

1. 자율주행 인지에 관련된 Data Set

1. BDD100K

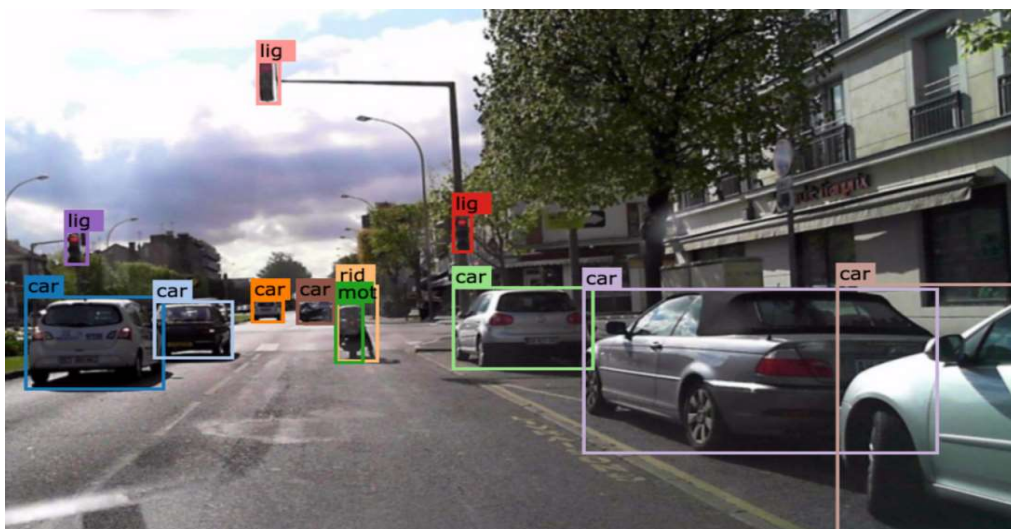


최근 UC버클리 인공지능연구 실험실에서 BDD100K라고 불리는 데이터베이스를 공개하고 데이터셋 개발과 관련된 논문을 발표했다. BDD100k는 Berkeley Deep Drive의 약자로 40초의 비디오 시퀀스, 720픽셀 해상도, 30FPS의 고화질 비디오시퀀스 10만개로 구성된다.

주행 환경구현, GPS 정보, IMU 데이터 및 타임 스탬프가 포함되어 있다. 녹화된 비디오는 비오는 날씨, 흐린 날씨, 맑은 날씨, 안개와 같은 다양한 날씨 조건이 기록되어 있고 데이터 세트는 낮과 밤이 적절한 비율로 기록되어 있다.

위사진을 보면 쉽게 주석 처리하기 위해, 수직 차선은 적색, 평행 차선은 청색으로 구분하였다. 적색 표시 있는 운전 경로와 청색 표시가 있는 대안 운전 경로로 주행 가능 구역을 구분한다.

데이터셋은 버스, 신호등, 교통 표지, 사람, 자전거, 트럭, 모터, 자동차, 기차 및 라이더를 위해 100,000개 이미지에 주석이 달린 2D Bounding Box가 포함되어 있다.



위의 사진처럼 사물이 어떤 것인지 구별하고 인지하기 위해 엄청난 양의 데이터셋이 있다. 이 데이터는 도로보행자 탐지를 위해 사용할 수도 있고 이를 위해 85000개가 넘는 보행자 데이터가

있다. 이 외에도 차량용 컴퓨터 비전 및 기계 학습을 위해 현재 100만대의 자동차, 30만개가 넘는 표지판, 13만개의 보행자 등으로 구성 되어있다. 이런 방대한 양의 데이터셋은 다른 자율주행 데이터 셋보다 훨씬 규모가 크고 방대하다.

