Efficient learning of local image descriptors

Vassileios Balntas

A thesis presented for the degree of
Doctor of Philosophy

University of Surrey
UK
June 2016

Abstract

One of the most important tasks of modern computer vision with a vast amount of applications is finding correspondences between local patches extracted from different views of a physical scene. In this thesis, we investigate three main axes of this problem.

We first provide a critical review of the prior work related to methods for extracting local image descriptors. Next, we show that the intrinsic visual characteristics of a patch may fundamentally alter its matching process, and we show how to exploit this phenomenon to improve the matching performance. One of the main contributions of this thesis is a novel approach to describing and matching image patches. We introduce a per-patch adapted method which makes it possible to generate feature descriptors that use simple binary tests, but match the performance of methods of significantly higher complexity. We also demonstrate that our technique can be successfully generalised to other descriptors, thus showing its potential for more general applications.

We then propose novel methods to learn compact and efficient patch representations using convolutional neural networks. We show that typically used approaches such as architectural expansions or hard negative mining are not essential for the success of such methods. Our convolutional descriptors outperform the state of the art approaches at a significant fraction of the computational cost.

Lastly, we demonstrate that most of the work in the area suffers from non-reproducibilty and inconsistency of evaluation results. To that end, we introduce a novel dataset accompanied with improved protocols and benchmarks that will allow for reproducible results. More importantly, the scale of our dataset allows for experimentation with learning local feature descriptors from real-world data,

something that has not been feasible so far due to the lack of data. This will allow improved results and new experiments especially in the context of deep learning and convolutional neural networks.

keywords: feature descriptor, local features, image patches, binary descriptor, convolutional neural networks.

List of Figures

List of Tables

Contents

List Of Abbreviations

BOLD ..... Binary Online Learned Descriptor

CNN ..... Convolutional Neural Network

FN ..... False Negatives

FP ..... False Positives

FPR95 ..... False Positive Rate at 95% True Positive Rate

mAP ..... mean Average Precision

NN ..... Nearest Neighbour

ROC ..... Receiver Operating Characteristic

TN ..... True Negatives

TP ..... True Positives

List Of Publications

- V. Balntas, L. Tang and K. Mikolajczyk. *BOLD - Binary Online Learned Descriptor For Efficient Image Matching.* CVPR, 2015
- V. Balntas, L. Tang and K. Mikolajczyk. *Binary Online Learnt Descriptors.* PAMI, 2016 (submitted)
- V. Balntas, E. Johns, L. Tang and K. Mikolajczyk. *PN-Net: Conjoined Triple Deep Network for Learning Local Image Descriptors.* arXiv 2016
- V. Balntas, E. Riba, D. Ponsa and K. Mikolajczyk. *Learning local feature descriptors with triplets and shallow convolutional neural networks.* BMVC, 2016

Acknowledgements

First of all, I want to thank my supervisor Krystian Mikolajczyk who has been a constant source of new ideas and helped me out throughout all these years. His immense knowledge of the subject was a major source of inspiration for this thesis. Secondly, I also want to greatly thank my co-supervisor Lilian Tang, for giving me opportunities and providing me with support across many different occasions during my studies.

A big thank you to my examiners Vincent Lepetit from TU Graz and Richard Bowden from the University of Surrey, for initiating a great discussion and providing excellent feedback on my thesis.

I want to thank Edgar Riba from UAB, Edward Johns, Rigas Kouskouridas, Andreas Doumanoglou and T-K Kim from Imperial College, and Phil Smith from the University of Surrey.

My parents Kostantinos and Eftychia, and my sister Christina, have always believed in the importance of education. My family has provided me with great love and I am forever grateful to them for supporting me throughout my undergraduate and graduate studies.

Last but not least, I want to thank Chrysa, who had to put up with several late-night submission deadlines and many periods of significant amount of work. Looking forward to our next adventures.

Vassileios Balntas, Oct 2016.

Chapter 1

Introduction

One of the most important tasks of computer vision is to represent distinctive local image patches in a way that their representation is invariant under different viewing conditions. This is a crucial step in multiple applications such as structure from motion, image retrieval, object recognition, simultaneous localisation and mapping (SLAM) and tracking. Recent interest in self-driving cars has made this area a very important part of any relevant advancement. In its most basic form, this problem is presented in Figure 1.1. We show the same physical area in the 3D world, captured from two very different angles. The goal of a robust feature descriptor is to represent a specific area of the image on the left in such a way that it can be easily matched with the equivalent area on the right image.

Local feature descriptors should be robust to various transformations, such as blurring, affine projections and illumination changes, while at the same time being efficient to compute, low in memory requirements and fast to match. Influential early work in this area focused on real-valued feature vectors extracted from distributions of image characteristics such as gradients and colours. However the computational complexity of estimating distributions using real valued features limits the set of applications where such features could be efficiently employed. In addition, large-scale methods such as searching among billions of examples, require descriptors to have smallest memory footprint possible.

Driven by the need for faster extraction and lower memory, researchers explored the use of simple binarised intensity differences in the place of computa-

Figure 1.1: Local feature description is concerned with methods to match using feature description specific areas of the image on the left with the image on the right. Note the change in viewpoint, illumination, reflections and focus.

tionally demanding pooling and gradient operators. Gradient operators differentiate neighbouring pixel intensities while pooling operators accumulate the resulting values within small regions into local histograms. On the other hand, the binarised intensity methods were able to run on devices with very limited computational capabilities such as mobile robots, embedded systems and smartphones. The main advantage of such methods is their efficient matching which reduces down to computing Hamming distance between two sets of bits, and can be evaluated using XOR operations directly in the hardware. This property of the intensity operators, together with the significantly lower storage requirements and the fast distance computation, has led to extensive development and use of such methods in wide range of relevant bibliography. However, such methods cannot currently reach the discriminative power of more robust methods based on differentiation and pooling due to the weak discriminative nature of the individual intensity tests used for computing the features. Thus, they are limited to the problems where the need for computational efficiency outweighs the requirement of very accurate results.

Another important research direction that has gained a lot of attention is the use of machine learning techniques in order to learn the optimal configurations of the pooling regions. While early work focused on non-analytical or non-convex approaches, and therefore were not guaranteed to find an optimal solution, recently a convex optimisation method was presented with improved performance by ensuring that the global optimum is found [Simonyan et al., 2014]. However, all the

approaches in this category were based on optimising features already extracted or optimising the configuration architecture for a set of pre-defined filtering methods such as Gaussian kernels or wavelets. Currently, the best performing methods of learning such representations, are the hierarchical learning of filters via convolutional neural networks and deep learning. Such methods operate with convolutions and simple non-linear operations, in consecutive layers, where each layer learns more abstract representations than the previous one. The success of deep learning methods across all areas of computer vision in recent years highlights the importance of hierarchical learning in knowledge representations. Subsequently, it has also strongly impacted the area of local patch description with many works exploring optimising such networks as feature descriptors. Unfortunately, previous work in this area is built upon networks that are very computationally demanding in terms of both learning time and extraction time, thus prohibiting real-time applications or applications in embedded devices. The recent interest in implementing convolutional neural networks in FPGAs [Lacey et al., 2016] shows that there is strong interest in this field. In addition, allowing for such methods to run on devices with limited computational power, will lead to a new wave of applications that are not currently possible.

Above all, the basis of improving the state of the art in any field, is a well designed evaluation method that aims to be as reproducible as possible. Several important works in the field of local feature descriptors managed to clearly define evaluation metrics and protocols, something that has led to a much needed standardisation of the results. However, there are still issues with the most commonly used benchmarks, which are related to the lack of reproducibility. Furthermore, the volume of the majority of the currently available feature descriptor datasets makes them insufficient for training deep learning methods which could show significant improvement in their performance if exposed to large scale training data.

1.1 Research objectives

The main objectives of this thesis are to provide insights on the following general research problems related to the field of local patch description and matching.

More specifically this thesis focuses on the following objectives

- *Simultaneous improvement of robustness and computational efficiency.* There is a trade-off between efficiency and discriminative power of feature descriptors. We design methods that provide improvements in both areas.

- *Combination of computationally demanding learning methods with fast online learning approaches.* Typically used machine learning methods are too complex to be applied in an online manner. Our next significant challenge is to implement methods that are locally and online adapted but exhibit no significant decrease in efficiency.

- *Convolutional neural networks as feature descriptors.* Investigate the feasibility of using convolutional neural networks as feature descriptors, with specific focus on low computational requirements.

- *Shortcomings of existing benchmarks and possible improvements.* Arguably reproducible and rigorous evaluations are more important than novel approaches, whose true value is assessed by such evaluations. There is scope for improvement in this area since the currently available evaluation benchmarks suffer from several limitations.

1.2 Contributions

In this thesis, we present three main contributions in the area of patch description, that aim to address some of the challenges discussed in the previous section.

Our first contribution is introducing a **novel paradigm in matching patch descriptors**, where the computation of a distance between two patches is not based on a full and fixed set of feature dimensions, but on carefully selected subsets for each patch, that improve robustness for each individual feature dimension. We then show that such a technique can achieve the speed of the fast binary descriptors while maintaining the discriminative ability of the more computationally intensive gradient pooling descriptors.

While our local adaptation of patch descriptors can lead to significant improvements, the state of the art in terms of discriminative power is feature representations based on convolutional neural networks. Previous work in this area consists of complex architectures and inefficient extraction methods that do now allow for large scale applications and real time performance. To that end, we explore methods to learn very **fast convolutional feature descriptors**, that can be extracted (with GPUs) as fast as the most efficient binary descriptors available.

Lastly, we show that the commonly used datasets and evaluations in the area of feature descriptors are not consistent, and we critically discuss the benchmarking methods used. In addition, we introduce a **novel large-scale dataset** which jointly enables training methods that require a vast amount of data such as convolutional feature descriptors, and can lead to more robust evaluations for all feature descriptors.

1.3 Outline

This thesis is organised as follows. Chapter 2 discusses previous work on patch descriptors and provides a categorisation in terms of different design processes and goals. In Chapter 3 we introduce a binary descriptor that locally adapts the feature description and matching process to each individual patch. We then show that such methods can be generalised to other descriptors of both binary and floating point types. In Chapter 4, we explore different methods to build efficient patch descriptors using convolutional neural networks. In Chapter 5, we discuss some problems with the current benchmarks, and introduce a new large-scale evaluation dataset that aims to standardise the matching process. Finally Chapter 6 summarises the work, and provides some possible directions for future research.

Chapter 2

Related work

In this chapter, we give an overview of the history of local feature descriptors. We first provide the required definitions to follow the relevant bibliography, and subsequently describe and categorise previously developed methods related to feature descriptors.

2.1 Definitions

Following the terminology from [Vedaldi and Fulkerson, 2008] we define a *local feature frame* or *frame* as a an abstract geometric object that defines a specific area of the image. Most common types of frames include *points, squares, circles, ellipses, oriented circles* and *oriented ellipses*. Each of these categories, is defined by the mathematical characteristics of the respective geometric class. Thus, a *point* is defined by its center $\boldsymbol{x}$, a circle by a center $\boldsymbol{x}$ and a radius σ, and an ellipse by a center $\boldsymbol{x}$ and three parameters $\boldsymbol{c}$ that satisfy the elliptical equation in the plane.

Given image $\boldsymbol{i} \in \mathbb{R}^{M \times N}$, and *frame* $\boldsymbol{f}$, we define a *patch* as geometrically normalised (e.g. elliptical affine warping) sub-image $\boldsymbol{i}_f \in \mathbb{R}^{K \times K}$ extracted from $\boldsymbol{i}$ using local frame $\boldsymbol{f}$. Normally, the frame is enlarged by an enlargement factor ρ in order to accommodate more information in the description process. Note that there is vast bibliography on how to identify robust frames inside an image, and the interested readers can refer to specific literature and extended surveys on local

feature detectors and local covariant features [Tuytelaars and Mikolajczyk, 2008, Mikolajczyk and Schmid, 2005].

In this thesis we focus on the *description-matching* part of the *detection-description-matching* pipeline, and we assume that a list of measurements is extracted from local frames and are available as normalised patches. Given normalised patch $\boldsymbol{p} \in \mathbb{R}^{K \times K}$, we define as *descriptor* of this patch, a vector $\boldsymbol{d_p} \in \mathbb{R}^D$ which by design, is expected to be more invariant to application-specific deformations than $\boldsymbol{p}$ itself. Note that while many authors regard the vectorised version of a patch as a descriptor, it is more desirable to aim for representations that are (a) of much lower dimensionality than the patch (b) more discriminative and robust to visual appearance changes.

A large number of feature descriptors have been introduced in the relevant literature, and thus a full enumeration would be out of the scope of this thesis. However, below we group the approaches based on their common characteristics, and we discuss several members of each group. In addition, we also discuss a general categorisation of descriptors that we will use throughout this thesis. It aims to divide the relevant work into four distinct categories according to the design process, and the representation of the output.

Floating point versus binary. The first categorisation that we consider is the representation of the output of the resulting feature vector. We refer to a descriptor $\boldsymbol{d}$ as floating-point if $\boldsymbol{d}_i \in \mathbb{R}$. A special case is a binary descriptor where $\boldsymbol{d}_i \in \{0, 1\}$.

Engineered versus learnt. The second categorisation refers to the design philosophy behind patch descriptors. We refer to an *engineered* descriptor where it is specifically designed based on some domain knowledge, and does not involve an optimisation based on data related to the domain. On the other hand, a descriptor is considered as *learnt*, if there is a specific optimisation process that adjusts the descriptor design to a collected training dataset.

2.2 Real-valued patch descriptors

We first discuss the floating point descriptors which were historically predecessors of the binary counterparts.

2.2.1 Engineered real-valued patch descriptors

The straightforward technique to address the challenges related to illumination invariance of the raw patch feature vectors, is some further processing such as the zeroed-mean-unit-variance patch ($ZMUV$) normalisation, which is defined as $\hat{\boldsymbol{p}} = \frac{mean(\boldsymbol{p})}{std(\boldsymbol{p})}$. This approach makes such descriptors more robust to illumination changes [Mikolajczyk and Schmid, 2005]. Nevertheless, the performance of such methods is limited, since it is not invariant to simple geometric deformations. In addition, the dimensionality of such a descriptor can be very high even for very small normalised patches e.g. it can reach 2^{10} for a 32×32 patch.

In order for a descriptor to be invariant to simple geometric affine deformations, non-uniform illumination changes and at the same time exhibit much lower dimensionality, it has to be built on statistics collected from the patch via some operators (e.g. filtering), preferably applied locally, and aggregated into histograms that can subsequently be used as the final feature descriptor. This basic design and process of filtering and aggregating gradients from local neighbourhoods, gave rise to very sucesfull and influential early feature descriptors.

An important moment in the history of local feature descriptors is the introduction of **SIFT** [Lowe, 1999]. The SIFT descriptor is a spatial histogram of a quantised version of the patch gradients. The local spatial pooling of the descriptor is based on a rectangular grid that partitions the patch into several regions. Assuming the patch is divided into M rectangular areas, and the gradients are quantised to K angle bins, the resulting K dimensional histograms concatenated from M areas, will be represented by a point in the $\mathbb{R}^{M*K}$ space. In the case of the original implementation of SIFT, 16 grid quanta were combined with 8 angular bins, resulting in final dimensionality of 128. Later [Dalal and Triggs, 2005] used overlapping rectangular regions to extract the **HoG** descriptor leading to much higher dimensional vectors, and showed that this representation is very

successful in addressing the problem of human detection in images. Several researchers proposed to alter the rectangular grid in order to make the process more invariant to deformations, such as rotations. [Mikolajczyk and Schmid, 2005] used polar spatial regions for improved robustness in their proposed **GLOH** descriptor. However in all of these previous works, the underlying principle of spatial and angular aggregation of gradients to histograms remains unchanged. A large number of references follow this approach such as **CHoG** [Chandrasekhar et al., 2009]. The **SURF** descriptor, introduced by [Bay et al., 2006], simplified the aforementioned methodology by using integral images to speed up the process, without significantly decreasing the performance. **DAISY** [Tola et al., 2010] is based on a similar procedure, but with much more complex sampling patterns, and a dense application throughout the whole image.

Several authors identified potential problems with spatially quantising the patch in order to aggregate local gradients. When dividing a patch into a spatial grid, each area of the grid will have boundaries that are not invariant to deformations such as rotations. To address this problem a new family of real-valued descriptors was introduced, which is not based on local spatial aggregation of gradients, but on local ordering methods. Such methods are based on results from sorting gradients or intensity values, and aggregating the sorted information. Note that these methods can be perfectly invariant to monotonic illumination changes and rotations. Prominent examples of this family of real-valued descriptors include **LIOP** [Wang et al., 2011a] **LUCID** [Ziegler et al., 2012] and **MROGH**,**MRRID** [Fan et al., 2012].

It is also worth noting that several authors identified that aggregation across different scales or different affine viewpoints into a single feature vector can improve the discriminative power of the descriptor, albeit at the price of much higher computational cost [Dong and Soatto, 2014, Yu and Morel, 2011, Wang et al., 2014c, Tsun-Yi Yang and Chuang, 2016]. Normally, such methods of aggregating across N different samples, results in a slowdown of a factor of N, and thus are not practical for most applications.

2.2.2 Learnt real-valued patch descriptors

A basic application of a simple machine learning method to the field of patch descriptors was the use of PCA to project the SIFT features into a lower dimensional yet more discriminative linear subspace [Ke and Sukthankar, 2004]. PCA is commonly used as a method to reduce the number of dimensions [Mikolajczyk and Schmid, 2005, Wang et al., 2014c]. However, due to its unsupervised nature, it is not very discriminative and exhibits limited performance gains.

Learning with supervised data is one of most successful trends in modern machine learning, and has lead to significant improvements in a vast number of applications [Halevy et al., 2009]. Learning patch descriptors is no exception and with the introduction of the Photo Tourism patches dataset [Winder and Brown, 2007], several works utilised the large number of labelled training data. This lead to descriptors that were optimised to this specific dataset, which consists of a large set of *positive* and *negative* patch pairs. The term *positive* pair refers to a pair of patches extracted from the same physical point in space but under different viewpoints, and the term *negative* refers to patches that come from different physical points in the space, and are thus less likely to share visually similar appearance.

Learning discriminative projections & discriminative configurations

The first set in works of supervised learning methods focused on learning discriminative projections for the most commonly used descriptors such as *SIFT*. Assuming the original patch descriptor is $\boldsymbol{x_p} \in \mathbb{R}^D$, the goal of the discriminative projection methods is to learn a function $\phi(\boldsymbol{x_p})$ such that when projecting the original descriptor $\boldsymbol{x_p}$, the distance between positive pairs is reduced, and the distance between negative pairs is increased. When the ϕ function is linear, then the projection can be represented as a matrix multiplication with the original features i.e. $W\boldsymbol{x}_p$ with $W \in R^{Q \times D}$, where Q is the dimensionality of the projected feature descriptor. A typical approach to learning a linear projection is via the linear discriminant analysis framework. Works that follow this approach are [Mikolajczyk and Matas, 2007, Cai et al., 2011, Hua et al., 2007].

Another line of work on learnt real valued descriptors is to identify filtering and pooling configurations aiming for better performance based on evaluation on

large-scale test datasets. Such methods investigate a collection of subsets of all possible configurations, thus aim to optimise the design of the descriptor extraction process, in contrast to optimising a discriminative projection function ϕ to be applied to an existing descriptor Such methods were presented in [Winder and Brown, 2007] and in [Winder et al., 2009]. The work in [Simonyan et al., 2014], is an important milestone for this category, since the learning framework is based on a convex optimisation, instead of non-analytic or non-convex methods. It is also worth noting that this work combines both approaches, as both the configuration of the pooling regions and the subsequent projection to discriminative subspaces are jointly optimised.

Deep convolutional patch descriptors

The rise of convolutional neural networks (CNN) as optimisation tools, gave a remarkable boost to many areas of computer vision [Lecun et al., 2015, Schmidhuber, 2015], and thus was also very influential in the area of local feature descriptors. The interest in CNNs in the area of descriptors was sparked by results shown in [Fischer et al., 2014] that the features from the last layer of a convolutional deep network trained on ImageNet dataset [Russakovsky et al., 2015] collected for general objects classification can outperform SIFT. In fact, such features approach the performance of descriptors resulting from convex optimisation [Simonyan et al., 2014], something that shows the remarkable power of hierarchically learned representations. This was a significant result, as the convolutional features trained on ImageNet were not specifically optimised for such local representations.

Early work on learning convolutional neural networks as feature descriptors specifically for local patches, was done in [Jahrer et al., 2008], but was not immediately followed possibly due to lack of rigorous evaluation. After the impressive success that convolutional neural networks exhibited in ImageNet [Krizhevsky et al., 2012] object classification, several authors revisited the idea, leading to works investigating convolutional architectures in the context of local feature descriptors. End-to-end learning of patch descriptors using identical multiple copies of a single CNN [Bromley et al., 1993, Chopra et al., 2005, Hadsell et al., 2006] have been attempted and more recently revisited in several works [Fischer et al., 2014,

Zagoruyko and Komodakis, 2015, Simo-Serra et al., 2015, Han et al., 2015] with consistent improvements on the state of the art descriptors.

Note that in [Zagoruyko and Komodakis, 2015, Han et al., 2015] both feature layers and metric layers are jointly learnt in the same network. Thus, the final computation of the distance is optimised in terms of the abstract metric learned in the last layer of the network. On the contrary, [Simo-Serra et al., 2015] directly use the features extracted after the convolutional layers of the CNN, without training a specialised distance layer, which allows the extracted descriptors to be used in traditional matching pipelines with euclidean L2 distance. In addition, such an architecture results in much faster matching, since the fully conected distance layer is much slower than a simple $L2$ distance computation. However, the experiments from [Zagoruyko and Komodakis, 2015] show that metric learning performs better than generic $L2$ matching.

2.3 Binary feature descriptors

Typically, for floating point descriptors the comparison is done using the $L2$ norm, which is computationally demanding especially for embedded and low capability devices. There is also a question of whether such fine precision is needed at the cost of efficiency when matching local image descriptors. For example, previous work has shown that reducing the floating point descriptors to `uint8` representations is sufficient for good matching [T. Trzcinski and Lepetit, 2013]. With this motivation, several works started investigating the possibility of using binary strings as description vectors.

Note that since the elements of the feature vector are limited to binary values, the euclidean distance is reduced to the hamming distance which can be implemented very efficiently by summing the results of a `XOR` operation. A significant motivation behind this research direction, was the introduction of the `SSSE3` *popcount* commands in [Intel, 2010], which leads to extremely fast `XOR` operations between binary values.

The first attempts to produce a binary feature descriptor focused on binarising extracted SIFT or other descriptors. This was done by applying a set of N random

hashes to produce an N-dimensional binary feature vector [Shakhnarovich, 2005, Torralba et al., 2008, Strecha et al., 2012]. Projecting floating point to binary representation makes it possible to reduce the amount of required memory by a factor of 32 and speed up the distance calculation. However, the fact that a floating point representation needs to be first extracted and then projected to the binary space, makes such methods very resource demanding. Thus, researchers focused on directly producing binary strings from raw patches.

2.3.1 Engineered binary feature descriptors

The **BRIEF** binary descriptor was introduced in [Calonder et al., 2010], and was based on binarised results of intensity tests as the extraction process for individual feature dimensions. An intensity test is a simple feature, that uses the signed result of intensity comparison between a pair of pixels, to provide a binary bit as the comparison result. K such tests could be used to create a binary string of K bits, that can be used as the final descriptor. **BRIEF** was the first descriptor that was very fast both to build and to match, due to its binarised intensity test and extremely fast matching with a distance based on `XOR` operation. Remarkably, BRIEF performs similarly to SURF,especially in benchmarks that simply measure the matching success rate (wthout considering precision and recall) in terms of nearest-neighbour search. A sampling pattern of a binary descriptor is a set of locations of intensity tests. Five distinct sampling patterns were examined in [Calonder et al., 2010], showing that a random Gaussian sampling around the centre of the patch, was able to outperform the remaining more regular and deterministic sampling patterns.

Several works proposed non-random sampling patterns, that experimentally outperformed the random Gaussian method, which indicates that the deterministic patterns included in the evaluation by Calonder et al. [2010] were not the optimal. For example, **BRISK** [Leutenegger et al., 2011] is based on a circular sampling pattern with different radii, inspired by the design pattern of the earlier **DAISY** descriptor [Winder et al., 2009]. The sampling pattern of **FREAK** [Alahi et al., 2012] is inspired by the strengths of the human visual system. Other works, such as **LDB** [Yang and Cheng, 2012] and **AKAZE** [Alcantarilla et al., 2013], extended

the intensity tests to contain grids instead of areas showing improved performance.

While [Calonder et al., 2010] claim that the intensity tests act as an approximation of the gradient difference, [Ziegler et al., 2012] suggest that its not the gradient approximation that contributes to the success of the intensity test based methods, but the fact that such methods are a locality sensitive hashing of full distances. This assumption is based on experiments showing that for 32×32 patches, using the full $\binom{32^2}{2}$-dimensional distance (formally known as Caley distance) performs poorly compared to **BRIEF** which is a random sampling of N tests out of the possible $\binom{32^2}{2}$.

2.3.2 Learnt binary feature descriptors

Inspired by the success of machine learning techniques in the real valued descriptor described in section 2.2.2, several authors identified learning based methods as a suitable tool to improve the discriminability of binary feature descriptors.

Oriented fast and Rotated Brief (**ORB**), introduced in [Rublee et al., 2011], was based on the simple idea that the intensity tests should not be sampled randomly out of the pool of all the available tests (*BRIEF*) or carefully selected based on a biological system (*FREAK*). Rather, they should be chosen such as to exhibit maximum variance across different samples, and minimum correlation in-between them. Note that identifying such tests, does not require pairs of labelled positive and negative patches. A similar idea was explored by [Fan et al., 2013], where the non-discriminative intensity tests were given lower weight in a weighted hamming distance computations. The **LATCH** descriptor [Levi and Hassner, 2016] extents this idea to sampling triplets instead of pairs, something that aims to improve the discriminative ability.

[Trzcinski and Lepetit, 2012] utilise pairs of positive and negative patches to learn modified signed binary features based on integral images, and optimised using the LDA principle similarly to [Winder and Brown, 2007]. Thus, specific box filters are learned to output binary results when applied to integral images, in a way that minimises the distance between positive pairs, and maximises the distance between negative pairs. The resulting **DBRIEF** however, performs poorly in image matching scenarios [Levi and Hassner, 2016], despite the fact that its the-

oretical foundations are sound, and that it exhibits strong performance in terms of differentiating positive from negative patch pairs.

Boosting was also shown to be very successful in learning a robust combination of filters applied to gradients. **BINBOOST**, the descriptor that was based on this idea and is described in [T. Trzcinski and Lepetit, 2013], can achieve state of the art results when evaluated in a similar settings to the training process. However, similarly to DBRIEF, it was shown that it does not generalise well in a matching scenario [Balntas et al., 2015, Tsun-Yi Yang and Chuang, 2016].

2.4 Conclusion

In this chapter, we proposed and discussed a categorisation of feature descriptors, and we identified some general trends in the field. Previous works show that learning based methods can achieve impressive results compared to engineered descriptors, however at a cost of both high training time and low run-time computational efficiency. In addition, convolutional feature descriptors which are currently the state of the art remain inefficient for most practical applications. This thesis introduces methods that aim to address these issues, and provide excellent discriminative power combined with improved efficiency.

Chapter 3

Per-patch adaptation of feature descriptors

In this chapter, we discuss a method that adapts the feature description and matching process to each individual patch. Such a method leads to improved performance, and can bridge the gap between the fast binary descriptors and more accurate pooling based methods. This chapter is organised as follows. In Section 3.1, we introduce the problem, and we then discuss some motivations in Section 3.2. In Section 3.3 we present the main methodology, and finally in Section 3.5 we present extended experimental evaluations that validate the performance and efficiency of the proposed method.

