Landmark Recognition Kaggle Challenge 2019

Ekaterina Kastrama
kastrama@stanford.edu

Abstract

*In this project, I am going to evaluate a range of deep learning architectures to tackle Google Landmark Recognition 2019 competition, investigating the problem of large scale image classification***.**
Image recognition for problems similar to landmark problem usually tightly connected to image retrieval methods, relevant features extracted during image retrieval algorithms are useful attention mechanism for image classification as well. Combaning common CNN architectures like ResNet50 with different attention mechanism showed improved performance for Landmark recognition task.

1. Introduction

Image detection, retrieval, object detection are the most fundamental and challenging tasks in modern computer vision. Last years there were a lot of advances in computer vision methods due to improved methods and algorithms, as well appropriate software and hardware support.

Image classification task on various types of images successfully being solved by Convolutional Neural Networks. Applying different combination of convolutional filters and other layers allows to extract high level and low level features from the image that allows to distinguish and classify images.

Large scale datasets allow to assess neural networks ability to generalize as images are represented in different conditions, variations of viewpoint, different illumination.

For particular task of landmark recognition, proper classification and retrieval will allow, for example, to organize people's photos, render additional sight information on the fly.

2. Problem statement.

Kaggle 'Google Landmark Recognition 2019' competition' s goal is to classify landmarks on dataset of 4M images. The input is list of images urls and expected output is one of 200k landmark id labels.

Previously, landmark recognition research was lacking large annotated datasets. Competition provides largest worldwide dataset to date, to foster progress in this problem.This competition challenges Kagglers to build models that recognize the correct landmark (if any) in a dataset of challenging test images.

Comparing to previous image classification tasks like ImageNet, the number of classes is much higher (there are more than 200K classes in this challenge), and the number of training examples per class may not be very large.

This competition goal is to build models that recognize the correct landmark (if any).

Challenges of this kaggle competitions are:

- dataset size total train daset size is 500GB
- data is heterogeneous: images can represent different aspects of the landmark, for example image from a museum may contain outdoor images showing the building and indoor images depicting a statue located in the museum. Competition organisers left it 'as is' to avoid bias.
- Many classes with few class members.
- In large scale datasets some challenging conditions might appear such as clutter,occlusion, variation of viewpoint and illumination

3. Related Work

Some winning approaches of last year challenges were based only on ensembling of different CNN architectures like ResNet50, ResNet101 [4], Inception and VGG with different weights. But most research papers considering Landmark recognition as a part of image retrieval techniques due to volume of data and number of possible classification classes. Calculated and ranked or filtered, based on their relevance, features for images retrieval can be used as features of classification algorithms, so image retrieval methods are applicable as improvements of classification algorithms. Oxford5k, Oxford105k, Paris6k and Paris106k datasets are usually used for evaluation of the task.

3.2 Generalized Average pool layer as attention mechanism

Generalized Average pool layer (GeM) was introduced in image retrieval research paper [11] as a method to extract relevant features.

It takes output on any Convolutional layer and produces a vector f as an output of the pooling process. This vector in the case of the conventional global max pooling

The feature vector finally consists of a single value per feature map, i.e. the generalized-mean activation, and its dimensionality is equal to K (K is number of activations of previous CNN layer) The pooling parameter pk can be manually set or learned since this operation is differentiable and can be part of the back-propagation.

3.3. Attention aware GeM (aGeM)

Attention aware GeM (aGeM) [12] is an extension and improvement of GeM.

The network architecture consists of two branches. First, there is the main branch which is exactly same as GeM before the final pooling layer that takes an input image and produces feature maps. For the attention branch, ther are three attention units, denoted Att1, Att2 1, and Att2 2, which are applied to feature maps On different activation layers of base CNN network. The final output of the network applies attention residual learning as in [17] and produces feature maps, followed by GeM pooling.

