Revisiting Unsupervised Local Descriptor Learning

Wufan Wang[1], **Lei Zhang**[1], **Hua Huang**[2*]

[1]School of Computer Science and Technology, Beijing Institute of Technology
[2]School of Artificial Intelligence, Beijing Normal University
{wwf20, leizhang}@bit.edu.cn, huahuang@bnu.edu.cn

Abstract

Constructing accurate training tuples is crucial for unsupervised local descriptor learning, yet challenging due to the absence of patch labels. The state-of-the-art approach constructs tuples with heuristic rules, which struggle to precisely depict real-world patch transformations, in spite of enabling fast model convergence. A possible solution to alleviate the problem is the clustering-based approach, which can capture realistic patch variations and learn more accurate class decision boundaries, but suffers from slow model convergence. This paper presents *HybridDesc*, an unsupervised approach that learns powerful local descriptor models with fast convergence speed by combining the rule-based and clustering-based approaches to construct training tuples. In addition, *HybridDesc* also contributes two concrete enhancing mechanisms: (1) a Differentiable Hyperparameter Search (DHS) strategy to find the optimal hyperparameter setting of the rule-based approach so as to provide accurate prior for the clustering-based approach, (2) an On-Demand Clustering (ODC) method to reduce the clustering overhead of the clustering-based approach without eroding its advantage. Extensive experimental results show that *HybridDesc* can efficiently learn local descriptors that surpass existing unsupervised local descriptors and even rival competitive supervised ones.

Introduction

Computing feature descriptors for image patches around interest points serves as the cornerstone of many computer vision tasks, such as image retrieval (Radenović, Tolias, and Chum 2019), object detection (Ren and Li 2016) and augmented reality (Piao and Kim 2019). The goal is to construct a discriminative feature space where matching patches are projected to neighboring locations while non-matching ones are mapped apart from each other.

Recently, supervised local descriptors based on Convolutional Neural Networks (CNNs) have achieved remarkable performance. However, the heavy reliance of these approaches on class-wise labels (or pairwise matching/non-matching labels) greatly hinders their application in many scenarios where patch annotations are costly to acquire (e.g., medical or hyperspectral images). On the contrary, learning local descriptors from unlabelled patches is more flexible and has lately attracted lots of research attention.

Essentially, supervised and unsupervised local descriptor learning share the same procedure: first construct training tuples involving matching and non-matching patches (e.g., the triplet tuple $\langle$anchor, positive, negative$\rangle$), then calculate a loss function defined over the tuples and minimize it. The difference between the two types of approaches lies in whether patch labels are available to guide the tuple construction, which also accounts for their performance gap. Therefore, the key to bridging this performance gap for unsupervised approaches is constructing *accurate* training tuples, which implies two requirements: *accurately describing pairwise patch relationships, and fully capturing the patch variations undergoing in real-world scenarios.*

The state-of-the-art unsupervised approach (Fan et al. 2021) relies on heuristic rules to construct tuples: the positive samples in the tuples are generated by applying heuristic geometric transformation rules to the original patch, i.e., the anchor sample. Although the generated positive pairs (i.e., positives) have relatively accurate pairwise relationships which enable fast model convergence, the heuristic rules fall short to simulate the diverse and complicated real-world transformations due to their limited capability. As a result, the constructed tuples with unrealistic patch variations can considerably hinder the CNN model from learning accurate class decision boundaries.

A possible solution to ameliorate the problem is utilizing the clustering-based approach (Xie, Girshick, and Farhadi 2016; Caron et al. 2018). Starting with a randomly initialized CNN model, it first uses the CNN model to cluster samples and then constructs tuples with the estimated pseudo-labels to train the CNN model. The clustering and optimization processes are conducted alternately to progressively improve the model. Since all samples come from the training dataset, the positives constructed by such an approach naturally embody more realistic patch variations and allow to learn more powerful models. However, the clustering-based approach suffers from inaccurate pairwise relationships inferred by the less powerful CNN model at the early learning period, leading to impaired model performance and slower convergence than the rule-based approach.

