Generating synthetic styled Chu Nom characters

Jonas Diesbach[1], Andreas Fischer[1,2], Marc Bui[3], and Anna Scius-Bertrand[1,2,3]

[1] iCoSys, HES-SO, Fribourg, Switzerland
{jonas.diesbach, andreas.fischer, anna.scius-bertrand}@hefr.ch
[2] DIVA, University of Fribourg, Switzerland
[3] EPHE-PSL, Paris, France
marc.bui@ephe.sorbonne.fr

Abstract Images of historical Vietnamese steles allow historians to discover invaluable information regarding the past of the country, especially about the life of people in rural villages. Due to the sheer amount of available stone engravings and their diverseness, manual examination is difficult and time-consuming. Therefore, automatic document analysis methods based on machine learning could immensely facilitate this laborious work. However, creating ground truth for machine learning is also complex and time-consuming for human experts, which is why synthetic training samples greatly support learning while reducing human effort. In particular, they can be used to train deep neural networks for character detection and recognition. In this paper, we present a method for creating synthetic engravings and use it to create a new database composed of 26,901 synthetic Chu Nom characters in 21 different styles. Using a machine learning model for unpaired image-to-image translation, our approach is annotation-free, i.e. there is no need for human experts to label character images. A user study demonstrates that the synthetic engravings look realistic to the human eye.

Keywords: Synthetic handwriting generation · Generative Adversarial Networks (GAN) · Contrastive Unpaired Translation (CUT) · historical Vietnamese steles · Chu Nom characters.

1 Introduction

Created mostly between the 17th to 19th century, Vietnamese steles contain very valuable information about the history of the country [17,18]. The reason for this is that the current knowledge of Vietnamese history is mainly based on annals from the royal court and clergy, and those official records only tell about the great national history, diplomatic conventions, nomination of mandarins and wars. In contrast, the steles are the sole historical source able to describe the village life. They not only speak about the large history of the country, but also about the social, economic and cultural life of the rural communities. In short, they contain a vast amount of important information for historians.

To preserve the ancient history written on the stone steles, whose average size exceeds a square meter, the French School of the Far East (EFEO) began creating a collection of stamps between 1910 to 1954. This work was then later continued by the Han-Nôm institute, starting in 1995. A stamp is created by first gluing paper onto the steles with banana-juice acting as the adhesive. Then, an ink-covered roll is used to paint over the whole sheet leaving the characters in white and coloring the background in black. Thus, creating a faithful representation of the engravings while keeping the original scale. Finally, the stamps are photographed to obtain digital images of the steles. Examples of stele images are shown in Figure 1, illustrating the heterogeneous nature of both the stele backgrounds as well as the engravings.

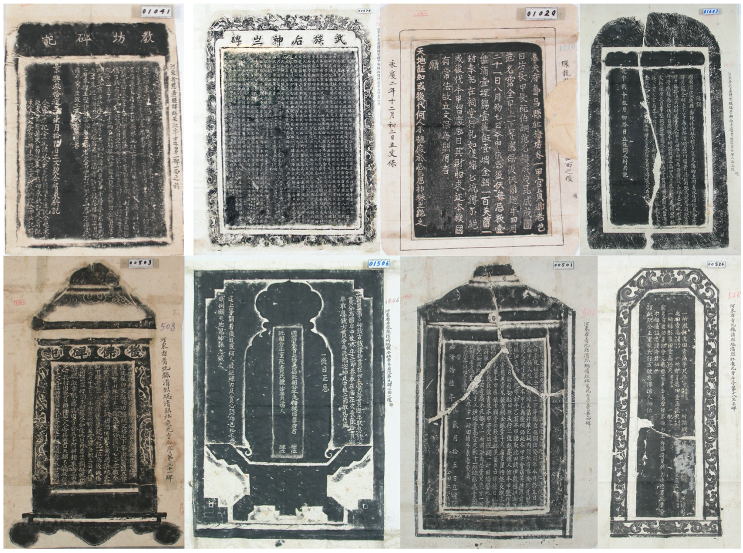


Figure 1: Historical Vietnamese stele images.

Many steles have now disappeared, but their stamps remain. Because there exist images of thousands of unique steles, automated reading of the Chu Nom characters would greatly support the historians, who are currently investigating the steles in the context of the Vietnamica[4] project. A significant step towards automated reading has been taken in [22], where an annotation-free character detection system has been introduced. Avoiding human labelling effort, the system is using printed Chu Nom characters to train deep convolutional detection networks and performs unsupervised self-calibration in order to adapt to to the stele images. The result of this study are hundreds of thousands of automatically detected character images.

