electronics

MDPI

Article

An End-to-End Generation Model for Chinese Calligraphy Characters Based on Dense Blocks and Capsule Network

Weiqi Zhang [1], Zengguo Sun [1,2,*] and Xiaojun Wu [1,2]

[1] School of Computer Science, Shaanxi Normal University, Xi'an 710119, China; zhangweiqi@snnu.edu.cn (W.Z.); xjwu@snnu.edu.cn (X.W.)
[2] Key Laboratory of Intelligent Computing and Service Technology for Folk Song, Ministry of Culture and Tourism, Xi'an 710119, China
* Correspondence: sunzg@snnu.edu.cn

Abstract: Chinese calligraphy is a significant aspect of traditional culture, as it involves the art of writing Chinese characters. Despite the development of numerous deep learning models for generating calligraphy characters, the resulting outputs often suffer from issues related to stroke accuracy and stylistic consistency. To address these problems, an end-to-end generation model for Chinese calligraphy characters based on dense blocks and a capsule network is proposed. This model aims to solve issues such as redundant and broken strokes, twisted and deformed strokes, and dissimilarity with authentic ones. The generator of the model employs self-attention mechanisms and densely connected blocks to reduce redundant and broken strokes. The discriminator, on the other hand, consists of a capsule network and a fully connected network to reduce twisted and deformed strokes. Additionally, the loss function includes perceptual loss to enhance the similarity between the generated calligraphy characters and the authentic ones. To demonstrate the validity of the proposed model, we conducted comparison and ablation experiments on the datasets of Yan Zhenqing's regular script, Deng Shiru's clerical script, and Wang Xizhi's running script. The experimental results show that, compared to the comparison model, the proposed model improves SSIM by 0.07 on average, reduces MSE by 1.95 on average, and improves PSNR by 0.92 on average, which proves the effectiveness of the proposed model.

Keywords: calligraphy generation; generative adversarial network; capsule network; self-attention; perceptual loss

check for updates

Citation: Zhang, W.; Sun, Z.; Wu, X. An End-to-End Generation Model for Chinese Calligraphy Characters Based on Dense Blocks and Capsule Network. *Electronics* **2024**, *13*, 2983. https://doi.org/10.3390/electronics13152983

Academic Editor: Xenophon Zabulis

Received: 17 May 2024
Revised: 20 July 2024
Accepted: 21 July 2024
Published: 29 July 2024

1. Introduction

Calligraphy is a highly valued art form in Chinese culture, with works by renowned calligraphers held in high regard. However, many pieces written by famous calligraphers have been damaged or lost over time, making authentic forms of calligraphy difficult to trace [1,2]. With the rapid advancements in artificial intelligence technology, computers can now assist in reproducing these works [3]. The application of deep learning is significant in the generation of calligraphy fonts that are damaged or lost. The use of deep learning technology to generate Chinese calligraphy characters can effectively inherit and promote this traditional art form, realize the organic combination of art and technology, provide new ideas and inspirations for the creation of calligraphy, promote the innovation and development of the art of calligraphy, and bring people a new esthetic experience.

In recent years, researchers have started modeling calligraphy characters and training networks to learn the mapping from source-printed characters to target calligraphy characters [4]. The field of Generative Adversarial Networks (GANs) has gained considerable interest. A GAN can generate high-quality images through a competitive learning process that involves a generator and a discriminator. Additionally, several improvements have been made to the original GAN framework, leading to the development of various enhanced versions [5,6].

In the research of calligraphy characters generation, GANs have been widely used. GANs can be trained on a dataset of existing calligraphy to generate characters that mimic the style of the examples. The generator network improves over time by producing higher-quality calligraphy characters as it deceives the discriminator. The use of a GAN in calligraphy character generation has significant applications. Zhang et al. proposed an approach which uses a multi-scale GAN for Chinese calligraphy style transformation [7]. Kong et al. proposed a generative adversarial network model which introduces a component-aware module that can supervise the generator to separate the content at a finer level, leading to significant results in character generation [8]. Li et al. proposed a modified version of the zi2zi calligraphy character generation method [9]. The method introduced residual blocks, context-aware attention, and spectral normalization techniques to enhance the overall visual effect of the generated calligraphy characters. Wang et al. proposed a dual-attention network structure. They embedded this network structure into the encoding and decoding layers of the zi2zi model to effectively enhance the efficiency of calligraphy character generation [10]. However, its reliance on paired real data during the training process poses a major challenge. Obtaining a complete dataset of authentic calligraphy is particularly difficult due to the loss or damage of many valuable works. A Cycle-Consistent Generative Adversarial Network (CycleGAN) is a model that can be trained without pairs of data and has been widely used in generating calligraphy characters [11].

However, generating calligraphy characters with CycleGAN still presents some challenges. The calligraphy characters generated using CycleGAN have discontinuous strokes, resulting in unnatural visual effects. Additionally, the generated characters may not sufficiently resemble authentic calligraphy in terms of style. Further improvements to the model are necessary to address these issues. An end-to-end generation model for Chinese calligraphy characters based on dense blocks and the capsule network is proposed in this paper to address stroke problems and dissimilarity with authenticity in generated calligraphy characters. The generator uses self-attention mechanisms [12] and densely connected blocks [13] to reduce redundant strokes and broken strokes. Additionally, a capsule network [14] is employed to design the discriminator, enhancing its discriminative ability and reducing twisted and deformed strokes. Furthermore, we introduce perceptual loss to enhance the similarity between the generated calligraphy characters and authentic ones [15]. The Chinese calligraphy characters generated using the proposed model in this paper effectively reduce the problems of missing strokes, redundant strokes, and low stylistic similarity with authentic handwriting compared to other models.

The following are the principal contributions of this paper:

(1) A self-attention mechanism and a densely connected module are employed to reduce redundant and missing strokes.
(2) To reduce twisted and deformed strokes, a capsule network and a fully connected network are employed in the design of the discriminator.
(3) Additionally, perceptual loss is introduced to enhance the similarity of calligraphy style between the generated calligraphy and authentic ones.

The rest of this paper is organized as follows. Section 2 discusses the work related to image-to-image translation and calligraphy character generation. Section 3 provides details on the construction of the proposed model. Section 4 presents the experimental design and the analysis of the experimental results, while Section 5 offers the conclusions of the study.

2. Related Work

This section introduces the work related to image-to-image translation and calligraphy generation, respectively. Image-to-image translation is the technique or process of converting an image from one visual style or feature to another. Calligraphy generation refers to the use of computer technology and artificial intelligence algorithms to generate calligraphy fonts.

2.1. Image-to-Image Translation

In the realm of deep learning, image translation is the process of transforming one image into another, typically to modify its style, content, or both [16]. GAN is one of the most widely used models for image translation, comprising a generator and a discriminator [17]. The objective of the generator is to generate new, lifelike images, while the goal of the discriminator is to distinguish real data from the data created using the generator. During the adversarial training process, the generator improves its ability to create images that are difficult for the discriminator to distinguish authentic ones from generated ones. Another noteworthy model is the Variational Autoencoder (VAE) [18], which learns a latent representation of the input data and generates new images via sampling from this latent space. VAEs are recognized for their capacity to generate a wide range of realistic images while maintaining a high level of control over the generation process. Recently, models such as the U-Net [19] have been developed for image-to-image translation tasks. The U-Net architecture comprises an encoder–decoder structure with skip connections to preserve detailed information throughout the translation process. This makes it especially suitable for tasks such as image segmentation, style transfer and super-resolution. Pix2Pix [20] uses conditional GAN to enable paired image-to-image conversion, while CycleGAN [11] addresses unpaired image translation by incorporating a cycle consistency loss mechanism. Tumanyan et al. proposed a new framework that takes text-to-image synthesis to the realm of image-to-image translation [21]. Parmar et al. proposed an image-to-image translation method that can preserve the original image's content without manual prompting [22]. Ko et al. proposed SuperstarGAN, training an independent classifier by using data augmentation techniques to address the overfitting issue in the classification of StarGAN structures [23].

2.2. Calligraphy Generation

Calligraphy generation aims to generate digital versions of traditional Chinese calligraphy using deep learning techniques. This involves replicating the unique brush strokes, styles, and techniques of professional calligraphers by training on large datasets of authentic historical calligraphy. Huang et al. [24] proposed a method based on decomposition rendering, which can achieve style transfer for Chinese calligraphy characters. This method enables both few-shot learning and zero-shot learning. It utilizes a technique known as "base decomposition" to break down calligraphy characters into bases and components, representing them as vectors. It then uses GAN to perform style translation. Gao et al. [25] introduced a GAN-based calligraphy style transfer method that utilizes skeleton translation and stroke rendering to achieve migration between different style fonts. The method utilizes contextual information to improve the semantic and visual consistency of characters. Xiao et al. [26] proposed a model for calligraphy style transfer called CS-GAN, which uses structural alignment to transfer one calligraphy image into another style of calligraphy image. Zhang et al. [27] presented a method that uses multi-scale GAN to achieve calligraphy style transfer. The method begins by generating the style mask of the target calligraphy style. This mask is then merged through multi-scale GAN to generate calligraphy characters with the desired style. Kong Y et al. [28] proposed a GAN that uses a new perception module. The model supervises the generator to decouple content at a finer granularity level in the generation of calligraphy characters. Wen et al. [27] proposed ZiGAN, a small sample style transfer calligraphy character generation model based on CycleGAN. This model does not require paired calligraphy characters and can generate calligraphy characters in a specific style using only a small number of character samples as input. Although these calligraphy character generation methods can generate calligraphy characters, the generated calligraphy characters have problems such as missing strokes, broken strokes, excess ink, and low similarity to authentic ones. Therefore, to address the above problems, it is necessary to construct a new model for calligraphy character generation.

3. Method

This proposed model aims to solve issues such as redundant and broken strokes, twisted and deformed strokes, and dissimilarity with authentic ones. This section introduces the construction details of the proposed model, including the network structure, generator, discriminator and loss function.

3.1. Network Architecture

The network architecture of the proposed model is shown in Figure 1 and consists of three main components: generator, discriminator, and the loss function. During training, the generator receives printed characters and generates calligraphy characters in a specified style. The generated calligraphy characters are then evaluated for authenticity using the discriminator. The loss value between the generated calligraphy characters and the authentic ones is calculated concurrently. The generated calligraphy characters are then adjusted to more closely resemble the authentic ones by reducing the loss of value during training.

The generator in the proposed model has a self-attention mechanism to improve its perception of calligraphy strokes, allowing it to focus on the main stroke information and reduce redundant strokes. Additionally, the generator is enhanced with dense blocks to extract calligraphy stroke features. This helps to reduce the problems of broken strokes, ensuring that the generated calligraphy characters are complete and coherent. A CapsNet and FCN are used to construct the discriminator, allowing the model to more accurately extract the positional information of calligraphy strokes, thereby reducing twisted and deformed strokes. Perceptual loss $\mathcal{L}_{\text{per}}$ is introduced to further improve the calligraphy style recognition ability of the model. This leads to the generated calligraphy style being closer to the authentic one.

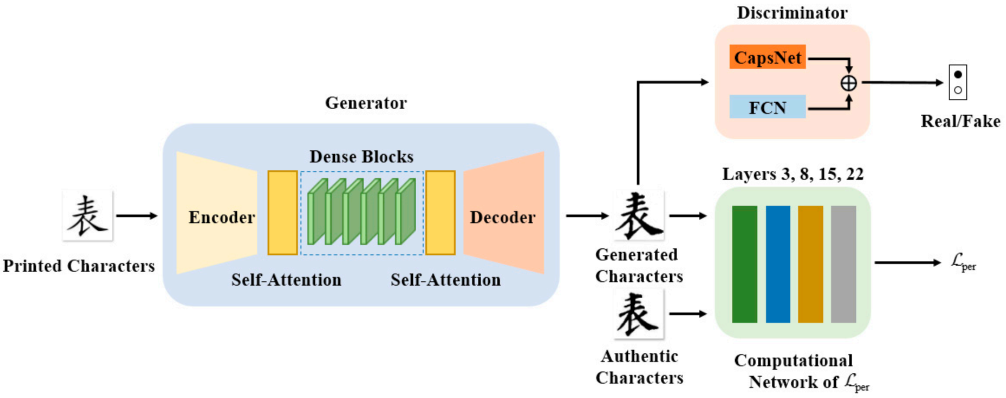


Figure 1. Overall network structure (The Chinese characters are named Biao).

3.2. Generator

The proposed model consists of two generators: a printed character generator and a calligraphy character generator. The printed character generator is responsible for transferring the calligraphy character into the printed character, while the calligraphy character generator transfers the printed character into the calligraphy character. Both generators share the same network architecture. Figure 2 shows how printed characters are transferred to calligraphy characters, using the calligraphy character generator as an example.

3.2.1. Generator Structure

To address the problem of redundant and broken strokes in generated calligraphy characters, we propose the use of densely connected blocks (dense blocks) and self-attention mechanisms in the design of the generator. Dense blocks enhance feature propagation, effectively improving the ability of the generator for capture and restore stroke details. Self-attention mechanisms enable the model to precisely identify and suppress unnecessary strokes during the generation process.

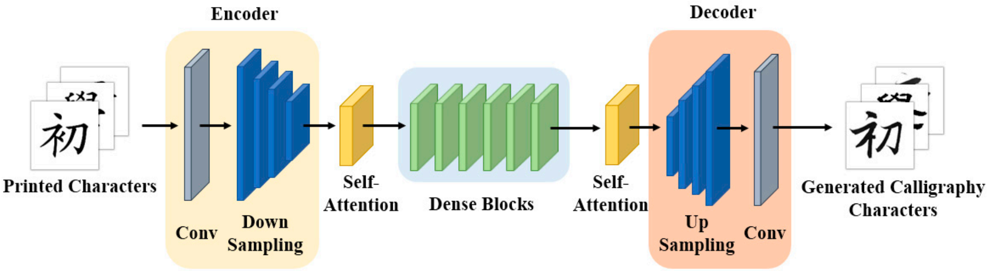


Figure 2. Network structure of calligraphy character generator (The Chinese characters are named Chu).

In the process of generating calligraphy characters, the calligraphy character generator extracts character features from the printed characters through the encoder layers. The encoder layer includes convolutional layers and down-sampling layers. Then, the main stroke information from these features is extracted via the self-attention mechanism. Dense blocks transfer the features of printed characters to the features of calligraphy characters, while maintaining the continuity and artistry of the strokes. After translation, self-attention mechanisms are used to extract the main stroke information of the calligraphy characters, ensuring stroke precision in the generated calligraphy characters. Finally, the calligraphy characters features are transferred into generated calligraphy characters through decoder layers.

3.2.2. Dense Blocks

Dense blocks are designed and incorporated into the generator to address the problem of partial character feature loss due to convolution during multilayer network propagation. This helps reduce broken strokes in the generated calligraphy characters. Dense blocks decrease the loss of character features by efficiently transmitting and reusing character feature information within the network. This improves the ability of the generator to capture character details and enhances the overall quality and coherence of the generated calligraphy characters [13].

Figure 3 shows the network architecture of dense blocks, which is designed based on the core idea of densely connected networks. To prevent the loss of character feature information during propagation, our model designs a dense block that combines two "Instance Normalization (IN)-Activation Function (ReLU)-Convolutional Layer (Conv)" operations. This combination helps to extract and transmit character features more effectively. There is a total of 6 dense blocks which can effectively extract and transfer character features.

In dense blocks, each block receives and integrates the output information of all previous blocks. This weighted and value transfer of information ensures the smooth circulation of character features between layers. Meanwhile, pooling between dense blocks reduces the information and contributes negatively. This is removed to further retain the feature maps. After six successive processing blocks, a convolutional layer is set up to return the dimensionality of the accumulated character feature information to the initial state at the time of inputting the blocks. The convolutional layer can effectively prevent dimensionality explosion [28]. The use of dense blocks significantly improves the ability of the generator to extract and retain character feature information, particularly in reducing the loss of stroke feature information during network propagation. This reduces the problem of broken strokes in generated calligraphy characters, making them more similar to authentic ones.

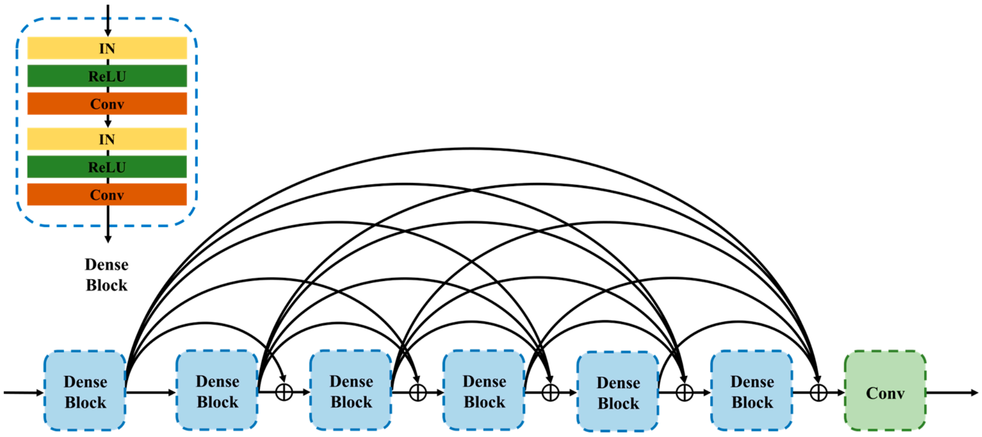


Figure 3. Network architecture of dense blocks.

3.2.3. Self-Attention Mechanism

Calligraphy characters are made up of a series of strokes, including both the main structural strokes and the smaller secondary strokes. However, when the generator extracts the secondary stroke feature, it assigns similar weights to these strokes as the main strokes through indiscriminate convolution operations. This results in visually redundant strokes and larger strokes than the authentic ones.

To address this issue, the generator incorporates self-attention mechanisms, as illustrated in Figure 2. These mechanisms enhance the importance of primary strokes and diminish the significance of secondary strokes based on their relevance.

The design of the generator includes self-attention mechanisms between the down-sampling layers and dense blocks, as well as between dense blocks and up-sampling layers. The initial self-attention mechanism adjusts stroke weights in printed characters, while the second self-attention mechanism adjusts stroke weights in generated calligraphy characters. In addition, the activation function Leaky ReLU is used in self-attention to solve the gradient direction zig-zagging dynamics in the weight gradient updates [29].

3.3. Discriminator

The proposed model includes two discriminators: a printed character discriminator and a calligraphy character discriminator. The printed character discriminator is responsible for judging the authenticity of printed characters, while the calligraphy character discriminator judges the authenticity of calligraphy characters. Both discriminators share the same network architecture. Using the calligraphy character discriminator as an example, the calligraphy character discriminator demonstrates how the discriminator authenticates the input calligraphy characters, as shown in Figure 4.

3.3.1. Discriminator Structure

Calligraphy characters consist of strokes that have significant directionality and relative positional relationships between them. However, the CNN-based discriminator cannot use these features. This results in the inability of the discriminator to recognize twisted and deformed strokes. To solve this problem, the proposed model designs the discriminator using a Capsule Network (CapsNet) and a fully convolutional network (FCN). CapsNet extracts the direction and relative position information of strokes from calligraphy characters, so the discriminator can judge the authenticity of calligraphy characters in terms of the direction and position of calligraphy strokes, thereby reducing the problem of stroke twisting and deformation in generated calligraphy characters [30].

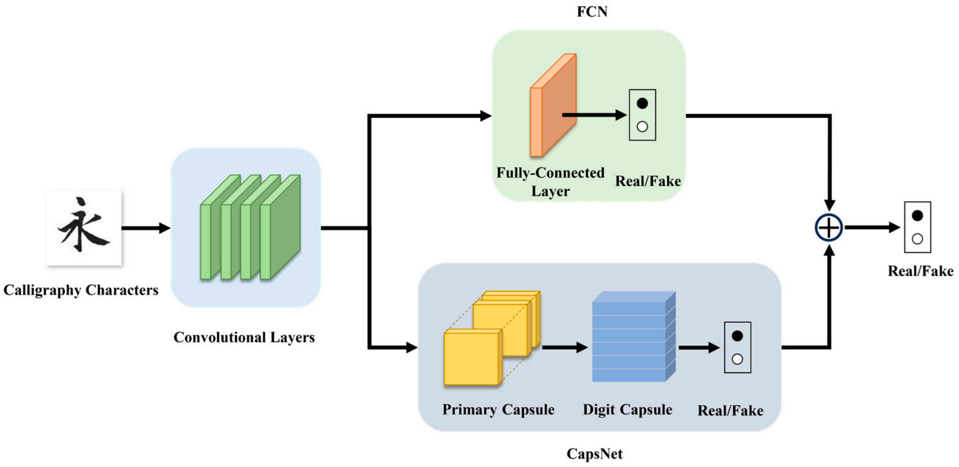


Figure 4. Network architecture of discriminator (The Chinese character is named Yong).

