Learning-Based Dimensionality Reduction for Computing Compact and Effective Local Feature Descriptors

Hao Dong Xieyuanli Chen Mihai Dusmanu Viktor Larsson Marc Pollefeys Cyrill Stachniss

***Abstract*— A distinctive representation of image patches in form of features is a key component of many computer vision and robotics tasks, such as image matching, image retrieval, and visual localization. State-of-the-art descriptors, from hand-crafted descriptors such as SIFT to learned ones such as HardNet, are usually high dimensional; 128 dimensions or even more. The higher the dimensionality, the larger the memory consumption and computational time for approaches using such descriptors. In this paper, we investigate multi-layer perceptrons (MLPs) to extract low-dimensional but high-quality descriptors. We thoroughly analyze our method in unsupervised, self-supervised, and supervised settings, and evaluate the dimensionality reduction results on four representative descriptors. We consider different applications, including visual localization, patch verification, image matching and retrieval. The experiments show that our lightweight MLPs trained using supervised method achieve better dimensionality reduction than PCA. The lower-dimensional descriptors generated by our approach outperform the original higher-dimensional descriptors in downstream tasks, especially for the hand-crafted ones. The code is available at https://github.com/PRBonn/descriptor-dr.**

I. Introduction

Local feature descriptors [24], [6], [33] are used to represent characteristics of image patches and are designed to be robust to partial occlusions, viewpoint changes, and variations in illumination. They play an essential role in many robotics applications such as robot localization [35], object recognition [14], and image retrieval [28].

The traditional pipeline for local feature extraction often starts by detecting the position, scale, and orientation of keypoints in the image. Then, a normalized image patch is extracted with respect to the estimated keypoint, which usually provides the basis for the descriptor computation. Distinctive and invariant keypoints and descriptors are key to achieving good performance in the subsequent matching, retrieval, and localization tasks. This paper focuses on the descriptor part of this pipeline, specifically the dimensionality reduction of local feature descriptors to generate compact and at the same time effective features.

Multiple visual feature descriptors have been introduced in the literature. Among all the hand-crafted descriptors, SIFT [24] is one of the most famous because of its robustness under blurring, translation, rotation, and scale changes. With the advent of neural networks, more and more learning-based descriptors have been proposed [40], [45], [38], [29] and pushed the state-of-the-art forward in benchmarks for image matching, patch verification, and image retrieval. While the results are promising, a common issue for both hand-crafted and learned methods is that the dimensionality of the generated descriptors is usually high. When the image database increases, substantial time and space might be needed for computing and storing such high-dimensional descriptors. For example, the database of extracted local feature descriptors can easily be tens of GB per scene, which hinders the application to mobile and resource-constraint robots. Several principal component analysis (PCA) [19] based dimensionality reduction methods have been proposed to alleviate this problem. For example, Valenzuela et al. [16] apply PCA to reduce the dimensionality of SIFT and SURF descriptors. Instead of the original SIFT's smoothed weighted histograms, Ke et al. [20] apply PCA to patches for generating lighter descriptors. PCA performs a linear projection from high- to low-dimensional descriptor space, which has limited capabilities of generating high-quality dimensionally reduced descriptors. Moreover, the components with low eigenvalues by PCA are not necessarily less important but down-weighted or even eliminated, which will cause information loss and performance degradation.

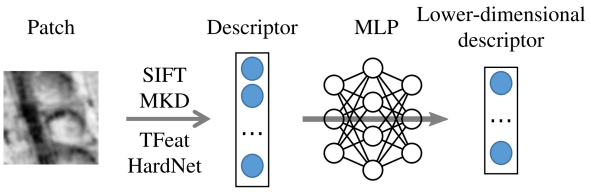

Fig. 1: Overview of our approach. We first compute descriptors of given image patches. Then an MLP-based network is used for dimensionality reduction. We aim to learn an MLP-based projection better than PCA to generate lower-dimensional descriptors.

Unlike previous PCA-based methods, we aim to learn an MLP-based dimensionality reduction to better transform large descriptors into lighter ones. To this end, we thoroughly analyze the MLP-based network for dimensionality reduction and design three learning schemes, including unsupervised, self-supervised, and supervised methods. For the unsupervised scheme, we use an auto-encoder with reconstruction and distance losses to improve the projection. While for the self-supervised one, we iteratively cluster descriptors using k-means and use cluster assignments as pseudo-labels to train the MLPs. We also propose a supervised method that uses ground truth patch labels with triplet loss to supervise the MLPs generating more distinctive lower-dimensional descriptors. We evaluate our MLP-based method with all

H. Dong, X. Chen and C. Stachniss are with the University of Bonn, Germany. M. Dusmanu, and M. Pollefeys are with ETH Zürich, Switzerland. V. Larsson is with Lund University, Sweden. M. Pollefeys is additionally with Microsoft. C. Stachniss is additionally with the Department of Engineering Science at the University of Oxford, UK, and with the Lamarr Institute for Machine Learning and Artificial Intelligence, Germany.

Corresponding author: Xieyuanli Chen (xieyuanli.chen@igg.uni-bonn.de)

proposed learning schemes on four common descriptors: SIFT [24], MKD [30], TFeat [5], and HardNet [29]. We train our network only on one dataset and apply it directly to other datasets with different downstream tasks, including visual localization, patch verification, image matching, and image retrieval. The experimental results show that our method consistently outperforms the PCA-based method with a strong generalization ability. Furthermore, using the lower-dimensional descriptors generated by our supervised MLP, we achieve even better performance in downstream tasks than the original higher-dimensional descriptors, especially for hand-crafted ones with faster speed and less memory consumption. Overall, our contributions are as follows:

- We propose and evaluate an MLP-based network for descriptor dimensionality reduction and show its superiority over PCA on multiple descriptors in various tasks;
- We demonstrate that the lighter descriptors by our supervised MLP projection achieve even better performance in downstream tasks than the original descriptors;
- We thoroughly analyze the improvement of using our method for different descriptors on multiple datasets and show a good generalization of our method.

