How to Cite:
Chohan, A., & Hablani, R. (2022). Inception generators for generative adversarial networks. *International Journal of Health Sciences*, *6*(S1), 10495–10507. https://doi.org/10.53730/ijhs.v6nS1.7535

Inception generators for generative adversarial networks

Abhishek Chohan
Department of Computer Science & Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur, India
Corresponding author email: chohanabhishek844@gmail.com

Dr. Ramchand Hablani
Department of Computer Science & Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur, India

***Abstract*---**Image-to-image translation aims at learning a mapping between two different picture styles. The style and texture of one image can be transferred to another. It is possible to do paired and unpaired image-to-image translations, both of which have a specified goal translation to aim for translating images. Using generative adversarial networks, we offer an overview of different generator designs for unpaired image-to-image translation and paired image-to-image translation. The Residual blocks or U-net architecture are often employed in generators of generative adversarial networks for image-to-image translation. In this research paper, we propose the use of an Inception module and their evaluation in unpaired image-to-image translation tasks by translating photos to artist images and also with unpaired data for converting Maps to aerial images.

***Keywords*---**Inception module, generative adversarial networks, image to image translation, Computer Vision, Machine Learning, Deep learning.

Introduction

Fecund Artists in the 1800s were limited by the local demographic that they stayed in and thus a large proportion of their exemplars were inspired by these. What if they weren't limited to their local demographics, Deep learning helps us in making this a reality. Gatys [4] developed a neural style transfer algorithm that takes a pre-trained model such as VGG-19 [5] as the feature extractor, gets outputs from certain layers of the model to compute the style and content loss, and then uses it to update the content image. Unpaired translation using cycle-gans [6] puts forward a generative adversarial network [7] like no other, wherein

we get a generator for converting an image from style A to B while retaining the features of A, as a plus point of the cycle architecture we also get a generator for converting images from style B to A. CycleGANs are also used in image segmentation as shown in [6]. The original pix2pix [8] paper used a U-net [2] architecture while the Resnet was used in the CycleGAN [6]. In this paper, we provide an overview of cycle-gan [6] using the generator with a residual block [1] and a U-net [2] block architecture and a proposal of an Inception block consisting of features from the Inception module [3] and residual block [1]. We have chosen CycleGANs specifically for their bidirectional conversion operations, for evaluation of the Resnet, U-net generators, and the proposed inception generators.

Previous Works

U-net ARCHITECTURE

The U-net [2] architecture has upsampling and downsampling layers with skip connections between them. The encoder part of the network consists of a series of downsampling layers, each downsampling layer consists of a convolutional layer and an instance normalization layer reducing the image size gradually while cumulatively increasing the depth of the image. The decoder, on the other hand, is made up of transpose convolutional layers that increase the image size while decreasing the image depth. There are skip connections between the same level layers in the network. The intuition behind skip connections is that it allows us to pass some earlier information to the later layers. The final dense layer in the model gives us the translated image. The original application of U-net was image segmentation for the tasks where the output image is of similar size to the input image.

*Due to computation limitations, in this experiment, we used 8 skip connections for our model**.**

Resnet ARCHITECTURE

The input layer is followed by a convolutional layer performing 7x7 convolutions, followed by two convolutional layers performing 3x3 convolutions, whose output is fed to a series of Resnet [7] blocks. The Resnet [7] block consists of Convolutional layers and Instance normalization after each layer. The Resnet architecture aims at accuracy and is computationally expensive. Here we use a skip connection as in U-net [2], by concatenating the block's output with its input, the Resnet, like U-net, permits information to pass from older layers to travel to later layers via skip connections. The outputs from the Resnet block are upsampled using a series of transpose convolutional layers, the final dense layer giving the Translated image.

We propose the use of inception module in the generator for GANs having use-cases in segmentation and style transfer tasks, with a comparative analysis with the Resnet, U-net generators, and the proposed inception generators with the inception generators achieving better Fréchet Inception distance(FID) score with less number of parameters.

Implementation

Here we have no specific target output, the aim is to learn a mapping between the two different types of image style and examine common elements between the two types: content and style respectively. Unlike the paired image to image translation, this method is unsupervised as stated in the unpaired image to image translation paper [6]. The cycle gan [6] has four components which are: Generator for converting images from style A to B, Generator for converting images from style B to A, Discriminator D_A to identify whether an image input of style A is fake or not, Discriminator D_B to identify whether an image input of style B is fake or not. First $G_{A\rightarrow B}$ will map a real image A to a fake image of B and then the other generator will map B back to source style A. What cycle consistency expects for the fake A generated to be exactly the same as the real one because only styles should've changed in the process. The pixel to pixel difference is added to our loss function and we aim for the two images to be as close as possible. We use $_{cycle}$ tune this loss accordingly.

Identity loss is used to preserve the color in the image and, takes the real image A and puts it through the $G_{B\rightarrow A}$ to get fake A, since the real image A already has the desired style/features in the image we want these images to be nearly identical and we implement identity loss to minimize the distance between these images. Here we used Least Square Loss [9] as our adversarial training loss.

The Total loss can be given by:

$$L_{identity}(G_{A\rightarrow B},G_{B\rightarrow A}) = E_{B\text{-}Pdata(B)}[||\ G_{A\rightarrow B}(B) - B||] + E_{A\text{-}Pdata(A)}[||\ G_{B\rightarrow A}(A) - A||] \quad (1)$$

$$L_{GAN}(G_{A\rightarrow B},D_B,A,B) = E_{B\text{-}Pdata(B)}[\log D_B(B)] + E_{A\text{-}Pdata(A)}[\log(1-D_Y(G_{A\rightarrow B}(A))] \quad (2)$$

$$L_{GAN}(G_{B\rightarrow A},D_A,B,A) = E_{A\text{-}Pdata(A)}[\log D_A(A)] + E_{B\text{-}Pdata(B)}[\log(1-D_A(G_{B\rightarrow A}(B))] \quad (3)$$

$$L_{cycle}(G_{A\rightarrow B},G_{B\rightarrow A}) = E_{A\text{-}Pdata(A)}[||\ G_{B\rightarrow A}(G_{A\rightarrow B}(A)) - A||] + E_{B\text{-}Pdata(B)}[||\ G_{A\rightarrow B}(G_{B\rightarrow A}(B)) - B||] \quad (4)$$

$$L(G_{A\rightarrow B},G_{B\rightarrow A},D_A,D_B) = L_{GAN}(G_{A\rightarrow B},D_B,A,B) + L_{GAN}(G_{B\rightarrow A},D_A,B,A) + \lambda_{cycle}L_{cycle}(G_{A\rightarrow B},G_{B\rightarrow A}) + \lambda_{identity}L_{identity}(G_{A\rightarrow B},G_{B\rightarrow A}) \quad (5)$$

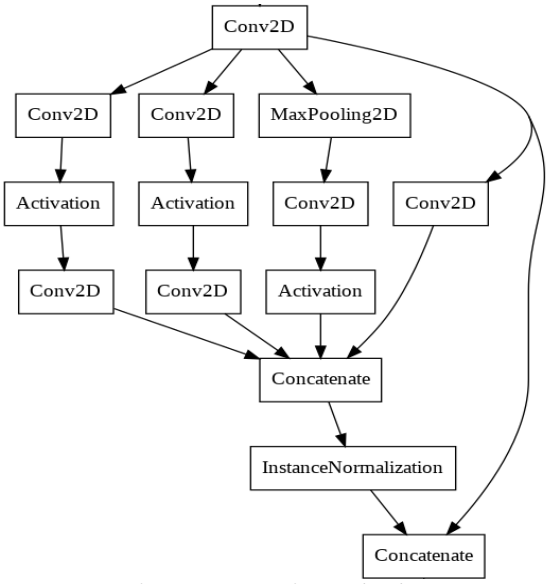

Fig. 1: Inception Block.

There are four parallel pathways for processing data in this module. The first path consists of a 1x1 convolution layer, followed by a 3x3 convolution layer, and then a ReLu activation layer. The second path consists of a 1x1 convolution layer, followed by a 3x3 convolution layer, and finally a ReLu activation layer. The third path is made up of a 1x1 convolution layer, a 5x5 convolution layer, and a ReLu activation layer. The fourth path includes 3x3 maximum pooling, 1x1 convolutions, and a ReLu activation layer. The main idea behind employing multiple pathways is that we don't know which filter will work best for the task, so we use three alternative types of filters and stack the results against each other, and we keep the padding 'same' for each of them, so they're the same height and breadth. The concatenation of these pathways is passed through an instance normalization layer, which normalizes each output channel. The result is then combined with the entire block's input. As training advances, the model learns which filter size or combination of filter sizes performs better when translating images from source to target style.

Similar to the Resnet block, here we concatenated the output of the block with its input to pass the information from earlier layers to the later layers. In the experiment conducted, the input layer is followed by a convolutional layer performing 7x7 convolutions, followed by two convolutional layers performing 3x3 convolutions the output of which is fed to the Inception blocks, the generator consists of 7 such Inception blocks, output from these blocks is upsamples using 2 transpose convolutional layers, followed by the last convolutional layer with 7x7 filters and ‘tanh’ activation, giving us the final generated image.
*For all the generator architectures, we used the discriminator as Patch gan [10] architecture which outputs a patch of 30x30.

Evaluation

In this research paper, we trained three models of gans for each dataset with generators as the Inception, U-net [2] block, and the Resnet block respectively. Evaluating gans for image translation tasks is challenging as there is no target output and therefore it's an open area. There are various measures that could be taken between the target distribution and the predicted distribution such as Inception Score, Fréchet Inception Distance, Inception Score [11], Perceptual path length [12], and many more as mentioned in [13]. In this experiment, we took Fréchet Inception distance(FID). FID was first introduced by [14] and was then introduced as a better version of the Inception Score. FID is a measure of the difference between the real image from target style and generated fake image, it is the multivariate Fréchet distance calculated using the Inception model [3].

$$FID = ||\ \mu_x - \mu_x||^2 + Tr(\mathbf{\Sigma}_x + \mathbf{\Sigma}_y - (2*\mathbf{\Sigma}_x\mathbf{\Sigma}_y)^{½}) \quad (6)$$

Image-Paintings

In this section, we evaluate the performance of CycleGANs with U-net, Resnet, and Inception generator models. As the training advances, we can observe each generator getting closer and closer to the target distribution based on the FID values of the produced distribution calculated over the Image↔Paintings with the respective art style taken for training. We fed the images to the InceptionV3 model and then calculated the mean and covariance values and then used the formula in the equation above to calculate the Fréchet Inception distance for the two distributions. For the Image↔Painting translation we trained the respective models on four datasets used in [6]. FID values were used as the deciding factor while evaluating the quality of distribution produced.

Fig. 2: Image translation from image to Van Gogh, Cézanne, Ukiyo-e, and Monet from left to right respectively obtained using the Resnet Inception model.

Table 1
Fréchet Inception distance at an interval of 10 epochs for each model for various artists

Artist	$Resnet_{10}$	$Resnet_{20}$	$Resnet_{30}$	$U\text{-}net_{10}$	$U\text{-}net_{20}$	$U\text{-}net_{30}$	$Resnet\text{-}Inception_{10}$	$Resnet\text{-}Inception_{20}$	$Resnet\text{-}inception_{30}$
Cézanne	225.36	209.00	203.37	203.92	197.12	195.88	261.48	205.85	202.56
Monet	132.56	130.42	131.09	131.71	120.31	104.26	132.31	125.27	111.71
Van Gogh	297.89	280.65	272.00	315.02	278.12	267.14	270.34	266.65	285.41
Ukiyo-e	332.51	319.66	309.41	405.03	352.42	318.87	349.45	322.15	307.99

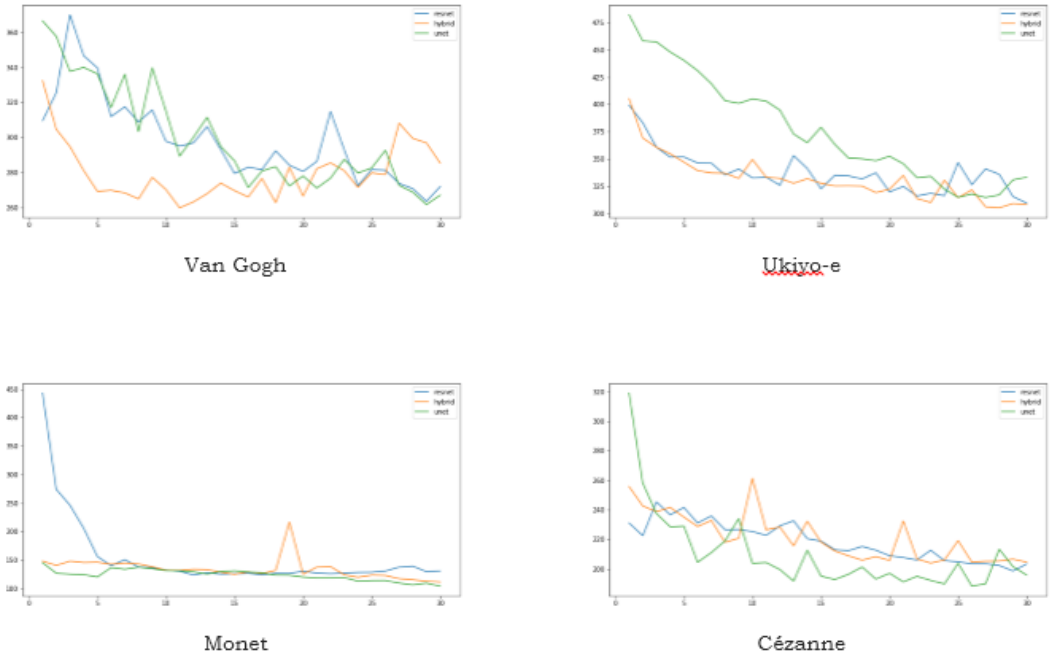

Fig. 3: FID progress for three models calculated for Van Gogh, Ukiyo-e, Monet and Cézanne art styles Respectively

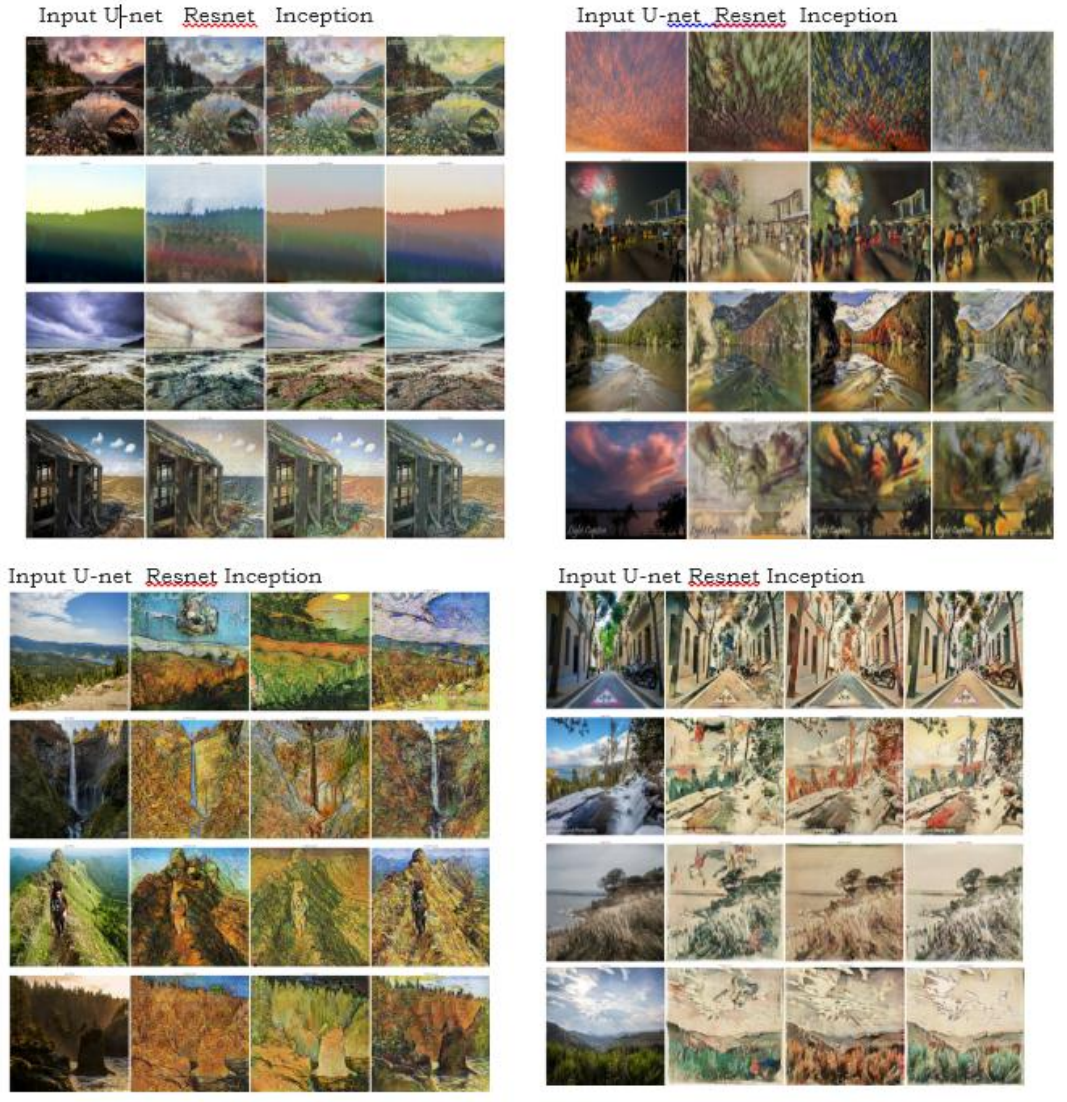

Fig. 4:Image↔Monet, Image↔Cézanne, Image↔Van Gogh, Image↔Ukiyo-e conversion for U-net, Resnet, and the Inception block architecture

Maps-Areal

In order to evaluate the models on paired data, we also trained each model type on the maps-aerial [15] dataset. Here we monitored the cycle loss as well to keep track of the image quality of the image cycle. We added the pixel-to-pixel loss between the generated image and the target image to the adversarial loss for the generator loss.

Table 2
Evaluation metrics for Maps-areal translation

	$\text{CycleLoss}_{A\to B\to A}$	$\text{CycleLoss}_{B\to A\to B}$	FID
Inception block	0.4647	0.3853	260.284
U-net block	0.4957	0.3865	268.897
Resnet block	0.6074	0.4533	315.742

Table 3
Comparison of different generator architectures

	Parameters	Size (in MB)
Inception block	5.1 M	8.4
U-net block	54.4 M	212.6
Resnet block	35.2 M	138

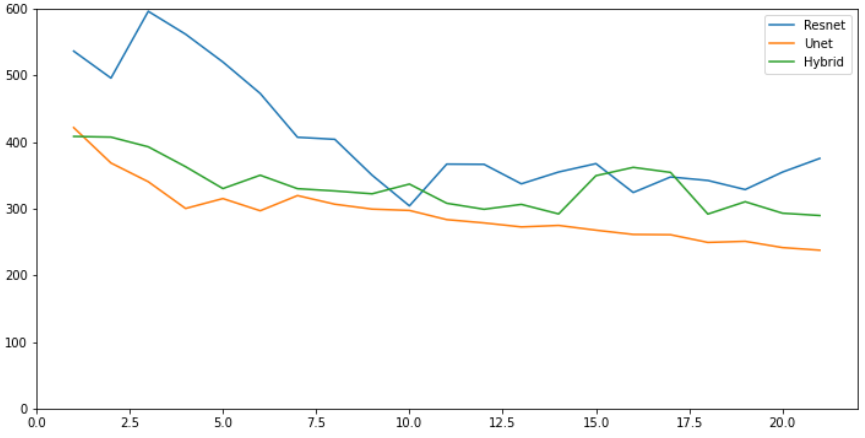

Fig. 5: FID progress for three models calculated for maps→photo conversion.

Fig. 6: Maps→photo conversion and back to maps for the U-net, Resnet, and Inception models

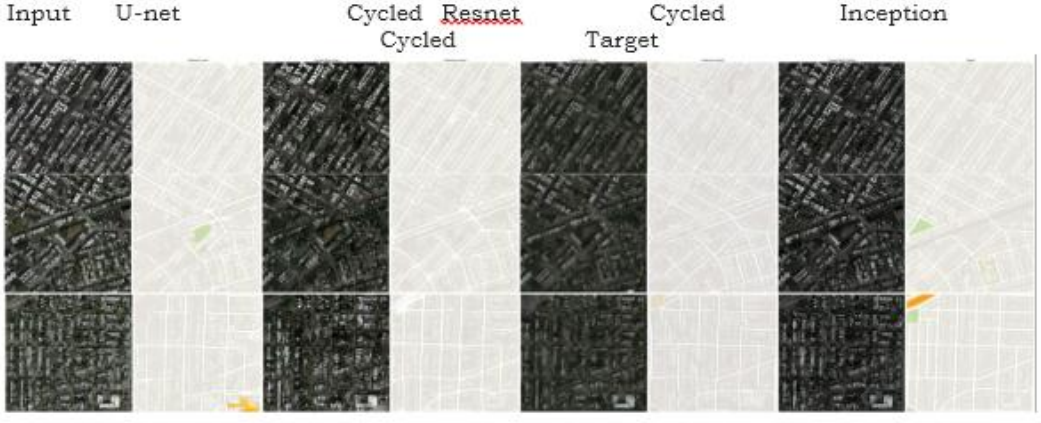

Fig. 7: photo→map conversion and back to maps for the U-net, Resnet, and Inception models

Fig. 8: Images Cycle back for the Images→Paintings translation

Training: While training the GAN, The optimizer adam was used with the values of learning rate as 2e-4, $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ as 0.5 and 0.999 respectively. Images were batched with a batch size of 1. For the Images→Paintings and maps→photo tasks, $\lambda_{cyc}$and λ_{id} were both set to 0.5 and 10 respectively. The T100 GPU was used for training the models. Each model was trained for 30 epochs, each epoch consisting of 1500 steps. Data Preprocessing: We resize the images to a scale of 256x256, and were normalized across all three channels [16]

Datasets

For the maps→photo we used a maps-aerial [15] dataset. For the image-> paintings datasets we used the dataset used in [6], In our experiment, we used paintings datasets from artists Claude Monet, Vincent van Gogh, Ukiyo-e, and Paul Cézanne.

Conclusions

The model's parameter count is drastically reduced by replacing the nine R256 blocks with the Inception block and concatenating the output with the layer input. Eventually, each model converged, putting the Frechet Inception score on the distribution in the same ballpark as the others. Apart from the FID values the translated images had different colors than The cycled images, however, were different in each case. The cycled images in the U-net model sometimes contained patches, whereas the Resnet and Inception were good at preserving the features from the source image.

Using Inception to reduce the size of the parameters had little to no effect on the translated images. We found that the Inception model produced images of

comparable, if not superior, quality to the distribution's FID, and that it has the added benefit of being small, lowering computational costs. The U-net model produced patches in the image after about 10 epochs of training, while the Resnet block architecture aimed for accuracy. We can conclude that the Inception generator maintained the translation and was less computationally expensive. When the fakes are cycled back to the original source, the features in the source image are well preserved. Thus Inception blocks can be used when there are computational constraints and can also be used in other generative adversarial networks for image translation tasks in place of U-net and Resnet.

References

1 He, K., Zhang, X., Ren, S., & Sun, J. (2016, June). Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). http://dx.doi.org/10.1109/cvpr.2016.90

2 Ronneberger, O., Fischer, P., &Brox, T. (2015, May 18). U-Net: Convolutional networks for biomedical image segmentation. ArXiv.Org. http://arxiv.org/abs/1505.04597

3 Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., &Rabinovich, A. (2015, June). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). http://dx.doi.org/10.1109/cvpr.2015.7298594

4 Gatys, L., Ecker, A., &Bethge, M. (2016). A neural algorithm of artistic style. Journal of Vision, 16(12), 326. https://doi.org/10.1167/16.12.326

5 Simonyan, K., & Zisserman, A. (2014, September 4). Very deep convolutional networks for large-scale image recognition. ArXiv.Org. https://arxiv.org/abs/1409.1556

6 Zhu, J.-Y., Park, T., Isola, P., &Efros, A. A. (2017, October). Unpaired image-to-image translation using cycle-consistent adversarial networks. 2017 IEEE International Conference on Computer Vision (ICCV). http://dx.doi.org/10.1109/iccv.2017.244

7 Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., &Bengio, Y. (2020). Generative adversarial networks. Communications of the ACM, 63(11), 139–144. https://doi.org/10.1145/3422622

8 Isola, P., Zhu, J.-Y., Zhou, T., &Efros, A. A. (2016, November 21). Image-to-Image translation with conditional adversarial networks. ArXiv.Org. http://arxiv.org/abs/1611.07004

9 Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., &Smolley, S. P. (2017, October). Least squares Generative Adversarial Networks. 2017 IEEE International Conference on Computer Vision (ICCV). http://dx.doi.org/10.1109/iccv.2017.304

10 Isola, P., Zhu, J.-Y., Zhou, T., &Efros, A. A. (2017, July). Image-to-Image translation with conditional adversarial networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). http://dx.doi.org/10.1109/cvpr.2017.632

11 Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016, June 10). Improved techniques for training gans. ArXiv.Org. http://arxiv.org/abs/1606.03498

12 Karras, T., Laine, S., & Aila, T. (2019, June). A style-based generator

architecture for generative adversarial networks. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). http://dx.doi.org/10.1109/cvpr.2019.00453

13 Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, & Li Fei-Fei. (2009, June). ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition. http://dx.doi.org/10.1109/cvpr.2009.5206848

14 Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., &Hochreiter, S. (2017, June 26). GANs trained by a two time-scale update rule converge to a local nash equilibrium. ArXiv.Org. http://arxiv.org/abs/1706.08500

15 Bastani, F., & Madden, S. (2021, October 10). Beyond Road Extraction: A Dataset for Map Update using Aerial Images. ArXiv.Org. http://arxiv.org/abs/2110.04690

16 Zhang, H. (2014). Stochastic theta method for a reflected stochastic differential equation. Numerical Functional Analysis and Optimization, 35(6), 752–776. https://doi.org/10.1080/01630563.2013.83706