

Deep Learning

Data Labeling

강사 양석환



딥러닝과 데이터 라벨링

AiDA
Lab.



- 딥러닝에서 가장 중요한 것은 데이터(특히 학습을 위한 데이터)
- 같은 계열의 목적을 가진 모델이라면
 - 코드의 변형없이 데이터만 교체해서 서로 다른 분야에 적용할 수 있음
- CNN을 비롯한 영상인식, 분류 등을 목적으로 하는 딥러닝 모델은
 - 학습의 결과가 옳은지, 그른지 확인할 수 있도록 각각의 영역을 구분하고 표시를 해 둠
 - 이 작업 과정이 **라벨링(Labeling)** 작업 과정

MNIST ?

데이터가 없다!!!

어쩌라고?



Dataset
Download

ImageNet ?

**실력 / 기술이 없어서 못하는 것이 아니라
해 보지 않아서 몰라서 못하는 것이다.**

차량 번호판 인식용 학습 데이터셋 만들기

AiDA
Lab.



• 참여자

- 포항공대 2~3학년 인턴사원 3명
- AI / 프로그래밍에 대한 기본적인 교육, 실습은 학교에서 이수
- 데이터셋 개발 경험 없음
- 실습 프로젝트 경험은 있으나 실무 관련 프로젝트 경험은 없음

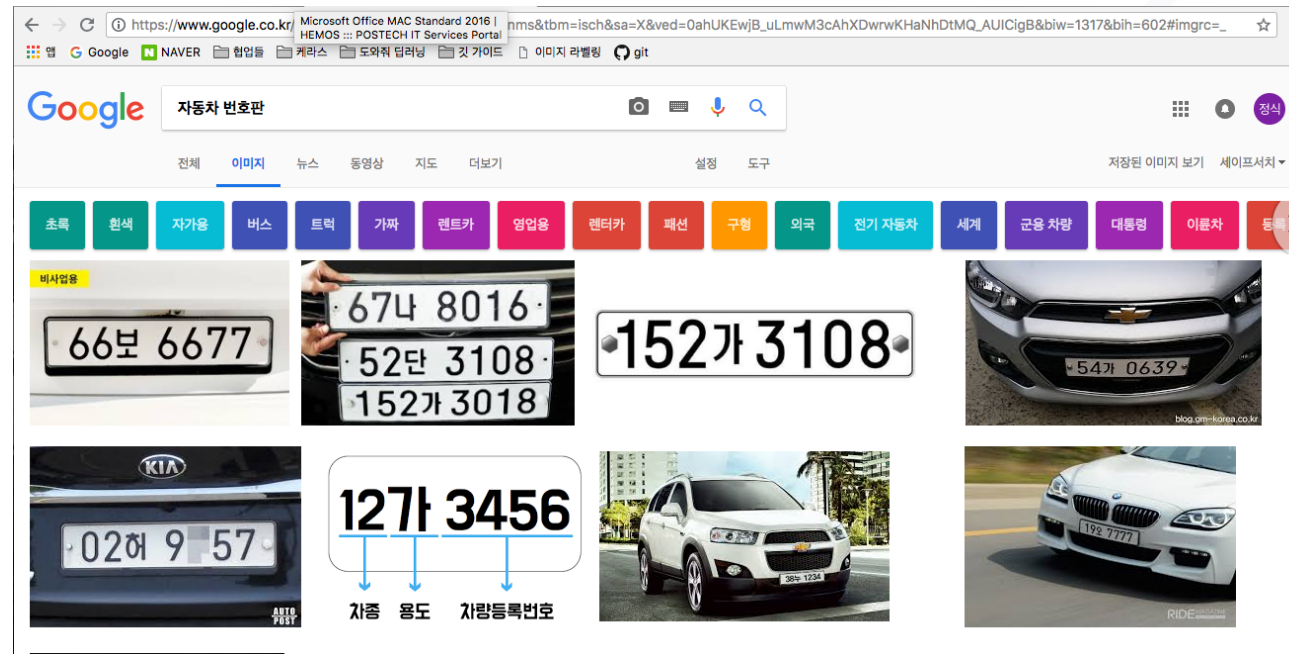


- **프로젝트의 내용**
 - 실시간(동영상)으로 자동차 번호판 검출하기
- **필요한 데이터 확인**
 - 번호판을 장착한 자동차의 사진
 - 자동차의 사진에서 번호판 영역에 적용된 레이블링 데이터
 - 다양한 형태의 번호판 사진
 - 기타 학습 조건들...



