Proceedings of 10th IOE Graduate Conference
Peer Reviewed
ISSN: 2350-8914 (Online), 2350-8906 (Print)
Year: 2021 Month: October Volume: 10

Landscape Image Season Transfer Using Generative Adversarial Networks

Bishnu Hari Paudel [a], Rupesh Kumar Sah [b]

[a, b] *Department of Electronics and Computer Engineering, Paschimanchal Campus, IOE, Tribhuvan University, Nepal*
Corresponding Email: [a] bishnuhari@wrc.edu.np, [b] rupesh@wrc.edu.np

Abstract

Image season transfer problem is an application domain of image-to-image translation and defined as transferring image from one season to another; for instance, transferring summer image to winter and it's vice versa. Image-to-image (I2I) translation involves generating a new synthetic form of a given input image with a specific alteration by keeping the source image attributes intact and their mapping from source to target domain. I2I is one of the popular applications of deep learning neural networks. One of successful variants of Generative Adversarial Networks (GANs), CycleGAN has been implemented with unpaired data samples. The CycleGAN is two domains, unsupervised approach to cyclic consistency which can be trained without pair image samples. Residual Network (ResNet) is used for generative model and PatchGAN is for discriminative model in the first CycleGAN, and hence ResNet generator became general practice. The Residual Network architecture is replaced here with U-Net architecture. U-Net is considered as a fast neural network and works fine even with small size of dataset.This study uses two sets of landscape images to train the GAN model and hence transfer the season from summer to winter or in opposite direction.

Keywords

Image-to-Image Translation, CycleGAN, U-Net, PatchGAN, Cycle-Consistency, Adversarial, Composite Model

1. Introduction

Generative Adversarial Networks (GANs) are big enables for Image-to-image (I2I) translation problems. GAN is an unsupervised deep learning framework with two models, a generator model (say G) and a discriminator model (say D), are trained simultaneously [1]. Both models compete against each other to reach the state where the discriminator is no more able to distinguish the real image from data distribution or fake image from generator pool. This is the most desired state and commonly known as Nash equilibrium point. Hence, the generator model now tries to generate plausible image to fool the discriminator model, while the discriminator model tries to distinguish real and fake samples fed to it as input image.

Image season transfer, typically image from summer to its winter image; is an application domain of image to image conversion. It is highly challenging task to construct a deep neural networks that can translate an image from one season to another. In recent years, generative adversarial networks (GAN) and their variants have been used to provide learning-edge solutions to image-to-image conversion problems [2]. The generative and discriminative models both can have any algorithm as long as the generative models have the capability to learn from training data distributions and discriminative models have the capability to extract the feature for classifying the generator output. [1] [2]. GAN has several successful variants from supervised to unsupervised learning, from single modal to multi-modal output, and from single domain to multi-domain translation [3].

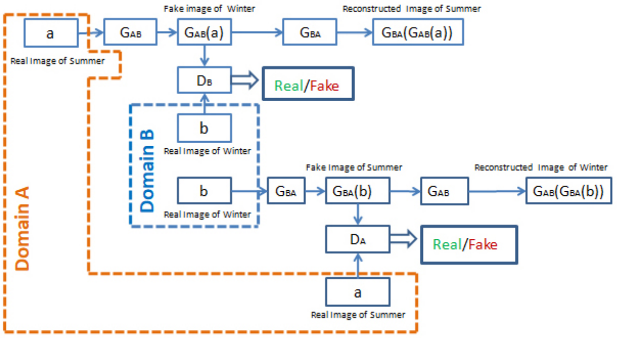

Figure 1: CycleGAN Operations

The CycleGAN is an approach to cyclic consistency, single-modal, unsupervised and two domain image to image translation which can be trained without paired samples [4]. The CycleGAN uses two generative models (say G_{AB} and G $_{BA}$), and two discriminative models (say D_A and D_B). Particularly, $G_{AB}(a)$ is generated image (a fake winter image) of a summer image 'a' from domain A, generated by G_{AB} generator model. The $G_{AB}(a)$ output goes to discriminator D_B for real or fake detection. At the same time, $G_{AB}(a)$ goes to generator model G_{BA} and generates back to $G_{BA}G_{AB}(a)$ as a reconstructed summer image which is going to compare with original summer image 'a' and hence completes a cycle as shown in figure above. Similarly, generator G_{BA} takes 'b' as in input and generates $G_{BA}(b)$, a fake image of winter sample 'b' from domain B. The generator output is sent to discriminator D_A and compared with real summer sample, 'a' sent from domain A. At the same time, the generator output also goes to first generator, G_{AB} and now produces reconstructed winter image $G_{BA}G_{AB}(a)$ for cycle-consistency.

