Pre-trained and Shared Encoder in Cycle-Consistent Adversarial Networks to Improve Image Quality

Runtong ZHANG[1][0000−0002−8198−8457], Yuchen WU[1][0000−0002−6554−467X], 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. Images generated from Cycle-Consistent Adversarial Network (CycleGAN) become blurry especially in areas with complex edges because of loss of edge information in downsampling of encoders. To solve this problem, we design a new model called ED-CycleGAN based on original CycleGAN. The key idea is using a pre-trained encoder: training an Encoder-Decoder Block (ED-Block) at first in order to get a difference map, which we call an edge map and is produced by the subtraction of input and output of the block. Then, the encoder part of a generator in CycleGAN share the parameters with the trained encoder of ED-Block and they will be frozen during training. Finally, by adding the output from a generator to the edge map, higher quality images can be produced. This structure performs excellently on "Apple2Orange", "Summer2Winter" and "blond-hair2brown-hair" datasets. We use SSIM and PSNR to evaluate resolution of results and our method achieved the highest evaluation scores among CycleGAN, Unit and DiscoGAN.

Keywords: ED-Block · edge map · pre-trained Encoder · Cycle-Consistent Adversarial Networks.

1 Introduction

CycleGAN [10] realizes domain translation in the absence of paired data. The structure of two generators consists of three parts: an encoder, a transformer and a decoder. The size of images shrinks in the encoder, stays constant in the transformer, and expands again in the decoder. Because of downsampling process realized by stride-2 convolution layers of encoder, the information of edges in original images are lost. After being processed by the transformer, the images expands in the decoder but only a part of edge information are restored. Therefore, the output images will be blurry. Especially for some complex pictures, edges are mixed and it is difficult to distinguish object shapes.

Our contribution is to suppose a new network based on CycleGAN to improve image quality with appropriate amount of parameters and time expense. This network works well on "Apple2Orange" and "Summer2Winter" datasets provided by Jun-Yan Zhu et al. [10] and also performs well on "blond-hair2brown-hair" datasets, which we collected from celebA dataset provided by [8]. Our

network can be easily trained by only the highly successful backpropagation and model freezing. This method need high texture similarity between input and output and focus on color translation. Therefore, it can be used in enhancement of domain translated images, virtual makeup, hair-color changing and so on, all of which need color translation and high quality images.

2 Related Work

2.1 Cycle-Consistent Adversarial Networks

CycleGAN [10] realizes unpaired training data in image-to-image translation. This network consists of two Generative Adversarial Networks (GANs) [1] and inputs are two images in different domains. One input is translated from domain X to domain Y and the other is from Y to X. The key to CycleGAN's success is 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 two generators adopted from Justin Johnson et al. [3] is the encoder-transformer-decoder: the input images will be shrunk in the encoder and expanded in the decoder. Information of edges are lost in this processing since downsampling is irreversible and transposed convolution layers cannot totally restore edge information in the decoder part. So when observing the output from CycleGAN, we can find some areas in images are blurry and indistinct. To solve this problem, our method is adding a new block called ED-Block in CycleGAN, which can extract edge map from input image. In this way, the edge information is protected from being lost in the encoder. By adding the edge map to the output from the generator, we can get the much clearer image as an output.

2.2 Super-Resolution

Super-Resolution (SR) refers to the reconstruction of corresponding high-resolution images from observed low-resolution images, which has important application value in monitoring equipment, satellite images and medical imaging. Super-Resolution Generative Adversarial Network (SRGAN) [6] is a SR problem method based on deep learning, using GAN [1]. The key point is that: since traditional method cannot make results enough smooth when the magnification of images is too large, SRGAN uses GAN to generate appropriate edge information to improve image resolution. But the generated edge information is irrelevant to input. Hence, when we zoom in the results, we can observe that although the generated edges have a good holistic visual feeling in whole image, they are visually meaningless in small visual field.

Therefore, we propose ED-Block, which can extract edge information of input, to remain the relevance between edge information and input images. In this way, we need not to train a GAN network to generate edge information, which costs too much time expense and memory cost, and we can get highly relevant edge information, which will be added to output to improve image quality.

