This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915233, IEEE Transactions on Pattern Analysis and Machine Intelligence

HPatches: A benchmark and evaluation of handcrafted and learned local descriptors

Vassileios Balntas*, Karel Lenc*, Andrea Vedaldi, Tinne Tuytelaars, Jiri Matas and Krystian Mikolajczyk

Abstract—In this paper, a novel benchmark is introduced for evaluating local image descriptors. We demonstrate limitations of the commonly used datasets and evaluation protocols, that lead to ambiguities and contradictory results in the literature. Furthermore, these benchmarks are nearly saturated due to the recent improvements in local descriptors obtained by learning from large annotated datasets. To address these issues, we introduce a new large dataset suitable for training and testing modern descriptors, together with strictly defined evaluation protocols in several tasks such as matching, retrieval and verification. This allows for more realistic, thus more reliable comparisons in different application scenarios. We evaluate the performance of several state-of-the-art descriptors and analyse their properties. We show that a simple normalisation of traditional hand-crafted descriptors is able to boost their performance to the level of deep learning based descriptors once realistic benchmarks are considered. Additionally we specify a protocol for learning and evaluating using cross validation. We show that when training state-of-the-art descriptors on this dataset, the traditional verification task is almost entirely saturated.

Index Terms—local features, feature descriptors, image matching, patch classification

1 Introduction

Local feature descriptors remain an essential component of image matching and retrieval systems and it continues to be a very active area of research. With the success of learnable representations and the availability of increasingly large labelled datasets, research on local descriptors has seen a renaissance. End-to-end learning allows to fully optimise descriptors on available benchmarks, significantly outperforming fully [1] or semi-handcrafted features [2], [3].

Surprisingly, the adoption of these reportedly better descriptors has been limited in applications, with SIFT [1] still dominating the field. We believe that this is due to the inconsistencies in reported performance evaluations based on the existing benchmarks [4], [5]. The datasets are either small, or lack diversity to generalise well to various applications of descriptors. The progress in descriptor technology and application requirements has not been matched by a comparable development of benchmarks and evaluation protocols. As a result, while a novel descriptor may be highly optimised for a specific scenario, it is unclear whether it will work well in more general cases e.g. outside the specific dataset used to train it. In fact, solely comparing descriptors based on published experiments is difficult and inconclusive as demonstrated in Table 1.

In this paper, we introduce a novel benchmark suite for local feature descriptors, significantly larger, with clearly defined protocols and better generalisation properties, that has all the properties to supersede currently used datasets. This is inspired by the success of the Oxford matching dataset [4], the most widely-adopted and still very popular benchmark for the evaluation of local features, despite consisting of only 48 images. This is woefully insufficient for evaluating modern descriptors in the era of deep learning and large scale datasets. While some larger datasets exist, as discussed in Section 3, these have other important shortcomings in terms of data and task diversity, evaluation metrics and experimental reproducibility. We address these shortcomings by identifying and satisfying crucial requirements from such a benchmark in Section 4.

TABLE 1: Contradictory conclusions reported in the literature while evaluating the same descriptors on the same benchmark (Oxford affine covariant features [4]). Rows report inconsistent evaluation conclusions due to variations of the implicit parameters e.g. of feature detectors.

LIOP > SIFT	[6], [7]	•	SIFT > LIOP	[8]
BRISK > SIFT	[6], [9]	•	SIFT > BRISK	[10]
ORB > SIFT	[11]	•	SIFT > ORB	[6]
BinBoost > SIFT	[3], [10]	•	SIFT > BinBoost	[8], [12]
ORB > BRIEF	[11]	•	BRIEF > ORB	[10]

Data diversity is considered especially important for evaluating various properties of descriptors. To this end, we collect a large number of multi-image sequences of different scenes under real and varying capturing conditions, as discussed in Section 5. Scenes are selected to be representative of different use cases and captured under varying viewpoint, illumination, or temporal conditions, including challenging nuisance factors often encountered in applications. The images are annotated with ground-truth transformations, that allow to identify unique correspondences necessary to assess the quality of matches established by descriptors.

Reproducibility and fairness of comparisons is crucial in benchmarks. This is addressed by eliminating the influence of detector parameters. Hence, the benchmark is based on extracted local image patches rather than whole images, which brings important benefits: i) it allows comparing descriptors modulo, but independently of the choice of detectors, ii) it simplifies the

- *V. Balntas and K. Lenc contributed equally to this work.*
- *K. Lenc, V. Balntas and A. Vedaldi are with the Department of Engineering Science, University of Oxford, UK.*
- *K. Mikolajczyk is with the Department of Electrical and Computer Engineering, Imperial College London, UK*
- *T.Tuytelaars is with the Katholieke Universiteit Leuven, Belgium.*
- *J.Matas is with the Center for Machine Perception, Czech Technical University, Prague, Czech Republic.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915233, IEEE Transactions on Pattern Analysis and Machine Intelligence

process and makes the experiments reproducible, and, importantly, iii) it avoids various biases, e.g. the number or size of measurement regions or semi-local geometric constraints that make the results from image-based benchmarks incomparable (Section 3).

Task diversity is another requirement rarely addressed in existing evaluation benchmarks. To this end, we define three complementary benchmarking tasks in Section 6: patch verification (classification of patch pairs), image matching, and patch retrieval. These are representative of different use cases and, as we show in the experiments, descriptors rank differently depending on the task considered.

While this work focuses on local descriptors, the proposed dataset contains ground-truth, including pairwise geometric transformations, that is suitable for future evaluations of feature detectors as well. We believe that this benchmark will enable the community to gain new insights in state-of-the-art local feature matching since it is more diverse and significantly larger than any existing dataset used in this field, such as those implemented in the VLBenchmarks [13]. We assess various methods including simple baselines, handcrafted ones, and state-of-the-art learned descriptors in Section 7. The experimental results show that descriptor performance and their ranking may vary in different tasks, and differ from the results reported in the literature. This further highlights the importance of introducing a large, varied and reproducible evaluation benchmark for local descriptors.

This manuscript extends the original conference paper [14] by refining and clarifying descriptions of methods, evaluation protocols, and experimental results. Mainly, we have defined three splits of the dataset sequences into training and test set and we have defined a cross validation protocol for training and testing new descriptors over these splits. We use a similar methodology for finding the best descriptor normalisation. We provide results for either learning the data whitening on a separate dataset or using cross validation on the presented dataset. Additionally, we provide more detailed results across different variants of the benchmark.

All descriptors, benchmark data and code implementing the evaluation protocols are made publicly available[1].

2 Review of Local Feature Methods

In this section we review the state-of-the-art in feature detection and description. The focus of this paper is on evaluating keypoint descriptors but this cannot be done without extracting keypoints, we therefore briefly review some influential works on keypoint detectors. Figure 1 shows the timeline and some of the key contributions in the past two decades, showing increasing interest in CNN based and binary descriptors.

2.1 Keypoint detectors

Handcrafted detectors. There is a large body of research on keypoint detectors with a wide variety of approaches, as well as a number of surveys that discuss their properties in detail, in particular for the handcrafted methods [15]. Many of the widely used detectors are based on various convolutional filters. Detectors such as Harris [16] or Hessian [17] are based on first and second order derivatives and SIFT [1] uses Difference of Gaussians (DoG). SIFER [18] and D-SIFER [19] use Cosine Modulated Gaussian filters and tenth order Gaussian derivative filters. More recently, KAZE [20] introduces a non-linear gradient based diffusion as a preprocessing of images in contrast to isotropic filters in other detectors.

Many other hand-crafted detectors are tuned to find specific structures within the image but typically generalize to other patterns that exhibit large signal variations in local areas. Edge Foci [21] uses edges to improve robustness to illumination changes and WADE [22] exploits symmetries in local structures. Some detectors have resulted from a combination of the existing ones. Harris-Laplace [17] detector is a successful hybrid of Harris and Laplacian-of-Gaussian (LoG) method addressing the scale change problem. This was further extended to affine changes in Harris-Affine and Hessian-Affine detectors [17]. Other methods such as MSER [23] or FAST [24] detect features by comparing intensity levels of regions or individual pixels, respectively.

In [25] orientation of local gradients, instead of being quantised into bins as for SIFT, are mapped into explicit feature maps which approximate the distance metric for angles. This method is further improved in [26] by selecting better initial parametrisations of the local image gradients.

Learned Keypoint detectors. One of the first successful attempts was FAST [24] and FAST-ER [27], which mainly addressed the efficiency of feature extraction for real time applications. ORB [11] extended this idea to rotation changes. Similarly, BRISK [9] extends FAST and its derivative AGAST [28] by searching for maxima not only in the spatial coordinates, but also in a scale-space using the FAST score as a measure for saliency. [29], [30] were the first to exploit boosting and Haar filters trained on points detected with another handcrafted method but focus on highly repeatable points only. Optimizing filters for keypoint detection was also considered in [31]. Other machine learning algorithms include the use of Genetic Programming [32], or Structure-from-Motion [33] to predict keypoints with high matching score.

CNN based detectors. Advances in Convolutional Neural Networks have also made an impact on keypoint detection. Notably, [34] proposed a model to learn from a large dataset to identify and describe meaningful keypoints. A similar idea is exploited LIFT [35], TILDE [36] and SuperPoint [37].

Other detectors learn a CNN network using a geometry (covariant) constraint only [38] or a combination of geometry and a patch appearance loss [39]. Similarly, the covariant constraint is extended for affine adaptation in [40].

From the methods discussed above Harris, Hessian and DoG have been the most widely used for the past decade and are capable of providing a large number of regions, we therefore use these detectors to extract regions for descriptor evaluations.

2.2 Keypoint descriptors

Handcrafted descriptors. The design and implementation of local descriptors has undergone a remarkable evolution over the past two decades ranging from differential [41], [42] or moment invariants [43], correlations, PCA projected patches (PCA-SIFT) [44], histograms of gradients such as SIFT [1] HoG [45], GLOH [4], DAISY [5], DSP-SIFT [46] or other measurements such as LBP [47], BRIEF [48] etc. An overview of pre-2005 descriptors with SIFT [1] identified as the top performer can be found in [4]. The associated benchmark data accelerated the progress in this field and there have been a number of notable contributions. Efficient computation of features similar to SIFT [1] was targeted in SURF [49], which approximates convolutional kernels by a set of box-type filters and integral images. Despite many research

1. https://hpatches.github.io

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915233, IEEE Transactions on Pattern Analysis and Machine Intelligence

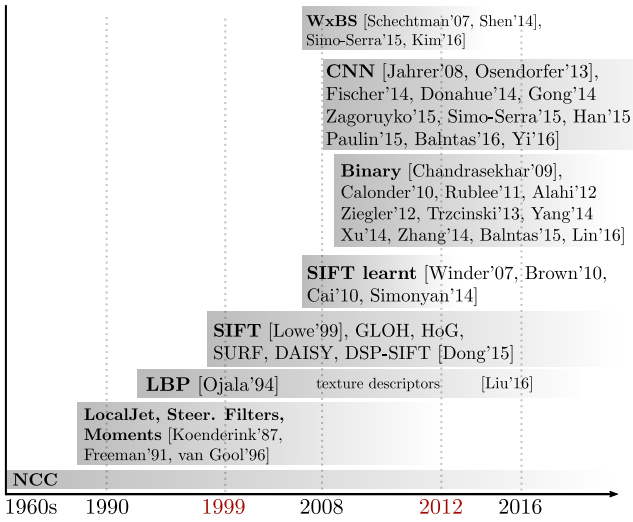

Fig. 1: Illustrative timeline of significant trends in development of local feature descriptors. For example SIFT based descriptors, introduced in 1999, have received continuous acclaim while for other descriptor families the interest has faded more quickly.

efforts in this area the improvements proposed by various methods were not convincing enough to supersede SIFT in general.

