



YouB

2021년 4월 4일

—

이윤재, 임주완

캡스톤 28조

멘토 : 기업멘토님, 김은영 멘토님

개요

비 전공자 혹은 중-고등학생들을 위한 유저 친화적이고 별도의 설치가 필요없는 객체 탐지 AI 모델을 제공하는 웹 서비스의 개발.

목표

1. 별도의 설치가 필요없는 객체 탐지 AI 교육용 웹 서비스의 개발.
2. Javascript와 HTML 기반으로한 AI 학습 과정을 포함한다.

설명

YouB는 You only use Browser의 약자로, 간단히 브라우저를 이용하여 객체 탐지 AI를 체험할 수 있다는 것을 의미한다. 사용자 관점에서, 브라우저를 통해 해당 페이지 접속만 하면 객체 탐지 모델에 대한 개념을 배우고 직접 사용자 정의 데이터를 넣어 AI를 학습시키고 테스트할 수 있는 환경을 제공한다.

산학 과제 지정 기업 : RoboLink

로보링크는 로봇, 드론 에듀테인먼트 회사로서, 교육용 로봇, SW 등을 개발하고 공급하는 회사이다. 현재 로보링크는 비전공자 혹은 학생들이 느낄 수 있는 코드에 대한 거부감, 복잡한 설치 및 실행 환경 등을 지양한 순수 브라우저 기반의 교육용 웹 서비스의 개발을 목표로 하고 있다.

목차

- I. 데이터 구성
- II. 모델 학습
- III. 모델 테스트
- IV. 문제점
- V. 향후 계획

I. 데이터 구성

모델 Configure

```
const config = {  
  "withSeparableConv": true,  
  "iouThreshold": 0.4,  
  "anchors": [  
    { "x": 1.08, "y": 1.19 },  
    { "x": 3.42, "y": 4.41 },  
    { "x": 6.63, "y": 11.38 },  
    { "x": 9.42, "y": 5.11 },  
    { "x": 16.62, "y": 10.52 }  
  ],  
  "classes": ["banana", "coca", "emart", "pepsi"],  
  "objectScale": 5,  
  "noObjectScale": 1,  
  "coordScale": 1,  
  "classScale": 1  
}
```

학습할 데이터 셋에 대한 Configure 의 작성. 테스트할 데이터 셋에 따라 클래스의 종류를 기입해주었다. 모델 학습의 검증을 위해 4개의 클래스 (바나나우유, 코카콜라, 이마트물병, 펄시콜라)와 각 클래스 당 5장의 이미지를 데이터 셋으로 구성한다. 각 이미지에 대한 Annotation 과정은 다음과 같다.



예) Annotating - Bounding Box의 생성

annotation JSON의 예시

```
[{"label": 0, "x": 0.212962962962963, "y": 0.246031746031746, "width": 0.47, "height": 0.54}]
```

Label : 해당 객체의 클래스가 config에 작성된 클래스 배열의 인덱스

x, y, width, height : 해당 객체의 Bounding BOX의 상대적인 위치로, 전체 이미지 사이즈에 비례한다.

II. 모델 학습

모델의 학습에 필요한 시간은 데이터 셋의 구성에 따라 달라진다. 현재까지 테스트해본 결과 3024 x 4032 픽셀 사이즈의 jpg 이미지 20장을 4개의 클래스로 구성하여 학습한 결과, 30분의 학습 시간을 가져야 Loss가 0.09아래로 수렴하는 것을 확인하였다. 다음 사진은 그 학습 과정에서 Loss의 값을 브라우저의 콘솔로 출력한 결과이다.

Epoch - 학습의 세대로, 학습의 반복 횟수를 의미한다.

totalLoss[index] - 데이터 셋에서 Index번째의 사진에 대한 전체 손실을 의미한다.

Epoch 0

```
epoch : 0
ground truth boxes: 1 (416)
noObjectLoss[0]: 211
objectLoss[0]: 0.002
coordLoss[0]: 0.6391
classLoss[0]: 0.8271
totalLoss[0]: 212.4683
```

Epoch 1

```
epoch : 1
ground truth boxes: 1 (416)
noObjectLoss[0]: 38.2188
objectLoss[0]: 0.4015
coordLoss[0]: 0.6686
classLoss[0]: 0.7737
totalLoss[0]: 40.0628
```

Epoch 2

```
epoch : 2
ground truth boxes: 1 (416)
noObjectLoss[0]: 0
objectLoss[0]: 0.5872
coordLoss[0]: 0.9587
classLoss[0]: 0.8013
totalLoss[0]: 2.3474
```

III. 모델 테스트

학습 결과 Epoch 1까지 큰 Loss가 발생하지만 Epoch 2부터 Loss가 급격하게 줄어들었다. 그러나 이후 Epoch가 늘어나도 Loss의 극적인 변화는 없었다.

Epoch가 50이 되었을 때에도 Epoch가 1~2일 때와 비슷한 Loss값을 가졌기 때문에 Loss가 수렴했다고 가정하고, 해당 모델을 저장하고 테스트를 해보았으나 아무런 객체도 탐지해내지 못하였다.

Epoch를 150까지 늘려 Loss가 더 줄어드는 지 관찰하였고, Loss가 초기보다 더 줄어든 상태까지의 학습이 가능한 것을 확인했다. 이 때의 Loss는 평균 0.05보다 아래의 값들을 가졌다.

따라서 CheckPoint를 저장한 후 해당 모델을 브라우저 상에 불러와 테스트를 진행하였다. 결과는 다음과 같다.



학습한 이미지와 전혀 다른 이미지로 테스트하였을 때, 해당 객체의 클래스를 탐지했고 그에 따른 Score Threshold 값을 계산했다. 여기서 Score Threshold란 '해당 객체를 몇 퍼센트만큼 확신하는 가'이다.

VI. 문제점

데이터의 양 : 얼마나 많은 데이터를 학습 시켜야 하는가?

- 이미지의 개수와 클래스에 따라서 학습 시간이 달라진다. GPU 가속이 없기 때문에 방대한 양의 데이터 셋을 학습하는 데에 한계가 있고 이를 사용자가 알 수 없다. 따라서 데이터의 양에 제한을 두어야 한다.

과적합의 문제 : 얼마나 길게 학습 시켜야 하는가?

- 학습 시간이 길게되면 과적합이 발생하는데, 사용자가 과적합이 언제 발생하는 지 알 수가 없다. 또한 사용자가 입력한 데이터 셋에 따라 과적합이 발생하는 시기도 다르다. 따라서 과적합을 방지하기 위해 내부적으로 Javascript 상에서 추가적인 설계가 필요할 것 같다.

VII. 향후 계획

- 웹 캠을 사용하여 이미지를 캡처하고 Annotating 할 수 있는 HTML 페이지 추가
- Model Configure를 사용자가 추가한 데이터 셋에 따라 생성되는 프로세스 개발
- 학습 시, Loss 그래프를 시각화하여 제공
- 실시간으로 웹 캠을 통하여, 자신이 학습시킨 AI 모델을 테스트할 수 있는 데모 개발