CENTER FOR
MACHINE PERCEPTION

CZECH TECHNICAL
UNIVERSITY IN PRAGUE

DOCTORAL THESIS

ISSN 1213-2365

Visual Retrieval with Compact Image Representations

A Doctoral Thesis presented to the Faculty of the Electrical Engineering of the Czech Technical University in Prague in fulfillment of the requirements for the Ph.D. Degree in Study Programme No. P2612 - Electrical Engineering and Information Technology, Branch No. 3902V035 - Articial Intelligence and Biocybernetics, by

Filip Radenović

filip.radenovic@cmp.felk.cvut.cz

Thesis Advisor
Doc. Mgr. Ondřej Chum, Ph.D.

CTU–CMP–2019–01

May, 2019

Available at
ftp://cmp.felk.cvut.cz/pub/cmp/articles/radenovic/Radenovic-TR-2019-01.pdf

The author was supported by the MSMT LL1303 ERC-CZ, GAČR 19-23165S, and SGS17/185/OHK3/3T/13 grants.

Research Reports of CMP, Czech Technical University in Prague, No. 1, 2019

Published by

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University in Prague
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

Visual Retrieval with Compact Image Representations

Filip Radenović

May, 2019

I owe my deepest gratitude to Ondřej Chum, my advisor, without whom this thesis would not have been possible. He has transferred his passion, enthusiasm, and devotion towards computer vision onto me. I cannot thank him enough for his support, ideas and motivation, and for sharing his knowledge and expertise throughout my research. Many thanks to Andrej Mikulík for providing guidance at the beginning of my PhD. I also wish to express my sincere gratitude to Giorgos Tolias for numerous long discussions, for sharing his valuable comments and ideas, and for helping me through my PhD studies.

I am very thankful I had the chance to work in the Visual Recognition Group within the Center for Machine Perception, and especially I must thank Prof. Jiří Matas for doing a great job at leading such a high quality group and maintaining the workplace with an amazing energy. My PhD days were full of joy and happiness because of all the colleagues and friends I have met throughout the past 6 years, and for that I am eternally grateful.

Most importantly, I want to thank my family and friends for their unconditional love and support that made it possible for me to pursue my dreams. I dedicate this thesis to two of the most recent additions to my family, my clever little nephew Grigorije and my beautiful little niece Simona.

"There is, indeed, nothing more annoying than to be, for instance, fairly rich, of good family, of pleasing appearance, fairly well educated, not stupid, rather kind-hearted even, and at the same time to possess no talent, no special quality, no eccentricity even, not a single idea of one's own, to be precisely *like everyone else*."

– Fyodor Mikhailovich Dostoevsky, The Idiot

Abstract

This thesis addresses the problem of visual retrieval in large-scale image datasets, where the goal is to find all images of an object instance. The object is specified by a query image, which can be a photograph, painting, edge-map, human-drawn sketch *etc.* Solutions to this problem can be widely used in many applications such as place or location recognition, copyright violation detection, product search, 3D reconstruction, *etc.* The task of visual retrieval of an object instance is a challenging one, as the representation of the object appearance has to handle: significant viewpoint, scale, and illumination change; heavy occlusions; and, different image modalities (photograph, painting, cartoon, sketch). At the same time, the search has to be performed online, *i.e.*, when a user submits the query, the response should be immediate, even when searching through millions of images. Towards this goal, we propose methods for compact image representation, that achieve high accuracy, while maintaining low memory and computational requirements.

A number of image retrieval related problems is stated, studied and resolved in the theses. Two conceptually different approaches to compact image representation are proposed. First, a method of joint dimensionality reduction of multiple vocabularies for bag-of-words-based compact representation is proposed. Second, a method to fine-tune convolutional neural networks (CNNs) for compact image retrieval from a large collection of unordered images in a fully automated manner is proposed. We additionally show that the CNN trained with edge maps of landmark images, instead of photographs, improves performance in the cases where shape is carrying the dominant information. The proposed compact representations are evaluated on a range of different tasks, providing improvements on challenging cases of instance image retrieval, generic sketch-based image retrieval or its fine-grained counterpart, and domain generalization.

We address the issue of image retrieval benchmarking. We extend standard and popular Oxford Buildings and Paris datasets by novel annotations, protocols, and queries. The novel protocols allow fair comparison between different methods, including those using a dataset pre-processing stage. An extensive comparison of the state-of-the-art methods is performed on the new benchmark. The results show that image retrieval is far from being solved.

Finally, we introduce the concept of target mismatch attack for deep learning based retrieval systems to generate an adversarial image to conceal the query image. The adversarial image looks nothing like the user intended query, but leads to identical or very similar retrieval results. We evaluate the attacks on standard retrieval benchmarks and compare the results retrieved with the original and adversarial image.

Abstrakt

Tématem této práce je vyhledávání v rozsáhlých kolekcích obrázků pomocí obrazové informace s cílem najít všechny obrázky zobrazující konkrétní objekt. Hledaný objekt, tzv. dotaz, je také definován obrázkem, což může být fotografie, malba, hranový obrázek, náčrt, atd. Řešení tohoto problému má široké uplatnění v mnoha aplikacích jako je rozpoznání místa nebo polohy kamery, detekce porušení autorských práv, vyhledávání produktů nebo 3D rekonstrukce. Úloha vizuálního vyhledávání konkrétního objektu je náročná, protože reprezentace vzhledu objektu musí brát v úvahu: podstatnou změnu úhlu pohledu, meřítka či osvělení; podstatné zakrytí objektu; a různé formy vyobrazení (fotografie, malba, kresba, náčrt). Dalším požadavkem je, že vyhledávání musí běžet v reálném čase. V okamžiku, kdy uživatel odešle dotaz na vyhledání, musí okamžitě dostat výsledek, i pokud jsou prohledávány miliony obrázků. Abychom toho docílili, navrhujeme metody pro kompaktní reprezentace obrázků, které dosahují vysoké přesnosti a současně mají nízké paměťové a výpočetní nároky.

Řada problémů souvisejících s vizuálním vyhledáváním je formulována, studována a vyřešena v této práci. Jsou navženy dva konceptuálně odlišné přístupy ke kompaktním reprezentacím obrázků. První je metoda pro současné snížení dimenze více slovníků pro kompaktní reprezentace založené na metodě vizuálních slov. Druhým přístupem je plně automatické dotrénování konvolučních neuronových sítí pro kompaktní reprezentaci obrázků a následné vyhledávání v rozsáhlých neuspořádaných kolekcích. Dále ukazujeme, že konvoluční neuronové sítě trénované s hranovými obrázky budov namísto jejich fotografií dosahují lepších výsledků v případech, kde je tvar dominující informací v obrázku. Tyto kompaktní reprezentace jsou vyhodnoceny na různých úlohách a poskytují zlepšení u náročných dotazů vizuálního vyhledávání, vyhledávání kreseb v případě obecných i konkrétních objektů, a u zobecnění domény dat.

Problém testování přesnosti vizuálního vyhledávání je také dotčen. Rozšiřujeme standardní a oblíbené datasety Oxford Buildings a Paris o nové anotace, vyhodnocovací protokoly a vizuální dotazy. Nové vyhodnocovací protokoly nám umožňují spravedlivé porovnání různých metod, včetně těch, které kolekce obrázků předzpracovávají před samotným vyhledáváním. Pomocí těchto protokolů je provedeno rozsáhlé porovnání nejmodernějších state-of-the-art metod. Výsledky ukazují, že problém vizualního vyhledávání ještě není vyřešený.

Nakonec představujeme koncept útoku na záměnu cíle pro vizuální vyhledávání založeném na popisech z hlubokých sítí tak, že vytváříme zástupný obrázek zakrývající obsah původního dotazu.

Contents

Chapter 1

Introduction

Computer vision is a research area dedicated to developing methods that can automatically perform tasks that the *human visual system* can do. In a similar way to humans who learn to understand their environment by processing the input they receive from the eyes, given digital images and/or videos from cameras as the input, computer vision develops methods that provide a high-level understanding of the world. In order to achieve optimal performance, one would need an extremely large amount of images to teach and develop such automated systems. Due to a recent boom in photo-sharing websites, *e.g.* Flickr[1], Facebook[2], and Instagram[3], there are millions, possibly even billions, of new images and videos appearing on the Internet every day. Thanks to such a large amount of available content, a very fast development of computer vision has been accomplished. Many recently developed state-of-the-art computer vision methods have become a crucial part of the commercial systems. For example, tasks that were once dependent on human interaction, are now, with the help of computer vision, performed better by a machine, *e.g.*, electronic toll collector with a plate recognition system, self-driving car with a full visual system, video assistant referee in football, Hawk-Eye system in various sports such as tennis, badminton, volleyball, *etc.* On the other hand, systems that are fully machine-operated are more vulnerable to cyberattacks.

[1] www.flickr.com
[2] www.facebook.com
[3] www.instagram.com

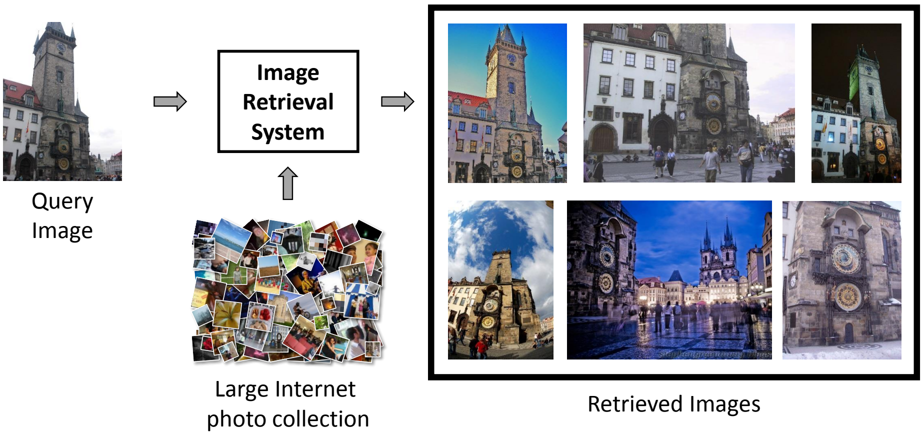


Figure 1.1. Simple illustration of the instance image retrieval task.

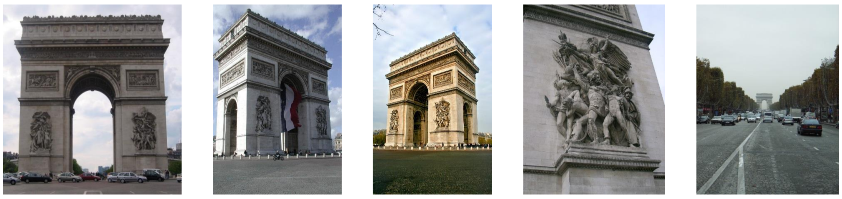


Figure 1.2. Example of the same object with significant viewpoint and/or scale changes.

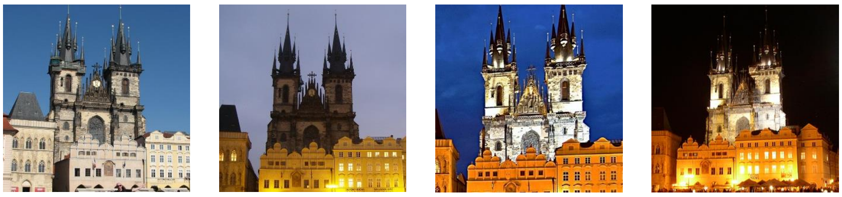


Figure 1.3. Example of the same object photographed under significant illumination changes.

Nowadays, there is a vast variety of computer vision problems involving images, videos, or both. The most popular ones are image classification, image retrieval, object detection, video tracking, 3D reconstruction, *etc.* This thesis is mainly focusing on the problem of instance image retrieval, that is formulated in the following section.

1.1. Addressed challenges

Instance image retrieval (or particular object image retrieval) is a task in which a query image with an object of interest is given as an input. The goal is to find the object depicted in the query image in a large unordered collection of images. Ideally, the image retrieval system should return all images from the collection that contain the object in question, see Figure 1.1. Unlike in image classification or object detection, where the object of interest can only be from a predefined set of classes, there are no assumptions imposed in the instance image retrieval, *i.e.*, the set of all possible different object instances is arbitrarily large. Hence, the solutions developed for image classification and object detection tasks are not directly applicable here. As an example, if a user queries with an image of a particular landmark, such as *Prague Astronomical Clock*, the results should only contain images of that specific building, as in Figure 1.1.

Nowadays, there are commercial products to search vast number of images that appear on the Internet, *e.g.*, Google Image Search[4], Bing Image Feed[5], TinEye[6], *etc.* Popular e-commerce websites, such as Amazon[7] and eBay[8], integrated instance image retrieval systems to help users better find their favorite products. Even so, instance image retrieval is an open problem due to a high variation in the visual appearance of the same object. The desired representation of the image has to be robust enough to deal with the following challenges:

[4] images.google.com

[5] www.bing.com/images

[6] www.tineye.com

[7] www.amazon.com

[8] www.ebay.com

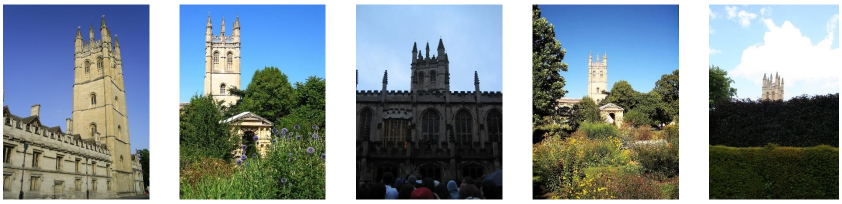


Figure 1.4. Example of the same object covered with different levels of occlusion.

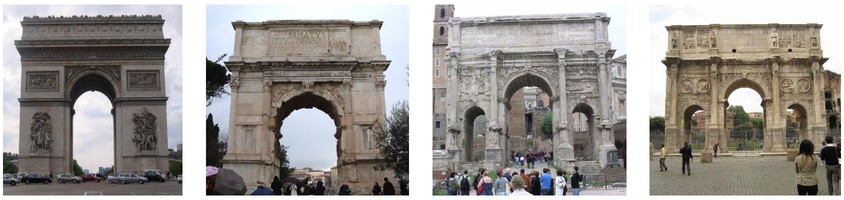


Figure 1.5. Example of visually similar but different object instances.

- Significant viewpoint and/or scale change is caused by the fact that the same object can be photographed from a variety of locations, see Figure 1.2.

- Significant illumination change is caused by different lighting conditions, *e.g.* day/night, or seasonal change, see Figure 1.3.

- Severe occlusions that cover some or most of the object of interest happen because of a cluttered environment, see Figure 1.4.

- Visually similar but different objects should not be retrieved together, see Figure 1.5.

- Same object instance can be depicted in different image modalities, such as, photograph, painting, cartoon drawing, free-hand sketch drawing, *etc.*, see Figure 1.6.

At the same time, the retrieval system has to be able to handle billions of images, from the memory requirement, processing time, and search time point of views. First successful approach to deal with these challenges was proposed by Sivic and Zisserman [151], details of which are given in the related work in Chapter 2, Section 2.1.1. The success of this method, denoted as bag-of-words (BoW), stems from hand-crafted local features [98, 96, 121] and descriptors [92, 4], that are specifically designed to deal with mentioned challenges: viewpoint and scale change are handled by the affine covariant local features and invariant descriptors; illumination invariance is treated by color-normalization of the feature descriptors; occlusion is handled with the locality of the features and geometric verification [125]; finally, similar but different objects are disambiguated by the discriminability of local features and geometric verification. To handle image retrieval at scale, local descriptors are quantized to visual words, which are efficiently matched by using an inverted file data structure [151]. In return, quantization process sacrifices discriminability to some extent. Still, BoW approach can only handle few million images on the single machine, so a compact BoW image representation was proposed [68]. In this manuscript we improve the compact BoW-based image representation by creating high-variable multiple vocabularies to reduce the quantization artifacts and boost the performance after the joint dimensionality reduction.

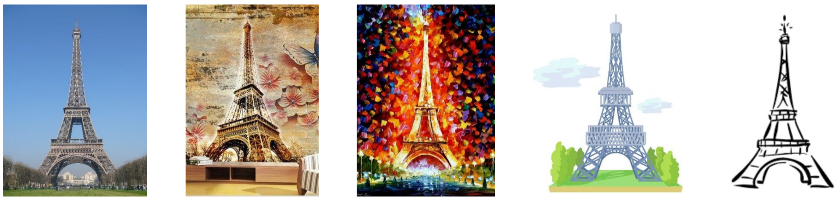


Figure 1.6. Example of the same object depicted in different image modalities.

Another popular approach to produce compact image representations for retrieval is based on convolutional neural networks (CNNs). First attempts simply utilized the activations of CNNs trained for image classification task [55, 8, 76, 138, 168, 191]. Although CNN representation did show certain generalization abilities, the performance of these methods suffered due to the task shift between image classification and retrieval. For example, in a training set for classification there is a *building* class, thus pushing a CNN to embed images of all buildings in the same part of the image representation space. In instance image retrieval, we want to be able to disambiguate particular buildings, *i.e.* we want to embed images of particular buildings close to each other in the representation space, but sufficiently far from images of different buildings. Ideally, learned representation would be able to deal with the challenges depicted in Figures 1.2-1.6. To overcome this, a large annotated training set for image retrieval is required. Constructing such dataset requires man-years of manual effort. For the sake of bypassing costly human annotators, in this manuscript, we propose to utilize a sophisticated BoW pipeline in order to guide the CNN training procedure without any human interaction. As a result, the student surpassed the teacher, *i.e.*, a compact CNN-based image representation is obtained, achieving higher accuracy, faster search time, and lower memory requirement than the BoW pipeline used for automatic training set selection.

Both hand crafted local-feature-based approaches such as BoW, and CNN-based approaches tend to rely on texture or colour in the images. While this is very successful in most of the common use-case scenarios of image retrieval, there are some specific cases where it is not enough. One such case, where the same object is depicted in completely different image modalities, is given in Figure 1.6. Any method that relies heavily on the texture or colour information will surely under-perform in this problem, as the important information is carried in the shapes of the image content. To alleviate the challenge of different image modalities, we propose a descriptor that is specifically designed to capture the shape of the objects depicted in images. We show that the proposed shape descriptor can be successfully combined with a standard texture-based image descriptor to embed both natural images, as well as paintings, cartoon drawings, and sketch drawings close to each other in the image representation space. Additionally, proposed shape representation improves image retrieval, and can be successfully used for cross-modal image matching, *e.g.*, sketch-based image retrieval, in which a hand-drawn sketch of an object is used as a query to retrieve images of the same object.

By changing image pixels of a query, the outcome of an image retrieval system can be heavily affected. For example, making unrealistic image edits by changing its texture and colours, one can negatively impact the success of the system. In fact, it has been shown that even an imperceptible non-random image perturbations can be learned to mislead a CNN-based image representation. These image perturbations are popularly known as adversarial attacks, and were firstly introduced and tested on image classification by Szegedy *et al.* [158]. If the image pixels are perturbed in order for the image to be missclassified to any other (wrong) class, an attack is denoted as a non-targeted missclassification, otherwise, if the perturbed image is changed to be missclassified to a specific (target) class, then the attack is denoted as a targeted missclasification. Similarly to image classification, adversarial attacks can be performed in the domain of image retrieval too. A non-targeted attack attempts to generate an image that for a human observer carries the same visual information, while for the CNN it appears dissimilar to other images of the same object [87, 91, 192]. This way, a user protects personal images and does not allow them to be indexed for content-based search, even when the images are publicly available. Only non-targeted attacks have been studied in image retrieval, until now. In this work, we address targeted attacks aiming to retrieve images that are related to a hidden target query without explicitly revealing the query image.

Finally, evaluating an image retrieval system is a non-trivial problem by itself. It requires a separate annotation of the whole database for each possible query in the benchmark. The most popular benchmarks [125, 126] to evaluate instance image retrieval were introduced more than 10 years ago, and as such they are becoming outdated, because the annotation was done with a different idea of image retrieval limits in mind. Also, the largest annotated benchmark [125] is up to 100k images. In this thesis, we redefine large-scale image retrieval benchmarking, and perform an extensive evaluation of many state-of-the-art approaches, including ours, on a newly proposed annotation with a set of difficulty levels and more than a million images. With this, we show that image retrieval is far from being solved, and there are many challenges to be faced in the following years.

Throughout the thesis we use the term *image retrieval* for brevity, for the task that is more precisely defined as instance image retrieval or particular object image retrieval.

1.2. Contributions

The main contributions of the thesis:

- As a first contribution, Chapter 4 addresses the problem of large-scale image retrieval test dataset construction. More specifically, new annotation for Oxford and Paris datasets is generated, the evaluation protocol is updated, new more difficult queries are defined, and a new set of over a million challenging distractors is created. This work has been published in [130].

- Chapter 5 addresses the construction of a BoW-based compact image representation for large-scale image retrieval. A method of joint dimensionality reduction of multiple visual vocabularies is proposed. More precisely, a variety of vocabulary generation techniques are studied: different k-means initializations, different descriptor transformations, different measurement regions for descriptor extraction. This work has been published in [131].

- In the following contribution, a costly BoW-based approach, with many bells and whistles, is utilized to perform the training of CNNs for image retrieval, without any human intervention. This approach focuses on producing compact image representation, as well. In particular, our combined retrieval and structure-from-motion (SfM) pipeline, both using local features, is first exploited to reconstruct all 3D models in an unordered dataset. The retrieval-SfM pipeline has been developed in a collaboration with the group from the University of North Carolina, and author focused on developing the retrieval part of the pipeline. This work has been published in [146, 132]. Next, reconstructed 3D models and their SfM information is used to enforce, not only hard non-matching, but also hard-matching examples for CNN training. This is shown to enhance the derived image representation. This work has been published in [133, 135] and the contribution is described in Chapter 6, Section 6.2.

- Other contributions in the CNN-based image retrieval pipeline are further proposed. Namely, whitening learned in a supervised way is proposed. Its effect is complementary to fine-tuning and it further boosts performance. Next, a trainable pooling layer that generalizes existing popular pooling schemes for CNNs is proposed. It significantly improves the retrieval performance while preserving the same descriptor dimensionality. Improved multi-scale representation based on the same pooling is shown to increase image retrieval accuracy, as well. Finally, a novel α-weighted query expansion technique is proposed, that is more robust compared to the standard average query expansion technique widely used for compact image representations. These contributions have been described in Chapter 6, Section 6.1, and have been published in [133, 135]. We also set a new state-of-the-art result on standard image retrieval benchmarks, which is validated by the experiments in Chapter 6, Section 6.3.

- In Chapter 7, we perform large-scale image retrieval evaluation, on our newly proposed test datasets introduced in Chapter 4. As a contribution, extensive evaluation of image retrieval methods is provided, ranging from local-feature based to CNN-descriptor based approaches, including various methods of re-ranking. This work has been published in [130].

- The novel concept of target mismatch attack for CNN-based image retrieval systems is formulated in Chapter 8. It is used to generate an adversarial image to conceal the query image and protect user privacy. The adversarial image looks nothing like the user intended query, but leads to identical or very similar retrieval results. We show successful attacks to partially unknown systems, by designing various loss functions for the adversarial image construction. This work originates from [167].

- Finally, Chapter 9 deals with problems from the area of shape matching. A new CNN-based compact shape descriptor is proposed, which is shown to be highly beneficial for two problems: (i) domain generalization in the case of classification, and (ii) cross modality matching of sketches to images. Additionally, shape information is shown to be useful even in the specific cases of traditional image retrieval. This work originates from [134, 136].

Several other contributions were proposed by the author. However, they are left out of this manuscript to keep it more focused and easier to follow. A full list of authors publications is given in the following section.

1.3. Publications

This thesis build on the results previously published in the following publications. The references are ordered chronologically and under numbers used in the rest of the thesis.

[131] F. Radenovic, H Jegou, O. Chum. **Multiple Measurements and Joint Dimensionality Reduction for Large Scale Image Search with Short Vectors**. *ICMR*, 2015.

[146] J. L. Schonberger, F. Radenovic, O. Chum, J. Frahm. **From Single Image Query to Detailed 3D Reconstruction**. *CVPR*, 2015.

[132] F. Radenovic, J. L. Schonberger, D. Ji, J. Frahm, O. Chum, J. Matas. **From Dusk till Dawn: Modeling in the Dark**. *CVPR*, 2016.

[133] F. Radenovic, G. Tolias, O. Chum. **CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples**. *ECCV*, 2016.

[135] F. Radenovic, G. Tolias, O. Chum. **Fine-tuning CNN Image Retrieval with No Human Annotation**. *TPAMI*, 2018.

[130] F. Radenovic, A. Iscen, G. Tolias, Y. Avrithis, O. Chum. **Revisiting Oxford and Paris: Large-Scale Image Retrieval Benchmarking**. *CVPR*, 2018.

[134] F. Radenovic, G. Tolias, O. Chum. **Deep Shape Matching**. *ECCV*, 2018.

[136] F. Radenovic, G. Tolias, O. Chum. **Deep Shape Matching for Domain Generalization and Cross-Modal Retrieval**. Under submission, 2019.

[167] G. Tolias, F. Radenovic, O. Chum. **Query with a Flower to Retrieve the Tower: Adversarial Attack to Conceal the Query Image**. Under submission, 2019.

The following publications were not included in the thesis, in order to keep the thesis more focused and easier to follow:

[103] A. Mikulik, F. Radenovic, O. Chum, J. Matas. **Efficient Image Detail Mining**. *ACCV*, 2014.

[17] M. Cadik, J. Vasicek, Hradis M., F. Radenovic, O. Chum. **Camera Elevation Estimation from a Single Mountain Landscape Photograph**. *BMVC*, 2015.

[144] T. Sattler, M. Havlena, F. Radenovic, K. Schindler, M. Pollefeys. **Hyperpoints and Fine Vocabularies for Large-Scale Location Recognition**. *ICCV*, 2015.

[104] A. Mishchuk, D. Mishkin, F. Radenovic, J. Matas. **Working hard to know your neighbor's margins: Local descriptor learning loss**. *NIPS*, 2017.

[105] D. Mishkin, F. Radenovic, J. Matas. **Repeatability Is Not Enough: Learning Affine Regions via Discriminability**. *ECCV*, 2018.

1.4. Structure of the thesis

This thesis is organized as follows. Related work and state-of-the-art approaches in related areas are described in Chapter 2. Standard test datasets and their evaluation protocols are introduced in detail in Chapter 3, while our newly proposed large-scale image retrieval test datasets, with over million images, are described in Chapter 4. Chapters 5 and 6 present our proposed contributions for the image retrieval with compact codes, using both hand-crafted and CNN-based features. An extensive evaluation of many state-of-the-art image retrieval approaches, including our own, is performed on the newly proposed test datasets in Chapter 7. Non-targeted adversarial attack on image retrieval to conceal the query image is formulated and tackled in Chapter 8. Chapter 9 describes our contributions for shape matching via the training of CNNs. Finally, Chapter 10 gives closing discussions and concluding remarks.

1.5. Authorship

I hereby certify that the results presented in this thesis were achieved during my own research, in cooperation with my thesis advisor Ondřej Chum, published in [131, 146, 132, 133, 135, 130, 134, 136, 167], with Giorgos Tolias, published in [133, 135, 130, 134, 136, 167], with Johannes L. Schönberger and Jan-Michael Frahm, published in [146, 132], with Hervé Jégou, published in [131], with Dinghuang Ji and Jiří Matas, published in [132], and with Ahmet Iscen and Yannis Avrithis, published in [130].

Chapter 2

Related Work

THIS chapter provides an overview of related work to the main contributions of the thesis. In Section 2.1 we describe all relevant methods for standard image retrieval task, *i.e.*, when a query is given by a user defined image or image region. Next, Section 2.2 provides overview of methods focused on the sketch-based image retrieval task, in which a query is given as a human-drawn *sketch*. Finally, domain generalization related work is given in Section 2.3.

2.1. Image retrieval

Image retrieval is a task where given a query the system has to retrieve related images from a large unordered collection. First successful retrieval methods utilize local features and bag-of-words image representation [151, 113], and are further improved by spatial verification [125], Hamming embedding [69], selective match kernel [163], and query expansion [31]. Further attempts are mostly focused on creating compact image representations, which started with compact aggregation of local descriptors [72], or extreme dimensionality reduction of sparse bag-of-word vectors [68]. Nowadays, the best performing compact image representations for retrieval are based on convolutional neural networks (CNNs). The earliest CNN-based image retrieval approaches use the network trained on ImageNet [140] and the activations of its fully-connected layer as the global image representations [9, 55]. Subsequent attempts show that focusing on the activations of convolutional layers provide much stronger image descriptors [138].

In the following, we split work related to image retrieval into two main parts. Namely, the first one in Section 2.1.1 describes approaches utilizing the local features extracted from images, and the second in Section 2.1.2, describes CNN-based approaches.

2.1.1. Local-feature-based methods

Image representation. Typical pipelines start with local feature detection and descriptor extraction. For each image in the dataset, regions of interest are detected [98, 96, 121] and described by an invariant descriptor which is d-dimensional [92, 4]. Normally, each image contains a few thousand of such local descriptors. Next, the descriptors of the whole dataset are clustered into k clusters using a variant of k-means algorithm, which creates a visual vocabulary. Sivic and Zisserman [151] cluster local descriptor of the database with on the order of ten thousand visual words. First scalable retrieval is achieved by Nister and Stewenius [113] with a hierarchical k-means that efficiently reaches several million visual words. Philbin *et al.* [125] show that an approximate k-means algorithm can be used to achieve higher performance with similar computational cost as hierarchical k-means. Finally, in [126] authors investigate how using an independent dataset to learn the vocabulary influences the performance. Our work is focused on this realistic scenario, *i.e.*, always using an independent training dataset to perform any kind of learning.

Local features for each image in the database are assigned to a respective visual word from the visual vocabulary. Bag-of-words (BoW) image representation [151] is then computed as a histogram of occurrences of visual words, with vector components weighted by inverse document frequency (*idf*) terms. An improvement over BoW, called Hamming embedding [69], provides binary signatures that refine the matching based on visual words, and is further extended by selective match kernels [163]. Methods that utilize significantly less visual words than a standard BoW are also proposed. Perronnin *et al.* [123] apply Fisher kernel, while Jegou *et al.* [72] propose vector of locally aggregated descriptors (VLAD) that accumulates, for each visual word, the differences of the respective visual word and the vectors assigned to it.

Image search. Inspired by text search, cosine similarity is used to compare two image representations. All aforementioned image representations are usually l_2 normalized and the search is performed as a nearest neighbor search between the query and database vectors, using the Euclidean distance. Ranking based on Euclidean distance of l_2 normalized vectors is equivalent to the ranking based on cosine similarity. For the BoW representation with a large vocabulary, the search is efficiently implemented via inverted file structure [151]. After the initial search is performed, one can utilize the geometry of the local features and perform spatial verification (SP) [125]. This is usually performed only on a set of top ranked images, which are in result re-ranked based on the number of inliers from SP, and the standard solution is to use the RANSAC algorithm [49]. SP can also be performed at the same time as the initial search, if the score of an image is computed during traversing of the inverted file in a *document at a time* (DAAT) manner [155]. Inverted file in this case contains both visual words and respective feature geometries [155]. Another popular post-processing step is query expansion (QE), introduced in the image retrieval domain by Chum *et al.* [31]. This method combines the verified images in the ranked list, to issue a new enhanced query and boost the recall of the system. An extension of QE, denoted as Hamming query expansion (HQE) [166] is a combination of QE and HE.

Compact image representation. A most common approach to obtain compact image representation from local-feature-based methods is by principal component analysis (PCA) [12] dimensionality reduction. In [73] and [124], aggregated descriptors (VLAD and Fisher vector respectively) are used followed by PCA to produce low dimensional image descriptors. Jegou and Chum [68] analyze the effects of PCA dimensionality reduction on the BoW and VLAD vectors. They propose a joint dimensionality reduction of multiple vocabularies. Image representation vectors are separately power-law normalized [124] for each vocabulary, concatenated and then jointly PCA-reduced and whitened [68]. In a paper about VLAD [5], authors propose a method for adaptation of the vocabulary built on an independent dataset and intra-normalization method that l_2 normalizes all VLAD components independently, which suppresses the burstiness effect [70]. In [74], a *democratic* weighted aggregation method for burstiness suppression is introduced. Compact *binary* representations are proposed as well [169, 177], with a few hundred *bits* per image, for efficient retrieval on billions of images.

In this work, we extend the approach of Jegou and Chum [68], by combining multiple vocabularies that are differing not just in random initialization of clustering procedure, but also in the data used for clustering. In this way, created vocabularies will be more complementary and joint dimensionality reduction of concatenated image vectors originating from several vocabularies will carry more information, resulting in a higher retrieval accuracy while maintaining the same compact dimensionality.

2.1.2. Convolutional-neural-network-based methods

CNN-based representation is appealing for image retrieval and in particular for retrieval with compact image representations. In this work, instance retrieval is cast as a metric learning problem, *i.e.*, an image embedding is learned so that the Euclidean distance captures the similarity well. Typical architectures for metric learning, such as the two-branch siamese [26, 59, 62] or triplet networks [175, 147, 61] employ *matching* and *non-matching* pairs in the training. Here, the problem of annotations is even more pronounced, *i.e.*, for classification one needs only object category label, while for particular objects the labels have to be per image pair. Two images from the same object category could potentially be completely different, *e.g.*, different viewpoints of the building or different buildings. We solve this problem in a fully automated manner, without any human intervention, by starting from a large unordered image collection and utilizing BoW-based retrieval pipeline and SfM modelling.

In the following text we discuss the related work for our main contributions regarding the CNN-based image retrieval, *i.e.*, the training data collection, the pooling approaches to construct a global image descriptor, the descriptor whitening, the multi-scale representation, and the query expansion techniques for CNN-based image representation.

Training data. A variety of related methods apply CNN activations on the task of image retrieval [55, 8, 76, 138, 168, 191]. The accuracy achieved on retrieval is evidence of generalization properties of CNNs. The employed networks are trained for image classification using ImageNet dataset [140] by minimizing classification error. Babenko *et al.* [9] go one step further and re-train such networks with a dataset that is closer to the target task. In [9], authors perform training with object classes that correspond to particular landmarks/buildings. Performance is improved on standard retrieval benchmarks. Despite the achievement, still, classification loss is optimized, instead of directly learning the metric used in image retrieval.

Constructing retrieval training datasets requires manual effort. In recent work, geotagged datasets with timestamps provide the ground for weakly-supervised fine-tuning of a triplet network [3]. Two images taken far from each other can be easily considered as non-matching. Matching examples are picked by the most similar nearby images, where, in the approach of [3], the similarity is defined by the current representation of the CNN. This is the first approach that performs end-to-end fine-tuning for image retrieval and, in particular, for the geo-localization task. The training images are now relevant to the final task. We differentiate by discovering matching and non-matching image pairs in an unsupervised way. At the same time, we derive matching examples based on 3D reconstructions which allows for harder examples.

Even though hard-negative mining is a standard process [54, 3], this is not the case with hard-positive examples. Mining of hard positive examples have been exploited in the work Simo-Serra *et al.* [149], where patch-level examples were extracted though the guidance from a 3D reconstruction. Hard-positive pairs have to be sampled carefully. Extreme hard-positive examples (such as minimal overlap between images or extreme scale change) do not allow to generalize and lead to over-fitting. For example, if the scale change between positive-pair images is too big, the network can learn proper representation for this specific landmark by memorizing, but it cannot generalize to unseen ones.

A concurrent work to ours also uses local features and geometric verification to select positive examples [57, 58]. In contrast to our fully unsupervised method, they start from a landmarks dataset, which had to be manually cleaned, and the landmark labels of the dataset, rather than the geometry, were used to avoid exhaustive evaluation. The same training dataset is used by Noh *et al.* [114] to learn global image descriptors using a saliency mask. However, during test time the CNN activations are seen as local descriptors, indexed independently, and used for a subsequent spatial-verification stage. Such approach boosts accuracy compared to global descriptors, but at the cost of much higher time and space complexity.

Pooling method. Early application of CNNs for image retrieval included methods that use the fully-connected layer activations as the global image descriptors [9, 55]. The work by Razavian *et al.* [138] moves the focus to the activations of convolutional layers followed by a global-pooling operation. A compact image representation is constructed in this fashion with dimensionality equivalent to the number of feature maps of the corresponding convolutional layer. In particular, authors of [138] propose to use max pooling, which is later generalized with integral max pooling [168] over all possible regions.

Sum pooling was initially proposed by Babenko and Lempitsky [8], which was shown to perform well especially due to the subsequent descriptor whitening. One step further is the weighted sum pooling of Kalantidis *et al.* [76], which can also be seen as a way to perform transfer learning. Popular encodings such as BoW, VLAD, and Fisher vectors are adapted in the context of CNN activations in the work of Mohedano *et al.* [106], Arandjelovic *et al.* [3], and Ong *et al.* [115], respectively. In the end, sum pooling of the feature embeddings is performed over all feature locations.

A hybrid scheme is the R-MAC method [168], which performs max pooling over regions and finally sum pooling of the regional descriptors. Mixed pooling is proposed globally for retrieval [110] and the standard local pooling is used for object recognition [84]. It is a linear combination of max and sum pooling. A generalization scheme, similar to our contribution presented in Chapter 6, Section 6.1.2, is proposed in the work of Cohen *et al.* [32] but in a different context. Cohen *et al.* [32] replace the standard local max pooling with the generalized one. Finally, generalized mean is used by Morere *et al.* [109] to pool the similarity values under multiple transformations.

Descriptor whitening. Whitening the data representation is known to be very essential for image retrieval since the work of Jegou and Chum [68]. Their interpretation lies in down-weighting co-occurrences and, thus, handling the problem of over-counting. The benefit of whitening is further emphasized in the case of CNN-based descriptors [137, 8, 168]. Whitening is commonly learned in a generative manner. More specifically, it is learned in an unsupervised way by PCA on an independent dataset.

We propose to learn the whitening transform in a discriminative manner, using the same acquisition procedure of the training data from 3D models. A similar approach has been used to whiten local-feature descriptors by Mikolajczyk and Matas [97]. In constrast, Gordo *et al.* [58] learn the whitening in the CNN in an end-to-end manner.

Multi-scale representation. Multi-scale processing is done during test time without any additional learning. It was introduced by Gordo *et al.* [58], and done by feeding the image to the network at multiple scales. The resulting descriptors are finally sum-pooled and re-normalized to constitute a multi-scale global image representation.

We adopt a different pooling of multi-scale descriptors, which utilizes a parameter learned during training, and we show that this approach is consistently superior to standard average pooling.

Query expansion. CNN global image descriptors can be combined with simple average query expansion (AQE) [8, 168, 76, 58] to boost the search recall. An initial query is issued by Euclidean search and AQE acts on the top-ranked images by average pooling of their descriptors. However, selecting an appropriate number of top-ranked images to be averaged, across different datasets, can be a non-trivial task. If the top-ranked images used for AQE are non-matching to the query, topic drift can easily happen [31]. We generalize AQE method by a weighted average approach, where the weights depend on the similarity between retrieved images and the respective query image. Our experiments verify that this is a more robust choice even for datasets with a significantly different statistics.

Query expansion can be additionally combined by a database-side augmentation (DBA) [4], which replaces every image signature in the database by a combination of itself and its neighbors. This procedure is done only once for the whole database, during the offline pre-processing. Again, the number of neighbors to be used in DBA is not an easy choice, as it depends on unknown dataset statistics.

2.1.3. Adversarial attacks on image retrieval

Adversarial attacks were introduced by Szegedy *et al.* [158] on the task of image classification. In that context, adversarial attacks are divided into two categories, namely *non-targeted* and *targeted*. The goal of non-targeted attacks is to change the prediction of a test image to an arbitrary class [108, 107], while targeted attacks attempt to make a specific change of the network prediction, *i.e.*, to misclassify the test image to a predefined target class [158, 22, 43]. Follow up approaches are categorized to *white-box* attacks [158, 56] if there is complete knowledge of the model or to *black-box* [118, 119] otherwise. Adversarial images are generated by various methods in the literature, such as optimization-based approaches using box-constrained L-BFGS optimizer [158], gradient descent with change of variable [22]. A fast gradient sign method [56] and variants [82, 43] are designed to be *fast* rather than optimal, while DeepFool [108] analytically derives an optimal solution method by assuming that neural networks are totally linear. All these approaches solve an optimization problem given a test image and its associated class in the case of non-targeted attacks or a test image and a target class in the case of targeted attacks. A universal non-targeted approach is proposed by Moosavi *et al.* [107], where an image-agnostic Universal Adversarial Perturbation (UAP) is computed and applied to unseen images to cause network misclassification.

Adversarial attacks on image retrieval are studied by recent work [87, 91, 192] in a non-targeted scenario for CNN-based approaches. Liu *et al.* [91] and Zheng *et al.* [192] adopt the optimization-based approach [158], while Li *et al.* [87] adopt the UAP [107]. Similar attacks on classical retrieval systems that are based on SIFT local descriptors [92] have been addressed in an earlier line of work by Do *et al.* [38, 37]. In this work, we deal with the targeted attacks exclusively. To the best of our knowledge, no existing work focuses on targeted adversarial attacks for image retrieval.

2.2. Sketch-based image retrieval

Sketch-based image retrieval has been, until recently, handled with hand-crafted descriptors [47, 63, 141, 120, 176, 14, 128, 142, 180, 164, 165]. Deep learning methods have been applied to the task of sketch-based retrieval [11, 129, 184, 143, 15, 148, 90] much later than to the related task of image retrieval. We attribute the delay to the fact that the training data acquisition for sketch-based retrieval is much more tedious compared to image-based retrieval because it not only includes labeling the images, but also sketches must be drawn in large quantities. Methods with no learning typically carry no assumptions on the depicted categories, while the learning based methods often include category recognition into training. The method proposed in this work aims at generic sketch-based image retrieval, not limited to a fixed set of categories; it is, actually, not even limited to objects.

2.2.1. Learning-free methods

Learning-free methods have followed the same initial steps as in the traditional image search. These include the construction of either global [23, 141, 128] or local [46, 139, 63, 18, 176] image and/or sketch representations. Local representations are also using vector quantization to create a bag-of-words model [93]. The domain gap between hand-drawn sketches and images is handled by applying representations that are easily applicable on both domains. Histogram of gradients [36] is a popular choice for both global [141, 128] and local representations [63]. The latter is also extended to color instances [14]. Further cases are symmetry-aware and flip invariant descriptors [18], and descriptors that are based on local contours [139] or line segments [176]. Recently, asymmetric feature maps (AFM) are used to derive a short vector image representation, that supports efficient scale and translation invariant sketch-based image retrieval [164, 165]. Despite their small dimensionality, these short codes provide query localization in the retrieved image. An efficient approximation of Chamfer matching allows [20, 156] to scale the searchable collections to millions or even billion images. However, precision is sacrificed along with the transformation invariance. In contrast, the method proposed in this work offers high precision, is fully translation invariant, and scalable, because it reduces to nearest-neighbor search in a descriptor space.

2.2.2. Learning-based methods

Learning-based methods require annotated data in both domains, typically for a fixed set of object categories, making the methods [174, 11, 129, 184, 143, 148, 90, 15, 153] to be category specific and may limit a good performance to those categories. End-to-end learning methods are applied to both category level [90, 15] and to fine-grained, *i.e.* sub-category level retrieval [184, 143, 148, 153], while sometimes a different model per category has to be learned [88, 184, 152, 153]. A common characteristic of these deep-learning methods [184, 143, 148, 90, 15] is that a sequence of different learning and fine-tuning stages is applied. These include training with a category loss on images and/or sketches, ranking loss of category level similarity, fine-grained similarity, and cross-view pairwise loss. Deep learning has not been restricted to learning sketch/image descriptors; learning hash codes is achieved with a combination of different losses [90].

Training data for all these stages are required, which involves massive manual effort at various stages. For example, the Sketchy dataset [143] required going through about 70k images and selecting those that are *sketchable*. More than 600 users collectively spent about 4k hours to create 5 sketches per sketchable image. On the contrary, our proposed fine-tuning does not require any manual annotation.

2.3. Domain generalization

Domain generalization is handled in a variety of ways, ranging from learning domain invariant features [52, 111], to learning domain invariant classifiers [181, 78], or both [85, 13]. Then, well known faster R-CNN [53] object detection approach is extended for domain generalization with new deep learning components in the work of Chen *et al.* [25]. Several approaches assume that multiple domains are available during training and the goal is to generalize inference to an unseen domain. This is the case in, *e.g.*, the work of Mancini *et al.* [95], where a separate classifier per training domain is learned and their combination is used for testing. Adversarial examples are shown useful in learning an invariant representation across the multiple training domains [86]. Adversarial training is also used in a work [173] with no assumptions about multiple available domains during training, where data augmentation is used to improve the generalization. Finally, sometimes the focus is on one-way shift between two domains, such as sketch-based retrieval (addressed below) or learning on real photos and testing on art [34, 35].

Research on domain generalization is facilitated by the advent of new appropriate benchmarks. This is the case with the benchmark released in the work of Li *et al.* [85], where four domains of increasing visual abstraction are used, namely *photos*, *art*, *cartoon*, and *sketches* (PACS). Prior domain generalization methods [52, 111, 181] are shown effective on PACS, while simply training a CNN on all the available (seen) domains is a very good baseline [85]. Another relevant benchmark is the *Behance Artistic Media* (BAM) dataset [179]. In the work of [179], authors evaluate baseline approaches, such as directly testing on an unseen domain a CNN classifier that is trained on multiple training domains. We tackle the same problem from the representation point of view and focus on the underlined shapes. Our shape descriptor proposed in this work is extracted and the class labels are used only to train a linear classifier. In this fashion, we are able to train on a single domain and test on all the rest. This is in contrast to many prior domain generalization approaches that require different domains present in the training set.

Chapter 3

Standard Test Datasets and Evaluation Protocols

In this chapter, we thoroughly describe all standard test datasets used throughout this work, as well as their evaluation protocols. We split the benchmarks into three main categories: (i) image retrieval benchmarks are described in Section 3.1; (ii) sketch-based image retrieval benchmarks are discussed in Section 3.2; and, (iii) domain genralization benchmark is introduced in Section 3.3.

One of the contributions of this work is addressing of issues with image retrieval benchmarking on standard and popular Oxford5k [125] and Paris6k [126] datasets. Newly proposed benchmark is described in Chapter 4, and an extensive comparison of the state-of-the-art methods that is performed on it is described in detail in Chapter 7.

3.1. Image retrieval datasets

We start by describing two most popular image retrieval datasets in Section 3.1.1, that depict popular landmarks of two cities, namely Oxford and Paris. Then, two datasets made by a selection of creators personal photographs are described in Section 3.1.2, covering a large variety of scenes, man-made as well as natural. Finally, a distractor set of around 100k images, usually added to evaluate retrieval at scale, is presented in Section 3.1.3.

3.1.1. Oxford5k and Paris6k

Oxford Buildings [125] and Paris [126] datasets, commonly denoted as *Oxford5k* and *Paris6k*, contain a set of 5,062 and 6,392 high-resolution (1024 × 768) images, respectively. Both datasets contain 11 different annotated landmarks together with distrac-

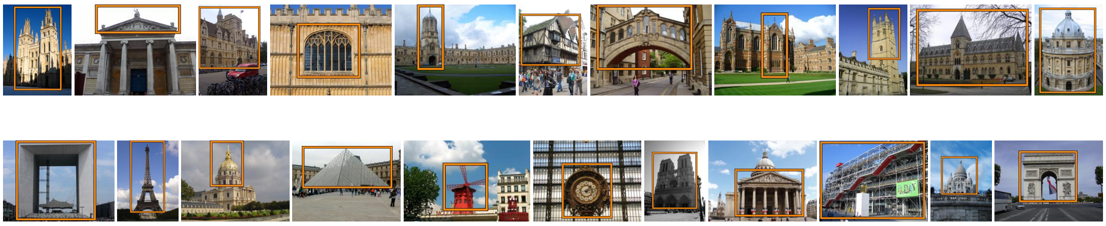


Figure 3.1. Sample queries for 11 landmarks of Oxford5k (top) and Paris6k (bottom) datasets.

Figure 3.2. Sample queries for Holidays (top) and Copydays (bottom) datasets.

tors, downloaded from Flickr[1] by searching for tags of popular landmarks from Oxford and Paris, respectively. For each of the 11 landmarks there are 5 different images with query regions defined by a bounding box, which are supposed to cover approximately the same physical surface of the respective landmark. This results in a total of 55 query regions per dataset. Sample query regions for 11 landmarks of Oxford5k and Paris6k are given in Figure 3.1.

For each image and landmark in this dataset, one of four possible labels is given by a consensus of human annotators [125]:

- *Good:* A nice, clear picture of the object/building.
- *OK:* More than 25% of the object is clearly visible.
- *Junk:* Less than 25% of the object is visible, or there are very high levels of occlusion or distortion.
- *Absent:* The object is not present.

For the performance evaluation three labels are essentially used, *i.e.*, *good* and *ok* images are used as *positive* examples of the landmark in question, *absent* images as *negative* examples and *junk* as *null* examples that are ignored (the evaluation is performed as if they were not present in the database).

The performance is evaluated as follows [125]. The performance for a single query is evaluated as the average precision (AP) measure computed as the area under the precision-recall curve. Precision is defined as the ratio of retrieved positive images to the total number of images retrieved, while recall is defined as the ratio of the number of retrieved positive images to the total number of positive images in the database. To reach an ideal precision-recall curve, the image retrieval system has to obtain precision 1 over all recall levels, which will result in an average precision equal to 1. Next, AP is averaged over all queries in the dataset (55 queries for Oxford5k and Paris6k) to obtain a mean average arecision (mAP), which is a single number that evaluates the overall performance. Note that, all images in these datasets have the natural upright orientation.

3.1.2. Holidays and Copydays

INRIA Holidays [69] and INRIA Copydays [44] datasets, commonly denoted as *Holidays* and *Copydays*, is a selection of personal holidays photos (1,491 and 3,055 high-resolution images, respectively) from INRIA, including a large variety of scene types (natural, man-made, water and fire effects, *etc.*). For Holidays, a set of 500 images from the whole dataset is selected for query purposes. For Copydays, original 157 images are

[1] www.flickr.com

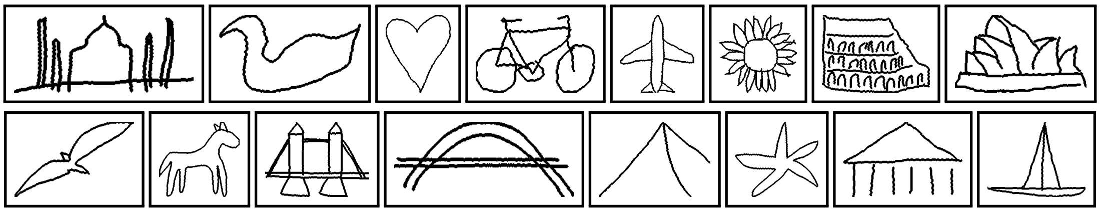


Figure 3.3. Sample query sketches for Flickr15k dataset.

selected as queries, while the rest are created by distorting original images with three kinds of artificial attacks: JPEG, cropping and "strong". Sample query images from Holidays and Copydays datasets are given in Figure 3.2.

The performance for both datasets is reported as mean average precision (mAP) [125], as described in Section 3.1.1, after excluding query image from the results, *i.e.*, assuming it to be *null* example that is ignored from the evaluation. Unlike Oxford5k and Paris6k, images in Holidays dataset are not always in the natural upright orientation. More specifically, most of the images are correctly oriented (or can be with the help of EXIF orientation tag). However about 5%–10% of the images, spread over the groups, are rotated (unnaturally for a human observer). Because of this, two additional versions of Holidays dataset emerged: (i) images were rotated using EXIF information only [121]; (ii) images were manually rotated whenever correct upright orientation was obvious [57].

3.1.3. Oxford Distractors

Oxford Distractors dataset [125], also known as *Oxford100k*, is created by crawling Flickr with its 145 most popular tags and consists of 99,782 high-resolution (1024 × 768) images. It is frequently used in combination with standard image retrieval datasets, to allow for evaluation at larger scale. We denote the combination of Oxford100k distractors and Oxford5k, Pairs6k, and Holidays as *Oxford105k*, *Paris106k*, and *Holidays101k*, respectively.

3.2. Sketch-based image retrieval datasets

Four standard sketch-based image retrieval benchmarks used throughout the thesis are presented in this section. Section 3.2.1 describes a dataset designed to evaluate category-level performance, while for fine-grained performance datasets in Sections 3.2.2 and 3.2.3 are used. Finally, sketch-based image retrieval at large scale is tested on a dataset presented in Section 3.2.4.

3.2.1. Flickr15k

Flickr15k dataset [63] consists of 14,660 database images collected by crawling Flickr images for objects that represent distinct types of shape, using a set of semantic tags (*e.g.*, Louvre, pyramids, *etc.*). The images are manually labeled into final 33 categories, which include particular object instances (Brussels Cathedral, Colosseum, Arc de Triomphe, *etc.*), generic objects (airplane, bicycle, bird, *etc.*), and shapes (circle shape, star shape, heart, balloon, *etc.*). Selected set of images exhibit significant affine variation in appearance and in presence of background clutter.

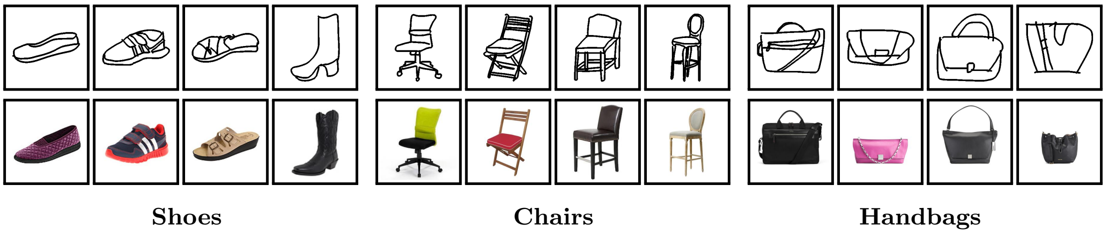


Figure 3.4. Sample sketch–photo pairs for Shoes, Chairs, and Handbags datasets, respectively.

Ten non-expert sketchers were recruited [63] to provide free-hand sketched queries for each category. Participants were shown the example images per category before the drawing, then the sketches were drawn solely based on memory. This resulted in 330 query sketches in total, *i.e.*, one per category for each of the 10 sketchers. Sample queries from Flickr15k dataset are presented in Figure 3.3. The performance is reported as mean average precision (mAP) [125] over all 330 queries, as described in Section 3.1.1.

3.2.2. Shoes, Chairs, and Handbags

Shoes, *Chairs*, and *Handbags* [184, 153] datasets contain images and sketches from a single category each, *i.e.* shoe, chair, and handbag category, respectively. The image part of datasets was collected from online shopping websites (IKEA, Amazon, and Taobao) or product datasets (UT-Zap50k [183]). The sketch part of datasets was created by recruiting 22 non-expert volunteer sketchers [184]. One photo of shoe/chair/handbag was shown to a volunteer on a tablet for 15 seconds, then, he/she had to sketch the object they just saw using their fingers on the tablet. As a result, these datasets consist of pairs of a photo and a corresponding hand-drawn detailed sketch of this photo, both in a 256 × 256 resolution. Examples of sketch–photo pairs for all three datasets are illustrated in Figure 3.4.

There are 419, 297, and 568 sketch–photo pairs of shoes, chairs, and handbags, respectively. Out of these, 304, 200, and 400 pairs are selected for training, and 115, 97, and 168 for testing shoes, chairs, and handbags, respectively. At evaluation, the search is performed on test images using test sketches as queries. The underlying task is quite different compared to Flickr15k (see Section 3.2.1 for more details). The photograph used to generate the sketch is to be retrieved, while all other images are considered false positives. The performance is measured via the matching accuracy at the top K retrieved images, averaged over all sketch queries, denoted by acc.@K [184].

3.2.3. Sketchy

Sketchy [143] dataset consists of 12,500 images and 75,471 sketches (roughly 5 sketches per photo), spanning 125 categories of common objects like horse, apple, axe, guitar, *etc.* These categories are chosen using the criteria in [45]: exhaustive, recognizable, and specific; with an additional *sketchability* criterion [143]. To pick *sketchable* photographs, a total of 69,495 images were reviewed with 24,819 being selected, using a subjective ranking by answering the question “How easily could a novice artist capture the subject category and pose?” As part of this process, volunteer annotators ranked each image with a subjective score ranging from 1 (very easy to sketch) to 5 (very difficult to sketch). Then, for each of the 125 categories, 100 images are chosen at random with a

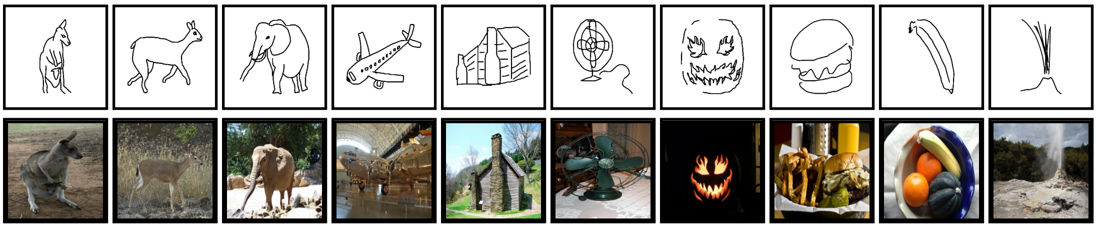


Figure 3.5. Sample sketch–photo pairs for Sketchy dataset.

distribution of 40 very easy, 30 easy, 20 average, 10 hard, and 0 very hard photographs. This constitutes 12,500 examples for the image part of the Sketchy dataset.

For the collection of the sketch part of the dataset, volunteer sketchers were employed [143]. Each participant is given a randomly selected category name, a random image from the respective category, and a blank canvas on which to sketch. The image can be seen for 2 seconds at a time, as many times as needed, but the canvas is cleared every time the volunteer decides to take a look at the image. Sketchers are instructed to sketch the object from the named category with a pose similar to that of the object in the photograph, sketch only the object itself without the image clutter, and avoid shading. This procedure forces workers to draw from memory, rather than directly copying the boundaries. Each photograph was sketched five times by different participants. In total, 644 individuals were involved in this procedure, over the course of 6 months, and they collectively spent 3,921 hours sketching. Finally, all sketches were manually validated to remove erroneous ones, resulting in a total of 75,471 sketches for the sketch part of the Sketchy dataset. Examples of sketch–photo pairs for the Sketchy dataset are given in Figure 3.5.

For the evaluation purposes, 1,250 database photos and 6,312 query sketches are selected, still spanning the same 125 categories. At test time, the search is performed on all images using sketches as queries. Each sketch query is associated to a single image, the one that prompted the creation of this particular sketch. The performance is measured via recall at various ranks, where recall@K for a particular sketch query is 1 if the corresponding photo is within the top K retrieved results and 0 otherwise [143]. The results are averaged over all queries to produce one final recall@K for the whole dataset. Note that recall@K is basically the same as acc.@K of the Shoes/Chairs/Handbags datasets (see Section 3.2.2 for more details).

3.2.4. SBIR175

SBIR175 [120] dataset consists of 1.2M images and 175 sketch queries. The image part of the dataset is a combination of images taken from the MIR-Flickr-1M [64] and ImageNet [42] image collections. Around 1M images are taken from the first collection, while around 200k images from the second collection, in order to cover many common objects. Five volunteer participants drew sketches on a touch-based tablet, which resulted in 75 sketches [120]. The other 100 are taken from a crowd-sourced sketch database of Eitz *et al.* [45]. Query sketches depict objects from 40 different categories, and sample query sketches are presented in Figure 3.6.

The performance is measured via precision at K top-ranked images per query, and average precision over all queries is reported [120]. This dataset has no available annotation, so we use external annotators to manually evaluate the top retrieved images for each query and evaluated method. We evaluate the results based on a *particular*

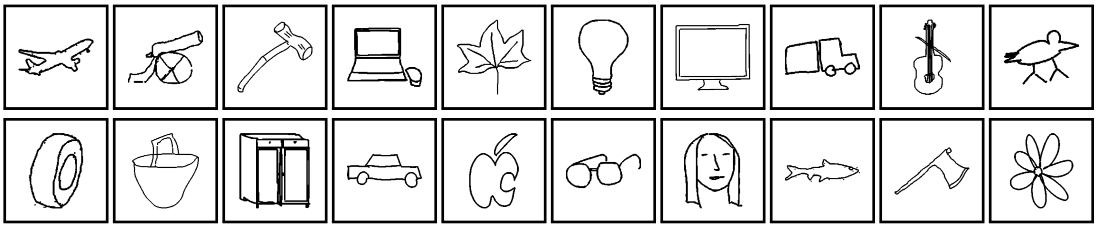


Figure 3.6. Sample query sketches for SBIR175 dataset.

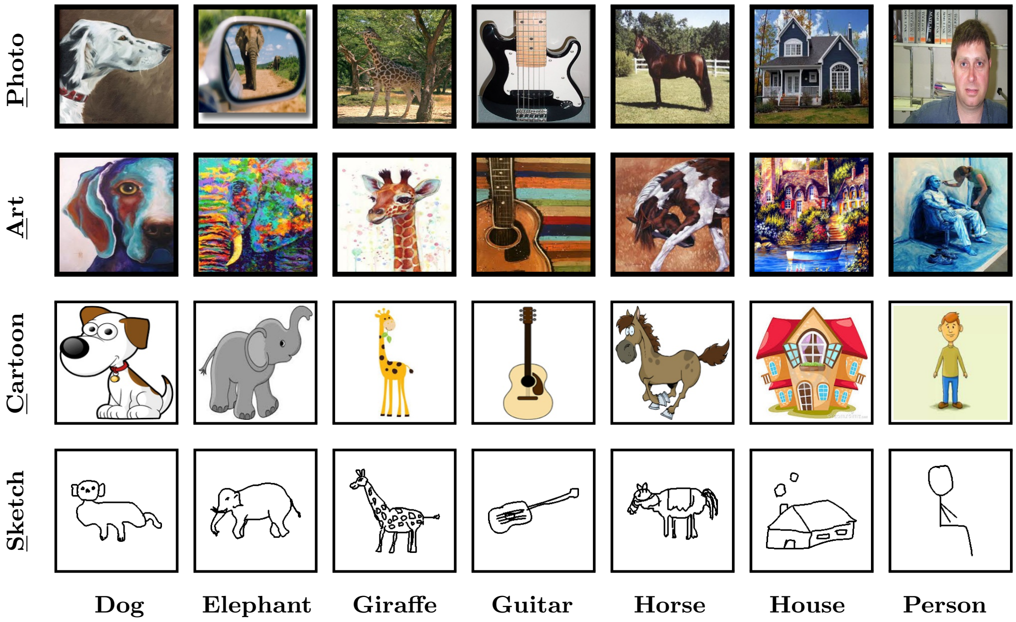


Figure 3.7. Sample images per domain (Photo, Art (painting), Cartoon, Sketch) and per category (dog, elephant, giraffe, guitar, horse, house, person) for PACS dataset.

instance retrieval paradigm. In other words, results are annotated per query instance, according to its shape, but not according to the general category that the query belongs to. External annotators did evaluation for a single query at a time, using following instructions:

- *Positive:* An object that is of the correct category and the same pose or similar as the query. An object that is of a different category but very similar in shape of the sketch, in cases where a similar sketch could have been drawn with the intention to retrieve the image in question.
- *Negative:* Retrieved image and sketch contain objects with different shapes, while the category might be matching or not.

3.3. Domain generalization datasets

To evaluate domain generalization performance of image representations, one needs a dataset that covers various object categories, where each category is contained in different image domains. A popular dataset designed to achieve this goal is presented in Section 3.3.1.

3.3.1. PACS

PACS is a recently introduced domain generalization dataset by Li *et al.* [85]. It consists of 9,991 images coming from 4 domains with varying level of abstraction, namely: *Photo*, *Art* (painting), *Cartoon*, and *Sketch*. Images are labeled according to 7 categories: *dog*, *elephant*, *giraffe*, *guitar*, *horse*, *house*, and *person*. A sample image per domain and per category is given in Figure 3.7. This dataset is created by intersecting the classes found in Caltech256 (Photo), Sketchy (Photo, Sketch) [143], TU-Berlin (Sketch) [45], and Google Images (Art, Cartoon, Photo).

For evaluation, each time, one domain is considered unseen, also called *target* or *test* domain, while the images of the other 3 are used for training. Finally, multi-class accuracy is evaluated on the unseen domain.

Chapter 4

Large-Scale Image Retrieval Test Datasets

Image retrieval methods have gone through significant development in the last decade. In order to measure the progress and compare different methods, standardized image retrieval benchmarks are used. Besides the fact that a benchmark should simulate a real-world application, there are a number of properties that determine the quality of a benchmark: the *reliability of the annotation*, the *size*, and the *challenge level*.

Errors in the annotation may systematically corrupt the comparison of different methods. Too small datasets are prone to over-fitting and do not allow the evaluation of the efficiency of the methods. The reliability of the annotation and size of the dataset are competing factors, as it is difficult to secure accurate human annotation of large datasets. The size is commonly increased by adding a distractor set, which contains irrelevant images that are selected in an automated manner (different tags, GPS information, *etc.*) Finally, benchmarks where all the methods achieve almost perfect results [83] cannot be used for further improvement or quantitative comparison.

Many datasets have been introduced to measure the performance of image retrieval. Most popular are Oxford [125] and Paris [126]. Numerous methods of image retrieval [31, 121, 29, 102, 166, 8, 168, 76, 135, 58] and visual localization [51, 3] have used these datasets for evaluation. Reason for their popularity is that, in contrast to datasets that contain small groups of 4-5 similar images like Holidays [69] and UKB [113], Oxford and Paris contain queries with up to hundreds of positive images.

Despite the popularity, there are known issues with the two datasets, which are related to all three important properties of evaluation benchmarks. First, there are errors in the annotation, including both false positives and false negatives, see Figure 4.2. Further inaccuracy is introduced by queries of different sides of a landmark, sharing the annotation despite being visually distinguishable. Second, the annotated datasets are relatively small (5,062 and 6,392 images respectively). Third, current methods report near-perfect results on both the datasets. It has become difficult to draw conclusions from quantitative evaluations, especially given the annotation errors [67].

The lack of difficulty is not caused by the fact that non-trivial instances are not present in the dataset, but due to the annotation. The annotation was introduced at the early years of image retrieval. At that time, the annotators had different perception of what the limits of image retrieval are. Many instances that are nowadays considered as a change of viewpoint expected to be retrieved, are *de facto* excluded from the evaluation by being labelled as *Junk*.

The size issue of the datasets is partially addressed by the Oxford 100k *distractor set*. However, this contains false negative images, as well as images that are not challenging. The state-of-the-art methods maintain near-perfect results even in the presence of these

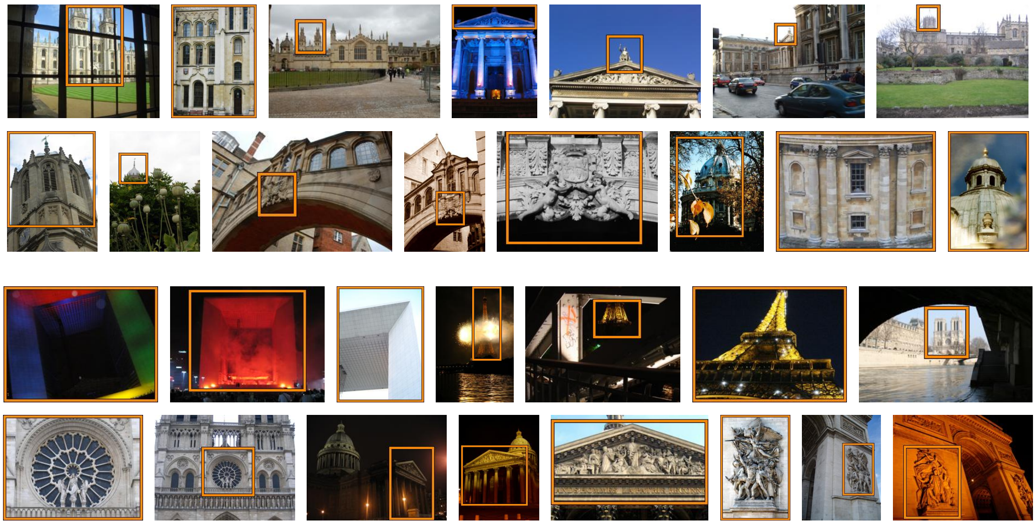


Figure 4.1. The newly added queries for $\mathcal{R}$Oxford (top) and $\mathcal{R}$Paris (bottom) datasets. Merged with the original queries, they comprise a new set of 70 queries in total.

distractors. As a result, additional computational effort is spent with little benefit in drawing conclusions.

As a contribution, we generate new annotation for Oxford and Paris datasets, update the evaluation protocol, define new, more difficult queries, and create new set of challenging distractors. As an outcome we produce *Revisited Oxford*, *Revisited Paris*, and an accompanying distractor set of one million images. We refer to them as $\mathcal{R}$Oxford, $\mathcal{R}$Paris, and $\mathcal{R}$1M respectively. Another contribution connected to these revisted datasets is described in Chapter 7, where we provide extensive evaluation of image retrieval methods, ranging from local-feature based to CNN-descriptor based approaches, including various methods of re-ranking.

The contents of this chapter have been published in [130]. The revisited benchmark, along with the new distractor images, is publicly available[1]. The rest of the chapter is organized as follows. The original Oxford and Paris datasets are briefly introduced in Section 4.1. We describe how and why we revisit the annotation and add new difficult queries in Section 4.2, while a new evaluation protocol with three difficulty setups is described in Section 4.3. Newly proposed distractor set with more than a million challenging images is presented in Section 4.4. Finally, concluding remarks are given in Section 4.5.

4.1. The original datasets

The original Oxford and Paris datasets consist of 5,063 and 6,392 high-resolution (1024 × 768) images, respectively. Each dataset contains 55 queries comprising 5 queries per landmark, coming from a total of 11 landmarks. Given a landmark query image, the goal is to retrieve all database images depicting the same landmark. The original annotation (labeling) is performed manually and consists of 11 ground truth lists since 5 images of the same landmark form a *query group*. Three labels are used, namely, *positive*, *junk*, and *negative*[2].

[1] cmp.felk.cvut.cz/revisitop

[2] We rename the originally used labels {good, ok, junk, and absent} for the purpose of consistency with our terminology. Good and ok were always used as positives.

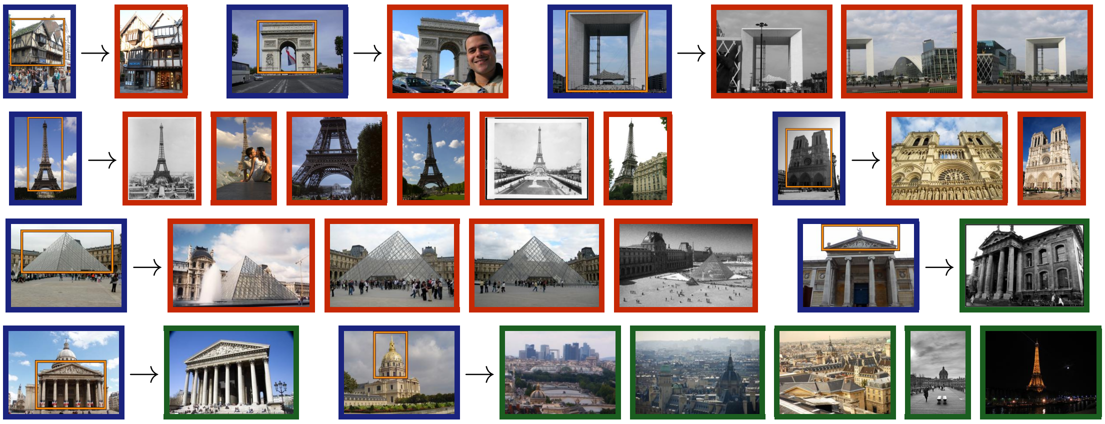


Figure 4.2. Examples of *extreme* labeling mistakes in the original labeling. We show the **query (blue)** image and the associated database images that were originally marked as **negative (red)** or **positive (green)**. Best viewed in color.

Positive images clearly depict more than 25% of the landmark, junk less than 25%, while the landmark is not shown in negative ones. The performance is measured via mean average precision (mAP) [125] over all 55 queries, while junk images are ignored, *i.e.* the evaluation is performed as if they were not present in the database. More details on the original Oxford and Paris datasets is given in Chapter 3, Section 3.1.1.

4.2. Revisiting the annotation

The annotation is performed by 5 annotators, and it is performed in the following steps.

Query groups. Query groups share the same ground-truth list and simplify the labeling problem, but also cause some inaccuracies in the original annotation. *Balliol* and *Christ Church* landmarks are depicted from a different (not fully symmetric) side in the 2nd and 4th query, respectively. *Arc de Triomphe* has three day and two night queries, while day-night matching is considered a challenging problem [172, 132]. We alleviate this by splitting these cases into separate groups. As a result, we form 13 and 12 query groups on Oxford and Paris, respectively.

Additional queries. We introduce new and more challenging queries (see Figure 4.1) compared to the original ones. There are 15 new queries per dataset, originating from five out of the original 11 landmarks, with three queries per landmark. Along with the 55 original queries, they comprise the new set of 70 queries per dataset. The query groups, defined by visual similarity, are 26 and 25 for $\mathcal{R}$Oxford and $\mathcal{R}$Paris, respectively. As in the original datasets, the query object bounding boxes are simulating not only a user attempting to remove background clutter, but also cases of large occlusion.

Labeling step 1: Selection of potential positives. Each annotator manually inspects the whole dataset and marks images depicting any side or version of a landmark. The goal is to collect all images that are originally incorrectly labeled as negative. Even uncertain cases are included in this step and the process is repeated for each landmark. Apart from inspecting the whole dataset, an interactive retrieval tool is used to actively search for further possible positive images. All images marked in this phase are merged

Table 4.1. Number of images switching their labeling from the original annotation (positive, junk, negative) to the new one (easy, hard, unclear, negative).

$\mathcal{R}$Oxford				
Labels	Easy	Hard	Uncl.	Neg.
Positive	438	50	93	1
Junk	50	222	72	9
Negative	1	72	133	63768

$\mathcal{R}$Paris				
Labels	Easy	Hard	Uncl.	Neg.
Positive	1222	643	136	6
Junk	91	813	835	61
Negative	16	147	273	71621

together with images originally annotated as positive or junk, creating a list of *potential positives* for each landmark.

Labeling step 2: Label assignment. In this step, each annotator manually inspects the list of potential positives for each query group and assigns labels. The possible labels are *Easy*, *Hard*, *Unclear*, and *Negative*. All images not in the list of potential positives are automatically marked negative. The instructions given to the annotators for each of the labels are as follows.

- *Easy:* The image clearly depicts the query landmark from the same side, with no large viewpoint change, no significant occlusion, no extreme illumination change, and no severe background clutter. In the case of fully symmetric sides, any side is valid.
- *Hard:* The image depicts the query landmark, but with viewing conditions that are difficult to match with the query. The depicted (side of the) landmark is recognizable without any contextual visual information.
- *Unclear:* (a) The image possibly depicts the landmark in question, but the content is not enough to make a certain guess about the overlap with the query region, or context is needed to clarify. (b) The image depicts a different side of a partially symmetric building, where the symmetry is significant and discriminative enough.
- *Negative:* The image is not satisfying any of the previous conditions. For instance, it depicts a different side of the landmark compared to that of the query, with no discriminative symmetries. If the image has any physical overlap with the query, it is never negative, but rather unclear, easy, or hard according to the above.

Labeling step 3: Refinement. For each query group, each image in the list of potential positives has been assigned a five-tuple of labels, one per annotator. We perform majority voting in two steps to define the final label. The first step is voting for {easy,hard}, {unclear}, or {negative}, grouping easy and hard together. In case majority goes to {easy,hard}, the second step is to decide which of the two. Draws of the first step are assigned to unclear, and of the second step to hard. Illustrative examples are (EEHUU) → E, (EHUUN) → U, and (HHUNN) → U. Finally, for each query group, we inspect images by descending label entropy to make sure there are no errors.

Revisited datasets: $\mathcal{R}$Oxford and $\mathcal{R}$Paris. Images from which the queries are cropped are excluded from the evaluation dataset. This way, unfair comparisons are avoided in the case of methods performing off-line preprocessing of the database [4, 67]; any preprocessing **should not include any part of query images**. The revisited datasets, namely, $\mathcal{R}$Oxford and $\mathcal{R}$Paris, comprise 4,993 and 6,322 images respectively, after removing the 70 queries.

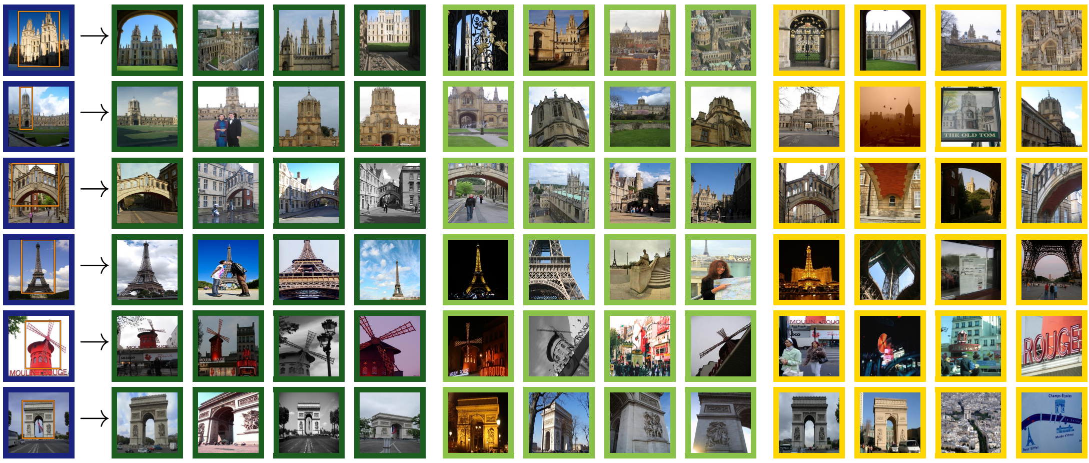


Figure 4.3. Sample **query (blue)** images and images that are respectively marked as **easy (dark green)**, **hard (light green)**, and **unclear (yellow)**. Best viewed in color.

In Table 4.1, we show statistics of label transitions from the old to the new annotations. Note that errors in the original annotation that affect the evaluation, *e.g.* negative moving to easy or hard, are not uncommon. The transitions from junk to easy or hard are reflecting the greater challenges of the new annotation. Representative examples of *extreme* labeling errors of the original annotation are shown in Figure 4.2. In Figure 4.3, representative examples of easy, hard, and unclear images are shown for several queries. This will help understanding the level of challenge of each evaluation protocol listed below.

4.3. Evaluation protocol

Only the cropped regions are to be used as queries; never the full image, since the ground-truth labeling strictly considers only the visual content inside the query region.

The standard practice of reporting mean average precision (mAP) [125] for performance evaluation is followed. Additionally, mean precision at rank K (mP@K) is reported. The former reflects the overall quality of the ranked list. The latter reflects the quality of the results of a search engine as they would be visually inspected by a user. More importantly, it is correlated to performance of subsequent processing steps [31, 77]. During the evaluation, positive images should be retrieved, while there is also an ignore list per query. Three evaluation setups of different difficulty are defined by treating labels (easy, hard, unclear) as positive or negative, or ignoring them:

- **Easy (E):** *Easy* images are treated as positive, while *Hard* and *Unclear* are ignored (same as *Junk* in [125]).

- **Medium (M):** *Easy* and *Hard* images are treated as positive, while *Unclear* are ignored.

- **Hard (H):** *Hard* images are treated as positive, while *Easy* and *Unclear* are ignored.

If there are no positive images for a query in a particular setting, then that query is excluded from the evaluation.

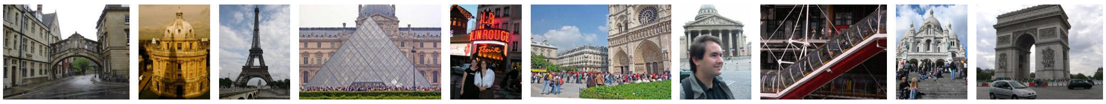


Figure 4.4. Sample false negative images in Oxford100k.

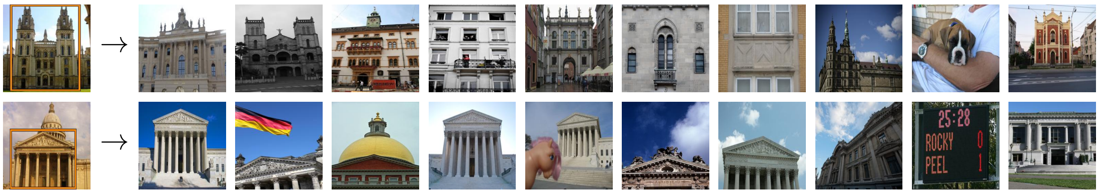


Figure 4.5. The most distracting images per query for two queries.

The original annotation and evaluation protocol is closest to our **Easy** setup. Even though this setup is now trivial for the best performing methods, it can still be used for evaluation of *e.g.* near duplicate detection or retrieval with ultra short codes. The other setups, **Medium** and **Hard**, are challenging and even the best performing methods achieve relatively low scores. See Chapter 7 for details.

4.4. Distractor set $\mathcal{R}$1M

Large scale experiments on Oxford and Paris dataset are commonly performed with the accompanying distractor set of 100k images, namely Oxford100k [125]. Recent results [67, 66] show that the performance only slightly degrades by adding Oxford100k in the database compared to a small-scale setting. Moreover, it is not manually cleaned and, as a consequence, Oxford and Paris landmarks are depicted in some of the distractor images (see Figure 4.4), hence adding further noise to the evaluation procedure.

Larger distactor sets are used in the literature [125, 126, 69, 161] but none of them are standardized to provide a testbed for direct large scale comparison nor are they manually cleaned [69]. Some of the distractor sets are also biased, since they contain images of different resolution than the Oxford and Paris datasets.

We construct a new distractor set with exactly 1,001,001 high-resolution (1024 × 768) images, which we refer to as $\mathcal{R}$1M dataset. It is cleaned by a semi-automatic process. We automatically pick hard images for a number of state-of-the-art methods, resulting in a challenging large scale setup.

YFCC100M and semi-automatic cleaning. We randomly choose 5M images with GPS information from YFCC100M dataset [160]. Then, we exclude UK, France, and Las Vegas; the latter due to the *Eiffel Tower* and *Arc de Triomphe* replicas. We end up with roughly 4.1M images that are available for downloading in high resolution. We rank images with the same search tool as used in labeling step 1. Then, we manually inspect the top 2k images per landmark, and remove those depicting the query landmarks (faulty GPS, toy models, and paintings/photographs of landmarks). In total, we find 110 such images.

Un-biased mining of distracting images. We propose a way to keep the most challenging 1M out of the 4.1M images. We perform all 70 queries into the 4.1M database

Table 4.2. Performance (mAP) evaluation with the Medium protocol for different distractor sets. The methods considered are (1) Fine-tuned ResNet101 with GeM pooling [135]; (2) Off-the-shelf AlexNet with MAC pooling [138]; (3) HesAff–rSIFT–ASMK* [163]; (4) Fine-tuned ResNet101 with R-MAC pooling [58]; (5) HesAff–rSIFT–VLAD [72]. The sanity check includes evaluation for different distractor sets, *i.e.* all, hardest subset chosen by method (1,2,3), (1,2,3,4), (1,2,4,5), and a random 1M sample.

Distractor set	$\mathcal{R}$Oxford					$\mathcal{R}$Paris				
	Method									
	(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)
4M	33.3	11.1	33.2	33.7	15.6	40.7	11.4	30.0	45.4	17.9
1M (1,2,3)	33.9	11.1	34.8	33.9	17.4	44.1	11.8	31.7	48.1	19.6
1M (1,2,3,4)	33.7	11.1	34.8	33.8	17.5	43.8	11.8	31.8	47.7	19.7
1M (1,2,3,5)	33.7	11.1	34.6	33.9	17.2	43.5	11.7	31.4	47.7	19.2
1M (random)	37.6	13.7	37.4	38.9	20.4	47.3	16.2	34.2	53.1	21.9

with a number of methods. For each query and for each distractor image we count the fraction of easy or hard images that are ranked after it. We sum these fractions over all queries of $\mathcal{R}$Oxford and $\mathcal{R}$Paris and over different methods, resulting in a measurement of how *distracting* each distractor image is. We choose the set of 1M most distracting images and refer to it as the $\mathcal{R}$1M *distractor set*.

Three complementary retrieval methods are chosen to compute this measurement. These are fine-tuned ResNet with GeM pooling [135], pre-trained (on ImageNet) AlexNet with MAC pooling [138], and ASMK [163]. More details on these methods are given in Chapter 7, Section 7.1. Finally, we perform a sanity check to show that this selection process is not significantly biased to distract only those 3 methods. This includes two additional methods, VLAD [72] and fine-tuned ResNet with R-MAC pooling by Gordo *et al.* [58]. As shown in Table 4.2, the performance on the hardest 1M distractors is hardly affected whether one of those additional methods participates or not in the selection process. This suggests that the mining process is not biased towards particular methods.

Table 4.2 also shows that the distractor set we choose (version 1M (1,2,3) in the Table) is much harder than a random 1M subset and nearly as hard as all 4M distractor images. Example images from the set $\mathcal{R}$1M are shown in Figure 4.5.

4.5. Concluding remarks

We have revisited two of the most established image retrieval datasets, that were perceived as performance saturated. To make it suitable for modern image retrieval benchmarking, we address drawbacks of the original annotation. This includes new annotation for both datasets that was created with an extra attention to the reliability of the ground truth, and an introduction of 1M hard distractor set.

Chapter 5

Improving Bag-of-Words-Based Compact Image Retrieval

This chapter tackles the problem of generating compact image representation starting from a high-dimensional bag-of-words approach [151]. In fact, methods proposed here can be successfully used in any image retrieval method based on local features that depends on generating visual vocabularies, *e.g.*, vector of locally aggregated descriptors (VLAD) [72] or Fisher vectors [124]. The contents of this chapter have been published in [131]. After this publication, convolutional-neural-network-based approaches provided compact representations leading to higher accuracy. We discuss the convolutional-neural-network-based approaches, and our respective contributions in that area, in Chapter 6.

The BoW vectors are high dimensional (up to 64 million dimensions in [102]), and as such they require inverted file structure for the efficient search of several million images on a single machine. Building a vocabulary with millions of visual words is a demanding process, making the offline preprocessing stage computationally expensive. Additionally, quantizing a query image is often a bottleneck of online stage, again due to the large vocabulary size. There are more scalable approaches that tackle memory and computational limitations problem by generating compact image representations [169, 124, 72], where the image is described by a short vector that can be additionally compressed into compact codes using binarization [169, 177], product quantization [71], or additive quantization [7] techniques. In this work we propose and experimentally evaluate simple techniques that additionally boost retrieval performance, but at the same time preserve low memory and computational costs.

Short vector image representations are often generated using the principal component analysis (PCA) [12] technique to perform the dimensionality reduction over high-dimensional vectors. Jegou and Chum [68] study the effects of PCA on BoW representations. They show that both steps of PCA procedure, *i.e.*, centering and selection of de-correlated (orthogonal) basis minimizing the dimensionality reduction error, improve retrieval performance. Centering (mean subtraction) of BoW vectors provides a boost in performance by adding a higher value to the negative evidence: given two BoW vectors, a visual word jointly missing in both vectors provides useful information for the similarity measure [68]. Additionally, the authors of [68] advocate the joint dimensionality reduction with multiple vocabularies to reduce the quantization artifacts underlying BoW and VLAD. These vocabularies are created by using different initializations for the k-means algorithm, which may produce highly correlated vocabularies.

In this chapter, we propose to reduce the redundancy of the joint vocabulary representation (before the joint dimensionality reduction) by varying parameters of the local feature descriptors prior to the k-means quantization. In particular, we propose:

(i) different sizes of measurement regions for local description, (ii) different power-law normalizations of local feature descriptors, and (iii) different linear projections (PCA learned) to reduce the dimensionality of local descriptors. In this way, created vocabularies will be more complementary and joint dimensionality reduction of concatenated BoW vectors originating from several vocabularies will carry more information. Even though the proposed approaches are simple, we show that they provide significant boosts to retrieval performance with no memory or computational overhead at the query time.

The rest of the chapter is organized as follows. Section 5.1 gives a brief overview of used datasets and evaluation protocols, as well as an overview of several methods: bag-of-words (BoW), efficient PCA dimensionality reduction of high dimensional vectors, and baseline retrieval with multiple vocabularies. Section 5.2 introduces novel methods for joint dimensionality reduction of multiple vocabularies and presents extensive experimental evaluations. Concluding remarks are given in Section 5.3.

5.1. Evaluation, background and baseline

This section gives a short overview of the evaluation datasets, background of bag-of-words (BoW) based image retrieval, and the baseline method used in [68]. Key steps, ideas, and implementation details are discussed in higher detail to help the understanding of the chapter.

5.1.1. Datasets and evaluation

Results of our methods are evaluated on the datasets that are widely used in the image retrieval area, namely, Oxford5k [125], Paris6k [126], and Holidays [69]. Also, we compare our results with other approaches evaluated on the same datasets. All of the aforementioned datasets consist of a set of query images that are used to rank the database images, and the performance is evaluated as mean average precision (mAP) [125] given the ground-truth defining which images are relevant per query. In order to evaluate the search performance on a large scale, Oxford Distractors dataset [125] is used in the combination with Oxford5k, denoted as Oxford105k. For more details on all of these datasets see Chapter 3, Section 3.1.

For the purposes of our experiments we use Paris6k as a training dataset in order to learn the visual vocabulary and projections of PCA dimensionality reduction. When evaluating our methods on Oxford5k, Oxford105k, and Holidays, we always use the data learned on Paris6k.

5.1.2. Bag-of-words image representation

First efficient image retrieval based on BoW image representation was proposed by Sivic and Zisserman [151]. They use local descriptors extracted in an image in order to construct a high-dimensional global descriptor. This procedure follows four basic steps:

1. For each image in the dataset, regions of interest are detected [98, 96] and described by an invariant descriptor which is d-dimensional. In this work we use the multi-scale Hessian-Affine [121], Harris-Affine [100], and MSER [96] detectors, followed by SIFT [92] or RootSIFT [4] descriptors. The rotation of the descriptor is either determined by the detected dominant orientation [92], or by the gravity vector assumption [121]. The descriptors are extracted from different sizes of measurement regions [96], as described in detail in Section 5.2.

2. Descriptors extracted from the training (independent) dataset (see Section 5.1.1) are clustered into k clusters using the k-means algorithm, which creates a visual vocabulary.

3. For each image in the dataset, a histogram of occurrences of visual words is computed. Different weighting schemes can be used, the most popular is inverse document frequency (*idf*), which generates a D dimensional BoW vector ($D = k$).

4. All resulting vectors are l_2 normalized, as suggested in [151], producing final global image representations used for searching.

5.1.3. Efficient PCA of high dimensional vectors

In most of the cases BoW image representations have very high number of dimensions (D can take values up to 64 million [102]). In these cases the standard PCA method (reducing D to D') computing the full covariance matrix is not efficient. The dual gram method (see Paragraph 12.1.4 in [12]) can be used to learn the first D' eigenvectors and eigenvalues. Instead of computing the $D \times D$ covariance matrix $\mathbf{C}$, the dual gram method computes the $n \times n$ matrix $\mathbf{X}^\top\mathbf{X}$, where $\mathbf{X}$ is a set of vectors used for learning, and n is the number of vectors in the set $\mathbf{X}$. Eigenvalue decomposition is performed using the Arnoldi algorithm, which iteratively computes the D' desired eigenvectors corresponding to the largest eigenvalues. This method is more efficient than the standard covariance matrix method if the number of vectors n of the training set is smaller than the number of vector dimensions D, which is usually the case in the BoW approach.

Jegou and Chum [68] analyze the effects of PCA dimensionality reduction on the BoW and VLAD vectors. They show that even though PCA successfully deals with the problem of negative evidence (higher importance of jointly missing visual words in compared BoW vectors), it ignores the problem of co-occurrences (co-occurrences lead to over-count some visual patterns when comparing two image vector representations, see [28]). In order to tackle the over-counting problem, they propose performing a whitening operation, similar to the one done in independent component analysis [33] (implicitly performed by the Mahalanobis distance), jointly with the PCA. In our experiments we use dimensionality reduction from D to D' components, as done in [68]:

1. Every image vector $\mathbf{v} = [\mathrm{v}_1, \ldots, \mathrm{v}_D]$ is post-processed using power-law normalization [124]: $\bar{\mathrm{v}}_i = |\mathrm{v}_i|^\beta \mathrm{sign}(\mathrm{v}_i)$, with $0 \leq \beta < 1$ as a fixed constant. Vector $\bar{\mathbf{v}}$ is l_2 normalized after processing. It has been shown [73] that this simple procedure reduces the impact of multiple matches and visual bursts [70]. In all our experiments $\beta = 0.5$, denoted as signed square rooting (SSR).

2. First D' eigenvectors of matrix $\mathbf{C}$ are learned using power-law normalized training vectors $\mathbf{X} = [\bar{\mathbf{v}}_1^{\mathrm{tr}}| \ldots |\bar{\mathbf{v}}_n^{\mathrm{tr}}]$, corresponding to the largest D' eigenvalues $\lambda_1, \ldots, \lambda_{D'}$.

3. Every power-law normalized image descriptor used for searching $\bar{\mathbf{v}}$ is PCA-projected, and at the same time whitened and re-normalized to a new vector $\bar{\mathbf{v}}^{(w)}$ that is the final short vector representation with dimensionality D':

$$\bar{\mathbf{v}}^{(w)} = \frac{\mathrm{diag}(\lambda_1^{-\frac{1}{2}}, \ldots, \lambda_{D'}^{-\frac{1}{2}})\mathbf{P}^\top (\bar{\mathbf{v}} - \boldsymbol{\mu})}{\left\| \mathrm{diag}(\lambda_1^{-\frac{1}{2}}, \ldots, \lambda_{D'}^{-\frac{1}{2}})\mathbf{P}^\top (\bar{\mathbf{v}} - \boldsymbol{\mu}) \right\|}, \tag{5.1}$$

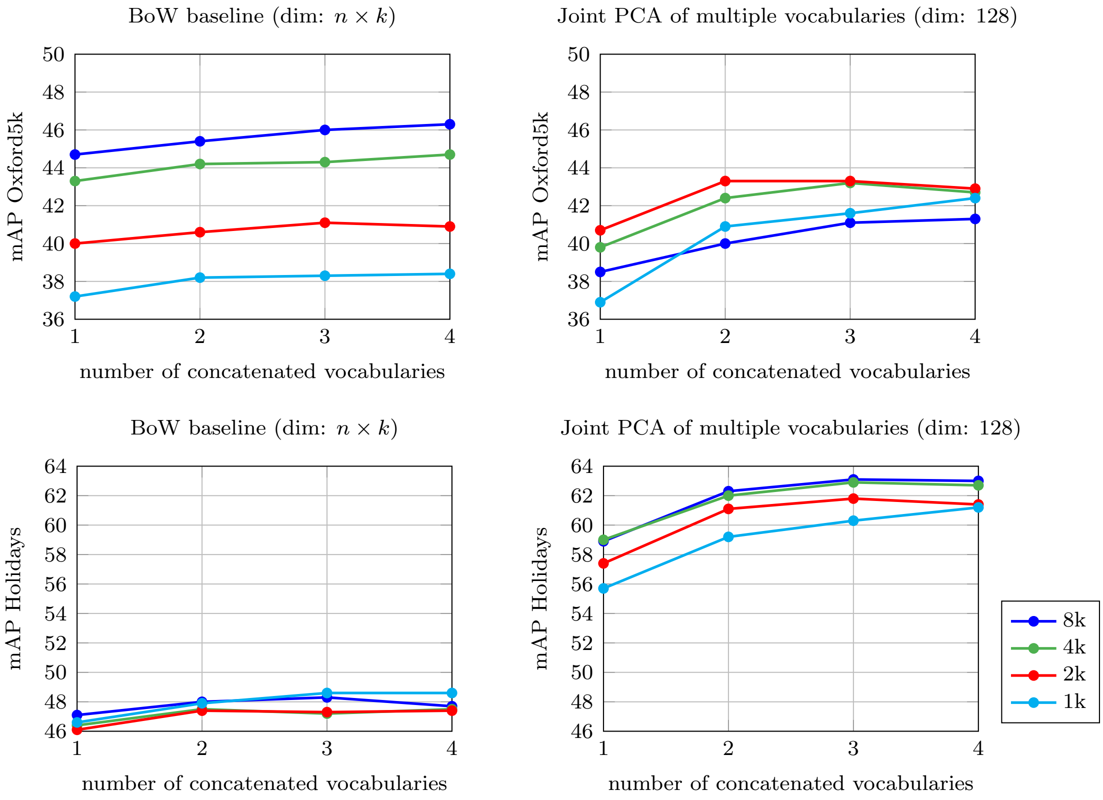


Figure 5.1. Performance evaluation of the baseline methods. Left plots show mAP performance on Oxford5k (upper plot) and Holidays (lower plot) after straightforward concatenation of BoW vectors (no PCA dimensionality reduction performed) generated using multiple vocabularies. Note that dimensionality of BoW grows linearly with every new concatenation. Right plots present mAP performance on Oxford5k and Holidays after joint PCA dimensionality reduction of concatenated BoW representations to a $D' = 128$ dimensional vector.

where $\boldsymbol{\mu}$ is the mean vector to perform centering, and the $D \times D'$ matrix $\mathbf{P}$ is formed by the largest eigenvectors calculated in the previous step. Comparing two vectors after this dimensionality reduction with the Euclidean distance is now similar to using a Mahalanobis distance. It has been argued that the re-normalization step is critical for a better comparison metric, see [68].

In order to compare results in a fair manner, we will use $D' = 128$ dimensions for all our experiments following the trend of previous research in short image representations.

5.1.4. The baseline method

This work builds upon the work [68], which is briefly reviewed in this section. In [68], a joint dimensionality reduction of multiple vocabularies is proposed. Image representation vectors are separately SSR normalized for each vocabulary, concatenated and then jointly PCA-reduced and whitened as explained in the Section 5.1.3. The *idf* term is ignored, and it is noted that the influence is limited when used with multiple vocabularies. Results of this method are shown in Figure 5.1 (right plots). Comparing to the straightforward concatenation (Figure 5.1, left plots) where the results do not noticeably improve after adding multiple vocabularies, it can be noticed that an improvement in performance is achieved even when keeping low memory requirements by using PCA dimensionality reduction. However, for some vocabularies (*i.e.* $k = 2$k), performance is dropping after only few vocabularies used.

Table 5.1. Complexity of vocabularies used throughout the experiments. Complexity is given as a number of vector comparisons per local descriptor during the construction of the final BoW image representation.

Vocabulary	Complexity
8k	8192
4k	4096
2k	2048
1k	1024
4k+2k+. . . +128	8064
2k+1k+. . . +128	3968
1k+512+256+128	1920
512+256+128	896

5.2. Sources of multiple codebooks

We propose combining multiple vocabularies that are differing not just in random initialization of clustering procedure, but also in the data used for clustering. The feature data are alternated in the process of local features description. This process is not trying to synthesize appearance deformations, but rather varying certain design choices in the pipeline of feature description, such as the relative size of the measurement region. Vocabularies created in this manner will contain less redundancy. This is combined with joint PCA dimensionality reduction (as described in Sections 5.1.3 and 5.1.4) in order to produce short-vector image representations that are used for searching the most similar images in the dataset.

Quantization complexity for all vocabularies used in experiments is given in Table 5.1. As stated in [68], time necessary to quantize 2000 local descriptors of a query image, for four k = 8k vocabularies, on 12 cores is 0.45s, using a multi-threaded exhaustive search implementation. Timings are proportional to the vocabulary size, *i.e.*, to the number in the right column of Table 5.1.

5.2.1. Multiple measurement regions

An affine invariant descriptor of an affine covariant region can be extracted from any affine covariant constructed measurement region [96]. As an example of a measurement region that is, in general, of a different shape than the detected region, is an ellipse fitted to the regions, as proposed by [170] and also used for MSERs [96]. An important parameter is the relative scale of the measurement region with respect to the scale of the detected region. Since the output of the detector is designed to be repeatable, it is usually not discriminative. To increase the discriminability of the descriptor, it is commonly extracted from area larger than the detected region. In case of [121], the relative change in the radius is $r = 3\sqrt{3}$. The larger the region, the higher discriminability of the descriptor, as long as the measurement region covers a close-to-planar surface. On the other hand, larger image patches have higher chance of hitting depth discontinuities and thus being corrupted. An example of multiple measurement regions is shown in Figure 5.2. To take the best of this trade off, we propose to construct multiple vocabularies over descriptors extracted at multiple relative scales of the measurement regions. Including lower scales leverages the disadvantages of large measurement regions, while joint dimensionality reduction eliminates the dependencies between the representations.

We consider using different sizes of measurement regions: $0.5 \times r$, $0.75 \times r$, $1 \times r$, $1.25 \times$

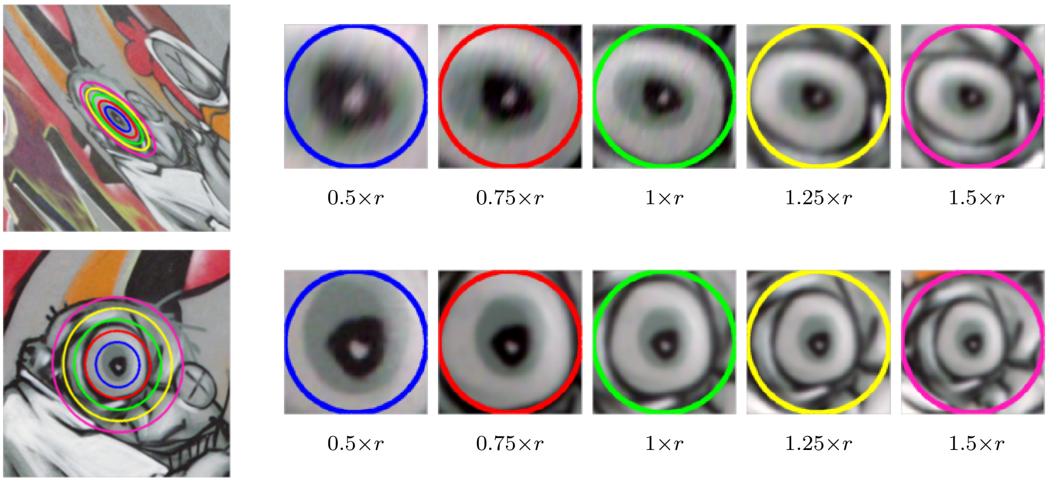


Figure 5.2. An example visualization of multiple measurement regions. A corresponding feature is detected in two images (left). Multiple measurement regions for a single detected feature are shown in each row. The normalized patches (right) show different image content described by the respective descriptor.

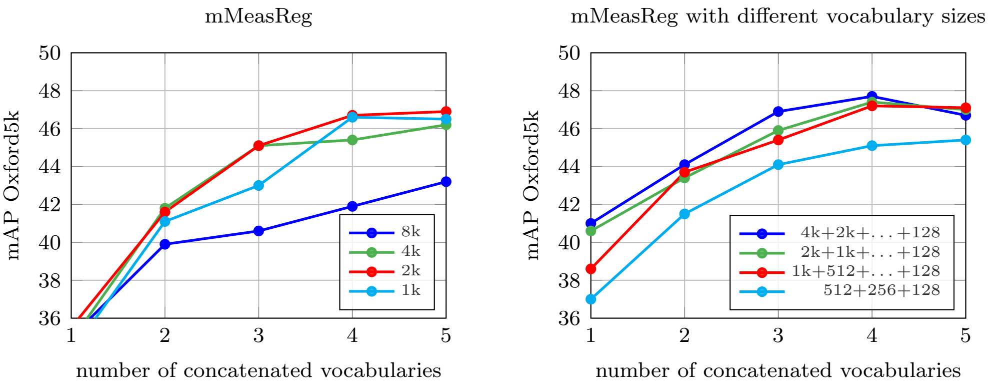


Figure 5.3. Performance evaluation of the multiple measurement regions (mMeasReg) approach. Performance improvement on Oxford5k, measured via mAP, after PCA reduction to $D' = 128$ of concatenated BoW vectors produced on vocabularies created using SIFT descriptors with different measurement regions: 0.5×r, 0.75×r, 1×r, 1.25×r, 1.5×r.

r, $1.5 \times r$; creating slightly different SIFT descriptors used to learn every vocabulary. Implementation is very simple and during online stage the computation has to be done only for the features from query image region. Though simple, this method provides significant improvement even when concatenating vocabularies of small sizes (*i.e.* $k =$ 2k and $k =$ 1k), see Figure 5.3 (left plot). We also explore the use of vocabularies with different sizes. All BoW vectors in this case are weighted proportionally to the logarithm of their vocabulary size [68]. In each step we concatenate a new bundle of vocabularies with multiple sizes, calculated with a different measurement region. We notice improvement when using multiple vocabulary sizes as well, see Figure 5.3 (right plot). For presentation of results on both plots in Figure 5.3, in every step we are adding a different vocabulary created on SIFT vectors with measurement regions in predefined order: $0.5 \times r$, $0.75 \times r$, $1 \times r$, $1.25 \times r$, $1.5 \times r$. This approach is denoted as *mMeasReg*.

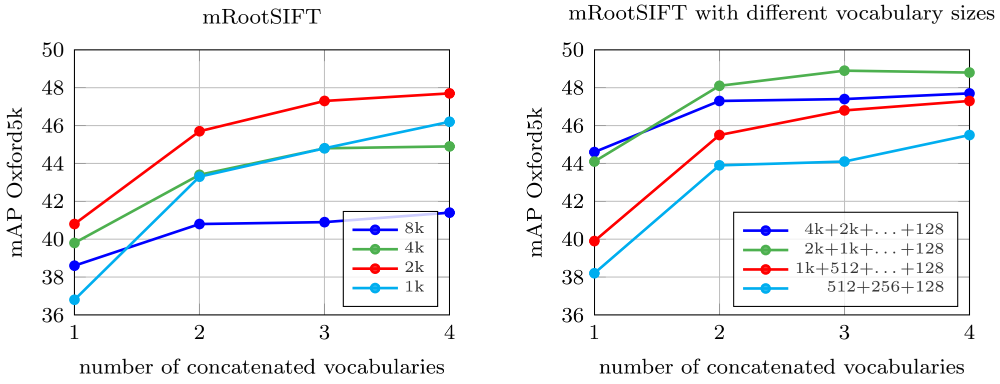


Figure 5.4. Performance evaluation of the multiple power-law normalized SIFT descriptors (mRootSIFT) approach. Performance improvement on Oxford5k, measured via mAP, after PCA reduction to $D' = 128$ of concatenated BoW vectors produced on vocabularies created using multiple local feature descriptors: SIFT, SIFT$^{0.4}$, SIFT$^{0.5}$, SIFT$^{0.6}$.

5.2.2. Multiple power-law normalized SIFT descriptors

SIFT descriptors [92] were the popular choice in most of the image retrieval systems for a long time. Arandjelovic *et al.* [4] show that using a Hellinger kernel instead of standard Euclidean distance to measure the similarity between SIFT descriptors leads to a noticeable performance boost in retrieval system. The Hellinger kernel for two l_1 normalized vectors, $\mathbf{x}$ and $\mathbf{y}$, is defined as:

$$H(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{D} \sqrt{x_i y_i}. \tag{5.2}$$

In the case of SIFT descriptors, the kernel is implemented by a simple two step procedure: (i) SIFT vector is l_1 normalized (originally it has unit l_2 norm); (ii) each element of SIFT vector is square rooted, which in turn makes the resulting vector l_2 normalized again. Resulting vector is denoted as RootSIFT [4]. Using Euclidean distance on RootSIFT descriptors will give the same result as using Hellinger kernel on the original SIFT descriptors:

$$||\sqrt{\mathbf{x}} - \sqrt{\mathbf{y}}||^2 = 2 - 2H(\mathbf{x}, \mathbf{y}). \tag{5.3}$$

In general, a power-law normalization [124] with any power $0 \leq \beta \leq 1$ can be applied to the descriptors ($\beta = 0.5$ resulting in RootSIFT [4]). Voronoi cells constructed in power-law normalized descriptor spaces can be seen as non-linear hyper-surfaces separating the features in the original (SIFT) descriptor space. Concatenation of such feature space partitionings reduces the redundant information.

There is no additional memory required and the change can be done on-the-fly with virtually no additional computational cost using simple power operation. We consider building four different vocabularies using: SIFT and SIFT with every component to the power of 0.4, 0.5, 0.6 (denoted as SIFT$^{0.4}$, SIFT$^{0.5}$, SIFT$^{0.6}$ respectively). Performance improves when concatenating vocabularies created using different power-law normalized SIFT descriptors, see Figure 5.4, left plot. Additional improvement can be achieved by using a bundle of vocabularies with different sizes, see Figure 5.4, right plot. Adding all SIFT modifications to the process of vocabulary creation achieves noticeable improvement of retrieval performance in the case of all vocabulary sizes. We denote this method as *mRootSIFT*.

Combining vocabularies of different SIFT exponents improves over combining different vocabularies of a single SIFT exponent. For example, for 4 × 2k vocabularies, the mAP on Oxford5k is 46.5 for 4 × SIFT$^{0.5}$, and 47.7 (Figure 5.4 left) for exponent combination.

5.2.3. Multiple linear projections of SIFT descriptors

In locality sensitive hashing (random) linear projections are commonly used to reduce the dimensionality of the space while preserving locality. The idea pursued in this part of the chapter is to use linear projections on the feature descriptors (SIFTs) before the vocabulary construction via k-means. However, random projections do not reflect the structure of the descriptors, resulting in noisy descriptor space partitionings. We propose to use PCA learned linear projections of SIFTs, learned on different training sets or subsets. The projections learned this way account for the statistics given by the training sets and hence produce meaningful distances, while inserting different biases into the vocabulary construction.

The improvement is twofold: (i) increased performance measured by mAP, and (ii) shorter query quantization time due to more compact local descriptors after dimensionality reduction. On the other side there is a small amount of storage required to save learned projection matrices for every vocabulary, which we reuse at query. We consider and evaluate three different approaches for learning the eigenvectors used to project SIFT vectors from D to D' dimensions:

1. Eigenvectors are learned on Paris6k dataset and reduce the 128-dimensional SIFT descriptors to $D' = 80, 64, 48, 32$ in the respective order for every newly created vocabulary ($mPCA_1$-$SIFT$). Results of this experiment are shown in Figure 5.5, 1st row.
2. Eigenvectors are learned on different datasets: Paris6k, Holidays, University of Kentucky benchmark (UKB) [113], PASCAL VOC 2007 training set [48], in the respective order, for every newly created vocabulary ($mPCA_2$-$SIFT$). Dimension of SIFT descriptors is reduced to $D' = 80$ in all cases. For the mAP performance on Oxford5k, see Figure 5.5, 2nd row.
3. Eigenvectors are learned on different datasets: Paris5k, Holidays, UKB, PASCAL VOC 2007 training set, and reduce the dimension of SIFT descriptors differently for each dataset ($D' = 80, 64, 48, 32$ respectively) creating different vocabularies ($mPCA_3$-$SIFT$). Performance is presented in Figure 5.5, 3rd row.

Note that first vocabulary in all three different approaches is produced using standard SIFT descriptors without PCA reduction. A new vocabulary is added in every step of the experiment having joint dimensionality reduction of 5 concatenated BoW vectors in the end.

5.2.4. Multiple feature detectors

In the Video Google approach [151] the authors combine vocabularies created from two different feature types. In this work we attempt to combine Hessian-Affine [121] and MSER [96] detectors. Even though straightforward concatenation of BoW vectors created on $k = 8$k vocabularies (48.7 mAP on Oxford5k) gives improvement over using single BoW representations with Hessian-Affine (44.7) and MSER (40.1) features, after joint PCA reduction there is a decrease of performance when combining features (37.0 mAP on Oxford5k) compared to only doing PCA reduction on a single Hessian-Affine vocabulary (38.6), and an increase in performance when compared to PCA-reduced

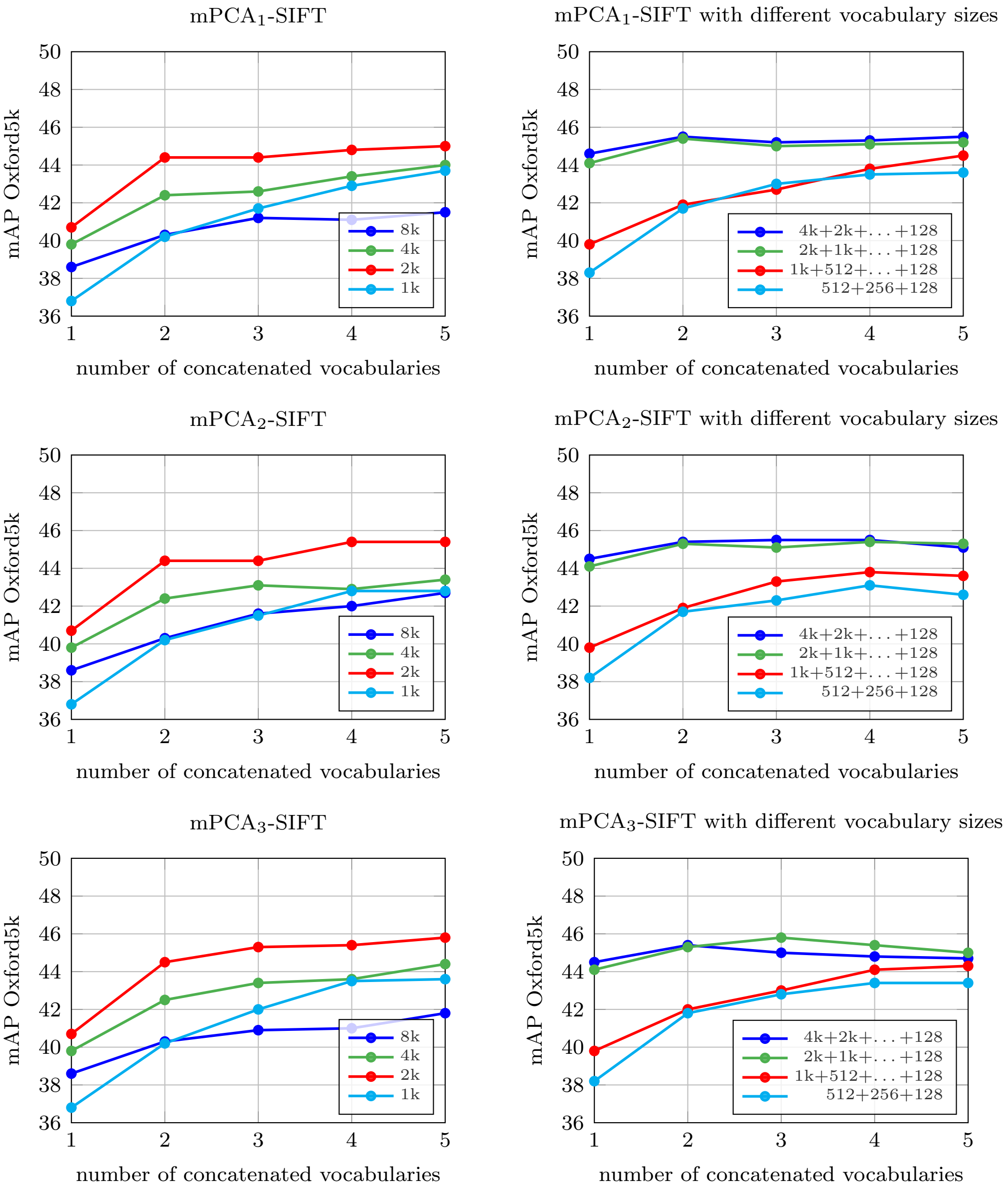


Figure 5.5. Performance evaluation of the multiple linear projections of SIFT descriptors (mPCA-SIFT) approach. Performance improvement on Oxford5k, measured via mAP, after PCA reduction to $D' = 128$ of concatenated BoW vectors produced on vocabularies created using different PCA-reduced SIFT descriptors. For more details about all three presented methods see Section 5.2.

BoW vectors built on a single MSER vocabulary (24.4). Similar conclusions are made when combining smaller vocabulary sizes, *i.e.*, there is always a drop in performance when comparing PCA reduction on a single vocabulary with Hessian-Affine features and PCA on combined vocabularies with Hessian-Affine and MSER features; mAP drop: from 39.8 to 39.1, from 40.7 to 38.7, from 36.8 to 35.1 for k = 4k, 2k, 1k respectively. We also experimented with combining Harris-Affine [100] with Hessian-Affine features in the same manner as with MSER, but the improvement is not significant. PCA reduction of a single k = 8k vocabulary on Hessian-Affine yields 38.6 mAP on Oxford5k while joint PCA after adding a vocabulary of the same size built on Harris-Affine improves mAP to 39.0, which is smaller improvement than using two vocabularies built on Hessian-Affine features with different randomization (40.0 mAP).

Table 5.2. Comparison with the state-of-the-art on short vector image representation based on local features. Dimensionality of all presented methods is fixed to 128. Results in the first section of the table are mostly obtained from [73], except for the recent method on triangulation embedding and democratic aggregation with rotation and normalization ($\phi_\Delta+\psi_\mathrm{d}$+RN) proposed in [74]. In the second section we present results from methods that are using joint PCA and whitening of high dimensional vectors as we do. Results marked with † are obtained after our reimplementation of the methods using feature detector and descriptor as described in Section 5.1.2 and Paris6k as a learning dataset. In the last section of the table we present results of our methods, which are marked with $\star$. Previous state of the art is highlighted in **bold**, new state of the art in **red outline**. Best viewed in color.

Method	Vocabulary	Oxford5k	Oxford105k	Holidays
BoW [151]	k=20k	19.4	–	45.2
Improved Fisher [124]	k=64	30.1	–	56.5
VLAD [72]	k=64	–	–	51.0
VLAD+SSR [73]	k=64	28.7	–	55.7
$\phi_\Delta+\psi_\mathrm{d}$+RN [74]	k=16	43.3	35.3	61.7
mVocab/BoW [68]†	k=4×8k	41.4	33.2	63.0
mVocab/BoW [68]†	k=2×(32k+...+128)	42.9	35.1	**64.5**
mVocab/VLAD [68]	k=4×256	–	–	61.4
mVocab/VLAD+adapt+innorm [5]	k=4×256	**44.8**	**37.4**	62.5
$\star$ mMeasReg/mVocab/BoW	k=5×2k	46.9	38.9	66.9
$\star$ mMeasReg/mVocab/BoW	k=4×(4k+...+128)	47.7	39.2	**67.3**
$\star$ mRootSIFT/mVocab/BoW	k=4×2k	47.7	39.8	64.3
$\star$ mRootSIFT/mVocab/BoW	k=4×(2k+...+128)	**48.8**	**41.4**	65.6
$\star$ mPCA$_3$-SIFT/mVocab/BoW	k=5×2k	45.8	38.1	63.2
$\star$ mPCA$_1$-SIFT/mVocab/BoW	k=5×(4k+...+128)	45.5	37.8	64.6

5.2.5. Effective vocabulary size

In order to better understand the impact of using multiple vocabularies we count the number of unique assignments in the product vocabulary. It corresponds to the number of non-empty cells of the descriptor space generated by all vocabularies simultaneously. The maximum possible number of unique assignments is equal to the product of number of clusters (cells) of all joint vocabularies. The number is related to the precision of reconstruction of each feature descriptor from its visual word assignments. For combination of vocabularies with different SIFT exponents (mRootSIFT) the number of unique assignments for Oxford5k dataset is shown in Figure 5.6. The plots are similar for all vocabulary combinations.

5.2.6. Comparison with the state-of-the-art

Comparison with the current methods dealing with short vector image representation based on local features is given in Table 5.2. Authors of the baseline approach on multiple vocabularies (mVocab) did not provide results for Oxford5k and Oxford105k datasets using all of their proposed methods, so we reimplemented and presented the corresponding results. Compared to their best method on Oxford5k that achieves 42.9 mAP, our best method (48.8 mAP) obtains significant relative improvement of 13.8%. In fact, all our methods outperform mVocab baseline methods on Oxford5k by a noticeable margin, with an improvement of 6.1% in the case of our worst performing method. When evaluating large-scale retrieval on Oxford105k dataset our methods again outperform the baseline method, relative improvement is 17.9% for our best performing

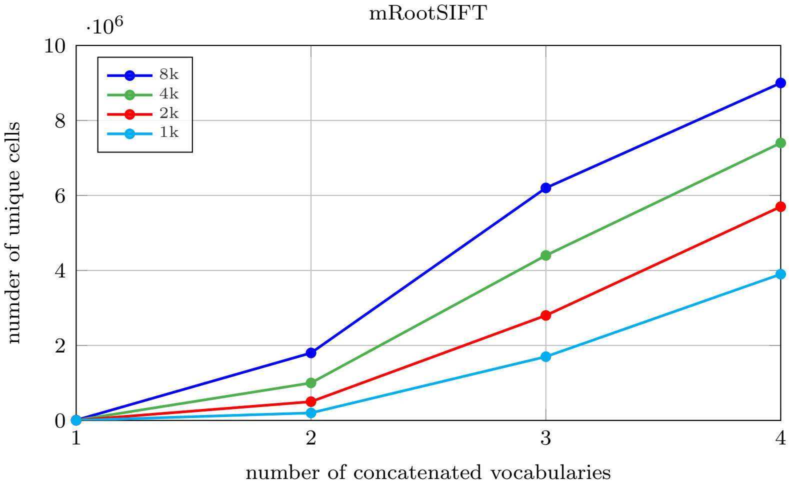


Figure 5.6. Number of unique assignments (vocabulary cells) for Oxford5k dataset when combining vocabularies built on multiple power-law normalized SIFT descriptors (mRootSIFT): SIFT, SIFT$^{0.4}$, SIFT$^{0.5}$, SIFT$^{0.6}$.

method, and 7.7% for the worst performing one. In order to make a fair comparison when evaluating on Holidays dataset we again reimplemented the baseline approach, using Paris6k for learning the vocabularies and PCA projections (as we did in all our methods). In this case, the relative improvement is 4.3% with our best method (from 64.5 mAP to 67.3 mAP). We also compare our methods to two recent state-of-the-art approaches on short representations [5, 74]. On Oxford5k and Oxford105k we improve as much as 8.9% and 10.7%, respectively, compared to VLAD based approach [5], and 12.7% and 17.3%, respectively, compared to T-embedding based approach [74]. On Holidays dataset relative improvement is 7.7% compared to the former and 9.1% compared to the latter. Note that the dataset used for learning of the meta-data for Holidays is different: we use Paris6k, while both [5] and [74] are using an independent dataset comprising of 60k images downloaded from Flickr.

5.2.7. Discussion

Even though this chapter is dedicated to packing bag of words, we apply the same idea to pooled activations of convolutional neural networks for comparison. All used convolutional neural networks were trained on the image classification task. Final representation is always PCA-reduced to $D' = 128$, whitened, and evaluated on Oxford5k dataset. AlexNet [81] with max and sum pooling of its last convolutional activations achieves 40.1 and 43.7 mAP, respectively, while VGG [150] with max and sum pooling achieves 50.9 and 52.6 mAP, respectively. Concatenating AlexNet max-pooled and sum-pooled representations and jointly PCA-reducing them achieves 44.8, while the same experiment with VGG obtains 54.7 mAP. Finally, concatenating all four combinations, *i.e.*, both AlexNet and VGG with sum and max-pooling, obtains 54.3 mAP. We conclude that joint dimensionality reduction of representations originating from different convolutional-neural-network sources brings improvements, as well.

5.3. Concluding remarks

Methods for multiple vocabulary construction were studied and evaluated in this chapter. Following [68], the concatenated BoW image representations from multiple vocabularies were subject to joint dimensionality reduction to 128-dimensional descriptors. We have experimentally shown that generating diverse multiple vocabularies has crucial impact on search performance. Each of the multiple vocabularies was learned on local feature descriptors obtained with varying parameter settings. That includes feature descriptors extracted from measurement regions of different scales, different power-law normalizations of the SIFT descriptors, and applying different linear projections to feature descriptors prior to k-means quantization. The proposed vocabulary constructions improve performance over the baseline method [68], where only different initializations were used to produce multiple vocabularies. More importantly, *all* of the proposed methods exceed the state-of-the-art results [5, 74] by a large margin. The choice of the optimal combination of vocabularies to combine still remains an open problem.

Chapter 6

Training Convolutional Neural Networks for Image Retrieval

Convolutional neural networks (CNNs) achieve state-of-the-art performance in many computer vision tasks. However, this achievement is preceded by extreme manual annotation in order to perform either training from scratch or fine-tuning for the target task. An approach to train CNNs for image retrieval, without any human annotations of images, is proposed in this chapter. Our proposed approach produces a high-quality compact representation for image retrieval.

Neural networks have attracted a lot of attention after the success of Krizhevsky *et al.* [81] in the image-classification task. Their success is mainly due to the use of very large annotated datasets, *e.g.* ImageNet [140]. The acquisition of the training data is a costly process of manual annotation, often prone to errors. Networks trained for image classification have shown strong adaptation abilities [6]. Specifically, using activations of CNNs, which were trained for the task of classification, as off-the-shelf image descriptors [41, 137] and adapting them for a number of tasks [54, 65, 55] have shown acceptable results. In particular, for image retrieval, a number of approaches directly use the network activations as image features and successfully perform image search [55, 138, 8, 76, 168].

Fine-tuning of the network, *i.e.* initialization by a pre-trained classification network and then training for a different task, is an alternative to a direct application of a pre-trained network. Fine-tuning significantly improves the adaptation ability [187, 116]; however, further annotation of training data is required. The first fine-tuning approach

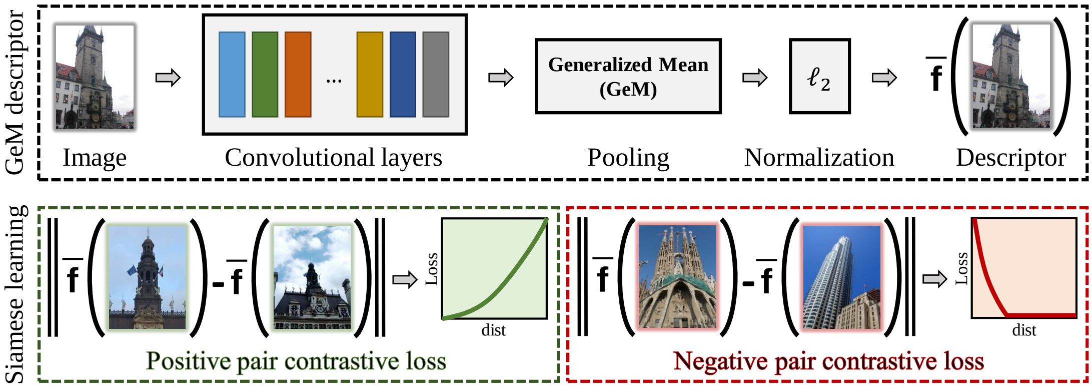


Figure 6.1. The architecture of our network with the contrastive loss used at training time. A single vector $\bar{\mathbf{f}}$ is extracted to represent an image.

for image retrieval is proposed by Babenko *et al.* [9], in which a significant amount of manual effort was required to collect images and label them as specific building classes. The approach of Babenko *et al.* [9] improved retrieval accuracy; however, their formulation is much closer to classification than to the desired properties of instance retrieval. In another approach, Arandjelovic *et al.* [3] perform fine-tuning guided by geo-tagged image databases and, similar to our work, they directly optimize the similarity measure to be used in the final task by selecting *matching* and *non-matching* pairs to perform the training.

In contrast to previous methods of training-data acquisition for image search, we dispense with the need for manually annotated data or any assumptions on the training dataset. We achieve this by exploiting the geometry and the camera positions from 3D models reconstructed automatically by a structure-from-motion (SfM) pipeline. Our state-of-the-art retrieval-SfM pipeline takes an unordered image collection as input and attempts to build all possible 3D models. To make the process efficient, fast image clustering is employed. A number of image clustering methods based on local features have been introduced [27, 178, 127]. Due to spatial verification, the *clusters* discovered by these methods are reliable. In fact, the methods provide not only clusters, but also a matching graph or sub-graph on the cluster images. The SfM filters out virtually all mismatched images and provides image-to-model matches and camera positions for all matched images in the cluster. The whole process, from unordered collection of images to detailed 3D reconstructions, is fully automatic. Finally, the 3D models guide the selection of matching and non-matching pairs. We propose to exploit the training data acquired by the same procedure in the descriptor post-processing stage to learn a discriminative whitening.

An additional contribution of this work lies in the introduction of a novel pooling layer after the convolutional layers. Previously, a number of approaches have been used. These range from fully-connected layers [9, 55], to different global-pooling layers, *e.g.* max pooling [138], average pooling [8], hybrid pooling [110], weighted average pooling [76], and regional pooling [168]. We propose a 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. Our experiments show that it offers a significant performance boost over standard non-trainable pooling layers. Our architecture is shown in Figure 6.1.

To summarize, in this chapter we address the unsupervised fine-tuning of CNNs for image retrieval. In particular, we make the following contributions: (i) We exploit SfM information and enforce, not only hard non-matching (*negative*), but also hard-matching (*positive*) examples for CNN training. This is shown to enhance the derived image representation. We show that compared to previous supervised approaches, the variability in the training data from 3D reconstructions delivers superior performance in the image-retrieval task. (ii) We show that the whitening traditionally performed on short representations [68] is, in some cases, unstable. We propose to learn the whitening through the same training data. Its effect is complementary to fine-tuning and it further boosts the performance. (iii) We propose a trainable pooling layer that generalizes existing popular pooling schemes for CNNs. It significantly improves the retrieval performance while preserving the same descriptor dimensionality. (iv) In addition, we propose an improvement of the multi-scale representation, and a novel α-weighted query expansion that is more robust compared to the standard average query expansion technique widely used for compact image representations. (v) Finally, we set a new state-of-the-art result for Oxford5k, Paris6k, and Holidays datasets by re-training the commonly used CNN architectures, such as AlexNet [81], VGG [150], and ResNet [60].

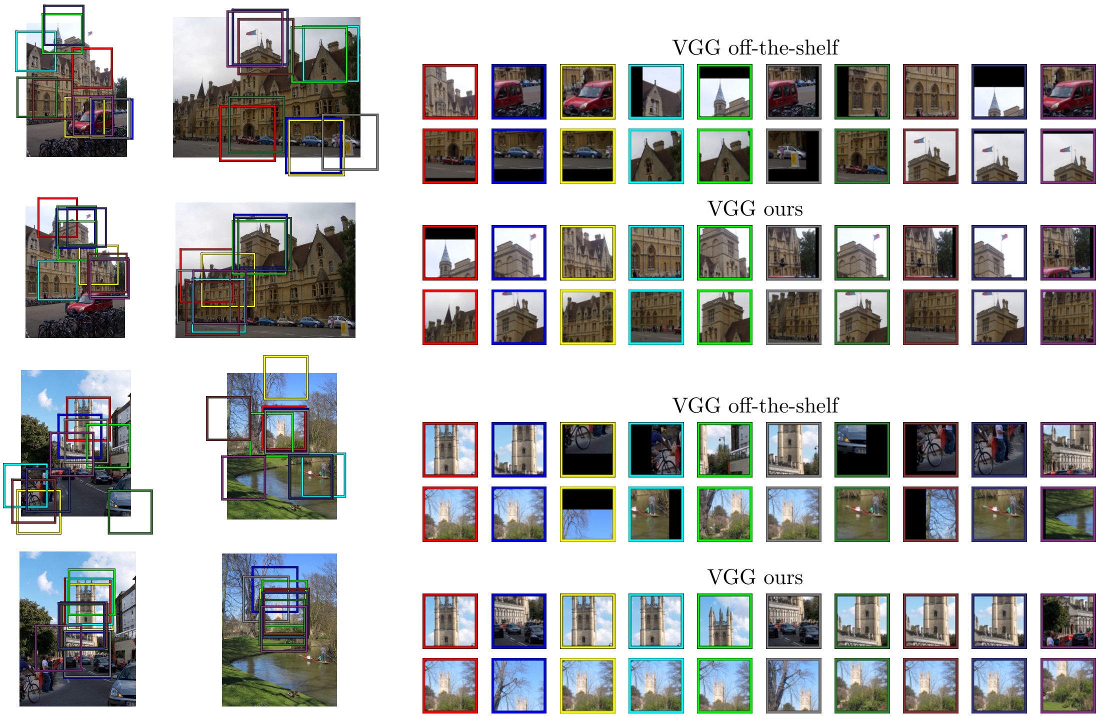


Figure 6.2. Visualization of image regions that correspond to MAC descriptor dimensions that have the highest contribution, *i.e.* large product of descriptor elements, to the pairwise image similarity. The example uses VGG before (top) and after (bottom) fine-tuning. Same color corresponds to the same descriptor component (feature map) per image pair. The patch size is equal to the receptive field of the last local pooling layer.

The work described in this chapter has been published in the following papers [146, 132, 133, 135]. More specifically, the retrieval-SfM pipeline, described in Section 6.2.1, has been published in [146, 132], while the rest has been published in [133, 135]. Training data, trained models, and code (using MATLAB/MatConvNet[1] and Python/PyTorch[2] frameworks) are publicly available[3].

The rest of the chapter is organized as follows. Our network architecture, learning procedure, and search process is presented in Section 6.1, and our proposed automatic acquisition of the training data is described in Section 6.2. Finally, in Section 6.3 we perform an extensive quantitative and qualitative evaluation of all proposed novelties with different CNN architectures and compare to the state of the art. Concluding remarks are given in Section 6.4.

6.1. Architecture, learning, search

In this section we describe the network architecture and present the proposed generalized-pooling layer. Then we explain the process of fine-tuning using the contrastive loss and a two-branch network. We describe how, after fine-tuning, we use the same training data to learn projections that appear to be an effective post-processing step. Finally, we describe the image representation, search process, and a novel query expansion scheme. Our proposed architecture is depicted in Figure 6.1.

[1] github.com/filipradenovic/cnnimageretrieval
[2] github.com/filipradenovic/cnnimageretrieval-pytorch
[3] cmp.felk.cvut.cz/cnnimageretrieval

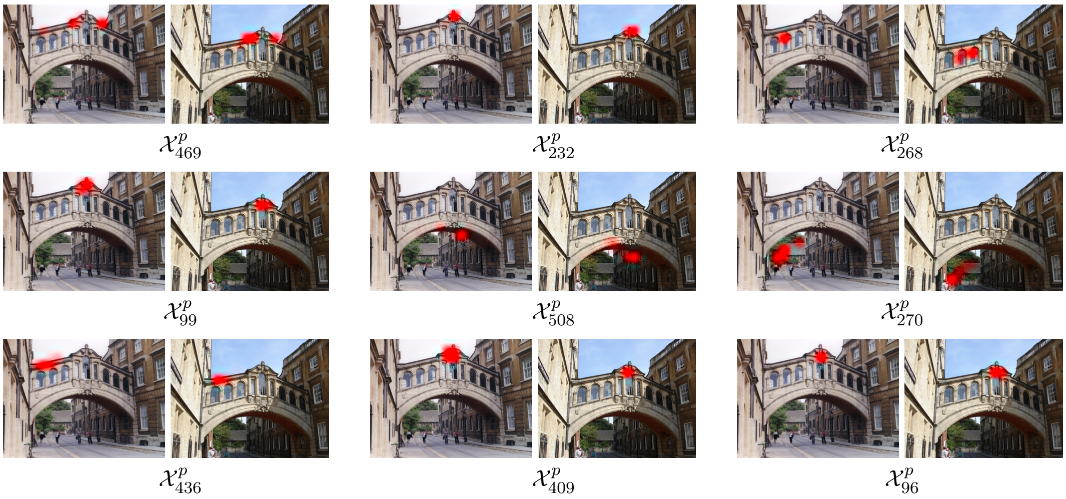


Figure 6.3. Visualization of $\mathcal{X}^p_k$ projected on the original image for a pair of query-database image. The 9 feature maps shown are the ones that score highly, *i.e.* large product of GeM descriptor components, for the database image (right) but low for the top-ranked non-matching images. The example uses fine-tuned VGG with GeM and single p for all feature maps, which converged to 2.92.

6.1.1. Fully convolutional network

Our methodology applies to any fully convolutional CNN [117]. In practice, popular CNNs for generic object recognition are adopted, such as AlexNet [81], VGG [150], or ResNet [60], while their fully-connected layers are discarded. This provides a good initialization to perform the fine-tuning.

Now, given an input image, the output is a 3D tensor $\mathcal{X}$ of $W \times H \times K$ dimensions, where K is the number of feature maps in the last layer. Let $\mathcal{X}_k$ be the set of $W \times H$ activations for feature map $k \in \{1 \dots K\}$. The network output consists of K such activation sets or 2D feature maps. We additionally assume that the very last layer is a Rectified Linear Unit (ReLU) such that $\mathcal{X}$ is non-negative.

6.1.2. Generalized-mean pooling and image descriptor

We now add a pooling layer that takes $\mathcal{X}$ as an input and produces a vector $\mathbf{f} \in \mathbb{R}^K$ as an output of the pooling process. This vector in the case of the conventional global max pooling (MAC vector [138, 168]) is given by

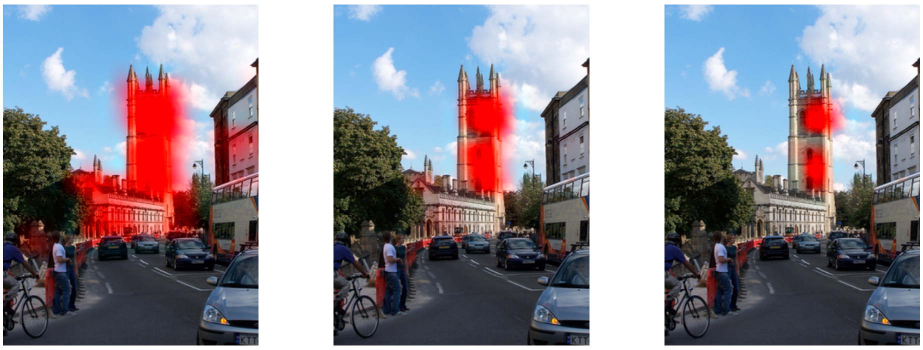
$$\mathbf{f}^{(m)} = [\mathrm{f}_1^{(m)} \dots \mathrm{f}_k^{(m)} \dots \mathrm{f}_K^{(m)}]^\top, \qquad \mathrm{f}_k^{(m)} = \max_{x \in \mathcal{X}_k} x, \tag{6.1}$$

while for average pooling (SPoC vector [8]) by

$$\mathbf{f}^{(a)} = [\mathrm{f}_1^{(a)} \dots \mathrm{f}_k^{(a)} \dots \mathrm{f}_K^{(a)}]^\top, \qquad \mathrm{f}_k^{(a)} = \frac{1}{|\mathcal{X}_k|} \sum_{x \in \mathcal{X}_k} x. \tag{6.2}$$

Instead, we exploit the generalized mean [39] and propose to use generalized-mean (GeM) pooling whose result is given by

$$\mathbf{f}^{(g)} = [\mathrm{f}_1^{(g)} \dots \mathrm{f}_k^{(g)} \dots \mathrm{f}_K^{(g)}]^\top, \quad \mathrm{f}_k^{(g)} = \left(\frac{1}{|\mathcal{X}_k|} \sum_{x \in \mathcal{X}_k} x^{p_k} \right)^{\frac{1}{p_k}}, \quad p_k \in [1, \infty] \,. \tag{6.3}$$



$p = 1$ $p = 3$ $p = 10$

Figure 6.4. Visualization of $\mathcal{X}_k^p$ projected on the original image for three different values of p. Case $p = 1$ corresponds to SPoC, and larger p corresponds to GeM before the summation of (6.3). Examples shown use the off-the-shelf VGG.

Pooling methods (6.1) and (6.2) are special cases of GeM pooling given in (6.3), *i.e.*, max pooling when $p_k \rightarrow \infty$ and average pooling for $p_k = 1$. 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. For many popular networks this is equal to 256, 512 or 2048, making it a compact image representation.

The pooling parameter p_k can be manually set or learned since this operation is differentiable and can be part of the back-propagation. The corresponding derivatives (while skipping the superscript (g) for brevity) are given by

$$\frac{\partial \mathrm{f}_k}{\partial x_i} = \frac{1}{|\mathcal{X}_k|} \mathrm{f}_k^{1-p_k} {x_i}^{p_k-1}, \tag{6.4}$$

$$\frac{\partial \mathrm{f}_k}{\partial p_k} = \frac{\mathrm{f}_k}{p_k^2} \left(\log \frac{|\mathcal{X}_k|}{\sum_{x \in \mathcal{X}_k} x^{p_k}} + p_k \frac{\sum_{x \in \mathcal{X}_k} x^{p_k} \log x}{\sum_{x \in \mathcal{X}_k} x^{p_k}} \right). \tag{6.5}$$

There is a different pooling parameter per feature map in (6.3), but it is also possible to use a shared one. In this case $p_k = p, \forall k \in \{1 \dots K\}$ and we simply denote it by p and not p_k. We examine such options in the experimental section and compare to hand-tuned and fixed parameter values.

Max pooling, in the case of MAC, retains one activation per 2D feature map. In this way, each descriptor component corresponds to an image patch equal to the receptive field. Then, pairwise image similarity is evaluated via descriptor inner product. Therefore, MAC similarity implicitly forms patch correspondences. The strength of each correspondence is given by the product of the associated descriptor components. In Figure 6.2 we show the image patches in correspondence that contribute most to the similarity. Such implicit correspondences are improved after fine-tuning. Moreover, the CNN fires less on ImageNet classes, *e.g.* cars and bicycles.

In Figure 6.4 we show how the spatial distribution of the activations is affected by the generalized mean. The larger the p the more localized the feature map responses are. Finally, in Figure 6.3 we present an example of a query and a database image matched with the fine-tuned VGG with GeM pooling layer (GeM layer in short). We show the feature maps that contribute the most into making this database image being distinguished from non-matching ones that have large similarity, too.

The last network layer is an l_2-normalization layer. In the rest of the chapter, GeM vector corresponds to the l_2-normalized vector $\bar{\mathbf{f}}$ and constitutes the image descriptor.

6.1.3. Siamese learning and loss function

We adopt a siamese architecture and train a two-branch network. Each branch is a clone of the other, meaning that they share the same parameters. The training input consists of image pairs (i, j) and labels $Y(i, j) \in \{0, 1\}$ declaring whether a pair is non-matching (label 0) or matching (label 1). We employ the contrastive loss [26] that acts on matching and non-matching pairs and is defined as

$$\mathcal{L}^{(c)}(i,j) = \begin{cases} \frac{1}{2}||\bar{\mathbf{f}}(i) - \bar{\mathbf{f}}(j)||^2, & \text{if } Y(i,j) = 1 \\ \frac{1}{2}\left(\max\{0, \tau - ||\bar{\mathbf{f}}(i) - \bar{\mathbf{f}}(j)||\}\right)^2, & \text{if } Y(i,j) = 0 \end{cases} \tag{6.6}$$

where $\bar{\mathbf{f}}(i)$ is the l_2-normalized GeM vector of image i, and τ is a margin parameter defining when non-matching pairs have large enough distance in order to be ignored by the loss. We train the network using a large number of training pairs created automatically (see Section 6.2). In contrast to other methods [175, 147, 61, 3], we find that the contrastive loss generalizes better and converges at higher performance than the triplet loss [24]

$$\mathcal{L}^{(t)}(q, m(q), n(q)) = \max\{0, ||\bar{\mathbf{f}}(q) - \bar{\mathbf{f}}(m(q))||^2 - ||\bar{\mathbf{f}}(q) - \bar{\mathbf{f}}(n(q))||^2 + \tau\}, \tag{6.7}$$

where $\bar{\mathbf{f}}(q)$, $\bar{\mathbf{f}}(m(q))$, $\bar{\mathbf{f}}(n(q))$ are the l_2-normalized GeM vectors of query image q, and its matching $m(q)$ and non-matching $n(q)$ image, and τ is a margin parameter defining zero loss when the distance between the query and the non-matching image is greater by a margin than the distance between the query and the matching image.

6.1.4. Whitening and dimensionality reduction

In this section, the post-processing of fine-tuned GeM vectors is considered. Previous methods [8, 168] use PCA of an independent set for whitening and dimensionality reduction, *i.e.* the covariance matrix of all descriptors is analyzed. For more details on PCA whitening see Chapter 5, Section 5.1.3. We propose to leverage the labeled data provided by the 3D models and use linear discriminant projections originally proposed by Mikolajczyk and Matas [97] in the context of local feature descriptors. The projection is decomposed into two parts: whitening and rotation. The whitening part is the inverse of the square-root of the intraclass (matching pairs) covariance matrix $\mathbf{C}_m^{-\frac{1}{2}}$, where

$$\mathbf{C}_m = \sum_{Y(i,j)=1} \left(\bar{\mathbf{f}}(i) - \bar{\mathbf{f}}(j)\right)\left(\bar{\mathbf{f}}(i) - \bar{\mathbf{f}}(j)\right)^\top . \tag{6.8}$$

The rotation part is the PCA of the interclass (non-matching pairs) covariance matrix in the whitened space $\text{eig}(\mathbf{C}_m^{-\frac{1}{2}}\mathbf{C}_n\mathbf{C}_m^{-\frac{1}{2}})$, where

$$\mathbf{C}_n = \sum_{Y(i,j)=0} \left(\bar{\mathbf{f}}(i) - \bar{\mathbf{f}}(j)\right)\left(\bar{\mathbf{f}}(i) - \bar{\mathbf{f}}(j)\right)^\top . \tag{6.9}$$

The linear transformation $\mathbf{P} = \mathbf{C}_m^{-\frac{1}{2}}\text{eig}(\mathbf{C}_m^{-\frac{1}{2}}\mathbf{C}_n\mathbf{C}_m^{-\frac{1}{2}})$ is then applied as $\mathbf{P}^\top(\bar{\mathbf{f}}(i) - \boldsymbol{\mu})$, where $\boldsymbol{\mu}$ is the mean GeM vector that is subtracted to perform the centering. To reduce the descriptor dimensionality from D to D' dimensions, only eigenvectors corresponding to D' largest eigenvalues are used. Projected vectors are subsequently l_2-normalized.

Our approach uses all available training pairs efficiently in the optimization of the whitening. It is not optimized in an end-to-end manner and it is performed without using batches of training data. We first optimize the GeM descriptor and then optimize the whitening.

The described approach acts as a post-processing step, once the fine-tuning of the CNN is finished. We additionally compare with the end-to-end learning of the whitening. Whitening consists of vector shifting and projection which is modeled in a straightforward manner by a fully connected layer[4]. The results favor our approach and are discussed in the experimental section.

6.1.5. Image representation and search

Once the training is finished, an image is fed to the network shown in Figure 6.1. The extracted GeM descriptor is whitened and re-normalized. This constitutes the global descriptor for an image at a single scale. Scale invariance is learned to some extent by the training samples; however, additional invariance is added by multi-scale processing during test time without any additional learning. We follow a standard approach [58] and feed the image to the network at multiple scales. The resulting descriptors are finally pooled and re-normalized, producing a multi-scale global image representation. We adopt GeM pooling for this state too, which is shown, in our experiments, consistently superior to the standard average pooling.

Image retrieval is simply performed by exhaustive Euclidean search over database descriptors *w.r.t.* the query descriptor. This is equivalent to the inner product evaluation of l_2 normalized vectors, *i.e.* vector-to-matrix multiplication, and sorting. CNN-based descriptors are shown to be highly compatible with approximate-nearest neighbor search methods, in fact, they are very compressible [58]. In order to directly evaluate the effectiveness of the learned representation, we do not consider this alternative in this work. In practice, each descriptor requires 4 bytes per dimension to be stored.

It has recently become a standard policy to combine CNN global image descriptors with simple average query expansion (AQE) [168, 76, 8, 58]. An initial query is issued by Euclidean search and AQE acts on the top-ranked n_{QE} images by average pooling of their descriptors. Herein, we argue that tuning n_{QE} to work well across different datasets is not easy. AQE corresponds to a weighted average where n_{QE} descriptors have unit weight and all the rest zero. We generalize this scheme and we propose performing weighted averaging, where the weight of the i-th ranked image is given by $(\bar{\mathbf{f}}(q)^\top \bar{\mathbf{f}}(i))^\alpha$. The similarity of each retrieved image matters. We show in our experiments that AQE is difficult to tune for datasets of different statistics, while this is not the case with the proposed approach. We refer to this approach as α-weighted query expansion (αQE). The proposed αQE reduces to AQE for $\alpha = 0$.

6.2. Training dataset

In this section we summarize the tightly-coupled bag-of-words (BoW) image-retrieval and structure-from-motion (SfM) 3D reconstruction system that is employed to automatically select our training data. Then, we describe how we use the 3D information to select harder matching pairs and hard non-matching pairs with larger variability.

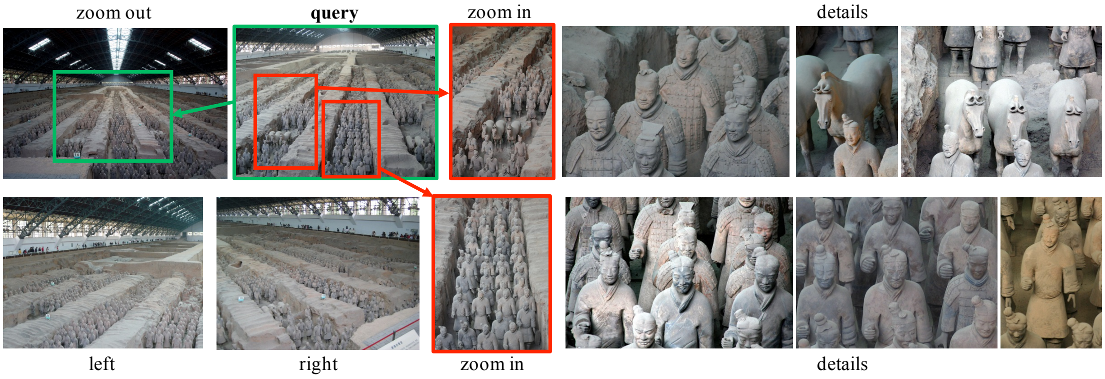


Figure 6.5. Different image retrieval mining techniques for the example query image: context of the query image (zoom out – top left), two examples of mid-level detail (zoom in), and three high-level detailed images for each of the mid-level details (rightmost). Two examples of the left and right sideways crawl of the query are shown in the bottom left.

6.2.1. BoW and 3D reconstruction

First, we detail our efficient reconstruction of 3D models contained in a given image database. The retrieval engine used here builds upon BoW with fast spatial verification [125]. It uses Hessian affine local features [100], RootSIFT descriptors [4], and a fine vocabulary of 16M visual words [102]. Then, query images are chosen via min-hash and spatial verification, as in [27]. Image retrieval based on BoW is used to collect images of the objects/landmarks. These images serve as the initial matching graph for the succeeding SfM reconstruction, which is performed using the state-of-the-art SfM pipeline [50, 1, 145]. Different mining techniques illustrated in Figure 6.5, *e.g.* zoom in/out [101, 103], sideways crawl [146], help to build a larger and more complete model.

Clustering. To seed our iterative reconstruction process efficiently, we find independent sets of spatially overlapping images using the clustering approach by Chum *et al.* [27]. This approach first indexes all database images in a min-Hash table and then uses spatially verified hash collisions as cluster seeds. Next, an incremental query expansion [125, 31] with spatial verification extends the initial clusters with additional images of the same landmark. The nearest-neighbor images in this query expansion step then define the graph of overlapping images, the so-called scene graph. Given that query expansion is a depth first search strategy, the resulting scene graph is only sparsely connected. However, in order to achieve a successful reconstruction, SfM requires a denser scene graph than provided by the clustering method. Therefore, we first densify the scene graph as described in the following paragraph before using it in SfM. Rather than seeding the reconstruction with all images in the database, this clustering procedure reduces the number of seeds by 3 orders of magnitude.

Densification. Next, we densify the initially sparse scene graph for improved reconstruction robustness and completeness. We exploit the spatially verified image pairs and their visual word matches along with an affine model to serve as hypotheses for subsequent exhaustive feature matching and epipolar verification. From this exhaustive verification, we not only obtain a higher number of feature correspondences but we also determine additional image pairs to densify the scene graph. More importantly, be-

[4] The bias is equal to the projected mean vector used to center the data.

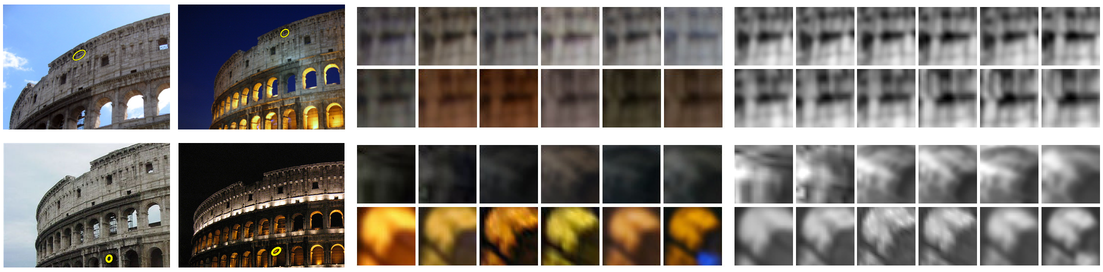


Figure 6.6. Two feature tracks containing both day and night images/features. Each row depicts two images from day and night modality, respectively, followed by a subset of feature patches depicted in two rows, one for day and one for night features, respectively. Intensity normalized patches, grayscale versions used for feature description, are shown to the right of the respective color patches. Notice the variation in lighting conditions for day and night, expressed as a significant color difference of patches. Best viewed in color.

yond the benefit of additional image pairs, the significantly increased number of feature correspondences is essential for establishing feature tracks from day to night images through dusk and dawn [132]. Only through these transitive connections, we are able to reliably register day and night images into a single 3D model. Examples of 3D point tracks that contain features from both day and night images are shown in Figure 6.6.

Structure-from-Motion. The densified scene graph is the input to the subsequent incremental SfM algorithm, which treats each edge in the graph as a putative image pair for reconstruction and attempts to reconstruct every connected component within a cluster. Connected components with less than 20 registered images are discarded. Figure 6.7 shows the SfM reconstructions for a variety of scenes.

Extension. To boost registration completeness, a final extension step issues further queries for all registered images in each reconstructed connected component. If new images are found and spatially verified, we again perform scene graph densification and use SfM to register the new views into the previously reconstructed models. While significantly increasing the size of the reconstructed models, the extension process also improves the performance of the day/night modeling. Typically, the initial set of images obtained in clustering often only contains images from one modality, *i.e.*, either day or night, even though our large-scale image database contains images of both modalities for almost all landmarks. The iterative extension overcomes this problem by incrementally growing the model from day to night or vice versa through transition images during dusk and dawn. Figure 6.8 demonstrates the improved completeness and accuracy of night models produced with our approach.

Non-overlapping 3D models. We drop redundant (overlapping) 3D models, that might have been constructed from different seeds. Models reconstructing the same landmark but from different and disjoint viewpoints are considered as non-overlapping. Finally, for each image, the estimated camera position is known, as well as the local features registered on the 3D model.

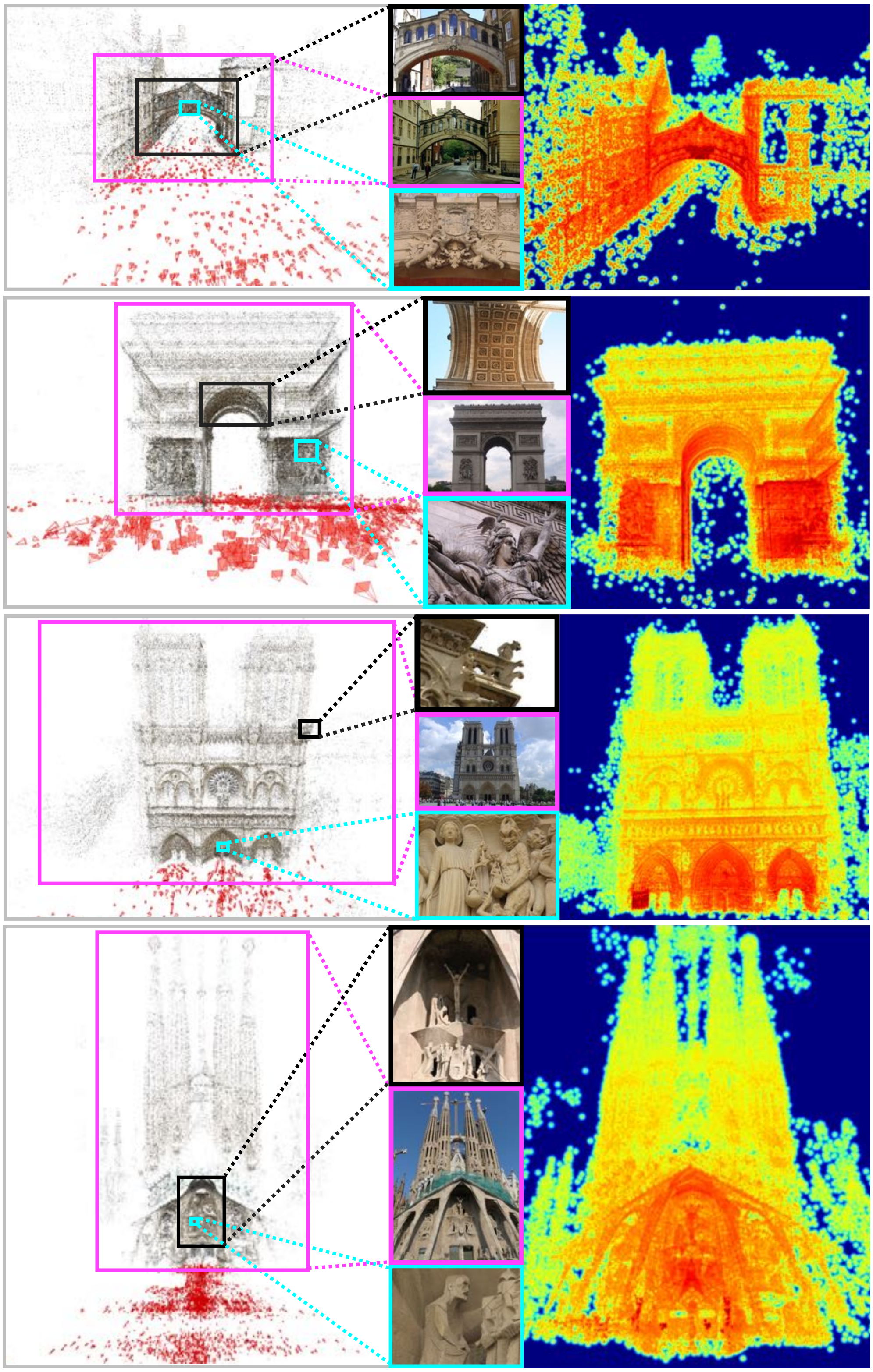


Figure 6.7. Structure-from-motion reconstructions from top to bottom: Bridge of Sighs, UK; Arc de Triomphe, France; Notre Dame, France; Sagrada Familia, Spain. Left: 3D model obtained from our retrieval and reconstruction system. Middle: registered images illustrating the range of views from overview images to images of a specific architectural detail. Right: visualization of the surface resolution from high resolution in red (approximately 1mm surface resolution) to low resolution in blue. Presented 3D models are produced by Johannes Schonberger in COLMAP (github.com/colmap/colmap).

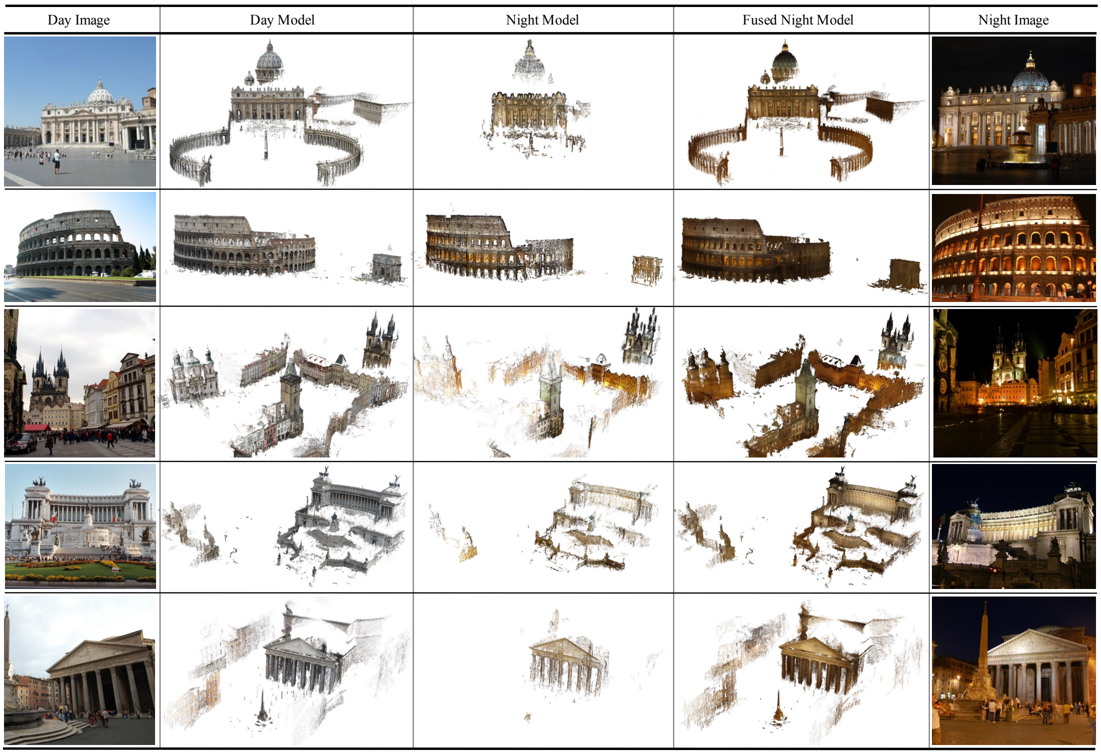


Figure 6.8. Examples of final 3D models produced by our joint retrieval and reconstruction system for: St. Peter's Basilica, Vatican; Colosseum in Rome, Italy; Astronomical Clock in Prague, Czech Republic; Altare della Patria in Rome, Italy; and Pantheon in Rome, Italy. Presented 3D models are produced by Johannes Schonberger in COLMAP (github.com/colmap/colmap).

6.2.2. Selection of training image pairs

A 3D model is described as a bipartite visibility graph $\mathbb{G} = (\mathcal{I} \cup \mathcal{P}, \mathcal{E})$ [89], where images $\mathcal{I}$ and points $\mathcal{P}$ are the vertices of the graph. The edges of this graph are defined by visibility relations between cameras and points, *i.e.* if a point $p \in \mathcal{P}$ is visible in an image $i \in \mathcal{I}$, then there exists an edge $(i, p) \in \mathcal{E}$. The set of points observed by an image i is given by

$$\mathcal{P}(i) = \{p \in \mathcal{P} : (i, p) \in \mathcal{E}\}. \tag{6.10}$$

We create a dataset of tuples $(q, m(q), \mathcal{N}(q))$, where q represents a query image, $m(q)$ is a positive image that matches the query, and $\mathcal{N}(q)$ is a set of negative images that do not match the query. These tuples are used to form training image pairs, where each tuple corresponds to $|\mathcal{N}(q)| + 1$ pairs. For a query image q, a pool $\mathcal{M}(q)$ of candidate positive images is constructed based on the camera positions in the 3D model of q. It consists of the k images with camera centers closest to the query. Due to the wide range of camera orientations, these do not necessarily depict the same object. We therefore compare three different ways to select the positive image. The positive examples are fixed during the whole training process for all three strategies.

Positive images: CNN descriptor distance. The image that has the lowest descriptor distance to the query is chosen as positive, formally

$$m_1(q) = \operatorname*{argmin}_{i \in \mathcal{M}(q)} ||\bar{\mathbf{f}}(q) - \bar{\mathbf{f}}(i)||. \tag{6.11}$$

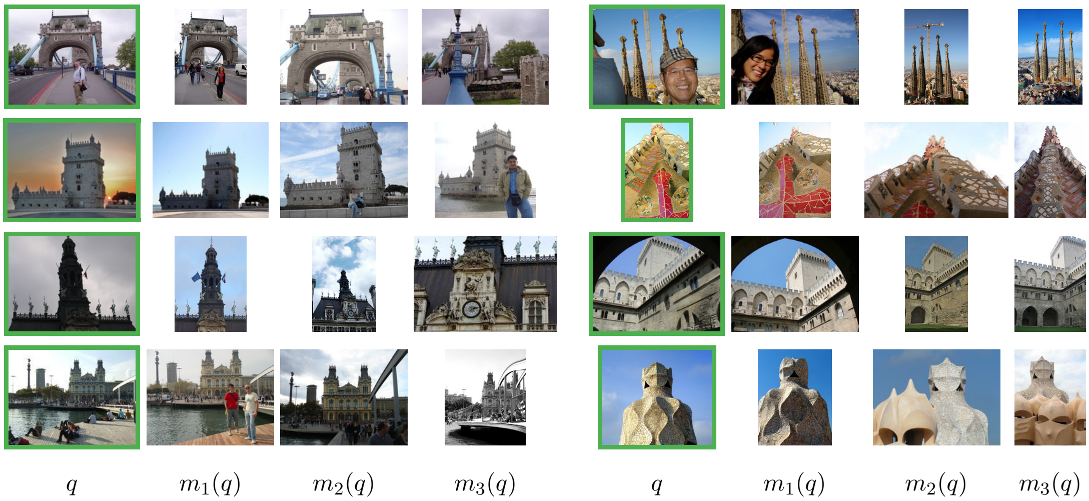


Figure 6.9. Examples of training query images (green border) and matching images selected as positive examples by methods: $m_1(q)$ – the most similar image based on the current network; $m_2(q)$ – the most similar image based on the BoW representation; and our proposed $m_3(q)$ – a hard image depicting the same object.

This strategy is similar to the one followed by Arandjelovic *et al.* [3]. Authors of [3] adopt this choice since only GPS coordinates are available and not camera orientations. As a consequence, the chosen matching images already have small descriptor distance and, therefore, small loss too. The network is thus not forced to drastically change/learn by the matching examples, which is the drawback of this approach.

Positive images: maximum inliers. In this approach, the 3D information is exploited to choose the positive image, independently of the CNN descriptor. In particular, the image that has the highest number of co-observed 3D points with the query is chosen. That is,

$$m_2(q) = \operatorname*{argmax}_{i \in \mathcal{M}(q)} |\mathcal{P}(q) \cap \mathcal{P}(i)|. \tag{6.12}$$

This measure corresponds to the number of spatially verified features between two images, a measure commonly used for ranking in BoW-based retrieval. As this choice is independent of the CNN representation, it delivers more challenging positive examples.

Positive images: relaxed inliers. Even though both previous methods choose positive images depicting the same object as the query, the variance of viewpoints is limited. Instead of using a pool of images with similar camera position, the positive example is selected at random from a set of images that co-observe sufficient number of points with the query, but do not exhibit too extreme of a scale change. We want to have the scale change that is large but can still be captured by the receptive field of the network, otherwise, the network cannot learn from it. The positive example in this case is

$$m_3(q) = \texttt{rnd}\left\{ i \in \mathcal{M}(q) : \frac{|\mathcal{P}(i) \cap \mathcal{P}(q)|}{|\mathcal{P}(q)|} \geq t_i,\ \texttt{scale}(i,q) \leq t_s \right\}, \tag{6.13}$$

where $\texttt{scale}(i,q)$ is the scale change between the two images. This method results in selecting harder matching examples that are still guaranteed to depict the same object. Method m_3 chooses different image than m_1 on 86.5% of the queries. In Figure 6.9 we present examples of query images and the corresponding positives selected with the three different methods. The relaxed method increases the variability of viewpoints.

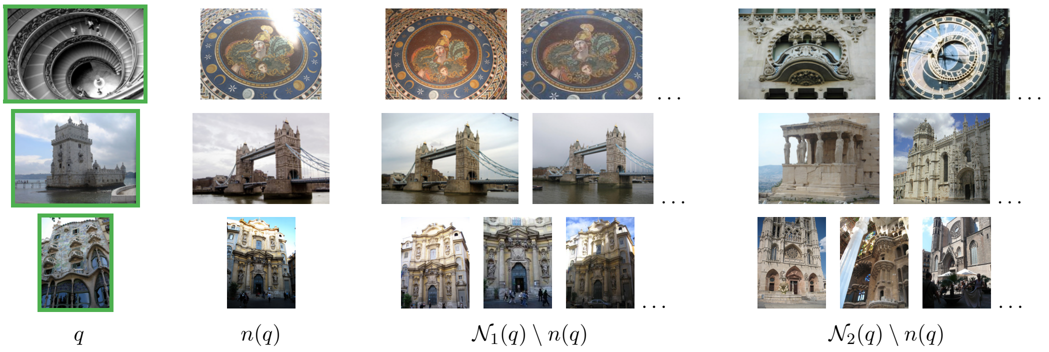

q
$n(q)$
$\mathcal{N}_1(q) \setminus n(q)$
$\mathcal{N}_2(q) \setminus n(q)$
...

Figure 6.10. Examples of training query q (one per row shown in green border), and their corresponding negatives chosen by different strategies. We show the hardest non-matching image $n(q)$, and the additional non-matching images selected as negative examples by $\mathcal{N}_1(q)$ and our method $\mathcal{N}_2(q)$. The former chooses k-nearest neighbors among all non-matching images, while the latter chooses k-nearest neighbors but with at most one image per 3D model.

Negative images. Negative examples are selected from 3D models different than the model of the query image, as the models are non-overlaping. We choose hard negatives [149, 54], that is, non-matching images with the most similar descriptor. Two different strategies are proposed: In the first, $\mathcal{N}_1(q)$, k-nearest neighbors from all non-matching images are selected. In the second, $\mathcal{N}_2(q)$, the same criterion is used, but at most one image per 3D model is allowed. While $\mathcal{N}_1(q)$ often leads to multiple, and very similar, instances of the same object, $\mathcal{N}_2(q)$ provides higher variability of the negative examples, see Figure 6.10. While positives examples are fixed during the whole training process, hard negatives depend on the current CNN parameters and are re-mined multiple times per epoch.

6.3. Experiments

In this section we discuss implementation details of our training, evaluate different components of our method, and compare to the state of the art.

6.3.1. Training setup and implementation details

Structure-from-Motion. Our training samples are derived from a generic dataset, which consists of 7.4 million images downloaded from Flickr using keywords of popular landmarks, cities and countries across the world. The clustering procedure [27] gives around 20k images to serve as query seeds. The extensive retrieval-SfM reconstruction of the whole dataset results in $1,474$ reconstructed 3D models. Removing overlapping models leaves us with 713 3D models containing more than 163k unique images from the initial dataset. The initial dataset contains, on purpose, all images of Oxford5k and Paris6k datasets. In this way, we are able to exclude 98 3D models that contain any image (or their near duplicates) from these test datasets.

Training pairs. The size of the 3D models varies from 25 to 11k images. We randomly select 551 models (around 133k images) for training and 162 (around $30k$ images) for validation. The number of training queries per 3D model is 10% of its size and limited to be less or equal to 30. Around $6,000$ and $1,700$ images are selected for training and validation queries per epoch, respectively.

Each training and validation tuple contains 1 query, 1 positive and 5 negative images. The pool of candidate positives consists of $k = 100$ images with the closest camera centers to the query. In particular, for method m_3, the inlier-overlap threshold is $t_i = 0.2$, and the scale-change threshold $t_s = 1.5$. Hard negatives are re-mined 3 times per epoch, *i.e.* roughly every $2,000$ training queries. Given the chosen queries and the chosen positives, we further add 20 images per model to serve as candidate negatives during re-mining. This constitutes a training set of around 22k images per epoch when all the training 3D models are used. The query-tuple selection process is repeated every epoch. This slightly improves the results.

Learning configuration. To perform the fine-tuning as described in Section 6.1, we initialize by the convolutional layers of AlexNet [81], VGG16 [150], or ResNet101 [60]. AlexNet is trained using stochastic gradient descent (SGD), while training of VGG and ResNet is more stable with Adam [79]. We use initial learning rate equal to $l_0 = 10^{-3}$ for SGD, initial stepsize equal to $l_0 = 10^{-6}$ for Adam, an exponential decay $l_0 \exp(-0.1i)$ over epoch i, momentum 0.9, weight decay 5×10^{-4}, margin τ for contrastive loss 0.7 for AlexNet, 0.75 for VGG, and 0.85 for ResNet, justified by the increase in the dimensionality of the embedding, and a batch size of 5 training tuples. All training images are resized to a maximum size of 362×362, while keeping the original aspect ratio. Training is done for at most 30 epochs and the best network is selected based on performance, measured via mean Average Precision (mAP) [125], on validation tuples. Fine-tuning of VGG for one epoch takes around 2 hours on a single TITAN X (Maxwell) GPU with 12 GB of memory.

We overcome GPU memory limitations by associating each query to a tuple, *i.e.*, query plus 6 images (5 positive and 1 negative). Moreover, the whole tuple is processed in the same batch. Therefore, we feed 7 images to the network, which represents 6 pairs. In a naive approach, when the query image is different for each pair, 6 pairs require 12 images.

6.3.2. Test datasets and evaluation protocol

Test datasets. We evaluate our approach on Oxford5k [125], Paris6k [126] and Holidays[5] [69] datasets. The first two are closer to our training data, while the last is differentiated by containing similar scenes and not only man-made objects or buildings. These are also combined with 100k distractors from Oxford Distractors dataset [125] to allow for evaluation at larger scale. The performance is measured via mAP. We follow the standard evaluation protocol for Oxford5k and Paris6k and crop the query images with the provided bounding box. The cropped image is fed as input to the CNN. For more details on these test datasets and evaluation protocols see Chapter 3, Section 3.1. Further evaluation of our approach on the benchmark proposed in Chapter 4 is given in Chapter 7.

Single-scale evaluation. The dimensionality of the images fed into the CNN is limited to 1024×1024 pixels. In our experiments, no vector post-processing is applied if not otherwise stated.

[5] We use the up-right version of Holidays dataset where images are manually rotated so that depicted objects are up-right. This makes us directly comparable to [58]. A different version of up-right Holidays is used in our earlier work [133], where EXIF metadata is used to rotate the images.

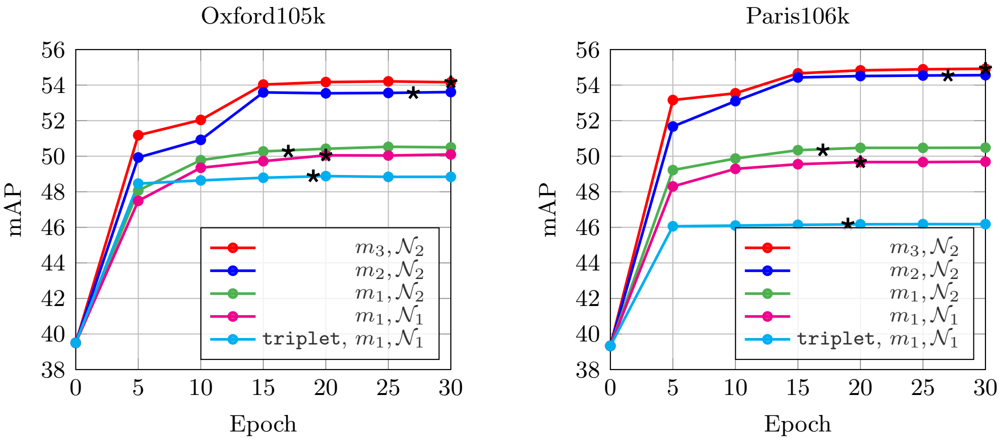


Figure 6.11. Performance comparison of methods for positive and negative example selection. Evaluation is performed with AlexNet MAC on Oxford105k and Paris106k datasets. The plot shows the evolution of mAP with the number of training epochs. Epoch 0 corresponds to the off-the-shelf network. All approaches use the contrastive loss, except if otherwise stated. The network with the best performance on the validation set is marked with $\star$.

Multi-scale evaluation. Multi-scale representation is only used during test time. We resize the input image to different sizes, then feed multiple input images to the network, and finally combine the global descriptors from multiple scales into a single descriptor. We compare the baseline average pooling [58] with our generalized mean whose pooling parameter is equal to the value learned in the global pooling layer of the network. In this case, the whitening is learned on the final multi-scale image descriptors. In our experiments, a single-scale evaluation is used if not otherwise stated.

6.3.3. Results on image retrieval

Learning. We evaluate the off-the-shelf CNN and our fine-tuned ones after different number of training epochs. The different methods for positive and negative selection are evaluated independently in order to isolate the benefit of each one. Finally, we also perform a comparison with the triplet loss [3], trained on the same training data as the contrastive loss. Note that a triplet forms two pairs. Results are presented in Figure 6.11. The results show that positive examples with larger viewpoint variability and negative examples with higher content variability acquire a consistent increase in the performance. The triplet loss[6] appears to be inferior in our context; we observe oscillation of the error in the validation set from early epochs, which implies over-fitting. In the rest of the chapter, we adopt the m_3,$\mathcal{N}_2$ approach.

Dataset variability. We perform fine-tuning by using a subset of the available 3D models. Results are presented in Figure 6.12 with 10, 100 and 551 (all available) 3D models, while keeping the amount of training data, *i.e.* number of training queries, fixed. In the case of 10 and 100 models, we use the largest ones. It is better to train with all 3D models due to the resulting higher variability in the training set. Remarkably, significant increase in performance is achieved even with 10 or 100 models. However, the network is able to over-fit in the case of few models. In the rest of our experiments we use all 551 3D models for training.

[6] The margin parameter for the triplet loss is set equal to 0.1 [3].

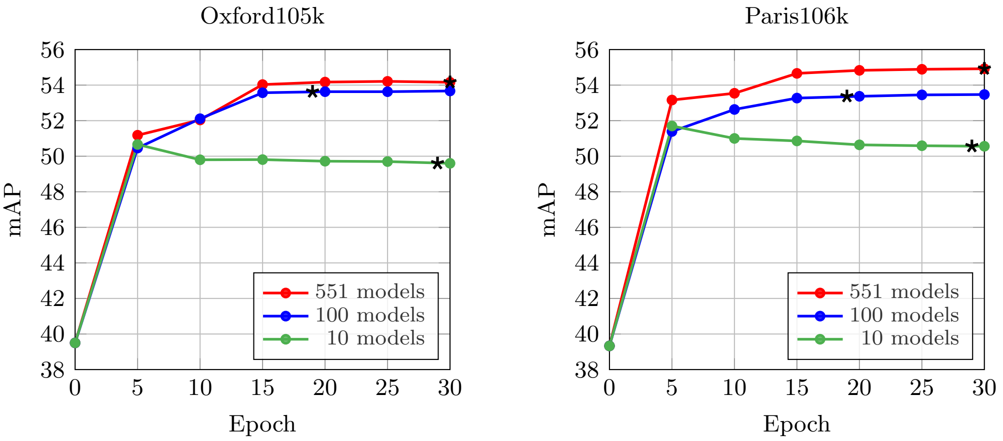


Figure 6.12. Influence of the number of 3D models used for CNN fine-tuning. Performance is evaluated with AlexNet MAC on Oxford105k and Paris106k datasets using 10, 100 and 551 (all available) 3D models. The network with the best performance on the validation set is marked with ⋆.

Table 6.1. Performance (mAP) comparison after CNN fine-tuning for different pooling layers. GeM is evaluated with a single shared pooling parameter or multiple pooling parameters (one for each feature map), which are either fixed or learned. A single value or a range is reported in the case of a single or multiple parameters, respectively. Results reported with AlexNet and with the use of L_w. The best performance highlighted in **bold**.

Pooling	Initial p	Learned p	Oxford5k	Oxford105k	Paris6k	Paris106k	Holidays	Hol101k
MAC	inf	–	62.2	52.8	68.9	54.7	78.4	66.0
SPoC	1	–	61.2	54.9	70.8	58.0	79.9	70.6
GeM	3	–	**67.9**	60.2	74.8	61.7	83.2	73.3
	[2, 5]	–	66.8	59.7	74.1	60.8	**84.0**	73.6
	[2, 10]	–	65.6	57.8	72.2	58.9	81.9	71.9
	3	2.32	67.7	**60.6**	**75.5**	**62.6**	83.7	**73.7**
	3	[1.0, 6.5]	66.3	57.8	74.0	60.5	83.2	72.7
	[2, 10]	[1.6, 9.9]	65.3	56.4	71.4	58.6	81.4	70.8

Pooling methods. We evaluate the effect of different pooling layers during CNN fine-tuning. We present the results in Table 6.1. GeM layer consistently outperforms the conventional max and average pooling. This holds for each of the following cases, (i) a single shared pooling parameter p is used, (ii) each feature map has different p_k and (iii) the pooling parameter(s) is (are) either fixed or learned. Learning a shared parameter turns out to be better than learning multiple ones, as the latter makes the cost function more complex. Additionally, the initial values seem to matter to some extent, with a preference for intermediate values. Finally, a shared fixed parameter and a shared learned parameter perform similarly, with the latter being slightly better. This is the case which we adopt for the rest of our experiments, *i.e.* a single shared parameter p that is learned.

Table 6.2. Performance (mAP) comparison of CNN vector post-processing: no post-processing, PCA-whitening [68] (PCA_w) and our learned whitening (L_w). No dimensionality reduction is performed. Fine-tuned AlexNet (Alex) produces a 256D vector and fine-tuned VGG a 512D vector. The best performance highlighted in **bold**, the worst in **blue**. The proposed method consistently performs either the best (22 out of 24 cases) or on par with the best method. On the contrary, PCA_w [68] often hurts the performance significantly. Best viewed in color.

Net	Post	Dim	Oxford5k		Oxford105k		Paris6k		Paris106k		Holidays		Hol101k	
			MAC	GeM	MAC	GeM	MAC	GeM	MAC	GeM	MAC	GeM	MAC	GeM
Alex	–	256	60.2	**60.1**	**54.2**	54.1	67.5	**68.6**	**54.9**	**56.9**	**74.5**	**78.7**	64.8	**70.9**
	PCA_w		**56.9**	63.7	**44.1**	**53.7**	**64.3**	73.2	**46.8**	57.4	75.4	82.5	**63.1**	71.8
	L_w		**62.2**	**67.7**	52.8	**60.6**	**68.9**	**75.5**	54.7	**62.6**	**78.4**	**83.7**	**66.0**	**73.7**
VGG	–	512	82.0	**82.0**	76.0	**76.9**	**78.3**	**79.7**	71.2	**72.6**	**79.9**	**83.1**	**69.4**	**74.5**
	PCA_w		**78.4**	83.1	**71.3**	77.7	80.6	84.5	**70.9**	76.9	82.2	86.6	70.0	75.9
	L_w		**82.3**	**85.9**	**77.0**	**81.7**	**83.8**	**86.0**	**76.2**	**79.6**	**84.1**	**87.3**	**71.9**	**77.1**

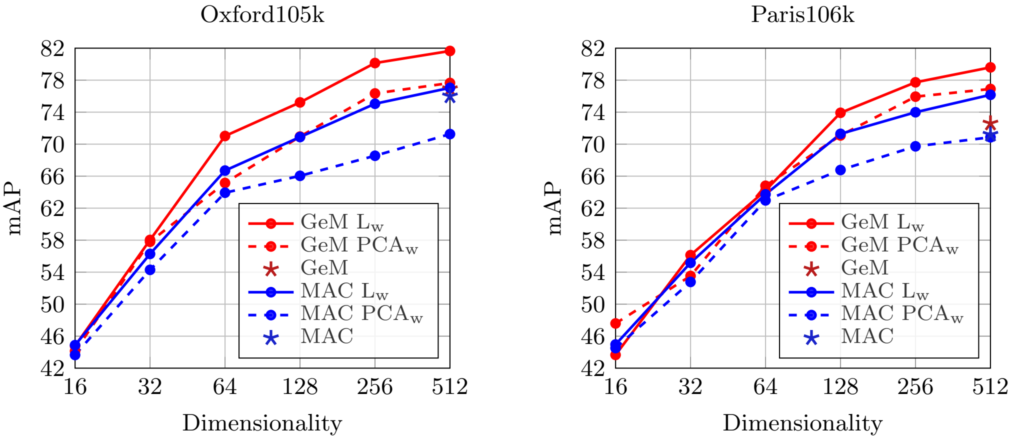


Figure 6.13. Performance comparison of the dimensionality reduction performed by PCA_w and our L_w with the fine-tuned VGG with MAC layer and the fine-tuned VGG with GeM layer on Oxford105k and Paris106k datasets.

Learned projections. The PCA-whitening [68] (PCA_w) is shown to be essential in some cases of CNN-based descriptors [9, 8, 168]. On the other hand, it is shown that on some datasets, the performance after PCA_w substantially drops compared to the raw descriptors (max pooling on Oxford5k [8]). We perform comparison of the traditional whitening methods and the proposed learned discriminative whitening (L_w), described in Section 6.1.4. Table 6.2 shows results without post-processing, with PCA_w and with L_w. Our experiments confirm that PCA_w often reduces the performance. In contrast to that, the proposed L_w achieves the best performance in most cases and is never the worst-performing method. Compared with the no post-processing baseline, L_w reduces the performance twice for AlexNet, but the drop is negligible compared to the drop observed for PCA_w. For VGG, the proposed L_w *always* outperforms the no post-processing baseline.

We conduct an additional experiment by appending a whitening layer at the end of the network during fine-tuning. In this way, whitening is learned in an end-to-end manner, along with the convolutional filters and with the same training data in batch-

Table 6.3. Performance (mAP) evaluation of the multi-scale representation using the fine-tuned VGG with GeM layer. The original scale and down-sampled versions of it are jointly represented. The pooling parameter used by the generalized mean is the same as the one learned in the GeM layer of the network and equal to 2.92. The results reported include the use of $\mathrm{L_w}$.

Pooling over scales	Scale					Oxford5k	Oxf105k	Paris6k	Par106k	Holidays	Hol101k
	$1/1$	$1/\sqrt{2}$	$1/2$	$1/\sqrt{8}$	$1/4$						
–	■					85.9	81.7	86.0	79.6	87.3	77.1
Average	■	■				86.8	82.6	86.7	80.2	88.1	79.3
	■	■	■			87.2	82.4	87.3	80.6	89.1	79.6
	■	■	■	■		86.6	81.9	88.2	81.3	89.9	79.9
	■	■	■	■	■	85.1	80.1	88.8	81.6	90.6	80.5
Generalized mean	■	■				87.3	83.1	86.9	80.5	88.1	79.5
	■	■	■			87.9	83.3	87.7	81.3	89.5	79.9
	■	■	■	■		87.7	83.2	88.7	82.3	89.9	80.2
	■	■	■	■	■	86.8	82.4	89.4	82.7	91.1	81.4

mode. Dropout [154] is additionally used for this layer which we find to be essential. We observe that convergence of the network comes much slower in this case, *i.e.* after 60 epochs. Moreover, the final achieved performance is not higher than our $\mathrm{L_w}$. In particular, end-to-end whitening on AlexNet MAC achieves 49.6 and 52.1 mAP on Oxford105k and Paris106k, respectively, while our $\mathrm{L_w}$ on the same network achieves 52.8 and 54.7 mAP on Oxford105k and Paris106k, respectively. Therefore, we adopt $\mathrm{L_w}$ as it is much faster to train and more effective.

Dimensionality reduction. We compare dimensionality reduction performed with $\mathrm{PCA_w}$ [68] and with our $\mathrm{L_w}$. The performance for varying descriptor dimensionality is plotted in Figure 6.13. The plots suggest that $\mathrm{L_w}$ works better in most dimensionalities.

Multi-scale representation. We evaluate multi-scale representation constructed at test time without any additional learning. We compare the previously used averaging of descriptors at multiple image scales [58] with our generalized-mean of the same descriptors. Results are presented in Table 6.3, where there is a significant benefit when using the multi-scale GeM. It also offers some improvement over average pooling. In the rest of our experiments we adopt multi-scale representation, pooled by generalized mean, for scales 1, $1/\sqrt{2}$, and $1/2$. Results using the supervised dimensionality reduction by $\mathrm{L_w}$ on the multi-scale GeM representation are shown in Table 6.4.

Query expansion. We evaluate the proposed αQE, which reduces to AQE for $\alpha = 0$, and present results in Figure 6.14. Note that Oxford and Paris have different statistics in terms of the number of relevant images per query. The average, minimum, and maximum number of positive images per query on Oxford is 52, 6, and 221, respectively. The same measurements for Paris are 163, 51, and 289. As a consequence, AQE behaves in a very different way across these dataset, while our αQE is a more stable choice. We finally set $\alpha = 3$ and $n_{\mathrm{QE}} = 50$.

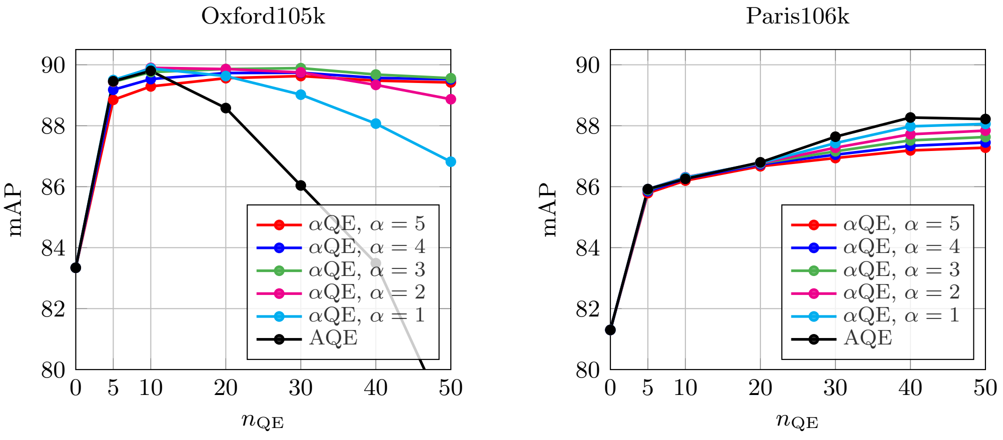


Figure 6.14. Performance evaluation of our α-weighted query expansion (αQE) with the VGG with GeM layer, multi-scale representation, and L_w on Oxford105k and Paris106k datasets. We compare the standard average query expansion (AQE) to our αQE for different values of α and number of images used n_{QE}.

Over-fitting and generalization. In all experiments, all 3D models including any image (not only query landmarks) from Oxford5k or Paris6k datasets are removed. We now repeat the training using all 3D models, including those of Oxford and Paris landmarks. In this way, we evaluate whether the network tends to over-fit to the training data or to generalize. The same amount of training queries is used for a fair comparison. We observe negligible difference in the performance of the network on Oxford and Paris evaluation results, *i.e.* the difference in mAP was on average +0.3 over all testing datasets. We conclude that the network generalizes well and is relatively insensitive to over-fitting.

Comparison with the state of the art. We extensively compare our results with the state-of-the-art performance on compact image representations and on approaches that do query expansion. The results for the fine-tuned GeM based networks are summarized together with previously published results in Table 6.5. The proposed methods outperform the state of the art on all datasets when the VGG network architecture and initialization are used. Our method is outperformed by the work of Gordo *et al.* [58] on Paris with the ResNet architecture, while we have the state-of-the-art score on Oxford. We are on par with the state-of-the-art on Holidays. Note, however, that we did not perform any manual labeling or cleaning of our training data, while in the work of [58] landmark labels were used. We additionally combine GeM with query expansion and further boost the performance.

Visualization with t-SNE. We use t-distributed Stochastic Neighbor Embedding (t-SNE) [171] to perform dataset visualization and examine the similarities. The illustration is given in Figure 6.15 for database images of both Oxford5k and Paris6k. It is clearly visible that images of the same landmark are grouped closely together, even in the cases of significant viewpoint, scale, and illumination (day–night) change.

Table 6.4. Performance (mAP) evaluation for varying descriptor dimensionality after reduction with L_w. Results reported with the fine-tuned VGG with GeM and the fine-tuned ResNet (Res) with GeM. Multi-scale representation is used at the test time for both networks.

Net	Dim	Oxford5k	Oxford105k	Paris6k	Paris106k	Holidays	Hol101k
VGG	512	87.9	83.3	87.7	81.3	89.5	79.9
	256	85.4	79.7	85.7	78.2	87.8	77.2
	128	81.6	75.4	83.4	74.9	84.4	72.6
	64	77.0	69.9	77.4	66.7	81.1	66.2
	32	66.9	57.4	72.2	58.6	72.9	54.3
	16	56.2	44.4	63.5	45.5	60.9	36.9
	8	34.1	25.7	43.9	29.0	43.4	13.8
Res	2048	87.8	84.6	92.7	86.9	93.9	87.9
	1024	86.2	82.4	91.8	85.3	92.5	86.1
	512	84.6	80.4	90.0	82.6	90.6	83.2
	256	83.1	77.3	87.5	78.8	88.4	80.2
	128	79.5	72.2	84.5	74.3	85.9	76.5
	64	74.0	65.8	78.4	65.3	80.3	66.9
	32	57.9	48.5	70.8	56.1	71.2	51.9
	16	40.3	31.8	61.8	45.6	56.4	31.3
	8	25.3	16.3	44.3	27.8	37.8	11.4

6.4. Concluding remarks

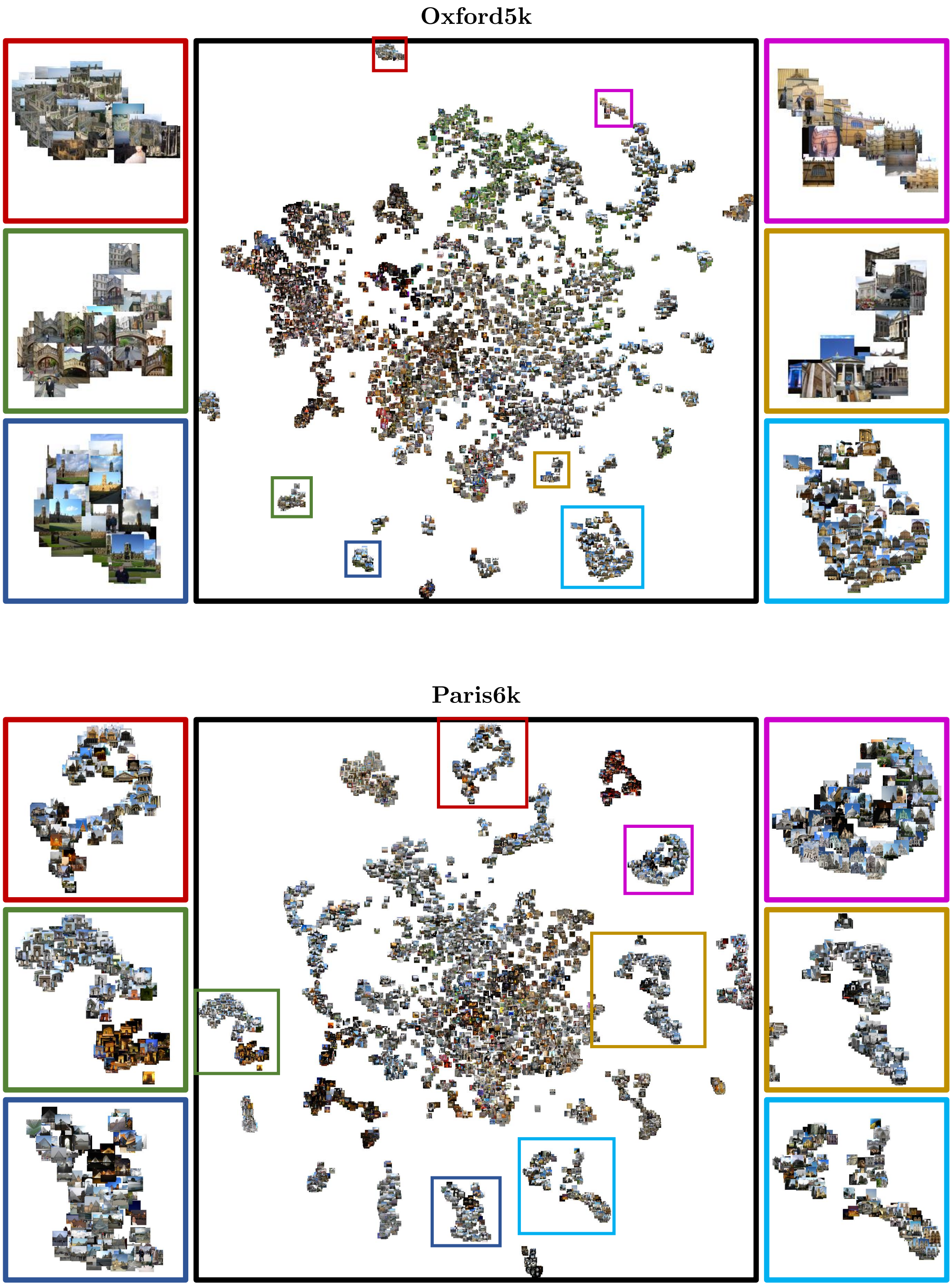
We addressed fine-tuning of CNN for image retrieval. We propose to fine-tune CNN for image retrieval from a large collection of unordered images in a fully automated manner. Tightly coupled state-of-the-art retrieval and SfM methods are employed to obtain 3D models, which are used to guide the selection of the training data for CNN fine-tuning. The reconstructions consist of buildings and popular landmarks; however, the same process is applicable to any rigid 3D objects. We show that larger and more complete 3D models are beneficial as they allow for both hard positive and hard negative examples, which in turn enhance the final performance in instance image retrieval with compact codes. The proposed method does not require any manual annotation and yet achieves top performance on standard benchmarks. The achieved results reach the level of the best systems based on local features with spatial matching and query expansion while being faster and requiring less memory. The proposed pooling layer that generalizes previously adopted mechanisms is shown to improve the retrieval accuracy while it is also effective for constructing a joint multi-scale representation. Training data, trained models, and code are publicly available[7].

[7] cmp.felk.cvut.cz/cnnimageretrieval

Table 6.5. Performance (mAP) comparison with the state-of-the-art image retrieval using VGG and ResNet (Res) deep networks, and using local features. F-tuned: Use of the fine-tuned network (yes), or the off-the-shelf network (no), not applicable for the methods using local features (n/a). Dim: Dimensionality of the final compact image representation, not applicable (n/a) for the BoW based methods due to their sparse representation. Our methods are marked with ⋆ and they are always accompanied by the multi-scale representation and our learned whitening $\mathrm{L_w}$. Previous state of the art is highlighted in **bold**, new state of the art in **red outline**. Best viewed in color.

Net	Method	F-tuned	Dim	Oxf5k	Oxf105k	Par6k	Par106k	Hol	Hol101k
Compact representation using deep networks									
VGG	MAC [138]†	no	512	56.4	47.8	72.3	58.0	79.0	66.1
	SPoC [8]†	no	512	68.1	61.1	78.2	68.4	83.9	**75.1**
	CroW [76]	no	512	70.8	65.3	79.7	72.2	85.1	–
	R-MAC [168]	no	512	66.9	61.6	83.0	75.7	86.9‡	–
	BoW-CNN [106]	no	n/a	73.9	59.3	82.0	64.8	–	–
	NetVLAD [3]	no	4096	66.6	–	77.4	–	88.3	–
	NetVLAD [3]	yes	512	67.6	–	74.9	–	86.1	–
	NetVLAD [3]	yes	4096	71.6	–	79.7	–	87.5	–
	Fisher Vector [115]	yes	512	81.5	76.6	82.4	–	–	–
	R-MAC [57]	yes	512	**83.1**	**78.6**	**87.1**	**79.7**	**89.1**	–
	⋆ GeM	yes	512	**87.9**	**83.3**	**87.7**	**81.3**	**89.5**	**79.9**
Res	R-MAC [168]‡	no	2048	69.4	63.7	85.2	77.8	91.3	–
	R-MAC [58]	yes	2048	**86.1**	**82.8**	**94.5**	**90.6**	**94.8**	–
	⋆ GeM	yes	2048	**87.8**	**84.6**	92.7	86.9	93.9	**87.9**
Re-ranking (R) and query expansion (QE)									
n/a	BoW+R+QE [29]	n/a	n/a	82.7	76.7	80.5	71.0	–	–
	BoW-fVocab+R+QE [102]	n/a	n/a	84.9	79.5	82.4	77.3	75.8	–
	HQE [166]	n/a	n/a	88.0	84.0	82.8	–	–	–
VGG	CroW+QE [76]	no	512	74.9	70.6	84.8	79.4	–	–
	R-MAC+R+QE [168]	no	512	77.3	73.2	86.5	79.8	–	–
	BoW-CNN+R+QE [106]	no	n/a	78.8	65.1	84.8	64.1	–	–
	R-MAC+QE [57]	yes	512	**89.1**	**87.3**	**91.2**	**86.8**	–	–
	⋆ GeM+αQE	yes	512	**91.9**	**89.6**	**91.9**	**87.6**	–	–
Res	R-MAC+QE [168]‡	no	2048	78.9	75.5	89.7	85.3	–	–
	R-MAC+QE [58]	yes	2048	**90.6**	**89.4**	**96.0**	**93.2**	–	–
	⋆ GeM+αQE	yes	2048	**91.0**	**89.5**	95.5	91.9	–	–

†: Our evaluation of MAC and SPoC with $\mathrm{PCA_w}$ and with the off-the-shelf network.
‡: Evaluation of R-MAC by [58] with the off-the-shelf network.



Oxford5k

Paris6k

Figure 6.15. Visualization of the Oxford5k (top) and Paris6k (bottom) datasets with t-SNE.

Chapter 7

Image Retrieval: State of the Art Evaluation

This chapter describes an extensive evaluation of state-of-the-art image retrieval methods as of the year 2018. The evaluation is performed on a newly proposed benchmark, which is described in Chapter 4. The methods range from local-feature based to convolutional-neural-network (CNN) based approaches, including various methods of re-ranking. Note that, all of the methods evaluated here were developed before this benchmark was proposed, and, as such, they were not tuned or tailored for it. We set up a fair environment for the comparison, using the same independent dataset to perform any additional processing, such as visual vocabulary (codebook) learning, or descriptor whitening learning. Besides performance, we also provide time and memory requirements across the representative methods.

The contents of this chapter have been published in [130]. The revisited benchmark, along with the new distractor images, is publicly available[1]. The rest of the chapter is organized as follows. A large variety of different state-of-the-art methods is briefly introduced in Section 7.1, and then, they are extensively evaluated on the new benchmark in Section 7.2. Concluding remarks are given in Section 7.3.

7.1. Extensive evaluation

We evaluate a number of state-of-the-art approaches on the new benchmark and offer a rich testbed for future comparisons. We list the approaches in this section and we split them into two main categories, namely, classical retrieval approaches using local features and CNN-based methods producing global image descriptors.

7.1.1. Local-feature-based methods

Methods based on local invariant features [99, 100] and the bag-of-words (BoW) model [151, 125, 31, 126, 30, 102, 166, 19, 189, 193, 159] were dominating the field of image retrieval until the advent of CNN-based approaches [138, 8, 168, 76, 3, 58, 135, 106, 186]. A typical BoW pipeline consists of invariant local feature detection [100], local descriptor extraction [99], quantization with a visual codebook [151], typically created with k-means, assignment of descriptors to visual words and finally descriptor aggregation in a single embedding [73, 124] or individual feature indexing with an inverted file structure [162, 125, 121]. In particular, we use up-right Hessian-affine (HesAff) features [121], RootSIFT (rSIFT) descriptors [4], and create the codebooks on the landmark dataset from [135], same as the one used for the whitening of CNN-based methods. Note that we

[1] cmp.felk.cvut.cz/revisitop

always crop the queries according to the defined region and then perform any processing to be directly comparable to CNN-based methods.

We additionally follow the same BoW-based pipeline while replacing Hessian-affine and RootSIFT with the deep local attentive features (DELF) [114]. The default extraction approach is followed (*i.e.* at most 1000 features per image), but we reduce the descriptor dimensionality to 128 and not to 40 to be comparable to RootSIFT. This variant is a bridge between classical approaches and deep learning.

VLAD. The Vector of Locally Aggregated Descriptors [72] (VLAD) is created by first-order statistics of the local descriptors. The residual vectors between descriptors and the closest centroid are aggregated *w.r.t.* a codebook whose size is 256 in our experiments. We reduce its dimensionality down to 2048 with PCA, while square-root normalization is also used [68].

SMK*. The binarized version of the Selective Match Kernel [163] (SMK*), a simple extension of the Hamming Embedding [69] (HE) technique, uses an inverted file structure to separately indexes binarized residual vectors while it performs the matching with a selective monomial kernel function. The codebook size is 65,536 in our experiments, while burstiness normalization [70] is always used. Multiple assignment to three nearest words is used on the query side, while the hamming distance threshold is set to 52 out of 128 bits. The rest are the default parameters.

ASMK*. The binarized version of the Aggregated Selective Match Kernel [163] (ASMK*) is an extension of SMK* that jointly encodes local descriptors that are assigned to the same visual word and handles the burstiness phenomenon. Same parametrization as SMK* is used.

SP. Spatial verification (SP) is known to be crucial for particular object retrieval [125] and is performed with the RANSAC algorithm [49]. It is applied on the 100 top-ranked images, as these are formed by a first filtering step, *e.g.* the SMK* or ASMK* method. Its result is the number of inlier correspondences, which is one of the most intuitive similarity measures and allows to detect true positive images. To assume that an image is spatially verified, we require 5 inliers with ASMK* and 10 with other methods.

HQE. Query expansion (QE), firstly introduced by Chum *et al.* [31] in the visual domain, typically uses spatial verification to select true positive among the top retrieved result and issues an enhanced query including the verified images. Hamming Query Expansion [166] (HQE) is combining QE with HE. We use same soft assignment as SMK* and the default parameters.

7.1.2. CNN-based global descriptor methods

We list different aspects of a CNN-based method for image retrieval, which we later combine to form different baselines that exist in the literature.

CNN architectures. We include 3 highly influential CNN architectures, namely AlexNet [81], VGG-16 [150], and ResNet101 [60]. They have different number of layers, complexity, and also produce descriptors of different dimensionality (256, 512, and 2048, respectively).

Table 7.1. Performance (mAP) on Oxford (Oxf) and Paris (Par) with the original annotation, and $\mathcal{R}$Oxford and $\mathcal{R}$Paris with the newly proposed annotation with three different protocol setups: Easy (E), Medium (M), Hard (H).

Method	Oxf	$\mathcal{R}$Oxford			Par	$\mathcal{R}$Paris		
		E	M	H		E	M	H
HesAff–rSIFT–SMK*	78.1	74.1	59.4	35.4	74.6	80.6	59.0	31.2
R–[O]–R-MAC	78.3	74.2	49.8	18.5	90.9	89.9	74.0	52.1
R–[135]–GeM	87.8	84.8	64.7	38.5	92.7	92.1	77.2	56.3
R–[135]–GeM+DFS	90.0	86.5	69.8	40.5	95.3	93.9	88.9	78.5

Table 7.2. Time and memory measurements. Extraction time on a single thread GPU (Tesla P100) / CPU (Intel Xeon CPU E5-2630 v2 @ 2.60GHz) per image of size 1024x768, the memory requirements and the search time (single thread CPU) reported for the database of $\mathcal{R}$Oxford+$\mathcal{R}$1M images. Feature extraction + visual word assignment is reported for ASMK*. SP: Geometry information is loaded from the disk and the loading time is included in search time. We did not consider geometry quantization [121].

Method	Memory	Time (sec)		
		Extraction		Search
	(GB)	GPU	CPU	
HesAff–rSIFT–ASMK*	62.0	n/a + 0.06	1.08 + 2.35	0.98
HesAff–rSIFT–ASMK*+SP				2.00
DELF–ASMK*+SP	10.3	0.41 + 0.01	n/a + 0.54	0.52
A–[135]–GeM	0.96	0.12	1.99	0.38
V–[135]–GeM	1.92	0.23	31.11	0.56
R–[135]–GeM	7.68	0.37	14.51	1.21

Pooling. A common practice is to extract a convolutional feature map and perform a pooling mechanism to construct a global image descriptor. We consider max-pooling (MAC) [138, 168], sum-pooling (SPoC) [8], weighted sum-pooling (CroW) [76], regional max-pooling (R-MAC) [168], generalized mean-pooling (GeM) [135], and NetVLAD pooling [3]. The pooling is always applied on top of the last convolutional feature map.

Multi-scale. The input image is resized to a maximum 1024×1024 size. Then, three re-scaled versions with scaling factor of 1, $1/\sqrt{2}$, and $1/2$ are fed to the network. Finally, the resulting descriptors are combined into a single descriptor by average pooling [58] for all methods, except for GeM where generalized-mean pooling is used [135]. This is shown to improve the performance of the CNN-based descriptors [58, 135].

Off-the-shelf vs. retrieval fine-tuning. Networks that are pre-trained on ImageNet [140] (off-the-shelf) are directly applicable on image retrieval. We further consider the following cases of fine-tuning for the task. Radenovic *et al.* [133] fine-tune a network with landmarks photos using contrastive loss [59]. This is available with MAC [133] and GeM pooling [135]. Similarly, Gordo *et al.* [58] fine-tune R-MAC pooling with landmark photos and triplet loss [175]. Finally, NetVLAD [3] is fine-tuned using street-view images and GPS information.

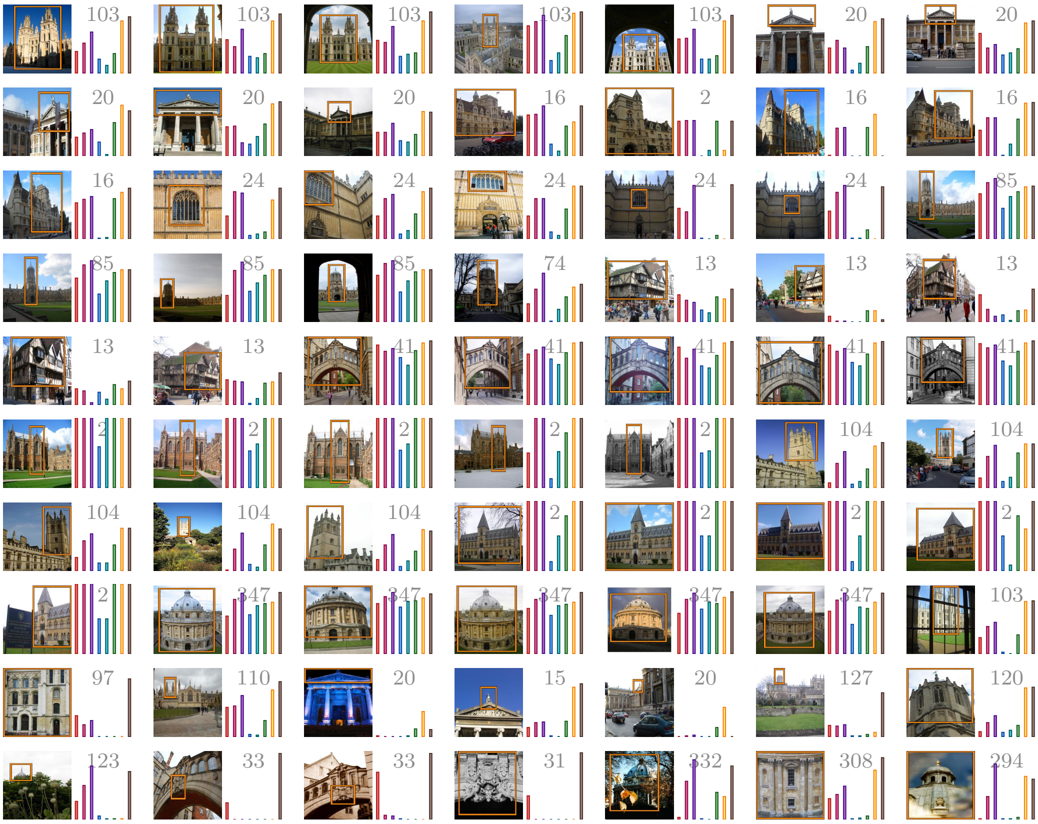


Figure 7.1. Performance (AP) per query on $\mathcal{R}$Oxford + $\mathcal{R}$1M with Medium setup. AP is shown with a bar for 8 methods. The methods, from left to right, are HesAff–rSIFT–ASMK*+SP, DELF–ASMK*+SP, DELF–HQE+SP, V–[O]–R-MAC, R–[O]–GeM, R–[135]–GeM, R–[135]–GeM+DFS, HesAff–rSIFT–ASMK*+SP → R–[135]–GeM+DFS. The total number of easy and hard images is printed on each histogram. Best viewed in color.

Descriptor whitening is known to be essential for such descriptors. We use the same landmark dataset [135] to learn the whitening for all methods. We use PCA whitening [68, 8] for all the off-the-shelf networks, and supervised whitening with SfM labels [97, 133] for all the fine-tuned ones. One exception is the tuning that includes the whitening in the network [58].

Query Expansion is directly applicable on top of global CNN-based descriptors. More specifically, we use α query expansion (αQE) [135] and diffusion (DFS) [67].

7.2. Results

We report a performance comparison between the old and the revisited datasets. Additionally, we provide an extensive evaluation of the state-of-the-art methods on the revisited dataset, with and without the new large-scale distractor set, setting up a testbed for future comparisons.

The evaluation includes local feature-based approaches (see Section 7.1.1 for details and abbreviations), referred to by the combination of local feature type and representation method, *e.g.* HesAff–rSIFT–ASMK*. CNN-based global descriptors are denoted with the following abbreviations. Network architectures are AlexNet (A), VGG-16 (V), and ResNet101 (R). The fine-tuning options are triplet loss with GPS guided mining [3],

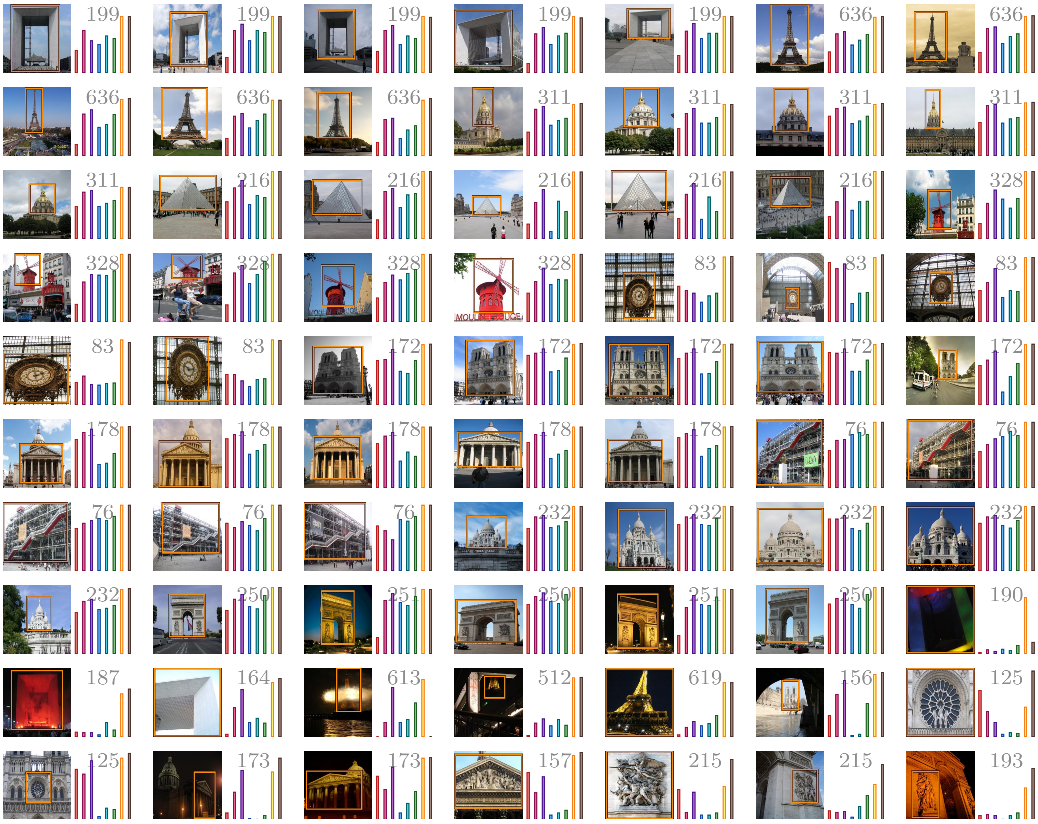


Figure 7.2. Performance (AP) per query on $\mathcal{R}$Paris + $\mathcal{R}$1M with Medium setup. AP is shown with a bar for 8 methods. The methods, from left to right, are HesAff–rSIFT–ASMK*+SP, DELF–ASMK*+SP, DELF–HQE+SP, V–[O]–R-MAC, R–[O]–GeM, R–[135]–GeM, R–[135]–GeM+DFS, HesAff–rSIFT–ASMK*+SP → R–[135]–GeM+DFS. The total number of easy and hard images is printed on each histogram. Best viewed in color.

triplet loss with spatially verified positive pairs [58], contrastive loss with mining from 3D models [133] and [135], and finally the off-the-shelf [O] networks. Pooling approaches are as listed in Section 7.1.2. For instance, ResNet101 with GeM pooling that is fine-tuned with contrastive loss and the training dataset by Radenovic *et al.* [135] is referred to as R–[135]–GeM.

Revisited vs. original. We compare the performance when evaluated on the original datasets, and the revisited annotation with the new protocols. The results for four representative methods are presented in Table 7.1. The old setup appears to be close to the new **Easy** setup, while **Medium** and **Hard** appear to be more challenging. We observe that the performance of the **Easy** setup is nearly saturated and, therefore, we only evaluate **Medium** and **Hard** setups in the subsequent experiments.

State of the art evaluation. We perform an extensive evaluation of the state-of-the-art methods for image retrieval. We present time/memory measurements in Table 7.2 and performance results in Table 7.3. We additionally show the average precision (AP) per query for a set of representative methods in Figures 7.1 and 7.2, for $\mathcal{R}$Oxford and $\mathcal{R}$Paris, respectively. The representative set covers the progress of methods over time in the task of image retrieval. In the evaluation, we observe that there is no single method achieving the highest score on every protocol per dataset. Local-feature-based methods

Table 7.3. Performance evaluation (mAP, mP@10) on $\mathcal{R}$Oxford ($\mathcal{R}$Oxf) and $\mathcal{R}$Paris ($\mathcal{R}$Par) without and with $\mathcal{R}$1M distractors. We report results with the revisited annotation, using Medium and Hard evaluation protocols. We use a color-map that is normalized according to the minimum (white) and maximum (**green** / **orange**) value per column.

Method	Medium								Hard							
	$\mathcal{R}$Oxf		$\mathcal{R}$Oxf+$\mathcal{R}$1M		$\mathcal{R}$Par		$\mathcal{R}$Par+$\mathcal{R}$1M		$\mathcal{R}$Oxf		$\mathcal{R}$Oxf+$\mathcal{R}$1M		$\mathcal{R}$Par		$\mathcal{R}$Par+$\mathcal{R}$1M	
	mAP	mP@10	mAP	mP@10	mAP	mP@10	mAP	mP@10	mAP	mP@10	mAP	mP@10	mAP	mP@10	mAP	mP@10
HesAff–rSIFT–VLAD	33.9	54.9	17.4	34.8	43.6	90.9	19.6	76.1	13.2	18.1	5.6	7.0	17.5	50.7	3.3	21.1
HesAff–rSIFT–SMK*	59.4	83.6	35.8	64.6	59.0	97.4	34.1	89.1	35.4	53.7	16.4	27.7	31.2	72.6	10.5	47.6
HesAff–rSIFT–ASMK*	60.4	85.6	45.0	76.0	61.2	97.9	42.0	95.3	36.4	56.7	25.7	42.1	34.5	80.6	16.5	63.4
HesAff–rSIFT–SMK*+SP	59.8	84.3	38.1	67.1	59.2	97.4	34.5	89.3	35.8	54.0	17.7	30.3	31.3	73.6	11.0	49.1
HesAff–rSIFT–ASMK*+SP	60.6	86.1	46.8	79.6	61.4	97.9	42.3	95.3	36.7	57.0	26.9	45.3	35.0	81.7	16.8	65.3
DELF–ASMK*+SP	67.8	87.9	53.8	81.1	76.9	99.3	57.3	98.3	43.1	62.4	31.2	50.7	55.4	93.4	26.4	75.7
A– [O] – MAC	28.3	44.7	14.1	28.3	47.3	88.6	18.7	69.4	8.8	15.5	3.5	5.1	23.1	61.6	4.1	29.0
A– [O] – GeM	33.8	51.2	16.3	32.4	52.7	90.1	23.8	78.1	10.4	16.7	3.9	6.3	26.0	68.0	5.5	31.6
A– [133] – MAC	41.3	62.1	23.9	43.0	56.4	92.9	29.6	85.4	17.8	28.2	8.4	11.9	28.7	69.3	8.5	40.9
A– [135] – GeM	43.3	62.1	24.2	42.8	58.0	91.6	29.9	84.6	17.1	26.2	9.4	11.9	29.7	67.6	8.4	39.6
V– [O] – MAC	37.8	57.8	21.8	39.7	59.2	93.3	33.6	87.1	14.6	27.0	7.4	11.9	35.9	78.4	13.2	54.7
V– [O] – SPoC	38.0	54.6	17.1	33.3	59.8	93.0	30.3	83.0	11.4	20.9	0.9	2.9	32.4	69.7	7.6	30.6
V– [O] – CroW	41.4	58.8	22.5	40.5	62.9	94.4	34.1	87.1	13.9	25.7	3.0	6.6	36.9	77.9	10.3	45.1
V– [O] – GeM	40.5	60.3	25.4	45.6	63.2	94.6	37.5	88.6	15.7	28.6	7.6	12.1	38.8	79.0	14.2	55.9
V– [O] – R-MAC	42.5	62.8	21.7	40.3	66.2	95.4	39.9	88.9	12.0	26.1	1.7	5.8	40.9	77.1	14.8	54.0
V– [3] – NetVLAD	37.1	56.5	20.7	37.1	59.8	94.0	31.8	85.7	13.8	23.3	6.0	8.4	35.0	73.7	11.5	46.6
V– [133] – MAC	58.4	81.1	39.7	68.6	66.8	97.7	42.4	92.6	30.5	48.0	17.9	27.9	42.0	82.9	17.7	63.7
V– [135] – GeM	61.9	82.7	42.6	68.1	69.3	97.9	45.4	94.1	33.7	51.0	19.0	29.4	44.3	83.7	19.1	64.9
R– [O] – MAC	41.7	65.0	24.2	43.7	66.2	96.4	40.8	93.0	18.0	32.9	5.7	14.4	44.1	86.3	18.2	67.7
R– [O] – SPoC	39.8	61.0	21.5	40.4	69.2	96.7	41.6	92.0	12.4	23.8	2.8	5.6	44.7	78.0	15.3	54.4
R– [O] – CroW	42.4	61.9	21.2	39.4	70.4	97.1	42.7	92.9	13.3	27.7	3.3	9.3	47.2	83.6	16.3	61.6
R– [O] – GeM	45.0	66.2	25.6	45.1	70.7	97.0	46.2	94.0	17.7	32.6	4.7	13.4	48.7	88.0	20.3	70.4
R– [O] – R-MAC	49.8	68.9	29.2	48.9	74.0	97.7	49.3	93.7	18.5	32.2	4.5	13.0	52.1	87.1	21.3	67.4
R– [135] – GeM	64.7	84.7	45.2	71.7	77.2	98.1	52.3	95.3	38.5	53.0	19.9	34.9	56.3	89.1	24.7	73.3
R– [58] – R-MAC	60.9	78.1	39.3	62.1	78.9	96.9	54.8	93.9	32.4	50.0	12.5	24.9	59.4	86.1	28.0	70.0
Query expansion (QE) and diffusion (DFS)																
HesAff–rSIFT–HQE	66.3	85.6	42.7	67.4	68.9	97.3	44.2	90.1	41.3	60.0	23.2	37.6	44.7	79.9	20.3	51.4
HesAff–rSIFT–HQE+SP	71.3	88.1	52.0	76.7	70.2	98.6	46.8	93.0	49.7	69.6	29.8	50.1	45.1	83.9	21.8	61.9
DELF–HQE+SP	73.4	88.2	60.6	79.7	84.0	98.3	65.2	96.1	50.3	67.2	37.9	56.1	69.3	93.7	35.8	69.1
R– [O] – R-MAC+αQE	51.9	70.3	30.8	49.7	77.3	97.9	55.3	94.7	21.8	35.2	5.2	15.9	57.0	87.6	28.0	76.1
V– [135] – GeM+αQE	66.6	85.7	47.0	72.0	74.0	98.4	52.9	95.9	38.9	57.3	21.1	34.6	51.0	88.4	25.6	75.0
R– [135] – GeM+αQE	67.2	86.0	49.0	74.7	80.7	98.9	58.0	95.9	40.8	54.9	24.2	40.3	61.8	90.6	31.0	80.4
R– [58] – R-MAC+αQE	64.8	78.5	45.7	66.5	82.7	97.3	61.0	94.3	36.8	53.3	19.5	36.6	65.7	90.1	35.0	76.9
V– [135] – GeM+DFS	69.6	84.7	60.4	79.4	85.6	97.1	80.7	97.1	41.1	51.1	33.1	49.6	73.9	93.7	65.3	93.1
R– [135] – GeM+DFS	69.8	84.0	61.5	77.1	88.9	96.9	84.9	95.9	40.5	54.4	33.1	48.2	78.5	94.6	71.6	93.7
R– [58] – R-MAC+DFS	69.0	82.3	56.6	68.6	89.5	96.7	83.2	93.3	44.7	60.5	28.4	43.6	80.0	94.1	70.4	89.1
HesAff–rSIFT–ASMK*+SP → R– [135]–GeM+DFS	79.1	92.6	74.3	87.9	91.0	98.3	85.9	97.1	52.7	66.1	48.7	65.9	81.0	97.9	73.2	96.6
HesAff–rSIFT–ASMK*+SP → R– [58]–R-MAC+DFS	80.2	93.7	74.9	87.9	92.5	98.7	87.5	97.1	54.8	70.6	47.5	62.4	84.0	98.3	76.0	96.3
DELF–ASMK*+SP → R– [58]–R-MAC+DFS	75.0	87.9	68.7	83.6	90.5	98.0	86.6	98.1	48.3	64.0	39.4	55.7	81.2	95.6	74.2	94.6

perform very well on $\mathcal{R}$Oxford, especially at large scale, achieving state-of-the-art performance, while CNN-based methods seem to dominate on $\mathcal{R}$Paris. We observe that BoW-based classical approaches are still not obsolete, but their improvement typically comes at significant additional cost. Recent CNN-based local features, *i.e.* DELF, reduce the number of features and improve the performance at the same time.

CNN fine-tuning consistently brings improvements over the off-the-shelf networks. The new protocols make it clear that improvements are needed at larger scale and the hard setup. Many images are not retrieved, while the top 10 results mostly contain false positives. Interestingly, we observe that query expansion approaches (*e.g.* diffusion) degrade the performance of queries with few relevant images (see Figures 7.1 and 7.2). This phenomenon is more pronounced in the revisited datasets, where the the query images are removed from the preprocessing. We did not include separate regional representation and indexing [138], which is previously shown to be beneficial. Our experiments with ResNet and GeM pooling show that it does not deliver improvements that are significant enough to justify the additional memory and complexity cost.

The best of both worlds. The new dataset and protocols reveal space for improvement by CNN-based global descriptors in cases where local features are still better. Diffusion performs similarity propagation by starting from the query's nearest neighbors according to the CNN global descriptor. This inevitably includes false positives, especially in the case of few relevant images. On the other hand, local features, *e.g.* with ASMK*+SP, offer a verified list of relevant images. Starting the diffusion process from geometrically verified images obtained by BoW methods combines the benefits of the two worlds. This combined approach, shown at the bottom part of Table 7.3, improves the performance and supports the message that both worlds have their own benefits. Of course this experiment is expensive and we perform it to merely show a possible direction to improve CNN global descriptors. There are more methods that combine CNNs and local features [190], but we focus on the results related to methods included in our evaluation.

7.3. Concluding remarks

We have revisited two of the most established image retrieval datasets in Chapter 4, that were perceived as performance saturated. This includes new annotation for both datasets that was created with an extra attention to the reliability of the ground truth, and an introduction of 1M hard distractor set.

The goal of this chapter was not to propose a new method, but rather to evaluate the best performing approaches as of the year 2018. An extensive evaluation, performed in this chapter, provides a fair testbed for future comparisons. Evaluated methods range from local-feature based to CNN based approaches, and there is no single approach that achieves the maximum performance on all datasets and difficulty settings. The best results are achieved by taking the best of the two worlds, but at a heavy computational and memory cost. Thus, we conclude that image retrieval is still an open problem, especially at large scale and under difficult viewing conditions. In fact, image retrieval appears far from being solved.

Chapter 8

Adversarial Attack to Conceal the Query Image

Access to online visual search engines implies sharing of private user content – the query images. We introduce the concept of target mismatch attack for deep learning based retrieval systems to generate an adversarial image to conceal the query image. The adversarial image looks nothing like the user intended query, but leads to identical or very similar retrieval results, see Figure 8.1.

Information about users is valuable. Websites, service providers, and even operating systems collect and store user data. The collected data have various forms, *e.g.* visited websites, interactions between users in social networks, hardware fingerprints, keyboard typing or mouse movement patterns, *etc.* Internet search engines record what the users search for, as well as the responses, *i.e.* clicks, to the returned results.

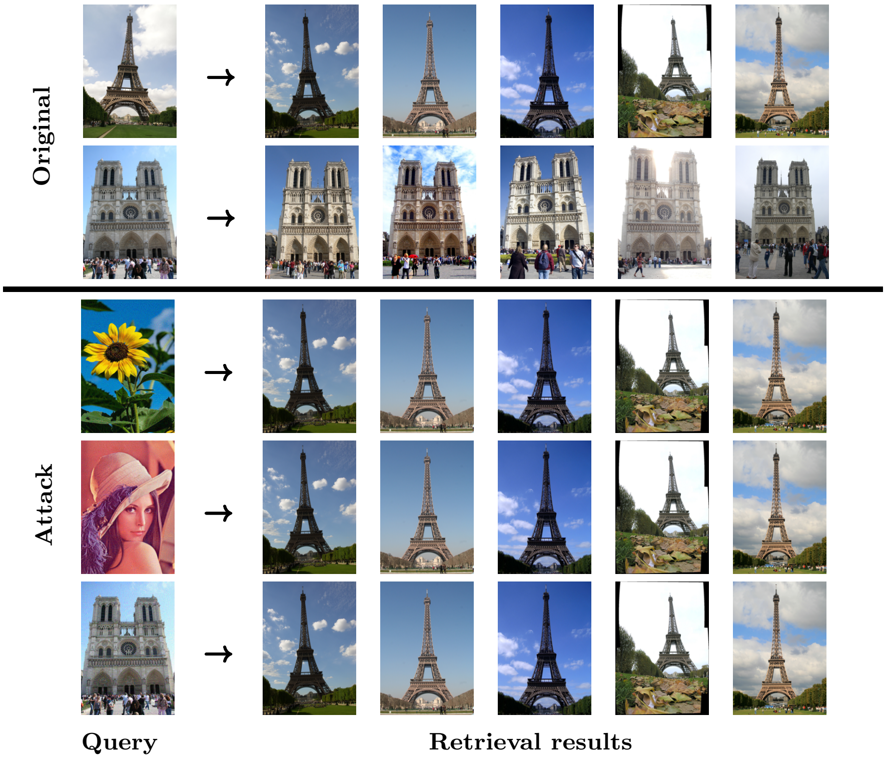


Figure 8.1. Top two rows show retrieval results to the user query image (target). Bottom three rows show the results of our attack where a carrier image (flower, Lena, Notre Dame) has been perturbed to have identical descriptor to that of the target in the first row. Identical results are obtained without disclosing the target.

Recent development in computer vision allowed efficient and precise large scale image search engines to be launched, such as Google Image Search. Nevertheless, similarly to text search engines, queries – the images – are stored and further analyzed by the provider[1]. In this work, we protect the user image (*target*) by constructing a novel image. The constructed image is visually dissimilar to the target, however, when used as a query, identical results are retrieved as with the target image. Large-scale search methods require short-code image representation, both for storage minimization and for search efficiency, which are usually extracted with Convolutional Neural Networks (CNN). We formulate the problem as an adversarial attack on CNNs.

Adversarial attacks, as introduced by Szegedy *et al.* [158], study *imperceptible* non-random image perturbations to mislead a neural network. The first attacks were introduced and tested on image classification. In that context, adversarial attacks are divided into two categories, namely *non-targeted* and *targeted.* The goal of non-targeted attacks is to change the prediction of a test image to an arbitrary class [108, 107], while targeted attacks attempt to make a specific change of the network prediction, *i.e.*, to misclassify the test image to a predefined target class [158, 22, 43].

Similarly to image classification, adversarial attacks have been proposed in the domain of image retrieval too. An non-targeted attack attempts to generate an image that for a human observer carries the same visual information, while for the neural network it appears dissimilar to other images of the same object [87, 91, 192]. This way, a user protects personal images and does not allow them to be indexed for content-based search, even when the images are publicly available. In this chapter, we address targeted attacks aiming to retrieve images that are related to a hidden target query without explicitly revealing the image (see Figure 8.1). A concept that bears resemblance to ours exists in the speech recognition, but in a malicious context. Carlini *et al.* [21] generate *hidden voice commands* that are imperceivable to human listeners but are interpreted as commands by devices. We investigate adversarial attacks beyond the white-box scenario, in which all the parameters and design choices of the retrieval system are known. Specifically, we analyze the cases of unknown indexing image resolution and unknown global pooling used in the network.

The work described in this chapter originates from [167]. This chapter is organized as follows. Section 8.1 gives the background on adversarial attacks on both image classification and image retrieval domains. In Section 8.2, we formulate the targeted attack on image retrieval problem, and propose an approach to address it. We validate the success of our proposed attack in Section 8.3, and finally, we give concluding remarks in Section 8.4.

8.1. Background

We provide the background for non-targeted and targeted adversarial attacks in the domain of image classification, then detail the basic components of CNN-based image retrieval approaches, and finally discuss non-targeted attacks for image retrieval. All variants presented in this section assume white-box access to the network classifier for classification or the feature extractor network for retrieval.

[1] Google Search Help: “The pictures you upload in your search may be stored by Google for 7 days. They won’t be a part of your search history, and we’ll only use them during that time to make our products and services better.”

8.1.1. Image classification attacks

We denote the initial RGB image, called the *carrier image*, by tensor $\mathbf{x}_c \in [0,1]^{W \times H \times 3}$, and its associated label by $y_c \in \{1 \dots K\}$. A CNN trained for K-way classification, denoted by function $f : \mathbb{R}^{W \times H \times 3} \to \mathbb{R}^K$, produces vector $f(\mathbf{x}_c)$ comprising class confidence values. Adversarial attack methods for classification typically study the case of images with correct class prediction, *i.e.* $\arg\max_i f(\mathbf{x}_c)_i$ is equal to y_c, where $f(\mathbf{x}_c)_i$ is the i-th dimension of vector $f(\mathbf{x}_c)$. An adversary aims at generating *adversarial image* $\mathbf{x}_a$ that is visually similar to the carrier image but is classified incorrectly by f. The goal of the attack can vary [2] and corresponds to different loss functions optimizing $\mathbf{x} \in [0,1]^{W \times H \times 3}$.

Non-targeted misclassification is achieved by reducing the confidence for class y_c, while increasing for all other classes. It is achieved by minimizing loss function

$$L_{\text{nc}}(\mathbf{x}_c, y_c; \mathbf{x}) = -\ell_{\text{ce}}(f(\mathbf{x}), y_c) + \lambda \, ||\mathbf{x} - \mathbf{x}_c||^2. \tag{8.1}$$

Function $\ell_{\text{ce}}(f(\mathbf{x}), y_c)$ is the cross-entropy loss which is maximized to achieve the misclassification. In this way, misclassification is performed to any wrong class. The second term $||\mathbf{x} - \mathbf{x}_c||^2$, called *carrier distortion* or simply *distortion*, is the squared l_2 norm of the perturbation $\mathbf{r} = \mathbf{x} - \mathbf{x}_c$.

Targeted misclassification has the goal of generating an adversarial image that gets classified into target class y_t. It is achieved by minimizing loss function

$$L_{\text{tc}}(\mathbf{x}_c, y_t; \mathbf{x}) = \ell_{\text{ce}}(f(\mathbf{x}), y_t) + \lambda \, ||\mathbf{x} - \mathbf{x}_c||^2. \tag{8.2}$$

In contrast to (8.1), cross-entropy loss is minimized *w.r.t.* the target class instead of maximized *w.r.t.* the carrier class.

Optimization of (8.1) or (8.2) generates the adversarial images given by

$$\mathbf{x}_a = \arg\min_{\mathbf{x}} L_{\text{nc}}(\mathbf{x}_c, y_c; \mathbf{x}), \tag{8.3}$$

or

$$\mathbf{x}_a = \arg\min_{\mathbf{x}} L_{\text{tc}}(\mathbf{x}_c, y_t; \mathbf{x}), \tag{8.4}$$

respectively. In the literature [158, 22], various optimizers such as Adam [79], or L-BFGS [16] are used. The box constraints, *i.e.* $\mathbf{x} \in [0,1]^{W \times H \times 3}$, are ensured by projected gradient descent, clipped gradient descent, change of variables [22], or optimization algorithms that support box constraints such as L-BFGS. It is a common practice to perform line search for weight $\lambda > 0$ and keep the attack of minimum distortion. The optimization is initialized by the carrier image.

8.1.2. Image retrieval components

This work focuses on attacks on CNN-based image retrieval with global image descriptors. An image is mapped to a high dimensional descriptor by a CNN with a global pooling layer. The descriptor is consequently normalized to have unit l_2 norm. Then, retrieval from a large dataset *w.r.t.* a *query image* reduces to nearest neighbor search via inner product evaluation between the query descriptor and dataset descriptors. The model for descriptor extraction consists of the following components or parameters.

Image resolution: The input image $\mathbf{x}$ is re-sampled to image $\mathbf{x}^s$ to have maximum resolution equal to $s \times s$.

Feature extraction: Image $\mathbf{x}^s$ is fed as input to a Fully Convolutional Network (FCN), denoted by function $g : \mathbb{R}^{W \times H \times 3} \to \mathbb{R}^{w \times h \times d}$, which maps $\mathbf{x}^s$ to tensor $g(\mathbf{x}^s)$. When the image is processed at its original resolution we denote it by $g(\mathbf{x})$.

Pooling: A global pooling operation $h : \mathbb{R}^{w \times h \times d} \to \mathbb{R}^d$ maps the input tensor $g(\mathbf{x}^s)$ to descriptor $(h \circ g)(\mathbf{x}^s)$. We assume that l_2 normalization is included in this process, so that the output descriptor has unit l_2 norm. We consider various options for pooling, namely, max pooling (MAC) [138, 168], sum pooling (SPoC) [8], generalized mean pooling (GeM) [135], regional max pooling (R-MAC) [168], and spatially and channel-wise weighted sum pooling (CroW) [76]. The framework can be extended to multiple other variants [3, 115, 106].

Whitening: Descriptor post-processing is performed by function $w : \mathbb{R}^d \to \mathbb{R}^d$, which includes centering, whitening and l_2 re-normalization [135]. Finally, input image $\mathbf{x}^s$ is mapped to descriptor $(w \circ h \circ g)(\mathbf{x}^s)$.

For brevity we denote $\mathbf{g}_\mathbf{x} = g(\mathbf{x})$, $\mathbf{h}_\mathbf{x} = (h \circ g)(\mathbf{x})$, and $\mathbf{w}_\mathbf{x} = (w \circ h \circ g)(\mathbf{x})$. In the following, we consider an extraction model during the adversarial image optimization and another one during the testing of the retrieval/matching performance. In order to differentiate between the two cases we refer to the components of the former as *attack-model*, *attack-resolution*, *attack-FCN*, *attack-pooling* and *attack-whitening* and the latter as *test-model*, *test-resolution*, *test-FCN*, *test-pooling* and *test-whitening.*

8.1.3. Image retrieval attacks

Adversarial attacks for image retrieval are so far limited to the non-targeted case.

Non-targeted mismatch aims at generating an adversarial image with small perturbation compared to the carrier image and descriptor that is dissimilar to that of the carrier. This is formulated by loss function

$$\begin{aligned} L_{\text{nr}}(\mathbf{x}_c; \mathbf{x}) &= \ell_{\text{nr}}(\mathbf{x}, \mathbf{x}_c) + \lambda \, ||\mathbf{x} - \mathbf{x}_c||^2 \\ &= \mathbf{h}_\mathbf{x}^\top \mathbf{h}_{\mathbf{x}_c} \quad + \lambda \, ||\mathbf{x} - \mathbf{x}_c||^2. \end{aligned} \tag{8.5}$$

The adversarial image is given by minimizer

$$\mathbf{x}_a = \arg\min_{\mathbf{x}} L_{\text{nr}}(\mathbf{x}_c; \mathbf{x}). \tag{8.6}$$

In this way, the adversary modifies images into their non-indexable counterpart. The exact formulation in (8.5) has not been addressed; the closest is the work of Li *et al.* [87] which learns universal adversarial perturbations (UAP) by maximizing l_1 descriptor distance instead of minimizing cosine similarity.

