R2D2: Repeatable and Reliable Detector and Descriptor

Jerome Revaud Philippe Weinzaepfel César De Souza Noe Pion
Gabriela Csurka Yohann Cabon Martin Humenberger
NAVER LABS Europe
firstname.lastname@naverlabs.com

Abstract

Interest point detection and local feature description are fundamental steps in many computer vision applications. Classical methods for these tasks are based on a detect-then-describe paradigm where separate handcrafted methods are used to first identify repeatable keypoints and then represent them with a local descriptor. Neural networks trained with metric learning losses have recently caught up with these techniques, focusing on learning repeatable saliency maps for keypoint detection and learning descriptors at the detected keypoint locations. In this work, we argue that salient regions are not necessarily discriminative, and therefore can harm the performance of the description. Furthermore, we claim that descriptors should be learned only in regions for which matching can be performed with high confidence. We thus propose to jointly learn keypoint detection and description together with a predictor of the local descriptor discriminativeness. This allows us to avoid ambiguous areas and leads to reliable keypoint detections and descriptions. Our detection-and-description approach, trained with self-supervision, can simultaneously output sparse, repeatable and reliable keypoints that outperforms state-of-the-art detectors and descriptors on the HPatches dataset. It also establishes a record on the recently released Aachen Day-Night localization benchmark.

1. Introduction

Being able to accurately find and describe similar points of interest (or simply *keypoints*) across images is crucial in many applications such as large-scale visual localization [53, 45], object detection [7], pose estimation [31], Structure-from-Motion (SfM) [49] and 3D reconstruction [20]. In these applications, extracted keypoints should be sparse, repeatable, and discriminative, in order to minimize the memory footprint while maximizing matching accuracy.

Classical approaches to implement this ability are based on a two-stage pipeline that first detects keypoint locations [28, 17, 27, 26] and then computes a local descriptor for each keypoint [4, 25]. Specifically, the role of the keypoint detector is to look for scale-space locations with covariance with respect to camera viewpoint changes and invariance with respect to photometric transformations. A large number of handcrafted keypoints have been shown to work well in practice, such as corners [17] or blobs [27, 26, 25]. As for the description, various schemes based on histograms of local gradients [5, 4, 23, 42], whose most well known instance is SIFT [25], have been developed and are still extensively used nowadays.

Despite this apparent success, this paradigm was recently challenged by several approaches willing to replace the handcrafted parts by data-driven approaches [56, 29, 34, 62, 59, 16, 55, 32, 48]. Arguably, handcrafted methods are limited by the a priori knowledge researchers have about the tasks at hand. The point is thus to let a deep network discover automatically which feature extraction process and representation are most suited to the data. The few attempts for learning keypoint detectors [48, 35, 59, 9, 11] have only focused on the repeatability.

On the other hand, metric learning techniques applied to learning local robust descriptors [32, 55] have recently outperformed traditional descriptors, including SIFT [19]. They are trained on the repeatable locations provided by the detector, which may harm the performance in regions that are repeatable but where accurate matching is not possible. Figure 1 shows such an example with a checkerboard image: every corner or blob is repeatable but matching cannot be performed due to repetitiveness of the cells. In natural images, common textures such as the tree leafage, skyscraper windows or sea waves are also salient, but hard to match. In this work, we claim that detection and description are inseparably tangled since good keypoints should not only be *repeatable* but should also be *reliable* for matching. We thus propose to learn jointly the descriptor reliability seamlessly with the detection and description processes. Our method separately estimates a confidence map for each of these two aspects and selects only keypoints which are both repeatable and reliable to improve the overall feature matching pipeline.

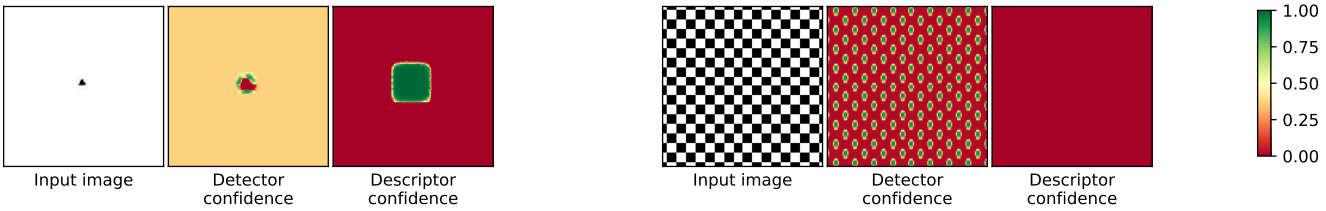


Figure 1. Toy examples to illustrate the key difference between repeatability (2nd column) and reliability (3rd column) for a given image. Repeatable regions in the first image are only located near the black triangle, however, all patches containing it are equally reliable. In contrast, all squares in the checkerboard pattern are salient hence repeatable, but none of them is discriminative due to self-similarity. Both confidence maps were estimated by our network.

More precisely, our network, illustrated in Figure 2, outputs dense local descriptors (one for each pixel) as well as two associated repeatability and reliability confidence maps. The two maps are in fact estimates of the probabilities that a keypoint is respectively repeatable and that its descriptor is discriminative, *i.e.*, it can be accurately matched with high confidence. Finally, keypoints correspond to locations that maximize both confidence maps.

To train the keypoint detector, we employ a novel unsupervised loss that encourages repeatability, sparsity as well as a uniform coverage of the image. As for the local descriptor, it is trained with a listwise ranking loss, leveraging recent advances in metric learning based on an approximated Average Precision (AP) metric, instead of using a standard triplet or contrastive loss [14, 50, 38]. We jointly learn a reliability confidence value to predict which pixels will have descriptors with a high AP, *i.e.*, that are both discriminative, robust and in the end that can be accurately matched. Our experiments on several benchmarks show that our formulation elegantly combines the repeatability and sparsity of the detector with a discriminative and robust descriptor.

In summary, we make three contributions:

- We propose a novel unsupervised loss to learn a keypoint detector: our keypoints are sparse while still uniformly covering the image and are more repeatable than other methods.
- A new loss to learn reliable local descriptors while explicitly estimating their reliability at the same time.
- Our combined pipeline selects keypoints which are both repeatable and reliable and achieves state-of-the-art results.

2. Related work

Local feature extraction and description play a vital role in several high-order methods in computer vision and has received a continuous influx of attention in the past several years (*c.f.* surveys in [8, 13, 43, 57]). Most existing works rely on a *detect-then-describe* approach and we focus here on the learning approaches only.