Invariance to various image transformations was also actively researched topic by engineering descriptors with built in invariance to rotation or illumination changes. Orientation estimation/normalization was addressed in [1] via main mode of gradient orientation histogram. Central gradient orientation was also proposed in [50], and direction from the patch centre to intensity mass centre was used in [11]. Rotation invariance was implemented differently in MRRID and MROGH descriptors [51] by pooling local features based on their intensity orders in multiple support regions. This concept is further exploited in LIOP [7] together with segmentation based location grid in contrast to a square grid in SIFT or SURF. LUCID [52] exploited linear time permutation distances between the ordering of RGB values of two image patches as descriptor. Siamese CNN was also exploited in this context in [53] to estimate two values (derivatives) for arctan2 of corresponding orientations.

Binary descriptors. From the family of local binary descriptors BRIEF [48] is one of the broadly adopted methods, based on randomised intensity comparison. This inspired a number of follow up works such as ORB [11], FREAK [54], BRISK [9], D-BRIEF [55], OSRI [56], USB [57], BOLD [12]. BRIEF was improved in ORB [11] by selecting uncorrelated tests that maximize the variance across training patches. BRISK [9] further optimizes BRIEF by using decision trees. FREAK [54] attempted to model a retina in human eye with cascade of binary strings from comparing image intensities over a retinal sampling pattern. Learning of discriminant and low dimensional spaces has also been applied to binary descriptors. D-BRIEF [55] is built by using the inter to intra class distance objective adapted to a binary descriptor. A set of discriminative projections is computed and approximated with a set of predefined dictionaries in order to generate a binary feature vector. The recently proposed descriptor BinBoost [3], [58] applies boosting to learn a set of binary hash functions that achieve the performance comparable to real-valued descriptors. Both D-BRIEF and BinBoost are not based on binary intensity tests therefore the extraction process is less efficient. A different research direction is to use coding methods to make the descriptors representation compact [59]. Histograms of quantised intensity for each pixel in the patch (HIPs) [60] were converted into binary codes for efficient extraction and matching using bitwise operations. Ordinal and spatial information of regional invariants by computing difference tests over a rotationally invariant sampling pattern was investigated in OSRI [56]. Ultrashort binary descriptors USB [57] of 24 bits allowed to use large number of efficiently extracted and matched features. An online learnt binary descriptor was proposed in [12], which used online transformed patches to adapt binary tests to each patch independently. Learning and deep neural networks were also recently used for extracting compact binary descriptors by optimizing quantization loss, evenly distributed codes and uncorrelated bits in [61].

Learned descriptors. Large datasets with correspondence ground truth enabled learning methods to be used to improve performance of existing descriptors [62]. One such approach consists of optimally learning descriptor parameters [5]. Another research direction is learning discriminative projections from high dimensional feature space to subspaces with better discriminating power. In [63], [64] the descriptor optimization is similar to the LDA based projections, which simultaneously minimizes intra-class and maximizes inter-class distances, where each patch is considered a class. A similar idea was exploited in [65] where LDA like projections were learnt and applied to gradient based features and optimized thresholds were then used to binarize the dimensions resulting in a binary descriptor. Convex optimization for descriptor learning was proposed in [66]. In contrast to learnt projections of existing descriptors an interesting observation was made in [67], which improved SIFT by applying simple square root normalization.

CNN based descriptors. Compared to shallow learning based descriptors, CNNs differ in terms of the applied learning techniques, volume of training data and computational efficiency therefore direct comparison shows significant differences in performance and speed. Preliminary works on using CNNs for extracting local descriptors have been done in [68], [69].

The interest in CNNs based descriptors started from results shown in [70] that the features from the last layer of a convolutional deep network trained on ImageNet can outperform SIFT even though the networks were not specifically optimized for such local representations. Deep convolutional activation features were investigated in [71] as generic image descriptors for a range of visual tasks. To improve the invariance of such features [72] extracts CNN activations for patches at multiple scales and performs aggregation similar to VLAD [73]. An unsupervised patch descriptor based on Convolutional Kernel Networks was attempted in [74]. End-to-end learning of patch descriptors using Siamese networks and the hinge contrastive loss has recently been re-attempted in several works which include siamese MatchNet [75] and DeepCompare [76] with a distance metric learning for convolutional features, DeepDesc [77] exploiting hard-negative mining, and TFeat descriptor [78] based on shallow convolutional networks, triplet learning constraints and fast hard negative mining. The WLRN descriptor [79] is also based on a shallow convolutional network, however the optimisation process is focused on utilising weakly-labelled data. In contrast, HardNet [80] implements SIFT's second nearest neighbour matching criterion in the loss function, that maximizes the distance between the closest positive and closest negative example in the batch. This can be viewed as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915233, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 2: Matching performance measured by mean average precision (mAP) for different magnification ρ of region size parameter . Parameter ρ represents the scaling of the size of the measurement region i.e. increasing the detected DoG keypoint region by a factor of ρ. Columns 1 / X show matching (mAP) between the first and the X-th image in the sequence.

ρ	1 / 2	1 / 3	1 / 4	1 / 5	1 / 6
1	31	13	5	3	1
4	68	44	24	15	11
12	80	67	54	42	35
20	87	77	69	55	50

a variant of popular triplet learning with hard negative mining. L2Net [81] applies progressive sampling from a large training data and a loss function that emphasises Euclidean distance as a similarity metric.

Given the multitude of techniques sometimes differing on the level of implementation details or loss functions, an objective evaluation protocol and data become more important then ever to guide users in their choice of methods or developers in their research directions.

3 Review of Existing Benchmarks

In this section we review existing datasets and benchmarks for the evaluation of local descriptors and discuss their main shortcomings.

3.1 Image-based benchmarks

In image matching benchmarks, descriptors are used to establish correspondences between images of the same objects or scenes. Local features, extracted from each image by a co-variant detector, are matched by comparing their descriptors, typically with a nearest-neighbour approach. Then, putative matches are assessed for compatibility with the known geometric transformation between images (usually a homography) and the relative number of correspondences is used as the evaluation measure.

All these datasets share an important shortcoming that leaves scope for variations in different descriptor evaluations: there is no pre-defined set of regions to match. As a consequence, results depend strongly on the choice of detector (method, implementation, and parameters), making the comparison of descriptors very difficult and unreliable. This is demonstrated in Table 1 where different papers reach different conclusions even when the same data and the same protocol are used for evaluation.

Defining centre locations of features to match does not constrain the evaluation sufficiently either. For example, this does not fix the region of the image used to compute the descriptor, typically referred to as the *measurement region*. Usually the measurement region is set to a fixed but arbitrarily set scaling of the feature size returned by a detector, and this parameter is often not reported or varies in papers. Unfortunately, this has a major impact on performance [66]. Table 2 shows matching scores for different scaling factors of the measurement region in the Oxford data.[2] Variations of more than 50% mAP occur; in fact, due to the planarity of such scenes, larger measurement regions lead to improved matching results.

2. mAP is computed on the Leuven sequence in the Oxford matching dataset using the DoG detector and SIFT descriptor.

In order to control for the size of the measurement region and other important parameters such as the amount of blurring, resolution of the normalized patch used to compute a descriptor [82], or use of semi-local geometric constraints, we argue that a descriptor benchmark should be based on *image patches* rather than whole images. Thus, all such ambiguities are removed and a descriptor can be represented and evaluated as a function $f(\mathbf{x}) \in \mathbb{R}^D$ that maps a patch $\mathbf{x} \in \mathbb{R}^{H \times H \times 3}$ to a D-dimensional feature vector. This type of benchmark is discussed next.

3.2 Patch-based benchmarks

Patch based benchmarks consist of patches extracted from interest point locations in images. The patches are then normalised to the same size, and annotated pair- or group-wise with labels that indicate positive or negative examples of correspondence. The annotation is typically established by using image ground-truth, such as geometric transformations between images. In case of image based evaluations the process of extracting, normalising and labelling patches leaves scope for variations and its parameters differ between evaluations.

The first popular patch-based dataset was *PhotoTourism* [5] (sometimes referred as Brown dataset). Since its introduction, the many benefits of using patches for benchmarking (Section 6.3) became apparent. PhotoTourism introduced a simple and unambiguous evaluation protocol, which we refer to as *patch verification*: given a pair of patches, the task is to predict whether they match or not, which reduces the matching task to a binary classification problem. This formulation is particularity suited for learning-based methods, including CNNs and metric learning in particular due to the large number patches available in this dataset. The main limitation of PhotoTourism is its scarce data diversity (there are only three scenes: Liberty, Notre-Dame and Yosemite), task diversity (there is only the patch verification task), and feature type diversity (only DoG features were extracted). The *CVDS dataset* [83] addresses the data diversity issue by extracting patches from five MPEG-CDVS: Graphics, Paintings, Video, Buildings and Common Objects. Despite its notable variety, experiments have shown that the state-of-the-art descriptors achieve high performance scores on this data [84]. The *RomePatches dataset* [85] consider a query ranking task that reflects image retrieval scenario, but is limited to $10K$ patches, which makes it an order of magnitude smaller than PhotoTourism.

3.3 Metrics

In addition to choosing data, patches, and tasks, the choice of evaluation metric is also important. For classification, the Receiver Operating Characteristic (ROC) curves have often been used [86], [87] as the basis for comparison. However, patch matching is intrinsically highly unbalanced, with many more negative than positive correspondence candidates; ROC curves are less representative for unbalanced data and, as a result, a strong performance in ROC space does not necessarily generalise to a strong performance in applications, such as the nearest-neighbor matching [8], [12], [55], [77]. Several papers [3], [5], [55] reported at a single point on the ROC curve (FPR95, i.e. the false positive rate at 95% true positive recall) which is more appropriate for unbalanced data than the equal error rate or the area under the ROC curve; however, this reduces the information provided by the whole curve. Precision-Recall and mean Average Precision (mAP) are much better choices of metrics for unbalanced datasets – for example DBRIEF [55] is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915233, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 3: Comparison of existing and ℍPatches dataset.

dataset	patch	diverse	real	large	multitask
Photo Tourism [62]	✓		✓	✓	
DTU [88]			✓	✓	
Oxford-Affine [4]		✓	✓		
Synth. Matching [89]		✓	✓		
CVDS [83]	✓	✓		✓	
Edge Foci [21]		✓	✓		
RomePatches [85]	✓		✓		
RDED [90]		✓	✓		
ℍPatches	✓	✓	✓	✓	✓

excellent in ROC space but has very low (≈ 0.01) mAP for the Oxford dataset [10].

4 Benchmark Design

To address the shortcomings of the existing datasets, discussed in Section 3, we identify the following requirements for a good benchmark:

- *Reproducible, patch-based*: descriptor evaluation for a good benchmark should be done on patches to eliminate the detector related-factors. This leads to a standardisation across different works and makes results directly comparable.
- *Diverse*: representative of many different scenes and image capturing conditions.
- *Real*: real data have been found to be more challenging than a synthesized one due to nuisance factors that cannot be modelled in image transformations.
- *Large*: to allow accurate and stable evaluation, as well as to provide substantial training sets for learning based descriptors.
- *Multitask*: representative of several use cases, from matching image pairs to image retrieval. This allows cross-task comparison of descriptors performance within the same data.

Based on these desired properties, we introduce a new large-scale dataset of image sequences (Section 5) annotated with ground-truth homographies. This is used to generate a patch-based benchmark suite for evaluating local image descriptors (Section 6). Table 3 compares the proposed dataset to existing benchmarks in terms of the properties stated above.

5 Images and Patches

Images are collected from various sources, including existing datasets. We have captured 51 new sequences, 33 sequences were manually generated by finding suitable scenes from [91], 12 scenes are from [88], 5 scenes from [90], 4 scenes from [4], 2 scenes from [92] and 1 scene from [93]. Some of the sequences are illustrated in Figure 2. In 57 of the scenes, the main nuisance factors are photometric changes and the remaining 59 sequences show significant geometric deformations due to viewpoint change.

A sequence includes a reference image and 5 target images with varying photometric of geometric changes. The sequences are captured such that the geometric transformations between images can be well approximated by homographies from the reference image to each of the target images. The homographies are estimated following [4].

