Neural Codes for Image Retrieval

Artem Babenko[1,3], Anton Slesarev[1],
Alexandr Chigorin[1], and Victor Lempitsky[2]

[1] Yandex, Russia
[2] Skolkovo Institute of Science and Technology (Skoltech), Russia
[3] Moscow Institute of Physics and Technology, Russia

Abstract. It has been shown that the activations invoked by an image within the top layers of a large convolutional neural network provide a high-level descriptor of the visual content of the image. In this paper, we investigate the use of such descriptors (neural codes) within the image retrieval application. In the experiments with several standard retrieval benchmarks, we establish that neural codes perform competitively even when the convolutional neural network has been trained for an unrelated classification task (e.g. Image-Net). We also evaluate the improvement in the retrieval performance of neural codes, when the network is retrained on a dataset of images that are similar to images encountered at test time.

We further evaluate the performance of the compressed neural codes and show that a simple PCA compression provides very good short codes that give state-of-the-art accuracy on a number of datasets. In general, neural codes turn out to be much more resilient to such compression in comparison other state-of-the-art descriptors. Finally, we show that discriminative dimensionality reduction trained on a dataset of pairs of matched photographs improves the performance of PCA-compressed neural codes even further. Overall, our quantitative experiments demonstrate the promise of neural codes as visual descriptors for image
retrieval.

Keywords: image retrieval, same-object image search, deep learning, convolutional neural networks, feature extraction.

1 Introduction

Deep convolutional neural networks [13] have recently advanced the state-of-the-art in image classification dramatically [10] and have consequently attracted a lot of interest within the computer vision community. A separate but related to the image classification problem is the problem of image retrieval, i.e. the task of finding images containing the same object or scene as in a query image. It has been suggested that the features emerging in the upper layers of the CNN learned to classify images can serve as good descriptors for image retrieval. In particular, Krizhevsky et al. [10] have shown some qualitative evidence for that.

D. Fleet et al. (Eds.): ECCV 2014, Part I, LNCS 8689, pp. 584–599, 2014.

Here we interesed in establishing the quantitative performance of such features (which we refer to as *neural codes*) and their variations.

We start by providing a quantitative evaluation of the image retrieval performance of the features that emerge within the convolutional neural network trained to recognize Image-Net [1] classes. We measure such performance on four standard benchmark datasets: INRIA Holidays [8], Oxford Buildings, Oxford Building 105K [19], and the University of Kentucky benchmark (UKB) [16]. Perhaps unsurprisingly, these deep features perform well, although not better than other state-of-the-art holistic features (e.g. Fisher vectors). Interestingly, the relative performance of different layers of the CNN varies in different retrieval setups, and the best performance on the standard retrieval datasets is achieved by the features in the middle of the fully-connected layers hierarchy.

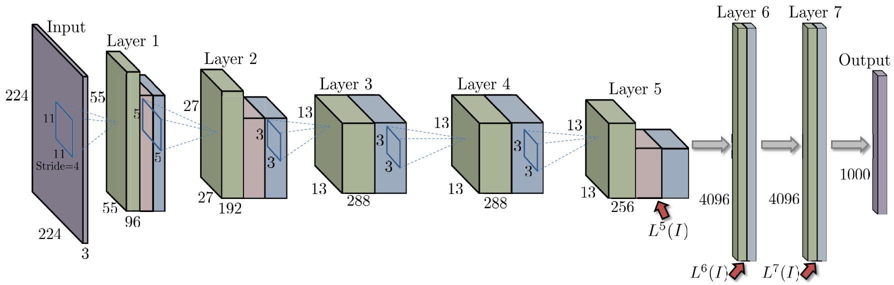

Fig. 1. The convolutional neural network architecture used on our experiments. Purple nodes correspond to input (an RGB image of size 224 × 224) and output (1000 class labels). Green units correspond to outputs of convolutions, red units correspond to the outputs of max pooling, and blue units correspond to the outputs of rectified linear (ReLU) transform. Layers 6, 7, and 8 (the output) are fully connected to the preceding layers. The units that correspond to the neural codes used in our experiments are shown with red arrows. Stride=4 are used in the first convolutional layer, and stride=1 in the rest.

The good performance of neural codes demonstrate their universality, since the task the network was trained for (i.e. classifying Image-Net classes) is quite different from the retrieval task we consider. Despite the evidence of such universality, there is an obvious possibility to improve the performance of deep features by adapting them to the task, and such adaptation is the subject of the second part of the paper. Towards this end, we assemble a large-scale image dataset, where the classes correspond to landmarks (similar to [14]), and retrain the CNN on this collection using the original image-net network parameters as initialization. After such training, we observe a considerable improvement of the retrieval performance on the datasets with similar image statistics, such as INRIA Holidays and Oxford Buildings, while the performance on the unrelated UKB dataset degrades. In the second experiment of this kind, we retrain the initial network on the Multi-view RGB-D dataset [12] of turntable views of different objects. As expected, we observe the improvement on the more related UKB dataset, while the performance on other datasets degrades or stays the same.

Finally, we focus our evaluation on the performance of the compact versions of the neural codes. We evaluate the performance of the PCA compression and observe that neural codes can be compressed very substantially, e.g. to 128 dimensions, with virtually no loss of the retrieval accuracy. Overall, the degradation from the PCA compression incurred by the neural codes is considerably smaller than the degradation incurred by other holistic descriptors. This makes the use of neural codes particularly attractive for large-scale retrieval applications, where the memory footprint of a descriptor often represents the major bottleneck.

Pushing the compression to the extreme, to e.g. 16 dimensions leads to considerable degradation, as long as PCA is used for the compression. We experiment with discriminative dimensionality reduction learned on an automatically collected large collection of pairs of photos depicting the same object (around 900K pairs). When trained on such a dataset, the discriminative dimensionality reduction performs substantially better than PCA and achieves high retrieval accuracy for very short codes (e.g. 0.368 mAP on Oxford Buildings for 16-dimensional features).

2 Related Work

Our paper was inspired by the strong performance of convolutional neural networks (CNN) in image classification tasks, and the qualitative evidence of their feasibility for image retrieval provided in [10]. A subsequent report [4] demonstrated that features emerging within the top layers of large deep CNNs can be reused for classification tasks dissimilar from the original classification task. Convolutional networks have also been used to produce descriptors suitable for retrieval within the *siamese architectures* [3].

In the domain of "shallow" architectures, there is a line of works on applying the responses of discriminatively trained multiclass classifiers as descriptors within retrieval applications. Thus, [24] uses the output of classifiers trained to predict membership of Flickr groups as image descriptors. Likewise, very compact descriptors based on the output of binary classifiers trained for a large number of classes (*classemes*) were proposed in [23]. Several works such as [11] used the outputs of discriminatively trained classifiers to describe human faces, obtaining high-performing face descriptors.

The current state-of-the-art holistic image descriptors are obtained by the aggregation of local gradient-based descriptors. Fisher Vectors [18] is the best known descriptor of this kind, however its performance has been recently superceded by the triangulation embedding suggested in [9] (another recent paper [22] have introduced descriptors that can also achieve very high performance, however the memory footprint of such descriptors is at least an order of magnitude larger than uncompressed Fisher vectors, which makes such descriptors unsuitable for most applications).

In [7], the dimensionality reduction of Fisher vectors is considered, and it is suggested to use Image-Net to discover discriminative low-dimensional subspace. The best performing variant of such dimensionality reduction in [7] is based on adding a hidden unit layer and a classifier output layer on top of Fisher vectors. After training on a subset of Image-Net, the low-dimensional activations of the hidden layer are used as descriptors for image retrieval. The architecture of [7] therefore is in many respects similar to those we investigate here, as it is deep (although not as multi-layered as in our case), and is trained on image-net classes. Still, the representations derived in [7] are based on hand-crafted features (SIFT and local color histograms) as opposed to neural codes derived from CNNs that are learned from the bottom up.

There is also a large body of work on dimensionality reduction and metric learning [26]. In the last part of the paper we used a variant of the discriminative dimensionality reduction similar to [21].

Independently and in parallel with our work, the use of neural codes for image retrieval (among other applications) has been investigated in [20]. Their findings are largely consistent with ours, however there is a substantial difference from this work in the way the neural codes are extracted from images. Specifically, [20] extract a large number of neural codes from each image by applying a CNN in a "jumping window" manner. In contrast to that, we focus on holistic descriptors where the whole image is mapped to a single vector, thus resulting in a substantially more compact and faster-to-compute descriptors, and we also investigate the performance of compressed holistic descriptors.

Furthermore, we investigate in details how retraining of a CNN on different datasets impact the retrieval performance of the corresponding neural codes. Another concurrent work [17] investigated how similar retraining can be used to adapt the Image-Net derived networks to smaller classification datasets.

3 Using Pretrained Neural Codes

Deep Convolutional Architecture. In this section, we evaluate the performance of neural codes obtained by passing images through a deep convolution network, trained to classify 1000 Image-Net classes [10]. In particular, we use our own reimplementation of the system from [10]. The model includes five convolutional layers, each including a convolution, a rectified linear (ReLU) transform ($f(x) = \max(x, 0)$), and a max pooling transform (layers 1, 2, and 2). At the top of the architecture are three fully connected layers ("layer 6", "layer 7", "layer 8"), which take as an input the output of the previous layer, multiply it by a matrix, and, in the case of layers 6, and 7 applies a rectified linear transform. The network is trained so that the layer 8 output corresponds to the one-hot encoding of the class label. The softmax loss is used during training. The results of the training on the ILSVRC dataset [1] closely matches the result of a single CNN reported in [10] (more precisely, the resulting accuracy is worse by 2%). Our network architecture is schematically illustrated on Figure 1.

The network is applicable to 224×224 images. Images of other dimensions are resized to 224×224 (without cropping). The CNN architecture is feed-forward, and given an image I, it produces a sequence of layer activations. We denote with $L^5(I)$, $L^6(I)$, and $L^7(I)$ the activations (output) of the corresponding layer *prior* to the ReLU transform. Naturally, each of these high-dimensional vectors represent a *deep* descriptor (a *neural code*) of the input image.

Benchmark Datasets. We evaluate the performance of neural codes on four standard datasets listed below. The results for top performing methods based on holistic descriptors (of dimensionality upto 32K) are given in Table 1.

Oxford Buildings Dataset [19] (Oxford). The dataset consists of 5062 photographs collected from Flickr and corresponding to major Oxford landmarks. Images corresponding to 11 landmarks (some having complex structure and comprising several buildings) are manually annotated. The 55 hold-out queries evenly distributed over those 11 landmarks are provided, and the performance of a retrieval method is reported as a mean average precision (mAP) [19] over the provided queries.

Oxford Buildings Dataset+100K [19] (Oxford 105K). The same dataset with the same associated protocol, but with additional 100K distractor images provided by the dataset authors.

INRIA Holidays Dataset [8] (Holidays). The dataset consists of 1491 vacation photographs corresponding to 500 groups based on same scene or object. One image from each group serves as a query. The performance is reported as mean average precision over 500 queries. Some images in the dataset are not in a natural orientation (rotated by ±90 degrees). As deep architectures that we consider are trained on the images in a normal orientation, we follow several previous works, and manually bring all images in the dataset to the normal orientation. In a sequel, all our results are for this modified dataset. We also experimented with an unrotated version and found the performance in most settings to be worse by about 0.03 mAP. Most of the performance drop can be regained back using data augmentation (rotating by ±90) on the dataset and on the query sides.

University of Kentucky Benchmark Dataset [16] (UKB). The dataset includes 10,200 indoor photographs of 2550 objects (4 photos per object). Each image is used to query the rest of the dataset. The performance is reported as the average number of same-object images within the top-4 results, and is a number between 0 and 4.

Results. The results for neural codes produced with a network trained on ILSVRC classes are given in the middle part of Table 1. All results were obtained using L2-distance on L2-normalized neural codes. We give the results corresponding to each of the layers 5, 6, 7. We have also tried the output of layer 8 (corresponding to the ILSVRC class probabilities and thus closely related to previous works that used class probabilities as descriptors), however it performed considerably worse (e.g. 0.02 mAP worse than layer 5 on Holidays).

Among all the layers, the 6th layer performs the best, however it is not uniformly better for all queries (see Figure 2 and Figure 3). Still, the results obtained using simple combination of the codes (e.g. sum or concatenation) were worse than $L^6(I)$-codes alone, and more complex non-linear combination rules we experimented with gave only marginal improvement.

Overall, the results obtained using $L^6(I)$-codes are in the same ballpark, but not superior compared to state-of-the-art. Their strong performance is however remarkable given the disparity between the ILSVRC classification task and the retrieval tasks considered here.

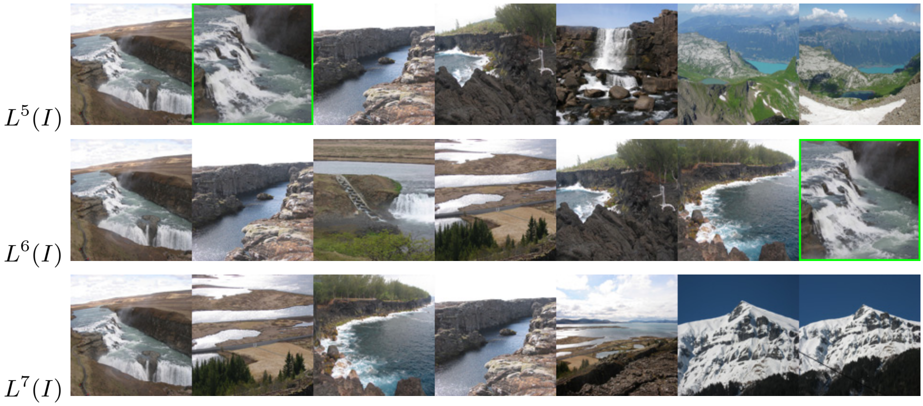

Fig. 2. A retrieval example on Holidays dataset where Layer 5 gives the best result among other layers, presumably because of its reliance on relatively low-level texture features rather than high level concepts. The left-most image in each row corresponds to the query, correct answers are outlined in green.

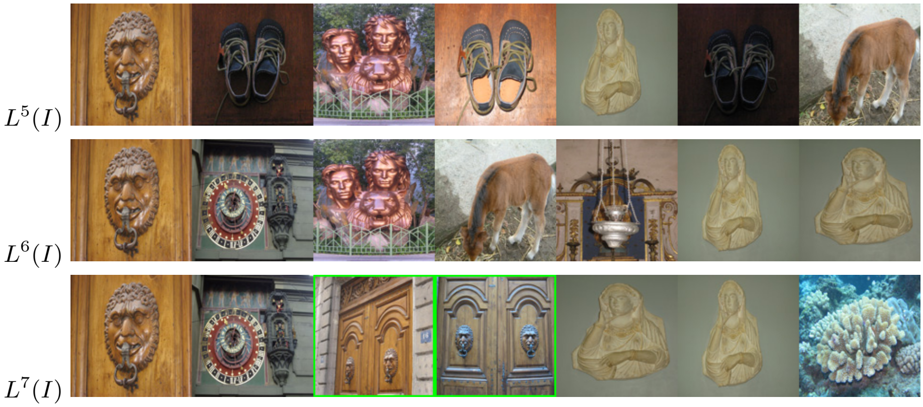

Fig. 3. A retrieval example on Holidays dataset where Layer 7 gives the best result among other layers, presumably because of its reliance on high level concepts. The left-most image in each row corresponds to the query, correct answers are outlined in green.

