

# ArtCoder: An End-to-end Method for Generating Scanning-robust Stylized QR Codes

Hao Su<sup>1</sup>, Jianwei Niu<sup>1,2,3\*</sup>, Xuefeng Liu<sup>1</sup>, Qingfeng Li<sup>1</sup>, Ji Wan<sup>1</sup>, Mingliang Xu<sup>2</sup>, Tao Ren<sup>1,3</sup>

<sup>1</sup>State Key Lab of VR Technology and System, School of Computer Science and Engineering, Beihang University

<sup>2</sup>Industrial Technology Research Institute, School of Information Engineering, Zhengzhou University

<sup>3</sup>Hangzhou Innovation Institute, Beihang University

{bhsuhao, niujianwei, liu\_xuefeng, liqingfeng, wanji}@buaa.edu.cn, iexumingliang@zzu.edu.cn

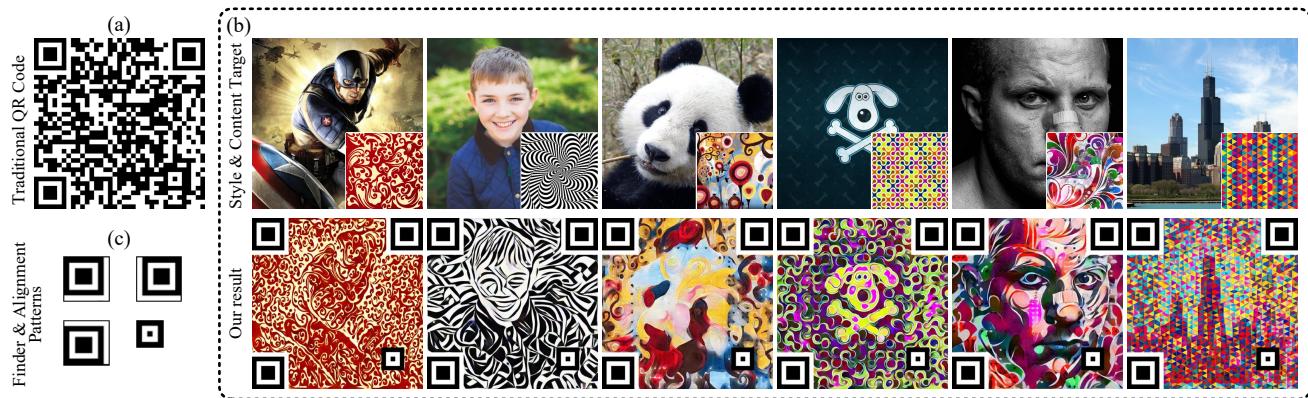


Figure 1: (a) Traditional QR code. (b) Samples of our stylized QR codes. These codes combine the visual effect of stylization and the functionality of QR codes, which are personalized, attractive, and scanning-robust. (c) Finder and alignment patterns are used to determine a QR code's location and angle, thus we preserve their traditional appearances.

## Abstract

Quick Response (QR) code is one of the most worldwide used two-dimensional codes. Traditional QR codes appear as random collections of black-and-white modules that lack visual semantics and aesthetic elements, which inspires the recent works to beautify the appearances of QR codes. However, these works adopt fixed generation algorithms and therefore can only generate QR codes with a pre-defined style. In this paper, combining the Neural Style Transfer technique, we propose a novel end-to-end method, named ArtCoder, to generate the stylized QR codes that are personalized, diverse, attractive, and scanning-robust. To guarantee that the generated stylized QR codes are still scanning-robust, we propose a Sampling-Simulation layer, a module-based code loss, and a competition mechanism. The experimental results show that our stylized QR codes have high-quality in both the visual effect and the scanning-robustness, and they are able to support the real-world application.

\*The corresponding author.

## 1. Introduction

With the ubiquity of smartphones, the *Quick Response (QR) code* [1] has become one of the most-used types of two-dimensional codes, and has been popularly applied in many scenarios including social networks, mobile payments, and advertisements. Traditional QR codes are matrix codes consisting of black-and-white squares *modules* that are visual-unpleasant and meaningless to human vision [Fig. 1(a)]. An appealing QR code will attract more people to scan the code and increase the link visits [2–4], which inspires the recent works to beautify the appearances of QR codes [2–7] (i.e., endowing QR codes with visual semantics or aesthetic elements). However, these works adopt fixed generation algorithms and therefore can only generate QR codes with a pre-defined style, [Fig. 2(b)–(f)], which limits the personalized choices of users.

In this paper, employing the Neural Style Transfer (NST) technique, we propose an end-to-end method, named *ArtCoder*, to generate the stylized QR codes. These stylized codes combine the visual effect of stylized images and the functionality of QR codes, which are personalized, diverse,

and scanning-robust [as shown in Fig. 1(b)]. Although the recent NST works (e.g., [8–12]) have made great progress on stylizing images, however, for generating stylized QR codes, the big extra challenge is to guarantee the scanning-robustness of output codes after giving them art styles. To address this issue, Xu *et al.* [13] propose a two-staged method, i.e., a QR code is first stylized by the NST model [12], and then all error modules caused by stylization are repaired by a post-processing algorithm. Although this method can produce the stylized QR codes, as shown in Fig. 2(h) upper row, the repaired modules are distracting and cannot be well fused with the entire image, due to the asynchrony of stylization and module repair.

Unlike existing works, our ArtCoder is the first end-to-end method to stylize an image and fuse it with the QR code message simultaneously. Moreover, ArtCoder not only blends the black/white modules in an invisible and attractive manner [Fig. 1(b)], but also preserves the scanning-robustness. To improve the performances in both scanning-robustness and visual quality, we propose three key improvements as follows. First, we analysis the relationship between the convolutional layers and the sampling process of QR code readers, and propose a Sampling-Simulation (SS) layer to extract the encoding message of QR codes. Second, we propose a module-based *code loss* to control the scanning-robustness of the stylized QR codes. Third, we propose a competition mechanism between the visual quality and the scanning-robustness to improve their performances.

The main contributions of our work are three-fold:

- we propose a novel end-to-end method ArtCoder to generate the stylized QR codes that are personalized, diverse, and scanning-robust.
- we propose a Sampling-Simulation (SS) layer to extract the message of QR codes, and introduce the module-based code loss to preserve the scanning-robustness of the stylized QR codes.
- we propose a competition mechanism to guarantee the high-quality of the stylized QR codes in both scanning-robustness and visual effect.

## 2. Related Work

Below we summarize the related works that involve two main topics, neural style transfer, and aesthetic QR code.

### 2.1. Neural style transfer

The methods of Neural Style Transfer (NST) can be basically classified as parametric and non-parametric [14].

**Parametric.** Parametric methods iteratively update an initial image until the desired global statistics are satisfied. Gatys *et al.* [8] pioneer the parametric NST method by employing the power of CNN and Gram matrices. Afterwards, the follow-up parametric researches have been presented to improve their performances on visual quality [15–18],

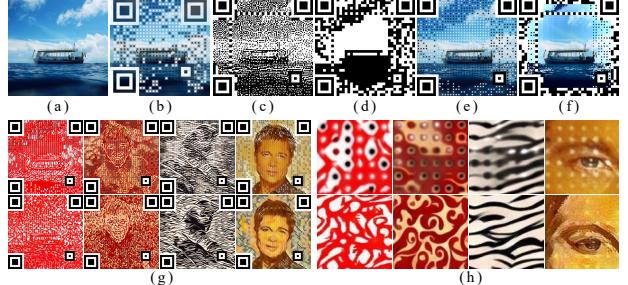


Figure 2: (a) Blended image. (b) Visualead [5]. (c) Halftone QR code [2]. (d) Qart code [27]. (e) Artup [4]. (f) Efficient QR code [3]. (g) SEE QR code [13] (upper) and our results (bottom). (h) Enlarged view of (g).

generating speed [10, 12, 19–21], and multimedia extension [13, 22–24].

**Non-parametric.** Non-parametric methods use a simple patch representation, and find the most similar patches by nearest neighbor search. Non-parametric NST method is pioneered by Li *et al.* [9], and they reformulate the style transfer using the Markov Random Field (MRF), i.e., searching neural patches from the style image to match the structure of content image. Afterwards, various researchers follow the idea of patch-based matching to optimize the stylized results by de-VGG networks [25], semantic-level patch [26], feature reshuffle [14], etc.

### 2.2. Aesthetic QR code

Below we review the methods of blend-type aestheticic QR codes [13] that can blend images with QR codes (Fig. 2), and these methods are mainly based on three directions, module-deformation, module-reshuffled, and NST.

**Module-deformation.** The idea of methods based on module-deformation is first to deform and reduce the regions of square modules, and then insert images in the saved regions, where the representative works are Visualead [5] and Halftone QR codes [2]. Visualead [5] beautifies QR codes by deforming modules and keeping the contrast between modules and blended images [Fig. 2(b)]. Halftone QR codes [2] divide each module into  $3 \times 3$  submodules with keeping the color of the center sub-modules, and then make the other sub-modules to match the halftone map of the blended image [Fig. 2(c)].

**Module-reshuffle.** Recent methods based on module-reshuffle are inspired by the pioneering work Qart code [27] which proposes that the *Gauss-Jordan Elimination Procedure* can be employed to reshuffle modules’ locations to satisfy the features of blended images [Fig. 2(d)]. Afterwards, aiming at improving the visual quality of QR codes, the follow-up works design different strategies to reshuffle modules using different image features, e.g., region of interesting [4], central saliency [3], global gray values [13].

**NST-based method.** Xu *et al.* [13] first introduce the NST technique to generate stylized QR codes, and pro-

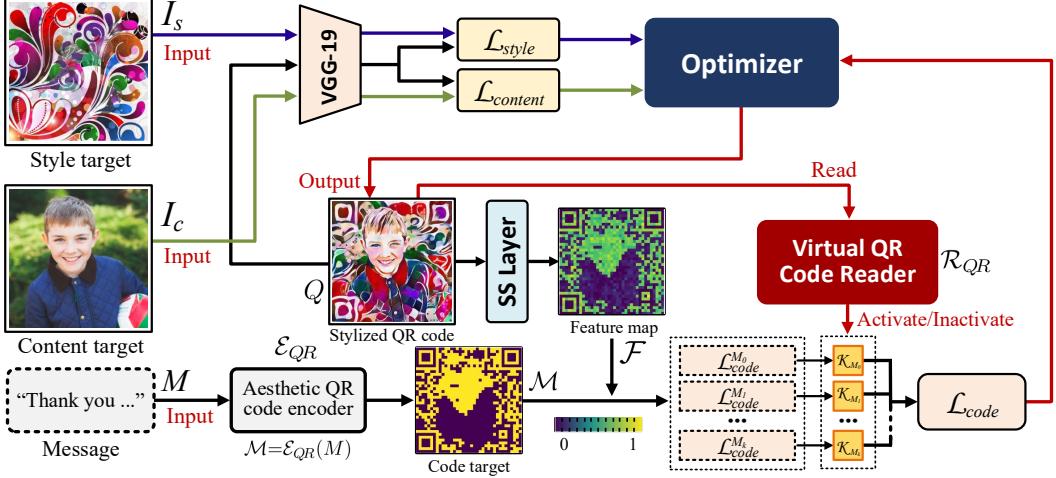


Figure 3: System Pipeline. Given a style target image  $I_s$ , a content target image  $I_c$ , and a message  $M$ , our method is modeled as a function  $\Psi$  to generate a stylized QR code  $Q=\Psi(I_s, I_c, M)$ . For visual effect,  $Q$  combines the style feature of  $I_s$  and the semantic content of  $I_c$ ; for functionality,  $Q$  can be decoded to show message  $M$  by any standard QR code reader.

pose the SEE (Stylized aEsthetic) QR codes [Fig. 2(g) upper row] that are personalized and machine-readable. Their method address the issue that the style transfer will compromise the scanning-robustness, however, the error modules caused by stylization are repaired by a post-processing algorithm, which generates the distracting modules that cannot be well fused with the entire image [Fig. 2(h) upper row].

### 3. Method

#### 3.1. Overview

Given a style target image  $I_s$ , a content target image  $I_c$ , and a message  $M$ , our method is modeled as a function  $\Psi$  to generate a stylized QR code  $Q=\Psi(I_s, I_c, M)$ . For visual effect,  $Q$  combines the style feature of  $I_s$  and the semantic content of  $I_c$ ; for functionality,  $Q$  can be scanned to show message  $M$  by any standard QR code reader. The total objective function  $\mathcal{L}_{total}$  of  $Q=\Psi(I_s, I_c, M)$  is defined as

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{style}(I_s, Q) + \lambda_2 \mathcal{L}_{content}(I_c, Q) + \lambda_3 \mathcal{L}_{code}(M, Q), \quad (1)$$

where  $\lambda_1$  to  $\lambda_3$  are used to balance the multiple objectives. Style loss  $\mathcal{L}_{style}$ , content loss  $\mathcal{L}_{content}$ , and code loss  $\mathcal{L}_{code}$  are minimized by the optimizer, to control the style feature, semantic content, and readability of  $Q$ , respectively.

