Unsupervised Extraction of Local Image Descriptors via Relative Distance Ranking Loss

Xin Yu*[1], Yurun Tian[2], Fatih Porikli[1], Richard Hartley[1], Hongdong Li[1],
Huub Heijnen[3], Vassileios Balntas[3]
[1] Australian Centre for Robotic Vision, Australian National University
[2] University of Chinese Academy of Science, China
[3] Scape Technologies
{xin.yu, fatih.porikli, richard.hartley, hongdong.li}@anu.edu.au,
yurun.tian@nlpr.ia.ac.cn, {huub,vassileios}@scape.io

Abstract

State-of-the-art supervised local descriptor learning methods heavily rely on accurately labelled patches for training. However, since the process of labelling patches is laborious and inefficient, supervised training is limited by the availability and scale of training datasets. In comparison, unsupervised learning does not require burdensome data labelling; thus it is not restricted to a specific domain. Furthermore, extracting patches from training images involves minimal effort. Nevertheless, most of the existing unsupervised learning based methods are inherently inferior to the handcrafted local descriptors, such as the Scale-Invariant Feature Transform (SIFT).

*In this paper, we aim to leverage unlabelled data to learn descriptors for image patches by a deep convolutional neural network. We introduce a Relative Distance Ranking Loss (RDRL) that measures the deviation of a generated ranking order of patch similarities against a reference one. Specifically, our approach yields a patch similarity ranking based on the learned embedding of a neural network, and the ranking mechanism minimizes the proposed RDRL by mimicking a reference similarity ranking based on a competent handcrafted feature (*i.e.*, SIFT). To our advantage, after the training process, our network is not only able to measure the patch similarity but also able to outperform SIFT by a large margin on several commonly used benchmark datasets as demonstrated in our extensive experiments.*

1. Introduction

Obtaining a robust descriptor for local image patches is an essential task in many computer vision applications including image registration, simultaneous localization and mapping (SLAM) and large-scale 3D reconstruction. To this end, widely popular descriptors, such as scale-invariant feature transform (SIFT) [29] and speed-up robust features (SURF) [5], employ a set of handcrafted spatial filters and ad-hoc operations to interpret local patch patterns into vectorized representations. However, such filters and nonlinear operations are often empirically determined based on human experience. Handcrafted features may not adequately and fully express the useful information available in image patches, thus restricting their performance.

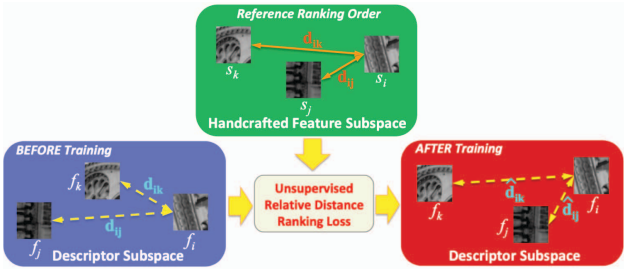


Figure 1. *We illustrate the motivation of our Relative Distance Ranking Loss (RDRL). Handcrafted features are used to establish a reference ranking of patch similarity. Our method mimics this ranking without enforcing any further constraints on the learned distances. Note that all patches are unlabelled.*

To strengthen the representation capacity of handcrafted features, supervised learning methods (in particular, boosting and kernel-based approaches [41, 6, 49, 44, 17]) aim at obtaining more discriminative feature representations by leveraging the available labelled training data. These approaches are built on top of handcrafted features and designed to learn a mapping function (often a nonlinear one) that allows features to be easily separated in high-dimensional spaces. However, deriving effective and computationally feasible mapping functions is not a straightfor-

*Research conducted while Xin and Yurun were interns at Scape Technologies

ward process, and still remains an open problem.

As an alternative, deep supervised learning based methods [18, 47, 30, 58] do not require users to define handcrafted features and nonlinear mapping functions. Instead, they learn image patch representations in an end-to-end fashion via neural networks by imposing a similarity metric on local image patches, which is intended to be invariant to image transformations. Unfortunately, deep supervised learning based methods require a vast amount of labelled data for training. For instance, patches need to be captured in different illumination conditions and views.

Unlike supervised learning, unsupervised methods draw inferences from datasets without relying on labelled data. Such methods have been successfully incorporated into high-level tasks such as clustering [10, 20, 55], data retrieval [38, 35] and image generation [16, 36, 23]. Still, to the best of our knowledge, learning representations for low-level local image patches in an unsupervised fashion has not been thoroughly investigated. Using an unsupervised approach for local patches is also preferable; extracting a large number of patches from images can be done with minimal effort since this operation does not require burdensome manual annotations. State-of-the-art unsupervised learning methods primarily focus on extracting low-level features driven by optimizing either a quantization error loss [27, 13] or a generative adversarial loss [60]. However, these loss functions do not directly evaluate the affinity between image patches. Consequently, their existing applications to patch matching related tasks have not outperformed traditional handcrafted features yet.

In this paper, we introduce a new loss function, called Relative Distance Ranking Loss (**RDRL**), to evaluate the patch similarity directly in the objective function of a convolutional neural network. We first employ a handcrafted feature descriptor (*i.e.*, SIFT) to obtain a relative distance ranking between patches as a reference, such as "patch X is more visually similar to patch Y than X is to Z", where X,Y and Z are randomly chosen unlabelled local patches. These relative rankings coming from the handcrafted features are a suitable indicator of similarity, since they can be considered as a proxy for visual appearance. However, handcrafted features rely on user-defined spatial filters to extract local information. Hence, they are limited by the types and ranges of filters. Our main idea is that by using the first level of granularity that comes from the handcrafted features and a process of learning filters inside a convolutional neural network, we can obtain a better feature descriptor in an unsupervised manner, *i.e.*, without using pairs of positive and negative patches during training.

