Large-Scale Landmark Recognition using Deep Learning

Aniket Shenoy Prathmesh Jangam Angad Beer Singh
School of Informatics, Computing and Engineering
Indiana University
{ashenoy,pjangam,adhillon}@iu.edu

Abstract

Research in large-scale object recognition and image classification has shown tremendous improvement over the years. Computer vision research is now gaining traction in more fine-grained and instance-level recognition tasks over general entities. Until now, the task of landmark recognition was restricted by the lack of large annotated datasets. However, the recent provision of Google's Landmark dataset has provided an opportunity to address this problem. The dataset consists of over a million images of about 15000 unique natural and man-made landmarks around the world. In this paper, we address this classification problem using deep learning techniques and Convolutional Neural Networks (CNNs).

1. Introduction

The exponential boom in social media websites over the past few years has resulted in copious amounts of photos online, with only Instagram hosting more than 40 million images daily [6]. Most websites provide trivial techniques for browsing images such as keywords based filtering and they lack an effective mechanism for organizing photos. Additionally, not all images uploaded to social media are geotagged i.e. they might not have any Global Positioning System (GPS) metadata associated with them. This poses as an open problem for image retrieval and demands for techniques that can automatically recognize content in large-scale image collections. The task of image retrieval for organizing, browsing and searching images by landmark can be treated as a classification problem of landmark recognition. [5]

The availability of millions of Internet-scale image collections and advancements in state-of-the-art deep learning architectures have provided challenging opportunities in the field of object recognition. [12] Large-scale landmark recognition is one such challenging computer vision problem which involves much more fine-grained and instance-level recognition. For example, as opposed to recognizing general entities such as buildings or mountains, landmark recognition involves recognizing a building as the Colosseum or a mountain as Everest. One of the greatest obstructions in this task yet was the presence of large datasets with annotated ground truths. [1]

Figure 1. Sample images

In this paper, we discuss our approaches to this classification problem on the largest available landmarks dataset to date provided by Google. Although Google has provided a large annotated dataset, there are some caveats associated with it. Many of the less popular landmarks do not have much training data. Another key observation is that since the appearance of landmarks does not change all that much across different images of it, the only variations in the images are due to camera artifacts, capture conditions, occlusions, different viewpoints, weather and illumination. Figure 1 shows a few sample images from the dataset and Figure 2 gives an overview of the geographic distribution of the landmarks in the dataset around the world. [13]

The rest of this paper is structured as follows. In section 2, we give a detailed breakdown of the dataset. In section 3, we review some of the related work in the field of landmark recognition. We present our methodologies and models used in section 4 and in section 5, we present our experimental results. Finally, we summarize and conclude our discussion in section 6.

Figure 2. Geological Distribution of Landmarks

2. Data

The dataset we use was released by Google as part of a Kaggle competition, namely Google Landmark Recognition. The dataset was constructed by clustering images based on geo-locations using an algorithm described in [17]. The ground truth labels between images and landmarks were generated by human annotators. The training data contains 1225029 images of 14951 famous as well as not so famous landmarks from all around the world. The number of images for every predictor landmark class vary highly from class to class. Maximum images in a class are 50337 whereas around 169 classes have only 1 training image. The median number of images for our predictor classes are only 14. In addition to this, 8797 classes have less than 20 images for training. As we know that deep learning methods rely on large data for every class, this problem provides a unique challenge to classify images based on landmarks. Figure 3 depicts the how the frequency of images varies by class.

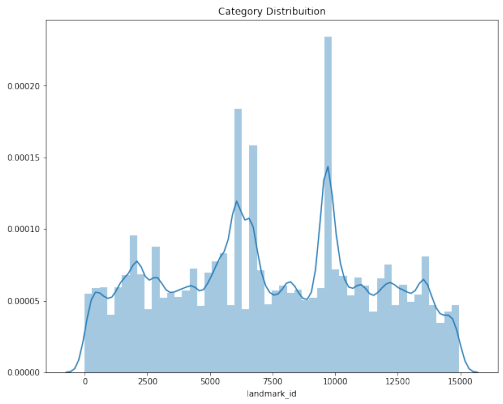

Figure 3. Distribution of classes

3. Related Work

Crandal et al. address the problem of landmark recognition in large-scale social image collections by comparing traditional computer vision techniques to recent breakthroughs in deep learning. They constructed their data by scraping about 2 million images belonging to 500 categories from Flickr based on geotagged metadata. Their first model consists of a Support Vector Machine (SVM) which uses bag-of-words with hand designed image features. Next, inspired by the recent success of deep learning in many computer vision problems, they use CNNs which dramatically outperform the previous approach and even beat humans in some cases.

As pointed out by Crandal et al., the unique selling point of deep learning is the ability to learn both features as well as the classifier in the same framework rather than first hand crafting features and then creating a model to learn them. Inspired by the success of AlexNet in the 2012 ImageNet challenge, they used the model proposed by by Krizhevsky et al. The model was trained using Stochastic Gradient Descent (SGD) with batch size 128 starting with learning rate 0.0001 which decayed by an order of magnitude every 2500 batches. Training continued for 25,000 batches. The CNNs gave an accuracy of 81.4% which was almost 25% higher than baseline for 10-way problem and also outperformed humans (68%). Even with more classes, CNNs still performed better than baseline (40.58% vs 23.88% for 500-way). Thus, Crandal et al. reiterate the power of deep learning over traditional features for problems with large datasets. [5]

Chen at al. propose novel CNN-bases image features for place recognition that identify salient regions and create their regional representations directly from the convolutional activations. These features seem to have better precision-recall characteristics compared to state-of-the-art and are quite robust against changes in viewpoints. General approach involves feeding images directly to a pre-trained CNN which directly flattens activations from a CNN layer to create image representations. Since these representations are created from entire images they are not invariant to occlusions and viewpoint changes. The authors propose a novel feature encoding method that uses one convolutional layer for local feature extraction and another at a higher level for richer semantic details. Images are then represented by distinct regions represented by rectangular boxes. [3]

Sunderhauf et al. [16] established the following results from their experiments for place recognition using ConvNets:

- higher layer features encode semantic information about places
- middle layer features help to provide robustness against variation in appearance because of weather and illumination.
- top level features are robust against viewpoint changes

4. Experiments

In this section we discuss various methods to handle data imbalance as well as various deep learning architectures used to solve this problem.

4.1. Alexnet

Looking at the success of AlexNet in landmark recognition by Crandal et al., we decided to start off by fine-tuning the architecture proposed by Krizhevsky et al [11]. This composed of five convolutional layers followed by three fully connected layers with max pooling and normalization layers between many of the convolutional layers. We then modified the last fully connected layer to accommodate our 100-way classification and initialized the last layer weights using Xavier Initialization [7]. To minimize training time and to get a jumpstart, we adopted transfer learning [14] with pre-trained ImageNet weights. The model was trained using Adam Optimizer [9] with a learning rate of 0.001 and a batch size of 32. Images were serialized to have dimensions (227, 227, 3) to work with this architecture.

4.2. Handling Data Imbalance

Due to our data being highly imbalanced, we try to use various image augmentation techniques to minimize difference in the number of images per class. We initially start by tackling the data imbalance by solving a 100-way problem for top the 100 classes. The training data for our top 100 classes consists of about 400K images with the top class having around 50000 images and our 100th class having around 1000 images. We perform a 80-20 split on our data, where we keep 20% images from every class for validation and use the remaining for training. We then perform data augmentation on the classes with less than 10,000 images in the training set and randomly sample 10,000 images from the classes having more than 10K images. To augment images, we perform various transformations like horizontal flip, rotation at random angles within 25 degrees, shear and skew. These specific transformations help add variations to our images by maintaining the quality of images and minimizing loss of information. Our augmented training set consists of 999896 images with 100 predictor classes.

4.3. Inception

Szegedy et al. proposed an architecture codenamed Inception which allowed the network to choose what spatial features it wants to learn from images. The "Inception module" (shown in Figure 4) consists of multiple convolutional filters of various sizes (5x5, 3x3, 1x1) in parallel and their outputs are concatenated. This enables the learning of both low level and high level features. The network contains many of these ConvNets stacked up one after the other.

Studies have shown state-of-the-art place recognition performance using CNNs pre-trained on object recognition datasets. [4, 15, 16] Inspired by this, we fine-tuned the Inception v3 architecture [2] which contains many improvements over the naive Inception architecture, resulting in a wider and deeper network while keeping the computation cost relatively constant.

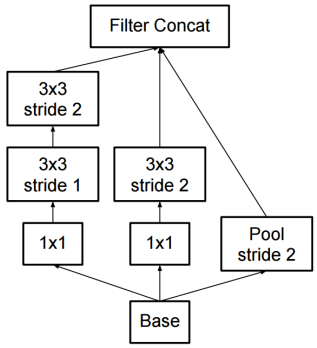

Figure 4. Inception Module

4.3.1 Setup I

In accordance with the Inception architecture, we serialized the images to (299,299,3). Initially, we trained our model for the 100-way problem for 2 epochs and a batch size of 32. Since we initialize the model with pre-trained ImageNet weights, we start off with SGD with a relatively low learning rate of 0.0001 and momentum of 0.9. After the last convolutional layer we apply global average pooling and use a fully connected layer with 1024 hidden units, ReLU activation and dropout for regularization.

4.3.2 Setup II

Next we trained the model on the entire training data encompassing all the classes. For this task, we performed on-the-fly data augmentation. We also perform an additional augmentation namely ZCA Image Whitening which decorelates features using linear algebra, thereby highlighting structures better in images. A number of hyperparameters such as dropout rate, number of hidden units, hidden layers, learning rate, momentum were tried as part of grid search. This particular model took about 10 hours per epoch for training.

4.4. One Shot Classification using Siamese Networks

Since our data has hardly any images available for training for about 9K classes, we decided to perform a One shot classification. In this approach we try to differentiate between landmarks rather than classifying images into landmarks. Additionally using this approach, we only need to have around 1-2 images per class. For our classes having

only 1 image for training we decided to perform a single image augmentation, which resulted in us having at least two images in every class. In this model we use the concept of Siamese networks [10] where our network consists of two identical networks. Both the networks have the same architecture and same weights. We then feed a pair of images to these identical networks, where the images can or cannot belong to the same class. We then define our loss function as:

$$(1-Y)\frac{1}{2}(D_w)^2 + (Y)\frac{1}{2}(max(0, m - D_w))^2$$

where Y is the label indicating if the two images are similar or dissimilar. If a image is of the same class, then Y will be 0. D_w is the euclidean distance between the features generated by the last layer of our convolution network. m is the margin which can be set to a value greater than 0 [8]. Keeping a margin helps our network to decide if the dissimilar images should contribute to our loss. The max value helps the network to identify the dissimilar images even if the network decides that the images are similar. The network was trained using Adam optimizer that minimizes this loss function. We use a convolution network with AlexNet architecture to train our Siamese networks for 100 epochs with learning rate of 0.0001. To calculate the similarity between two images, we calculate the metric D_w which gives us a similarity score. The class that gives the lowest similarity score can be assigned to our test images. We then reserve some images from random classes as validation set to calculate similarity scores.

5. Experimental Results

Method	Accuracy
AlexNet	38% (Top 5)
Inception Setup I	96.88%

Table 1. Results on Top 100 landmarks

After training the model for all 14951 classes, we test the performance of our architecture by sampling 20% of images from every class. Generating a test set from our original training data was challenging as numerous classes had very few images. For the classes having less than 5 images and greater than 1 image, we decided to keep only 1 image for testing. For classes having only 1 image, we did not include them in our testing data but used them to train our model. The inception model seemed to work much better than the Siamese architecture giving us a test accuracy of about 57.21%. We evaluate our model architecture using Siamese networks on the data reserved for validation. The validation set for this architecture consisted of random unseen images from every class. This generated a validation accuracy of around 45.04%. All the accuracy mentioned here indicate the Top-5 accuracy for our models, except for the 100-way classification using Inception.

