

HDD-Net: Hybrid Detector Descriptor with Mutual Interactive Learning

Axel Barroso-Laguna¹, Yannick Verdie²,
Benjamin Busam^{2,3}, and Krystian Mikolajczyk¹

¹ Imperial College London
`{axel.barroso17, k.mikolajczyk}@imperial.ac.uk`

² Huawei Noah’s Ark Lab
`{yannick.verdie, benjamin.busam}@huawei.com`

³ Technical University of Munich

Abstract. Local feature extraction remains an active research area due to the advances in fields such as SLAM, 3D reconstructions, or AR applications. The success in these applications relies on the performance of the feature detector, descriptor, and its matching process. While the trend of detector-descriptor interaction of most methods is based on unifying the two into a single network, we propose an alternative approach that treats both components independently and focuses on their interaction during the learning process. We formulate the classical hard-mining triplet loss as a new detector optimisation term to improve keypoint positions based on the descriptor map. Moreover, we introduce a dense descriptor that uses a multi-scale approach within the architecture and a hybrid combination of hand-crafted and learnt features to obtain rotation and scale robustness by design. We evaluate our method extensively on several benchmarks and show improvements over the state of the art in terms of image matching and 3D reconstruction quality while keeping on par in camera localisation tasks.

1 Introduction

At its core, a feature extraction method identifies locations within a scene that are repeatable and distinctive, so that they can be detected with high localisation accuracy under different camera conditions and be matched between different views. The results in vision applications such as image retrieval [1], 3D reconstruction [2], camera pose regression [3], or medical applications [4], among others, have shown the advantage of using sparse features over direct methods.

Classical methods independently compute keypoints and descriptors. For instance, SIFT [5] focused on finding blobs on images and extracting gradient histograms as descriptors. Recently proposed descriptors, especially the patch-based [8,9], are often trained for DoG keypoints [5], and although they may perform well with other detectors [12], their performance can be further improved if the models are trained with patches extracted by the same detector. Similarly, detectors can benefit by training jointly with their associated descriptor [13]. Therefore, following the trend of using the descriptor information to

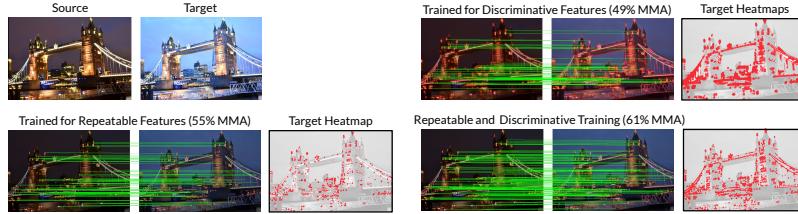


Fig. 1: **Effect of different training strategies on the result.** Correct matches and target detection response maps on *London Bridge* sequence (HPatches) when optimising the detector’s features to be repetitive, discriminative, or both.

infer the detections [14,13,15], we reformulate the descriptor hard-mining triplet cost function [9] as a new detector loss. The new detector term can be combined with any repeatability loss, and consequently, keypoint locations can be optimised based on the descriptor performance jointly with the detector repeatability. This approach leads to finding in a single score map both, repeatable and discriminative features, as shown in figure 1. We extend the network trainings to a multi-scale framework, such that the detector/descriptor learns to use different levels of detail when making predictions.

Our two-networks approach is motivated by the observations that jointly learnt detector-descriptor models [14,17] lack keypoint localisation accuracy, which is critical for SLAM, SfM, or pose estimations [12], and the fact that keypoints are typically well localised on simple structures such as edges or corners, while descriptors require more context to be discriminative. We argue that despite the recent tendency for end-to-end and joint detector-descriptor methods, separate extractors allow for shallow models that can perform well in terms of accuracy and efficiency, which has recently been observed in [12]. Besides that, in contrast to patch-based descriptors, dense image descriptors make it more difficult to locally rectify the image regions for invariance. To address this issue, we introduce an approach based on a block of hand-crafted features and a multi-scale representation within the descriptor architecture, making our network robust to small rotations and scale changes. We term our approach as HDD-Net: Hybrid Detector and Descriptor Network.

In summary, our contributions are:

- A new detector loss based on the hard-mining triplet cost function. Although the hard-mining triplet is widely used for descriptors, it has not been adapted to improve the keypoint detectors.
- A novel multi-scale sampling scheme to jointly train both architectures at multiple scales by combining local and global detections and descriptors.
- We improve the robustness to rotation and scale changes with a new dense descriptor architecture that leverages hand-crafted features together with multi-scale representations.

2 Related Work

We focus the review of related work on learnt methods, and refer to [18,19,12,20,21,22] for further details.

Detectors. Machine learning detectors were introduced with FAST [23], a learnt algorithm to speed up the detection of corners in images. Later, TILDE [24] proposed to train multiple piecewise regressors that were robust under photometric changes in images. DNET [25] and TCDET [26] based its learning on a formulation of the covariant constraint, enforcing the architecture to propose the same feature location in corresponding patches. Key.Net [27] expanded the covariant constraint to a multi-scale formulation, and used a hybrid architecture composed of hand-crafted and learnt feature blocks. More details about the latest keypoint detectors can be found in [21], which provides an extensive detector evaluation.

