Two-stage Discriminative Re-ranking for Large-scale Landmark Retrieval

Shuhei Yokoo
University of Tsukuba
yokoo@cvlab.cs.tsukuba.ac.jp

Kohei Ozaki
Preferred Networks, Inc.*
ozaki@preferred.jp

Edgar Simo-Serra
Waseda University
ess@waseda.jp

Satoshi Iizuka
University of Tsukuba
iizuka@cs.tsukuba.ac.jp

Abstract

We propose an efficient pipeline for large-scale landmark image retrieval that addresses the diversity of the dataset through two-stage discriminative re-ranking. Our approach is based on embedding the images in a feature-space using a convolutional neural network trained with a cosine softmax loss. Due to the variance of the images, which include extreme viewpoint changes such as having to retrieve images of the exterior of a landmark from images of the interior, this is very challenging for approaches based exclusively on visual similarity. Our proposed re-ranking approach improves the results in two steps: in the sort-step, k-nearest neighbor search with soft-voting to sort the retrieved results based on their label similarity to the query images, and in the insert-step, we add additional samples from the dataset that were not retrieved by image-similarity. This approach allows overcoming the low visual diversity in retrieved images. In-depth experimental results show that the proposed approach significantly outperforms existing approaches on the challenging Google Landmarks Datasets. Using our methods, we achieved 1st place in the Google Landmark Retrieval 2019 challenge on Kaggle. Our code is publicly available here: https://github.com/lyakaap/Landmark2019-1st-and-3rd-Place-Solution

1. Introduction

Image retrieval is a fundamental problem in computer vision where given a query image, similar images must be found in a large dataset. In the case of landmark images, the variation between points of view and different parts of the landmark can be extreme, proving challenging for humans without deep knowledge of the landmark in question. One such complicated example is shown in Fig. 1. The Scuderie del Quirinale is very visually similar to other structures such as the Vatican obelisk and the Inco Superstack, leading to erroneous retrievals. Our proposed re-ranking approach is able to exploit labeled information from the training dataset to improve the retrieval results, even when the correct images are very visually dissimilar such as drawings, different viewpoints, diverse illumination, etc.

Instance image retrieval can be seen as the task of converting the image information into an embedding where similar images are nearby. Similar to recent approaches,
we focus on learning this embedding with a convolutional neural network (CNN). We adopt a cosine softmax loss to train the neural network for the retrieval task. Afterward, the same prediction is performed for a query online, and initial retrieval results are obtained by k-NN search. For re-ranking, we assign “positive” or “negative” to the retrieval results based on their prediction results. Final results are obtained by two-stage discriminative re-ranking.

To summarize our contributions, (1) an effective pipeline for high quality landmark retrieval, (2) a re-ranking approach based on exploiting labels, and (3) results that significantly outperform existing approaches on challenging datasets.

2. Related Work

Instance Image Retrieval. Image retrieval is usually posed as a problem of finding an image embedding in which similar images have small distance, and has been traditionally done based on local descriptor based methods [9, 24, 34, 42], including the popular SIFT [30], RootSIFT [2], and SURF [4]. Bag-of-Words [9, 42] model and its variants (VLAD [24], Fisher Vector [34], etc.) have been popular in image retrieval previous to the advent of learning-based approaches, and construct image embeddings by aggregating local descriptors. More recently, DELF [32] has been proposed as a deep learning-based local descriptor method, which uses the attention map of CNN activation learned by only image-level annotation. See [51] for a survey of instance image retrieval.

After the emergence of deep learning, many image retrieval methods based on deep learning have been presented. Most recent image retrieval approaches are based on deep learning [1, 3, 15, 25, 26, 37, 39]. Both utilizing off-the-shelf CNN activations as an image embedding [3, 25, 26, 39] and further fine-tuning to specific datasets [1, 15, 37] are popular approaches. An extension of VLAD called NetVLAD which is differentiable and trainable in an end-to-end fashion has also been recently proposed [1]. Gordo et al. [15] proposed using a region proposal network to localize the landmark re-

Figure 2: Overview of our approach. In an offline step, image embeddings of the labeled set and index set are calculated, and the instance-id of each index set image is predicted by k-NN soft-voting. Afterward, the same prediction is performed for a query online, and initial retrieval results are obtained by k-NN search. For re-ranking, we assign “positive” or “negative” to the retrieval results based on their prediction results. Final results are obtained by two-stage discriminative re-ranking.
a method that checks the geometric consistency using local descriptors and RANSAC [14], can be combined with QE to filter images used for expansion [8]. SP can be used as re-ranking [7, 36] to improve precision, but it has an efficiency problem. Therefore, it is performed generally on a shortlist of top-ranked images only. HQE [45] leverages Hamming Embedding [23] to filter images instead of SP.

Diffusion, also known as similarity propagation, is a major manifold-based approach which can also be used for re-ranking [12]. Many diffusion approaches have been proposed for enhancing the performance of instance image retrieval [12, 21, 22, 49]. Diffusion can capture the image manifold in the feature space by random-walk on $k$-NN graph. However, because diffusion process tends to be expensive, spectral methods have been proposed to reduce computational cost [21], and Yang et al. [49] proposed decoupling diffusion into online and offline processes to reduce online computation. EGT [5] is a recently proposed $k$-NN graph traversal algorithm, which outperforms diffusion methods in terms of performance and efficiency.

Conventional re-ranking methods are unsupervised, which means they do not consider label information even when label information is available. In contrast, our re-ranking method can exploit label information, commonly available in many problems, and shows excellent performance in landmark retrieval tasks.

## 3. Method

Our approach consists of training an embedding space using a cosine softmax loss to train a CNN. Afterward, retrieval is done based on $k$-NN search which is corrected and improved using two-stage discriminative re-ranking.

### 3.1. Embedding Model

Our model is based on a CNN that embeds each image into a feature-space amenable for $k$-NN search. Our model is based on a ResNet-101 [17] augmented with Generalized Mean (GeM)-pooling [37] to aggregate the spatial information into a global descriptor.

The reduction of a descriptor dimension is crucial since it dramatically affects the computational budget and alleviates the risk of over-fitting. We reduce the dimension to 512 from 2048 by adding a fully-connected layer after the GeM-pooling layer. Additionally, a one-dimensional Batch Normalization [20] after the fully-connected layer is used to improve the generalization ability.

Training is done using the ArcFace [11] loss with $L_2$ weight regularization defined as follows:

$$ L = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^s(\cos(\theta_{y_i} + m))}{e^s(\cos(\theta_{y_i} + m)) + \sum_{j=1, j \neq y_i}^{n} e^s(\cos(\theta_{y_j}))} + \beta(||W||_2^2 + ||W_M||_2^2), \quad (1) $$

where $L_2$ regularization is applied to prevent over-fitting. The loss function encourages the cosine similarity between the predicted and the target classes to be as close as possible.

---

**Retrieval Loss Functions.** Instance image retrieval requires image embedding that captures the similarity well, and the loss used during learning plays an important role. Using CNN off-the-shelf embeddings has been effective for image retrieval [3, 25, 26, 39]. Babenko and Lempitsky [3] proposed using sum-pooling of CNN activation, and Lin et al. [26] proposed max-pooling of multiple regions of CNN activation. However, training specifically for the task of instance retrieval has shown more effective with contrastive loss [6] and triplet loss [18, 41, 48] being some of the more used losses in image retrieval [13, 15, 19, 37]. Recently, the Average-Precision (AP) loss [40], which optimizes the global mean average precision directly by leveraging list-wise loss formulations, has been proposed and achieved state-of-the-art results. In face recognition field, recently cosine softmax losses [11, 27, 28, 38, 46, 47, 50] have shown astonishing results and have become more favorable than other losses [31]. Cosine softmax losses impose L2-constraint to the features which restricts them to lie on a hypersphere of a fixed radius, with popular approaches being SphereFace [27, 28], ArcFace [11], and CosFace [46, 47], using multiplicative angular margin penalty, additive angular margin penalty, and additive cosine margin penalty, respectively. While contrastive loss and triplet loss require training techniques such as hard negative mining [1, 41], cosine softmax losses do not and easy to implement and stable in training. We show their successes are not only in face recognition but also in instance image retrieval by comparative experiments.

