MSEmbGAN: Multi-Stitch Embroidery Synthesis via Region-Aware Texture Generation

Xinrong Hu, Chen Yang, Fei Fang✉, Jin Huang, Ping Li, *Member, IEEE*, Bin Sheng, *Member, IEEE*, and Tong-Yee Lee, *Senior Member, IEEE*

Abstract—Convolutional neural networks (CNNs) are widely used for embroidery feature synthesis from images. However, they are still unable to predict diverse stitch types, which makes it difficult for the CNNs to effectively extract stitch features. In this paper, we propose a multi-stitch embroidery generative adversarial network (MSEmbGAN) that uses a region-aware texture generation sub-network to predict diverse embroidery features from images. To the best of our knowledge, our work is the first CNN-based generative adversarial network to succeed in this task. Our region-aware texture generation sub-network detects multiple regions in the input image using a stitch classifier and generates a stitch texture for each region based on its shape features. We also propose a colorization network with a color feature extractor, which helps achieve full image color consistency by requiring the color attributes of the output to closely resemble the input image. Because of the current lack of labeled embroidery image datasets, we provide a new multi-stitch embroidery dataset that is annotated with three single-stitch types and one multi-stitch type. Our dataset, which includes more than 30K high-quality multi-stitch embroidery images, more than 13K aligned content-embroidered images, and more than 17K unaligned images, is currently the largest embroidery dataset accessible, as far as we know. Quantitative and qualitative experimental results, including a qualitative user study, show that our MSEmbGAN outperforms current state-of-the-art embroidery synthesis and style-transfer methods on all evaluation indicators. Our demo and dataset sample can be found on the website https://csai.wtu.edu.cn/TVCG01/index.html.

Index Terms—Multi-stitch embroidery synthesis, region-aware texture generation, style transfer, generative adversarial networks.

1 Introduction

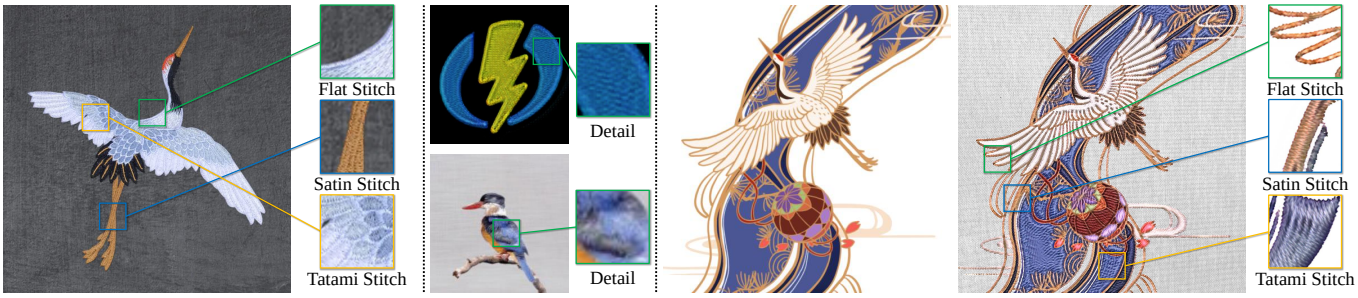
Embroidery, an ancient art form, intricately weaves threads into a fabric to create vivid patterns and textures through diverse stitch types. Historically manual, this technique is evolving with technology. Increasingly, designers employ automated tools like Wilcom ES to create embroidery designs. While these tools enhance speed, they still require considerable time for designers to manually select and adjust stitch types and colors to achieve the desired artistic effect, often involving iterative adjustments and substantial time commitments. This work introduces a novel approach utilizing generative neural network models that not only accelerates this process but also enhances precision in pattern replication and color fidelity—critical aspects in embroidery. Unlike conventional approaches that prioritize style transfer, our method ensures that each stitch in the generated embroidery aligns with professional standards, supporting designers in producing work that meets high aesthetic and technical criteria, and significantly reducing the time spent on manual adjustments.

In the early days, some traditional methods [3]–[7] have explored the influence of stitch types on embroidery textures. Our method draws partial inspiration from these pioneering efforts. Traditional methods are commendable for their ability to produce embroidery textures that closely resemble real stitch styles and offer clear color and textural fidelity that is highly valued in both digital entertainment and law enforcement applications. However, the primary limitation of traditional methods lies in their inability to generate a diverse array of stitch types. While they excel at producing highly realistic single-stitch textures, they often fall short in replicating the complex multi-stitch effects that modern embroidery design software can achieve. This limitation significantly restricts their applicability in contemporary design practices that demand a richer and more varied textural representation. Current general image-to-image generation methods [8]–[14] have been adapted to create embroidery images, but they exhibit notable deficiencies. Specifically, some of these methods struggle with color accuracy, often resulting in a mismatch between the color palette of the generated images and the original inputs. Others fail to produce textures that authentically reflect the intricate details of embroidery, thus lacking the depth and realism required for high-quality embroidery design.

Recent advancements [1], [2] have employed neural networks specifically to generate embroidery images, aiming to incorporate multiple stitch styles. These neural network-based methods strive to accurately reproduce multi-stitch

- *Xinrong Hu is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the School of Computer Science and Artificial Intelligence, Wuhan Textile University, Wuhan 430200, China. E-mail: hxr@wtu.edu.cn.*
- *Chen Yang, Fei Fang, and Jin Huang are with the School of Computer Science and Artificial Intelligence, Wuhan Textile University, Wuhan 430200, China. E-mail: fangfei@wtu.edu.cn, jhuang@wtu.edu.cn.*
- *Ping Li is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, and also with the School of Design, The Hong Kong Polytechnic University, Hong Kong. E-mail: p.li@polyu.edu.hk.*
- *Bin Sheng is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: shengbin@sjtu.edu.cn.*
- *Tong-Yee Lee is with the Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan 70101, Taiwan. E-mail: tonylee@ncku.edu.tw.*

Manuscript received 20 December 2022; revised 29 January 2024. ✉ *Fei Fang is the corresponding author (Email:fangfei@wtu.edu.cn).*



(a) Real embroidery artwork (b) State-of-the-art result (c) MSEmbGAN

Fig. 1: MSEmbGAN synthesizes a multi-stitch embroidery image containing various styles similar to real embroidery. Extant models cannot synthesize multi-stitch patterns and textures. (a) Real embroidery pieces often contain a variety of stitch styles. (b) Comparisons of EmbGAN [1] (top) and Weis [2] (bottom) methods. (c) Our MSEmbGAN model synthesizes multi-stitch embroidery images (i.e., flat, satin, and tatami stitches). For better visual effects, we filled the white regions in the image with cloth patterns.

textures, essential for achieving the depth and realism seen in traditional embroidery, yet with the versatility that modern applications require. Furthermore, when using neural network-based style transfer models for multi-stitch embroidery synthesis, these models are typically trained on extensive datasets from similar domains. Although this training enhances their performance, the datasets most applicable to style transfer often do not contain the variety of patterns, textures, and color features needed for authentic multi-stitch embroidery tasks. As a result, these neural network-based methods can struggle with stability in generating complex embroidery textures, and the outcomes may deviate significantly from the expected ground truth textures. For instance, studies such as those by [15] and [16], while innovative, do not always effectively preserve the original color integrity of the ground truth images, often leading to color shifts. This highlights the ongoing challenge of developing a neural network method that can both capture the detailed characteristics of multiple stitches and maintain the color accuracy of the original designs.

This paper proposes the first convolutional neural network (CNN)-based multi-stitch embroidery generative adversarial network (MSEmbGAN) and a large annotated multi-stitch embroidery dataset. In the field of deep learning, our work focuses for the first time on the influence of both single and multiple stitches and generates superior results, as shown in Fig. 1 Specifically, MSEmbGAN can identify diverse stitch textures from different embroidery regions in the input image and maintain consistent color features in the resulting image. We propose two sub-networks to achieve these effects. The region-aware texture generation network generates the embroidery texture of the local region corresponding to the suitable stitch, and the colorization network optimizes the full embroidery image by maintaining color consistency. Specifically, the region-aware texture generation network can guarantee that each color region has a specific embroidery stitch with our novel stitch classifier to detect the multiple color regions of the input image and mark the stitch type of the color regions. Meanwhile, the colorization network contains a color feature extractor to achieve full image color consistency by forcing the color features of the result to approximate the input image as much as possible. Owing to the lack of relevant datasets, we also compile and annotate the field's largest multi-stitch embroidery dataset for model training and testing. Each image is assigned one of three single-stitch labels (i.e., satin, tatami, or flat) or a multi-stitch label that reflects the appropriate mixture of stitches. This dataset has more than 30K high-quality multi-stitch embroidery images, including over 13K aligned content embroidery images and over 17K unaligned images. In summary, our work provides the following three main contributions:

- We propose MSEmbGAN, the first learning-based model to successfully synthesize multi-stitch embroidery images that contain a variety of stitch textures and colorful.
- We propose two collaborative sub-networks: a region-aware texture generation network to ensure that embroidery textures are diverse while maintaining accurate stitch features, and a colorization network to ensure color consistency between input and output images.
- We built the largest multi-stitch embroidery dataset available. It is also the first embroidery dataset that is carefully annotated by single and multi-stitch labels.

Extensive experiments show that our MSEmbGAN can generate embroidery images containing diverse stitches. Moreover, our results are more realistic and vivid compared with state-of-the-art (SOTA) embroidery synthesis methods, as shown in Fig. 1. We establish two websites to showcase partial samples of our multi-stitch embroidery dataset [1] and a demo of our MSEmbGAN network [2], respectively.

2 Related Work

2.1 Single-Stitch Embroidery Style Synthesis

Most existing embroidery-style synthesis models focus on single-stitch embroidery-style synthesis. Zhou et al. [17] first added new stitch attributes for each region. Subsequent works gradually began to focus on the stitch styles of the

1. https://drive.google.com/drive/folders/1VEg01D0vmO82oAYHr9D7KzDUCRza-DmU?usp=sharing
2. https://csai.wtu.edu.cn/TVCG01/index.html

generated images. For example, Yang et al. [18], [19] focused on the nature of the stitch construct to uncover different artistic styles related to random-needle embroidery. Qian et al. [20] added noise to gray images and applied a style transfer method to improve model training. Wei et al. [2] used a semantic segmentation method to extract the target content image from the embroidery synthesis process. Beg et al. [1] proposed an unsupervised image-to-image translation method to generate embroidery images, but the method can only render single-stitch features.

There is a more sophisticated embroidery style called random needle embroidery. The above general embroidery generation methods cannot produce this embroidery pattern well. For this reason, Yang et al. [21] proposed a multi-layer rendering technique to specially generate random needle embroidery patterns. Chen et al. [22] proposed a random needle embroidery pattern generator that uses the Markov chain model. Qian et al. [23] proposed a CNN containing two loss functions to generate random needle embroidery images. Since random needle embroidery cannot be used to examine the combinations of multi-stitch texture, it can only be deconstructed into single-stitch constituent styles. Because the use of multi-stitch embroidery styles is more common in practical applications, we begin to pay attention to the generation of multi-stitch embroidery styles. In this work, we focus on the effects of combined stitch types on the quality and complexity of the generated embroidery images. In our previous work [24], we implemented the embroidery image generation using the embroidery channel attention. In this work, we aimed to maintain color consistency between the input images and output embroidery images. In contrast to our previous work, in this paper, we add multi-stitch features to the resulting images while maintaining color consistency.

2.2 Multi-Stitch Embroidery Style Synthesis

Compared with single-stitch embroidery, multi-stitch embroidery focuses on a rich variety of combined stitch styles. Chen et al. [3] proposed a technique to automatically generate embroidery patterns from line drawings. Other works [4], [25] have supplemented the handling of lighting issues during the embroidery rendering process. Despite their advancements, these methods rely on the users to setup the parameters about image segmentation and stitch style [26], which makes them time-consuming and laborious. These methods make it difficult for users without professional knowledge to carry out embroidery design and creation. In recent years, some methods have attempted to address these concerns [7], [27] [2]. Unfortunately, these works cannot generate dynamic and flexible stitch textures owing to their reliance on traditional algorithms. Thus, we consider the use of adversarial deep learning to ensure that the generated stitch texture is both lifelike and retains the color of the original image.

2.3 Image-to-Image Translation

Image-to-image translation [28]–[30], which maps one image domain to another, is a hot research field. Cai et al. [28] proposed an automatic retouching method for scratched photographs under the context of scratch or background. Xu et al. [29] proposed a dual-task deep learning method to simultaneously separate the effect and restore the content from a cartoon animation. Xiao et al. [30] discovered an intermediate domain that can bridge the large inherent gap between two domains to convert a human portrait into anime style. Previous to this, GANs are common and proven effective tools [31] for image-to-image translation tasks. Pix2Pix [13] is an image-to-image translation model that leverages a conditional GAN [32]. CycleGAN [15] learned the translation between two domains by introducing cycle consistency loss without paired data. However, the above methods cannot generate multi-stitch embroideries due to the inability to input information on stitch types.

StyleGAN [33] can modify the resulting facial information at latent code space, but we found it cannot achieve fine results in embroidery synthesis tasks in our experiment. The UNIT model [34] was designed based on the assumption that different data spaces share the same latent space. Hence, an image-to-image translation problem can be treated as a latent space translation problem. MUNIT [14] defined the shared latent space as content space and the difference latent space as the style space that is used to perform multimodal image-to-image translation tasks. Some methods [35], [36] forced the latent code to be converted to a Gaussian distribution, whereas others [37], [38] added conditional information based on a variational auto-encoder (VAE). The diverse image-to-image disentangled representation translation (DRIT++) [16] also used a VAE to perform the task of disentangling representations. However, these methods cannot generate embroidery images with a variety of stitches and lack the ability to handle complex patterns. To address these problems, our MSEmbGAN provides a region-aware texture generation network to generate multi-stitch styles and a colorization network to solve the color shift problem between input and output images.

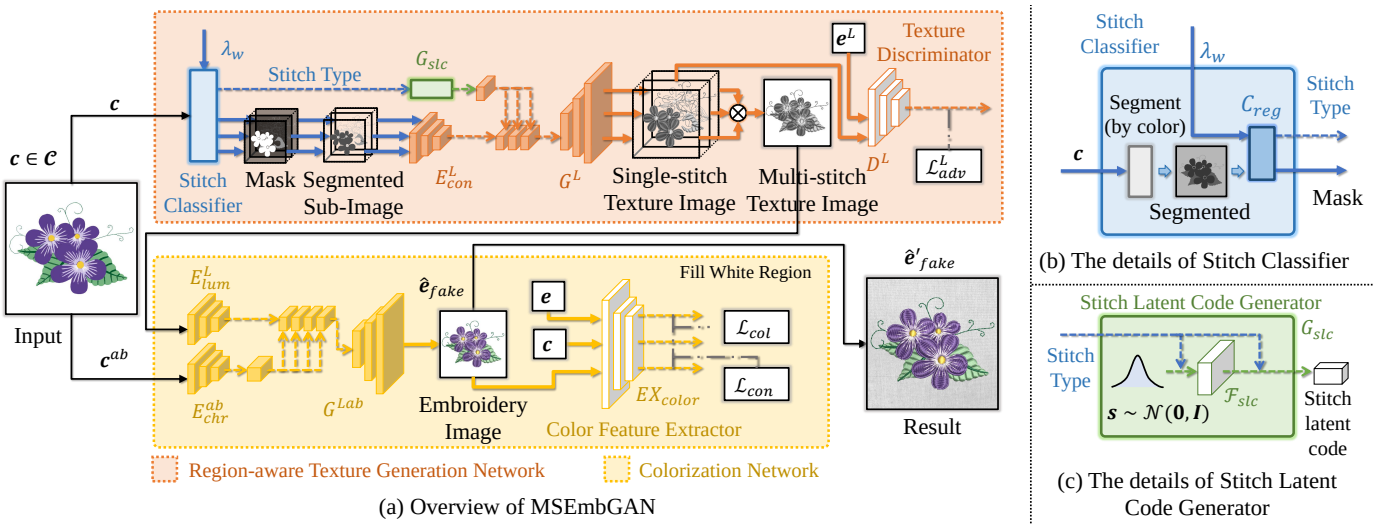
3 METHOD

Our proposed MSEmbGAN comprises two sub-networks: a region-aware texture generation network that generates grayscale embroidery texture images $\hat{\boldsymbol{e}}^L_{fake}$, and a colorization network that optimizes the color features of the final result $\hat{\boldsymbol{e}}_{fake}$.

To support multi-stitch embroidery image learning, we decouple the region-aware texture generation network into a stitch classifier module C_{st}, a stitch latent code generator module G_{slc}, a content encoder E^L_{con} and a texture generator G^L. Fig. 2 presents an overview of the MSEmbGAN architecture. To train the model to closely approximate the color of the input image, we incorporate a color feature extractor and a color consistency loss function.

In particular, the high-frequency features of embroidery images are the key learning objectives of our network. We use the Lab color space to help optimize the luminance L and chrominance ab of the generated images.

Due to the complexity of the region-aware texture generation network, we apply two-step training for it. The first step is to generate the embroidery texture, and the second step is to reconstruct the color information. In this paper, our task is to map the content image domain $\mathcal{C}$ to the embroidery image domain $\mathcal{E}$. MSEmbGAN is trained



(a) Overview of MSEmbGAN

(b) The details of Stitch Classifier

(c) The details of Stitch Latent Code Generator

Fig. 2: Overview of the MSEmbGAN architecture. (a) MSEmbGAN contains two sub-networks, namely the region-aware texture generation network (orange box) and the colorization network (yellow box). The region-aware texture generation network consists of a stitch classifier module (blue box) and a stitch latent code generator (green box). (b) The stitch classifier classifies the input image $\boldsymbol{c}$ into one of three stitch types, namely satin, tatami and flat, according to the shape of the color regions $\mathcal{C}_{reg}$. (c) Given a random Gaussian distribution $\mathbf{s}$ and stitch-type labels, the stitch latent code generator generates the latent code of the stitch texture.

with paired training data $X_{data} = \{(\boldsymbol{c}_i, \boldsymbol{e}_i, st_i)|\boldsymbol{c}_i \in \mathcal{C}$ and $\boldsymbol{e}_i \in \mathcal{E}, st_i \in \{s, t, f\}, i = 1, 2, \ldots, n\}$, where n is the number of content-embroidery pairs in the training dataset and st represents the stitch type. In Section 3.1, we introduce the three basic embroidery stitches involved. We present the network architecture of MSEmbGAN, including the sub-networks, in Section 3.2 and introduce the optimization strategy in Section 3.3. Finally, we further introduce the training and testing mechanisms in Section 3.4.

3.1 Basic Embroidery Stitch

An embroidery artwork usually comprises diverse stitch types and color forms expressed by embroidered patterns. Specially, the stitch type directly determines the characteristics of the final texture effect (see Fig. 1 and Fig. 3). In order to enable the network to better extract features from different types of stitches, we chose three distinct stitch styles as the basic stitches, namely satin, tatami, and flat, as illustrated in Fig. 3. The characteristics of these three stitch types are described as follows: (1) *Satin stitch (st=s)* is a commonly used embroidery stitch type. The needle is dropped on both sides of the object's outline, and the folded line advances in a snakelike pattern. The angle, density, and length of the satin stitch vary from object to object. (2) *Tatami stitch (st=t)*, which is also known as the Xiwen stitch, resembles the tatami grass-weave pattern. The embroidery stitch forms a block surface in the way of line walking to form a unified and orderly needle group. (3) *Flat stitch (st=f)* is another common embroidery stitch in which the needle is continuously stitched along a linear path that traces a pattern with varying expressions of thickness and shape.

3.2 Network Architecture

MSEmbGAN is partially inspired by the image-to-image translation method DRIT++ [16] and follows the same representation model. In order to get higher-quality and more expressive resulting images, we have greatly modified the network architecture of the basic representation model. Our MSEmbGAN first recognizes the stitch types within the input image regions, generates corresponding embroidery textures based on the identified stitch types, and finally optimizes the overall color of the results.

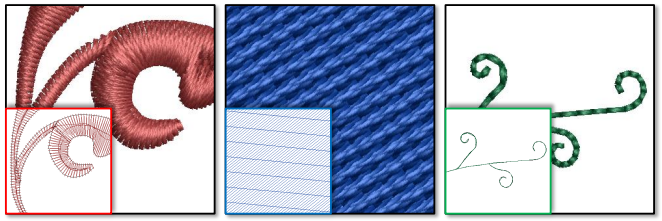


Fig. 3: Three different types of embroidery stitches. The small images are vector graphics corresponding to the stitches. The big images are embroidery patterns corresponding to the stitches.

