Generating Artistic Portrait Drawings from Images

Ran Yi, Yong-Jin Liu, Yu-Kun Lai, and Paul L. Rosin

Abstract This chapter addresses generating artistic portrait drawings (APDrawings) from images, and we focus on two methods based on generative adversarial networks (GANs). We first introduce the genre of portrait line drawings, and review some existing methods for generating them from images. We also describe the Artistic Portrait Drawing (APDrawing) dataset, which contains 140 high-resolution face photos and corresponding portrait drawings executed by a professional artist. We then describe the APDrawingGAN method, which is a hierarchical GAN model that learns from paired data of face photos and portrait drawings, and the QMUPD method, which can learn from unpaired data of face photos and drawing. APDrawingGAN uses a novel distance transform loss to learn stroke lines in the drawings, and a local transfer loss to capture different drawing styles for different facial regions. QMUPD uses an asymmetric cycle mapping to preserve important facial features, and a quality metric to guide the generation towards high-quality drawings. We further introduce some recent developments which are based on multiple scale analysis, 3D information and multi-modal information. Finally, we describe the evaluation of artistic portrait drawings, which is a challenging task since there are many possible drawings that would be considered by experts to be acceptable.

R. Yi (✉)
Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
e-mail: ranyi@sjtu.edu.cn

Y.-J. Liu
MOE-Key Laboratory of Pervasive Computing, BNRist, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
e-mail: liuyongjin@tsinghua.edu.cn

Y.-K. Lai · P. L. Rosin
School of Computer Science and Informatics, Cardiff University, CF24 3AA Cardiff, UK
e-mail: laiy4@cardiff.ac.uk

P. L. Rosin
e-mail: rosinpl@cardiff.ac.uk

Z. Lyu (ed.), *Applications of Generative AI*,
https://doi.org/10.1007/978-3-031-46238-2_22

1 Introduction

This chapter focuses on the genre of portrait line drawings, and is therefore circumscribed both by medium (typically pen or pencil) and topic (typically human faces, although some artists specialise in non-humans, e.g. the horse portraits painted by George Stubbs). Nevertheless, portrait line drawings still cover a large range of styles, in part due to their long historical development, as well as their different applications. For instance, more than 2000 years ago, the ancient Greeks produced thousands of painted pottery vases, and Fig. 1a shows a portrait line drawing from the red-figure classical period, which looks fairly contemporary. While these drawings consisted of clean outlines, the consequence of eliminating colour and texture and using just lines, is that it becomes hard to capture shading. Removing colour also introduces problems, although both artists and researchers have developed solutions [2]. One method for retaining some aspect of shading is to introduce hatching Fig. 1b, although the main methods described in this chapter aim towards generating drawings with fewer lines. This is in the spirit of our earlier work whose goal was to perform minimal rendering with lines as well as regions [29]. Depending on their goals, artists might switch between the different styles, as seen in the example by Leonardo da Vinci in Fig. 1c. Here, one version of the head is drawn in detail with careful hatching to provide good modelling of the surface, whereas another version is more an outline for quickly exploring some possible design options. Moving forwards to the twentieth century, Fig. 2 shows different styles, e.g. minimal/clean, messy, highly stylised, and minimal/cartoon. Figure 3 shows how, from the basis of similar photographs, an artist can derive dissimilar artworks, e.g. realist versus highly stylised.

Moving to computer generated line portraits, early work in the non-photorealistic rendering (NPR) community developed various approaches, many of which involved lines and solid black regions. For instance, Gooch et al. [12] extracted lines using difference of Gaussian filters (DoG) at multiple scales followed by global threshold-

Fig. 1 Examples of portrait line drawings from the last 2500 years. **a** section of a vase in the Attic red-figure style by Aristophanes (410–400 B.C.), **b** Self-portrait with Long Bushy Hair (1629–1633) by Rembrandt, **c** Study for the head of Leda (1503–1507) by Leonardo da Vinci

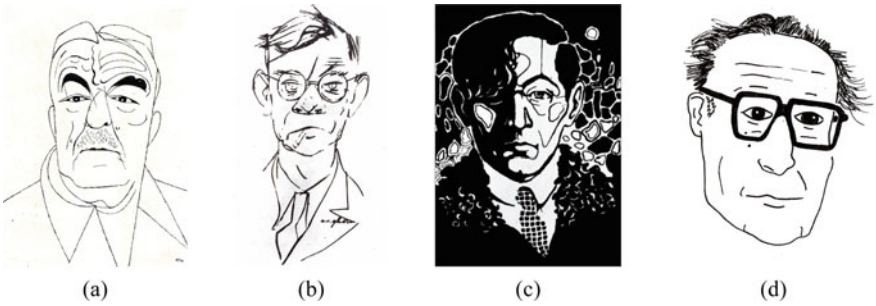

Fig. 2 Twentieth century portrait line drawings. **a** Fernand Léger (1956) by Adolf Hoffmeister, **b** Hans Fallada (1943) by Erich Ohser, **c** Self portrait (1921) by Geo Milev, **d** André Gorz (2022) by HerB104

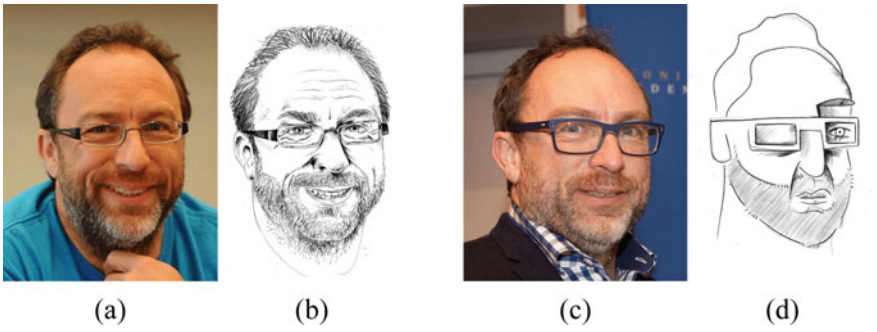

Fig. 3 Portraits of Jimmy Wales by Jericó Delayah derived from the source photographs

ing. The lines were combined with dark regions that were extracted from the source intensity image by thresholding. As Fig. 4b shows the results are reasonable, but in the absence of additional filtering are somewhat noisy even for simple input images. The results from Meng et al. [24] in Fig. 4c are meant to simulate paper-cuts. These are effectively binary renderings with the extra constraint that the black pixels form a single connected region. Their approach is more complex, involving a hierarchical composition model which represents the face by an AND-OR graph in which the nodes represent facial components. Facial features are located in the source image by fitting an active appearance model [9], from which local thresholding produces a set of "proposal" regions which are matched to the graph. Finally, post-processing is applied to extract the hair and clothing using graph cut segmentation, and enforce connectivity by inserting some curves. Like Gooch et al., Rosin and Lai [29] combine lines and regions. In their case lines are extracted using Kang et al.'s [19] coherent line drawing algorithm, which constructs a smooth edge tangent flow following the salient image edges that determines the kernel shape for the DoG filter. Both black and white (negative) lines are extracted. Regions are extracted by applying thresholding followed by GrabCut [33]. The results shown in Fig. 4d are cleaner than those of Gooch et al., although some details have been lost (e.g. the jaw-line).

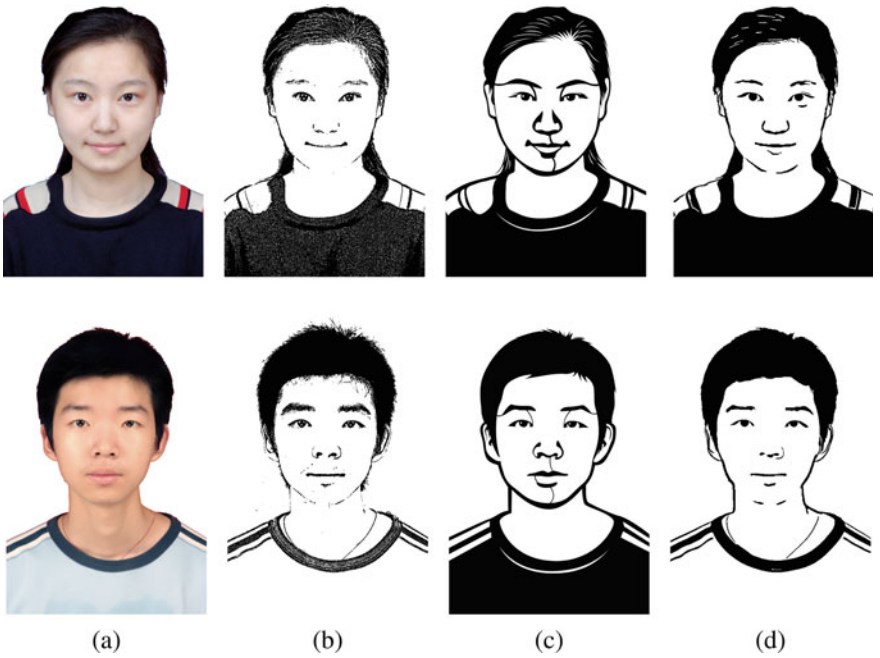

Fig. 4 Various black and white line and region NPR portrait renderings. **a** Source image, **b** Gooch *et al.* [12], **c** Meng *et al.* [24], **d** Rosin and Lai [29]

Figure 5 shows further examples of line drawings, some specifically designed for portraits, while others are general purpose such as Chiu et al.'s circular scribble art [6]. This produces a whimsical circular scribble pattern that is attractive, but does not really capture the portrait's identity, see Fig. 5b, c. Their system first generates a virtual tracing path that takes the image's intensity and edge structure into account. Circular scribbles are synthesized along the virtual path, with the circle radius controlled by the local intensity. Figure 5d shows a binary version of the heavily stylised Julian Opie effect produced by Rosin and Lai's portrait stylisation method [30]; it uses the black and white lines and regions produced by their earlier minimal rendering style [29] and a template to create the facial features. Stippling is a popular general purpose image stylisation approach, widely used both by artists and the wider community. Given that it uses a huge number of graphical elements (stipples), it is the antithesis of this chapter's focus on more minimal stylisation. However, there is a portrait-specific variant of stippling called *hedcut*, shown in Fig. 5e that produces a clearer effect. This result was generated with Son et al.'s [38] algorithm, that uses a regularly spaced grid of dots and hatching lines which are deformed to fit the image. Rosin and Lai [31] created an engraving stylisation using a dither matrix (i.e. a spatially-varying threshold) that generates a pattern of black and white lines forming cross hatching. A simple cylindrical model of the face warped the dither matrix so that the lines curve around the face, providing a pseudo-3D effect—see

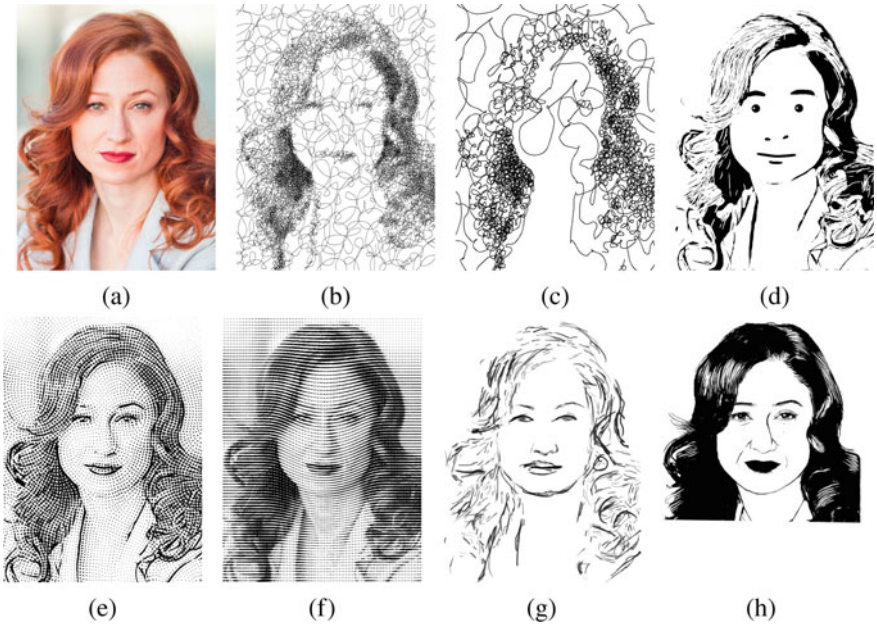

Fig. 5 Various black and white portrait stylisations. **a** original image, **b**, **c** Chiu et al.'s circular scribble art [6], **d** Rosin and Lai's 'Julian Opie' style [30], **e** Son et al.'s hedcut [38], **f** Rosin and Lai's engraving [31], **g** Berger et al.'s [3] portrait sketching, **h** Yi et al.'s APDrawingGAN [49]

Fig. 5f. Berger et al. [3] use the statistics of a set of drawings of artists to drive an algorithm that creates a contour image, detects facial features, and then modifies the face geometry to follow the specific artist's geometric style. Finally, contours are drawn using strokes from the artist's stroke database, see Fig. 5g. The style is intentionally sketchy, which enables it to effectively hide errors in rendering. The last result, Fig. 5h, shows a stylisation by APDrawingGAN [49]—this method will be described in more detail in the next section.

It can be seen that generating high quality portrait line drawings is challenging, and this comes from two fronts. First, the use of a sparse set of lines rather than a dense set of graphical primitives (e.g. painting strokes or stipples) means that any errors in these lines is significant. A mislocalisation or deformation of even a single line can become evident to the viewer, and spoil the artistic effect. In comparison, an error in an individual stipple will barely be visible. Second, the human visual system is especially sensitive to the human face, and will quickly perceive any errors. For instance, a missing eye on a portrait is unacceptable, and of much greater consequence than, e.g. a missing finger.

2 APDrawingGAN

APDrawingGAN [49, 50] is a Hierarchical Generative Adversarial Network (GAN) model dedicated to face structure and Artistic Portrait Line Drawing (APDrawing) styles for transforming face photos to high-quality APDrawings. To effectively learn different drawing styles for different facial regions, the APDrawingGAN architecture involves several local networks dedicated to facial feature regions, along with a global network to capture holistic characteristics. To further cope with line-stroke-based style and imprecisely located elements in artists' drawings, APDrawingGAN proposed a novel distance transform (DT) loss to learn stroke lines in APDrawings.

2.1 Challenges

APDrawingGAN addressed the following five challenges to improve the quality of artistic portrait drawings (APDrawing). In addition to the two previously mentioned, namely sparse graphical elements and sensitivity of the human visual system to faces, some additional challenges include:

- In previous methods, different facial areas may be rendered in different styles (*e.g.,* eyes vs. hair).
- APDrawings will make some trade-offs to the elements of the original face, posing a challenge for methods based on pixel correspondence (*e.g.,* Pix2Pix [18]).
- In APDrawings, some lines are not directly related to low level features in the view or photograph of the person.

Figure 6 gives some examples. These examples include lines in the hair indicating the flow, or lines indicating the presence of facial features even if the image contains no intensity or colour discontinuities. Such elements of the drawings are hard to learn. Therefore, many image style transfer algorithms (*e.g.,* [11, 18, 20, 21, 37, 57]) inevitably fail to produce good and expressive artistic portraits (Fig. 7).

To solve the above challenges, APDrawingGAN firstly uses a Hierarchical GAN architecture for artistic portrait drawing synthesis from a face photo, which can generate high-quality and expressive artistic portrait drawings. To best emulate artists, who use multiple graphical elements when creating a drawing, APDrawingGAN separates the GAN's rendered output into multiple layers, each of which is controlled by separate loss functions. The APDrawing dataset is constructed to facilitate research in this area, and contains 140 high-resolution face photos and corresponding portrait drawings executed by a professional artist. Fig. 7 shows the qualitative results of APDrawingGAN and the comparison with seven neural style transfer and image-to-image translation methods.

Fig. 6 Some examples of image pairs (each pair contains a face photo and an artist's portrait drawing) in the APDrawing dataset [49]

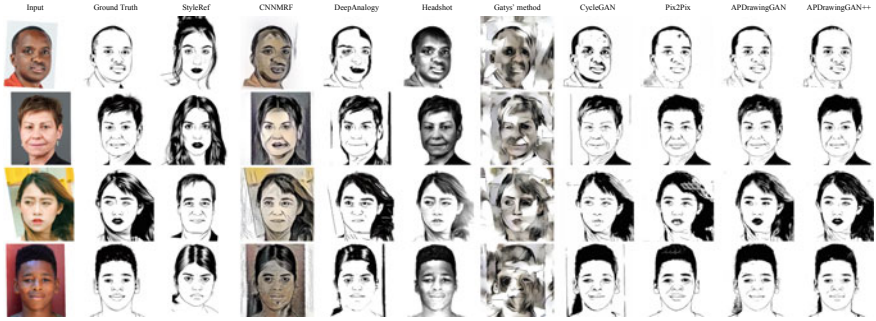

Fig. 7 Qualitative results of APDrawingGAN/APDrawingGAN++ and comparison with seven neural style transfer and image-to-image translation methods. From left to right: input face photos, ground truth APDrawings, the randomly-chosen style images for methods which take one content and one style image as input, CNNMRF [20] results, Deep Image Analogy [21] results, Headshot Portrait [37] results, Gatys [11] results, CycleGAN [57] results, Pix2Pix [18] results, the original APDrawingGAN [49] results, APDrawingGAN++ [47] results. Compared with the original APDrawingGAN, APDrawingGAN++ uses auto-encoders, classifiers for lip and hair, and line continuity loss for better qualitative results

2.2 Technical Details of APDrawingGAN

The process of learning to transform face photos to APDrawings can be modeled as a function Ψ which maps the face photo domain $\mathcal{P}$ into a black-and-white line-stroke-based APDrawing domain $\mathcal{A}$. The function Ψ is learnt from paired training data $S_{data} = \{(p_i, a_i) | p_i \in \mathcal{P}, a_i \in \mathcal{A}, i = 1, 2, \ldots, N\}$, where N is the number of photo-APDrawing pairs in the training set. The discussion in this section focuses on the extended version called APDrawingGAN++ [47], which uses additional auto-encoders for fine facial features, classification for lips and hair, and line continuity

loss to improve the line quality. To streamline the text, unless explicitly specified, we use APDrawingGAN to refer to the extended model.

APDrawingGAN consists of a generator G and a discriminator D, both of which are convolutional neural networks specifically designed for line drawing-based APDrawings in the style of artist's drawings. The generator G learns the APDrawing of the output $\mathcal{A}$, while the discriminator D serves to determine whether an image is real or generated by the generator.

The discriminator D is trained to classify the real $a_i \in \mathcal{A}$ and the synthetic image $G(p_i)$, $p_i \in \mathcal{P}$ as accurately as possible, while G is trained to minimize this probability. The loss function, denoted $L(G, D)$, is specifically designed with five terms $L_{adv}(G, D), L_{\mathcal{L}_1}(G, D), L_{DT}(G, D), L_{local}(G, D)$ and $L_{conti}(G, D)$. Then the function Ψ can be formulated using the function $L(G, D)$ to solve the following min-max problem:

$$\begin{aligned}\min_G \max_D L(G, D) = L_{adv}(G, D) + \lambda_1 L_{\mathcal{L}_1}(G, D) \\ + \lambda_2 L_{DT}(G, D) + \lambda_3 L_{local}(G, D) + \lambda_4 L_{conti}(G, D).\end{aligned} \tag{1}$$

2.2.1 Hierarchical Generator G

The hierarchical generator G converts the input face photos into APDrawings. The model is trained on one style of APDrawings at a time. In the hierarchy of $G = \{G_{global}, G_{l*}, E_*, C_*, G_{fusion}\}$, G_{global} is a global generator, $G_{l*} = \{G_{l\_eye\_l}, G_{l\_eye\_r}, G_{l\_nose}, G_{l\_mouth}, G_{l\_hair}, G_{l\_bg}\}$ is a set of six local generators. $E_* = \{E_{eye\_l}, E_{eye\_r}, E_{nose}, E_{lip\_b}, E_{lip\_w}\}$ is a set of five auto-encoders, $C_* = \{C_{lip}, C_{hair}\}$ is a set of two classifiers and G_{fusion} is a fusion network.

The generator G uses the U-Net structure [28]. $G_{l\_eye\_l}$, $G_{l\_eye\_r}$, G_{l_nose} and G_{l_mouth} are all U-Nets with three downward and three upward convolutions. G_{l_hair} and G_{l_bg} are U-Nets with four downward and four upward convolution blocks. In G_{l*}, the role of the local generator is to learn the drawing styles of different local facial features; for example, the hair style for hair (i.e., capturing the soft, flowing details of individual strands of hair with short or long strokes), the delicate line style for eyes and noses, and the solid or line style for mouths. A U-Net with skip connections can incorporate multi-scale features and provide sufficient but not excessive flexibility to learn the artist's drawing techniques for different facial regions in APDrawings.

Local Generators. The inputs of $G_{l\_eye\_l}$, $G_{l\_eye\_r}$, G_{l_nose}, and G_{l_mouth} are local regions centered on facial elements (i.e., left eye, right eye, nose and mouth) as the centered local regions, obtained from the MTCNN model [54]. The input of G_{lbg} is the background region detected by the portrait segmentation method [36]. The input of G_{hair} is the remaining region in the face photo. The outputs of all local generators are blended into an aggregated picture I_{local} by using min pooling in the overlapping regions. This min pooling effectively preserves the responses of individual local generators, because in artistic pictures, low intensities are considered as the responses of black pixels.

Global Generator. G_{global} is a U-Net with eight lower and eight upper convolutional blocks, which handles the global structure of the face. G_{fusion} consists of a vanilla convolution block (the feature map size stays the same), six residual blocks and a final convolution layer. G_{fusion} is used to fuse I_{local} and I_{global} (i.e., the output of G_{global}) to obtain the final synthetic map of G. In many previous GAN models (*e.g.,* [13, 17]), some noise is usually input or injected to the generator network. Following [18], instead of adding noise explicitly in G, APDrawingGAN uses dropout [39] as noise in the U-Net block.

Fusion Network. The fusion Network G_{fusion} is used to fuse the output from the local and global generators together for final portrait drawing synthesis. This block helps combine different drawing techniques learnt by different generators (G_{global}, G_{l*}).

Handling Multiple Styles for Lips and Eyes. In the APDrawing dataset, lips and hair exhibit multiple styles, *e.g.,* white/black lips, and dark/middle/light hair (Fig. 6). We use two classifiers for lip and hair (C_{lip}, C_{hair}) to detect the target style for the lip and hair regions respectively, and the detected class information is then used to guide the generation toward the desired style.

Autoencoders for Fine APDrawing. In the original APDrawingGAN [49], the generator G only consists of local generators, a global generator and a fusion network, where the main loss function was calculated on the fused result output from the fusion network, while the local generators' outputs are only supervised by a local loss. Therefore, the local drawings output from the local generators $G_{l\_eye\_l}$, $G_{l\_eye\_r}$, G_{l_nose}, G_{l_lip} are not as delicate as the artist drawn drawings. In APDrawingGAN++ [47], a set of auto-encoders E_{eye_l}, E_{eye_r}, E_{nose}, $E_{lip\_b/w}$ (corresponding to the left eye, right eye, nose and lip) are designed to improve local drawings and generate better facial feature drawings in fine detail. Both the coarse input and fine output of these auto-encoders are parts of APDrawings. Trained with the APDrawing dataset, each auto-encoder learns a good feature representation and reconstructs high-quality APDrawings close to the artist drawings.

2.2.2 Hierarchical Discriminator D

The discriminator D distinguishes whether the input drawing is a genuine portrait of the artist. In the hierarchy of $D = \{D_{global}, D_{l*}\}$, D_{global} is a global discriminator, and $D_{l*} = \{D_{l\_eye\_l}, D_{l\_eye\_r}, D_{l\_nose}, D_{l\_mouth}, D_{l\_hair}, D_{l\_bg}\}$ is a set of six local discriminators. D_{global} examines the whole drawing to determine the overall APDrawing features, and the local discriminators in D_{l*} examine different local areas to evaluate the quality of the details.

D_{global} and all local discriminators in D_{l*} use the Markovian discriminator in Pix2Pix [18]. The only difference is the input: the whole drawings or different local regions. The Markovian discriminator processes each 70×70 patch in the input image and examines the style of each patch. Local patches from different granularities (i.e., coarse and fine levels at global and local input) allow the discriminator to

learn local patterns and better discriminate real artists' drawings from synthesized drawings.

2.2.3 Loss Function

There are five terms in the loss function in Eq. 1, which are explained as follows.

Adversarial loss. L_{adv} models the ability of the discriminator to correctly distinguish between true and false APDrawings. According to Pix2Pix [18], the adversarial loss is formulated as

$$L_{adv}(G, D) = \sum_{D_j \in D} \mathbb{E}_{(p_i, a_i) \sim S_{data}}[\log(D_j(p_i, a_i) + \log(1 - D_j(p_i, G(p_i)))].$$

When $D_j \in D_{l*}$, the images p_i, a_i and $G(p_i)$ are restricted to the local region specified by D_j. Since D maximizes this loss and G minimizes it, L_{adv} forces the synthesized picture to become closer to the target domain $\mathcal{A}$.

Pixel-wise loss. $L_{\mathcal{L}_1}$ drives the synthesised image close to the ground truth APDrawing image in a pixel-wise way. The loss of $L_{\mathcal{L}_1}$ is computed for each pixel in the entire drawing:

$$L_{\mathcal{L}_1}(G, D) = \mathbb{E}_{(p_i, a_i) \sim S_{data}}[\|G(p_i) - a_i\|_1]. \tag{2}$$

Using $\mathcal{L}_1$ norm usually results in less blurry output than $\mathcal{L}_2$ norm, so it is more suitable for APDrawing style.

Line-promoting distance transform loss. Since the position of elements in APDrawings does not precisely correspond to the intensity of the image, L_{DT} is a loss specifically designed to facilitate line strokes in the APDrawing style. L_{DT} is designed to tolerate the small misalignments often found in artist portraits and to better learn the lines in APDrawings. It relies on the Distance Transformation (DT) and Chamfer matching.

A DT (also known as a "distance map") can be represented as a digital image in which each pixel stores a distance value. Given a real or synthetic APDrawing x, the two DTs of x are defined as the images $I_{DT}(x)$ and $I'_{DT}(x)$: Suppose that $\hat{x}$ is the binarized image of x, each pixel in $I_{DT}(x)$ stores the distance to the closest black pixel in $\hat{x}$ and each pixel in $I'_{DT}(x)$ stores the distance to its nearest white pixel.

Two convolutional neural networks are used to detect the black and white lines in APDrawings, denoted as Θ_b and Θ_w, respectively. The Chamfer matching distance between APDrawings x_1 and x_2 is defined as:

$$\begin{aligned} d_{CM}(x_1, x_2) = & \sum_{(j,k) \in \Theta_b(x_1)} I_{DT}(x_2)(j, k) \\ & + \sum_{(j,k) \in \Theta_w(x_1)} I'_{DT}(x_2)(j, k), \end{aligned} \tag{3}$$

where $I_{DT}(x)(j,k)$ and $I'_{DT}(x)(j,k)$ are the distance values of pixels (j,k) in the images $I_{DT}(x)$ and $I'_{DT}(x)$, respectively. $d_{CM}(x_1, x_2)$ measures the sum of the distances from each line pixel in x_1 to the nearest pixel of the same type (black or white) in x_2. Then L_{DT} is defined as:

$$\begin{aligned} L_{DT}(G, D) = \mathbb{E}_{(p_i,a_i)\sim S_{data}}[d_{CM}(a_i, G(p_i)) \\ + d_{CM}(G(p_i), a_i)]. \end{aligned} \tag{4}$$

Local transfer loss. L_{local} imposes additional constraints on the intermediate outputs of the six local generators in G_{l*}, which are then used as regularization terms for the loss function. The six local regions of APDrawing x are denoted by $El(x)$, $Er(x)$, $Ns(x)$, $Mt(x)$, $Hr(x)$, and $Bg(x)$. L_{local} is defined as

$$\begin{aligned} L_{local}(G, D) = \mathbb{E}_{(p_i,a_i)\sim S_{data}}\big[&||G_{l\_eye\_l}(El(p_i)) - El(a_i)||_1 \\ &+ ||G_{l\_eye\_r}(Er(p_i)) - Er(a_i)||_1 \\ &+ ||G_{l\_nose}(Ns(p_i)) - Ns(a_i)||_1 \\ &+ ||G_{l\_mouth}(Mt(p_i)) - Mt(a_i)||_1 \\ &+ ||G_{l\_hair}(Hr(p_i)) - Hr(a_i)||_1 \\ &+ ||G_{l\_bg}(Bg(p_i)) - Bg(a_i)||_1\big]. \end{aligned} \tag{5}$$

Line continuity loss. Line continuity is important in APDrawings, and the lines in the human artist drawings are often continuous. To promote the model to generate more continuous lines, a line continuity loss can be used to guide the model training. A line continuity prediction network R_{conti} is designed to predict the line continuity score from a drawing patch, which is trained from the artist patches (which are assigned the highest continuity score) and manufactured defect patches (which are generated by randomly inverting line or non-line pixels in artist patches and are assigned lower continuity scores).

In detail, the line continuity prediction network R_{conti}, which contains three flat-convolutions and a fully-connected layer, takes an 11×11 patch as input and outputs a single value for line continuity. As described above, the line continuity score of an APDrawing I can be defined as:

$$S_{conti}(x) = \mathbb{E}_{\rho_k\sim P(x)} R_{conti}(\rho_k), x \in I, \tag{6}$$

where $P(x)$ is the set of all patches that are not pure white or pure black, extracted from I, and ρ_k is the k-th patch in this set. The higher the line continuity score, the more continuous the lines in APDrawing I. For face and non-face patches, there are patch sets $P_{face}(x)$ and $P_{non-face}(x)$. And the line continuity loss can be defined as:

$$L_{conti}(G, D) = \mathbb{E}_{(p_i,a_i)\sim S_{data}} \mathbb{E}_{\rho_k\sim P(G(p_i))} w_k(1 - R_{conti}(\rho_k)), \tag{7}$$

where weight $w_k = 2$ if $\rho_k \in P_{face}(G(p_i))$, and $w_k = 1$ if $\rho_k \in P_{nface}(G(p_i))$. Since face patches have a complicated set of lines, the lines in the face area are often less continuous and need to be given higher weights to avoid them being unfairly penalised.

3 Unpaired Portrait Drawing Generation (UPD)

APDrawingGAN [49, 50] introduced in the previous section is trained using paired data consisting of face photos and APDrawings. However, paired data is costly to obtain, requiring professional artists hours to draw each delicate APDrawing. In comparison, unpaired training data collected from websites is easier to obtain. But training APDrawingGAN to perform generation from unpaired training data is more challenging than learning from paired training data, because: (1) Paired training data provides a more direct guidance for learning the photo-to-drawing mapping, while unpaired training data cannot provide such direct guidance; (2) Paired training data is usually specially collected and drawn by a few artists, which means the samples are both high quality and uniform in style, whereas this is less easy to achieve when collecting the necessarily large sets of unpaired samples.