The next step would be not only to detect where the characters are located but also to read them. Due to the lack of annotated data, a promising

[4] https://vietnamica.hypotheses.org/

line of research is to generate a large amount of synthetic engravings that are close to the real characters in order to train systems for keyword spotting and automatic transcription. In this paper, we leverage the automatically detected character images and use generative adversarial networks (GANs) to transfer different engraving styles to printed Chu Nom characters. The synthesis remains annotation-free for creating characters, that is no human labelling of character images is required. However, there are still a few human interactions necessary for the proposed method. First, we use a printed Chu Nom font that was created manually in the past (but not specifically for our task). Secondly, a brief human interaction was necessary to manually choose 21 different engraving styles among the stele images.

In the present paper, we rely not only on CycleGAN [29], but also on recent advancements in image-to-image translation, namely we use Contrastive Unpaired Translation (CUT) [19], a machine learning model which is based on the GAN framework and uses contrastive learning. The latter model is used to generate a total of 21 synthetic engraving styles of about 26,901 printed Chu Nom characters each. We made the database freely available[5] with the aim to support the development of automatic reading systems in the future. To evaluate the general quality of the synthesis, we have conducted a user study that will be discussed below.

2 Related Work

Various methods to generate styled handwritten word or character images have been studied. Many of which are based on GAN, a fairly recent and powerful generative model, which is based on a generator-discriminator architecture where the two networks compete against each other to improve the generated results. Due to the impressive results achieved by GANs as well as their wide applicability, they have also been largely adopted for synthetic image generation tasks.

In the context of Latin scripts, one of the most recent works focusing on the generation of handwritten words based on the Latin alphabet is GANwriting [11], which generates words conditioned on a given style. Even more recently, HiGAN [3] was proposed, which, in contrast to GANwriting, can generate variable-length handwritten words. Both of those approaches use a threepart objective: An adversarial loss to discriminate between real and fake samples, a word recognition loss to preserve the textual content and a writer identification loss to match the calligraphic style. Unfortunately, we cannot use a word (or more accurately in our case, a character) recogniser, because we do not have the necessary labelled data. Removing the recogniser part from those networks noticeably affects the results, which is why we decided to not further investigate in this direction.

Other works include SC-GAN [5] and JointFontGAN [27], both of which would require skeletons of the extracted Chu Nom characters, which are not

[5] https://github.com/asciusb/21SyntheticStylesNom-Database/

available. Because of the noise in the image, the skeletons risk to not be readable without a considerable work of denoising before. GlyphGAN [7] is a style-consistent font generation model, which takes a style and a character class vector as its input. The latter is a one-hot encoded vector associated with the character class of each sample image. In our case, this would require manual annotation of each and every individual stele character image with its character class label, which is not feasible.

In the context of Asian logograms, many different works to generate Chinese characters have already been proposed. One of them is zi2zi [23], which is an extension of the Pix2Pix framework [9] specifically built to model Chinese characters. Other works such as CalliGAN [26] and MTfontGAN [25] show that multiple calligraphy styles of a character can be generated using a single model.

Further proposed methods all serve different purposes: [24] handles the generation of thin strokes, SCFont [10] tries to generate characters with correct structure and without artefacts, MSMC-CGAN [13] generates realistic multi-scale and multi-class handwritten characters, TH-GAN [1] helps improve historical Chinese character recognition and [20] uses a DenseNet-Pix2Pix model to restore incomplete calligraphy fonts. Furthermore, LSCGAN [12] proposed a stroke-based font generation method where the styles of two existing font characters are fused together to create a new style and FontGAN [14] synthesises characters with a specified style using character stylisation and de-stylisation to improve content consistency. However, not all proposed works focus exclusively on Chinese characters. For example, [15] uses the Pix2Pix architecture to generate Bangla characters and DM-Font [2] decomposes each glyph into several components (sub-glyphs) before reassembling them to new Korean or Thai glyphs. Unfortunately, all of the aforementioned methods have one thing in common: They all use paired training data. Such a dataset would require a large amount of annotated steles, which do not yet exist.

