香港城市大學
City University of Hong Kong

Department of Computer Science

BSCCS Final Year Project Report
2021-2022

21CS016

AI-Powered Portrait Photo Enhancer

retouching photos to remove artifacts

(Volume 1 of 1)

Student Name : **WANG Enrui**

Student No. : **55668875**

Programme Code : **BSCEGU4**

Supervisor : **Dr LIAO, Jing**

1st Reader :

2nd Reader :

For Official Use Only

Student Final Year Project Declaration

I have read the project guidelines and I understand the meaning of academic dishonesty, in particular plagiarism and collusion. I hereby declare that the work I submitted for my final year project, entitled:

AI-Powered Portrait Photo Enhancer

does not involve academic dishonesty. I give permission for my final year project work to be electronically scanned and if found to involve academic dishonesty, I am aware of the consequences as stated in the Project Guidelines.

Student Name:	WANG Enrui	Signature:	WANG Enrui
Student ID:	55668875	Date:	March 19, 2022

Table of Contents

Abstract

Portrait photos retouching is popular nowadays. Young people like to share their lives on social media. However, most retouching techniques are not using deep learning methods. We want to find a deep learning model for beautifying facial photos. Recent research showed that the overall effect and accuracy of the AI retouching technique are much higher than the traditional retouching algorithm. The deep learning retouching algorithm is based on a massive amount of data, no longer with standard artificial fixed parameters settings but with adaptive hyperparameters. AI retouching no longer relies on simple traditional image processing algorithms. We can obtain powerful deep learning models if we have enough quantity and excellent quality of portrait images as training data.

In this project, we will use Conditional Generative Adversarial Networks (cGAN) to train our retouching model. I will focus on removing artifacts on the facial skin while my teammate focuses on relighting the images. As we all know, GAN is powerful, but it is weak in high-resolution output photos. We aim to use a new method for high-resolution AI image retouching. As a result, our model is robust in portrait retouching. Using a novel adversarial network, we can use 1MP images as input to generate up to 1024*1024 output photos.

Acknowledgments

Firstly, I wish to show my sincere appreciation to my supervisor, Dr. Jing LIAO. She not only led me in researching for computer vision area but also gave me some valuable comments on my final year project. Without her support, I might not be able to finish this project at today's level. I wish that I could have the opportunity to cooperate with her in the future.

In the meantime, I would like to express my heartfelt gratitude to my teammate YAM Yau Wai. We worked on this project together. With his help, this project became easier to finish. We had a meaningful and joyful time in our final year together.

Finally, I would like to show my thankfulness to my beloved family. During my 4-year bachelor's study in Hong Kong, they always encouraged me and supported me for study. They are my motivation for finishing my bachelor's degree.

1.Introduction

1.1 Background information and motivation

Portrait photos are commonly used in our daily life to record memorable times, such as graduation, birthday, wedding. We can easily use our mobile phones to capture the scene or take selfies for ourselves. However, current mobile phone cameras are not perfect for photographing. They need to reconstruct from a raw sample to get a complete image output. When the sample is taken in bad condition so that it has much noise, the performance of images is often unsatisfying [6].In addition, most users do not specialize in photography, limiting the quality of phone portraits. Overexposure, low contrast, and lousy lighting often affect the quality of photographs. People started to pursue retouching to make their pictures more beautiful.

To improve the users' experience of taking portraits by themselves on mobile phones, a widely accepted solution is to retouch the photos manually, which is time-consuming and inconvenient. Moreover, some retouching applications like PhotoShop also require having some professional skills and training. To improve the quality of portraits in an easier and more advanced way, we will develop a mobile application with a deep learning method to automatically retouch the portraits. All users can use this application to promote their portraits, which benefits both ordinary users and photographers.

There are two special requirements in Portrait Photos Retouching(PPR) [1]: First is Human-Region Priority (HRP), which requires recognizing human-related pixels in a photo and giving them higher priority. With HRP, we can pay more attention to the human region. The second is Group Level Consistent (GLC), which requires a large set of portraits in the same scene and same person, but different in viewpoints, lighting to be adjusted to the same tone.

However, the datasets that satisfy these two requirements are limited, which means that we do not have enough training data to do supervised learning. So we plan to combine supervised learning (e.g., Pix2pix[5]) and unsupervised learning (e.g., CycleGAN[2]) together to train our learning model. We hope that our deep learning model can enhance the users' experience of photographing.

1.2 Aims and scope

This project aims to provide an intuitive way to retouch the portraits. My teammate focuses on enhancing photos taken in bad lighting conditions. I work on removing the artifacts in the facial region in images, such as acne, dark spots, pimples, and skin blemishes.

