Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Engineering

Day-to-Night Image Translation with CycleGAN and Time-Lapse Training

Fergal Riordan

Supervisor: Dr. François Pitié

April 14, 2024

A dissertation submitted in partial fulfilment
of the requirements for the degree of
MAI (Electronic & Computer Engineering)

Declaration

I hereby declare that this dissertation is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`.

I have completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at `http://tcd-ie.libguides.com/plagiarism/ready-steady-write`.

I consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

I agree that this thesis will not be publicly available, but will be available to TCD staff and students in the University's open access institutional repository on the Trinity domain only, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Signed: Fergal Riordan Date: 14/4/2024

Abstract

Day-to-night image translation is a complex task that involves transforming an image of a daytime scene into a depiction of the same scene in night-time lighting conditions. This translation has applications in cinema post-production and the development of perception systems for autonomous vehicles. This study explores enhancements to a basic CycleGAN architecture for day-to-night image translation. Initially, the CycleGAN model was optimised by altering the generator architecture to a U-Net structure with a pre-trained ResNet-18 encoder, resulting in improved performance. Further refinements included an encoder-sharing scheme, where the two CycleGAN generators share a single ResNet-18 encoder and map via a shared latent space. This adaptation not only improved colour distributions but also increased the network's interpretability and flexibility for further modifications. Finally, the development of a single generator capable of handling both day-to-night and night-to-day translations via a timestamp parameter demonstrated promising results, which were further enhanced by incorporating time-lapse data into the training process. This comprehensive exploration not only highlights several enhancements to the basic CycleGAN model but also lays a foundation for future research into the refinement of day-to-night image translation techniques.

Lay Abstract

Transforming an image of a daytime scene to one depicting the same scene at night is a challenging task, often performed in cinema post-production and in developing the perception systems of self-driving cars. Computer programs, especially ones that use machine learning, can be developed to perform this operation automatically. Despite their potential, these programs often have issues such as unrealistic colours and other imperfections that render the images unusable. This study aims to enhance the quality of these artificially created night images by exploring modifications to the program's design. The program that we are using as a starting point consists of two separate systems that enable it to do two things: transforming day images into night images and transforming night images into day images. Most of the changes investigated in this study focus on sharing information between the two halves of the system, to see if this sharing of information improves performance. The final issue is whether it is possible to adapt the program to not only transform day images into night images and night images into day images but also to generate fake time-lapses. The result of this study is that some adaptations have been identified to improve the quality of the images that the program generates. Additionally, a basic method for generating fake time-lapses was developed, but further work is needed to make these time-lapses more convincing.

Acknowledgements

I would like to thank Dr. François Pitié for his patience, guidance and support throughout this project. I would also like to express my gratitude to John Squires, Clément Bled and Darren Ramsook for their assistance in overcoming certain obstacles in this research.

Finally, I would like to thank my parents for their support and encouragement throughout my academic journey.

Contents

List of Figures

List of Tables

Nomenclature

D_X	Input domain discriminator
D_Y	Target domain discriminator
F	Forward generator of CycleGAN network
G	Reverse generator of CycleGAN network
$\mathcal{L}_{\text{CycleGAN}}$	Overall training loss of CycleGAN
$\mathcal{L}_{\text{GAN}}$	Adversarial loss
$\mathcal{L}_{\text{cyc}}$	Cycle-consistency loss
X	Input image domain
Y	Target image domain
Z	Latent space
d	Dimension of deep feature representation
k	Kernel function
Σ	Covariance
β_1	Decay rate for the first moment estimates in Adam optimizer
β_2	Decay rate for the second moment estimates in Adam optimizer
λ	Cycle-consistency loss weight
λ_{i}	Identity loss weight
λ_{m}	Mid-cycle loss weight
μ	Mean
CycleGAN	Cycle-consistent generative adversarial networks
FID	Fréchet Inception Distance
GAN	Generative adversarial networks
IS	Inception Score
KID	Kernel Inception Distance
LADN	Local Adversarial Disentangling Network
LPIPS	Learned Perpetual Image Patch Similarity
OT	Optimal Transport
PSNR	Peak signal-to-noise ratio
SSIM	Structural Similarity Index
VAE	Variational auto-encoder

1 Introduction

Colour grading is the artistic process of altering the colour attributes of an image to match a target colour palette. It is a common step in cinema post-production for a variety of purposes: first to colour-correct shots to ensure continuity throughout a film, and then later to apply any desired effects, such as the conversion of scenes filmed during the daytime to night-time lighting conditions. Night-time scenes are often filmed during the day to provide better control over lighting and visibility, reduce production costs, increase convenience for the cast and crew, and enhance safety. Currently, colour grading in the post-production industry is performed by skilled professionals using sophisticated and costly software suites. Therefore, automating complex colour grading tasks like day-to-night translation could lead to substantial savings in both time and costs.

The automation of colour grading is often framed as a *colour transfer* task, where the goal is to alter the colour palette of an input image to match that of a reference image [1]. Traditional techniques like Optimal Transport (OT) have been applied with impressive results in simple colour transfer tasks [2] [3], however, they fail to capture the complex dynamics of tasks like day-to-night translation. Translating between daytime and night-time is challenging due to the changes in light sources that occur as a scene transitions from day to night. The presence of artificial light sources in night-time images leads to *colour inversions*, where some of the darkest regions in the daytime image may be among the brightest in the night-time image, and vice versa. Neural colour transfer displays improved performance in these more complex colour transfer tasks [4], however, the reliance on a reference image is not optimal for translations like day-to-night, which ideally should be achievable without relying on a specific reference.

1.1 Image-to-Image Translation and CycleGAN

To perform day-to-night translation without a reference image, it can be considered under the more general problem of *image-to-image translation*. In image-to-image translation, we seek to map an input image from its original domain to its representation in a target image domain. This provides a general framework that can be applied to a variety of image

processing tasks, and neural network-based approaches have been developed based on this principle for a multitude of image-to-image translations, ranging from trivial examples like converting images of horses to zebras to more practical tasks such as image super-resolution or matting in cinema post-production [5] [6].

The primary issue in day-to-night image translation is the lack of readily available paired datasets for training. Producing image pairs of the same scene during the day and at night is challenging due to the dynamic nature of most scenes. Objects, enter, leave and move within a scene over time. By the time a scene has completed the transition from day to night, there will likely have been significant changes in its underlying structure.

The lack of paired training data for day-to-night image translation means an unsupervised learning technique is required, and CycleGAN is an obvious choice in this regard. Cycle-consistent generative adversarial networks, or CycleGAN [7], was developed for image-to-image translation with unpaired training data. CycleGAN involves training two GANs simultaneously - one that learns the forward mapping from the input domain to the target domain, and another that learns the reverse mapping from the target domain to the input domain. This creates a network with a cyclical structure, where the generated image in the target domain can be translated back to the original input domain, thus completing a full *cycle* through the network. The purpose of this cyclical structure is to facilitate the calculation of a *cycle-consistency loss* term, which measures the difference between the input image and the image that is produced after a full cycle through the network. This loss term, coupled with the adversarial loss, acts as a replacement for the comparison of the generated image in the target domain against a ground truth image, which is only possible with paired training data.

CycleGAN has produced impressive results in a variety of image-to-image translation tasks and has been applied to day-to-night image translation before, specifically to synthesise training data for the perception system of an autonomous vehicle [8]. In this case, CycleGAN was used to translate daytime images with semantic labels into synthetic night-time images while retaining the semantic segmentation, enabling the training of an object detection model on semantically labelled night-time images without the need for explicit annotations in the night-time domain.

While CycleGAN can be a powerful tool for day-night image translation in cases where the night-time image fidelity is not the primary concern, the performance of the technique off-the-shelf is not strong enough for domains such as cinema post-production where image quality and the complete absence of visual artefacts is critical. In some instances, basic CycleGAN can produce high-quality synthetic night images, as shown in Figure 1.1, however, its performance is inconsistent. Artefacts such as the well-documented *checkerboard artefact* [9] frequently occur. Therefore, further refinement of this technique is required to

Figure 1.1: An example of successful day-to-night translation using the original CycleGAN implementation of Zhu et al. [7] The input day image (left) is translated to a synthetic night image (right).

ensure the consistent generation of realistic, high-quality outputs.

1.2 Adapting CycleGAN for Day-to-Night

One possible method of improving CycleGAN is to force the network to produce meaningful latent representations of the image's underlying content before decoding it to the target domain. Specifically, a *disentanglement* scheme may be employed to separate the domain-invariant content from the stylistic information, producing domain-agnostic latent codes. The decoder portion of the generator is then tasked with refilling the latent representation with domain-specific style information. This can be implemented for a variety of purposes, such as extending a CycleGAN network to map to a greater number of domains. In this research, the effect of disentanglement on overall output quality is under investigation. While a disentanglement scheme similar to the Local Adversarial Disentangling Network (LADN) of Gu et al. [10] was suggested in the Interim Report, the disentangling network that was ultimately implemented is closer to the StarGAN of Choi et al. [11] where only the domain-invariant image content is mapped to a shared latent space.

Another consideration in day-to-night image translation is the nature of the translation itself. While CycleGAN and similar techniques are typically applied to translate between distinct image domains, the translation of images between daytime and night-time stands out as a fundamentally different operation. Rather than existing within two distinct stylistic domains, daytime and night-time images exist along a continuous domain of outdoor lighting conditions. To only consider the extreme ends of this spectrum of lighting conditions fails to capitalise on the unique opportunities presented by such a continuous setting. Specifically, time-lapse data is a rich source of information that illustrates how a scene transitions from one end of the stylistic domain to another. To capitalise on this source of information, adaptations must be made to the network architecture to produce a network capable of

mapping to intermediate points along the spectrum of lighting conditions.

1.3 Research Objectives and Contributions

This research has three primary objectives: to use transfer learning to produce an optimal CycleGAN model for day-to-night image translation, to implement a disentanglement scheme to observe its effects on output quality and to adapt the CycleGAN architecture and training process to exploit time-lapse training data.

1.3.1 Improve CycleGAN with Transfer Learning

The first task is to incorporate a pre-trained ResNet-18 [12] encoder into the CycleGAN generator. This should lead to faster convergence and a reduction in visual artefacts. This optimised CycleGAN architecture stands as a contribution in its own right, but also as a baseline for comparison in the subsequent experiments.

1.3.2 Implement a Content-Style Disentanglement Scheme

The next task is to adapt the network architecture to use a single, shared encoder in the two CycleGAN generators. This will produce a more meaningful latent representation of the image content before decoding. A shared encoder should constrain the network so that the two generators encode to and decode from a single, shared latent space. This implies that when the encoder is passed an input image, the latent code that it outputs should contain only the domain-agnostic image content, while the domain-specific style information is discarded.

1.3.3 Translation Along a Continuous Stylistic Domain

To reflect the assumption that daytime and night-time images exist along a continuous stylistic domain in the network architecture, the encoder-sharing principle is extended to the decoder. In other words, a single, shared generator is used. The decoder portion of the generator takes an additional timestamp input, indicating the target lighting conditions. This architecture facilitates the incorporation of time-lapse data into the training process. The resulting network should be capable of generating synthetic time-lapses.

1.4 Report Structure

The remainder of this report will focus on the three primary research objectives. Chapter 2 will cover the current state-of-the-art and relevant literature for the topics addressed in this research. Chapter 3 will focus on the first research objective; improving a basic CycleGAN

model through transfer learning. Chapter 4 will introduce the encoder-sharing scheme. It will also investigate the performance of a timestamped generator in simple day-to-night translation. In Chapter 5, the timestamped generator will be trained to translate to intermediate points between daytime and night-time using time-lapse data. Finally, Chapter 6 will discuss the conclusions that can be drawn from this research and possible areas of future work.

2 Literature Review

Day-to-night image translation resides within a well-established problem space extensively covered in scientific literature. This discussion will delve into various facets of this topic, categorised under the following headings: colour transfer, image-to-image translation, CycleGAN, the mapping of latent variables, specific considerations within day-to-night image translation and quantitative metrics for model comparison.

2.1 Colour Transfer

Colour transfer as described by Reinhard et al. [1] refers to the task of finding the colour gradings to apply to an input image to match the colour palette of a target image. One approach to day-to-night image translation is to frame it as a colour transfer task. This requires two input images: a daytime image and a night-time image that represents the target colour distribution that is to be transferred to the daytime image.

A wide variety of approaches to colour transfer exist [13]. If there are pixel correspondences between the input image and the target image, the colour grading can simply be obtained through optimisation. In the absence of pixel correspondences, the grading can be obtained using moments matching or histogram matching. These simple techniques of matching colour distributions can provide satisfactory results in very simple colour correction tasks but are not sufficient in most cases due to the limitations associated with only considering colour distributions.

An alternative to these simple techniques is the use of the theory of Optimal Transport (OT). The link between OT and colour transfer was first made by Morovic and Sun [2] and Pitie et al. [3]. In OT as formulated by Monge (1781), the mapping between input and output must not only match the distributions but must do so while minimising its displacement cost. For colour transfer, this translates to matching colour distributions while minimising the overall colour changes. The original formulation of OT by Monge is limited in a colour transfer context by its one-to-one mapping constraint. However, Kantorovitch's OT formulation (1942) provides a relaxation of the one-to-one mapping by estimating a transportation plan estimating the proportion of pixels mapped to specific colours.

OT can produce compelling results in simple colour transfer tasks, especially when applied with the Matting Laplacian post-processing framework proposed by Levin et al. [14], which can significantly reduce the artefacts produced by the colour transfer operation.

The OT map is the gradient of a convex function, thus it prevents colour inversions. While in many contexts this is a desirable feature, it poses a major issue when dealing with day-night colour transfer: colour inversions are actually required in this transformation. OT's poor performance in complex colour transfer is also attributable to some of the assumptions that are inherent to the technique. For instance, OT assumes that the colour distributions of the input and target image should match, but this will not be the case in most contexts as the input and target image may depict entirely different scenes. Furthermore, in day-to-night colour transfer, the system must learn not only the colour mappings: accurately re-lighting an image requires an understanding of the scene's geometry, materials and lighting setup.

Day-to-night translation as a colour transfer operation was investigated by Pitie [13], and a selection of results of this experimentation are included in Figure 2.1. A baseline approximation of the OT solution [15] is compared against the same OT solution augmented with semantic masks, and a deep learning-based approach to style transfer [4]. The Matting Laplacian post-processing filter of Levin et al. [14] was applied to the outputs from the three techniques. The baseline approximation of the OT solution fails to perform the requisite colour inversions for day-to-night translation and therefore displays the weakest performance. The incorporation of semantic masks into the OT approach, where the technique is applied independently on each of the semantic labels, makes colour inversions possible and dramatically improves performance in comparison with the baseline approach. However, the neural colour transfer technique outperforms both OT-based techniques.

While the conceptualisation of day-to-night translation as a style or colour transfer operation is a valid approach, and the target image can be used to exercise more granular control over the specific colour palette and lighting conditions of the output image, the requirement of a target image is also a significant limitation of this approach. A method of translating a daytime image into its night-time representation without the need for a reference night-time image presents a more sophisticated and intuitive approach to the task. Therefore, while neural colour transfer appears capable of performing the translation effectively, a method of translating between day and night without the need for a reference image is desired.

2.2 Image-to-image translation

Day-to-night image translation, along with many other tasks in computer vision, can be conceptualised as an image-to-image translation, where an input image is translated to its

Figure 2.1: Results from formulating day-to-night translation as a colour transfer operation, from the work of Pitie [13]. The colour distribution/style of the destination image (DST) is to be transferred to the underlying content of the input image (SRC). IDT represents a baseline approximation of the OT solution. The subscript *s* indicates that semantic masks were used. DPST represents a deep photo style transfer approach. In all four cases, the neural colour transfer technique outperforms Optimal Transport.

representation in a target domain. Simple colour transfer methods approach these problems as per-pixel classification or regression, treating the output pixels as conditionally independent from all others given the input image. These methods require highly task-specific and carefully designed loss functions in order to produce satisfactory results.

The advent of generative adversarial networks [16] paved the way for an alternative approach. The key to the success of GANs is in the adversarial loss, which encourages the generator output to be indistinguishable from real examples in the target domain. Thus, GANs remove the need to explicitly define highly task-specific loss functions, providing a flexible framework for a variety of computer vision tasks.

Specifically, conditional GANs (cGANs) are useful in image-to-image translation. By conditioning the GAN on an input image, we hope to produce an output that aligns with the desired transformation on the input image. Isola et al. [17] implemented cGANs in this way in their *pix2pix* framework to provide a general solution for image-to-image translation in a supervised setting.

In many cases, there is a lack of paired training data and thus an unsupervised technique is

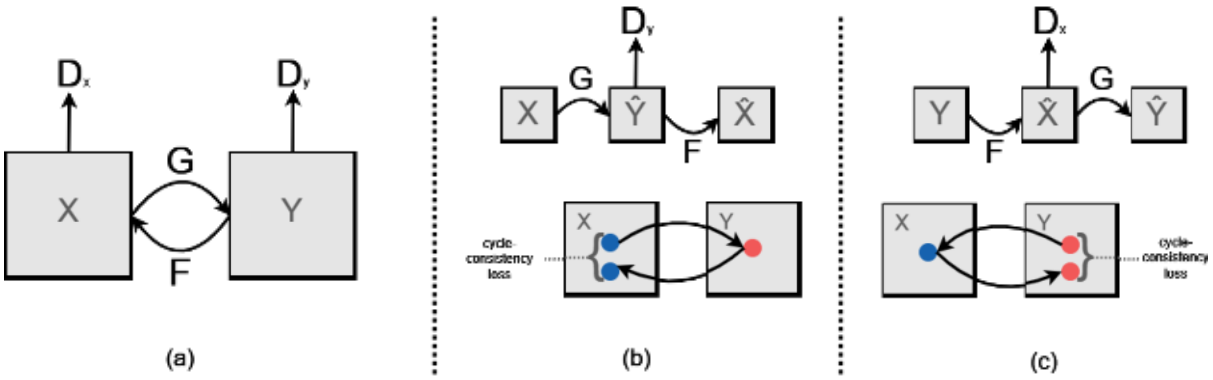

Figure 2.2: CycleGAN as formulated by Zhu et al. [7] The model contains two mapping functions $G : X \rightarrow Y$ and $F : Y \rightarrow X$ and two adversarial discriminators D_X and D_Y. To constrain the mappings, two cycle-consistency losses are used to ensure that if an image is translated from one domain to the other and then back again, it should arrive back at its original representation. Forward cycle consistency: $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$. Backward cycle consistency: $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$.

required. A variety of approaches to unsupervised image-to-image translation exist. One approach proposed by Resales et al. [18] is a Bayesian technique for predicting the most likely output image based on a patch-based Markov random field obtained from the source image and a likelihood term learned from sample images in the target domain. More recent approaches are typically GAN-based, such as CoGAN [19] which uses a *coupling term* in the objective function and shares weight parameters corresponding to high-level features in the generators and discriminators in order to learn a joint distribution over images from the two domains.

2.3 CycleGAN

The introduction of cycle-consistent adversarial networks (CycleGAN) by Zhu et al. [7] provided a powerful general-purpose tool for unpaired image-to-image translation by introducing the concept of *cycle-consistency loss*.

Given a set of sample images in an input domain X and a separate set of sample images in the target domain Y, we can train a cGAN to learn a mapping F that translates the input set X to an output set $\hat{Y}$ that is distributed identically to Y. The issue with this training setup in isolation is that there are an infinite number of mappings that can be learned that produce the correct output distribution, but there is no guarantee that a given input will have a meaningful relationship with its generated output and there is nothing to prevent mode collapse from occurring.

Further constraints must be added to the network to enforce a meaningful relationship between input-output pairs. A solution is to simultaneously train a reverse mapping, G, from

Figure 2.3: Example applications of CycleGAN from the original work of Zhu et al. [7] The bidirectional nature of the network means that the mapping between two image domains is learnt in both directions. From left to right, CycleGAN is applied successfully for the following translations: Monet painting ↔ Photograph, Zebra ↔ Horse, Summer ↔ Winter.

the target distribution Y back to the input distribution X, as illustrated in Figure 2.2. A cycle consistency loss term representing the difference between x and G(F(x)) can then be calculated (encourages F(G(x)) $\approx x$ and G(F(y)) $\approx y$). The network is trained to minimise this term, encouraging the network to return each training input to its original form after first translating it to the target domain. A CycleGAN network can therefore be thought of as two autoencoders working simultaneously, where the bottleneck of the the autoencoder is the representation of the input image in another domain.

The full objective of CycleGAN therefore combines the adversarial losses with the cycle consistency loss term.

$$\begin{aligned}\mathcal{L}_{\text{CycleGAN}}(G, F, D_X, D_Y) =& \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) \\ &+ \lambda \cdot \mathcal{L}_{\text{cyc}}(F, G)\end{aligned} \tag{2.1}$$

Other losses such as an identity loss term can also be added to improve feature preservation. However, even without any additional loss terms, CycleGAN and similar approaches that also exploit a cycle consistency loss such as DiscoGAN [20] and DualGAN [21] produce impressive results for a wide variety of translations, as displayed in Figure 2.3.

2.4 Mapping of Latent Variables

The task of image-to-image translation is often framed as the encoding of an input image to a latent representation from which it can be decoded to its representation in a new domain. The manner in which these latent representations are constructed can have significant effects on the utility of the resulting network. CycleGAN maps images from the input domain and target domain into separate latent spaces, but this is not the only possible approach.

2.4.1 Direct Augmentation of the CycleGAN Latent Space

While the original CycleGAN network architecture results in a deterministic, one-to-one mapping, in many applications a stochastic mapping is required. Almahairi et al. [22] address this by proposing Augmented CycleGAN, where the latent spaces of the forward and reverse mappings are augmented with auxiliary latent spaces. By sampling from these auxiliary latent spaces with a Gaussian prior during the mapping process, a stochastic mapping is achieved, where a single input can produce a variety of output images in the target domain.

2.4.2 A Shared Latent Space

A significant body of work exists based on the concept of mapping to a shared latent space. The belief is that through the use of a shared latent representation, the domain-specific image attributes are discarded and only the domain-invariant information is retained. UNIT [23] builds the assumption of a shared latent space made in CoGAN [19] into a CycleGAN architecture. It uses a variational autoencoder (VAE) loss term [24] to push the latent vectors at the centre of the generator into a Gaussian distribution. Thus, much like in Augmented CycleGAN, a diverse set of outputs can be produced for each input image.

While the one-to-one mapping constraint of CycleGAN was overcome by these techniques, the mapping is still restricted to only two domains at a time. ComboGAN [25] addresses this by decoupling the encoders and decoders and training a single encoder and decoder for each domain. While in the two-domain case ComboGAN is identical to CycleGAN, the network scales linearly with additional domains, eliminating the need to train a dedicated CycleGAN for each mapping. For ComboGAN to work as intended, it is implied that as the network is extended to additional domains the encoders must be mapping to a shared latent representation. The decoders then learn to refill the domain-agnostic latent representation with the necessary details that define the characteristics of the target domain.

StarGAN [11] as formulated by Choi et el. also addresses the mapping to multiple domains but approaches the task in a different way. In StarGAN, the generators and discriminators of CycleGAN are melded into a single GAN, with the output domain determined by a vector that is passed to the network along with the input image. The discriminator is trained to output the domain of each image along with the real/fake label. Therefore, in both StarGAN and ComboGAN a similar latent mapping is achieved: a shared, domain-invariant latent representation that facilitates the decoding of images to multiple separate domains. StarGAN is of particular interest in the context of day-to-night translation, as it was developed for the transfer of facial features and facial expressions. The technique models each feature or expression as a separate domain, but this is a very similar task to the translation to intermediate points between daytime lighting and night-time lighting

conditions. The effectiveness of the single generator in StarGAN in mapping to closely related domains sets an example for day-to-night translation, suggesting that a similar strategy could be effective in the generation of synthetic time-lapses.

2.4.3 Disentangled Representations

While a shared latent representation facilitates the retention of domain-invariant information, in many cases the domain-specific information should also be retained. This is the motivation for *disentangled* latent representations, where the domain-invariant *content* and domain-specific *style* are mapped to separate latent spaces.

MUNIT [26], a direct follow-up to UNIT, seeks to extend the network for many-to-many mappings much like in ComboGAN and StarGAN. In addition to the shared latent representation of the image content, the image style is mapped to a separate latent space. Style information is randomly sampled from this latent space when mapping to the given domain. Lee et al. [27] leverage disentanglement in a similar manner to increase the diversity of the possible outputs for each input image.

Another motivating factor for disentangled representations is to boost the interpretability of the network. InfoGAN [28] is an example of a disentangling network that was developed with this goal in mind. The network is encouraged to learn interpretable and meaningful latent representations by maximising the mutual information between a subset of the GAN's noise variables and the observations. Similarly, Beta-VAE [29] attempts to provide disentangled latent representations, and introduces a parameter, *beta*, to control the degree of disentanglement. Cao et al. [30] propose a method of disentanglement analysis to improve the level of disentanglement in the latent space to improve the quality of the resulting mappings.

Finally, the Local Adversarial Disentangling Network (LADN) proposed by Gu et al. [10] uses disentanglement in a style transfer operation. Specifically, disentanglement is applied to facial makeup and de-makeup, with an asymmetric loss scheme that encourages the mapping of domain-invariant image content (facial features) to one latent space, and the style information (makeup) to a separate space. Much like in MUNIT, the retention of this style information facilitates the sampling of these features, resulting in a network that is capable of swapping makeup between two input images. This disentanglement scheme, coupled with a system of multiple, overlapping discriminators focusing on specific image regions, produces compelling results in the style transfer task.

2.5 Day-to-Night Image-to-Image Translation

Day-to-night image translation as a problem in its own right has a limited body of research. The translation generally arises as a subset of broader image-to-image translation or as an intermediate step in the development of other techniques, such as generating synthetic data for training night-time perception systems [8] [31].

A greater body of work exists focused on night-to-day translation, primarily for the perception systems of autonomous vehicles. For example, ToDayGAN was introduced to convert night-time images to daytime representations for improved visual localisation in robotics systems. It builds upon the ComboGAN architecture with specialised discriminators to focus on specific aspects of input images such as texture, colours and gradients [32].

While the literature focused solely on day-to-night image translation is limited, the translation has been addressed under the general concept of translating along continuous domains. Pizzati et al. [33] proposed considering day-to-night transitions as non-linear, *cyclic* translations. They attribute the failures of CycleGAN and similar techniques in this task to the assumption of piece-wise or entirely linear domain manifolds. To address this, they proposed CoMoGAN, a novel image-to-image translation framework specifically designed for non-linear continuous translations on unsupervised target data.

The key innovation in CoMoGAN is its use of a physics-inspired model to enable non-linear transformations between day and night images. By considering day-to-night transitions as cyclic translations rather than linear mappings, CoMoGAN aims to overcome the limitations of linear assumption-driven models. The technique also incorporates a disentangling mechanism, relaxing model dependency via a continuous disentanglement of domain features. Without this measure, the technique would simply learn to mimic the model, but this is avoided by allowing the target domain and model domain to have shared model features but also private non-modelled features

The technique outperformed the literature on test datasets but is not without its limitations. It faces challenges when dealing with complex real-world scenarios due to the inherent simplifications of the physics-inspired models it employs. The oversimplifications of these models could potentially limit CoMoGAN's ability to handle diverse lighting conditions and other intricate details inherent in day-to-night transitions.

2.6 Quantitative Metrics

The goal of this work is to produce synthetic images that are indistinguishable from real images to a human observer. Therefore, metrics that attempt to mimic the perceptual traits of the human visual system are required, as opposed to traditional metrics like peak

signal-to-noise ratio (PSNR) which, being a pixel-based metric, may not accurately reflect human perception of image quality. One such *perceptual metric* is the Structural Similarity Index (SSIM) [34], which measures the perceived change in structural information as a result of an operation on an image. More recent perceptual metrics such as the Learned Perceptual Image Patch Similarity (LPIPS) [35] are based on the feature representations from pre-trained convolutional neural nets. However, both SSIM and LPIPS are *full reference metrics*, meaning they require a reference image for direct comparison. Therefore, when evaluating output image quality in an unsupervised setting (as is the case with CycleGAN and similar models) these metrics are not applicable.

For the unsupervised setting, a variety of perceptual metrics exist that seek to compare the distributions of synthetic and real images. Two prominent metrics of this type are the Inception Score (IS) [36] and the Fréchet Inception Distance (FID) [37], both of which are based on the high-level features extracted by the Inception V3 model [38] trained on ImageNet [39].

The Inception Score evaluates the quality of a set of generated images using probabilities calculated by a pre-trained Inception model. It aims to measure both the clarity of objects within the generated images and the diversity of the image distribution. It accomplishes this by considering two key aspects: the precision in object representation within individual images and the overall diversity across the image set. Specifically, the Inception Score examines the conditional probability of class labels given a generated image (reflecting object clarity) and the marginal distribution of class labels across all generated images (reflecting the diversity of the set). The Inception Score is defined as the Kullback-Leibler divergence between the two probability distributions, therefore a large score indicates a high-quality output image set, resulting from low entropy associated with the first distribution and high entropy associated with the latter.

While the Inception Score only considers the distribution of generated images, the FID builds on this by comparing the distribution of generated images to a distribution of real images. This comparison is facilitated by the assumption that the features computed by the Inception model for both real and generated images have a Gaussian distribution. Thus, known metrics for Gaussians can be used; specifically the Fréchet distance between two multivariate Gaussian distributions. For both real and synthetic data, Gaussian distributions are fitted to the features extracted by the Inception model. The Fréchet distance between the real and generated distributions is explicitly solvable using their means and covariances in the following formula:

$$FID = ||\mu_r - \mu_g||^2 + \mathrm{Tr}\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r\Sigma_g\right)^{\frac{1}{2}}\right) \tag{2.2}$$

While the Inception Score and FID have been widely used as evaluation metrics since their introduction, some significant issues have been found with these techniques. Chong et al. [40] found that for both the Inception Score and FID, the expected value of the score computed for a finite sample set is not the true value of the score. They also found that the degree of this *bias* depends on the model in question, thus rendering the metrics highly unreliable for the comparison of model architectures. To address this, they proposed IS_∞ and FID_∞; metrics derived from a method of extrapolating the scores generated from a finite set of samples to *bias-free* estimates of the scores for an infinite number of samples.

An alternative metric that is also derived from the Inception model is the Kernel Inception Distance (KID) [41], which does not rely on the Gaussian distribution assumption inherent to the FID. KID uses the Maximum Mean Discrepancy (MMD) to compare the feature distributions of the real and synthetic images directly, without the need to estimate the distributions themselves. This is achieved using the *kernel trick* with a polynomial kernel function. The kernel that was originally proposed is a cubic kernel, defined as:

$$k(\mathbf{x}, \mathbf{y}) = \frac{1}{d}(\mathbf{x}^T\mathbf{y} + 1)^3 \tag{2.3}$$

where d represents the dimension of the deep feature representation. While the KID is purported to be more suitable for smaller image sets than the Inception Score and FID, its significant variance limits its effectiveness in model comparison [42], therefore it must still be applied judiciously when comparing models.

A recent investigation into the evaluation of generative models by Betzalel et al. [43] included the comparison of a variety of metrics including IS, FID, KID, IS_∞ and FID_∞. They found that while these metrics generally correlate with improved performance, their rankings of models of a similar quality varied significantly, rendering them unreliable for the fine-grained comparison of models. They found that KID outperformed both IS and FID, and that IS_∞ and FID_∞ were significantly superior to their finite equivalents. Additionally, they scrutinised the reliance of these perceptual metrics on the pre-trained Inception network and found that using features from the Contrastive Language-Image Pre-Training (CLIP) neural network [44] outperformed the Inception model for datasets other than ImageNet. Among their recommendations, they include dropping the Inception Score and using multiple metrics (such as FID_∞ and KID) to mitigate the issues associated with using these metrics in isolation.

2.7 Conclusions

The scientific literature surrounding the topic of day-to-night image translation encompasses various facets, each offering insights and limitations within this field. Simple colour transfer

techniques such as Optimal Transport are useful for simpler transformations but are not sophisticated enough for day-to-night colour transfer. GAN-based image-to-image translation, particularly CycleGAN, provides a flexible framework for this task. However, successful adaptation of this framework requires thoughtful adjustments in parameters, especially concerning the mapping of latent variables. In the case of day-to-night translation, which can be viewed as a single, continuous domain, it is possible that a shared latent representation similar to the one implemented in StarGAN could be suitable. Finally, an exploration of numeric performance metrics for generative models underscores the necessity of cautious application, with the optimal approach appearing to be the application of a combination of metrics with careful interpretation of their results, emphasising their role as supplementary to qualitative analysis.

3 Improving the CycleGAN Baseline for Day-to-Night Translation

The generator architecture used in the original CycleGAN implementation of Zhu et al. [7] is adapted from the work of Johnson et al. [45] who implemented a network for style transfer and super-resolution that exploits residual blocks. Provisional experimentation with this generator architecture for day-to-night translation indicated a poor understanding of the semantic content of scenes, with basic image regions such as the sky being assigned incorrect and inconsistent colours. Therefore, alternative generator architectures were considered. An obvious choice in this regard is a U-Net architecture [46], which was initially introduced for semantic segmentation in a biomedical context but has displayed strong performance in a wide variety of semantic segmentation tasks [47] [48]. While explicit semantic segmentation is not required for day-to-night translation, a generator architecture that is well-suited to extracting semantic content may ensure that basic image features (sky, buildings, grass, water, etc.) are translated properly.

An additional motivation for changing to a U-Net generator architecture is that this architecture has an encoder-decoder structure, which facilitates transfer learning: the encoder portion of the U-Net can be easily substituted with a pre-trained network [49]. The datasets available for training a day-to-night translation model are quite limited, but through the use of a pre-trained encoder which has been trained on a large set of generic images, the bulk of the training task with the day and night data is limited to the decoder portion of the network, which hopefully leads to better results. The ResNet-18 model [12] was deemed a suitable choice for this transfer learning, due to its size (roughly equivalent to the existing encoder) and its training on generic ImageNet data.

A final reason for changing to a U-Net generator architecture is that the encoder-decoder structure aids in the experimentation with encoder sharing in Chapter 4.

3.1 Proposed Improvements

3.1.1 Network Architectures

As discussed, the first proposal is to replace the original CycleGAN generator with an alternative architecture. The proposed alternative is a U-Net architecture, which can exploit transfer learning through the inclusion of a pre-trained encoder. To investigate the effects of these changes, three generator architectures were implemented: the original CycleGAN generator, a basic U-Net generator, and a U-Net with a pre-trained ResNet-18 encoder.

Original CycleGAN Generator

The original CycleGAN network consists of three initial convolutional layers a series of residual blocks (the number of which varies depending on the resolution of the training image size), two fractionally strided convolutions with a stride of $\frac{1}{2}$ and a final convolution to map features to RGB.

U-Net Generator

As previously discussed, an alternative architecture is a U-Net structure. This architecture is known to be especially adept at semantic segmentation but can be applied to a variety of generative tasks. The U-Net generator has an encoder-decoder structure, with the encoder capturing contextual information while reducing the spatial dimensions of the input, and the decoder decoding the encoded latent representation to an image in the target domain. The U-Net also employs skip connections between the encoder and decoder at each level of spatial dimension to aid in the decoding process and to improve feature preservation.

The encoder therefore consists of a series of convolutional layers, with periodic down-sampling by max-pooling. Each level of the decoder consists of a bilinear up-sampling step followed by concatenation of the up-sampled tensor with the information from the corresponding skip connection from the encoder, before input to a pair of convolutional layers. A final convolutional layer at the top level maps the features to RGB.

U-Net with Pre-Trained ResNet-18 Encoder

A popular adaptation of the basic U-Net structure is to replace the encoder portion of the network with a pre-trained network. The ResNet-18 model is a common choice for this transfer learning [50] [51]. The ResNet-18 architecture, as illustrated in Figure 3.1, consists of a series of residual blocks. To adapt this model for use in the image-to-image generator, the final residual layer, average pooling step and fully connected layer were discarded, and the preceding portion of the network was used as the encoder portion of the

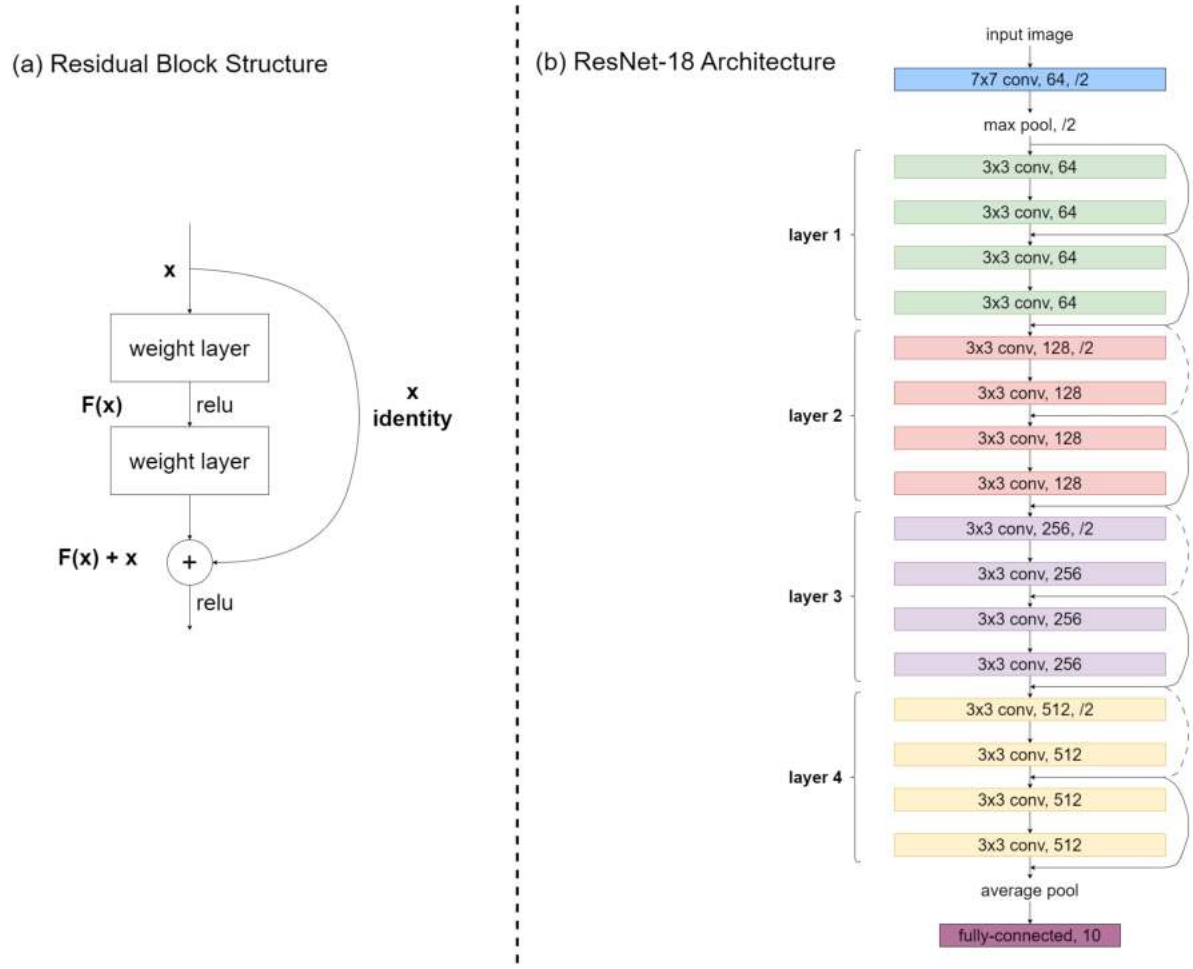

Figure 3.1: Residual block structure and the overall ResNet-18 architecture. (a) A residual block uses a skip connection to bypass its weight layers, which mitigates the issue of vanishing and exploding gradients, facilitating the training of deeper networks. (b) ResNet-18 consists of four residual *layers*, each with two residual blocks.

U-Net. This ensured that the generator size was proportionate to the rest of the network, which is crucial to ensure proper convergence during training.

The decoder portion of the generator is unchanged from the other U-Net decoder apart from the absence of a skip connection at the uppermost level due to the immediate down-sampling effect of the first convolutional layer in the ResNet-18 model. This adapted generator architecture is contrasted against the other two generators in Figure 3.2.

While some provisional experimentation was performed using the CycleGAN example provided by Keras [52], the code used to generate the results detailed in this report was adapted from the PyTorch example provided at medium.com [53]. The U-Net generator drew inspiration from a variant detailed on GitHub [54].

Instance normalisation is used in all three of the generators, therefore the normalisation of the pre-trained ResNet-18 encoder was changed from batch normalisation to instance

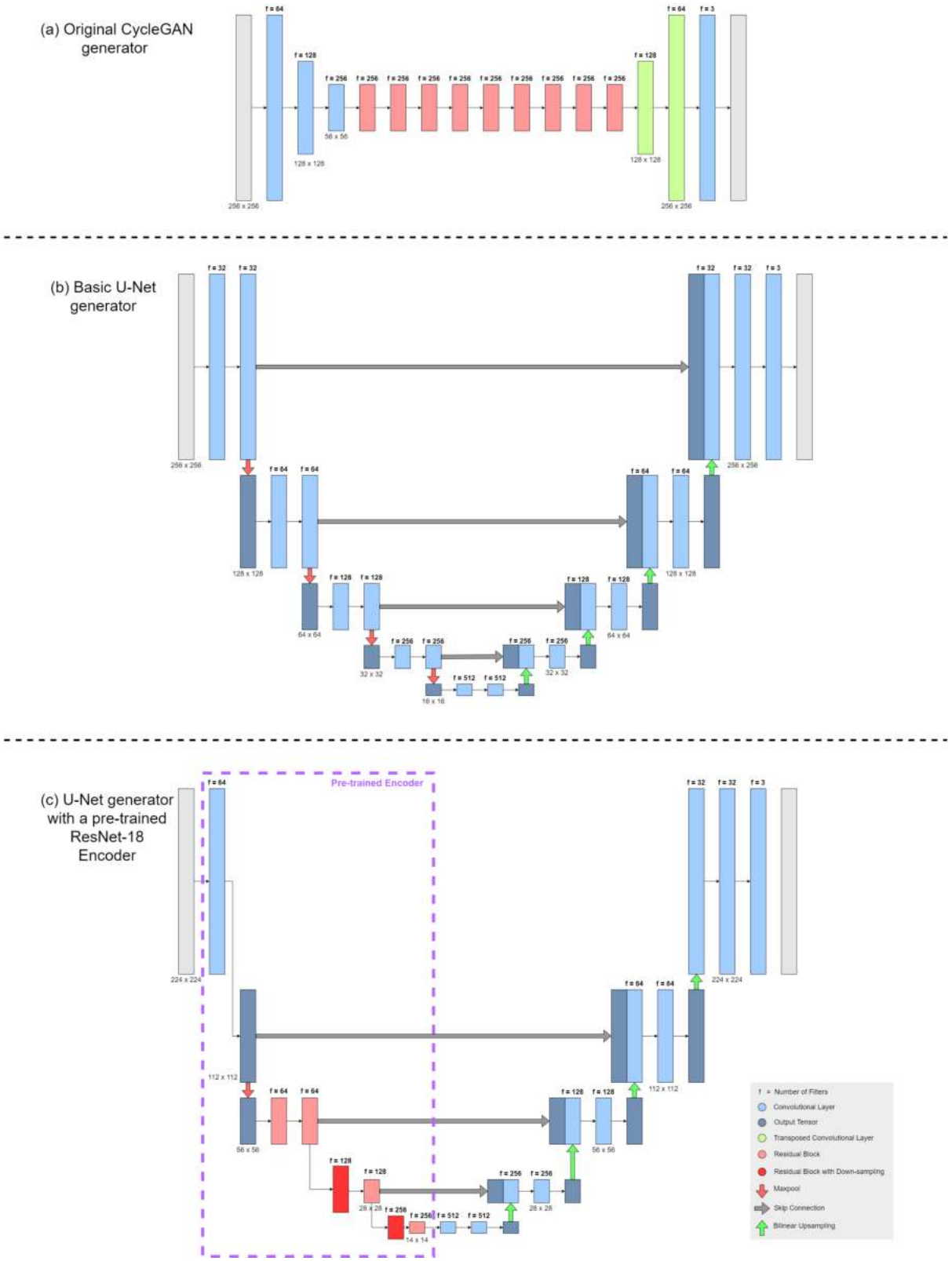

Figure 3.2: The three generator architectures that were implemented for comparison. The encoder-decoder structure of the U-Net generator facilitates the inclusion of a pre-trained ResNet-18 model. (a) The original CycleGAN generator architecture (b) U-Net generator architecture (c) U-Net with a pre-trained ResNet-18 encoder

normalisation to maintain consistency across the network. Finally, as previously discussed, when adapting the ResNet-18 model for use in the U-Net, the final residual layer (layer 4 as illustrated in Figure 3.1) was not included due to the need to keep the generator proportionate to the rest of the network. Including layer 4 would greatly increase the size of the resulting U-Net, resulting in poor convergence during training due to the disproportionate generator compared to the rest of the network. Furthermore, due to the limited training data, such a large generator is not required. Therefore, only the initial convolutional layer and the first three residual layers were included in the generator.

The discriminator architecture is unchanged from the implementation of Zhu et al. [7]: a *PatchGAN* classifier is employed, which focuses on overlapping 70x70 patches. The discriminator consists of four convolutional layers with instance normalisation and Leaky ReLU activations with a slope of 0.2, before a final convolution and a sigmoid layer, outputting a map of predictions across the overlapping image patches.

3.1.2 Loss Function Make-Up

Another consideration is the make-up of the loss terms in the objective function of the network. As discussed in Section 2.3, the basic CycleGAN loss function consists of adversarial losses and a cycle-consistency loss term. The equation is repeated for convenience.

$$\begin{aligned}\mathcal{L}_{\text{CycleGAN}}(G, F, D_X, D_Y) =& \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) \\ &+ \lambda \cdot \mathcal{L}_{\text{cyc}}(F, G)\end{aligned} \tag{3.1}$$

Provisional experimentation with CycleGAN using only the aforementioned loss terms indicated poor colour preservation. This is especially apparent in night-to-day translation, where the generated day images have weak colours and poor preservation of the original colours of the input image. To address this, the effects of an identity loss term are investigated.

An identity loss term is a common additional loss term in the training of CycleGAN models. It is calculated by measuring the L1 difference between original day images and their corresponding outputs from the day generator, as well as between original night images and their outputs from the night generator. This term encourages the generators to leave an image from the target domain unchanged if it is passed as an input, which may aid in preserving the colours of features that should remain unchanged when mapping between day and night. For the comparison of the generator architectures, this loss term was given a weight of $\lambda_{\text{i}} = 0.5$.

In addition to the experimentation with alternative generator architectures, 10 separate

models are trained, each using varying identity loss weights during training to explore the effects of this loss term on model performance.

3.2 Experiments and Results

3.2.1 Experimental Setup and Training

The primary dataset for this research is the *Unpaired Day and Night Cityview Images dataset* on kaggle.com [55] which consists of 749 cityscape images: 522 daytime images and 227 night-time images. In addition to this dataset, 93 additional night-time images from the *Aachen Day-Night v1.1 dataset* [56] were included to reduce the imbalanced nature of the dataset. Therefore, in total 522 daytime images and 320 night-time images were used. Of these images, 10 daytime images and 10 night-time images were reserved for validation and the remainder were used for training. Random resized crops of the training images were taken before performing a series of data augmentation steps including flipping, blurring, distorting and altering the colour and brightness. In the case of the generator using a pre-trained ResNet-18 encoder, crops of 224x224 were used to maintain consistency with the 224x224 ImageNet images used to train the ResNet-18 model. For the other two generators, images of 256x256 were used, maintaining consistency with the original CycleGAN training process.

The small size of the training set is a significant limitation that likely hampers the performance of the models. However, understanding how these models perform under such constraints can offer valuable insights for achieving high-quality results when working with larger training sets. Moreover, the limited training data presents an opportunity to rigorously optimize every aspect of model development. In contrast, scenarios with vast amounts of training data may pose less of a challenge, making it easier to overlook subtle optimization opportunities.

The hyperparameters that were used are consistent with the original CycleGAN implementation, with a learning rate of 0.0002 for both the generators and the discriminators. Adam optimisation was used with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. A batch size of 1 was used.

The fundamental CycleGAN training loss terms described in Equation 2.1 were used, with the cycle-consistency loss weight $\lambda = 10$. In addition to these basic loss terms, an identity loss term was calculated as previously discussed, with a weight of $\lambda_i = 0.5$ during the comparison of the generators.

3.2.2 Evaluation Methods

Subjective Visual Assessment

The main method of model evaluation and comparison in this research is direct visual assessment. This visual assessment focuses not only on the desired colour inversions when translating between night and day (the relighting of windows, street lamps, etc) but on a wide variety of properties. The criteria include the absence of visual artefacts, the consistent and realistic colouring of image regions - especially of major image regions such as the sky which should have a consistent colour and an absence of grain - and sharp, high-quality output images. The images that are included in this report are presented at high resolution. Therefore, they are detailed enough to allow for close inspection when viewed in digital format.

The visual assessment is performed using a validation set of 20 images, thus it also provides an indication of the ability of the models to generalise to unseen data.

Quantitative Metrics

As discussed in Section 2.6, a variety of perceptual metrics are available to assess the performance of generative models, though their reliability remains in question. While these metrics can reveal trends and offer insights into model performance, direct visual and subjective assessment remains the most effective method for comparison. Thus, the metrics serve only as supplementary information rather than the primary method of comparison.

In this work, the Fréchet Inception Distance (FID) and Kernel Inception Distance (KID) are used. To calculate these values, sets of real and synthetic images are compared by extracting their deep features using a layer of the Inception V3 model. In this instance, the 2048-dimensional feature space is used.

In the case of FID, the distributions produced by the real and synthetic images are assumed to be Gaussian, and the FID is calculated by using their means and covariances to compute the Fréchet distance between the two distributions. For statistical stability, the image sets are split into batches and a number of FID scores are computed, before finally computing a mean value. An FID of 0 indicates that the feature distributions of the real and synthetic images are identical. Therefore, a lower FID score indicates that the generated images are closer to the real images in terms of their distribution in the Inception V3 model's feature space, and therefore indicates stronger generator performance.

In calculating the KID, the feature distributions are not assumed to be Gaussian. Instead, the Maximum Mean Discrepancy (MMD) is used to compare the Inception model's feature distributions for the real and synthetic images. As explained in Section 2.6, the *kernel trick*

is used with the polynomial kernel outlined in Equation 2.3. As the 2048-dimensional feature space of the Inception V3 model was used, the function is given by:

$$k(\mathbf{x}, \mathbf{y}) = \frac{1}{2048}(\mathbf{x}^T\mathbf{y} + 1)^3 \tag{3.2}$$

Similar to the FID, the KID is typically calculated for a number of subsets before finally calculating an overall mean value. A KID of 0 indicates that the two distributions are identical, thus a lower KID is indicative of a strong generator.

Due to the reliance of FID and KID on statistical properties like the mean and standard deviation, a large image set is required to ensure statistical stability and therefore the reliability of the metrics. This requirement, coupled with the limited data available for training, eliminates the possibility of calculating these metrics on an unseen validation set, thus the values were calculated using the training data instead. While this approach may affect the reliability of the metric values, it is important to note that the training process does not attempt to optimise these metric values directly. Therefore, monitoring their progress during training and their final values upon completion of training can still yield insights into model performance.

Another consideration when using these metrics is the work of Betzalel et al. [43] who found that metrics such as FID and KID are unreliable for comparing models of similar quality. While the metrics may provide insights when contrasting different iterations of the same model under varying parameters, they should be used cautiously when performing cross-model evaluations. Therefore, in this research, especially in the comparison of alternative generator architectures, emphasis is placed on direct visual comparison of the images rather than FID or KID values.

3.2.3 Generator Comparison

To compare the three generator architectures, each model was trained for 100 epochs, keeping all parameters consistent apart from the training image size as previously discussed (224x224 for the ResNet-18 generator, 256x256 for the other generators).

Visual Assessment

A selection of the validation outputs from the day-to-night translation is displayed in Figure 3.3. While the outputs of the three generators appear to be of very similar quality on initial viewing, closer inspection reveals the strengths and limitations of each model.

Checkerboard artefacts can be seen in several of the images produced by the original CycleGAN generator, especially in the outputs for input images 2 and 4. The two U-Net

generators do not produce checkerboard artefacts, due to their use of bilinear upsampling instead of transposed convolutions. However, the basic U-Net occasionally produces flare-like artefacts, which can be seen in its output for input image 4. In contrast, the U-Net with a pre-trained encoder produces fewer artefacts. Therefore, it is likely that the artefacts are arising due to the need for the basic U-Net to learn all the image features from scratch, while the U-Net with a pre-trained encoder has a head start and therefore converges more quickly.

Another point of difference between the basic U-Net and the U-Net with a pre-trained encoder is that the outputs from the basic U-Net are slightly sharper, though this difference is quite subtle. This is most likely a result of the lack of a skip connection at the uppermost level of the U-Net with a ResNet-18 encoder, which leads to a loss of detail at a pixel level. However, the U-Net with a ResNet-18 encoder makes up for its slight reduction in image quality by producing slightly more realistic colour distributions than the other two generators. This is particularly apparent in its outputs for input images 1 and 5 in Figure 3.3.

A subset of validation outputs from the night-to-day translation are displayed in Figure 3.4. The relative difficulty of this mapping is immediately apparent, as the synthetic day images are significantly less realistic than the night images generated in the day-to-night translation. The results from this reverse mapping also emphasise the conclusions drawn from the analysis of the day-to-night translation. While the original CycleGAN generator does not produce any obvious checkerboard artefacts in this translation, the basic U-Net continues to produce flare-like artefacts, which can be detected in small quantities in its outputs for all five validation images. The greater realism of the colour distributions in the outputs from the U-Net with a ResNet-18 encoder is also more readily apparent in the night-to-day translation. The outputs have more consistent and appropriate colours in image regions such as the sky, roads and buildings, reinforcing the belief that the use of the ResNet-18 encoder improves the semantic understanding of the network.

The visual artefacts produced by the original CycleGAN generator and basic U-Net generator are highlighted in Figure 3.5. While the overall performance of the three models is quite similar, the general absence of these severe artefacts in the outputs from the generator with a pre-trained encoder is a significant advantage.

Quantitative Metrics

To supplement the visual comparison, the FID and KID values of each model for both day-to-night and night-to-day translation were evaluated after every 5 epochs of training to observe their evolution. The plots of these values can be seen in Figure 3.6.

One immediate observation from the plots is that the scores for the generation of day images are consistently higher than the night image scores, again reinforcing the increased

difficulty of mapping from night to day. Therefore, it is noteworthy that the basic U-Net has a much smaller gap between its day and night scores, while the other two generators have a significant disparity between their day scores and night scores. The basic U-Net appears to perform the strongest in both the FID and KID, though the original CycleGAN generator eventually converges to slightly lower values after 100 epochs. The U-Net with a ResNet-18 encoder has the highest average values in both metrics during training, suggesting the weakest performance of the three models. However, as previously discussed these metrics are flawed for cross-model comparison, and the visual assessment indicated that using a ResNet-18 encoder led to improved performance.

A final point of interest is the volatility of the FID and KID scores during training. The two U-Nets appear to be significantly less volatile on these metrics in comparison to the basic CycleGAN generator, which has significant peaks and troughs as it converges towards its minimum FID and FID values.

3.2.4 Identity Loss Investigation

Following the comparison of the generators, the identity loss term was investigated by training 10 separate networks (using the ResNet-18 encoded generator) again for 100 epochs, with the identity loss weight varying from $\lambda_i = 0$ to $\lambda_i = 0.9$.

Visual Assessment

A selection of validation outputs from the experimentation with the identity loss term can be seen in Figure 3.7. The expectation of superior colour preservation with an increased identity loss weight appears to be validated by the results, as the colours appear to be slightly stronger and more reflective of the colours of the input image as λ_i is increased. This is particularly evident in the outputs for input image 2 in Figure 3.7. Increasing λ_i leads to a stronger red colour on the wall which is illuminated by red light in the night-time image. While the preservation of the red light is not preferable, as this lighting would not be present in the daytime, the increased emphasis on preserving the colours of the input image also leads to the correct colouring of the trees in front of the building, which are incorrectly assigned a red colour when λ_i is too low. This poses an obvious dilemma, as the increased identity loss weight has the effect of preserving the colours of the input image, but this has both positive and negative impacts.

As λ_i is increased, it eventually reduces the emphasis on the cycle-consistency loss term to the point that the model struggles to converge properly, with an increase in visual artefacts at higher λ_i values, before failing quite dramatically as λ_i is increased further, which is illustrated in the outputs for $\lambda_i = 0.9$. In this case, none of the required colour inversions are performed properly, and even the basic recolouring of major image regions such as the sky

are inconsistent.

Overall, while the identity loss term does appear to affect colour preservation, this effect seems to be quite small, as the outputs for input images 1, 3 and 4 appear largely unaffected by the loss term until its weight is increased to $\lambda_i = 0.9$.

Quantitative Metrics

The FID and KID values of each of the 10 generators with varied identity loss weights after training for 100 epochs are displayed in Figure 3.8.

The scores for the generation of day images do not follow any clear pattern, however, the night images produce curves that are consistent with the observations made in the visual assessment. The introduction of an identity loss term does appear to make a slight improvement in model performance, hence the initial decrease in the FID and KID values for the generation of night images as λ_i is increased, reaching their lowest, optimum values when $\lambda_i = 0.4$. However, as λ_i is increased further, the values increase again, indicating a reduction in the perceived quality of the output images. This is consistent with the increased prevalence of visual artefacts in the outputs from the models trained with higher λ_i values.

Interestingly, when the model fails dramatically at $\lambda_i = 0.9$, this does not cause a sharp increase in the FID and KID values for both the day and night generators as would be expected. Instead, a sharp divergence can be seen between the curves. This could be indicative of the unreliability of these metrics, but it is also possible that it simply highlights the disparity in the difficulty of the two mappings: a poor day-to-night generator can still produce acceptable results on these metrics, while a poor night-to-day generator is much more easily detected. This is supported by the outputs in Figure 3.7: the failure of the model in night-to-day translation when $\lambda_i = 0.9$ is immediately apparent, whereas the failure to properly translate from day-to-night requires closer inspection. This could explain why the FID and KID scores increase dramatically when the night-to-day translation (the more difficult task) fails, while they actually decrease when the day-to-night translation is not performed properly. Another possible conclusion from this unexpected behaviour is that it supports the suspicion that stronger, more consistent generators will perform equally well in terms of their FID and KID values for the generation of day and night images, while poor generators may display more volatile behaviour on these metrics, with one translation task potentially outperforming the other in terms of the FID and KID scores.

3.3 Conclusions

This chapter investigated the effectiveness of different generator architectures on CycleGAN performance in day-to-night image translation. This exploration focused specifically on comparing the original CycleGAN generator architecture, which consists primarily of residual blocks, to a basic U-Net architecture and a U-Net with a pre-trained ResNet-18 as the encoder. Both U-Net generators outperformed the original generator, with the basic U-Net producing the sharpest images but the U-Net with a pre-trained encoder displaying quicker convergence and fewer visual artefacts. Therefore, a U-Net with a pre-trained ResNet-18 encoder appears to be the strongest of the three options.

In addition to the investigation of model architectures, the inclusion of an identity loss term was also explored. This loss term appears to provide a slight improvement in colour preservation and therefore in overall model performance. However, the inclusion of this additional loss term also serves to reduce the emphasis on the cycle-consistency loss term, and therefore leads to poor convergence if the identity loss weight is too large. An identity loss weight of $\lambda_{\text{i}} = 0.4$ appears to be optimal in this case.

Overall, an optimal CycleGAN architecture has been identified, which uses a U-Net generator with a pre-trained ResNet-18 encoder. It is trained with an additional identity loss term, assigned a weight of $\lambda_{\text{i}} = 0.4$. This model serves as the baseline for comparison against in the experiments in Chapter 4.

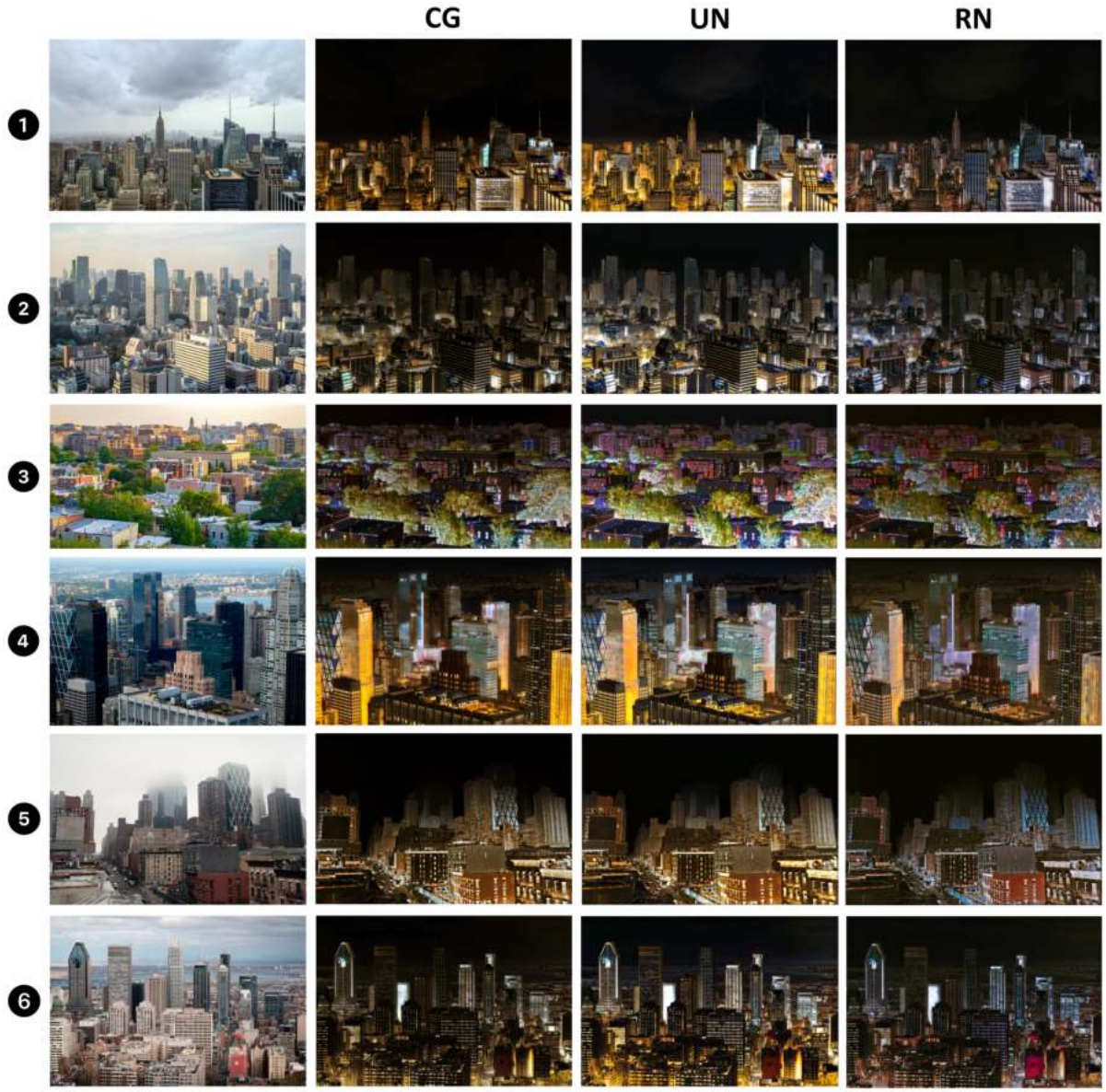

Figure 3.3: Day-to-night translation with three alternative generator architectures. The images in the left column are the original day images. From left to right, the other columns display the synthetic night images generated by the original CycleGAN generator (CG), a basic U-Net generator (UN), and a U-Net with a pre-trained ResNet-18 encoder (RN).

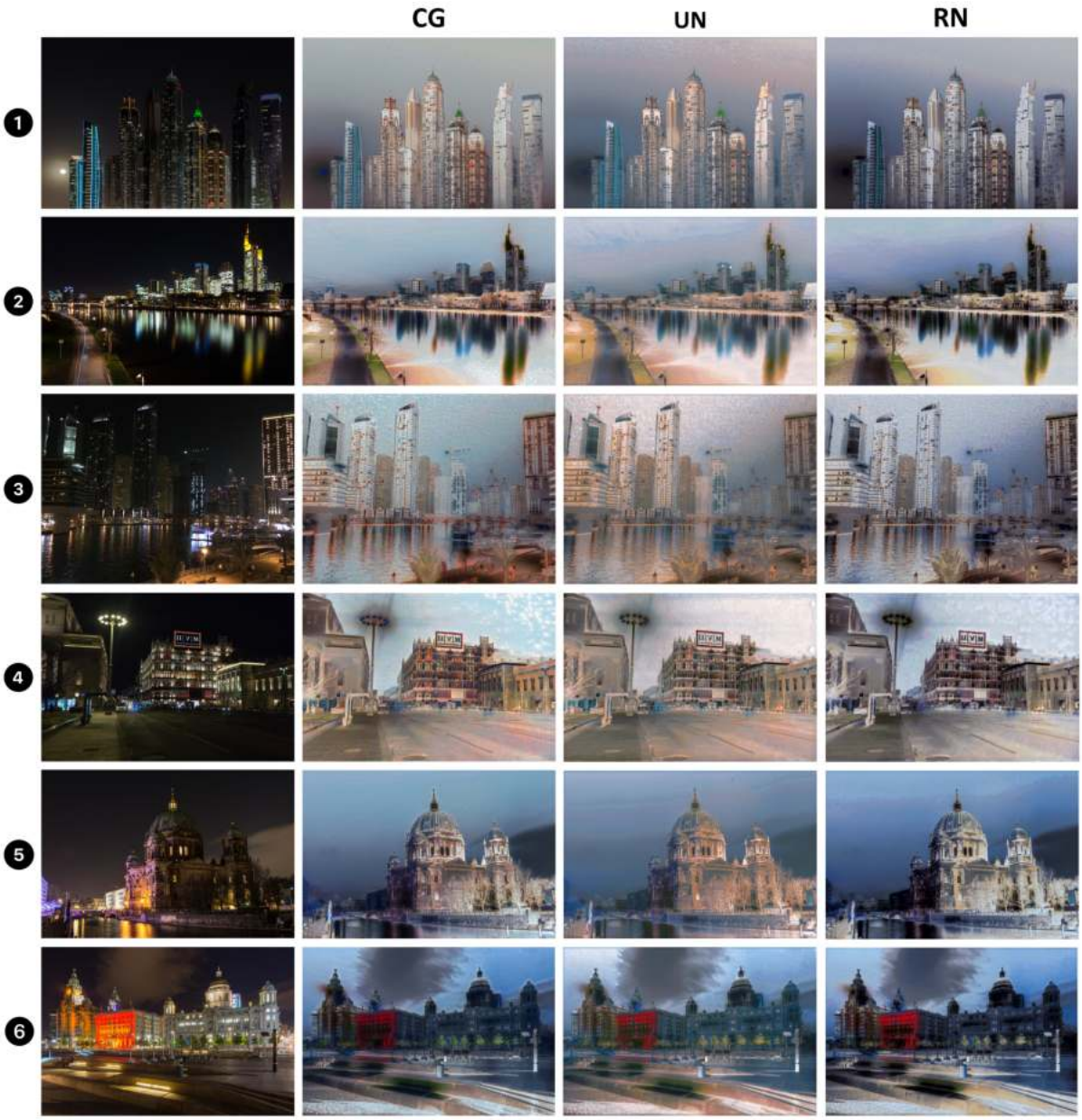

Figure 3.4: Night-to-day translation with three alternative generator architectures. The images in the left column are the original night images. From left to right, the other columns display the synthetic day images generated by the original CycleGAN architecture (CG), a basic U-Net generator (UN), and a U-Net with a pre-trained ResNet-18 encoder (RN).

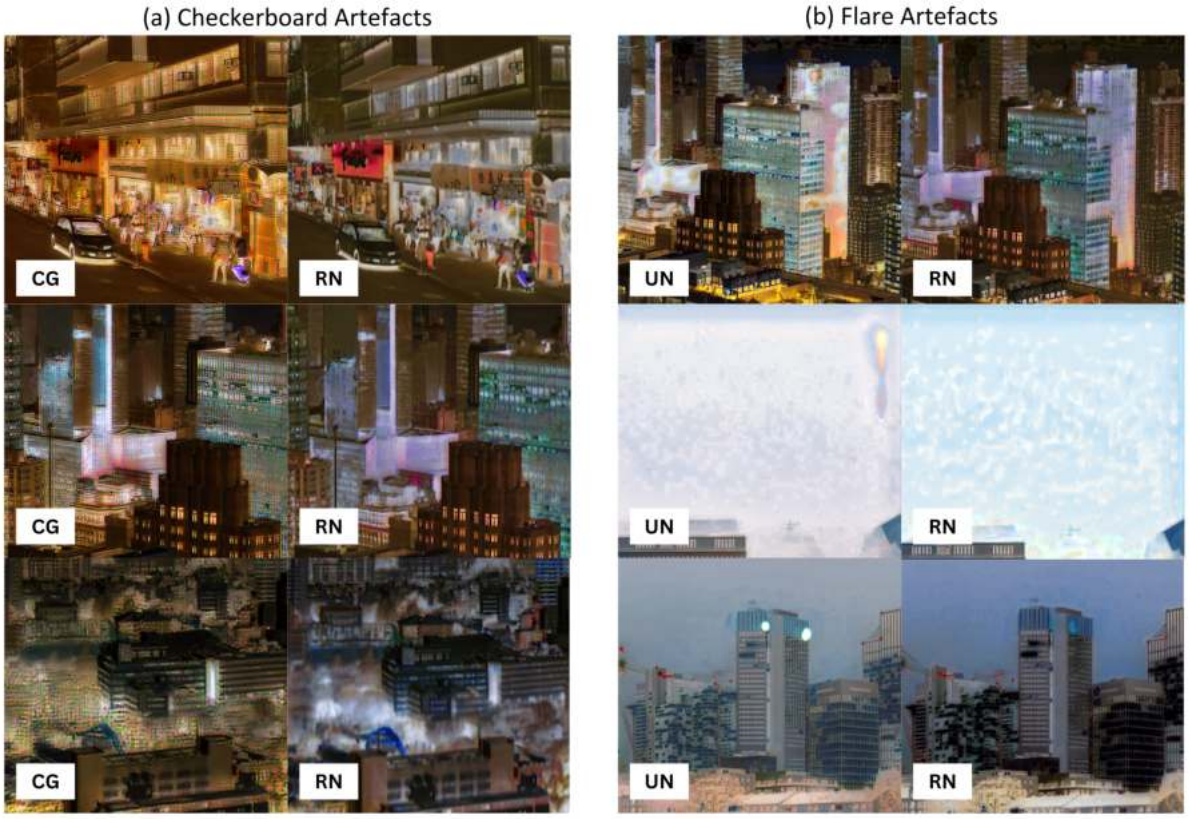

Figure 3.5: Comparison of the visual artefacts generated by the three generators. (a) The checkerboard artefacts that are produced by the original CycleGAN generator (CG), are not produced by the U-Net with a pre-trained ResNet-18 encoder (RN). (b) The spotty and flare-like artefacts that are occasionally produced by the basic U-Net generator (UN) are contrasted against the same image patches in the outputs from the U-Net with a pre-trained ResNet-18 encoder (RN), which does not produce these artefacts.

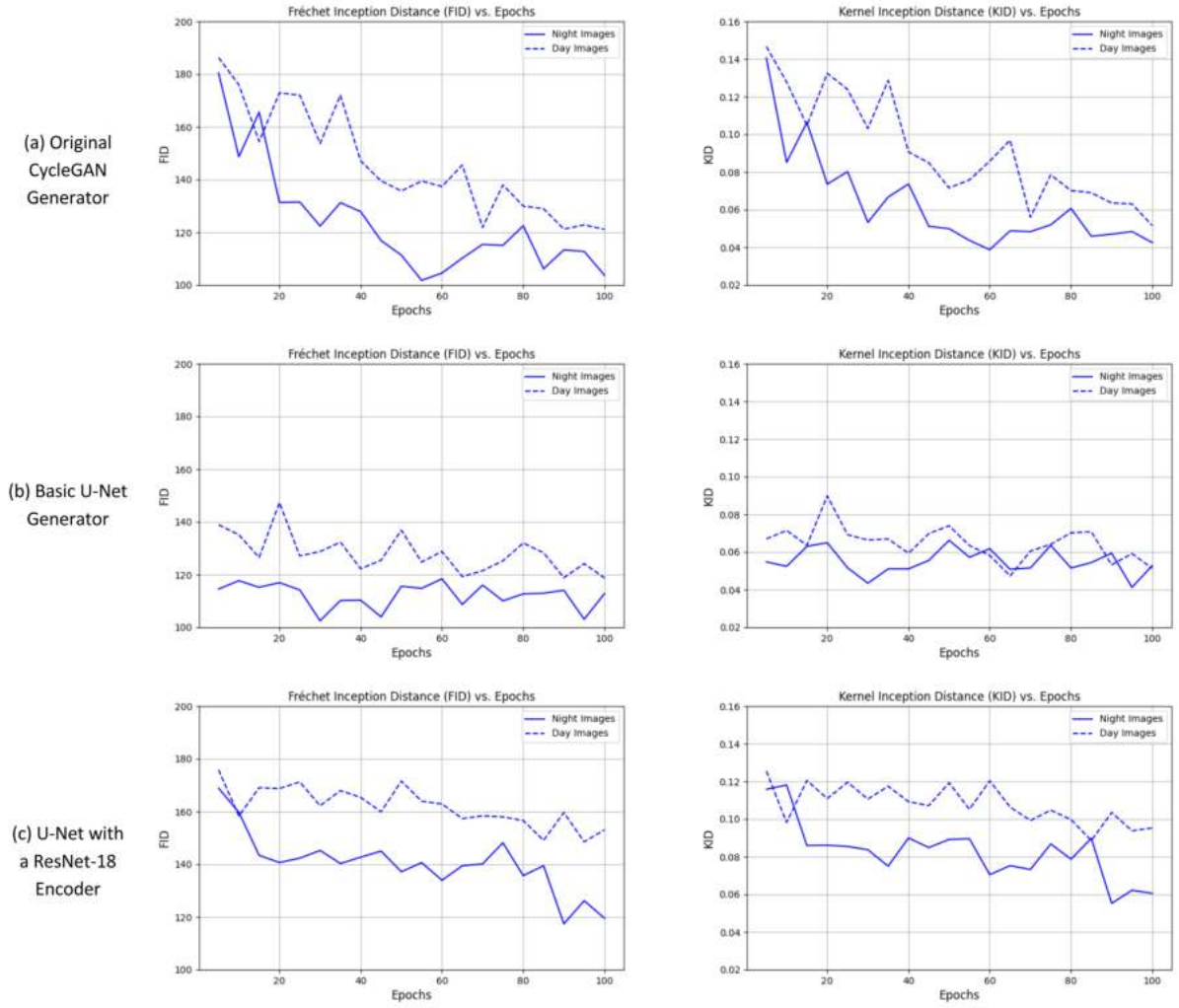

Figure 3.6: Plots of the FID scores (left) and KID scores (right) of the generators over 100 epochs of training. The scores are plotted for day-to-night translation using a full line and night-to-day translation with a dashed line. The plots for the basic CycleGAN generator are shown at the top, with the plots for the basic U-Net in the middle and the plots for the U-Net with a pre-trained ResNet-18 at the bottom.

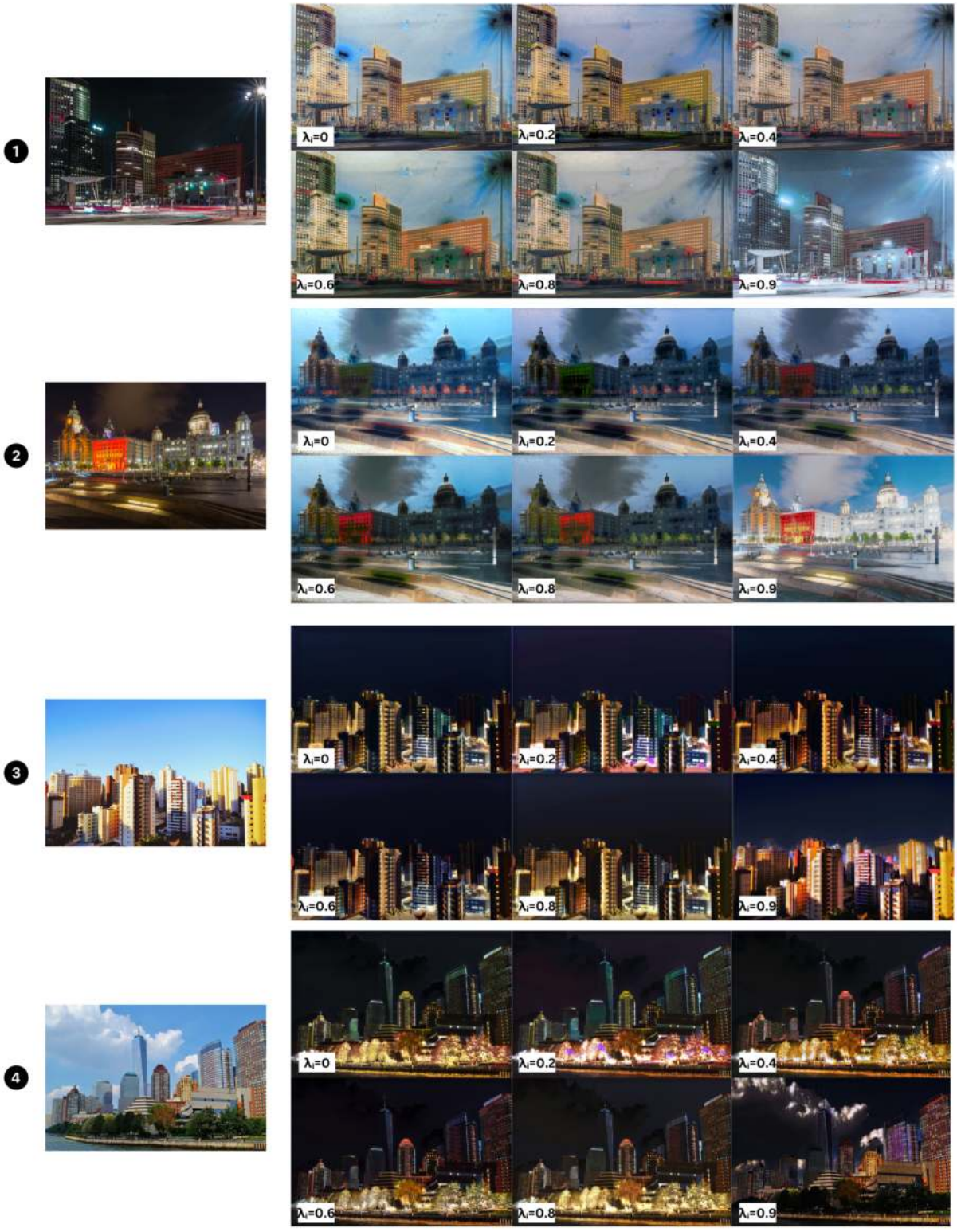

Figure 3.7: The effect of the identity loss term is illustrated by comparing the images that are generated by separate models trained with different values of the identity loss weight, λ_i. The images on the right are the original input images, and the synthetic images are displayed to the right. The outputs are shown from models trained with the identity weight λ_i varying from 0 to 0.9.

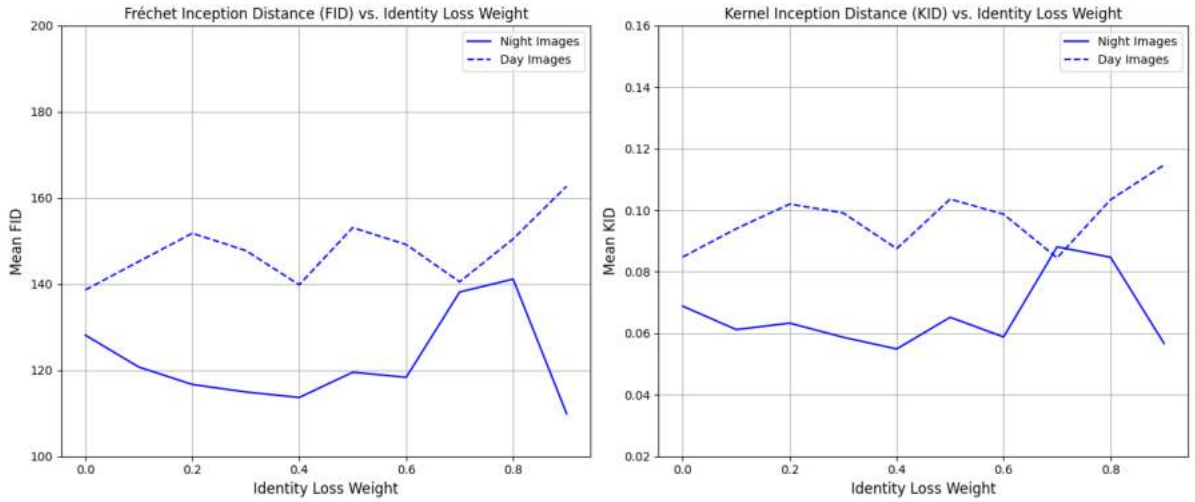

Figure 3.8: Plots of the FID scores (left) and KID scores (right) of separate networks trained with different values of the identity loss weight, λ_i. A full line is used for the scores for day-to-night translation, and a dashed line is used for the scores for night-to-day translation. The scores for each network are calculated after 100 epochs of training. The metric values are displayed on the y-axis, and the identity loss weight used in training is represented on the x-axis.

4 Sharing Generators

Having established an optimal CycleGAN model for day-to-night image translation, the next objective is to explore the effects of the latent representations on model performance. The CycleGAN U-Net generator can be modelled as encoding an input image into a latent representation which can be decoded to the target domain by the decoder. As discussed in Section 2.4, the construction of this latent representation is an important consideration.

The two encoder-decoder pairs in a basic CycleGAN architecture can be thought of as translating via separate latent spaces. There is no sharing between the two generators, nor is there a constraint on the network to encourage it to produce a common latent representation of a given image's content across the two translations. The separation between the two latent spaces in CycleGAN means that the network may be trained to effectively perform a translation task without producing meaningful, interpretable representations of the input images at the bottleneck between the encoder and decoder. In this context, a meaningful latent representation abstracts the input image into a latent code that captures some of its intrinsic properties in a manner that is reusable for tasks beyond the original training objective. An interpretable latent space is valuable in its own right: it is vital to have an understanding of the inner workings of the network rather than viewing it as an inscrutable *black box*. Additionally, a constrained latent representation facilitates a variety of augmentations, such as mapping to a greater number of target domains [25] [11] or disentangling image content and style for style transfer [10].

In this chapter, the goal is to produce a meaningful latent representation by encouraging the network to retain only the domain-invariant content information, disentangling it from the domain-specific style information, which should be discarded. The motivation for this adaptation is to investigate the effect of constrained latent mappings on output quality, as well as to produce a network that is compatible with the modelling of daytime and night-time images as existing within a continuous stylistic domain. Meaningful latent representations are commonly used to increase the number of target domains. A generator that seeks to map to intermediate points between the original two domains will benefit from the same structure: in a sense, the model of a continuous domain can be thought of as an

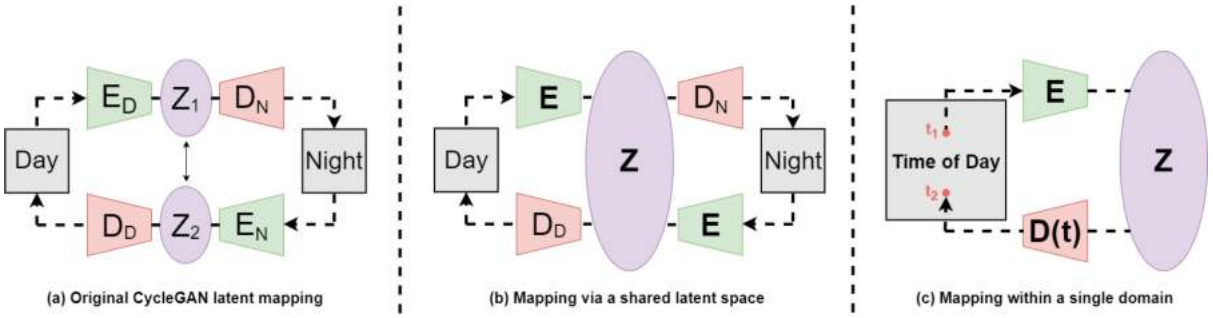

Figure 4.1: The mapping of latent variables can have a significant impact on the translation task. (a) CycleGAN keeps the forward and reverse mappings entirely separate, and therefore it can be thought of as mapping input images from the two domains to separate latent spaces. (b) By using a single shared encoder, the network is encouraged to map images to a shared latent space. Through this adaptation, the expectation is that the encoder will learn to retain only the domain-invariant information when encoding images to the latent space Z while discarding the domain-specific style information. The decoders then learn to fill the domain-agnostic latent representations with the domain-specific style. (c) A domain-agnostic latent space provides the possibility of training a single generator that uses a timestamp input to determine whether it maps input images to either daytime or night-time.

infinite number of target domains.