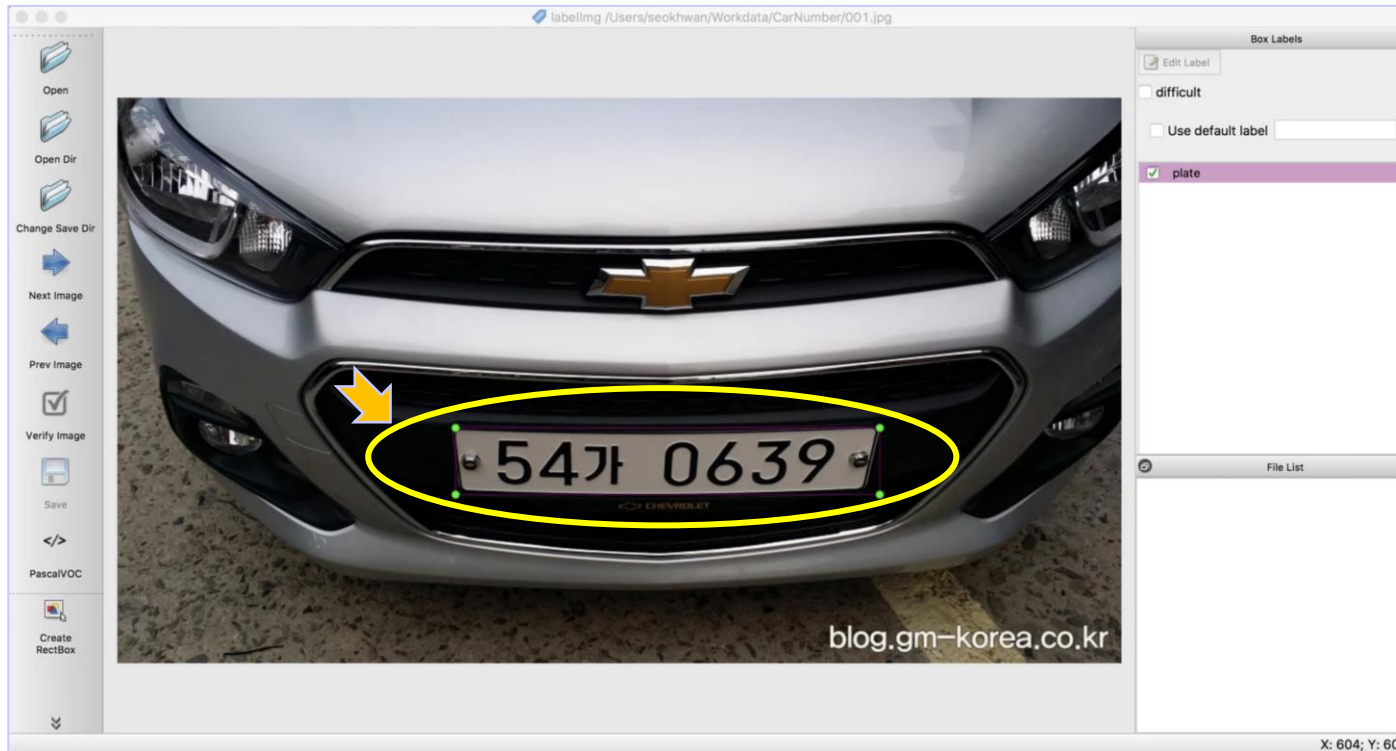
• 이미지데이터 수집

- Google 검색을 통하여 약 50개의 차량 이미지 수집
- 전체 프로세스에 대한 과정을 처음 시도함
- 진행 방법이 올바른가에 대한 확인은 불가능한 상태



- 이미지 라벨링 작업

- 라벨링 도구: LabelImg



```
<annotation>
  <folder>CarNumber</folder>
  <filename>001.jpg</filename>
  <path>/training/001.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>980</width>
    <height>551</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>plate</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>330</xmin>
      <ymin>321</ymin>
      <xmax>744</xmax>
      <ymax>386</ymax>
    </bndbox>
  </object>
</annotation>
```

- Detection을 위한 딥러닝 모델 적용(1차)

- Faster R-CNN 모델

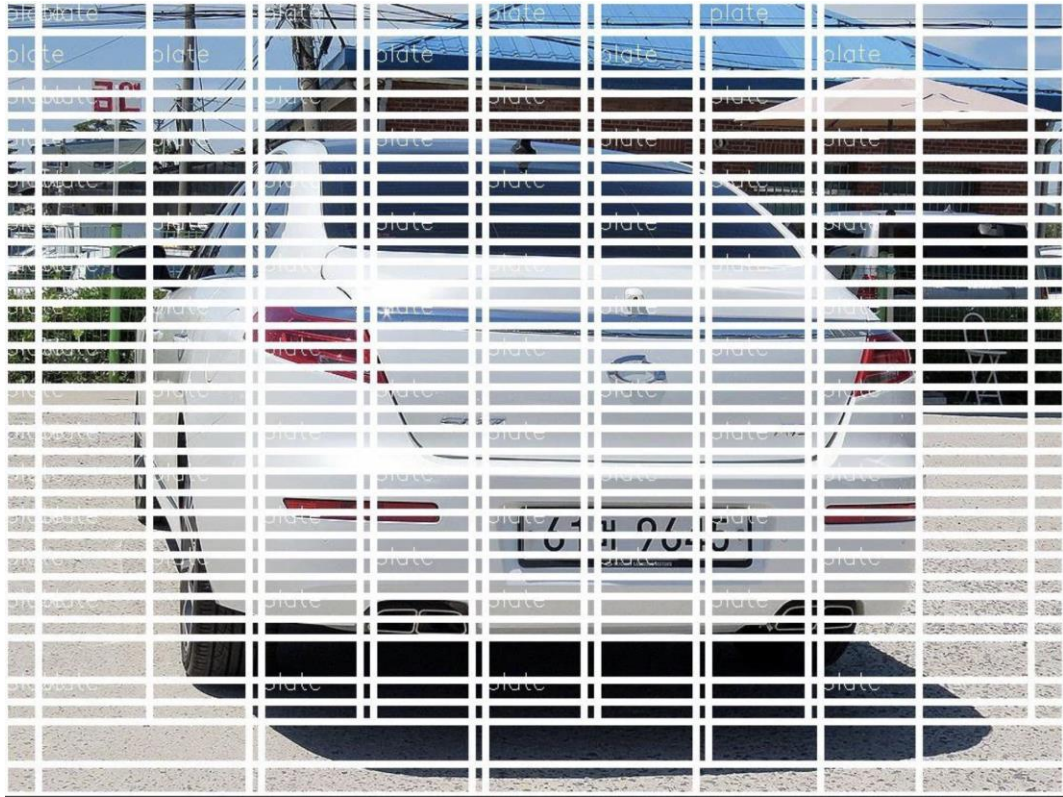


성공!!

그러나 정지영상에 대한 테스트
→ 동영상은 모델이 느려서 적용 불가
→ YOLO 모델을 적용해보기로 결정

- Detection을 위한 딥러닝 모델 적용(1차)

- YOLO 모델













• Detection을 위한 딥러닝 모델 적용(2차)

• 원인 파악

- Faster RCNN 모델에서 인식을 성공하였으므로 작업의 방향성은 맞는 것으로 간주함
- 데이터 부족을 의심함 → 데이터 추가, 학습 횟수 추가 결정
 - 중고차 매매 사이트를 통해 100개의 고화질 이미지 확보
 - 새로 추가된 각 이미지에 대하여 라벨링 작업 수행
 - 100개의 이미지를 이용하여 1,000번의 학습 수행