If we combine both operations, we find CycleGAN framework as shown in figure below. The first cycle is called as forward cycle (summer to winter) and depicted here with red cycle. The blue cycle in figure represents backward cycle. The red cycle is named as forward cycle and blue cycle as backward cycle (winter to summer).

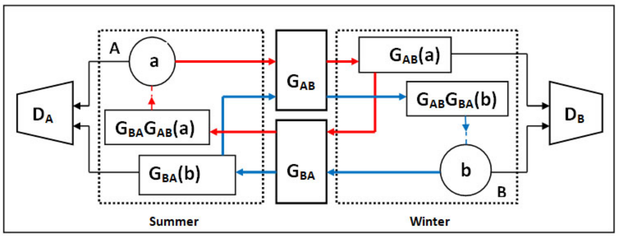

Figure 2: CycleGAN Framework

2. Problem Statement

Seasons are division of year that marked by changed in weather, ecology, and amount of daylight. Change in seasonality in the nature has a great impact on the environmental and visual features of landscape [5]. For various studies and planning, the landscape image is compared with its counterpart season (summer to winter and vice-versa). Taking the real picture of landscape image in different season, there will be a long wait of almost six months. And it is practically hard, and expensive to collect such paired image dataset having outdoor natural landscape. In such situation, unpaired dataset from different domains are used for image-to-image conversion problems. The generative adversarial networks are deep learning approaches; they usually need large dataset for model training.

In case of unpaired dataset from different domain, CycleGAN is a typical model of generative adversarial networks approach of image to image conversion. The CycleGAN uses two loss functions namely adversarial loss (mean square) and cycle-consistency (absolute) loss. The new CycleGAN is built with U-Net generator and PatchGAN discriminator and hence the loss functions need to be assessed during model training. The training image data from two different domains (i.e. summer and winter) simultaneously passed to the model through two separate standard GANs but in opposite directions.

3. Related Works

Since 2014, Generative Adversarial Network (GAN) is in existence in the field of deep neural networks.As a deep learning framework, the generative model and discriminative models both can be trained simultaneously[1]. On the other hand, Alotaibi A.[2] elaborated different approaches of deep generative adversarial network's algorithms for image-to-image conversion and classified the algorithms in two major classes: discriminative feature learning and generative feature learning. And different variants of GANs are categories with respect to their applications.Especially for unpaired datasets, CycleGAN, DiscoGAN, DualGAN and AsyGAN are common GAN variants for various research studies of unsupervised translation with cyclic-consistency approach. CycleGAN is able to learn well with one-to-one mapping, however other variants such as DiscoGAN and DualGAN can also work with many-to-may learning mode.

Zhu et al.[4] proposed CycleGAN as a new generative adversarial network for unpaired image-to-image translation using cycle-consistency scheme. It uses two generative networks and two discriminative networks. The adversarial network contains three convolutional layers, several residual blocks with skip connections, two fractionally-strided convolutional layers with stride value of 0.5, and one final convolutional layer that maps features to RGB color image. This architecture uses instance normalization. For the discriminator networks, they have used 70x70 PatchGAN. The two objective functions of CycleGAN are adversarial loss and cycle-consistency loss.

Ronneberger et al. [6] developed U-Net, an image segmentation architecture for deep learning neural networks. Authors claim that U-Net is fast network. The architecture consists of a downsampling path to capture content and an upsampling path that enables precise localization; connected by a latent space. They claim that the network can be trained end-to-end even with small sized dataset and outperforms the prior best methods.

