AI 개발을 위한 노 코드 플랫폼의 개발 방향

신유진 양희진 장다영 장현준 *고석주 **한동희
경북대학교 컴퓨터학부 **데이터센트릭
syouj98@naver.com yangjinee@naver.com jdyoung1002@naver.com
dngh0212@naver.com *sjkoh@knu.ac.kr **handh@datacentric.kr

The direction of development of the no code platform for AI model development

Yujin Shin Huijin Yang Dayoung Jang Hyeonjun Jang
*Seokju Koh **Donghee Han
Kyungpook National University **DataCentric

요 약

4차 산업혁명이 시작된 이래로 다양한 산업 분야에서 AI가 활용되고 있고, 그 중에서도 컴퓨터 비전 분야에서 딥러닝 기술이 각광받고 있다. 하지만 딥러닝 기술은 높은 전문 지식이 요구되어 관련 지식이 없는 일반인들은 활용하기 어렵다. 본 논문에서는 AI 관련 배경지식이 없는 사용자들도 UI를 통해 쉽게 이미지 분류 모델을 학습시킬 수 있는 노 코드 플랫폼에 관하여 기술하고, django 프레임워크를 이용해 웹 개발과 딥러닝 모델 학습을 통합 개발을 위한 아키텍처와 방향성을 제시하고자 한다. 사용자가 웹서버에 업로드한 이미지들을 웹 인터페이스를 통해 라벨링 하여 학습 데이터를 생성한 후, 이 데이터를 사용하여 모델을 학습시킨다. CNN 모델에 데이터를 학습시키는 과정과 생성된 모델 기반으로 이미지 예측하는 모듈을 통해 전문지식이 없는 사용자가 딥러닝 기술에 대해 쉽게 이해하고 이용하는 것을 기대할 수 있다.

1. 서론

제 4차 산업혁명이 시작된 이래로 다양하고 많은 기술이 쏟아져 나오고 있다. 그 중에서도 4차 산업혁명의 핵심이라고 할 수 있는 AI는 국방, 의료, 교육, 게임 등 다양한 산업 분야에서 활용되고 있다. 그중 컴퓨터 비전 분야의 딥러닝 기술이 많은 주목을 받고 있다. 연구개발특구진흥재단에 따르면 딥러닝 시장은 2018년 31억 7,600만 달러에서 연평균 성장률 41.7%로 증가하여, 2023년에는 181억 5,640만 달러에 이를 것으로 전망된다[1].

하지만 AI 개발의 경우 상당한 전문 지식을 요구하기 때문에 국내 기업에서는 적합한 기술을 보유한 전문 인력을 고용하는 데 가장 큰 어려움을 겪고 있다고 한다[2]. 그렇기 때문에 인공지능을 개발하기 위해 큰 배경지식이 없는 일반인들도 쉽게 개발할 수 있도록 하는 노 코드 플랫폼의 필요성이 대두된다. 노 코드 플랫폼을 이용하면 개발자의 부담을 줄이고 전문 지식이 없는 일반인들의 AI에 대한 진입 장벽을 낮출 수 있다.

본 논문에서는 웹 UI를 통해 직접적인 코딩 없이 사용자가 업로드한 이미지를 라벨링을 하여 미리 서버에 구현해 놓은 CNN(Convolutional Neural Network)모델을 학습시켜 이미지를 분류해주는 AI개발 노 코드 플랫폼에 관하여 기술하고 웹과 딥러닝 모델의 통합 개발을 위한 아키텍처와 방향을 제시한다.

본 논문의 구성은 다음과 같다. 2절에서는 AI 노 코드 플랫폼의 전반적인 기능과 DB에 대해 설명하고 3절에서는 이미지 분류 모델의 학습과 예측, 웹과 딥러닝 모델의 통합개발에 대해 서술한다. 마지막으로 4절에서는 본 논문에 대한 결론을 내린다.

