Large-Scale Learning of Discriminative Image Representations

D.Phil Thesis

Robotics Research Group
Department of Engineering Science
University of Oxford

Supervisors:
Professor Andrew Zisserman
Doctor Antonio Criminisi

Karen Simonyan
Mansfield College

Trinity Term, 2013

Karen Simonyan
Mansfield College

Doctor of Philosophy
Trinity Term, 2013

Large-Scale Learning of Discriminative Image Representations

Abstract

This thesis addresses the problem of designing discriminative image representations for a variety of computer vision tasks. Our approach is to employ large-scale machine learning to obtain novel representations and improve the existing ones. This allows us to propose descriptors for a variety of applications, such as local feature matching, image retrieval, image classification, and face verification. Our image and region descriptors are discriminative, compact, and achieve state-of-the-art results on challenging benchmarks.

Local region descriptors play an important role in image matching and retrieval applications. We train the descriptors using a convex learning framework, which learns the configuration of spatial pooling regions, as well as a discriminative linear projection onto a lower-dimensional subspace. The convexity of the corresponding optimisation problems is achieved by using convex, sparsity-inducing regularisers: the L^1 norm and the nuclear (trace) norm. We then extend the descriptor learning framework to the setting, where learning is performed from large image collections, for which the ground-truth feature matches are not available. To tackle this problem, we use the latent variables formulation, which allows us to avoid pre-fixing correct and incorrect matches based on heuristics.

Image recognition systems strongly rely on discriminative image representations to achieve high accuracy. We propose several improvements for the Fisher vector and VLAD image descriptors, showing that better image classification performance can be achieved by using appropriate normalisation and local feature transformation. We then turn to the face image domain, where image descriptors, based on hand-crafted facial landmarks, are currently widely employed. Our approach is different: we densely compute local features over face images, and then encode them using the Fisher vector. The latter is then projected onto a learnt low-dimensional subspace, yielding a compact and discriminative face image representation. We also introduce a deep image representation, termed the Fisher network, which can be seen as a hybrid between shallow representations (which it generalises) and deep neural networks. The Fisher network is based on stacking Fisher encodings, which is feasible due to the supervised dimensionality reduction, injected between encodings.

Finally, we address the problem of fast medical image search, where we are interested in designing a system, which can be instantly queried by an arbitrary Region of Interest (ROI). To facilitate that, we present a medical image repository representation, based on the pre-computed non-rigid transformations between selected images (exemplars) and all other images. This allows for a fast retrieval of the query ROI, since only a fixed number of registrations to the exemplars should be computed to establish the ROI correspondences in all repository images.

This thesis is submitted to the Department of Engineeering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Karen Simonyan, Mansfield College

Acknowledgements

I would like to thank my supervisor, Professor Andrew Zisserman, for his guidance, support, and advice. I am also very grateful to my co-supervisor, Dr. Antonio Criminisi, and a long-term collaborator, Dr. Andrea Vedaldi, for the many fruitful discussions we had. I would like to thank Microsoft Research for providing financial support through the PhD Scholarship Programme. I also thank everyone in VGG for making it such a nice environment to work in. Finally, I would like to thank my parents for all their support and understanding.

Contents

Chapter 1

Introduction

1.1 Objective

This thesis addresses the problem of learning discriminative image representations. By that we mean the representation of images or their regions as vectors in the finite-dimensional Euclidean space. Such representations are a corner stone of the vast majority of computer vision frameworks, since the latter rely on a suitable representation of the image data they are dealing with.

Probably the most obvious and simplistic representation of an image or its part consists in vectorising it by stacking image pixel intensities one-by-one into a vector. As will be discussed in more detail below, such a representation has a disadvantage of the high dimensionality and low robustness. Throughout the last few decades, a plethora of more advanced image representations have been proposed, most of them based on the hand-crafted designs. In this work, we seek to obtain superior image representations by employing large-scale machine learning to obtain the representations, which are tailored to the computer vision task in question.

Note on terminology. In this work, we discuss vector representations for both images and image regions. To denote the image representations, we interchangeably

use the terms *global descriptor* and *image descriptor*. To denote the local region representations, we employ the terms *region descriptor*, *local descriptor*, *local feature*, and *feature descriptor*.

1.2 Motivation and Applications

Being able to describe an image or its region(s) using an effective and efficient representation, well-suited for a particular problem, is essential for a variety of tasks. They include, but are not limited to, the following applications, discussed in this thesis:

Wide baseline image matching. Matching a pair of images taken from substantially different viewpoints, known as wide baseline matching, is an important component of 3-D reconstruction systems. It is usually carried out by first detecting salient regions in each of the images, followed by matching them based on the distance in the region descriptor space (e.g. by nearest-neighbour matching). This brings up the importance of having a region descriptor, equipped with a discriminative Euclidean distance, i.e. the distance between the descriptors of regions corresponding to the same part of a scene should be smaller than the distance between descriptors of regions coming from different parts of the scene. We address this problem by learning an image region descriptor based on the formulation, which enforces such discriminative distance constraints.

Large-scale visual search. With image-capturing devices being in abundance, the problem of large-scale image search based on visual, rather than textual, cues has become particularly relevant. One of the typical visual search use cases consists in searching for a particular object, specified by a user, in a large image collection. The object can be, for example, an architectural landmark, or an image of an item in

a store. A conventional approach to visual search, proposed by Sivic and Zisserman [2003], is based on the tf-idf retrieval scheme, adopted from text retrieval. It relies on the representation of images using visual words, which are obtained by quantising image region descriptors. In this case, learning better region descriptors will lead to more discriminative visual words representation and boost the retrieval accuracy.

Object category recognition. The object category recognition task (sometimes referred to as the "image classification" task) is defined as follows: given an image, determine the category (class label) of its contents. The set of categories is predefined, and, in general, can include both object types (e.g. "human", "car", "dog") and scene types (e.g. "forest", "sunset"). Like any generic classification task, it can be solved by coupling an image representation of choice with a generic classification model, such as the Support Vector Machine (SVM) or the Nearest Neighbour (NN). The discriminative power of the representation has a determinative effect on the classification accuracy, which motivates us to seek for more advanced and rich image representations.

Object instance recognition. The object instance recognition task is to determine if two given images depict the same object instance, e.g. the same person or the same car. In this thesis, we consider the face verification problem – determine whether two images contain the face of the same person, or not. Face verification has numerous and important applications in surveillance, access control, and search. It is inherently a binary classification problem, since it can be seen as classifying face image pairs into "the same person" and "different people". Similarly to the image classification task, the discriminative and concise representation of face images is a key component of accurate large-scale face verification systems.

1.3 Challenges

Large-scale learning of discriminative image representations poses a number of challenges regarding the desired qualities of the learnt representations, as well as the learning framework itself.

Representation desiderata. We seek to obtain the representations of images and image regions, which are discriminative, robust, and allow for fast processing. We elaborate on these desired qualities below. The descriptors should be discriminative in a sense that they should allow for the discrimination between images of different object categories or instances (in the case of image descriptors) or between different parts of the scene (in the case of local descriptors). At the same time, the representations should be robust with respect to a variety of photometric and geometric deformations, such as the change of lighting conditions or the change of object location within an image. Here, by robustness we mean that the distortion of an input image or region should not lead to a significant change of its representation. A related notion is that of the invariance to transformations; in this case, the representation should not change at all. Finally, with the increase of the amount visual data, processed by modern computer vision systems, it is important that the image representations are fast to process. This can be achieved, for example, by reducing the dimensionality of the descriptors (while preserving their discriminative ability), by utilising fast-to-process quantised representations (e.g. binary codes), or by doing both simultaneously. As can be seen, the aforementioned requirements are somewhat contradictory, and meeting them all simultaneously is a challenging target, which we propose to achieve using machine learning.

Learning framework desiderata. The descriptor learning frameworks, which we seek to develop, should be able to efficiently and effectively exploit large amounts

of training data, including the cases where the full supervision is not available. By efficiency we mean that thousands or millions of training samples should be processed in the matter of hours on the CPUs of a conventional workstation (or a cluster). We also seek the convexity of the learning formulations, which would allow us to obtain the global optimum guarantee for the model optimisation procedure. Finally, the training data might not be fully annotated; in this challenging scenario, we need to develop a learning formulation, which can automatically infer the latent training signal and learn the descriptor model.

1.4 Contributions

In this section, we list the main contributions made in this thesis.

1. Convex formulations for learning local descriptors. Local descriptors can sample the input image region in various ways. In prior art, the sampling (spatial pooling) pattern was usually set up by hand, which is sub-optimal. We propose a convex formulation for an optimal selection of pooling region configurations from a large candidate set. To reduce the dimensionality of the resulting descriptor, as well as further improve its discriminative ability, we perform the dimensionality reduction by a linear projection. The projection is also learnt using a convex formulation, by optimising over the Mahalanobis matrix, regularised by the nuclear norm. The result is a compact discriminative image region descriptor, which achieves state-of-the-art performance on the region matching task, using both real-valued and binarised representations. Our convex descriptor learning formulations are presented in Chapter 3.

2. Local descriptor learning from weak supervision. We extend our descriptor learning framework to the case of extremely weak supervision, where learning

is performed from unannotated image collections. For this scenario, we introduce a self-paced learning formulation, which uses latent variables to model region matching uncertainty. This allows us to learn region descriptors from image collections, such as Oxford5K [Philbin et al., 2007], and achieve state-of-the-art image retrieval performance. The details are given in Chapter 4.

3. Improved feature encodings. We then move to the image descriptors, and start with proposing a number of improvements for VLAD [Jégou et al., 2010] and Fisher vector [Perronnin et al., 2010] local feature encodings. We demonstrate that burstiness-reducing intra-normalisation scheme [Arandjelović and Zisserman, 2013] leads to an improved classification accuracy on a challenging PASCAL VOC 2007 benchmark [Everingham et al., 2010]. As a result, we obtain the state-of-the-art results on this dataset among the classification methods based on the encoding of densely computed SIFT [Lowe, 2004] features. We also propose two ways of improving the VLAD representation for the classification tasks: by using an unsupervised whitening projection and a discriminatively trained projection of local features.

4. Fisher vector representation of face images. The Fisher vector encoding [Perronnin et al., 2010] of densely computed SIFT features has been shown to achieve state-of-the-art performance on several image classification benchmarks [Chatfield et al., 2011, Sánchez et al., 2013]. We show that this generic, off-the-shelf, image representation can also be applied to face image description, leading to the state-of-the-art results on a challenging Labelled Face in the Wild dataset [Huang et al., 2007b]. This is in stark contrast to the majority of face descriptors, which are ad-hoc and rely on sampling around face landmarks (computed by a carefully tuned detector). To make the Fisher vector face representation more discriminative, and also decrease its dimensionality, it is passed through discriminative dimensionality reduction, which we learn using a large-margin metric learning formulation. The

Fisher vector face representation is described in Chapter 6.

5. Deep Fisher network representation. Recently, it has been shown [Krizhevsky et al., 2012] that deep convolutional networks outperform the Fisher vector encoding on large-scale image classification tasks. To assess the performance benefits, brought by deep image representations, we propose to extend the conventional shallow Fisher vector encoding by stacking several layers of Fisher encoders (termed Fisher layers) on top of each other. This is made possible by discriminative dimensionality reduction of Fisher vectors, which prevents the explosion in the number of parameters. The resulting architecture, termed the Fisher network, outperforms the shallow Fisher encoding and closes the gap to the deep convolutional networks, while being more practical to train on the CPU. The Fisher network is discussed in Chapter 7.

6. Visual search framework for medical images. Apart from discriminative learning of image representations, this thesis discusses the problem of scalable medical image retrieval. We are interested in designing a system, which allows a clinician to carry out a structured visual search in large medical repositories, i.e. query by a particular region of a medical image. This is in contrast to conventional medical image search systems, which are designed to retrieve globally (rather than locally) similar image. Here we abandon the off-the-shelf object search framework [Sivic and Zisserman, 2003], based on the visual words, and propose a different retrieval scheme, based on fast medical image registration using transform composition. Our medical image search framework is described in Chapter 8.

1.5 Publications

The region descriptor learning framework, described in Chapters 3 and 4, was presented at ECCV 2012 [Simonyan et al., 2012b] and submitted to publication in PAMI [Simonyan et al., 2013b]. The face verification method in Chapter 6 was published at BMVC 2013 [Simonyan et al., 2013a]. The Fisher network architecture in Chapter 7 was accepted for publication at NIPS 2013 [Simonyan et al., 2013c]. The medical image search framework (Chapter 8) was published at MICCAI 2011 [Simonyan et al., 2011]; its extension to 3-D image retrieval was presented at the MICCAI MCBR-CDS 2012 workshop [Simonyan et al., 2012a].

Chapter 2

Literature Review

In this chapter we review some of the related work on image representations and machine learning. We begin with the review of local image region detection and description methods in Sect. 2.1. In Sect. 2.2 we present an overview of global image representations, computed over the whole image. Finally, in Sect. 2.3 we discuss relevant dimensionality reduction methods.

2.1 Image Region Description

In this section we give an overview of various approaches to image region description. The image description task can be defined as follows. Given an image region, it should be encoded into a vector representation, which simplifies its further processing. The notion of processing is application-dependent, but in general the following requirements are imposed on region descriptors:

- **Robustness to region transformations.** The descriptor should not change much in the case of small perturbations in region localisation, or in the case of intensity changes, such as bias and gain (additive and multiplicative intensity transform).

- **Compactness and processing speed.** The region representation should have a low memory footprint to allow for a large number of descriptors to be stored and processed. This can be achieved by reducing the dimensionality of the descriptor, by descriptor compression, or by constraining the descriptor to be a binary, rather than real-valued, vector (which requires 1 bit to store each dimension).

On the input, the region descriptor receives a region, which is localised using a method appropriate for a particular application. For the sake of completeness, in Sect. 2.1.1 we briefly discuss some of the most popular region localisation techniques. Then, we review two families of region description methods, based on spatial pooling (Sect. 2.1.2) and relative comparisons (Sect. 2.1.3). Finally, in Sect. 2.1.4 we discuss descriptor compression methods, some of them are also applicable to the global image representations.

2.1.1 Image Region Localisation

Image region localisation methods can be divided into two groups depending on the spatial sparsity of the regions they generate.

Sparse region detection methods produce a limited set of distinctive regions, usually called feature regions. These regions are supposed to be repeatable, i.e. reliably appear on particular object parts in different images of the same scene. The fact that the detected regions are repeatable and limited in number means that the methods of this kind are particularly suitable for wide-baseline image matching [Pritchett and Zisserman, 1998] and retrieval [Sivic and Zisserman, 2003, Philbin et al., 2007].

A conventional approach to feature region detection is based on defining a saliency measure, and searching for its local maxima on the image plane (which produces the feature region centre) or the image scale-space [Lindeberg, 1998] (which

produces both the feature region centre and scale). The saliency measure can be defined in various ways. The classical (and still widely used) approaches include the determinant of Hessian [Beaudet, 1978], the Harris operator [Harris and Stephens, 1988], and the absolute value of the Laplacian operator [Lindeberg, 1998]. The Harris detector fires on corner-like structures, while the Laplacian and Hessian saliency measures are sensitive to blobs. The regions corresponding to the scale-space saliency maxima are inherently circular, and are invariant to the similarity geometric transformation.

In the wide baseline matching scenario, the invariance to a wider class of transformations may be required. The affine transformation invariance can be achieved through the affine normalisation procedure of Baumberg [2000], which was utilised by Schaffalitzky and Zisserman [2002], as well as Mikolajczyk and Schmid [2002], to derive Harris-Affine and Hessian-Affine feature methods, which detect affine-invariant elliptical image regions. Another notable approach is that of Matas et al. [2002], who defined feature regions as Maximally Stable Extremal Regions (MSER), i.e. connected components of a thresholded image, which are maximally stable with respect to the threshold change. The resulting regions are invariant to affine intensity changes and the projective geometric transformation. A thorough evaluation of various affine-invariant feature detectors can be found in [Mikolajczyk et al., 2005].

There also have been a number of methods aimed at increasing the speed of feature detection. One way of doing it is based on the saliency function approximation. For instance, Lowe [2004] proposed the Difference of Gaussians (DoG) detector, which is a fast approximation of the Laplacian detector [Lindeberg, 1998]. Similarly, Bay et al. [2006] approximated the Hessian detector [Beaudet, 1978] using fast box filters and integral image techniques. Another way of speeding-up feature detection consists in learning a decision model, which approximates the output of the original detector, and is faster to compute [Sochman and Matas, 2009, Rosten

et al., 2010].

Dense region sampling [Leung and Malik, 2001] is different from sparse feature region detection, as it consists in the dense sampling of region location and size. Unlike sparse feature regions, dense regions do not exhibit transformation invariance properties, but are well-suited for image recognition tasks [Nowak et al., 2006], as they cover the whole image plane.

In the case of both dense and sparse region sampling, we can assume that the output of a detector, passed to the descriptor, is a square image intensity patch. Indeed, dense sampling produces square image regions by design. As far as sparse feature regions are concerned, it is beneficial to capture a certain amount of context around a detected feature, as noted in [Matas et al., 2002, Mikolajczyk et al., 2005]. Therefore, each detected feature region is first isotropically enlarged by a constant scaling factor to obtain the descriptor computation region (the measurement region). The latter is then transformed to a square patch using the affine rectification procedure [Mikolajczyk et al., 2005], and can be optionally rotated with respect to the dominant orientation to ensure in-plane rotation invariance. In the sequel, we use the terms "descriptor measurement region" and "descriptor patch" interchangeably.

2.1.2 Pooling-Based Descriptors

Given an image patch, its representation can be obtained in various ways. In the early works on image matching [Zhang et al., 1995, Beardsley et al., 1996, Pritchett and Zisserman, 1998], the feature regions were compared by computing the normalised cross-correlation between the vectors formed of patch pixel intensities. It is easy to see that this is equivalent to computing the Euclidean inner product (or distance) between the whitened intensity vectors. Here, by whitening we mean the element-wise subtraction of the vector mean and division by the variance. Such a representation is invariant with respect to the affine intensity transformation, but is

not robust to region localisation errors and occlusion. For instance, if the detected regions are misaligned, and one of descriptor patches is shifted by 1 pixel compared to another one, their patch vectorisations will be different, making their matching difficult.

The invariance of a descriptor to shift and other perturbations can be achieved by pooling (aggregating) the intensity signal (or its transformation) over spatially localised sub-regions – *descriptor pooling regions*, or receptive fields. Such a design choice is also motivated by the structure of the visual cortex in the mammals brain, discovered by Hubel and Wiesel in the early 1960s [Hubel and Wiesel, 1962]. They identified two basic types of cells in the primary visual cortex (V1): simple and complex. The simple cells respond to specific edge-like stimulus patterns within their receptive field. Complex cells have larger receptive fields and are locally invariant to the exact position of the stimulus inside the receptive field. In other words, simple cells can be seen as (oriented) edge detectors, the output of which is further pooled by the complex cells, resulting in the shift invariance. A number of visual recognition architectures based on interleaving simple and complex cells have been proposed, e.g. Neocognitron [Fukushima, 1980]. Convolutional Neural Networks [LeCun et al., 1998], HMAX [Serre et al., 2007]. Since most of them were originally designed for the whole image representation, they will be discussed in Sect. 2.2.

As far as feature region description is concerned, one of the most widely used pooling-based methods is the Scale-Invariant Feature Transform (SIFT) introduced by Lowe [1999, 2004]. The descriptor is based on the histograms of intensity gradient orientations, computed over 16 square pooling regions, forming a 4×4 grid. Within each such region, a gradient orientation histogram is computed using 8 orientation bins, thus the resulting length of SIFT is $4 \times 4 \times 8 = 128$. The histograms are gathered in a robust way: the contribution of a gradient sample is weighted by its magnitude and the Gaussian window centred at the feature point. Moreover, a gradient sample

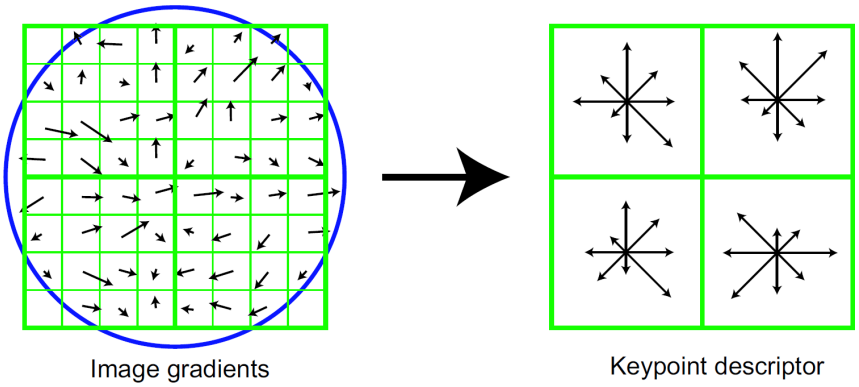


Figure 2.1: **Overview of SIFT computation.** The descriptor is computed by the spatial pooling of oriented gradient features. A 2×2 pooling grid is shown in the figure, but 4×4 is used in practice. The figure was taken from [Lowe, 2004].

contributes not only to the pooling region it belongs to, but to the neighbouring regions as well, which helps to alleviate the boundary effects. Finally, the descriptor is L^2 normalised to make it invariant to the intensity gain. Additional robustness to abrupt intensity changes is achieved by thresholding the normalised descriptor at a fixed threshold and re-normalisation. From the biological vision perspective, the SIFT histogram computation can also be seen as computing 8 oriented gradient feature channels (simple cells), followed by sum-pooling (integration), carried out by the complex cells. The descriptor computation procedure is illustrated in Fig. 2.1.

SIFT has demonstrated a good performance in various computer vision tasks and gave rise to a whole family of methods based on the similar idea of high-pass filtering followed by spatial pooling. For instance, Mikolajczyk and Schmid [2005] proposed Gradient Location-Orientation Histogram (GLOH) descriptor. It is computed over log-polar grid, then the descriptor dimensionality is reduced with principal component analysis. Speeded-Up Robust Features (SURF), proposed by Bay et al. [2006], are built on the distribution of Haar filter responses instead of the gradient orientations. Coupled with the use of integral images, this allows for lower computational complexity compared to SIFT, while maintaining a comparable performance level. Tola et al. [2008] introduced the DAISY descriptor, optimised for the dense compu-

tation at every image pixel (without prior feature detection). To this end, a special configuration of circle-shaped histogram pooling regions is employed. Brown et al. [2011] generalised this approach to a more generic pipeline, defined by the selection of high-pass filters, pooling region configurations, normalisation and quantization techniques. The parameters of the pipeline were found by optimising a non-convex cost function on the ground-truth feature matching set using the method of Powell [1964], which is prone to local minima. In [Boix et al., 2013], gradient encoding using sparse quantisation was used to derive features, pooled using conventional SIFT or DAISY pooling regions.

Certain pooling-based descriptors do not take into account the gradient orientation explicitly, but do it implicitly by sampling the presence of edges at different locations of the input patch. Belongie and Malik [2002] proposed a Shape Context descriptor, which is a histogram of edge point locations computed on a log-polar grid. The Geometric Blur descriptor of Berg et al. [2005] is based on sampling the edge signal, blurred by a spatially varying kernel. The use of the blur makes the descriptor robust to deformations, following the assumption that the closer a pixel is to a feature point, the more important it is in the feature point description.

2.1.3 Comparison-Based Descriptors

The local descriptors, reviewed in the previous section, directly encode the pooled feature channels. A different approach to image region description is to encode the results of the comparison tests, carried out on the descriptor patch.

Lepetit and Fua [2006] introduced a keypoint (feature region) recognition approach to feature description and matching, casting these tasks into a multi-class classification framework. The key idea is that features lying on the same part of scene in different images form a separate class, which defines a set of classes for a given scene. Given a new image of the same scene, its feature regions can be

described by classifying them into one of those classes. The authors employed a random forest [Breiman, 2001] classification framework, using a comparison of pixel intensities as a tree node test. Due to the simplicity of the test, the computational complexity of the keypoint recognition scheme is lower than that of SIFT. It was further decreased in [Ozuysal et al., 2007], where the random forest was replaced with the random ferns classifier. It should be noted that such an approach is suitable only for feature description in images containing the same scene as the training one.

The approach was generalised to images of unseen scenes by Calonder et al. [2008]. They proposed to train the random forest classifier on a hold-out image set and then use the vector of predicted class posteriors as the region descriptor in an image of a new, previously unseen, scene. The descriptor, termed "keypoint signature" is intrinsically sparse, so it can be compressed, as proposed in [Calonder et al., 2009]. The disadvantage of using the classifier output for description is that the optimised classification objective is not relevant to the descriptor distance computation. This has been addressed by Trzcinski et al. [2012, 2013], where they optimised the patch tests in a boosting framework with respect to the descriptor distance constraints. In [Trzcinski et al., 2012], it was also proposed to perform dimensionality reduction using the projections corresponding to the largest eigenvalues of the learnt Mahalanobis matrix. Such an approach is ad-hoc, since dimensionality reduction is not taken into account in the learning objective.

Instead of optimising the parameters of patch tests using machine learning, in a number of works it was proposed to use hand-crafted (BRISK [Leutenegger et al., 2011], ORB [Rublee et al., 2011], FREAK [Alahi et al., 2012]) or even randomly selected (BRIEF [Calonder et al., 2010]) tests. The resulting descriptor is binary, as it is composed of the binary test outcomes.

2.1.4 Descriptor Compression

Binarisation. Binary descriptors have recently attracted much attention due to the low memory footprint and very fast matching times. The low footprint is explained by the fact that a binary descriptor needs just 1 bit to encode each dimension, while 32 bits/dimension are required for the real-valued descriptors in the IEEE single precision format. Additionally, the Hamming distance between binary descriptors can be computed very quickly using the XOR and POPCNT (population count) instructions of the modern CPUs.

There are two major approaches to the binary descriptor computation. First, it possible to obtain an inherently binary representation by recording the "true" / "false" results of binary tests [Calonder et al., 2010, Leutenegger et al., 2011, Rublee et al., 2011, Alahi et al., 2012] (Sect. 2.1.3). A different approach is based on the binarisation of real-valued descriptors. For instance, in LDAHash [Strecha et al., 2012], the binary descriptor is computed by LDA-projection of SIFT (Sect. 2.3.2), followed by binary thresholding. It was proposed to compute each component of the threshold vector separately using one-dimensional search. Instead of SIFT, the vectorised image patch was used in [Trzcinski and Lepetit, 2012]. The binarisation algorithm [Jégou et al., 2012a], used in this work (Sect. 3.6), also performs a linear transformation followed by thresholding. It is thus related to Locality Sensitive Hashing (LSH) with random projections [Charikar, 2002] and Iterative Quantisation (ITQ) [Gong and Lazebnik, 2011]. It differs in that the binary code length is higher than the original descriptor dimensionality, and the projection matrix forms a Parseval tight frame [Kovacevic and Chebira, 2008].

Product Quantisation (PQ). Another popular compression method, which is efficient for both local and global descriptors, is Product Quantisation (PQ), proposed by Jégou et al. [2010]. Similarly to Vector Quantisation (VQ) [Sivic and

Zisserman, 2003], its aim is to represent a vector with an index of the corresponding codeword in a codebook. To decrease the loss incurred by quantisation, PQ splits the original vector into non-overlapping sub-vectors, and trains a separate vocabulary for each of them (e.g. using k-means clustering). As a result, the total number of codewords is large, as it equals the product of the individual codebook sizes. For example, a 128-D SIFT vector, compressed with PQ using 8-D sub-vectors and 256 words in each codebook, can be stored in just 16 bytes (1 byte per each sub-vector, and 1 bit per dimension – as in binary descriptors). At the same time, the total number of different vectors, which can be encoded by such a representation, is large: 256^{16}, which would be unachievable if the descriptor was vector-quantised as a whole. The computation of the distance between two PQ-compressed vectors can be speeded-up using lookup tables.

2.2 Global Image Descriptors

In this section we review image description methods, which aim at representing the whole image as a vector. As noted in Sect. 1.2, such representations are widely employed in various computer vision tasks, such as: object instance recognition, object category recognition, image retrieval, etc. Similarly to local region descriptors, image descriptors are expected to possess the following qualities: robustness to object location, scale, pose perturbation, occlusion, as well as intensity changes (e.g. caused by different lighting conditions). Taking this into account, a state-of-the-art approach to image description is to compute local region descriptors over the image, and use them to derive a global image representation. It should be noted that in some of the early works on image description [Turk and Pentland, 1991, Belhumeur et al., 1997, Cootes et al., 1998], an image was represented using its vectorised intensity. Such a representation is not robust with respect to the change of object

location in the image, and other, more complex, deformations. In this review we concentrate on more modern and robust representations, based on local features.

Image representations, based on local region descriptors, essentially model an image as an ordered or unordered set of local regions. This allows to achieve a certain level of robustness against changes in the object pose, as well as to exploit the robustness against local deformations, provided by the local descriptors. Below we discuss two families of images descriptors: those, which are based on local descriptor encodings, and those which use the "raw" (i.e. non-encoded) local descriptors.

An alternative subdivision of global descriptor methods is based on the underlying local region sampling pattern. Certain global descriptors [Fergus et al., 2005, Everingham et al., 2006, Chen et al., 2013] rely on local descriptors of sparse salient feature regions, which can be obtained using methods reviewed in Sect. 2.1.1 or using domain-specific detectors (e.g. face landmark detectors). Another possible strategy is to compute local descriptors densely, sampling local region location and size over a grid. This produces a large number of regions, covering the whole image, and saves from the need to run a potentially unreliable and time-consuming salient region detector.

2.2.1 Using Raw Local Descriptors

A straightforward way of utilising region descriptors in an image representation is to combine them together by stacking. This approach is viable if the image category is known, so that category-specific salient regions can be reliably detected in each image. For instance, stacking is the underlying idea of many face image descriptors [Everingham et al., 2006, Guillaumin et al., 2009, Chen et al., 2013]. Leveraging on the image domain knowledge, these methods localise face-specific regions (e.g. corners of eyes and mouth), compute local region descriptors around them, and stack the descriptors to obtain the face representation. A more detailed

overview of the face description methods will be given in Sect. 6.1.

Image descriptors based on local descriptor stacking are useful in the controlled scenarios. They are not applicable, however, in the general case, where repeatable salient regions can not be obtained. Additionally, using stacked representations of densely compute features would lead to enormous image descriptor dimensionality, and would not be robust to object translation. One way of tackling these problems is based on encoding and spatial pooling of local features, as will be discussed in Sect. 2.2.2. An alternative is to keep the "raw" (not encoded) descriptors, computed on a dense grid, and use them to implicitly represent the manifold, populated by the descriptors sampled from images of a particular class. Such an approach was employed in the Naive Bayes Nearest Neighbour (NBNN) classifier [Boiman et al., 2008], which infers the image class based on the sum of distances between each of the local descriptors and a set of descriptors sampled from the training set images. A kernelised version of the method, suitable for discriminative learning using SVM, was proposed in [Tuytelaars et al., 2011]. In the case of NBNN-based methods, an image representation is essentially an *unordered* set of local descriptors, so it is invariant to the change of object location within an image. This is different from keeping an ordered set of descriptors, as done by stacking methods above. However, the necessity to store a large number of raw descriptors, sampled from the training images, makes it challenging to apply the method at large scale.

2.2.2 Local Descriptor Encodings

As noted above, keeping a large number of local descriptors is not scalable due to the prohibitively high dimensionality of the resulting representation, which grows linearly with local descriptor number and dimensionality. In this section, we review a large family of methods, which are built on local feature *encodings* – non-linear transformations, which make the descriptors amenable to aggregation over all local

image regions:

$$\Phi = \text{pool}\left(\{\phi(\mathbf{x}_p)\}_{p=1}^{N}\right), \tag{2.1}$$

where $\phi(\mathbf{x}_p)$ is the encoding of a local descriptor $\mathbf{x}_p$, N is the number of local descriptors, and pool is the pooling (aggregation) function. A typical choice of the pooling function is average (sum-pooling): $\Phi = \frac{1}{N}\sum_{p=1}^{N}\phi(\mathbf{x}_p)$ or element-wise max (max-pooling): $\Phi = \max_{p=1}^{N}\phi(\mathbf{x}_p)$. In these cases, Φ has the same dimensionality as ϕ, which does not depend on the number of features N, unlike stacking (Sect. 2.2.1). This means that an arbitrarily large number of features can be represented by a constant-size image descriptor Φ. From (2.1), it can also be seen that the non-linear encoding function ϕ is required to prevent the elements of $\mathbf{x}$ from cancelling out each other during the pooling operation.

Apart from the pooling function (discussed above), there are several choices to make when constructing an image representation of the form (2.1). First is the type of the local descriptor $\mathbf{x}_p$ and its sampling strategy. In recognition tasks, a popular choice is a densely computed SIFT descriptor (dense SIFT), which achieves a very competitive performance, when encoded using state-of-the-art encoding techniques [Chatfield et al., 2011]. As was shown in [Nowak et al., 2006], a dense sampling strategy is better suited for recognition than the sparse feature detection. In the case of wide-baseline image search, however, the SIFT descriptor is typically computed on affine-invariant feature regions [Sivic and Zisserman, 2003, Philbin et al., 2007]. The second design choice is the local descriptor encoding function ϕ. Third, the image descriptor Φ can be post-processed to improve its performance. Finally, it should be noted that the additive representation (2.1) is invariant to the location of descriptors $\mathbf{x}$ on the image plane. While it can be seen as a virtue, such invariance can decrease the discriminative power of the image representation. Therefore, several approaches have been proposed to incorporate spatial information

into the image descriptor Φ. In the sequel, we provide a brief overview of state-of-the-art options for feature encoding, post-processing, and incorporating the spatial information.

Bag of visual Words (BoW) encoding, also known as the "bag of features" encoding, is an approach adopted from text retrieval, and applied to image search by Sivic and Zisserman [2003] and category recognition by Csurka et al. [2004]. It consists in vector-quantisation of a local descriptor $\mathbf{x}$ into visual words $\mathbf{v}_k$, forming a visual codebook (vocabulary) $V = \{\mathbf{v}_k\}_{k=1}^{K}$. The descriptor can then be encoded using a sparse K-dimensional vector with 1 in the position, corresponding to the nearest (in the Euclidean space) visual word, and all other elements set to 0. BoW is usually used with sum-pooling, and it is easy to see that in this case the global descriptor Φ is essentially a histogram of visual word occurrences in the image. The visual codebook is learned on a training set and effectively represents the variability of local descriptors in training images. A conventional way of codebook learning for the BoW encoding is k-means clustering.

The main disadvantage of BoW representation is the quantisation loss, caused by representing a feature using a single visual word. One way of decreasing the quantisation error (albeit at the cost of higher encoding dimensionality) is to use larger codebooks. For instance, Philbin et al. [2007] proposed to use the approximate k-means method to learn large codebooks containing up to 1M visual words. The quantisation loss can also be alleviated by replacing hard assignment of local descriptors to visual words with the soft assignment. E.g. in [Philbin et al., 2008, van Gemert et al., 2008], the soft assignment was computed using the exponential kernel.

Sparse coding can be seen as a variation of the soft-assignment BoW encoding, which enforces the soft assignment of features to only a limited (but larger than 1)

number of codewords. This can be seen as the sparsity constraint on the encoding ϕ, which, when used in vocabulary learning, will enforce it to contain less redundant visual codewords. Yang et al. [2009] used the following sparse coding [Olshausen and Field, 1997] formulation for learning the vocabulary V:

$$\begin{aligned} &\arg \min_{\{\phi_m\},V} \sum_{m=1}^{M} \| \mathbf{x}_m - V\phi_m \|_2^2 + \lambda \|\phi_m\|_1 \\ &\text{s.t. } \|v_k\|_2 \leq 1 \ \forall k, \end{aligned} \tag{2.2}$$

where M is the number of local descriptors in the vocabulary training set, and λ is a regularisation parameter. At test time, the same optimisation problem is solved, but only with respect to the sparse encodings ϕ_m, as the vocabulary is set to the one learnt on the training set. Given V, the optimisation problem over ϕ is convex, but relatively slow to solve, which is a significant disadvantage in practice. This issue has been addressed in the LLC method of Wang et al. [2010], which is uses a different, locality-enforcing, regularisation penalty instead of the L^1 norm in (2.2), and speeds-up the encoding by considering only several Euclidean nearest neighbours as the bases $\mathbf{v}_k$ for the soft assignment. The vocabulary for sparse coding can also be trained discriminatively, e.g. as proposed by Mairal et al. [2008] and Boureau et al. [2010]. Sparse coding can be used with both sum-pooling and max-pooling, but the latter was found to perform better in practice [Yang et al., 2009, Wang et al., 2010]. Similarly to the BoW encoding, the dimensionality of the sparse coding is equal to the size of the visual vocabulary V.

Vector of Locally Aggregated Descriptors (VLAD) is a representation, also aimed at mitigating the quantisation error, but a using a different technique. It retains the k-means codebook, hard assignment, and sum-pooling of BoW, but encodes the displacement of each encoded feature $\mathbf{x}$ with respect to its hard-assigned

visual word $\mathbf{v}_k$. More formally, the encoding of a d-dimensional feature $\mathbf{x}$ can be written as:

$$\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \ldots, \phi_K(\mathbf{x})]\,, \tag{2.3}$$

$$\phi_k(\mathbf{x}) = \begin{cases} \mathbf{x} - \mathbf{v}_k & \text{if } k = \arg\min_j \| \mathbf{x} - \mathbf{v}_j \|_2 \\ \vec{0} & \text{otherwise} \end{cases}$$

where K is the codebook size. From (2.3) it is clear that the VLAD encoding is the stacking of K d-dimensional vectors ϕ_k, only one of which is non-zero for a given feature $\mathbf{x}$. Thus, VLAD of an individual local feature $\mathbf{x}$ is sparse and Kd-dimensional. In other words, each visual word corresponds to a d-dimensional "slot" in the VLAD vector, and a feature $\mathbf{x}$ is encoded by putting the displacement from its visual word $\mathbf{v}_k$ into the corresponding k-th slot. After VLAD is pooled over all encoded features (see (2.1)), each of these slots stores the first-order statistics of the features assigned to the corresponding visual word.

Fisher Vector (FV) encoding also aggregates a set of vectors into a high-dimensional vector representation. In general, this is done by fitting a parametric generative model, e.g. the Gaussian Mixture Model (GMM), to the features, and then encoding the derivatives of the log-likelihood of the model with respect to its parameters [Jaakkola and Haussler, 1998]. The representation is made amenable to linear classification by multiplying it by the Cholesky decomposition of the Fisher information matrix.

Fisher vector representation has been first applied to visual recognition by Perronnin and Dance [2007], who used a GMM with diagonal covariances to model the distribution of local SIFT descriptors. The use of diagonal covariances allows for the closed form computation of the Fisher matrix decomposition, which takes the form

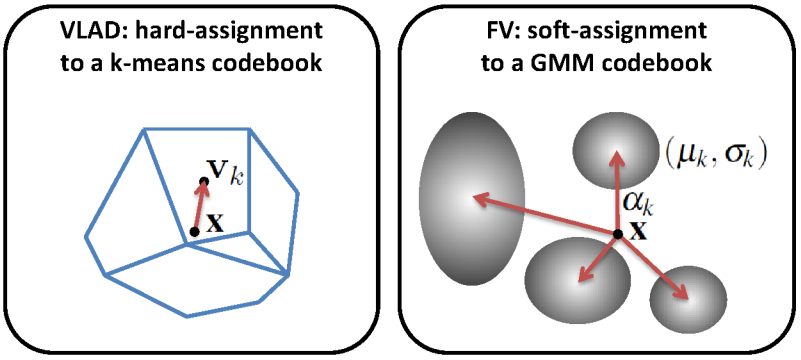


Figure 2.2: **Left:** hard assignment of features to clusters, used in VLAD. **Right:** soft assignment, used in FV.

of whitening. In [Perronnin and Dance, 2007], only the derivatives with respect to the Gaussian mean and variances were considered, which leads to the representation, capturing the average first and second order statistics between the encoded feature $\mathbf{x}$ and each of the GMM centres $\{\mu_k\}_k$:

$$\phi_k^{(1)}(\mathbf{x}) = \frac{1}{\sqrt{\pi_k}}\alpha_k(\mathbf{x})\left(\frac{\mathbf{x}-\mu_k}{\sigma_k}\right), \quad \phi_k^{(2)}(\mathbf{x}) = \frac{1}{\sqrt{2\pi_k}}\alpha_k(\mathbf{x})\left(\frac{(\mathbf{x}-\mu_k)^2}{\sigma_k^2} - 1\right) \tag{2.4}$$

Here, $\{\pi_k, \mu_k, \sigma_k^2\}_k$ are the mixture weights, means, and diagonal covariances of the GMM, which is computed on the training set using the Expectation-Maximisation (EM) algorithm. The division by the vectors σ_k, σ_k^2 is element-wise in (2.4). The soft assignment $\alpha_k(\mathbf{x})$ of the feature $\mathbf{x}$ to the k-th Gaussian is computed as the responsibility of the GMM component k:

$$\alpha_k(\mathbf{x}) = \frac{\pi_k \mathcal{N}_k(\mathbf{x})}{\sum_j \pi_j \mathcal{N}_j(\mathbf{x})}, \tag{2.5}$$

where $\mathcal{N}_k(\mathbf{x}) \sim \exp(-\frac{1}{2}(\mathbf{x}-\mu_k)^T\sigma_k^{-2}(\mathbf{x}-\mu_k))$ is the likelihood of the k-th Gaussian.

An FV encoding of the feature $\mathbf{x}$ is then obtained by stacking the statistics (2.4): $\phi(\mathbf{x}) = \left[\phi_1^{(1)}(\mathbf{x}), \phi_1^{(2)}(\mathbf{x}), \ldots, \phi_K^{(1)}(\mathbf{x}), \phi_K^{(2)}(\mathbf{x})\right]$. Its dimensionality is $2Kd$, where K is the codebook size (the number of Gaussians in the GMM), and d is the dimensionality of the local descriptor $\mathbf{x}$. Similarly to VLAD, Fisher encoding is able to

produce discriminative, high-dimensional feature encodings using small codebooks. Using the same codebook size, BoW and sparse coding are only K-dimensional and less discriminative, as demonstrated in [Chatfield et al., 2011]. From another point of view, given the desired encoding dimensionality, these methods would require $2d$-times larger codebooks than needed for FV, which would lead to impractical computation times.

When sum-pooled over all features in an image (2.1), the encoding describes how the distribution of features of a particular image differs from the distribution fitted to the features of all training images. It should be noted that to make the (SIFT) features amenable to modelling using a diagonal-covariance GMM, they should be first decorrelated, e.g. by Principal Component Analysis (PCA).

It can be shown that the VLAD encoding is a special, non-probabilistic, case of the Fisher vector encoding [Jégou et al., 2012b] (see Fig. 2.2 for illustration). A related representation, termed Super Vector (SV) encoding [Zhou et al., 2010], combines first-order codeword assignment statistics (as in VLAD), the BoW representation, and the soft assignment.

Encoding post-processing. The image descriptor (2.1) can be post-processed (e.g. normalised) to improve its invariance properties and make it more suitable for classification using linear SVM models. In the case of BoW encoding, which is essentially an L^1-normalised histogram, significant improvements can be achieved by passing it through the explicit map [Vedaldi and Zisserman, 2010] of a kernel, suitable for histogram comparison, such as chi-squared, intersection, or Hellinger. In particular, the Hellinger map, which takes the simple form of element-wise (signed) square-rooting (SSR), followed by L^2 normalisation, has been found to be beneficial for a number of image representations, including both global [Guillaumin et al., 2009, Perronnin et al., 2010] and local [Arandjelović and Zisserman, 2012] descriptors.

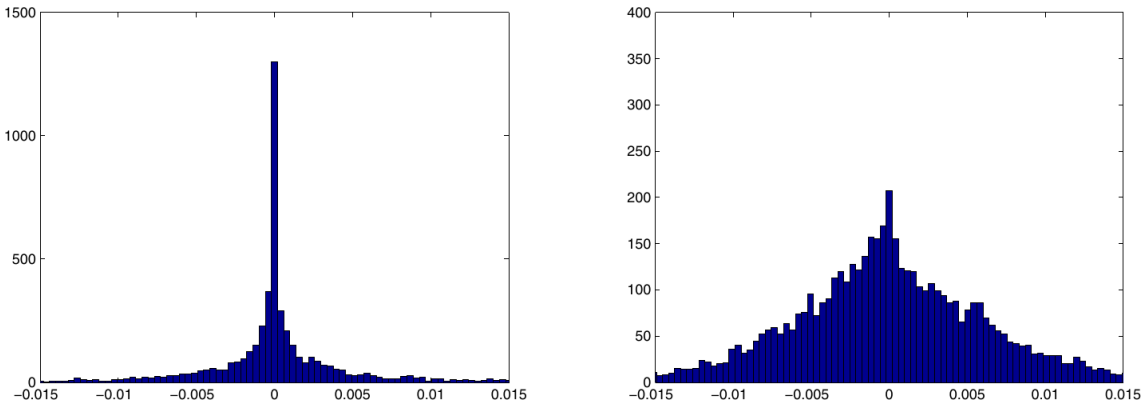


Figure 2.3: **Signed square-rooting reduces the feature burstiness effect.** The histograms show the distribution of the values in the first dimension of the Fisher vector before (left) and after square-rooting (right). The figure was taken from [Perronnin et al., 2010].

For instance, the Fisher vector encoding, coupled with SSR of the form $\text{sgn}(z)\sqrt{|z|}$, significantly outperforms the unnormalised FV encoding, and was termed the "improved Fisher encoding" by Perronnin et al. [2010]. The improvement, brought by the square-rooting transformation, can be explained by the fact that it reduces the effect of the frequently occurring bursty features [Jégou et al., 2009]. As can be seen from Fig. 2.3, it is achieved by decreasing the large components of the encoding and increasing the small ones.

Incorporating the spatial information. The feature encodings, described above, do not explicitly take into account the spatial configuration of local descriptors in an image. One, particularly popular, way of incorporating the spatial information into the image descriptor is called Spatial Pyramid Matching (SPM), and was proposed by Lazebnik et al. [2006]. The SPM representation is built by splitting an image into a grid of rectangular regions (cells), and then describing each region using a separate image descriptor. The resulting descriptors are then stacked to obtain the final image representation. Typically, several grids are combined to produce a multi-scale representation, e.g. $4 \times 4, 2 \times 2, 1 \times 3, 1 \times 1$ (the latter corresponds

to the whole image). Thus, SPM can be seen as a meta-algorithm in a sense that it can be used on top of any image descriptor. The advantage of SPM is that it incorporates rough spatial information, while maintaining the invariance with regard to small object translations (a change of feature location within a cell will not affect the descriptor). The disadvantage is that the descriptor dimensionality grows linearly with the number of SPM cells. This limits the number of cells which can be used in the case of large-scale recognition with high-dimensional descriptors (e.g. only 4 SPM cells were used in [Sánchez and Perronnin, 2011] for ImageNet ILSVRC classification [Berg et al., 2010] using FV features).

Another technique, which leads to only a marginal increase in descriptor dimensionality, is based on the probabilistic modelling of the local feature location (apart from its appearance). For instance, Krapac et al. [2011] proposed to train a separate generative model (e.g. GMM) for the location of local features, assigned to each visual word (in the case of BOW) or Gaussian (in the case of FV). After that, the Fisher vector encoding of image features can be computed on the joint likelihood of their appearance and location. A special case of this approach is the method of [Sánchez et al., 2012], which consists in learning a single GMM on the local features, augmented with their spatial coordinates. Namely, each local region descriptor $\mathbf{u}_{xy}$, computed at the image location (x, y), is concatenated with its normalised spatial coordinates: $\left[\mathbf{u}_{xy}; \frac{x}{w} - \frac{1}{2}; \frac{y}{h} - \frac{1}{2}\right]$, where w and h are the width and height of the image. As a result, the GMM, trained on such features, simultaneously encodes both feature appearance and location.

Spatial information can also be encoded by capturing the spatial co-occurrence statistics of visual words [Savarese et al., 2006].

2.2.3 Deep Image Representations

In this section we discuss *deep* image representations, where by "deep" we mean a computation model which involves layered processing, with the output of one layer being the input for the next one. Such a design choice is motivated by the observation that the mammal visual cortex has a layered structure [Hubel and Wiesel, 1962], which has led to a number of architectures designed to emulate the visual recognition process in the human brain. Due to their biological plausibility, neural networks [Rosenblatt, 1958] have often been employed as layers, resulting in the Deep Neural Network (DNN) architecture.

One of the early DNNs is Neocognitron by Fukushima [1980]. It comprises a set of interleaving simple-cell and complex-cell layers, designed to mimic the processes in simple and complex cells of the visual cortex. Namely, simple cell layers carry out feature extraction using filters with local receptive fields (the same filters are applied at each spatial location). They are followed by complex cell layers, which perform spatial pooling and subsampling on the filters' responses to achieve a certain degree of shift invariance (see also Sect. 2.1.2). A related representation is a Convolutional Neural Network (CNN) of LeCun et al. [1989, 1998], which used back-propagation [Rumelhart et al., 1986] for the supervised training of the whole network. The network is called "convolutional", since applying the same set of local filters densely across the spatial plane can be seen as the convolution operation, followed by a non-linear activation function (e.g. hyperbolic tangent). A classical CNN architecture, called LeNet-5 [LeCun et al., 1998], is shown in Fig. 2.4. It was designed for character and digit recognition in 1990s.

CNNs have been shown to achieve a very good performance on the MNIST digit recognition benchmark [LeCun et al., 1998], but until recently their application to complex natural-image recognition tasks was rather limited due to the large computational complexity of training, as well as the need to train on the large

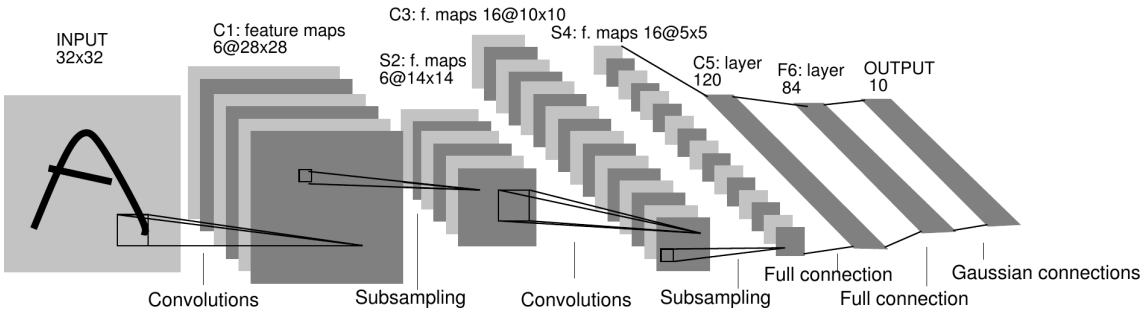


Figure 2.4: **Architecture of the LeNet-5 convolutional neural network.** The figure was taken from [LeCun et al., 1998].

amount of data to avoid over-fitting. The advent of massively-parallel GPUs has recently made it possible to train deep convolutional networks on a large scale with excellent performance [Krizhevsky et al., 2012, Ciresan et al., 2012]. To reduce the over-fitting, the training set was augmented with images generated by jittering – applying random transformations to the original training images. Additionally, the co-adaptation of neurons can be reduced by the "dropout" technique of Hinton et al. [2012], which consists in random "dropping" (switching off) a half of the network on each training sample. In both [Krizhevsky et al., 2012, Ciresan et al., 2012] it was also demonstrated that averaging the outputs of independently trained DNNs can further improve the accuracy, albeit at the cost of training additional models.

Apart from the discriminative supervised DNN training discussed above, other training paradigms exist, which first use unannotated data to initialise the network (which is known as "pre-training"), and then it can be further optimised discriminatively (the "fine-tuning" step). A major use case is the training setting with the large amount of unannotated data, but only a small amount of annotated data, which, if used alone, would lead to severe over-fitting. One example of such a framework is the Deep Belief Network (DBN), proposed by Hinton et al. [2006]. The network is constructed by stacking several layers of Restricted Boltzmann Machines (RBM), which is a generative model. A DBN is trained using a greedy unsupervised layer-by-layer procedure. Instead of RBMs, Bengio et al. [2006] proposed several

types of layers for stacking, each of which can be trained in a greedy, layer-wise manner. One of them is a neural network with a single hidden layer. It is trained with supervision, and, after removing the output layer, the hidden layer is added to the DNN stack. Another, unsupervised, option is a (sparse) auto-encoder. It is a generative model which learns a low-dimensional (or sparse) representation of the input data, such that the input can be optimally reconstructed from it. The resulting network, termed deep auto-encoder, was recently used by Le et al. [2012] to mine high-level visual features from large image sets. Interestingly, they did not employ the weight-sharing principle of CNNs, i.e. different locally-connected filters were applied to different image locations. It should be noted that on the large-scale ImageNet classification task [Deng et al., 2009] (10K categories, 9M images), the sparse auto-encoder [Le et al., 2012] was outperformed by the deep CNN of Krizhevsky et al. [2012].

2.3 Linear Dimensionality Reduction

Linear dimensionality reduction algorithms are aimed at reducing the dimensionality of the vector space by the means of a linear projection:

$$\mathbf{z} = W\,\mathbf{x}, \tag{2.6}$$

where $\mathbf{x} \in \mathbb{R}_n$ and $\mathbf{z} \in \mathbb{R}_m$ are the original and target (dimensionality-reduced) vector representations respectively, and $W \in \mathbb{R}_{m \times n}$ is the linear projection matrix, which is learnt from the training set. This can be done based on the different objectives, e.g. to minimise the reconstruction error, incurred by dimensionality reduction, or to enforce a certain discriminative property of the projected space. In this section we review dimensionality reduction methods, both unsupervised (Sect. 2.3.1) and supervised (Sect. 2.3.2 – 2.3.4). Without loss of generality, here we assume that the

data is zero-centred, which can be achieved by subtracting the mean of the training set features from each of the features.

2.3.1 Unsupervised Dimensionality Reduction

Principal Component Analysis (PCA). Probably the most well-known and widely applied dimensionality reduction method is based on PCA. Proposed by Pearson [1901], PCA can be defined as an orthogonal linear projection $W_{PCA} \in \mathbb{R}_{n\times n}$ onto a lower-dimensional subspace, such that the first coordinate has the highest variance among all possible linear projections, the second coordinate – the second highest, and so on. As a result, PCA reveals the main directions (principal components) along which the data are varying, and the m-th coordinate in the projected space is called the m-th principal component.

Considering that the last principal components tend to have small variance and, as such, might be less important in the data representation, the PCA dimensionality reduction to m dimensions is performed by keeping only the first m PCA components, i.e. by setting the projection matrix W to the first m rows of W_{PCA}. It can also be shown that such W is the minimiser of the reconstruction error: $\min_{W\in\mathbb{R}_{m\times n}} \sum_{\mathbf{x}} \| \mathbf{x} - W^T W \mathbf{x} \|_2^2$, which measures how well the original high-dimensional data can be recovered from the compressed low-dimensional representation. Another interpretation of PCA dimensionality reduction is based on the equivalence between PCA and classical Multi-Dimensional Scaling (MDS), when the latter is applied to the (squared) Euclidean distances [Cox and Cox, 2001]. From this point of view, PCA dimensionality reduction approximates the Euclidean distance in the original space, being the minimiser of the objective:

$$\min_{W\in\mathbb{R}_{m\times n}} \sum_{i<j} \left(\| W\mathbf{x}_i - W\mathbf{x}_j \|_2^2 - \| \mathbf{x}_i - \mathbf{x}_j \|_2^2 \right)^2, \tag{2.7}$$

where $\mathbf{x}_i$ is the i-th vector of the set PCA is applied to. An important property of PCA is that it performs decorrelation, i.e. the correlation between different principal components is zero. As a result, the covariance matrix of the PCA-transformed data is diagonal.

Given a set of vectors $X \in \mathbb{R}_{n \times K}$ (each column i contains an n-dimensional data vector $\mathbf{x}_i$), PCA can be computed from the covariance matrix $C = XX^T \in \mathbb{R}_{n \times n}$ as follows: $W = V^T_{1...m} \in \mathbb{R}_{m \times n}$, where VDV^T is the eigen-decomposition of C, V is the matrix of eigenvectors (one per column, in the decreasing order of eigenvalues), and $V_{1...m}$ are the first m columns of V. As can be seen, computing PCA using this method involves the eigen-decomposition of $\mathbb{R}_{n \times n}$ matrix C, which does not depend on the number of data samples K, but becomes infeasible in the case of high dimensionality n. Computing PCA using the Singular Valued Decomposition (SVD) of the data matrix X generally suffers from the same problem. An alternative method, suitable for a limited number of high-dimensional vectors ($K \ll n$), is based on the eigen-decomposition of the Gram matrix $G = X^T X \in \mathbb{R}_{K \times K}$ which in this case is much smaller than the covariance matrix C. It should be noted that other PCA computation techniques exist, e.g. online PCA [Warmuth and Kuzmin, 2008].

It should be mentioned that PCA can be extended to a non-linear feature space using the "kernel trick" [Scholkopf et al., 1998], but such formulations are outside the scope of this review.

Whitening transformations. As noted above, PCA decorrelates the data, making the covariance matrix diagonal. The variances of the transformed data, however, are not equal: the first principal components have the highest variance, while the last ones – the lowest. In other words, the elements of the PCA-projected vectors are weighted by the square-roots of the corresponding eigenvalues. Such a weighting may not be desirable in certain applications. For instance, it can hamper the

regularisation of discriminative linear models, learnt on top of PCA-projected features. For such models, the feature vectors with balanced components are more desirable, since it allows the learning procedure to determine the importance of the components without being biased towards the prior, eigenvalue-based, weighting.

To equalise the variances of the PCA-projected vector, one can multiply it by the diagonal matrix with the inverse square roots of eigenvalues on its main diagonal. The resulting transformation (PCA followed by re-weighting) is called **PCA-whitening**, and takes the following form:

$$W = \sqrt{D^{-1}_{1\dots m}} V^T_{1\dots m} \in \mathbb{R}_{m\times n}, \tag{2.8}$$

where $D_{1\dots m} \in \mathbb{R}_{m\times m}$ is the diagonal matrix of the top-m eigenvalues, corresponding to the eigenvectors in V As a result, the PCA-whitened data has an identity covariance matrix.

PCA-whitening can be performed with dimensionality reduction ($m < n$) or without it ($m = n$). Another linear transformation, performing whitening without dimensionality reduction, is called **Zero-Phase Component Analysis (ZCA)** [Bell and Sejnowski, 1997]. It corresponds to rotating the PCA-whitened data back to the original space. The ZCA projection is thus computed as:

$$W = V\sqrt{D^{-1}}V^T \in \mathbb{R}_{n\times n}, \tag{2.9}$$

It can be shown that across all rotations of PCA-whitened data, ZCA minimises the squared distortion between the original and the whitened data.

Random projections. One practical shortcoming of PCA is its computational complexity, especially when performed on a large set of high-dimensional vectors. A less computationally demanding method of generating the projection matrix W (2.6)

is based on the random projections [Bingham and Mannila, 2001]. In this case, the elements of W are randomly generated (typically sampled from a zero-mean Gaussian distribution). The random projections method is based on the Johnson-Lindenstrauss lemma, which states that if the points in a high-dimensional space are projected onto a low-dimensional subspace using a random orthogonal projection, the distances between the points are preserved up to a multiplicative factor. In [Bingham and Mannila, 2001] it was shown that the orthogonality constraint can be omitted in practice.

Locality Preserving Projections (LPP). The LPP method [He and Niyogi, 2004, He et al., 2005] aims at finding a linear projection which preserves the local neighbourhood structure of the input data. It can be seen as the linear version of the non-linear Laplacian eigenmaps [Belkin and Niyogi, 2001]. The local neighbourhood is encoded using an adjacency graph, which connects two data points iff they are close in the original high-dimensional space. The feature space proximity can be defined by requiring the L^2 distance between the points to be smaller than a threshold, or by requiring one of the points to be among the k nearest neighbours of the other. Once the graph is constructed, its edges (i, j) are weighted, e.g. using the exponential kernel with the bandwidth σ: $\alpha_{ij} = \exp\left(-\frac{\| \mathbf{x}_i - \mathbf{x}_j \|_2^2}{\sigma^2}\right)$. After that, the projection learning objective is formulated so as to minimise the weighted distance between the adjacent points in the graph after the projection: $W = \arg\min_{W \in \mathbb{R}_{m \times n}} \sum_{ij} \alpha_{ij} \| W\,\mathbf{x}_i - W\,\mathbf{x}_j \|_2^2$. To prevent the degenerate solution $W = 0$, the following normalisation constraint is enforced: $\sum_i \left(\sum_j \alpha_{ij}\right) \| W\,\mathbf{x}_i \|_2^2 = 1$. The optimisation problem is not convex, but an approximate solution can be computed in the closed form as the first m eigenvectors of the generalised eigenproblem involving the graph Laplacian. The approximation scheme will be explained in more detail in the LDA sub-section below.

2.3.2 Supervised Projection Learning Using Eigen-Decomposition

In the previous section we discussed the ways of computing the projection matrix W in the unsupervised setting. A typical objective in such case is to approximate the Euclidean distance in the original high-dimensional space (as done by PCA) or preserve the local neighbourhood structure (as done by LPP). However, in the presence of data annotation, it can be possible to utilise it and learn the projection W in the supervised setting, optimising the application-specific loss (e.g. classification loss). In this section we review the algorithms for the supervised learning of linear projections using Eigen-Decomposition. Large-margin learning formulations will be discussed in Sect. 2.3.3 and 2.3.4.

Linear Discriminant Analysis (LDA). The classical supervised method for learning discriminative projections was proposed by Fisher [1936]. Here we discuss it with the application to dimensionality reduction (rather than just learning a single projection vector). Given a class label annotation for each of the data samples, the linear transformation W is learnt to maximize the ratio of between-class to within-class variance. This means that in the projected space the variance between the samples of different classes should be large, while the variance between the samples of the same class should be small. More formally, given data vectors $\mathbf{x}_i$, annotated into C classes (Ω_c is the set of indices of samples belonging to class c), the objective function for learning LDA projection W is defined as follows:

$$
\begin{aligned}
W &= \arg\max_{W \in \mathbb{R}_{m \times n}} \frac{\mathrm{Tr}\left(W^T S_b W\right)}{\mathrm{Tr}\left(W^T S_c W\right)} \\
S_b &= \frac{1}{C} \sum_{c=1}^{C} \mu_c^T \mu_c, \\
S_c &= \frac{1}{C} \sum_{c=1}^{C} \sum_{i \in \Omega_c} (\mathbf{x}_i - \mu_c)^T (\mathbf{x}_i - \mu_c),
\end{aligned}
\tag{2.10}
$$

where S_b is the covariance of the class means μ_c, and S_c is the sum of within-class covariances.

The "trace ratio" ("trace quotient") optimisation problem (2.10) is non-convex and hard to solve. In practice, it is often approximated by the "ratio trace" problem [Wang et al., 2007]: $\arg\max_{W \in \mathbb{R}_{m \times n}} \operatorname{Tr}\left(\frac{W^T S_b W}{W^T S_c W}\right)$. The latter can be solved in the closed form by assigning the rows of W to the first m eigenvectors w_i of the generalised eigenproblem $S_b w_i = \lambda_i S_c w_i$, $\forall i = 1 \dots m$. Such an approximation technique is also used in other linear embedding methods, e.g. LPP, which was discussed above.

Local Discriminant Embedding (LDE) [Chen et al., 2005] combines the locality enforcing property of LPP with class discrimination property of LDA. In more detail, the algorithm constructs two adjacency graphs. They are similar to the adjacency graph used in LPP, but here one graph has an edge between each pair of samples with the same label, while another graph has an edge connecting each pair of samples with a different label. The affinity weights are computed for each of the graphs similarly to the LPP technique, and the learning objective is formulated so that the (weighted) distance between projected adjacent points of the second graph is maximised, while the distance between the projected adjacent points of the first graph is minimised. As in the case of LPP and LDA, an approximate solution can be found in the closed form by solving a generalised eigenproblem. A similar formulation was used in [Hua et al., 2007], but in that case the two graphs were connecting all samples with the same and different labels respectively, without taking into account their proximity in the feature space.

2.3.3 Supervised Convex Metric Learning

The methods described above formulate the learning objective in terms of the projection matrix W, which leads to non-convex formulations. At the same time, the (squared) Euclidean distance in the W-projected space can be seen as the generalised Mahalanobis distance in the original space:

$$d^2_W(\mathbf{x}_i, \mathbf{x}_j) = \|W\,\mathbf{x}_i - W\,\mathbf{x}_j\,\|^2_2 = (\mathbf{x}_i - \mathbf{x}_j)^T W^T W(\mathbf{x}_i - \mathbf{x}_j) = \tag{2.11}$$

$$d^2_A(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T A(\mathbf{x}_i - \mathbf{x}_j), \tag{2.12}$$

where $A = W^T W$ is the generalised Mahalanobis matrix. Proposed by Mahalanobis [1936], the distance was originally defined under the assumption of data Gaussianity by setting $A = C^{-1}$, where C is the data covariance. It corresponds to the Euclidean distance after whitening (Sect. 2.3.1), meaning that the contribution of each component is normalised based on its correlation with the others. However, in the presence of supervision, it is possible to learn A, thus tailoring the resulting generalised Mahalanobis distance to the task in question. The key property, which allows for the convex formulations of Mahalanobis distance learning, is that d_A is linear in A.

It should be noted that the distance function d_M (2.12), corresponding to an arbitrary matrix $A \in \mathbb{R}_{n\times n}$, does not define a metric. For this to hold, A must be positive-definite. In metric learning algorithms, A is usually constrained to be Positive Semi-Definite (PSD): $A \succeq 0$, which is a convex constraint, and makes d_M a pseudo-metric. This means that the distance d_A between certain non-equal vectors can be zero, which is a desirable property, as the same entity can potentially have several different representations in the original feature space, and the distance between them should be learnt to be zero. Given $A \succeq 0$, it is possible to obtain the corresponding projection W from the eigen-decomposition $A = VDV^T$ in the

following way: $W = \sqrt{D}V^T$. Therefore, optimising over the projection matrix W is equivalent to optimising over a PSD matrix A, which can be exploited in the convex formulations. In general, however, the projection W, corresponding to the learnt A, can be a full-rank $n \times n$ matrix, which does not perform dimensionality reduction. In Chapter 3, we will show how to enforce the dimensionality reduction property through a convex constraint on the Mahalanobis matrix A.

Due to the PSD constraint $A \succeq 0$, the convex optimisation problems, which arise in this setting, belong to the family of Semi-Definite Programming (SDP) problems. Solving them at large scale and/or in an online learning scenario can be intractable, so the optimisation is typically performed using gradient-based methods. In this case, the projection onto the feasible set $A \succeq 0$ (the cone of PSD matrices) can be computed by cropping negative eigenvalues in the eigen-decomposition of A.

Convex formulation for metric learning was first proposed by Xing et al. [2002]. They considered learning a discriminative distance for clustering, based on the supervision in the form of the set of pairs of similar points $\mathcal{P}$ and the set of pairs of dissimilar points $\mathcal{N}$. For instance, $\mathcal{P}$ can be formed of the pairs of samples with the same label, and $\mathcal{N}$ – with different labels. In another interpretation, $\mathcal{P}$ can be seen as the set of "positive" pairs, which should be close in the feature space, and $\mathcal{N}$ contains "negatives" pairs which should be far.

Given the sets of positive and negative pairs, the distance is learnt so that the distance between pairs from $\mathcal{P}$ is small, while the distance between pairs from $\mathcal{N}$ is large:

$$\begin{aligned} &\arg\min_{A} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{P}} d_A^2(\mathbf{x},\mathbf{y}) \\ &\text{s.t.} \quad \sum_{(\mathbf{u},\mathbf{v})\in\mathcal{N}} d_A^2(\mathbf{u},\mathbf{v}) \geq 1, \quad A \succeq 0 \end{aligned} \tag{2.13}$$

The formulation is related to LDA and its variants (Sect. 2.3.2), but here the distance d_A (2.12) is parametrised by A, which makes the optimisation problem (2.13) convex. Due to the convexity, the globally optimal A can be found by the projected gradient descent method, or one of its variants.

Pseudometric Online Learning Algorithm (POLA). In [Shalev-Shwartz et al., 2004], Shalev-Shwartz et al. proposed a large-margin convex formulation based on the classification hinge loss, similar to the one used in the SVM classification. Let y_i be the label of a pair $(\mathbf{x}_i, \mathbf{y}_i)$, so that $y_i = 1$ for positive pairs, and $y_i = -1$ for the negative pairs. Then the following convex formulation can be used to learn the distance model, such that the distance between positive pairs is smaller than the threshold b (by a unit margin), and larger for the negative pairs:

$$\begin{aligned} &\arg\min_{A,b} \sum_i \max\left\{y_i\left(d_A^2(\mathbf{x}_i, \mathbf{y}_i) - b\right) + 1, 0\right\} \\ &\text{s.t.} \quad A \succeq 0 \end{aligned} \tag{2.14}$$

The Mahalanobis matrix A and threshold b can be found by a sub-gradient method, which is well suited for the online learning use case, considered in [Shalev-Shwartz et al., 2004]. At test time, the learnt distance d_A can be used for the binary classification of pairs into positive and negative by comparing it with the threshold b. A similar objective, but based on the smooth logistic loss (Logistic Discriminant Metric Learning, LDML), was proposed by Guillaumin et al. [2009].

Large Margin Nearest Neighbour (LMNN) method, proposed by Weinberger et al. [2006], Weinberger and Saul [2009], is similar to POLA in that it uses a convex large-margin objective. The learning constraints are different though, as LMNN learns a distance for the k-NN classification, where a sample is assigned to the most

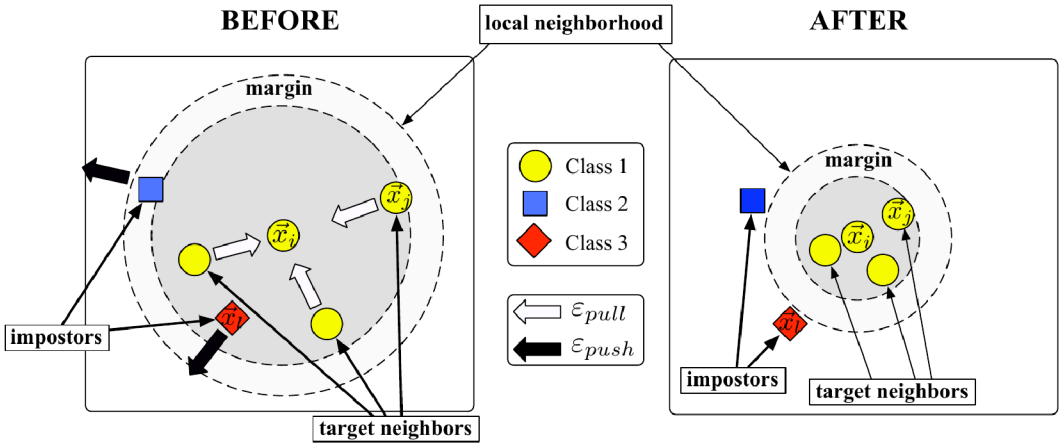


Figure 2.5: **Overview of the large margin nearest neighbour (LMMN) learning objective. Left:** feature vectors before the discriminative projection. **Right:** After the projection, the features of the same class (yellow) are close to each other in terms of the Euclidean distance. The figure was taken from [Weinberger and Saul, 2009].

frequently occurring class among its k Nearest Neighbours (NN). This motivates the use of the distance ranking constraints, which stipulate that for each sample, the distance to the k NNs of the same class ("target neighbours") should be smaller (by a unit margin) than the distance to the samples of other classes ("impostors"). Additionally, the absolute distance between target neighbours is minimised. This is illustrated in Fig. 2.5. The corresponding optimisation problem is as follows (for each sample i, the set of target neighbours is denoted as $\mathcal{P}(i)$, impostors – $\mathcal{N}(i)$):

$$\arg\min_{A} \sum_{i} \left(\sum_{\mathbf{y} \in \mathcal{P}(i)} d_A^2(\mathbf{x}_i, \mathbf{y}) + \sum_{\substack{\mathbf{y} \in \mathcal{P}(i) \\ \mathbf{u} \in \mathcal{N}(i)}} \max\left\{ d_A^2(\mathbf{x}_i, \mathbf{y}) - d_A^2(\mathbf{x}_i, \mathbf{u}) + 1, 0 \right\} \right) \tag{2.15}$$

$$\text{s.t.} \quad A \succeq 0$$

The problem of this approach is that during training, the target neighbours are determined based on the Euclidean distance in the original space, while at test time the learnt distance is used.

2.3.4 Supervised Large-Margin Projection Learning

In Sect. 2.3.3 we discussed the distance learning formulations, defined over the Mahalanobis matrix $A = W^T W$. While they have an advantage of being convex, they generally suffer from two problems in the dimensionality reduction scenario, where the initial dimensionality m is large. First, as noted in Sect. 2.3.3, the projection matrix W, corresponding to the learnt A, is not guaranteed to perform dimensionality reduction. A solution to this problem will be described in Chapter 3. The second problem, however, is more imminent: the matrix $A \in \mathbb{R}_{n \times n}$ has n^2 elements, which is prohibitively large if $n \sim O(10^4)$ or larger, which holds for high-dimensional feature encodings (Sect. 2.2.2). In particular, projecting A onto the set of positive semi-definite matrices involves the eigen-decomposition of A, which is intractable for such n.

Thus, if the dimensionality n is large, one might have to trade convexity for computational tractability, and optimise directly over the projection matrix $W \in \mathbb{R}_{m \times n}$, $m \ll n$, which has mn parameters as opposed to n^2. In Sect. 2.3.2, we reviewed methods based on the eigen-decomposition. This section is dedicated to the large-margin non-convex methods, which learn the projection W.

Large Margin Component Analysis (LMCA) was proposed by Torresani and Lee [2007]. Similarly to LMNN, the method aims at learning a distance, such that for each point the distance to target neighbours is smaller than the distance to impostors by a margin. However, in LMCA, the distance is parametrised by the projection W, so in (2.15), the distance d_A (2.12) is replaced with d_W (2.11), and the PSD constraint $A \succeq 0$ is dropped. This leads to non-convex, but tractable optimisation, which can be carried out using sub-gradient methods. A related formulation of [Guillaumin et al., 2010] uses classification constraints and logistic, rather than hinge, loss.

Bilinear decision function. Supervised dimensionality reduction methods, described above, defined the learning constraints in terms of the distance in the projected space. While such constraints are relevant to the clustering tasks (e.g. k-means) and distance-based classifiers (e.g. k-NN), they can be suboptimal for dot-product-based classification methods (e.g. linear SVM). This was addressed in the WSABIE method by Weston et al. [2010], who employed a bilinear decision function, corresponding to the SVM in the projected space: $f_{W,c}(\mathbf{x}) = v_c^T(W\,\mathbf{x})$, where W is the projection matrix, and v_c is the (apriori unknown) linear SVM model for the class c in the projected space. Similar decision functions were also used by Farhadi et al. [2009] and Gordo et al. [2012]. The formulation of the latter is more relevant to this work, and it takes the following form:

$$\arg\min_{W,\{v_c\}} \sum_i \sum_{\widehat{c}\neq c_i} \max\left\{v_{\widehat{c}}^T(W\,\mathbf{x}_i) - v_c^T(W\,\mathbf{x}_i) + 1, 0\right\} \tag{2.16}$$

Here, c_i is the ground-truth class label of the sample i, for which the decision function should be larger than for any other label $\widehat{c}$. The optimisation is performed over both the projection W and the set of large-margin classifiers $\{v_c\}$. Even though the optimisation problem (2.16) is not convex over both W and $\{v_c\}$ simultaneously, it becomes convex when one of them is fixed.

