

이미지 분류를 위한 대화형 인공지능 블록 개발

박영기* · 신유현**

춘천교육대학교 컴퓨터교육과* · 인천대학교 컴퓨터공학부**

요약

엔트리, Machine Learning for Kids, Teachable Machine과 같이 블록 기반 프로그래밍 언어에서 활용할 수 있도록 인공지능을 간단히 학습시킬 수 있는 다양한 플랫폼들이 존재한다. 그러나 이와 같은 플랫폼들은 별도의 메뉴를 통해 인공지능 학습을 진행한 다음, 학습된 모델을 코드 에디터에서 활용하는 방식을 따르고 있다. 이와 같은 방식은 학습되는 과정을 학생들이 더 직관적으로 살펴볼 수 있다는 장점이 있지만, 학습 메뉴와 코드 에디터를 모두 활용해야 한다는 단점도 존재한다. 본 논문에서는 코드 에디터에서 인공지능 학습과 코딩을 모두 진행할 수 있는 인공지능 블록을 개발한다. 본 인공지능 블록은 스크래치 블록으로 제시되지만 실제 학습 과정은 파이썬 서버를 통해 수행된다. 파란색 펜과 빨간색 펜을 분류하는 모델, 덴탈 마스크와 KF94 마스크를 분류하는 모델을 학습하는 과정을 통해 본 블록에 대해 상세히 기술한다. 또, 학습 성능 면에서 Teachable Machine과 큰 차이가 없음을 실험적으로 나타내었다.

키워드 : 인공지능, 스크래치, 파이썬, 대화형 블록, 이미지 분류

The Development of Interactive Artificial Intelligence Blocks for Image Classification

Youngki Park* · Youhyun Shin**

Department of Computer Education, Chuncheon National University of Education*

Department of Computer Science and Engineering, Incheon National University**

Abstract

There are various educational programming environments in which students can train artificial intelligence (AI) using block-based programming languages, such as Entry, Machine Learning for Kids, and Teachable Machine. However, these programming environments are designed so that students can train AI through a separate menu, and then use the trained model in the code editor. These approaches have the advantage that students can check the training process more intuitively, but there is also the disadvantage that both the training menu and the code editor must be used. In this paper, we present a novel artificial intelligence block that can perform both AI training and programming in the code editor. While this AI block is presented as a Scratch block, the training process is performed through a Python server. We describe the blocks in detail through the process of training a model to classify a blue pen and a red pen, and a model to classify a dental mask and a KF94 mask. Also, we experimentally show that our approach is not significantly different from Teachable Machine in terms of performance.

Keywords : Artificial Intelligence, Scratch, Python, Interactive Blocks, Image Classification

이 논문은 2021년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2020R111A3068836)
교신저자 : 신유현(인천대학교 컴퓨터공학부)

논문투고 : 2021-12-07

논문심사 : 2021-12-08

심사완료 : 2021-12-08

1. 서론

블록 기반 프로그래밍 언어에 기반한 다양한 교육용 인공지능 학습 플랫폼들이 존재한다. Machine Learning for Kids [1], Teachable Machine [2], 엔트리[3], Cognimates [4]와 같은 플랫폼들이 대표적이다. 이를 활용하여 학생들은 이미지, 텍스트, 소리, 숫자 등 다양한 형태의 데이터를 활용하여 마우스로 드래그하며 머신러닝 모델을 간단히 학습시켜볼 수 있다. 이때 학습시킨 모델은 엔트리나 스크래치와 같은 블록 프로그래밍 언어에서 활용 가능하다.

기존의 인공지능 교육용 플랫폼들은 머신러닝 모델을 만들기 위한 별도의 화면을 제공한다는 특징을 가지고 있다. 해당 메뉴에서 학생들은 마우스로 학습데이터를 드래그하는 과정을 통해 모델을 학습시킬 수 있고, 학습시킨 모델의 성능이 어느 수준인지를 자신이 가지고 있는 데이터를 이용하여 테스트해 보는 것도 가능하다. 충분히 학습을 시켰다고 판단된다면 코드 에디터에 접속하여 해당 모델을 활용한 프로그래밍을 시작할 수 있다.

이와 같이 머신러닝 모델을 학습시키기 위한 별도의 메뉴를 마련하는 것은, 처음 인공지능을 접하는 학생들이 자신들의 머신러닝 모델이 어떻게 학습되었는지를 시각적으로 확인할 수 있다는 점에서 교육적으로 큰 효과가 있다고 볼 수 있다. 그러나 한편으로는 인공지능 학습을 위한 메뉴의 수가 많아지고, 화면 전환 횟수가 많아진다는 것은 코딩 시 약간의 학습 부담을 야기한다 [5]. 같은 맥락으로, [6]에서는 처음 코딩을 시작하는 학생들의 경우 블록과 메뉴를 간소화하는 것이 도움이 된다고 주장한다. 실제 프로그래머들의 경우에도 머신러닝 모델을 학습시키기 위해 별도의 GUI를 활용하는 경우는 많지 않다.