3.1 Introduction

The various feature descriptors reviewed in Chapter 2 differ in design, theory and implementation, but a common approach is the computation of the final feature vector from a fixed set of measurements applied to all described patches. It follows that the measurement process is not varied depending on the content of the patch. This is based on important practical considerations which primarily include convenience in using various distance metrics and efficient matching techniques for large scale problems. For example, it would not be possible to directly compare two binary strings $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ extracted from different sampling patterns of in-

tensity tests, since the corresponding dimensions of the two descriptors would be physically incompatible.

Moreover, learning based components are preferably trained offline as they are typically too computationally intensive for any online processing. This is even more important in applications such as tracking and matching, where speed is one of the main concerns. In the BRIEF descriptor [Calonder et al., 2010], five different arbitrarily designed configurations of binary tests were evaluated on a sub-sequence of the Oxford matching dataset [Mikolajczyk and Schmid, 2005] and the best performing configuration was selected. This can be viewed as a greedy selection of the global set of intensity tests that perform well.

The main idea presented in this chapter, is based on a hypothesis that different patch appearances can be best represented by different sets of measurements. For example, the results from [Tuytelaars and Schmid, 2007] show that recognition performance can be improved by adapting the spatial structure of SIFT-based descriptors to each class. Thus, it is also desirable to adapt the description extraction process to each individual patch, instead of using the same process across all different samples.

To that end, we propose an approach which combines the advantages of efficient binary descriptors with the improved performance of learning-based descriptors. We demonstrate that there is no single set of measurements that is globally optimal for all patches in a dataset and significant improvement can be gained by adapting the binary tests to the content of each patch. The measurements are first designed to maximise the inter-class distances and then a subset is selected online for each patch to minimise the intra-class distances. This concept is illustrated in Figure 3.1. The selection is done efficiently during matching by using a binary mask in such a way that the extraction time is comparable to other binary descriptors. In Figure 3.2, we show the matching framework of our mask descriptors, and we compare it with the typical feature descriptors. We bypass the fact that dimensional distance metrics can be used only when each dimension is based on a common feature representation, by utilising subsets of a larger set of features.

The proposed online adaptation of discriminative features per patch can be applied to other techniques such as decision trees or ferns. Nearest neighbour

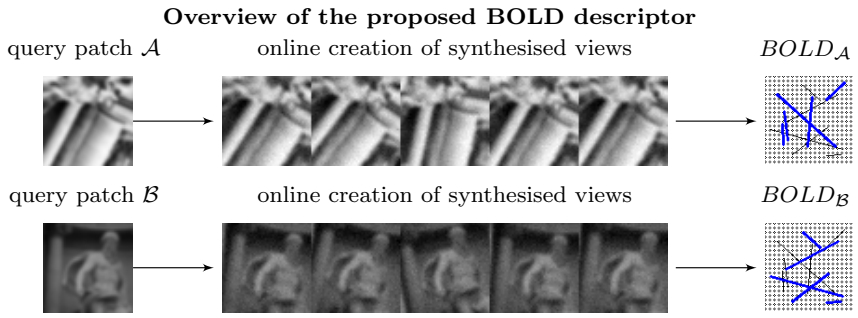


Figure 3.1: In contrast to typical approaches that use the same measurements for all patches, we adapt our Binary Online Learned Descriptor (BOLD) online to each patch. The blue intensity tests indicate the selected binary tests from a common super-set, based on the measurements from the synthesised views of each patch. Utilising a locally adapted subset from a common set of intensity tests, allows efficient sequential matching and common database storage.

matching of descriptors is also efficient by calculating our modified masked Hamming distance. We evaluate the proposed descriptor on different benchmarks and demonstrate that its performance matches that of SIFT, with computational efficiency that matches that of BRIEF.

3.2 Motivation [1]

In this section, we present the basic motivation behind our work. We first show that the subset of intensity tests included in a intensity test based descriptor can greatly alter its discriminative ability for specific queries. Secondly, we illustrate the instability of the binary intensity tests, and we show that it is related to the internal patch structure. Lastly, we present similar instability results in a tracking-by-detection based method, which is based on a classifier built on pixel-wise intensity tests.

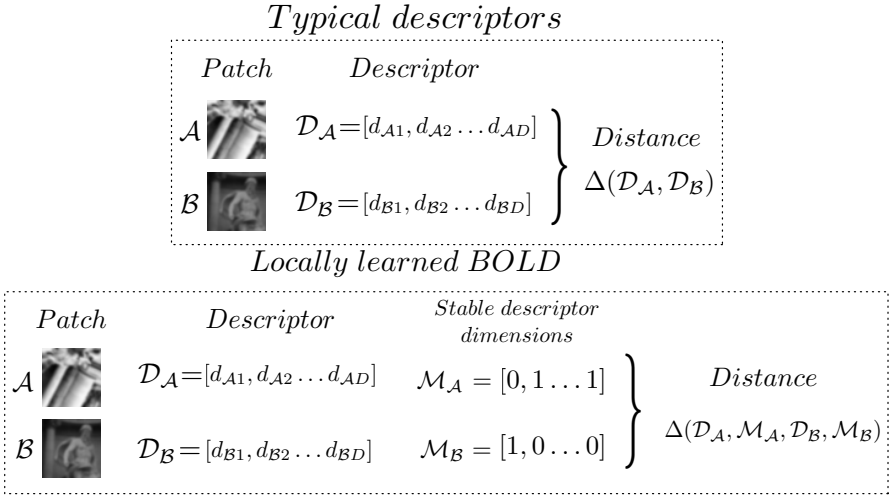


Figure 3.2: Distance computation with typical feature descriptors (top) and with our proposed locally adapted descriptors (bottom). Note that the final distance in our method incorporates a mask to indicate which features should contribute to the final result based on patch-specific invariance.

3.2.1 On the performance of random BRIEF descriptors

The space of all possible intensity tests for a patch of size is $N \times N$ is $\binom{N^2}{512}$. For instance, for a patch of size 32×32 it is approximately 5×10^{300}. The randomness in the creation process in BRIEF descriptors should, in theory, not be an important factor in terms of the final performance. In other words, since creating a random BRIEF descriptor includes setting a random seed to the pseudorandomness generation process, one would expect that this decision would not have great effect on the discriminative power of the final product. Altering the random seed changes the position of the pixel intensity tests, however the Gaussian distribution from which the tests are sampled remains intact. Note that the developers of `openCV` [Bradski, 2000] have arbitrarily set 42 as the seed in the test creation process [1].

Using the available BRIEF code from `OpenCV`, we create 4 different descriptors, by changing the random seed used in the creation process. This has the effect of altering the intensity tests that are included in each descriptor. Surprisingly, we find that altering this seed greatly alters the discriminative ability of each

[1] https://gist.github.com/vbalnt/

Table 3.1: Quantitative results for the experiment described in Figure 3.3. Note the significant variation in mAP and *sucess rate*.

descriptor	mAP	success rate %
$BRIEF_1$	0.831	61
$BRIEF_2$	0.835	59
$BRIEF_3$	0.811	65
$BRIEF_4$	0.842	58

configuration when considering individual patches.

Figure 3.3 shows the 4 BRIEF sampling patterns as well as positive and negative matches obtained with these descriptors. Although the descriptors have 512 intensity tests, only the first 50 are plotted for clarity. We form a set of 500 query patches, together with a true positive matching patch for each of the query patches, totalling a set of 1000 patches. For each patch, we find the nearest neighbour by Hamming distance brute-force search and compare with the ground truth. Table 3.1 shows the mean average precision and the success rate of matching 1000 patch pairs. By success rate we define the percentage of times that the retrieved nearest neighbour returned by brute force matching was the correct result. We can see that the results vary for different sets of binary tests. Also note that by choosing individually an appropriate BRIEF descriptor for each query patch, one could achieve 100% success rate for the subset of 15 queries shown in Figure 3.3. This demonstrates that careful sampling of the intensity tests per patch, instead of using a global set can lead to low distance for the positive matches and high distance for the negative ones thus improving the correct matching rate.

3.2.2 On the instability of pairwise intensity features

In this experiment, we illustrate the instability of the intensity test features even under very minimal visual changes. In Figure 3.4 we plot the distribution of the number of pairwise intensity tests that change their sign under a small rotation. For each patch, we create rotated versions of itself by rotating it by 5 & 10 degrees and we compare the extracted descriptors from the rotated versions to the descrip-

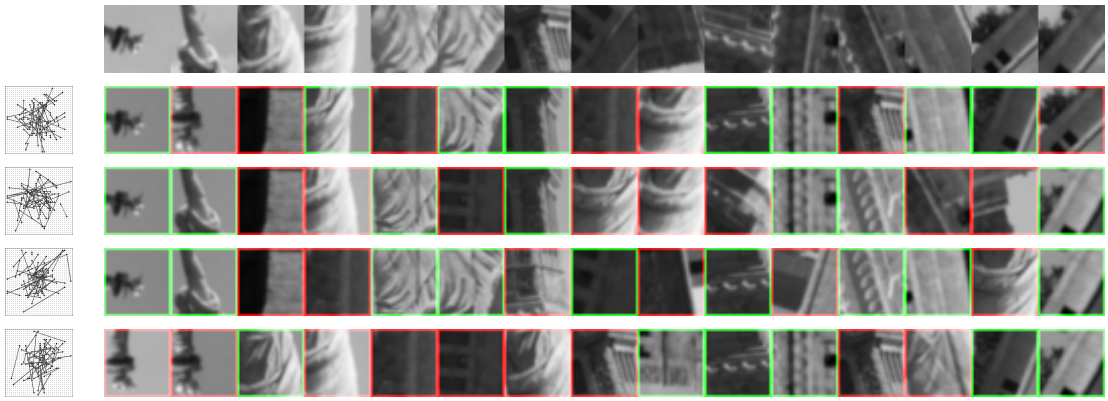


Figure 3.3: The effect of the randomness in the creation of the BRIEF intensity tests on the final query nearest neighbour matching performance. The top row of patches represents the query images, and the remaining 4 rows below show the retrieval results from the dataset, where in each row the respective BRIEF descriptor is used. The true matching positives results are shown in green, and the false positives are shown in red. It is clear that different BRIEF versions, are more discriminative than others for different queries and also make different mistakes. Thus, choosing the right BRIEF descriptor for a given query will give rise to a more robust matching method.

tor extracted from the original version in terms of hamming distance. Surprisingly, even for 5^o rotation which presents no significant visually change in the appearance of the patch (see Table 3.2), we observe patches where up to 30 out of 512 binary tests flip signs. With 10^o rotation, we note cases where 40% of the binary tests flip sign. We identify this instability as the main problem of the family of BRIEF-like binary descriptors, since such small transformations are frequent in real world applications and are almost certain to be observed in all real applications, e.g. due to the non-perfect nature of feature detectors.

In Table 3.2 we present some extreme cases of robustness and instability for specific patch pairs. In the first row, we show patch pairs where no binary bits flip sign, thus their Hamming distance is 0. In the bottom row, we show cases where 10% of the binary tests flip signs between the two patches of the pair. Note that in these cases, the rotation is only (1^o). The inability of the intensity tests to limit the Hamming distance to very low values, demonstrates their very sensitive nature. Interestingly, we can observe that patch pairs where the intensity tests do not perform well are high frequency patches rich in texture and edges, something that makes it difficult for the simple intensity tests to encode.

In Figure 3.4 we plot the Distribution of Hamming distances when comparing a patch with minimal rotated versions of itself (5^o and 10^o). For comparison, we also plot intra and inter-class distributions for real positive and negative pairs that a exhibit a large set of deformations such as illumination changes and affine transformations. Surprisingly, we can see that a small rotation of 10^o gives rise to a distribution with similar mean value to the real world-deformations.

3.2.3 Tracking performance with intensity features

Due to their efficiency, pairwise intensity tests are often exploited in the context of real time tracking or object detection. In a video with a moving object, small affine transformations are very frequent. We consider this application to further demonstrate the stability issue of the intensity tests. The Tracking-Learning-Detection approach [Kalal et al., 2012] uses an online learnt object detector based on a randomised fern classifier with a set of intensity tests as measurements. The classifier is an essential part of the system that allows re-detection of the object

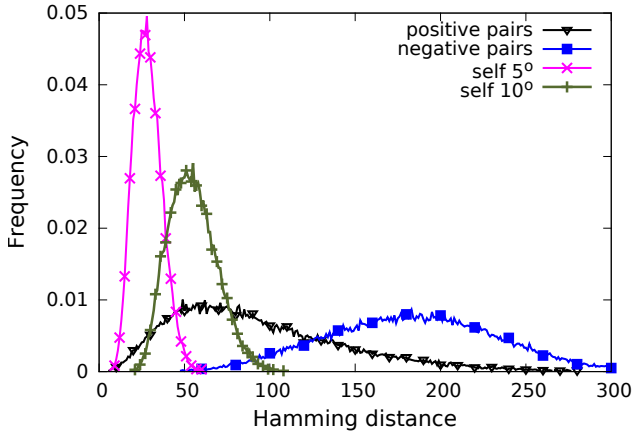
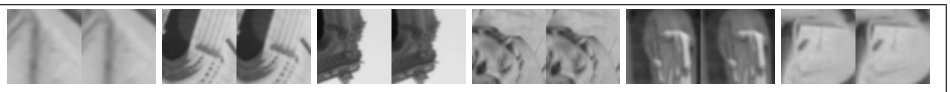
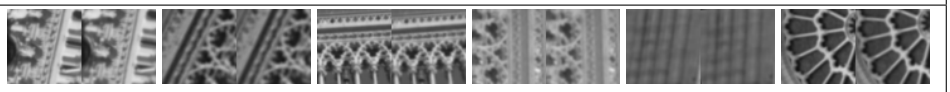


Figure 3.4: Distribution of Hamming distances when comparing a patch with minimal rotated versions of itself (5^o and 10^o). Note that even a 5^o rotation can lead to a large number of bit flips.

Table 3.2: Extreme cases of test (in)stability. Top row contains patch pairs where no bit flips occur with 1^o rotation. Bottom row contains pairs where 10% of intensity tests flip. It is clear that the more complex patches with rich structure are more sensible.

0% flips	
10% flips	

when the tracker drifts or object temporarily disappears. Such fern based keypoint classifiers were studied in Özuysal et al. [2010], and have proved to be an accurate yet very efficient method to recognise keypoints.

In Figure 3.5 we show how the changes in the intensity tests used to form the classifier, can have significant impact on the tracking results. To evaluate that, we used different seeds in the random test initialization [2]. The results are surprising, as they show $\approx 10\%$ performance change between the original code which uses the

[2]We used the original TLD implementation https://github.com/zk00006/OpenTLD

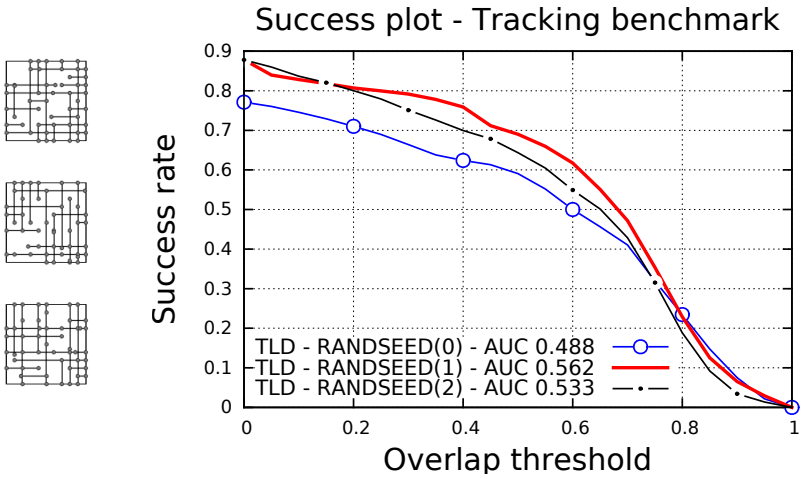


Figure 3.5: The effect of altering the intensity tests used in the online learned detector of TLD [Kalal et al., 2012]. (Left) The three different fern classifiers we used in our experiments. (Right) Results using the benchmark of [Wu et al., 2013]. Note the varying performance of the fern classifiers based on different intensity tests.

seed 0, and a different seed e.g. 1.

From the experiments presented above it is clear that a method that can adapt the features included in the descriptor measurements, can greatly improve the discriminative ability of the final descriptor. We present such a method below, together with experiments to show that it outperforms global non-adapted cases.

3.3 Local adaptation of binary descriptors

In this section, we discuss the theoretical justifications of our work, and we propose two different ways to adapt locally a global set of binary features to each individual patch. The first method is inspired by the optimisation of the ratio of intra to inter-class distances, and the second method is based on utilizing a special subset of features for each patch that remain stable under view deformations.

3.3.1 Locally adapted descriptors

Let $\mathbf{f}_{\mathcal{L}}, \mathbf{f}_{\mathcal{R}} \in \{0,1\}^D$ represent binary descriptor extracted from patches $\mathcal{L}, \mathcal{R}$ using D binary tests. Patches $\mathbf{f}_{\mathcal{L}}, \mathbf{f}_{\mathcal{R}}$ are from the same class (e.g. they represent the same interest point from two different views). The hamming distance is then defined as

$$\mathbb{H}(\mathbf{f}_{\mathcal{L}}, \mathbf{f}_{\mathcal{R}}) = \frac{1}{D} \sum_{i=1}^{D} |\mathbf{f}_{\mathcal{L},i} - \mathbf{f}_{\mathcal{R},i}| \tag{3.1}$$

Our goal is to identify the unstable bits in $\mathbf{f}_{\mathcal{L}}$ and $\mathbf{f}_{\mathcal{R}}$. Once this is done we can associate binary masks $\mathbf{m}_{\mathcal{L}}, \mathbf{m}_{\mathcal{R}} \in \{0,1\}^D$ with $\mathbf{f}_{\mathcal{L}}, \mathbf{f}_{\mathcal{R}}$ respectively, to suppress the contribution from unstable bits during Hamming distance calculation

$$\begin{aligned} \mathbb{H}_m(\mathbf{f}_{\mathcal{L}}, \mathbf{f}_{\mathcal{R}}, \mathbf{m}_{\mathcal{L}}, \mathbf{m}_{\mathcal{R}}) = &\sum_{i=1}^{D} \mathbf{m}_{\mathcal{L},i} \wedge |\mathbf{f}_{\mathcal{L},i} - \mathbf{f}_{\mathcal{R},i}| + \\ &\sum_{i=1}^{D} \mathbf{m}_{\mathcal{R},i} \wedge |\mathbf{f}_{\mathcal{L},i} - \mathbf{f}_{\mathcal{R},i}| \end{aligned} \tag{3.2}$$

The dimensions that are suppressed in both masks do not contribute to the final Hamming distance. Subsequently, the ℓ_0-"norm" of the combined masks $||\mathbf{f}||_0 = \sum_{n=1}^{D} (\mathbf{m}_{\mathcal{L},i} \vee \mathbf{m}_{\mathcal{R},i})$ indicates the final dimensionality of the masked descriptors for patches $\mathcal{L}$ and $\mathcal{R}$. Note that the masks are adapted independently to each patch hence the dimensionality can differ for different pairs. We term the dimensions that are included in the mask $\mathbf{m}_{\mathcal{P}}$ for a patch $\mathcal{P}$ as *stable dimensions*.

The remaining task is to identify stable dimensions of a given a patch $\mathcal{P}$, that can be included in mask $\mathbf{m}_{\mathcal{P}}$. We first explore a technique inspired by Linear Discriminant Analysis (LDA) based on covariance of inter and intra class features.

3.3.2 Learning discriminative descriptors

It has been frequently demonstrated that descriptors perform better when the ratio of the intra- and inter-class distances is maximised. Given a set of labelled matching and non-matching image patches, methods such as [Cai et al., 2011, Winder and Brown, 2007] seek to find a projection $\mathbf{w}^*$ s.t. $\mathbf{w}^* = \arg\max_{\mathbf{w}} (\mathbf{w}^T \mathbf{A} \mathbf{w})/(\mathbf{w}^T \mathbf{B} \mathbf{w})$

which is the ratio of the inter **A** to intra-class **B** covariance along the direction **w**. Intuitively, such methods minimise the expected distance between patches annotated as similar and maximise the expected distance between patches annotated as dissimilar. This has been done globally for real-valued descriptors in [Cai et al., 2011, Winder and Brown, 2007, Trzcinski and Lepetit, 2012] with the use of a large set of negative and positive pairs of patches in an offline learning process.

In the following we propose an approach that exploits this idea to optimise a binary descriptor for each patch independently.

3.3.3 Properties of binary tests

Let I represent the intensity image, and $\boldsymbol{t}$ represent a specific location on the image defined by the row and column values. Features (intensity tests) $\mathbf{f}_m, i = \{I(\boldsymbol{t}_1) > I(\boldsymbol{t}_2)\}_i$ are binary tests that consist of comparing pixel intensities in pairs of locations $\mathbf{t_1}$ and $\mathbf{t_2}$ within the patch. For a grid of $P \times P$ locations within a patch, the total number of tests is $M = \binom{P^2}{2}$. The locations are typically generated randomly but further constraints on how tests are generated can be introduced. These may include only horizontal and vertical pairs or exclude locations on patch boundaries, large distances between $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ etc.

Let $\{\mathbf{f}_1, \mathbf{f}_2, \cdots \mathbf{f}_N\}$ denote a set of binary descriptors of dimensionality D, extracted from N patches which can be arranged in a matrix $\mathbf{F}$ of size $N \times D$. Each column c_i with $i \in [1, ...D]$ represents a test/dimension of the binary descriptors and can be viewed as a binary string of length N that follows a Bernoulli distribution with a certain probability of values 1 or 0. Matrix $\mathbf{F}$ can then be expressed as the outcome of N trials of D Bernoulli distributions $\mathcal{B}_i$. If the mean value of $\mathcal{B}_i$ is ρ_i, then the variance is $\sigma_i = \rho_i(1 - \rho_i)$ where ρ_i is the ratio of `1s` and $(1 - \rho_i)$ is the ratio of `0s` in column c_i. Variance σ_i of the i^{th} dimension has a direct relation with the Shannon entropy of the binary string of the corresponding column c_i i.e. $\mathcal{E}_i = -\rho_i \cdot log_2\rho_i - (1 - \rho_i)log_2(1 - \rho_i)$.

A required characteristic of such binary strings is to exhibit a high variance–entropy values if descriptors $\mathbf{F}$ belong to different classes and a low variance–entropy values if descriptors belong to the same class. For the former, the discriminative dimensions are the ones where the variance reaches the maximum possible

value of 0.25 (entropy reaches 1). The latter implies that the process that generates the values for this specific descriptor dimension, is stable and robust to noise, deformations, illumination changes etc. In an ideal case, with a perfect descriptor all columns of intra class descriptors $\mathbf{F}$ would have entropy and variance equal to zero. Given $\mathbf{F}$ and Bernoulli distributions $\mathcal{B}_i(\rho_i, \sigma_i)$ associated with test/dimension i of $\mathbf{F}$, the expected average distance $\mathbb{E}[\Delta]$ between descriptors in $\mathbf{F}$ is related to the sum of the variances σ_i. This can be derived from:

$$\mathbb{E}[\Delta_{intra}] = \frac{1}{D}\sum_{i=1}^{D}\mathbb{E}[\Delta_i] \tag{3.3}$$

where Δ_i is the intra-class distance when taking into consideration only the i^{th} dimension, and $\mathbb{E}[\Delta_i]$ is the expected intra-class distance value for dimension i:

$$\mathbb{E}[\Delta_i] = \frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}|\mathbf{f}_{m,i} - \mathbf{f}_{n,i}|_{\oplus} \tag{3.4}$$

and $|\mathbf{f}_{m,i} - x_{n,j}|_{\oplus}$ is the Hamming distance between two binary values. Since $|\mathbf{f}_{m,i} - \mathbf{f}_{n,j}|_{\oplus} = (\mathbf{f}_{m,i} - \mathbf{f}_{n,j})^2$ we obtain:

$$\begin{aligned}\mathbb{E}[\Delta_i] = \frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}\mathbf{f}_{m,i}^2 - 2\frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}\mathbf{f}_{m,i}\mathbf{f}_{n,i} \\ +\frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}\mathbf{f}_{n,i}^2 = 2\mathbb{E}[\mathbf{f}_i^2] - 2\mathbb{E}[\mathbf{f}_i]^2\end{aligned} \tag{3.5}$$

The variance of dimension i is therefore directly reflected by the fraction of 1s in column i of matrix $\mathbf{F}$. From the above it is clear that dimensions with high variance increase the intra-class distances, and dimensions with low variance decrease it. Low variance is required for descriptors from the same class (positive patches) and high variance for descriptors from different classes (negative patches).

It was demonstrated in [Cai et al., 2011] that discriminant projections of SIFT dimensions can be achieved in a two stage process which first diagonalises the intra-class covariance and then performs a global PCA. Thus the dimensions are decorrelated and oriented along dominant directions in the real-valued space. This process can be adapted to learning of discriminative binary descriptors by first

selecting uncorrelated tests/dimensions that maximise the inter-class distances globally and then by short-listing tests that minimise the intra-class distances locally. Correlation $\mathbb{C}_{ij}$ between tests i and j can be measured on inter-class patches by the Hamming distance between the corresponding columns:

$$\mathbb{C}_{ij} = |\frac{2}{N}\sum_{m=1}^{N} |\mathbf{f}_{m,i} - \mathbf{f}_{m,j}|_{\oplus} - 1| \tag{3.6}$$

Thus the value of $\mathbb{C}_{ij}$ varies between 0 and 1, with 1 for perfectly correlated tests. Suitable dimensions can be chosen by thresholding this measure.

The first two steps of the process, the global selection of discriminative dimensions and the decorrelation can be done offline from a large set of possible binary tests and random patches. The final selection of dimensions that minimise the intra-class variance has to be done per patch and online, which requires an efficient implementation.

3.3.4 Efficient extraction of online learned descriptors

In this section we present the technical details of our online learned descriptor. This is done in two steps, namely inter-class offline optimisation and intra-class online selection of tests.

Global optimisation

In global optimisation the goal is to identify the subset of discriminative features leading to maximization of inter-class distances. This can be done offline on a large set of N diverse image patches different from the test data. In the case of binary tests, it consists of finding features that give a large variance across inter-class examples as discussed in section 3.3.3. It requires calculation of all test responses in each of the N patches. This results in a set of N binary strings of dimensionality M with $\mathbf{f}_n$ representing the bitstring of patch n. $\mathbf{F}$ is a matrix with descriptors $\mathbf{f}_n$ as rows. We then calculate the fraction of 1s in column i of $\mathbf{F}$ and sort the columns according to that measure. This ranks the discriminative tests in relation to their variance across a random set of inputs.

The next step is to select a subset of uncorrelated features. We follow the greedy approach from [Rublee et al., 2011] which starts by selecting the first high variance tests from the ranked list and then searches for another high variance test with the correlation score $\mathbb{C}_{ij} < \tau_C$ (eg $\tau_C = 0.2$). The process continues by verifying at each iteration the correlation between the candidate and all selected tests. The selection stops when a defined number of G tests have been found (eg $G = 512$).

Note that the global optimisation is done offline as it is concerned with the whole set of possible tests and diverse image patches that represent negative examples in section 3.3.3.

Local online learning

In this section, we discuss the method to adapt the descriptor on each individual patch, resulting to our proposed Binary Online Learned Descriptor (BOLD).

As demonstrated in [T. Trzcinski and Lepetit, 2013, Rublee et al., 2011] a set of globally optimised tests outperform a set of random tests in terms of matching error rates. However, to fully benefit from the LDA-like optimisation, intra-class distances have to be minimised. As we show in Figure 3.1 different subsets of tests minimise the intra-class distances for individual classes of patches and can achieve superior performance compared to the globally optimised features.

We consider each patch as a separate class, therefore in many applications this optimisation has to be performed online during descriptor extraction. Given that a patch is a single instance from a class, additional examples have to be synthetically generated to estimate intra-class variance $\mathbb{E}[\Delta_i]$. This approach has proved successful in many applications, in particular in the context of local image patches, affine projections are typically applied [Cai et al., 2011, Özuysal et al., 2010].

Generating various geometric views of the same patch can be done easily (e.g. with affine matrices and bilinear interpolation), but in large datasets or real time applications the computational complexity would grow significantly. However, given the globally optimised set of binary tests, which is of a limited size, instead of bilinear patch warping we can apply the geometric transformations directly to

the pixel locations $(\mathbf{t_1}, \mathbf{t_2})_i$ of each test $\mathbf{f}_i$. For each test, a new set of test can be created, which consist of its affine-transformed versions. Furthermore, since the set of tests is fixed, the locations of tests under various affine transformations can be stored in a lookup table rather than calculated online. Thus, our set of tests is extended $\mathbf{f}_{ia} = \{I(\mathbf{t_1}) > I(\mathbf{t_2})\}_{ia}$ where a indicates an affine transformation of test $\mathbf{f}_i$.

Given the binary strings generated by tests $\mathbf{f}_{ia}$ represented in intra-class matrix $\mathbf{F}$, a subset of tests that minimises the variance along dimension a is selected. In our implementation we select only the tests for which the variance is 0. However more complex methods can be applied, such as variance sorting and thresholding. Having identified the sets that are to be included in the BOLD descriptor, each patch is represented by the results $\mathbf{f}_n$ of the adapted binary tests and a second binary string $\mathbf{m}_n$ of length D where 1s indicate which tests are stable dimensions for patch n. Thus, the number of 1s (e.g. $\sum_{i=1}^{D} \mathbf{m}_{n,i}$) may differ for every patch. This can be adressed, by introducing a normalisation term that divides the distances with the number of stable dimensions.

3.4 Analysis of online learnt descriptor

In this section we analyse the properties of the proposed descriptor and investigate various implementation options. We first discuss the relations to some hashing methods. We then investigate the parameters of transformations suitable for generating intraclass examples as well as alternative implementations of the descriptor. The experiments are done, unless stated otherwise, by using 100k Trevi data for globally optimising the tests and 100k Liberty data for testing the descriptors, from the Photo Tourism dataset [Winder et al., 2009].

3.4.1 Binary codes in related areas

Biometrics. We can also relate our approach with previous works in the field of biometrics, particularly iris recognition. A similar binarisation technique is used to encode the image of iris as a string of bits. It has been known from the research

in this area that "not all bits in the iris code are equally likely to flip" [Bolle et al., 2004]. In the context of biometrics, several images from the same eye are used in order to identify the *fragile bits*. These bits are likely to change value across the training dataset and they may be different for every individual. The distance measure is therefore weighted in terms of how fragile a bit is for a particular individual from the training data. Note that as demonstrated in Figure 3.1, stable feature dimensions change per query, similarly to fragile bits in the iris codes. It was demonstrated in [Hollingsworth et al., 2009] that the false negative recognition rate improves by splitting the iris code into two groups, one that includes the bits that flip with high probability and the other group with low probability flip. The bits from the latter one are then used to model the iris. Our approach acts in a similar manner and by creating synthetic positive examples thus empirically identifying the bits that flip.

Binary hashing. The feature extraction in BRIEF and similar methods can be considered as closely related to binary hashing functions [Zhang et al., 2013] where given $\mathbf{x}$ as the input observation, and D intensity tests, each test can be considered a hash function $f_k(\mathbf{x}, \mathbf{i_k}, \mathbf{j_k}) = \mathbf{x}(\mathbf{i_k}) - \mathbf{x}(\mathbf{j_k})$. The thresholded binarization of f_k is then performed according to $f_k < T_k$, which provides a cut-off threshold to make it more robust to noise. However, all BRIEF like methods use $T_k = 0$.

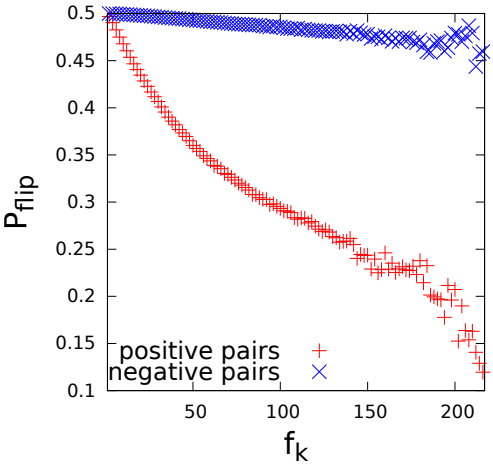
Based on the above formulation, we can examine if the hypothesis proposed in [Zhang et al., 2013] that correlates the value of $|f_k - T_k|$ with the probability of hash bit flip holds in the binary feature descriptors. Note that the hash functions are typically linear projections from higher dimensional space in contrast to our simple intensity tests. The ability of a hash function f_k to map similar data points to the same bit (0/1) is their discriminating power which can be modelled as the probability of similar data points being mapped to the same output bit by a single hash method f_k. The bits in Hamming distance are then weighted according to this probability. However, in the case of intensity tests, and for typical deformations such as rotations and translations, their discriminating power is much lower and the probability of a flip for a specific patch and its bits is less reliable. Thus, we expect that the distance from the threshold will not have a significant impact on improving the stability of the tests.

To test this assumption, we perform the following experiment. Following the formulation of [Zhang et al., 2013], we hypothesise that for a given patch p, intensity tests with high values of $|f_k - T_k|$ are less stable than tests with low values of $|f_k - T_k|$. For brevity, we use the term Threshold Distance (TD) for the term $|f_k - T_k|$. To that end, we split the original $D - dimensional$ binary descriptor into two groups, according to their TD values. First we identify the median TD_m, and then we create two masks termed low-TD and high-TD groups. A test t_k belongs to the low-TD or high-TD group if $|f_k - T_k| < TD_m$ or $|f_k - T_k| > TD_m$ respectively.

Were the hypothesis true, we would expect that the low-TD group would perform better than the high-TD group, since the intensity tests belonging there, would be much more stable. In Figure 3.6 we report the results in the $100K$ patches from the *notredame* dataset, in terms of FPR95 values. We can see that indeed using only the low-TD tests for computing the descriptor, results in a much less discriminative descriptor than using all the binary tests. However, we note unlike what is reported for the commonly used high-dimensional hashing methods in [Zhang et al., 2013], there is no relation between a high-TD value and the stability of an intensity test, for real world patch deformations such as the ones found in the *notredame* dataset [Winder and Brown, 2007]. This is because the descriptor computed with the high-TD tests, cannot manage to outperform the global parent descriptor. In contrast, when the masking is based on affine deformations, such as the ones commonly found in the *notredame* dataset, we see that the performance is significantly better, resulting in 53% improvement in terms of discriminative power.

The above experiment also indicates that the random noise is not the main issue in patch matching. Intuitively, patches are blurred before sampling therefore Gaussian noise is minimised and the probability of bit flip is less related to the magnitude of the intensity difference. The viewpoint change, rotation and other geometric transformations are the main factors affecting the matching performance.

It is also worth noting, that the f_k statistics from different intensity tests significantly vary in terms of their mean values and standard deviations. In Figure 3.7,



method	FPR95
low-TD	61.17
high-TD	43.81
random	43.86
affine	28.87

Figure 3.6: (left) Relation between $P_k(flip)$ and TD_k (right) Error rates for different methods of choosing subsets of hash functions (intensity tests).

we plot the f_k distributions for two different intensity tests. We see that both intra and inter distributions follow a Gaussian distribution and differ in terms of the overlap. In Figure 3.7 (left) the centres of the intra and inter distributions are close which may result in frequent switch of the bit from binarised intensity test due to noise. In contrast distributions in the right are further apart and thus more robust to noise.

Figure 3.8 (top) shows the distribution of intra and inter class distances for 512 globally optimised tests. Positive patch pairs from Yosemite dataset represent intra-class and negative pairs correspond to inter-class. The selected tests exhibit high variance across negative patch pairs and small correlation $\mathbb{C}_{ij}$ between tests (eg < 0.2). In contrast, Figure 3.8 (bottom) shows distance distributions for our locally optimised tests, where is each patch was described by a different subset of tests from the globally optimised set. The intersection between the distributions for globally optimised tests is 13.95% and for patch adapted ones is 9.75% which corresponds to 30% of relative improvement.

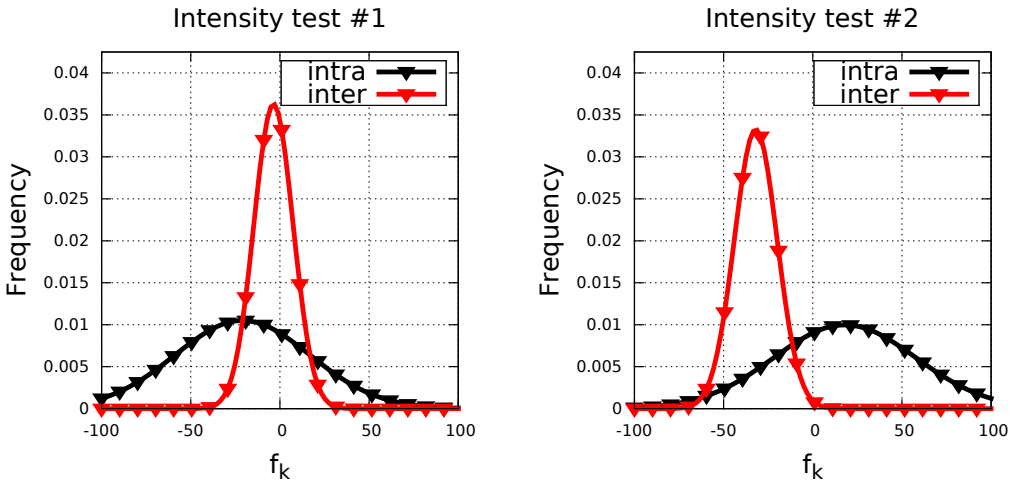


Figure 3.7: Distribution of f_k in the case of intra and inter class variations for two different intensity tests.

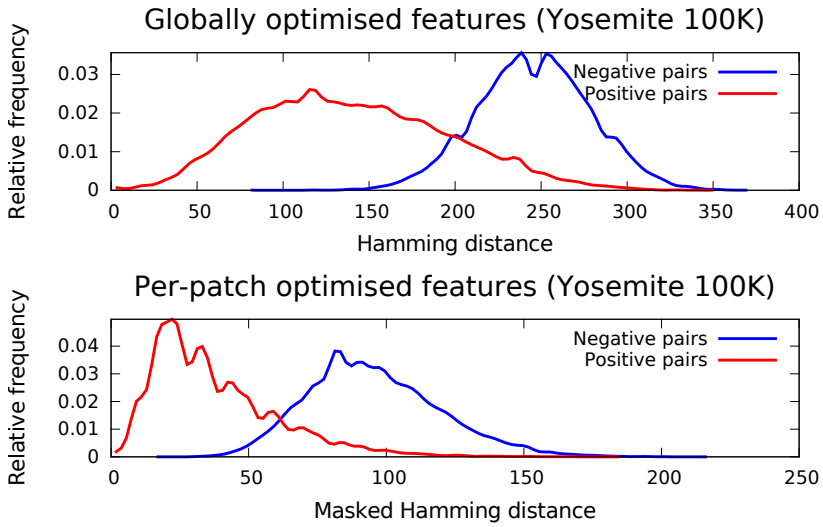


Figure 3.8: Negative (inter-class) and positive (intra-class) distance distribution of globally (top) and locally (bottom) optimised descriptors. The intersection area between the two distributions is reduced from 13.97% to 9.75% for globally vs. locally optimised descriptors. Thus the number of mismatches is reduced.

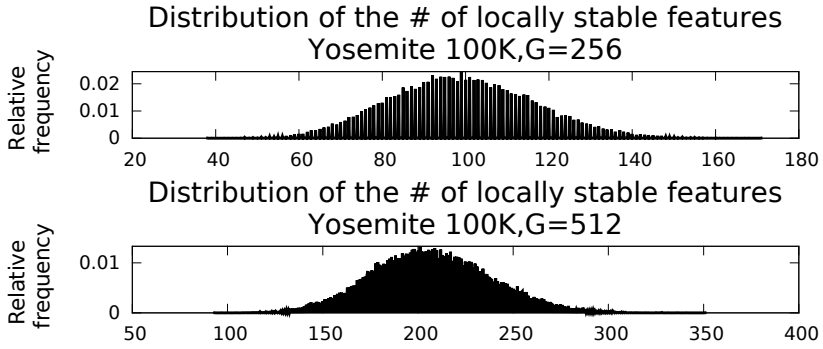


Figure 3.9: Histograms of descriptor dimensionality after the online selection of locally stable tests using a single synthesised positive rotated example. G denotes the dimensionality of the globally optimised feature set.

3.4.2 Intra class adaptation

Modelling intra-class distribution is crucial for successful selection of stable dimensions. The intra-class patches are generated with common affine deformations such as scaling, translation and rotation. We therefore investigate the effect of different sets for intra-class optimisation.

Number of stable dimensions. Our proposed online adaptation approach relies on the assumption that patches inside a positive pair should match will have a very similar number of dimensions indicated by the masks. Furthermore, since the number of selected dimensions may vary we investigate the extent of this variation. Figure 3.10 shows the histogram of stable dimensions for positive pairs (left) and negative pairs (right). The majority of positive pairs have a similar number of stable tests ranging from 60 to 80 while patches in negative pairs have a much broader distribution. i.e. the number of dimensions is significantly different.

Figure 3.9 shows the histograms of descriptor dimensionality after online selection of stable tests. G denotes the number of globally optimised tests. We can observe that for $G = 256$, the average number of locally stable tests is ≈ 100 and for $G = 512$ it is ≈ 200, which is approximately half of G. This shows that for each patch, only approximately half of the binary tests are robust to simple affine deformations.

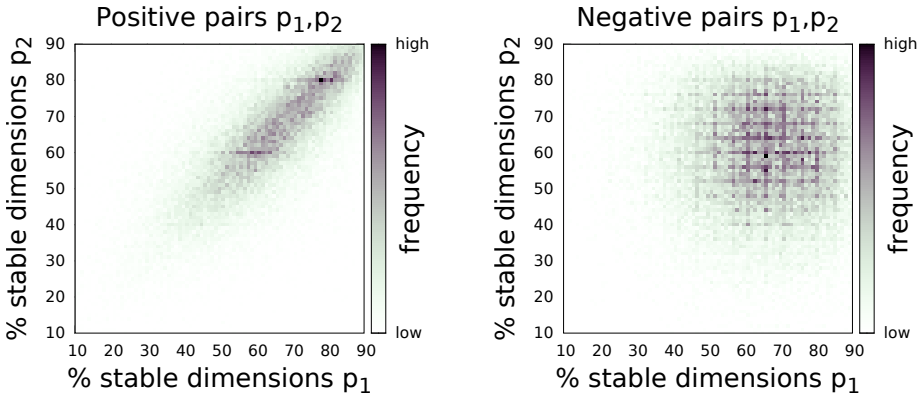


Figure 3.10: 2D histogram of numbers of stable dimensions for positive and negative pairs. The distribution is more compact and the numbers of stable dimensions are more similar for positive pairs than for the negative ones. This clearly shows that patches from the same class, have similar (although not identical) stable dimensions masks. In addition, this implies that the masks themselves can be used as a form of feature descriptor (albeit not a significantly discriminative one).

Robustness to affine transformations. Intuitively, since descriptors based on intensity tests such as BRIEF, BRISK, ORB are not scale, translation or rotation invariant, the percentage of binary test that remain stable is inversely proportional to the extent of the transformations. We experimentally quantify this by creating transformed views of 10k patches from the *notredame* dataset and measuring the number of dimensions that change bits after patch deformation while increasing transformation parameters. The results are presented in Figure 3.11. We observe that 90% of tests are stable for very small transformations i.e. up to 5 degree rotation, 1.05 scaling or 2 pixel translation. These are minor deformations that are easily exceeded in real applications. Typical keypoint detectors introduce larger error in its location and scale estimation. Also the orientation estimation methods often used quantization bins of 10 degrees. The results show that nearly 50% of tests fail with translation by 5 pixels, scaling of 1.15 or rotation of 30 degrees. Descriptors such as SIFT were engineered to be robust to such deformations but pairwise intensity tests are more sensitive.

Online learning with patch transformations. Figure 3.12 shows the 95%

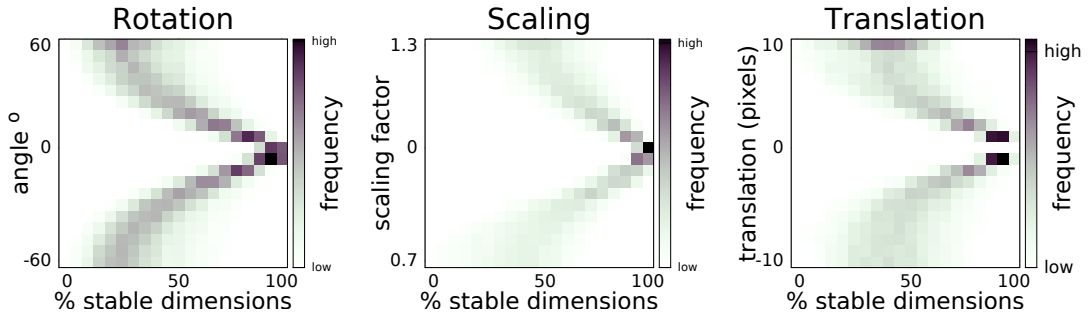


Figure 3.11: The distribution of percentages of stable dimensions in the mask across different magnitudes of rotations translations and scalings. As expected, the more severe the deformation is, the less dimensions survive as being stable, and flip binary sign.

error rate for descriptors with stable dimensions selected based on 2, 4, 8, or 16 synthetically generated patches with different rotation angles including the original one. The smallest error is given by tests selected with patches transformed with up to 20 degree rotation. This may be related to the error that is typically introduced with orientation estimation within SIFT patch rectification. The second observation is that the error is not significantly dependent on the number of patches used to model intra class variations. The results show that the identification of the stable dimensions can be done with as few as two examples. In fact, in our experiments, we also found that even a single synthetic example with a minimal rotation of 10^o, can lead to good results. In that case, the mask is defined as $\mathbf{m} = \neg(\mathbf{f} \oplus \mathbf{f}')$, with $\mathbf{f}'$ being the transformed patch of query $\mathbf{f}$. This is an important observation, since using a single perturbed version of the input patch to identify the stable dimensions significantly reduces the complexity of online extraction of the descriptor. This avoids the need for more complex methods such as thresholding and full ranking of variances.

3.4.3 Descriptor variants

Our binary online learnt descriptor consists of a descriptor string $\mathbf{f}$ and mask $\mathbf{m}$. This doubles the number of bits a patch is represented with. To demonstrate that the improvements are due to our masking framework and not due to the increased

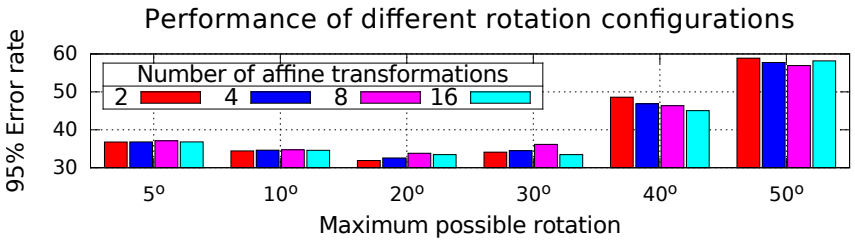


Figure 3.12: 95% matching error rate for Yosemite 100k with respect to various magnitude of rotations and number of synthetically rotated examples for the computation of the intra-class matrix, and the identification of the stable dimensions. Small rotations and few examples are sufficient to achieve low error rate.

memory requirement we perform additional experiments. We double the size of the original BRIEF to 1024 bits and compare with our descriptor that consists of 512+512 bits including the mask. Note that these two descriptors have exactly the same memory footprint. Figure 3.13 (left) shows the ROC curves for matching the 100k Liberty data. The descriptor combined with the mask improves upon 1024 bit BRIEF by up to 5%. This shows that masking out unstable bits reduces the intra-class variations, and the extra gain in performance does not come from the extra information used in the hamming distance computation.

Another approach to suppress the noisy dimensions is to zero the unstable bits directly in the descriptor instead of using a mask. This results in the variant denoted with `512-descr&mask` in Figure 3.13 (left). Note that this configuration also performs better than both the 512 and 1024 versions of BRIEF, although it does not need to store any extra information for the mask bits. Interestingly it is only slightly worse in terms of performance than the full mask equivalent, which needs to store double the amount of information. This implies that this variant can be used when memory is more important than discriminative power.

When comparing two descriptors $\mathbf{f}_L$ and $\mathbf{f}_R$ with their respective masks $\mathbf{m}_L$ and $\mathbf{m}_R$, there exist three possible ways to generate a masked distance: $\{$ with $\mathbf{m}_L$, with $\mathbf{m}_R$, with both $\mathbf{m}_L, \mathbf{m}_R\}$. Note that only the last option is using online adaptation for both patches, the two uni-lateral options are asymmetric. The comparison for

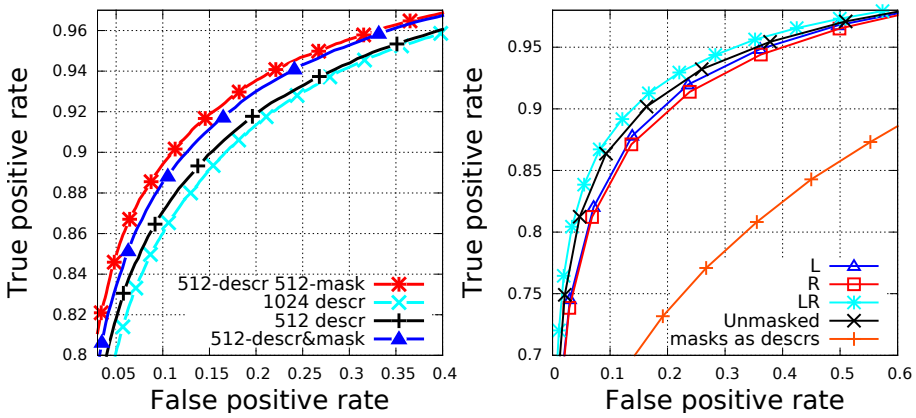


Figure 3.13: (left) Matching performance for variants of descriptor and distance measures (right) Symmetric and asymmetric distance variants.

these approaches including the unmasked descriptor is presented in Figure 3.13 (right). The asymmetric distances with single mask only give noticeably lower scores than the unmasked descriptor, with the one using both masks obtaining the top score.

Interestingly, since the mask is learnt for every patch it can be considered a characteristic of the patch and used as a descriptor on its own. Figure 3.13 (right) shows the performance for matching masks compared to random scores. Masks are scoring below the full descriptor as they only carry information about which tests are stable and not the actual value of the test but still significantly higher than a random classifier which would perform along the $x = y$ line.

3.5 Experimental evaluation

In this section we evaluate our descriptor and compare to other state-of-the-art methods in terms of patch pair classification, patch matching and tracking. In addition, we show how our method can be successfully applied to other binary descriptors, and even floating point cases. Lastly, we provide evaluations to indicate the computational performance of our method.

3.5.1 Patches

We first evaluate the performance in terms of matching accuracy in distinguishing positive from negative patch pairs on the Photo Tour dataset Winder and Brown [2007]. This dataset consists of three subsets *Liberty*, *Yosemite* & *Notredame* containing more than 500k patch pairs extracted around specific feature points. We follow the protocol proposed by the authors where the ROC curve is generated by thesholding the distance scores between patch pairs. The number reported here is the false positive rate at 95% true positive rate. For the evaluation we use the $100K$ patch pairs proposed by the authors which are resised to 32×32. For SURF, BRIEF, BinBoost, and DBRIEF, the original implementations provided by the authors were used. For ORB, we use the set of 256 binary tests that are included in OpenCV. For SIFT, we use the implementation from VLFeat.

In Figure 3.14 (top) we plot the ROC curves for the full set of the globally optimised binary features of 512 bits compared to the per-patch optimised subsets of our proposed BOLD descriptor. Our method outperforms the global set of features for all false positive rates. This is significant, since it shows a clear advantage of per-patch optimisations compared to global per-dataset optimisations. It has to be noted, that although the final BOLD descriptor has significantly less dimensions involved in the computation of the distances and it is always a strict subset of the globally optimised tests, it outperforms the parent superset of feature dimensions.

In Figure 3.14 (bottom), we present the results of the comparison between our descriptor and other widely used descriptors such as BinBoost, SIFT, SURF, ORB, DBRIEF, and BRIEF. It is important to note that out of the best performing descriptors i.e. BinBoost, SIFT and BOLD, our descriptor is the only one to use simple binary intensity tests. Both SIFT and BinBoost use quantised gradient responses which capture significantly more information about the patch statistics. Recently, in [T. Trzcinski and Lepetit, 2013] it was shown that intensity binary tests are less effective as descriptor dimensions compared to features based on quantised gradients when optimised globally with the same theoretical framework. Our results show however that their performance can be greatly improved by simply using our online per-patch adaptation framework. Thus, with proper local adaptation, the intensity test features can reach the performance of the gradient

pooling methods, which is is something not previously possible.

The real power of the proposed method, can be indicated by comparing the results of the BOLD descriptor directly with the other descriptors that are based on simple intensity tests such as BRIEF and ORB (indicated by $*$) in Figure 3.14, where we observe a reduction of the error rates by a factor of two.

3.5.2 Matching

In this section, we evaluate the proposed descriptor in image matching, following the framework proposed in [Mikolajczyk and Schmid, 2005]. Using the Harris-Laplace detector [Tuytelaars and Mikolajczyk, 2008], we extract a set of keypoints from each of the images and normalise them under a canonical representation. We extract a set of descriptors from all those patches and evaluate them with the original protocol from [Mikolajczyk and Schmid, 2005]. The results are reported in terms of recall vs. 1-precision, which is computed based on different matching thresholds.

In Figure 3.15 (top) we plot the results for a pair of images from each sequence that represents a significant transformation. Results of other image pairs are consistent. Interestingly, SIFT gives the best results overall. However, BOLD outperforms SIFT for the high precision part of the curves in Boat, Bikes and Bark sequences. It is worth noting that although BinBoost performs well in the patch dataset, it is ranked third in the matching experiment behind SIFT and BOLD. This may be due to a different training data used to optimise BinBoost and different feature points. This also indicates that the patch classification problem is a different problem than the nearest neighbour (NN) patch matching, and the two benchmarks are not interchangeable.

In Figure 3.15 (bottom) we plot the improvement introduced by our online selection of stable binary tests in the intra-class optimisation. The advantage of per-patch vs. global optimisation is significant and consistently observed in all our experiments on different datasets.

