Proceedings of the 4th International Conference on Computing Innovation and Applied Physics
DOI: 10.54254/2753-8818/83/2025.20028

Image-to-pixel-art Translation Based on CycleGAN

Yifei Hang[1,a,*]

[1]*Applied and Computational Mathematical Sciences Program, University of Washington, WA, USA*
a. yhang2@uw.edu
**corresponding author*

Abstract: Image style transfer has gained significant attention in the computer vision fields in recent years, especially with the emergence of generative models. While numerous style transfer tasks have been handled by various models, image-to-pixel-art translations were not extensively explored, which is seemingly trivial yet requires delicacy in practice. To this end, this paper introduces the Pixel-Landscape-CycleGAN (PL-CycleGAN), which is a CycleGAN model that addresses the translation from, but not limited to, real-world landscape images to pixel art. The model is quantitatively evaluated using Frechet Inception Distance (FID) and Kernel Inception Distance (KID) scores, and got 85.472 for FID and 0.0366 for KID. These scores achieved a 52.46% and 64.98% reduction in comparison to the scores from interpolation. Further visual analysis also proved the model's efficacy at translating to pixel art while maintaining the initial content of landscape images, as well as its constant performance when encountering non-scenery objects within images.

Keywords: Image style transfer, CycleGAN, deep learning, landscape.

1. Introduction

Image style transfer is a computer vision task that fuses the artistic style of one image with the content of another, "re-drawing" artworks using particular styles without the need of experienced artists [1]. Although earlier algorithms or frameworks, such as non-photorealistic rendering and image analogies developed around the 2000s, had decent performances, they were often limited by their ability to generalize or capture image structures effectively [1-3]. Within the past decade, neural-network-based algorithms emerged with the progressions in deep learning. Early attempts in the field included neural style transfer developed by Gatys et al., which employed convolutional neural networks (CNNs) for extracting and combining features from content and style images [4]. This method's success proved the efficacy of neural network methods and paved the way for further research.

With further advancements in the field, researchers designed and implemented numerous algorithms with greater power and flexibility, in particular the generative models. Some prevalent models include: Pix2Pix, a conditional GAN (cGAN) that performs image-to-image translation on paired datasets; Cycle-Consistent Adversarial Network (CycleGAN), an image-to-image translation GAN model on unpaired sets of images; StarGAN, a GAN for translation across multiple image sets, and Palette, a diffusion-based model utilizing a U-Net architecture for the translation tasks [5-8]. These models differ from neural style transfer in that rather than re-styling one content image with one style image using a pre-trained convolutional neural network, they learn higher-level features in

DOI: 10.54254/2753-8818/83/2025.20028

sets of images. The translated images are thereby more generalizable and structural similar to the target image set.

Amongst the generative models, CycleGAN stood out as the most representative method with its crucial and innovative characteristic, unpaired image-to-image translation where the two sets of images for training do not need to be in correspondent pairs. Such feature provides convenience in data collection and flexibility in the choice of tasks. Furthermore, GAN models, including CycleGAN, are more preferable than diffusion models, such as Palette, due to their lower memory consumption, fewer data sample necessity, and faster inference speeds.

CycleGAN has been extensively applied and tested on numerous translation tasks that ranges from various art forms to medical imaging, such as Object transfiguration, painting-to-photo translation, Chinese font translation, dance style transfer, and MRI image translation [6, 9-11]. However, its translation to pixel art style was rather overlooked. Pixel art was initially introduced due to limited screen resolutions and colors for gaming consoles and computers, but it remains a common choice of art style for various media due to its uniqueness and accessibility [12]. While creating pixel art seems to be trivial, pixel artists need adequate techniques to place pixels and represent items in a pixelized canvas [12]. Similarly, translating an image to pixel style could not be done by simply down-scaling it to a lower resolution, but requires specific alternations in shapes and colors. Therefore, a model that converts images to pixel art can simplify the translation process and assist pixel artists' creations. It can further be a tool for non-artists to create pixel art with ease.

This paper introduces a CycleGAN model named Pixel-Landscape-CycleGAN (PL-CycleGAN) that addresses the particular problem of translating real-world landscape images to pixel art. The model produces pixelized landscape images that could be used in various settings of diverse styles, such as background images for pixel-style games or artworks. The model was trained on two sets of images, the real-world landscape dataset and the pixel art dataset, and the model performance was evaluated quantitatively and qualitatively. Quantitative metrics of Frechet Inception Distance (FID) and Kernel Inception Distance (KID) were applied to generated samples along with qualitative observations.