1.3 Deliverables

Our project will provide a web retouching application to enhance the quality of portraits. When users input the pictures they took, it should automatically generate beautified outputs without artifacts and bad lighting. The output images should look like they were taken by the users, not manufactured by the machine.

2.Literature Review

2.1 Generative adversarial networks (GANs)

Generative adversarial networks (GANs) are widely used for image generation. This method uses a model to generate new images that the discriminative network can not distinguish. After training the generative network and the discriminative network, we can get relatively realistic image outputs.

2.2 Pix2pix

Pix2pix is a supervised learning method to apply the image-to-image translation using Conditional Generative Adversarial Networks (cGANs)[5]. cGANs uses conditional probability to train the model, while unconditional GAN uses other regression methods such as L2 regression. This method is more general by using the cGANs. Its setup is considerably more straightforward than others. We need to provide a training set with paired images for this method to run the supervised learning. However, the training data with paired images is relative hard to get or generate. This method often suffers from a lack of training data.

2.3 CycleGAN

CycleGAN is mainly used for unsupervised training with unpaired images[2]. It also uses GANs but not cGANs for image generation. It combines the cycle-consistency and GANs to perform image-to-image translation. However, compared to Pix2pix, CycleGAN is relatively unstable on the output. The result of this method is far from generally positive. It still can cause failure due to generator issues or the limitations of the training dataset.

2.4 CoGAN

CoGAN[8] uses a tuple of GANs to learn a joint distribution of multi-domain images from data. This method used the idea of a Deep Neural Network to do the learning for a hierarchical feature. This approach can generate a set of novel images, mainly used in movies and games for image transformation. However, this approach can not beautify the photos, and its requirement on the training dataset is more strict.

2.5 Two-way GANs

[6] has developed a photo enhancer with two-way GANs that only need a set of "good" photos as input. They make improvements to avoid instability for traditional GANs to make high-quality results. First, they augment the U-Net to set

global features so that the local features can be better performed. Second, they make an adaptive weighting scheme for WGAN[7]. WGAN is sensitive to weighting, so they adjust the algorithm to control the penalty. Finally, they take different batch normalization layers for input and output for the generator to improve the performance of input data distribution. However, this method can amplify the noise of the pictures when the input image is taken in bad light conditions with lots of noise.

2.6 Conclusion

We noticed a lack of methods that focus on portrait retouching, so we wish to find a model that generates better output images specific to portraits. We plan to base on the Pix2pix to find a new learning method for portrait retouching.

3.Methodology

3.1 System Design

This system is called pix2pixHD, an updated version of pix2pix. Pix2Pix is a condition GAN that can transform images from one domain to another under supervised learning. However, the resolution of the generated image is only 256x256, and it is a challenge to create high-resolution photos using GAN. Our model can generate 2k HD images as output. In this system, we use a coarse-to-fine generator to improve the pix2pix model, together with a multi-scale discriminator and a powerful adversarial learning function. These three improvements will be introduced in detail later.

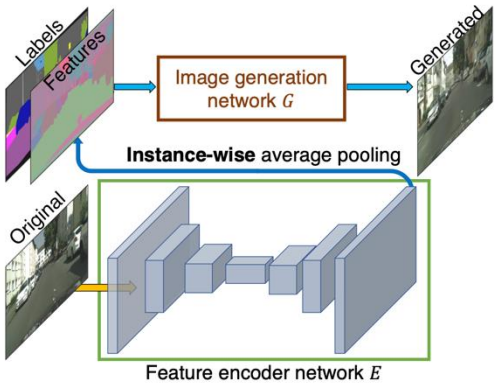


Figure 1

Our system structure can be simplified, as shown in Figure 1.
We will first train the feature encoder network E. Then, we can use this encoder to extract features from the original image data. We will use the labeling information to do instance-wise average pooling to get the Features in Figure 1. If there are enough input images, then the value of each pixel in Features represents the prior distribution of this kind of object. The encoder extracts the features for all input training images. Then k-means clustering is performed to get k cluster centers, with different information such as color and texture represented by k cluster centers. When generating output, in addition to input semantic label information, the system will randomly choose one out of the k clustering centers, i.e., choose a color/texture style. The corresponding image is obtained by selecting features for each instance.

3.2 Major Components

3.2.1 Coarse-to-fine generator

The new generator consists of two sub-network: G1 and G2, where G2 can be further divided into two parts. Figure 2 is the network architecture of the generator.
We first train the residual network G1 on low resolution (256*256). It is an end-to-end UNet structure, which is the same as the pix2pix original design. Then we add a new residual network, G2. The G2 on the left-hand side will extract features for input. Then the right-hand side will combine the features and output of G1.