II. Related Work

Various local image descriptors have been proposed over the past decades ranging from hand-crafted ones like SIFT [24], BRIEF [9], and ORB [33] to learned ones like TFeat [5], HardNet [29], and SOSNet [41]. Hand-crafted descriptors are typically based on human insights into which qualities are invariant under certain transformations, such as differential or moment invariants, correlations, and gradients histograms. For example, SIFT by Lowe [24] generates descriptors based on the gradient distribution in the detected patches. BRIEF by Calonder et al. [9] uses simple binary intensity comparisons between pixels in an image patch. More details of classical hand-crafted descriptors can be found in surveys [26], [13]. Benefiting from large-scale datasets [8], learning-based descriptors recently achieved state-of-the-art performances [29], [5], [25], [12], [17], [37]. For example, TFeat by Balntas et al. [5] uses a CNN with hard-negative mining by anchor swap in the triplet loss to compute descriptors. In contrast, Tian et al. [40] propose L2-Net, adding different error terms in the loss function to improve the distinctiveness of the descriptors. HardNet by Mishchuk et al. [29] also uses a simple triplet margin loss for hard negative mining which outperforms other descriptors with an advanced sampling procedure. Both, state-of-the-art hand-crafted and learning-based descriptors are often high dimensional.

Dimensionality reduction techniques can be used to shorten the dimensionalities of feature descriptors. Traditional dimensionality reduction usually refers to reducing the dimension of data while keeping as much information as possible. Classical examples are backward elimination [27], forward selection [27], and random forests [7]. Another type is to find a combination of new features to describe the data. For example. linear dimensionality reduction methods include PCA [19], factor analysis [44], and linear discriminant analysis [2], and non-linear methods including Kernel PCA [36], t-distributed stochastic neighbor embedding (t-SNE) [43], and isometric mapping [32]. Among them, PCA [19] has been widely used for dimensionality reduction of image patch-based descriptors. For example, Gil et al. [15] and Valenzuela et al. [16] use PCA to reduce the dimensionality of SIFT and SURF descriptors. Ke et al. [20] apply PCA to the patches instead of using smoothed weighted histograms in SIFT. There are few works using neural networks for dimensionality reduction. The work most related to ours is the one by Loquercio et al. [23], which uses a supervised method to train a linear projection. However, they work on hand-crafted descriptors like FREAK [1] and focus only on the visual localization task. Different from them, we thoroughly analyze MLP-based non-linear projections using unsupervised, self-supervised, and supervised training schemes on multiple downstream tasks. Moreover, our method works on both hand-crafted descriptors such as SIFT [24] and MKD [30], and learned ones such as TFeat [5] and HardNet [29], and shows a strong generalization ability.

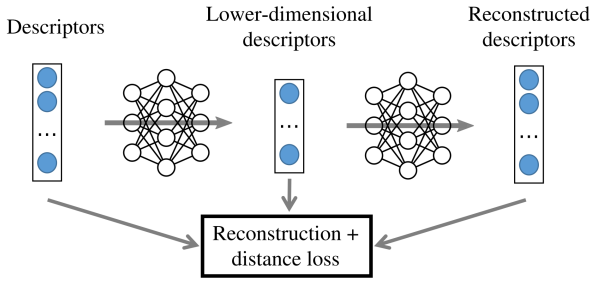

Fig. 2: The pipeline of using an auto-encoder for unsupervised dimensionality reduction. It consists of an encoder and a decoder. The encoder maps the original descriptors to lower-dimensional descriptors. The decoder tries to reconstruct the original descriptors from the projected lower-dimensional descriptors.

III. Methodology

We aim to reduce the dimensionality of local feature descriptors using an MLP network. To understand the ability of the MLP-based method in this task, we investigate three learning schemes, including unsupervised, self-supervised, and supervised methods. This section will introduce the principle of each approach in detail.

A. Unsupervised Reduction

Auto-encoders [3] are unsupervised learning techniques for dimensionality reduction. Here, we use it to see whether an MLP-based network can learn a good projection from high to low dimensionality in an unsupervised way. To achieve fast and lightweight dimensionality reduction, we build our auto-encoder using MLPs with no more than two hidden layers as shown in Fig. 2. Our auto-encoder consists of a symmetric encoder and decoder. The encoder projects the input feature into lower-dimensional embeddings, while the decoder reconstructs the original input from the lower-dimensional embeddings. By checking the consistency between inputs and outputs, the auto-encoder learns how to

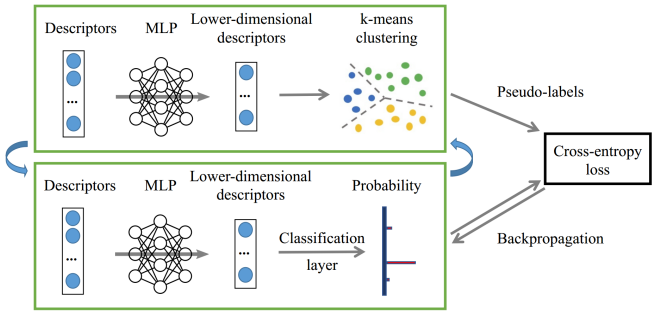

Fig. 3: Pipeline of our self-supervised method. We iteratively cluster descriptors and use the clustering assignments as pseudo-labels to train the network.

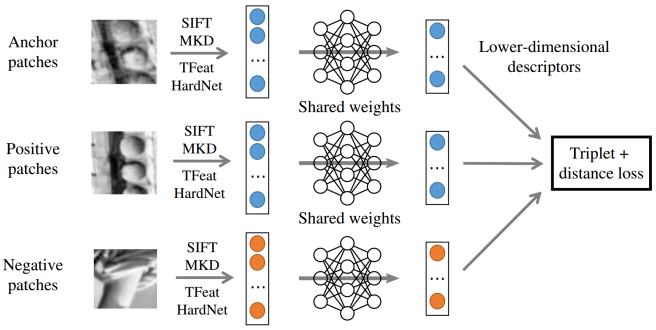

Fig. 4: Pipeline of the supervised method. A training sample includes an anchor with positive and negative patches. The extracted descriptors are fed into MLPs to get lower-dimensional descriptors and a triplet loss is applied to supervise the MLPs.

