GANdan-fontmaker: Web Service for Handwritten-Hangul Font Generation

Seongbin Yoon[1], Deukyun Nam[2], Dasol Lee[3], and Sungeun Wee[3]

[1] Sungkyunkwan University, Electronic and Electrical Engineering
[2] Sungkyunkwan University, Mathematics
[3] Sungkyunkwan University, Systems Management

Abstract. Hangul is a language that is composed of initial, medial, and final consonant. This leads to 11,172 combinations of characters in Hangul. For this reason, the current method of designing all the characters by hand is very expensive and time-consuming compared to English. We propose web-service(GANdan-fontmaker) based on Dual Memory-Augmented Font Generation Network(DM-font). Our service will provide two distinctive functions; combining people's font style And re-correcting font style for satisfying client's need.

Keywords: GAN · Font Generation · Web Service

1 Introduction

As technology advances, the era of single-person media has gradually come. Showing and using 'letter' in the media is as important as the media itself. In this situation, the copyright of the font bothers personal media. The only way to use fonts while avoiding copyright infringement is to use commercially available free fonts or to produce own fonts. However, unlike English, Hangul fonts have a remarkably small number of commercially available free fonts, and the process of making fonts is a long and expensive task. Various trials based on Style Transfer[1] and Generative Adversarial Network(GAN)[2] have been conducted to manage the task. GAN is appropriate to generate customized fonts through few inputs, while style transfer generates fonts using whole character sets. In this project, we propose the service based on GAN font generation technique(DM-font)[3] which requires small sample of user input.

2 Objective

First, reduce the cost and simplify the process of making font in Korean by applying font generation model, DM-font.

Second, Develop web-service and provide functions such as creating own font design and transforming it in .ttf file.

Third, complement and improve the output of font design to increase user satisfaction by applying Ensemble.

Fourth, increase user interest by providing a function which enables font production with combining inputs of several users.

3 Related Work

3.1 Font generation models

3.1.1 Font generation using CNN based Style Transfer

Tian, Y. (2016 [4]) proposed neural style transfer for Chinese characters, "Rewrite". The network is top-down CNN structure where font design process is formulated as a style transfer from a standard font to a target font. The network but only manages to learn one style at a time, also lots of characters (approximately 2000) are require learning style.

Neural style transfer uses CNN for image style transfer. By modifying neural style transfer, Atarsaikhan et al.(2017 [5]) proposed neural font style transfer. This method transfers the style of one font image to another using the style features extracted from the intermediate layers of the CNN. In this approach, the images of various font styles of different languages, such as Arabic, Japanese, and Korean, are used to generate the font images.

3.1.2 GAN based font generation

As generative model is advanced, lots of font generation studies are conducted using GAN framework. Hayashi et al.(2019 [6]) proposed GlyphGAN. It creates new fonts using a deep convolutional GAN (DCGAN) while maintaining style consistency over all characters. In GlyphGAN, the input vector is composed with character class vector, which is one-hot vector, and style vector, which is a uniform random vector. However, The problem is that the style of the synthesized characters cannot be controlled based on the user's preference, as the style vector is always random.

Zi2zi was proposed for generating handwritten Chinese characters[7] to solve one-to-one mapping problem on pix2pix[8]. Zi2zi adds one-hot category embedding for one-to-many modelling and includes AC-GAN architecture and a domain transfer network (DTN) to reduce category losses in multi-class. Though zi2zi can generate lots of letters simultaneously, it has a problem that the quality of the letters is poor, and the volume of learning data is too vast. To improve the model, Jiang et al.(2017 [9]) proposed DCFont: an end-to-end deep Chinese font generation system. The DCFont network consists of two networks: a network that reconstructs font features with a pretrained VGGNet for extracting a font style and a network that transmits font styles.

3.1.3 Font Generation in Korean

Ko et al.[12] proposed SKFont: skeleton-driven Korean font generator with conditional deep adversarial networks. When given 114 target characters, the system automatically generates the rest of the characters in the same given font style by extracting the skeletons of the synthesized characters. It improves the problem of blurriness, breaking, and a lack of sophistication by using the skeleton-driven approach.

Cha et al.[3] proposed Dual Memory-Augmented Font Generation Network (DM-Font), which enables to generate a high-quality font library with only a few samples. DM-Font utilized the compositionality of Korean scripts with 68 components labels. Also, it employs dual memory structure to efficiently capture the global glyph structure and the local component-wise styles.

3.2 Ensemble Learning

3.2.1 AdaBoost AdaBoost is a statistical classification meta-algorithm.[10] It can be used in conjunction with many other types of learning algorithms to improve performance. The idea of AdaBoost is to generate a strong model by combining weak models which are incrementally learned with focus on difficult cases.

3.2.2 AdaGAN AdaGAN is a kind of AdaBoost.[11] AdaGAN is a meta-algorithm to construct a "strong" mixture of GANs, trained sequentially. It deals with missing mode problem, where the model is not able to produce examples in certain regions of the space, by Boosting. In GAN, the missing mode problem often arises in such a way that some instance types cannot be generated. AdaGAN proposes a way of resolving this issue with specialized way of updating train weights and forming mixture of weak generators.

3.3 Font generate service

3.3.1 Naver "나눔손글씨" event In 2019, on the occasion of Hangul Day, Naver produced user handwriting using Clova's artificial intelligence technology and distributed 109 letters for free. Naver Clova's OCR team applied Korean handwriting font generation technology to the service to improve the recognition rate of OCR handwriting. There are the stages of recognizing handwriting in the image, analyzing and learning the characteristics of handwriting, upgrading fonts based on pre-learning models, and finally completing handwriting fonts. What user is required in this process is the 256 handwriting and 30 minutes of waiting. After the contest, the OCR team proposed a DM-font that generates a higher quality font with fewer letters. They taught the model by dividing the initial, medial, and final consonant from one letter, and understand the combination of Hangul to create the word "감" from "거" + "아" + "흠". As a result, handwriting can be created in 20 seconds even after receiving only 28 characters from the user.

3.3.2 Fontto The SW Maestro Twiiiks team developed Fontto with the aim of allowing the general public to create their own unique fonts and to sell and share them. When a user writes a given sentence up to five sentences, the remaining thousands of letters are automatically generated through machine learning technology. The Fontto service was created using three technologies: OCR, Variational AutoEncoder (VAE), and GAN. Based on the letters recognized as OCR and the general form of them configured through VAE, the user's font will be created using GAN technology. This service restored the handwriting of righteous person such as Ahn Jung Geun.

4 Proposed Solution

4.1 Overall Architecture

Web pages will be created using React, and servers that exchange information with web pages will be created using Node Js. We will make another server that communicates with Node Js and run PyTorch to perform deep learning and it will be made using Flask. DataBase, which stores user's information and generated font, is accessible from Node JS and Flask server.

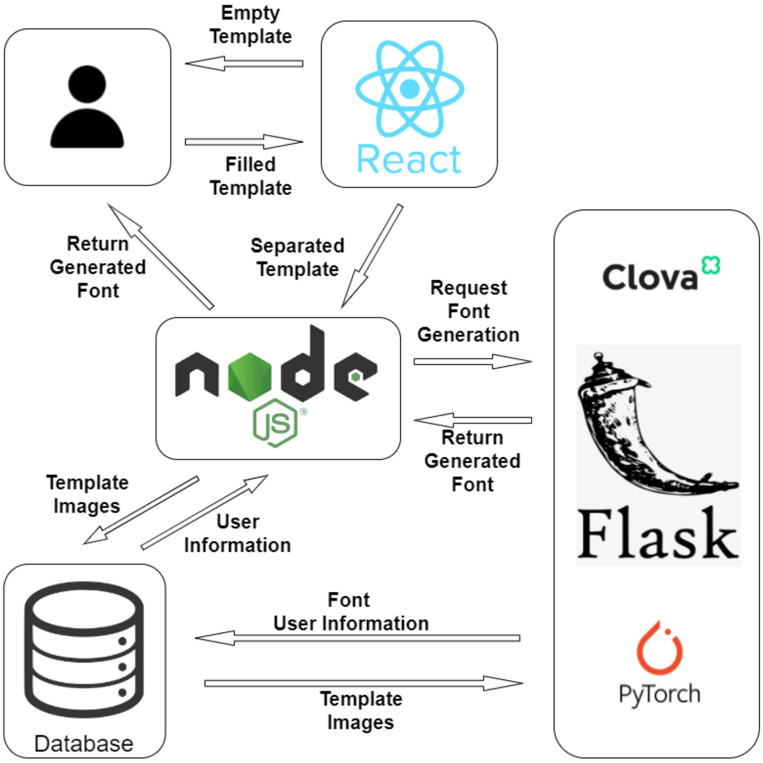


Fig. 1. Overall architecture

4.2 Servers

4.2.1 Web Web services largely provide two functions. First function is sending the generated font based users' font to the user. Second is receiving the user's feedback, modifying the generated font and sending it again.

