

# Hangul Font Dataset for Korean Font Research Based on Deep Learning

Debbie Honghee Ko<sup>†</sup> · Hyunsoo Lee<sup>††</sup> · Jungjae Suk<sup>†††</sup> · Ammar UI Hassan<sup>††††</sup> · Jaeyoung Choi<sup>†††††</sup>

## ABSTRACT

Recently, as interest in deep learning has increased, many researches in various fields using deep learning techniques have been conducted. Studies on automatic generation of fonts using deep learning-based generation models are limited to several languages such as Roman or Chinese characters. Generating Korean font is a very time-consuming and expensive task, and can be easily created using deep learning. For research on generating Korean fonts, it is important to prepare a Korean font dataset from the viewpoint of process automation in order to keep pace with deep learning-based generation models. In this paper, we propose a Korean font dataset for deep learning-based Korean font research and describe a method of constructing the dataset. Based on the Korean font data set proposed in this paper, we show the usefulness of the proposed dataset configuration through the process of applying it to a deep learning Korean font generation application.

Keywords : Deep Learning, Font Data, Automatic Font Generation, Hangul Font Dataset, Hangul Font

## 딥러닝 기반의 한글 폰트 연구를 위한 한글 폰트 데이터셋

도 흥 희<sup>†</sup> · 이 현 수<sup>††</sup> · 석 정 재<sup>†††</sup> · Ammar UI Hassan<sup>††††</sup> · 최 재 영<sup>†††††</sup>

## 요 약

최근 딥러닝에 대한 관심이 증가하면서 이를 이용한 다양한 분야에서 연구가 진행되고 있다. 그러나 딥러닝 기반의 생성 모델을 이용하는 폰트의 자동 생성 연구들은 로마자 및 한자와 같은 몇 언어들에 국한되어 연구되고 있다. 한글 폰트 디자인은 매우 큰 시간과 비용이 들어가는 작업으로, 딥러닝을 이용하면 손쉽게 생성할 수 있다. 한글 폰트를 생성하는 연구는 딥러닝 기반의 생성 모델들과 발맞추기 위해 프로세스 자동화 관점에서 한글 폰트 데이터셋을 준비하는 것이 중요하다. 이를 위하여 본 논문에서는 딥러닝 기반의 한글 폰트 연구를 위한 한글 폰트 데이터셋을 제안하고, 그 데이터셋을 구성하는 방법을 기술한다. 본 논문에서 제안하는 한글 폰트 데이터셋을 기반으로 딥러닝 한글 폰트 생성 어플리케이션에 적용하는 과정을 통해 제안하는 데이터셋 구성의 유용성을 보인다.

키워드 : 딥러닝, 폰트데이터, 폰트자동생성, 한글데이터셋, 한글폰트

## 1. 서 론

최근 구글 딥마인드에서 개발한 바둑 인공지능 알파고(AlphaGo)의 등장으로 인공지능에 대한 관심이 증가하면서, 이를 이용한 다양한 연구들이 진행되고 있다. 인공지능의 여러 기법 중 하나인 기계 학습은 기계가 코드로 명시되지 않은 동작을 데이터로부터 학습하여 실행할 수 있는 알고리즘을

개발하는 연구 분야이다. 딥러닝은 사람의 사고방식을 컴퓨터에게 가르쳐 학습된 모델을 이용하는 인공지능 및 기계학습의 하위 기술로서, 컴퓨터 비전, 음성 인식, 자연어 처리, 음성 신호처리 등과 같은 다양한 분야에 적용되고 있다[1].

타이포그래피 분야에서도 딥러닝을 이용한 다양한 연구들이 진행되고 있다. 특히 딥러닝 기반의 글자 인식과 생성 모델을 이용하여 자동으로 폰트를 생성하는 연구들이 최근에 발표되었다[2, 3]. 이들은 로마자의 대소문자, 숫자, 특수 기호를 포함하여 200여자를 디자인하는 과정을 거쳐 폰트를 생성한다. 반면에 한글 폰트는 초성 19자 중성 21자 종성 27자를 조합하여 만들어낼 수 있는 글자의 모든 경우를 고려해야 하며, 그 수는 총 11,172자 (19×21×(27+1), 마지막 1은 중성이 없는 경우임)에 달하며, 폰트 디자이너들이 새로운 한글 폰트를 제작하기 위해서는 엄청난 시간과 비용이 투입된다. 이에 모든 글자를 제작하지 않고, 사용빈도가 높은 대표

\* 이 논문은 2020년도 과학기술정보통신부의 재원으로 정보통신기술평가원의 지원(과제번호 2016-0-00166)과 2020년도 산업통상자원부의 창의산업융합특성화인재양성사업의 지원(과제번호 N0000717)을 받아 연구되었음.

