilde{y}}(G(x, \tilde{s})))]$$

- F learns to provide the style code $\tilde{s}$ that is likely in the target domain $\tilde{y}$, and G learns to utilize $\tilde{s}$ and generate an image $G(x, \tilde{s})$ that is indistinguishable from real images of the domain $\tilde{y}$.

Style Reconstruction

- In order to enforce the generator G to utilize the style code $\tilde{s}$, we employ a style reconstruction loss

$$\mathcal{L}_{sty} = \mathbb{E}_{x,\tilde{y},z}\left[\left\|\tilde{s} - E_{\tilde{y}}\big(G(x,\tilde{s})\big)\right\|_1\right]$$

- We use the distance(norm) between $\tilde{s}$ and style code created by $\mathrm{E}_{\tilde{y}}$ using generated image $G(x,\tilde{s})$.

Style Diversification

- To further enable the generator G to produce diverse images, we explicitly regularize G with the diversity sensitive loss
$$\mathcal{L}_{ds} = \mathbb{E}_{x,\tilde{y},z_1,z_2}[\|G(x,\widetilde{s_1}) - G(x,\widetilde{s_2})\|_1]$$
- This loss forces G to explore the image space and discover meaningful style features to generate diverse images.
- Since this function's goal is make G to explore, we removed this term as training progressed.

Preserving Source Characteristics

- We need to insure generated image $G(x, \tilde{s})$ preserve the domain invariant characteristics of its input image x.
- We employ the cycle consistency loss

$$\mathcal{L}_{cyc} = \mathbb{E}_{x,y,\tilde{y},z}[||x - G(G(x, \tilde{s}), \hat{s})||_1]$$

- Where $\hat{s} = E_y(x)$, the estimated style code of the input image x, and y is the original domain of x.

Full Objective

- So, our full objective function can be summarized as follows:

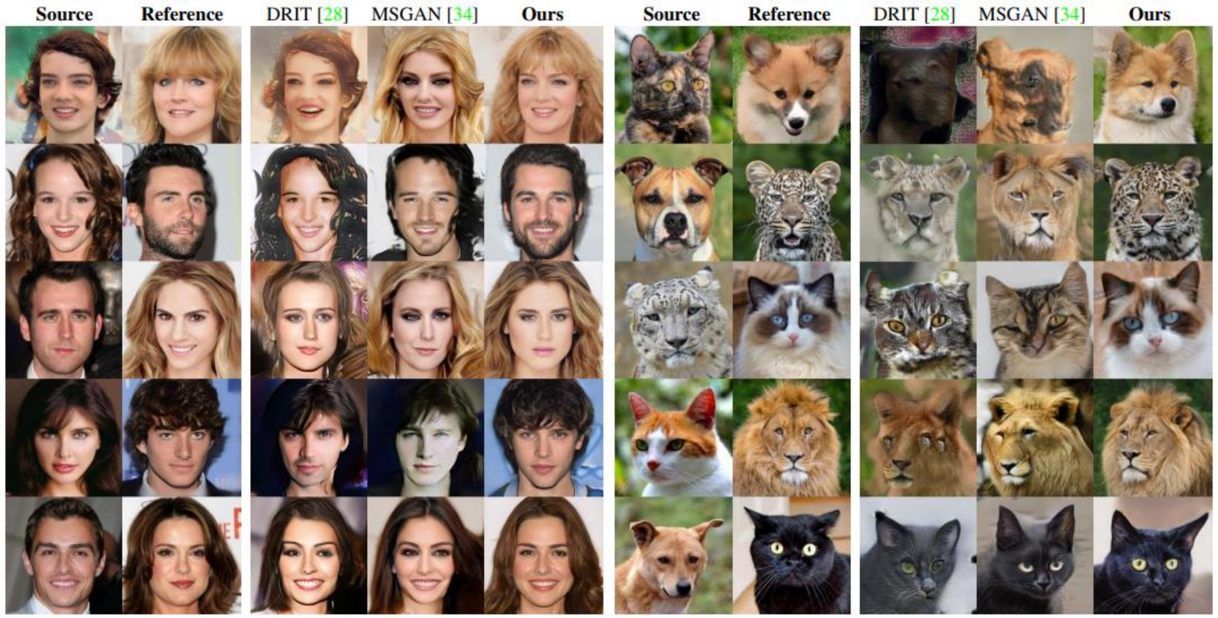
$$\mathcal{L}_d = -\mathcal{L}_{adv}$$

$$\mathcal{L}_{G,F,E} = \mathcal{L}_{adv} + \lambda_{sty}\mathcal{L}_{sty} - \lambda_{ds}\mathcal{L}_{ds} + \lambda_{cyc}\mathcal{L}_{cyc}$$

- Where $\lambda_{sty}, \lambda_{ds}\ and\ \lambda_{cyc}$ are hyperparams for each term.

Result

- StarGAN v2 shows superior results then previous methods like DRIT or MSGAN on Reference-guided synthesis task.



(a) Reference-guided synthesis on CelebA-HQ

(b) Reference-guided synthesis on AFHQ

Figure 6. Qualitative comparison of reference-guided image synthesis results on the CelebA-HQ and AFHQ datasets. Each method translates the source images into target domains, reflecting the styles of the reference images.

Method	CelebA-HQ		AFHQ	
	Quality	Style	Quality	Style
MUNIT [16]	6.2	7.4	1.6	0.2
DRIT [28]	11.4	7.6	4.1	2.8
MSGAN [34]	13.5	10.1	6.2	4.9
StarGAN v2	**68.9**	**74.8**	**88.1**	**92.1**

Table 4. Votes from AMT workers for the most preferred method regarding visual quality and style reflection (%). StarGAN v2 outperforms the baselines with remarkable margins in all aspects.

Figure from paper
H.-Y. Lee et al. Diverse image-to-image translation via disentangled representations. ECCV 2018
Q. Mao et al. Mode seeking generative adversarial networks for diverse image synthesis. CVPR 2019

Result

- StarGAN v2 addresses two major challenges in image-to-image translation:
  - An image of one domain to diverse images of a target domain.
  - Supporting multiple target domains.
- Model can generate images with rich styles across multiple domains.

Figure from official github repo

StarGAN v2:
Diverse Image Synthesis for Multiple Domains

Paper: https://arxiv.org/abs/1912.01865

Official PyTorch Implementation: https://github.com/clovaai/stargan-v2

Thank you for watching!

Gio Paik
giopaik@naver.com