2. Methods

2.1. Revisiting GANs

Generative Adversarial Network (GAN) is a type of neural network framework initially proposed by Goodfellow et al. in 2014, and is now widely used in textual, image, and audio processing [13]. GANs stood out from other frameworks due to its unique adversarial structure, which consists of two models, a generator and a discriminator, that are trained simultaneously with the adversarial loss. The generator is an implicit density model that learns the data distribution without any explicit hypothesis or parameter fitting procedures [14, 15]. Taking samples from random noise variables with defined distribution as input, the generator aims to map them from the noise space to the data space using a differentiable function. On the contrary, the discriminator is trained as a binary classification model, which learns to discriminate the generated data from the true data.

The adversariality of the model is introduced by its objective function, also known as the adversarial loss, where a minimax game is played between the generator and the discriminator. The value of the game is defined as the formulation (1):

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim P_{data}(x)}\left[\log D(x)\right] + \mathbb{E}_{Z\sim P_Z(z)}\left[\log(1 - D\left(G(z)\right)\right] \quad (1)$$

where z is the noise input, x is the true data, G is the mapping from noise space to data space, and D is the discriminator that outputs whether the given data is true or fake.

DOI: 10.54254/2753-8818/83/2025.20028

During the training process, the generator and the discriminator alternate between fixing and updating parameters. The generator first fixes its parameters after generating a batch of fake data, while the discriminator trains to minimize the following expectation:

$$\mathcal{J}(D) = \mathbb{E}_{x\sim P_{data}(x)}\left[-\log D(x)\right] + \mathbb{E}_{Z\sim P_z(z)}\left[-\log(1 - D\big(G(z)\big)\right] \tag{2}$$

where $-\log D(x)$ calculates the cross-entropy loss between 1, the label for real data, and $D(x)$, the discriminator's output for real data x. Similarly, $-\log(1 - D\big(G(z)\big)$ calculates the cross-entropy loss between 0, the label for fake data, and $D\big(G(z)\big)$, the discriminator's output for generated fake data $G(z)$. Essentially, the discriminator aims to improve its ability of assigning correct labels to the real or fake data, thereby distinguishing the generated from the true. Then, the discriminator fixes its parameters while the generator trains to maximize the generation loss, as:

$$\mathcal{J}(G) = \mathbb{E}_{Z\sim P_z(z)}\left[-\log(1 - D\big(G(z)\big)\right] \tag{3}$$

Contrary to the discriminator's objective, the generator trains itself to generate fake data that confuses the discriminator, hence improving the quality of the generated data.

2.2. CycleGAN

CycleGAN was designed by Zhu et al. as a representative variant of the vanilla GANs framework, which addressed the unpaired image-to-image translation task [5]. In contrast to vanilla GAN, which learns a mapping from noise space to data space, CycleGAN learns to map between two different data spaces. For the two domains X and Y, CycleGAN trains two generators for mappings $G: X \to Y$ and $F: Y \to X$. To evaluate the two sets of generated data, $F(Y)$ and $G(X)$, the model further uses two discriminators, D_X and D_Y, that aims to distinguish x from $F(y)$ and y from $G(x)$, respectively.

Similar to the vanilla GAN, CycleGAN's objective function includes the adversarial loss. With two distinct discriminators and mappings, two separate adversarial losses are introduced for each generator-discriminator pair. For the mapping G and discriminator D_Y, the adversarial loss is expressed as:

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y\sim p_{data}(y)}\left[\log D_Y(y)\right] + \mathbb{E}_{x\sim p_{data}(x)}\left[\log(1 - D_Y\big(G(x)\big)\right] \tag{4}$$

The loss for mapping F and discriminator D_X is defined in a similar manner by:

$$\mathcal{L}_{\text{GAN}}(F, D_X, Y, X) = \mathbb{E}_{x\sim p_{data}(x)}\left[\log D_X(x)\right] + \mathbb{E}_{y\sim p_{data}(y)}\left[\log(1 - D_X\big(F(y)\big)\right] \tag{5}$$

While using solely the adversarial loss can theoretically learn the two mappings, the generators can map an input sample to an arbitrary sample in the target data space, which does not guarantee the preservation of input sample's content in the generated data. To get a more meaningful generated data, the model adds another crucial property known as cycle consistency, or that for each sample mapped from its original data domain to the target domain, mapping it back to its original domain should produce something similar to, if not the same as, the original sample. For a sample x from the domain X, for instance, $F(G(x))$ should be approximately equal to x, and similarly, $G(F(y))$ should be approximately equal to y for all y in domain Y. Based on the above property, cycle consistency loss is introduced to minimize the difference between the original data and the cyclically-mapped data, defined as:

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x\sim p_{data}(x)}\left[\left\|F\big(G(x)\big) - x\right\|_1\right] + \mathbb{E}_{y\sim p_{data}(y)}\left[\left\|G\big(F(y)\big) - y\right\|_1\right] \tag{6}$$

A final component of the model's objective function tackles the situation where the input sample is already in the target domain. The transformed data should ideally remain unchanged, or that

DOI: 10.54254/2753-8818/83/2025.20028

$F(x) \approx x$ for all x in X and $G(y) \approx y$ for all y in Y. Such identity mapping is driven by the identity mapping loss, defined as:

$$\mathcal{L}_{\text{identity}}(G,F) = \mathbb{E}_{x\sim p_{data}(x)}\left[\|F(x)-x\|_1\right] + \mathbb{E}_{y\sim p_{data}(y)}\left[\|G(y)-y\|_1\right] \quad (7)$$

Although the identity mapping loss was not included in the initial implementation from Zhu et al.'s paper, they discovered that this additional loss helped to preserve the color of the input after mapping to the target domain.

The loss functions add up to the full objective of CycleGAN, as:

$$\mathcal{L}_{\text{cyc}}(G,F,D_X,D_Y) = \mathcal{L}_{\text{GAN}}(G,D_Y,X,Y) + \mathcal{L}_{\text{GAN}}(F,D_X,Y,X)$$
$$+\lambda_1\mathcal{L}_{\text{cyc}}(G,F) + \lambda_2\mathcal{L}_{\text{identity}}(G,F) \quad (8)$$

where λ_1 and λ_2 controls the importance balance between losses. Again, a minimax game is played between the generators and the discriminators, aiming to solve:

$$G^*,F^* = \arg\min_{G,F}\max_{D_X,D_Y}\mathcal{L}(G,F,D_X,D_Y) \quad (9)$$

2.3. Data selection and preprocessing

The PA-CycleGAN model was trained on two sets of image data from Kaggle: the Curated Pixel Art dataset with approximately 1000 pixel art images, and the Landscape Pictures dataset with over 4000 real-world landscape images of higher resolution then the pixel art dataset [16, 17]. The landscape dataset was chosen based on an empirical comparison between higher-resolution and lower-resolution datasets regarding the quality and generalizability of translated images. Due to the relatively small sample size of the pixel art dataset, data augmentation of horizontal flipping was applied, which expanded the size by two.

3. Results

3.1. Quantitative Evaluation

The model's performance was quantitatively evaluated by two evaluation metrics: Frechet Inception Distance (FID) and Kernel Inception Distance (KID), and the results for PL-CycleGAN vs simple interpolation are shown in Table 1. The comparison between the two methods aims to demonstrate the efficacy of the generative model in pixel art translation.

Table 1: Frechet Inception Distance (FID), Kernel Inception Distance (KID) scores and KID standard deviation of generated samples by the PL-CycleGAN model vs interpolated images

Methods	FID	KID	KID-SD
PL-CycleGAN	85.472	0.0366	0.0060
Interpolation	179.775	0.1045	0.0133

FID measures the Wasserstein-2 distance between the multi-dimensional Gaussian distributions fitted to extracted features of the true and the generated datasets [18]. A smaller FID score indicates better performance of a model. The FID score of the generated pixel landscapes was 85.472, which was about 52.46% less than the FID of interpolated landscape images, which was 179.775. The significantly lower score indicated that the model successfully learned a mapping that translates real-world images closer to pixel style. It is also worth noting that FID can be highly biased against smaller

DOI: 10.54254/2753-8818/83/2025.20028

sample size, or that the FID score from a small dataset is an overestimation of the true dataset [19]. While 50k is the reasonable minimum [19], the size of the pixel art dataset was only about 1k, which indicated that the true FID was smaller than presented and that the model was effectively generating pixel landscape images.

Similar to FID, KID also measures the distance between the two feature distributions. But rather than fitting a Gaussian distribution, KID measures maximum mean discrepancy (MMD) over multiple subsets of features, and then calculate their mean and standard deviation [20]. The mean is known as the KID score, which was 0.0366 for PL-CycleGAN model. Comparing to the KID score of interpolation, which was 0.1045, PL-CycleGAN model had a more significant improvement in KID score by 64.98%, proving the model's pixelization ability. On the other hand, the standard deviation of feature subsets' MMDs reveals consistency of generated images. The standard deviation score of 0.060 suggested the model's ability of generating consistent images.

