CS230

Unsupervised Face-to-Manga Translation with CycleGAN

Marcella Cindy Prasetio
Department of Computer Science
Stanford University
mcp21@stanford.edu

Abstract

We present a deep learning network for a novel task, unsupervised face-to-manga image translation. We explore variants of discriminator architecture for CycleGAN [14] and implement spectral normalization and self-attention into the CycleGAN to stabilize the learning process. From the quantitative and qualitative results, our models manage to generate good face-to-manga translations.

1 Introduction

Image-to-image translation aims to translate images from a source domain into a target domain. One use case for this task is image stylization in social applications. Face-to-manga translation is a rare task among image-to-image translations. For as we know, there is no preceding work for this specific task. The lack of face-manga pair images makes supervised learning an unsuitable approach for this task. Thus, we explore an unsupervised approach through CycleGAN [14]. To be specific, the CycleGAN model will take face images as input and output the translated images in the style of manga characters. One challenge for this project is the unstable learning in GANs. We explore the use of spectral normalization [9] and self-attention module [13] to stabilize the training process.

This is a shared project with CS236: Deep Generative Models. For CS236, we focus more on the generative model architecture, such as implementing the baseline convnet, the use of CycleGAN, and self-attention. For CS230, we focus more on the architecture and training experiments, such as hyperparameter and architecture searches, spectral normalization, applying deep learning knowledge for training and debugging, and analyzing the impact of different components of the model.

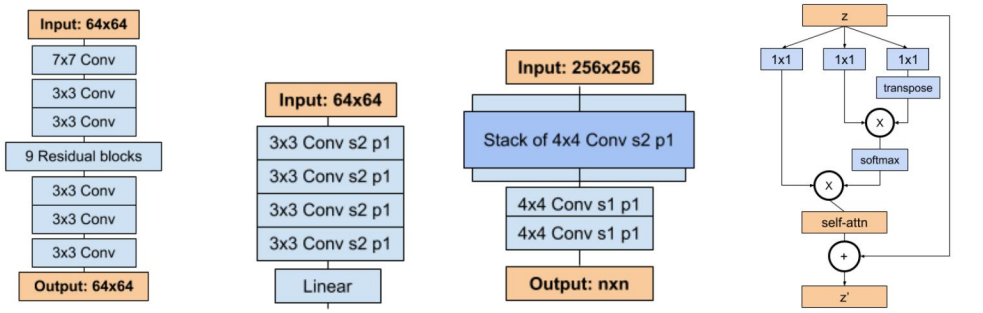
2 Related work

A common approach for image-to-image style translation is neural style transfer, which is proposed by Gatys et al. [1]. This work utilizes convolutional neural network to transfer style while preserving the content image. However, this depends only on a single style image in the target domain and might not capture the whole target domain, as this limits what the model can produce, as discussed in Huang et al. [4]. Generative adversial Network (GAN) [2], in general, is more flexible in capturing the collective style of the target domain, albeit the unstable performance. Zhu et al. [14] proposed CycleGAN, which is a state-of-the-art GAN model that achieves satisfactory result on unsupervised image-to-image translation tasks by optimizing on adversarial and cycle-consistency loss. The cycle-consistency loss guides the model to generate images that can be reconstructed back to the original images. This constraint helps the model to learn source-to-target mapping.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

(a) CelebA (b) "Aisazu Niha Irarenai" ©Yoshi Masako (c) "Rising Girl" ©Hikochi Sakuya

Figure 1: Sample images from CelebA (a) and Manga109 (b) (c)



(a) ResNet generator (b) 4-layer convnet (c) PatchGAN (d) Self-Attention module

Figure 2: Network architecture for generator and discriminator.

However, unsupervised learning in GANs can be very unstable and unpredictable. As a response, we explore the use of spectral normalization and Self-Attention GAN. Spectral normalization, proposed by Miyato et al. [9], constraints the spectral norm of the weights and does not require an extra hyperparameter to tune, making it computationally more efficient. Through this normalization, we can prevent large weight magnitudes, prevent overfitting, and help stabilize training. Meanwhile, Zhang et al. [13] proposed Self-attention GAN, where the self-attention module adds non-local feature computation into the local features from convolutional layers. This enables the network to model relationships between features that are separated in different spatial regions. CycleGAN and these two expansions are the focus of the project.

3 Dataset

We use CelebA [7] as the source domain, which contains 202,599 face images. For the target domain, we use Manga109 [1] [8] [10] dataset, which contains 10,619 Japanese comic pages and 26,602 character faces. They are available through the website sources for non-commercial use. The images are cropped, centered, and resized to 64x64. Figure 1 shows examples of the preprocessed images. We yield 2,000 images from CelebA for the final result comparison. We use open source CycleGAN model from [14] [2]. We created, modified the data and training pipelines, and extended the model.