As shown in Fig. 3, the features of style and content are extracted by VGG-19 [28], and the code feature is extracted by the proposed Sampling-Simulation (SS) layer. In each iteration of optimizer, a virtual QR code reader  $\mathcal{R}_{QR}$  will read the stylized result  $Q$  to discriminate all error and correct modules. For the  $k$ -th module  $M_k$ , if  $M_k$  is error (or correct), we control an activation map  $\mathcal{K}$  to activate (or deactivate) the  $k$ -th sub-code-loss  $\mathcal{L}_{code}^{M_k}$ , to optimize its robustness and may compromise the representations of style

and content (or  $\mathcal{L}_{style}$  and  $\mathcal{L}_{content}$  will try their best to optimize the visual quality).

We will detail the losses of style and content, the SS layer, the code loss, and the virtual QR code reader in Sec. 3.2, Sec. 3.3, Sec. 3.4, and Sec. 3.5, respectively.

#### 3.2. Losses of style and content

The style loss  $\mathcal{L}_{style}$  and the content loss  $\mathcal{L}_{content}$  are not the key points of our work, thus we basically follow the literature [8, 12] as

$$\begin{aligned} \mathcal{L}_{style}(I_s, Q) &= \frac{1}{C_s H_s W_s} \|G[f_s(I_s)] - G[f_s(Q)]\|_2^2, \\ \mathcal{L}_{content}(I_c, Q) &= \frac{1}{C_c H_c W_c} \|f_c(I_c) - f_c(Q)\|_2^2, \end{aligned} \quad (2)$$

where  $G$  indicates the Gram matrix [8, 12], and  $f_s$  (or  $f_c$ ) is the feature map of shape  $C_s \times H_s \times W_s$  (or  $C_c \times H_c \times W_c$ ) that extracted from the  $s$ -th (or  $c$ -th) layer of the pre-trained VGG-19 network [28],  $s \in \{\text{relu1\_2, relu2\_2, relu3\_3, relu4\_3}\}$ , and  $c \in \{\text{relu3\_3}\}$ .

#### 3.3. Sampling-Simulation layer

**Sampling of QR codes:** for decoding QR codes, the most used project *Google ZXing* [29] rules that a QR code reader only samples the center pixel of each module in a QR code, and then binarizes and decodes these pixels. In other words, a QR code is still readable if replacing all original square modules with smaller concentric modules. Meanwhile, the probability of sampling correct pixels is not fixed and proportional to the module sizes, due to the external factors (e.g., camera resolution, scanning distance) when QR codes are scanned by smartphones. To theorize this point, [4, 13] propose that the pixels closer to the module center have a higher probability to be sampled, and the probability follows the Gaussian distribution  $\mathcal{G}$  as

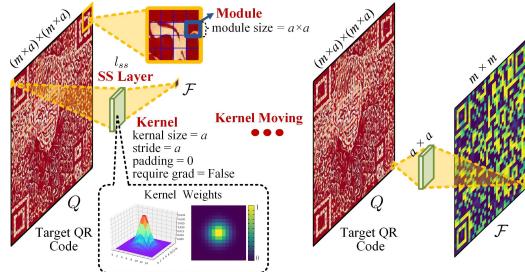


Figure 4: Framework of Sampling-Simulation (SS) layer  $l_{ss}$ . For target QR code  $Q$  that consists of  $m \times m$  modules of size  $a \times a$  pixels,  $l_{ss}$  extracts an  $m \times m$  feature map  $\mathcal{F}$  from  $Q$ , and  $\mathcal{F}$  indicates the sampled colors of all modules in  $Q$ . The kernel weight of  $l_{ss}$  follows the Gaussian distribution as eq.(3), since the pixels closer to module center are more important for the scanning-robustness.

$$\mathcal{G}_{M_k(i,j)} = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}}, \quad (3)$$

where  $(i, j)$  is the coordinate of a pixel in module  $M_k$ , and the origin at the module center, and  $\mathcal{G}_{M_k(i,j)}$  indicates the probability of sampling the pixel  $(i, j)$ .

**Sampling Simulation:** If using a conv layer to simulate the sampling process of a QR code reader, we can control the robustness of QR code by the losses back propagation. To achieve this goal, we analyze the relationship between convolution and sampling, and further design the Sampling-Simulation (SS) conv layer  $l_{ss}$ .

As shown in Fig. 4, for a stylized QR code  $Q$  that consists of  $m \times m$  modules of size  $a \times a$ ,  $l_{ss}$  is designed to have kernel size  $a$ , stride  $a$ , padding 0, and the kernel weights are fixed to follow the Gaussian weight as eq.(3). When we input  $Q$  to  $l_{ss}$ , the kernel will convolve each module of  $Q$  once, and output an  $m \times m$  feature map  $\mathcal{F}=l_{ss}(Q)$ , and  $\mathcal{F}$  indicates the sampled results of  $Q$ . Each bit  $\mathcal{F}_{M_k}$  in  $\mathcal{F}$  is correspond to the module  $M_k$  in  $Q$ , represented as

$$\mathcal{F}_{M_k} = \sum_{(i,j) \in M_k} \mathcal{G}_{M_k(i,j)} \cdot Q_{M_k(i,j)}, \quad (4)$$

where  $\mathcal{G}_{M_k(i,j)}$  defined as eq.(3).

### 3.4. Code loss and competition mechanism

**Code loss:** the code loss  $\mathcal{L}_{code}$  is based on the module of QR code  $Q$ , that is, we set a sub-code-loss  $\mathcal{L}_{code}^{M_k}$  for each module  $M_k \in Q$ , and sum them up to get the total code loss  $\mathcal{L}_{code}$  as

$$\mathcal{L}_{code} = \sum_{M_k \in Q} \mathcal{L}_{code}^{M_k}. \quad (5)$$

For the input message  $M$ , we encode  $M$  to a code target  $\mathcal{M}=\mathcal{E}_{QR}(M)$  by an aesthetic QR code encoder  $\mathcal{E}_{QR}$  [4, 27], aiming to reshuffle the module locations to follow the visual features of the content image  $I_c$ .  $\mathcal{M}$  is an  $m \times m$  matrix consisting of 1 or 0, which marks the ideal color of each module (0/1 means black/white).  $\mathcal{L}_{code}^{M_k}$  is defined as

$$\mathcal{L}_{code}^{M_k} = \mathcal{K}_{M_k} \cdot \|\mathcal{M}_{M_k} - \mathcal{F}_{M_k}\|^2, \quad (6)$$

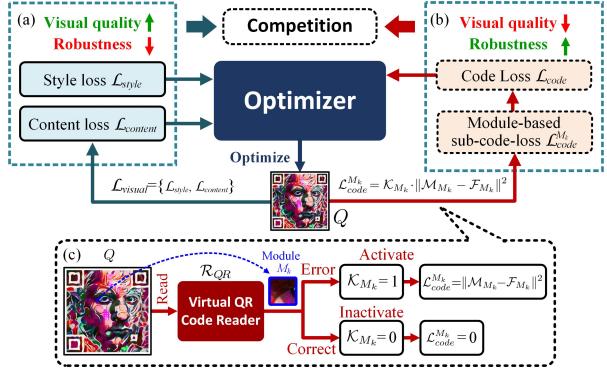


Figure 5: Pipeline of the competition mechanism. (a)-(b)  $\mathcal{L}_{visual} = \{\mathcal{L}_{style}, \mathcal{L}_{content}\}$  (or  $\mathcal{L}_{code}$ ) optimizes the visual quality (or robustness) and compromises the robustness (or visual quality), and these losses compete to optimize  $Q$ . (d) In each iteration,  $\mathcal{R}_{QR}$  read all modules, and activate (or inactivate) the sub-code-losses for error (or correct) modules.

where  $\mathcal{F}$  is the feature map extracted by SS layer,  $\mathcal{K}$  is an activation map computed by the competition mechanism, and  $\mathcal{K}_{M_k} \in \mathcal{K}$  is adopted to activate the sub-code-loss  $\mathcal{L}_{code}^{M_k}$ .

**Competition mechanism:** As shown in Fig. 5, the main idea of the competition mechanism is that through controlling the activation map  $\mathcal{K}$ ,  $\mathcal{L}_{code}$  tries to make each module scanning-robust and compromises the visual quality [Fig. 5(b)], meanwhile,  $\mathcal{L}_{style}$  and  $\mathcal{L}_{content}$  try to improve the visual quality of  $Q$  and compromises the scanning-robustness [Fig. 5(a)].

Specifically, in each iteration, a virtual QR code reader  $\mathcal{R}_{QR}$  reads  $Q$  to find out all error modules, and then constructs the activation map  $\mathcal{K}$  defined as

$$\mathcal{K}_{M_k} = \begin{cases} 1, & \text{if } \mathcal{R}_{QR}(Q_{M_k}) \oplus \mathcal{M}_{M_k} = 1 \\ 0, & \text{if } \mathcal{R}_{QR}(Q_{M_k}) \oplus \mathcal{M}_{M_k} = 0 \end{cases}, \quad (7)$$

where  $\mathcal{M}_{M_k}$  is defined in eq.(6), and  $\mathcal{R}_{QR}(Q_{M_k})$  is the reading result of the  $k$ -th module  $Q_{M_k}$  of  $Q$ .

Combining eq.(1) and (5)-(7), if a module  $Q_{M_k}$  is correct (robust), then  $\mathcal{K}_{M_k} \leftarrow 0$ ,  $\mathcal{L}_{code}^{M_k} \leftarrow 0$ , and our model will try the best to optimize  $\mathcal{L}_{visual} = \{\mathcal{L}_{style}, \mathcal{L}_{content}\}$ , to improve the style and content feature. Afterwards, if these modifications make  $Q_{M_k}$  error, then  $\mathcal{K}_{M_k} \leftarrow 1$ , and  $\mathcal{L}_{code}^{M_k}$  will be activated to optimize the robustness of  $Q_{M_k}$ . As shown in Fig. 5(c), the competition between  $\mathcal{L}_{code}$  and  $\mathcal{L}_{visual}$  will make the final output  $Q$  reach a stable state with preserving both the artistic style and the robustness.

### 3.5. Virtual QR Code Reader

In this subsection, following the binarization theory of a QR code reader, we design a mechanism to trade-off the visual quality and the scanning-robustness.

**Binarization of QR codes.** After a QR code reader scans a QR code  $Q$ , the sampled colored pixels will be con-

$\mathcal{M}$  is an  $m \times m$  binary matrix;  $\mathcal{F}$  is an  $m \times m$  feature map; Since  $\mathcal{M} \in \{0, 1\}$ , and  $\mathcal{F} \in [0, 1]$ , and they can be compared directly.

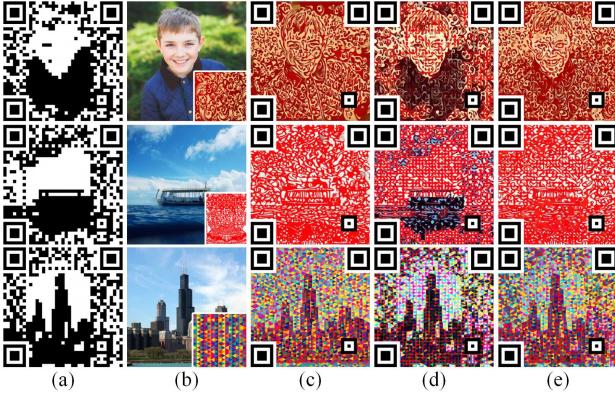


Figure 6: (a) Code target  $\mathcal{M}$ . (b) Content and style targets. (c) W/O  $\mathcal{L}_{code}$ . (d) W/O competition mechanism. (e) Ours.

verted to grayscale and binarized to 0 or 1 by a threshold  $\mathcal{T}$ , defined as

$$Q_{M_k}^b = \xi(Q_{M_k}, \mathcal{T}) = \begin{cases} 0, & \text{if } Q_{M_k} < \mathcal{T} \\ 1, & \text{if } Q_{M_k} \geq \mathcal{T} \end{cases}, \quad (8)$$

where  $Q_{M_k}^b$  is the binarized result of module  $Q_{M_k}$ .

According to eq.(8), for black modules, replacing the black color with a lighter color (e.g., dark red) whose gray value below the threshold  $\mathcal{T}$ , which can still preserve the correct module data (white modules are similar). This manner can make the color of the stylized result more similar to the style target that has a higher visual quality.

