

이미지 생성모형을 활용한 한글 손글씨 생성 서비스

조영원⁰, Avinash Kumar, 양의종, 장진영, 김다은, 최재영

송실대학교

{ywjo, kumar, calzone0404, k09176, adune}@soongsil.ac.kr, choi@ssu.ac.kr

Korean Handwriting Font Generation Service using Image Generation Model

Youngwon Jo⁰, Avinash Kumar, Uijong Yang, Jinyeong Jang, Daeun Kim, Jeayoung Choi
Soongsil university

요약

손글씨는 개인의 고유한 개성과 특징을 표현하는 수단이다. 종이 혹은 태블릿에서 작성된 손글씨를 디지털 환경에서 사용하려면 폰트 파일로 변환해야 한다. 로마자와 달리 한글 폰트 파일을 제작하기 위해서는 완성형 2,350자 혹은 유니코드 11,172자의 글자를 디자인해야 하기 때문에 많은 시간과 노력이 요구된다. 최근 컴퓨터 기술의 발달로 손글씨를 폰트 파일로 제작할 수 있게 되었으며, 상업적인 서비스도 제공되기 시작하였다. 본 논문에서는 이미지 생성 모형을 활용하여 손글씨 폰트를 생성하는 '내손글' 서비스를 제안하고 손글씨 폰트 제작 과정과 함께 서비스 과정과 손글씨 생성 결과를 보여준다.

주제어: 한글 손글씨, 폰트 생성, 이미지 생성 모델, 손글씨 서비스

1. 서론

손글씨는 기계를 사용하지 않고, 펜이나 붓 등의 필기를 이용해 손으로 직접 쓴 글씨를 의미한다. 손글씨는 형식에 제한이 없기 때문에 활자와 달리 개인의 개성과 특징을 표현할 수 있으며, 창의적인 표현이 가능하다. PC와 스마트폰 같은 디지털 환경에서 손글씨를 사용하기 위해서는, 사용자에게 폰트 파일로 제공되어야 한다.

로마자와 달리 한글 폰트 파일을 제작하기 위해서는 완성형 2,350자 혹은 유니코드 11,172자를 한 글자씩 디자인해야 하기 때문에, 전문적인 폰트 디자이너들도 한글 폰트 1벌을 제작하기 위해서는 많은 시간과 노력이 소요된다. 최근 컴퓨터 기술의 발달로 인해 개인의 손글씨도 쉽게 폰트 파일로 제작할 수 있게 되었다.

스마트폰 및 PC의 보급과 함께 자신의 개성과 취향을 나타내려는 욕구가 많아지면서, 저렴한 가격으로 개인의 손글씨를 제작해주는 상업적인 서비스도 제공되기 시작하였다. 이러한 서비스는 폰트 디자이너가 직접 손글씨를 디자인하거나 머신러닝 기술을 활용하여 폰트를 제작한다. 머신러닝에서 이미지 생성은, 사람이 개입하지 않고 한 번에 원하는 이미지를 생성하는 end-to-end 모델이 바람직하다고 여겨진다. 그러나 우리는 오랜 기간 동안 많은 시행착오를 거치면서 폰트 생성에 관한 연구를 진행해왔고, 만족할만한 품질의 손글씨를 한 번에 바로 생성하는 것은 쉽지 않은 일이라고 결론에 도달하였다. 2,350자 혹은 11,172자의 많은 글자들을 똑같은 스타일의 폰트를 제작하기 위해서는 글자 자체의 고유한 구조적 특징이 유지되도록 여러 단계를 거쳐야 한다.

본 논문에서는 한글 손글씨를 생성하고, 이를 서비스하기 위한 '내손글' 서비스를 제안한다. 먼저 이미지 생성모형을 활용하여 한글 손글씨 폰트 생성하는 과정을

간단히 설명한다. 각 과정은 머신러닝 기술을 이용하여 이미지를 생성하는 창의적이고 복잡한 모델을 사용하고 있으며, 각각의 과정들은 향후에 다른 논문들로 상세히 기술할 예정이다. 본 논문의 2장에서는 관련 연구를, 3장에서는 손글씨 폰트 생성 과정을 기술하였다, 4장에서는 '내손글' 서비스 제작 과정과 함께, 우수한 품질의 손글씨가 생성되는 과정을 각 과정별로 결과 이미지와 함께 보여준다. 5장에서는 생성된 폰트들의 품질을 평가하였으며, 6장에서는 결론과 향후 연구를 기술한다.

2. 관련 연구

2.1 머신러닝을 활용한 폰트 생성 기술

머신러닝의 발전에 따라 GAN (Generative Adversarial Networks)[1]이나 이미지간 변환 모델인 Pix2Pix (Paired Image-to-Image translation)[2]와 같은 생성 모델이 연구되고 있으며, 폰트 분야에서도 zi2zi[3]와 DCFont[4] 같은 생성 모델을 기반으로 폰트 이미지 변환 및 생성에 관한 연구들이 진행되었다.

Zi2zi[3]는 글자 이미지에서 스타일 전이를 수행하기 위해 글자 자체의 정보와 스타일 정보를 각각 인코딩하여, 하나의 콘텐츠에 다양한 스타일을 적용할 수 있는 모델이다. DCFont[4]는 zi2zi보다 정밀하고 자연스러운 글꼴 변환을 위해 콘텐츠와 글자의 획을 인코딩하여 특정 글자의 세부적인 스타일을 학습한다. 이를 통해 단일 글자의 스타일을 학습한 뒤 나머지 글자에도 동일한 스타일을 전이가 가능하다. 그러나 이러한 모델들은 주로 중국어를 대상으로 개발되었기 때문에 한글을 생성하였을 때는 글자의 구조적 특징이 손실될 수 있다.

2.2 한글 폰트 모델

우리는 지난 5년 이상 한글 폰트생성에 관한 연구를 꾸준히 진행해왔으며. 그 결과를 CKFont [5]와 CKFont2 [6], 그리고 위치 기반 구성요소를 사용한 한글 폰트 이미지 생성[7] 등 여러 논문으로 발표하였다. CKFont [5] 모델은 한글을 구성하는 67개 (쌍/복자음 및 복모음을 포함한 초성 19개, 중성 21개, 종성 27개)의 구성요소에서, 조합 특성을 이용하여 초성과 중복되는 16개 중성을 제외한 51개의 서로 다른 구성요소를 포함하는 28개의 글자를 이용하여 한글 1벌을 생성하는 모델이다. CKFont2[6] 모델은 쌍/복자음, 복모음과 같이 기본 자모의 결합으로 생성될 수 있는 구성요소를 제외한, 24개의 기본적인 구성요소만을 이용하였고, 이를 포함하는 14개 글자로 모든 한글을 생성하는 모델이다.

위치 기반 구성요소를 사용한 한글 폰트 이미지 생성 [7]에서는 글자의 위치 기반의 구성요소를 정보를 활용하여, 한글의 구조적 특성을 이해하고 이를 통해 글자를 생성한다. 해당 연구에서는 객체 탐지 모델인 YOLO (You Only Look Once)[8] 모델을 사용하여 초성, 중성, 종성의 위치에 따라 67개의 자소를 분리하고 GAN을 통해 재결합하는 방법을 사용하였다.

2.3 손글씨 폰트 생성 서비스 동향

손글씨를 이용하여 폰트를 생성하는 서비스는 크게 폰트 디자이너가 직접 제작하거나 생성 모델 기반의 폰트를 제작하는 방법이 있다. 디자이너가 직접 제작하는 서비스는 ‘문자동맹’ 이 폰트 생성서비스를 제공하고 있다. 문자동맹은 기본 기호와 완성형 한글 2,350자를 제작하기 위해 약 500여자의 손글씨를 작성하고 제출하여야 하며, 제작에는 2주 정도의 시간이 소요된다.

네이버는 2019년 한글날에 생성 모델을 이용하여 손글씨를 폰트로 제작해주는 ‘한글날 손글씨 공모전’ 행사 [9]를 진행하였다. 한 벌의 폰트를 생성하기 위해 100자 이상의 손글씨를 입력받아 손글씨 폰트를 생성하였으며, 생성된 글자의 품질을 높이기 위하여 폰트 디자이너가 직접 검증 및 보정을 진행하였다.

보이저엑스는 생성 모델을 이용하여 폰트를 생성하는 ‘온글잇(OwnGlyph)’ 서비스를 제공하였다. 이 서비스는 80여자의 손글씨를 입력받아, 완성형 2,350자를 포함하는 2,780자를 생성하고, 결과를 폰트 디자이너가 마무리하는 방법을 이용하였다 (저렴한 기본 서비스에서는 12글자를 입력으로 받아 2,350자를 생성함.) 하지만 온글잇은 2023년에 서비스가 종료되었다.

본 논문에서는 다양한 폰트 생성 연구를 기반으로, 손글씨를 입력받아 전체 글자를 생성하고 보정하여 폰트를 제작하고 사용자에게 제공하는 ‘내손글’ 서비스를 제안하며 이는 단시간에 고품질의 폰트를 제작할 수 있다.

3. 손글씨 폰트 생성 과정

‘내손글’ 서비스의 손글씨 폰트 생성 과정은 크게 다

음과 같은 세 가지의 과정을 거치게 되는데, 본 절에서는 각 과정을 세부적으로 설명한다.

1. 한글의 조합성과 위치적 정보를 활용하여, 입력된 글자의 자소를 분리하고, 이를 결합하여 전체 글자를 생성하는 ‘위치 기반 한글 생성’
2. 주어진 글자의 구조적 특징을 보정하기 위한 ‘Font2Font’
3. 주어진 폰트의 스타일을 보정하기 위한 ‘AFAGAN’

3.1 자소 위치 기반 한글 생성

조합형 한글 생성과정

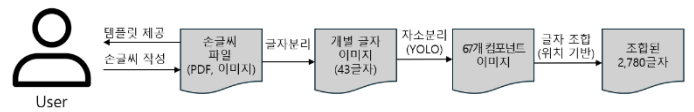


그림 1. 자소 위치 기반의 조합형 한글 생성 과정

그림 1은 자소의 위치를 기반으로 2,780자의 한글을 생성하는 과정을 보여준다. 먼저 한글의 조합성과 위치적 정보를 활용하여 글자를 생성하기 위해 43자의 글자를 입력받아 개별 글자 이미지로 분리한다. 그 다음에 YOLO 모델을 사용하여 개별 글자 이미지들을 다시 위치가 포함된 자소로 분리하여 67개의 기본적인 자소를 추출한다. 이를 기반으로 위치 기반의 글자 조합방식으로 2,780자의 한글을 생성한다. 이 과정을 통해 생성된 이미지는 생성 모델에 활용된다.

3.2 Font2Font

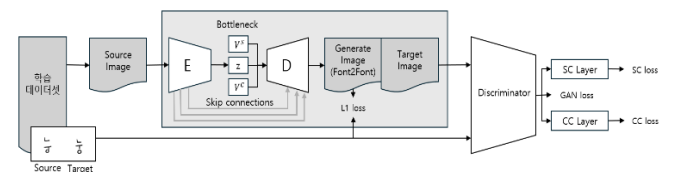


그림 2. Font2Font 구조

글자의 구조적 특징을 보정하고 자연스러운 글자를 생성하기 위해 Font2Font를 사용한다. Font2Font는 SkelGAN [10]의 글자를 골격화하는 아이디어를 바탕으로, 글자별로 구조적 특징 정보를 학습하고 입력되는 이미지의 구조적 특징을 보정한다.

학습 과정에 이용되는 데이터는 Target 이미지와 Source 이미지를 이용한다. Target 이미지는 목표하는 대상이 될 글자이고 Source 이미지는 위치 기반으로 생성된 글자 이미지다. 모델의 구조는 그림 2와 같이 단일 인코더와 단일 디코더로 구성된 생성자와 판별자로 구성하였다. 모델의 학습은 Source 이미지를 생성자의 인코더에 입력하고 글자 구조 특징을 계산하기 위해 이미지를 z 벡터로 변환한다. 변환된 z 벡터는 다양한 스타일을 적용하기 위해 V_s (스타일 벡터)와 문자의 종류를 파

악하기 위한 V_c (클래스 벡터)가 결합되어 단일 글꼴 문자가 여러 구조 스타일을 가질 수 있도록 구성되고, 계산된 벡터는 디코더를 거쳐 이미지로 생성된다. 생성된 이미지는 판별자로 입력되어 생성된 이미지와 Target 이미지를 비교하는 손실 함수로 계산한다.

모델의 손실 함수로는 GAN 모델에서 범용적으로 사용하는 GAN loss와 L1 loss를 사용하고 스타일과 클래스가 존재하는 글자 생성을 위해 SC loss, CC loss를 추가적으로 사용한다. SC loss (Style Classification Loss)는 생성자가 다양한 스타일을 학습하도록 유도한다. 생성된 이미지가 입력 이미지의 스타일에 맞게 생성되도록 하며, 판별자에 스타일 분류층을 추가하고 훈련 데이터 세트의 모든 스타일 수와 동일한 갯수와 동일한 수로 분류하여 생성된 이미지들이 어떤 스타일에 속하는지를 판단하는 손실 함수이다.

$$Loss_{sc} = -E_{x \sim \{X, X^t\}} \left(\sum_{i=1}^N s_i \text{Log}(\hat{s}_i) \right). \quad (1)$$

수식 1은 전체 데이터 X 와 특정 데이터 X^t 에서 샘플링된 항목 x 에 대해 실제 값 s_i 와 모델이 예측한 값 \hat{s}_i 사이의 로그 손실을 계산하고, 이를 기대값 E 로 여러 손실을 평균화하여 모델이 주어진 스타일을 분류할 수 있다. CC loss (Character Classification Loss)는 생성자가 생성하는 이미지가 목표 문자와 의미적으로 일치하도록 한다. 판별자에 문자 분류 층을 추가하고, 해당 층은 훈련 데이터 세트의 모든 문자의 갯수와 동일한 수로 분류하여 생성된 이미지들이 올바르게 인식되는지를 판단하는 손실 함수이다.

$$Loss_{cc} = -E_{x \sim \{X\}} \left(\sum_{i=1}^N c_i \text{Log}(\hat{c}_i) \right). \quad (2)$$

수식 2는 주어진 전체 데이터 X 에서 샘플링된 항목 x 에 대해 실제 값 c_i 와 모델이 예측한 값 \hat{c}_i 사이의 로그 손실을 계산하고 이를 기대값 E 로 여러 손실을 평균화하여 생성하는 이미지가 목표 문자와 의미적으로 일치하도록 한다.

3.3 AFAGAN (Attention based Font Alignment GAN)

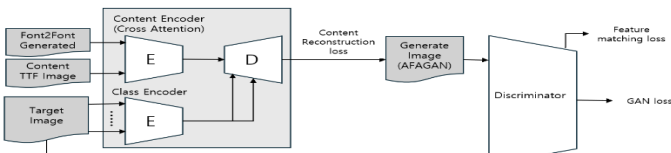


그림 3. AFAGAN 구조

글자 이미지에 손글씨 스타일을 더하기 위해 AFAGAN (Attention based Font Alignment GAN)을 사용한다. AFAGAN은 FUNIT[11] 기반의 모델이며, FUNIT 인코더에 입력을 추가하여 사용한다.

학습 과정에서 이용되는 데이터는 Font2Font 모델의 생성 결과와 Content TTF 이미지, 그리고 Target 이미지

가 이용된다. Content TTF 이미지는 글자의 구조와 글자의 의미를 유지하기 위해 활자 스타일의 폰트 파일에서 출력한 이미지이다. 모델의 구조는 그림 3과 같이 두 개의 인코더와 한 개의 디코더로 구성된 생성자와 판별자로 구성하였다.

모델의 학습은 콘텐츠 인코더에 Font2Font 생성 결과와 Content TTF 이미지가 입력된다. 두 이미지는 인코더에서 Cross Attention 과정을 거쳐 벡터로 출력된다. 스타일 인코더에서는 Few-Shot으로 학습되기 때문에 적은 수의 스타일로도 스타일이 전이되며, 타겟 스타일 이미지들을 입력받아 벡터로 출력된다. 출력된 두 벡터는 디코더에서 AdaIN Residual blocks을 통해 결합되어 이미지가 생성된다.

사용되는 손실 함수로는 GAN loss, 이미지의 콘텐츠 정보를 유지하기 위한 Reconstruction loss, 그리고 스타일을 유사하게 생성하기 위한 Feature matching loss가 사용된다. Reconstruction loss는 타겟 이미지와 생성된 이미지 간의 특징 표현 차이를 계산하고 이미지의 콘텐츠를 유지하는 손실 함수이다.

$$Loss_c(G) = E_{x_a, t_g} \left[\left| \mu(G(x_a, s_b)) - \mu(t_g) \right|_1 \right]. \quad (3)$$

수식 3의 $G(x_a, s_b)$ 는 콘텐츠 x_a 와 스타일 s_b 를 사용하여 생성된 이미지이며 t_g 는 타겟 이미지이다. 각 이미지는 특징 맵의 평균 연산 μ 을 통해 두 이미지 간의 차이를 계산한다. 마지막으로 두 이미지의 절대값 차이들의 평균을 L1 norm으로 계산하여 이미지의 콘텐츠 정보를 유지하도록 한다.

Feature matching loss는 타겟 이미지와 생성된 이미지의 특징 차이를 계산하고 유사한 스타일을 생성하도록 하는 손실 함수이다.

$$Loss_f(G) = E_{x_a, t_g} \left[\left| \mu(\phi(G(x_a, s_b))) - \mu(\phi(t_g)) \right|_1 \right]. \quad (4)$$

수식 4의 $G(x_a, s_b)$ 는 콘텐츠 x_a 와 스타일 s_b 를 사용하여 생성된 이미지이며 t_g 는 타겟 이미지이다. 생성된 이미지와 타겟 이미지는 ϕ 를 통해 특징이 추출되고 μ 를 통해 특징 맵의 평균 연산을 계산한다. 마지막으로 두 이미지의 평균값들 차이를 L1 norm으로 측정한다.

4. 손글씨 폰트 생성 서비스

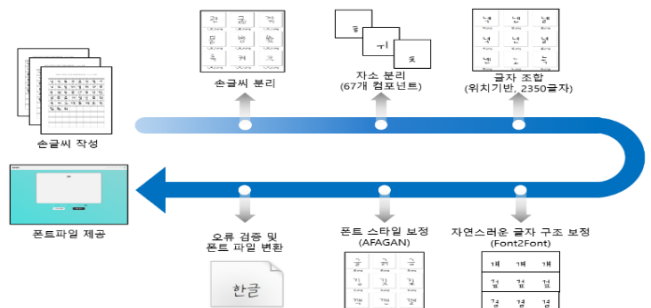


그림 4. '내손글' 폰트 생성 서비스 구조



그림 5. 개별 글자 이미지



그림 6. 자소로 분리된 이미지

그림 4는 제안한 손글씨 폰트 생성 서비스 과정이다. 생성 서비스는 크게 4 단계로 진행되며, 본 절에서는 각 과정을 세부적으로 설명한다.

1. 손글씨 입력 및 위치 기반 한글 생성
2. 주어진 글자의 구조적 특징을 보정하기 위한 'Font2Font'
3. 주어진 폰트의 스타일을 보정하기 위한 'AFAGAN'
4. 오류폰트 재생성 및 폰트 파일 제공

또한 모델의 학습은 공개되어 있는 69벌의 손글씨 폰트 파일을 사용하여 완성형 한글 2,350자로 출력한 162,840장의 이미지를 사용하고, NVIDIA GeForce RTX 4090 환경에서 진행하였다.

4.1 손글씨 입력 및 위치 기반 한글 생성



그림 7. 위치 기반 조합 결과 (일부)

손글씨 폰트 생성 서비스를 위해 먼저 사용자는 서비

스에서 제공되는 양식을 받아, 한글을 구성하는 67개의 자소를 포함하는 43자의 특정 글자를 손글씨로 작성하여 업로드한다.

'내손글' 서비스는 43개의 글자를 입력받아 2,780자의 폰트 1벌을 생성하기 위해 다음의 과정을 거친다. 입력받은 손글씨는 그림 5와 같이 개별 글자 이미지로 분리되고, 개별 글자는 다시 자소로 분해되어 그림 6과 같이 자소로 분리된 이미지가 만들어진다. 만들어진 자소는 위치 기반의 조합 방식을 통해 그림 7과 같이 2,780자의 글자가 생성된다. 이 과정을 통해 생성된 이미지는 Font2Font 모델에 활용된다.

4.2 글자의 구조적 특징 보정

Input (위치 기반)	Output (Font2Font)	Input (규칙 기반)	Output (Font2Font)	Input (규칙 기반)	Output (Font2Font)
덜	덜	츄	츄	ㅌ	ㅌ
뽕	뽕	괘	괘	ㅈ	ㅈ

그림 8. 구조적 특징 보정 결과

위치 기반으로 생성된 글자는 구조적 특징을 보정하기 위해 사전 학습된 Font2Font 모델을 이용한다. 모델은 사용자에게 입력받은 손글씨 이미지로 미세 조정을 진행하고, 그림 8과 같이 글자의 구조가 보정된 결과를 출력한다.

