Northumbria Research Link

www.northumbria.ac.uk/nrl

northumbria
UNIVERSITY NEWCASTLE

Generative Adversarial Nets in Robotic Chinese Calligraphy*

Fei Chao[1], Jitu Lv[1], Dajun Zhou[1], Longzhi Yang[2], Chih-Min Lin[1,3], Changjing Shang[4], and Changle Zhou[1]

Abstract— Conventional approaches of robotic writing of Chinese character strokes often suffer from limited font generation methods, and thus the writing results often lack of diversity. This has seriously restricted the high quality writing ability of robots. This paper proposes a generative adversarial nets-based calligraphic robotic framework, which enables a robot to learn writing fundamental Chinese strokes with rich diversity and good originality. In particular, the framework considers the learning process of robotic writing as an adversarial procedure which is implemented by three interactive modules including a stroke generation module, a stroke discriminative module and a training module. Noting that the stroke generative module included in the conventional generative adversarial nets cannot solve the non-differentiable problem, the policy gradient commonly used in reinforcement learning is thus adapted in this work to train the generative module by regarding the outputs from the discriminative module as rewards. Experimental results demonstrate that the proposed framework allows a calligraphic robot to successfully write fundamental Chinese strokes with good quality in various styles. The experiment also suggests the proposed approach can achieve human-level stroke writing quality without the requirement of a performance evaluation system. This approach therefore significantly boosts the robotic autonomous creation ability.

I. INTRODUCTION

Applications of robotics in promoting human culture and civilisation, such as robotic writing and drawing, are a major topic which is often neglected by the traditional robotics research. The research of robotic writing focuses on the design of control algorithms to drive robotic end-effectors to write complex characters or letters [1]–[5]. Since Chinese character writing must consider the spatial collocations of character strokes [6], high quality writing must locate well-shaped strokes in the right positions and accordingly the writing quality of Chinese characters essentially relies on the quality of the character strokes. Existing high quality stroke writing requires a calligraphy robot to simultaneously control various joints such that the pen moves accurately. Consequently, only characters existing in the robot's predefined font databases

can be written by the robots. However, this leads the current calligraphy robots to facing a challenging difficulty that these robots cannot generate strokes with good diversity. In other words, the writing results by the current calligraphy robots are all in the same style, lacking of the creativity that human calligraphers usually have.

A number of recent studies applied direct programming methods to attach large font databases to robots control systems in an effort to solve the aforementioned difficulty [7], [8]. Such a method requires complicated and massive human input to convert font information into manipulator's trajectories [1]. Other scientists applied the follow-up ability of manipulators to obtain rich font information [4], [9], [10], which successfully imparts human calligraphic styles to robots. However, the method still entails significant human work to enable manipulators to produce sufficient font information. In addition, several robotic scientists attempted to use deep neural networks to learn writing and drawing abilities [5]. Indeed, it is very appealing that a calligraphy robot possesses a generation mechanism that can automatically produce various styles. In addition, the generation mechanism is desired to have a good learning ability, such that the mechanism can be trained using training data, rather than manually setting up with a big human labor consumption.

This paper proposes a new robotic writing approach using the generative adversarial nets (GAN), which enables a robot to learn to write fundamental Chinese strokes and thus be more appealing. The GAN is a cutting-edge tool for training generative models, and GAN has been widely applied in computer vision and natural language processing [11]. Due to its powerful creation ability, GAN appears to be a promising method for the robotic calligraphy. However, the GAN is not readily applicable to robotic manipulators in generating writing sequences of Chinese strokes. This is because GAN is designed for generating images, but it has difficulties in directly generating sequences of discrete data, such as stroke trajectories or robot motions. More specifically, the utilisation of a robotic manipulator in a GAN model leads to the fact that gradient informant cannot be propagated from the model's output to its generative network. In this case, an alternative training method must be used in order to build the GAN-based robotic calligraphy.

A generative adversarial framework is therefore proposed which uses policy gradient for a robot in learning to write fundamental Chinese strokes. The framework applies the GAN to generate robotic writing trajectories, which are performed by a multiple-degree robotic manipulator. Through the adversarial processes of the framework, the writing

performance is evaluated by the discriminative network and the evaluation results are used as the gradients to train the generative network. As a result, the robot is able to write various types of Chinese strokes. The main contributions of this work are summarized as follows: (1) introducing the GAN to robotic manipulator to generate writing movements, which drives a brush pen to produce various writing types of Chinese strokes (as detailed in Section II-A), (2) adopting the "Policy Gradient" training method in the generative net to solve the difficulty that error information cannot be propagated back to the GAN (as discussed in Section II-D).