Isola et al.[7] introduced the concept of PatchGAN in 2018, the discriminative model. It works convolutionally as just penalizes structure at the predefined scale of patches N dimension for the given image. The overall output from PatchGAN is finally calculated by averaging the all response from the patches and gives a probability; if it above the threshold the image is real otherwise fake image. Zhang et al.[8] clearly describes the concept of receptive field of PatchGAN. As name imply, PatchGAN uses a small patch for its computation. The size of the patch for each layer is described by its receptive filed.

4. Methodology

GAN has two sub-models: the generative model that we train to generate new samples, and the discriminative model that tries to classify the input sample provided, as either real from the data distribution or fake from the generated image pool. These two sub-models are trained together in zero-sum game perspective, in adversarial way. The discriminative model maps the image features to class labels. When both models satisfy with Nash equilibrium, the generated image is the desired output of the whole system.

CycleGAN is one of the first models to allow for unpaired image-to-image training. There are two collections of images and they are unpaired, i.e. they don’t have the exactly same scenes in winter and summer. CycleGAN utilizes two GANs, and each GAN has a discriminator and a generator model, i.e. there are four models all together. The first GAN generates images of winter for given photos of summer, and the second GAN generates images of summer for the given image of winter. There are two types of training losses with CycleGAN: Adversarial loss and Cycle-consistency loss as shown in figure below.

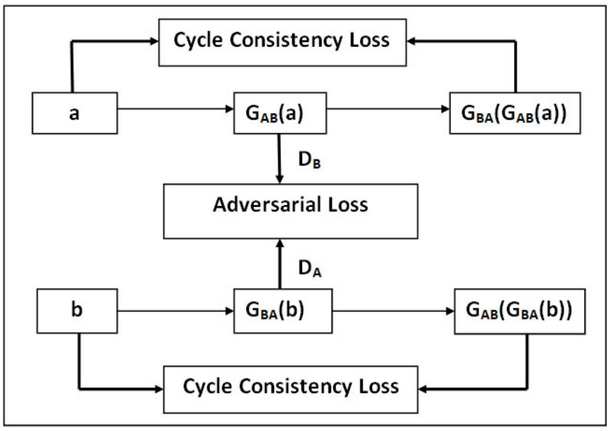

Figure 3: CycleGAN Loss Functions.

Adversarial Loss: The adversarial loss here corresponds to the standard generative adversarial loss function. In CycleGAN, it is considered as discriminator loss.
For mapping function $\mathrm{G_{AB}}$: A –>B and discriminator $\mathrm{D_B}$, where $\mathrm{G_{AB}}$(a) is generated winter image.

$$L_1 = E_{b1} + E_{a1} \tag{1}$$

where,
$L_1 : L_{GAN1}(G_{AB}, D_B, A, B)$
$E_{b1} : E_{b-Pdata(b)} Log[D_B(b)]$
$E_{a1} : E_{a-Pdata(a)}[Log(1 - D_B(G_{AB}(a)))]$

Similarly, for mapping function $\mathrm{G_{BA}}$: B –>A and discriminator $\mathrm{D_A}$, where $\mathrm{G_{BA}}$(b) is generated summer image.

$$L_2 = E_{a2} + E_{b2} \tag{2}$$

where,
$L_2 : L_{GAN2}(G_{BA}, D_A, B, A)$
$E_{a2} : E_{a-Pdata(a)} Log[D_A(a)]$
$E_{b2} : E_{b-Pdata(b)}[Log(1 - D_A(G_{BA}(b)))]$

Above two equatoins carry following information: $\mathrm{E_{a\ -\ Pdata(a)}}$ and $\mathrm{E_{b\ -\ Pdata(b)}}$ denote the data distribution in domain A (summer) and in domain B (winter) respectively. Here ’a’ represents a sample image form summer data collection and ’b’ represents a sample from winter data collection. The term Log [$\mathrm{D_B}$(b)] in the above equation (1) evaluates that the particular image ’b’ is real or not.

