https://doi.org/10.31436/iiumej.v27i1.3824

Enhanced Beach Photo Translation using Modified Unsupervised GAN with Regularization

KARTIKA FITHRIASARI*, BENEDICTUS KENNY TJAHJONO

Department of Statistics, Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia

**Corresponding author: kartika_f@its.ac.id*

(Received: 1 July 2025; Accepted: 26 September 2025; Published online: 12 January 2026)

ABSTRACT: To optimize time and cost, tourists often require tools to modify the sky background and atmosphere in beach photos, such as replacing blue-sky views with sunsets or vice versa. Independent modification of the sky and sea is difficult because their color palettes are similar. Another problem that often occurs in image translation is the scarcity of paired datasets, and beach photo datasets are particularly limited in lighting conditions, weather variations, and viewing perspectives. This limitation can cause Generative Adversarial Networks (GAN) models to lose their generalization ability, become prone to overfitting, and produce visual artifacts in the outputs. Therefore, this study proposes an unsupervised GAN approach using a modified CycleGAN and improves its performance for beach image translation by integrating identity mapping, λ-parameter optimization, a multiscale kernel, and regularization techniques. CycleGAN consists of two generators and two discriminators. The sunset generator translates a blue sky into a sunset sky; the generated output is then passed to the sunset discriminator to determine whether it is real or fake. The generator input image is resized and normalized through preprocessing. The generator architecture is structured to enhance image reconstruction and feature extraction. The details of the translation results are fine-tuned using a 30x30 PatchGAN discriminator and a multiscale kernel convolutional layer. The effect of the hyperparameter λ, which strikes a balance between cycle consistency, structural preservation, and color fidelity, is also investigated in this work. The findings indicate that while higher λ values increase generator loss, they also improve consistency, making it harder to handle dark objects and white clothing. To overcome this issue, regularization techniques, namely photometric augmentation and spectral normalization (SN), together with multiscale kernel convolutional (MSCov), have been applied. Photometric augmentation and MSCov are used to enhance the model's robustness to photographic variations, while SN improves its efficiency and stability. The results of the study show that the proposed method improves image translation accuracy as measured by Mean Squared Error (MSE), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS).

ABSTRAK: Bagi mengoptimum masa dan kos, pelancong sering memerlukan alat menukar foto latar belakang langit dan suasana pantai, seperti mengganti pemandangan langit biru dengan matahari terbenam atau sebaliknya. Pengubahsuaian bebas langit dan laut sukar dilakukan kerana palet warna langit dan laut adalah serupa. Masalah lain sering berlaku dalam menterjemah imej adalah ketiadaan set data foto berpasangan dan foto pantai sering mengalami kepelbagaian terhad, terutama dari segi pencahayaan, variasi cuaca dan perspektif tontonan. Had ini boleh menyebabkan model Rangkain Generatif Adversari (GAN) kehilangan keupayaan generalisasi, terdedah kepada terlebih muat dan penghasilan artifak visual dalam hasil terjemahan. Oleh itu, kajian ini mencadangkan pendekatan GAN tanpa pengawasan menggunakan CycleGAN yang diubah suai bagi meningkatkan prestasi terjemahan imej pantai melalui penyepaduan pemetaan identiti, pengoptimuman parameter , kernel berbilang skala dan menggunakan teknik regularisasi. CycleGAN terdiri daripada dua generator dan dua rangkaian neural diskriminator. Generator matahari terbenam digunakan

dalam menterjemah langit biru kepada langit matahari terbenam, kemudian imej terhasil dimajukan kepada diskriminator matahari terbenam bagi menentukan sama ada imej terhasil dikelaskan sebagai imej sebenar atau imej palsu. Imej input generator diubah saiz dan dinormalkan melalui prapemprosesan. Seni bina generator distrukturkan dengan meningkatkan pembinaan semula imej dan pengekstrakan ciri. Butiran hasil terjemahan diperhalusi menggunakan diskriminator PatchGAN 30x30 dan lapisan konvolusi kernel berbilang skala. Kesan hiper parameter turut dikaji bagi mencapai keseimbangan antara ketekalan kitaran, pemeliharaan struktur, dan kesetiaan warna. Dapatan kajian menunjukkan bahawa walaupun nilai lebih tinggi ianya meningkatkan kehilangan generator dan konsistensi, menjadikannya lebih sukar dalam mengendali objek gelap dan pakaian putih. Bagi mengatasi isu ini, teknik regularization iaitu pembesaran fotometrik dan normalisasi spektral (SN) bersama Konvulasi Kernel Skala Berbilang (MSCov) telah digunakan. Pembesaran fotometrik dan MSCov dilaksanakan dalam meningkatkan keteguhan model pada variasi fotografi, manakala SN digunakan bagi meningkatkan kecekapan dan kestabilan model. Hasil kajian menunjukkan kaedah ini mampu meningkatkan ketepatan hasil terjemahan imej berdasarkan Ralat Purata Kuasa Dua (MSE), Indeks Persamaan Struktur (SSIM) dan Persamaan Tampalan Perseptual Imej Terpelajar (LPIPS).

KEYWORDS: *Beach Photo, CycleGAN, Multiscale Kernel Convolutional, Unsupervised GAN, Image Translation*

1. INTRODUCTION

Among the most visited tourist spots is the beach. The sunset view is one of the most common themes in tourist beach photography. Due to conditions such as cloudy weather or limited afternoon time, tourists may be able to take photos only in the morning or evening, rather than at sunset. To save time and money, a method is needed to change the atmosphere of beach photos as desired; for example, changing a beach photo with a blue sky to a sunset view and vice versa. Achieving a realistic transformation is challenging. The main difficulty is that the sky, which must be transformed, has a color similar to that of the sea, an element that must be preserved. An efficient method is needed to replace the sky background while maintaining the other elements, ensuring that the result realistically reflects the desired atmosphere and scenery.

The image-to-image translation process entails analyzing the image's content and modifying it to conform to the target domain's features. If X refers to the source domain and Y refers to the target domain, then the translation process can be denoted as $X \rightarrow Y$. For example, in beach photography, the X may represent photographs of the beach during the day, while the Y could represent the images during sunset, or vice versa. Image-to-image translation, using computer vision algorithms, offers a practical solution to challenges in beach photography, enabling tourists to capture better images and preserve their holiday memories.

Generative Adversarial Networks (GANs) are neural networks designed to solve image-to-image translation tasks, particularly in applications that require paired images [1]. In reality, obtaining paired images is often impossible. This led to the development of unsupervised GANs. One notable unsupervised GAN developed is CycleGAN [1]. CycleGAN incorporates a cycle-consistency loss that aims to preserve the image's structure when it is converted back to its original domain. This study further improves CycleGAN's performance by modifying its network architecture. The generator uses a transposed convolutional layer, while the discriminator uses a PatchGAN with a 30×30 resolution. This is intended to improve the recognition of fine details in CycleGAN. The hyperparameter λ controls the balance between image realism and structural preservation, thereby affecting the quality of the resulting image

https://doi.org/10.31436/iiumej.v27i1.3824

[2,3]. Optimization of λ is necessary to improve the performance of CycleGAN, so in this research, the impact of hyperparameters was evaluated.

The main challenge in translating daytime beach images into sunset scenes is the substantial color similarity between the sky and the sea, which can cause unwanted changes across the entire image. To address this, we incorporate identity mapping into CycleGAN, ensuring that the generator learns to preserve regions that require little modification. This prevents distortions of the natural landscape's structure and helps maintain the original color balance.

Furthermore, we investigate the role of the hyperparameter λ, which regulates both cycle consistency loss and identity mapping. By fine-tuning λ, we control the trade-off between realism and structural preservation, allowing for a more precise and visually coherent transformation. Our approach ensures that changes are selectively applied to areas requiring modification while preserving the integrity of unaltered regions.

Another challenge in beach photo datasets is that images are often limited to specific lighting conditions or weather states, which limits the model's ability to capture the underlying visual complexity of beach environments. As a result, models trained on such data may struggle to generalize, leading to less robust translations and reduced realism in generated images. Addressing these issues requires approaches to increase image variation, particularly in lighting and color, to improve model stability. Therefore, this study applies photometric augmentation to the data to enhance the accuracy of CycleGAN [4]. In unpaired image translation tasks, the stability of the discriminator is crucial to ensure that the generator continues to learn effectively from the discriminator's feedback. To maintain signal stability and improve efficiency, spectral normalization (SN) is applied to the discriminator.