본 논문에서는 블록 코딩이 가능한 화면에서 이미지 분류를 위한 인공지능 학습과 코딩을 모두 진행할 수 있는 인공지능 블록을 개발한다. 이를 위해 선행 연구 [5]에 기반하여 새로운 스크래치[7, 8] 블록을 개발한다. 스크래치는 전세계에서 가장 많이 사용되고 있는 블록 프로그래밍 언어로, 컴퓨팅 사고력을 신장시키는 데에 효과적이다[9]. 본 인공지능 블록은 대화형 블록으로써, 친구와 대화하는 것과 같은 방식으로 인공지능을 학습시키고, 그 학습된 결과를 확인할 수 있다. 본 논문에서

는 해당 블록을 활용하여 파란색 펜과 빨간색 펜을 분류하는 모델, 덴탈 마스크와 KF94 마스크를 분류하는 모델을 학습시킴으로써 본 블록의 사용 방법에 대해 자세히 기술한다. 또, 학습 성능의 면에서 다른 인공지능 교육 플랫폼들과 큰 차이가 없음을 실험적으로 나타낸다.

본 논문에서 공헌한 바는 다음과 같다.

- (3절) 본 연구에서 제시하는 기술에 대해 설명한다. 구체적으로, 이미지 분류를 위한 머신러닝 모델을 학습시키고 활용할 수 있는 새로운 스크래치 인공지능 블록을 개발한다. 본 블록은 선행 연구[5]를 확장한 것으로, 선행 연구에서 제시하지 않았던 백엔드(back-end) 프로그램 구현 방법을 추가로 제시한 것이다.
- (4절) 3절에서 제안한 대화형 인공지능 블록을 활용하는 2가지 예제 프로그램을 제시하였다. 파란색 펜과 빨간색 펜을 분류하는 프로그램, 덴탈 마스크와 KF94 마스크를 분류하는 프로그램을 어떻게 구현할 수 있는지 상세히 기술하였다.
- (5절) 3절에서 제시한 인공지능 블록이 성능 면에서 학생들이 사용하기에 적절하다는 것을 Teachable Machine과 비교하며 나타내었다.

2. 관련 연구

블록 프로그래밍 언어에 기반한 가장 대표적인 인공지능 교육 플랫폼은 엔트리[3], Machine Learning for Kids [1], Teachable Machine [2] 등이 있다. 이들은 모두 이미지 데이터를 활용하여 지도 학습 모델을 학습시키는 것이 가능하다. 뿐만 아니라 플랫폼에 따라 텍스트 데이터, 음성 데이터, 숫자 데이터, 포즈 데이터 등을 학습하는 것도 가능하며, 비지도 학습 기능을 제공하기도 한다. 본 논문에서는 이미지 분류에 초점을 맞추고 있으므로, 본 절에서는 이미지 분류를 위해 각 인공지능 플랫폼들이 어떤 방식으로 학습 기능을 제공하고 있는지에 대해 기술한다.

엔트리는 국내에서 가장 대표적인 교육용 프로그래밍 언어이며, 이미지 분류를 위한 학습 기능을 제공하고 있다. 로그인 후 ‘인공지능’ 메뉴에서 ‘인공지능 모델 학습

하기'를 클릭하면 이미지 분류, 텍스트 분류, 음성 분류 등 학습하기를 원하는 유형이 나타난다. '이미지 분류'를 클릭하면 각 클래스마다 학습 데이터를 추가할 수 있는 별도의 화면이 나타난다. 학습할 파일을 업로드하거나 웹캠을 이용하여 학습 데이터를 촬영한 다음, '모델 학습하기' 버튼을 클릭하면 학습이 진행된다. 학습된 모델은 마찬가지로 파일 선택 또는 웹캠을 이용하여 테스트가 가능하며, '적용하기' 버튼을 클릭하면 코드 에디터에서 모델을 사용할 수 있다. 학습 시 세대(Epoch), 배치 크기(Batch Size), 학습률(Learning Rate), 검증 데이터 비율(Validation Rate)을 설정할 수 있는 기능도 제공하고 있다. 학습 과정이 끝나면 코드 에디터에서는 학습된 모델을 활용할 수 있다. 현재 웹캠으로 촬영 중인 모습을 분류할 수도 있고, '학습한 모델로 분류하기' 블록을 활용하면 사용자가 별도의 추가 메뉴를 통해 이미지를 업로드할 수도 있다.

Machine Learning for Kids는 세계적으로 가장 널리 쓰이는 블록 프로그래밍 언어인 스크래치를 확장하여 머신 러닝 모델을 학습할 수 있는 기능을 제공하는 플랫폼이다. 엔트리와 마찬가지로 로그인 후 학습 기능을 활용할 수 있으며, 엔트리보다 학습 및 활용 과정이 더 세분화되어 있어 더 많은 스크린 전환이 요구된다. 이 플랫폼에서는 '훈련', '학습 & 평가', '만들기' 등 총 3개의 메뉴를 제공한다. 훈련 메뉴에서는 각 클래스별 이미지 데이터를 마련한다. 엔트리와 이 과정이 거의 유사하지만, 로컬 컴퓨터에 존재하는 이미지 파일을 올릴 수 없는 대신 웹소스를 활용하여 이미지를 업로드할 수 있는 점과, 그리기 기능을 이용할 수 있는 점이 차이점이다. 엔트리와 마찬가지로 5장 이상의 사진을 마련해야 학습이 가능하지만, 학습 방식을 선택할 수 있는 세부 옵션은 제공되지 않는다. '학습 & 평가' 메뉴에서는 학습된 모델을 직접 테스트해 볼 수 있다. 마지막으로 '만들기' 메뉴를 통해 코드 에디터를 실행하고, 코드 에디터에서 학습된 모델을 활용한 프로그래밍을 수행할 수 있다.

Teachable Machine의 이미지 분류를 위한 학습 메뉴는 엔트리, Machine Learning for Kids와 크게 다르지 않다. 각 클래스별 이미지를 웹캠 또는 로컬 컴퓨터의 파일을 이용하여 업로드할 수 있다. 학습 전 에포크, 배치 크기, 학습률 등을 조절할 수 있고, '모델 학습시

기' 버튼을 클릭하는 간단한 과정을 통해 학습을 진행할 수 있다. 다른 플랫폼과 달리 1개의 이미지만 학습하는 것도 가능하도록 설계되어 있다. 이 플랫폼의 가장 큰 특징은 '모델 내보내기' 기능을 제공한다는 점이다. 예를 들어 Tensorflow 모델로 내보내기를 할 경우 Tensorflow 모델을 활용할 수 있는 프로그래밍 언어라면 어떤 것이든 이 모델을 활용할 수 있다. 즉, 스크래치와 같은 블록 프로그래밍 언어뿐만 아니라 파이썬, 자바스크립트와 같은 프로그래밍 언어에서도 학습된 모델을 활용 가능하다.

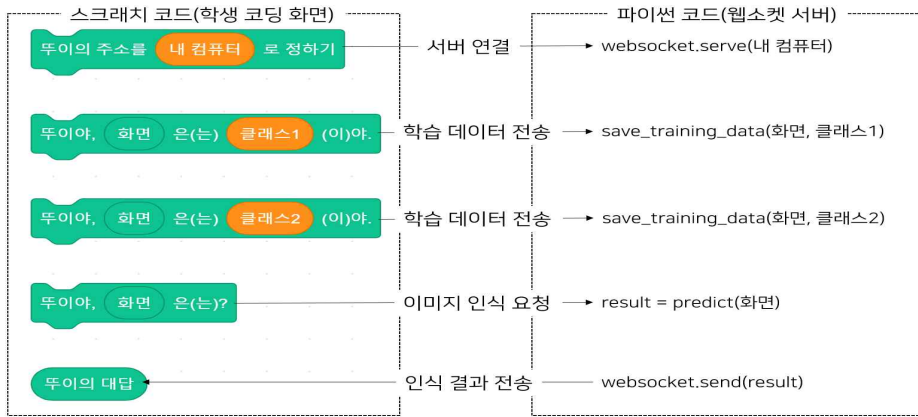
엔트리, Machine Learning for Kids, Teachable Machine의 가장 큰 장점은 학습을 위한 별도 GUI를 제공함으로써 학생들이 단순히 마우스를 드래그하여 학습 데이터를 쉽게 입력할 수 있다는 점이다. 이를 통해 학습하는 과정을 직관적으로 확인할 수 있고, 학습된 모델이 잘 동작하지 않는 경우 추가 데이터를 제공하는 것도 어렵지 않아 교육적으로도 활용 가능성이 높다. 그러나 인공지능 학습을 위해 추가적인 메뉴를 활용해야 한다는 점은 학생들에게 부담이 될 수 있다. 코드 에디터와 학습 메뉴를 번갈아 가며 사용해야 하는 불편함이 존재한다. 실제로 전문 프로그래머들의 경우 코드 에디터에서 학습 및 테스트를 모두 진행하는 것이 일반적으로 사용되는 방법이다. 이와 같은 문제를 해결하기 위해 3절에서는 스크래치의 코드 에디터만을 활용하여 이미지 학습이 가능한 인공지능 블록을 제시한다.