**Virtual QR Code Reader:** the virtual QR code reader  $\mathcal{R}_{QR}$  is designed to discriminate the correctness of each module of  $Q$ . Combining eq.(7) and (8), the binarization results of  $\mathcal{R}_{QR}$  are computed by

$$\mathcal{R}_{QR}(Q_{M_k}) = \begin{cases} 0, & \text{if } \mathcal{M}_{M_k}=0 \text{ and } Q_{M_k} < \mathcal{T}_b \\ 1, & \text{if } \mathcal{M}_{M_k}=0 \text{ and } Q_{M_k} \geq \mathcal{T}_b \\ 0, & \text{if } \mathcal{M}_{M_k}=1 \text{ and } Q_{M_k} < \mathcal{T}_w \\ 1, & \text{if } \mathcal{M}_{M_k}=1 \text{ and } Q_{M_k} \geq \mathcal{T}_w \end{cases}, \quad (9)$$

where  $\mathcal{T}_b$  (or  $\mathcal{T}_w$ ) indicates the virtual threshold adopted to binarize black (or white) modules, and  $\mathcal{R}_{QR}(Q_{M_k}) \oplus \mathcal{M}_{M_k} = 1$  (or 0) means module  $Q_{M_k}$  is error (or correct).

According to Sec. 3.4, reducing the strictness of error module discrimination, the optimizer will more focus on optimizing the visual quality. Following eq.(9), the strictness of discrimination is mainly influenced by  $\mathcal{T}_b$  and  $\mathcal{T}_w$ . In grayscale space, we suppose that the distance between the real threshold  $\mathcal{T}$  and the virtual threshold  $\mathcal{T}_b/\mathcal{T}_w$  is proportional to the robustness  $\eta$  for black/white modules, and their relationships follow the *Uniform Distribution* as  $\eta = \frac{|\mathcal{T}-\mathcal{T}_b|}{\mathcal{T}} = \frac{|\mathcal{T}_w-\mathcal{T}|}{(255-\mathcal{T})}$ , where  $\mathcal{T} \in [0, 255]$ ,  $\mathcal{T}_b \in [0, \mathcal{T}]$ ,  $\mathcal{T}_w \in [\mathcal{T}, 255]$ . Therefore, we can set  $\eta$  to control  $\mathcal{T}_w$  and  $\mathcal{T}_b$ , and further trade-off the visual quality and the robustness.

## 4. Experiment

In the following experiments, we evaluate the performances of our stylized QR codes in two aspects, stylization



Figure 7: Influences of weights setting.

quality and scanning-robustness.

### 4.1. Implementation

**Dataset.** The datasets we used in experiments are divided in two parts, the content image dataset  $\mathcal{D}_c$ , and the style image dataset  $\mathcal{D}_s$ .  $\mathcal{D}_c$  contains 100 images of size  $512 \times 512$  with various visual contents (e.g., portrait, cartoon, scenery, animal, logo), and  $\mathcal{D}_s$  contains 30 images with different artistic styles.

**Experimental setting.** We implement our program in PyTorch [30] and all experiments are performed on a computer with a NVIDIA Tesla V100 GPU. In all experiments to evaluate the scanning-robustness, all QR codes are displayed on a 27-inch, 144Hz, and  $3840 \times 2160$  IPS-panel monitor screen. The network adopted to extract features of style and content is VGG-19 [12, 28] pre-trained on MSCOCO [31], and the optimizer is Adam. For all experiments, by default, we set  $\lambda_1=10^{15}$ ,  $\lambda_2=10^7$ ,  $\lambda_3=10^{20}$  in eq.(1),  $s \in \{\text{relu1\_2, relu2\_2, relu3\_3, relu4\_3}\}$ , and  $c \in \{\text{relu3\_3}\}$  in eq.(2), learning rate is 0.001, robust parameter  $\eta$  is 0.6, and each stylized QR code is output at  $10^4$  iterations. Moreover, each QR code is generated in version 5 [1], of size  $592 \times 592$  ( $37 \times 37$  modules, each module of size  $16 \times 16$ ).

### 4.2. Stylization Quality

**Ablation study.** Below we conduct an ablation study on our improvements. The produced results under different improvement as shown in Fig. 6, and the performances are summarized in Tab. 1. We observe that each of our improvement is essential to produce high-quality results.

The code loss  $\mathcal{L}_{code}$  is essential to preserve the scanning-robustness. As shown in Fig. 6(c), without  $\mathcal{L}_{code}$ , the stylized results will lose the functionality of QR codes. The competition mechanism is essential to balance the scanning-robustness and the visual quality. As shown in Fig. 6(d), without the mechanism, the modules in stylized results are uncontrolled, and appear as mess black/white patches that are undesirable and visual-unpleasant.

Table 1: Ablation study on our improvements

	W/O $\mathcal{L}_{code}$	W/O competition mechanism	Ours
Visually pleasant	✓	✗	✓
Scanning-robust	✗	✓	✓

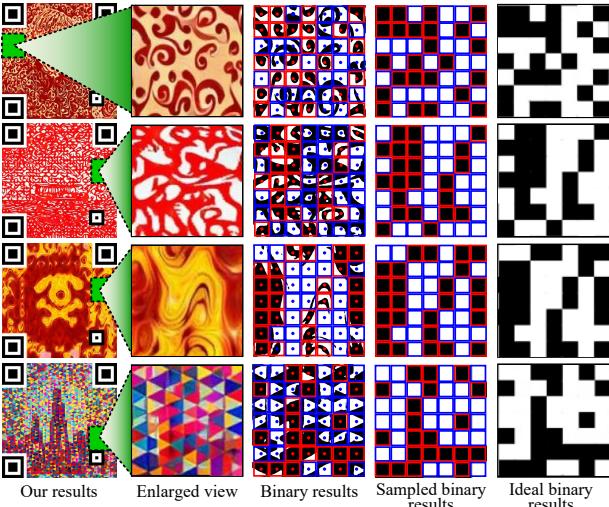


Figure 8: Analysis of preserving scanning-robustness. Red or blue boxes mark the black or white modules, and colored dots mark the pixels sampled by a standard QR code reader.

