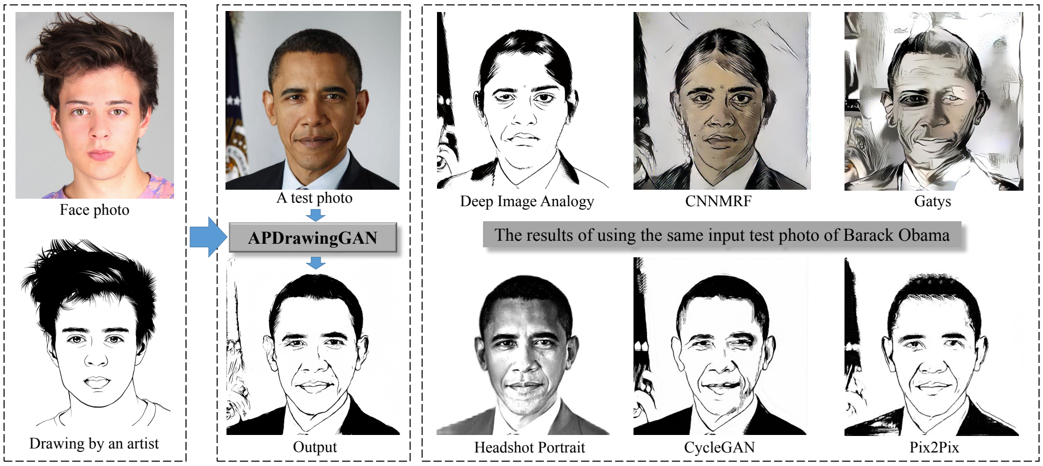
APDrawingGAN: Generating Artistic Portrait Drawings from Face Photos with Hierarchical GANs

Ran Yi, Yong-Jin Liu*
CS Dept, BNRist
Tsinghua University, China
{yr16,liuyongjin}@tsinghua.edu.cn

Yu-Kun Lai, Paul L. Rosin
School of Computer Science and Informatics
Cardiff University, UK
{LaiY4,RosinPL}@cardiff.ac.uk



(a) An image pair in the training set

(b) Our method

(c) Existing methods

Figure 1: (a) An artist draws a portrait drawing using a sparse set of lines and very few shaded regions to capture the distinctive appearance of a given face photo. (b) Our APDrawingGAN learns this artistic drawing style and automatically transforms a face photo into a high-quality artistic portrait drawing. (c) Using the same input face photo, six state-of-the-art style transfer methods cannot generate desired artistic drawings: Deep Image Analogy [20], CNNMRF [18], Gatys [11] and Headshot Portrait [32] change facial features or fail to capture style, CycleGAN [40] and Pix2Pix [15] produce false details around hair, eyes or corners of the mouth.

Abstract

Significant progress has been made with image stylization using deep learning, especially with generative adversarial networks (GANs). However, existing methods fail to produce high quality artistic portrait drawings. Such drawings have a highly abstract style, containing a sparse set of continuous graphical elements such as lines, and so small artifacts are more exposed than for painting styles. Moreover, artists tend to use different strategies to draw different facial features and the lines drawn are only loosely related to obvious image features. To address these challenges, we propose APDrawingGAN, a novel GAN based architecture that builds upon hierarchical generators and discriminators combining both a global network (for images as a whole) and local networks (for individual facial regions). This allows dedicated drawing strategies to be learned for different facial features. Since artists' drawings may not have lines perfectly aligned with image features, we develop a novel loss to measure similarity between generated and artists' drawings based on distance transforms, leading to improved strokes in portrait drawing. To train APDrawingGAN, we construct an artistic drawing dataset containing high-resolution portrait photos and corresponding professional artistic drawings. Extensive experiments, and a user study, show that APDrawingGAN produces significantly better artistic drawings than state-of-the-art methods.

1. Introduction

Portrait drawings are a longstanding and distinct art form, which typically use a sparse set of continuous graph-

*Corresponding author

ical elements (e.g., lines) to capture the distinctive appearance of a person. They are drawn in the presence of the person or their photo, and rely on a holistic approach of observation, analysis and experience. An artistic portrait drawing should ideally capture the personality and the feelings of the person. Even for an artist with professional training, it usually requires several hours to finish a good portrait (Fig. 1a).

Training a computer program with artists' drawings and automatically transforming an input photo into high-quality artistic drawings is much desired. In particular, with the development of deep learning, *neural style transfer* (NST), which uses CNNs to perform image style transfer was proposed [11]. Later on, *generative adversarial network* (GAN) based style transfer methods (e.g., [15, 40, 2, 5]) have achieved especially good results, by utilizing sets of (paired or unpaired) photos and stylized images for learning. These existing methods are mostly demonstrated using cluttered styles, which contain many fragmented graphical elements such as brush strokes, and have a significantly lower requirement for the quality of individual elements (i.e., imperfections are much less noticeable).

Artistic portrait drawings (APDrawings) are substantially different in style from portrait painting styles studied in previous work, mainly due to the following five aspects. First, the APDrawing style is highly abstract, containing a small number of sparse but continuous graphical elements. Defects (such as extra, missing or erroneous lines) in APDrawings are much more visible than other styles such as paintings (e.g., impressionist and oil painting) involving a dense collection of thousands of strokes of varying sizes and shapes. Second, there are stronger semantic constraints for APDrawing style transfer than for general style transfer. In particular, facial features should not be missing or displaced. Even small artifacts (e.g., around the eye) can be clearly visible, distracting and unacceptable. Third, the rendering in APDrawings is not consistent between different facial parts (e.g., eyes vs. hair). Fourth, the elements (e.g. the outline of facial parts) in APDrawings are not precisely located by artists, posing a challenge for methods based on pixel correspondence (e.g., Pix2Pix [15]). Finally, artists put lines in APDrawings that are not directly related to low level features in the view or photograph of the person. Examples include lines in the hair indicating the flow, or lines indicating the presence of facial features even if the image contains no discontinuities. Such elements of the drawings are hard to learn. Therefore, even state-of-the-art image style transfer algorithms (e.g., [11, 15, 18, 20, 32, 40]) often fail to produce good and expressive APDrawings. See Fig. 1c for some examples.