3. 이미지 분류를 위한 대화형 인공지능 블록 개발

본 절에서는 이미지 분류를 위한 대화형 인공지능 블록을 개발하였다. 3.1절에서는 본 인공지능 블록이 스크래치에서 보여지는 인터페이스에 대해 소개하고, 3.2절에서는 3.1절에서 제안하는 스크래치 블록들이 통신허게 될 파이썬 코드의 구현 방법에 대해 기술한다. 3.3절에서는 학습에 활용된 알고리즘에 대해 상세히 기술한다.

3.1. 스크래치 블록 인터페이스 구성

본 절에서 제안하는 이미지 학습을 위한 스크래치 블록의 인터페이스는 (Fig. 1)에 나타난 것과 같이 선행



(Fig. 1) An overview of our interactive artificial intelligence blocks. Note that we reuse the block interfaces in existing study [5].

연구[5]를 통해 제공되는 블록 인터페이스를 재구성하여 사용하였다. 구체적으로, 아래와 같이 (1) 접속 블록, (2) 학습 블록, (3) 예측 블록으로 나누어 구성하였다.



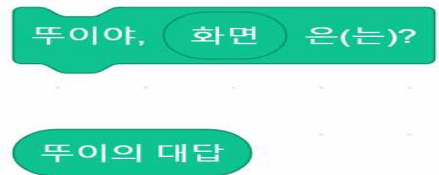
(Fig. 2) A block interface for connecting to a Python server via WebSockets

(Fig. 2)은 파이썬 웹소켓 서버에 접속하기 위한 블록을 나타낸다. 스크래치 내에서 학습을 진행하지 않고 별도의 서버에서 학습을 진행하기 위해 활용되는 블록이다. '내 컴퓨터' 변수는 서버의 IP 주소를 가지게 된다.



(Fig. 3) Block interfaces for adding the current screen to the training data

(Fig. 3)는 학습 데이터를 구축하기 위한 블록을 나타낸다. 그림의 첫 번째 블록을 실행하게 되면, 현재 보이는 화면이 클래스1에 해당된다는 것을 파이썬 웹소켓 서버에 전달한다. 마찬가지로 그림의 두 번째 블록을 실행하게 되면, 현재 보이는 화면이 클래스2에 해당된다는 것을 웹소켓 통신을 통해 파이썬 서버에 전달한다. 이때 통신하게 되는 파이썬 서버는 (Fig. 2)에 나타난 접속 블록을 통해 설정된 서버와 같다.



(Fig. 4) Block interfaces for predicting the class of the current screen

(Fig. 4)의 첫 번째 블록은 학습된 모델을 이용하여 현재 화면에 나타난 이미지의 클래스를 예측하기 위한 블록을 나타낸다. 예측된 클래스는 이 그림의 두 번째 블록에 저장된다. 구체적으로 기술하면, 이 그림의 첫 번째 블록이 실행되었을 때 현재의 화면이 파이썬 웹소켓 서버로 전송되고, 파이썬 서버에서는 3.2절에서 기술하는 알고리즘에 따라 딥러닝 모델을 학습하고, 학습된 모델을 활용하여 예측된 클래스가 이 그림의 두 번째 블록에 저장된다.

3.2. 인공지능 학습을 위한 서버 프로그램 구현

본 세부 절에서는 3.1절에 나타난 스크래치 블록을 실행했을 때 전달되는 정보를 파이썬 웹소켓 서버에서 어떻게 처리하는지에 대해 기술한다. (Fig. 1)에 나타난 접속 블록은 단순히 연결을 수립하는 것이기에 크게 중요하지 않지만, (Fig. 3)에 나타난 ‘학습’ 블록과 (Fig. 3)에 나타난 ‘예측’ 블록의 경우 서버에서 처리하는 방식이 블록의 성능에 큰 영향을 미치게 된다.

(Fig. 3)의 학습 블록이 실행될 경우, 웹소켓 서버에서는 Base64로 인코딩된 PNG 이미지와 클래스 레이블을 전달받는다. 이때 바로 학습을 진행하지 않고 단순히 학습데이터 폴더에 해당레이블의 폴더를 만들어 PNG 파일을 저장한다. 스크래치의 해상도가 너비 480 픽셀, 높이 360 픽셀이지만, 학습 속도를 빠르게 하기 위해 PNG 이미지의 해상도는 너비 240 픽셀, 높이 180 픽셀로 축소되어 전달 및 저장된다. 바로 학습을 시작하지 않고 학습데이터를 저장하기만 하는 것은 데이터 수정을 용이하게 하기 위함이다. 만약 학생들이 잘못된 학습데이터를 입력했다면, 추후 학습이 진행되기 전까지 해당 폴더에서 PNG 파일을 삭제할 수 있다. 또, 만약 학생들이 Kaggle이나 Github와 같은 사이트에서 대용량의 이미지 데이터를 다운로드 받아 마련한 상황이라면, 이 데이터를 해당 폴더에 복사하기만 하면 추후 학습이 진행될 때 해당 데이터를 학습하게 된다.

(Fig. 4)의 예측 블록이 실행되면 파이썬 웹소켓 서버는 Base64로 인코딩된 PNG 파일 이미지(너비 240 픽셀, 높이 180 픽셀)를 전달받는다. 만약 학습데이터 폴더에 존재하는 모든 이미지들을 학습한 상황이라면, 다시 학습을 진행하지 않고 이미 학습된 모델을 활용하여 전달받은 이미지에 대한 레이블을 예측하여 스크래치로 전달한다. 만약 학습데이터 폴더에 존재하는 모든 이미지를 학습하지는 않은 상황이라면, 학습을 진행한 후, 전달받은 이미지에 대한 레이블을 예측하여 스크래치로 전달한다. 학습이 완료된 모델은 특정 폴더에 .h5 파일의 형태로 저장되며, 추후 프로그램을 종료하고 다시 실행하더라도 해당 파일에서 모델을 읽어 재사용할 수 있도록 구현하였다.

3.3. 학습 알고리즘

본 세부 절에서는 파이썬 웹소켓 서버 프로그램에서 사용된 학습 알고리즘에 대해 기술한다. 관련 연구의 경우, Teachable Machine의 첫 번째 버전[10]은 SqueezeNet [11]을, 두 번째 버전[12]은 MobileNet [13]을 사용한다. 우리가 아는 한 Machine Learning for Kids, 엔트리 등 다른 플랫폼의 경우 어떤 학습 알고리즘을 사용하고 있는지 알려지지 않았다. 본 서버 프로그램에서는 기존 프로그램들과 대등한 학습 성능을 나타내기 위하여 MobileNetV2 [14]를 활용하였다.

본 학습 알고리즘에서는 사전 학습된 MobileNetV2의 100번째 층부터 학습 데이터를 이용하여 미세 조정을 함으로써 전이학습을 하고자 했다. 옵티마이저는 RMSprop, 손실함수는 Binary Cross Entropy를 활용하였다. 하이퍼파라미터의 경우 학습률은 0.0001, 배치 사이즈는 1로 설정했다. 최대 100 에포크 학습하도록 설정하였지만, 검증 데이터에 대한 손실 값이 0.01 이상 좋아지지 않을 경우 학습을 조기 종료하도록 설정하였다. 이는 과적합을 방지하고자 하는 목적도 있지만, 블록 에디터에서 학습이 이루어지기 때문에 최소한의 학습 시간을 통해 실행되는 것이 중요하다고 판단했기 때문이다. 검증 데이터는 각 클래스의 1번째 학습 데이터로 설정하였다. 즉, 학습 데이터의 수가 1개일 경우 학습 데이터와 검증 데이터가 동일하다. 그렇지만 인공지능 교육을 실시할 때 보통 캠을 이용하여 학습 데이터 및 테스트 데이터를 수집하는 교육자료가 많기 때문에, 학습 데이터와 테스트 데이터가 유사할 것이라는 가정 하에 이와 같이 설정하였다. 데이터가 매우 적은 상태에서 데이터 증폭 알고리즘을 활용하는 것은 오히려 학습 성능을 떨어뜨릴 위험이 있었기 때문에, 별도의 데이터 증폭은 수행하지 않았다.

4. 대화형 블록을 활용한 예제 프로그램 구현

본 절에서는 3절에서 제시한 대화형 블록을 활용한 두 가지 예제 프로그램을 제시한다. 이를 통해 본 블록을 교육용 프로그램으로 활용 가능함을 나타낸다.



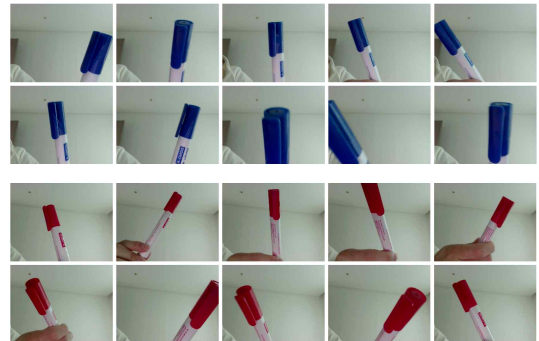
(Fig. 5) An example program that trains and predicts blue and red pens

4.1. 파란펜/빨간펜 분류 프로그램

본 세부 절에서는 대화형 블록을 활용하여 파란펜과 빨간펜을 학습시키고, 테스트하는 프로그램을 만드는 방법을 제시한다. (Fig. 5)는 예제 스크래치 프로그램을 나타낸다. 이 프로그램은 학습하는 코드와 테스트(예측)하는 코드를 모두 포함한다.

