

State Farm Distracted Driver Detection

- 이미지 확인해 본다. 그리고 이미지를 gif로 만들어본다.
- 개발 환경 : Kaggle Notebook
- 문제 유형 : Multi-class Classification(다중 클래스 분류)
- 평가 척도 : Multi-class Logarithmic Loss
- 대회 개요 : CDC 자동차 안전 본부에 따르면, 자동차 사건 5건 중 1건은 산만한 운전자로 인해 발생. 스테이트 팜은 대시보드 카메라를 통해 주의가 산만한 운전자들을 자동으로 감지하여 고객을 보호하고 사건을 예방하고자 함.
 - 학습용, 테스트 데이터 셋 양이 적절함.
 - 이미지 대회 중, 커널 공유 내용에 대해 저작권 이슈가 없다.
- 목표 : 차량내에 설치된 대시보드 카메라 이미지 세트를 기반으로 스테이트 팜은 캐글러들에게 각 운전자들의 행동을 분류해 주기를 바란다.
- 제공 데이터 셋
 - 2만개의 학습 데이터와 8만개의 테스트 데이터를 제공
- 코드 참조 : <https://www.kaggle.com/code/titericz/just-relax-and-watch-some-cool-movies>
- 파일 내용 확인 결과
- /kaggle/input/state-farm-distracted-driver-detection/sample_submission.csv
- /kaggle/input/state-farm-distracted-driver-detection/driver_imgs_list.csv
- ~ /imgs/test/img_00000.jpg
- ~ /imgs/train/c0/img_00000.jpg
- ~ /imgs/train/c1/img_00000.jpg
- ~ /imgs/train/c2/img_00000.jpg
- ~ /imgs/train/c3/img_00000.jpg
- ~ /imgs/train/c4/img_00000.jpg
- ~ /imgs/train/c5/img_00000.jpg
- ~ /imgs/train/c6/img_00000.jpg
- ~ /imgs/train/c7/img_00000.jpg
- ~ /imgs/train/c8/img_00000.jpg
- ~ /imgs/train/c9/img_00000.jpg

```
In [ ]: # 파일 확인
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import os
# for dirname, _, filenames in os.walk('/kaggle/input'):
#     for filename in filenames:
#         print(os.path.join(dirname, filename))
base_dir = "/kaggle/input/state-farm-distracted-driver-detection/"

# 테스트 데이터 셋 개수
test_num = len( os.listdir(base_dir + "/imgs/test/"))
print( "테스트 데이터 셋 크기 :", test_num )

sum = 0
```

```

for i in range(10):
    num = len( os.listdir(base_dir + "/imgs/train/c" + str(i) + "/" ) )
    print( "c" + str(i) + "의 이미지 개수 : ", num )
    sum += num

# 학습용 데이터 셋 개수
print( "학습용 데이터 셋 크기 :", sum)

```

```

테스트 데이터 셋 크기 : 79726
c0의 이미지 개수 : 2489
c1의 이미지 개수 : 2267
c2의 이미지 개수 : 2317
c3의 이미지 개수 : 2346
c4의 이미지 개수 : 2326
c5의 이미지 개수 : 2312
c6의 이미지 개수 : 2325
c7의 이미지 개수 : 2002
c8의 이미지 개수 : 1911
c9의 이미지 개수 : 2129
학습용 데이터 셋 크기 : 22424

```

- 테스트 데이터 셋 : 79,736개의 이미지 * 학습용 데이터 셋 : 22,424개의 이미지

01. 라이브러리 불러오기

```

In [ ]: import numpy as np
import pandas as pd
import cv2

import matplotlib.pyplot as plt
import sys
from skimage import io, transform
import matplotlib.animation as animation

```

02. 데이터 불러오기

```

In [ ]: train = pd.read_csv( '/kaggle/input/state-farm-distracted-driver-detection/dr
sub = pd.read_csv( '/kaggle/input/state-farm-distracted-driver-detection/samp

train.shape, sub.shape

```

```

Out[ ]: ((22424, 3), (79726, 11))

```

```

In [ ]: train.head()

```

```

Out[ ]:
  subject  classname      img
0    p002         c0  img_44733.jpg
1    p002         c0  img_72999.jpg
2    p002         c0  img_25094.jpg
3    p002         c0  img_69092.jpg
4    p002         c0  img_92629.jpg

```

```

In [ ]: sub.head()