† 비 회 원 : 송실대학교 컴퓨터공학과 박사과정

†† 비 회 원 : 송실대학교 IT융합물류학과 석사과정

††† 비 회 원 : 송실대학교 ㈜팅크웨어 연구원

†††† 준 회 원 : 송실대학교 컴퓨터공학과 박사과정

††††† 종신회원 : 송실대학교 컴퓨터학부 교수

Manuscript Received : October 28, 2020

Accepted : December 5, 2020

\* Corresponding Author : Jaeyoung Choi(choi@ssu.ac.kr)

글자 2,350자만 제작하기도 하지만, 이를 생성하기 위해서도 많은 비용이 들어간다[4, 5]. 딥러닝 기술을 활용하여 한글 폰트를 생성한다면, 한글 폰트를 제작하는데 들어가는 시간과 비용을 획기적으로 단축시킬 수 있을 것이다.

한글 폰트를 생성하는 딥러닝 어플리케이션의 처리과정을 살펴보면 크게 전처리-학습-검증의 3단계로 구성된다. 특히 전처리 단계에서는 딥러닝의 핵심 처리과정인 학습과정에 적합한 데이터셋이 입력으로 주어진다. 하지만 로마자와는 다르게 한글 폰트의 데이터셋은 글자 수가 아주 많고 글자 및 자소의 배치, 자소가 갖는 획의 디자인 요소들이 매우 다양하여 한글 폰트의 데이터셋을 구성하는 것이 용이하지 않다.

본 논문에서는 한글 폰트 글자의 특성을 반영한 데이터 셋을 제안한다. 또한 한글 폰트의 데이터셋을 설명하기 위한 구성정보와 이들을 처리하는 방법에 대해서 기술한다. 제안하는 데이터셋은 배포된 폰트들의 글자들이 갖는 특징을 값의 형태로 제공하여 딥러닝 학습과정에 입력정보로 활용할 수 있다. 실험에서는 제안하는 한글 폰트의 데이터셋을 활용한 딥러닝 어플리케이션을 통해 한글 폰트 생성과정을 보임으로써, 제안하는 한글 데이터셋의 유용성을 보였다.

## 2. 관련 연구

현재 딥러닝 연구들에서 사용되는 공개되어 있는 데이터셋들을 알아본다. 공개되어 있는 대표적인 데이터셋으로는 MNIST (Modified National Institute of Standards and Technology database)[6]가 있다. 이 데이터셋은 미국 국립표준기술연구소에서 만든 것으로 사람의 손글씨로 작성된 숫자들의 데이터베이스이다. MNIST 데이터셋은 딥러닝 모델의 이미지 인식, 이미지 분류 및 이미지 생성 분야에서 많이 사용된다.

CIFAR-10 (Canadian Institute For Advanced Research)[7] 데이터셋은 비행기, 차, 새, 고양이, 사슴, 개, 개구리, 말, 배, 그리고 트럭과 같은 10종류의 개체들이 포함된 사진들을 각각 분류하여 모아놓은 데이터셋이다. CIFAR-10은 대표적으로 이미지 분류 및 이미지 인식 분야에서 사용된다.

또한 딥러닝의 학습과정을 단축시키기 위해 연구된 Few-shot 폰트 스타일 전이학습연구에서는 흑백 및 컬러로 구성된 로마자 데이터셋을 이미지로 제공한다[8]. 특히 Few-shot 폰트 스타일의 전이학습 연구는 최근에 각광받는 연구로서, 로마자 외에도 한자 폰트를 생성하기 위한 다른 연구가 진행되었으며, 이 연구에서는 로마자와 한자들을 흑백과 컬러로 표현한 데이터셋을 이미지로 제공한다[9].