비교하기

사진정보	차명(모델명)/사고내역	미션1↓	년식1↓	연료1↓	주행거리1↓	가격1↓
<input type="checkbox"/>	 [혜보레] 크루즈 1.8LT 코드가	오토	2014 년식	휘발유	111,977 Km	660 만원
<input type="checkbox"/>	 [기어] 올뉴쏘울 1.6 디젤 노블레스 기본형 ★무사고★입고후 연진오일교환★UVO	오토	2014 년식	경유	57,492 Km	1,360 만원
<input type="checkbox"/>	 [혜보레] 크루즈 1.8LTZ + ★무사고급★1년키로수무제한보증(오카워)	오토	2014 년식	휘발유	25,074 Km	1,100 만원
<input type="checkbox"/>	 [혜보레] 크루즈 2.0LTZ + (The Perfect Black) 무사고,완무,영연 차량.	오토	2014 년식	경유	98,995 Km	990 만원
<input type="checkbox"/>	 [현대] 더 뉴 아반떼 1.6 GDI 모던 스마트키//네비//윈투프//전국 최저	오토	2014 년식	휘발유	88,676 Km	829 만원
<input type="checkbox"/>	 [현대] 더 뉴 아반떼 1.6 GDI 스마트 무사고,1인신조소유,정품네비게이션,하	오토	2014 년식	휘발유	80,780 Km	910 만원
<input type="checkbox"/>	 [상성] 뉴SM3(신형) SE ▶ 무사고, 1인신조, 짧은주행거리.	오토	2014 년식	휘발유	40,451 Km	860 만원
<input type="checkbox"/>	 [혜보레] 크루즈 2.0LT + 무사고	오토	2014 년식	경유	135,773 Km	710 만원
<input type="checkbox"/>	 [혜보레] 크루즈 2.0LTZ + 신조/완전무사고/코드입금/윈투프/네비	오토	2014 년식	경유	34,235 Km	1,130 만원
<input type="checkbox"/>	 [기어] K3 1.6GDI 트렌디 ★완전무사고 (100%무사고)	오토	2014 년식	휘발유	78,621 Km	1,030 만원

• Detection을 위한 딥러닝 모델 적용(2차)

• 데이터 추가, 라벨링 작업

The image displays two overlapping windows. The left window is a car information page for a '쌍성 뉴SM3(신형) SE'. It lists details such as license plate '20수 8267', registration year '2014', and mileage '40,451 Km'. The right window is a 'labeling' software interface showing a red Hyundai car with a bounding box around its license plate '0러 4038'. The interface includes a file list on the right and a 'Box Labels' panel with a 'plate' label selected.

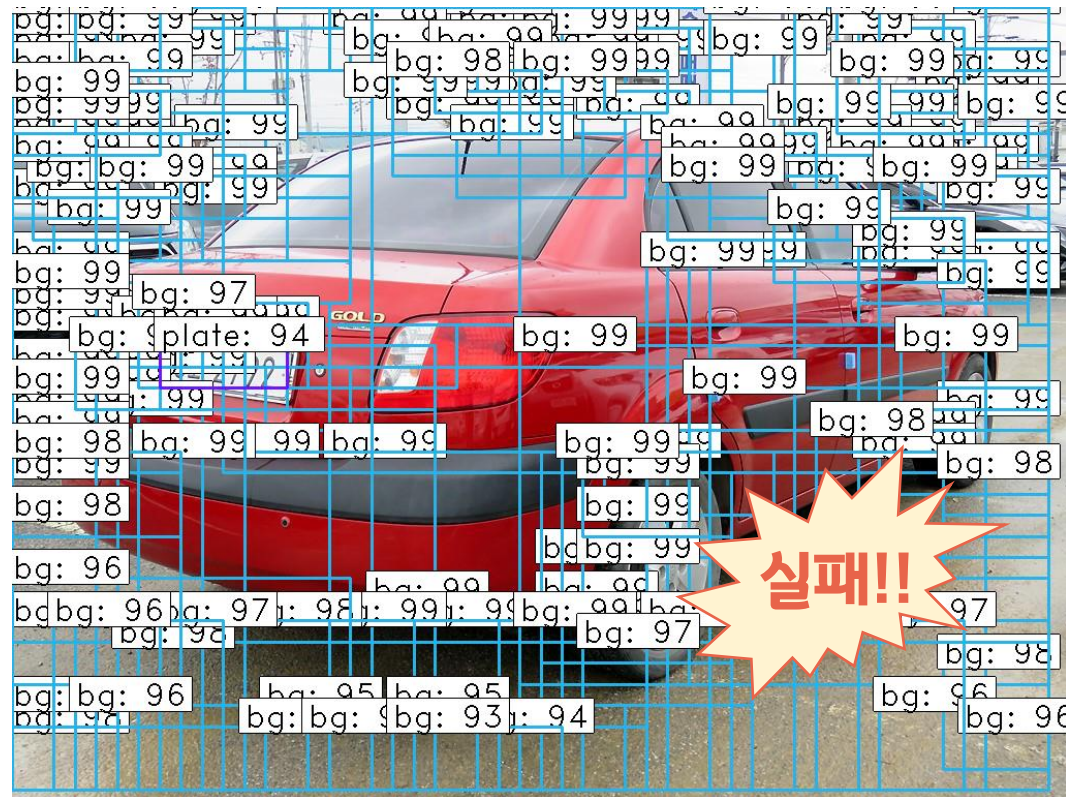
항목	내용
상성	뉴SM3(신형) SE
차량번호	20수8267 > 사고이력조회하기 판매가: 8
년형/등록	2014 년형 2014.01 최초등록
연료	휘발유
변속기	오토
주행거리	40,451 Km
색상	검정
차대번호	
압류/저당	
제시번호	
상능번호	
사고정보	
조합정보	
판매자	

- Detection을 위한 딥러닝 모델 적용(2차)
 - 추가 데이터 적용 결과

Faster RCNN 모델



YOLO 모델



• Detection을 위한 딥러닝 모델 적용(3차)

• 실패 원인 분석

- 사용하는 데이터 구조가 모델 별로 다름
 - Faster RCNN 사용 데이터: XML
 - YOLO 사용 데이터: txt
- 데이터 구조의 차이에 대한 설명은 없었음
- 다수의 사이트, 교재에서는 데이터 셋 개발에 대하여 아예 다루지 않거나 다운로드 데이터에 대해서도 Labellmg 툴을 사용한다고만 표시

Faster RCNN용 XML 데이터

```
<annotation>
  <folder>CarNumber</folder>
  <filename>001.jpg</filename>
  <path>/training/001.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>980</width>
    <height>551</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>plate</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>330</xmin>
      <ymin>321</ymin>
      <xmax>744</xmax>
      <ymax>386</ymax>
    </bndbox>
  </object>
</annotation>
```

YOLO용 TXT 데이터

```
0 0.611328 0.700521 0.400391 0.640625
```

- Detection을 위한 딥러닝 모델 적용(3차)

- Faster RCNN 모델 → YOLO 모델

- 데이터 변환기를 개발하여 Labellmg용 XML 데이터를 YOLO용 TXT 데이터로 변환함

```
1 import xml.etree.ElementTree as ET
2
3 for i in range(1674,1676): # 파 이 로 이 레 - 0 시 로 지
4     if (i<10) :
5         filename = "00" + str(i) + '.xml'
6     elif (i<100) :
7         filename = "0" + str(i) + '.xml'
8     else :
9         filename = str(i) + '.xml'
10
11 # parse xml file
12 doc = ET.parse('/Users/exem/learning_data/xml_only_plate/' + filename)
13 # get root node
14 root = doc.getroot()
15
16 for country in root.iter("size"):
17     width = country.findtext("width")
18     width = float(width)
19     height = country.findtext("height")
20     height = float(height)
21
22 for xy in root.iter("bndbox"):
23     xmin = xy.findtext("xmin")
24     xmin = float(xmin)
25     ymin = xy.findtext("ymin")
26     ymin = float(ymin)
27     xmax = xy.findtext("xmax")
28     xmax = float(xmax)
29     ymax = xy.findtext("ymax")
30     ymax = float(ymax)
31     # 0 0.276563 0.661806 0.132812 0.118856
32
33     xmin = round((xmin/width),6) # 소 수 지수 마 레 6 자 리 파 지 나
34     ymin = round((ymin/height),6)
35     xmax = round((xmax/width),6)
36     ymax = round((ymax/height),6)
37
38     width = xmax - xmin
39     height = ymax - ymin
40
41     xmin2 = xmin + (width/2)
42     ymin2 = ymin + (height/2)
43
```

Faster RCNN용 XML 데이터

```
</source>
<size>
  <width>980</width>
  <height>551</height>
  <depth>3</depth>
</size>
<segmented>0</segmented>
<object>
  <name>plate</name>
  <pose>Unspecified</pose>
  <truncated>0</truncated>
  <difficult>0</difficult>
  <bndbox>
    <xmin>330</xmin>
    <ymin>321</ymin>
    <xmax>744</xmax>
    <ymax>386</ymax>
  </bndbox>
</object>
</annotation>
```

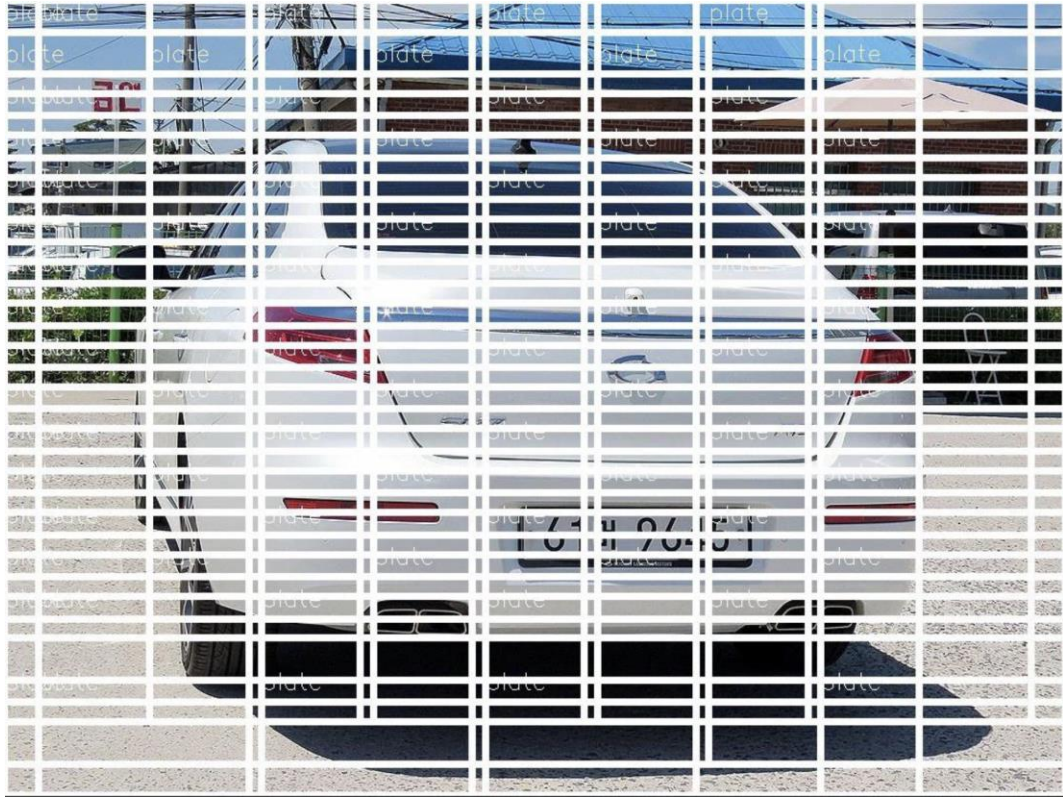
YOLO용 TXT 데이터

0 0.611328 0.700521 0.400391 0.640625

이미지 크기를 기준으로 하여
0~1 값으로 정규화, 파일 변환

- Detection을 위한 딥러닝 모델 적용(3차)

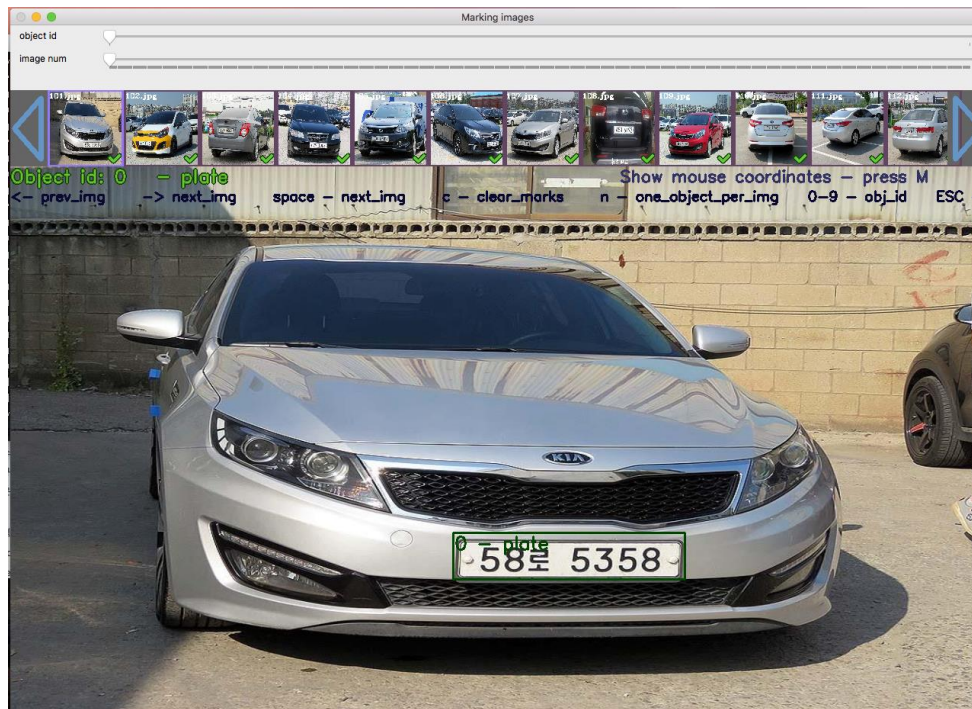
- YOLO 모델



AiDA
Lab.

