D2-Net: A Trainable CNN for *Joint Description and Detection* of Local Features

Mihai Dusmanu[1,2,3]   Ignacio Rocco[1,2]   Tomas Pajdla[4]   Marc Pollefeys[3,5]

Josef Sivic[1,2,4]   Akihiko Torii[6]   Torsten Sattler[7]

[1]DI, ENS   [2]Inria   [3]Department of Computer Science, ETH Zurich   [4]CIIRC, CTU in Prague
[5]Microsoft   [6]Tokyo Institute of Technology   [7]Chalmers University of Technology

Abstract

In this work we address the problem of finding reliable pixel-level correspondences under difficult imaging conditions. We propose an approach where a single convolutional neural network plays a dual role: It is simultaneously a dense feature descriptor and a feature detector. By postponing the detection to a later stage, the obtained keypoints are more stable than their traditional counterparts based on early detection of low-level structures. We show that this model can be trained using pixel correspondences extracted from readily available large-scale SfM reconstructions, without any further annotations. The proposed method obtains state-of-the-art performance on both the difficult Aachen Day-Night localization dataset and the InLoc indoor localization benchmark, as well as competitive performance on other benchmarks for image matching and 3D reconstruction.

1. Introduction

Establishing pixel-level correspondences between images is one of the fundamental computer vision problems, with applications in 3D computer vision, video compression, tracking, image retrieval, and visual localization.

Sparse local features [6–8, 13, 14, 19, 30, 32–34, 50, 55, 56, 60, 65] are a popular approach to correspondence estimation. These methods follow a *detect-then-describe* approach that first applies a feature detector [7,13,19,30,32,34,50,65] to identify a set of keypoints or interest points. The detector then provides image patches extracted around the keypoints to the following feature description stage [6–8, 14, 30, 33, 55, 56, 60, 65]. The output of this stage is a compact representation for each patch. Sparse local features offer a set of advantages: Correspondences can be *matched effi-*

---

[1]Département d'informatique de l'ENS, École normale supérieure, CNRS, PSL Research University, 75005

[4]Czech Institute of Informatics, Robotics, and Cybernetics, Czech Technical University in Prague

Figure 1: **Examples of matches obtained by the D2-Net method.** The proposed method can find image correspondences even under significant appearance differences caused by strong changes in illumination such as day-to-night, changes in depiction style or under image degradation caused by motion blur.

ciently via (approximate) nearest neighbor search [37] and the Euclidean distance. Sparse features offer a *memory efficient* representation and thus enable approaches such as Structure-from-Motion (SfM) [21,53] or visual localization [26, 47, 58] to scale. The keypoint detector typically considers low-level image information such as corners [19] or blob-like structures [30, 32]. As such, local features can

often be *accurately localized* in an image, which is an important property for 3D reconstruction [17, 53].

Sparse local features have been applied successfully under a wide range of imaging conditions. However, they typically perform poorly under extreme appearance changes, *e.g.*, between day and night [71] or seasons [46], or in weakly textured scenes [59]. Recent results indicate that a major reason for this observed drop in performance is the lack of repeatability in the keypoint detector: While local descriptors consider larger patches and potentially encode higher-level structures, the keypoint detector only considers small image regions. As a result, the detections are unstable under strong appearance changes. This is due to the fact that the low-level information used by the detectors is often significantly more affected by changes in low-level image statistics such as pixel intensities. Nevertheless, it has been observed that local descriptors can still be matched successfully even if keypoints cannot be detected reliably [46, 59, 62, 71]. Thus, approaches that forego the detection stage and instead densely extract descriptors perform much better in challenging conditions. Yet, this gain in robustness comes at the price of higher matching times and memory consumption.

In this paper, we aim at obtaining the best of both worlds, *i.e.*, a sparse set of features that are robust under challenging conditions and efficient to match and to store. To this end, we propose a *describe-and-detect* approach to sparse local feature detection and description: Rather than performing feature detection early on based on low-level information, we propose to postpone the detection stage. We first compute a set of feature maps via a Deep Convolutional Neural Network (CNN). These feature maps are then used to compute the descriptors (as slices through all maps at a specific pixel position) *and* to detect keypoints (as local maxima of the feature maps). As a result, the feature detector is tightly coupled with the feature descriptor. Detections thereby correspond to pixels with locally distinct descriptors that should be well-suited for matching. At the same time, using feature maps from deeper layers of a CNN enables us to base both feature detection and description on higher-level information [69]. Experiments show that our approach requires significantly less memory than dense methods. At the same time, it performs comparably well or even better under challenging conditions (*c.f.* Fig. 1) such as day-night illumination changes [46] and weakly textured scenes [59]. Our approach already achieves state-of-the-art performance without any training. It can be improved further by fine-tuning on a large dataset of landmark scenes [27].

Naturally, our approach has some drawbacks too: Compared to classical sparse features, our approach is less efficient due to the need to densely extract descriptors. Still, this stage can be done at a reasonable efficiency via a single forward pass through a CNN. Detection based on higher-
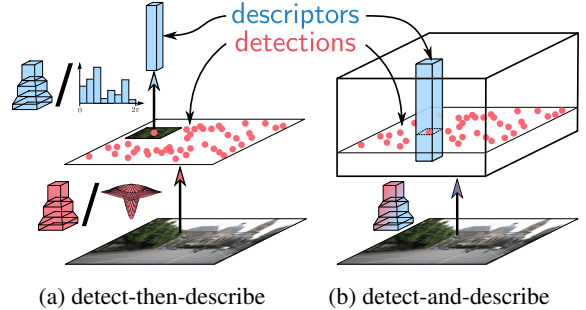

Figure 2: **Comparison between different approaches for feature detection and description.** Pipeline (a) corresponds to different variants of the two-stage detect-then-describe approach. In contrast, our proposed pipeline (b) uses a single CNN which extracts dense features that serve as both descriptors and detectors.

level information inherently leads to more robust but less accurate keypoints – yet, we show that our approach is still accurate enough for visual localization and SfM.

2. Related Work

Local features. The most common approach to sparse feature extraction – the detect-then-describe approach – first performs feature detection [7, 19, 30, 32, 34] and then extracts a feature descriptor [7, 9, 25, 30, 45] from a patch centered around each keypoint. The keypoint detector is typically responsible for providing robustness or invariance against effects such as scale, rotation, or viewpoint changes by normalizing the patch accordingly. However, some of these responsibilities might also be delegated to the descriptor [68]. Fig. 2a illustrates the common variations of this pipeline, from using hand-crafted detectors [7,19,30,32,34] and descriptors [7,9,25,30,45], replacing either the descriptor [6,55,56] or detector [50,70] with a learned alternative, or learning both the detector and descriptor [39,65]. For efficiency, the feature detector often considers only small image regions [65] and typically focuses on low-level structures such as corners [19] or blobs [30]. The descriptor then captures higher level information in a larger patch around the keypoint. In contrast, this paper proposes a *single branch describe-and-detect* approach to sparse feature extraction, as shown in Fig. 2b. As a result, our approach is able to detect keypoints belonging to higher-level structures and locally unique descriptors. The work closest to our approach is SuperPoint [13] as it also shares a deep representation between detection and description. However, they rely on different decoder branches which are trained independently with specific losses. On the contrary, our method shares all parameters between detection and description and uses a joint formulation that simultaneously optimizes for both tasks. Our experiments demonstrate that our describe-and-detect strategy performs significantly bet-

ter under challenging conditions, *e.g.*, when matching day-time and night-time images, than the previous approaches.

Dense descriptor extraction and matching. An alternative to the detect-then-describe approach is to forego the detection stage and perform the description stage densely across the whole image [10, 15, 49, 53]. In practice, this approach has shown to lead to better matching results than sparse feature matching [46, 59, 71], particularly under strong variations in illumination [71]. This identifies the detection stage as a significant weakness in detect-then-describe methods, which has motivated our approach.

Image retrieval. The task of image retrieval [3, 18, 38, 41, 61, 62] also deals with finding correspondences between images in challenging situations with strong illumination or viewpoint changes. Several of these methods start by dense descriptor extraction [3, 38, 61, 62] and later aggregate these descriptors into a compact image-level descriptor for retrieval. Works most related to our approach are [38, 61]: [38] develops an approach similar to ours, where an attention module is added on top of the dense description stage to perform keypoint selection. However, their method is designed to produce only a few reliable keypoints as to reduce the false positive matching rate during retrieval. Our experiments demonstrate that our approach performs significantly better for matching and camera localization; [61] implicitly detects a set of keypoints as the global maxima of all feature maps, before pooling this information into a global image descriptor. [61] has inspired us to detect features as local maxima of feature maps.

Object detection. The proposed describe-and-detect approach is also conceptually similar to modern approaches used in object detection [29, 42, 43]. These methods also start by a dense feature extraction step, which is followed by the scoring of a set of region proposals. A non-maximal-suppression stage is then performed to select only the most locally-salient proposals with respect to a classification score. Although these methods share conceptual similarities, they target a very different task and cannot be applied directly to obtain pixel-wise image correspondences.

This work builds on these previous ideas and proposes a method to perform joint detection and descriptions of keypoints, presented next.

3. Joint Detection and Description Pipeline

Contrary to the classical detect-then-describe approaches, which use a two-stage pipeline, we propose to perform dense feature extraction to obtain a representation that is simultaneously a detector and a descriptor. Because both detector *and* descriptor share the underlying representation, we refer to our approach as D2. Our approach is illustrated in Fig. 3.