8.2. Method

We formulate the problem of targeted mismatch attack and then propose various loss functions to address it and construct concealed query images.

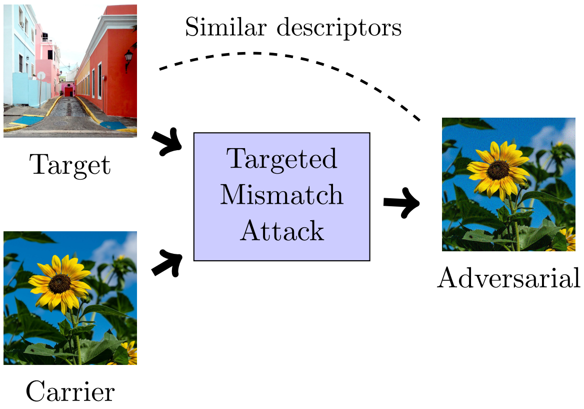


Figure 8.2. In targeted mismatch attack we generate an adversarial image given a carrier and a target image. The adversarial image should match the descriptor of the target image but be visually dissimilar to the target; visual dissimilarity to the target is achieved via visual similarity to the carrier. The attack is formed by a retrieval query using the adversarial image, where the goal is to obtain identical results as with the target query while keeping the target image private.

8.2.1. Problem formulation

The adversary tries to generate an adversarial image with the goal of using it as a (concealed) query for image retrieval instead of a *target image*. The objective is to obtain the same retrieval results without disclosing visual information about the target image itself.

We assume a target image $\mathbf{x}_t \in \mathbb{R}^{W\times H\times 3}$ and a carrier image $\mathbf{x}_c$ with the same resolution (see Figure 8.2). The goal of the adversary is to generate an adversarial image $\mathbf{x}_a$ that has high *descriptor similarity* but very low *visual similarity* to the target. Visual (human) dissimilarity is not straightforward to model; we model visual similarity *w.r.t.* another image, *i.e.* the carrier, instead. We refer to this problem as *targeted mismatch attack* and the corresponding loss function is given by

$$L_{\mathrm{tr}}(\mathbf{x}_c, \mathbf{x}_t; \mathbf{x}) = \ell_{\mathrm{tr}}(\mathbf{x}, \mathbf{x}_t) + \lambda \; ||\mathbf{x} - \mathbf{x}_c||^2. \tag{8.7}$$

In the following we propose different instantiations of the *performance loss* ℓ_{tr} according to the known and unknown components of the test-model.

8.2.2. Targeted mismatch attacks

In all the following, we assume a white-box access to the FCN, while the whitening is assumed unknown and is totally ignored during the optimization of the adversarial image; its impact on the attack is evaluated by adding it to the test-model. In general, if all the parameters of the test-model are known, the task is to generate an adversarial image that reproduces the descriptor of the target image. Then, nearest neighbor search will retrieve identical results as if querying with the target image. Choosing a different performance loss introduces invariance or robustness to some parameters of the attacked retrieval system, when these parameters are unknown. We list different performance loss functions used to minimize (8.7).

Global descriptor. Loss function

$$\ell_{\mathrm{desc}}(\mathbf{x}, \mathbf{x}_t) = 1 - \mathbf{h}_{\mathbf{x}}^{\top} \mathbf{h}_{\mathbf{x}_t}. \tag{8.8}$$

is suitable when all parameters of the retrieval system are known, including the pooling, and when the image is processed by the neural network at its original resolution. Pooling function h is MAC, SPoC, or GeM in our experiments.

Activation tensor. In this scenario, the output of the FCN should be the same for the adversarial and target image, at the original resolution. This is achieved by minimizing the mean squared difference of the two activation tensors

$$\ell_{\text{tens}}(\mathbf{x}, \mathbf{x}_t) = \frac{||\mathbf{g}_{\mathbf{x}} - \mathbf{g}_{\mathbf{x}_t}||^2}{w \cdot h \cdot d}. \tag{8.9}$$

Identical tensors guarantee identical descriptors computed on top of these tensors, including those where spatial information is taken into account. This covers all global or regional pooling operations, and even deep local features, *e.g.* DELF [114]. However, our experiments show that preserving the activation tensor may result in transferring the target's visual content on the adversarial image (see Figure 8.7). Further, the visual appearance of the target image can be partially recovered by inverting [94] the activation tensor of the adversarial image.

Activation histogram. Preserving channel-wise first order statistics of the activation tensor, at the original resolution, is a weaker constraint than preserving the exact activation tensor. It guarantees identical descriptors for all global pooling operations that ignore spatial information. Activation histogram loss function is defined as

$$\ell_{\text{hist}}(\mathbf{x}, \mathbf{x}_t) = \frac{1}{d} \sum_{i=1}^{d} ||u(\mathbf{g}_{\mathbf{x}}, \mathbf{b})_i - u(\mathbf{g}_{\mathbf{x}_t}, \mathbf{b})_i||, \tag{8.10}$$

where $u(\mathbf{g}_{\mathbf{x}}, \mathbf{b})_i$ is the histogram of activations from the i-th channel of $\mathbf{g}_{\mathbf{x}}$ and $\mathbf{b}$ is the vector of histogram bin centers. Histograms are created with soft assignment by an RBF kernel. We use

$$e^{\frac{(x-b)^2}{2\sigma^2}}, \tag{8.11}$$

where $\sigma = 0.1$, x is a scalar activation normalized by the maximum activation value of the target, and b is the bin center. We uniformly sample bin centers in [0,1] with step 0.05. Compared with the tensor case, the histogram optimization does not preserve the spatial distribution, is significantly faster, and does not suffer from undesirable disclosure artifacts.

Different image resolution. We require an adversarial image at the original resolution of the target ($W \times H$), which when down-sampled to resolution s, it retrieves similar results as the target image down-sampled to the same resolution. This is achieved by loss function

$$L^s_{\text{tr}}(\mathbf{x}, \mathbf{x}_t; \mathbf{x}) = \ell_{\text{tr}}(\mathbf{x}^s, \mathbf{x}^s_t) + \lambda \; ||\mathbf{x} - \mathbf{x}_c||^2, \tag{8.12}$$

where ℓ_{tr} can be any of the descriptor, tensor, or histogram based performance loss functions. Note that (8.12) is different from (8.7), the performance loss is computed from re-sampled images, while the distortion loss is still on the original images.

A common down-sampling method used in CNNs is bi-linear interpolation. We have observed that different implementations of such a layer result in different descriptors. The difference is caused by the presence of high-frequencies in the high-resolution image. The adversarial perturbation tends to be high-frequency, therefore different down-sampling results may significantly alternate the result of attack. In order to reduce the sensitivity to down-sampling, we introduce high-frequency removal by Gaussian blurring in the optimization. Instead of (8.12), the following loss is used

$$L_{\text{tr}}^{\hat{s}}(\mathbf{x}, \mathbf{x}_t; \mathbf{x}) = \ell_{\text{tr}}(\mathbf{x}^{\hat{s}}, \mathbf{x}_t^{\hat{s}}) + \lambda \, ||\mathbf{x} - \mathbf{x}_c||^2, \tag{8.13}$$

where $\mathbf{x}^{\hat{s}}$ is image $\mathbf{x}$ blurred with Gaussian kernel with σ_b and then down-sampled. Our experiments show, that blurring plays an important role when the attack-resolution s does not exactly match the test-resolution s', *i.e.* $s' = s + \Delta$.

Ensembles. We perform the adversarial optimization for a combination of the aforementioned loss functions by minimizing their sum. Some examples follow.

- The test-pooling operation is unknown but there is a set $\mathcal{P}$ of possible pooling operations. Minimization of (8.7) is performed for performance loss

$$\ell_{\mathcal{P}}(\mathbf{x}, \mathbf{x}_t) = \frac{\sum_{p \in \mathcal{P}} \ell_p(\mathbf{x}, \mathbf{x}_t)}{|\mathcal{P}|}. \tag{8.14}$$

- The test-resolution is unknown. Joint optimization for a set $\mathcal{S}$ of resolutions is performed with

$$L_{\text{tr}}^{S}(\mathbf{x}, \mathbf{x}_t; \mathbf{x}) = \frac{\sum_{s \in \mathcal{S}} \ell_{\text{tr}}(\mathbf{x}^s, \mathbf{x}_t^s)}{|S|} + \lambda \, ||\mathbf{x} - \mathbf{x}_c||^2. \tag{8.15}$$

  Any performance loss ℓ_{tr} is used, with or without blurring.

8.2.3. Optimization

The optimization is performed with Adam and projected gradient descent is used to apply the box constraints, *i.e.* $\mathbf{x} \in [0, 1]^{W \times H \times 3}$. The adversarial image is initialized by the carrier image, while after every update its values are clipped to be in $[0, 1]$. The adversarial image is given by

$$\mathbf{x}_a = \arg\min_{\mathbf{x}} L_{\text{tr}}(\mathbf{x}_c, \mathbf{x}_t; \mathbf{x}), \tag{8.16}$$

where L_{tr} can be L_{desc} (with “desc” equal to MAC, SPoC, or GeM), $L_{\mathcal{P}}$, L_{hist}, or L_{tens} according to the variant, while the variants with multiple scales are denoted *e.g.* by $L_{\text{hist}}^{\mathcal{S}}$ without blur or $L_{\text{hist}}^{\hat{\mathcal{S}}}$ with blur.

8.3. Experiments

Given a test architecture, we validate the success of the targeted mismatch attack in two ways. First, by measuring the cosine similarity between descriptors of the adversarial image $\mathbf{x}_a$ and the target $\mathbf{x}_t$ (should be as high as possible), and second, by using $\mathbf{x}_a$ as an image retrieval query and compare its performance with that of the target query (should be as close as possible).

8.3.1. Datasets and evaluation protocol

We perform experiments on four standard image retrieval benchmarks, namely Holidays [69], Copydays [44], $\mathcal{R}$Oxford [130], and $\mathcal{R}$Paris [130]. They all consist of a set of query images and a set of database images, while the ground-truth denotes which are the relevant dataset images per query. We choose to perform attacks only with the first

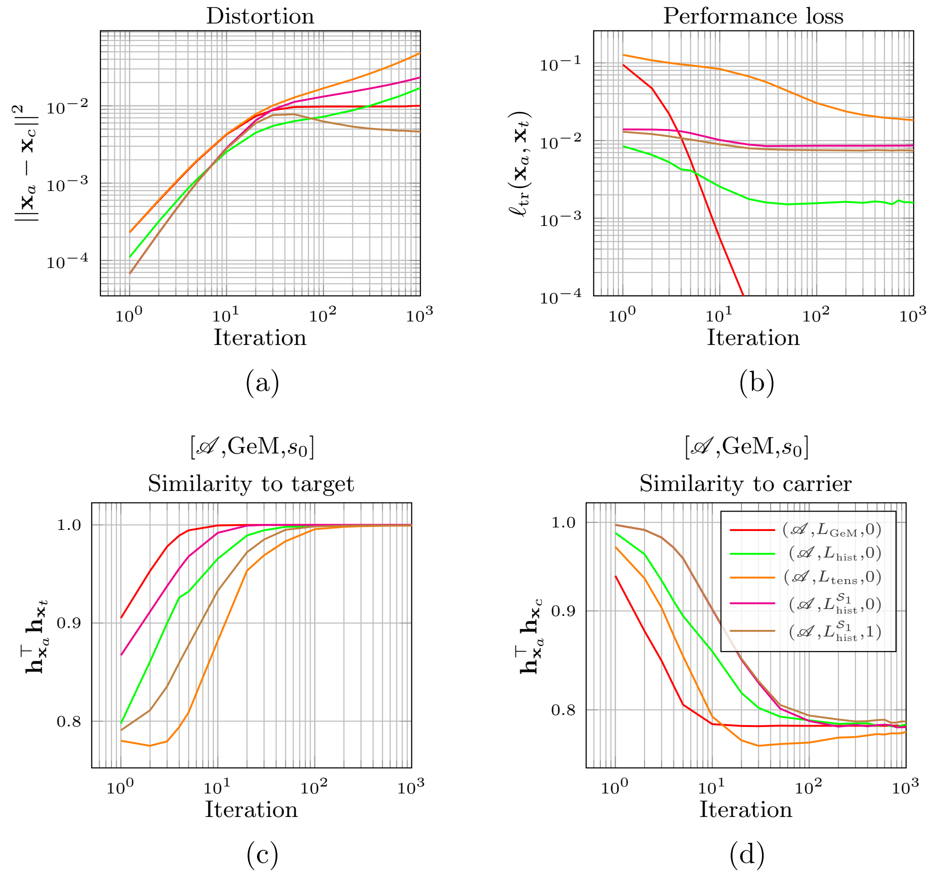


Figure 8.3. We generate adversarial images with different loss function and report various measurements as they evolve with the number of iterations. We show (a) the distortion *w.r.t.* the carrier image, (b) the performance loss from (8.7), (c) descriptor similarity of the adversarial image to the target for test case $[\mathscr{A}$,GeM,$s_0]$ and (d) descriptor similarity of the adversarial image to the carrier for test case $[\mathscr{A}$,GeM,$s_0]$. The target and carrier images are the ones shown in Figure 8.7.

50 queries for Holidays and Copydays to form adversarial attack benchmarks of reasonable size, while for $\mathcal{R}$Oxford and $\mathcal{R}$Paris we keep all 70 of them. All queries are used as targets to form an attack and retrieval performance is measured with mean Average Precision (mAP). Unless otherwise stated we use the "flower" of Figure 8.1 as the carrier; it is cropped to match the aspect ratio of the target. All images are re-sampled to have maximum image resolution equal to 1024 × 1024, this is the original image resolution. $\mathcal{R}$Oxford and $\mathcal{R}$Paris are treated differently than the other two due to the cropped image queries[2]; the relative scale change between queries and database images should be preserved not to affect the ground truth. When the image resolution for descriptor extraction is different than the original one, we down-sample the cropped image with the same scaling factor that the un-cropped one should have been down-sampled with. Results are reported for the *Medium* evaluation setup of these two benchmarks. More details on Holidays and Copydays datasets and their evaluation protocol are given in Chapter 3, Section 3.1.2. $\mathcal{R}$Oxford and $\mathcal{R}$Paris datasets are introduced and described in detail in Chapter 4.

[2] The cropped image region that defines the query is used as a target.

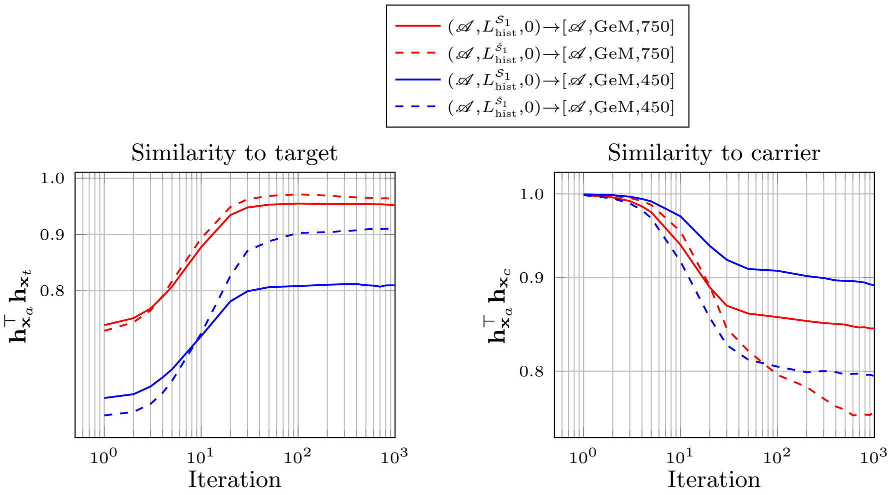


Figure 8.4. Descriptor similarity between the adversarial image and the target or the carrier as it evolves with the number of iterations. We compare the cases without (solid) and with (dashed) blurring for test-resolutions that are not in the attack-resolutions. We use → to denote the adversarial optimization (left) and the test model (right). The target and carrier images are from Figure 8.7.

8.3.2. Implementation details and experimental setup

We set the learning rate equal to 0.01 in all our experiments and perform 100 iterations for L_{desc} and L_{hist}, while 1000 iterations for L_{tens}. If there is no convergence, we decrease the learning rate by a factor of 5 and increase the number of iterations by a factor 2 and re-start. We normalize the distortion term with the dimensionality of $\mathbf{x}$; this is skipped in the loss function of Sections 8.1 and 8.2 for brevity. Moreover, in order to handle the different range of activations for different FCNs, we normalize activation tensors with the maximum target activation before computing the mean squared error in (8.9). Image blurring at resolution s in (8.13) is performed by a Gaussian kernel with $\sigma_b = 0.3 \max(W, H)/s$. The exponent of GeM pooling is always set equal to 3.

Setting $\lambda = 0$ provides a trivial solution to (8.7), *i.e.* $\mathbf{x}_a = \mathbf{x}_t$. However, we observe that initialization by $\mathbf{x}_c$ converges to a local minima that are significantly closer to $\mathbf{x}_c$ than $\mathbf{x}_t$ even for the case of $\lambda = 0$. In this way, we satisfy the non-disclosure constraint, *i.e.* the adversarial image is visually dissimilar to the target, and do not sacrifice the performance loss. The image distortion *w.r.t.* to the carrier image does not sacrifice the goal of concealing the target and preserving user privacy. Therefore, in our experiments we mostly focus on cases with $\lambda = 0$, but also validate cases with $\lambda > 0$ to show the impact of the distortion term or in order to promote the non-disclosure constraint for the case of L_{tens}.

We experiment with different loss functions for targeted mismatch attacks. We define $\mathcal{S}_0$, $\mathcal{S}_1$ and $\mathcal{S}_2$ as sets of attack-resolutions

$$\mathcal{S}_0 = \{s_0\}, s_0 = 1024, \tag{8.17}$$
$$\mathcal{S}_1 = \mathcal{S}_0 \cup \{300, 400, 500, 600, 700, 800, 900\}, \tag{8.18}$$
$$\mathcal{S}_2 = \mathcal{S}_1 \cup \{350, 450, 550, 650, 750, 850, 950\}. \tag{8.19}$$

We denote AlexNet [81], ResNet18 [60], and VGG16 [150] by $\mathscr{A}$, $\mathscr{R}$, and $\mathscr{V}$, respectively, while we only keep their fully convolutional part. The ensemble of AlexNet and ResNet18 is denoted by $\mathscr{E}$; mean loss over the two networks is minimized. We report

L_{tr} h	Original	L_{GeM}	$L_{\mathcal{P}}$	L_{hist}	L_{tens}
	mAP	mAP difference to original			
GeM	41.3	−0.0	−0.0	−0.5	−0.2
MAC	37.0	−0.5	−0.0	−1.3	−0.0
SPoC	32.9	−4.4	−0.1	−0.2	−0.8
R-MAC	44.1	−1.1	−0.5	−0.9	−0.1
CroW	38.2	−1.3	−0.4	−0.2	−0.5
	$\mathbf{x}_t^\top \mathbf{x}_a$				
GeM	1.000	1.000	1.000	0.997	0.998
MAC	1.000	0.972	1.000	0.985	0.997
SPoC	1.000	0.910	1.000	0.999	0.996
R-MAC	1.000	0.972	0.978	0.979	0.997
CroW	1.000	0.968	0.994	0.996	0.998

Table 8.1. Performance evaluation for attacks based on AlexNet, various loss functions optimized at the original image resolution s_0, and $\lambda = 0$. We test on [$\mathscr{A}$,desc,s_0] for multiple types of descriptor/pooling. We show mAP on $\mathcal{R}$Paris and mean descriptor similarity between the adversarial image and the target across all queries. *Original* corresponds to queries without attack.

the triplet attack-model, loss function and value of λ to denote the kind of adversarial optimization, for example ($\mathscr{A}$,$L^{\mathcal{S}_1}_{\text{hist}}$,0). Similarly for other variants. For testing, we report the triplet test-model, test-pooling and test-resolution, for example [$\mathscr{A}$,GeM,s_0].

8.3.3. Results

For each adversarial image we perform the following measurements. We compute its similarity to the target and to the carrier by cosine similarity of the corresponding descriptors, we measure the carrier distortion and, lastly, we perform an attack by using it as a query and measure the average precision which is compared to that of the target image.

Optimization iterations. We perform the optimization for different loss functions and increasing number of iterations. Multiple measurements are reported in Figure 8.3. Optimizing global descriptor or histogram converges much faster than the tensor case and results in significantly lower distortion. This justifies our choice of using a lower number of iterations for the two approaches. Increasing the value of λ keeps the distortion lower but sacrifices the performance loss, as expected.

In Figure 8.4 we show how the similarity to the target and the carrier evolves for test-resolution that is not included in the set of attack-resolutions. Processing the images with image blurring offers significant improvements, especially for the smaller resolutions.

Robustness to unknown test-pooling. In Table 8.1 we present the evaluation comparison for different loss functions and test-pooling. The case of same attack- and test-resolution is examined first. If the test-pooling is directly optimized (L_{GeM} or $L_{\mathcal{P}}$ case), then perfect performance is achieved. The histogram and tensor based approaches both perform well for a variety of test-descriptors.

Robustness to unknown test-resolution. Cases with different attack-resolution and test-resolution are evaluated and results are presented in Figure 8.5. Resolutions that

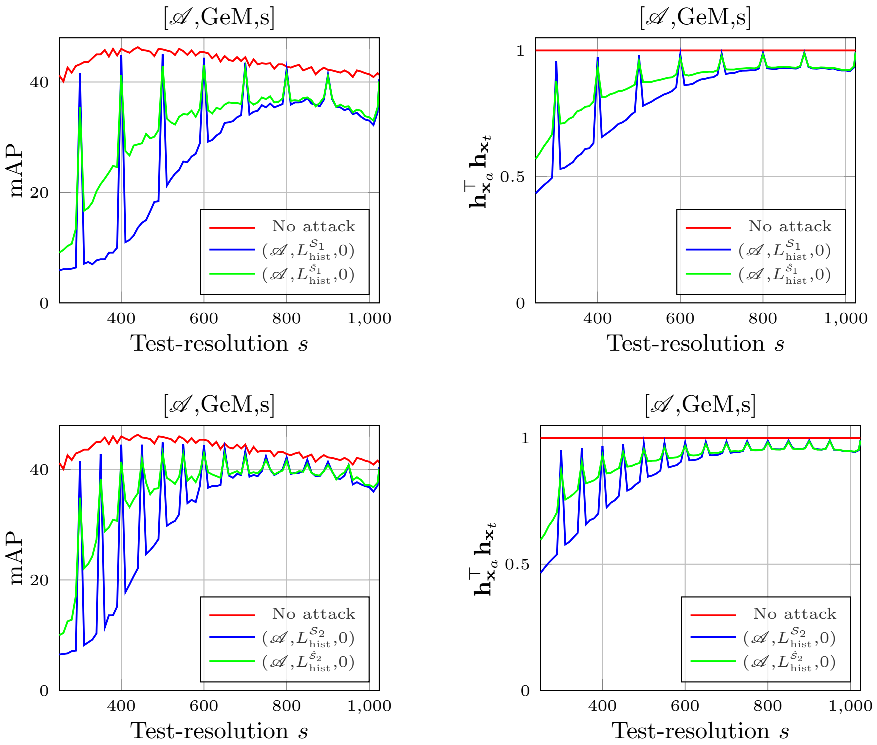


Figure 8.5. Performance evaluation for attack based on AlexNet and a set of attack-resolutions. We show mAP on $\mathcal{R}$Paris and mean descriptor similarity between the adversarial image and the target across all queries, and at increasing test-resolution. Comparison using two sets of attack-resolutions: $\mathcal{S}_1$ (top), and $\mathcal{S}_2$ (bottom); and comparison for optimization without ($\mathcal{S}$) and with ($\hat{\mathcal{S}}$) image blurring.

were not part of the attack-resolutions suffer from significant drop in performance when blurring is not performed, while blurring improves it. We clearly observe how the retrieval performance and descriptor similarity between adversarial image and target are correlated.

Impact of the distortion term. We evaluate $[\mathscr{A}, \text{GeM}, s_0]$ on queries of $\mathcal{R}$Paris for $(\mathscr{A}, L_{\text{hist}}^{\hat{\mathcal{S}}_2}, \lambda)$ and λ equal to 0, 0.1, 1, 10. The average similarity between the adversarial image and the target is 0.990, 0.987, 0.956, and 0.767, respectively, while the average distortion is 0.0177, 0.0083, 0.0026, and 0.0008, respectively. Examples of adversarial images are shown in Figure 8.6.

Impact of the whitening in the test-model. We now consider the case that the test-model includes descriptor whitening. The whitening is unknown during the time of the adversarial optimization. We evaluate the performance of $\mathcal{R}$Paris while learning whitening with PCA on $\mathcal{R}$Oxford. Testing without whitening and $[\mathscr{A}, \text{GeM}, s_0]$ or $[\mathscr{A}, \text{GeM}, 768]$ achieves 41.3, and 40.2 mAP, respectively. After applying whitening the respective performances increase to 47.5 and 48.0 mAP. Attacks with $(\mathscr{A}, L_{\text{hist}}^{\hat{\mathcal{S}}_2}, 0)$ achieve 40.2, and 39.4 mAP when tested in the aforementioned cases without whitening. Attacks with $(\mathscr{A}, L_{\text{hist}}^{\hat{\mathcal{S}}_2}, 0)$ achieve 47.3, and 42.9 mAP when tested in the aforementioned cases with whitening. Whitening introduces some additional challenges, but the attacks seem effective with slightly reduced performance.

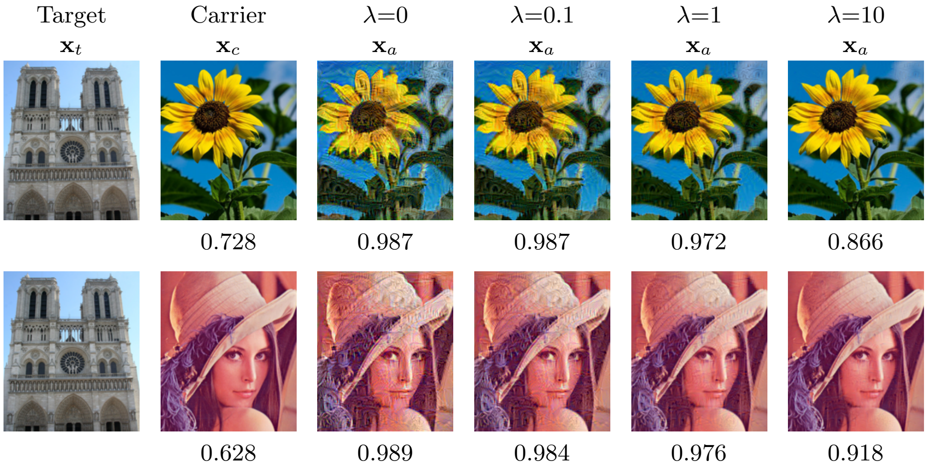

Target $\mathbf{x}_t$
Carrier $\mathbf{x}_c$
λ=0 $\mathbf{x}_a$
λ=0.1 $\mathbf{x}_a$
λ=1 $\mathbf{x}_a$
λ=10 $\mathbf{x}_a$
0.728
0.987
0.987
0.972
0.866
0.628
0.989
0.984
0.976
0.918

Figure 8.6. Adversarial examples for a carrier image and two different targets while optimizing $(\mathscr{A},L^{\hat{S}_2}_{\text{hist}},\lambda)$ for various values of λ. We report descriptor similarity for $[\mathscr{A},\text{GeM},s_0]$.

Concealing/revealing the target. We generate adversarial images for different loss functions and show examples in Figure 8.7. The corresponding tensors show that spatial information is only preserved in the tensor-based loss function. The tensor-based approach requires the distortion term to avoid revealing visual structures of the target (adversarial images in 6-th and 7-th column). We now pose the question "can the FCN activations of the adversarial image reveal the content of the target?". To answer, we invert tensor $\mathbf{g}_{\mathbf{x}_a}$ at multiple resolutions using the method of Mahendran and Vedaldi [94]. The tensor-based approach indeed reveals the target's content in the reconstruction, while no other approach does. This reveals the benefits of the proposed histogram-based optimization. Note that the reconstructed image resembles the target less if the resolutions used in the reconstruction are not the same as the attack-resolutions (rightmost column).

Timings. We report the average optimization time per target image on Holidays dataset and on a single GPU (Tesla P100) for some indicative cases. Optimizing $(\mathscr{A},L_{\text{GeM}},0)$, $(\mathscr{A},L^{\hat{S}_1}_{\text{GeM}},0)$, $(\mathscr{A},L^{\hat{S}_1}_{\text{hist}},0)$, $(\mathscr{A},L^{\hat{S}_2}_{\text{hist}},0)$, and $(\mathscr{A},L^{\hat{S}_1}_{\text{tens}},0)$ takes 1.9, 7.5, 12.3, 22.9, and 68.4 seconds, respectively. Using ResNet18 $(\mathscr{R},L_{\text{GeM}},0)$ and $(\mathscr{R},L^{\hat{S}_2}_{\text{hist}},0)$ take 3.9 and 40.6 seconds, respectively.

Multiple attacks. We show results of multiple attacks in Table 8.2. We present the original retrieval performance together with the difference in the performance caused by the attack. It summarizes the robustness of the histogram and tensor based optimization to unknown pooling operations. It emphasizes the challenges of unknown test-resolution and the impact of the blurring; this outcome can be useful in various different attack models. The very last row suggests that transfer-attacks to different FCNs are hard to achieve and much harder than the case of attacks on classification [158].

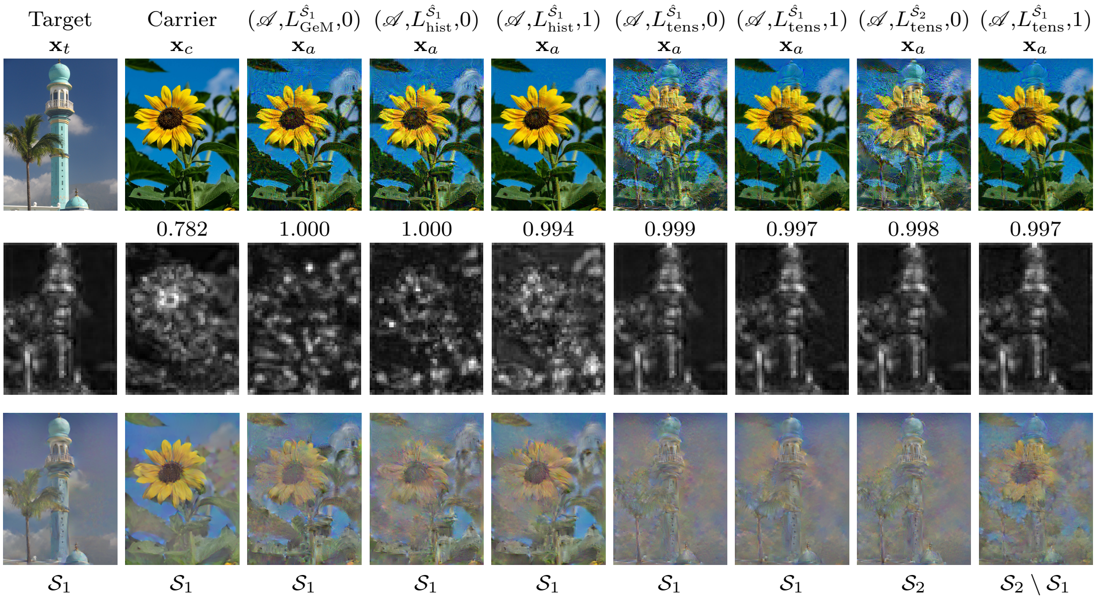

Target
$\mathbf{x}_t$
Carrier
$\mathbf{x}_c$
$(\mathscr{A},L^{\hat{\mathcal{S}}_1}_{\text{GeM}},0)$
$\mathbf{x}_a$
$(\mathscr{A},L^{\hat{\mathcal{S}}_1}_{\text{hist}},0)$
$\mathbf{x}_a$
$(\mathscr{A},L^{\hat{\mathcal{S}}_1}_{\text{hist}},1)$
$\mathbf{x}_a$
$(\mathscr{A},L^{\hat{\mathcal{S}}_1}_{\text{tens}},0)$
$\mathbf{x}_a$
$(\mathscr{A},L^{\hat{\mathcal{S}}_1}_{\text{tens}},1)$
$\mathbf{x}_a$
$(\mathscr{A},L^{\hat{\mathcal{S}}_2}_{\text{tens}},0)$
$\mathbf{x}_a$
$(\mathscr{A},L^{\hat{\mathcal{S}}_1}_{\text{tens}},1)$
$\mathbf{x}_a$
0.782
1.000
1.000
0.994
0.999
0.997
0.998
0.997
$\mathcal{S}_1$
$\mathcal{S}_1$
$\mathcal{S}_1$
$\mathcal{S}_1$
$\mathcal{S}_1$
$\mathcal{S}_1$
$\mathcal{S}_1$
$\mathcal{S}_2$
$\mathcal{S}_2 \setminus \mathcal{S}_1$

Figure 8.7. We show target, carrier and adversarial images for different variants (top image row), a summary of tensor $\mathbf{g}_\mathbf{x}$ by depth-wise maximum (middle image row) and the inversion of $\mathbf{g}_{\mathbf{x}_t}$, $\mathbf{g}_{\mathbf{x}_c}$, or $\mathbf{g}_{\mathbf{x}_a}$, respectively, over multiple resolutions (bottom image row); the resolutions for inversion are reported below the bottom row. The tensor inversion shows whether the target, or any information about it, can be reconstructed from the adversarial image. The first two inversions are presented as a reference. We report descriptor similarity to the target below the first image row for [$\mathscr{A}$,GeM,1024].

Attack	Test	$\mathcal{R}$Oxford	$\mathcal{R}$Paris	Holidays	Copydays
$(\mathscr{A},L^{\hat{\mathcal{S}}_2}_{\text{hist}},0)$	[$\mathscr{A}$,GeM,s_0]	26.9 / +0.2	41.3 / -1.1	81.5 / +0.0	80.4 / -0.5
$(\mathscr{R},L^{\hat{\mathcal{S}}_2}_{\text{GeM}},0)$	[$\mathscr{R}$,GeM,s_0]	21.5 / -0.7	46.9 / -0.4	82.9 / -0.3	69.3 / -0.7
	[$\mathscr{R}$,GeM,768]	23.0 / -4.2	47.9 / -3.8	83.6 / -2.2	75.8 / -3.1
	[$\mathscr{R}$,GeM,512]	22.5 / -6.1	49.9 / -11.5	83.8 / -2.1	81.1 / -9.5
$(\mathscr{R},L^{\mathcal{S}_2}_{\text{hist}},0)$	[$\mathscr{R}$,GeM,s_0]	21.5 / -1.2	46.9 / -1.9	82.9 / -0.4	69.3 / -1.4
	[$\mathscr{R}$,GeM,768]	23.0 / -4.2	47.9 / -7.4	83.6 / -2.9	75.8 / -6.4
	[$\mathscr{R}$,GeM,512]	22.5 / -11.7	49.9 / -20.7	83.8 / -22.1	81.1 / -18.8
$(\mathscr{R},L^{\hat{\mathcal{S}}_2}_{\text{hist}},0)$	[$\mathscr{R}$,GeM,s_0]	21.5 / -0.8	46.9 / -2.0	82.9 / -2.5	69.3 / -1.4
	[$\mathscr{R}$,GeM,768]	23.0 / -5.2	47.9 / -5.7	83.6 / -1.9	75.8 / -4.3
	[$\mathscr{R}$,GeM,512]	22.5 / -6.9	49.9 / -12.2	83.8 / -5.4	81.1 / -10.3
$(\mathscr{R},L^{\hat{\mathcal{S}}_2}_{\mathcal{P}},0)$	[$\mathscr{R}$,CroW,s_0]	22.0 / -1.2	45.0 / -0.6	81.0 / +1.0	67.0 / -1.6
$(\mathscr{R},L^{\hat{\mathcal{S}}_2}_{\text{hist}},0)$		22.0 / -1.0	45.0 / -0.8	81.0 / +1.7	67.0 / -0.9
$(\mathscr{R},L^{\hat{\mathcal{S}}_2}_{\text{tens}},0)$		22.0 / -1.0	45.0 / -0.2	81.0 / -3.6	67.0 / -2.9
$(\mathscr{E},L^{\hat{\mathcal{S}}_2}_{\text{hist}},0)$	[$\mathscr{A}$,GeM,s_0]	26.9 / -2.7	41.3 / -5.6	81.5 / -4.7	80.4 / -5.0
	[$\mathscr{R}$,CroW,s_0]	22.0 / -0.8	45.0 / -0.7	81.0 / +1.1	67.0 / -1.0
	[$\mathscr{V}$,GeM,s_0]	38.1 / -35.0	54.0 / -47.4	85.7 / -72.9	80.0 / -72.8

Table 8.2. Performance evaluation for multiple attacks, test-models, and datasets. We report mAP over the original queries, together with the mAP difference to the original caused by the attack. The parameters of the adversarial optimization during the attack are shown in the leftmost column, while the type of test-model used is shown in the second column.

8.4. Concluding remarks

We have introduced the problem of targeted mismatch attack for image retrieval and address it in order to construct concealed query images instead of the initial intended query. We show that optimizing the first order statistics is a good way to generate images that result in the desired descriptors without disclosing the content of the intended query. We analyze the impact of image re-sampling, which is a natural component of image retrieval systems and reveal the benefits of simple image blurring in the adversarial image optimization. Finally, we show that transfer-attacks to new FCNs are much more challenging than their image classification counterparts.

Chapter 9

Training Convolutional Neural Networks for Shape Matching

In a number of computer vision problems, colour and/or texture in the images is not available or misleading. Three examples are shown in Figure 9.1. In the case of sketches or outlines, there is no colour or texture available at all. In the case of artwork, colour and texture are present, but often can be unrealistic to stimulate certain impression rather than exactly capture the reality. Finally, under extreme illumination changes, such as a day-time versus night images, colour may be significantly distorted and the texture weakened. On the other hand, image discontinuities in colour or texture, as detected by modern edge detectors, and especially their shapes, carry the information about the content, independent of, or insensitive to, the illumination changes, artistic drawing and outlining.

This chapter is targeting at shape matching, in particular the goal is to extract a descriptor that captures the shape depicted in the image. The shape descriptors are extracted by a convolutional neural network (CNN) which is fed with image edge maps. The network is fine-tuned without any human supervision or image, sketch or shape annotation. Starting from a pre-trained classification network stripped off the fully connected layers, the CNN is fine-tuned using a simple contrastive loss function. To acquire the training data, the domain of landmark photographs is used, as richer information than shapes is available in such images. Matching and non-matching image pairs are obtained based on the 3D models and estimated camera positions. Edge maps detected on these images provide training data for the network. Examples of positive

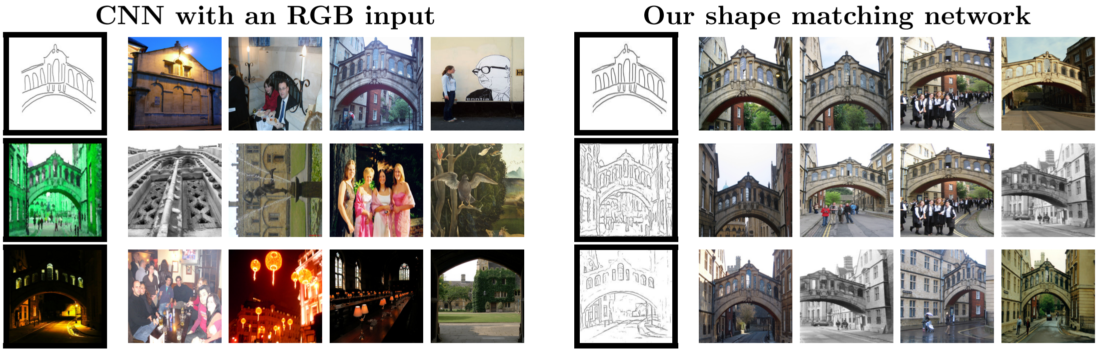
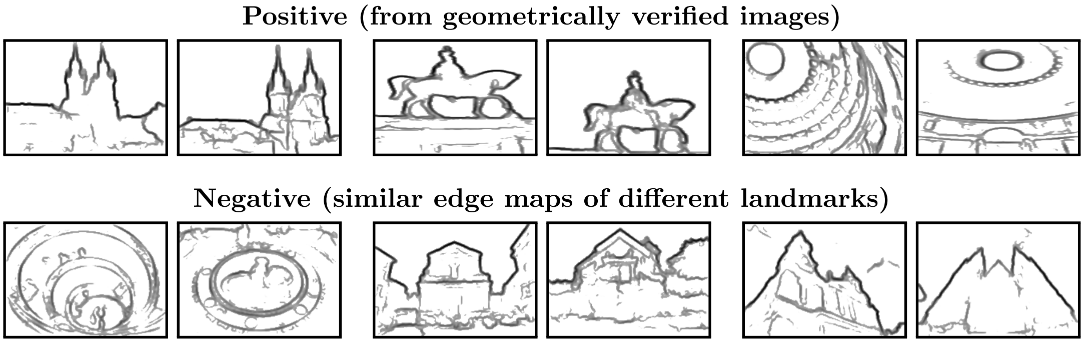


Figure 9.1. Three examples where **shape** is the only relevant information: sketch, artwork, extreme illumination conditions. Top retrieved images from the Oxford Buildings dataset [125]: CNN with an RGB input [133] (left), and our shape matching network (right). Query images are shown with black border.



Positive (from geometrically verified images)

Negative (similar edge maps of different landmarks)

Figure 9.2. Edge maps extracted from matching and non-matching image pairs that serve as training data for our network.

and negative pairs of edge maps are shown in Figure 9.2. Given our trained network, shape descriptors of photographs and paintings are extracted from the corresponding image edge maps, while sketches or black and white line drawings are simply considered as a special type of an edge map.

We show the importance of shape matching on two problems: (i) domain generalization in the case of classification, and (ii) cross modality matching of sketches to images. The domain generalization task is evaluated by performing object recognition. We extract the learned descriptors and train a simple classifier on the seen domain(s), which is later used to classify images of the unseen domain(s). We show, that for some combinations of seen-unseen domains, such as artwork and photograph, descriptors using colour and texture are useful. However, for some combinations, such as photograph and line drawing, the shape information is crucial. Combining both types of descriptors outperforms the state-of-the-art approach in all settings.

In the case of cross modality matching, it is commonly assumed that annotated training data is available for both modalities [14, 143]. Once more, we apply the domain generalization approach by using the descriptors learned on edge maps of building images. We evaluate the performance on sketch-based image retrieval datasets. Modern sketch-based image retrieval takes the path of object recognition from human sketches [185]. Rather than performing shape matching, the networks are trained to recognize simplified human drawings. Such an approach requires very large number of annotated images and drawn sketches for each category of interest [14, 143]. Recently, Yelamarthi *et al.* [182] notice that existing models for sketch-based image retrieval that are trained in a discriminative setting learn only class specific mappings and fail to generalize to the unseen classes. Our extensive quantitative and qualitative experiments support this claim. On the contrary, even though our proposed network is *not trained* to recognize human-drawn object sketches, our experiments show that it performs well on standard benchmarks.

Our training is performed once and the same network is used to extract shape descriptors for both domain generalization for object recognition and multiple benchmarks for sketch-based image retrieval.

The contents of this chapter originate from [134, 136]. Training data, trained models, and code are publicly available[1]. The rest of the chapter is organized as follows. Our training and evaluation approach are presented in Section 9.1. In Section 9.2 we evaluate our network on domain generalization and sketch-based retrieval and compare with the state of the art. Finally, we give conclusions in Section 9.3.

[1] cmp.felk.cvut.cz/cnnimageretrieval

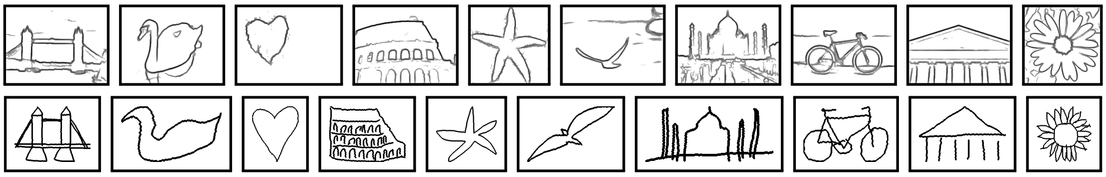


Figure 9.3. Filtered edge maps (top row) from a random sample of the Flickr15k sketch dataset and sketch queries (bottom row).

9.1. Method

In this section we describe the proposed approach. The process of fine-tuning the CNN is described in Section 9.1.1, while the final representation and the way it is used for retrieval and classification is detailed in Section 9.1.2.

We break the end-to-end process of image description into two parts. In the first part, the images are turned into edge maps. In particular, throughout our main experiments we use the edge detector of Dollar and Zitnick [40] due to its great trade-off between efficiency and accuracy, and the tendency not to consider textured regions as edges. Our experiments show that marginally worse or better results are achieved with a CNN-based edge detector [80], depending on the descriptor network architecture. An image is represented as an edge map, which is a 2D array containing the edge strength in each image pixel. The edge strength is in the range of $[0, 1]$, where 0 represents background. Sketches, in the case of sketch-to-image retrieval, are represented as a special case of an edge map, where the edge strength is either 0 for the background or 1 for a contour.

The second part is a fully convolutional network extracting a global image descriptor. The two part approach allows, in a simple manner, to unify all modalities at the level of edge maps. Jointly training these two parts, *e.g.*, in the case of a CNN-based edge detector [80], can deliver an image descriptor too. However, this descriptor may not be based on shapes. It is unlikely that such an optimization would end in a state where the representation between the two parts actually corresponds to edges. Enforcing this with additional training data in the form of edge maps and a loss on the output of the first part is exactly what we are avoiding in this work.

9.1.1. Training

We use a network architecture previously proposed for image classification [81, 150], in particular, we use all convolutional layers and the activations of the very last one, *i.e.*, the network is stripped of the fully-connected layers. The CNN is initialized by the parameters learned on a large scale annotated ImageNet [42] dataset. This is a fairly standard approach adopted in a number of problems, including image search [3, 133, 57, 58, 135]. The network is then fine-tuned with pairs of image edge maps.

The network. The image classification network expects an RGB input image, while the edge maps are only two dimensional. We sum the first convolution filters over RGB. Unlike in RGB input, no mean pixel subtraction is performed to the input data. To obtain a compact, shift invariant descriptor, a global max-pooling [138] layer is appended after the last convolutional layer. This approach is also known as Maximum Activations of Convolutions (MAC) vector [168]. After the MAC layer, the vectors are l_2 normalized.

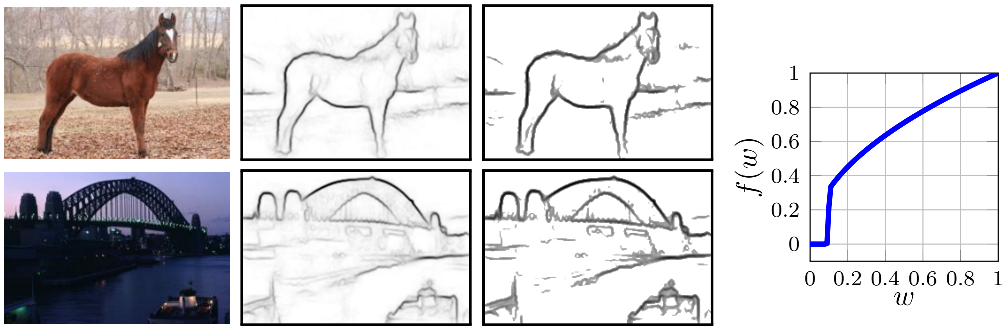


Figure 9.4. Sample images, the output of the edge detector, the filtered edge map, and the learned edge-filtering function.

Edge filtering. A typical output of edge detectors is a strength of an edge in every pixel. We introduce an edge filtering layer to address two frequent issues with edge responses. First, the background often contains close-to-zero responses, which typically introduce noise into the representation. This issue is commonly handled by thresholding the response function. Second, the strength of the edges provides ordering, *i.e.*, higher edge response implies that the edge is more likely to be present, however its value typically does not have practical interpretation. Prior to the first convolution layer, a continuous and differentiable function is pre-pended. This layer is trained together with the rest of the network to transform the edge detector output with soft thresholding by a sigmoid and power transformation. Denote the edge strength by $w \in [0, 1]$. Edge filtering is performed as

$$f(w) = \frac{w^p}{1 + e^{\beta(\tau - w)}}, \tag{9.1}$$

where p controls the contrast between strong and weak edges, τ is the threshold parameter, and β is the scale of the sigmoid choosing between hard thresholding and a softer alternative. The final function (9.1) with learned parameters is plotted in Figure 9.4 (right). The figure also visually demonstrates the effect of application of the filtering. The weak edges are removed on the background and the result appearance is closer to a rough sketch (see Figure 9.3), while the uncertainty in edges is still preserved.

Fine tuning. The CNN is trained with Stochastic Gradient Descent in a Siamese fashion with contrastive loss [26]. The positive training pairs are edge maps of matching images (similarity of the edge maps is not considered), while the negative pairs are similar edge maps (according to the current network state) of non-matching images.

Given a pair of vectors $\mathbf{x}$ and $\mathbf{y}$, the loss [26] is defined as their squared Euclidean distance $||\mathbf{x} - \mathbf{y}||^2$ for positive examples, and as $\max\{(m - ||\mathbf{x} - \mathbf{y}||)^2, 0\}$ for negative examples. Hard-negative mining is performed several times per epoch which has been shown to be essential [3, 133, 57].

Training data. The training images for fine tuning the network are collected in a fully automatic way. In particular, we use our publicly available dataset described in Chapter 6, Section 6.2 and follow the same methodology, briefly reviewed in the following. A large unordered image collection is passed through a 3D reconstruction system based on local features and bag-of-words retrieval [146, 132]. The outcome consists of a set of 3D models which mostly depict outdoor landmarks and urban scenes. For each landmark, a maximum of 30 six-tuples of images are being selected. The six-tuple consists of: one image as the training query, then one matching image to the training query, and five similar non-matching images. This gives arise to one positive and five negative pairs. The geometry of the 3D models, including camera positions, allows to mine matching images, *i.e.*, those that share adequate visual overlap. Negative-pair mining is facilitated by the 3D models, too: negative images are chosen only if they belong to a different model.

Data augmentation. A standard data-augmentation, *i.e.*, random horizontal flipping (mirroring) procedure is applied to introduce further variance in the training data and to avoid over-fitting. The training query and the positive example are jointly mirrored with 50% probability. Negative examples are sought after eventual flipping. We propose an additional augmentation technique for the selected training queries. Their edge map responses are thresholded with a random threshold uniformly chosen from $[0, 0.2]$ and the result is binarized. Matching images (in positive examples) are left unchanged; negative images are selected after the transformation. This augmentation process is applied with a probability of 50%. It offers a level of shape abstraction and mimics the asymmetry of sketch-to-edge map matching. The randomized threshold can be also seen as an approximation of the stroke removal in [184].

9.1.2. Representation, classification and search

We use the trained network to extract image and sketch descriptors capturing the underlying shapes, which are then used to perform cross-modal image retrieval, in particular sketch-based, and object recognition via transfer learning, in particular domain generalization.

Representation. The input to the descriptor extraction process is always resized to a maximum dimensionality of 227×227 pixels. A multi-scale representation is performed by processing at 5 fixed scales, *i.e.*, re-scaling the original input by a factor of $1/2$, $1/\sqrt{2}$, 1, $\sqrt{2}$, 2, and, with the additional mirroring, 10 final instances are produced. *Images* undergo edge detection and the resulting edge map [40] is fed to the CNN[2]. *Sketches* come in the form of strokes, thin line drawings, or brush drawings, depending on the input device or the dataset. To unify the sketch input, a simple morphological filter is applied to a binary sketch image. Specifically, a morphological thinning followed by dilation is performed. After the pre-processing, the sketch is treated as an edge map. As a consequence of the rescaling and mirroring, an image/sketch is mapped to 10 vectors. We refer to these l_2 normalized vectors as EdgeMAC descriptors. They are subsequently mean-pooled or indexed separately, depending on the evaluation benchmark, see Section 9.2 for more details.

Classification. EdgeMAC descriptors are extracted from labeled images and a multi-class linear classifier [122] is trained to perform the task of object recognition. This is especially useful for transfer learning when the training domain is different from the target/testing one. In this case, no labeled images of the training domain are available during the training of our network and no labeled images of the target domain are available during classifier training.

[2] We perform zero padding by 30 pixels to avoid border effects.

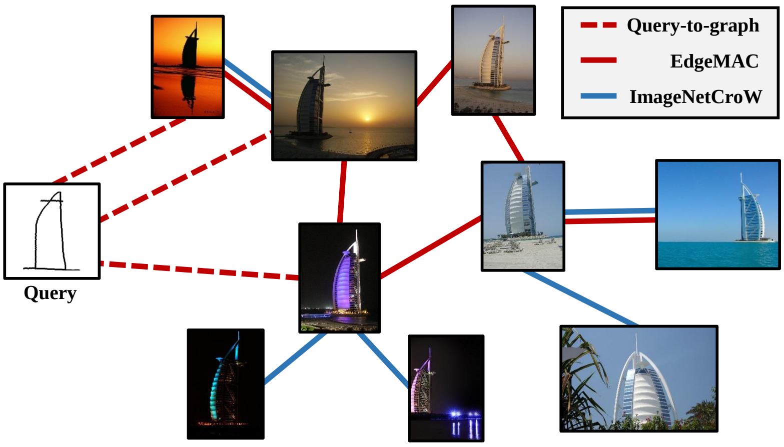


Figure 9.5. Neighborhood graph fusing different similarities. Links (graph edges) of different descriptors are shown in different color and solid line, while the links to the initial nearest neighbors of the query (EdgeMAC only) are shown with dashed line.

Search. An image collection is indexed by simply extracting and storing the corresponding EdgeMAC descriptors for each image. Search is performed by nearest-neighbors search of the query descriptor in the database. This makes retrieval compatible with approximate methods [112, 75] that can speed up search and offer memory savings.

Search boosting. We use the derived representation to perform Query Expansion (QE), which is a popular category of techniques in image retrieval that boost the recall [31]. We employ global diffusion, as proposed by Iscen *et al.* [67], where the ranking is based on a neighborhood graph, which is a mutual kNN-graph of a dataset. We construct the neighborhood graph by combining kNN-graphs built on two different similarities [10, 188]: edge-map similarity and image similarity. The image descriptors are generated using an off-the-shelf CNN [150] and are used only for the kNN-graph construction, unlike [164, 11] where the image descriptors had to be stored together with the sketch descriptors. A toy example of such a graph is illustrated in Figure 9.5.

QE has been used before for sketch-based image retrieval [164, 165], where sketch matching is performed as an initial stage and then only image appearance matching is used to perform QE. A similar concept is used by Bhattacharjee *et al.* [11] who perform max-flow on a graph of top-K retrieved region proposals.

9.1.3. Implementation details

In this section we discuss implementation details. The training dataset used to train our network is presented. We train a single network, which is then used for different tasks. Training sets provided for specific tasks are not exploited.

Training data. We use the training set which comprises landmarks and urban scenes [133]. There are around 8k tuples. Due to the overlap of landmarks contained in the training set and one of the test sets involved in our evaluation, we manually excluded these landmarks from our training data. We end up with with 5,969 tuples for training and 1,696 for validation. Hard negatives are re-mined 3 times per epoch from a pool of around 22k images.

Training implementation. We initialize the convolutional layers by VGG16 [150] (results in 512D EdgeMAC descriptor) trained on ImageNet and sum the filters of the first layer over the feature maps dimension to accommodate for the 2D edge map input instead of the 3D image. The edge-filtering layer is initialized with values $p = 0.5$, $\tau = 0.1$ and β is fixed and equal to 500 so that it always approximates hard thresholding. Additionally, the output of the egde-filtering layer is linearly scaled from $[0, 1]$ to $[0, 10]$. Initial learning rate is $l_0 = 0.001$ with an exponential learning rate decay $l_0 \exp(-0.1j)$ over epoch j; momentum is 0.9; weight decay is 0.0005; contrastive loss margin is 0.7; and batch size is equal to 20 training tuples. All training images are resized so that the maximum extent is 200 pixels, while keeping the original aspect ratio.

Training time. Training is performed for at most 20 epochs and the best network is chosen based on the performance on validation tuples. The whole training takes about 10 hours on a single GeForce GTX TITAN X (Maxwell) GPU with 12GB of memory.

9.2. Experiments

We evaluate EdgeMAC descriptor on domain generalization and sketch-based image retrieval. However, we also go beyond sketch-dependent tasks, and show that our proposed method can be successfully used in any problem where the important information is encoded in the shape. We train the network once and apply it across different tasks proving the generic nature of the representation, *i.e.*, there is no per-task re-training.

9.2.1. Domain generalization through shape matching

In this experiment, we evaluate on domain generalization to validate the effectiveness of our representation on shape matching. The EdgeMAC descriptors are extracted from images, mean-pool descriptors of rescaled and mirrored instances are l_2 normalized to produce one descriptor per image. A linear classifier [122] is trained on the labelled images to perform object recognition.

PACS dataset [85] consists of 10k images coming from 4 domains with varying level of abstraction: *art* (painting), *cartoon*, *photo*, and *sketch*; labeled according to 7 categories: dog, elephant, giraffe, guitar, horse, house, and person. More details on this dataset can be found in Chapter 3, Section 3.3.

For evaluation, each time, one domain is considered unseen, also called *target* or *test* domain, while the images of the other 3 are used for training. Finally, multi-class accuracy is evaluated on the unseen domain. Additionally, we perform classifier training using a single domain and then test on the rest. We find this scenario to be realistic, especially in the case of training on photos and testing on the rest. The domain of realistic photos is the richest in terms of annotated data, while others such as sketches and cartoons are very sparsely annotated.

Baselines. We are interested in translation invariant representations and consider the two following baselines. First, MAC [168] descriptors extracted using a network that is pre-trained on ImageNet. Second, MAC descriptors extracted by a network that is fine-tuned for image retrieval in a siamese manner [133] (presented in Chapter 6). These two baselines have the same descriptor extraction complexity as ours, *i.e.*, multi-scale and mirroring, and are extracted on RGB images, while ours on edge maps. Note, that

Table 9.1. Multi-class accuracy on PACS dataset for 4 different descriptors. The combined descriptor (pre-trained + ours) is constructed via concatenation. A: Art, C: Cartoon, P: Photo, S: Sketch, 3: all 3 other domains.

	Pre-trained (RGB)				Siamese [133] (RGB)				Ours (edge map)				Pre-trained+Ours			
Test →	A	C	P	S	A	C	P	S	A	C	P	S	A	C	P	S
Train A	N/A	59.2	95.0	33.1	N/A	59.5	86.3	42.9	N/A	55.9	61.2	65.6	N/A	61.6	94.9	38.4
Train C	71.7	N/A	86.8	37.0	61.0	N/A	77.0	51.6	45.2	N/A	57.3	74.8	69.3	N/A	85.0	55.3
Train P	72.5	33.3	N/A	24.8	66.0	38.0	N/A	31.9	45.4	42.3	N/A	46.3	73.3	34.0	N/A	27.6
Train S	31.9	49.5	42.5	N/A	38.7	49.3	44.4	N/A	34.8	63.0	43.3	N/A	33.7	59.3	43.4	N/A
Train 3	78.0	68.0	94.4	47.1	71.5	64.3	85.1	56.0	53.8	67.9	64.5	74.7	80.0	68.7	93.7	62.7
Mean 3	71.9				69.2				65.2				76.2			

we treat all domains as images with our approach and extract edge maps, *i.e.*, we do not perform any special treatment on sketches as in the case of sketch retrieval.

Performance comparison. We evaluate our descriptor, the two baselines, and the concatenated version of ours and the descriptor of the pre-trained baseline network, and report results in Table 9.1. Our representation significantly improves sketch recognition while training on a single or all seen domains. Similar improvements are observed for cartoon recognition when training on photos or sketches, while when training on artwork the color information appears to be beneficial. We consider the case of training only on photos and testing on other domains to be the most interesting and realistic one. In this scenario, we provide improvements, compared to the baselines, for sketch recognition (15% and 22%) and cartoon recognition (4% and 9%). Finally, the combined descriptor reveals the complementarity of the representations in several cases, such as artwork and cartoon recognition while training on all seen domains, or training on single domain when artwork is involved, *e.g.*, train on P (or A) and test on A (or C). The best reported score on PACS is 69.2 [85] by fine-tuning AlexNet on PACS. The achieved score by our descriptor with fine-tuned VGG (PACS not used during network training) is 76.2, which is significantly higher. The same experiment with AlexNet achieves 70.9. Performance is reported per category in Figure 9.6. The proposed descriptor achieves significant improvements on most categories for sketch recognition, while the combined is a safe choice in majority of the cases. Interestingly, our experiments reveal that the siamese baseline slightly improves shape matching, despite being trained on RGB images.

Visualization with t-SNE. We use t-distributed Stochastic Neighbor Embedding (t-SNE) [171] to reduce the dimensionality of descriptors to 2 and visualize the result for the pre-trained baseline and our descriptor in Figure 9.7. Different modalities are brought closer with our descriptor. Observe how separated is the sketch modality with the pre-trained network that receives an RGB image for input.

9.2.2. Sketch-based image retrieval

We extract the proposed descriptors to index an image collection, morphologically pre-process sketch queries as described in Section 9.1.2 and perform sketch-based image retrieval via simple nearest neighbor search.

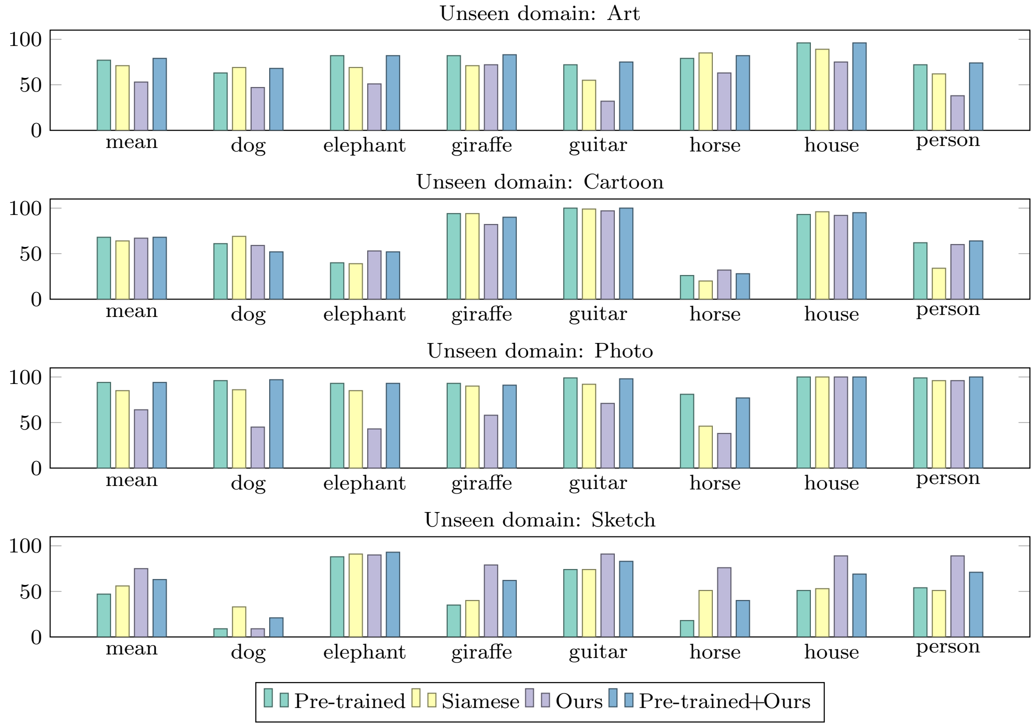


Figure 9.6. Classification accuracy on PACS dataset with different descriptors. Testing is performed on one unseen domain each time, while training is performed on the other three.

Test datasets and evaluation protocols. The method is evaluated on four standard sketch-based image retrieval benchmarks and using the protocols defined with those datasets. We give brief description of those datasets in the following, while the more detailed presentation can be found in Chapter 3, Section 3.2.

Flickr15k [63] consists of 15k images and 330 sketch queries from 33 categories, including particular object instances, generic objects, and shapes. The performance is measured via mean average precision (mAP) [125]. We mean-pool EdgeMAC descriptors of rescaled and mirrored instances and l_2 normalize to produce one descriptor per image. Search is performed by a cosine similarity nearest-neighbor search.

Shoes/Chairs/Handbags [184, 153] datasets contain images from one category only, *i.e.* shoe/chair/handbag category respectively. It consists of pairs of a photo and a corresponding hand-drawn detailed sketch of this photo. There are 115, 97, and 168 sketch–photo pairs for testing shoes, chairs, and handbags, respectively. The performance is measured via the matching accuracy at the top K retrieved images, denoted by acc.@K. We follow the standard protocol [184] which is as follows. Descriptors are extracted from 5 image crops (corners and center) and their horizontally mirrored counterparts. This holds for database images and the sketch query. During search, these 10 descriptors are compared one-to-one and their similarity is averaged. For fair comparison, we adopt this protocol and do not use a single descriptor per image/sketch for this benchmark. However, instead of image crops, we extract EdgeMAC descriptors at 5 image scales and their horizontally mirrored counterparts, as these are defined in Section 9.1.2.

Sketchy [143] test dataset consists of 1,250 database photos and 6,312 query sketches spanning 125 categories of common objects like horse, apple, axe, guitar, *etc.* We evaluate on the test set, but we *do not* use the training set. Each sketch query is

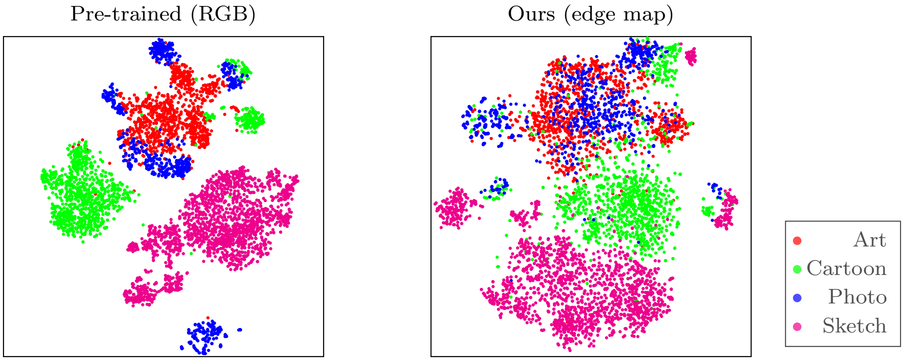


Figure 9.7. Visualization of PACS images with t-SNE (more overlap is better).

Table 9.2. Performance evaluation of the different components of our method on Flickr15k dataset. Network: off-the-shelf (O), fine-tuned (F).

Component	Network							
	O	O	F	F	F	F	F	F
Train/Test: Edge filtering		■	■	■	■	■	■	■
Train: Query binarization				■	■	■	■	■
Test: Mirroring					■		■	■
Test: Multi-scale						■	■	■
Test: Diffusion								■
mAP	25.9	27.9	38.4	41.9	43.4	45.6	46.1	68.9

associated to a single image, the one that prompted the creation of this particular sketch. The performance is measured via recall at various ranks, where recall@K is basically the same as acc.@K of the Shoes/Chairs/Handbags datasets. We mean-pool EdgeMAC of rescaled and mirrored instances and l_2 normalize to produce one descriptor per image. Search is performed by a cosine similarity nearest-neighbor search.

SBIR175 [120] dataset consists of 1.2M images and 175 sketch queries. Query sketches depict objects from 40 different categories. The performance is measured via precision at K top-ranked images per query, and average precision over all queries is reported. We mean-pool EdgeMAC descriptors of rescaled and mirrored instances and l_2 normalize to produce one descriptor per image. Search is performed by a cosine similarity nearest-neighbor search. This dataset has no available annotation, so we use external annotators to manually evaluate the top retrieved images for each query and evaluated method (see Chapter 3, Section 3.2.4 for details on the annotation).

Impact of different components. Table 9.2 shows the impact of different components on the final performance of the proposed method as measured on Flickr15k dataset. Direct application of the off-the-shelf CNN on edge maps already outperforms most prior hand-crafted methods (see Table 9.4). Adding the edge-filtering layer to the off-the-shelf network improves the precision. In this case, the initial parameters for filtering are used. Fine-tuning brings significant jump to 38.4 mAP, which is already the state-of-the-art on this dataset. Random training-query binarization and multi-scale with mirroring representation further improve the mAP score to 46.1. This constitutes our final descriptor which is used throughout all our experiments.

Table 9.3. Performance evaluation on the Flickr15k dataset for different CNN architectures used as a feature extractor and for different edge detectors. Feature extractor: AlexNet [81], VGG [150]. Edge detector: DollarEdge [40], DeepEdge [80]. Evaluation mode: single-scale (SS), multi-scale (MS), mirror (MR).

Architecture			Evaluation mode (mAP)			
Feature extractor	Edge detector					
	Train	Test	SS	SS+MR	MS	MS+MR
AlexNet	DollarEdge	DollarEdge	32.2	34.3	37.9	39.2
		DeepEdge	30.4	32.0	36.2	36.8
	DeepEdge	DollarEdge	32.2	33.9	37.0	38.3
		DeepEdge	31.4	32.9	36.6	37.4
VGG	DollarEdge	DollarEdge	41.9	43.4	45.6	46.1
		DeepEdge	43.1	44.5	46.4	46.9
	DeepEdge	DollarEdge	39.7	40.8	42.6	43.1
		DeepEdge	43.3	44.4	46.2	46.7

Finally, the diffusion process based on the combination of edge-map and image similarity kNN graphs boosts the performance to 68.9 mAP. Image-to-image similarity for the kNN graph is computed based on CroW descriptors [76] extracted from RGB images using the off-the-shelf VGG network. The proposed diffusion is superior to alternative methods, such as average QE on edge-map descriptors (57.3 mAP), average QE on image descriptors (61.7 mAP – needs additional set of descriptors), diffusion on edge-map kNN graph (66.2 mAP), and diffusion on image kNN graph (65.9 mAP).

Impact of different architectures. Table 9.3 shows the impact of using different architectures both for the edge detector part of our pipeline, and the convolutional feature extractor part, as measured on Flickr15k dataset. We experiment with the light-weight AlexNet [81] and the more computationally heavy VGG [150]. Our experiments show that VGG provides a significant performance boost, even though our result on AlexNet already exceeds the state of the art on this dataset (see Table 9.4). We further evaluate performance when using two different edge detectors, the very efficient DollarEdge detector [40], and the more costly DeepEdge detector [80]. We also evaluate cases where one detector is used for training and another during testing. Both detectors achieve similar performance, and changing the detector during testing does not sacrifice the performance. All the rest of our experiments use VGG as feature extractor and DollarEdge as edge detector.

Performance evolution during learning. We report the performance of the fine-tuned network at different stages (epochs) of training. The same network is evaluated for all datasets as we train a single network for all tasks. The performance is shown in Figure 9.8 for three benchmarks. On all datasets, the fine-tuning significantly improves the performance already from the first few epochs.

As a sanity check, we also perform a non-standard sketch-to-sketch evaluation on the Flicker15k and on Sketchy datasets. On the Flickr15k dataset, the 330 sketches are used both as database and as query set (the query sketch is removed from the evaluation), the task is to retrieve sketches of the same category. On Sketchy dataset, all 6,312 sketches form the database, each sketch is also used as a query. The goal

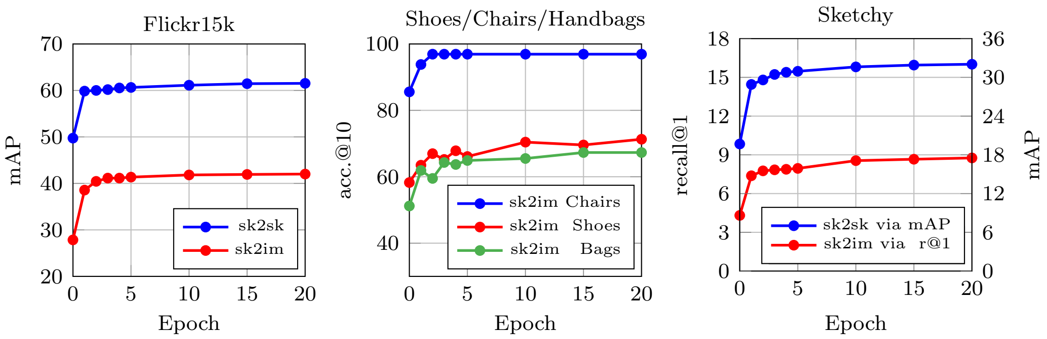


Figure 9.8. Performance evaluation of the fine-tuned network over training epochs for the single-scale representation. All shown datasets and their evaluation protocols are described in Section 9.2.2. Evaluation: sketch-to-image (sk2im), sketch-to-sketch (sk2sk).

is to retrieve sketches generated from the same image as the query sketch. Sketches of the same category but generated for different images are excluded from the query evaluation. The sketch-to-sketch retrieval is evaluated by mAP and the performance is presented in Figure 9.8. The evolution of the performance shows similar behavior as the sketch-to-image search, *i.e.*, the learning on edge maps improves the performance on sketch-to-sketch retrieval.

Comparison with the state of the art. We extensively compare our method with the state-of-the-art performing methods on all before-mentioned benchmarks. Whenever code and trained models are publicly available, we additionally evaluate them on test sets they were not originally applied on. In cases that the provided code is used for evaluation on Flickr15k and SBIR175 we center and align the sketches appropriately in order to achieve high scores, while our method is translation invariant so there is no such need. First we give a short overview of the best performing and most relevant methods. Finally, a comparison via quantitative results is given.

Siamese network [129] is a two-branch network, with a newly proposed architecture that is similar to Sketch-a-Net [185]. Training is performed from scratch with contrastive loss on Flickr15k dataset. Training pairs are selected by randomly choosing a sketch and its category-level positive and negative image. Then, the sketch is fed in one and the image edge map in the other branch.

Shoes/Chairs/Handbags networks [184, 153] are trained from scratch based on the Sketch-a-Net architecture [185]. This is achieved by the following steps [184][3]: (i) Training with classification loss for 1k categories from ImageNet-1K data with edge maps input. (ii) Training with classification loss for 250 categories of TU-Berlin [45] sketch data. (iii) Training a triplet network with shared weights and ranking loss on TU-Berlin sketches and ImageNet images. (iv) Finally, training separate networks for fine-grain instance-level ranking using the Shoes/Chairs/Handbags training datasets. This approach is later improved [153] by adding an attention module with a coarse-fine fusion (CFF) into the architecture, and by extending the triplet loss with a higher order learnable energy function (HOLEF). Such a training involves various datasets, with annotation at different levels, and a variety of task-engineered loss functions. Note that the two models available online achieve higher performance than the ones reported in [184], due to parameter retuning. We compare our results to their best performing models.

[3] Networks/code available at github.com/seuliufeng/DeepSBIR

Table 9.4. Performance comparison via mean Average Precision (mAP) with the state-of-the-art sketch-based image retrieval on the Flickr15k dataset. Best result is highlighted in **red**, second best in **bold**. Query expansion methods are shown below the horizontal line and are highlighted separately. Our evaluation of the methods that do not originally report results on Flickr15 is marked with †.

Hand-crafted methods

Method	Dim	mAP
GF-HOG [63]	n/a	12.2
S-HELO [141]	1296	12.4
HLR+S+C+R [176]	n/a	17.1
GF-HOG extended [14]	n/a	18.2
PerceptualEdge [128]	3780	18.4
LKS [142]	1350	24.5
AFM [165]	243	30.4
AFM+QE [165]	755	**57.9**

CNN-based methods

Method	Dim	mAP
Sketch-a-Net+EdgeBox [11]	5120	27.0
Siamese network [129]	64	19.5
Shoes network [184]†	256	29.9
Chairs network [184]†	256	29.8
Sketchy network [143]†	1024	34.0
Quadruplet network [148]	1024	32.2
Triplet no-share network [15]	128	**36.2**
⋆ EdgeMAC	512	**46.1**
Sketch-a-Net+EdgeBox+GraphQE [11]	n/a	32.3
⋆ EdgeMAC+Diffusion	n/a	**68.9**

TU-Berlin network is a baseline considered in [143]. It is a GoogLeNet [157] network fine-tuned for classification with the 250 sketch categories from TU-Berlin dataset. Edge maps are used as an input for photos during testing time. This is the only network in the work of [143] that is evaluated on Sketchy testset without being trained on its training counter-part.

Sketchy network [143] consists of two asymmetric sketch and image branches, both initialized with GoogLeNet. The training involves the following steps[4]: (i) Training for classification on TU-Berlin sketch dataset. (ii) Separate training of the sketch branch with classification loss on 125 categories of Sketchy dataset and training of the image branch with classification loss on the same categories with additional 1000 Flickr photos per category. (iii) Training both branches in a triplet network with ranking loss on the Sketchy sketch–photo pairs. The last part involves approximately 100k positive and a billion negative pairs.

Quadruplet network [148] tackles the problem in a similar way as Sketchy network, however, they use ResNet-18 [60] architecture with shared weights for both sketch and image branches. The training involves the following steps: (i) Training with classification loss on Sketchy dataset. (ii) Training a network with triplet loss on Sketchy dataset, while mining three different types of triplets.

Triplet no-share network [15] consists of asymmetric sketch and image branches initialized by Sketch-a-Net and AlexNet [81], respectively. The training involves: (i) Separate training of the sketch branch with classification loss on TU-Berlin and training of the image branch with classification loss on ImageNet. (ii) Training a triplet network with ranking loss on TU-Berlin sketches augmented with 25k corresponding photos harvested from the Internet. (iii) Training a triplet network with ranking loss on Sketchy dataset.

Performance comparison. We compare our network with other methods on all the aforementioned benchmarks. Cases that the original publication did not report perfor-

[4] Network/code available at github.com/janesjanes/sketchy

Table 9.5. Performance comparison via accuracy at rank K (acc.@K) with the state-of-the-art sketch-based image retrieval on the Shoes/Chairs test datasets. Best result is highlighted in **red**, second best in **bold**. Note that [184] and [153] train a separate network per object category. †We evaluate the publicly available networks, because the performance is higher than the one originally reported in [184].

Method	Dim	Shoes		Chairs		Handbags	
		acc.@1	acc.@10	acc.@1	acc.@10	acc.@1	acc.@10
BoW-HOG + rankSVM [184]	500	17.4	67.8	28.9	67.0	2.4	10.7
Dense-HOG + rankSVM [184]	200K	24.4	65.2	52.6	93.8	15.5	40.5
Sketch-a-Net + rankSVM [184]	512	20.0	62.6	47.4	82.5	9.5	44.1
CCA-3V-HOG + PCA [180]	n/a	15.8	63.2	53.2	90.3	–	–
AFM [165]	243	32.2	79.1	59.8	89.7	–	–
Shoes net [184]†	256	52.2	**92.2**	65.0	92.8	23.2	59.5
Chairs net [184]†	256	30.4	75.7	72.2	**99.0**	26.2	58.3
Handbags net [153]	256	–	–	–	–	39.9	82.1
Shoes net + CFF + HOLEF [153]	512	**61.7**	**94.8**	–	–	–	–
Chairs net + CFF + HOLEF [153]	512	–	–	**81.4**	95.9	–	–
Handbags net + CFF + HOLEF [153]	512	–	–	–	–	**49.4**	**82.7**
⋆ EdgeMAC	512	40.0	76.5	**85.6**	95.9	35.1	70.8
⋆ EdgeMAC + whitening	512	**54.8**	**92.2**	**85.6**	**97.9**	**51.2**	**85.7**

mance on a particular dataset are evaluated by ourselves by using the publicly available networks. Results on the Flickr15k dataset are presented in Table 9.4, where our method significantly outperforms both hand-crafted descriptors and CNN-based that are learned on a variety of training data. This holds for both plain search with the descriptors, and for methods using re-ranking techniques, such as query expansion [31] and diffusion [67].

Results on the fine-grained Shoes/Chairs/Handbags benchmark are shown in Table 9.5. In this experiment, we also report the performance after applying descriptor whitening which is learned in a supervised way [133] by using the descriptors of the training images of this benchmark. Details on supervised learning of descriptor whitening can be found in Chapter 6, Section 6.1.4. A single whitening transformation is learned for all three datasets. Such a process takes only a few seconds once descriptors are given. It is orders of magnitude faster than using the training set to perform network fine-tuning. We achieve the top performance in 2 out of 3 categories and the second best in the other one. The approach of [184] and [153] train a separate network per category (3 in total), which is clearly not scalable to many objects. In contrast our approach uses a single generic network. We evaluated the publicly available Shoes and Chairs networks on categories they were not trained on. The observd drop in performance, see Table 9.5, confirms that these are single-purpose networks.

On Sketchy dataset, superior results by large margin are achieved by the Sketchy network [143] and the Quadruplet network [148]. These networks are designed for this particular task, using sub-category level annotation during training. Our generic method outperforms all other methods on this dataset, see Table 9.6. Training with sub-category level annotation of the Sketchy dataset appears to be essential for good performance on this dataset. We include detailed discussion and a qualitative comparison in the following paragraphs. On other datasets, the best performing Sketchy [143] and Quadruplet [148] networks are inferior to ours and a number of other approaches.

Table 9.6. Performance comparison via recall at one (recall@1) with the state-of-the-art sketch-based image retrieval on Sketchy testset. Best result is highlighted in **red**, second best in **bold**. Our evaluation of the methods that do not originally report results on Sketchy dataset is marked with †.

Method	Dim	recall@1
GALIF [47]	2500	3.9
Shoes network [184]†	256	6.1
Chairs network [184]†	256	6.5
TU-Berlin network [143]	1024	5.2
Sketchy network [143]	1024	**37.1**
Quadruplet network [148]	1024	**42.2**
⋆ EdgeMAC	512	9.6

Finally, we compare the performance of our network with the state-of-the-art Sketchy network on the large-scale SBIR175 [120] dataset. To better understand the difference between the proposed approach and Sketchy, we further divide the queries by the category of the depicted object into two groups. The first group consists of categories that are used to train the Sketchy network, the other contains categories that the Sketchy network has not seen during the training. In particular, 29 out of 40 categories from this large-scale dataset coincide with categories used in the process of the Sketchy network training. Out of 175 test queries, 146 belong to these 29 categories. The second, smaller group, contains 11 categories and 29 queries. We follow the same procedure as reported by the Sketchy network, *i.e.*, resize sketch queries to a 256 × 256 so that a longer sketch side is occupying 78% of the canvas. This appeared crucial in order to achieve high performance with Sketchy network. Our proposed network has the same parameters as in the other experiments. The quantitative comparison is provided in Table 9.7. Precision at 5, 10, and 25 results respectively was measured. The performance of our method remains the same for both groups of queries. However, the performance of the Sketchy network is remarkably lowered for categories not used in the training. Overall, both networks perform similarly for precision at 5, while Sketchy is better for precision at more images. We conclude, that the Sketchy network strongly benefits from the category recognition, where it achieves better results. The proposed method performs a generic sketch-based image search relying on the shapes and thus generalizes better to unseen categories (as all categories are unseen for our method).

Qualitative comparison. We perform qualitative evaluation on the dataset with the largest scale, *i.e.*, SBIR175 dataset [120]. We first visually demonstrate the bias of the Sketchy network towards categories used to train the category loss and towards specific examples used for those categories. Figure 9.9 shows example queries from the first group, *i.e.* from categories used to train the Sketchy network. We observe that Sketchy tends to find images of the correct category (such as an airplane), or images with objects similar to the category (such as lens for the canon query), but not necessarily of the correct pose or shape. The proposed method finds objects of the correct pose / shape (as with airplanes) or fails (canons) when the shape is not present. We find the result comparison for the *glasses* query quite interesting. Most training examples of the Sketchy network contain images with glasses on faces. These kind of images are correctly retrieved but not the images of glasses without faces (first

Table 9.7. Performance comparison via precision at rank K (p@K), with the state-of-the-art sketch-based image retrieval on the large-scale dataset [120]. †Our evaluation of the Sketchy network [143].

Method	Dim	p@5	p@10	p@25
29 categories (146 queries) appearing in Sketchy dataset				
Sketchy network [143]†	1024	77.4	77.2	76.5
⋆ EdgeMAC	512	69.9	67.0	61.5
11 categories (29 queries) not appearing in Sketchy dataset				
Sketchy network [143]†	1024	37.9	35.9	32.1
⋆ EdgeMAC	512	65.5	63.1	59.6
All 40 categories (175 queries)				
Sketchy network [143]†	1024	70.9	70.3	69.2
⋆ EdgeMAC	512	69.1	66.3	61.1

appears at rank 1779). This happens, despite the fact that these images contain shapes almost identical to the sketch query. Figure 9.10 shows example queries of the second group, *i.e.* categories not seen by the Sketchy network. Our method retrieves objects of similar shape to the query, while Sketchy fails to retrieve relevant images, even though the shape is very simple, such as in the 'tire' query. In some queries, such as the 'bulb' query, the sketch is similar to the hot-air balloon class, which is retrieved by Sketchy.

We further analyze the behavior of the different methods in Figure 9.11 where we show that Sketchy correctly links *glasses* sketch (glasses category was used to train Sketchy) to faces with glasses. However, the second query, *face* sketch (not used to train Sketchy) fails to find faces. Our method in both cases performs shape matching, not always retrieving a correct category. Since for both methods, images and sketches are mapped into the same space, we can also find nearest sketches to query images. The last query in Figure 9.11 shows a query with a *face* (with no glasses) image. Sketchy still classifies that as glasses and the nearest sketches are all three glasses sketches present in the dataset. Our method retrieves similar shapes, starting with all three face sketches first. From this experiment we conclude that the Sketchy network strongly relies on recognition of known classes, while objects of unknown classes will be unavoidable misclassified. Finally, in Figure 9.12 we show examples of queries that are not related to any object category; they are rather characterized as "scenes", where our approach performs much better than Sketchy.

Visualization with t-SNE. We use t-distributed Stochastic Neighbor Embedding (t-SNE) [171] to perform dataset visualization. We jointly visualize hand-drawn sketches and images (based on their edge-maps) to examine their similarities. We repeat the process both for our network and the off-the-shelf network. The illustration is given in Figure 9.13 for Flickr15k.

It is clearly visible that images of the same category are grouped closer after the learning. Also, after the training, query sketches are, in most cases, grouped together with images of the same category, which was not the case with the off-the-shelf representation.

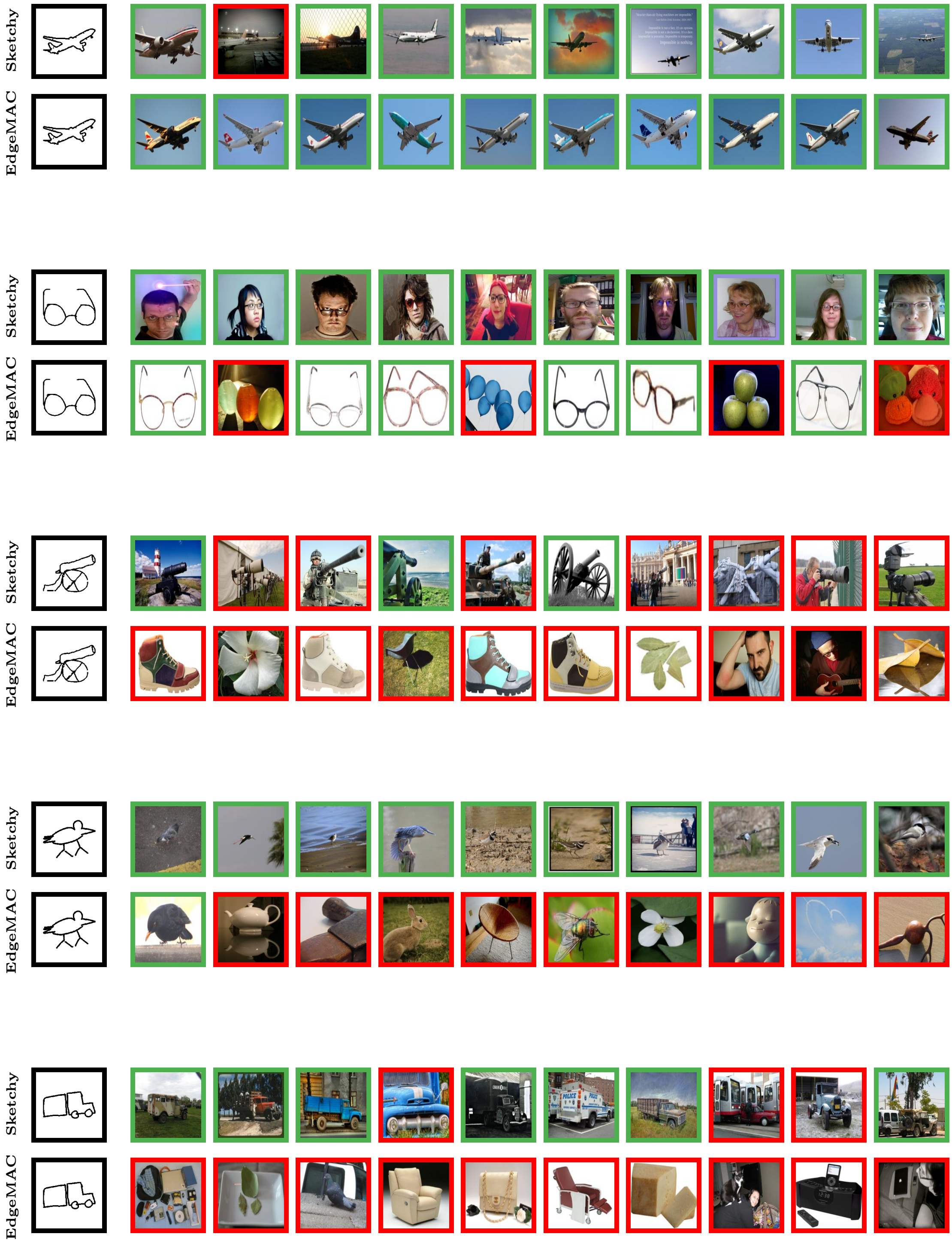


Figure 9.9. Selection of sketch queries on SBIR175 for categories that **appear** in the Sketchy network training set [143]. We show the top ranked images using two networks: Sketchy network [143] (top row), and our network (bottom row).

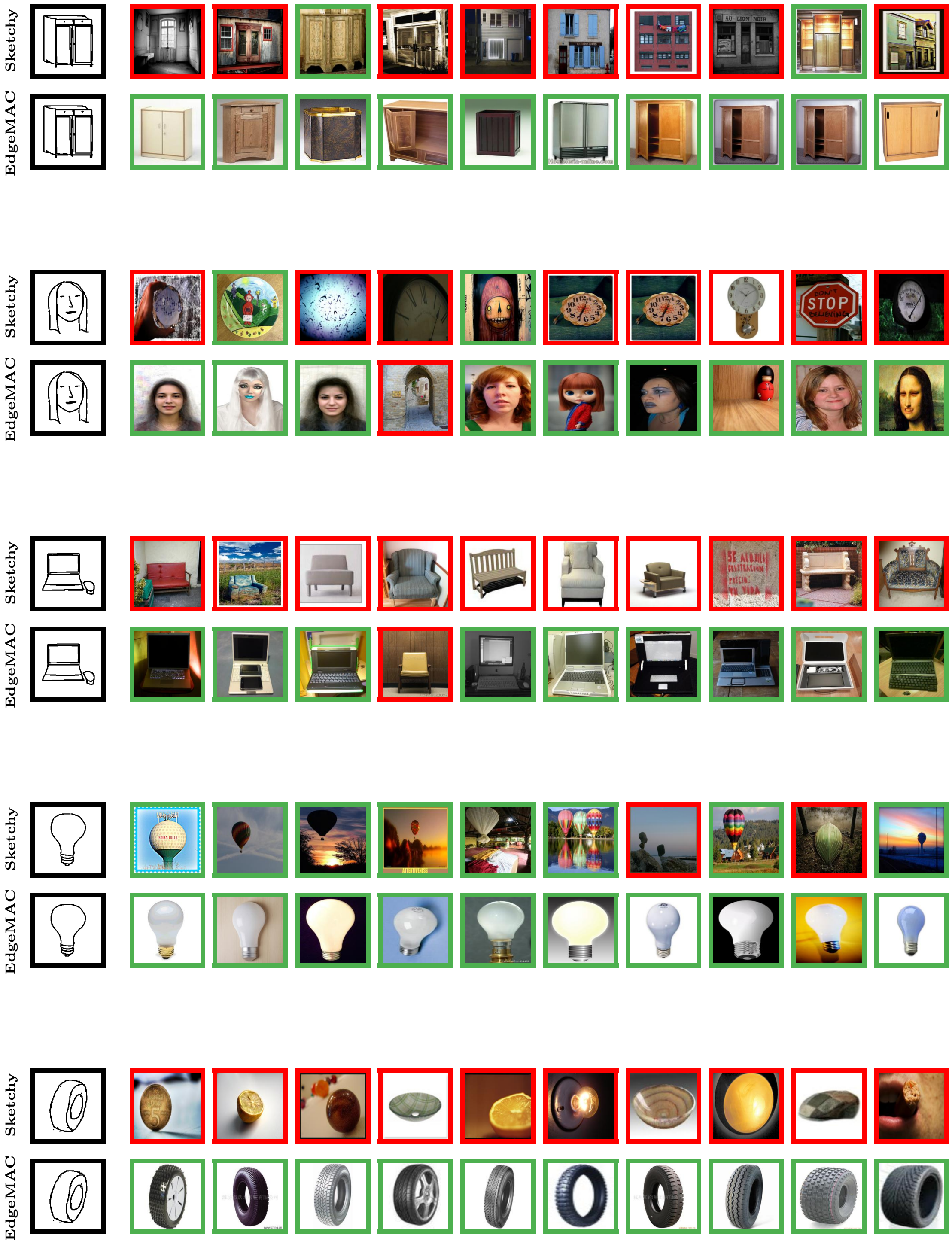


Figure 9.10. Selection of sketch queries on SBIR175 for categories that **do not appear** in the Sketchy network training set [143]. We shows top ranked images using two networks: Sketchy network [143] (top row), and our network (bottom row).

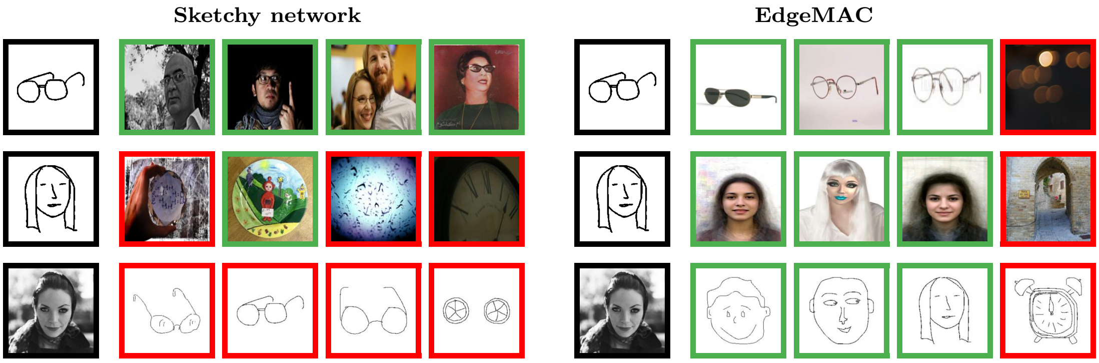


Figure 9.11. Retrieval examples for 2 sketch queries on SBIR175 and for 1 image query searching in the set of sketches (reverse scheme). We show the top ranked images/sketches for the Sketchy network (left) and our network (right). The *glasses* category is part of the Sketchy network training set [143], while the *face* category is not.

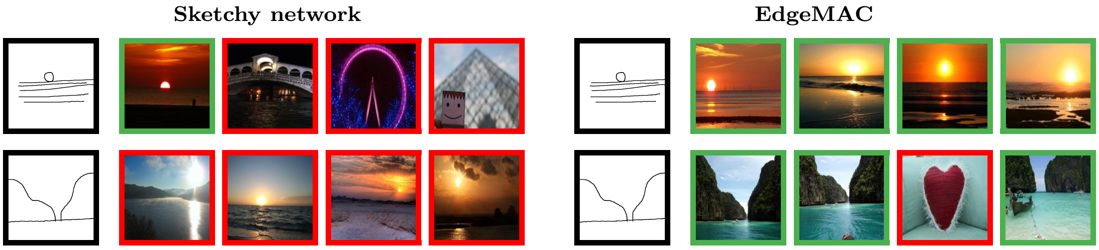


Figure 9.12. Not all sketch queries are object classes. Sketchy network (left) vs our network (right) on non-object queries from Flickr15k.

The number of parameters. Our reported results use the VGG16 network stripped off the fully connected layers (FC), leaving ∼15M parameters. The number of parameters of Sketch-A-Net [185] is ∼8.5M parameters, while when used for SBIR in two different branches (Shoes, Chairs, Handbags [184]) there is ∼17M parameters. Triplet no-share network [15] uses two branches (Sketch-a-Net with additional FC layer and AlexNet [81]) leading to ∼115M, and Sketchy [143] uses 2× GoogLeNet leading to ∼26M parameters. Our network has the smallest number of parameters from the competing methods.

9.2.3. Edge-map to edge-map image retrieval

We depart from the original motivation of domain generalization and sketch-based image retrieval and apply the trained network on image-to-image matching, *i.e.* traditional image retrieval. To demonstrate the applicability of our approach beyond sketches, we downloaded from Google image search images of Oxford and Paris that are difficult to match with known image retrieval methods. These include dark night images, paintings, images with unrealistic colors due to editing, *etc.*. Such images are used as queries for two standard image benchmarks, *i.e.*, Oxford Buildings [125] and Paris [126] datasets, containing 5k and 6k database images respectively, which are described in detail in Chapter 3, Section 3.1.1. We compare the proposed method applied to edge-maps on query and database images (EdgeMAC) with the state-of-the-art image retrieval [133] (RgbMAC) applied to RGB images, presented in Chapter 6. Note that both methods are trained on the same set of images, while the network architecture is the same, as it is based on VGG and MAC layer.

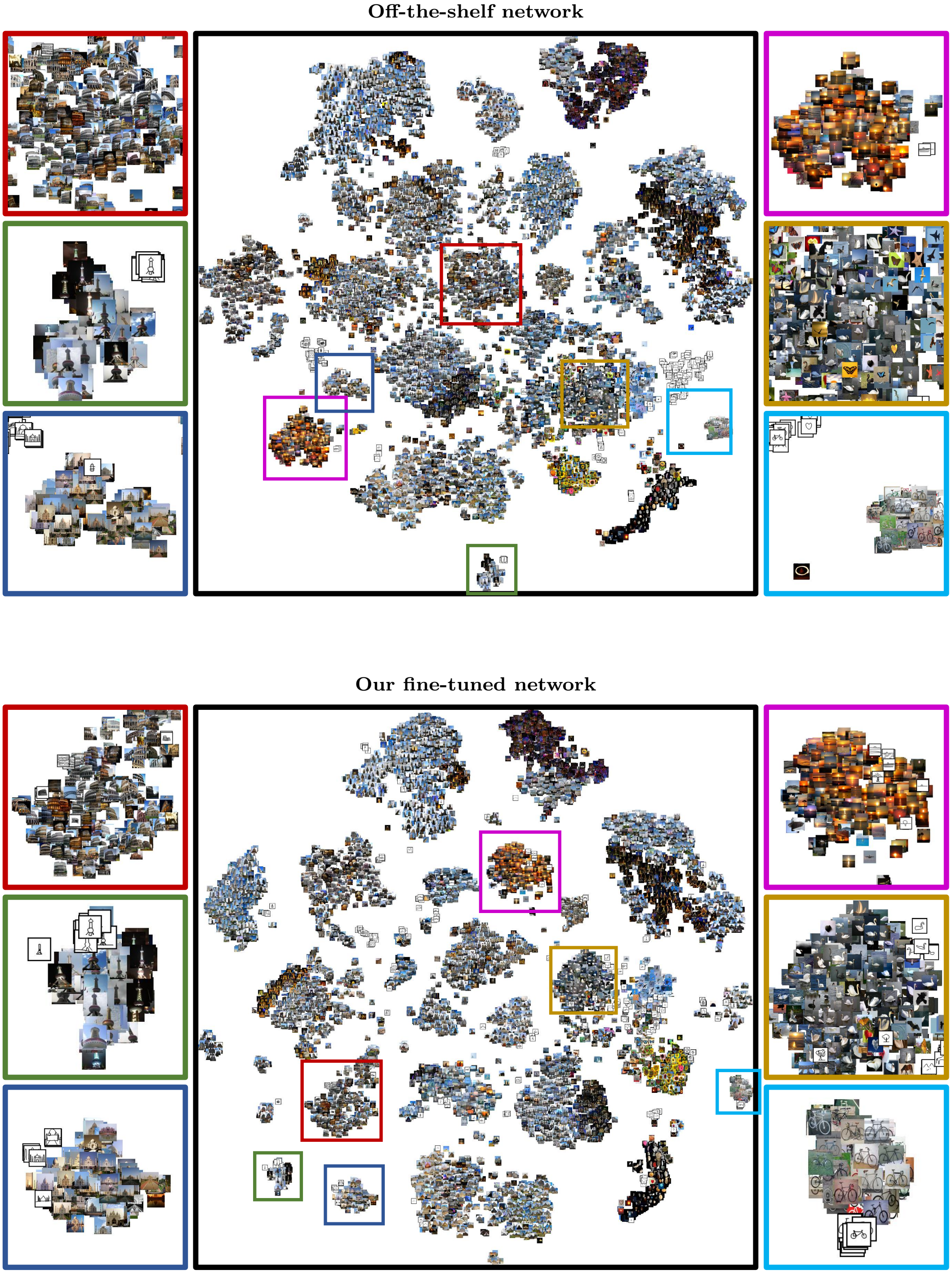


Figure 9.13. Visualization of the Flickr15k dataset with t-SNE in the case of descriptors with the off-the-shelf CNN (top) and our trained network (bottom). Edge-maps (for images) or sketches are fed to the networks in both cases.

A qualitative evaluation is shown in Figure 9.14. EdgeMAC performs significantly better than RgbMAC when the reliable information is only in the shape of the objects. In other examples, *e.g.*, query with a night image, both methods fully retrieve positive images but RgbMAC mostly retrieves images of the same modality, while EdgeMAC gets images from both day and night modality. We conclude that the proposed method has a potential contribution beyond sketch retrieval, in particular in cross-modality retrieval problems and in the presence of significant change in illumination.

9.3. Concluding remarks

We have introduced shape matching for domain generalization and cross-modal retrieval. Images are described by a CNN-based shape descriptor. The network training is perfomed in the domain of landmark images, where the training data is aquired without any manual annotation, by mining image pairs from large scale 3D reconstruction. The network is trained once and then applied to different tasks.

The generic applicability of the representation is supported by validating on a variety of cases. The descriptor is shown beneficial for object recognition via transfer learning, especially to classify images of unseen domains, such as cartoons and sketches, where the amount of annotated data is limited. Remarkably, the same network is applied in all the different tasks. The state-of-the-art results are achieved on standard benchmarks for sketch-based image retrieval, while we have further demonstrated the applicability beyond sketch-based image retrieval. Promising results were shown for queries with different modality (artwork) and significant change of illumination (day-night retrieval). Training data, trained models, and code used in this work are publicly available[5].

[5] cmp.felk.cvut.cz/cnnimageretrieval

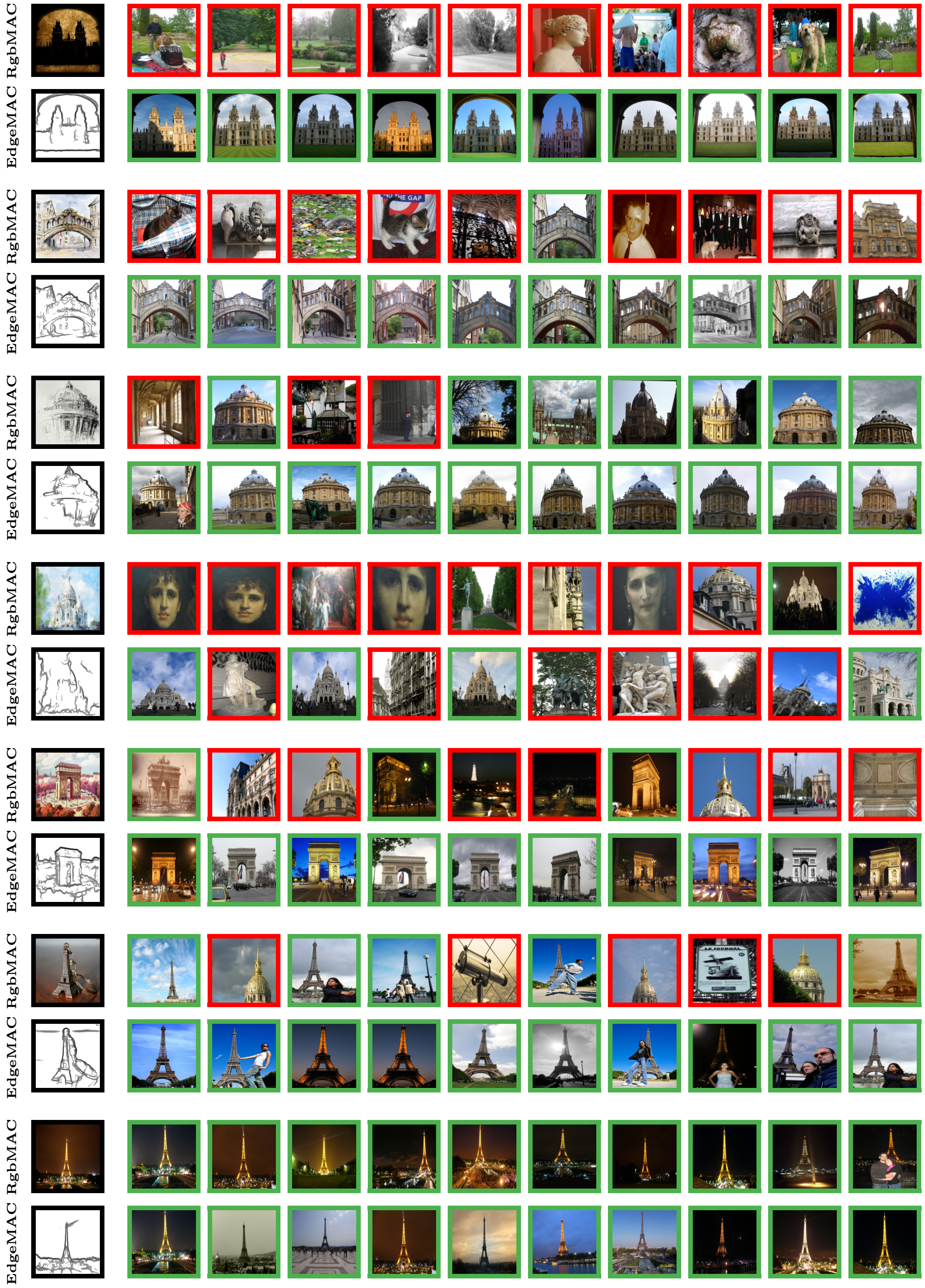


Figure 9.14. Image retrieval examples for query images collected from Google. Retrieval is performed on the Oxford Buildings [125] and Paris [126] datasets with two networks: a network that receives RGB input and is trained for retrieval [133] (top row), and our network that receives edge-map input (bottom row). Edge-maps are used both for query and database images, but we only show it for the query images.

Chapter 10

Conclusions

Various aspects of visual retrieval with compact representations are studied in this thesis. Novel compact representations are proposed in Chapters 5, 6, and 9. In Chapter 5, a variety of vocabulary generation techniques is studied to improve the performance of joint dimensionality reduction of multiple vocabularies for bag-of-words. We show that different combinations of vocabularies, each partitioning the descriptor space in a different yet complementary manner, results in a significant performance improvement. Chapter 6 describes fine-tuning of convolutional neural networks (CNNs) for compact image retrieval without human interaction. We show that more complete and detailed 3D reconstructions are achieved by tightly coupling structure from motion and retrieval, especially retrieval with constraints such as sideways crawl, zoom-out, zoom-in or detail mining. We also show that both hard positive and hard negative examples, selected by exploiting the geometry and the camera positions from the 3D models, enhance the performance of instance image retrieval. Our proposed CNN descriptor whitening discriminatively learned from the same training data outperforms commonly used PCA whitening. We additionally show that our novel trainable generalized-mean (GeM) pooling layer boosts retrieval performance. GeM has become a standard pooling for retrieval, used by a majority of well-performing entries in competitions such as Google Landmark Recognition and Retrieval Challenge 2018. CNN trained with edge maps of landmark images, instead of photographs, is described in Chapter 9. Compact shape representation is learned in this manner, providing improvements on challenging cases of domain generalization, generic sketch-based image retrieval or its fine-grained counterpart. In contrast to other methods that learn a different model per task, object category, or domain, our single network achieves state-of-the-art results in multiple benchmarks.

Chapter 4 addresses the issues of image retrieval benchmarking by expanding existing Oxford Buildings and Paris datasets. In particular, annotation errors, the size of the dataset, and the level of challenge are addressed: new annotation for both datasets is created with an extra attention to the reliability of the ground truth. Three new protocols of varying difficulty are introduced. For each dataset, 15 new challenging queries are introduced. Finally, a new set of 1M hard, semi-automatically cleaned distractors is selected. An extensive comparison of the state-of-the-art methods is performed on the new benchmark in Chapter 7. Different types of methods are evaluated, ranging from local-feature-based to modern CNN-based methods. We conclude that image retrieval is far from being solved, and believe that the newly proposed benchmark will be used to improve future approaches.

Chapter 8 introduces the concept of target mismatch attack for deep-learning-based retrieval systems to generate an adversarial image to conceal the query image. Transfer attacks to fully unseen networks are challenging, and are left for future research. Successful attacks to partially unknown systems are achieved, by designing various loss functions for the adversarial image construction. These include loss functions for unknown global pooling operation or unknown resolution change by the retrieval system.

Appendix A

Common Image Retrieval Terms and Acronyms

Average Precision (AP). The performance for a single query in image retrieval is often evaluated as the average precision (AP) [125] measure computed as the area under the precision-recall curve. Precision is defined as the ratio of retrieved positive images to the total number of images retrieved, while recall is defined as the ratio of the number of retrieved positive images to the total number of positive images in the database. To reach an ideal precision-recall curve, the image retrieval system has to obtain precision 1 over all recall levels, which will result in an average precision equal to 1.

Bag of words (BoW) image representation [151] is an l_2-normalized histogram of occurrences of visual words. Usually, inverse document frequency (*idf*) weighting of visual words is used before the histogram is computed. The dimensionality of BoW representation is equal to the number of clusters in the visual vocabulary.

Contrastive loss. The training input consists of image pairs (i, j) and labels $Y(i, j) \in \{0, 1\}$ declaring whether a pair is non-matching (label 0) or matching (label 1). Contrastive loss acts on matching and non-matching pairs and is defined as [26]

$$\mathcal{L}^{(c)}(i,j) = \begin{cases} \frac{1}{2}||\bar{\mathbf{f}}(i) - \bar{\mathbf{f}}(j)||^2, & \text{if } Y(i,j) = 1 \\ \frac{1}{2}\left(\max\{0, \tau - ||\bar{\mathbf{f}}(i) - \bar{\mathbf{f}}(j)||\}\right)^2, & \text{if } Y(i,j) = 0 \end{cases} \tag{A.1}$$

where $\bar{\mathbf{f}}(i)$ is the l_2 normalzied global representation vector of image i, and τ is a margin parameter defining when non-matching pairs have large enough distance in order to be ignored by the loss.

Cosine similarity for two vectors, $\mathbf{x}$, $\mathbf{y} \in \mathbb{R}^D$ is defined as:

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{D} x_i y_i}{||\mathbf{x}||||\mathbf{y}||}, \tag{A.2}$$

where $||\mathbf{x}||$ is the l_2 norm of vector $\mathbf{x}$. If the vectors are l_2 normalized, cosine similarity and Euclidean distance have a monotonic relation, therefore, they can be interchanged to provide identical ordering of the results:

$$||\mathbf{x} - \mathbf{y}|| = \sqrt{2 - 2\cos(\mathbf{x}, \mathbf{y})}. \tag{A.3}$$

CroW. Cross-dimensional weighted (CroW) representation [76] is obtained by weighted sum pooling over all locations of a convolutional activations tensor. Weights are applied both spatial- and channel-wise. See also global image representation.

Fisher vectors image representation [123] is created by modelling the visual words with a Gaussian mixture model (GMM), restricted to diagonal variance matrices for each of the k components of the mixture. Deriving a diagonal approximation of the Fisher matrix of a GMM, $(2d+1) \times k - 1$ dimensional image representation is obtained, or $d \times k$ dimensional when considering only the components associated with either the means or the variances of the GMM, where d corresponds to the local descriptor dimensionality and k is the visual vocabulary size. See also global image representation.

GeM. Generalized-mean (GeM) representation [135] is obtained by a spatial generalized-mean pooling over all locations of a convolutional activations tensor. See also global image representation.

Global image representation encodes the whole image into a single D dimensional vector, $\mathbf{x} \in \mathbb{R}^D$. Global representation is often computed by local-feature or convolutional-neural-network-feature aggregation. Examples of local-feature-based global representations are BoW [151], Fisher vectors [123], VLAD [72], *etc.* Examples of convolutional-neural-network-based global representations are MAC [138, 168], SPoC [8], CroW [76], R-MAC [168], GeM [135], NetVLAD [3].

Hamming embedding (HE) improves the visual vocabulary by subdividing its clusters [69]. This results in binary signatures associated to every visual word that refines the local descriptor matching quality. HE is further extended by the binarized selective match kernel (SMK*) [163] that uses an inverted file structure to separately index binarized residual vectors while it performs the matching with a selective monomial kernel function. Binarized version of the aggregated selective match kernel [163] (ASMK*) additionally extends SMK* by jointly encoding local descriptors that are assigned to the same visual word.

Hellinger kernel for two l_1 normalized vectors, $\mathbf{x}$, $\mathbf{y} \in \mathbb{R}^D$, is defined as:

$$H(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{D} \sqrt{x_i y_i}. \tag{A.4}$$

Square-rooting l_1 normalized vectors makes them l_2 normalized, so the Euclidean distance can be used instead of Hellinger kernel, to achieve the same result:

$$||\sqrt{\mathbf{x}} - \sqrt{\mathbf{y}}|| = \sqrt{2 - 2H(\mathbf{x}, \mathbf{y})}. \tag{A.5}$$

Local features and descriptors. A distinctive pattern or structure found in an image is reffered to as local feature. The features usually differ from its immediate surroundings by texture, color, or intensity. Popular local-feature detectors are multi-scale Hessian-Affine [121], Harris-Affine [100], MSER [96], *etc.* Local features are represented by a position in the image, scale, and shape (usually an ellipse). Such features are typically geometrically normalized into a cannonical form. The cannonical form is transformed into a compact vector representation denoted as a local descriptor. Most popular and widely used local descriptors in image retrieval are SIFT [92], RootSIFT [4], or HardNet [104]. One image is typically represented by a few thousand local features and descriptors. See also bag of words (BoW).

MAC. Maximum activations of convolutions (MAC) representation [138, 168] is obtained by a spatial max pooling over all locations of a convolutional activations tensor. See also global image representation.

Multi-scale image representation with CNN. In order to improve robustness of the image representation to scale change, image is typically resized to a set of pre-defined scales and each re-scaled version is fed to the network. Finally, the resulting descriptors are combined into a single descriptor by average pooling [58], or generalized-mean pooling [135].

Nearest neighbor (NN) search is an optimization problem of finding the data point in a dataset that is the closest (or the most similar) to a given point (query). Often, the top k nearest neighbors to the query are identified. This is referred to as k-nearest neighbor (k-NN) search. Commonly used similarity metrics in image retrieval NN search are cosine similarity, or Hellinger kernel.

NetVLAD. A generalized VLAD representation NetVLAD [3] treats a convolutional activations tensor as a set of local descriptors, which are describing each spatial location. Then, the descriptor aggregation is performed in a similar fashion as VLAD [72], in a differentiable manner. See also global image representation.

Off-the-shelf and fine-tuned convolutional neural network (CNN). Networks that are pre-trained for some other task, often for image classification on ImageNet [140], are denoted as off-the-shelf, and can be directly applied on image retrieval. Starting from pre-trained ones, networks are additionally trained with the metric suitable for the image retrieval task, and these are denoted as fine-tuned. Metric learning losses commonly used in image retrieval are contrastive loss [26] and triplet loss [24].

Pooling of CNN features. A common practice with CNN image retrieval is to consider a convolutional feature map, represented by a 3D tensor, and perform a pooling mechanism to construct a global image descriptor. The global pooling also introduces translation invariance which is in contrast with fully-connected layers typically used in classification. The pooling is always applied on the last convolutional feature map. Most popular and widely adopted pooling schemes are MAC [138, 168], SPoC [8], CroW [76], R-MAC [168], GeM [135], NetVLAD [3], *etc.* Usually, the resulting global image representation is l_2 normalized.

Query expansion (QE) with BoW typically uses spatial verification to select true positive among the top retrieved result. The estimated image-to-image mapping is then used to back-project local features into the query region [31]. New and enhanced query is finally issued.

Query expansion (QE) with CNN. Similarly as with local-features-based QE, top-ranked images after initial ranking are selected, and their CNN representations are averaged together with the query image, thus creating a new improved query representation. Instead of standard averaging of the descriptors, weighted average can be used, where the weights depend on the similarity between the query and the retrieved image [135]. This is denoted as α query expansion (αQE).

R-MAC. Regional maximum activations of convolutions (R-MAC) representation [168] first performs a spatial max pooling over convolutional activations regions, and finally sum pooling of the regional descriptors. See also global image representation.

Spatial verification (SP) utilizes the location, and possibly scale and shape of features, to verify the spatial consistency between the query and top-ranked retrieved images [125]. This is achieved by a fast and robust hypothesize-and-test procedure that estimates an affine transformation between the query and the target image, often performed with the random sample consensus (RANSAC) algorithm [49]. SP result is the number of inlier correspondences, which is one of the most intuitive similarity measures and allows to detect true positive images with a high confidence.

SPoC. Sum-pooled convolutional (SPoC) representation [8] is obtained by a spatial sum pooling over all locations of a convolutional activations tensor. See also global image representation.

Tf-idf. Term frequency-inverse document frequency. See bag of words (BoW).

Triplet loss. The training input consists of image triplets $(q, m(q), n(q))$, and $\bar{\mathbf{f}}(q)$, $\bar{\mathbf{f}}(m(q))$, $\bar{\mathbf{f}}(n(q))$ are the l_2 normalzied global representation vectors of query image q, and its matching $m(q)$ and non-matching $n(q)$ image. Triplet loss is then defined as [24]

$$\mathcal{L}^{(t)}(q, m(q), n(q)) = \max\{0, ||\bar{\mathbf{f}}(q) - \bar{\mathbf{f}}(m(q))||^2 - ||\bar{\mathbf{f}}(q) - \bar{\mathbf{f}}(n(q))||^2 + \tau\}, \tag{A.6}$$

where τ is a margin parameter defining zero loss when the distance between the query and the non-matching image is greater by a margin than the distance between the query and the matching image.

Visual vocabulary (codebook) and visual words. A set of local descriptors is clustered into k distinct clusters, and those clusters form a visual vocabulary (codebook). This can be done by a standard k-means [151], hierarchical k-means [113], or approximate k-means [125] algorithm. The objective is to vector quantize a local descriptor into a single cluster id value, denoted as visual word. Term *visual vocabulary size* corresponds to the number k of clusters in the visual vocabulary.

VLAD. Vector of Locally Aggregated Descriptors image representation [72] is created by aggregating statistics of the local descriptors beyond a simple histogram. The residual vectors between descriptors and the closest centroid are aggregated *w.r.t.* a visual vocabulary. The dimensionality of VLAD representation is equal to the product of the local descriptor dimensionality and the visual vocabulary size. See also global image representation.

Whitening of descriptors. Whitening the data representation is known to be very essential for image retrieval [68]. Interpretation of [68] lies in down-weighting co-occurrences and, thus, handling the problem of over-counting. Whitening is commonly learned in a generative manner. More specifically, it is learned in an unsupervised way by PCA on an independent dataset. Additionally, whitening can be learned in a discriminative manner, using the training data with pairs of matching descriptors [135]. Whitening improves performance of local descriptors, local-feature-based image representations such as BoW [151] and VLAD [72], and CNN-based image representations such as MAC [138, 168], SPoC [8], CroW [76], R-MAC [168], GeM [135], NetVLAD [3].

Bibliography

[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building Rome in a day. *Commun. ACM*, 2011. 56

[2] N. Akhtar and A. Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 2018. 81

[3] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. *TPAMI*, 2017. 15, 16, 29, 50, 54, 60, 63, 69, 71, 73, 74, 76, 82, 95, 96, 118, 119, 120

[4] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012. 7, 13, 17, 32, 38, 43, 56, 71, 118

[5] R. Arandjelovic and A. Zisserman. All about VLAD. In *CVPR*, 2013. 14, 46, 47, 48

[6] H. Azizpour, A. Sharif Razavian, J. Sullivan, A. Maki, and S. Carlsson. From generic to specific deep representations for visual recognition. In *CVPRW*, 2015. 49

[7] A. Babenko and V. Lempitsky. Additive quantization for extreme vector compression. In *CVPR*, 2014. 37

[8] A. Babenko and V. Lempitsky. Aggregating deep convolutional features for image retrieval. In *ICCV*, 2015. 8, 15, 16, 17, 29, 49, 50, 52, 54, 55, 65, 69, 71, 73, 74, 82, 118, 119, 120

[9] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *ECCV*, 2014. 13, 15, 16, 50, 65

[10] S. Bai, S. Sun, X. Bai, Z. Zhang, and Q. Tian. Smooth neighborhood structure mining on multiple affinity graphs with applications to context-sensitive similarity. In *ECCV*, 2016. 98

[11] S. D. Bhattacharjee, J. Yuan, W. Hong, and X. Ruan. Query adaptive instance search using object sketches. In *ACM Multimedia*, 2016. 18, 98, 105

[12] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006. 14, 37, 39

[13] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *NIPS*, 2016. 19

[14] T. Bui and J. Collomosse. Scalable sketch-based image retrieval using color gradient features. In *ICCV*, 2015. 18, 94, 105

[15] T. Bui, L. Ribeiro, M. Ponti, and J. Collomosse. Generalisation and sharing in triplet convnets for sketch based visual search. In *arXiv:1611.05301*, 2016. 18, 105, 111

[16] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SISC*, 1995. 81

[17] M. Cadik, J. Vasicek, M. Hradis, F. Radenovic, and O. Chum. Camera elevation estimation from a single mountain landscape photograph. In *BMVC*, 2015. 11

[18] X. Cao, H. Zhang, S. Liu, X. Guo, and L. Lin. SYM-FISH: A symmetry-aware flip invariant sketch histogram shape descriptor. In *ICCV*, 2013. 18

[19] Y. Cao, C. Wang, Z. Li, L. Zhang, and L. Zhang. Spatial-bag-of-features. In *CVPR*, 2010. 71

[20] Y. Cao, C. Wang, L. Zhang, and L. Zhang. Edgel index for large-scale sketch-based image search. In *CVPR*, 2011. 18

[21] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou. Hidden voice commands. In *USENIX Security*, 2016. 80

[22] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *SSP*, 2017. 17, 80, 81

[23] A. Chalechale, G. Naghdy, and A. Mertins. Sketch-based image matching using angular partitioning. *IEEE Trans. Systems, Man, and Cybernetics*, 2005. 18

[24] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *JMLR*, 2010. 54, 119, 120

[25] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool. Domain adaptive faster R-CNN for object detection in the wild. In *CVPR*, 2018. 19

[26] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005. 15, 54, 96, 117, 119

[27] O. Chum and J. Matas. Large-scale discovery of spatially related images. *TPAMI*, 2010. 50, 56, 61

[28] O. Chum and J. Matas. Unsupervised discovery of co-occurrence in sparse high dimensional data. In *CVPR*, 2010. 39

[29] O. Chum, A. Mikulik, M. Perdoch, and J. Matas. Total recall II: Query expansion revisited. In *CVPR*, 2011. 29, 69

[30] O. Chum, M. Perdoch, and J. Matas. Geometric min-hashing: Finding a (thick) needle in a haystack. In *CVPR*, 2009. 71

[31] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total Recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, 2007. 13, 14, 17, 29, 33, 56, 71, 72, 98, 106, 119

[32] N. Cohen, O. Sharir, and A. Shashua. Deep SimNets. In *CVPR*, 2016. 16

[33] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 1994. 39

[34] E. J. Crowley and A. Zisserman. The state of the art: Object retrieval in paintings using discriminative regions. In *BMVC*, 2014. 19

[35] E. J. Crowley and A. Zisserman. The art of detection. In *ECCV*, 2016. 19

[36] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 18

[37] T.-T. Do, E. Kijak, L. Amsaleg, and T. Furon. Security-oriented picture-in-picture visual modifications. In *ICMR*, 2012. 17

[38] T.-T. Do, E. Kijak, T. Furon, and L. Amsaleg. Challenging the security of content-based image retrieval systems. In *MMSP*, 2010. 17

[39] P. Dollar, Z. Tu, P. Perona, and S. Belongie. Integral channel features. In *BMVC*, 2009. 52

[40] P. Dollar and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013. 95, 97, 103

[41] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014. 49

[42] W. Dong, R. Socher, L. Li-Jia, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 25, 95

[43] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In *CVPR*, 2018. 17, 80

[44] M. Douze, H. Jegou, H. Sandhawalia, L. Amsaleg, and C. Schmid. Evaluation of GIST descriptors for web-scale image search. In *CIVR*, 2009. 22, 85

[45] M. Eitz, J. Hays, and M. Alexa. How do humans sketch objects? *ACM Trans. Graphics*, 2012. 24, 25, 27, 104

[46] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. An evaluation of descriptors for large-scale image retrieval from sketched feature lines. *Computers & Graphics*, 2010. 18

[47] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, and M. Alexa. Sketch-based shape retrieval. *ACM Trans. Graphics*, 2012. 18, 107

[48] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes challenge 2007 (VOC2007) results, 2007. 44

[49] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981. 14, 72, 120

[50] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building Rome on a cloudless day. In *ECCV*, 2010. 56

[51] S. Gammeter, L. Bossard, T. Quack, and L. Van Gool. I know what you did last summer: object-level auto-annotation of holiday snaps. In *CVPR*, 2009. 29

[52] M. Ghifary, W. Bastiaan Kleijn, M. Zhang, and D. Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *ICCV*, 2015. 19

[53] R. Girshick. Fast R-CNN. In *CVPR*, 2015. 19

[54] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 15, 49, 61

[55] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*, 2014. 8, 13, 15, 16, 49, 50

[56] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 17

[57] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. In *ECCV*, 2016. 16, 23, 69, 95, 96

[58] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. End-to-end learning of deep visual representations for image retrieval. *IJCV*, 2017. 16, 17, 29, 35, 55, 62, 63, 66, 67, 69, 71, 73, 74, 75, 76, 95, 119

[59] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006. 15, 73

[60] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 50, 52, 62, 72, 87, 105

[61] E. Hoffer and N. Ailon. Deep metric learning using triplet network. In *ICLRW*, 2015. 15, 54

[62] J. Hu, J. Lu, and Y.-P. Tan. Discriminative deep metric learning for face verification in the wild. In *CVPR*, 2014. 15

[63] R. Hu and J. Collomosse. A performance evaluation of gradient field hog descriptor for sketch based image retrieval. *CVIU*, 2013. 18, 23, 24, 101, 105

[64] M. J. Huiskes and M. S. Lew. The MIR flickr retrieval evaluation. In *ICMR*, 2008. 25

[65] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer. DenseNet: Implementing efficient ConvNet descriptor pyramids. In *arXiv:1404.1869*, 2014. 49

[66] A. Iscen, Y. Avrithis, G. Tolias, T. Furon, and O. Chum. Fast spectral ranking for similarity search. In *CVPR*, 2018. 34

[67] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum. Efficient diffusion on region manifolds: Recovering small objects with compact CNN representations. In *CVPR*, 2017. 29, 32, 34, 74, 98, 106

[68] H. Jegou and O. Chum. Negative evidences and co-occurrences in image retrieval: The benefit of PCA and whitening. In *ECCV*, 2012. 7, 13, 14, 15, 16, 37, 38, 39, 40, 41, 42, 46, 48, 50, 65, 66, 72, 74, 120

[69] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008. 13, 14, 22, 29, 34, 38, 62, 72, 85, 118

[70] H. Jegou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *CVPR*, 2009. 14, 39, 72

[71] H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *TPAMI*, 2011. 37

[72] H. Jegou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010. 13, 14, 35, 37, 46, 72, 118, 119, 120

[73] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *TPAMI*, 2012. 14, 39, 46, 71

[74] H. Jegou and A. Zisserman. Triangulation embedding and democratic aggregation for image search. In *CVPR*, 2014. 14, 46, 47, 48

[75] J. Johnson, M. Douze, and H. Jegou. Billion-scale similarity search with GPUs. In *arXiv:1702.08734*, 2017. 98

[76] Y. Kalantidis, C. Mellina, and S. Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In *ECCVW*, 2016. 8, 15, 16, 17, 29, 49, 50, 55, 69, 71, 73, 82, 103, 117, 118, 119, 120

[77] Y. Kalantidis, G. Tolias, Y. Avrithis, M. Phinikettos, E. Spyrou, P. Mylonas, and S. Kollias. VIRaL: Visual image retrieval and localization. *MTA*, 2011. 33

[78] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012. 19

[79] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 62, 81

[80] I. Kokkinos. Pushing the boundaries of boundary detection using deep learning. In *ICLR*, 2016. 95, 103

[81] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 47, 49, 50, 52, 62, 72, 87, 95, 103, 105, 111

[82] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In *ICLRW*, 2017. 17

[83] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 1998. 29

[84] C.-Y. Lee, P. W. Gallagher, and Z. Tu. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In *AISTATS*, 2016. 16

[85] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, 2017. 19, 27, 99, 100

[86] H. Li, S. J. Pan, S. Wang, and A. C. Kot. Domain generalization with adversarial feature learning. In *CVPR*, 2018. 19

[87] J. Li, R. Ji, H. Liu, X. Hong, Y. Gao, and Q. Tian. Universal perturbation attack against image retrieval. In *arXiv:1812.00552*, 2018. 9, 17, 80, 82

[88] Y. Li, T. M. Hospedales, Y.-Z. Song, and S. Gong. Fine-grained sketch-based image retrieval by matching deformable part models. In *BMVC*, 2014. 18

[89] Y. Li, N. Snavely, and D. P. Huttenlocher. Location recognition using prioritized feature matching. In *ECCV*, 2010. 59

[90] L. Liu, F. Shen, Y. Shen, X. Liu, and L. Shao. Deep sketch hashing: Fast free-hand sketch-based image retrieval. In *CVPR*, 2017. 18

[91] Z. Liu, Z. Zhao, and M. Larson. Who's afraid of adversarial queries? the impact of image modifications on content-based image retrieval. In *arXiv:1901.10332*, 2019. 9, 17, 80

[92] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 7, 13, 17, 38, 43, 118

[93] C. Ma, X. Yang, C. Zhang, X. Ruan, M.-H. Yang, and O. Coporation. Sketch retrieval via dense stroke features. In *BMVC*, 2013. 18

[94] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, 2015. 84, 90

[95] M. Mancini, S. R. Bulo, B. Caputo, and E. Ricci. Best sources forward: domain generalization through source-specific nets. In *ICIP*, 2018. 19

[96] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 2004. 7, 13, 38, 41, 44, 118

[97] K. Mikolajczyk and J. Matas. Improving descriptors for fast tree matching by optimal linear projection. In *ICCV*, 2007. 16, 54, 74

[98] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 2004. 7, 13, 38

[99] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *TPAMI*, 2005. 71

[100] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 2005. 38, 45, 56, 71, 118

[101] A. Mikulik, O. Chum, and J. Matas. Image retrieval for online browsing in large image collections. In *SISAP*, 2013. 56

[102] A. Mikulik, M. Perdoch, O. Chum, and J. Matas. Learning vocabularies over a fine quantization. *IJCV*, 2013. 29, 37, 39, 56, 69, 71

[103] A. Mikulik, F. Radenovic, O. Chum, and J. Matas. Efficient image detail mining. In *ACCV*, 2014. 11, 56

[104] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *NIPS*, 2017. 11, 118

[105] D. Mishkin, F. Radenovic, and J. Matas. Repeatability is not enough: Learning affine regions via discriminability. In *ECCV*, 2018. 11

[106] E. Mohedano, K. McGuinness, N. E. O'Connor, A. Salvador, F. Marques, and X. Giro-i Nieto. Bags of local convolutional features for scalable instance search. In *ICMR*, 2016. 16, 69, 71, 82

[107] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. In *CVPR*, 2017. 17, 80

[108] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. DeepFool: a simple and accurate method to fool deep neural networks. In *CVPR*, 2016. 17, 80

[109] O. Morere, J. Lin, A. Veillard, L.-Y. Duan, V. Chandrasekhar, and T. Poggio. Nested invariance pooling and RBM hashing for image instance retrieval. In *ICMR*, 2017. 16

[110] A. Mousavian and J. Kosecka. Deep convolutional features for image based retrieval and scene categorization. In *arXiv:1509.06033*, 2015. 16, 50

[111] K. Muandet, D. Balduzzi, and B. Scholkopf. Domain generalization via invariant feature representation. In *ICML*, 2013. 19

[112] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*, 2009. 98

[113] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006. 13, 29, 44, 120

[114] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large-scale image retrieval with attentive deep local features. In *ICCV*, 2017. 16, 72, 84

[115] E.-J. Ong, S. Husain, and M. Bober. Siamese network of deep fisher-vector descriptors for image retrieval. In *arXiv:1702.00338*, 2017. 16, 69, 82

[116] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014. 49

[117] G. Papandreou, I. Kokkinos, and P.-A. Savalle. Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection. In *CVPR*, 2015. 52

[118] N. Papernot, P. McDaniel, and I. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. In *arXiv:1605.07277*, 2016. 17

[119] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *ASIACCS*, 2017. 17

[120] S. Parui and A. Mittal. Similarity-invariant sketch-based image retrieval in large databases. In *ECCV*, 2014. 18, 25, 102, 107, 108

[121] M. Perdoch, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *CVPR*, 2009. 7, 13, 23, 29, 38, 41, 44, 71, 73, 118

[122] F. Perronnin, Z. Akata, Z. Harchaoui, and C. Schmid. Towards good practice in large-scale learning for image classification. In *CVPR*, 2012. 97, 99

[123] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007. 14, 118

[124] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier. Large-scale image retrieval with compressed Fisher vectors. In *CVPR*, 2010. 14, 37, 39, 43, 46, 71

[125] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. 7, 9, 13, 14, 21, 22, 23, 24, 29, 31, 33, 34, 38, 56, 62, 71, 72, 93, 101, 111, 114, 117, 120

[126] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008. 9, 14, 21, 29, 34, 38, 62, 71, 111, 114

[127] J. Philbin, J. Sivic, and A. Zisserman. Geometric latent dirichlet allocation on a matching graph for large-scale image datasets. *IJCV*, 2011. 50

[128] Y. Qi, Y.-Z. Song, T. Xiang, H. Zhang, T. Hospedales, Y. Li, and J. Guo. Making better use of edges via perceptual grouping. In *CVPR*, 2015. 18, 105

[129] Y. Qi, Y.-Z. Song, H. Zhang, and J. Liu. Sketch-based image retrieval via siamese convolutional neural network. In *ICIP*, 2016. 18, 104, 105

[130] F. Radenovic, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. Revisiting Oxford and Paris: Large-scale image retrieval benchmarking. In *CVPR*, 2018. 9, 10, 11, 12, 30, 71, 85

[131] F. Radenovic, H. Jegou, and O. Chum. Multiple measurements and joint dimensionality reduction for large scale image search with short vectors. In *ICMR*, 2015. 9, 11, 12, 37

[132] F. Radenovic, J. L. Schonberger, D. Ji, J.-M. Frahm, O. Chum, and J. Matas. From dusk till dawn: Modeling in the dark. In *CVPR*, 2016. 10, 11, 12, 31, 51, 57, 96

[133] F. Radenovic, G. Tolias, and O. Chum. CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples. In *ECCV*, 2016. 10, 11, 12, 51, 62, 73, 74, 75, 76, 93, 95, 96, 98, 99, 100, 106, 111, 114

[134] F. Radenovic, G. Tolias, and O. Chum. Deep shape matching. In *ECCV*, 2018. 10, 11, 12, 94

[135] F. Radenovic, G. Tolias, and O. Chum. Fine-tuning CNN image retrieval with no human annotation. *TPAMI*, 2018. 10, 11, 12, 29, 35, 51, 71, 73, 74, 75, 76, 82, 95, 118, 119, 120

[136] F. Radenovic, G. Tolias, and O. Chum. Deep shape matching for domain generalization and cross-modal retrieval. *under submission*, 2019. 10, 11, 12, 94

[137] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *CVPRW*, 2014. 16, 49

[138] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki. Visual instance retrieval with deep convolutional networks. *TMTA*, 2016. 8, 13, 15, 16, 35, 49, 50, 52, 69, 71, 73, 76, 82, 95, 118, 119, 120

[139] H. Riemenschneider, M. Donoser, and H. Bischof. Image retrieval by shape-focused sketching of objects. In *CVWW*, 2011. 18

[140] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. ImageNet large scale visual recognition challenge. *IJCV*, 2015. 13, 15, 49, 73, 119

[141] J. M. Saavedra. Sketch based image retrieval using a soft computation of the histogram of edge local orientations (S-HELO). In *ICIP*, 2014. 18, 105

[142] J. M. Saavedra and J. M. Barrios. Sketch based image retrieval using learned keyshapes (LKS). In *BMVC*, 2015. 18, 105

[143] P. Sangkloy, N. Burnell, C. Ham, and J. Hays. The Sketchy database: Learning to retrieve badly drawn bunnies. *ACM Trans. Graphics*, 2016. 18, 19, 24, 25, 27, 94, 101, 105, 106, 107, 108, 109, 110, 111

[144] T. Sattler, M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *ICCV*, 2015. 11

[145] J. L. Schonberger and J.-M. Frahm. Structure-from-Motion revisited. In *CVPR*, 2016. 56

[146] J. L. Schonberger, F. Radenovic, O. Chum, and J.-M. Frahm. From single image query to detailed 3D reconstruction. In *CVPR*, 2015. 10, 11, 12, 51, 56, 96

[147] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 15, 54

[148] O. Seddati, S. Dupont, and S. Mahmoudi. Quadruplet networks for sketch-based image retrieval. In *ICMR*, 2017. 18, 105, 106, 107

[149] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, and F. Moreno-Noguer. Fracking deep convolutional image descriptors. In *arXiv:1412.6537*, 2014. 15, 61

[150] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *arXiv:1409.1556*, 2014. 47, 50, 52, 62, 72, 87, 95, 98, 99, 103

[151] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 7, 13, 14, 37, 38, 39, 44, 46, 71, 117, 118, 120

[152] J. Song, Y.-Z. Song, T. Xiang, T. Hospedales, and X. Ruan. Deep multi-task attribute-driven ranking for fine-grained sketch-based image retrieval. In *BMVC*, 2016. 18

[153] J. Song, Q. Yu, Y.-Z. Song, T. Xiang, and T. Hospedales. Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In *ICCV*, 2017. 18, 24, 101, 104, 106

[154] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 2014. 66

[155] H. Stewenius, S. H. Gunderson, and J. Pilet. Size matters: Exhaustive geometric verification for image retrieval. In *ECCV*, 2012. 14

[156] X. Sun, C. Wang, C. Xu, and L. Zhang. Indexing billions of images for sketch-based retrieval. In *ACM Multimedia*, 2013. 18

[157] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 105

[158] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. 9, 17, 80, 81, 90

[159] R. Tao, E. Gavves, C. G. Snoek, and A. W. Smeulders. Locality in generic instance search from one example. In *CVPR*, 2014. 71

[160] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. YFCC100M: The new data in multimedia research. *Commun. ACM*, 2016. 34

[161] G. Tolias and Y. Avrithis. Speeded-up, relaxed spatial matching. In *ICCV*, 2011. 34

[162] G. Tolias, Y. Avrithis, and H. Jegou. To aggregate or not to aggregate: Selective match kernels for image search. In *ICCV*, 2013. 71

[163] G. Tolias, Y. Avrithis, and H. Jegou. Image search with selective match kernels: aggregation across single and multiple images. *IJCV*, 2015. 13, 14, 35, 72, 118

[164] G. Tolias and O. Chum. Asymmetric feature maps with application to sketch based retrieval. In *CVPR*, 2017. 18, 98

[165] G. Tolias and O. Chum. Efficient contour match kernel. *Image and Vision Computing*, 2018. 18, 98, 105, 106

[166] G. Tolias and H. Jegou. Visual query expansion with or without geometry: refining local descriptors by feature aggregation. *Pattern Recognition*, 2014. 14, 29, 69, 71, 72

[167] G. Tolias, F. Radenovic, and O. Chum. Query with a flower to retrieve the tower: Adversarial attack to conceal the query image. *under submission*, 2019. 10, 11, 12, 80

[168] G. Tolias, R. Sicre, and H. Jegou. Particular object retrieval with integral max-pooling of CNN activations. In *ICLR*, 2016. 8, 15, 16, 17, 29, 49, 50, 52, 54, 55, 65, 69, 71, 73, 82, 95, 99, 118, 119, 120

[169] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In *CVPR*, 2008. 14, 37

[170] T. Tuytelaars and L. J. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *BMVC*, 2000. 41

[171] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *JMLR*, 2008. 67, 100, 108

[172] Y. Verdie, K. Yi, P. Fua, and V. Lepetit. TILDE: a temporally invariant learned detector. In *CVPR*, 2015. 31

[173] R. Volpi, H. Namkoong, O. Sener, J. Duchi, V. Murino, and S. Savarese. Generalizing to unseen domains via adversarial data augmentation. In *NIPS*, 2018. 19

[174] F. Wang, L. Kang, and Y. Li. Sketch-based 3D shape retrieval using convolutional neural networks. In *CVPR*, 2015. 18

[175] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014. 15, 54, 73

[176] S. Wang, J. Zhang, T. X. Han, and Z. Miao. Sketch-based image retrieval through hypothesis-driven object boundary selection with HLR descriptor. *IEEE Trans. Multimedia*, 2015. 18, 105

[177] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *NIPS*, 2009. 14, 37

[178] T. Weyand and B. Leibe. Discovering details and scene structure with hierarchical iconoid shift. In *ICCV*, 2013. 50

[179] M. J. Wilber, C. Fang, H. Jin, A. Hertzmann, J. Collomosse, and S. J. Belongie. BAM! The Behance Artistic Media dataset for recognition beyond photography. In *ICCV*, 2017. 19

[180] P. Xu, Q. Yin, Y. Huang, Y.-Z. Song, Z. Ma, L. Wang, T. Xiang, W. B. Kleijn, and J. Guo. Cross-modal subspace learning for fine-grained sketch-based image retrieval. *Neurocomputing*, 2017. 18, 106

[181] Z. Xu, W. Li, L. Niu, and D. Xu. Exploiting low-rank structure from latent domains for domain generalization. In *ECCV*, 2014. 19

[182] S. K. Yelamarthi, S. K. Reddy, A. Mishra, and A. Mittal. A zero-shot framework for sketch based image retrieval. In *ECCV*, 2018. 94

[183] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In *CVPR*, 2014. 24

[184] Q. Yu, F. Lie, Y.-Z. Song, T. Xian, T. Hospedales, and C. C. Loy. Sketch me that shoe. In *CVPR*, 2016. 18, 24, 97, 101, 104, 105, 106, 107, 111

[185] Q. Yu, Y. Yang, Y.-Z. Song, T. Xiang, and T. M. Hospedales. Sketch-a-Net that beats humans. In *BMVC*, 2015. 94, 104, 111

[186] J. Yue-Hei Ng, F. Yang, and L. S. Davis. Exploiting local features from deep networks for image retrieval. In *CVPR*, 2015. 71

[187] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based R-CNNs for fine-grained category detection. In *ECCV*, 2014. 49

[188] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas. Query specific fusion for image retrieval. In *ECCV*, 2012. 98

[189] L. Zheng, S. Wang, Z. Liu, and Q. Tian. Lp-norm IDF for large scale image search. In *CVPR*, 2013. 71

[190] L. Zheng, S. Wang, J. Wang, and Q. Tian. Accurate image search with multi-scale contextual evidences. *IJCV*, 2016. 77

[191] L. Zheng, Y. Zhao, S. Wang, J. Wang, and Q. Tian. Good practice in CNN feature transfer. In *arXiv:1604.00133*, 2016. 8, 15

[192] Z. Zheng, L. Zheng, Z. Hu, and Y. Yang. Open set adversarial examples. In *arXiv:1809.02681*, 2018. 9, 17, 80

[193] W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian. Spatial coding for large scale partial-duplicate web image search. In *ACM Multimedia*, 2010. 71