In this section, we introduce the Quality-Metric-guided Unpaired Portrait line Drawing Generation method (QMUPD) [47, 48], which targets the scenario in which only unpaired training data is available. Previous methods for unpaired image-to-image translation [52, 56] use a cycle structure to regularize training. Due to the significant imbalance in information richness between photos and drawings, some existing unpaired transfer methods, such as CycleGAN [56], tend to indiscriminately embed invisible reconstruction information throughout the drawings, resulting in important facial features partially lost in the drawings (*e.g.,* Fig. 8 second column). The problem mentioned above can be solved using a new asymmetric cycle mapping by forcing the reconstruction information to be visible (via truncation loss) and embedded only in selective facial regions (via a relaxed forward cycle consistency loss). Together with local discriminators for eyes, nose and lips, the asymmetric cycle mapping well preserves all important facial features in the generated portraits. By introducing a style classifier and taking into account style features, the Quality-Metric-guided Unpaired Portrait line Drawing Generation can learn to generate multiple styles of portraits using a single network. Figures 8 and 9 show the results of the Quality-Metric-guided Unpaired Portrait line Drawing Generation and comparison methods. Due to the use of unpaired data, we consider three typical APDrawing styles, trained using APDrawing images collected from the internet.

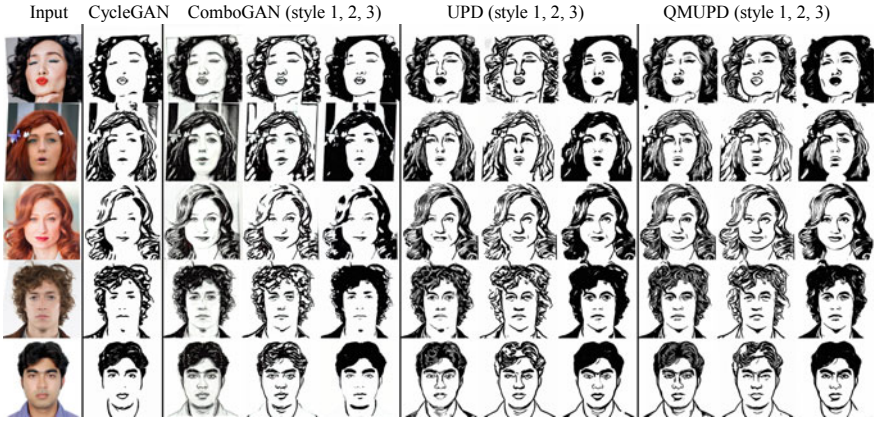

Fig. 8 Quality-Metric-guided Unpaired Portrait line Drawing Generation (QMUPD) qualitative comparisons. From left to right: input face photos, CycleGAN [56] results, ComboGAN [1] results (styles 1, 2, 3), UPD [47] (styles 1, 2, 3), and QMUPD [48] (styles 1, 2, 3). The input face photos are from [32]

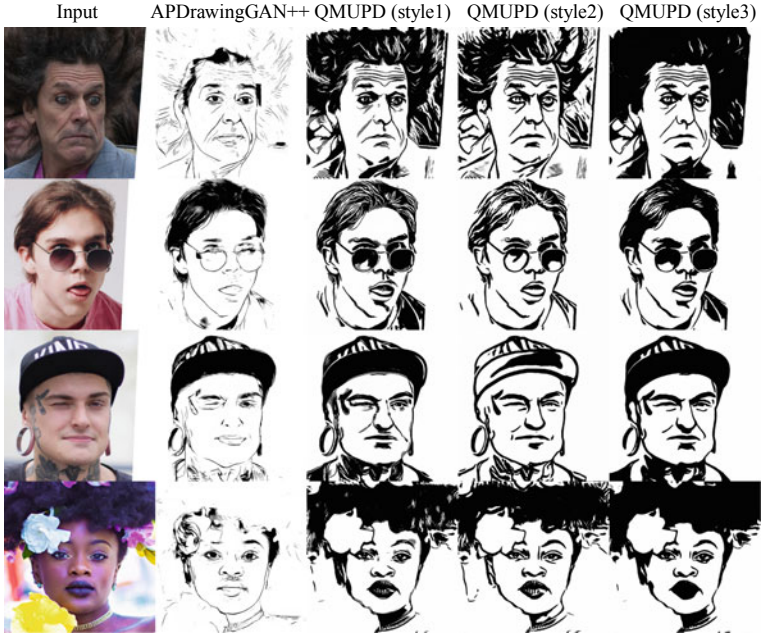

Fig. 9 Quality-Metric-guided Unpaired Portrait line Drawing Generation (QMUPD) qualitative comparisons. From left to right: input face photos, APDrawingGAN++ [50], QMUPD [48] (styles 1, 2, 3). The input face photos are from [32]

3.1 Challenge

In practical applications, the data we have access to are often unpaired. Compared to paired training data, APDrawing generation for learning from unpaired data is more challenging but more relevant. Previous unpaired image-to-image translation methods [52, 56] use a cycle structure to normalize the training. Although cycle consistency loss can be learned from unpaired data, when they are applied to face photo-to-APDrawing translation, due to the apparent imbalance in information richness between the two data types, these methods tend to indiscriminately embed invisible reconstruction information throughout the APDrawing, resulting in degraded quality of the generated APDrawings, such as important facial features are partially lost.

3.2 Quality Metric for APDrawings

For high-quality APDrawing generation, it is not sufficient to decide whether such a drawing is real or fake; the generator needs a quality metric during training for the high-quality synthesized drawing. Observing that humans can easily decide the quality of a portrait line drawing without knowing the original face photo, this section introduces a quality metric for portrait line drawings by learning from human preference, which can then be used to encourage the model to generate good looking portrait line drawings. The metric can be modelled by a regression network to calculate the quality score of each drawing based on human preference and predict the quality score of an APDrawing.

Human preference scores. Human preference scores were obtained based on pairwise comparison between portrait line drawings of the same style. A user study was conducted, where the user was shown three portrait line drawings of the same style in a single question and asked to rank the three drawings. The best of the three shown drawings gets +2 reward score, while the middle placed drawing gets no reward and the worst one gets -2 score. 250 drawings for each of the three target styles were chosen for making the questionnaire and 2450–3450 question responses were collected for each style. After summarizing all question responses for a style, the score for each drawing of this style was calculated and the global ranking was obtained based on the score. Finally, the scores were normalized to the range $[0.1, 1]$ for later steps.

Network architecture. Given the portrait drawing data and the normalized quality score (given by humans), a regression network is trained to predict APDrawing quality. The regression network is based on the Inception v3 [40] architecture. It takes an APDrawing as input and outputs a quality value. Since the quality metric model behaviour is learnt from human evaluation, the predicted score can be used as a constraint item to guide the drawing generator toward better quality.

3.3 Technical Details

The Quality-Metric-guided Unpaired Portrait line Drawing Generation uses a new GAN with an asymmetric cycle structure for face photo to APDrawing conversion without paired training data. Let $\mathcal{P}$ and $\mathcal{D}$ be the face photo domain and the APDrawing domain, and no pairing needs to exist between these two domains. The model uses the training data $S(p) = \{p_i | i = 1, 2, \ldots, N\}$ and $S(d) = \{d_j | j = 1, 2, \ldots, M\}$ to learn a function Φ that maps from $\mathcal{P}$ to $\mathcal{D}$. N and M are the numbers of training photos and APDrawings. The asymmetric cyclic mapping model consists of two generators—a generator G that converts face photos to portrait drawings and an inverse generator F that converts drawings back to face photos—and two discriminators, $D_{\mathcal{D}}$ for discriminating generated drawings from real drawings and $D_{\mathcal{P}}$ for discriminating generated photos from real photos.

