Korean Font Synthesis with GANs

Debbie Honghee Ko, Ammar Ul Hassan, Jungjae Suk, and Jaeyoung Choi

***Abstract*—Font synthesis for CJK based languages that consists of large number of characters and complex structures is still a major challenge and ongoing research problem for computer vision and AI. In this paper, we propose a generative model based on GANs as a solution for Korean font synthesis problem with a small set of characters. Korean Hangul includes 11,172 characters and composes of writing in multiple patterns. Normally font design involves heavy loaded human labor that can easily hit to one year to finish for one style set. Various methods have been proposed to solve this character generation problem using generative models such as GANs, but the results are often blurry or broken and are far from realistic. We generate visually appealing Korean Hangul characters with a skeleton-driven approach. We demonstrate that this approach is effective at synthesizing characters from their corresponding skeletons. With 114 samples, the proposed method automatically generates the rest of the characters in the same given font style. Our approach resolves long overdue shortfalls such as blurriness, breaking, and unrealistic shapes and styles of characters using GANs. We demonstrate via our experiments that our approach has better quality then other methods.**

***Index Terms*—GANs, typography, domain translation, style transfer.**

I. Introduction

Traditional fonts are mainly characterized by their components such as thickness, strokes, and serifs. Font designer needs at least months to make a new font set with couple of strokes, serifs, and thickness. Unlike the English font characters where the font designer designs just 26 characters, Korean Hangul comprises of 11,172 characters which can easily take up to one to two years. Furthermore, the complex structures and shapes of Hangul characters make this font designing process more difficult. Font designer needs at least months to make a new font set with couple of font weights and italics.

In the last few years, deep neural networks have emerged to solve the problems like image style transfer, image classification, and image synthesis. Generative models are widely used for image synthesis problems. Generative adversarial networks (GANs) [1] are a class of generative models that aim to model the real images distribution by forcing the generated images to be indistinguishable from the real images using adversarial training.

Image-to-Image translation framework "pix2pix" [2] was proposed based on GANs where the goal is to translate an input image in one domain to an output image in another domain given input-output image pair as training data. Fig. 1 shows the paired dataset concept where one image in Domain^S must have its corresponding image pair in Domain^F.

Fig. 1. Paired dataset example. Each skeleton image in Domain^S has its corresponding font image in Domain^F.

Font synthesis can be modeled as an Image-to-Image translation problem. Where the goal is to translate a font in one domain to another font in another domain. This process can also be referred as style transfer. Various methods have adopted this font domain translation mechanism to generate font styles.

Zi2zi [3] method was proposed as a font to font translation solution for generating variety of font styles. This method was built on pix2pix framework. The key difference between their method and vanilla pix2pix was that they added the idea of category embedding. This addition helped the conditional GAN framework to have control over the style of generated target domain font.

DCFont [4] was proposed after zi2zi which was inspired by the overall method of pix2pix and zi2zi. They used an additional style feature reconstruction network which was used to extract the style of the target domain font. They improved the results of zi2zi on handwritten Chinese characters.

Various approaches have made for special artistic font style transfer by analyzing few samples, using deep CNN [5], cGAN architecture by stacking glyph nets and ornament nets [6], and deep convolutional GAN (DCGAN) [7] by controlling character and style vector independently to generate various glyphs [8]. Another approach was a patch-based style transfer model for special effects like burning flames [6].

In this paper, we discuss a new approach of high-quality Korean Hangul font images from skeletons. The skeleton of a character in our system is composed with thin width structure. We use a python module to generate skeletons of font

Manuscript received March 9, 2020; revised June 1, 2020.

Debbie Honghee Ko, Ammar Ul Hassan, Jungjae Suk, and Jaeyoung Choi are with School of Computer Science and Engineering, Soongsil University (e-mail: debbie.pust@gmail.com, ammar.instantsoft@gmail.com, jjsuk256@gmail.com, choi@ssu.ac.kr).

DOI: 10.7763/IJCTE.2020.V12.1270

characters. We show that through our proposed skeleton-driven approach our system synthesizes photo-realistic, non-blurry results than up-to-date deep generative adversarial models in font synthesis studies.