The user accesses the web page. Users can download templates from the homepage, fill user informaton and templates and post them on the web page, and the information is delivered to the Node Js server. Node Js server sends the information to DataBase, and requests deep learning to Flask server. After receiving the request, the Flask server accesses DataBase, takes the modified template, executes the model, generates user font-style information, sends the generated font to DataBase, and sends a completion request to Node Js. Node Js server accesses DataBase, obtains the genertaed font, and delivers it to the user's email.

After the user sees the result delivered to the email, the user accesses the web page to enter the user's information and the web page sends the information to the Node Js server. The Node Js server checks the information and stored information in DataBase and transmits whether they match or not to the web page. If the information matches, the user can post feedback on the web page, and the web page sends the information to the Node Js server. The Node Js server transmits the user's feedback to DataBase and requests feedback deep learning to the Flask server. The Flask server accesses DataBase, takes the previously generated font and user feedback, executes deep learning again, sends the modified font to DataBase, and sends a completion request to Node Js. Node Js server accesses DataBase, takes the modified result, and delivers it to the user's email.

4.2.2 Applying DM-Font

In case of, single user - First generation, we will apply naive DM-Font. However, to handle several users or font revise request case, we should focus on how the actual weights for each 68 components in initial, neutral and final works especially in decoding(generation) phase.

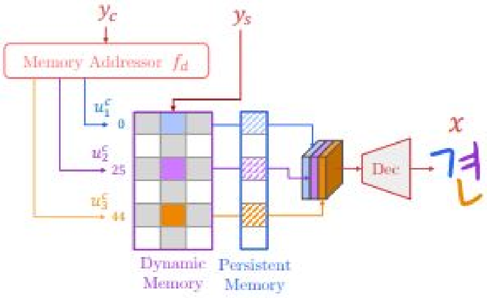
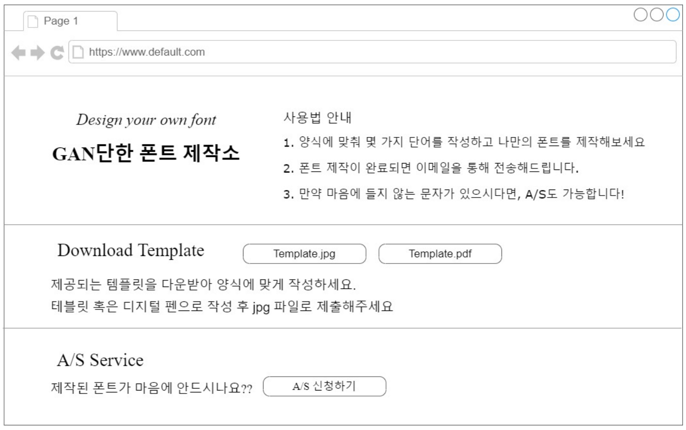


Fig. 2. DM-font Decoding Phase

4.3 Web design outline

4.3.1 main page(index.html)



Page 1

https://www.default.com

Design your own font

GAN단한 폰트 제작소

사용법 안내

1. 양식에 맞춰 몇 가지 단어를 작성하고 나만의 폰트를 제작해보세요

2. 폰트 제작이 완료되면 이메일을 통해 전송해드립니다.

3. 만약 마음에 들지 않는 문자가 있으시다면, A/S도 가능합니다!

Download Template

Template.jpg Template.pdf

제공되는 템플릿을 다운받아 양식에 맞게 작성하세요.
테블릿 혹은 디지털 펜으로 작성 후 jpg 파일로 제출해주세요

A/S Service

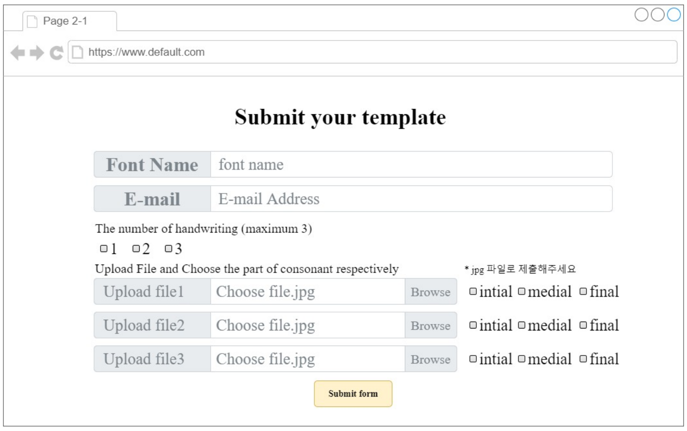
제작된 폰트가 마음에 안드시나요?? A/S 신청하기

Fig. 3. index.html outline

function of main page is as below.

- How to use
- Download font template for user handwritten input
- Get to form-one.html
- Get to form-two.html

4.3.2 Font generation request page(form-one.html)



Page 2-1

https://www.default.com

Submit your template

Font Name	font name
E-mail	E-mail Address

The number of handwriting (maximum 3)

□1 □2 □3

Upload File and Choose the part of consonant respectively

* jpg 파일로 제출해주세요

Upload file1	Choose file.jpg	Browse	□intial □medial □final
Upload file2	Choose file.jpg	Browse	□intial □medial □final
Upload file3	Choose file.jpg	Browse	□intial □medial □final

Submit form

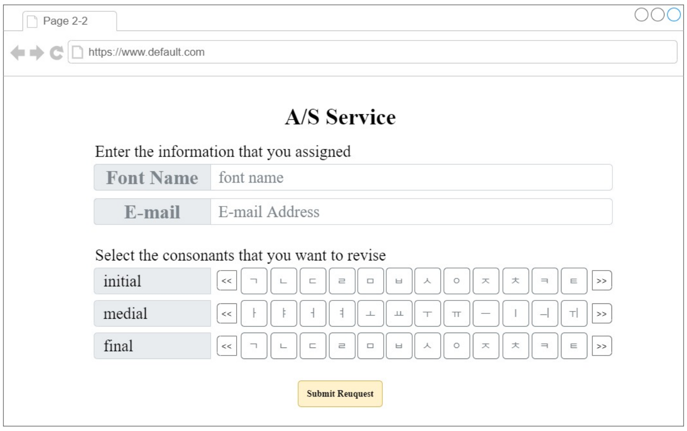
Fig. 4. form-one.html

Form contents of font generation request page is as below.

- font name
- e-mail
- templates

After it post the form, page redirects to request-done.html User can upload at most three templates at one time. User should select the position in Korean letter(Cho, Jung and Jong) for each templates.

4.3.3 Font revise request page(form-two.html)



A/S Service

Enter the information that you assigned

Font Name	font name
E-mail	E-mail Address

Select the consonants that you want to revise

initial	<<	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㅌ	>>
medial	<<	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅣ	ㅢ	ㅟ	>>
final	<<	ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㅌ	>>

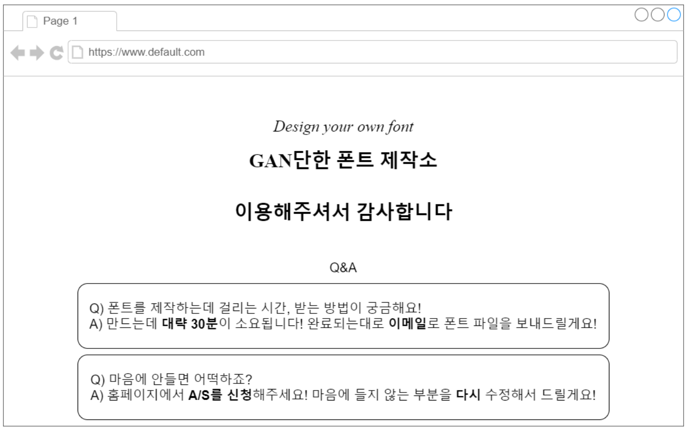
Fig. 5. form-two.html

Font revise request page ask for

- font name
- e-mail

if there exist such font generation request, then redirects to request-done.html else redirects to main.html.

4.3.4 request done page (request-done.html)



Design your own font

GAN단한 폰트 제작소

이용해주셔서 감사합니다

Q&A

Q) 폰트를 제작하는데 걸리는 시간, 받는 방법이 궁금해요!
A) 만드는데 **대략 30분**이 소요됩니다! 완료되는대로 **이메일**로 폰트 파일을 보내드릴게요!

Q) 마음에 안들면 어떡하죠?
A) 홈페이지에서 **A/S를 신청**해주세요! 마음에 들지 않는 부분을 **다시** 수정해서 드릴게요!

