LANDMARK RECOGNITION

COURSE: BIG DATA CONTENT ANALYTICS

ELEFTHERIA APOSTOLAKI (P2821803) – MATILDA TSAKA (P2821826)
MASTER'S IN BUSINESS ANALYTICS

I. Problem Description & mission

The purpose of this assignment is to use landmark recognition technology to predict landmark labels directly from image pixels.

An earth-scale landmark recognition engine is tremendously useful for many vision and multimedia applications. First, by capturing the visual characteristics of landmarks, the engine can provide clean landmark images for building virtual tourism of a large number of landmarks. Second, by recognizing landmarks, the engine can facilitate both content understanding and geo-location detection of images and videos. Third, by geographically organizing landmarks, the engine can facilitate an intuitive geographic exploration and navigation of landmarks in a local area, so as to provide tour guide recommendation and visualization. To build such an engine, the following issues, however, must be tackled:

(a) there is no readily available list of landmarks in the world;

(b) even if there was such a list, it is still challenging to collect true landmark images;

(c) efficiency is a non-trivial challenge for such a large-scale system

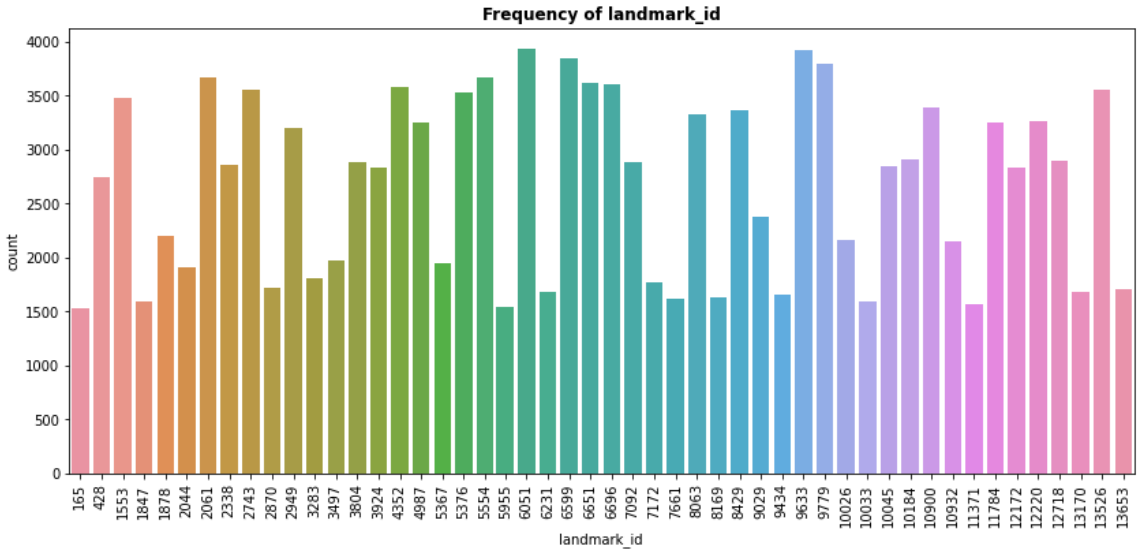
II. Data

The data used for this assignment were found in GitHub (https://www.kaggle.com/c/landmark-recognition-2019/data). These data initially included almost 1.200.000 images which were processed as follows:

- The images not related with any web URL were removed.
- The top 50 landmarks were chosen leading to the selection of 133.114 images in total out of which 90.531, 22.632 and 19.951 images consisted the training, the validation and the test data set respectively. These images were converted to RGB images and also resized to 150x150 pixels in parallel while they were being downloaded (using terminal commands).
- Normalization technique was used as ImageDataGenerator class rescaled pixel values from the range of 0-255 to the range 0-1 preferred

for neural network models. With this image augmentation technique, the variations of images are also artificially increased in our dataset by using zoom, horizontal flips, horizontal/vertical shifts and rotations.

- The input data has a shape of (64, 150, 150, 3) as batch size, height, width and depth respectively. The depth of the image represents the number of color channel which in our case is 3 for RGB images.
- As the below plot indicates, the landmark ID for which the max number of images exists is 9633 which in fact depicts the Papal Basilica of St. Peter in the Vatican City.



Below you may see some indicative images for the specific landmark.

IV. METHODOLOGY

- After further investigation, we concluded that image classification problems and image processing in general can be most suitably solved with Convolutional Neural Networks as RNNs are commonly used when working with texts and sequences and MPLs are redundant in high dimensions and thus insufficient for modern advanced computer vision tasks. In CNNs the weights are smaller and shared — less wasteful, easier to train than MLP, more effective and can also go deeper.