To address the above challenges, we propose APDrawingGAN, a novel Hierarchical GAN architecture dedicated to face structure and APDrawing styles for transforming face photos to high-quality APDrawings (Fig. 1b). To effectively learn different drawing styles for different facial regions, our GAN 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, we propose a novel distance transform (DT) loss to learn stroke lines in APDrawings.

The main contributions of our work are three-fold:

- We propose a Hierarchical GAN architecture for artistic portrait drawing synthesis from a face photo, which can generate high-quality and expressive artistic portrait drawings. In particular, our method can learn complex hair style with delicate white lines.
- Artists use multiple graphical elements when creating a drawing. In order to best emulate artists, our model separates the GAN's rendered output into multiple layers, each of which is controlled by separated loss functions. We also propose a loss function dedicated to APDrawing with four loss terms in our architecture, including a novel DT loss (to promote line-stroke based style in APDrawings) and a local transfer loss (for local networks to preserve facial features).
- We pre-train our model using 6,655 frontal face photos collected from ten face datasets, and construct an APDrawing dataset (containing 140 high-resolution face photos and corresponding portrait drawings by a professional artist) suitable for training and testing. The APDrawing dataset and code is available.[1]

2. Related Work

Image stylization has been widely studied in non-photorealistic rendering and deep learning research. Below we summarize related work in three aspects.

2.1. Style transfer using neural networks

Gatys et al. [11] first proposed an NST method using a CNN to transfer the stylistic characteristics of a style image to a content image. For a given image, its content and style features are represented by high layer features and texture information captured by Gram matrices [10] in a VGG network, respectively. Style transfer is achieved by optimizing an image to match both the content of the content image and the style of the style image. This method performs well on oil painting style transfer of various artists. However, their style is modeled as texture features, and thus not suitable for our target style with little texture.

Li and Wand [18] used a Markov Random Field (MRF) loss instead of the Gram matrix to encode the style, and proposed the combined MRF and CNN model (CNNMRF).

[1] https://cg.cs.tsinghua.edu.cn/people/~Yongjin/Yongjin.htm

CNNMRF can be applied in both non-photorealistic (artwork) and photo-realistic image synthesis, since local patch matching is used in MRF loss and promotes local plausibility. However, local patch matching restricts this method to only work well when the style and content images contain elements of similar local features.

Liao et al. [20] proposed Deep Image Analogy for visual attribute transfer by finding semantically meaningful dense correspondences between two input images. They compute correspondence between feature maps extracted by a CNN. Deep Image Analogy was successfully applied to photo-to-style transfer, but when transferring APDrawing style, image content is sometimes affected, making subjects in the resulting images less recognizable.

Johnson et al. [16] proposed the concept of perceptual-loss-based on high-level features and trained a feed forward network for image style transfer. Similar to [11], their texture-based loss function is not suitable for our style.

In addition to aforementioned limitations for APDrawing style transfer, most existing methods require the style image to be close to the content image.

2.2. Non-photorealistic rendering of portraits

In the field of NPR, many methods have been developed for generating portraits [29]. Rosin and Lai [28] proposed a method to stylize portraits using highly abstracted flat color regions. Wang et al. [38] proposed a learning-based method to stylize images into portraits which are composed of curved brush strokes. Berger et al. [3] proposed a data-driven approach to learn the portrait sketching style, by analyzing strokes and geometric shapes in a collection of artists' sketch data. Liang et al. [19] proposed a method for portrait video stylization by generating a facial feature model using extended Mask R-CNN and applying two stroke rendering methods on sub-regions. The above methods generate results of a specific type of art, e.g., curved brush stroke portrait, portrait sketching. However, none of them study the style of artistic portrait drawing.

There are also some example-based stylization methods designed for portraits. Selim et al. [30] proposed a portrait painting transfer method by adding spatial constraints into the method [11] to reduce facial distortion. Fišer et al. [9] proposed a method for example-based stylization of portrait videos by designing several guiding channels and applying the guided texture synthesis method in [8]. However, all these methods use similar texture synthesis approaches that make them unsuitable for the APDrawing style.

2.3. GAN-based image synthesis

Generative Adversarial Networks (GAN) [12] have achieved much progress in solving many image synthesis problems, in which closely related to our work are Pix2Pix and CycleGAN.

Pix2Pix [15] is a general framework for image-to-image translation, which explores GANs in a conditional setting [22]. Pix2Pix can be applied to a variety of image translation tasks and achieves impressive results on various tasks including semantic segmentation, colorization and sketch to photo translation, etc.

CycleGAN [40] is designed to learn translation between two domains without paired data by introducing cycle-consistency loss. This model is particularly suitable for tasks in which paired training data are not available. When applied to a dataset with paired data, this method produces results similar to the fully supervised Pix2Pix, but with much more training time.

Neither Pix2Pix nor CycleGAN works well for APDrawing styles and often generates blurry or messy results due to the five challenges summarized in Sec. 1 for APDrawings.

3. Overview of APDrawingGAN

We model the process of learning to transform face photos to APDrawings 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 learned from paired training data $S_{data} = \{(p_i, a_i) | p_i \in \mathcal{P}, a_i \in \mathcal{A}, i = 1, 2, ..., N\}$, where N is the number of photo-APDrawing pairs in the training set.

Our model is based on the GAN framework, consisting of a generator G and a discriminator D, both of which are CNNs specifically designed for APDrawings with line-stroke-based artist drawing style. The generator G learns to output an APDrawing in $\mathcal{A}$ while the discriminator D learns to determine whether an image is a real APDrawing or generated.

Since our model is based on GANs, the discriminator D is trained to maximize the probability of assigning the correct label to both real APDrawings $a_i \in \mathcal{A}$ and synthesized drawings $G(p_i)$, $p_i \in \mathcal{P}$, and simultaneously G is trained to minimize this probability. Denote the loss function as $L(G, D)$, which is specially designed to include four terms $L_{adv}(G, D)$, $L_{\mathcal{L}_1}(G, D)$, $L_{DT}(G, D)$ and $L_{local}(G, D)$. Then the function Ψ can be formulated by solving the following min-max problem with the function $L(G, D)$:

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

In Sec. 4, we introduce the architecture of APDrawingGAN. The four terms in $L(G, D)$ are presented in Sec. 5. Finally, we present the training scheme in Sec. 6. An overview of our APDrawingGAN is illustrated in Fig. 2.

4. APDrawingGAN Architecture

Unlike the standard GAN architecture, here we propose a hierarchical structure for both generator and discrimina-

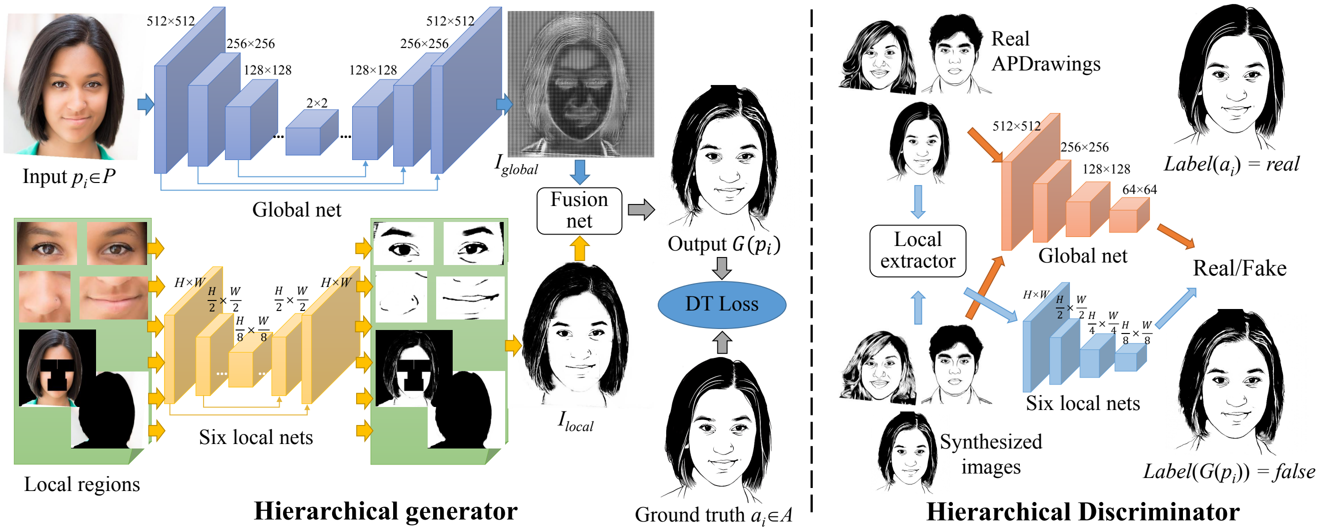


Figure 2: The framework of the proposed APDrawingGAN. The hierarchical generator G takes a face photo $p_i \in \mathcal{P}$ as input and can be decomposed into a global network (for global facial structure), six local networks (for four local facial regions, the hair and the background region) and a fusion network. Outputs of six local nets are combined into I_{local} and fused with the output I_{global} of the global network to generate the final output $G(p_i)$. The loss function includes four terms, in which a novel *DT loss* is introduced to better learn delicate artistic line styles. The hierarchical discriminator D distinguishes whether the input is a real APDrawing or not based on the classification results by combining both a global discriminator and six local discriminators.

tor, each of which includes a global network and six local networks. The six local networks correspond to the local facial regions of the left eye, right eye, nose, mouth, hair and the background. Furthermore, the generator has an additional fusion network to synthesize the artistic drawings from the output of global and local networks. The reason behind this hierarchical structure is that in portrait drawing, artists adopt different drawing techniques for different parts of the face. For example, fine details are often drawn for eyes, and curves drawn for hair usually follow the flow of hair but do not precisely correspond to image intensities. Since a single CNN shares filters across all locations in an image and is very difficult to encode/decode multiple drawing features, the design of hierarchical global and local networks with multiple CNNs can help the model better learn facial features in different locations.

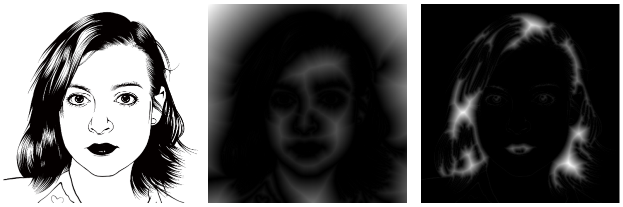
4.1. Hierarchical generator G

The generator G transforms input face photos to APDrawings. The style of APDrawings is learned once the model is trained. In the hierarchy of $G = \{G_{global}, G_{l*}, G_{fusion}\}$, 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, and G_{fusion} is a fusion network.

We design G using the U-Net structure [26]. Each of $G_{l\_eye\_l}$, $G_{l\_eye\_r}$, G_{l_nose} and G_{l_mouth} is a U-Net with three down-convolution and three up-convolution blocks. Each of G_{l_hair} and G_{l_bg} is a U-Net with four down-convolution and four up-convolution blocks. The role of local generators in G_{l*} is to learn the drawing style of different local face features; e.g., hairy style for hair (i.e., repeated wispy details by short choppy or long strokes to capture the soft wispiness of individual hair strands), delicate line style for eyes and nose, and solid or line style for mouth. A U-Net with skip connections can incorporate multi-scale features and provide sufficient but not excessive flexibility to learn artists' drawing techniques in APDrawings for different facial regions.

The inputs to $G_{l\_eye\_l}$, $G_{l\_eye\_r}$, G_{l_nose}, G_{l_mouth} are local regions centered at the facial landmarks (i.e., left eye, right eye, nose and mouth) obtained by the MTCNN model [39]. The input to G_{lbg} is the background region detected by a portrait segmentation method [31]. The input to G_{hair} is the remaining region in the face photo. We blend outputs of all local generators into an aggregated drawing I_{local}, by using the min pooling at overlapping regions. This min pooling can effectively retain responses from individual local generators, as low intensities are treated as responses for black pixels in artistic drawings.

G_{global} is a U-Net with eight down-convolution and eight up-convolution blocks, which deals with the global structure of the face. G_{fusion} consists of a flat convolution block, two residual blocks and a final convolution layer. We use G_{fusion} to fuse together I_{local} and I_{global} (i.e., the out-



(a) An APDrawing x (b) $I_{DT}(x)$ (c) $I'_{DT}(x)$

Figure 3: Two distance transforms $I_{DT}(x)$ and $I'_{DT}(x)$ of an APDrawing x.

put of G_{global}) for obtaining the final synthesized drawing of G. In many previous GAN models (e.g., [12, 14]), usually some noise is input or added in the generator network. Following [15], we do not add noise in G explicitly, but use dropout [33] in U-Net blocks to work as noise.

4.2. Hierarchical discriminator D

The discriminator D distinguishes whether the input drawing is a real artist's portrait drawing or not. 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 judge the holistic APDrawing features, while the local discriminators in D_{l*} examine different local regions to evaluate the quality of fine details.

We implement D_{global} and all local discriminators in D_{l*} using the Markovian discriminator in Pix2Pix [15]. 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.

5. Loss Function

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

Adversarial loss L_{adv} models the discriminator's ability to correctly distinguish real or false APDrawings. Following Pix2Pix [15], the adversarial loss is formulated as:

$$L_{adv}(G, D) = \sum_{D_j \in D} \mathbb{E}_{(p_i,a_i)\sim S_{data}}[\log(D_j(p_i, a_i) + \log(1 - D_j(p_i, G(p_i)))]. \quad (2)$$

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

Pixel-wise loss $L_{\mathcal{L}_1}$ drives the synthesized drawings close to ground-truth drawings in a pixel-wise manner. We compute the $L_{\mathcal{L}_1}$ loss for each pixel in the whole drawing:

$$L_{\mathcal{L}_1}(G, D) = \mathbb{E}_{(p_i,a_i)\sim S_{data}}[\|G(p_i) - a_i\|_1] \quad (3)$$

Using the $\mathcal{L}_1$ norm generally outputs less blurry results than the $\mathcal{L}_2$ norm and so is more suitable for APDrawing style.

Line-promoting distance transform loss L_{DT} is a novel measure specially designed for promoting line strokes in the style of APDrawings. Since the elements in APDrawings are not located precisely corresponding to image intensities, we introduce L_{DT} to tolerate the small misalignments — that are often present in artists' portrait drawings — and to better learn stroke lines in APDrawings. To do so, we make use of distance transform (DT) and Chamfer matching as follows.

A DT (a.k.a. distance map) can be represented by a digital image, in which each pixel stores a distance value. Given a real or synthesized APDrawing x, we define two DTs of x as images $I_{DT}(x)$ and $I'_{DT}(x)$: assuming $\hat{x}$ is the binarized image of x, each pixel in $I_{DT}(x)$ stores the distance value to its nearest black pixel in $\hat{x}$ and each pixel in $I'_{DT}(x)$ stores the distance value to its nearest white pixel in $\hat{x}$. Fig. 3 shows an example.

We train two CNNs[2] to detect black and white lines in APDrawings, denoted as Θ_b and Θ_w. The Chamfer matching distance between APDrawings x_1 and x_2 is defined as

$$d_{CM}(x_1, x_2) = \sum_{(j,k)\in\Theta_b(x_1)} I_{DT}(x_2)(j,k) + \sum_{(j,k)\in\Theta_w(x_1)} I'_{DT}(x_2)(j,k) \quad (4)$$

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

$$L_{DT}(G, D) = \mathbb{E}_{(p_i,a_i)\sim S_{data}}[d_{CM}(a_i, G(p_i)) + d_{CM}(G(p_i), a_i)] \quad (5)$$

Local transfer loss L_{local} puts extra constraints on the intermediate output of six local generators in G_{l*}, and then behaves as a regularization term in the loss function. Denote the six local regions of an APDrawing x as $El(x)$, $Er(x)$, $Ns(x)$, $Mt(x)$, $Hr(x)$ and $Bg(x)$. L_{local} is defined as

$$\begin{aligned} &L_{local}(G, D) = \\ &\mathbb{E}_{(p_i,a_i)\sim S_{data}}[||G_{l\_eye\_l}(El(p_i)) - El(a_i)||_1 \\ &+||G_{l\_eye\_r}(Er(p_i)) - Er(a_i)||_1 \\ &+||G_{l\_nose}(Ns(p_i)) - Ns(a_i)||_1 \\ &+||G_{l\_mouth}(Mt(p_i)) - Mt(a_i)||_1 \\ &+||G_{l\_hair}(Hr(p_i)) - Hr(a_i)||_1 \\ &+||G_{l\_bg}(Bg(p_i)) - Bg(a_i)||_1] \end{aligned} \quad (6)$$

[2] We use two-tone NPR images and the corresponding lines generated by the NPR algorithm [27] as data to train the two CNN models.

