mathematics

MDPI

Article

Unsupervised Image Translation Using Multi-Scale Residual GAN

Yifei Zhang *, Weipeng Li, Daling Wang and Shi Feng

School of Computer Science and Engineering, Northeastern University, Shenyang 110169, China
* Correspondence: zhangyifei@cse.neu.edu.cn; Tel.: +86-24-8368-7776

Abstract: Image translation is a classic problem of image processing and computer vision for transforming an image from one domain to another by learning the mapping between an input image and an output image. A novel Multi-scale Residual Generative Adversarial Network (MRGAN) based on unsupervised learning is proposed in this paper for transforming images between different domains using unpaired data. In the model, a dual generater architecture is used to eliminate the dependence on paired training samples and introduce a multi-scale layered residual network in generators for reducing semantic loss of images in the process of encoding. The Wasserstein GAN architecture with gradient penalty (WGAN-GP) is employed in the discriminator to optimize the training process and speed up the network convergence. Comparative experiments on several image translation tasks over style transfers and object migrations show that the proposed MRGAN outperforms strong baseline models by large margins.

Keywords: image translation; generative adversarial network; unsupervised learning; object migration; multi-scale residual network

MSC: 68Q32; 68T07; 68T45

Citation: Zhang, Y.; Li, W.; Wang, D.; Feng, S. Unsupervised Image Translation Using Multi-Scale Residual GAN. *Mathematics* **2022**, *10*, 4347. https://doi.org/10.3390/math10224347

Academic Editors: Vladimir V. Arlazarov and Konstantin Bulatov

Received: 17 October 2022
Accepted: 14 November 2022
Published: 19 November 2022

1. Introduction

Image-to-image translation is a longstanding topic in some image processing and computer vision tasks, e.g., image enhancing, colorization, semantic segmentation, artistic stylization, which can be seen as a translation from an input image to an output image. Due to the use of deep learning models in recent years, the image translation tasks have been constantly explored in research fields from style transfer to scene changing, and then to object migration, etc. [1–10]. Meanwhile, there are more and more in-depth applications in our daily life from the former Prisma [11], Ostagram [12] and Deep Forger [13] to the latest Versa [14]. The learning methods have been improved from pixel-based iterations to generative models for enhancing networks. In short, great progress has been made in all aspects of image-to-image translation.

At present, GANs [15] and conditional GANs (cGANs) [16] are mainly used as image translation models to generate the target images [4,10,17–21]. Considering the forms of input images, these models can be divided into supervised learning and unsupervised learning. In supervised models, a large number of input-to-output image pairs are need for training networks. Although the networks are learned faster, it could be difficult and costly to find enough labeled pairs in real life, even the desired outputs cannot be well-defined in some tasks, such as object migrations and artistic stylization. Therefore, some unsupervised models are proposed to address this problem, which only need images from two domains with different styles, instead of paired data. The models can automatically learn the semantic difference between two domains and transfer an image from the source domain into the analog image looking like from the target domain. For example, in [21], a horse in an image can be transformed into a zebra with the same posture and behavior as the horse, where the original background still remains. In this way, unsupervised translation models not only do not require paired images for training, but can even produce good results when

the image sizes of two domains differ by one or more orders of magnitude, as long as the image style of each domain is consistent.

Although unsupervised models can learn the difference between two image domains and capture domain-specific characteristics, some important semantic information is often lost due to the lack of contexts from one-to-one image pairs. For example, if missing the object information of a given input, a horse grazing on green grass is likely to be converted into a zebra munching on withered grass since the background of images carries on adversarial learning together. In addition, if the structure is not stable, it may also cause color leakage in the target output. How to preserve the same context and structure with its input image in a target output besides transformed objects and styles is still an issue of unsupervised learning.

In this paper, we propose an unsupervised image translation model, MRGAN, with two generators and one discriminator based on GAN. For the issues of color leakage and structure instability caused by the lack of target image guidance, we design a multi-scale layered structure in the generators, which adopts a bilinear interpolation algorithm to capture image layered features and integrate them into every-level residual unit for retaining enough semantic and structure information. In addition, the WGAN-GP network [22] is employed in the discriminator, combining with the randomly cropped image patches as inputs to optimize the training process and enhance the robustness of the model. The key contributions of our work are summarized in the following: (1) The multi-scale layered structure with residual units is used in the generators to eliminate image semantic loss in the process of encoding. (2) In the discriminator, inspired by [23,24], a WGAN-GP architecture is employed for gradient penalty instead of weight pruning to preserve stability in network training. (3) We use an improved objective function combining the reconstructed loss and the adversarial loss in MRGAN to further optimize the training process and speed up network convergence. Extensive experiments show that the approach can successfully eliminate the dependence of paired images and retain enough semantic details in the generated images, which can meet the needs of a wide range of applications including object migration, season transfer, style conversion, and so on.

2. Related Work

This section presents the relevant works including GANs and image-to-image translation, which will highlight the background and current research state in this area.

2.1. Generative Adversarial Networks (GANs)

GAN was first proposed in 2014 by Ian J. Goodfellow et al. [15]. As one of the most valuable generative models on deep learning, GANs have received considerable attention in the field of computer vision for image generating [9,21,25–29]. GANs are one type of models that can generate more similar samples by learning the probability distributions of training set, which aim to generate samples through a zero-sum game between a discriminator, trying to discriminate between real and fake data, and a generator, seeking to generate data that resemble the real ones. In recent years, many studies have focused on optimizing the training stability of GANs [22,30] or utilizing additional information to release the potential of GANs in different domains [31]. cGAN is a typical GAN model that uses conditioned information to guide the image generation [16,32], which has been successfully utilized for various applications on image translations [17,18].

2.2. Image-to-Image Translation

The idea of image translations traces back to Image Analogies [33], which proposed a nonparametric texture model on a single input–output image pair. Recent works in the field of image-to-image translation have achieved impressive results [5,6,10,18,19,34]. Pix2pix is the first general framework using a cGAN for image-to-image translations [4], followed by its enhanced version, pix2pixHD [17]. Based on pix2pix, BicycleGAN is proposed to achieve multi-domain image-to-image translation using paired images [6]. Similar models have

been employed on various applications for generating images such as from sketches [18,35] and from attributes and semantic layouts [36].