- Detection을 위한 딥러닝 모델 적용(4차)

- 그냥 처음부터 라벨링을 새로 하기로 결정함
- 검색 결과 YOLO 전용 도구(YOLO-Mark) 발견 → 라벨링 데이터 재작성

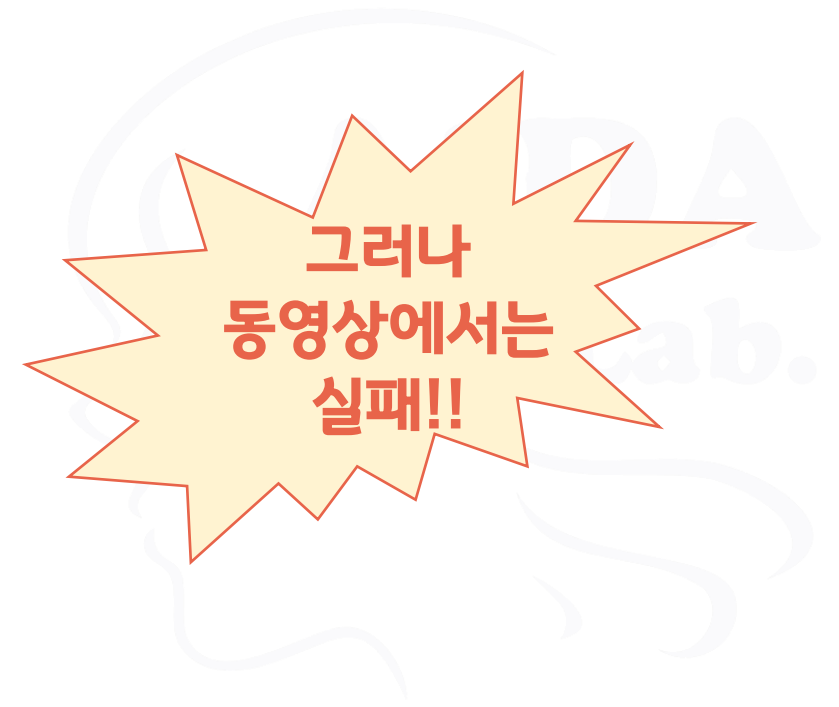


현재의 Labellmg 버전에서는
YOLO 데이터 타입 생성을 지원함

- Detection을 위한 딥러닝 모델 적용(4차)

- YOLO 모델 적용 결과

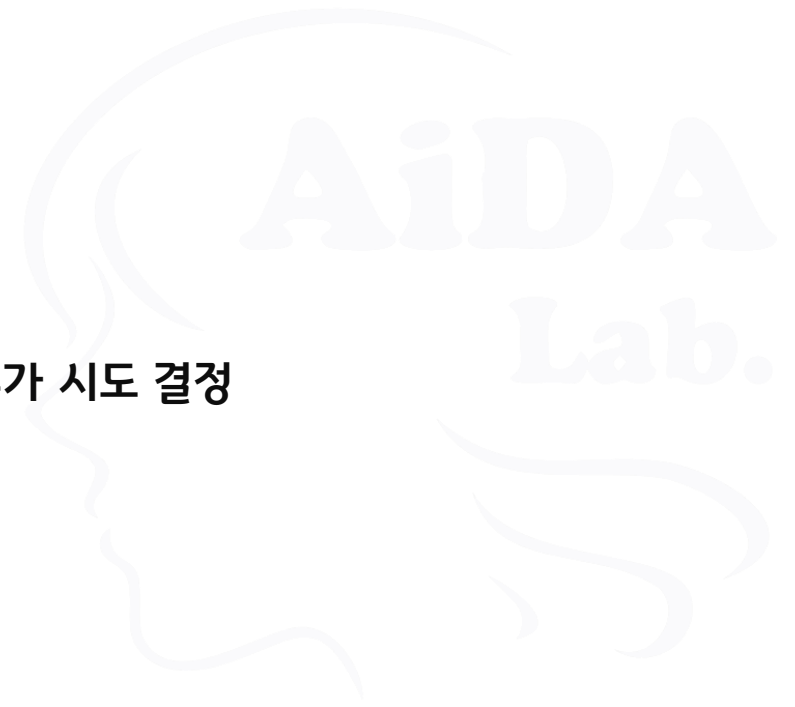
- 정지영상 → 성공



• Detection을 위한 딥러닝 모델 적용(5차)

• 원인 분석

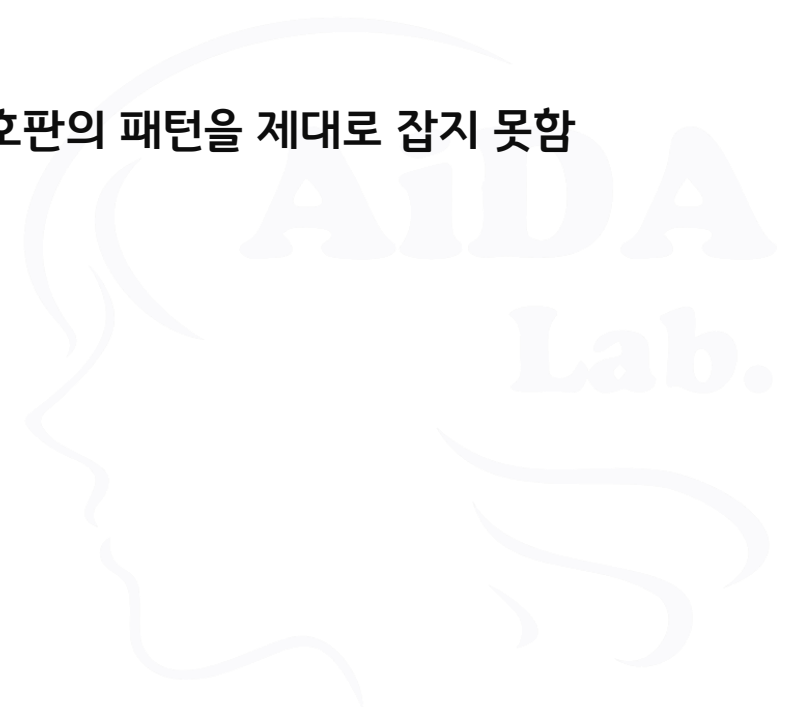
- 100개의 이미지를 이용하여 1,000번의 학습 수행
 - 임계값: 80% → 동영상에 포함된 차량번호판의 약 10%정도 인식
 - 차량번호판이 아닌 유사 패턴 발견 시 → 차량번호판으로 오인식
 - 인식성공률은 낮지만 대체로 안정적(꾸준히 10%대... ππ)
 - 정지영상(이미지)에 대한 인식성공률은 약 90% → 안정적
- 데이터 분량 부족이 원인일 가능성 제시 → 데이터 추가 및 학습 횟수 추가 시도 결정



- **Detection을 위한 딥러닝 모델 적용(5차)**

- **데이터 추가, 학습 횟수 추가 작업**

- 1,675개의 이미지를 이용하여 2,000번의 학습 수행
- 임계값: 80% → 정지영상 인식 성공(약 90%), 동영상 인식 실패
- 인식하고자 하는 차량번호판의 범위가 지나치게 크게 설정되어 차량번호판의 패턴을 제대로 잡지 못함



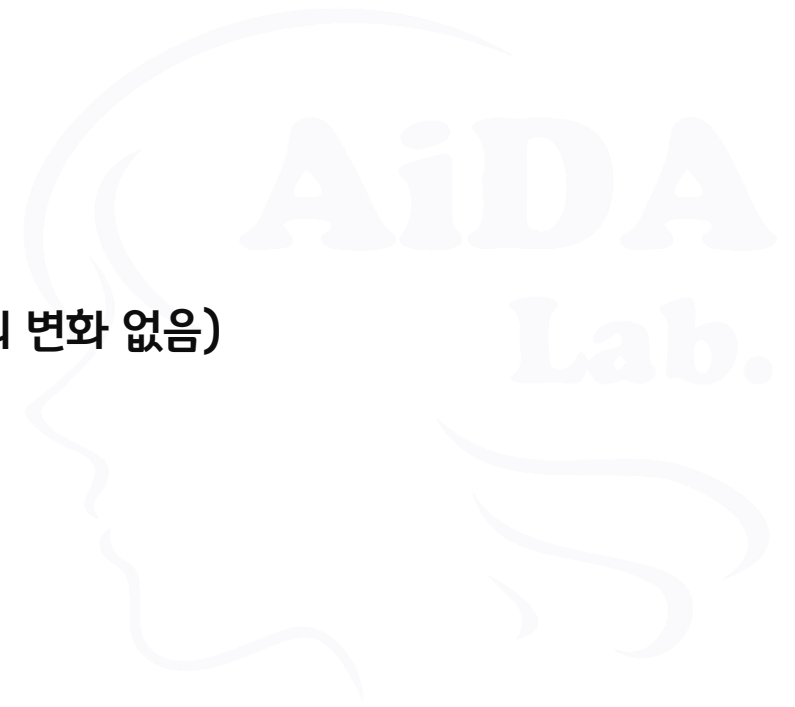
• Detection을 위한 딥러닝 모델 적용(6차)

• 원인 분석

- 증가한 학습대상 이미지에 비하여 지나치게 적은 학습 횟수가 원인일 수 있다고 판단 → 학습 횟수 추가 결정

• 학습 횟수 추가 작업

- 1,675개의 이미지를 이용하여 45,000번의 학습 수행
- 임계값: 80% → 정지영상 인식 성공(약 90%), 동영상 인식 실패 (거의 변화 없음)
- 임계값: 30% → 동영상에서 일부 차량번호판 인식 성공
- 인식 기대값이 낮게 계산됨 → 처음부터 인식을 시도하지 않음



- **Detection을 위한 딥러닝 모델 적용(7차)**

- **원인 분석**

- 너무 선명한 이미지만을 학습대상으로 사용한 것이 원인일 가능성 제시
- 동영상에서 캡처한 이미지를 일부 학습에 적용하기로 결정

- **데이터 추가(from 동영상) 작업**

- 동영상에서 캡처한 338장의 이미지 → 레이블링 작업 추가 진행 → 2,013개의 이미지로 2,000번의 학습 수행
- 임계값: 50% → 인식 실패
- 임계값: 30% → 인식 수량 증가 확인
 - 중복 인식 증가, 유사패턴에 대한 오인식 증가, 어긋난 위치에 대한 인식 표시 증가

• Detection을 위한 딥러닝 모델 적용(8차)