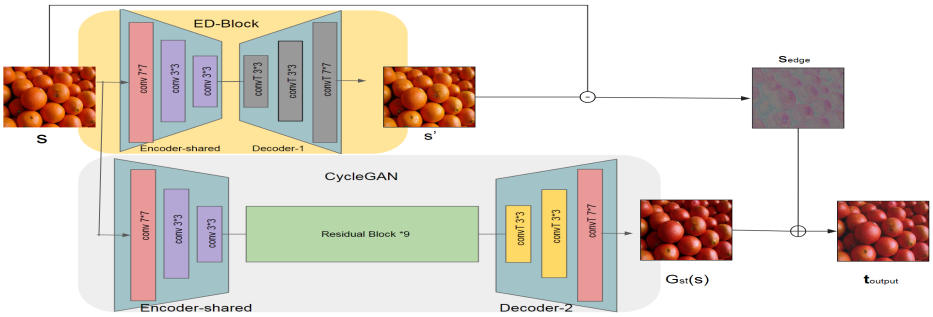

Fig. 1. The structure of our proposed network. The orange part is ED-Block consisting of only an encoder and an decoder. The gray part is original CycleGAN. ED-Block and CycleGAN share the same frozen encoder but their decoders are different.

3 Proposed method

Edge information will be lost in the encoder because of irreversible downsampling and cannot be fully restored in the decoder. The key to our idea is extracting edge map of original input by a Encoder-Decoder Block (ED-Block) and adding them to the output from generator to restore the edge information.

3.1 Encoder-Decoder Block

Structure of Encoder-Decoder Block The architecture of ED-Block is shown in Figure 1 and is adopted from the generator of CycleGAN. First, an input image will be processed by a convolution layer followed by an instance normalization layer to transfer from 3 to 64 channels and kernel size is 7×7. Then, the image will be shrunk two times in the encoder, of which the downsampling is realized by two convolution layers using stride 2. Both of them are followed by an instance normalization layer and a ReLu layer. In the decoder, feature maps are expanded by two transpose convolution layers to original size. Both of two layers are followed by an instance normalization and a ReLu layer. Finally, feature maps are processed by a convolution layer to transfer to 3 channels and the kernel size of the layer is also 7×7 as that in the encoder. After Tanh operation, the recovered image are output to be used in subtraction.

Training Encoder-Decoder Block ED-Block is trained in advance and then it will be frozen before training CycleGAN part. The work of ED-Block is to restore the input as much as possible even though restoring the original edge information is very difficult. In order to guarantee that the output is similar enough to input, we propose recover loss $\mathcal{L}_{recover}$ to train the ED-Block, which is L1 loss to measure the difference between recovered image s' and the original

image s. This process can be shown below:

$$\mathcal{L}_{recover}(s, s') = ||s - s'||_1 \tag{1}$$

After the ED-Block has been trained well, the main visual difference between output images from it and input images are the edges and boundaries. Therefore, if we subtract the output images s' to the original input s, we can get edge map s_{edge}, which saves the necessary edge information. This process can be shown as Eq.2.

$$s_{edge} = s - s' \tag{2}$$

The number of training ED-Block epoch is very significant. If the epoch is set too high, the edge map we get cannot has enough edge information. Since by more developed block, recovered images are more detailed and less information can be extracted by subtraction. Besides, if the epoch is set too small, it is difficult for block to reconstruct the image and some color information will be remained in edge map, which is not needed. we set the training epoch of ED-Block to 200 in our experiment and the reason will be illustrated in Sect.4.2.

Comparison with CycleGAN with Skip-Connection Skip-connection proposed in U-Net [9] realizes the effective use of feature maps of each layer in subsequent calculations by transmitting data of low-level layers directly, which improves accuracy of semantic segmentation. Therefore, CycleGAN with skip-connection can also avoid data lost caused by encoding process. However, the transmitted data in skip-connection channel not only includes edge information, but also includes color information of images. Because our ED-Block can selectively only extract edge information, the performance of CycleGAN with skip-connection cannot be better than ours. To make comparison, we design a CycleGAN with skip-connection structure, in which convolution layers of the encoder and the decoder with the same input sizes are connected by skip-connection channel to transmit data to the other end of network. Table 2, Table 3 and Table 4 show the evaluation scores of ours and CycleGAN with skip-connection, which are symbolized as "Ours" and "CycleGAN-Skip" separately. The output images of CycleGAN-Skip are shown in the Fig. 6.

3.2 Partly Frozen CycleGAN

Structure of Generator The generator has three parts: an encoder, a transformer and a decoder. The architecture is adopted from J. Johnson et al. [3]. Therefore, the encoder part is the same as that in Encoder-Decoder Block in order to share parameters while training. The transformer consists of nine residual blocks and each residual block consists of two stride-1 convolution layers, two instance normalization layers and a ReLu layer.