Due to the lack of paired training data, a model which works with unpaired data is needed. And indeed, such a method has already been published in 2021 to generate handwritten Chinese characters. The model, named StrokeGAN [28], is based on CycleGAN and introduces a one-bit stroke encoding to alleviate the mode collapse issue. Unfortunately, this method cannot be easily applied to Chu Nom characters, since the required stroke encodings are not available.

In the present paper, we chose two annotation-free models for generating synthetic Chu Nom engravings that do not require paired data, namely a CycleGAN model built on Pix2Pix and a CUT model. Both are described in more detail in the next section.

3 Unpaired Generative Adversarial Networks

The concept of Generative Adversarial Networks was introduced in 2014 by Ian Goodfellow et al. [4]. They then became rapidly popular because of their successful application in various domains, such as image processing, computer vision, music generation, natural language processing and also in the medical field [6,8]. GANs proved to be especially useful for image-to-image translation tasks, e.g. CycleGAN [29].

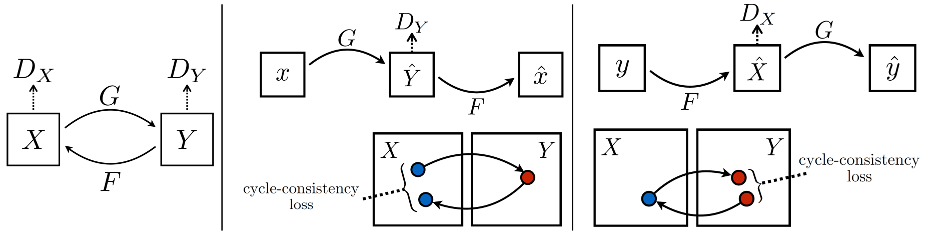
The basic underlying structure of a GAN consists of a generator and a discriminator model. The goal of the generator is to create samples which look indistinguishable from real ones. The discriminator, which is usually a binary classifier, tries to accurately discriminate between real and generated samples. The general idea of this network is that the generator tries to deceive the discriminator and the discriminator tries to detect the generated samples from the generator. Thus, GANs introduced the concept of adversarial learning.

One of the main issues with GANs is non-convergence, of which the most common catastrophic problem is mode collapse [4]. This problem occurs when the generator learns to map several different input values to the same mode, even though samples of the missing modes existed in the training data. In the worst case of a complete collapse, the generator produces only a single output.

Another typical limitation of GANs is the requirement of paired samples between the source domain and the target domain. In the following, two unpaired GAN models are described, which are able to overcome this limitation, namely CycleGAN and CUT.

3.1 CycleGAN

CycleGAN [29], which is built upon the Pix2Pix framework, is able to translate images from a source domain X into a target domain Y when there is no paired data available. The goal is to learn a mapping $G : X \rightarrow Y$ such that the generated samples $G(x)$, where $x \in X$, are indistinguishable from the real images $y \in Y$. An adversarial loss $\mathcal{L}_{GAN}$ is used for this purpose. However, this translation does not suffice to produce compelling results, because the mapping G is highly under-constrained. Moreover, optimisation of the standard adversarial objective ($\mathcal{L}_{GAN}$) in practice often leads to mode collapse, which is why an inverse mapping $F : Y \rightarrow X$ is introduced to exploit the cycle consistency property of translations (see Fig. 2a).



(a) (b) (c)

Figure 2: Overview of the CycleGAN architecture and the two cycle-consistency losses. Images from [29].

The meaning behind cycle consistency is that a translation from source domain to target domain and back should return the original sample, i.e. the learned mapping functions F and G should be cycle-consistent and thus enforce $x \rightarrow G(x) \rightarrow F(G(x) \approx x$ (forward cycle-consistency, see Fig. 2b) and $y \rightarrow F(y) \rightarrow G(F(y) \approx y$ (backward cycle-consistency, see Fig. 2c). This behaviour is encouraged by the following cycle consistency loss:

$$\mathcal{L}_{cyc}(G,F) = \mathbb{E}_x[\|F(G(x)) - x\|_1] + \mathbb{E}_y[\|G(F(y)) - y\|_1]. \tag{1}$$

where $\mathbb{E}_x, \mathbb{E}_y$ are the expected values with respect to the source and target domain, respectively.

Additionally, adversarial losses are applied to both mapping functions using a discriminator D_X to distinguish between images x and generated images $F(y)$ and similarly, a discriminator D_Y to discriminate between y and $G(x)$. Combining the aforementioned losses yields the final objective of the CycleGAN model:

$$\begin{aligned}\mathcal{L}_{CycleGAN}(G,F,D_X,D_Y) =& \mathcal{L}_{GAN}(G,D_Y) + \mathcal{L}_{GAN}(F,D_X)\\ &+ \lambda\mathcal{L}_{cyc}(G,F).\end{aligned} \tag{2}$$

3.2 CUT

Recently, a new unpaired image-to-image translation model using contrastive learning named CUT [19] was proposed by the authors of CycleGAN. The idea behind this method is to maintain the content of the source image by maximising the mutual information between corresponding input and output patches, which is done via a noise contrastive estimation (NCE) framework [16]. This method only requires to learn a mapping in one direction, which simplifies the training. To do so, the generator G is split up into an encoder and a decoder. Combined sequentially, they generate an output image $G(x) = G_{dec}(G_{enc}(x))$, where x is an image from the source domain X. A sample image-to-image translation problem demonstrating this method is displayed in Figure 3.

In contrastive learning, a *query* z should be strongly associated to its corresponding *positive* counterpart z^+ while being disassociated from all N *negatives* z_n^-. An $(N+1)$-way classification problem is created, where the softmax cross entropy loss $\ell(z, z^+, z^-)$ is used to predict the probability of the query belonging to the same class as its positive.

The query, positive and negatives are sampled from different layers $l \in \{1,\ldots,L\}$ and spatial locations $s \in \{1,\ldots,S_l\}$ of the encoder and are passed through a small two-layer Multilayer Perceptron H_l to project both the input and output patches into a shared embedding space. With respect to the embedded query z_l^s sampled from the output image $G(x)$, the embedded positive $z_l^{s,+}$ at the same position in the input image x, and embedded negatives $z_l^{S\backslash s,-}$ at positions different to s in the input image, the PatchNCE loss is defined as

$$\mathcal{L}^X_{PatchNCE}(G,H) = \mathbb{E}_x[\sum_{l=1}^{L}\sum_{s=1}^{S_l}\ell(z_l^s, z_l^{s,+}, z_l^{S\backslash s,-})]. \tag{3}$$

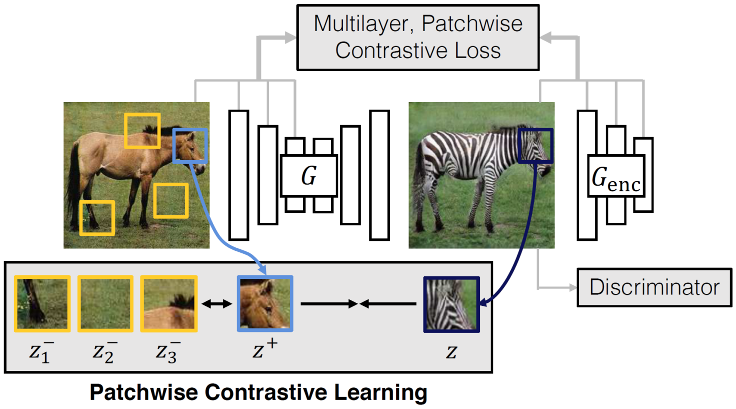

Multilayer, Patchwise Contrastive Loss
G
G_{enc}
Discriminator
z_1^-
z_2^-
z_3^-
z^+
z
Patchwise Contrastive Learning

Figure 3: Patchwise Contrastive Learning method overview. Figure from [19].

The same loss $\mathcal{L}^Y_{PatchNCE}(G, H)$ can also be computed for images of the target domain to prevent the generator from making changes to images of the target domain. Finally, by combining the GAN loss with the two PatchNCE losses, the objective function of the Contrastive Unpaired Translation (CUT) is defined as

$$\mathcal{L}_{CUT}(G, D, H) = \mathcal{L}_{GAN}(G, D) + \mathcal{L}^X_{PatchNCE}(G, H) + \mathcal{L}^Y_{PatchNCE}(G, H). \quad (4)$$

4 Data

The combined collection of stone engravings by the EFEO and the Han-Nôm institute comprises about 40,000 unique copies. From all those stele stamp images, we had a subset of 2,036 images at our disposal. A single image consists of around three million pixels on average. The only preprocessing step performed on these images was to invert their colour, turning the characters black. Note that all those images are of varying sizes, have different amounts of text columns and characters in total. Furthermore, certain steles have irregular text columns or even columns which split into two. Other steles show clear signs of deterioration, which were caused by the weather or are simply due to their age. Most notably, though, is the huge range of different styles, be that due to the size (stele itself, text area, characters, borders), the ornamentation or the handwriting [21].

In addition to the stele images, we also had the bounding box coordinates of automatically detected characters at our disposal for about 1,800 steles, which were obtained by the annotation-free character detection system [22]. For this subset, the detection system was able to locate at least 100 characters on each stele with sufficient confidence.

Important to note here is that the bounding boxes not necessarily encircle an actual character, because the coordinates only show where the detection system assumes a character is located. Note further that the characters were only detected automatically but not classified, i.e. we do not know to which characters they correspond to. The average size of the extracted character images is 19.1 × 20.0 pixels. The set of all character images constitutes the superset of style target domains used in future image-to-image translations. The source domain consists of 26,901 printed characters (black font on white background), which are based on a Chu Nom font, courtesy of the Vietnamese Nom Preservation Foundation[6].

5 Evaluation

To evaluate the performance of the proposed method for generating synthetic Chu Nom characters, we comment in the following sections on the training setup and behavior of CycleGAN and CUT, followed by a user study to assess whether or not the generated characters look realistic to the human eye.

5.1 Setup

Because neither a single CycleGAN nor CUT model can generate multiple different styles, we tried using K-Means clustering to find character images from various steles with similar engraving styles such that a large range of styles can be covered with few trained models. Unfortunately, this did not work well, since the difference between character images from the same cluster was too large.

Instead of retraining a model on a new style for some additional epochs, we decided to manually select a handful of styles, because we achieved better results with individual models. Therefore, we combed through a subset of 300 steles, where we chose 21 steles from which we extracted the characters in order to create the target domain set for the training of the models. The advantages of this approach are twofold. First, a noticeable difference between the various styles can be guaranteed, which would have not been the case with automated clustering. Second, a good quality of training data can be ensured, because steles with a lot of poorly detected characters can simply be discarded as long as there exists a stele of a very similar style with better detection results.

The datasets used to train the two models consist of a source and a target set. The extracted character images from a single engraving style constitute the target domain. Depending on the selected stele, there are as few as 202 images and as many as 544 images. If there were more than 600 detected characters of a given stele, the set was reduced to 400 images by randomly removing characters to improve the models training performance. On average, for the 21 engraving styles, there were 366 images in the target domain. The source domain consists of the same amount of binary printed font character images, which were created anew for each style.

[6] http://www.nomfoundation.org

5.2 CycleGAN

To train the model on our data, we used the default settings of CycleGAN, the sole change being a smaller input image size. Concretely, we only scaled the extracted character images up to 128 × 128 pixels instead of 256 × 256. The networks are trained from scratch over 200 epochs with a learning rate of 0.0002 for the first 100 epochs. The learning rate linearly decays over the remaining 100 epochs to zero and the hyperparameter λ in Equation 2 is set to 10.

Figure 4 displays a selection of sample results after training the CycleGAN model on one of the engraving style datasets. The generated images (middle row) should show the source characters (top row) in the style of target domain (bottom row), but they clearly display different occurrences of the mode collapse issue.

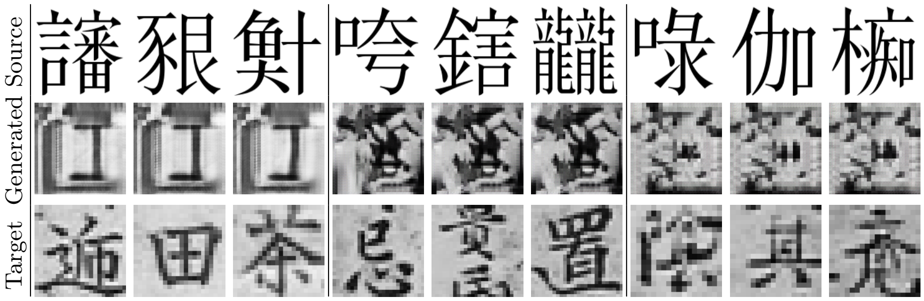


Figure 4: Three separate instances of a CycleGAN mode collapse.