Chapter 3

Local Descriptor Learning

In this chapter we describe a framework for learning local feature descriptors, based on the convex learning formulations for pooling region selection and dimensionality reduction. As discussed in the previous chapters, local descriptors are an important component of many computer vision algorithms. Here, we are interested in learning descriptors for image matching and retrieval tasks. For instance, in large scale matching, such as the Photo Tourism project [Snavely et al., 2006], and large scale image retrieval [Philbin et al., 2007], the discriminative power of descriptors and their robustness to image distortions are a key factor in the performance. A multitude of local descriptors have been proposed in the literature (an overview is given in Sect. 2.1). Most of these methods are hand-crafted (e.g. SIFT [Lowe, 2004]), though recently machine learning techniques have been applied to learning descriptors matching and retrieval [Philbin et al., 2010, Brown et al., 2011, Trzcinski et al., 2013]. However, although these methods succeed in improving over the performance of SIFT, they use non-convex learning formulations, which are sensitive to the initialisation, and, in general, produce sub-optimal models.

Here we demonstrate that, by leveraging on recent powerful methods for large-scale learning of sparse models, it is possible to learn the descriptors more effectively

than previous techniques. This chapter is structured as follows. First, we describe our descriptor computation pipeline (Sect. 3.1). Then, in Sect. 3.2, we formulate the learning of the configuration the spatial pooling regions of a descriptor as the problem of selecting a few regions among a large set of candidate ones. The significant advantage compared to previous approaches is that the selection can be performed by optimising a sparsity-inducing L^1 regulariser, yielding a convex problem and ultimately a globally-optimal solution. We then proceed with descriptor dimensionality reduction by learning a low-rank metric through penalising the nuclear norm of the Mahalanobis matrix (Sect. 3.3). The nuclear norm is a convex surrogate of the matrix rank, and can be seen as the equivalent of an L^1 regulariser for subspaces. The advantage of our approach is that the low-rank subspace is learnt discriminatively to optimise the matching quality, while still yielding a convex problem and a globally optimal solution. The learning of the pooling regions and of the discriminative projections are formulated as large-scale max-margin learning problems with sparsity enforcing regularisation terms. In order to optimise such objectives efficiently, we employ an effective stochastic learning technique [Xiao, 2010], discussed in Sect. 3.5. Finally, we show that our learnt low-dimensional real-valued descriptors are amenable to binarisation technique based on the Parseval tight frame expansion [Jégou et al., 2012a] to a higher-dimensional space, followed by thresholding (Sect. 3.6). By changing the space dimensionality, we can explore the trade-off between the binary code length and discriminative ability. The result is that we have a principled, flexible, and convex framework for descriptor learning which produces both real-valued and binary descriptors with a low memory footprint and state-of-the-art performance.

As we demonstrate in the experiments of Sect. 3.7, the proposed method outperforms state-of-the-art real-valued descriptors [Philbin et al., 2010, Brown et al., 2011, Trzcinski et al., 2012, Arandjelović and Zisserman, 2012] and binary descrip-

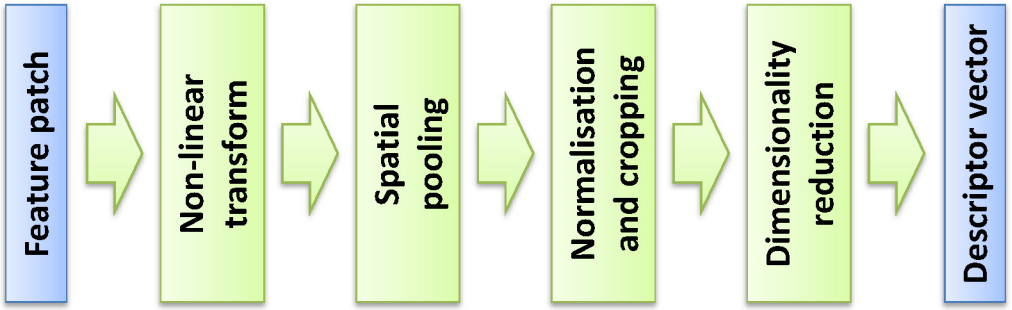


Figure 3.1: **Descriptor computation flowchart.**

tors [Trzcinski et al., 2013, Boix et al., 2013] on a challenging, large-scale dataset of local patches [Brown et al., 2011]. Furthermore, the descriptor learning is efficient and is able to complete within a few hours on a single core for very large scale problems.

3.1 Descriptor Computation Pipeline

We begin with the outline of our descriptor computation pipeline (Fig. 3.1), which is reminiscent of [Brown et al., 2011]. The input is an image patch $\mathbf{x}$ which is assumed to be pre-rectified with respect to affine deformation and dominant orientation. A compact discriminative descriptor $\Psi(\mathbf{x})$ of the patch is computed from the local gradient orientations through the following steps:

Non-linear transform (gradient orientation binning). First, Gaussian smoothing is applied to the patch $\mathbf{x}$. Then the intensity gradient is computed at each pixel and soft-assigned to the two closest orientation bins, weighted by the gradient magnitude as in [Lowe, 2004, Tola et al., 2008, Brown et al., 2011]. This results in p feature channels for the patch, where p is the number of contrast-sensitive orientation bins covering the $[0; 2\pi]$ range (we used $p = 8$ as in SIFT).

Spatial pooling. The oriented gradients computed at the previous step are spatially aggregated via convolution with a set of kernels (e.g. Gaussian or box filters,

normalised to a unit mass) with different location and spatial support (Sect. 3.2); we refer to them as descriptor *Pooling Regions* (PR). Pooling is applied separately to each feature channel, which results in the descriptor vector $\widetilde{\phi}(\mathbf{x})$ with dimensionality pq, where q is the number of PRs.

Normalisation and cropping. The vector of pooling filter responses $\widetilde{\phi}(\mathbf{x})$ is divided by a scalar normalisation factor $T(\mathbf{x})$ and thresholded to obtain the descriptor $\phi(\mathbf{x})$ invariant to intensity changes and robust to outliers.

Discriminative dimensionality reduction. After pooling, the dimensionality of the descriptor $\phi(\mathbf{x})$ is reduced by projection onto a lower-dimensional subspace using the matrix W learnt to improve descriptor matching (Sect. 3.3). The resulting descriptor $\Psi(\mathbf{x}) = W\phi(\mathbf{x})$ can be used in feature matching directly, quantised [Sivic and Zisserman, 2003, Jégou et al., 2010] or binarised (Sect. 3.6).

3.2 Learning Pooling Regions

In this section, we present a framework for learning pooling region configurations. First, a large pool of putative PRs is created, and then sparse learning techniques are used to select an optimal configuration of a few PRs from this pool.

The candidate PRs are generated by sampling a large number of PRs of different size and location within the feature patch. In this work, we mostly consider reflection-symmetric PR configurations, with each PR being an isotropic Gaussian kernel

$$k(u, v; \rho, \alpha, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{1}{2}\frac{(u - \rho\cos\alpha)^2 + (v - \rho\sin\alpha)^2}{\sigma^2}\right] \tag{3.1}$$

where (ρ, α) are the polar coordinate of the centre of the Gaussian relative to the centre of the patch and σ is the Gaussian standard deviation. As shown in Fig. 3.2, the candidate pooling regions ρ, α, σ are obtained by sampling the parameters in

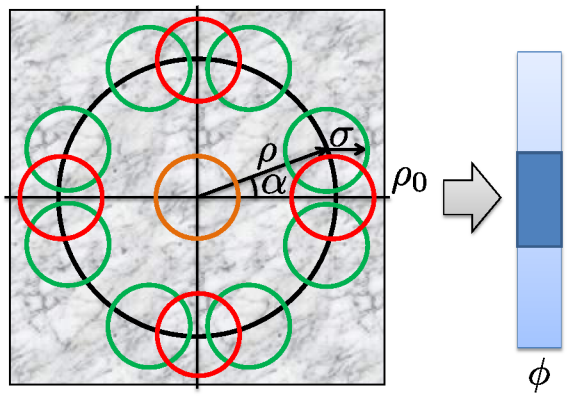


Figure 3.2: **Pooling region candidate rings.** The orange circle shows a ring of a single PR, the red circles – four PRs, the green circles – eight PRs. Each ring corresponds to a sub-vector in the descriptor ϕ (shown on the right).

the ranges: $\rho \in [0; \rho_0]$ (half-pixel step), $\alpha \in [0, 2\pi)$ (step of $\pi/16$), $\sigma \in [0.5; \rho_0]$ (half-pixel step), and then reflecting the resulting PRs (ρ_0 is the patch radius).

Rather than working with individual PRs $(\rho_j, \alpha_j, \sigma_j)$, $j = 1, \ldots, M$, these are grouped by symmetry into *rings* Ω of regions that will be either all selected or discarded. Assuming that the detector choses a natural orientation for the image patch (e.g. the direction parallel or orthogonal to an edge), it is natural to consider rings symmetric with respect to vertical, horizontal, and diagonal flips. Of the 32 regions of equal ρ and σ, this results in two groups of four regions and three groups of eight regions, for a total of five rings (Fig. 3.2). $\rho = 0$ is a special case that has only one pooling region. Since there is a set of five rings for each choice of ρ and σ, the total number of rings is still fairly large, but significantly smaller than the number of individual regions. For example, in Sect. 4.3 the number of candidate rings $\Omega_1, \ldots, \Omega_N$ for 31×31 patches is $N = 4650$.

Selecting pooling regions. This paragraph shows how to select a few PR rings from the N available candidates such that the resulting descriptor discriminates between *positive* (correctly matched) and *negative* (incorrectly matched) feature pairs. More formally, let ϕ be the descriptor defined by PRs pool subset encoded

by the w vector:

$$\phi_{i,j,c}(\mathbf{x}) = \sqrt{w_i}\Phi_{i,j,c}(\mathbf{x}) \tag{3.2}$$

where $\Phi_{i,j,c}(\mathbf{x})$ is the "full" descriptor induced by **all** PRs from the pool $\{\Omega_i\}$, i indexes over PR rings Ω_i, j is a PR index within the ring Ω_i, and c is the feature channel number. The elements of w are non-negative, with non-zero elements acting as weights for the PR rings selected from the pool (and zero weights corresponding to PR rings that are not selected). Due to the symmetry of PR configuration, a single weight w_i is used for all PRs in a ring Ω_i.

We put the following margin-based constraints on the distance between feature pairs in the descriptor space [Weinberger et al., 2006]:

$$d(\mathbf{x},\mathbf{y}) + 1 < d(\mathbf{u},\mathbf{v}) \quad \forall(\mathbf{x},\mathbf{y}) \in \mathcal{P}, (\mathbf{u},\mathbf{v}) \in \mathcal{N} \tag{3.3}$$

where $\mathcal{P}$ and $\mathcal{N}$ are the training sets of positive and negative feature pairs, and $d(\mathbf{x},\mathbf{y})$ is the distance between descriptors of features $\mathbf{x}$ and $\mathbf{y}$. To measure the distance, the squared L^2 distance is used (at this point we do not consider descriptor dimensionality reduction):

$$d(\mathbf{x},\mathbf{y}) = \|\phi(\mathbf{x}) - \phi(\mathbf{y})\|_2^2 = \sum_{i,j,c} \left(\sqrt{w_i}\Phi_{i,j,c}(\mathbf{x}) - \sqrt{w_i}\Phi_{i,j,c}(\mathbf{y})\right)^2 = \tag{3.4}$$

$$\sum_i w_i \sum_{j,c} \left(\Phi_{i,j,c}(\mathbf{x}) - \Phi_{i,j,c}(\mathbf{y})\right)^2 = \sum_i w_i \psi_i(\mathbf{x},\mathbf{y}) = w^T \psi(\mathbf{x},\mathbf{y}),$$

where $\psi(\mathbf{x},\mathbf{y})$ is an N-dimensional vector storing in the i-th element sums of squared differences of descriptor components corresponding to the ring Ω_i:

$$\psi_i(\mathbf{x},\mathbf{y}) = \sum_{j,c} \left(\Phi_{i,j,c}(\mathbf{x}) - \Phi_{i,j,c}(\mathbf{y})\right)^2 \ \forall i = 1 \ldots N \tag{3.5}$$

Now we are set to define the learning objective for PR configuration learning.

Substituting (3.4) into (3.3) and using the soft-margin formulation of the constraints, we derive the following non-smooth *convex* optimisation problem:

$$\arg\min_{w\geq 0} \sum_{\substack{(\mathbf{x},\mathbf{y})\in\mathcal{P}\\(\mathbf{u},\mathbf{v})\in\mathcal{N}}} \mathcal{L}\left(w^T\left(\psi(\mathbf{x},\mathbf{y})-\psi(\mathbf{u},\mathbf{v})\right)\right)+\mu_1\|w\|_1 \tag{3.6}$$

where $\mathcal{L}(z)=\max\{z+1,0\}$ is the hinge loss, and the L^1 norm $\|w\|_1$ is a sparsity-inducing regulariser which encourages the elements of w to be zero, thus performing PR selection. The parameter $\mu_1>0$ sets a trade-off between the empirical ranking loss and sparsity. We note that "sparsity" here refers to the number of PRs, not their location within the image patch, where they are free to overlap. The formulation (3.6) can be seen as an instance of SVM-rank [Joachims, 2002] with L^1 regularisation and non-negativity constraints. It maximises the area under ROC curve corresponding to thresholding the descriptor distance (3.4). The large-scale optimisation of the objective (3.6) is described in Sect. 3.5.

During training, all PRs from the candidate rings are used to compute the vectors $\psi(\mathbf{x},\mathbf{y})$ for training feature pairs $(\mathbf{x},\mathbf{y})$. While storing the full descriptor Φ is not feasible for large training sets due to its high dimensionality (which equals $n_0=p\sum_{i=1}^{N}|\Omega_i|$, i.e. the number of channels times the number of PRs in the pool) the vector ψ is just N-dimensional, and can be computed in advance before learning w.

Descriptor normalisation and cropping. Once a sparse w is learnt, at test time only PRs corresponding to the non-zero elements of w are used to compute the descriptor. This brings up the issue of descriptor normalisation, which should be consistent between training and testing to ensure good generalisation. The conventional normalisation by the norm of the pooled descriptor $\widetilde{\phi}$ would result in different normalisation factors, since the whole PR pool is used during training, but only a (learnt) subset of PRs – in testing. Here we explain how to compute the descriptor

normaliser $T(\mathbf{x})$ which does not depend on PRs. This ensures that in both training and testing the same normalisation is applied, even though different sets of PRs are used.

The un-normalised descriptor $\widetilde{\phi}(\mathbf{x})$ is essentially a spatial convolution of gradient magnitudes distributed across orientation bins. Such a descriptor is invariant to an additive intensity change, but it does vary with intensity scaling. To cancel out this effect, a suitable normalisation factor $T(\mathbf{x})$ can be computed from the patch directly, independently of the PR configuration. Here, we set $T(\mathbf{x})$ to the ζ-quantile of gradient magnitude distribution over the patch. Given $T(\mathbf{x})$, the response of each PR is normalised and cropped to 1 for each PR independently as follows:

$$\phi_i(\mathbf{x}) = \min\left\{\widetilde{\phi}_i(\mathbf{x})/T(\mathbf{x}), 1\right\} \ \forall i. \tag{3.7}$$

We employ the quantile statistic to estimate the threshold value such that only a small ratio of pixels have the gradient magnitude larger than it. These pixels potentially correspond to high-contrast or overexposed image areas, and to limit the effect of such areas on the descriptor distance, the corresponding gradient magnitude is cropped (thresholded). The thresholding quantile ζ was set to 0.8 in all experiments. An alternative way of computing the threshold $T(\mathbf{x})$ is to use the sum of the gradient magnitude mean and variance, as done in [Simonyan et al., 2012b]. In this work, we use the quantile statistic as a more principled way of threshold value computation. As a result of the normalisation and cropping procedure, the descriptor $\phi(\mathbf{x})$ is invariant to affine intensity transformation, and robust to abrupt gradient magnitude changes.

3.3 Learning Dimensionality Reduction

This section proposes a framework for learning discriminative dimensionality reduction using a convex formulation. The aim is to learn a linear projection matrix W such that (i) W projects descriptors onto a lower dimensional space; (ii) positive and negative descriptor pairs are separated by a margin in that space.

The first requirement can be formally written as $W \in \mathbb{R}^{m \times n}, m < n$ where m is the dimensionality of the projected space and n is the descriptor dimensionality before projection. The second requirement can be formalised using a set of constraints similar to (3.3):

$$d(\mathbf{x}, \mathbf{y}) + 1 < d(\mathbf{u}, \mathbf{v}) \quad \forall (\mathbf{x}, \mathbf{y}) \in \mathcal{P}, (\mathbf{u}, \mathbf{v}) \in \mathcal{N} \tag{3.8}$$

As explained in Sect. 2.3.3, parametrising the distance function by the generalised Mahalanobis matrix $A = W^T W$ leads to convex optimisation problems. Therefore, we set

$$d_A(\mathbf{x}, \mathbf{y}) = \theta(\mathbf{x}, \mathbf{y})^T A \theta(\mathbf{x}, \mathbf{y}), \tag{3.9}$$

where $\theta(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) - \phi(\mathbf{y})$, and $A \in \mathbb{R}^{n \times n}, A \succeq 0$ is a positive semi-definite matrix. The constraints (3.8), (3.9) are convex in A, but in general, the learnt A can have a full rank, so the corresponding W does not perform dimensionality reduction Sect. 2.3.3. We begin with explaining why the *low rank* of A is important for dimensionality reduction by showing the equivalence between learning a low-rank A and dimensionality-reducing W. Then, we will explain how to enforce the low rank of A in a convex manner.

If $A \in \mathbb{R}^{n \times n}$ has a low rank, i.e. $\text{rank}(A) = m < n$, then a dimensionality reduction projection $W \in \mathbb{R}_{m \times n}$ can be obtained from the eigen-decomposition

$A = VDV^T$. Due to the low rank, the diagonal matrix of eigenvalues $D \in \mathbb{R}^{n\times n}$ has only m non-zero elements. Let $D_r \in \mathbb{R}^{m\times n}$ be the matrix obtained by removing the zero rows from D. Then W can be constructed as $W = \sqrt{D_r}V^T$. Conversely, if $W \in \mathbb{R}^{m\times n}$ and $\text{rank}(W) = m$, then $\text{rank}(A) = \text{rank}(W^TW) = \text{rank}(W) = m$. Thus, a dimensionality reduction constraint on W can be equivalently transformed into a rank constraint on A. However, the direct optimisation of $\text{rank}(A)$ is not tractable due to its non-convexity. The convex relaxation of the matrix rank is described next.

Nuclear norm regularisation. The nuclear norm $\|A\|_*$ of matrix A (also referred to as the trace norm) is defined as the sum of singular values of A. For positive semi-definite matrices the nuclear norm equals the trace. The nuclear norm performs a similar function to the L^1 norm of a vector – the L^1 norm of a vector is a convex surrogate of its L^0 norm, while the nuclear norm of a matrix is a convex surrogate of its rank [Fazel et al., 2001, Recht et al., 2010].

Using the soft-margin formulation of the constraints (3.8), (3.9) and the nuclear norm in place of rank, we obtain the non-smooth *convex* objective for learning A:

$$\arg\min_{A\succeq 0} \sum_{\substack{(\mathbf{x},\mathbf{y})\in\mathcal{P}\\(\mathbf{u},\mathbf{v})\in\mathcal{N}}} \mathcal{L}\left(\theta(\mathbf{x},\mathbf{y})^T A\theta(\mathbf{x},\mathbf{y}) - \theta(\mathbf{u},\mathbf{v})^T A\theta(\mathbf{u},\mathbf{v})\right) + \mu_*\|A\|_*, \tag{3.10}$$

where the parameter $\mu_* > 0$ trades off the empirical ranking loss versus the dimensionality of the projected space: the larger μ_*, the smaller the dimensionality. We note that this formulation gives no direct control over the projected space dimensionality. Instead, the dimension can be tuned by running the optimisation with different values of μ_*.

Nuclear norm regularisation has been recently applied to a wide range of problems, e.g. max-margin matrix factorisation [Rennie and Srebro, 2005], low-rank ker-

nel learning [Jain et al., 2010], multi-class classification [Harchaoui et al., 2012]. In our case, we use it to learn a dimensionality-reducing linear projection. To optimise the non-smooth (due to the hinge loss) objective (3.10), we use the Regularised Dual Averaging [Xiao, 2010] optimisation method, described in Sect. 3.5.

3.4 Discussion

Our descriptor learning algorithm includes two stages: learning a sparse pooling region configuration (Sect. 3.2) and learning a low-rank projection (Sect. 3.3) for the selected PRs. It is natural to consider whether the two stages can be combined and the sparse pooling configuration and low rank projection learned simultaneously.

In fact, the two stages provide a computationally feasible way of solving one, extremely large-scale, low-rank metric learning problem. Selecting a small set of PR rings and, simultaneously, performing their dimensionality reduction corresponds to projecting the full descriptor $\Phi \in \mathbb{R}^{n_0}$ (3.2) with a rectangular matrix $V \in \mathbb{R}^{m \times n_0}, m \ll n_0$, which has a special structure. Namely, to select only a few PR rings from the pool, V must have a column-wise group sparsity pattern, such that a group of $p\,|\Omega_i|$ columns, corresponding to i-th PR ring, can only be set to zero all together (meaning that the i-th ring is not selected from the candidate pool). Optimisation over the projection matrix V is large-scale (the number of parameters $mn_0 \approx 19M$ for $m = 64$ and $n_0 \approx 298K$) and non-convex (Sect. 3.3). A convex optimisation of the corresponding Mahalanobis matrix $B = V^T V \in \mathbb{R}^{n_0 \times n_0}$ would incur learning $n_0^2 \approx 89{\cdot}10^9$ parameters under a non-trivial group sparsity constraints, which is not feasible.

So what has been lost by the two stage learning? Ideally, we would like our loss function to only involve the final dimensionality reduced descriptor – so that the loss measures how positive and negative descriptor pairs are separated by a

margin in the projected space as in (3.8) of Sect. 3.3. Instead at the first stage (Sect. 3.2) we have to use a proxy loss (3.3) which involves the descriptors *before* dimensionality reduction. In our case the advantage is that we effectively factorise the projection V as $V = WV_{PR}$, where $V_{PR} \in \mathbb{R}^{n \times n_0}$ is a rectangular diagonal matrix, induced by PR-selecting sparse vector $w \in R^N, N = 4650$ (Sect. 3.2), and $W \in \mathbb{R}^{m \times n}$ is further reducing the dimensionality of the selected PRs (Sect. 3.3); and also both projections, W and V_{PR}, are learnt using computationally tractable convex formulations.

3.5 Regularised Stochastic Learning

In sections 3.2 and 3.3 we proposed convex optimisation formulations for learning the descriptor PRs as well as the discriminative dimensionality reduction. However, the corresponding objectives (3.6) and (3.10) yield very large problems as the number of summands is $|\mathcal{P}|\,|\mathcal{N}|$, where typically the number of positive and negative matches is in the order of 10^5 – 10^6 (Sect. 3.7). This makes using conventional interior point methods infeasible.

To handle such very large training sets, we propose to use *Regularised Dual Averaging* (RDA), the recent method by [Xiao, 2010, Nesterov, 2009]. To the best of our knowledge, RDA has not yet been applied in the computer vision field, where, we believe, it could be used in a variety of applications beyond the one presented here. RDA is a stochastic proximal gradient method effective for problems of the form

$$\min_w \frac{1}{T} \sum_{t=1}^{T} f(w, z_t) + R(w) \tag{3.11}$$

where w is the weight vector to be learnt, z_t is the t-th training (sample, label) pair, $f(w, z)$ is a convex loss, and $R(w)$ is a convex regularisation term. Compared to proximal methods for optimisation of smooth losses with non-smooth regularisers

(e.g. FISTA [Beck and Teboulle, 2009]), RDA is more generic and applicable to *non-smooth* losses, such as the hinge loss employed in our framework. As opposed to other stochastic proximal methods (e.g. FOBOS [Duchi and Singer, 2009]), RDA uses more aggressive thresholding, thus producing solutions with higher sparsity. A detailed description of RDA can be found in [Xiao, 2010]; here we provide a brief overview.

At iteration t RDA uses the loss sub-gradient $g_t \in \delta_w f(w, z_t)$ to perform the update:

$$w_{t+1} = \arg\min_w \left(\langle \bar{g}_t, w \rangle + R(w) + \frac{\beta_t}{t} h(w) \right) \tag{3.12}$$

where $\bar{g}_t = \frac{1}{t} \sum_{i=1}^{t} g_i$ is the average sub-gradient, $h(w)$ is a strongly convex function such that $\arg\min_w h(w)$ also minimises $R(w)$, and β_t is a specially chosen non-negative non-decreasing sequence. We point out that $\bar{g}_t$ is computed by averaging sub-gradients across iterations, not samples. If the regularisation $R(w)$ is not strongly convex (as in the case of L^1 and nuclear norms), one can set $h(w) = \frac{1}{2}\|w\|_2^2$, $\beta_t = \gamma\sqrt{t}$, $\gamma > 0$ to obtain the convergence rate of $O(1/\sqrt{t})$.

It is easy to derive the specific form of the RDA update step for the objectives (3.6) and (3.10). For the sparse pooling region weight learning (3.6), we have:

$$w_{t+1} = \max \left\{ -\frac{\sqrt{t}}{\gamma} \left(\bar{g}_t + \mu_1 \mathbb{I} \right), 0 \right\}, \tag{3.13}$$

where $\bar{g}_t$ is the average sub-gradient of the hinge loss, and $\mathbb{I}$ is the vector with 1 in each element. At iteration t, given a positive feature match $(\mathbf{x}_t, \mathbf{y}_t)$ and a negative match $(\mathbf{u}_t, \mathbf{v}_t)$, the hinge loss sub-gradient is computed as follows:

$$g_t = \begin{cases} \psi(\mathbf{x}_t, \mathbf{y}_t) - \psi(\mathbf{u}_t, \mathbf{v}_t), & \text{if } d(\mathbf{x}_t, \mathbf{y}_t) + 1 > d(\mathbf{u}_t, \mathbf{v}_t) \\ 0, & \text{otherwise} \end{cases} \tag{3.14}$$

where $d(\mathbf{x}_t, \mathbf{y}_t)$ is the distance, defined in (3.4). As can be seen, the sub-gradient is zero if the constraint (3.3) is violated.

For a low-rank Mahalanobis matrix learning (3.10), the RDA update is similar:

$$A_{t+1} = \Pi\left(-\frac{\sqrt{t}}{\gamma}\left(\bar{g}_t + \mu_* \mathbb{I}\right)\right), \tag{3.15}$$

where $\mathbb{I}$ is the identity matrix and Π is the projection onto the cone of positive semi-definite matrices, computed by cropping negative eigenvalues in the eigen-decomposition. In this case, the sub-gradient is the difference of outer products if the constraint (3.8) is violated, and 0 otherwise:

$$g_t = \begin{cases} \theta(\mathbf{x}_t, \mathbf{y}_t)\theta(\mathbf{x}_t, \mathbf{y}_t)^T - \theta(\mathbf{u}_t, \mathbf{v}_t)\theta(\mathbf{u}_t, \mathbf{v}_t)^T, \text{ if } d_A(\mathbf{x}_t, \mathbf{y}_t) + 1 > d_A(\mathbf{u}_t, \mathbf{v}_t) \\ 0, \text{ otherwise} \end{cases} \tag{3.16}$$

3.6 Binarisation

In this section we describe how a low-dimensional real-valued descriptor $\Psi \in \mathbb{R}^m$ can be binarised to a code $\beta \in \{0,1\}^q$ with the bit length q *higher* or equal to m. To this end, we adopt the method of [Jégou et al., 2012a], which is based on the descriptor expansion using a Parseval tight frame, followed by thresholding (taking the sign).

In more detail, a *frame* is a set of $q \geq m$ vectors generating the space of descriptors $\Psi \in \mathbb{R}^m$ [Kovacevic and Chebira, 2008]. In the matrix form, a frame can be represented by a matrix $U \in \mathbb{R}^{q \times m}$ composed of the frame vectors as rows. A *Parseval tight frame* has the additional property that $U^\top U = \mathbb{I}$. An expansion with such frames, $U\Psi \in \mathbb{R}^q$, is an overcomplete representation of $\Psi \in \mathbb{R}^m$, which preserves the Euclidean distance. Due to the overcompleteness, binarisation of the expanded

vectors leads to a more accurate approximation of the original vectors Ψ. Assuming that the descriptors Ψ are zero-centred, the binarisation is performed as follows:

$$\beta = \text{sgn}(U\Psi), \tag{3.17}$$

where sgn is the sign function: $\text{sgn}(a) = 1$ iff $a > 0$ and 0 otherwise. Following [Jégou et al., 2012a], we compute the Parseval tight frame U by keeping the first m columns of an orthogonal matrix obtained from a QR-decomposition of a random $q \times q$ matrix.

In spite of the binary code dimensionality q being not smaller than the dimensionality m of the real-valued descriptor, the memory footprint of the binary code is smaller if $q < 32m$ (see Sect. 2.1.4). Changing q allows us to generate the binary descriptors with any desired bitrate $q \geq m$, balancing the matching accuracy vs the memory footprint.

3.7 Experiments

In this section, we rigorously assess the components of the proposed framework (Sect. 3.2, 3.3, 3.6) on the Local Image Patches Dataset [Brown et al., 2011], where feature patches are available together with the ground-truth annotation into matches and non-matches. The descriptor performance in this case is measured based on a fixed operating point on the descriptor matching ROC curve. We demonstrate that our learnt representations achieve state-of-the-art results among real-valued and binary descriptors.

3.7.1 Dataset and Evaluation Protocol

The evaluation is carried out on the Local Image Patches Dataset [Brown et al., 2011]. It consists of three subsets, Yosemite, Notre Dame, and Liberty, each of

which contains more than 450,000 image patches (64×64 pixels) sampled around Difference of Gaussians (DoG) feature points. The patches are rectified with respect to the scale and dominant orientation. Each of the subsets was generated from a scene for which 3D reconstruction was carried out using multiview stereo algorithms. The resulting depth maps were used to generate 500,000 ground-truth feature pairs for each dataset, with equal number of positive (correct) and negative (incorrect) matches.

To evaluate the performance of feature descriptors, we follow the evaluation protocol of [Brown et al., 2011] and generate ROC curves by thresholding the distance between feature pairs in the descriptor space. We report the false positive rate at 95% recall (FPR95) on each of the six combinations of training and test sets, as well as the mean across all combinations. Considering that in [Brown et al., 2011, Boix et al., 2013] only four combinations were used (with training on Yosemite or Notre Dame, but not Liberty), we also report the mean for those, denoted as "mean 1–4". Following [Brown et al., 2011], for training we used 500,000 feature matches of one subset, and tested on 100,000 matches of the others. Note that training and test sets were generated from images of different scenes, so the evaluation protocol assesses the generalisation of the learnt descriptors.

3.7.2 Descriptor Learning Results

We compare our learnt descriptors with the state-of-the-art unsupervised [Arandjelović and Zisserman, 2012] and supervised descriptors [Brown et al., 2011, Trzcinski et al., 2012, 2013, Boix et al., 2013] in three scenarios. First, we evaluate the performance of the learnt pooling regions (*PR*, Sect. 3.2) and compare it with the pooling regions of [Brown et al., 2011]. Second, our complete descriptor pipeline based on projected pooling regions (*PR-proj*, Sect. 3.2–3.3) is compared against other real-valued descriptors [Arandjelović and Zisserman, 2012, Brown et al., 2011, Trzcin-

ski et al., 2012]. Finally, we assess the compression of our descriptors, for which we consider the binarisation method (*PR-proj-bin*, Sect. 3.6), as well as a conventional product quantisation technique [Jégou et al., 2010] (*PR-proj-pq*). We compare the compressed descriptors with state-of-the-art binary descriptors [Trzcinski et al., 2013, Boix et al., 2013], which were shown to outperform unsupervised methods, such as BRIEF [Calonder et al., 2010] and BRISK [Leutenegger et al., 2011] as well as earlier learnt descriptors of [Strecha et al., 2012, Trzcinski and Lepetit, 2012].

In the comparison, apart from the FPR95 performance measure, for each of the descriptors we indicate its memory footprint and type. For real-valued descriptors, we specify their dimensionality as ⟨dim⟩f, e.g. 64f for 64-D descriptors. Assuming that the single-precision float type is used, each real-valued descriptor requires $(32 \times \text{dim})$ bits of storage. For compressed descriptors, their bit length and type are given as ⟨bits⟩⟨type⟩, where ⟨type⟩ is "b" for binary, and "pq" for product-quantised descriptors.

To learn the descriptors, we randomly split the set of 500,000 feature matches into 400,000 training and 100,000 validation. Training is performed on the training set for different values of μ_1, μ_* and γ, which results in a set of models with different dimensionality-accuracy tradeoff. Given the desired dimensionality of the descriptor, we pick the model with the best performance on the validation set among the ones whose dimensionality is not higher than the desired one.

Learning pooling regions. Table 3.1 compares the error rates reported in [Brown et al., 2011] (5-th column) with those of the PR descriptors learnt using our method. The 4-th column corresponds to the descriptors with the dimensionality limited by 384, so that it is not higher than the one used in [Brown et al., 2011]; in the 3rd column, the dimensionality was limited by 640 (a threshold corresponding to ≤ 80 PRs selected). In Fig. 3.3 (top) we plot the error rate of the learnt descriptors as

Table 3.1: **False positive rate (%) (at 95% recall) for learnt pooling regions.** Yos: Yosemite, ND: Notre Dame, Lib: Liberty.

Train set	Test set	PR ≤ 640-D	PR ≤ 384-D	Brown et al.
Yos	ND	**9.49 (544f)**	9.88 (352f)	14.43 (400f)
Yos	Lib	**17.23 (544f)**	17.86 (352f)	20.48 (400f)
ND	Yos	11.11 (576f)	**10.91 (352f)**	15.91 (544f)
ND	Lib	**16.56 (576f)**	17.02 (352f)	21.85 (400f)
Lib	Yos	**11.89 (608f)**	12.99 (384f)	N/A
Lib	ND	**9.88 (608f)**	10.51 (384f)	N/A
mean		12.69	13.20	N/A
mean (1–4)		13.60	13.92	18.17

Table 3.2: **False positive rate (%) (at 95% recall) for real-valued descriptors.** Yos: Yosemite, ND: Notre Dame, Lib: Liberty.

Train set	Test set	PR-proj ≤80-D	PR-proj ≤64-D	PR-proj ≤32-D	Brown et al.	Trzcinski et al.	rootSIFT	rootSIFT-proj ≤80-D
Yos	ND	**6.82 (76f)**	7.11 (58f)	9.99 (32f)	11.98 (29f)	13.73 (64f)	22.06 (128f)	14.60 (77f)
Yos	Lib	**14.58 (76f)**	14.82 (58f)	16.7 (32f)	18.27 (29f)	21.03 (64f)	29.65 (128f)	22.20 (77f)
ND	Yos	**10.08 (73f)**	10.54 (63f)	13.4 (32f)	13.55 (36f)	15.86 (64f)	26.71 (128f)	19.00 (70f)
ND	Lib	**12.42 (73f)**	12.88 (63f)	14.26 (32f)	16.85 (36f)	18.05 (64f)	29.65 (128f)	20.11 (70f)
Lib	Yos	**11.18 (77f)**	11.63 (58f)	14.32 (32f)	N/A	19.63 (64f)	26.71 (128f)	19.96 (76f)
Lib	ND	**7.22 (77f)**	7.52 (58f)	9.07 (32f)	N/A	14.15 (64f)	22.06 (128f)	13.99 (76f)
mean		**10.38**	10.75	12.96	N/A	17.08	26.14	18.31
mean (1–4)		**10.98**	11.34	13.59	15.16	17.17	27.02	18.98

a function of their dimensionality.

The PR configuration of a 576-D descriptor learnt on the Notre Dame set is depicted in Fig. 3.4 (left). Pooling regions are shown as circles with the radius equal to their Gaussian σ (the actual size of the Gaussian kernel is 3σ). The pooling regions' weights are colour-coded. Note that σ increases with the distance from the patch centre, which is also specific to certain hand-crafted descriptors, e.g. DAISY [Tola et al., 2008]. In our case, no prior has been put on the pooling region location and size: the PR parameters space was sampled uniformly, and the optimal configuration was automatically discovered by learning. Even though the PR weights near the patch centre are mostly small, the contribution of the pixels in the patch centre is higher than that of the pixels further from it, as shown in Fig. 3.4 (middle). This is explained by the fact that each Gaussian PR filter is normalised to a unit mass,

Table 3.3: **False positive rate (%) (at 95% recall) for compressed descriptors.** Yos: Yosemite, ND: Notre Dame, Lib: Liberty.