3.2. Qualitative Evaluation

The generalized pixel-style landscapes were also evaluated qualitatively by a comparative observation of the original versus generated landscape images, with examples shown in figure 1. The generated images successfully kept the real-world images' content while showing evident characteristics of pixel art, such as fewer and more vibrant colors, pixelized edges with defined boundaries, and removal of unnecessary details. These observations further proved the model's efficacy of translating landscape images from real-world to pixel-style.

Furthermore, the pixelization was not limited to pure scenery images. As shown in figure 2, the model continued to produce promising results with the existence of non-landscape objects, whether they were animal creatures or human beings. These objects also remained recognizable after transforming into pixel style, indicating the model's ability of distinguishing meaningful and negligible information, as well as its generalizability beyond solely landscape images.

Figure 1: Examples of real-world landscape images (top) and pixelized images (bottom)

DOI: 10.54254/2753-8818/83/2025.20028

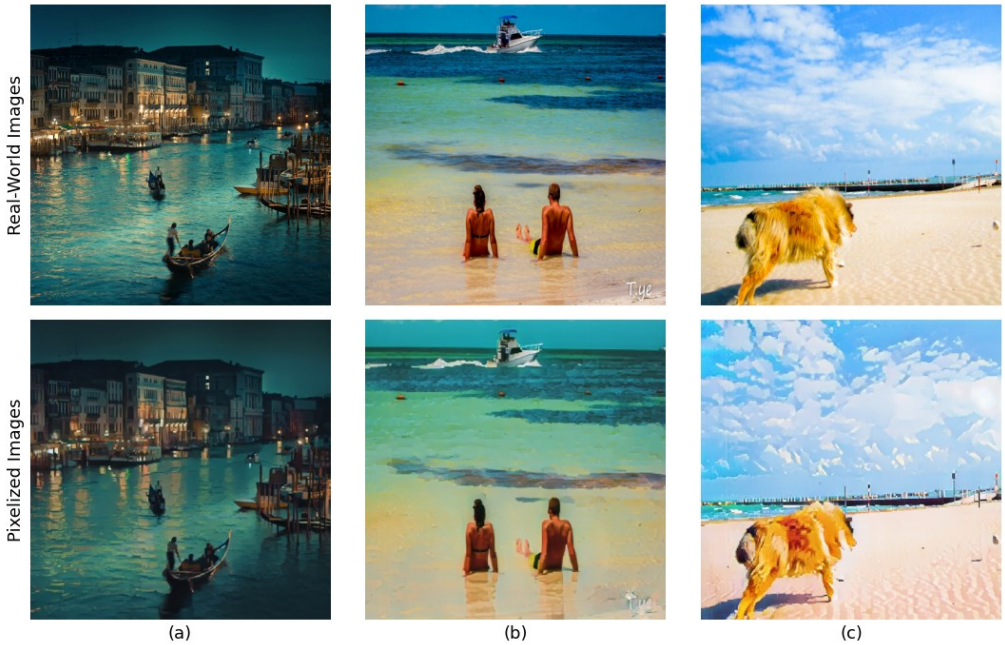

Figure 2: Examples of real-world images (top) and pixelized images (bottom) with non-landscape components

Despite the overall favorable results, the model exhibited occasional inconsistencies. In Figure 3a, for instance, while the model successfully pixelized the mountain and grassland of the original image, it created horizontal artifacts along the water body and the sky, as well as unnatural colors within the cloud, which resulted in a less-than-ideal visual effect. Further, grassland in Figure 3b was recognized as and pixelized in a rocky texture due to shades from roadside trees. One may also notice that the parasol in Figure 3c was rather realistic, indicating deficient pixelization. Potential causes of these noise may be insufficiency in data samples and content variations within each image dataset. Since the pixel art dataset was not exclusively consisted of landscape images, but rather included non-realistic scenes, fantasy creatures, and cartoon characters, the mapping between the two datasets may be of higher complexity than the model could learn given the limited data samples.

Figure 3: Examples of real-world images (top) and pixelized images (bottom) with noise

DOI: 10.54254/2753-8818/83/2025.20028

4. Discussion

Although PL-CycleGAN showed favorable results, further work could be done to improve the model's performance. One major limitation of this model was the datasets it was trained on, which were both small in sample sizes and variegated in content. Further training and testing could be done on datasets with considerably more samples. Regarding image content, one modification would be to collect more data of similar content, which in this case is landscape. To train a more generalizable model, a different approach could be to use the progressive growing method, where a model is trained on simpler tasks at the start and increases the complexity progressively [21]. Thereby, the model can learn more complex tasks with stabilized results. Applying this method to the pixel art translation task may lead to more consistent outputs.

