Neural Style Transfer for Walter Scott Engravings

Yifan Gong

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2021

Abstract

Style transfer in the field of computer vision refers to preserving the general content of an image while assigning a specific artistic style to it to achieve artistic outcome. Nowadays, style transfer using deep neural network models is very popular, which is called Neural Style Transfer (NST). In this dissertation, we aim to assign ordinary photographs to the style of engravings in the Corson Collection, which illustrate Walter Scott's novels. However, most current research on NST is concentrated on works by famous artists such as Van Gogh, Monet and Cézanne, and there are few researches on the style of engravings, leave alone on Walter Scott's engravings. In this paper, we will study the engravings of Walter Scott and apply both IOB (Image-Optimisation-Based) and MOB (Model-Optimisation-Based) approaches to achieve the effect of style transfer by using CNN and GAN models respectively. And then, we collected a few images that have similar content to the engraving images and transferred them into the stylised images for comparison. We qualitatively evaluate the generated images through comparative analysis and quantitatively evaluate their effect using criteria such as PSNR, SSIM and FID. Finally, we will conclude with the results of our experiments and give suggestions and future improvements for Walter Scott Engravings style.

Acknowledgements

I would like to thank my supervisor, Michael Herrmann, for his great help with this project and for his guidance on the dissertation. I would also like to thank my second supervisor, Scott Renton, who provided me with detailed information on Walter Scott Engravings and further help.

Secondly, I would like to thank China-UK Low Carbon College of Shanghai Jiao Tong University for providing us with an environment to study and live in, where we were able to have a normal study life during the epidemic.

Finally, I would like to I would like to thank my family and my friends for all their help and support in my studies and daily life, and I will never forget this experience of my postgraduate.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Yifan Gong*)

Table of Contents

Chapter 1

Introduction

1.1 Motivation

Walter Scott is a famous novelist, poet, historian and biographer who was born in Edinburgh, Scotland in 1771, and this year is the 250th anniversary birth of Sir Walter Scott. There are a number of fascinating engravings in his novels, which are provided by the Colson Collection at the University of Edinburgh Research Collections. With the help of these high-resolution images, we can study these engravings with the style transfer technology to facilitate the study of Walter Scott.

Style transfer has not only attracted the attention of artists, but has been studied by more and more computer researchers since the mid 1990s. Gatys et al. [9] first applied neural networks for style transfer in 2015, their work demonstrated the superior performance of neural networks for style transfer, following which a growing number of researchers have worked on using neural network for style transfer, producing much contributed work. Style transfer using neural networks has also become known as neural style transfer.

Nowadays, the application of neural style transfer methods in industry has also become widespread [32], based on which many well-known industrial applications have been popular all over the world, such as Prisma, Ostagram, and Deep Forger.

However, neural style transfer is a relatively new direction of research compared to classic computer vision tasks such as object detection, object recognition and object tracking, which have been studied for many years. Moreover, researchers have focused more on the artworks of famous painters such as Van Gogh, Monet or Cézanne, which means that there have not studied on style transfer for engraving style yet. As different artworks have specific styles and features, we need to study the Engraving in order to provide suggestions for generating engraving style images with better quality.

1.2 Objective and Contributions

In this dissertation, we will focus on the performance of two approaches in style transfer for the engraving datasets, namely IOB-based style transfer and MOB-based style transfer. In IOB method, we will use a photo as the content image and an engraving image as the style image to generate a stylised image and seek ways to improve the quality of the generated images, and in MOB method we will generate an engraving stylised image with general dataset style using unpaired image-to-image translation technology. We will also look at how to process the images in the dataset, and how to select the data for training. After these, we will evaluate the performance, advantages and disadvantages of both approaches by comparative analysis and quantitative evaluation using different metrics.

For the two approaches in style transfer, one of our contributions is to validate the effectiveness of IOB style transfer approach for engraving style, and to provide experimental optimisation for engraving style, providing references and suggestions for generating better engraving stylised images. Our another contribution is to verify the feasibility of using CycleGAN to transform the two domains between engraving images and realistic photographs. In addition, since there is no standard metric for the style transfer task, we propose to use several metrics to measure the quality of the style transfer and make suggestions for applying CycleGAN to the engraving style task based on the evaluation results.

1.3 Outline of dissertation

In the first chapter, the motivation, objectives and contributions of our thesis are presented. In the second chapter, we provide a brief overview of relevant knowledge in the dissertation, regarding Style Transfer, GAN and Image-to-Image Translation, where in the style transfer section, we briefly described the work before neural networks and style transfer with neural nets. In the third chapter, we present specific details of the two methods used, IOB and MOB, in a rigorous and lively manner by combining formulas and images. We begin the fourth chapter with a brief demonstration of the collecting and cropping of the dataset, followed by experiments in the IOB method to adjust and analyse the results obtained, and followed by a brief demonstration of the generated results of MOB method, and the we discussed and analysed the generated results through qualitative and quantitative evaluation. Finally, in the last chapter, we concluded the results of the experiment and proposed suggestions for improvements to both approaches and what can be done in recent research.

Chapter 2

Background and Related Work

2.1 Style Transfer

Style Transfer (ST) refers to for a given photo as content image, preserving the general content while assigning stylistic characteristics of another image (called style image), thus generating a stylised image that integrates content and style.

In recent years, improvements in computer hardware, particularly in storage, CPU and GPU performance, have made the application of neural networks possible, which are playing an increasingly important role in computer vision related work. However, research on style transfer started as early as the 1990s without neural networks. Therefore, we will divide our section into two parts, style transfer before neural networks and neural style transfer according to the different approaches to the study of style transfer.

2.1.1 Style Transfer Before Neural Networks

Since the mid-1990s, the study of style transfer has attracted the interest of computer scientists, deriving a new research direction called Non-Photorealistic Rendering (NPR) [54], which established the foundation of computer graphics. The study of stylisation for 2D images or videos [42] is known as Image-Based Artistic Rendering (IB-AR), which is a branch of Artistic Rendering (AR). In this section, we briefly describe these methods, dividing IB-AR-related studies into four sections, Stroke-based rendering (SBR) [19], Region-based rendering, Example-Based Rendering and Image Processing and Filtering, according to the classification criteria of Kyprianidis et al.[29].

The stroke based rendering (SBR) technique was first proposed by Haeberli [16] in 1990, it has been continuously improved and extended to produce stylised images with different brushes, such as oil painting, pastel painting, line drawing etc. Stroke-Based Rendering starts with a realistic photograph, uses the algorithm to extract the key features and construct the parameters of the stroke model to generate the stroke primitive, then determines the position of the primitive based on the features and positioning of the image and draws a stylised image, finally evaluates the result, if it meets the requirements then outputs the stylised image, otherwise draws the the next stroke until it meets the requirements [20]. Finally the rendering produces a non-realistic image with a specific artistic style. However, there is an obvious disadvantage that each specific stroke style needs to be designed separately, which makes it inflexible and requires a lot of effort to design the objective function.

With the development of SBR, there are a growing number of algorithms that segment and parse images and render internal regions independently according to their content, which is called region based rendering technique, by which different areas of the image can be segmented so that different patterns can be rendered for separate areas. Based on this technique, Collomosse & Hall [6] took ordinary photographs and generated cubist-style paintings by geometrically distorting and rendering the images. However, region-based rendering has the same drawback as SBR, which requires designing each style individually.

Example-based rendering (EBR) is based on the image analogy method of Hertzmann et al. [19], which is also mentioned in the following section of image to image translation. EBR aims to learn mapping relationships by supervised learning between paird images and then rendering the desired image, it can be divided into two categories: performing texture and colour transfer, where colour EBR learns mapping relationships in the colour histogram of images to perform tasks such as colouring; texture EBR [7] fills the image by seeking texture patches that are similar to each other in the image, while facing problems like realism and coherence. Zhao & Zhu [59] implemented style transfer for portraits by transferring brush strokes from portrait templates previously drawn by the artist and rendering portraits from photographs. However the inadequacy of the paired training dataset and only use the low-level features of the image limit the performance of EBR technique.

Image processing and filtering is another traditional style transfer technique, but as filters are suitable for the image reversion and restoration rather than for simplification in art makes there are not too many interesting results in this branch. It can be divided into grey scale or colour domain and gradient domain [1] depending on the domain used of algorithms. Winnemöller et al. [55] generated cartoon effects using the difference between bilateral [50] and Gaussian filters [12]. However image processing and filtering techniques did not been used much in the style transfer field.

The aforementioned style transfer algorithms before neural nets achieved some good results in specific areas, however, most of these methods required manual modelling, which means that for a new style, these methods could not be applied directly. Therefore, with the development of style transfer techniques, neural networks have played a significant role and have been improved to address the shortcomings of lack of generalisation.

2.1.2 Neural Style Transfer

Gatys et al.[9] first applied neural networks for style transfer in 2015, their main idea is to exploit the ability of generalised features (high-level semantic features) generated by Convolutional Neural Network (CNN) to independently capture the content and style of images for style transfer task, they built a new feature space based on Gram Matrix statistics on a pre-trained CNN model (VGG19 [47]), which captures image style by including multiple layers of feature correlation. Their work demonstrated the ability of CNN to separate and reorganise the content and style of arbitrary images [23], showing great superiority over traditional methods. Since then, using neural networks for style transfer becomes popular in style transfer task, which is called neural style transfer (NST).

NST can be divided into two procedures, style representation and image reconstruction. Style representation refers to how to extract features from an image, which is the first and most important step in NST [25]. Image reconstruction refers to reconstructing the complete image from the extracted style representation, which is the opposite process of style representation.

The style representation can also be referred to texture modelling, and the methods of texture modelling are mostly based on either statistical summaries or Markov Random Fields (MRFs). Texture Modelling with Summary Statistics approach focuses on modelling textures as N-th order statistics, which was first proposed by Julesz in the 1960s [27], based on this, research such as [18], [41] and others have made outstanding contributions in this area. The style representation in the method proposed by Gatys et al. [10] is also belong to this approach. They use the Gram matrix for extracting styles, which is a second order statistic, more details of the Gram matrix can be found in Methodology section. They used the Gram matrix to compute correlations from feature maps in VGG19 to obtain style statistics, and their experiments demonstrated that using the Gram matrix to extract the style of an image is valid for most textures [8]. However the Gram matrix is not capable to capture texture with long-range symmetric structures. Berger and Memisevic [4] modified the Gram matrix from Gatys et al. by computing the feature map F^l with the transformed feature map $T(F^l)$, which horizontally and vertically translate feature maps rather than computing co-occurrence between multiple features in the map, allowing merging remote structures into image generation and rendering images with various symmetry constraints, and their method is effective in style transfer of seasons of images.

According the study by Li et al [32], style optimisation for the generated image is equivalent to minimising the Maximum Mean Discrepancy (MMD) based on the second-order kernel function between the statistical distributions of the two domains, which means that the procedure of transferring the style to the generated image can be thought as the process of making the second-order statistical distribution of the generated image continuously close to the style image. While style representation is not limited to second-order statistics, other methods such as first-order statistics, polynomial kernel and Gaussian kernel can also be used for NST. Shen et al. [46] introduced meta-learning to style transfer using the Hyper network method in meta-learning, which uses a network to generate parameters for another network [15], dynamically generating parameters for a style transfer network by learning first-order statistics about the distribution of features of the input style image. Jing et al. [24] propose a method for arbitrary style transfer based on the MobileNet, which is a lightweight architecture and introduces the Dynamic Instance Normalization (DIN) module to encode styles as learnable convolutional parameters, combined with a light-weight content encoder for fast style transfer.

Non-parametric texture modelling approach is based on MRFs, which assumes that each pixel of texture in an image is characterised entirely by its spatial neighborhood [25]. The method first divides the style image and the generated image into several patches, and finds and approximates the closest style patch for each patch in the reconstructed result image. Li Wand [30] combine MRF with CNN-based texture modelling methods [8] to reduce the loss of high-level semantic information in content graphs, and their algorithm preserves information such as local structure well.

After obtaining style representation, the next procedure of style transfer is to reconstruct image, which generates the stylised image based on the style features extracted. Depending on the algorithms, we can classify the algorithms for image reconstruction into Image-Optimisation-Based (IOB) approach and Model-Optimisation Based approach [25].

The IOB method extracts features from the given content image and the style image, and then initialise a white noise image as generated image for iteratively optimisation, reducing the error between the generated image and source content and style image to deliver a stylised image with the original content and new style. Further details of the steps will be explained in Methodology Chapter.

The optimisation strategies of the IOB-based style transfer algorithms are mostly similar, their main difference lies in the style representation, where two aforementioned different methods of texture modelling based on summary statistics or MRFS are used. The method of Gatys et al. [10] falls under the IOB method using summary statistics based texture modeling method.

Although the outcomes of IOB based algorithm demonstrate high perceptual quality, an obvious limitation of their algorithm is that it requires high computational resources and takes a long time to process the style transfer task, which is due to the fact that their algorithm optimises the generated image from a white noise image by iteratively minimising the loss. Essentially, this method does not have a model for the style transfer task, the model can be thought of as the generated image, thus this method requires each image generated to complete an iterative optimisation process, which cannot be reused and thus appears to be time consuming and has no generalisation capability. In addition to this, the time taken to generate the image takes more time as the image size becomes larger.

The proposed MOB method solves the problem that the IOB method requires iterative optimization for each generated image. The basic idea is to train and optimize a feedforward neural network for one or more stylized images, and then the model can directly generate the stylized image. Johnson et al. [26] proposed to train a feedforward neural network with advanced feature-aware loss function for image style transfer, which significantly improved the speed of achieving style transfer for real-time tasks by three orders of magnitude over Gatys et al. [10]. Furthermore, their model is better at dealing with image detail and edges and their work also includes high-resolution image style transfer. Crucially, their work makes it possible to output a stylised image after this feedforward neural network has been trained, requiring only the input content image. Similar to Johnson et al., the texture network proposed by Ulyanov et al. [51]

uses a multi-scale generative network for style transfer.