Train set	Test set	PR-proj-bin 48f→64b (64b)	PR-proj-bin 64f→128b (128b)	PR-proj-bin 80f→1024b (1024b)	PR-proj-pq 64f→64pq (64pq)	PR-proj-pq 80f→1024pq (1024pq)	Trzcinski et al. (64b)	Boix et al. (1360b)
Yos	ND	14.37	10.0	**7.09**	12.91	**6.82**	14.54	8.52
Yos	Lib	23.48	18.64	**15.15**	20.15	**14.59**	21.67	15.52
ND	Yos	18.46	13.41	**8.5**	19.32	**10.07**	18.97	8.81
ND	Lib	20.35	16.39	**12.16**	17.97	**12.42**	20.49	15.6
Lib	Yos	24.02	19.07	**14.84**	22.11	**11.22**	22.88	N/A
Lib	ND	15.2	11.55	**8.25**	14.82	**7.22**	16.90	N/A
mean		19.31	14.84	**11.0**	17.88	**10.39**	19.24	N/A
mean (1–4)		19.17	14.61	**10.73**	17.59	**10.98**	18.92	12.11

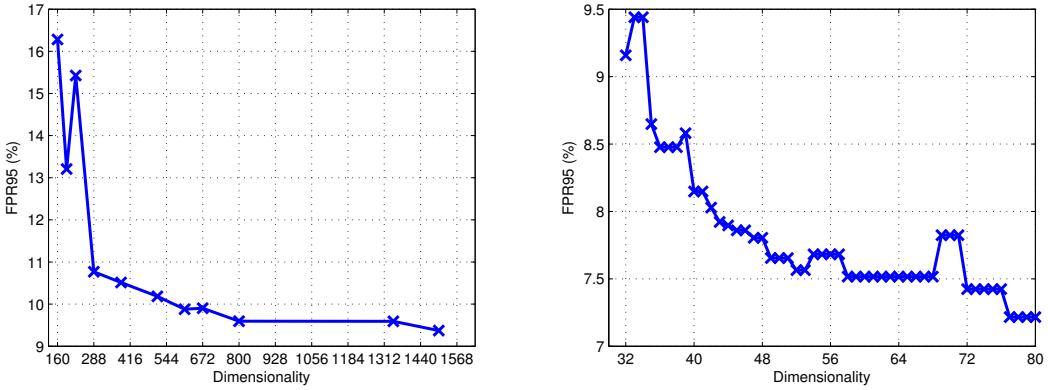


Figure 3.3: **Dimensionality vs error rate.** Training was performed on Liberty, testing – on Notre Dame. **Left**: learnt pooling regions. **Right**: learnt projections for 608-D PR descriptor on the left.

so the relative contribution of pixels is higher for the filters of smaller radius (like the ones selected in the centre). Interestingly, the pattern of pixel contribution, corresponding to the learnt descriptor, resembles the Gaussian weighting employed in hand-crafted methods, such as SIFT.

In Fig. 3.4 (right) we show the PR configuration learnt without the symmetry constraint, i.e. individual PRs are not organised into rings. Similarly to the symmetric configurations, the radius of PRs located further from the patch centre is larger than the radius of PRs near the centre. Also, there is a noticeable circular pattern of PR locations, especially on the left and right of the patch, which justifies our PR symmetry constraint. We note that this constraint, providing additional

regularisation, dramatically reduces the number of parameters to learn: when PRs are grouped into the rings of 8, a single weight is learnt for all PRs in a ring. In other words, a single element of the w vector (Sect. 3.2) corresponds to 8 PRs. In the case of asymmetric configurations, each PR has its own weight, so for the same number of candidate PRs, the w vector becomes 8 times longer, which significantly increases the computational burden. We did not observe any increase in performance when using asymmetric configurations, so in the following experiments, symmetric PR configurations are used.

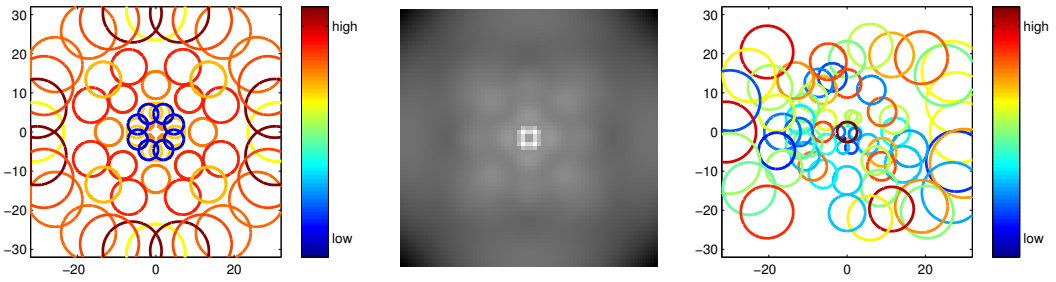


Figure 3.4: **Left:** learnt symmetric pooling regions configuration in a 64×64 feature patch. **Middle:** relative contribution of patch pixels (computed by the weighted averaging of PR Gaussian filters using the learnt weights, shown on the left). **Right:** learnt asymmetric pooling regions configuration.

Learning discriminative dimensionality reduction. For dimensionality reduction experiments, we utilised learnt PR descriptors with dimensionality limited by 640 (third column in Table 3.1) and learnt linear projections onto lower-dimensional spaces as described in Sect. 3.3. In Table 3.2 we compare our results with the best results presented in [Brown et al., 2011] (6-th column), [Trzcinski et al., 2012] (7-th column), as well as the unsupervised rootSIFT descriptor of [Arandjelović and Zisserman, 2012] and its supervised projection (rootSIFT-proj), learnt using the formulation of Sect. 3.3 (columns 8–9). Of these four methods, the best results are achieved by [Brown et al., 2011]. To facilitate a fair comparison, we learn three types of descriptors with different dimensionality: $\leq$80-D, $\leq$64-D, $\leq$32-D (columns 3–5).

As can be seen, even with low-dimensional 32-D descriptors we outperform all other methods in terms of the average error rate over different training/test set combinations: 13.59% vs 15.16% for [Brown et al., 2011]. It should be noted that we obtain projection matrices by discriminative supervised learning, while in [Brown et al., 2011] the best results were achieved using PCA, which outperformed LDA in their experiments. In our case, both PCA and LDA were performing considerably worse than the learnt projection. Our descriptors with higher (but still reasonably low) dimensionality achieve even lower error rates, setting the state of the art for the dataset: 10.75% for $\leq$64-D, and 10.38% for $\leq$80-D.

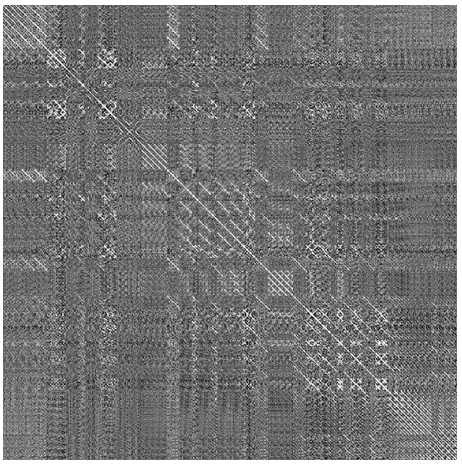


Figure 3.5: **Learnt Mahalanobis matrix A.** The matrix corresponds to projection from 576-D to 73-D space (brighter pixels correspond to larger values).

In Fig. 3.3 (bottom) we show the dependency of the error rate on the projected space dimensionality. As can be seen, the learnt projections allow for significant (order of magnitude) dimensionality reduction, while lowering the error at the same time. In Fig. 3.5 (left) we visualise the learnt Mahalanobis matrix A (Sect. 3.3) corresponding to discriminative dimensionality reduction. It has a clear block structure, with each block corresponding to a group of pooling regions. This indicates that the dependencies between pooling regions within the same ring and across the rings are learnt together with the optimal weights for the neighbouring orientation

bins within each PR.

Descriptor compression. The PR-proj descriptors evaluated above are inherently real-valued. To obtain a compact and fast-to-match representation, the descriptors can be compressed using either binarisation or product quantisation. We call the resulting descriptors PR-proj-bin and PR-proj-pq respectively, and compare them with the state-of-the-art binary descriptors of [Trzcinski et al., 2013, Boix et al., 2013]. The binary descriptor of [Trzcinski et al., 2013] is low-dimensional (64-D), while [Boix et al., 2013] proposes a more accurate, but significantly longer, 1360-D, representation.

As pointed out in Sect. 3.6, binarisation based on frame expansion can produce binary descriptors with any desired dimensionality, as long as it is not smaller than the dimensionality of the underlying real-valued descriptor. The dependency of the mean error rate on the dimensionality is shown in Fig. 3.6 for PR-proj-bin descriptors computed from different PR-proj descriptors. Given a desired binary descriptor dimensionality (bit length), e.g. 64-D, it can be computed from PR-proj descriptors of different dimensionality (32-D, 48-D, 64-D in our experiments). Higher-dimensional PR-proj descriptors have better performance (Table 3.2), but higher quantisation error (Sect. 3.6) when compressed to a binary representation. For instance, compressing 48-D PR-proj descriptors to 64 bit leads to better performance than compressing 64-D PR-proj (which has higher quantisation error) or 32-D PR-proj (which has worse initial performance). In general, it can be observed (Fig. 3.6) that using higher-dimensional (80-D) PR-proj for binarisation consistently leads to best or second-best performance.

In columns 3–5 of Table 3.3 we report the performance of our PR-proj-bin binary descriptors. The 64-bit descriptor has on average 0.07% higher error rate than the descriptor of [Trzcinski et al., 2013], but it should be noted that they employed a

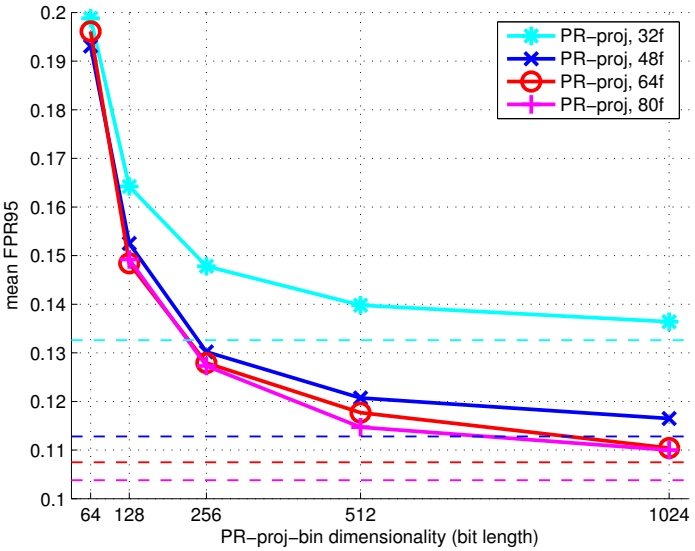


Figure 3.6: **Mean error rate vs dimensionality for binary PR-proj-bin descriptors.** The descriptors were computed from real-valued 32-D, 48-D, 64-D, and 80-D PR-proj descriptors. The error rates of the PR-proj descriptors are shown with dashed horizontal lines of the same colour as used for the respective binary descriptors.

dedicated framework for binary descriptor learning, while in our case we obtained the descriptor from our real-valued descriptors using a simple, but effective procedure of Sect. 3.6. Also, in [Trzcinski et al., 2013] it is mentioned that learning higher-dimensional binary descriptors using their framework did not result in performance improvement. In our case, we can explore the "bit length – error rate" trade-off by generating a multitude of binary descriptors with different length and performance. Our 1024-bit descriptor (column 5) significantly outperforms both [Trzcinski et al., 2013] and [Boix et al., 2013] (by 8.24% and 1.38% respectively), even though the latter use a higher dimensinal descriptor. We also note that the performance of 1024-bit PR-proj-bin descriptor is close to that of 80-D (2560 bit) PR-proj descriptor, which was used to generate it. Finally, our 128-bit PR-proj-bin descriptor provides a middle ground, with its 4.47% lower error rate than 64-bit descriptor, but still compact representation. Using LSH [Charikar, 2002] to compress the same PR-proj

descriptor to 128-bit leads to 3.07% higher error rate than frame expansion, which mirrors the findings of [Jégou et al., 2012a].

We also evaluate descriptor compression using (symmetric) product quantisation [Jégou et al., 2010] (see also Sect. 2.1.4). The error rates for the compressed 64-bit and 1024-bit PR-proj-pq descriptors are shown in columns 6–7 of Table 3.3. Compression using PQ is more effective than binarisation: 64-bit PR-proj-pq has 1.43% lower error than 64-bit PR-proj-bin, while 1024-bit PR-proj-pq outperforms binarisation by 0.61% and, in fact, matches the error rates of the uncompressed 80-D PR-proj descriptor (column 3 of Table 3.2).

While PQ compression is more effective in accuracy, in terms of the matching speed binary descriptors are the fastest: average Hamming distance computation time between a pair of 64 bit descriptors was measured to be 1.3ns (1ns=10^{-9}s) on an Intel Xeon L5640 CPU. PQ-compressed descriptors with the same 64 bit footprint (speeded-up using lookup tables) require 38.2ns per descriptor pair. For reference, SSE-optimised L^2 distance computation between 64-D single-precision vectors requires 53.5ns.

Summary. Both our pooling region and dimensionality reduction learning methods significantly outperform those of [Brown et al., 2011]. It is worth noting that the non-linear feature transform we used (Sect. 3.1) corresponds to the T1b block in [Brown et al., 2011]. According to their experiments, it is outperformed by more advanced (and computationally complex) steerable filters, which they employed to obtain their best results. This means that we achieve better performance with a simpler feature transform, but more sophisticated learning framework. We also achieve better results than [Trzcinski et al., 2012], where a related feature transform was employed, but PRs and dimensionality reduction were learnt using greedy optimisation based on boosting.

Our binary descriptors, obtained from learnt low-dimensional real-valued descriptors, achieve lower error rates than the recently proposed methods [Trzcinski and Lepetit, 2012, Trzcinski et al., 2013, Boix et al., 2013], where learning was tailored to binary representation.

The ROC curves for our real-valued and compressed descriptors are shown in Fig. 3.7 for all combinations of training and test sets.

3.8 Conclusion

In this chapter we introduced a generic framework for learning two major components of feature descriptor computation: spatial pooling and discriminative dimensionality reduction. We also demonstrated that the learnt descriptors are amenable to compression using product quantisation and binarisation. Rigorous evaluation showed that the proposed algorithm outperforms state-of-the-art real-valued and binary descriptors on a challenging dataset. This was achieved via the use of convex learning formulations, coupled with large-scale regularised optimisation techniques. Each of the two presented learning frameworks can be used independently and applied to other computer vision tasks, e.g. object part discovery and face verification.

3.8.1 Scientific Relevance and Impact

Since our framework was published in [Simonyan et al., 2012b], it has been cited by several relevant works [Trzcinski et al., 2012, 2013, Boix et al., 2013, Berg and Belhumeur, 2013, Wang et al., 2013], which we briefly discuss here. Of particular relevance are the recently proposed descriptor learning methods [Trzcinski et al., 2012, 2013, Boix et al., 2013], reviewed in Sect. 2.1.2. As can be seen from the comparison in Sect. 3.7, their results on Local Image Patches Dataset are still somewhat worse than ours. One of the reasons for that could be that they use non-convex

optimisation procedures, which can result in the suboptimal descriptor models being learnt. In [Berg and Belhumeur, 2013], a large number of mid-level features for fine-grained recognition was trained in such a way that each feature is constrained to a certain spatial support region. In their case, the region selection was performed by thresholding the weights learnt by an L^2-regularised SVM. A more principled way of support region selection would be based on the sparsity-inducing L^1 regularisation, as we used for pooling region selection in Sect. 3.2. In [Wang et al., 2013], a learning formulation, similar to ours, was used to learn the dimensionality reduction for kernel descriptors. Following our work, the optimisation of the Mahalanobis matrix, regularised by the nuclear norm, was carried out using the RDA optimisation method.

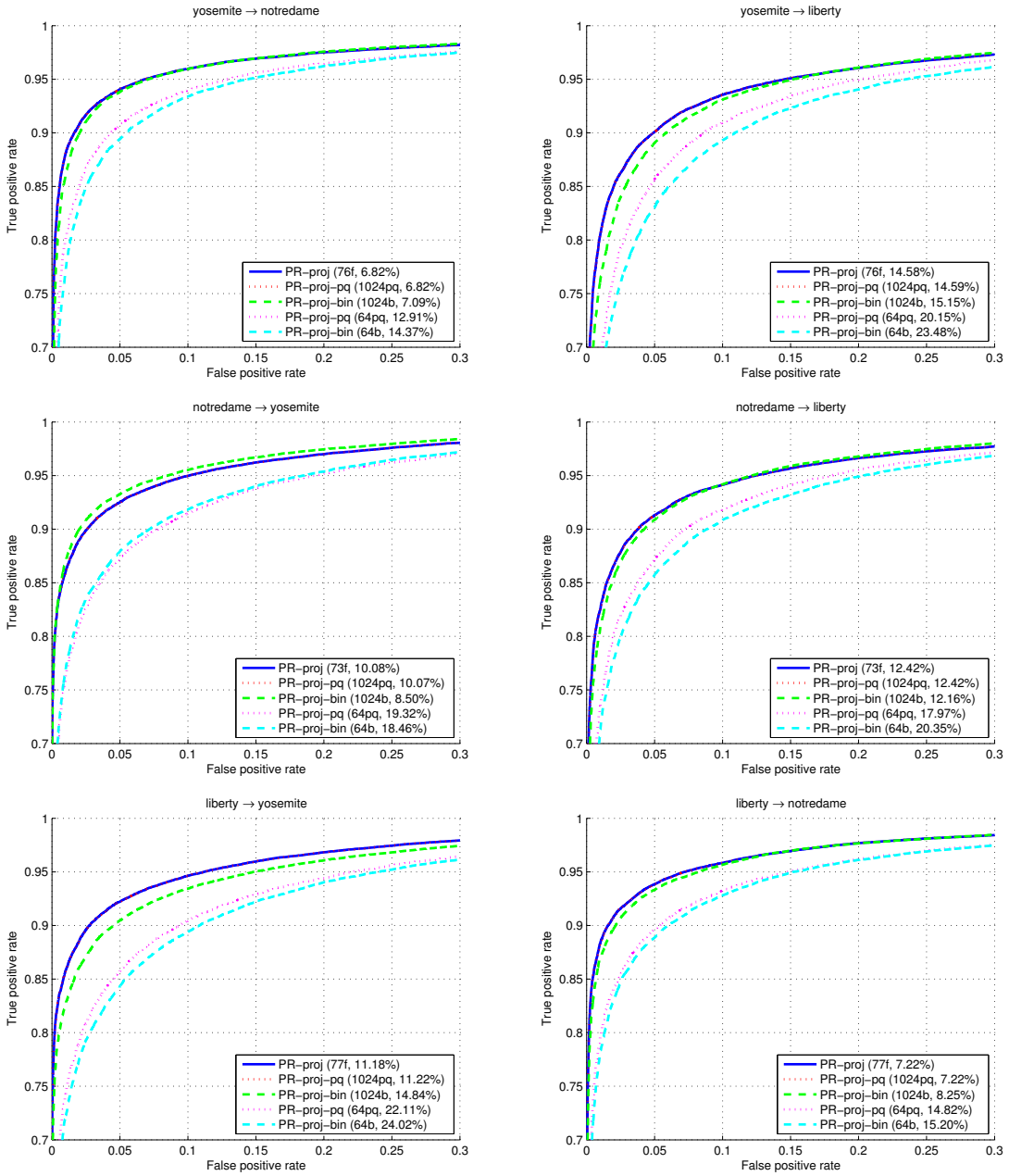


Figure 3.7: **Descriptor matching ROC curves for six combinations of training and test sets of the Patches dataset [Brown et al., 2011].** For each of the plots, the sets are indicated in the title as "training→test". For each of the compared descriptors, its dimensionality, type, and false positive rate at 95% recall are given in parentheses (see also Table 3.2 and Table 3.3).

Chapter 4

Learning Descriptors from Unannotated Image Collections

In the previous chapter we described a framework for learning local descriptors from the full supervision, i.e. when a training set of matching and non-matching pairs of patches is available. One possible way of obtaining the feature correspondences for descriptor learning would be to compute the 3-D reconstruction [Brown et al., 2011] of scenes present in the dataset, but this requires a large number of images of the same scene to perform well, which is not always practical. In this chapter, we describe a novel formulation for obtaining feature correspondences from image datasets using only extremely weak supervision. Together with the learning frameworks of Chapter 3 this provides an algorithm for automatically learning descriptors from such datasets. In this challenging scenario, the only information given to the algorithm is that *some* (but unknown) pairs of dataset images contain a common *part*, so that correspondences can be established between them. The assumption is valid for the image collections considered in this chapter (Sect. 4.3).

The rest of the chapter is organised as follows. In Sect. 4.1, we describe the automatic training data generation stage, which computes the data required for de-

scriptor learning. The details of the learning formulation are then given in Sect. 4.2. The learnt descriptors are then plugged into a conventional image retrieval engine [Philbin et al., 2007], and evaluated using retrieval-specific evaluation protocol on Oxford5K and Paris6K image collections (Sect. 4.3). Apart from showing the superiority of the learnt descriptors, we also demonstrate that the choice of the underlying feature region detection method and its parameters strongly affects the retrieval performance.

4.1 Training Data Generation

The purpose of this step is to automatically extract learning data from an image collection, so that it is can further be used in the learning procedure. In particular, we would like to extract a set of non-matched feature regions pairs together with the set of *putative* matches. This proceeds in two stages: first, homographies are established between randomly sampled image pairs using nearest-neighbour SIFT descriptor matches and RANSAC [Philbin et al., 2010]; second, region correspondences are established between the image pairs using only the homography (not SIFT descriptors). This ensures that the resulting correspondences are independent of SIFT.

In more detail, we begin with automatic homography estimation between the random image pairs. This involves a standard pipeline [Mikolajczyk et al., 2005] of: affine-covariant (elliptical) region detection, computing SIFT descriptors for the regions, and estimating an affine homography using the robust RANSAC algorithm on the putative SIFT matches. Only the pairs for which the number of RANSAC inliers is larger than a threshold (set to 50 in our experiments) are retained. Then, in stage two, for each feature $\mathbf{x}$ of the reference image, we compute the sets $P(\mathbf{x})$ and $N(\mathbf{x})$ of putative positive and negative matches in the target image based on the

homographies and the descriptor measurement region overlap criterion [Mikolajczyk et al., 2005] as follows. Each descriptor measurement region (an upscaled elliptical detected region) in the target image is projected to the reference image plane using the estimated homography, resulting in an elliptical region. Then, the overlap ratio between this region and each of the measurement regions in the reference image is used to establish the "putative positive" and "negative" matches by thresholding the ratio with high (0.6) and low (0.3) thresholds respectively. Feature matches with the region overlap ratio between the thresholds are considered ambiguous and are not used in training (see Fig. 4.1 for illustration).

Figure 4.1: **A close-up of a pair of reference (left) and target (right) images from the Oxford5K dataset.** A feature region in the reference image is shown with solid blue. Its putative positive, negative, and ambiguous matches in the target image are shown on the right with green, red, and magenta respectively. Their projections to the reference image are shown on the left with dashed lines of the same colour. The corresponding overlap ratios (with the blue reference region ellipse) are: 0.74 for positive, 0.04 for negative, and 0.33 for ambiguous matches.

4.2 Self-Paced Descriptor Learning Formulation

Given a set of tuples $(\mathbf{x}, P(\mathbf{x}), N(\mathbf{x}))$, automatically extracted from the training image collection (Sect. 4.1), here we aim at learning a descriptor such that the NN of each feature $\mathbf{x}$ is one of the positive matches from $P(\mathbf{x})$. This is equivalent to enforcing the minimal (squared) distance from $\mathbf{x}$ to the features in $P(\mathbf{x})$ to be smaller

than the minimal distance to the features in $N(\mathbf{x})$:

$$\min_{\mathbf{y}\in P(\mathbf{x})} d_\eta(\mathbf{x},\mathbf{y}) < \min_{\mathbf{u}\in N(\mathbf{x})} d_\eta(\mathbf{x},\mathbf{u}), \tag{4.1}$$

where for brevity η denotes the descriptor parameters, such as PR weights w (Sect. 3.2) or the metric A (Sect. 3.3).

In certain cases, the reference image feature $\mathbf{x}$ can not be matched to a geometrically corresponding feature in the target image purely based on appearance. For instance, the target feature can be occluded, or the repetitive structure in the target image can make reliable matching impossible. Using such unmatchable features $\mathbf{x}$ in the constraints (4.1) introduces an unnecessary noise in the training set and disrupts learning. Therefore, we introduce a binary latent variable $b(\mathbf{x})$ which equals 0 iff the match can not be established. This leads to the optimisation problem:

$$\begin{aligned} &\arg\min_{\eta,b,\mathbf{y}_P} \sum_{\mathbf{x}} b(\mathbf{x})\,\mathcal{L}\left(d_\eta\left(\mathbf{x},\mathbf{y}_P(\mathbf{x})\right) - \min_{\mathbf{u}\in N(\mathbf{x})} d_\eta(\mathbf{x},\mathbf{u})\right) + R(\eta) \\ &\text{s.t.}\;\; \mathbf{y}_P(\mathbf{x}) = \arg\min_{\mathbf{y}\in P(\mathbf{x})} d_\eta(\mathbf{x},\mathbf{y});\; b(\mathbf{x})\in\{0,1\};\; \sum_{\mathbf{x}} b(\mathbf{x}) = K \end{aligned} \tag{4.2}$$

where $\mathbf{y}_P(\mathbf{x})$ is a latent variable storing the nearest-neighbour of the feature $\mathbf{x}$ among the putative positive matches $P(\mathbf{x})$, $R(\eta)$ is the regulariser (e.g. sparsity-enforcing L^1 norm or nuclear norm), and K is a hyper-parameter, which sets the number of samples to use in training and prevents all $b(\mathbf{x})$ from being set to zero. As can be seen, each feature $\mathbf{x}$ is equipped with two latent variables: binary $b(\mathbf{x})$, which denotes the plausibility of feature matching based on appearance, and $\mathbf{y}_P(\mathbf{x})$, which stores the correct match, if matching is possible.

The objective (4.2) is related to large margin nearest neighbour (Sect. 2.3.3) and self-paced learning [Kumar et al., 2010], and its local minimum can be found by alternation. Namely, with $b(\mathbf{x})$ and $\mathbf{y}_P(\mathbf{x})$ fixed for all $\mathbf{x}$, the optimisation prob-

lem (4.2) becomes convex (due to the convexity of $-\min$), and is solved for η using RDA (Sect. 3.5). Then, given η, $\mathbf{y}_P(\mathbf{x})$ can be updated; finally, given η and $\mathbf{y}_P(\mathbf{x})$, we can update $b(\mathbf{x})$ by setting it to 1 for $\mathbf{x}$ corresponding to the smallest K values of the loss $\mathcal{L}\left(d_\eta\left(\mathbf{x}, \mathbf{y}_P(\mathbf{x})\right) - \min_{\mathbf{u}\in N(\mathbf{x})} d_\eta(\mathbf{x}, \mathbf{u})\right)$. Each of these three steps reduces the value of the objective (4.2), which gives the convergence guarantee. The optimisation is repeated for different values of K, and the resulting model is selected on the validation set as the one which maximises the feature matching recall, i.e. the ratio of features $\mathbf{x}$ for which (4.1) holds.

Discussion. Our method accounts for the weak supervision and feature matching uncertainty using the latent variables formalism (4.2). It should be noted that even though we effectively select K *easiest feature pairs* for training, the *hardest negative feature* $\min_{\mathbf{u}\in N(\mathbf{x})} d_\eta(\mathbf{x}, \mathbf{u})$ is used within each of these pairs. This is different from the training set generation technique of Philbin et al. [2010], who constrained the positives to be those SIFT Nearest Neighbours (NN), which have been marked as inliers by the RANSAC estimation procedure. As negatives, they employed a fixed set of NN outliers and non-NN matches. This means that the positives can already be matched by SIFT, while our goal is to learn a better descriptor. Also, using a fixed subset of negative matches can result in missing hard negatives, which are important for training. Another alternative of ignoring appearance and finding correspondences purely based on geometry is also problematic. It can pick up occlusions and repetitive structure, which, being unmatchable based on appearance, would disrupt learning.

4.3 Experiments

In this section the proposed learning framework is evaluated on challenging Oxford Buildings (Oxford5K) and Paris Buildings (Paris6K) datasets and compared against

the rootSIFT baseline [Arandjelović and Zisserman, 2012], as well as the descriptor learning method of [Philbin et al., 2010].

4.3.1 Datasets and Evaluation Protocol

The evaluation is carried out on the Oxford Buildings and the Paris Buildings datasets. The Oxford Buildings dataset consists of 5062 images capturing various Oxford landmarks. It was originally collected for the evaluation of large-scale image retrieval methods [Philbin et al., 2007]. The only available annotation is the set of queries and ground-truth image labels, which define relevant images for each of the queries. The Paris Buildings dataset includes 6412 images of Paris landmarks and is also annotated with queries and labels. Both datasets exhibit a high variation in viewpoint and illumination.

The performance measure is specific to the image retrieval task and is computed in the following way. For each of the queries, the ranked retrieval results (obtained using the framework of [Philbin et al., 2007]) are assessed using the ground-truth landmark labels. The area under the resulting precision-recall curve (average precision) is the performance measure for the query. The performance measure for the whole dataset is obtained by computing the mean Average Precision (mAP) across all queries.

In the comparison, we employed three types of the visual search engine [Philbin et al., 2007]: *tf-idf* uses the tf-idf index computed on quantised descriptors (500K visual words); *tf-idf-sp* additionally re-ranks the top 200 images using RANSAC-based spatial verification. The third engine is based on nearest-neighbour matching of raw (non-quantised) descriptors and RANSAC-based spatial verification. We use tf-idf and tf-idf-sp in the majority of experiments, since using raw descriptors for large-scale retrieval is not practical. Considering that tf-idf retrieval engines are based on vector-quantised descriptors, the descriptor dimensionality is not crucial

in this scenario, so we learn the descriptors with dimensionality similar to that of SIFT (128-D).

4.3.2 Feature Detector and Measurement Region Size

Here we assess the effect that the feature detection method and the measurement region size have on the image retrieval performance on the Oxford5K dataset. For completeness, we begin with a brief description of the conventional feature extraction pipeline [Mikolajczyk et al., 2005] employed in our retrieval framework. In each image, feature detection is performed using an affine-covariant detector, which produces a set of elliptically-shaped feature regions, invariant to the affine transformation of an image. As pointed out in [Matas et al., 2002, Mikolajczyk et al., 2005], it is beneficial to capture a certain amount of context around a detected feature. Therefore, each detected feature region is isotropically enlarged by a constant scaling factor to obtain the descriptor measurement region. The latter is then transformed to a square patch, which can be optionally rotated w.r.t. the dominant orientation to ensure in-plane rotation invariance. Finally, a feature descriptor is computed on the patch.

In [Philbin et al., 2007, 2010, Simonyan et al., 2012b] feature extraction was performed using the Hessian-Affine (HesAff) detector [Mikolajczyk et al., 2005], $\sqrt{3}$ measurement region scaling factor, and rotation-invariant patches. We make two important observations. First, *not* enforcing patch rotation invariance leads to 5.1% improvement in mAP, which can be explained by the instability of the dominant orientation estimation procedure, as well as the nature of the data: landmark photos are usually taken in the upright position, so in-plane rotation invariance is not required and can reduce the discriminative power of the descriptors. Second, significantly higher performance can be achieved by using a higher measurement region scaling factor, as shown in Fig. 4.2 (red curve).

One of alternatives to the Hessian operator for feature detection is the Difference of Gaussians (DoG) function [Lowe, 2004]. Initially, DoG detector was designed to be (in)variant to the similarity transform, but affine invariance can also be achieved by applying the affine adaptation procedure [Mikolajczyk and Schmid, 2002, Schaffalitzky and Zisserman, 2002] to the detected DoG regions. We call the resulting detector DoGAff, and evaluate the publicly available implementation in VLFeat package [Vedaldi and Fulkerson, 2010]. For DoGAff, not enforcing the patch orientation invariance also leads to 5% mAP improvement. The dependency of the retrieval performance on measurement region scaling factor is shown in Fig. 4.2 (blue curve). As can be seen, using DoGAff leads to considerably higher retrieval performance than HesAff. It should be noted, however, that the improvement comes at the cost of a larger number of detected regions: on average, HesAff detects 3.5K regions per image on Oxford5K, while DoGAff detects 5.5K regions.

In the sequel, we employ DoGAff feature detector (with 12.5 scaling factor and without enforcing the in-plane rotation invariance) for two reasons: it achieves better performance and the source code is publicly available. The same detected regions are used for all compared descriptors.

4.3.3 Descriptor Learning Results

In the descriptor learning experiments, we used the Oxford5K dataset for training and both Oxford5K and Paris6K for evaluation. We note that ground-truth matches are not available for Oxford5K; instead, the training data is extracted *automatically* (Sect. 4.2). The evaluation on Oxford5K corresponds to the use case of learning a descriptor for a particular image collection based on extremely weak supervision. At the same time, the evaluation on Paris6K allows us to assess the *generalisation* of the learnt descriptor to different image collections. Similarly to the experiments in Sect. 3.7, we learn a 576-D PR descriptor (shown in Fig. 4.3, right) and its

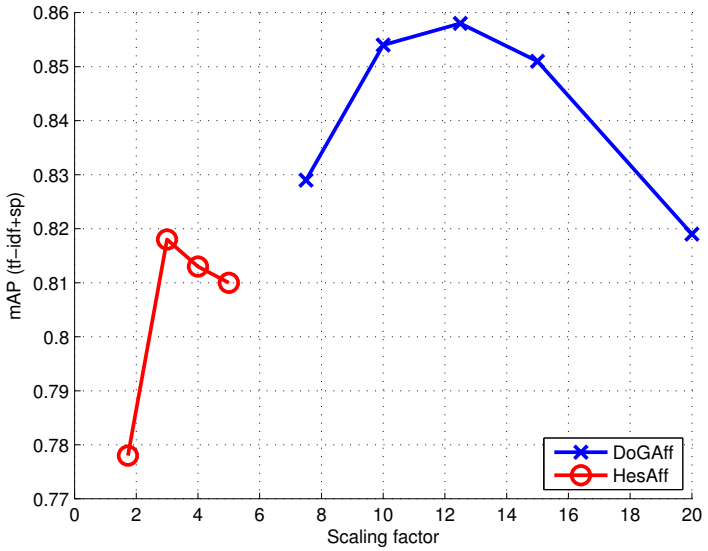


Figure 4.2: **The dependency of retrieval mAP on the feature detector and the measurement region scaling factor.** The results were obtained on the Oxford5K dataset using the rootSIFT descriptor and tf-idf-sp retrieval engine.

discriminative projection onto 127-D subspace.

The mAP values computed using different "descriptor – search engine" combinations are given in Table 4.1. First, we note that the performance of rootSIFT can be noticeably improved by adding a discriminative linear projection on top of it, learnt using the proposed framework. As a result, the projected rootSIFT (rootSIFT-proj) outperforms rootSIFT on both Oxford5K (+2.5%/3.0% mAP using tf-idf/tf-idf-sp respectively) and Paris6K (+2.2%/2.1% mAP). Considering that rootSIFT has already moderate dimensionality (128-D), there is no need to perform dimensionality reduction in this case, so we used Frobenius-norm regularisation of the Mahalanobis matrix A in (3.10), (4.2).

The proposed PR-proj descriptor (with both pooling regions and low-rank projection learnt) performs similarly to rootSIFT-proj on Oxford5K: +3.0%/2.5% compared to the rootSIFT baseline, and +0.5%/−0.5% compared to rootSIFT-proj. On Paris6K, PR-proj outperforms both rootSIFT (+3.0%/3.1%) and rootSIFT-proj

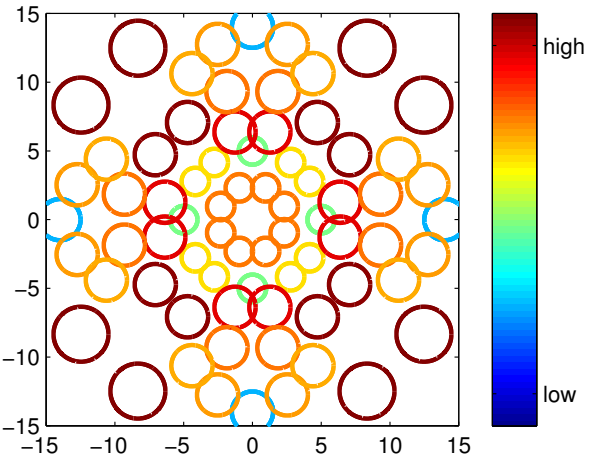


Figure 4.3: **Pooling region configuration, learnt on Oxford5K.** It corresponds to a 576-D descriptor (before projection).

(+0.8%/1%). When performing retrieval using raw descriptors without quantisation, PR-proj performs better than rootSIFT-proj on both Oxford5K (92.6% vs 91.9%) and Paris6K (86.9% vs 86.2%).

In summary, both learnt descriptors, rootSIFT-proj and PR-proj, lead to better retrieval performance compared to the rootSIFT baseline. The mAP improvements brought by the learnt descriptors are consistent for both datasets and retrieval engines, which indicates that our learnt models generalise well.

Table 4.1: **mAP on Oxford5K and Paris6K for learnt descriptors and rootSIFT** [Arandjelović and Zisserman, 2012]. For these experiments, DoGAff feature detector was used (Sect. 4.3.2).

Descriptor	mAP	
	tf-idf	tf-idf-sp
Oxford5K		
rootSIFT baseline	0.795	0.858
rootSIFT-proj	0.820	**0.888**
PR-proj	**0.825**	0.883
Paris6K		
rootSIFT baseline	0.780	0.796
rootSIFT-proj	0.802	0.817
PR-proj	**0.810**	**0.827**

Comparison with [Philbin et al., 2010]. We note that our baseline retrieval system (DoGAff–rootSIFT–tf-idf-sp) performs significantly better (+21.1%) than the one used in [Philbin et al., 2010]: 85.8% vs 64.7%. This is explained by the following reasons: (1) different choice of the feature detector (Sect. 4.3.2); (2) more discriminative rootSIFT descriptor [Arandjelović and Zisserman, 2012] used as the baseline; (3) differences in the retrieval engine implementation. Therefore, to facilitate a fair comparison with the best-performing linear and non-linear learnt descriptors of [Philbin et al., 2010], in Table 4.2 we report the results [Simonyan et al., 2012b] obtained using our descriptor learnt on top of the same feature detector as used in [Philbin et al., 2007, 2010]. Namely, we used HesAff with $\sqrt{3}$ measurement region scaling factor and rotation-invariant descriptor patches. With these settings, our baseline result gets worse, but much closer to [Philbin et al., 2010]: 66.7% using HesAff–SIFT–tf-idf-sp. To cancel out the effect of the remaining difference in the baseline results, we also show the mAP improvement relative to the corresponding baseline for our method and [Philbin et al., 2010].

