Attention Guided Unsupervised Image-to-Image Translation with Progressively Growing Strategy

Yuchen WU$^{1[0000-0002-6554-467X]}$, Runtong ZHANG$^{1[0000-0002-8198-8457]}$, and Keiji YANAI$^{2[0000-0002-0431-183X]}$

[1] University of Electronic Science and Technology of China, Chengdu, China
[2] The University of Electro-Communications, Tokyo, Japan

Abstract. Unsupervised image-to-image translation such as CycleGAN has received considerable attention in recent research. However, when handling large images, the quality of generated images are not in good quality. Progressive Growing GAN has proved that progressively growing of GANs could generate high pixels images. However, if we simply combine PG-method and CycleGAN, it must bring model collapse. In this paper, motivated from skip connection, we propose Progressive Growing CycleGAN (PG-Att-CycleGAN), which can stably grow the input size of both the generator and discriminator progressively from 256×256 to 512×512 and finally 1024×1024 using the weight α. The whole process makes generated images clearer and stabilizes training of the network. In addition, our new generator and discriminator cannot only make the domain transfer more natural, but also increase the stability of training by using the attention block. Finally, through our model, we can process high scale images with good qualities. We use VGG16 network to evaluate domain transfer ability.

Keywords: Cycle-Consistent Generative Adversarial Networks · skip connection · attention block · progressive growing strategy

1 Introduction

CycleGAN [1] makes a big progress in unpaired domain translation, which is useful in industrial such as person re-identification [3] and video re-targeting [4]. Larger size pictures are appealing to all of them. With the development of the high-tech camera, there are more and more high pixels images exited. It will be a trend to do domain translation on large size pictures(1024×1024pixels).

Progressive Growing GAN (PG-GAN) [5] presents progressive growing methods for GANs to process large images, but if we simply cite the progressive growing method in CycleGAN, increasing the layers progressively. However, the generated images are not in good quality, which is shown in Fig. 1. This is because the span of the receptive field is enormous between the layers of CycleGAN, which will easily cause the model collapse. Shown in Fig. 1, they only change the color of the whole images, but not the domain.

To prevent such a case, we re-design the generator, whose architecture uses sampling to substitute the stride-2 convolution layers that are used in the original

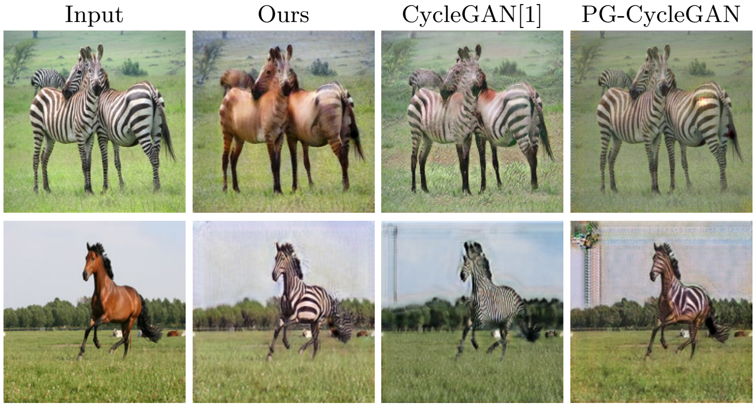

Fig. 1. The result of our GAN, CycleGAN[1], PG-CycleGAN (progressively growing the layer of original CycleGAN), for translating a zebra with a size of 1024 × 1024 to a horse and a horse to a zebra. By taking the details in the pictures, it is clear to see our generated horse and zebra are more rounded and nature.

CycleGAN. Besides, the kernel size of the first convolution layers of generators changes from 7 × 7 to 1 × 1 to reducing the reconstruction damage caused by encoding and decoding. Moreover, we replace all the transpose convolution layers with bilinear interpolation upsampling layer to erase the checkerboard effect. From Fig. 1, the results from our model have better-translated textures when handling the 1024 × 1024 size images.

We begin training with the 256 × 256 size, and after fully trained, we double the size to encourage on fine details. Besides, we use the attention block that protects the high-frequency information to have a clearer image. Comparing with the original CycleGAN and simply growing CycleGAN structure, we qualitatively and quantitatively show that explicitly our new progressive model can do well in domain translation for high pixels pictures.

2 Related Work

2.1 Cycle-Consistent Adversarial Networks

Cycle-Consistent Generative Adversarial Networks (CycleGAN) [1] introduced by Jun-Yan Zhu et al. uses two adversarial processes with two generators and two discriminators to realize two-way domain translation. The key to CycleGAN's success is the cycle-consistency loss, which represents cycle consistency and guarantees that the learned function can map an individual input to a desired output. The structure of the generator consists of encoders, transformers, and decoders, which result in a serious problem: edge information will be damaged in the encoding process and cannot be recovered in the decoding process. Therefore, some parts of the generated images are blurry and indistinct. To improve the image quality, we use skip-connection to connect encoder and decoder.

Thus the architecture can prevent edge information from being damaged and will be directly transmitted to generated image. Besides, we softly enlarge the generator, which prevents the model collapse. These methods can get better results compared with other models. Finally, our model based on the growing technology can well handle the large scale images.

2.2 Skip Connection

Olaf Ronneberger et al. introduce U-Net [6] to make convolution networks could work for bio-medical images. In terms of the high-quality images, which are important in medical, they use the skip connection between the sampling and upsampling layers. To increase the speed of the architecture, many structures use the sampling to minimize the size of the processing images, which will throw away the high-frequency information that includes the edge information. With the help of the skip connection, the detail information directly transfers to the upsampling layers, thus can have clearer images. We adopt the skip connection between the encoder and decoder inside the generator. Besides, we will also establish a new skip connection with the network growing. Though the network is much deeper, it will still have good quality in generated pictures.

2.3 Progressive Growing of GANs

Progressive Growing of GANs (PG-GANs) [5] realizes size increase by using a progressive growth strategy. In this training process, our model begins from a small output size and gradually adds new layers in output end to expand size, which is realized by weight α changing from 0 to 1. When α increases to 1 as the training process, the new layer is completely added to the model and the output size is expanded. Different from the PG-GAN, our model base on the image input-output structure. Motivated by this progressive growing strategy, we also use weight α to linearly add new layers in both input end and output end to increase image size. Besides, we also increase the size of the discriminator, to prevent model collapse caused by the situation when the discriminator is over trained.

2.4 Artifact

Youssef A. et al. introduce Attention CycleGAN [7] to protect the background information in datasets like horse2zebra and summer2winter. Using the attention block that only focuses on the domain part, which won't do superfluous translation on the background. Odena et al. [8] discover the checkerboard effect in image processing, which is caused by transpose convolution layers. In this model, we not only present an alternative attention block to keep the milieu but also choose upsampling layers instead of transpose convolution layers to solve the checkerboard effect.

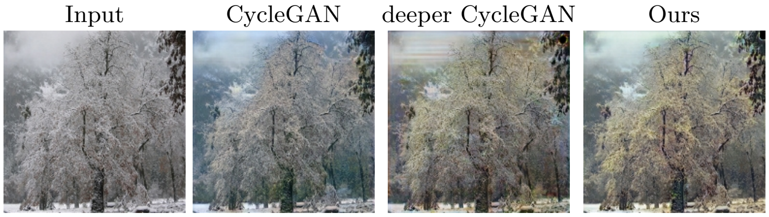

Fig. 2. The result of CycleGAN, deeper CycleGAN(CycleGAN has more layers to deepen the network), our model for translating the picture from winter to summer

3 Proposed Method

Image translation that aims to learn mapping function from source domain to target domain with two sets of independent data, is realized by an Image Transform Net[9]. Style change effect can be improved with more encoder and decoder layers, but the reconstruction loss will increase due to the downsampling process in encoder. Shown in Fig. 2, the deeper CycleGAN can change the color of the tree, while it also causes the sky distorted. To solve this problem, we combine the progressive growth strategy with CycleGAN to propose a new architecture, which can smoothly add new layers to generators after adequate training. Fig. 3 visualizes this process.

3.1 Network Structure

Base Structure of Generator In the generator, the first layer named fromRGB is a convolution layer with 1 × 1 kernel size adopted from PG-GAN[5], which has a good performance on generating high-quality images. The architecture of encoder is adopted from Image Transform Net[9] consisting of two sampling blocks including two 1-stride convolution layers followed by Instance normalization[10] and ReLU, average sampling to shrink images. Same as CycleGAN[1], we use nine residual blocks as the transformer part. Besides, before the transformer part, a skip-connection with weight transmits data skipping the transformer to decoder. For the decoder part, motivated by Stack GAN[11], we choose two bilinear interpolation upsampling layers integrated with two 1-stride convolution layers to expand image size instead of transpose convolution. Moreover, the input of the second upsampling layer is the integration of output from the last layer and data from the first convolution layer in encoder transmitted by a tunnel. Finally, the output is fed in a 1 × 1 convolution layer named toRGB to reduce the dimension back to the RGB image.

Base Structure of Discriminator We use three 2-stride convolution layers followed by Instance Normalization and LeakyReLU to make quick judgments, which is inspired by FCN[12]. Due to the flexibility of FCN, we can easily add layers to achieve a progressively growing effect.