To this end, the first step in this chapter is to use a shared encoder across the two generators, coupled with a *mid-cycle consistency* loss term to train a network that produces a common latent representation of daytime and night-time images. The second task is to implement a single generator based on the shared encoder scheme, where the domain-specific decoders are replaced with a single, shared decoder that takes a timestamp input to determine the target lighting conditions. A timestamp of 0 indicates that daytime is the target, while a timestamp of 1 indicates that night-time is the target. The architectural adaptations are contrasted against the original CycleGAN architecture in Figure 4.1. The timestamped generator architecture should be capable of interpolating between daytime and night-time when it is given an intermediate timestamp value (between 0 and 1). This capability is explored in Chapter 5 by incorporating time-lapse data into the training process.

The timestamped generator is similar to the single generator used in StarGAN [11]. However, while the single generator of StarGAN was implemented to map to multiple domains, our single generator is implemented with the end goal of interpolating between two extremes (day and night) to generate images with intermediate lighting conditions. Furthermore, while the single generator of StarGAN is coupled with a single discriminator that outputs a predicted domain in addition to its prediction of whether the image is real or synthetic, in this work the two CycleGAN discriminators are left unchanged.

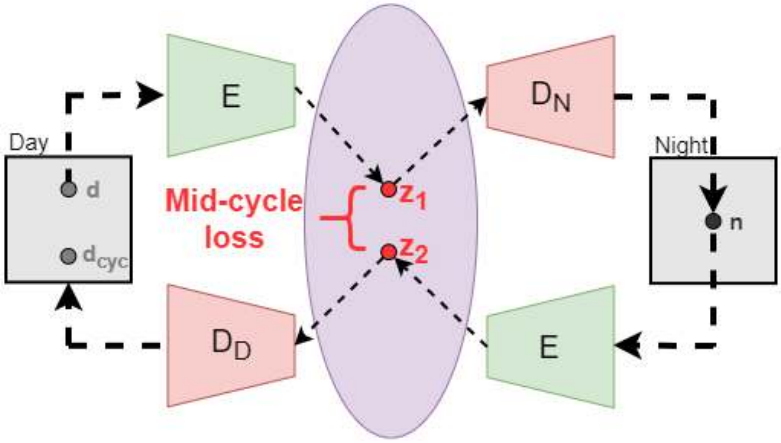

Figure 4.2: The mid-cycle consistency loss term is calculated as the L1 difference between the two latent representations of an input image during a full cycle through the network. If the shared encoder only retains image content, the two latent codes z_1 and z_2 should contain the same information, as they relate to the same image content, simply encoded from different stylistic representations. By encouraging the network to keep these representations consistent, it should learn to retain only domain-invariant information.

4.1 Proposed Improvements

4.1.1 A Shared Encoder

The first improvement that is proposed is the use of a single, shared ResNet-18 encoder. This encoder is used to map both daytime and night-time images to a single, shared latent space, from which they can be decoded into either daytime or night-time images. By sharing the latent space, the network should learn to retain the domain-invariant image content while discarding the domain-specific style information.

To further constrain the network and encourage it to map to a shared latent space, a *mid-cycle-consistency* loss term is introduced. As illustrated in Figure 4.2, the mid-cycle consistency loss is computed as the L1 difference between the two latent representations of a training input as it completes a full cycle through the network. The two latent codes that are outputted from the encoder are the result of inputting two images with the same underlying content but different stylistic information. Therefore, if the encoder is performing the desired disentanglement and retaining only the domain-invariant information, the two latent representations should be the same.

The use of a mid-cycle consistency loss term may aid in producing more consistent cycles during training and therefore more effective performance of the image-to-image translation task. Furthermore, a shared encoder and the resulting content-style disentanglement results in a more flexible network that can be adapted for a wider range of translation tasks.

4.1.2 A Single Generator

Instead of training separate decoders to map from the shared latent space to either night or day, a single decoder is proposed that takes a timestamp input in addition to the latent representation from the encoder. A timestamp value of 0 indicates that the target style is full daytime lighting conditions, and a timestamp of 1 indicates a target of night-time lighting conditions. This timestamp parameter is passed to the decoder as a scalar value. It is then transformed into a high-dimensional representation across a pre-defined number of embedding channels, which is expanded to match the spatial dimensions of the output from the encoder. The timestamp information is then concatenated with the encoded representation of the input image. The decoder then decodes the concatenated representation to the target domain.

Therefore, the network is reduced to a single GAN, with one generator that can translate to either night or day. However, the network still employs two discriminators - a night discriminator and a day discriminator - to provide adversarial losses in the training of this generator. The training process for this single generator is unchanged from the previous models: a cycle-consistency loss term is still required to constrain the mappings.

4.2 Experiments and Results

4.2.1 Experimental Setup and Training

The training process is unchanged from Chapter 3, with the same training data, preprocessing steps and hyperparameter values. In the direct comparison of basic CycleGAN to a network with a shared encoder and a network with a single generator, an identity loss term is included with a weight of $\lambda_{\text{i}} = 0.4$, as this was previously identified as the optimal weight. In the investigation of the mid-cycle consistency loss term, λ_{i} is set to zero.

4.2.2 Evaluation Methods

The evaluation methods are unchanged from Chapter 3. Once again, direct visual assessment is the primary method of evaluation, with the FID and KID providing supplementary numerical information.

4.2.3 Generator Comparison

To compare the performance of the proposed architectures against CycleGAN, the optimal CycleGAN identified in Chapter 3 was used. This CycleGAN uses a U-Net generator with a pre-trained ResNet-18 encoder.

Visual Assessment

A subset of the validation results from the day-to-night translation is displayed in Figure 4.3. The colour distributions and overall quality of the outputs from the baseline CycleGAN model and the network with a shared encoder are very similar for this translation. The network with a single generator provides noticeably different results, with a wider variation in quality. While the outputs from the timestamped generator are still largely acceptable, some failure cases can be observed, such as its output for image 5, which has unrealistic colouring and a significant amount of visual artefacts. Similarly, the colours of the night image it generates for day image 1 are unrealistic and noticeably worse than the outputs from the other two models.

Similarly, a sample of results for night-to-day translation is displayed in Figure 4.4. The generator with a shared encoder appears to outperform the other two techniques in terms of colour preservation, possibly due to more meaningful latent representations from the shared encoder enabling the two decoders to focus on producing stronger, more realistic colour distributions. However, it is important to note that the stochastic nature of GAN training may be contributing to these differences. Furthermore, the stronger colours from the encoder-sharing model are accompanied by some significant errors in colour assignments, such as the green colouring of the street and the street lamp in its output for image 4.

The basic CycleGAN model outperforms the timestamped generator in night-to-day translation, though the timestamped generator displays improved performance in night-to-day translation compared to its performance in day-to-night translation.

While the encoder-sharing network appears the strongest model in night-to-day translation due to its superior colour distributions, the network produces more artefacts than the baseline model, as illustrated in Figure 4.5. Specifically, the sharing network produces a spotty artefact in several of its outputs. This likely stems from the fact that the constraint on the network to produce meaningful latent representations makes the learning process during training more complex. Extending the training process to a greater number of epochs may enable the network to refine the translation and reduce the occurrence of these artefacts.

Overall, the encoder-sharing model appears to perform at least as well as the baseline model, if not slightly better due to its improved colour preservation in the night-to-day translation. However, the increased prevalence of visual artefacts from this model detracts from the quality of its outputs. The use of a single, timestamped generator appears to reduce the quality of the translation, though this reduction in quality is not significant in most cases.

Quantitative Metrics

Similar to the comparison of generator architectures in Chapter 3, the FID and KID values for the two translation tasks are calculated for each model throughout the training process. The results are displayed in Figure 4.6. An immediate observation in these plots is the proximity of the two curves for the two sharing models compared to the significant gap between the day image and night image values of the baseline model. This suggests that sharing the encoder across the two generators leads to more consistent performance of the two translation tasks. The timestamped generator also displays a more volatile convergence than the other two models.

The proximity of the two curves for the sharing models indicates that a shared encoder results in a more meaningful latent representation of the input images, translating to more consistent performance across different translation tasks. However, as previously discussed, these metrics have significant flaws and are not entirely reliable, which was emphasised by the weak performance of the ResNet-18 encoder on these metrics relative to the other models in Chapter 3, despite the visual assessment indicating that it was the strongest model. Therefore, the FID and KID values in this chapter must be considered judiciously.

4.2.4 Mid-Cycle Loss Investigation

The mid-cycle consistency loss term is used to encourage the generation of domain-agnostic latent representations of image content. However, the necessity of this additional loss term remains in question: it is possible that by using a shared encoder the network is implicitly encouraged to converge to a shared latent space, rendering the mid-cycle loss term redundant. Furthermore, the actual influence of a shared latent representation on the output quality is uncertain, with the function of this constraint potentially limited to simplifying the process of integrating additional target domains into the mapping process. With these considerations in mind, a brief investigation into the effects of the mid-cycle loss term is conducted by training four separate encoder-sharing networks with the mid-cycle loss weight λ_m varying from $\lambda_m = 0$ to $\lambda_m = 3$.

Visual Assessment

A subset of the results from the experimentation with the mid-cycle loss weight is shown in Figure 4.7. There is no significant change in the characteristics of the output images as λ_m is increased, though eventually at $\lambda_m = 3$ the emphasis on the cycle-consistency loss term is reduced to the point that the network fails to converge properly, similar to the failure when too much emphasis is placed on the identity loss weight λ_i in Section 3.2.4. However, it is noteworthy that the network tolerates a higher value of λ_m before failing, compared to the

identity loss investigation when the failure occurs at $\lambda_i = 0.9$. This observation highlights the mid-cycle loss's constraining effect, which is based on a similar principle to the cycle-consistency loss.

Quantitative Metrics

The FID and KID values of the four models with varied mid-cycle loss weights were also computed after 100 training epochs, and are plotted in Figure 4.8. The conclusion that can be drawn from these plots is consistent with that of the visual assessment: the mid-cycle loss has no significant effect on output quality until it is increased beyond a threshold value that reduces the emphasis on the cycle-consistency loss term to the point that the model fails to converge properly. This is reflected by the increase in the FID and KID scores for both translation tasks when $\lambda_m = 3$, indicating worsened performance.

4.3 Conclusions

This chapter has explored several adaptations to the CycleGAN architecture aimed at improving both the quality of the outputs as well as the versatility of the network, particularly to facilitate compatibility with time-lapse training, which is explored in Chapter 5. The introduction of a shared encoder displays particular promise, especially in enhancing the colour distributions of output images. This suggests that a shared latent space may lead to improved image-to-image translation. However, the improvements observed in this chapter are subtle and must be interpreted cautiously, considering the inherent stochastic variations in GAN performance.

The exploration of the mid-cycle consistency loss term yielded inconclusive results. Although the term appears to enforce a level of constraint on the mapping that is similar to the function of the cycle-consistency loss term, no added benefit is seen in the output images when this term is included. It is possible that the term leads to a tighter shared latent space, which may improve performance when translating to a greater number of domains, but the constraints of this research did not permit further investigation of this possibility. Another possible explanation for the lack of impact of this loss term on output quality is that it is simply redundant and that a shared encoder is sufficient to encourage the formation of a domain-agnostic latent space. Regardless, there is little justification for the inclusion of this term in the experiments with time-lapse training in Chapter 5.

The implementation of a single, timestamped generator simplifies the network architecture but also appears to reduce the quality of the translated images. However, the primary objective in the introduction of this architecture was to introduce a generator that is compatible with time-lapse training. This has been accomplished without degrading the translation quality too much. Therefore, a generator architecture has been established that

opens the compelling possibility of exploiting the opportunities presented by time-lapse data. This is explored further in Chapter 5.

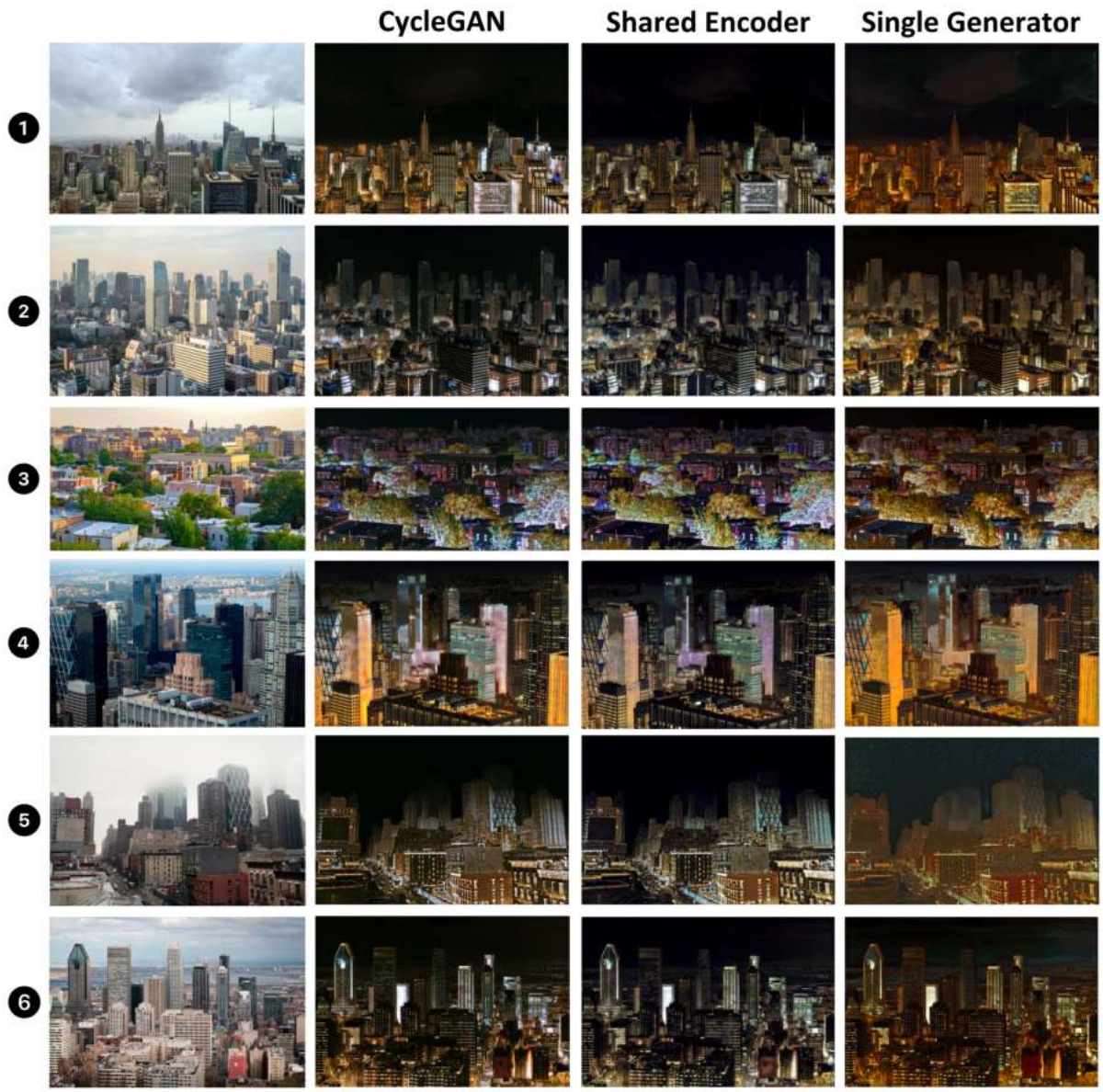

Figure 4.3: Day-to-night translation with three alternative network architectures. The images displayed in the left column are the original day images. From left to right, the other columns display the synthetic night images generated by a baseline CycleGAN architecture using pre-trained ResNet-18 encoders, a network with a single, shared encoder, and a network with a single generator that takes a timestamp input in addition to the input image to determine the target lighting conditions.