2. AI 노 코드 플랫폼

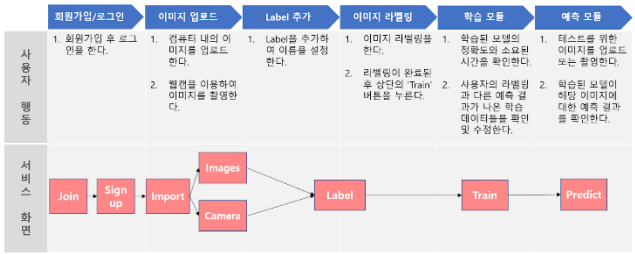
2.1 사용자 시나리오

본 플랫폼은 사용자가 이미지를 업로드한 후 지정한 label로 분류할 수 있는 딥러닝 모델을 제공해준다. 해당 플랫폼을 구현하기 위한 세부적인 기능들을 서비스 화면 기준으로 정의하였다.

사용자에게 보여지는 페이지는 크게 회원가입/로그인 페이지, Label 페이지, Train 페이지, Predict 페이지이다. 사용자가 회원가입 후 로그인을 하면 Label 페이지로 넘어가게 된다. Label 페이지에서는 모델 학습을 위한 이미지들을 업로드 또는 웹캠으로 촬영한 후 라벨링 한다. 사용자는 label을 추가하여 이름을 설정할 수 있다. 사용자가 이미지에 대한 라벨링을 끝내고 상단의 'Train' 버튼을 누르면, 웹 서버에선 해당 이미지들을 CNN 모델에 학습시킨다.

모델 학습 완료 후, Train 페이지로 넘어가게 된다. Train 페이지에서는 딥러닝 모델의 학습 상황을 보여준다. 모델에 학습시킨 이미지에 대한 학습 결과가 표시되고, 사용자가 지정한 label과 다른 결과가 나온 이미지를 확인 및 수정할 수 있다.

Predict 페이지에서는 학습된 딥러닝 모델을 테스트를 할 수 있다. 사용자가 새로운 이미지를 업로드 또는 촬영한 후 학습된 모델을 이용해 결과를 예측하여 인터페이스에 띄워준다. 해당 과정을 통해 사용자는 딥러닝 학습 모델의 성능을 평가할 수 있다.

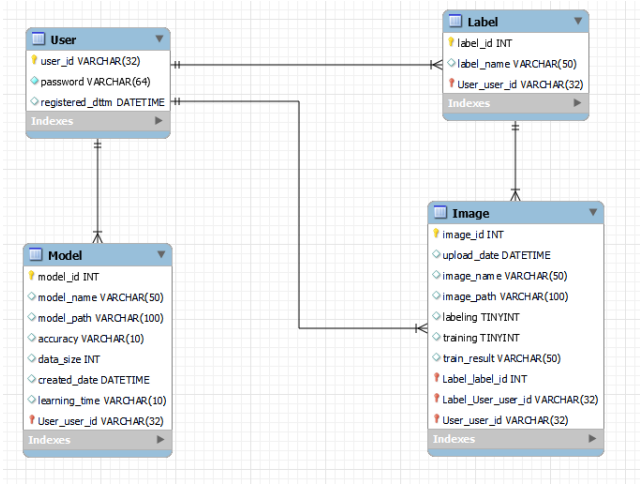


[그림1] 서비스 화면 기준 사용자 시나리오

2.2 노 코드 플랫폼을 위한 DB 설계

이미지 학습 AI 노 코드 플랫폼을 구현하려면 필요한 데이터들은 사용자 정보, 이미지 정보, 이미지의 라벨링 정보, 학습 모델 정보 총 4가지이다. 따라서 RDBMS인 PostgreSQL을 활용해 User, Image, Label, Model 총 4가지 table들을 생성하였다. User에서는 사용자의 아이디, 암호화된 비밀번호를 관리하고, Image에서는 사용자가 웹서버에 업로드한 이미지의 이름, 경로, 라벨링 여부 등을 관리하도록 하였다. 또한 Label에서는 사용자가 이미지 분류 모델 학습을 위해 필요한 Label 이름을 관리할 수 있게 하였고, Model에서 학습 데이터 모델 파일 이름, 경로와 정확도, 학습 시간을 관리하였다.

학습 모델 경우, 사용자가 해당 플랫폼에 접속할 때마다 이미지들을 학습시켜 결과를 확인하지 않고 재접속해도 학습 결과가 유지되게 해야 하므로, 사용자가 딥러닝 학습을 실행할 때마다 모델의 아키텍처와 가중치를 파일로 저장하도록 하였다. 따라서 새로운 이미지에 대해 클래스 예측을 할 땐 해당 모델 파일을 불러와 예측을 할 수 있도록 하였다. 데이터들의 관계를 나타낸 ERD(Entity Relationship Diagram)는 다음과 같다.

