Exploring Cycle-Consistent Adversarial Networks for Smartphone Photography: Translating iPhone to DSLR Images

Soon Jing Yi (P2308940)

DAAA/FT/2B/22

Abstract

This paper investigates the application of Cycle-Consistent Adversarial Networks (CycleGAN) [17] for enhancing smartphone photography, focusing on translating images between iPhone and Digital Single-Lens Reflex (DSLR) cameras. The research explores the effectiveness of CycleGAN in improving smartphone photographs to resemble DSLR quality without requiring paired training data. A CycleGAN model was developed using TensorFlow and Keras, utilizing ResNet-based generators and PatchGAN discriminators architectures [17], and trained on the iPhone2DSLR_flower dataset [2]. The study presents both qualitative and quantitative evaluations, demonstrating the effectiveness of CycleGAN in translating images with limited computational resource.

1. Introduction

Over the years, computational photography has advanced significantly, particularly in the context of smartphone cameras [16]. As techniques such as High Dynamic Range (HDR) imaging, image stacking, and deep learning models become increasingly integrated into camera applications, modern smartphones are now capable of producing high quality images and videos even in notoriously challenging environments, such as nighttime scenes [16]. This paper evaluates whether images captured with smartphones, which often exhibit minimal background blur, lack of bokeh, and limited contrast and detail due to their small sensor sizes and compact lenses [16], can be transformed to resemble those taken with high-end DSLR cameras through unpaired image-to-image translation [17].

Unpaired image-to-image translation methods, such as CycleGAN, offer a significant advantage by eliminating the need for paired training data, which is often challenging and costly to obtain [17]. For example, obtaining exact pairs of images taken with different devices, such as DSLR and iPhone cameras, is difficult due to variations in framing, sensor characteristics, and image processing.

This paper examines the CycleGAN model, focusing on its application for enhancing smartphone photographs using TensorFlow and Keras [10, 17]. In this study, the CycleGAN model is applied to the iPhone2DSLR_flower dataset. Additionally, the paper discusses the limitations of CycleGAN and highlights recent advancements made by other researchers in the field.

2. Related works

2.1 Generative Adversarial Networks (GANs)

First introduced in 2014 [4], GANs continue to be highly relevant a decade later, serving as fundamental building blocks for generative models. The generator network aims to produce synthetic data that is indistinguishable from real data, while the discriminator network attempts to differentiate between real and generated samples [4, 15]. This adversarial training process can be formulated as a minimax game, where the generator tries to minimize the probability of the discriminator correctly classifying samples, while the discriminator tries to maximize this probability [4, 15].

2.2 Image-to-Image Translation

The goal of Image-to-image translation is to learn a mapping between an input image and an output image [12, 13]. This field has evolved significantly over the past two decades, with approaches ranging from non-parametric methods to deep learning-based techniques [12, 13]. The concept of image-to-image translation can be traced back to Hertzmann et al.'s work on "Image Analogies" in 2001 [7].

Building upon GANs, the pix2pix framework was proposed in 2017, which marked a significant milestone in image-to-image translation [9]. Pix2pix introduced conditional GANs for paired image-to-image translation, providing a general-purpose solution for many image-to-image translation tasks [9]. The model learned both the mapping from input image to output image and an appropriate loss function to train this mapping [9].

However, the main problem of paired image-to-image translation is obtaining the dataset itself. Obtaining such paired data can be challenging and resource-intensive. For many tasks, the desired output is complex, requiring artistic or highly specific input-output pairs [17]. In the context of translating iPhone images to DSLR images using paired training data, it is necessary for the same scene or subject to be captured by both devices with as much similarity as possible. This requirement poses significant challenges, as it demands that the framing, surrounding environment, and composition be closely aligned within each image pair. Furthermore, relying only on paired data restricts the amount of training data that can be used, as unpaired data in either domain cannot be leveraged [17].

2.3 Unpaired Image-to-Image Translation (CycleGAN)

To address the limitations of paired data, CycleGAN, a model for unpaired image-to-image translation was introduced [17]. The use of CycleGAN has been demonstrated in various domains, including translating photographs to artistic styles and enhancing low-quality to high-quality images. [17]

CycleGAN addresses the issue of paired image-to-image translation by learning to translate between domains without paired input-output examples. [17]. The goal is to train a mapping G:X→Y such that the output $\hat{y} = G(x)$ is indistinguishable from images in Y, using an adversarial network. [17]

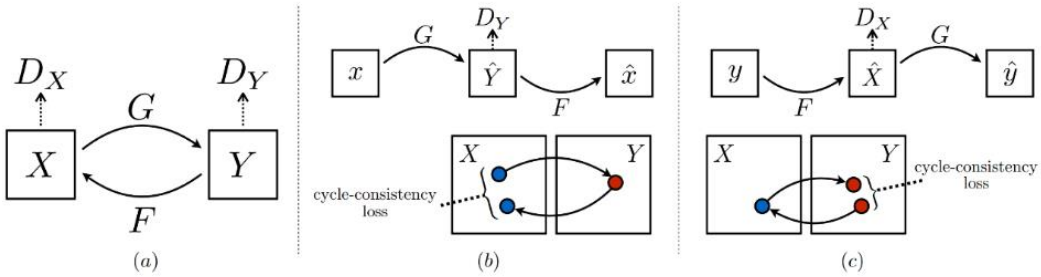

Figure 1: Visual of cycle-consistency loss [17]

However, the primary challenge with adversarial training is that it does not ensure that a specific input image will be mapped to a specific desired output image [17]. Rather, the adversarial loss functions are designed to only ensure that the generated images are statistically similar to the real images in the target domain [17]. This means the model learns to produce outputs that conform to the general characteristics of the target domain, but it does not guarantee a precise, one-to-one correspondence between each individual input and its intended output [17].

To solve this challenge and preserve the key structures of the input image so that the generated image retains a close resemblance to the original input, CycleGAN introduces the concept of cycle consistency [17]. This means that translating an image from one domain to another and then back should return the original image, as shown in Figure 1(a) [17]. Mathematically, for mappings G:X→Y and F:Y→X, $F(G(x))$ should approximate x, as shown in Figure 1(b), and $G(F(y))$ should approximate y, shown in Figure 1(c) [17].

Researchers from the original CycleGAN paper introduced a good analogy to explain the concept of cycle consistency in CycleGAN [17]. They described the model as training two "autoencoders" simultaneously [3, 17]. An autoencoder is a type of neural network that learns to compress data into a lower-dimensional representation and then reconstruct it [3]. In CycleGAN, instead of simply compressing and reconstructing an image, the networks translate it into another domain and then back again [17]. The key idea is that after this round-trip translation, the resulting output should closely match the original image.

Therefore, combining cycle consistency loss and adversarial loss helps produce realistic translations, preventing the model from making arbitrary changes that don't preserve the content of the original images [17]. This approach allows CycleGAN to learn meaningful translations between domains without requiring paired examples, making it a powerful tool for various image-to-image translation tasks [17].

2.4 PatchGAN

PatchGAN is a variant of Generative Adversarial Networks (GANs) that introduces a local discriminator for improving image generation quality [8]. Unlike traditional GANs, which use a global discriminator to assess the entire image [4], PatchGAN employs a discriminator that evaluates individual patches of the image to determine whether each patch is real or fake [8, 17]. This approach is beneficial because PatchGAN generally has fewer parameters, operates more quickly, and can be applied to images of any size [8, 17]. Additionally, PatchGAN's focus on local texture and detail enables finer classification of images, effectively serving as a measure of texture or style loss [8, 17]

2.5 Residual Networks (ResNets)

ResNets were introduced to address the challenges of training very deep neural networks. ResNets employ residual blocks that include shortcut connections, which bypass one or more layers in the network [6]. This architecture enables the training of much deeper networks by mitigating the vanishing gradient problem and facilitating the learning of residual mappings [6]. In the context of CycleGAN, ResNet's residual blocks are utilized in the generator networks to enhance image translation quality and ensure stable training [17]. The incorporation of ResNets allows CycleGAN to generate more coherent and visually appealing images by leveraging the residual learning framework to preserve detailed information throughout the network [17].

3. Approach

3.1 Dataset

The iPhone2DSLR_flower dataset contains images of flowers captured with iPhones, where the backgrounds are in focus, as well as DSLR images with a shallow depth of field, higher resolution, and more stylized color grading. For this study, the dataset was downloaded from the CycleGAN dataset collection provided by the Berkeley Artificial Intelligence Research Laboratory at the University of California [2]. Most images featuring sceneries, human portraits, and other subjects unrelated to flowers were manually removed for data consistency.

A random sample of 500 images from iPhones and DSLRs each was selected for training, and 200 images from iPhones and DSLRs each for testing. This subset was used to train and evaluate the CycleGAN model. The reduction in dataset size was necessary due to limited computational resources as this study was conducted on a laptop [14]. Although this approach helped manage computational constraints, it resulted in a reduction in data diversity and overall training data.

3.2 Model

The CycleGAN model, built using TensorFlow and Keras in Python, for this study was adapted from the official Keras documentation [10], where it was originally built to translate images of horses into zebras using the horse2zebra dataset [2]. In this study, the code was modified to work with the iPhone2DSLR_flower dataset, and additional evaluations, including training curves and image reconstruction were incorporated.

The CycleGAN model uses two ResNet generators and two PatchGan discriminators.

The ResNet generators used consists of an initial convolutional layer with 64 filters and a large kernel size of 7×7, followed by reflection padding to manage border artifacts [5, 6]. This is followed by two downsampling blocks, where the number of filters doubles after each block, progressively capturing more complex features [6]. They are used to reduce the spatial dimensions and increase the filter count. In the core of the generator, nine residual blocks are used to capture intricate image details and preserve important features through residual connections [6]. Each residual block consists of convolutional layers with ReLU activations, which help in learning residual mappings effectively. The network then includes two upsampling blocks to restore the image's spatial dimensions and

reduce the number of filters [6]. The final output is generated through a convolutional layer with 3 filters, corresponding to the RGB colour channels, followed by a hyperbolic tangent activation function to produce the output image in the normalized target domain [6, 17]. There are two ResNet generators in total: generator G transforms images from domain X (iPhone photographs) to domain Y (DSLR photographs), while generator F transforms images from domain Y back to domain X [4].

The PatchGAN discriminators in CycleGAN uses a convolutional architecture to classify image patches as real or fake. In this case, each discriminator starts with 64 filters and a 4×4 kernel, applying strides of 2 to downsample and capture basic features [8]. The model includes three downsampling blocks, where filters double each time, with strides of 2 for the first two blocks and 1 for the final block to capture finer details. It ends with a single filter convolutional layer, producing a one-channel output indicating the real or fake probability of each patch [4]. There are two PatchGAN discriminators in total: discriminator X is used to distinguish between real images from domain X and fake images generated by generator F, while discriminator Y is used to distinguish between real images from domain Y and fake images generated by generator G [4].

3.3 Training

The generators and discriminators were trained using Adam optimizers [11] with the same learning rate of 2e-4 and a beta 1 value of 0.5. The CycleGAN model was then trained with the dataset for 50 epochs using an NVIDIA GeForce RTX 3060 Laptop GPU [14]. The training process averaged approximately 350 seconds per epoch, resulting in a total training duration of around 5 hours.

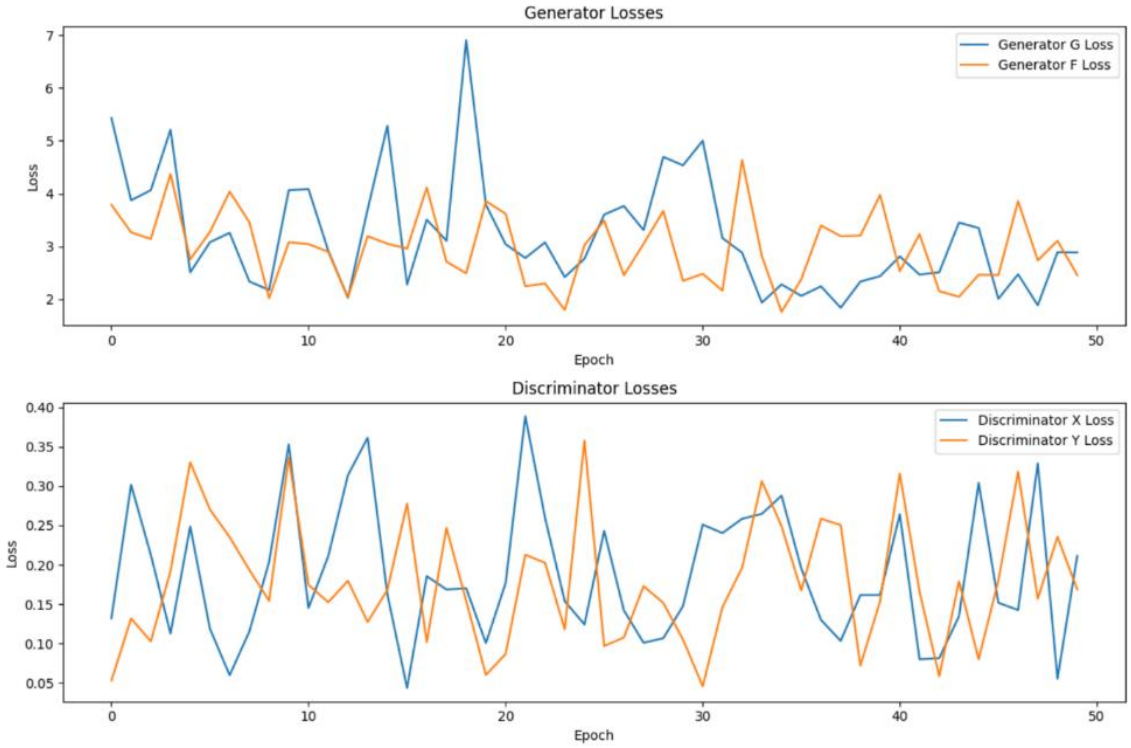

Figure 2: Generator and discriminator losses of the CycleGAN over 50 epochs of training. Both the generator losses for G and F show an overall decreasing trend, though with significant fluctuations. The loss for generator G exhibited more fluctuations but showed a higher overall decrease, reaching an approximated final loss of 3 at epoch 50. This decline suggests improvements in the generator's performance in producing realistic translations during training [17]. However, it had not yet stabilized or converged, indicating the need for further training and potential improvement in performance. The discriminator losses for both X and Y domains are similar and exhibited fluctuations between 0.40 and 0.05, centering around 0.20. Extending training beyond 50 epochs could potentially lead to further loss reduction and convergence [9]. Additionally, fine-tuning hyperparameters, particularly the learning rate, may help mitigate fluctuations and improve convergence [9].

3.4 Results

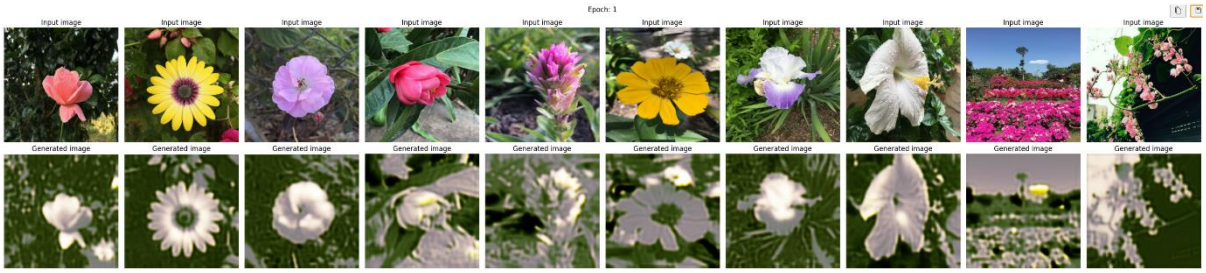

Figure 3: input and generated images by CycleGAN during the first epoch.

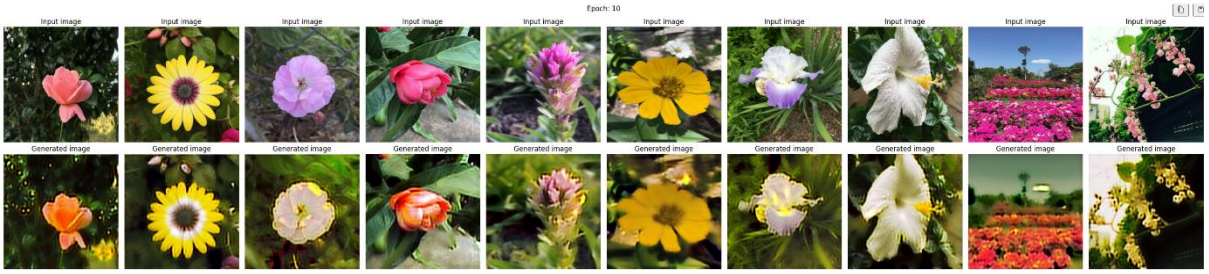

Figure 4: input and generated images by CycleGAN during the 10th epoch.