The contributions of this research include improving CycleGAN's performance for beach image transformation by integrating hyperparameter optimization and regularization techniques, namely photometric augmentation and SN. These methods are applied and evaluated using Mean Squared Error (MSE), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) to measure the quality of the translation results. This study not only offers a practical solution for tourists aiming to enhance their beach photography but also contributes to advancements in image processing and computer vision.

2. METHODS

2.1. Digital Image

An image can be represented as a matrix of RGB intensity values with U rows and V columns, where (u, v) denotes spatial coordinates, and $x(u, v)$ denotes the color intensity at a given point. RGB in an image consists of three primary color channels: red, green, and blue, each represented by eight bits, resulting in a total of 24 bits per pixel.

Understanding the structure of a digital image is essential for analyzing its characteristics, such as color distribution and contrast. Such distribution patterns can provide insight into the variation of pixel intensities in an image. The frequency of pixel intensities in a given color channel at a specified intensity scale can be determined from the distribution plot.

2.2. Photometric Augmentation

For outdoor images such as beach landscapes, the translation model must be capable of handling a wide range of photo conditions, especially in unpaired data translation tasks. In such cases, the model is prone to overfitting. Variations in lighting conditions can lead to a mismatch

https://doi.org/10.31436/iiumej.v27i1.3824

between the source and target domain distributions. To address this issue, photometric augmentation is applied to increase the diversity of training data images [5].

These photometric transformations are applied before the image is fed into the generator network. The transformations used in the augmentation process include brightness adjustment, contrast variation, saturation shift, and cutout (to simulate information loss), applied randomly with parameters constrained to specified ranges. If I denotes the intensity of the image to be augmented, the brightness transformation is performed according to Eq. (1) with the parameter $\alpha \in [-a, a]$. This transformation produces image variations that are either darker or brighter.

$$I^* = I + \alpha. \tag{1}$$

For the contrast variation augmentation technique, intensity rescaling is performed using Eq. (2) with the parameter $\beta \in [0, \infty)$ and μ representing the mean image intensity.

$$I^* = (I - \mu) \cdot \beta + \mu. \tag{2}$$

Image saturation is transformed by converting the RGB image into the HSV (Hue, Saturation, Value) color space and modifying the Saturation (S) component as shown in Eq. (3) with the parameter $\gamma \in [0, \infty)$. After modification, the image is converted back to the RGB format.

$$S^* = \gamma \cdot S. \tag{3}$$

Translation augmentation is used to enable the model to capture positional variation. Let $(\Delta p, \Delta q)$ represent the horizontal and vertical shift values, respectively. Then, a random shift can be performed using Eq. (4).

$$I^*(p, q) = I(p - \Delta p, q - \Delta q). \tag{4}$$

Cutout augmentation is a data augmentation technique applied to images by randomly masking out a square region of the input during training. The cutout augmentation process begins with an input image $x \in \mathbb{R}^{H \times W \times C}$ of height H, width W, and color channels C. The size of the cutout region is determined by computing the side length of the square mask as $s = \lfloor \rho \cdot \min(H, W) \rfloor$, where $\rho \in (0,1]$ is a ratio parameter controlling the proportion of the image to be masked. Next, a random position is selected by sampling an offset u uniformly from $[0, H - s]$ to define the row position and v uniformly from $[0, W - s]$ to define the column position. A binary mask $M \in \{0,1\}^{H \times W \times C}$ is created, where the values inside the square region $[u, u + s) \times [v, v + s)$ are set to zero, and the values outside this region are set to one. Finally, the augmented image was obtained by multiplying the original image by the mask element-wise, given in Eq. (5). This sets all pixels within the cutout region to zero (black), leaving the remainder unchanged.

$$x^* = x \cdot M. \tag{5}$$

If x is an image from the source domain X, then the pipeline after applying the augmentation $T(\cdot)$ is given by Eq. (6),

$$x^* = T(x), y = G_Y(x^*), \tag{6}$$

where y is the translation result, and G_Y is the generator that translates the image from the actual image in domain X to the generated image in domain Y.

2.3. Neural Network

One of the machine learning methods inspired by the structure and function of the human brain is a neural network (NN) [6]. To improve accuracy, the optimization process is applied

https://doi.org/10.31436/iiumej.v27i1.3824

during parameter estimation. The objective function of optimization is to minimize a loss function that quantifies the difference between predicted and actual values [7]. Stochastic Gradient Descent (SGD) is a widely used optimization method that updates parameters by following the negative gradient of the loss function. When the SGD learning rate is fixed, this method exhibits instability [8]. To overcome this problem, RMSprop and Adam have been developed. RMSprop is more stable because the method prevents oscillation by adjusting the learning rate based on a moving average of the squared gradient. Adam, an extension of RMSprop, offers faster, more efficient convergence by combining momentum-based optimization with adaptive learning rate updates based on the first and second moments of the gradient [9].

Convolutional Neural Networks (CNNs) are a type of neural network designed to analyze images or patterns. CNNs can extract spatial features from images automatically [10]. Convolutional, pooling, and fully connected layers are the core components of CNNs. In the convolutional layer, features are extracted using kernels to form feature maps that representmage patterns [11]. This process is illustrated in Figure 1.

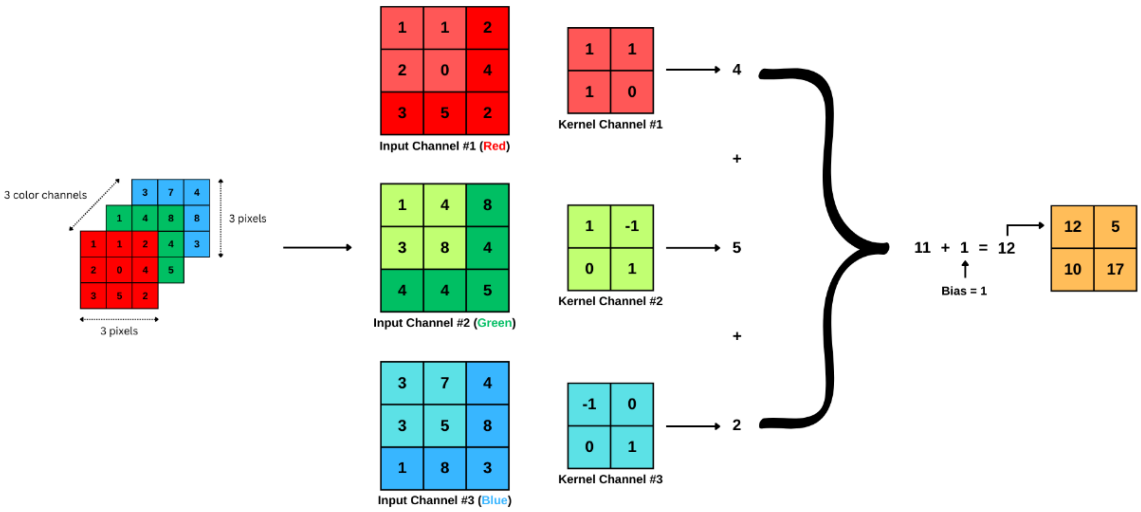

Figure 1. The convolutional operation

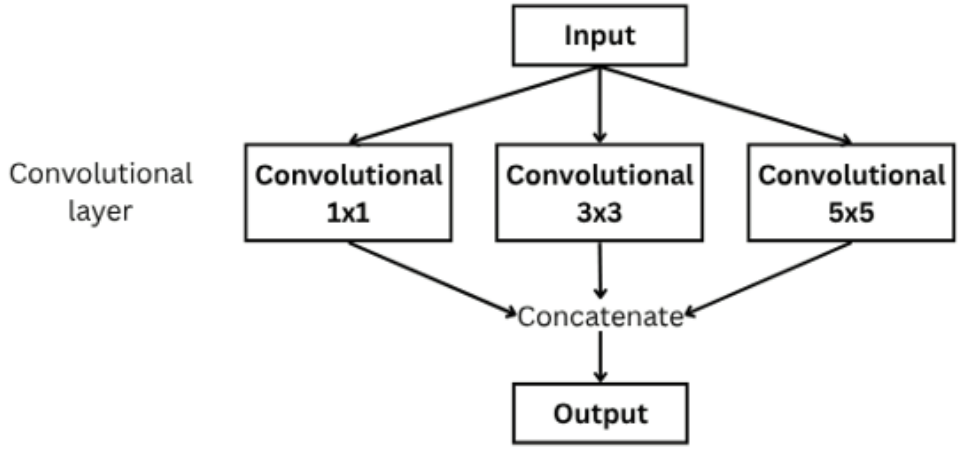

Figure 2. Multiscale kernel convolutional (MSCov) layer

A single kernel size is often insufficient to capture all variations in a feature because objects in an image frequently appear at different sizes and positions. To address this issue, a

multiscale kernel convolutional (MSCov) layer approach is used, combining multiple kernel sizes (e.g., 1×1, 3×3, 5×5) in parallel or sequentially within the convolutional layer [12], as illustrated in Figure 2. A pooling operation is applied to reduce the feature map dimensions and mitigate overfitting. The pooling method can use average or max pooling and is applied in the next layer [13].

The final process in a CNN is the fully connected layer. The layer consists of a flatten layer and a dense layer. Multidimensional feature maps are converted into one-dimensional vectors in the flatten layer. Furthermore, the dense layer learns the complex relationships between features to make accurate predictions [14].

2.4. Generative Adversarial Networks (GAN)

Generative Adversarial Networks (GAN) are a method for pairwise image-to-image translation [15]. Pairwise training data must be available when using GAN. The GAN framework consists of two neural networks: a generator and a discriminator. The generator aims to produce realistic synthetic images that resemble the original, while the discriminator's role is to distinguish the original from the generated image. The generator continues to improve at producing realistic-looking images, while the discriminator simultaneously improves at detecting fake images. The results from both networks will improve the GAN's ability to produce high-quality, realistic image translations. Figure 3 illustrates the GAN framework.

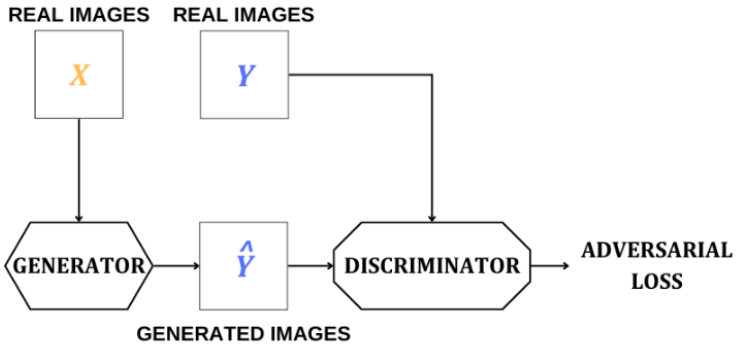

Figure 3. GAN framework

In this framework, X represents real images from the source domain, while Y denotes real images from the target domain. The generator produces synthetic images ($\hat{Y}$), which the discriminator evaluates to compute adversarial loss. Initially, a set of real images X is fed into the generator network, which translates them into synthetic images, denoted as $\hat{Y}$. These generated images are then combined with the real images Y and passed to the discriminator, which assesses their authenticity. Through this process, the adversarial loss function quantifies the discrepancy between real and generated images, providing a metric for evaluating translation accuracy.

The adversarial loss function is derived from the Bernoulli distribution probability density function. More formally, let Z be a random variable where $z = \{0,1\}$, with $z = 0$ representing a fake image and $z = 1$ representing a real image. Consequently, Z follows a Bernoulli distribution with a probability density function expressed as follows,

$$P(Z = z) = p^z(1-p)^{(1-z)}, \text{ for } z \in \{0,1\}. \quad (7)$$

https://doi.org/10.31436/iiumej.v27i1.3824

In Eq. (7), the value of p represents the probability that the discriminator correctly identifies an image as real. Thus, we redefine p as $\hat{z}$. Applying the log-likelihood to Eq. (7) leads to Eq. (8).

$$L(z,\hat{z}) = z \log \hat{z} + (1-z) \log (1-\hat{z}). \quad (8)$$

In image-to-image translation, the generator attempts to learn a mapping between the X and Y domains [15]. Hence, when a real image Y is input to the discriminator, written as $\hat{z} = D(Y)$, the value of z should be 1. However, whenever the generated fake image $G(X)$ is input to the discriminator, written as $\hat{z} = D(G(X))$, the value of z should be 0. Applying each condition to Eq. (8) and summing up the results will provide a new equation called $V(D,G)$, where D refers to the discriminator, and G refers to the generator. The $V(D,G)$ equation is expressed in Eq. (9).

$$V(D,G) = \log D(Y) + \log \Big(1 - D\big(G(X)\big)\Big). \quad (9)$$

Since Eq. (9) originates from the Bernoulli distribution, it applies to a single observation. However, when dealing with multiple images, we extend it using expectation. Furthermore, assuming that pixel distributions in domain X follow p_X with parameter θ_X and those in domain Y domain follow p_Y with parameter θ_Y, the adversarial loss function can be formulated as Eq. (10). Other methods that utilize GAN have been proposed to generate more realistic images.

$$\mathcal{L}_{GAN}(G,D,X,Y) = \mathbb{E}_{Y\sim p_Y(\theta_Y)}[\log D(Y)] + \mathbb{E}_{X\sim p_X(\theta_X)}\Big[\log\Big(1 - D\big(G(X)\big)\Big)\Big]. \quad (10)$$

2.5. CycleGAN

A limitation of GANs is the requirement for paired data, which is often difficult to obtain. One unsupervised GAN method for two-domain translation without paired data is CycleGAN. Cycle consistency in CycleGAN ensures that images translated from one domain to another can be reconstructed to their original form. The optimization process in this method involves an objective function that shows the difference between the original input and the reconstructed version after translation. This function is the cycle-consistency loss.

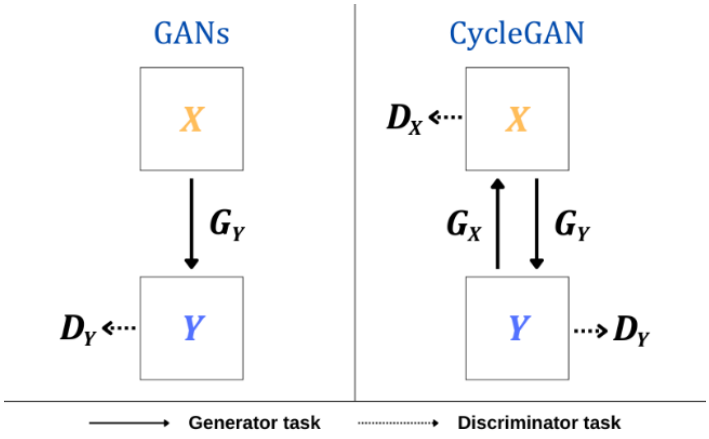

Figure 4. GAN and CycleGAN Framework Comparison

To facilitate this bidirectional translation, CycleGAN employs two generator-discriminator pairs. Suppose two generators are G_Y and G_X, then generator G_Y translates the image from the actual image of domain X to the generated image of domain Y, and generator G_X performs the inverse transformation from Y to X. Each generator is paired with a

https://doi.org/10.31436/iiumej.v27i1.3824

corresponding discriminator D_Y checks whether the output image of G_Y is real or fake according to domain Y, and vice versa for discriminator D_X. Both discriminators evaluate the realism of the generated image by comparing it with the real equivalent in each domain [1]. This bidirectional process aims to effectively learn the mapping between the two domains while preserving the critical image structure. Figure 4 illustrates the comparison between the conventional GAN and the CycleGAN. Unlike standard GAN, which performs one-way translation ($X \rightarrow Y$), CycleGAN introduces a bidirectional process ($X \leftrightarrow Y$) using two generator-discriminator pairs to enforce cycle consistency. This translation bidirectional process is to keep the generated image identical to the original image [1].