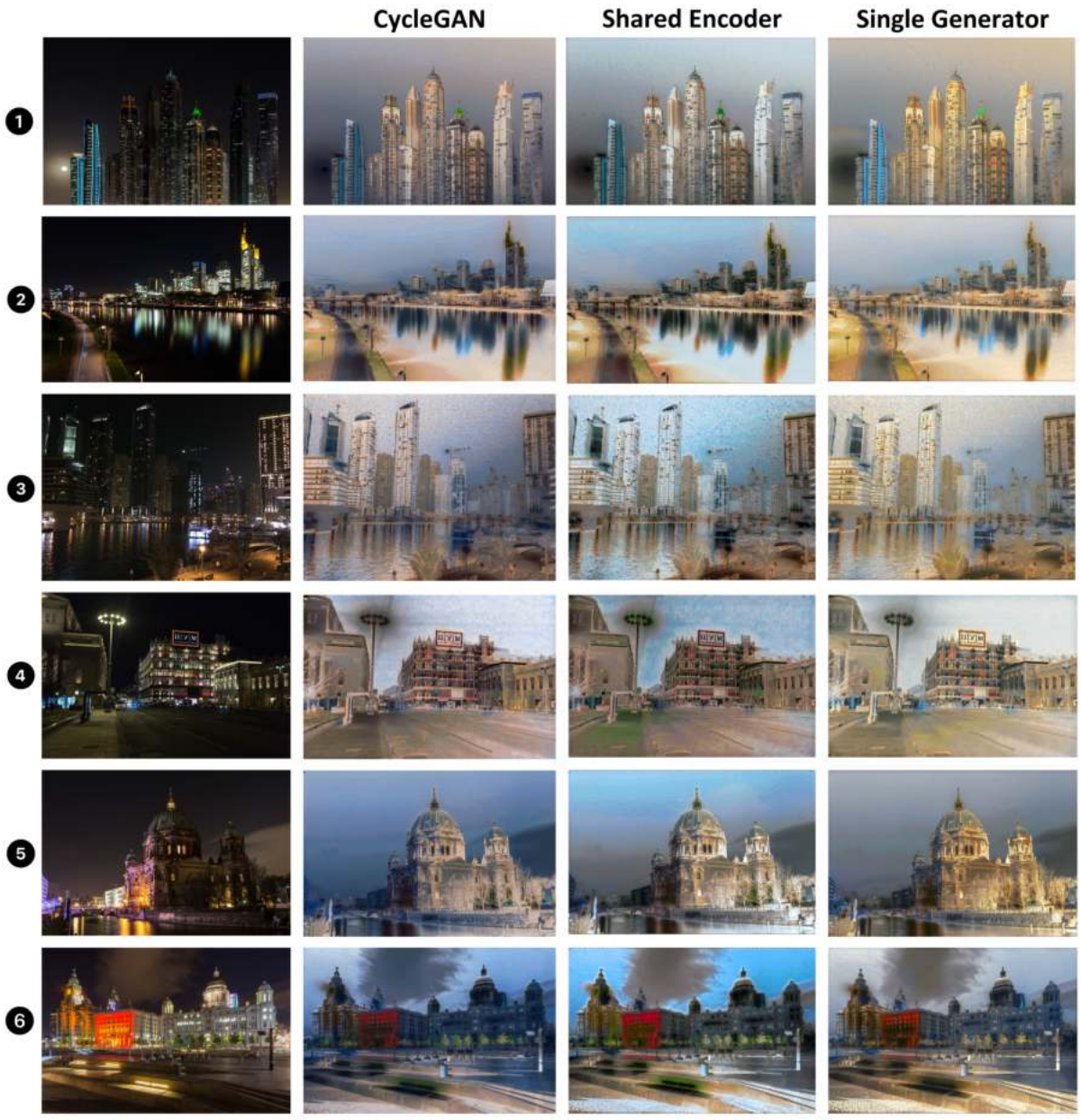

Figure 4.4: Night-to-day translation with three alternative network architectures. The images displayed in the left column are the original night images. From left to right, the other columns display the synthetic day images generated by a baseline CycleGAN with pre-trained ResNet-18 encoders, a network with a single, shared encoder and a network with a single generator that takes a timestamp input in addition to the input image to determine the target lighting conditions.

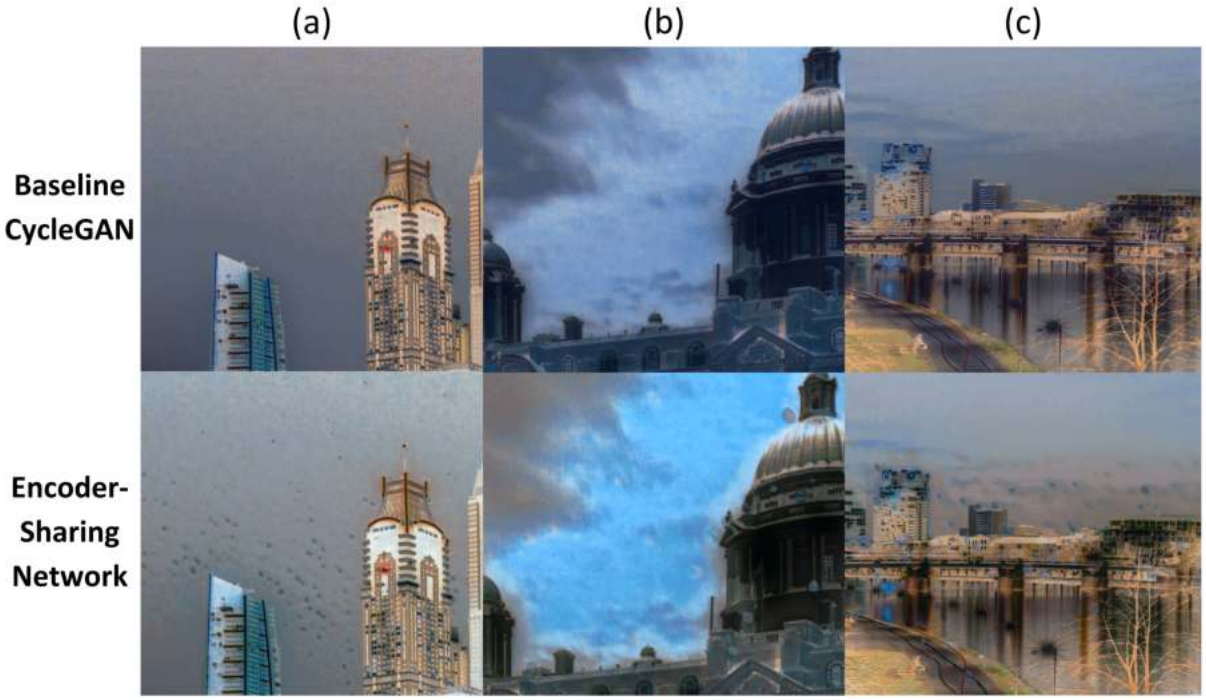

Figure 4.5: The superior colour distributions of the outputs from the encoder-sharing network come with an increase in visual artefacts. Examples of these artefacts can be seen in the bottom row. They are compared against the same image patches in the outputs from the baseline CycleGAN model in the top row. (a) The grey spotty artefacts produced by the encoder-sharing model are not seen in the outputs from the baseline model. (b) The encoder-sharing network produces a halo-like effect around buildings and other image features. The spotty artefact can also be seen in this example. (c) The spotty artefact often appears near the horizon and at the edges of the image.

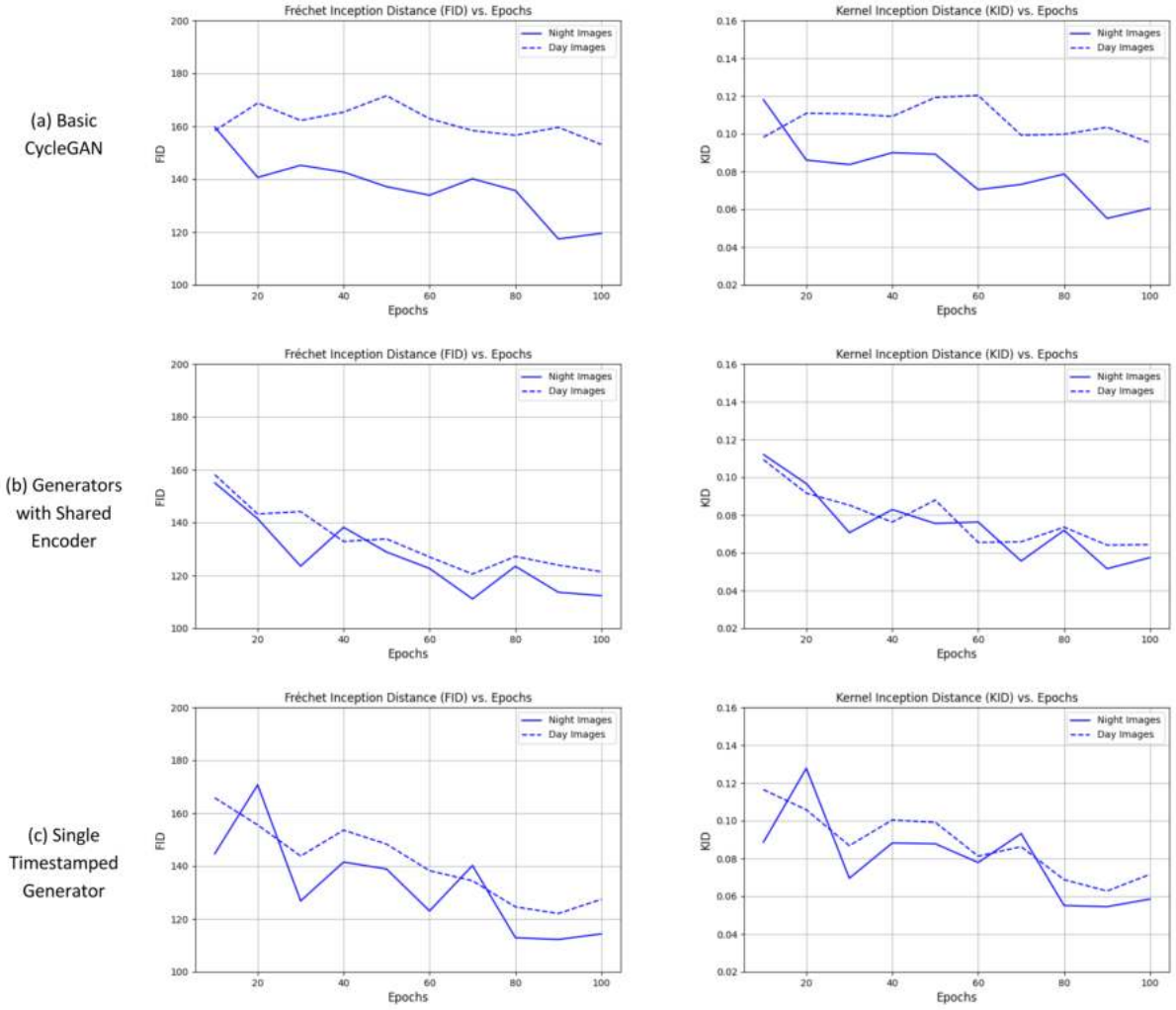

Figure 4.6: Plots of the FID scores (left) and KID scores (right) of the three alternative architectures over 100 epochs of training. The scores are plotted for day-to-night translation with a full line and night-to-day translation with a dashed line. The plots for the baseline CycleGAN model are shown in the top row, with the plots for the encoder-sharing network in the middle row and the plots for the timestamped generator in the bottom row.

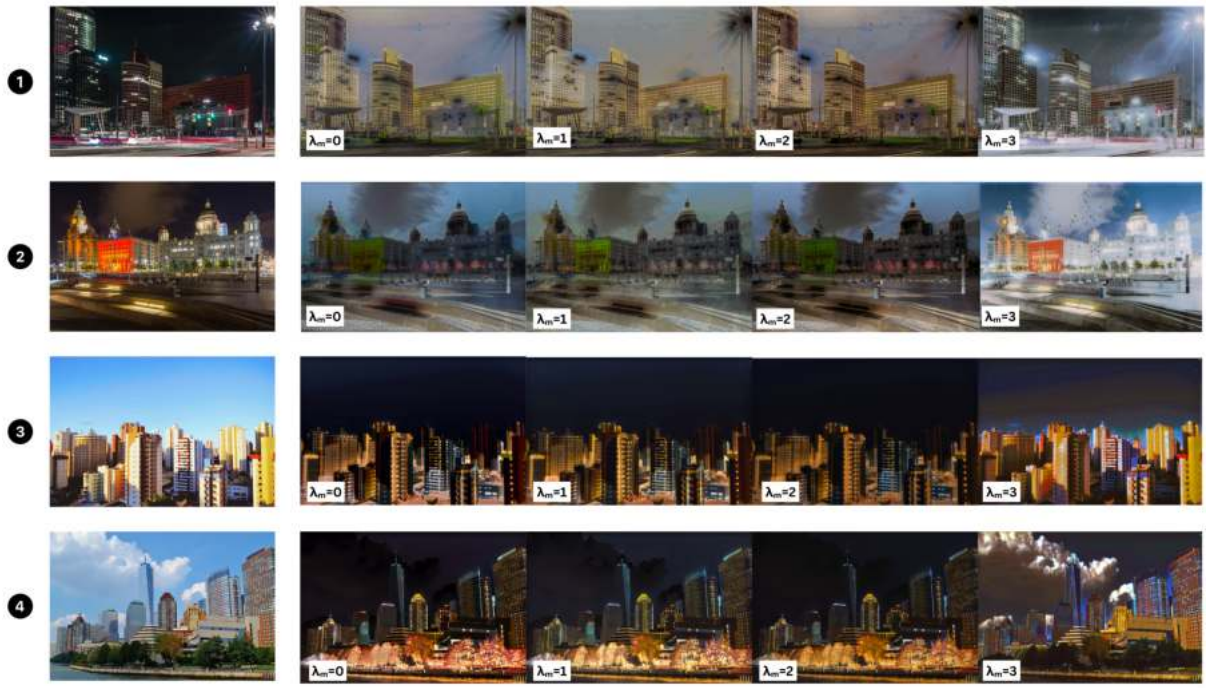

Figure 4.7: The effect of the mid-cycle loss term is illustrated by comparing the images generated by separate models trained with different values of the mid-cycle loss weight, λ_m. The images on the right are the original input images, and the synthetic images are displayed to the right. The outputs are shown from models trained with the identity weight λ_m varying from 0 to 3.

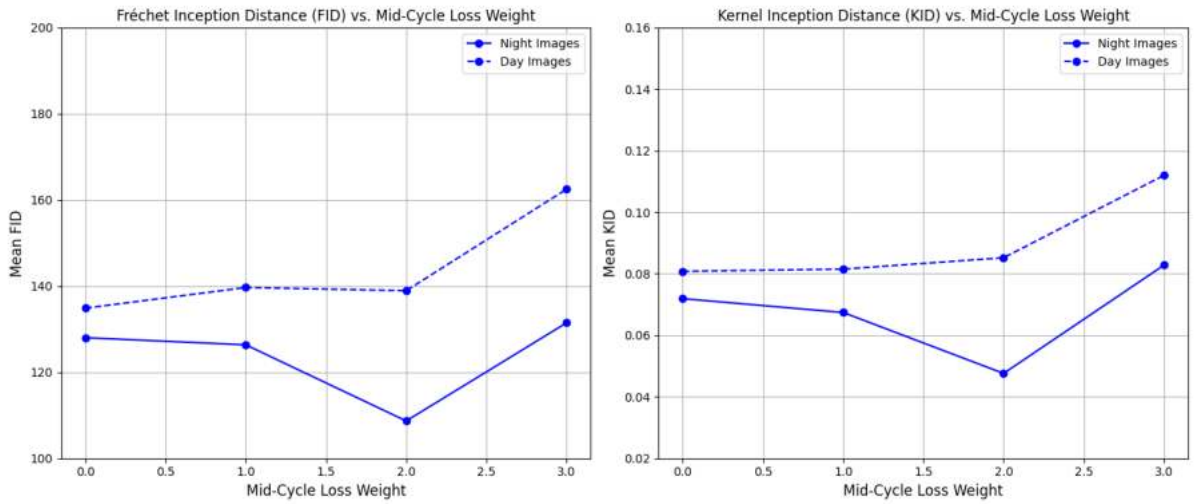

Figure 4.8: Plots of the FID scores (left) and KID scores (right) of separate networks trained with different values of the mid-cycle loss weight, λ_m. A full line is used for the scores for day-to-night translation, and a dashed line is used for the scores for night-to-day translation. The scores for each network are calculated after 100 epochs of training. The metric values are displayed on the y-axis, and the mid-cycle loss weight used in training is represented on the x-axis.

5 Synthetic Time-Lapses

The timestamped generator developed in Chapter 4 can be enhanced through the use of time-lapse data. Time-lapses are a rich source of information about the subtle changes that occur as a scene transitions from day to night, and this should be exploited in the development of a day-to-night translation system. The overall objective in this chapter is therefore to produce a model that is capable of producing smooth, realistic synthetic time-lapses.

No further architectural changes are introduced in this phase of the research; the only change is in the training process of the timestamped generator. The training process is structured in two phases: an initial phase that is unchanged from Chapter 4, training for 100 epochs with the same hyperparameters and training data that were previously used, followed by a secondary phase of training that uses time-lapse data. This approach is designed to first establish a model that is capable of translating between day and night, before refining the model's ability to interpolate between these two extremes. This chapter details the methodology of this secondary training phase and its effects on the network's ability to interpolate.

5.1 Experiments and Results

5.1.1 Experimental Setup and Training

As previously established, the timestamped generator that was trained for 100 epochs in Chapter 4 is reused for this investigation. The secondary training phase that is applied to this model involves the use of 17 time-lapse sequences. The time-lapse data is therefore very limited, much more so than the day-night training data.