```

```

Out[ ]:
  img  c0  c1  c2  c3  c4  c5  c6  c7  c8  c9
0  img_1.jpg  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1

```

	img	c0	c1	c2	c3	c4	c5	c6	c7	c8	c9
1	img_10.jpg	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
2	img_100.jpg	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
3	img_1000.jpg	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
4	img_100000.jpg	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

```
In [ ]: subj = np.unique( train['subject'] )
        subj
```

```
Out[ ]: array(['p002', 'p012', 'p014', 'p015', 'p016', 'p021', 'p022', 'p024',
               'p026', 'p035', 'p039', 'p041', 'p042', 'p045', 'p047', 'p049',
               'p050', 'p051', 'p052', 'p056', 'p061', 'p064', 'p066', 'p072',
               'p075', 'p081'], dtype=object)
```

```
In [ ]: subj[:2]
```

```
Out[ ]: array(['p002', 'p012'], dtype=object)
```

- base : /kaggle/input/state-farm-distracted-driver-detection/
- ~ /imgs/train/c0/img_00000.jpg
- ~ /imgs/train/c1/img_00000.jpg
- ~ /imgs/train/c2/img_00000.jpg
- ~ /imgs/train/c3~c9/img_00000.jpg

03. 주제별 이미지 확인을 위한 전처리 및 gif만들기(1)

```
In [ ]: for subj in np.unique( train['subject'][:2]): # ['p002', 'p012']
        imagem = train[ train['subject']==subj ]
        print(imagem.shape) # 각각의 주제별 이미지의 개수

        imgs = []
        t = imagem.values[0] # 주제별 이미지 하나의 데이터 내용 (['p002' 'c0' 'img_447
        print(t)
        for t in imagem.values: # 각 주제별 데이터 개수만큼 반복.
            base = "/kaggle/input/state-farm-distracted-driver-detection/imgs/tra
            image_path = base + t[1] + "/" + t[2]
            # print(image_path) # 이미지의 경로

            img = cv2.imread( image_path, 3 )
            img = cv2.resize( img, (160, 120))
            img = cv2.cvtColor( img, cv2.COLOR_BGR2RGB )
            imgs.append( img )

        print( len(imgs) )

(725, 3)
['p002' 'c0' 'img_44733.jpg']
725
(823, 3)
['p012' 'c0' 'img_10206.jpg']
823
```

03. 주제별 이미지 확인을 위한 전처리 및 gif만들기(2)

```
In [ ]: fig = plt.figure()
        subj = np.unique( train['subject'][:2]
        print(subj)
```

```

for subj in np.unique( train['subject'] )[:2]: # ['p002', 'p012']
    imagem = train[ train['subject']==subj ]
    print(imagem.shape) # 각각의 주제별 이미지의 개수

    imgs = []
    t = imagem.values[0] # 주제별 이미지 하나의 데이터 내용 (['p002' 'c0' 'img_447
    print(t)
    for t in imagem.values: # 각 주제별 데이터 개수만큼 반복.
        base = "/kaggle/input/state-farm-distracted-driver-detection/imgs/tra
        image_path = base + t[1] + "/" + t[2]
        # print(image_path) # 이미지의 경로

        # 각 이미지별 전처리
        img = cv2.imread( image_path, 3 )
        img = cv2.resize( img, (160, 120))
        img = cv2.cvtColor( img, cv2.COLOR_BGR2RGB )
        imgs.append( img )

    # print( len(imgs) )
    ax = fig.add_subplot(111)
    ax.set_axis_off()

    fig.subplots_adjust(left=0, bottom=0, right=1, top=1, wspace=None, hspace=
    fname = "MOVIE_subject_" + subj + ".gif"
    imgs = [ (ax.imshow(img),
               ax.set_title(t[0]),
               ax.annotate(n_img,(5,5))) for n_img, img in enumerate(imgs) ]

    img_anim = animation.ArtistAnimation(fig, imgs, interval=125,
                                         repeat_delay=1000, blit=False)

    print("Writing : ", fname)
    img_anim.save(fname, writer='imagemagick', dpi=20)
    fig.clf()

print ( 'Now relax and watch some movies!!!' )

['p002' 'p012']
(725, 3)
['p002' 'c0' 'img_44733.jpg']
Writing : MOVIE_subject_p002.gif
(823, 3)
['p012' 'c0' 'img_10206.jpg']
Writing : MOVIE_subject_p012.gif
Now relax and watch some movies!!!
<Figure size 432x288 with 0 Axes>

```

In []: