

A Large Dataset for Improving Patch Matching

Rahul Mitra¹, Nehal Doiphode¹, Utkarsh Goutam¹, Sanath Narayan², Shuaib Ahmed², Sharat Chandran¹ and Arjun Jain¹

¹Indian Institute of Technology Bombay, Mumbai, India

²Mercedes Benz Research and Development India, Bengaluru, India

{rmitter, sharat, ajain}@cse.iitb.ac.in, nehald@ee.iitb.ac.in, utkarshg@iitb.ac.in,
{sanath.narayan, shuaib.ahmed}@daimler.com

Abstract

We propose a new dataset for learning local image descriptors which can be used for significantly improved patch matching. Our proposed dataset consists of an order of magnitude more number of scenes, images, and positive and negative correspondences compared to the currently available Multi-View Stereo (MVS) dataset from Brown et al. [1]. The new dataset also has better coverage of the overall viewpoint, scale, and lighting changes in comparison to the MVS dataset. Our dataset also provides supplementary information like RGB patches with scale and rotations values, and intrinsic and extrinsic camera parameters which as shown later can be used to customize training data as per application.

We train an existing state-of-the-art model on our dataset and evaluate on publicly available benchmarks such as HPatches dataset [2] and Strecha et al. [3] to quantify the image descriptor performance. Experimental evaluations show that the descriptors trained using our proposed dataset outperform the current state-of-the-art descriptors trained on MVS by 8%, 4% and 10% on matching, verification and retrieval tasks respectively on the HPatches dataset. Similarly on the Strecha dataset, we see an improvement of 3-5% for the matching task in non-planar scenes.

1. Introduction

Finding correspondences between images using descriptors is important in many computer vision tasks such as 3D reconstruction, structure from motion (SFM) [4], wide-baseline matching [3], stitching image panoramas [5], and tracking [6, 7]. However, due to changes in viewpoints, scale variations, occlusion, variations in illumination, and shading in the real world scenarios, finding correspon-

dences in-the-wild is challenging and it is an active area of research.

Traditionally, handcrafted descriptors such as SIFT [8], SURF [7], LIOPI [9] were used. These type of descriptors encode pixel, super-pixel or sub-pixel level statistics. However, handcrafted features do not have ability to capture higher structural level information. On the other hand, learning based descriptors using Convolutional Neural Networks (CNNs) have the potential to capture higher level structural information and also to generalize well. Hence, CNN based descriptors are gaining more importance in recent years [10, 11, 12, 13, 14, 15, 16].

Many research works using CNN based descriptors, focus on the architecture [11], defining better loss function [13, 16], and improving training strategies [14, 15] to enhance the quality and achieve state-of-the-art results. As noted in [2], it is unclear that these descriptors can be used for applications where data is not representative of the dataset they are trained with. This is because few datasets are small [17, 18], few lack diversity [1, 19], and in few datasets scenes are obtained through controlled laboratory experiments using small toys [19]. As a result, despite a wide variety of datasets being available in the literature [1, 19, 18, 17, 20], they cannot be employed to design descriptors for applications in-the-wild.

Recently, Hpatches dataset [2] has been proposed as a benchmark for evaluation of local features. This dataset is large and diverse with clear protocols for evaluation metrics and reproducibility . Hpatches dataset has overcome the shortcomings of older smaller datasets such as Oxford-Affine [18] that were used as evaluation benchmarks. Although Hpatches dataset is an excellent benchmark for evaluation, this dataset is seldom used for training as the images in its scenes are related only by 2D homography and such assumptions cannot be made for real-world applications.

Frequently used dataset for training and learning local

descriptors is the Multi-View Stereo (MVS) dataset from Brown *et al.* [1]. The MVS dataset comprises of matching and non-matching pairs for training obtained from scenes of real world objects captured at different viewpoints. However, MVS dataset consists of only 3 scenes and cannot be considered as diverse enough. Data augmentation is one of the traditional method employed to increase the size of dataset. Mishchuk *et al.* [15] highlighted the importance of data augmentation and achieved state-of-the-art results. Regardless, data augmentation cannot substitute the advantages of training with a larger and diverse dataset. These drawbacks of the current datasets limit the potential of powerful CNN based approaches and highlight the necessity for an improved, next generation dataset as concluded in [21].

In this paper, we introduce a novel dataset for training CNN based descriptors that overcomes many drawbacks of current datasets such as MVS. It has sufficiently large number of scenes, is diverse, and has better coverage of the overall viewpoint, scale, and illumination. Moreover, this dataset contains RGB patches including information such as location, scale, and rotation to reverse map them onto the scene. Additionally, this dataset also has intrinsic and extrinsic camera parameters for all the images in a scene which enables one to incorporate the functionality of setting scale and viewpoint variations for matching correspondences. With all the ingredients, this dataset is conducive and ideal for learning descriptors which can also be customized to various diverse tasks of learning including narrow base line matching and wide baseline matching.

A sampling technique for generating matching correspondences is also introduced. This type of sampling ensures that the training dataset has sufficient variations in viewpoint and scale while generating patch-pairs and avoids the generation of redundant patch-pairs having similar contextual information.

We use the current state-of-the-art Hardnet model [15] and train using the proposed dataset, while maintaining the training strategy identical to [15]. We show that, using our dataset for training improves the performance over Hardnet for various tasks on different benchmark datasets.

2. Related Work

The success of CNNs in various computer vision tasks can be partly attributed to availability of large datasets for training. An ideal dataset for learning a particular task should capture the all the real world scenario involved with the task. An example being the ImageNet [22] dataset for image classification. In the context of learning patch descriptors the dataset provided by Brown et al. [1] is the most widely used for training. The dataset contains 3 scenes *viz.*, *liberty*, *notredame* and *yosemite*. Each scene consists of a large collection of images. Dense 3D point cloud and visibility maps are estimated from the set of images. The 3D

points are projected in different reference images accounting visibility to extract patches. Each scene contains more than 400,000 patches. Patches belonging to same 3D point form matching pairs. However, the dataset suffers from two major drawbacks. Firstly, it lacks data diversity as it contains only 3 scenes. Secondly, inconsistencies in the predicted visibility maps produce noisy matching pairs. In Fig. 1, few noisy matching pairs from *liberty* and *notredame* scenes are shown. These limitations severely restrict the performance of the descriptors trained with the dataset as shown in Sec. 5.

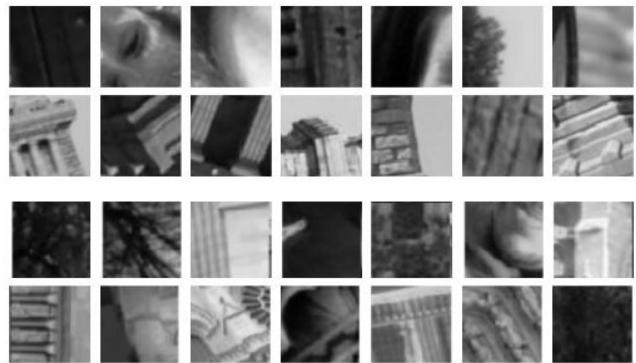


Figure 1: The top two rows show incorrect matching pairs from the *liberty* scene. Patches in a column form a pair. The bottom two rows shows the same for *notredame*

The *DTU* dataset [19] contains images and 3D point clouds of small objects obtained using a robotic arm in a controlled laboratory environment. Images are taken from different view points with varying illumination. Although the size of the dataset is big in number of images, it does not capture intricacies of images in the wild.

The CDVS dataset [20] is another large patch based dataset offering more number of scenes than the MVS dataset. However, as shown in Fig. 2 the matching pairs in the dataset does not have severe deformations. A quantitative analysis depicting the weakness of this dataset is presented in [23].



Figure 2: Shows sample matching pairs from the CDVS dataset [20]. It is evident that the pairs do not encompass the necessary challenges encountered in the wild.

The Oxford-Affine dataset [18] is a small dataset containing 8 scenes with sequence of 6 images per scene. The images in a sequence are related by homographies. Al-

though the dataset is suitable for benchmarking evaluations, it is too small for training CNN models. Similar to Oxford-Affine, another dataset exists where matching pairs are created synthetically [17]. In this dataset, every scene contains a reference image and a collection of images which are transformations of the reference image. The dataset has good variations in scene content and deformations. However, the deformations are only limited to homographies. Table 1 gives a comparison of the various publicly available datasets.

Dataset \ Features	Diverse	Real	Large	Non-Planar
MVS [1]		✓	✓	✓
DTU [19]		✓	✓	✓
CDVS [20]	✓		✓	
Oxford-Affine [18]	✓	✓		
Synthetic [17]	✓	✓		
Proposed	✓	✓	✓	✓

Table 1: Shows a comparison of different features of various publicly available datasets. It is evident that while our proposed dataset incorporates all the features, others exhibit a subset of it.

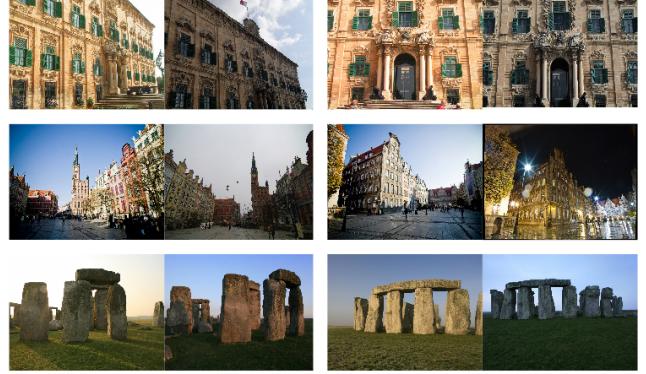
Mishchuk *et al.* [15] used the MVS dataset for training their network and noted that the state-of-the-art results can be achieved by using better CNN architectures and training procedures. However, Schoenberger *et al.* [21], through extensive experiments, highlighted the importance and the necessity of a better training dataset for learning patch descriptors.

Based on all these considerations, the contributions of the paper are:

- (a). A large and novel PS dataset for learning patch descriptors, created from real-world photo-collections, having a good coverage of viewpoint, scale and illumination is proposed.
- (b). A sampling technique to generate high quality matching correspondences without resulting in redundant patch matches is proposed.
- (c). By training the current state-of-the-art model on the proposed dataset and outperforming the model, we show that alongside having better models and training procedures, the quality of the training dataset is also important in realizing the potential of the CNN.