Patches are extracted using the following protocol. Several scale invariant interest point detectors i.e. DoG, Hessian-Hessian and Harris-Laplace are used to extract features[3] for scales larger than $1.6px$, which give stable points. Near-duplicate regions are discarded based on their intersection-over-union (IoU) overlap (> 0.5) and one region per cluster is randomly retained. This keeps regions that overlap less than 0.5 IoU. Approximately 1,300 regions per image are then randomly selected.

3. VLFeat implementations [82] of detectors are used.

TABLE 4: The range of geometric noise distributions, in units of a patch scale. Rotation θ translation t scale s and anisotropic distortion a in Equation (1) are sampled from a uniform distributions limited by θ_{max}, t_{max}, s_{max} and a_{max}.

Noise	θ_{max}	t_{max}	s_{max}	a_{max}
EASY	10°	0.15	0.15	0.2
HARD	20°	0.3	0.3	0.4
TOUGH	30°	0.45	0.5	0.45

For each sequence, patches are detected in the reference image and projected on the target images using the ground-truth homographies. This sidesteps the limitations of the detectors, which may fail to provide corresponding regions in every target image due to significant viewpoint or illumination variations. Furthermore, it allows to extract more patches thus better evaluate descriptors in such scenarios. Regions that are not fully contained in all target images are discarded. Hence, a set of corresponding patches contains one from each image in the sequence.

In practice, when a detector extracts corresponding regions in different images, it does so with a certain amount of noise. In order to simulate this noise, detections are perturbed using three settings: EASY, HARD and TOUGH. This is obtained by applying a random transformation $T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ to the reference region before projection. Thus, a sequence of matches includes the detected region and 5 randomly transformed ones that are projected to target images with homography. This process is visualised in Figure 3. Assuming that the region centre is the coordinate origin, random transformation

$$T = R(\theta) \cdot \begin{bmatrix} s/\sqrt{a} & 0 & m\,t_x \\ 0 & s \cdot \sqrt{a} & m\,t_y \end{bmatrix}, \qquad (1)$$

includes rotation $R(\theta)$ by angle θ, anisotropic scaling by $s/\sqrt{a}$ and $s\sqrt{a}$, and translation by $[m\ t_x, m\ t_y]$ where m is the detected region scale, and $R(\theta)$ is a rotation of angle θ. Thus the translation is proportional to the detection scale m. The transformation parameters are uniformly sampled from the intervals $\theta \in [-\theta_{max}, \theta_{max}]$, $t_x, t_y \in [-t_{max}, t_{max}]$, $\log_2(s) \in [-s_{max}, s_{max}]$, $\log_2(a) \in [-a_{max}, a_{max}]$, whose values for each setting are given in Table 4.

These settings reflect the typical overlap accuracy of the Hessian and Hessian-Affine detectors on the Oxford matching benchmark. There, images in each sequence are sorted by increasing transformation, resulting in increased detector noise. To identify the levels of noise in the sequence we measure the average overlap IoU for corresponding regions between the reference image and the target images, which is presented in Figure 4. The noise level indeed increases in the sequence therefore average IoU decreases and the baseline noise level from the first target image differs for different sequences. We choose the noise parameters such that EASY group approximately corresponds to regions extracted in image pairs 1-2, 1-3 HARD to regions from 1-4, 1-5 and TOUGH to pairs 1-5,1-6. The noise parameters values are sampled from uniform distributions limited by factors listed in Table 4.

Detected regions are scaled with a factor $\rho = 5$ (see Section 3). The smallest patch size in the reference image is 16×16px since

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915233, IEEE Transactions on Pattern Analysis and Machine Intelligence

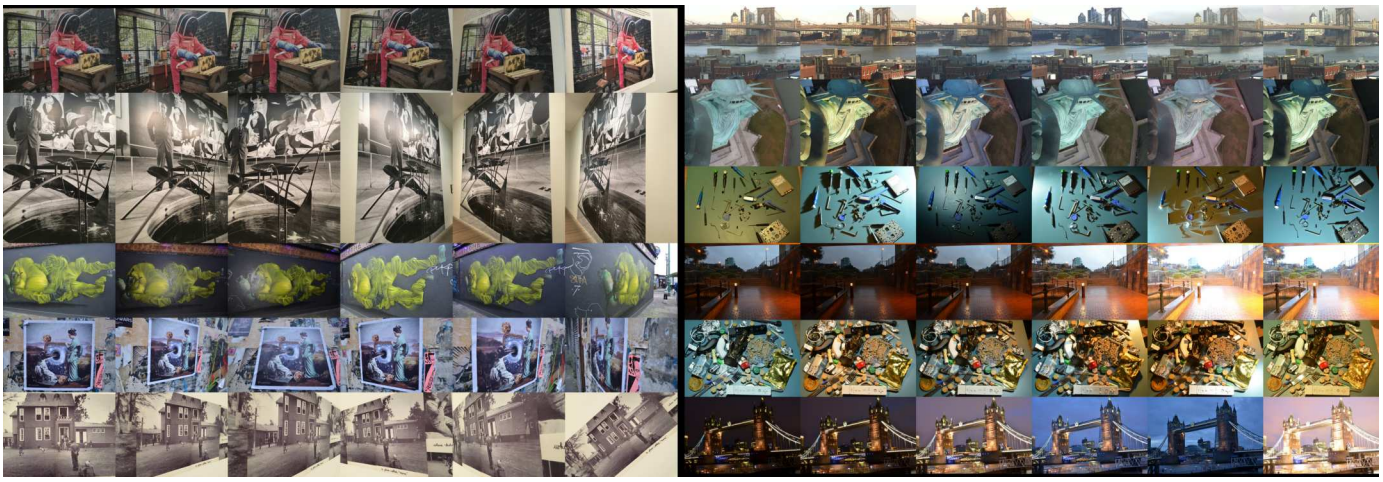

Fig. 2: Examples of the image sequences contributing to ℍPatches; note the diversity of scenes and nuisance factors, including viewpoint (left), illumination (right), focus, reflections and other changes.

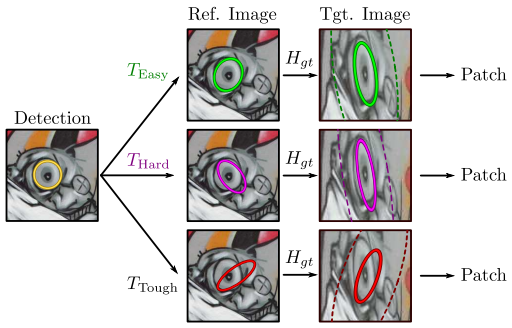

Fig. 3: Construction of regions for patch extraction. Each detected feature frame (yellow) is reprojected in the ref. image with a random transformation T for EASY, HARD or TOUGH. Using the ground truth homography H_{gt}, these regions are reprojected to target images where the patch is extracted from a measurement region visualised by dashed ellipse for factor $\rho = 3$.

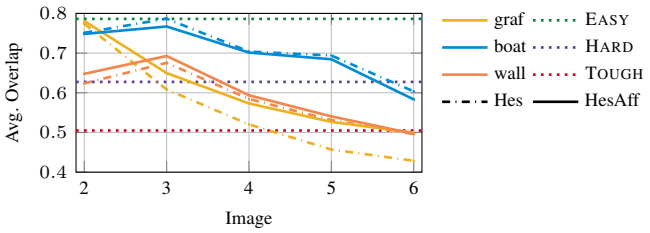

Fig. 4: Average overlap of the Hessian and Hessian-Affine detectors on the viewpoint sequences of [94]. Line color encodes dataset and line style a detector. The selected overlaps of the EASY and HARD variants are visualised with a dotted line.

only regions from detection scales above 1.6px are considered. In each region the dominant orientation angle is estimated using a histogram of gradient orientations [1]. Regions are rectified by normalizing the transformed region to a circle using bilinear interpolation and extracting a square of 65×65 pixels. Example of extracted patches are shown in Figure 5, where the effect of the increasing detector noise is clearly visible.

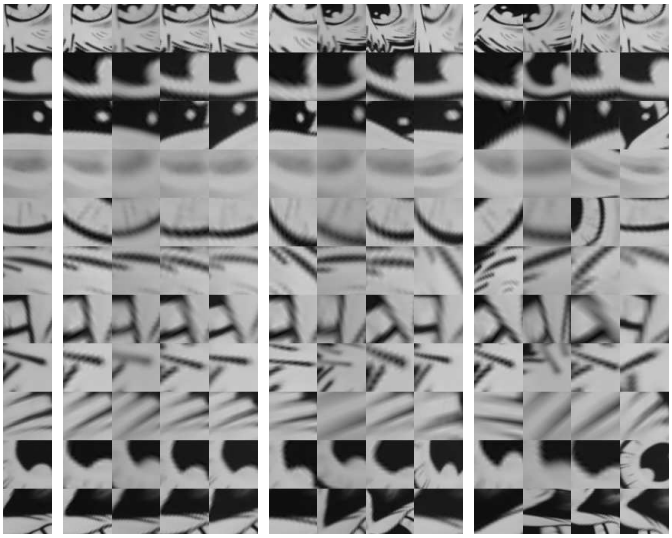

Fig. 5: Geometric noise visualized on a sample set of patches. Left column shows the reference patches and the other columns show the corresponding patches from EASY, HARD and TOUGH distributions.

6 BENCHMARK TASKS

In this section, we define the benchmark metrics, tasks and their evaluation protocols for patch verification, image matching and patch retrieval.

The tasks are designed to imitate typical use cases of local descriptors. Patch verification (Section 6.2) is based on [5] and measures the ability of a descriptor to classify whether two patches are extracted from the same interest point. Image matching (Section 6.3), inspired by [4], tests to what extent a descriptor can correctly identify correspondences between two images, which can be later used for accurate homography estimation. Finally, patch retrieval (Section 6.4) tests how well a descriptor can match a query patch to a pool of patches extracted from a large set of distractor patches. This is a proxy to local feature based image indexing [85], [95].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915233, IEEE Transactions on Pattern Analysis and Machine Intelligence

6.1 Evaluation metric

We first define the precision and recall evaluation metric used in HPatches. Let $\mathbf{y} = (y_1, \dots, y_N) \in \{-1, 0, +1\}^N$ be labels for a ranked list of patches returned for a query, indicating negative, *to ignore*, and positive match, respectively. Then *precision* and *recall* at rank i are given by $P_i(\mathbf{y}) = \sum_{k=1}^{i} [y_k]_+ / \sum_{k=1}^{i} |y_k|$ and $R_i(\mathbf{y}) = \sum_{k=1}^{i} [y_k]_+ / \sum_{k=1}^{N} [y_k]_+$; the *average precision* (AP) is given by $AP(\mathbf{y}) = \sum_{k:y_k=+1} P_k(\mathbf{y}) / \sum_{k=1}^{N} [y_k]_+$, with $[z]_+ = \max\{0, z\}$. The main difference w.r.t. the standard definition of PR is that some entries can be ignored $y_k = 0$. This is used in the retrieval evaluation protocol to ignore the query patch from the retrieved results. All descriptors including the queries are in a KD-Tree data structure for fast search of nearest neighbours.

6.2 Patch verification

In *patch verification*, descriptors are used to classify whether two patches are in correspondence or not. The benchmark starts from a list $\mathcal{P} = ((\mathbf{x}_i, \mathbf{x}'_i, y_i),\ i = 1, \dots, N)$ of positive and negative patch pairs, where $\mathbf{x}_i, \mathbf{x}'_i \in \mathbb{R}^{65\times65}$ are patches and $y_i = \{-1, 1\}$ is the corresponding label, with -1 and 1 denoting a negative and positive pair respectively.

The dataset is used to evaluate a matching approach $\mathcal{A}$ that, given any two patches $\mathbf{x}_i, \mathbf{x}'_i$, produces a confidence score $s_i \in \mathbb{R}$ that the two patches form a match. The quality of the approach is measured as the average precision of the ranked patches, namely $AP(y_{\pi_1}, \dots, y_{\pi_N})$ where π is the permutation that sorts the scores in decreasing order (i.e. $s_{\pi_1} \geq s_{\pi_2} \geq \dots \geq s_{\pi_n}$).