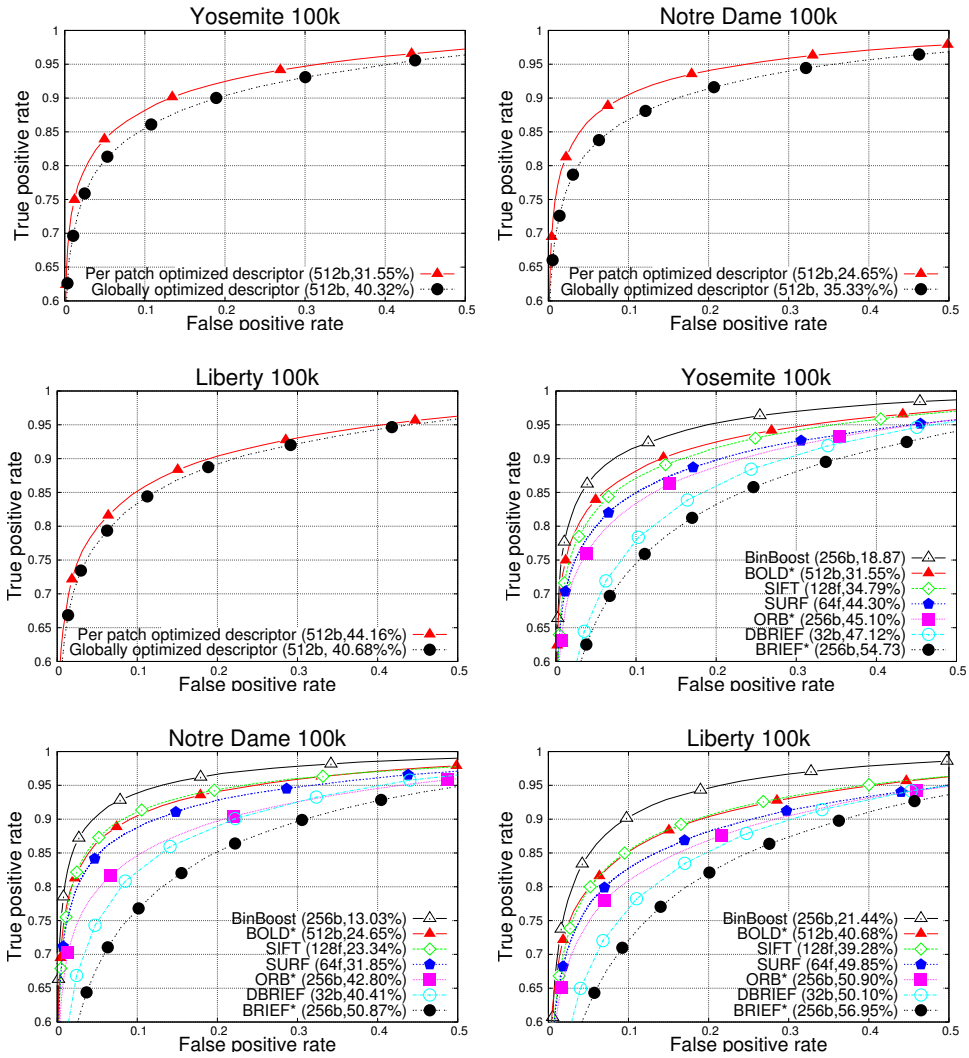


Figure 3.14: (First 3 graphs) Globally vs. locally optimised features. We see that our per-patch adaptation leads to improved performance across all false positive rates. (Remaining 3 graphs) BOLD compared to several state of the art descriptors. Descriptors with * are based on simple intensity tests. Using our per-patch optimisation framework, performance of SIFT can be matched with simple intensity tests instead of gradient statistics.

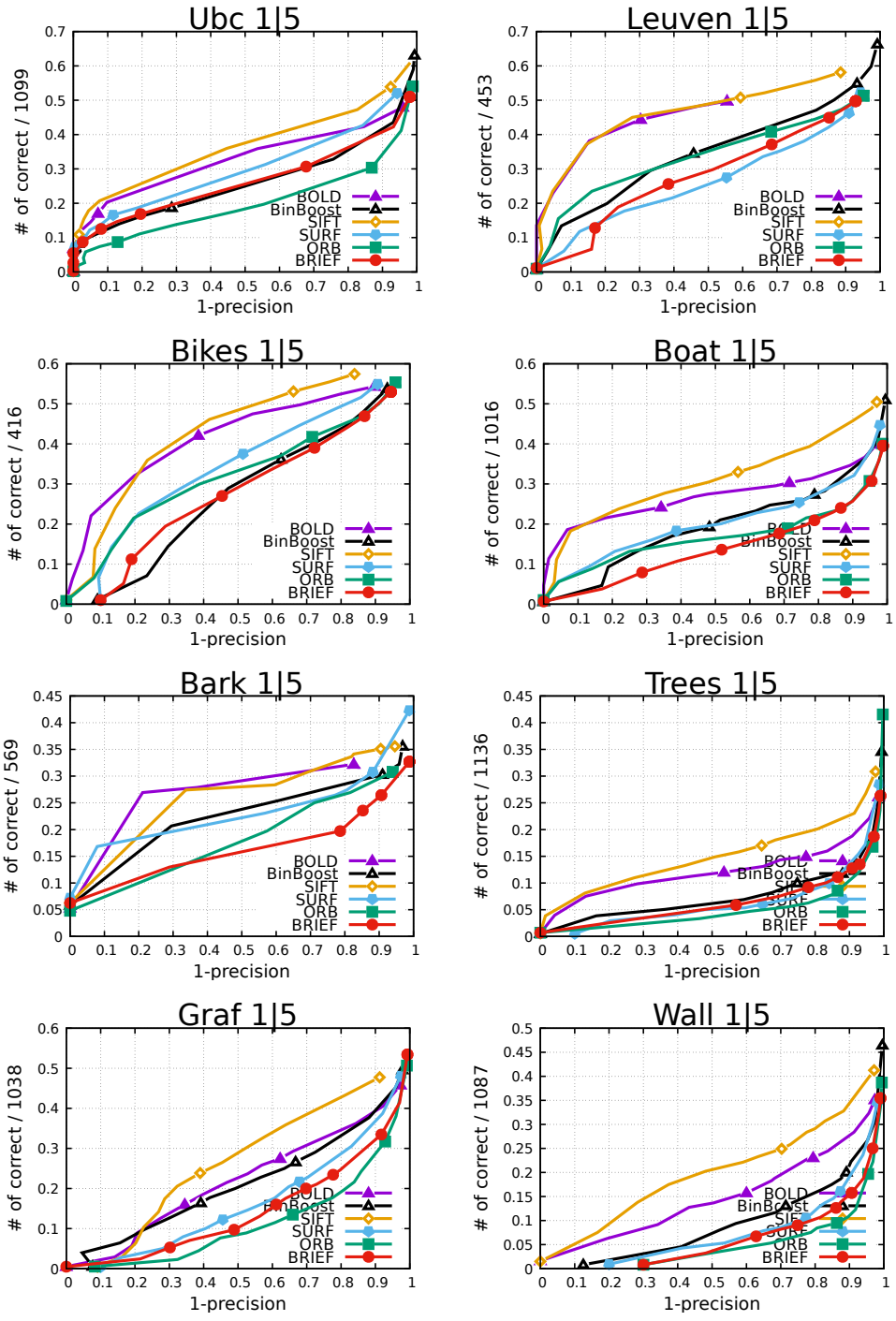


Figure 3.15: Keypoint matching experiment of the benchmark from [Mikolajczyk and Schmid, 2005]. Note that our descriptor approaches, or in some precision areas outperforms SIFT, which is the state of the art in matching scenarios. Also note the significant advantages compared to other descriptors based on simple intensity tests, such as ORB and BRIEF.

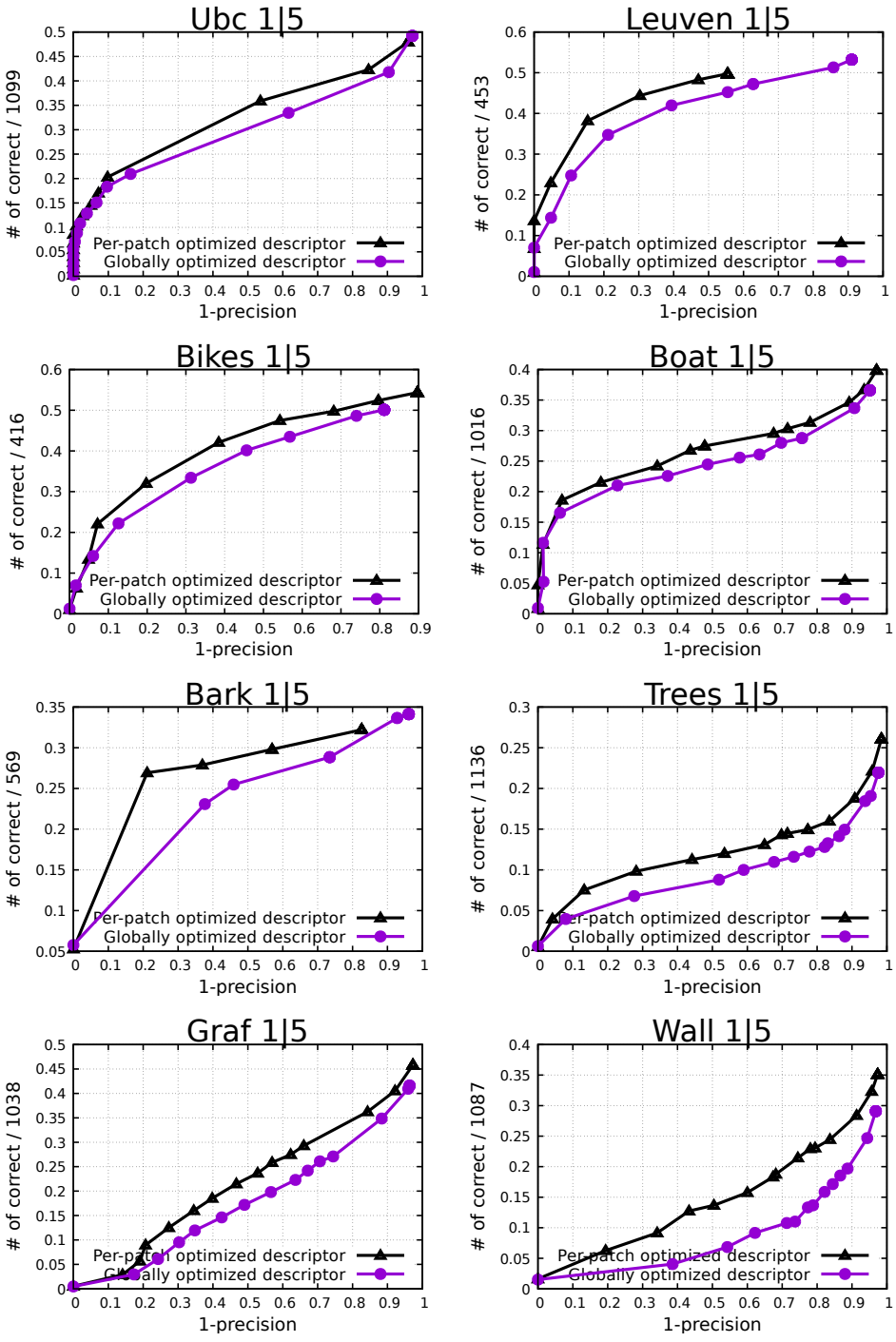


Figure 3.16: Comparison of the unmasked version of our descriptor with the masked version. We can see that masking has a positive effect across all precision levels, and leads to significant gains in performance, with similar computational efficiency.

3.5.3 Tracking

In this section, we demonstrate the application of our method to the tracking by detection problem. Several works [Kalal et al., 2012, Hare et al., 2011] follow the tracking by detection approach in which a model is initialised in the first frame, and updated online in order to account for appearance changes.

For our experiment, we use the tracking-by-detection mechanism from [Kalal et al., 2012] where the online learned detector is based on random ferns [Özuysal et al., 2010]. We build a detector that is trained in the first frame but it is not updated online to avoid the influence of various training examples that can be collected online and alleviate the problem of weak binary tests. Our goal is to show the impact our optimisation on the binary tests adapted to the object to be detected by the fern classifier.

Similarly to [Özuysal et al., 2010] we create a classification system based on a set of N simple binary features of intensity differences, identical to the ones in BRIEF and ORB. Following a sliding window approach, which is common among the state of the art detectors, our goal is to classify each window candidate as object or background.

Since each of the f_i features is a simple test, a number of those tests are required to achieve good detection performance. The authors of [Özuysal et al., 2010] apply ≈ 300 while the fern classifier in [Kalal et al., 2012] uses ≈ 130. A complete representation of the posterior probabilities for each of the background and object classes is therefore impractical due to the large number of used binary tests. Thus in [Özuysal et al., 2010] N features are divided into M groups of size $\frac{N}{M}$. Each of those groups forms a fern. The conditional probability becomes $P(f_1, f_2, ...f_N|object) = \prod_{i=1}^{n} P(F_k|object)$. Following [Kalal et al., 2012], we use a sum of the probabilities instead of a multiplication, and a threshold $t_{object} = 0.5$. Thus, if $\sum_{i=1}^{n} P(F_k|object) \geq t_{object}$ we consider it a valid detection.

The goal of this experiment is to demonstrate that the performance of the fern detector depends on the choice of the tests f_i. Full randomization in all stages is proposed in [Özuysal et al., 2010], but based on our results from matching the descriptors, we investigate if the per-object adaptation of the binary features that

are included in the ferns, can have an effect on the final result. For the results in Table 3.3, we use the same detector configuration as in TLD with 10 ferns, each consisting of 13 binary intensity tests. The posterior $P(F_k|object)$ for each fern is learned only from the first frame, using a set of 200 affine transformations of the original patch plus noise.

We generate a pool of 20 ferns, and compare two strategies for the selection of the final 10 that will act as the classifier, one global and one adapted per object. In the first case, we follow the approach of [Özuysal et al., 2010, Kalal et al., 2012] by randomly selecting a subset. For the second approach, we evaluate the posteriors of each fern in our set of 200 positive examples generated from the object, and we choose the 10 ferns that minimise the intra-class Hamming binary distance across the synthesised 200 positive examples.

We test this method in 10 sequences from the recently published tracking benchmark [Wu et al., 2013]. We report the recall, which is $\frac{\#\ correct\ detections}{\#\ frames}$. We do not report the precision, since this simple detector/tracker does not update its model online, its precision is therefore 1 or very close to 1 in most cases.

The results reported in Table 3.3 compare the randomly generated tests to object-adapted ferns based on our approach. The per-object optimised ferns perform significantly better than the random tests. Similar to per-patch online adaptation of descriptors, per-object adaptation of ferns improves the recall of the detectors. Object tracking by detection is an excellent application for the proposed method, as due to the efficiency requirements the learning has to be done online while most powerful machine learning methods that require a large set of training examples have limited use in such application.

3.5.4 Low bit-rate locally adapted descriptors

We experiment with low dimensional locally adapted descriptors to examine how their performance would vary in a more memory constrained environment. Such a system could be used in several memory and computationally intensive applications such as embedded systems and tracking scenarios. In Figure 3.17, we plot several different versions of our locally adapted descriptors, with dimensionality as low as 32 bits. It is clear that local adaptation is much better across all dimensionalities,

Sequence	Global Ferns	Ferns adapted per object
Subway	0.19	0.28
Jumping	0.26	0.46
Girl	0.44	0.58
Suv	0.25	0.42
Woman	0	0.1
Freeman1	0.07	0.13
Freeman4	0.09	0.16
Deer	0.04	0.18
Crossing	0.3	0.45
Couple	0.03	0.1
Average	0.17	0.29

Table 3.3: Recall results for 10 sequences of the tracking evaluation benchmark from [Wu et al., 2013] for a global set of ferns, compared to per-patch adaptation of the fern object classifier. We observe that adapting the subset of ferns per object outperforms a global set of ferns fixed across all objects.

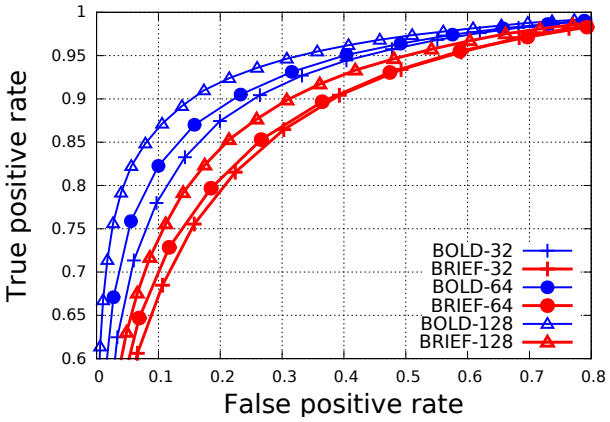


Figure 3.17: Low bit-rate versions of our locally adapted descriptor, compared with low bit-rate BRIEF descriptors.

and surprisingly, our 32-bit dimensional descriptors performs better than even the 128 dimensional BRIEF equivalent. In fact, our experiments show that the 32-dimensional version of our descriptor is on par with a 300-dimensional BRIEF descriptor.

3.5.5 Adapting other binary descriptors

In this section we present results that show that the local adaptation of features based on the idea of masking unstable dimensions can be extended to other binary descriptors.

The results are presented in Figure 3.18, where for each descriptor X, we refer to the online version with local adaptation as $X - o$. We observe that the local adaptation provides significant improvement to all the descriptors that are based on binary intensity tests. In contrast, it provides only a modest improvement to Binboost. This can be explained by the fact that Binboost is based on gradients, and not on simple intensity tests. Features that are based on gradient masks, can be considered as being more robust to local adaptation, hence the identification of *stable dimensions* will only provide a slight improvement. On the contrary, due to the extremely noise prone nature of the simple intensity tests, the local adaptation helps to significantly improve the results.

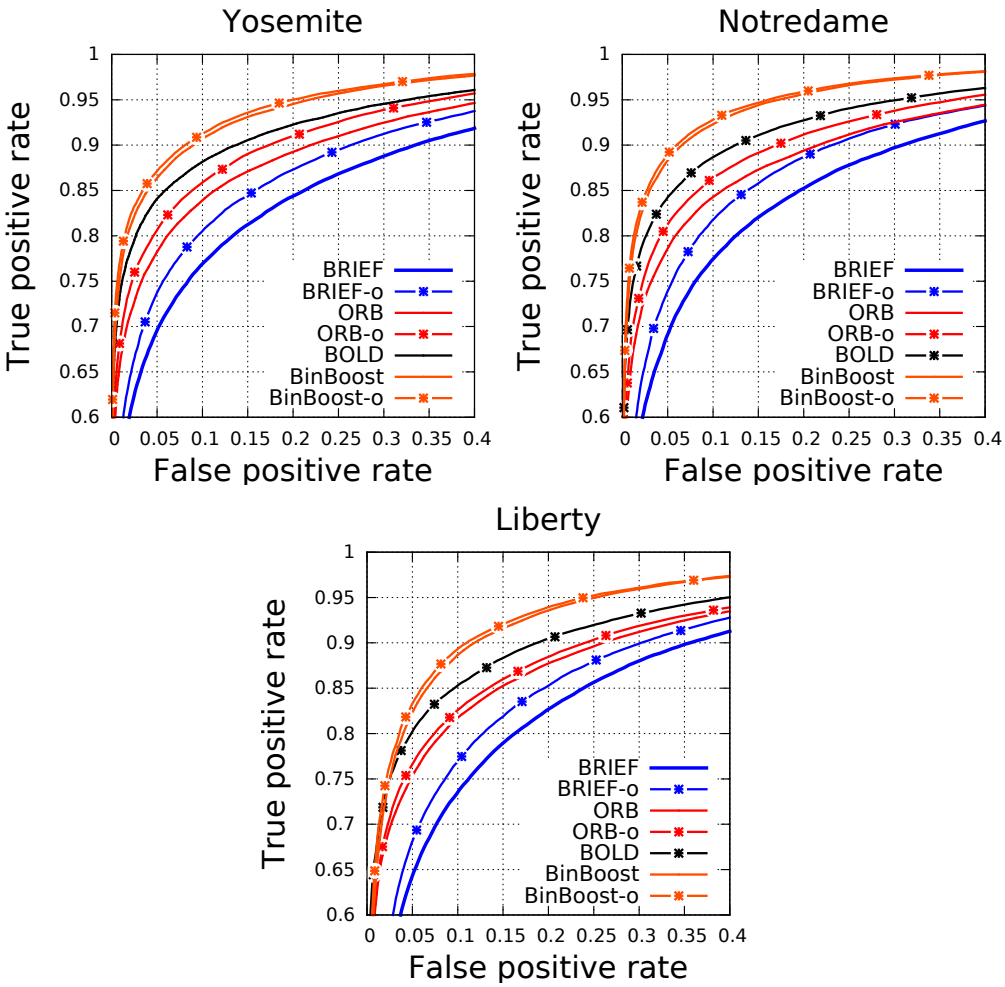


Figure 3.18: Extending the local adaptation of binary features to other binary descriptors. X-o represents the locally adapted version of the X descriptor. We see that there is a consistent improvement across all the different descriptors, especially for descriptors that are based on intensity tests.

Distance (512 dimensions)	μs
$\mathbf{x}_L \oplus \mathbf{x}_R$	220
$(\mathbf{x}_L \oplus \mathbf{x}_R \wedge \mathbf{y}_L) + (\mathbf{x}_L \oplus \mathbf{x}_R \wedge \mathbf{y}_R)$	340

Table 3.4: Computational efficiency of the proposed masked Hamming distance, across 1000 pairs of patches. Our proposed masked Hamming distance presents similar efficiency to the original Hamming distance, while being more discriminative.

3.5.6 Speed

One of the main advantages of the proposed BOLD descriptor is its extraction and matching speed. We therefore discuss the computational efficiency of the proposed masked Hamming distance (cf. Section 3.3.1). The results are averaged on a set of $100k$ patches from the Liberty dataset. All the experiments were done on an Intel i7-Haswell processor with the avx-2 instruction set enabled, and all the possible SIDM optimisations were included (i.e. `popcount`).

In Table 3.4, we compare the calculation time of our masked distance to the regular Hamming distance when matching two binary descriptors. Despite the introduction of the symmetric masked Hamming distance and thus longer binary string, the computational efficiency remains very high i.e. only $340\mu s$, and comparable to the regular Hamming distance of $220\mu s$. The only additional operation is the logical AND with the masks otherwise the optimised instructions compensate for longer strings.

In Table 3.5, we report the running times for extraction and matching for several of the descriptors reported in the results. We can observe that BOLD presents much better results in terms of 95% error rate and remains competitive with BRIEF in terms of both extraction and matching speed. Real valued descriptors such as SIFT and SURF have a long extraction and matching time i.e. 5-40 times slower than BOLD. BinBoost is the slowest in this set of descriptors. ORB and BRIEF are still three times faster as no optimisation is applied during extraction.

Furthermore, Figure 3.19 presents the performance of each descriptor in with

Descriptor	extraction	matching	total
BinBoost	713	0.11	713.11
SIFT	417	10	427
SURF	48.2	5	53.2
BOLD	10.5	0.34	10.84
DBRIEF	6.8	0.02	6.82
ORB	2.7	0.11	2.88
BRIEF	2.7	0.11	2.88

Table 3.5: Comparison of efficiency per operation for various feature descriptors. Time is reported in μs per descriptor. We can observe that the proposed descriptor exhibits similar computational efficiency to the fastest available binary descriptors.

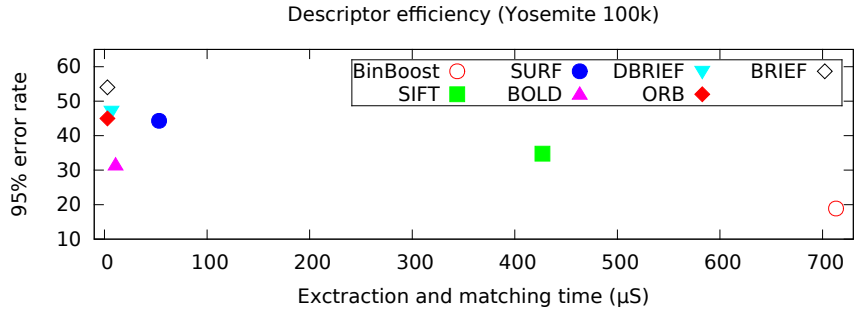


Figure 3.19: The proposed BOLD descriptor has good properties of low error rates and high computational efficiency.

respect to its computational requirements. An ideal descriptor would be close to the $(0, 0)$ point in this graph. With the proposed framework, we achieve error rates similar to the SIFT descriptor, with extraction times on the level of BRIEF descriptor. The top performance is with BinBoost, which however is 70 times less efficient than BOLD. More importantly, as we have seen, BinBoost is not able to perform close to state of the art in a NN matching scenario.

3.5.7 Adapting floating point descriptors

In this section, we investigate the possibility of extending the masking methods presented in this chapter to floating point descriptors. Contrary to the binary

case, where the invariant dimensions could be identified by their zero variance, in the floating point case a sorting or thresholding of variance is needed. In the experiments below, we investigate the sorted variances to produce the masks. More precisely, let $X \in \mathbb{R}^{M \times D}$ denote the feature matrix of M patches, and a D-dimensional descriptor. Similarly to the binary case, we extract a vector v of D variances, by computing the variance of each column of X. Subsequently, we create a mask m with $\sum_i^D m_i = S$ by selecting either the top S features from the sorted v, or the bottom S features i.e. we select high or low variance features respectively. We refer to the former group as *low* σ, and the latter as *high* σ. According to the results from the binary descriptors, the *high* σ group is expected to perform significantly worse than the *low* σ one. Below we present some experiments that validate this assumption, and subsequently indicate that the masking framework can be extended to floating point descriptors.

Firstly we examine the behaviour of SIFT on a set of patches extracted from the same interest point. This collection of patches is shown in Figure 3.20(a). The variance of the SIFT individual dimensions across all the patches is shown in Figure 3.20(b). We observe that there is significant fluctuation of the variance, and more surprisingly we see that there are specific SIFT dimensions which present very high variances although visually the differences between patches are not significant, and the deformations are minimal.

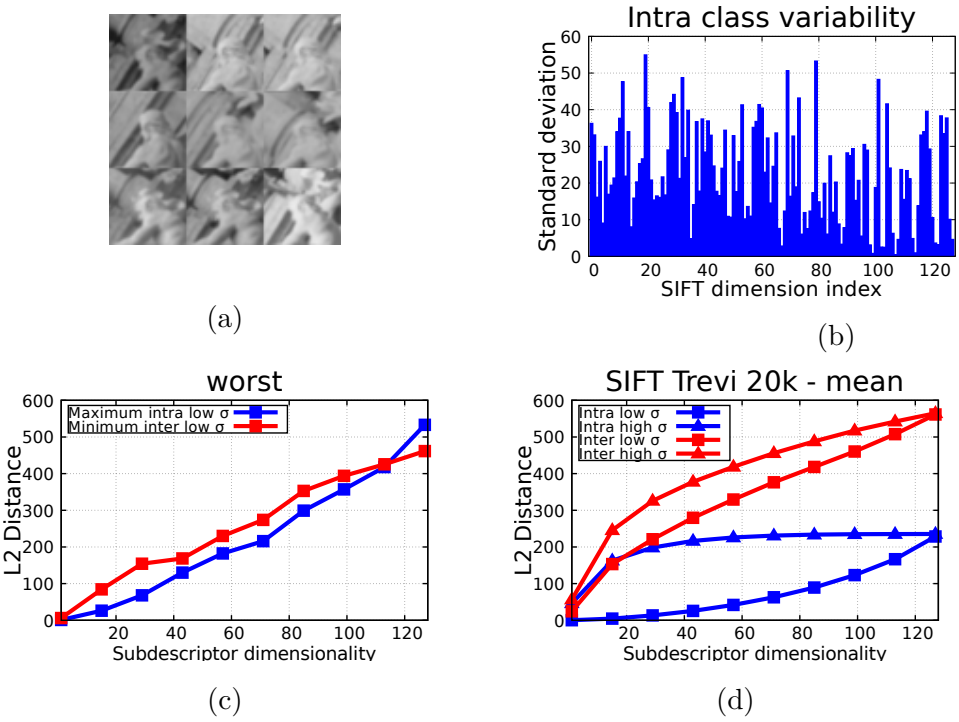
Secondly, we plot the results for the *worst case* values of inter vs intra class distances in Figure 3.20(d). These are extracted by considering the worst case scenario, which is defined as the comparison of the maximum measured intra-class distance with minimum measured inter-class distance. We use random patches in order to compute the inter-class variances, and we reduce the dimensions of the original descriptor from 128 to S (where S is plotted on the $x - axis$) by masking the *high* σ dimensions. By analysing the form of the graphs, we see clearly the effect of the use of the *low* σ dimensions in the final descriptor performance. For the full 128 SIFT descriptor, the the minimum inter-class distance is lower than the intra-class distance. Note that in real world matching scenarios, the fact that the minimum intra-class distance is lower than than the maximum inter-class distance, might lead to incorrect nearest-neighbour matching, and reduce the success rate

and the mean average-precision (mAP) of matching algorithms. However, we see that if we mask the dimensions with very high variance (e.g remove ≈ 10 of them), we can effectively discriminate the classes, because the minimum inter-class becomes higher than the maximum intra-class distance.

