Image Season Transitions using GANs

1st P.W.O. van Aken
Graduate School of Natural Sciences
Utrecht University
Utrecht, The Netherlands
p.w.o.vanaken@students.uu.nl - 6208878

2nd F.L.G. Blom
Graduate School of Natural Sciences
Utrecht University
Utrecht, The Netherlands
f.l.g.blom@students.uu.nl — 5988918

3rd R.R. Cuevas
Graduate School of Natural Sciences
Utrecht University
Utrecht, The Netherlands
r.ricocuevas@students.uu.nl — 1243012

4th J.A.W. Markus
Graduate School of Natural Sciences
Utrecht University
Utrecht, The Netherlands
j.a.w.markus@students.uu.nl — 6134165

5th M.A. Scheeres
Graduate School of Natural Sciences
Utrecht University
Utrecht, The Netherlands
m.a.scheeres@students.uu.nl — 9640455

6th A. Tsiamis
Graduate School of Natural Sciences
Utrecht University
Utrecht, The Netherlands
a.tsiamis@students.uu.nl — 5223652

***Abstract*—This paper aims to improve the state of the art technology employed for solving the unpaired image-to-image translation problem in the domain of summer-to-winter images. To achieve this, a deep learning framework comprised of two generator and two discriminator models that compete against each other pairwise known as CycleGAN will be exploited. The current state of the art (CNNcycleGAN) corresponds to a setup where all of the four sub-models that together make up the CycleGAN architecture adopt a convolutional neural network (CNN) form. Instead, we propose using Vision Transformers (ViTs) as the discriminators in the global architecture. The reason behind this choice has to do with the fact that ViTs have been shown to outperform CNNs in image classification tasks. Performance-wise the training time of our proposed implementation (ViTcycleGAN) took 17% longer than CNNcycleGAN over the course of 5 epochs on the selected Yosemite dataset. However, we did observe better qualitative results in the test images translated by ViTcycleGAN. Images produced by ViTcycleGAN's generator generally look less synthetic (more realistic) than those produced by CNNcycleGAN's generator.**

***Index Terms*—Computer vision, Unsupervised image translation, CycleGAN, Convolutional Neural Network, Vision Transformer**

Supplementary material:
https://github.com/MatthewScheeres/PRFinalProject

I. Introduction

Computer vision has gained important ground in the last few years, contributing to projects ranging from automatic object detection in self-driving cars to face detection in smartphone cameras. According to Ballard & Brown "*Computer vision is the enterprise of automating and integrating a wide range of processes and representations used for vision perception*" [1].

An important subfield of computer vision is image-to-image translation. Image-to-image translation is defined as "*the task of translating one possible representation of a scene into another, given sufficient training data*" [7]. Image-to-image translation comes in two flavours, the supervised and the unsupervised setting. Years of research have produced very powerful image translation systems in the supervised setting like the Pix2Pix model [7]. In the supervised setting, a coupled image data set of the form

$$\{(x_1, y_1), \ldots, (x_N, y_N)\} \tag{1}$$

is available. Each instance of the labeled data set (1) is made up of an image x_i and its translation y_i. For example, when aiming to translate images from day to night, x_i could be a picture of Utrecht central station during the day and y_i would correspond to a picture of the exact same location taken at night. One might very well argue that gathering such a data set would be a very tedious task. In our example, one would have to take the first picture using a camera with a tripod during the day and then leave it at the exact same location until night time to take the next picture. Despite being an extremely time-consuming process it still theoretically doable. But what if we wanted to translate images of horses to images of zebras? In this scenario gathering a coupled data set would be impossible. We would be in the unsupervised setting and have two unlabeled data sets

$$\{x_1, \ldots, x_{N_1}\} \tag{2}$$

and

$$\{y_1, \ldots, y_{N_2}\}. \tag{3}$$

The first one (2) corresponding to N_1 horse pictures and the second one (3) made up of N_2 pictures of zebras. We note that due to the nature of the data, the unsupervised setting would be far more common than the supervised one. It was not until 2017 when the first deep learning model able to deal with unsupervised image-to-image translation (CycleGAN) was proposed [15].