To achieve this goal, we rank the similarities between the features generated by our network and compare our estimated rankings with the reference rankings as shown in Fig. 1. In other words, our network learns to rank in accordance with the relative distance rankings provided by a handcrafted descriptor so as to generate discriminative features for local image patches. Our method only uses SIFT to establish the reference relative distance rankings, yet it can significantly outperform SIFT on standard benchmarks [54, 2] after training. We conclude that combining the feature extraction power of convolutional networks with our RDRL significantly boosts the performance of the reference handcrafted features without requiring any labelled data.

In addition, we apply a direct binarization to our learned descriptors to derive compact descriptors. As demonstrated in our experiments, the binarized descriptors not only achieve state-of-the-art performance but also retain much shorter code lengths. This phenomenon implies that our network extracts even more discriminative features than the conventional handcrafted features, and the new loss function is suitable for learning local image descriptors in an unsupervised manner.

Overall, the contributions of this paper are in four aspects:

- We present an unsupervised learning method to generate discriminative features for local image patches. Our algorithm achieves 43.87% and 47.83% improvements on patch matching performance over the state-of-the-art handcrafted features (*i.e.*, SIFT) and unsupervised learning based methods on the UBC benchmark, respectively.
- We introduce a novel objective function, Relative Distance Ranking Loss (RDRL), for training our convolutional network. Since RDRL is designed to measure the similarity between local patches directly, our network is suitable for patch matching tasks.
- To the best of our knowledge, our method is the first attempt to learn local descriptor networks by leveraging handcrafted features in an unsupervised fashion.
- More importantly, by employing RDRL our network can outperform the reference handcrafted features. This rather "counter-intuitive" phenomenon has not been noticed or explored by previous unsupervised descriptor learning works, and we believe that our results would motivate other vision tasks.

2. Related Works

2.1. Handcrafted Local Features

The evolution of local descriptors has achieved remarkable progress over the past three decades, including differential filter based [24], moment invariant based [52], and histograms of gradients based features, such as HOG [11], LBP [31], DAISY [54], SIFT [29] and SURF [5]. We refer the readers to the comprehensive literature survey [28].

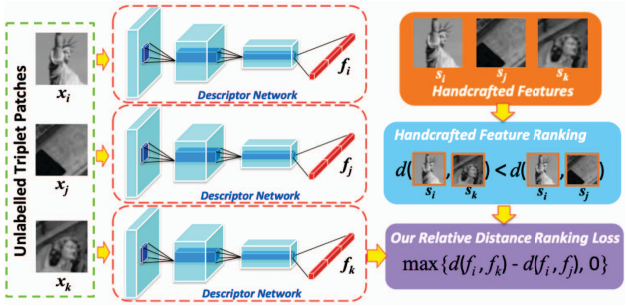


Figure 2. *The pipeline of our proposed network. Unlike most recent descriptor learning methods, our network takes an unlabelled triplet of patches to generate a relative distance ranking, and aims to match the generated distance ranking to the reference ranking which is computed using handcrafted features.*

In order to achieve more compact and efficient descriptors, binary descriptors also gain a great amount of attention. BRIEF [8] exploits randomized intensity comparison to generate binary descriptors. ORB [37] maximizes the variance across training patches by selecting uncorrelated intensity tests, while BRISK [25] optimizes BRIEF by using decision trees. FREAK [1] constructs a cascade of binary strings by comparing image intensities with a retinal sampling pattern.

2.2. Supervised Local Descriptor Learning

To achieve more discriminative features, some works [6, 7, 46, 44, 42] simultaneously minimize intra-class and maximize inter-class distances by exploiting discriminative projections. D-BRIEF [51] adapts the inter and intra class distance objectives to binary descriptors. BinBoost [50] applies boosting to learn a set of binary hash functions while [4] presents an online learned binary descriptor.

Driven by the success of deep neural networks, CNN-based descriptors [15, 43, 48] achieve impressive results by exploiting large-scale labelled data. This demonstrates the power of feature extraction and representation of CNNs. Recently, end-to-end local descriptor learning methods have been developed by employing the architecture of Siamese networks and triplet or contrastive losses [18, 57, 43, 3, 30], while L2Net [47] employs Euclidean distance as a similarity metric to learn descriptors. Nevertheless, their outstanding performance is restricted to the training domains and those methods may be also limited due to lack of sufficient labelled data.

2.3. Unsupervised Local Descriptor Learning

Unlike supervised methods, unsupervised deep learning based methods [27, 14, 38, 13, 60] are less domain-specific and do not need to label any data. Thus, unsupervised learning becomes especially important where labelled data are difficult to obtain, for example, medical imaging and hyperspectral imaging.

Previous work [33] trains a Gaussian Restricted Boltzmann Machine (GRBM) in an unsupervised way and uses the extracted features from the network as local descriptors. Similarly, [34] presents unsupervised patch descriptors based on a convolutional kernel network. However, the network needs to be carefully optimized in a layerwise manner. Deep Hashing (DH) [14] employs a neural network as an encoding function to find a binary representation that minimizes the quantization loss while maximizing the entropy of bit values. Furthermore, DH takes the features of input images extracted by handcrafted descriptors, such as GIST [32], as its inputs. DeepBit [27] replaces handcrafted descriptors with a pretrained VGG network [45] to extract image features, and further improves its performance with data augmentation. In order to reduce quantization losses, DBD-MQ [13] reformulates binarization as a multi-quantization task and solves it by a K-AutoEncoders network. BinGAN [60] employs the framework of generative adversarial networks [16, 36] to learn image features and then binarizes the feature representations of the penultimate layer from its learned discriminator.