4 Methods

Generator (G, F) We use ResNet-9 blocks as our generator, as shown in Figure 2a. This architecture, proposed by Johnson et al. [6], consists of two stride-2 convolutions, 9 residual blocks, two stride-$\frac{1}{2}$ convolutions, instance normalization, and Relu activations except for the last tanh activation.

Discriminator (D_Y, D_X) We use two architectures for our discriminator. The first is a 4-layer convolutional network, which will be our baseline as shown in Figure 2b. The second is PatchGAN [5]. As shown in Figure 2c, it consists of a stack of 4x4 convolutional layers with strides 2 and 2 4x4 convolutional layers with stride 1 that use instance normalization and LeakyRelu activations. It downsamples the image into an nxn output array, where n depends on the depth of the 4x4 convolutional layer stack, as shown in Table 2. The deeper this stack is the smaller the final output

[1] http://www.manga109.org/en/

[2] https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix

Table 1: Training settings

Hyperparameters	Value
α	0.0002
optimizer	Adam
β_1	0.5
decay rate	0.1

Table 2: PatchGAN output dimensions.

Depth (d)	Final output dimensions
3	30 x 30
4	14 x 14
5	6 x 6
6	2 x 2

size is. Each entry in this output signifies whether each patches in the input image is real or fake. The benefit of this architecture from the baseline convnet is it requires fewer number of parameters.

Loss Formulation We minimize the adversarial loss, L_{GAN}. Given source images $x \in X$ and target images $y \in Y$, the model learns the mapping function $G : X \rightarrow Y$ and the inverse mapping $F : Y \rightarrow X$ simultaneously. For the mapping function G:

$$L_{GAN}(G, D_Y) = E_{y \sim P_{data}(y)}\big[\log D_Y(y)\big] + E_{x \sim P_{data}(x)}\big[\log(1 - D_Y(G(x)))\big]$$

, where the generator optimizes to generate fake images $G(x)$ to fool D_Y and D_Y optimizes to distinguish $G(x)$ from real images y. Both components plays a minimax game, where $min_G max_{D_Y} L_{GAN}(G, D_Y)$ and vice versa for $L_{GAN}(F, D_X)$. In addition, we add a constraint through cycle-consistency loss, L_{Cycle}, where the constraint is $F(G(x)) \approx x$ and vice versa.

$$L_{Cycle}(G, F) = E_{y \sim P_{data}(y)}\big[||G(F(y)) - y||_1\big] + E_{x \sim P_{data}(x)}\big[||F(G(x)) - x||_1\big]$$

, where we minimizes the L1 distance between the reconstructed images and the original images. Overall, we minimize the full-objective function: $L(G, F, D_Y, D_X) = L_{GAN}(G, D_Y) + L_{GAN}(F, D_X) + L_{Cycle}(G, F)$. To handle the unstable learning, we implement these two following expansions.

Spectral Normalization (SN) [9] Spectral normalization fixes the spectral norm of each convolutional layer by replacing the weights W with $\frac{W}{\sigma(W)}$, where $\sigma(W)$ is the largest singular value of W. It is a form of regularization. We compute $\sigma(W)$ through power iteration, where $\sigma(W) = u^T W v$ for $W \in R^{n\,x\,m}$, $u \in R^m$ and $v \in R^n$. We randomly initialized u and v. Then, at every learning step we update u, v and calculate $\sigma(W)$. For learning step $(t + 1)$, the updates are: $u_{t+1} = W v_t,\ v_{t+1} = W^T u_{t+1}$. We replace instance normalization with spectral normalization in the discriminator.

Self-Attention Module (Attn) [13] The self-attention block consists of 1x1 convolutions and takes in features from a convolutional layer, computes the self-attention feature maps, and append it to the input features. Figure 2d shows the structure of the non-local block. Zhang et al. [13] proposes a combination of self-attention GAN and spectral normalization to stabilize GAN training. We attach this non-local block to the convolutional layers in the discriminator.

5 Experiments and Results

Table 1 shows our training setup. We train for 200 epochs. We notice the loss for the generator and discriminator fluctuates greatly. Instead of using a lower learning rate, we decay the learning rate after half of the overall epochs. We choose to use Adam optimizer for more adaptive updates, considering the unstable loss.

As quantitative metrics, we use Inception Score (IS) [12], which measures the KL divergence between marginal and conditional class distribution, and Fréchet Inception distance (FID) [3], which measures the distance between real and generated images. A higher IS indicates better image quality and a lower FID indicates a closer distance to real data. We are not using the ImageNet-pretrained Inception network as our evaluator. Instead, we are using a VGG network that is pretrained to classify anime characters[3]. We think this is a more suitable network for evaluation given that it operates in a similar domain as our task. For qualitative evaluation, we include images from a VGG neural style transfer model, which is trained on 1 style image, as an additional qualitative comparison.