In this project we present a modification of the current state of the art CycleGAN architecture to obtain an image translation system such that when given a landscape image taken during summertime is able to generate an image of the same landscape during wintertime and vice-versa in a more realistic way than the current state of the art. Such an image translation system could have several applications, for example as a "season-changing" tool for photo editing.

II. Related Work

The research conducted in this paper builds upon previous research done in the field of Generative Adversarial Networks (GANs) and image translation. In order to enable a thorough understanding of this research, it needs to be properly embedded into the relevant literature and related work.

The first implementation of a GAN model originates from research by Goodfellow et al. (2014). The research incorporates a setup consisting of two competing models; a generative model and a discriminative model. The goal of this setup is to force accurate results from the generative model by instructing the discriminative model to classify instances as part of the training data or as generated by the generative model. In turn, the generative model is instructed to deceive the discriminative model. This manner of back-and-forth training between the two models effectively creates a competition between the two, which theoretically should yield a high quality generative model which produces images that are not able to be distinguished as real or fake by the discriminator model. The results from Goodfellow et al.'s research suggest that the chosen model setup is to be considered a viable option [4]. More details will be provided in Section III.

An extension of the research by Goodfellow et al. (2014) was made by Mirza & Osindero (2014). The authors propose a new and more specific structure of the GAN architecture called conditional GAN (or cGAN). The main difference between a GAN and a cGAN lies in the fact that in a cGAN the model is conditioned on more information causing the model to be able to create a specific type of data. Mirza & Osindero attain this by using the label of a data instance as an additional parameter for the model. This ensures that the type of data that is being outputted by the model matches the type of data in the training set [10]. The additional information that is used to condition on can be of any kind and as we will later see it can even be an image from the domain.

Using GANs in the field of image-to-image translation has become increasingly popular in recent years. One of the dominant approaches for this specific task originates from research by Isola et al. (2016) [7] in which a 'Pix2Pix' approach is used. Isola et al. implement a conditional generative adversarial network to train a generative model that maps input images to output images. The mapping between the in- and output is learned using a training set of aligned image pairs. The results from Isola et al.'s research suggest that the chosen approach significantly outperforms other approaches (for the same task).

An issue with the image-to-image translation model applied by Isola et al. is the requirement of a training dataset consisting of paired examples. As was mentioned in the introduction, this is a limitation given the fact that these examples are either difficult or impossible to gather. Therefore, attempts have been made to create a model for unpaired image translation, i.e. the setting without a supervised training data set. In this new context, the goal becomes figuring out the characteristics of the input domain and learning how these translate and relate to the output domain. Rosales et al. proposed a method which uses a Bayesian framework for inferring the most likely output image. The translation technique involves the consideration of a prior probability on the output that corresponds to a patch-based Markov random field obtained from the input [12].

Another successful method suitable for unpaired image-to-image translation based on the GAN architecture was proposed by Zhu et al., who developed an extension of the GAN architecture called CycleGAN that works for the unpaired translation problem. Zhu et al. tested the CycleGAN architecture on several types of image translation problems, including paintings to photos, maps to aerial photos and summer to winter landscapes. The quantitative analysis of their results suggests that the CycleGAN model clearly outperforms other previous techniques developed for unpaired image translation problems [15]. More details about the CycleGAN architecture will be provided in the following section.

III. Methodology

A. Generative Adversarial Networks (GANs)

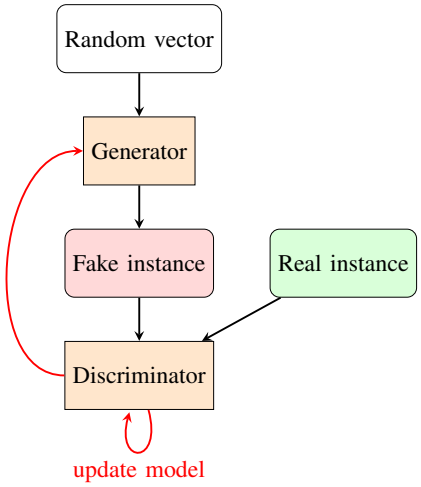
The GAN architecture provides a deep learning framework to solve a set of problems called *unsupervised generative modeling* problems. Given an unlabeled data set, generative modeling refers to the process of automatically generating new plausible data instances. The GAN architecture allows the training of a generative model by essentially transforming the unsupervised problem to a supervised one. This is achieved by combining two types of networks, which together make up the GAN:

1) *The generator model:* A Neural Network (NN) model (usually a type of Convolutional Neural Network (CNN) when dealing with image data [5]) that generates new plausible examples from the domain of the data that it is trained with.
2) *The discriminator model:* A classifier that tries to find out whether the input data it is given is real or fake (synthetically generated). Like the generator model, the discriminator usually also employs a CNN architecture.

These two models compete against each other during training, each using the other's outputs to continuously update its own 'strategy' in either generating synthesized outputs, or labeling them as real or fake. This is where the model's name comes from; the generator is the discriminator's adversary, and vice versa.

As such, these models work against each other in tandem, each strengthening the other model turn after turn, until the generator creates synthetic instances that are (nearly) indistinguishable from the original ones. This is done as in the following way; the generator produces samples, taking a random vector of inputs (usually drawn from a Gaussian distribution) used to seed the generative process. Afterwards, these generated samples together with instances from the training data set are provided to the discriminator model to be classified as real or fake. After doing so, depending on the results of the discriminator, the weights of the models are carefully nudged following a zero-sum game. This means

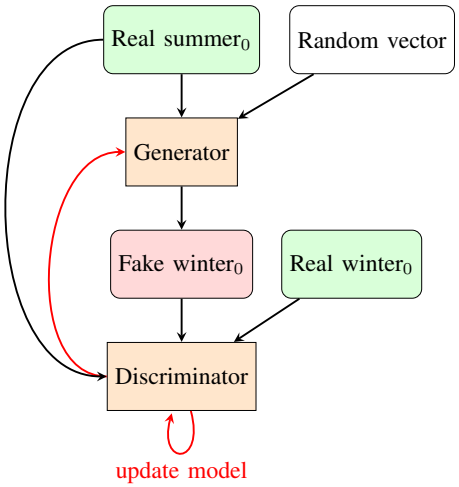
that if the generator fooled the discriminator, the discriminator gets punished and its weights get severely modified while the generator remains untouched. This also works the other way around, i.e. if the generator was not able to fool the discriminator, then the generator gets its weights strongly nudged while the discriminator remains untouched. This process can be visualized as follows:



The loop in this process is repeated until the discriminator model can't discern generated examples from real ones. At this point, the discriminator model is discarded and the generator is kept. After the training, the generator should be able to produce instances of data from the domain of the training data set that are nearly indistinguishable from the original ones.

B. Conditional GANs (cGANs)

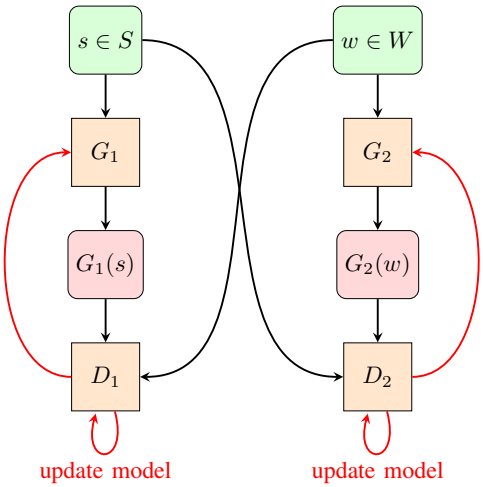
It is important to note that GANs can be used to generate data instances that very closely resemble the ones in the used training data set, but can only do so in a purely random way. An extension to the GAN architecture that addresses this issue is cGAN. In this case, the generator model is provided input that is conditioned by some additional information about the input (like an image or text data). Here, the discriminator is conditioned in the same way, i.e. it's provided with an instance that is either real or fake, together with the mentioned additional information. This way, a cGAN can generate examples from a domain in a controlled way. The information we choose to condition on can be of diverse nature. In image-to-image translation problems, the conditioning is usually performed on an image from the domain. In our specific domain, transforming summer to winter, the generator is provided a random vector as well as real summer photos as input. Similarly, the discriminator is provided examples of real and generated winter photos as well as real summer photos as input. More precisely, it's given a real summer image and a real or generated winter paired image, and must determine whether the paired image is real or fake. Therefore, the generator model is not only trained to fool the discriminator model but also to minimize the loss between the generated image and the expected target image as well. Graphically, the architecture would adopt the following form:



Note that this approach requires a coupled data set.

C. CycleGAN

As noted by Zhu et al., the the CycleGAN architecture is an extension to the GAN architecture that aims to capture characteristics of the types of image collections, and finds out how these characteristics could optimally be applied to transform new images, without the burden of needing a data set of paired images [15]. The CycleGAN architecture is comprised of two GANs which will be denoted by the names GAN_1 and GAN_2. In turn, these are made up of a generator model and a discriminator one, i.e. G_1, D_1 and G_2, D_2 respectively. In our specific domain, GAN_1 will translate summer photos (S) to winter photos (W) and GAN_2 will do this in reverse. Thus, G_1 will take summer photos and produce winter ones and G_2 and D_2 will do the opposite task, i.e., $G_1 : S \longmapsto W$, and $G_2 : W \longmapsto S$. Additionally, the discriminator D_1 aims to distinguish between winter images $w \in W$ and translated summer images $G_1(s)$; and the discriminator D_2 aims to discriminate between summer images s and translated winter images $G_2(w)$. Graphically, the CycleGAN architecture adopts the following structure (Note that the random vector feed into the generators has been omitted for visual clarity).



We want to learn the generators and discriminators given unlabeled data sets $\{s_i\}_{i=1}^{N} \subset S$ and $\{w_j\}_{j=1}^{M} \subset W$. The discriminators D_1, D_2 and generators G_1, G_2 are trained under the usual adversarial loss like in the standard GAN model (red arrows in the diagram). Mathematically, this is represented by the following adversarial loss terms:

$$\begin{aligned}\mathcal{L}_{GAN}(G_1, D_2, S, W) = \mathbb{E}_{p_W(w)}[log(D_2(w))] \quad + \\ \mathbb{E}_{p_S(s)}[log(1 - D_2((G_1(s)))],\\ \mathcal{L}_{GAN}(G_2, D_1, W, S) = \mathbb{E}_{p_S(s)}[log(D_1(s))] \quad + \\ \mathbb{E}_{p_W(w)}[log(1 - D_1((G_2(w)))],\end{aligned}$$

where $p_S(s)$ and $p_W(w)$ denote the summer and winter data distributions respectively. Discriminator D_2 aims to minimize the first objective against generator G_1 that tries to maximize it, i.e.,

$$\min_{G_1} \max_{D_2} \mathcal{L}_{GAN}(G_1, D_2, S, W).$$

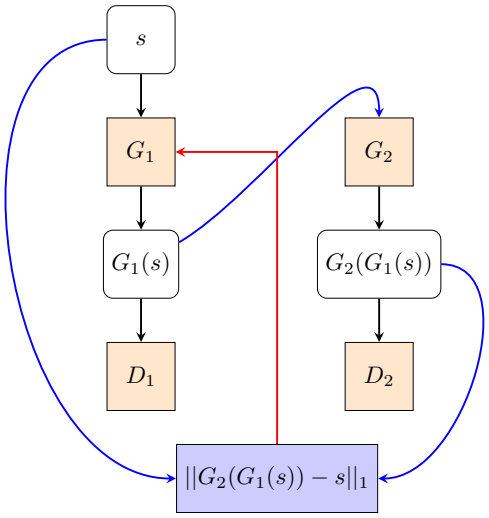
Similarly, discriminator D_1 aims to minimize the first objective against generator G_1 that tries to maximize it, i.e.,