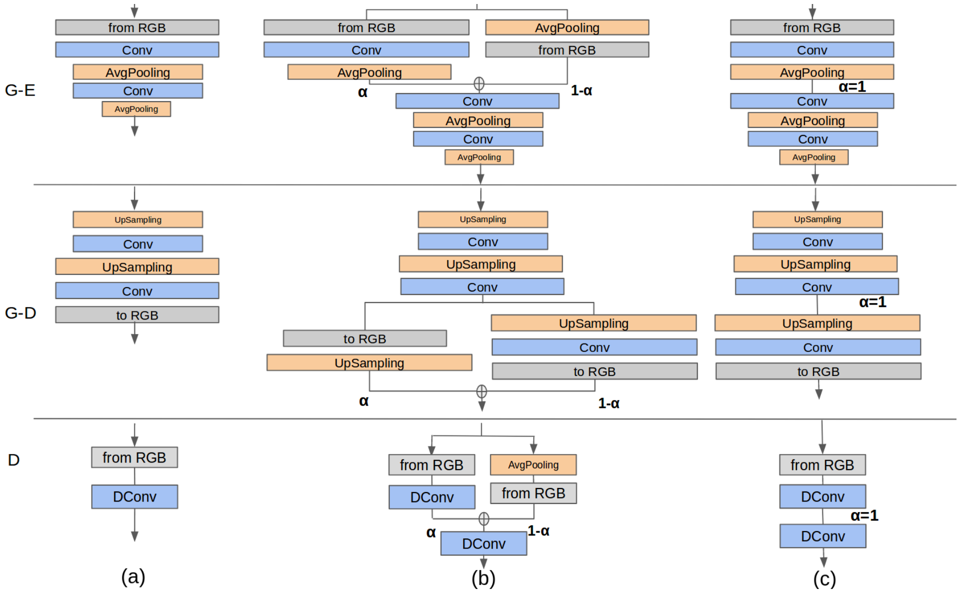

Fig. 3. When growing the layers of the encoder of the generator (G-E), the decorder of the generator (G-D) and discriminator (D), we fade in the new layers smoothly. This example illustrates the translation of the deepening parts, (a) to (c). During the transition (b), we grow α linearly from 0 to 1. fromRGB represents RGB to feature vectors, using the 1×1 convolution layers. toRGB is feature vectors to RGB. Conv means 2 stride-1 3×3-convolution layers. Dconv is a stride-2 3×3-convolution layer. When training the discriminator, we feed in real images that are downsampled to increase the judgement on semantic information

3.2 Progressive Growing Strategy

We adopted the progressive growing method of PG-GAN and modified it to suit the encoder-decoder style since the PG-GAN only generates images from random noise of 1×512 codes. The progressively added layers method is shown in Fig.3. When the network has been trained after adequate training epochs, the progressive growing stage will begin. Firstly, the input image size needs to be enlarged from the original 256×256 to 512×512. Since we softly add the layer, we use two ways to gradually shade, shown in Fig. 3. For the original round, there will be a new pooling layer before fromRGB to adjust the size because it can only accept 256×256 images. For the other way, the growing layers, which consists of a fromRGB layer of 512×512 and two 1-stride convolution layers with average sampling, will be gradually added to the well-trained structure. There are two weights α and $1 - \alpha$ working on growing layers and original layers separately. With α growing from 0 to 1, the original way will be gradually abandoned and adding layers will progressively integrated well with other parts of this network and a new architecture will be completed. In the decoder part, the process is similar to that in the encoder. Due to the above idea, our method has great training stability.

3.3 Attention Block

Our attention block aims to find the target domain part and the source port in the pictures. If we add layers into the attention block with the training growing, it will destroy the well-trained attention block, which will need to train again in turn. On the contrary, we keep input 256×256 size images, using bicubic [13] to increase the size to maintain the stability. Like Wang et al's work [14], we use residual units in our network to increase the accuracy of an attention block.

3.4 Training

The work of domain translation is using a generator $\boldsymbol{G}_{st}$ that translates input image s from a source domain into s' in target domain which is based on a possibility of P_t. At first, we use an attention network A_s, which can locate the source domain part in the images. For the output of the A_s, it is an attention map with per-pixel [0, 1], allowing the network to learn how to compose edges. After the attention block, we can get an image only with the domain part $A_s(s)$ and an image only with the background 1-$A_s(s)$, and the other part is just pixels with zero value. Finally, we put the domain part inside the generation and can get the target domain image. We use '$\odot$' to represent the element-wise product. Thus, the mapping from the source domain to the target domain is:

$$s' = (1 - A_s(s)) \odot s + A_s(s) \odot G_{st}(s) \tag{1}$$

We use the progressively growing method to deepen the network, which can handle the large scale images. Before adding a new layer, the model should be fully trained. Through a lot of experiments, we observed that after 100 epochs, it would change a little in the original model. Therefore, after 100 epochs, we will grow the layers in the generation and attention block. We use the G_{stnew} to represent the latest generation and use $A*_s$ (we do not change the attention block after 30 epochs) as the latest attention block. Using α as weight in progress. The progressively mapping is:

$$s' = (1 - A *_s (s)) \odot s + (\alpha G_{stnew} + (1 - \alpha) G_{st})(A *_s (s) \odot s) \tag{2}$$

