

<https://doi.org/10.7236/JIIBC.2021.21.3.43>
JIIBC 2021-3-7

딥블록: 웹 기반 딥러닝 교육용 플랫폼

DeepBlock: Web-based Deep Learning Education Platform

조진성*, 김근모*, 고현민**, 김성민**, 김지섭**, 김봉재***

Jinsung Cho*, Geunmo Kim*, Hyunmin Go**,
Sungmin Kim**, Jisub Kim**, Bongjae Kim***

요약 최근 인공지능을 사용한 연구나 기업의 프로젝트가 활발하게 이루어지고 다양한 서비스나 시스템이 인공지능 기술과 접목되어 점점 더 지능화되고 있다. 이에 따라 인공지능의 기법 중 하나인 딥러닝에 대한 관심과 이를 학습하려는 사람들이 증가했다. 딥러닝을 학습하기 위해서는 딥러닝 이론 이외에도 컴퓨터 프로그래밍, 수식 등 많은 지식들이 요구된다. 이는 초심자에게 높은 진입장벽으로 작용한다. 따라서 본 연구에서는 초심자가 프로그래밍 및 수식 등을 고려하지 않고 DNN, CNN 등과 같은 딥러닝의 기본적인 모델을 구현할 수 있는 DeepBlock이라는 웹 기반 교육용 딥러닝 플랫폼을 설계 및 구현하였다. 제안한 DeepBlock을 이용하여 딥러닝에 관심을 가진 학생들이나 초심자들의 교육에 활용이 가능하다.

Abstract Recently, researches and projects of companies based on artificial intelligence have been actively carried out. Various services and systems are being grafted with artificial intelligence technology. They become more intelligent. Accordingly, interest in deep learning, one of the techniques of artificial intelligence, and people who want to learn it have increased. In order to learn deep learning, deep learning theory with a lot of knowledge such as computer programming and mathematics is required. That is a high barrier to entry to beginners. Therefore, in this study, we designed and implemented a web-based deep learning platform called DeepBlock, which enables beginners to implement basic models of deep learning such as DNN and CNN without considering programming and mathematics. The proposed DeepBlock can be used for the education of students or beginners interested in deep learning.

Key Words : Artificial Intelligence, Block Coding, Cloud Service, Deep Learning, Education Platform

1. 서론

인공지능(Artificial Intelligence)은 4차 산업혁명의 핵심 기술이다. 인공지능을 구현하는 방법 중 하나인 딥

러닝(Deep Learning) 기법은 2016년 이세돌과 알파고의 바둑대전 이후로 관심도가 급증했다. 이에 따라 인공지능에 대한 기대치가 증가하고 있다. 딥러닝은 현재 기업이나 다양한 연구 등 어느 한 곳에 국한되지 않고 다양

*준회원, 충북대학교 전기·전자·정보·컴퓨터공학부

**준회원, 선문대학교 컴퓨터공학부

***정회원, 충북대학교 컴퓨터공학과

접수일자 2021년 3월 29일, 수정완료 2021년 4월 29일
게재확정일자 2021년 6월 4일

Received: 29 March, 2021 / Revised: 29 April, 2021 /

Accepted: 4 June, 2021

***Corresponding Author: bjkim@chungbuk.ac.kr

Department of Computer Engineering, Chungbuk National University, Korea

하게 활용되고 있으며 다양한 문제를 해결하고 있다 [1][2][3].

딥러닝에 대한 세간의 관심이 높아짐에 따라 이를 공부하고자 하는 전공자 및 비전공자의 수가 급격히 증가하였다. 하지만 최적의 딥러닝 모델을 구현하기 위해서는 기본적인 딥러닝 이론 뿐 만 아니라 컴퓨터 공학의 전반적인 전공지식과 많은 수학 지식이 필요하다. 또한 프로그래밍에 대한 전반적인 이해가 필요하기 때문에 비전공자가 딥러닝을 공부한다는 것은 결코 쉬운 일이 아니다. 따라서 본 연구에서는 초심자의 컴퓨터 프로그래밍에 대한 의존도를 낮추고, 딥러닝 이론에 대한 전반적인 이해만으로 딥러닝 모델을 구축할 수 있는 DeepBlock이라는 웹 기반 딥러닝 교육용 플랫폼을 제안한다.

제안하는 DeepBlock은 초심자가 GUI를 기반으로 딥러닝 모델을 구성하는데 필요한 구성요소를 블록 형태로 조합하여 딥러닝 모델을 구축할 수 있도록 지원한다. 사용자가 직접 학습 및 검증에 필요한 데이터셋을 구성할 수 있으며, 학습과정과 학습결과를 실시간으로 그래프 및 도식화된 형태로 확인할 수 있다. 또한 제안한 DeepBlock은 클라우드 서비스^[4]를 기반 동작하도록 설계하고 구현되었다. 따라서 관련 서비스를 활용하는데 있어 장소, 플랫폼, 환경 설정 등의 부가적인 제약이 받지 않는다. 이와 같은 기능을 제공하는 DeepBlock은 딥러닝을 교육하는데 효과적으로 사용될 수 있다.

이후 2장에서는 GUI(Graphic User Interface) 기반으로 딥러닝 모델을 구축할 수 있는 다른 서비스들을 다루고 3장에서는 관련 연구들의 한계와 제안하는 DeepBlock에 대해서 설명하고 DeepBlock의 설계에 대해서 다룬다. 4장에서는 제안한 DeepBlock의 구현 결과에 대하여 다룬다. 5장에서 본 논문의 결과와 향후 연구계획에 대해서 서술하면서 논문을 마친다.

II. 관련 연구

딥러닝에 대한 세간의 관심이 높아지면서 세계적인 선두 기업에서 초심자를 위한 딥러닝 플랫폼을 제공하거나 GUI 기반의 딥러닝 전용 통합 개발 환경을 제공하여 딥러닝 모델을 구축하고 시각화하여 분석할 수 있도록 하였다^{[5][6]}.

각 기업에서는 사용자가 딥러닝을 보다 편리하게 사용할 수 있도록 여러 노력을 기울이고 있다. 하지만 기업에서 개발한 서비스 들은 사용자가 직접 모델을 재구성할

수 없거나 개발 환경을 구축하는 것 자체가 초심자에게는 어려운 경우가 대다수이다. 또한 고급 사용자를 위한 서비스는 딥러닝 전용 통합 개발 환경이기 때문에 초심자가 사용하기에는 고려해야할 점이 많다. 따라서 이러한 서비스들은 딥러닝에 대한 간단한 지식을 가지고 있는 초심자를 위한 교육 서비스라 보기 어렵다. 따라서 우리는 초심자를 위한 딥러닝 학습 플랫폼 DeepBlock을 제안한다.

III. DeepBlock의 설계

본 논문에서 제안하는 DeepBlock은 다음과 같은 특징을 가진다. 사용자는 별도의 프로그래밍이나 전처리 없이 모델을 구성하고 데이터를 사용할 수 있으며 DNN, CNN 등 통상적으로 많이 사용되는 네트워크들을 기본적으로 이용할 수 있다. 사용자는 네트워크의 파라미터 값을 바꾸거나 레이어를 직접 수정해가며 변경한 값이 학습에 어떠한 영향을 미치는지를 실시간으로 그래프 형태로 결과를 확인할 수 있다. 이를 통해 DeepBlock을 사용하는 초심자는 이를 사용하는 과정에서 딥러닝 모델을 자연스럽게 배우고 습득할 수 있다.

본 연구에서 제안하는 모델은 클라우드 플랫폼을 기반으로 동작하도록 설계하였으며, 프론트엔드와 백엔드가 분리되어 있는 구조를 갖는다. 프론트엔드에서는 백엔드의 Rest API를 기반으로 서버로 DeepBlock에서 지원하는 특정 기능을 요청하고 백엔드에서는 해당 요청에 대하여 컨트롤러를 통해 데이터베이스에 접근하거나 특정 로직을 수행한 후, 최종적으로 결과에 대해서 응답하는 형태로 동작한다.