Finally, to show that this phenomenon is not only true for the small set of patches from Figure 3.20(e), but it is a general trend in the datasets, we report the large scale results for $20K$ patch classes from the Trevi dataset of Winder and Brown [2007]. We plot the average inter and intra-class distances across a set of 20k patch classes, and we report the results of masking low and high variance dimensions. We observe that reducing the dimensions with the *high* σ method results in low separation between the inter and intra-class distances. On the contrary, the use of the *low* σ method, results in higher margin between the inter and intra class distances, which leads to better discriminability.

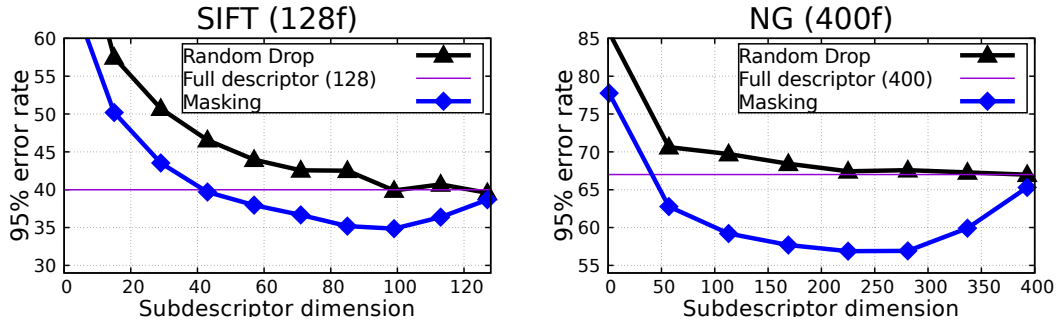
To validate the above, we also perform an experiment with the commonly used 95% error rate. For this experiment, in order generate the intra-class matrix and identify the low variance dimensions, we use the framework from Cai et al. [2011] to generate affine deformations of a patch. We generate 8 affine versions of each patch, with the transformation parameters reported. We then use the $8 \times D$ matrix to identify the low variance dimensions. This number of affine synthetic deformations was chosen experimentally as a good compromise between computational efficiency and good results.

In Figure 3.21 we plot the results for two different features, SIFT and Normalised Grayscale (NG) Cai et al. [2011]. The NG features are extracted after converting a raw patch to zero-mean unit-variance representation. For each case, the subdescriptor of dimensionality S ($x - axis$) is created by including only the best S dimensions with lowest variance in the final distance computation. In all both cases, the locally adapted descriptor improves the performance of the global descriptor. We also plot the performance of *Random Drop* which consists of simply selecting a random subset of cardinality S. As expected, such a method presents a steady deterioration of performance as more dimensions of the original descriptor are dropped, and it never outperforms the global superset, in contrast to our masking method.



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

Figure 3.20: **(a)** A single interest point under a set of deformations (9 samples) **(b)** Intra-class SIFT dimensions variance for the patches from (a). Note that several dimensions present very high variance across the different views of the same patch, although the deformations from (a) are visually minimal.**(c)** Comparing the maximum intra-class distance with the minimum inter-class distance. **(d)** Distribution of *low* σ and *high* σ masking methods for 20K patch classes.



(a) (b)

Figure 3.21: Plots of the proposed local adaption method for two floating point cases, SIFT and zero-mean-unit variance features. Our masking method outperforms the global descriptor, despite the fact that it is always a subset of the original.

For SIFT in Figure 3.21 (a) our locally adapted subdescriptors, outperform the parent global descriptor across all dimensionalities from 40 to 127. Interestingly, we can see that only 40 dimensions are required in a locally adapted descriptor to match the performance of the full descriptor which is approximately 3 times larger with 128 dimensions. The effect is even greater when considering the NG case. Similarly to the binary case, this validates that the less discriminative an individual feature is, the more it can benefit from online adaptation.

3.6 Conclusion

In this chapter, we investigated a novel method of improving the performance of binary descriptors based on modified distance computations that are locally adapted to each individual patch for better discriminability. Several experiments across different fields, indicate the merits of this method. In addition, we briefly demonstrate that such a method is also possible in floating point descriptors. Our masking method has been successfully applied both for identifying features invariant to motion [Zhang et al., 2016] and invariant to scales [Tsun-Yi Yang and Chuang, 2016], which shows the great potential of locally adapted descriptors across different applications.

Chapter 4

Learning feature descriptors with shallow convolutional neural networks

In this section, we investigate the use of shallow convolutional neural networks and novel loss methods as fast and robust feature descriptors. In section 4.1 we give a brief introduction to the problem, and in section 4.2 we present the main methodology. Finally, in section 4.3, we present extensive evaluations that show that our method provides state of the art discriminative power combined with several orders of magnitude more efficient operation that previously used complex architectures.

4.1 Introduction

The method for local adaptation of descriptors to patches presented in the previous chapter, can reach state of the art in terms of nearest neighbour matching when considering binary descriptors and is able to provide significant performance improvement, with minimal computation costs. Nevertheless, the discriminative power of intensity tests as individual features, is very limited compared to methods such as pooling filtered response histograms [T. Trzcinski and Lepetit, 2013].

The significant gains of using deep learning methods across almost all areas of

computer vision, has indicated that learning hierarchical representations, specifically adapted to individual domains is much more effective than either directly using engineered features, or applying machine learning methods to these engineered features to enforce higher discriminability and thus higher performance. Specifically, deep learning can offer significant advantages when applied to local feature descriptors where the goal is to learn a representation that remains invariant to viewing conditions, as well as a distance metric to compare the descriptors.

In this chapter, we investigate the use of convolutional neural networks for learning local feature descriptors. We focus on shallow networks, that present the advantage of limited overfitting and low computational complexity. Typically such networks are trained with examples that consist of pairs of data samples. In contrast, in this work we show that by using a triplet based optimisation method, we can avoid complex architectures and hard negative mining of the training samples, techniques that are essential components of many state of the art methods. An additional advantage is that our networks do not contain specifically trained metric layers, since our training process involves directly optimising the $L2$ difference between the learned local feature representations. We then show that our convolutional feature descriptor outperforms the state of the art, while operating with significantly lower complexity.

This chapter is organised as follows. In section 4.2 we give the basic description of our method, and we briefly discuss some implementation details. Section 4.3 experimentally illustrates the improved performance of our proposed method compared to the previous state of the art in terms of convolutional feature descriptors.

4.2 Learning convolutional patch descriptors

In this section, we first formulate our optimisation problem, and discuss the two commonly used loss functions applied to learning with triplets, following with a brief investigation of their properties. In our experiments, a descriptor is considered as a non-linear encoding of a patch resulting from extracting the response of the final layer of a convolutional neural network. Note that several authors also refer to this procedure as feature embedding [Hadsell et al., 2006, Chopra et al.,

2005]. Since we are interested in extracting representations for patches centred on features, we refer to the embedding process, as learning a *convolutional feature descriptor*.

4.2.1 Problem formulation

A deep neural network can be represented as the composition of multiple functions, where each function represents one layer. Formally, given input $\boldsymbol{x} \in \mathbb{R}^{n \times n}$ (gray-level patch), the network can be represented as follows [Lecun et al., 2015]

$$f(\boldsymbol{x}) = g_n(g_{n-1}(g_{n-2}(...(g_1(\boldsymbol{x})...)))) \tag{4.1}$$

where g_i represents the i^{th} layer of the network. Such layers can be convolutional filters, fully connected, max pooling and a large number of other specialised layers.

Note that we are interested in networks that directly output the final feature representation instead of learning a separate metric layer. This has the benefit of avoiding specialised fully connected distance layers that will add a large magnitude of parameters to the network and might encourage over-fitting. In addition, by using a convolutional network only to learn features, and not learn metric layers makes it possible for the extracted features to be directly usable in all the previously developed matching and indexing methods in the bibliography.

Using the notation from Equation 4.1 it is clear that the convolutional patch descriptor is the result from the output of the layer $g_n(\cdot)$. Thus, output dimensionality of the final layer $f(\boldsymbol{x}) \in \mathbb{R}^D$, indicates the dimensionality of our descriptor, and it can be adjusted according to specific memory requirements.

The goal of the optimisation process when learning convolutional patch descriptors is to adjust the parameters of the network in a way that $||f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)||_2$ is low if $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are extracted from the same physical point location (i.e. constitute a positive pair), and high otherwise. In the following sections, we discuss the possible ways to optimise such features, using parallel training architectures.

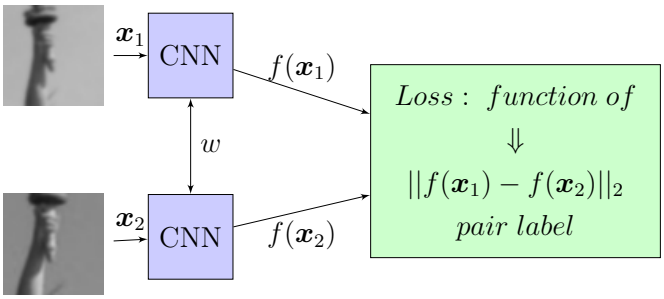


Figure 4.1: Training with pairs. The loss is computed based on the distance between the feature embeddings of either positive or negative patch pairs. Note that the two convolutional networks have shared parameters (indicated by w). Thus the weights are adjusted simultaneously and in an identical way, based on loss results from pairs of input training data.

4.2.2 Learning with pairs

The most commonly used architecture for learning feature embeddings with convolutional neural networks is the *siamese* architecture consisting of two cloned copies of one convolutional neural network. This architecture is shown in Figure 4.1. Such a learning system was used in Bromley et al. [1993] to verify pairs of signatures. Subsequently, it was used across many different fields and has proven to be very efficient in learning robust non-linear embeddings for several kinds of inputs [Hadsell et al., 2006, Altwaijry et al., 2016].

More formally, learning with pairs and a siamese architecture involves training from samples of the form $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ell\}$, with ℓ being a training label. By using $\ell = -1$ for negative pairs, and $\ell = 1$ for positive pairs, the contrastive loss for a set of of N samples (mini-batch), is defined as

$$\sum_{i=1}^{N} l(\boldsymbol{x}_1, \boldsymbol{x}_2; \ell) + \lambda \cdot ||\boldsymbol{w}||_2^2 \tag{4.2}$$

where

$$l(\boldsymbol{x}_1, \boldsymbol{x}_2; \ell) = \begin{cases} ||f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)||_2 & \text{if } \ell = 1 \\ max(0, \mu - ||f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)||_2) & \text{if } \ell = -1 \end{cases} \tag{4.3}$$

and μ is an arbitrarily set margin, that does not have any significant effect, and is normally set to 1. λ is the $L2$ regularisation factor also known as weight decay. Regularisation is crucial in this setting, in order to avoid overfitting, and to control the weights of the network, since we are interested in feature learning. Intuitively this loss penalises positive pairs that have large distance and negative pairs that have small distance (less than μ).

Learning local feature descriptors is a more specific problem than general image classification such as in ImageNet [Russakovsky et al., 2015], since the transformations a local patch can undergo are limited compared to varied objects under the same visual category. In addition, patches in pairs representing negative examples are usually very different, thus making it easy for the learning process to optimise the distances. To illustrate this point, we show some sample positive and negative training pairs from the Photo Tourism dataset in Figure 4.3. It is clear, that some of the negative pairs are very easy to optimise, since even some non-discriminative filters could easily produce features that adhere with the loss formulation of Equation 4.3.

This issue is also discussed in [Simo-Serra et al., 2015], where the majority of the negative patch pairs ($\ell = -1$) do not contribute to the update of the gradients in the optimisation process as their distance is already larger than the μ margin parameter in Eq. (4.3). To address this issue the authors propose hard negative mining inspired by its success in the object detection field [Felzenszwalb et al., 2010]. The hard negative training pairs were identified by their low distance during the learning process, and a subset of these examples were re-fed to the network for gradient update of each iteration. Thus, a first optimisation step with all training data inside a mini-batch is done, and subsequently, the process is repeated inside the mini-batch, with only a subset of the training data (i.e the hard negatives). Note that while this process leads to more discriminative convolutional features, it also comes at a very high computational cost, since in each epoch a large subset of the training data needs to be forwarded and backpropagated again through the network. Specifically, the best performing architecture from Simo-Serra et al. [2015], required 67% of the computational cost to be spent for mining hard negatives.

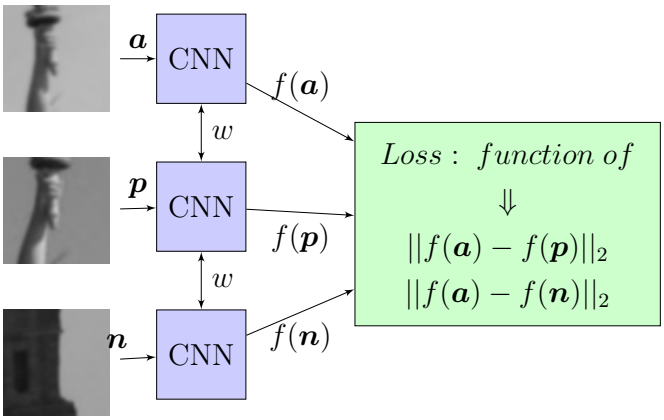


Figure 4.2: Training with triplets. Each triplet consists of an anchor (a), a positive example (p) and a negative example (n). The loss is based on the comparison between the anchor-positive and the anchor-negative distances inside a triplet. w indicates shared weights (cloning) between the three convolutional networks.

4.2.3 Learning with triplets

Recent work in [Hoffer and Ailon, 2014] shows that learning with triplets of training data, outperforms learning with pairs using the same network in terms of classification accuracy. This work expands on earlier work by Wang et al. [2014b] that only focused on the performance of triplet-based learning on image retrieval. Inspired by these significant improvements in other fields, we focus on learning convolutional feature descriptors based on triplets of local patches.

Learning with triplets involves training from samples in the form $\{\boldsymbol{a}, \boldsymbol{p}, \boldsymbol{n}\}$, where a is the *anchor*, p *positive*, which is a different sample of the same class as a, and n *negative* is a sample belonging to a different class. In the case of learning local features, a and p are patches extracted from the same physical point, but with different conditions (e.g. affine transformations and illumination variations), and n is extracted from a different physical point. A figure of the training process is shown on Figure 4.2.

The goal of the optimisation process, is to updated the parameters of the network in such way that a and p are closer in the embedded feature space, and a and n are further apart in terms of their $L2$ distances. For brevity, we shall write

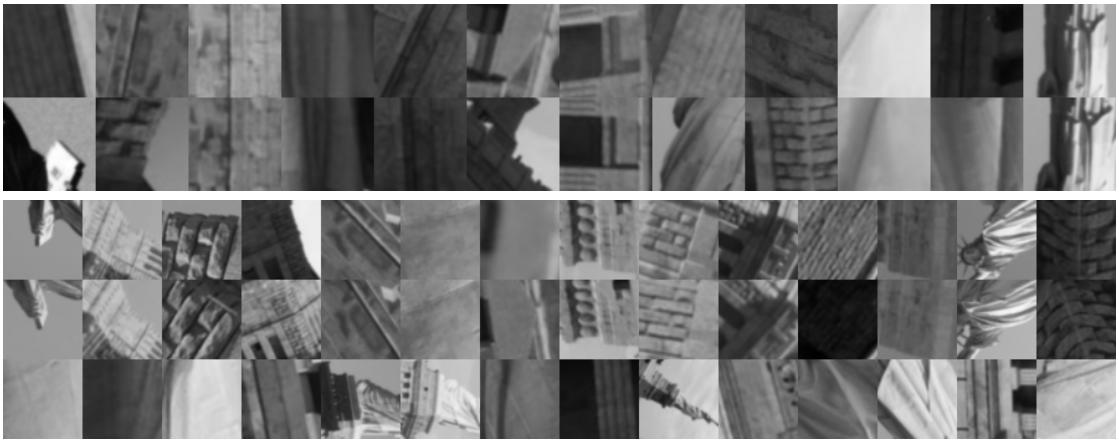


Figure 4.3: Pair based networks are trained with positive and negative pairs of patches (top). Our triplet network is trained with triplets of patches (bottom), two extracted from the same point representing a positive match example, and one from a different point in space giving two negative match examples per triplet.

that $\delta_+ = ||f(\boldsymbol{a}) - f(\boldsymbol{p})||_2$ and $\delta_- = ||f(\boldsymbol{a}) - f(\boldsymbol{n})||_2$. We can categorise the loss functions that have been proposed in the literature for learning convolutional embeddings with triplets into two groups, the *ranking-based losses* and the *ratio-based losses* Wohlhart and Lepetit [2015], Hoffer and Ailon [2014], Wang et al. [2014b]. Below we give a brief review of both categories, and discuss their differences.

Margin ranking loss

This ranking loss that was first proposed for learning embeddings using convolutional neural networks in Wang et al. [2014b]. For a set of N training samples of the form $\{a, p, n\}$ (mini-batch),it is defined as

$$\sum_{i=1}^{N} l_{rank}(\delta_+, \delta_-) + \lambda \cdot ||\boldsymbol{w}||_2^2 \tag{4.4}$$

where

$$l_{rank}(\delta_+, \delta_-) = max(0, \mu + \delta_+ - \delta_-) \tag{4.5}$$

where μ is a margin parameter. The margin ranking loss is a convex approximation to the non-convex $0-1$ ranking error loss, which measures the violation

of the ranking order of the embedded features inside the triplet, which should be $\delta_- > \delta_+ + \mu$. If that is not the case, then the network adjusts its weights to achieve this result. As it can be seen, the formulation also involves a margin, similarly to Eq.(4.3). Note that if this marginal distance difference is respected, the loss is 0, and thus the weights are not updated. Figure 4.4 (b) illustrates the loss surface of $l_{rank}(\delta_+, \delta_-)$. The loss remains 0 until the margin is violated, and after that, there is a linear increase. Also note that the loss in not upper bounded, only lower bounded to 0.

Ratio loss

In contrast to the ranking loss that forces the embeddings to be learned such that they satisfy ranking of the form $\delta_- > \delta_+ + \mu$, a ratio loss is investigated in Hoffer and Ailon [2014] which optimises the ratio distances within triplets. This loss aims to learn feature embeddings such that $\frac{\delta_-}{\delta_+} \to \infty$.

$$\sum_{i=1}^{N} l_{ratio}(\delta_+, \delta_-) + \lambda \cdot ||\boldsymbol{w}||_2^2 \tag{4.6}$$

where

$$l_{ratio}(\delta_+, \delta_-) = (\frac{e^{\delta_+}}{e^{\delta_+} + e^{\delta_-}})^2 + (1 - \frac{e^{\delta_-}}{e^{\delta_+} + e^{\delta_-}})^2 \tag{4.7}$$

As one can examine from Equation 4.7, the goal of this loss function is to force $(\frac{e^{\delta_+}}{e^{\delta_+}+e^{\delta_-}})^2$ to 0, and $(\frac{e^{\delta_-}}{e^{\delta_+}+e^{\delta_-}})^2$ to 1. Note that both are achieved by the first term of the equation since setting the first term to 0, automatically sets the second term to 1, but we report here the original formulation. There is no margin associated with this loss, and by definition we have $0 \leq l_{ratio} \leq 1$ for all values of δ_-, δ_+. Note that unlike the margin-ranking loss, where $l_{rank} = 0$ is possible, every training sample in this case is associated with some non-negative loss value. Figure 4.4 (d) shows the loss surface of $\hat{\lambda}(\delta_+, \delta_-)$, which compared to the ranking based loss has a clear slope between the two loss levels, and the loss reaches a plateau quickly when $\delta_- > \delta_+$. Also note that this loss is upper bounded to 1.

To experimentally illustrate the difference between the two formulations, we plot in Figure 4.4 (e) the distribution of the computed loss numerical values for a

total for 500 mini-batches of consisting of 128 samples each. We forward the mini-batches to the network, and we record the loss computed using the embeddings learned from the networks and equations 4.7 and 4.5. We note that the average ratio loss is dominated by two distinct areas concentrating either around 0, or around 1. On the other hand, the loss with the ranking loss follows a more normal distribution, which can be explained by the linear curve in the loss surface.

4.2.4 In-triplet hard negative mining with anchor swap

All previous works that exploit the idea of triplet based learning utilise the δ_- and δ_+ distances from the anchor to the positive example, and the anchor to the negative example respectively. However, this accounts for only two of the possible three distances within each triplet w.r.t. one sample used as an *anchor*, thus ignoring the third distance $\delta_-^{'} = ||f(\boldsymbol{p}) - f(\boldsymbol{n})||_2$. Note that since the feature embedding network already computes the representations for $f(\boldsymbol{a}), f(\boldsymbol{p}), f(\boldsymbol{n})$, there is no need for extra overhead of evaluating convolutional filters or re-propagating examples through the network. The only overhead in this case, is the computation of a single $L2$ distance between the $f(\boldsymbol{p})$ and $f(\boldsymbol{n})$ embeddings.

Our key observation, is that this extra distance can be used in the learning procedure, since it might provide extra information to the gradient computation process. To do this, we dynamically evaluate the anchor in each training datum, instead of fixing it offline similarly to previous work.

We define the *in-triplet hard negative distance* as $\delta_* = min(\delta_-, \delta_-^{'})$. The name is inspired from the fact that the minimum distance between δ_- and $\delta_-^{'}$, can be thought of as the hard negative distance inside this specific triplet. Thus, intuitevly, the anchor comparison between the anchor-positive and anchor-negative distances should be made using the hardest negative inside a triplet, which represents a more challenging scenario for the convolutional network. To dynamically choose the anchor, if $\delta_* = \delta_-^{'}$, we swap $\{a, p\}$, and thus p becomes the *anchor*, and a becomes the *positive* sample. This ensures that the hardest negative inside the triplet is used for backpropagation. Subsequently, the margin ranking loss becomes $\ell(\delta_+, \delta_*) = max(0, \mu + \delta_+ - \delta_*)$. A similar expression can be devised for the ratio loss. This simple technique can lead to improved results without computational

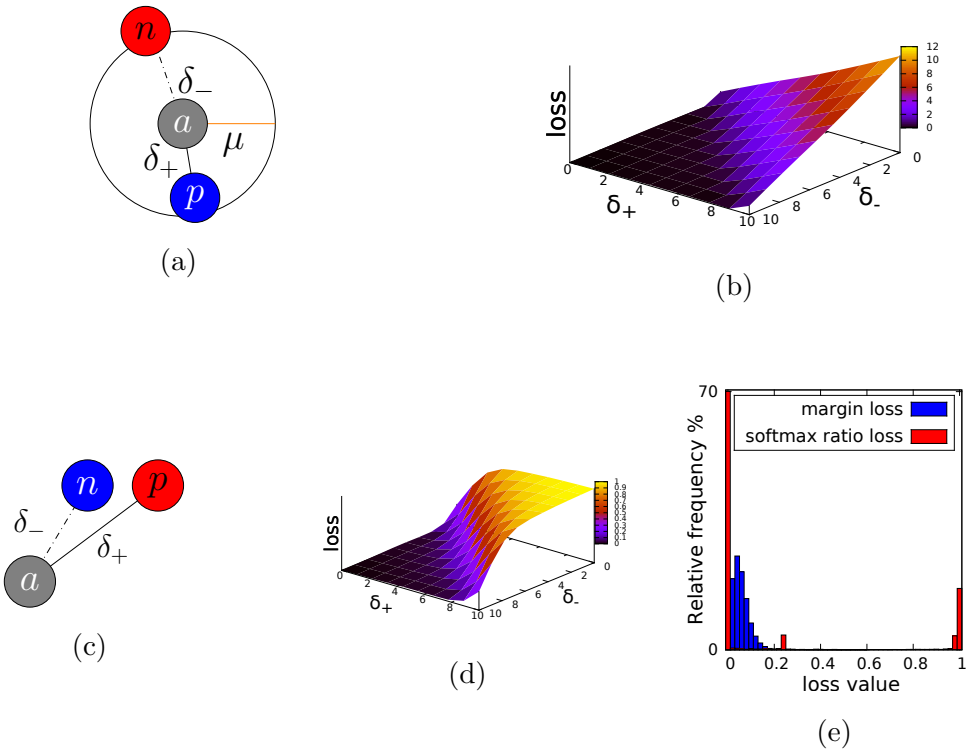


Figure 4.4: (a) Margin ranking loss. This loss seeks to push n outside the circle defined by the margin μ, and pull p inside. (b) Plot of the margin ranking loss values in function of the two distance inside a triplet, δ_-, δ_+ (c) Ratio loss. It seeks to force δ_+ to be much smaller than δ_-. (d) Ratio loss values in function of δ_-, δ_+ (e) Distribution of loss values for the two loss functions for forwarding through the network a set of 500 mini-batches. We can observe that the margin loss has a more normal distribution, something that can be attributed to the linear increase compared to the ratio loss.

overhead, as we experimentally show in section 4.3.1. Note that since the weights of the network are adjusted during the learning process, the dynamic anchor choice using our in-triplet hard negative method is not constant throughout the learning process for each individual training triplet.

4.2.5 Network depth

Together with the training method, the network architecture is very crucial in the learning process, as previous work in [Zagoruyko and Komodakis, 2015] has shown. The recent work on learning local convolutional feature descriptors reports excellent results with networks that are relatively shallow compared to the ones used in larger-scale image recognition problems such as the CIFAR dataset [Krizhevsky and Hinton, 2010]. This can be attributed both to the limited set of deformations local features adhere to, and the low dimensionality of the input. On the contrary, recognition networks have to deal with considerable intra-class variations.

Albeit shallower than their recognition counterparts, all the convolutional networks proposed in the recent literature for learning patch descriptors are still very complex and significantly slower than the previous state of the art feature descriptors such as SIFT or BRIEF, which limits their practical use in either problems where speed is the main concern or large scale applications with millions of patches.

Our goal is to produce a network that can be characterised by very efficient training and testing performance, thus allowing it to be used in novel ways that are not feasible with the current state of the art. To that end, we focus on a network consisting only of two convolutional layers, and a fully connected layer that reduces the filter responses to the desired output dimensionality. In order to be consistent with the pre-convolutional state of the art, we fix our outputs to a dimensionality of 128. Note however, that this parameter can be adjusted just by altering the output size of the final fully connected layer. We fully describe the architecture of our network in Table 4.1($2conv$).

It is worth noting that while our network consists of only two convolutional layers, all of the other state-of-the art deep feature descriptors consist of four or more layers ranging up to 10, including a series of fully connected layers. [Zagoruyko and Komodakis, 2015, Simo-Serra et al., 2015, Han et al., 2015]. Our design is also

inspired by the approach introduced in [Simonyan et al., 2014], where pooling of the responses of Gaussian filters and a simple linear projection produced very good results. Thus, we build a simple hierarchical network that is based on 100 convolutional filters, followed by a linear transformation that projects the responses of the filters to the desired output dimensionality. Note that our method also has the added benefit of learning hierarchical representations, which is not possible with the described convex optimisation method.

Lastly, note that several other implementation variants are possible such as different non-linearity layers, extra normalisation layers, different pooling configurations or multi-scale architectures. While these are likely to further improve the results, they are beyond the scope of this work, which is investigate the feasibility of shallow networks and triplet losses (Equations 4.5 & 4.7) for learning convolutional patch descriptors.

Recent work in the image recognition field, proposed that a series of more convolutional layers with smaller 3×3 kernels, can outperform, a configuration with less layers and larger filters. To that end, we also investigate a second configuration, which is slightly deeper and with significantly smaller convolutional kernels compared to the larger filters in the $2conv$ architecture.

To evaluate the two networks, we use the testing set of the Photo Tourism liberty dataset. We train both networks using exactly the same training triplets, and we train them using both margin ranking and ratio loss. In addition, we also show the loss when learning with pairs instead of triplets. The optimisation process carried out until convergence, which with our implementation and our configuration is met usually after the 60^{th} epoch. More analysis on the implementation details can be found on section 4.2.6.

We show the results in terms of 95% error rate in Figure 4.5. Interestingly, we can observe that when learning with pairs the deeper network is able to quickly reach a good performance and outperforms the shallower network. On the contrary, when learning with triplets, the shallow network performs much better based on both loss functions.

Nevertheless, we can confirm that learning with pairs results in significantly lower performance than learning features with triplets, regardless of the triplet loss

Table 4.1: Two different architectures for learning local convolutional feature descriptors. 2*conv* is more shallow with larger filters, and 4*conv* deeper with smaller filters. Note that configurations of the 4*conv* form have been reported to outperform configurations of the 2*conv* form in convolutional networks used for large-scale image recognition [Simonyan and Zisserman, 2014].

2*conv*

Convolutional Layer	1	2
Filter size	7x7	6x6
Out channels	32	64
Nonlinearity	tanh	tanh
Max pooling	2x2	2x2

4*conv*

Convolutional Layer	1	2	3	4
Filter size	3x3	3x3	3x3	3x3
Out channels	32	64	128	32
Nonlinearity	tanh	tanh	tanh	tanh
Max pooling	2x2	2x2	2x2	2x2

used, which was also previously reported by [Hoffer and Ailon, 2014] for classification accuracy. Thus, since learning with triplets leads to better results overall, and the shallow network is more suitable for learning with triplets, for our experiments we will use the 2*conv* network that leads to a 40% improved efficiency in both training and testing time.

4.2.6 Training details

In order to generate the training data for our experiments, we sample $5M$ triplets on-the-fly using patches from the Photo Tourism dataset. We do not use data augmentation as is typical in convolutional neural networks (CNNs) for general classification or convolutional feature descriptors [Zagoruyko and Komodakis, 2015,

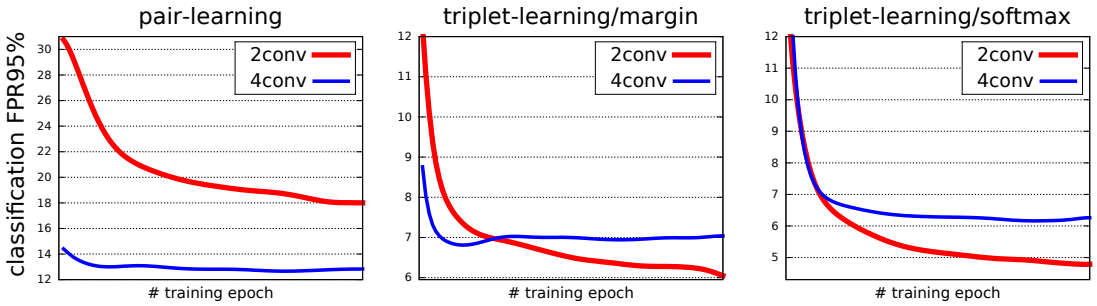


Figure 4.5: Comparison of a shallow architecture with large convolutional filters, and a deep architecture with small filters for the task of learning convolutional feature descriptors. Note that in both cases the networks are trained with exactly the same training set. Interestingly, the shallow network performs better when learning with triplets, while the deeper network over-fits easily. On the other hand, the deep network is more suitable when learning with pairs. Note however, that the general performance of the pair-based learning using the contrastive loss method (Equation (4.2)) is very limited compared to the triplet based methods.

Han et al., 2015]. When forming a triplet, we choose randomly a positive pair of patches that originate from the same physical point and a randomly sampled patch from another keypoint. This is in contrast to other works where carefully designed schemes of choosing the training data are used in order to enhance the performance. For the optimisation the Stochastic Gradient Descent (SGD) Bottou [2012] is used, and the training is done in batches of 128 samples, with a learning rate of 0.1 which is temporally annealed. In addition, we use momentum of 0.9 and weight decay of 10^{-4}. The convolutional methods are from the NVIDIA cuDNN library Chetlur et al. [2014]. The training of a single epoch with 5 million training triplets takes approximately 10 minutes on an NVIDIA Titan X GPU.

4.3 Experimental evaluation

In this section we evaluate the proposed local feature descriptor within the two popular benchmarks in the field of local feature descriptors. We compare our method to SIFT [Lowe, 2004], convex optimisation [Simonyan et al., 2014] and the

recently introduced convolutional feature descriptors MatchNet [Han et al., 2015], DeepCompare [Zagoruyko and Komodakis, 2015] and DeepDesc [Simo-Serra et al., 2015], which are currently the state of the art in terms of matching accuracy. The original code provided by the authors was used in all the experiments while for SIFT we use the `vl_feat` library [Vedaldi and Fulkerson, 2008].

We train our 2*conv* network with both margin ranking and ratio triplet losses, and either with or without in-triplet hard negative mining with dynamic anchor swapping. These four configurations, result in the following four variants: `TFeat-ranking` for the networks learnt with the ranking loss, `TFeat-ranking*` for the networks learnt with the ranking loss with anchor swap, `TFeat-ratio` for the ratio loss, and `TFeat-ratio*` for the ratio loss with anchor swap.

The evaluation is done with the two different evaluation metrics also previously used in Chapter 3, patch pair classification success in terms of ROC curves, and mean average precision in terms of correct matching of feature points between pairs of images [Mikolajczyk and Schmid, 2005]. Note that in the experiments below, we report the mAP instead of plotting the full graphs, which is the area under the precision-recall curve, and can be thought as an average performance across all testing sequences.

4.3.1 Patch pair classification

The patch pair classification benchmark measures the ability of a descriptor to discriminate positive patch pairs from negative ones in the Photo Tour dataset. For the evaluation we use the proposed $100K$ patch pairs as defined in the author's benchmarking protocol. Note that DeepDesc [Simo-Serra et al., 2015] does not report performance with training based on a single dataset, therefore for each test set, the training is performed on the other two datasets.

The results for each of the combinations of training and testing using the three subsets of the Photo Tour dataset are shown in Table 4.2 including the average across all possible combinations. Our networks outperform all the previously introduced single-scale convolutional feature descriptors, and in some cases with large margins except from one training-test combination where the 4096-dimensional version of MatchNet outperforms our TFeat variants. However, even in this case,

the version of MatchNet with comparable dimensionality to our descriptors is outperformed by three of our variants. Also note that MatchNet is specifically designed for patch pair classification, since it also includes a similarity metric layer trained on top of the feature layer. Interestingly, when we use the metric layer of MatchNet in a task different than classifying pairs of positive and negative pairs (e.g. NN patch retrieval), the results deteriorate significantly.

Note that we include the $2stream$ version of the DeepCompare descriptor, which operates on two difference scales, since it crops a smaller 32×32 patch inside the larger 64×64 input. This is not a fair comparison, since multi-scale approaches introduce information from different samples in the scale-space in the description process, something that has been shown to lead to significant improvements in terms of matching accuracy [Dong and Soatto, 2014]. Such approach can be used for various descriptors (e.g. MatchNet-2str, TFeat-2str, DeepDesc-2str), resulting in improved performance for each individual descriptor. Thus a fairer comparison would be to compare the $2stream$ DeepCompare architecture with our own $2stream$ implementation.

Interestingly, our single scale version outperforms the two scale version of DeepCompare, something which demonstrates the discriminative power of a convolutional feature descriptor learned with our shallow network and our triplet based losses.

It is also worth noting the fact that our method performs significantly better than the DeepDesc descriptor which involves hard negative mining. This implies that a simpler configuration with a better choice of loss function and training architecture, can outperform computationally expensive hard negative mining, in a significant fraction of the time needed for training.

In terms of the performance differences between our different variants, we can see that all our variants perform better than the state of the art, with the *Tfeat-margin** being slightly better than the others. In addition, we can observe that the use of the dynamic anchor swap leads to apparent improvements, which are more significant for the ratio loss. Note that the results reported are for the training-converged versions of our descriptors, after usually 50 to 100 epochs. Both the ratio and margin versions do not suffer from any over-fitting issues, since they

perform well in all combinations of training and testing datasets.

4.3.2 Nearest neighbour patch matching

To measure the nearest neighbour matching performance, we establish correspondence ground truth using the homographies and the overlap error from Mikolajczyk and Schmid [2005]. We consider two feature points between the two images in correspondence if the overlap error between the detected regions is less than 50%. Note that a region from one image can be in correspondence with several regions from the other image. Each image has an associated set of approximately 1K patches. More specifically, for each patch from the left image we find its nearest neighbour in the right image. Based on the ground truth overlap we identify the false positives and true positives, and generate precision-recall curves. The area under the precision-recall curve is the reported mean-average precision similarly to previous works in the area of convolutional feature descriptors. For this experiment, we use the `vl_benchmarks` Vedaldi and Fulkerson [2008] library (`vl_covdet` function), with some minor modifications to limit the descriptors extracted from an image to one thousand, which is important to avoid bias by different numbers of features in different images. For all the experiments below, the descriptors are trained on Liberty-DoG patches.

For the nearest neighbor matching protocol two datasets are mainly used in the literature, *Oxford matching dataset* Mikolajczyk and Schmid [2005], which is of small size, but includes images acquired by a camera in a real-world conditions, and the *generated matching dataset* Fischer et al. [2014] which is much larger in volume but is created synthetically. In the following sections, we first discuss our findings on the differences of the ratio and the margin ranking losses, and subsequently, we perform matching experiments comparing our four variants with the state of the art in terms of mean average precision.

Ratio loss vs. margin loss

As discussed above, the patch pair classification experiment shows that there are no over-fitting issues for both the margin and the ratio losses. We are interested in examining the behaviour of these losses in terms of nearest neighbour matching,

Table 4.2: Results form the Photo-Tour dataset. Numbers are reported in terms of *FPR95* following the state of the art in this field (see text for more details). *Italics* indicate the descriptors introduced here, and **bold** numbers indicate the top performing descriptor. Yos:Yosemite, Lib:Liberty, Not:Notredame. SIFT [Lowe, 2004], ImageNet-4*conv* [Fischer et al., 2014], ConvexOpt [Simonyan et al., 2014], DeepCompare [Zagoruyko and Komodakis, 2015], DeepDesc Simo-Serra et al. [2015], MatchNet [Han et al., 2015]. Note that our networks perform better than the state of the art, outperforming multi-scale architectures (DeepCompare-2str), joint learning of feature and metric layers (MatchNet), and hard negative mining (DeepDesc), while being much faster to extract and to train than all of the above.

Training		Not	Lib	Not	Yos	Yos	Lib	
Testing		Yos		Lib		Not		
Descriptor	#							mean
SIFT	128	27.29		29.84		22.53		26.55
ImageNet$_{4conv}$	128	30.22		14.26		9.64		18.04
ConvexOpt	80	10.08	11.63	11.42	14.58	7.22	6.17	10.28
DeepCompare $_{siam}$	256	15.89	19.91	13.24	17.25	8.38	6.01	13.45
Deepcompare $_{siam2str}$	512	13.02	13.24	8.79	12.84	5.58	4.54	9.67
DeepDesc	128	16.19		8.82		4.54		9.85
MatchNet	512	11	13.58	8.84	13.02	7.7	4.75	9.82
MatchNet	4K	8.39	10.88	**6.90**	10.77	5.76	3.87	7.75
TFeat-ratio	128	**8.32**	**10.25**	8.93	**10.13**	**4.12**	**3.79**	**7.59**
*TFeat-ratio**	128	**7.24**	**8.53**	8.07	**9.53**	**4.23**	**3.47**	**6.84**
TFeat-margin	128	**7.95**	**8.10**	7.64	**9.88**	**3.83**	**3.39**	**6.79**
*TFeat-margin**	128	**7.08**	**7.82**	7.22	**9.79**	**3.85**	**3.12**	**6.47**

which is a significantly different setting than the one used for both training and testing in the patch pair classification experiments.

Figure 4.6 shows the performance of the same network trained for the same number of epochs on the Liberty dataset with the two different triplet loss functions. It can be observed that the margin based loss increases the performance as more epochs are used in the training process. While no over-fitting is noticed when training and testing on patch classification (e.g. training with ratio loss on Liberty and testing with ratio-loss on Yosemite or Notredame). Interestingly, the ratio loss seems to have a decreased performance in patch matching as the network is trained for more epochs.

This also hints that other methods from the literature that were only tested in the patch classification scenario, may not perform well in matching. In our view, this shows that evaluating descriptors only in terms of ROC curves is not representative for realistic matching scenarios. More details on this subject can be found on Chapter 5.

Finally, we observe the results show that the loss functions with anchor swapping perform better than without swapping, which was also verified in the patch classification experiments. It further confirms that this simple technique can lead to improved results with no additional computational overhead and we argue that it should be adopted in all triplet-based learning optimisation problems.

Keypoint matching

Figure 4.7 presents the mAP results for Oxford benchmark, across all image sequences from the Oxford dataset, for two different keypoint detectors, DoG and Harris-Affine. Note that all networks are trained on DoG keypoints. In the case of our ratio loss, we use the networks from the first epoch, since all the next epochs would exhibit lower performance (cf. Figure 4.6), while for the margin methods we use the 50^{th} epoch. In the case of the DoG keypoints, our networks outperforms all the others in terms of mAP. The second best performing descriptor is the *DeepDesc* descriptor from Simo-Serra et al. [2015] which is based on a deep network and hard negative mining. Notably, the *DeepDesc* descriptor is well below the state of the art in terms of ROC curves and FPR95 as shown in Table 4.2.

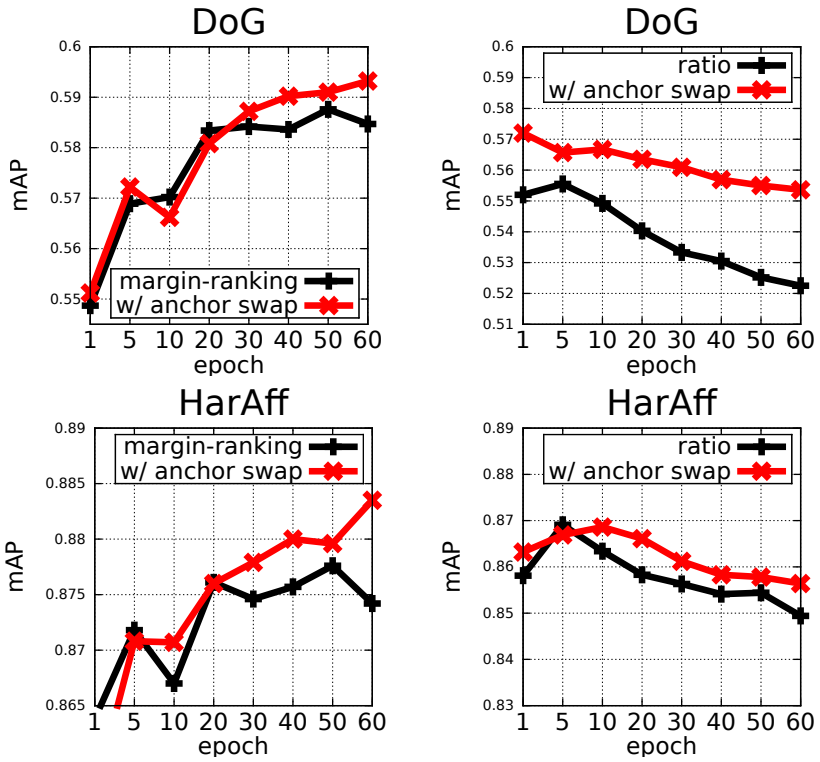


Figure 4.6: Average mAP for the 8 sequences of the Oxford matching dataset, in function of the number of epochs that the networks were trained using the training data from the Photo Tourism patches dataset. Ratio based loss function overfits in the process of separating the positive and negative pairs within a triplet, and does not perform well in the nearest neighbour matching experiment. On the contrary, learning with triplets and margin ranking does not suffer from this problem which shows that ranking methods are more suitable for nearest neighbour matching scenarios. Additionally, it can also be observed that the anchor swapping method results in consistent improvements across all methods.

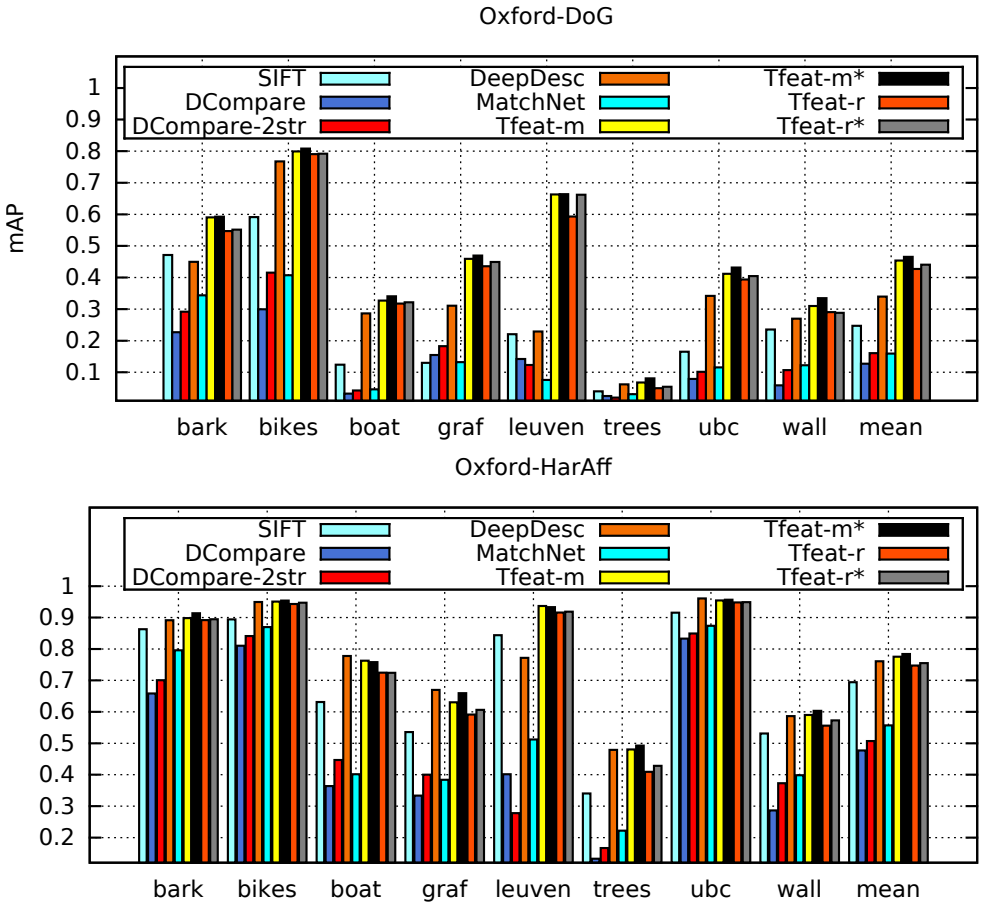


Figure 4.7: Evaluation on the Oxford image matching dataset, for two different types of feature extractors, DoG and HarrisAffine. Our *TFeat* variants outperform the state of the art in all cases.

This confirms our findings that the classification benchmark is not a representative measure for the common real-world application of descriptors which often relies on nearest neighbor matching. When using Harris-Affine keypoints our descriptor still outperforms the others, although with a smaller margin.

We demonstrated that our proposed descriptors are consistently on the level of the state of the art for both benchmarks, something which validates the fact that the optimisation with triplets, and our shallow networks lead to better generalisation properties.

Image transformations - synthetic dataset

Figure 4.8 shows the results across various synthetic transformations of image pairs as found in the Synthetic matching dataset from [Fischer et al., 2014]. Note that this dataset is generated synthetically and is not based on captured images. It can be observed again that our descriptor gives the top scores in most sequences. It is also worth noting, that even though this dataset has some extreme deformations as well as nonlinear filtering, the overall performance for both types of feature extractors is higher than for the Oxford dataset. This shows that synthetic deformations are less challenging for descriptors than some real-world changes as the ones found in Oxford dataset.

Qualitative results

In Figure 4.9 we can see some very challenging cases our descriptor was able to identify as true positive matches (top). It is robust to significant blurring, and extreme affine projections. In addition, it is clear that our descriptor is able to generalise well from the Liberty dataset to other matching datasets. We also show some examples of the false positive matches in the bottom row. It is worth noting that most of the false positive matches are very difficult to distinguish, thus confirming the expected behaviour of our descriptor.

Figure 4.10 visualises two layers of the CNN for the proposed Tfeat-ratio* as well as for DeepCompare Zagoruyko and Komodakis [2015]. Convolutional filters of Tfeat seem to be more smooth e.g. more regularised compared to DeepCompare. We believe it is the effect of simultaneous use of positive and negative pairs in the loss function during training, together with the smaller network that discourages overfitting. The difference becomes ever more significant in the second layer of the network.

4.3.3 Computational efficiency

One of the main motivations behind the work introduced in this Chapter, was the need for a fast and practical feature descriptor based on CNN. Indeed, the proposed convolutional feature descriptor is very efficient in terms of both training

Figure 4.8: Evaluation on the generated-matching dataset for two different types of feature extractors, DoG and HarrisAffine. Note that while there are some very difficult deformations for the descriptors to cope with, the general performance is better than the one reported for the Oxford dataset. This indicates the significantly more challenging nature of real world datasets compared to the synthetically generated ones, due to the fact that they capture all the factors that make matching process more challenging such as illumination changes and reflections.

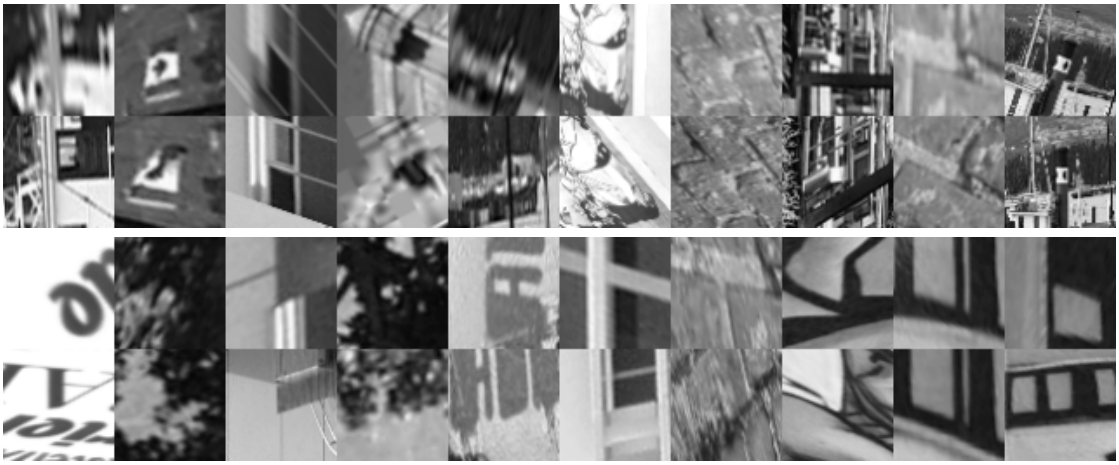


Figure 4.9: Examples of true (top) and false positive (bottom) nearest neighbour matching our large patch matching dataset. Note the extreme variations that the proposed descriptor can cope with, ranging from rotations and blurring to jpeg artefacts and full affine transformations.

and extraction time.

In Figure 4.11 (left) we present results for the extraction times with respect to the dimensionality. For the GPU implementations of the deep networks, all experiments were done with an NVIDIA GTX TITAN X GPU. When compared with the recently proposed deep feature descriptors, the proposed shallow network is both faster and smaller in dimensionality, while at the same time performs better in the benchmarks. Note that both axes are in logarithmic scale.

In addition, we show the extraction times for several descriptors in Figure 4.11 (right). From these measurments, we can conclude that the GPU version of the TFeat is close to the speed of the CPU implementation of BRIEF. This gives a significant advantage over the previously proposed descriptors and makes CNN based descriptors applicable to practical problems with large datasets, or very fast processing requirements. In more detail, the rate of $10\mu s$ per patch, allows for extracting thousands of feature descriptors from an images in real time.

In terms of GPU implementations of the popular CPU descriptors, several works have attempted to port SIFT to GPU such as Sinha et al. [2011], with speedups ranging from 5 to 20 compared to the CPU version. Even when con-

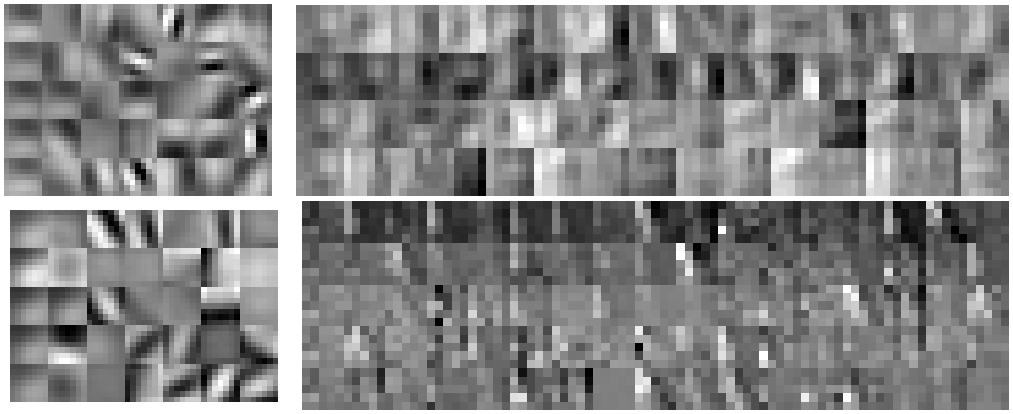
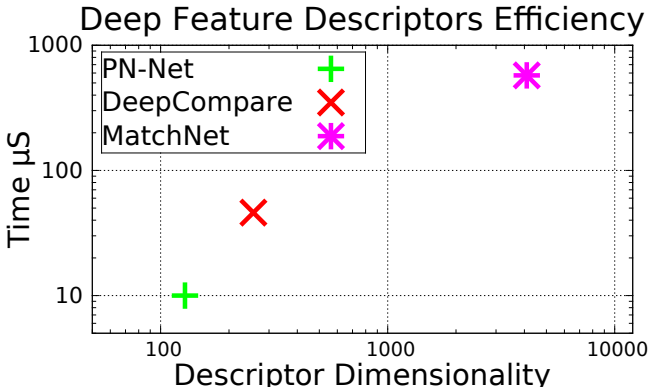


Figure 4.10: The weights learned in the first layer (left) and in the second layer (right) of the CNN. (Top) is our shallow network and (bottom) is DeepCompare. We see that our shallow architecture paired with our triplet based-loss, results in smoother filters especially in the second layer.

sidering such speedups, the proposed descriptor is still faster to compute mainly due to the convolutional operations libraries Chetlur et al. [2014], which are highly optimised for GPU execution.