The need for pairwise training samples is one of the main disadvantages of supervised image-to-image translations. However, this fatal problem has been well solved in unsupervised image translations, which do not need paired data, as long as the style of images remains consistent in the same set. CycleGAN [21], DualGAN [19], DiscoGAN [20] are typical unpaired image-to-image translation models which work by introducing the cycle consistency constraint cross two domains. In addition to the definitions of loss functions, these three models have many similarities in structures and training methods. For instance, CycleGAN has achieved good results in the object migration task between horses and zebras, where only the target objects are migrated while the rest of the images are retained. Nevertheless, there still exists a training unbalance on CycleGAN that leads to mode collapse, where the background colors of images change as the migrated objects change although the original positions and structures are maintained.

Multi-domain image-to-image translations recently have been extensively studied for image translation tasks. StarGAN [34] is proposed to transform images among multiple domains simultaneously by sharing one generator–discriminator pair with the aid of an auxiliary classifier [37]. As a result of sharing a common mapping for different tasks of transforming and inverting, it is harder to optimize the generator for good generalization ability on both tasks. For this, G^2GAN [10] uses two generators to respectively transform and reconstruct images between the two domains. For optimizing the training process, two task-specific generators in G^2GAN share different-level parameters and perform strict cycle-consistency loss on feature channels. Therefore, G^2GAN is particularly appropriate for conversing between different attributes of similar images, such as facial attribute transfer and expression convertion on face images. Similarly, more extra information can also be used in diverse multi-generator/multi-discriminator models for various image translation tasks, such as artistic portrait styles [38], text and object descriptions [31,36,39,40], semantic or sketch labels [41].

In this paper, a novel unpaired image translation model, MRGAN, is designed with dual generators and one discriminator, which solves the semantic information retention problem of unsupervised translations. A Multi-Scale layered structure is employed in the generator networks, where a bilinear interpolation algorithm is used to adjust the size of images instead of down-sampling in the network for retaining more image information. The discriminator carries out a full convolution PatchGANs [4,25,42], where WGAN-GP method [23] is used to constraint the training process of networks and accelerate the convergence of the model by combining the reconstructed loss and the adversarial loss.

3. Proposed Model

Our proposed MRGAN performs the image-to-image translation using unpaired training data. Suppose A and B represent two image fields converted to each other and no pairings need to exist between these two domains, our model learns a function ϕ that maps from A to B using training data. As shown in Figure 1, the model consists of two generators and one discriminator—a generator G_1 transforming images from domain A into the images with the style of domain B, and an inverse generator G_2 transforming images from domain B into the images with the style of domain A, and a discriminator D discriminating the generated images from the real images with different domain labels. Meanwhile, two generators can be used to reconstruct images from the output domain into the original domain. The operations of two generators can be regarded as being onto mappings between two image domains, where an image from each domain can be rendered into the style of the other one.

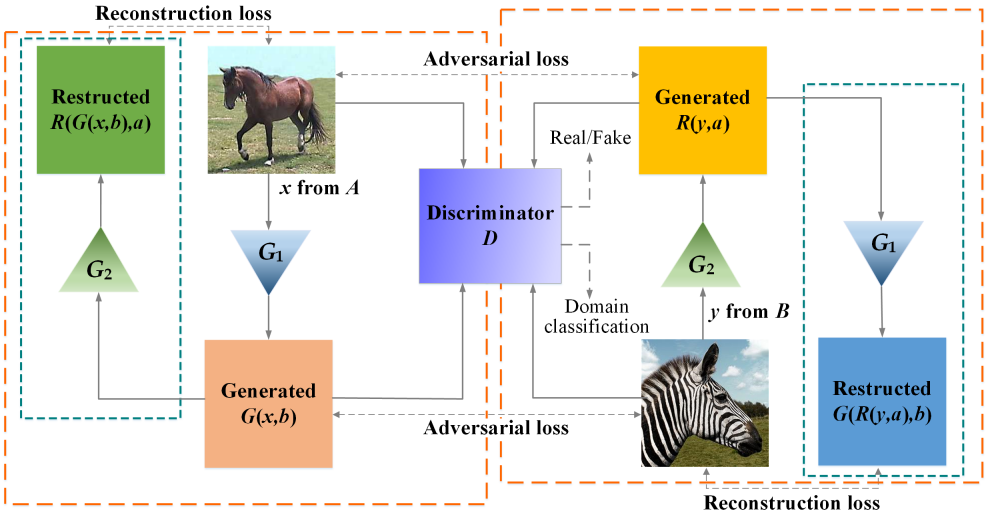

Figure 1. MRGAN contains two generators for transferring two styles of images each other. Generator G_1 is used for learning an mapping from domain A to domain B. Similarly, Generator G_2 is used for learning an mapping from domain B to domain A. The discriminator aims to distinguish generated images from real images of two domains.

3.1. Generator

The role of a generator is to map images from one domain into another one so that the style of generated images is sufficiently similar to that from the target domain and enough to confuse the discriminator. What is more, a generator is used to detect the distribution of the original domain and then match it to the consistent distribution from the target domain.

Figure 2 shows a generator network. A multi-scale layered structure is proposed in this paper to preserve the details of images as much as possible in the generator, of which a bilinear interpolation method is used to obtain three scales of images for training the generator network. In our work, an RGB image with 256 × 256 pixels is inputted into the network, and after two bilinear interpolations, the images with 128 × 128 and 64 × 64 are obtained.

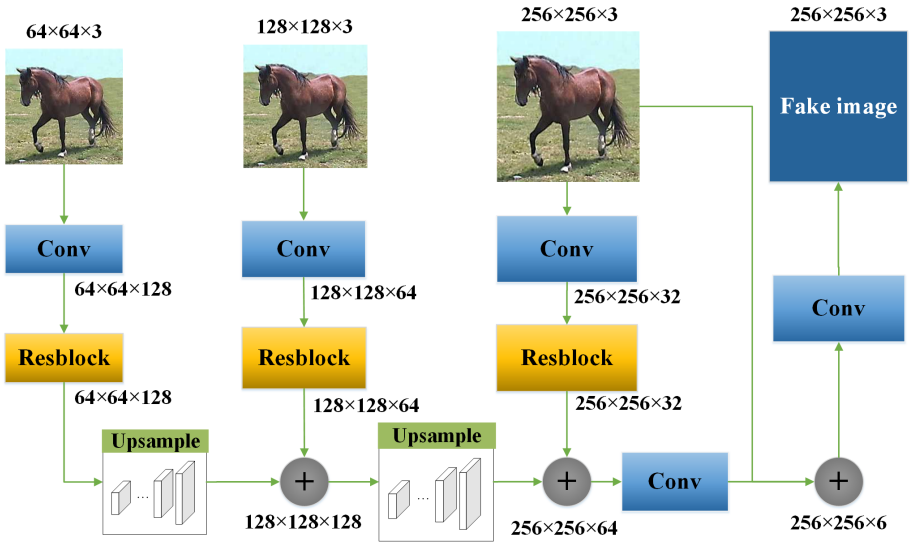

Figure 2. The generator network uses a multi-scale layered residual structure to preserve the details of images as much as possible, where three input scales of images are obtained by a bilinear interpolation method, combined with residual operators and convolutions for training the network.

One of the greatest advantages of multi-scale structure is that the convolution kernel can obtain different receptive fields, when images with multiple scales pass through convolution layers with a fixed size convolution kernel. It can be known from the Convolutional Neural

Network (CNN) principle that the smaller the image scale is, the larger the perception field obtained by the convolution kernel is, and the higher order of information in the image can be extracted. On the contrary, the larger the image scale is, the smaller the receptive field obtained by the convolution kernel is, and it is easier to find the subtle features in the image. Therefore, we directly expand the receptive fields of convolution kernel by reducing the image sizes to replace down-sampling operations, and thus can effectively eliminate information loss and simplify the network structure. After images with different scales are entered into the generator, the following steps are performed for generating images:

- First, a convolution processing is executed on each channel for adjusting the number of channel dimensions;
- Second, residual units are carried out to output the feature maps with the corresponding sizes;
- Third, through layer-by-layer up-samplings, the smaller feature maps are spliced with the larger ones;
- Fourth, after a series of up-sampling operations, the feature map with the same size as the original image is obtained;
- Last, the obtained feature map is spliced with the original image, and through the last convolutional layer, the final stylized image is obtained.

It deserves to be mentioned that in our generator networks, the up-sampling operations are completed by first magnifying feature maps using a bilinear interpolation and then adjusting channel dimensions through a transposed convolutional layer.

3.2. Discriminator

In the MRGAN model, the discriminator D is designed based on PatchGAN [4] and it contains two branches, where one classification branch is to classify images into the correct image domains, and another discrimination branch is to discriminate generated images from real ones. An improved Full Convolution Network (FCN) is shared by the two branches. The output of the convolution layer is directly regarded as the return value of the network for calculating the adversarial loss, while the active layer is used for the binary classification. We use 70×70 PatchGANs in the discriminator network, which accepts image inputs randomly cut to reduce computational cost of FCN on the premise that the output is credible.

In the discriminator network, the convolution layers are constructed only using 2-step and 1-step sizes of the convolution kernel. Actual operations on convolution layers can be regarded as down-sampling, where the dimensions of feature maps can be reduced by steps as well as the structure features and the color information of images can be extracted. The discriminator network with this structure actually can convert the classification problem into the fitting problem of the data distribution. That is, it can fit the image distribution from one domain into another one, so that the data distribution is as close as possible to the target one after adversarial training.

3.3. Loss Functions

Loss functions are also called objective functions. The training process of the network is to optimize objective functions through iterative learning. Our goal is to learning two mapping functions $G : A \rightarrow B$ and $R : B \rightarrow A$ between two domains, A and B, as illustrated in Figure 1. Given training image $x_i \in A$ ($i = 1, \ldots, M$) and $y_j \in B$ ($j = 1, \ldots, N$), where M and N are the example numbers of A and B. There are two generators and one discriminator in MRGAN. Generator G_1 generates image $G(x_i, b)$ from image x_i and G_2 reconstructs image $R(G(x_i, b), a)$ from generated $G(x_i, b)$, where a and b, respectively, are labels of target domain A and B. In the same way, Generator G_2 generates $R(y_i, a)$ from y_i and G_1 reconstructs image $G(R(y_j, a), b)$ from generated $R(y_j, a)$. Discriminator D is to distinguish $G(x_i, b)$ and $R(y_i, a)$ from the real images guided by classifier label a and b. Therefore, two types of losses are used for generating training and adversarial training: adversarial loss for distinguishing the generated samples from the real ones as much as possible and reconstructed loss for maintaining the generative consistency of mapping G and R.

3.3.1. Adversarial Loss

Here, we used the adversarial loss of WGAN-GP with RMSProp optimization [23]. Since two generators have the same structure and only generate images in opposite directions, we use a unified function $M(x, l)$ to represent two inverse mapping G and R, where x is an input image, and $l \in \{a, b\}$ is the label of the target domain. The discriminant function D_d of discriminator D tries to distinguish generated $M(x, l)$ from real image y. The adversarial loss of D is optimized as

$$max_D L(M, D_d) = E_{y \sim P(y)}[D_d(l|y)] - E_{x \sim P(x)}[D_d(l|M(x, l))] + \lambda \times gp \quad (1)$$

where $P(\cdot)$ is the intrinsic distribution of an image, and $D_d(l|y)$ and $D_d(l|M(x, l))$, respectively, are the predicted softmax probabilities of y and $M(x, l)$ by discriminator D over label l. The gradient penalty item $\lambda \times gp$ over the discriminator is used to speed up the convergence and optimize the objective, as λ is a learning rate of the gradient penalty. Gradient penalty is defined as

$$gp = E_{y'' \sim P_{y''}}[(\| \bigtriangledown_{y''} (y'')\|_2 - 1)^2] \quad (2)$$

where $\| \cdot \|_2$ denotes the second moment norm of the gradient, y'' is a linear combination of the real sample y and the generated sample $M(x, l)$, i.e.,

$$y'' = y + \beta(M(x, l) - y) \quad (3)$$

where β is a real number sampled from the uniform distribution [0, 1].

3.3.2. Reconstructed Loss

The reconstructed loss aims to ensure the generative consistency between A and B. Intuitively, each image x from domain A can be mapped into $G(x, b)$ following the distribution of domain B, and then $G(x, b)$ can be reconstructed back to domain A by $R(G(x, b), a)$ too. From Figure 1, it can be seen that two generators should satisfy the generative consistency respectively by mapping G and R. Additionally, an original image and its reconstructed image should be as similar as possible. Here, we encourage this consistency using the reconstructed loss:

$$rl = \|R(G(x, b), a) - x\|_1 + \|G(R(y, a), b) - y\|_1 \quad (4)$$

where $\|x - y\|_1$ denotes the L_1 distance of x and y. The generator G_1 aims to generate image $G(x, b)$ looking like an image from B. Considering that the reconstructed loss is used to preserve the generative consistency, we optimize the objective of G_1 by

$$max_G L = -E_{x \sim P_A(x)}[D_d(b|G(x, b))] + \delta \times rl \quad (5)$$

where δ is a real coefficient for reconstructed loss.

Similarly, we can define the objective of G_2 and optimize it, i.e., $max_R L$. Since we have defined the reconstructed loss and the adversarial loss for the generators and the discriminator, respectively, MRGAN can optimize these loss functions along at the same time until the final convergence.

4. Experiments

4.1. Datasets and Setup

The datasets used in our work include three open image sets, *horse2zebra*, *summer2winter* and *maps*, from [4]. As shown in Figure 3, each image set contains images from two domains for transforming each other. *horse2zebra* is used for object migration between horses and zebras in two domain images. *summer2winter* includes landscape photos of summer and winter sceneries for season conversion. *maps* consists of two types of map images, Google maps and aerial photos, for map transformation between plane

maps and satellite aerial photographs. In addition, we also crawled some images online to extend the datasets so that each domain contains at least 1000 images for training and testing. All images in the datasets are RGB images with a 256 × 256 size.

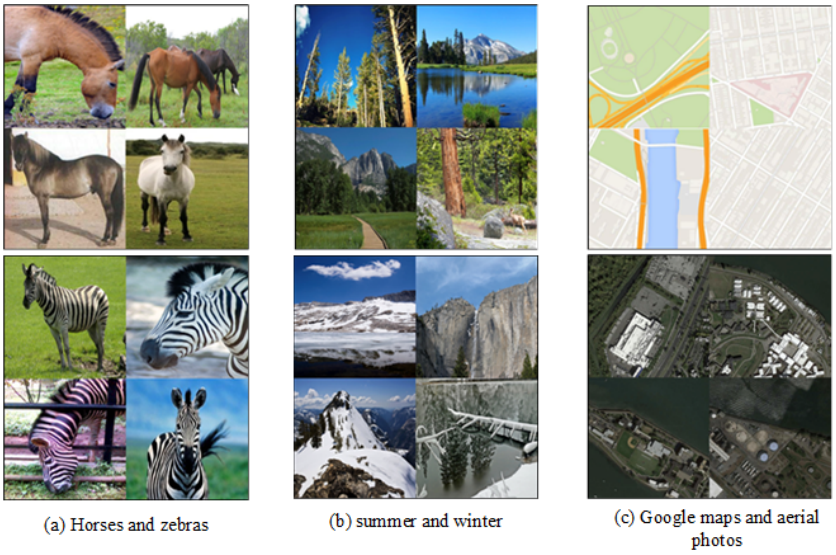

Figure 3. The datasets include *horse2zebra*, *summer2winter* and *maps*, where each dataset contains the images respective from two domains. (**a**) shows the examples of horses and zebras from *horse2zebra*, (**b**) shows seasonal landscape photos from *summer2winter*, and (**c**) shows Google maps and aerial photos from *maps*.

The evaluations for the model refer to the comparison of image translation results and the analysis of time performance. Since there are no quantitative metrics for unsupervised image translation tasks owning to missing target images in datasets, we can only evaluate experimental results through the visual presentations of result images. Considering that people's visual comments are subjective to some extent, in order to ensure the objective and effective experimental analysis, we conducted user studies to validate our work, which carried out a manual evaluation for the image translated results using different models on different data sets. We designed the different visual metrics for the three tasks according on the difference of image styles, respectively. Then, we provided the participants with the shuffled result images of three tasks generated, respectively, by MRGAN and two baseline models, and a scoring table including scoring items for different metrics on different styles. Three score ratings are set from 1 to 3, according to the satisfactory degree on each metric, respectively. For received available feedback results, we computed the average scores for each metric on the different tasks, and obtained the overall scores of different models according to all metrics. Then, we analyzed the results and evaluated the effects of different models according to their average scores and overall scores. Training speed (S_{train}) and generation speed (S_{gen}) are used as the evaluation metrics for measuring the time performance in our work. S_{train} is the average time consumption for one iteration, S_{gen} expresses the the average time consumption for generating one image. The definitions of the performance metrics are expressed below:

$$S_{train} = \frac{t_{train}}{iters} \tag{6}$$

$$S_{gen} = \frac{t_{gen}}{counts} \tag{7}$$

where t_{train} denotes the total time to update the network weights, $iters$ is the iterations, t_{gen} denotes the total time to generate images, and $counts$ is the number of images generated.

Our experiments were conducted on GPU: Nvidia GeForce GTX 1060 with 3 GB RAM, and the proposed and reproduced methods were implemented using Python program language.

4.2. Model Settings

Since there are two independent generators in MRGAN, we will train these two networks at the same time. Different from CycleGAN [4] and WGAN-GP [23] models, each generator performs one loop, and then the discriminator executes 5 loops using the same training data for training a single generator network. In training, each input of the generator is only one image, and inputs of the discriminator network are image blocks with a size of 70×70 pixels randomly cut from an input.

4.3. Training Method and Strategy

As the performance of a model is affected by the quality of training data, scattered data may lead to too slow convergence of networks and even to severe pattern collapse. Therefore, data first need to be standardized before being inputted into networks. In our work, the pixel values of RGB images are first mapped to $[-1, 1]$ to prevent the occurrence of gradient disappearance on training.

4.3.1. Group Normalization

In the training of multi-layer networks, the input data on each layer need to be normalized to preserve the uniform data distribution. Although Batch normalization [43] is shown to be effective, it is not applicable for our work because it is equivalent to instance normalization [44] in case where only one image is processed in the network at a time as a batch. The group normalization method [45] is used in our model, where the data first are grouped by channel dimensions, and then the data of each group are normalized for avoiding the single batch dimension. The channel data are divided into 64 groups in our networks, as well as the training being twice as fast as using batch normalization without gradient disappearance and mode collapse.

4.3.2. Weight Initialization and Activation Functions

Weight initialization plays an important role in training networks. If the initial weights of networks are too small, the gradient signals may be weakened layer by layer until disappear. Conversely, if the weights are initialized too large, the signals will be amplified between two layers, resulting in gradient explosion. For that, we use the Xavier initializer [46] to initialize the network weights, which tunes the initial weighs according to the distribution of input data. In order to convey gradient information better, we adopt Leaky ReLU as the activation function in the down-sampling process, while we use ReLU in up-sampling.

4.3.3. Optimization and Learning Rate

MRGAN adopts WGAN-GP architecture to replacing pruning by gradient punishment on the weights of discriminator, where the Adam optimization algorithm based on momentum [47] is used in order to make the training more stable. Learning rate decay strategy is applied in MRGAN for accelerating the model convergence. The decay mode of learning rate α is

$$\alpha = \alpha_0 - \eta \times epochs \tag{8}$$

where α_0 is initial learning rate and set as 2×10^{-4}, η is decay factor, and *epochs* represents iteration numbers. Here, the decrement item $\eta \times epochs$ is always required to be less than α_0, since it is verified through experiments that the decrement of the learning rate is about 2×10^{-6} at a time by updating the weights with all the data, in order to make the decay frequency (i.e., iterations) less than 100, we set η as 2×10^{-6}. Obviously, if the learning rate α is too large, the convergence of the model is too fast and it may cross the optimal solution; if it is too small, the convergence of the model will be too slow. Likewise, if η is greater, the decay of α is greater and the convergence is gradually slowed; reversely, the convergence becomes faster.