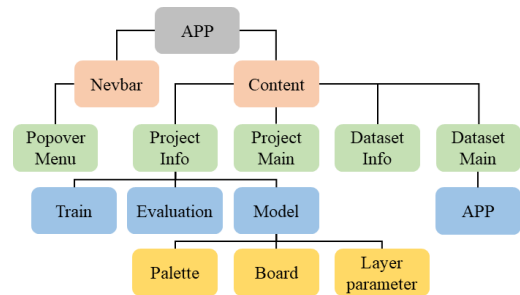


그림 1. 컴포넌트 구성도
Fig. 1. Component diagram

1. 프론트엔드

제안하는 DeepBlock은 단순히 딥러닝 모델을 구현하는 웹서비스가 아닌, 초심자가 딥러닝의 전반적인 이해를 마친 이후 실제로 딥러닝 모델을 구현하고 모델을 분석하는 과정을 돕기 위한 웹 기반 서비스를 제공한다. 따라서 모델 학습 단계에서 사용자에게 딥러닝 학습에 필요한 최대한의 기능을 제공하고자 하였다. 또한 모델을 구성하고 학습시키는데 있어 복잡한 기능을 제외하고 필요한 최소한의 기능을 제공하였다. 뿐 만 아니라, 사용자가 모델을 학습하고 테스트하는 과정에서 텍스트 기반의 결과 대신 정형화된 시각화 자료를 제공하여 딥러닝 학습의 부담감을 덜고자 하였다. 때문에 우리는 실시간으로 서버의 처리결과를 렌더링할 수 있는 반응형 웹 페이지 프레임워크인 Vue.js를 사용한다.

Vue.js^[7]는 javascript 기반의 프론트엔드 프레임워크이며 많은 모듈을 제공하기 때문에 사용자에게 친숙한 UI를 제공하는데 여러 이점을 얻을 수 있다. 또한 그림 1과 같이 기능별로 여러 컴포넌트로 나누어 모듈화된 형태로 설계하였다.

2 백엔드

고성능의 백엔드 서버를 구현하기 위해 Javascript 기반의 프레임워크인 Node.js를 채택하였다^[8]. 프론트엔드의 요청을 받고 응답하기 위해 서버의 API를 Restful^[9]하게 구축하였다. 또한 HTTP와 Connect 컴포넌트를 기반으로 하는 웹 프레임워크인 Express.js를 이용한다. 사용자의 요청을 효율적으로 처리하기 위해 라우터를 적용하였다. 사용자의 요청이 발생했을 때 백엔드에서는 컨트롤러를 사용해 처리한다. 사용자의 정보와 각종 데이터를 처리하기 위해 관계형 데이터베이스 관리 시스템인 MySQL을 사용하였으며 MySQL을 좀 더 쉽게 다룰 수 있도록 도와주는 라이브러리인 Sequelize를 사용하였다. 사용자는 바로 데이터베이스에 접근할 수 없으며 컨트롤러를 통해서 데이터베이스에 접근가능하다. Node.js에서 CNN, DNN에 필요한 기능을 지원하기 위해 자바스크립트 기반 TensorFlow인 TensorFlow.js를 사용하였다^[10]. 이러한 구조는 Javascript라는 단일 프로그래밍 언어를 백엔드에서 사용할 수 있다는 장점을 가진다.

가. Rest API

Rest API는 HTTP Method(GET, POST, DELETE, UPDATE)를 사용하여 어떤 자원에 대한 CRUD (Create,

Read, Update, Delete) 연산을 수행하기 위해 서버의 특정 URI(Uniform Resource Identifier)에 요청을 보낸다^[11]. 요청을 보내기 위해 AJAX(Asynchronous Javascript and XML)를 사용하였으며 요청을 위한 자원은 특정한 형태로 표현된다. 모든 요청은 JWT 토큰을 사용하여 사용자를 확인한다. 본 플랫폼에서 사용된 주요 Rest API는 각각 기능 별로 다음과 같이 정의되어 있다.

표 1. 프로젝트 관리 Rest API

Table 1. Rest API for project management

기능	메소드	URI
생성	POST	/api/u/project
조회	GET	/api/u/project
변경	PUT	/api/u/project/:project_id
삭제	DELETE	/api/u/project/:project_id

표 1는 프로젝트 관리 기능을 위한 Rest API를 세부적으로 명시해 놓은 것이다. 사용자는 각각의 딥러닝 모델을 다수의 프로젝트로 생성하고 삭제 할 수 있으며 사용자가 프로젝트 이름 변경, 삭제 요청 시 서버에 프로젝트 고유번호를 URI에 포함시켜 요청을 보내는 것으로 프로젝트에 접근할 수 있다.

표 2. 데이터셋 관리 Rest API

Table 2. Rest API for dataset management

기능	메소드	URI
생성	POST	/api/u/dataset
조회	GET	/api/u/dataset
변경	PUT	/api/u/dataset/:dataset_id
삭제	DELETE	/api/u/dataset/:dataset_id

표 2은 데이터 셋 관리 기능을 위한 Rest API를 세부적으로 명시해 놓은 것이다. 사용자는 여러 개의 데이터 셋 디렉터리를 만들어 자신의 데이터를 저장할 수 있으며 데이터 셋 고유번호를 URI에 포함시켜 명시해 놓은 것으로 사용자는 자신의 데이터 셋을 사용하여 모델을 학습하고 테스트 할 수 있다.