	≈ time μs
SIFT (CPU)	400
BRIEF (CPU)	3
Ours	10
MatchNet	575
DeepCompare	46
DeepCompare	

Figure 4.11: (left) Dimensionality and efficiency of the proposed feature descriptors compared to other methods. We report the time required to extract a descriptor from a single input patch in μs. Note that both axes are in logarithmic scale. (right) We note that the GPU version of our implementation approaches the efficiency of BRIEF which is the fastest CPU descriptor available.

Note that the proposed descriptor also has an advantage in the computational efficiency during training time. While other works mention that their optimisation methods take from several hours Simonyan et al. [2014] to several days Zagoruyko and Komodakis [2015], Han et al. [2015], our work reaches the state of the art performance in 40-60 training epochs, which translates to 2-5 hours of training on a single state of the art GPU. Surprisingly even after a single epoch, i.e. after two to five minutes of training, we get a descriptor that performs very close to the state of the art.

4.3.4 Conclusion

This chapter introduced a new approach to training convolutional neural networks for extracting local image descriptors in the context of patch matching. The presented work shows that a combination of a triplet loss and a shallow architecture, results in a more discriminative descriptor, faster learning and faster execution. We show that due to these properties the proposed network is less prone to overfitting and has good generalisation properties. In addition, the high computational cost of hard negative mining has been successfully replaced by the very efficient triplet based loss.

We also demonstrate that ratio-loss based methods are more suitable for patch pair classification, and margin-loss based methods work better in nearest neighbour matching applications. This indicates that a good performance on patch classification does not necessarily generalise to a good performance in nearest neighbour based frameworks. We will also refer to this problem in the next chapter, where we critically discuss the feature descriptor evaluation methods.

Chapter 5

A large scale benchmark for evaluating feature descriptors

One of the most important issues in the context of evaluating novel methods is the existence of clearly defined, large scale and reproducible benchmarks. The field of local feature descriptors has seen remarkable growth in the recent years, leading to a significant amount of works that provide experimental evidence of improving over the state of the art, using a set of the most common datasets and benchmarks in the field. Unfortunately as we illustrate in this chapter, there is little consistency in most of the published work, due to the lack of a strict protocol for the experimental process.

In this chapter, we first illustrate the disagreement between several feature descriptor evaluations in terms of the state of the art across multiple published works, and we discuss some possible reasons for such significant differences in the reported results. Subsequently, we briefly discuss the existing benchmarking methods and datasets in terms of their strengths and drawbacks. Lastly, we introduce a novel large-scale dataset suitable for both learning and evaluation of local feature descriptors. This is the only dataset that exhibits many important properties for benchmarking local feature descriptors.

5.1 Critical discussion of commonly used datasets & metrics

In this section we discuss the commonly used metrics and datasets for evaluating descriptor performance in the literature, and we identify issues that might limit the reproducibility and interpretation of the results reported in several previous publications. We discuss our findings in two benchmarking methods that are prevalent in the field, namely patch pair classification and image matching.

5.1.1 Image matching

The first dataset that was introduced for this type of evaluation was the Oxford image matching dataset[1] [Mikolajczyk and Schmid, 2005]. The dataset consists of a set of 8 image sequences, each containing 6 images. Homographies are known between the first and the subsequent images, resulting in 5 image pairs per sequence where the first image is used as a reference. Using local feature *detectors*, a set of local features frames (recall definition from section 2.1) are extracted from both images, and a set of descriptors is computed around these frames. Subsequently, brute-force matching between the two sets of descriptors is applied, and two feature frames are considered a match if the distance is below a threshold t. If their frame overlap (for example the intersection over the union of two elliptical frames) is more than a specified threshold (typically 0.5), then the matching is considered as a true positive. Altering the value of t, results in different confusion matrices, indicating the values for false positives, false negatives, true positives and true negatives. These values can be subsequently used to create precision recall curves and compute the mean average precision, such as the ones presented in the experimental evaluation sections of the previous chapters in sections 3.5.1 and 4.3.1.

The Oxford image matching dataset is suitable for evaluating the performance of local feature *detectors*, which is a process that is only concerned with the repeatability of feature frames across different viewpoints. Unfortunately, it is not clear how to adapt this benchmark to a method that only evaluates the discrim-

[1] http://www.robots.ox.ac.uk/~vgg/research/affine/

inability of the feature descriptor, mainly due to the fact that there is no defined protocol on how to extract measurement regions centred around feature frames. In addition, no predefined set of feature frames is provided with the dataset so that researchers can reach reproducible results and directly compare across different works. Subsequently, each new work firstly extracts independently a set of local feature frames using a detector, and then compute and *evaluate* descriptors around this particular set of frames.

However there are differences between the different detectors in terms of the characteristics of the frames that they extract, for example, differences in scale, dominant orientation estimation, and number of frames detected per image. Thus, this makes it very difficult to design a *descriptor* evaluation benchmark that is fair and is reproducible. In fact, even within the different configurations of a singular detector, several factors such as the number of feature points detected, the non-maxima suppression process that filters overlapping features, and different values of threshold that identify interesting areas in the image, can significantly alter the results.

In addition, even with a clearly defined common set of feature frames for each image, results would still vary, since different works use different methods to extract a normalised patch around the frames, in order to compute the descriptor. For example, three crucial parameters that affect the matching performance have been shown to be the enlargement factor of the measurement area of a feature frame that incorporates more information into descriptors and the blurring of the normalised patch. More information and discussions on this subject can be found in [Vedaldi and Fulkerson, 2008].

To illustrate this issue more clearly, we collect some results from the state of the art works that focus on evaluating the best performing feature *descriptors*. To show the incosistency, we present results in terms of the best performing method between pairs of feature descriptors (e.g. SIFT versus LIOP, BRISK versus BRIEF etc.) in Table 5.1. Obviously, if the evaluations were based on a well defined protocol, the results would not be inconsistent, and would only indicate the performance of the description process.

We observe that different works greatly disagree on the results in terms of best

performing methods, and surprisingly, the state of the art may change depending on the particular reference cited. From this, it is obvious that no clear answer can be given on the best performing descriptor, since different works incorrectly assume that the the quality of the description process is equivalent to the quality of the function $f(\boldsymbol{x}) \in \mathbb{R}^D$ that maps a normalised input patch to the $\boldsymbol{x}$ D-dimensional descriptor space.

To further prove this point and discuss the extent of variability such changes can introduce, we present quantitative experiments that illustrate the variability of the descriptor performance in relation to two configurations of the underlying factors that control the patch description, the enlargement factor of the measurement area and the patch normalisation process.

Enlargement factor

The feature frames extracted using a detector can be relatively small in terms of pixel area, and thus it is a common method to augment the area of the keypoint, by enlarging the size isometrically. This scaling factor, that we will denote ρ is not a fixed factor, and more importantly, is often set experimentally.

Unsurprisingly, this factor can have a significant effect on the performance of the descriptor, since using small factors can lead to patches that are not very discriminative and do not include significant information, while using large values results in patches that are not local, and thus vulnerable to occlusions and scene changes.

To show the effect this parameter can have on the final descriptor performance, we present in Table 5.2 the mAP results for a matching experiment using SIFT with different values of ρ. This experiment is conducted on the Leuven sequence of the Oxford dataset using a standard DoG detector. It is clear, that different values of ρ can greatly affect the performance, with the ratio of the best to the worst performing parameter reaching approximately 3.

Unfortunately, there is no clearly defined protocol for the value of ρ, and different implementations use different values. In addition, several works do not report the magnification factor that was used in their experiments, thus making it difficult to reproduce the experiments.

Table 5.1: Inconsistency of the evaluation results in published works on feature descriptors. Note that for all the references below, we report the results based on the same evaluation metric, i.e. the performance in terms of precision recall curve when matching sets of patches with nearest neighbours techniques, on the Oxford matching dataset. It is clear that although all these works are concerned *only* with feature description, there are hidden detector and patch extraction based factors that lead to such results.

LIOP outperforms SIFT	SIFT outperforms LIOP
[Miksik and Mikolajczyk, 2012]	[Tsun-Yi Yang and Chuang, 2016]
[Wang et al., 2011b]	

BRISK outperforms SIFT	SIFT outperforms BRISK
Leutenegger et al. [2011]	[Levi and Hassner, 2016]
Miksik and Mikolajczyk [2012]	

ORB outperforms SIFT	SIFT outperforms ORB
Rublee et al. [2011]	Miksik and Mikolajczyk [2012]

BinBoost outperforms SIFT	SIFT outperforms BinBoost
[Levi and Hassner, 2016]	[Balntas et al., 2015]
[T. Trzcinski and Lepetit, 2013]	[Tsun-Yi Yang and Chuang, 2016]

ORB outperforms BRIEF	BRIEF outperforms ORB
[Rublee et al., 2011]	[Levi and Hassner, 2016]

Table 5.2: The effect of the enlargment factor (ρ) for the patch measurement area has on the SIFT descriptor mAP, for the Leuven sequence of the Oxford dataset. For this experiment we use the DoG detector. 1|X represents the result between the first and the X image in the sequence.

ρ	1|2	1|3	1|4	1|5	1|6
1	0.31	0.13	0.05	0.03	0.01
2	0.46	0.23	0.09	0.06	0.04
4	0.68	0.44	0.24	0.15	0.11
8	0.74	0.57	0.43	0.32	0.24
12	0.80	0.67	0.54	0.42	0.35
20	0.87	0.77	0.69	0.55	0.50

Case study: implementation method

We focus on two different functions included in the `vl_feat` library: `vl_covdet` and `vl_sift`, with both functions being able to compute SIFT descriptors. The motivation behind this experiment is that some works [Simonyan et al., 2014, T. Trzcinski and Lepetit, 2013] use the `vl_covdet` method which first extracts a normalised patch and subsequently computes a descriptor, while others [Tsun-Yi Yang and Chuang, 2016, Dong and Soatto, 2014] use the default SIFT extraction process (`vl_sift`). Note that the two functions differ in parameters such as the enlargement factor, use of image pyramids, number of dominant orientations extracted for a feature frame and several other factors related only to the detection and patch extraction process. On the contrary, the description process is identical in both cases.

We can clearly see from Table 5.3 that the differences between the `vl_sift` and `vl_feat` methods, can greatly affect the results. For example, our proposed TFeat descriptor outperforms the `vl_sift` implementation in two out of five pairs. However, a fair descriptor comparison can only be performed between TFeat and SIFT when using the `vl_covdet` method, since this is the only case where both descriptors are extracted from identical patches. Clearly, since the `vl_sift` method acts on different patches, there can be no meaningful direct comparison between

Table 5.3: Comparing two different methods that are used in the literature for extracting SIFT descriptors with the TFeat descriptor proposed in Chapter 4. Results are presented for the Leuven sequence. While the `vl_sift` version of SIFT outperforms our Tfeat in 3 of the image pairs, note that the results between Tfeat and SIFT are only comparable with the `vl_covdet` function, due to the fact that both act on the same normalised patch. Note that the `vl_sift` version has several configuration options such as pyramids and multiple detection of dominant orientations. However, in order for descriptor evaluations to be reliable, they should act on a set of predefined and normalised identical raw patches, which is possible by applying the SIFT descriptor directly on extracted normalised patches and subsequently comparing with the other descriptors.

descr	1|2	1|3	1|4	1|5	1|6
SIFT `vl_sift`	0.47	0.40	**0.46**	**0.41**	**0.48**
SIFT `vl_covdet`	0.32	0.14	0.18	0.1	0.076
Tfeat-margin*	**0.68**	**0.48**	0.41	0.35	0.23

the two methods.

Other image matching datasets

Several other image matching datasets were introduced after the Oxford matching dataset, and we give a brief review as below.

The matching dataset [Fischer et al., 2014], is based on synthetically deforming images with various transformations. This has the advantage of easily generating a large set of test images. In total, a set of 16 images deformed in 26 different ways, result in 416 image pairs compared to the 48 pairs found in the Oxford dataset. However, as we have shown in Figure 4.8, this dataset is not very challenging compared to the Oxford matching dataset, presumably due to the synthetic nature of deformations.

The DTU [Aanæs et al., 2012] matching dataset is based on images extracted from a controlled experimental setup, consisting of constrained light and a small set of everyday objects placed in photography booths. Note that under such

controlled environment, the variability of the dataset is limited. For example, the background in all the images is uniformly black.

Another one, namely the Edge Foci dataset [Zitnick and Ramnath, 2011], is based on challenging scenarios of image pairs that change dramatically. However, we identify two problems with this dataset. Firstly, the scenes are not planar and thus cannot be used for accurate homography and ground truth estimation, and secondly that the content of the image pairs is not optimal for general descriptor evaluation but are mainly intended for detectors that can be robust to significant variation.

Note that all the above datasets are based on sequences of images, similarly to the Oxford dataset. Thus, they all suffer from the same limitations mentioned above, concerning a lack of well-defined protocol to extract patches from full images.

5.1.2 Patch classification

Another influential descriptor evaluation approach is based on the *patch classification* protocol introduced in [Winder and Brown, 2007]. In this protocol, a set of positive and negative pairs of patches are used in order to create a receiver operating characteristic curve (ROC). This protocol examines the feature descriptor as a classifier that produces two distinct distance probability distributions (for positive and negative pairs), and is concerned with the separation between the two distributions.

The patch pair classification evaluation was a significant contribution, since it allowed for large scale and reproducible experiments, with a clearly defined protocol. In addition, no detector is involved in the experimental process, since the patches are already extracted and pre-normalised. Finally, a set of ground truth pairs is provided by the authors, thus enabling subsequent works to directly compare to the state of the art. On the contrary, as we have demonstrated, this is not possible with the image matching protocol.

There are two large-scale datasets available for this evaluation protocol, namely the *Photo Tourism* dataset [Winder and Brown, 2007], and the *Stanford CDVS Patches Dataset* [Chandrasekhar et al., 2014]. Recently, a third patch dataset

Figure 5.1: Positive pairs from the CDVS (top) and Photo Tourism (bottom) datasets. The CDVS patch pairs contain clear and well aligned shapes which make them less challenging compared to the Photo Tourism dataset. This result can also be verified experimentally as we show in Figure 5.2.

focusing on patch classification and retrieval was introduced [Paulin et al., 2015], however it is of very limited size. Despite the fact the above patch-based datasets solve the problem of consistent patch extraction from feature frames, they do suffer from some significant limitations. Below we discuss a series of issues related to both the evaluation protocol and the datasets that are used.

Existing patch classification datasets are not challenging

In Figure 5.1 we show some positive patch pairs from the CDVS and Photo Tourism datasets. We observe that the pairs from the CDVS datasets are easily distinguishable, and present no severe deformations. On the other hand, the range of deformations in the Photo Tourism dataset is more significant, which range from incorrect estimation of principal direction to significant affine transformations and severe illumination variances. This is also experimentally verified in Figure 5.2 where we plot the ROC curves for the SIFT descriptor for both datasets. Note that the performance of SIFT on the CVDS reaches a lower error rate than the error rate of the best performing convolutional descriptors in the Photo Tourism dataset (section 4.3.1). From the results described, it is clear that further experi-

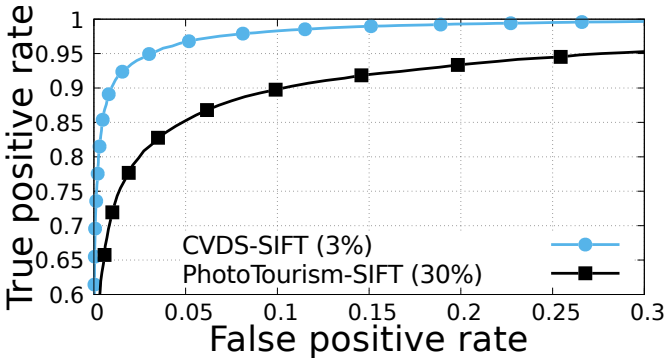


Figure 5.2: ROC curves for Photo Tourism and CVDS datasets. Note that the results in the CVDS dataset even with the SIFT descriptor which is not the state of the art, reach almost perfect separation. This experimentally validates the hypothesis that the CVDS dataset is not particularly challenging, even for a descriptor that does not represent the state of the art such as SIFT.

ments for the CVDS dataset will not produce any meaningful results representing the state of the art, especially regarding the modern convolutional feature descriptors which are expected to dominate the future work. However, such a dataset might still be of merit, in terms of evaluating the state of art of lower accuracy methods such as binary descriptors.

In addition, as experiments have shown in section 4.3.1, modern convolutional feature descriptors have reached a significant performance level on the Photo Tourism dataset, and further improvements will not lead to a major score advancement. Thus, in terms of evaluating more vigorously the state of the art convolutional descriptors, currently there exists no challenging large-scale evaluation benchmark.

Limited visual variability in existing datasets

The Photo Tourism dataset is comprised of three distinct collections of patches, namely the Liberty, Notre-Dame and Yosemite patch sets. The Liberty patches are extracted from images of a statue, the Notre-Dame patches from a building, and the Yosemite dataset from a mountain. Thus, the patches do not represent a large

set of possible variations across different physical objects, and can be criticised for having limited variability. In addition, structures such as buildings or mountains, contain many self-similar areas that might significantly affect both the learning and the testing process. This issue was identified as one of the driving factors behind the introduction of the CVDS dataset which included patches extracted from a large set of distinct objects. However, as we have seen, the CVDS consists of non-challenging patch pairs, thus reducing its utility.

ROC curve results are inconsistent with matching scores

ROC have been sucessfully used as indicator of performance in the machine learning community [Davis and Goadrich, 2006, Fawcett, 2004], and provide valuable insights in terms of the discriminability of a classifier. Note that such curves represent the False Positive (FP) versus the True Positive (TP) values for different distance thresholds and show how well a feature descriptor acts as a classifier that discriminates a set of positive and negative pairs. However, the matching process is by nature different and heavily unbalanced. Thus, ROC curves are not necessarily good indicators of the actual Nearest Neighbour (NN) matching performance of a feature descriptor.

For example, as we have seen in section 3.5.2, while BinBoost presents a 30% performance gain compared to SIFT in the patch classification experiment, in a real world matching scenario it performs poorly. Another notable example is while the DBRIEF descriptor [Trzcinski and Lepetit, 2012] performs well in terms of ROC curves, it gives very low mAP values(≈ 0.01) on the Oxford dataset according to the evaluations in [Levi and Hassner, 2016].

It is worth noting that certain works [Winder et al., 2009, T. Trzcinski and Lepetit, 2013, Trzcinski and Lepetit, 2012] *solely* evaluate several critical parameters of their methods based solely on the FPR95 value, which is given from the TP rate when the FP rate is equal to 0.95. However, reducing many design decisions to a single number, is not fully representative. In addition, there is no guarantee that two ROC curves do not cross, when only the FPR95 are examined. Finally, decision factors that are based on the ROC curves, do not necessarily generalise to the real world application of feature descriptors, which is matching.

From the above discussed issues, we argue that there is a need for a novel dataset that combines the strengths of the currently available datasets, while at the same time eliminating their limitations. We introduce such a dataset in the next section.

5.2 A reproducible and large scale benchmark of feature descriptors

In this section, we introduce a new benchmarking dataset that aims to adress the limitations of the commonly used datasets, and enable a large scale, rigorous and reproducible evaluation method for local feature descriptors.

Properties of a good benchmarking method for feature descriptors Below, we present some design goals in terms of a good benchmarking method for evaluating local feature descriptors, and we briefly discuss the underlying factor motivating each goal. These goals are designed upon the need to alleviate the weaknesses and further improve the strengths of the currently available datasets.

- *Patch-based*: Descriptors should be evaluated solely using normalised patches, in order to remove any detector related-factors from the evaluation process.
- *Diverse*: Patches should be extracted from a large set of distinct images to encourage visual diversity.
- *Reproducible*: Evaluation should be done on common sets of pre-extracted patches, in order to encourage reproducible results among researchers. This is inspired by the sets of training and testing data provided in the Photo Tourism dataset, which lead to standardisation of the reported performance across different works.
- *Real-world captured:* As we have presented, the use of a synthetic dataset results in less challenging benchmarks. Thus, the images need to be captured with a real camera, and represent real-world, challenging and with unconstrained illumination and view-point changes.

- *Large-scale*: The scale of the available data should make it possible for large scale evaluations, and more importantly, enable the possibility of learning for recent convolutional descriptors, that require vast amounts of data for training.

- *Multiple evaluation metrics*: The dataset should allow for different experimental evaluations such as patch matching, patch classification and patch retrieval.

To that end, we introduce a large-scale dataset of image pairs, manually collected using cameras and fully annotated with homography matrices among image pairs. Our dataset includes several challenging factors, such as illumination changes, reflections, viewpoint changes and temporal scene changes. In Table 5.4 we present a categorisation of the available feature descriptor evaluation datasets, and we show that the proposed dataset is the only one that combines the desired properties of being large scale, unconstrained, diverse and leading to reproducible results. Note that compared to the previously used equivalent real world matching dataset (Oxford matching), our dataset is 15 times larger, thus enabling a large set of potential novel uses, such as enabling deep learning of local feature descriptors and detectors.

The introduced dataset and the evaluation protocols together with the benchmarking software were can be accessed at https://github.com/featw. In the next sessions we present a brief discussion that indicates the significance of the introduced dataset, and we briefly illustrate as a proof of concept a novel training method that is made possible due to the richness of the annotation.

5.2.1 Dataset details

The introduced dataset consists of 130 image sequences, each of which contains 6 images, similarly to the Oxford matching dataset. Homographies are known between the first image in each sequence and the remaining 5, leading to a total of 650 image pairs. We extract 1500 feature frames per image, resulting in a total of almost a million distinct patches. Note that this is almost twice as large as the largest previously available patch dataset, Photo Tourism.

Table 5.4: Qualitative description of dataset attributes for a list of the most commonly used feature descriptor evaluation datasets. Photo Tourism [Winder and Brown, 2007], DTU [Aanæs et al., 2012], Oxford [Mikolajczyk and Schmid, 2005], Fischer [Fischer et al., 2014], CVDS [Chandrasekhar et al., 2014], Edge Foci [Zitnick and Ramnath, 2011], RomePatches [Paulin et al., 2015]. Note that the introduced dataset is the only one that is large scale, unconstrained, diverse and reproducible.

dataset	diverse	reproducible	real-world	large-scale	multiple metrics
Photo Tourism		✓	✓	✓	
DTU			✓	✓	
Oxford	✓		✓		
Fischer				✓	
CVDS	✓	✓	✓	✓	
Edge Foci	✓		✓		
RomePatches		✓	✓		
Ours	✓	✓	✓	✓	✓

To extract patches from the dataset we utilise the following protocol. We generate a set of feature frames from each image, using the `vl_convdet` function found in `vl_feat`, and we extract normalised patches from the feature frames using a scaling factor of 3 that was previously shown to give good results [Lenc, 2013]. We then apply all our evaluation benchmarks directly on the patches, without dealing with the full images. To help researchers test their descriptors on patches extracted from different detectors, we provide patches for DoG, $Harris$ and $HarrisLaplace$. Note that in all cases, we provide the affine normalised version of the frames, as it is described in [Vedaldi and Fulkerson, 2008].

Note that one of the main reasons behind the creation of the Synthetic Matching Dataset and the DTU dataset was the need to evaluate the descriptor performance on a large scale. However, both these datasets were captured in controlled and constrained environments. On the contrary, our dataset is the only one of this size, that is based on real-world images, which are captured and then annotated. In addition, as one can observe in Figure 5.3, our dataset consists of a large set of distinct objects and image conditions.

5.3 Description of evaluation protocols

In this section, we briefly review three different evaluation protocols that can be used with the proposed dataset. We provide software tools for all these benchmarks [2], and we also provide all the evaluation sets of already extracted patches. Note that the original images are also made available together with their respective homographies, to enable researchers to experiment with large-scale evaluation of feature detectors. Undoubtedly, the introduced dataset can lead to valuable insights in the field of feature detectors, since it is approximately 15 times larger than the previous ones.

5.3.1 Patch matching

The evaluation method is identical to the protocol for image matching described in [Mikolajczyk and Schmid, 2005]. The performance is measured by matching

[2] http://github.com/vbalnt

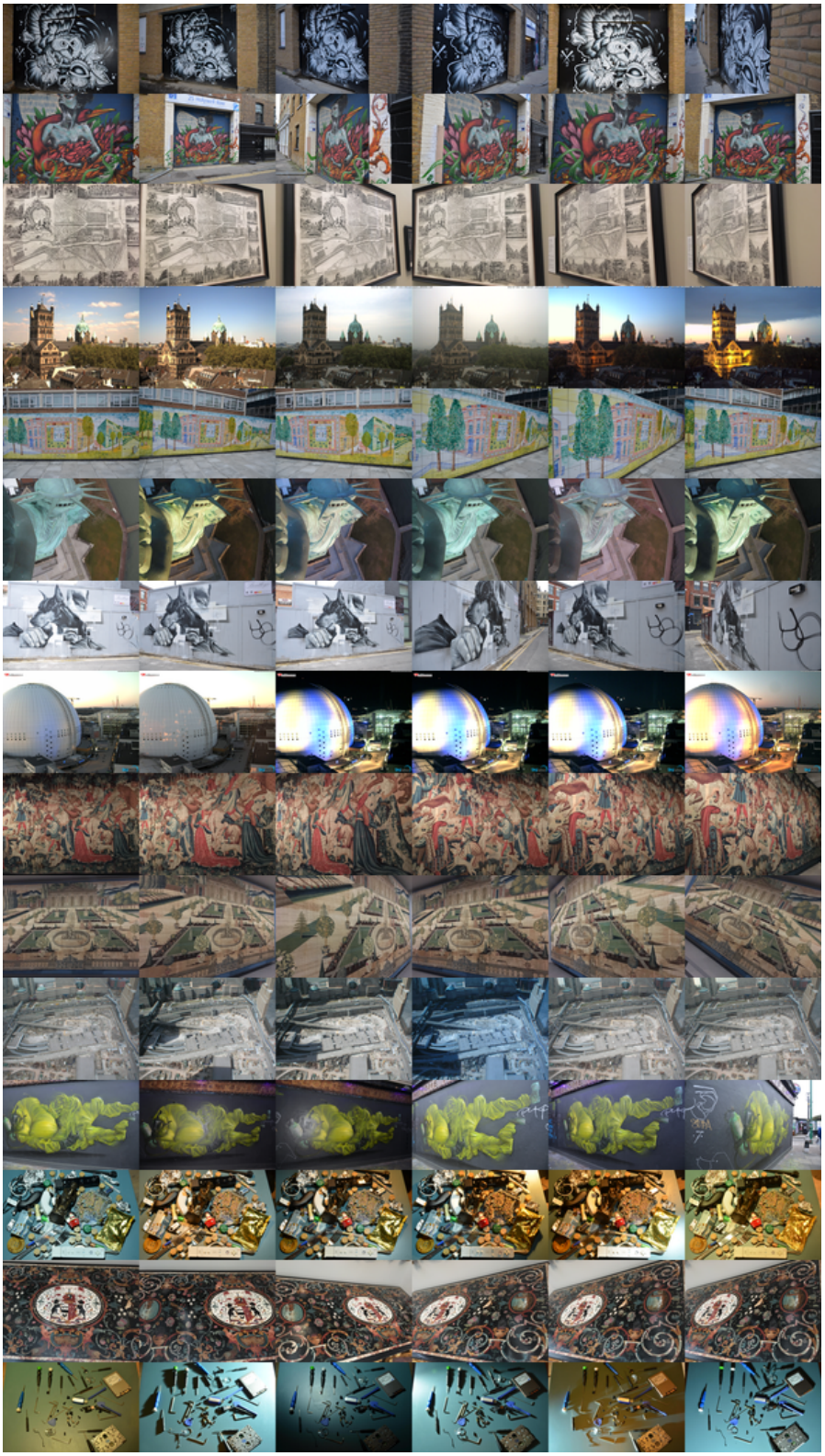


Figure 5.3: Sample sequences from our large-scale matching dataset. Note the extent and unconstrained nature of deformations that the sequences exhibit such as focus changes, reflections and non-linear illumination changes.

