Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 5
AIDML 2024

Exploiting Improved Cyclic Generative Adversarial Networks to Generate Images from Different Angles of Views

Yiquan Ding *

School of Computer Engineering, Nanjing Institute of Technology, Nanjing, Jiangsu, 211167, China

* Corresponding Author Email: Yiquan_Ding@njit.edu.cn

Abstract. Virtual Reality (VR), Augmented Reality (AR), and other similar technologies are becoming increasingly popular nowadays. However, despite the increasing number of related devices being launched, the feedback tends to be disappointing. A significant contributor to this dissatisfaction is the blurred viewpoints and unclear levels, which often lead to a suboptimal user experience. To address this issue, this research focuses on training a modified version of the Cycle-Consistent Generative Adversarial Networks (CycleGAN) model to generate an image from another angle of view based on a given image. This work has fine-tuned specific layers of the CycleGAN to better suit the requirements of this work. These improvements give the model enhanced adaptivity in handling visual image transformations. Consequently, the proposed model achieves superior results compared to other models, with minimal image distortion. This is likely due to the minimal angular difference between the generated images. In the future, it is expected to expand the number of viewpoints generated and enhance the model's efficiency in processing images with varying resolutions.

Keywords: CycleGAN; image generation; view of image; angle of view.

1. Introduction

Nowadays, there is a high demand for Virtual Reality (VR), Augmented Reality (AR), and various other technologies, along with their corresponding devices. They are used in many applications, whether in daily recreation, agriculture, or industrial production [1, 2]. These technologies are aimed at providing safety and efficiency to the users. For the human eye to be more comfortable and immersed in using the device, the coordination of the left and right eye perspectives is essential [3].

People can perceive three-dimensional space because of the convergence and regulation mechanism of the eyes, which makes the spatial information collected by the human eye have a certain horizontal parallax [4]. The brain makes joint decisions through physiological stereopsis and mental stereopsis and then acquires stereoscopic visual perception. This type of problem often involves taking an image as input, and then the model outputs another or even multiple new views. These views are similar to the original image but with little difference in perspective. However, the depth of confidence they build out is even more immersive.

Previously, several works are based on deep learning models. Oliveira, et al., adapted and used a hierarchical image super-pixel algorithm to maintain the structural features of the scene in the process of image reconstruction [5]. Xuesong, et al., used the Depth Image Based Rendering (DIBR) algorithm to synthesize at any position between two reference cameras based on deep learning. Post-processing techniques are utilized to reduce the noise generated by rendering, including hole filling and median filtering [6]. Olivia, et al., created SynSin, a tool that can transform a 3-dimensional (3D) point cloud of a feature into a target view. It helps test interpretable operations on the feature space and generates realistic output images. SynSin can also generate high-resolution images and work with other input resolutions [7]. Richard, et al., introduced scale-invariant view synthesis for supervision. By breaking the scene into multiple planes and modeling and rendering each plane separately. Then fusing information from different planes to reconstruct the depth and detail of the scene more accurately [8]. Miaomiao, et al., combines images and geometric information. Their network can better understand the structure and shape of the scene and respect the scene structure more [9].

This paper improves the Cycle-Consistent Generative Adversarial Networks (CycleGAN) model and trains it to generate binocular images [10]. The generated results are evaluated by commonly used evaluation metrics, expecting to achieve better visual effects in visual sensation and image quality.

2. Method

2.1. CycleGAN

The core principle of CycleGAN lies in acquiring the essence of the source domain image and the artistic style of the target domain image. Its original mission was style transfer, as demonstrated in Fig. 1.

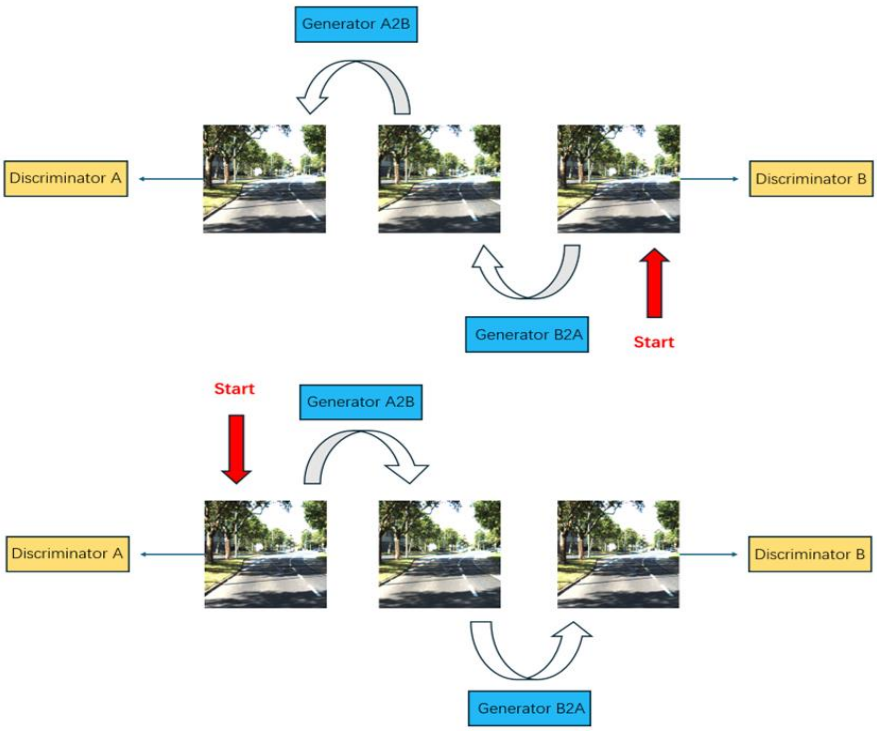

Fig. 1 CycleGAN explanation diagrams (Figure Credits: Original).

"Cycle" is the central concept of CycleGAN. One can translate a domain X sample x to a domain Y distribution x2y, which in turn should map back to the original domain X sample x2y2x. The cyclic consistency process is this one. One way to conceptualize the concept of CycleGAN is as the transformation of one class of images into another. In other words, there are now two sample spaces, X and Y, and it is anticipated that the samples in X will be converted to samples in Y. CycleGAN can help us convert images to each other. CycleGAN does not require data pairing to convert images.

To cope with the dual-view task (left and right eye view transformation), this work fine-tunes the model and lets it learn the identical information and difference between the left and right images while training. After training these view conversions, the model can generate the virtual view for the input image.

Adversarial loss could be denoted as:

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

Adversarial loss is present in almost all GAN functions. The objective is to guarantee that the difference between the produced image and the initial image is minimal. The above public notation is constraining the mapping function X to Y. Similar functions are used on the mapping function Y to X as well.

Cycle consistency loss could be denoted as:

$$\mathcal{L}_{\text{cyc}}(G,F)\& = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\| F(G(x)) - x \|_1] + \mathbb{E}_{y\sim p_{\text{data}}(y)}[\| G(F(y)) - y \|_1] \quad (2)$$

However, if the network's capacity is sufficient, it can convert input photos to arbitrary images in the target domain. The correct mapping should be X to Y to X. Instead of using a learned mapping X to a to X to match output distribution to target distribution, where a is meaningless. Therefore, it is not sufficient to use only the adversarial loss as a constraint, but also need to add the cyclic consistency loss to further constrain the space of the mapping function.

The total loss could be:

$$\mathcal{L}(G,F,D_X,D_Y)\& = \mathcal{L}_{\text{GAN}}(G,D_Y,X,Y) + \mathcal{L}_{\text{GAN}}(F,D_X,Y,X) + \lambda\mathcal{L}_{\text{cyc}}(G,F) \quad (3)$$

Cycle Consistency Loss and Adversarial Loss Combination is the loss function used in this paper. It greatly improves the generalization ability of the model, reduces the time required for training, and can better avoid overfitting.

2.2. Discriminator Architecture

The structure of the Discriminator network used in this study consists of a series of Conv2d layers, LeakyReLU activation function, and InstanceNorm2d layers. The Discriminator finally outputs a value indicating the probability that the generated image is a true alternate view image. More specifically, the network consists of four Conv2d layers, each followed by a LeakyReLU activation function. After the 2nd and 3rd Conv2d layers, this work also added an InstanceNorm2d layer for normalization operations on the image channels, which enhances the model's capacity for generalization. Finally, this method added a Conv2d layer for outputting the Discriminator's predictions. By employing this architecture, the suggested model has the ability to grasp intricate characteristics from authentic images while also being capable of distinguishing whether the produced image is similar to the authentic one.

2.3. Generator Architecture

The Generator network structure used in this study is composed of a series of Conv2d layers, ReLU activation function, ConvTranspose2d, Residual Block, and InstanceNorm2d layers. The Generator finally outputs a value, which is the generated image of the other viewpoint. In detail, the network consists of five Conv2d layers. First, the input image is boundary-filled through the ReflectionPad2d layer, and the edges of the input image are mirror-filled to preserve the information on the image edges. The next step involves the conversion of the input into a feature map with the help of the Conv2d layer. After this conversion, each batch is subjected to instance normalization using InstanceNorm2d to enhance the model's generalization ability. Subsequently, the process is repeated, expanding the output until it meets the resolution requirements of the generated image. The Generator network also incorporates a Residual Block, which serves to help the network better learn the details and textures of the image by adding jump connections. The Residual Block allows information to be directly transferred in the network by adding a jump between the front and back convolutional layers, allowing the information to be transferred directly in the network. By adding a jump connection between the front and back convolutional layers, the Residual Block allows information to pass directly through the network without being transformed by multiple layers of convolutional operations. Thus it helps the network to better capture the subtle features of an image. This design helps mitigate the problems of gradient vanishing and gradient explosion and helps speed up the training and optimization process of the network. Finally, the required output is obtained by passing the ConvTranspose2d layer and then activating it with the Tanh function.

2.4. Evaluation Indexes

Two indexes are leveraged for validation.

Peak Signal-To-Noise Ratio (PSNR): The peak signal-to-noise ratio is a commonly used metric that compares a new image to the raw image based on the difference between each pixel. The variable *I* is the raw image. The variable *K* is the generated image. The function Max takes the maximum pixel number.

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE}\right) \tag{4}$$

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{5}$$

Structural Similarity Index (SSIM): Structural similarity is a metric, that reflects the similarity between two images from several aspects which are luminance, contrast, and structure. μ_x, μ_y is the mean of x, y.σ_x, σ_y is the variance of x, y. σ_{xy} is the covariance of x and y.

$$\text{SSIM}\ (x,y) = [l(x,y)]^{\alpha}[c(x,y)]^{\beta}[s(x,y)]^{\gamma} \tag{6}$$

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \tag{7}$$

$$c(x,y) = \frac{2\sigma_{xy} + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \tag{8}$$

$$s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \tag{9}$$

3. Result and Discussion

3.1. Dataset

This paper uses the KITTI dataset, which is an open dataset that is frequently utilized in computer vision and autonomous driving research [11]. The dataset contains a large number of images and point cloud data of real-world scenes acquired from multiple sensors carried on vehicles as shown in Fig. 2.

The KITTI dataset consists of several sub-datasets covering a variety of scenes, such as city streets, country roads, highways, etc. This work use 11089 images for each view, which they are paired with. Those data have high quality, which helps me train and test computer vision models.

Fig. 2 Representative images from the KITTI dataset [11].

3.2. Quantitative Results

Experiments are conducted using a learning rate of 0.0002 with Adam optimizer. Fig. 3. is the change of 5 indicators with the training epoch and shows the process of training. As the training goes on, the loss_G_GAN value should gradually increase and the other 4 values should decrease. As you can see, the 3 indicators of loss_G, loss_G_cycle, and loss_G_identity are constantly reduced, which means that the model is gradually converging. Although the curves of the other two indicators go up and down, it could be found that a valley point is near 80. From these loss values and the test results corresponding to the epoch, this work finally chooses the epoch training results at 80 as the final model parameters.