Structure of Discriminator For the discriminator networks, we use 16×16 PatchGANs [2] aiming to distinct whether 94×94 overlapping image patches are real or fake. This PatchGANs consists of five convolution layers using stride-4 followed by instance normalization and Leaky ReLu layers.

Training CycleGAN First, we apply the parameters of the pre-trained encoder of ED-Block to two generators in CycleGAN and freeze them. With the same frozen parameters in the encoder, the lost edge information are the same in both ED-Block and the generator. Therefore, the extracted edge information from ED-Block can have high relevance to the output from the generator. The initialization of rest parameters of the other layers are using Gaussian distribution $N(0, 0.02)$.

We adopt adversarial losses [1] and cycle consistency loss [10] to train CycleGAN. For adversarial loss, G_{st} aims to transfer image s of source domain into image $G_{st}(s)$ of target domain to fake D_T, while D_T aims to correctly identify the real image t and fake image, which is integration of G_{st}(s) and s_{edge} calculated by Eq.2. Eq.3 is the adversarial loss, which G_{st} tries to minimize but D_T tries to maximize.

$$\mathcal{L}^s_{adv}(G_{st}, D_T) = \mathbb{E}_{t\sim P_T(t)}[\log D_T(t)] + \mathbb{E}_{s\sim P_S(s)}[\log(1 - D_T(G_{st}(s) + s_{edge}))] \quad (3)$$

According to Jun-Yan Zhu et al. [10], input s can be mapped to any random permutation of images since it is hard to guarantee the learned mapping function only with adversarial loss. Therefore, cycle consistency loss is used to further reduce the space of possible mapping functions. G_{st} aims to transfer input to target domain and G_{ts} aims to transfer fake images, which combines $G_{st}(s)$ with s_{edge}, back to source domain,which is shown in Eq.4. The final recovered image s'' will be calculated with original input s to get the difference to back propagation. This process is shown in Eq.5.

$$s'' = G_{ts}\big(G_{st}(s) + s_{edge}\big) + s_{edge} \quad (4)$$

$$\mathcal{L}^s_{cyc}(s, s'') = ||s - s''||_1 \quad (5)$$

Moreover, s'' is got from G_{ts} plus s_{edge} rather than plus t_{edge} in Eq.4. Because s_{edge} is extracted from input s and t_{edge} is from the generated image in the target domain, the edge information of s_{edge} is better than that of t_{edge}. Hence, we choose s_{edge} rather than t_{edge} in cycle consistency process. Besides, we shows the evaluation scores of using s_{edge} and t_{edge} separately, which are symbolized as "Ours" and "Ours (with t_{edge})" in Table 2, Table 3 and Table 4. The output images of "Ours (with t_{edge})" are shown in the Fig. 6.

Finally, we combine the adversarial and cycle-consistency losses for both source and target domains to optimize the final energy, which is shown below:

$$\mathcal{L}(G_{st}, G_{ts}, D_s, D_t) = \mathcal{L}^s_{adv} + \mathcal{L}^t_{adv} + \lambda(\mathcal{L}^s_{cyc} + \mathcal{L}^t_{cyc}) \quad (6)$$

We use the loss hyper-parameters λ=10 in our experiments. The optimal parameters of $\mathcal{L}$ are obtained by solving the minimax optimization problem:

$$G^*_{st}, G^*_{ts}, D^*_s, D^*_t = \underset{G^*_{st}, G^*_{ts}}{argmin}\Big(\underset{D^*_s, D^*_t}{argmax}\mathcal{L}\big(G^*_{st}, G^*_{ts}, D^*_s, D^*_t\big)\Big) \quad (7)$$

4 Experiments

4.1 Training setting

The usage of s_{edge} need high texture similarity between output and input. Therefore, the domain translation should not translate the object textures and edges but only translate color. To ensure this premise, we use "Apple to Orange" and "Summer to Winter" datasets provided by Jun-Yan Zhu et al. [10] and "blond-hair2brown-hair" datasets including about 3000 blond and brown hair images from celebA dataset provided by [8] to train models and we adopt CycleGAN's notation to illustrate our structures. For example, "c7s1-k-R" means a 7×7 convolution layer with stride 1 and k filters, followed by a ReLU activation and "ct3s2-k-LR" means a 3×3 transposed convolution layer with stride 2 and k filters, followed by a LeakyReLU activation with slop 0.2. "rk" denotes a residual block with k filters. A tanh activation is indicated by 'T'. Moreover, we apply instance normalization after all convolution layers and transposed convolution layers except the first convolution layer of discriminator and the last convolution layers of decoder and discriminator. The structures of our model are shown below:

ED-Block structure is: c7s1-64-R, c3s2-128-R, c3s2-256-R, ct3s2-128-R, ct3s2-64-R, c7s1-3-T.

