Machine Learning Engineer Nanodegree

Capstone Project Proposal
Google Landmark Recognition Challenge

Amit Kapoor
August 14, 2018

Domain Background

Image classification has been explored extensively over the years and it has shown significant improvement over the past few years. ImageNet classification challenge has played a significant role for this improvement where error rates have been impressive substantially every year. Landmark recognition is being in focus these days to advance the state of the art in computer vision. It can predict landmark labels directly from image pixels, to help better understand and organize photos.

[2] Crandall et al's paper on landmark classification described the improvement achieved using extra features like keywords however deep learning models looks more promising as compared to classification using K-means. Deep learning techniques appears can handle large set of data like provided in this Kaggle challenge. Similarly, [4] Xception model showed great results on a larger image classification dataset comprising 350 million images and 17,000 classes.

The biggest hurdle in landmark recognition research is being the lack of large annotated datasets. In this competition, Google presents the largest worldwide dataset to make progress and build models that recognize the correct landmark.

Problem Statement

Problem statement is "to build model that recognize the correct landmark".

The Kaggle competition (https://www.kaggle.com/c/landmark-recognition-challenge) is to classify about approx. 15000 landmark categories from images. Landmark recognition has been challenging in its own way. Landmark recognition can predict landmark labels directly from image pixels, to help better understand and organize photo collections.

The main challenges included in this competition are

1. Larger number of categories (15k approx.)
2. Number of training examples per class may not be very large.
3. Recognize the correct landmark (if any) in a dataset of challenging test images.
4. Calculate Global Average Precision (GAP)

Datasets and Inputs

The dataset for this challenge is available in following links:

https://www.kaggle.com/c/landmark-recognition-challenge/data
https://www.kaggle.com/google/google-landmarks-dataset

The dataset contains 2 csv files with required information to make predictions.

train.csv - This file contains a large number of images labeled with their associated landmarks. These images will be used for training the models. The training set images depict exactly one landmark. The 3 columns in this file, contains the train image ID, its URL and its label respectively. Number of rows and columns in this file are **1225030** and 3 respectively.

id	→	image ID (a hash)
url	→	image URL

landmark_id → image label (an integer)

test.csv - the test set containing the test images to predict landmarks. Number of rows and columns in this file are **117704** and 2 respectively

id → image ID (a hash)
url → image URL

All images will be resized after download per the model being used for training. Size considered here is 256x256 which seems good enough to represent the images. These csv files contain the Image URLs only. All the images will be downloaded through Python script. The actual image data size is very huge so the link is being provided here. Since the dataset is huge and it requires lot of computation power, I will first start with relatively smaller dataset and test the model performance. It will then gradually increase it to larger dataset.

Solution Statement

I will use KNN (K nearest neighbors) model for classification. The result from KNN will be used as baseline to compare against. Next, I am planning to use Transfer learning using pre-trained models. To achieve this, I will utilize Keras Deep Convolutional Neural Network (CNN) models to classify the images. In this case, CNN models that will be used are VGG16, Xception and Residual Network (ResNet) model. Hyper parameters optimization will be performed as needed.

Benchmark Model

Benchmark model considered in this case is KNN (K nearest neighbors) approach. K nearest neighbors requires less computational overhead as compared to CNN (Convolution Neural Networks) and hence being considered here for baseline.

Along with that since this is a Kaggle challenge, the end goal will also be to have my model achieve comparative score against the ones listed in leader board of this challenge.

Evaluation Metrics

Since this is a Kaggle Competition, we already have an evaluation metric defined as Global Average Precision (GAP). This metric is also known as micro Average Precision (microAP), as per [1]. It works as follows:

For each query image, prediction will be one landmark label and a corresponding confidence score. The evaluation treats each prediction as an individual data point in a long list of predictions (sorted in descending order by confidence scores), and computes the Average Precision based on this list.

If a submission has N predictions (label/confidence pairs) sorted in descending order by their confidence scores, then the Global Average Precision will be computed as:

N

$$GAP = 1/M \sum_{i=1} P(i) rel(i)$$

where:

→ N is the total number of predictions returned by the system, across all queries
→ M is the total number of queries with at least one landmark from the training set visible in it (note that some queries may not depict landmarks)
→ P(i) is the precision at rank i
→ rel(i) denotes the relevance of prediction i: it's 1 if the i-th prediction is correct, and 0 otherwise

Project Design

I will utilize AWS/Google Colab to setup Deep Learning environment as this challenge requires lots of computation power. Python script to download the images has been provided by Kaggle. Sklearn library will be utilized to split the data into training, validation and testing sets. I will use KNN (K nearest neighbors) model and Deep Convolutional Neural Network (CNN) models VGG16, Xception and Residual Network (ResNet). Keras library will be used for implementing Deep CNN models. Hyper parameters optimization will be performed as needed. Hyper parameter tuning will include checking various optimizers like ADAM and RMSprop.

Tools and Libraries used: *Jupyter Notebook, Python, scikit learn, pandas, matplotlib, seaborn, tensorflow, keras. If needed, other libraries would be added.*

Reference:

[1] F. Perronnin, Y. Liu, and J.-M. Renders, "A Family of Contextual Measures of Similarity between Distributions with Application to Image Retrieval," Proc. CVPR'09

[2] David J. Crandall, Yunpeng Li, Stefan Lee, and Daniel P. Huttenlocher "Recognizing Landmarks in Large-Scale Image Collections". IEEE, 2009.

[3] Andre Araujo and Tobias Weyand "Google-Landmarks: A New Dataset and Challenge for Landmark Recognition" Google Research, Thursday, March 1, 2018.
https://ai.googleblog.com/2018/03/google-landmarks-new-dataset-and.html

[4] François Chollet. "Xception: Deep Learning with Depthwise Separable Convolutions" asXiv 2016.

https://www.kaggle.com/c/landmark-recognition-challenge
https://www.kaggle.com/google/google-landmarks-dataset
https://aws.amazon.com/blogs/machine-learning/get-started-with-deep-learning-using-the-aws-deep-learning-ami/
https://colab.research.google.com/notebooks/welcome.ipynb