5.3 CUT

Similar to CycleGAN, we have also used the standard setup of CUT when training the model, i.e. no parameters were fine-tuned. The source and target domain images were scaled up to 256 × 256 pixels and the networks were trained for 400 epochs, which takes around 12 to 13 hours on a GeForce GTX 1080 with a training dataset consisting of 400 images in both domains. Once a model has been trained, a single character image can be generated in around one eight of a second, which means it still takes almost a full hour to produce all 26,901 Chu Nom characters in one engraving style.

Figure 5 illustrates some exemplary results after training the CUT models until convergence. The generated images (middle row) combine the content of the printed source images (top row) with the style of the target domain (bottom row), thus allowing us to recreate the entire printed font with 26,901 Chu Nom characters in the 21 engraving styles.

We notice that the visual features of the generated characters are not perfect, e.g. some strokes are missing, making it difficult or even impossible to read

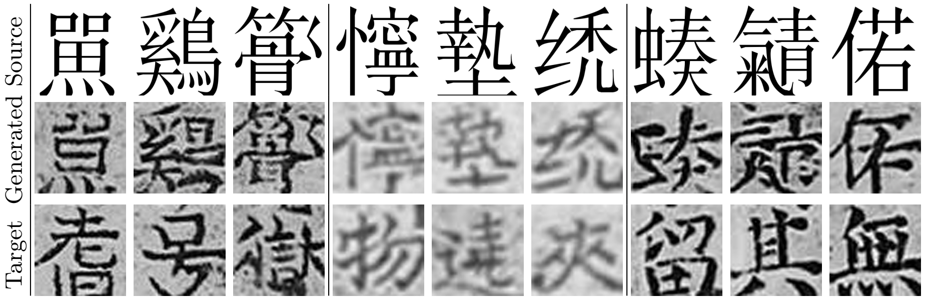


Figure 5: Generated CUT results of three different styles

some of the characters. However, on the stele images similar problems are encountered with real characters, which are not well readable due to low resolution for small characters, damages in the stone, etc. Therefore, we would expect that using the generated characters for training downstream tasks, such as character detection, keyword spotting, and transcription, may still be helpful, especially when combined with real images.

5.4 User study

In order to evaluate the quality of our generated character images, we conducted a user study with 14 participants. Due to the superior quality of the CUT results, we exclusively used those generated character images for this study. A survey was handed out to the participants either as a hard copy or in electronic form as a PDF. In total, there were 14, mostly male, participants of ages ranging from 23 to 59 years with various occupations. We also provided some example images of steles and real character images to at least give the participants a rough idea how the actual steles and characters look like, because none of them had prior experience with ancient Vietnamese stone engravings (nor with Asian logograms in general). We presented each participant with the same grid of 54 characters images, which consists of equal parts real and synthetic images but this information was withhold from the participants. The 27 real character images stem from the 21 steles used as basis for the generated styles. At least one character and at most two were selected from each stele. Similarly, 27 generated images were also chosen from a small random subset (32 images) of each generated style. Each participant was then asked to mark every single image they thought was synthetically generated.

To evaluate the results of the user study, we compute different metrics, which are shown in Table 1. When compared with a random selection process, i.e. an expected recall and precision of 0.5, the participants found significantly less synthetic images (average recall of 0.333). However, among the images marked as synthetic, the participants performed slightly better than random (average precision of 0.582) hinting at the possibility that there might be some visual artefacts in the synthetic samples that can be identified.

Table 1: Quantitative evaluation of the user study. Average metrics for the detection of synthetic characters.

Recall	Precision	F1 Score
0.333	0.582	0.413

We have also evaluated the qualitative feedback that was given by the participants. The task of distinguishing real from fake character images was perceived as difficult by every single one of them. Some of the most frequent reasons are as follows: The amount of different and diverse styles made it hard to identify a synthetic character of a particular style. Also, the participants were unable to detect any obvious patterns, i.e. they were unable to find characteristics of fake images which would have given them away. Individual character images are qualitatively consistent and do not contain visible artefacts or impeccable areas. Additionally, some characters are unrecognisable, incomplete or there does not exist a clear distinction between symbol and background, which adds to the difficulty as well. Furthermore, individual character images are rather small and not of very high-resolution quality. Finally, the lack of familiarity with logograms, not to mention Chu Nom symbols, also made the task more complicated. To illustrate the difficulty of the task, Figure 6 shows a subset of 27 images from the survey.