Table 5.5: Comparison of mean and std values of the mAP results on the introduced dataset compared to the previously used Oxford dataset, for the baseline of SIFT and DoG detector. Note that our dataset is significantly more difficult in terms of image nearest neighbour matching.

descr	mean mAP	std mAP
Oxford	0.48	0.26
Ours	0.21	0.19

the patches from one image to the other, through brute force matching to find the nearest neighbour. Comparing the true matches with the matches returned by nearest neighbour brute force matching, provides precision-recall curves. Note that in this case, the precision is measured in terms of the patch matches that can be matched in theory.

For each image, we provide the set of patches extracted, together with the overlap matrices. For each patch, we also save the feature frame associated with it, so that homography correspondences can be computed. A modified version of the *vl_benchmarks* library is provided for the computation of the precision recall curves. In addition, the benchmarking software also computes the mAP. Note that our software can be interfaced with any descriptor, thus allowing plug-and-play descriptor evaluation from any arbitrary programming language.

In Table 5.5 we show the mean and std dev values for the mAP values for our dataset, compared to the equivalent values from the Oxford matching dataset. For this experiment we use the patches extracted with the DoG method. Note that our dataset is significantly more difficult than the Oxford matching, with the mAP being 56% lower. This can be attributed to the wider range of deformations available in our dataset, that are not present in the Oxford data. For example, our datasets contains images captured across different times of the day, and with different weather conditions. Notably, our dataset also presents a slightly reduced variance between the mAP precision results, something than can be attributed to the more standardised method of collecting the data.

5.3.2 Patch classification

For this evaluation method, we adapt the Photo Tourism ROC based evaluation, for the diverse set of patches extracted from our dataset. Note that the comparison of two patch sets extracted from two images in terms of NN matching, will not result in the ROC curve leading to meaningful results, due to the fact that the positive and negative patch pairs are not balanced, and the ROC curve is not representative for heavily unbalanced data [Fawcett, 2004]. For example, considering a set of 1000 patches extracted from each image, and by hypothesising that the positive matched features overlap perfectly, there are 1000 positive pairs, and almost 1×10^6 negative pairs. The ROC curve has been shown to be not an informative measure in cases with heavily unbalanced data.

Thus, for the unbalanced patch classification benchmark, the evaluation should be done in terms of Precision-Recall curves, similarly to the work on [Simo-Serra et al., 2015]. Note that in this case, the negative pairs include all the possible negative combinations, and are not limited to only the matches that can theoretically be matched, as happens on the patch matching benchmark.

Note that for completeness, we also include a balanced version of patch classification in our dataset, in order to enable experiments using ROC curves so as to directly compare with previous works. The balanced version of our patch classification dataset is created by randomly selecting a subset of the negative matches that is identical in size to the positive matches, thus leading to equal amounts of positive and negative pairs. This also allows us to directly compare the difficulty between the proposed dataset and the previously used Photo Tourism dataset. To illustrate that our dataset is significantly more challenging, we show in Figure 5.4 the result of applying our state of the art TFeat descriptor on our dataset, compared to the most difficult subset of the Photo Tourism dataset, namely the *liberty* sequence. Note that the proposed dataset is significantly more challenging than the *liberty* dataset, something that is evident from the shape of the ROC curves.

In addition, in Figure 5.5 we show some examples of positive patch pairs from our dataset, to illustrate the more challenging scenarios compared to the previous datasets. For example, our dataset exhibits a large number of illumination changes, significant shadow variances, reflections, and temporal differences, factors that

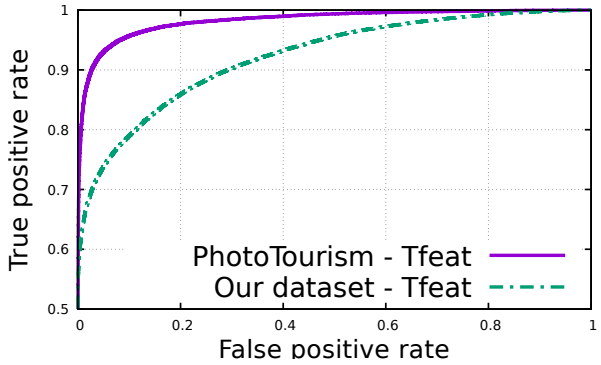


Figure 5.4: Patch classification ROC curves, for the proposed dataset and the most challenging Photo Tourism dataset (Liberty) for our state of the art Tfeat descriptor. Note that the introduced dataset is significantly more challenging than the previously commonly used dataset, something that is able to lead to more advanced feature descriptors.

make the description process much more challenging.

5.3.3 Patch retrieval

Inspired by the RomePatches dataset [Paulin et al., 2015], we also create a retrieval dataset that is concerned with the ability of feature descriptors to successfully rank relevant results for queries from a large set of patches.

For a pair of images representing the same scene in different conditions, we detect a set of K non-overlapping feature frames from one image, and using the homography matrix, we project the extracted frames to the remaining 5 images, and extract normalised patches around each projected frame. Repeating this process on N sequences, results in six sets of patches each of size $K \times N$. The first set of $K \times N$ can be used as queries, and the second set as the search set.

Note that this evaluation is theoretically possible also on the Photo Tourism dataset, since the authors provide around $100K$ classes (feature frames), with each containing on average 5 patches. However, practically one could not gain meaningful results from such an evaluation, since the classes are not guaranteed to be distinct,and thus there is no clear distinction between a false positive match and a true positive match. Our patch retrieval dataset solves this problem, since

Figure 5.5: Sample positive pairs from our large-scale challenging dataset. From top to bottom *calder*, *steps*, *tools*, *kions*. Note that reflections on the *calder* patches, the illumination changes on the *steps* and *kions* patches and the shadows and specularities on the *tools* patches. The range of deformations that exist in our dataset is significantly larger than the ones previously found on the Photo Tourism dataset.

each class is guaranteed to be distinct. There are some cases where repeating patterns could affect the false positive rates, but these cases are not a measurable proportion of our data.

The full dataset together with a sample implementation of the above evaluation metrics can be found in http://www.iis.ee.ic.ac.uk/~vbalnt/.

5.4 Application: Learning feature descriptors using feature frame overlaps

The large-scale nature of our dataset, is able to give rise to a plethora of novel training and testing methods. Unlike the Photo Tourism dataset, our data consist of both sets of patches and fully available homographies between those sets. Thus, we can enforce a quantitative measure in terms of patch-similarity, and abandon the commonly used class membership method (i.e. positive and negative patch pairs).

For example, let $\mathcal{S}_L$ represent a set of patches extracted from one image, and $\mathcal{S}_R$ extracted from another image, representing the same scene. In addition, let $\mathcal{F}_L, \mathcal{F}_R$ be the feature frames that are samples for the extraction of the normalised patches. Note that the homography H_{12} is known, and thus for each $f_L \in \mathcal{F}_\mathcal{L}$ and $f_R \in \mathcal{F}_\mathcal{R}$ we can compute a frame overlap measure $o(f_L, f_R)$. This allows us to have training data of the form $\{f_L, f_R, o(f_L, f_R)\}$. Subsequently, such training data can provide fine grained-metric learning that is based on feature frame overlaps, instead of qualitative distinctions between positive and negative pairs.

To illustrate the novel methods that are made possible using our large scale dataset, we show below the effect of a learning method that incorporates overlaps as a continuous quantitative label on the performance of the learning process.

We generate a dataset with $100K$ patch pairs, and their respective overlaps. Similarly to the commonly used method, we threshold the overlaps at 0.5, and we create positive and negative pairs. In this case, the learning method applied is identical to the contrastive loss from Equation 4.3. We define this method as *class membership learning*. We also introduce a separate learning method, namely *overlap learning* in which all the pairs with an overlap value larger than 0.5 are

treated as positives, and all the pairs with 0 overlap as negative. For the *overlap learning*, we modify the hinge contrastive loss as follows

$$l(\boldsymbol{x}_1, \boldsymbol{x}_2, o) = \begin{cases} o \cdot ||f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)||_2 & \text{if } o > 0.5 \\ max(0, \mu - ||f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)||_2) & \text{if } o \leq 0.5 \end{cases} \tag{5.1}$$

where μ is a margin parameter, and o is the overlap parameter.

Note that the addition of the quantitative o parameter, transforms the problem from a class membership problem (positive and negative pairs), to a more fine grained representation that also incorporates a similarity factor for overlapping patch and feature frame pairs. Intuitively, with the introduction of the o parameter, the loss is not only dependent on the $L2$ distance between the patches, but also their respective feature frame overlaps. The margin-based subsection of the loss for the negative patches remains intact.

In Figure 5.6 we plot the results on our balanced patch classification dataset for the class membership contrastive loss, and our overlap contrastive loss. Note that optimising the CNN with the *overlap learning* method, results in a better separation between the positive and negative pairs across all thresholds, as it is evident from the ROC curves.

5.5 Conclusion

In this chapter, we discussed the inconsistencies that previous work exhibit in terms of evaluating the state of the art feature descriptors. We investigate possible reasons for such inconsistency, and we identify the problem as the large set of detector and patch normalisation parameters, unrelated to the description process but nevertheless affecting the evaluation results.

In order to facilitate large scale and reproducible experiments, we introduce a novel dataset captured from real world data, which is more than an order of magnitude larger than the previous similar dataset, and significantly larger than other available and synthetically created dataset. In addition, we discuss three evaluation protocols that can be used in conjunction with the proposed dataset, and we provide evaluation software to enable researchers to directly compare results and identify the state of the art with reproducible experiments.

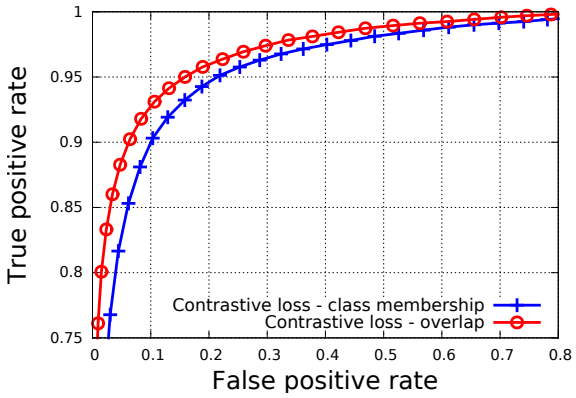


Figure 5.6: Comparison of learning with positive and negative pairs (class membership), and learning with a continuous overlap loss function. Note that while exactly the same data are used, learning with a continuous overlap loss leads to a better result, due to the fine grained nature of the learning process. Learning based on overlaps is made possible with our large scale dataset that also contains overlap information, instead of binary class membership labels.

Lastly, we show that the scale and richness of information in our dataset can lead to novel experiments and learning methods. As a proof of concept, we discuss such a novel learning process where the optimisation of the convolutional feature descriptors is based on feature frame overlaps, rather than on typically applied positive and negative patch pairs. Note that such a technique, could improve all descriptors that were previously learnt with positive and negative pairs as it provides a more fine-grained measure of similarity compared to a simple indicator of binary class membership.

Chapter 6

Conclusions

In this thesis, we investigate three different issues related to the extraction and matching of local feature descriptors. We focus on introducing novel description methods that are both efficient to compute and are the state of the art in terms of accuracy in their respective categories. We also identify the reasons that lead to inaccuracies in the evaluations of the previous works in the field, and to that end, we collected and annotated a large scale matching dataset. We introduce methods and software that lead to reproducible results, allowing for a fairer evaluation of feature descriptors.

We first proposed a method to elevate the performance of intensity-test based binary feature descriptors to the levels of the more computationally costly pooling based methods. This has been achieved by introducing a novel matching method between descriptors that masks specific binary tests that are more likely to flip sign under small deformations. Our method is of generic nature, and can be applied to other fields such as tracking, and to other types of features such as Haar-like features or local binary patterns. In addition, we show that the framework can be extended to floating point features. The masking of non-invariant features that we propose as a novel matching method, has been shown to be applicable directly to other tasks such as motion related features [Zhang et al., 2016] and scale related features [Tsun-Yi Yang and Chuang, 2016], which validates the merits of the proposed method.

Despite the fact that our per-patch adaptation of feature descriptors leads to

state of the art results in terms of fast, binary intensity based descriptors, the performance of such methods is very limited compared to the state of the art in local feature descriptors based on convolutional neural networks. Unfortunately, such methods are still computationally prohibitive for most applications. To that end, we introduce novel methods of training shallow convolutional networks, based on triplets, and we show our convolutional feature descriptors outperform the state of the art in terms of both accuracy, and computational efficiency. We hope that the efficiency and the accuracy of our methods will enable novel applications for convolutional feature descriptors, that were not possible before.

Finally, we turn our focus to the benchmarking protocols and datasets that have been used in the field to evaluate new methods and to identify the state of the art in local feature descriptors. We discuss several issues with the design and application of most benchmarks that prevent reproducibility. In order to both create rigorous evaluations, and to rectify the issues with the previously used datasets, we introduce a new large scale and challenging dataset. We anticipate that the strict benchmarking protocol based on normalised patches, will be a step towards a more systematic approach to evaluating local feature descriptors, decoupling the descriptor evaluations from the latent unwanted effects of any underlying detector parameters. In addition, the scale of the proposed dataset enables training of convolutional feature descriptors with deep learning methods and real world data, which has not been possible possible before. The challenge in the introduced dataset gives space for any future feature descriptor to improve upon the current low baseline, and is crucial for the further advancement of novel methods, as currently available datasets are becoming trivial for the modern methods. Lastly, we show that the design of our benchmarking dataset, gives rise to novel possible learning methods, that can lead to more discriminative descriptors.

Future work

Albeit the fact that the descriptors introduced in this thesis provide state of the art results in their categories, there are still many possible improvements to be made in all areas.

In terms of the locally adapted binary descriptors, a possibility of future work is

to investigate more advanced online feature selection algorithms from the relevant methods presented in the literature [Wang et al., 2014a, Pudil et al., 1994]. Such methods could be applied to other descriptor-related fields such as 3D [Wohlhart and Lepetit, 2015] and shape description [Fang et al., 2015].

Significant work can also be done in terms of implementing novel and fast methods to compute several positive examples from a single patch, for the case of filtering-based approaches. Such methods could also be combined with the recently proposed epitomic methods [Papandreou, 2014], in order to generate intra-class variance matrices for sample patches by simply perturbing the measurment filters. This will lead to the benefit of avoiding the costly process of generating affine positives with random homography matrices and bilinear interpolations.

In the area of convolutional feature descriptors our work has focused on learning fast and discriminative floating point descriptors. Future work can explore methods to binarise the output of such convolutional descriptors, in order to allow for methods with lower storage requirements, and faster matching. In addition, recent works on representing the weights of convolutional networks in the binary or ternary bases [Rastegari et al., 2016, Sung et al., 2015], might lead to convolutional descriptors that are several orders of magnitude faster to compute than the current state of the art. Future work on learning convolutional feature descriptors from patch data can also focus on designing novel loss functions that use the available training data in ways that will lead to a more robust correlation between the performance on the training datasets and in real world-matching applications.

A very promising line of future work is the use of the introduced large scale dataset in novel ways for training convolutional descriptors. For example, we showed as a proof of concept that methods which focus on learning feature descriptors based on feature overlaps allow for a more discriminative learning of the convolutional embeddings. The only previously available data suitable for learning convolutional descriptors were based on positive and negative pairs, and thus do not allow for fine grained metric learning of similarity. In addition, the scale of the introduced dataset will allow novel designs of evaluation methods.

Though our proposed dataset is a significant step towards large scale and reproducible evaluation of any future work on local feature descriptors, collecting a

significant number of sequences and manually annotating them is a burden. Thus, a natural extension to our work, would be to investigate novel benchmarking methods that are based on photo-realistic synthetically created data. However, this will require a large set of high quality rendered scenes, together with methods to control the scene camera parameters, lightning and materials. In order to enable such methods to reach the challenging level of the real-world captured datasets, all these factors need to be incorporated to the rendering process, which may dramatically increase the computational overhead. However, we believe that the recent advances in state of the art GPUs will enable such photo-realistic synthetic experiments at large scale in the near future.

Bibliography

H. Aanæs, A.L. Dahl, and K. Steenstrup Pedersen. Interesting interest points. *International Journal of Computer Vision*, 97:18–35, 2012.

A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 510–517, June 2012. doi: 10.1109/CVPR.2012.6247715.

P. F. Alcantarilla, J. Nuevo, and A. Bartoli. Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *British Machine Vision Conf. (BMVC)*, 2013.

Hani Altwaijry, Eduard Trulls, James Hays, Pascal Fua, and Serge Belongie. Learning to match aerial images with deep attentive architectures. *Computer Vision and Pattern Recognition*, 2016.

V. Balntas, L. Tang, and K. Mikolazyk. A large-scale and reproducible benchmark for evaluating feature descriptors. In *arXiv*, 2016.

Vassileios Balntas, Lilian Tang, and Krystian Mikolajczyk. Bold - binary online learned descriptor for efficient image matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *In ECCV*, pages 404–417, 2006.

Ruud M Bolle, Sharath Pankanti, Jonathan H Connell, and Nalini K Ratha. Iris individuality: A partial iris model. In *Pattern Recognition, 2004. ICPR 2004.*

Proceedings of the 17th International Conference on, volume 2, pages 927–930. IEEE, 2004.

Léon Bottou. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade*, pages 421–436. Springer, 2012.

G. Bradski. The opencv library. 2000.

Jane Bromley, James W Bentz, Léon Bottou, Isabelle Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. Signature verification using a siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688, 1993.

Hongping Cai, Krystian Mikolajczyk, and Jiri Matas. Learning linear discriminant projections for dimensionality reduction of image descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):338–352, 2011. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2010.89.

Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 778–792, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15560-X, 978-3-642-15560-4. URL http://dl.acm.org/citation.cfm?id=1888089.1888148.

V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod. Chog: Compressed histogram of gradients a low bit-rate feature descriptor. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2504–2511, June 2009. doi: 10.1109/CVPR.2009.5206733.

V. Chandrasekhar, G. Takacs, D. M. Chen, S. S. Tsai, M. Makar, and B. Girod. Feature matching performance of compact descriptors for visual search. In *2014 Data Compression Conference*, pages 3–12, March 2014. doi: 10.1109/DCC.2014.50.

Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. cudnn: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759*, 2014.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In Cordelia Schmid, Stefano Soatto, and Carlo Tomasi, editors, *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005. URL http://lear.inrialpes.fr/pubs/2005/DT05.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

Jingming Dong and Stefano Soatto. Domain-size pooling in local descriptors: DSP-SIFT. *CoRR*, abs/1412.8556, 2014.

B. Fan, F. Wu, and Z. Hu. Rotationally invariant descriptors using intensity order pooling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 (10):2031–2045, Oct 2012. ISSN 0162-8828. doi: 10.1109/TPAMI.2011.277.

Bin Fan, Qingqun Kong, Xiaotong Yuan, Zhiheng Wang, and Chunhong Pan. Learning weighted hamming distance for binary descriptors. In *ICASSP*, pages 2395–2399. IEEE, 2013. URL http://dblp.uni-trier.de/db/conf/icassp/icassp2013.html#FanKYWP13.

Yi Fang, Jin Xie, Guoxian Dai, Meng Wang, Fan Zhu, Tiantian Xu, and Edward Wong. 3d deep shape descriptor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2319–2328, 2015.

Tom Fawcett. Roc graphs: Notes and practical considerations for researchers. 2004.

Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

Philipp Fischer, Alexey Dosovitskiy, and Thomas Brox. Descriptor matching with convolutional neural networks: a comparison to SIFT. *CoRR*, abs/1405.5769, 2014. URL http://arxiv.org/abs/1405.5769.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 1735–1742. IEEE, 2006.

Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009. ISSN 1541-1672. doi: http://doi.ieeecomputersociety.org/10.1109/MIS.2009.36.

Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Sam Hare, Amir Saffari, and Philip HS Torr. Struck: Structured output tracking with kernels. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 263–270. IEEE, 2011.

Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. *CoRR*, abs/1412.6622, 2014. URL http://arxiv.org/abs/1412.6622.

Karen P Hollingsworth, Kevin W Bowyer, and Patrick J Flynn. The best bits in an iris code. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(6):964–973, 2009.

Gang Hua, Matthew Brown, and Simon Winder. Discriminant embedding for local image descriptors. In *International Conference on Computer Vision*, October 2007. URL http://research.microsoft.com/apps/pubs/default.aspx?id=74479.

Intel. *Intel SSE4 Programming Reference*, 2010. URL http://softwarecommunity.intel.com/isn/Downloads/Intel%20SSE4%20Programming%20Reference.pdf.

Michael Jahrer, Michael Grabner, and Horst Bischof. Learned local descriptors for recognition and matching. In *Computer Vision Winter Workshop*, 2008.

Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, 2012. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2011.239.

Yan Ke and R. Sukthankar. Pca-sift: a more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–506–II–513 Vol.2, June 2004. doi: 10.1109/CVPR.2004.1315206.

Alex Krizhevsky and G Hinton. Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40, 2010.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

Griffin Lacey, Graham W. Taylor, and Shawki Areibi. Deep learning on fpgas: Past, present, and future. *CoRR*, abs/1602.04283, 2016. URL http://arxiv.org/abs/1602.04283.

Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521 (7553):436–444, 5 2015. ISSN 0028-0836. doi: 10.1038/nature14539.

Karel Lenc. *Evaluation and Improvements of Image Interest Regions Detectors and Descriptors*. PhD thesis, České vysoké učení technické v Praze, 2013.

Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2548–2555, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-1-4577-1101-5. doi: 10.1109/ICCV.2011.6126542. URL http://dx.doi.org/10.1109/ICCV.2011.6126542.

Gil Levi and Tal Hassner. LATCH: learned arrangements of three patch codes. In *Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016. URL http://www.openu.ac.il/home/hassner/projects/LATCH.

David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94. URL http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

K. Mikolajczyk and J. Matas. Improving descriptors for fast tree matching by optimal linear projection. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007. doi: 10.1109/ICCV.2007.4408871.

Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, 2005. doi: 10.1109/TPAMI.2005.188. URL http://dx.doi.org/10.1109/TPAMI.2005.188.

Ondrej Miksik and Krystian Mikolajczyk. Evaluation of local detectors and descriptors for fast feature matching. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2681–2684. IEEE, 2012.

Mustafa Özuysal, Michael Calonder, Vincent Lepetit, and Pascal Fua. Fast keypoint recognition using random ferns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(3):448–461, 2010.

George Papandreou. Deep epitomic convolutional neural networks. *CoRR*, abs/1406.2732, 2014. URL http://arxiv.org/abs/1406.2732.

Mattis Paulin, Matthijs Douze, Zaid Harchaoui, Julien Mairal, Florent Perronin, and Cordelia Schmid. Local convolutional features with unsupervised training for image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 91–99, 2015.

Pavel Pudil, Jana Novovičová, and Josef Kittler. Floating search methods in feature selection. *Pattern recognition letters*, 15(11):1119–1125, 1994.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. *arXiv preprint arXiv:1603.05279*, 2016.

E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, Nov 2011. doi: 10.1109/ICCV.2011.6126544.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015. doi: 10.1016/j.neunet.2014.09.003. Published online 2014; based on TR arXiv:1404.7828 [cs.NE].

Gregory Shakhnarovich. *Learning Task-specific Similarity*. PhD thesis, Cambridge, MA, USA, 2005. AAI0809132.

Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. *International Conference on Computer Vision*, 2015.

K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1573–1585, Aug 2014. ISSN 0162-8828. doi: 10.1109/TPAMI.2014.2301163.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Sudipta N Sinha, Jan-Michael Frahm, Marc Pollefeys, and Yakup Genc. Gpu-based video feature tracking and matching. 2011.

C. Strecha, A. Bronstein, M. Bronstein, and P. Fua. Ldahash: Improved matching with smaller descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):66–78, Jan 2012. ISSN 0162-8828. doi: 10.1109/TPAMI.2011.103.

Wonyong Sung, Sungho Shin, and Kyuyeon Hwang. Resiliency of deep neural networks under quantization. *arXiv preprint arXiv:1511.06488*, 2015.

M. Christoudias T. Trzcinski and V. Lepetit. Learning image descriptors with boosting. *submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2013.

E. Tola, V. Lepetit, and P. Fua. DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830, May 2010.

A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. doi: 10.1109/CVPR.2008.4587633.

T. Trzcinski and V. Lepetit. Efficient Discriminative Projections for Compact Binary Descriptors. In *European Conference on Computer Vision*, 2012.

Yen-Yu Lin Tsun-Yi Yang and Yung-Yu Chuang. Accumulated stability voting: A robust descriptor from descriptors of multiple scales. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2008.

Tinne Tuytelaars and Cordelia Schmid. Vector quantizing feature space with a regular lattice. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

Jialei Wang, Peilin Zhao, Steven CH Hoi, and Rong Jin. Online feature selection and its applications. *Knowledge and Data Engineering, IEEE Transactions on*, 26(3):698–710, 2014a.

Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 1386–1393, Washington, DC, USA, 2014b. IEEE Computer Society. ISBN 978-1-4799-5118-5. doi: 10.1109/CVPR.2014.180. URL http://dx.doi.org/10.1109/CVPR.2014.180.

Zhenhua Wang, Bin Fan, and Fuchao Wu. Local intensity order pattern for feature description. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 603–610, Washington, DC, USA, 2011a. IEEE Computer Society. ISBN 978-1-4577-1101-5. doi: 10.1109/ICCV.2011.6126294. URL http://dx.doi.org/10.1109/ICCV.2011.6126294.

Zhenhua Wang, Bin Fan, and Fuchao Wu. Local intensity order pattern for feature description. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 603–610. IEEE, 2011b.

Zhenhua Wang, Bin Fan, and Fuchao Wu. Affine subspace representation for feature description. *CoRR*, 2014c.

Simon Winder and Matthew Brown. Learning local image descriptors. In *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, June 2007. URL http://research.microsoft.com/apps/pubs/default.aspx?id=74480.

Simon Winder, Gang Hua, and Matthew Brown. Picking the best daisy. In *Computer Vision and Pattern Recognition*. IEEE Computer Society, June 2009. URL http://research.microsoft.com/apps/pubs/default.aspx?id=79807.

Paul Wohlhart and Vincent Lepetit. Learning descriptors for object recognition and 3d pose estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

Xin Yang and Kwang-Ting Cheng. Ldb: An ultra-fast feature for scalable augmented reality on mobile devices. In *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, pages 49–57, Nov 2012. doi: 10.1109/ISMAR.2012.6402537.

Guoshen Yu and Jean-Michel Morel. ASIFT: An Algorithm for Fully Affine Invariant Comparison. *Image Processing On Line*, 1, 2011. doi: 10.5201/ipol.2011.my-asift.

Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Guangcong Zhang, Mason J. Lilly, and Patricio A. Vela. Learning binary features online from motion dynamics for incremental loop-closure detection and place recognition. *CoRR*, abs/1601.03821, 2016.

Lei Zhang, Yongdong Zhang, Jinhu Tang, Ke Lu, and Qi Tian. Binary code ranking with weighted hamming distance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1586–1593, 2013.

Andrew Ziegler, Eric Christiansen, David Kriegman, and Serge J. Belongie. Locally uniform comparison image descriptor. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1–9. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4706-locally-uniform-comparison-image-descriptor.pdf.

C Lawrence Zitnick and Krishnan Ramnath. Edge foci interest points. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 359–366. IEEE, 2011.