The notebook is used as a "first look" of the Google competition "Landmark Recognition 2019" and an

Recap on the available train and the 2020 competition is meant to be general-purpose and we will use 2019 competition train set as a proxy for the 2020 competition.

Kaggle's 2019 competition train set was released as 224 tar.gzip files (contains all the character strings randomly typed separated by UT Kaggle hashtag) for a single 224 tar.gzip file (contains 224 files released in alphabetical order) and Kaggle "Kernel" offering is which contain the image tarballs.

It can be found here

```
!ls -lh /kaggle/input/
```

```
total 4.0K
drwxrwxr-x 3 root root 4096 Jun  3 19:37 landmark-recognition-2019
drwxrwxr-x 3 root root 4096 Jun  3 19:37 landmarks-dataset-224-tar-gzip
```

```
# List the train data in /kaggle/input/landmark-recognition-2019/train
!ls -lh /kaggle/input/landmark-recognition-2019/train
```

```
total 4.0K
drwxrwxr-x 4 kaggle kaggle
drwxrwxr-x 3 kaggle kaggle
drwxrwxr-x 2 kaggle kaggle 4096 Jun  3 19:37 0
drwxrwxr-x 2 kaggle kaggle 4096 Jun  3 19:37 1
drwxrwxr-x 2 kaggle kaggle 4096 Jun  3 19:37 2
drwxrwxr-x 2 kaggle kaggle 4096 Jun  3 19:37 3
drwxrwxr-x 2 kaggle kaggle 4096 Jun  3 19:37 4
drwxrwxr-x 2 kaggle kaggle 4096 Jun  3 19:37 5
drwxrwxr-x 2 kaggle kaggle 4096 Jun  3 19:37 6
drwxrwxr-x 2 kaggle kaggle 4096 Jun  3 19:37 7
drwxrwxr-x 2 kaggle kaggle 4096 Jun  3 19:37 8
drwxrwxr-x 2 kaggle kaggle 4096 Jun  3 19:37 9
```

```
# List the train data in /kaggle/input/landmarks-dataset-224-tar-gzip
!ls -lh /kaggle/input/landmarks-dataset-224-tar-gzip
```

```
total 4.0K
drwxrwxr-x 2 root root 4096 Jun  3 19:37 0
drwxrwxr-x 2 root root 4096 Jun  3 19:37 1
drwxrwxr-x 2 root root 4096 Jun  3 19:37 2
drwxrwxr-x 2 root root 4096 Jun  3 19:37 3
drwxrwxr-x 2 root root 4096 Jun  3 19:37 4
drwxrwxr-x 2 root root 4096 Jun  3 19:37 5
drwxrwxr-x 2 root root 4096 Jun  3 19:37 6
drwxrwxr-x 2 root root 4096 Jun  3 19:37 7
drwxrwxr-x 2 root root 4096 Jun  3 19:37 8
drwxrwxr-x 2 root root 4096 Jun  3 19:37 9
```

```
# Count the number of tar.gz files in the train data
!ls -lh /kaggle/input/landmarks-dataset-224-tar-gzip/*/*.tar.gz | wc -l
```

```
224
```

As we can see there are a lot of 224 tarball in the landmarks column. That mean that a lot of the objects in the train set are not unique.

We would like to look at the image from the train set.

```
# Import the required libraries
import os, sys, glob, tarfile, shutil, tqdm, numpy as np, pandas as pd, matplotlib as mpl, matplotlib.pyplot as plt
%matplotlib inline
from tqdm.notebook import tqdm
from PIL import Image, ImageDraw, ImageFont
from IPython.display import display, HTML
from sklearn.preview.image_extractor import ImageCollectionExtractor
from sklearn.preview.image_extractor.io.tarfile import TarfileImageCollection
from sklearn.preview.image_extractor.io.base import ImageCollectionWithIndex
from sklearn.preview.image_extractor.io.base_tarfile import TarfileImageCollectionWithIndex
from sklearn.preview.image_extractor.io.base_tarfile_lazy import LazyTarfileImageCollectionWithIndex
from sklearn.preview.image_extractor.io.base_tarfile_lazy_cache import LazyTarfileImageCollectionWithIndexAndCache
from sklearn.preview.image_extractor import ExtractorConfig, ImageExtractor
%matplotlib inline
%load_ext autoreload
%autoreload 2
%load_ext pretty_print_ast
%load_ext pretty_print_ast.ipy_ipython_ext
%load_ext pretty_print_ast.ipy_ipython_ext.pretty_print_ast_ipython_ext
```

```
# Define the path to the train data
train_dir = '/kaggle/input/landmark-recognition-2019/train'
# Define the path to the landmarks dataset
landmarks_dir = '/kaggle/input/landmarks-dataset-224-tar-gzip'
# Define the path to the output directory
output_dir = '/kaggle/working'
# Define the path to the output directory for the landmarks dataset
landmarks_output_dir = '/kaggle/working/landmarks
```

Train set Images - Using MLRSD

```
train_1 = TRUE
```

```
train_2 = TRUE
train_set = list.concatenate(c_to_fnd_path(train_dir/train_ex_fnd_path(train_dir/image) for ex in train_1*[image] + tra
train_ds = tf.ds(train_set, shuffle=True, batch=32, cache=True, ).map(lambda x: (x[0], x[1]))
img_train = tf.ds(train_set).map(lambda x: x[0]).map(lambda x: x[0]).take(9).stack().numpy()
show_images(img_train, ncols=3)
resize_square(img_train[0], size=224)
train_ds.cardinality().numpy()
```

```
img_train, _ = next(iter(train_ds))
show_images(img_train, ncols=3,figsize=(6,6))
train_ds.cardinality().numpy()
train_ds.shape(224,224,3)
train_ds = train_ds.map(lambda x, y: (resize(x, (224,224)), y), num_parallel=8)
train_ds = train_ds.map(lambda x, y: (x/255.0, y), num_parallel=8)
```

train set image - train_70/80

```
train_1 = TRUE
train_2 = TRUE
train_set = list.concatenate(c_to_fnd_path(train_dir/train_ex_fnd_path(train_dir/image) for ex in train_1*[image] + tra
train_ds = tf.ds(train_set, shuffle=True, batch=32, cache=True, ).map(lambda x: (x[0], x[1]))
img_train = tf.ds(train_set).map(lambda x: x[0]).map(lambda x: x[0]).take(9).stack().numpy()
show_images(img_train, ncols=3)
resize_square(img_train[0], size=224)
train_ds.cardinality().numpy()
```

```
img_train, _ = next(iter(train_ds))
show_images(img_train, ncols=3,figsize=(6,6))
train_ds.cardinality().numpy()
train_ds.shape(224,224,3)
train_ds = train_ds.map(lambda x, y: (resize(x, (224,224)), y), num_parallel=8)
train_ds = train_ds.map(lambda x, y: (x/255.0, y), num_parallel=8)
```


train set image - train_70/80

train set image - train_70/80

As we can see from the images above, sometimes the train and the rear set images are not similar. They can differ in color, angle, and distance.

In class-1RNN for example, the model's is trained on the rear-view images. The train set images include images of the middle from the inside and from the outside while the test images include only images from the inside.

In class-3MLRS, the model's is trained on a balanced policy. The train set images are images taken from all around the greater which is quite different than the test set images taken from the middle of the train.

In class-7MLRS, the model's is trained on balanced policy. The train images include images from different part of the train while the test image is a close up of the rear.

Three differences between the train set and the test set will make our recognition mission very difficult and even more

This is the ERP-test set that will use to evaluate our algorithm.