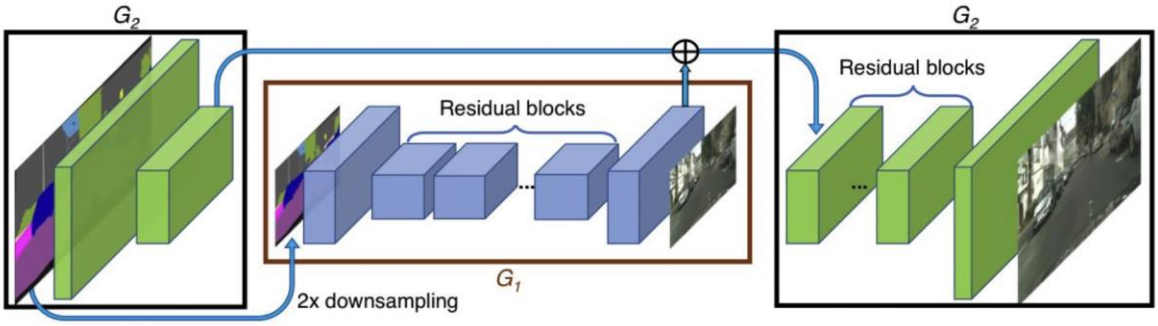


Figure 2

3.2.2 Multiscale discriminators

How to perform discrimination is also challenging for high-resolution images. We build multi-scale discriminators to solve this problem. There are three different discriminators in our system for different scales. We can simply call them D1, D2, and D3. The three discriminators have identical architecture, D1 is for the 100% scale, D2 is 50%, and D3 is for the 25% scale of the original photo. This encourages the generator to achieve a more global consistent output.

$$\min_G \max_{D_1,D_2,D_3} \sum_{k=1,2,3} \mathcal{L}_{\text{GAN}}(G, D_k).$$

3.2.3 Adjusted adversarial loss

We improve the GAN loss based on the discriminator for feature matching loss. The discriminator has different layers. We obtain features from it and then we output the synthesized images. For easy notation, we mark the ith-layer feature extractor of discrimination D_k as $D_k^{(i)}$. The feature matching loss is:

$$\mathcal{L}_{\text{FM}}(G, D_k) = \mathbb{E}_{(\mathbf{s},\mathbf{x})} \sum_{i=1}^{T} \frac{1}{N_i} [||D_k^{(i)}(\mathbf{s}, \mathbf{x}) - D_k^{(i)}(\mathbf{s}, G(\mathbf{s}))||_1],$$

where T is the total number of layers and N_i denotes the number of elements in ea

For the origin GAN loss from pix2pix, it is as follow:

$$\mathbb{E}_{(\mathbf{s},\mathbf{x})}[\log D(\mathbf{s}, \mathbf{x})] + \mathbb{E}_{\mathbf{s}}[\log(1 - D(\mathbf{s}, G(\mathbf{s}))].$$

For the final content loss, we combine the GAN loss and the feature matching loss as:

$$\min_G \left(\left(\max_{D_1,D_2,D_3} \sum_{k=1,2,3} \mathcal{L}_{\text{GAN}}(G, D_k) \right) + \lambda \sum_{k=1,2,3} \mathcal{L}_{\text{FM}}(G, D_k) \right)$$

Where λ controls the weights of two terms. Noted that in feature matching loss $\mathcal{L}_{FM}$, D_k only works as the feature extractor, does not perform any maximization for $\mathcal{L}_{FM}$.

3.3 Training Dataset

Two datasets are used for our experiments. The first one is the PPR10K dataset. The PPR10K dataset is built based on PPR by Liang [1] to fulfill the requirement of retouching portraits, which contains 11161 portraits in 1681 groups, and each group contains around 3~18 portraits. This data set was labeled by three experts with more than five years of working experience using the CameraRaw in PhotoShop. This data is large in scale with high quality.

Another available data set is the Flickr-Faces-HQ-Retouching (FFHQR) Dataset made by A. Shafaei [3]. It was generated from the original FFHQ dataset, which has 70,000 1 MP portraits that were collected by Flickr [4]. It contains portraits that vary in age, lighting conditions, ethnicity, which perfectly meets the requirement of our project.