이처럼 딥러닝 연구에서 사용될 수 있는 다양한 데이터셋이 공개되어 있어서, 이 데이터셋을 기반으로 많은 딥러닝 연구들이 진행되었다. 영어와 한자 폰트 데이터셋은 공개되어 있지만, 한국어의 경우 공개되어 있는 데이터셋이 없다. 따라서 한글 폰트 연구를 진행하기 위해서는 각 연구자들이 데이터를 수집하고 이를 연구에 활용할 수 있도록 처리해야 하는

번거로움이 있다. 게다가 글자를 디자인하는 딥러닝 어플리케이션들은 단순히 글자 이미지 데이터셋들을 사용하는데, 한글 폰트는 로마자 폰트보다 더 많은 특징을 갖기 때문에 이들을 활용할 수 있는 정보가 필요하다

이를 해결하기 위해, 본 논문에서 딥러닝 기반의 한글 폰트 연구에서 사용할 수 있는 한글 폰트 데이터셋을 수집하였고, 이를 처리한 방법을 데이터셋과 함께 공개하였다.

## 3. 한글 폰트 데이터셋의 구성과 처리

본 절에서는 먼저 본 논문에서 제안하는 한글 폰트 데이터셋의 구성을 설명하고, 그 다음에 한글 폰트 데이터셋의 처리 방법에 대해 기술한다. 한글 폰트 데이터셋은 <https://github.com/stemfont/hangul-font-dataset>에서 다운로드 받을 수 있다. 현재 한글 고딕 폰트 31벌과 한글 명조 폰트 30벌의 한글 폰트 데이터셋 61벌이 공개되어 있다. 이 폰트셋은 네이버의 '나눔' 계열 폰트 17벌, 공공누리 1유형 폰트 39벌, 그리고 어도비와 구글이 합작하여 제작한 '본명조' 폰트 6벌 등 총 61벌로 구성되어 있다.

### 3.1 한글 폰트 데이터셋 구조

Table 1은 제안하는 한글 폰트 데이터셋이 갖는 정보들의 목록과 설명을 나타낸다. 한글 폰트 데이터셋은 key-value 구조를 갖는다. 각각의 key에 해당되는 항목은 리스트 형식으로 저장되어 있다.

Image 항목은 폰트 파일에서 생성한 글자 이미지의 픽셀 정보를 배열로 나타낸 값을 갖는다. 폰트 파일에서 생성한 글자는 흑백 이미지이다. 흑백 이미지를 배열로 변환하게 되면 2차원 배열로 표현할 수 있는데, 제안하는 폰트 데이터셋에서는 딥러닝 학습과정에서 처리되는 정보를 간소화시킬 수 있도록 1차원으로 변환한 후에 그 값을 저장한다.

Image size 항목은 Image 항목에 저장되어 있는 글자 이미지의 원본 크기로서, 글자 이미지의 가로 및 세로의 정보를 튜플(Tuple) 형태로 갖는다. 1차원 배열로 저장된 Image 항목을 원본 이미지로 복원하기 위해 사용된다.

Font name 항목은 폰트 파일 내부에 정의된 폰트 이름이다. 또한 Family name 항목은 해당 폰트가 어떤 패밀리에 속해있는지를 표현한다. 여기서 패밀리란 폰트 디자인의 토대가 되는 폰트의 이름을 말하며, 볼드(Bold), 이탤릭(Italic) 등과 같은 스타일이 단수 또는 다수개의 폰트 파일에서 각각 정의될 때 이들의 상위 또는 최상위 집합을 나타내는 정보이다. 예를 들어 "NanumGothic," "NanumGothicBold" 폰트의 경우, 모두 "NanumGothic" 패밀리에 속하는 폰트이므로 이 두 폰트의 Family name은 둘 다 "NanumGothic"이라는 값을 갖는다.

Width 항목은 글자의 폭 값을 나타내며, VWidth 항목은 글자의 높이 값을 의미한다. 한글은 VWidth 값이 대부분 고정되어 있다. Bound 항목은 글자를 감싸는 정사각형의 영역

Table 1. Description for the Proposed Hangul Font Dataset

KEY	TYPE	VALUE
Image	Array	image array
Image size	tuple(width, height)	image's width and height
Font name	String	Font name from source font files
Family name	String	Font Family name
Width	Number	Glyph's width
VWidth	Number	Glyph's vwidth
Bound	tuple(xmin, ymin, xmax, ymax)	Glyph's Bound
Contour number	Number	Number of glyph's contours
Weight	String	Thickness of glyph expressed in letter
TTFWeight	Number	Thickness of glyph expressed in value
SFD	String	Values for Spline Font Database
TTFWeight	Number	Thickness of glyph expressed in value
Glyph id	Number	Glyph's id using at SFD
Unicode	Number	Glyph's unicode
Copyright	String	Font copyright

을 표시하는 정보이다. 이 값은 x좌표와 y좌표의 최소값 및 최대값 튜플로 표현된다. Contour number 항목은 글자의 외곽선 개수를 나타낸다. 예를 들어 “가”의 경우에는 “ㄱ” 외곽선과 “ㅏ” 외곽선으로 표현할 수 있으므로, Contour number의 값은 2이다.