Although neither approach can achieve the desired performance, we observe that the rule-based approach has the

*Corresponding author.

advantage of fast model convergence while the clustering-based approach possesses the advantage of learning models with more accurate class decision boundaries. To construct accurate training tuples, our key idea is to leverage the advantages of both approaches to address their respective shortcomings. To achieve the idea, the learning task is divided into two stages. The first stage focuses on learning regular transformations depicted by transformation rules to quickly learn a relatively powerful CNN model to boost the subsequent learning. In the second stage, the clustering-based approach is utilized to learn from tuples with realistic patch variations. Based on the accurate prior, it can learn more powerful local descriptors than the one bootstrapped by a randomly initialized CNN model.

Based on the idea, this paper proposes *HybridDesc*, an unsupervised approach that leverages the advantages of both worlds to learn more powerful local descriptors with fast convergence speed (see Figure 1). To further improve the performance, *HybridDesc* addresses two main challenges in each learning stage:

(1) *How to set hyperparameters in the transformation rules?* Selecting values for hyperparameters in the transformation rules (e.g., magnitudes of rotation) is a non-trivial task since it determines whether accurate prior can be learned to boost the second-stage learning. Manually selecting values from the prohibitively large hyperparameter space (Fan et al. 2021) requires extensive tuning and struggles to find the optimal setting. To solve the problem, a **differentiable hyperparameter search (DHS) strategy** is designed. The DHS strategy uses the similarity distributions of positives and negatives as the feedback signal to measure the quality of the current hyperparameter setting, and search for the optimal setting in a differentiable manner by regulating the similarity distributions to the desired ones.

(2) *How to exploit the advantage of clustering while avoiding its high computation overhead?* In the second stage, clustering should be conducted iteratively to utilize the increasing power of the CNN model. Existing clustering-based approaches (Caron et al. 2018, 2019) cluster all samples in the training dataset at each iteration, which can incur considerable training overhead due to the large number of samples. We thus introduce an **On-Demand Clustering (ODC) method**, which only clusters samples whose pseudo-labels are likely to change. Since these samples only account for a small portion, the ODC method can effectively reduce the number of samples for clustering and its incurred overhead without degrading model performance.

In summary, the contributions of *HybridDesc* are: (i) combining the advantages of the rule-based and clustering-based approaches to improve local descriptor learning, (ii) design of the DHS strategy to automatically search for the optimal hyperparameters of transformation rules, (iii) design of ODC that can reduce the clustering overhead without hurting model performance, (iv) extensive evaluation of *HybridDesc* on several benchmarks covering a wide range of tasks.

Related Work

This section reviews the training tuple construction methods in unsupervised local descriptor learning. We also discuss the clustering-based unsupervised learning methods. Although they are not designed for the task of local descriptor learning, they inspire our design.

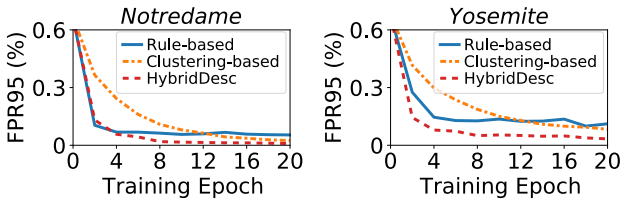


Figure 1: Evolution of the patch verification performance. *HybridDesc* learns a better local descriptor model than other approaches with a fast convergence speed.

Training tuple construction. Existing approaches construct training tuples relying on either hand-crafted descriptors or heuristic rules. A typical approach of the first type is the Relative Distance Ranking Loss (RDRL) (Yu et al. 2019), which infers pairwise relationships between patches based on the similarities of their corresponding SIFT descriptors (Lowe 2004). Since the manually designed spatial filters in hand-crafted descriptors struggle to cope with drastic patch appearance changes, pairwise relationships inferred by hand-crafted descriptors are error-prone.

Rule-based approaches (Lin et al. 2019; Fan et al. 2021) generate positives for training tuples by applying heuristic geometric transformation rules to the original patch, i.e., the anchor sample. The underlying assumption is that a patch shares the same class label as its geometric transformed ones. Although heuristic rules can simulate certain transformations and generate positives with relatively accurate pairwise relationships, it is difficult for them to cover the diverse and complex patch transformations in real-world scenarios. As a result, the constructed training tuples struggle to capture realistic patch variations, making it hard to learn robust local descriptors against patch appearance changes. Moreover, the performance of rule-based approach is sensitive to hyperparameter settings, and improper settings further deteriorate the quality of constructed training tuples.

Clustering-based approach. Several clustering-based approaches have been recently developed to address the image-level unsupervised learning problem. One typical clustering-based approach that can be applied to our task is DeepCluster (Caron et al. 2018). Their main idea is leveraging the CNN model to cluster samples and in turn using the obtained pseudo-labels to supervise the CNN model training. The above two steps are usually conducted alternately to progressive improve the CNN model. Since all positives are collected from the training dataset, the constructed tuples can capture more realistic patch variations than the heuristic rule based approach and enable to learn local descriptors with more accurate decision boundaries. However, clustering-based approaches suffer from slow convergence and impaired performance caused by the error-prone pairwise relationships inferred by the less powerful CNN model at the early learning period.

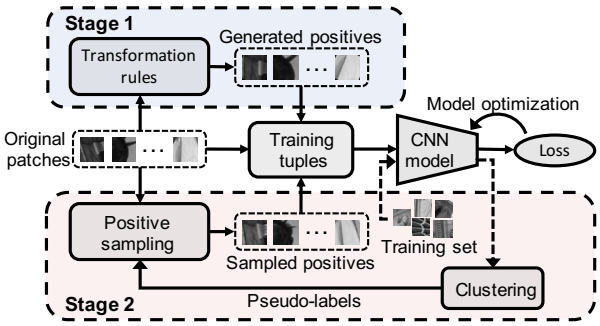


Figure 2: Overview of *HybridDesc*. Steps marked by the solid and dashed lines in Stage 2 are conducted alternately.

Design of *HybridDesc*

Figure 2 overviews *HybridDesc*, which constructs tuples in a hybrid way and learns local descriptors in two stages.

Stage 1: *HybridDesc* utilizes the rule-based approach to construct training tuples. The hyperparameters of the transformation rules are identified by the DHS strategy instead of being manually set, which helps learn accurate prior to boost the clustering-based approach in stage 2.

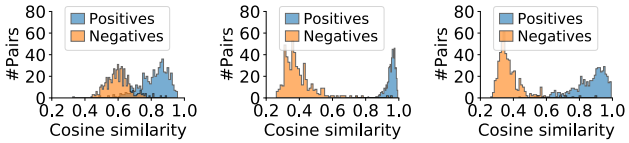
Stage 2: Based on the prior learned in stage 1, *HybridDesc* clusters samples with the ODC method and constructs training tuples based on the estimated pseudo-labels. During learning, ODC and model optimization are alternately conducted to progressively learn powerful local descriptors.

Differentiable Hyperparameter Search (DHS)

The hyperparameter setting of transformation rules has a significant influence on the quality of constructed training tuples, which determines whether accurate prior can be learned to boost the subsequent clustering-based approach. Existing approaches (Fan et al. 2021; Lin et al. 2019) manually set these hyperparameters, which requires substantial efforts for tuning and struggles to find the optimal setting. Although some hyperparameter searching methods have been developed for automatic data augmentation (Cubuk et al. 2019; Hataya et al. 2020), they are not applicable to our task due to their requirement for sample labels.

To overcome the challenge, we introduce the DHS strategy to efficiently find the optimal hyperparameter setting. Our key insight is that the similarity distributions of positives and negatives in the training batch reflect the quality of the current hyperparameter setting, and can be used as the feedback signal to guide the search process. During the search, the DHS strategy adjusts the hyperparameters via backpropagation to regulate current similarity distributions to the desired ones. The optimal setting is found when the similarity distributions match the desired ones. Next, we will discuss details of the DHS strategy with respect to its search space, search objective and the search algorithm.

Search space. In this work, 7 operations are utilized as the transformation rules including *scaling x/y*, *translation x/y*, *shear x/y* and *rotation*, which cover the common geometric transformations in real-world scenarios. The positive sample of a patch is generated by sequentially applying these operations to the input patch. Similar to the previous study (Fan et al. 2021), this work mainly concerns the magnitude of each transformation operation. The search space of DHS is thus a combinational continuous space jointly determined by the magnitude ranges of all transformation operations.



(a) Too large values (b) Too small values (c) Proper values

Figure 3: Similarity distributions of positives and negatives in a training batch under different hyperparameter settings.

Search objective. Intuitively, the hyperparameter setting directly affects the similarity distributions of positives and negatives in a training batch. As shown in Figure 3, setting too large values can cause a large overlap between two distributions and impair the discriminative ability of the learned model, while setting too small values can overwhelm the model with easy positives and degrade its robustness to patch appearance changes. On the contrary, the ideal setting can maintain a good balance between two distributions as shown in Figure 3c, which helps learn both discriminative and robust local descriptors. These observations suggest that the similarity distributions of positives and negatives can be leveraged as the feedback signal to reflect the quality of the hyperparameter setting. Moreover, *the objective of searching the optimal setting can be converted to regulating the similarity distributions to the desired balance*.

Search algorithm. Inspired by the efficiency advantage of differentiable search methods (Hataya et al. 2020) over black-box search methods (Cubuk et al. 2019), the DHS strategy also searches the optimal hyperparameter setting in a differentiable manner. To sculpt the desired distributions, it simultaneously minimizes the overlap of two similarity distributions and unfolds the similarity distribution of positives. In this work, the histogram loss (Ustinova and Lempitsky 2016) is adopted to minimize the distribution overlap. Given two sets $\mathcal{S}^+$ and $\mathcal{S}^-$ that represent the similarities of positives and negatives in a training batch, the histogram loss first estimates the probability distributions of similarities regarding positives p^+ and negatives p^- using histograms. Specifically, considering the R-dimensional similarity histogram of positives H^+ where nodes $n_1, n_2, \cdots, n_R$ are uniformly distributed between $[-1, 1]$ with a step size $\Delta = \frac{2}{R-1}$, the value of H^+ at node r is $h_r^+ = \frac{1}{|\mathcal{S}^+|} \sum_{s_i \in \mathcal{S}^+} \delta_{s_i,r}$, where $\delta_{s_i,r}$ is computed by

$$\delta_{s_i,r} = \begin{cases} (s_i - n_{r-1})/\Delta, & s_i \in [n_{r-1}, n_r] \\ (n_{r+1} - s_i)/\Delta, & s_i \in [n_r, n_{r+1}] \\ 0, & otherwise \end{cases} \quad (1)$$

The similarity histogram of negatives H^- can be estimated in a similar way. Once p^+ and p^- have been estimated, the discrete form of the probability that the similarity of a randomly selected negative pair is higher than that of a randomly selected positive pair can be denoted by

$$L_{overlap} = \sum_{r=1}^{R}(h_r^- \sum_{q=1}^{r} h_q^+) = \sum_{1}^{R} h_r^- \phi_r^+ \quad (2)$$

where ϕ_r^+ represents the cumulative sum of H^+. By minimizing the above equation, the overlap between p^+ and p^- can be effectively reduced. More details can be referred to the histogram loss (Ustinova and Lempitsky 2016) paper. In addition, to obtain a spread-out similarity distribution of positives, the following loss function is introduced

$$L_{dist} = \frac{1}{|\mathcal{S}^+|} \sum_{i}^{|\mathcal{S}^+|} (1 - \frac{(d_i^+)^2}{2}) \quad (3)$$

where d_i^+ is the Euclidean distance between positives. Finally, the loss for hyperparameter search is expressed as:

$$L_{hyper} = L_{overlap} + \lambda \times L_{dist} \quad (4)$$

where λ is a weighting factor to balance the two parts.

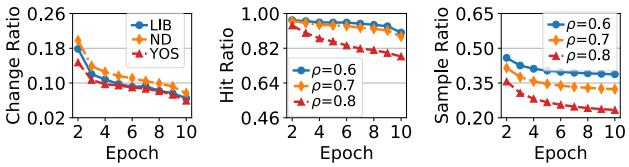
For the transformation operations considered in this work, their gradients w.r.t. their magnitude parameters can be easily obtained, thus can be directly optimized via backpropagation. Considering the contradiction between the goal of minimizing the distance between positives during model optimization and maximizing the distance between positives during hyperparameter search, *HybridDesc* adopts an alternate optimization strategy to alternatively conduct these two optimization processes.

On-Demand Clustering (ODC)

For the clustering-based approach, clustering needs to be conducted iteratively so as to utilize the increasingly powerful CNN model. Existing clustering-based approaches (Caron et al. 2018, 2019) cluster all samples at each iteration, which incurs considerable training overhead due to the large number of samples in the training dataset (i.e., a patch dataset typically contains nearly millions of patches). Different from existing approaches, the proposed ODC method only clusters the samples whose pseudo-labels are likely to change, which helps considerably reduce the overhead incurred by clustering without losing its benefit.