Descriptors. Descriptors have attracted more attention than detectors, particularly patch-based methods [28,8,9] due to the simplicity of the task and available benchmarks. TFeat [28] moved from loss functions built upon pairs of examples to a triplet based loss to learn more robust representations. In [8], L2-Net architecture was introduced. L2-Net has been adopted in the following works due to its good optimisation and performance. HardNet [9] introduced the hard-mining strategy, selecting only the hardest examples as negatives in the triplet loss function. SOSNet [10] added a regularisation term to the triplet loss to include second-order similarity relationships among descriptors. DOAP [29] reformulated the training of descriptors as a ranking problem, by optimising the mean average precision instead of the distance between patches. GeoDesc [11] integrated geometry constraints to obtain better training data.

Joint Detectors and Descriptors. LIFT [15] was the first CNN based method to integrate detection, orientation estimation, and description. LIFT was trained on quadruplet patches which were previously extracted with SIFT detector. SuperPoint [17] used a single encoder and two decoders to perform dense feature detection and description. It was first pretrained to detect corners on a synthetic dataset and then improved by applying random homographies to the training images. This improves the stability of the ground truth positions under different viewpoints. Similar to LIFT, LF-Net [32] and RF-Net [33] computed position, scale, orientation, and description. LF-Net trained its detector score and scale estimator in full images without external keypoint supervision, and RF-Net extended LF-Net by exploiting the information provided by its receptive fields. D2-Net [14] proposed to perform feature detection in the descriptor space, showing that an already pre-trained network could be used for feature extraction even though it was optimized for a different task. R2D2 [13] introduced a dense version of the L2-Net [8] architecture to predict descriptors and two keypoint score maps, which were each based on their repeatability and reliability. ASLFeat [16] proposed an accurate detector and invariant descriptor with multi-level connections and deformable convolutional networks [34,35].

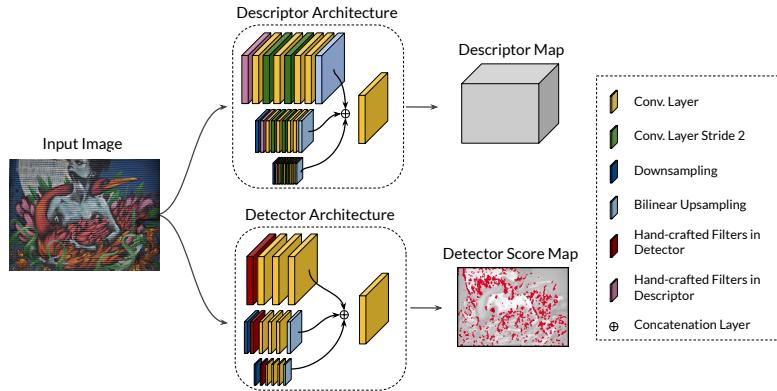


Fig. 2: **HDD-Net Architecture.** HDD-Net is composed by two independent architectures. Instead of sharing a common feature extractor as in [17,14,13,16], HDD-Net focuses its detector-descriptor interaction at the learning level.

3 Method

3.1 HDD-Net Architecture

HDD-Net consists of two independent architectures for inferring the keypoint and descriptor maps, allowing to use different hand-crafted blocks that are designed specifically for each of these two tasks. Figure 2 shows the two independent blocks within the HDD-Net’s feature extraction pipeline.

Descriptor. As our method estimates dense descriptors in the entire image, an affine rectification of independent patches or rotation invariance by construction [36] is not possible. To circumvent this, we design a hand-crafted block that explicitly addresses the robustness to rotation. We incorporate this block before the architecture based on L2-Net [8]. As in the original L2-Net, we use stride convolutions to increase the size of its receptive field, however, we replace the last convolutional layer by a bilinear upsampling operator to upscale the map to its original image resolution. Moreover, we use a multi-scale image representation to extract features from resized images, which provides the network with details from different resolutions. After feature upsampling, multi-scale L2-Net features are concatenated and fused into a final descriptor map by a final convolutional layer. The top part of figure 2 shows the proposed descriptor architecture.

Rotation Robustness. Transformation equivariance in CNNs has been extensively discussed in [37,38,39,40,41]. The two main approaches differ whether the transformations are applied to the input image [42] or the filters [43,41], we follow the latest methods and decide to rotate the filters. Rotating filters is more efficient since they are smaller than the input images, and therefore, have fewer memory requirements. Unlike [43], our rotational filter is not learnt. We show in

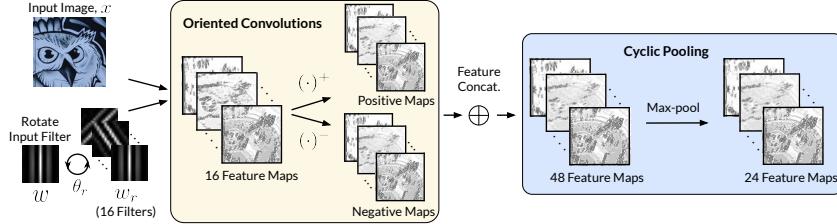


Fig. 3: Hand-crafted Block. Given an input image, x , a designed filter, w , and a set of orientations, θ_r , the rotation robustness is given by extracting and features from x with each of the oriented filters, w_r . Additionally, $(\cdot)^+$ and $(\cdot)^-$ operators split positive and negative maps before the cyclic max-pooling block.

section 4.1 that the pre-designed filters offer a strong feature set that benefits the learning of consecutive CNN blocks. Moreover, in contrast to [43], which applies the rotation to all the layers in their convolutional model, we only focus on the input filter, which further reduces the computational complexity. However, we apply more rotations than [43] to the input filter to provide sufficient robustness. In [41], authors proposed a method that applied multiple rotations to each convolutional filter. Different than estimating a pixel-wise vector field to describe angle and orientation [41], our rotation block returns multiple maxima through a cyclic pooling. The cyclic pooling operator returns local maxima every three neighbouring angles. We experimentally found that returning their local maxima provides better results than only using the global one. Thence, our hand-crafted block applies our input filter, w , at $R = 16$ orientations, each corresponding to the following angles:

$$\theta_r = \frac{360}{R}r \quad \text{and} \quad r \in [1, 2, \dots, R]. \quad (1)$$

A rotated filter is generated by rotating θ_r degrees around the input filter's center. Since our rotated filter is obtained by bilinear interpolation, we apply a circular mask to avoid possible artifacts on the filter's corners:

$$w_r = m \cdot f(w, \theta_r), \quad (2)$$

with m as a circular mask around filter's center and f denoting the bilinear interpolation when rotating the filter, w , by θ_r degrees. Given an input image I , and our designed filter, w , we obtain a set of features $h(I)$ such as:

$$h_r(I) = (I * w_r) \quad \text{and} \quad r \in [1, 2, \dots, R], \quad (3)$$

where $*$ denotes the convolution operator. Before the cyclic max-pooling block, and because max-pool is driven to positive values, we additionally split and concatenate the feature maps in a similar fashion to Descriptor Fields [44]:

$$\mathcal{H}_r(I) = [h_r(I), (h_r(I))^+, -1 \cdot (h_r(I))^-], \quad (4)$$

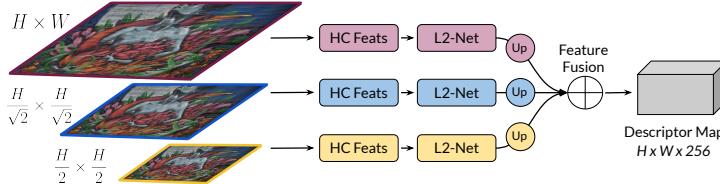


Fig. 4: Multi-Scale Hybrid Descriptor. Gaussian pyramid is fed into our multi-scale descriptor. Each of the re-scaled input images go into one stream, which is composed by the hand-crafted block detailed in section 3.1 and a L2-Net architecture. At the end, multi-scale L2-Net features are upsampled and combined through a final convolution.

with $(\cdot)^+$ and $(\cdot)^-$ operators respectively keeping the positive and negative parts of the feature map $h_r(I)$. Descriptor Fields proved to be effective under varying illumination conditions [44]. Our new set of features, $\mathcal{H}_r(I)$, are concatenated into a single feature map, $\mathcal{H}(I)$. Finally, we apply a cyclic max-pooling block on $\mathcal{H}(I)$. Instead of defining a spatial max-pooling, our cyclic pooling is applied in the channel depth, where each channel dimension represents one orientation, θ_r , of the input filter. Cyclic max-pooling is applied every three neighbouring feature maps with a channel-wise stride of two, meaning that each feature map after max-pooling represents the local maxima among three neighbouring orientations. The full hand-crafted feature block is illustrated in figure 3.

Scale Robustness. Gaussian scale-space has been extensively exploited for local feature extraction [6,45,15]. In [32,33,27], the scale-space representation was used not only to extract multi-scale features but also to learn to combine their information. However, the fusion of multi-scale features is only used during the detection, while, in deep descriptors, it is either implemented via consecutive convolutional layers [17] or by applying the networks on multiple resized images and combining the detections at the end [13,14,16]. In contrast to [14,16], we extend the Gaussian pyramid to the descriptor part by designing a network that takes a Gaussian pyramid as input and fuses the multi-scale features before inferring the final descriptor. To fuse the extracted features, the network upsamples them into the original image resolution in each of the streams. Afterward, features are concatenated and fed into the last convolution, which maps the multi-scale features towards the desired descriptor size dimension as shown in figure 4. The descriptor encoder shares the weights on each multi-scale branch, hence, boosting its ability to extract features robust to scale changes.

Detector. We adopt the architecture of Key.Net [27] as shown in figure 2. Key.Net combines specific hand-crafted filters for feature detection and a multi-scale shallow network. It has recently shown to achieve the state of the art results in repeatability [27,12]. Key.Net extended the covariant loss function proposed in [25] to a multi-scale level, which was termed M-SIP. M-SIP splits the input im-

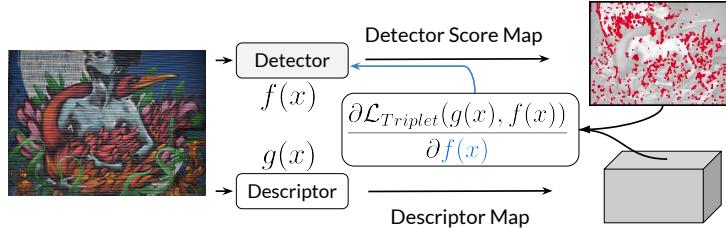


Fig. 5: Detector-Descriptor Interaction. The proposed triplet loss detector term optimises keypoint locations based on the descriptor map, refining the feature candidates towards more discriminative positions.

ages into smaller windows of size $s_1 \times s_1$ and formulates the loss as the difference between soft-argmaximum positions in corresponding regions. M-SIP repeats the process multiple times but splitting the images each time with different window sizes, $s_n \times s_n$. The final loss function proposed by M-SIP between two images, A and B , with their matrix transformation, $H_{b,a}$, is computed as the loss of all windows from all defined scale levels:

$$\mathcal{L}_{M-SIP}(A, B) = \sum_i \| [u_i, v_i]_a^T - H_{b,a} [u_i, v_i]_b^T \|^2. \quad (5)$$

We refer to [27] for further details.

3.2 Descriptor-Detector Training

The detector learning has focused on localising features that are repeatable in a sequence of images [17,32,33,24,21,27], with a few works that determine whether these features are adequate for the matching stage [46,13,15,14]. Since a good feature should be repeatable as well as discriminative [18], we formulate the descriptor triplet loss function as a new detector learning term to refine the feature candidates towards more discriminative positions. Unlike AffNet [46], which estimates the affine shape of the features, we refine only their locations, as these are the main parameters that are often used for end tasks such as SfM, SLAM, or AR. R2D2 [13] inferred two independent response maps, seeking for discriminativeness of the features and their repeatability. Our approach combines both objectives into a single detection map. LIFT [15] training was based on finding the locations with closest descriptors, in contrast, we propose a function based on a triplet loss with a hard-negative mining strategy. D2-Net [14] directly extracts detections from its dense descriptor map, meanwhile, we use KeyNet [27] architecture to compute a score map that represents repeatable as well as discriminative features.

Detector Learning with Triplet Loss. Hard-negative triplet learning maximises the Euclidean distance between a positive pair and their closest negative

sample. In the original work [9], the optimisation happens in the descriptor part, however, we propose to freeze the descriptor such that the sampling locations proposed by the detector are updated to minimise the loss term as shown in figure 5. Then, given a pair of corresponding images, we create a grid on each image with a fixed window size of $s_1 \times s_1$. From each window, we extract a soft-descriptor and its positive and negative samples as illustrated in figure 6. To compute the soft-descriptor, we aggregate all the descriptors within the window based on the detection score map, so that the final soft-descriptor and the scores within a window are entangled. Note that if Non-Maximum Suppression (NMS) was used to select the maximum coordinates and its descriptor, we would only be able to back-propagate through the selected pixels and not the entire map. Consider a window w of size $s_1 \times s_1$ with the score value r_i at each coordinate $[u, v]$ within the window. A softmax provides:

$$p(u, v) = \frac{e^{r(u, v)}}{\sum_{j,k}^{s_1} e^{r(j, k)}}. \quad (6)$$

The window w has the associated descriptor vector d_i at each coordinate $[u, v]$ within the window. We compute the soft-score, \bar{r} , and soft-descriptor, \bar{d} , as:

$$\bar{r} = \sum_{u,v}^{s_1} r(u, v) \odot p(u, v) \quad \text{and} \quad \bar{d} = \sum_{u,v}^{s_1} d(u, v) \odot p(u, v). \quad (7)$$

We use L2 normalisation after computing the soft-descriptor. Similar to previous works [47,33], we sample the hardest negative candidate from a non-neighbouring region. This geometric constraint is illustrated in figure 6. We can define our detector triplet loss with soft-descriptors in window w as:

$$\mathcal{L}(w) = \mathcal{L}(\delta^+, \delta^-, \bar{r}, \mu) = \bar{r} \max(0, \mu + \delta^+ - \delta^-), \quad (8)$$

where μ is a margin parameter, and δ^+ and δ^- are the Euclidean distances between positive and negative soft-descriptors pairs. Moreover, we weight the contribution of each window by its soft-score to control the participation of meaningless windows *e.g.*, flat areas. The final loss is defined as the aggregation of losses on all N_1 windows of size $s_1 \times s_1$:

$$\mathcal{L}_{Trip}(s_1) = \sum_n^{N_1} \mathcal{L}(w_n) = \sum_n^{N_1} \mathcal{L}(\delta_n^+, \delta_n^-, \bar{r}_n, \mu). \quad (9)$$

Multi-Scale Context Aggregation. We extend equation 9 to a multi-scale approach to learn features that are discriminative across a range of scales. Multi-scale learning was used in keypoint detection [27,32,33], however, we extend these works by using the multi-scale sampling strategy not only on the detector but also on the descriptor training. Thus, we sample local soft-descriptors with varying window sizes, s_j with $j \in [1, 2, \dots, S]$, as shown in figure 6, and combine their losses with control parameters λ_j in a final term:

$$\mathcal{L}_{MS-Trip} = \sum_j \lambda_j \mathcal{L}_{Trip}(s_j), \quad (10)$$

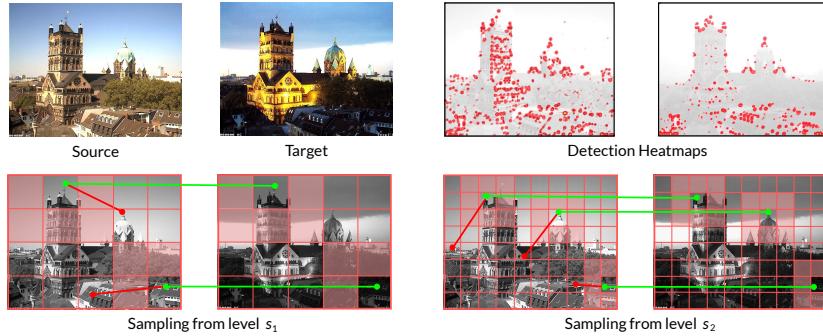


Fig. 6: Triplet Formation Pipeline. Soft-descriptors are extracted from each window together with their respective positives and the hardest negatives. The negatives are extracted only from non-neighbouring areas (non-red areas).

Repeatable & Discriminative. The detector triplet loss optimises the model to find locations that can potentially be matched. As stated in [18], discriminativeness is not sufficient to train a suitable detector. Therefore, we combine our discriminative loss and the repeatability term M-SIP proposed in [27] with control parameter β to balance their contributions:

$$\mathcal{L}_{R \& D} = \mathcal{L}_{M-SIP} + \beta \mathcal{L}_{MS-Trip}, \quad (11)$$

Entangled Detector-Descriptor Learning. We frame our joint optimisation strategy as follows. The detector is optimised by equation 11, meanwhile, the descriptor learning is based on the hard-mining triplet loss [9]. For the descriptor learning, we use the same sampling approach as in figure 6, however, instead of sampling soft-descriptors, we sample a point-wise descriptor per window. The location to sample the descriptor is provided by an NMS on the detector score map. Hence, the descriptor learning is conditioned by the detector score map sampling, meanwhile, our triplet detector loss term refines its candidate positions using the descriptor space. The interaction between parts tightly couples the two tasks and allows for mutual refinement. We alternate the detector and descriptor optimisation steps during training until a mutual convergence is reached.

3.3 Implementation Details

Training Dataset. We synthetically create pairs of images by cropping and applying random homography transformations to ImageNet images [48]. The image's dimensions after cropping are 192×192 , and the random homography parameters are: rotation $[-30^\circ, 30^\circ]$, scale $[0.5, 2.0]$, and skew $[-0.6, 0.6]$. However, illumination changes are harder to perform synthetically, and therefore, for tackling the illumination variations, we use the AMOS dataset [30], which

Dense-L2Net	1^{st} Order	2^{nd} Order	Gabor Filter	Fully Learnt	$(\cdot)^+$ & $(\cdot)^-$	Multi- Scale	MMA (%)
✓	-	-	-	✓	-	-	41.8
✓	-	-	-	-	-	-	42.0
✓	✓	-	-	-	-	-	42.5
✓	-	✓	-	-	-	-	43.1
✓	-	-	✓	-	-	-	43.3
✓	-	-	-	-	-	✓	43.4
✓	-	-	✓	-	✓	-	43.6
✓	-	-	✓	-	-	✓	44.1
✓	-	-	✓	-	✓	✓	44.5

Table 1: **Ablation Study.** Mean matching accuracy (MMA) on Heinly dataset [49] for different descriptor designs. Best results are obtained with Gabor filters in the hand-crafted block, $(\cdot)^+$ and $(\cdot)^-$ operators, and multi-scale feature fusion.

contains outdoor webcam sequences of images taken from the same position at different times of the year. We experimentally observed that removing long-term or extreme variations *i.e.*, winter-summer, helps the training of HDD-Net. Thus, we filter AMOS dataset such that we keep only images that are taken during summertime between sunrise and midnight. We generate a total of 12,000 and 4,000 images for training and validation, respectively.

HDD-Net Training and Testing. Although the detector triplet loss function is applied to the full image, we only use the top K detections for training the descriptor. We select $K = 20$ with a batch size of 8. Thus, in every training batch, there is a total of 160 triplets for training the descriptor. On the detector site, we use $j = [8, 16, 24, 32]$, $\lambda_j = [64, 16, 4, 1]$, and set $\beta = 0.4$. The hyper-parameter search was done on the validation set. We fix HDD-Net descriptor size to a 256 dimension since it is a good compromise between performance and computational time. Note that the latest joint detector-descriptor methods do not have a standard descriptor size, while [13] is derived from 128-d L2-Net [8], the works in [17,32] use 256-d and [14] is 512-d. During test time, we apply a 15×15 NMS to select candidate locations on the detector score map. HDD-Net is implemented in TensorFlow 1.15 and is available on GitHub⁴.

4 Experimental Evaluation

This section presents the evaluation results of our method in several application scenarios. Due to the numerous possible combinations of independent detectors and patch-based descriptors, the comparison focuses against end-to-end and joint detector-descriptor state of the art approaches.

⁴ <https://github.com/axelBarroso/HDD-Net>

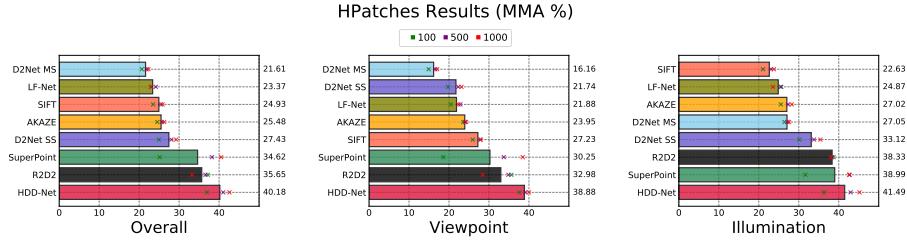


Fig. 7: Mean Matching Accuracy (MMA) on HPatches dataset for top 100, 500 and 1,000 points. HDD-Net gets the best results on both, viewpoint and illumination sequences.

4.1 Architecture Design

Dataset. We use the Heinly dataset [49] to validate our architecture design choices. We focus on its homography set and use only the sequences that are not part of HPatches [20]. We compute the Mean Matching Accuracy (MMA) [50] as the ratio of correctly matched features within a threshold of 5 pixels and the total number of detected features.

Ablation Study. We evaluate a set of hand-crafted filters for extracting features that are robust to rotation. Specifically, 1st and 2nd order derivatives as well as a Gabor filter. Besides, we further test a fully learnt approach without the hand-crafted filters. We also report results showing the impact of splitting the hand-crafted positive and negative features. Finally, our multi-scale approach is tested against a single-pass architecture without multi-scale feature fusion.