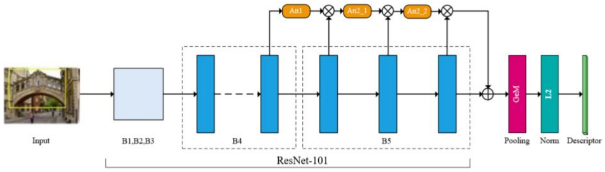
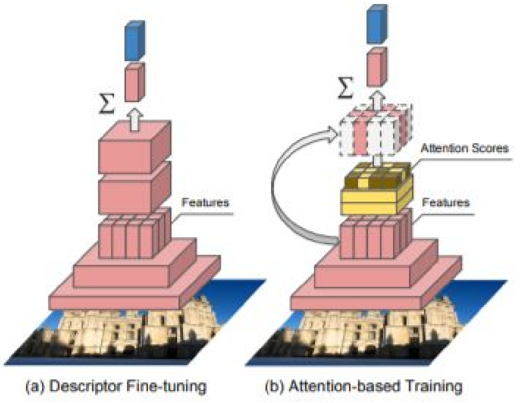


Fig. 1. aGeM architecture

3.1.DELF (Deep Local Features)

DELF architecture (Deep Local Features)[6] is CNN with attention. It allows to train with weak supervision using image-level class labels only, without the need of object- and patch-level annotation . It aims to retrieve semantically meaningful features which can be used for object detection and retrieval tasks. It also showed good results in rejecting false positive examples.

DELF architecture consists of several steps: (i) dense localized feature extraction, (ii) keypoint selection, (iii) dimensionality reduction and (iv) indexing and retrieval.



(a) Descriptor Fine-tuning (b) Attention-based Training

Fig 2. DELF Architecture [8]

On first step of dense feature extraction (local descriptors) fully connected layer utilize ResNet50 conv4_x output. Pixel coordinates of the center of the receptive field considered as feature location.

Attention Network is used to predict features' relevancy score function.

Score function α(fn; θ) is calculated for each feature, where θ denotes the parameters of function α(·). weighted sum defined as

$$\mathbf{y} = \mathbf{W}\left(\sum_n \alpha(\mathbf{f}_n;\theta)\cdot\mathbf{f}_n\right),$$

Random rescalling showed improved attention result. Final steps are dimensionality reduction to 40 using PCA. For image retrieval task KD-tree was used.

4. Method

To find best architecture for Landmark recognition task I tried several approaches such as different loss functions and architecture variation.

For initial analysis, as a starter base algorithms, Resnet50 [4], ResNet101, Inception and VGGPlaces365[16] were explored.

VGGPlaces365 is VGG pretrained “Places” dtaabase, containing more than 10 million images comprising 400+ unique scene categories. The dataset features 5000 to 30,000 training images per class, consistent with real-world frequencies of occurrence.

Experimental architecture was used to combine output of pretrained VGG Places365 [16]

Network with ResNet50 with last freeze layers concatenating output of those.

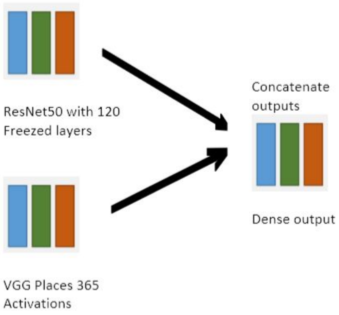


Fig. 3 Alternative Architecture

Different loss functions were tried as well:

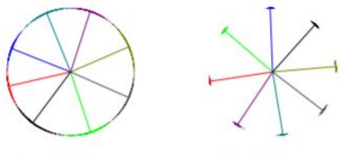
1) custom binary cross entropy reweighted loss that rerank confident wrong classes. This loss was not computationally effective.

2) Additive Angular Margin Loss (ArcFace) [10], which showed good results on face recognition task, defined as

$$L_3 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}}$$

Where **m** additive angular margin penalty between xi and Wy, **s** - hypersphere radius.

Arc Loss believed to enlarge gaps between nearest classes.



(a) Softmax (b) ArcFace

Those losses didn’t show good results within acceptable performance.

Main model was using standard binary cross entropy defined as

$$BCE = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot log(\hat{y}_i) + (1-y_i)\cdot log(1-\hat{y}_i)$$

3.1. Generalized Average pool layer (GEM).

Generalized Average pooling layer is trainable pooling layer to suppress irrelevant convolution layer features.

It can be applied on the top of any fully convolutional CNN, such as AlexNet, VGG, or ResNet, while their fully-connected layers are discarded.

Previously,global pooling layers were widely used e.g. max pooling [9], average pooling [10], hybrid pooling [15], weighted average pooling [13], and regional pooling [14]. Pooling layer based on a generalized-mean that has learnable parameters, either one global or one per output dimension. Both max and average pooling are its special cases. GEM showed significant performance boost over standard non-trainable pooling layers.

GEM layer defined as:

$$\mathbf{f}^{(g)} = [\mathrm{f}_1^{(g)} \dots \mathrm{f}_k^{(g)} \dots \mathrm{f}_K^{(g)}]^\top, \quad \mathrm{f}_k^{(g)} = \left(\frac{1}{|\mathcal{X}_k|}\sum_{x\in\mathcal{X}_k} x^{p_k}\right)^{\frac{1}{p_k}}.$$

Where x is activation, pk is GEM parameter

3.2 Final Architecture

The most effective network had following architecture:

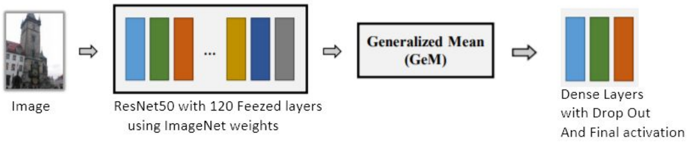


Fig. 4 Final Architecture

5. Dataset

The training dataset[9] consists of 4 132 914 training images and was constructed by mining web landmark images with permissive licenses, leveraging crowdsourced labels. Compared to last year’s challenge, the dataset is 3X larger and more diverse (10X larger number of classes), and noisy without any data preprocessing by organizers.

Training dataset highly unbalanced, containing 203 094 classes, some classes have only few examples.

For this project subset of training set was used that corresponds to labels that have at least 250 samples, which resulted in 1067 classes and 478 577 images.

For test and validation subset of 10 241 samples was used.

Original images has different size, but was rescaled for training and optimization purposes to 128x128.

Random corps were used as argumentation technique during training

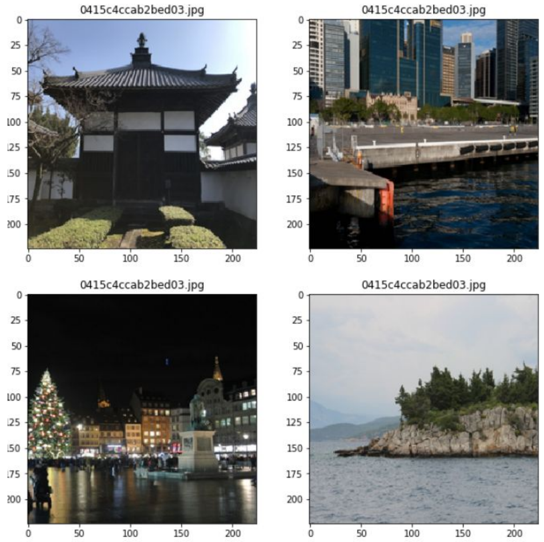


Fig. 5: Landmark 2019 Dataset examples

6. Experiments and Results.

6.1 Hyperparameters

Training process was using ADAM optimizer with manually decreasing learning rate from 0.001 to 0.00001. Batch size of 128 samples was most memory efficient.

Number of image crops during pre-porcessing was also tuned as parameter.

6.2 Evaluation metrics

Training and optimization was performed using two evaluation metrics: Class Accuracy and Global Average Precision (GAP) at k, where k=1(also known as micro Average Precision (microAP)) [1][2]:

$$GAP = \frac{1}{M}\sum_{i=1}^{N} P(i)rel(i)$$

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

6.3 Results

From all experimentations with loss and Network Architecture the most effective was combination of Resnet50 and GeM average pooling.

Alternative architecture, which was concatenating Outputs of VGG places and ResNet was showing promising results, turned out extremely heavy and caused to reduce batch size to 32, which slowed down training process.

Even though Resnet50+ GeM was showing better results than just ResNet it prone to overfit, especially for training with crops. Adding Dropout layers was effective but was slowing down the training improvements speed

Adding original image crops was busting accuracy but was overfitting even more.

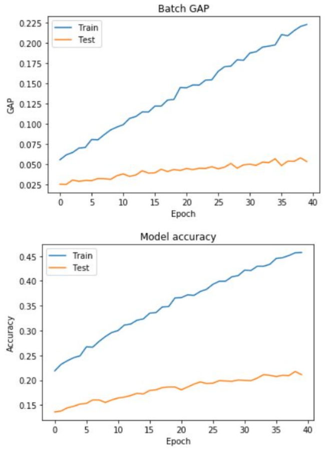


Fig 6. Training process for architecture without random Crops

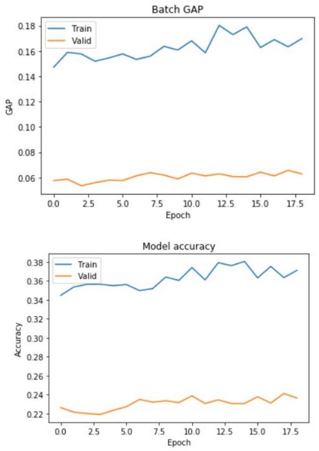


Fig 7. Overfitted Training process for architecture with random Crops

	Global average precise	Class Accuracy
Baseline: 2018 leaderboard top (different dataset) [7]	0.30	
Baseline: 2019 leaderboard top [8]	0.37	
ResNet50 test set	0.012	0.14
TEST ResNet50 +GEM	0.04	0.19
TRAIN ResNet50 +GEM	0.2	0.45
TEST ResNet50 +GEM with crops	0.05	0.25
TRAIN ResNet50 +GEM with crops	0.16	0.36
TRAIN VGGPlaces 365 +ResNet50	0.1249	0.32

7. Conclusion and Future Work

Classic CNN like ResNet50 with combination of attention and relevance mechanism of Global Generalized average pooling showed improved performance and acceptable computational performance comparing to ResNet50 network alone. This is proving efficiency of GeM approach. Random crops also showed better results. There is room for future improvements of Landmark recognition classification solution, utilizing other feature extraction and ranking approaches, like DELF.

8. Code

Hardware: We used various GPU's on a Google Cloud VM Instance

Github: https://github.com/EkateKK

References

[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] https://www.kaggle.com/c/landmark-recognition-2019/overview/evaluation

[3] Previous competition results https://drive.google.com/file/d/12Zb0NZL3Ys6SPLiAvCroW5w3gZIYCB2X/view

[4] Deep Residual Learning for Image Recognition https://arxiv.org/pdf/1512.03385.pdf

[5] Detect-to-Retrieve: Efficient Regional Aggregation for Image Search https://arxiv.org/pdf/1812.01584.pdf

[6] Large-Scale Image Retrieval with Attentive Deep Local Features https://arxiv.org/pdf/1612.06321.pdf

[7] https://www.kaggle.com/c/landmark-recognition-challenge/leaderboard

[8] https://www.kaggle.com/c/landmark-recognition-2019/leaderboard

[9] Landmark recognition 2019 data https://github.com/cvdfoundation/google-landmark

[10] ArcFace: Additive Angular Margin Loss for Deep Face Recognition https://arxiv.org/pdf/1801.07698.pdf

[11] Generalized-mean pooling

https://arxiv.org/pdf/1711.02512.pdf
[12] ATTENTION-AWARE GENERALIZED MEAN POOLING FOR IMAGE RETRIEVAL
https://arxiv.org/pdf/1811.00202.pdf
[13][Y. Kalantidis, C. Mellina, and S. Osindero, “Cross-dimensional
weighting for aggregated deep convolutional features,” in ECCVW, 2016. 1, 3, 6, 12
[14] G. Tolias, R. Sicre, and H. Jegou, “Particular object retrieval with ´
integral max-pooling of CNN activations,” in ICLR, 2016. 1, 3, 4,
6, 10, 12
[15] A. Mousavian and J. Kosecka, “Deep convolutional features for image based retrieval and scene categorization,” in
arXiv:1509.06033, 2015.
[16] VGGPlaces365
https://github.com/CSAILVision/places365
[17] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, “Residual attention network for image classification,” in CVPR, 2017.