	위치 기반	Font2Font	AFAGAN	Target	위치 기반	Font2Font	AFAGAN	Target	위치 기반	Font2Font	AFAGAN	Target
동근인연	띠	띠	띠	띠	코	코	코	코	귀	귀	귀	귀
	갠	갠	갠	갠	녹	녹	녹	녹	죤	죤	죤	죤
따약단단	띠	띠	띠	띠	코	코	코	코	귀	귀	귀	귀
	갠	갠	갠	갠	녹	녹	녹	녹	죤	죤	죤	죤
말에게 엄마가	띠	띠	띠	띠	코	코	코	코	귀	귀	귀	귀
	갠	갠	갠	갠	녹	녹	녹	녹	죤	죤	죤	죤

그림 10. 서비스 과정별 손글씨 출력 결과

4.3 폰트의 스타일 보정

Input (Font2Font)	Output (AFAGAN)	Input (Font2Font)	Output (AFAGAN)	Input (Font2Font)	Output (AFAGAN)
더 2	더 2	췌 인	췌 인	췌 20	췌 20
뵁 0	뵁 0	과 ㅁ	과 ㅁ	쟝 0	쟝 0

그림 9. 스타일 보정 결과

글자의 구조적 특징이 보정된 이미지는 스타일이 적용된 선명한 글자를 위해 사전 학습된 AFAGAN을 이용한다. 모델은 사용자에게 입력받은 손글씨 이미지로 미세 조정을 진행하고 Font2Font로 생성된 결과를 입력하여 그림 9와 같이 스타일이 보정된 결과를 출력한다.

4.4 오류폰트 재생성 및 폰트 파일 제공

마지막으로 생성된 글자들이 내용적으로 올바르게 생성되었는지 (콘텐츠 검증), 그리고 글자들이 같은 스타일로 생성되었는지를 (스타일 검증) 확인한다.

먼저 생성된 글자들을 분류기(Classifier)에서 글자들의 콘텐츠를 검증하고, 동시에 임베딩 공간에서 동일한 스타일로 생성되었는지를 판단하는 스타일 검증을 진행한다. 이 과정에서 잘못 생성되었다고 판단된 오류폰트들을 분리하고 이들을 다시 Font2Font 모델을 사용하여 재생성된다.

최종적으로 검증 작업이 완료된 글자 이미지 세트는 ttf 파일로 변환되어 사용자에게 제공된다.

5. 생성된 폰트 평가

본 절에서는 서비스에서 생성된 결과를 평가한다. 평가에 사용된 폰트는 네이버에서 제공하는 나눔손글씨 5 종류를 사용하였으며, 서비스 과정에서 출력한 결과를 시각적인 비교와 함께, 평가 지표를 사용하여 정량적인 수치를 비교한다.

5.1 서비스 과정별 생성 결과

모델의 성능을 평가하기 위해 각 과정에서 생성된 이미지를 시각적으로 비교한다. 그림 10은 서비스 과정에서 생성된 이미지를 보여준다. 왼쪽부터 위치 기반으로 생성한 이미지이며, Font2Font는 위치 기반으로 생성된 이미지를 Font2Font 모델에 입력하여 생성된 결과이다. AFAGAN은 Font2Font 모델의 출력결과를 AFAGAN 모델에 입력하여 생성한 결과이며, 마지막으로 Target은 목표로 하는 최종 이미지이다.

Font2Font 모델에서 출력된 글자는 구조적 특징을 보정하기 위해 초성이나 중성의 위치가 이동되는 것을 확인할 수 있었으며, AFAGAN 모델에서 출력된 글자는 일부분의 글자 이미지가 흐려지거나 잘린 자소의 획을 보정하는 모습과 Target 이미지의 스타일과 유사한 결과를 생성해내는 것을 확인할 수 있었다.

5.2 평가 지표

사용된 평가 지표는 이미지 픽셀 기반의 오류를 측정하는 RMSE (Root Mean Squared Error), 이미지의 구조적 유사성을 평가하는 SSIM (Structural Similarity Index), 이미지의 시각적 유사성을 평가하는 LPIPS (Learned Perceptual Image Patch Similarity), 그리고 이미지 간 분포 차이를 비교할 수 있는 FID (Fréchet Inception

Distance)를 사용하였다.

표 1. 모델별 평가 지표 결과

모델	RMSE(↓)	SSIM(↑)	LPIPS(↓)	FID(↓)
위치 기반	0.543	0.93	0.097	19.114
Font2Font	2.081	0.954	0.055	22.786
AFAGAN	1.961	0.958	0.047	5.155

각 모델의 정량적인 평가를 위해 평가 지표를 사용한 결과를 표 1에 정리하였다. 위치 기반으로 생성된 이미지는 RMSE가 0.543으로 픽셀 기반에서 원본과 가장 유사함을 확인할 수 있었다. Font2Font는 SSIM이 0.954으로 구조가 유사한 이미지를 생성해내는 것을 확인할 수 있었다. 또한 AFAGAN 모델은 SSIM, LPIPS, FID가 각각 0.958, 0.047, 5.155으로, 모델이 원본 이미지와 시각적으로 비슷하고 유사한 구조와 분포를 가진 이미지를 생성하는 결과를 확인하였다. 이를 통해 각 모델이 목적에 맞는 결과를 제공하며, 생성 과정을 통해 손글씨 스타일을 효과적으로 만들어낼 수 있음을 확인할 수 있었다.

6. 결론

본 논문은 이미지 생성모델을 활용하여 단시간에 고품질의 폰트 파일을 제공하는 서비스를 제안하였다. 한글 손글씨 폰트 제작을 위해 글자의 구조적 특징을 보정하는 Font2Font와 글자의 스타일을 보정하는 AFAGAN을 사용하였고 이 두모델은 결과를 통해 글자가 보정되는 것을 볼 수 있었다.

향후에는 중국어와 태국어 등 조합형 다국어의 손글씨 폰트를 생성하는 연구와 함께, 같은 스타일의 영문자 및 숫자, 그리고 특수 기호를 추가하는 연구도 진행할 예정이다.

감사의 글

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학사업의 지원을 받아 수행되었음 (2024-0-00071).

참고문헌

- [1] 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 the International Conference on Neural Information Processing Systems (NIPS), pp. 2672-2680, 2014.
- [2] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1125-1134, 2017.
- [3] Y. Tian, "zi2zi: Master chinese calligraphy with conditional adversarial networks," <https://github.com/kaonashi-tyc/zi2zi>, 2021.
- [4] 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.
- [5] Park, J., Hassan, A. U., & Choi, J. "Few-Shot Korean Font Generation Based on Hangul Composability," KIPS Transactions on Software and Data Engineering, 10(11), 473-482, Nov. 2021.
- [6] Park, J., Hassan, A. U., & Choi, J. "CKFont2: An Improved Few-Shot Hangul Font Generation Model Based on Hangul Composability," KIPS Transactions on Software and Data Engineering, vol. 11, no. 12, pp. 499-508, 2022.
- [7] Avinash Kumar, "Korean Font Image Generation Using Position-Based Components," Master's Thesis, Soongsil University Graduate School, 2024.
- [8] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," CoRR, vol. abs/1506.02640, 2015
- [9] 네이버 클로바, "한글한글아름답게", [Internet] <https://clova.ai/handwriting>, 2019.
- [10] D. H. Ko, A. U. Hassan, S. Majeed, and J. Choi, "SkelGAN: A Font Image Skeletonization Method," Journal of Information Processing Systems, vol. 17, no. 1, pp. 1-13, Feb. 2021.
- [11] Liu, Ming-Yu, Huang, Xun, Mallya, Arun, Karras, Tero, Aila, Timo, Lehtinen, Jaakko, & Kautz, Jan. "Few-Shot Unsupervised Image-to-Image Translation," CoRR, abs/1905.01723, 2019.
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition," CoRR, vol. abs/1512.03385, 2015.
- [13] Liu, Haokun, Tam, Derek, Muqeeth, Mohammed, Mohta, Jay, Huang, Tenghao, Bansal, Mohit, & Raffel, Colin. "Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning". ArXiv, abs/2205.05638, 2022.
- [14] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.
- [15] Zhu, Jun-Yan, Park, Taesung, Isola, Phillip, & Efros, Alexei A. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks". CoRR, abs/1703.10593, 2017.
- [16] 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.