Above all, unsupervised learning based methods mainly employ energy based objective functions, generative adversarial losses, or quantization minimization losses to optimize neural networks. However, those losses do not tackle the patch matching problem directly and thus lead to sub-optimal solutions.

3. Proposed Method

We propose an unsupervised local descriptor learning network, which benefits from the advantages of both human expertise and deep convolutional neural networks. Manually designed filters, such as *Gaussian* or *Gabor* filters, are the basis of various handcrafted features (*e.g.*, SIFT or SURF). Due to the simplicity of those filters, the feature extraction ability of handcrafted features has been limited. On the contrary, CNNs demonstrate their powerful feature extraction ability, but it is challenging to design a suitable loss function to learn a network for patch matching tasks in an unsupervised manner. In our method, we employ a CNN to extract features and exploit handcrafted features to provide a reference ranking of patch similarity for optimizing our network. The pipeline of our algorithm is shown in Fig. 2.

3.1. Relative Distance Ranking

In unsupervised descriptor learning, not only label information of patches is unknown but also the number of patch clusters is numerous and there are few samples in each cluster. Thus, clustering patches based on similarity is not suitable. Furthermore, the absolute distance between two unlabelled patches does not provide any clue for training our network, *e.g.*, whether the network should force these two patches closer or not in the feature space. Choosing or

designing a proper distance metric becomes the key to the success of learning a discriminative descriptor.

Motivated by relative distance comparison (RDC) [40], widely used in supervised learning methods (such as triplet loss), we propose a Relative Distance Ranking (RDR) metric based on three patches for our unsupervised learning method. Different from the work [40], where RDC is used to maximize distances of non-matching pairs and minimize distances of matching pairs, our metric only yields a ranking order of patch similarity. Specifically, we randomly choose three patches, *e.g.*, x_i, x_j and x_k, and feed them into the network to obtain their representations, f_i, f_j and f_k. Note that the feature representations have been normalized to unit vectors, and the distance between two patches refers to the distance between the descriptors of those two patches. Thus, we obtain two absolute distance values:

$$d(x_i,x_j)=d(f_i,f_j)=\|\Phi_\theta(x_i)-\Phi_\theta(x_j)\|_2=\|f_i-f_j\|_2,$$
$$d(x_i,x_k)=d(f_i,f_k)=\|\Phi_\theta(x_i)-\Phi_\theta(x_k)\|_2=\|f_i-f_k\|_2,$$

where Φ represents our local descriptor network and θ indicates the parameters of the network. Note that, standard triplet loss (*i.e.*, $\max\{0, \mu + d(x_i,x_j) - d(x_i,x_k)\}$, where μ represents a margin) is not suitable to apply it in our case, for instance, by pulling two randomly chosen patches x_i and x_j closer, while pushing patch x_k further away, because those unlabelled patches might come from either the same or different classes. Instead, we define our RDR metric as:

$$\begin{cases} d(x_i,x_j) < d(x_i,x_k), & \text{if } x_i \text{ is closer to } x_j \text{ than } x_k, \\ d(x_i,x_j) > d(x_i,x_k), & \text{if } x_i \text{ is closer to } x_k \text{ than } x_j. \end{cases} \tag{1}$$

As indicated in Eqn. 1, our RDR only evaluates the relative relationship among three patches instead of the absolute distance between two patches.

3.2. Proposed Relative Distance Ranking Loss

Although RDR alleviates erroneous clustering and provides a metric for objective functions to optimize neural networks, the objective functions still require a reference affinity relationship among three patches x_i, x_j and x_k.

Handcrafted local descriptors encode sophisticated human expertise and are designed for different tasks, such as image registration, retrieval and classification, as well as different domains, like medical imaging and hyperspectral imaging. SIFT, one of robust handcrafted features, has been widely used in many tasks, such as image matching [29], image retrieval [59] and medical image registration [9]. Hence, we use SIFT features to provide our reference RDR between patches.

However, it is possible that SIFT features may also encode two patches from different classes closer in the feature space. Thus, taking inaccurate RDR estimation from SIFT features into account, we impose a margin m on the reference relative distance between $d(s_i,s_j) = \|s_i - s_j\|_2$ and $d(s_i,s_k) = \|s_i - s_k\|_2$, where s_i, s_j and s_k indicate the SIFT features of the patches x_i, x_j and x_k respectively. In other words, $d(s_i,s_k)$ should be larger than $d(s_i,s_j)$ by a margin m, or vice versa. By imposing a margin between the reference relative distance, we obtain a more reliable ranking order from SIFT features. Therefore, our proposed relative distance ranking loss (RDRL) $\mathcal{L}$ is formulated as:

$$\begin{aligned}\mathcal{L}(x_i,x_j,x_k) = \\ \mathcal{I}(d(s_i,s_k)-d(s_i,s_j)-m)\ [d(x_i,x_j)-d(x_i,x_k)]_+ \\ +\mathcal{I}(d(s_i,s_j)-d(s_i,s_k)-m)\ [d(x_i,x_k)-d(x_i,x_j)]_+ ,\end{aligned} \tag{2}$$

where $[\alpha]_+$ represents the hinge loss $\max\{\alpha, 0\}$ and $\mathcal{I}(\cdot)$ is an indicator function, defined by $\mathcal{I}(\alpha) = 1$ if $\alpha > 0$, otherwise $\mathcal{I}(\alpha) = 0$.