표 3. 클래스 관리 Rest API

Table 3. Rest API for class management

기능	메소드	URI
생성	POST	/api/u/dataset/:dataset_id/class
조회	GET	/api/u/dataset/:dataset_id/class
변경	PUT	/api/u/dataset/:dataset_id/class/:class_id
삭제	DELETE	/api/u/dataset/:dataset_id/class/:class_id

표 3는 사용자가 데이터 셋의 각각의 데이터들을 클래스별로 나누어 관리하기 위한 Rest API를 보여준다. 데이터 셋, 클래스의 고유번호를 URI에 포함시켜 명사해 놓은 것으로 해당 데이터 셋의 데이터들이 어떤 클래스별로 나누어져 있는지 알 수 있다.

표 4. 이미지 관리 Rest API
Table 4. Rest API for image management

기능	메소드	URI
업로드	POST	/api/u/dataset/:dataset_id/class/:class_id/image
이미지 불러오기	GET	/api/u/dataset/:dataset_id/class/:class_id/image?limit=?&page=?
원본 이미지 불러오기	GET	/api/u/dataset/:dataset_id/class/:class_id/image/:image_id
삭제	DELETE	/api/u/dataset/:dataset_id/class/:class_id/image/:image_id

표 4는 사용자가 클래스별로 나누어 놓은 데이터 셋 안의 이미지들을 관리하기 위한 Rest API를 보여준다. 데이터 셋, 클래스, 이미지 고유번호를 URI에 포함된다.

표 5. 모델 관리 Rest API
Table 5. Rest API for model management

기능	메소드	URI
모델 학습	POST	/api/u/project/:project_id/model/train
모델 테스트	POST	/api/u/project/:project_id/model/test
모델 불러오기	GET	/api/u/project/:project_id/model
학습 결과 불러오기	GET	/api/u/project/:project_id/train
테스트 결과 불러오기	GET	/api/u/project/:project_id/test
예측 결과 불러오기	GET	/api/u/project/:project_id/test/:test_id/prediction
예측 결과 이미지 불러오기	GET	/api/u/project/:project_id/test/:test_id/prediction/:predict_id
모델 저장	PUT	/api/u/project/:project_id/model

표 5은 사용자가 자신의 프로젝트별로 만든 모델을 학습하고 테스트한 결과들을 관리하기 위한 Rest API를 보여준다. 프로젝트, 테스트 고유번호가 URI에 포함되어 있는 형태이다. 사용자는 자신의 데이터 셋을 모델에 적용하여 학습하고 나온 결과를 JSON 파일 형식으로 데이터 셋 폴더에 저장을 해두고 저장된 결과를 이용하여 테스트를 할 수 있다. 테스트는 사용자가 자신의 데이터 셋 폴더에 저장한 모델의 학습 결과를 불러와 진행할 수 있다.

나. 데이터베이스

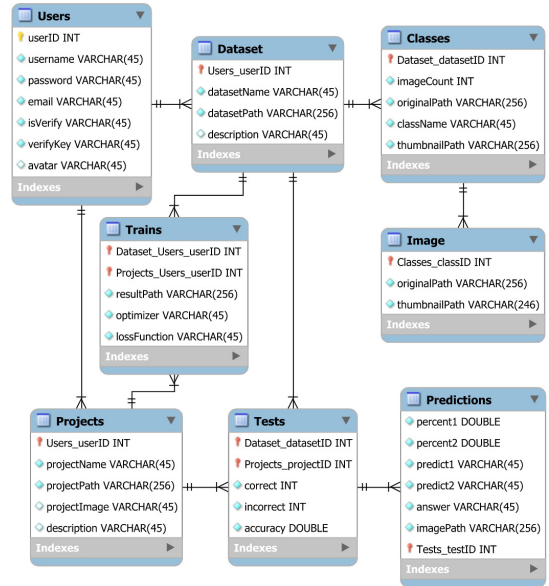


그림 2. 데이터베이스 테이블 관계도
Fig. 2. ERD of the database table

그림 2의 각각의 테이블은 위와 같은 관계를 가진다. User 테이블은 사용자의 정보를 다루기 위해 사용되고 사용자는 여러 데이터 셋과 프로젝트를 생성하고 삭제할 수 있다. 그래서 각각의 Datasets, Projects 테이블과 1:N 관계를 가진다. Classes 테이블은 데이터 셋의 많은 이미지를 관리하기 위해 있으며, Tests 테이블과 Trains 테이블은 사용자가 자신이 만든 모델에 데이터 셋을 사용하여 학습하고 테스트를 한 후의 결과를 저장하기 위해 사용된다. 또한 Tests 테이블은 사용자에게 예측된 값을 보여주기 위해 학습한 모델이 테스트 하고 나온 예측 값을 저장하는 Predictions 테이블과 1:1 관계를 가진다.