6. Conclusion

In this paper, we experimented with various architectures to find what works best for this problem. We faced many challenges due to the size of training examples being really low for majority of the classes. We can say that classes having fairly 1k images can be augmented to get a significant accuracy. The Inception model seems to work much better than less sophisticated CNN models like AlexNet, after handling data imbalance. The problem of classifying not so popular landmarks still prevails as our architectures relied on classes to have ample data for training. The Siamese network did not provide much success, due to the variety of images for a class as well as occlusions in images.

7. Future Work

For future work in this project, we would like to implement DEep Local Feature Extractor (DELF) which extracts features from images based on unsupervised K-means clustering on image data [13]. DELF can be useful for large-scale instance-level recognition as it detects semantic local features. With these extracted features, we would like to match the features of our test images with the features extracted from an image from every class of the training data. The images that have the most common features can be said to have taken at the same landmark. Figure 5 shows an example of how DELF can be used to geometrically verify images having the same instance.

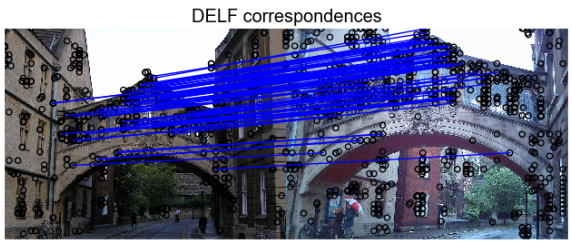

Figure 5. Visualization of local feature matches

8. Acknowledgements

The authors thank Prof. Minje Kim for providing GPU, compute and storage resources and for his valuable feedback throughout this project. We would also like to thank Prof. David Crandal for his advice.

References

[1] A. Araujo and T. Weyand. Google-landmarks: A new dataset and challenge for landmark recognition. *Google Research Blog*, 2018.

[2] V. V. C. Szegedy, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *CVPR*, 2016.

[3] Z. Chen, F. Maffra, I. Sa, and M. Chli. Only look once, mining distinctive landmarks from convnet for visual place recognition. *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Con- ference*, pages 9–16, 2017.

[4] Z. Chen, L. Obadiah, A. Jacobson, and M.Milford. Convolutional neural network based place recognition. *Australiaian Conference on Robotics and Automation*, 2014.

[5] D. Crandal, Y. Li, S. Lee, and D. P. Huttenlocher. Recognizing landmarks in large-scale social image collections. *Large-Scale Visual Geo-Localization*, pages 121–144, 2016.

[6] D. Etherington. Instagram reports 90m monthly active users, 40m photos per day and 8500 likes per second. *TechCrunch*, 2013.

[7] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. *AISTATS*, 9:249–256, 2010.

[8] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. *CVPR*, 2006.

[9] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[10] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. *ICML Deep Learning workshop*, 2015.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *NIPS*, pages 1106–1114, 2012.

[12] Y. Li, D. Crandal, and D. P. Huttenlocher. Landmark classification in large-scale image collections. *ICCV*, 2009.

[13] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large-scale image retrieval with attentive deep local features. *ICCV*, 2016.

[14] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. *CVPR*, 2014.

[15] N. Sunderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, and B. Upcroft. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Proceedings of Robotics: Science and Systems XII*, 2015.

[16] N. Snderhauf, F. Dayoub, S. Shirazi, B. Upcroft, and M. Milford. On the performance of convnet features for place recognition. *IROS*, 2015.

[17] Y.-T. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua, and H. Neven. Tour the world: Building a web-scale landmark recognition engine. *CVPR*, 2009.