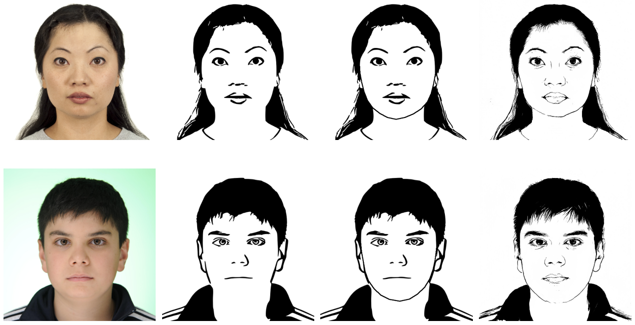


Figure 4: From left to right: original face photos, NPR results [27], NPR results adding clear jaw contours (used for pre-training) and the results of APDrawingGAN. Face photos are from the datasets of CFD [21] and Siblings [36].

6. Training APDrawingGAN

APDrawing dataset. To train the proposed APDrawingGAN, we build a dataset containing 140 pairs of face photos and corresponding portrait drawings. To make the training set distribution more consistent, all portrait drawings were drawn by a single professional artist. All images and drawings are aligned and cropped to 512 × 512 size. Some examples are illustrated in supplemental material.

Initialization with pre-training. Since it is time-consuming and laborious for an artist to draw each portrait drawing, our constructed dataset consists of only a small number of image pairs, which makes the training particularly challenging. To address this issue, we use a coarse-level pre-training to make the training starting at a good initial status. We collect 6,655 frontal face photos taken from ten face datasets [37, 21, 6, 25, 24, 7, 35, 34, 4, 36]. For each photo, we generate a synthetic drawing using the two-tone NPR algorithm in [27]. Since it often generates results without clear jaw lines (due to low contrast in the image at these locations), we use the face model in OpenFace [1] to detect the landmarks on the jaws and subsequently add jaw lines to the NPR results. Two examples are illustrated in Fig. 4. Note that the drawings synthesized in this simple way are only a coarse approximation and still far from ideal APDrawings. We use a pre-trained model after 10 epochs as the initialization for the subsequent formal training. Since our NPR generated drawings (unlike artists' drawings) are accurately aligned to the photos, we do not use the distance transform loss in pre-training.

Formal training. We partition our APDrawing dataset into a training set of 70 image pairs and a test set of 70 image pairs. Then we apply data augmentation of small-angle rotation (-10°∼10°) and scaling (1∼1.1) to the training set. Furthermore, we apply the Adam optimizer [17] with learning rate 0.0002 and momentum parameters $\beta_1 = 0.5, \beta_2 = 0.999$ and batch size of 1.

7. Experiments

We implemented APDrawingGAN in PyTorch [23] and conducted experiments on a computer with an NVIDIA Titan Xp GPU. The input and output of the generator G are color photos and gray drawings, respectively, and so the numbers of input and output channels are 3 and 1. In all our experiments, the parameters in Eq. 1 are fixed at $\lambda_1 = 100$, $\lambda_2 = 0.1$, $\lambda_3 = 25$. All the evaluation results presented in this section are based on the test set to ensure fairness.

7.1. Ablation study in APDrawingGAN

We perform an ablation study on some key factors in APDrawingGAN and the following results show that all of them are essential to APDrawingGAN and they jointly produce high-quality results of APDrawing stylization.

Local networks (i.e., G_{l*} and D_{l*}) in APDrawingGAN are essential to capture the style of each facial region. Since the style of an APDrawing contains several independent rendering techniques in different local regions, without local networks, the model cannot learn the varying styles well with a location-independent fully convolutional network. As shown in Fig. 5, without local networks, the model generates messy results, where both facial region and hair region exhibit messy hairy style, leading to obvious defects.

Line-promoting DT loss L_{DT} is essential to produce good and clean results with delicate lines. Without the DT loss, there are fewer delicate lines in the hair region and some undesirable white patches appear instead, as shown in the second row in Fig. 5. Moreover, some unattractive lines appear around the jaw, leading to drawings unlike the input photo, as shown in both results in Fig. 5. These lines are effectively avoided by using the DT loss.

Initialization using the model pre-trained on the NPR data helps the model to generate good results in less time. The results without initialization are worse in having more messy lines in the facial region and fewer delicate white lines in the hair region, as shown in the chin region of both results and hair region of the second result in Fig. 5. The pre-training helps the model to quickly converge to a good result, avoiding such artifacts.

7.2. Comparison with state-of-the-art

We compare APDrawingGAN with six state-of-the-art style transfer methods: Gatys [11], CNNMRF [18], Deep Image Analogy [20], Pix2Pix [15], CycleGAN [40] and Headshot Portrait [32]. Since the input to Gatys (with average Gram matrix), CycleGAN and Pix2Pix is different from the input to CNNMRF, Deep Image Analogy and Headshot Portrait, we compare them separately.