Hence the first part of the equation deals with real sample. Additionally, the term Log(1-D_B(G_{AB}(a))) in the equation (1) evaluates that the particular generated image G_{AB}(a) is fake or not. So, this part deals with generated sample. Similarly, the first part of equation (2) evaluates that the particular image 'a' is real or not and the second part, Log(1-D_A(G_{BA}(b))) evaluates the generated image namely, G_{BA}(b) is fake or not. Adversarial loss is calculated as mean square error form.

Cycle Consistency Loss: The cycle consistency loss function corresponds to reconstruction of the image. In CycleGAN network, it is termed as generator loss function. For each image 'a' from domain A, the image season transfer cycle should be able to generate it back to the original type image, and hence the original sample 'a' is compared with double generated sample G_{BA}(G_{AB}(a)) as shown in equation (3) below. It is known as forward cycle, expressed as L_{c1} and calculated as absolute error.

$$L_{c1}(G_{AB},G_{BA}) = E_{a-Pdata(a)}||G_{BA}(G_{AB}(a))-a|| \quad (3)$$

Similarly, for each image 'b' from domain B, G_{AB} and G_{BA} should be able to bring it back to the original image, and hence the sample 'b' is compared with double generated sample G_{AB}(G_{BA}(b)) as shown in equation (4) below. It is known as backward cycle, denoted by L_{c2} and calculated as absolute value.

$$L_{c2}(G_{BA},G_{AB}) = E_{b-Pdata(b)}||G_{AB}(G_{BA}(b))-b|| \quad (4)$$

4.1 The U-Net Generator

U-Net architecture consists of two parts; downsampling and upsampling joined by a bottleneck section which holds latent space. The downsampling part is contracting and the upsampling part is expansive in nature[6]. The standard architecture is modified here with eight convolutional layers encoder and eight convolutional layers decoder as shown in figure below. Intuitively with each downsampling encoding step, convolutional layer doubles up the number of features channels by reducing the size of the image, and ultimately reaches to latent space of 1x1x512.

The upsampling part of U-Net uses a latent vector and gradually processes up and generates an equal sized image as input i.e. 256x256x3. On the way to generated image, the latent vector travels through upsampling in each layer that halves the number of feature channels. After each upsampling, a convolution is performed and the result is normalized with instance normalization and concatenated with the analogously feature map from the downsampling path. For the first three upsampling decoding steps, dropout function is applied to protect the architecture from vanishing gradient problem. Now the output is passed through activation function ReLU before entering to next step of upsampling. For the final convolutional layer, tanh is used as activation function with no concatenation and normalization function.

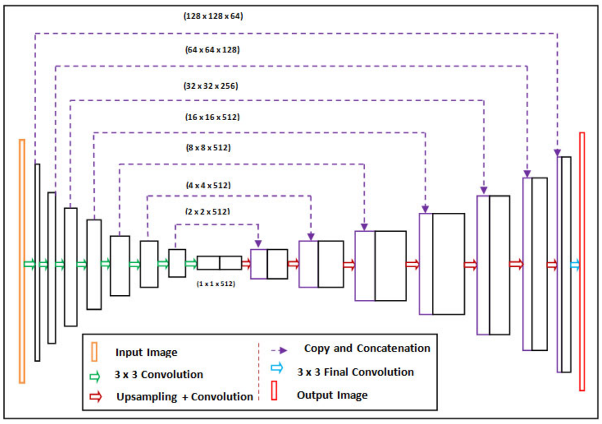

Figure 4: The Generative Unit.

The beauty of U-Net architecture is its skip connections between equivalently shaped layers in the upsampling and downsampling part of the network[6]. These shortcut paths of the network are quick enabler to regain the abstract information from the original image.

The intuition here is that with each subsequence layer in the downsampling part of the network, the model increasingly captures the 'what' of the image (i.e. content of the image), and losses information about 'where' from the image. At the apex of the U, the feature map will have learned a contextual part of the image with very little understanding of where it is located[9]. As U-Net is originally designed for image segmentation; it does not flatten here but goes up with required feature vector and finally gives an image of input size.

4.2 The PatchGAN Discriminator

PatchGAN, by nature penalizes only structure at the scale of local image sections commonly known as patches. PatchGAN tries to distinguish if each NxN patch in an image is real or fake. The input image for the PatchGAN layers is divided in to patches of square of length N. The discriminator patch runs convolutionally across the image and finally the ultimate prediction from discriminator is calculated by averaging the all responses from the patches[7]. The patch responses are completely independent to each other.The patch size used in each layer of PatchGAN discriminator is called receptive field for that particular layer. A PatchGAN is like a convolutional layers network, however the receptive fields of the discriminator layers turn out of 70x70 patches in the input image[8].

The advantage of implementing a PatchGAN discriminator is that the loss function measured is found better on their ‘style’ rather that their ‘content’[9].The discriminator model network creates a single channel of 30 x 30 feature map to represent the loss.The discriminator model network creates a single channel of 30 x 30 feature map to represent the loss.The PatchGAN handles the arbitrary sizes of input image; known as patch size,as far as the labels have been predicted so that they are the equal size as the loss map. It measures the quality of the input image as per to the quality of local patches. That means it concerns with local property rather than the global property of the image.

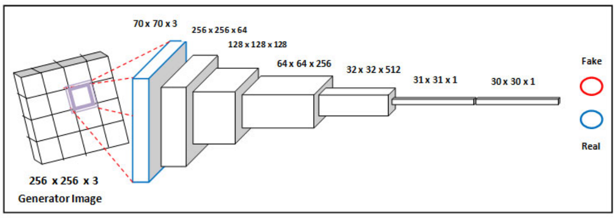

Figure 5: The Discriminative Unit.

4.3 The Composite Model

The discriminative models are trained straight away from real images from data distribution of training dataset and the fake images from generated image pool of the generator. However, the generator models are not trained then after via their discriminator models output. The generators are trained to update for minimizing the training function loss predicted by the discriminator for the generated images; defined by mean square error and known as adversarial loss, as such motivated generator to generate images that better fit into the target domain [10].

Side by side, the generator models are also updated based on how effectively they are at the reconstruction of an image from source domain; defined by mean absolute error and known as cycle-consistency loss, that is further divided into two forms namely forward cycle and backward cycle . Altogether, a generator is updated with the combination of three loss functions: one mean square error (L2) and two mean absolute errors (L1).

The forward cycle consistency loss involves connecting the output of the generator to the other generator designed for reconstruction of the source image. And, the backward cycle consistency loss involves the image from the target domain and the reconstructed image from the other generator.

A composite model for each generator model (G_{AB} and G_{BA}) is needed. And a composite model needs two inputs for the real image fro source domain (summer) and target domain (winter) and three output: the discriminator output, forward cycle generated image (summer-winter-summer) and backward cycle generated image (winter-summer-winter). Only the weights of first or main generator model are updated for the composite model. As comparing with adversarial loss, cycle loss has huge influence during training and hence nearly 10 times of adversarial loss is assigned for weight balancing purpose.

4.4 Dataset

Yosemite Valley images collection is available as Yosemite dataset. The dataset contents random images taken from Yosemite national park area. The dataset is properly categorized as summer and winter training and test data. The dataset contains more than 1500 summer images and more than 1200 winter images. After removing less significant images for landscape property and duplicate images; there are about thousand images of each season. The image dimension is 256x256x3.

Ronneberger et al.[6] claim that U-Net not only a fast network, but works very fine with small sized dataset and outperforms the prior best methods.And data augmentation and implementation is considered as a part of future extension of the study.

5. Results and Discussion

After successful training of 50,000 iterations, the model is able to perform for producing plausible image of season transfer. The generator performances and corresponding loss function recordings at different stages of iterations are shown below. The discriminator loss is categorized as discriminator loss with real samples and fake samples. The generated loss plotted here is two generators of two composite model of CycleGAN. As we go through loss functions, the discriminator losses and generator losses are gradually decreasing, and reach to acceptable range common practice. Generally, with GANs the discriminator loss is assumed to supress to 0.5 and for generator, around 2.0.

5.1 Training Performance

The model training iterations at the beginning measured the discriminator loss of 5.0 for both real and fake samples;similarly, the generator loss recorded up to 20 for both samples. At 5000 iterations, the discriminator loss is settled to 0.3 and the generator loss came around 3.0. Since that the measures decrease gradually and reach to average of 0.15 and 1.7 respectively at 50,000 iterations.