A selection of frames from each sequence is divided into five categories, representing timestamps 0, 0.25, 0.5, 0.75 and 1. A timestamp of 0 represents daytime, and a timestamp of 1 represents night-time. While 10 possible translations can be trained from these 5 timestamps, for simplicity only four are trained, as outlined in Table 5.1.

Dedicated discriminators are used for each timestamp, meaning five separate discriminators

Timestamps	Description	Training Ratio
$0 \leftrightarrow 0.25$	Day $\leftrightarrow$ Early Dusk	1
$0 \leftrightarrow 0.50$	Day $\leftrightarrow$ Dusk	1
$0 \leftrightarrow 0.75$	Day $\leftrightarrow$ Late Dusk	1
$0 \leftrightarrow 1.00$	Day $\leftrightarrow$ Night	2

Table 5.1: Overview of the transitions between timestamps that were explicitly trained, including the frequency with which each mapping is trained relative to others. This structure aims to ensure balanced training across all specified transitions. Emphasis is maintained on the day-to-night translation, as it was found empirically that the strength of this translation is degraded as the intermediate mappings are trained.

are used. The day and night discriminators that were developed along with the timestamped generator in the first phase of training are reused for timestamps 0 and 1, but the intermediate discriminators are trained from scratch. Each mapping undergoes training in the same manner as typical CycleGAN models, continuing to rely on a cycle-consistency loss term. To prevent the model from overfitting specific transitions, the training process is designed to cycle through each mapping sequentially, allocating double the training iterations to the 0 to 1 mapping, as outlined in Table 5.1. This added emphasis on the day-night mapping is included as it was found empirically that the strength of this mapping gradually degrades throughout the training process.

5.1.2 Evaluation Methods

While the FID and KID were used to provide supplementary numeric information in the previous Chapters, in the experimentation with time-lapse generation these metrics are not applicable, as there is an insufficient number of samples to ensure statistical stability in calculating these values. Therefore, the sole method of evaluation in this experimentation is direct visual assessment.

Given the limited size of the training set during the secondary training phase, it is unrealistic to expect the network to learn complex features associated with the intermediate timestamps. Instead, the primary focus of this phase is to assess whether the secondary training enhances the model's interpolation capabilities. This is evaluated by generating and comparing intermediate images both before and after the time-lapse training, thereby examining improvements in the model's ability to create transitions between timestamps as a result of the time-lapse training. Therefore, the overall smoothness of the transition is the primary consideration in the visual assessment.

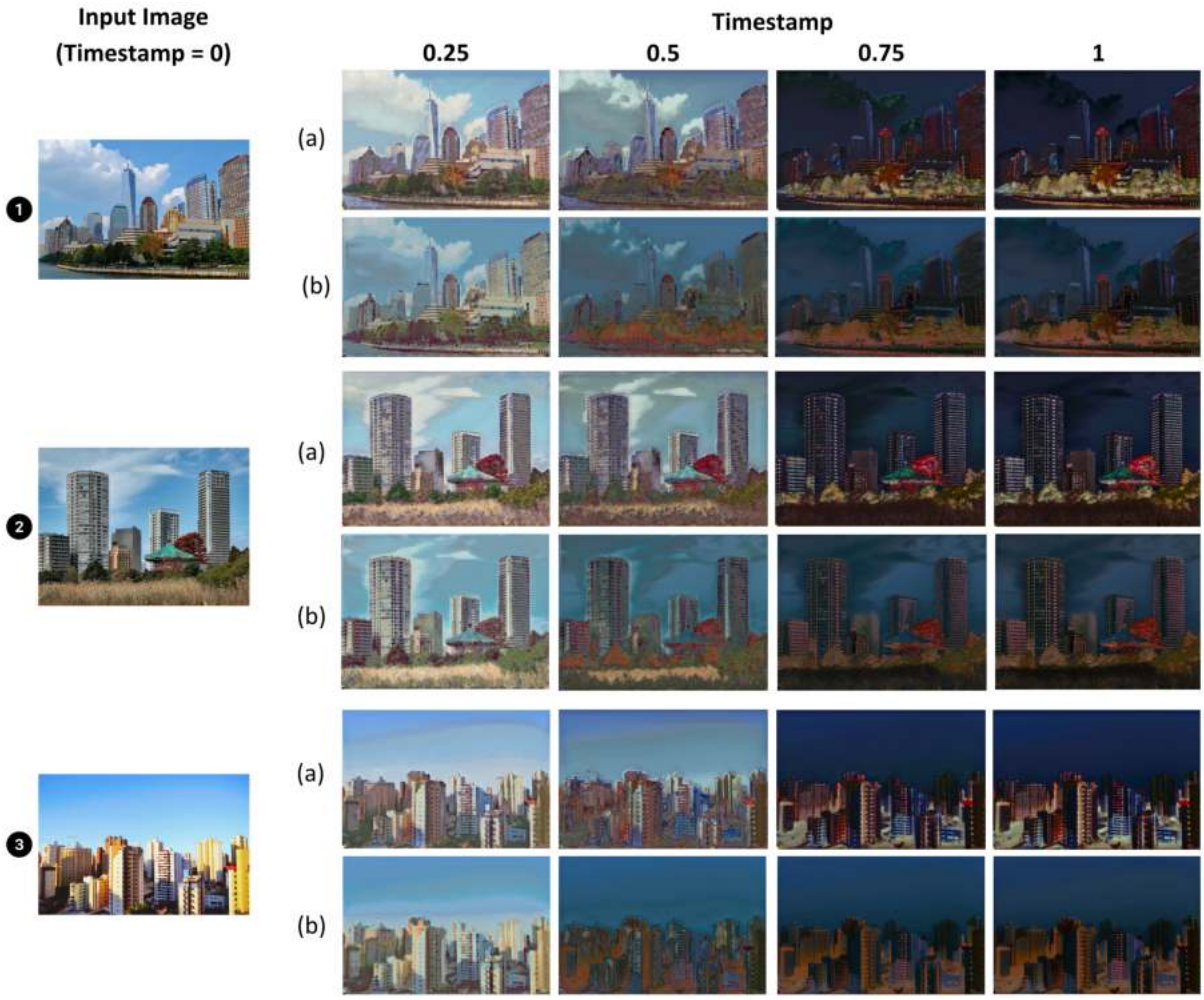

Figure 5.1: The effect of a secondary training phase that uses time-lapse data on the ability of the model to interpolate is investigated by comparing the outputs from the model before the secondary training phase to the outputs after the secondary training phase. The images on the left are the original input images. To the right, the synthetic images generated for different timestamp inputs are shown, ranging from a timestamp of 0.25 to a timestamp of 1. For each input image, the outputs are shown from the model (a) before time-lapse training, and (b) after time-lapse training.

5.1.3 Results

A subset of the validation outputs from the timestamped generator before and after the time-lapse training phase are compared in Figure 5.1. The time-lapse training appears to smoothen the transition between daytime and night-time, but this comes with the trade-off of a reduction in overall image quality. It also seems that learning the intermediate values degrades the strength of the mapping between the two extremes (day and night). However, while the synthetic time-lapses from the timestamped generator are far from perfect, the time-lapse data does appear to improve the intermediate outputs overall.

5.2 Conclusion

The results of this investigation suggest that using time-lapse training data improves the interpolation of the timestamped generator between daytime and night-time lighting conditions. However, it also introduces a trade-off in the form of diminished feature preservation and overall image quality.

While some insights can be drawn from the experimentation in this chapter, the time-lapse dataset size is a significant limitation that impedes the exploration of the full potential of the timestamped generator. A larger dataset could potentially enable the model to learn more complex features and interpolate better without sacrificing image quality. Furthermore, in a scenario with a greater quantity of time-lapse data, more timestamps could be used to generate fine-grained synthetic time-lapses. A larger training set size would also facilitate augmentations to the training process to refine the model's ability to interpolate, such as the introduction of random noise to the timestamp values during training. This would likely improve the robustness of the model. However, these additional elements were not implemented, and only five timestamps were used due to the desire to simplify the training process to account for the limited training data.

Another consideration is the use of dedicated discriminators for each of the five timestamps. This is likely not an optimal setup, especially in the context of a limited amount of training data, due to the need to train the intermediate discriminators from scratch. The generator likely overpowers the intermediate discriminators quite quickly during training, eliminating the need to learn any high-level features to fool the intermediate discriminators. This likely results in only superficial modifications in the intermediate outputs, such as adjustments to overall brightness. The generator is not encouraged to learn intricate details such as the dimming of window lights or other complex transformations due to the intermediate discriminators' inability to recognise these nuanced features. Altering the discriminator portion of the network might solve this issue: instead of multiple discriminators, a single, timestamped discriminator could be implemented that outputs the predicted timestamp in addition to the prediction of whether the image is real or fake. This would reduce the network to a single, simple GAN with one generator and one discriminator, which could lead to a more stable and effective training process.

Despite the challenges faced in this experimentation, the fundamental concept of incorporating time-lapse data into the training process displays promise. The network manages to perform the desired interpolation effectively, even with the constraints of a limited training set, suggesting that the fundamental design is sound. The results of this investigation establish a strong basis for future research, which could either involve refining the architecture or applying a larger training set to the current design.

6 Conclusions and Future Work

This research has established several methods of improving the output quality of a basic CycleGAN architecture for day-to-night image translation. In particular, the change to a U-Net structure with a pre-trained ResNet-18 encoder provides significant benefits by accelerating convergence during training and reducing visual artefacts. The exploration of encoder sharing has shed light on the effects of the latent representations of the network on output quality. Furthermore, the development of a single generator that models day and night as existing within a continuous domain is a significant contribution, showing promising results despite the limitations posed by a small training dataset.

The results of this research also lay a foundation for several intriguing areas of future work. One potential research topic is applying the timestamped generator to other translation tasks that benefit from a continuous model, such as modifying weather conditions in images or altering the apparent age in facial photos. A recurring issue in this project was the lack of training data. Therefore, another opportunity for future research is to replicate these experiments with a larger training set. Finally, another topic for future work is optimising the timestamped generator network. As previously discussed, replacing the multiple discriminators with a single, timestamped discriminator could improve performance. Alternatively, in a scenario with a greater quantity of time-lapse data, multiple discriminators may be the superior option, especially if the training process can be reduced to a single phase that uses time-lapse data from the beginning. In this scenario, the intermediate discriminators would not be overpowered by the timestamped generator and the ability of the network to learn complex features for the intermediate translations may be improved as a result.

In conclusion, this report has thoroughly addressed the initial research objectives, offering insights into possible improvements of CycleGAN and similar techniques. This work contributes to the ongoing development of image translation techniques and sets a solid groundwork for future research.

Bibliography

[1] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic tone reproduction for digital images," *ACM Trans. Graph.*, vol. 21, no. 3, p. 267–276, jul 2002. [Online]. Available: https://doi.org/10.1145/566654.566575

[2] J. Morovic and P.-L. Sun, "Accurate 3d image colour histogram transformation," *Pattern Recognition Letters*, vol. 24, no. 11, pp. 1725–1735, 2003.

[3] F. Pitie, A. C. Kokaram, and R. Dahyot, "N-dimensional probability density function transfer and its application to color transfer," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2. IEEE, 2005, pp. 1434–1439.

[4] F. Luan, S. Paris, E. Shechtman, and K. Bala, "Deep photo style transfer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4990–4998.

[5] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong, and L. Lin, "Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[6] H. Yu, N. Xu, Z. Huang, Y. Zhou, and H. Shi, "High-resolution deep image matting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 4, pp. 3217–3224, May 2021. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16432

[7] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[8] V. F. Arruda, T. M. Paixao, R. F. Berriel, A. F. De Souza, C. Badue, N. Sebe, and T. Oliveira-Santos, "Cross-domain car detection using unsupervised image-to-image translation: From day to night," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[9] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, 10 2016.

[10] Q. Gu, G. Wang, M. T. Chiu, Y.-W. Tai, and C.-K. Tang, "Ladn: Local adversarial disentangling network for facial makeup and de-makeup," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[11] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," 2018.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[13] F. Pitié, "Advances in colour transfer," *IET Computer Vision*, vol. 14, no. 6, pp. 304–322, 2020.

[14] A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 228–242, 2008.

[15] F. Pitié, A. C. Kokaram, and R. Dahyot, "Automated colour grading using colour distribution transfer," *Computer Vision and Image Understanding*, vol. 107, no. 1, pp. 123–137, 2007, special issue on color image processing. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314206002189

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[17] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[18] Resales, Achan, and Frey, "Unsupervised image translation," in *Proceedings Ninth IEEE International Conference on Computer Vision*. IEEE, 2003, pp. 472–478.

[19] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," *Advances in neural information processing systems*, vol. 29, 2016.

[20] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 1857–1865.

[21] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2849–2857.

[22] A. Almahairi, S. Rajeshwar, A. Sordoni, P. Bachman, and A. Courville, "Augmented cyclegan: Learning many-to-many mappings from unpaired data," in *International conference on machine learning*. PMLR, 2018, pp. 195–204.

[23] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/dc6a6489640ca02b0d42dabeb8e46bb7-Paper.pdf

[24] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[25] A. Anoosheh, E. Agustsson, R. Timofte, and L. Van Gool, "Combogan: Unrestrained scalability for image domain translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 783–790.

[26] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 172–189.

[27] H.-Y. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. Singh, and M.-H. Yang, "Drit++: Diverse image-to-image translation via disentangled representations," *International Journal of Computer Vision*, vol. 128, pp. 2402–2417, 2020.

[28] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," *Advances in neural information processing systems*, vol. 29, 2016.

[29] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *International conference on learning representations*, 2016.

[30] J. Cao, O. Katzir, P. Jiang, D. Lischinski, D. Cohen-Or, C. Tu, and Y. Li, "Dida: Disentangled synthesis for domain adaptation," *arXiv preprint arXiv:1805.08019*, 2018.

[31] D. Shiotsuka, J. Lee, Y. Endo, E. Javanmardi, K. Takahashi, K. Nakao, and S. Kamijo, "Gan-based semantic-aware translation for day-to-night images," in *2022 ieee international conference on consumer electronics (icce)*. IEEE, 2022, pp. 1–6.

[32] A. Anoosheh, T. Sattler, R. Timofte, M. Pollefeys, and L. Van Gool, "Night-to-day image translation for retrieval-based localization," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5958–5964.

[33] F. Pizzati, P. Cerri, and R. de Charette, "Comogan: continuous model-guided image-to-image translation," 2022.

[34] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[35] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[36] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/8a3363abe792db2d8761d6403605aeb7-Paper.pdf

[37] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf

[38] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," 2015.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "Imagenet: a large-scale hierarchical image database," 06 2009, pp. 248–255.

[40] M. J. Chong and D. Forsyth, "Effectively unbiased fid and inception score and where to find them," 2020.

[41] M. Bińkowski, D. Sutherland, M. Arbel, and A. Gretton, "Demystifying mmd gans," 01 2018.

[42] S. V. Ravuri and O. Vinyals, "Seeing is not necessarily believing: Limitations of biggans for data augmentation," 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:195527523

[43] E. Betzalel, C. Penso, A. Navon, and E. Fetaya, "A study on the evaluation of generative models," 2022.

[44] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021.

[45] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," 2016.

[46] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[47] P. Ulmas and I. Liiv, "Segmentation of satellite imagery using u-net models for land cover classification," 2020.

[48] K. Zou, X. Chen, Y. Wang, C. Zhang, and F. Zhang, "A modified u-net with a specific data argumentation method for semantic segmentation of weed images in the field," *Computers and Electronics in Agriculture*, vol. 187, p. 106242, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169921002593

[49] V. Iglovikov and A. Shvets, "Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation," 2018.

[50] G. Rahmon, F. Bunyak, G. Seetharaman, and K. Palaniappan, "Motion u-net: Multi-cue encoder-decoder network for motion segmentation," in *2020 25th international conference on pattern recognition (ICPR)*. IEEE, 2021, pp. 8125–8132.

[51] N. Sharma and S. Gupta, "Semantic segmentation of gastrointestinal tract using unet model with resnet 18 backbone," in *2023 International Conference in Advances in Power, Signal, and Information Technology (APSIT)*, 2023, pp. 226–230.

[52] Keras, "Cyclegan example," 2020, [Online] Available at: https://keras.io/examples/generative/cyclegan/.

[53] Denaya, "CycleGAN—Introduction + PyTorch Implementation — chilldenaya," https://medium.com/@chilldenaya/cyclegan-introduction-pytorch-implementation-5b53913741ca, [Accessed 03-04-2024].

[54] "GitHub - a7med12345/Cycle-GAN-with-Unet-as-GENERATOR — github.com," https://github.com/a7med12345/Cycle-GAN-with-Unet-as-GENERATOR, [Accessed 03-04-2024].

[55] H. Hwang, "Unpaired day and night cityview images," 2021, [Online] Available at: https://www.kaggle.com/datasets/heonh0/daynight-cityview.

[56] "Benchmarking Long-term Visual Localization — visuallocalization.net," https://www.visuallocalization.net/datasets/, [Accessed 04-04-2024].