• 원인 분석

- 임계값 변경에 따른 인식을 변화 확인 → 학습 횟수의 증가가 필요하다고 판단

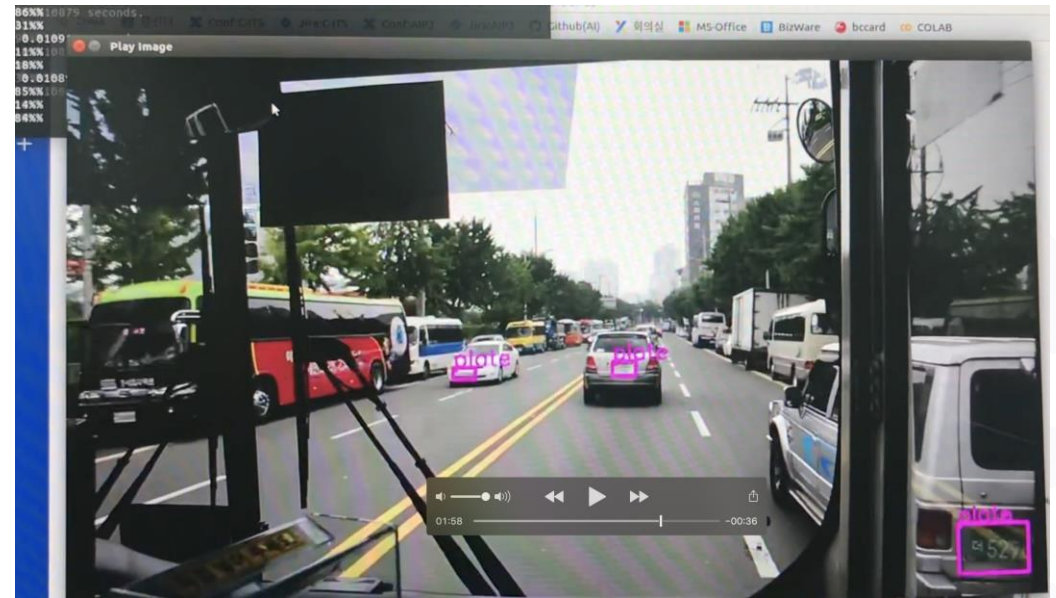
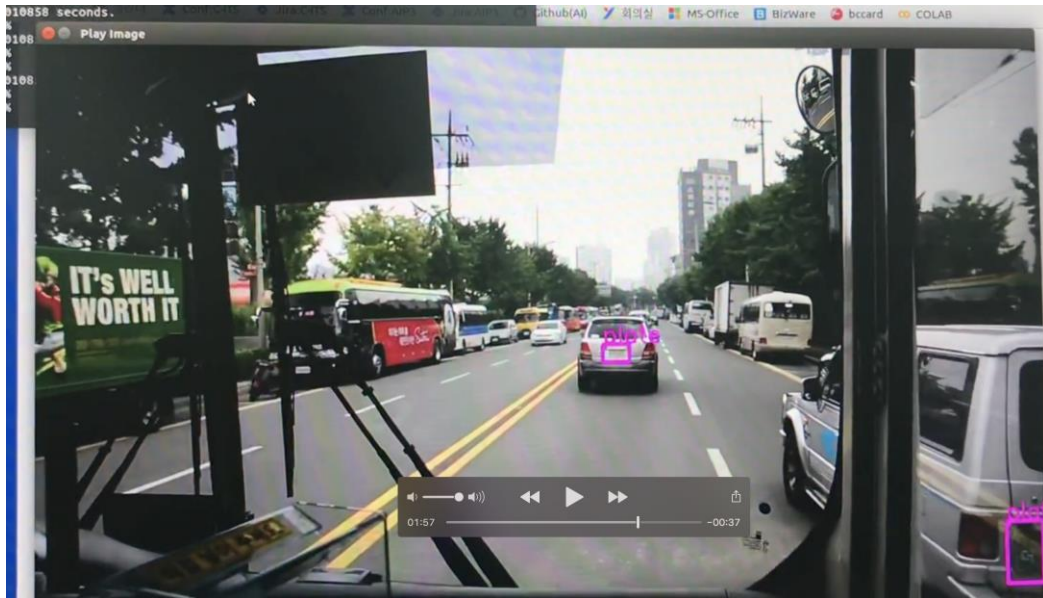
• 학습 횟수 추가 작업

- 2,013개의 이미지를 이용하여 45,000번의 학습 수행
- 임계값: 60% → 웬만한 경우는 인식 성공(기대값 평균 65%~86%)
- 차량번호판이 아닌 유사 패턴 발견 시 → 오인식 거의 없음
- 거의 정확한 번호판의 위치 표시



- Detection을 위한 딥러닝 모델 적용(8차)

- 적용 결과



일부만 표시된 차량번호판도 인식 성공

• Labeling 영역의 지정에 대한 규칙 필요

- 인식하고자 하는 대상과의 거리
- 시선의 초점으로부터 허용되는 좌우 범위(각도)
- 시선의 초점으로부터 허용되는 상하 범위(각도)
- 인식 대상에 대한 선명도(흐릿함)의 허용 범위
- 인식 대상에 적용되는 밝기의 허용 범위
- 인식 대상의 크기는 다양하게 적용 가능하도록 이미지 선택

정확한 수치 규정일 필요는 없음
대략적인 느낌을 통해 적용

- 사전 고려사항 예시 1



- 사전 고려사항 예시 2



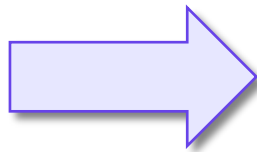
• 사전 고려사항 예시 3



• 사전 고려사항 예시 4



이런 이미지만 학습하면



이런 크기의 대상은 인식하지 못하는 경우가 증가함

- **영상의 전처리 범위에 대한 규칙 필요**

- 입력 영상의 주변 환경 (실내, 실외, 밝음, 흐림 등)을 기준으로 입력 영상에 대한 표준 밝기
- 입력 영상의 해상도를 기준으로 하는 이미지의 표준 크기
- 그 외, 특정 환경에서의 인식이 필요한 경우, 해당 환경에 맞는 밝기, 크기 등의 사전 규칙 결정



- 테슬라 AI 디렉터, 안드레아 카파시의 인터뷰 내용에서...



4년이 지났는데도 레이블링 작업흐름이 아직 "해결"되지 않았습
니다. 레이블링, QA, 최종 QA, 자동 레이블링, 오류 발견, 다양성
마사지, 레이블링 문서 + 버전 지정, PPL 교육, 에스컬레이션, 데
이터 정리, 처리량 & 품질 통계, 평가 세트 + 분류 & 부스팅, ...

- 이미지나 영상 내의 객체를 레이블링 하는 것이 레이블링 작업의 전부라고 단순히 생각하기 쉽다.
- 그러나 안드레아가 나열한 항목들을 보면 레이블링 워크플로우(labeling workflows)는 그보다 더 많은 작업을 수반하고 있음을 알 수 있다.

• 라벨링 워크플로우

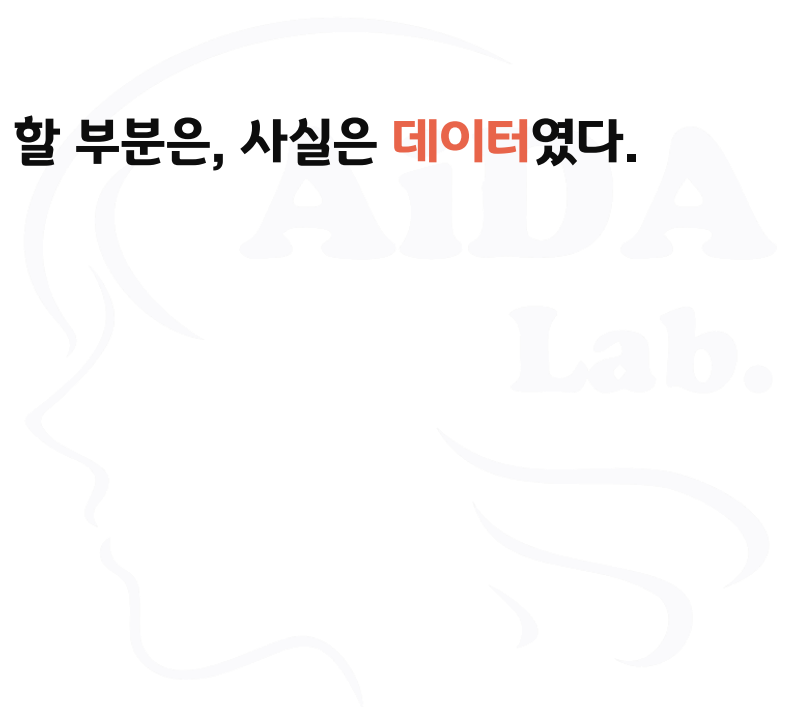
• 라벨링 워크플로우를 구성하는 작업들

- 데이터에 대한 라벨링 유형 기준 마련
 - 라벨링 작업 결과물에 대한 품질관리(QA), 피드백 수집 과정 구축
 - 라벨링 작업자의 훈련과 퍼포먼스 측정 등을 포함하는 인력관리
 - 라벨링 과정 중 이슈 발생시 커뮤니케이션 비용 관리
 - 라벨링을 통해 구축된 데이터셋의 버전관리 등
- 나열된 것은 라벨링 작업 중 예상되는 워크플로우 범위 중 일부일 뿐...