Our paper is composed in this manner, First, we define the detail method of our approach in Section II. Then in Section I, we perform qualitative and quantitate experiments. In the last section, we give concluding remarks.

II. Method

The GAN framework consists of two neural networks named as Generator (*G*) and Discriminator (*D*). *G* takes a random noise z as an input and tries to generate a fake image. The goal of *G* is to estimate the distribution of the real images without even seeing them. On the other hand, *D* focuses on differentiating between the real images and the fake images generated by *G*. This forces *G* to generate realistic images close to the real images. During the training time *G* and *D* play the minimax game. This noise based image generation using *G* and *D* based neural networks is known as Vanilla GAN. The objective function of Vanilla GAN is given by:

$$min\ max\ V(D,G) = Ex \sim pdata(x)\ [log\ D(x)]$$

$$+\ Ez \sim pz(z)\ [log\ (1\text{-}D(G(z)))], \quad (1)$$

where *pdata(x)* and *pz(z)* are the probability distributions for real images and fake images respectively. *D(x)* is the discriminator output against the real images *(x)* whereas, *D(G(z))* is the output of the discriminator against fake images generated by the generator *G*.

G(z) is the generator function that takes a random noise *z* as an input and generates a fake image. The goal of *D* is to maximize the probability of real images i.e. 1 whereas minimize the probability against fake images i.e. 0. On the other hand, goal of *G* is to force *D* to maximize the probability against fake images i.e. 1 by generating images that are close to the real images distribution.

This Vanilla GAN architecture produces good synthesize images in general but the drawback with this noise z based framework is that we cannot control the generated image. To overcome this output controlling problem of Vanilla GAN, Mirza and Osindero proposed a GAN named as *c*GAN (conditional generative adversarial network) [9].

For our Korean Hangul synthesis problem, we extend "pix2pix" framework which is based on conditional GAN (*c*GAN). *c*GAN simply utilizes an additional conditional information *c* in both the generator and discriminator for controlling the output. *c*GAN objective function is given by:

$$min\ max\ V(D,G) = Ex \sim pdata(x)\ [log\ D(x|c)]$$

$$+Ez\sim pz(z)[log(1\text{-}D(G(z|c)))], \quad (2)$$

where the class information is added to both the networks *D* and *G*, respectively. With this class information *c*, the generated output can be controlled according to the class label. This *c*GAN based framework solves many computer vision problems like Image-to-Image translation, where the source image in one domain is translated into a target domain image in another domain as we described in Section I of this paper. Our model is also based on this I2I framework. In the next section, we describe our network architecture in detail.

A. Method Architectures

Our model consists of conditional adversarial networks; a skeleton-to-font network to generate the whole set of characters as depicted in Fig. 2. First, we extract the font images of the hangul font characters. Then we pass these font images into a python module which generates the corresponding skeletons. After extracting the skeletons, we create pair dataset i.e. each skeleton of a character paired with the corresponding characters font image. The skeletons generated from python module are fed to skeleton to font network in order to learn the target domain font style by down-sampling and up-sampling with the aids of adversarial knowledge. The whole process of our network is defined in Fig. 2.