In the calligraphy character discriminator, the convolutional layer first performs feature extraction on the input calligraphy characters. These extracted calligraphy features are then fed into the CapsNet and FCN for authentication. FCN primarily uses the character stroke feature for authentication, while CapsNet focuses on using the direction and position information of character strokes for authentication. The CapsNet within the discriminator architecture consists of two core components: the primary capsule layer and the digit capsule layer. The role of the primary capsule layer is to perform convolution operations on the input character feature matrix, converting these features into vectorial capsules. These capsules not only encapsulate the stroke feature information, but also capture the directional and positional information of the strokes. This provides a rich context for subsequent judgments. The digit capsule layer receives the generated capsules after primary processing and transfers them into a matrix using dynamic routing. A norm operation is then performed on this matrix to obtain a digital scalar D_{caps} between 0 and 1, representing CapsNet's judgment of the input calligraphy character. In Figure 4, a fully connected network refers to a neural network in which each neuron applies a linear transformation to the input vector through a weight matrix. As a result, all possible connections layer-to-layer are present, meaning every input of the input vector influences every output of the output vector. In the calligraphy character discriminator, the FCN is used to evaluate the authenticity of the input calligraphy character based on the features of calligraphy strokes. To enable the discriminator to utilize both the stroke feature information and the direction and position information of the characters, the discriminator finally performs a weighted summation of the FCN's judgment result D_{fcn} and the CapsNet's judgment result D_{caps}, as shown in Equation (1):

$$D = \lambda_{\text{fcn}} \cdot D_{\text{fcn}} + \lambda_{\text{caps}} \cdot D_{\text{caps}} \tag{1}$$

where λ_{fcn} and λ_{caps} are the weights of D_{fcn} and D_{caps}, respectively. In the proposed model, both λ_{fcn} and λ_{caps} are set to 0.5 to balance the contributions of the two networks. By adopting this weighted fusion strategy, the discriminator can take into account stroke characteristics, and the direction and position information of the strokes. This significantly improves the accuracy of judging the authenticity of calligraphy characters and reduces the twisted and deformed strokes in the generated calligraphy characters.

3.3.2. CapsNet

CapsNet is a novel neural network architecture designed to overcome the limitations of traditional Convolutional Neural Networks (CNNs) [31,32]. It uses a series of neurons called "capsules" to detect specific features within an image. Each capsule in CapsNet outputs a vector containing information about the image features, such as position, orientation, and scale. Each capsule has a weight, which is continually updated during the training process. CapsNet can better understand the spatial relationships of elements in images by recognizing rotated and scaled elements. CapsNet is widely used in image recognition and object detection because of its ability to use directional information and the relative position information of elements [33,34].

Unlike scalar neurons, vector neurons in CapsNet encapsulate the information they need to carry and use weight matrices to store spatial information and other relationships between neurons. The vector neuron model is shown in Figure 5. First, the magnitude and the direction of the input vector are encapsulated into a prediction vector, i.e., x_i in Figure 5. At the same time, the relationship between the feature detected using the low-level capsule and the prediction of the feature via the high-level capsule is obtained. This relationship is coded to obtain the weight matrix w_i. The weight matrix w_i is used to process the prediction vector x_i to obtain a new input vector X_i. Second, the coupling coefficients C_i are set to multiply the input vectors X_i by them, respectively [31]. Again, after setting the coupling coefficients C_i, the vectors X_iC_i are summed to obtain the vector form of the capsule $\sum X_iC_i$. Finally, to ensure that the directionality of the capsules is preserved and the length of the capsules is reasonably constrained, the vector form of the capsule $\sum X_iC_i$ is converted to the vector V as the output of the capsule in this layer using the vectorized compression function Squash [35], as shown in Equation (2):

$$\text{Squash}(x) = \frac{\|x\|^2}{1+\|x\|^2}\frac{x}{\|x\|} \tag{2}$$

where x represents the vector form of the capsule $\sum X_iC_i$. The squash function preserves the directional information of the capsules while constraining their lengths to between 0 and 1, ensuring effective information transfer and processing stability.

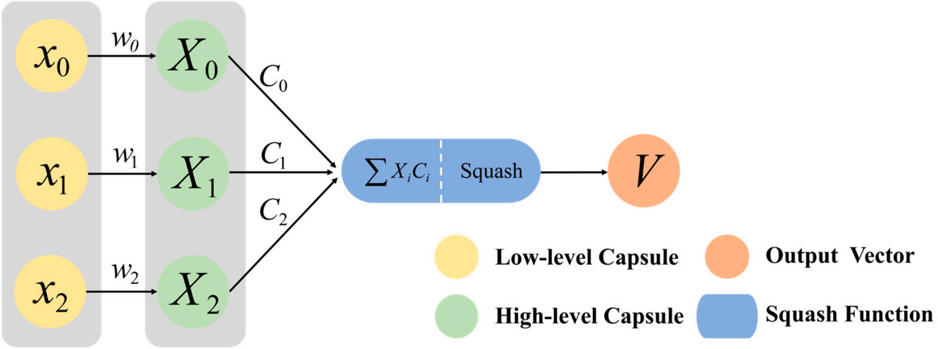


Figure 5. Vector neuron model.

3.4. Loss Function

To further improve the generation quality of the proposed model and to make the generated calligraphy characters more similar to authentic ones, a perceptual loss is introduced to construct the loss function [15]. The purpose of the perceptual loss is to force the model to pay more attention to the overall architecture and global features of the calligraphy characters, and to better capture and reproduce these key elements during the generation process. This ensures that the generated calligraphy characters match the authentic ones not only in stroke details but also in overall style and architecture.

The loss function $\mathcal{L}$ of the proposed model consists of three parts: adversarial loss $\mathcal{L}_{\text{GAN}}$, cycle consistency loss $\mathcal{L}_{cyc}$, and perceptual loss $\mathcal{L}_{\text{per}}$, to improve the similarity between the generated calligraphy and the authentic calligraphy, as shown in Equation (3):

$$\mathcal{L} = \mathcal{L}_{\text{GAN}} + \lambda \cdot \mathcal{L}_{cyc} + \gamma \cdot \mathcal{L}_{\text{per}} \tag{3}$$

where λ and γ are the weights of $\mathcal{L}_{cyc}$ and $\mathcal{L}_{\text{per}}$, respectively. After several experiments, the generation result of the proposed is optimal when λ is set to 10 and γ is set to 1/7.

Below, we introduce the individual loss functions that make up $\mathcal{L}$.

3.4.1. Adversarial Loss $\mathcal{L}_{\text{GAN}}$

The goal of $\mathcal{L}_{\text{GAN}}$ is to minimize the distance between the generated distribution of data and the actual ones. In the generation of calligraphy characters, $\mathcal{L}_{\text{GAN}}$ asks the discriminator to judge the generated characters as fake and the authentic characters as real. In the proposed model, $\mathcal{L}_{\text{GAN}}$ consists of the calligraphy character adversarial loss $\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y)$ and the printed character adversarial loss $\mathcal{L}_{\text{GAN}}(F, D_X, Y, X)$, as shown in Equation (4):

$$\mathcal{L}_{\text{GAN}} = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) \tag{4}$$

where G represents the calligraphy character generator, F represents the printed character generator, X represents the printed character domain, and Y represents the calligraphy character domain. The calligraphy character adversarial loss $\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y)$ and printed character adversarial loss $\mathcal{L}_{\text{GAN}}(F, D_X, Y, X)$ are, respectively, defined in Equations (5) and (6):

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = E_{y\sim p_{data(y)}}[\log D_Y(y)] + E_{x\sim p_{data(x)}}[\log(1 - D_Y(G(x)))] \tag{5}$$

$$\mathcal{L}_{\text{GAN}}(F, D_X, Y, X) = E_{x\sim p_{data(x)}}[\log D_X(x)] + E_{y\sim p_{data(y)}}[\log(1 - D_X(F(y)))] \tag{6}$$

The goal of $\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y)$ is to recognize the authentic calligraphy characters as real and the generated ones as fake. The goal of $\mathcal{L}_{\text{GAN}}(F, D_X, Y, X)$ is to recognize the real printed characters as real and the generated ones as fake.

3.4.2. Cycle Consistency Loss $\mathcal{L}_{cyc}$

$\mathcal{L}_{cyc}$ is a loss function used to train image translation models, whose main goal is to ensure that the output image of the model is similar to the input image after two translations. $\mathcal{L}_{cyc}$ is designed based on the principle of cycle consistency, which means that an image should be able to return to its original state after a series of translations [35]. The cycle consistency loss function measures the difference between the output after two translations and the original input, as shown in Equation (7):

$$\mathcal{L}_{cyc}(G, F) = E_{x\sim p_{data}(x)}[\|F(G(x)) - x\|_1] + E_{y\sim p_{data}(y)}[\|G(F(y)) - y\|_1] \tag{7}$$

where the printed character x is input into the calligraphy character generator G, generating the corresponding calligraphy character $G(x)$. Then, the generated calligraphy character $G(x)$ is input into the printed character generator F, aiming to keep the generated printed character $F(G(x))$ as consistent as possible with the original printed character x. The calligraphy character y is calculated in the same way as the printed character x.

3.4.3. Perceptual Loss $\mathcal{L}_{\text{per}}$

To improve the overall style similarity between the generated calligraphy characters and authentic ones, perceptual loss $\mathcal{L}_{\text{per}}$ is introduced to construct the loss function $\mathcal{L}$. $\mathcal{L}_{\text{per}}$ is used to calculate the mean square error between the feature of the real images and the feature of the generated images, both of which are extracted from the pre-trained networks [36]. With the help of $\mathcal{L}_{\text{per}}$, errors due to small differences in individual pixel positions can be avoided by comparing high-level features. The introduction of $\mathcal{L}_{\text{per}}$ encourages the generator to pay more attention to the reproduction of the overall style of calligraphy characters, thereby making the generated calligraphy characters closer to the authentic ones in overall style [37].

The perceptual loss $\mathcal{L}_{\text{per}}$ in the proposed model is composed of the perceptual loss of printed characters $\mathcal{L}^{x}_{\text{per}}$ and the perceptual loss of calligraphy characters $\mathcal{L}^{y}_{\text{per}}$, as shown in Equation (8):

$$\mathcal{L}_{\text{per}} = \mathcal{L}^{x}_{\text{per}}(x, F(G(x))) + \mathcal{L}^{y}_{\text{per}}(y, G(F(y))) \tag{8}$$

where x represents real printed characters, y represents authentic calligraphy characters, G represents the calligraphy character generator, and F represents the printed character generator. The perceptual loss of printed characters $\mathcal{L}^x_{\text{per}}$ is calculated using the generated printed characters $F(G(x))$ and the real ones x. Similarly, the perceptual loss of calligraphy characters $\mathcal{L}^y_{\text{per}}$ is calculated using the generated calligraphy characters $G(F(y))$ and the authentic ones y. $\mathcal{L}^x_{\text{per}}$ and $\mathcal{L}^y_{\text{per}}$ are summed to obtain the perceptual loss $\mathcal{L}_{\text{per}}$.

The calculation of $\mathcal{L}^x_{\text{per}}$ and $\mathcal{L}^y_{\text{per}}$ relies on a pre-trained network model that extracts the feature from the input images. The proposed model uses VGG16 as the pre-trained model. In the calculation of $\mathcal{L}^x_{\text{per}}$ and $\mathcal{L}^y_{\text{per}}$, the generated characters and real ones are input into the VGG16. The feature values are extracted and calculated to obtain loss values at layer 3, layer 8, layer 15, and layer 22 of the VGG16. $\mathcal{L}^x_{\text{per}}$ and $\mathcal{L}^y_{\text{per}}$ are obtained by summing the loss values from each layer [38]. The process of using the VGG16 model to calculate $\mathcal{L}^y_{\text{per}}$ is shown in Figure 6.

The method for calculating the perceptual loss of printed characters $\mathcal{L}^x_{\text{per}}$ and the perceptual loss of calligraphy characters $\mathcal{L}^y_{\text{per}}$ is shown in Equations (9) and (10):

$$\mathcal{L}^x_{\text{per}}(x, F(G(x))) = \frac{1}{C_j H_j W_j} \|\phi_j(x) - \phi_j(F(G(x)))\|^2_2 \tag{9}$$

$$\mathcal{L}^y_{\text{per}}(y, G(F(y))) = \frac{1}{C_j H_j W_j} \|\phi_j(y) - \phi_j(G(F(y)))\|^2_2 \tag{10}$$

where x represents real printed characters, y represents authentic calligraphy characters, G represents the calligraphy character generator, F represents the printed character generator, ϕ represents the pre-trained network VGG16, and j represents layer j of the pre-trained network. C_j, H_j and W_j, respectively, represent the number of channels, height, and width of the features.

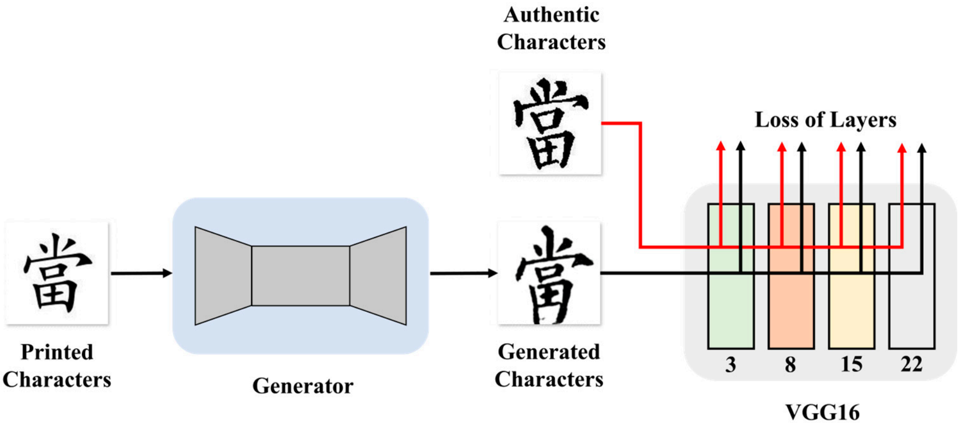


Figure 6. Perceptual loss of calligraphy characters $\mathcal{L}^y_{\text{per}}$ (The Chinese characters are named Dang).

3.5. Discussion of Proposed Method

An end-to-end generation model for Chinese calligraphy characters based on dense blocks and capsule networks is proposed. This model aims to solve problems such as redundant and broken strokes, twisted and deformed strokes, and dissimilarity to authentic strokes. The generator of the model uses self-attention mechanisms and densely connected blocks to reduce redundant and broken strokes. The discriminator consists of a capsule network and a fully connected network to reduce twisted and deformed strokes. In addition, the loss function includes a perceptual loss to increase the similarity between the generated calligraphy characters and the authentic ones.

4. Experiment

We experimentally evaluate the effectiveness of the proposed model using a self-constructed Chinese calligraphy character dataset. This dataset is used primarily for research on generating different styles of calligraphy characters based on deep learning, with the aim of promoting the integration of traditional Chinese calligraphy with modern technology. To comprehensively evaluate the performance of our model, three different style generation experiments are designed: Yan Zhenqing's regular script, Deng Shiru's clerical script, and Wang Xizhi's running script. In evaluating the experimental results, three quantitative evaluation metrics are chosen: the Structural Similarity Index [39], Mean Square Error [40], and the Peak Signal-to-Noise Ratio [41]. These evaluation metrics are intended to comprehensively measure the performance of the model in generating different styles of calligraphy characters from different perspectives.

4.1. Dataset

The Chinese Calligraphy Character Dataset is derived from ancient texts and stele inscriptions, ensuring the authenticity and authority of the data. The dataset includes three categories: regular script, clerical script, and running script. Through the pre-processing and cropping the images, samples of calligraphy characters with a pixel size of 128 × 128 are constructed for model training and evaluation.

First, the gray scale of the original image is inverted so that the white characters on a black background become black characters on a white background. Second, the inverted gray-scale image is pre-processed, which involves techniques such as denoising to prevent noise from affecting the quality of the sample. Then, to ensure the uniform format of the samples, the pre-processed black-on-white image is directly cropped to obtain the calligraphy character. Next, to ensure the clarity of the sample images and to facilitate the subsequent use of the dataset, the Skimage method was chosen to normalize the calligraphy character slices to obtain a single calligraphy character image with a pixel size of 128 × 128. Finally, all the calligraphy character images were binarized. After the above steps, a sample of calligraphy characters was obtained with a uniform format and size.

In the regular script category, works by four master calligraphers are selected, including Ouyang Xun's "Inscription on the Sweet Spring in the Jiucheng Palace", Yan Zhenqing's "Duobao Pagoda Stele", Liu Gongquan's "Xuanmi Pagoda Stele", and Zhao Mengfu's "Danba Stele". These works are hailed as classics by the "Four Masters of Regular Script" in China and hold immense artistic and historical value [42]. In the clerical script category, Deng Shiru's "Shaoxue Qinshu Clerical Script Album" is selected. Deng Shiru's clerical script is known for its tight structure, simplicity, and elegance, demonstrating a high level of artistry and serving as a typical representation of clerical script [43]. In the running script category, Wang Xizhi's "Orchid Pavilion Preface" is selected. With its fluid and natural style, it is hailed as the most important running script under the sky [44]. Detailed information on the Chinese calligraphy character dataset is given in Table 1. This dataset not only provides rich data resources for generating calligraphy characters, but also helps to study the characteristics and artistic values of different calligraphy styles in depth.

Table 1. Dataset of Chinese calligraphy characters.

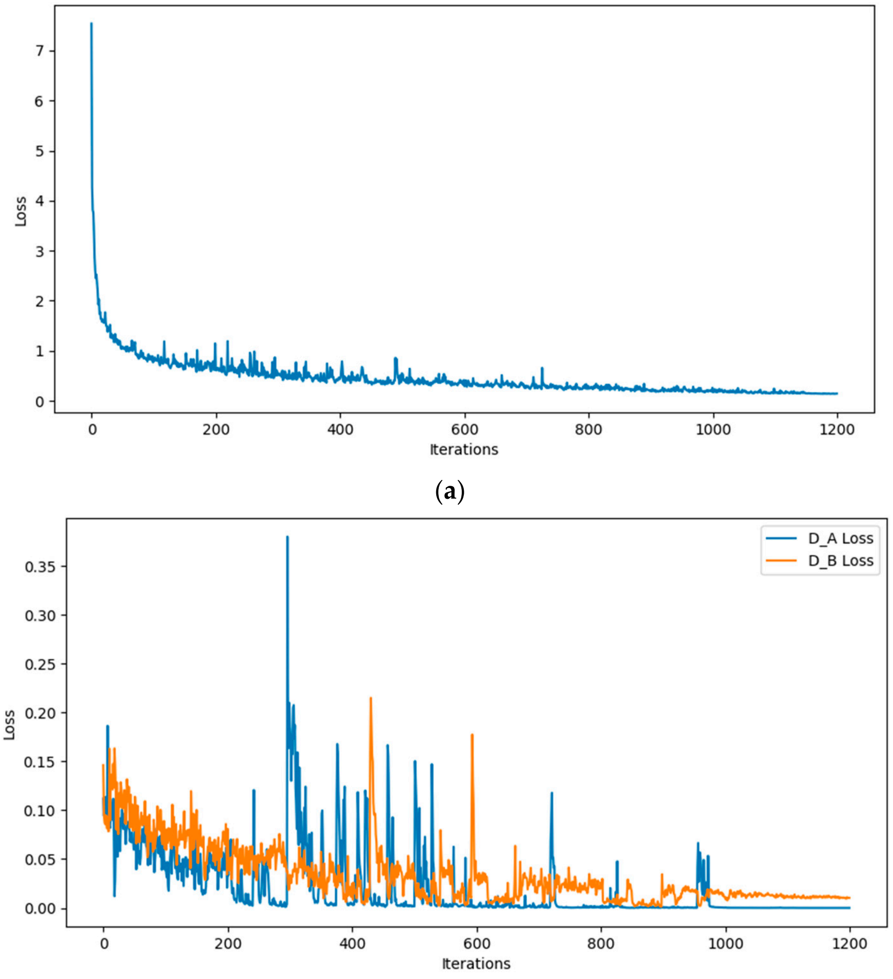
Sample Category		**Quantity**
Regular Script	Ouyang Xun's "Inscription on the Sweet Spring in the Jiucheng Palace"	1107
	Yan Zhenqing's "Duobao Pagoda Stele"	479
	Liu Gongquan's "Xuanmi Pagoda Stele"	1315
	Zhao Mengfu's "Danba Stele"	902
Clerical Script	Deng Shiru's "Shaoxue Qinshu Clerical Script Album"	245
Running Script	Wang Xizhi's "Orchid Pavilion Preface"	324
	Total	4372

4.2. Training Process

The specific configuration of the computer system used in the experiment is as follows: the operating system is Ubuntu and the processor is an Intel Xeon Silver 4216 quad-core CPU with 8 GB of RAM. The graphics card is an NVIDIA RTX3080 with 8 GB of video memory, of which 7 GB is used during model training. The programming language used is Python 3.8.15 and the implementation is based on the Pytorch 1.12.1 framework.

The following hyperparameters are set during model training: the number of iterations is 200, the batch size is 8, the learning rate is 0.0002, and the number of decay iterations is 100. These hyperparameters are carefully selected and adjusted to ensure the stability and effectiveness of the model training.

The proposed model analyzes the training process in detail by recording changes in the loss values of the model. Taking the training process for generating Yan Zhenqing's regular script as an example, the variation in the loss value of the generator is shown in Figure 7a, and the variation in the loss value of the discriminator is shown in Figure 7b. Here, D_A Loss and D_B Loss are the loss values for the calligraphy character discriminator and the printed one, respectively. The loss values for both the generator and the discriminator progressively decrease, while the fluctuations decrease and eventually stabilize. This trend in the loss function demonstrates the effectiveness and stability of the proposed model during the training process.



(**a**)

(**b**)

Figure 7. Plot of loss function (Yan Zhenqing's regular script). (**a**) Loss function of generator. (**b**) Loss function of discriminator.

4.3. Comparative Experiments

In this paper, the proposed methods are compared qualitatively and quantitatively with four other methods. The following models are chosen: AGGAN [45], StarGAN-v2 [46], pix2pix [20], and CycleGAN. AGGAN is an improved GAN model that enhances the quality and diversity of the generated images by introducing an auxiliary classifier. StarGAN-v2 is an advanced image translation model specifically designed for multi-domain style translation. Pix2pix is an image translation model based on conditional GAN. CycleGAN is an image translation model for image translation between different domains. The proposed model can be comprehensively evaluated through comparative experiments with these four models by generating characters of three style calligraphy styles: Yan Zhenqing's regular script, Deng Shiru's clerical script, and Wang Xizhi's running script.

4.3.1. Qualitative Comparison

The comparative experimental results for the regular script, clerical script, and running script are shown, respectively, in Figure 8, Figure 9, and Figure 10.

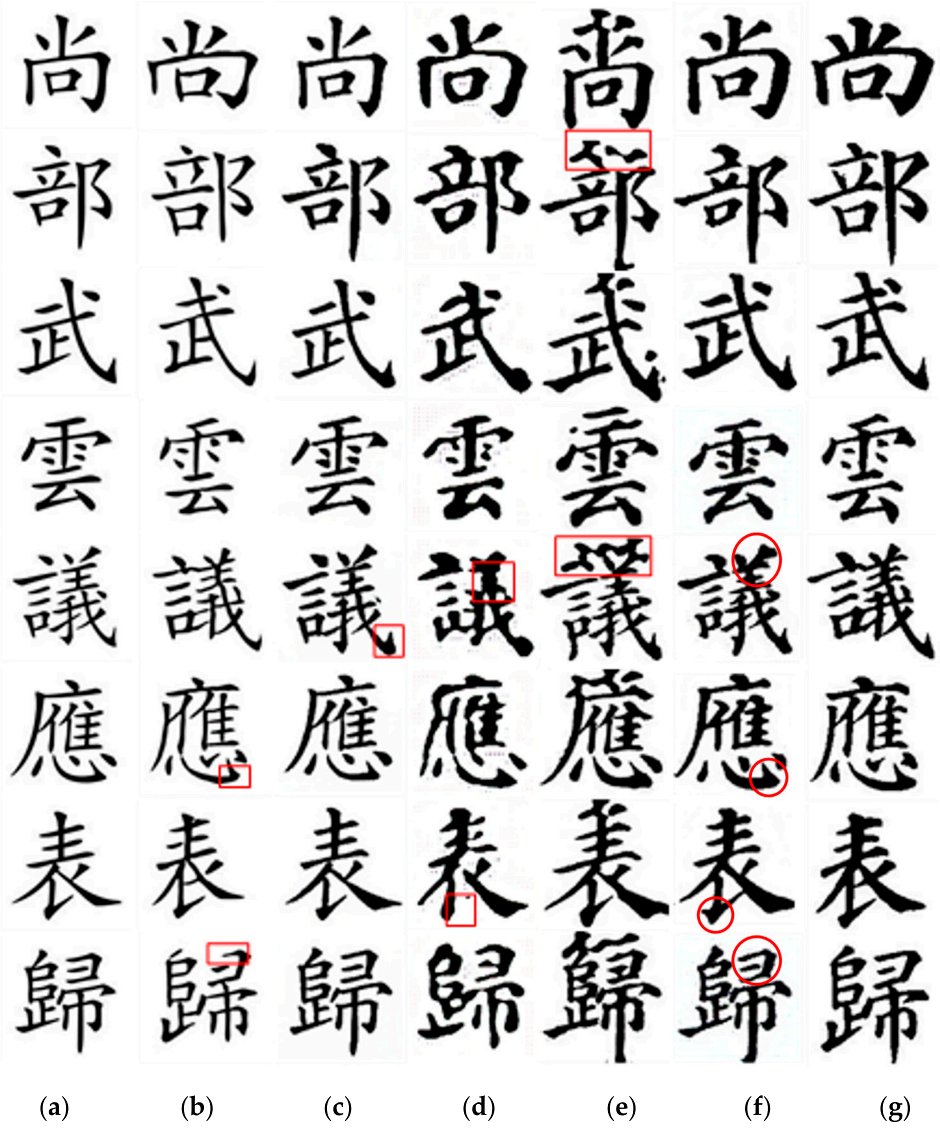


Figure 8. Comparison on regular script generation (The Chinese characters from upper to lower lines are named Shang, Bu, Wu, Yun, Yi, Ying, Biao, and Gui, respectively. The red squares and circles are used for comparison of details). (**a**) Printed characters. (**b**) AGGAN. (**c**) StarGAN-v2. (**d**) pix2pix. (**e**) CycleGAN. (**f**) Ours. (**g**) Authentic ones.

(1) Yan Zhenqing's regular script

The generation results for Yan Zhenqing's regular script are shown in Figure 8. In the comparative experiment, the calligraphy characters generated using AGGAN suffer from problems such as too thin strokes, missing strokes, and incomplete character structures. The calligraphy characters generated using StarGAN-v2 have a low similarity in style to the authentic ones, with significant deviations in detail and style. Calligraphy characters generated using pix2pix have problems with missing and distorted strokes. Calligraphy characters generated using CycleGAN have redundant strokes. On the other hand, the calligraphy characters generated using the proposed model, with clear strokes and complete structures, are more similar to the authentic ones in style and detail, including the rigorous structure and varied details of the strokes.

(2) Deng Shiru's clerical script

Clerical script, as a unique style of Chinese characters, is characterized by a slightly flattened writing effect and a structure in which horizontal strokes are long and vertical strokes are short. It is challenging to generate clerical script due to the differences between clerical script and modern character forms [43].

The generation results for clerical calligraphy are shown in Figure 9. In the comparative experiment, the calligraphy characters generated using AGGAN suffer from noticeably thin strokes, which results in a lack of the solidity characteristic of clerical script. The calligraphy characters generated using StarGAN-v2 have a low stylistic similarity to the authentic ones. The calligraphy characters generated using pix2pix have problems with missing and distorted strokes. CycleGAN generates calligraphy characters with redundant strokes. In contrast, the calligraphy characters generated using the proposed model have clear strokes and complete structures. It has a slightly flattened structure and long horizontal strokes with short vertical ones, and is more similar in style and detail to the authentic characters.

(3) Wang Xizhi's running script

Wang Xizhi's "Orchid Pavilion Preface" is celebrated as the most important running script in ancient China, and its unique personal style adds artistic charm to each calligraphy character. It is challenging to generate these calligraphy characters [47].

The results of running script generation are shown in Figure 10. In the comparison experiment, the calligraphy characters generated using AGGAN have problems with missing strokes. The calligraphy characters generated using StarGAN-v2 have less style similarity to the authentic ones, and have some blurred strokes. The calligraphy characters generated using pix2pix have problems with missing and distorted strokes. CycleGAN generated calligraphy characters with missing strokes. In comparison, the proposed model shows the highest generation quality when generating the running script. The calligraphy characters generated using the proposed model are more similar to the authentic calligraphy characters in style and detail, including their fluid strokes, natural rhythm, and unique personal style.

Table 2 shows the results of the qualitative analyses of the proposed and comparative models. The proposed model has shown significant advantages in the generation of regular script, clerical script, and running script. First, by introducing dense blocks into the generator, the model improves its ability to extract features from calligraphy strokes, effectively reducing problems with broken strokes. The dense blocks are able to capture richer detail information, ensuring that the generated calligraphy characters are structurally more complete. In addition, the proposed model introduces self-attention mechanisms into the generator, which further enhances the model's perception of calligraphy strokes. The self-attention mechanism allows the model to focus on key strokes, thereby reducing the generation of redundant strokes. Second, the proposed model employs the CapsNet in the discriminator. This allows the model to effectively extract the positional information of calligraphy strokes, thereby reducing the problem of stroke distortion. The CapsNet has superiority in handling spatial information, which leads to a better understanding of the structure of calligraphy characters. Finally, the proposed model introduces a perceptual loss function. This strategy aims to improve the calligraphy style recognition ability of the model, making the generated calligraphy characters more similar to the authentic ones in style. Through the perceptual loss, the proposed model can better capture the unique charm and artistic characteristics of calligraphy characters.

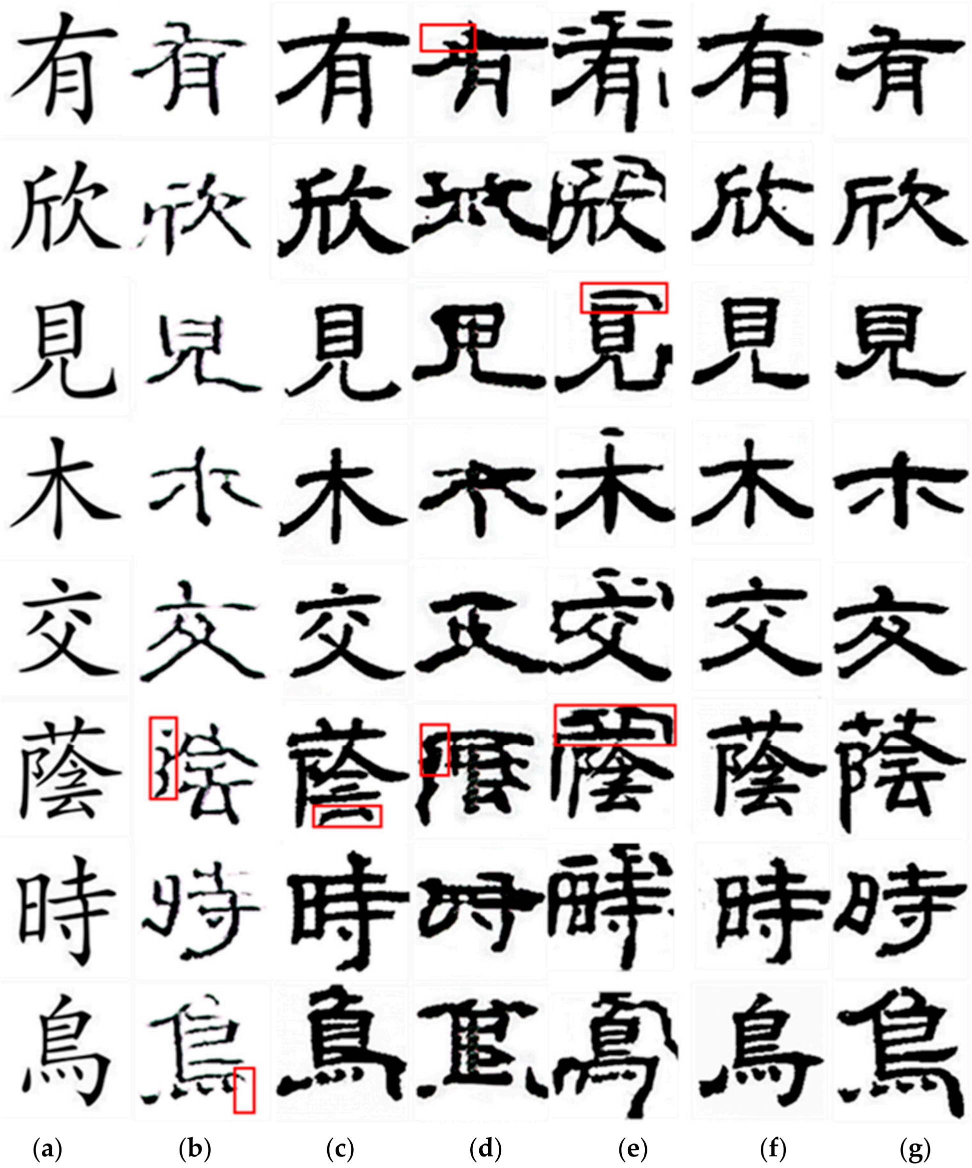


Figure 9. Comparison on clerical script generation (The Chinese characters from upper to lower lines are named You, Xin, Jian, Mu, Jiao, Yin, Shi, and Niao, respectively. The red squares are used for comparison of details). (**a**) Printed characters. (**b**) AGGAN. (**c**) StarGAN-v2. (**d**) pix2pix. (**e**) CycleGAN. (**f**) Ours. (**g**) Authentic ones.

Table 2. Comparison of qualitative evaluation.

Methods	Strokes	Structure	Style
pix2pix	broken	incomplete	dissimilar
AGGAN	broken	incomplete	dissimilar
CycleGAN	distortion	deformation	dissimilar
StarGAN-v2	redundant	deformation	dissimilar
Ours	clear	complete	similar

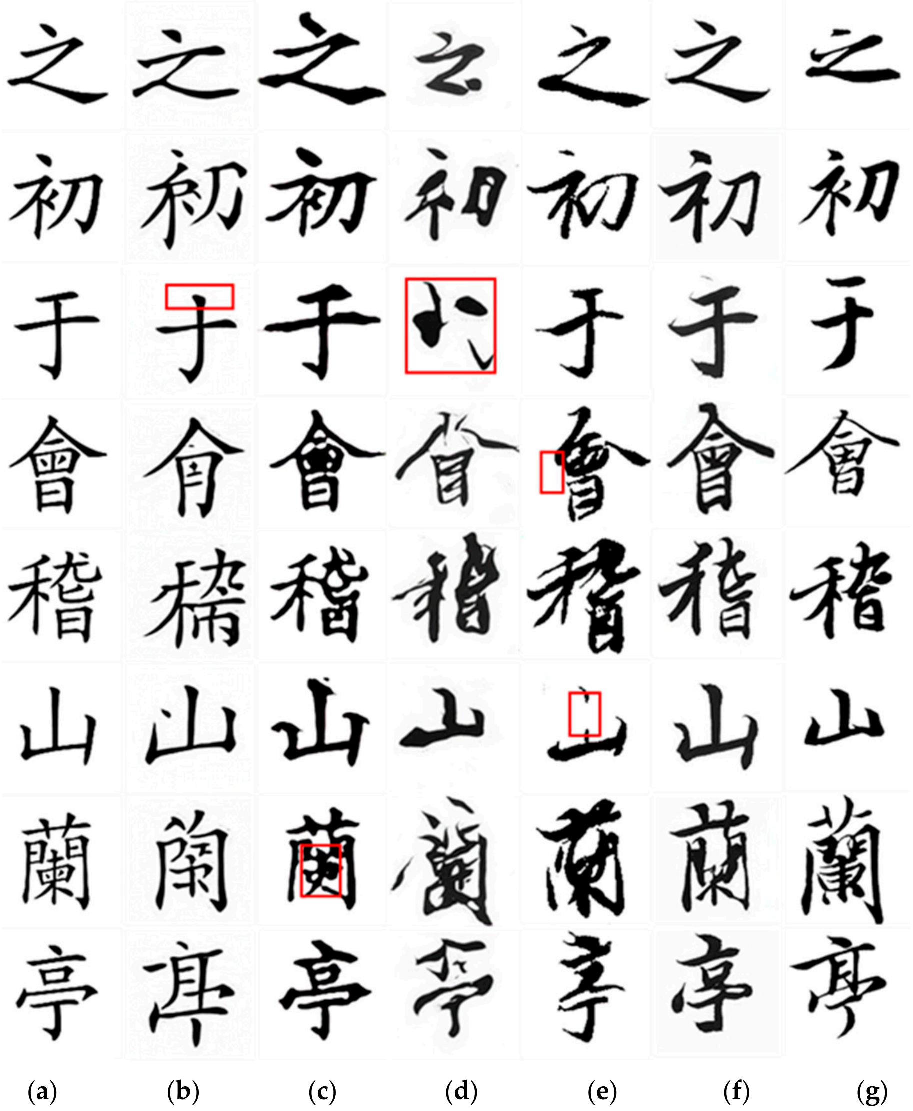


Figure 10. Comparison on running script generation (The Chinese characters from upper to lower lines are named Zhi, Chu, Yu, Kuai, Ji, Shan, Lan, and Ting, respectively. The red squares are used for comparison of details). (**a**) Printed characters. (**b**) AGGAN. (**c**) StarGAN-v2. (**d**) pix2pix. (**e**) CycleGAN. (**f**) Ours. (**g**) Authentic ones.

In summary, the proposed model significantly improves the quality of generated calligraphy characters through modular design and optimization strategies. It not only effectively solves problems such as stroke discontinuity, redundant strokes, and distortion, but also more accurately captures the style of authentic calligraphy characters, thus imbuing the generated calligraphy characters with the essence of the authentic ones. The proposed model is trained using the dataset of authentic calligraphy characters, while the dataset of authentic characters has a limited amount of data, which leads to the fact that the calligraphy characters generated using the proposed model are still a little different from the corresponding authentic characters.

4.3.2. Quantitative Comparison

From a quantitative perspective, we analyze the generation results of three types of calligraphy fonts: regular script, clerical script, and running script. To objectively evaluate the quality of the generated results, we use three quantitative evaluation metrics: Structural Similarity Index (SSIM), Mean Square Error (MSE), and Peak Signal-to-Noise Ratio (PSNR).

SSIM is a widely used metric to measure image quality [33]. A higher SSIM indicates that the generated calligraphy characters are structurally closer to the authentic ones. MSE is a metric used to measure the similarity of image pixels [34]. By achieving a lower MSE, the quality of the generated calligraphy characters can be significantly improved, making them much closer to the authentic ones at the pixel level. PSNR is an important measure of image quality [35]. A higher PSNR indicates that the generated calligraphy characters are visually closer to the authentic ones.

Table 3, Table 4, and Table 5, which correspond to Figure 8, Figure 9, and Figure 10, respectively, show the quantitative metrics of various models on regular script, clerical script, and running script. Among them, pix2pix generates calligraphy characters with problems such as missing strokes and distorted strokes, which differ significantly from the authentic ones; thus, its three metrics are the worst of all models. The calligraphy characters generated using AGGAN are different from the authentic ones, with thin strokes and some missing strokes. But it is better than the results generated using pix2pix, so the metrics of AGGAN are better than the metrics of pix2pix, but worse than the metrics of other models. The calligraphy characters generated using CycleGAN have no missing strokes, but have problems such as redundant strokes. So, the metrics of CycleGAN are better than the metrics of pix2pix and the metrics of AGGAN, but worse than the metrics of the proposed model. StarGAN-v2 generates calligraphy characters without missing or redundant strokes, but the overall calligraphy style differs significantly from the authentic ones. Thus, its three metrics are better than those of pix2pix, AGGAN, and CycleGAN, but worse than those of the proposed model.

Targeting the characteristics of calligraphy characters, self-attention, dense blocks, CapsNet, and perceptual loss $\mathcal{L}_{\text{per}}$ are used to design the proposed model. Therefore, the calligraphy characters generated using the proposed model are superior to pix2pix, AGGAN, CycleGAN, and StarGAN-v2 in terms of SSIM, MSE, and PSNR metrics.

Table 3. Comparison of quantitative evaluation on regular script.

Methods	**SSIM (↑)**	**MSE (↓)**	**PSNR (↑)**
pix2pix	0.622	29.761	10.240
AGGAN	0.630	29.459	10.481
CycleGAN	0.635	29.216	10.483
StarGAN-v2	0.742	29.120	10.629
Ours	0.758	28.680	10.853

Table 4. Comparison of quantitative evaluation on clerical script.

Methods	**SSIM (↑)**	**MSE (↓)**	**PSNR (↑)**
pix2pix	0.549	31.823	9.259
AGGAN	0.558	31.687	9.684
CycleGAN	0.563	30.837	9.741
StarGAN-v2	0.571	30.587	9.839
Ours	0.613	28.484	10.753

Table 5. Comparison of quantitative evaluation on running script.

Methods	**SSIM (↑)**	**MSE (↓)**	**PSNR (↑)**
AGGAN	0.527	30.497	9.429
StarGAN-v2	0.537	29.791	9.562
pix2pix	0.511	30.685	9.203
CycleGAN	0.532	29.910	9.482
Ours	0.594	27.843	10.655

Finally, we discuss the computational complexity of the proposed model. The size of the weight parameters of the generator of the proposed model is 160 MB and the size of the weight parameters of the discriminator is 20 MB. The single training time of the proposed model is about 12 h and the single testing time is about 10 s.

4.4. Ablation Study

To evaluate the effect of self-attention, dense blocks, CapsNet, and perceptual loss $\mathcal{L}_{\text{per}}$ on the image translation results, we designed ablation experiments on the generation of a running script, as shown in Figure 11. The ablation experiment includes four sets of comparison experiments, corresponding to models without different parts. The proposed model without $\mathcal{L}_{\text{per}}$ is denoted as "Proposed model—$\mathcal{L}_{\text{per}}$", the proposed model without CapsNet is denoted as "Proposed model—CapsNet", the proposed model without dense blocks is denoted as "Proposed model—Dense blocks", and the proposed model without self-attention is denoted as "Proposed Model—self-attention".

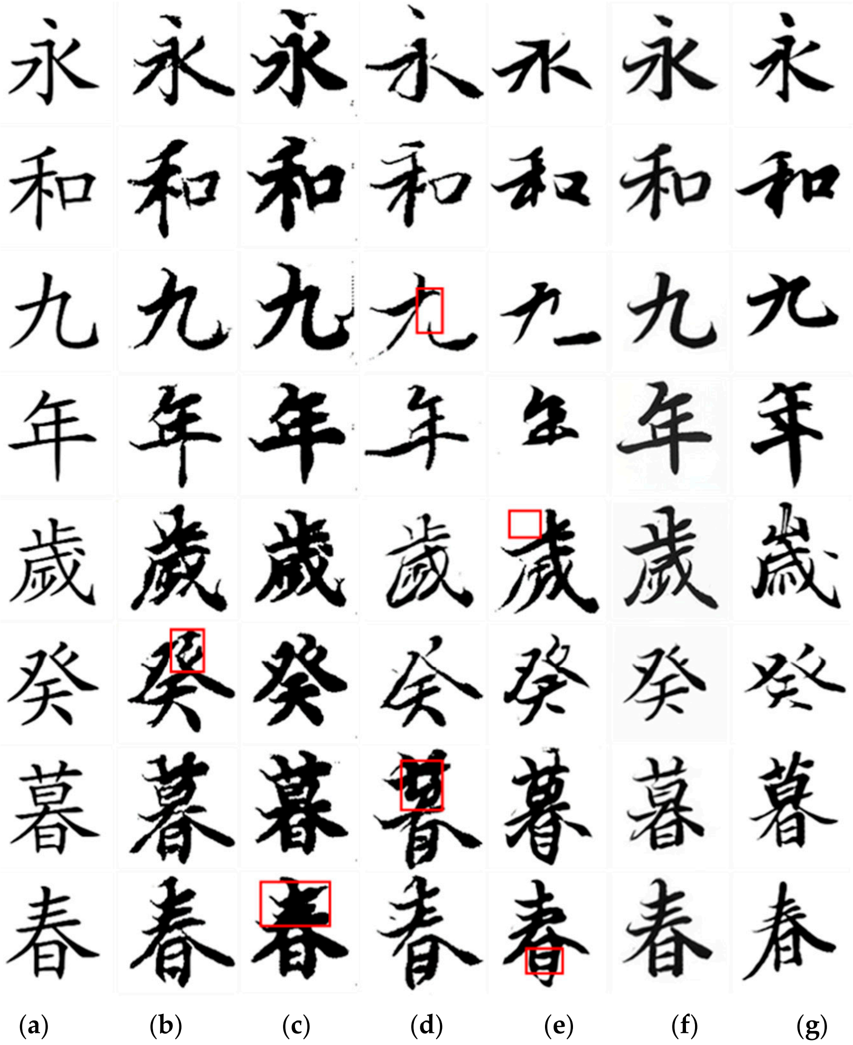


Figure 11. Ablation experiment on running script generation (The Chinese characters from upper to lower lines are named Yong, He, Jiu, Nian, Sui, Gui, Mu, and Chun, respectively. The red squares are used for comparison of details). (**a**) Printed characters. (**b**) Proposed model—$\mathcal{L}_{\text{per}}$. (**c**) Proposed model—CapsNet. (**d**) Proposed model—dense blocks. (**e**) Proposed model—self-attention. (**f**) Proposed model. (**g**) Authentic ones.

In Figure 11, there is a degree of degradation in the generation effect for each ablation model. Without $\mathcal{L}_{\text{per}}$, the generated running script has problems such as stroke distortion due to the lack of the in-depth learning of calligraphy styles. Without CapsNet, the model fails to accurately extract the positional information of strokes, resulting in problems such as strokes being too thick and sticking together in the generated running script. Without dense blocks, the model has limited ability to extract the features of strokes in the running script. The generated strokes are thin, and some strokes are missing. Without self-attention, the model's ability to perceive the running script decreases and the generated calligraphy characters have problems with missing and distorted strokes. The running script characters generated using the proposed model with self-attention, dense blocks, CapsNet, and $\mathcal{L}_{\text{per}}$ are more similar to the authentic ones. In addition, the quantitative metrics obtained from the ablation experiments in Figure 11 are compared in Table 6. The metrics of the ablation models are lower compared to the proposed model, further proving the effectiveness of each structural component.

Table 6. Comparison of quantitative evaluation for the ablation experiments on running script.

Methods	SSIM (↑)	MSE (↓)	PSNR (↑)
Ours	0.594	27.843	10.655
Ours—$\mathcal{L}_{\text{per}}$	0.532	31.292	9.488
Ours—CapsNet	0.526	31.717	9.711
Our—dense blocks	0.518	32.727	9.578
Ours—self-attention	0.510	31.012	9.763

4.5. Discussion

The proposed model aims to solve problems such as redundant and broken strokes, twisted and deformed strokes, and dissimilarity to authentic strokes. The model's generator uses self-attention mechanisms and densely connected blocks to reduce redundant and broken strokes. The discriminator consists of a capsule network and a fully connected network to reduce twisted and deformed strokes. In addition, the loss function includes a perceptual loss to increase the similarity between the generated calligraphy characters and the authentic ones. To demonstrate the validity of the proposed model, we conducted comparison and ablation experiments on the datasets of Yan Zhenqing's regular script, Deng Shiru's clerical script, and Wang Xizhi's running script. The experimental results show that, compared with the comparison model, the proposed model improves SSIM by 0.07 on average, reduces MSE by 1.95 on average, and improves PSNR by 0.92 on average, which proves the effectiveness of the proposed model.

5. Conclusions

In this paper, an end-to-end generation model for Chinese calligraphy characters based on dense blocks and a capsule network is proposed. Experiments were conducted on Yan Zhenqing's regular script, Deng Shiru's clerical script, and Wang Xizhi's running script. The experimental results indicate that the proposed model can not only generate calligraphy characters in different styles but can also significantly reduce problems such as redundant and broken, twisted, and deformed strokes. Compared with other current models, the proposed method achieves superior generation effects. By studying the generation models of calligraphy characters and improving the quality of generated calligraphy, we can better explore the intrinsic rules and artistic characteristics of calligraphy, thus providing technical support for the inheritance and innovation of calligraphy art.

Due to the limited amount of the authentic calligraphy characters in our dataset, the training effect on the proposed model is restricted. As a result, the generated calligraphy characters still exhibit some differences from the authentic ones, particularly in terms of details. In future research, expanding the training samples and optimizing the network model should be considered to make the generated calligraphy characters more similar to the authentic ones.

Author Contributions: Conceptualization, Z.S.; data curation, W.Z.; funding acquisition, Z.S.; investigation, W.Z.; methodology, Z.S.; supervision, Z.S.; validation, W.Z. and X.W.; writing—original draft, W.Z.; writing—review and editing, W.Z., Z.S. and X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Research and Development Program of China (No.2017YFB1402102), the National Natural Science Foundation of China (No. 62377033), the Shaanxi Key Science and Technology Innovation Team Project (No. 2022TD-26), the Xi'an Science and Technology Plan Project (No. 23ZDCYJSGG0010-2022), and the Fundamental Research Funds for the Central Universities (No. GK202205036, GK202101004).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yuan, S.; Dai, A.; Yan, Z.; Liu, R.; Chen, M.; Chen, B.; Qiu, Z.; He, X. Learning to Generate Poetic Chinese Landscape Painting with Calligraphy. *arXiv* **2023**, arXiv:2305.04719.
2. Wu, R.; Chao, F.; Zhou, C.; Chang, X.; Yang, L.; Shang, C.; Zhang, Z.; Shen, Q. Internal model control structure inspired robotic calligraphy system. *IEEE Trans. Ind. Inform.* **2023**, *20*, 2600–2610. [CrossRef]
3. Wu, S.J.; Yang, C.Y.; Hsu, J.Y. Calligan: Style and structure-aware chinese calligraphy character generator. *arXiv* **2020**, arXiv:2005.12500.
4. Zhou, P.; Zhao, Z.; Zhang, K.; Li, C.; Wang, C. An end-to-end model for chinese calligraphy generation. *Multimed. Tools Appl.* **2021**, *80*, 6737–6754. [CrossRef]
5. Chai, X.; Wang, Y.; Chen, X.; Gan, Z.; Zhang, Y. TPE-GAN: Thumbnail preserving encryption based on GAN with key. *IEEE Signal Process. Lett.* **2022**, *29*, 972–976. [CrossRef]
6. Jiang, F.; Ma, J.; Webster, C.J.; Li, X.; Gan, V.J. Building layout generation using site-embedded GAN model. *Autom. Constr.* **2023**, *151*, 104888. [CrossRef]
7. Zhang, Z.; Zhou, X.; Qin, M.; Chen, X. Chinese Character Style Transfer Based on Multi-scale GAN. *Signal Image Video Process.* **2022**, *16*, 559–567. [CrossRef]
8. Kong, Y.; Luo, C.; Ma, W.; Zhu, Q.; Zhu, S.; Yuan, N.; Jin, L. Look Closer to Supervise Better: One-shot Font Generation via Component-based Discriminator. In Proceedings of the International Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13482–13491.
9. Li, Y.; Duan, J.; Su, X.; Zhang, L.; Yu, H.; Liu, X. A calligraphy character generation algorithm based on improved adversarial network. *J. Zhejiang Univ.* **2023**, *57*, 1326–1334.
10. Wang, X.; Hui, L.; Li, C.; Sun, Z.; Xiao, Y. A Study of Calligraphy Font Generation Based on DANet-GAN. In Proceedings of the Chinese Control Conference, Tianjin, China, 24–26 July 2023; pp. 8473–8478.
11. Zhou, H.; Liu, Q.; Weng, D.; Wang, Y. Unsupervised cycle-consistent generative adversarial networks for pan sharpening. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. [CrossRef]
12. Li, K.; Wang, Y.; Zhang, J.; Gao, P.; Song, G.; Liu, Y.; Li, H.; Qiao, Y. Uniformer: Unifying convolution and self-attention for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 12581–12600. [CrossRef]
13. Girdhar, N.; Sinha, A.; Gupta, S. DenseNet-II: An improved deep convolutional neural network for melanoma cancer detection. *Soft Comput.* **2023**, *27*, 13285–13304. [CrossRef] [PubMed]
14. Chen, Z.; Zhang, L.; Sun, J.; Meng, R.; Yin, S.; Zhao, Q. DCAMCP: A deep learning model based on capsule network and attention mechanism for molecular carcinogenicity prediction. *J. Cell. Mol. Med.* **2023**, *27*, 3117–3126. [CrossRef] [PubMed]
15. Song, J.; Yi, H.; Xu, W.; Li, B.; Li, X. Gram-GAN: Image Super-Resolution Based on Gram Matrix and Discriminator Perceptual Loss. *Sensors* **2023**, *23*, 2098. [CrossRef] [PubMed]
16. Pang, Y.; Lin, J.; Qin, T.; Chen, Z. Image-to-image translation: Methods and applications. *IEEE Trans. Multimed.* **2021**, *24*, 3859–3881. [CrossRef]
17. Torbunov, D.; Huang, Y.; Yu, H.; Huang, J.; Yoo, S.; Lin, M.; Viren, B.; Ren, Y. Uvcgan: Unet vision transformer cycle-consistent gan for unpaired image-to-image translation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 2–7 January 2023; pp. 702–712.
18. Pinheiro Cinelli, L.; Araújo Marins, M.; Barros da Silva, E.A.; Netto, S.L. *Variational Autoencoder//Variational Methods for Machine Learning with Applications to Deep Networks*; Springer International Publishing: Cham, Switzerland, 2021; pp. 111–149.
19. Siddique, N.; Paheding, S.; Elkin, C.P.; Devabhaktuni, V. U-net and its variants for medical image segmentation: A review of theory and applications. *IEEE Access* **2021**, *9*, 82031–82057. [CrossRef]
20. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July; IEEE: Piscataway, NJ, USA, 2017; pp. 1125–1134.

21. Tumanyan, N.; Geyer, M.; Bagon, S.; Dekel, T. Plug-and-play diffusion features for text-driven image-to-image translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 1921–1930.
22. Parmar, G.; Kumar Singh, K.; Zhang, R.; Li, Y.; Lu, J.; Zhu, J. Zero-shot image-to-image translation. In Proceedings of the ACM SIGGRAPH 2023 Conference Proceedings, Los Angeles, CA, USA, 6–10 August 2023; pp. 1–11.
23. Ko, K.; Yeom, T.; Lee, M. Superstargan: Generative adversarial networks for image-to-image translation in large-scale domains. *Neural Netw.* **2023**, *162*, 330–339. [CrossRef]
24. Huang, Y.; He, M.; Jin, L.; Wang, Y. Rd-gan: Few/zero-shot chinese character style transfer via radical decomposition and rendering. In Proceedings of the Computer Vision—ECCV, Glasgow, UK, 23–28 August 2020; Volume 12351, pp. 156–172.
25. Gao, Y.; Wu, J. Gan-based unpaired chinese character image translation via skeleton transformation and stroke rendering. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 646–653. [CrossRef]
26. Xiao, Y.; Lei, W.; Lu, L.; Chang, X.; Zheng, X.; Chen, X. CS-GAN: Cross-structure generative adversarial networks for Chinese calligraphy translation. *Knowl.-Based Syst.* **2021**, *229*, 107334. [CrossRef]
27. Wen, Q.; Li, S.; Han, B.; Yuan, Y. Zigan: Fine-grained chinese calligraphy font generation via a few-shot style transfer approach. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual, 20–24 October 2021; pp. 621–629.
28. Wei, M.; Wu, Q.; Ji, H.; Wang, J.; Lyu, T.; Liu, J.; Zhao, L. A Skin Disease Classification Model Based on DenseNet and ConvNeXt Fusion. *Electronics* **2023**, *12*, 438. [CrossRef]
29. Zhou, M.; Liu, X.; Yi, T.; Bai, Z.; Zhang, P. A superior image inpainting scheme using Transformer-based self-supervised attention GAN model. *Expert Syst. Appl.* **2023**, *233*, 120906. [CrossRef]
30. Shao, G.; Huang, M.; Gao, F.; Liu, T.; Li, L. DuCaGAN: Unified dual capsule generative adversarial network for unsupervised image-to-image translation. *IEEE Access* **2020**, *8*, 154691–154707. [CrossRef]
31. Wei, Y.; Liu, Y.; Li, C.; Cheng, J.; Song, R.; Chen, X. TC-Net: A Transformer Capsule Network for EEG-based emotion recognition. *Comput. Biol. Med.* **2023**, *152*, 106463. [CrossRef] [PubMed]
32. Lei, Y.; Wu, Z.; Li, Z.; Yang, Y.; Liang, Z. BP-CapsNet: An image-based Deep Learning method for medical diagnosis. *Appl. Soft Comput.* **2023**, *146*, 110683. [CrossRef]
33. Liu, X.; Li, X.; Fiumara, G.; De Meo, P. Link prediction approach combined graph neural network with capsule network. *Expert Syst. Appl.* **2023**, *212*, 118737. [CrossRef]
34. Long, J.; Qin, Y.; Yang, Z.; Huang, Y.; Li, C. Discriminative feature learning using a multiscale convolutional capsule network from attitude data for fault diagnosis of industrial robots. *Mech. Syst. Signal Process.* **2023**, *182*, 109569. [CrossRef]
35. He, J.; Wang, C.; Jiang, D.; Li, Z.; Liu, Y.; Zhang, T. CycleGAN with an improved loss function for cell detection using partly labeled images. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 2473–2480. [CrossRef] [PubMed]
36. Satchidanandam, A.; Al Ansari, R.M.S.; Sreenivasulu, A.L.; Rao, V.S.; Godla, S.R.; Kaur, C. Enhancing Style Transfer with GANs: Perceptual Loss and Semantic Segmentation. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *11*, 321–329. [CrossRef]
37. Johnson, J.; Alahi, A.; Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In Proceedings of the Computer Vision—ECCV, Amsterdam, The Netherlands, 11–14 October 2016; Volume 14, pp. 694–711.
38. Chen, K.; He, K.; Xu, D. Multi-autoencoder with Perceptual Loss-Based Network for Infrared and Visible Image Fusion. In Proceedings of the 2023 6th International Conference on Image and Graphics Processing, Chongqing, China, 6–8 January 2023; pp. 104–110.
39. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef]
40. Wang, Z.; Bovik, A.C. Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE Signal Process. Mag.* **2009**, *26*, 98–117. [CrossRef]
41. Korhonen, J.; You, J. Peak signal-to-noise ratio revisited: Is simple beautiful? In Proceedings of the 2012 Fourth International Workshop on Quality of Multimedia Experience, Melbourne, VIC, Australia, 5–7 July 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 37–38.
42. Nian, F. *The Traditional Treasure of Chinese Character Printing Fonts*; China Academy of Art: Beijing, China, 2009.
43. Hu, C.; Wu, J. An Essay on the Calligraphy of Deng Shiru. *Academics* **2010**, *7*, 164–173.
44. He, L. *Orchid Pavilion Preface and Its Cultural Significance in Calligraphy*; China Academy of Art: Beijing, China, 2015.
45. Tang, H.; Xu, D.; Sebe, N.; Yan, Y. Attention-guided generative adversarial networks for unsupervised image-to-image translation. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–8.
46. Choi, Y.; Uh, Y.; Yoo, J.; Ha, J.W. Stargan v2: Diverse image synthesis for multiple domains. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 8188–8197.
47. Zhang, S. Appreciation of Wang Xizhi's Classic Calligraphy Work "Orchid Pavilion Preface". *Collect. Investig.* **2022**, *13*, 149–151.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.