Results in table 1 show that the Gabor filter obtains better results than 1st or 2nd order derivatives. Gabor filters are especially effective for rotation since they are designed to detect patterns under specific orientations. Besides, results without constraining the rotational block to any specific filter are slightly lower than the baseline. The fully learnt model could be improved by adding more filters, but if we restrict the design to a single filter, hand-crafted filter with $(\cdot)^+$ and $(\cdot)^-$ operators give the best performance. Lastly, a notable boost over the baseline comes from our proposed multi-scale pyramid and feature fusion within the descriptor architecture.

4.2 Image Matching

Dataset. We use the HPatches [20] dataset with 116 sequences, including viewpoint and illumination changes. We compute results for sequences with image resolution smaller than 1200×1600 following the approach in [14]. To demonstrate the impact of the detector and to make a fair comparison between different methods, we extend the detector evaluation protocol proposed in [21] to the

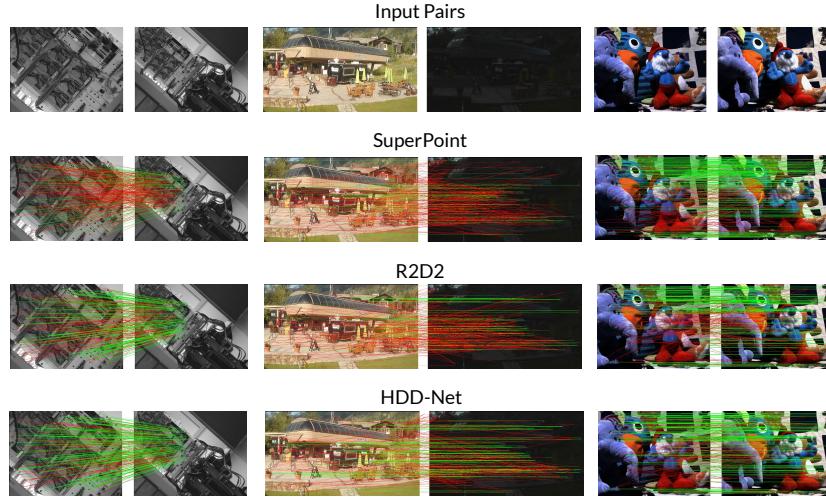


Fig. 8: Qualitative examples on *v_bip*, *i_bridger*, and *i_smurf* from the HPatches dataset. Illustrated sequences display extreme scale and rotation changes, as well as outdoor and indoor illumination variations.

matching metrics by computing the MMA score for the top 100, 500, and 1,000 keypoints. As in section 4.1, MMA is computed as the ratio of correctly matched features within a threshold of 5 pixels and the total number of detected features.

Effect of Triplet Learning on Detector.

Table 2 shows HDD-Net results when training its detections to be repeatable (\mathcal{L}_{MS-SIP}) or/and discriminative ($\mathcal{L}_{MS-Trip}$). The performance of $\mathcal{L}_{MS-Trip}$ only is lower than \mathcal{L}_{MS-SIP} , which is in line with [13]. Being able to detect repeatable features is crucial for matching images, however, best results are obtained with $\mathcal{L}_{R \& D}$, which combines \mathcal{L}_{MS-SIP} and $\mathcal{L}_{MS-Trip}$ with $\beta = 0.4$, and shows the benefits of merging both principles into a single detection map.

Comparison to SOTA. Figure 7 compares our HDD-Net to different algorithms. HDD-Net outperforms all the other methods for viewpoint and illumination sequences on every threshold, excelling especially in the viewpoint change, that includes the scale and rotation transformations for which HDD-Net was designed. SuperPoint [17] performance is lower when using only the top 100 keypoints, and even though no method was trained with such constraint, the other models keep their performance very close to their 500 or 1,000 results. When

	HPatches (MMA)	
	View	Illum
$\mathcal{L}_{MS-Trip}$	26.4	34.9
\mathcal{L}_{MS-SIP}	38.3	35.5
$\mathcal{L}_{R \& D}$ (eq.11)	38.9	41.5

Table 2: MMA (%) results for different detector optimisations.

Madrid Metropolis (448 Images)				Gendarmenmarkt (488 Images)				Tower of London (526 Images)				
Reg.	Sp.	Track	Rep.	Reg.	Sp.	Track	Rep.	Reg.	Sp.	Track	Rep.	
Ims	Pts	Len	Err.	Ims	Pts	Len	Err.	Ims	Pts	Len	Err.	
SIFT	27	1140	4.34	0.69	132	5332	3.68	0.86	75	4621	3.21	0.71
LF-Net	19	467	4.22	0.62	99	3460	4.65	0.90	76	3847	4.63	0.56
SuperPoint	39	1258	5.08	0.96	156	6470	5.93	1.21	111	5760	5.41	0.75
D2Net-SS	–	–	–	–	17	610	3.31	1.04	10	360	2.93	0.94
D2Net-MS	–	–	–	–	14	460	3.02	0.99	10	64	5.95	0.93
R2D2	22	984	4.85	0.88	115	3834	7.12	1.05	81	3756	6.02	1.03
HDD-Net	43	1374	5.25	0.80	154	6174	6.30	0.98	116	6039	5.45	0.80

Table 3: 3D Reconstruction results on the ETH 3D benchmark. Dash symbol (–) means that COLMAP could not reconstruct any model.

constraining the number of keypoints, D2Net-SS [14] results are higher than for its multi-scale version D2Net-MS. D2Net-MS was reported in [14] to achieve higher performance when using an unlimited number of features. In figure 8, we show matching results for the three best-performing methods on hard examples from HPatches. Even though those examples present extreme viewpoint or illumination changes, HDD-Net can match correctly most of its features.

4.3 3D Reconstruction

Dataset. We use the ETH SfM benchmark [51] for the 3D reconstruction task. We select three sequences; *Madrid Metropolis*, *Gendarmenmarkt*, and *Tower of London*. We report results in terms of registered images (Reg. Ims), sparse points (Sp. Pts), track length (Track Len), and reprojection error (Rep. Err.). Top 2,048 points are used as in [12], which still provides a fair comparison between methods at a much lower cost. The reconstruction is performed using COLMAP [2] software where we used one-third of the images to reduce the computational time.

Results. Table 3 presents the results for the 3D reconstruction experiments. HDD-Net and SuperPoint obtain the best results overall. While HDD-Net recovers more sparse points and registers more images in *Madrid* and *London*, SuperPoint does it for *Gendarmenmarkt*. D2-Net features did not allow to reconstruct any model on *Madrid* within the evaluation protocol *i.e.*, small regime on the number of extracted keypoints. Due to challenging examples with moving objects and in distant views, recovering a 3D model from a subset of keypoints makes the reconstruction task even harder. In terms of a track length, that is the number of images in which at least one feature was successfully tracked, R2D2 and HDD-Net outperform all the other methods. LF-Net reports a smaller reprojection error followed by SIFT and HDD-Net. Although the reprojection error is

small in LF-Net, their number of sparse points and registered images are below other competitors.

4.4 Camera Localisation

Dataset. The Aachen Day-Night [52] contains more than 5,000 images, with separate queries for day and night⁵. Due to the challenging data, and to avoid convergence issues, we increase the number of keypoints to 8,000. Despite that, LF-Net features did not converge and are not included in table 4.

Results. The best results for the most permissive error threshold are reported by D2-Net networks and R2D2. Note that D2-Net and R2D2 are trained on MegaDepth [53], and Aachen datasets, respectively, which contains real 3D scenes under similar geometric conditions. In contrast, SuperPoint and HDD-Net use synthetic training data, and while they perform better on image matching or 3D reconstruction, their performance is lower on localisation. As a remark, results are much closer in the most restrictive error, showing that HDD-Net and SuperPoint are on par with their competitors for more accurate camera localisation.

Threshold	Aachen Day-Night		
	Correct	Localised	Queries (%)
	0.5m, 2°	1m, 5°	5m, 10°
SIFT [5]	33.7	52.0	65.3
SuperPoint [17]	42.9	61.2	85.7
D2-Net SS [14]	44.9	65.3	88.8
D2-Net MS [14]	41.8	68.4	88.8
R2D2 [13]	45.9	66.3	88.8
HDD-Net	43.9	62.2	82.7

Table 4: Aachen Day-Night results on localisation. The higher the better.

5 Conclusion

In this paper, we have introduced a new detector-descriptor method based on a hand-crafted block and multi-scale image representation within the descriptor. Moreover, we have formulated the triplet loss function to not only learn the descriptor part but also to improve the accuracy of the keypoint locations proposed by the detector. We validate our contributions in the image matching task, where HDD-Net outperforms the baseline with a wide margin. Furthermore, we show through extensive experiments across different tasks that our approach outperforms or performs as well as the top joint detector-descriptor algorithms in terms of matching accuracy and 3D reconstruction, despite using only synthetic viewpoint changes and much fewer data samples for training.

Acknowledgements. This research was supported by UK EPSRC IPALM project EP/S032398/1.

⁵ We use the benchmark from the CVPR 2019 workshop on Long-term Visual Localization.

References

1. Teichmann, M., Araujo, A., Zhu, M., Sim, J.: Detect-to-retrieve: Efficient regional aggregation for image search. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 5109–5118
2. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 4104–4113
3. Sattler, T., Zhou, Q., Pollefeys, M., Leal-Taixe, L.: Understanding the limitations of cnn-based absolute camera pose regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 3302–3312
4. Busam, B., Ruhkamp, P., Virga, S., Lentes, B., Rackerseder, J., Navab, N., Hengersperger, C.: Markerless inside-out tracking for 3d ultrasound compounding. In: Simulation, Image Processing, and Ultrasound Systems for Assisted Diagnosis and Navigation. Springer (2018) 56–64
5. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60** (2004) 91–110
6. Alcantarilla, P.F., Nuevo, J., Bartoli, A.: Fast explicit diffusion for accelerated features in nonlinear scale spaces. BMVC (2013)
7. Leutenegger, S., Chli, M., Siegwart, R.: Brisk: Binary robust invariant scalable keypoints. In: 2011 IEEE international conference on computer vision (ICCV), Ieee (2011) 2548–2555
8. Tian, Y., Fan, B., Wu, F.: L2-net: Deep learning of discriminative patch descriptor in euclidean space. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 661–669
9. Mishchuk, A., Mishkin, D., Radenovic, F., Matas, J.: Working hard to know your neighbor’s margins: Local descriptor learning loss. In: Advances in Neural Information Processing Systems. (2017) 4826–4837
10. Tian, Y., Yu, X., Fan, B., Wu, F., Heijnen, H., Balntas, V.: Sosnet: Second order similarity regularization for local descriptor learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 11016–11025
11. Luo, Z., Shen, T., Zhou, L., Zhu, S., Zhang, R., Yao, Y., Fang, T., Quan, L.: Geodesc: Learning local descriptors by integrating geometry constraints. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 168–183
12. Yuhe, J., Mishkin, D., Mishchuk, A., Matas, J., Fua, P., Moo Yi, K., Trulls, E.: Image matching across wide baselines: From paper to practice. In: arXiv preprint arXiv:2003.01587. (2020)
13. Revaud, J., Weinzaepfel, P., De Souza, C., Pion, N., Csurka, G., Cabon, Y., Humenberger, M.: R2d2: Repeatable and reliable detector and descriptor. Advances in Neural Information Processing Systems (2019)
14. Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T.: D2-net: A trainable cnn for joint detection and description of local features. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
15. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: Lift: Learned invariant feature transform. In: European Conference on Computer Vision, Springer (2016) 467–483
16. Luo, Z., Zhou, L., Bai, X., Chen, H., Zhang, J., Yao, Y., Li, S., Fang, T., Quan, L.: Aslfeat: Learning local features of accurate shape and localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2020)

17. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2018) 224–236
18. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: a survey. Foundations and Trends in Computer Graphics and Vision (2008)
19. Csurka, G., Dance, C.R., Humenberger, M.: From handcrafted to deep local features. arXiv preprint arXiv:1807.10254 (2018)
20. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 5173–5182
21. Lenc, K., Vedaldi, A.: Large scale evaluation of local image feature detectors on homography datasets. BMVC (2018)
22. Bojanic, D., Bartol, K., Pribanic, T., Petkovic, T., Donoso, Y.D., Mas, J.S.: On the comparison of classic and deep keypoint detector and descriptor methods. In: 2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA), IEEE (2019) 64–69
23. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: European conference on computer vision, Springer (2006) 430–443
24. Verdie, Y., Yi, K., Fua, P., Lepetit, V.: Tilde: a temporally invariant learned detector. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 5279–5288
25. Lenc, K., Vedaldi, A.: Learning covariant feature detectors. In: European Conference on Computer Vision, Springer (2016) 100–117
26. Zhang, X., Yu, F.X., Karaman, S., Chang, S.F.: Learning discriminative and transformation covariant local feature detectors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 6818–6826
27. Barroso-Laguna, A., Riba, E., Ponsa, D., Mikolajczyk, K.: Key.net: Keypoint detection by handcrafted and learned cnn filters. (2019) 5836–5844
28. Balntas, V., Riba, E., Ponsa, D., Mikolajczyk, K.: Learning local feature descriptors with triplets and shallow convolutional neural networks. In: BMVC. Volume 1. (2016) 3
29. He, K., Lu, Y., Sclaroff, S.: Local descriptors optimized for average precision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 596–605
30. Pultar, M., Mishkin, D., Matas, J.: Leveraging outdoor webcams for local descriptor learning. Proceedings of CVWW 2019 (2019)
31. Tian, Y., Barroso-Laguna, A., Ng, T., Balntas, V., Mikolajczyk, K.: HyNet: Local descriptor with hybrid similarity measure and triplet loss. arXiv preprint arXiv:2006.10202 (2020)
32. Ono, Y., Trulls, E., Fua, P., Yi, K.M.: Lf-net: learning local features from images. In: Advances in Neural Information Processing Systems. (2018) 6234–6244
33. Shen, X., Wang, C., Li, X., Yu, Z., Li, J., Wen, C., Cheng, M., He, Z.: Rf-net: An end-to-end image matching network based on receptive field. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 8132–8140
34. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. (2017) 764–773
35. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 9308–9316

36. Ebel, P., Mishchuk, A., Yi, K.M., Fua, P., Trulls, E.: Beyond cartesian representations for local descriptors. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 253–262
37. Cohen, T., Welling, M.: Group equivariant convolutional networks. In: International conference on machine learning. (2016) 2990–2999
38. Follmann, P., Bottger, T.: A rotationally-invariant convolution module by feature map back-rotation. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE (2018) 784–792
39. Worrall, D.E., Welling, M.: Deep scale-spaces: Equivariance over scale. Advances in Neural Information Processing Systems (2019)
40. Dieleman, S., Willett, K.W., Dambre, J.: Rotation-invariant convolutional neural networks for galaxy morphology prediction. Monthly notices of the royal astronomical society **450** (2015) 1441–1459
41. Marcos, D., Volpi, M., Komodakis, N., Tuia, D.: Rotation equivariant vector field networks. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 5048–5057
42. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: Advances in neural information processing systems. (2015) 2017–2025
43. Dieleman, S., De Fauw, J., Kavukcuoglu, K.: Exploiting cyclic symmetry in convolutional neural networks. ICML (2016)
44. Crivellaro, A., Lepetit, V.: Robust 3d tracking with descriptor fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 3414–3421
45. Mikolajczyk, K., Schmid, C.: Indexing based on scale invariant interest points. ICCV (2001)
46. Mishkin, D., Radenovic, F., Matas, J.: Repeatability is not enough: Learning affine regions via discriminability. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 284–300
47. Mishkin, D., Matas, J., Perdoch, M.: Mods: Fast and robust method for two-view matching. Computer Vision and Image Understanding **141** (2015) 81–93
48. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
49. Heinly, J., Dunn, E., Frahm, J.M.: Comparative evaluation of binary features. In: European Conference on Computer Vision, Springer (2012) 759–773
50. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE transactions on pattern analysis and machine intelligence **27** (2005) 1615–1630
51. Schonberger, J.L., Hardmeier, H., Sattler, T., Pollefeys, M.: Comparative evaluation of hand-crafted and learned local features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 1482–1491
52. Sattler, T., Maddern, W., Toft, C., Torii, A., Hammarstrand, L., Stenborg, E., Safari, D., Okutomi, M., Pollefeys, M., Sivic, J., et al.: Benchmarking 6dof outdoor visual localization in changing conditions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 8601–8610
53. Li, Z., Snavely, N. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 2041–2050