extract lower-dimensional descriptors from the original ones without labels. We apply the consistency constraints by minimizing the reconstruction loss between inputs and outputs. The reconstruction loss $\mathcal{L}^R$ measures the differences between the input descriptors $\{\boldsymbol{x}_i\}_{i=1}^N$ in a training mini-batch and the reconstructed output descriptors $\{\boldsymbol{x}'_i\}_{i=1}^N$ as

$$\mathcal{L}^R = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{x}'_i\|_2 \,. \tag{1}$$

We propose an additional distance loss $\mathcal{L}^D$ for HardNet. The distance loss calculates the difference between the distance of the original high-dimensional descriptors and the distance of the lower-dimensional descriptors in the embedding space. Given two descriptors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, their corresponding lower-dimensional descriptors in the embedding space are $\hat{\boldsymbol{x}}_i$ and $\hat{\boldsymbol{x}}_j$. The loss $\mathcal{L}^D$ is to make the ℓ_2 distance between the lower-dimensional descriptors $d\left(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j\right)$ as similar as possible compared to that between the original descriptors $d\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$

$$\mathcal{L}^D = \frac{1}{N(N-1)}\sqrt{\sum_{i=1}^{N}\sum_{j\neq i}\left(d\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) - d\left(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j\right)\right)^2}, \tag{2}$$

where $d\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$ is the ℓ_2 distance and the loss is calculated on all different pairs in a mini-batch. The final loss for HardNet is a weighted sum of the reconstruction and distance losses

$$\mathcal{L} = \mathcal{L}^R + \alpha\mathcal{L}^D, \tag{3}$$

more explanation about this design is given in Sec. V.

Note that we need no other information but image patches to train our auto-encoder. Moreover, the non-linearity of the auto-encoder allows it to learn better projection than PCA, thus generating better lower-dimensional descriptors. This will be shown in the experimental evaluation.

B. Self-Supervised Reduction

Unlike unsupervised methods, self-supervised methods usually use traditional heuristic-based methods to generate pseudo-labels and guide networks to learn certain tasks. It combines human priors with learning-based methods to achieve good performance. Inspired by deep feature clustering [10], [22], we propose a self-supervised method for MLP-based dimensionality reduction. The main idea is to apply k-means clustering with descriptors and use the clustering assignments as pseudo-labels, i.e., descriptors in the same cluster considered to have the same label. A classification layer [11] is added during training to guide our MLPs learning to generate similar lower-dimensional descriptors if their high-dimensional ones have the same pseudo-label. In the first training epoch, we cluster the original descriptors into different groups and use them as labels to supervise our MLPs. From that on, we use the clusters of the lower-dimensional descriptors from the previous epoch as supervision. Given a set of descriptors extracted from image patches, k-means clustering generates a centroid matrix $\boldsymbol{C}$ and the clustering assignments $\boldsymbol{y}_i$ for each descriptor $\boldsymbol{x}_i$ by solving

$$\min_{C\in\mathbb{R}^{d\times k}} \frac{1}{N}\sum_{i=1}^{N} \min_{\boldsymbol{y}_i\in\{0,1\}^k} \|\boldsymbol{x}_i - \boldsymbol{C}\boldsymbol{y}_i\|_2^2 \;,\; \text{s.t. } \boldsymbol{y}_i^\top \mathbf{1}_k = 1, \tag{4}$$

where $\mathbf{1}_k$ is a vector whose all elements are 1 and $\boldsymbol{y}_i$ is a one-hot vector.

The training loss used for our self-supervised method is the standard cross-entropy loss with the pseudo-labels as targets.

C. Supervised Reduction

We next introduce our supervised method for dimensionality reduction. As shown in Fig. 4, we use ground truth patch labels together with the triplet loss to train the MLP. A batch of matching patches is denoted as $\{a_i, p_i\}_{i=1\ldots N}$, where a stands for the anchor patch and p for the positive patch. The non-matching negative patches $\{n_i\}_{i=1\ldots N}$ are sampled by the hardest-within-batch strategy as introduced by Mishchuk et al. [29]. For all training patches, we first use existing methods to generate higher-dimensional descriptors and feed them to our MLPs to generate low-dimensional descriptors where the triplet margin loss is applied

$$\mathcal{L}^T = \frac{1}{N}\sum_{i=1}^{N} \max\left(0, m + d(f(a_i), f(p_i)) - d(f(a_i), f(n_i))\right), \tag{5}$$

where f is the MLP-based projection. The main idea is to learn an MLP-based projection f such that the distance

between the anchor and positive descriptors $d(f(a_i), f(p_i))$ is smaller than that between the anchor and negative descriptors $d(f(a_i), f(n_i))$ with a margin m in the lower-dimensional embedding space. We also add a distance loss term for HardNet as used in the auto-encoder to target for the similarity of the distance between input descriptors and embedding descriptors in a batch. The final loss for HardNet is a weighted sum of the triplet margin and distance losses

$$\mathcal{L} = \mathcal{L}^T + \beta\mathcal{L}^D. \quad (6)$$

D. Training and Parameters

We use the UBC Phototour Liberty dataset [8] for training. After training our networks on this dataset, we apply the trained model to other tasks and datasets without fine-tuning.

For the auto-encoder, we train the network for 5 epochs using Adam [21] with a learning rate of 0.001 and a batch size of 1024. We choose the distance loss weighting factor $\alpha = 0.1$ for HardNet. For the self-supervised method, we train the network for 200 epochs using Adam with a learning rate of 0.001 and a batch size of 256. The number of cluster for k-means is 100 000. The clustering is repeated and the classification layer is re-initialized every 10 epochs. For the supervised method, we choose the margin $m = 1$ and train the network for 10 epochs using Adam with a learning rate of 0.001 and a batch size of 1024. The learning rate is linearly decayed to zero within 10 epochs. We choose the weighting of the distance loss $\beta = 3$ for HardNet. We use ReLU followed by batch normalization [18] after each linear layer except the last one. The embeddings are ℓ_2-normalized. For hand-crafted descriptors, we use two hidden layers for our MLPs, while for learning-based ones we use only one hidden layer. More detailed parameters and network architectures for each method can be found in our open-source implementation. For PCA as the baseline, we use the implementation from scikit-learn [31].