**Re-ranking Methods.** Re-ranking is an essential approach to enhance the retrieval results on the image embedding. Query expansion (QE)-based techniques are simple and popular ways of re-ranking for improving recall of retrieval system. AQE [8] is the first work that applies query expansion in vision field, and is based on averaging embeddings of top-ranked images retrieved by an initial query, and using the averaged embedding as a new query. αQE [37] uses weighted average of descriptors of top-ranked images. Heavier weights are put on as the rank gets higher. DQE [2] uses an SVM classifier and its signed distance from the decision boundary for re-ranking. Spatial verification (SP) [33, 35] is
with

$$\theta_{y_i} = W^T_y f(x_i; W_M) \quad \text{and} \quad \theta_j = W^T_j f(x_i; W_M),$$  \hspace{1cm} (2)

where $x_i$ is the input image with target class $y_i$, $N$ is the batch size, $W$ denotes the weights of the last layer, $W_M$ is the parameters of the whole network excluding the last layer, $f(x; W_M)$ is the embedding of $x$ using $W_M$, $s$ is a scaling hyperparameter, and $m$ is a margin hyperparameter. We note that $\|W\|_2 = 1$ and $\|x_i\|_2 = 1$ is enforced by normalizing at every iteration.

### 3.2. Two-stage Discriminative Re-ranking

The diversity of images belonging to the same instance is one of the main problems in image retrieval. For example, an instance of church may contain diverse samples, such as outdoor and indoor images. These images are extremely hard to identify as the same landmark without any context. Furthermore, the visual dissimilarity makes it nearly impossible to retrieve them using only visual-based embeddings. To overcome this issue, we propose two-stage discriminative re-ranking that exploits the label information. An overview of our re-ranking approach is shown in Fig. 2.

Our proposed method is composed of an auxiliary offline step and two re-ranking stages. Suppose we have a query, an index set and a labeled set. The index set is a database for which we perform image retrieval and has no labels, only images. First, we predict the instance-id of each sample from the index set by $k$-NN search with soft-voting, where each sample from the index set is regarded as a query, and the labeled set as a database.

The score of each instance-id is calculated by accumulating the cosine similarities of the $k$ nearest samples as follows

$$v(x, c) = \frac{1}{k} \sum_{x' \in N(x)} f(x')^T f(x) \cdot \mathbf{1}(\text{label}(x') = c),$$  \hspace{1cm} (3)

where $f(\cdot) \in \mathbb{R}^d$ is the feature embedding function with $W_M$ omitted for brevity, $N(x)$ is the set of $k$ nearest neighbours in the labeled set, and $\mathbf{1}(\cdot)$ is an indicator function. The prediction then becomes the class that maximizes $v(x, c)$ for $x$. The index set prediction can be computed in an offline manner once.

When a query is given, its instance-id is also predicted in the same way described above. Index set samples that are predicted to be the same id of the query sample are treated as “positive samples”, and those of different id as “negative samples”, and play an important role in our re-ranking approach.

Our re-ranking method is illustrated in Figure 3 and consists of a sort-step and insert-step. The top row in the figure shows a query (in blue) and retrieved samples from the index set by $k$-NN search with positive samples shown in green and negative samples shown in red. Here, we consider images on the left to be more relevant to the query than the ones on the right. In the sort-step, positive samples are moved to the left of the negative samples in the ranking, maintaining the relative order of them. This re-ranking step can make results more reliable, becoming less dependent on factors such as lighting, occlusions, etc.

In the insert-step, we insert positive samples from the entire index set, which are not retrieved by the $k$-NN search, after the re-ranked positive samples in descending order of scores which is calculated by $k$-NN cosine similarities. This step enables us to retrieve visually dissimilar samples to a query by utilizing the label information of the labeled set. Here, the predicted instance-id may not always be reliable, especially when the prediction score is low. If the instance of query does not exist in the labeled set, there is a tendency that the prediction score becomes very low. Thus, we do not perform insertion to the sample of which the sum of the prediction score between query and sample considered to be

![Figure 3: Overview of our re-ranking procedure. “Positive” represents the samples predicted the same id as the predicted id of a query sample. “Negative” represents the samples predicted the different id from a predicted id of a query sample. Re-ranking is performed in each step based on their predicted id results.](image-url)

inserted is lower than a threshold $\tau_{\text{score}}$ to deal with this.

4. Dataset

The Google Landmarks Dataset (GLD) is the largest dataset of instance image retrieval, which contains photos of landmarks from all over the world. The photos include a lot of variations, e.g., occlusion, lighting changes. GLD has three versions: v1, v2, and v2.1 and we overview their differences in Table 1. GLD-v1 [32] which is the first version of GLD has released in 2018. This dataset has more than 1 million samples and around 15 thousand labels. GLD-v1 was created based on the algorithm described in [52], and uses visual features and GPS coordinates for ground-truth correction. Simultaneously, the Google Landmarks Challenge 2018 was launched and GLD-v1 was used at this challenge. Currently, we can still download the dataset, but cannot evaluate with it since ground-truth was not released. GLD-v2 \(^1\) used for the Google Landmarks Challenge 2019, is the largest worldwide landmark recognition dataset available at the time. This dataset includes over 5 million images of more than 200 thousands of different landmarks. It is divided into three sets: train, test, and index. Only samples from the train set are labeled.

Since GLD-v2 was constructed by mining web landmark images without any cleaning step, each category may contain quite diverse samples: e.g., images from a museum may contain outdoor images showing the building and indoor images depicting a statue located in the museum. In comparison with the GLD-v1, there is significantly more noise in the annotations. The GLD-v2.1 is a minor update of GLD-v2. Only ground truth of test set and index set are updated.

Automated Data Cleaning. The train set of GLD-v2 is very noisy because it was constructed by mining web landmark images without any cleaning step. Furthermore, training with the entire train set of GLD-v2 is complicated due to its huge scale. Therefore, we consider to automatically remove noises such as mis-annotation inspired by [15], leading to reduction of dataset size and training budget, while avoiding adverse effects of the noise for deep metric learning.

To build a clean train set, we apply spatial verification \([35]\) to filtered images by $k$-NN search. Specifically, cleaning the train set consists of a three-step process. First, for each image descriptor $x_i$ in the train set, we search its 1000 nearest neighbors from the train set. This image descriptor is obtained by our embedding model learned from the GLD-v1 dataset. Second, spatial verification is performed on up to the 100 nearest neighbors assigned to the same label as $x_i$. For spatial verification, we use RANSAC \([14]\) with affine transformation and deep local attentive features (DELF) \([32]\). If an inlier-count between $x_i$ and nearest neighbor image descriptor is greater than 30, we consider the nearest neighbor as a verified image. Finally, if the count of verified images in the second step reaches the threshold $\tau_{\text{freq}}$, $x_i$ is added to the cleaned dataset. We set $\tau_{\text{freq}} = 3$ in our experiment.

This automated data cleaning is very costly due to the use of spatial verification, however, it only has to be run once. Table 1 summarizes the statistics of the dataset used in our experiments. We show the effectiveness of using our cleaned dataset through our experiments in the following sections.

