Style Transfer: from pix2pix to CocosNet

SHI Juluan
SID: 1155160208

Abstract

In recent years, the field of image generation has witnessed noteworthy advancements, including the introduction of stable diffusion2[11] last year, which has enabled the generation of high-quality paintings within a short time. While stable diffusion2 belongs to the text-to-image domain, the image-to-image field has also experienced significant progress and notable achievements. In this paper, I have re-implemented two influential models, namely pix2pix[5] and cycleGAN[16], and applied them to generate comic-style faces from photographic-style inputs. The outcomes of this study have revealed both the strengths and limitations of these models. Furthermore, I have explored recent developments in this area and would like to introduce a more advanced framework called CocosNet[15], which is an exemplar-based image translation model.

1 Introduction

Since the advent of AlexNet in 2012, convolutional neural networks (CNNs) have become a dominant model in various computer vision tasks, including classification and segmentation. However, image style transfer poses a distinct challenge due to its subjective nature and the need to balance content and style effectively. CNN and its variations have shown remarkable capacity for extracting hidden features from input images, enabling the synthesis of new images based on the information contained in the input image. Nevertheless, to achieve satisfactory and robust results, careful model architecture design and loss function formulation are still needed.

For this project, I intend to delve into the history of the image style transfer task and provide an overview of the significant contributions made by various renowned papers. In particular, I have re-implemented two influential models, namely pix2pix and cycleGAN, and have employed them to generate comic-style faces. The results of the experiment indicate that pix2pix performs impressively in converting photographic images into comic-style images, although certain artifacts persist. However, the limitations of the pix2pix model remain significant, as it lacks robustness when tested with images sourced from the internet. In contrast, CycleGAN does not present similar issues and is effective in injecting color and texture style into input images. However, for certain cases, the cycle consistency loss appears to constrain the shape of the image, resulting in unsuccessful geometric alterations.

My original objective for this research project was to re-implement CocosNet, an impressive framework for image-to-image translation. However, during the course of my work, I encountered certain challenges. Specifically, the input data required for the CocosNet source code were preprocessed in a specific way, and the authors did not mention the techniques used in the paper, making it difficult for me to reproduce their results. Furthermore, successful training of the model required significant computational resources, which were not readily available to me. As a result, I have decided to focus my attention on the re-implementation of pix2pix and cycleGAN, as these models have shown promising results in style transferring task. Despite of above limitations, I believe it is still worth mentioning the technical details of CocosNet. CocosNet addresses the issue of geometric alterations in image style transfer that can occur with cycleGAN, and also provides more control over both the texture style and geometric aspects of the output images.

2 Related Work

One of the most intuitive methods for achieving image style transfer involves extracting the content from one image and the style from another, followed by combining them to produce a novel image. Gatys' group[3] successfully adopted this approach by utilizing the VGG model[13] for feature extraction and constructing a loss function that combines content loss and style loss for training. While the content representation can be easily extracted from each feature map layer, defining and quantifying style is a more challenging task. To address this issue, they used the Gram matrix to obtain the feature correlation (style representation) of the image. Despite the remarkable performance of this approach, the relationship between the Gram matrix and style representation remains unclear. In this regard, Yanghao Li[7] further proposed the idea that matching the Gram matrices of features maps is equivalent to minimize the Maximum Mean Discrepancy (MMD) with the second order polynomial kernel. That means neural style transfer is to match the feature distributions between the style images and the generated images.

While the above method for image style transfer have demonstrated outstanding performance and a clear understanding of the underlying principles, manually designing loss functions for the feature maps is widely regarded as a challenging task. Moreover, above method are primarily suitable for creating artistic images that do not require a high level of detail. It can be challenging to design a loss function that converts daytime images to nighttime images while preserving the detailed features of the original images. In fact, Qifeng Chen et al[1] used a pretrained VGG to construct the loss function for generating high quality photographic images from semantic layouts and achieved satisfactory results later on. However, the success of Chen's story also shows that the design of loss function is demanding for image generation tasks.