Fig. 6. request-done.html

request done page shows that users request on font generation or revise had been successfully pased to server.

5 Planning in Detail

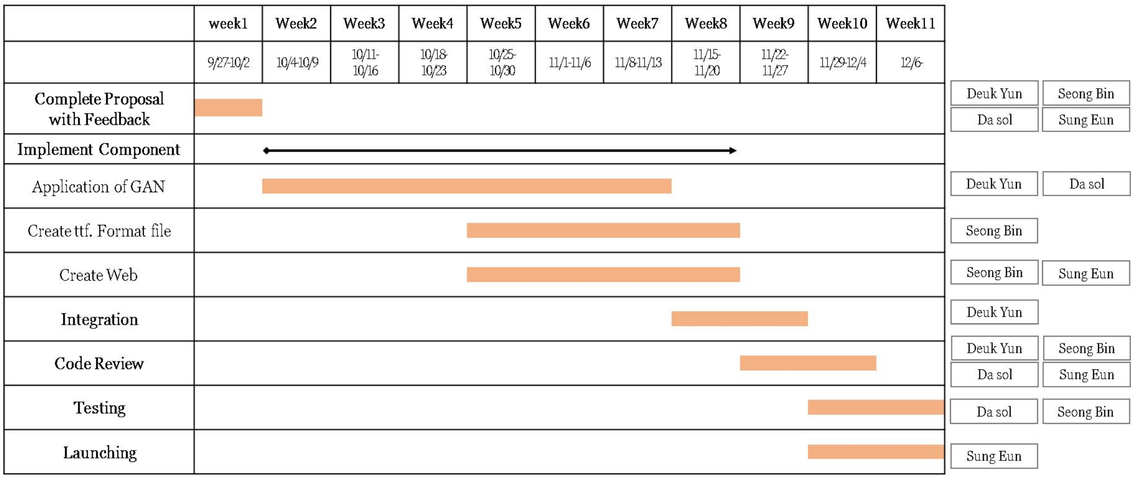


Fig. 7. Project plan timeline

We divide works in three parts.

- Application of DM-Font
- Create module for .png to .ttf (or .hdf5) for DM-font input.
- Implement web-server/client.

We divide the roles as the name on the each works. But we will manage in fluidly and everyone will contribute to all parts. We plan to finish implementation until Week8. After implementation, we will do test and refactoring. Finally we plan to launch the service on week11.

References

1. Gantugs Atarsaikhan, et al. ”Neural Font Style Transfer.” International Conference on Document Analysis and Recognition (IAPR) (2017).
2. A. Odena, C. Olah, and J. Shlens, “Conditional Image Synthesis with Auxiliary Classifier GANs,” Proceeding of the 34th International Conference on Machine Learning, ArXiv Preprint arXiv:1610.09585, 2017.
3. Cha, Junbum, et al. ”Few-shot compositional font generation with dual memory.” Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16. Springer International Publishing, 2020.
4. Tian, Yuchen. ”Rewrite: Neural style transfer for chinese fonts, 2016.” Retrieved Nov 23 (2016): 2016.
5. Atarsaikhan, Gantugs, et al. ”Neural font style transfer.” 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). Vol. 5. IEEE, 2017.
6. Hayashi, Hideaki, Kohtaro Abe, and Seiichi Uchida. ”GlyphGAN: Style-consistent font generation based on generative adversarial networks.” Knowledge-Based Systems 186 (2019): 104927.
7. Tian, Yuchen. ”zi2zi: Master chinese calligraphy with conditional adversarial networks.” Internet] https://github. com/kaonashi-tyc/zi2zi (2017).

8. Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
9. Jiang, Yue, et al. "Dcfont: an end-to-end deep chinese font generation system." SIGGRAPH Asia 2017 Technical Briefs. 2017. 1-4.
10. Freund, Yoav, Robert Schapire, and Naoki Abe. "A short introduction to boosting." Journal-Japanese Society For Artificial Intelligence 14.771-780 (1999): 1612.
11. Tolstikhin, Ilya, et al. "Adagan: Boosting generative models." arXiv preprint arXiv:1701.02386 (2017).
12. Ko, Debbie Honghee, et al. "SKFont: skeleton-driven Korean font generator with conditional deep adversarial networks." International Journal on Document Analysis and Recognition (IJDAR) (2021): 1-13.