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

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

This supplementary material provides the following additional information: Section 1 details how we chose the threshold for Lowe's ratio test [5] used for the 3D reconstructions in Section 5.2 in the paper. As mentioned in Section 4.3 in the paper, Section 2 provides implementation details on the architecture. In addition, the section also evaluates another backbone architecture (ResNet [3]). Section 3 provides additional details on the loss function used to train our method. Section 4 shows qualitative examples for the matches found with our approach on the InLoc [14] and Aachen Day-Night [8, 9] datasets.

1. Impact of the ratio test on D2 features

Throughout our experiments on the local feature evaluation benchmark [11], we noticed that Lowe's ratio test [5] 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 [5]: 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 [1] 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 [10, 12] 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.

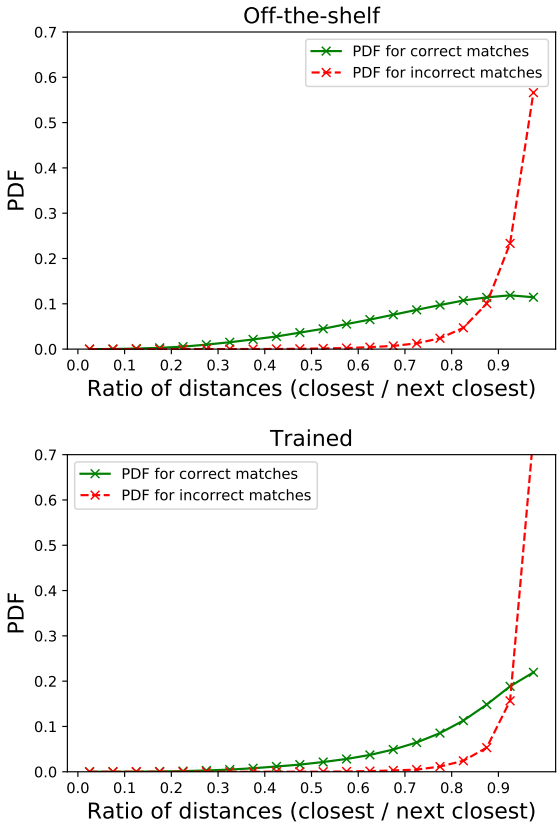


Figure 1: **Ratio PDFs for D2 multi-scale features.** PDF in terms of ratio on the full HPatches [1] image pairs dataset for the D2 off-the-shelf and fine-tuned features. There is no clear separation between the mean ratios of correct and incorrect matches as in the case of SIFT [5].

[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 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 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.

2. Details of the backbone architecture

For the feature extraction network $\mathcal{F}$, we used a VGG16 network pretrained on the ImageNet dataset [2], 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 [4, 15] 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 1 and 2, respectively.

We additionally assess the choice of the network used for feature extraction, by performing a comparison between the chosen VGG16 [13] architecture and ResNet50 [3] (which 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 [2]. In the case of ResNet50, the network was truncated after `conv4_6` (following the approach in DELF [7]). 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.

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
`pool2` - 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

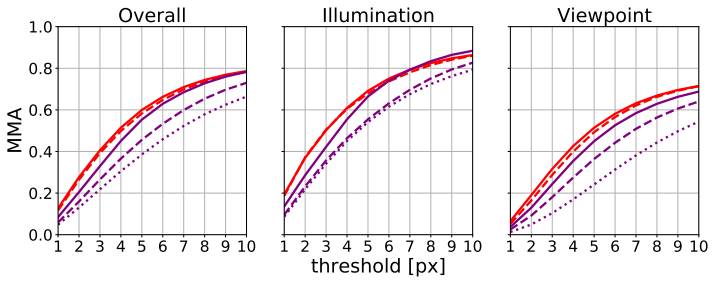
Table 1: **Training architecture.** During training, we use the default VGG16 [13] architecture up to `conv4_3`, and fine-tune the last layer (`conv4_3`).

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
`pool2` - 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 2: **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.

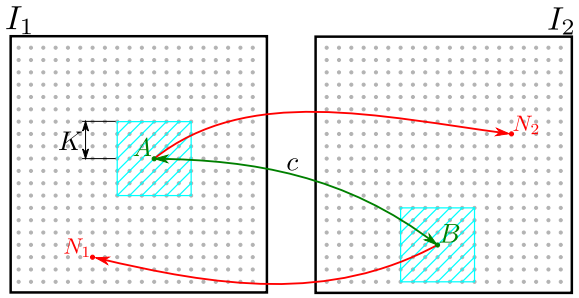
The results can be seen in Figure 2. Dilated convolutions [4, 15] increase the number of detections and the performance of D2 features especially in the case of viewpoint changes. The ResNet50 features also benefit from dilated convolutions and the increase in the resolution. However, although ResNet50 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.

[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.



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

Figure 2: **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.



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

Figure 3: **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.

3. 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 3 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 [6]. 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 4 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.

4. Qualitative examples

Figures 5 and 6 show examples from the InLoc [14] dataset: firstly, we show a few good matches in challenging conditions (significant viewpoint changes and textureless areas) and then we illustrate the main failure modes of D2 features on indoors scenes (repeated objects / patterns). Figure 7 shows some example matches on the difficult scenes from the Aachen Day-Night [8, 9] camera localization challenge.

Figure 4: **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 5: **Examples of correctly matched image pairs from the InLoc [14] 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 6: **Failure cases from the InLoc [14] 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.

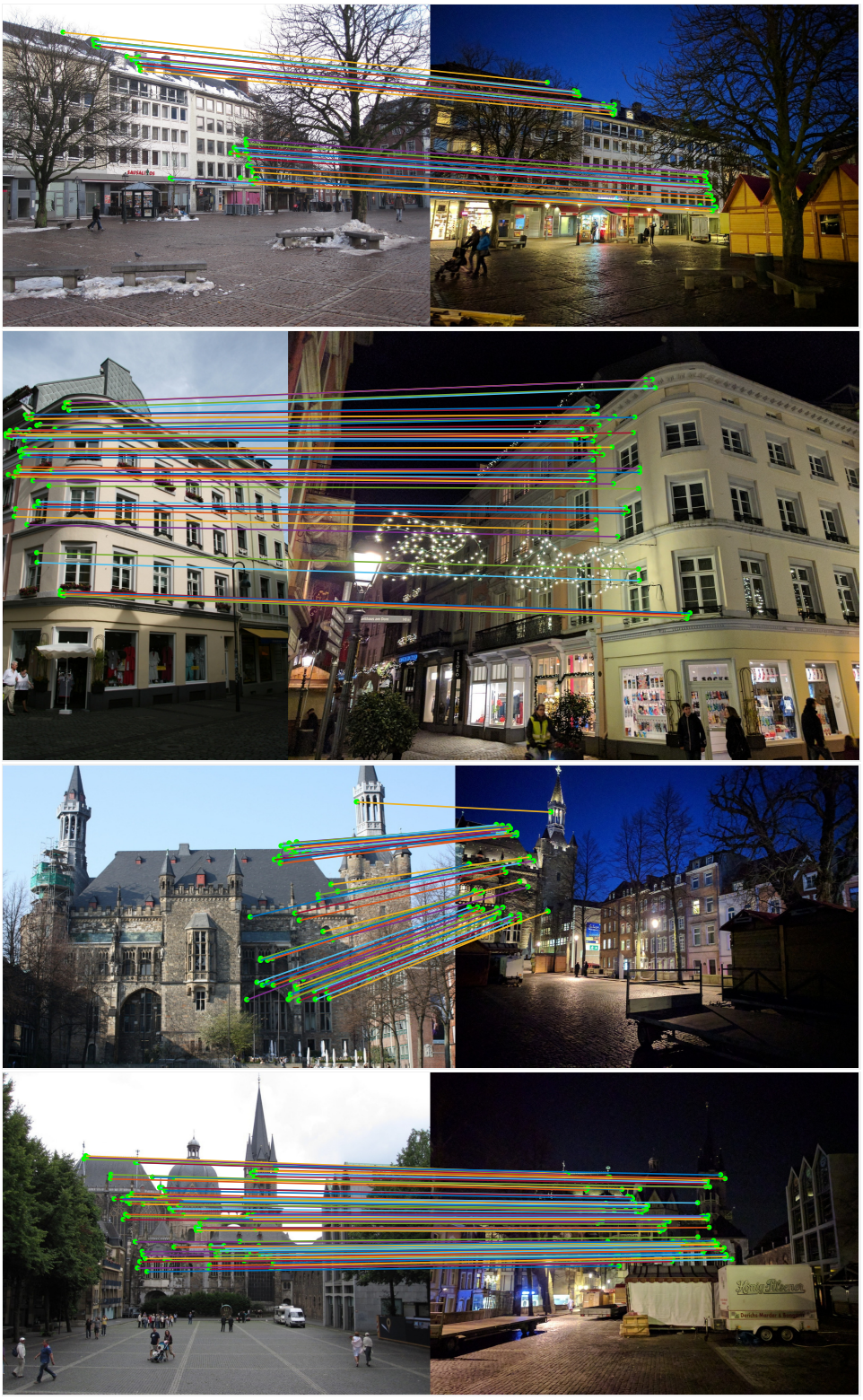


Figure 7: **Examples of correctly matched image pairs from the Aachen Day-Night [8, 9] dataset.** Our features consistently provide a significant number of good matches between images with strong illumination changes. The first two image pairs come from scenes where no other method was able to register the night-time image. For the last two, DELF [7] was the only other method that succeeded.

References

[1] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proc. CVPR*, 2017. 1

[2] 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. 2

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 1, 2

[4] 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. 2

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

[6] Anastasiya Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *NIPS*, 2017. 3

[7] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Largescale image retrieval with attentive deep local features. In *Proc. ICCV*, 2017. 2, 6

[8] 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. 1, 3, 6

[9] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *Proc. BMVC.*, 2012. 1, 3, 6

[10] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proc. CVPR*, 2016. 1

[11] Johannes L. Schönberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *Proc. CVPR*, 2017. 1

[12] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *Proc. ECCV*, 2016. 1

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015. 2

[14] 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. 1, 3, 5

[15] Fisher Yu and Vladlen Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. In *ICLR*, 2016. 2