IV. DeepBlock의 구현 결과

1. 프론트엔드 및 백엔드 서버 컴퓨팅 환경

앞서 설명한 DeepBlock을 설계한 데이터베이스, Rest API, 필요한 알고리즘과 컴포넌트 구성을 적용하여 클라우드를 기반으로 구현하였다. 표 6과 표 7은 백엔드 서버와 프론트엔드 서버의 컴퓨팅 환경이다. 이후 구현 결과를 기능별로 나누어 설명한다.

표 6. 백엔드 컴퓨팅 환경

Table 6. Back-end computing environment

항목	세부정보
운영체제	Ubuntu 18.04
CPU	Intel i7-6850k (6 core, 3.6 GHz)
RAM	16GB (1 * 16GB)
GPU	NVIDIA GTX 2080Ti
CUDA 버전	CUDA Toolkit-10.0
서버 프레임워크	Node.js 12.16.1
데이터베이스	MySql 5.7
딥러닝 프레임워크	TensorFlow.js

표 7. 프론트엔드 컴퓨팅 환경

Table 7. Front-end computing environment

항목	세부정보
운영체제	Ubuntu 18.04
CPU	Intel i7-6850k (6 core, 3.6 GHz)
RAM	16GB (1 * 16GB)
웹 프레임워크	Vue.js 2.6.11
컴포넌트 프레임워크	Vuetify.js 2.2.32

2. 데이터 관리

사용자가 모델을 학습하는 데는 데이터셋이 필요하다. 데이터셋은 다수의 클래스로 구성되며 클래스는 각 개체들을 분류하는 단위이다.

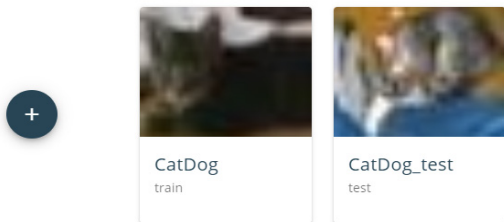


그림 3. 데이터셋 관리 페이지

Fig. 3. Page for dataset management

그림 3은 데이터셋 관리 페이지의 구현 결과이다. 데이터셋 추가, 삭제가 가능하며 해당 사용자가 생성한 데이터셋의 목록을 확인할 수 있으며, 각 데이터셋의 샘플 이미지를 확인할 수 있다. 데이터 셋을 클릭하면 해당 데이터셋의 자세한 정보를 확인할 수 있다.

그림 4는 데이터셋의 상세정보를 볼 수 있는 페이지이다. 해당 페이지에서 클래스 추가, 삭제, 이름 변경이 가능하다. 또한 사용자는 업로드된 이미지를 확인할 수 있으며 특정 이미지를 삭제할 수 있다.

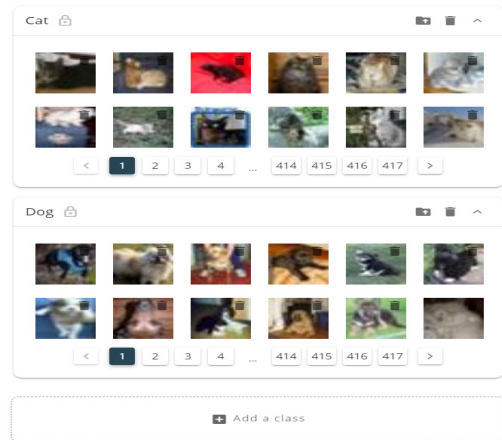


그림 4. 데이터셋 상세 정보

Fig. 4. Detailed dataset information

3. 모델 관리

각각의 프로젝트를 관리하기 위한 페이지가 제공되며, 사용자의 딥러닝 모델은 프로젝트 단위로 관리된다. ProjectMain은 로그인한 사용자가 자신의 프로젝트를 만들고 관리하는 페이지이다.