IV. Experiments

We present our experiments to show the capabilities of our MLP-based methods for different tasks. We choose SIFT [24], MKD [30], TFeat [5], and HardNet [29] as the base descriptors. Their original dimensions are all 128. We convert these descriptors to lower dimensions of 64, 32, 24, and 16 and apply them on three publicly available datasets, HPatches [4], Aachen Day-Night v1.1 [42], and InLoc [39] for different downstream tasks, including visual localization, patch verification, image matching, and patch retrieval. We name our unsupervised method 'Ours-US', the self-supervised method 'Ours-SS', and the supervised method 'Ours-SV' in all experiments.

A. Visual Localization

In the first experiment, we evaluate how well our reduced features perform in robot visual localization tasks. We analyze different methods using two challenging localization datasets, with severe illumination changes and complex indoor scenes. We use the hierarchical localization toolbox [34] to achieve visual localization, but replace the feature extractors with generated lower-dimensional descriptors.

	Aachen Day-Night v1.1		**InLoc**	
	day	night	duc1	duc2
distance [m]	0.25 / 0.5 / 5.0	0.5 / 1.0 / 5.0	0.25 / 0.5 / 1.0	0.25 / 0.5 / 1.0
orient. [deg]	2 / 5 / 10	2 / 5 / 10	N / A	N / A
SIFT	88.1 / 94.7 / 98.4	64.9 / 77.5 / 92.1	31.3 / **46.0** / 56.1	21.4 / 33.6 / 43.5
PCA-16	84.5 / 90.7 / 95.9	**38.2** / **47.1** / 58.6	19.7 / 32.3 / 37.9	11.5 / 20.6 / 25.2
Ours-US-16	84.5 / **92.4** / 96.0	35.6 / 45.0 / **60.7**	22.2 / 33.8 / 39.4	13.0 / 20.6 / 24.4
Ours-SS-16	85.1 / 90.9 / 95.6	34.0 / 41.9 / 57.6	23.7 / 35.4 / 41.4	**16.0** / 22.1 / 26.0
Ours-SV-16	**85.9** / 91.5 / **96.2**	34.6 / 46.1 / 58.6	**24.2** / **36.9** / **43.9**	15.3 / **26.7** / **30.5**
PCA-24	86.2 / 93.2 / **97.5**	50.3 / 60.7 / 77.5	24.2 / 34.8 / 46.5	15.3 / 26.0 / 29.8
Ours-US-24	86.8 / 93.3 / 97.2	50.3 / **64.4** / **78.5**	24.2 / 39.9 / 48.5	**21.4** / 29.8 / 35.9
Ours-SS-24	87.7 / 93.4 / 97.3	48.7 / 59.7 / 74.9	25.8 / 42.4 / 48.5	17.6 / 26.7 / 32.8
Ours-SV-24	**87.9** / **93.8** / 97.3	**51.3** / 62.8 / **78.5**	**30.8** / **43.4** / **52.0**	**21.4** / **34.4** / **41.2**
PCA-32	87.1 / 93.0 / 97.6	56.5 / 69.1 / 80.6	28.3 / 40.9 / 53.5	21.4 / 29.0 / 35.9
Ours-US-32	87.0 / 94.1 / 97.8	**58.1** / 71.2 / **83.2**	25.8 / 40.9 / 52.5	19.8 / 32.1 / 37.4
Ours-SS-32	87.9 / 93.8 / 98.1	57.1 / 66.0 / 82.7	**30.8** / **43.9** / **55.6**	24.4 / 32.8 / 38.2
Ours-SV-32	**88.7** / **94.4** / **98.3**	**58.1** / **71.7** / 82.7	29.8 / 42.4 / 54.5	**27.5** / **38.9** / **44.3**
PCA-64	87.1 / 94.3 / 98.2	60.2 / 74.3 / 88.5	28.8 / 41.4 / 53.0	19.8 / 31.3 / 40.5
Ours-US-64	87.6 / **95.3** / 98.5	**66.0** / **79.6** / 90.6	31.3 / 43.9 / 57.1	**26.7** / 37.4 / 45.0
Ours-SS-64	87.9 / 94.9 / 98.2	60.2 / 75.9 / 89.0	30.8 / 43.4 / 54.5	23.7 / **39.7** / **45.8**
Ours-SV-64	**89.1** / 94.8 / **98.8**	63.4 / **79.6** / **92.7**	**33.3** / **46.0** / **57.6**	26.0 / **39.7** / **45.8**

TABLE I: Evaluation of the localization on the Aachen Day-Night v1.1 and InLoc datasets with SIFT [24] features. We report the recall [%] at different distances and orientation thresholds. The overall best results are in **red** and the best results with the same low dimension are in **blue**.

	Aachen Day-Night v1.1		**InLoc**	
	day	night	duc1	duc2
distance [m]	0.25 / 0.5 / 5.0	0.5 / 1.0 / 5.0	0.25 / 0.5 / 1.0	0.25 / 0.5 / 1.0
orient. [deg]	2 / 5 / 10	2 / 5 / 10	N / A	N / A
MKD	88.7 / 95.1 / 98.8	**67.0** / 79.6 / **92.7**	**33.3** / **50.5** / **65.2**	26.0 / 40.5 / 49.6
PCA-16	**85.7** / 92.5 / 96.1	37.2 / **45.0** / **56.0**	23.7 / 35.4 / 42.9	17.6 / 25.2 / 28.2
Ours-US-16	84.6 / 91.3 / 95.5	34.0 / 41.4 / 52.4	22.2 / 32.3 / 39.4	14.5 / 19.8 / 25.2
Ours-SS-16	85.0 / 92.2 / 96.5	36.6 / 44.0 / 55.5	**24.2** / 34.8 / 43.9	14.5 / 26.0 / 31.3
Ours-SV-16	**85.7** / **93.0** / **96.8**	**37.7** / 42.9 / 53.9	23.2 / **35.9** / **45.5**	**21.4** / **28.2** / **32.8**
PCA-24	87.4 / 93.8 / 97.5	49.2 / 59.2 / 70.7	26.8 / 37.9 / 48.0	20.6 / 31.3 / 37.4
Ours-US-24	87.6 / 93.8 / 97.2	47.6 / 60.2 / 73.8	24.7 / 38.4 / 49.0	18.3 / 27.5 / 35.1
Ours-SS-24	**88.0** / **94.1** / **97.9**	51.3 / **64.4** / **76.4**	27.8 / **40.4** / 51.5	18.3 / 30.5 / 38.2
Ours-SV-24	87.5 / 93.7 / 97.3	**53.9** / 63.9 / 75.4	**29.3** / **40.4** / **53.5**	**22.9** / **35.9** / **39.7**
PCA-32	87.6 / 93.9 / 98.2	52.4 / 67.0 / 81.7	30.8 / 42.4 / 51.5	26.0 / 36.6 / 43.5
Ours-US-32	87.6 / 94.2 / 98.4	59.2 / 71.2 / **85.9**	27.8 / 40.9 / 55.6	27.5 / 37.4 / 42.0
Ours-SS-32	**88.6** / **94.7** / 98.2	56.0 / 70.7 / 83.2	30.3 / **44.4** / **57.6**	**28.2** / **42.7** / **47.3**
Ours-SV-32	88.3 / 94.5 / **98.7**	**60.7** / **73.8** / **85.9**	**31.3** / 43.4 / 56.6	26.7 / 38.9 / 46.6
PCA-64	88.1 / 94.9 / 98.5	62.3 / **80.6** / 92.1	**31.3** / 47.5 / 61.1	26.0 / 41.2 / 48.1
Ours-US-64	88.6 / 94.5 / 98.5	**65.4** / **80.6** / **92.7**	30.8 / 47.5 / 59.6	28.2 / 42.0 / 48.1
Ours-SS-64	88.8 / **95.6** / **98.9**	62.3 / 79.6 / 92.1	**31.3** / 44.9 / 58.1	**30.5** / 44.3 / 50.4
Ours-SV-64	**89.0** / 95.1 / 98.5	63.4 / 79.6 / 90.1	**31.3** / **49.5** / **61.6**	28.2 / **45.0** / **51.9**

TABLE II: Evaluation of the localization on the Aachen Day-Night v1.1 and InLoc datasets with MKD [30] features.

For the first day-night localization challenge, we evaluate different dimensionality reduction methods on the Aachen Day-Night v1.1 dataset [42]. It contains 6 697 day-time database images from an old European town and 1 015 queries (824 taken in day and 191 in night conditions). We use the code and evaluation protocol from [42] and report the percentage of day-night queries localized within a given error bound on the estimated camera position and orientation. For the complex indoor localization challenge, we exploit the InLoc dataset [39]. It is a challenging indoor localization dataset with large differences in viewpoint and illumination between the query and database images. We also use the code and evaluation protocol from [42] and report the percentage of queries localized within a given error bound on the estimated camera position.

Tables I to IV report the quantitative localization results and Fig. 5 shows the qualitative results. For all four descriptors, our MLP-based method with auto-encoder, self-supervised, and supervised learning schemes perform better

	Aachen Day-Night v1.1		InLoc	
	day	night	duc1	duc2
distance [m]	0.25 / 0.5 / 5.0	0.5 / 1.0 / 5.0	0.25 / 0.5 / 1.0	0.25 / 0.5 / 1.0
orient. [deg]	2 / 5 / 10	2 / 5 / 10	N / A	N / A
TFeat	87.4 / 93.9 / 98.2	53.4 / 72.8 / 83.8	32.8 / **44.9** / 55.6	24.4 / 41.2 / 45.8
PCA-16	82.6 / 89.4 / 94.7	30.4 / 37.7 / **49.7**	20.7 / 29.8 / 39.9	**16.0** / **24.4** / **29.8**
Ours-US-16	**84.5** / **91.1** / 95.5	30.4 / **39.8** / **49.7**	21.2 / 31.3 / 39.9	13.7 / 19.1 / 24.4
Ours-SS-16	84.0 / 89.9 / 95.4	**31.4** / 36.6 / 47.1	21.2 / 30.3 / 37.9	10.7 / 19.8 / 22.1
Ours-SV-16	83.9 / 90.8 / **95.6**	28.8 / 34.0 / 47.1	**21.7** / **32.3** / **41.9**	14.5 / 20.6 / 24.4
PCA-24	86.3 / 93.2 / 96.6	**42.9** / 55.0 / 67.0	23.7 / 38.9 / 49.0	20.6 / 30.5 / 35.1
Ours-US-24	86.3 / **93.8** / **97.1**	42.4 / 55.0 / 63.4	26.3 / 37.9 / 48.0	19.1 / 31.3 / 35.9
Ours-SS-24	**86.8** / 92.7 / 96.4	36.6 / 46.6 / **67.5**	27.3 / 38.9 / 48.0	**22.9** / **32.1** / **36.6**
Ours-SV-24	86.0 / 92.5 / 96.8	**42.9** / **56.0** / **67.5**	**27.8** / **39.4** / **50.5**	19.1 / 28.2 / 34.4
PCA-32	87.3 / 93.7 / 97.2	47.1 / 61.8 / 71.7	25.8 / 38.9 / 52.0	22.9 / 33.6 / **42.7**
Ours-US-32	87.6 / **94.3** / **98.1**	**48.2** / 61.3 / **75.4**	**30.3** / 40.9 / **53.0**	22.1 / **35.9** / 40.5
Ours-SS-32	87.3 / 93.1 / 97.7	45.5 / 59.2 / 73.3	**30.3** / 40.4 / 50.5	22.9 / 35.1 / 41.2
Ours-SV-32	**88.1** / 94.2 / **98.1**	**48.2** / **62.3** / 72.8	**30.3** / **41.4** / 51.5	**23.7** / 35.1 / 39.7
PCA-64	88.2 / **94.4** / 98.2	52.9 / 70.2 / 79.6	29.3 / 43.4 / 54.0	28.2 / 38.9 / 46.6
Ours-US-64	87.6 / 93.9 / 98.3	55.0 / **73.3** / **84.8**	29.3 / 41.9 / 58.6	26.0 / 39.7 / **50.4**
Ours-SS-64	**89.0** / 93.9 / **98.7**	**56.5** / 69.1 / 82.7	**33.3** / **44.4** / **60.1**	23.7 / 39.7 / 46.6
Ours-SV-64	87.0 / **94.4** / 98.4	55.0 / 71.2 / **84.8**	32.3 / **44.4** / 57.1	**32.1** / **44.3** / **50.4**

TABLE III: Evaluation of the localization on the Aachen Day-Night v1.1 and InLoc datasets with TFeat [5] features.

	Aachen Day-Night v1.1		InLoc	
	day	night	duc1	duc2
distance [m]	0.25 / 0.5 / 5.0	0.5 / 1.0 / 5.0	0.25 / 0.5 / 1.0	0.25 / 0.5 / 1.0
orient. [deg]	2 / 5 / 10	2 / 5 / 10	N / A	N / A
HardNet	88.8 / 95.6 / **99.3**	**63.9** / **81.2** / 92.7	37.9 / 56.6 / 70.7	31.3 / 44.3 / 53.4
PCA-16	**84.7** / **91.6** / **96.2**	27.2 / 34.6 / 46.1	21.7 / **36.4** / **43.9**	**15.3** / **24.4** / **29.0**
Ours-US-16	84.1 / 91.1 / 95.0	29.8 / 34.0 / 45.0	20.7 / 33.3 / 40.4	12.2 / 19.8 / 24.4
Ours-SS-16	84.0 / 90.7 / 96.0	28.3 / 35.1 / 42.4	22.7 / 32.8 / 41.9	12.2 / 19.8 / 22.9
Ours-SV-16	84.0 / 91.3 / 95.8	**30.4** / **36.1** / **47.6**	**24.7** / 34.8 / 42.4	13.0 / 22.9 / 26.7
PCA-24	86.9 / 93.4 / **98.3**	44.5 / 55.0 / 69.6	29.3 / **42.9** / 53.0	22.1 / **35.9** / **38.9**
Ours-US-24	**87.4** / **94.1** / 97.2	40.8 / **58.6** / 69.6	28.3 / 38.9 / 52.5	18.3 / 29.0 / 35.9
Ours-SS-24	86.4 / 93.6 / 97.6	46.1 / 57.1 / 66.5	**29.8** / 41.4 / 50.0	**22.9** / 32.8 / 35.9
Ours-SV-24	**87.4** / **94.1** / 97.8	**48.7** / **58.6** / **71.2**	25.8 / **42.9** / **55.6**	21.4 / 31.3 / 36.6
PCA-32	88.0 / 94.8 / **98.9**	**55.0** / **69.6** / 82.7	32.3 / 49.0 / 60.1	**28.2** / 38.9 / 44.3
Ours-US-32	87.9 / 94.9 / 98.7	53.9 / 65.4 / 81.2	**33.8** / 48.5 / 59.1	21.4 / 37.4 / 46.6
Ours-SS-32	**88.6** / 94.8 / 98.3	53.9 / 68.1 / 79.6	33.3 / 48.5 / 59.6	25.2 / 38.2 / 45.0
Ours-SV-32	88.3 / **95.1** / **98.9**	**55.0** / 69.1 / **84.8**	33.3 / **51.0** / **61.1**	**28.2** / **42.0** / **47.3**
PCA-64	88.7 / 95.6 / **99.3**	60.7 / 78.5 / **93.2**	35.4 / 56.1 / 68.2	30.5 / 45.0 / 53.4
Ours-US-64	88.8 / 95.5 / 99.2	57.6 / **80.1** / 92.1	**40.9** / 56.6 / **71.2**	**32.1** / 45.8 / 51.9
Ours-SS-64	89.1 / 95.3 / 99.2	61.3 / 79.6 / 91.1	35.9 / 53.0 / 66.7	31.3 / 45.8 / **55.7**
Ours-SV-64	**89.2** / **95.9** / 99.2	**62.8** / 79.1 / 92.1	38.9 / **58.6** / 70.2	30.5 / **48.1** / 54.2

TABLE IV: Evaluation of the localization on the Aachen and InLoc datasets with HardNet [29] features.

in terms of visual localization than PCA and even the original descriptors for most queries and lower dimensions. This indicates that the learned lower-dimensional descriptors are more distinctive and invariant under challenging environments, which improves the visual localization performance. Besides, the degree of improvement for hand-crafted descriptors is larger than the learned ones. We will analyze and discuss this phenomenon in Sec. V.

B. Applications on Patch Verification, Image Matching, and Patch Retrieval

This experiment shows more robotics applications using our dimensionality-reduced features, including the patch pair verification, image matching, and patch retrieval tasks on the HPatches dataset [4]. There are 116 sequences with over 1.5 million patches in this dataset. 59 sequences of them show significant viewpoint changes and 57 sequences have significant illumination changes. The patches are divided into three groups: easy, hard, and tough, based on the levels of geometric noise. Evaluation results with SIFT is shown in Fig. 6. Note that, all models of our MLP are pre-trained *only* on Liberty sequence of UBC Phototour and we test them directly on HPatches dataset in a zero-shot fashion to show the good generalization of our proposed MLP-based method.

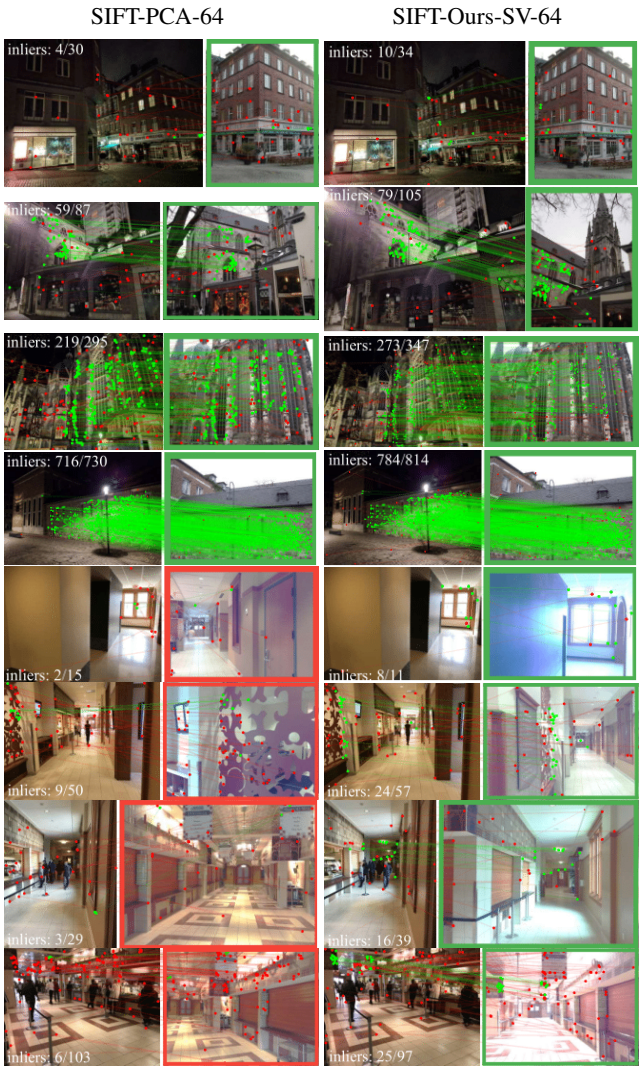

Fig. 5: Localization with SIFT-PCA-64 and SIFT-Ours-SV-64 on Aachen Day-Night v1.1 and InLoc. For each image pair, the left image is the query and the right image is the retrieved database image with the most inlier matches, as returned by PnP+RANSAC. Red lines represent wrong feature matches and red boxes represent wrong localization results. Green lines and boxes are correct. Best view in color and zoom in.

As can be seen from Fig. 6, our methods with supervised and self-supervised learning schemes perform better than PCA and auto-encoder for all three tasks and all lower dimensions with a large margin. Besides, we also observe that the 64-dimensional descriptors generated by our method even outperform the original 128-dimensional descriptors. Even the learned 24-dimensional descriptors have the on par performances as the original 128-dimensional SIFT descriptors, which shows the superiority of our proposed learned MLP-based dimensionality reduction. More explanation will be given in Sec. V.

C. Ablation Study

In this section, we perform several experiments to provide a more in-depth analysis of how each component in MLP contributes to the final performance. We choose different numbers of hidden layers (0, 1, and 2) with different sizes of 96, 128, 256, 512, and 1 024. We make evaluations on

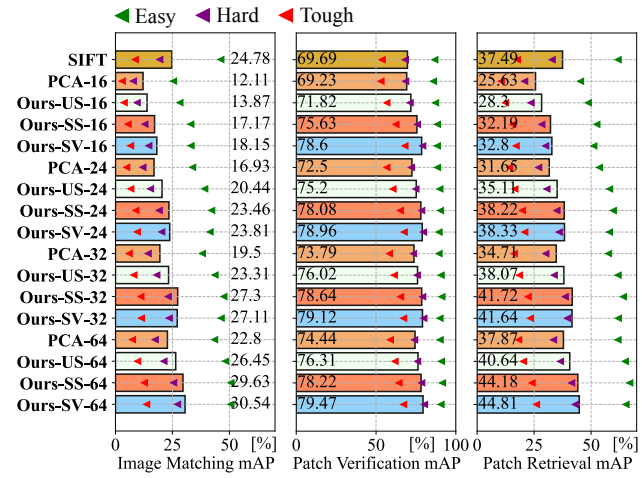

Fig. 6: Verification, matching, and retrieval results of SIFT on test set 'a' of HPatches dataset. None of the MLPs is trained on HPatches. Different colors represent different difficulties and the numbers are the average mAP values.

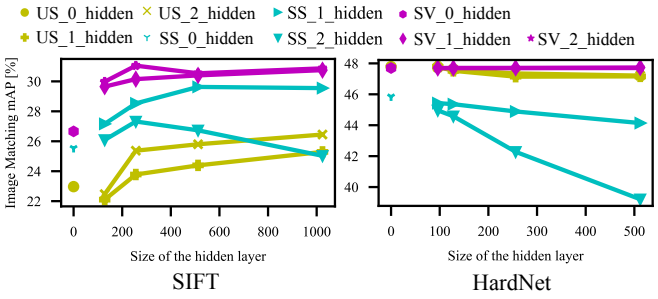

Fig. 7: Impact of the number and the size of hidden layers for image matching mAP on Hatches dataset. For SIFT, the more number and size of hidden layers, the better the results are for most cases. For HardNet, the trend is opposite.

the matching task of the HPatches benchmark. We select the lower dimension of 64 for all experiments, and Fig. 7 shows the ablation study results with SIFT and HardNet. We can see that the more number and size of hidden layers, the better the results for the hand-crafted descriptor. However, for the learned descriptor, more hidden layers do not help.

D. Runtime and Memory Evaluation

For each image patch, our MLP-based method adds less than 0.002 ms extra computational time compared to calculating the original descriptors, which can be basically ignored. Meanwhile, the lower-dimensional descriptors can save the memory 2, 4, 5, and 8 times respectively for the 64, 32, 24, and 16-dimensional descriptors, which shows the advantages of the proposed MLP-based method.

V. Discussion

From the above experiments, we find that our learned MLP projections using unsupervised, supervised, and self-supervised methods achieve better performances than reduction using PCA. Furthermore, our features are partially even better than the original 128-dimensional ones. The degree of improvement for different descriptors is different. A key result is that the improvement for hand-crafted descriptors, like SIFT and MKD, is larger than those of learned ones, like TFeat and HardNet. Moreover, the performance with hand-crafted descriptors benefits from complex network architectures, while for learned ones, simple network architectures achieve better results.

We interpret the results as follows. For SIFT and MKD, the descriptor space is not optimized for the ℓ_2 metric because of its hand-crafted design, and there are overlaps between non-matching patches. When applying PCA directly on them, matching and non-matching patches will still overlap. However, after learning a more discriminative representation using triplet loss, the projected descriptor space will be more distinctive for the lower-dimensional descriptors. This might also be why our method's performance on hand-crafted descriptors benefits from the complex network architecture - more parameters are needed to rearrange the descriptor space. This is also the reason that we do not need additional distance loss for hand-crafted ones.

For learning-based descriptors, such as HardNet and TFeat, a relatively discriminative descriptor space has already been learned using triplet loss. Similar features are close in the descriptor space, otherwise further apart. Therefore, this kind of distinctive distance information may also be preserved after the PCA projection. This might be the reason why different dimensionality reduction methods perform similarly on learning-based descriptors. However, since the descriptor space is already quite regular, it is very easy for an MLP to overfit. Thus simple architectures obtain better results for dimensionality reduction. We added an additional distance loss for HardNet to restrict the learned descriptor space and avoid overfitting. Since HardNet is trained with more advanced hard-negative mining techniques than TFeat, the performance of TFeat can still be improved significantly using our supervised MLP-based method. Due to the page limitation, we put more results and discussion in our code repository: https://github.com/PRBonn/descriptor-dr.

VI. Conclusion

In this paper, we investigated an MLP-based network with unsupervised, self-supervised, and supervised learning schemes for dimensionality reduction of local feature descriptors. We thoroughly evaluate our method on four descriptors including hand-crafted and learning-based on multiple datasets with various downstream tasks. The experimental results show that our MLP-based projections work better than PCA in challenging tasks, including visual localization, patch verification, image matching, and patch retrieval for most cases. Besides, the 64-dimensional descriptors generated from learning-based projections even outperform the original 128-dimensional descriptors. We also provided ablation studies and analyzed the degree of improvement for different descriptors in terms of the distribution of the descriptor space. Additional memory and runtime experiments show that learned lower-dimensional descriptors can be used for saving memory consumption without adding extra runtime and thus are useful for real-world robotics applications.

REFERENCES

[1] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[2] S. Balakrishnama and A. Ganapathiraju. Linear discriminant analysis-a brief tutorial. *Institute for Signal and information Processing*, 18(1998):1–8, 1998.

[3] P. Baldi. Autoencoders, unsupervised learning, and deep architectures. In *Proc. of the Int. Conf. on Machine Learning (ICML)*, 2012.

[4] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Proc. of British Machine Vision Conference (BMVC)*, 2016.

[6] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2006.

[7] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[8] M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 33(1):43–57, 2011.

[9] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. Brief: Computing a local binary descriptor very fast. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 34(7):1281–1298, 2012.

[10] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2018.

[11] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[12] P. Ebel, A. Mishchuk, K.M. Yi, P. Fua, and E. Trulls. Beyond cartesian representations for local descriptors. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2019.

[13] B. Fan, Z. Wang, and F. Wu. *Local Image Descriptor: Modern Approaches*, volume 108. Springer, 2016.

[14] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[15] A. Gil, O. Reinoso, O. Martínez-Mozos, C. Stachniss, and W. Burgard. Improving Data Association in Vision-based SLAM. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[16] R.E. González Valenzuela, W.R. Schwartz, and H. Pedrini. Dimensionality reduction through pca over sift and surf descriptors. In *Proc. of the IEEE Intl. Conf. on Cybernetic Intelligent Systems (CIS)*, 2012.

[17] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A.C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[18] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. of the Int. Conf. on Machine Learning (ICML)*, pages 448–456. PMLR, 2015.

[19] I.T. Jolliffe. *Principal Component Analysis*. Springer, 2002.

[20] Y. Ke and R. Sukthankar. Pca-sift: a more distinctive representation for local image descriptors. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[21] D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] M. Larsson, E. Stenborg, C. Toft, L. Hammarstrand, T. Sattler, and F. Kahl. Fine-grained segmentation networks: Self-supervised segmentation for improved long-term visual localization. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2019.

[23] A. Loquercio, M. Dymczyk, B. Zeisl, S. Lynen, I. Gilitschenski, and R. Siegwart. Efficient descriptor learning for large scale localization. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.

[24] D.G. Lowe. Distinctive image features from scale-invariant keypoints. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2004.

[25] Z. Luo, T. Shen, L. Zhou, S. Zhu, R. Zhang, Y. Yao, T. Fang, and L. Quan. Geodesc: Learning local descriptors by integrating geometry constraints. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2018.

[26] J. Ma, X. Jiang, A. Fan, J. Jiang, and J. Yan. Image matching from handcrafted to deep features: A survey. *Intl. Journal of Computer Vision (IJCV)*, 129(1):23–79, 2021.

[27] K.Z. Mao. Orthogonal forward selection and backward elimination algorithms for feature subset selection. *IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1):629–634, 2004.

[28] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2001.

[29] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, 2017.

[30] A. Mukundan, G. Tolias, A. Bursuc, H. Jégou, and O. Chum. Understanding and improving kernel local descriptors. *Intl. Journal of Computer Vision (IJCV)*, 127:1723–1737, 2019.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[32] T.M. Rassias. Properties of isometric mappings. *Journal of Mathematical Analysis and Applications*, 235(1):108–121, 1999.

[33] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2011.

[34] P.E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[35] T. Sattler, B. Leibe, and L. Kobbelt. Efficient amp; effective prioritized matching for large-scale image-based localization. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 39(9):1744–1756, 2017.

[36] B. Schölkopf, A. Smola, and K.R. Müller. Kernel principal component analysis. In *Proc. of the Intl. Conf. on artificial neural networks (ICANN)*, 1997.

[37] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2015.

[38] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 36(8):1573–1585, 2014.

[39] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[40] Y. Tian, B. Fan, F. Wu, et al. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[41] Y. Tian, X. Yu, B. Fan, F. Wu, H. Heijnen, and V. Balntas. Sosnet: Second order similarity regularization for local descriptor learning. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[42] C. Toft, W. Maddern, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, T. Pajdla, F. Kahl, and T. Sattler. Long-term visual localization revisited. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 44(4):2074–2088, 2022.

[43] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[44] M.W. Watkins. Exploratory factor analysis: A guide to best practice. *Journal of Black Psychology*, 44(3):219–246, 2018.

[45] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.