Weight 항목은 글자의 굵기를 키워드로 나타낸 값이다. 대표적으로는 “Book,” “Bold,” “Demi”가 있다. TTFWeight 항목은 글자의 굵기를 수치로 나타낸 것으로, Weight 값과 TTFWeight 값은 함께 제공된다. Weight는 일반적으로 “Book”이면 TTFWeight 값이 400으로 대응되며, “Bold”일 경우 700으로 대응된다.

SFD 항목은 Spline Font Database (SFD)의 정보를 담은 항목으로, SFD는 ASCII 코드로 표현되는 글자들을 네이티브 폰트로 처리하기 위한 글자의 벡터정보들을 나타낸 집합이다. 이들은 폰트를 편집할 수 있는 소프트웨어 중 하나인 fontforge에서 지원해주는 파일 형식이며[10], 이 데이터는 벡터형식의 폰트들을 학습하는 딥러닝 어플리케이션에서 사용될 수 있는 정보이다. 특히 SFD는 RNN 기반의 모델을 학습하는 과정에서 사용될 수 있다.

Glyph id 항목은 폰트에서 각 글자에 식별자를 할당한 값이며, 이 값은 SFD에서 같은 자소의 식별자를 참조할 때 사용되는 값이다. 이는 글자에 대응되는 Unicode와는 별개의 값을 가진다. Unicode 항목은 글자에 해당되는 유니코드 표준을 따르는 값이며, 상기한 Glyph id의 목적에 따라 별도의 값을 갖는다[11].

마지막으로 Version은 폰트 파일의 버전을, 그리고 Copyright는 폰트 파일에 명시되어 있는 저작권 정보를 의미한다.

제안하는 한글 폰트 데이터셋에 표현된 항목들은 이미 제작된 한글 폰트에 포함된 속성 정보를 따른다. 일반적으로 폰트

디자이너가 정의한 항목들은 Font name, Family name, Weight, Version, Copyright 등이 있다.

### 3.2 한글 폰트 데이터셋 처리방법

Fig. 1은 제안하는 한글 폰트 데이터셋을 구성하는 방법을 나타낸 것이며, 여섯 단계의 순서를 가진다.

1단계는 fontforge API를 이용하여 폰트 파일을 열어 정보를 읽는 것으로 처리과정이 시작된다. 그 다음으로 2단계는 폰트 파일이 사용자가 만들고자 하는 글자가 존재하는지를 검사한다. 이를 검사하는 방식은 다음과 같다. 먼저 데이터셋을 생성하기 전에 사용자가 생성할 글자의 로캘(Locale) 정보와 그에 해당되는 유니코드 집합을 고르는 과정을 수행하며, 그 후에 유니코드 집합에 들어있는 값을 이용하여 폰트 파일에 유니코드 값과 대응되는 글자들이 있는지의 여부를 담은 리스트를 생성하고 다음 단계로 넘어간다[12]. 3단계로 유니코드 값을 이용하여 폰트에 글자가 있다는 것을 확인하면, 폰트 파일에 저장되어 있는 글자 정보를 불러온다. 여기서 불러오는 글자 정보는 3.1절에서 기술한 한글 폰트 데이터셋 구조의 글자의 Width, VWidth, Glyph id, Bound, Family name, Weight, TTFWeight, Version, Copyright이다. 이렇게 불러온 정보는 별도 저장되고 다음단계로 넘어간다.