The MOB approach takes significantly less time to generate new stylised images after the model has been trained, making the style transfer model much more reusable and allowing the application of style transfer to be implemented in industry.

2.2 Generative Adversarial Network

Generative Adversarial Networks (GAN) was first proposed in 2014 by Goodfellow et al. [13] and getting more and more popular in related artificial intelligence academia since it was first proposed.

GAN is inspired by the zero-sum game in game theory, which means that the benefits of both sides in the game are summed to zero, with one side gaining exactly what the other side loses [11]. Following this idea, the two sides of GAN game are designed as two players, a generator and a discriminator, which usually have different network structures for different tasks. The generator first receives a random initialised noise and generates a fake but looking real sample through the generative model to trick the discriminator. Conversely, the discriminator uses a discriminative network to distinguish whether the input data is real or fake data that comes from the generator via the discriminant network.

In order to win the game, the generator and the discriminator need to continuously optimise to improve their generative and discriminatory abilities respectively, the discriminator will try to give the maximum possible probability value to the real samples and the minimum possible probability value to the generated samples, and the generator will generate samples as real as possible by the probability of generated samples given by the discriminator and learning to capture the distribution of the real data continuously, this learning and optimisation process is called Minimax game.

Through the adversarial training, the generator would maximize the probability of the discriminator making mistakes until the discriminator cannot distinguish whether the sample is from the real samples or the generated samples, that is, the output probability of the discriminator is 0.5, resulting in a Nash equilibrium, which means that each player's strategy makes to reach the maximum of its desired benefit [3].

2.3 Image-to-Image Translation

Image-to-image translation refers to translating an image from a source domain A to another target domain B while preserving the content of the image [40]. It has a broad range of applications in computer vision research like style transfer [60], [57], [37], image segmentation [56], [14], [31], and image inpainting, [34], [58]. To achieve this goal, we need to train the model to get the mappings in these two domains to generate the translated images. Depending on the data required to train the model, in the following we it into paired (also called supervised) image-to-image translation, and unpaired (alternatively called unsupervised) image-to-image translation.

2.3.1 Paired Image-to-Image Translation

The idea of paired image translation first came from Hertzmann et al. [19] , as they were inspired by work related to texture synthesis at the time [53], [2] and proposed image analogy method based on simple multi-scale autoregression. The idea of their method is to use paired images to train a model, and later using the trained filter in the model on a new target image to achieve similar results as the paired images in the training dataset. Following this, in order to find the semantic correspondences in paired data, Liao et al. [33] proposed to integrate image analogy with neural network, that is, matching and analogy of features extracted from convolutional neural networks, using a coarse-to-fine strategy to compute nearest neighbours to generate results, called deep image category. Their work achieved relatively good results for work such as style transfer, sketch-to-photo, but still failed to construct correct correspondences for scenes that were semantically related but differed significantly in scale and perspective.

Significant progress has been made in CNN-based image-to-image translation, but the methods using CNN require human effort to design an effective loss function, which means that it is labor intensive and error prone. This problem is well addressed by the proposal of generative adversarial networks, which do not require an artificially designed style loss function for minimisation, which can be achieved by adversarial training simply given a higher-level goal. Isola et al. [22] proposed applying GAN to supervised image-to-image translation where supervised indicates that the training data is paired, which is a classic paper known as pix2pix. The pix2pix approach uses a conditional GAN to achieve the image translation task, which can control the kind of the expected image by adding conditional information to the input and learning the mapping from the input image to the output image to get the specified output image. Compared to unconditional GAN [13], it avoids the disadvantage that the generation of images based on random noise is difficult to control.

The pix2pix algorithm is shown in Figure 2.2, where x is the sketch of the object and y is the real image of the object paired with x. After giving the input image x to the generator G, the generator will generate a fake image $G(x)$, and then concat x with $G(x)$ and input it to the discriminator D. The discriminator determines whether the input image is the paired real image and gives the corresponding probability value. The discriminator is trained with true paired images x and y as input, with the goal of giving larger probability values for the true paired images and smaller probability values for the false paired images. While the goal of the generator is to continuously improve $G(x)$ so that D gives the fake paired images with the higher probability value possible, in adversarial training, the generator G and the discriminator D are continuously optimized to accomplish the goal.

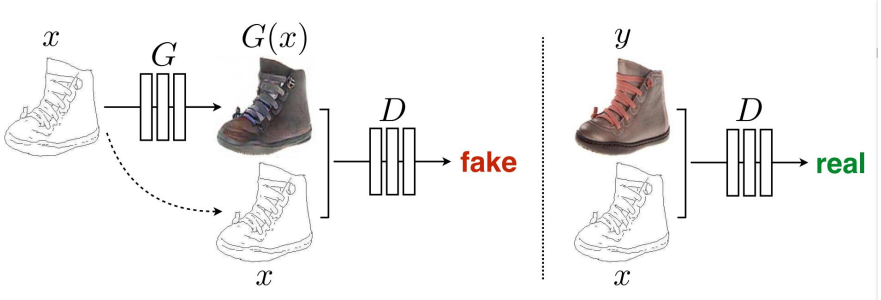


Figure 2.1: ConditionalGAN for Image-to-Image Translation

2.3.2 Unpaired Image-to-Image Translation

Paired image-to-image translation can usually achieve relatively good style transfer results, and if we could somehow find realistic photographs that correspond to the engravings, it would greatly improve our efficiency and enhance the effect of style transfer. However, finding the matching data is extremely difficult especially in this case, as the engravings of Walter Scott do not represent the modern world, and the landscapes and portraits do not exist in the current world, which makes it very difficult to find realistic photographs of these scenes. Moreover, as there are only 1077 images in Walter Scott dataset and some of them are not applicable as they are not engravings, although we find a few similar images in the real world, these are not sufficient to train a style transfer model that would work well, so paired image-to-image translation

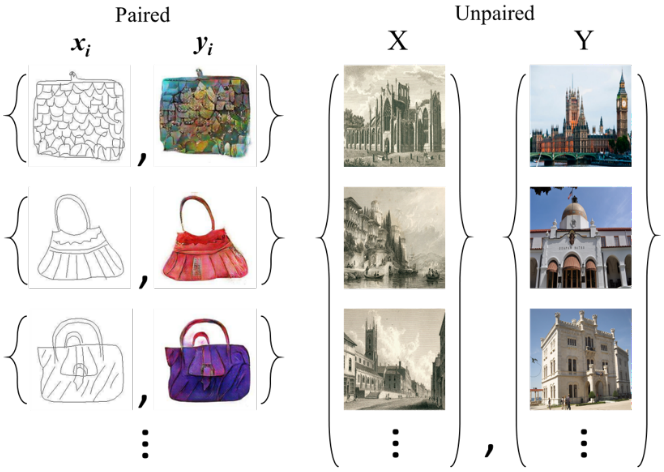


Figure 2.2: Paired and Unpaired Image-to-Image Translation

method is not possible for our style transfer task.

In contrast to paired image-to-image translation, unpaired image-to-image translation approach is a good way for those tasks that do not have paired training data. It refers to mapping between two piles of image styles, which essentially means finding the common content contained in these two different piles of images and the differences in their previous textures, colours etc. In our style transfer task, we have two different style piles of images X and Y, where X is our engraving image dataset and Y is real-world photograph dataset that they do not have a one-to-one correspondence. We wish our style transfer model to learn the general style features from one pile of images to another, such that they can be transformed into each other. In our task, after given a realistic photograph, we expect our model to translate it into the style of engraving, that is, it looks like an image in the engraving dataset, but in fact its content is still that of our given photograph, where only the stylistic elements are changed, this is the key to this image translation task as each pile of images has commonalities and stylistic differences, we want the trained model to be able to find the common points between these two different piles of images and keep these common elements and only transfer these unique elements to each pile to learn the mapping between the two piles of images.

Chapter 3

Methodology

In this chapter, we will introduce and explain the different two approaches used for Walter Scott engravings style transfer. Image-Optimisation-Based method is applied in the first section for the style transfer task, which takes a content image and a style image as inputs. In the second part Model-Optimisation-Based method is applied, which works on style transfer by training a generative neural network that can generate a stylised image only using a normal photograph as input. In the last section, the selection of the engraving dataset images and the realistic photo dataset are given a review.

3.1 IOB Style Transfer

3.1.1 Image Representation

The performance of the VGG19 network for the neural style transfer task was well demonstrated and proven, hence we used the pre-trained VGG19 model for the feature representation of the images.

VGG networks is a deep neural network proposed by Visual Geometry Group of Oxford [47], which are trained for object detection and localisation. It inherits the basic idea of AlexNet, but instead of using the larger convolutional filters of AlexNet, such as those of 5x5, 7x7 and 11x11 sizes, the VGG network uses multiple small 3x3 convolutional filters instead, making it to have a smaller number of parameters and a very simple network structure, which allows it to increase the number of layers and thus improve the performance of the model. The VGG network immediately became the most popular convolutional neural network model of its time due to its simplicity and practicality.

There are two types of VGG networks, VGG16 and VGG19, which only differ in network depth. VGG16 contains 16 hidden layers with 13 convolutional layers and 3 full connected layers, and VGG19 contains 19 hidden layers with 16 convolutional layers and 3 full connected layers, the rest of the structure is the same, and the network structure of VGG is shown in Appendix A.1.

In CNN networks, lower level layer convolutional filters tend to capture more detailed feature representations of the image, while higher level layer convolutional filers tend to capture more overall feature representations of the image, therefore, for the style transfer task, the style features and content features of the image are respectively represented using the lower level and higher level layers in VGG19 .

3.1.2 Content Representation

Content representation refers to the feature representation in the higher levels of the CNN network structure. When a CNN model is trained, its lower-level convolutional filters are sensitive to the exact pixel values of the image as they capture the representation such as edges, textures and other features. As the depth of the neural network increases, the feature representation of the CNN becomes clearer, and the higher-level convolutional kernels are able to capture the global representation of features such as contours and content of the image, while not being sensitive to the exact pixel value of the image content. Therefore, we first generate a white noise as the generative image, which is initially generated randomly, and later reduce the loss between the generative image and the content image by iteratively using the gradient descent algorithm, by calculating the loss at the specified convolution layer with the content image in the VGG network, thus obtaining a similar content feature representation whose exact pixel values are not the same as the content image. Here, we calculate the error between the generated image $\vec{x}$ and content image $\vec{p}$ using squared-error loss function in layer l:

$$L_{\text{content}}\left(\vec{p},\vec{x},l\right)=\frac{1}{2}\sum_{i,j}\left(F_{ij}^{l}-P_{ij}^{l}\right)^{2} \tag{3.1}$$

where F_{ij}^{l} and P_{ij}^{l} denote the activation of the i^{th} filter at position j of specific layer l. With the defined loss function, we can calculate the gradient by using standard error back-propagation and iteratively update the resulting image.

The procedure for the content representation can be illustrated more obviously in Figure 3.1, where the content image F and the generated image P are encoded by the

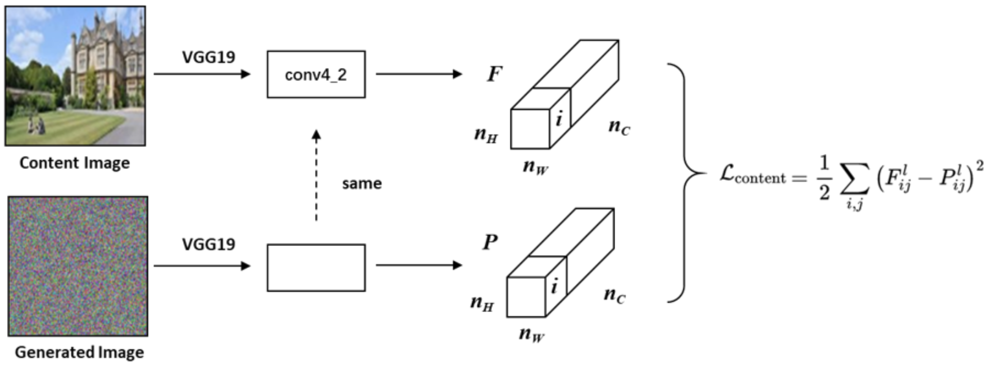

Content Image
Generated Image
VGG19
VGG19
conv4_2
same
F
P
i
i
n_H
n_W
n_C
n_H
n_W
n_C
$\mathcal{L}_{\text{content}} = \frac{1}{2}\sum_{i,j}\left(F_{ij}^l - P_{ij}^l\right)^2$

Figure 3.1: Content Loss Function

pre-trained VGG19 model, and then the content representation is specified in a specific convolutional layer (conv4_2 in the figure, more information about VGG19 network architecture can be found in Figure A.1 in the previous section), and the loss function is computed on the same convolutional layer using squared-error loss function. n_H,n_W,n_C respectively refer to height, width and channel of the filter and i denote the i^{th} filter.

3.1.3 Style Representation

The key to the style transfer task is the representation of style. Gatys et al. [10] suggest that the style of an image can be represented by the correlation between channels, and therefore they propose to use the Gram Matrix for style representation and extraction. The Gram Matrix is a useful way to calculate whether two vectors are linearly correlated in linear algebra, it can be thought of as the eccentric covariance matrix between image features. In a feature map, each variable represents the strength of a particular feature, which is calculated by the convolution of a particular filter at a particular location. As the Gram Matrix calculates correlations between features, allowing it to derive the relationships between different features, such as preferring simultaneous occurrences or preferring one over the other, etc. The diagonal elements of the Gram Matrix are able to reflect the amount of each feature that appears in the image. Due to these advantages, the Gram Matrix has proven to be useful for extracting the overall style of the style image. In our style transfer task, the Gram matrix is computed by calculating the correlation between the original image and the corresponding filter response of the generated image, following the mathematical formulation:

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l \tag{3.2}$$

where i and j represent the i-th the j-th feature map of the layer l, and k is the k-th term of the feature map.

The idea of style reconstruction is similar to that of content representation, where we can use gradient descent to minimize the style loss which is calculated by mean-square distance between the generated image and style image to get similar style representation, a point of difference from content reconstruction is that in the calculation of style differences, the style gap is calculated as the difference in the Gram Matrix, rather than the difference in the image.

In the l-th convolutional layer, there are N_l feature maps and their size is M_l, where M_l is the height h of the feature map multiplied by its width w, we can get the style error E between output image G^l and style image A^l by the mathematical formula:

$$E_l = \frac{1}{4N_l^2M_l^2}\sum_{i,j}\left(G_{ij}^l - A_{ij}^l\right)^2 \tag{3.3}$$

Since the style representation is pretty abstract which cannot be extracted and represented by a specific convolutional layer in the VGG19 network, we can combine different layers at lower levels of the convolutional neural network for style representation and style reconstruction by giving specific weights w_l to the different layers l of the set of selected layers L. The total loss can then be calculated by summing the results of each layer error E_l:

$$L_{\text{style}}\left(\vec{a},\vec{x}\right) = \sum_{l=0}^{L} w_l E_l \tag{3.4}$$

Given the above formulations, we can then compute the derivatives of E for layer l activation and use back propagation to update the resulting image to make it closer to the style image:

$$\frac{\partial E_l}{\partial F_{ij}^l} = \begin{cases} \frac{1}{4N_l^2M_l^2}\left(\left(F^l\right)^{\mathrm{T}}\left(G^l - A^l\right)\right)_{ji} & \text{if } F_{ij}^l > 0 \\ 0 & \text{if } F_{ij}^l < 0 \end{cases} \tag{3.5}$$

Figure 3.2 shows the procedure of style representation more visually. The choice of convolutional layers in the figure follows Gatys et al. by using 'conv1_1', 'conv2_1', 'conv3_1', 'conv4_1' and 'conv5_1' in the pretrained VGG19 model. Depending on the actual style image, the convolutional layer chosen for the style representation will have a great impact on the style transfer results, thus we will discuss and analyse it in the next chapter and choose the appropriate convolutional layer to achieve better style transfer results.

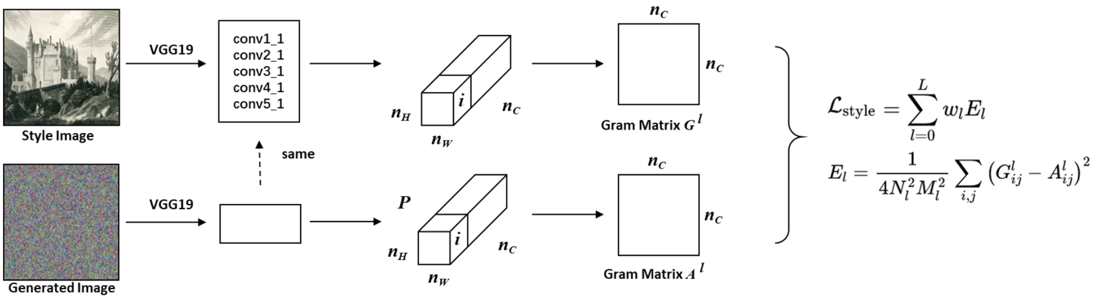


Figure 3.2: Style Loss Function

3.1.4 General Model Structure

As we have defined the content loss and style loss, all that we need to do is to integrate the content loss and style loss and iterative optimise and update the output until it produces the effect as expected. The total loss L_{total} of the generated image $\vec{x}$ with respect to the source content image $\vec{p}$ and the source style image $\vec{a}$ can be obtained by linearly integrate the content loss with weights α and style loss with β respectively:

$$L_{\text{total}} = \alpha L_{\text{content}} (\vec{p}, \vec{x}) + \beta L_{\text{style}} (\vec{a}, \vec{x}) \tag{3.6}$$

The general model architecture is more visually illustrated in Figure 3.3. At first a white noise is initialised randomly, and then the content and style loss are calculated by computing the difference in features respectively in the specific convolutional layer(s) of the VGG19 pre-trained model, assigning them different weights and then summing them together to get the total loss, and finally iteratively updating the generated image

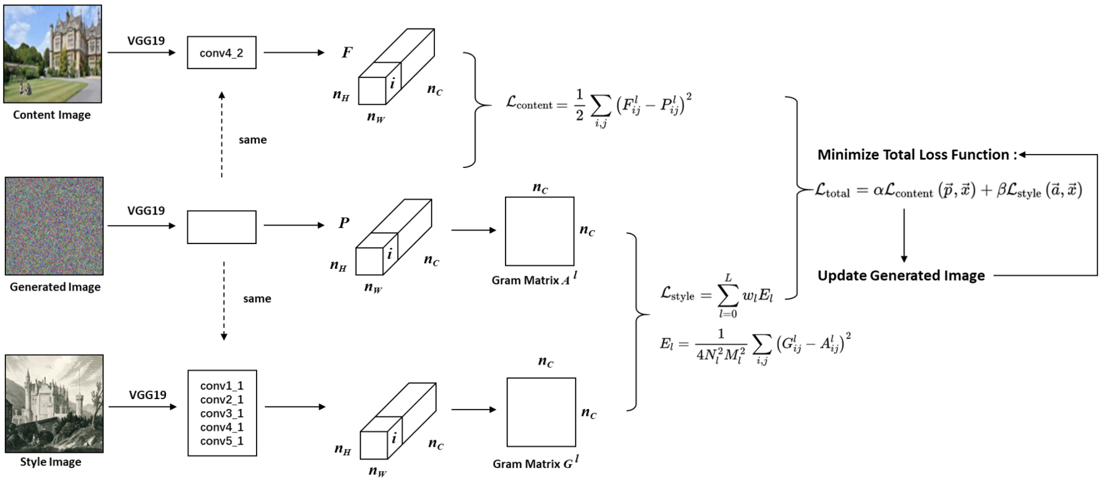

Content Image
VGG19
conv4_2
F
n_H
n_W
n_C
i
same
$\mathcal{L}_{\text{content}} = \frac{1}{2} \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2$
Generated Image
VGG19
P
Gram Matrix A^l
same
Style Image
VGG19
conv1_1
conv2_1
conv3_1
conv4_1
conv5_1
Gram Matrix G^l
$\mathcal{L}_{\text{style}} = \sum_{l=0}^{L} w_l E_l$
$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2$
Minimize Total Loss Function :
$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{\text{content}} (\vec{p}, \vec{x}) + \beta \mathcal{L}_{\text{style}} (\vec{a}, \vec{x})$
Update Generated Image

Figure 3.3: Model Structure

by back-propagation until the generated image has similar features to the content and style image.

3.2 MOB Style Transfer

Although the IOB method is capable to transfer a normal photograph into the style of a particular engraving image, each of the generated stylised images needs to be optimised iteratively, which requires a lot of computational power and a long waiting time, the MOB method mentioned previously solves these problems in the IOB method.
However, there is an issue that a single engraving image cannot represent the whole set of images in the engraving dataset, and the generated stylised image is more like the style of a particular image rather than the style of the concept of engraving. Therefore, we need to build a model for the general engraving image that can extract the style features of the engravings used in the dataset, and then apply this feature for style transfer. Since we only have the engraving data without the realistic images corresponding to these engraving images, we need to use unpaired image-to-image translation techniques, Zhu et al. [60] proposed to add cycle-consistent in GAN for the task without paired data, this network structure is called cycleGAN, which allows images from two domains to be translated into each other with very satisfactory results. Therefore, in the MOB approach, we will use the cycleGAN method proposed by Zhu et al. for style transfer.
Essentially, CycleGAN is comprised of two mirror-symmetric GANs that form a cyclic network, meaning that it contains two generators G_{P2E}, G_{E2P} and two discriminators D_E, D_P, where G_{P2E} represents the generator that converts a photograph into an engraving, G_{E2P} represents the generator that converts an engraving into a photograph and D_E represents the discriminator that discriminates whether the received image is an engraving or not, and D_P represents the discriminator that discriminates whether the received image is a photograph or not.

3.2.1 Cycle Consistency

In order to transfer the style between the images that comes from two domains while ensuring the geometry and spatial relationships of the objects in the image remain unchanged during the style transfer, we need to introduce the concept of cycle consistency to train the model [38].

Cycle consistency is a very important component in cycleGAN, which helps to transfer uncommon style elements between two GANs, while maintaining general content. Cycle consistency draws on the concept in machine translation, when translating a sentence from language X to language Y and then translating back from Y to X, the translated sentence should be the same as the source sentence.

In our task, cycle consistency means that given an image x that belonging to the realistic photo domain, after translating it into the engraving domain $\hat{y}$, and then converting it back into the realistic image $\hat{x}$, $\hat{x}$ should ideally be the same as the original image x. However, it will be impossible even in the field of machine translation to guarantee that a sentence will be identical to the original after it has been translated, let alone image-to-image translation task. The reason for this is that the generator can only try to generate an image that is as realistic as possible based on the features in the image domain, and some of the feature information will be lost in the translation. Although GANs can learn a mapping between two domains, if the network is large enough, they are more likely to learn how to copy elements in thetarget domain to confuse the discriminator rather than transform the image[60], therefore, an additional loss needs to be added to guarantee the cycle consistency of translated image $\hat{x}$ and $\hat{y}$ which are generated by $B(A(x))$ and $A(B(y))$.

$$L_{\text{cyc}}(A,B) = \mathbb{E}_{\text{domain }(x)}[\|B(A(x)) - x\|_1] + \mathbb{E}_{\text{domain }(y)}[\|A(B(y)) - y\|_1] \tag{3.7}$$

The cycle consistency loss reduces the possible set of mappings that the network can learn and forces A and B to perform the opposite transformation, which allows the network to learn more meaningful mappings.

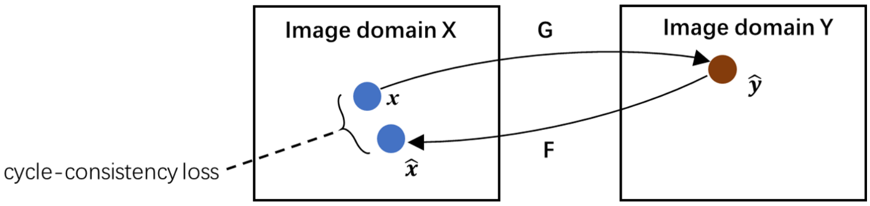


Figure 3.4: Cycle Consistency

3.2.2 Related Formulations

As a derivation of generative adversarial networks, CycleGAN also incorporates the adversarial loss:

$$L_{\text{GAN}}(A,D_Y) = \mathbb{E}_{\text{domain }(x)}[\log D_Y(y)] + \mathbb{E}_{\text{domain }(y)}[\log(1 - D_Y(A(x))] \tag{3.8}$$

where D_Y represents the discriminator that identifies whether an image is from the Y domain, and A is the mapping from domain(x) to domiain(y).
Stability in the training of GAN is a serious problem, one source is the BCE loss used as an adversarial loss, thus the design of the loss function is important. Zhu et al. [60] in their paper mention that in the training process of cycleGAN, the Least Square loss function is used instead of the negative loglikelihood objective. Least Square is an important concept in statistics which is used to minimise the sum of residual squares. The idea is to fit a line to several points so that the distance from these points to the line is the minimum, therefore we can calculate the distance from the point to the line and minimise it. For the GAN task, these points correspond to 1 (true) and 0 (false), thus the adversarial loss in cycleGAN can be written as follows:

$$\begin{aligned} L_{GAN}\left((A, D_Y\right) = \mathbb{E}_{\text{domain }(y)}\left[\left(D_Y(y)-1\right)^2\right] + \mathbb{E}_{\text{domain }(x)}\left[D_Y(A(x))^2\right] \\ + \mathbb{E}_{\text{domain}(x)}\left[\left(D_Y(A(x))-1\right)^2\right] \end{aligned} \tag{3.9}$$

Thus Least Square will only have flat gradient if the prediction is exactly correct, which can helps avoiding vanishing gradients and mode collapse.
While the generator A is trying to generate an image A(x) that is similar to the data domain Y, the discriminator D_Y is also trying to distinguish the generated image from the original image. The target of the adversarial training process can be expressed by the following equation:

$$\min_{A} \max_{D_Y} L_{GAN}\left(A, D_Y\right). \tag{3.10}$$

Since the cycleGAN contains two GANs, the adversarial loss of the second GAN mapping from Y to X is shown in the following equation:

$$\begin{aligned} L_{GAN}\left(B, D_X\right) = \mathbb{E}_{domain(x)}\left[\left(D_X(x)-1\right)^2\right] + \mathbb{E}_{domain}(y)\left[D_X(B(y))^2\right] \\ + \mathbb{E}_{domain}(y)\left[\left(D_X(B(y))-1\right)^2\right]. \end{aligned} \tag{3.11}$$

The second adversarial training target can be expressed by the following equation:

$$\min_{B} \max_{D_X} L_{GAN}\left(B, D_X\right). \tag{3.12}$$

And the general target of cycleGAN can be represent as follows:

$$A^*, B^* = \arg\min_{A,B} \max_{D_X, D_Y} L\left(A, B, D_X, D_Y\right). \tag{3.13}$$

The aforementioned formula for cycle consistency loss is as follows:

$$L_{\text{cyc}}\left(A, B\right) = \mathbb{E}_{\text{domain }(x)}\left[\|B(A(x)) - x\|_1\right] + \mathbb{E}_{\text{domain }(y)}\left[\|A(B(y)) - y\|_1\right] \tag{3.14}$$

Next, summing the loss functions of these two GANs and adding the cyclic consistency loss gives the whole loss function of cycleGAN:

$$L(A, B, D_X, D_Y) = L_{GAN}(A, D_Y right) + L_{GAN}(B, D_X) + \lambda L_{cyc}(A, B) \tag{3.15}$$

where λ controls the relative importance of the two objects [60].

3.2.3 Model Architecture

The discriminator(s) of CycleGAN is based on the PatchGAN proposed by Isola et al. [22] in pix2pix algorithm. PatchGAN is entirely composed of convolutional layers, which can also be called fully convolutional GAN. The input image is first fed into the PatchGAN, and after passing through the convolution layers it is mapped into a $N \times N$ matrix, where each position (true or false) represents a small region of the original image, called a patch, and all responses are averaged to get a discriminant value. This allows PatchGAN to focus on more regions than the original GAN, which only outputs a discriminant value.

The architecture of discriminator is shown in Figure 3.5. The discriminator consists of five convolutional layers layers $Conv1 \sim Conv5$, where $Conv1 \sim Conv4$ are used to extract features and $Conv5$ is used to convert them into a one-dimensional feature vector, and finally gets the discriminated result. Instead of using false images generated directly during training, the training image is randomly selected among the generated images and the 50 most recently generated images for the calculation of the loss function. Adam (Adaptive Moment Estimation) is used for optimisation in training, which dynamically adjusts the learning rate of each parameter using first-order moment estimation and second-order moment estimation of the gradient, with the learning rate having a defined range in each iteration, with advantages such as training smoothness and low memory requirements.

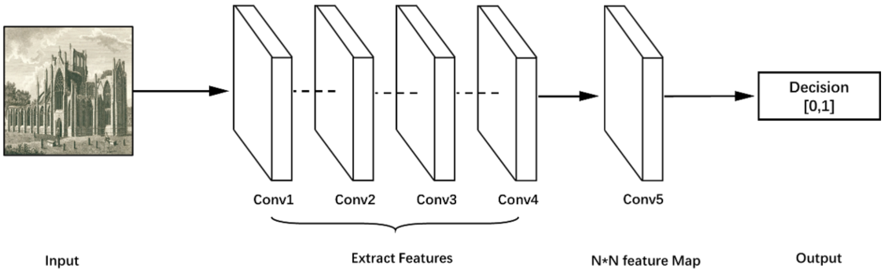


Figure 3.5: Discriminator Architecture

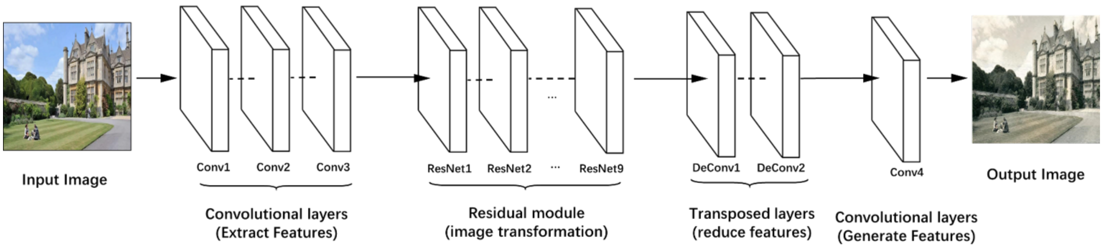


Figure 3.6: Generator Architecture

The generator consists of an encoder, a transformer and a decoder with a total of 4 convolutional layers, 9 residual network layers, and 2 deconvolutional layers. As shown in Figure 3.6, the encoder on the left side of the figure extracts features from the input image through a convolutional layer, which is transformed into a 64*64*256 size. The middle part is called the transformation module, which consists of 9 residual network layers(ResNet1 to ResNet9), which preserves the integrity of the image better and improves the problem of gradient disappearance in the deep network. The output features of the encoder are transformed from X-domain features to Y-domain features by the transform module. The decoder module is on the right side of the figure, it uses the deconvolution layer to restore the low-level features of the image, and finally get an image that matches the Y-domain features.

The complete model of CycleGAN is shown in Figure 3.7. In the mapping of photos to engravings, an image is first input, G_{P2E} transforms it into a generated engraving image, which is then passed to the discriminator D_E to determine whether the generated image is from the same style as the engraving. Then G_{E2P} reconverts the generated engraving into the style of the real image to obtain the Regenerated photo, and the discriminator D_P determines whether the Regenerated photo is from the photo domain. A comparison is made between the input photo and the Regenerated photo, which is used to calculate the cyclic consistency loss. The mapping of engraving to the photo in the other direction is its mirror, and will not be expanded upon here.

3.3 Dataset Overview

3.3.1 Engraving dataset

In this dissertation, we use the Walter Scott Image Collection as our engraving dataset, which is primarily based on the images and materials contained in the Corson Collection at the University of Edinburgh Library. The Walter Scott Image Collection

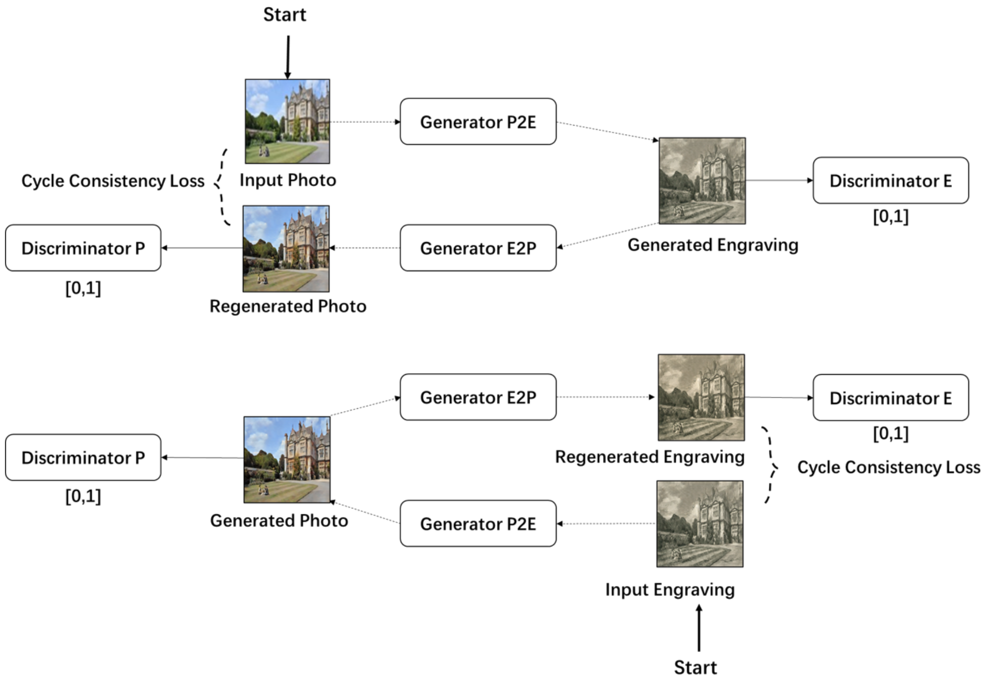
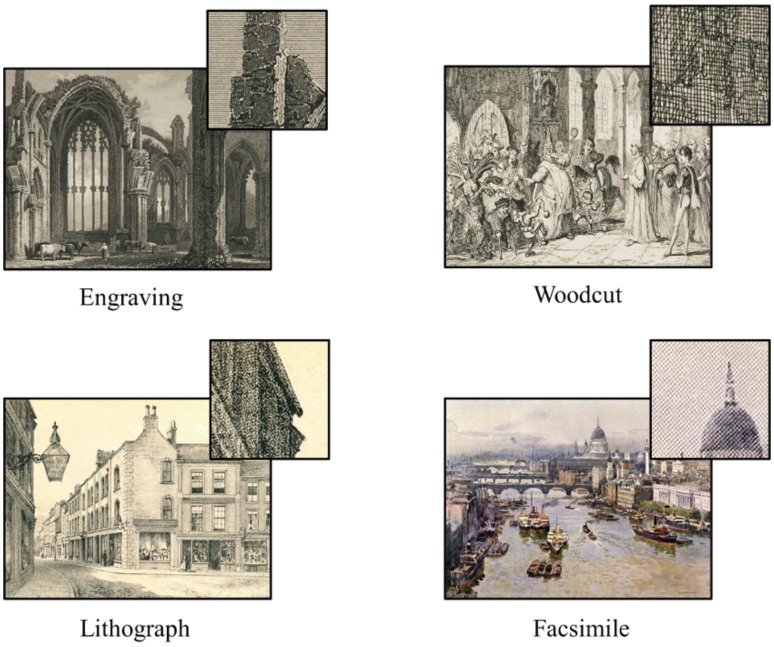


Figure 3.7: CycleGAN Architecture

contains a total of 1077 images of engravings, etchings, lithographs, oil and watercolour paintings, drawings, and photographs in a variety of styles.

Although we have many different categories of data, we need to select a specific style for style transfer. The label categories in the Walter Scott Image Collection image information list include Engraving, Woodcut, Lithograph, Facsimile, Drawing, Painting, Manuscript, and many others. However, there are too many labels for a dataset of only 1077 images, therefore we have classified the images in the dataset into five categories based on their labels combined with subjective judgement of detail texture, including Engraving, Woodcut, Lithograph, Facsimile and the rest (some images are not classified into corresponding categories based on their labels), as shown in Figure 3.8. We chose Engraving as our dataset, which has the most images.

In addition, we divided the engravings into two parts, landscape and portrait according to their content. Since most style transfer algorithms do not have a specific mechanism for the specific content, to ensure the effect of style transfer, we chose as the dataset engravings with architectural content, which contains approximately 400 images.



Engraving

Woodcut

Lithograph

Facsimile

Figure 3.8: Dataset Overview

3.3.2 Realistic Photos Dataset

We chose the Google-Landmarks Dataset [39] as our realistic photo dataset for comparison with the engravings dataset which features landscapes and architecture based on our requirement for unpaired style transfer task. The Google-Landmarks Dataset contains famous (and not-so-famous) landmarks, which contains over two million images depicting 30,000 unique landmarks from around the world, with different viewpoints, weather and lighting, and further information can be found in Kaggle [1]. However two million is too much for our style transfer task as we donnot have to many engraving images, therefore we only selected around a thousand of these photos as our realistic photos dataset. in order to allow our model to learn more, we manually removed some of the photos from the dataset, including modern architecture, night images and content that are too different from those in the engravings. In Appendix A.2, we provide some examples of images in this dataset. For the Adam optimizer in the IOB-based style transfer, we also found that increasing the number of iterations and early stop did not improve the generated results obviously, although there was a reduction in the value of the content and style loss.

[1] `https://www.kaggle.com/google/google-landmarks-dataset`

Chapter 4

Experiments and Results

Our experiments are mainly based on Pytorch, we use Google's Colab platform for experiments on style transfer, in addition we subscribe to Colab Pro for better computational resources. In our experiments, we used ' Tesla P100-PCIE-16GB' graphics card from Colab, with NVIDIA-SMI version 460.32.03 and CUDA version 11.2, which has 16GB of memory and high computational power.

4.1 Data Processing

4.1.1 Dataset Images Downloading

Based on the list of image information provided by Digital Library Development Systems of the University of Edinburgh, we can access these images based on their URLs to build our dataset. Our initial approach is to go to the Walter Scott Image Collection webpage and find the image url by analysing the HTML code that makes up the webpage, for example by opening the webpage of an image and searching for the HTML code that contains the '.jpg' element, we can get the following HTML code that contains '.jpg' element:

```
<meta property = "og:image" content="https://images.is.ed.ac.uk/
MediaManager/srvr?mediafile=/Size4/UoEwal-1-NA/1001/0030028d.jpg">
```

and we can easily find the image url: `https://images.is.ed.ac.uk/MediaManager/srvr?mediafile=/Size4/UoEwal-1-NA/1001/0030028d.jpg`, where 0030028d is the ID of the image, so we are able to download the image by entering the ID of the image. However, there are some images whose links do not conform to this rule, which require us to look for new patterns in these images, making downloading the engraving dataset a tedious task.

Thanks to the guidance of Digital Library Development Systems of the University of Edinburgh, there is a simpler way to access the original images by using the International Image Interoperability Framework (IIIF) [48] of the University of Edinburgh to download images. Following the guidelines, we can obtain the original engraving links by modifying the given URL in the information list. For example, for the URL `https://images.is.ed.ac.uk//luna/servlet/detail/UoEwal~1~1~58112~100356`, replacing the 'detail' item to the 'iiif' item and add '/full/full/0/default.jpg' to the suffix, we will get the URL of the original image, which is `https://images.is.ed.ac.uk//luna/servlet/iiif/UoEwal~1~1~58112~100356/full/full/0/default.jpg`.

As each image download procedure is time-consuming, which is network-intensive IO and disk-intensive IO that involves a lot of waiting time. In order to speed up the downloads of images, a multi-threaded script was developed. The script first puts the processed source image URL into a queue, and then constructs several sub-threads to read the image URL from the queue and call the download function, so that we can finish downloading the images in the dataset in less than a minute.

4.1.2 Images Cropping

As the original images(on the left of Figure 4.1) contain not only the content of the engravings but also a border of the image with information about the author, content , etc. In order to get better outcomes for style transfer task, we need to crop the source image to get the content(on the right of Figure 4.1) . However, cropping by hand is very time consuming and tedious, hence we chose to use OpenCV to crop the original image, which is capable of many traditional computer vision tasks. We can divide the automatic cropping algorithm into the following four steps:

1. Setting the threshold. It can be easily noticed that most of the engravings are rectangular and there is a clear difference between the content area and the border area of the background. We can thus set a threshold to initially separate the content area from the border. The image is first transformed into a grey-scale image and then smoothed using the median filter, which sets the grey-scale value of each pixel point to the median of the grey-scale values in a certain neighbourhood window pixels at that point, and then transforms the image into a binary image according to the set threshold.