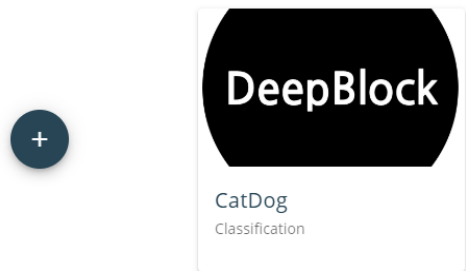


그림 5. 프로젝트 관리 페이지

Fig. 5. Page for project management

그림 5에서와 같이 '+' 버튼을 사용하여 새로운 프로젝트를 만들 수 있고, 삭제를 원하는 경우 'x' 버튼을 눌러 삭제할 수 있다. 모든 프로젝트는 백엔드 서버에 있는 DB와 연동된다.

로그인한 사용자가 프로젝트를 만들고 클릭하면 Model 컴포넌트로 라우팅 되어 딥러닝 모델을 만들 수 있다. Model은 Board, Train, Evaluation이라는 자식 컴포넌트를 갖는다. Board에서 사용자가 블록을 드래그 앤 드롭 방식으로 레이어를 쌓고 직접 파라미터 정보를 넣어 모델을 만들 수 있다. Train과 Evaluation에서는 Board에서 만든 모델을 기반으로 학습된 결과를 확

인할 수 있다. 표 8은 Board 컴포넌트를 부모 컴포넌트로 갖는 자식 컴포넌트에 대한 기능을 설명해 놓은 것이다.

표 8. Board의 자식 컴포넌트
Table 8. Board's child component

컴포넌트	기능
Palette	지원하는 레이어들이 모인 곳
Block	블록 형태로 레이어를 쌓는 곳
LayerParameter	레이어의 파라미터 값을 입력 하는 곳

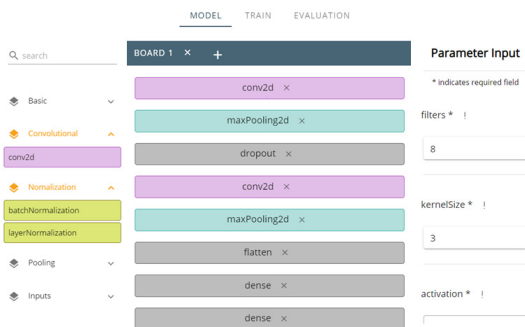


그림 6. Board 페이지
Fig. 6. Board page

해당 Board 페이지 구성은 그림 6과 같으며, 좌측부터 Palette, Block, LayerParameter 순으로 구성되어 있다. 사용자는 Palette에 적합한 레이어를 드래그 앤 드롭으로 Block 컴포넌트에 쌓을 수 있으며, 각각의 레이어를 클릭하여 LayerParameter 컴포넌트에서 각 레이어에 맞는 파라미터 값을 입력할 수 있다. 각 레이어의 파라미터 값은 TensorFlow.js 문서에 정의된 파라미터를 사용하며, 각 파라미터 값에 대한 설명을 볼 수 있다. 또한 필수 파라미터 값과 그렇지 않은 값을 구분하여 입력받고, 입력 받은 파라미터 값은 JSON형태로 서버에 전송된다.

그림 7은 모델을 학습할 수 있는 페이지이다. 사용자가 보유하고 있는 데이터셋을 선택하고 생성한 모델 학습을 시작한다. 이후 백엔드 서버에서 사용자가 선택한 데이터셋의 이미지를 불러온 후 실제 학습이 진행된다. 학습 과정은 그림 7에서 확인할 수 있듯이 그래프 형태로 실시간으로 사용자에게 보여주며, 각 Epoch의 정확도와 손실 값을 확인할 수 있다. 또한 페이지 상단에서 학습의 진행률을 확인할 수 있다.

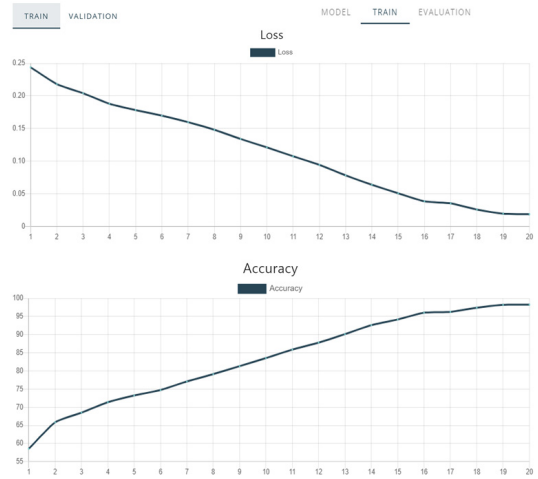


그림 7. 모델 학습 페이지
Fig. 7. Page for model training

그림 8은 모델을 학습한 이후에 모델을 평가하는 페이지이다. 모델 평가는 평가에 사용할 데이터 셋을 선택한 후 평가 시작 버튼을 클릭하면 진행된다. 평가가 완료되면 평가기록은 서버에 저장되며 사용자는 정확도, 정답 개수, 오답 개수를 확인할 수 있다. 또한 각 이미지가 어떻게 분류되었는지 퍼센트로 확인할 수 있다.

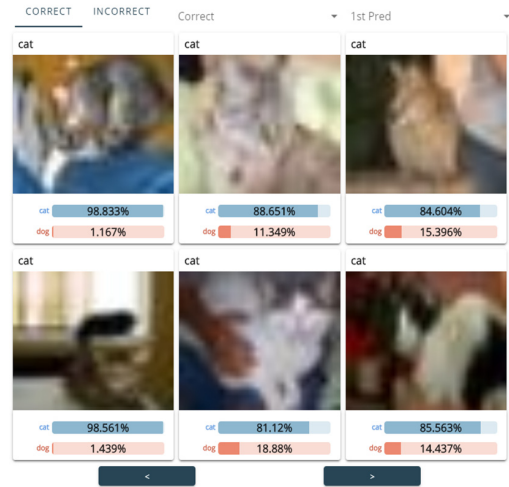


그림 8. 모델 평가 페이지
Fig. 8. Page for model evaluation

각 이미지를 클릭하면 그림 9와 같이 해당 이미지의 예측 결과에 대한 상세정보를 확인할 수 있다.

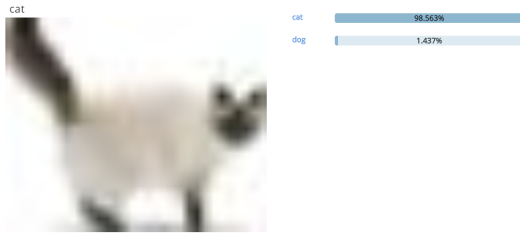


그림 9. 예측 결과 상세정보
Fig. 9. Detailed information of a predicted result

4. 학습 및 관리

사용자가 웹에서 구축한 딥러닝 모델을 실제로 학습시키기 위하여 실제 딥러닝이 가능한 코드로 변환하는 과정이 필요하였다. 사용자가 웹에서 딥러닝 모델을 구성한 후 저장하면 서버에 사용자가 저장한 모델이 JSON 포맷으로 저장된다. 저장된 JSON 파일은 사용자가 학습을 시작하면 서버의 로컬 저장소에서 읽어온 후 각 레이어의 구성 정보를 TensorFlow.js 코드로 변환한다. 변환된 각 레이어에 대한 코드는 순서대로 TensorFlow에서 제공하는 Sequential 모델에 쌓이게 되고 모델 구성, 모델의 입력 채널 크기와 사용자가 선택한 데이터 셋의 크기에 대한 논리적인 오류가 없을 시 정상적으로 컴파일된다. 컴파일이 완료된 후 즉시 학습을 시작하며 학습은 비동기로 처리된다. 학습 과정에서 Epoch 별로 학습 결과를 저장하며 임의의 시점에 프론트엔드에서 Epoch 당 학습결과를 요청할 시 현재까지 진행된 학습결과를 전송하여 실시간으로 그래프를 생성할 수 있다.

V. 결 론

본 연구에서는 초심자를 위한 DeepBlock이라는 교육용 웹 기반 딥러닝 모델 구축 서비스를 제안하고 구현하였다. 본 논문에서 구현한 DeepBlock은 클라우드 기반의 서비스로 사용자가 언제, 어디서든 딥러닝을 경험할 수 있도록 하였다. 또한 간단한 딥러닝 이론에 대한 지식만으로 딥러닝을 체험할 수 있도록 하였다. 즉, 딥러닝의 진입장벽을 낮추는 것에 초점을 두었다.

DeepBlock은 딥러닝 이론에 대한 지식은 있지만, 프로그래밍과 수식에 익숙하지 않은 초심자를 위해 별도의 데이터 전처리 과정을 요구하지 않는다. 또한 블록을 드

래그 앤 드롭 방식으로 딥러닝 모델을 구축할 수 있는 서비스를 제공하였다. 모델의 학습, 테스트 결과를 텍스트가 아닌 그래프와 사진과 같은 직관적인 형태로 제공하여 초심자들이 더욱 쉽게 이해하도록 하였다.

현재까지의 우리의 DeepBlock은 교육과 이미지 분류에만 초점이 맞춰져 있어 경험의 한계가 존재한다. 추후 연구에서는 초심자가 영상, 음성 등의 다양한 형태의 데이터를 다루고 경험할 수 있도록 할 것이다.

References

- [1] Soon-duck Yoo, "Research on the Methodology for Policy Deriving to active Artificial Intelligence", The Journal of the Institute of Internet, Broadcasting and Communication Vol. 20, No. 5, pp. 187-193, Oct 2020. DOI: <https://doi.org/10.7236/IIBC.2020.20.5.187>
- [2] Nwankpa, Chigozie, et al, "Activation functions: Comparison of trends in practice and research for deep learning", arXiv preprint arXiv:1811.03378, 2018. DOI: <https://arxiv.org/abs/1811.03378>
- [3] Jong-Bok Kim, et al, "Recognition of dog's front face using deep learning and machine learning", Journal of the Korea Academia-Industrial cooperation Society(JKAIS), Vol. 21, No. 12, pp. 1-9, Dec 2020. DOI: <http://dx.doi.org/10.5762/KAIS.2020.21.12.1>
- [4] Ga-Ae Ryu, Kwan-Hee Yoo, "A DRM Based Content Protection Method in Content Oriented Educational Cloud System", The Journal of KIIT, Vol. 12, No. 12, pp. 137-147, 2019. DOI: <https://doi.org/10.14801/jkiit.2014.12.12.137>
- [5] Carney, Michelle, et al, "Teachable machine: Approachable Web-based tool for exploring machine learning classification", Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1-8, 2020. DOI: <https://doi.org/10.1145/3334480.3382839>
- [6] Wang, Chih-Min, et al, "The halal logo classification by using NVIDIA DIGITS", 2018 International Conference on Applied Information Technology and Innovation (ICAITI) IEEE, pp. 162-165, 2018. DOI: <https://doi.org/10.1109/ICAITI.2018.8686730>
- [7] Wohlgethan, Eric, "Supporting Web Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue.js", Diss. Hochschule für Angewandte Wissenschaften Hamburg, 2018.
- [8] Tilkov, Stefan, and Steve Vinoski, "Node.js: Using JavaScript to build high-performance network programs", IEEE Internet Computing Vol. 14, No. 6, pp. 80-83, 2010. DOI: <https://doi.org/10.1109/MIC.2010.145>
- [9] Fielding, Roy T., and Richard N. Taylor, "Architectural

styles and the design of network-based software architectures”, University of California, Irvine, 2000.

- [10] Smilkov, Daniel, et al, “Tensorflow. js: Machine learning for the web and beyond”, arXiv preprint arXiv:1901.05350, 2019.
DOI: <https://arxiv.org/abs/1901.05350>
- [11] Masse, Mark, “REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces”, O'Reilly Media, Inc, 2011.

저 자 소 개

조 진 성(준회원)



- 2021.3~: 충북대학교 전기·전자·정보·컴퓨터공학부 석사 재학
- 2020.2: 선문대학교 컴퓨터공학부 학사

김 근 모(준회원)



- 2021.3~: 충북대학교 전기·전자·정보·컴퓨터공학부 석사 재학
- 2020.2: 선문대학교 컴퓨터공학부 학사

고 현 민(준회원)



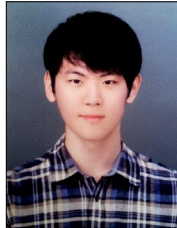
- 2020.2: 선문대학교 컴퓨터공학부 학사

김 성 민(준회원)



- 2020.2: 선문대학교 컴퓨터공학부 학사

김 지 섭(준회원)



- 2020.2: 선문대학교 컴퓨터공학부 학사

김 봉 재(정회원)



- 2021.03~: 충북대학교 컴퓨터공학과 조교수
- 2016.03~2021.02: 선문대학교 컴퓨터공학부 조교수
- 2015.01~2016.02: 한국전자기술연구원 임베디드·SW연구센터 선임연구원
- 2014.09~2014.12: 삼성전자 SW개발팀(메모리) 책임연구원
- 2014.2: 서울대학교 전기·컴퓨터공학부 박사
- 2008.2: 광운대학교 컴퓨터공학부 학사