The collaborative operation of two generators and two discriminators of CycleGAN in maintaining cycle consistency is illustrated in Figure 5. The bidirectional translation process of G_Y and G_X ensures smooth connectivity across domains, thereby strengthening the principle of cycle consistency. This mechanism translates images between domains and reconstructs them to preserve structural integrity. By enforcing bidirectional consistency, CycleGAN ensures that translating an image to the target domain and then back yields a reconstruction that is similar to the original.

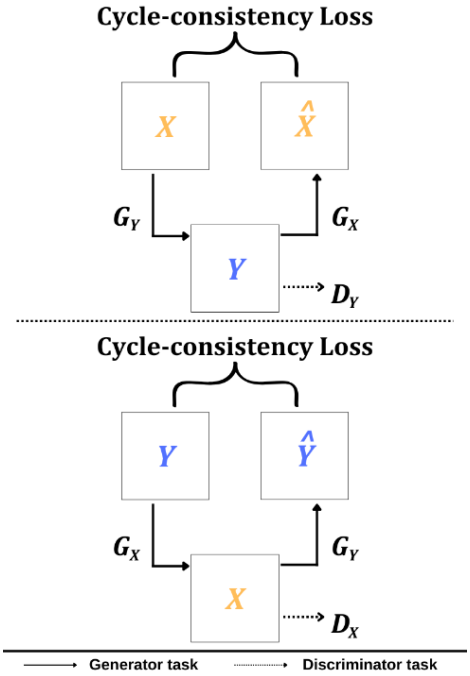

Figure 5. CycleGAN's cycle-consistency mechanism

The cycle consistency loss value, which is a quantification of the difference between the original and reconstructed images, is expressed in Eq. (11).

$$\mathcal{L}_{cyc}(G_Y, G_X) = \mathbb{E}_{X\sim p_X(\theta_X)}\left[\left\|G_X\left(G_Y(X)\right) - X\right\|\right] + \mathbb{E}_{Y\sim p_Y(\theta_Y)}\left[\left\|G_Y\left(G_X(Y)\right) - Y\right\|\right]. \quad (11)$$

In addition to the cycle-consistency loss function, another loss function must be considered: the identity-mapping loss. Identity mapping is used to ensure that the image maintains its original structure when processed by a generator from the same domain. The identity mapping is illustrated in Figure 6. In this process, when images are passed through their respective domain generators (X through G_X or Y through G_Y), they remain unchanged. This enforces an identity-mapping loss, ensuring that the generator in the same domain preserves the original structure.

https://doi.org/10.31436/iiumej.v27i1.3824

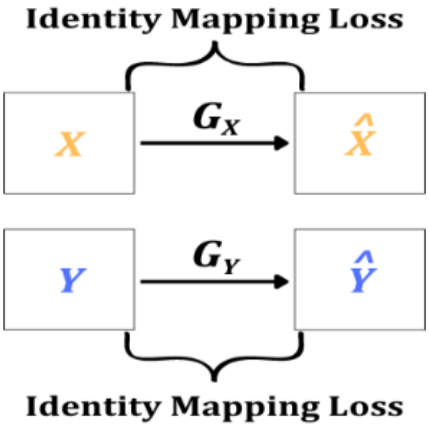

Figure 6. Identity mapping in CycleGAN

If an image from domain X is used as input to generator G_X, or when an image from domain Y is used as input to generator G_Y, then the output domain does not change. When the input and output domains are the same, the ideal transformation process should maintain the original characteristics of the input image. The identity mapping loss function is expressed in Eq. (12)

$$\mathcal{L}_{identity}(G_Y, G_X) = \mathbb{E}_{X\sim p_X(\theta_X)}[\|G_X(X) - X\|] + \mathbb{E}_{Y\sim p_Y(\theta_Y)}[\|G_Y(Y) - Y\|]. \quad (12)$$

Building on the principles of CycleGAN, this study incorporates three distinct loss functions to facilitate unpaired image-to-image translation. Given that CycleGAN employs two generator-discriminator pairs, each pair contributes an adversarial loss function. As a result, the overall loss function formulated in this study is represented in Eq. (13).

$$\begin{aligned} \mathcal{L}(G_Y, D_Y, G_X, D_X) &= \mathcal{L}_{GANYX} + \lambda\mathcal{L}_{cyc}(G_Y, G_X) + \frac{\lambda}{2}\mathcal{L}_{identity}(G_Y, G_X), \\ \mathcal{L}_{GANYX} &= \mathcal{L}_{GAN}(G_Y, D_Y, X, Y) + \mathcal{L}_{GAN}(G_X, D_X, X, Y). \end{aligned} \quad (13)$$

The value of λ (lambda) in CycleGAN serves as a weighting factor or hyperparameter that regulates the influence of cycle-consistency loss and identity mapping loss on the overall objective function. A high λ value prioritizes maintaining cycle consistency and identity mapping between the source and target domains, producing more structurally consistent translations at the potential expense of realism. Conversely, a low λ value places greater emphasis on generating visually realistic translations by assigning greater weight to adversarial loss, potentially sacrificing strict consistency across domains.

2.6. Spectral Normalization

Spectral Normalization (SN) is a regularization technique that mitigates unstable predictions arising from data distributions that are not well represented in the training data [16]. This technique normalizes the weight matrix W by using the most significant singular value of W as a scaling factor [17]. The weights in the convolutional layers of the discriminator network are the values within the filters (kernels) used to extract features. The normalization for the weight matrix W is given by Eq. (14)

$$\bar{W} = \frac{W}{\sigma(W)}, \quad (14)$$

where $\sigma(W)$ is the most significant singular value of W. The singular value is obtained from the singular value decomposition (SVD) of W. SN is more efficient compared to the gradient penalty method, making it suitable for large-scale data [16].

https://doi.org/10.31436/iiumej.v27i1.3824

2.7. Experimental Setup

This study was conducted using two image domains, each comprising 450 samples: a blue-sky domain and a sunset domain. All images were preprocessed by resizing to 128×128 pixels and normalizing pixel values. Since CycleGAN is trained under unpaired learning conditions, images from the two domains were shuffled and randomly paired during separate training.

The training process consisted of 100 epochs, each comprising 450 steps, for a total of 45,000 optimization updates. A batch size of 1 was applied, consistent with standard practice in CycleGAN research, as this setting supports stable adversarial training. Two discriminators and two generators were trained with the Adam optimizer with a learning rate of 2×10^{-4} and a momentum of 0.5. Adversarial learning was employed in the least-squares GAN (LSGAN) setting with the mean-squared error loss, and cycle-consistency and identity-mapping losses were introduced to enforce structural and color consistency. The right weight factors were used as $\lambda = 10$.

To improve generalization with a relatively small dataset, DiffAugment was applied during training, combining random color perturbations, spatial translations, and cutout operations. All experiments were conducted on a Windows 11 (64-bit) system without a dedicated GPU; all model training and evaluation were performed on an Intel i3-1005G1 CPU (1.20 GHz, 4 threads) with 8 GB RAM.

3. RESULTS AND DISCUSSION

3.1. Domain Characteristics

The data used in this study were taken from the Places365 dataset http://places2.csail.mit.edu/download.html and the Summer2Winter-Yosemite dataset https://www.kaggle.com/datasets/balraj98/summer2winter-yosemite. The characteristics of color intensity across different domains will yield distinct data distributions. To determine the dominant color and the colors in each domain, a descriptive analysis was conducted. Two domains are studied: the sunset and blue-sky domains. Each image in these domains consists of three primary color channels: red (R), green (G), and blue (B). The distribution of the three colors is depicted in Figure 7. The histograms indicate that the red color channel predominates in sunset images, whereas the blue channel predominates in blue-sky images, reflecting the natural scene characteristics of each domain. At sunset, the warm atmosphere on the beach is influenced by sunlight. This atmosphere produces a reddish-orange sky, which results in the dominance of red. Conversely, when the beach image is reflected in a blue sky, it accentuates the dark blue of the sky and sea, resulting in the dominance of blue in this area.