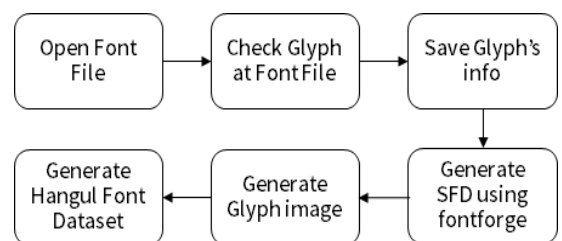


Fig. 1. Process for Generating Hangul Font Dataset

INDEX KEY	0(‘가’)	300(‘납’)	1342(‘썬’)
Image	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
Image size	64,64	64,64	64,64
Glyph id	1086	1386	2428
Weight	Book	Book	Book
Family Name	NanumGothic	NanumGothic	NanumGothic
Bound	[60.27777777777778, -140.0, 923.0, 782.0]	[121.0, -132.0, 923.0, 782.0]	[43.0, -122.0, 825.0, 782.0]
Width	940	940	940
Copyright	Copyright (c) 2011 NHN Corporation. All rights...	Copyright (c) 2011 NHN Corporation. All rights...	Copyright (c) 2011 NHN Corporation. All rights...
Version	3.021;PS 1;hotconv 1.0.57;makeotf.lib2.0.21895	3.021;PS 1;hotconv 1.0.57;makeotf.lib2.0.21895	3.021;PS 1;hotconv 1.0.57;makeotf.lib2.0.21895
Font name	NanumGothic	NanumGothic	NanumGothic
Unicode	44032	45225	50448
VWidth	1000	1000	1000
SFD	SplineSet\n507 635 m 0,0,1\n 502 549 502 549 4...	SplineSet\n274 -132 m 2,0,1\n 250 -132 2 50 -13...	SplineSet\n526 716 m 0,0,1\n 522 697 522 697 5...
Contour number	2	4	4
TTFWeight	400	400	400

Fig. 2. Examples of “NanumGothic” Font Dataset Based on the Proposed Dataset Structure

4단계에서는 fontforge API를 이용하여 이진 형식으로 저장되어 있는 폰트 파일을 Spline Font Dataset 형식으로 바꿔준다. Spline Font Dataset을 이용하여 글자의 외곽선 개수와 글자의 굵기 정보를 가져온다. 5단계에는 1단계 입력자료의 폰트 파일을 참조하여 글자 이미지를 생성한다. 폰트 파일로부터 생성되는 이미지는 2차원 배열로 표현된 흑백 이미지이며, 3.1절에서 기술한 딥러닝 학습과정에서 간소화된 정보로 입력자료를 활용하기 위해 이들을 1차원 배열로 변환한다.

마지막으로 6단계에서는 1~5단계를 통해 처리된 데이터를 병합하여 하나의 폰트 정보에 대해서 이미지와 글자의 특성을 각각 폰트의 데이터셋으로 만들어주는 과정을 수행한다. Fig. 2는 “나눔고딕” 폰트를 이용하여 생성한 폰트 데이터셋을 속성별로 나타낸 것이며, 가장 위의 0, 300, 1342의 수치들은 폰트 각 글자들의 인덱스 값을 나타낸 것이다. 이 때, 0번째 글자는 “가” 300번째 글자는 “납” 1342번째 글자는 “썬”이다. 아래 항목들은 한글 폰트 데이터셋의 key 항목과 각 글자별 key 항목들에 대응되는 value 항목들을 나타낸 것이다.

제안된 한글 폰트 데이터셋을 적용할 때, 구조를 변경해야 하는 경우에는 제공된 프로그램을 통해 수정가능하다. 각 처리과정은 모듈화되어 있으므로, 생성 과정을 수정하여 변경할 수 있다. 예를 들어 사용자가 한글 폰트 데이터셋에 있는 특정 정보가 필요없다면 해당 정보를 생성하는 모듈을 삭제하고 나머지 프로그램을 실행시킨다. 반대로, 특정 속성이 필요하다면 이를 처리하는 모듈을 명시하여 생성한다.

#### 4. 실험

본 절에서는 제안하는 한글 폰트 데이터셋을 이용하여 한글 글자 이미지를 생성하는 실험을 진행하였다. 실험을 통해 제안된 한글 폰트 데이터셋이 딥러닝 기반의 폰트 생성 어플리케이션에 적용한 결과와 효용성을 보인다. 딥러닝 학습과정으로 언급된 단계 중 전처리단계에서 제안된 한글 폰트 데이터셋이 적용되며, 3.2절에서 언급된 처리 단계 중에서 3단계인 폰트 이미지와 폰트의 특성을 파일에 저장하는 기능은 Python의 pickle 패키지를 이용하였고, 5단계의 폰트 파일의 이미지를 생성하는 과정은 Python의 Pillow 패키지를 이용하였다. 또한 딥러닝 학습과정이 동작하는 환경은 python 3.6.5 버전 및 tensorflow 1.15.0 버전을 이용하였다.