We use F_{st} and F_{ts} to represent the domain translating. D_t and D_s present the process of discriminators. So the adversarial loss function can be shown as:

$$\mathcal{L}^s_{adv}(F_{st}, A_s, D_s) = \mathbb{E}_{t \sim P_t(t)}[\log D_t(t)] + \mathbb{E}_{s \sim P_s(s)}[\log 1 - D_t(s')] \tag{3}$$

In addition, we enforce network by using cycle consistency loss: calculate the difference between original image s and inverse mapping image s", which is s transferred back to original domain by F_{st} and F_{ts}. This process is shown below:

$$\mathcal{L}^s_{cyc}(\boldsymbol{s}, \boldsymbol{s}'') = ||\boldsymbol{s} - \boldsymbol{s}''||_1 \tag{4}$$

The cycle consistency loss can further reduce the space of possible mapping functions and increase the attention block. Finally, we combine the attention loss and cycle consistency as:

$$\mathcal{L}(F_{st}, F_{ts}, A_s, A_t, D_s, D_t) \quad = \quad \mathcal{L}^s_{adv} \ + \ \mathcal{L}^t_{adv} \ + \ \lambda(\mathcal{L}^s_{cyc} \ + \ \mathcal{L}^t_{cyc})) \tag{5}$$

The optimal parameters of λ e obtained by solving the minimax optimization problem:

$$F*_{st}, F*_{ts}, A*_s, A*_t, D*_s, D*_t = \underset{F_{st}, F_{ts}, A_s, A_t}{argmin} \\ (\underset{D_s, D_t}{argmax}\mathcal{L}(F_{st}, F_{ts}, A_s, A_t, D_s, D_t)) \tag{6}$$

For discriminator. At first, the attention block is not precise enough if we just focus on the target part, which will cause the model collapse by combining the information of the background, e.g., in the horse2zebra is the living condition of zebra. To overcome this problem, we train the discriminator with the full image before the first 30 epochs and switch to only the attention part after attention block has developed.

Unpaired image translation generate the pictures will also influenced by the background. Unlike traditional attention block, we should make the boundary sharper to decrease the influence of background. We calculate the attention map as follows:

$$t_{new} = \begin{cases} t \ \ \textit{if } A_t(t) > \tau \\ 0 \ \ \textit{otherwise} \end{cases} \tag{7}$$

$$s'_{new} = \begin{cases} F_{st}(s) \ \ \textit{if } A_s(s) > \tau \\ \quad 0 \quad \textit{otherwise} \end{cases} \tag{8}$$

t_{new} and s'_{new} are masked versions of target sample t and translated source sample s', which only contain pixels exceeding a user-defined attention threshold τ, which we set to 0.1.

Finally, we update the adversarial loss L of Equation (3) to:

$$\mathcal{L}^s_{sdv}(F_{st}, A_s, D_t) \ = \ \mathbb{E}_{t\sim P_t(t)}[\log D_t(t_{new})] + \mathbb{E}_{s\sim P_s(s)}[\log 1 - D_t(s'_{new})] \tag{9}$$

When optimizing the objective in Equation (8) beyond 30 epochs, real image inputs to the discriminator is now also dependent on the learned attention maps. This can lead the model to collapse if the training is not performed carefully. For instance, if the mask returned by the attention network is always zero.

$$\mathcal{L}^s_{sdv}(F_{st}D_t) \quad = \quad \mathbb{E}_{t\sim P_t(t)}[\log D_t(t)] \ + \ \mathbb{E}_{s\sim P_s(s)}[\log 1 - D_t(s')))] \tag{10}$$

Our model is based on the circulation from source domain to target domain, and back. Which is shown as $\phi_s \to \phi_{st} \to \phi_{sts}$, so the cycle consistency is same as function(4). To combine them, the full object is:

$$\mathcal{L}(F_{st}, F_{ts}, D_s, D_t) = \mathcal{L}^s_{adv} + \mathcal{L}^t_{adv} + \lambda(\mathcal{L}^s_{cyc} + \mathcal{L}^t_{cyc}) \tag{11}$$

The solution is similar to the model with attention block, just without the attention part, which is :

$$F^*_{st}, F^*_{ts}, D^*_s, D^*_t = \underset{F^*_{st},F^*_{ts}}{argmin}(\underset{D^*_s,D^*_t}{argmax}\mathcal{L}(F^*_{st}, F^*_{ts}, D^*_s, D^*_t) \qquad (12)$$

4 Experiments

4.1 Training setting

We use the 'Apple to Orange' (A2O) and 'Horse to Zebra' (H2Z) datasets provided by Jun-Yan Zhu et al.[1] to train our model with attention block since such images have exact foreground object. For our model without attention block, we choose the datasets celeba datasets HD from Tero Karras et al.[5], summer2winter(Yosemite) and monet2photo, which are also from CycleGan[1].

We adopt CycleGAN's notation [1], "c3s1-*k*-R" denotes a 3*3 convolution with stride 1 and *k* filters, followed by a ReLU activation ('R'), while Leaky ReLU activation with slope 0.2 ('LR'). "ap" denotes an average pooling layer halving the input layer. "r*k*" denotes a residual block formed by two 3*3 convolutions with *k* filters, stride 1 and a ReLU activation. "up" denotes a upsampling layer doubling the heights and widths of its input. A Sigmoid activation is indicated by 'S' and 'tanh' by 'T'. We apply Instance Normalization after all layers apart from the last layer.

Final generator architecture is: c1s1-32-R, c3s1-64-R, c3s1-64-R, ap, c3s1-64-R, c3s1-64-R, ap, c3s1-128-R, c3s1-128-R, ap, r128, r128, r128,r128, r128, r128, r128, r128, r128, up, c3s1-64-LR, c3s1-64-LR, up, c3s1-32-LR, c3s1-32-LR,up, c3s1-32-LR, c3s1-32-LR, c1s1-3-T.

Attention block architecture is: c7s1-32-R,c3s2-64-R, r64, up, c3s1-64-R, up, c3s1-32-R, c7s1-1-S.

Final discriminator architecture is: c3s1-64-LR, c4s2-32-LR,,c4s2-64-LR, c4s2-128-LR, c4s2-256-LR, c4s1-512-LR, c4s1-1

Similar to CycleGAN, we use the Adam solver with a batch size of 1. All networks were trained from scratch with a learning rate of 0.0002. We keep the same learning rate for the 200 epochs. Weights are initialized from a Gaussian distribution $\mathcal{N}$(0,0.02). Layers are added in 140, 170 epochs.

4.2 PG-Method and Attention block

Observing the Fig. 5, we can see that the generated images are getting more and more fine details through training, which means our progressively growing method work. When it in step1, there are only limited strips on the generated zebras, but as the step grows, the strips are getting more and more. Finally,

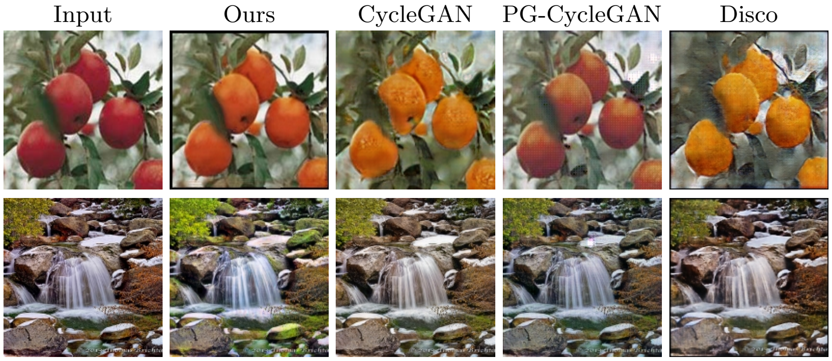

Fig. 4. Translation results. From top to bottom. zebra to horse, horse to zebra, apple to orange, summer to winter, winter to summer, Monet to picture and blond to brown. For the first three translations, our results are generated with attention block, and for the last four translation, attention block is not used.

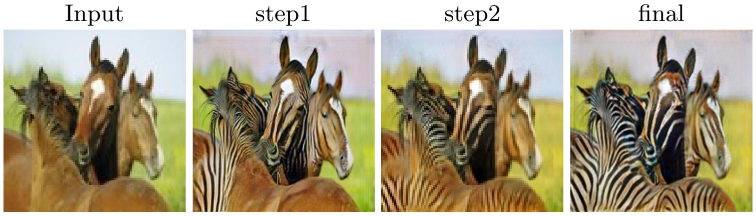

Fig. 5. Domain translation for generating a zebra images by a horse image. Results of step1(fully trained with 256×256 size images, which is same as CycleGAN), step2(fully trained with 512×512 images), and step3(fully trained with 1024×1024 pictures, the model already finished growing). With the layers growing, some fine details are added. The strips of generated zebra is adding with the step increasing.

all the generated zebras are covered with strips, which makes them really like zebras.

The function of the attention block is to tract the domain part in the image, which will protect the background information while in translation. In Fig. 6, looking at the attention maps (the grey images), each of them can accurately find the source domain. As a result, shown in the photos after the attention block, the generated pictures will have the same background as the original images have.

4.3 Baselines

Nowadays, there are many famous GANs performing well in domain transferring using different loss. CycleGAN [1] with least-squared GAN[16] loss and DiscoGAN [15] with Standard loss[17] use a circulation to train adversarial network. Dual GAN[19] uses Wasserstein GAN loss [18] to solve the model collapse. To prove our model really work well on high pixels images, we compare our

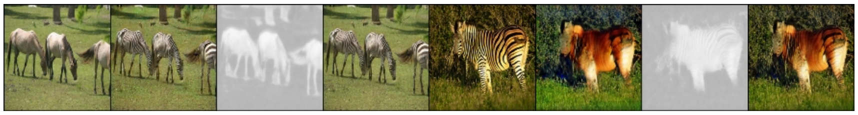

Fig. 6. Results of attention block for zebra to horse and horse to zebra domain translation of four group. The order inside each group is input image, generated image without attention map, attention map and generated image with attention map. The attention block can correctly tract the domain part inside the images.

Table 1. VGG perception for each model

Model	H2Z	Z2H	A2O	O2A	M2P	P2M
VGG(accuracy)	0.99	0.99	0.96	0.96	0.98	0.98
Ours	0.78	0.94	0.84	0.83	0.84	0.27
CycleGAN	0.82	0.90	0.77	0.87	0.83	0.11
PG CycleGAN	0.75	0.65	0.62	0.59	0.72	0.09
Disco GAN	0.63	0.19	0.80	0.19	0.74	0.35

[a] Higher scores mean better model.

[b] H (horse),Z(zebra),A(apple),O(orange),M(monet), P(photo)

model with CycleGAN[1], CycleGAN[1] with progressively growing method[5] and DiscoGAN [15] on 1024 × 1024 images.

4.4 Qualitative Results and Quantitative Results

Fig. 4 shows the results of Horse2zebra, Apple2Orange, Summer2Winter, Monet2Photo, and blond2brown datasets. Although CycleGAN has a strong ability in domain transfer, the background will be changed by trained mapping function due to loss function, which is based on whole the image. Moreover, when we zoom in the image from CycleGAN, obvious checkerboard artifact resulted from transposed convolution layers can be observed. The simple combination of PG-GAN and CycleGAN do not have good performance. Because the receptive field changes a lot when new layers are added, the model collapsed is easier to occur. DiscoGAN focuses on the relationship between two domains, but can only realize unidirectional domain translation, such as horse2zebra in Fig.4. By incorporating the progressive growing strategy, attention block and replacing transpose convolution with bilinear interpolation upsampling, our results have less checkerboard effect, more natural background and stronger ability of domain transfer. Our model successfully makes output more realistic compared with other models and manages to solve the checkerboard artifacts.

We use the VGG-16 network [19] to quantitatively evaluate the authenticity of our generated images. VGG-16 is a classical model in Image Identification. Comparing with AlexNet [20], VGG-16 used stacked small convolution kernels increasing the depth of the network with fewer parameters. We prepare a unique VGG-16 network for each datasets, expect winter2summer that VGG-16 only has 70% accuracy. For the training datasets are the same with the datasets we used in domain translation. As we wanted our domain translation is more natural, which means our results should have a higher grade in VGG-16 network. Therefore, we calculate the accuracy of each dataset in table 1, where we also list the accuracy of VGG-16 network for test images.

Our approach reaches the highest score in most domain translation, while the CycleGAN is the second, which means it does well in domain translating. Although PG CycleGAN uses the progressive growing method, it is third one of all, because the model is not stable enough, then adding layer will always just learn to change the color instead of the domain translation. Because of the loss function used by DiscoGAN, it can reach good results in one direction domain translation. Finally, comparing with these GANs, it is obvious that our model deepens the understanding of the semantic information through progressive training and enhances the vision of the attention block.

5 Conclusion

Simply combination of a progressively growing method with CycleGAN will easily cause model collapse. In this paper, we present a more stable GAN–PG-Att-CycleGAN. The architecture trains an adversarial network gradually with the help of attention block, and fix the generator to reach the goal. Our method can greatly reduce the damage of the deep layer to the spatial information. Besides, with the help of the increased number of layers and skip connection, we can generate images with more natural textures.

References

1. Jun-Yan Zhu,Taesung Park, Phillip Isola, Alexei A.Efros: Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In:ICCV,(2017)
2. Pavel Pecina, Ondřej Dušek, Lorraine Goeuriot, Jan Hajič, Jaroslav Hlaváčová, Gareth J.F.Jones, Liadh Kelly, Johannes Leveling, David Mareček, Michal Novák, Martin Popel, Rudolf Rosa, Aleš Tamchyna, Zdeňka Urešová:Adaptation of machine translation for multilingual information retrieval in the medical domain. In:Artificial Intelligence in Medicine Volume 61, Issue 3, July 2014, Pages 165-185.
3. Longhui Wei, Shiliang Zhang, Wen Gao, Qi Tian: Person Transfer GAN to Bridge Domain Gap for Person Re-Identification In:CVPR,(2018)
4. Aayush Bansal, Shugao Ma, Deva Ramanan, Yaser Sheikh: Recycle-GAN: Unsupervised Video Retargeting In:ECCV,(2018)
5. Tero Karras, Timo Aila, Samuli Laine , Jaakko Lehtinen:Progressive Growing of GANs for Improved Quality, Stability and Variation. In:ICLR,(2018).
6. Olaf Ronneberger, Philipp Fischer, Thomas Brox:U-Net: Convolutional Networks for Biomedical Image Segmentation. In: MICCAI,(2015).
7. Youssef A., Christian Richardt, James Tompkin, Darren Cosker. "Unsupervised Attention-guided Image-to-Image Translation". In NIPS, 2018.
8. Augustus Odena, Vincent Dumoulin, Chris Olah:Deconvolution and Checkerboard Artifacts. Distill, (2016).
9. Justin Johnson, Alexandre Alahi, Li Fei-Fei:Perceptual Losses for Real-Time Style Transfer and Super-Resolution. arXiv preprint arXiv:1603.08155,(2016)

10. Dmitry Ulyanov, Andrea Vedaldi, Victor Lempitsky:Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022,(2016)
11. Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, Dimitris Metaxas:StackGAN:Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks. arXiv preprint arXiv:1612.03242,(2016).
12. Jonathan Long, Evan Shelhamer:Fully Convolutional Networks for Semantic Segmentation. In:CVPR,(2015)
13. R Bellman, B K ashef, R Vasudevan. Dynamic programming and bicubic spline interpolation. In Journal od Mathematical Analysis and Applications 44, pages 160-174, 1973.
14. Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, Xiaoou Tang. Residual Attention Network for Image Classification. In arXiv preprint arXiv:1704.06904, 2017.
15. Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, Jiwon Kim:Learning to Discover Cross-Domain Relations with Generative Adversarial Networks. In:InternationalConference on Machine Learning,(2017)
16. Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, Stephen Paul Smolley:Least Squares Generative Adversarial Networks. arXiv preprint arXiv:1611.04076,(2016)
17. Ian Goodfellow, Jean PougetAbadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio:Generative adversarial nets. In:NIPS,(2014)
18. Martin Arjovsky, Soumith Chinatala, Léon Bottou:Wassertein GAN. arXiv preprint arXiv:1701.07875,(2017)
19. Karen Simonyan, Andrew Zisserman:Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556,(2016)
20. Alex Krizhevsky, Ilya Sutskever, Geoffrey E.Hinton:ImageNet Classification with Deep Convolutional Neural Networks. In:NIPS,(2012)

