

```
In [2]: cd
/home/notebook
```

```
In [5]: !python -m venv venv
```

```
In [13]: !source venv/bin/activate
```

## 1. 패키지 불러오기 및 경로 설정

```
In [5]: import pandas as pd
import numpy as np
import os
from PIL import Image
import random
from PIL import ImageDraw
import matplotlib.pyplot as plt
```

```
In [6]: !pwd
/home/work/sample-notebooks/Miso
```

```
In [7]: os.chdir('/home/work/sample-notebooks/train')
```

## 2. train/labeled\_data/images 확인

```
In [8]: tr_lb_img = os.listdir('./labeled_data/images')
tr_lb_img.sort()
del tr_lb_img[0]
tr_lb_img[:10]
```

```
Out[8]: ['sk_tr_000000.jpg',
'sk_tr_000001.jpg',
'sk_tr_000002.jpg',
'sk_tr_000003.jpg',
'sk_tr_000004.jpg',
'sk_tr_000005.jpg',
'sk_tr_000006.jpg',
'sk_tr_000007.jpg',
'sk_tr_000008.jpg',
'sk_tr_000009.jpg']
```

```
In [9]: path = './labeled_data/images/'
for i in tr_lb_img[:2]:
    im = Image.open(path + i)
    print(im.size)
    im.show()
```

(1920, 1080)



(1012, 800)  
2017-12-10 09:16:04.901 台GHJ688 车道:2



```
In [10]:  
path = './labeled_data/images/'  
size = set()  
for i in tr_lb_img[:100]:  
    im = Image.open(path + i)  
    size.add(im.size)  
size
```

```
Out[10]: {(320, 240),  
(932, 506),  
(982, 847),
```

```
(982, 1350),
(997, 1350),
(1001, 611),
(1012, 800),
(1200, 1664),
(1271, 1012),
(1920, 810),
(1920, 1080)}
```

### 3. train/labeled\_data/labels 확인

#### label 분포

- label 형식
  - header 없음
  - class\_id, x1, y1, x2, y2
  - x1,y1: bbox의 left-top corner의 x,y좌표값 (정규화한 값; x좌표/width, y좌표/height)
  - x2,y2: bbox의 right-bottom corner의 x,y좌표값 (정규화한 값)

#### 3-1) 각 클래스 개수

```
In [11]: tr_lb_txt = os.listdir('./labeled_data/labels')
tr_lb_txt.sort()
tr_lb_txt
del tr_lb_txt[0]
tr_lb_txt
tr_lb_txt[:10]
```

```
Out[11]: ['sk_tr_000000.txt',
'sk_tr_000001.txt',
'sk_tr_000002.txt',
'sk_tr_000003.txt',
'sk_tr_000004.txt',
'sk_tr_000005.txt',
'sk_tr_000006.txt',
'sk_tr_000007.txt',
'sk_tr_000008.txt',
'sk_tr_000009.txt']
```

```
In [12]: path = './labeled_data/labels/'
df = pd.read_table(path + tr_lb_txt[0], sep = ' ', header = None, names = ['class', 'x1', 'y1', 'x2', 'y2'])
df
```

	class	x1	y1	x2	y2
<b>0</b>	0	0.583333	0.147222	0.734375	0.592592
<b>1</b>	0	0.367708	0.365741	0.617188	0.675925

```
In [13]: # 전체 txt파일 합친 데이터프레임 생성

for i in tr_lb_txt:
    df_temp = pd.read_table(path + i, sep = ' ', header = None, names = ['class', 'x1', 'y1', 'x2', 'y2'])
    df = pd.concat([df, df_temp])
df
```

	class	x1	y1	x2	y2
<b>0</b>	0	0.583333	0.147222	0.734375	0.592592

	class	x1	y1	x2	y2
1	0	0.367708	0.365741	0.617188	0.675925
0	0	0.583333	0.147222	0.734375	0.592592
1	0	0.367708	0.365741	0.617188	0.675925
0	1	0.082562	0.615723	0.337191	0.934082
...	...	...	...	...	...
9	1	0.430170	0.071777	0.447531	0.088379
10	3	0.487654	0.035156	0.499228	0.051758
11	0	0.589120	0.131348	0.599151	0.166504
12	0	0.530478	0.086914	0.537809	0.105957
0	0	0.425000	0.427778	0.607812	0.822222

21431 rows × 5 columns

### class 유형

- 0 : 사람
- 1 : 차량
- 2 : 택시
- 3 : 트럭
- 4 : 버스
- 5 : 삼륜오토바이/세발자전거
- 6 : 오토바이/자전거

In [14]: df['class'].value\_counts()

Out[14]:

## 4. image와 label 동시에 확인

샘플 추출해서 label 일치 여부와 bounding box 확인

```
In [63]: def draw_rect(image, point1, point2): # left-top, right-bottom # point1,2는 리스트
    draw = ImageDraw.Draw(image)
    for i in range(len(point1)):
        draw.rectangle((point1[i], point2[i]), outline = (0,0,255), width = 3)

    return image

# 0부터 999 사이의 난수 생성
import random
n = random.randrange(0,1000)
print(n)
```

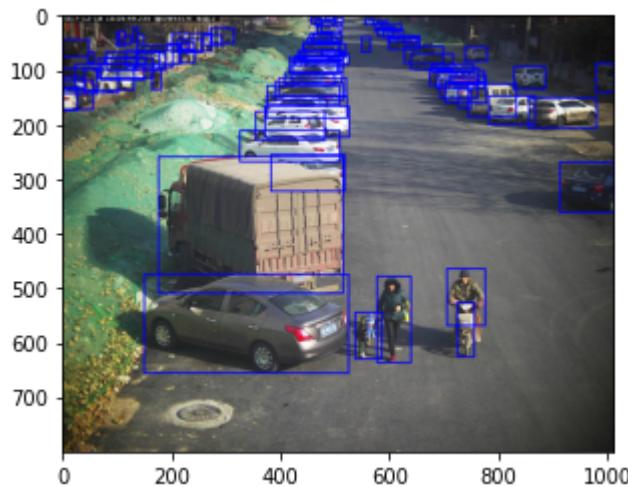
```
# open image (파일명 리스트: tr_lb_img)
path_img = './labeled_data/images/'
image = Image.open(path_img + tr_lb_img[n])
size = image.size

# open txt (파일명 리스트: tr_lb_txt)
path_txt = './labeled_data/labels/'
df_txt = pd.read_table(path_txt + tr_lb_txt[n], sep = ' ', header = None, names = [''])
point1 = []
point2 = []
for i in range(df_txt.shape[0]):
    point1_x = round(df_txt['x1'][i]*size[0], 3)
    point1_y = round(df_txt['y1'][i]*size[1], 3)
    point2_x = round(df_txt['x2'][i]*size[0], 3)
    point2_y = round(df_txt['y2'][i]*size[1], 3)

    point1.append((point1_x, point1_y))
    point2.append((point2_x, point2_y))

image = draw_rect(image, point1, point2)
plt.imshow(np.array(image))
plt.show()
```

599



In [ ]:

In [ ]: