딥러닝 기반 한글 전자 필기 연습 및 분석 앱 개발에 대한 연구

고주은, 오지은, 민경원
이화여자대학교 소프트웨어학부 컴퓨터공학전공
kjunewha@ewhain.net, 0909oje@gmail.com, mkkw2047@gmail.com

A Study on Hangeul Mobile Handwriting Practice and Analyzing Application Development Based on Deep Learning

Ju-Eun Ko, Jee-Eun Oh, Kyoung-Won Min
Dept. of Computer Science and Engineering, Ewha Womans University

요 약

전 세계적으로 코로나바이러스가 유행함에 따라 비대면 활동을 비롯하여 전자 필기 이용 및 상품 소비가 증가하였다. 전자 필기에 대한 수요가 늘어남에 따라 전자 필기 글씨체 교정에 대한 관심 또한 증가하는 추세이다. 본 논문에서는 전자 필기 이미지에서 음절과 음소 영역을 추출하여 글씨를 분석하고, 이를 사용하여 사용자의 손글씨에서 개선점을 찾아낼 수 있는 딥러닝 알고리즘을 제안한다. 제안한 알고리즘을 통해 사용자가 원하는 전자 필기 글씨체를 효과적으로 습득할 수 있도록 사용자 글씨에 대해 구체적인 피드백을 제공하는 딥러닝 기반 태블릿 PC 용 한글 전자 필기 연습 및 분석 앱에 대한 연구를 소개하였다.

1. 서론

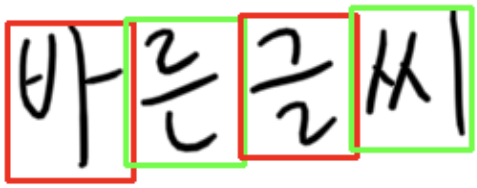
전 세계적으로 코로나바이러스가 유행함에 따라 비대면 강의, 온라인 수업 등이 증가하였다. 미국 시장조사업체 IDC 에 따르면 태블릿 PC 출하량은 2021 년 글로벌 태블릿 PC 출하량은 1 억 6880 만대로 2020 년 대비 3.2% 증가했다.[1] 전자 필기 이용 및 상품 소비가 증가하였고[2] 전자 필기 증가에 따라 “예쁘게” 전자 필기하는 것에 대한 관심도 빠르게 증가하였다. Z 세대에 태블릿 PC 를 이용한 전자 다이어리 꾸미기가 유행하면서 전자 다이어리 속지 등 전자 다이어리 관련 상품 시장이 커지고 있다.[3] YouTube 와 같은 플랫폼에 아이패드 등 태블릿 PC 에서 글씨를 잘 쓰는 방법과 같은 전자 필기 글씨체 교정 관련 콘텐츠들이 100 만 이상의 높은 조회수를 보유하고 있으며 영상의 댓글엔 자신의 전자 필기 글씨체의 교정의 어려움을 호소하는 반응이 다수 존재한다. 기존 서비스를 조사한 결과, 교본만을 제공하여 사용자가 교본 위에 따라 쓰면서 연습할 수 있는 앱과 교본을 제공하고 교본의 글씨와 사용자의 글씨의 대략적인 유사도 점수를 제공하는 서비스가 존재하였다. 하지만 기존 서비스에는 공통적으로 사용자 글씨에 대한 구체적인 피드백을 제공하지 않아 사용자의 글씨 교정에 효과적이지 않다고 판단하였다. 이에 본 연구는 딥러닝을 통해 한글 전자 필기의 형태를 분석하여 사용자 글씨에 대한 구체적인 피드백을 생성해 사용자가 원하는 한글 글씨체를 가질 수 있도록 도와주는 딥러닝 기반 한글 전자 필기 연습 및 분석 앱 ‘바른글씨’를 연구 및 개발하였다.

2. 관련 연구

2.1. CRAFT 를 이용한 한글 음절 영역 추출

본 연구에서는 줄글 형태로 쓰여진 사용자 글씨 이미지를 한글 음절 단위로 분류하기 위해 CRAFT(Character Region Awareness for Text Detection)[4]을 사용하였다. CRAFT 는 기존 단어 단위로 학습하던 모델의 한계를 해결하고 문자 단위의 detection 정보를 활용하여 이미지에서 텍스트 영역을 찾아내는 CNN 기반의 텍스트 검출 모델이다. CRAFT 는 Region score 로 이미지에서 각각의 문자들을 지역화하고 Affinity score 를 통해 문자들을 하나의 단어 객체로 그룹화함으로써 다양한 모양의 텍스트들을 검출해낼 수 있다.

본 연구에서는 NAVER CLOVA 팀에서 제공하는 SynthText, IC15, 그리고 IC17 데이터셋으로 훈련한 General 모델을 사용하여 사용자 글씨 이미지를 음절 단위로 분리하였다.

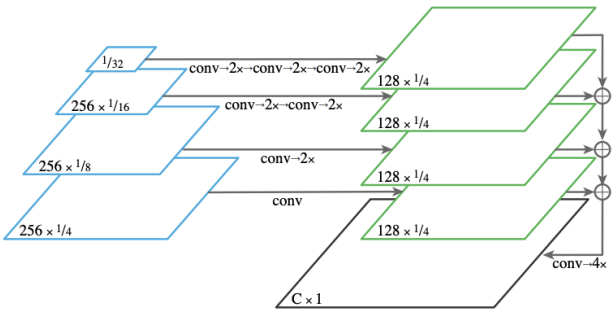


(그림 1) CRAFT 를 이용해 추출한 한글 음절 이미지 예시.

2.2. 데이터셋 수집

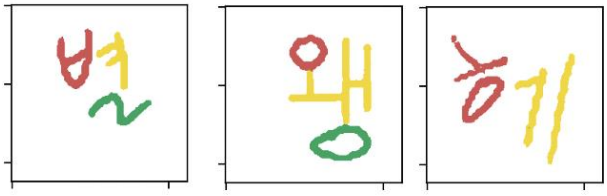
한글 음소 감지 모델 학습을 위해 한글 음절에 대해 자음(초성), 모음(중성), 받침(종성) 3 개의 클래스로 분류하여 데이터를 수집했다. 한글 음절 이미지 데이터 생성은 NAVER CLOVA 의 나눔손글씨 글꼴 13 종을 이용하였으며 각각 초성 19 자, 중성 21 자, 종성 27 자에 대한 이미지를 생성하였다. 이후 초성, 중성, 종성 이미지에서 세 클래스에 대한 마스크를 생성한 뒤 이를 요소 이미지로 사용하여 KS X 1001 완성형에 포함된 현대 한글 2350 자에 대해 한글 음절 이미지를 생성하였다. 초성, 중성, 종성 이미지를 합칠 때 데이터 증강 기법을 활용하여 각 요소에 변화를 주었으며 위와 같은 과정으로 음소로 나뉘어진 한글 음절 이미지 데이터셋 30,550 장을 제작하였다.

2.3. FPN 을 이용한 한글 음소 영역 추출



(그림 2) FPN 아키텍처[5].

semantic segmentation 은 이미지내에서 픽셀 단위로 해당하는 클래스로 분류하는 것을 목적으로 하며 이미지에 존재하는 각 객체들의 경계 및 영역을 추출하여 자율주행자동차, 의료 이미지 분석 등 다양한 분야에서 활용되고있다. 본 연구에서는 FPN(Feature Pyramid Network)[6]을 이용하여 한글 음절 이미지의 각 픽셀을 자음, 모음, 받침 3 개의 클래스로 분류하였다. FPN 은 top-down 방식으로 특징을 추출하고 lateral connection 을 사용하여 고해상도와 저해상도 특징 맵을 결합한다. 각 레벨에서 독립적으로 특징을 추출하여 하나의 이미지에서 1/32 에서 1/4 까지의 특징 피라미드를 얻어 객체를 탐지하는데, 이때 각 피라미드 레벨은 같은 채널을 가진다. 이를 통해 FPN 은 단일 크기의 이미지를 입력하여 서로 다른 크기의 특징 맵을 출력하기때문에 더 높은 감지 성능을 보이고, 속도와 메모리면에서 효율적이다. 이러한 과정으로 추출된 각 특징들을 convolution 으로 upsampling 하고 이에 pixel-wise summation 을 적용하여 픽셀 단위의 semantic segmentation 결과를 얻을 수 있다. 본 연구에서는 resnext50 을 backbone 으로 한 FPN 기반 모델을 한글 음절 이미지 데이터셋으로 학습하였고, 학습 결과 테스트 데이터셋 3,055 장에 대해 96%의 정확도를 보였다.



(그림 3) 손글씨 이미지에 대한 예측 결과.

2.4. 폰트 글씨체와 사용자 글씨 비교

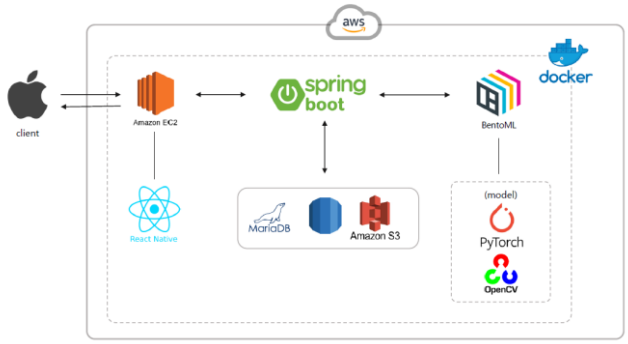
손글씨 필기는 각 문자와 문자를 구성하는 음소들의 크기, 위치, 기울기를 분석하여 그 모양을 알 수 있다. 본 연구에서는 사용자가 지정한 폰트 글씨와 비슷하게 글씨를 쓸 수 있도록 피드백을 제공하는 것을 목적으로 하고 있어 폰트 글씨체의 특징과 사용자 글씨의 기하학적 특징을 각각 비교·분석하여 둘의 유사성을 중심으로 글씨가 어떻게 쓰였는가를 판단하였다. 비교 과정은 글에서 음절들의 크기와 중심점들이 이루는 기울기가 균일한가를 통해 글의 전체적인 모양을 판단하고, 각 음절들을 구성하는 음소(초성, 중성, 종성)들에서 추출한 바운딩박스의 크기 비율, 각 원소들 간의 상대적 거리가 폰트 글씨체를 기준으로 한 한글의 황금비율을 충족하는가를 검사하는 것으로 이루어진다.

3. 설계 및 구성

3.1. 개발환경

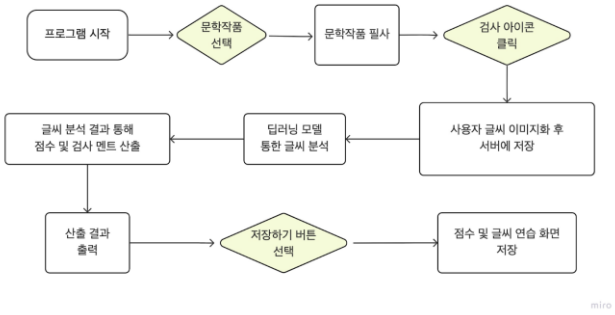
(그림 4) 은 앱의 개발환경이다. 앱은 React Native, 서버는 Spring Boot 로 개발하였다. AWS RDS 와 S3 를

이용하여 데이터베이스를 구현하였고 AWS EC2 에 서버를 구축하여 배포하였다. 딥러닝 서빙은 BentoML, 모델 구현은 PyTorch 와 OpenCV 를 사용한다.



(그림 4) 개발환경.

3.2. 흐름도



(그림 5) 순서도.

(그림 5)는 앱의 프로그램 순서도이다. 사용자가 앱을 실행한 후 필기 연습을 진행하고 싶은 문학 작품을 선택해 해당 문학 작품 화면으로 이동한다. 이 화면에서 사용자는 해당되는 문학작품의 필사를 마치고 검사 아이콘을 누른다. 이 때 앱은 사용자가 필사한 내용을 이미지화하여 웹서버로 넘겨준다. 웹서버는 글씨 이미지를 일차적으로 저장한 후 딥러닝 서버로 넘겨준다. 딥러닝 서버는 글씨 이미지를 통해 주어진 예시 문장 내 음절과 대응되는 사용자의 문장 내 음절을 찾아 두 음절의 기하학적 특징을 비교하여 글씨 분석을 진행한다. 서버는 글씨 분석 결과를 받아와 사용자의 글씨 점수와 피드백 할 부분을 산출한 후 이 결과를 앱에 보내준다. 앱은 이 결과를 바탕으로 사용자의 글씨 점수와 검사 피드백 멘트를 화면에 출력한다. 이후 저장하기 버튼을 누르면 사용자의 점수와 이미지화 했던 필사 내용을 웹서버에 저장하고 이후 사용자는 저장한 결과를 내 서랍 화면에서 확인할 수 있다.

3.3. 시스템 기능

(그림 6) 글씨 연습 화면.

(그림 6)은 사용자가 문학작품 필사를 진행할 수 있는 글씨 연습 화면이다. 초기 화면에서 사용자는 필사하고 싶은 문학작품을 선택하면 (그림 6)과 같은 화면으로 이동할 수 있다. 이 화면에서 사용자는 문학작품 필사를 진행하고 글씨 검사를 시작하기 위해 상단의 돋보기 아이콘을 클릭한다.

(그림 7) 글씨 분석 결과 화면.

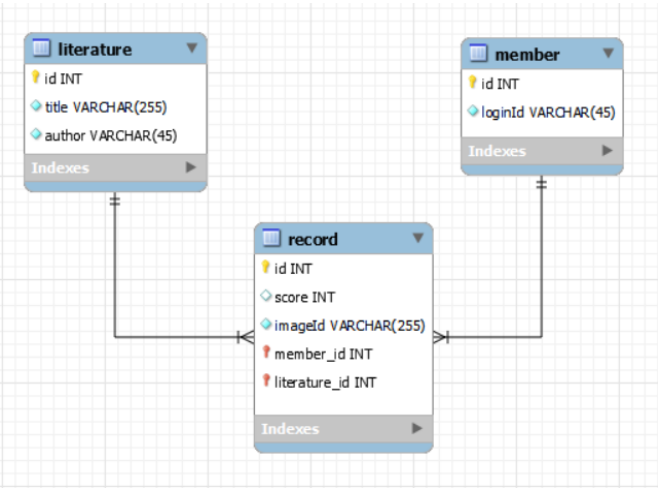
(그림 8) 검사 멘트 모달.

(그림 7)는 사용자가 (그림 6)에서 돋보기 아이콘을 클릭했을 때 보여지는 글씨 검사 결과 화면이다. 사용자가 돋보기 아이콘을 클릭하면 사용자가 필사한 내용이 웹서버를 거쳐 딥러닝 서버로 넘겨지게 되고 딥러닝 서버는 사용자의 글씨 분석 결과를 다시 웹서버를 거쳐 앱으로 전달해준다. 앱은 이 결과를 각각 점수와 느낌표 마크로 화면에 출력한다. 이 때 느낌표 마크를 클릭하면 검사 멘트가 (그림 8)와 같이 모달창 형식으로 화면에 출력된다.

(그림 9) 점수 그래프 화면.

(그림 7)에서 사용자가 저장하기 버튼을 누르면 서버에 사용자의 글씨 점수와 글씨 이미지가 저장되고 사용자는 이를 (그림 9)와 같이 자신의 점수 추이를 그래프 형식으로 확인할 수 있다.

3.4. 데이터베이스



(그림 20) E-R 다이어그램.

본 앱은 AWS EC2 로 운영체제가 Linux 인 서버를 구축하였다. EC2 에 AWS RDS 에 MariaDB 로 회원 정보, 문학작품 정보, 기록들을 저장하였고 AWS S3 에 앱에서 전달받은 이미지를 저장하여 이미지 조회 및 딥러닝 서버에 전달하는 것에 사용하였다. (그림 10)는 본 앱의 데이터베이스의 E-R 다이어그램이다. literature 테이블에 문학작품 정보를 저장하였고, member 테이블에 회원 정보를 저장하였다. record 테이블에는 회원 Id 와 문학작품 Id 와 S3 에 저장된 이미지의 이름과 해당 이미지에 대하여 딥러닝 서버로부터 받은 점수를 저장하였다.

4. 결론

본 연구에서는 한글을 이루는 원소들의 크기, 위치와 같은 데이터를 통해 추출한 사용자 글씨와 폰트 글씨의 유사도를 기반으로 글씨를 분석하였으며 딥러닝을 기반으로 한글의 음절 및 음소 영역을 추출하여 서비스의 정확도를 높였다. 본 논문에서 소개한 딥러닝 기반 한글 전자 필기 연습 및 분석 앱을 이용하여 사용자는 본 앱에서 제공하는 구체적인 피드백을 통해 사용자는 효과적으로 글씨체를 습득할 수 있을 것으로 보인다. 본 연구는 현재 한글 구성요소의 크기, 기울기, 위치관계에 기초하여 한글의 형태를 추출하고 이를 분석하여 사용자의 글씨체가 잘 쓰여졌는가를 판단하고있으나 이러한 분석 과정은 한글의 기하학적 지식을 바탕으로 개선시킬 수 있는 여지가 있다. 이와 같은 분석 기술을 바탕으로 글씨 학습에 대한 교보재 형태로의 확장 또한 기대할 수 있다.

참고문헌

[1] 아시아타임즈, “‘글로벌 태블릿 PC 大戰’ 삼성전자 VS 애플, 승자는?”, https://www.asiatime.co.kr/article/20220401500075

[2] 한국일보, “디지털 필기상품 전문… 작년 거래액 9배↑”, https://m.hankookilbo.com/News/Read/A2022032812090003320

[3] 한국연예스포츠신문, “나만의 것이 좋아! Z 세대에 부는 커스터마이징 열풍”, http://www.koreaes.com/news/articleView.html?idxno=360363

[4] Youngmin Baek et al. "Character Region Awareness for Text Detection" arXiv preprint arXiv, 1904.01941, 2019

[5] Tsung-Yi Lin1 et al. "Feature Pyramid Networks for Object Detection" arXiv preprint arXiv, 1612.03144, 2017

[6] Alexander Kirillov et al. "Panoptic Feature Pyramid Networks" arXiv preprint arXiv, 1901.02446, 2019