3. Proposed PS Dataset

The dataset proposed in this paper is called PhotoSynth (PS) dataset as images were collected by crawling through Microsoft PhotoSynth. This section focuses on various aspects of the dataset. The description about the scenes and



(a) Image pairs showing illumination variation.



(b) Image pairs showing viewpoint variation.



(c) Image pairs showing scale variation.

Figure 3: Shows sample image pairs from our dataset exhibiting different transformation.

images collected to form the dataset is detailed in Sec. 3.1 followed by the methodology adopted to create data for learning local descriptors out of the vast collection of images and the format of dataset in Sec. 3.2 and Sec. 3.3

3.1. Description of the PS dataset

The PS dataset¹ consists of a total of 30 scenes with 25 scenes for training and 5 scenes for validation. Sample image pairs from the dataset are shown in Fig. 3. It can be observed from Fig. 3 that the diversity of the proposed PS dataset in terms of scene content, illumination, and geometric variations is large.

Each scene in the dataset contains 200 RGB images on an average. The resolution of the images varies from $2000px \times 1350px$ to $800px \times 550px$. The number of patches extracted per scene on an average is 250,000. The number of correspondences depend on the threshold imposed on scale and viewpoint variations. For the training data used in Sec. 4.1, matching correspondences were obtained by setting scale and viewpoint threshold to 2.5 and $\{50^\circ, 75^\circ\}$ respectively. The higher viewpoint threshold is used for scenes which have planar structures. With these thresholds, on an average, 300,000 matching correspondences per scene are generated. Detailed statistics about each scene is provided in the supplementary material.

3.2. Creating the dataset

Structure From Motion (SFM) is adapted to create ground truth pairs of correspondence. To generate the 3D reconstructions, Colmap [24, 25] SFM software is used. The SFM process outputs a 3D point cloud with each point having a list of feature points from different images, with which it is triangulated, and predicted intrinsic and extrinsic camera parameters of each image in the scene. Difference of Gaussian (DOG) [8] feature points are used in our reconstructions.

Patches are extracted by traversing through the list of feature points associated with each 3D point. An extracted patch is scale and rotation normalized by cropping the patch around the feature point with size $12 \times s$, and then rotating the patch by degree r , where s and r are the scale and rotation values of the feature point respectively. The value of s has been limited in the range $[1.6, 15]$, so that minimum and maximum crop sizes are of $20px$ and $128px$ respectively. The resultant patch is then scaled to $48px \times 48px$. All of the experiments reported in this paper are based on patch size of $32px \times 32px$ which is cropped around the center pixel. This facilitates in avoiding border artifacts when applying data augmentation techniques.

As the PS dataset is constructed from photo collections, there are many instances where a particular scene has images that are captured from almost similar viewpoint and scale. Therefore a sampling technique has been adopted to ensure that the sampled correspondence pairs belonging to a particular 3D point have good coverage of viewpoint and

¹The dataset along with trained models is publicly available at <https://github.com/rmitra/PS-Dataset>

scale.

```

Input:  $p_i, \mathbf{p}, \mathbf{v}, \mathbf{f}, \mathbf{d}$ 
Result:  $m_i$  set of matching correspondences of  $p_i$ 
compute matrix A; where  $A[i][j]$  contains the angle
between  $v_i$  and  $v_j$ 
match-found  $\leftarrow$  true
while match-found == true do
    Choose the patch  $p_j$  such that,
     $j = \text{argmax}_{k \in \mathbf{p}} \min_{h \in m_i} A[h][k]$  and
     $j > i$ ,  $A[i][j] \leq \text{MAX\_V\_TH}$ 
     $MVD_j = \min_{h \in m_i} A[h][j]$ 
     $r = \text{argmin}_{h \in m_i} A[h][j]$ 
     $s_{ij} = \max(f_i/d_i, f_j/d_j)/\min(f_i/d_i, f_j/d_j)$ 
     $s_{rj} = \max(f_r/d_r, f_j/d_j)/\min(f_r/d_r, f_j/d_j)$ 
    if ( $MVD_j \geq \text{MIN\_V\_TH} \parallel s_{rj} > 1.5$ ) &&  $s_{ij} <$ 
         $SC\_TH$  then
             $| m_i = m_i \cup p_j$ 
        else
             $| \text{match-found} = \text{false}$ 
        end
    end

```

Algorithm 1: Algorithm to sample matches for patch p_i from \mathbf{p} having suitable scale and viewpoint variation.

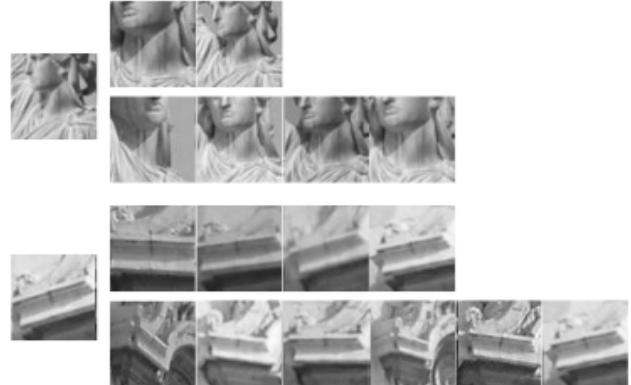


Figure 4: Examples of sampled patches. The left-most column shows two reference patches. For each reference patch, the matching set in top row and bottom row is generated with $\text{MAX_V_TH} = 40^\circ$ and 100° , respectively.

3.3. Sampling matching correspondences

Let \mathbf{P} be a 3D point and $\mathbf{p} = \{p_1, p_2, \dots, p_n\}$ be the set of patches associated with \mathbf{P} . Let $\mathbf{f} = \{f_1, f_2, \dots, f_n\}$ be the estimated focal lengths and $\mathbf{v} = \{v_1, v_2, \dots, v_n\}$ be viewing directions of cameras of \mathbf{p} . Let $\mathbf{c} = \{c_1, c_2, \dots, c_n\}$ be the camera centers. We calculate $\mathbf{d} = \{d_1, d_2, \dots, d_n\}$ to be the distance of \mathbf{P} from camera centers \mathbf{c} in the direction of \mathbf{v} i.e. $d_i = v_i \cdot (\mathbf{P} - c_i)$.

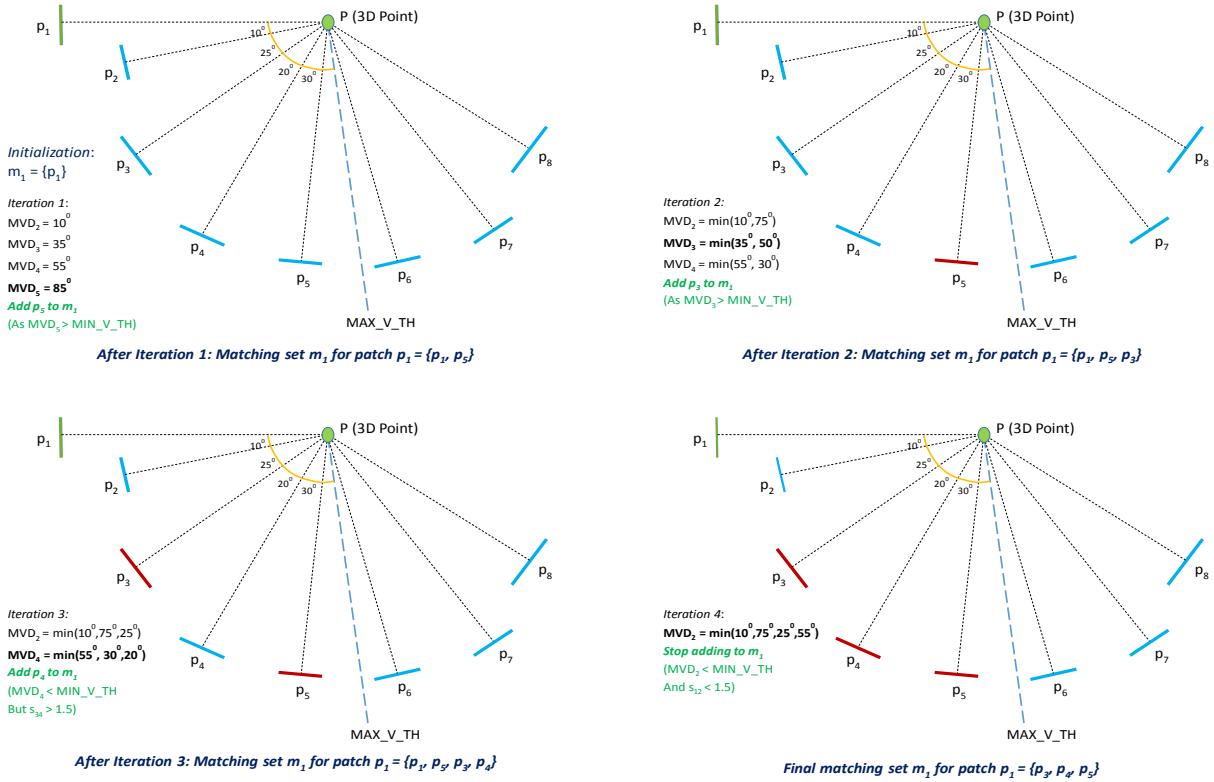


Figure 5: An example of sampling technique for identifying matching pairs is shown (figure not to scale). 8 patches p_i of 3D point P are considered. Here MIN_V_TH and MAX_V_TH are 25° and 100° respectively. Figure shows the iterations for generating the matching set for reference patch p_1 (in green). Patches beyond MAX_V_TH from p_1 are not considered. Each sub-figure shows the patches (apart from p_1) in matching set, before start of that iteration, in red. The patch with maximum MVD (in bold) is considered in every iteration. The algorithm stops when no patch is added to the matching set.

The scale between two patches can be estimated by comparing their f/d ratio. Let SC_TH , MIN_V_TH , MAX_V_TH be user defined thresholds for scale, minimum viewpoint difference and maximum viewpoint difference between the pairs. To form matching correspondences with varied viewpoint changes, we initially compute the angle between all possible pairs from \mathbf{p} . Next, given a patch p_i , its matching set m_i is initialized by p_i . Algorithm 1 has been used to fill the matching set m_i .

The algorithm works in an iterative approach. In each iteration, a patch p_k in $\mathbf{p} - m_i$ and within MAX_V_TH from p_i , is assigned a minimum viewpoint difference (MVD) value. The value for p_k is computed as follows. The pairwise viewpoint differences (or angles) between p_k and all patches in m_i are computed and the minimum of these differences is assigned as the MVD for p_k in that iteration. This is repeated for all remaining patches in $\mathbf{p} - m_i$ and within MAX_V_TH from p_i . The patch p_j in $\mathbf{p} - m_i$ having the highest MVD in that iteration is considered. The patch p_j is added to the set m_i if angle between p_i and p_j is more than MIN_V_TH or the scale between the two patches dif-

fers by at least 1.5. The iterations stop when the algorithm fails to add a patch to the set m_i in an iteration. The sampling technique avoids adding redundant pairs to m_i which are very similar to already existing pairs. Hence we can obtain the required coverage in viewpoint and scale without creating all possible pairs. Once m_i is computed, patches in the set $m_i \sim \{p_i\}$ is paired with p_i forming valid matching correspondences.

An example of sampling method for identifying matching pairs is portrayed in Fig. 5 and few examples of matching pairs obtained using the sampling technique is shown in Fig. 4.

4. Experimental Setup

Details of experimental setup used for evaluating various models are discussed in this section. Sec. 4.1 gives the detail about procedure followed to train the model using proposed PS dataset. Description about evaluation is given in Sec. 4.2.

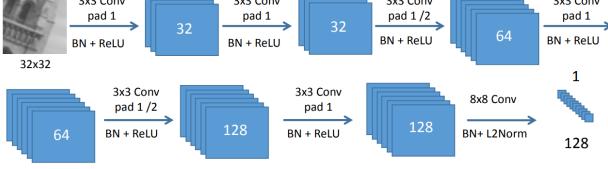


Figure 6: The architecture of the network used for training and evaluation. It is the same as the one used by HardNet [15] without any dropouts. Each convolutional layer is followed by batch normalization and ReLU, except the last one. Similar to HardNet, convolutions with stride 2 are used, instead of pooling in the 2nd and 5th layer.

4.1. Training Procedure

For training purpose, the CNN architecture is adapted from *Hardnet* [15] (also *L2-net* [14] has similar architecture). Since, the CNN is trained on proposed PS dataset, we call it as HardNet-PS. Schematic diagram of the CNN architecture is shown in Fig. 6. It should be noted that the original HardNet and its better variant HardNet+ are trained on MVS dataset [1].

For comparison with HardNet+, the same loss function as described in [15] is adapted. In each iteration, m unique 3D points were randomly sampled, where m is the batch size. For a 3D point \mathbf{P} if there are $\mathbf{p} = \{p_1, p_2, \dots, p_n\}$ patches then the hardest from all the m_i 's (see Sec. 3.2) are chosen based on descriptor distance. Selecting matching pairs from 3D points gives a list of matching pairs $\chi = (a_i, b_i)$; for $i = 1 \dots m$. Next, a pairwise distance matrix D is formed of size $m \times m$, where $D(i, j) = dist(a_i, b_j)$ and function $dist()$ is the L2 distance between the descriptors of a_i and b_j . The selection of the nearest non-matching pair a_{jmin} of a_i and b_{kmin} of b_k are modified as follows:

$$\begin{aligned} a_{jmin} &= \underset{j \in valid_{a_i}}{\operatorname{argmin}} D(a_i, b_j) \\ b_{kmin} &= \underset{k \in valid_{b_i}}{\operatorname{argmin}} D(b_i, a_k) \end{aligned}$$

where $valid_{a_i}$ contains a set of valid b_j 's. Given a_i , a patch b_j is valid w.r.t it, when 3D point \mathbf{P} and \mathbf{Q} corresponding to a_i and b_j have at-least one image in common and their projections in that common image differ by 50% of the un-normalized patch size, i.e before scaling to 48 pixels as done in Sec. 3.2. The average loss over the batch is given in Eq. 1,

$$L = \frac{1}{m} \sum_{i=1,m} \max(0, margin + D(a_i, b_j) - \min(D(a_i, b_{jmin}), D(b_i, a_{kmin})) \quad (1)$$

To reduce generalization error, augmentation of data is carried out by randomly rotating the patches between $[-22.5^\circ, 22.5^\circ]$ and scaling within $[1.0, 1.1]$.

4.2. Evaluation procedure

Two evaluation benchmark were used for fair performance comparison, namely, HPatches for planar objects and Strecha for non-planar objects. The procedure followed to evaluate them are given in Sec. 4.2.1 and 4.2.2 respectively. As in the case with all other descriptors, HardNet-PS is also not trained using any of these two evaluation datasets.

4.2.1 HPatches Benchmark

The *HPatches* benchmark dataset contains image sequences which vary either in viewpoint or in illumination. It has 59 scenes with geometric deformations (viewpoint) and 57 scenes with photometric changes (illumination). Three type of detectors namely DOG, Hessian, and Harris affine are used to extract key points. While extracting key points, additional geometric noise in 3 levels were introduced, namely easy, medium, and hard. Brief overview of the three evaluation procedures or protocols in HPatches are listed below [2],

Patch verification: Verification is to classify a list of pair of patches as matching or non-matching. Each pair is also assigned a similarity score based on the L2 distance of the descriptors of the two patches. Classification is done on the basis of similarity score. Mean Average Precision (mAP) is calculated based on the list of similarity scores.

Image matching: It is a task of matching key points from reference image to target image. This is done using nearest neighbor on descriptors of the key points. Each predicted match is also associated with a similarity score like patch verification and mAP is calculated over the list of predictions.

Patch retrieval: In this protocol, a patch is queried in a large collection of patches majorly consisting of distractors. A similarity score coherent with the previous evaluations is computed between the query patch and collection of patches. The evaluation is carried out by varying the number of distractors and taking mean.

4.2.2 Strecha Benchmark

The *HPatches* benchmark evaluatoin provides a comprehensive evaluation for image sequences related by 2D homography. However, it does not capture image pairs in-the-wild which are non-planar, having self and external occlusions. Hence, the *Herjesu* and *Fountain* scenes from [3] which have wide-baseline image pairs on non-planar objects has been adapted to evaluate critically. The dataset provides images with projection matrices and a dense point cloud of the scene. The *Herjesu-P8* scene contains 8 images indexed from 0 to 7 with gradual shift in viewpoint when iterated in order. In other words, the image pair $\{0, 7\}$ has the highest viewpoint difference. Similarly, the *Fountain-P11* scene

has a sequence with 11 images.

To ensure high repeatability we assume one of the image in the sequence as the reference image and extract keypoints from it and transfer them to the other images. The following steps are used to transfer a point p_r from the reference image to a target image:

1. Project all 3D points in the reference image.
2. Find the 3D point P_{nn} whose projection onto the reference image is nearest to p_r and within 3 pixels distance.
3. if P_{nn} exists, project it to the other image.

The reference images used in *Fountain-P11* and *Herzjesu-P8* are index “5” and index “4” respectively.

DOG key-points with 4 octaves and 3 scales per octave were used. The peak and edge threshold are set to 0.02/3 and 10 respectively. Points with scales larger than 1.6 are retained for stability with at-most 2 orientations per point. *vl_covdet* [26] is used to extract patches from the images with default parameters values. This makes the smallest patch extracted of size 19×19 which is similar to the support window used by SIFT. In both scenes, we pair all other images with image indexed “0” to form the list of image pairs. We divide the image pairs into 3 categories {narrow, wide, very-wide} on the basis of viewpoint difference. Range of viewpoint change for “Narrow”, “Wide” and “Very-Wide” has been categorized as $[0^\circ - 30^\circ]$, $[30^\circ - 75^\circ]$ and $[75^\circ - 130^\circ]$ respectively. Table. 2 lists the categorized image pairs of both scenes. Since, *Herzjesu* sequence does not have any image pair differing more than 75° in viewpoint, the category “Very-Wide” is not applicable to it.

Baseline Dataset \\\diagdown	Narrow	Wide	Very-Wide
Fountain-P11	“0”-“1”, “0”-“2”, “0”-“3”	“0”-“4”, “0”-“5”, “0”-“6”	“0”-“7”, “0”-“8”, “0”-“9”
Herzjesu-P8	“0”-“1”, “0”-“2”, “0”-“3”	“0”-“4”, “0”-“5”, “0”-“6”, “0”-“7”	NA

Table 2: List of image pairs belonging to different baseline categories for the 2 scenes of Strecha dataset.

Key-point matching is used as metric and followed the same protocol used in HPatches to calculate mAP values. Given a pair of images, we compute the mAP values on 2000 random points visible to both images.

5. Results

Quantitative comparisons between models trained on MVS dataset and HardNet+ trained on our dataset are described in this section. As described in Sec. 4, Hardnet-PS indicates Hardnet+ trained on proposed PS dataset. Results on Hpatches benchmark evaluation and the Strecha benchmark are discussed in Sec. 5.1 and Sec. 5.2 respectively.

5.1. Comparisons on HPatches evaluation benchmark

Results for matching task are shown in Table 3. The results are categorized into illumination and viewpoint sequences. As can be observed, in overall score, HardNet-PS outperforms HardNet+ by a margin of 8%. It is noteworthy that HardNet-PS outperforms all the viewpoint sequences especially on the ‘Hard’ and ‘Tough’ sequences by a large margin of 15.5% and 19.2%, respectively, over the state-of-the-art.

Noise Method \\\diagup	Easy		Hard		Tough		Mean
	Illum	View	Illum	View	Illum	View	
SIFT [8]	43.7	49.4	18.8	21.9	08.7	09.9	25.4
DeepDesc [10]	42.0	46.9	23.9	27.4	13.1	14.7	28.1
Tfeat-m [13]	46.7	57.6	26.3	33.7	13.9	17.4	32.7
Hardnet+ [15]	63.3	71.0	47.0	53.7	30.6	34.9	50.1
Hardnet-PS	58.6	79.5	49.3	69.2	37.9	54.1	58.3

Table 3: Performance comparison for image matching task on HPatches dataset. Illum: illumination sequence. View: viewpoint sequence. Hardnet-PS: Hardnet trained on proposed PS dataset.

The performance comparison on the verification task is shown in Table 4. As in the matching task, the sequences can be categorized into same-sequence (intra) and different sequence (inter). Overall, Hardnet-PS is better than Hardnet+ by 4.4%. The improvement over Hardnet+ increases as the difficulty level of the scenes increase. As it can be seen from Table 4, Hardnet-PS performs notably better by nearly 10% over Hardnet+ on the ‘Tough’ scenes.

Noise Method \\\diagup	Easy		Hard		Tough		Mean
	Inter	Intra	Inter	Intra	Inter	Intra	
SIFT [8]	86.4	79.9	67.4	59.2	52.8	44.9	65.1
DeepDesc [10]	91.3	86.2	84.0	76.2	74.2	64.8	79.5
Tfeat-m [13]	92.5	88.8	85.9	79.7	76.0	68.1	81.8
Hardnet+ [15]	95.4	93.0	91.0	86.8	81.1	76.1	87.1
Hardnet-PS	95.6	93.0	94.1	90.4	90.5	85.4	91.5

Table 4: Performance comparison for patch verification task on HPatches dataset.

The results of the retrieval task in the Hpatches evaluation are reported in Table 5. The Hardnet-PS outperforms the current state-of-the-art Hardnet+ around 10% on

an average. Again, as in the previous tasks, the margin of improvement for Hardnet-PS is higher for the 'Hard' and 'Tough' scenes by 9.3% and 16.5% respectively.

Method \ Noise	Easy	Hard	Tough	Mean
SIFT [8]	64.6	37.4	22.7	41.7
DeepDesc [10]	67.2	52.2	37.8	52.4
Tfeat-m [13]	68.4	50.8	34.7	51.3
Hardnet+ [15]	79.7	68.7	52.6	66.7
Hardnet-PS	82.5	78.0	69.1	76.5

Table 5: Performance comparison for patch retrieval task on HPatches dataset.

5.2. Comparisons on Strecha Dataset

The mAP values of different models for the matching task on the two datasets of Strecha *et al.* [3] is shown in Table. 6 and 7, respectively. Hardnet-PS performs better than the state-of-the-art by nearly 5% and 3.5% on the Fountain-P11 and HerzJesu-P8 scenes respectively. The margin of improvement over Hardnet+ is higher in the 'Very-Wide' category for the Fountain-P11 and the 'Wide' category for the HerzJesu-P8 scene.

Baseline \ Method	Narrow	Wide	Very-Wide	Mean
DeepDesc [10]	76.3	40.8	9.2	40.0
Tfeat-m [13]	86.6	62.8	21.9	57.1
Hardnet+ [15]	92.4	83.2	35.0	70.2
Hardnet-PS	92.8	85.3	47.0	75.0

Table 6: Performance comparison for image matching task on *Fountain-P11* scene

Baseline \ Method	Narrow	Wide	Mean
DeepDesc [10]	64.4	13.1	35.1
Tfeat-m [13]	76.6	27.4	48.5
Hardnet+ [15]	85.1	44.5	61.9
Hardnet-PS	85.1	50.6	65.4

Table 7: Performance comparison for image matching task on *HerzJesu-P8* scene

Qualitative comparison for the matching task on the *Fountain-P11* from the Strecha benchmark is shown in Figure 7. It can be seen that for wide baseline and very wide baseline, the matches from the proposed Hardnet-PS model are better than the matches from Hardnet+ model.

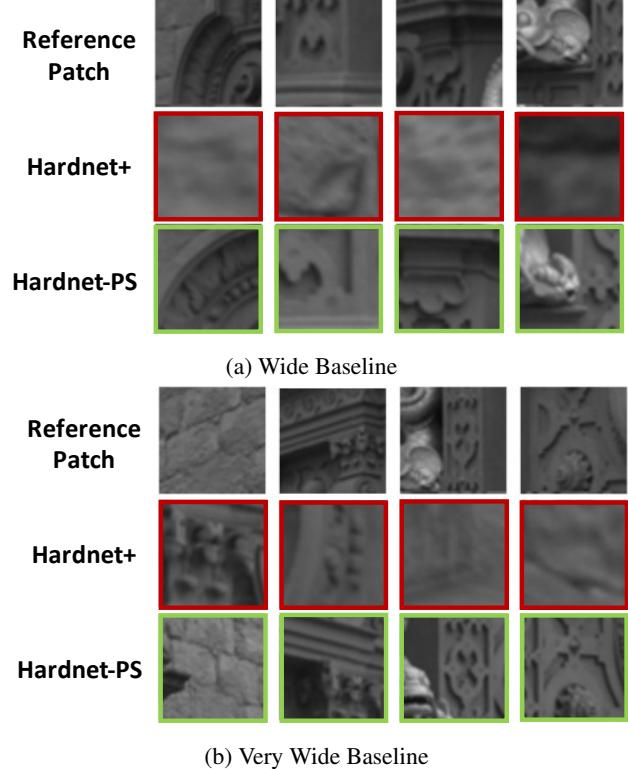


Figure 7: Examples of incorrect matches made by Hardnet+ [15] while matching “wide” and “very-wide” image pairs from scene *Fountain-P11*. The top row in both (a) and (b) represent the patches from the source image “0”. The corresponding predictions are given in the same column. Incorrect and correct predictions are shown in red and green respectively

The results on the HPatches and the Strecha benchmarks indicate a common pattern. The Hardnet+ and the Hardnet-PS models yield comparably close mAP scores for the 'Easy' scenes (HPatches) and 'Narrow' category (Strecha). But, when the difficulty in the scenes increase ('Hard' and 'Tough' or 'Wide' and 'Very-Wide'), the Hardnet-PS model trained on the PS dataset outperforms the state-of-the-art Hardnet+ model by larger margin.

6. Conclusion

In this paper, we have introduced a novel dataset for training CNN based descriptors that overcomes many drawbacks of current datasets such as MVS. It has sufficiently large number of scenes, better coverage of viewpoint, scale, and illumination. We trained the state-of-the-art CNN model available in the literature with the proposed dataset and evaluated on the Hpatches and Strecha benchmark evaluation datasets. On these benchmarks, it has been observed that the model trained with the proposed dataset out-

performs the current state-of-the-art significantly, and the margin of improvement is higher for the difficult scenes ('Hard' and 'Tough' in Hpatches and 'Wide' and 'Very-Wide' scenes in Strecha). With these new state-of-the-art results, we conclude that alongside improving the CNN architecture and the training procedure, a good dataset, such as the proposed PS dataset, conforming to the real-world is also necessary to learn high-quality widely-applicable descriptor.

References

- [1] S. Winder, G. Hua, and M. Brown, "Picking the best daisy," *CVPR*, 2009. [1](#), [2](#), [3](#), [6](#)
- [2] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "Hpatches: A benchmark and evaluation of hand-crafted and learned local descriptors," *CVPR*, 2017. [1](#), [6](#)
- [3] C. Strecha, W. V. Hansen, L. V. Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," *CVPR*, 2008. [1](#), [6](#), [8](#)
- [4] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3D," *ACM SIGGRAPH*, 2006. [1](#)
- [5] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *IJCV*, 2007. [1](#)
- [6] W. He, T. Yamashita, H. Lu, and S. Lao, "Surf tracking," *ICCV*, 2009. [1](#)
- [7] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," *ECCV*, 2006. [1](#)
- [8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, 2004. [1](#), [4](#), [7](#), [8](#)
- [9] B. F. Z. Wang and F. Wu, "Local intensity order pattern for feature description," *ICCV*, 2011. [1](#)
- [10] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," *ICCV*, 2015. [1](#), [7](#), [8](#)
- [11] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," *CVPR*, 2015. [1](#)
- [12] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "Matchnet: Unifying feature and metric learning for patch-based matching," *CVPR*, 2015. [1](#)
- [13] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks," *BMVC*, 2016. [1](#), [7](#), [8](#)
- [14] Y. Tian, B. Fan, and F. Wu, "L2-net: Deep learning of discriminative patch descriptor in euclidean space," *CVPR*, 2017. [1](#), [6](#)
- [15] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," *NIPS*, 2017. [1](#), [2](#), [3](#), [6](#), [7](#), [8](#)
- [16] X. Zhang, F. X. Yu, S. Kumar, and S. Chang, "Learning spread-out local feature descriptors," *ICCV*, 2017. [1](#)
- [17] P. Fisher, A. Dosovitskiy, and T. Brox, "Descriptor matching with convolutional neural networks: a comparison to sift," *arXiv preprint arXiv:1405.5769*, 2014. [1](#), [3](#)
- [18] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE TPAMI*, vol. 27, no. 10, pp. 1615–1630, 2005. [1](#), [2](#), [3](#)
- [19] H. Aans, A. L. Dahl, T. Sattler, and K. S. Pedersen, "Interesting interest points," *IJCV*, 2011. [1](#), [2](#), [3](#)
- [20] V. Chandrasekhar, G. Takacs, D. M. Chen, S. S. Tsai, M. Makar, and B. Girod, "Feature matching performance of compact descriptors for visual search," *Data Compression Conference*, 2014. [1](#), [2](#), [3](#)
- [21] J. L. Schonberger, H. Hardmeier, T. Sattler, and M. Pollefeys, "Comparative evaluation of hand-crafted and learned local features," *CVPR*, 2017. [2](#), [3](#)
- [22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *IJCV*, 2015. [2](#)
- [23] V. Balntas, "efficient learning of local image descriptors," *PhD Thesis, University of Surrey*, 2016. [2](#)
- [24] J. L. Schönberger and J. M. Frahm, "Structure-from-motion revisited," *CVPR*, 2016. [4](#)
- [25] J. L. Schönberger, E. Zheng, M. Pollefeys, and J. M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," *ECCV*, 2016. [4](#)
- [26] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," 2008. [7](#)