According to Eqn. 2, when the RDR generated by our neural network violates the reference RDR output by SIFT, the RDRL will be back-propagated to update the network. Furthermore, if the reference relative distance is smaller than the margin, our method will not use the ranking information of the sampled patches to update our network. Note that the introduced margin m is different from the margin μ in triplet losses [3, 30]. The margin μ forces the distances between inter-class samples to be larger than the distances between intra-class samples, while our margin m is presented to mitigate the impact of erroneous rankings of the handcrafted features.

3.3. Objective Function and Network Architecture

Section 3.2 describes, using a randomly chosen triplet of patches, how to evaluate the loss function in Eqn. 2. In order to make computation more efficient and improve the performance of our network, we employ a strategy of selecting triplet patches in the training phase. Inspired by the mining strategy in [30], we construct a hard triplet for each patch in a batch. Different from the hard mining strategy employed in [30, 58], our hard triplets are selected by the SIFT feature extractor instead of our learned network. The details of our mining strategy and hard triplet selection are further explained in Sec. 3.5. After obtaining the training triplet patches, the final objective of our network is expressed as:

$$\mathcal{L}_\mathcal{T} = \frac{1}{N}\sum_{i,j,k}\mathcal{L}(x_i,x_j,x_k), \tag{3}$$

where N represents the number of patches in a batch, and the triplet patches (x_i, x_j, x_k) refer to a hard triplet.

Similar to supervised learning based methods [47, 30, 58], we also aim to learn a lightweight local descriptor network without leveraging pre-trained networks, such as VGG [45] and ResNet [19]. Since pre-trained models are trained on supervised tasks such as classification, it is hard to tell whether the feature extraction ability of the networks

comes from their original tasks or our proposed RDRL. Thus, we adopt the architecture from [47], which consists of seven convolutional layers and is regularized with Batch Normalization. To prevent from overfitting, we employ a drop-out layer with a drop rate 0.1 before the last convolutional layer. We apply the objective in Eqn. 3 to train our network from scratch with randomly initialized weights.

3.4. Binarizing Local Image Descriptors

Binary local descriptors are desirable for many applications [8, 37, 27, 60], due to the low computational requirements and high memory efficiency for image retrieval and matching. Since batch-normalization [21] is used as the output layer, our descriptors have been normalized to zero mean in every dimension. To achieve binary local descriptors, we can directly binarize the real-valued local descriptors generated by our network. Specifically, we apply the function $\text{sign}(\cdot)$ to the real-valued descriptors and then map the codes from $\{-1, 1\}$ to $\{0, 1\}$. Note that, we do not deliberately design a loss function for optimizing binary descriptors. As suggested in [47], if the real-value descriptors are discriminative enough, their corresponding binary descriptors should be discriminative as well. The performance of binary descriptors also in turn reflects the discriminative ability of real-valued descriptors.

3.5. Training Details

Unlike previous unsupervised methods [27, 13, 60], which use each patch individually to optimize networks, our approach employs unlabelled triplet patches to evaluate RDR on both handcrafted features and deep features extracted by our network. As mentioned in Sec. 3.2, specific triplets of patches might not be used to update our network if their reference rankings do not satisfy the margin constraint. In order to reduce redundant computation, we construct triplet patches by a mining strategy for our RDRL in each training batch. We first extract SIFT features $s_i, i = 1, 2, ..., N$ on the training data in a batch, and then calculate the distance matrix M between every two patches. Given a patch x_i, as an anchor patch, we first choose a patch x_j, which is the most similar patch to the anchor x_i based on M. Another patch x_k, regarded as a hard neighbour, is selected if its distance to the anchor x_i is the smallest one among the distances larger than the distance M_{ij} between s_i and s_j with a margin m. Then we obtain a hard triplet (x_i, x_j, x_k).

Since our objective function and all the layers in the network are differentiable, we employ the Adam optimizer [22] to update the parameters θ of our network with a learning rate 10^{-5} and the decay rates for the first and second moment estimates are set to 0.9 and 0.99 respectively.

4. Experiments

We test our proposed method on three popular patch-based benchmarks: UBC Phototour [54], HPatches [2] and ETH dataset [39]. These benchmarks are used to evaluate the patch matching performance. The inputs to the network are gray-scale patches and are resized to 32×32 pixels. To reduce the impact of illumination changes, we normalize each input patch by subtracting the mean value of its intensities and then dividing by the standard deviation of the intensities.

4.1. UBC Phototour

In UBC Phototour dataset [54], patches are extracted from three image sequences: Liberty, Notredame and Yosemite. Following the standard training/test configuration, one of the sequences is used for training and the other two are used for testing. Note that ground-truth label information is not provided in training. We report patch matching performance in terms of false positive rates at 95% recall (FPR@95).

Since SIFT [29] is used to provide the reference RDR in our loss, we employ SIFT as our baseline method. Four state-of-the-art unsupervised learning based binary descriptors, BinGAN [60], DeepBit [27], DBD-MQ [13] and Boosted SSC [41], are chosen to serve as our baselines. We also compare handcrafted binary descriptors, BRISK [25], BRIEF [8] and ORB [37], with our binarized descriptor Ours_bin. Another widely used real-valued handcrafted feature SURF [5] is also included for comparisons. Moreover, we employ our network architecture to regress real-valued SIFT features, marked as SIFT Reg, as another baseline. Since DeepBit [27] exploits a pretrained VGG network [45] (excluding the classifier part) to extract features, we include pretrained VGG as a baseline. We also retrain BinGAN to achieve its real-valued descriptors, marked as BinGAN† (128 dimension) and BinGAN‡ (256 dimension).