3.3.1 Face Photo to Drawing Generator G

The generator G takes a face photo p and a style feature s as input and outputs a portrait line graph $G(p, s)$ with a style specified by s.

Style features. A classifier C (based on VGG19) was used to classify the portrait line drawings into three styles, using the network drawing data labelled with style classes. Then, the output of a final fully connected layer and a softmax layer were used to compute a 3-dimensional vector as a style feature for each drawing (including unlabelled ones).

Network structure. G is an encoder-decoder with a residual block [15] in the middle. It starts with a flat convolution (the feature map size stays the same) and two down convolution blocks to encode the face photos and extract useful features. The stylized features are then mapped as 3-channel feature maps and inserted into the network by concatenating with the feature maps of the second lower convolution block. Additional flat convolution is used to merge the style feature maps with the extracted feature maps. After that, the remaining blocks of nine identical structures are used to construct content features and transfer them to the target domain. Then, the output drawing is reconstructed by two upward convolutional blocks and a final convolutional layer.

3.3.2 Drawing Discriminator $D_{\mathcal{D}}$

The drawing discriminator $D_{\mathcal{D}}$ has two tasks: (1) to distinguish the generated portrait line drawings from the real ones, and (2) to classify a drawing into three selected styles, where the real one d is expected to be assigned to the correct style label (given by C) and the generated one $G(p, s)$ is expected to be assigned to the style specified by the 3-dimensional style feature s.

For the first task, to ensure the presence of important facial features in the generated drawings, in addition to the discriminator D that analyzes the whole drawing, three local discriminators D_{ln}, D_{le}, D_{ll} are used here to discriminate the drawing of nose, eyes and lips, respectively. The inputs of these local discriminators are masked drawings where the mask is obtained from the face resolution network [14]. The $D_{\mathcal{D}}$ consists of D, D_{ln}, D_{le}, D_{ll}.

Network structure. The global discriminator D is based on PatchGAN [18] and modified to have two branches. These two branches share three down convolution blocks. The branch D_{rf} includes two down convolution blocks to output the true/false prediction maps for each patch in the drawing. And the other classification branch D_{cls} includes more down convolution blocks to output the probability values of the three style labels. The local discriminators D_{ln}, D_{le}, D_{ll} also use the PatchGAN structure.

3.3.3 Drawing to Face Photo Generator F and Photo discriminator $D_{\mathcal{P}}$

The generator F in the inverse direction takes a portrait line drawing d as input and outputs a face photo $F(d)$. It uses an encoder-decoder architecture with nine remaining blocks in the middle. The photo discriminator $D_{\mathcal{P}}$ discriminates generated face photos from real ones and also adopts the PatchGAN structure.

3.3.4 Loss Functions

There are six types of losses in Quality-Metric-guided Unpaired Portrait line Drawing Generation model training.

Adversarial loss. The adversarial loss evaluates the ability of the discriminator $D_{\mathcal{D}}$ to assign correct labels to real and synthesized drawings. It is formulated as:

$$\begin{aligned} L_{adv}(G, D_{\mathcal{D}}) = & \sum_{D \in D_{\mathcal{D}}} \mathbb{E}_{d \in S(d)}[\log D(d)] \\ & + \sum_{D \in D_{\mathcal{D}}} \mathbb{E}_{p \in S(p)}[\log(1 - D(G(p, s))] \end{aligned} \tag{8}$$

where s is randomly chosen from the style features of the drawings in $S(d)$ for each p. Since $D_{\mathcal{D}}$ maximizes this loss and G minimizes it, this loss drives the generated drawings closer to real drawings.

An adversarial loss for the photo discriminator $D_{\mathcal{P}}$ and the inverse mapping F is:

$$\begin{aligned} L_{adv}(F, D_{\mathcal{P}}) = & \mathbb{E}_{p \in S(p)}[\log D_{\mathcal{P}}(p)] \\ & + \mathbb{E}_{d \in S(d)}[\log(1 - D_{\mathcal{P}}(F(d))] \end{aligned} \tag{9}$$

Relaxed forward cycle-consistency loss. As mentioned earlier, there is much less information in the domain $\mathcal{D}$ than in the domain $\mathcal{P}$. It is infeasible for $p \rightarrow G(p, s) \rightarrow F(G(p, s))$ to be pixel-wise similar to p, but the edge information in p and $F(G(p, s))$ needs to be similar, which is achievable. Edges are extracted from p and $F(G(p, s))$ using HED [46], and the similarity of edges is evaluated by the LPIPS perceptual metric [55]. Using H to denote the HED and L_{lpips} to denote the perceptual metric, the relaxed cycle consistency loss is formulated as:

$$L_{relaxed-cyc}(G, F) = \mathbb{E}_{p \in S(p)}[L_{lpips}(H(p), H(F(G(p, s))))] \tag{10}$$

Strict backward cycle-consistency loss. On the other hand, the information in the generated face photo is sufficient to reconstruct the drawing. Therefore, it is important that $d \rightarrow F(d) \rightarrow G(F(d), s(d))$ is pixelwise similar to d, where the style feature $s(d)$ is the style feature of d. The strict cycle consistency loss in the backward cycle is then formulated as:

$$L_{strict-cyc}(G, F) = \mathbb{E}_{d \in S(d)}[||d - G(F(d), s(d))||_1] \tag{11}$$

Truncation loss. The truncation loss is designed to prevent the generated drawing from hiding information in small values. It has the same format as the relaxed cycle-consistency loss, except that the generated drawing $G(p, s)$ is first truncated to 6 bits (a general digital image stores intensity in 8 bits) to ensure that the encoded information is clearly visible, and then fed into F to reconstruct the photo. Denoting the truncation operation as $T[\cdot]$, the truncation loss is formulated as:

$$L_{trunc}(G, F) = \mathbb{E}_{p \in S(p)}[L_{lpips}(H(p), H(F(T[G(p, s)])))] \tag{12}$$

During the first training period, the weight for the truncation loss is kept low, otherwise it would be too hard for the model to optimize. The weight is gradually increased as the training progresses.

Style loss. Style loss is introduced to help G generate multiple styles with different style properties. Denoting the classification branch in $D_{\mathcal{D}}$ as D_{cls}, the style loss is formulated as:

$$\begin{aligned} L_{cls}(G, D_{\mathcal{D}}) = \mathbb{E}_{d \in S(d)}[-\sum_{c} p(c) \log D_{cls}(c|d)] \\ + \mathbb{E}_{p \in S(p)}[-\sum_{c} p'(c) \log D_{cls}(c|G(p, s))] \end{aligned} \tag{13}$$

For a real drawing d, $p(c)$ is the probability of the style label c given by the classifier C, and $D_{cls}(c|d)$ is the maximum softmax probability of D_{cls} prediction for c. The probability $p(c)$ is used to account for real drawings that may not belong to a single style but lie between two styles, *e.g.,* the softmax probability [0.58, 0.40, 0.02]. For the generated drawing $G(p, s)$, $p'(c)$ denotes the probability on the style label c, specified by the style feature s, and $D_{cls}(c|G(p, s))$ is the softmax probability

predicted on c. This classification loss motivates D_{cls} to classify the drawing into the correct style and motivates G to generate a drawing close to the given style feature.

Quality loss based on the quality metric model. The quality loss is designed for generating high quality APDrawings. The quality metric model M gives a quality score ($\in [0.1, 1]$) of an APDrawing about how consistent it is with human perception, where better looking drawings get higher prediction scores. The quality loss $L_{quality}$ is then defined as:

$$L_{quality}(G) = \mathbb{E}_{p \in S(p)}[1 - M(G(p, s))]. \tag{14}$$

4 Recent Developments

Multi-scale methods are popular for finding the relation between the source domain and the target domain. For face photo to sketch transfer, MvDT [26] relies on a multiview domain translation method to bridge the domain discrepancy between an input test image in the source domain and a collection of images in the target domain, which flexibly integrates a Convolutional Neural Network (CNN) representation with hand-crafted features in an optimal way. Duan et al. [10] propose a multi-scale gradient self-attention residual learning framework for face photo-sketch transformation, and their method utilizes the relationship between features to selectively enhance the characteristics of specific information through self-attention distribution.

3D shapes are used in some works for sketch drawing. Neural Contours [23] is proposed for learning to generate line drawings from 3D models, and the network of Neural Contours incorporates a differentiable module operating on geometric features of the 3D model and an image-based module operating on view-based shape representations. Neural Strokes [22] takes a 3D shape and a viewpoint as input, and outputs a drawing with textured strokes, with variations in stroke thickness, deformation, and color learnt from an artist's style.

Network architecture is another important factor for high-quality generation. Sketch-Transformer [58] contains a multi-scale feature and position encoder for a patch-level feature and position embedding, a self-attention module for capturing long-range spatial dependency, and a multi-scale spatially-adaptive de-normalization decoder for image reconstruction. CA-GAN (Composition-Aided GAN) [53] utilises paired inputs, including a face photo/sketch and the corresponding pixelwise face labels for generating a sketch/photo and stacked CA-GANs (SCA-GANs) to further rectify defects and add compelling details.

Sometimes, we would like the model to be controlled by some conditions. Wang et al. [43] propose a GAN transfer method that depends on the user input sketches. SoftGAN [5] decouples the latent space of portraits into a geometry space and a texture space, therefore it can generate high-quality portrait images with independently controllable geometry and texture attributes.

Recently, with the development of cross-modal methods, text information is embedded into visual generation models. CLIPasso [42] utilises CLIP [27], a joint image and text model, to distill semantic concepts from sketches and images alike, defines a sketch as a set of Bézier curves and uses a differentiable rasteriser to optimise the parameters of the curves directly with respect to a CLIP-based perceptual loss. CLIPasso can generalize to various categories and cope with challenging levels of abstraction while maintaining the semantic visual clues that allow for instance-level and class-level recognition. CLIPascene [41] further improves the CLIP-based method, and converts a given scene image into a sketch using different types of abstraction (precise to loose) and multiple levels of abstraction (detailed to sparse). Chan et al. [4] think that line drawings are encodings of scene information, and they propose a geometry loss to convey 3D shape and a semantic loss to match the CLIP features of a line drawing with its corresponding photograph.

5 Evaluation

Evaluation of artistic portrait drawings, such as those described in this chapter, is obviously important, but it is a challenging task. Whereas tasks such as image classification and object detection have large, annotated benchmark datasets and various natural and effective evaluation metrics (e.g. accuracy), this is largely absent from APDrawings. Probably this is a consequence of the lack of unique ground truth for image stylisation or image generation; even for a single basic style (e.g. APDrawing) there are many possible drawings that would be considered by experts to be acceptable. In contrast, for tasks such as image classification and object detection their ground truth values, at least to a first order approximation, are expected to be unique and well defined. Thus their metrics can rely on simple techniques such as counting the proportion of correct decisions. However, for APDrawings evaluation involves aesthetics, which makes it complex and difficult to measure, and is moreover subjective.

Consequently, a popular approach to performing APDrawing evaluation is to carry our a user study where the task is to assign a rating to an image or indicate a preference between several images (e.g. a two-alternative forced choice). Since carrying out such user studies is time consuming, and is also not exactly repeatable, a recent alternative has become popular: the use of the Frećhet Inception distance (FID) [16] which is computed as the distance between the distribution of Inception feature vectors extracted from two sets of images. Alternatively a single image FID version [35] is available that is applied to internal patch statistics of the image. However, FID does not capture the quality of content preservation achieved by a stylisation, and moreover, is biased [7]. In addition, differences between Inception feature vectors do not always reflect human perception, although retraining the Inception network on an art dataset shows significant improvement [45].

Another approach is the 'deception score' which measures the proportion of stylised images classified by a VGG network as being artworks of the artist for which the stylisation was produced [34]. However, this means that its application is limited to cases in which a style is tightly specified (such that a style classifier can be trained). Also, like FID it ignores content preservation.

The NPRportrait benchmark v1.0 [32] took a different approach for evaluating content preservation. Gender, age, and ethnicity were considered to be basic features to describe faces, and, along with attractiveness, it was expected that good stylisations could preserve these characteristics unless the styles were highly abstracted. The four characteristics were estimated from the source images by a user study and taken as as ground truth, while a subsequent user study estimated the characteristics from stylised versions of the images. The distances between these distributions were taken as an indication of content loss.

NPRportrait v1.0 [32] structured the benchmark images into three levels of difficulty. As the levels increased, elements such as lighting, pose, expression, etc. were less constrained. This enabled the robustness of the stylisation algorithms to be tested by measuring their performance (e.g. according to user studies, FID, etc.) across the three levels.

6 Conclusions

This chapter focuses on the genre of portrait line drawings, and is therefore circumscribed both by medium (typically pen or pencil) and topic (typically human faces, although some artists specialise in non-humans). Nevertheless, the topic has been of interest in various forms over a long history. We describe various approaches to generating line portraits, including early work from the non-photorealistic rendering field, as well as APDrawingGAN and QMUPD, which are deep generative models. Present methods have addressed some of the challenges of line portrait generation, and achieved decent results. APDrawingGAN realises high-quality drawing with several generators for different face regions. QMUPD uses the asymmetric cycle mapping to train the generator with unpaired data, further allowing more diverse line drawing results to be learnt.

A good portrait should do more than just record a person's physical features. An expert artist can also use a portrait to reveal the subject's character, personality, social status, and so on [44]. Furthermore, the artist may be required to present a certain image of the subject, e.g. to idealise (or even caricature) them, send a political message, etc. In fact, traditional portraiture involves many subtleties; on one hand it aimed to provide a generic view of the sitter, but in contrast to this, also to feature the individual's distinguishing characteristics [25]. Also, historical portraits follow many conventions regarding pose, dress, etc. and make use of emblems and symbols to communicate additional information that goes beyond mere likeness [44]. In the other direction, modernist portraitures such as Picasso or Miró involve such distortions and

abstractions that literal likeness is mostly lost, and the artist has to rely on other means to capture the person's identity.

Most of this is currently beyond the capabilities of computerised art. However, strong progress has been made in recent years, and some work has succeeded in capturing or incorporating aspects such as identity [51], semantics and 3D [4], and emotion [8]. We are happy to see more exploration in this area and hope that it will support the development of AI-Generated Content (AIGC).

Acknowledgements Ran Yi was supported by Shanghai Sailing Program (22YF1420300), CCF-Tencent Open Research Fund (RAGR20220121), Young Elite Scientists Sponsorship Program by CAST (2022QNRC001), National Natural Science Foundation of China (62272447), Beijing Natural Science Foundation (L222117).

References

1. Anoosheh, A., Agustsson, E., Timofte, R., & Gool, L. V. (2018). ComboGAN: unrestrained scalability for image domain translation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, pp. 783–790.
2. Baluja, S. (2022). A natural representation of colors with textures. *The Visual Computer, 38*(9), 3267–3278.
3. Berger, I., Shamir, A., Mahler, M., Carter, E., & Hodgins, J. (2013). Style and abstraction in portrait sketching. *ACM Transactions on Graphics (TOG), 32*(4), 55.
4. Chan, C., Durand, F., & Isola, P. (2022). Learning to generate line drawings that convey geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7915–7925).
5. Chen, A., Liu, R., Xie, L., Chen, Z., Su, H., & Yu, J. (2022). SofGAN: A portrait image generator with dynamic styling. *ACM Transactions on Graphics (TOG), 41*(1), 1–26.
6. Chiu, C. C., Lo, Y. H., Lee, R. R., & Chu, H. K. (2015). Tone-and feature-aware circular scribble art. In *Computer Graphics Forum* (Vol. 34, pp. 225–234).
7. Chong, M. J., & Forsyth, D. (2020) Effectively unbiased FID and inception score and where to find them. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6070–6079).
8. Colton, S., Valstar, M. F., & Pantic, M. (2008). Emotionally aware automated portrait painting. In *Proceedings of the 3rd International Conference on Digital Interactive Media in Entertainment and Arts* (pp. 304–311).
9. Cootes, T. F., Edwards, G. J., & Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(6), 681–685.
10. Duan, S., Chen, Z., Wu, Q. J., Cai, L., & Lu, D. (2020). Multi-scale gradients self-attention residual learning for face photo-sketch transformation. *IEEE Transactions on Information Forensics and Security, 16*, 1218–1230.
11. Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)* (pp. 2414–2423).
12. Gooch, B., Reinhard, E., & Gooch, A. (2004). Human facial illustrations: Creation and psychophysical evaluation. *ACM Transactions on Graphics, 23*(1), 27–44.
13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS '14)* (pp. 2672–2680).

14. Gu, S., Bao, J., Yang, H., Chen, D., Wen, F., & Yuan, L. (2019). Mask-guided portrait editing with conditional GANs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3436–3445).
15. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778).
16. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in neural information processing systems* (Vol. 3).
17. Huang, R., Zhang, S., Li, T., & He, R. (2017). Beyond face rotation: Global and local perception GAN for photorealistic and identity preserving frontal view synthesis. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV '17)* (pp. 2439–2448).
18. Isola, P., Zhu, J., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '17)* (pp. 1125–1134).
19. Kang, H., Lee, S., & Chui, C. K. (2007). Coherent line drawing. In *ACM Symposium Non-photorealistic Animation and Rendering* (pp. 43–50).
20. Li, C., & Wand, M. (2016) Combining markov random fields and convolutional neural networks for image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)* (pp. 2479–2486).
21. Liao, J., Yao, Y., Yuan, L., Hua, G., & Kang, S. B. (2017). Visual attribute transfer through deep image analogy. *ACM Transactions on Graphics (TOG), 36*(4), 120:1–120:15.
22. Liu, D., Fisher, M., Hertzmann, A., & Kalogerakis, E. (2021). Neural strokes: Stylized line drawing of 3d shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 14204–14213).
23. Liu, D., Nabail, M., Hertzmann, A., & Kalogerakis, E. (2020). Neural contours: Learning to draw lines from 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5428–5436).
24. Meng, M., Zhao, M., & Zhu, S. C. (2010). Artistic paper-cut of human portraits. In *Proceedings of the 18th International Conference on Multimedia* (pp. 931–934). ACM.
25. Panofsky, E. (2019). *Early Netherlandish painting: Its origins and character*. Routledge.
26. Peng, C., Wang, N., Li, J., & Gao, X. (2020). Universal face photo-sketch style transfer via multiview domain translation. *IEEE Transactions on Image Processing, 29*, 8519–8534.
27. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning* (pp. 8748–8763). PMLR.
28. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI '15)* pp. 234–241.
29. Rosin, P. L., & Lai, Y. K. (2013). Artistic minimal rendering with lines and blocks. *Graphical Models, 75*(4), 208–229.
30. Rosin, P. L., & Lai, Y. K. (2015). Non-photorealistic rendering of portraits. In *Proceedings of the Workshop on Computational Aesthetics* (pp. 159–170). Eurographics Association (2015).
31. Rosin, P. L., & Lai, Y. K. (2020). *Image-based portrait engraving*. arXiv preprint arXiv:2008.05336.
32. Rosin, P. L., Lai, Y. K., Mould, D., Yi, R., Berger, I., Doyle, L., Lee, S., Li, C., Liu, Y. J., Semmo, A., et al. (2022). NPRportrait 1.0: A three-level benchmark for non-photorealistic rendering of portraits. *Computational Visual Media, 8*(3), 445–465.
33. Rother, C., Kolmogorov, V., & Blake, A. (2004). "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics, 23*(3), 309–314.
34. Sanakoyeu, A., Kotovenko, D., Lang, S., & Ommer, B. (2018). A style-aware content loss for real-time HD style transfer. In *European Conference on Computer Vision* (pp. 698–714).
35. Shaham, T. R., Dekel, T., & Michaeli, T. (2019). SinGAN: Learning a generative model from a single natural image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 4570–4580).

36. Shen, X., Hertzmann, A., Jia, J., Paris, S., Price, B., Shechtman, E., & Sachs, I. (2016). Automatic portrait segmentation for image stylization. *Computer Graphics Forum, 35*(2), 93–102.
37. Shih, Y., Paris, S., Barnes, C., Freeman, W. T., & Durand, F. (2014). Style transfer for headshot portraits. *ACM Transactions on Graphics (TOG), 33*(4), 148:1–148:14.
38. Son, M., Lee, Y., Kang, H., & Lee, S. (2011). Structure grid for directional stippling. *Graphical Models, 73*(3), 74–87.
39. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, 15*(1), 1929–1958.
40. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2818–2826).
41. Vinker, Y., Alaluf, Y., Cohen-Or, D., & Shamir, A. (2022). *CLIPascene: Scene sketching with different types and levels of abstraction*. arXiv preprint arXiv:2211.17256.
42. Vinker, Y., Pajouheshgar, E., Bo, J. Y., Bachmann, R. C., Bermano, A. H., Cohen-Or, D., Zamir, A., & Shamir, A. (2022). CLIPasso: Semantically-aware object sketching. *ACM Transactions on Graphics (TOG), 41*(4), 1–11.
43. Wang, S.Y., Bau, D., & Zhu, J. Y. (2021). Sketch your own GAN. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 14050–14060).
44. West, S. (2004). *Portraiture*. Oxford: Oxford University Press.
45. Wright, M., & Ommer, B. (2022). ArtFID: Quantitative evaluation of neural style transfer. In *Proceedings of 44th DAGM German Conference* (pp. 560–576).
46. Xie, S., & Tu, Z. (2015). Holistically-nested edge detection. In *IEEE International Conference on Computer Vision (ICCV)* (pp. 1395–1403).
47. Yi, R., Liu, Y., Lai, Y., & Rosin, P. L. (2020). Unpaired portrait drawing generation via asymmetric cycle mapping. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 8214–8222).
48. Yi, R., Liu, Y., Lai, Y., & Rosin, P. L. (2023). Quality metric guided portrait line drawing generation from unpaired training data. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45*(1), 905–918.
49. Yi, R., Liu, Y. J., Lai, Y. K., & Rosin, P. L. (2019). APDrawingGAN: Generating artistic portrait drawings from face photos with hierarchical GANs. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 10743–10752).
50. Yi, R., Xia, M., Liu, Y., Lai, Y., & Rosin, P. L. (2021). Line drawings for face portraits from photos using global and local structure based GANs. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*(10), 3462–3475.
51. Yi, R., Ye, Z., Fan, R., Shu, Y., Liu, Y. J., Lai, Y. K., & Rosin, P. L. (2022). Animating portrait line drawings from a single face photo and a speech signal. In *ACM SIGGRAPH 2022 Conference Proceedings* (pp. 1–8).
52. Yi, Z., Zhang, H. R., Tan, P., & Gong, M. (2017). DualGAN: Unsupervised dual learning for image-to-image translation. In *IEEE International Conference on Computer Vision (ICCV)* (pp. 2868–2876).
53. Yu, J., Xu, X., Gao, F., Shi, S., Wang, M., Tao, D., & Huang, Q. (2020). Toward realistic face photo-sketch synthesis via composition-aided GANs. *IEEE Transactions on Cybernetics, 51*(9), 4350–4362.
54. Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters, 23*(10), 1499–1503.
55. Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 586–595).
56. Zhu, J., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)* (pp. 2242–2251).

57. Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV '17)* (pp. 2223–2232).
58. Zhu, M., Liang, C., Wang, N., Wang, X., Li, Z., & Gao, X. (2021). A sketch-transformer network for face photo-sketch synthesis. In *IJCAI* (pp. 1352–1358).