The remainder of the paper is organised as follows. Section II details the proposed approach which allows calligraphy robotic to learn to write strokes with good diversity. Section III specifies the experimental set up and discusses the experimental results. Section IV concludes the paper with directions of future work suggested.

II. PROPOSED APPROACH

A. The Approach Overview

The proposed approach, as shown in Fig. 1, consists of three parts: 1) stroke generative module, 2) stroke discriminative module, and 3) training module. The entire method is built upon the GAN. The task of the discriminative module is to distinguish samples from the stroke generative module or from the real training data; meanwhile, the stroke generative module's task is to maximally confuse the discriminative module. Thus, the training objectives of the approach are summarized to: 1) train the discriminative module to maximize the probability of the real stroke data and minimize the probability of the stroke image written by the robot; and 2) train the generative module to minimize the probability that the discriminative module recognizes the robotic written images.

In the stroke generative module, a generative network, G, uses random numbers as input to produce probability distributions of stroke trajectory points. Then, the calligraphy robot applies a sampling method to obtain the stroke's position information from the probability distributions. From this, the robot uses the obtained position to write the stroke, which is captured by a camera representing as a stroke image. The discriminative network, D, receives stroke images, either captured from the written strokes by the robot or sampled from calligraphy textbooks, and produces a discriminative result for each image.

The original training approach proposed in Goodfellow's GAN used the gradient descent method. However, the gradient descent method must face a non-differentiable problem during the training phase in the proposed framework, due to the involvement of the robotic manipulator in the proposed approach. Such problem cannot be solved by the back-propagation algorithm, and thus the policy-gradient typically used in the reinforcement learning algorithm is applied here to train the framework, motivated by the work of Yu et al. [12]. For each output of the G network, the discriminative result generated by the D network indirectly reflects the performance of the G network. In this case, outputs of D

network are used as rewards of the robotic actions, which are generated by the G network.

Therefore, the training objective of G network is changed to obtain maximum rewards in the proposed system. The G network must learn to increase the occurring probability of outputs such that the writing performances can be improved. Based on this consideration, if a stroke writing image of the robot has better performance determined by D, the G network must increase the probability of the robotic trajectory points. The detailed implementations of these three modules are described below.

B. Stroke Discriminative Module

The stroke discriminative module is established by a feed-forward neural network. The network consists of three layers, including input layer, hidden layer, and output layer. The dimension of the input layer is set as 784, since the size of each input image is 28×28. The hidden layer contains 128 neurons; and the dimension of the output is 1. The network's input data X consist of two types of images: 1) real stroke images, X_{real}, extracted from Chinese calligraphy copybooks; and 2) "fake" stroke images, X_{fake}, written by our calligraphy robot. The fake images are captured by a camera mounted on the robot's gripper. The output values predict the probability that x came from the X_{real} data distribution against that from the robot-generated images.

C. Stroke Generative Module

The stroke generative module has a Gaussian noise input, which produces random numbers to seed the generative module. In this paper, a feed-forward neural network is adopted to implement the stroke generative module. The G network also has three layers. Input of G is an array of generated random numbers, Z, whose dimension is set as 128 in this paper. The hidden layer of G has $128 \times T_N$ hidden neurons, where T_N denotes the number of the trajectory points. The output of G contains two probability distributions: 1) trajectory point position distribution, p_{tpp}, and trajectory width distribution, p_{tw}. Thus, the dimension of the output layer, N_{output}, is defined as follows:

$$N_{output} = 784 \cdot T_N + R_N \cdot T_N, \qquad (1)$$

where R_N denotes the level number of trajectory width. In this work, R_N is set as 20, which means a trajectory has 20 types of widths.

For each trajectory point position, a "softmax" function is applied to normalize G network's output; thus, the softmax functions of p_{tpp} and p_{tw} are defined as:

$$p_{tpp}(V)_j = \frac{e^{V_j}}{\sum_{k=1}^{K} e^{V_k}} \qquad (2)$$

$$p_{tw}(S)_i = \frac{e^{S_i}}{\sum_{m=1}^{M} e^{S_m}} \qquad (3)$$

where for p_{tpp}, j denotes the jth pixel of the image and $K = 784$; and for p_{tw}, i denotes the ith level of width and $M = 20$. When all the probability distributions are
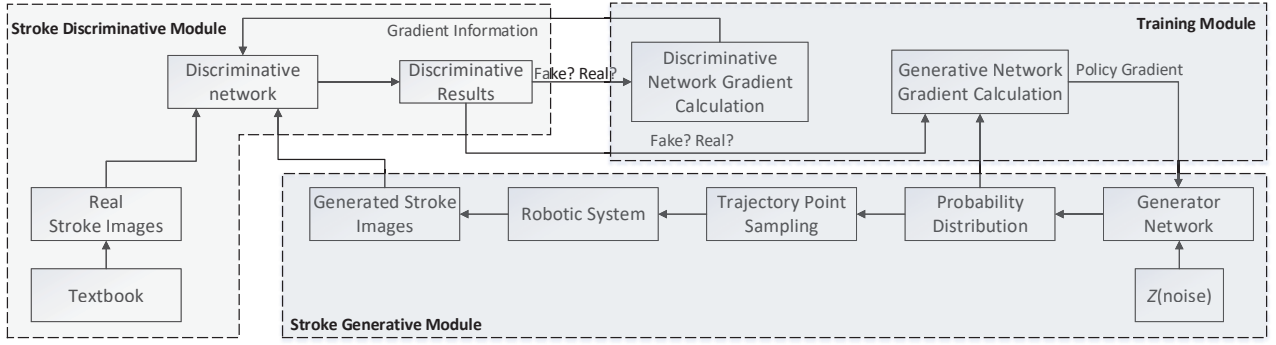
Fig. 1: The flowchart of the proposed approach to robotic handwriting. The approach contains three modules: (a) stroke generative module, (b) stroke discriminative module, and (c) training module.

generated, the random sampling method is used to generate five trajectory points from the probability distributions. After sampling, both the position and width information of the five points are sent to the robotic system.

D. Training Module

The original training objective of GAN can be represented as:

$$
\min_{G} \max_{D} V(D, G) =
$$
$$
E_{x \sim p_r}[\log D(x)] + E_{z \sim p_z}[\log(1 - D(G(z)))], \quad (4)
$$

where $D(\cdot)$ denotes the D network's output; $G(\cdot)$ denotes the G network's output; and $E[\cdot]$ denotes the network expectation. However, each $G(\cdot)$ is a stroke's trajectory, which is written by the robotic system, rather than obtained from an image. Therefore, the objective function can be rewritten as:

$$
\min_{G} \max_{D} V(D, G) =
$$
$$
E_{x \sim p_r}[\log D(x)] + E_{z \sim p_z}[\log(1 - D(W(G(z))))], \quad (5)
$$

where $W(\cdot)$ denotes the writing process of the robotic system. Thus, the D network objective is expressed as the following loss function:

$$
D_{loss} = -E_{x \sim p_r}[\log D(x)] - E_{z \sim p_z}[\log(1 - D(W(G(z))))]. \quad (6)
$$

Based on the proposed approach shown in Fig. 1, the G network must produce a number of trajectory points to obtain higher awards from the D network. Therefore, the training objective is to increase the occurring probability of the trajectories with higher rewards. The loss function of the G network is obtained by:

$$
G_{loss} = E_{z \sim p_z}[(\log \prod_{i=1}^{n} G(z)_i) \cdot D(W(G(z)))], \quad (7)
$$

where n denotes the number of a stroke's trajectory points; $G(z)_i$ denotes the occurring probability of the ith trajectory point; $\prod_{i=1}^{n} G(z)_i$ indicates the occurring probability of a stroke, which is calculated by multiplying the probabilities of all trajectory points of the stroke; and $D(W(G(z))$ denotes

the output from the D network, with value ranging from 0 to 1.

Following Yu et al.'s work [12], τ denotes a stroke trajectory, $R(\tau)$ denotes the trajectory's reward, $P(\tau, \theta)$ denotes the occurrence probability of τ, θ denote the parameters of the policy network (G network); then, the objective of the reinforcment learning is to fined the optimal θ:

$$
\max_{\theta} J(\theta) = \max_{\theta} \sum_{\tau} \log P(\tau, \theta) R(\tau) \quad (8)
$$

Use $D_{\theta}(W(G_{\theta}(z)))$ to replace $R(\tau)$ and use $G_{\theta}(z)_i)$ to replace $P(\tau, \theta)$, the gradient of the objective function $J(\theta)$ with the G network's parameters is obtained by:

$$
\nabla_{\theta} J(\theta) = E_{z \sim p_z}[\nabla_{\theta}(\log \prod_{i=1}^{n} G_{\theta}(z)_i) \cdot D_{\theta}(W(G_{\theta}(z)))]. \quad (9)
$$

Since the expectation $E[\cdot]$ can be approximated by sampling, the G network's parameters are then updated by:

$$
\theta \leftarrow \theta + \alpha \nabla_{\theta} J(\theta), \quad (10)
$$

where α is the learning rate. Since the D network is implemented by a MLP network, and its training is implemented using the advanced gradient algorithms such as Adam and RMSprop.

A training process for the above model is defined here. The G and D networks are trained alternatively. As the G network gets progressed via training on g-steps updates, the D network must be retrained periodically to retain a good training pace with the G network. When training the D network, real stroke examples are extracted from calligraphy textbooks; whereas fake examples are generated from the calligraphy robot. In order to keep the balance, the number of fake examples generated for each d-step is equivalent to that of the real examples. The entire training procedure is illustrated in the pseudo-code as detailed in Algorithm 1.

E. Robotic System

Fig. 2 illustrates the robotic hardware system, which includes a 5-DOFs industrial robotic arm, a camera, and a writing board. The positions of both the robotic arm and the writing board are fixed. Four of the arm's five joints are

Algorithm 1 Training Phase of GAN-based Calligraphic Robot

Require: Real stroke image dataset X_{real} and random number Z;

1: Initialize G and D with random weights;
2: **repeat**
3:    **for** g-step **do**
4:       Produce a new random number Z;
5:       Input Z into G;
6:       **for** t in $1 : T_N$ **do**
7:          Sample a trajectory point position based on Eq. 2;
8:       **end for**
9:       **for** t in $1 : T_N$ **do**
10:         Sample a trajectory point width for each trajectory point;
11:       **end for**
12:       Robot writes the trajectory, then the writing result is convert to an image;
13:       Update G parameters via policy gradient in Eq. 10;
14:    **end for**
15:    **for** d-step **do**
16:       Produce current G and robot to generate new trajectory images and combine with X_{real};
17:       Train D by Eq. 6;
18:    **end for**
19: **until** GAN Converges

---

applied to achieve a writing task. A soft pen is mounted at the tip of the arm, and the writing takes place within the arm's working range. Then camera, mounted above the soft pen, captures input Chinese character pictures, which are taken from Chinese calligraphic textbooks. The inverse kinematics analysis of the robot arm is based on forward kinematics. Thus, if the positions of the manipulator are obtained, the manipulator's four joints are calculated by inverse kinematics by following the approach specified in the work of [9].

A conversion function is created to convert positions of the stroke trajectory to that in the calligraphic robot coordinates. The function is defined as:

$$\begin{cases} x_r = x_S + \gamma \cdot x_p \\ y_r = y_S + \gamma \cdot y_p \\ z_r = z_S + \tau \end{cases} \quad (11)$$

where γ denote a scale parameter, which controls the size of each stroke; x_S, y_S, and z_S jointly define an initial position for each stroke; and τ is obtained by sampling R_N in G.

III. EXPERIMENTATION

The approach proposed above was applied to a task of Chinese character stroke writing, for system validation and evaluation. The information and examples of training data (obtained from "real" images included in calligraphic book) were established first. Then, the training procedure and writing actions took place, and the writing results demonstrated
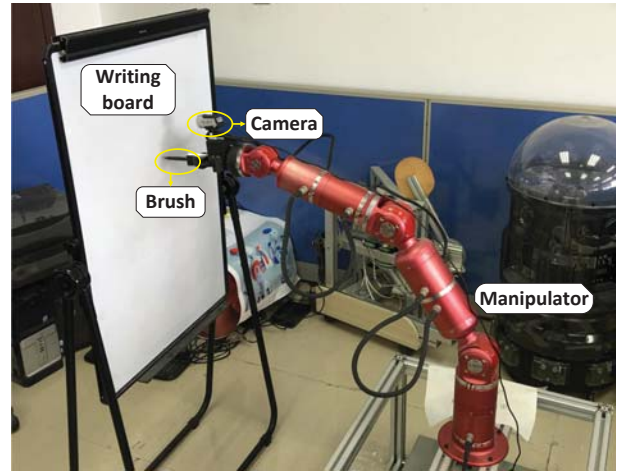

Fig. 2: The robotic hardware for writing Chinese strokes.

the learning performance of the policy gradient method. In the experiments, the dimension of the noise space was set as 128; and the value of each noise data instance was ranged from -1 to 1.

A. Training Data

In the experiment, six different Chinese character strokes were used to train the proposed approach. These strokes were extracted from simple Chinese characters from Chinese calligraphic textbooks. The stroke extraction method was created by using the work of Lian et al. [13]. Each stroke has more than 500 samples; thus, the total number of the training samples is more than 3,000. Some illustrative training samples of each stroke is shwon in Fig. 3. Each row shows one type of a stroke with various variants. The stroke types from the top to bottom rows are: "short left-falling stroke", "horizontal stroke", "horizontal and left-falling stroke", "right-falling stroke", "long left-falling stroke", and "vertical, turn-right and hook stroke".
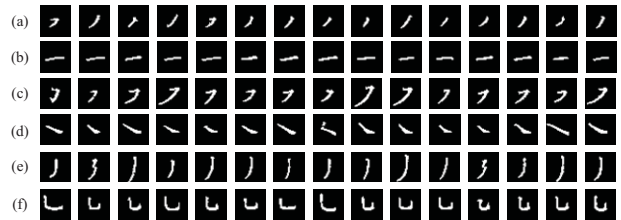

Fig. 3: Illustrative training samples used in the experiment, each row shows one type of a stroke with various variants.

B. Training Phase and Writing Results

Several randomly-selected robotic writing results of the "vertical, turn-right and hook" stroke is shown in Fig. 4; this stroke is the most complicated one used in this experiment, during the training phase. This figure shows the writing results of three stages: 1) early stage, 2) medium stage, and 3) final stage. The training phase of the other six strokes showed similar training situations with that of the "vertical,
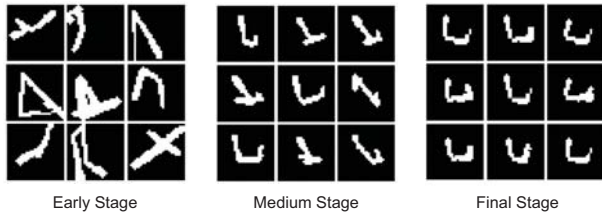
Fig. 4: Robotic writing results during the training stages.

Fig. 6: Training epoches of the vertical, turn-right and hook stroke.

turn-right and hook" stroke. Note that, the stroke images were processed by the "Invert Color" (convert the black trajectories with white background to the white trajectories with black background); since the image patters of both the training data and generated data must be identical.

All the nine writing results in the early stage are amorphous, and consequently it is very difficult to recognize the written stroke. In contrast, the samples in the medium stage showed several quality writing results, that is, the bottom-left trajectory has showed a rough shape of the target stroke. However, the rest trajectories are still far different from the target stroke. The writing results in the final stage demonstrated very high quality performance. Every generated trajectory in this stage is visually close to the target stroke; in particular, the top-left and bottom-right ones exhibit consistent performance with human-level writing.
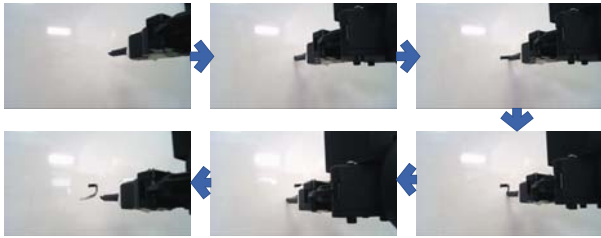

Fig. 5: Robotic arm writing a stroke in action.

As an example of the robot's writing actions, Fig. 5 shows the robotic arm writing the "horizontal and left-falling stroke" stroke in action. The arrows in the figure indicate the action sequence. The gripper holding a soft pen starts its writing at a predefined starting position; then, the arm moves by following the stroke's trajectory points, which have been converted to joint angles of the arm by inverse kinematic equations used in [9]. When all trajectory points are traveled, the gripper returns to the starting position.

The evaluation results of the G network's output on the long left-falling stroke is illustrated in Fig. 6. The performance line shown in this figure was calculated by the D network using Eq. 6. The training epochs of this stroke was set as 5,000. At the first several hundred training epochs, the D network could not have the ability to determine whether each image was generated by the manipulator; therefore, the evaluation scores were assigned randomly. The average score during this stage was around 1.4, $(2 \times \ln 0.5)$. Then, the D network gained the discriminative ability more rapidly; in contrast, the G network did not generate quality writing. In
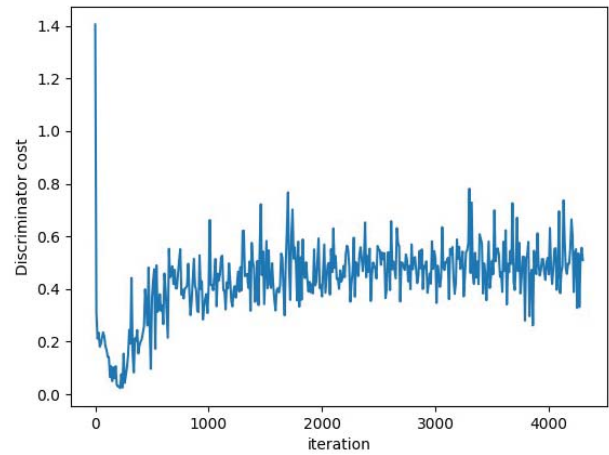
this case, the score decreased sharply. However, due to the policy-gradient mechanism generated heuristic feedback information, several good quality writing results had appeared. After around 500th epochs, the score gradually climbed; although the slope was not steep, the score finally reached at over 0.6, which was not very close to the ideal network's output; however, the writing quality has reached to a very high level. Furthermore, because of the sampling process in the framework, the curve contains many significant shocks.