Fortunately, with the advent of Generative Adversarial Networks (GANs)[2][4], solving some of these challenging tasks has become feasible and the solution is intuitive to understand. The discriminator in GANs can differentiate between labeled and generated fake images, thereby assisting the generator in producing higher quality results that are indistinguishable from labeled images. In this context, Phillip Isola and his team[5] proposed pix2pix using conditional GANs (cGANs) for more general image style transfer tasks. The loss function is composed of standard GAN loss and L1 loss. The GAN loss is used for encouraging the generated images to be indistinguishable from target images while the L1 loss is used for measuring the absolute difference between target images and generated images. Building on pix2pix, Ting-Chun Wang[14] used a multi-scale generator and discriminator architecture to generate style-transferred images with higher quality. While traditional image style transfer tasks have achieved satisfactory performance, generating high-quality photographic images from semantic layouts still has some potential challenges. Unconditional-based normalization methods may clear the semantic information during training, resulting in unrealistic generated results. To tackle this problem, Park and his team[10] proposed spatially-adaptive normalization (SPADE), which uses a set of learned affine parameters that are conditioned on the semantic label of the input image. These affine parameters are used to normalize the activations of the feature map in a way that is specific to the semantic content of the image.

Thanks to the efforts of many researchers, generating high-quality images from conditional inputs has become practical. In recent years, Rui Liu and his team[8] integrated contrasting learning[9] with existing conditional generative adversarial networks to produce diverse output images. It is clear that AI's performance in computer vision is becoming increasingly impressive.

In addition to the powerful supervised learning models mentioned above, cycleGAN[16] is also worth mentioning because it relies only on unpaired images for style transfer tasks. As the model's name suggests, it was trained in a cyclical manner using two pairs of generator and discriminator to learn the style and content of images. However, the cycle consistency loss constraint in the model tends to cause the output image to follow the shape of the input images, which can make it difficult to generate images that require changes in input shapes.

In this project, I focused on re-implementing pix2pix and cycleGAN models from the ground up to generate comic-style faces and evaluate their effectiveness.

3 Data

The dataset utilized in my project is the Comic faces (paired, synthetic) v2, which comprises of 10,000 pairs of facial and darkish red style comic images. The images in the original dataset have a resolution of 1024x1024 pixels, but for the purpose of training and testing, I resized them to a resolution of 256x256 pixels. This dataset was specifically designed for face-to-comic image translation tasks and was well-suited for for the re-implemented pix2pix and cycleGAN models.

Dataset Link: `https://github.com/Sxela/face2comics`

Figure 1: face2comic dataset samples

4 Approach

Overall, I followed the implementation details outlined in the papers to re-implement pix2pix and cycleGAN, and paid significant efforts to ensure the correctness of my implementation. Both of these models use the concept of adversarial loss, whereby a generator is assisted by a discriminator in producing output that is indistinguishable from the desired output. The primary difference between the two models is that cycleGAN can be trained without requiring paired data, thereby lowering the dataset requirements. Unlike traditional GANs that use a random noise vector as input, pix2pix and cycleGAN use GANs in a conditional setting. This means that the input condition gives additional information to guide the generator in producing the desired output. Consequently, the generator can learn a mapping from the input condition to the target output.

4.1 Pix2Pix Model

The Pix2pix model comprises of two crucial components, namely the generator and discriminator (see Figure 2).

Generator I utilized the U-Net model[12], a typical model utilized for generating high-resolution outputs. However, I believe that any model with the capability to produce high-resolution images can also be utilized as a generator. The essential concept is that the model needs to be first compressed into a latent space, and then the latent code is passed through a sequence of upsampling layers. As I understand it, the high-level structure of the input images is similar to the desired output, and the bottleneck can capture the high-level idea of the input images. Meanwhile, the upsampling layers will concentrate on the details and style transfer aspects to infuse new style into the latent code. Compared to the standard encoder and decoder architecture, U-Net incorporates skip connections between the downsampling and upsampling layers, providing more guidance for low-level information like the position of edges.

Discriminator I employed PatchGAN as the discriminator, which deviates from the traditional approach where the discriminator maps the input image to a single scalar value to distinguish between "real" and "fake" inputs. Instead, PatchGAN is a fully convolutional network that maps from 256x256 to an NxN array. The benefit of using PatchGAN is that it saves computational power since it does

not include a linear layer, and each grid can focus on a specific receptive field in the input image. Additionally, since the L1 loss already encourages the generator to capture the low-level frequencies of target images, PatchGAN complements the L1 loss and focuses solely on high-frequency structure.

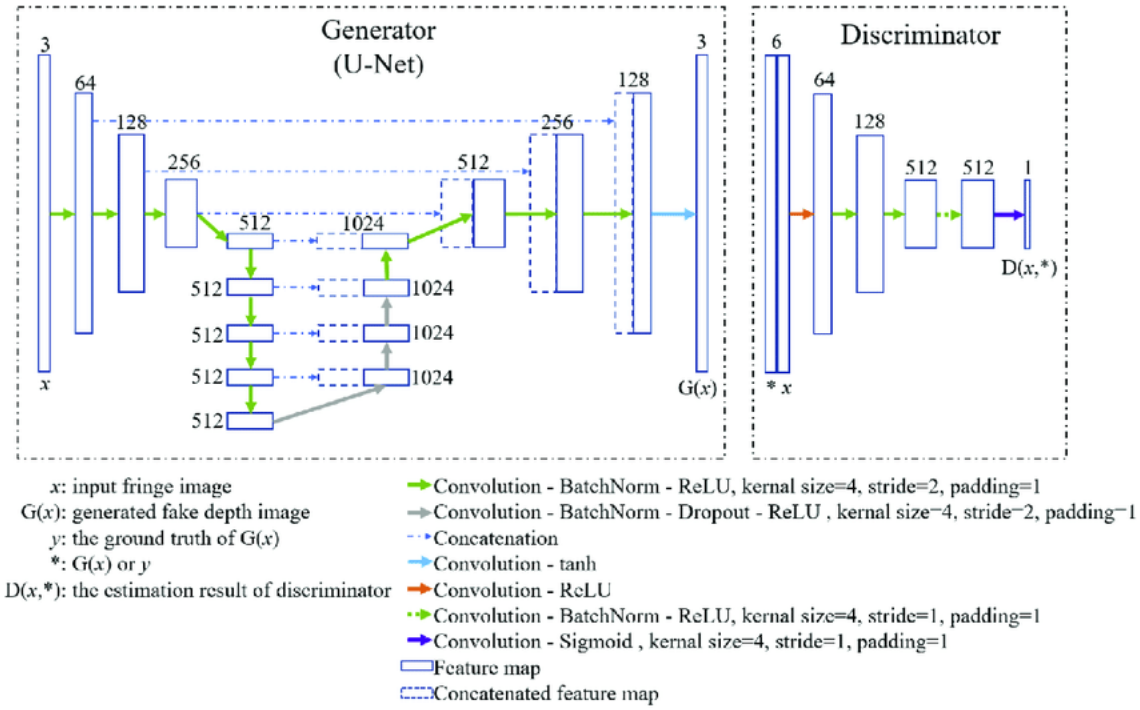

Figure 2: pix2pix architecture

Loss function The loss function comprises two components: adversarial loss and L1 loss:

$$\mathcal{L}_{pix2pix} = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_x[\log(1 - D(x, G(x)))] + \lambda \mathbb{E}_x[y - G(x)_1] \quad (1)$$

In the training process, the discriminator (D) aims to maximize the loss function while the generator (G) aims to minimize it. This results in the generator being able to learn from the discriminator and generate higher quality output. The decision to use L1 loss instead of L2 loss is due to the fact that L2 loss may promote more blurring.

4.2 CycleGAN

While pix2pix can handle multiple image-to-image generation problems, it demands high-quality paired images, which can be difficult to collect and may require extensive manual work for dataset creation. On the other hand, cycleGAN is famous for eliminating the requirement for paired data and can be trained without it. The cycleGAN utilizes two generators and two discriminators for style transfer.

Generator CycleGAN did not selected the U-Net architecture as the generator, instead, it chose the generative network suggested by Johnson et al[6] due to its impressive super-resolution performance. Nonetheless, I believe that other advanced generative models, including U-Net, could also be utilized as the generator for cycleGAN. Moreover, if we incorporate recently developed transformer-based generators, cycleGAN could potentially achieve even better performance. The generator used in this project conforms to the standards set by the original paper. It comprises of two downsampling layers and two upsampling layers, with 9 residual blocks for handling the hidden features in between. Instance normalization is used as the normalization technique for the generator (see Figure 3).

Discriminator Due to the advantages of PatchGAN, it has also been chosen as the discriminator for cycleGAN.

