In [94]:

```
import torch
import os
from os import path
import numpy as np
import shutil
import random
from sklearn.utils import shuffle
import pickle
```

In [11]:

```
if torch.cuda.is_available():
    device = torch.device("cuda")
    print("There are %d GPU(s) available." % torch.cuda.device_count())
    print("We will use the GPU:", torch.cuda.get_device_name(0))
else:
    print("No GPU available, using the CPU instead.")
    device = torch.device("cpu")
```

```
There are 1 GPU(s) available.
We will use the GPU: Tesla T4
```

In [14]:

```
os.chdir('/home/work/sample-notebooks/train')
```

데이터 로드

In [20]:

```
tr_lb_img = os.listdir('./labeled_data/images')
tr_lb_img.sort()
del tr_lb_img[0]
tr_lb_img[:5]
```

Out[20]:

```
['sk_tr_000000.jpg',
 'sk_tr_000001.jpg',
 'sk_tr_000002.jpg',
 'sk_tr_000003.jpg',
 'sk_tr_000004.jpg']
```

In [21]:

```
tr_lb_txt = os.listdir('./labeled_data/labels')
tr_lb_txt.sort()
del tr_lb_txt[0]
tr_lb_txt[:5]
```

Out[21]:

```
['sk_tr_000000.txt',
 'sk_tr_000001.txt',
 'sk_tr_000002.txt',
 'sk_tr_000003.txt',
 'sk_tr_000004.txt']
```

In [22]:

```
print(len(tr_lb_img))
print(len(tr_lb_txt))
```

```
1000
1000
```

In [37]:

```
# 원본 images 폴더 전체 복사 (shutil.copy)
img_org = './labeled_data/images/'    # 원본 폴더
img_dst = './labeled_data/images_org/'    # 빈 폴더
img_files = os.listdir(img_org)
img_files.sort()
```

```
del img_files[0]
for file in img_files:
    if not path.exists(img_dst + file):
        shutil.copy(img_org + file, img_dst + file)
```

In [38]:

```
# 원본 images, labels 폴더 전체 복사 (shutil.copy)
lbs_org = './labeled_data/labels/'    # 원본 폴더
lbs_dst = './labeled_data/labels_org/'    # 빈 폴더
lbs_files = os.listdir(lbs_org)
lbs_files.sort()
del lbs_files[0]
for file in lbs_files:
    shutil.copy(lbs_org + file, lbs_dst + file)
```

In [23]:

```
!mkdir ./labeled_data/test_images
!mkdir ./labeled_data/test_annotations
```

In [29]:

```
random.seed(23)
idx = random.sample(range(1000),100)
```

In [58]:

```
# images에서 test용 파일 100개 이동 (shutil.move)
test_img_path = './labeled_data/test_images/'
for file in np.array(img_files)[idx]:
    shutil.move(img_org + file, test_img_path + file)
```

In [59]:

```
# labels에서 test용 파일 100개 이동 (shutil.move)
test_lbs_path = './labeled_data/test_annotations/'
for file in np.array(lbs_files)[idx]:
    shutil.move(lbs_org + file, test_lbs_path + file)
```

In [ ]:

```
# images 폴더 -> Rename to 'train_images'
# labels 폴더 -> Rename to 'train_annotations'
```

In [86]:

```
# train 데이터 개수 확인
train_img_path = './labeled_data/train_images'
train_img_list = os.listdir(train_img_path)
train_img_list.sort()
del train_img_list[0]
print(len(train_img_list))

train_lbs_path = './labeled_data/train_annotations'
train_lbs_list = os.listdir(train_lbs_path)
train_lbs_list.sort()
del train_lbs_list[0]
print(len(train_lbs_list))
```

```
900
900
```

In [87]:

```
# test 데이터 개수 확인
test_img_path = './labeled_data/test_images'
test_img_list = os.listdir(test_img_path)
test_img_list.sort()
print(len(test_img_list))
```

```
test_lbs_path = './labeled_data/test_annotations'
test_lbs_list = os.listdir(test_lbs_path)
test_lbs_list.sort()
print(len(test_lbs_list))
```

```
100
100
```

In [88]:

```
## 데이터 shuffle - img와 lbs 같은 random_state로 지정
# train 데이터 shuffle
train_img_list = shuffle(train_img_list, random_state = 23)
train_lbs_list = shuffle(train_lbs_list, random_state = 23)

# test 데이터 shuffle
test_img_list = shuffle(test_img_list, random_state = 23)
test_lbs_list = shuffle(test_lbs_list, random_state = 23)
```

In [101…

```
## 데이터 파일명 리스트 -> pkl 파일로 저장
# train 데이터
with open('./labeled_data/train_img.pkl', 'wb') as file:
    pickle.dump(train_img_list, file, protocol = pickle.HIGHEST_PROTOCOL)
with open('./labeled_data/train_lbs.pkl', 'wb') as file:
    pickle.dump(train_lbs_list, file, protocol = pickle.HIGHEST_PROTOCOL)

# test 데이터
with open('./labeled_data/test_img.pkl', 'wb') as file:
    pickle.dump(test_img_list, file, protocol = pickle.HIGHEST_PROTOCOL)
with open('./labeled_data/test_lbs.pkl', 'wb') as file:
    pickle.dump(test_lbs_list, file, protocol = pickle.HIGHEST_PROTOCOL)
```

In [115…

```
# pkl 파일 로드하여 확인
with open('./labeled_data/train_img.pkl', 'rb') as file:
    tr_img = pickle.load(file)
print(len(tr_img))
tr_img[:10]
```

```
900
```

Out[115…

```
['sk_tr_000863.jpg',
 'sk_tr_000822.jpg',
 'sk_tr_000841.jpg',
 'sk_tr_000990.jpg',
 'sk_tr_000477.jpg',
 'sk_tr_000654.jpg',
 'sk_tr_000480.jpg',
 'sk_tr_000470.jpg',
 'sk_tr_000173.jpg',
 'sk_tr_000732.jpg']
```