

Article

# Vegetation Greening for Winter Oblique Photography Using Cycle-Consistence Adversarial Networks

Xiaowei Xue <sup>1</sup> , Chunxue Wu <sup>1</sup> , Ze Sun <sup>2</sup>, Yan Wu <sup>3</sup>  and Neal N. Xiong <sup>1,\*</sup> 

<sup>1</sup> School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China; xxw199423@163.com (X.X.); wcx@usst.edu.cn (C.W.)

<sup>2</sup> School of Tourism and Geography Science, Jilin Normal University, Changchun 130024, China; sun\_ze@sina.com

<sup>3</sup> School of Public and Environment Affairs, Indiana University, Bloomington, IN 47408, USA; wuyan8910@126.com

\* Correspondence: xiongnaxue@gmail.com; Tel.: +86-150-0184-2414

Received: 27 May 2018; Accepted: 18 July 2018; Published: 20 July 2018



**Abstract:** A 3D city model is critical for the construction of a digital city. One of the methods of building a 3D city model is tilt photogrammetry. In this method, oblique photography is crucial for generating the model because the visual quality of photography directly impacts the model's visual effect. Yet, sometimes, oblique photography does not have good visual quality due to a bad season or defective photographic equipment. For example, for oblique photography taken in winter, vegetation is brown. If this photography is employed to generate the 3D model, the result would be bad visually. Yet, common methods for vegetation greening in oblique photography rely on the assistance of the infrared band, which is not available sometimes. Thus, a method for vegetation greening in winter oblique photography without the infrared band is required, which is proposed in this paper. The method was inspired by the work on CycleGAN (Cycle-consistence Adversarial Networks). In brief, the problem of turning vegetation green in winter oblique photography is considered as a style transfer problem. Summer oblique photography generally has green vegetation. By applying CycleGAN, winter oblique photography can be transferred to summer oblique photography, and the vegetation can turn green. Yet, due to the existence of "checkerboard artifacts", the original result cannot be applied for real production. To reduce artifacts, the generator of CycleGAN is modified. As the final results suggest, the proposed method unlocks the bottleneck of vegetation greening when the infrared band is not available and artifacts are reduced.

**Keywords:** intelligence big data; adversarial network; oblique photography; vegetation greening

## 1. Introduction

A 3D city model is critical for the construction of a digital city. It is broadly used to provide information for urban planning, construction, management and emergency response. One of the methods to build a 3D city model is tilt photogrammetry [1]. Such a method generates the model using oblique photography, in which the visual quality of oblique photography directly impacts the final results' visual effect. Accordingly, it is important to take oblique photography with good visual quality.

Yet, due to the bad effects from defective photographic equipment, bad weather or season, oblique photography may have poor visual quality sometimes. One of the situations is taking oblique photography in winter. For oblique photography taken in winter, the vegetation is usually brown, and the image color is not bright. If this photography were used to generate a 3D model [2], the result would be poor, as shown in Section 4. To generate a model with good visual quality, it is required to improve the oblique photography's visual quality. Yet, common methods like [3] need oblique

photography's infrared band to help turn the vegetation green, which is not available sometimes. Hopefully, a method capable of turning vegetation green without the infrared band can be found.

In recent years, the convolutional neural network has been applied in a large number of domains, and great success has been achieved. In the image-to-image translation area, Cycle-consistency Adversarial Networks (CycleGAN) [4] arouses much attention by virtue of its excellent performance. It is capable of capturing features of one image collection and finding out how these features could be translated into another image collection. One of the impressive examples is its amazing transformation between summer style and winter style. Inspired by this, vegetation greening in winter oblique photography could serve as a style transfer problem for the reason that oblique photography taken in summer usually has green vegetation. To verify the feasibility of this assumption, winter oblique photography is converted into summer oblique photography using CycleGAN. As the result suggests, vegetation becomes green after applying CycleGAN in winter oblique photography. Yet, "checkerboard artifacts" are also found in transferred winter oblique photography. To reduce "checkerboard artifacts", CycleGAN's generator is modified with respect to its kernels. As the final result suggests, vegetation becomes green after the transformation, and artifacts are successfully reduced, as shown in Section 4.

To sum up, the contributions of our work are listed as follows.

- (1) Vegetation greening in winter oblique photography is achieved. In comparison with common methods, the infrared band is no longer required.
- (2) Checkerboard artifacts are reduced after CycleGAN is modified. The transferred photography can be applied in production.
- (3) The model can be trained with unpaired images, which is practical.

The rest of this paper is organized as follows: In the next section, we review the relevant work about unpaired image-to-image translation, GAN and cycle consistency. The proposed method is illustrated in Section 3. The comparison between the proposed method and other methods is drawn in Section 4. The last section draws the conclusions and discusses the future work.

## 2. Related Works

Unpaired image-to-image translation: The concept of image-to-image translation was first proposed in [5]. Then, numerous methods for image-to-image translation have been proposed. On the whole, these methods can fall into two groups. The first one is based on paired images for training, and the second one is based on unpaired images.

For methods [5–10] in the first one, a lack of paired images poses a big challenge [4]. In practice, it is hard and expensive to prepare paired images for training. One of the examples is artistic stylization. For every input image to be stylized, it is hard to prepare its corresponding output because these desired outputs are highly sophisticated. To overcome the limitation of insufficient paired images, methods that do not require paired images for training have been proposed.

In [11], image-to-image translation based on unpaired images was achieved using a Bayesian network. In [12], a Bayesian network was combined with a neural network to perform efficient inference, so that a direct probabilistic model can be learned. In [13], an unsupervised image-to-image translation network based on a variational autoencoder and GAN was proposed. It helped realize learning without paired images under GAN. In [14], GAN was also employed to help build a Coupled Generative Adversarial Network (CoGAN), in which the network can learn joint distributions of different styles of images. In recent work, CycleGAN [4] aroused great attention by virtue of its state-of-the-art performance. One example was its amazing translation between summer and winter.

Generate adversarial network: The Generate Adversarial Network (GAN) has a short history. In 2014, GAN was firstly proposed in [15], and great success had been achieved. Then, different types of GAN were proposed. In [16], a Laplacian pyramid was applied to an adversarial network, so that coarse images could be made fine. In [17], a deep convolutional generative adversarial network was proposed. It narrowed the gap between supervised learning and unsupervised learning. The work

in [18] proposed a recurrent adversarial network. It could generate image samples for training. The work in [19] came up with an interpretable representation learning using Information maximizing Generative Adversarial Network (InfoGAN). It was capable of learning disentangled representations in a completely unsupervised condition. The work in [20] listed several methods for better training of GAN, and [21] explained the principles of GAN in terms of energy.

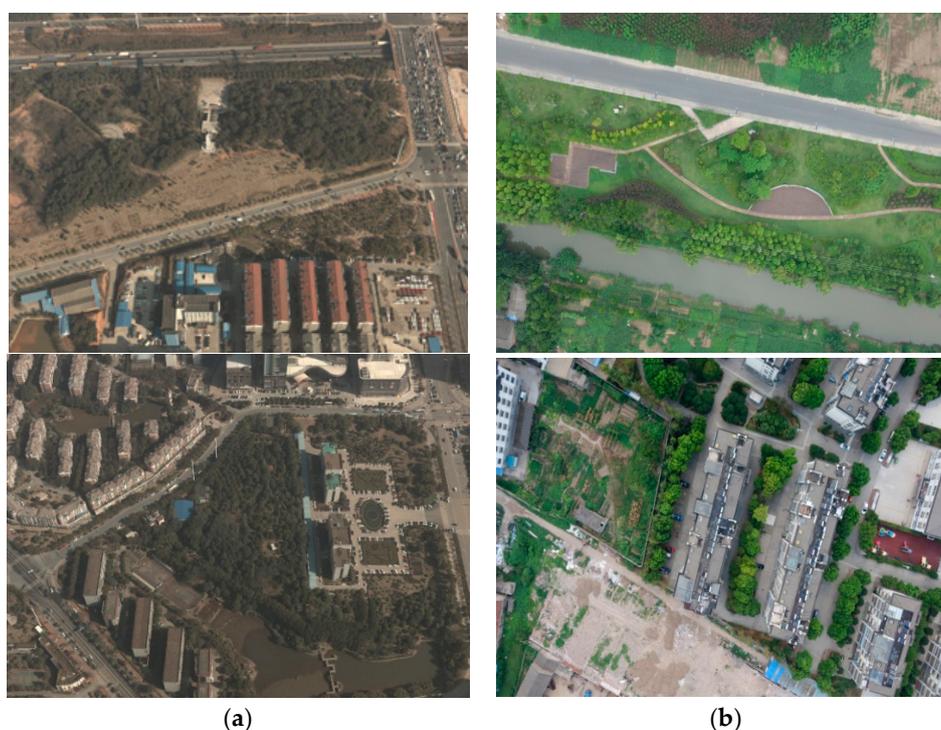
Cycle consistency: The use of cycle consistency as a way to regularize data has a long history. Cycle consistency consists of forward consistency and backward consistency. It has served as a trick for decades [22]. In [23–25], higher-order cycle consistency was used in different tasks, like human translation, 3D shape matching and depth estimation. Especially, in the work of [25,26], cycle consistency loss served as a method to train the neural network. This forms a strategy in CycleGAN.

### 3. The Proposed Method

Our method aims to turn vegetation green in winter oblique photography when the infrared band is not available. By using CycleGAN, this purpose can be achieved when winter oblique photography is transferred to summer oblique photography. Beside, CycleGAN's generator is modified by adjusting the kernel's size to reduce "checkerboard artifacts" in transferred photography.

#### 3.1. CycleGAN

Image-to-image translation might be difficult because paired images are hard to prepare for the model's training. For instance, in our case, it is hard to find winter oblique photography's corresponding to summer oblique photography. To handle this problem, CycleGAN is introduced to achieve an image-to-image translation. After training without paired images, the well-trained CycleGAN model can be employed to realize a mapping from winter oblique photography to summer oblique photography. Figure 1 shows an example of winter oblique photography and summer oblique photography.

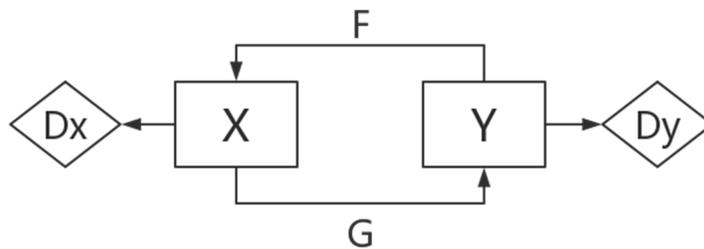


**Figure 1.** A comparison of vegetation in winter oblique photography and summer oblique photography. (a) is winter oblique photography, and (b) is summer oblique photography.

First and foremost, we define  $X$  as winter oblique photography's style domain and  $Y$  as summer oblique photography's style domain. These domains' distributions are denoted as  $x \sim p_{data}(x)$  and  $y \sim p_{data}(y)$ , respectively. Every oblique photography that pertains to winter is denoted as  $\{x_i\}_{i=1}^N$ , where  $x_i \in X$ . Every oblique photography that pertains to summer is denoted as  $\{y_j\}_{j=1}^M$ , where  $y_j \in Y$ . Besides, there are two mappings  $G: X \rightarrow Y$  and  $F: Y \rightarrow X$ . The first one translates winter oblique photography to summer oblique photography, and the second one translates summer oblique photography to winter oblique photography. The transferred winter oblique photography is denoted as  $\{G(x)\}$ . Likewise, the transferred summer oblique photography is denoted as  $\{F(y)\}$ .

In addition, there are also two discriminators  $D_X$  and  $D_Y$  in CycleGAN.  $D_X$  is responsible for distinguishing real winter photography  $\{x_i\}_{i=1}^N$  and fake winter photography (transferred summer photography)  $\{F(y)\}$ .  $D_Y$  is responsible for distinguishing real summer photography  $\{y_j\}_{j=1}^M$  and fake summer photography (transferred winter photography)  $\{G(x)\}$ .

The structure of CycleGAN is shown in Figure 2.



**Figure 2.** Structure of Cycle-consistence Adversarial Networks (CycleGAN).

CycleGAN consists of two mappings  $G: X \rightarrow Y$  and  $F: Y \rightarrow X$  and two discriminators  $D_X$  and  $D_Y$ , as shown in Figure 2.  $D_Y$  helps  $G: X \rightarrow Y$  better translates winter photography to summer photography, and the same goes for  $F: Y \rightarrow X$  and  $D_X$ .

### 3.2. Adversarial Loss

Adversarial loss [15] is used for both mapping  $G: X \rightarrow Y$  and  $F: Y \rightarrow X$ . For the mapping  $G: X \rightarrow Y$  and its discriminator  $D_Y$ , the loss is defined as:

$$L_{GAN}(G, D_Y, X, Y) = E_{y \sim p_{data}(y)}[\log D_Y(y)] + E_{x \sim p_{data}(x)}[\log(1 - D_Y(G(x)))] \quad (1)$$

In CycleGAN,  $G$  tries to generate photography  $\{G(x)\}$  that is close to the distribution of domain  $Y$ , while  $D_Y$  seeks to distinguish fake  $\{G(x)\}$  and real  $\{y_j\}_{j=1}^M$ .  $G$  attempts to minimize Equation (1), while  $D_Y$  tries to maximize it. Thus, the target of mapping  $G: X \rightarrow Y$  is written as:

$$\min_G \max_{D_Y} L_{GAN}(G, D_Y, X, Y) \quad (2)$$

Likewise, the target of mapping  $F: Y \rightarrow X$  can be written as:

$$\min_F \max_{D_X} L_{GAN}(F, D_X, Y, X) \quad (3)$$

### 3.3. Cycle Consistency Loss

The increase of cycle consistency loss [4] aims to ensure that learned mapping can map  $\{x_i\}_{i=1}^N$  to desired output  $\{y_j\}_{j=1}^M$ . For  $x$  from domain  $X$ , the image cycle translation should be able to bring it to its origin, which is written as  $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$ . Equally, there is  $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$ . Then, cycle consistency loss is defined as:

$$L_{cyc}(G, F) = E_{x \sim p_{data}(x)} [\|F(G(x)) - x\|_2] + E_{y \sim p_{data}(y)} [\|G(F(y)) - y\|_2] \quad (4)$$

where  $E_{x \sim p_{data}(x)} [\|F(G(x)) - x\|_2]$  and  $E_{y \sim p_{data}(y)} [\|G(F(y)) - y\|_2]$  are forward cycle loss and backward cycle loss, respectively.

### 3.4. Total Loss

The final objective is:

$$L(G, F, D_X, D_Y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \lambda L_{cyc}(G, F) \quad (5)$$

where  $\lambda$  determines the importance of cycle consistency loss. By experiments,  $\lambda$  is set as 11 here.

### 3.5. Elimination of Checkerboard Artifacts

We can find some small artifacts [27] called “checkerboard artifacts” if taking a close look at the image generated by the neural network. These artifacts make the generated image look poor in detail.

Figure 3a is the input photography, and Figure 3b is the output photography. Figure 3c,d shows the parts of Figure 3b where “checkerboard artifacts” are obvious.

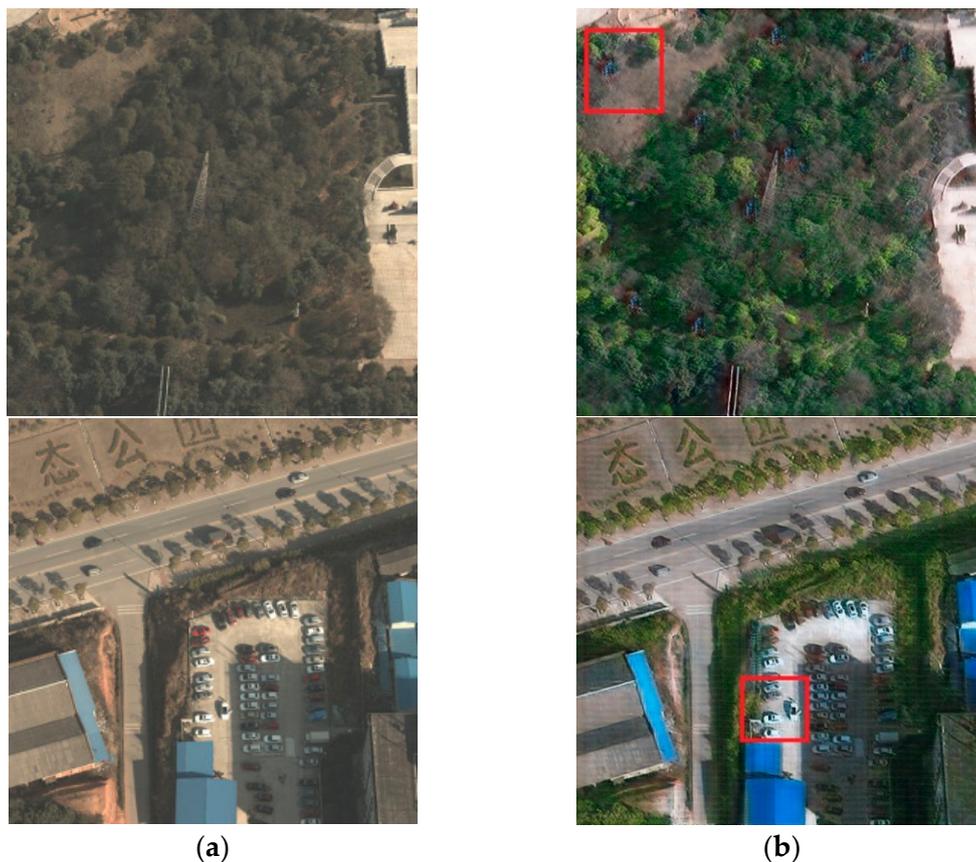


Figure 3. Cont.



**Figure 3.** An example of “checkerboard artifacts” in the generated photography: (a) input photography; (b) parts of artifacts in generated photography; (c,d) are enlarged parts of (b) where artifacts are obvious.

In [28], the cause of artifacts is clarified. In brief, the neural network often uses the deconvolution operation to build images from low resolution to high resolution. In this period, uneven overlap is created, which leads to the appearance of artifacts, especially in an image’s dark region.

To reduce artifacts in generated images, [29] came up with several solutions.

- (1) Renounce the use of the deconvolution operation. The method instead is: first, use up-sampling methods to build the image in the desired size; then, use the convolution operation to process the image. The choices of up-sampling methods are the nearest neighbor method and the bilinear method. The author of [29] recommended the nearest neighbor method.
- (2) Adjust the kernel’s size in the model’s generator. Adjust the kernel’s size to enable it to be split by stride. In CycleGAN’s generator, some layer’s kernel size is 3 with a stride of 2. Following the instruction of [29], these kernels’ size is modified to 4, so that it can be divided by 2.

After experiments, Solution (2) is adopted to reduce artifacts of Solution (1), which does not reduce artifacts obviously. More details can be seen in Section 4.

The whole process can be illustrated as follows.

---

**Process 1:** CycleGAN training process.

---

**Preparation:** Training images of winter  $X$  and training images of summer  $Y$ , mapping  $G$  with generated parameters  $\theta_G$  and mapping  $F$  with yielded parameters  $\theta_F$ , discriminator  $D_X$  with yielded parameters  $\theta_{D_X}$  and discriminator  $D_Y$  with yielded parameters  $\theta_{D_Y}$ .

**Input:**  $x \in X$  and  $y \in Y$

**Do**

Step1: update  $\theta_G, \theta_{D_Y}$  to minimize  $L_{GAN}(G, D_Y, X, Y)$  and  $E_{x \sim p_{data}(x)}[\|F(G(x)) - x\|_2]$ .

Step2: update  $\theta_F, \theta_{D_X}$  to minimize  $L_{GAN}(F, D_X, Y, X)$  and  $E_{y \sim p_{data}(y)}[\|F(G(y)) - y\|_2]$ .

**Until** convergence

---

## 4. Experimental Results

### 4.1. Dataset

Training data of winter oblique photography were acquired from Changsha, Hunan Province. Summer oblique photography was captured from Jingjiang, Jiangsu Province. It is noteworthy that all the photography should be at the same resolution. Otherwise, the training would be hard.

#### 4.2. Implementation Details

The generator used in CycleGAN was from [30]. For promotion, some kernels' size was changed, so that "checkerboard artifacts" could be reduced. The structure of the modified generator is defined in Table 1.

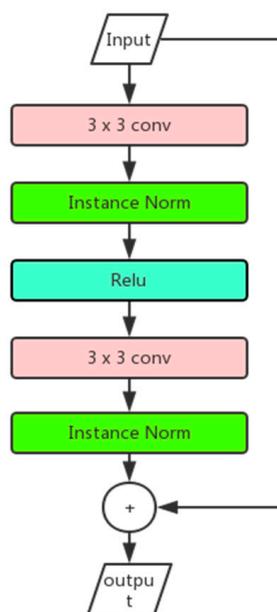
**Table 1.** Structure of our generator.

Layer
Input
Reflection padding ( $3 \times 3$ )
$64 \times 7 \times 7$ conv, Stride 1, Instance Norm, ReLU
$128 \times 4 \times 4$ conv, Stride 2, Instance Norm, ReLU
$256 \times 4 \times 4$ conv, Stride 2, Instance Norm, ReLU
Residual Block, 256 filter (9 blocks)
$128 \times 4 \times 4$ deconv, Stride 2, Instance Norm, ReLU
$64 \times 4 \times 4$ deconv, Stride 2, Instance Norm, ReLU
$3 \times 7 \times 7$ conv, stride 1
Tanh

*conv* means convolution, *deconv* means deconvolution, *Tanh* is activation function.

This generator has 4 layers with the kernel size of 4 and 9 residual blocks [31] under the instance norm [32].

The structure of the residual block is illustrated in Figure 4.



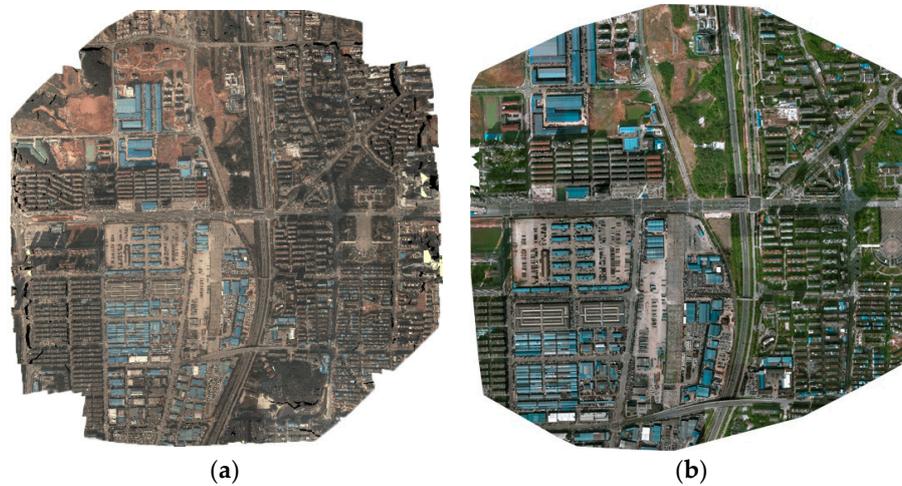
**Figure 4.** Structure of residual block.

Our experiments were performed on an NVIDIA Titan XP GPU. The operation system was Windows 7, and PyTorch served as the deep learning framework. It took us 3 days to finish the training. Because of the paper's typesetting, the images of some figures may be compressed. For the original images, see <https://github.com/carlblocking/results-of-my-first-sci-paper>.

#### 4.3. Results and Comparison

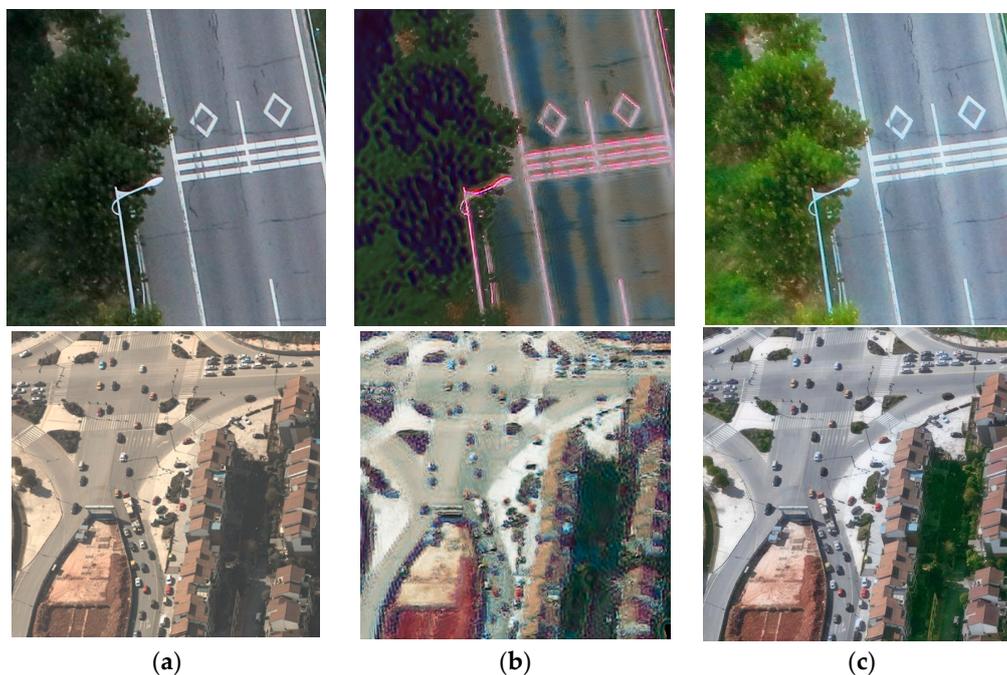
First and foremost, a comparison of the 3D model with good and bad visual quality is shown as mentioned in Section 1. Figure 5a is the result of the original photography, and Figure 5b is

from transferred photography. These images were captured from the 3D model's look-down angle. They were generated using the software smart3D. It is obvious that Figure 5b has better visual quality than 5a. The vegetation is greener, and the image is brighter.



**Figure 5.** Comparison of the 3D model of bad and good visual quality. (a) is the 3D model generated by the original photography and (b) is generated by the transferred photography.

Then, the solutions mentioned in Section 3 with the aim to reduce artifacts were tested. The results suggest that Solution (2) produces a satisfactory outcome, as shown in Figure 6.



**Figure 6.** Results from different artifact reduction solutions in Section 3. (a) is the input photography; (b) is the result of Solution (1); (c) is the result of Solution (2).

Figure 6a is the input photography. Figure 6b is Solution (1)'s output. Figure 6c is Solution (2)'s output. It is obvious that Figure 6c has better visual quality than Figure 6b because Figure 6b has an effect like an oil painting. Furthermore, artifacts are reduced in Figure 6c. Accordingly, Solution (2) served as an improvement to CycleGAN's generator here.

In CycleGAN's code [4], there are two different generators. One is from paper [30], which was applied in the realization of CycleGAN [4]. The other is U-net [33]. To compare different generators' performance, experiments based on our modified generator and these two generators were performed. First, these generators were tested on winter oblique photography taken in Hengyang, Hunan Province. The result is shown in Figure 7.

In Figure 7, inputs are the oblique photography of a building and a garden. In the result of the generator of [30], artifacts can be found at the edge of buildings and the garden's shadow. Artifacts were not obvious in results of the U-net generator and our modified generator. The difference may not be obvious due to the compression of image. To better show the results, SSIM (Structure Similarity index) is introduced here [34].



**Figure 7.** Results of different generators. From left to right are: input, results of the generator of [30], U-net generator and our modified generator.

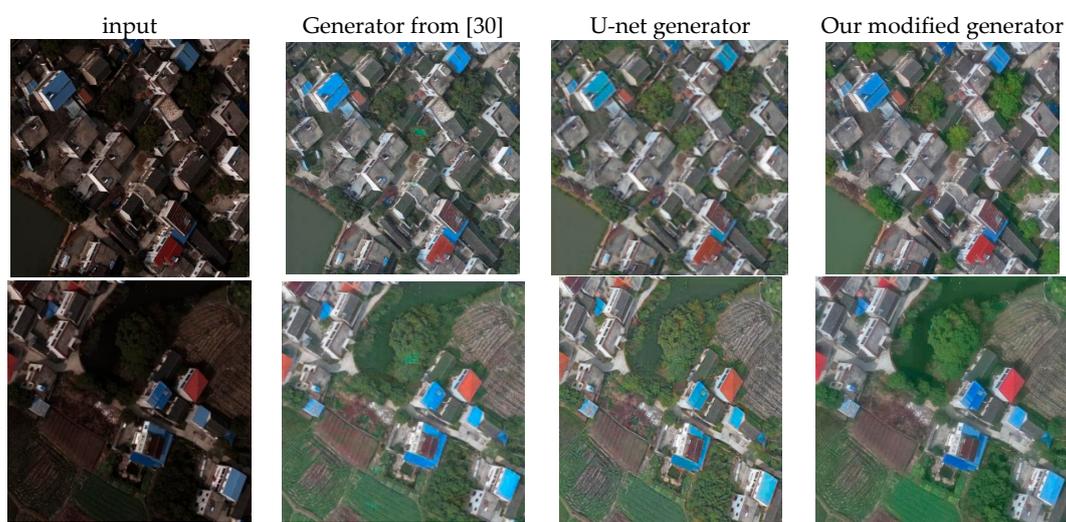
In Figure 7, photography from different generators is compared with the input photography in terms of SSIM. The results are listed in Table 2. The results of the generator of [30] achieved the lowest SSIM, suggesting the existence of artifacts. The U-net generator and our modified generator achieved higher SSIM, suggesting fewer artifacts. Yet, this suggests that our modified generator achieved lower SSIM than the U-net generator. This is because the results from our modified generator were greener than those of U-net generator's, which made the generated photography more different from the original input photography. To verify this, these generators were tested on another group of oblique photography.

**Table 2.** Comparison of SSIM from different generators.

Generator from [30]	U-Net Generator	Our Modified Generator
0.7346	0.9189	0.8978
0.6279	0.8379	0.8363

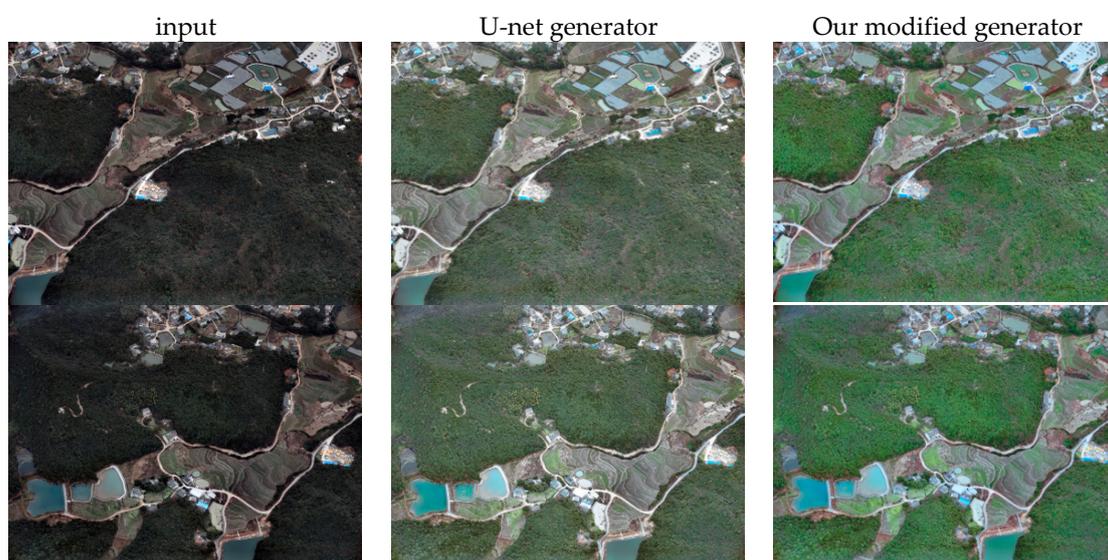
In Figure 8, the inputs are the oblique photography of the countryside in Qiqihar, Heilongjiang Province. It shows another type of oblique photography with poor visual quality. They were taken in bad weather, so that the brightness was low. In general, these generators have successfully improved these inputs' visual quality. Yet, in the results of the generator from [30], artifacts remained. In the results of the U-net generator and our modified generator, artifacts were reduced. Yet, U-net generator's result was less green in vegetation in comparison with the results from our modified generator. To better

show this difference, we tested the U-net generator and our modified generator on another group of photographs.



**Figure 8.** Results of different generators on another dataset. From left to right are the results from: input, the generator of [30], the U-net generator and our modified generator.

In Figure 9, the input oblique photography involves the mountains of Changsha, Hunan Province. The photography was taken in winter. The vegetation was not green. After transformation, the vegetation turned green in both generators' results. Yet, it is observed from Figure 9 that our modified model achieved better performance than U-net as the vegetation was greener.



**Figure 9.** Results of different generators tested on another photograph set. From left to right are the results from: input, the U-net generator and our modified generator.

To evaluate the performances of different generators, they were compared in terms of forward cycle loss and backward cycle loss.

Forward cycle loss was generally lower than backward cycle loss, as listed in Table 3. Our model achieved the lowest loss in both forward cycle loss and backward cycle loss. As defined in Section 3, forward cycle loss evaluates the performance of mapping  $F: Y \rightarrow X$ , which translates summer oblique

photography to winter oblique photography. Backward cycle loss evaluates the performance of mapping  $G: X \rightarrow Y$ , which translates winter oblique photography to summer oblique photography.

One assumption can explain why forward cycle loss was lower than backward cycle loss: that it might be easier for mapping  $F: Y \rightarrow X$  to degrade the visual quality of the photography. Conversely, it is difficult for mapping  $G: X \rightarrow Y$  to recover photography from low visual quality to high visual quality. Thus, mapping  $G: X \rightarrow Y$  is subject to higher loss than that of mapping  $F: Y \rightarrow X$ .

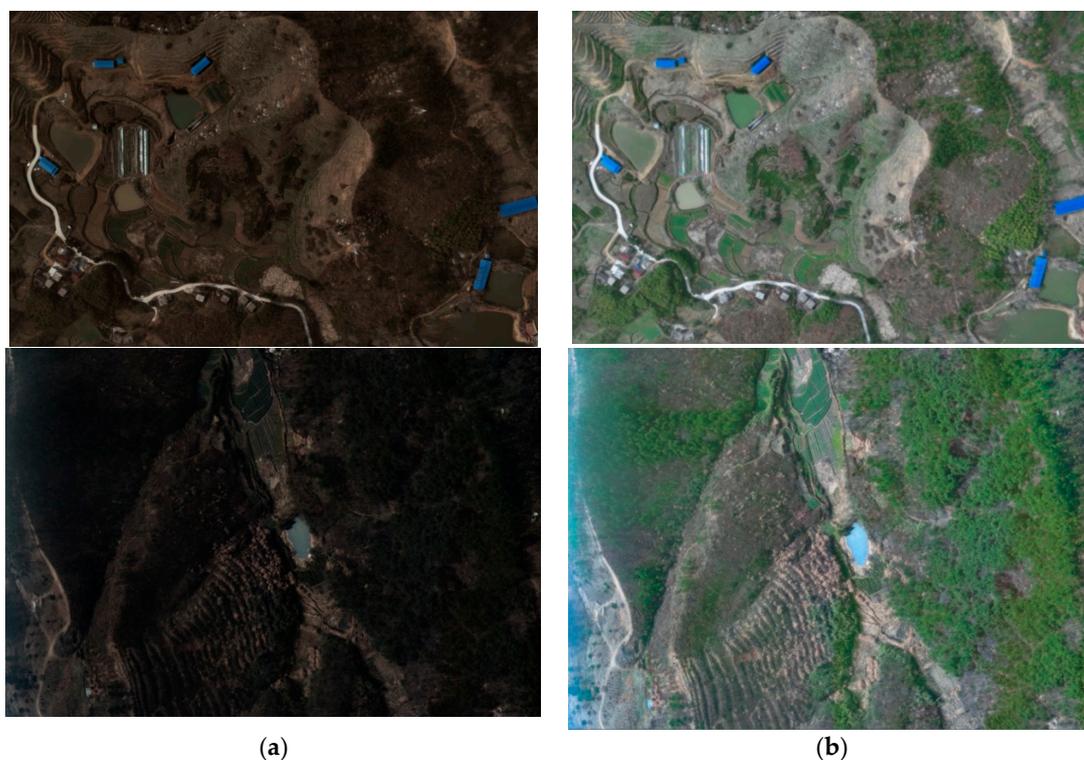
**Table 3.** Comparison of forward cycle loss and backward cycle loss with different generators.

Generator	Forward Cycle Loss	Backward Cycle Loss
Generator from [30]	0.05	0.49
U-net generator	0.11	0.55
Our modified generator	0.015	0.32

Besides, in backward cycle loss, the generator from [30] obtains the highest value. This could be attributed to the existence of artifacts. Furthermore, our modified generator's loss value was lower than that of the U-net generator. This was probably because our modified generator produced better results than U-net, as the vegetation was greener. Yet, none of these assumptions have theoretical proof. Hence, deeper research is required in the future.

Finally, an experiment of using transferred oblique photography to build the 3D model was performed.

First, the original input photography and its corresponding transferred results are shown in Figure 10. In Figure 10a is the oblique photography taken in bad weather, and Figure 10b is the transferred result of this photograph. The original input photography was not bright, and its vegetation was not green. Common methods cannot turn the vegetation green when the infrared band is not available. Using our well-trained model, the photography can be improved significantly.



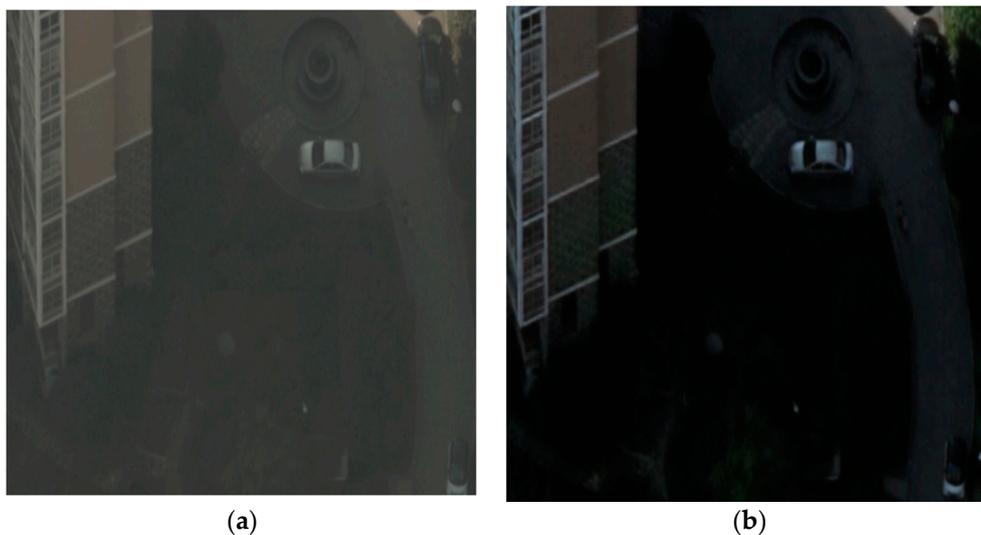
**Figure 10.** (a) is the input, and (b) is the output.

Then, the transferred photography was employed to generate a 3D model. It is clear from Figure 11 that the generated 3D model had a good visual effect.



**Figure 11.** 3D model generated by transferred photography from Figure 10b.

Yet, in very rare cases, our model made the building slightly green in darkness, as shown in Figure 12. We are now finding the reasons for and solutions to this problem. In Figure 12, the wall becomes slightly green in darkness.



**Figure 12.** (a) Input; (b) output.

## 5. Conclusions and Discussion

In this paper, a new vegetation greening method using CycleGAN for oblique photography was proposed. Unlike common methods, the infrared band is not required to help turn the vegetation green. By adjusting the kernel size of CycleGAN's generator, "checkerboard artifacts" are reduced in the final result.

Yet, in very few cases, buildings in darkness become green. Deeper research to handle these problems is needed in the future.

**Author Contributions:** Z.S. conceived of and design the experiments. He also provided the data and hardware. X.X. performed the experiments and analysis of the data. He also wrote the paper. C.W. and N.N.X. proofread the paper. Y.W. obtained the findings.

**Acknowledgments:** This research was supported by Shanghai Science and Technology Innovation Action Plan Project (16111107502, 17511107203) and Shanghai key lab of modern optical system.

**Conflicts of Interest:** The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analysis or interpretation of data; in the writing of the manuscript; nor in the decision to publish the result.

## References

- Wei, W.; Wenwen, H.; Jiao, Z. Pictometry Oblique Photography Technique and its application in 3D City Modeling. *Geomat. Spat. Inf. Technol.* **2011**, *34*, 181–183.
- Lou, B.; Geng, Z.X.; Wei, X.F.; Wang, L.F.; Li, Y.-J. Texture mapping of 3D city model based on Pictometry oblique image. *Eng. Surv. Mapp.* **2013**, *22*, 70–74. [[CrossRef](#)]
- Wang, L.; Sousa, W.P.; Peng, G.; Biging, G.S. Comparison of IKONOS and QuickBird images for mapping mangrove species on the Caribbean coast of Panama. *Remote Sens. Environ.* **2004**, *91*, 432–440. [[CrossRef](#)]
- Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2242–2251.
- Hertzmann, A.; Jacobs, C.E.; Oliver, N.; Curless, B.; Salesin, D.H. Image Analogies. In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'01), Los Angeles, CA, USA, 12–17 August 2001; pp. 327–340.
- Zhang, R.; Isola, P.; Efros, A.A. Colorful Image Colorization. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 649–666.
- Wang, X.; Gupta, A. Generative Image Modeling Using Style and Structure Adversarial Networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 318–335.
- Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Honolulu, HI, USA, 21–26 July 2017; pp. 5967–5976.
- Johnson, J.; Alahi, A.; Li, F.F. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 694–711.
- Laffont, P.Y.; Ren, Z.; Tao, X.; Qian, C.; Hays, J. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Trans. Graph.* **2014**, *33*, 1–11. [[CrossRef](#)]
- Rosales, R.; Achan, K.; Frey, B. Unsupervised Image Translation. In Proceedings of the IEEE International Conference on Computer Vision, Beijing, China, 13–16 October 2003; Volume 471, pp. 472–478.
- Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. *arXiv* **2013**, arXiv:1312.6114.
- Liu, M.Y.; Breuel, T.; Kautz, J. Unsupervised Image-to-Image Translation Networks. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 8 December 2017.
- Liu, M.Y.; Tuzel, O. Coupled Generative Adversarial Networks. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016.
- Goodfellow, I.J.; Pougetabadié, J.; Mirza, M.; Xu, B.; Wardefarley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *Adv. Neural Inf. Process. Syst.* **2014**, *3*, 2672–2680.

16. Denton, E.; Chintala, S.; Szlam, A.; Fergus, R. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. In Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015; pp. 1486–1494.
17. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv* **2015**, arXiv:1511.06434.
18. Im, D.J.; Kim, C.D.; Jiang, H.; Memisevic, R. Generating images with recurrent adversarial networks. *arXiv* **2016**, arXiv:1602.05110.
19. Chen, X.; Duan, Y.; Houthoofd, R.; Schulman, J.; Sutskever, I.; Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016.
20. Arjovsky, M.; Bottou, L. Towards Principled Methods for Training Generative Adversarial Networks. *arXiv* **2017**, arXiv:1701.04862.
21. Zhao, J.; Mathieu, M.; Lecun, Y. Energy-based Generative Adversarial Network. *arXiv* **2016**, arXiv:1609.03126.
22. Sundaram, N.; Brox, T.; Keutzer, K. *Dense Point Trajectories by GPU-Accelerated Large Displacement Optical Flow*; Springer: London, UK, 2010; pp. 438–451.
23. Brislin, R.W. Back-translation for cross-cultural research. *J. Cross Cult. Psychol.* **1970**, *1*, 185–216. [[CrossRef](#)]
24. Huang, Q.X.; Guibas, L. Consistent Shape Maps via Semidefinite Programming. *Comput. Graph. Forum* **2013**, *32*, 177–186. [[CrossRef](#)]
25. Godard, C.; Aodha, O.M.; Brostow, G.J. Unsupervised Monocular Depth Estimation with Left-Right Consistency. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
26. Zhou, T.; Krähenbühl, P.; Aubry, M.; Huang, Q.; Efros, A.A. Learning Dense Correspondence via 3D-Guided Cycle Consistency. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 117–126.
27. Shi, W.; Caballero, J.; Theis, L.; Huszar, F.; Aitken, A.; Ledig, C.; Wang, Z. Is the deconvolution layer the same as a convolutional layer? *arXiv* **2016**, arXiv:1609.07009.
28. Gauthier, J. Conditional Generative Adversarial Nets For Convolutional Face Generation. Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter Semester. 2014. Available online: <http://www.foldl.me/uploads/2015/conditional-gans-face-generation/paper.pdf> (accessed on 20 July 2018).
29. Odena, A.; Dumoulin, V.; Olah, C. Deconvolution and Checkerboard Artifacts. *Distill* **2016**, *1*. [[CrossRef](#)]
30. Dosovitskiy, A.; Brox, T. Generating Images with Perceptual Similarity Metrics based on Deep Networks. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
32. Ulyanov, D.; Vedaldi, A.; Lempitsky, V. Instance Normalization: The Missing Ingredient for Fast Stylization. *arXiv* **2016**, arXiv:1607.08022.
33. Dan, C.C.; Giusti, A.; Gambardella, L.M.; Schmidhuber. Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 2852–2860.
34. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]