Loss function The crucial aspect of cycleGAN lies in the design of its loss function, which comprises of two essential components: the adversarial loss and the cycle consistency loss. The adversarial

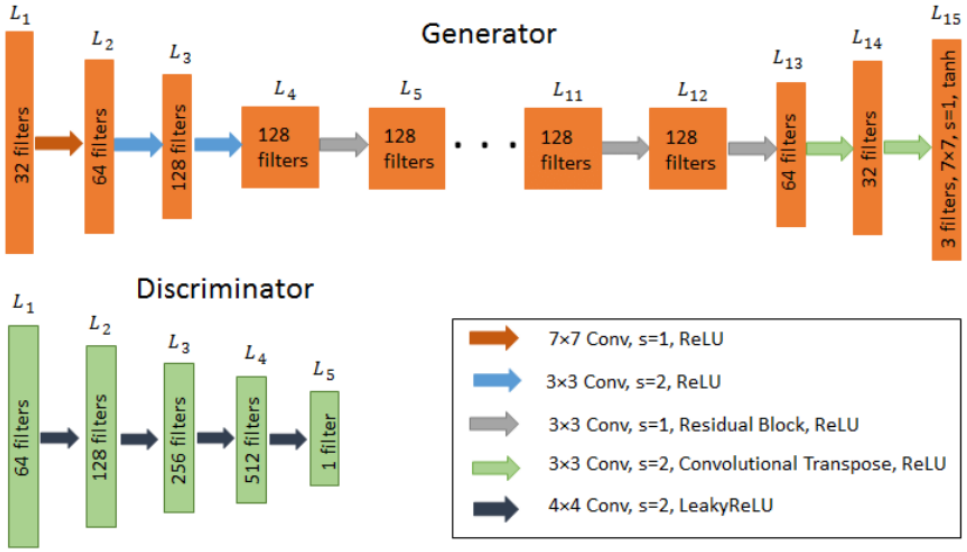

Figure 3: CycleGAN architecture

loss in cycleGAN is similar to the one used in the pix2pix model. Its primary objective is to ensure that the generated image's style is accurate and cannot be distinguished from the target images. In unpaired datasets, a significant challenge is ensuring that the high-level information or content in the images meets our expectations after transformation. Since the generator has too much freedom to generate any content as long as the style is correct, the cycle consistency loss is necessary to restrict the generator's creativity. The equation of cycle consistency loss is shown below:

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}x \sim pdata(x)[F(G(x)) - x_1] + \mathbb{E}y \sim pdata(y)[G(F(y)) - y_1] \quad (2)$$

The cycle consistency term serves as a means to limit the geometric alterations of the original images by requiring the generated image to transform back to the original image. Combining above two

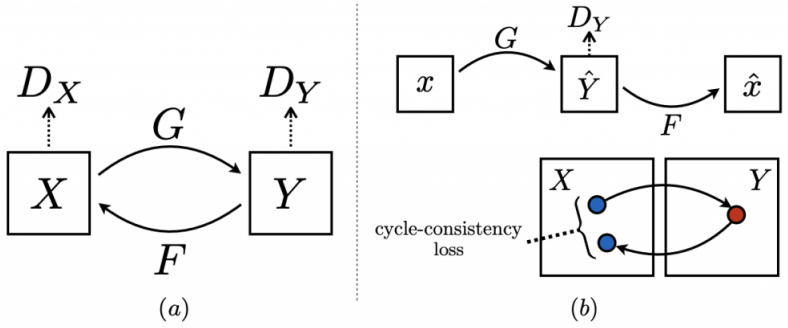

Figure 4: cycle consistency

terms, we can reach our final objective function:
Loss(G,F,D$_X$, D_Y) $= \mathbb{E}y \sim p_{data}(y)[\log D_Y(y)] + \mathbb{E}x \sim pdata(x)[\log(1 - D_Y(G(x)))] + \mathbb{E}x \sim pdata(x)[\log D_X(x)] + \mathbb{E}y \sim pdata(y)[\log(1 - D_X(F(y)))] + \lambda\mathbb{E}x \sim pdata(x)[F(G(x)) - x_1] + \lambda\mathbb{E}y \sim pdata(y)[G(F(y)) - y_1]$
In general, the training process involves the use of two generators where one generator is trained to transform an image from style X to style Y, and the other generator is trained to convert an image from style Y to style X. Additionally, two discriminators are used to ensure that the generated images conform to the desired style.

4.3 CocosNet

The two models mentioned above have shown success in various style transfer tasks, including converting a human face to a comic-style image. Nevertheless, these models do not provide us sufficient control over the output style. For instance, if we train the models to convert segmentations to buildings, they may produce an output image that matches the input segmentation. However, the issue is that we lack control over the style of output buildings. To address the aforementioned issue, I would like to introduce the concept of exemplar-based image translation. The goal of exemplar-based image translation is to convert input images to the output images based on the style given by an exemplar (see Figure 5). The full name of CocosNet is cross-domain correspondence network.

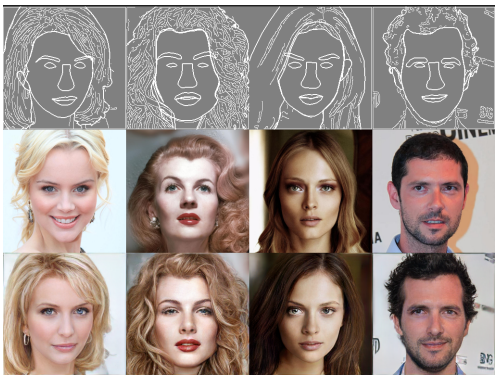

Figure 5: exemplar-based image translation

CocosNet consists of two sub-networks that work together to achieve the desired results. The first sub-network is the Crossdomain Correspondence Network, which is responsible for converting inputs from different domains into an intermediate feature domain where it becomes possible to establish dense correspondence between the inputs. The second sub-network is the Translation Network, which generates the output image progressively with the aid of a warped exemplar that is semantically aligned to the mask or edge keypoints map according to the estimated correspondence. To achieve this, the Translation Network employs a series of spatially-variant de-normalization blocks. Both sub-networks are trained together end-to-end using novel loss functions (see Figure 6). Initially, my plan was to re-implement CocosNet. Unfortunately, I was unable to figure out the preprocessing techniques employed in the paper, and I faced limitations with computational resources that prevented me from training the model. Nevertheless, there are still some valuable ideas from this model that can inspire and be shared.

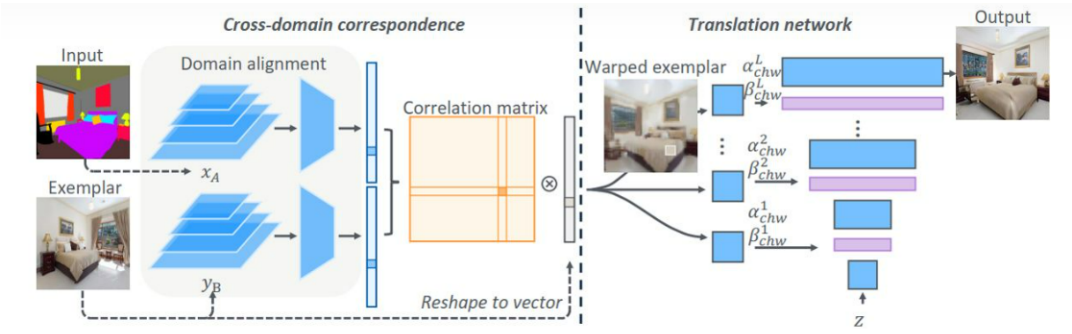

Figure 6: CocosNet architecture

Cross-domain correspondence network One of the innovative aspects of CocosNet is the proposal of a shared domain S that can capture the meaning of both input domains, ensuring reliable semantic correspondence within S. To achieve this, the authors utilized a pyramid network to extract features from both the input and exemplar images, as it has a superior ability to extract multi-scale deep features. However, I believe the pyramid network is not the only option, and any convolution network

with strong feature extraction ability could be used instead. Following feature extraction, the extracted feature maps are converted to representations in S.

$$x_S = F_{A \to S}(x_A) \tag{3}$$

$$y_S = F_{B \to S}(y_B) \tag{4}$$

As depicted in Figure 6 and the above equation, the input image x_A and exemplar y_B are converted to a hidden representation x_S ($x_S \in \mathrm{R}^{HW \times C}$) and y_S ($y_S \in \mathrm{R}^{HW \times C}$) in the S domain. This transformation enables a comparison between the two representations using a similarity measure. To get the correlation matrix, we can follow the following equation:

$$\mathcal{M}(u, v) = \frac{\hat{x}_S(u)^T \hat{y}_S(v)}{||\hat{x}_S(u)||||\hat{y}_S(v)||} \tag{5}$$

where $\hat{x}_S(u)$ and $\hat{y}_S(u) \in \mathrm{R}^C$ represents the channel-wise centralized feature of x_S and y_S in position u and v. Since we do not have the ground truth for the correspondence matrix, we have to train correspondence network with image translation network. We can warp y_B according to M with the following equation:

$$r_{y->x}(u) = \sum_v softmax_v(\alpha \mathcal{M}(u, v)) \cdot y_B(v) \tag{6}$$

where α is the hyperparameter. The image translation network is expected to perform well if the referred warped $r_{y->x}$ is accurate (Refer to Figure 6 for more intuitive and detailed information).

Translation network The main concept behind the translation network is to generate the ultimate output by utilizing the warped exemplar. In this regard, the authors of the paper have taken inspiration from a prior work called SPADE[10]. This is because SPADE blocks have demonstrated exceptional results in transforming label maps into photo-realistic images. While other generators could also be used instead of the proposed network, incorporating SPADE blocks in the architecture has been found to enhance the overall performance significantly.

Loss function From my understanding, the choice of the specific model being used may not be very important because there are always newer and better models available. Instead, what really matters are the goals that we set and the loss function that we use to achieve those goals. In the case of CocosNet, I find the way they designed the correspondence network and the loss functions to be very helpful. They used a paired dataset, where we have input-output pairs x_A, x_B, where x_B is the desired output. The input is just a distorted version of x_B, allowing us to create pseudo exemplar pairs. The main loss term is the feature matching loss, which aims to match the final output with the target output (see below).

$$\mathcal{L}_{feature} = \sum_l \lambda_l ||\phi_l(\mathcal{G}(x_A, x'_B)) - \phi_l(x_B)||_1 \tag{7}$$

In this case, ϕ_l refers to the activation of a specific layer l in the VGG-19 model that was pretrained on a large dataset. λ_l is a hyperparameter that is used to weight the contribution of this particular layer's activation to the feature matching loss. The main difference between this loss term and the L1 loss used in other models like pix2pix and cycleGAN is that it supervises multiple layers of the network, not just the output. Additionally, it is important to ensure that the domain alignment network is performing its intended function properly. Therefore, to achieve this, the loss function presented below is proposed

$$\mathcal{L}^{l_1}_{domain} = ||F_{A->S}(x_A) - F_{B->S}(x_B)||_1 \tag{8}$$

In order to further restrict the network and ensure the accuracy of both high-level information and low-level style, two additional loss terms are proposed: the perceptual loss term and the contextual loss term. These terms are calculated using a pre-trained VGG network to extract various levels of information. Finally,he cycle consistency loss is applied to regulate the learned correspondence (see the equation below) and adversarial loss is adopted to ensure the overall quality of the generated output.

$$\mathcal{L}_{reg} = ||r_{y->x->y} - y_B||_1 \tag{9}$$

Combining all of the 6 loss terms mentioned above, we can get our final objective function:

$$\mathcal{L}_\theta = \min_{\mathcal{F},\mathcal{T},\mathcal{G}} \max_{\mathcal{D}} \lambda_1 \mathcal{L}_{feature} + \lambda_2 \mathcal{L}_{perc} + \lambda_3 \mathcal{L}_{context} + \lambda_4 \mathcal{L}^{\mathcal{G}}_{adv} + \lambda_5 \mathcal{L}^{l_1}_{domain} + \lambda_6 \mathcal{L}_{reg} \tag{10}$$

I have a great appreciation for the method used in this work, which involves transferring the style of a specific region in the exemplar image to the corresponding region in the output image. I find the topic to be very significant and inspiring because it provides a lot of flexibility in controlling the final output. However, I do believe that the loss functions used in this approach are overly complex, and it would be beneficial in the future to simplify them.

5 Experiments

Because of the complexity of the CocosNet model and the limited availability of computational resources, I did not successfully implement this approach. However, I did experiment with the pix2pix and cycleGAN models to achieve comic-style image transfer, and I was able to obtain decent results.