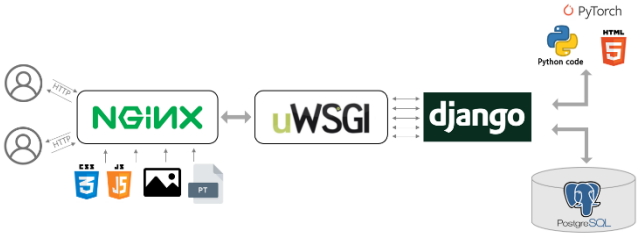


[그림2] ERD

User와 Model table을 1:n 관계로 구축 하였으므로 한 회원이 여러 프로젝트를 생성해 딥러닝 학습을 진행하는 경우도 구현 가능하다. 이 경우 웹 서버에서 관리해야 할 이미지들이 기하급수적으로 증가하기 때문에, 별도의 이미지 서버를 구축하는 것을 권장한다.

2.3 통합 개발을 위한 아키텍처

딥러닝 모델 학습과 웹을 하나의 웹 서버에서 통합 개발하고자 하였다. 따라서 Python 기반 프레임워크인 Django를 사용해 딥러닝 학습 모듈과 웹 개발 언어를 python으로 통일하였다. 딥러닝 서버와 웹 서버를 따로 분리하지 않음으로써 개발 기간을 최소화할 수 있었고 서버 구축에 대한 부담을 줄일 수 있었다. 웹 서버 소프트웨어로 Nginx를 사용하였고, 웹 서버에서 이미지와 학습 모델 파일을 관리한다. 웹 서버와 Django 간의 연결을 위해 웹 어플리케이션 서버인 uWSGI가 필요하다. 마지막으로 django에서 클라이언트 요청에 따라 응답을 처리하고 딥러닝 학습 모듈을 실행하도록 하였다.



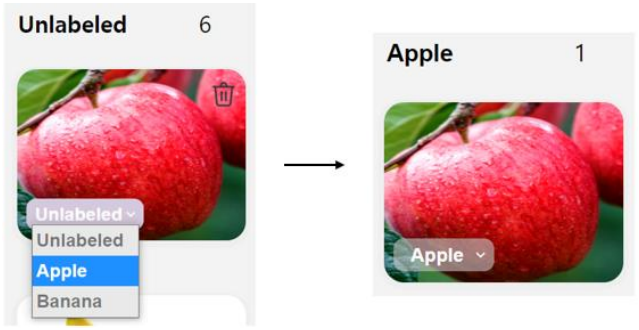
[그림3] 웹 서버 구조

2.4 사용자의 편의를 고려한 UI

본 플랫폼은 사용자가 쉽고 편하게 이미지를 분류할 수 있는 방법들을 고안하여 웹 UI 를 구현하였다.

먼저 Label 의 이름을 수정할 때 엔터키를 누르면 값이 전달되는 방식을 적용하였다. 기존 텍스트 입력 방식은 텍스트를 입력 후 submit 버튼을 직접 눌러 주어야 값이 전달됐다. 하지만 본 플랫폼은 텍스트를 입력 후 따로 마우스를 움직일 필요 없이 엔터키를 누르면 웹 서버에 값이 전송되도록 구현하였다.

다음으로 이미지 라벨링을 select box 로 구현하였다. select box 를 이용해 이미지에 텍스트를 입력하는 방식이 아닌 마우스 클릭을 하는 방식을 적용하여 사용자가 보다 쉽고 편하게 이미지 분류를 할 수 있도록 하였다.



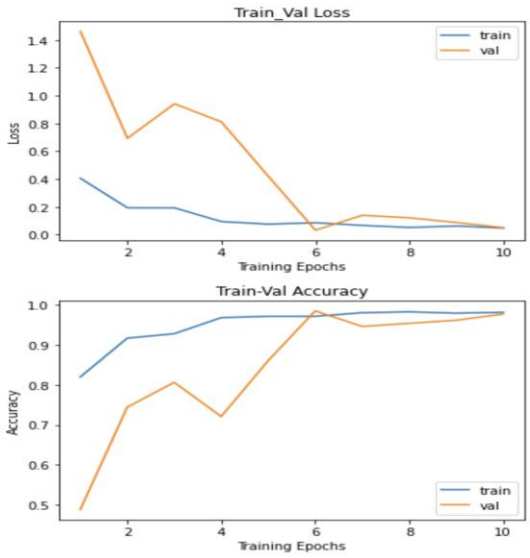
[그림4] 'Apple'로 이미지 라벨링

이미지 목록에서는 사용자가 라벨링 한 이미지들과 아직 라벨링 하지 않은 이미지들을 띄워준다. 사용자가 이미지 라벨링을 모두 완료하면, 상단의 'Train' 버튼을 눌러 만들어 놓은 데이터 셋으로 모델을 학습시킬 수 있다.

3. 딥러닝 모델

3.1 모델 선택 및 학습방안

딥러닝 모델 중 CNN 모델은 필터링 기법을 인공 신경망에 적용하여 이미지를 분류하는데 적합한 모델이다. 본 연구에서는 개와 고양이, 말과 사람 이미지 데이터셋을 이용하여 CNN모델 중 PyTorch 기반의 여러 모델들을 테스트하여 딥러닝 모델을 선정하였다. 모델 선정 후보군으로 GoogLeNet, ResNet, SENet, EfficietnNet이 있었으며 개와 고양이, 말과 사람 등 약 1,000장의 준비된 이미지 데이터셋을 모델에 학습시켜 정확도와 속도를 중점으로 비교해 최종 모델을 선정하였다[그림 5].



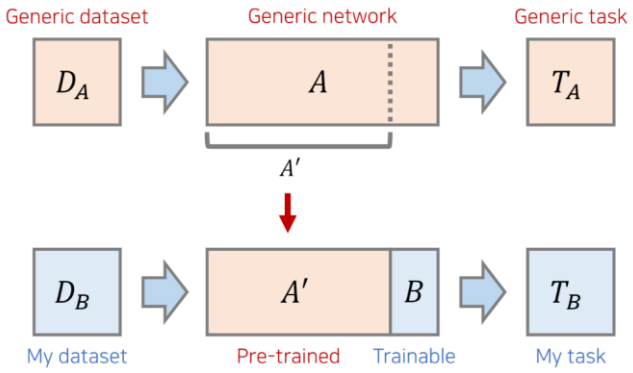
[그림5] ResNet 모델의 학습 Loss와 Accuracy 값

정확도와 속도를 고려하여 ResNet 모델을 최종 딥러닝 모델로 선정하였고 이미지 데이터셋의 분류에는 정확성이 충분하고 웹서버와 연동 시 빠르고 효율적으로 딥러닝 기술을 서비스하기 위해 미리 학습된 여러 ResNet 모델 중 가장 가벼운 ResNet18 모델을 최종적으로 선정하여 이미지 데이터셋 분류 및 예측에 사용하였다.

모델의 학습과 예측 성능 향상을 위해 데이터 전처리 과정으로 이미지를 불러올 때 224*224 크기로 해상도 크기를 맞추고 데이터 증진을 위해 RandomHorizontalFlip 으로 확률적으로 이미지 데이터를 좌우반전 한 뒤 Normalize 를 이용하여 데이터 정규화를 해주었다.

모델의 학습방법으로 전이학습방법을 사용하였다. 미리 학습된 ResNet18 의 네트워크를 다운로드 하여 앞쪽 레이어를 가져와 학습이 필요한 데이터셋을 다시 학습함으로써 학습과 예측을 하게 하였다[그림 6]. 모델을 불러온 뒤 모델의 가장 마지막 레이어만 별도의 Linear 레이어로 바꾼 뒤 학습할 이미지 데이터셋을 입력하고 학습을 진행하였다. 사용자가 학습시킬 이미지 데이터셋의 클래스 개수와 마지막 출력 클래스의 개수가 동일해야 한다. 따라서 가져온 모델의 출력 뉴런 수를 이미지셋의 클래스 수와 같게 교체하여 학습을 수행할 수 있도록 하였다. 학습횟수는 총 50 epoch 로 설정하였고

정확도를 90%로 설정하여 학습도중 목표 정확도에 도달하게 되면 학습을 중단하게 설정하였다.



[그림6] 전이 학습 구성도

모델학습이 완료된 뒤 학습된 모델에 예측할 이미지 파일을 배치 값만큼씩 넣어 예측하고 이미지 파일 하나당 예측 결과와 실제 결과 그리고 전체 예측결과에 대한 정확성을 확인할 수 있다.

3.2 이미지 분류 모델 인터페이스

이미지 분류 모델 학습을 위해서는 라벨링 된 이미지 데이터셋이 필요하다. 사용자가 원하는 이미지를 분류하는 것이 목적이므로 DB의 Image table에서 현재 사용자의 ID에 해당되는 Image객체를 가져온다. 가져온 객체에서 사용자가 업로드한 이미지 중 라벨링 된 이미지와 그 이미지에 대응되는 Label로 데이터셋을 만들어 입력 데이터로 사용하여 모델을 학습시킨다.

그 다음 학습된 모델을 저장한다. 모델 파일은 웹 서버에 저장되고 DB의 Model table에 학습된 모델의 정보가 저장된다. 여러 모델이 있을 수 있으므로 모델 구별을 위해 사용자의 ID 이용하여 모델의 저장될 이름을 명명한다.

모델이 제대로 작동하고 있는지 학습하는 과정을 직접적으로 볼 수 없기 때문에 모델을 모니터링하는 것은 중요하다. 학습 과정 중 한 epoch마다 Loss, Accuracy, Time을 출력하여 현재 모델의 학습 상황을 확인할 수 있도록 하였다.

```
#0 Loss: 0.4489 Acc: 79.3532% Time: 8.1616s
#1 Loss: 0.1159 Acc: 95.0249% Time: 15.5297s
#2 Loss: 0.1029 Acc: 97.0149% Time: 22.7854s
#3 Loss: 0.0603 Acc: 97.7612% Time: 30.0000s
#4 Loss: 0.0875 Acc: 97.0149% Time: 37.1481s
```

[그림 7] 모델 학습 과정 모니터링 출력 데이터 예시

마지막으로 사용자 ID를 통해 DB에서 Model table에 Model 객체를 읽은 뒤, 이 객체를 이용하여 해당 모델 파일을 웹 서버에서 불러온다. 그 후 사용자가 업로드한 분류할 이미지를 입력 데이터로 받아 데이터셋을 만들어 웹에서 바로 예측한다. 사용자가 설정한 여러 개의 Label로 name class를 만들어 예측한 결과에 해당되는 class의 값을 출력한다.

4. 결론

현재 컴퓨터 비전 분야가 확대되고 있고, '마켓앤마켓'은 인공지능 시장에서 컴퓨터 비전 시장이 2026년까지 26.3% 성장할 것으로 내다보았다 [3]. 하지만 딥러닝 기술은 높은 전문지식을 필요로 하기 때문에 일반인들이 접근하기 어렵다는 문제점이 있다. 따라서, 본 논문에서는 이미지 분류 모델을 UI를 통해 쉽게 학습시킬 수 있는 노 코드 플랫폼을 제시하였고, Django를 이용한 웹 개발과 딥러닝 모델 학습을 통합하여 개발할 수 있는 방향과 구조에 대해 기술하였다.

해당 노 코드 플랫폼을 통해 사용자는 쉽게 이미지 분류 모델 학습 과정을 수행하고 결과를 시각적으로 파악할 수 있을 것이다. 점차 확대되는 노 코드 플랫폼 시장과 딥러닝 기술의 수요 증가로 노 코드 플랫폼을 통한 딥러닝 기술의 활용이 발전할 것으로 예상된다.

현재 개발한 플랫폼에서는 웹 서버의 용량 문제로 사용자 1명 당 하나의 딥러닝 학습 공간을 지원한다. 따라서 대용량 이미지 서버를 추가로 구축해, 하나의 사용자가 여러 프로젝트를 생성할 수 있도록 구현할 예정이다.

Acknowledgement

"본 연구는 과학기술정보통신부 및 정보통신기획평가원의 sw중심대학사업 지원을 통해 수행되었음"(2021-0-01082)

5. 참고문헌

[1] 연구개발특구진흥원, "글로벌 시장 보고서", pp.6, 2021.

[2] 산업연구원, "기업의 AI 도입 및 활용 확대를 위한 정책과제", 제 105호, p. 7, 2021

[3] marketsandmarkets, "AI in Computer Vision Market With Covid-19 Impact by Component, Machine Learning Models, Function, Application (Industrial, Non-Industrial), End-Use Industry (Security & Surveillance, Consumer Electronics) and Geography - Global Forecast to 2026", 2021