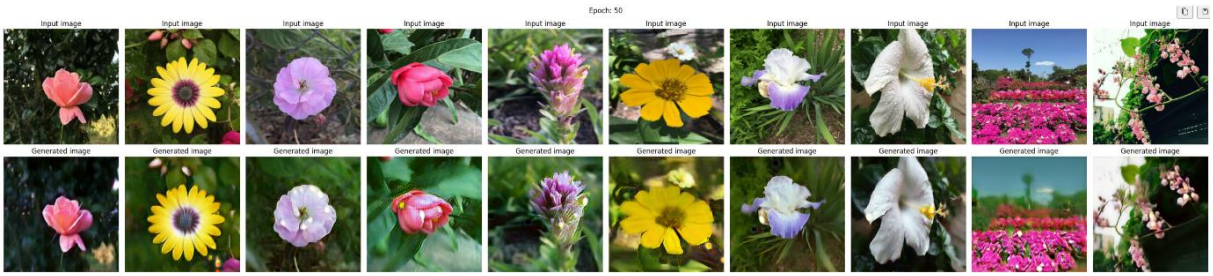

Figure 5: input and generated images by CycleGAN during the 50th epoch.

Figures 3, 4, and 5 illustrate the progression of the CycleGAN model from Epoch 1 to Epoch 50. In Epoch 1, shown in Figure 3, the generated images largely retain the structural features of the input images, preserving the overall shapes and structures of the flowers. However, the colors are inaccurately rendered, with flowers appearing gray and backgrounds dark green. This suggests that while the model has begun to differentiate between flowers and background elements, it has not yet learned to preserve color fidelity, resulting in an almost grayscale output.

By Epoch 10, the model starts to restore some of the original colors, although inconsistencies remain. For instance, as seen in Figure 4, the third image from the left shows the flower being translated from purple to yellow, and the last image from the left depicts the color of the flower changing from pink to yellow. Additionally, the perceived resolution of the generated images is lower, with noticeable color noise which was not present in the original images.

By Epoch 50, the model demonstrates significant improvements. It accurately reproduces the colors of the flowers and incorporates color grading similar to that seen in the DSLR photographs. For example, the first and second images from the right in Figure 5 show a blurred background with a focused foreground, effectively differentiating between flowers and the background. Despite these improvements, some generated images remain noisy compared to the originals and contain "hallucinations". For example, the fourth image from the left contains a yellow artifact in one of the flower's petals, which is absent in the original image.

Figure 6: An iPhone image from the 'testA' dataset was fed into generator G to produce a translated DSLR image. This generated DSLR image was then passed through generator F to reconstruct it back to the original iPhone image.

Figure 7: An iPhone image from the 'testA' dataset was fed into generator G to produce a translated DSLR image. This generated DSLR image was then passed through generator F to reconstruct it back to the original iPhone image. Time taken for translation to DLSR was 0.09126 seconds.

Using the CycleGAN model with the final weights from Epoch 50, image reconstruction was conducted to demonstrate the outputs of generators G and F, which are crucial for cycle consistency [17]. Initially, an iPhone image is input into generator G to produce a DSLR-styled image. This generated DSLR image is then processed by generator F to reconstruct the original iPhone image.

Figures 6 and 7 show that iPhone images, characterized by a deep depth of field and in-focus backgrounds, are transformed by generator G into DSLR-styled images with a shallow depth of field with bokeh effects. The generated DSLR images exhibit lower overall exposure and are generally darker compared to the originals, as shown in Figure 6, where the sky in the generated image is less overexposed. Artifacts are present, particularly in Figure 6, where the flower details become indistinct, indicating that the generator struggled to preserve finer details during translation.

The reconstructed iPhone images, shown in the right-most images in Figures 6 and 7, reveal that generator F successfully restores the background focus and wider depth of field, found in the original iPhone images. Additionally, the saturation and overall brightness of the image is increased. These results suggest that generator F effectively translates the output from generator G to closely resemble the original iPhone images.

Figure 8: iPhone input image and corresponding generated image from CycleGAN with a person as the main subject instead of flowers

4. Discussion

One drawback of using CycleGAN for enhancing image quality from smartphone cameras is the need for diverse training data. As shown in Figure 8, the iPhone2DSLR_flower dataset primarily consists of images where flowers are the main subject. Consequently, when inputting an image with a person, the results may be suboptimal, as the CycleGAN model may struggle to accurately translate the features of a person.