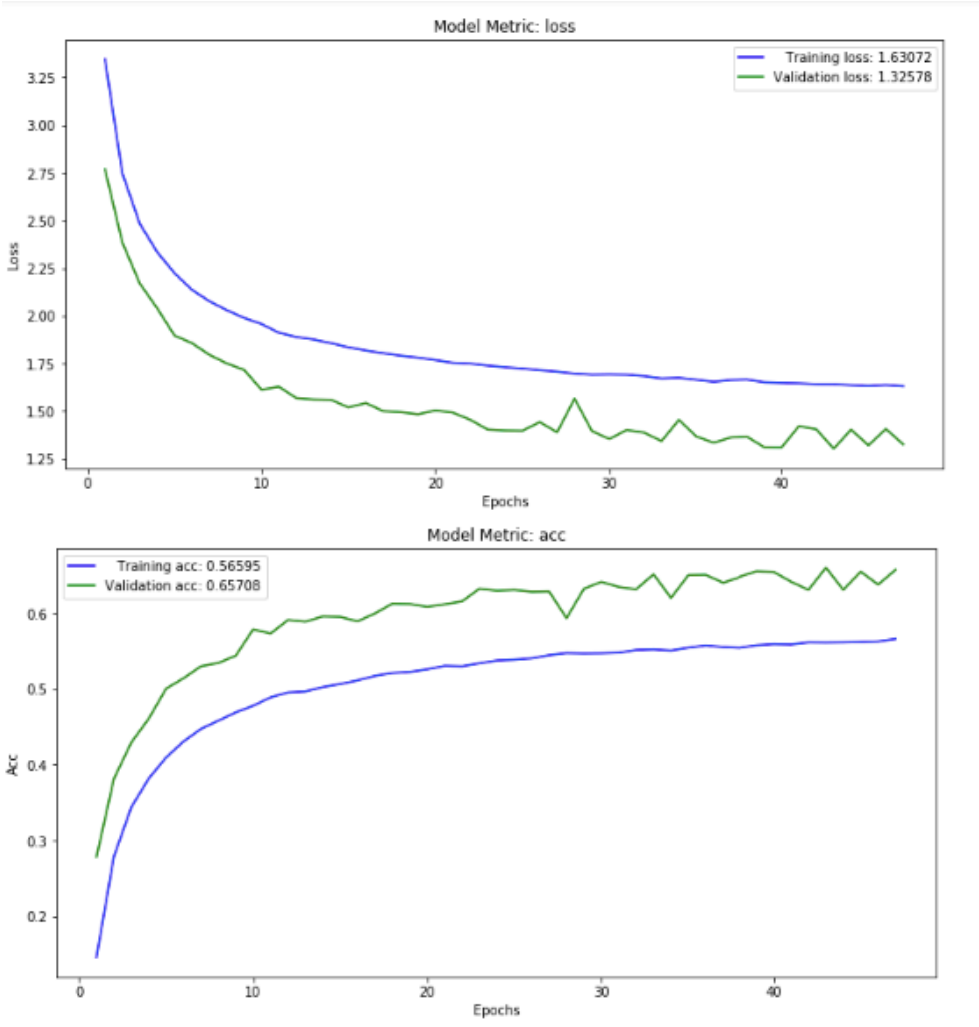
- The created CNN model consists of the following parts:
    - Input layer (Feature selection technique used to extract the images)
    - Hidden layers: (a) Convolutional layer (Relu technique), (b) Pooling layer (Max Pooling technique)
    - Output layer (Softmax classification technique)
- The parameters of this model are defined as follows:
    - Batch size = 64
    - Number of epochs = 50
    - Kernel size = 5
    - Pool size = 2
    - Strides size = 2
    - Convolutional depth = 32
    - Convolutional depth 2 = 64
    - Dropout probability = 0.5
    - Dropout probability 2 = 0.5
    - Hidden size = 512
- As far as the hyper-parameters are concerned, the adaptive learning rate method of Adam optimizer is used as it learns the fastest and it is more stable than the other optimizers since it doesn't suffer any major decreases in accuracy.
- The evaluation measures used were confusion matrix and classification report to calculate the ratio of correct predictions to total predictions made and to create a summary of prediction results on our multi-classification problem.

V. RESULTS

- The Google Colab is used to develop the CNN model on the GPU. The training process lasted about 9 hours.
- The major problem encountered during this assignment was related with the small sample initially used to train the model because of which

inadequate results were produced. Due to this problem, a significant time effort was invested to re-download a greater sample of images from the available ones.

- As it can be seen, in the majority of epochs, validation loss goes lower and validation accuracy goes higher as we move on from one to another epoch. This means that the model is in general well-trained. However, for some cases, e.g. in epoch 21, validation loss is increased, and validation accuracy is decreased meaning that model is cramming values and not learning.