As can be seen, a linear projection on top of SIFT (SIFT-proj) learnt using our framework results in a bigger improvement over SIFT than that of [Philbin et al., 2010]. Learning optimal pooling regions leads to further increase of performance, surpassing that of non-linear SIFT embeddings [Philbin et al., 2010]. In our case, the drop of mAP improvement when moving to a different image set (Paris6K) is smaller than that of [Philbin et al., 2010], which means that our models generalise better.

The experiments with two different feature detection methods, presented in this section, indicate that the proposed learning framework brings consistent improvement irrespective of the underlying feature detector.

Table 4.2: **mAP on Oxford5K and Paris6K for learnt descriptors (ours and those of [Philbin et al., 2010]) and SIFT.** Feature detection was carried out using the HesAff detector to ensure a fair comparison with [Philbin et al., 2010].

Descriptor	mAP		mAP impr. (%)	
	tf-idf	tf-idf-sp	tf-idf	tf-idf-sp
Oxford5K				
SIFT baseline	0.636	0.667	-	-
SIFT-proj	0.673	0.706	5.8	5.8
PR-proj	**0.709**	**0.749**	**11.5**	**12.3**
Philbin et al., SIFT baseline	0.613	0.647	-	-
Philbin et al., SIFT-proj	0.636	0.665	3.8	2.8
Philbin et al., non-linear	0.662	0.707	8	9.3
Paris6K				
SIFT baseline	0.656	0.668	-	-
PR-proj	**0.711**	**0.722**	**8.4**	**8.1**
Philbin et al., SIFT baseline	0.655	0.669	-	-
Philbin et al., non-linear	0.678	0.689	3.5	3

4.4 Conclusion

In this chapter, we have proposed an algorithm for learning discriminative local descriptors from image collections, where the ground-truth matches are not available. Our method builds on the formulations of Chapter 3, which are extended to accommodate such a weak supervision. The resulting local descriptor has been shown to improve the retrieval performance, compared to both supervised and unsupervised baselines. We have also shown that the performance of a conventional retrieval system [Philbin et al., 2007] can be substantially improved by using an affine-adapted DoG detector [Lowe, 2004] with a large descriptor measurement region size. It should be noted that our learnt descriptor provides a significant performance boost even when compared with this strong baseline.

Chapter 5

Improving VLAD and Fisher Vector Encodings

In this chapter we discuss the ways of improving the Fisher Vector (FV) [Perronnin et al., 2010] and VLAD [Jégou et al., 2010] feature encodings. These encodings, reviewed in Sect. 2.2.2, are known to achieve state-of-the-art performance on a number of image classification and retrieval benchmarks [Chatfield et al., 2011, Jégou et al., 2012b, Sánchez et al., 2013]. As shown in [Perronnin et al., 2010], an important part of the success of the FV representation lies in the appropriate projection of the encoded features, as well as the post-processing of the encoding. Namely, the PCA projection of the local descriptors (such as SIFT) was shown to be important, as was the Hellinger kernel mapping of the FV, which corresponds to signed square-rooting and L^2 normalisation.

Here, we further investigate the ways of improving VLAD and FV encodings for the image classification task, and make the following contributions. First, we evaluate the improvement brought by the intra-normalisation scheme [Arandjelović and Zisserman, 2013], applied to both VLAD and FV encodings in the image classification scenario (Sect. 5.2). Equipped with this normalisation scheme, in Sect. 5.2.1

we evaluate the extensions, such as the spatial coordinate augmentation [Krapac et al., 2011, Sánchez et al., 2012] and the hard-assignment FV – a fast variant of FV, which we propose for time-critical applications. Then, we show that PCA-whitening of local features significantly improves the VLAD encoding in the classification task (Sect. 5.3.1). Finally, in Sect. 5.3.2 we propose a method for learning linear transformations of local features, which allows us to improve the performance of VLAD even further, bridging the gap between VLAD and FV classification results.

5.1 Evaluation Protocol

We begin with describing our evaluation protocol. To compare different feature encodings, we use a conventional classification pipeline, similar to the one used in the comparison [Chatfield et al., 2011], and run it on the PASCAL VOC 2007 dataset [Everingham et al., 2010]. The dataset consists of about 10K images, split into training, testing, and validation sets, and labelled with 20 object classes. For each class, we learn and evaluate a linear SVM in the one-vs-rest manner, and the final performance is measured as the mean Average Precision (mAP) across all classes.

Our pipeline settings, except for the feature transformations and encoding methods, are fixed for all the experiments, and are similar to those of [Sánchez et al., 2012]. In more detail, SIFT is extracted densely using 32×32 patches and 4 pixels step. The extraction is carried out over 7 scales by starting from the image at twice the original resolution, and then downsampling it by a factor of $\sqrt{2}$ at each iteration. Dense SIFT features are then linearly transformed to facilitate encoding. In general, the linear transformation can be either dimensionality-preserving (e.g. the PCA rotation) or dimensionality-reducing (e.g. the PCA projection onto a lower-dimensional subspace). The transformed features are then encoded using FV

or VLAD. Taking into account that for the same size of a codebook, FV encoding is two times longer than VLAD (see Sect. 2.2.2), our VLAD codebook is twice as big as the FV codebook to ensure the same dimensionality of the encoding. Unless otherwise stated, we used 512 visual words for VLAD, 256 Gaussians for FV, and the spatial information was incorporated using Spatial Pyramid pooling (SPM), where the feature encodings were pooled over 8 cells: 2×2 grid, 3×1 (three horizontal stripes), and 1×1 (the whole image). Eight cell encodings were then stacked together and L^2 normalised to produce the final image representation.

5.2 Encoding Normalisation

In this section, we discuss the impact of the encoding normalisation scheme on the classification accuracy. The SIFT transformation is fixed to PCA due to the fact that PCA decorrelates features, making them amenable to modelling with diagonal-covariance GMM used in FV [Perronnin et al., 2010]. PCA was also shown to be beneficial for VLAD encoding in the image retrieval scenario [Jégou and Chum, 2012, Delhumeau et al., 2013]. Unless otherwise stated, we do not reduce the dimensionality of local features – by default, we perform the PCA rotation. As a reference point, we employ the signed square-rooting post-processing scheme, which was shown to improve the results of both FV [Perronnin et al., 2010] and VLAD [Jégou and Chum, 2012]. It consists in the following element-wise transform: $\text{sgn}(z)\sqrt{|z|}$, which is followed by the L^2 normalisation of the encoding (applied to the whole SPM-pooled vector in our case). This baseline is compared with two recently proposed alternatives: intra-normalisation [Arandjelović and Zisserman, 2013] and residual-normalisation [Delhumeau et al., 2013]. Both methods were originally applied to the VLAD encoding and evaluated in the image retrieval scenario.

Intra-normalisation of VLAD [Arandjelović and Zisserman, 2013] consists in the

Table 5.1: **Image classification results (mAP, %) on VOC 2007 for different combinations of encodings and normalisation schemes.** SPM – spatial pyramid pooling; AUG – spatial coordinate augmentation (Sect. 5.2.1).

encoding	square-rooting	residual-norm	intra-norm
PCA-SIFT + VLAD (SPM)	60.0	59.3	**61.1**
PCA-SIFT + FV (SPM)	62.5	N/A	**65.0**
PCA-SIFT + FV (AUG)	62.0	N/A	**63.8**

individual L^2 normalisation of each of the visual word "slots" (see (2.3) in Sect. 2.2.2). The benefit of such normalisation is that it equalises the contribution of different visual words, reducing the adverse burstiness effect [Jégou et al., 2009] of the SIFT distribution in real-world images. It can also be seen from the multiple kernel learning point of view: each visual word corresponds to a part (slot) of the VLAD vector, which, in turn, corresponds to a separate linear kernel. Normalisation of the feature vectors, corresponding to each of these kernels, leads to a better regularisation of the learning problem.

We also extend intra-normalisation to the FV encoding by the separate L^2 normalisation of the first and second order statistics of each k-th Gaussian (2.4):

$$\sum_p \phi_k^{(i)}(\mathbf{x}_p) \;\rightarrow\; \frac{1}{\| \sum_p \phi_k^{(i)}(\mathbf{x}_p) \|_2} \sum_p \phi_k^{(i)}(\mathbf{x}_p), \;\; \forall k, \;\; i = 1, 2 \tag{5.1}$$

Another VLAD normalisation technique, which we consider here, is the residual normalisation [Delhumeau et al., 2013], which is performed by the L^2 normalisation of the displacement of each feature $\mathbf{x}$ from its visual word $\mathbf{v}_k$ (2.3): $\mathbf{x}_p - \mathbf{v}_k \;\rightarrow\; \frac{\mathbf{x}_p - \mathbf{v}_k}{\| \mathbf{x}_p - \mathbf{v}_k \|_2}$. Such normalisation is more extreme than intra-normalisation, in a sense that it equalises the contribution of each local descriptor to the image encoding. In [Delhumeau et al., 2013] it was shown to outperform intra-normalisation on the image retrieval task, but here we show that the opposite holds true for the supervised classification scenario.

As can be seen from Table 5.1, intra-normalisation outperforms other normal-

isation methods on the VOC 2007 classification task. We stress that it provides a significant boost for both VLAD and FV encodings, in spite of the fact that it was originally proposed for VLAD. Our baseline result, achieved using FV encoding, spatial pyramid (SPM) pooling, and signed square-rooting, is 62.5% mAP. It is close to 63.0%, reported for the pipeline with similar settings in [Sánchez et al., 2012], which means that our implementation is valid. By using intra-normalisation, we get a significant improvement of 2.5%, achieving state-of-the-art classification performance of 65.0% mAP (among dense SIFT feature encoding methods with SPM pooling).

5.2.1 Additional Fisher Vector Experiments

Now that we have shown that intra-normalisation is beneficial for both VLAD and FV encodings on the VOC 2007 classification benchmark, we present the results of some additional experiments with the Fisher vector.

Spatial coordinate augmentation. First, we assess the the spatial coordinate augmentation scheme [Sánchez et al., 2012] (discussed in Sect. 2.2.2), which is an alternative way of incorporating the spatial information into the image representation. As can be seen from Table 5.1 (the last row), the coordinate augmentation (AUG) also benefits from the intra-normalisation, but performs worse than SPM (with the same number of Gaussians, set to 256). However, since the spatial pyramid pooling is not involved, the FV-AUG image representation is ~ 8 times shorter than FV-SPM (we use 8 SPM cells). This allows us to increase the number of Gaussians in the GMM, while keeping the FV dimensionality tractable. As noted in [Sánchez et al., 2012], this leads to better performance than that of SPM, and our results, reported in Table 5.2, confirm that the same holds true for the intra-normalised FV encodings. Namely, increasing the number of Gaussians from 256 to 512 leads to

the mAP improvement from 63.8% to 65.4%, and further to 66.5% when using 1024 Gaussians. This is considerably better than 65.0% mAP, which we achieved using higher-dimensional intra-normalised FV encoding, based on 256 Gaussians and SPM.

We note that the augmentation can not be immediately combined with the VLAD encoding. The reason is that GMM, used in FV, can automatically balance the appearance and the location parts of the spatially augmented SIFT descriptor, but the k-means clustering, used in VLAD, can not achieve that. Therefore, to make the augmentation scheme compatible with VLAD, one would have to multiply the feature spatial coordinates by a cross-validated balancing constant, which we have not tried in this work.

Hard-assignment Fisher vector. In the original FV encoding formulation, each feature $\mathbf{x}$ is soft-assigned to all K Gaussians of the GMM by computing the assignment weights (2.5) as the responsibilities of the GMM component k for the feature $\mathbf{x}$ (see Sect. 2.2.2 for details). The assignment to several (or all) Gaussians, however, increases the computation time, potentially putting FV at a disadvantage compared to VLAD in time-critical applications, e.g. on-the-fly category retrieval [Chatfield and Zisserman, 2012].

As a trade-off between the encoding efficiency and the classification accuracy, here we propose the hard-assignment FV encoding (hard-FV), which can be seen as the middle ground between VLAD and the conventional soft-assignment FV. The only difference between FV and hard-FV is that the latter replaces the soft-assignment (2.5) with the hard assignment of the feature $\mathbf{x}$ to the Gaussian with

the max likelihood:

$$\alpha_k(\mathbf{x}) = \begin{cases} 1 & \text{if } k = \arg\max_j \pi_j \mathcal{N}_j(\mathbf{x}) \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

We note that in spite of the hard assignment, hard-FV is different from VLAD (and its second-order extensions [Picard and Gosselin, 2011]), since it uses GMM clustering instead of the k-means clustering, which allows it to exploit the second-order information.

On VOC 2007, the hard-FV encoding of spatially augmented, PCA-rotated SIFT features achieved mAP of 65.2% and 66.2% using 512 and 1024 Gaussians respectively, which is close to 65.4% and 66.5% achieved using the conventional FV with the same GMM (Table 5.2). In terms of the computation speed, our MEX-optimised Matlab implementation of hard-FV encoding was measured to be ~ 4 times faster than the conventional FV implementation used in [Chatfield et al., 2011].

Summary of FV results. The results, reported above, were obtained using the PCA-rotated SIFT without dimensionality reduction. Considering that in a number of prior works [Perronnin et al., 2010, Chatfield et al., 2011, Sánchez et al., 2012] the SIFT dimensionality is reduced before encoding, in Table 5.2 we summarise our best FV results, and report mAP for both PCA rotation to 128-D and PCA projection to 64-D. As can be seen, the best performance is achieved without dimensionality reduction. At the same time, reducing the local feature dimensionality by a factor of 2 leads to an insignificant drop of performance, while being beneficial in terms of the processing speed and memory footprint. Also, the hard-assignment FV is close to the soft-assignment FV, while being significantly faster.

Our best result (66.5% with 1024 Gaussians in the GMM) sets the new state of the art on VOC 2007 classification benchmark among the methods, solely based on

Table 5.2: **Image classification results (mAP, %) on VOC 2007 for different FV pipeline settings and PCA-SIFT dimensionalities.** Image descriptor dimensionality is specified in parentheses. For each setting, we specify the number of Gaussians in the GMM, as well as the method of incorporating spatial information: SPM – spatial pyramid pooling, AUG – spatial coordinate augmentation.

pipeline settings	PCA-SIFT, 64-D	PCA-SIFT, 128-D
GMM 256, SPM, intra-norm, FV	64.6 (262K)	65.0 (524K)
GMM 512, AUG, intra-norm, FV	65.3 (68K)	65.4 (133K)
GMM 512, AUG, intra-norm, hard-FV	65.1 (68K)	65.2 (133K)
GMM 1024, AUG, intra-norm, FV	66.1 (135K)	**66.5** (266K)
GMM 1024, AUG, intra-norm, hard-FV	66.0 (135K)	**66.2** (266K)

dense SIFT encodings. It is higher than 64.8% mAP reported by [Sánchez et al., 2012] for spatial augmentation and 2048 Gaussians, which can be explained by the fact that we used intra-normalisation.

5.3 Local Descriptor Transformation for VLAD

In the previous section, we PCA-rotated SIFT before the VLAD encoding, since PCA tends to improve the performance of the image retrieval methods [Jégou and Chum, 2012, Delhumeau et al., 2013]. However, as we will demonstrate in this section, PCA is not helpful when VLAD is used for classification, and the classification results can be improved by using more appropriate transformations. First, we show that an unsupervised whitening transform of local features significantly improves the performance (Sect. 5.3.1). Then, we propose a formulation for discriminative learning of local feature transforms (Sect. 5.3.2).

5.3.1 Unsupervised Whitening

Here we show that whitening of local SIFT features is beneficial for VLAD classification. Linear whitening transformations have been discussed in Sect. 2.3.1. As noted, PCA-whitened features are more suitable for discriminative classifier learning than

Table 5.3: **Image classification results (mAP, %) on VOC 2007 for different linear transformations of SIFT features.** In all experiments, the VLAD encoding was intra-normalised (Sect. 5.2).

transformation	mAP
none	61.2
PCA, 128-D	61.1
PCA, 64-D	61.1
PCA-whitening, 128-D	62.9
PCA-whitening, 64-D	**63.3**
ZCA, 128-D	**63.3**

just PCA-projected, since whitening equalises the relative importance of the feature vector components. Additionally, whitening transform can be advantageous for the k-means clustering (used in the VLAD codebook construction), since it removes the second order statistics of the data, which k-means can not exploit.

The results of different linear transforms are reported in Table 5.3. In all experiments, the transformed features were encoded using VLAD with intra-normalisation (Sect. 5.2). It is clear that both whitening transforms, PCA-whitening (2.8) and ZCA (2.9), lead to a significant ($> 2\%$) improvement on the PCA rotation and dimensionality reduction, as well as the "no transformation" setting. This indicates that local feature whitening is important for achieving higher classification accuracy.

At the same time, it should be noted that the VLAD encoding of whitened features, proposed here, is not necessarily applicable to the unsupervised image retrieval task. In that case, whitening can amplify the noise in the last principal components, and there is no discriminatively learnt (SVM) weighting vector to re-adjust the components' importance. We have also experimented with PCA-whitening of SIFT for FV encoding, and obtained worse results than with PCA. This can be explained by the fact that unlike k-means, GMM can handle different variances of the data, and the FV encoding effectively performs whitening internally (note the division by σ_k in (2.4)).

The comparison of the results of the improved VLAD (63.3% mAP with 512

words) and FV (65.0% mAP with 256 Gaussians) shows that VLAD is performing somewhat worse than FV for classification (SPM pooling used in both cases). In the next section we will show how the classification mAP gap between VLAD and FV can be reduced by discriminative learning.

5.3.2 Supervised Linear Transformation

Having demonstrated the importance of unsupervised whitening in the previous section, now we turn to the discriminatively trained local feature projections. Our aim is to learn a linear transformation W for local features $\mathbf{x}$, which improves the image classification based on the VLAD encoding of the transformed features $W\,\mathbf{x}$.

To learn W, we would like to formulate the objective function based on the multi-class classification constraints [Crammer and Singer, 2001]: for each image i, the classification score of the correct class $c(i)$ should be larger than the scores of the other classes c' by a unit margin:

$$v_{c(i)}^T \Phi_i > v_{c'}^T \Phi_i + 1 \quad \forall c' \neq c(i),\ \forall i, \tag{5.3}$$

where Φ_i is the VLAD representation of the image i, and v_c is a linear classifier of the class c. For brevity, we do not explicitly include the class-specific biases here, but they can be easily incorporated by concatenating the image descriptor Φ with a constant. Learning the linear transform W from the constraints (5.3) is challenging due to a complex dependency of Φ on W. To obtain a tractable optimisation problem, in the sequel we derive the "surrogate VLAD" representation, linear in W.

First, we modify the intra-normalised VLAD encoding by replacing the L^2 normalisation of the visual word slots with the normalisation by the number of features assigned to the corresponding visual word (refer to (2.1) and (2.3) in Sect. 2.2.2

for the VLAD formulation and notation). The modified VLAD encoding Φ of W-transformed local descriptors $\mathbf{x}_p$ then takes the following form:

$$\Phi = \left[\frac{1}{|\Omega_k|} \sum_{p \in \Omega_k} W\,\mathbf{x}_p - \mathbf{v}_k \right]_k , \tag{5.4}$$

where Ω_k is the set of indices of features, assigned to the k-th cluster $\mathbf{v}_k$, and $[\dots]_k$ is the stacking operator, which concatenates the sums of displacements across all clusters k.

It should be noted that in (5.4) the visual words $\mathbf{v}_k$ are obtained by the k-means clustering of the transformed features $W\,\mathbf{x}$. This means that they are computed on the training set as

$$\mathbf{v}_k = \frac{1}{|\widetilde{\Omega_k}|} \sum_{q \in \widetilde{\Omega_k}} W\,\mathbf{x}_q, \tag{5.5}$$

where $\widetilde{\Omega_k}$ is the set of training set descriptors assigned to the cluster k (which is different from Ω_k – the set of the image descriptors assigned to k). Now, (5.4) can be re-written as follows:

$$\Phi = \left[W \left(\frac{1}{|\Omega_k|} \sum_{p \in \Omega_k} \mathbf{x}_p - \frac{1}{|\widetilde{\Omega_k}|} \sum_{q \in \widetilde{\Omega_k}} \mathbf{x}_q \right) \right]_k = \widehat{W}\widehat{\Phi} \tag{5.6}$$

where $\widehat{W}$ is a block-diagonal matrix, which contains K replications of W along its main diagonal – one for each cluster slot in VLAD:

$$\widehat{W} = \begin{bmatrix} W & & \\ & W & \\ & & W \end{bmatrix}, \tag{5.7}$$

and $\widehat{\Phi}$ can be seen as a VLAD-like image representation, corresponding to untrans-

formed descriptors:

$$\widehat{\Phi} = \left[\left(\frac{1}{|\Omega_k|} \sum_{p \in \Omega_k} \mathbf{x}_p - \frac{1}{|\widetilde{\Omega_k}|} \sum_{q \in \widetilde{\Omega_k}} \mathbf{x}_q \right) \right]_k \tag{5.8}$$

We note that the representation (5.6)–(5.8) is not yet linear in W due to the assignments of the transformed descriptors $W\,\mathbf{x}_p$ to clusters Ω_k being dependent on W. However, once we fix these assignments, the "surrogate" VLAD (5.6) becomes linear in W, which makes learning feasible.

Now that we have "linearised" VLAD with respect to the linear transform W (with visual word assignments fixed), we can set up a learning framework, which alternates between learning W, given the assignments, and updating the assignments, given a new W. The large-margin objective, based on the constraints (5.3), takes the following form:

$$\sum_i \sum_{c' \neq c(i)} \max \left\{ \left(v_{c'} - v_{c(i)} \right)^T \widehat{W} \widehat{\Phi}_i + 1, 0 \right\} + \frac{\lambda}{2} \sum_c \|v_c\|_2^2 + \frac{\mu}{2} \|W\|_2^2 \tag{5.9}$$

Given the visual word assignments, it is biconvex in linear transformation W and classifiers v_c. This means that a local optimum of (5.9) can be found by performing another alternation between the convex learning of W (given v_c) and the convex learning of v_c (given W). This is similar to the WSABIE projection learning formulation [Weston et al., 2010]. It should be noted that after updating the visual word assignments, there is no guarantee that the objective will not increase, so in general there is no convergence guarantee for our optimisation procedure. In practice, the optimisation is performed until the performance on the validation set stops improving. After the optimisation is finished, the classifiers v_c are discarded, and only the linear transformation W is kept.

Table 5.4: **Image classification results (mAP, %) on VOC 2007 for the VLAD encoding of learnt and unsupervised linear transformations of SIFT features.** For all experiments, VLAD was computed with intra-normalisation and spatial pyramid pooling (Sect. 5.2).

transformation	64-D	128-D
whitening (unsupervised)	63.3	63.3
learnt	**64.4**	**64.6**

Evaluation. To train the SIFT transform using the formulation (5.8), we used a separate image set – a subset of the ImageNet ILSVRC-2010 dataset [Berg et al., 2010], which contains 200 randomly selected classes (out of 1000 in the full set). The use of the different, larger, set for training W allowed us to avoid over-fitting and assess the generalisation ability of the learnt model (since the sets of image classes are different). The learning was initialised by setting the feature transform W to PCA-whitening. Once W is learnt, we proceed with the standard evaluation pipeline (Sect. 5.1). The results of the learnt SIFT transformations to 128-D (no dimensionality reduction) and 64-D spaces are shown in Table 5.4. As can be seen, the learnt transformations outperform unsupervised whitening (Sect. 5.3.1). Namely, the VLAD encoding of discriminatively transformed SIFT features achieves 64.4% and 64.6% mAP using 64-D and 128-D representations respectively. This is comparable with the results of the intra-normalised FV encoding with SPM pooling, which achieves 64.6% and 65.0% respectively (Sect. 5.2). In spite of the slightly worse results, the VLAD representation is generally faster to compute than the FV coding.

5.4 Conclusion

In this chapter, we have proposed and evaluated a number of improvements for VLAD and FV feature encodings. In particular, intra-normalisation [Arandjelović and Zisserman, 2013] was shown to consistently improve the classification perfor-

mance of both VLAD and FV on the VOC 2007 dataset, while feature whitening turned out to be helpful for VLAD. The conclusions regarding the performance of FV encoding and its modifications will be exploited in the following sections. Namely, in Chapter 6 we will use the FV encoding of spatially augmented PCA-SIFT features to derive a discriminative human face representation. The hard-assignment version of the FV encoding will be used in the deep encoding framework of Sect. 7, where using conventional FVs is computationally intractable.

It should be noted, however, that computer vision datasets tend to have specific biases, caused by the way they are collected [Torralba and Efros, 2011]. While intra-normalisation of Fisher vectors is helpful on VOC 2007 dataset, it did not bring any consistent performance improvement on the tasks, discussed in Chapters 6 and 7, so we used the conventional signed square-rooting there. The explanation for such a behaviour could be that in VOC 2007, the objects, corresponding to the image category label, often occupy a small area of the image. In other words, only a subset of dense local features covers the object. In that case, the negative effect of the local feature burstiness [Jégou et al., 2009] is more pronounced, making intra-normalisation beneficial.

Chapter 6

Compact Discriminative Face Representations

In this chapter we address the problem of discriminative face image representation. In particular, we are interested in designing a face descriptor, suitable for recognition tasks, e.g. face verification (Sect. 6.1). To this end, we adopt an off-the-shelf image descriptor based on the Fisher Vector (FV) encoding of dense SIFT features [Perronnin et al., 2010]. The Fisher vector is then subjected to discriminative dimensionality reduction (Sect. 6.2). The resulting representation, termed Fisher Vector Face (FVF) descriptor (Sect. 6.3), is compact and discriminative. As will be shown in Sect. 6.4, it achieves state-of-the-art accuracy, performing on par or better than hand-crafted face representations.

6.1 Introduction

In this section, we set up the face verification problem and review the related work on face representations. The face verification problem is defined as follows: given a pair of face images, one needs to determine if both images portray the same person.

Figure 6.1: **Various face landmark configurations.** Designing an appropriate configuration is a challenging problem, which might require a significant amount of hand-crafting. The figure was taken from [Chen et al., 2013].

A typical face verification system is built on several key components, such as: face extraction, discriminative face description, and a distance (or similarity) function. We discuss them in more detail below.

The face extraction stage can be seen as pre-processing. Given an image containing a face, it localises the face (face detection) and then, optionally, maps it to a pre-defined coordinate frame (face alignment). Face detection is typically carried out with the face detector of Viola and Jones [2001]. Face alignment consists in transforming the face images so that the same spatial location in different images (roughly) corresponds to the same point of the face. This can be done, for example, by detecting a set of face-specific salient points (known as face landmarks) and mapping them to the pre-defined locations in a canonical (reference) frame. Examples of face landmark configurations are shown in Fig. 6.1. For instance, Everingham et al. [2009] proposed to detect nine landmarks (corners of eyes, mouth, and nose) using pictorial structures and map them to the canonical frame in the least-squares sense using an affine transform. A more complicated landmark detection scheme, proposed by Belhumeur et al. [2011], uses annotated face images as exemplars, which define the prior on the landmark location. It is then combined with the results of independent landmark detectors to obtain 29 landmarks. An extension of this alignment technique was used by Berg and Belhumeur [2012], where 95 landmarks were

detected, divided into inner and outer points. Another family of alignment methods (called "funnelling") was developed by Huang et al. [2007a, 2012b]. In their case, they perform a sequence of transformations, which maximises the likelihood of each pixel under a pixel-specific generative model. In other words, the algorithm tries to align all face pixels, not just the landmarks. The alignment step can also be omitted so that the face, cropped from the Viola-Jones bounding box, is directly passed to the face descriptor.

In this work, our main focus is on face description and distance function learning. As noted in the literature review (Sect. 2.2.1), conventional face descriptors are usually domain-specific and are based on the stacking of multiple local descriptors, such as LBP [Wolf et al., 2008, Chen et al., 2013], SIFT [Guillaumin et al., 2009], or both [Taigman et al., 2009, Wolf et al., 2009, Li et al., 2012]. Due to the stacking-based descriptor aggregation, the number of local features is limited. Therefore, the local descriptors are either computed over a sparse regular grid [Wolf et al., 2008, 2009, Taigman et al., 2009], or around sparse facial landmarks [Everingham et al., 2009, Guillaumin et al., 2009, Chen et al., 2013]. In the former case, the stacked representation is not invariant to face deformations due to the fixed location of the grid. Computing local descriptors around landmarks can alleviate this problem (if the landmarks are reliably detected), since the location of the landmark changes together with the face pose. An example of landmark-based descriptor is the method of Everingham et al. [2006], where a configuration of nine landmarks was detected using pictorial structures, and then described using a normalised intensity descriptor. In [Guillaumin et al., 2009], the 128-D SIFT descriptors were computed at three scales around these landmarks, leading to $3 \times 9 \times 128 = 3456$ face representation. This approach was taken to the extreme by Chen et al. [2013], who used a state-of-the-art face landmark detector [Cao et al., 2012] to detect 27 landmarks. After that, a local LBP descriptor [Ahonen et al., 2006] was densely extracted around

each of these landmarks, leading to 100K-dimensional face image descriptor. Other methods [Kumar et al., 2009, Berg and Belhumeur, 2012] describe the face in terms of its attributes (e.g. "has a moustache") and similarities to other faces. This is accomplished by training attribute-specific classifiers which, in turn, rely on the low-level representations, e.g. those based on landmarks, as described above.

It should be noted that the set of landmarks used for alignment is, in general, different from the set of landmarks used for descriptor sampling. For instance, in [Berg and Belhumeur, 2012], 95 landmarks were used for alignment, but only a subset of them – for sampling. On the contrary, in [Chen et al., 2013], only 5 landmarks were used for alignment, but 27 – for sampling. Using the landmarks to drive feature sampling means that a lot of hand-crafting should be put into the design of the landmark configuration (Fig. 6.1), since it is not immediately clear which landmarks are important for face description. Additionally, erroneous landmark detection can hamper the face descriptor computation.

To overcome the problems, associated with landmark-driven face sampling, we propose to compute local features (SIFT, in our case) densely in scale and space, and, instead of stacking, use Fisher Vector (FV) feature encoding (see review in Sect. 2.2.2) to aggregate a large number of local features. This lifts the limitation on the number of local features, and removes the dependency of the feature sampling on landmark detection. We should note that in some of the very recent works on face description a similar approach was employed, e.g. Sharma et al. [2012] used the Fisher vector encoding of local intensity differences, while in [Cui et al., 2013], the sparse coding of whitened intensity patches was used.

Given the descriptors of the two compared face images, face verification is carried out by computing the distance (or the similarity) between the face representations and comparing it to a threshold. The distance function can be unsupervised (e.g. Euclidean distance) or learnt (e.g. using one of the dimensionality reduction/distance

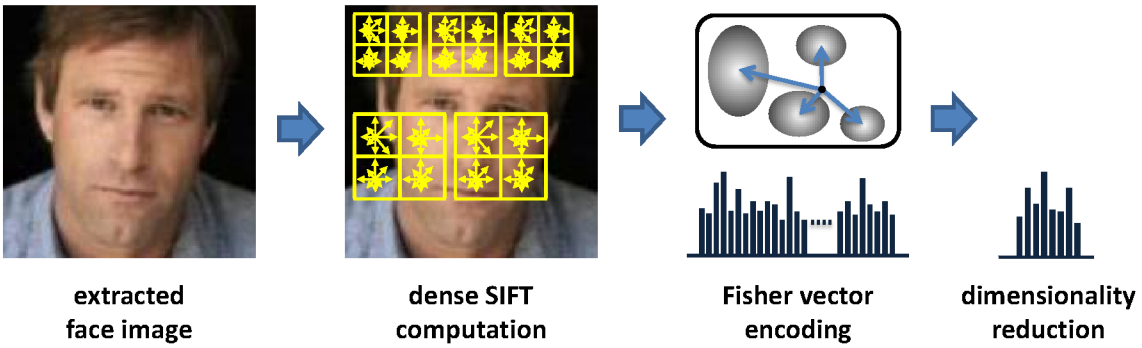


Figure 6.2: **Method overview:** a face is encoded in a discriminative compact representation.

learning formulations, described in Sect. 2.3). Another popular approach is based on exemplar SVMs [Wolf et al., 2008, Taigman et al., 2009, Wolf et al., 2009], where each of the compared images is first compared to a held-out reference set of faces, and then the results of those comparisons are aggregated to produce the verification score. We propose to project the high-dimensional FV face representation onto a low-dimensional subspace, which results in a compact and discriminative face descriptor (Fig. 6.2). The combination of dense SIFT, FV encoding, and linear large-margin models achieves a very competitive performance on a variety of generic image classification benchmarks [Perronnin et al., 2010, Chatfield et al., 2011, Sánchez and Perronnin, 2011]. Here we show that the face image domain is not an exception.

6.2 Large-Margin Dimensionality Reduction

In this section we explain how a high-dimensional FV encoding (Sect. 2.2.2) of a face image is compressed to a small discriminative representation. The compression is carried out using a linear projection, which serves two purposes: (i) it dramatically reduces the dimensionality of the face descriptors, making them applicable to large-scale datasets; and (ii) it improves the recognition performance by projection onto a subspace equipped with a discriminative Euclidean distance.

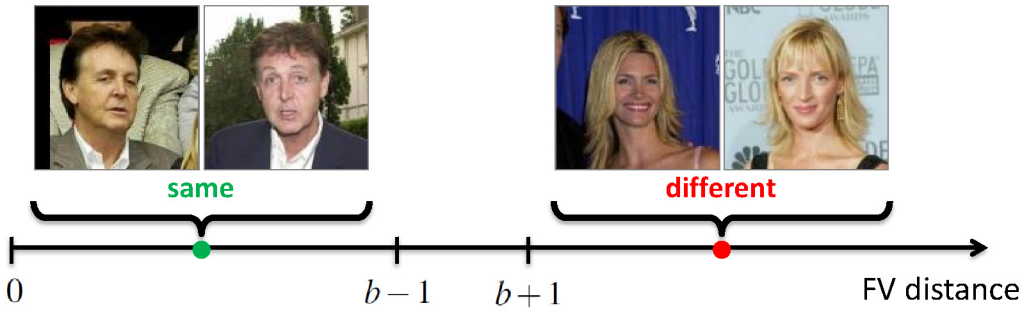


Figure 6.3: **Large-margin distance learning constraints:** the distance between images of the same person should be smaller than $b-1$, between images of different people – larger than $b+1$.

In more detail, the aim is to learn a linear projection $W \in \mathbb{R}^{m \times n}$, $m \ll n$, which projects high-dimensional Fisher vectors $\phi \in \mathbb{R}^n$ to low-dimensional vectors $W\phi \in \mathbb{R}^m$, such that the squared Euclidean distance $d^2_W(\phi_i, \phi_j) = \|W\phi_i - W\phi_j\|^2_2$ between images i and j is smaller than a learnt threshold $b \in \mathbb{R}$ if i and j are the images of the same person, and larger otherwise. Here we take "smaller" and "larger" in the large-margin sense, as illustrated in Fig. 6.3. Given the learnt W, at test time we will be able to classify an image pair by comparing the Euclidean distance between their W-projected Fisher vectors with the threshold b.

As discussed in Sect. 2.3.4 and 3.3, linear dimensionality reduction learning can be formulated as either low-rank Mahalanobis metric learning, or learning the projection directly. While the former leads to a convex objective (which we proposed for local descriptor dimensionality reduction in Sect. 2.3.4), it is only feasible if the original dimensionality n is moderate ($n \leq 10^3$). In the case of Fisher encodings, however, the dimensionality is larger: $n \sim O(10^4\text{–}10^5)$. This dictates the need to use non-convex optimisation over the projection W, similar to LMCA of Torresani et al. and LDML of Guillaumin et al. (see Sect. 2.3.4 for a review). Considering that our task is the classification of image pairs into "same person" and "different people",

we impose the classification constraints, giving the following optimisation problem:

$$\arg\min_{W,b} \sum_{i,j} \max\left\{1 - y_{ij}\left(b - (\phi_i - \phi_j)^T W^T W (\phi_i - \phi_j)\right), 0\right\}, \tag{6.1}$$

where $y_{ij} = 1$ iff images i and j contain the faces of the same person, and $y_{ij} = -1$ otherwise. The minimiser of (6.1) is found using a stochastic sub-gradient method. At each iteration t, the algorithm samples a single pair of face images (i, j) (sampling with equal frequency positive and negative labels y_{ij}) and performs the following update of the projection matrix:

$$W_{t+1} = \begin{cases} W_t & \text{if } y_{ij}\left(b - d^2_W(\phi_i, \phi_j)\right) > 1 \\ W_t - \gamma y_{ij} W_t (\phi_i - \phi_j)(\phi_i - \phi_j)^T & \text{otherwise} \end{cases} \tag{6.2}$$

where γ is a constant learning rate, determined on the validation set. Note that the projection matrix W_t is left unchanged if the margin constraint is not violated, which speed-ups learning (due to the large size of W, performing matrix operations at each iteration is costly). We choose not to regularise W explicitly; rather, the algorithm stops after a fixed number of learning iterations (1M in our case).

Since the objective (6.1) is not convex in W, the initialisation is important. In practice, we initialise W with the PCA-whitening matrix (see (2.8) in Sect. 2.3.1). Compared to the standard PCA, the magnitude of the dominant eigenvalues is equalised, since the less frequent modes of variation can be amongst the most discriminative. It is important to note that PCA-whitening is only used to *initialise* the learning process, and the learnt metric substantially improves over its initialisation (Sect. 6.4). In particular, this is not the same as learning a metric on the low-dimensional data *after* PCA or PCA-whitening (p^2 parameters). Mahalanobis metric learning in a low-dimensional space has been done by [Guillaumin et al., 2009, Chen et al., 2013], but this is suboptimal as the first, unsupervised, dimensionality

reduction step may lose important discriminative information. Instead, we learn the projection W on the *original* descriptors ($pd \gg p^2$ parameters), which allows us to fully exploit the available supervision.

6.2.1 Joint Metric-Similarity Learning.

Recently, a "joint Bayesian" approach to face similarity learning has been employed in [Chen et al., 2012, 2013]. It effectively corresponds to joint learning of a low-rank Mahalanobis distance $d_W(\phi_i, \phi_j) = (\phi_i - \phi_j)^T W^T W(\phi_i - \phi_j)$ *and* a low-rank kernel (inner product) $s_V(\phi_i, \phi_j) = \phi_i^T V^T V \phi_j$ between face descriptors ϕ_i, ϕ_j. Then, the difference between the distance and the inner product $d_W(\phi_i, \phi_j) - s_V(\phi_i, \phi_j)$ can be used as a score function for face verification. We consider it as another option for comparing face descriptors, and incorporate joint metric-similarity learning into our large-margin learning formulation (6.1). The resulting formulation takes the following form:

$$\arg\min_{W,V,b} \sum_{i,j} \max\left\{1 - y_{ij}\left(b - \frac{1}{2}(\phi_i - \phi_j)^T W^T W(\phi_i - \phi_j) + \phi_i^T V^T V \phi_j\right), 0\right\}, \tag{6.3}$$

We added the $1/2$ multiplier for the brevity of the sub-gradient derivations below. In that case, we perform stochastic updates on both low-dimensional projections W (6.2) and V:

$$V_{t+1} = \begin{cases} V_t & \text{if } y_{ij}\left(b - \frac{1}{2}d_W^2(\phi_i, \phi_j) + d_V(\phi_i, \phi_j)\right) > 1 \\ V_t + \gamma y_{ij} V_t\left(\phi_i \phi_j^T + \phi_j \phi_i^T\right) & \text{otherwise} \end{cases} \tag{6.4}$$

It should be noted that when using this joint approach, each high-dimensional FV is compressed to two *different* low-dimensional representations $W\phi$ and $V\phi$.

6.3 Implementation Details

Face alignment. Our face descriptor does not require any particular type of face alignment, and, in principle, can be applied to unaligned faces as well. Unless otherwise noted, the face images were aligned using the method of Everingham et al. [2009], applied to faces detected by the Viola-Jones algorithm [Viola and Jones, 2001]. In this case, nine detected facial landmarks are mapped to the predefined locations in a canonical frames using a similarity transform. The descriptor is then computed on a 160×125 face region, cropped from the centre of the canonical frame. It should be noted that the landmarks are used solely for alignment, and not for descriptor computation.

Face descriptor computation. For dense SIFT computation and Fisher vector encoding, we utilised publicly available packages [Vedaldi and Fulkerson, 2010, Chatfield et al., 2011]. In more detail, SIFT was computed densely on 24×24 pixel patches with a stride of 1 or 2 pixels. The SIFT computation was performed over 5 scales, with a scaling factor of $\sqrt{2}$. As a result, each face was represented by $\sim$ 25K SIFT descriptors. After that, the SIFT features were passed through the explicit feature map of the Hellinger kernel, also known as rootSIFT [Arandjelović and Zisserman, 2012]. In the remainder of this chapter, we use the terms "SIFT" and "rootSIFT" interchangeably.

Fisher vector computation was carried out as described in Sect. 2.2.2; rootSIFT features were decorrelated using PCA (with dimensionality reduced to 64) and augmented with their spatial coordinates, resulting in a 66-D local region representation. The GMM codebook was computed on the training set using the Expectation-Maximisation (EM) algorithm. The resulting Gaussian mixture models the distribution of both appearance and location of local features (due to the spatial augmentation). We visualise the Gaussians in 6.4, where each Gaussian is shown as an

ellipse with the centre and radii set to the mean and variances of the Gaussian's spatial components. As can be seen, the Gaussians are sptailly distributed over the whole image plane. Given the GMM and rootSIFT features, we compute their (improved) Fisher vector encoding [Perronnin et al., 2010], followed by square-rooting and normalisation. In the case of 512 Gaussians in the GMM, this results in the 67584-D face representation.

Dimensionality reduction learning, described in Sect. 6.2, is implemented in MATLAB and takes a few hours to compute on a single CPU core. Given an aligned and cropped face image, our MATLAB implementation (speeded up with C++ MEX functions) takes 0.6s to compute the proposed face descriptor on a single core (in the case of 2 pixel SIFT density).

Horizontal flipping. Following [Huang et al., 2012a], we considered the augmentation of the test set by taking the horizontal reflections of the image pair. Given the two compared images, each of them is horizontally reflected (left-right flipping), and the distances between the four possible combinations of the original and reflected images are computed and averaged. This makes the verification procedure invariant to the horizontal reflection, which is important, since the compared images can contain faces with different orientation. An alternative approach would be to augment the training set, and incorporate the invariance through learning.

6.4 Experiments

6.4.1 Dataset and Evaluation Protocol

Our framework is evaluated on the popular "Labeled Faces in the Wild" dataset (LFW) [Huang et al., 2007b], which contains 13233 images of 5749 people, downloaded from the Web. This challenging, large-scale face image collection has become

the *de-facto* evaluation benchmark for face-verification systems, promoting the rapid development of new face representations. For evaluation, the data is divided into 10 disjoint splits, which contain different identities and come with a list of 600 pre-defined image pairs for evaluation (as well as training as explained below). Of these, 300 are "positive" pairs portraying the same person and the remaining 300 are "negative" pairs portraying different people. We follow the recommended evaluation procedure [Huang et al., 2007b] and measure the performance of our method by performing a 10 fold cross validation, training the model on 9 splits, and testing it on the remaining split. All aspects of our method that involve learning, including PCA projections for SIFT, Gaussian mixture models, and the discriminative Fisher vector projections, were trained independently for each fold.

Two evaluation measures are considered. The first one is the *Receiving Operating Characteristic Equal Error Rate* (ROC-EER), which is the accuracy at the ROC operating point where the false positive and false negative rates are equal [Guillaumin et al., 2009]. This measure reflects the quality of the *ranking*, obtained by scoring image pairs, and does not depend on the learnt bias. ROC-EER is used to compare the different stages of the proposed framework, since we found it to be more sensitive to the changes in the verification pipeline, compared to the classification accuracy. In order to allow a direct comparison with published results, however, our final classification performance is reported in terms of the classification accuracy (percentage of image pairs correctly classified) – in this case the bias is important.

The LFW benchmark specifies a number of evaluation protocols, two of which are considered here. In the "restricted setting", only the pre-defined image pairs for each of the splits (fixed by the LFW organisers) can be used for training. Instead, in the "unrestricted setting" one is given the identities of the people within each split and is allowed to form an arbitrary number, in practice much larger, of positive and negative training pairs.

6.4.2 Framework Parameters

First, we explore how the different parameters of the method affect its performance. The experiments were carried out in the unrestricted setting using unaligned LFW images and a simple alignment procedure described in Sect. 6.3. We explore the following settings: SIFT density (the step between the centres of two consecutive descriptors), the number of Gaussians in the GMM, the effect of spatial augmentation, dimensionality reduction, distance function, and horizontal flipping. The results of the comparison are given in Table 6.1. As can be seen, the performance increases with denser sampling and more clusters in the GMM. Spatial augmentation boosts the performance with only a moderate increase in dimensionality (caused by the addition of the (x, y) coordinates to 64-D PCA-SIFT). Our dimensionality reduction to 128-D achieves 528-fold compression and further improves the performance. We found that using projection to higher-dimensional spaces (e.g. 256-D) does not improve the performance, which can be caused by over-fitting.

As far as the choice of the FV distance function is concerned, a low-rank Mahalanobis metric outperforms both full-rank diagonal metric and unsupervised PCA-whitening, but is somewhat worse than the function obtained by the joint large-margin learning of the Mahalanobis metric and inner product. It should be noted that the latter comes at the cost of slower learning and the necessity to keep two projection matrices instead of one. Finally, using horizontal flipping consistently improves the performance. In terms or the ROC-EER measure, our best result is 93.13%.

6.4.3 Learnt Model Visualisation

Here we demonstrate that the learnt model can indeed capture face-specific features. To visualise the projection matrix W, we make use of the fact that each GMM

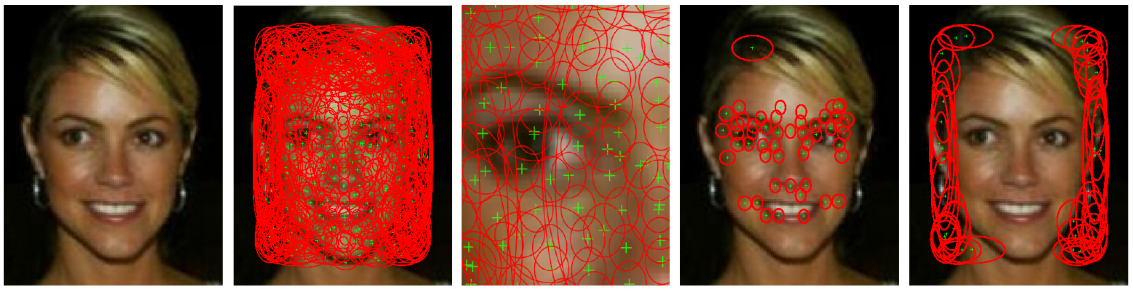
SIFT density	GMM Size	Spatial Aug.	Desc. Dim.	Distance Function	Hor. Flip.	ROC-EER,%
2 pix	256		32768	diag. metric		89.0
2 pix	256	✓	33792	diag. metric		89.8
2 pix	512	✓	67584	diag. metric		90.6
1 pix	512	✓	67584	diag. metric		90.9
1 pix	512	✓	128	low-rank PCA-whitening		78.6
1 pix	512	✓	128	low-rank Mah. metric		91.4
1 pix	512	✓	256	low-rank Mah. metric		91.0
1 pix	512	✓	128	low-rank Mah. metric	✓	92.0
1 pix	512	✓	2×128	low-rank joint metric-sim.		92.2
1 pix	512	✓	2×128	low-rank joint metric-sim.	✓	93.1

Table 6.1: **Framework parameters**: The effect of different FV computation parameters and distance functions on ROC-EER. All experiments done in the unrestricted setting.

component corresponds to a part of the Fisher vector and, in turn, to a group of columns in W. This makes it possible to evaluate how important certain Gaussians are for comparing human face images by computing the energy (Euclidean norm) of the corresponding column group. In Fig. 6.4 we show the GMM components which correspond to the groups of columns with the highest and lowest energy. As can be seen from Fig. 6.4-d, the 50 Gaussians corresponding to the columns with the highest energy match the facial features without being explicitly trained to do so. They have small spatial variances and are finely localised on the image plane. On the contrary, Fig. 6.4-e shows how the 50 Gaussians corresponding to the columns with the lowest energy cover the background areas. These clusters are deemed as the least meaningful by our projection learning; note that their spatial variances are large.

6.4.4 Effect of Face Alignment

It was mentioned above that our face descriptor does not depend on the facial landmarks for image sampling (since it uses dense sampling), so it can be coupled



(a) (b) (c) (d) (e)

Figure 6.4: **Coupled with discriminative dimensionality reduction, a Fisher vector can automatically capture the discriminative parts of the face.** (a): an aligned face image; (b): unsupervised GMM clusters densely span the face; (c): a close-up of a face part covered by the Gaussians; (d): 50 Gaussians corresponding to the learnt projection matrix columns with the highest energy; (e): 50 Gaussians corresponding to the learnt projection matrix columns with the lowest energy.

with any face alignment/extraction technique. In this section, we assess the effect of face alignment on performance. To this end, the descriptor settings are fixed, training is conducted in the LFW-unrestricted setting, and we only change the underlying face extraction method. In more detail, we compute spatially-augmented SIFT with 1 pixel step, use 512 Gaussians, FV dimensionality reduction to 128 dimensions (using the distance learning formulation), and horizontal flipping. As reported in Table 6.1, this gives 92.0% ROC-EER accuracy, when used with the original LFW images, aligned using the method of [Everingham et al., 2009].

Apart from the original (unaligned) images, the organisers of the LFW benchmark have also released several pre-aligned datasets. On the LFW-funneled dataset, obtained using the alignment technique of [Huang et al., 2007a], our descriptor achieves 91.7%. Following [Li et al., 2013], the descriptor was computed on the centred 150×150 crops. On a recently released LFW-deep-funneled dataset [Huang et al., 2012b], the performance is 92.0% (as before, we used centred 150×150 crops). As can be seen, the performance is very close for all three alignment techniques, even though the alignment algorithms are very different.

Finally, we consider the use case, where there is no face alignment at all, and the compressed Fisher vector representation is computed directly on the face detected by the Viola-Jones method. The face verification performance is then 90.9%, which is competitive with respect to the best results obtained with aligned images (92.0%). This demonstrates that our face representation is robust enough to deal with unaligned face images. It should be noted though, that this conclusion might not be applicable to other datasets with more extreme face variation (LFW is frontal-view only).

6.4.5 Comparison with the State of the Art

Unrestricted setting. In this scenario, we compare against the best published results obtained using both single (Table 6.2, bottom) and multi-descriptor representations (Table 6.2, top). Similarly to the previous section, the experiments were carried out using unaligned LFW images, processed as described in Sect. 6.3. This means that the outside training data is only utilised in the form of a simple landmark detector, trained by [Everingham et al., 2009].

Our method achieves 93.03% face verification accuracy, closely matching the state-of-the-art method of [Chen et al., 2013], which achieves 93.18% using LBP features sampled around 27 landmarks. It should be noted that (i) the best result of [Chen et al., 2013] using SIFT descriptors is 91.77%; (ii) we do not rely on multiple landmark detection, but sample the features densely. The ROC curves of our method as well as the other methods are shown in Fig. 6.5.

Restricted setting. In this strict setting, no outside training data is used, even for the landmark detection. Following [Li et al., 2013], we used centred 150×150 crops of the pre-aligned LFW-funneled images. We found that the limited amount of training data, available in this setting, is insufficient for dimensionality reduction

Method	Mean Acc.
LDML-MkNN [Guillaumin et al., 2009]	0.8750 ± 0.0040
Combined multishot [Taigman et al., 2009]	0.8950 ± 0.0051
Combined PLDA [Li et al., 2012]	0.9007 ± 0.0051
face.com [Taigman and Wolf, 2011]	0.9130 ± 0.0030
CMD + SLBP [Huang et al., 2012a]	0.9258 ± 0.0136
LBP multishot [Taigman et al., 2009]	0.8517 ± 0.0061
LBP PLDA [Li et al., 2012]	0.8733 ± 0.0055
SLBP [Huang et al., 2012a]	0.9000 ± 0.0133
CMD [Huang et al., 2012a]	0.9170 ± 0.0110
High-dim SIFT [Chen et al., 2013]	0.9177 ± N/A
High-dim LBP [Chen et al., 2013]	0.9318 ± 0.0107
Our Method	**0.9303 ± 0.0105**

Table 6.2: **Face verification accuracy in the unrestricted setting**. Using a single type of local features (dense SIFT), our method outperforms a number of methods, based on multiple feature types, and closely matches the state-of-the-art results of [Chen et al., 2013].

learning. Therefore, we trained a weighted Euclidean (diagonal Mahalanobis) metric on the full-dimensional Fisher vectors, which incurs learning an n-dimensional weight vector instead of a $m \times n$ projection matrix. It was carried out using a convex linear SVM formulation, where features are the vectors of squared differences between the corresponding components of the two compared FVs. We did not observe any improvement by enforcing the positivity of the learnt weights, so it was omitted in practice (i.e. the learnt function is not strictly a metric).

Achieving the verification accuracy of 87.47%, our descriptor sets a new state of the art in the restricted setting (Table 6.3), outperforming the recently published result of [Li et al., 2013] by 3.4%. It should be noted that while [Li et al., 2013] also use GMMs for dense feature clustering, they do not utilise the compressed Fisher vector encoding, but keep all extracted features for matching, which imposes a limitation on the number of features that can be extracted and stored. In our case, we are free from this limitation, since the dimensionality of an FV does not depend on the number of features it encodes. The best result of [Li et al., 2013]

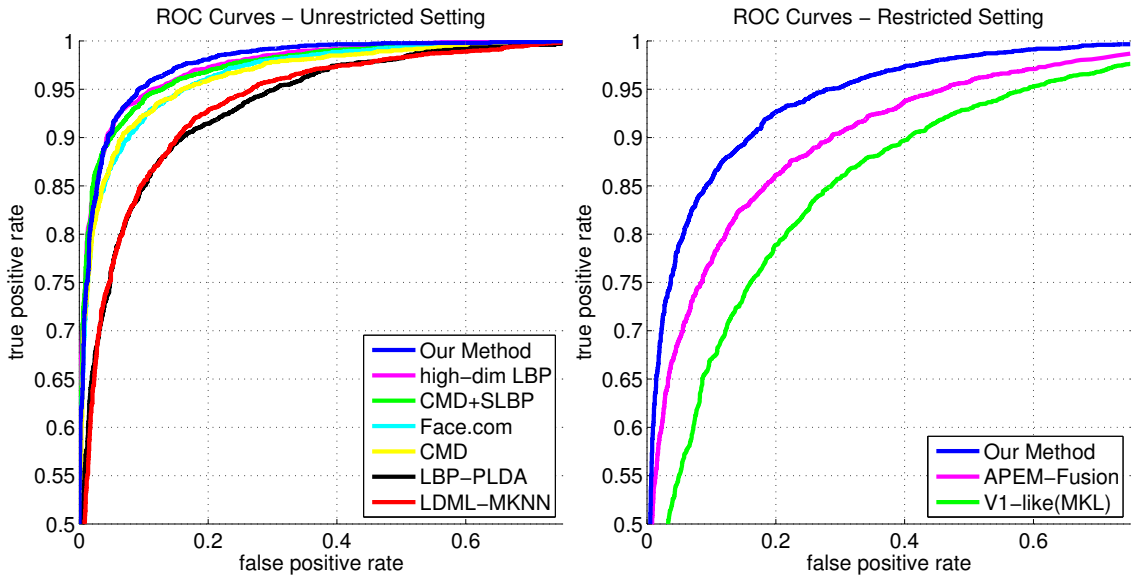


Figure 6.5: **Comparison with the state of the art:** ROC curves of our method (plotted in blue) and the state-of-the-art techniques in LFW-unrestricted (left) and LFW-restricted (right) settings.

Method	Mean Acc.
V1-like/MKL [Pinto et al., 2009]	0.7935 ± 0.0055
PEM SIFT [Li et al., 2013]	0.8138 ± 0.0098
APEM Fusion [Li et al., 2013]	0.8408 ± 0.0120
Our Method	**0.8747 ± 0.0149**

Table 6.3: **Right: Face verification accuracy in the restricted setting (no outside training data).** Our method achieves the new state of the art in this strict setting.

was obtained using two types of features and GMM adaptation ("APEM Fusion"). When using non-adapted GMMs (as we do) and SIFT descriptors ("PEM SIFT"), their result is 6% worse than ours.

Our results in both unrestricted and restricted settings confirm that the proposed face descriptor can be used in both small-scale and large-scale learning scenarios, and is robust with respect to the face alignment and cropping technique.

6.5 Conclusion

In this chapter, we have shown that an off-the-shelf image representation based on dense SIFT features and Fisher vector encoding achieves state-of-the-art performance on the challenging "Labeled Faces in the Wild" dataset (in spite of being based on a single feature type). The use of dense features allowed us to avoid applying a large number of sophisticated face landmark detectors. Also, we have presented a large-margin dimensionality reduction framework, well suited for high-dimensional Fisher vector representations. As a result, we obtain an effective and efficient face descriptor computation pipeline, which can be readily applied to large-scale face image repositories.

Chapter 7

Learning Deep Image Representations

In the previous chapters we explored the Fisher vector encoding in terms of both application areas and potential extensions. Namely, we proposed several improvements for VLAD and FV encodings in Chapter 5, and successfully applied FV encoding of dense SIFT features to the face recognition task in Chapter 6. However, in both cases the image classification pipeline remained rather shallow. That is, the local features (e.g. SIFT) were encoded with the Fisher vector representation, which was then used as a feature vector for classification with linear SVMs. In this chapter, we increase the depth of the Fisher vector pipeline, bridging the gap between the conventional classification frameworks and the deep neural networks (reviewed in Sect. 2.2.3). This allows us to explore how far we can get in terms of performance, when using off-the-shelf image representations, organised into a deeper framework. To this end we make the following contributions: (i) we introduce a *Fisher Vector Layer*, which is a generalization of the standard FV to a level architecture suitable for stacking; (ii) we demonstrate that by stacking and discriminatively training several such layers, a competitive performance (with respect to a deep convolutional

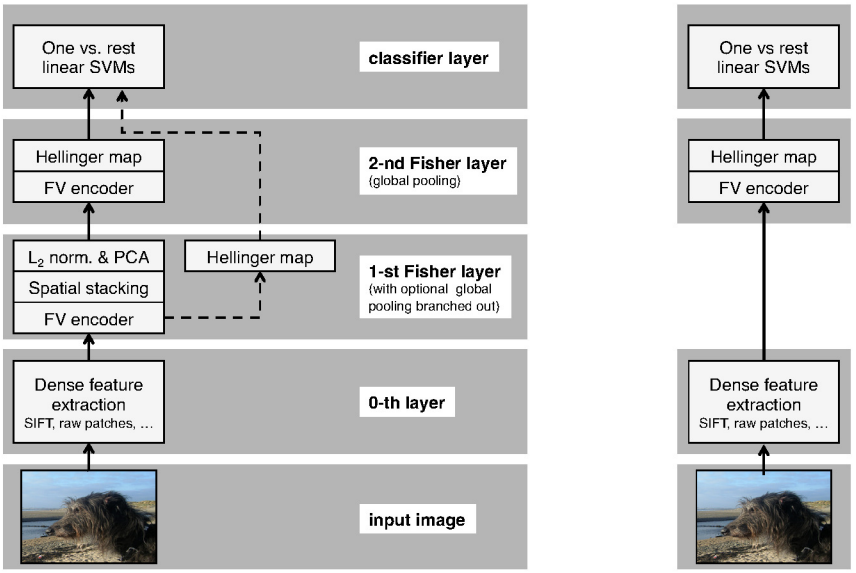


Figure 7.1: **Left: Fisher network (Sect. 7.2) with two Fisher layers. Right: conventional pipeline using a shallow Fisher vector encoding.** As shown in Sect. 7.4, making the conventional pipeline slightly deeper by injecting a single Fisher layer substantially improves the classification accuracy.

network [Krizhevsky et al., 2012]) can be achieved whilst staying in the realms of conventional SIFT and colour features and FV encodings; and (iii) we show that the linearity of the (unnormalised) Fisher encoding enables efficient earning and application of discriminative dimensionality reduction.

The rest of the chapter is organised as follows. In Sect. 7.1, we show how the Fisher vector representation can be modified to be used as a layer in a deeper architecture, and how the latter can be discriminatively learnt to yield a deep Fisher network (Sect. 7.2). After discussing important details of the implementation (Sect. 7.3), the improvements brought by stacking multiple Fisher layers are evaluated on the ImageNet image classification benchmark (Sect. 7.4).

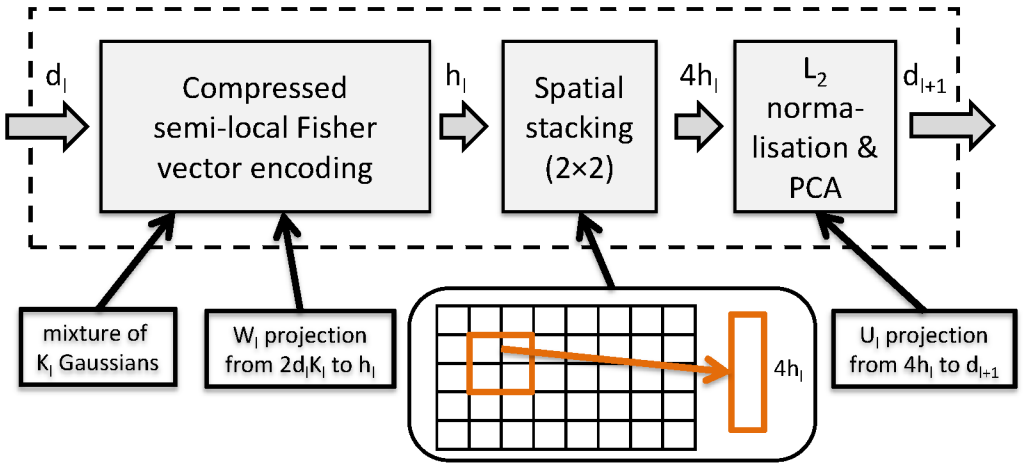


Figure 7.2: **The architecture of a single Fisher layer.** Top: the arrows illustrate the data flow through the layer; the dimensionality of *densely* computed features is shown next to the arrows. The layer is described in detail in Sect. 7.1.

7.1 Fisher Layer

7.1.1 Overview

The conventional FV representation of an image (Sect. 2.2.2), effectively encodes each local feature (e.g. SIFT) into a high-dimensional representation, and then aggregates these encodings into a single vector by global sum-pooling over the whole image (followed by normalisation). This means that the representation describes the image in terms of the local patch features, and can not capture more complex image structures. Deep belief networks are able to model the feature hierarchies by passing an output of one feature computation layer as the input to the next one. We adopt a similar approach here, and devise a feed-forward feature encoding layer (which we term a *Fisher layer*), which is based on off-the-shelf Fisher vector encoding. The layers can then be stacked into a deep network, which we call a *Fisher network*.

The architecture of the l-th Fisher layer is depicted in Fig. 7.2. On the input, it receives d_l-dimensional features ($d_l \sim 10^2$), densely computed on the regular image grid. The features are assumed to be decorrelated using PCA. The layer performs feed-forward feature transformation in three sub-layers. The first one computes

semi-local FV encodings of the spatial neighbourhood of each of the input features. As a result, the input features are "replaced" with more discriminative features, each of which encodes a larger image area.

The FV encoder (Sect. 2.2.2) uses a layer-specifc GMM with K_l components, so the dimensionality of each FV is $2K_l d_l$, which, considering that FVs are computed densely, might be too large for practical applications. Therefore, we decrease FV dimensionality by projection onto h_l-dimensional subspace using a *discriminatively trained* linear projection $W_l \in \mathbb{R}^{h_l \times 2K_l d_l}$. In practice, this is carried out using an efficient, specialised implementation of FV encoder, described in Sect. 7.3. In the second sub-layer, the spatially adjacent features are stacked in a 2×2 window, which produces $4h_l$-dimensional dense feature representation. Finally, the features are L^2-normalised and PCA-projected to d_{l+1}-dimensional subspace using the linear projection $U_l \in \mathbb{R}^{d_{l+1} \times 4h_l}$, and passed as the input to the $(l+1)$-th layer. The next section explains each sub-layer in more detail.

7.1.2 Sub-layer Details

Multi-scale Fisher vector pooling (sub-layer 1). The key idea behind our layer design is to aggregate the FVs of individual features over a semi-local spatial neighbourhood, rather than globally or over a large spatial pyramid cell (as it is done in the conventional setting [Perronnin et al., 2010]). As a result, instead of a single FV, describing the whole image, the image is represented by a large number of densely computed semi-local FVs, each of which describes a spatially adjacent set of local features, computed by the previous layer. Thus, the new feature representation can capture more complex image statistics with larger spatial support. We note that due to additivity, computing the FV of a spatial neighbourhood corresponds to the sum-pooling over the neighbourhood, a stage widely used in DBNs. However, unlike many DBN architectures, which use a single pooling window size per layer,

we employ multiple pooling window sizes, so that a single layer can encode multi-scale statistics. The pooling window size of layer l is denoted as q_l, and the stride as δ_l. In Sect. 7.4 we show that multi-scale pooling indeed brings an improvement, compared to a fixed pooling window size.

The high dimensionality of Fisher vectors, however, brings up the computational complexity issue, as storing and processing thousands of dense FVs per image (each of which is $2K_l d_l$-dimensional) is prohibitive at large scale. We tackle this problem by employing discriminative dimensionality reduction for high-dimensional FVs, which makes the layer learning procedure *supervised*. The dimensionality reduction is carried out using a linear projection onto an h_l-dimensional subspace. As will be shown in Sect. 7.3, dense, compressed FVs can be computed very efficiently, without the need to compute the full-dimensional FVs first, and then project them down.

A similar approach (passing the output of a feature encoder to another encoder) has been previously employed by [Agarwal and Triggs, 2006, Coates et al., 2011, Yan et al., 2012], but in their case they used bag-of-words or sparse coding representations. As noted in [Coates et al., 2011], such encodings require large codebooks to produce a discriminative feature representations. This, in turn, makes these approaches hardly applicable to the datasets of ImageNet scale [Berg et al., 2010]. As explained in Sect. 2.2.2, FV encoders do not require large codebooks, and by employing supervised dimensionality reduction, we can preserve the discrimativeness of FVs even after the projection onto a low-dimensional space, similarly to [Gordo et al., 2012].

Spatial stacking (sub-layer 2). After the dimensionality-reduced FV pooling (Sect. 7.1.2), an image is represented as a spatially dense set of relatively low-dimensional discriminative features ($h_l = 10^3$ in our experiments). It should be noted that local sum-pooling, while making the representation invariant to small

translations, is agnostic to the relative location of aggregated features. To capture the spatial structure within each feature's neighbourhood, we incorporate the stacking sub-layer, which concatenates the spatially adjacent features in a 2×2 window. This step is similar to 4×4 stacking employed in SIFT.

Normalisation and PCA projection (sub-layer 3). After stacking, the features are L^2 normalised, which improves their invariance properties. This procedure is closely related to Local Contrast Normalisation, widely used in DBNs. Finally, before passing the features to the FV encoder of the next layer, PCA dimensionality reduction is carried out, which serves two purposes: (i) features are decorrelated so that they can be modelled using diagonal-covariance GMMs of the next layer; (ii) dimensionality is reduced from $4h_l$ to d_{l+1} to keep the image representation compact and the computational complexity limited.

7.2 Fisher Network

7.2.1 Architecture

Our image classification pipeline, which we coin *Fisher network* (shown in Fig. 7.1) is constructed by stacking several (at least one) Fisher layers (Sect. 7.1) on top of dense features, such as SIFT or raw image patches. The penultimate layer, which computes a single-vector image representation, is the special case of the Fisher layer, where sum-pooling is only performed globally over the whole image. We call this layer the *global* Fisher layer, and it effectively computes a full-dimensional normalised Fisher vector encoding (the dimensionality reduction stage is omitted since the computed FV is directly used for classification). The final layer is an off-the-shelf ensemble of one-vs-rest binary linear SVMs. As can be seen, a Fisher network generalises the standard FV pipeline of [Perronnin et al., 2010], as the latter corresponds to the

network with a single global Fisher layer.

Multi-layer image descriptor. Each subsequent Fisher layer is designed to capture more complex, higher-level image statistics, but a very competitive performance of shallow FV-based frameworks [Perronnin et al., 2012] suggests that low-level SIFT features are already discriminative enough to distinguish between a number of image classes. To fully exploit the hierarchy of Fisher layers, we branch out a globally pooled, normalised FV from each of the Fisher layers, not just the last one. These image representations are then concatenated to produce a rich, multi-layer image descriptor. A similar approach has previously been applied to convolutional networks by [Sermanet and LeCun, 2011].

7.2.2 Learning

The Fisher network is trained in a supervised manner, since each Fisher layer (apart from the global layer) depends on discriminative dimensionality reduction. The network is trained greedily, layer by layer. Here we discuss how the (non-global) Fisher layer can be efficiently trained in the large-scale scenario, and introduce two options for the projection learning objective.

Projection learning proxy. As explained in Sect. 7.1.2, we need to learn a discriminative projection W onto a low-dimensional space for high-dimensional FV encodings, sum-pooled over semi-local image areas. To do so, we ideally need a class label for each area, but the only available annotation in our case is a class label for each image. This defines a weakly supervised learning problem, and one way of solving it would be to assign the image label to all its semi-local areas. This, however, is not feasible at large scale (with $\sim 10^6$ training images), since the number of densely sampled areas is large ($\sim 10^4$ per image). Sampling a small number (e.g.

one) of semi-local FVs per image does not guarantee that the object, corresponding to the image label, will be covered by the sampled FVs, so using image annotation is unreliable in this case.

Therefore, we construct a learning proxy by computing the average Φ of all unnormalised semi-local FVs ϕ_s of an image, $\Phi = \frac{1}{S}\sum_{s=1}^{S}\phi_s$, and defining the learning constraints on Φ. The image label is used as the label of the average FV. Considering that $W\Phi = \frac{1}{S}\sum_{s=1}^{S} W\phi_s$, the projection W, learnt for Φ, is also applicable to individual semi-local FVs ϕ_s. The advantages of the proxy are that the image-level class annotation can now be utilised, and during projection learning we only need to store a single vector Φ per image. In the sequel, we define two options for the projection learning objective, which are then compared in Sect. 7.4.

Bi-convex max-margin projection learning. One approach to discriminative dimensionality reduction learning consists in finding the projection onto a subspace, where the image classes are as linearly separable as possible [Weston et al., 2011, Gordo et al., 2012]. This corresponds to the bilinear class scoring function: $v_c^T W\Phi$, where W is the linear projection which we seek to optimise and v_c is the linear model (e.g. an SVM) of the class c in the projected space. The max-margin optimisation problem for W and the ensemble $\{v_c\}$ takes the following form:

$$\sum_i \sum_{c' \neq c(i)} \max\left[\left(v_{c'} - v_{c(i)}\right)^T W\Phi_i + 1, 0\right] + \frac{\lambda}{2}\sum_c \|v_c\|_2^2 + \frac{\mu}{2}\|W\|_F^2, \tag{7.1}$$

where c_i is the ground-truth class of an image i, λ and μ are the regularisation constants. The learning objective is bi-convex in W and v_c, and a local optimum can be found by alternation between the convex problems for W and $\{v_c\}$, both of which can be solved in primal using a stochastic sub-gradient method [Shalev-Shwartz et al., 2007]. We initialise the alternation by setting W to the PCA-whitening

matrix W_0. Once the optimisation has converged, the classifiers v_c are discarded, and we keep the projection W.

Projection onto the space of classifier scores. Another dimensionality reduction technique, which we consider in this work, is to train one-vs-rest SVM classifier $\{u_c\}_{c=1}^{C}$ on the full-dimensional FVs Φ, and then use the C-dimensional vector of SVM outputs as the compressed representation of Φ. This corresponds to setting the c-th row of the projection matrix W to the SVM model u_c. This approach is closely related to attribute-based representations and classemes [Lampert et al., 2009, Torresani et al., 2010], but in our case we do not use any additional data annotated with a different set of (attribute) classes to train the models; instead, the $C = 1000$ classifiers trained directly on the ILSVRC dataset are used. If a specific target dimensionality is required, PCA dimensionality reduction can be further applied to the classifier scores [Gordo et al., 2012], but in our case we applied PCA after spatial stacking (Sect. 7.1.2).

The advantage of using SVM models for dimensionality reduction is, mostly, computational. As we will show in Sect. 7.4, both formulations exhibit a similar level of performance, but training C one-vs-rest classifiers is much faster than performing alternation between SVM learning and projection learning in (7.1). The reason is that one-vs-rest SVM training can be easily parallelised, while projection learning is significantly slower even when using a parallel gradient descent implementation.

7.3 Implementation Details

Hard-assignment Fisher vector. To facilitate an efficient computation of a large number of dense FVs per image, we utilise hard-assignment FV encoding (hard-FV), introduced in Sect. 5.2.1. The encoding of a single feature is based on its assignment to the Gaussian, which best explains the feature. The resulting hard-

FV is inherently sparse; this allows for the fast computation of the projection of the sum of FVs: $W_l \sum_{\mathbf{x}} \phi(\mathbf{x})$. Indeed, it is easy to show that

$$W_l \sum_{\mathbf{x}} \phi(\mathbf{x}) = \sum_{k=1}^{K} \sum_{\mathbf{x} \in \Omega_k} \left(W^{(k,1)} \phi_k^{(1)}(\mathbf{x}) + W^{(k,2)} \phi_k^{(2)}(\mathbf{x}) \right), \tag{7.2}$$

where Ω_k is the set of encoded features, hard-assigned to the GMM component k, and $W^{(k,1)}, W^{(k,2)}$ are the sub-matrices of W_l, which correspond to the 1st and 2nd order statistics $\phi_k^{(1),(2)}(\mathbf{x})$ of feature $\mathbf{x}$ with respect to the k-th Gaussian (2.4). This suggests the fast computation procedure: each d_l-dimensional input feature $\mathbf{x}$ is first hard-assigned to a Gaussian k based on (5.2). Then, the corresponding d_l-D differences $\phi_k^{(1),(2)}(\mathbf{x})$ are computed and projected using small $h_l \times d_l$ sub-matrices $W^{(k,1)}, W^{(k,2)}$, which is fast. The algorithm avoids computing high-dimensional FVs, followed by the projection using a large matrix $W_l \in \mathbb{R}^{h_l \times 2K_l d_l}$, which is prohibitive since the number of dense FVs is high.

Implementation. We implemented our framework in Matlab with certain parts of the code in C++ MEX. The computation is carried out on CPU without the use of GPU (our pipeline would potentially benefit from a GPU implementation). Training the Fisher network on top of SIFT descriptors on 1.2M images of ILSVRC-2010 [Berg et al., 2010] dataset takes about one day on a 200-core cluster. Image classification time is $\sim$ 2s on a single core.

