

딥러닝을 활용한 나만의 손글씨 글꼴 생성 및 분석

*조권영 **박구만

서울과학기술대학교 전자IT미디어공학과

*dc05777@naver.com

Making and Analyzing My Handwriting Font Using Deep Learning

*Cho, Gwon-Yeong **Park, gooman

Seoul National University Of Science And Technology

요 약

다양한 분야에서 전자기기들을 사용함으로써 인해 문서를 작성할 때 디지털 글꼴을 통해 작성하게 되는데, 이로 인해 글꼴을 종류가 여러 형태로 증가하면서 다양한 글꼴들을 사용하고 있다. 하지만, 글꼴마다 저작권을 가지고 있어서 마음에 든다고 해서 함부로 사용할 수도 없는 것이 문제점이다. 또한, 한글은 다른 언어에 비해 글자 조합방식이 많아서 폰트로 제작하기엔 많은 시간과 비용이 든다는 문제도 있다. 이러한 문제들을 해결하기 위해서 딥러닝을 통해 글꼴을 제작하게 된다면 적은 글자를 입력해 많은 글자의 결과를 도출함으로써, 시간과 비용을 절감해 효율적으로 만들고자 하였다.

이에 본 논문은 GAN을 기반으로 한 손글씨 폰트 제작을 하는 가운데 글꼴을 만들기 위해 입력에 어떤 글자들이 필요한 지에 대해 연구하였다. 다양한 분석적 요소를 갖고 실험을 하여 입력에 따라 결과가 어떻게 달라지는지를 알아보고 이를 바탕으로 글꼴을 생성하였다.

1. Introduction

컴퓨터를 통해 문서를 작성할 때는 글꼴 선택이 중요하다. 상황에 맞는 글꼴에 따라 글의 분위기가 다르게 느껴지기 때문이다. 그래서 다양한 글꼴을 찾아 사용하지만, 문제는 원하는 글꼴이 없는 경우와 글꼴도 하나의 창작물이기 때문에 저작권을 가지고 있어 원하는 글꼴을 사용할 수 없다는 단점들도 존재한다. 또한, 한글은 다른 언어들에 비해 언어 조합 방식이 많아서 글꼴을 만들기 위해 일일이 작업을 하기에는 많은 시간과 비용이 든다. 이러한 문제들을 해결하기 위해 딥러닝을 활용해 글꼴을 제작하게 되면, 원하는 글꼴을 저작권 문제없이 경제적으로 만들 수 있다. 따라서, 딥러닝 중 GAN을 기반으로 해 210자 정도의 적은 글자 수를 가지고 모든 한글 글자들을 도출할 수 있는 시스템을 만드는데 있어서 다양한 분석적 요소를 설정해 입력에 어떤 글자들이 필요한지에 대해 비교 분석하는 과정을 통해 손글씨 글꼴 생성 시스템을 조금 더 효율적으로 구현할 수 있는 연구를 했다.

2. Model Architecture

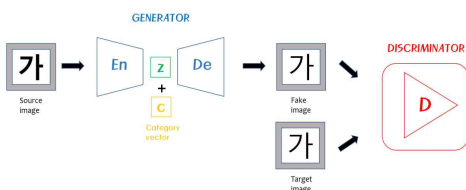


그림2. VAE를 사용해 만든 GAN 최종 모델

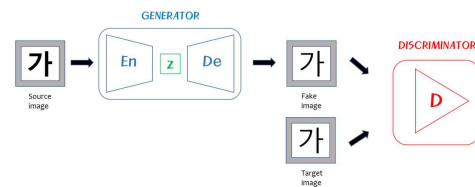


그림1. Auto Encoder를 사용한 GAN 모델

손글씨 글꼴 제작을 위해 GAN에 Source Image를 Input 하기 위해서 GAN의 Generator 부분을 Auto Encoder 형태로 구성한다. 모델은 AutoEncoder와 GAN을 합친 형태(Pix2Pix)로 구성된다. Generator는 여러 폰트 종류의 Target Image에 맞게 Source Image에 약간의 변형을 줘서 같은 글자에 다른 종류의 폰트를 생성하는데 이를 위해서 Generator에 VAE 모델이 필요하다. VAE 모델은 정해진 한 폰트의 글자를 입력으로 넣으면 sampling을 통해 해당 글자에 대한 다양한 폰트의 이미지를 생성하는 generative 모델의 역할을 수행하게 된다. 그래서 VAE 모델은 Encoder를 거쳐 특징이 추출된 z벡터에 Category Vector인 c벡터를 붙여 Decoder에 입력해주는 과정으로 폰트 종류에 따른 새로운 Fake Image를 생성한다.

3. Training Method

모델을 학습시키기 위한 Data Set을 준비한다. Data Set은 30가지 글꼴에, 글꼴마다 2350자를 랜덤으로 선정한 약 70,000장의 이미지를 준비했다. Data Set을 사용할 때 이미지가 (128x128)의 흰 캔버스 중앙에 위치하지 않는 경우와 각자 다른 크기의 이미지로 생성될 수 있는 부분에 대비해 Data Set의 균질화 작업인 Preprocessing(전처리) 과정을 실행한다. Preprocessing(전처리)는 Crop - Resize - Padding 순서대로 실행했다. Pre-Training(사전 학습)모델을 만들기 위해서 Preprocessing(전처리) 과정을 거친 약 70,000장의 Data Set을 200epoch으로 학습시켰다. 다양한 글자들을 학습한 거대한 Pre-Training 모델을 가지고 Encoder는 동결시킨 후, 준비한 손글씨 210자를 학습시키는 Transfer Learning을 실행했다. Transfer Learning은 배치사이즈를 105로 설정해 2,000 epoch으로 돌렸다. Transfer Learning을 실행하는 과정에서 2가지 분석적 요소를 가지고 실험하기 위해 210자의 손글씨는 3종류 구성으로 준비했는데, 첫번째로는 자모음+흡반침+겹반침이 모두 있는 210자, 두번째로는 자모음으로만 구성된 글자 210자, 마지막으로 흡반침으로만 구성된 글자 210자를 준비했다. 이 3가지 손글씨 구성을 갖고 실시한 첫 실험의 분석적 요소는 글자 수가 같은 조건이라면 자모음으로 구성된 글자들과 자모음+흡반침+겹반침이 모두 섞인 글자들을 비교했을 때 얼마나 차이가 나는지에 대한 것이었다. 이는 자모음 글자와 같이 받침이 없는 글자를 입력했을 때, 받침 있는 글자를 도출할 수 있는지를 분석하기 위해 실험했다. 두 번째 실험의 분석적 요소는 글자 수가 같은 조건이라면 흡반침으로만 구성된 글자들과 자모음+흡반침+겹반침이 모두 섞인 글자들을 비교했을 때 겹반침 도출 값이 얼마나 차이 나는지에 대한 것이었다. 이는 테스트 글자를 겹반침으로만 구성하여 흡반침만 입력했을 때도 겹반침의 결과가 잘 도출되는지를 분석하기 위해 실험했다.

4. Results and Discussion



그림3. 자모음 (2000epoch)

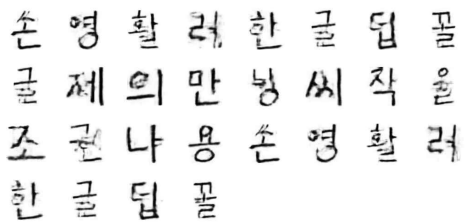


그림4. 자모음+흡반침+겹반침 (2000epoch)

첫 실험의 결과는 Pre-training을 통해 약 70,000자 정도를 학습시켰기 때문에 자모음만 입력했을 때도 받침이 있는 글자 형태가 나오는 걸 그림3을 통해 알 수 있다. 하지만, 그림4의 흡반침과 겹반침까지 같이 있는 글자의 결과와 비교했을 때 자모음만으로 실험한 결과는 손글씨의 특징을 살리지 못했다. 결과가 손글씨에 가깝다기보다는 입력할 때 설정되어있는 고딕체에 더 가까운 것을 볼 수 있었다. 이는 학습이 잘되지 않았다는 뜻이기 때문에, 입력에 자모음만으로 손글씨 글꼴을 만드는 것은 부적절하다는 걸 실험을 통해 확인할 수 있었다.

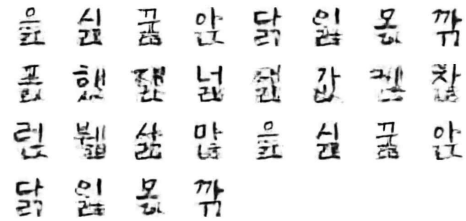


그림5. 흡반침 (2000epoch)

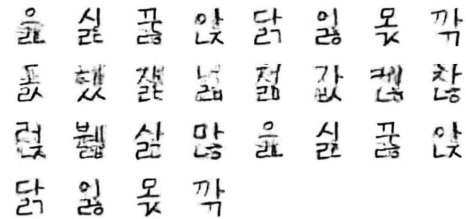


그림6. 자모음+흡반침+겹반침 (2000epoch)

두 번째 실험 결과는 흡반침과 자모음+흡반침+겹반침 모두 받침이 있는 글자라서 손글씨와 비슷한 글꼴이 도출되었지만, 그림5를 보게 되면 흡반침으로만 실험한 결과는 겹반침의 완성도가 떨어져 알아보기 어려웠다. 이는 겹반침을 구현하기 위해선 무조건 겹반침 입력이 필요하다는 뜻이기에 최소한의 겹반침을 입력해줘야 한다는 걸 실험을 통해 확인할 수 있었다.

첫 번째 실험에서 자모음으로만 입력했을 때 손글씨 특징을 살리지 못한 것을 두 번째 실험에서 흡반침만 입력했을 때도 손글씨 특징이 나온 것으로 보아 손글씨의 특징을 살려 글꼴로 만들기 위해선 받침 입력이 꼭 필요하다는 것까지 확인할 수 있었다.

따라서 받침 있는 글자로 입력하되 겹반침을 꼭 넣어줘야 그림4와 그림6의 결과와 같이 손글씨의 특징과 겹반침의 완성도를 높여주는 결과를 도출할 수 있다.

5. Conclusion

본 연구에서는 GAN을 이용하여 나만의 손글씨 글꼴을 만들기 위해 어떤 입력값이 필요한지에 대한 분석과 이를 바탕으로 손글씨 글꼴 생성 모델을 제시하였다. 이를 통해 손글씨 글꼴 생성을 위해 어떤 입력값이 필요한지 확인할 수 있었고, 손글씨로 글꼴 생성이 가능함을 보였다. 효율적이고 좋은 성능을 보여 GAN을 활용한 손글씨 제작이 활성화될 것

으로 보인다. 하지만 본 연구는 시각적 분석에서 그쳤을 뿐 뚜렷한 기준점을 가지고 수치적 분석을 하지 못했다. 따라서 실제 손글씨와 GAN의 결과를 수치를 통해 명확하게 분석할 추가적인 연구가 필요하다.

REFERENCES

- Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. Generative Adversarial Networks. arXiv:1406.2661, 2014.
- Diederik P Kingma, Max Welling. Auto-Encoding Variational Bayes. arXiv:1312.6114, 2013
- GAN based Fonts Generation, 한국컴퓨터정보학회 하계학술대회 논문집 제27권 제2호 (2019. 7)
- "GAN-handwriting-styler", <https://jeinalog.tistory.com/15>