5. Experiments

We pre-train the model on ImageNet \([10]\) and the train set of GLD-v1 \([32]\) first, before being trained on cleaned GLD-v2 train set with a cosine softmax loss. We use $p = 3.0$ for the Generalized Mean-pooling, and use 512-dimension embedding space. We use a margin of 0.3 for the ArcFace loss and $\beta = 10^{-5}$ for the regularization term. For re-ranking we use $\tau_{\text{score}} = 0.6$ and $k = 3$ for $k$-NN soft-voting.

We train each network for 5 epochs with commonly used data augmentation methods such as brightness shift, random cropping, and scaling. In particular, images are randomly scaled between 80% and 120% of their original size and then either cropping or zero-padding is used to return the image to the original resolution, depending on whether the image was downscaled or up-scaled. Brightness is randomly modified by 0% to 10%. When constructing mini-batches for training, the images are resized to be the same size for efficient training. This might cause distortions to the input images, degrading the accuracy of the network \([16]\). To avoid this, we choose mini-batch samples so that they have similar aspect ratios, and resize them to a particular size. The size is determined by selecting tuple of width and height from \{(512, 352), (512, 384), (448, 448), (384, 512), (352, 512)\} depending on their aspect ratio.

Model training is done by using the stochastic gradient descent with momentum, where initial learning rate, momentum, and batch size are set to 0.001, 0.9, and 32, respectively. The cosine annealing \([29]\) learning rate scheduler is used during training.

For other approaches we compare to, we follow the settings described in their respective papers. However, we have changed some hyperparameters which would found to give non-competitive results. In particular, spatial verifica-

---

\(^1\)https://github.com/cvdfoundation/google-landmark

<table>
<thead>
<tr>
<th>Dataset (train set)</th>
<th># Samples</th>
<th># Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLD-v1</td>
<td>1,225,029</td>
<td>14,951</td>
</tr>
<tr>
<td>GLD-v2/2.1</td>
<td>4,132,914</td>
<td>203,094</td>
</tr>
<tr>
<td>GLD-v2/2.1 (clean)</td>
<td>1,580,470</td>
<td>81,313</td>
</tr>
</tbody>
</table>

Table 1: Dataset statistics used in our experiments. The index and test images are not included here. GLD-v2 and GLD-v2.1 only differ in the index set and test set and thus are shown together for the train set.
Table 2: Comparison of our re-ranking against the other state-of-the-art re-ranking methods on top of our baseline. We report mAP@100 in GLD-v2 and mAP@100, P@10, and MeanPos in GLD-v2.1. mAP@100 is mean average precision at rank 100. P@10 is mean precision at rank 10 and higher is better. MeanPos is the mean position of the first relevant image (if no relevant image in top-100, use 101 as position) and lower is better.

<table>
<thead>
<tr>
<th>Method</th>
<th>GLD-v2 Private</th>
<th>GLD-v2 Public</th>
<th>GLD-v2.1 Private</th>
<th>GLD-v2.1 Public</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mAP@100</td>
<td>mAP@100</td>
<td>mAP@100 P@10 MeanPos</td>
<td>mAP@100 P@10 MeanPos</td>
</tr>
<tr>
<td>k-NN search</td>
<td>30.22</td>
<td>27.81</td>
<td>29.63 30.76 27.02</td>
<td>27.66 28.87 32.60</td>
</tr>
<tr>
<td>SP [36]</td>
<td>23.75</td>
<td>22.40</td>
<td>23.29 24.72 28.72</td>
<td>22.15 23.46 33.18</td>
</tr>
<tr>
<td>AQE [8]</td>
<td>32.17</td>
<td>30.47</td>
<td>31.60 32.97 27.44</td>
<td>30.28 31.35 31.54</td>
</tr>
<tr>
<td>αQE [37]</td>
<td>32.21</td>
<td>30.34</td>
<td>31.71 33.04 26.67</td>
<td>30.23 31.03 31.13</td>
</tr>
<tr>
<td>Iscen et al.’s DFS [22]</td>
<td>32.01</td>
<td>30.55</td>
<td>31.91 32.51 29.52</td>
<td>30.81 30.50 33.23</td>
</tr>
<tr>
<td>Yang et al.’s DFS [49]</td>
<td>31.20</td>
<td>29.36</td>
<td>30.90 31.48 29.87</td>
<td>29.29 29.63 33.83</td>
</tr>
<tr>
<td>EGT [5]</td>
<td>30.33</td>
<td>28.44</td>
<td>31.00 32.89 34.82</td>
<td>29.77 30.74 38.19</td>
</tr>
<tr>
<td>Ours</td>
<td>36.85</td>
<td>34.89</td>
<td>36.04 36.27 24.43</td>
<td>34.41 33.40 29.23</td>
</tr>
<tr>
<td>Ours + αQE</td>
<td>37.34</td>
<td>35.59</td>
<td>36.55 36.68 24.44</td>
<td>35.12 33.85 28.11</td>
</tr>
</tbody>
</table>