To increase the diversity of the training dataset and improve the model's generalization, data augmentation was applied via several randomized transformations. The augmentation parameters were set as follows: brightness adjustment randomly sampled between -0.5 and 0.5, saturation variation between 0 and 2, contrast adjustment between 0 and 0.5, spatial shift (horizontal and vertical) randomly sampled from -0.125 to 0.125 of the image dimensions, and cutout augmentation applied randomly to up to 50% of the image area. These transformations were designed to simulate a wide range of visual variations likely encountered in real-world conditions.

https://doi.org/10.31436/iiumej.v27i1.3824

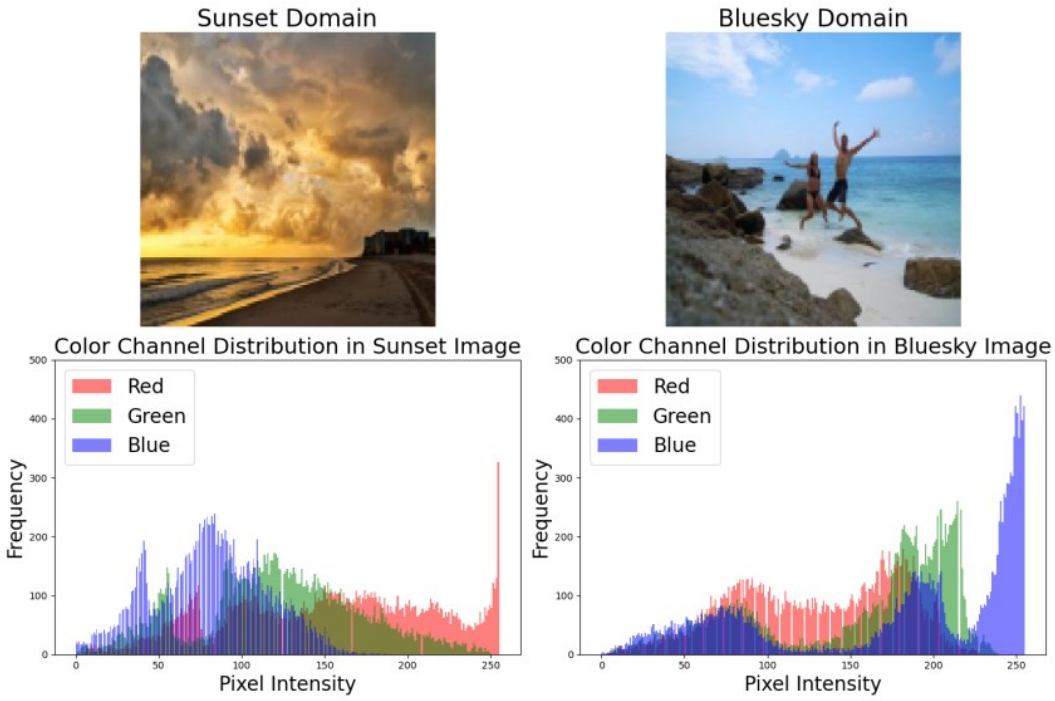

Figure 7. Color Distribution in Sunset and Blue-sky Domains

3.2. Model Architecture and Loss

In this study, the CycleGAN architecture comprises two generators and two discriminators for each translation task. The generator model is used to transfer sky ambiance between blue-sky and sunset conditions. The discriminator is used to classify whether images generated by the generator are real or fake. The architecture of the generator model used in this study is presented in Table 1, and the discriminator model is presented in Table 2. Generator networks include eight stride-2 convolutions with leaky ReLU activation functions and nine stride-1/2 convolutions with ReLU activation functions. To determine if the overlapping images were real, the discriminator networks used PatchGAN. Despite size variations, these architectural decisions align with those of prior studies [6].

In the CycleGAN architecture, different λ values were tried, namely 2, 5, and 10. The impact of changes in the λ value was observed, and the loss value for each λ was recorded. A comparison of the loss values for each model is given in Table 3. In the discriminator model, the model with a value of $\lambda = 10$ has the lowest loss value. This indicates that the discriminator in this model is most effective at distinguishing between real and fake images. Conversely, the loss value of the generator model with a value of $\lambda = 10$ has the highest loss value.

The λ hyperparameter is responsible for balancing the cycle consistency loss with adversarial loss and identity loss by controlling the weight of the cycle consistency loss. It is essential to recognize that lambda directly affects the generator's loss function; comparing generator performance solely by loss value would be misleading. Therefore, a more accurate assessment of the generator's quality should be based on the visual fidelity of the generated images. To improve the quality and stability of the generative model in CycleGAN with $\lambda = 10$, an MSCov layer was applied to the generator (Table 1). In contrast, SN was applied to the discriminator (Table 2). The MSCov layer uses 3 kernel sizes: 1x1, 3x3, and 5x5. To evaluate the model's performance, image translation experiments were conducted in two ways: (1) translating a blue-sky image to the sunset domain; and (2) translating a sunset image to the blue-sky domain. The quality of this translation was measured using MSE, SSIM, and LPIPS

https://doi.org/10.31436/iiumej.v27i1.3824

between the original image and the generated image after double translation (e.g., translating a blue-sky image to a sunset scene and then back to a blue-sky image).

Table 1. The architecture of the generator

Generator without MSCov Layer		Generator with MSCov Layer	
Layer Type	**Output Size**	**Layer Type**	**Output Size**
InputLayer	(None, 128, 128, 3)	InputLayer	(None, 128, 128, 3)
Sequential	(None, 64, 64, 64)	Sequential	(None, 64, 64, 64)
Sequential	(None, 32, 32, 128)	Sequential	(None, 32, 32, 128)
Sequential	(None, 16, 16, 128)	Sequential	(None, 16, 16, 128)
Sequential	(None, 8, 8, 128)	Conv2D	(None, 16, 16, 256)
Sequential	(None, 4, 4, 128)	Conv2D	(None, 16, 16, 256)
Sequential	(None, 2, 2, 128)	Concatenate	(None, 16, 16, 512)
Sequential	(None, 4, 4, 128)	Conv2D	(None, 16, 16, 256)
Concatenate	(None, 4, 4, 256)	Instance Normalization	(None, 16, 16, 256)
Sequential	(None, 8, 8, 128)	LeakyReLU	(None, 16, 16, 256)
Concatenate	(None, 8, 8, 256)	Sequential	(None, 8, 8, 128)
Sequential	(None, 16, 16, 128)	Conv2D	(None, 8, 8, 256)
Concatenate	(None, 16, 16, 256)	Conv2D	(None, 8, 8, 256)
Sequential	(None, 32, 32, 128)	Concatenate	(None, 8, 8, 512)
Concatenate	(None, 32, 32, 256)	Conv2D	(None, 8, 8, 256)
Sequential	(None, 64, 64, 64)	Instance Normalization	(None, 8, 8, 256)
Concatenate	(None, 64, 64, 128)	LeakyReLU	(None, 8, 8, 256)
Sequential	(None, 128, 128, 64)	Sequential	(None, 4, 4, 128)
Conv2D Transpose	(None, 128, 128, 3)	Sequential	(None, 2, 2, 128)
		Sequential	(None, 4, 4, 128)
		Concatenate	(None, 4, 4, 256)
		Sequential	(None, 8, 8, 128)
		Concatenate	(None, 8, 8, 384)
		Sequential	(None, 16, 16, 128)
		Concatenate	(None, 16, 16, 384)
		Sequential	(None, 32, 32, 128)
		Concatenate	(None, 32, 32, 256)
		Sequential	(None, 64, 64, 64)
		Concatenate	(None, 64, 64, 128)
		Sequential	(None, 128, 128, 64)
		Conv2D Transpose	(None, 128, 128, 3)

Table 2. The architecture of the discriminator

Discriminator without SN		Discriminator with SN	
Layer Type	**Output Size**	**Layer Type**	**Output Size**
InputLayer	(None, 128, 128, 3)	InputLayer	(None, 128, 128, 3)
Sequential	(None, 64, 64, 64)	Spectral Normalization	(None, 64, 64, 64)
Sequential	(None, 32, 32, 128)	Leaky ReLU	(None, 64, 64, 64)
Sequential	(None, 16, 16, 256)	Spectral Normalization	(None, 32, 32, 128)
Zero Padding 2D	(None, 18, 18, 256)	Layer Normalization	(None, 32, 32, 128)
Conv2D	(None, 15, 15, 512)	Leaky ReLU	(None, 32, 32, 128)
Instance Normalization	(None, 15, 15, 512)	Spectral Normalization	(None, 16, 16, 256)
Leaky ReLU	(None, 15, 15, 512)	Layer Normalization	(None, 16, 16, 256)
Zero Padding 2D	(None, 17, 17, 512)	Leaky ReLU	(None, 16, 16, 256)
Conv2D	(None, 14, 14, 1)	Spectral Normalization	(None, 8, 8, 256)
		Layer Normalization	(None, 8, 8, 256)
		Leaky ReLU	(None, 8, 8, 256)
		Spectral Normalization	(None, 4, 4, 256)
		Layer Normalization	(None, 4, 4, 256)
		Leaky ReLU	(None, 4, 4, 256)
		Spectral Normalization	(None, 4, 4, 1)

https://doi.org/10.31436/iiumej.v27i1.3824

The models are evaluated using the metrics MSE, SSIM, and LPIPS to measure the reconstruction accuracy and perceptual quality. Across all three measures, the proposed CycleGAN with regularization and MSCov improves network performance, as reported in Table 4. These findings confirm that applying regularization and MSCov significantly improves CycleGAN's translation quality.