As indicated in Tab. 1, our real-valued descriptors, denoted as Ours, outperform the state-of-the-art unsupervised methods by a large margin of 11.87% on the average FPR@95. Note that, among previous unsupervised methods and handcrafted features, SIFT achieves the lowest errors. Although our binary descriptors are directly binarized from our real-valued descriptors without utilizing any specific binarization regularization, they also attain superior performance. Benefiting from our proposed RDRL, our network is able to extract features from patches and cluster similar patches more closely in the feature space.

4.2. HPatches

HPatches [2] is composed of over 2.5 million patches extracted from 116 image sequences, where the patches contain different viewpoints and illuminations. According to

Table 1. *Quantitative comparisons on the UBC Phototour dataset in terms of false positive rates at 95% true positives (FPR@95) across all the splits of the training and testing configurations. Ours and Ours_bin represent our learned real-valued and binary descriptors by using SIFT to provide reference RDR, respectively. Ours† indicates our learned real-valued descriptors by using our learned network,* i.e., *Ours, to provide reference RDR.*

Methods	Train / Test	Liberty / Notredame	Liberty / Yosemite	Notredame / Yosemite	Notredame / Liberty	Yosemite / Notredame	Yosemite / Liberty	Average FPR@95%
Handcrafted descriptors								
BRISK [25]	512 bits	74.88	73.21	73.21	79.36	74.88	79.36	75.81
BRIEF [8]	256 bits	51.13	52.18	52.18	56.30	51.13	56.30	53.20
ORB [37]	256 bits	42.80	45.10	45.10	50.90	42.80	50.90	46.27
SURF [5]	64 bytes	31.85	44.30	44.30	49.85	31.85	49.85	42.00
SIFT [29]	128 bytes	25.17	27.77	27.77	30.76	25.17	30.76	27.90
Binary unsupervised learning based descriptors								
Boosted SSC [41]	128 bits	72.95	77.99	76.00	70.35	72.20	71.59	73.51
DeepBit [27]	256 bits	26.66	57.61	63.68	32.06	29.60	34.41	40.67
DBD-MQ [13]	256 bits	25.78	57.15	57.24	31.10	27.20	33.11	38.59
BinGAN [60]	128 bits	27.24	50.48	39.44	27.92	32.72	39.44	36.21
BinGAN⋆ [60]	256 bits	23.20	49.48	44.72	**24.44**	21.44	33.64	32.82
Ours_bin	128 bits	**20.96**	**23.20**	**23.23**	27.59	**20.79**	**29.25**	**24.17**
Real-valued unsupervised learning based descriptors								
BinGAN† [60]	128 bytes	27.24	54.56	45.68	24.72	41.76	48.92	40.48
SIFT Reg	128 bytes	39.29	29.45	41.38	51.42	30.29	41.21	38.84
VGG [45]	512 bytes	27.56	59.07	59.07	29.85	27.56	29.85	38.83
BinGAN‡ [60]	256 bytes	24.60	48.12	45.72	21.92	22.72	36.48	33.26
mcRBM [33]	512 bytes	25.10	34.50	33.00	34.00	22.30	31.20	30.02
Ours	128 bytes	13.04	17.09	15.17	19.70	12.15	19.05	16.03
Ours†	128 bytes	**12.56**	**16.22**	**14.92**	**19.65**	**11.70**	**18.92**	**15.66**

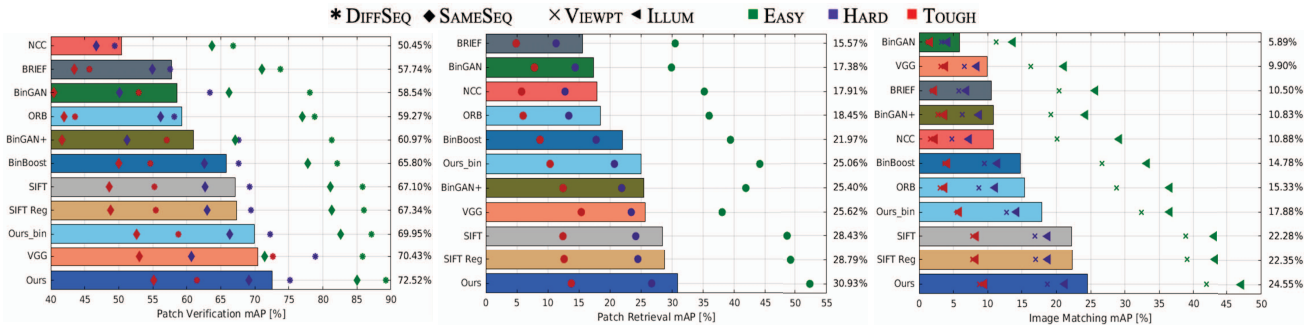


Figure 3. *Results on HPatches dataset [2]. All the results are evaluated on the test set of the "a" split.*

the amount of geometric noise, the test cases are grouped into different levels of difficulty, easy, hard, and tough. Three tasks are evaluated in ascending order of difficulty: patch verification, patch retrieval, and image matching.

In this experiment, we employ four handcrafted features as baselines: SIFT, ORB, BRIEF and Normalized Cross Correlation (NCC) [26]. We also apply unsupervised deep learning based methods BinGAN and its real-valued variant BinGAN† as our baselines. For a fair comparison, we use their 128-dimensional descriptors as baselines since the code length of our learned descriptors is 128. Since the pretrained VGG achieves better results than DeepBit on the UBC Phototour benchmark, we use the pretrained VGG-net for comparison. Notice that the length of the VGG descriptor (512 bytes) is 4 times larger than our descriptors. Moreover, SIFT-Reg serves as another baseline.

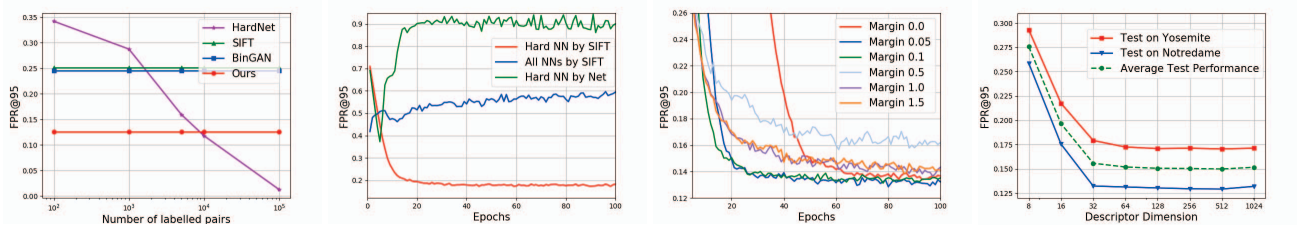
As shown in Fig. 3, our real-valued descriptors outperform the state-of-the-art unsupervised methods for all the three tasks. Furthermore, our method is able to use the shortest code length to achieve the best performance. This also indicates that by exploiting our proposed RDRL our network is able to cluster patches effectively. Additionally, our binary descriptors outperform the unsupervised learning-based and handcrafted binary descriptors.

4.3. ETH Dataset

ETH benchmark [39] focuses on evaluating descriptors for a Structure from Motion (SfM) task. This benchmark investigates the performance of different descriptors in terms of building a 3D model from a set of 2D images. Specifically, the SfM performance of a method is measured by the number of registered images, reconstructed sparse points, image observations, mean track length, mean reprojection

Table 2. Evaluation results on ETH benchmark for SfM. The red color indicates the best performance.

		# Images	# Reg.	# Sparse Pts	# Observ.	Track Length	Reproj. Error	# Inlier Matches
Fountain	SIFT	11	11	15.6K	74.8K	4.77	0.40	138.9K
	TFeat		11	14.2K	67.5K	4.73	0.37	113.9K
	LIFT		11	6.0K	28.2K	4.71	0.58	52.2K
	Ours		11	15.8K	75.7K	4.79	0.41	144.0K
South Building	SIFT	128	128	150K	754K	5.02	0.54	2677K
	TFeat		128	102K	604K	5.91	0.51	1751K
	LIFT		128	42K	233K	5.47	0.73	711K
	Ours		128	153K	767K	5.02	0.54	2728K
Gendarmenmarkt	SIFT	1463	1098	612K	2207K	3.60	0.72	90M
	TFeat		902	280K	1324K	4.72	0.69	15M
	LIFT		959	143K	819K	5.73	0.84	5M
	Ours		1118	641K	2335K	3.63	0.72	90M



(a) Impact of labelled data (b) Impact of mining strategies (c) Impact of margins (d) Performance wrt. dimensions

Figure 4. *Ablation study of our method on UBC Phototour benchmark. For all the cases, we train our network on the Liberty dataset. We either use Notredame for validation and test on Yosemite or vice versa.*

error and inlier matches.

In this experiment, we compare our descriptor with SIFT to demonstrate the effectiveness of our RDRL. We also include the performance of two supervised learning based descriptors provided in [39][1], *i.e.*, TFeat [3] and LIFT [56].

Table 2 indicates the evaluation results of the 3D reconstruction. Our method outperforms other methods in terms of metrics related to the density of the reconstructed 3D model, *i.e.*, the number of registered images, the number of registered sparse points, the number of observations and the number of inlier matches. In most cases, the tracking length of our method is longer than SIFT as well. Furthermore, since the reprojection errors are less than 1 pixel for all descriptors, this metric may not reflect performance differences between descriptors in practice. Overall, by employing RDRL, our method significantly improves the performance of SIFT and is also competitive with supervised methods in the SfM task.

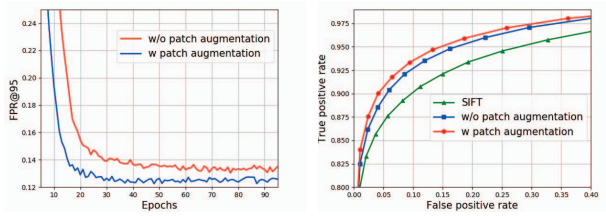
4.4. Discussion

Comparison with Supervised Descriptors: Since our method is trained on unlabelled patches, it would be unfair to compare our method with supervised ones. However, we illustrate how the number of labelled patch pairs affects the performance of the supervised methods in Fig. 4(a). Specifically, we train a supervised method (*i.e.*, Hardnet [30]) on Liberty given different amount of labelled training pairs and test it on Notredame. As shown in Fig. 4(a), when the number of labelled patch pairs is less than 10^4, our method even outperforms [30] since [30] suffers overfitting. This further demonstrates that our method is very useful when labelled data is unavailable.

Impact of Mining Strategies: To demonstrate the effectiveness of our mining strategy, we also compare two other possible mining strategies, as shown in Fig. 4(b). In our method, we use a handcrafted feature, SIFT, to find the most similar patch to the given patch and their hard neighbour in the feature space and then construct the training triplet, with the validation error curve being shown in red in Fig. 4(b). We also illustrate that the results of using our learned network to mine the training triplet on the fly, as shown in the green curve in Fig. 4(b). In this case, the network fails to cluster all the patches much closer while pushing dissimilar ones apart. As illustrated by the validation curve, the network diverges as the training progresses. We opt to use all the patches whose distances are larger than the distance between the anchor patch and its nearest neighbour by a margin m, in a batch to construct training triplets. In this way, the updating direction of the given patch is averaged by all the relative distances ranking losses. Regarding noisy RDR estimation of handcrafted features, the averaging updating direction does not reduce the training loss, as shown in the blue curve in Fig. 4(b). Figure 4(b) also implies that training our unsupervised network is nontrivial.

Selection of Distance Margins: We employ a margin in our RDRL to alleviate the impact of inaccurate estimation of handcrafted features. Figure. 4(c) shows the impact of different margins on the validation errors. When the margin is set to 0.05, our validation curve obtains the lowest error. Hence, we set the margin to 0.05. We also observe that

[1]The results are provided by the authors and the full evaluation is provided in the supplementary material.



(a) Validation curves (b) Test performance

Figure 5. *Performance impacted by increasing training patches.*

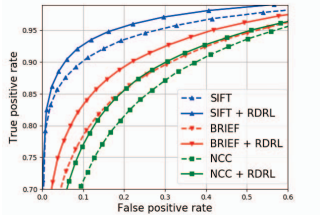


Figure 6. *Performance of our RDRL using different handcrafted descriptors as references.*

when the margin is set to 0.5, the validation errors are larger than the others. This also implies that for a given patch using all the other patches to construct training triplets is not suitable, as illustrated in the blue curve in Fig. 4(b).

Descriptor Performance in Different Dimensions: As visible in Fig. 4(d), the performance of our descriptors varies as the dimension increases. We use Liberty as our training dataset and Notredame and Yosemite as our validation and test datasets. For instance, we employ Yosemite as our validation set and Notredame as our test set. The green curve in Fig. 4(d) indicates the average FPRs in different dimensions when true positive rate (TPR) is 95%. When the dimension of our real-valued descriptor is larger than 128, the performance of our learned descriptors does not increase significantly on the UBC benchmark. Therefore, we set the dimension of our real-valued descriptor to 128.

Impact of Increasing the Training Dataset: Dataset augmentation is a widely known technique to enhance the performance in supervised methods, but it requires extra laborious labelling effort. However, increasing the amount of training images/patches can be regarded as "free" for unsupervised learning methods. Therefore, we enlarge our training dataset by increasing the variety of training patches. Due to the similarity between Liberty and Notredame, we extend our training dataset Liberty with randomly sampled patches from Yosemite and use Notredame as the validation set. As shown in Fig. 5(a), by using both datasets our network achieves a lower validation error rate compared with only using one dataset, Liberty, in training. Moreover, our descriptors also obtain higher matching performance on the Yosemite dataset, as indicated in Fig. 5(b). Although some patches from Yosemite appear in both the training and testing phase, our network does not try to overfit the test dataset since there are no ground-truth labels provided in training.

Learning from Different Handcrafted Features: We demonstrate that using our SIFT based RDRL to train our descriptor network, our network can achieve better performance than our baseline handcrafted feature, SIFT. Nevertheless, our RDRL is also able to improve other handcrafted features. To the best of our knowledge, SIFT is still one of the best off-the-shelf handcrafted features and we regard SIFT as a "Sophisticated" descriptor. Note that, some state-of-the-art handcrafted features [12, 53] require the exact scale information of the patches or may sample outside patch regions to achieve their best performance, their performance degrades dramatically if the above conditions are not satisfied. For comparison, we also select BRIEF as a "Simple" descriptor and NCC as a "Trival" descriptor. As visible in Fig. 6, our RDRL can improve different levels of handcrafted features. Furthermore, we use our descriptor to provide reference RDR and then train our network from scratch. As indicated in Tab. 1, we can further improve the performance of our descriptor network by our proposed RDRL, denoted by Ours†. However, using the descriptor network Ours† to provide reference RDR, we train our network again, and do not see significant improvement since the distribution of the learned descriptors tends to be stable.

Outperforming SIFT: First, deep networks can represent images more discriminatively than handcrafted features due to their complex and various filters. Since our RDRL is designed to measure patch similarity, the network can learn more discriminative deep filters than the handcrafted filters in SIFT. Thus, our network can represent patches more discrminatively. Second, we use relative distance rankings instead of absolute distances as our objective, and the absolute distance between two patches estimated by SIFT can be larger than the distance of our deep features, or vice verse. Therefore, our network can achieve better patch matching performance than SIFT.

5. Conclusion

We present an unsupervised local descriptor network to generate real-valued feature descriptors by using our proposed relative distance ranking loss. Our proposed loss yields direct measurement of patch similarity, thereby generating more discriminative descriptors. Our method combines the sophisticated human experience from handcrafted features with the feature extraction power of deep neural networks, and outperforms both handcrafted features and unsupervised learning based methods.

Acknowledgement. This work is supported by the Australian Research Council Centre of Excellence for Robotic Vision (project number CE140100016), National Natural Science Foundation of China (No. 61976017) and Scape Technologies.