한글 글자 생성을 위해 딥러닝의 모델을 이용하여 한글 글자 이미지를 생성하는 실험을 진행하였다. 대표적인 딥러닝 생성 모델인 GAN (Generative Adversarial Network)을 사용하였다. 본 절에서는 GAN의 대표적 모델인 pix2pix[13] 모델을 변형한 SDKorFont[14] 모델을 이용하여 실험을 진행하였다.

pix2pix 모델은 입력 이미지(Source)를 목표 이미지(Target)로 변환시키는 이미지 변환 모델로서, 입력 이미지와 목표 이미지의 쌍(Pair)을 입력으로 주고, 목표 이미지와 유사한 출력이미지를 output으로 생성한다. SDKorFont 모델은 pix2pix 기반으로 한글 글자 이미지를 생성하기 위해 특화된 딥러닝 모델이며, pix2pix와 동일한 형태의 입력자



Fig. 3. Example of Input Dataset Image for Deep Learning Applications (SDKorFont)

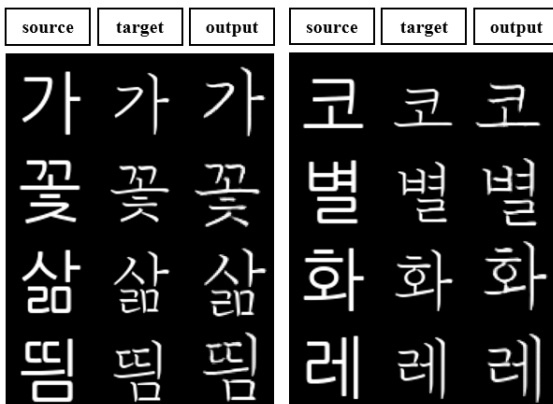


Fig. 4. Result from Deep Learning Applications (SDKorFont) by Source, Target, and Output

료를 통해 결과물을 생성한다. 입력 자료는 “나눔고딕” 폰트 글자 이미지 2,350장을, 목표 글자 이미지는 “Mapo꽃섬” 폰트 글자 이미지 2,350장을 사용하였다. Fig. 3은 입력자료로 사용된 8개의 글자를 나타낸 것이다. SDKorFont 모델을 이용하여 입력자료를 학습하는 과정에서의 파라미터로 epoch를 30으로 설정하였다. 그 결과 Fig. 4와 같은 결과물을 얻을 수 있었으며, 입력과 출력을 비교하기 위해서 source, target, 그리고 output으로 나누어진 글자들을 통해 목표 글자 이미지와 유사한 결과물을 만들어낸 것을 확인할 수 있다.

### 5. 결 론

본 논문에서는 딥러닝 기반의 폰트 생성 연구를 위한 한글 폰트 데이터셋을 제안하였다. 데이터셋에는 한글 폰트 61벌

과 함께, 폰트들의 속성들을 포함하였으며, 딥러닝 어플리케이션에 적용하는 과정을 보였다.

제안하는 한글 폰트 데이터셋은 기존의 로마자와 한자 등에서 활용할 수 있는 입력정보인 글자 이미지와 한글 폰트 특성을 반영하였다. 또한 한글 폰트 데이터셋의 처리과정의 각 단계는 수정 가능한 모듈로 구현하였으므로, 사용자의 연구 목적에 따라 데이터셋의 구조를 유연하게 변경하여 사용할 수 있다. 실험을 통해서 딥러닝 기반의 한글 폰트 생성 과정을 보였다.

본 논문의 실험에서는 한글 데이터셋을 이용하여 딥러닝 기반의 한글 생성을 진행하였지만, 한글 폰트와 관련된 다른 연구에서도 이 데이터셋이 사용될 수 있을 것으로 기대한다.