Denoting the total number of positive pairs $N_{pos} = \sum_1^N y_i$, it follows that $N_{neg} = N - N_{pos}$. The ratio $\frac{N_{pos}}{N_{neg}}$ indicates the balance of $\mathcal{P}$. In [5], where the verification protocol was first introduced in the context of evaluating local feature descriptors, the evaluation was done in terms of ROC curves [87] and a list $\mathcal{P}$ such that $N_{pos} = N_{neg}$. However, most applications of local feature descriptors involve highly imbalanced scenarios (i.e. $N_{neg} \gg N_{pos}$) [77]. To test the effect of the positive to negative samples ratio, we generate two variants of the *verification* task, namely BALANCED and IMBALANCED, by altering the ratio of $\frac{N_{pos}}{N_{neg}}$ from 1 to 0.2. For the BALANCED variant, the performance of a descriptor is measured by the area under the ROC curve. As shown in [86] for an imbalanced scenario the ROC curve can be misleading, thus for the IMBALANCED variant, the evaluation is done in terms of the area under the precision-recall (PR) curve, which is the average precision.

Another important factor in the verification task is the *negative pair sampling method*, due to problems such as repetitive structures that might be present in a scene. We generate two variants of the task according to this, namely INTRA-SEQUENCE and INTER-SEQUENCE negative patch sampling. INTRA-SEQUENCE indicates that negative pairs of patches are sampled randomly from images within the same sequence, while for the INTER-SEQUENCE variant, the negative pairs are sampled randomly from different sequences.

Thus, there are several possible scenarios in the patch verification experiment, arising from the following options: BALANCED and IMBALANCED, INTRA-SEQUENCE and INTER-SEQUENCE, as well as EASY, HARD and TOUGH.

It is worth noting that the verification task only requires scores s_i, which in the context of local feature descriptors are typically defined as a negative distance measure. However, in some cases such a similarity can be directly learned from data [75], [76].

6.3 Image matching

In *image matching*, an image is represented by a collection of N patches $L_k = (\mathbf{x}_{ik},\ i = 1, \dots, N)$ and descriptors are used to match patches from a reference image to a target one. Consider a pair of images $\mathcal{L} = (L_0, L_1)$, where L_0 is the reference and L_1 the target. Based on homography, patches are sorted in each image such that $\mathbf{x}_{i0}$ is in correspondence with $\mathbf{x}_{i1}$.

The pair $\mathcal{L}$ is used to evaluate an algorithm $\mathcal{A}$ that, given a reference patch $\mathbf{x}_{i0} \in L_0$, determines the index $\sigma_i \in \{1, \dots, N\}$ of the best matching patch $\mathbf{x}_{\sigma_i 1} \in L_1$, as well as the corresponding confidence score $s_i \in \mathbb{R}$. Then, the benchmark labels the assignment σ_i as $y_i = 2[\sigma_i = i] - 1$, and computes $AP(y_{\pi_1}, \dots, y_{\pi_N}; N)$, where π is the permutation that sorts the scores in decreasing order (note that the number of positive results is fixed to N; see Section 6.1).

The overall performance of an algorithm $\mathcal{A}$ is computed as the mean average precision for all such image pairs across all the sequences available in HPatches. In addition, the performance can also be measured by the *matching success rate*, which is defined as the number of correct matches over all possible N matches in the collection L_k. Note that the benchmark only requires the indexes σ_i with scores s_i computed by the algorithm $\mathcal{A}$ for each image pair $\mathcal{L}$. Typically, these can be computed by extracting patch descriptors and matching them using nearest-neighbour.

This evaluation protocol is designed to closely resemble the one from [4]. A notable difference is that, since the patch datasets are constructed in such a way that each reference patch has a corresponding patch in each target image, the maximum recall is always 100%. Note also that similarly to the verification task, the benchmark evaluates the combined performance of the descriptor and similarity metric provided by the tested algorithm.

6.4 Patch retrieval

In *patch retrieval* descriptors are used to find patch correspondences in a large collection of patches, the great majority of which are distractors, i.e. extracted from irrelevant images. Consider a collection $\mathcal{Q} = (\mathbf{x}_i, i = 0, \dots, K)$ consisting of a query patch $\mathbf{x}_0$, extracted from a reference image L_0 in a sequence, and all patches corresponding to $\mathbf{x}_0$ from the intra-sequence images L_j, with $j \in [1, K]$. Note that for HPatches, $K = 5$.

Furthermore, consider a collection of N distractors $\mathcal{D} = (\bar{\mathbf{x}}_i, i = 1, \dots, N)$ consisting of randomly sampled patches across a large set of sequences. Note that $\mathcal{D}$ is built such that it does not include patches extracted from sequence $\mathcal{Q}$. The idea is that such patches are not detrimental for the purpose of retrieving the correct image, and such innocuous errors may occur frequently in the case of repeated structures in images.

We compute N similarity scores between the query $\mathbf{x}_0$ and all the items in the distractor pool $\mathcal{D}$. In addition, we also compute the K scores between the query and the remaining correspondence pool $\mathcal{Q}$. We then use the ground truth labels of the $\mathcal{D}$ and $\mathcal{Q}$ collections, which are 0 and 1 respectively, and their similarity scores to generate a precision-recall curve and compute the average precision. Note that for a perfect descriptor all the similarity scores between the query and the patches from $\mathcal{Q}$, would be higher than all the similarity scores between the query and the $\mathcal{D}$ collection.

In terms of the benchmarking process, we randomly sample a fixed set of all the query collections $\mathcal{Q}$, and we generate multiple variants of the retrieval task by altering the cardinality of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915233, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 5: Basic properties of the selected descriptors. For binary descriptors, the dimensionality is in bits (*), otherwise in number of single precision floats. The speed is measured in units of 1000 patches per second i.e. the higher the faster.

Desc.	Dim.	Input size	Speed [kP/s] CPU	GPU
MSTD	2	65	67.0	-
RESZ	36	65	3.0	-
SIFT	128	65	2.3	-
RSIFT	128	65	2.2	-
KDE	147	65	0.3	-
MKD	238	65	0.1	-
BRIEF	*256	32	333.0	-
BBOOST	*256	32	2.0	-
ORB	*256	32	333.0	-
DC-S	256	64	0.3	10.0
DC-S2S	512	64	0.2	5.0
DDESC	128	64	0.1	2.3
TFEAT-M	512	32	0.6	83.0
TNET	256	64	0.4	83.0
L2NET	256	64	0.1	63.3
HNET	128	32	0.7	3.1

distractor set $\mathcal{D}$. This is done to test the performance of the descriptors across increasing distractor pool sizes. The results for each variant are reported in terms of mean average precision across all the query collections $\mathcal{Q}$. The average number of features per reference image multiplied by the number of sequences gives the total number of query collections.

The design of this benchmark is inspired by classical image retrieval systems such as [85], [95], [96], which use patches and their descriptors as entries in image indexing. A similar evaluation may be performed by using the PhotoTourism dataset, which includes $\sim 1000K$ sets of positive patch pairs. Unfortunately, since these small sets are not maximal, there is no certainty that a set *does not* correctly correspond to another set, which makes the evaluation noisy, i.e. two sets may correspond to the same physical point in space.

7 EXPERIMENTAL RESULTS

In this section we evaluate several local descriptors using the newly introduced $\mathbb{H}$Patches benchmark. We start by presenting the details of the evaluation (Sections 7.1 and 7.2) followed by a discussion of a selection of notable descriptors and of corresponding empirical results.

In more detail, in Sections 7.3 to 7.7 we illustrate several variants of the benchmark tasks considering, for succinctness and clarity, only a *subset* of the descriptors. In Section 7.6 we investigate the stability of the results across different splits of the dataset, in Section 7.7 we show the effect of *training* CNN descriptors on different data splits, and in in Section 7.8 we study the effect of choosing different methods for descriptor normalisation. Finally, we show the results for all tasks and *all* selected descriptors in a compact form in Section 7.9.

7.1 $\mathbb{H}$Patches dataset details

Contents. As discussed in Section 5, our benchmark data comprises 116 image sequences, 57 of which contain illumination/photometric VIEWP transformations and 59 viewpoint/geometric ILLUM transformations. Each sequence consists of one reference image and 5 target images with increasing transformation magnitude. The reference image is related to each of its target images by a ground truth homography H_{gt}. There are approx 1.3k regions of interest for each image extracted by combining the DoG, Hessian-Hessian and Harris-Laplace detectors, resulting in 157k reference features and 785k target features for each of the geometric noise variants EASY, HARD, and TOUGH. Regions are extracted and normalised to patches of size 65×65 pixels.

Splits. In order to assess the variance in the experimental results, as well as the overfitting characteristics of different learnable descriptors, $\mathbb{H}$Patches is divided into multiple training and testing splits. Three generic splits, called SET A, SET B and SET C, are generated by randomly splitting the data in 80 sequences for training and and 36 for testing, repeating the selection three times with different seeds. These splits are used for cross-validation as the dataset does not define a hidden test set. In cases where a descriptor is trained or a normalisation is selected based on the $\mathbb{H}$Patches data, the results are reported as an average over the splits' test sets.

In order to test the effect of different transformations, furthermore, two more splits that only contain sequences from the same transformation type, namely VIEWP and ILLUM, are generated as well. Note that the randomly sampled SET A, SET B and SET C splits all contain both VIEWP and ILLUM sequences.

Verification details. For the verification task, 1M negative and 1M positive patch pairs are generated for each split. For the BALANCED experiment, all positive patch pairs are kept, whereas for the IMBALANCED only $0.25M$ positive pairs are kept in order to achieve a 1 to 4 imbalance between positive and negative matches.

Matching details. After all the descriptors are extracted for both the reference image patches and the target image patches, one-to-one matching is performed, and the correct matches are used to compute average precisions. Results are then averaged across all image pairs. Since the goal is to assess and compare descriptors on an equal footing, matches are performed directly by comparing descriptors without further filtering steps such as Lowe's 1st to 2nd NN distance criterion. Such filtering steps can still be employed in applications to improve matching accuracy further.

Retrieval details. For each split, we generate 10k query patches, each having 5 ground-truth matching target patches. For each query patch approximately 20k distractor patches are randomly sampled from all sequences. In experiments, in order to examine the effect of the size of the distractor pool, the latter is gradually reduced to 20k, 15k, 10k, 5k, 2k, 1k, 100 patches.

Metric. As discussed in Section 6.1, *Mean Average Precision* (*m*AP), i.e. the area under the precision-recall curve, is used to measure the performance of descriptors for all benchmark tasks and scenarios. The exception is the BALANCED verification scenario — since the latter is balanced, *m*AP is not appropriate and the *Area Under the Curve* (AUC) for the *Receiver Operating Characteristic* (ROC) graph is used instead

7.2 Descriptors

We evaluate the following descriptors, summarized in Table 5. We include two trivial **baselines**: MSTD, $[\mu, \sigma]$ which is the average μ and standard deviation σ of a patch, and RESZ, the vector obtained by resizing the patch to 6×6 pixels and normalizing it by subtracting μ and dividing by σ. Additionally, in some

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915233, IEEE Transactions on Pattern Analysis and Machine Intelligence

Desc.	EASY	HARD	TOUGH	AVG
TFEAT-M	91.55	84.61	74.77	80.36
DDESC	90.43	82.98	73.28	78.99
LIOP	80.24	73.35	65.55	70.45
DC-S	84.50	72.41	60.94	68.80
BBOOST	81.37	66.52	53.38	62.71
SIFT	84.95	65.68	51.25	62.48
RSIFT	77.83	58.21	45.50	56.28
ORB	78.02	57.02	42.38	54.26
BRIEF	72.72	56.32	44.43	53.86
NCC	66.52	49.28	39.02	48.19

TABLE 6: Verification task — mean average precision (*m*AP) for the IMBALANCED scenario and different levels of geometric noise. Increasing geometric noise leads to lower scores but the ranking of descriptors is not affected.

experiments we use Normalised Cross Correlation (NCC) [97] over the grid of 24×24 different shifts which is closely related to Pearson correlation coefficient. This method is often used as a basic template matching algorithm. For **SIFT-based** descriptors we include SIFT [1] and its variant RootSIFT (RSIFT) [67]. We also include KDE [25] and MKD [26] as examples of more recent hand-crafted descriptors.

From the family of **binary descriptors**, we test BRIEF [48], based on randomised intensity comparison, ORB [11], which optimises the binary tests by learning, and BBOOST [3], where binary tests are selected using boosting. We also evaluate several recent **deep descriptors** including the Siamese variants of DeepCompare [76] (DC-S, DC-S2S) with a distance metric learning for convolutional features, DeepDesc [77] (DDESC), which exploits hard-negative mining, and the TFeat (TFEAT-M) descriptor [78], based on shallow convolutional networks, triplet learning constraints and fast hard negative mining. L2Net [81] applies progressive sampling of large training data and HardNet [80] implements a loss inspired by the second nearest neighbour ratio. *Unless otherwise stated*, the learning-based descriptors were trained on PhotoTourism data.

Table 5 shows the dimensionality, size of the measurement region in pixels, and speed of each descriptor. DeepCompare [76] variants have the highest dimensionality of 256 and 512, whereas the other real-valued descriptors have 128 dimension with the exception of MSTD and RESZ. The size of all binary descriptors is 256 bits. In terms of speed, the binary descriptors BRIEF and ORB are approximately 30 times faster than the most efficient CNN based descriptor, namely TFEAT-M. Other descriptors are an order of magnitude slower, BBOOST, MKD and KDE being the slowest.

7.3 Patch verification

In this section, we discuss results on the patch verification task. For the IMBALANCED variant, results in Table 6 indicate that there is no significant difference in the ranking of descriptors evaluated on different amounts of geometric noise (EASY, HARD, and TOUGH). Thus, for the rest of the verification experiments only results computed on the HARD variant are considered.

Positive/negative imbalance. Sorting descriptors by the area under the ROCs in the BALANCED scenario and the PR curves in the IMBALANCED scenario results in approximately the same ranking. Hence, while ROC results are overoptimistic for applications as imbalanced scenarios are much more common in practice, they are still useful to compare descriptors.

The operating point has however an effect on descriptor ranking, as their relative performance differs at different operation points. For example, in the IMBALANCED scenario, NCC outperforms SIFT-like descriptors at the FPR95, even though their performance is significantly below SIFT on average. Overall, averaged metrics such as *m*AP and ROC AUC are more likely to offer a balanced evaluation of descriptors.

Methods that are based on convolutional neural networks tend to perform significantly better in the verification task than other descriptors. A likely reason is that the formulation of the verification task is essentially the same as the formulation used to train most of these descriptors, which are learned from pairs or triplets of negative and positive patches.

Source of distractors. In Figure 6 (right), we plot the results for three selected descriptors, TFEAT-M, DDESC and LIOP, using either the INTRA-SEQUENCE or INTER-SEQUENCE sampling schemes. It is clear that INTRA-SEQUENCE sampling consistently makes verification more challenging. The reason is that sampling from the same sequence returns patches that exhibit similar statistical properties, and are thus more difficult to discriminate. On the contrary, randomly sampling negative pairs across different sequences leads to slightly easier discrimination. This confirms the intuition that the presence of repetitive patterns, or other self-similarities structures in the image, is likely to significantly affect the performance of all descriptors.

To summarise, CNN-based features are very good at the verification task, with AP scores in the 50% to 80% range, whereas handcrafted features' AP is approximately 20% lower. For all descriptors, performance is strongly negatively affected by increasing geometric noise, by imbalanced sampling (IMBALANCED scenario), and by sampling distractors from the same sequence (INTRA-SEQUENCE scenario). However, the relative rankings of descriptors is generally stable across all such scenarios.

7.4 Image matching

In this section, we discuss the image matching results. Table 7 reports the *m*AP computed over all test sequences and Table 8 additionally reports the *matching success rate*, namely the percentage of query patches for which the correct result ranked first.

In general, we can observe that *m*AP scores are much lower than for the verification task. This can be explained by the fact that distractor patches are sampled from images in the same sequence as the matching patches, which, as explained above, results in harder negatives. In addition, the set of negative examples is much larger than the set of positive ones, which makes the data highly imbalanced, resulting in a much higher likelihood of retrieving hard negatives.

This observation indicates that patch verification performance, which is often used in the literature to compare descriptors, may be far better than the performance of the descriptors in image matching applications. Furthermore, descriptors that perform well in verification may drop in the ranking significantly for matching. For example, LIOP and BBOOST outperform SIFT in Table 6 but are far worse in Table 7. We also observe that the *matching success rate*, which has been reported in several papers [3], often results in a different ranking than *m*AP, as well as in different relative performance gaps.

7.5 Patch retrieval

Figure 7 shows the *m*AP results for the retrieval task. Note that the size of the distractor pool $\mathcal{D}$ (see Section 6.4) has a significant

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915233, IEEE Transactions on Pattern Analysis and Machine Intelligence

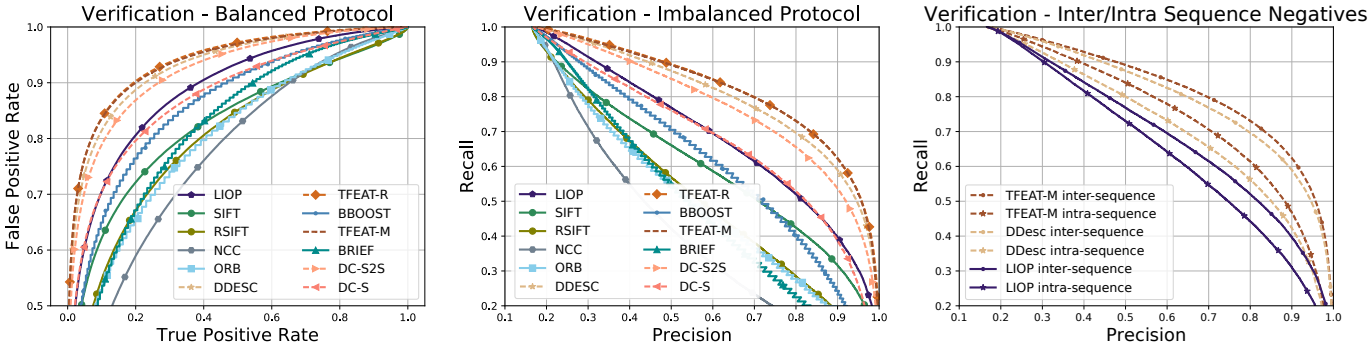

Fig. 6: Patch verification results for the state of the art feature descriptors, on the HARD geometric noise. (left) ROC curves for BALANCED scenario - ROC curve (middle) Performance on the IMBALANCED scenario - PR curve. Note that the ranking of the best performing descriptors remains similar between the BALANCED and IMBALANCED tasks. (right) Performance of INTRA-SEQUENCE and INTER-SEQUENCE negative pair sampling. INTRA-SEQUENCE sampling leads to a more challenging verification task, due to repetitive patterns and self-similarities that are commonly present on images.

Desc.	EASY	HARD	TOUGH	AVG
TFEAT-M	50.0	27.5	13.9	30.5
DDESC	43.0	23.9	12.5	26.5
RSIFT	48.2	20.9	9.4	26.1
DC-S2S	42.2	23.0	11.3	25.5
SIFT	45.3	19.3	8.6	24.4
DC-S	40.0	19.9	9.5	23.1
LIOP	32.6	18.0	9.9	20.2
ORB	30.0	8.6	2.9	13.9
BBOOST	27.2	9.1	3.3	13.2
NCC	22.5	5.2	1.7	9.8
BRIEF	20.5	5.4	1.7	9.2

TABLE 7: Image matching task — mean average precision (*m*AP) for different levels of geometric noise.

Desc.	EASY	HARD	TOUGH	AVG
TFEAT-M	58.4	39.5	25.4	41.1
DDESC	52.4	36.0	24.1	37.5
DC-S2S	52.2	35.9	23.3	37.1
DC-S	52.5	34.5	22.3	36.4
RSIFT	56.7	31.7	18.8	35.7
SIFT	55.4	31.0	18.5	35.0
LIOP	42.8	30.1	21.2	31.3
BBOOST	41.5	21.9	12.4	25.3
ORB	39.8	17.6	9.0	22.2
NCC	39.2	17.2	9.2	21.9
BRIEF	35.1	16.4	8.8	20.1

TABLE 8: Image matching task — success rate % for different levels of geometric noise.

effect on the retrieval performance of all evaluated descriptors (the abscissa is logarithmic): for HARD and TOUGH patches, the performance drops by more than half as the distractor pool size is increased from 10^2 to 10^4. Note that this is a significant drop in performance considering that in application the pool size is likely to be much larger than this.

CNN-based descriptors again outperform the other methods. The good performance of LIOP is also notable, especially for the TOUGH variant of the retrieval task, where it outperforms all other non-deep descriptors and approaches the performance of some of the latter (see for instance Figure 7-right). Another interesting observation is that, while RSIFT performs very well in the EASY scenario, its performance drastically decreases in the HARD and TOUGH scenarios. These results, which are different from what we found in verification and matching, highlight the different nature of that patch retrieval task and emphasize the need for multi-task benchmarks in order to compare descriptors meaningfully.

7.6 Performance across splits & datasets

Next, we assess the statistical stability of the results via cross-validation. In order to do so, we examine the effect of evaluating feature descriptors on the SET A, SET B and SET C splits of $\mathbb{H}$Patches. These splits represent a scenario of average difficulty as they contain patches from the EASY, HARD and TOUGH scenarios. The VIEWP and ILLUM transformation-specific splits are also considered. Table 9 reports results for the three evaluation tasks and the five different splits for the TFEAT-M, DDESC, SIFT and LIOP descriptors.

The absolute values of *m*AP vary within 3% between the SET A, SET B and SET C splits. The difference is thus small compared to the difference between descriptors, which is up to 10%. This suggests that different image sequences are of comparable difficulty, resulting in moderate variance as a function of data selection. Once can therefore expect the ranking of descriptors to be stable with respect to the choice of split.

The difficulty of matching patches with illumination changes versus viewpoint changes can be assessed using the results for the VIEWP and ILLUM splits, that contain only illumination or viewpoint transformations respectively. Across all tasks, the VIEWP is consistently less challenging than the ILLUM. This is interesting as many techniques proposed in the literature focus on geometric rather than photometric invariance. However, $\mathbb{H}$Patches contains sequences with extreme illumination changes (e.g. from night to daylight) which are very challenging to match. Still, descriptors ranking is quite stable, with the notable exception of the improved performance of DDESC in the retrieval task for the ILLUM sequences. This indicates that DDESC is somewhat more robust to illumination changes than other descriptors.

7.7 Training descriptors on $\mathbb{H}$Patches

In order to test the generalization properties of the descriptors, Table 10 reports the results of training and testing TFEAT-M across different splits of $\mathbb{H}$Patches as well as across different

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915233, IEEE Transactions on Pattern Analysis and Machine Intelligence

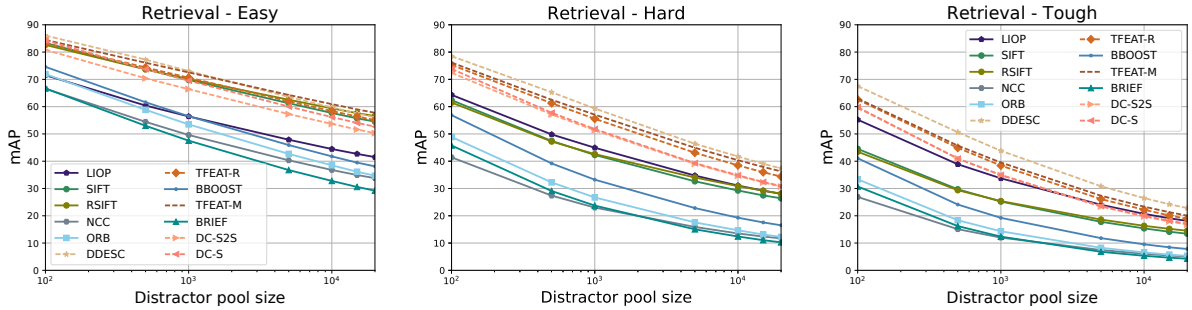

Fig. 7: Patch retrieval results for several state of the art feature descriptors, measured in *m*AP for the EASY, HARD and TOUGH settings. We can observe that the performance significantly decreases with the size of distractor pool.