- Moreover, from the below generated classification report and confusion matrix we can see that high precision is not achieved for any of the 50 different landmarks. Thus, we consider that other approaches would fit better for which more investigation would be needed. This could be accomplished if we had more time and human resources.

Classification Report

	precision	recall	f1-score	support
0	0.02	0.02	0.02	323
1	0.02	0.01	0.02	237
2	0.03	0.04	0.04	424
3	0.03	0.03	0.03	435
4	0.02	0.01	0.02	498
5	0.02	0.02	0.02	322
6	0.01	0.00	0.00	233
7	0.02	0.02	0.02	481
8	0.02	0.01	0.02	416
9	0.02	0.02	0.02	483
10	0.01	0.01	0.01	425
11	0.02	0.03	0.02	252
12	0.03	0.03	0.03	530
13	0.00	0.00	0.00	255
14	0.02	0.03	0.02	516
15	0.01	0.01	0.01	229
16	0.01	0.00	0.00	237
17	0.02	0.02	0.02	327
18	0.02	0.01	0.02	285
19	0.02	0.03	0.02	548
20	0.02	0.02	0.02	423
21	0.03	0.03	0.03	531
22	0.02	0.02	0.02	255
23	0.03	0.04	0.03	470
24	0.01	0.01	0.01	269
25	0.03	0.02	0.03	295
26	0.02	0.02	0.02	432
27	0.03	0.03	0.03	423
28	0.03	0.02	0.03	408
29	0.03	0.02	0.03	533
30	0.02	0.04	0.03	488
31	0.01	0.01	0.01	290
32	0.03	0.03	0.03	520
33	0.03	0.03	0.03	540
34	0.02	0.02	0.02	227
35	0.02	0.01	0.02	585
36	0.01	0.01	0.01	249
37	0.03	0.03	0.03	569
38	0.03	0.03	0.03	534
39	0.03	0.03	0.03	537
40	0.02	0.02	0.02	427
41	0.02	0.02	0.02	264
42	0.00	0.00	0.00	241
43	0.03	0.04	0.03	495
44	0.02	0.02	0.02	244
45	0.03	0.02	0.02	500
46	0.02	0.02	0.02	355
47	0.03	0.04	0.04	240
48	0.03	0.02	0.02	585
49	0.02	0.01	0.02	566
accuracy			0.02	19951
macro avg	0.02	0.02	0.02	19951
weighted avg	0.02	0.02	0.02	19951

Confusion Matrix

```
[[ 7  2  8 ...  4  5  5]
 [ 5  3  8 ...  3  2  8]
 [ 8  4 16 ...  3  5  6]
 ...
 [ 2  1  5 ...  9  7  5]
 [13  4 17 ...  7 11 11]
 [ 9  4 14 ...  1  9  8]]
```

VI. Roles

As our team consists only of two members with business background, the roles could not be well-defined and thus both of us run this project as data scientists, data engineers, business analysts and machine learning engineers.

VII. Bibliography

[1] https://www.kaggle.com/c/landmark-recognition-2019/overview
[2] http://openaccess.thecvf.com/content_ICCV_2017/papers/Noh_Large-Scale_Image_Retrieval_ICCV_2017_paper.pdf
[3] http://pages.cs.wisc.edu/~dyer/cs534-spring10/papers/google_landmark_recognition.pdf
[4] https://towardsdatascience.com/wtf-is-image-classification-8e78a8235acb
[5] http://cs231n.github.io/convolutional-networks/
[6] https://www.datacamp.com/community/tutorials/convolutional-neural-networks-python?utm_source=adwords_ppc&utm_campaignid=898687156&utm_adgroupid=48947256715&utm_device=c&utm_keyword=&utm_matchtype=b&utm_network=g&utm_adpostion=1t1&utm_creative=332602034352&utm_targetid=aud-299261629574:dsa-473406581915&utm_loc_interest_ms=&utm_loc_physical_ms=9061576&gclid=Cj0KCQjwqs3rBRCdARIsADe1pfTa07Puw94IV1H3_BwB0gWfnOVsLxrUKgNFRPZwNw3KHfcM3CAUuGwaAkHbEALw_wcB

VIII. Time Plan

The following time plan was scheduled and carried out in general by the team.

Tasks	5/8 - 11/8	12/8 - 18/8	19/8 - 25/8	26/8 - 1/9	2/9 - 8/9
Bibliography research	✓				
Dataset preprocessing	✓				
Images download		✓	✓		
Images preprocessing			✓		
Build the CNN model			✓	✓	
Analysis of the results				✓	✓
Report					✓

VIX. COMMENTS

As mentioned above, both the members of our team do not have a technical background, and this is why a significant effort was needed to investigate and write the code required for both handling the dataset and building the model. On top of that, we needed approximately 2 weeks to download the images as our first trials to download several subsets led to insufficient results.