Residual Block structure is: c3s1-256-R, c3s1-256

Generator structure is: c7s1-64-R, c3s2-128-R, c3s2-256-R, r256, r256, r256, r256, r256, r256, r256, r256, r256, ct3s2-128-R, ct3s2-64-R, c7s1-3-T.

Discriminator structure is: c4s2-64-LR, c4s2-128-LR, c4s2-256-LR, c4s2-512-LR, c4s1-1

All batch sizes of ours and other models for comparison are 4 and we use Adam solver [5] to optimize parameters and initial learning rate is 0.0002, which will decrease in each epoch after 100 epochs. We use GPU Intel Core i7-4790(3.60GHz) to train models.

4.2 Training Epoch of ED-Block

The epoch of training ED-Block should be proper to reconstructed the input image better with smaller loss $\mathcal{L}_{recover}$ and less time expense. Figure 2 shows the generated results when the number of training epochs are 10, 50, 100, 200, and 500. We can observe that in the results of 10, 50, 100 epochs, there are a little needless remaining color information and its amount decreases by epochs. In the edge maps of 200 and 500 epochs, the color information is eliminated. Therefore, the ED-Block should be trained in enough epochs to get rid of color influence. Figure 3 shows the graph of $\mathcal{L}_{recover}$ during training. We can observe

Fig. 2. The results of ED-Block by different training epochs. In the lower training epochs, such as 10, 50, 100 epochs, color information remains in edge map. The higher training epochs, the less color residues. In 200 and 500 epochs, remained color information is not visually obvious.

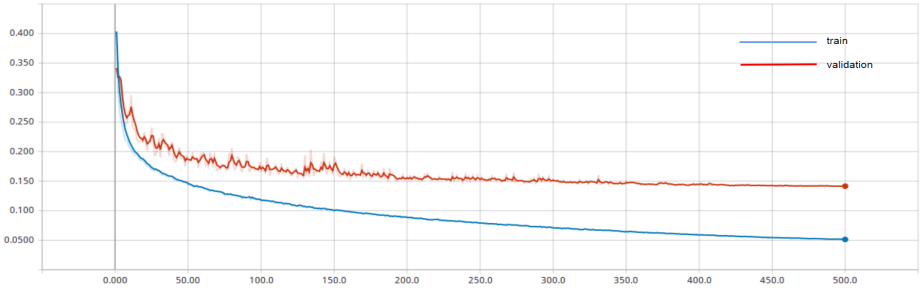

Fig. 3. The graph of $\mathcal{L}_{recover}$ during 500 training epochs. Loss function drops rapidly in first 100 epochs and tends to be gentle later.

that $\mathcal{L}_{recover}$ in 500 epochs is only a little lower than that in 200 epochs but with much higher time expense. Based on Figure 2 and Figure 3, we set training epoch of ED-Block to 200.

4.3 Results

Figure 4 represent the generator loss decrease during 200 training epochs. Because the encoder is pre-trained and frozen and only the transformer and the decoder part need to be trained, our models has much faster convergence speed compared with CycleGAN. Figure 5 shows the results from ours, original CycleGAN, Unit [7] and DiscoGAN [4] in 200 training epochs. Figure 6 shows the results from from our model, our model using t_{edge} in cycle consistency equation, CycleGAN and CycleGAN with skip-connection channels. We can observe that our model remains much more edge information so that the results are far clearer and have more texture compared with other results, which are obviously blurry. Therefore, our network has achieved a great progress in improving the image quality.

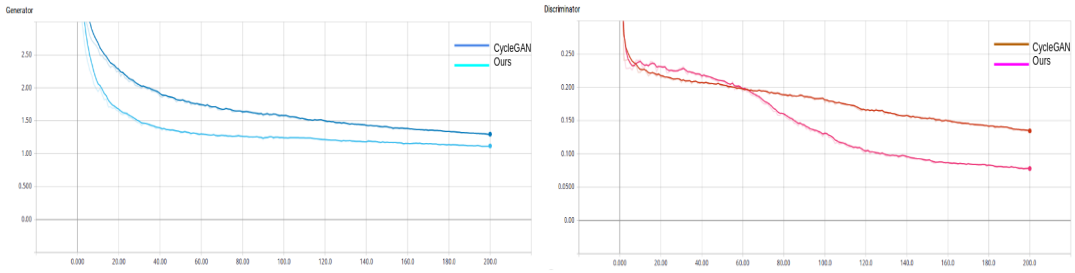

Fig. 4. The left side graph shows loss value of two generators and right side graph shows loss value of two discriminator during 200 epochs.

4.4 Evaluation and Discussion

Table 1 shows the comparison on quantity of parameters and time training expense on "Apple2Orange" dataset. The amount of parameters of our model is same as CycleGAN and much less than DiscoGAN while Unit has least parameters. The time expenses of ours, CycleGAN and DiscoGAN do not have large difference and much less than that of Unit. To summarize, our model has a compromise between the amount of parameters and training time expense.

Table 1. Models Comparison.

Model	Parameters (10^5)	Time expense (hours)
ED-Block	7.56	1.533
Our-CycleGAN	275.30	16.898
CycleGAN	282.86	15.35
Unit	270.66	21.198
Disco	598.1	16.844

We use SSIM and PSNR, of which input are real images and cycle consistency image shown in Fig. 7 and Fig. 8, to evaluate image resolution. Table 2, Table 3 and Table 4 show the SSIM and PSNR scores of Ours, Ours (with t_{edge}), CycleGAN, CycleGAN-Skip, Unit [7] and DiscoGAN [4] after 200 training epochs on "Apple2Orange", "Summer2Winter" and "blond-hair2brown-hair" datasets.

Our model achieves the highest scores among them except PSNR score of consistent-orange. Ours (with t_{edge}) achieves the second highest SSIM scores on "apple2orange" and "Summer2Winter" datasets and only a little lower than CycleGAN on "blond-hair2brown-hair" dataset but PSNR scores are not ideal. Since PSNR realizes image evaluation based on the mean-square error between corresponding pixels and does not take into account the visual characteristics of human eyes, the ability to capture perceptually relevant differences is very limited and it is acceptable that the score is inconsistent with people's subjective feelings in some cases. But in Table 7 and Table 8, we can observe that

Table 2. SSIM and PSNR scores on "Apple2Orange" dataset.(200 epochs)

Model	SSIM		PSNR(dB)	
	consistent-apple	consistent-orange	consistent-apple	consistent-orange
Ours	**0.8029**	**0.7503**	**19.266**	17.068
Ours (with t_{edge})	0.7535	0.7021	16.764	15.582
CycleGAN	0.7329	0.6927	19.035	17.985
CycleGAN-Skip	0.7412	0.7061	18.654	17.743
Unit	0.6948	0.6705	18.449	**18.297**
Disco	0.4403	0.4107	13.424	14.462

Table 3. SSIM and PSNR scores on "Summer2Winter" dataset.(200 epochs)

Model	SSIM		PSNR(dB)	
	consistent-summer	consistent-winter	consistent-summer	consistent-winter
Ours	**0.8410**	**0.8318**	**21.267**	**20.622**
Ours (with t_{edge})	0.8059	0.8089	19.433	19.314
CycleGAN	0.7842	0.7911	20.259	20.013
CycleGAN-Skip	0.7726	0.7850	19.903	20.054
Unit	0.7025	0.7188	19.031	18.878
Disco	0.6688	0.6699	18.765	18.124

Table 4. SSIM and PSNR scores on "Blond-hair2Brown-hair" dataset.(200 epochs)

Model	SSIM		PSNR(dB)	
	consistent-blond	consistent-brown	consistent-blond	consistent-brown
Ours	**0.8682**	**0.8903**	**24.679**	**24.744**
Ours (with t_{edge})	0.8189	0.8283	22.188	20.731
CycleGAN	0.8222	0.8554	22.799	23.553
CycleGAN-Skip	0.6821	0.6910	17.028	16.206
Unit	0.7889	0.8046	22.521	22.583
Disco	0.7637	0.7924	20.539	21.248

our cycle-consistency orange image is really clearer than those of other models. CycleGAN and CycleGAN-Skip get similar evaluation scores on "apple2orange" and "summer2winter" datasets but have a large disparity on "blond-hair2brown-hair" dataset, which illustrates that only adding skip-connection channel to generators cannot improve images quality greatly since transmitted data consists of not only edge information but also color information. The transmitted color information even have a bad perceptual effect on cycle consistent images to cause low evaluation score. Finally, DiscoGAN obtains the lowest scores among them on "apple2orange" and "Summer2Winter" datasets and also performs not well on "blond-hair2brown-hair" dataset because of deeper convolution layers in encoder without any solution to protect edge information. To summarize, our model have achieved the highest evaluation scores and our results are visually excellent.

5 Conclusions

In this paper, we proposed ED-CycleGAN to improve the image quality of CycleGAN. The ED-Block extracts edge maps of input firstly to prevent edge information from being destroyed in the encoder processing. Then two generators of CycleGAN share the pre-trained and frozen encoder of ED-Block during training. And finally processed images are integrated with edge maps as final outputs of generator. Our model, ED-CycleGAN, improves the image quality of generators with less time expense and get highest SSIM and PSNR scores compared with CycleGAN, Unit and DiscoGAN.

References

1. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 2672–2680. Curran Associates, Inc. (2014)
2. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on (2017)
3. Johnson, J., Alahi, A., Li, F.: Perceptual losses for real-time style transfer and super-resolution. ECCV **abs/1603.08155** (2016)
4. 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 - Volume 70. pp. 1857–1865. ICML'17, JMLR.org (2017)
5. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
6. Ledig, C., Theis, L., Huszar, F., Caballero, J., Aitken, A.P., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. CVPR (2017)
7. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 700–708. Curran Associates, Inc. (2017)
8. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV) (December 2015)
9. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. pp. 234–241. Springer International Publishing, Cham (2015)
10. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networkss. In: Computer Vision (ICCV), 2017 IEEE International Conference on (2017)

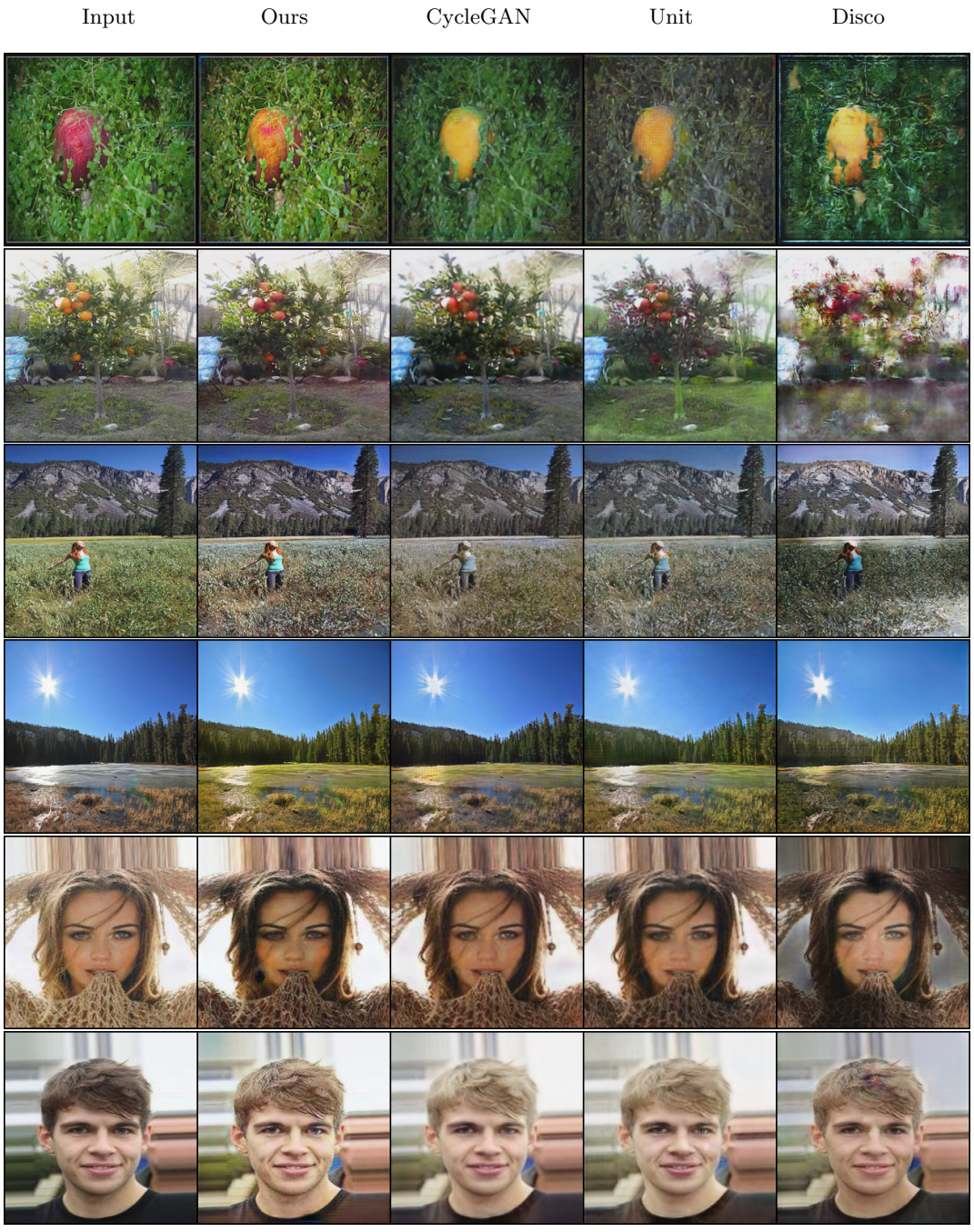

Fig. 5. Image translation results from "Apple2Orange", "Summer2Winter" and "Blond-hair2Brown-hair" datasets in 200 epochs. From top to bottom is apple to orange, orange to apple, summer to winter, winter to summer, blond hair to brown hair and brown hair to blond hair.

Fig. 6. Image translation results from "Apple2Orange", "Summer2Winter" and "Blond-hair2Brown-hair" dataset in 200 epochs. From top to bottom is apple to orange, orange to apple, summer to winter, winter to summer, blond hair to brown hair and brown hair to blond hair.

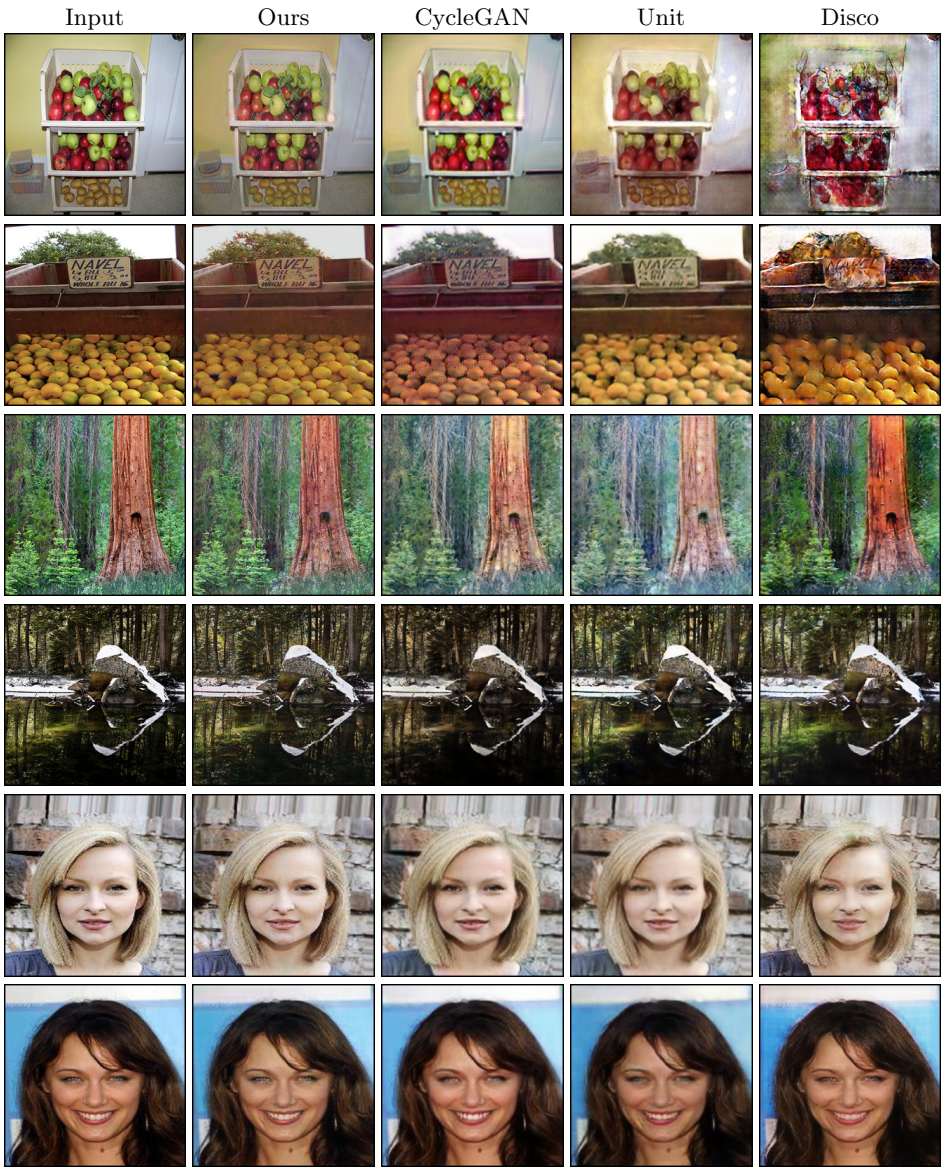

Fig. 7. The cycle consistency images in 200 epochs of Ours, CycleGAN, Unit and DiscoGAN. The results from our models has less artifacts and are much clearer and more similar to input.