The evaluation curves of the rest four strokes are very similar to that of the long left-falling stroke shown in Fig. 6. The numbers of the training epochs of the six strokes are 8,000 for the short left-falling stroke, 5,000 for the horizontal stroke, 14,500 for the horizontal and left-falling stroke, 14,000 for the right-falling stroke, 5,000 for the long left-falling stroke, and vertical, and 18,000 for the turn-right and hook stroke.

Fig. 7 shows the final writing results of all the six strokes without invert-color process. These writing results were obtained by inputting various random numbers to the G network, when the training phase had completed. In other words, each random numbers had its corresponding output trajectory. After writing a stroke, the robot moved to a predefined position to capture the trajectory by an image through the mounted camera. Each stroke type had 64 samples. The experiment demonstrated that all these trajectories were not identical for each type of strokes. By inputting different random number, each writing trajectory was unique which provides good writing style diversity. Therefore, the proposed approach is able to address the font limitation problem as discussed in Section I.

Several trajectories generated by the robot using the proposed approach were very close to those written by human, in addition to fact that the overall writing quality of the writing results are better than those of our previous studies. For example, as shown in Figs. 7 (a) short left-falling and 7 (d) right-falling strokes, the robot obtained the ability to control changes in thickness while it was writing one trajectory. This

(a) Short left-falling stroke
(b) Horizontal stroke
(c) Horizontal and left-falling stroke
(d) Right-falling stroke
(e) long left-falling stroke
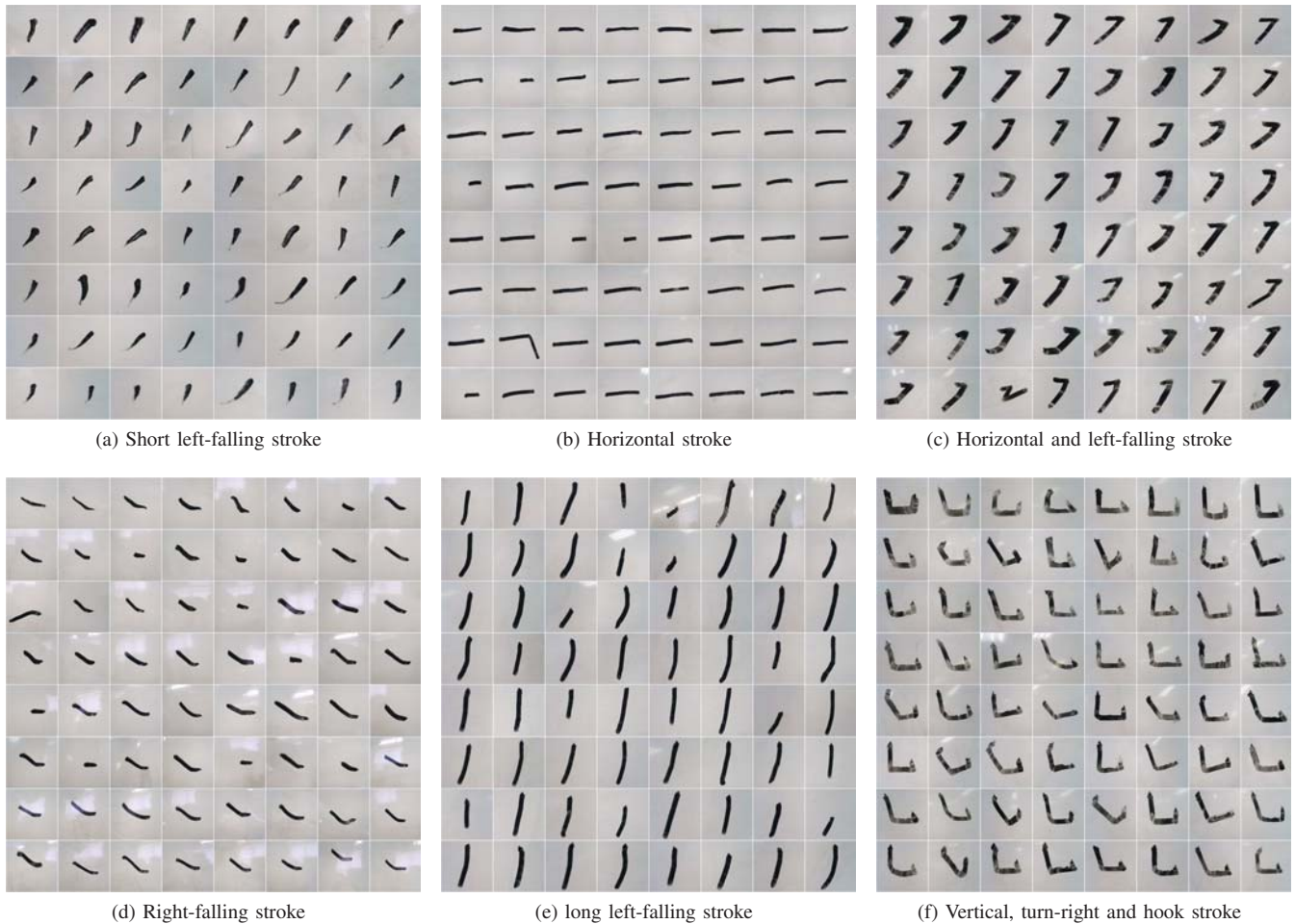(f) Vertical, turn-right and hook stroke

Fig. 7: Final writing results of all the six strokes.

ability required the robot to adjust the distance between the pen and the write board. Due to the width output "softmax" in the G network, the robot was able to handle such difficult task. Furthermore, the last segment (the hook) of the 7 (f) stroke is another complicated structure for calligraphy robots, since the trajectory in the hook becomes short and sharp. However, several trajectories generated based on the proposed approach include the hook with quality shape, which also proves the success of our approach.

It is necessary to note, however, that a few incorrect trajectories still existed in the final writing results. In particular, the errors of several incorrect trajectories in the 7 (b) horizontal and 7 (d) right-falling strokes are very obvious. These errors might be caused by the noise of the training data (incorrect strokes existed in the training data) simply imply that the training still require more training epochs.

C. Discussion and Comparison

The experimental results demonstrated the ability of the proposed approach in writing Chinese character strokes. In contrast to existing approaches reported in the literature, the proposed one has three distinctive advantages:

1) Without the requirement in creating evaluation mechanisms by human engineers: A large number of existing robotic calligraphy systems did not involve an evaluation mechanism to assess their writing results. Since these robot systems did not use the close-loop structure, these systems did not possess the learning ability and did not improve the writing performance by the robots themselves. Also, existing robotic writing systems via learning must contain the evaluation mechanisms, so that these systems can have the gradient information or optimisation objectives. Such evaluation mechanisms must be established by algorithm engineers, which requires significant human efforts.

In contrast, our system alternatively takes the full advantage of the discriminative network from the GAN model, which is able to automatically build the evaluation mechanism without human intervention. The proposed system only requires sufficient training data, which can be readily extracted from existing stroke database or Chinese character decomposing algorithms, and thus, the proposed robot system can learn the probability distributions of the training data. From this sense, the trained discriminative networks should be able to be used in evolutionary computation algorithms, where the trained discriminative networks work as the fitness function. The further investigation on this

remains as future work.

2) Diverse font shapes: The font database based calligraphy robot systems suffer from limited stroke styles. Also, the learning-based calligraphy robots merely imitate to writing shapes of the target strokes, the number of which is still limited. However, the generative network produces massive number of various trajectories for each stroke. In addition, our generative module is able to adjust the trajectory width. therefore, one trajectory with various widths can exhibit different shapes. From this, the proposed approach is able to address the limitation of the styles of stroke styles in the existing approaches.

Since the diversity is driven by the network's random inputs, it is difficult to control the robot to follow a specific style, and actually the system just randomly generates different trajectory shapes for each stroke. Therefore, it is necessary to use labels to determine the stroke types. Fortunately, several cutting-edge GAN modules have addressed this, such as conditional-GAN [14] and InfoGAN [15]. In particular, the Info-GAN can automatically find the distributions of stroke types. Therefore, the introduction of these new types GAN into the proposed system can further develop the robotic writing ability, which again remains as future work.

3) High writing quality with simple learning system: Using GAN to generate realistic images has become a hot research topic in computer vision and machine learning areas. The application of a GAN model in stroke images generation is not a creative study, but it is interesting in this work to involve a robotic manipulator using a brush pen to write Chinese strokes. Several important trajectories in Chinese calligraphy require multiple complicated writing motions of human calligraphers. It is hard to obtain human-level writing performance using only computational generation methods. In contrast, the proposed approach is able to simulate human-level writing from writing hardware, which is therefore very promising in calligraphic robotics.