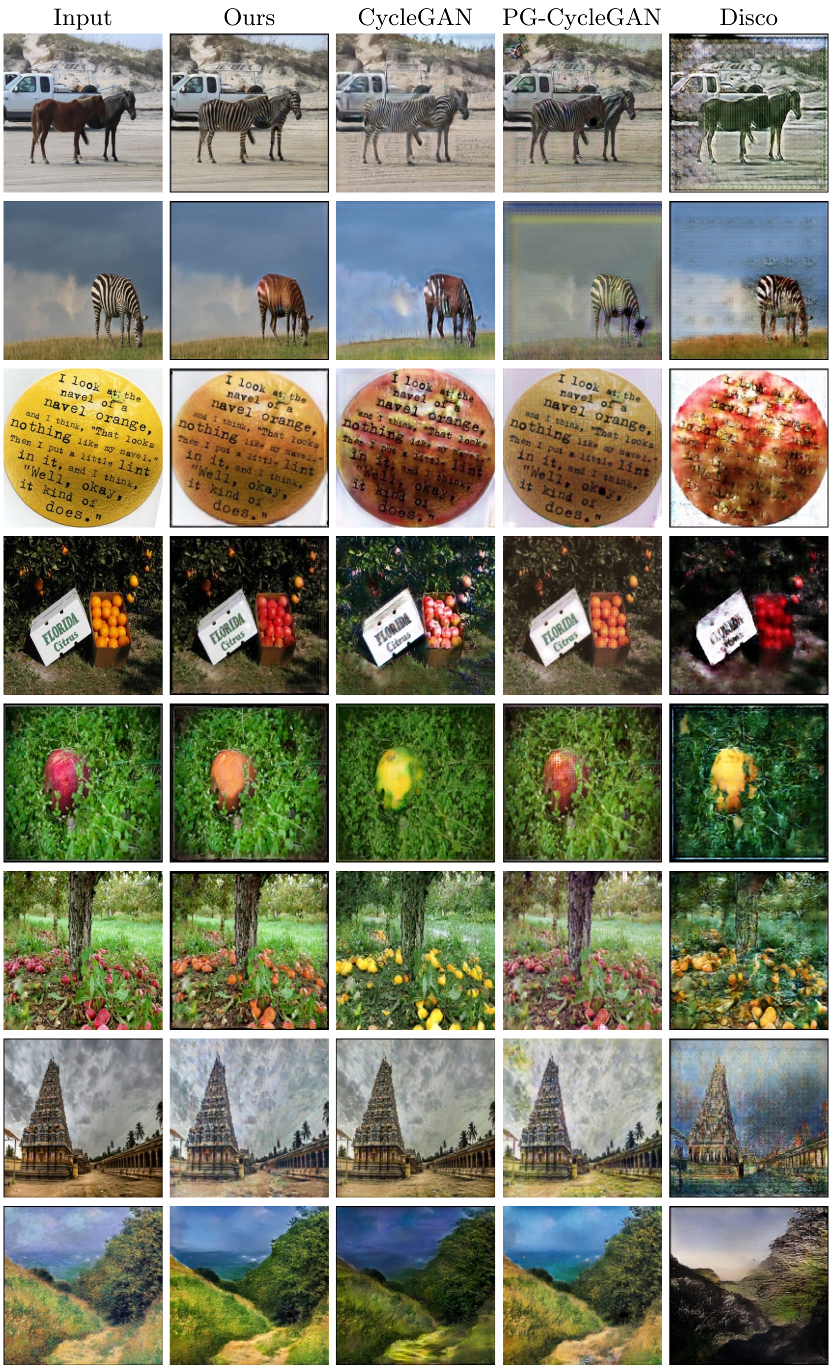

Fig. 7. Translation results. From top to bottom. horse to zebra, zebra to horse, orange to apple, apple to orange, picture to monet, monet to picturen. For the first five translations, our results are generated with attention block, and for the last three translation, attention block is not used.

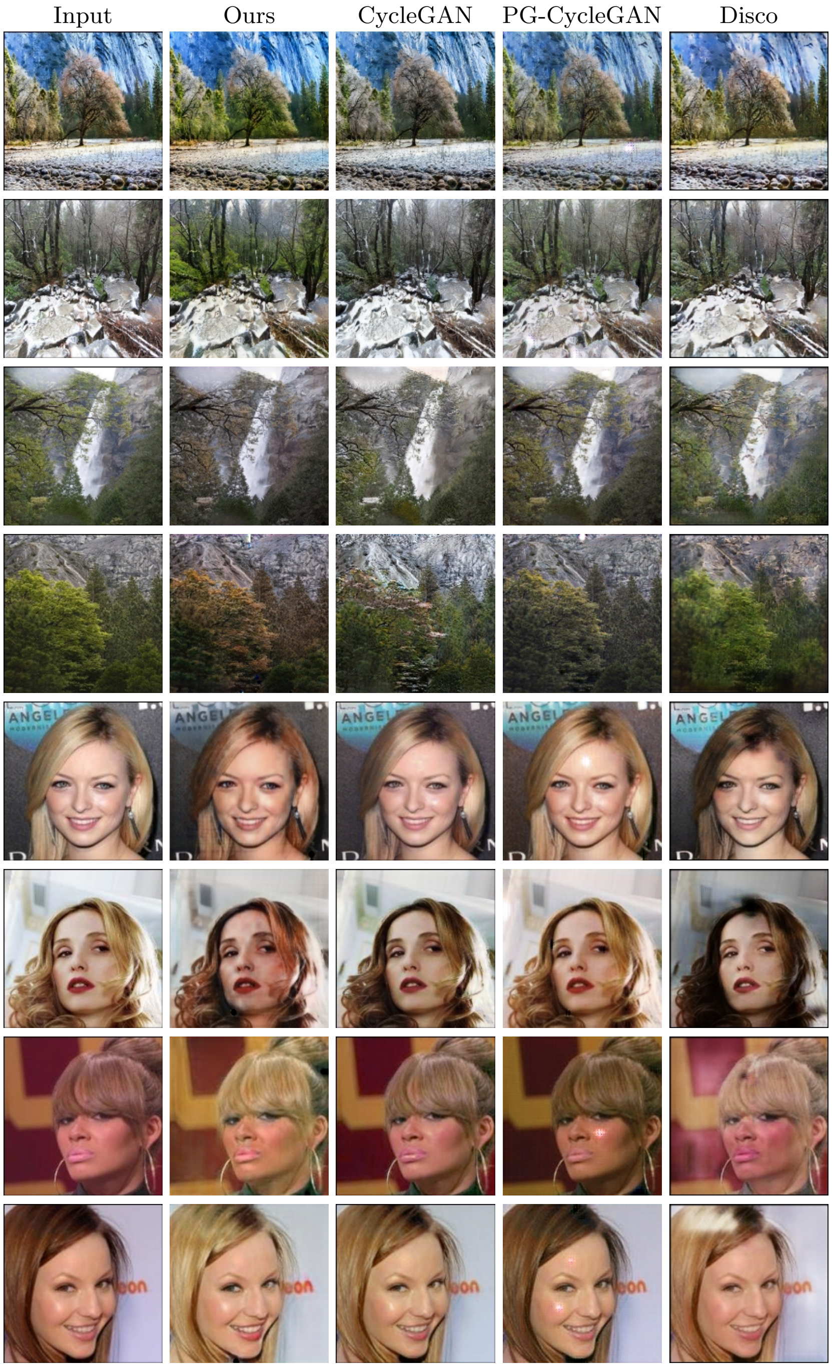

Fig. 8. Translation results. From top to bottom. winter to summer, summer to winter, blond hair to brown hair, brown hair to blond hair. Attention Block is not used.