Table 3. The loss values for each model

Losses	**λ**		
	2	**5**	**10**
Sunset Generator	0.9720	1.3730	1.7399
Blue-sky Generator	1.0018	1.3828	1.7473
Sunset Discriminator	0.6846	0.6557	0.6239
Blue-sky Discriminator	0.6756	0.6605	0.6297

Table 4. Model performance metrics

Metrics	**Basic CycleGAN**	**CycleGAN with SN + Augmentation + MSCov**
Blue sky - Sunset – Blue sky		
MSE*	0.0050	0.0008
SSIM**	0.7258	0.9592
LPIPS*	0.2180	0.0497
Sunset – Blue sky - Sunset		
MSE*	0.0044	0.0013
SSIM**	0.7750	0.9660
LPIPS*	0.2073	0.0831

*The lower, the better
**The higher, the better

3.3. Sunset Generator Performance

The CycleGAN model, with various λ values, translates a blue sky to a sunset. Overall, all generators perform well. The translation result of the sunset image produces a perfect atmosphere. The generator produces realistic sunset images. The blue sky is transformed into a convincing sunset atmosphere. The result is free from overlapping objects and errors. The color of the water and other objects in the generated image is successfully translated, naturally harmonizing with the sunset atmosphere. Figure 8 compares CycleGAN-generated sunset transformations across different λ values (2, 5, and 10), illustrating how the model transforms blue-sky beach scenes into sunset settings.

Furthermore, the cyclical consistency property of each object is maintained. The locations of the items and the shapes of the objects remain unchanged. The translation result is given in Figure 8. The figure shows a successful translation of blue-sky scenes into realistic sunsets while maintaining object integrity and spatial consistency. The water and surrounding elements change naturally with the sunset atmosphere. No artifacts or distortions are visible in the translation result. As the lambda value increases, the image quality improves.

Model 1 ($\lambda = 2$) produces a recognizable transformation but suffers from lighting consistency. The colors are also less well adapted. Model 2 ($\lambda = 5$) achieves better harmony between the sky and landscape. The translation results in this model are smoother than those in model 1. Model 3 ($\lambda = 10$) produces the most realistic results, with more lifelike colors, more natural lighting, and better reflections in the water and sky. These results indicate that

https://doi.org/10.31436/iiumej.v27i1.3824

higher lambda values are associated with greater cycle consistency, better structural preservation, and more accurate color adaptation.

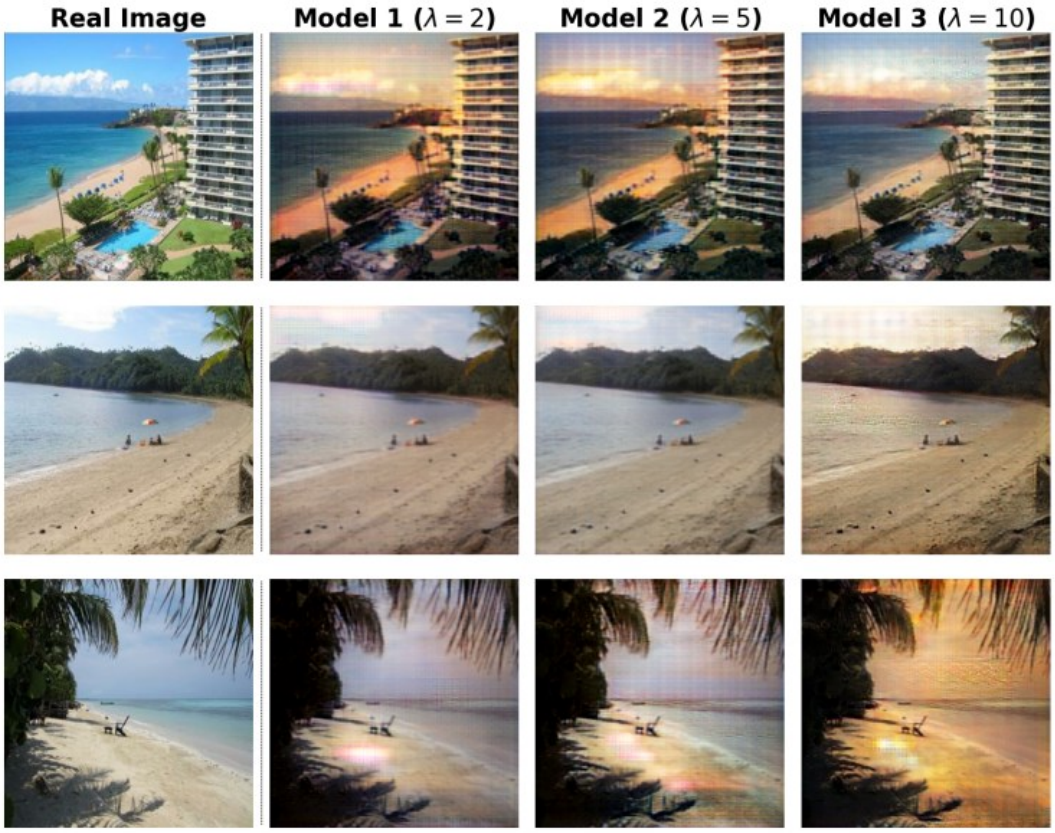

Figure 8. Results of sunset generator with different λ values

In the same domain as the previous case, an image with a human subject is presented in Figure 9 to evaluate the generator's effectiveness in translating images of human subjects, showing how different λ-values influence color adaptation and structural consistency. The sunset generator effectively integrates the human figure into the transformed scene without causing visual inconsistencies. Smooth color transitions, preservation of object shapes, and maintenance of spatial relationships indicate strong cycle consistency and realistic transformation.

The effects of increasing the lambda value are clearly evident, particularly in human-subject studies and complex scenes. Model 1 ($\lambda = 2$) implements the sunset effect well, but some color blends, especially around the human figure, appear less smooth, sometimes disturbing the natural reflections and shadows. Model 2 ($\lambda = 5$) improves the overall consistency, especially in the reflections of the sky and water. The sunset transformation results are more cohesive. Model 3 ($\lambda = 10$) produces the most realistic results. These results demonstrate better integration of human subjects, supported by improved lighting effects and more natural color adaptation. Model 3 ($\lambda = 10$) produces the most visually appealing and realistic results. The sunset scene is well integrated. Colors, textures, and structural details are preserved. This model shows excellent cycle consistency. Human contours and object placement are preserved. The resulting color transitions are also in harmony with the sunset scene. In addition, identity mapping helps maintain the original characteristics of objects within the same domain, prevents unnecessary changes, and enhances the realism of the transformation.

https://doi.org/10.31436/iiumej.v27i1.3824

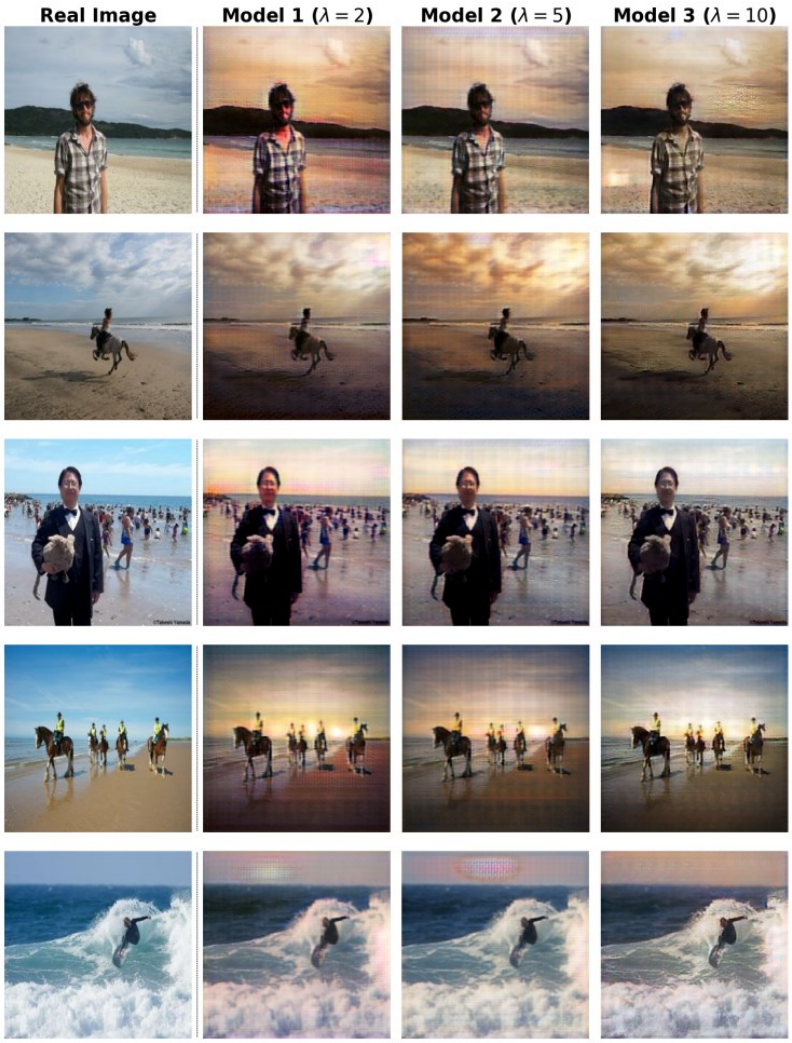

Figure 9. Sunset Generator Performance on Human Subjects

Figure 10. Comparison of CycleGAN and modified CycleGAN on the Sunset Domain

https://doi.org/10.31436/iiumej.v27i1.3824

The experiment to improve translation quality was continued by applying regularization and MSCov to the CycleGAN model with $\lambda = 10$. The comparison of translation results is given in Figure 10. The figure presents a side-by-side comparison between the baseline CycleGAN and the modified CycleGAN, highlighting the improvements in realism introduced by regularization and MSCov. The integration between regularization and MSCov improves the image quality and is more realistic with MSE, SSIM, and LPIPS values as summarized in Table 4.

3.4. Blue-Sky Generator Performance

The performance of CycleGAN in translating sunset images to blue sky under various λ values is illustrated in Figure 11, where sunset images are transformed into daytime conditions with restored lighting and atmospheric adjustments. In general, the generated images are consistent. The generated results demonstrate strong realism and cycle consistency. Similar to the sunset generator, the blue-sky generator effectively transforms the scene while preserving critical structural details. In the CycleGAN model, the generator and discriminator work in a conflicting framework. The training process aims to balance realism and structural preservation. If the generator outperforms the discriminator, an imbalance may arise. The result is an overly convincing image. Conversely, if the discriminator becomes very dominant, the generated image will be unrealistic.

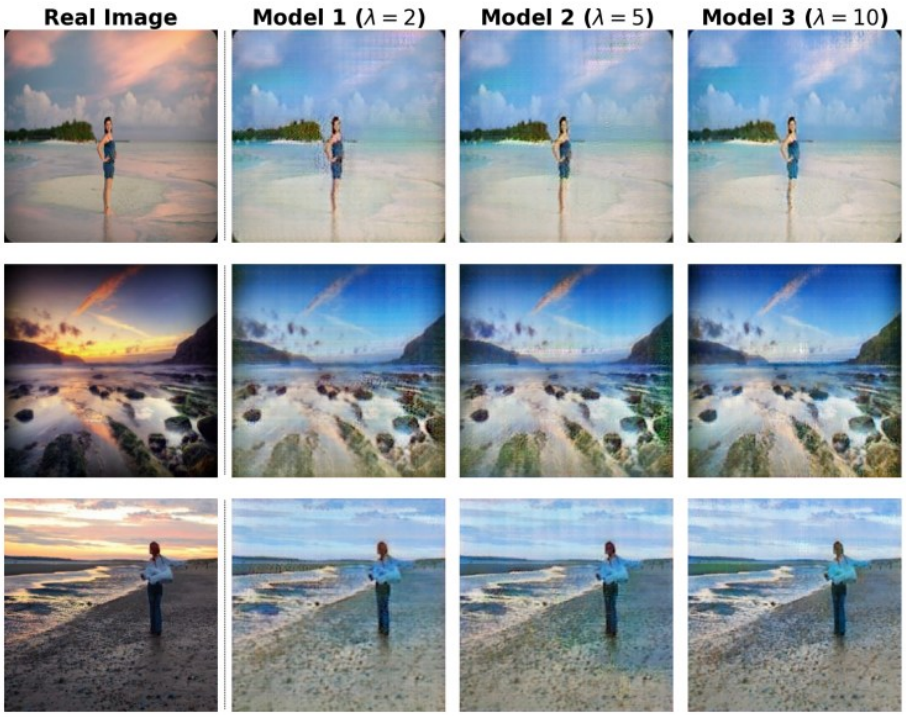

Figure 11. Blue-Sky Generator Results

Model 3 ($\lambda = 10$) has a slightly higher loss value than the other models. This is due to the increase in the lambda parameter. This small increase in loss is tolerable because the model's cycle consistency increases. This indicates a higher-quality, more visually appealing transformation. Therefore, Model 3 is the best.

Building on Model 3, augmentation is applied to the images, and CycleGAN is modified by adding SN and MSCov. A comparison of translation results before and after regularization and MSCov is shown in Figure 12. The figure compares the baseline CycleGAN with the modified CycleGAN on the blue-sky generator, highlighting improvements in realism such as

enhanced atmospheric clarity and the appearance of a sun element in the generated images. MSE, SSIM, and LPIPS values in Modified CycleGAN are also improved, as presented in Table 4.

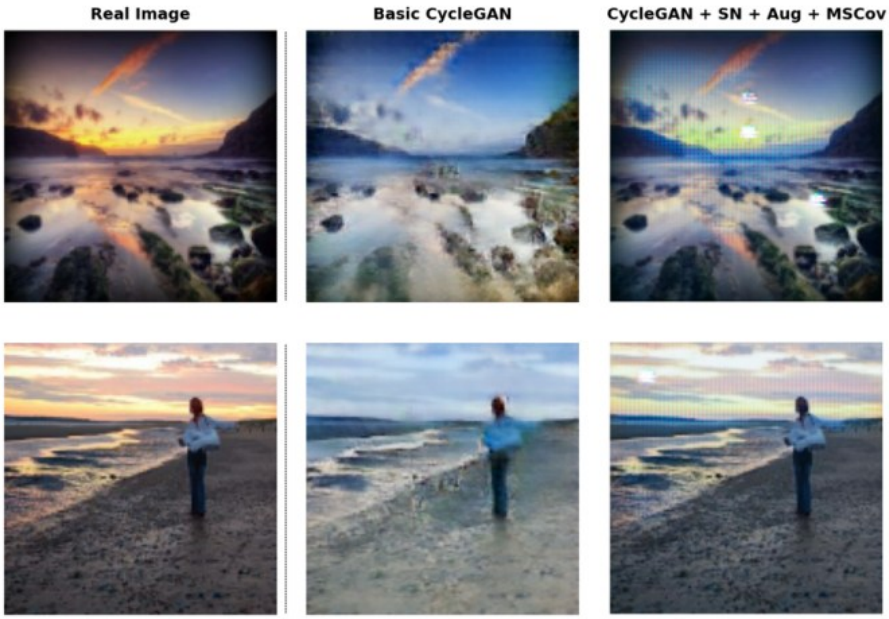

Figure 12. Comparison of CycleGAN and Modified CycleGAN on Blue-Sky Domain

3.5. Limitations

Although each generator appears effective at generating images, failures occur during the image-generation process. The failure cases of the sunset and blue-sky generators are illustrated in Figures 13 and 14, respectively. In Figure 13, the primary issue arises from the inaccurate translation of the shirt's white color. The generator has difficulty distinguishing the shirt's white from the clouds' white. The translation of cloud color affects the shirt's color. This results in the white color not being preserved, as it instead turns orange, reflecting the sunset sky. These cases illustrate how the sunset generator sometimes misinterprets white clothing as cloud formations, leading to undesirable color transformations. As a result, the transformation sacrifices realism by changing elements that should remain unchanged.

The failure of the blue-sky generator is of a different type, as shown in Figure 14. Dark objects appear in the sunset domain due to reduced lighting. The challenge in translating to blue-sky is to render dark objects as bright. The generator must adapt these objects to suit the new atmosphere. However, the generator often leaves them unchanged, resulting in inconsistencies. These failures reveal the blue-sky generator's difficulty in adapting dark objects to brighter conditions, which reduces the realism of the transformation. This demonstrates the limitations of the generator's ability to accurately interpret and adjust the texture and color of dark objects during the transformation process. Image translation failures mainly occur when color adaptation conflicts with learned patterns. Identity mapping is crucial because it helps the generator maintain structural integrity during image translation.

https://doi.org/10.31436/iiumej.v27i1.3824

Figure 13. Failure Cases in Sunset Generation

Figure 14. Failure Cases in Blue-sky Generation

4. CONCLUSIONS

This study investigates the transformation of blue-sky beach images into realistic sunset scenes using CycleGAN, an unpaired image-to-image translation framework. By leveraging adversarial learning and cycle-consistency constraints, the research demonstrates that CycleGAN effectively adapts images between distinct visual domains without requiring paired datasets. The experimental results, based on varying λ (2, 5, and 10), reveal that while higher λ values lead to increased generator loss, they also enhance image realism and structural preservation. This finding underscores that λ influences the generator's optimization process

but does not directly correlate with perceptual image quality, underscoring the importance of evaluating results using perceptual quality measures rather than loss metrics alone. This study showcases CycleGAN's practical potential for enhancing automated image editing and tourist imagery. The model with $\lambda = 10$ produced the most visually convincing results, reaffirming that careful hyperparameter tuning is critical for achieving high-quality transformations. In addition, the results of augmentation, SN, and MSCov indicate that the modified CycleGAN can generate more realistic images.

For the transformation of a blue-sky beach image into a sunset scene, the modified CycleGAN reduces MSE and LPIPS by 84% and 77%, respectively, while increasing SSIM by 32%. In contrast, for the sunset-to-blue-sky transformation, the proposed method reduces MSE and LPIPS by 70% and 60%, respectively, and increases SSIM by 25%.

A color channel histogram analysis confirmed that the dominant hues in the sunset and blue-sky domains, red and blue, respectively, align with real-world conditions, demonstrating the model's capacity to learn meaningful transformations. Additionally, the models effectively maintained cycle consistency, ensuring that key objects remained identifiable as they adapted to the target domain. However, limitations were observed: sunset generators occasionally misinterpreted white clothing as cloud formations, whereas blue-sky generators struggled to render the textures and colors of dark objects accurately. These challenges suggest that, despite robust identity-mapping constraints, certain object-specific transformations remain problematic.

This research contributes to the field of unpaired image-to-image translation by providing insights into how CycleGAN can be fine-tuned for complex environmental adaptations. Future work should focus on refining object-level transformations, particularly by improving the model's ability to distinguish between visually ambiguous elements. Investigating enhanced architectural modifications or hybrid approaches that integrate CycleGAN with other deep learning techniques could further improve realism and consistency. Additionally, extending this methodology to more diverse photographic scenarios would validate its broader applicability. Baseline comparisons with advanced models such as UNIT, MUNIT, or CUT, as well as exploring higher-resolution outputs beyond 128×128, could further improve applicability. By addressing these areas, future studies can advance image-to-image translation techniques and expand their impact across various real-world applications.

ACKNOWLEDGEMENT

This research was financially supported by Institut Teknologi Sepuluh Nopember through the Scientific Research Grant Scheme 2024 under Grant Number 1177/PKS/ITS/2024. The authors gratefully acknowledge this support.

REFERENCES

[1] Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. Proceedings of the IEEE International Conference on Computer Vision, 2223-2232. https://openaccess.thecvf.com/content_iccv_2017/html/Zhu_Unpaired_Image-To-Image_Translation_ICCV_2017_paper.html

[2] Tong, Z. (2024). Exploring the Impact of Hyperparameters on the Generation Quality of CycleGAN. Transactions on Computer Science and Intelligent Systems Research, 5, 265–271. https://doi.org/10.62051/01M93A63

[3] Hu, Y. (2024). Impact of Hyperparameters on the Quality of Image Translation Using CycleGAN. Transactions on Computer Science and Intelligent Systems Research, 5, 487–492. https://doi.org/10.62051/M04WSD55

[4] Zhao, S., Liu, Z., Lin, J., Zhu Adobe, J.-Y., & Song Han, C. (2020). Differentiable Augmentation for Data-Efficient GAN Training. Advances in Neural Information Processing Systems, 33, 7559–7570. https://proceedings.neurips.cc/paper_files/paper/2020/file/55479c55ebd1efd3ff125f1337100388-Paper.pdf

[5] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1), 1–48. https://doi.org/10.1186/S40537-019-0197-0

[6] Lu, Z., Pu, H., Wang, F., Hu, Z., & Wang, L. (2017). The Expressive Power of Neural Networks: A View from the Width. Advances in Neural Information Processing Systems, 30. https://proceedings.neurips.cc/paper/2017/hash/32cbf687880eb1674a07bf717761dd3a-Abstract.html

[7] Qin, H., Gong, R., Liu, X., Shen, M., Wei, Z., Yu, F., & Song, J. (2020). Forward and backward information retention for accurate binary neural networks. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2250–2259. http://openaccess.thecvf.com/content_CVPR_2020/html/Qin_Forward_and_Backward_Information_Retention_for_Accurate_Binary_Neural_Networks_CVPR_2020_paper.html

[8] Bottou, L., Curtis, F. E., & Nocedal, J. (2018). Optimization Methods for Large-Scale Machine Learning. SIAM Review, 60(2), 223–311. https://doi.org/10.1137/16M1080173

[9] Kingma, D. P., & Ba, J. L. (2014). Adam: A method for stochastic optimization. ArXiv Preprint ArXiv, 1412.6980. https://arxiv.org/abs/1412.6980

[10] Aggarwal, C. C. (2018). Neural networks and deep learning: a textbook. Springer. https://dlib.hust.edu.vn/handle/HUST/24439

[11] Javier, F., Morales, O., & Roggen, D. (2016). Deep Convolutional Feature Transfer Across Mobile Activity Recognition Domains, Sensor Modalities and Locations. Proceedings of the 2016 ACM International Symposium on Wearable Computers, 92 – 99. https://doi.org/10.1145/2971763.2971764

[12] Brahimi, S., Ben Aoun, N., Ben Amar, C., Benoit, A., & Lambert, P. (2018). Multiscale Fully Convolutional DenseNet for Semantic Segmentation. WSCG J, 26(2), 104–111. https://doi.org/10.24132/JWSCG.2018.26.2.5

[13] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. ArXiv Preprint ArXiv, 1409.1556. https://arxiv.org/abs/1409.1556

[14] Bello, A., Ng, S. C., & Leung, M. F. (2024). Skin cancer classification using fine-tuned transfer learning of DENSENET-121. Applied Sciences, 14(17), 7707. https://doi.org/10.3390/APP14177707

[15] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Nets. In Advances in Neural Information Processing Systems (Vol. 27). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2014/file/f033ed80deb0234979a61f95710dbe25-Paper.pdf

[16] Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). Spectral Normalization for Generative Adversarial Networks. 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings. https://arxiv.org/abs/1802.05957

[17] Yoshida, Y., & Miyato, T. (2017). Spectral Norm Regularization for Improving the Generalizability of Deep Learning. ArXiv Preprint ArXiv, 1705.10941. https://arxiv.org/abs/1705.10941