2. Erosion and dilation. As there is no fixed value for the content and background

of the image, the image we separate using the threshold has a lot of speckles. Thus we try to remove these spots by repeating erosion and dilation operations, where the erosion operation will erode the white pixels in the image to remove the small spots and the expansion operation will expand the remaining white pixels and grow them back to avoid removing the content of the image.

3. Edge detection and contouring. We use the Canny function in OpenCV for edge detection and then extract the contours. Although there are less spots in the processed image, this does not mean that we will get only one contour, so we use the contourArea function to calculate the area of the contours and sort them to select the largest contour, and then we use the approxPolyDP function to get a strict rectangle.

4. Get coordinates to crop the image. We use the convexHull function of OpenCV to detect the Convex Hulls of the maximum contour and then obtain the coordinates of the rectangle vertices and crop the image by calculating the position.

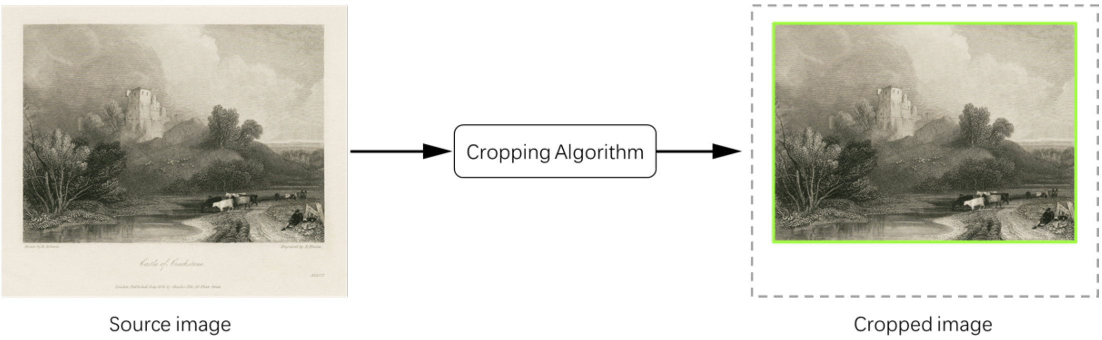


Figure 4.1: Cropping Process

After application, our cropping algorithm is effective in cropping most of the images where the content and background are distinct, but it has limitations in cropping some images where the content is integrated into the background.

4.2 IOB Style Transfer

In our IOB based style transfer experiments, we reproduced the work of Gatys et al. [10] by referring to the code in the pytorch tutorial. Following the results of their experiments, we choose ’conv1_1’, ’conv2_1’, ’conv3_1’, ’conv4_1’ and ’conv5_1’ layers

from VGG19 for the extraction of style features and 'conv5_1' for the content representation (specific information about the network structure can be found in the previous section) and set $\beta/\alpha = 1e^3(1000)$ for content weight and style weight.
For initialisation, we choose different types of images for iterative optimisation, a randomly initialised white noise and the content image, which is chosen because it may be able to preserve the content information better and reduce the image optimisation process. The generated results are shown in Figure 4.2, where the second image on the right is initialised with white noise and the first image on the right is initialised with a content image.

From the figure we can observe that the image starting from white noise does not achieve a good result, it only retains the contours of the content image but the style differs significantly from the style image. Whereas the image initialised from the content image retains most of the content information and is relatively close to the style image, although the effect of style is not obvious. We find from the results that changing the ratio of α and β has a greater effect on the generated stylised image, and by increasing the weights, the tone of the stylised image can be made more towards the tone of the engraving. However, this does not mean that increasing the weight of the style will always improve the quality of the generated image, as we can observe from the two

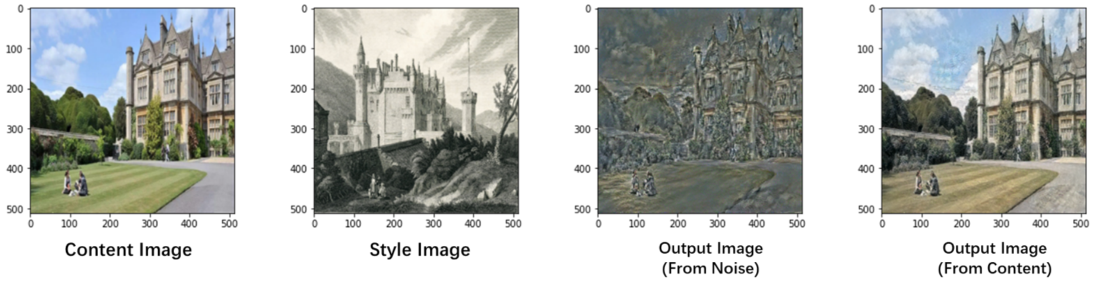


Figure 4.2: Comparison of the generation for different initialised images

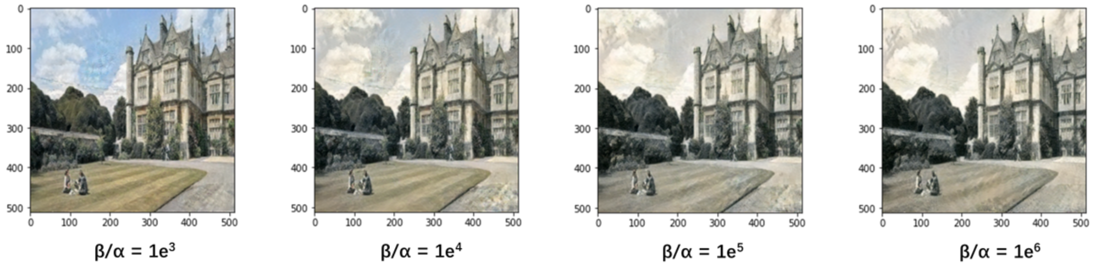
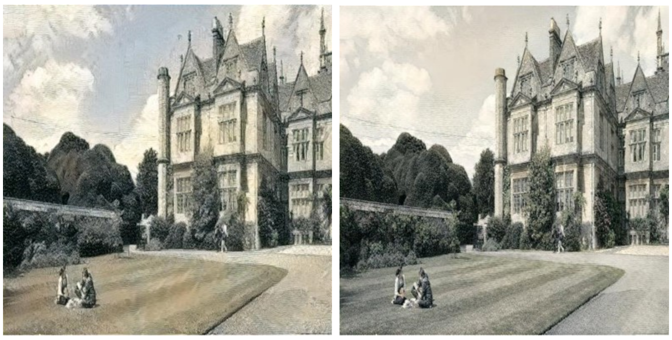


Figure 4.3: Comparison of the generation for different ratios

generated results on Figure 4.3 that excessive weights can have an effect on destroying the overall structure of the image and thus blurring parts of the image. Smaller weight ratios on the other hand do not have a significant impact on the generated image initialised with the content image. Therefore, after experimentation, we consider that a ratio of β and α up to $1e^4$(1000) and $1e^5$(10000) will produce better results.

Although the range of β/α was determined, the generated image was only more similar to the engraving style image in terms of image tone, but still differed considerably in terms of image detail. Thus we then try different style represent layers to achieve a better effect. Although the results of Gatys et al. [10] are effective in many style transfer tasks, their choice of convolutional layers may not give good results for the engraving style, where the textures are more towards the pixel level whereas the ’conv4_1’and ’conv5_1’ layers in the VGG network refer to higher level convolutional layers that are difficult to capture at lower levels and thus cannot produce a better result. Therefore, we chose to use lower level convolutional layers like ’conv1_1’, ’conv1_2’, ’con2_1’, ’conv2_2’ and ’conv3_1’ in VGG network for style layer representation in order to be more similar to engraving style.



(a) Results with Higher Layers (b) Results with Lower Layers

Figure 4.4: Results for different style represent layers

Furthermore, as our generated images are initialised with the content image, which means that we do not need to optimise the content loss too much, instead of using the high level convolutional layer ’conv5_1’ for the representation of content features, which would make the content image abstract by negative optimisation and not match the clearer content in the engraving style. In experiment, we found that using the ’conv3_1’ layer gives better results. With all other parameters identical, the results of our choice of new content and style representation layers (Figure 4.4(b)) and the results of using the Gatys et al. representation layers (Figure 4.4(a)).

The mean and standard deviation of images provide a good overview of the information and features of the images, where the mean is the average of a set of data, and the standard deviation represents the dispersion of the data. And by using these two data for normalisation, the gradient acts on each image equally, which means there is no proportional mismatch, avoiding the inability of the gradient operation to take into account the downward trend of different features in different dimensions at different levels, which makes the loss oscillate. When dealing with realistic images in general, we generally use calculations based on those derived from millions of images in ImageNet. However, there are large differences between prints and photographs, and for better style transfer results, we calculated the standard deviation and mean values for the Engraving dataset, which are shown in Table 4.3:

where the values are in [R, G, B] format for the red, green and blue channels.

	Mean	Standard Deviation
ImageNet	[0.485, 0.456, 0.406]	[0.229, 0.224, 0.225]
Engraving	[0.489, 0.480, 0.417]	[0.190, 0.178, 0.169]

Table 4.1: Mean and standard deviation for different dataset

By normalising the mean and standard deviation calculated in our engraving dataset, the generated result is significantly improved. As shown in Figure 4.6, it can be noticed that the stylised image generated by using the standards parameters for normalisation has a distinctly prominent patch in the image with unusual colours and no texture, while the stylised result of the normalisation calculated using the engraving data is not only more consistent in overall colour, close to the colour of the engraving, but the corresponding graphic block has the same texture as the stylised image. The choice

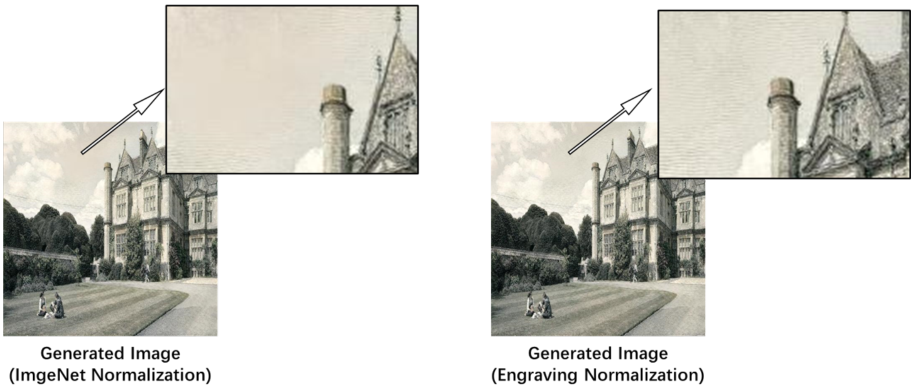
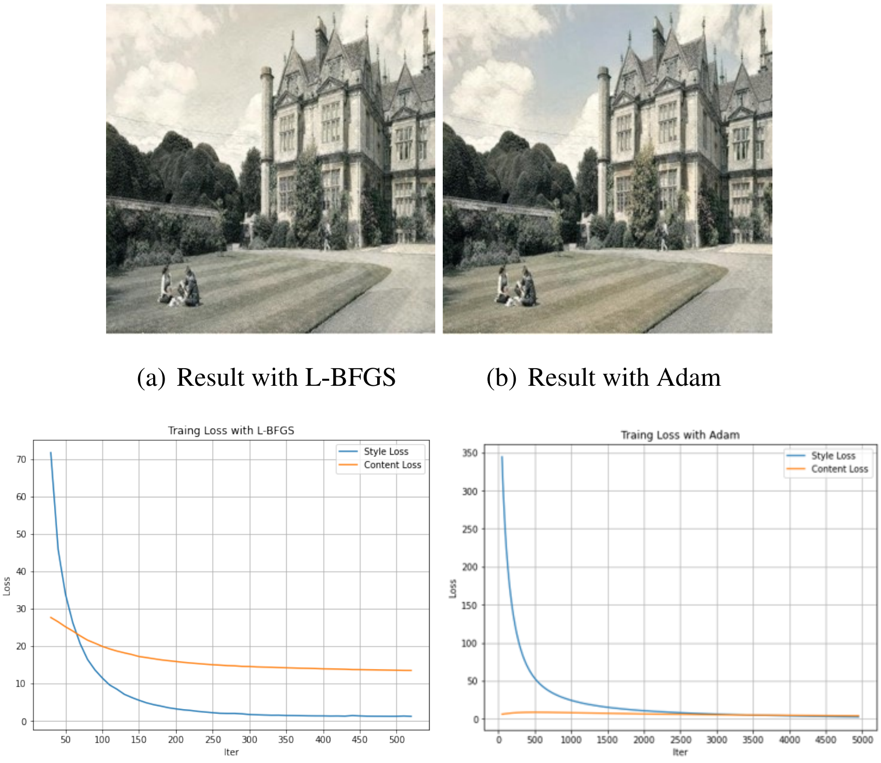


Figure 4.5: Comparison of the generation for different normalization

of optimizer also greatly affects the effect of style transfer. Many papers have shown that using L-BFGS gives better outcomes [10] [43] [36], while Adam is more stable and less memory intensive when trained with a larger style weight [28], hence we tried to use L-BFGS and Adam as optimizers for style transfer respectively. Maintaining the other parameters as constant (except for iteration times), the results are shown in Figure 4.6. We can find that using L-BFGS as the optimiser enables the content loss and style loss of the image to drop quickly, while Adam as a comparison, its style loss quarter drops very slowly and the generated image does not differ much from the content image (as we choose to optimise the content image as the initial image rather than a white noise image). In order to better check the effect of using Adam, we tried to keep increasing the iteration times, and the image generated after 5000 iterations is shown in Figure 4.6(b), compared to the result after 500 iterations using L-BFGS as the optimiser. This shows that using L-BFGS for 500 iterations provides a better result than using Adam for 5000 iterations. We have also tried setting a larger learning rate for Adam to speed up the training, whereas using larger values of the learning rate can indeed enhance the texture, but tends to over suppress the content. The generated



(a) Result with L-BFGS (b) Result with Adam

(c) Loss of LBFGS (d) Loss of Adam

Figure 4.6: Comparison of the results and loss for different optimiser

images for different learning rates using Adam as the optimizer are shown in Appendix B.1.