Feature extraction. Our feature extraction follows that of [Perronnin et al., 2012]. Images are rescaled so that the number of pixels is 100K. Dense SIFT is computed on 24×24 patches over 5 scales (scale factor $\sqrt[3]{2}$) with the 3 pixel step. We also employ SIFT augmentation with the patch spatial coordinates [Sánchez et al., 2012]. During training, high-dimensional FVs, computed by the 2nd Fisher layer, are compressed using product quantisation [Sánchez and Perronnin, 2011].

7.4 Evaluation

In this section, we evaluate the proposed Fisher network on the large-scale image classification benchmark, introduced for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2010 [Berg et al., 2010]. The dataset contains images of 1000 categories, with 1.2M images available for training, 50K for validation, and 150K for testing. Following the standard evaluation protocol for the dataset, we report both top-1 and top-5 accuracy (%) computed on the test set. Top-1 is the proportion of images that are correctly classified; top-5 relaxes this notion by allowing five guesses per image. Sect. 7.4.1 evaluates the variants of the Fisher network on a subset of ILSVRC to identify the best one. Then, Sect. 7.4.2 evaluates the complete framework.

7.4.1 Fisher Network Variants

We begin with comparing the performance of the Fisher network under different settings. The comparison is carried out on a subset of ILSVRC, which was obtained by random sampling of 200 classes out of 1000. To avoid over-fitting indirectly on the test set, comparisons in this section are carried on the validation set. In our experiments, we used SIFT as the first layer of the network, followed by two Fisher layers (the second one is global, as explained in Sect. 7.2.1).

Dimensionality reduction, stacking, and normalisation. Here we quantitatively assess the three sub-layers of a Fisher layer (Sect. 7.1). We compare the two proposed dimensionality reduction learning schemes (bi-convex learning and classifier scores), and also demonstrate the importance of spatial stacking and L^2 normalisation. The results are shown in Table 7.1. As can be seen, both spatial stacking and L^2 normalisation improve the performance, and dimensionality reduction via projection onto the space of SVM classifier scores performs on par with the

Table 7.1: **Evaluation of dimensionality reduction, stacking, and normalisation sub-layers on the subset of ILSVRC-2010.** The following configuration of Fisher layers was used: $d_1 = 128$, $K_1 = 256$, $q_1 = 5$, $\delta_1 = 1$, $h_1 = 200$ (number of classes), $d_2 = 200$, $K_2 = 256$. The baseline performance of a shallow FV encoding is 57.03% and 78.9% (top-1 and top-5 accuracy).

dim-ty reduction	**stacking**	**L2 norm-n**	**top-1**	**top-5**
classifier scores		✓	59.69	80.29
classifier scores	✓		59.42	80.44
classifier scores	✓	✓	**60.22**	80.93
bi-convex	✓	✓	59.49	**81.11**

projection learnt using the bi-convex formulation (7.1). In the following experiments we used the classifier scores for dimensionality reduction, since their training can be parallelised and is significantly faster.

Multi-scale pooling and multi-layer image representation. In this experiment, we compare the performance of semi-local FV pooling using single and multiple window sizes (Sect. 7.1), as well as single- and multi-layer image representations (Sect. 7.2.1). From Table 7.2 it is clear that using multiple pooling window sizes is beneficial compared to a single window size. When using multi-scale pooling, the pooling stride was increased to keep the number of pooled semi-local FVs roughly the same. Also, the multi-layer image descriptor obtained by stacking globally pooled and normalised FVs, computed by the two Fisher layers, outperforms each of these FVs taken separately. We also note that in this experiment, unlike the previous one, both Fisher layers utilized spatial coordinate augmentation of the input features, which leads to a noticeable boost in the shallow baseline performance (from 78.9% to 80.50% top-5 accuracy).

7.4.2 Evaluation on ILSVRC-2010

Now that we have evaluated various Fisher layer configurations on a subset of ILSVRC, we assess the performance of our framework on the full ILSVRC-2010

Table 7.2: **Evaluation of multi-scale pooling and multi-layer image description on the subset of ILSVRC-2010.** The following configuration of Fisher layers was used: $d_1 = 128$, $K_1 = 256$, $h_1 = 200$, $d_2 = 200$, $K_2 = 256$. Both Fisher layers used spatial coordinate augmentation. The baseline performance of a shallow FV encoding is 59.51% and 80.50% (top-1 and top-5 accuracy).

pooling window size q_1	**pooling stride δ_1**	**multi-layer**	**top-1**	**top-5**
5	1		61.56	82.21
$\{5, 7, 9, 11\}$	2		62.16	82.43
$\{5, 7, 9, 11\}$	2	✓	**63.79**	**83.73**

Table 7.3: **Performance on ILSVRC-2010 using dense SIFT and colour features.** We also specify the dimensionality of SIFT-based image representations. For reference, the top-1 and top-5 accuracies of the deep convolutional network [Krizhevsky et al., 2012] without test set augmentation are 61% and 81.7% respectively.

pipeline setting	SIFT only			SIFT & colour	
	dimension	**top-1**	**top-5**	**top-1**	**top-5**
1st Fisher layer	82K	45.79	68.25	54.53	75.79
2nd Fisher layer	131K	48.25	71.29	N/A	N/A
1st and 2nd Fisher layers	213K	**52.09**	**73.51**	**58.83**	**78.72**
Sánchez and Perronnin [2011]	524K	N/A	67.9	54.3	74.3

dataset. We use off-the-shelf SIFT and colour features [Perronnin et al., 2010] in the feature extraction layer, and demonstrate that significant improvements can be achieved by injecting a single Fisher layer into the conventional FV-based pipeline [Sánchez and Perronnin, 2011].

The following configuration of Fisher layers was used: $d_1 = 80$, $K_1 = 512$, $q_1 = \{5, 7, 9, 11\}$, $\delta_1 = 2$, $h_1 = 1000$, $d_2 = 256$, $K_2 = 256$. On both Fisher layers, we used spatial coordinate augmentation of the input features. The first Fisher layer uses a large number of GMM components K_l, since it was found to be beneficial for shallow FV encodings [Sánchez and Perronnin, 2011], used here as a baseline.

The results are shown in Table 7.3. First, we note that the globally pooled Fisher vector, branched out of the first Fisher layer (which effectively corresponds to the conventional FV encoding), results in better accuracy than reported in [Sánchez and Perronnin, 2011], which validates our implementation. Using the 2nd Fisher

layer on top of the 1st one leads to a significant performance improvement. Finally, stacking the FVs, produced by the 1st and 2nd Fisher layers, pushes the accuracy even further.

The state of the art on the ILSVRC-2010 dataset was obtained using an 8-layer convolutional network [Krizhevsky et al., 2012], i.e. twice as deep as the Fisher network considered here. Using training and test set augmentation (not employed here), they achieved 62.5% and 83.0% for top-1 and top-5 accuracy. Without test set augmentation, their result is 61% / 81.7% [Krizhevsky et al., 2012], while we get 58.8% / 78.7%. By comparison, the baseline shallow FV accuracy is 54.53% / 75.79%. We conclude that injecting a single intermediate layer induces a quite significant performance boost (+4.27% top-1 accuracy), but deep convolutional networks are still somewhat better (+2.2% top-1 accuracy). These results are however quite encouraging since they were obtained by using a standard off-the-shelf feature encoding reconfigured to add a single intermediate layer. Notably, the model did *not* require an optimised GPU implementation to be trained, nor it was necessary to control over fitting by techniques such as random drop-out [Krizhevsky et al., 2012].

7.5 Conclusion

We have shown that Fisher vectors, a standard image encoding method, are amenable to be stacked in multiple layers, in analogy to the state-of-the-art deep neural network architectures. Adding a single layer is in fact sufficient to significantly boost the performance of these shallow image encodings, bringing their performance closer to the state of the art in the large-scale classification scenario [Krizhevsky et al., 2012]. The fact, that off-the-shelf image representations can be simply and successfully stacked, indicates that deep schemes may extend well beyond neural networks.

Chapter 8

Medical Image Search Engine

This chapter addresses the problem of scalable, real-time medical image retrieval. In contrast to the previous chapters, which proposed discriminative *image representations*, here we discuss an *image repository representation*, tailored to medical image retrieval tasks. In particular, we are interested in designing a system, which allows a clinician to carry out a structured visual search in large medical repositories, i.e. query by a particular region of a medical image.

The rest of the chapter is organised as follows. We begin with introducing the problem of structured medical image retrieval in Sect. 8.1, where we also discuss the related work. After that, we propose a generic framework for medical image retrieval in Sect. 8.2, and introduce a scalable method for medical image registration (Sect. 8.3). We then consider two applications for the framework: retrieval of 2-D X-ray images (Sect. 8.4) and 3-D Magnetic Resonance Imaging (MRI) volumes (Sect. 8.5). We mention the implementation details in Sect. 8.6 and conclude the chapter in Sect. 8.7.

8.1 Introduction

The exponential growth of digital medical image repositories of recent years poses both challenges and opportunities. Medical centres now need efficient tools for analysing the plethora of patient images. At the same time, myriads of archived scans represent a huge source of data which, if exploited, can inform and improve current clinical practice. Medical images and corresponding clinical cases, stored in these large collections, capture a wide range of disease population variability due to numerous covariates (diagnosis, age, co-morbidities, etc). Instant image retrieval from such repositories could be of great value for clinical practice, e.g. by providing a "second opinion" based on the corresponding diagnostic information or course of treatment. Apart from the processing speed, another important aspect of a practical retrieval system is the ability to focus the search on a particular part (structure) of the image which is of most interest.

Here we present a scalable framework for the immediate retrieval of medical images and structures of interest within them ("structured search"). Given a query image (e.g. from a new patient) and a user-drawn Region Of Interest (ROI) in it, we seek to retrieve repository images with the corresponding ROI (e.g. the same bone in the hand) located. The returned images can then be ranked based on the contents of the ROI.

Why immediate structured image search? Given a patient with a condition (e.g. a tumour in the spine) retrieving other generic spine X-rays may not be as useful as returning images of patients with the same pathology, or of exactly the same vertebra. The structured search with an ROI is where we differ from conventional content-based medical image retrieval methods which return images that are *globally* similar to a query image [Müller et al., 2004]. The immediate aspect of our work enables a flexible exploration, as it is not necessary to specify in advance what region

(e.g. an organ or anomaly), to search for – every region is searchable.

Clinical applications. The use cases of structured medical image search include: conducting population studies on specific anatomical structures; tracking the evolution of anomalies efficiently; and finding similar anomalies or pathologies in a particular region. The ranking function can be modified to order the returned images according to the similarity between the query and target ROI's shape or image content. Alternatively, the ROI can be classified, e.g. on whether it contains a particular anomaly such as cysts on the kidney, or arthritis in bones, and ranked by the classification score.

8.1.1 Related Work

The problem of content-based medical image retrieval has a vast literature. Most conventional approaches [Müller et al., 2004] consist in retrieving images that are *globally* similar to the query image. Recently, the problem of ROI-level search has been addressed in [Lam et al., 2007, Avni et al., 2011, Burner et al., 2011]. These works describe retrieval systems, which can be queried by an ROI. However, the algorithm of [Avni et al., 2011] returns the repository images, similar to the query ROI, without detecting the corresponding ROI inside. In [Burner et al., 2011], the target ROI were restricted to super-pixels, i.e. over-segmentation of the target images. Similarly, in [Lam et al., 2007], the target ROIs were restricted to the lung nodules, pre-annotated by the experts.

Our approach is inspired by the image retrieval work of [Sivic and Zisserman, 2003, Philbin et al., 2007], who considered unconstrained ROI search in natural image datasets. However, the direct application of these techniques to medical images is not feasible (as shown in Sect. 8.4.4) because the feature matching and registration methods of these previous works do not account for inter-subject non-

rigid transformations and the repeating structures common to medical images (e.g. phalanx or spine bones). Instead, we employ non-rigid registration methods, well suited to medical images.

8.2 Structured Image Retrieval Framework

Our framework is based on the observation that medical images are obtained from a limited, standardised set of viewpoints. This makes it possible to split the medical image repository into a set of classes (depending on the modality, body part, viewpoint, etc., e.g. "X-ray images of hands, anterior view") and compute registrations between images of the same class. This can be done off-line, so that at run time the correspondences of a query ROI in target images can be obtained immediately.

To enable immediate ROI retrieval at run time, processing is divided into off-line and on-line parts, as summarised in Fig. 8.1. The **off-line part** consists in classifying the images and pre-computing the registrations between images of the same class. It should be noted that the registration can be performed using *any* off-the-shelf method suitable for a particular class of images. **At run time,** given the query image and ROI, three stages are involved. First, the class of the image is determined, so that the ROI correspondences are only considered between images of the same class (target images). Then, the corresponding ROI in the target images is found based on the pre-computed transformations. Finally, once the regions of interest have been localised in the target images, they can be ranked, e.g. based on an application-specific clinically relevant score. In the following sections, we will present two implementations of the framework, one operating on a multi-class dataset of 2-D X-ray images (Sect. 8.4), and another – on a single-class dataset of 3-D brain MRI scans (Sect. 8.5).

The on-line retrieval steps, mentioned above, are carried out differently, depend-

1. **On-line (given a user-specified query image and ROI bounding box)**
   - Select the target image set (repository images of the same class as the query).
   - Using the pre-computed registration and transform composition (Sect. 8.3), compute the ROIs corresponding to the query ROI in all images of the target set.
   - Rank the ROIs using the similarity measure of choice.
2. **Off-line (pre-processing)**
   - Classify the repository images into a set of pre-defined classes.
   - Compute the registration for all pairs of images of the same class. (Sect. 8.3).

Figure 8.1: **The on-line and off-line parts of the retrieval engine.**

ing on whether the query image is taken from the dataset. If it is, then the retrieval is instant: the class of the image is known, and the registrations are already computed. If the query image is not in the repository, it should be added there first, by classifying it and registering it with the repository images of the same class. This brings up the issue of computational efficiency in the case of large datasets. To alleviate this problem, we propose an exemplar-based registration technique, described next.

8.3 Exemplar-Based Registration

Carrying out non-rigid registration of the query image with each of the target images scales badly with the number of repository images, as non-rigid medical image registration is computationally complex, and the number of registrations equals the number of images. Moreover, storing all pairwise registrations is prohibitive due to high storage requirements of non-rigid transforms (e.g. B-spline warps computed over a dense 3-D grid).

The key idea behind scalable exemplar-based registration is that instead of reg-

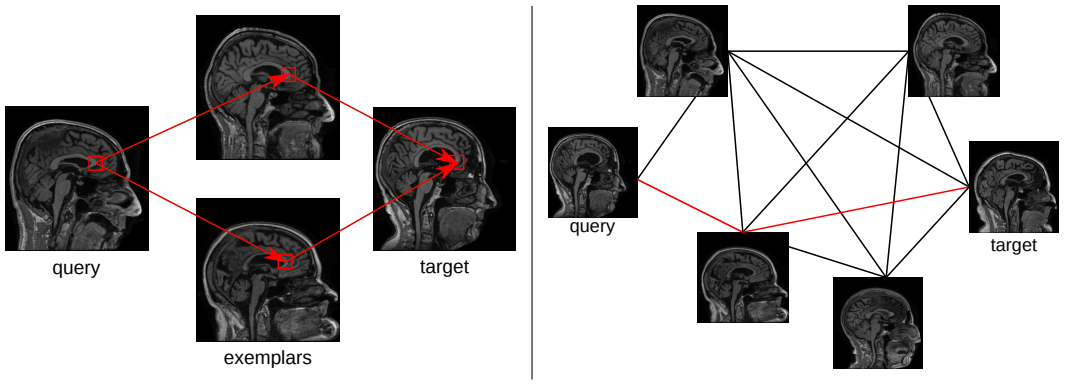


Figure 8.2: **Left: exemplar-based registration. Right: repository graph.** The red line illustrates the path from the query to the target through an exemplar image.

istering a query image with each of the repository images by pairwise registration, the query is registered with only a few fixed images (called *exemplars*), which effectively define several reference spaces. The remaining repository images will have already been pre-registered with exemplars, so they can be registered with the query by composing the two transforms. Finally, to obtain a single correspondence from several exemplars, the composed transforms are aggregated. The exemplar-based registration is schematically illustrated in Fig. 8.2 (left).

More formally, for a dataset of N images, a query image I_q is registered with only a subset of $K = \text{const}$ exemplar images, which results in K transforms $T_{q,k}$, $k = 1 \dots K$. The transformations $T_{k,t}$ between an exemplar I_k and each of the remaining repository images I_t are pre-computed. Then the transformation between images I_q and I_t can be obtained by composition of transforms (computed using different exemplars) followed by aggregation:

$$T_{q,t}(\mathbf{x}) = \text{agg}\left(\{T_{k,t} \circ T_{q,k}\}\right)(\mathbf{x}) \tag{8.1}$$

where $\mathbf{x}$ is a point in the query image and agg is the aggregation function.

The advantage of exemplar-based registration scheme is that for a query image

only $K \ll N$ registrations should be computed, and the transform composition complexity is negligible. Thus, pairwise registrations between all images can be computed in $O(KN)$ rather than $O(N^2)$. The same estimates apply to the storage requirements for the computed registrations, which allows them to be stored in RAM for fast access. Compared to the group-wise registration algorithms [Cootes et al., 2005], transform composition does not rely on the computation of a group mean model, and is scalable in the case of rapidly growing datasets. Additionally, the use of several transformations instead of one improves the registration robustness. The technique is related to the multi-atlas segmentation scheme of [Isgum et al., 2009], but here we use composition for registration.

8.3.1 Exemplar Selection and Aggregation

There are two choices to make in setting up the composition scheme (8.1): how to select the exemplars and how to define the function, aggregating the transforms obtained using different exemplars. One possibility is a non-deterministic scheme, where the exemplars are selected randomly, and the aggregation is performed by taking a coordinate-wise median. We use it in the implementation of Sect. 8.4. Another option is to select the exemplars and perform the aggregation based on the image registration accuracy. In this section, we describe deterministic ways of exemplar selection and transform aggregation, which will be compared in the context of the MRI retrieval framework of Sect. 8.5.

Exemplar images selection. The objective of exemplar selection is to pick a fixed number (K) of repository images, such that they can be accurately registered with the remaining ones. Let $\varepsilon_{ij} \in [0; 1]$ be the registration error between a pair of images (i, j), with 0 corresponding to a perfect registration. In general, the error can be computed using different cues, e.g. intensity, deformation field smoothness,

re-projection error, etc. In our experiments, we employed inverse normalised mutual information.

One way of selecting the exemplars is to pick K images, such that the sum of registration error between them and all other images is minimal. The set of exemplars is then obtained by ranking the images in the ascending order of $\sum_j \varepsilon_{ij}$ and then selecting the first-K images as exemplars. We call this technique "min-sum" selection.

Another approach is based on clustering the repository images into K clusters, followed by the selection of a single exemplar in each of these clusters. Using $1-\varepsilon_{ij}$ as the similarity between images i and j, we use the spectral clustering technique [Shi and Malik, 2000] to split the images into a set of clusters such that the similarity between images in different clusters is small, and the similarity between images in the same cluster is large. Once the images are divided into clusters, a single exemplar is selected in each of the clusters as the image with minimal sum of registration errors to the others.

Transform aggregation. Once the exemplars are selected and fixed, the way of aggregating several registrations into one should be defined (function agg in (8.1)). In general, taking the mean or median does not account for the exemplars registration error, which can be large for certain pairs of query and target images. One of the possible ways to account for these errors is to pick a single registration which corresponds to the shortest path in the graph from the query to the target vertices and goes through exactly one exemplar (Fig. 8.2, left). In other words, for a given (query, target) pair of images, only one exemplar is selected, which has the lowest

sum of registration errors with these images:

$$\mathrm{agg}(q,t)(\mathbf{x}) = \left(T_{s,t} \circ T_{q,s}\right)(\mathbf{x}), \tag{8.2}$$
$$s = \arg\min_k \varepsilon_{qk} + \varepsilon_{kt}$$

8.4 2-D X-ray Image Retrieval

In this section, we present an implementation of the real-time structured visual search framework, tailored to 2-D X-ray images. The implementation follows the generic architecture laid out in Sect. 8.2. In Sect. 8.4.1, we provide the details of the classification step. Then, Sect. 8.4.2 describes the non-rigid registration method, well suited to X-ray images. Section 8.4.3 gives examples of ROI ranking functions, and Sect. 8.4.4 assesses the retrieval performance.

Dataset. Our dataset is based on the publicly available IRMA collection of medical images [Deserno, 2009]. It contains X-ray images of five classes: `hand`, `spine`, `chest`, `cranium`, `background (the rest)`. Each class is represented by 205 images. The background class contains images of miscellaneous body parts, not included in the other classes. The images are stored in the PNG format without any additional textual metadata. Images within each class exhibit a high amount of variance, e.g. scale changes, missing parts, new objects added (overlaid writings), anatomy configuration changes (e.g. phalanges apart or close to each other). Each of the classes is randomly split into 65 testing, 70 training, and 70 validation images.

8.4.1 Image Classification

The aim of this step is to divide the X-ray images into the five classes. Certain image retrieval methods take the textual image annotation into account, which can

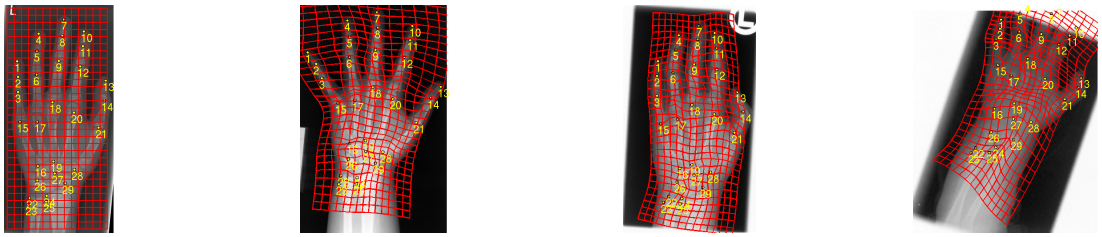
be available in the DICOM clinical meta-data. However, as shown in [Gueld et al., 2002], the error rate of the DICOM information is high, which makes it infeasible to rely on text annotation for classification. Therefore, we perform classification solely based on the visual cues.

We employ the multiple kernel (MKL) method of [Varma and Ray, 2007, Vedaldi et al., 2009] and train a set of binary SVM classifiers on multi-scale dense-SIFT and self-similarity visual features in the "one-vs-rest" manner. The MKL formulation can exploit different, complementary image representations, leading to high-accuracy classification, which was measured to be 98%. The few misclassifications are caused by the overlap between the `background` class and other classes, which can happen if the background image partially contains the same body part.

8.4.2 Robust Non-Rigid Registration

In this section, we describe the non-rigid registration algorithm for a pair of 2-D images. This algorithm is the basic workhorse that is used to compute registrations between all X-ray images of the same class. In our case, the registration method should be robust to a number of intraclass variabilities of our dataset (e.g. child vs adult hands) as well as additions and deletions (such as overlaid writing, or the wrists not being included). At the same time, it should be reasonably efficient to allow for the fast addition of a new image to the dataset.

The method, adopted here, is a sequence of robust estimations based on sparse feature point matching. The process is initialized by a coarse registration based on matching the first and second order moments of the detected feature points distribution. This step is feasible since the pairs of images to be registered belong to the same class and similar patterns of detected points can be expected. Given this initial transform T_0, the algorithm then alternates between feature matching (guided by the current transform) and Thin-Plate Spline (TPS) transform estimation (using



(a) (b) (c) (d)

Figure 8.3: **Robust thin plate spline matching.** (a): query image with a rectangular grid and a set of ground-truth (GT) landmarks (shown with yellow numbers); (b)-(d): target images showing the GT points mapped via the *automatically* computed transform (GT points not used) and the induced grid deformation.

the current feature matches). This approach is related to [Chui and Rangarajan, 2003]. We differ in that we perform feature matching based on visual descriptors (rather than just spatial coordinates), and the Thin Plate Spline (TPS) transform estimation is carried out using robust RANSAC procedure. The feature matching and transform estimation stages are described next.

Guided feature matching. We use Harris feature regions (Sect. 2.1.1), and the neighbourhood of each point is described by a SIFT descriptor [Lowe, 2004]. Feature matching is carried out as follows. Let I_q and I_t be two images to register and T_k the current transform estimate between I_q and I_t. The subscripts i and j indicate matching features in images I_q and I_t with locations $\mathbf{x}_i$, $\mathbf{y}_j$ and descriptor vectors Ψ_i and Ψ_j respectively. Feature point matching is formulated as a linear assignment problem with unary costs $\mathtt{C}_{ij}$ defined as:

$$\mathtt{C}_{ij} = \begin{cases} +\infty & \text{if } \mathtt{C}_{ij}^{geom} > R \\ w^{desc}\,\mathtt{C}_{ij}^{desc} + w^{geom}\,\mathtt{C}_{ij}^{geom} & \text{otherwise.} \end{cases} \tag{8.3}$$

It depends on the descriptors distance $\mathtt{C}_{ij}^{desc} = \|\Psi_i - \Psi_j\|_2$ as well as the symmetric transfer error $\mathtt{C}_{ij}^{geom} = \|T_k(\mathbf{x}_i) - \mathbf{y}_j\|_2 + \|\mathbf{x}_i - T_k^{-1}(\mathbf{y}_j)\|_2$. The hard threshold R on $\mathtt{C}_{ij}^{geom}$ allows matching only within a spatial neighbourhood of a feature. This

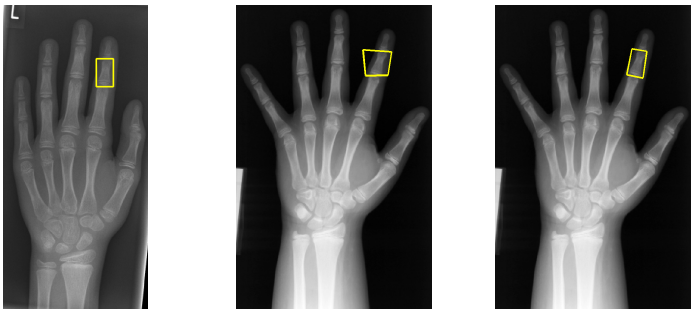
increases matching robustness, while reducing computational complexity.

Robust thin plate spline estimation. Direct TPS computation based on all feature point matches computed at the previous step leads to inaccuracies due to occasional mismatches. To filter them out we employ the LO-RANSAC [Chum et al., 2004] framework. In our implementation, two transformation models of different complexity are utilised for hypothesis testing. A similarity transform with a *loose* threshold is used for fast initial outlier rejection, while a TPS is fitted only to the inliers of the few promising hypotheses. The resulting TPS warp T_{k+1} is the one with the most inliers. The examples of the computed registrations are visualised in Fig. 8.3.

ROI localisation refinement. Given an ROI in the query image, we wish to obtain the corresponding ROI in the target image, i.e. the ROI covering the same "object". The TPS transform T, registering the query and target images, provides a rough estimate of the target ROI as a quadrilateral R_t^0 which is a warp of the query rectangle R_q. However, possible inaccuracies in T may cause R_t^0 to be misaligned with the actual ROI, and in turn this may hamper ROI ranking. To alleviate this problem, the detected ROI can be adjusted by locally maximizing the normalised intensity cross-correlation between the query rectangle and the target quadrilateral. This task is formulated as a constrained non-linear least squares problem where each vertex is restricted to a box to avoid degeneracies. An example is shown in Fig. 8.4.

8.4.3 ROI Ranking Functions

At this stage we have obtained ROIs in a set of target images, corresponding to the ROI in the query image. The question then remains of how to order the images for the retrieval system, and this is application dependent. We consider three



(a) (b) (c)

Figure 8.4: **ROI refinement**. (a): query; (b): target ROI before the local refinement; (c): target ROI after the local refinement.

choices of the ranking function defined as the similarity $S(I_q, R_q, I_t, R_t)$ between the query and target ROIs, R_q, R_t and images I_q, I_t. The retrieval results are ranked in decreasing order of S. The similarity S can be defined to depend on the ROI Appearance ($ROIA$) only. For instance, the normalised cross-correlation (NCC) of ROI intensities can be used. The S function can be readily extended to accommodate the ROI Shape ($ROISA$) as $S = (1-w)\min(E_q, E_t)/\max(E_q, E_t) + w\,\mathrm{NCC}(R_q, R_t)$, where E_q and E_t are elongation coefficients (ratio of major to minor axis) of query and target ROIs, and $w \in [0, 1]$ is a user tunable parameter. At the other extreme, the function S can be tuned to capture global Image Geometry (IG) cues. If similar scale scans are of interest, then S can be defined as: $S(I_q, R_q, I_t, R_t) = (1-w)\min\{\Sigma, 1/\Sigma\} + w\,\mathrm{NCC}(R_q, R_t)$, where $\Sigma > 0$ is the scale of the similarity transform computed from feature point matches, and $w \in [0, 1]$ is a user tunable parameter.

Fig. 8.5 shows the top ranked images retrieved by these functions. This is an example of how local ROI cues can be employed for ranking, which is not possible with global, image-level visual search. In clinical practice, ranking functions specifically tuned for a particular application could be used, e.g. trained to rank on the presence of a specific anomaly (such as nodules or cysts).

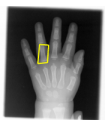
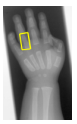
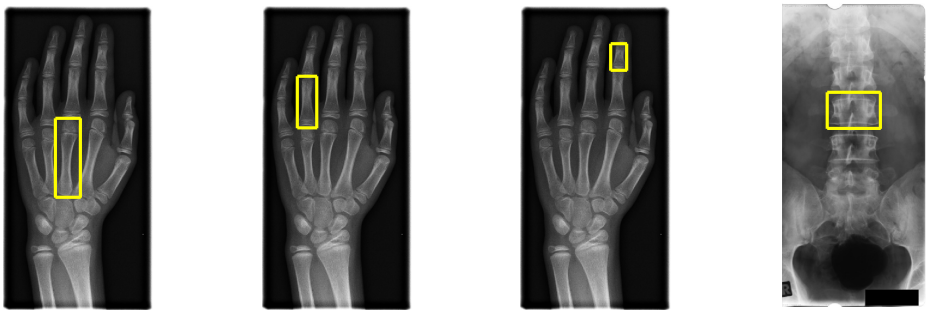
Query image and ROI	Ranking function	Top-5 retrieved images with detected ROI				
	IG ($w = 0.5$)					
	ROISA ($w = 0.5$)					
	ROIA					

Figure 8.5: **The effect of different ranking functions on ROI retrieval.** ROIs are shown in yellow. IG retrieves scans with similar image cropping; ROISA ranks paediatric hands high because the query is paediatric; ROIA ranks based on ROI intensity similarity.

8.4.4 Evaluation

Accuracy of structured image retrieval. To evaluate the accuracy of ROI retrieval from the dataset, we annotated test hand and spine images with axis-aligned bounding boxes around the same bones, as shown in Fig. 8.6. The ROI retrieval evaluation procedure is based on that of PASCAL VOC detection challenge [Everingham et al., 2010]. A query image and ROI are selected from the test set and the corresponding ROIs are retrieved from the rest of the test set using the proposed algorithm. A detected ROI quadrangle is labelled as correct if the overlap ratio between its axis-aligned bounding box and the ground truth one is above a threshold. The retrieval performance for a query is assessed using the Average Precision (AP) measure computed as the area under the "precision vs recall" curve. Once the retrieval performance is estimated for each of the images as a query, its mean (meanAP) and median (medAP) over all queries are taken as measures. We compare the retrieval performance of the framework (ROIA ranking, no ROI refinement) using different registration methods: the proposed one (Sect. 8.3), baseline feature



hand1 hand2 hand3 spine1

Figure 8.6: **Four annotated bones used for the retrieval performance assessment.**

Table 8.1: **Comparison of X-ray image retrieval accuracy.**

Method	hand1		hand2		hand3		spine1	
	meanAP	medAP	meanAP	medAP	meanAP	medAP	meanAP	medAP
Proposed	**0.81**	**0.89**	**0.85**	**0.90**	**0.65**	**0.71**	**0.49**	**0.51**
Baseline	0.68	0.71	0.66	0.71	0.38	0.36	0.35	0.35
`elastix`	0.62	0.67	0.61	0.68	0.38	0.37	0.22	0.19

matching with affine transform [Philbin et al., 2007], and `elastix` B-splines [Klein et al., 2010]. All three methods compute pairwise registration (i.e. no exemplars).

The proposed algorithm outperforms the others on all types of queries (Table 8.1). As opposed to the baseline, our framework can capture non-rigid transforms; intensity-based non-rigid `elastix` registration is not robust enough to cope with the diverse test set. Compared to hand images, worse performance on the spine is caused by less consistent feature detections on cluttered images.

8.5 3-D MRI Image Retrieval

In the previous section, we applied the retrieval framework of Sect. 8.2 to the task of 2-D X-ray image retrieval. Here, we apply the same framework to a more computationally challenging task of 3-D MRI image retrieval. We also evaluate several exemplar selection and transform aggregation methods, described in Sect. 8.3.1.

Dataset and applications. MRI data has been shown to provide reliable quantification of the atrophy process in the brain caused by Alzheimer's disease (AD) [Jack et al., 2004] or other neurodegenerative disorders. There are numerous natural history studies, the Alzheimer's Disease Neuroimaging Initiative (ADNI) [Mueller et al., 2005], launched in 2003, being the most prominent. Our dataset consists of 90 brain MRI scans randomly selected from the ADNI dataset [Mueller et al., 2005] (`http://www.loni.ucla.edu/ADNI/Data/`). The subset contains an equal number of images (30) of each of the three subject groups: Alzheimer's disease, control, and MCI (mild cognitive impairment).

Searching through brain MRI datasets on ROI level can be of interest to clinicians, since it can aid in differential diagnosis, as there are discriminating patterns between numerous forms of dementia. For example, the hippocampal deterioration is increasingly being considered as a way of identifying subjects who have a higher risk of developing AD. Providing the images with relevant ROI and their respective diagnosis to clinicians will aid in their decision process.

Registration and ranking. In the case of MRI data, we set-up the framework Sect. 8.2 using off-the-shelf algorithms. First of all, we should note that in this case there is no need to perform the image classification step, since all images are MRI images of human brain, taken with the same field of view. Thus, it is possible to establish correspondences between all of them, which was carried out using a non-rigid registration method, based on the Free-Form Deformations of Rueckert et al. [1999]. Briefly, it consists of a cubic B-Spline parametrisation model where the Normalised Mutual Information (NMI) is used as a measure of similarity. We used an efficient implementation [Modat et al., 2010] that is freely available as a part of the NiftyReg package. Our ranking function is the χ^2 distance between the brain tissue type distributions in the query and target ROI. The distributions were com-

puted using the GMM-based probabilistic segmentation algorithm [Cardoso et al., 2011].

8.5.1 Evaluation

In this section, we evaluate the registration accuracy of different combinations of exemplar selection and transform aggregation techniques, described in Sect. 8.3.1, as well as random exemplar selection and median aggregation, used in the implementation of 2-D search engine (Sect. 8.4). For exemplar selection, we consider random selection ("rand"), "min-sum" selection, and spectral clustering selection. For transform aggregation, "median", "mean", and the shortest path exemplar ("single") are compared.

The evaluation was performed on the brain MRI dataset (described above), which was randomly split into 45 training and 45 testing images. Exemplar selection was performed on the training set, registration evaluation – on the test set. The experiment was repeated three times. For the evaluation purposes, in each of these images we computed the "gold standard" segmentation into 83 brain anatomical structures using the method of [Cardoso et al., 2012].

For each pair of test images, the accuracy of registration was assessed using two criteria. First, we measured the mean distance (in mm) between points projected using pairwise (between query and target) and exemplar-based transformations. The measure describes how different exemplar-based registration is from the pairwise registration. The points were selected to be the centers of mass of the 83 anatomical structures. The second measure is the mean overlap ratio (Jaccard coefficient) of 83 anatomical structure bounding boxes, projected from the query image to the target image, with the bounding boxes in the target image. We used the bounding boxes of the anatomical structure volumes instead of the volumes themselves because it more closely follows the search engine use case scenario, where we operate on the

level of bounding boxes. We note that this measure is noisy due to the possible inaccuracies of the "gold standard" segmentation.

In Table 8.2 we report the mean and standard deviation of the two measures across all test image pairs for different number K of exemplar images. Based on the presented results, we can conclude that all three exemplar selection methods (including the random choice) exhibit similar levels of performance when coupled with robust median aggregation. Aggregation based on the shortest path selection performs worse, and the mean aggregation is the worst. The reason for such a behaviour could be that the global registration error, which we used for exemplar selection, does not account for the local inaccuracies. Another reason for similar performance can be the lack of strong image variation in our dataset. At the same time, using a single exemplar ($K = 1$) results in worse accuracy compared to several exemplar images. The accuracy of exemplar-based registration with median aggregation is at the same level as that of pairwise registration without exemplars. The average distance between the points projected using the two registrations is less than 1.4 mm.

Considering its low computational complexity, in our practical implementation we used the randomised selection of $K = 5$ exemplars and the median aggregation of the composed transforms. The average ROI registration time in this case is 0.06s per image (on a single CPU core), which allows for the fast retrieval when the system is rolled out on a multi-core server. Additional implementation details are presented next.