$$\min_{G_2} \max_{D_1} \mathcal{L}_{GAN}(G_2, D_1, W, S).$$

It is important to remark that the model cannot exploit the cGAN architecture to produce translations of the input images because of the lack of a coupled data set. To make up for this, the concept of *cycle consistency*, originally employed in machine translation, is used instead. As we just mentioned, the discriminators D_1, D_2 and generators G_1, G_2 are trained under the usual adversarial loss like in the standard GAN model, but to make up for the lack of a labeled data set an additional loss measure is added, the *cycle consistency loss*. This is intended to force the generated images to be translations of the input images. In order to achieve this, we enforce G_1 and G_2 to be *cycle consistent*, i.e. for any summer image s and winter image w, we should have $G_2(G_1(s)) = s$ and $G_1(G_2(w)) = w$ respectively. Therefore the cycle consistency loss adopts the form:

$$\begin{aligned}\mathcal{L}_{cyc}(G_1, G_2) = \mathbb{E}_{p_S(s)}[G_2(G_1(s)) - s] \quad + \\ \mathbb{E}_{p_W(w)}[G_1(G_2(w)) - w].\end{aligned}$$

As we observe, the cycle consistency loss is composed by two terms. In the first one, GAN_1 receives a summer image s and generates a winter image $G_1(s)$. $G_1(s)$ is fed as input to GAN_2, which generates an summer image $G_2(G_1(s))$. The first term of the cycle consistency loss is then computed as the difference between $G_2(G_1(s))$ and s (in L_1 norm for example). Graphically:



The second one is symmetric to the first but the roles of GAN_1 and GAN_2 are swapped. Combining all the pieces together, the full objective becomes

$$\begin{aligned}\mathcal{L}(G_1, G_2, D_1, D_2) = &\mathcal{L}_{GAN}(G_1, D_2, S, W) \quad + \\ &\mathcal{L}_{GAN}(G_2, D_1, W, S) \quad + \\ &\lambda \mathcal{L}_{cyc}(G_1, G_2),\end{aligned}$$

where λ controls the importance of the cycle-consistency term and we will set it to 10 as in the work by Zhu et al. [15]. Therefore, we want to find

$$G_1^*, G_2^* = \arg \min_{G_1, G_2} \max_{D_1, D_2} \mathcal{L}(G_1, G_2, D_1, D_2).$$

D. Vision Transformers

Vision Transformers (ViTs) are a relatively new type of transformers that are mainly applied to tasks that involve vision processing, like image recognition. ViTs were first proposed in 2021 ([17], [21]) as an extension of 'regular' transformers (which had mainly been used for NLP tasks since their introduction in 2017 [18]).

Transformers introduced the concept of self-attention, which ViTs use to capture relationships between different *patches*, or small segmented portions of the image. This is the main addition Visual Transformers make, and it has been shown that ViTs can achieve performance comparable or superior to CNNs in image classification [20]. As we will see next, ViTs are used in this research to provide a new implementation that aims to improve the original CycleGAN model proposed by Zhu et al. [15].

E. Experimental Setup

The experimental setup of this paper's research will compare two different implementations of the CycleGAN architecture. These are distinctly differing in the way the discriminators are structured. In the first implementation (CNNcycleGAN), standard multi-layer CNNs are used as the discriminator models (replicating the work proposed by Zhu et al. [15]). In the second implementation (ViTcycleGAN), which we propose in this paper, the Visual Transformer (ViT)

originally proposed in [17] is used as the discriminator models of the CycleGAN architecture. It is important to note that both discriminator models described in this paper were optimized using the least squares loss as opposed to the negative log likelihood since it has been shown to achieve a more stable training [15].

1) CNNcycleGAN Implementation: The CNNcycleGAN implementation considers a CNN structure for D_1, D_2, G_1 and G_2, originally described by Brownlee [16]. The generators G_1 and G_2 are two CNNs each of which consists of three convolutional layers, followed by nine residual blocks, two deconvolution layers, and one final convolutional layer (for an in-depth description of the conv net, please refer to A). The kernels of all layers were initialized using a Gaussian distribution with $\sigma = 0.02$.

The discriminators D_1 and D_2 are two CNNs each of which consists of five convolutional layers, after which the output is patched into a final convolutional layer without an activation function. The specific architecture is also described in A.

2) ViTcycleGAN Implementation: The implementation we propose in this paper differs from the CNNcycleGAN implementation by incorporating ViTs as the architecture discriminators instead of CNNs. The main reason behind this, like mentioned in III-D, is that Vision Transformers have been shown to be comparable and even outperform in some cases CNNs when it comes to image classification. The expected result is thus a slight improvement in the the final performance of the generator models corresponding to the ViTcycleGAN implementation. This has to do with the fact that even though booth implementations share the same generators, ViTs, given that they are better image classifiers, will be generally harder to fool than CNNs. This forces the ViTcycleGAN's generators to produce images that would look more realistic or, equivalently, less synthetic.

F. Training Setup

Both networks were trained on a windows machine on a Gigabyte NVIDIA 2080 SUPER GPU with a default 1650 MHz GPU clock speed (1845 HMz boost clock) and 1938 MHz Memory clock. Using the CUDA 11.2 toolkit and cuDNN 8.1.1 library. The `Yosemite dataset` was used to train both CycleGAN implementations. It is comprised of 2731 unique pictures of the Yosemite National Park in California. The dataset is divided into four folders (*testA, testB, trainA, trainB*), with summer pictures being located in the *A*-folders, and winter pictures in the *B*-folders. Both CycleGAN models were trained for 5 epochs, for a total of 7700 training iterations. These models were then compared on both training performance and translation quality.

G. Evaluation Metrics

For the evaluation of the result we will firstly examine and discuss a sample of the translated test images. Besides this discussion we will run an experiment to quantitatively analyze the results and the difference between the generative models. This experiment will be based on the protocol used by Zhu et al. and Isola et al. (2016) and consists of a perceptual study [15] [7]. In the experiment we will gather data from 20 participants. Each participant will be shown two generated images simultaneously one generated by the CNNcycleGAN and the other one generated by the ViTcycleGAN model. Consequently, the participants will be asked to select the image that looks most realistic. Every participant is shown 10 pairs of images (5 summer-to-winter generated images and 5 winter-to-summer images). The order with which the images are shown is random for each participant. Given the fact that in this study we use a different dataset than Zhu et al., the results of this evaluation cannot directly be compared to those results. Therefore, this evaluation will only provide an insight into the relative performance of the two models.

IV. Results

A. Training performance

The CNNcycleGAN implementation finished 5 epochs of training in 12804.07 seconds while on average utilising 80% of the GPU's computational power. Our ViTcyleGAN implementation finished the 5 epochs of training in a longer period of 14996.64 seconds while on average utilising 70% of the GPU's computational power.

In the ViTcycleGAN implementation we see a $\approx$ 17% increase in training time accompanied by a 12.5% decrease in GPU utilisation. In the first place, the increase in training time could be explained by the fact that since ViTs are better classifiers than CNNs, at every training iteration the network parameters are more severely modified therefore taking more time. In the second place, the decrease in GPU utilisation has to do with the fact that the CycleGAN is not as optimized for training in the with ViTs present as with CNNs.

B. Translation Quality

For the following examples of the CNNcycleGAN and ViTcycleGAN quality-performance comparison we have cherry picked one test image per domain translation that we think highlights the qualitative differences between the networks.

Fig. 1. Summer $\rightarrow$ Winter: original image (Left), CNNcycleGAN (middle), ViTcycleGAN (right)

Figure 1 shows the difference between the original, CNNcycleGAN, and ViTcycleGAN networks respectively. Both networks seem to have learned to translate the foliage colour from a green to a brown colour, as would typically be the case in winter scenery. We think that the ViTcycleGAN network yields the better results in this regard. Both networks appear to attempt to add some sort of snow/frost to the top of the bushes

at the edge of the water, which we judge to be more realistic in the ViTcycleGAN network, although this might be due to the already more believable colouring of the vegetation. A more objective aspect would be the amount of artefacts present in the CNNcycleGAN image (better visible in the Appendix's 3). Here we see certain discolourations/patterns not present in the original image, which the ViTcycleGAN image also lacks. We assume this might be a product of insufficient training, but the fact that such artefacts are absent in the ViTcycleGAN picture would imply that in the ViTcycleGAN network, the ViT was able to more efficiently detect and in turn train the generator to not produce such artefacts at an earlier stage of training.

Fig. 2. Winter → Summer: original image (Left), CNNcycleGAN (middle), ViTcycleGAN (right)

Figure 2 shows the translation of the winter to summer domain. Here we can see an interesting divergence in the strategy with which the two networks translate an image. The CNNcycleGAN network approaches the translation by primarily painting the foliage a more vibrant green colour, this might in turn be enough to trick the (CNN) discriminator to classify the image as summer. The ViTcycleGAN network has not (yet) learned to associate a greener foliage colour with the summer season. The ViTcycleGAN generator instead appears to attempt to remove/reduce the amount of snow in the image, this can be seen by the colour transition that appears more pronounced on the snow-covered trees than the background trees. As well as the log on the riverbank to the right of the image, in the original image both the riverbank and the log are covered in snow. In the CNNcycleGAN image this snow has increased in brightness as if the image is overexposed, leaving little trace of the log in the image. In the ViTcycleGAN image the snow still suffers from a high brightness, but we can detect a larger part of the log (that hasn't been covered in snow) compared to the CNNcycleGAN image.

C. Results Perceptual Study

Model	**Summer-To-Winter** *Labeled More Realistic*	**Winter-To-Summer** *Labeled More Realistic*
CNNcycleGAN	22%	42%
ViTcycleGAN	78%	58%

TABLE I
RESULTS OF PERCEPTUAL STUDY

The results of the perceptual study, which setup is discussed in section III are presented in the table above. The findings of the perceptual study suggest that the images generated by the ViTcycleGAN are deemed more realistic in both the summer-to-winter and the winter-to-summer translation settings. However, the difference in score is higher for the summer-to-winter translations. This causes us to conclude that the ViTcycleGAN substantially outperforms the CNNCycleGAN when it comes to summer-to-winter translations and that the difference between the performance of the model is less pronounced on winter-to-summer translations.

V. Discussion

In the comparison of the CNNcycleGAN and ViTcycleGAN we have found that both networks perform similarly on a large part of the training data, but on a select number of images the ViTcycleGAN appears to have learned a more desirable translation model.

It is important to appreciate the fact that these networks are severely under-trained with only 5 epochs, as can be seen from the still very high variability of the reported losses. This means that translations made with our current CycleGAN models might not hold for training sessions featuring more epochs, as both networks will improve their generator models.

A. Future Work

In future research we aim to train both networks for a longer period, either in the form of a fixed number of epochs and comparing the image quality, as we did in section IV-B. Or by letting the networks run until they reach a certain loss threshold and comparing the time difference between training sessions.

VI. Conclusion

Image-to-image translation is a very exciting domain whitin the ever-growing field of computer vision. In this project, we trained two two different implementations of the CycleGan architecture in order to translate summer to winter images and vice-versa. Our proposed setup that incorporates ViTs as part of the CycleGAN architecture showed to qualitatively and quantitatively outperform the current state of the art in the translation task. These results pave the way for more research in this field in the future.

References

[1] D. H. Ballard and C. M. Brown, "Computer vision". Prentice-Hall, 1982.

[2] Q. Chen and V. Koltunn, "Photographic image synthesis with cascaded refinement networks," Proceedings of the IEEE international conference on computer vision, pp.1511–1520, 2017.

[3] E. Denton and S. Chintala and A. Szlam and R. Fergus, "Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks," arXiv: 1506.05751, 2015.

[4] I. Goodfellow and J. Pouget-Abadie and M. Mirza and B. Xu and D. Warde-Farley and S. Ozair and A. Courville and Y. Bengio, " Generative adversarial nets," Advances in neural information processing systems, Volume 27, 2014.

[5] G. James and D. Witten and T. Hastie and R. Tibshirani. An Introduction to Statistical Learning : with Applications in R. New York : Springer, 2013.

[6] L. Hsin-Ying and T. Hung-Yu and H. Jia-Bin and S. Maneesh and Y. Ming-Hsuan, "Diverse Image-to-Image Translation via Disentangled Representations," Proceedings of the European Conference on Computer Vision (ECCV), September 2018.