4.4. Results and Analysis

We have mainly compared our model against the recent unsupervised methods, CycleGAN [4] for image-to-image translation and G^{2}GAN [10] for multiple-domain translation, and analyzed the experimental results. All models are trained and tested under the same hardware and software environment. The deep learning framework used for implementing the models is TensorFlow with Python API [37].

4.4.1. Comparative Results and Analysis

Figures 4–6 show all types of image translation results after five training rounds of three models. Figure 4 shows the object migration results of local and global images on *horse*2*zebra* dataset. In Figure 4a, we can see that MRGAN can retain enough semantic information of original images compared to two other methods with better boundary and texture of objects. On the other hand, there exist irregular white blocks in the transformed images of CycleGAN and G^{2}GAN, which means information loss with serious impacts on the quality of generated images. The results of global image translation in Figure 4b show that the MRGAN proposed in our method can maintain more realistic image background than CycleGAN and G^{2}GAN in object migration tasks after the same training rounds.

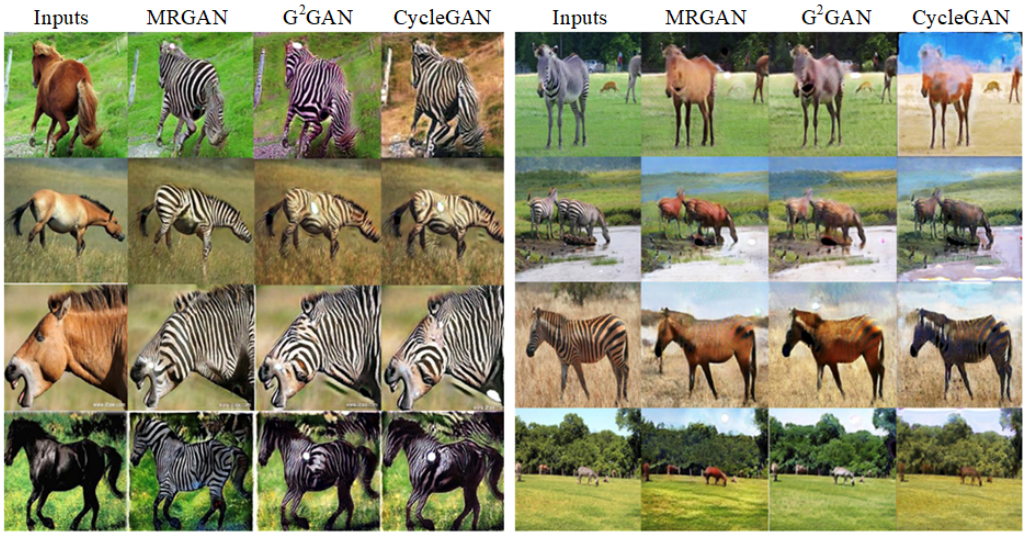

Figure 4. The experimental results on *horse*2*zebra* for object migration using MRGAN, G^{2}GAN and CycleGAN. (**a**) The result images of object migration from horses to zebras. (**b**) The result images from zebras to horses.

The experimental results of season conversion on *summer*2*winter* dataset are shown in Figure 5. It can be seen that the color of winter images generated by MRGAN is the most realistic in Figure 5a, followed by G^{2}GAN, and CycleGAN has the worst results. In the generated results of MRGAN, the overall environment with cool color and the grassland covered with snow tend to be more consistent with the human visual perception. Nevertheless, the generated results of CycleGAN seem to be more "warm" with more yellow, and G^{2}GAN's results feature an excessive "wintering" with more blue. Similar results are obtained for visual perception in Figure 5b.

Figure 6 shows the results of transformation between satellite photographs and 2D plane maps on *maps*. Since the color and texture composition of two style images is not very sophisticated and the boundary information is relatively significant, it is hard to intuitively distinguish the outcomes of three models. The generated roads may not be straight enough in the result maps because of obstructed buildings and trees in aerial images. In contrast, the transformation from two-dimensional maps to aerials can get more satisfactory results.

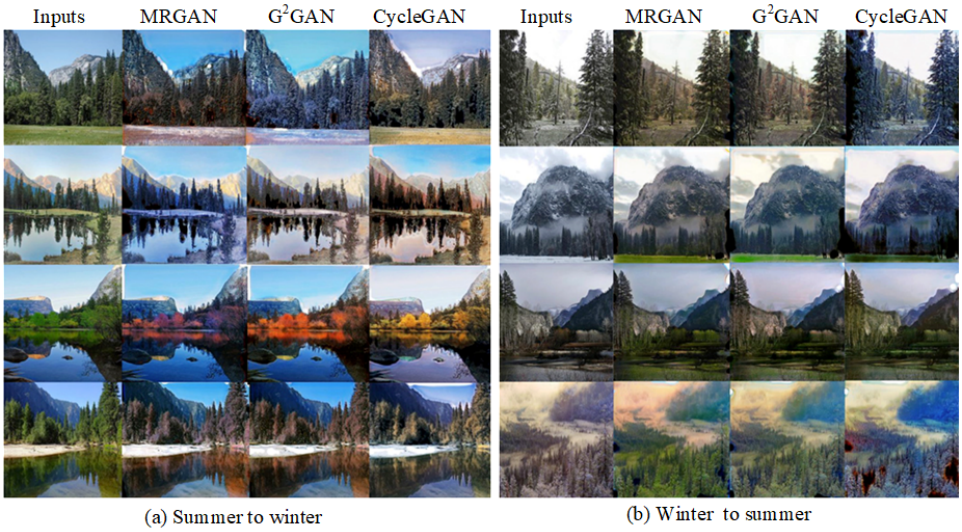

Figure 5. The experimental results on *summer*2*winter* for season conversion using MRGAN, G^{2}GAN and CycleGAN. (**a**) The result images from summer to winter. (**b**) The result images from winter to summer.

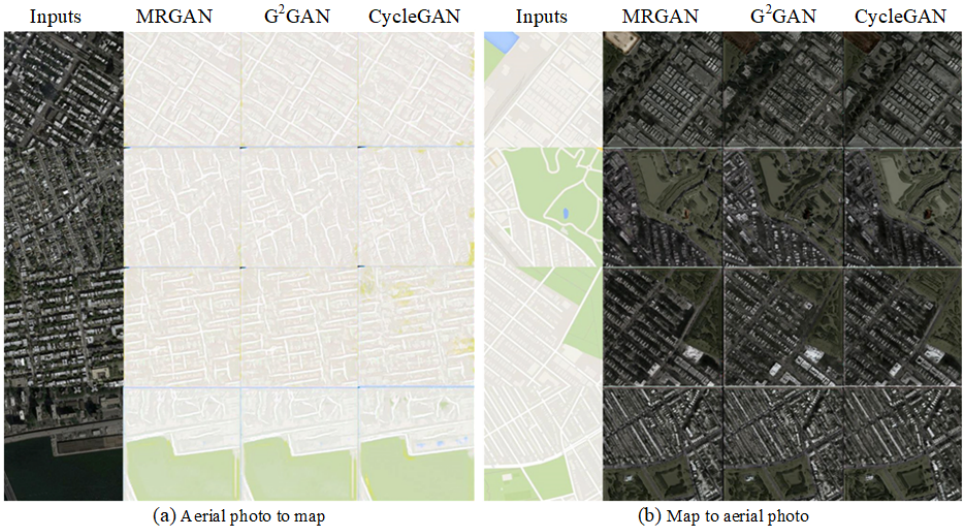

Figure 6. The results on map transformation task. (**a**,**b**) show the result images respectively using MRGAN, G^{2}GAN and CycleGAN for map transformation between aerial photos and 2D maps.

4.4.2. Time Performance Analysis

All models in the experiments were run on Nvidia GeForce GTX 1060, and we analyzed the time spent on training iterations on three models. Tables 1 and 2, respectively, report the average time consumption of different iterations after certain iterations (S_{train}) and the average time consumption of image generation after certain amounts (S_{gen}), where all trainings are performed for object migration on *horse*2*Zebra*. Table 1 shows that the training of MRGAN with multi-scale layered architecture is fastest on the same iteration numbers, followed by CycleGAN, and G^{2}GAN require the longest elapsed time. We can observe that the multi-scale network in MRGAN is more productive than the multi-channel network in G^{2}GAN when they both just use one discriminator. This may be because the multi-channel network requires to separately learn the attributes of an image over three channels of the same size. Additionally, the training time of MRGAN is about two-thirds as long as that of CycleGAN because it does not need to train two discriminators simultaneously. Table 2 shows the image average generating times of three models, and similarly MRGAN is fastest, followed by CycleGAN and the slowest is G^{2}GAN. The average time consumption of different iterations and image average generating times of the three models are basically stable and hardly change with the increase of iteration numbers and generated amounts.

Table 1. Average time consumption under different training iterations (s).

Iterations	MRGAN	G^{2}GAN	CycleGAN
10	1.57	2.53	2.13
50	1.60	2.48	2.14
100	1.52	2.58	2.13
200	1.57	1.54	2.12

Table 2. Average time consumption under different number of generated images (s).

Iterations	MRGAN	G^{2}GAN	CycleGAN
10	0.238	0.355	0.290
50	0.242	0.353	0.290
100	0.239	0.354	0.291
200	0.239	0.354	0.293

4.4.3. Analysis on Different Inputs and Training Strategies

Figure 7 shows the generated results when full images (MRGAN *w/o* MRF) and random image blocks with 70 × 70 pixels (MRGAN) are inputted into the discriminator. We can observe that the generated images with two types of inputs are almost the same, with very small differences in details. Such a patch-input discriminator has fewer parameters than a full-image discriminator, and can be applied for arbitrarily sized images with faster execution.

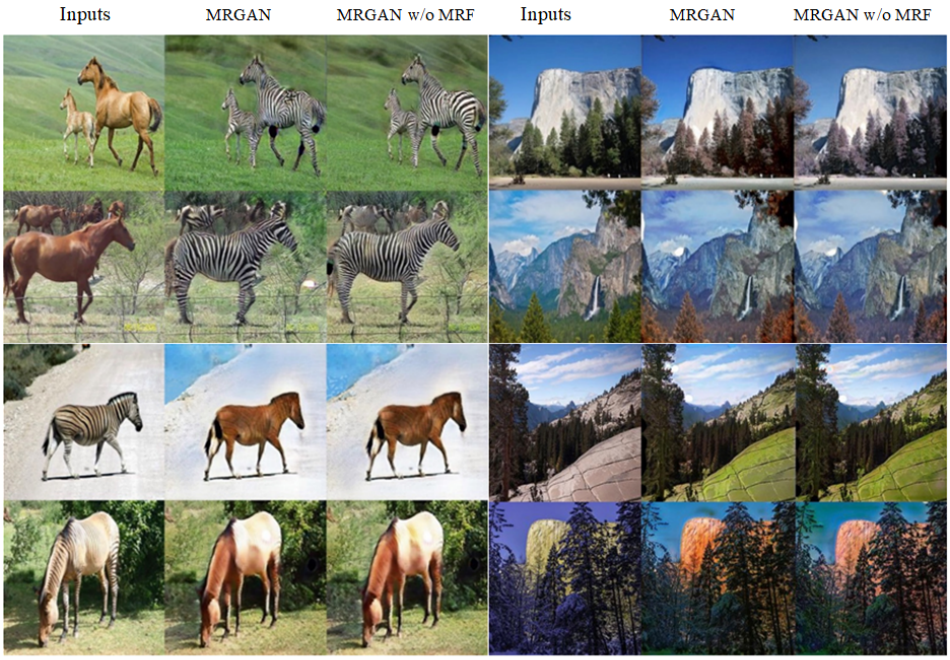

Figure 7. The effects on results using different types of inputs for the discriminator. (**a**) The migration between zebras and horses. (**b**) The conversion between winter and summer. From left to right in (**a**,**b**): input images, the results using random image block inputs with 70 × 70 pixels, and the results using full-image inputs.

In Figure 8, we have compared the effects of different normalization methods on the results after five rounds. The model normalized by instances is denoted as MRGAN w/IN. The figure shows that the results of MRGAN with group normalization are more satisfied than those of MRGAN w/IN, while there is a more reasonable stripe distribution on zebras and clearer objects' contours on the former.

Figure 9 shows the translated results using different initializations after five rounds, Xavier parameter initialization (MRGAN) and Gaussian initialization (MRGAN w/N). The scenery color of generated images by MRGAN using the Xavier parameter initializer is more real, which demonstrates that the Xavier initializer has a positive impact on model performance.

Figure 8. The effects on results using different normalization methods on the object migration between horses and zebras after five rounds. From left to right: input images, the results using the group normalization, and the results using the instance normalization.

Figure 9. The effects on results using different initializations after five rounds. From left to right: input images, the results using Xavier parameter initialization, and the results using Gaussian initialization.

4.4.4. User Studies

We conducted user studies to validate our work. Twenty images of each type were randomly selected separately from three test sets as inputs, and the image translation results of three tasks were obtained respectively by CycleGAN, MRGAN and DRUGAN. After shuffling the corresponding results of each input image as one group, 40 groups of images to be evaluated were finally obtained on each task. Since each task corresponds to a different set of images, we set different evaluation aspects for three tasks according to the characteristics of images, as shown in Figure 10. First, we assessed three aspects including object color, background color and object contour for the object migration between horses and zebras. We collected 36 responses per aspect on average, and a total score in each aspect for each group of results is 100. The overall scores of models were calculated according to the average scores of three aspects. Figure 10a shows that MRGAN achieves the best overall effect in the object migration between horses and zebras. We found that MRGAN obtained the best evaluations in all three aspects, especially in the aspects of the background color and the object contour, which confirms that our approach can better solve the problems of color leakage and structure instability. With two generators, G^2GAN had relatively strong background processing capability, and also did not differ much from MRGAN in overall evaluation, while CycleGAN achieved the lowest rating due to significant semantic loss. Second, we assessed in the same way three aspects including trees color, ground color and image tone for the season conversion between summer and winter images. On average, 36 responses per aspect were collected and the total score in each aspect was 100. Figure 10b shows the evaluated results of three aspects and the overall average scores. The results shows that MRGAN still has a significant advantage in ground color, while outscoring G^2GAN in trees color and tone too. Last, we conducted a third study to estimate the map transformation results. Since it is hard to intuitively distinguish the outcomes of map transformation, only boundary clarity was selected to evaluate its results. We only collected 23 available responses per group of results on average. Figure 10c shows the results of three models separately from the transformation of aerials to 2D maps and 2D maps to aerials. Although the results of three models are visually similar, the manual evaluation results can provide a reference for the analysis of the models. MRGAN had the highest overall score for transforming aerials to 2D maps, but it is only 2.1 percent higher than the lowest score of CycleGAN. This also shows the advantage of MRGAN for boundary processing. In the conversion of 2D maps to aerials, although MRGAN scored slightly lower than G^2GAN, it nonetheless produces faithful outputs.

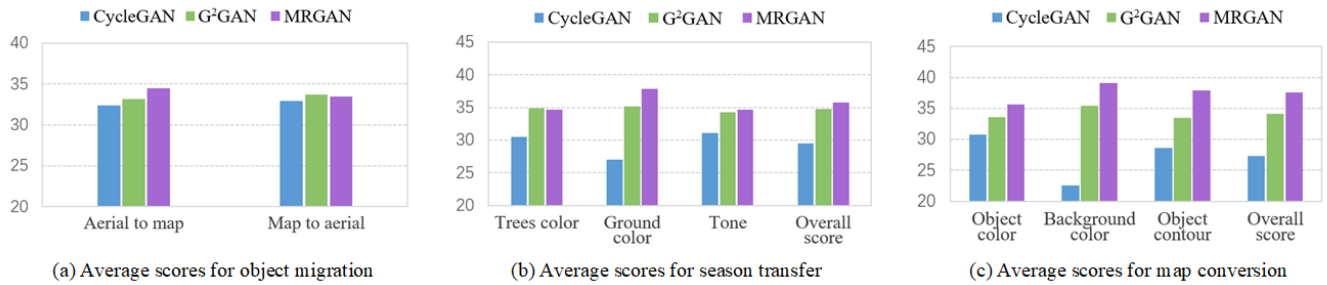

Figure 10. User study results confirm that our approach produces the best image translation results on three tasks. (**a**) The evaluation scores for different aspects in object migration between horses and zebras. (**b**) The evaluation scores for different aspects in season transfer between summer and winter images. (**c**) The evaluation scores for the boundary clarity in map conversion.

5. Conclusions

In this paper, an unsupervised model, MRGAN, is proposed for image-to-image transformation tasks, which consists of two generators and one discriminator. Compared with the existing methods, our proposed MRGAN can preserve the contextual structural and semantic information to avoid color leakage that might result from the lack of target guidance. Specifically, the generators of MRGAN adopt a multi-scale layered structure to

integrate the multi-grained context features into the residual units to eliminate the semantic loss in the image encoding process. Further, the WGAN-GP architecture is employed in the discriminator for gradient penalty instead of weight pruning to prevent instability in network training, as well as the randomly cropped image patches being used as inputs for the discriminator to enhance the robustness of the model. Last, the objective function of combining the reconstructed loss and the adversarial loss is used to optimize training and speed up converging. Experimental results show that MRGAN can generate more realistic images with less training time. However, there are some shortcomings in the detail of stylized images using MRGAN, such as uneven edges, which need to be resolved in our future work. At the same time, we will consider adopting our network structure on multi-domain image-to-image translation models for more generic applications. In future work, we will consider how to combine the pre-training models and multi-task learning for general image-to-image translation, with prompts or signal rules to guide the learning between different style images for improving the effect of general image translation tasks.

Author Contributions: Conceptualization, Methodology, Original draft preparation, Y.Z.; Data processing, Experiments and analysis, W.L.; Supervision, Funding acquisition, Reviewing and editing, D.W. and S.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (62172086, 61872074).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors are grateful to Isola, P. and Jun-Yan Zhu for providing the datasets used in the experiments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Luan, F.J.; Paris, S.; Shechtman, E.; Bala, K. Deep photo style transfer. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6997–7005.
2. Dumoulin, V.; Shlens, J.; Kudlur, M. A Learned Representation For Artistic Style. In Proceedings of the 5th International Conference on Learning Representations (ICLR)—Conference Track Proceedings, Toulon, France, 24–26 April 2017.
3. Johnson, J.; Alahi, A.; Li, F.F. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In Proceedings of the European Conference of Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016.
4. Isola, P.; Zhu, J.-Y.; Zhou, T.H.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5967–5976.
5. Zhu, J.-Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2242–2251.
6. Zhu, J.Y.; Zhang, R.; Pathak, D.; Darrell, T.; Efros, A.A.; Wang, O.; Shechtman, E. Toward multimodal image-to-image translation. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 466–477.
7. 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*, 149–159. [CrossRef]
8. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
9. Wang, X.; Gupta, A. Generative image modeling using style and structure adversarial networks. In Proceedings of the European Conference of Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 318–335.
10. Tang, H.; Xu, D.; Wang, W.; Sebe, N. Dual Generator Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In Proceedings of the 14th Asian Conference on Computer Vision (ACCV), Perth, Australia, 2–6 December 2018; pp. 3–21
11. Turn Memories into Art Using Artificial Intelligence. Available online: http://prisma-ai.com (accessed on 13 April 2021).
12. Ostagram. Available online: https://www.ostagram.me (accessed on 13 April 2021).
13. Deep Forger: Paint Photos in the Style of Famous Artists. Available online: http://deepforger.com (accessed on 13 April 2021).
14. Versa. Available online: https://www.versa-ai.com (accessed on 13 April 2021).
15. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.

16. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv* **2014**, arXiv:1411.1784.
17. Wang, T.C.; Liu, M.Y.; Zhu, J.Y.; Tao, A.; Kautz, J.; Catanzaro, B. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8798–8807.
18. Liu, Y.F.; Qin, Z.C.; Wang, H.; Luo, Z. Auto-painter: Cartoon Image Generation from Sketch by Using Conditional Wasserstein Generative Adversarial Networks. *Neurocomputing* **2018**, *311*, 78–87. [CrossRef]
19. Yi, Z.; Zhang, H.; Tan, P.; Gong, M.L. DualGAN: Unsupervised Dual Learning for Image-to-Image Translation. In Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2868–2876.
20. Kim, T.; Cha, M.; Kim, H.; Lee, J.K.; Kim, J. Learning to Discover Cross-Domain Relations with Generative Adversarial Networks. In Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017; Volume 4, pp. 2941–2949.
21. Zhu, J.-Y.; Krahenbuhl, P.; Shechtman, E.; Efros, A.A. Generative visual manipulation on the natural image manifold. In Proceedings of the European Conference of Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; LNCS 9909, pp. 597–613.
22. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein GAN. *arXiv* **2017**, arXiv:1701.07875.
23. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A. Improved Training of Wasserstein GANs. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 5768–5778.
24. Cao, F.D.; Zhao, H.C.; Liu, P.F.; Li, P.X. Input limited Wasserstein GAN. In Proceedings of the SPIE, Second Target Recognition and Artificial Intelligence Summit Forum, Shenyang, China, 28–30 August 2019; Volume 11427.
25. Ledig, C.; Theis, L.; Huszar, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image superresolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
26. Zhang, J.; Shu, Y.; Xu, S.; Cao, G.; Zhong, F.; Liu, M.; Qin, X. Sparsely grouped multi-task generative adversarial networks for facial attribute manipulation. In Proceedings of the ACM Multimedia Conference on Multimedia Conference (MM), Seoul, Korea, 22–26 October 2018; pp. 392–401.
27. Mathesul, S.; Bhutkar, G.; Rambhad, A. AttnGAN: Realistic Text-to-Image Synthesis with Attentional Generative Adversarial Networks. In Proceedings of the IFIP Conference on Human-Computer Interaction, Bari, Italy, 30 August–3 September 2021.
28. Denton, E.L.; Chintala, S.; Szlam, A.; Fergus, R. Deep generative image models using a laplacian pyramid of adversarial networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2015; pp. 1486–1494.
29. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. In Proceedings of the International Conference on Learning Representations (ICLR)—Conference Track Proceedings, San Juan, PR, USA, 2–4 May 2016.
30. Mao, X.D.; Li, Q.; Xie, H.R.; Lau, R.Y.K.; Wang, Z.; Smolley, S.P. Least squares generative adversarial networks. In Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2813–2821.
31. Zhang, H.; Xu, T.; Li, H. StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5908–5916.
32. Jiang, W.; Liu, S.; Gao, C.; Cao, J.; He, R.; Feng, J.S.; Yan, S.C. PSGAN: Pose and Expression Robust Spatial-Aware GAN for Customizable Makeup Transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
33. Hertzmann, A.; Jacobs, C.E.; Oliver, N.; Curless, B.; Salesin, D.H. Image analogies. In Proceedings of the ACM SIGGRAPH Conference on Computer Graphics, Los Angeles, CA, USA, 12–17 August 2001; pp. 327–340.
34. Choi, Y.; Choi, M.; Kim, M.; Ha, J.-W.; Kim, S.; Choo, J. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8789–8797.
35. Sangkloy, P.; Lu, J.W.; Fang, C.; Yu, F.; Hays, J. Scribbler: Controlling deep image synthesis with sketch and color. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6836–6845.
36. Karacan, L.; Akata, Z.; Erdem, A.; Erdem, E. Learning to generate images of outdoor scenes from attributes and semantic layouts. *arXiv* **2016**, arXiv:1612.00215.
37. Odena, A.; Olah, C.; Shlens, J. Conditional Image Synthesis With Auxiliary Classifier GANs. In Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017; pp. 2642–2651.
38. Yi, R.; Liu, Y.; Lai, Y.; Rosin, P.L. Unpaired Portrait Drawing Generation via Asymmetric Cycle Mapping. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 8214–8222.
39. Liu, Y.; Nadai, M.D.; Cai, D.; Li, H.Y.; Alameda-Pineda, X.; Sebe, N.; Lepri, B. Describe What to Change: A Text-guided Unsupervised Image-to-Image Translation Approach. In Proceedings of the 28th ACM International Conference on Multimedia (MM), Seattle, WA, USA, 12–16 October 2020; pp. 1357–1365.
40. Li, J.; Xiong, Z.; Liu, D.; Chen, X.J.; Zha, Z.J. Semantic Image Analogy with a Conditional Single-Image GAN. In Proceedings of the 28th ACM International Conference on Multimedia (MM), Seattle, WA, USA, 12–16 October 2020; pp. 637–645.

41. Lee, C.-H.; Liu, Z.; Wu, L.; Luo, P. MaskGAN: Towards Diverse and Interactive Facial Image Manipulation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 5548–5557.
42. Li, C.; Wand, M. Precomputed real-time texture synthesis with markovian generative adversarial networks. In Proceedings of the European Conference of Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 702–716.
43. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015; pp. 448–456.
44. Ulyanov, D.; Vedaldi, A.; Lempitsky, V. Instance Normalization: The Missing Ingredient for Fast Stylization. *arXiv* **2016**, arXiv: 1607.08022.
45. Wu, Y.; He, K. Group Normalization. *Int. J. Comput. Vis.* **2020**, *128*, 742–755. [CrossRef]
46. Glorot, X.; Bengio, Y. Understanding the Difficulty of Training Deep Feedforward Neural Networks. *J. Mach. Learn Res.* **2010**, *9*, 249–256.
47. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980.