To achieve the above functions, we propose two sub-networks. The region-aware texture generation network detects multiple color regions of the input image $\boldsymbol{c}$ and generates the gray-scale single-stitch embroidery images $\hat{\boldsymbol{e}}^L_{fake|st}$ (L means the luminance and st means the stitch type) according to the shape features of each local color region $\{\boldsymbol{r}_i\}_{i=1}^N$ (N is the total number of the regions). The colorization network further refines the overall image $\hat{\boldsymbol{e}}^L_{fake}$, ensuring that the color of the generated multi-stitch embroidery image closely approximates the the color of the input image. In the rest of this section, we describe the architecture of the two sub-networks in detail.

3.2.1 Region-Aware Texture Generation Network

The region-aware texture generation network consists of a stitch classifier, a stitch latent code generator, and a texture generator, as shown in Fig. 2. Given an input image $\boldsymbol{c} \in \mathcal{C}$, the stitch classifier module extracts its shape feature and applies the quick shift algorithm [39] and stitch classification algorithm C_{reg} to divide the input image into regions $\{\boldsymbol{r}_i\}_{i=1}^{N}$ and assign the $\{\boldsymbol{r}_i\}_{i=1}^{N}$ corresponding 0/1 masks $\{\boldsymbol{b}_i\}_{i=1}^{N}$ to the regions, where N is the total number of the color regions. This process results in a set of stitch masks labeled with the stitch type st.

Then the stitch latent code generator generates the stitch latent code $\boldsymbol{z}_{s|st}$ for each region. Through the above steps, the region-aware texture generation network generates the gray-scale single-stitch embroidery images $\hat{\boldsymbol{e}}^{L}_{fake|st}$ corresponding to st for each color region according to the masks labeled for each region.

We define parameter λ_w as the region-width threshold, which bounds the size of the circle kernel $\boldsymbol{K}^{\lambda_w}$ to erode the binary mask image of $\boldsymbol{r}_i$. Meanwhile, the stitch type per region, denoted by st_i, are split as follows:

$$st_i = \begin{cases} f & \|\boldsymbol{b}_i \ominus \boldsymbol{K}^{\lambda_f}\|_1 = 0, \\ s & \|\boldsymbol{b}_i \ominus \boldsymbol{K}^{\lambda_f}\|_1 \neq 0 \textbf{ and } \|\boldsymbol{b}_i \ominus \boldsymbol{K}^{\lambda_w}\|_1 = 0, \\ t & \|\boldsymbol{b}_i \ominus \boldsymbol{K}^{\lambda_w}\|_1 \neq 0. \end{cases} \quad (1)$$

where $\boldsymbol{r}_i$ is the region to be filled by the satin stitch when $st_i = s$. Otherwise, it is filled by the tatami stitch when $st_i = t$ or the flat stitch when $st_i = f$. The symbol $(\boldsymbol{r}_i \ominus \boldsymbol{K}^{\lambda_w})$ indicates that the region $\boldsymbol{r}_i$ is eroded by a circle kernel $\boldsymbol{K}^{\lambda_w}$ with diameter λ_w, and $\boldsymbol{K}^{\lambda_f}$ indicates the circle kernel of diameter λ_f. Obviously, $\lambda_w > \lambda_f$ because the region of tatami stitch is the widest. The region mask set can be divided into three sets (i.e., $\{\boldsymbol{b}_{i|st=s}\}_{i=1}^{S}$, $\{\boldsymbol{b}_{i|st=t}\}_{i=1}^{T}$ and $\{\boldsymbol{b}_{i|st=f}\}_{i=1}^{F}$) according to the stitch types. Each set of region masks are then merged into corresponding sub-images, as shown in Fig. 4. Overall, the input image $\boldsymbol{c} \in \mathcal{C}$ is split into three sub-images (i.e., $\boldsymbol{c}_s$, $\boldsymbol{c}_t$, and $\boldsymbol{c}_f$) according to the stitch types of embroidery:

$$\boldsymbol{c}_s = (\sum_i \boldsymbol{b}_{i|s}) * \boldsymbol{c},\ \boldsymbol{c}_t = (\sum_i \boldsymbol{b}_{i|t}) * \boldsymbol{c},\ \boldsymbol{c}_f = (\sum_i \boldsymbol{b}_{i|f}) * \boldsymbol{c} \quad (2)$$

where $*$ represents element-wise multiplication. The sub-images $\{\boldsymbol{c}_s, \boldsymbol{c}_t, \boldsymbol{c}_f\}$ are then transferred to colorization network as inputs to generate single-stitch embroidery textures of type $st \in \{s, t, f\}$.

The stitch latent code generator module consists of residual blocks $\mathcal{F}_{slc}$ and a random Gaussian noise generator. It generates $\boldsymbol{z}_{s|st}$ based on the stitch type, st, and random prior stitch texture latent code $\boldsymbol{s} \sim \mathcal{N}(0, \boldsymbol{I})$. This process can be expressed as $\boldsymbol{z}_{s|st} = \mathcal{F}_{slc}(\boldsymbol{s}, st)$.

The region-aware texture generation network that generates the stitch texture corresponding to the stitch type of each segmented region consists of four parts: an embroidery encoder E^{L}_{emb}, a content encoder E^{L}_{con}, a luminance generator G^{L} and a texture discriminator D^{L}, where L is the luminance channel and ab is chrominance channel of the Lab color space. Consequently, we obtain a grayscale reconstructed embroidery image $\hat{\boldsymbol{e}}^{L}_{recon|st} = G^{L}(\mathcal{F}_{slc}(E^{L}_{emb}(\boldsymbol{e}^{L}, st), st), E^{L}_{con}(\boldsymbol{c}^{L}))$ and a grayscale generated single-stitch embroidery image $\hat{\boldsymbol{e}}^{L}_{fake|st} = G^{L}(\boldsymbol{z}_{s|st}, E^{L}_{con}(\boldsymbol{c}^{L}))$.

(a) Segmented Mask (b) $\boldsymbol{c}^{L}_{s}$ (c) $\boldsymbol{c}^{L}_{t}$ (d) $\boldsymbol{c}^{L}_{f}$

Fig. 4: Examples of intermediate processes for the stitch classifier: (a) visualization of 0/1 masks merged by three stitch types (red: satin, blue: tatami, green: tatami, gray: none); (b) grayscale content subimage of the satin stitch; (c) grayscale content subimage of the tatami stitch; and (d) grayscale content subimage of the flat stitch.

3.2.2 colorization network

The colorization network adds the color information to the generated grayscale multi-stitch embroidery image $\hat{\boldsymbol{e}}^{L}_{fake}$ fused from the grayscale single-stitch embroidery images $\hat{\boldsymbol{e}}^{L}_{fake|st}$, so we can obtain the final embroidery image $\hat{\boldsymbol{e}}_{fake}$. The network consists of a luminance encoder E^{L}_{lum}, a chrominance encoder E^{ab}_{chr}, a chrominance generator G^{Lab}, and a color feature extractor EX_{color}. The luminance encoder E^{L}_{lum} encodes the luminance channel of the generated embroidery image $\hat{\boldsymbol{e}}^{L}_{fake}$. The chrominance encoder E^{ab}_{chr} encodes the chrominance channel of the input image $\boldsymbol{c}^{ab}$. The embroidery image $\hat{\boldsymbol{e}}_{fake}$ is generated from the chrominance generator G^{Lab}. Thus, $\hat{\boldsymbol{e}}_{fake} = G^{Lab}(E^{L}_{lum}(\hat{\boldsymbol{e}}^{L}_{fake}), E^{ab}_{chr}(\boldsymbol{c}^{ab}))$. Finally, we use the hyper-parameter λ_{lum} to adjust the brightness of the resulting images: $\hat{\boldsymbol{e}}_{fake} = \lambda_{lum}\hat{\boldsymbol{e}}^{L}_{fake} + \hat{\boldsymbol{e}}^{ab}_{fake}$. In our experiment, we set $\lambda_{lum} = 0.9$.

3.3 Optimization

The total loss function contains two parts that correspond to the two sub-networks. The texture generation loss encourages the region-aware texture generation network (Section 3.3.1) to synthesize the texture $\hat{\boldsymbol{e}}^{L}_{fake|st}$ that matches the stitch type st. The color consistency loss encourages the colorization network (Section 3.3.2) to add color details similar to the input image to the generated image for further optimization.

3.3.1 Texture Generation Loss

To speed up the convergence of the network, we provide a two-step training process for the region-aware texture generation network. In each step, the region-aware texture generation network is optimized by a step loss function. Now we describe the two training steps separately.

In the reconstruction step, inspired by the work of [16], we apply an embroidery reconstruction process to encourage the network to retain the color information of the input image. The step loss is formulated as follows:

$$\mathcal{L}^{L}_{step-1} = \mathcal{L}^{L}_{recon} + \lambda_{KL}\mathcal{L}^{L}_{KL} \quad (3)$$

where λ_{KL} are weights of the given Kullback–Leibler (KL) divergence loss $\mathcal{L}^{L}_{KL}$. The above loss forces the latent code

generated by the embroidery encoder E^L_{emb} to approximate the Gaussian distribution.

In this step, we use the KL divergence loss $\mathcal{L}_{KL}$ [35], [36] during training to push the distribution of the embroidery texture latent code toward a Gaussian distribution $\mathcal{N}(0, \boldsymbol{I})$:

$$\begin{aligned}\mathcal{L}_{KL} &= \mathbb{E}_{\boldsymbol{c}\sim X_{data}} \sum [E(\boldsymbol{c}) \log \frac{E(\boldsymbol{c})}{\boldsymbol{n}}] \\ &= \frac{1}{2}[\boldsymbol{\mu}^T\boldsymbol{\mu} + \sum(\boldsymbol{\sigma}^T\boldsymbol{\sigma} - \log(\boldsymbol{\sigma}^T\boldsymbol{\sigma}) - 1)]\end{aligned} \tag{4}$$

where the encoder E outputs the mean $\boldsymbol{\mu}$ and the logarithm of covariance $\log(\boldsymbol{\sigma}^T\boldsymbol{\sigma})$ of the latent code, and $\boldsymbol{n} \sim \mathcal{N}(0, \boldsymbol{I})$ is an eight-dimensional random vector. The latent code $\boldsymbol{z}$ can be sampled by $\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{n} * (\boldsymbol{\sigma}^T\boldsymbol{\sigma})$. To force the reconstructed embroidery image $\hat{\boldsymbol{e}}^L_{recon|st}$ to approximate the input image $\boldsymbol{e}^L_{st}$, we use the following reconstruction loss $\mathcal{L}^L_{recon}$ to optimize the network:

$$\begin{aligned}&\mathcal{L}^L_{recon}(E^L_{emb}, E^L_{con}, G^L, \mathcal{F}_{slc}) \\ &\quad = \mathbb{E}_{(\boldsymbol{c}^L, \boldsymbol{e}^L, st)\sim X_{data}}[\|\boldsymbol{e}^L_{st} - \hat{\boldsymbol{e}}^L_{recon|st}\|_1]\end{aligned} \tag{5}$$

where $\| * \|_1$ is the L1-norm whose output is sparse to prevent over-fitting.

In the generation step, the sampled random latent code $\boldsymbol{s} \sim \mathcal{N}(0, \boldsymbol{I})$ replaces the embroidery encoder E^L_{emb} to generate embroidery results. Unlike the work of [16], we sample a latent vector $\boldsymbol{s} \sim \mathcal{N}(0, \boldsymbol{I})$ as the texture representation and reconstruct it. Additionally, we add stitch type of the input content sub-image st as a condition to our region-aware texture generation network to generate the corresponding fake texture sub-image. The generation step loss is as follows:

$$\mathcal{L}^L_{step-2} = -\mathcal{L}^L_{advD} + \mathcal{L}^L_{advG} + \lambda_{latent}\mathcal{L}^L_{latent} \tag{6}$$

where λ_{latent} is a hyper-parameter that is used to balance each term in the above formula.

In this step, we reconstruct the the resulting image in the latent space. The latent regression loss $\mathcal{L}^L_{latent}$ is used to construct an invertible mapping between the image space and latent space:

$$\begin{aligned}&\mathcal{L}^L_{latent}(E^L_{emb}, E^L_{con}, G^L, \mathcal{F}_{slc}) \\ &\quad = \mathbb{E}_{(\boldsymbol{c}^L, \boldsymbol{e}^L, st)\sim X_{data}}[\|\hat{\boldsymbol{s}}_{recon|st} - \boldsymbol{s}\|_1]\end{aligned} \tag{7}$$

where the reconstructed texture latent code is $\hat{\boldsymbol{s}}_{recon|st} = E^L_{emb}(\hat{\boldsymbol{e}}^L_{fake|st}, st)$. Instead of using normal GAN [31], we use the Wasserstein GAN (WGAN) [40] to alleviate the vanishing gradient problem (see Section 4.1). The following adversarial loss is used to approximate the ground truth embroidery:

$$\begin{aligned}&\mathcal{L}^L_{advD}(D^L) \\ &\quad = \mathbb{E}_{(\boldsymbol{c}^L, \boldsymbol{e}^L, st)\sim X_{data}}[D^L(\boldsymbol{e}^L_{st}) - D^L(\hat{\boldsymbol{e}}^L_{fake|st})] \\ &\mathcal{L}^L_{advG}(E^L_{emb}, E^L_{con}, G^L, \mathcal{F}_{slc}) \\ &\quad = \mathbb{E}_{(\boldsymbol{c}^L, \boldsymbol{e}^L, st)\sim X_{data}}[D^L(\hat{\boldsymbol{e}}^L_{fake|st})]\end{aligned} \tag{8}$$

3.3.2 color consistency Loss

With embroidery stylization, it is crucial to ensure that the generated images preserve the color of the input images.

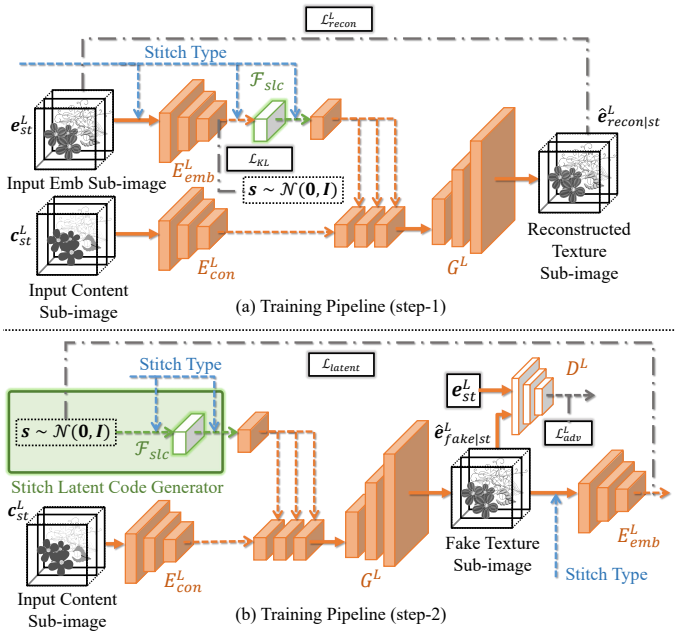


Fig. 5: Two training steps of the region-aware texture generation network: (a) In Step 1, the KL divergence loss encourages the stitch texture representation $q(E^L_{emb}(\boldsymbol{e}^L_{st}, st)|\boldsymbol{e}^L_{st}, st)$ with stitch type st to approximate a prior Gaussian distribution, $\mathcal{N}(0, \boldsymbol{I})$; (b) In Step 2, we construct an invertible mapping between the image and latent space.

MSEmbGAN uses the high-level feature maps of the pre-trained color feature extractor EX_{color} to measure the color preservation of generated embroidery images.

To extract the high-level color feature of an image, the color feature extractor EX_{color} is pre-trained by the color consistency loss. Then the color feature of the input and output images are extracted separately by EX_{color}. We define the color consistency loss $\mathcal{L}_{col}$ to encourage the color similarity of the high-level features between the input and resulting images:

$$\begin{aligned}&\mathcal{L}_{col}(EX_{color}) \\ &\quad = \mathbb{E}_{(\boldsymbol{c},\boldsymbol{e})\sim X_{data}}[\|EX_{color}(\boldsymbol{c}) - EX_{color}(\boldsymbol{e})\|_1]\end{aligned} \tag{9}$$

In addition, we define the content loss $\mathcal{L}_{con}$ to encourage the color feature of the generated embroidery images to approximate those of the corresponding input content images:

$$\begin{aligned}&\mathcal{L}_{con}(E^L_{lum}, E^{ab}_{chr}, G^{Lab}) \\ &\quad = \mathbb{E}_{\boldsymbol{c}\sim X_{data}}[\|EX_{color}(\boldsymbol{c}) - EX_{color}(\hat{\boldsymbol{e}}_{fake})\|_1]\end{aligned} \tag{10}$$

3.4 Training & Testing

The details of the two-step training pipeline are shown in Fig. 5. In Step 1, we use a reconstruction network to retain as many of the original image features as possible. The distribution of the latent texture code $q(E^L_{emb}(\boldsymbol{e}^L_{st}, st)|\boldsymbol{e}^L_{st}, st)$ is encouraged to approximate a random prior Gaussian distribution $\mathcal{N}(0, \boldsymbol{I})$. In Step 2, we use the prior Gaussian distribution $\boldsymbol{s} \sim \mathcal{N}(0, \boldsymbol{I})$ instead of the latent texture code $E^L_{emb}(\boldsymbol{e}^L_{st}, st)$ to generate embroidery without the dataset. The GAN framework is used to generate the embroidery images $\hat{\boldsymbol{e}}_{fake|st}$ and reconstruct $\boldsymbol{s}$ to $\hat{\boldsymbol{s}}_{recon|st}$.

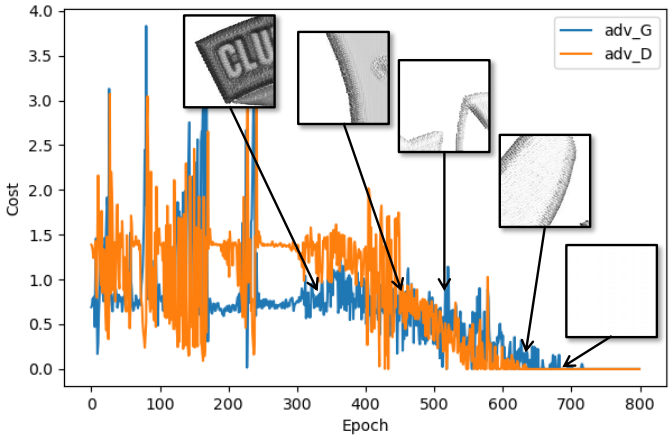


Fig. 6: Due to the vanishing gradient problem, the embroidery texture of output images gradually disappears. Hence, the generator will eventually stop working. From left to right, epochs = 387, 413, 491, 537, 679, and 686 in order.

In the testing pipeline, we add a stitch classifier to our framework in the second step of the training pipeline, as shown in Fig. 1. After we pass the input image into the testing network, the stitch classifier separates the content image and corresponding stitch types of the single stitch subregion. Then the corresponding single-stitch texture map is generated through the region-aware texture generation network. Afterwards, we fuse the three single-stitch texture images into a multi-stitch texture image. Finally, we obtain the final resulting image after the color information is added to the multi-stitch texture image through the colorization network.

4 Experimental Results

4.1 Implementation Details

We implement MSEmbGAN with PyTorch [41] and conduct experiments on a computer with a v100 GPU. We use RMSprop optimizer [42] with exponential decay rates $\beta_1 = 0.5$, $\beta_2 = 0.999$, a batch size of 2, and a learning rate of 0.0002. The input and output channels of the encoders and the generators are $E^L_{emb} : 1$, $E^L_{con} : 1$, $G^L : 1$, $E^L_{lum} : 1$, $E^{ab}_{chr} : 2$, and $G^{Lab} : 3$. The color feature extractor, pretrained for 200 epochs with color consistency loss, produces color consistency loss to jointly optimize the framework. We train the MSEmbGAN until 800 epochs or upon network convergence. For proper training of the region-aware texture generation network, we set the hyper-parameters to $\lambda_{KL} = 0.001$ in Eq. (3) and $\lambda_{latent} = 10$ in Eq. (6).

As shown in Fig. 6, during network training, we find that the vanishing gradient problem rose when $epochs > 470$. To solve this problem, we use the WGAN [40] instead of the regular GAN with the RMSprop [42] optimizer instead of Adam [43]. This framework can make the discriminator Lipschitz continuous. The color feature extractor consists of seven convolution layers with a stride of 2 and a kernel size of 3×3, followed by one convolution layer with a stride of 1 and a kernel size of 1×1. Besides, we add a spectral normalization block [44] for each convolution layer. For detailed network architecture, please refer to the supplemental material.

4.2 Multi-Stitch Embroidery Dataset

We create more than 30K images by the professional embroidery software (Wilcom 9.0) including the embroidery images and the corresponding content images. All images are resized to the resolution of 256×256. We contribute our multi-stitch embroidery dataset to the research community. Our dataset is annotated with one multi-stitch type and three single-stitch types, and the embroidery images are rendered by professional embroidery design software, as shown in Fig. 7. The dataset contains a total of 30K aligned or unaligned embroidery and content images. The detailed distributions of different stitches are shown in Fig. 8. For making our multi-stitch embroidery dataset, the steps for embroidery dataset image-making are as follows:

Draw content image. Before constructing an embroidery plate, our embroidery artists must draw a content image containing embroidery color information as a template. Most content images are simple in color and unambiguous in shape, allowing for faster network convergence.

Stitch Design. For content images with different shapes, a stitch must be selected to fill each region. Embroidery designers match an appropriate stitch type to the shape of each region. Additionally, each stitch's related parameters (e.g., spacing and direction) must be reasonably set for the subsequent embroidery rendering task.

Create Embroidery Dataset. Our embroidery designers use professional embroidery software (Wilcom 9.0) to design and create embroidery patterns and render corresponding embroidery images. These results are the embroidery images in the dataset and the ground-truth images in our work. See more information about our dataset in the supplemental material.

4.3 Quantitative and Qualitative Comparision

4.3.1 *Quantitative Evaluation*

We compare our method with the recently published stylization methods: Pix2Pix [13], CycleGAN [15], MUNIT [14], and DRIT++ [16]. We quantify the comparison results and calculate the learned perceptual image patch similarity (LPIPS) [45] and Fréchet inception distance (FID) [46], as shown in Table 1. Specifically, to better measure human perceptual similarity, we calculate the LPIPS distance on the entire test set to measure the perceptual similarity between the generated embroidery images and the real ones. Compared with other methods, our method had a lower LPIPS distance, which means our resulting embroidery images are closer to the real embroidery images perceptually. We also use FID to measure the feature distribution of our generated embroidery images and ground truth. We also evaluated the FID scores, and the results show that the embroidery image generated by our MSEmbGAN is closest to the ground truth.

We use our method and four comparison methods to generate 100 images respectively and calculate the time required by each method for generate one image. In order to enhance the stitch styles in the resulting embroidery images, we added some extra modules compared to other methods.

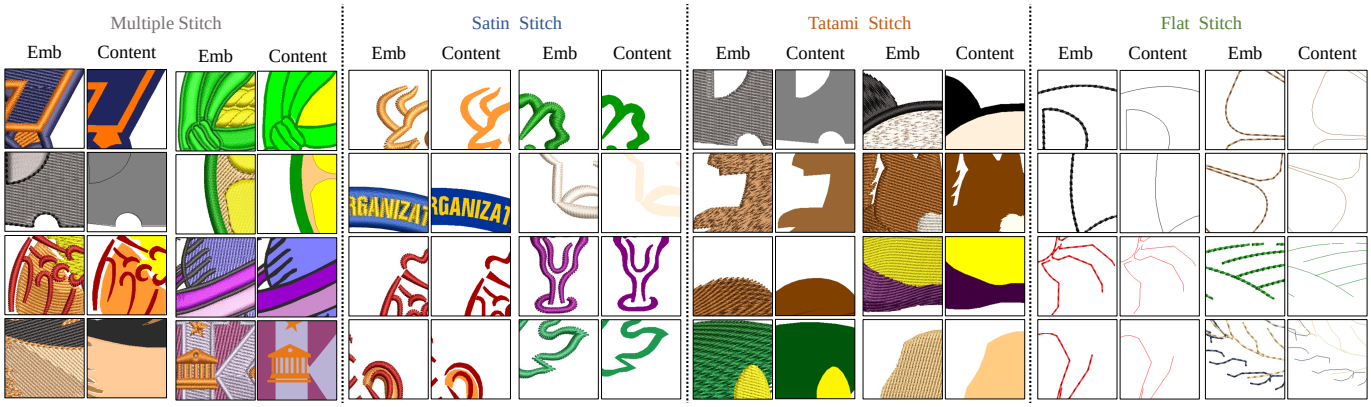


Fig. 7: Images of our Multi-stitch Embroidery Dataset is annotated with four kinds of labels corresponding to three kinds of single-stitch types (i.e., satin, tatami, and flat) and a multi-stitch type that refers to a mixture of the three single-stitch types. Our dataset has aligned and unaligned parts, and the aligned part includes the content and corresponding embroideries.

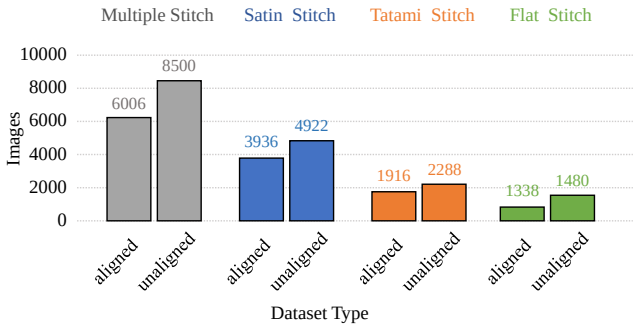


Fig. 8: Data distribution of the multi-stitch embroidery dataset. Each stitch type in our dataset contains paired and unpaired parts.

TABLE 1

Average LPIPS and FID distance between ground truth images and generated embroidery images on the entire test dataset of the four comparison methods, the two ablation models and our MSEmbGAN on the entire test dataset.

Methods	LPIPS	FID
CycleGAN [15]	0.296	153.49
Pix2Pix [13]	0.432	239.89
MUNIT [14]	0.332	146.14
DRIT++ [16]	0.305	171.15
w/o C_{reg}, G_{slc}	0.318	146.43
w/o SG, SC	0.305	138.25
MSEmbGAN	**0.262**	**122.44**

All testing was performed on an NVIDIA GeForce RTX 2060 GPU with images at a resolution of 256x256 pixels. Despite these enhancements, the results in Table 2 show that our MSEmbGAN does not have a large speed gap compared to other models.

TABLE 2

Average time consuming of the four comparison methods and our method.

Methods	consuming / ms
CycleGAN [15]	214
Pix2Pix [13]	187
MUNIT [14]	230
DRIT++ [16]	260
MSEmbGAN	**243**

4.3.2 *Qualitative Evaluation*

We further compare the embroidery images synthesized by our MSEmbGAN, Pix2Pix [13], CycleGAN [15], Multimodal UNsupervised Image-to-image Translation (MUNIT) [14] and DRIT++ [16], as shown in Fig. 9. Note that all methods were trained on our multi-stitch embroidery dataset.

According to the results in Fig. 9, the resulting image of CycleGAN [15] tends to lose color features (i.e., color shift). Although the large regions are filled with embroidery patterns, the textures are missing. The resulting images of CycleGAN lost the texture features of the tatami stitch, which means the accuracy of CycleGAN is lower than that of the supervised method, and CycleGAN is unable to restore embroidery stitches. The resulting images of the Pix2Pix method [13] only retain the content features of the input image and do not look like embroidery images, especially in terms of texture features and visible artifacts. Moreover, the results have almost no embroidery textures and have lost parts of the color information. With MUNIT [14], the textures of the generated embroidery images are relatively smooth and lack the proper embroidery textures. Because the model of MUNIT cannot accurately predict the contours of the embroidery pattern, some noise is generated. For more complex patterns, contours are incorrectly recognized, and some color information is distorted. In some light-colored regions, the embroidery texture cannot be effectively synthesized. The results of DRIT++ [16] only retain a small part of the embroidery texture, and the color features are

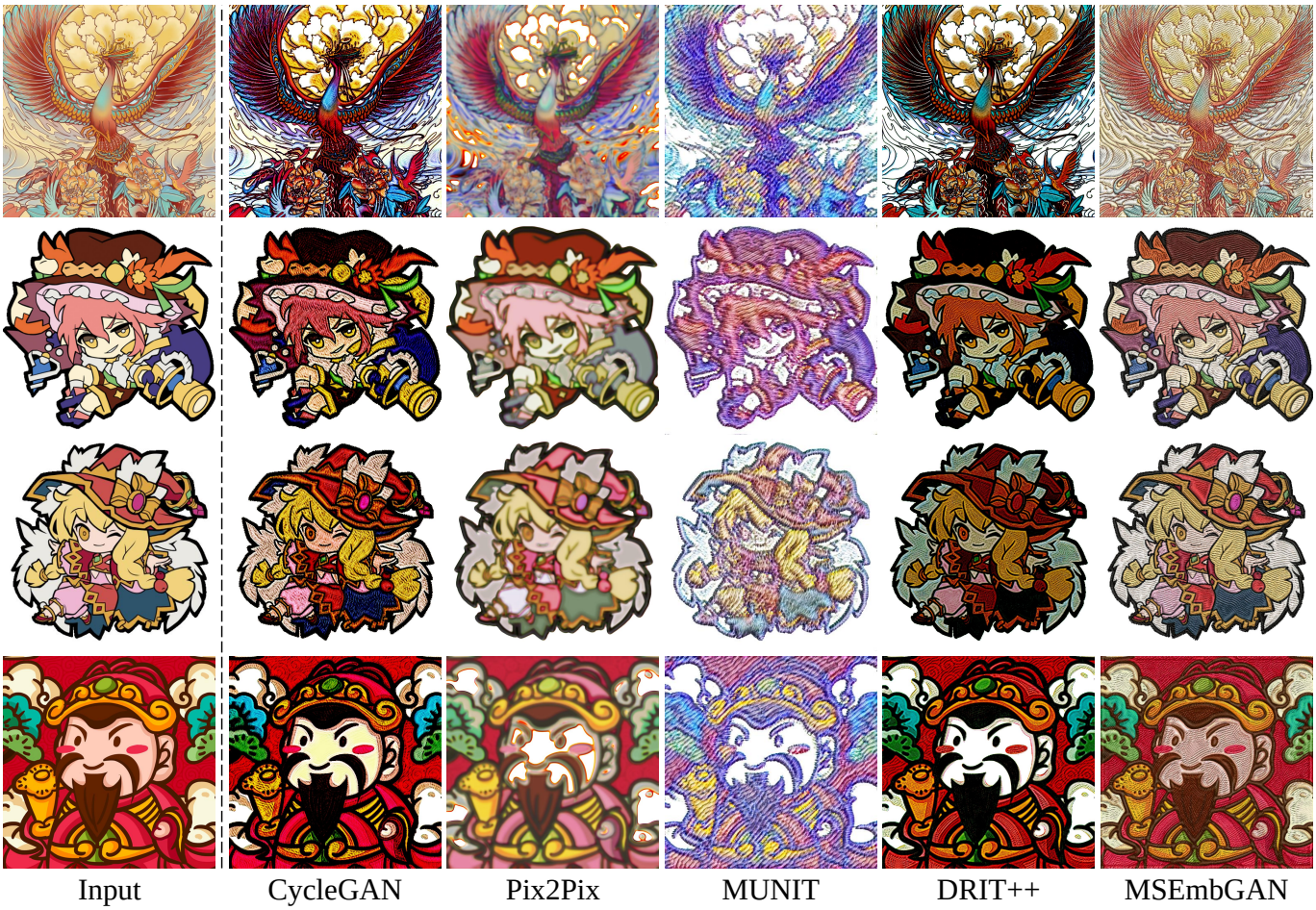


Fig. 9: Comparison of stitch styles generated by MSEmbGAN and four other style-transfer networks (i.e., CycleGAN [15], Pix2Pix [13], MUNIT [14], and DRIT++ [16]). We use the region-aware texture generation network to maintain stitch texture authenticity and color fidelity, making the results generated by our MSEmbGAN possess highly diverse stitch textures. The results of MSEmbGAN are better than those of existing methods in terms of texture and color. Specifically, the textures of our results are more similar to real embroidery textures, and the color are closer to those of the input images.

(a) Input (b) Guan et al. 2021 (c) MSEmbGAN

Fig. 10: Comparison with the state-of-the-art embroidery synthesis method: (a) input image; (b) result presented in the original paper of Guan et al. [7]; (c) result of our MSEmbGAN with the same input.

completely lost. The texture and color styles of the original embroidery image are obvious, but due to the interference among features, the common end-to-end network cannot retain the original color features well, which leads to messy textures [13] and inconsistent colors [16]. Moreover, the difficult training process of DRIT++ often leads to mode collapse. Notably, the baseline DRIT++ lacks texture feature processing functions, so the style of the generated embroidery image is randomly selected. Furthermore, the textures of different stitches cannot be accurately reflected according to the texture style of the input embroidery image.

As shown in Fig. 9, at present, there are varying degrees of color-shift issues when using neural networks to generate embroidery images. Since the brightness of our resulting images are slightly adjusted (See Section 3.2.2), the color of the resulting images generated by our MSEmbGAN are the closest to the input image among all the comparison methods. In terms of the embroidery texture and color features, the resulting images of our MSEmbGAN are most similar to the ground-truth images (see Fig. 9). Our method is even superior to SOTA embroidery synthesis methods [7] because it maximizes the color features of the input image and forces the resulting textures to become clear and realistic, as shown in Fig. 10. Please refer to the supplemental material for additional qualitative evaluation.

Our previous work, [24], is the state-of-the-art method

TABLE 3

Result of our user study, where higher score means better quality. Row 1 and 2 represent the mean and standard error of the embroidery quality score, row 3 and 4 represent the mean and standard error of the color quality score, and row 5 and 6 represent the mean and standard error of the image quality score.

Methods	CycleGAN [15]	Pix2Pix [13]	MUNIT [14]	DRIT++ [16]	MSEmbGAN
Embroidery quality, mean / std	2.670 / 0.848	2.696 / 0.810	2.367 / 0.819	2.822 / 0.785	**3.834** / **0.721**
color quality, mean / std	2.659 / 0.905	2.937 / 0.750	2.751 / 0.812	2.347 / 0.847	**3.734** / **0.702**
Image quality, mean / std	2.739 / 0.865	2.438 / 0.878	2.731 / 0.822	2.865 / 0.763	**3.711** / **0.714**

(a) Input (b) Yang et al. 2022 (c) MSEmbGAN

Fig. 11: Comparison with our previous work: (a) input image; (b) results of our previous method [24]; (c) results of our MSEmbGAN. We zoom in and compare some regions of the images in the yellow boxes in (b) and (c).

for embroidery generation using neural networks. Although the color of the resulting images of [24] is consistent with the color of the input image, the results of MSEmbGAN also incorporate multi-stitch features. As shown in Fig. 11, the texture of the resulting images generated by MSEmbGAN has more diverse stitch types.

Performance on high-resolution images. We apply our model to high-resolution image generation to synthesize resulting images with high-fidelity colors and textures, as shown in Fig. 12. Due to the application of fully convolutional network, MSEmbGAN can process input images of any size.

Illustration of Controllability As shown in Fig. 13, the distribution of stitch types in the synthetic embroidery images can be adjusted by changing the region width threshold λ_w.

4.4 User Study

The evaluation of embroidery image quality is often affected by users' subjective factors, so we conduct user studies to get subjective feedback from users. We prepare 14 images, and each of them is processed using our method and four SOTA methods: [13], [15], [14], [16]. We invite 25 candidates to score each image in the range of 1–5 points according to the following criteria:

Embroidery quality: whether the resulting images have embroidery-related features and vivid textures.

color quality: the color similarity between input and resulting images.

Image quality: the degree of texture distortions, color shifts, high-frequency noise, and other artifacts.

We collected 5,250 scores and calculated the average and standard error of each criteria, as shown in Table 3. The results clearly show that our method is superior on all three criteria. The results of the user study indicate that our region-aware texture generation network does a great job of generating various embroidery stitch textures stably, which leads to superior embroidery quality scores. In addition, with our colorization network, our method can also obtain the highest color quality evaluation scores. Because our method adds global information when generating local embroidery textures, we have the smallest standard error in all three criteria, which indicates that the comprehensive performance of our method is more stable. This is due to the fact that the stitch latent code generator can successfully stabilize the stitch texture generation process in local regions.

4.5 Ablation Study

We conduct an ablation study to identify the impacts of the stitch latent code generator module with stitch classifier and the colorization network with color feature extractor and content loss. The quantitative evaluation results of our ablation study are shown in Table 1.

4.5.1 Effect of Stitch Classifier and Stitch Latent Code Generator

In this task, we ablate the region-aware texture generation network (i.e., stitch classifier, stitch latent code generator) and retain other parts of MSEmbGAN as shown in Fig. 14(b). Specifically, We use the original baseline [16] instead of the region-aware texture generation network. We removed the C_{reg} and stitch type st and use the embroidery encoder E^L_{emb} instead of stitch latent code generator G_{slc}.

The quantitative evaluation results of this ablation study are shown in Table 1, which reflects the efficacy of these modules. The stitch classifier is the key to classifying regions according to shape features, and it allows the texture generation network to accurately capture the specific stitch features of each color region. Meanwhile, the stitch latent code generator generates the corresponding texture latent code according to the input stitch type. Without the stitch classifier and the stitch latent code generator, the resulting embroidery images synthesized by the network suffer from two major problems. First, the texture styles in the resulting images are singular, and the multi-stitch characteristics are not preserved. Second, we observe an unstable and abnormal texture generation process.