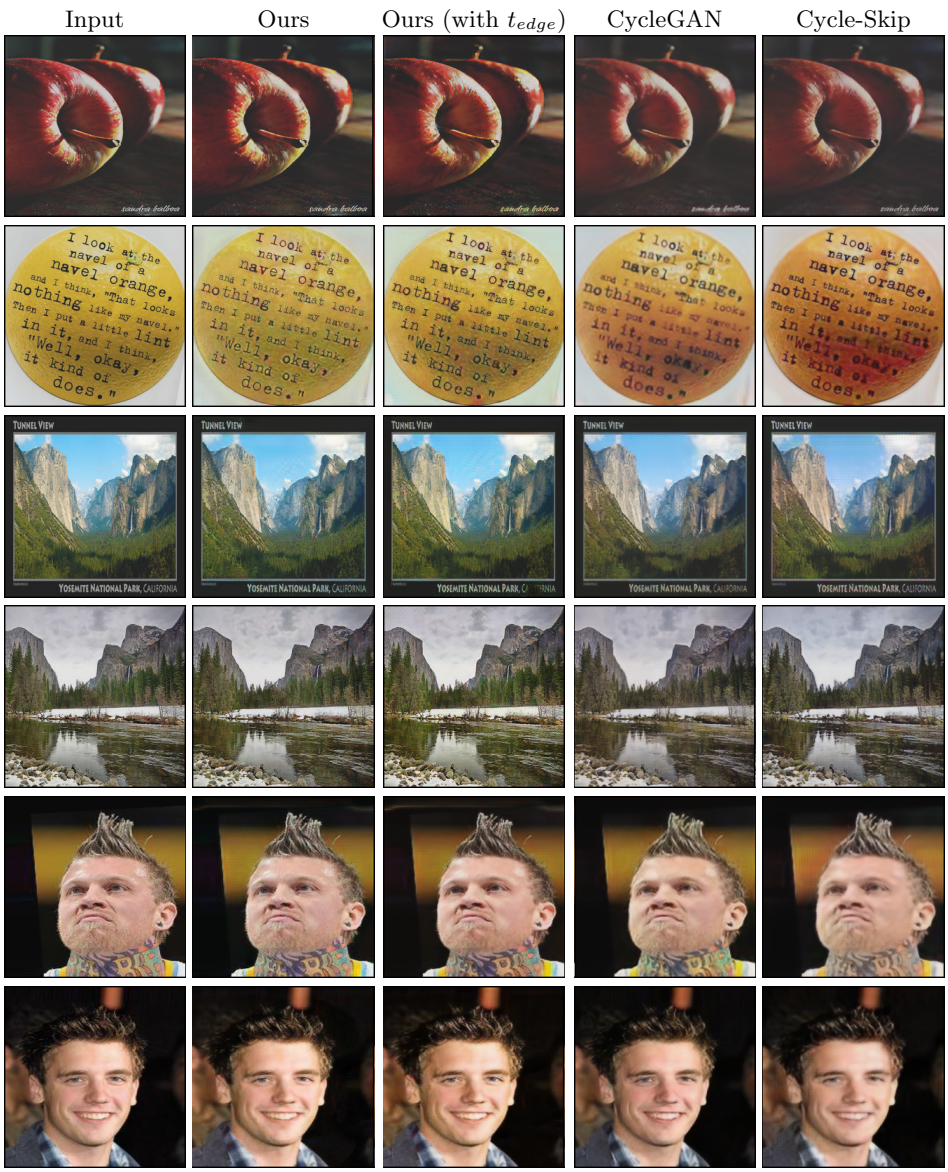

Fig. 8. The cycle consistency images in 200 epochs of Ours, Ours (with t_{edge}), CycleGAN, and CycleGAN with skip-connection. The results from our models has less artifacts and are much clearer and more similar to input.