		TFEAT-M	DDESC	SIFT	LIOP
Verification	SET A	**77.43**	74.73	59.41	69.15
	SET B	**80.33**	76.98	61.16	73.81
	SET C	**77.51**	74.70	61.46	69.63
	VIEWP	**80.71**	76.61	63.27	75.47
	ILLUM	**77.18**	74.86	59.83	67.75
Matching	SET A	**30.45**	26.47	24.41	20.17
	SET B	**35.29**	30.21	26.67	25.87
	SET C	**32.45**	28.25	26.16	22.82
	VIEWP	**36.26**	29.78	27.14	28.86
	ILLUM	**28.97**	26.31	23.77	18.68
Retrieval	SET A	38.02	**38.75**	31.44	29.21
	SET B	**41.82**	41.25	33.93	33.36
	SET C	36.06	**36.71**	30.69	28.87
	VIEWP	**43.45**	40.66	33.40	36.90
	ILLUM	32.33	**35.06**	28.36	23.86

TABLE 9: Performance (*m*AP) of several descriptors on random subsets of HPatches. While the absolute scores can vary, the ranking of the methods is generally stable. Descriptors achieve lower performance on ILLUM sequences compared to VIEWP sequences. For this experiment, TFEAT-M and DDESC are trained on the LIBERTY dataset.

datasets. For the latter, we consider training on the LIBERTY set from the PhotoTourism dataset [63] and testing on HPatches. Note that when training and testing on SET A, SET B and SET C, non-overlapping sets of sequences are used as train and test sets in order to avoid overfitting occurs in this experiment as each split defines training and test set for a holdout cross-validation (80 and 36 sequences respectively).

As expected, results are significantly higher when descriptors are trained and tested on samples drawn from the same distribution even if the individual examples differ in the training and testing splits. Additionally, we can observe that training on ILLUM and testing on VIEWP leads to significantly higher *m*AP, which might indicate that viewpoint sequences do not have much of illumination differences, therefore a descriptor trained for them, does not generalise well for illumination robustness. The above, further demonstrate how specific results can lead to confusing conclusions when the same CNN methods are trained on different datasets.

Finally, TFEAT-M performs nearly 10% better when trained and tested on the same dataset. The inability to perfectly generalize across dataset is not surprising, but it should be noted when assessing descriptors. In particular, this provides some context to many results reported in the literature, where descriptors are trained and tested on PhotoTourism dataset only [63].

Train	Test	Verification	Matching	Retrieval
SET A	SET A	94.46	45.62	70.01
SET B	SET B	95.41	49.83	73.03
SET C	SET C	94.17	47.39	67.22
VIEWP	ILLUM	90.14	32.94	54.52
ILLUM	VIEWP	95.22	50.78	74.43

TABLE 10: Performance of TFEAT-M when trained and tested within the same domain. Each of the sets was split into training and test data. The performance is significantly higher than the results reported in Table 9 where the training data is from Photo-Tourism LIBERTY.

7.8 Descriptor normalisation

It has been shown in [25], [44], [67] that normalisation can often improve the performance of descriptors substantially. In order to study this effect, we consider ZCA whitening [98, p. 299-300]. The ZCA normalised patches are computed as:

$$\tilde{\mathbf{d}} = U\Lambda^{-1/2}U^\top(\mathbf{d}-\bar{\mathbf{d}}), \quad \bar{\mathbf{d}} = \frac{1}{|\mathcal{F}|}\sum_{\mathbf{d}\in\mathcal{F}}\mathbf{d} \tag{2}$$

$$\Sigma = \frac{1}{|\mathcal{F}|-1}\sum_{\mathbf{d}\in\mathcal{F}}(\mathbf{d}-\bar{\mathbf{d}})(\mathbf{d}-\bar{\mathbf{d}})^\top = U\Lambda U^\top \tag{3}$$

where $\mathcal{F} = \{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{P}\}$ is the set of all descriptors computed on a set of patches $\mathcal{P}$.

The method exits in a number of variants; here eigenvalues $\Lambda = diag(\lambda_1, \ldots, \lambda_d)$, sorted in decreasing order, are clipped by thresholding their cumulative distribution by α, defined in [99] as:

$$\lambda_i = \max(\lambda_i, \lambda_r) \quad r = \min k \;\; s.t. \frac{\sum_{i=k}^{d}\lambda_i}{\sum_{i=1}^{d}\lambda_i} < \alpha. \tag{4}$$

This is followed by power law normalisation [67] and then L^2 normalization.

ZCA is unsupervised but still requires a training set of example patches for computing statistics; to this end, we consider a subset of the training dataset from PhotoTourism [5]. Note that SIFT and RSIFT versions are normalized "out of the box" using the L^2 and $L^{\frac{1}{2}}$ norms respectively, but no PCA or ZCA projections, which we add here.

The results, presented in Table 11, clearly show that some descriptors obtain significant performance gains via normalisation; for example, the performance of RSIFT increases by nearly 20% in the verification task. CNN based descriptors benefit less, but there is still a noticeable improvement; for example, normalization

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915233, IEEE Transactions on Pattern Analysis and Machine Intelligence

	Verification		Matching		Retrieval	
norm	no	ZCA_{pl}	no	ZCA_{pl}	no	ZCA_{pl}
TFEAT-M	77.43	78.75	30.45	33.23	38.02	39.09
DDESC	74.73	79.26	26.47	34.37	38.75	44.76
DC-S	65.08	78.18	23.12	31.00	33.54	40.61
SIFT	59.41	69.02	24.41	24.94	31.44	39.02
RSIFT	51.53	70.11	26.12	**35.35**	33.06	41.64

TABLE 11: Effect of ZCA whitening and power low normalization [25], [67]. Surprisingly, for matching and retrieval, normalised version of hand-crafted descriptors perform on par with the recent deep learning methods. Results are presented for SET A.

boosts DDESC by up to 8%, while TFEAT-M gains slightly less than 3%.

It may be surprising that learned descriptors can be improved by applying off-the-shelf normalization to them. This may be due to the use of learning formulation that do not perfectly match the task or to overfitting the training data.

Overall, results clearly show that it is worth applying ZCA normalization to all descriptors, especially due to the simplicity and low computational cost of this procedure. Consistent improvements can be expected in the verification, matching and retrieval tasks.

7.9 Evaluation of state-of-the-art descriptors

So far we have explored different variants of our dataset and evaluation protocol, using for this purpose a subset of descriptors We now consider the most informative scenarios identified above and use them to evaluate systematically all our state-of-the-art descriptors on those. Results are presented compactly in Figure 8.

The aim is to tune each descriptor and normalization procedure in order to maximize performance while avoiding overfitting on $\mathbb{H}$Patches. For this purpose, descriptors and normalization parameters are learned (when applicable) either on LIBERTY (Figure 8) or on $\mathbb{H}$Patches (Figure 9). In the latter case evaluation uses cross-validation.

Figure 8 clearly shows that verification is the least challenging of the tasks, with top performers reaching 90% *m*AP. The results vary significantly for different experimental settings. The most important factors of variations are: choosing between the EASY, HARD, and TOUGH scenarios, choosing between the viewpoint VIEWP and illumination ILLUM splits (with the latter being more challenging), and sampling distractors from the same (INTRA-SEQUENCE) or different (INTER-SEQUENCE) sequences.

Descriptor normalisation. Next, we tune projection and power law normalisation (Section 7.8) to maximize the performance of each descriptor. In practice, descriptor normalisation has many variants, such as using ZCA or PCA whitening with or without Eigen value clipping, use of power law normalisation or L^2 normalisation. In may cases, the decision of what normalisation to use depends not only on the descriptor, but also on the task and data. To account for these factors, we use grid search to find the best performing method for each descriptor in two different dataset: LIBERTY (*Normalisation with* LIBERTY in Figure 8) or over splits of the $\mathbb{H}$Patches (*Normalisation with* $\mathbb{H}$*Patches* in Figure 9). In order to avoid overfitting when searching for the best normalisation on $\mathbb{H}$Patches (as we evaluate on $\mathbb{H}$Patches), we perform cross-validation on the three splits of the dataset. Each split comprises a training and testing subsets; in order to also obtain a validation subset for each split (used to pick the best normalisation), we divide its training into 20 sequences used for validation of the normalisation method and 60 sequences for training — in this case for computing the ZCA whitening. This means that for each descriptor we obtain three different normalisation methods, one for each split. We divide the training set in order to avoid fitting the best normalisation on the held out test set for each split. The final score for a descriptor is an average performance over the remaining 36 sequences of each split.

Dataset	ZCA α				Power Law Norm				L2 Norm			
	Lib	HA	HB	HC	Lib	HA	HB	HC	Lib	HA	HB	HC
SIFT	.4	.3	.2	.3	✓	✓	✓	✓	✓	✓	✓	✓
RSIFT	.4	.2	.2	.2	✓	✓	✓	✓	✓	✓	✓	✓
KDE	.4	.1	.1	.1	✓	✓	✓	✓	✓	✓	✓	✓
MKD	-	.2	.1	.1		✓	✓	✓		✓	✓	✓
TFEAT-M	-	.4	.4	.4	✓	✓	✓	✓		✓	✓	✓
TNET	-	-	-	-		✓	✓	✓		✓	✓	✓
L2NET	-	.3	.2	.2		✓	✓	✓		✓	✓	✓
HNET	-	-	-	-					✓	✓	✓	✓

TABLE 12: Best normalisation method found for the real-valued descriptors using different datasets. The method is found using grid search over different combinations of normalisation methods. The ZCA parameter α is defined in [99, Eq. 13].

The resulting best normalisation methods found with grid search for each descriptor and dataset are summarised in Table 12. For the ZCA parameter α (eq. (4)) in the grid search we consider $\alpha \in \{0, .05, .075, .1, .15, .2, .25, .3, .4\}$. From the table we can see, that the normalisation method differs mainly between datasets. For example, descriptors learnt on LIBERTY, generally perform best without any normalisation (or simple L^2 normalisation). However, we can see that choice of normalisation method is relatively stable between different splits of $\mathbb{H}$Patches. The main difference between normalisations found for LIBERTY and $\mathbb{H}$Patches is the ratio of eigenvalues being clipped.

As discussed in Section 7.8, normalisation significantly improves the absolute score of all descriptors, although the CNN based methods seem to benefit less compared to handcrafted ones, with HNET seeing almost no improvement whatsoever. Contrary to that, we can see that for hand-crafted descriptors it is important to adjust the normalisation to the target domain, mainly for the verification task (e.g.RSIFT has 67.56% *m*AP when normalised with LIBERTY, while achieves 76.6% when normalised with a subset of the target dataset). Overall, in both cases the normalisation has only limited effect on the descriptor's rank.

Training with $\mathbb{H}$Patches. At the bottom of Figure 9 we show the average results of two learned descriptors, TFEAT-M and HNET, trained and tested on the three splits SET A, SET B, SET C of the $\mathbb{H}$Patches[4], reporting cross-validated average scores (hence these numbers are not directly comparable to other results from Figure 8).

We can see that, compared to TFEAT-M, HNET benefits the most from training on $\mathbb{H}$Patches, with the largest gains obtained on viewpoint sequences. However, both descriptors saturate the verification task. This confirms the observations from Section 7.6 that within-domain training can lead to overoptimistic conclusions.

