UDACITY

Machine Learning Engineer Nanodegree
Capstone Proposal
Landmark Recognition

Fady Morris Milad Ebeid

December 3, 2019

Contents

1 Domain Background

The research in computer vision dates back to 60 years ago. Computer scientist have been working on researching new ways to make computers extract meaningful information from given input images.

Computer vision algorithms include methods for acquiring, processing, analyzing and understanding digital images, and extraction of data from the real world. It is an interdisciplinary field that deals with how can computers gain a high-level understanding of digital images. It aims to mimic human vision.

Since 2010 , there was an acceleration in the development of deep learning techniques and technologies. With deep learning, we're now able to use high performance computers and GPUs to train deep learning models that improve over time and provide these models to businesses.

In this problem, I have chosen to use convolutional neural networks which outperform traditional computer vision algorithms. I will train a convolutional neural network to solve the Google Landmark Recognition 2019 Problem.

Convolutional networks are able to successfully capture the Spatial and Temporal dependencies in an image by applying filters. They performs a better fitting to the image dataset due to the decrease in the number of parameters used and reusability of weights.

Some academic research relevant to this problem domain can be found in : [Zhe+09] and [CCF16]

2 Problem Statement

Did you ever go through your vacation photos and ask yourself: What is the name of this temple I visited in China? Who created this monument I saw in France? Landmark recognition can help! This technology can predict landmark labels directly from image pixels, to help people better understand and organize their photo collections.

This problem was inspired by Google Landmark Recognition 2019 Challenge on Kaggle.

Landmark recognition is a little different from other classification problems. It contains a much larger number of classes (there are a total of 15K classes in this challenge), and the number of training examples per class may not be very large. Landmark recognition is challenging in its own way.

This problem is a multi-class classification problem. We will build a classifier that can be trained using the given dataset and predict the landmark class from a given input image.

3 Datasets and Inputs

The dataset description is in Kaggle Google Landmark Recognition and can be downloaded from Common Visual Data Foundation[1] Google Landmarks Dataset v2. This dataset was used in [Noh+17].

It currently has 15K classes with 4,132,914 images. And it has some problems :

- It is a .csv file that has links to download images from.
- The dataset is very large $\approx$ 85 Gigabytes
- it has a large number of classes $\approx$ 15K
- Some classes have a low count of training examples.
- Images in the links have very large resolutions

I will use only a small subset of this dataset. I will select 10 landmarks(classes) which have a large count of training example (> 1000) and download approximately 1000 images for each class, so I will have $\approx$ 10,000 images. Then I will divide this subset into 70% training, 15% validation, 15% test datasets.

4 Solution Statement

I will use the transfer learning technique to train a simple sequential model from Keras (Python deep learning library). I will use pre-trained weights from a pre-trained model from Keras Applications, such as VGG16, VGG19 or Xception.

5 Benchmark Model

My benchmark model will be a simple convolutional neural network that I will create. I will measure the accuracy of this benchmark model and compare it to the solution model that will use transfer learning. I will also compare the performance of the two models, such as training time.

The convolutional neural network will be modeled using Keras library and will contain convolutional layers, max-pooling layers, fully connected layers for output and dropout layers to prevent overfitting.

The Pooling layer is responsible for reducing the spatial size of the Convolved Feature. The fully connected layer at the end of the network converts 2D feature maps into a 1D feature vector whose length corresponds to the number of classes.

[1] The Common Visual Data Foundation is a 501(c)(3) non-profit organization with a mission to enable open community-driven research in computer vision.

6 Evaluation Metrics

The evaluation metric for this problem is the Accuracy Score. Accuracy is the fraction of predictions our model got right.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

$$\texttt{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i)$$

7 Project Design

7.1 Data Preprocessing

I will take a subset of Google Landmarks Dataset v2. First I will download the train.csv that contains links to 4,132,914 image samples and ids of the labels. Then I will select 10 labels that have $\approx$ 1000 sample images or more.

I will use the train_label_to_category.csv to resolve landmark names from landmark ids.

7.2 Subset Data Splitting

I will split the selected subset that is approximately 10,000 samples into 70% training, 15% validation, 15% test datasets.

7.3 Data Download

Then using the three subsets train, validation, test I will download the images from the web using landmark-recognition-challenge-image-downloader.py script provided by Kaggle. I will modify this script to save a scaled verison of downloaded images that have a resolution approximate to 640x480 to save space and reduce computations needed to process the images.

7.4 Model Training and Evaluation

I will import downloaded images into my project. I will do image scaling to the size of 244x244 and data augmentation (crop, shear, and rotation) to use as an input to my basic CNN network, which I will use as a Benchmark Model. Then, I will train my benchmark model and choose the best weights every epoch using the validation set. Then, I will save the best model and evaluate it on the test data and record the test accuracy score (See Evaluation Metrics).

Next, I will use transfer learning techniques to develop my solution model (See Solution Statement). I will obtain bottleneck features from pre-trained Keras model such as VGG16, VGG19 or Xception and build a simple model and evaluate it on the testing set. Then I will record the accuracy score.

I will choose the best pre-trained model as a solution model and compare it to my benchmark model.

I will plot some visualizations of the frequency of landmark sample images, learning curves for training and validation accuracies and transformations that will be done on the training data to augment them.

References

[CCF16] Jiuwen Cao, Tao Chen, and Jiayuan Fan. "Landmark recognition with compact BoW histogram and ensemble ELM". In: *Multimedia Tools and Applications* 75.5 (Mar. 2016), pp. 2839–2857. ISSN: 1573-7721. DOI: 10.1007/s11042-014-2424-1. URL: https://doi.org/10.1007/s11042-014-2424-1.

[Noh+17] Hyeonwoo Noh et al. "Large-Scale Image Retrieval with Attentive Deep Local Features". In: Oct. 2017, pp. 3476–3485. DOI: 10.1109/ICCV.2017.374.

[Zhe+09] Yantao Zheng et al. "Tour the World: Building a web-scale landmark recognition engine". In: June 2009, pp. 1085–1092. DOI: 10.1109/CVPRW.2009.5206749.