As a very important factor in style transfer, iteration times have an important impact and too many iterations will greatly reduce the efficiency, therefore, based on the experimental results, we choose to use L-BFGS as the optimizer for better results and faster training time.

Meanwhile, there are limitations in our IOB-based results. Although the optical and informational result is acceptable as ”engraving-like”, the results are sometimes blurred-grey but not finely chiselled engraving lines. One possible reason for this is that in order to minimise computational power consumption, we chose to pre-process the image to transform it into the size of 512×512, which causes the detailed textures to be blurred and and lost as the image is compressed, and therefore our stylised image does not produce finely engraved lines in the details of the texture. Another possible reason is that we chose to initialize the generated image with the content image instead of the white noise image in order to make it more coherent, which largely reduces the times of optimization and therefore the optimisation of detailed textures is not sufficient.

Therefore, we attempted to use high-resolution images for style transfer rather than compressing image quality to reduce the optimisation process. It should be noted that generating stylised images with high resolution suffers from training instability, thus we reduced the times of optimisation to reduce the probability of this occurrence, and the generated results can be found in Appendix B.2. Based on the results we can observe that the aforementioned experimental parameters are still valid for high-resolution image style migration and that the generated textures are more similar to the finely chiselled engraving lines. However, the speed of the generated results is significantly reduced compared to the image with 512×512 size.

In our experiments of IOB based style transfer, we experimented with different initialised images, style and content weights, iteration times, the effect of normalisation , optimisers and other features on the effect of style transfer that need to be selected manually, analysing and discussing these results in relation to the relevant features of the engravings, and from this our IOB based method is able to generate better results.

4.3 MOB Style Transfer

In the MOB-based style transfer, we use the cycleGAN approach since we do not have paired photographs corresponding to Walter Scott Engravings whereas cycleGAN is capable of unpaired image style transfer. We need to prepare two domain datasets, one is the Google-Landmarks Dataset photograph dataset and the other is the Engraving dataset, which crops down the content by previous method (without cropping cycleGAN would capture border feature feature and the generated images would carry straight lines).

As cycleGAN training consumes extensive time while the parameters and structures involved cannot be adjusted incrementally based on the results as in the IOB method, we chose default parameters to train the network, where the learning rate was set to $2e^{-4}$(0.0002) and the learning rate was kept constant for the first hundred training rounds and dynamically adjusted for the second hundred rounds. The generative adversarial network was set to lsgan, which uses Least Square as the loss function.

Figure 4.7 shows some examples of transforming photographs into engravings. We can see that after 200 training epochs, the generator is able to nicely transform photographs into engravings, which are very similar to those in terms of colour. Since cycleGAN contains two mirror-structured GANs, there is another generator that we can use to transform the engraving into a photograph, which is generally referred to as image restoring. We can see that our model does a good job of removing specific textures and tones from the engravings and restoring the colours of the sky, trees and grass. Some examples of restoring engravings to photographs are shown in Figure 4.8: The training loss of the generators, discriminators and cycle consistency from both engravings and photographs sides are shown in Appendix B.3. Where G_{E2P} represents the generator

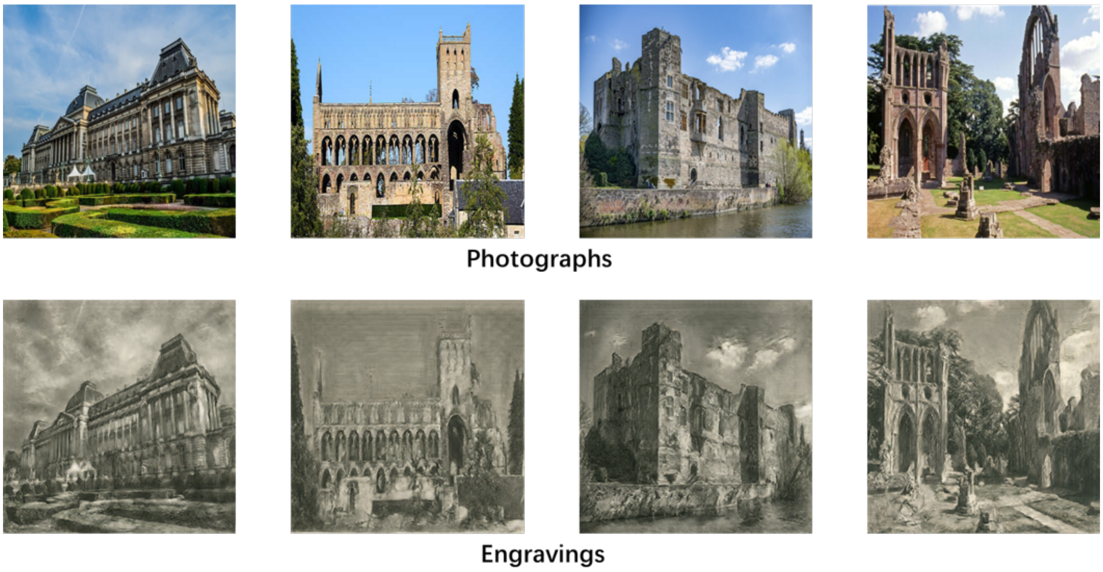


Figure 4.7: Results of photographs to engravings

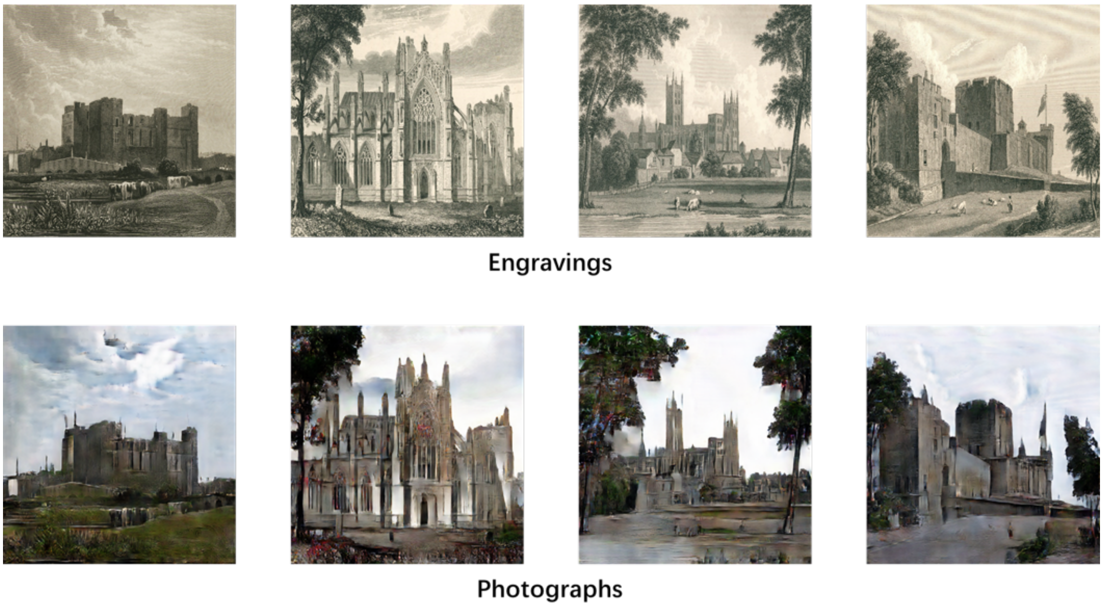


Figure 4.8: Results of engravings to photographs

that transforms engravings into photographs and G_{P2E} is the opposite. D_P represents the discriminator that determines whether an image is photograph and D_E represents the discriminator that determines whether it is an engraving.

From these loss changes, we can observe that the losses of both generators and discriminators sides are continuously changing and fluctuating, which is a game-playing process, and the fluctuation of their losses is becoming smaller, which means that their ability to capture features is improving. In addition, we can see that the cycle consistency losses are converging, which imply that our model is improving for the interconversion between the two domains, thus we validate the two cycles : photo to engraving and back to photo (as shown in Appendix B.4), and engraving to photo and then back to engraving (Appendix B.5). We can notice form the results that cycleGAN is able to retain the information well for this cycle process, and the difference between the transformed image and the original image is very small.

4.4 Comparison, Evaluation and Discussion

Based on the location information contained in the Walter Scott Engravings dataset, we collected a few photographs that are similar to the content of the engravings for evaluation. We use the collected photographs as our content images and the corresponding engravings as our style images to generate stylised images based on the IOB approach, and we also use the photographs as realistic images to generate engraving style images through the MOB approach, and we will then explore the advantages and disadvantages of these two different images through qualitative and quantitative analysis.

The generated example is shown in Figure 4.9, where the first column is the realistic photographs similar to those in the engravings, the 1nd column is the engravings, the 3rd column is the result of the IOB approach and the fourth column is the result of the MOB approach.

By comparing the results, it is easy to find that the style of the IOB results are more similar to that of the corresponding engravings with similar content, as they are optimised using the corresponding engravings as the style images. Furthermore, as our IOB method uses the content image as the initial image for optimisation, it retains the

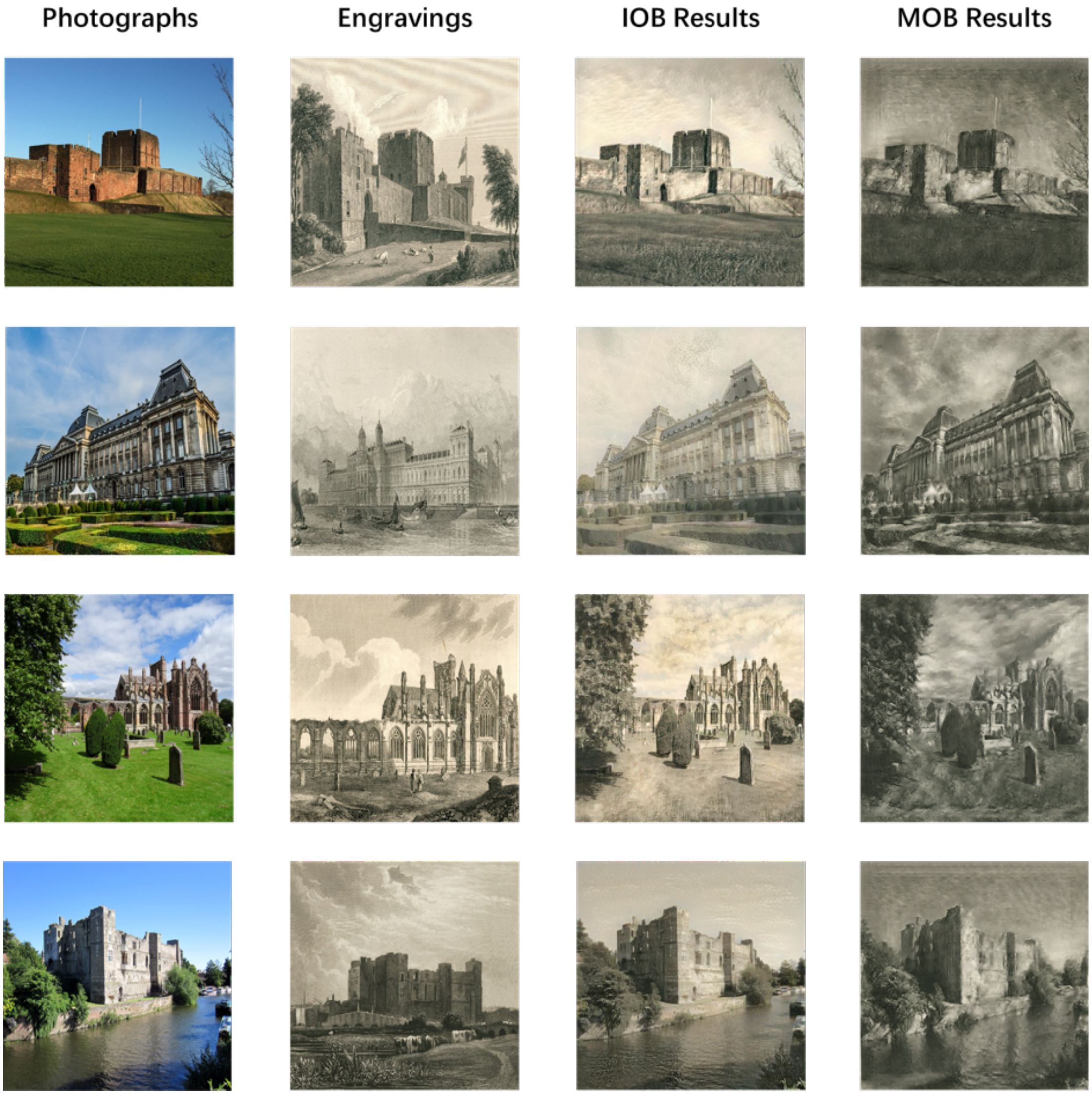


Figure 4.9: Results of different methods

content better than the MOB method. There are also some problems with the IOB method, for example the sky of the image in Sample 3 has a serious cluttered textures. While the MOB method produces more consistent and coherent results, but the contours of the content in the image are blurred and the overall tone is darker than the IOB method results and the engraving examples.

