Unsupervised Photo-to-Caricature Generation with Adaptive Select Layer-Instance Normalization and Semi-cycle Consistency

Author Zhiwei,Li 1011908207@qq.com
Nanjing Normal University

Author Weiling,Cai 24577075@qq.com
Nanjing Normal University

Author Cairun,Wang 1252835047@qq.com
Nanjing Normal University

Editors: Emtiyaz Khan and Mehmet Gönen

Abstract

Unpaired photo to caricature generation is a challenging but meaningful task. Generating high quality caricatures with rich texture/color and plausible exaggeration is important. Previous methods often respectively deal with the shape transformation and texture/color style. We argue that shape transformation can be treated as same as texture/color. Thereby, shape transformation and texture/color can be transferred at the same time. In this paper, we proposed a new method namely AdsSe-GAN for photo-to-caricature generation, which consists of a new normalization function called AdaSLIN and a new semi-cycle consistency loss. The AdaSLIN adaptively selects Layer Normalization or Instance Normalization to simultaneously transfer texture/color and shape transformation. Besides we present semi-cycle consistency loss which only imposes L1 norm on caricature-to-photo process, which is different from existing methods that apply cycle consistency loss to preserve the original domain information. In fact, while generating caricature, taking no account of the cycle restriction makes our model generate caricature with more distinct exaggeration and higher quality. Experimental results on a public caricature dataset, WebCaricature, show the effectiveness of our proposed method compared with the state-of-the-art models.

Keywords: Caricature generation, Generative Adversarial Nets, Style transfer.

1. Introduction

Caricatures are artistic drawings with specifically exaggerated features for a political or entertainment purpose. Different from cartoon, caricatures can have more artistic drawings, like sketching, pencil strokes and so on. Drawing a caricature needs specific skills and consumes huge amount of time of caricature master. Thus, automatic caricature generation has important research meaning.

There are several early methods Akleman (1997), Akleman et al. (2000) on transferring a photo-face into a caricature, but these methods rely on interaction of the user to generate exaggeration. And there are other methods Brennan (2007), Mo et al. (2004) which predefine the exaggeration rules to achieve automatic caricature generation.

There have been many explorations on style and shape. Gatys et al. (2016) first separate the content and the style of an image and change the style of an image without changing the content. Huang and Belongie (2017) use AdaIN which aligns the mean and variance of the content features to those of style features to transfer style. Adaptive Layer-Instance Normalization (AdaLIN) is proposed in Kim et al. (2020) which combines the Layer normalization and Instance normalization to handle the large shape changes between domains. It is observed that Layer Normalization can transfer shape. Most recently, Li et al. (2021) offer a new insight that local shapes can be treated as a kind of style like color/texture. Specifically, they use 1×1 convolution to combine the features processed by Layer normalization and Instance normalization. By rethinking these methods, we explore the impact of normalization on representation of the exaggeration. Recent caricature generation methods Cao et al. (2018), Shi et al. (2019), Gong et al. (2020) mostly decouple the style into texture/color and shape transformation to generate exaggeration. Here we hold different understanding that exaggeration can be seen as a kind of style just like texture/color. Thus, we can simultaneously transfer texture/color and exaggeration. It is explored in Li et al. (2021) that using specific method to combine LN and IN can achieve impressive results. However, their method is designed for anime generation and not suitable for caricature generation. Using their method causes the generated caricatures losing identity information corresponding to input photo.

Recently deep learning methods are widely used in image-to-image translation and achieve impressive results. Pix2Pix Isola et al. (2017) makes use of paired data to solve this problem. CycleGAN Zhu et al. (2017) utilizes a cycle structure making it possible to train the model in an unsupervised way. StarGAN Choi et al. (2018) uses only a single model to learn mapping among multiple domains. StarGAN-v2 Choi et al. (2020) extends StarGAN and generates diverse images across multiple domains. StyleGAN Karras et al. (2019) uses Adaptive Instance Normalization (AdaIN) Huang and Belongie (2017) to insert style code in the middle of the network, which achieves impressive results. There are piles of works based on StyleGAN. StyleCariGAN Jang et al. (2021) generates high quality caricatures via StyleGAN Feature Map Modualtion. DualStyleGAN Yang et al. (2022) delicately designs extrinsic style path to simultaneously handle color/texture and shape transfer. Since collecting proper paired data for photo and caricature can be very difficult and consuming, so it is better to train the model in an unsupervised way.