[3]https://github.com/abars/AnimeFaceClassifier

Table 3: Discriminator architecture difference

Variant	IS	FID
4-layer covnet	3.200	**0.453**
PatchGAN	**4.866**	0.523
PatchGAN + SN	4.520	0.487
PatchGAN + attn	3.960	0.478
PatchGAN + SN + attn	4.074	0.523

Table 4: Image transforms

Variant	IS	FID
$n_c = 3$	2.717	0.322
$n_c = 1$	3.059	**0.321**
$n_c = 1$ + Random flip	**3.800**	0.456

Table 5: α difference

Variant	IS	FID
$\alpha_D = 0.5\alpha$	3.800	0.456
$\alpha_D = \alpha$	3.885	0.463
$\alpha_D = 2\alpha$	**3.892**	**0.451**

Table 6: PatchGAN depth

Depth (d)	IS	FID
3	3.800	**0.456**
4	**4.866**	0.523
5	4.628	0.537
6	4.736	0.541

Table 7: Depth + SN + attention

Depth (d)	IS	FID
4	4.074	0.523
5	4.266	0.492
6	**4.330**	**0.470**

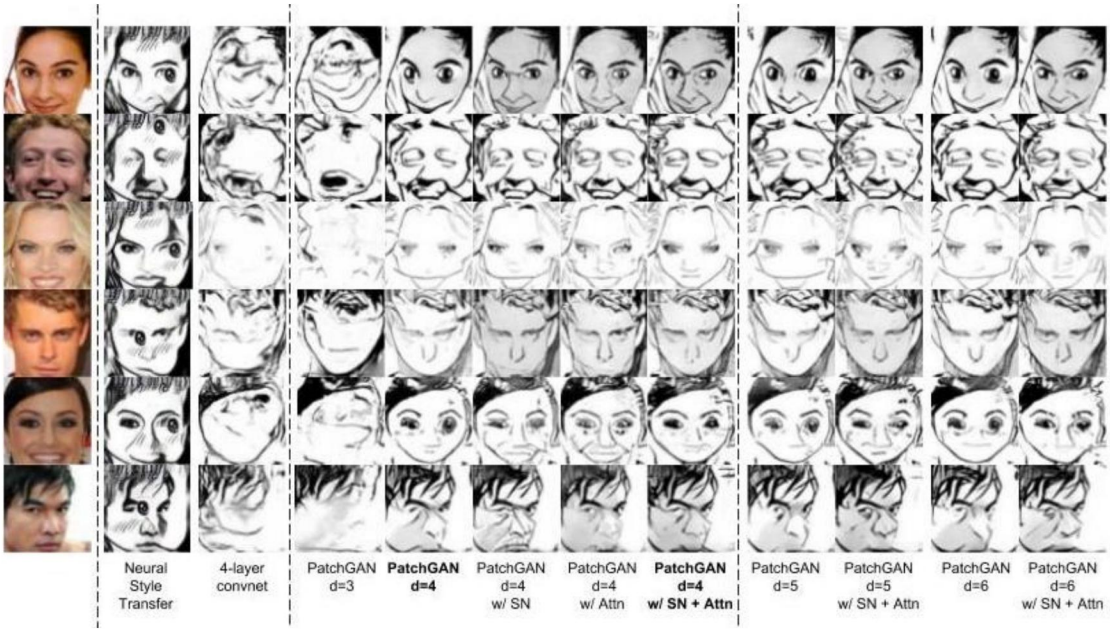


Figure 3: Generated image samples from the test set.

Architecture Search Results In Table 3, the baseline 4-layer convnet has significantly lower inception score than the PatchGAN discriminator, with slightly lower FID score. This indicates that the quality of the generated images is better with the PatchGAN discriminator. In Figure 3, the neural style transfer images have the style of manga characters but also include similar patches that most likely come from the style image. We observe that the CycleGAN models can generate more diverse manga style images. For the baseline, the model fails to properly translate the faces. We suspect that the generator can simply fool the discriminator with these images. In contrast, the models with PatchGAN discriminator shows decent translations, where we can clearly see the facial features. Given that PatchGAN uses patches of image to infer the realness of the image, we suspect that the generator is forced to generate more facial features throughout the image space. Furthermore, the spectral normalization adds more details, which we infer because it regularizes the discriminator, while the self-attention makes the images more realistic, which comes from the non-local feature computation. Combining the two components together, some of the generated images have better quality and more facial features. They have more complicated lines and shapes in the generated images. However, notice that there is no significant differences in the inception and FID scores between these two expansions and the vanilla CycleGAN as shown in Table 3. We suspect that the quantitative metric is not sensitive to small changes in the images. Thus, we use the quantitative