Learned descriptors. Most deep feature matching methods have focused on learning the descriptor component, applied either on a sparse set of keypoints [3, 51, 52, 29] detected using standard handcrafted methods or densely over the image [12, 47, 54, 32]. The descriptor is usually trained using a metric learning loss, such as the triplet loss [14, 50] or a contrastive loss [38]. Such loss formulation has been also used to improve descriptors for image patches [55, 16, 1]. Our approach has several advantages compared to these methods: (a) the detector is trained jointly with the descriptor, alleviating the drawbacks of sparse handcrafted keypoint detector; (b) the descriptor component is trained with an approximation of the AP loss, considering more descriptors per batch than standard ranking losses (c) we jointly estimate the descriptor reliability for local feature matching.

Learned detectors. A few attempts have been recently made to learn the detector component. The first approach for keypoint detection to rely on machine learning was FAST [41]. Later, Di *et al*. [10] learn to mimic the output of handcrafted detectors with a compact neural network. In [22], handcrafted and learned filters are combined to detect repeatable keypoints. These two approaches still rely on some handcrafted detectors or filters while ours is trained without such prior knowledge. QuadNet [48] is an unsupervised approach based on the idea that the ranking of the keypoint saliencies should be preserved by natural image transformations. Following a similar approach, Zhang *et al*. [60] additionally encourage peakiness of the saliency map for keypoint detector on textures. In this paper we employ a simpler unsupervised formulation that locally enforces the similarity of the saliency maps.

Jointly learned descriptor and detector. In the seminal LIFT approach, Yi *et al*. [59] introduced a pipeline where keypoints are detected and cropped regions are then fed to a second network to estimate the orientation before going throughout a third network to perform description. Recently, the SuperPoint approach by DeTone *et al*. [9] tackles keypoint detection as a supervised task learned from artificially generated training images containing basic structures like

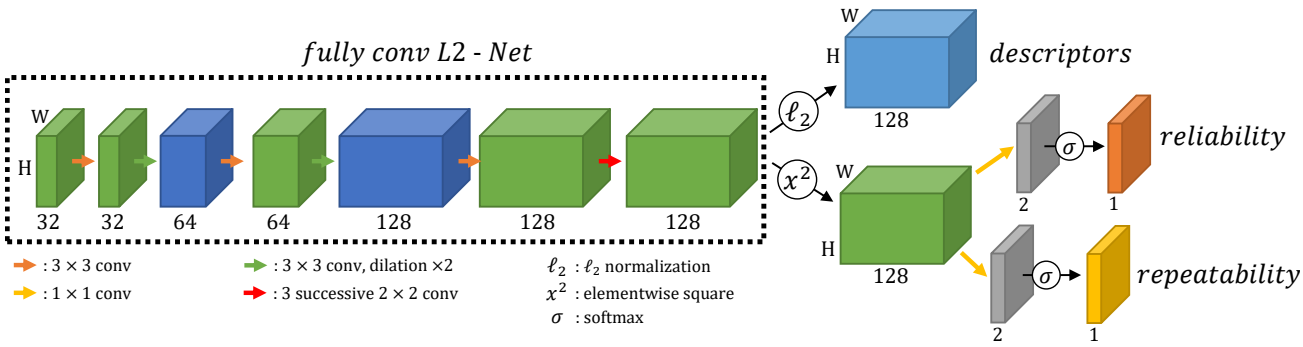


Figure 2. Overview of our network for jointly learning repeatable and reliable matches.

corners and edges. Keypoints are then arbitrarily defined as intersection of these structures or remarkable points within, and a deep descriptor is learned jointly, sharing some of the computation. In contrast, our approach does not introduce a bias in the locations of keypoints and also does not require to compute repeatability multiple times for a given test image with different homographies. Noh *et al*. [32] proposed DELF, an approach targeted for image retrieval that learns local features as a by-product of a classification loss coupled with an attention mechanism trained using a large-scale dataset of landmark images. In comparison, our approach is unsupervised and trained with relatively little data. More similar to our approach, Mishkin *et al*. [30] recently leverage deep learning to jointly enhance an affine regions detector and local descriptors. Nevertheless, their approach is rooted on a handcrafted keypoint detector that generates seeds for the affine regions, thus not truly learning keypoint detection.

More recently, D2-Net [11] uses a single CNN for joint detection and description that share all weights; the detection being based on local maxima across the channels and the spatial dimensions of the feature maps. Instead of arbitrarily defining keypoints as local maxima in the descriptor space, our approach explicitly estimates the keypoint reliability and repeatability. Finally, Ono *et al*. [35] train a network from pairs of matching images with a complicated asymmetric gradient backpropagation scheme for the detection and a triplet loss for the local descriptor. Compared to these works, for the first time we jointly train a sparse keypoint detector with a deep descriptor enhanced with a reliability confidence value such that ambiguous areas are avoided.

3. Joint learning reliable and repeatable detectors and descriptors

The proposed approach, referred to as R2D2, aims to predict a set of sparse locations of an input image I that are repeatable and reliable for the purpose of local feature matching. In contrast to classical approaches, we make an explicit distinction between repeatability and reliability, see Figure 1. We claim that they are in fact two complementary aspects that must be predicted separately.

We thus propose to train a fully-convolutional network (FCN) that predicts 3 outputs for an image I of size $H \times W$. The first one is a 3D tensor $\boldsymbol{X} \in \mathbb{R}^{H \times W \times D}$ that corresponds to a set of dense D-dimensional, one per pixel. The second one is a heatmap $\boldsymbol{S} \in [0,1]^{H \times W}$ whose goal is to provide sparse and repeatable keypoint locations. To achieve sparsity, we only extract keypoints at locations corresponding to local maxima in $\boldsymbol{S}$, while $\boldsymbol{S}$ is trained to contain strong and repeatable local maxima. The third output is an associated reliability map $\boldsymbol{R} \in [0,1]^{H \times W}$ that indicates the estimated reliability, *i.e*., discriminativeness, of descriptor $\boldsymbol{X}_{ij}$ at each pixel (i,j), with $i = 1..W$ and $j = 1..H$.

The network architecture is shown in Figure 2. The backbone is inspired by L2-Net [55]. Compared to L2-Net, we replace the last 8×8 convolutional layer by 3 2×2 convolutional layers, allowing to reduce the number of weights by a factor 5 for a similar or slightly better accuracy. The 128 dimensional output tensor serves as input to: (a) a ℓ_2-normalization layer to obtain descriptors $\boldsymbol{X}$, (b) an element-wise square operation followed by a 1×1 convolutional layer and a softmax to obtain the reliability confidence value $\boldsymbol{R}$ of each descriptor, and (c) the same operations to obtain the repeatability map $\boldsymbol{S}$. We now explain how we design the losses for training the network to obtain sparse, repeatable and reliable keypoints.

3.1. Learning repeatability

As observed in previous works [9, 59], keypoint repeatability is a problem that cannot be tackled by standard supervised training. In fact, using supervision essentially boils down in this case to copying an existing detector rather than discovering better and easier keypoints. We thus treat the repeatability as a self-supervised task and train the network such that the positions of local maxima in $\boldsymbol{S}$ are covariant to natural image transformations like viewpoint or illumination changes.

Let I and I' be two images of the same scene and let $U \in \mathbb{R}^{H \times W \times 2}$ be the ground-truth correspondences between them. In other words, if the pixel (i,j) in the first

image I corresponds to pixel (i', j') in the second image I', then $U_{ij} = (i', j')$. In practice, U can be estimated using existing optical flow or stereo matching algorithms if I and I' are natural images or can be obtained exactly if I' was generated synthetically with a known transformation, *e.g.* an homography [9], see Section 3.3. Let $\boldsymbol{S}$ and $\boldsymbol{S'}$ be the repeatability map for image I and I' respectively, and $\boldsymbol{S}'_U$ the heatmap from image I' warped according to U.

Ultimately, we want to enforce the fact that all local maxima in $\boldsymbol{S}$ correspond to the ones in $\boldsymbol{S}'_U$. Our key idea is to maximize the cosine similarity, denoted as $cosim$ in the following, between $\boldsymbol{S}$ and $\boldsymbol{S}'_U$. When $cosim(\boldsymbol{S}, \boldsymbol{S}'_U)$ is maximized, the two heatmaps are indeed identical and their maxima correspond exactly. However, this process assumes no occlusions, warp artifacts or border effects which strongly impacts performance in practice. We fix it by reformulating this idea *locally*, *i.e.*, averaging the cosine similarity over many small patches. We define the set of overlapping patches $\mathcal{P} = \{p\}$ that contains all $N \times N$ patches in $[1..W] \times [1..H]$ and define the loss as:

$$\mathcal{L}_{cosim}(I, I', U) = 1 - \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} cosim\left(\boldsymbol{S}\left[p\right], \boldsymbol{S}'_U\left[p\right]\right) \quad , \tag{1}$$

where $\boldsymbol{S}\left[p\right] \in \mathbb{R}^{N^2}$ denotes the vectorized (flattened) $N \times N$ patch p extracted from $\boldsymbol{S}$, and likewise for $\boldsymbol{S}'_U\left[p\right]$.

Note that $\mathcal{L}_{cosim}$ can be minimized trivially by having $\boldsymbol{S}$ and $\boldsymbol{S}'_U$ constant. To avoid that, we employ a second loss function that tries to maximize the local peakiness of the repeatability map:

$$\mathcal{L}_{peaky}(I) = 1 - \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \left(\max_{(i,j) \in p} \boldsymbol{S}_{ij} - \operatorname*{mean}_{(i,j) \in p} \boldsymbol{S}_{ij} \right) \quad . \tag{2}$$

Interestingly, this allows to choose the frequency of local maxima by varying the patch size N. Finally, the resulting repeatability loss is composed as a weighted sum of the first loss and second loss applied to both images as:

$$\begin{aligned} \mathcal{L}_{rep}(I, I', U) &= \mathcal{L}_{cosim}(I, I', U) \\ &+ \lambda \left(\mathcal{L}_{peaky}(I) + \mathcal{L}_{peaky}(I') \right) \quad . \end{aligned} \tag{3}$$

3.2. Learning reliability

To enforce reliability, our network not only computes the repeatability map $\boldsymbol{S}$ but jointly extracts dense local descriptors $\boldsymbol{X}$ and predicts for each descriptor $\boldsymbol{X}_{ij} \in \mathbb{R}^D$, a confidence value $\boldsymbol{R}_{ij} \in [0, 1]$ that estimates its reliability, *i.e.*, discriminativeness. The goal is to let the network learn to choose between making descriptors as discriminative as possible with a high confidence, or a low confidence in which case the loss will have low impact on the descriptor, such as for regions that cannot be made discriminative enough.

The descriptor matching problem can be seen as a ranking optimization problem, *i.e.*, given two images I and I', each descriptor from I is searched in I' as a query, ranking all descriptors from I' by increasing distance. Ranking losses have thus been extensively and successfully used to train local descriptors (*e.g.* triplet loss [6, 3, 21, 55, 29, 61]). At the exception of [19], only pairwise ranking losses such as the triplet loss have been used. These losses only perform local optimization, based on a pair, triplet, or quadruplet of training samples, which does not necessarily correlate well with a global metric like the Average Precision (AP). Recent work [19] suggested that directly optimizing the AP for patch descriptor matching significantly improves the performance. Inspired by recent advances in listwise losses [18, 58], He *et al*. [19] defined a differentiable approximation of the AP, a standard ranking metric, that can be directly optimized during training. Given a batch of ground-truth pairs of image patches, they use a convolutional neural network to compute their descriptors. They then compute the matrix of Euclidean distances between all patch descriptors from the batch. Each row in the matrix can be interpreted as the distances between a query patch from the first image and all patches from the second image, acting as database documents. Training thus consists in maximizing the AP computed for each query q in the batch B and averaged over the whole batch.

$$\mathcal{L}_{AP} = \frac{1}{B} \sum_{q=1}^{B} \mathcal{L}_{AP}(q), \quad \mathcal{L}_{AP}(q) = 1 - AP(q) \quad . \tag{4}$$

In this work, we follow a similar path. A major difference is that a standard keypoint detector is employed in [19] to extract patches, while our input is simply an image. The used L2-Net architecture [55] is applied patch by patch, which is quite slow. Applying this network in a fully-convolutional way is significantly more efficient. In our case, each pixel (i, j) from the first image defines a patch of size M that we can compare to all other patches in the second image. Knowing the ground-truth correspondence U, we can compute its AP, which is similar to the previous loss.

As a matter of fact, local descriptors can be extracted everywhere, but not all locations are equally interesting. In particular, uniform regions or elongated 1D patterns are known to lack the distinctiveness necessary for feature matching [15]. More interestingly, even well textured regions are also known to be unreliable from their semantic nature, such as tree leafages or ocean waves. It becomes thus clear that forcefully optimizing the patch descriptor even in meaningless regions of the image could hinder the training and runtime performance. We therefore propose a new loss to spare the network in wasting its efforts on undistinctive regions as:

$$\mathcal{L}_{AP\kappa}(i,j) = 1 - \left[AP(i,j)\boldsymbol{R}_{ij} + \kappa(1-\boldsymbol{R}_{ij})\right] \ , \quad (5)$$

where $\kappa \in [0,1]$ is a hyperparameter that indicates the minimum expected AP per patch. To minimize $\mathcal{L}_{AP\kappa}(i,j)$, the network should ideally predict $\boldsymbol{R}_{ij} = 0$ if $AP(i,j) < \kappa$ and $\boldsymbol{R}_{ij} = 1$ conversely. In practice, $\boldsymbol{R}_{ij}$ is between 0 and 1 and reflects the confidence of the network with respect to the reliability of patch i,j. We found that $\kappa = 0.5$ yields good results in practice. Note that a similar idea of jointly training the descriptor and an associated confidence was recently proposed in [33]. However, they used a triplet loss, not an AP loss, which prevents the use of an interpretable threshold κ as in our case.