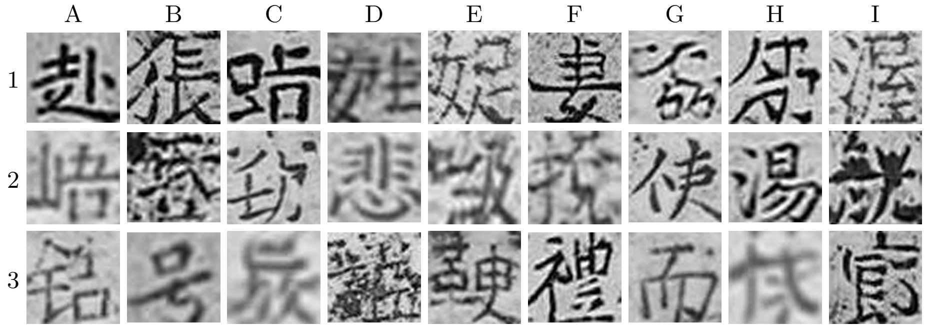


Figure 6: Selection of real and artificially generated character images. Try to identify the synthetic samples[7].

The bottom line is that the images are indeed visually convincing and it is not at all evident which ones are real and which ones were generated. There is not a single style which shows obvious signs that those particular images were computer-generated. Clearly, not every single image of the $21 * 26,901$ generated

[7] Synthetic: 1B, 1C, 1E, 1G, 1H, 1I, 2A, 2C, 2E, 2F, 2I, 3A, 3C, 3E, 3H and 3I

ones was inspected and evaluated in detail. Thus, there is the possibility that certain character images of some styles might display clear indications that they are not real. Nonetheless, from the images we have looked at so far, we can confidently say that the CUT model was successful in imitating styled stele characters. This statement is supported by the fact that, on average, two-thirds of generated images were missed and not a single participant found more than half of the generated characters. This indicates that a very good approximation of the real images has been achieved. Also, the qualitative feedback from the participants clearly emphasises that the generated characters seem genuine to the human eye.

6 Conclusion and Future Work

Due to a lack of annotated data, we explored the possibility of generating synthetic Chu Nom engravings with different styles to make a significant step towards automated reading. A lot of generative models for character synthesis use paired data, i.e. character images that are annotated with their correct character label, which are not available in our case. Therefore, we have investigated two models that do not require paired data: CycleGAN and CUT. With CycleGAN, we rapidly encountered mode collapse problems that produced unreadable results. With CUT, however, we succeeded in the generation of readable Chu Nom engravings, although some of the generated characters were missing important strokes. In a user study, it was demonstrated that the generated characters seemed realistic to an inexperienced human observer.

Although our method does not require human experts to annotate individual character images, a brief manual interaction was still necessary to choose 21 well-distinguishable engraving styles. In future work, we aim to automate this step as well, to cover a larger number of different styles present in the entire stele dataset.

Furthermore, an interesting line of future research would be to synthesize whole stele images for training and improving the annotation-free character detection method. Another line of research would be to use the synthetic characters for training keyword spotting and transcription systems.

References

1. Cai, J., Peng, L., Tang, Y., Liu, C., Li, P.: TH-GAN: Generative adversarial network based transfer learning for historical Chinese character recognition. In: Int. Conf. on Document Analysis and Recognition (ICDAR). pp. 178–183 (2019)
2. Cha, J., Chun, S., Lee, G., Lee, B., Kim, S., Lee, H.: Few-shot compositional font generation with dual memory. In: European Conf. on Computer Vision (ECCV) (2020)
3. Gan, J., Wang, W.: HiGAN: Handwriting imitation conditioned on arbitrary-length texts and disentangled styles. AAAI Conf. on Artificial Intelligence **35**(9), 7484–7492 (2021)

4. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Int. Conf. on Neural Information Processing Systems (NIPS). pp. 2672–2680 (2014)
5. Guan, M., Ding, H., Chen, K., Huo, Q.: Improving handwritten OCR with augmented text line images synthesized from online handwriting samples by style-conditioned GAN. In: Int. Conf. on Frontiers in Handwriting Recognition (ICFHR). pp. 151–156 (2020)
6. Gui, J., Sun, Z., Wen, Y., Tao, D., Ye, J.: A review on generative adversarial networks: Algorithms, theory, and applications. IEEE Trans. on Knowledge and Data Engineering pp. 1–20 (2021)
7. Hayashi, H., Abe, K., Uchida, S.: GlyphGAN: Style-consistent font generation based on generative adversarial networks. Knowledge-Based Systems **186**, 1–13 (2019)
8. Hong, Y., Hwang, U., Yoo, J., Yoon, S.: How generative adversarial networks and their variants work. ACM Computing Surveys **52**(1), 1–43 (2019)
9. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 5967–5976 (2017)
10. Jiang, Y., Lian, Z., Tang, Y., Xiao, J.: SCFont: Structure-guided Chinese font generation via deep stacked networks. AAAI Conf. on Artificial Intelligence **33**(01), 4015–4022 (2019)
11. Kang, L., Riba, P., Wang, Y., Rusiñol, M., Fornés, A., Villegas, M.: GANwriting: content-conditioned generation of styled handwritten word images. In: European Conf. on Computer Vision (ECCV). pp. 273—289 (2020)
12. Lin, X., Li, J., Zeng, H., Ji, R.: Font generation based on least squares conditional generative adversarial nets. Multimedia Tools and Applications **78**(1), 783–797 (2018)
13. Liu, J., Gu, C., Wang, J., Youn, G., Kim, J.U.: Multi-scale multi-class conditional generative adversarial network for handwritten character generation. The Journal of Supercomputing **75**(4), 1922–1940 (2017)
14. Liu, X., Meng, G., Xiang, S., Pan, C.: FontGAN: A unified generative framework for Chinese character stylization and de-stylization. CoRR abs/1910.12604 (2019)
15. Nishat, Z.K., Shopon, M.: Synthetic class specific Bangla handwritten character generation using conditional generative adversarial networks. In: Int. Conf. on Bangla Speech and Language Processing (ICBSLP). pp. 1–5 (2019)
16. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. CoRR abs/1807.03748 (2019)
17. Papin, P.: Aperçu sur le programme ≪ publication de l'inventaire et du corpus complet des inscriptions sur stèles du viêt-nam ≫. Bulletin de l'Ecole française d'Extrême-Orient **90**(1), 465–472 (2003)
18. Papin, P.: Les inscriptions anciennes du viêt-nam, source d'une nouvelle vision des xviie et xviiie siêcles. Good Morning **105** (2010)
19. Park, T., Efros, A.A., Zhang, R., Zhu, J.Y.: Contrastive learning for unpaired image-to-image translation. In: European Conf. on Computer Vision (ECCV). pp. 319–345 (2020)
20. Qin, M., Chen, X.: Restore the incomplete calligraphy based on style transfer. In: Chinese Control Conference (CCC). pp. 8812–8817 (2019)
21. Scius-Bertrand, A., Voegtlin, L., Alberti, M., Fischer, A., Bui, M.: Layout analysis and text column segmentation for historical vietnamese steles. In: Proc. 5th Int. Workshop on Historical Document Imaging and Processing (HIP). pp. 84–89 (2019)

22. Scius-Bertrand, A., Jungo, M., Wolf, B., Fischer, A., Bui, M.: Annotation-free character detection in historical vietnamese stele images. In: Int. Conf. on Document Analysis and Recognition (ICDAR). pp. 432–447 (2021)
23. Tian, Y.: Master Chinese calligraphy with conditional adversarial networks (2017), https://github.com/kaonashi-tyc/zi2zi
24. Wen, C., Pan, Y., Chang, J., Zhang, Y., Chen, S., Wang, Y., Han, M., Tian, Q.: Handwritten Chinese font generation with collaborative stroke refinement. In: IEEE/CVF Winter Conf. on Applications of Computer Vision (WACV). pp. 3882–3891 (2021)
25. Wu, L., Chen, X., Meng, L., Meng, X.: Multitask adversarial learning for Chinese font style transfer. In: Int. Joint Conf. on Neural Networks (IJCNN). pp. 1–8 (2020)
26. Wu, S.J., Yang, C.Y., Hsu, J.Y.J.: CalliGAN: Style and structure-aware Chinese calligraphy character generator. CoRR abs/2005.12500 (2020)
27. Xi, Y., Yan, G., Hua, J., Zhong, Z.: JointFontGAN: Joint geometry-content GAN for font generation via few-shot learning. ACM Int. Conf. on Multimedia pp. 4309–4317 (2020)
28. Zeng, J., Chen, Q., Liu, Y., Wang, M., Yao, Y.: StrokeGAN: Reducing mode collapse in chinese font generation via stroke encoding. CoRR abs/2012.08687 (2021)
29. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Int. Conf. on Computer Vision (ICCV). pp. 2242–2251 (2017)