4 Retrained Neural Codes

A straightforward idea for the improvement of the performance of neural codes is to retrain the convolutional architecture on the dataset with image statistics and classes that are more relevant for datasets considered at test time.

The Landmarks Dataset. We first focus on collecting the dataset that is relevant to the landmark-type datasets (Holidays and Oxford Buildings). The collection of such dataset is an untrivial task, and we chose a (semi)-automated approach for that. We start by selecting 10,000 most viewed landmark Wikipedia pages (over the last month). For each page, we used the title of the page as a query to Yandex image search engine[1], and then downloaded 1000 top images returned in response to the query (or less, if the query returned less images).

At the second stage, we eyeballed the returned images by looking at the hundred of photographs from the top of the response and at an another hundred sampled uniformly from the remaining images (900 or less). We then manually classify the downloaded list into one of the following three classes: (1) "take all" (at least 80% in both hundreds are relevant, i.e. are actual photographs of the landmark), (2) "take top" (at least 80% in the first hundred are relevant, but the second hundred has more than 20% non-relevant images, including logos, maps, portraits, wrong scenes/objects), (3) "unsuitable" (more than 20% non-relevant images even within the first hundred). Overall, in this way we found 252 "take all" classes, and 420 "take top" images. Figure 4 shows two typical examples of classes in the collected dataset. We then assembled the dataset out of these classes, taking either top 1000 images (for "take all" classes) or top 100 images (for "take top" classes) for each query. Overall the resulting dataset has 672 classes and 213,678 images. During the collection, we excluded queries related to Oxford, and we also removed few near-duplicates with the Holidays dataset from the final dataset. We provide the list of the queries and the URLs at the project webpage[2].

Our approach for a landmark dataset collection is thus different from that of [14] that uses Flickr crawling to assemble a similar dataset in a fully automatic way. The statistics of images indexed by image search engines and of geotagged user photographs is different, so it would be interesting to try the adaptation using the Flickr-crawled dataset.

We then used the collected dataset to train the CNN with the same architecture as for the ILSVRC (except for the number of output nodes that we changed to 672). We initialized our model by the original ILSVRC CNN (again except for the last layer). Otherwise, the training was the same as for the original network.

Results for Retrained Neural Codes. The results for neural codes produced with a network retrained on the landmark classes are given in Table 1. As expected, the difference with respect to the original neural codes is related to the similarity between the landmark photographs and the particular retrieval

[1] http://images.yandex.ru

[2] http://sites.skoltech.ru/compvision/projects/neuralcodes/

Table 1. Full-size holistic descriptors: comparison with state-of-the-art (holistic descriptors with the dimensionality up to 32K). The neural codes are competitive with the state-of-the-art and benefit considerably from retraining on related datasets (Landmarks for Oxford Buildings and Holidays; turntable sequences for UKB). $\star$ indicate the results obtained for the rotated version of Holidays, where all images are set into their natural orientation.

Descriptor	Dims	Oxford	Oxford 105K	Holidays	UKB
Fisher+color[7]	4096	—	—	**0.774**	3.19
VLAD+adapt+innorm[2]	32768	0.555	—	0.646	—
Sparse-coded features[6]	11024	—	—	0.767	**3.76**
Triangulation embedding[9]	8064	**0.676**	**0.611**	0.771	3.53
Neural codes trained on ILSVRC					
Layer 5	9216	0.389	—	0.690*	3.09
Layer 6	4096	0.435	0.392	0.749*	3.43
Layer 7	4096	0.430	—	0.736*	3.39
After retraining on the Landmarks dataset					
Layer 5	9216	0.387	—	0.674*	2.99
Layer 6	4096	0.545	0.512	**0.793***	3.29
Layer 7	4096	0.538	—	0.764*	3.19
After retraining on turntable views (Multi-view RGB-D)					
Layer 5	9216	0.348	—	0.682*	3.13
Layer 6	4096	0.393	0.351	0.754*	3.56
Layer 7	4096	0.362	—	0.730*	3.53

dataset. Thus, there is a very big improvement for Oxford and Oxford 105K datasets, which are also based on landmark photographs. The improvement for the Holidays dataset is smaller but still very considerable. The performance of adapted $L^6(I)$ features on the Holidays dataset is better then for previously published systems based on holistic features (unless much higher dimensionality as in [22] is considered). Representative retrieval examples comparing the results obtained with the original and the retrained neural codes are presented in Figure 5. We also tried to train a CNN on the landmarks dataset with random initialization (i.e. trained from scratch) but observed poor performance due to a smaller number of training images and a higher ratio of irrelevant images compared to ILSVRC.

Interestingly, while we obtain an improvement by retraining the CNN on the Landmarks dataset, no improvement over the original neural codes was obtained by retraining the CNN on the SUN dataset [25]. Apparently, this is because each SUN class still correspond to *different* scenes with the same usage type, while each class in the Landmark dataset as well as in the Holidays and Oxford datasets corresponds to the same object (e.g. building).

Adaptation on the Turntable Sequences. After retraining on the Landmarks collection, the performance on the UKB dataset drops. This reflects the fact that the classes in the UKB dataset, which correspond to multiple indoor views of different small objects, are more similar to some classes within ILSVRC

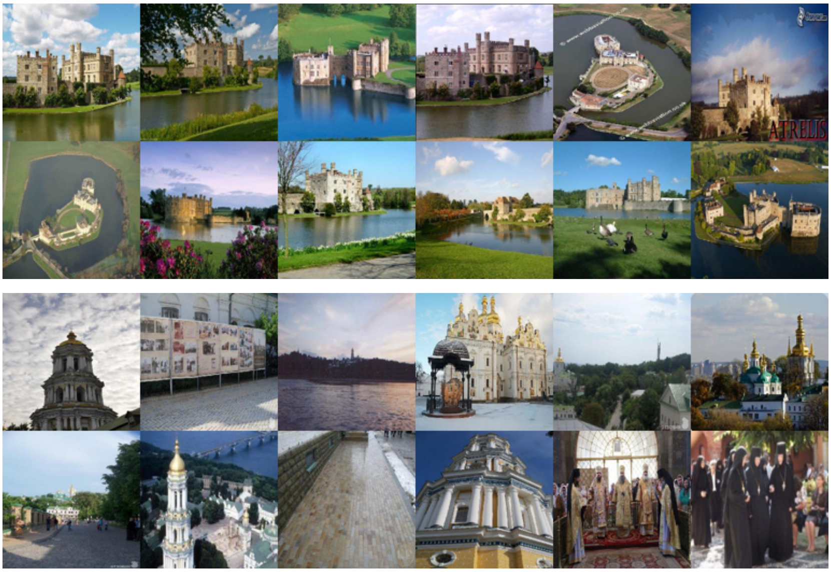

Fig. 4. Sample images from the "Leeds Castle" and "Kiev Pechersk Lavra" classes of the collected Landmarks dataset. The first class contains mostly “clean” outdoor images sharing the same building while the second class contains a lot of indoor photographs that do not share common geometry with the outdoor photos.

than to landmark photographs. To confirm this, we performed the second retraining experiment, where we used the Multi-view RGB-D dataset [12] which contains turntable views of 300 household objects. We treat each object as a separate class and sample 200 images per class. We retrain the network (again, initialized by the ILSVRC CNN) on this dataset of 60,000 images (the depth channel was discarded). Once again, we observed (Table 1) that this retraining provides an increase in the retrieval performance on the related dataset, as the accuracy on the UKB increased from 3.43 to 3.56. The performance on the unrelated datasets (Oxford, Oxford-105K) dropped.

5 Compressed Neural Codes

As the neural codes in our experiments are high-dimensional (e.g. 4096 for $L^6(I)$), albeit less high-dimensional than other state-of-the-art holistic descriptors, a question of their efficient compression arises. In this section, we evaluate two different strategies for such compression. First, we investigate how efficiency of neural codes degrades with the common PCA-based compression. An important finding is that this degradation is rather graceful. Second, we assess a more sophisticated procedure based on discriminative dimensionality reduction. We focus our evaluation on $L^6(I)$, since the performance of the neural codes

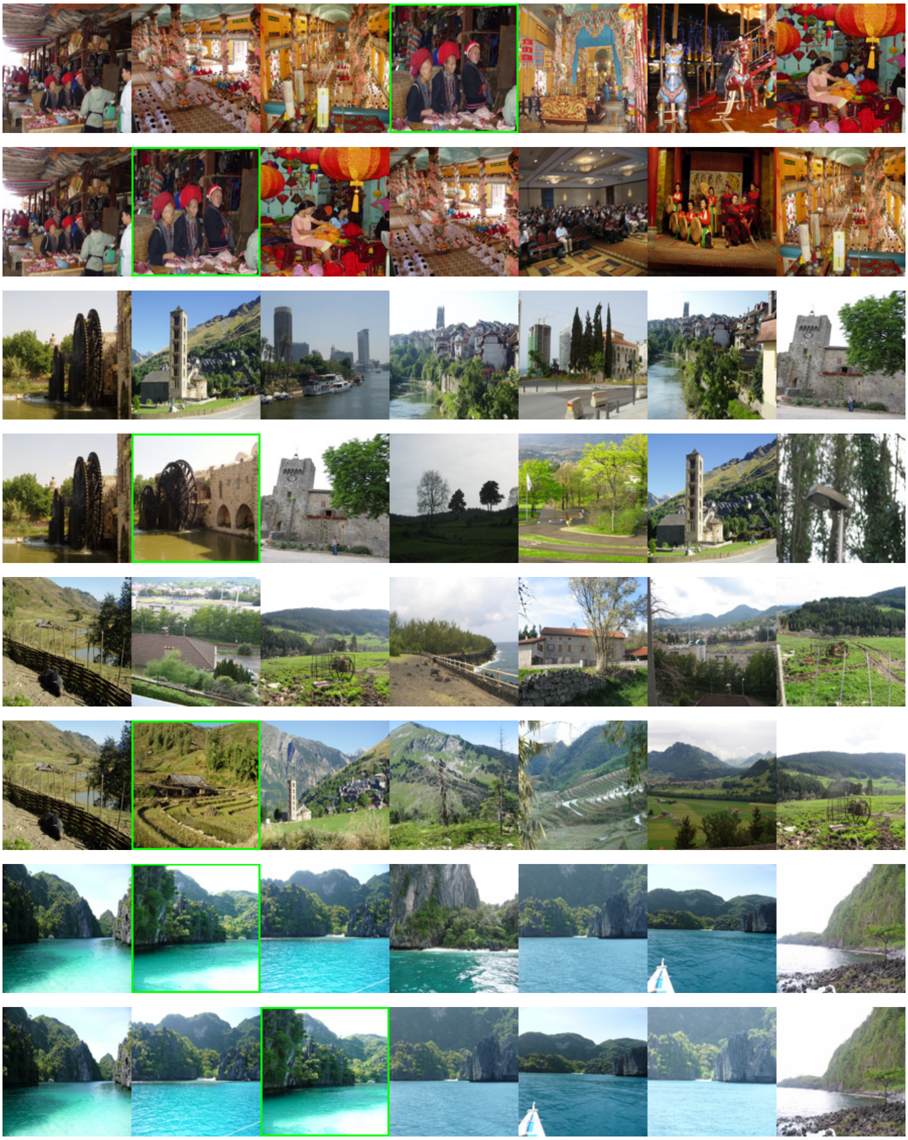

Fig. 5. Examples of Holidays queries with large differences between the results of the original and the retrained neural codes (retraining on Landmarks). In each row pair, the left-most images correspond to the query, the top row corresponds to the result with the original neural code, the bottom row corresponds to the retrained neural code. For most queries, the adaptation by retraining is helpful. The bottom row shows a rare exception.

Table 2. The performance of neural codes (original and retrained) for different PCA-compression rates. The performance of the descriptors is almost unaffected till the dimensionality of 256 and the degradation associated with more extreme compression is graceful.

Dimensions	16	32	64	128	256	512
Oxford						
Layer 6	0.328	0.390	0.421	0.433	0.435	0.435
Layer 6 + landmark retraining	0.418	0.515	0.548	0.557	0.557	0.557
Layer 6 + turntable retraining	0.289	0.349	0.377	0.391	0.392	0.393
Oxford 105K						
Layer 6	0.260	0.330	0.370	0.388	0.392	0.392
Layer 6 + landmark retraining	0.354	0.467	0.508	0.523	0.524	0.522
Layer 6 + turntable retraining	0.223	0.293	0.331	0.348	0.350	0.351
Holidays						
Layer 6	0.591	0.683	0.729	0.747	0.749	0.749
Layer 6 + landmark retraining	0.609	0.729	0.777	0.789	0.789	0.789
Layer 6 + turntable retraining	0.587	0.702	0.741	0.756	0.756	0.756
UKB						
Layer 6	2.630	3.130	3.381	3.416	3.423	3.426
Layer 6 + landmark retraining	2.410	2.980	3.256	3.297	3.298	3.300
Layer 6 + turntable retraining	2.829	3.302	3.526	3.552	3.556	3.557

associated with the sixth layer was consistently better than with the codes from other layers.

PCA Compression. We first evaluate the performance of different versions of neural codes after PCA compression to a different number of dimensions (Table 2). Here, PCA training was performed on 100,000 random images from the Landmark dataset.

The quality of neural codes $L^6(I)$ for different PCA compression rates is presented in Table 2. Overall, PCA works surprisingly well. Thus, the neural codes can be compressed to 256 or even to 128 dimensions almost without any quality loss. The advantage of the retrained codes persists through all compression rates. Table 3 further compares different holistic descriptors compressed to 128-dimensions, as this dimensionality has been chosen for comparison in several previous works. For Oxford and Holidays datasets, the landmark-retrained neural codes provide a new state-of-the-art among the low-dimensional global descriptors.

Discriminative Dimensionality Reduction. In this section, we further perform discriminative dimensionality reduction via the learning of a low-rank projection matrix W. The objective of the learning is to make distances between codes small in the cases when the corresponding images contain the same object and large otherwise, thus achieving additional tolerance for nuisance factors, such as viewpoint changes. For such learning, we collected a number of image pairs which contained the same object. Again, the challenge here was to collect a diverse set of pairs.

Table 3. The comparison of the PCA-compressed neural codes (128 dimensions) with the state-of-the-art holistic image descriptors of the same dimensionality. The PCA-compressed landmark-retrained neural codes establish new state-of-the-art on Holidays, Oxford, and Oxford 105K datasets.

Descriptor	Oxford	Oxford 105K	Holidays	UKB
Fisher+color[7]	—	—	0.723	3.08
VLAD+adapt+innorm[2]	0.448	0.374	0.625	—
Sparse-coded features[6]	—	—	0.727	**3.67**
Triangulation embedding[9]	0.433	0.353	0.617	3.40
Neural codes trained on ILSVRC				
Layer 6	0.433	0.386	0.747*	3.42
After retraining on the Landmarks dataset				
Layer 6	**0.557**	**0.523**	**0.789***	3.29
After retraining on turntable views (Multi-view RGB-D)				
Layer 6	0.391	0.348	0.756*	3.55

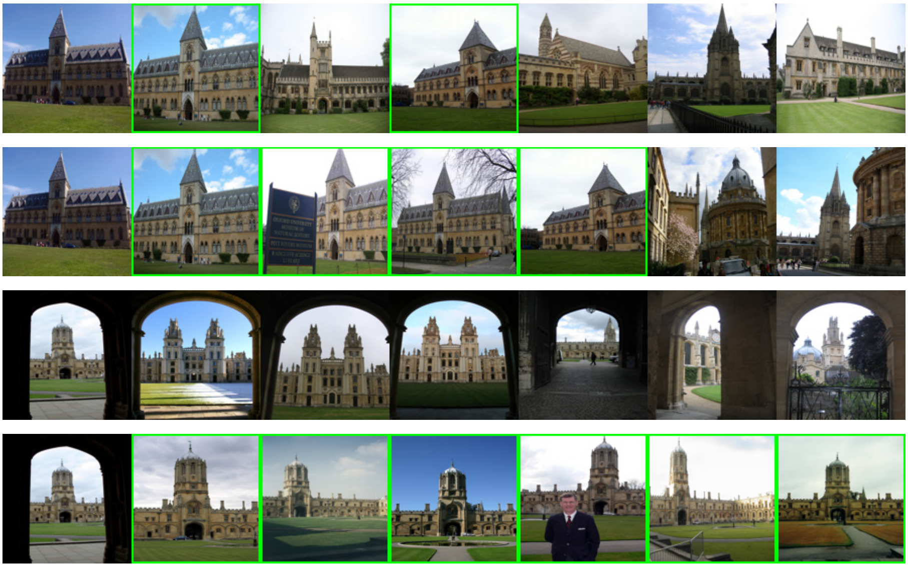

Fig. 6. Examples of queries with large differences between the results of the PCA-compressed and the discriminatively-compressed neural codes (for 32 dimensions). The correct answers are outlined in green.

To obtain such a dataset, we sample pairs of images within the same classes of the Landmark dataset. We built a matching graph using a standard image matching pipeline (SIFT+nearest neighbor matching with the second-best neighbor test [15] + RANSAC validation [5]). The pipeline is applied to all pairs of images belonging to the same landmark. Once the graph for the landmark is constructed, we took pairs of photographs that share at least one neighbor in

Table 4. The comparison of the performances of the PCA compression and a discriminative dimensionality reduction for the original neural codes on the Oxford dataset. Discriminative dimensionality reduction improves over the PCA reduction, in particular for the extreme dimensionality reduction.

D =	16	32	64	128
PCA-compression	0.328	0.390	0.421	0.433
Discriminative dimensionality reduction	0.368	0.401	0.430	0.439

the graph but are not neighbors themselves (to ensure that we do not focus the training process on near duplicates). Via such procedure we obtain 900K diverse image pairs (Figure 7). We further greedily select a subset of $100K$ pairs so that each photograph occurs at most once in such a subset, and use this subset of pairs for training.

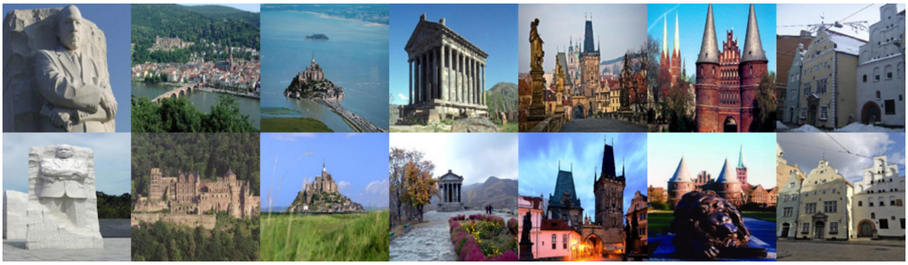

Fig. 7. Examples of training pairs for discriminative dimensionality reduction. The pairs were obtained through time-consuming RANSAC matching of local features and simple analysis of the resulting match graph (see the text for more details).

Given a dataset of matching pairs we learn a linear projection matrix W via the method from [21]. In the experiments with large compression rates ($D = 16, 32$) we project original 4096-dimensional codes. For the dimensionality $D =$ 64, 128, we observed significantn overfitting due to a large number of parameters within W. In this case we first performed PCA-compression to 1024 dimensions and then learned W for the preliminarily compressed 1024-dimensional codes.