• 레이블링 작업의 중요성

- 이미지, 영상 데이터의 레이블링 작업은 가장 덜 중요한 것으로 인식
- 그러나 컴퓨터 비전 문제를 푸는 데 있어서 가장 선도적인 테슬라 팀이 여전히 난항을 겪고 있는 부분이 라벨링 워크플로우라고 지적
- 상용화 단계에서 컴퓨터 비전 문제를 해결하기 위해 우리가 주목해야 할 부분은, 사실은 **데이터**였다.



• 컴퓨터 비전이 데이터를 다루기 어려운 이유

• 사람은 이미지를 보지만 컴퓨터는 숫자를 본다

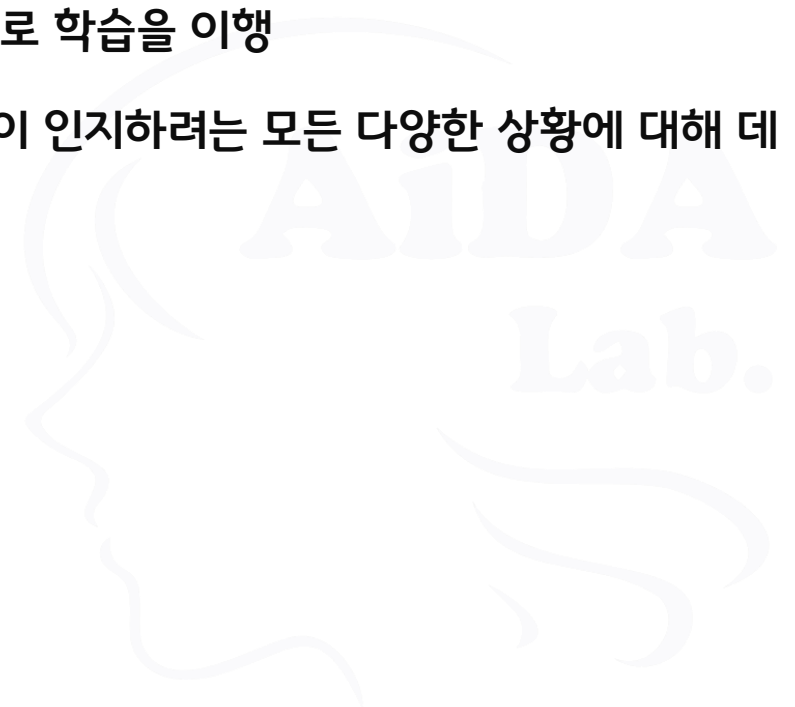
• 저명한 컴퓨터 비전 전문가, UC버클리 교수 Jitendra Malik(지텐드라 말릭)

- 사람이 이미지를 인식하는 과정이 무의식적으로 혹은 잠재 의식적으로 진행되기 때문에 흔히 컴퓨터가 비전을 처리하는 방식도 매우 쉬울 거라고 착각할 수 있다고 지적
- 실제 사람의 경우에도, 대뇌피질의 상당 부분은 이미지 프로세싱 처리에 집중
- 컴퓨터가 이미지를 처리하는 방식은 인간과 다름
- 인간이 매우 시각적이고 직관적인 방식으로 이미지를 인식하는 반면, 컴퓨터는 이미지의 모든 부분을 개별 픽셀로 환산하여 숫자로 인식
- 이미지를 인식할 때마다 처리해야 하는 데이터 양이 많다는 것을 의미
→ 복잡한 시각적 작업을 수행하기 위한 계산 및 데이터 리소스 또한 방대해짐을 의미

- **컴퓨터 비전이 데이터를 다루기 어려운 이유**

- **사람보다 더 많은 데이터를 필요로 하는 컴퓨터 비전 시스템**

- 현재 컴퓨터 비전 분야를 이끌어 가는 가장 고도화된 기법은 지도학습
- 지도학습은 '정답'이라고 여겨지는 특성 값을 라벨링한 데이터를 기반으로 학습을 이행
- 학습에 사용된 데이터를 기반으로 모델의 성능이 결정되기 때문에 모델이 인지하려는 모든 다양한 상황에 대해 데이터로 준비해서 학습시켜야 하는 것



- **이미지 라벨링 및 데이터 처리를 위한 기준은 어떻게 정리할 수 있나?**
 - 컴퓨터 비전 분야에 있어서 세계적으로 저명한 학회인 CVPR 21에서 안드레아 카파시의 발표
 - 기계가 이해하기 어려운 상황 혹은 자연스럽게 얻을 수 있는,
 - 데이터 수는 적지만 실생활에서 충분히 일어나는 상황 등을 221가지로 추려 나열한 후,
 - 해당 케이스의 데이터를 집중적으로 확보 → 엣지 케이스(Edge Case)
 - 확보된 엣지 케이스를 4개월 동안 7차례 반복 테스트 수행 중
 - 엣지 케이스를 넣어 학습을 반복하며 모델의 성능을 높이는 것
 - 데이터를 통해 모델의 성능을 높이는 전형적인 Data-centric(데이터 중심) 방법론

CVPR 21(Conference on Computer Vision and Pattern Recognition)

- 이미지 라벨링 및 데이터 처리를 위한 기준은 어떻게 정리할 수 있나?

- 결국 비전 데이터를 다루는 문제를 풀고 있는 기업에서는

- 모델 성능이 일정 수준에 도달하면

- 이렇게 데이터를 중심으로 하는 반복사이클을 지속할 수 있는 견고한 머신러닝 데이터 플랫폼에 대한 필요성을 절감할 수 밖에 없음

또한 이러한 사실은 이미지 / 비전 데이터에 국한되지 않고 딥러닝을 위한 모든 데이터의 영역에서 동일하게 적용됨

**THANK
YOU**