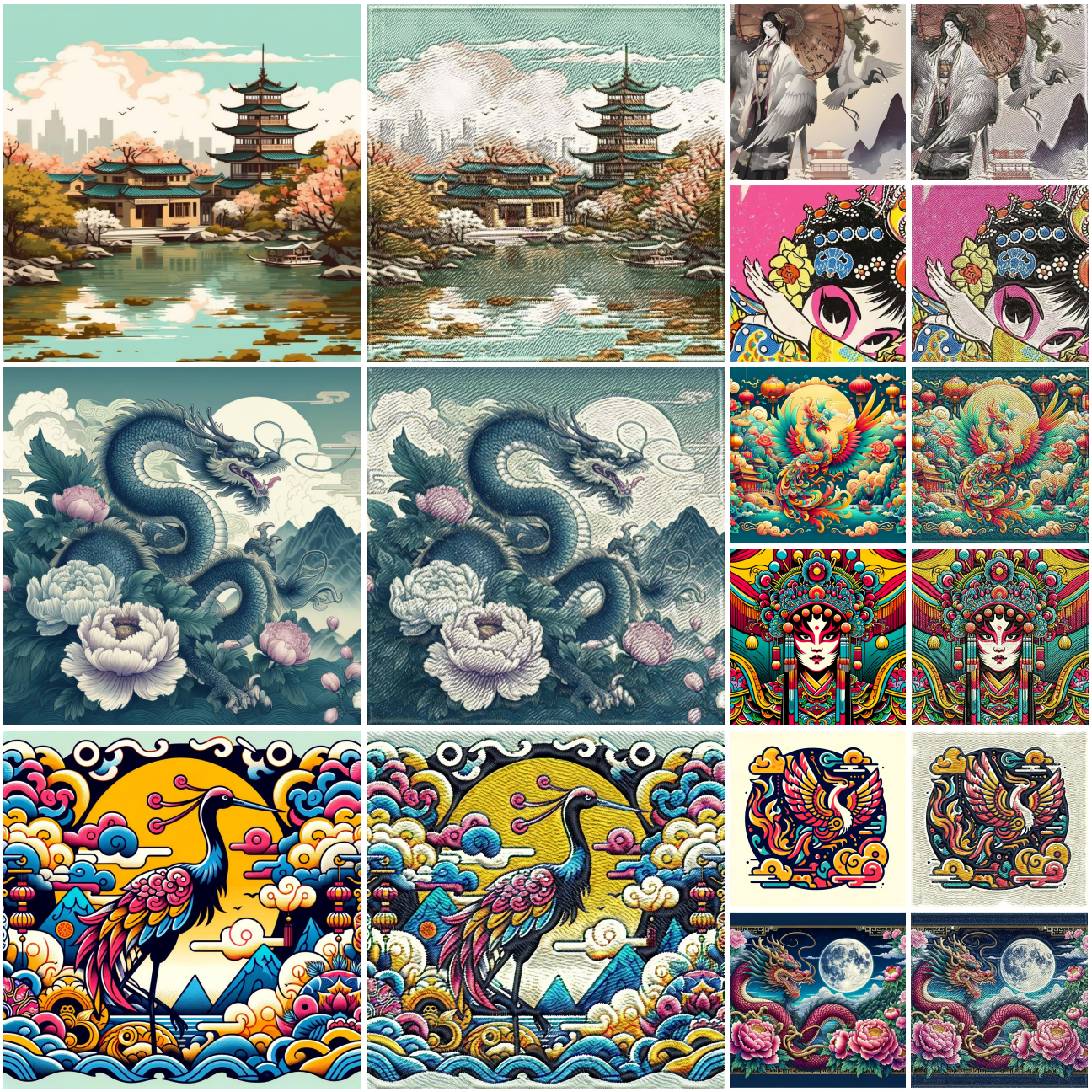


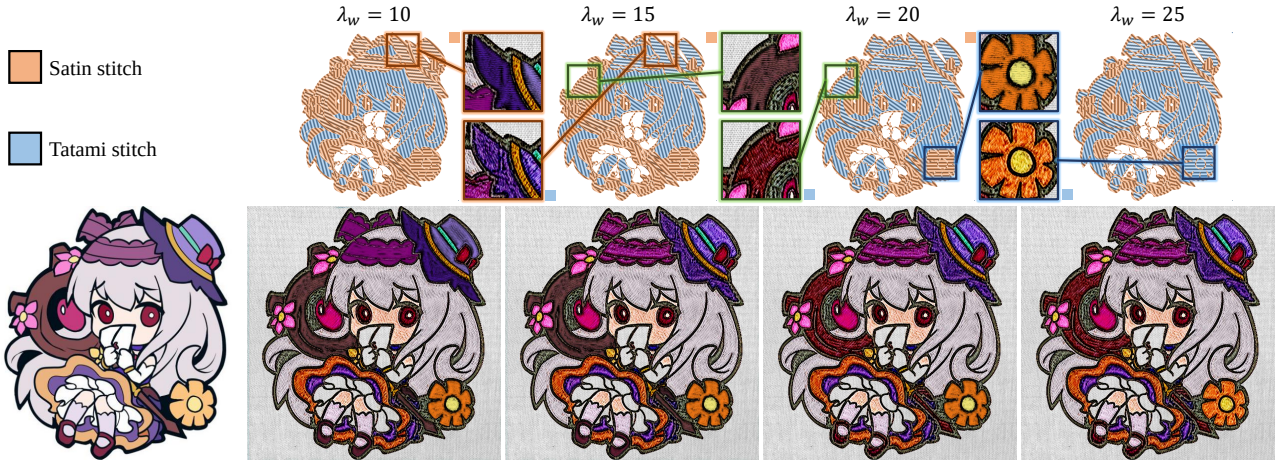
Fig. 12: Resulting images of our method with different inputs. Given the high-resolution images, our network demonstrates great performance. The resolution of the input and output images are all 1024×1024. The resulting images we generate are similar in texture and color to the real image, especially the colors are quite close to that of input images. All the resulting images contain various embroidery stitch styles. For a better visual effect, we fill the white regions in the result with cloth patterns.

4.5.2 *Effect of colorization network and color consistency*

As shown in Fig. 14, we ablate the colorization network and its color consistency (i.e., color feature extractor and color consistency loss), as shown in Fig. 14(c). Furthermore, we remove the colorization network (CN) with color consistency (CC), including the color feature extractor EX_{color} and color consistency loss. Meanwhile, we set the input and output of the region-aware texture generation network to the lab channel.

Without the colorization network and the corresponding color consistency, the embroidery results synthesized by our MSEmbGAN cannot maintain the color features, which results in obvious color shifts. Specifically, there is a great difference in the color distribution between the resulting images and the input images.

The colorization network helps the preservation of color information on the basis of the region-aware texture generation network. Meanwhile, the color consistency contains a pretrained color feature extractor EX_{color} and color consistency loss $\mathcal{L}_{col}$. These modules are used to guide the



(a) Input (b) Result in different λ_w

Fig. 13: Stitch distribution is adjusted by the region width threshold: λ_w = 10, 15, 20, and 25 from left to right. As the threshold increases, the distribution of the tatami stitch becomes larger. Users can balance the distribution of the tatami and satin stitches by adjusting the region width threshold. The filling textures in some regions vary with the threshold value λ_w. We filled the white regions in the results with cloth patterns.

color features of the resulting images to approximate the corresponding input images.

5 Conclusion

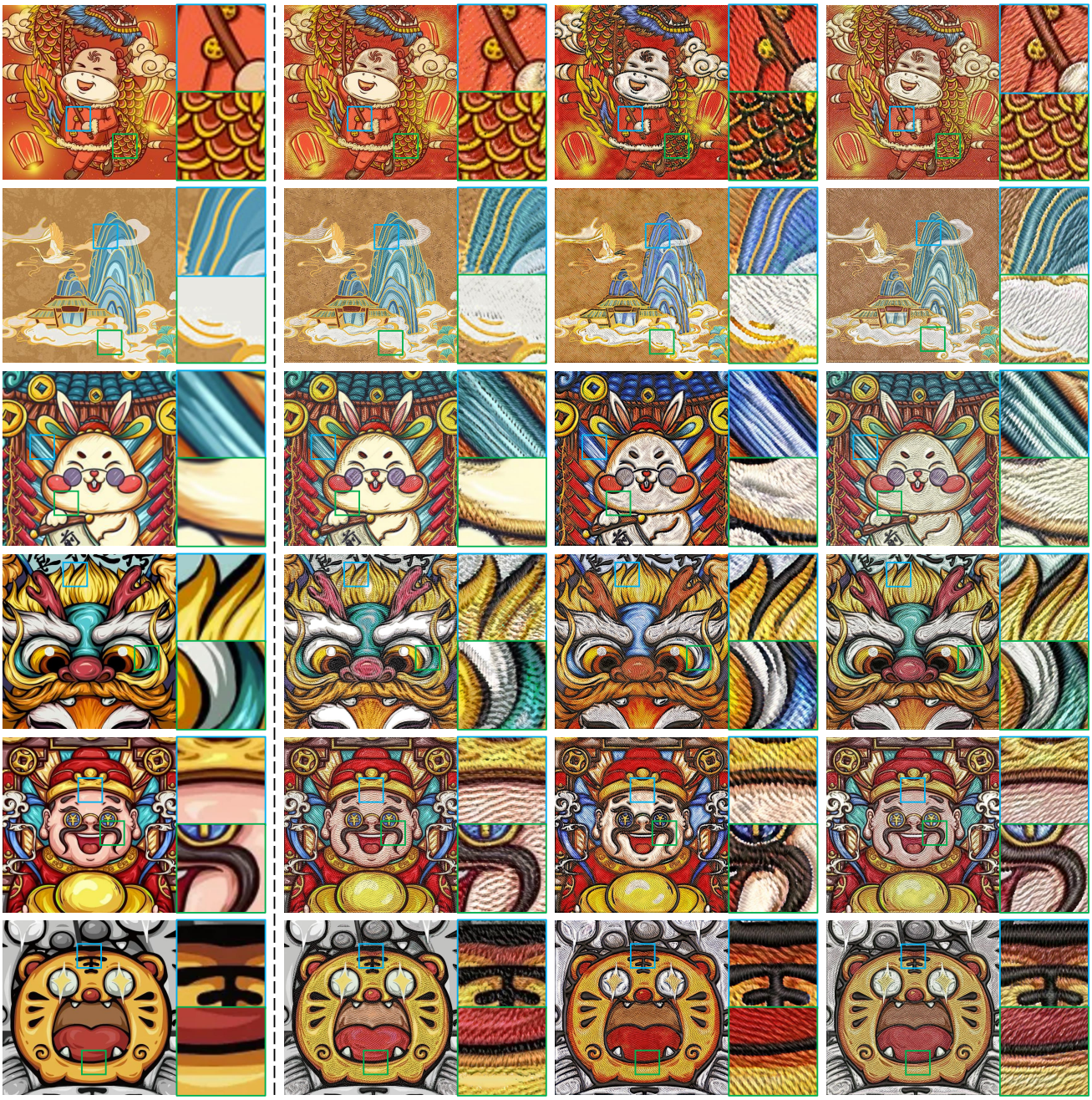
In this paper, we propose the field's first CNN-based GAN for multi-stitch embroidery synthesis. Our MSEmb-GAN model can generate realistic embroidery images containing diverse (multiple) stitches synthesized from input images. In the backbone of GAN, we design a region-aware texture generation network that learns an appropriate stitch for each color region according to its shape features and synthesizes the corresponding embroidery texture. We also proposed a colorization network that maintains the color consistency between synthesized and input images. Moreover, our model contributes a new high-quality multi-stitch embroidery dataset containing 30K multi-stitch embroidery images with mixed or single-stitch styles. The qualitative and quantitative results show that our method outperforms SOTA models in generating high-quality multi-stitch embroidery images with more realistic textures. Moreover, our method affords more diverse stitches with clearer textures and more stable colors for the resulting images. However, our method sometimes fails when dealing with complex images, and we plan to address the problem in our future work. We also plan to extend our work so that the model can handle more comprehensive embroidery problems, such as 3D embroidery synthesis.

Acknowledgement

This work was supported in part by the National Science and Technology Council under Grant No. 113-2221-E-006 - 161-MY3, Taiwan, National Natural Science Foundation of China under grant No. 62272298, and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102).