In this paper, we propose a new unsupervised photo-to-caricature method namely AdsSeGAN to explore more reasonable exaggerations and maintain the identity information at the same time. To adopt the model to the caricature generation situation, we introduce a new normalization method called AdaSLIN (Adaptive select Layer-Instance normalization), at the heart of the method is an attention mechanism that adaptively selects Layer normalization or Instance normalization. Besides, we argue that photo domain and caricature domain have great information gap. Compared to photo, Caricatures are more structured and have less information. We find that transferring a photo into a caricature without applying L1 norm won't lose the identity information and without the constraint of cycle consistency, we can generate caricatures with more various exaggerations and higher image quality. However, the identity information is lost when transferring a caricature into a photo. Thus, we present a semi-cycle consistency loss to handle this problem. In summary, the main contributions of our paper are as follows:

- We propose a novel normalization function, AdaSLIN, which adaptively selects Layer normalization or Instance normalization to generate caricatures with fine texture/color and plausible exaggerations.
- We propose a new semi-cycle consistency loss applied to our model to explore more possible exaggerations and generate caricatures with higher quality without hurting the identity information.
- We evaluate our proposed model on WebCaricature dataset. Experimental results show that AdsSe-GAN can not only transfer color/texture and reasonable exaggeration but also generates high quality caricatures with maintaining the identity information.

2. Related Work

2.1. Neural Style Transfer

Style transfer is the task of changing the style of an image in one domain to the style of an image in another domain. Gatys et al. (2016) firstly transfer the style of an image to the content of another image by matching feature statistics. Many works Huang and Belongie (2017), Li and Wand (2016) have been proposed to improve the quality and the speed. Huang and Belongie (2017) propose AdaIN achieving real time arbitrary style transfer. Specifically, they align the channel-wise mean and variance of a content image to those of the style image. AdaLIN Kim et al. (2020) combines the instance normalization and the layer normalization achieving impressive results in simultaneously transferring texture and shape. But AdaLIN combines instance normalization and layer normalization in a per-channel manner causing insufficiency of simultaneously transferring color/texture information and shape information. AdaPoLIN Li et al. (2021) uses convolution to combine the IN and LN to achieve all-channel combination.

2.2. Image-to-Image Translation Networks

With the advance of GANs, many GAN based Image-to-image methods have been proposed recently. Isola et al. (2017) propose Pix2Pix which learns a mapping function from input image to output image by using a cGAN framework with paired data. Wang et al. (2018) propose feature matching loss for high-resolution image-to-image translation improving Pix2Pix. CycleGAN Isola et al. (2017) uses a cycle consistency loss to get rid of the dependence of paired data. UNIT Zhu et al. (2017) uses an unsupervised image-to-image translation framework with the shared-latent space assumption to learn a joint distribution of images in different domains. MUNIT Huang et al. (2018) uses AdaIN Huang and Belongie (2017) to combine the content of an image with the style of another image extending UNIT Zhu et al. (2017) to multimodal image-to-image translation. StarGAN Choi et al. (2018) uses a single model achieving multi-domain Image-to-image translation and StarGANv2 Choi et al. (2018) extends StarGAN by using latent code injection Choi et al. (2020) and other methods achieving high quality and diverse style results. Applying image-to-image framework to caricature generation can be difficult because general image-to-image models often only succeed in transferring color/texture. But for unpaired photo-to-caricature translation, generating reasonable shape exaggerations and maintaining identity information at the same time are challenging but important. Sometimes, preserving the

identity information causes inferior image quality. In this paper, we propose a new normalization method to generates plausible shape exaggeration and a semi-cycle consistency to generate high quality caricature without losing the identity information.

2.3. Deep Caricature Generation

With the popularization of Deep Neural Network and GANs, there have been many studies on automatic caricature generation algorithms. CariGANs Cao et al. (2018) uses two different networks to decompose the caricature generation into geometric exaggeration and appearance stylization. They apply bidirectional cycle consistency and cosine similarity on landmark to preserve visual features. CariGAN Li et al. (2020) uses weakly paired data to enforce the output to have reasonable exaggeration and facial deformation. WarpGAN Shi et al. (2019) also decomposes the caricature generation into texture transfer and geometric exaggeration, specifically, WarpGAN Shi et al. (2019) automatically predicts a set of control points to generate geometric exaggeration. AutoToon Gong et al. (2020) also respectively handles the style transfer and shape exaggeration, and uses dense deformation fields to generates geometric exaggeration.

Previous methods often decompose the caricature generation into texture transfer and geometric exaggeration, which makes caricature generation easier, but the generated exaggerations of caricature are often less reasonable and the output caricature tends to have low image quality. It is obvious that our method is different from the previous. We do not decompose the problem into text/color transfer and shape transformation to simplify the problem. In this paper, we design specific model which uses AdaSLIN to adopt this problem generating caricatures with detailed exaggeration and high quality.