4.Experiment & Result

4.1 Experiment & Setup

To train this high-resolution model, I need to prepare a powerful GPU. Finally, I chose to use Google CoLab for training and testing. The detailed information will be listed as follows:

- Programming Language: Python 3 (with Jupyter Notebook), HTML5 (for the web application)
- Experiment Environment: Google Colab
- Computing Resources: NVIDIA Tesla P100 (larger than 25G memory)
- Operating System: Linux or macOS (Windows is OK but not recommended)
- Auxiliary Tools: Jupyter Notebook, Visual Studio Code, Chrome Browser
- Important Libraries: Pytorch, Numpy, Dominate

For the **dataset**, I picked 9,000 images out of the 70,000 images for training and 1,000 for testing. I haven't used all of them because the GPU is not powerful enough for large amounts of data. I have used some strengthen methods to make more artifacts on training images for the training data.

For the **baseline**, I want to compare my model with the state-of-the-art algorithms and existing solutions: pix2pix, cycleGAN. I trained my model with the default setting above. For the other model, I used the pre-trained model provided by the author to generate results.

4.2 Result & Comparison

4.2.1 Result Analysis

First, we will compare the performance of our model to the original photos and ground truth. As mentioned above, the ground truth was provided by experts manually. I want to compare with the ground truth to see the performance and compare with the original one to evaluate how many artifacts are removed by the model. From Table 1, we can clearly see that results from the pix2pixHD model are rich in color. Compared to the original inputs, most artifacts are successfully

removed. Overall, the outputs of the model look real in general, and the multi-scale discriminator works so well.

Table 1

4.2.2Result Comparison

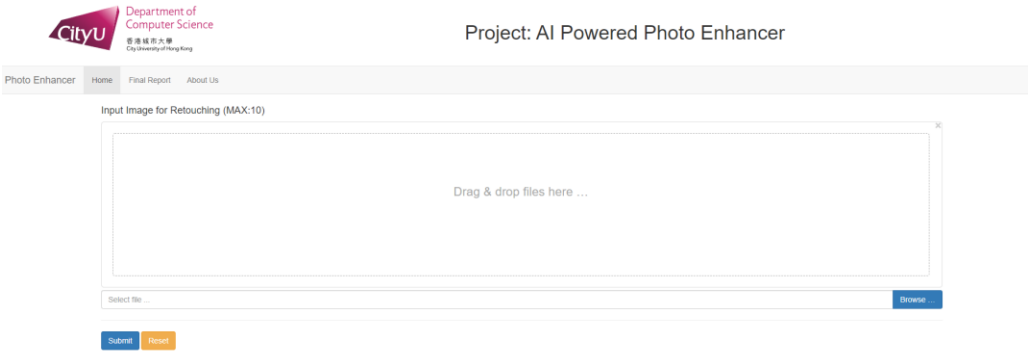
Pix2pix	Pix2pixHD	CycleGAN
		

Table 2

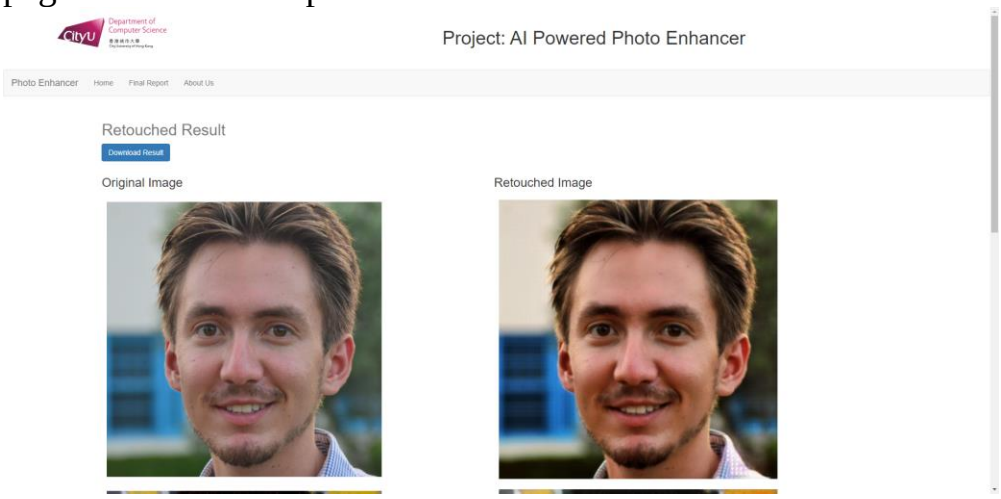
Next, I compare the results with two famous methods in this area. We can directly see that the pix2pix and cycleGAN results are closer to the ground truth, not so colorful as pix2pixHD. What's more, high resolution brings more detail in pix2pixHD. When we look at the first pictures, the cheek of the boys in the HD version is smoother than in others. Pix2pix and cycleGAN can only generate 256*256 output because of their network architecture.
But on the other hand, they are computationally cheap in the algorithm. Pix2pixHD requires much longer training times and high-quality training images.
In conclusion, except for the computing resources, pix2pixHD has advantages in resolution, color, and smoothness.
However, we need to admit that this model will generate some unexpected outputs for some exceptional cases. For example, some minorities like to wear some accessories on their faces, and some accessories similar to artifacts may confuse the model.