References

[1] M. A. Beg and J. Y. Yu, "Generating embroidery patterns using image-to-image translation," *arXiv preprint arXiv:2003.02909*, pp. 1–14, 2020.
[2] Z. Wei and Y. C. Ko, "Segmentation and synthesis of embroidery art images based on deep learning convolutional neural networks," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 36, no. 11, pp. 1–17, 2022.
[3] X. Chen, M. McCool, A. Kitamoto, and S. Mann, "Embroidery modeling and rendering," in *Graphics Interface*, 2012, pp. 131–139.
[4] D. Cui, Y. Sheng, and G. Zhang, "Image-based embroidery modeling and rendering," *Computer Animation and Virtual Worlds*, vol. 28, no. 2, pp. 1–12, 2017.
[5] D. Baeva, "Using lindenmayer systems for generative modeling of graphic concepts, set in elements of bulgarian folklore embroidery," in *International Conference on Computer Systems and Technologies*, 2019, pp. 234–239.
[6] D. Ma, M. Cheng, D. Zheng, X. Fan, W. Wang, and J. Fang, "Development of sichuan brocade with imitating embroidery effect based on free-floats interlacing weave," *Journal of Textile Science and Technology*, vol. 6, no. 1, pp. 11–18, 2019.
[7] X. Guan, L. Luo, H. Li, H. Wang, C. Liu, S. Wang, and X. Jin, "Automatic embroidery texture synthesis for garment design and online display," *The Visual Computer*, vol. 37, no. 9, pp. 2553–2565, 2021.
[8] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, pp. 1–16, 2015.
[9] Y. Men, Z. Lian, Y. Tang, and J. Xiao, "A common framework for interactive texture transfer," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6353–6362.
[10] Y. Shu, R. Yi, M. Xia, Z. Ye, W. Zhao, Y. Chen, Y.-K. Lai, and Y.-J. Liu, "GAN-based multi-style photo cartoonization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 10, pp. 3376–3390, 2022.
[11] F. Han, S. Ye, M. He, M. Chai, and J. Liao, "Exemplar-based 3D portrait stylization," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–14, 2021.
[12] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, "Neural style transfer: A review," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 11, pp. 3365–3385, 2019.
[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134.
[14] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *European Conference on Computer Vision*, 2018, pp. 172–189.



(a) Input (b) w/o C_{reg},G_{slc} (c) w/o CN,CC (d) MSEmbGAN

Fig. 14: Ablation study on the stitch classification algorithm, C_{reg} of the stitch classifier and latent code generator G_{slc}: (a) Input image; (b)The region-aware texture generation network was ablated. (c) The colorization network (CN) and its color consistency (CC) was ablated. (d) The resulting images synthesized using our whole MSEmbGAN.

[15] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *International Conference on Machine Learning*, 2017, pp. 1857–1865.

[16] H.-Y. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. Singh, and M.-H. Yang, "DRIT++: Diverse image-to-image translation via disentangled representations," *International Journal of Computer Vision*, vol. 128, no. 10, pp. 2402–2417, 2020.

[17] J. Zhou, Z.-x. Sun, and K.-w. Yang, "A controllable stitch layout strategy for random needle embroidery," *Journal of Zhejiang University SCIENCE C*, vol. 15, no. 9, pp. 729–743, 2014.

[18] K. Yang, Z. Sun, C. Ma, and W. Yang, "Paint with stitches: A random-needle embroidery rendering method," in *Computer Graphics International*, 2016, pp. 9–12.

[19] K. Yang and Z. Sun, "Paint with stitches: A style definition and image-based rendering method for random-needle embroidery," *Multimedia Tools and Applications*, vol. 77, no. 10, pp. 12 259–12 292, 2018.

[20] W. Qian, D. Xu, J. Cao, Z. Guan, and Y. Pu, "Aesthetic art simulation for embroidery style," *Multimedia Tools and Applications*, vol. 78, no. 1, pp. 995–1016, 2019.

[21] K. Yang, J. Zhou, Z. Sun, and Y. Li, "Image-based irregular needling embroidery rendering," in *International Symposium on Visual Information Communication and Interaction*, 2012, pp. 87–94.

[22] C. Ma and Z. Sun, "StitchGeneration: Modeling and creation of random-needle embroidery based on Markov chain model,"

Multimedia Tools and Applications, vol. 78, no. 23, pp. 34 065–34 094, 2019.
[23] W. Qian, J. Cao, D. Xu, R. Nie, Z. Guan, and R. Zheng, "CNN-based embroidery style rendering," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 34, no. 14, pp. 2 059 045:1–2 059 045:24, 2020.
[24] C. Yang, X. Hu, Y. Ou, S. Zhong, T. Peng, L. Zhu, P. Li, and B. Sheng, "Unsupervised embroidery generation using embroidery channel attention," in *Proceedings of the 18th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, 2022, pp. 1–8.
[25] Q. Shen, D. Cui, Y. Sheng, and G. Zhang, "Illumination-preserving embroidery simulation for non-photorealistic rendering," in *International Conference on Multimedia Modeling*, 2017, pp. 233–244.
[26] Y. Takahashi and T. Fukusato, "Stitch: A interactive design system for hand-sewn embroidery," in *ACM SIGGRAPH Posters*, 2018, pp. 17:1–17:2.
[27] Y. Liu, J. Wright, and A. Alvarado, "Making beautiful embroidery for "Frozen 2"," in *ACM SIGGRAPH Talks*, 2020, pp. 73:1–73:2.
[28] W. Cai, H. Zhang, X. Xu, S. He, K. Zhang, and J. Qin, "Contextual-assisted scratched photo restoration," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 10, pp. 5458–5469, 2023.
[29] C. Xu, W. Qu, X. Xu, and X. Liu, "Multi-scale Flow-based occluding effect and Content separation for cartoon animations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 9, pp. 4001–4014, 2023.
[30] W. Xiao, C. Xu, J. Mai, X. Xu, Y. Li, C. Li, X. Liu, and S. He, "Appearance-preserved portrait-to-anime translation via Proxy-guided domain adaptation," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–17, 2022.
[31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.
[32] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, pp. 1–7, 2014.
[33] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 852–863, 2021.
[34] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," *Advances in Neural Information Processing Systems*, vol. 30, pp. 1–9, 2017.
[35] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations*, 2014, pp. 1–14.
[36] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *International Conference on Machine Learning*, 2016, pp. 1558–1566.
[37] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in Neural Information Processing Systems*, vol. 28, pp. 3483–3491, 2015.
[38] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "CVAE-GAN: Fine-grained image generation through asymmetric training," in *IEEE International Conference on Computer Vision*, 2017, pp. 2745–2754.
[39] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *European Conference on Computer Vision*, 2008, pp. 705–718.
[40] A. Martín, C. Soumith, and B. Léon, "Wasserstein GAN," *arXiv preprint arXiv:1701.07875*, pp. 1–32, 2017.
[41] 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," *NeurIPS Workshop*, pp. 1–4, 2017.
[42] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, pp. 1–14, 2016.
[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015, pp. 1–15.
[44] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *International Conference on Learning Representations*, 2018, pp. 1–26.
[45] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595.
[46] D. Dowson and B. Landau, "The fréchet distance between multivariate normal distributions," *Journal of Multivariate Analysis*, vol. 12, no. 3, pp. 450–455, 1982.

Xinrong Hu received the Ph.D. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 2008. She is currently a Full Professor with the School of Computer Science and Artificial Intelligence, Wuhan Textile University, Wuhan, China, and a Visiting Scholar with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. She has published more than 60 papers in journals and refereed conferences, and has served on many program committees. Her current research interests include computer graphics, media computing, VR/AR, and deep learning.

Chen Yang received the B.Eng. degree in computer science from the Wuhan University, Wuhan, China, in 2018. He is currently pursuing the M.Eng. degree in computer science with the School of Computer Science and Artificial Intelligence, Wuhan Textile University, Wuhan, China. His current research interests include embroidery synthesis, texture generation, deep learning, and computer graphics.

Fei Fang received the B.Eng. degree in computer science and technology from the Zhengzhou University, Zhengzhou, China, in 2011, the M.Eng. degree in computer application technology from the Guangxi University, Nanning, China, in 2014, and the Ph.D. degree in computer application technology from the Wuhan University, Wuhan, China, in 2022. She is currently a Lecturer with the School of Computer Science and Artificial Intelligence, Wuhan Textile University, Wuhan, China. Her current research interests include computer graphics and deep learning.

Jin Huang received the Ph.D. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 2012. He is currently a Lecturer with the School of Computer Science and Artificial Intelligence, Wuhan Textile University, Wuhan, China. His current research interests include artificial intelligence and style transformation.

Ping Li (Member, IEEE) received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Hong Kong, in 2013. He is currently an Assistant Professor with the Department of Computing and an Assistant Professor with the School of Design, The Hong Kong Polytechnic University, Hong Kong. He has published over 200 top-tier scholarly research articles (e.g., TVCG, TPAMI, TIP, TNNLS, TMI, TMM, TCSVT, TCYB, TBME, TSMC, TII, AAAI, CVPR, ICCV, NeurIPS), pioneered several new research directions, and made a series of landmark contributions in his areas. He has an excellent research project reported by the *ACM TechNews*, which only reports the top breakthrough news in computer science worldwide. More importantly, however, many of his research outcomes have strong impacts to research fields, addressing societal needs and contributed tremendously to the people concerned. His current research interests include image/video stylization, colorization, artistic rendering and synthesis, realism in non-photorealistic rendering, computational art, and creative media.

Bin Sheng (Member, IEEE) received the B.A. degree in English and the B.Eng. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 2004, and the M.Sc. degree in software engineering from the University of Macau, Macau, in 2007, and the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Hong Kong, in 2011. He is currently a Full Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. His current research interests include virtual reality and computer graphics. He is an Associate Editor of the *IEEE Transactions on Circuits and Systems for Video Technology*.

Tong-Yee Lee (Senior Member, IEEE) received the Ph.D. degree in computer engineering from Washington State University, Pullman, in 1995. He is currently a Chair Professor with the Department of Computer Science and Information Engineering, National Cheng-Kung University (NCKU), Tainan, Taiwan. He leads the Computer Graphics Group, Visual System Laboratory, NCKU (http://graphics.csie.ncku.edu.tw). His current research interests include computer graphics, non-photorealistic rendering, medical visualization, virtual reality, and media resizing. He is a Senior Member of the IEEE and a Member of the ACM. He is an Associate Editor of the *IEEE Transactions on Visualization and Computer Graphics*.