In order to further analyse the generated results objectively, we use two evaluation metrics, Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM), to analyse the generated results quantitatively [21]. PSNR is the ratio of the maximum signal power to the noise power affecting the signal accuracy, and is also the top of the arrival noise ratio, which is calculated as follows:

$$PSNR = 10 \cdot \log_{10}\left(\frac{255^2}{MSE}\right) \tag{4.1}$$

Where 255 is the maximum pixel value of the image I. And MSE is calculated as follows:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{4.2}$$

where I is the image of size $m \times n$ and K is the noise image.

The higher the PSNR value indicates the lower the distortion and the higher the quality of the image reconstruction [21]. And SSIM measures the luminance, contrast and structure of the images [52], which is calculated as follows:

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_x\sigma_y + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{4.3}$$

Where μ_x, μ_y, σ_x, σ_y, σ_x^2, σ_y^2 are the mean, variance and covariance of image x and image y respectively. C_1 and C_2 are constants used to avoid the formulation dividing by zero. A Higher SSIM value indicates that the structure of the two images is more similar. The results are shown in Table 4.2. The data in the table shows that the PSNR values using the IOB method are lower than that of the MOB method, which indicates that the IOB generates images with better quality. While the data for sample 3 is the opposite, with IOB having a higher PSNR value than the other IOB methods and higher than the MOB method, suggesting that the same parameters and structure in the IOB method cannot generate good results for all engraving style images, and that means it needs to be adapted for each content image and style image. Whereas the data from the MOB method shows more stability, indicating that cycleGAN produces a more uniform image after training. The data in SSIM also shows that using the content image as the initial image retains the features of the image better compared to the

Metrics	IOB PNSR	MOB PNSR	IOB SSIM	MOB SSIM
Sample1	14.832	17.475	0.569	0.500
Sample2	13.425	16.954	0.536	0.560
Sample3	20.809	14.792	0.667	0.518
Sample4	13.974	17.003	0.542	0.522
Average	15.760	16.556	0.579	0.525

Table 4.2: Evaluation results of samples for IOB and MOB methods generation

dataset	IOB	MOB
Sample Engravings	151.472	218.711
General Engravings	400.029	350.903

Table 4.3: FID score for different methods

MOB method, however similar to the data in PSNR, the IOB method is also unstable in terms of image structure, while the MOB method generates images with a slightly lower SSIM than the IOB method, but the results are very stable.

As the colours of the images generated by the MOB method are somewhat different from the corresponding engravings, we suspect that one possible reason for this is that the colour features captured in the cycleGAN is more of a darker tone similar to that of the generated images, which comes from the entire dataset we used for training. Therefore, we use Fréchet Inception Distance (FID) for evaluation, which calculates the distance between the generated samples and the real samples in the feature space, which is closer to the real human perception [35]. Lower FID means higher quality and variety of images

FID extracts features through the Inception network [49], which concatenates convolutional kernels with different sizes and pooling layers instead of the manually determined the fixed size of filters in traditional CNNs and let the network learn the parameters by itself [44]. Then using Gaussian models for modelling the feature space to calculate the distance between the image features of two sample sets, which is often used in the evaluation of GAN-generated results, we use the Pytorch-based version published by Maximilian Seitzer on GitHub [45] to calculate FID. We can find from the data in the table that the FID for the sample engravings using the IOB method are lower than that of the MOB method, which means that the images generated by IOB are closer to the style of the sample engravings. However, we also find that for the entire engraving dataset, the FID values using the IOB method are greater than the MOB method, which means that the images generated by the IOB method are only better for the sample engravings, but are not as close to the characteristics of the entire engraving dataset as the images generated by the MOB method.

Chapter 5

Conclusion and Future Work

We studied Neural Style Transfer for engravings based on the IOB and MOB approaches. Based on our experimental results, we found that in the IOB approach, selecting lower-level CNN layers based on the features of the engravings and optimising with the content image as the initial image can produce a stylised image that combines the specified style features and retains the content well. For the whole concept of ”engravings”, the CycleGAN method we use in MOB is a good solution that allows style transfer without paired images, which can find the mapping between realistic images and engravings in non-paired images. After training, our model is able to generate stylised images with overall feature relationships in the engraving dataset in a very short time.

Through qualitative and quantitative evaluation, we can conclude that for a given content image and a style image, IOB is more suitable for specifying the style image of the style image, and more effective if the content image is similar to the overall content of the style image, however this takes longer time to optimise. In contrast, for the fast style transfer task, MOB approach is able to generate stylised images quickly, with image features that are closer to those of the whole dataset rather than specific content features.

However, our methods have their shortcomings. The quality of the images generated by the IOB method is unstable and is closely related to the choice of content and style expression layers, the number of optimization iterations, and the weighting of style and content, etc. Wrong selections cannot achieve the expected results, although our experiments have produced good results for more engraving style images, there are still some problems with generated images, such as over-optimisation of incorrect textures and lack of texture in some blocks. While in our MOB method, the contours of the generated images appear distorted, differing from the contours in the engravings, which are more clearly outlined, and for the architectural image, there are differences in colour between the generated images and the corresponding architectural images.

For future work, we talk in terms of two methods, the IOB method and the MOB method. For the IOB method, we can improve it in the direction of semantic segmentation, for example by using specific textures and parameters for different objects to optimise them, which can lead to more effective results. Furthermore, VGG19 is not the only option, we can also choose other pre-trained models for style transfer, such as ResNet [17]. Furthermore, we can also fine-tune the pre-trained model for the engraving dataset in order to better capture features. For the MOB approach, it can be further divided into two perspectives: data and model. For the data, due to the small size of the engraving sets, we only separate those engravings with obvious textural differences, but as shown in our results, there are still differences in colour and texture for specific content, hence we can expand our dataset by segmenting the training data more specifically and doing some data augmentation work. As for the model, CycleGAN can only transfer between two domains, but not for multiple domains. For the different styles in our dataset we need to train separately for one or two of the domains, whereas StarGAN proposed by Choi et al. [5] can achieve transformation between multiple domains. For our experimental results, the content of the object appears distorted and smeared, and we can add the SSIM metric used for evaluation to the loss function to make it optimised for the content of the object. In addition, the cycle consistency loss used by CycleGAN requires the assumption of a bidirectional mapping between the two domains, which is usually too strict. Park et al. [40] improved it by maximising the mutual information between the input image and the corresponding patch in the target domain based on the idea of contrast learning, which is able to retain the content information of the image well and improve the speed of training significantly.

The engravings vividly illustrate the work of Walter Scott, which depicts the contradictions and social life of Scotland, England and Europe from the Middle Ages to the bourgeois revolution. We hope that through our study on style transfer of his works, we can make Walter Scott more accessible to a wider audience and promote the study of his work.

Bibliography

[1] Amit Agrawal. Gradient domain manipulation techniques in vision and graphics. *ICCV2007 Cource*, 2007.

[2] Michael Ashikhmin. Synthesizing natural textures. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 217–226, 2001.

[3] Robert Aumann and Adam Brandenburger. Epistemic conditions for nash equilibrium. *Econometrica: Journal of the Econometric Society*, pages 1161–1180, 1995.

[4] Guillaume Berger and Roland Memisevic. Incorporating long-range consistency in cnn-based texture generation. *arXiv preprint arXiv:1606.01286*, 2016.

[5] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018.

[6] John P Collomosse and Peter M Hall. Cubist style rendering from photographs. *IEEE Transactions on Visualization and Computer Graphics*, 9(4):443–453, 2003.

[7] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, 2001.

[8] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. *Advances in neural information processing systems*, 28:262–270, 2015.

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

[10] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[11] Liang Gonog and Yimin Zhou. A review: Generative adversarial networks. In *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 505–510. IEEE, 2019.

[12] Bruce Gooch, Erik Reinhard, and Amy Gooch. Human facial illustrations: Creation and psychophysical evaluation. *ACM Transactions on Graphics (TOG)*, 23(1):27–44, 2004.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[14] Xi Guo, Zhicheng Wang, Qin Yang, Weifeng Lv, Xianglong Liu, Qiong Wu, and Jian Huang. Gan-based virtual-to-real image translation for urban scene semantic segmentation. *Neurocomputing*, 394:127–135, 2020.

[15] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

[16] Paul Haeberli. Paint by numbers: Abstract image representations. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 207–214, 1990.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] David J Heeger and James R Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 229–238, 1995.

[19] Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 453–460, 1998.

[20] Aaron Hertzmann. A survey of stroke-based rendering. Institute of Electrical and Electronics Engineers, 2003.

[21] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010.

[22] 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*, pages 1125–1134, 2017.

[23] Nanhao Jin. Cnn-based image style transfer and its applications. In *2020 International Conference on Computing and Data Science (CDS)*, pages 387–390. IEEE, 2020.

[24] Yongcheng Jing, Xiao Liu, Yukang Ding, Xinchao Wang, Errui Ding, Mingli Song, and Shilei Wen. Dynamic instance normalization for arbitrary style transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4369–4376, 2020.

[25] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*, 26(11):3365–3385, 2019.

[26] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[27] Bela Julesz. Visual pattern discrimination. *IRE transactions on Information Theory*, 8(2):84–92, 1962.

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[29] Jan Eric Kyprianidis, John Collomosse, Tinghuai Wang, and Tobias Isenberg. State of the" art": A taxonomy of artistic stylization techniques for images and video. *IEEE transactions on visualization and computer graphics*, 19(5):866–885, 2012.

[30] Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2479–2486, 2016.

[31] Rui Li, Wenming Cao, Qianfen Jiao, Si Wu, and Hau-San Wong. Simplified unsupervised image translation for semantic segmentation adaptation. *Pattern Recognition*, 105:107343, 2020.

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

[33] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual attribute transfer through deep image analogy. *arXiv preprint arXiv:1705.01088*, 2017.

[34] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10551–10560, 2019.

[35] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. *arXiv preprint arXiv:1711.10337*, 2017.

[36] Somshubra Majumdar, Amlaan Bhoi, and Ganesh Jagadeesan. A comprehensive comparison between neural style transfer and universal style transfer. *arXiv preprint arXiv:1806.00868*, 2018.

[37] Youssef A Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image to image translation. *arXiv preprint arXiv:1806.02311*, 2018.

[38] Quanzheng Mou, Longsheng Wei, Conghao Wang, Dapeng Luo, Songze He, Jing Zhang, Huimin Xu, Chen Luo, and Changxin Gao. Unsupervised domain-adaptive scene-specific pedestrian detection for static video surveillance. *Pattern Recognition*, 118:108038, 2021.

[39] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision*, pages 3456–3465, 2017.

[40] Yingxue Pang, Jianxin Lin, Tao Qin, and Zhibo Chen. Image-to-image translation: Methods and applications. *arXiv preprint arXiv:2101.08629*, 2021.

[41] Javier Portilla and Eero P Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International journal of computer vision*, 40(1):49–70, 2000.

[42] Paul Rosin and John Collomosse. *Image and video-based artistic stylisation*, volume 42. Springer Science & Business Media, 2012.

[43] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos. In *German conference on pattern recognition*, pages 26–36. Springer, 2016.

[44] Suriani Mohd Sam, Kamilia Kamardin, Nilam Nur Amir Sjarif, Norliza Mohamed, et al. Offline signature verification using deep learning convolutional neural network (cnn) architectures googlenet inception-v1 and inception-v3. *Procedia Computer Science*, 161:475–483, 2019.

[45] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. `https://github.com/mseitzer/pytorch-fid`, August 2020. Version 0.1.1.

[46] Falong Shen, Shuicheng Yan, and Gang Zeng. Neural style transfer via meta networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8061–8069, 2018.

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

[48] Stuart Snydman, Robert Sanderson, and Tom Cramer. The international image interoperability framework (iiif): A community & technology approach for web-based images. In *Archiving conference*, volume 2015, pages 16–21. Society for Imaging Science and Technology, 2015.

[49] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[50] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, pages 839–846. IEEE, 1998.

[51] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*, volume 1, page 4, 2016.

[52] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[53] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 479–488, 2000.

[54] Georges Winkenbach and David H Salesin. Computer-generated pen-and-ink illustration. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 91–100, 1994.

[55] Holger Winnemöller, Sven C Olsen, and Bruce Gooch. Real-time video abstraction. *ACM Transactions On Graphics (TOG)*, 25(3):1221–1226, 2006.

[56] Qianye Yang, Nannan Li, Zixu Zhao, Xingyu Fan, Eric I Chang, Yan Xu, et al. Mri cross-modality neuroimage-to-neuroimage translation. *arXiv preprint arXiv:1801.06940*, 2018.

[57] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE international conference on computer vision*, pages 2849–2857, 2017.

[58] Lei Zhao, Qihang Mo, Sihuan Lin, Zhizhong Wang, Zhiwen Zuo, Haibo Chen, Wei Xing, and Dongming Lu. Uctgan: Diverse image inpainting based on unsupervised cross-space translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5741–5750, 2020.

[59] Mingtian Zhao and Song-Chun Zhu. Portrait painting using active templates. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering*, pages 117–124, 2011.

[60] 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.