Qualitative results of comparison with Gatys, CycleGAN and Pix2Pix are shown in Fig. 6. Gatys' method [11]

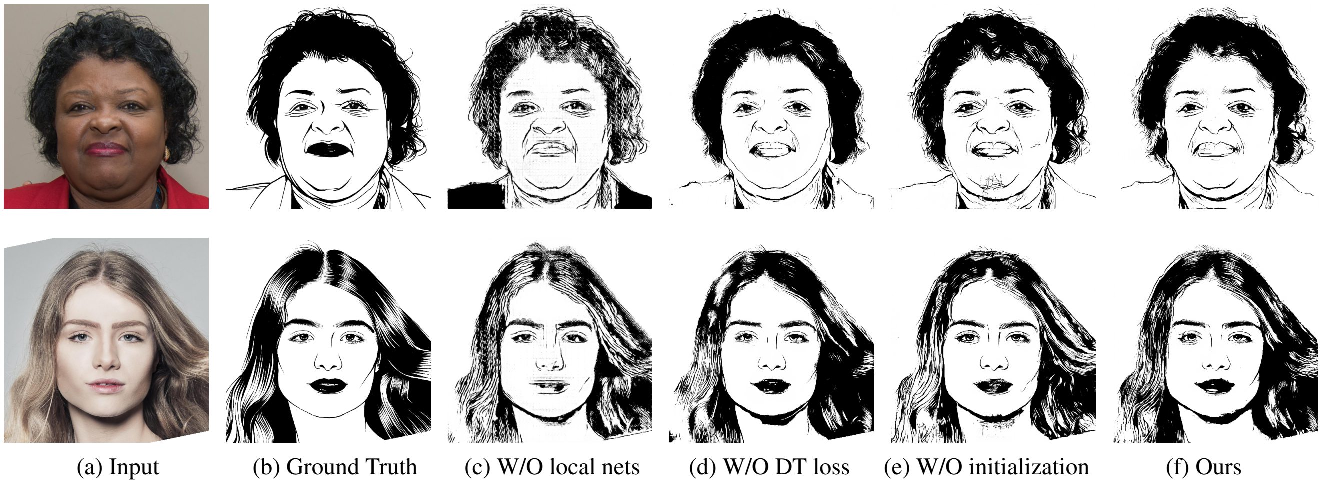


Figure 5: Ablation study: (a) input face photos, (b) ground truth drawings by an artist, (c) results of removing local networks G_{l*} and D_{l*} in APDrawingGAN, (d) results of removing line-promoting DT loss L_{DT} from Eq. 1, (e) results of not using model pre-trained on NPR data as initialization, (f) our results.

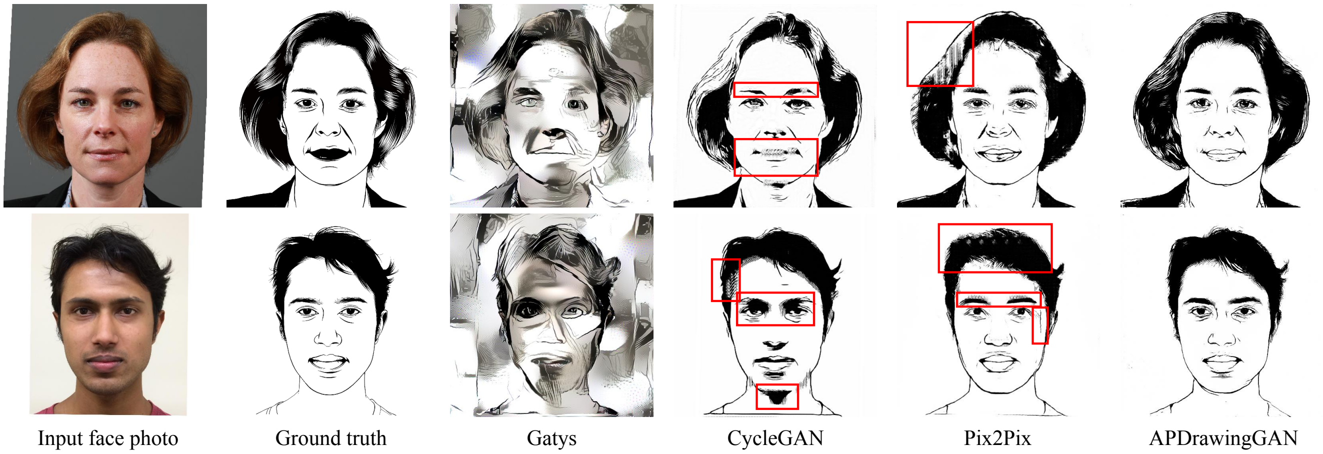


Figure 6: Comparison results with Gatys [11], CycleGAN [40], Pix2Pix [15] and our APDrawingGAN.

by default takes one content image and one style image as input. But for fair comparison, we use all the style images in the training set and compute the average Gram matrix to model the target style as in [40]. As shown in Fig. 6, Gatys' method generates poor results for APDrawing stylization: some facial features are missing in the stylized results, and different regions are stylized inconsistently. The reasons behind these artifacts are that the method models style as texture information in the Gram matrix, which cannot capture our target style with little texture, and its content loss based on VGG output cannot preserve facial features precisely.

CycleGAN [40] also cannot mimic the artistic portrait style well. As shown in Fig. 6, CycleGAN's results do not look like an artist's drawing, especially in the facial features. There are many artifacts, such as missing details in the eyes, blurred/dithered mouth region, dark patches (e.g. the eyes and chin in the bottom row) caused by shadows, not capturing eyebrow style. CycleGAN is unable to preserve facial features because it uses the cycle-consistency to constrain content, which is less accurate than a supervised method and leads to problems when one of the domains is not accurately recovered.

Pix2Pix [15] generates results that preserve some aspect of artistic drawings, but they also have many artifacts. There are many messy unwanted lines, making the stylized result unlike the input photo, and the white lines in the hair are not learned well. The reason is that a generator with one CNN is unable to learn several independent drawing techniques in different facial regions, and there is no specifically designed loss term dedicated to the APDrawing style.

In comparison, our method captures the different drawing techniques in different facial regions well and generates

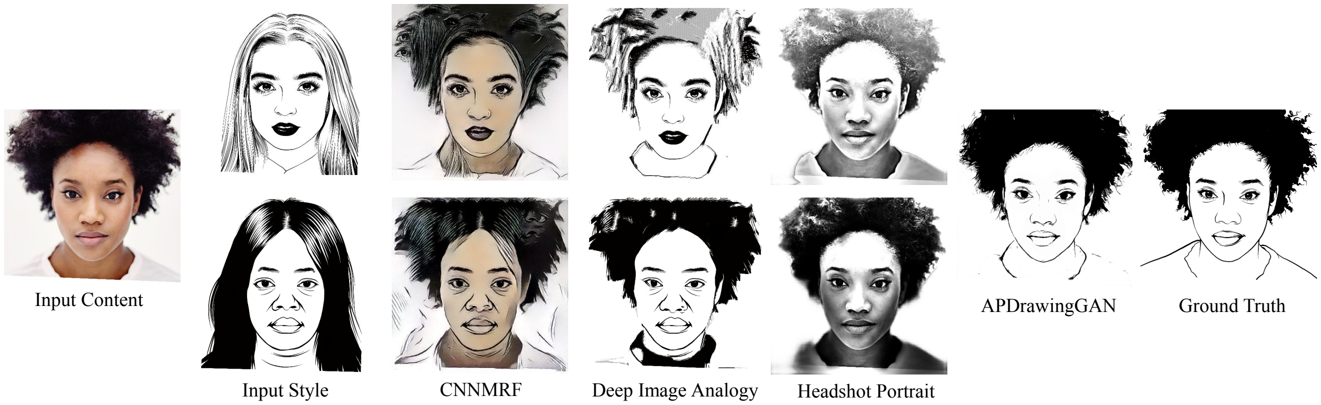


Figure 7: Comparison results with CNNMRF [18], Deep Image Analogy [20], Headshot Portrait [32] and APDrawingGAN.

high-quality results with delicate white lines in the hair and facial features drawn in the artist's drawing style.

Qualitative results of comparison with CNNMRF, Deep Image Analogy and Headshot Portrait are shown in Fig. 7. These methods take one content image and one style image as input, and require the two images to be similar. Given a content image in the test set, we select two style images in the training set that are semantically similar to the content image (i.e. they have similar facial features) as shown in Fig. 7. CNNMRF [18] generates results that do not exhibit the same color distribution as the target style. Both CNNMRF and Deep Image Analogy [20] generate results with facial features closer to the style image but unlike the input content image, i.e. content has been erroneously copied from the style image. Headshot Portrait [32] is a portrait specific method but it generates photo-realistic results, which are not the style of the target artist's portrait drawing. In comparison, our method generates drawings that both preserve the facial features in the face photo and capture the artistic portrait drawing style. Moreover, our results are high-quality and very close to the ground truth drawn by the artist.

For quantitative evaluation, we compare our APDrawingGAN with CycleGAN [40] and Pix2Pix [15] using the Fréchet Inception Distance (FID) [13], which is a widely used GAN evaluation metric. We evaluate the FID on the full test set to measure the similarity between generated APDrawings and real APDrawings. The comparison results are presented in Table 1. As a reference, we also report the FID metric between the real APDrawings in the training set and the test set. The results show that our method has a much lower FID value, indicating our generated distribution is closer to the real APDrawing distribution than CycleGAN and Pix2Pix.

Due to the subjective nature of image styles, we also conduct a user study to compare our results to CycleGAN and Pix2Pix, which shows that our APDrawingGAN ranks best in 71.39% of cases. More details on user study are presented in the supplementary material.

Table 1: Comparison of CycleGAN, Pix2Pix and our APDrawingGAN in terms of the FID metric. Our method shows a much lower FID value, indicating our generated distribution is closer to real APDrawing distribution than CycleGAN and Pix2Pix.

Methods	FID
CycleGAN [40]	87.82
Pix2Pix [15]	75.30
APDrawingGAN	62.14
Real (training vs test)	49.72

8. Conclusion and Future Work

In this paper, we propose APDrawingGAN, a Hierarchical GAN model to transform a face photo into an APDrawing. Our approach is dedicated to the human face and APDrawing style, and particularly aims to avoid the many artifacts produced by existing methods. Experimental results and a user study show that our method can achieve successful artistic portrait style transfer, and outperforms state-of-the-art methods.

Although our method can learn complex hair style with delicate white lines, the results are still not as clean as the artist's drawings, in hair and lip regions. We plan to address these in future work.

Acknowledgement

This work was supported by the Natural Science Foundation of China (61725204, 61521002), Royal Society-Newton Advanced Fellowship (NA150431) and MOE-Key Laboratory of Pervasive Computing.

References

[1] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. OpenFace: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016. 6

[2] Samaneh Azadi, Matthew Fisher, Vladimir Kim, Zhaowen Wang, Eli Shechtman, and Trevor Darrell. Multi-content GAN for few-shot font style transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '18, pages 7564–7573, 2018. 2

[3] Itamar Berger, Ariel Shamir, Moshe Mahler, Elizabeth Carter, and Jessica Hodgins. Style and abstraction in portrait sketching. *ACM Transactions on Graphics (TOG)*, 32(4):55:1–55:12, 2013. 3

[4] Olga Chelnokova, Bruno Laeng, Marie Eikemo, Jeppe Riegels, Guro Løseth, Hedda Maurud, Frode Willoch, and Siri Leknes. Rewards of beauty: the opioid system mediates social motivation in humans. *Molecular Psychiatry*, 19:746–751, 2014. 6

[5] Yang Chen, Yu-Kun Lai, and Yong-Jin Liu. CartoonGAN: Generative adversarial networks for photo cartoonization. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '18, pages 9465–9474, 2018. 2

[6] Rémi Courset, Marine Rougier, Richard Palluel-Germain, Annique Smeding, Juliette Manto Jonte, Alan Chauvin, and Dominique Muller. The caucasian and north african french faces (CaNAFF): A face database. *International Review of Social Psychology*, 31(1):22:1–22:10, 2018. 6