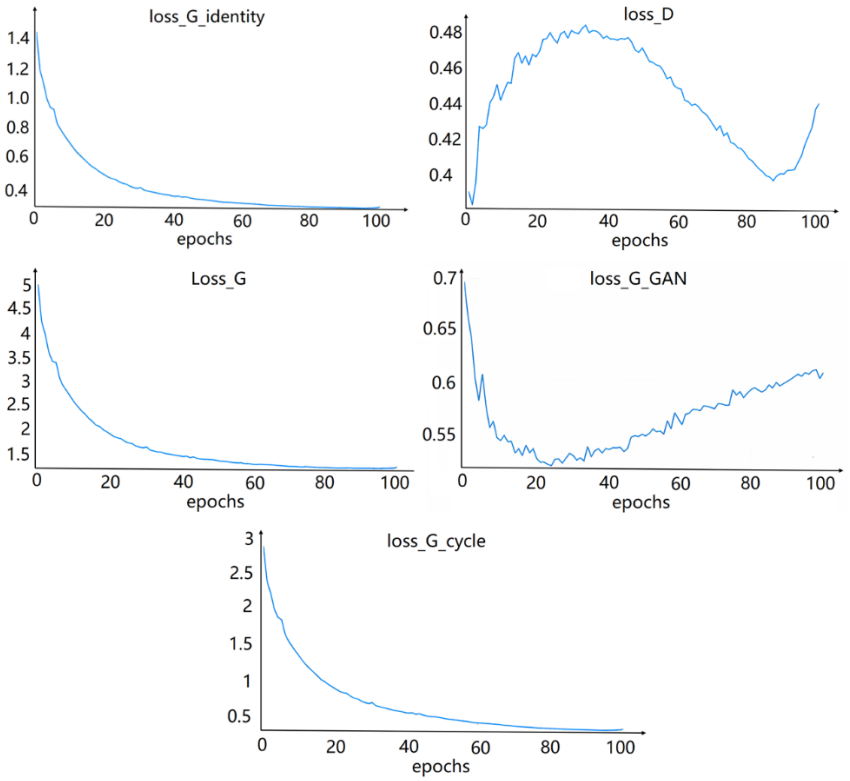

Fig. 3 Loss curves of the model (Figure Credits: Original).

By comparing the contents of the two red rectangles in Fig. 4, it could be observed that the bottom details of the tree are restored well. In addition to this Fig. 5, other test results also prove that the model has a certain ability to deal with partially occluded information.

This work applies the model to the KITTI test dataset, and the results speak for themselves. According to the assessment methods mentioned above, the proposed model scored 14.887 for PSNR and 0.382 for SSIM on the KITTI dataset, demonstrating its good performance.

Fig. 4 Detailed visualization result (Figure Credits: Original).

Fig. 5 Visualization results (Figure Credits: Original).

4. Conclusion

In fact, the difference between the two pictures is small, and the difficulty is mainly centered on the part of the difference between the different viewpoints. The larger the parallax, the greater the difference in information within the viewpoint. However, given that for the mainstream application environment (the human eye), the difference between the two viewpoints ranges between 65 mm and 75 mm. It is only necessary to generate a more approximate image of the other viewpoint. The model can be generated moderately according to the content of the input image, and good results can easily be achieved. Future work will further improve the clarity of the generated virtual viewpoints and continue to try to generate more viewpoints to further enhance the user's visual experience.

References

[1] Kavanagh, Sam, Andrew Luxton-Reilly, Burkhard Wuensche, and Beryl Plimmer. A systematic review of virtual reality in education. Themes in science and technology education, 2017, 10(2): 85-119.

[2] Anthes, Christoph, Rubén Jesús García-Hernández, Markus Wiedemann, and Dieter Kranzlmüller. State of the art of virtual reality technology. In 2016 IEEE aerospace conference, 2016: 1-19.

[3] Wang, Lei, Wei Chen, Wenjia Yang, Fangming Bi, and Fei Richard Yu. A state-of-the-art review on image synthesis with generative adversarial networks. Ieee Access, 2020, 8: 63514-63537.

[4] Seitz, Steven M., and Charles R. Dyer. Physically-valid view synthesis by image interpolation. In Proceedings IEEE Workshop on Representation of Visual Scenes, 1995: 18-25.

[5] De Oliveira, Adriano Q., Thiago LT da Silveira, Marcelo Walter, and Cláudio R. Jung. A hierarchical superpixel-based approach for DIBR view synthesis. IEEE Transactions on Image Processing, 2021, 30: 6408-6419.

[6] GAO, Xuesong, Keqiu Li, Weiqiang Chen, Zhongyao Yang, Wenguo Wei, and Yangang Cai. Free viewpoint video synthesis based on DIBR. In 2020 IEEE Conference on Multimedia Information Processing and Retrieval, 2020: 275-278.

[7] Wiles, Olivia, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020: 7467-7477.

[8] Tucker, Richard, and Noah Snavely. Single-view view synthesis with multiplane images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020: 551-560.

[9] Liu, Miaomiao, Xuming He, and Mathieu Salzmann. Geometry-aware deep network for single-image novel view synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: 4616-4624.

[10] Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision, 2017: 2223-2232.

[11] KITTI dataset. URL: https://www.kaggle.com/datasets/klemenko/kitti-dataset. Last Accessed Time: 2024/03/16.