Besides image quality, another room for improvement was variations in style. While it was ideal that the model consistently produced pixelized landscapes, the generated images were of similar, realistic style, which may fail to satisfy the needs of certain users. One potential direction is to explore the feasibility of combining conditional GAN (cGAN) with CycleGAN, leveraging the strengths of cGAN for condition input to control the level of realness of generated pixel art.

5. Conclusion

This paper presents a CycleGAN model named Pixel-Landscape-CycleGAN (PL-CycleGAN) that aimes to address the task of translating real-world landscape images to pixel art. Through quantitative analysis of the translated images, the model gained FID and KID scores that were both over 50% less than the scores of applying interpolation method, which suggested the model's efficacy in translating the original image to pixel art. Furthermore, one can tell from qualitative observations that the model successfully captured characteristics of pixel art, regarding both color choices and boundary styles. However, the model's dataset was limited in sample size and varied in image content, which led to noise in certain samples, including vertical artifacts, unnatural color and texture alternations, or lack of pixelization. Nonetheless, the model proved its ability of pixel art translation and potential of generating more flawless samples, and further work on the data and the model structure could be done for future improvements.

References

[1] *Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y., & Song, M. (2018). Neural style transfer: A Review. IEEE Transactions on Visualization & Computer Graphics, 26(11), pp. 3365-3385.*

[2] *Gooch, B., & Gooch, A. A. (2001). Non-Photorealistic Rendering. AK Peters/CRC Press.*

[3] *Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B., & Salesin, D. H (2001). Image analogies. SIGGRAPH '01: Proceedings of the 28th Annual Conference on Computer Graphics and interactive techniques. pp. 327-340.*

[4] *Gatys, L., Ecker, A., & Bethge, M. (2016). A neural algorithm of artistic style. Journal of Vision, 16(12).*

[5] *Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1125-1134.*

[6] *Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2020). Unpaired image-to-image translation using cycle-consistent adversarial networks. 2017 IEEE International Conference on Computer Vision, pp. 2242-2251.*

[7] *Choi, Y., Choi, M., Kim, M., Ha, J. W., Kim, S., & Choo, J. (2018). Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8789-8797.*

[8] *Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., & Norouzi, M. (2022). Palette: Image-to-image diffusion models. ACM SIGGRAPH 2022 conference proceedings, pp. 1-10.*

[9] *Zeng, J., Wang, Y., Chen, Q., Liu, Y., Wang, M., & Yao, Y. (2022). Strokegan+: Few-shot semi-supervised chinese font generation with stroke encoding. arXiv preprint arXiv:2211.06198.*

[10] *Yin, W., Yin, H., Baraka, K., Kragic, D., & Björkman, M. (2023). Dance style transfer with cross-modal transformer. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 5058-5067.*

[11] *Czobit, C., & Samavi, R. (2023). CycleGAN Models for MRI Image Translation. arXiv preprint arXiv:2401.00023.*

DOI: 10.54254/2753-8818/83/2025.20028

[12] *Zufri, T., Hilman, D., & Frans, O. (2022). Research on the application of Pixel Art in game character design. Journal of Games, Game Art, and Gamification, 7(1), pp. 27-31.*
[13] *Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems, 2. pp. 2672-2680.*
[14] *Goodfellow, I. (2016). Nips 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160.*
[15] *Gui, J., Sun, Z., Wen, Y., Tao, D., & Ye, J. (2021). A review on generative adversarial networks: Algorithms, theory, and applications. IEEE transactions on knowledge and data engineering, 35(4), 3313-3332.*
[16] *Vandaley, A. (2022). Curated pixel art 512X512. Kaggle. https://www.kaggle.com/datasets/artvandaley/curated-pixel-art-512x512*
[17] *Rougetet, A. (2020). Landscape Pictures. Kaggle. https://www.kaggle.com/datasets/arnaud58/landscape-pictures*
[18] *Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems, 30.*
[19] *Borji, A. (2022). Pros and cons of GAN evaluation measures: New developments. Computer Vision and Image Understanding, 215, 103329.*
[20] *Bińkowski, M., Sutherland, D. J., Arbel, M., & Gretton, A. (2018). Demystifying mmd gans. arXiv preprint arXiv:1801.01401.*
[21] *Karras, T. (2017). Progressive Growing of GANs for Improved Quality, Stability, and Variation. arXiv preprint arXiv:1710.10196.*