[7] Natalie C. Ebner, Michaela Riediger, and Ulman Lindenberger. FACES-a database of facial expressions in young, middle-aged, and older women and men: Development and validation. *Behavior Research Methods*, 42(1):351–362, 2010. 6

[8] Jakub Fišer, Ondřej Jamriška, Michal Lukáč, Eli Shechtman, Paul Asente, Jingwan Lu, and Daniel Sỳkora. StyLit: illumination-guided example-based stylization of 3d renderings. *ACM Transactions on Graphics (TOG)*, 35(4):92:1–92:11, 2016. 3

[9] Jakub Fišer, Ondřej Jamriška, David Simons, Eli Shechtman, Jingwan Lu, Paul Asente, Michal Lukáč, and Daniel Sỳkora. Example-based synthesis of stylized facial animations. *ACM Transactions on Graphics (TOG)*, 36(4):155:1–155:11, 2017. 3

[10] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, NeurIPS '15, pages 262–270, 2015. 2

[11] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '16, pages 2414–2423, 2016. 1, 2, 3, 6, 7

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, NeurIPS '14, pages 2672–2680, 2014. 3, 5

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, NeurIPS '17, pages 6629–6640, 2017. 8

[14] Rui Huang, Shu Zhang, Tianyu Li, and Ran He. 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, pages 2439–2448, 2017. 5

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '17, pages 1125–1134, 2017. 1, 2, 3, 5, 6, 7, 8

[16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, ECCV '16, pages 694–711, 2016. 3

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[18] Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '16, pages 2479–2486, 2016. 1, 2, 6, 8

[19] Dongxue Liang, Kyoungju Park, and Przemyslaw Krompiec. Facial feature model for a portrait video stylization. *Symmetry*, 10(10):442, 2018. 3

[20] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual attribute transfer through deep image analogy. *ACM Transactions on Graphics (TOG)*, 36(4):120:1–120:15, 2017. 1, 2, 3, 6, 8

[21] Debbie S. Ma, Joshua Correll, and Bernd Wittenbrink. The Chicago face database: A free stimulus set of faces and norming data. *Behavior Research Methods*, 47(4):1122–1135, 2015. 6

[22] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 3

[23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NeurIPS Workshop*, 2017. 6

[24] Peter Peer, Žiga Emeršič, Jernej Bule, Jerneja Žganec-Gros, and Vitomir Štruc. Strategies for exploiting independent cloud implementations of biometric experts in multibiometric scenarios. *Mathematical Problems in Engineering*, 2014:1–15, 2014. 6

[25] P. Jonathon Phillips, Harry Wechsler, Jeffrey Huang, and Patrick J. Rauss. The FERET database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing*, 16(5):295–306, 1998. 6

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, MICCAI '15, pages 234–241, 2015. 4

[27] Paul L. Rosin and Yu-Kun Lai. Towards artistic minimal rendering. In *International Symposium on Non-Photorealistic Animation and Rendering*, NPAR '10, pages 119–127, 2010. 5, 6

[28] Paul L. Rosin and Yu-Kun Lai. Non-photorealistic rendering of portraits. In *Proceedings of the workshop on Computational Aesthetics*, CAE '15, pages 159–170, 2015. 3

[29] Paul L. Rosin, David Mould, Itamar Berger, John P. Collomosse, Yu-Kun Lai, Chuan Li, Hua Li, Ariel Shamir, Michael Wand, Tinghuai Wang, and Holger Winnemöller. Benchmarking non-photorealistic rendering of portraits. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, NPAR '17, pages 11:1–11:12, 2017. 3

[30] Ahmed Selim, Mohamed A. Elgharib, and Linda Doyle. Painting style transfer for head portraits using convolutional neural networks. *ACM Transactions on Graphics (TOG)*, 35(4):129:1–129:18, 2016. 3

[31] Xiaoyong Shen, Aaron Hertzmann, Jiaya Jia, Sylvain Paris, Brian Price, Eli Shechtman, and Ian Sachs. Automatic portrait segmentation for image stylization. *Computer Graphics Forum*, 35(2):93–102, 2016. 4

[32] Yi-Chang Shih, Sylvain Paris, Connelly Barnes, William T. Freeman, and Frédo Durand. Style transfer for headshot portraits. *ACM Transactions on Graphics (TOG)*, 33(4):148:1–148:14, 2014. 1, 2, 6, 8

[33] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 5

[34] Nina Strohminger, Kurt Gray, Vladimir Chituc, Joseph Heffner, Chelsea Schein, and Titus Brooks Heagins. The MR2: A multi-racial, mega-resolution database of facial stimuli. *Behavior Research Methods*, 48(3):1197–1204, 2016. 6

[35] Carlos Eduardo Thomaz and Gilson Antonio Giraldi. A new ranking method for principal components analysis and its application to face image analysis. *Image and Vision Computing*, 28(6):902–913, 2010. 6

[36] Tiago F. Vieira, Andrea Bottino, Aldo Laurentini, and Matteo De Simone. Detecting siblings in image pairs. *The Visual Computer*, 30(12):1333–1345, 2014. 6

[37] Mirella Walker, Sandro Schönborn, Rainer Greifeneder, and Thomas Vetter. The Basel face database: A validated set of photographs reflecting systematic differences in big two and big five personality dimensions. *PLoS ONE*, 13(3):e0193190, 2018. 6

[38] Tinghuai Wang, John P. Collomosse, Andrew Hunter, and Darryl Greig. Learnable stroke models for example-based portrait painting. In *British Machine Vision Conference*, BMVC '13, 2013. 3

[39] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 4

[40] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *IEEE International Conference on Computer Vision*, ICCV '17, pages 2223–2232, 2017. 1, 2, 3, 6, 7, 8