본 예제에서 프로그램이 시작되면, '내 컴퓨터', '파란펜', '빨간펜' 변수의 값을 설정한다. 이 변수들의 값은 학생들이 코딩을 편하게 하기 위해 미리 교사들이 설정해 둘 수 있다. 직접 학생들이 파란펜과 빨간펜을 보여주면서 학습시킬 것이기 때문에, 비디오 블록을 켜고 투명도를 0으로 정한다. 마지막으로 뚜이의 주소를 설정함으로써 파이썬 웹소켓 서버에 접속하여, 인공지능 학습을 준비한다.

학습하는 코드는 다음과 같다. '1' 키를 누르면, 현재



(Fig. 6) Example data for classifying blue and red pens

보여지는 스프라이트를 숨기고 비디오 화면만 보여지게 된다. 이때 현재 보여지는 화면(Base64로 인코딩된 데이터)이 '파란펜'을 나타냄을 웹소켓 서버에 전송한다. 마찬가지로 '2' 키를 누르면, 현재 보여지는 스프라이트를 숨기고 현재 보여지는 화면이 '빨간펜'임을 웹소켓 서버에 전송한다. 웹소켓 서버는 각 레이블의 사진들을 정해진 폴더에 저장하고, 필요 시 학습을 진행한다.

스페이스 키를 누르게 되면, 현재의 화면이 웹소켓 서버로 전송되고, 웹소켓 서버에서는 학습된 모델을 이용하여 전송된 데이터의 레이블을 예측한다. 예측된 결과는 '뚜이의 대답' 블록에 저장된다. 이 블록에 저장된 값이 만약 '파란펜'이라면, 스프라이트의 모양을 '파란펜' 모양으로 바꾼다. 반대로 이 블록에 저장된 값이 만약 '빨간펜'이라면, 스프라이트의 모양은 '빨간펜'으로 바뀌어 화면에 나타난다. 빨간펜과 파란펜 모양은 이미 스프라이트에 마련되어 있다고 가정한다.

(Fig. 6)는 학습 및 테스트에 활용된 데이터셋의 일부 사진들을 나타낸다. 학생들이 카메라의 각도를 바꾸지 않고 학습시킬 것이라 가정하여, 학습 데이터와 테스트 데이터 모두 배경은 동일하게 지정하였다. 단, 카메라에 보이는 물체의 모습은 다르게 설정하였다. 실험 결과는 5절에서 자세히 기술하였다.

4.2. 덴탈 마스크/KF94 마스크 분류 프로그램

본 세부 절에서는 덴탈 마스크와 KF94 마스크를 분류하는 프로그램을 제시한다. 제시된 프로그램의 소스



(Fig. 7) Example data for classifying dental and KF94 masks

코드는 (Fig. 5)에 나타난 프로그램과 변수의 이름 외 기타 사항은 동일하다. ‘1’ 키를 눌렀을 때 현재 화면이 덴탈마스크라는 것을 서버에 전달하고, ‘2’ 키를 눌렀을 때는 현재 화면이 KF94 마스크라는 것을 전달한다. 마찬가지로 스페이스 키를 눌렀을 때 현재 화면의 레이블을 학습된 모델을 통해 예측하고, 그 결과에 따라 스포라이트의 모양을 바꾼다.

(Fig. 7)은 학습 및 테스트에 활용된 데이터셋을 예시로 나타낸 것이다. 학생들이 동일한 배경의 데이터를 수집할 것이라 가정하고 배경은 동일하게 구성하였으며, 마스크의 모양과 손의 위치를 바꾸어 가며 데이터를 수집하였다. 수집된 데이터는 240*180 픽셀의 크기로 저장되어 모델 학습 및 테스트에 활용되었다.

5. 실험

본 논문에서 제시된 스크래치 블록의 인터페이스는 선행 연구[5]에서 사용성이 검증되었다고 가정한다. 따라서 본 논문에서는 본 블록의 학습 정확도와 학습 시간을 기준으로 기존 플랫폼과의 비교를 수행하였다. 5.1절에서는 실험 방법에 대해 기술하고, 5.2절에서는 그 실험 결과에 대해 기술하였다.

5.1. 실험 방법

4절에서 제시한 두 가지 예제 프로그램인 ‘과란펜/빨간펜 분류’ 프로그램, ‘덴탈 마스크/KF94 마스크 분류’ 프로그램을 기준으로 하여 비교 실험을 수행하였다.

평가 지표는 예측 정확도(새로운 이미지에 대해 얼마나 정확히 레이블을 예측하는지를 측정)와 학습 시간(주어진 데이터에 대해 모델을 학습하는 시간)으로 측정하였다. 레이블을 예측하는 시간의 경우, 모든 플랫폼이 실시간으로 예측이 가능했기 때문에 비교 대상이 아니라 판단하였다.

평가 대상은 2절에서 제시된 관련 연구들 중 Teachable Machine으로 설정하였다. 별도의 학습 메뉴 없이 블록만으로 학습을 하는 특성 상, 학생들이 아주 적은 수의 데이터를 학습시키는 상황이 종종 일어날 것으로 보였다. 따라서 엔트리와 Machine Learning for Kids는 학습 데이터의 수가 5개 미만일 경우 학습이 되지 않기에 평가 대상에서 제외하였다.

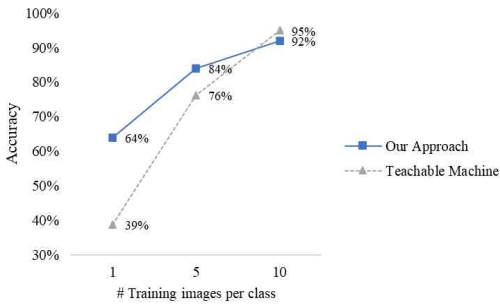
각 레이블마다 각각 1장, 5장, 10장의 이미지 데이터를 학습하는 상황을 가정하였고, 테스트에 활용된 데이터는 각 레이블마다 40장의 이미지 데이터이다. 학습 데이터와 테스트 데이터는 겹치는 이미지가 없도록 실험하였다. 구체적인 학습 및 테스트 데이터는 Github 저장소[15]에 정리하였다.

5.2. 실험 결과

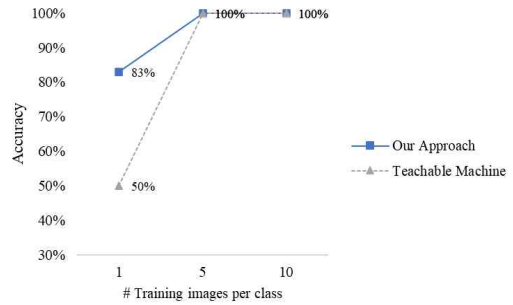
과란펜과 빨간펜을 분류하는 프로그램의 예측 정확도 실험 결과는 (Fig. 8)에, 덴탈 마스크와 KF94 마스크를 분류하는 프로그램의 예측 정확도 실험 결과는 (Fig. 9)에 나타내었다.

두 알고리즘 모두 학습 데이터가 10장이 되었을 때 매우 높은 예측 정확도(90% 이상)를 나타내었다. 특히 덴탈 마스크와 KF94 마스크를 분류하는 문제에서는 각 데이터들의 차이가 크지 않았기 때문에 두 알고리즘 모두 매우 높은 정확도를 나타냈다.

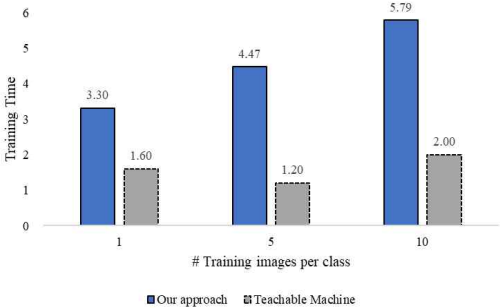
학습 데이터가 적을 때는 제안하는 알고리즘이 두 데이터셋 모두에 대해 Teachable Machine보다 더 높은 성능을 나타냈다. 특히 1장의 이미지만 학습할 때는 Teachable Machine의 경우 두 데이터셋 모두 정확도가 50% 이하를 기록하는 문제가 발생했다. 본 실험 결과를 일반화할 수 있다면, 학생들이 조금씩 데이터를 추가하며 학습 모델의 정확도를 높여나가는 경험을 하고자 하는 경우 본 논문의 알고리즘이 더 유용하다고 볼 수 있다.



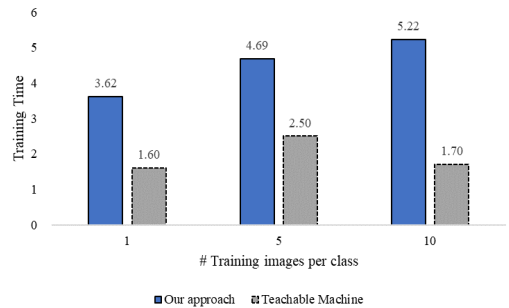
(Fig. 8) The accuracy of each algorithm according to the number of training images in the blue/red pen classification



(Fig. 9) The accuracy of each algorithm according to the number of training images in the dental/KF94 mask classification



(Fig. 10) The training time of each algorithm according to the number of training images in the blue/red pen classification



(Fig. 11) The training time of each algorithm according to the number of training images in the blue/red pen classification

(Fig. 10)과 (Fig. 11)에 나타난 것처럼, 학습 시간의 관점에서는 본 논문의 알고리즘이 Teachable Machine에 비해 전반적으로 약 2~3배의 시간을 필요로 했다. 두 데이터셋 모두 1장의 데이터를 학습시킬 때는 약 3~4초, 5장의 데이터를 학습시킬 때는 4~5초, 10장의 데이터를 학습시킬 때는 5~6초의 시간이 소요되었다. 비록 제안하는 알고리즘이 Teachable Machine에 비해 많은 학습 시간을 필요로 하기는 하지만, 엔트리와 Machine Learning for Kids의 경우에도 5~6초의 학습 시간을 필요로 한다는 점을 감안했을 때 교육적 목적에서 큰 문제가 되지 않을 것으로 예상된다. 또, 필요 시 학습 시간은 하이퍼파라미터 조절을 통해 쉽게 개선될 수 있을 것으로 예상된다.

실험 결과를 종합하면, 학습 성능의 관점에서 본 블록이 기존의 플랫폼에 비해 교육적 목적에서 활용하기

에 부족하지 않다고 판단된다. 본 블록을 활용하면 학생들이 별도의 학습 메뉴 없이 블록만으로 인공지능을 학습하고 테스트하는 경험을 할 수 있을 것으로 기대된다.

6. 결론

본 논문에서는 이미지 분류를 위한 대화형 인공지능 블록을 개발하였다. MobileNetV2에 기반하여 웹소켓 서버에서 수행할 학습 알고리즘을 제안하였고, 이를 인공지능 블록에 적용하는 방법을 제시하였다. 또, 본 인공지능 블록을 이용하여 만들 수 있는 두 가지 예제 프로그램을 제시하였다. 실험을 통해 본 블록을 이용하여 해당 예제 프로그램들을 만들 때 학습 시간이 오래 소요되지 않는다는 점과, 예측 정확도가 교육적 목적에서 우수하다는 것을 타 알고리즘과 비교하여 나타내었다.

본 연구는 이미지 학습을 위한 블록만을 개발하였다는 한계점을 지닌다. 향후 이미지 학습뿐만 아니라 텍스트, 소리, 숫자 등 다른 데이터에 대한 학습도 가능한 블록을 제시하고, 이와 관련된 실제 활용 사례들도 제안하고자 한다. 비지도 학습과 강화 학습을 가르칠 수 있는 블록을 개발하는 것도 향후 연구 목표 중 하나다.

참고문헌

- [18] Lane, D. (2021). Machine Learning for Kids: An Interactive Introduction to Artificial Intelligence. No Starch Press.
- [19] Carney, M., Webster, B., Alvarado, I., Phillips, K., Howell, N., Griffith, J., Jongejan, J., Pitaru, A., and Chen, A. (2020). Teachable machine: Approachable web-based tool for exploring machine learning classification. In Extended abstracts of the 2020 CHI conference on human factors in computing systems, 1-8.
- [20] Entry, <https://playentry.org/>
- [21] Druga, S. (2018). Growing up with AI: Cognimates: From coding to teaching machines. Ph.D. dissertation, Massachusetts Institute of Technology.
- [22] Park, Y. and Shin, Y. (2021). Tooee: A Novel Scratch Extension for K-12 Big Data and Artificial Intelligence Education Using Text-Based Visual Blocks. *IEEE Access*, 9, 149630-149646.
- [23] Tsur, M. and N. Rusk. (2018). Scratch microworlds: designing project-based introductions to coding. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 894-899.
- [24] Resnick, M., Maloney J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., and Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60-67.
- [25] Maloney, J., Resnick, M., Rusk, N., Silverman B., and Eastmond, E. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education*. 10(4), 1-15, 2010.
- [26] Park, Y. and Shin, Y. (2019). Comparing the effectiveness of scratch and app inventor with regard to learning computational thinking concepts. *Electronics*, 8(11), 1269-1280.
- [27] Teachable Machine v1, <https://www.infoq.com/news/2017/10/teachable-machine/>
- [28] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. arXiv preprint arXiv:1602.07360.
- [29] Teachable Machine v2, <https://teachablemachine.withgoogle.com/>
- [30] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, arXiv preprint arXiv:1704.04861.
- [31] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition, 4510-4520.
- [32] Learning Data & Test Data, Github, <https://github.com/TooeeAI/kaie2021/>

저자소개



박 영 기

2008 KAIST 전자전산학과전산학전공
(공학사)

2010 서울대학교 대학원컴퓨터공학과
(공학석사)

2015 서울대학교 대학원컴퓨터공학과
(공학박사)

2015~2016 삼성전자 종합기술원
전문연구원

2016~현재 춘천교육대학교
컴퓨터교육과 부교수

관심분야 : 초등컴퓨터과학교육,
초등인공지능교육

E-Mail : ypark@cnu.ac.kr



신 유 현

2013 고려대학교 컴퓨터교육과
(이학사)

2019 서울대학교 대학원컴퓨터공학과
(공학박사)

2020~현재 인천대학교컴퓨터공학부
조교수

관심분야 : 인공지능, 빅데이터

E-Mail : yhshin@inu.ac.kr