4. Please note that for obtaining these results, one has to train three different descriptors for each split which might be computationally demanding.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915233, IEEE Transactions on Pattern Analysis and Machine Intelligence

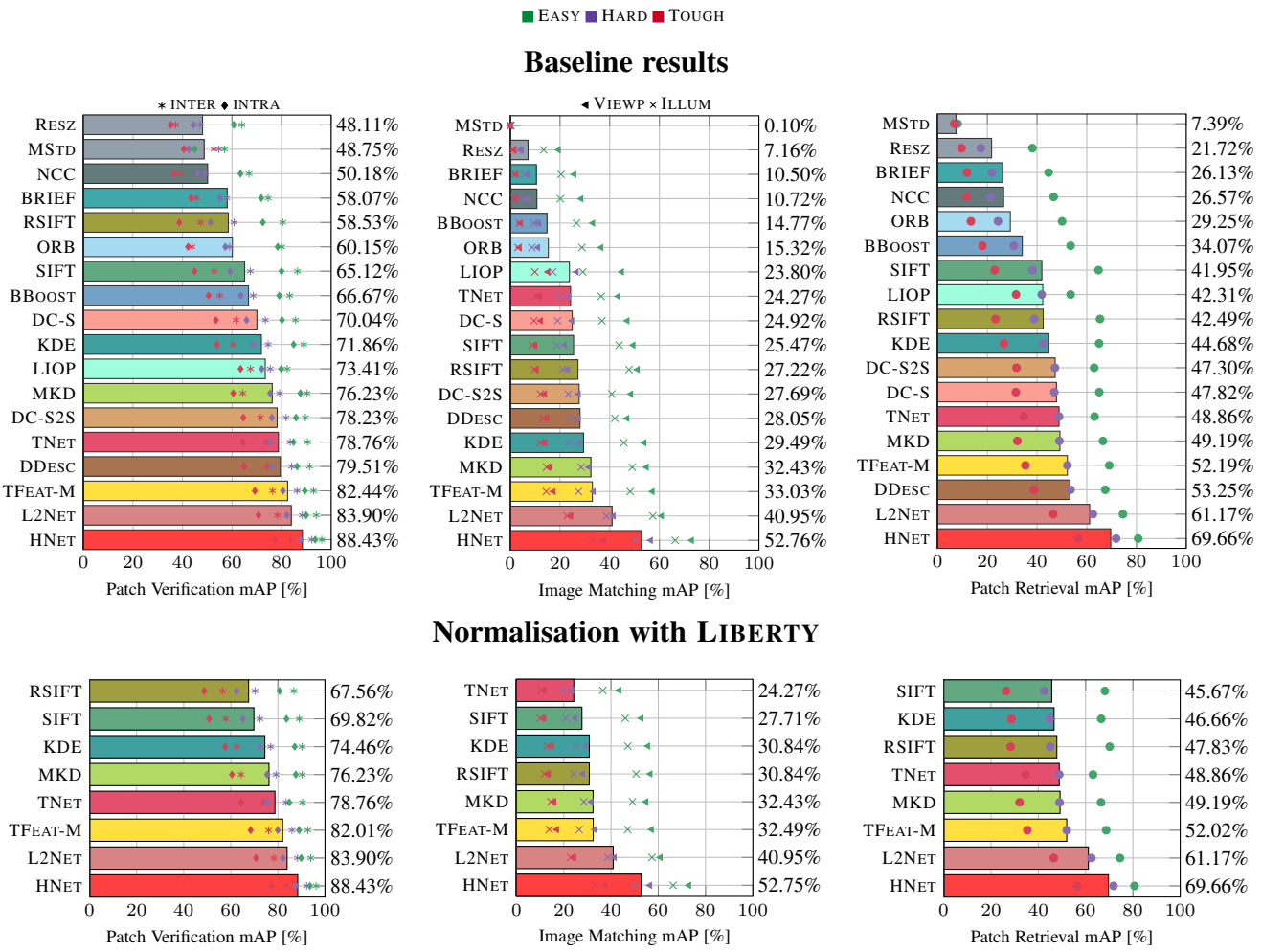

Fig. 8: Verification, matching and retrieval results for descriptors trained on LIBERTY and tested on the **all sequences of the HPatches**. Colour bars are means over all the variants of each task. Colour of the marker indicates EASY, HARD, and TOUGH geometric noise. The type of the marker corresponds to different experimental settings i.e. negative pair patches are sampled from the same image sequences INTRA-SEQUENCE or different INTER-SEQUENCE, the results are averaged for viewpoint VIEWP or illumination ILLUM sequences (see Section 7.3, 7.4 and 7.5 for details). Figures titled *Normalisation on* LIBERTY show results for best normalisation found using the LIBERTY dataset.

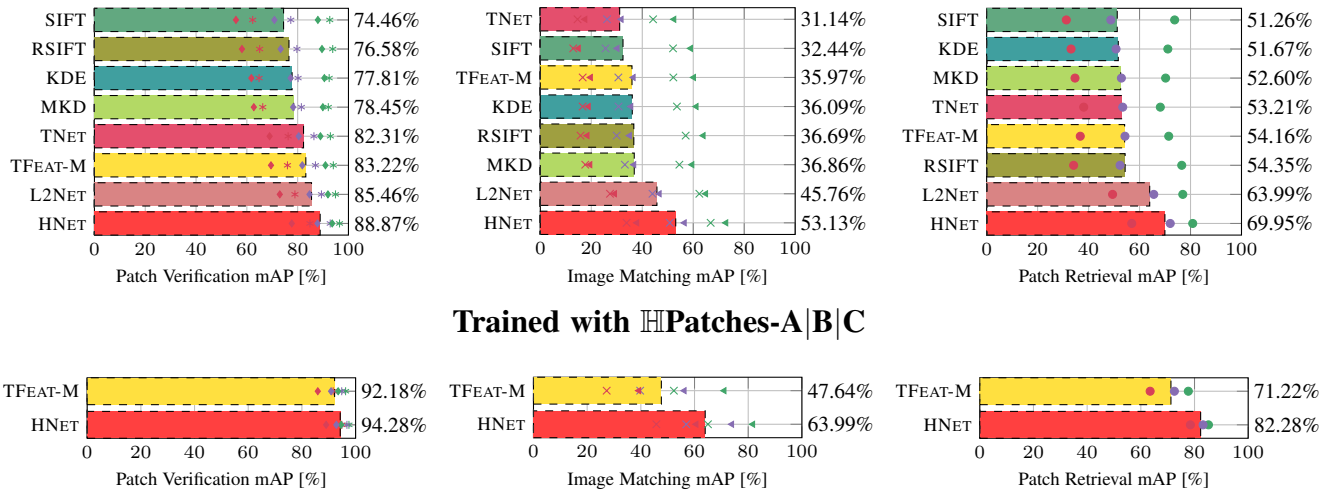

Fig. 9: Verification, matching and retrieval results for descriptors as an **average over three splits of HPatches** (SET A, SET B, SET C). This figure uses same visualisation as Figure 8. Figures titled *Normalisation on HPatches* show results using descriptors normalised with a best method found on the training set of the split. In this case, all descriptors are learnt on LIBERTY. Bottom figures report the average results for two descriptors trained and tested over three different splits of the HPatches dataset.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915233, IEEE Transactions on Pattern Analysis and Machine Intelligence

8 Conclusions

With the advent of deep learning, the development of novel and more powerful local descriptors has accelerated tremendously. However, as we have shown in this paper, the benchmarks commonly used for evaluating such descriptors are inadequate, making comparisons unreliable. In the long run, this is likely to be detrimental to further research. In order to address this problem, we have introduced $\mathbb{H}$Patches, a new public benchmark for local descriptors. The new benchmark is patch-based, removing many of the ambiguities that plagued the existing image-based benchmarks and favouring rigorous, reproducible, and large scale experimentation. This benchmark also improves on the limited data and task diversity present in other datasets, by considering many different scene and visual effects types, as well as three benchmark tasks close to practical applications of descriptors.

Despite the multitask complexity of our benchmark suite, using the evaluation is easy as we provide open-source implementation of the protocols which can be used with minimal effort. $\mathbb{H}$Patches can supersede datasets such as PhotoTourism and the older but still frequently used Oxford matching dataset, addressing their shortcomings and providing a valuable tool for researchers interested in local descriptors.

We have performed extensive evaluation of the state of the art descriptors and demonstrated their properties and relative performance in various experimental settings. The ranking of the descriptors is relatively stable subject to small variations in some settings but the top performers were HNet [80] and L2Net [81]. Even though the CNN based descriptors significantly outperform traditional handcrafted features, a proper post processing normalization of features can significantly boost the performance of all methods and bridge the gap between SIFT and modern learning based descriptors. However, for the handcrafted descriptors, this involves finding the best normalisation for the target domain.

Additionally, we introduce baseline results and a methodology for training on the $\mathbb{H}$Patches by defining three fixed splits of the dataset to perform a form of cross validation.

Acknowledgments

This research was supported by ERC 677195-IDIU and by EPSRC EP/N007743/1. We would like to thank Dmytro Mishkin for providing networks trained on the splits of our dataset, and for the helpful discussions. We would like to thank Giorgos Tolias for help with descriptor normalisation. We would also like to thank Akis Tsiotsios for providing the camera used for collecting the data.

References

[1] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. ICCV*, vol. 2, 1999, pp. 1150–1157.
[2] K. Mikolajczyk and J. Matas, "Improving descriptors for fast tree matching by optimal linear projection," in *Proc. ICCV*, 2007.
[3] T. Trzcinski, M. Christoudias, and V. Lepetit, "Learning image descriptors with boosting," *IEEE PAMI*, vol. 37, no. 3, pp. 597–610, 2015.
[4] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE PAMI*, vol. 27, no. 10, pp. 1615–1630, 2005.
[5] S. Winder, G. Hua, and M. Brown, "Picking the best daisy," in *Proc. CVPR*, 2009.
[6] O. Miksik and K. Mikolajczyk, "Evaluation of local detectors and descriptors for fast feature matching," in *Proc. ICPR*, 2012, pp. 2681–2684.
[7] Z. Wang, B. Fan, and F. Wu, "Local intensity order pattern for feature description," in *Proc. ICCV*, 2011, pp. 603–610.
[8] T.-Y. Yang, Y.-Y. Lin, and Y.-Y. Chuang, "Accumulated stability voting: A robust descriptor from descriptors of multiple scales," in *Proc. CVPR*, 2016, pp. 327–335.
[9] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. ICCV*, 2011, pp. 2548–2555.
[10] G. Levi and T. Hassner, "LATCH: learned arrangements of three patch codes," in *Winter Conference on Applications of Computer Vision (WACV)*, 2016. [Online]. Available: http://www.openu.ac.il/home/hassner/projects/LATCH
[11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. ICCV*, 2011, pp. 2564–2571.
[12] V. Balntas, L. Tang, and K. Mikolajczyk, "BOLD - binary online learned descriptor for efficient image matching," in *Proc. CVPR*, 2015.
[13] K. Lenc, V. Gulshan, and A. Vedaldi, "Vlbenchmkars," http://www.vlfeat.org/benchmarks/xsxs, 2011.
[14] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
[15] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: a survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2008.
[16] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. AVC*, 1988, pp. 147–151.
[17] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," *Proc. ECCV*, 2002.
[18] P. Mainali, G. Lafruit, Q. Yang, B. Geelen, L. Van Gool, and R. Lauwereins, "Sifer: scale-invariant feature detector with error resilience," *IEEE PAMI*, vol. 104, no. 2, pp. 172–197, 2013.
[19] P. Mainali, G. Lafruit, K. Tack, L. Van Gool, and R. Lauwereins, "Derivative-based scale invariant image feature detector with error resilience," *IEEE PAMI*, vol. 23, no. 5, pp. 2380–2391, 2014.
[20] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "Kaze features," in *Proc. ECCV*. Springer, 2012, pp. 214–227.
[21] C. L. Zitnick and K. Ramnath, "Edge foci interest points," in *Proc. ICCV*, 2011, pp. 359–366.
[22] S. Salti, A. Lanza, and L. Di Stefano, "Keypoints from symmetries by wave propagation," in *Proc. CVPR*, 2013.
[23] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.
[24] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," *Proc. ECCV*, pp. 430–443, 2006.
[25] A. Bursuc, G. Tolias, and H. Jégou, "Kernel local descriptors with implicit rotation matching," in *ACM ICMR*, 2015, pp. 595–598.
[26] A. Mukundan, G. Tolias, and O. Chum, "Multiple-kernel local-patch descriptor," in *British Machine Vision Conference*, 2017.
[27] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE PAMI*, vol. 32, no. 1, pp. 105–119, 2010.
[28] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *Proc. ECCV*. Springer, 2010, pp. 183–196.
[29] C. Strecha, A. Lindner, K. Ali, and P. Fua, "Training for task specific keypoint detection," in *Joint Pattern Recognition Symposium*, 2009.
[30] J. Šochman and J. Matas, "Learning fast emulators of binary decision processes," *IJCV*, vol. 83, no. 2, pp. 149–163, 2009.
[31] A. Richardson and E. Olson, "Learning convolutional filters for interest point detection," in *Proc. ICRA*. IEEE, 2013.
[32] L. Trujillo and G. Olague, "Using evolution to learn how to perform interest point detection," in *Proc. ICPR*, 2006.
[33] W. Hartmann, M. Havlena, and K. Schindler, "Predicting matchability," in *Proc. CVPR*, 2014.
[34] H. Altwaijry, A. Veit, and S. J. Belongie, "Learning to detect and match keypoints with deep architectures." in *Proc. BMVC*, 2016.
[35] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: Learned invariant feature transform," in *Proc. ECCV*. Springer, 2016, pp. 467–483.
[36] Y. Verdie, K. Yi, P. Fua, and V. Lepetit, "Tilde: a temporally invariant learned detector," in *Proc. CVPR*, 2015, pp. 5279–5288.
[37] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.
[38] K. Lenc and A. Vedaldi, "Learning covariant feature detectors," in *ECCV Workshop on Geometry Meets Deep Learning*, 2016.

[39] X. Zhang, F. X. Yu, S. Karaman, and S.-F. Chang, "Learning discriminative and transformation covariant local feature detectors," in *Proc. CVPR*, 2017, pp. 6818–6826.

[40] D. Mishkin, F. Radenovic, and J. Matas, "Learning discriminative affine regions via discriminability," *arXiv preprint arXiv:1711.06704*, 2017.

[41] J. J. Koenderink and A. J. van Doorn, "Representation of local geometry in the visual system," *Biological cybernetics*, vol. 55, no. 6, pp. 367–375, 1987.

[42] W. T. Freeman, E. H. Adelson *et al.*, "The design and use of steerable filters," *IEEE PAMI*, 1991.

[43] L. Van Gool, T. Moons, and D. Ungureanu, "Affine/photometric invariants for planar intensity patterns," in *Proc. ECCV*, 1996.

[44] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. CVPR*, vol. 2, 2004, pp. II–506.

[45] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[46] J. Dong and S. Soatto, "Domain-size pooling in local descriptors: Dsp-sift," in *Proc. CVPR*, 2015, pp. 5097–5106.

[47] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, vol. 1. IEEE, 1994, pp. 582–585.

[48] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Proc. ECCV*, 2010, pp. 778–792.

[49] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Proc. ECCV*, pp. 404–417, 2006.

[50] M. Brown, R. Szeliski, and S. Winder, "Multi-image matching using multi-scale oriented patches," in *Proc. CVPR*, 2005.

[51] B. Fan, F. Wu, and Z. Hu, "Aggregating gradient distributions into intensity orders: A novel local image descriptor," in *Proc. CVPR*. IEEE, 2011, pp. 2377–2384.

[52] A. Ziegler, E. Christiansen, D. Kriegman, and S. J. Belongie, "Locally uniform comparison image descriptor," in *Advances in Neural Information Processing Systems*, 2012, pp. 1–9.

[53] K. M. Yi, Y. Verdie, P. Fua, and V. Lepetit, "Learning to assign orientations to feature points," in *Computer Vision and Pattern Recognition (CVPR)*, no. EPFL-CONF-217982, 2016.

[54] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*. Ieee, 2012, pp. 510–517.

[55] T. Trzcinski and V. Lepetit, "Efficient discriminative projections for compact binary descriptors," in *Proc. ECCV*, 2012, pp. 228–242.

[56] X. Xu, L. Tian, J. Feng, and J. Zhou, "Osri: A rotationally invariant binary descriptor," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 2983–2995, 2014.

[57] S. Zhang, Q. Tian, Q. Huang, W. Gao, and Y. Rui, "Usb: Ultrashort binary descriptor for fast visual matching and retrieval," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3671–3683, 2014.

[58] T. Trzcinski, M. Christoudias, and V. Lepetit, "Learning image descriptors with boosting," *IEEE PAMI*, vol. 37, no. 3, pp. 597–610, 2015.

[59] 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 *Proc. CVPR*. IEEE, 2009, pp. 2504–2511.

[60] S. Taylor and T. Drummond, "Binary histogrammed intensity patches for efficient and robust matching," *IJCV*, 2011.

[61] K. Lin, J. Lu, C.-S. Chen, and J. Zhou, "Learning compact binary descriptors with unsupervised deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1183–1192.

[62] S. Winder and M. Brown, "Learning local image descriptors," in *Proc. CVPR*, 2007.

[63] M. Brown, G. Hua, and S. Winder, "Discriminative learning of local image descriptors," *IEEE PAMI*, vol. 33, no. 1, pp. 43–57, 2011.

[64] H. Cai, K. Mikolajczyk, and J. Matas, "Learning linear discriminant projections for dimensionality reduction of image descriptors," *IEEE PAMI*, vol. 33, no. 2, pp. 338–352, 2011.

[65] C. Strecha, A. Bronstein, M. Bronstein, and P. Fua, "Ldahash: Improved matching with smaller descriptors," *IEEE PAMI*, vol. 34, no. 1, pp. 66–78, 2012.

[66] K. Simonyan, A. Vedaldi, and A. Zisserman, "Learning local feature descriptors using convex optimisation," *IEEE PAMI*, vol. 36, no. 8, pp. 1573–1585, 2014.

[67] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. CVPR*, 2012, pp. 2911–2918.

[68] M. Jahrer, M. Grabner, and H. Bischof, "Learned local descriptors for recognition and matching," in *Proceedings of the Computer Vision Winter Workshop*, 2008.

[69] C. Osendorfer, J. Bayer, S. Urban, and P. Van Der Smagt, "Convolutional neural networks learn compact local image descriptors," in *International Conference on Neural Information Processing*. Springer, 2013, pp. 624–630.

[70] P. Fischer, A. Dosovitskiy, and T. Brox, "Descriptor matching with convolutional neural networks: a comparison to sift," *arXiv preprint arXiv:1405.5769*, 2014.

[71] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International conference on machine learning*, 2014, pp. 647–655.

[72] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *European conference on computer vision*. Springer, 2014, pp. 392–407.

[73] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 9, pp. 1704–1716, 2012.

[74] M. Paulin, J. Mairal, M. Douze, Z. Harchaoui, F. Perronnin, and C. Schmid, "Convolutional patch representations for image retrieval: an unsupervised approach," *IJCV*, 2017.

[75] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "Matchnet: Unifying feature and metric learning for patch-based matching," in *Proc. CVPR*, 2015, pp. 3279–3286.

[76] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proc. CVPR*, 2015.

[77] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," *Proc. ICCV*, 2015.

[78] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks," *Proc. BMVC*, 2016.

[79] N. Markus, I. S. Pandzic, and J. Ahlberg, "Learning local descriptors by optimizing the keypoint-correspondence criterion," in *Proc. ICPR*, 2016.

[80] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," 2017, pp. 4826–4837.

[81] B. F. Y. Tian and F. Wu, "L2-net: Deep learning of discriminative patch descriptor in euclidean space," in *Proc. CVPR*, vol. 2, 2017.

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

[83] 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 *Proc. Data Compression Conference*, 2014, pp. 3–12.

[84] V. Balntas, "Efficient learning of local image descriptors," Ph.D. dissertation, University of Surrey, 2016.

[85] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid, "Local convolutional features with unsupervised training for image retrieval," in *Proc. ICCV*, 2015, pp. 91–99.

[86] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proc. ICML*, 2006, pp. 233–240.

[87] T. Fawcett. (2004) Roc graphs: Notes and practical considerations for researchers.

[88] H. Aanæs, A. L. Dahl, and K. S. Pedersen, "Interesting interest points," *IJCV*, vol. 97, no. 1, pp. 18–35, 2012.

[89] P. Fischer, A. Dosovitskiy, and T. Brox, "Descriptor matching with convolutional neural networks: a comparison to sift," *arXiv preprint arXiv:1405.5769*, 2014.

[90] K. Cordes, B. Rosenhahn, and J. Ostermann, "Increasing the accuracy of feature evaluation benchmarks using differential evolution," in *Proc. SDE*, 2011, pp. 1–8.

[91] N. Jacobs, N. Roman, and R. Pless, "Consistent temporal variations in many outdoor scenes," in *Proc. CVPR*, 2007, pp. 1–6.

[92] V. Vonikakis, D. Chrysostomou, R. Kouskouridas, and A. Gasteratos, "Improving the robustness in feature detection by local contrast enhancement," in *Proc. IST*, 2012, pp. 158–163.

[93] G. Yu and J.-M. Morel, "ASIFT: an algorithm for fully affine invariant comparison," *Image Processing On Line*, vol. 1, 2011.

[94] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *IJCV*, vol. 65, no. 1-2, pp. 43–72, 2005.

[95] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. ICCV*, 2007, pp. 1–8.

[96] ——, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proc. CVPR*, 2008, pp. 1–8.
[97] J. P. Lewis, "Fast template matching," *Vision Interface*, vol. 95, pp. 120–123, 1995. [Online]. Available: http://www.idiom.com/~{}zilla/Papers/nvisionInterface/nip.html#VI95
[98] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1995.
[99] G. Hua, M. Brown, and S. Winder, "Discriminant embedding for local image descriptors," in *Proc. ICCV*, 2007, pp. 1–8.

Vassileios Balntas Vassileios Balntas is a postdoc at the University of Oxford. Previously he was a visiting PhD student at Computer Vision and Learning Lab, Imperial College London, UK, a PhD student at the University of Surrey, UK and a research assistant at National Technical University of Athens, Greece. He holds a MSc from University of Surrey, UK, and a MEng from Democritus University of Thrace, Greece.

Karel Lenc Karel Lenc is a Research Assistant at the Department of Engineering Science, University of Oxford.

Andrea Vedaldi is an associate professor in the Visual Geometry Group at the University of Oxford.

Tinne Tuytelaars is a professor in the Electrotechnical Department, KU Leuven. Her research focuses on image understanding, including object, scene, and action recognition. In 2009, she received an ERC starting independent researcher grant. She has been one of the program chair of the European Conference on Computer Vision 2014 and one of the general chairs of IEEE Conference on Computer Vision and Pattern Recognition 2016.

Jiri Matas is a professor at the Center for Machine Perception, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic. He is author or co-author of more than 250 papers in the area of computer vision and machine learning. His research interests include object recognition, image retrieval, tracking, sequential pattern recognition, invariant feature detection, and Hough transform and RANSAC-type optimization.

Krystian Mikolajczyk Krystian Mikolajczyk is an Associate Professor at Imperial College London. He completed his PhD degree at the Institute National Polytechnique de Grenoble and held a number of research positions at INRIA, University of Oxford and Technical University of Darmstadt, as well as faculty positions at the University of Surrey, and Imperial College London. His main area of expertise is in image and video recognition, in particular methods for image representation and learning. He has served in various roles at major international conferences co-chairing British Machine Vision Conference 2012, 2017 and IEEE International Conference on Advanced Video and Signal-Based Surveillance 2013. In 2014 he received Longuet-Higgins Prize awarded by the Technical Committee on Pattern Analysis and Machine Intelligence of the IEEE Computer Society.