Concerning the implementation of the pix2pix model, I included batch normalization in each convolutional block and established a learning rate of 0.0002 for both the generator and discriminator. In addition, I utilized the ADAM solver with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. To complete this project, I used the gpu52 GPU, which was specifically reserved for this course.

As for the cycleGAN model, I incorporated instance normalization in each convolutional block and selected a learning rate of 0.00005 for both the generator and discriminator. I also utilized the ADAM solver with $\beta_1 = 0.5$ and $\beta_2 = 0.999$.

Figure 7: Validation input of the models

Figure 8: Ground truth

Figure 9: Performance of pix2pix model

Figure 10: Performance of cycleGAN model

Figure 11: Comparison of the performance of pix2pix and cycleGAN models

Performance evaluation From the examples presented, it appears that both the pix2pix and cycleGAN models are capable of achieving style transfer to some extent. However, in comparison to the cycleGAN, the pix2pix model tends to introduce more geometric changes to the input image,

resulting in an output that visually appears closer to the ground truth. While there is a possibility of some artifacts being introduced, the pix2pix model produces results that are visually more similar to the ground truth, as observed qualitatively. Due to the limited size of validation set, quantitative comparisons were not performed, but the qualitative results are sufficiently clear to draw conclusions.

Figure 12: Test input of the models

Figure 13: Performance of pix2pix model

Figure 14: Performance of cycleGAN model

Figure 15: Comparison of the performance of pix2pix and cycleGAN models

Robustness evaluation In spite of the impressive performance of the pix2pix model in the validation set, it can be easily triggered to produce unreasonable output when tested with random images. Conversely, the cycleGAN model appears to produce more stable output. However, as seen in figure 14, there is also an example of failed output with Dr. Strange's image.

Potential Solutions Given that both pix2pix and cycleGAN were proposed several years ago, one possible way to enhance their performance is to replace their generators with more recent ones, such as the translation network utilized by CocosNet. However, to address the issues encountered in both the validation images and testing images, the design of loss functions is critical. To enhance the robustness of pix2pix, it may be beneficial to use a pre-trained VGG-19 model to supervise the output at various levels, allowing the generator to gain a more comprehensive understanding of high-level information. Regarding cycleGAN, the deficiency of geometric alteration could result from the cycle consistency loss, which is the only term regulating the shape of the target image. Since the patchGAN discriminator employed by cycleGAN only concentrates on low-level style and does not contribute to the regulation of geometric shape, incorporating two traditional discriminators to regulate the generated output's shape may be useful. This method may help identify the geometric patterns in the comic-style images. Finally, fine-tuning hyperparameters is always crucial for achieving successful results. Given the time constraints, I am unable to try all the aforementioned methods, however, I am confident that some of these approaches will aid in enhancing the overall performance of the models.

6 Conclusion

In this project, I re-implemented both the pix2pix and cycleGAN models to generate comic-style faces from human faces. I found that both models were able to effectively transfer the style of input images, with the pix2pix model performing better overall in terms of converting images to a comic-style. However, I observed that the performance of the pix2pix model was somewhat unstable when tested with random images sourced from the internet. On the other hand, I noted that the cycleGAN model did not introduce sufficient geometric changes, although it exhibited a more stable performance when tested with random images.

I also identified a limitation common to both models, in that they are restricted in terms of flexibility and can only perform one kind of style transfer with a single dataset. To address this, I introduced the essential concepts behind CocosNet, which allows us to control both the style and content of output images. I found that this model achieved impressive results, but its complexity may present challenges in terms of implementation. Simplifying the model could improve its efficiency and ease of use.

Overall, my project provides insights into the performance of pix2pix and cycleGAN models in generating comic-style faces from human faces and highlights the need for more flexible and efficient models for style transfer tasks.

References

[1] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1511–1520, 2017.

[2] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018.

[3] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[6] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016.

[7] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. *arXiv preprint arXiv:1701.01036*, 2017.

[8] Rui Liu, Yixiao Ge, Ching Lam Choi, Xiaogang Wang, and Hongsheng Li. Divco: Diverse conditional image synthesis via contrastive generative adversarial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16377–16386, 2021.

[9] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[10] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019.

[11] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas 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*, pages 234–241. Springer, 2015.

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[14] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.

[15] Pan Zhang, Bo Zhang, Dong Chen, Lu Yuan, and Fang Wen. Cross-domain correspondence learning for exemplar-based image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5143–5153, 2020.

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