Appendix A

Related Materials

A.1 VGG19

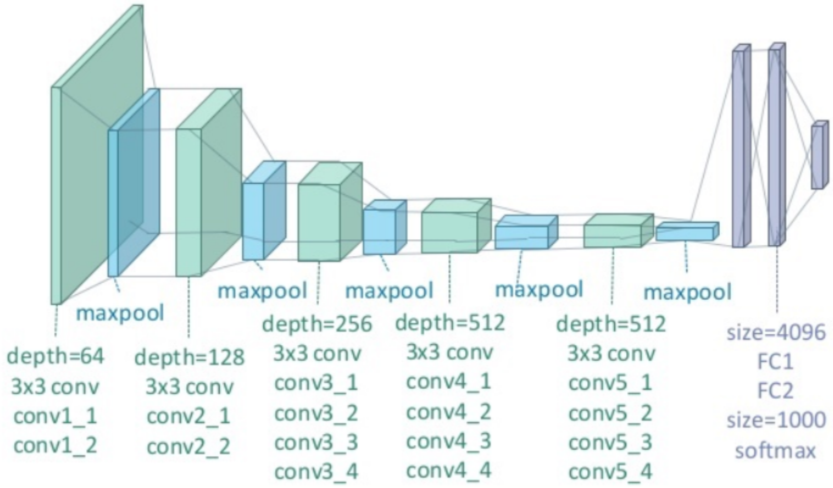


Figure A.1: VGG19 Model Structure

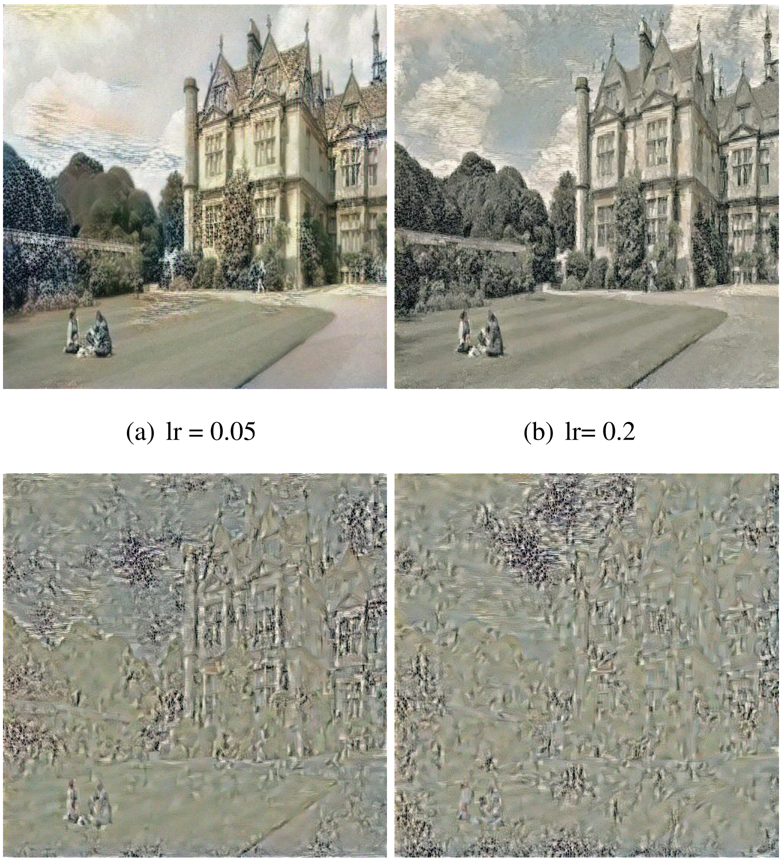
A.2 Google-Landmarks Dataset

Figure A.2: Photos in Google-Landmarks Dataset

Appendix B

Experiment results

B.1 IOB Results with Adam optimiser



(a) lr = 0.05

(b) lr= 0.2

(c) lr = 0.5

(d) lr = 1

Figure B.1: Results with different learning rate with Adam optimiser

B.2 IOB high-resolution generation results

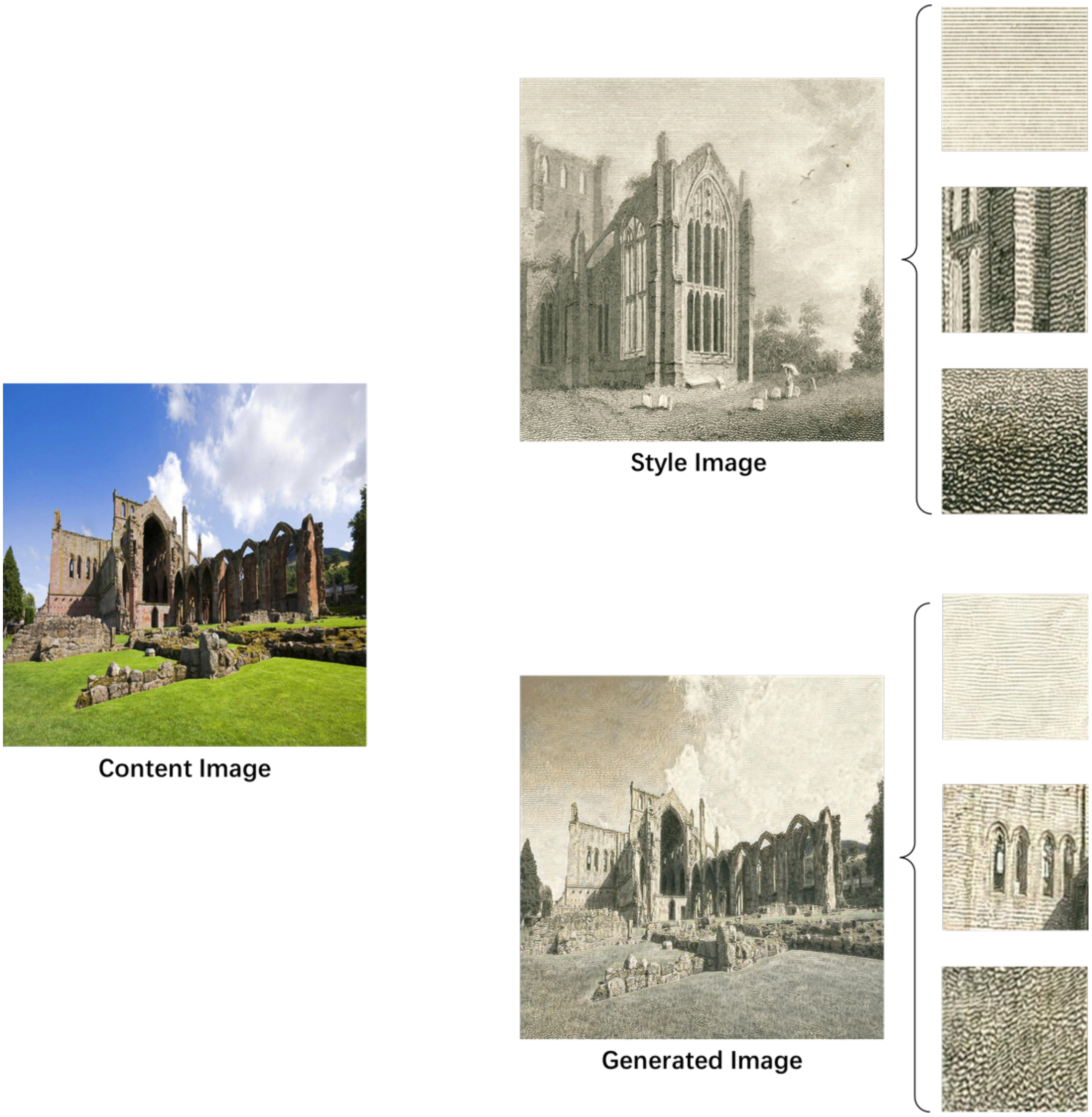
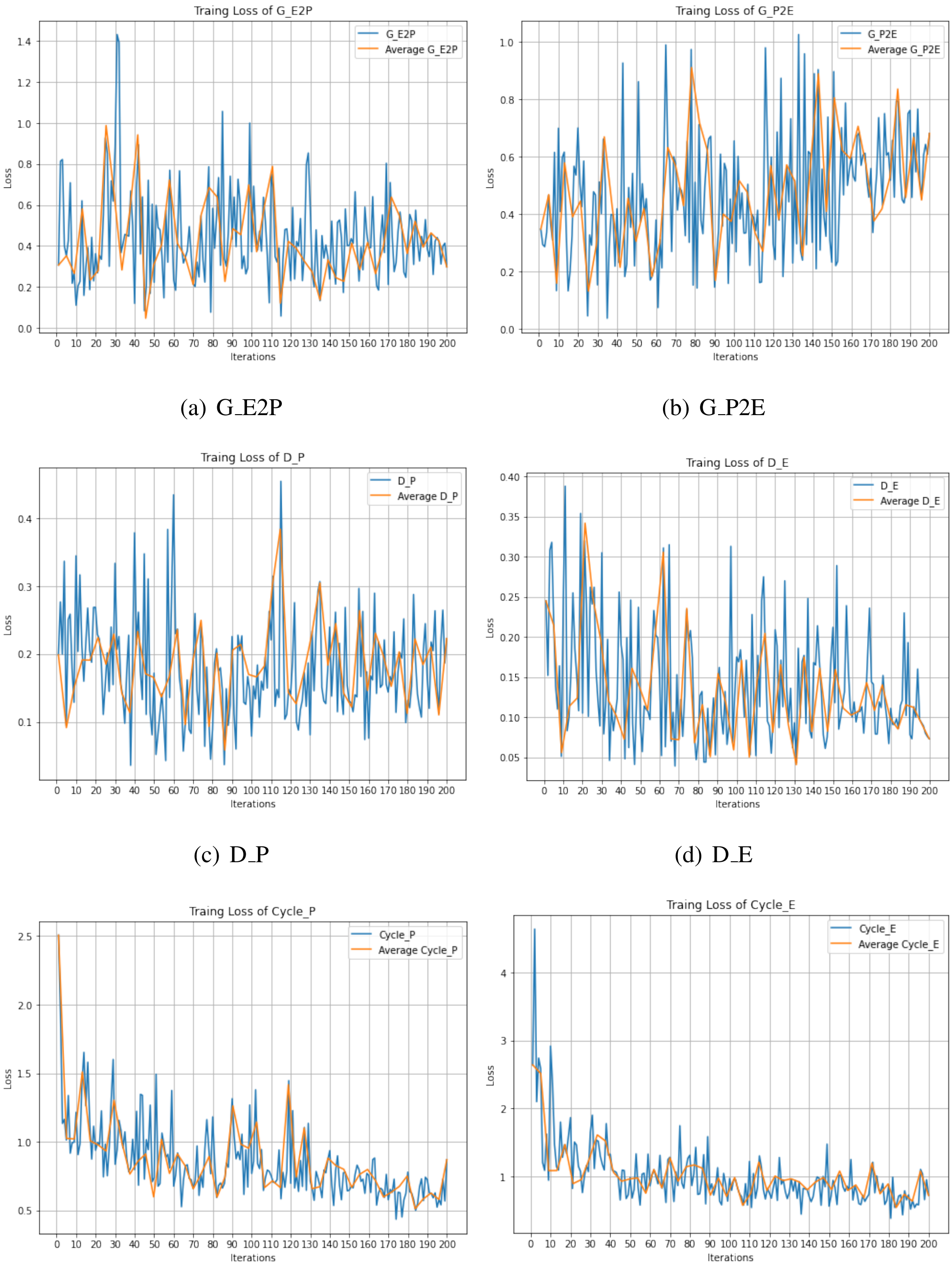


Figure B.2: High-resolution IOB results

B.3 MOB CycleGAN Traing losses



(a) G_E2P (b) G_P2E

(c) D_P (d) D_E

(e) Cycle_P (f) Cycle_E

Figure B.3: Training losses of cycleGAN

B.4 MOB Cycle consistency results

Figure B.4: Results from photographs to engravings to photographs

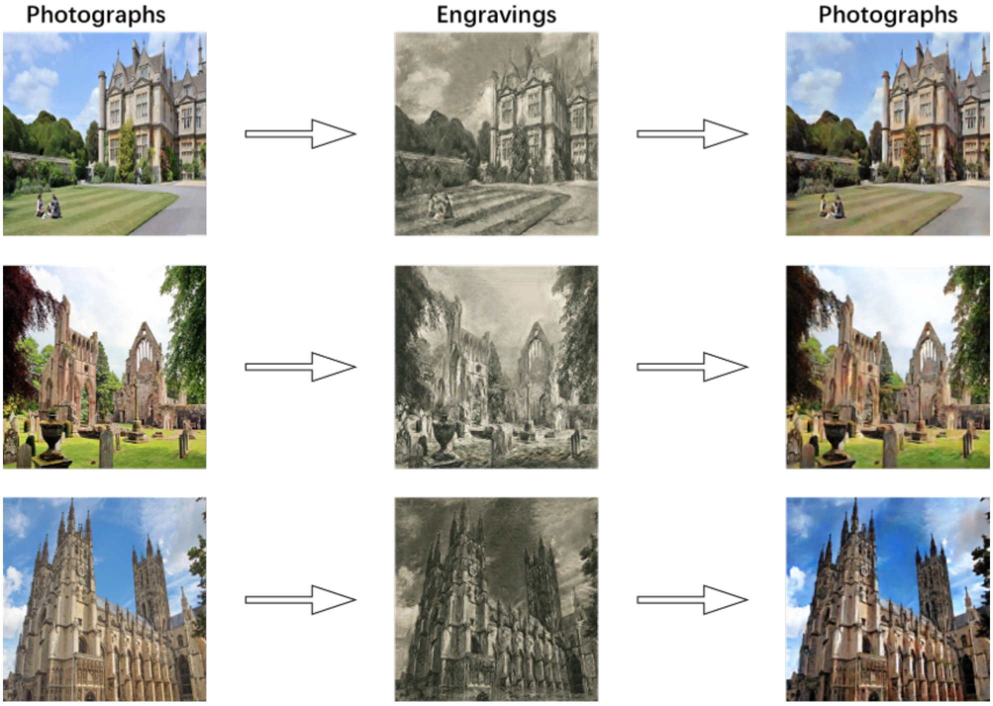


Figure B.5: Results from engravings to photographs to engravings