The first step of the method is to apply a CNN $\mathcal{F}$ on the input image I to obtain a 3D tensor $F = \mathcal{F}(I), F \in \mathbb{R}^{h \times w \times n}$, where $h \times w$ is the spatial resolution of the feature maps and n the number of channels.

3.1. Feature Description

As in other previous work [38, 44, 59], the most straightforward interpretation of the 3D tensor F is as a dense set of descriptor vectors $\mathbf{d}$:

$$\mathbf{d}_{ij} = F_{ij:}, \mathbf{d} \in \mathbb{R}^n \ , \tag{1}$$

with $i = 1, \ldots, h$ and $j = 1, \ldots, w$. These descriptor vectors can be readily compared between images to establish correspondences using the Euclidean distance. During the training stage, these descriptors will be adjusted such that the same points in the scene produce similar descriptors, even when the images contain strong appearance changes. In practice, we apply an L2 normalization on the descriptors prior to comparing them: $\hat{\mathbf{d}}_{ij} = \mathbf{d}_{ij}/\|\mathbf{d}_{ij}\|_2$.

3.2. Feature Detection

A different interpretation of the 3D tensor F is as a collection of 2D responses D [61]:

$$D^k = F_{::k}, D^k \in \mathbb{R}^{h \times w} \ , \tag{2}$$

where $k = 1, \ldots, n$. In this interpretation, the feature extraction function $\mathcal{F}$ can be thought of as n different feature detector functions $\mathcal{D}^k$, each producing a 2D response map D^k. These detection response maps are analogous to the Difference-of-Gaussians (DoG) response maps obtained in Scale Invariant Feature Transform (SIFT) [30] or to the cornerness score maps obtained in Harris' corner detector [19]. In our work, these raw scores are post-processed to select only a subset of locations as the output keypoints. This process is described next.

Hard feature detection. In traditional feature detectors such as DoG, the detection map would be sparsified by performing a spatial non-local-maximum suppression. However, in our approach, contrary to traditional feature detectors, there exist multiple detection maps D^k ($k = 1, \ldots, n$), and a detection can take place on any of them. Therefore, for a point (i, j) to be detected, we require:

$$(i, j) \text{ is a detection} \iff D_{ij}^k \text{ is a local max. in } D^k \ ,$$
$$\text{with } k = \arg\max_t D_{ij}^t \ . \tag{3}$$

Intuitively, for each pixel (i, j), this corresponds to selecting the most preeminent detector $\mathcal{D}^k$ (channel selection), and then verifying whether there is a local-maximum at position (i, j) on that particular detector's response map D^k.
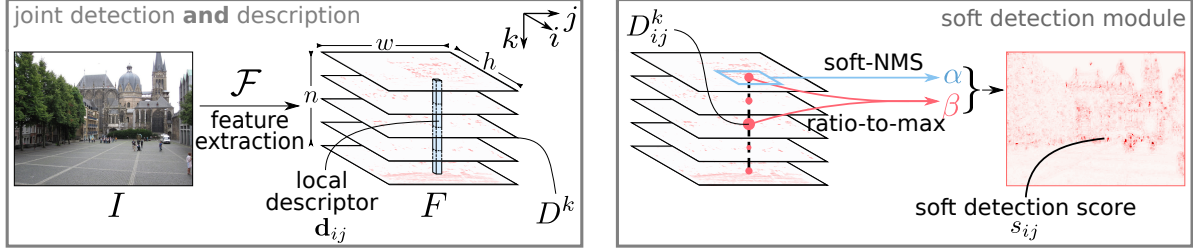
Figure 3: **Proposed detect-and-describe (D2) network.** A feature extraction CNN $\mathcal{F}$ is used to extract feature maps that play a dual role: (i) local descriptors $\mathbf{d}_{ij}$ are simply obtained by traversing all the n feature maps D^k at a spatial position (i, j); (ii) detections are obtained by performing a non-local-maximum suppression on a feature map followed by a non-maximum suppression across each descriptor - during training, keypoint detection scores s_{ij} are computed from a soft local-maximum score α and a ratio-to-maximum score per descriptor β.

Soft feature detection. During training, the hard detection procedure described above is softened to be amenable for back-propagation. First, we define a soft local-max. score

$$\alpha_{ij}^k = \frac{\exp\left(D_{ij}^k\right)}{\sum_{(i',j') \in \mathcal{N}(i,j)} \exp\left(D_{i'j'}^k\right)} \quad , \qquad (4)$$

where $\mathcal{N}(i, j)$ is the set of 9 neighbours of the pixel (i, j) (including itself). Then, we define the soft channel selection, which computes a ratio-to-max. per descriptor that emulates channel-wise non-maximum suppression:

$$\beta_{ij}^k = D_{ij}^k \Big/ \max_t D_{ij}^t \quad . \qquad (5)$$

Next, in order to take both criteria into account, we maximize the product of both scores across all feature maps k to obtain a single score map:

$$\gamma_{ij} = \max_k \left(\alpha_{ij}^k \beta_{ij}^k\right) \quad . \qquad (6)$$

Finally, the soft detection score s_{ij} at a pixel (i, j) is obtained by performing an image-level normalization:

$$s_{ij} = \gamma_{ij} \Big/ \sum_{(i',j')} \gamma_{i'j'} \quad . \qquad (7)$$

Multiscale Detection. Although CNN descriptors have a certain degree of scale invariance due to pre-training with data augmentations, they *are not* inherently invariant to scale changes and the matching tends to fail in cases with a significant difference in viewpoint.

In order to obtain features that are more robust to scale changes, we propose to use an image pyramid [2], as typically done in hand-crafted local feature detectors [28,30,32] or even for some object detectors [16]. This is only performed during *test time*.

Given the input image I, an image pyramid I^ρ containing three different resolutions $\rho = 0.5, 1, 2$ (corresponding to half resolution, input resolution, and double resolution) is constructed and used to extract feature maps F^ρ at each

resolution. Then, the larger image structures are propagated from the lower resolution feature maps to the higher resolution ones, in the following way:

$$\tilde{F}^\rho = F^\rho + \sum_{\gamma < \rho} F^\gamma \quad . \qquad (8)$$

Note that the feature maps F^ρ have different resolutions. To enable the summation in (8), feature maps F^γ are resized to the resolution of F^ρ using bilinear interpolation.

Detections are obtained by applying the post-processing described above to the fused feature maps $\tilde{F}^\rho$. In order to prevent re-detecting features, we use the following response gating mechanism: Starting at the coarsest scale, we mark the detected positions; these masks are upsampled (nearest neighbor) to the resolutions of the next scales; detections falling into marked regions are then ignored.

4. Jointly optimizing detection and description

This section describes the loss, the dataset used for training, and provides implementation details.

4.1. Training loss

In order to train the proposed model, which uses a single CNN $\mathcal{F}$ for both detection and description, we require an appropriate loss $\mathcal{L}$ that jointly optimizes the detection and description objectives. In the case of detection, we want keypoints to be *repeatable* under changes in viewpoint or illumination. In the case of description, we want descriptors to be *distinctive*, so that they are not mismatched. To this end, we propose an extension to the triplet margin ranking loss, which has been successfully used for descriptor learning [6, 35], to also account for the detection stage. We will first review the triplet margin ranking loss, and then present our extended version for joint detection and description.

Given a pair of images (I_1, I_2) and a correspondence $c : A \leftrightarrow B$ between them (where $A \in I_1$, $B \in I_2$), our version of the triplet margin ranking loss seeks to minimize the distance of the corresponding descriptors $\hat{\mathbf{d}}_A^{(1)}$ and $\hat{\mathbf{d}}_B^{(2)}$, while maximizing the distance to other confounding

descriptors $\hat{\mathbf{d}}^{(1)}_{N_1}$ or $\hat{\mathbf{d}}^{(2)}_{N_2}$ in either image, which might exist due to similarly looking image structures. To this end, we define the *positive* descriptor distance $p(c)$ between the corresponding descriptors as:

$$p(c) = \|\hat{\mathbf{d}}^{(1)}_A - \hat{\mathbf{d}}^{(2)}_B\|_2 \ , \tag{9}$$

The *negative* distance $n(c)$, which accounts for the most confounding descriptor for either $\hat{\mathbf{d}}^{(1)}_A$ or $\hat{\mathbf{d}}^{(2)}_B$, is defined as:

$$n(c) = \min\left(\|\hat{\mathbf{d}}^{(1)}_A - \hat{\mathbf{d}}^{(2)}_{N_2}\|_2, \|\hat{\mathbf{d}}^{(1)}_{N_1} - \hat{\mathbf{d}}^{(2)}_B\|_2\right) \ , \tag{10}$$

where the negative samples $\mathbf{d}^{(1)}_{N_1}$ and $\mathbf{d}^{(2)}_{N_2}$ are the hardest negatives that lie outside of a square local neighbourhood of the correct correspondence:

$$N_1 = \underset{P \in I_1}{\arg\min}\|\hat{\mathbf{d}}^{(1)}_P - \hat{\mathbf{d}}^{(2)}_B\|_2 \text{ s.t. } \|P - A\|_\infty > K \ , \tag{11}$$

and similarly for N_2. The triplet margin ranking loss for a margin M can be then defined as:

$$m(c) = \max\left(0, M + p(c)^2 - n(c)^2\right) \ . \tag{12}$$

Intuitively, this triplet margin ranking loss seeks to enforce the *distinctiveness* of descriptors by penalizing any confounding descriptor that would lead to a wrong match assignment. In order to additionally seek for the *repeatability* of detections, an detection term is added to the triplet margin ranking loss in the following way:

$$\mathcal{L}(I_1, I_2) = \sum_{c \in \mathcal{C}} \frac{s^{(1)}_c s^{(2)}_c}{\sum_{q \in \mathcal{C}} s^{(1)}_q s^{(2)}_q} m(p(c), n(c)) \ , \tag{13}$$

where $s^{(1)}_c$ and $s^{(2)}_c$ are the soft detection scores (7) at points A and B in I_1 and I_2, respectively, and $\mathcal{C}$ is the set of all correspondences between I_1 and I_2.

The proposed loss produces a weighted average of the margin terms m over all matches based on their detection scores. Thus, in order for the loss to be minimized, the most distinctive correspondences (with a lower margin term) will get higher relative scores and vice-versa - correspondences with higher relative scores are encouraged to have a similar descriptors distinctive from the rest.

4.2. Training Data

To generate training data on the level of pixel-wise correspondences, we used the MegaDepth dataset [27] consisting of 196 different scenes reconstructed from 1,070,468 internet photos using COLMAP [51, 54]. The authors provide camera intrinsics / extrinsics and depth maps from Multi-View Stereo for 102,681 images.

In order to extract the correspondences, we first considered all pairs of images with at least 50% overlap in the sparse SfM point cloud. For each pair, all points of the second image with depth information were projected into the first image. A depth-check with respect to the depth map of the first image was run to remove occluded pixels. In the end, we obtained 327,036 image pairs. This dataset was split in a validation dataset with 18,149 image pairs (from 78 scenes, each with less than 500 image pairs) and a training dataset from the remaining 118 scenes.

4.3. Implementation details

The VGG16 architecture [57], pretrained on ImageNet [12] and truncated after the `conv4_3` layer, was used to initialize the feature extraction network $\mathcal{F}$.

Training. The last layer of the dense feature extractor (`conv4_3`) was fine-tuned for 50 epochs using Adam [24] with an initial learning rate of 10^{-3}, which was further divided by 2 every 10 epochs. A fixed number (100) of random image pairs from each scene are used for training at every epoch in order to compensate the scene imbalance present in the dataset. For each pair, we selected a random 256×256 crop centered around one correspondence. We use a batch size of 1 and make sure that the training pairs present at least 128 correspondences in order to obtain meaningful gradients.

Testing. At test time, in order to increase the resolution of the feature maps, the last pooling layer (`pool3`) from $\mathcal{F}$ with a stride of 2 is replaced by an average pooling layer with a stride of 1. Then, the subsequent convolutional layers (`conv4_1` to `conv4_3`) are replaced with dilated convolutions [22] with a rate of 2, so that their receptive field remains unchanged. With these modifications, the obtained feature maps have a resolution of one fourth of the input resolution, which allows for more tentative keypoints and a better localization. The position of the detected keypoints is improved using a local refinement at feature map level following the approach used in SIFT [30]. The descriptors are then bilinearly interpolated at the refined positions.

Our implementation will be available at `https://github.com/mihaidusmanu/d2-net`.

5. Experimental Evaluation

The main motivation behind our work was to develop a local features approach that is able to better handle challenging conditions. Firstly, we evaluate our method on a standard image matching task based on sequences with illumination or viewpoint changes. Then, we present the results of our method in two more complex computer vision pipelines: 3D reconstruction and visual localization. In particular, the visual localization task is evaluated under extremely challenging conditions such as registering night-time images against 3D models generated from day-time

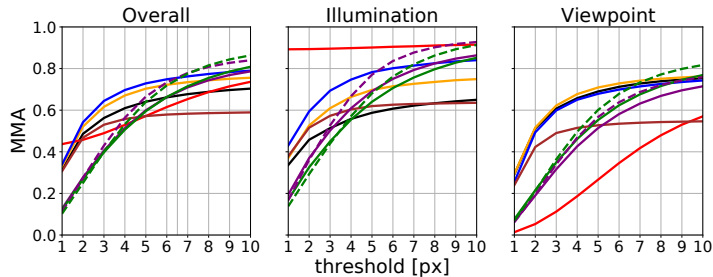| Method	# Features	# Matches
Hes. det. + RootSIFT	6.7K	2.8K
HAN + HN++ [35, 36]	3.9K	2.0K
LF-Net [39]	0.5K	0.2K
SuperPoint [13]	1.7K	0.9K
DELF [38]	4.6K	1.9K
D2 SS (ours)	3.0K	1.2K
D2 MS (ours)	4.9K	1.7K
D2 SS Trained (ours)	6.0K	2.5K
D2 MS Trained (ours)	8.3K	2.8K

Figure 4: **Evaluation on HPatches [5] image pairs.** For each method, the mean matching accuracy (MMA) as a function of the matching threshold (in pixels) is shown. We also report the mean number of detected features and the mean number of mutual nearest neighbor matches. Our approach achieves the best overall performance after a threshold of 6.5px, both using a single (SS) and multiple (MS) scales.

imagery [46, 48] and localizing images in challenging indoor scenes [59] dominated by weakly textured surfaces and repetitive structures. Qualitative examples of the results of our method are presented in Fig. 1. Please see the supplementary material for additional qualitative examples.

5.1. Image Matching

In a first experiment, we consider a standard image matching scenario where given two images we would like to extract and match features between them. For this experiment, we use the sequences of full images provided by the HPatches dataset [5]. Out of the 116 available sequences collected from various datasets [1, 5, 11, 23, 33, 63, 67], we selected 108.[1] Each sequence consists of 6 images of progressively larger illumination (52 sequences without viewpoint changes) or viewpoint changes (56 sequences without illumination changes). For each sequence, we match the first against all other images, resulting in 540 pairs.

Evaluation protocol. For each image pair, we match the features extracted by each method using nearest neighbor search, accepting only mutual nearest neighbors. A match is considered correct if its reprojection error, estimated using the homographies provided by the dataset, is below a given matching threshold. We vary the threshold and record the mean matching accuracy (MMA) [33] over all pairs, i.e., the average percentage of correct matches per image pair.

As baselines for the classical detect-then-describe strategy, we use RootSIFT [4, 30] with the Hessian Affine keypoint detector [32], a variant using a learned shape estimator (HesAffNet [36] - HAN) and descriptor (HardNet++ [35] - HN++[2]), and an end-to-end trainable variant (LF-Net [39]). We also compare against SuperPoint [13] and DELF [38], which are conceptually more similar to our approach.

Results. Fig. 4 shows results for illumination and viewpoint changes, as well as mean accuracy over both conditions. For each method, we also report the mean number of detected features and the mean number of mutual nearest

neighbor matches per image. As can be seen, our method achieves the best overall performance for matching thresholds of 6.5 pixels or more.

DELF does not refine its keypoint positions - thus, detecting the same pixel positions at feature map level yields perfect accuracy for strict thresholds. Even though powerful for the illumination sequences, the downsides of their method when used as a local feature extractor can be seen in the viewpoint sequences. For LF-Net, increasing the number of keypoints to more than the default value (500) worsened the results. However, [39] does not enforce that matches are mutual nearest neighbors and we suspect that their method is not suited for this type of matching.

As can be expected, our method performs worse than detect-then-describe approaches for stricter matching thresholds: The latter use detectors firing at low-level blob-like structures, which are inherently better localized than the higher-level features used by our approach. At the same time, our features are also detected at the lower resolution of the CNN features.

We suspect that the inferior performance for the sequences with viewpoint changes is due to a major bias in our training dataset - roughly 90% of image pairs have a change in viewpoint lower than $20°$ (measured as the angle between the principal axes of the two cameras).

The proposed pipeline for multiscale detection improves the viewpoint robustness of our descriptors, but it also adds more confounding descriptors that negatively affect the robustness to illumination changes.

5.2. 3D Reconstruction

In a second experiment, we evaluate the performance of our proposed describe-and-detect approach in the context of 3D reconstruction. This task requires well-localized features and might thus be challenging for our method.

For evaluation, we use three medium-scale internet-collected datasets with a significant number of different cameras and conditions (Madrid Metropolis, Gendarmenmarkt and Tower of London [64]) from a recent local feature evaluation benchmark [52]. All three datasets are small enough to allow exhaustive image matching, thus avoiding

---

[1]We left out sequences with an image resolution beyond 1200×1600 pixels as not all methods were able to handle this resolution.

[2]HardNet++ was trained on the HPatches dataset [5].

the need for using image retrieval.

Evaluation protocol. We follow the protocol defined by [52] and first run SfM [51], followed by Multi-View Stereo (MVS) [54]. For the SfM models, we report the number of images and 3D points, the mean track lengths of the 3D points, and the mean reprojection error. For the MVS models, we report the number of dense points. Except for the reprojection error, larger numbers are better. We use RootSIFT [4, 30] (the best perfoming method according to the benchmark's website) and GeoDesc [31], a state-of-the-art trained descriptor[3] as baselines. Both follow the detect-then-describe approach to local features.

Results. Tab. 1 shows the results of our experiment. Overall, the results show that our approach performs on par with state-of-the-art local features on this task. This shows that, even though our features are less accurately localized compared to detect-then-describe approaches, they are sufficiently accurate for the task of SfM as our approach is still able to register a comparable number of images.

Our method reconstructs fewer 3D points due to the strong ratio test filtering [30] of the matches that is performed in the 3D reconstruction pipeline. While this filtering is extremely important to remove incorrect matches and prevent incorrect registrations, we noticed that for our method it also removes an important number of correct matches (20%–25%)[4], as the loss used for training our method does not take this type of filtering into account.

5.3. Localization under Challenging Conditions

The previous experiments showed that our approach performs comparable with the state-of-the-art in standard applications. In a final experiment, we show that our approach sets the state-of-the-art for sparse features under two very challenging conditions: Localizing images under severe illumination changes and in complex indoor scenes.

Day-Night Visual Localization. We evaluate our approach on the Aachen Day-Night dataset [46, 48] in a local reconstruction task [46]: For each of the 98 night-time images contained in the dataset, up to 20 relevant day-time images with known camera poses are given. After exhaustive feature matching between the day-time images in each set, their known poses are used to triangulate the 3D structure of the scenes. Finally, these resulting 3D models are used to localize the night-time query images. This task was proposed in [46] to evaluate the perfomance of local features in the context of long-term localization without the need for a specific localization pipeline.

We use the code and evaluation protocol from [46] and report the percentage of night-time queries localized within

---

[3]In contrast to [31], we use the ratio test for matching with the threshold suggested by the authors - 0.89.

[4]Please see the supplementary material for additional details.

Dataset	Method	#Reg. Images	# Sparse. Points	Track Length	Reproj. Error	# Dense Points
Madrid Metropolis 1344 images	RootSIFT [4,30]	500	116K	6.32	**0.60px**	**1.82M**
	GeoDesc [31]	495	**144K**	5.97	0.65px	1.56M
	D2 MS (ours)	**501**	84K	6.33	1.28px	1.46M
	D2 MS trained (ours)	495	**144K**	**6.39**	1.35px	1.46M
Gendarmen-markt 1463 images	RootSIFT [4,30]	1035	338K	5.52	**0.69px**	**4.23M**
	GeoDesc [31]	1004	**441K**	5.14	0.73px	3.88M
	D2 MS (ours)	**1053**	250K	5.08	1.19px	3.49M
	D2 MS trained (ours)	965	310K	**5.55**	1.28px	3.15M
Tower of London 1576 images	RootSIFT [4,30]	**804**	239K	**7.76**	**0.61px**	**3.05M**
	GeoDesc [31]	776	**341K**	6.71	0.63px	2.73M
	D2 MS (ours)	785	180K	5.32	1.24px	2.73M
	D2 MS trained (ours)	708	287K	5.20	1.34px	2.86M

Table 1: **Evaluation on the Local Feature Evaluation Benchmark [52].** Each method is used for the 3D reconstruction of each scene and different statistics are reported. Overall, our method obtains a comparable performance with respect to SIFT and its trainable counterparts, despite using less well-localized keypoints.
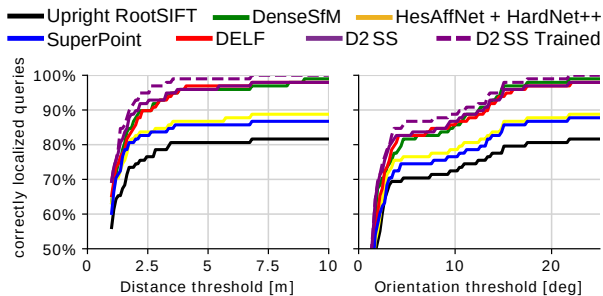
a given error bound on the estimated camera position and orientation. We compare against upright RootSIFT descriptors extracted from DoG keypoints [30], HardNet++ descriptors with HesAffNet features [35, 36], DELF [38], SuperPoint [13] and DenseSfM [46]. DenseSfM densely extracts CNN features using VGG16, followed by dense hierarchical matching (`conv4` then `conv3`).

For all methods with a threshold controlling the number of detected features (i.e. HAN + HN++, DELF, and SuperPoint), we employed the following tuning methodology: Starting from the default value, we increased and decreased the threshold gradually stopping as soon as the results started declining. Stricter localization thresholds were considered more important than looser ones. We reported the best results each method was able to achieve.

As can be seen from Fig. 5, our approach performs better than all baselines, especially for strict accuracy thresholds for the estimated pose. Our sparse feature approach even outperforms DenseSfM, even though the later is using significantly more features (and thus time and memory). The results clearly validate our describe-and-detect approach as it significantly outperforms detect-then-describe methods in this highly challenging scenario. The results also show that the lower keypoint accuracy of our approach does not prevent it from being used for applications aiming at estimating accurate camera poses.

Indoor Visual Localization. We also evaluate our approach on the InLoc dataset [59], a recently proposed benchmark dataset for large-scale indoor localization. The dataset is challenging due to its sheer size (∼10k database images covering two buildings), strong differences in viewpoint and / or illumination between the database and query images, and changes in the scene over time.

For this experiment, we integrated our features into two variants of the pipeline proposed in [59], using the code released by the authors. The first variant, **Direct Pose Es-**

Method	# Features	Correctly localized queries (%)			
		0.5m, 2°	1.0m, 5°	5.0m, 10°	10m, 25°
Upright RootSIFT [30]	11.3K	36.7	54.1	72.5	81.6
DenseSfM [46]	7.5K / 30K	39.8	60.2	84.7	99.0
HAN + HN++ [35, 36]	11.5K	39.8	61.2	77.6	88.8
SuperPoint [13]	6.6K	42.8	57.1	75.5	86.7
DELF [38]	11K	38.8	62.2	85.7	98.0
D2 SS (ours)	7K	41.8	66.3	85.7	98.0
D2 MS (ours)	11.4K	43.9	**67.3**	87.8	99.0
D2 SS Trained (ours)	14.5K	**44.9**	66.3	**88.8**	**100**
D2 MS Trained (ours)	19.3K	**44.9**	64.3	**88.8**	**100**

Figure 5: **Evaluation on the Aachen Day-Night dataset [46, 48].** We report the percentage of images registered within given error thresholds. Our approach improves upon state-of-the art methods by a significant margin under strict pose thresholds.

Method	Localized queries (%)		
	0.25m	0.5m	1.0m
Direct PE - Aff. RootSIFT [4, 30, 32]	18.5	26.4	30.4
Direct PE - D2 MS (ours)	**27.7**	**40.4**	**48.6**
Sparse PE - Aff. RootSIFT – 5MB	21.3	32.2	44.1
Sparse PE - D2 MS (ours) – 15MB	**35.0**	**48.6**	**62.6**
Dense PE [59] – 44MB	**35.0**	46.2	58.1
Sparse PE - Aff. RootSIFT + **Dense PV**	29.5	42.6	54.5
Sparse PE - D2 MS + **Dense PV** (ours)	38.0	54.1	65.4
Dense PE + **Dense PV** (= **InLoc**) [59]	41.0	56.5	69.9
InLoc + D2 MS (ours)	**43.2**	**61.1**	**74.2**

Table 2: **Evaluation on the InLoc dataset [59].** Our method outperforms SIFT by a large margin in both **Direct PE** and **Sparse PE** setups. It also outperforms the dense matching **Dense PE** method when used alone, while requiring less memory during pose estimation. By a combined approach of D2 and **InLoc** we obtained a new state-of-the art on this dataset.

timation (PE), matches features between the query image and the top-ranked database image found by image retrieval [3] and uses these matches for pose estimation. In the second variant, **Sparse PE**, the query is matched against the top-100 retrieved images, and a spatial verification [40] step is used to reject outliers matches. The query camera pose is then estimated using the database image with the largest number of verified matches.

Tab. 2 compares our approach with baselines from [59]: The original **Direct / Sparse PE** pipelines are based on affine covariant features with RootSIFT descriptors [4, 30, 32]. **Dense PE** matches densely extracted CNN descriptors between the images (using guided matching from the `conv5` to the `conv3` layer in a VGG16 network). As in [59], we report the percentage of query images localized within varying thresholds on their position error, considering only images with an orientation error of $10°$ or less. We also report the average memory usage of features per image. As can be seen, our approach outperforms both baselines.

In addition to **Dense PE**, the **InLoc** method proposed in [59] also verifies its estimated poses using dense information (**Dense Pose Verification (PV)**): A synthetic image is rendered from the estimated pose and then compared to the query image using densely extracted SIFT descriptors. A similarity score is computed based on this comparison and used to re-rank the top-10 images after **Dense PE**. Only this baseline outperforms our sparse feature approach, albeit at a higher computational cost. Combining our approach with **Dense PV** also improves performance, but not to the level of **InLoc**. This is not surprising, given that **InLoc** is able to leverage dense data. Still, our results show that sparse methods can perform close to this strong baseline.

Finally, by combining our method and **InLoc**, we were able to achieve a new state of the art — we employed a pose selection algorithm using the **Dense PV** scores for the top 10 images of each method. In the end, 182 **Dense PE** poses and 174 **Sparse PE** (using D2 MS) poses were selected.

6. Conclusions

We have proposed a novel approach to local feature extraction using a *describe-and-detect* methodology. The detection is not done on low-level image structures but postponed until more reliable information is available, and done jointly with the image description. We have shown that our method surpasses the state-of-the-art in camera localization under challenging conditions such as day-night changes and indoor scenes. Moreover, even though our features are less well-localized compared to classical feature detectors, they are also suitable for 3D reconstruction.

An obvious direction for future work is to increase the accuracy at which our keypoints are detected. This could for example be done by increasing the spatial resolution of the CNN feature maps or by regressing more accurate pixel positions. Integrating a ratio test-like objective into our loss could help to improve the performance of our approach in applications such as SfM.

Supplementary material

This supplementary material provides the following additional information: Section A details how we chose the threshold for Lowe's ratio test [30] used for the 3D reconstructions in Section 5.2 in the paper. As mentioned in Section 4.3 in the paper, Section B provides implementation details on the architecture. In addition, the section also evaluates another backbone architecture (ResNet [20]). Section C provides additional details on the loss function used to train our method. Section D shows qualitative examples for the matches found with our approach on the InLoc [59] and Aachen Day-Night [46, 48] datasets.

A. Impact of the ratio test on D2 features

Throughout our experiments on the local feature evaluation benchmark [52], we noticed that Lowe's ratio test [30] plays an important role because it significantly reduces the number of wrong registrations due to repetitive structures and semantically similar scenes.

In order to find an adequate ratio threshold for our features, we employ Lowe's methodology [30]: we compute the probability density functions (PDFs) of correct and incorrect matches with respect to the ratio test threshold. However, contrary to Lowe's evaluation, we considered only mutual nearest neighbors during the matching process.

Our evaluation was done on the entire HPatches [5] image pairs dataset consisting of 580 pairs from 116 sequences (57 with illumination changes and 59 with viewpoint changes). A match is considered correct if its projection error, estimated using the homographies provided by the dataset, is below 4 pixels - the default threshold in COLMAP [51, 54] during geometric verification and bundle adjustment. To take into account the possible errors in annotations and to have a clear separation between correct and incorrect matches, the threshold for incorrect matches is set to 20 pixels. Matches with projection errors between 4 and 20 pixels are therefore discarded during this evaluation.

Figure 6 shows the two PDFs. As can be seen, the D2 features do not work too well with ratio filtering because the mean ratio of correct matches is close to the one of incorrect matches. Still, we used thresholds of 0.90 for the off-the-shelf descriptors and 0.95 for the fine-tuned ones, which filter out 79.9% and 74.4% of incorrect matches, respectively. Unfortunately, these thresholds also discard a significant amount of correct matches (23.3% and 21.9%, respectively) which can have a negative impact on the number of registered images and sparse points.

In practice, we suggest not using the ratio test for camera localization under difficult conditions (e.g. day-night, indoors). For 3D reconstruction, using the threshold suggested above and / or increasing the minimum number of inlier matches required for an image pair to be considered
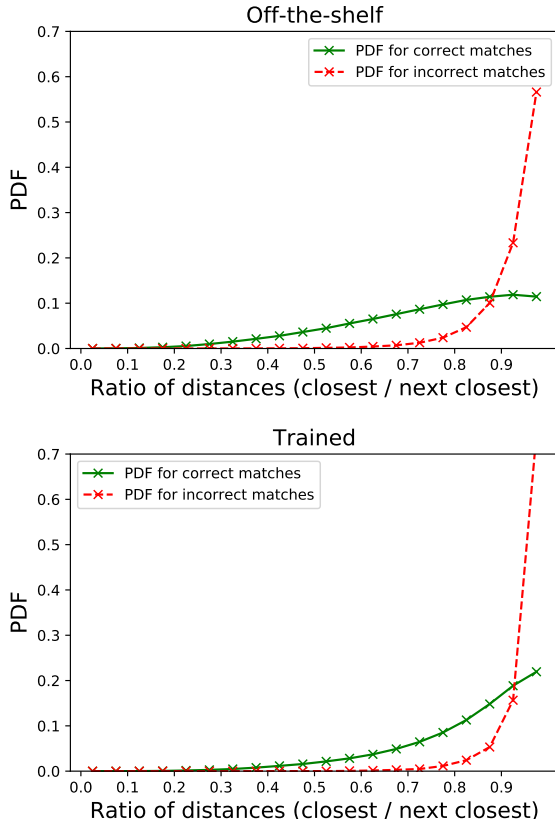

Figure 6: **Ratio PDFs for D2 multi-scale features.** PDF in terms of ratio on the full HPatches [5] image pairs dataset for the D2 off-the-shelf and fine-tuned features. There is no clear separation between the mean ratios of correct and incorrect matches as in the case of SIFT [30].

during Structure-from-Motion (SfM) should be sufficient to avoid most wrong registrations. Please note that, in the second case, the geometric verification can be significantly slower as RANSAC needs to handle a larger outlier ratio.

B. Details of the backbone architecture

For the feature extraction network $\mathcal{F}$, we used a VGG16 network pretrained on the ImageNet dataset [12], truncated after the `conv4_3` layer, as detailed in Section 4.3 of the paper. In addition, as also detailed in Section 4.3, we use a different image and feature resolution during training compared to testing. In particular, during testing, we take advantage of dilated convolutions [22, 66] to increase the resolution of the feature maps - this is not done in training due to memory limitations. More detailed descriptions of the network architectures during the training and testing phases are provided in Tables 3 and 4, respectively.

We additionally assess the choice of the network used for feature extraction, by performing a comparison between the chosen VGG16 [57] architecture and ResNet50 [20] (which

Layer	Stride	Dilation	ReLU	Resolution
input (256 × 256) - 3 ch.				×1
conv1_1 - 3 × 3, 64 ch.	1	1	✓	×1
conv1_2 - 3 × 3, 64 ch.	1	1	✓	×1
pool1 - 2 × 2, max.	2	1		×1/2
conv2_1 - 3 × 3, 128 ch.	1	1	✓	×1/2
conv2_2 - 3 × 3, 128 ch.	1	1	✓	×1/2
pool12 - 2 × 2, max.	2	1		×1/4
conv3_1 - 3 × 3, 256 ch.	1	1	✓	×1/4
conv3_2 - 3 × 3, 256 ch.	1	1	✓	×1/4
conv3_3 - 3 × 3, 256 ch.	1	1	✓	×1/4
pool3 - 2 × 2, max.	2	1		×1/8
conv4_1 - 3 × 3, 512 ch.	1	1	✓	×1/8
conv4_2 - 3 × 3, 512 ch.	1	1	✓	×1/8
conv4_3 - 3 × 3, 512 ch.	1	1		×1/8

Table 3: **Training architecture.** During training, we use the default VGG16 [57] architecture up to conv4_3, and fine-tune the last layer (conv4_3).

Layer	Stride	Dilation	ReLU	Resolution
input (∼ 1200 × 1600) - 3 ch.				×1
conv1_1 - 3 × 3, 64 ch.	1	1	✓	×1
conv1_2 - 3 × 3, 64 ch.	1	1	✓	×1
pool1 - 2 × 2, max.	2	1		×1/2
conv2_1 - 3 × 3, 128 ch.	1	1	✓	×1/2
conv2_2 - 3 × 3, 128 ch.	1	1	✓	×1/2
pool12 - 2 × 2, max.	2	1		×1/4
conv3_1 - 3 × 3, 256 ch.	1	1	✓	×1/4
conv3_2 - 3 × 3, 256 ch.	1	1	✓	×1/4
conv3_3 - 3 × 3, 256 ch.	1	1	✓	×1/4
pool3 - 2 × 2, avg.	1	1		×1/4
conv4_1 - 3 × 3, 512 ch.	1	2	✓	×1/4
conv4_2 - 3 × 3, 512 ch.	1	2	✓	×1/4
conv4_3 - 3 × 3, 512 ch.	1	2	[1]	×1/4

Table 4: **Testing architecture.** At test time, we slightly modify the training architecture: the last pooling layer pool3 is replaced by an average pooling with a stride of 1 and the following convolutional layers are dilated by a factor of 2. This maintains the same receptive field but offers higher resolution feature maps.

is the state of the art backbone architecture used in various other works). We evaluate them on the HPatches image pairs dataset using the same evaluation protocol that is described in Section 5.1 of the main paper.

For both architectures, we used weights trained on ImageNet [12]. In the case of ResNet50, the network was truncated after conv4_6 (following the approach in DELF [38]). At this point in the architecture, the resolution is $1/16^{\text{th}}$ of the input resolution and the descriptors are 1024-dimensional. However, in the case of the original VGG16, the output after conv4_3 has $1/8^{\text{th}}$ of the input resolution and 512 channels. In order to account for this difference in resolution, we use dilated convolutions (also sometimes referred to as "atrous convolutions") to increase the resolution for the ResNet50 network. In addition, dilated convolutions are applied to both networks to further increase the feature resolution to $1/4^{\text{th}}$ of the input resolution. For simplicity, only single-scale features are considered in this comparison.

The results can be seen in Figure 7. Dilated convolutions [22,66] increase the number of detections and the performance of D2 features especially in the case of viewpoint changes. The ResNet50 features also benefit from dilated convolutions and the increase in the resolution. However, although ResNet50 features seem slightly more robust to illumination changes and are able to outperform VGG16 features for thresholds larger than 6.5 pixels, they are less robust to viewpoint changes. Overall, ResNet50 features obtain worse results in this evaluation which motivated our decision to use VGG16.

C. Details of the training loss

This section gives more insight into the loss $\mathcal{L}$ that we used for fine-tuning the conv4_3 layer of the VGG16 network. In particular, in Figure 8 we explain in more detail the in-image-pair negative mining expressed in Equations (10) and (11) of the paper.

The parameter K controls the size of the neighbourhood from where negative samples are *not* selected. For a value of $K = 0$, all feature map pixels apart from the considered correspondence $c : A \leftrightarrow B$ are considered as possible negatives. In this case, a value of the margin loss $m(c)$ lower than M ($p(c) < n(c)$) signifies that A and B would be matched using mutual nearest neighbors. This is due to the symmetric negative selection. However, in practice, this is too restrictive since adjacent pixels have a significant overlap in their receptive field so the descriptors can be very close. Since the receptive field at the conv4_3 level is around 65×65 pixels at the input resolution, we choose a value of $K = 4$ at the feature map level, which enforces that potential negatives have less than 50% spatial overlap.

Another parameter of the training loss is the margin M. Since the descriptors are L2 normalized, the squared distance between two descriptors is guaranteed to be lower than 4. We have settled for $M = 1$ as in previous work [35]. It is worth noting that, due to the the negative mining scheme, this margin is rarely reached, *i.e.*, the detection scores continue to be optimized.

Figure 9 shows the soft detection scores before and after fine-tuning. As expected, some salient points have increased scores, while repetitive structures are weighted down. Even though most of our training data is from outdoors scenes, these observations seem to translate well to indoors images too.

---

[1]We noticed that ReLU has a significant negative impact on the off-the-shelf descriptors, but not on the fine-tuned ones. Thus, we report results without ReLU for the off-the-shelf model and with ReLU for the fine-tuned one.
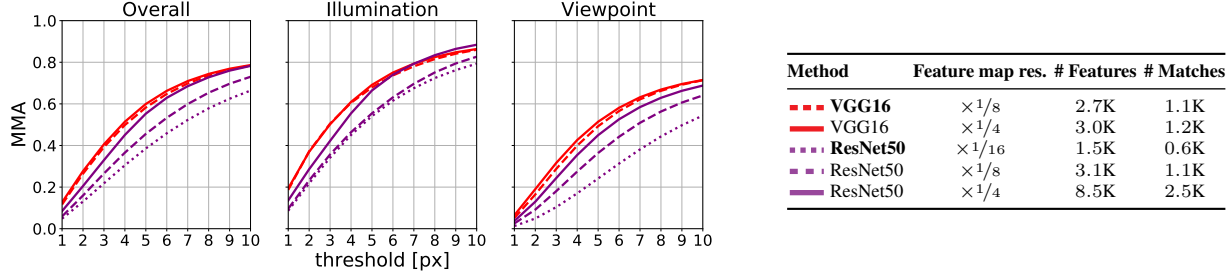
Figure 7: **Evaluation of different backbone architectures on the HPatches image pairs.** The original networks are in **bold** - the others were obtained by removing the stride of the deepest layers and adding dilations to the subsequent ones. Dilated convolutions offer more keypoints and better performance in viewpoint sequences. VGG16 outperforms ResNet50 by a significant margin even at a similar feature map resolution.

Method	Feature map res.	# Features	# Matches
VGG16	$\times 1/8$	2.7K	1.1K
VGG16	$\times 1/4$	3.0K	1.2K
ResNet50	$\times 1/16$	1.5K	0.6K
ResNet50	$\times 1/8$	3.1K	1.1K
ResNet50	$\times 1/4$	8.5K	2.5K

$$N_1 = \arg\min_{P \in I_1} \|d_P^{(1)} - d_B^{(2)}\|_2 \text{ s.t. } \|P - A\|_\infty > K$$
$$N_2 = \arg\min_{P \in I_2} \|d_A^{(1)} - d_P^{(2)}\|_2 \text{ s.t. } \|P - B\|_\infty > K$$
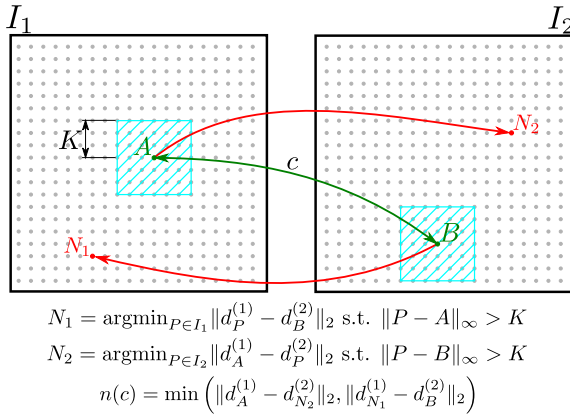$$n(c) = \min\left(\|d_A^{(1)} - d_{N_2}^{(2)}\|_2, \|d_{N_1}^{(1)} - d_B^{(2)}\|_2\right)$$

Figure 8: **In-image-pair negative mining procedure.** For each correspondence $c : A \leftrightarrow B$, the negative sample is chosen between the hardest negative of A in I_2 (N_2) or of B in I_1 (N_1). Since adjacent pixels at feature map level have overlapping receptive fields in the input image, the negative descriptor is chosen to be at least K pixels away from the ground-truth correspondence.

D. Qualitative examples

Figures 10 and 11 show examples from the InLoc [59] dataset: firstly, we show a few good matches in challenging conditions (significant viewpoint changes and texture-less areas) and then we illustrate the main failure modes of D2 features on indoors scenes (repeated objects / patterns). Figure 12 shows some example matches on the difficult scenes from the Aachen Day-Night [46, 48] camera localization challenge.

References

[1] Henrik Aanæs, Anders Lindbjerg Dahl, and Kim Steenstrup Pedersen. Interesting interest points. *IJCV*, 97(1):18–35, 2012. 6

[2] Edward H. Adelson, Charles H. Anderson, James R. Bergen, Peter J. Burt, and Joan M. Ogden. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41, 1984. 4

[3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proc. CVPR*, 2016. 3, 8

[4] Relja Arandjelovic and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *Proc. CVPR*, 2012. 6, 7, 8

[5] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proc. CVPR*, 2017. 6, 9

[6] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Proc. BMVC.*, 2016. 1, 2, 4

[7] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded Up Robust Features. In *Proc. ECCV*, 2006. 1, 2

[8] Matthew Brown, Gang Hua, and Simon Winder. Discriminative Learning of Local Image Descriptors. *IEEE PAMI*, 33(1):43–57, 2011. 1

[9] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary robust independent elementary features. In *Proc. ECCV*, 2010. 2

[10] Christopher B. Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal Correspondence Network. In *NIPS*, 2016. 3

[11] Kai Cordes, Bodo Rosenhahn, and Jörn Ostermann. Increasing the accuracy of feature evaluation benchmarks using differential evolution. In *IEEE Symposium on Differential Evolution (SDE)*, 2011. 6

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009. 5, 9, 10

[13] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *CVPR Workshops*, 2018. 1, 2, 6, 7, 8

[14] Jingming Dong and Stefano Soatto. Domain-size pooling in local descriptors: DSP-SIFT. In *Proc. CVPR*, 2015. 1

[15] Mohammed E. Fathy, Quoc-Huy Tran, M. Zeeshan Zia, Paul Vernaza, and Manmohan Chandraker. Hierarchical Metric Learning and Matching for 2D and 3D Geometric Correspondences. In *Proc. ECCV*, 2018. 3

[16] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively

trained part-based models. *IEEE PAMI*, 32(9):1627–1645, 2010. 4

[17] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE PAMI*, 32(8):1362–1376, 2010. 2

[18] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. End-to-End Learning of Deep Visual Representations for Image Retrieval. *IJCV*, 124(2):237–254, 2017. 3

[19] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference*, 1988. 1, 2, 3

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 9

[21] Jared Heinly, Johannes L. Schönberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the World* in Six Days *(As Captured by the Yahoo 100 Million Image Dataset). In *Proc. CVPR*, 2015. 1

[22] Matthias Holschneider, Richard Kronland-Martinet, Jean Morlet, and Ph Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets, Time Frequency Methods and Phase Space*, pages 286–297. 1990. 5, 9, 10

[23] Nathan Jacobs, Nathaniel Roman, and Robert Pless. Consistent temporal variations in many outdoor scenes. In *Proc. CVPR*, 2007. 6

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2015. 5

[25] Stefan Leutenegger, Margarita Chli, and Roland Siegwart. BRISK: Binary robust invariant scalable keypoints. In *Proc. ICCV*, 2011. 2

[26] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide Pose Estimation using 3D Point Clouds. In *Proc. ECCV*, 2012. 1

[27] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. In *Proc. CVPR*, 2018. 2, 5

[28] Tony Lindeberg. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2):225–270, 1994. 4

[29] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In *Proc. ECCV*, 2016. 3

[30] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1, 2, 3, 4, 5, 6, 7, 8, 9

[31] Zixin Luo, Tianwei Shen, Lei Zhou, Siyu Zhu, Runze Zhang, Yao Yao, Tian Fang, and Long Quan. GeoDesc: Learning local descriptors by integrating geometry constraints. In *Proc. ECCV*, 2018. 7

[32] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004. 1, 2, 4, 6, 8

[33] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE PAMI*, 27(10):1615–1630, 2005. 1, 6

[34] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *IJCV*, 65(1):43–72, 2005. 1, 2

[35] Anastasiya Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *NIPS*, 2017. 4, 6, 7, 8, 10

[36] Dmytro Mishkin, Filip Radenović, and Jiři Matas. Repeatability Is Not Enough: Learning Discriminative Affine Regions via Discriminability. In *Proc. ECCV*, 2018. 6, 7, 8

[37] Marius Muja and David G. Lowe. Scalable Nearest Neighbor Algorithms for High Dimensional Data. *IEEE PAMI*, 36:2227–2240, 2014. 1

[38] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Largescale image retrieval with attentive deep local features. In *Proc. ICCV*, 2017. 3, 6, 7, 8, 10, 16

[39] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net: Learning local features from images. In *Advances in Neural Information Processing Systems*, 2019. 2, 6

[40] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. CVPR*, 2007. 8

[41] Filip Radenović, Giorgos Tolias, and Ondrej Chum. Fine-tuning CNN Image Retrieval with No Human Annotation. *IEEE PAMI*, 2018. 3

[42] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, real-time object detection. In *Proc. CVPR*, 2016. 3

[43] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 3

[44] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *Proc. CVPR*, 2017. 3

[45] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proc. ICCV*, 2011. 2

[46] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6DoF outdoor visual localization in changing conditions. In *Proc. CVPR*, 2018. 2, 3, 5, 7, 8, 9, 11, 16

[47] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization? In *Proc. CVPR*, 2017. 1

[48] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *Proc. BMVC.*, 2012. 5, 7, 8, 9, 11, 16

[49] Nikolay Savinov, Lubor Ladicky, and Marc Pollefeys. Matching neural paths: transfer from recognition to correspondence search. In *NIPS*, 2017. 3

[50] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-networks: unsupervised learning to rank for interest point detection. In *Proc. CVPR*, 2017. 1, 2

[51] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proc. CVPR*, 2016. 5, 7, 9

[52] Johannes L. Schönberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *Proc. CVPR*, 2017. 6, 7, 9

[53] Johannes L. Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic Visual Localization. In *Proc. CVPR*, 2018. 1, 2, 3

[54] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *Proc. ECCV*, 2016. 5, 7, 9

[55] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative Learning of Deep Convolutional Feature Point Descriptors. In *Proc. ICCV*, 2015. 1, 2

[56] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Learning Local Feature Descriptors Using Convex Optimisation. *IEEE PAMI*, 36(8):1573–1585, 2014. 1, 2

[57] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015. 5, 9, 10

[58] Linus Svärm, Olof Enqvist, Fredrik Kahl, and Magnus Oskarsson. City-Scale Localization for Cameras with Known Vertical Direction. *IEEE PAMI*, 39(7):1455–1461, 2017. 1

[59] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor visual localization with dense matching and view synthesis. In *Proc. CVPR*, 2018. 2, 3, 6, 7, 8, 9, 11, 15

[60] Engin Tola, Vincent Lepetit, and Pascal Fua. DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo. *IEEE PAMI*, 32(5):815–830, 2010. 1

[61] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. In *Proc. ICLR*, 2016. 3

[62] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 Place Recognition by View Synthesis. In *Proc. CVPR*, 2015. 2, 3

[63] Vassilios Vonikakis, Dimitris Chrysostomou, Rigas Kouskouridas, and Antonios Gasteratos. Improving the robustness in feature detection by local contrast enhancement. In *IEEE International Conference on Imaging Systems and Techniques (IST)*, 2012. 6

[64] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *Proc. ECCV*, 2014. 6

[65] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned invariant feature transform. In *Proc. ECCV*, 2016. 1, 2

[66] Fisher Yu and Vladlen Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. In *ICLR*, 2016. 9, 10

[67] Guoshen Yu and Jean-Michel Morel. ASIFT: An algorithm for fully affine invariant comparison. *Image Processing On Line*, 1:11–38, 2011. 6

[68] Amir R. Zamir, Tilman Wekel, Pulkit Agrawal, Colin Wei, Jitendra Malik, and Silvio Savarese. Generic 3D Representation via Pose Estimation and Matching. In *Proc. ECCV*, 2016. 2

[69] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proc. ECCV*, 2014. 2

[70] Linguang Zhang and Szymon Rusinkiewicz. Learning to Detect Features in Texture Images. In *Proc. CVPR*, 2018. 2

[71] Hao Zhou, Torsten Sattler, and David W. Jacobs. Evaluating Local Features for Day-Night Matching. In *ECCV Workshops*, 2016. 2, 3
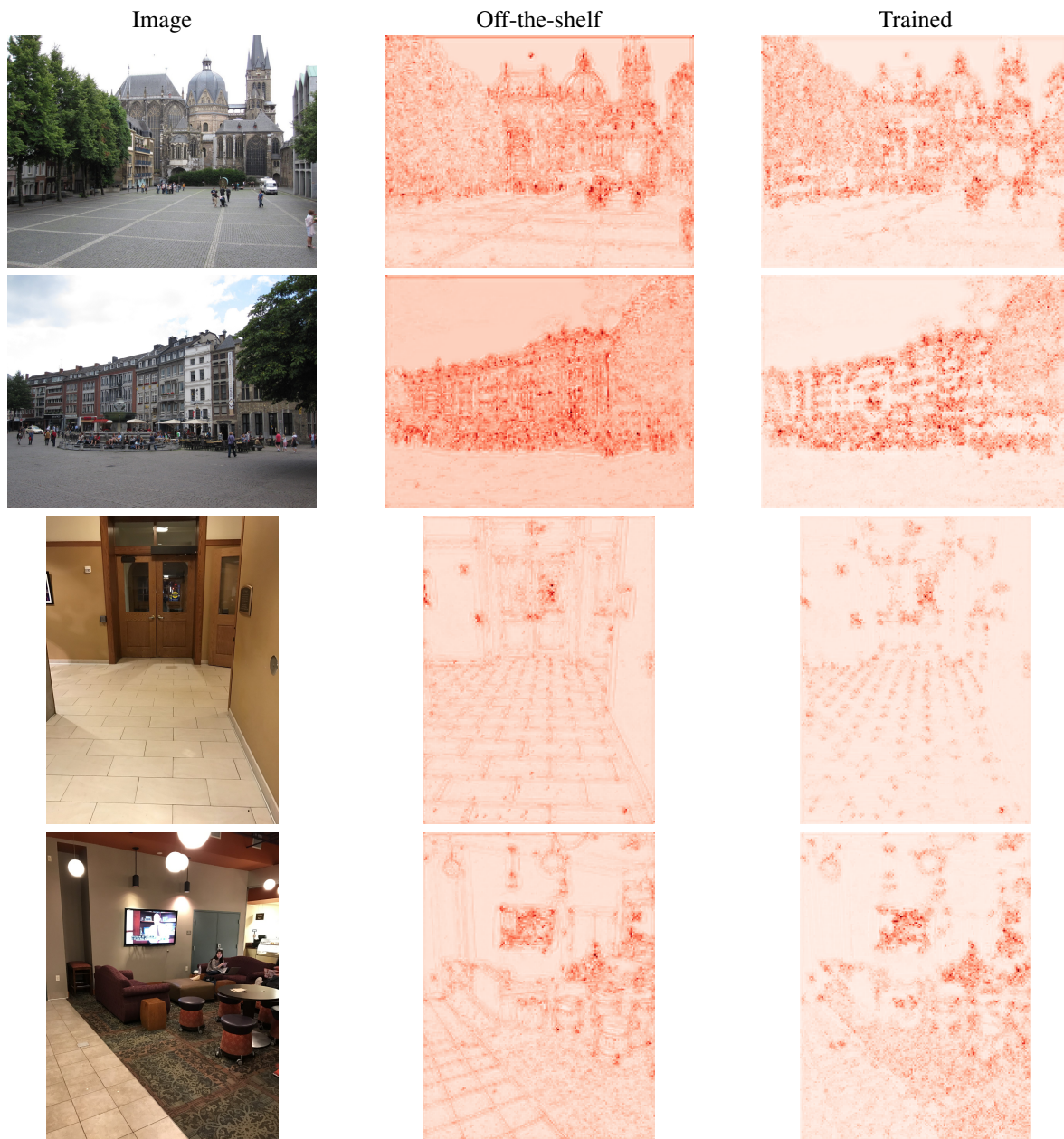
Figure 9: **Soft detection scores for different scenes before and after fine-tuning.** White represents low soft-detection scores while red signifies higher ones. The training lowers the soft-detection scores on repetitive structures (*e.g.* ground, floor, walls) while it enhances the score on more distinctive points. This shown by the increased contrast of the trained soft-detection maps with respect to their off-the-shelf counterparts.
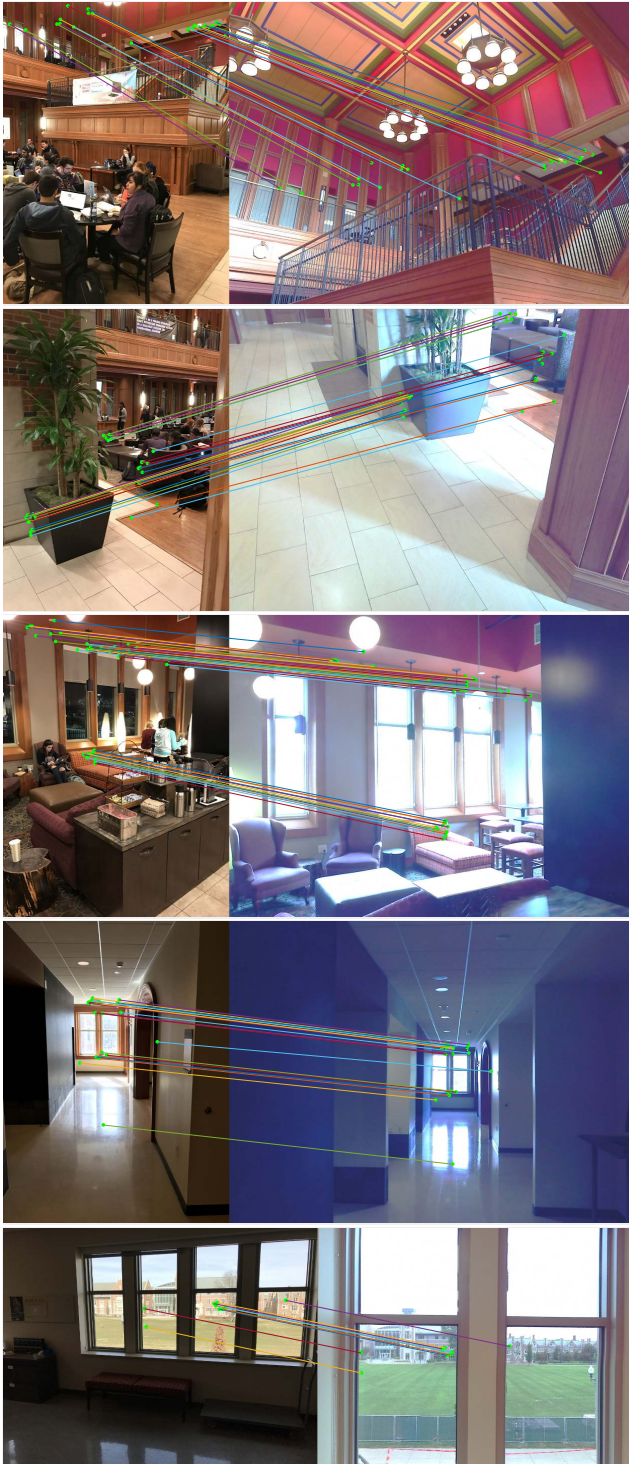
Figure 10: **Examples of correctly matched image pairs from the InLoc [59] dataset.** Our features are robust to significant changes in viewpoint as it can be seen in the first example. In textureless areas, our features act as an object matcher - correspondences are found between the furniture of different scenes. Sometimes, matches are even found across windows on nearby buildings.
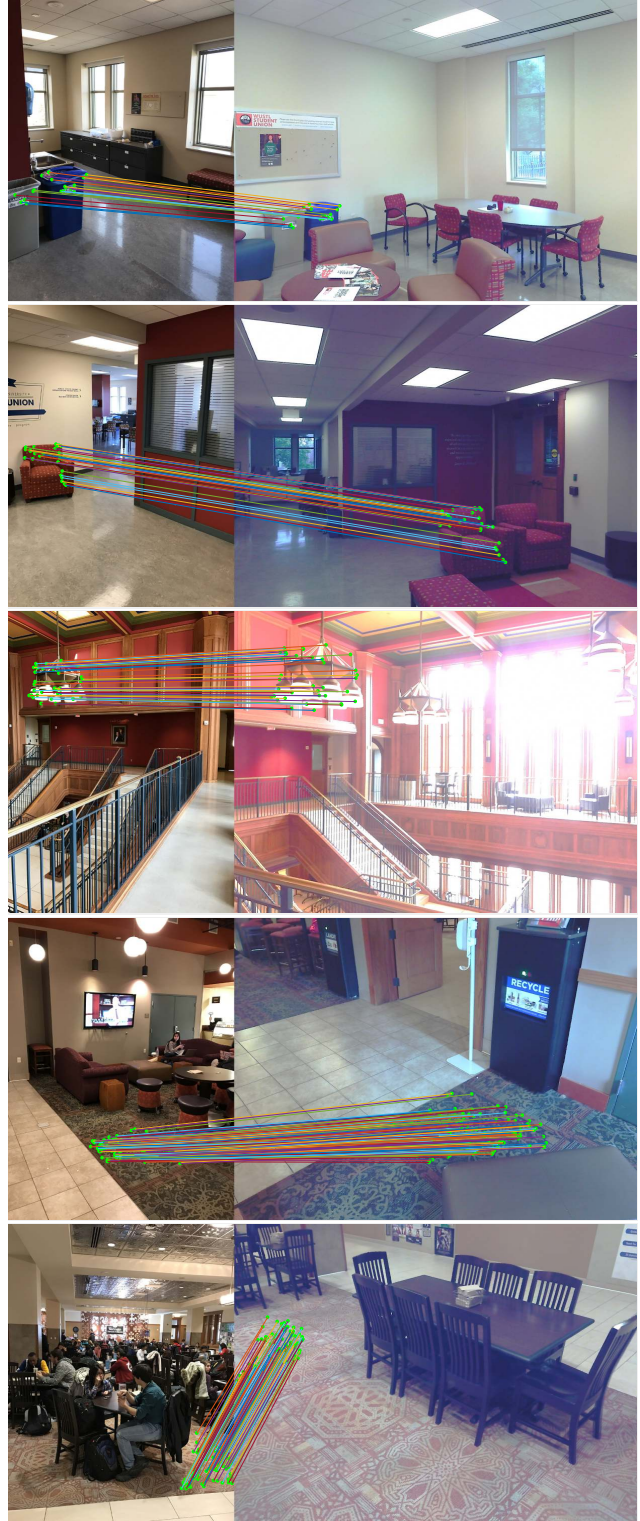

Figure 11: **Failure cases from the InLoc [59] dataset.** Even though they are visually correct, the matches sometimes put in correspondence identical objects from different scenes. Another typical error case is due to repeated patterns (e.g. on carpets) which yield a significant number of inliers.
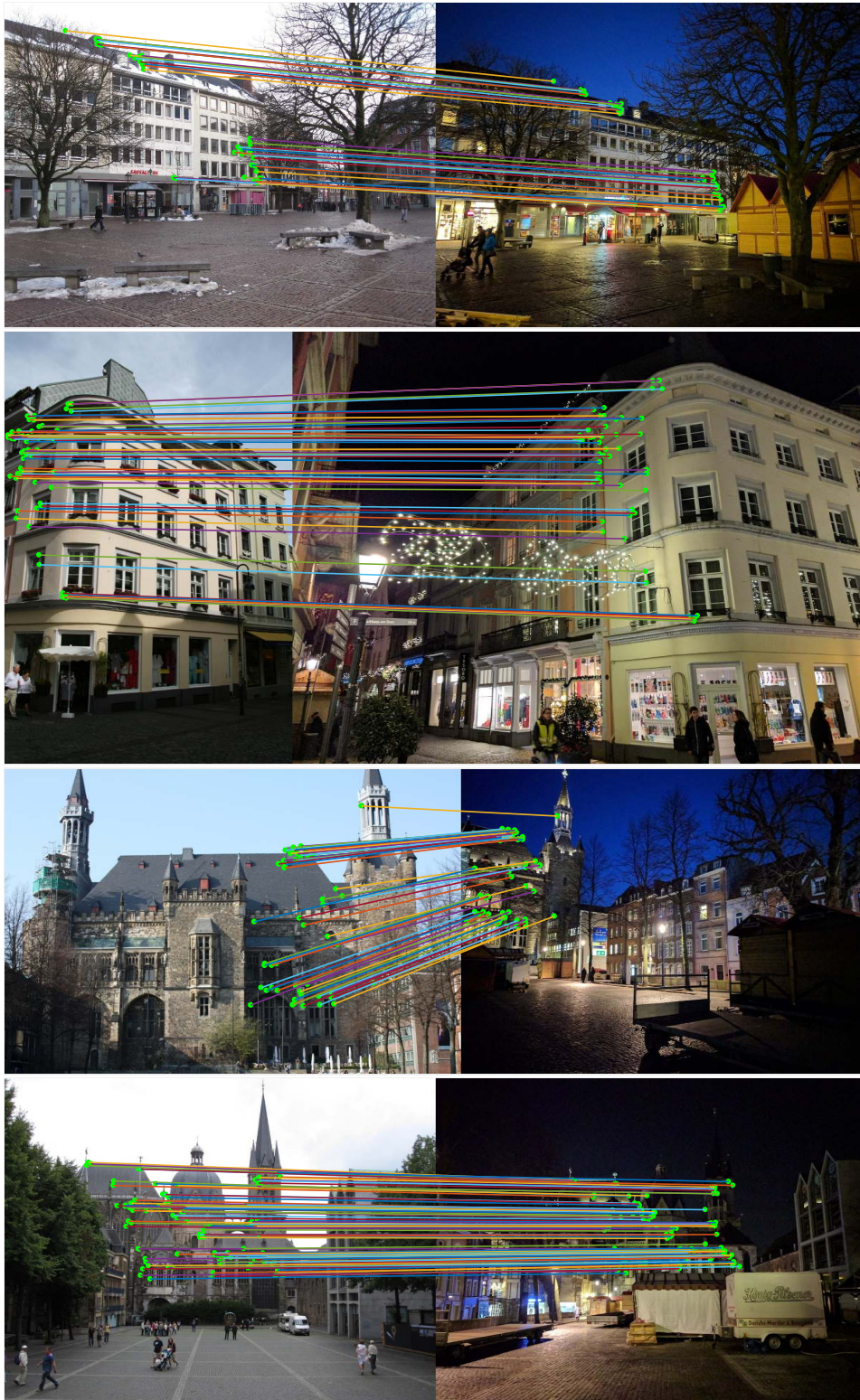
Figure 12: **Examples of correctly matched image pairs from the Aachen Day-Night [46,48] dataset.** Our features consistently provide a significant number of good matches between images with strong illumination changes. The first two image pairs come from scenes where no other method was able to register the night-time image. For the last two, DELF [38] was the only other method that succeeded.