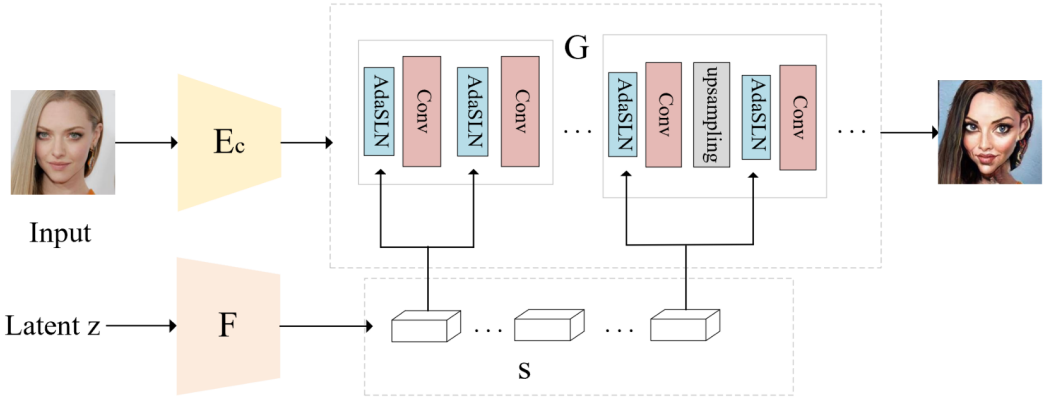
3. Proposed Method

3.1. Framework

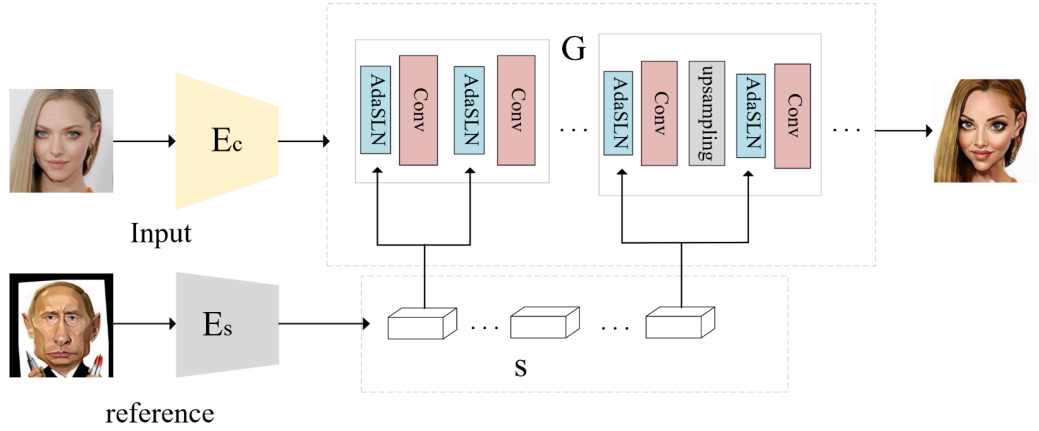
Let X be the real photo domain and Y be the caricature domain. Given a real photo $x \in X$, our goal is to generate caricatures with rich texture/color and plausible exaggerationsat the same time, maintaining the same identity information of input x. It is shown in Choi et al. (2018), Choi et al. (2020) that using only a single Generator can perform image-to-image translations for multiple domains. So, unlike most image-to-image methods Zhu et al. (2017), which design their model in a dual way with two generators, we design our model in a unified way with only one generator. Our generator can take photo as input and caricature as reference and also can take caricature as input and photo as reference vice versa.

We design our model to generate caricature in two different ways, which reflect in two ways to gain style vector s. One is from Gaussian distribution, and the other is from reference image. Thereby, there are two different network architectures for generating style vector s. Our overall architecture illustrated in Fig. 1 consists of five modules, which are mapping network F, style encoder E_s, content encoder E_c generator G and discriminator D.

The mapping network F consists of an MLP with two branches to generate style code for photo domain and caricature domain. Given a random Gaussian vector z, the mapping



(a) Latent-guide



(b) Reference-guide

Figure 1: Overall structure of our model. Our model can generate caricature in two situations, namely latent-guide and reference-guide. For latent-guide, we design F to transform z sampled from Gaussian distribution into style code s. For reference-guide, we design E_s to encode a reference caricature into style code s. The content encoder E_c takes an image as input, then transforms it into piles of feature maps, and then our generator by taking advantage of AdaSLIN inserts style code into different levels of feature maps, and reconstructs them to caricature.

network module F produces a style code $s = F(z)$ corresponding to photo domain X or caricature domain Y. The domain information is hidden in the style code. Our double-branch architecture allows F to learn better style representations.