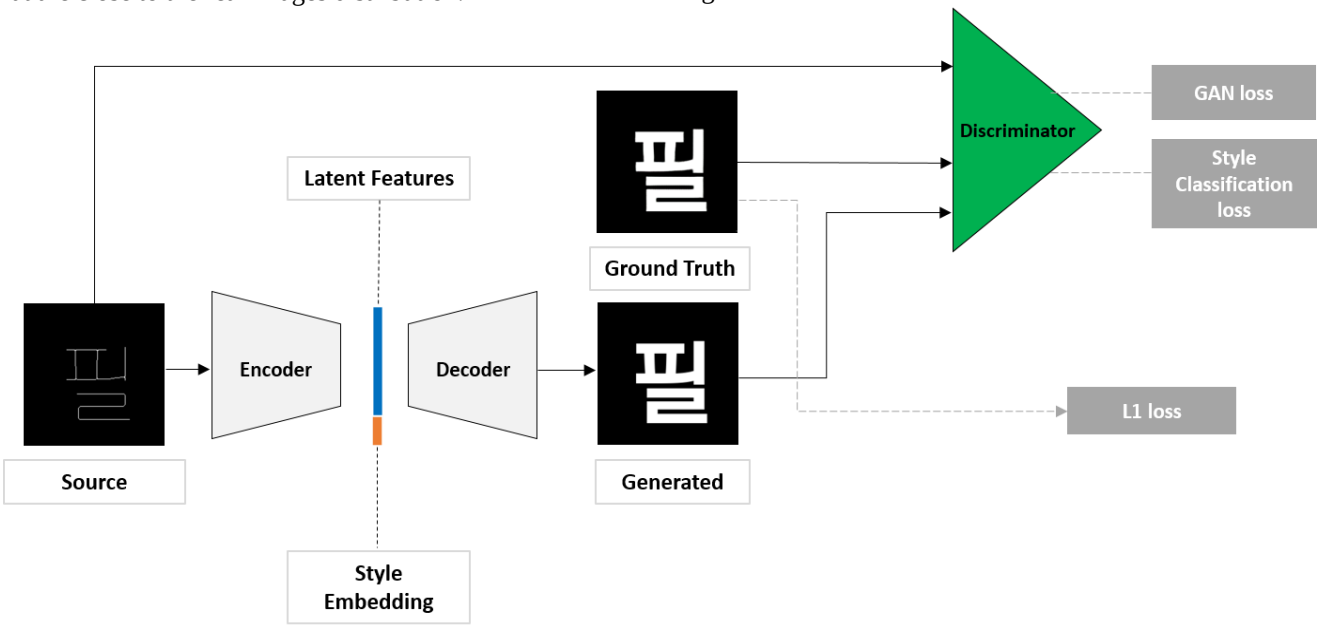

Fig. 2. Architecture of our proposed method. Our encoder takes a skeleton image as an input and passes it through the encoder which generates content embedding. This content embedding is then combined with style embedding and then passed to the decoder which up-samples the image and synthesizes the target font style. Discriminator then discriminates between real and fake image.

We employed PatchGAN [2] as our model's discriminator. Traditional GANs use a normal classifier as a discriminator that classifies whether an input image is real or fake by down-sampling the input to a single scalar output (1 or 0). Whereas, PatchGAN based discriminator outputs a multi-dimensional vector, where each point of the vector corresponds to a *N*N* patch of the input image. The discriminator job is to classify whether the given patch in the input image is real or fake. Different from the original PatchGAN discriminator, we added a fully connected classification layer in the end of it.

B. Loss Function

We used three loss functions. We define each loss here and how the weights are trained from these loss functions. Later we formulate our total loss function.

One is the GAN loss (LosscGAN) for both generator and discriminator. The Generator goal is to generate the font samples such that the Discriminator gives a 1 label i.e. real font image prediction to it. Whereas, the Discriminator aims to give real image label 1 to the real font images and 0 label to the fake font images generated by the Generator.

We introduce style embedding loss (LossSE) in our system that can handle multiple font styles at the same time in our networks as vanilla pix2pix cannot handle this. *D* (Discriminator) tells not only real/fake, but also those two pair of images are in the same font style or not (Dse, (*D*'s style embedding)).

The final reconstruction loss L_1 (LossL_1) is to minimize the distance between *GT y* and the generated output from *G(x,z)*. *L*1 encourages less blurring than L_2.

Therefore, our final objective is

$$G^* = arg\ min\ max\ LosscGAN(G,D) + LossSE(G,D) + LossL_1, \quad (3)$$

where it tries to minimize the objective for G against an adversary, minimizing pixel to pixel distances between GT and the generated.

III. Experimental Results

To evaluate the ability of our proposed method, we performed some font generation experiments. We evaluated our generated font images based on the following two points:

Style consistency: We qualitatively verify that the font generated by our proposed S2F based approach has style consistency via visual observation.

Character content accuracy: We quantitatively verify that the generated font images have high character classification accuracy via a Hangul character recognition experiment where we used a CNN model trained on real Hangul images. We also computed the l1 loss between the target images and the generated images.

A. Training and Testing Dataset

For our training dataset, we used 2,350 most commonly used Korean Hangul characters from 15 font files. We used a python based module that takes a font character as an input and generates its corresponding skeleton. Both the input skeleton image and the target font image in our system is RGB with a size of 256*256*3 (3-channels for RGB). In order to make the model robust during the pre-training process, we learn a one-to-one mapping function i.e. 2,350 skeletons for each font style to the corresponding target domain font images. This one-to-one learning during the pre-training ensures that the network learns diverse font style representations and structures.

For testing our model, we used several different Korean font styles. These font styles were chosen based on the overall style and structure of characters in reference and target domains. For learning any new font style, we fine tune our network with a small set of characters (114). The reason for selecting these specific 114 characters was that they represent the overall structure of all other Korean Hangul characters. Our network learns the new font style by observing 114 characters and during the testing time the proposed network can generate the rest of 2,236 characters in the newly learnt style (this can easily be extended to generate 11,172 characters).

B. Style Consistency

To evaluate the style consistency of our network generated font images, we performed the qualitative evaluation by visualizing the generated font images using the proposed S2F approach against zi2zi, pix2pix, and compared against the ground truth images. For fair comparison, we pre-trained all four networks with the same dataset and finetuned all with several unseen font styles as discussed above.

As shown in the Fig. 3, our method produces non-blurry photo realistic typefaces in one style compared with the other methods. More qualitative results are shown in Fig. 4 and Fig. 5 below in two different styles. When the synthesized images are zoomed in, poor smoothness, breaking of strokes, and blurriness are often found in the other methods.

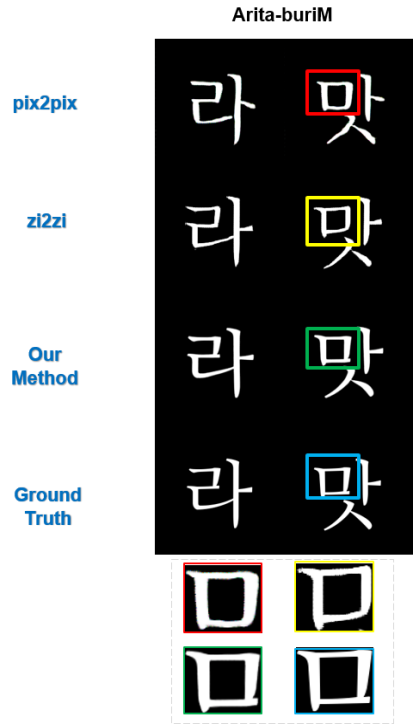

Fig. 3. Our method comparison with pix2pix. zi2zi and ground truth on three different styles and various characters.

Fig. 4. Our method detailed comparison with other methods in style 1.

Fig. 5. Our method detailed comparison with other methods in style 2.

TABLE I: QUALITATIVE COMPARISON

Font Styles	Pix2pix			Zi2zi			SDKorFont		
	HCCS	L1 Loss	L2 Loss	HCCS	L1 Loss	L2 Loss	HCCS	L1 Loss	L2 Loss
Arita-buriM	0.9603	0.2635	0.2529	0.9779	0.2661	0.2521	**0.9955**	**0.2304**	**0.2275**
BinggraeTaomB	0.9158	0.3347	0.2632	0.9236	0.2612	0.2475	**0.9234**	**0.2496**	**0.2439**
DXMyeongjo	0.9451	0.2682	0.2564	0.9583	0.2720	0.2553	**0.9789**	**0.2549**	**0.2515**

C. Character Content Accuracy

We also did a quantitative evaluation for the proposed model and the other two methods as shown in Table I. We computed L1 losses between the synthesized image and the ground truth image. Additionally, we also computed the HCCS (Hangul Character Classification Score). For HCCS, we pretrained a CNN model to correctly classify the Hangul characters. At inference time, we calculated the HCCS by predicting the characters label from our model and other two methods.

IV. CONCLUSION

In this paper, we propose a new approach of high-quality Korean Hangul font images by learning a mapping function from S2F (Skeletons2Font) instead of traditional F2F mapping function. Our network takes a style vector and structure vector as an input to the generator, thereby allowing hangul font generation with style consistency and photo realistic synthesis. This model markedly improves the traditional problems in font synthesis models, such as blurriness, severe artifacts, non-photo realistic results. In our future work, we will focus on generating font images without any fixed reference skeleton images.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Debbie Ko conducted the research and wrote the paper; Ammar Ul Hassan implemented this system and wrote the paper together; Jungjae conducted all the experiments and drew figures and tables; Jaeyoung Choi supervised this research and he is the corresponding author; all authors had approved the final version.

ACKNOWLEDGMENT

This work was supported by Institute of Information & communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIP) (No. 2016-0-00166, Technology Development Project for Information, Communication, and Broadcast)

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, Bi. Xu, D. Warde-Farley, S Ozair, A. Courville, and Y. Bengio, *Generative Adversarial Nets*, NIPS, 2014.
[2] P. Isola, J. Zhu, T. Zhou, and A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. CVPR*, 2017.
[3] Y. Tian. (2016). zi2zi: Master Chinese calligraphy with conditional adversarial networks. [Online]. Available: https://github.com/kaonashi-tyc/zi2zi
[4] Y. Jiang, Z. Lian, Y. Jianguo, and J. Xiao, "DCFont: An end-to-end deep Chinese font generation system," *SIGGRAPH Asia*, p. 22, 2017.
[5] S. Baluja, "Learning typographic style," *Google Research*, 2016.
[6] S. Azadi, M. Fisher, V. Kim, Z. Wang, E. Shechtman, and T. Darrell, "Multi-content GAN for few-shot style transfer," in *Proc. CVPR*, 2018, pp. 7564-7573.

[7] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. ICLR 2016*, 2016.
[8] H. Hayashi, K. Abe, and S. Uchida, "GlyphGAN: Style-consistent Font generation based on generative adversarial networks," *Knowledge-Based Systems*, vol. 186, December 2019.
[9] M. Mirza and S. Osindero, "Conditional generative adversarial nets," arXiv preprint arXiv: 1411.1784, 2014.

Debbie Honghee Ko received the B.S from Physics Department, Korea University in 1983, Seoul, South Korea. She then received her M.S. degree from Department of Computer Science, Florida State University in 1989, Florida, U. S.

She worked at the FL Department of Education for 10 years as a supervisor. She is currently taking her Ph.D. degree in Department of Computer Science, Soongsil University, Seoul, South Korea. Her current research area is deep learning on font synthesis using generative models.

Ammar Ul Hassan received the B.S. degree from Department of Software Engineering, International Islamic University Islamabad, Pakistan in 2013. He then received his M.S. degree in computer science from Soongsil University, Seoul, South Korea in 2018.

He is currently taking his Ph.D. degree in Department of Computer Science, Soongsil University Seoul, South Korea. He is working as a research associate in System Software Laboratory. His current research area is about deep learning, computer vision, generative models, making font environment for new fonts in Linux operating system.

Jungjae Suk received the B.S. degree from Department of Web Information Engineering, Hankyong National University, Angseong, Gueonggi, Korea, in 2017.

He is currently taking his M.S. degree in Department of Computer Science and Engineering, Soongsil University, Seoul, Korea. He is working as a research associate in System Software Laboratory. His current research area is about deep learning and typography.

Jaeyoung Choi received the B.S. degree from Department of Control and Instrumentational Engineering, Seoul National University, Seoul, Korea, in 1984, and the M.S. degree from Department of Electrical Engineering, University of Southern California in 1986, and the Ph.D. degree from School of Electrical Engineering, Cornell University in 1991.

He has previously worked at Oak Ridge National Laboratory (1992-1994) and University of Tennessee, Knoxville (1994-1995) as a postdoctoral research associate and a research assistant professor, respectively, where he had been involved with the ScaLAPACK project. He is currently a professor of School of Computer Science and Engineering, Soongsil University, Seoul, Korea. His current research interests include high performance computing and typography.