[7] P. Isola and J.Y Zhu and T. Zhou and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," arXiv: 1611.07004, 2016.
[8] M. Y. Liu and T. Breuel and J. Kautz, "Unsupervised Image-to-Image Translation Networks," arXiv: 1703.00848, 2018.
[9] J. Long and E. Shelhamer, Evan and T. Darrell, "Fully convolutional networks for semantic segmentation," Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp.3431–3440.
[10] M. Mirza and S. Osindero, "Conditional generative adversarial nets", arXiv preprint arXiv:1411.1784, 2014.
[11] A. Radford and L. Metz and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv preprint arXiv:1511.06434, 2015.
[12] R. Rosales and K. Achnan and B.J, Frey, "Unsupervised image translation", In ICVV (pp. 472-478), October 2003.
[13] T. Salimans and I. Goodfellow and W. Zaremba and V. Cheung and A. Radford and X. Chen, "Improved Techniques for Training GANs," arXiv: 1606.03498, 2016.
[14] T. C. Wang and M. Y. Liu and J. Y. Zhu and A. Tao and J. Kautz, Jan and B. Catanzaro, "High-Resolution Image Synthesis and Semantic Manipulation With Conditional GANs," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.
[15] J. Y. Zhu and T. Park and P. Isola and A. A. Efros, "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks," Proceedings of the IEEE International Conference on Computer Vision (ICCV), Oct 2017.
[16] J. Brownlee, "How to implement cycleGAN models", from "Generative Adversarial Networks with Python: Deep Learning Generative Models for Image Synthesis and Image Translation", 2019, pages 528–550
[17] Paul, S. & Chen, P. Vision Transformers are Robust Learners. *CoRR*. **abs/2105.07581** (2021), https://arxiv.org/abs/2105.07581
[18] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L. & Polosukhin, I. Attention Is All You Need. (2017)
[19] Paul, S. & Chen, P. Vision Transformers are Robust Learners. (2021)
[20] Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C. & Dosovitskiy, A. Do Vision Transformers See Like Convolutional Neural Networks?. (2021)
[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby. AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE. 2021.

APPENDIX

The specific setup of the generator and both discriminators are shown below.

- Generator model
  - Convolutional layer
    - 64 filters
    - 7x7 kernel, 1x1 strides
    - Padding: Half
    - Activation function: ReLU
  - Convolutional layer
    - 128 filters
    - 3x3 kernel, 2x2 strides
    - Padding: None
    - Activation function: ReLU
  - Convolutional layer
    - 256 filters
    - 3x3 kernel, 2x2 strides
    - Padding: None
    - Activation function: ReLU
  - x amount of Residual Blocks
  - Deconvolution layer
    - 128 filters
    - 3x3 kernel, 2x2 strides
    - Padding: Half
    - Activation function: ReLU
  - Deconvolution layer
    - 64 filters
    - 3x3 kernel, 1x1 strides
    - Padding: Half
    - Activation function: ReLU
  - Convolutional layer
    - 3 filters
    - 7x7 kernel, 1x1 strides
    - Padding: Half
    - Activation function: Tanh
- ResNet block
  - Convolutional layer
    - 256 filters
    - 3x3 kernel, 1x1 strides
    - Padding: Half
    - Activation function: ReLU
  - Convolutional layer
    - 256 filters
    - 3x3 kernel, 1x1 strides
    - Padding: Half
    - Activation function: ReLU
  - Channel-wise concatenation of ResNet input and output
- Discriminator model
  - Convolutional layer
    - 256 filters
    - 3x3 kernel, 1x1 strides
    - Padding: Half
    - Activation function: ReLU
  - Convolutional layer
    - 256 filters
    - 3x3 kernel, 1x1 strides
    - Padding: Half
    - Activation function: ReLU
  - Convolutional layer
    - 256 filters
    - 3x3 kernel, 1x1 strides
    - Padding: Half
    - Activation function: ReLU
  - Convolutional layer
    - 256 filters
    - 3x3 kernel, 1x1 strides
    - Padding: Half
    - Activation function: ReLU

Fig. 3. Bigger version of figure 1 for better visibility: original image (Left), CNNcycleGAN (middle), ViTcycleGAN (right)

Fig. 4. Bigger version of figure 2 for better visibility: original image (Left), CNNcycleGAN (middle), ViTcycleGAN (right)