**Influence of weights setting.** For the total objective function eq.(1), we empirically set the weight of  $\mathcal{L}_{code}$  with a larger number to give top priority to keep the scanning-robustness. With the help of the proposed competition mechanism, a larger  $\mathcal{L}_{code}$  does not compromise the visual quality, since the mechanism will inactivate  $\mathcal{L}_{code}$  for all robust modules, and  $\mathcal{L}_{code}$  will become 0 when each module is robust. In this experiment, we fix  $\lambda_1=10^{15}$ ,  $\lambda_3=10^{20}$ , and only modify the weight  $\lambda_2$  of loss  $\mathcal{L}_{content}$  to evaluate the visual changing. The comparison results as shown in Fig. 7, we observe that without compromising the scanning-robustness, fine-tuning weights can effectively trade-off the representation of content and style.

**Comparison with other methods.** We compare our methods with other state-of-the-art NST methods (containing Gatys *et al.* [8], Fast NST [12], AdaIN [32], and WCT [33]) and aesthetic QR codes methods (containing SEE QR code [13], Halftone QR code [2], Visualead [5], and Artup [4]). The comparison results as shown in Fig. 10, for the compared NST methods, our stylized results achieve a similar stylization quality with them, and outperform them on preserving the functionality of QR codes; for the works of aesthetic QR codes, our methods offer various generating styles that are more personalized, diverse, and artistic. Particularly, for the prior work of stylized QR code [13], we observe that the robustness of their codes relies on the repair by the post-processing algorithm, and the repaired modules appear as some round spots that are visible, undesired, and distracting. Contrarily, all modules in our stylized QR codes are more invisible and well fused with the entire image.

### 4.3. Scanning Robustness

Below we conduct a series of experiments to evaluate the scanning-robustness of our stylized QR codes in real-

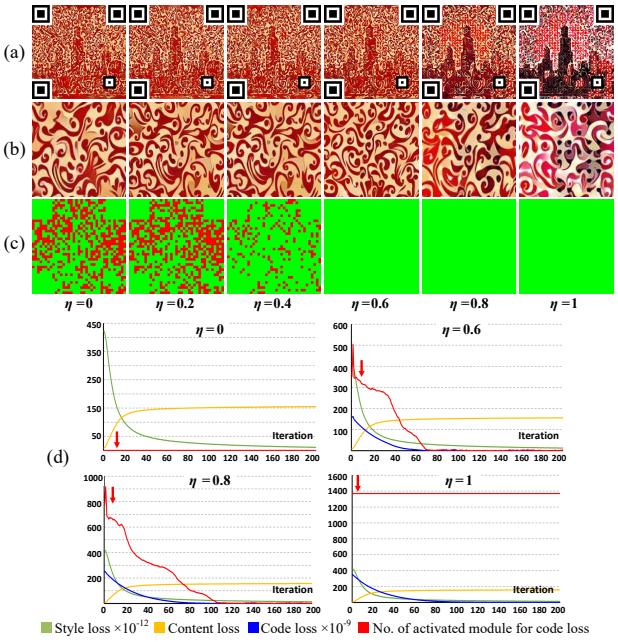


Figure 9: Influences of robustness parameter  $\eta$ . (a) Generated results. (b) Enlarged view of (a). (c) Error modules in (a) (marked with red). (d) Influences on losses.  $\eta$  is proportional (inversely proportional) to robustness (visual quality).

world application.

**Analysis of preserving scanning-robustness.** We first analyze and explain why our stylized results can preserve the scanning-robustness. Following the binarization theory of QR codes described in Sec. 3.5, for the sampled pixels of each module, no matter how their color changes, just preserving the same binary results with the ideal colors, can preserve the scanning-robustness.

As shown in Fig. 8, for our stylized codes, although the colors and shapes of their modules are invisibly blended with the entire image, and become varied and irregular, the sampled pixels still preserve the same binary results with the ideal QR codes. Therefore, our stylized QR codes can be robustly decoded by a standard QR code reader.

**Influence of robustness parameter  $\eta$ .** In Sec. 3.5, the parameter  $\eta$  controls the strictness of discriminating error modules (i.e., controls the robustness of modules). Specifically, when we set a higher  $\eta$ , each module must be blacker/whiter to be classified as a robust module [Fig. 9(a)(b)]. For all non-robust modules, their sub-code-losses need to be activated to optimize the robustness.

In experiment, we randomly select 20 content images and 10 style images from the dataset  $\mathcal{D}_c$  and  $\mathcal{D}_s$  respectively, to generate 20 stylized QR codes, and set  $\lambda_1/\lambda_2/\lambda_3=10^{15}/10^7/10^{20}$  in eq.(1). The experimental results as shown in Fig. 9, for visual quality, a higher  $\eta$  make all modules’ colors become blacker/whiter, which is more visual-unpleasant [Fig. 9(a)(b)]; for scanning-robustness, setting a higher  $\eta$  can make the network generate more ro-