The style encoder E_s learns to extract style code $s = E_s(y)$ from a given style image y. The style encoder E_s is also designed in a double-branch architecture. Thus, the domain information is also hidden in the style code s.

We adopt an auto-encoder structure to design our model. The content encoder E_c is used to extract the content of the input image into a pile of feature maps, and the generator G is to reconstruct image and insert style. To combine with the advantages of IN and LN, we apply our AdaSLIN to insert style. Our AdaSLIN enables the model to simultaneously transfer exaggeration and color/texture style.

3.2. AdaSLIN

Recently, Huang and Belongie (2017)propose AdaIN to transfer style, which aligns the mean and variance of the style image to the mean and variance of the content image. But, in our situation, not only the style but also the exaggeration needs to be transferred. It is shown in Kim et al. (2020) that Layer Normalization (LN) can change the shape of an image. Our goal is to generate reasonable exaggeration of a caricature, which can be influenced by LN. Specifically, AdaLIN introduces a parameter $\rho \in [0, 1]$ to control the degree of LN and IN. But we find that as the training goes on, AdaLIN fails to generate fine caricature with respect to the style such as color distribution. It is analyzed in Li et al. (2021) that AdaLIN ignores the correlations among channels, which makes the transfer process insufficient. PoLIN and AdaPoLIN is proposed in Li et al. (2021) to achieve all-channel combination of IN and LN. Specifically, PoLIN used a 1×1 convolutional layer to combine the IN and LN. However, different from anime generation, generating a caricature must maintain the identity of the original image, combining the IN and LN in an all-channel way will destroy the identity of the original image, or even worse, makes the reconstruction of the image unsuccessful.

To address the issues above, AdaSLIN introduces an attention mechanism Li et al. (2019) to balance the influence of IN and LN. Our goal is to adaptively select the normalization method. The structure of AdaSLIN is illustrated in Fig. 2.

For any given feature map $X \in \mathbb{R}^{C\times H\times W}$, we conduct IN and LN respectively, thereby we get the result A_{IN} after conducting IN and A_{LN} after conducting LN.

Firstly, we fuse A_{IN} and A_{LN} via an element-wise summation:

$$N = A_{IN} + A_{LN} \tag{1}$$

Then, we adopt global average pooling to produce channel-wise parameter $s \in \mathbb{R}^C$. The calculation of c-th element of s is:

$$s_c = F_{gp}(N_c) = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W} N_c(i, j) \tag{2}$$

Further, we use a fully connected layer to get a compact feature $z \in \mathbb{R}^{d\times 1}$. z is designed to supply guidance for the precise and adaptive selection of IN or LN:

$$z = F_{fc}(s) = Ws \tag{3}$$

where $W \in \mathbb{R}^{d\times c}$ is the parameter matrix.

$$d = \max(C/r, L) \tag{4}$$

where L denotes the minimal value of d ($L = 32$ as default setting). We follow the setting for d in Li et al. (2019).

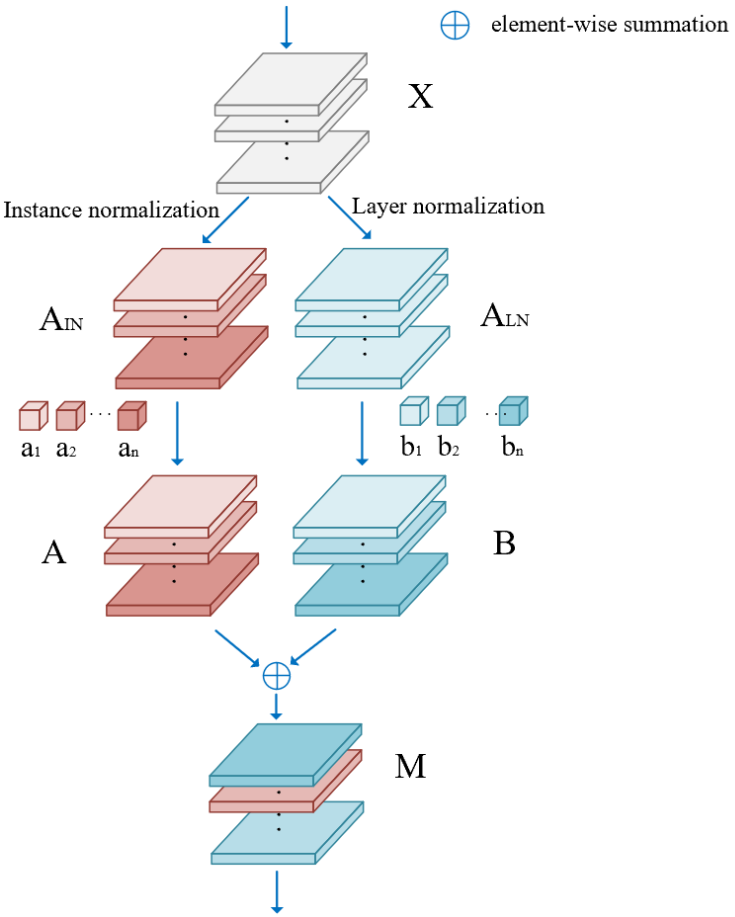


Figure 2: The structure of AdaSLIN. For any given feature map X, we conduct two normalization methods, then through our attention mechanism, we gain $a_1, \ldots, a_n$ for A_{IN} and $b_1, \ldots, b_n$ for A_{LN}. Finally, we combine A_{IN} and A_{LN} through element-wise summation to get M.

We use a soft attention across channels to adaptively select normalization method, which is guided by the compact feature descriptor z. Specifically, a softmax operator is conducted on the channel-wise digits:

$$a_c = \frac{e^{A_c \times z}}{e^{A_c \times z} + e^{B_c \times z}}, b_c = \frac{e^{B_c \times z}}{e^{A_c \times z} + e^{B_c \times z}} \tag{5}$$

where $A, B \in \mathbb{R}^{C \times d}$and denote the soft attention vector for A_{IN} and A_{LN}. The final feature map M is gained by the attention weights on IN and LN:

$$M_c = a_c \cdot A + b_c \cdot B, \quad a_c + b_c = 1 \tag{6}$$

Where $M = [M_1, M_2, \ldots, M_c]$ $M_c \in \mathbb{R}^{H \times W}$.
Finally, we embed the style codes into M to transfer style:

$$AdaSLIN(M, \gamma, \beta) = \gamma \cdot M + \beta \tag{7}$$

By the proposed AdaSLIN, which adaptively selects LN or IN through an attention mechanism, we can simultaneously transfer exaggeration and color/texture.

3.3. Semi-cycle Consistency Loss

The photo domain and the caricature domain are two asymmetric domains, which means there are information gap between them. Compared to caricature, photo carries more information, like more detailed texture and more intricate color. Adopting traditional cycle consistency loss does no good in such case. Actually, we find that applying L1 cycle consistency will degrade the quality of generated image. We reckon that generating caricature with exaggerations changes the structure of the source image to a certain degree. Applying cycle-consistency may make the generator G confused whether preserve the information of source image or change it. Based on experiments, we come to a conclusion that translating an image carries more information to an image carries less information is not likely to lose essential identity information. Although in our case, translating a caricature into a photo is not that important, we adjust the cycle consistency loss to a semi-consistency loss to make sure that translating a caricature into a photo does not lose the identity information. The proposed semi-cycle consistency loss is defined as:

$$L_{semi-cyc} = \mathbb{E}_{y,z}\left[\|y - G(G(y,s),\hat{s})\|_1\right] \tag{8}$$

Where y is an image from caricature domain, s is the style code corresponding to photo domain using latent guide or reference guide, and $\hat{s} = E_s(y)$ is the style code of the input.

3.4. Loss Function

Our model takes either a photo or a caricature as input. For simplicity, here we only formulate the situation of photo x as input, caricature y as reference.

Adversarial Loss. Given a reference y, then E_s generates a style code $s = E_s(y)$. Given a random Gaussian vector z, then F generates a style code $s = F(z)$, and G takes x, s as input, the adversarial loss is as follows:

$$L_{adv} = \mathbb{E}_x\left[log(D_X(x))\right] + \mathbb{E}_{x,y,z}\left[log(1 - D_Y(G(x,s)))\right] \tag{9}$$

Where D_X denotes the output of discriminator corresponding to photo domain, D_Y denotes the output of discriminator corresponding to caricature domain.

Style Reconstruction. In order to guarantee our Encoder E_s to generate style code that reflects the reference image and to enforce the generator G to utilize the style code s, we apply a style reconstruction loss:

$$L_{rec}^{sty} = \mathbb{E}_{x,y,z}\left[\|s - E_s(G(x,s))\|_1\right] \tag{10}$$

Where $s = F(z)$ or $s = E_s(y)$, if we take s as input then our generated image should have the same style code with s. Applying this term makes our style encoder learn to transform an image into style code.

Style Diversification. To encourage the generator G to generate different style image given different latent code z, we apply style diversity loss:

$$L_{ds}^{sty} = \mathbb{E}_{x,z_1,z_2}\left[\|G(x,s_1) - G(x,s_2)\|_1\right] \tag{11}$$

Where s_1, s_2 are generated by F given different random latent code z_1, z_2. By maximizing this term, we force our model to generate different results given the same input image but different latent code, thus to generate caricature with more diversity.

Full Objective. The full objective is as follows:

$$\max_{G,F,E} \min_{D} \quad L_{adv} + \lambda_{rec} L_{rec}^{sty} - \lambda_{ds} L_{ds}^{sty} + \lambda_{semi-cyc} L_{semi-cyc} \tag{12}$$

4. Experiment

4.1. Dataset

Our proposed model is trained on a large face-caricature dataset WebCaricature Huo et al. (2018). The dataset consists of 5,974 photos and 6,042 caricatures. All the images are aligned according to the facial landmarks provided in the dataset and resize to 256×256 resolution. Then we simply divide the dataset into training set and test set by a ratio of nine to one.

4.2. Training Details

We use Adam Kingma and Ba (2015) optimizer in Pytorch with $\beta_1 = 0$ and $\beta_2 = 0.99$ in the whole training process. We train the network for 50k iterations. The batch size is set to 8, each mini-batch is composed of random photo and caricature. The learning rate is set to 10^{-4} for G, D, E and 10^{-6} for F. We set $\lambda_{rec}, \lambda_{ds} = 1$, and $\lambda_{semi-cyc} = 0.1$. We conduct all experiments using Pytorch 1.9.0 with 4 GeForce RTX 3080 GPUs.

4.3. Comparison to State-of-the-Art

We qualitatively compare our method to both general image-to-image methods and deep caricature methods. Fig. 3 shows the results of comparison. For general image-to-image methods, we choose StarGAN-v2 Choi et al. (2020) and U-GAT-IT Kim et al. (2020) for comparison, as these two methods show good results in translating two domains. We train U-GAT-IT with their official implementations except setting –light to true for insufficient memory of GPU. U-GAT-IT cannot generate caricatures with rich exaggerations and texture/color changes, it merely only reconstructs the input image, sometimes it even fails to reconstruct the input image. We train StarGAN-v2 with their official implementations and same processing to the dataset with our method. StarGAN-v2 shows strong model representation ability and generates caricatures with rich texture change. But sometimes it generates artifacts. And the generated caricatures often have little exaggerations, because it is not designed for shape deformation. Compared to StarGAN-v2, our method can generate caricatures with richer texture, and detailed shape deformations. For example, our method generates caricatures with small eyes and big mouth exaggeration compared to StarGAN-v2 (2nd row), and our method generates caricatures with richer texture changes compared to StarGAN-v2 (fourth row). For deep caricature methods, we compare our method to WarpGAN Shi et al. (2019) and AutoToon Gong et al. (2020). These two methods are trained

Figure 3: Comparison with other state-of-the-art methods. We compare our method to both image-to-image translation methods and caricature generation methods. Our method can generate caricature with more detailed texture and reasonable exaggeration. The results of our method are generated given latent code.

using the author's official code in default setting. They are specially designed for generating caricatures with reasonable shape deformation. Through the results, we can see that WarpGAN cannot generate caricatures with high quality and reasonable exaggeration. That is because WarpGAN uses only 16 sparse control points to generate shape deformation which limits the rationality and the abundance of exaggeration. And WarpGAN has poor model presentation ability than our model, generated caricatures have some degree of distortion. AutoToon learns warping field to perform the facial exaggeration. And it is designed only for deformation, and needs to utilize other style transfer method to transfer style. Since we only use the official code of AutoToon, So the results don't seem like caricatures at all. But, in the results, we can find that AutoToon only generates tiny exaggerations, we think that might because AutoToon uses only 101 photo-caricature pairs to train. To summarize, these specially designed deep caricature methods often fail to generate visually pleasing results, U-GAT-IT changes only a little between the input photo and the generated caricature, StarGAN-v2 generates pretty good results, but has less exaggeration and poorer texture change than ours.

4.4. Caricature-to-Photo Translation

Since our model is a unified model, which takes both photo as input, caricature as reference and caricature as input, photo as reference, it is easy to translate caricature back into photo. We randomly sample a style code s from Gaussian distribution, then the generator takes a caricature image as input. The generated photos are shown in Fig. 4. With our semi-cycle loss our model can generate photo preserving the caricature's important features like big

Figure 4: Caricature-to-photo generation. Since we build our model in a unified way, which do not restrict the input to be photo, our model can take caricature as input, and generate fine face photo.

mouth (1st col), shape of the nose (2nd col), beard (3rd col). And, at the same time reversely deforms the exaggeration to normal. However, translating caricatures with extreme shape exaggerations can be difficult, our model is not able to learn a mapping like that, most of time, it just puts a face over the input caricature. Therefore, it should be further explored in this area.

Figure 5: Comparison with StarGAN-v2 given reference. The first column are reference images, the second column are source images, and the third and fourth column are results generated by StarGAN-v2 and our model.

4.5. Comparison with Stargan-v2 Given Reference Image

StarGAN-v2 achieves impressive results on multiple datasets, generates fine results on caricature generation and can easily generate caricature given reference image. Hence, we compare our method with StarGAN-v2. As shown in Fig. 5, StarGAN-v2 generates plausible caricatures with respect to the given reference, but StarGAN-v2 is not able to generate fine exaggeration. And if observing carefully, the caricatures generated by StarGAN-v2 tend to blur compared to our method (first row). In contrary, our method generates obvious exaggerations such as big or small eyes, exaggerated teeth. That is because we introduce

AdaSLIN into our model to capture exaggeration in caricature. With our semi-cycle loss, our model can generate higher quality caricatures, and as we analyzed before, the identity information is also well preserved.

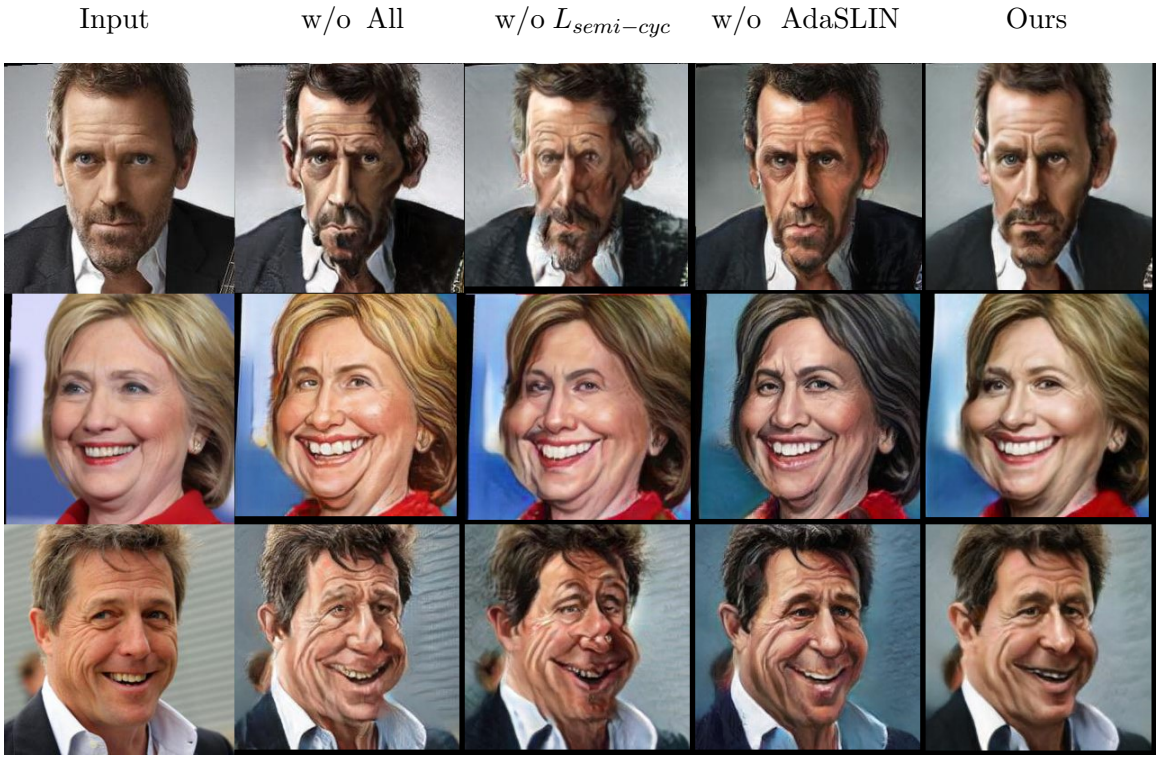


Figure 6: Ablation study for latent guide generation. The first column are inputs, the second column are results without AdaSLN and $L_{semi-cyc}$, the third column are results without $L_{semi-cyc}$, the fourth column are results without AdaSLIN, the fifth column are results of our model.

4.6. Ablation Study

To analyze our proposed AdaSLIN method and semi-cycle consistency loss, we conduct an ablation study to test the impact of different components. The results are shown in Fig. 6. Through the results, we can see that replacing the semi-cycle consistency loss with cycle consistency will result in huge decline in image quality. As we mentioned before, conducting cycle consistency will limit presentation ability of our model, because the model might be confused about generating reasonable exaggeration and preserving the original information. And our AdaSLIN through adaptively selecting LN or IN by an attention mechanism help the generator generate caricatures with a greater extent of exaggeration and fine texture.

5. Conclusion

In this paper, we propose a new method namely AdsSe-GAN for photo-to-caricature generation with AdaSLIN and semi-cycle consistency. AdaSLIN helps the model to generate

reasonable exaggeration and fine texture and semi-cycle consistency loss which only considers cycle consistency in caricature-to-photo process improves the image quality and enables our model to explore more possibility of exaggeration. There still are several problems need to be solved, even with these improvements, sometimes our model still cannot generate visually pleasing results on some special cases. And generating highly abstract but reasonable exaggerations are still difficult. In the future, we will further explore these challengeable yet interesting problems.

References

Ergun Akleman. Making caricatures with morphing. In *ACM SIGGRAPH 97 Visual Proceedings: The art and interdisciplinary programs of SIGGRAPH '97*, page 145. ACM, 1997.

Ergun Akleman, James Palmer, and Ryan Logan. Making extreme caricatures with a new interactive 2d deformation technique with simplicial complexes. In *Proceedings of visual*, pages 165–170. Citeseer, 2000.

Susan E Brennan. Caricature generator: The dynamic exaggeration of faces by computer. *Leonardo*, 40(4):392–400, 2007.

Kaidi Cao, Jing Liao, and Lu Yuan. Carigans: unpaired photo-to-caricature translation. *ACM Trans. Graph.*, 37(6):244, 2018.

Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 8789–8797. Computer Vision Foundation / IEEE Computer Society, 2018.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 8185–8194. Computer Vision Foundation / IEEE, 2020.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2414–2423. IEEE Computer Society, 2016.

Julia Gong, Yannick Hold-Geoffroy, and Jingwan Lu. Autotoon: Automatic geometric warping for face cartoon generation. In *IEEE Winter Conference on Applications of Computer Vision, WACV*, pages 349–358. IEEE, 2020.

Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE International Conference on Computer Vision, ICCV*, pages 1510–1519. IEEE Computer Society, 2017.

Xun Huang, Ming-Yu Liu, Serge J. Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu,

and Yair Weiss, editors, *Computer Vision - ECCV*, volume 11207 of *Lecture Notes in Computer Science*, pages 179–196. Springer, 2018.

Jing Huo, Wenbin Li, Yinghuan Shi, Yang Gao, and Hujun Yin. Webcaricature: a benchmark for caricature recognition. In *British Machine Vision Conference 2018, BMVC*, page 223. BMVA Press, 2018.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 5967–5976. IEEE Computer Society, 2017.

Wonjong Jang, Gwangjin Ju, Yucheol Jung, Jiaolong Yang, Xin Tong, and Seungyong Lee. Stylecarigan: caricature generation via stylegan feature map modulation. *ACM Trans. Graph.*, 40(4):116:1–116:16, 2021.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4401–4410. Computer Vision Foundation / IEEE, 2019.

Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwanghee Lee. U-GAT-IT: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. In *8th International Conference on Learning Representations, ICLR*. OpenReview.net, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR*, 2015.

Bing Li, Yuanlue Zhu, Yitong Wang, Chia-Wen Lin, Bernard Ghanem, and Linlin Shen. Anigan: Style-guided generative adversarial networks for unsupervised anime face generation. *IEEE Transactions on Multimedia*, 2021.

Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2479–2486. IEEE Computer Society, 2016.

Wenbin Li, Wei Xiong, Haofu Liao, Jing Huo, Yang Gao, and Jiebo Luo. Carigan: Caricature generation through weakly paired adversarial learning. *Neural Networks*, 132:66–74, 2020.

Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 510–519. Computer Vision Foundation / IEEE, 2019.

Zhenyao Mo, John P. Lewis, and Ulrich Neumann. Improved automatic caricature by feature normalization and exaggeration. In *International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH*, page 57. ACM, 2004.

Yichun Shi, Debayan Deb, and Anil K. Jain. Warpgan: Automatic caricature generation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 10762–10771. Computer Vision Foundation / IEEE, 2019.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 8798–8807. Computer Vision Foundation / IEEE Computer Society, 2018.

Shuai Yang, Liming Jiang, Ziwei Liu, and Chen Change Loy. Pastiche master: Exemplar-based high-resolution portrait style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7693–7702, 2022.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision, ICCV*, pages 2242–2251. IEEE Computer Society, 2017.