References

[1] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 510–517, 2012. 3

[2] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 4, page 6, 2017. 2, 5, 6

[3] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *British Machine Vision Conference (BMVC)*, 2016. 3, 4, 7

[4] V. Balntas, L. Tang, and K. Mikolajczyk. BOLD - binary online learned descriptor for efficient image matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[5] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, pages 404–417. Springer, 2006. 1, 2, 5, 6

[6] M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):43–57, 2011. 1, 3

[7] H. Cai, K. Mikolajczyk, and J. Matas. Learning linear discriminant projections for dimensionality reduction of image descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):338–352, 2011. 3

[8] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *European Conference on Computer Vision (ECCV)*, pages 778–792, 2010. 3, 5, 6

[9] W. Cheung and G. Hamarneh. N-sift: N-dimensional scale invariant feature transform for matching medical images. In *Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007. 4th IEEE International Symposium on*, pages 720–723. IEEE, 2007. 4

[10] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 539–546. IEEE, 2005. 2

[11] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893, 2005. 2

[12] J. Dong and S. Soatto. Domain-size pooling in local descriptors: Dsp-sift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5097–5106, 2015. 8

[13] Y. Duan, J. Lu, Z. Wang, J. Feng, and J. Zhou. Learning deep binary descriptor with multi-quantization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1183–1192, 2017. 2, 3, 5, 6

[14] V. Erin Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou. Deep hashing for compact binary codes learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2475–2483, 2015. 3

[15] P. Fischer, A. Dosovitskiy, and T. Brox. Descriptor matching with convolutional neural networks: a comparison to sift. *arXiv preprint arXiv:1405.5769*, 2014. 3

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, 2014. 2, 3

[17] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1458–1465. IEEE, 2005. 1

[18] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3279–3286, 2015. 2, 3

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 4

[20] G. Huang, M. Mattar, H. Lee, and E. G. Learned-Miller. Learning to align from scratch. In *Advances in Neural Information Processing Systems (NIPS)*, pages 764–772, 2012. 2

[21] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 5

[22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[23] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[24] J. J. Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biological cybernetics*, 55(6):367–375, 1987. 2

[25] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2548–2555, 2011. 3, 5, 6

[26] J. P. Lewis. Fast template matching. In *Vision interface*, volume 95, pages 15–19, 1995. 6

[27] K. Lin, J. Lu, C.-S. Chen, and J. Zhou. Learning compact binary descriptors with unsupervised deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1183–1192, 2016. 2, 3, 5, 6

[28] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen. Deep learning for generic object detection: A survey. *arXiv preprint arXiv:1809.02165*, 2018. 2

[29] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 1, 2, 4, 5, 6

[30] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4826–4837, 2017. 2, 3, 4, 7

[31] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, volume 1, pages 582–585. IEEE, 1994. 2

[32] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001. 3

[33] C. Osendorfer, J. Bayer, S. Urban, and P. van der Smagt. Unsupervised feature learning for low-level local image descriptors. *arXiv preprint arXiv:1301.2840*, 2013. 3, 6

[34] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid. Local convolutional features with unsupervised training for image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 91–99, 2015. 3

[35] F. Radenović, G. Tolias, and O. Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *European Conference on Computer Vision (ECCV)*, pages 3–20. Springer, 2016. 2

[36] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2, 3

[37] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2564–2571. IEEE, 2011. 3, 5, 6

[38] R. Salakhutdinov and G. Hinton. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978, 2009. 2, 3

[39] J. L. Schönberger, H. Hardmeier, T. Sattler, and M. Pollefeys. Comparative Evaluation of Hand-Crafted and Learned Local Features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5, 6, 7

[40] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *Advances in Neural Information Processing Systems (NIPS)*, pages 41–48, 2004. 4

[41] G. Shakhnarovich. *Learning task-specific similarity*. PhD thesis, Massachusetts Institute of Technology, 2005. 1, 5, 6

[42] F. Shen, Y. Xu, L. Liu, Y. Yang, Z. Huang, and H. T. Shen. Unsupervised deep hashing with similarity-adaptive and discrete optimization. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):3034–3044, 2018. 3

[43] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 3

[44] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1573–1585, 2014. 1, 3

[45] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3, 4, 5, 6

[46] C. Strecha, A. Bronstein, M. Bronstein, and P. Fua. Ldahash: Improved matching with smaller descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):66–78, 2012. 3

[47] Y. Tian, B. Fan, and F. Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 6, 2017. 2, 3, 4, 5

[48] Y. Tian, X. Yu, B. Fan, F. Wu, H. Heijnen, and V. Balntas. Sosnet: Second order similarity regularization for local descriptor learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11016–11025, 2019. 3

[49] T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit. Boosting binary keypoint descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2874–2881, 2013. 1

[50] T. Trzcinski, M. Christoudias, and V. Lepetit. Learning image descriptors with boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):597–610, 2015. 3

[51] T. Trzcinski and V. Lepetit. Efficient discriminative projections for compact binary descriptors. In *European Conference on Computer Vision (ECCV)*, pages 228–242, 2012. 3

[52] L. Van Gool, T. Moons, and D. Ungureanu. Affine/photometric invariants for planar intensity patterns. In *European Conference on Computer Vision (ECCV)*, 1996. 2

[53] Z. Wang, B. Fan, and F. Wu. Local intensity order pattern for feature description. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 603–610, 2011. 8

[54] S. Winder, G. Hua, and M. Brown. Picking the best daisy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 2, 5

[55] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning (ICML)*, pages 478–487, 2016. 2

[56] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. Lift: Learned invariant feature transform. In *European Conference on Computer Vision (ECCV)*, pages 467–483, 2016. 7

[57] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[58] X. Zhang, X. Y. Felix, S. Kumar, and S.-F. Chang. Learning spread-out local feature descriptors. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4605–4613, 2017. 2, 4

[59] L. Zheng, Y. Yang, and Q. Tian. Sift meets cnn: A decade survey of instance retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1224–1244, 2018. 4

[60] M. Zieba, P. Semberecki, T. El-Gaaly, and T. Trzcinski. Bingan: Learning compact binary descriptors with a regularized gan. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 2, 3, 5, 6