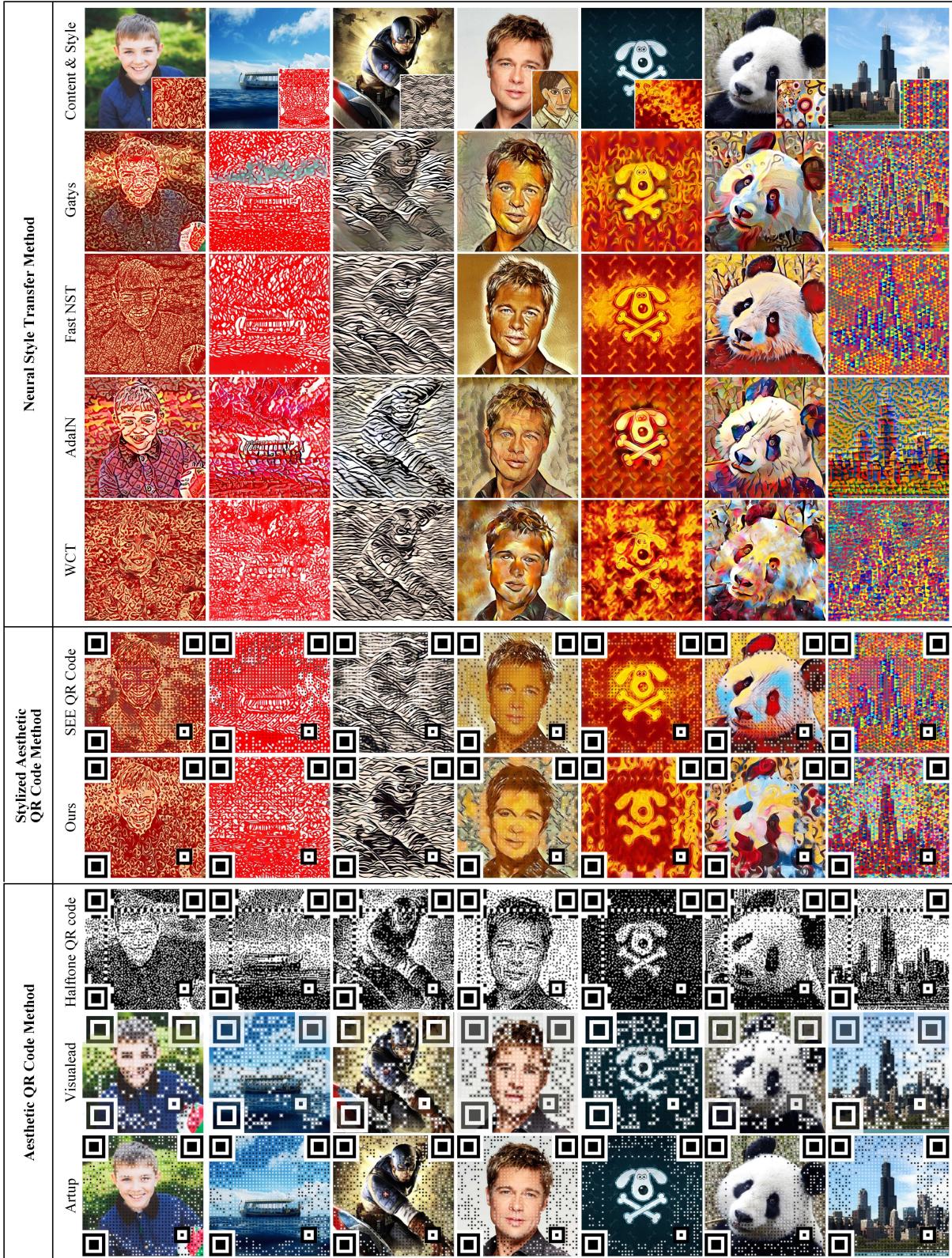


Figure 10: Comparison with previous NST methods and aesthetic QR codes methods. For NST methods, our stylized results achieve a similar stylization quality with them, and outperform them on preserving the functionality of QR codes. For aesthetic QR code methods, our results are more personalized, diverse, and have a higher quality of stylization.

Table 2: Average success rates

Mobile Phone	APP	Successful scanning / Scanning times $\times 100\%$		
		$(3\text{cm})^2$	$(5\text{cm})^2$	$(7\text{cm})^2$
Iphone 11	Twitter	100%	100%	100%
	Facebook	96%	100%	100%
	Wechat	100%	100%	100%
	Alipay	100%	100%	100%
Huawei Mate 20	Twitter	100%	100%	100%
	Facebook	100%	100%	100%
	Wechat	96%	100%	100%
	Alipay	100%	100%	100%
Vivo X30	Twitter	100%	100%	100%
	Facebook	100%	100%	100%
	Wechat	98%	100%	100%
	Alipay	100%	100%	100%
Xiaomi Note 3	Twitter	100%	100%	100%
	Facebook	100%	100%	100%
	Wechat	96%	100%	100%
	Alipay	98%	100%	100%

bust stylized codes that have fewer error module [Fig. 9(c)]; for losses changes, we obverse that a higher  $\eta$  make the model classify more modules as non-robust modules, and further activates these module’s sub-code-losses to optimize the robustness [red arrows in Fig. 9(d)]. To sum up, by modifying  $\eta$ , the proposed method can effectively trade-off the visual quality and robustness, and  $\eta$  is proportional to the robustness and inversely proportional to the visual quality.

**Influences of mobile phone and reader.** We evaluate the influences of different mobile phones and readers as follows. First, we randomly select 10 content images from  $\mathcal{D}_c$ , and 10 style images from  $\mathcal{D}_s$ , to generate a set  $\mathcal{D}_Q$  of 10 stylized QR codes ( $\eta=0.6$ ) with resolution  $512 \times 512$ . Then, each code in  $\mathcal{D}_Q$  is shown on the screen in three frequently-used sizes, i.e.,  $3\text{cm} \times 3\text{cm}$ ,  $5\text{cm} \times 5\text{cm}$ , and  $7\text{cm} \times 7\text{cm}$ . At a distance of  $20\text{cm}$ , we scan each of these 30 codes using different mobile phones and APPs, and record the average number of successful scanning in 50 scanning-times (a successful scanning is defined as the code can be decoded in 3 seconds).

The experimental results in Tab. 2 show that the average rates of successful scanning are always greater than 96% (the failure cases are still decodable, but the decoding time exceeds 3 seconds), which means our stylized codes are robust enough for real-world applications.

**Influences of scanning distances and angles.** To evaluation the influences of scanning distances and angles, we conduct a comparative experiment on related works of aesthetic QR codes, and our method. First, we generate 5 samples for each related work (including Visualead (VS) [5], Halftone QR codes (HF) [2], Efficient QR codes (EF) [3], Artup (AU) [4], SEE QR codes (SE) [13], traditional QR codes, and our results under different  $\eta$  setting). Second, each of these codes is fixed to  $5\text{cm} \times 5\text{cm}$  and is displayed on a screen, and then we scan each code by a same mobile phone at different distances and angles. The decoding time is recorded as the nearest element of  $\{0.5\text{s}, 1\text{s}, 1.5\text{s}, \dots, 5\text{s}\}$ ,

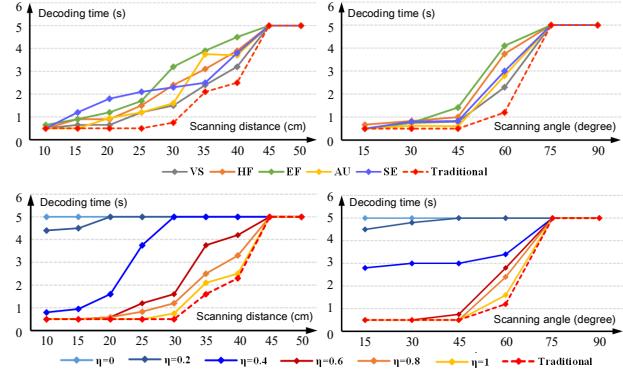


Figure 11: Robustness evaluation on different scanning distances/angles. **Upper:** results of related works of aesthetic QR codes. **Bottom:** our results under different  $\eta$  setting.

and we rule that the time exceeded 5s means the scanned code is a fail case.

The comparison results in Fig. 11 show that when  $\eta \geq 0.6$ , the robustness of our stylized codes achieve the similar performance with the related methods. Moreover, although the robustness of our codes is slightly lower than the traditional QR code, they are robust enough to support the real-world applications.

## 5. Discussion and Conclusion

In this paper, we propose ArtCode, which can generate the stylized QR codes that are personalized, diverse, and scanning-robust. To address the challenge that preserving the scanning-robustness after giving such codes style elements, we propose the SS layer, the module-based code loss, and the competition mechanism to improve the performances. Extensive experiments prove that our stylized QR codes have high-quality in both the visual effect and the scanning-robustness, which is able to support the real-world application.

The proposed ArtCoder is the first end-to-end method that can stylize an image while endowing it with QR code messages. Our current implementation is based on the classical NST method using iterative optimization (e.g., [8]). Although ArtCoder can generate stylized QR codes with good visual quality, compared with the recent fast NST methods (e.g., [32, 33]), ArtCoder still needs improvement in generation speed (it takes an average time of 384.2 seconds to produce a stylized QR code). How to further increase the generating speed without compromising the scanning-robust and the visual quality will become the vital points to be improved in our future work.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant numbers 61772060, 61976012).

## References

- [1] ISO, “Information technology automatic identification and data capture techniques code symbology QR Code,” *Int. Org. Standard.*, Geneva, Switzerland, ISO/IEC 18004: 2000. 1, 5
- [2] H. K. Chu, C. S. Chang, R. R. Lee, and N. J. Mitra, “Halftone QR codes,” *ACM Trans. Graph.*, vol. 32, no. 6, pp. 1–8, 2013. 1, 2, 6, 8
- [3] S.-S. Lin, M.-C. Hu, C.-H. Lee, and T.-Y. Lee, “Efficient QR code beautification with high quality visual content,” *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1515–1524, 2015. 1, 2, 8
- [4] M. Xu, Q. Li, J. Niu, S. Hao, X. Liu, W. Xu, P. Lv, and B. Zhou, “Art-up: A novel method for generating scanning-robust aesthetic qr codes,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2020. 1, 2, 3, 4, 6, 8
- [5] N. Aliva, U. Peled, and F. Itamar. (2012) “Visualead”. [Online]. Available: <http://www.visualead.com/>, Accessedon: Mar.2018. 1, 2, 6, 8
- [6] Y. Zhang, S. Deng, Z. Liu, and Y. Wang, “Aesthetic QR codes based on two-stage image blending,” *Springer Int. Publishing*, pp. 183–194, 2015. 1
- [7] Y.-H. Lin, Y.-P. Chang, and J.-L. Wu, “Appearance-based QR code beautifier,” *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 2198–2207, 2013. 1
- [8] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 2414–2423. 2, 3, 6, 8
- [9] C. Li and M. Wand, “Combining markov random fields and convolutional neural networks for image synthesis,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 2479–2486. 2
- [10] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, “Stylebank: An explicit representation for neural image style transfer,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2
- [11] H. Su, J. Niu, X. Liu, Q. Li, J. Cui, and J. Wan, “Unpaired photo-to-manga translation based on the methodology of manga drawing,” *arXiv preprint arXiv:2004.10634*, 2020. 2
- [12] J. Johnson, A. Alahi, and F.-F. Li, “Perceptual losses for real-time style transfer and super-resolution,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711. 2, 3, 5, 6
- [13] M. Xu, H. Su, Y. Li, X. Li, J. Liao, J. Niu, P. Lv, and B. Zhou, “Stylized aesthetic qr code,” *IEEE Transactions on Multimedia*, vol. 21, no. 8, pp. 1960–1970, 2019. 2, 3, 6, 8
- [14] S. Gu, C. Chen, J. Liao, and L. Yuan, “Arbitrary style transfer with deep feature reshuffle,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8222–8231. 2
- [15] Y. Zhang, Y. Zhang, and W. Cai, “Separating style and content for generalized style transfer,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8447–8455. 2
- [16] S. Gu, C. Chen, J. Liao, and L. Yuan, “Arbitrary style transfer with deep feature reshuffle,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8222–8231. 2
- [17] Y. Jing, Y. Liu, Y. Yang, Z. Feng, Y. Yu, D. Tao, and M. Song, “Stroke controllable fast style transfer with adaptive receptive fields,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 238–254. 2
- [18] Y. Men, Z. Lian, Y. Tang, and J. Xiao, “A common framework for interactive texture transfer,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6353–6362. 2
- [19] X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510. 2
- [20] F. Shen, S. Yan, and G. Zeng, “Neural style transfer via meta networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8061–8069. 2
- [21] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, “Universal style transfer via feature transforms,” in *Advances in neural information processing systems*, 2017, pp. 386–396. 2
- [22] D. Chen, J. Liao, L. Yuan, N. Yu, and G. Hua, “Coherent online video style transfer,” in *Proc. Intl. Conf. Computer Vis.*, 2017. 2
- [23] H. Huang, H. Wang, W. Luo, L. Ma, W. Jiang, X. Zhu, Z. Li, and W. Liu, “Real-time neural style transfer for videos,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 783–791. 2
- [24] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, “Stereoscopic neural style transfer,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6654–6663. 2
- [25] T. Q. Chen and M. Schmidt, “Fast patch-based style transfer of arbitrary style,” in *Proc. of NIPS*, 2016. 2
- [26] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang., “Visual attribute transfer through deep image analogy,” *Acm Trans. on Graphics*, vol. 36, no. 4, p. 120, 2017. 2
- [27] R. Cox. (2012) “Qartcodes”. [Online]. Available: <http://research.swtch.com/qart>, Accessedon: Oct.2017. 2, 4
- [28] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *Comput. Sci.*, 2014. 3, 5
- [29] O. S. (2013) “ZXing”. [Online]. Available: <https://github.com/zxing/zxing>, Accessedon:Mar.2018. 3
- [30] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” 2017. 5
- [31] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick, “Microsoft COCO: common objects in context,” in *Proc. Eur. Conf. Comput. Vis.*, vol. 8693, 2014, pp. 740–755. 5

- [32] X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510. [6](#), [8](#)
- [33] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, “Universal style transfer via feature transforms,” in *Advances in neural information processing systems*, 2017, pp. 386–396. [6](#), [8](#)