The results of the two compression strategies (PCA and the discriminative reduction) are compared for non-retrained codes for the Oxford dataset in Table 4. As can be seen, the biggest gain from discriminative dimensionality reduction is achieved for the extremely compressed 16-dimensional codes. We have also evaluated the discriminative dimensionality reduction on the neural codes retrained on the Landmarks dataset. In this case, however, we did not observed any additional improvement from the discriminative reduction, presumably because the network retraining and the discriminative reduction were performed using overlapping training data.

6 Discussion

We have evaluated the performance of the deep neural codes within the image retrieval application. There are several conclusions and observations that one can draw from our experiments.

First of all, as was expected, neural codes perform well, even when one uses the CNN trained for the classification task and when the training dataset and the retrieval dataset are quite different from each other. Unsurprisingly, this performance can be further improved, when the CNN is retrained on photographs that are more related to the retrieval dataset.

We note that there is an obvious room for improvement in terms of the retrieval accuracy, in that all images are downsampled to low resolution (224×224) and therefore a lot of information about the texture, which can be quite discriminative, is lost. As an indication of potential improvement, our experiments with Fisher Vectors suggest that their drop in performance under similar circumstances is about 0.03 mAP on Holidays.

Interestingly, and perhaps unexpectedly, the best performance is observed not on the very top of the network, but rather at the layer that is two levels below the outputs. This effect persists even after the CNN is retrained on related images. We speculate, that this is because the very top layers are too much tuned for the classification task, while the bottom layers do not acquire enough invariance to nuisance factors.

We also investigate the performance of compressed neural codes, where plain PCA or a combination of PCA with discriminative dimensionality reduction result in very short codes with very good (state-of-the-art) performance. An important result is that PCA affects performance of neural codes much less than the one of VLADs, Fisher Vectors, or triangulation embedding. One possible explanation is that passing an image through the network discards much of the information that is irrelevant for classification (and for retrieval). Thus, CNN-based neural codes from deeper layers retain less (useless) information than unsupervised aggregation-based representations. Therefore PCA compression works better for neural codes.

One possible interesting direction for investigation, is whether good neural codes can be obtained directly by training the whole deep architecture using the pairs of matched images (rather than using the classification performance as the training objective), e.g. using siamese architecture of [3]. Automated collection of a suitable training collection having sufficient diversity would be an interesting task on its own. Finally, we note that the dimensionality reduction to a required dimensionality can be realized by choosing the size of a network layer used to produce the codes, rather than a post-hoc procedure.

References

1. Berg, A., Deng, J., Fei-Fei, L.: Large scale visual recognition challenge, ILSVRC (2010), `http://www.image-net.org/challenges/LSVRC/2010/`
2. Arandjelović, R., Zisserman, A.: All about VLAD. In: Computer Vision and Pattern Recognition (2013)
3. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: Computer Vision and Pattern Recognition (2005)
4. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. CoRR abs/1310.1531 (2013)
5. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM (1981)
6. Ge, T., Ke, Q., Sun, J.: Sparse-coded features for image retrieval. In: British Machine Vision Conference (2013)
7. Gordo, A., Rodríguez-Serrano, J.A., Perronnin, F., Valveny, E.: Leveraging category-level labels for instance-level image retrieval. In: Computer Vision and Pattern Recognition (2012)
8. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
9. Jégou, H., Zisserman, A.: Triangulation embedding and democratic aggregation for image search. In: Computer Vision and Pattern Recognition (2014)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Neural Information Processing Systems (2012)
11. Kumar, N., Berg, A.C., Belhumeur, P.N., Nayar, S.K.: Attribute and simile classifiers for face verification. In: International Conference on Computer Vision, pp. 365–372 (2009)
12. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset. In: Neural Information Processing Systems (2011)
13. LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E., Jackel, L.D.: Handwritten digit recognition with a back-propagation network. In: Neural Information Processing Systems, pp. 396–404 (1989)
14. Li, Y., Crandall, D., Huttenlocher, D.: Landmark classification in large-scale image collections. In: International Conference on Computer Vision (2009)
15. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (2004)
16. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: Computer Vision and Pattern Recognition (2006)
17. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: Computer Vision and Pattern Recognition (June 2014)
18. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010)
19. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Computer Vision and Pattern Recognition (2007)

20. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. CoRR (2014)
21. Simonyan, K., Parkhi, O.M., Vedaldi, A., Zisserman, A.: Fisher Vector Faces in the Wild. In: British Machine Vision Conference (2013)
22. Tolias, G., Avrithis, Y., Jégou, H.: To aggregate or not to aggregate: selective match kernels for image search. In: International Conference on Computer Vision (2013)
23. Torresani, L., Szummer, M., Fitzgibbon, A.: Efficient object category recognition using classemes. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 776–789. Springer, Heidelberg (2010)
24. Wang, G., Hoiem, D., Forsyth, D.A.: Learning image similarity from flickr groups using stochastic intersection kernel machines. In: International Conference on Computer Vision (2009)
25. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: Computer Vision and Pattern Recognition (2010)
26. Yang, L., Jin, R.: Distance metric learning: A comprehensive survey, vol. 2. Michigan State Universiy (2006)