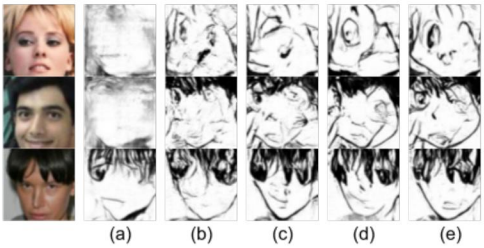


Figure 4: Different input and training settings. RGB input ($n_c = 3$) (a), Grayscale input ($n_c = 1$) (b), $n_c = 1$ + random flip (c), $n_c = 1$ + random flip + $\alpha_D = \alpha$ (d), $n_c = 1$ + random flip + $\alpha_D = 2\alpha$ (e)

metrics as a rough indicator of how the models are doing compared to the baseline. Note that the PatchGAN in Table 3 uses a stack depth of 4.

Input and Training Experiment Results From Table 4, using grayscale ($n_c = 1$) images and adding more noise with random flipping can increase the model's performance in terms of the quality measured by the inception score. Figure 4a, b, c shows the generated images, where for RGB inputs, the model fails to recreate faces entirely. We suspect that color channels add unnecessary complexity to the model learning process. Note that the PatchGAN in these experiments uses a stack depth of 3. Next, we explore the effect of different learning rate for the discriminator (α_D). From Table 5, making the discriminator learn more frequently, by increasing the learning rate, can improve the inception and FID scores. By increasing the learning rate, the discriminator can learn faster than the generator. We suspect that instead of simply modifying the learning rate, we can improve the architecture of the discriminator to improve the overall CycleGAN performance significantly. Thus, we focus more on different improvements on the discriminator side as we can see in the other experiments we present.

PatchGAN Depth Experiments Additionally, we experiment with different stack depth for PatchGAN. Table 2 shows the different stack depths and output dimensions. From the baseline comparison, using a PatchGAN of depth 4 gives a decent performance. In Table 6, without any additional components, PatchGAN with depth 4 performs better quantitatively. On the other hand, in Table 7, with spectral normalization and attention, the deeper PatchGAN is, the better the quantitative performance is. However, when we look into the generated images in Figure 3, there is no significant differences between different depths, although on some images, deeper stack gives better looking images. We surmise that having larger patches, which what deeper PatchGAN is doing, doesn't necessarily help identify whether the generated image is a fake or real manga image because manga images have less details than face images. As such, we infer that PatchGAN of depth 4 is already suitable for this task.

6 Conclusion/Future Work

To sum up, the CycleGAN with ResNet-9 blocks generator and PatchGAN discriminator learns a decent face-to-manga translation. Compared to a neural style transfer, the model manages to generate more diverse images. Moreover, we have seen that better discriminator architecture can significantly improve the performance of the model. This also applies to the spectral normalization and self-attention module. From the result, we observe that the spectral normalization adds more details to the images, generating clearer results. Meanwhile, the self-attention module makes the images more realistic, by highlighting more of the facial features than the other models. We infer that these two expansions capture more details and non-local features from the source and target domain.

Although, the models look promising, it is still a hard task to accurately measure the performance of the model with a quantitative metric. Therefore, as future work, we want to explore more suitable quantitative metrics or evaluation procedures that can better capture the performance of the models, especially for GANs and unsupervised image-to-image translation. Furthermore, this project has been more focused on exploring the different architecture for the discriminator. Thus, next, we want to explore and experiment with the generator. This can include using a deeper generator or other additions and expansions for the generator architecture. All in all, the final results of the project meet our expectation, such that we have a functional face-to-manga translation model.

7 Contributions

The project was carried out by Marcella Cindy Prasetio.

8 Code

The project is located in a public repository and is implemented mainly using PyTorch [11]:

https://bitbucket.org/MCindy/unsupervised-face-to-manga-translation/src/master/.

References

[1] 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, 2016.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[3] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017.

[4] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. *arXiv preprint arXiv:1804.04732*, 2018.

[5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks.

[6] Justin Johnson, Alexandre Alahi, and Fei-Fei Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016.

[7] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[8] Yusuke Matsui, Kota Ito, Yuji Aramaki, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *CoRR*, abs/1510.04389, 2015.

[9] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[10] Toru Ogawa, Atsushi Otsubo, Rei Narita, Yusuke Matsui, Toshihiko Yamasaki, and Kiyoharu Aizawa. Object detection for comics using manga109 annotations. *CoRR*, abs/1803.08670, 2018.

[11] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[12] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016.

[13] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

[14] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017.