Further improvements could include increasing the number of training epochs, tuning hyperparameters such as the learning rates, and increasing the number of images in the dataset [9]. Increasing the number of training epochs allows the model to learn more detailed features and improve its translation. Tuning hyperparameters can help optimize the performance and smoothen the training convergence of both the generators and discriminators, leading to better image translation results [9]. Expanding the dataset to include a wider variety of image types and conditions can help the model learn more comprehensive features and improve its performance on previously underrepresented subjects. However, these improvements would result in increased time and computational resource requirements, as they require longer training periods to achieve the desired results, which is feasible if more powerful GPUs are used. Additionally, advancements have been made since the introduction of CycleGAN, such as Augmented CycleGAN, which learns many-to-many mappings rather than a deterministic one-to-one mapping [1].

In terms of translation speed, the CycleGAN model demonstrated the capability to translate the iPhone image in Figure 7 in approximately 0.09 seconds. Given that the input image was compressed to only 256 x 256 pixels, it is substantially lower in resolution compared to standard smartphone images, which typically range from 8 MP (3456 x 2304 pixels) to 12 MP (4032 x 3024 pixels). Thus, it is likely that image translation using CycleGAN could require significantly more time for smartphone photographs in original resolutions, due to the increase in computational load on the model.

By utilizing a larger and more diverse dataset, including various subject types such as portraits and landscapes, and images captured by specific camera models, it is possible to develop sophisticated image processing techniques using CycleGAN. Integrating CycleGANs directly into smartphone camera applications would allow users to conveniently apply different stylistic transformations to any image, emulating the characteristics of specific DSLR models or film cameras. Furthermore, CycleGAN extends beyond image enhancement. It can also transform photographs into paintings in the style of renowned artists like Van Gogh, or modify the environment depicted in the images, such as changing the scene from summer to winter [17]. This suggests that there are endless possibilities for increasing the capabilities and enhancing the creativity of computational photography through unpaired image-to-image translation.

5. Conclusion

This study provided a comprehensive overview of CycleGAN, explaining its underlying concepts and architecture. The development and evaluation of a CycleGAN model showcased its ability to effectively transform iPhone images into DSLR-styled outputs, despite limitations in computational resources and training data. The insights gained highlight both the strengths and limitations of the model, suggesting avenues for future research. Potential improvements include extending training duration, refining hyperparameters, and expanding the dataset to enhance performance and applicability. Overall, this research demonstrated that CycleGAN can be used for image enhancement and highlights the many possibilities for integrating CycleGANs into applications such as smartphone cameras.

6. References

1. Almahairi, A., Rajeshwar, S., Sordoni, A., Bachman, P., & Courville, A. (2018). Augmented CycleGAN: Learning Many-to-Many Mappings from Unpaired Data. arXiv preprint arXiv:1802.10151. 1

2. Berkeley AI Research. (n.d.). CycleGAN Datasets. 2
3. D. Bank, N. Koenigstein, and R. Giryes. (2020). Autoencoders. arXiv preprint arXiv:2003.05991, 2021. 3
4. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Networks. arXiv preprint arXiv:1406.2661. 4
5. Liu, G., Shih, K. J., Wang, T.-C., Reda, F. A., Sapra, K., Yu, Z., Tao, A., & Catanzaro, B. (2018). Partial Convolution based Padding. arXiv preprint arXiv:1811.11718. 5
6. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. arXiv preprint arXiv:1512.03385. 6
7. Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B., & Salesin, D. H. (2001). Image analogies. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques (pp. 327-340). 7
8. Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. arXiv preprint arXiv:1611.07004. 8
9. Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive Growing of GANs for Improved Quality, Stability, and Variation. arXiv preprint arXiv:1710.10196. 9
10. Keras. (n.d.). CycleGAN. 10
11. Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980. 11
12. Liu, H., & Guo, Z. (2020). Image-to-Image Translation: Methods and Applications. arXiv preprint arXiv:2101.08629. 12
13. Liu, M. Y., Breuel, T., & Kautz, J. (2017). Unsupervised Image-to-Image Translation Networks. arXiv preprint arXiv:1703.00848. 13
14. Notebookcheck. (n.d.). NVIDIA GeForce RTX 3060 Laptop GPU. 14
15. Odena, A., Olah, C., & Shlens, J. (2017). Conditional Image Synthesis With Auxiliary Classifier GANs. arXiv preprint arXiv:1610.09585. 15
16. Visionary.ai. (n.d.). How is Computational Photography Revolutionizing Smartphone Cameras? 16
17. Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. arXiv preprint arXiv:1703.10593. 17