Table 3: Ablation study of each step of our re-ranking. We show the effect of adding the sort-step and both the sort-step and insert-step with respect to our strong baseline on the GLD-v2 dataset.

<table>
<thead>
<tr>
<th>Description</th>
<th>Private</th>
<th>Public</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>30.22</td>
<td>27.81</td>
</tr>
<tr>
<td>+ Sort-step</td>
<td>33.79</td>
<td>30.91</td>
</tr>
<tr>
<td>+ Insert-step</td>
<td>36.85</td>
<td>34.89</td>
</tr>
</tbody>
</table>

Spatial verification (SP) follows the procedure from [36] except for using DELF [32] trained with GLD-v1 as the local descriptor. In AQE [8] and αQE [37], the number of retrieved results used for query expansion are set to 10 including the query itself. The α of αQE is set to 3.0. SP is not used to filter samples for the construction of a new query in QE different from [8]. For Iscen et al.’s diffusion (DFS) [22] and Yang et al.’s diffusion (DFS) [49], the default hyperparameters are used. The threshold t of EGT [5] is set to inf. Hyperparameters of each method are tuned using the GLD-v2 Public split.

We use multi-scale feature extraction described in [15] during test time in whole experiments. The resulting features are finally averaged and re-normalized.

5.1. Evaluation Protocol

We use the Google Landmarks Dataset (GLD) [32] for experiments. GLD-v1 and GLD-v2 have three data splits: train, index and test set. The train set of GLD-v1 and GLD-v2 is used for training. Additionally, the train set of GLD-v2 is used as a labeled set for re-ranking. The index set and the test set of GLD-v2 and GLD-v2.1 are used for our evaluation. The index set and the test set of GLD-v1 are not used for our evaluation since we cannot obtain ground-truth of GLD-v1 and use evaluation server. Note that evaluation on GLD-v2 are performed on evaluation server of the competition page 2 and it shows only mAP@100. We report two split results, “Private” and “Public”. The Private split accounts for 67% and the Public split accounts for 33% of GLD-v2 and GLD-v2.1 respectively.

5.2. Results

We evaluate our re-ranking method and other state-of-the-art re-ranking methods on top of our baseline in Table 2, evaluating on the GLD-v2 and GLD-v2.1 datasets. Baseline is the retrieved results by k-NN search using descriptors extracted by our trained model. Surprisingly, spatial verification (SP) [36] harms the performance drastically in contrast to the common sense of instance image retrieval. After visual inspection of the results of SP, we hypothesize that this is likely caused by a large number of instances that are very similar. There are many cases where the RANSAC inlier count increases artificially due to geometrical consistency of partial region between even different instances, degrading accuracy as a result.

Experimental results show that our approach outperforms the previous re-ranking approaches on the challenging GLD dataset. Furthermore, a combination of ours and αQE boosts the performance, and it suggests that our re-ranking method can be combined with existing re-ranking methods to further improve performance.

We perform an ablation study and report the result in Table 3 to validate each step in our re-ranking approach. We can see that both the sort-step and insert-step significantly improve results with respect to the k-NN search-only baseline.

References


[39] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan,