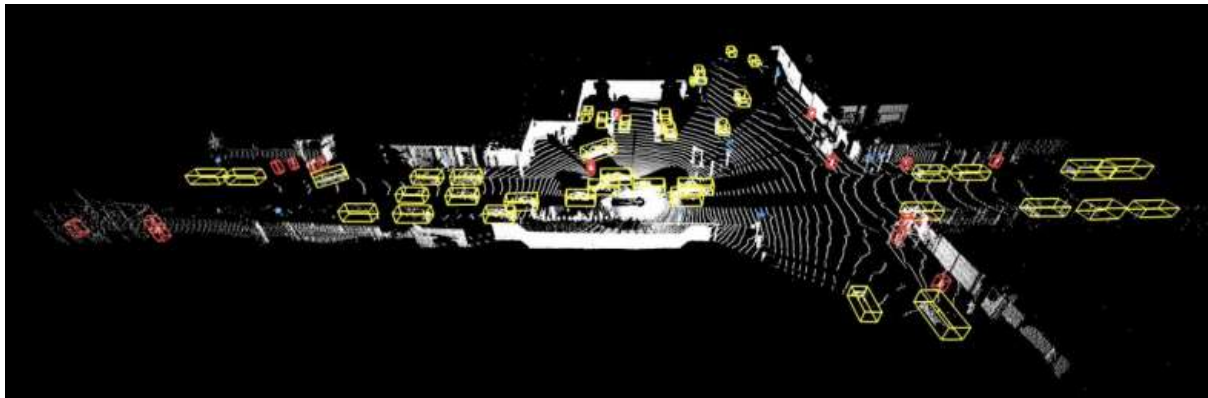
2. Mapillary Street-level Sequences Dataset

BDD100K는 차량과 사물 도로환경들을 주로 다뤘다면 Mapillary Street-level Sequences 데이터 셋은 주변환경과 건물에 관한 것이다. MSLS(Mapillary Street-level Sequences Dataset)는 도시 및 교외 장소를 위한 대규모 데이터셋이다. 이런 lifelong place 인식은 컴퓨터 비전 분야에서 중요하고 도전적인 과제이다. 이 데이터셋은 강력한 로컬라이제이션과 효율적인 대규모3D재구성을 응용하여 lifelong learning의 다양성을 보여주기 위해 설계되었다. MSLS데이터셋은 Mapillary협력 매핑 플랫폼에서 추출한 160만개 이상의 영상이 포함되어있다. 이 데이터셋은 6개 대륙 및 30개 주요 도시의 이미지, 서로 다른 관점의 수백대의 카메라 그리고 9년이라는 기간 동안 모든 계절에 걸쳐 수집한 것이다. 모든 이미지는 GPS와 나침반으로 위치를 찾아내며 높은 수준의 속성을 특징으로 한다.



3. Waymo open 데이터셋

구글의 연구진은 1150개의 장면(각각 약 20초의 길이)으로 구성된 대규모의 고품질 데이터셋과 다양한 도시 및 교외지역에서 캡처된 고품질 LIDAR와 카메라 데이터로 구성된 데이터 셋을 소개했다. 제시된 지리적 범위 지표에 근거하여 이용할 수 있는 초대형 카메라+라이더의 데이터 셋보다 웨이모 데이터셋의 다양성이 15배 더 높다고 한다. 데이터셋은 2D(카메라 이미지)와 3D(라이더) 바운딩 박스에 주석을 더 달았다. 프레임 전체에 걸쳐 일관된 식별자를 제공하고 3D 검출 및 추적 작업뿐만 아니라 2D에 대한 강력한 기초선을 제공한다.



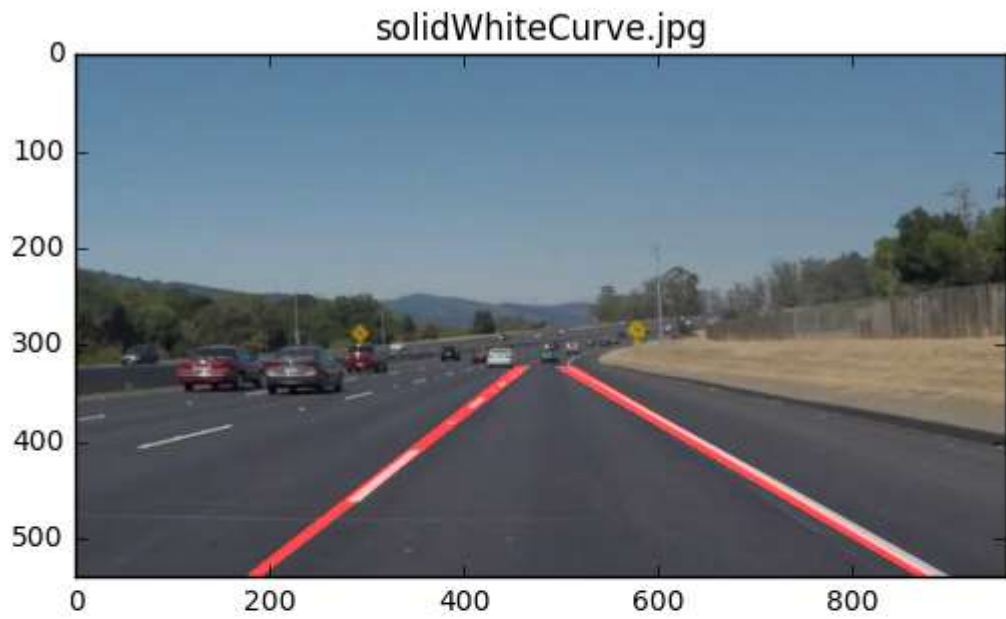
웨이모는 25개의 도시에서 약 1000만 마일에 달하는 자율주행테스트를 통해 데이터를 수집했다. 이 오픈데이터셋에는 1000개의 주행 세그먼트 데이터가 포함되어 있다. 각 세그먼트는 위에 말했듯이 센서 당 10HZ(20만 프레임)로 수집된 20초동안의 주행데이터이다.

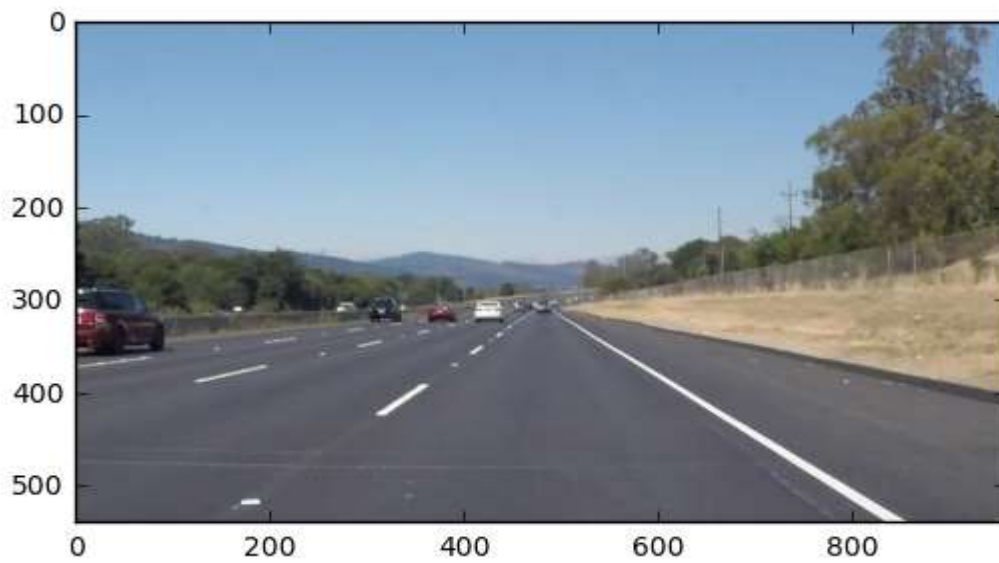
특히 자동차, 보행자, 자전거, 표지 4가지로 꼼꼼하게 구분해 표기한 라벨이 포함된 이미지와 라이더 프레임이 포함되어 있다.



2) 자율주행 인지에 관련된 2종 이상 Open Source 조사, 정리

(1) 차선 탐지(initial-lane-detector) Opencv





사진을 불러와서 사진에 대하여 학습 후 차선에 대한 정보를 불러온다. 이 학습한 것을 통해 동 영상에도 적용할 수 있다.

참고 :

<https://pinkwink.kr/1264>

<https://medium.com/@mrhwick/simple-lane-detection-with-opencv-bfeb6ae54ec0>

(2)CNN을 활용한 표지판 분류 모델

도로교통공단에서 표지판 이미지를 추출하여 데이터셋으로 사용하여 실제 표지판 사진으로 어떤 표지판인지 예측



```
result = model.predict(X)
label_to_str[np.argmax(result, axis=1)[0]]
```

executed in 7ms, finished 16:09:25 2020-06-07

'좌우로이중굽은도로'

위의 사진은 CNN으로 데이터셋을 이용하여 학습한 이미지가 아닌 실제 촬영 이미지 표지판이 어떤 것인지를 인지한다.

참고: [HowardHowonYu/traffic-sign-recognition: CNN을 활용한 표지판 분류 모델 \(github.com\)](https://github.com/HowardHowonYu/traffic-sign-recognition)

My git: [autonomus-driving/sign.ipynb at main · chans98/autonomus-driving \(github.com\)](https://github.com/chans98/autonomus-driving)

3) 2)의 정리한 코드 중 하나 실행해서 결과 확인

(1) 차선 탐지(initial-lane-detector) Opencv, computer vision

[autonomus-driving/line_detect.ipynb at main · chans98/autonomus-driving \(github.com\)](#)

위의 깃허브에 자세한 코드실행과정이 나와있습니다.

```
In [1]: # library
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import cv2
import numpy as np
%matplotlib inline

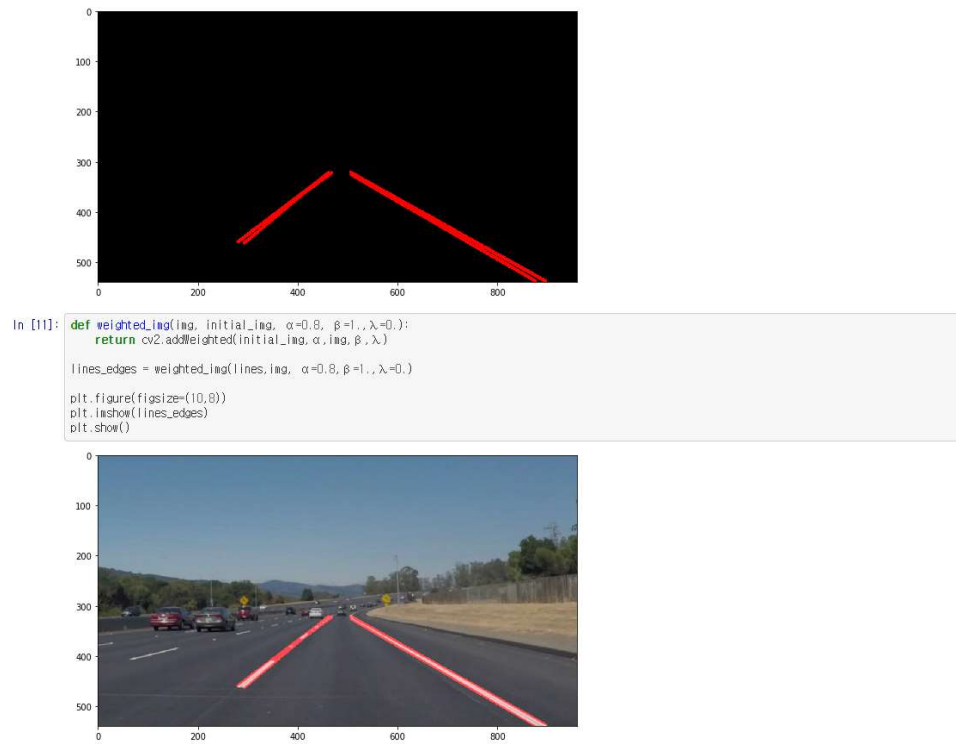
img = mpimg.imread('solidWhiteCurve.jpg')

plt.figure(figsize=(10, 8))
print('This image is:', type(img), 'with dimensions:', img.shape)
plt.imshow(img)
plt.show()
```

This image is: <class 'numpy.ndarray'> with dimensions: (540, 960, 3)



1. 원본사진 'solidwhiteCurve.jpg'로 가우시안 블러 처리하고 canny edge detection을 적용한다.
2. 그 다음 edge를 딴 다음 관심 영역을 재설정하고 검은색으로 채운다.
3. 관심을 가질 만한 영역을 잡아준다.
4. 하얀색 부분에 들어온 그림만 대상으로 선을 찾는다.
5. 관심영역을 vertices에 저장하고 반영한다.
6. 이제 직접 선을 그리는 함수를 만들어 둔다. 선을 그리는 함수로 HoughLinesP를 이용한다.
- 7.아래와 같이 나온다



4) Git 주소 및 간단설명

<https://github.com/chans98>

-autonomus-driving

Line_detect.ipynb : 차선탐지(코드 실행)

sign.ipynb : 표지판 탐지(코드 이해)

solidWhiteCurve.jpg : 차선탐지 이미지