Runtime. At test time, we run the trained network multiple times on the input image at different scales starting from the original scale, and downsampling by $2^{1/4}$ each time until the image is smaller than 128 pixels. For each scale, we find local maxima in $\boldsymbol{S}$ and gather descriptors from $\boldsymbol{X}$ at corresponding locations. Finally, we keep a shortlist of the best K descriptors over all scales where the descriptor score is computed as product $\boldsymbol{S}_{ij}\boldsymbol{R}_{ij}$, *i.e.*, requiring high values for both repeatability and reliability.

3.3. Training details

Training data. For training, the loss needs to be computed at potentially any image location as we do not know the salient regions in advance. To generate dense ground-truth matches, we consider two solutions: (a) using a pair of images where the second one is obtained by applying a known transformation to the first image (homographic transform, color jittering, etc.) [35]; (b) using a pair coming from an image sequence or a set of unordered images. In the latter case, in contrast to some previous work that focused on points verified by Structure-from-Motion (SfM), we designed a pipeline based on optical flow tools that can reliably extract dense correspondences given one image pair and a few sparse SfM-verified correspondences. As a first step, we run a SfM pipeline [49] that outputs a list of 3D points and 6D camera pose corresponding to each image. For each image pair with a sufficient overlap (*i.e.*, with some common 3D points), we then compute the fundamental matrix. We found that computing the fundamental matrix directly from the 2D SfM correspondences is more reliable than directly using the 6D camera pose. Next, we compute high-quality dense correspondences using EpicFlow [39]. We enhance the method by adding epipolar constraints in DeepMatching [40], the first step of EpicFlow that produces semi-sparse matches. In addition, we also predict a mask where the flow is reliable. Optical flow is by definition defined everywhere, even in occluded areas. However, we can obviously not train from these areas. We post-process the output of Deep Matching as follows: we compute a graph of connected consistent neighbors, and keep only matches belonging to large connected components (at least 50 matches). The mask is defined using a thresholded kernel density estimator on the verified matches. In practice, we use pairs of randomly transformed images from the distractors added recently to the Oxford and Paris retrieval dataset [37], that are basically images from the web. We also use pairs extracted (with the help of SfM) from the Aachen Day-Night dataset [44, 46] which contains images from the old inner city of Aachen, Germany.

Training sampling for AP loss. To have a setup as realistic as possible, given hardware constraints, we sub-sample query pixels (in the first image) on a regular grid of 8×8 pixels from cropped images of resolution 192×192. In the second image, we consider corresponding pixels of the queries as well as pixels sampled on a regular grid with a step of 8 pixels. To handle the inherent imperfection of flow and matches, we define the positives as the pixels within a radius of 4 pixels from the optical flow precision, and the negatives as all pixels at more than 8 pixels distance form this position.

Training parameters. We optimize the network using Adam with a batch size of 8, a learning rate of 0.001 and weight decay of 0.0005.

4. Experimental results

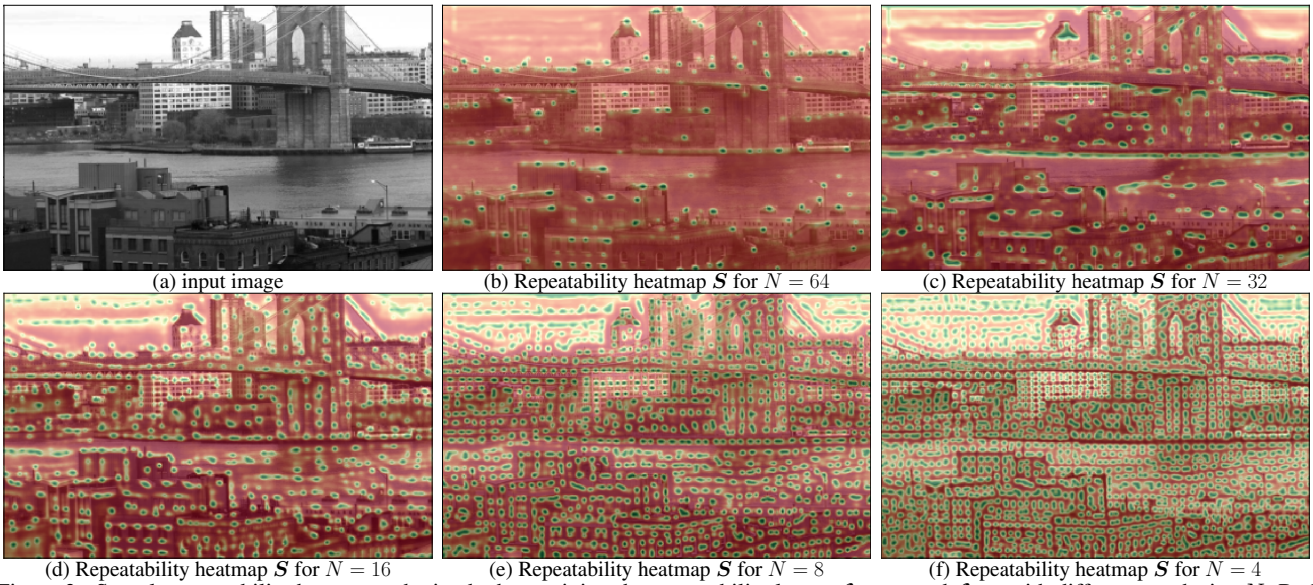
4.1. Dataset and metrics

We evaluate our method in the 116 full image sequences of the HPatches dataset [2]. The HPatches dataset contains 116 scenes where the first image is taken as a reference and subsequent images in a sequence are used to form pairs with increasing difficulty. This dataset can also be further separated into 57 sequences containing large changes in illumination and 59 with large changes in viewpoint.

Repeatability. Following [27], we compute the repeatability score for a pair of images as the number of point correspondences found between the two images divided by the minimum number of keypoint detections in the image pair. We report the average score over all image pairs.

Matching score (M-score). We follow the definitions given in [9, 59]. The matching score is the average ratio between ground-truth correspondences that can be recovered by the whole pipeline and the total number of estimated features within the shared viewpoint region when matching points from the first image to the second and the second image to the first one.

Mean Matching Accuracy (MMA). We use the same definition as in [11] where the MMA score is the average percentage of correct matches in an image pair considering multiple pixel error thresholds. We report the average score for each threshold over all image pairs.



(a) input image (b) Repeatability heatmap $\boldsymbol{S}$ for $N = 64$ (c) Repeatability heatmap $\boldsymbol{S}$ for $N = 32$

(d) Repeatability heatmap $\boldsymbol{S}$ for $N = 16$ (e) Repeatability heatmap $\boldsymbol{S}$ for $N = 8$ (f) Repeatability heatmap $\boldsymbol{S}$ for $N = 4$

Figure 3. Sample repeatability heatmaps obtained when training the repeatability losses $\mathcal{L}_{peaky}$ and $\mathcal{L}_{rep}$ with different patch size N. Red and green colors denote low and high values, respectively.

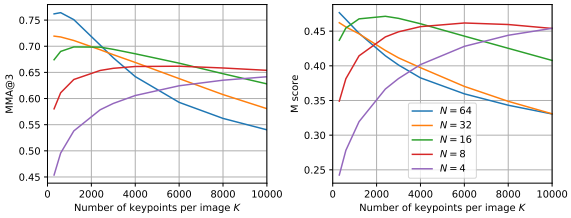


Figure 4. MMA@3 and M-score for different patch sizes N on the HPatches dataset, as a function of the number of retained keypoints K per image.

reliability	repeatability	M-score	MMA@3
✓		0.304	0.512
	✓	0.436	0.680
✓	✓	**0.461**	**0.686**

Table 1. Ablative study on HPatches. We report the M-score and the MMA at a 3px error threshold for our method (bottom row) as well as our approach without repeatability map (top row) or reliability map (middle row).

4.2. Impact of repeatability patch size

We first evaluate the impact of the patch size N used in the repeatability loss $\mathcal{L}_{rep}$, see Equation 3. The local patch size essentially controls the number of keypoints as the loss ideally encourages the network to output a single local maxima per window of size $N \times N$. We show in Figure 3 different repeatability maps obtained from the same input image for networks trained with different values of N. We observe that when N is large, the network outputs few highly-repeatable keypoints, and conversely for smaller values of N. Note that the networks even learn to populate empty regions like the sky with a grid-like pattern when N is small, while it avoids them when N is large.

We also plot the mean matching accuracy on the HPatches dataset in Figure 4 for various N as a function of the number of retained keypoints K per image. As expected, models trained with large N strongly outperforms models with lower N when the number of retained keypoints is low, since as seen above these keypoints have a higher quality. When keeping more keypoints, poor local maxima starts to get selected for these models (*e.g.* in the sky in Figure 3(b)) and the matching performance drops. However, having numerous keypoints is important for many applications such as visual localization because it augments the chance that at least a few of them will be correctly matched despite occlusions or other noise sources. There is therefore a trade-off between the number of keypoints and the matching performance. In the following experiments, and unless stated otherwise, we use a model trained with $N = 16$ and $K = 5000$ keypoints per image.

4.3. Impact of separate reliability and repeatability

Our main contribution is to show that repeatability and reliability can be predicted separately and help to jointly learn detector and descriptor. In Table 1, we report the performance when removing the repeatability $\boldsymbol{S}$, *i.e.*, keypoints are defined by maxima of the reliability map, or removing the reliability map $\boldsymbol{R}$, *i.e.*, learning the descriptor with the AP loss formulation of Equation 4 on all pixels. Without repeatability, the performance significantly drops both in terms of MMA@3 and M-score. This shows that repeatability is not well correlated with the descriptor reliability.

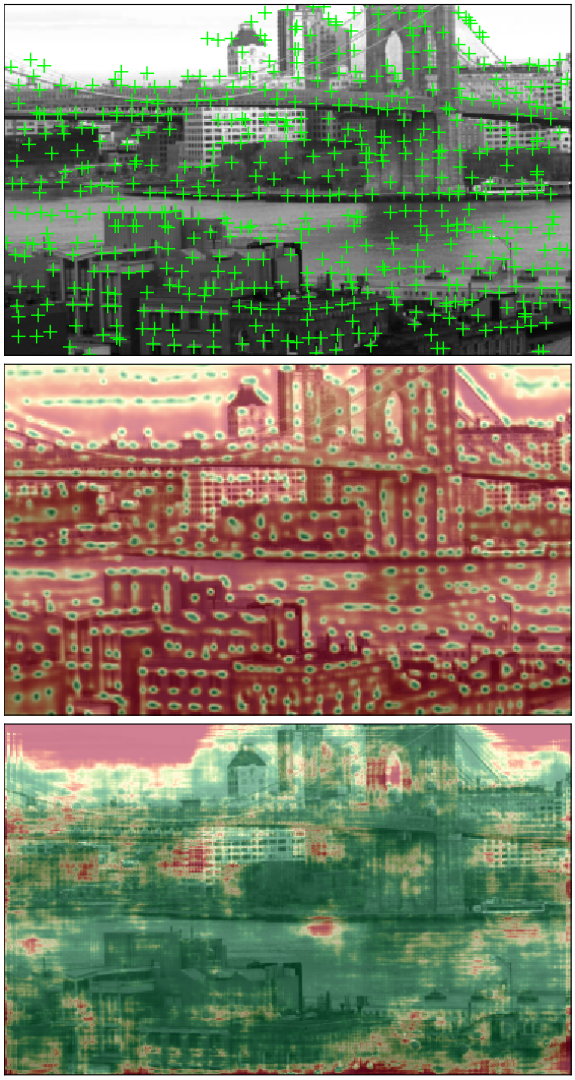


Figure 5. For one given input image (1st row), we show the repeatability (2nd row) and reliability heatmaps (3rd row) extracted at a single scale, overlaid onto the original image. Valid keypoints (both repeatable and reliable) are shown as crosses in the first image.

When training without estimating the descriptor reliability, the M-score decreases by 3% and the MMA@3 by 0.6%. This shows the importance of estimating the discriminativeness of descriptors. In Figure 5 we show the repeatability and reliability heatmaps obtained for the input image. Despite its small size, the network was able to learn that the sky region is a region that cannot be matches. More interestingly, more complex patterns are also rejected, such as 1-d patterns (under the bridge) or grid patterns (building windows). As a result, keypoints in those regions are scored low and are not retained in the top-K final output (see top row of Figure 5).

4.4. Comparison with the state of the art

We now compare our approach to state-of-the-art keypoint detectors and descriptors on the HPatches and Aachen datasets.

Detector repeatability. We first evaluate the keypoints extracted by our approach in term of repeatability. Following [48], we report the repeatability on the Oxford dataset [28], a subset of HPatches, for which the transformations applied to sequences is known and include jpeg compression (JPEG), blur (Blur), zoom and rotation (Z+R), luminosity (L), and viewpoint perspective (VP). Table 2 shows a comparison with QuadNet [48] and the handcrafted Difference of Gaussians (DoG) used in SIFT [25] on this dataset when varying the number of interest points. We observe that overall our approach significantly outperforms these two approaches, in particular for a high number of interest points. This demonstrates the excellent repeatability of our detector. Note that training on the Aachen dataset may obviously helps for street views. Nevertheless, our approach performs well even for the cases of blur or rotation (bark, boat), while we did not train the network for such challenging cases.

Mean Matching Accuracy. We next compare the mean matching accuracy with the state of the art, namely DELF [32], SuperPoint [9], LF-Net [35], multi-scale D2-Net [11], HardNet++ descriptors with HesAffNet regions [29, 30] (HAN + HN++) and a handcrafted Hessian affine detector with RootSIFT descriptor [36]. Figure 6 shows the results for illumination and viewpoint changes, as well as the overall performance on the HPatches dataset.

We observe that our method significantly outperforms the state of the art in particular for middle range thresholds. This is at the exception of DELF for illumination changes, which can be explained by the fact that they use a fixed grid of keypoints while this image subset has no spatial changes. Interestingly, our method significantly outperforms jointly detector and descriptor such as D2-Net [11], in particular at low level thresholds, which mean that our keypoints benefit from our joint training with repeatability and reliability.

Matching score. At 3px error threshold, we obtain a M-Score of 0.425 compared to 0.335 reported by LF-Net [35] and 0.288 for SIFT [25]. This demonstrates again the benefit of our matching approach with repeatability and reliability.

Qualitative results. We show in Figure 7 two examples of matching pair with a drastic change of viewpoint (left) and illumination change (right). We observe that our matches cover the entire image and most of them are correct (green dots).

4.5. Applications to visual localization

In this section, we evaluate our method for the task of visual localization [53, 45], where the goal is to estimate the camera position within a given environment using an

			Number of interest points				
Transformations	Data	Method	300	600	1200	2400	3000
Viewpoint Perspective (VP)	graf	DoG	0.21	0.0.2	0.18	-	-
		QuadNet	0.17	0.19	0.21	0.24	0.25
		Ours	**0.32**	**0.38**	**0.42**	**0.45**	**0.47**
	wall	DoG	0.27	0.28	0.28	-	-
		QuadNet	0.3	0.35	0.39	0.44	0.46
		Ours	**0.62**	**0.62**	**0.65**	**0.70**	**0.71**
Zoom and Rotation (Z+R)	bark	DoG	0.13	0.13	-	-	-
		QuadNet	0.12	0.13	0.14	0.16	0.16
		Ours	**0.27**	**0.33**	**0.37**	**0.44**	**0.47**
	boat	DoG	0.26	0.25	0.2	-	-
		QuadNet	0.21	0.24	0.28	0.28	0.29
		Ours	**0.33**	**0.39**	**0.45**	**0.54**	**0.57**
Luminosity (L)	leuven	DoG	0.51	0.51	0.5	-	-
		QuadNet	**0.7**	**0.72**	**0.75**	**0.76**	**0.77**
		Ours	0.65	0.69	0.73	**0.76**	**0.77**
Blur (B)	bikes	DoG	0.41	0.41	0.39	-	-
		QuadNet	0.53	0.53	0.49	0.55	0.57
		Ours	**0.66**	**0.67**	**0.71**	**0.75**	**0.76**
	trees	DoG	0.29	0.3	0.31	-	-
		QuadNet	**0.36**	**0.39**	0.44	0.49	0.5
		Ours	0.28	0.36	**0.45**	**0.55**	**0.6**
Compression (JPEG)	ubc	DoG	**0.68**	0.6	-	-	-
		QuadNet	0.55	**0.62**	**0.66**	**0.67**	**0.68**
		Ours	0.40	0.45	0.54	0.65	**0.68**

Table 2. Comparison with QuadNet [48] and a handcrafted difference of gaussian (DoG) in terms of detector repeatability on the Oxford dataset.

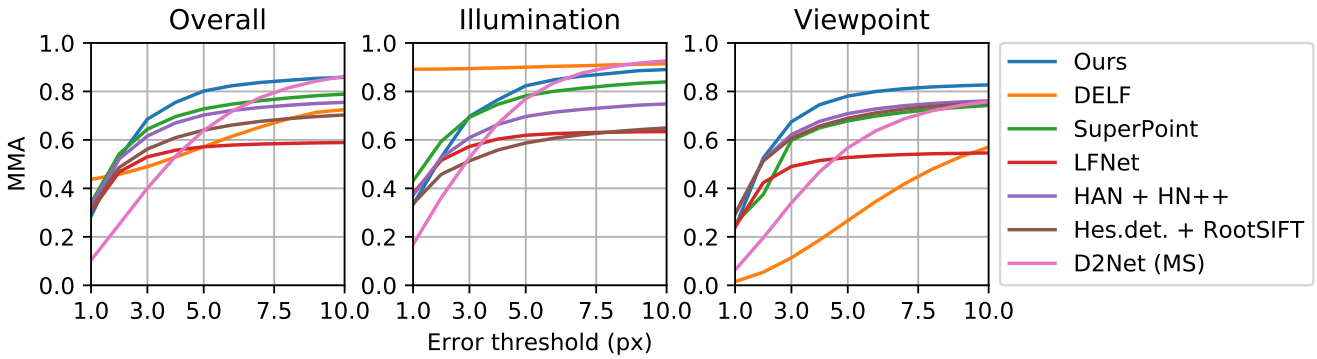


Figure 6. Comparison with the state of the art using the MMA for varying error thresholds on the HPatches dataset.

image. This is particularly interesting because robust local feature matching is crucial to enable visual localization in real-world conditions where it faces challenges such as day-night transitions and significant viewpoint changes between training and testing. First, we present a comparison with state of the art methods. Second, we present an ablative study in order to show the impact of training data.

Localization pipeline. The evaluation is done using *The Visual Localization Benchmark*[1], more specifically we use the local feature challenge of CVPR19. In order to evaluate a feature extraction method, a pre-defined visual localization pipeline[2] based on COLMAP [49] is used. First, the custom features (the ones to evaluate) are used to generate a

[1] https://www.visuallocalization.net

[2] https://github.com/tsattler/visuallocalizationbenchmark/tree/master/local_feature_evaluation

Figure 7. Sample results using reciprocal nearest matching. Correct and incorrect correspondences are shown as green dots and red crosses, respectively.

structure-from-motion model. Second, the test images are registered in this model again using the custom features. For feature matching, mutual nearest neighbor is used. Everything else follows COLMAP. The pipeline generates three result numbers representing the percentages of successfully localized images within three error tolerances $(0.5m, 2\,\text{deg})$ / $(1m, 5\,\text{deg})$ / $(5m, 10\,\text{deg})$, where the first number represents the max. position error in meters and the second number represents the max. orientation error in degrees. The dataset used is Aachen Day-Night [44, 46].

Impact of N and K. We have evaluated our approach in several configurations and report their performance in Table 3. Namely, we have evaluated our model trained with $N = 16$ for different numbers of keypoints K per image. For visual localization, it can be interesting to output more keypoints per image as it increases the chances of having at least few keypoints correctly matched despite occlusions or strong viewpoint/illumination changes, which in turn improves the localization accuracy. We therefore also evaluate our approach keeping $K = 10000$ keypoints per image, this time using $N = 8$ as we observed a higher MMA in this range (see Figure 4). For this latter model, we have also evaluated the impact of augmenting the size of the network by doubling the number of weights in the internal convolution layers. Our approach performs well in all configurations, including in the case with only $K = 2500$ keypoints per image. Quadrupling the number of keypoints leads in slightly higher localization accuracy. Augmenting the model size results in the best overall performance.

Comparison with the state of the art. Table 3 also provides a comparison to other methods submitted to the visual localization benchmark. Our augmented model for $K = 10000$ and $N = 8$ outperforms all competing methods by a significant margin at all error thresholds. The recent D2-Net approach [11] performs almost equally with only 1% less images localized within 0.5m. Interestingly, even our approach with $K = 5000$ performs better than most of the other methods, whereas it uses twice less keypoints per image. This demonstrates the high quality of our keypoint detection and scoring scheme. Indeed good keypoints for localization are ranked higher and even a shortlist with $K = 5000$ yield good results. In addition, we note that our local features have a relatively low dimensionality with respect to the features used in the other approaches (128 instead of 256, 512 or 1024 for others). Our network is also very compact as it contains only 1 million parameters, which is up to 15 times less than other competing learned methods. This shows the high efficiency or our joint detector and descriptor training based on direct AP minimization with separate repeatability and reliability.

Ablative study. Our network is trained from image pairs gathered from 4 different sources, see Section 3.3, in equal proportions: random web images (W), Aachen-day images (A), Aachen-night images obtained from automatic style transfer [24][3] (S). Image pairs are obtained synthetically from random homographies for W, A and S. Finally, real Aachen-day image pairs automatically annotated by computing the optical flow guided by the structure-from-motion model of the training images. We present in Table 4 the percentages of successfully localized images when training our networks on different subsets of the training data. Interestingly, our method performs well even for a network trained from only web images with homographies, significantly outperforming SIFT [25], SuperPoint [9] and the more recent HesAffNet [30]. Adding images from Aachen-day surprisingly does not result in any major change. This shows that our choice of a relatively small architecture prevents the network from overfitting. Synthetically generated night images enables a significant improvement for large error thresholds. In comparison, adding real image pairs annotated with optical flow enables a larger performance boost at all error thresholds. Finally, combining all 4 training sources yields to the best performance.

5. Conclusion

We proposed a new learning-based feature extraction method which jointly detects and describes keypoints in images. In contrast to traditional handcrafted features, our

[3]We used the code provided at `https://github.com/NVIDIA/FastPhotoStyle` specifically the version without semantic segmentation.

Method	#kpts	dim	#weights	0.5m, 2°		1m, 5°		5m, 10°	
RootSIFT[25]	11K	128	-	33.7	(-12)	52.0	(-14)	65.3	(-23)
HAN+HN[30]	11K	128	2M	37.8	(-8)	54.1	(-12)	75.5	(-13)
SuperPoint[9]	7K	256	1.3M	42.8	(-3)	57.1	(-9)	75.5	(-13)
DELF (new)[32]	11K	1024	9M	39.8	(-6)	61.2	(-5)	85.7	(-3)
D2-Net[11]	19K	512	15M	44.9	(-1)	66.3	(-0)	**88.8**	(-0)
R2D2, $N = 16$	2.5K	128	0.5M	43.9	(-2)	61.2	(-5)	84.7	(-4)
R2D2, $N = 16$	5K	128	0.5M	**45.9**	(-0)	65.3	(-1)	86.7	(-2)
R2D2, $N = 16$	10K	128	0.5M	44.9	(-1)	**67.3**	(+1)	87.8	(-1)
R2D2, $N = 8$	10K	128	0.5M	45.9	(-0)	63.3	(-3)	87.8	(-1)
R2D2, $N = 8$	10K	128	1.0M	**45.9**	-	66.3	-	**88.8**	-

Table 3. Comparison to the state of the art on the Aachen Day-Night dataset. We report the percentages of successfully localized images within 3 error thresholds (0.5m and 2°, 1m and 5°, 5m and 10°). The number in parenthesis indicates the performance difference compared to our best model in the last row.

W	A	S	F	0.5m, 2°		1m, 5°		5m, 10°	
✓				43.9	(-2)	61.2	(-4)	77.6	(-9)
✓	✓			42.9	(-3)	60.2	(-5)	78.6	(-8)
✓	✓	✓		42.9	(-3)	61.2	(-4)	84.7	(-2)
✓	✓		✓	43.9	(-2)	63.3	(-2)	**86.7**	(-0)
✓	✓	✓	✓	**45.9**	-	**65.3**	-	**86.7**	-

Table 4. Ablative study in terms of training data on the Aachen dataset. We report the percentages of successfully localized images within 3 error thresholds (0.5m and 2°, 1m and 5°, 5m and 10°). The number in parenthesis is the difference compared to the model trained on all data. All results are presented for $N = 16$ and $K = 5000$ keypoints per image. W=web images + homographies; A=Aachen-day images + homographies; S=Aachen-day-night from automatic style transfer + homographies; F=Aachen-day images pairs with optical flow.

method learns both keypoint repeatability and a confidence for keypoint reliability from relevant training data. Our network is trained with self-supervision using a mixture of synthetic (images with known transformations) and real data (point correspondences). Furthermore, we use style transfer methods to increase robustness against drastic illumination changes such as day-night transitions. Our experiments on the standard benchmark HPatches as well as for the task of visual localization show superior results of our approach in comparison to state-of-the-art methods.

References

[1] V. Balntas, E. Johns, L. Tang, and K. Mikolajczyk. Pn-net: Conjoined triple deep network for learning local image descriptors. *arXiv preprint arXiv:1601.05030*, 2016. 2

[2] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, 2017. 4.1

[3] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *BMVC*, 2016. 2, 3.2

[4] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006. 1

[5] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *ECCV*, 2010. 1

[6] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker. Universal correspondence network. In *NIPS*, 2016. 3.2

[7] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, 2004. 1

[8] G. Csurka, C. R. Dance, and M. Humenberger. From handcrafted to deep local invariant features. *arXiv preprint arXiv:1807.10254*, 2018. 2

[9] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR*, 2018. 1, 2, 3.1, 4.1, 4.4, 4.5, 4.5

[10] P. Di Febbo, C. Dal Mutto, K. Tieu, and S. Mattoccia. Kcnn: Extremely-efficient hardware keypoint detection with a compact convolutional neural network. In *CVPR*, 2018. 2

[11] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-net: A trainable cnn for joint detection and description of local features. In *CVPR*, 2019. 1, 2, 4.1, 4.4, 4.5, 4.5

[12] M. E. Fathy, Q.-H. Tran, M. Zeeshan Zia, P. Vernaza, and M. Chandraker. Hierarchical metric learning and matching for 2d and 3d geometric correspondences. In *ECCV*, 2018. 2

[13] S. Gauglitz, T. Höllerer, and M. Turk. Evaluation of interest point detectors and feature descriptors for visual tracking. *IJCV*, 2011. 2

[14] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. In *ECCV*, 2016. 1, 2

[15] K. Grauman and B. Leibe. Visual object recognition. *Synthesis lectures on artificial intelligence and machine learning*, 2011. 3.2

[16] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *CVPR*, 2015. 1, 2

[17] C. G. Harris, M. Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, 1988. 1

[18] K. He, F. Cakir, S. A. Bargal, and S. Sclaroff. Hashing as tie-aware learning to rank. In *CVPR*, 2018. 3.2

[19] K. He, Y. Lu, and S. Sclaroff. Local descriptors optimized for average precision. In *CVPR*, 2018. 1, 3.2, 3.2

[20] J. Heinly, J. L. Schonberger, E. Dunn, and J.-M. Frahm. Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In *CVPR*, 2015. 1

[21] B. Kumar, G. Carneiro, I. Reid, et al. Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In *CVPR*, 2016. 3.2

[22] A. B. Laguna, E. Riba, D. Ponsa, and K. Mikolajczyk. Key. net: Keypoint detection by handcrafted and learned cnn filters. *arXiv preprint arXiv:1904.00889*, 2019. 2

[23] S. Leutenegger, M. Chli, and R. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *ICCV*, 2011. 1

[24] Y. Li, M.-Y. Liu, X. Li, M.-H. Yang, and J. Kautz. A closed-form solution to photorealistic image stylization. In *ECCV*, 2018. 4.5

[25] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 4.4, 4.4, 4.5, 4.5

[26] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 2004. 1

[27] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *IJCV*, 2004. 1, 4.1

[28] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 2005. 1, 4.4

[29] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *NIPS*, 2017. 1, 2, 3.2, 4.4

[30] D. Mishkin, F. Radenovic, and J. Matas. Repeatability is not enough: Learning affine regions via discriminability. In *ECCV*, 2018. 2, 4.4, 4.5, 4.5

[31] J. Nath Kundu, R. MV, A. Ganeshan, and R. Venkatesh Babu. Object pose estimation from monocular image using multi-view keypoint correspondence. In *ECCV*, 2018. 1

[32] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large-scale image retrieval with attentive deep local features. In *CVPR*, 2017. 1, 2, 2, 4.4, 4.5

[33] D. Novotny, S. Albanie, D. Larlus, and A. Vedaldi. Self-supervised learning of geometrically stable features through probabilistic introspection. In *CVPR*, 2018. 3.2

[34] Y. Ono, E. Trulls, P. Fua, and K. M. Yi. Lf-net: Learning local features from images. In *NIPS*, 2018. 1

[35] Y. Ono, E. Trulls, P. Fua, and K. M. Yi. Lf-net: learning local features from images. In *NIPS*, 2018. 1, 2, 3.3, 4.4, 4.4

[36] M. Perd'och, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *CVPR*, 2009. 4.4

[37] F. Radenović, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *CVPR*, 2018. 3.3

[38] F. Radenović, G. Tolias, and O. Chum. CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples. In *ECCV*, 2016. 1, 2

[39] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015. 3.3

[40] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Deepmatching: Hierarchical deformable dense matching. *IJCV*, 2016. 3.3

[41] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *ICCV*, 2005. 2

[42] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, 2011. 1

[43] E. Salahat and M. Qasaimeh. Recent advances in features extraction and description algorithms: A comprehensive survey. In *ICIT*, 2017. 2

[44] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, and T. Pajdla. Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions. In *CVPR*, 2018. 3.3, 4.5

[45] T. Sattler, A. Torii, J. Sivic, M. Pollefeys, H. Taira, M. Okutomi, and T. Pajdla. Are large-scale 3d models really necessary for accurate visual localization? In *CVPR*, 2017. 1, 4.5

[46] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. Image Retrieval for Image-Based Localization Revisited. In *BMCV*, 2012. 3.3, 4.5

[47] N. Savinov, L. Ladicky, and M. Pollefeys. Matching neural paths: transfer from recognition to correspondence search. In *NIPS*, 2017. 2

[48] N. Savinov, A. Seki, L. Ladicky, T. Sattler, and M. Pollefeys. Quad-networks: unsupervised learning to rank for interest point detection. In *CVPR*, 2017. 1, 2, 4.4, 2

[49] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 3.3, 4.5

[50] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 1, 2

[51] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *ICCV*, 2015. 2

[52] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *IEEE Trans. on PAMI*, 2014. 2

[53] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson. City-scale localization for cameras with known vertical direction. *IEEE Trans. on PAMI*, 2016. 1, 4.5

[54] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *CVPR*, 2018. 2

[55] Y. Tian, B. Fan, and F. Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *CVPR*, 2017. 1, 2, 3, 3.2, 3.2

[56] T. Trzcinski, M. Christoudias, V. Lepetit, and P. Fua. Learning image descriptors with the boosting-trick. In *NIPS*, 2012. 1

[57] T. Tuytelaars, K. Mikolajczyk, et al. Local invariant feature detectors: a survey. *Foundations and trends® in computer graphics and vision*, 2008. 2

[58] E. Ustinova and V. Lempitsky. Learning deep embeddings with histogram loss. In *NIPS*, 2016. 3.2

[59] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. Lift: Learned invariant feature transform. In *ECCV*, 2016. 1, 2, 3.1, 4.1

[60] L. Zhang and S. Rusinkiewicz. Learning to detect features in texture images. In *CVPR*, 2018. 2

[61] X. Zhang, F. X. Yu, S. Kumar, and S.-F. Chang. Learning spread-out local feature descriptors. In *ICCV*, 2017. 3.2

[62] M. Zieba, P. Semberecki, T. El-Gaaly, and T. Trzcinski. Bingan: Learning compact binary descriptors with a regularized gan. In *NIPS*, 2018. 1