8.6 Implementation Details

In Sect. 8.4 and 8.5 we presented two ROI retrieval systems, based on the generic framework of Sect. 8.2. Both systems are implemented as Web-based applications,

Table 8.2: **Exemplar-based registration accuracy.** The overlap ratio of pairwise registration (without exemplars) is 0.568 ± 0.076. For the overlap ratio, higher is better; for the distance, smaller means closer to the direct registration without exemplars.

exemplar selection	aggregation function	overlap ratio			distance (mm)		
		$K = 1$	$K = 5$	$K = 7$	$K = 1$	$K = 5$	$K = 7$
rand	mean	0.555 ±0.072	0.532 ± 0.073	0.53 ± 0.073	2.04 ±0.28	1.44 ± 0.22	1.38 ± 0.21
	median		0.569 ± 0.076	$\mathbf{0.571 \pm 0.076}$		1.45 ± 0.23	1.37 ± 0.23
	single		0.557 ± 0.073	0.559 ± 0.073		1.99 ± 0.26	1.98 ± 0.25
min-sum	mean	0.557 ±0.072	0.531 ± 0.072	0.529 ± 0.072	1.94 ±0.26	1.42 ± 0.22	1.37 ± 0.22
	median		0.569 ± 0.076	0.57 ± 0.076		1.43 ± 0.23	1.36 ± 0.23
	single		0.558 ± 0.072	0.556 ± 0.072		1.94 ± 0.26	2.00 ± 0.32
cluster	mean		0.531 ± 0.072	0.529 ± 0.072		1.44 ± 0.22	1.39 ± 0.22
	median		0.569 ± 0.076	0.57 ± 0.076		1.45 ± 0.23	1.38 ± 0.23
	single		0.556 ± 0.072	0.556 ± 0.072		2.03 ± 0.32	2.03 ± 0.31

which can be accessed from any device, equipped with a Web browser (the "thin client" paradigm). In terms of the implementation, a retrieval system is split into a front-end and a back-end. The front-end, implemented in Python and JavaScript, allows a user to select a query image (or volume in the case of 3-D data), specify arbitrary axis-aligned ROI in it, and explore the retrieval results. The screenshots of the front-end of our 2-D (Sect. 8.4) and 3-D (Sect. 8.5) retrieval systems are shown in Fig. 8.7 (left and right, respectively). In certain use cases, using multiple query ROI can be beneficial, as it would allow one to select several relevant areas in a query image. Here we consider a single query ROI, but the extension to multiple ROI is rather straightforward. The back-end of the 2-D retrieval engine is implemented in Matlab, while the 3-D engine is implemented in Python. Both backends are fast enough to ensure immediate retrieval from our datasets, but could potentially benefit from a more optimised implementation.

8.7 Conclusion

In this chapter, we presented a practical structured image search framework, capable of instant retrieval of medical images (both 2-D and 3-D) and corresponding regions of interest from large datasets. Fast ROI alignment in repository images was made possible by representing the repository by non-rigid transformations between exemplar images and all other images. The advantage is that once the query image is registered with the exemplars, the exemplar-based representation allows for immediate ROI localisation using the transform composition technique.

It was shown that random exemplar image selection, coupled with robust median transform aggregation, achieves registration accuracy on par with pairwise registration without exemplars. The framework is fairly generic and can be extended to different modalities/dimensionalities with a proper choice of intra-class registration methods. Web-based demos of 2-D and 3-D ROI retrieval frameworks are available at http://www.robots.ox.ac.uk/~vgg/research/med_search/.

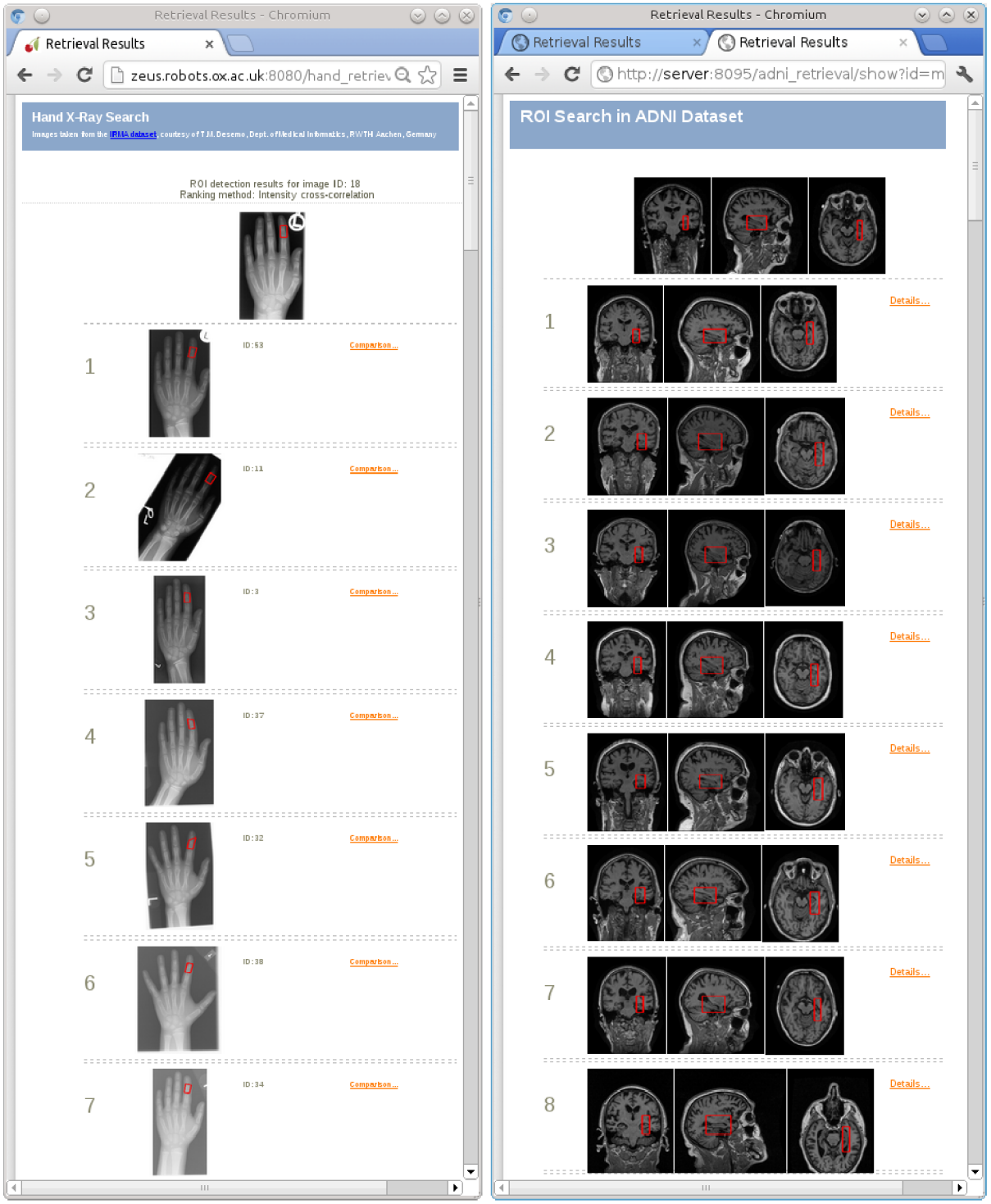


Figure 8.7: **Our Web-based medical image retrieval systems. Left:** hand X-ray retrieval; **right:** brain MRI retrieval. On the top of the page, the query image and ROI are shown, with the retrieval results below. The query and the retrieved ROI are shown in red. The system is accessed via a conventional Web browser.

Chapter 9

Conclusion

In this thesis we discussed the design of discriminative image representations for a variety of computer vision applications. Our research focus was on setting the parameters of these representations using large-scale machine learning, rather than hand-crafting. In this chapter, we summarise the contributions and the key results reported in this thesis (Sect. 9.1) and outline the directions of the future research (Sect. 9.2).

9.1 Contributions and Results

Local descriptor learning framework. In Chapter 3 we presented novel convex learning formulations for descriptor pooling region selection and dimensionality reduction. The convexity was achieved by using convex distance learning constraints and regularisers: the L^1 vector norm and the nuclear (trace) matrix norm. The former enforces sparsity, performing pooling region selection, while the latter enforces the low rank, performing dimensionality reduction. The large-scale stochastic optimisation of the learning objectives was performed using the recent regularised dual averaging (RDA) method [Xiao, 2010]. We also showed that our learnt real-valued

descriptor is amenable to binarisation using the frame expansion technique [Jégou et al., 2012a]. The resulting real-valued and binary descriptors set the state of the art on the Local Image Patches dataset (the comparison is given in Tables 3.2 and 3.3).

Local descriptor learning from weak supervision. In Chapter 4 we adapted our local descriptor learning algorithm to the weakly supervised setting, where ground truth feature matches are not available. In that case, we modelled the matches using latent variables, which allowed us to derive a tractable optimisation problem without the need to pre-set the matches based on heuristics, as was done in the prior art [Philbin et al., 2010]. We evaluated our learnt descriptors on Oxford and Paris Buildings datasets, and showed that they outperform unsupervised baselines and the learning method of [Philbin et al., 2010]. It should be noted that the retrieval performance of our baseline is already strong, which was achieved by using the affine-adapted DoG detector [Lowe, 2004] with a large descriptor measurement size. The main results can be found in Table 4.1.

Improved Fisher vector and VLAD encodings for classification. In Chapter 5 we proposed several ways of improving FV and VLAD encodings for classification. First, we adopted the intra-normalisation scheme [Arandjelović and Zisserman, 2013] to the Fisher vector encoding, achieving state-of-the-art results on PASCAL VOC 2007 benchmark (among the methods using only SIFT features). Second, we introduced a hard-assignment version of FV encoding, which performs similarly to the original FV at the fraction of the computation cost. Third, we demonstrated the importance of local feature whitening for classification using VLAD. Finally, we proposed a method for discriminative learning of local feature projections for VLAD. The results of the FV encoding on VOC 2007 are reported in Table 5.2, the VLAD encoding – in Table 5.3.

Fisher vector face representation. In Chapter 6 we focused on a particular image category – human face images. Our main contribution there is the application of the generic Fisher vector encoding of dense SIFT features to face images. This is different from ad-hoc face representations, built on top of carefully engineered face landmark detectors. To decrease the high dimensionality of Fisher vectors, as well as improve their discriminative ability on the face verification task, we proposed a large-margin dimensionality reduction learning formulation. The result is two-fold: (i) our face descriptor is low-dimensional (128-D), so it can used for face representation in large face image repositories; (ii) the verification accuracy of the descriptor is on par or better than the state of the art – refer to Tables 6.2 and 6.3 and Fig. 6.5 for the comparison.

Deep Fisher network. In Chapter 7, we proposed a novel deep image representation, which consists of several layers of Fisher vector encodings, interleaved with discriminative dimensionality reduction. Our deep descriptor can be seen as the middle ground between the shallow FV encoding (which it generalises) and the multi-layer deep convolutional networks [Krizhevsky et al., 2012] (which require specialised GPU implementations and training data augmentation to avoid over-fitting). The classification results on the ImageNet ILSVRC-2010 dataset (Table 7.3) reflect this positioning: our deep representation outperforms FV encoding, but a more complex deep CNN performs even better. This fact, however, does not devalue our contribution, since we did not augment training and test sets, and the training was carried out using Matlab implementation on a CPU cluster in less than a day.

Medical image search engine architecture. In Chapter 8, we presented a generic architecture of a medical image search engine, which allows one to search for a particular region of interest (ROI). The key idea behind the search engine is the representation of a medical image dataset by (non-rigid) transformations between

exemplar images and all other images. Computing such a representation is feasible in the medical image domain, since the images are typically obtained under standardised acquisition protocols (with pre-defined field of view, etc.). At run time, given an image with an ROI in it, the pre-computed transformations are used to locate the ROI in the repository images using a fast transform composition technique. We have presented two practical implementations of our ROI retrieval architecture, operating on 2-D X-ray and 3-D MRI medical image collections.

9.2 Future Work

This thesis has addressed the problem of devising image representations for a number of computer vision applications. In this section, we envisage the potential ways of improving these representations.

Improving conventional image descriptors. Off-the-shelf image representations, such as VLAD or FV feature encoding, while being relatively well studied [Sánchez et al., 2013], still have a potential for improvement. One of the areas, which can bring significant gains, is the descriptor post-processing, or normalisation. It has been shown in the literature [Perronnin et al., 2010, Arandjelović and Zisserman, 2013, Delhumeau et al., 2013] that an appropriate normalisation scheme leads to a significant improvement of the results. Our experiments in Chapter 5 further confirm this observation – we were able to achieve a noticeable gain in FV classification performance simply by changing the normalisation type. It should be noted though, that the intra-normalisation, which we found beneficial on VOC 2007 dataset, did not improve on the signed square-rooting on LFW and ImageNet datasets. As explained in Chapter 5, the reason could be in the amount of bursty local features, which can differ between the datasets. Designing a normalisation strategy, equally beneficial for a variety of image data, is one of the objectives

for the future work.

Another way of improving the image descriptors is to consider several, complementary, local feature types. For instance, our face descriptor (Chapter 6) is based on a single feature type – dense SIFT. Many face recognition systems are based on the LBP features, so the fusion of several feature types (e.g. SIFT and LBP) is likely to improve the results. This can be achieved by performing the late fusion (by concatenating FV encodings), or the early fusion (by concatenating local features, and learning a joint codebook for the feature combination).

Simultaneous learning of several processing stages. Using our convex local descriptor learning framework (Chapters 3 and 4), we managed to learn the optimal configuration of pooling regions given the feature channels. In our case, we used eight SIFT-like gradient orientation channels, but even better performance might be achieved by employing more complex features as suggested by [Brown et al., 2011]. One way of doing it would be to sample combinations of pooling region (PR) configurations and various feature channels, performing the convex selection of not only PRs, but also the features. This approach, however, has its limitations, since only a limited number of features can be tried due to the computational reasons, and these features should be pre-defined, rather than learnt. Therefore, an interesting problem for future research is to develop learning formulations, which optimise both feature computation filters and their pooling.

Joint optimisation of several pipeline stages should also be beneficial for shallow and deep image descriptors. In the case of shallow feature encodings (Chapter 5), one can consider optimising over both the classification models and the parameters of the encoding, such as the codebook. The first step in this direction was made in Sect. 5.3.2, where we optimised over the local feature transformation in the clustering-aware manner. A more principled approach would be to optimise (or

fine-tune) the k-means or GMM clusters.

When learning our deep Fisher network architecture (Chapter 7), we optimised the dimensionality reduction stage using a learning proxy, without taking into account the layers on top of it. While this allowed us to come up with a tractable optimisation problem, the learnt projection is suboptimal in a sense that it does not take into account the classification performance of the Fisher network in whole. Learning the Fisher layers simultaneously remains an interesting problem to address.

Deep learning architectures. Recently, deep image representations have been shown to achieve excellent performance on several recognition tasks, provided that the amount of training data is sufficient to prevent over-fitting. Additionally, due to the high computational complexity, an optimised implementation on the highly parallel hardware (such as GPUs) is an essential component of deep network training. Therefore, we are interested in exploring the middle ground between conventional shallow architectures and deep representations. In particular, a challenging, but relevant problem is that of training deep representations using fast techniques on limited amounts of training data (e.g. without training set augmentation).

One hybrid approach, based on stacking Fisher encodings in a deep architecture, was presented in Chapter 7. But our current implementation is built on top of hand-crafted SIFT and colour features, which potentially limits its descriptive power. An interesting extension would be to start directly from the image intensity patches. Our preliminary experiments indicate that two Fisher layers on top of the grey-scale patches exhibit similar classification performance to a single Fisher layer on top of SIFT. This indicates that it might be possible to completely abandon hand-crafted features and achieve a competitive classification performance by the Fisher network encoding of colour images.

Learning medical image ranking. In Chapter 8, we proposed an architecture for the fast ROI retrieval from medical image collections. However, the problem of semantically meaningful ranking of the ROIs remains open. Depending on the application, the notion of the correct ranking is different, so one way of constructing a ranking procedure is to learn it automatically based on the ground-truth ranking, provided by the clinicians. To this end, one can employ discriminative learning-to-rank formulations [Joachims, 2002], operating on the image representations, described in this thesis.

Bibliography

A. Agarwal and B. Triggs. Hyperfeatures - multilevel local coding for visual recognition. In *Proceedings of the European Conference on Computer Vision*, pages 30–43, 2006. 119

T. Ahonen, A. Hadid, and M. Pietikäinen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006. 99

A. Alahi, R. Ortiz, and P. Vandergheynst. FREAK: Fast retina keypoint. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517, 2012. 16, 17

R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 26, 45, 59, 63, 76, 80, 81, 105

R. Arandjelović and A. Zisserman. All about VLAD. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 6, 83, 85, 95, 151, 153

U. Avni, H. Greenspan, E. Konen, M. Sharon, and J. Goldberger. X-ray categorization and retrieval on the organ and pathology level, using patch-based visual words. *IEEE Transactions on Medical Imaging*, 30(3):733–746, 2011. 131

A. Baumberg. Reliable feature matching across widely separated views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 774–781, 2000. 11

H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision*, May 2006. 11, 14

P. A. Beardsley, P. H. S. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In *Proceedings of the 4th European Conference on Computer Vision, Cambridge, UK*, pages 683–695, 1996. 12

P. R. Beaudet. Rotationally invariant image operators. In *Proceedings of the International Conference on Pattern Recognition*, pages 579–583, 1978. 11

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. 56

P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997. 18

P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 545–552, 2011. 98

M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems*, volume 14, pages 585–591, 2001. 35

A. J. Bell and T. J. Sejnowski. The independent components of natural scenes are edge filters. *Vision Research*, 37(23):3327–333, 1997. 34

S. Belongie and J. Malik. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24), 2002. 15

Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems*, pages 153–160, 2006. 30

A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego*, June 2005. 15

A Berg, J Deng, and L Fei-Fei. Large scale visual recognition challenge (ILSVRC), 2010. URL http://www.image-net.org/challenges/LSVRC/2010/. 28, 95, 119, 124, 125

T. Berg and P. N. Belhumeur. Tom-vs-Pete classifiers and identity-preserving alignment for face verification. In *Proceedings of the British Machine Vision Conference*, 2012. 98, 100

T. Berg and P. N. Belhumeur. POOF: Part-based one-vs-one features for fine-grained categorization, face verification, and attribute estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 68, 69

E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, pages 245–250, 2001. 35

O. Boiman, E. Shechtman, and M. Irani. In defense of Nearest-Neighbor based image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 20

X. Boix, M. Gygli, G. Roig, and L. Van Gool. Sparse quantization for patch description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 15, 46, 59, 60, 65, 66, 68

Y. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2559–2566, 2010. 23

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. 16

M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1): 43–57, 2011. 15, 44, 45, 46, 58, 59, 60, 63, 64, 67, 70, 71, 154

A. Burner, R. Donner, M. Mayerhoefer, M. Holzer, F. Kainberger, and G. Langs. Texture bags: Anomaly retrieval in medical images based on local 3D-texture similarity. In *Proceedings of the MICCAI International Workshop on Content-Based Retrieval for Clinical Decision Support*, pages 116–127, 2011. 131

M. Calonder, V. Lepetit, and P. Fua. Keypoint signatures for fast learning and recognition. In *Proceedings of the European Conference on Computer Vision*, pages 58–71, 2008. 16

M. Calonder, V. Lepetit, P. Fua, K. Konolige, J. Bowman, and P. Mihelich. Compact signatures for high-speed interest point description and matching. In *Proceedings of the International Conference on Computer Vision*, pages 357–364, 2009. 16

M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary robust independent elementary features. In *Proceedings of the European Conference on Computer Vision*, 2010. 16, 17, 60

X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2887–2894, 2012. 99

M. J. Cardoso, M. J Clarkson, G. R Ridgway, M. Modat, N. C. Fox, and S. Ourselin. LoAd: A locally adaptive cortical segmentation algorithm. *NeuroImage*, 56(3): 1386–1397, 2011. 145

M. J. Cardoso, M. Modat, S. Ourselin, S. Keihaninejad, and D. Cash. Multi-STEPS: Multi-label similarity and truth estimation for propagated segmentations. In *Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 153–158, 2012. 145

M. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings on ACM Symposium on Theory of Computing*, pages 380–388, 2002. 17, 66

K. Chatfield and A. Zisserman. Visor: Towards on-the-fly large-scale object category retrieval. In *Proceedings of the Asian Conference on Computer Vision*, Lecture Notes in Computer Science. Springer, 2012. 88

K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *Proceedings of the British Machine Vision Conference*, 2011. 6, 21, 26, 83, 84, 89, 101, 105

D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *Proceedings of the European Conference on Computer Vision*, pages 566–579, 2012. 104

D. Chen, X. Cao, F. Wen, and J. Sun. Blessing of dimensionality: High dimensional feature and its efficient compression for face verification. In *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, 2013. 19, 98, 99, 100, 103, 104, 111, 112

H.-T. Chen, H.-W. Chang, and T.-L. Liu. Local discriminant embedding and its variants. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 846–853, 2005. 37

H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2-3):114–141, February 2003. 139

O. Chum, J. Matas, and Š. Obdržálek. Enhancing RANSAC by generalized model optimization. In *Proceedings of the Asian Conference on Computer Vision*, volume 2, pages 812–817, January 2004. 140

D. C. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649, 2012. 30

A. Coates, A. Y. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2011. 119

T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proceedings of the European Conference on Computer Vision*, pages 484–498, 1998. 18

T. F. Cootes, C. J. Twining, V. S. Petrovic, R. Schestowitz, and C. J. Taylor. Groupwise construction of appearance models using piece-wise affine deformations. In *Proceedings of the British Machine Vision Conference*, 2005. 135

T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman & Hall, 2001. 32

K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001. 92

G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004. 22

Z. Cui, W. Li, D. Xu, S. Shan, and X. Chen. Fusing robust face region descriptors via multiple metric learning for face recognition in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 100

J. Delhumeau, P.-H. Gosselin, H. Jégou, and P. Pérez. Revisiting the VLAD image representation. In *Proceedings of the ACM Multimedia Conference*, 2013. 85, 86, 90, 153

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 31

T. M. Deserno. IRMA dataset, 2009. URL `http://ganymed.imib.rwth-aachen.de/irma/datasets_en.php`. 137

J. C. Duchi and Y. Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10:2899–2934, 2009. 56

M. Everingham, J. Sivic, and A. Zisserman. "Hello! My name is... Buffy" – automatic naming of characters in TV video. In *Proceedings of the 17th British Machine Vision Conference, Edinburgh*, 2006. 19, 99

M. Everingham, J. Sivic, and A. Zisserman. Taking the bite out of automatic naming

of characters in TV video. *Image and Vision Computing*, 27(5), 2009. 98, 99, 105, 110, 111

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 6, 84, 142

A. Farhadi, M. K. Tabrizi, I. Endres, and D. A. Forsyth. A latent model of discriminative aspect. In *Proceedings of the International Conference on Computer Vision*, pages 948–955, 2009. 43

M. Fazel, H. Hindi, and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the American Control Conference*, pages 4734–4739, 2001. 53

R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego*, 2005. 19

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936. 36

K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980. 13, 29

Y. Gong and S. Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 17

A. Gordo, J. A. Rodríguez-Serrano, F. Perronnin, and E. Valveny. Leveraging category-level labels for instance-level image retrieval. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, pages 3045–3052, 2012. 43, 119, 122, 123

M. O. Gueld, M. Kohnen, D. Keysers, H. Schubert, B. Wein, J. Bredno, and T. M. Lehmann. Quality of DICOM header information for image categorization. In *Proceedings of the SPIE International Symposium on Medical Imaging*, volume 4685, pages 280–287, 2002. 138

M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? Metric learning approaches for face identification. In *Proceedings of the International Conference on Computer Vision*, 2009. 19, 26, 40, 99, 103, 107, 112

M. Guillaumin, J. Verbeek, and C. Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *Proceedings of the European Conference on Computer Vision*, pages 634–647, 2010. 42

Z. Harchaoui, M. Douze, M. Paulin, M. Dudík, and J. Malick. Large-scale image classification with trace-norm regularization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3386–3393, 2012. 54

C. G. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference, Manchester*, pages 147–151, 1988. 11

X. He and P. Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems*, 2004. 35

X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005. 35

G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006. 30

G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012. 30

G. Hua, M. Brown, and S. Winder. Discriminant embedding for local image descriptors. In *Proceedings of the International Conference on Computer Vision*, pages 1–8, 2007. 37

C. Huang, S. Zhu, and K. Yu. Large scale strongly supervised ensemble metric learning, with applications to face verification and retrieval. *CoRR*, abs/1212.6094, 2012a. 106, 112

G. B. Huang, V. Jain, and E. Learned-Miller. Unsupervised joint alignment of complex images. In *Proceedings of the 11th International Conference on Computer Vision, Rio de Janeiro, Brazil*, 2007a. 99, 110

G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, 2007b. 6, 106, 107

G. B. Huang, M. Mattar, H. Lee, and E. Learned-Miller. Learning to align from scratch. In *Advances in Neural Information Processing Systems*, pages 773–781, 2012b. 99, 110

D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160:106–154, 1962. 13, 29

I. Isgum, M. Staring, A. Rutten, M. Prokop, M. Viergever, and B. van Ginneken. Multi-atlas-based segmentation with local decision fusion – application to cardiac

and aortic segmentation in ct scans. *IEEE Transactions on Medical Imaging*, 28 (7):1000–1010, 2009. 135

T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems*, pages 487–493, 1998. 24

C. R. Jack, M. M. Shiung, J. L. Gunter, P. C. O'Brien, S. D. Weigand, D. S. Knopman, B. F. Boeve, R. J. Ivnik, G. E. Smith, R. H. Cha, E. G. Tangalos, and R. C. Petersen. Comparison of different MRI brain atrophy rate measures with clinical disease progression in AD. *Neurology*, 62(4):591–600, 2004. 144

P. Jain, B. Kulis, and I. S. Dhillon. Inductive regularized learning of kernel functions. In *Advances in Neural Information Processing Systems*, pages 946–954, 2010. 54

H. Jégou and O. Chum. Negative evidences and co-occurrences in image retrieval: the benefit of PCA and whitening. In *Proceedings of the European Conference on Computer Vision*, 2012. 85, 90

H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2009. 27, 86, 96

H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 6, 17, 47, 60, 67, 83

H. Jégou, T. Furon, and J.-J. Fuchs. Anti-sparse coding for approximate nearest neighbor search. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2029–2032, 2012a. 17, 45, 57, 58, 67, 151

H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local images descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012b. 26, 83

T. Joachims. Optimizing search engines using clickthrough data. In *ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, pages 133–142, 2002. 50, 156

S. Klein, M. Staring, K. Murphy, M.A. Viergever, and J.P.W. Pluim. elastix: A toolbox for intensity-based medical image registration. *IEEE Transactions on Medical Imaging*, 29(1):196–205, 2010. 143

J. Kovacevic and A. Chebira. An introduction to frames. *Foundations and Trends in Signal Processing*, 2(1):1–94, 2008. 17, 57

J. Krapac, J. Verbeek, and F. Jurie. Modeling spatial layout with fisher vectors for image categorization. In *Proceedings of the International Conference on Computer Vision*, pages 1487–1494, 2011. 28, 84

A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1106–1114, 2012. 7, 30, 31, 116, 127, 128, 152

M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197, 2010. 74

N. Kumar, A. C. Berg, P. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *Proceedings of the International Conference on Computer Vision*, 2009. 100

M. Lam, T. Disney, M. Pham, D. Raicu, J. Furst, and R. Susomboon. Content-based image retrieval for pulmonary computed tomography nodule images. In *Proceedings of SPIE*, volume 6516, 2007. 131

C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958, 2009. 123

S. Lazebnik, C. Schmid, and J Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New York*, 2006. 27

Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng. Building high-level features using large scale unsupervised learning. In *Proceedings of the International Conference on Machine Learning*, 2012. 31

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. 29

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 13, 29, 30

V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1465–1479, 2006. 15

T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, June 2001. 12

S. Leutenegger, M. Chli, and R. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *Proceedings of the International Conference on Computer Vision*, pages 2548–2555, 2011. 16, 17, 60

H. Li, G. Hua, J. Brandt, and J. Yang. Probabilistic elastic matching for pose variant face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 110, 111, 112, 113

P. Li, Y. Fu, U. Mohammed, J. H. Elder, and S. J. D. Prince. Probabilistic models for inference about identity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):144–157, November 2012. 99, 112

T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116, 1998. 10, 11

D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece*, pages 1150–1157, September 1999. 13

D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 6, 11, 13, 14, 44, 46, 78, 82, 139, 151

P. C. Mahalanobis. On the generalised distance in statistics. In *Proceedings of the National Institute of Science, India*, volume 2, pages 49–55, 1936. 38

J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *Advances in Neural Information Processing Systems*, pages 1033–1040, 2008. 23

J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from

maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 384–393, 2002. 11, 12, 77

K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*. Springer-Verlag, 2002. 11, 78

K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005. 14

K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1/2):43–72, 2005. 11, 12, 72, 73, 77

M. Modat, Z. Taylor, J. Barnes, D. Hawkes, N. Fox, and S. Ourselin. Fast free-form deformation using graphics processing units. *Computer Methods and Programs in Biomedicine*, 98(3):278–284, 2010. 144

S. G. Mueller, M. W. Weiner, L. J. Thal, R. C. Petersen, C. R. Jack, W. Jagust, J. Q. Trojanowski, A. W. Toga, and L. Beckett. Ways toward an early diagnosis in Alzheimer's disease: The Alzheimer's Disease Neuroimaging Initiative (ADNI). *Alzheimer's and Dementia*, 1(1):55–66, 2005. 144

H. Müller, N. Michoux, D. Bandon, and A. Geissbuhler. A review of content-based image retrieval systems in medical applications – clinical benefits and future directions. *International Journal of Medical Informatics*, 73(1):1–23, 2004. 130, 131

Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259, 2009. 55

E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *Proceedings of the European Conference on Computer Vision*, 2006. 12, 21

B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, 1997. 23

M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 16

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine and Journal of Science*, 6(2):559–572, 1901. 32

F. Perronnin and D. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 24, 25

F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *Proceedings of the European Conference on Computer Vision*, 2010. 6, 26, 27, 83, 85, 89, 97, 101, 106, 118, 120, 127, 153

F. Perronnin, Z. Akata, Z. Harchaoui, and C. Schmid. Towards good practice in large-scale learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3482–3489, 2012. 121, 124

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis*, 2007. 6, 10, 21, 22, 44, 72, 76, 77, 81, 82, 131, 143

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska*, 2008. 22

J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Descriptor learning for efficient retrieval. In *Proceedings of the European Conference on Computer Vision*, 2010. 44, 45, 72, 75, 76, 77, 81, 82, 151

D. Picard and P. H. Gosselin. Improving image similarity with vectors of locally aggregated tensors. In *Proceedings of the IEEE International Conference on Image Processing*, pages 669–672, 2011. 89

N. Pinto, J. J. DiCarlo, and D. D. Cox. How far can you get with a modern face recognition test set using only simple features? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 113

M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*, 7(2):155–162, 1964. 15

P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *Proceedings of the International Conference on Computer Vision*, pages 754–760, 1998. 10, 12

B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010. 53

J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the International Conference on Machine Learning*, pages 713–719, 2005. 53

F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958. 29

E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:105–119, 2010. 11

E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski. Orb: An efficient alternative to SIFT or SURF. In *Proceedings of the International Conference on Computer Vision*, pages 2564–2571, 2011. 16, 17

D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes. Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Transactions on Medical Imaging*, 18(8):712–721, 1999. 144

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. 29

J. Sánchez and F. Perronnin. High-dimensional signature compression for large-scale image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 28, 101, 124, 127

J. Sánchez, F. Perronnin, and T. Emídio de Campos. Modeling the spatial layout of images beyond spatial pyramids. *Pattern Recognition Letters*, 33(16):2216–2223, 2012. 28, 84, 87, 89, 90, 124

J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image Classification with the Fisher Vector: Theory and Practice. *International Journal of Computer Vision*, June 2013. 6, 83, 153

S. Savarese, A. Criminisi, and J. Winn. Discriminative object class models of ap-

pearance and shape by correlatons. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New York*, 2006. 28

F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, volume 1, pages 414–431. Springer-Verlag, 2002. 11, 78

B. Scholkopf, A. Smola, and K. R. M uller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998. 33

P. Sermanet and Y. LeCun. Traffic sign recognition with multi-scale convolutional networks. In *International Joint Conference on Neural Networks*, pages 2809–2813, 2011. 121

T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):411–426, 2007. 13

S. Shalev-Shwartz, Y. Singer, and A. Ng. Online and batch learning of pseudo-metrics. In *Proceedings of the International Conference on Machine Learning*, 2004. 40

S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient SOlver for SVM. In *Proceedings of the International Conference on Machine Learning*, volume 227, 2007. 122

G. Sharma, S. Hussain, and F. Jurie. Local higher-order statistics (LHS) for texture categorization and facial analysis. In *Proceedings of the European Conference on Computer Vision*, 2012. 100

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 136

K. Simonyan, A. Zisserman, and A. Criminisi. Immediate structured visual search for medical images. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, 2011. 8

K. Simonyan, M. Modat, S. Ourselin, D. Cash, A. Criminisi, and A. Zisserman. Immediate roi search for 3-d medical images. In *Proceedings of the MICCAI International Workshop on Content-Based Retrieval for Clinical Decision Support*, 2012a. 8

K. Simonyan, A. Vedaldi, and A. Zisserman. Descriptor learning using convex optimisation. In *Proceedings of the European Conference on Computer Vision*, 2012b. 8, 51, 68, 77, 81

K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher Vector Faces in the Wild. In *Proceedings of the British Machine Vision Conference*, 2013a. 8

K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. Technical report, Department of Engineering Science, University of Oxford, July 2013b. 8

K. Simonyan, A. Vedaldi, and A. Zisserman. Deep Fisher networks for large-scale image classification. In *Advances in Neural Information Processing Systems*, 2013c. 8

J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the 9th International Conference on Computer Vision, Nice, France*, volume 2, pages 1470–1477, 2003. 3, 7, 10, 17, 21, 22, 47, 131

N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, volume 25, pages 835–846, 2006. 44

J. Sochman and J. Matas. Learning fast emulators of binary decision processes. *International Journal of Computer Vision*, 83(2):149–163, 2009. 11

C. Strecha, Bronstein A. M., M. M. Bronstein, and P. Fua. LDAHash: Improved matching with smaller descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1), 2012. 17, 60

Y. Taigman and L. Wolf. Leveraging billions of faces to overcome performance barriers in unconstrained face recognition. *CoRR*, abs/1108.1122, 2011. 112

Y. Taigman, L. Wolf, and T. Hassner. Multiple one-shots for utilizing class label information. In *Proceedings of the British Machine Vision Conference*, 2009. 99, 101, 112

E. Tola, V. Lepetit, and P. Fua. A fast local descriptor for dense matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 14, 46, 61

A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1521–1528, 2011. 96

L. Torresani and K. Lee. Large margin component analysis. In *Advances in Neural Information Processing Systems*, pages 1385–1392. MIT Press, 2007. 42

L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classemes. In *Proceedings of the European Conference on Computer Vision*, pages 776–789, sep 2010. 123

T. Trzcinski and V. Lepetit. Efficient discriminative projections for compact binary descriptors. In *Proceedings of the European Conference on Computer Vision*, 2012. 17, 60, 68

T. Trzcinski, M. Christoudias, V. Lepetit, and P. Fua. Learning image descriptors with the boosting-trick. In *Advances in Neural Information Processing Systems*, pages 278–286, 2012. 16, 45, 59, 63, 67, 68

T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit. Boosting bnary keypoint descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 16, 44, 46, 59, 60, 65, 66, 68

M. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991. 18

T. Tuytelaars, M. Fritz, K. Saenko, and T. Darrell. The NBNN kernel. In *Proceedings of the International Conference on Computer Vision*, pages 1824–1831, 2011. 20

J. C. van Gemert, J. M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders. Kernel codebooks for scene categorization. In *Proceedings of the European Conference on Computer Vision*, 2008. 22

M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *Proceedings of the International Conference on Computer Vision*, Rio de Janeiro, Brazil, October 2007. 138

A. Vedaldi and B. Fulkerson. VLFeat - an open and portable library of computer vision algorithms. In *Proceedings of the ACM Multimedia Conference*, 2010. 78, 105

A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 26

A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Proceedings of the International Conference on Computer Vision*, 2009. 138

P. Viola and M. Jones. Robust real-time object detection. In *International Journal of Computer Vision*, volume 1, 2001. 98, 105

H. Wang, S. Yan, D. Xu, X. Tang, and T. Huang. Trace ratio vs. ratio trace for dimensionality reduction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 37

J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 23

P. Wang, J. Wang, G. Zeng, W. Xu, H. Zha, and S. Li. Supervised kernel descriptors for visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 68, 69

M. K. Warmuth and D. Kuzmin. Randomized online PCA algorithms with regret bounds that are logarithmic in the dimension. *Journal of Machine Learning Research*, 9:2287–2320, 2008. 33

K. Q. Weinberger and L.K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009. 40, 41

K. Q. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems*, 2006. 40, 49

J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: Learning to rank with joint word-image embeddings. In *Proceedings of the European Conference on Machine Learning*, 2010. 43, 94

J. Weston, S. Bengio, and N. Usunier. WSABIE: Scaling up to large vocabulary image annotation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2764–2770, 2011. 122

L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Faces in Real-Life Images Workshop in European Conference on Computer Vision*, 2008. 99, 101

L. Wolf, T. Hassner, and Y. Taigman. Similarity scores based on background samples. In *Proceedings of the Asian Conference on Computer Vision*, 2009. 99, 101

L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, 2010. 45, 54, 55, 56, 150

E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems*, volume 15, pages 505–512, 2002. 39

S. Yan, X. Xu, D. Xu, S. Lin, and X. Li. Beyond spatial pyramids: A new feature extraction framework with dense spatial sampling for image classification. In *Proceedings of the European Conference on Computer Vision*, pages 473–487, 2012. 119

J. Yang, K. Yu, Y. Gong, and T. S. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1794–1801, 2009. 23

Z. Zhang, R. Deriche, O. D. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87–119, 1995. 12

X. Zhou, K. Yu, T. Zhang, and T. S. Huang. Image classification using super-vector coding of local image descriptors. In *Proceedings of the European Conference on Computer Vision*, 2010. 26