### References

- [1] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.35, No.8, pp.1798-1828, 2013.
- [2] Y. Jiang, Z. Lian, Y. Tang, and J. Xiao, “DCFont: An end-to-end deep chinese font generation system,” *SIGGRAPH Asia 2017, Technical Briefs*, 2017.
- [3] Y. Tian, “zi2zi: Master chinese calligraphy with conditional adversarial networks,” [Internet] <https://github.com/kaonashi-tyc/zi2zi>.
- [4] J. Choi and S. Hong, “Aspects of the development of Korean font design in the digital era,” *Journal of Digital Design*, Vol.8, No.2, pp.173-182, Apr. 2008.
- [5] Korean Publishing Research Institute, *Basic research in Hangul Style (in Korean)*, Korean Publishing Research Institute, 1990.
- [6] Y. LeCun, C. Cortes, and C. Burges, “The mnist database of handwritten digit,” [Internet] <http://yann.lecun.com/exdb/mnist/>.
- [7] A. Krizhevsky, V. Nair, and G. Hinton, “The CIFAR-10 dataset,” [Internet] <https://www.cs.toronto.edu/~kriz/cifar.html>.
- [8] S. Azadi, M. Fisher, V. Kim, Z. Wang, E. Shechtman, and T. Darrell, “Multi-Content GAN for Few-Shot Font Style Transfer,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.7564-7573, 2018.
- [9] Y. Gao, Y. Guo, Z. Lian, Y. Tang, and J. Xiao, “Artistic glyph image synthesis via one-stage few-shot learning,” *ACM Transactions on Graphics*, Vol.38, No.6, 2019.
- [10] FontForge, Spline font Database [Internet], <https://fontforge.org/docs/techref/sfdformat.html>.
- [11] The Unicode Consortium, *The Unicode Standard* [Internet],

<https://www.unicode.org/standard/standard.html>.

- [12] H. Min, A. Hassan, J. Suk. and J. Choi, "Font image dataset auto generating module based on unicode (in Korean)," in *Proc. Korea Computer Congress 2020 (KCC2020)*, Vol.46, No.01, pp.1818-1820, 2019.
- [13] P. Isola, J.-Y. Zhu, T. Zhou, and A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [14] D. H. Ko, A. Hassan, J. Suk, and J. Choi, "Korean font synthesis with GANs," in *International Journal of Computer Theory and Engineering*, Vol.12, No.4, pp.92-96, 2020.



### 고 흥 희

<https://orcid.org/0000-0002-4510-0385>  
 e-mail : debbie.pust@gmail.com  
 1983년 고려대학교 물리학과(학사)  
 1989년 Florida State University  
 컴퓨터공학과(석사)  
 현 재 송실대학교 컴퓨터학과 박사과정

관심분야: 딥러닝, 자동 폰트 생성, 폰트 데이터셋



### 이 현 수

<https://orcid.org/0000-0001-9176-1670>  
 e-mail : leehyun121841@gmail.com  
 21016년 송실대학교 소프트웨어학부(학사)  
 2020년~현 재 송실대학교 IT유통물류학과  
 석사과정

관심분야: 딥러닝, 자동 폰트 생성, 폰트 데이터셋



### 석 정 재

<https://orcid.org/0000-0002-3855-9999>  
 e-mail : jjsuk@inavi.kr  
 2017년 한경대학교 컴퓨터공학과(학사)  
 2020년 송실대학교 컴퓨터공학과(석사)  
 2020년~현 재 (주)탱크웨어 연구원  
 관심분야: 딥러닝, 자동 폰트 생성, 폰트  
 데이터셋



### Ammar UI Hassan

<https://orcid.org/0000-0001-6744-507X>  
 e-mail : ammar.instantsoft@gmail.com  
 2013년 International Islamic University  
 Islamabad, Pakistan 컴퓨터공학부  
 (학사)  
 2018년 송실대학교 컴퓨터공학과(석사)

2018년~현 재 송실대학교 컴퓨터공학과 박사과정  
관심분야: 딥러닝, 자동 폰트 생성, 폰트 데이터셋



### 최 재 영

<https://orcid.org/0000-0002-7321-9682>  
 e-mail : choi@ssu.ac.kr  
 1984년 서울대학교 제어계측공학과(학사)  
 1986년 미국 남가주대학교 전기공학과  
 (컴퓨터공학)(석사)  
 1991년 미국 코넬대학교 전기공학부  
 (컴퓨터공학)(박사)

1992년~1994년 미국 국립오크리지연구소 연구원  
 1994년~1995년 미국 테네시 주립대학교 연구교수  
 1995년~현 재 송실대학교 컴퓨터학부 교수  
 관심분야: 디지털 타이포그래피, 시스템소프트웨어, 병렬/분산처리,  
 고성능컴퓨팅