4.3 Web Interface

Finally, I put my model into the web interface. People can use the website to the beautified portrait photos for themselves. Here is the demo:
This is the Home page. You can drop or browse the file manager to pick up inputs, then submit them to the model in the backend.



This is the Result page. After you submit the inputs, you will be directed to this page to view the output.



5.Conclusion

5.1 Summary

In this project, I use the pix2pixHD model to remove the artifacts on portrait photos. Pix2pixHD is a deep learning method that can save human resources and make it easier for beginners. At first, I choose the basic pix2pix model, but it can only generate 128, 256, or 512 due to the network of the GAN. So I want to improve the model based on pix2pix. I changed the generative and discriminative networks for higher resolution, but the computational complexity is too high. Then I tried to change the network architecture to pix2pixHD [11] model. I finally generated high-resolution and quality outputs using deep learning based on this model. This model breaks the restriction that GAN cannot generate high-resolution outcomes. Part of the new generator can do super-resolution on the 512*512 result in the central network. Then I put this new model on my local website. I hope that this method can work in an actual application to make our lives easier in the future.

5.2 Future works

In the future, we can find a better training dataset to improve the model in exceptional cases, such as wearing accessories, painting the face, or with a mask. Besides, the network architecture can be further developed to support 2k, 4k, or 8K images. In the meanwhile, I only implement the Pytorch version of this model. In the future, it may be better to support TensorFlow.

References

[1] J. Liang, H. Zeng, M. Cui, X. Xie, and L. Zhang, "PPR10K: A Large-Scale Portrait Photo. Retouching Dataset with Human-Region Mask and Group-Level Consistency," 2021.

[2] Jun-Yan Zhu, Taesung Park, Isola, P., & Efros, A. A. (2017). Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2242–2251. IEEE. https://doi.org/10.1109/ICCV.2017.244

[3] A. Shafaei, J. J. Little, and M. Schmidt, "AutoRetouch: Automatic Professional Face Retouching," in. *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 989–997, doi: 10.1109/WACV48630.2021.00103.

[4] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial. Networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. PP, pp. 1–1, 2020, doi: 10.1109/TPAMI.2020.2970919.

[5] Isola, P., Zhu, J., Zhou, T. and Efros, A., 2016. Image-to-Image Translation with Conditional Adversarial Networks. [online] arXiv.org. Available at: <https://arxiv.org/abs/1611.07004> [Accessed 9 November 2021].

[6] Y. Chen, Y. Wang, M. Kao and Y. Chuang, "Deep Photo Enhancer: Unpaired Learning for Image Enhancement from Photographs with GANs," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6306-6314, doi: 10.1109/CVPR.2018.00660.

[7] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. In arXiv preprint arXiv:1701.07875, 2017.

[8] M.-Y. Liu and O. Tuzel, "Coupled Generative Adversarial Networks," arXiv:1606.07536 [cs], Sep. 2016, Accessed: Nov. 10, 2021. [Online]. Available: http://arxiv.org/abs/1606.07536

[9] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz and B. Catanzaro, "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8798-8807, doi: 10.1109/CVPR.2018.00917.

[10] S. Xu, X. Ye, Y. Wu, F. Giron, J.-L. Leveque, and B. Querleux, "Automatic skin decomposition based on single image," *Computer vision and image understanding*, vol. 110, no. 1, pp. 1–6, 2008, doi: 10.1016/j.cviu.2006.12.002.

[11] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs," 2017.

Appendix

Monthly Log

Time	Activities	Remark
October	Talk with the supervisor about the idea of FYP Write project plan for FYP. Search for existing solutions for related topics	
November	Test existing solutions for related topics Select potentially helpful material for FYP Pick the model for the FYP (pix2pix) Finish Interim 1 report	
December	Start to train the model for this project Find a better loss function for the model	
January	I decided to change my model to pix2pixHD for better performance. Finish Interim 2 report	
February	Try to get a better parameter for the network Prepare the final report.	
March	Continue to train model Develop web-based application Prepare video demo and instruction for GitHub Finish the final report	