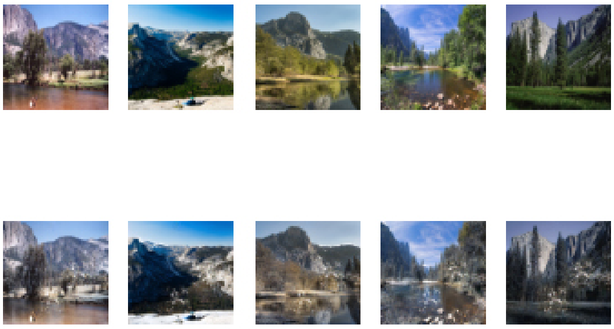

Figure 6: Summer to Winter at 50,000 Iterations

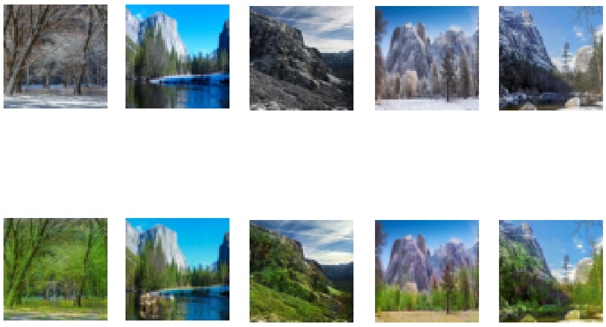

Figure 7: Winter to Summer at 50,000 Iterations

5.2 Output and Evaluation

After 50,000 successful training iterations of model learning, the discriminator part is separated and only the generator model is used to test new sample image. The input and output shape of image is 256x256x3. In a fig 9 below show summer to winter transfer; the first image in the row is original input image (summer), the image in the middle is generated image (winter), and the last image in the row is reconstructed image (summer) from the generated image

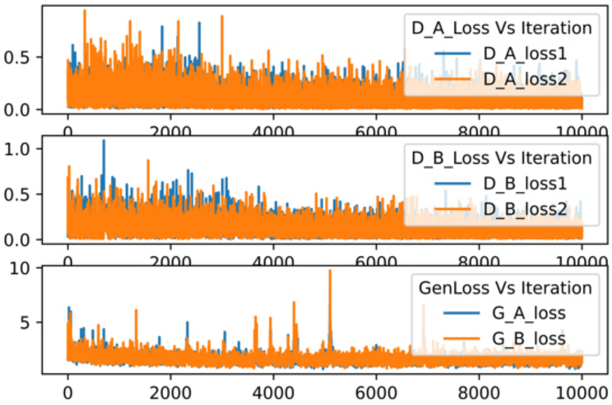

Figure 8: Model Training Loss Plot at 50,000 Iterations

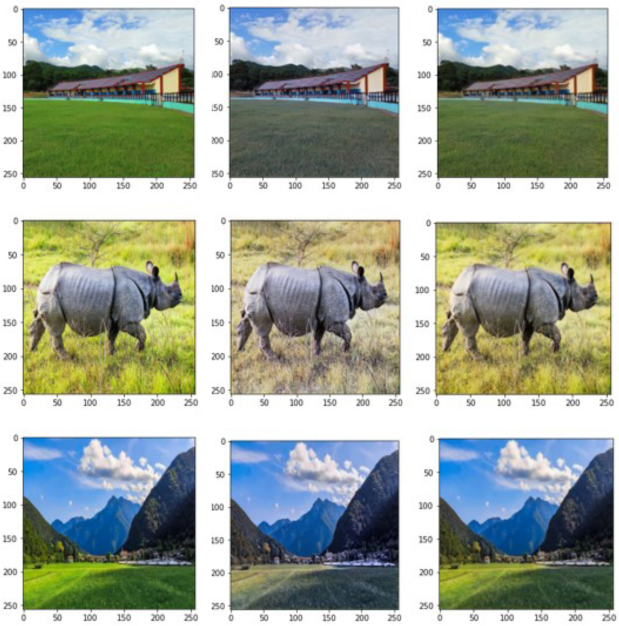

Figure 9: Summer to Winter Transfer and Reconstructed Image.

Fig 10 below holds sample output of winter to summer transfer. The first image in the row is original image(winter), the image in the middle is generated image (summer), and the last image in the row is reconstructed image (winter) from the generated image.

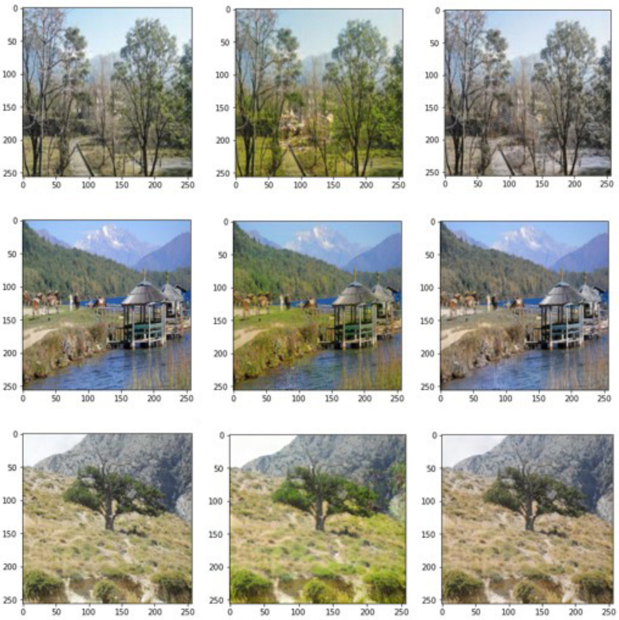

Figure 10: Winter to Summer Transfer and Reconstructed Image

The original input image is compared with its reconstructed image for similarity index. So, this metric examines the performance of CycleGAN generator models. One of the popular image assessment tool these days is the Structural Similarity Index Metric (SSIM), which is basically used to compare three features of original image and the reconstructed image that is gone through two generator models during a cycle. Those three features are: Luminance, Contrast, and Structure.

Basically, season transfer and style transfer problems are sensitive with contrast value of the images. SSIM considers luminance, contrast, and structural value at the same time. The tables below show that about 87 percent of original input image and the reconstructed image have similarity.

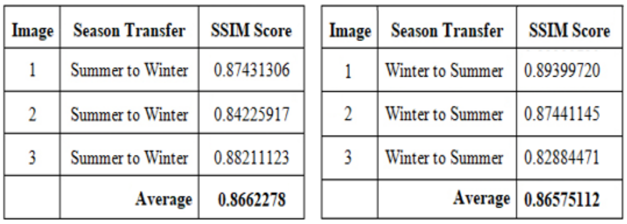

Image	Season Transfer	SSIM Score
1	Summer to Winter	0.87431306
2	Summer to Winter	0.84225917
3	Summer to Winter	0.88211123
	Average	**0.8662278**

Image	Season Transfer	SSIM Score
1	Winter to Summer	0.89399720
2	Winter to Summer	0.87441145
3	Winter to Summer	0.82884471
	Average	**0.86575112**

Figure 11: Structural Similarity Index Matric (SSIM)

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[2] Aziz Alotaibi. Deep generative adversarial networks for image-to-image translation: A review. *Symmetry*, 12(10):1705, 2020.

[3] Yingxue Pang, Jianxin Lin, Tao Qin, and Zhibo Chen. Image-to-image translation: Methods and applications. *arXiv preprint arXiv:2101.08629*, 2021.

[4] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[5] Rein Ahas, Anto Aasa, Siiri Silm, and Jüri Roosaare. Seasonal indicators and seasons of estonian landscapes. *Landscape Research*, 30(2):173–191, 2005.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[8] Jiexin Zhang, Jianjiang Zhou, and Xiwen Lu. Feature-guided sar-to-optical image translation. *IEEE Access*, 8:70925–70937, 2020.

[9] David Foster. *Generative deep learning: teaching machines to paint, write, compose, and play*. O'Reilly Media, 2019.

[10] Jason Brownlee. *Generative adversarial networks with python: deep learning generative models for image synthesis and image translation*. Machine Learning Mastery, 2019.