IV. Conclusion

This paper presented a new generative adversarial nets-based robotic writing system, which is able to write different Chinese strokes in different styles. The system takes real stroke images produced from Chinese calligraphy textbooks for training. During the training phase, the policy gradient information was applied to solve the problem that gradient cannot be back-propagated. The experimental results based on six strokes demonstrated the working of the proposed method, and several results achieved human-level quality. Coming with a Chinese character decomposing algorithm, our method is able to write Chinese characters with multiple styles.

While the proposed approach is promising, there is room for improvement. First, only the conventional GAN is adopted in the present work; and it is interesting to investigate if other GAN variants, such as conditional-GAN [14], can be applied to generate better results. Second, stroke writing sequence is another crucial issue in robotic writing [16], [17], which might be addressed by the application of recurrent neural network and thus further study is required. Third, the proposed algorithm ignored the trajectory order of the sampling points, further efforts will focus on this.

References

[1] H. Zeng, Y. Huang, F. Chao, and C. Zhou, "Survey of robotic calligraphy research," *CAAI Transactions on Intelligent Systems*, vol. 11, no. 1, pp. 15–26, 2016.

[2] Z. Ma and J. Su, "Aesthetics evaluation for robotic Chinese calligraphy," *IEEE Transactions on Cognitive and Developmental Systems*, vol. To appear, 2016.

[3] S. Mueller, N. Huebel, M. Waibel, and R. D'Andrea, "Robotic calligraphy - Learning how to write single strokes of Chinese and Japanese characters," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 1734–1739.

[4] Y. Sun, H. Qian, and Y. Xu, "Robot learns Chinese calligraphy from demonstrations," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 4408–4413.

[5] K. Sasaki, K. Noda, and T. Ogata, "Visual motor integration of robot's drawing behavior using recurrent neural network," *Robotics and Autonomous Systems*, vol. 86, pp. 184–195, 2016.

[6] F. Chao, Y. Sun, Z. Wang, G. Yao, Z. Zhu, C. Zhou, Q. Meng, and M. Jiang, "A reduced classifier ensemble approach to human gesture classification for robotic Chinese handwriting," in *IEEE Conference on Fuzzy Systems (FUZZ-IEEE)*, 2014, pp. 1720–1727.

[7] X. Ma, Q. Kong, W. Ma, and X. Zhang, "4-DOF lettering robots trajectory planning," in *Mechanical Engineering and Automation*, 2010, vol. 165, no. 5, pp. 161–163.

[8] Y. Man, C. Bian, H. Zhao, C. Xu, and S. Ren, "A kind of calligraphy robot," in *IEEE International Conference on Information Sciences and Interaction Sciences*, China, 2010, pp. 635–638.

[9] F. Chao, Y. Huang, X. Zhang, C. Shang, L. Yang, C. Zhou, H. Hu, and C.-M. Lin, "A robot calligraphy system: From simple to complex writing by human gestures," *Engineering Applications of Artificial Intelligence*, vol. 59, pp. 1–14, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197616302329

[10] F. Chao, F. Chen, Y. Shen, W. He, Y. Sun, Z. Wang, C. Zhou, and M. Jiang, "Robotic free writing of Chinese characters via human robot interactions," *International Journal of Humanoid Robotics*, vol. 11, no. 1, pp. 1 450 007–1–26, March 2014.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[12] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient." in *AAAI*, 2017, pp. 2852–2858.

[13] Z. Lian, B. Zhao, and J. Xiao, "Automatic generation of large-scale handwriting fonts via style learning," in *SIGGRAPH ASIA 2016 Technical Briefs*, ser. SA '16. New York, NY, USA: ACM, 2016, pp. 12:1–12:4. [Online]. Available: http://doi.acm.org/10.1145/3005358.3005371

[14] M. Mirza and S. Osindero, "Conditional generative adversarial nets," arXiv preprint arXiv:1411.1784, 2014.

[15] X. Chen, X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2016, pp. 2172–2180.

[16] H. I. Lin and Y. C. Huang, "Visual matching of stroke order in robotic calligraphy," in *2015 International Conference on Advanced Robotics (ICAR)*, July 2015, pp. 459–464.

[17] Z. Ma and J. Su, "Stroke reasoning for robotic chinese calligraphy based on complete feature sets," *International Journal of Social Robotics*, vol. 9, no. 4, pp. 525–535, Sep 2017. [Online]. Available: https://doi.org/10.1007/s12369-017-0410-2