Many pseudo-labels remain unchanged. Essentially, after model optimization in the first stage, the parameters of the local descriptor model tend to be stable, which has been recognized as the "slow drift" phenomena in the previous study (Wang et al. 2020b). As a result, a large portion of pseudo-labels remains unchanged in the second learning stage and the number of changing pseudo-labels gradually decreases as shown in Figure 4a. This provides us the opportunity to reduce the overall data scale for clustering by *only updating those pseudo-labels that are about to change*.

Predicting the change of pseudo-labels. The key is to accurately predict which pseudo-labels are likely to change. Intuitively, the pseudo-labels of ambiguous samples (e.g., samples close to multiple cluster centers) are prone to change due to their inherent instability. Inspired by the Lowe's matching criterion, we introduce the *distance ratio* to measure the uncertainty of a sample, which is defined as the ratio of its distances to the nearest and second nearest



(a) UBC Dataset (b) Liberty (c) Liberty

Figure 4: (a) the ratio of changing pseudo-labels, (b) the ratio of accurately selected pseudo-labels, (c) the ratio of samples selected by ODC.

Algorithm 1: Model optimization with the ODC method

Input: Training set $\mathcal{X}$, pre-trained CNN model $\mathcal{M}(\theta)$, cluster number C, epoch number N, distance ratio threshold ρ
Output: Optimized CNN model

Randomly select C patches from $\mathcal{X}$ as the cluster center set $\mathcal{C}$
Construct the query set $\mathcal{Q} \leftarrow \mathcal{X} - \mathcal{C}$
for i = 1,2,$\cdots$, N **do**
  Compute descriptor with $\mathcal{M}(\theta)$ for each patch $x \in \mathcal{C} \cup \mathcal{Q}$
  Initialize the ambiguous sample set $\mathcal{U} \leftarrow \emptyset$
  for j = 1,2,$\cdots$, $|\mathcal{Q}|$ **do**
    Retrieve top 2 nearest centers c_1 and c_2 in $\mathcal{C}$ for q_j
    Assign q_j to the nearest cluster center c_1
    if $d(f(q_j), f(c_1)) > \rho \times d(f(q_j), f(c_2))$ **then**
      $\mathcal{U} \leftarrow \mathcal{U} \cup q_j$
    end if
  end for
  Update the query set $\mathcal{Q} \leftarrow \mathcal{U}$
  Construct training tuples with obtained pseudo-labels
  Optimize θ by minimizing the loss computed over constructed training tuples.
end for
return $\mathcal{M}(\theta)$

cluster centers. A distance ratio close to 1 means that the pseudo-label of the sample is very likely to change and vice versa. Figure 4b shows that the distance ratio can effectively pick out samples whose pseudo-labels are about to change.

Clustering only uncertain samples. Based on the distance ratio, the ODC method actively selects and clusters ambiguous samples instead of the whole dataset, which considerably reduces the data scale for clustering as shown in Figure 4c. Specifically, at the beginning of the second-stage learning, the ODC method selects a fixed number of samples as cluster centers. Since randomly selecting cluster centers at each iteration brings no performance improvement, these samples are constantly used as cluster centers in subsequent learning. Before clustering at each iteration, ambiguous samples are selected from the ambiguous set of the previous iteration. Afterwards, the local descriptors of these samples along with samples corresponding to their top-two nearest cluster centers are calculated with the CNN model to be used for updating their cluster assignments. Detailed steps of applying the proposed ODC method for model op-

timization are given in Algorithm 1, where $f(\cdot)$ represents the descriptor of the input patch while $d(\cdot, \cdot)$ computes the distance between two input descriptors.

Implementation Details

The L2Net (Tian, Fan, and Wu 2017) is adopted as the CNN model to embed local patches to descriptors of length 128 and 256 for a fair comparison with existing varying-length descriptors. The $sign(\cdot)$ function is applied to the learned real-valued descriptors to obtain the binary descriptors. For model optimization, the "hardest-in-batch" mining strategy and the triplet margin loss are utilized following HardNet (Mishchuk et al. 2017). Data augmentation composed of random horizon flipping and 90-degree rotation is also used. The stochastic gradient descent (SGD) optimizer is adopted with an initial learning rate of 10, which is linearly decayed to zero. All models are trained for 60 epochs, with the first 10 epochs trained using the rule-based approach and the other 50 epochs trained using the clustering-based approach, unless otherwise stated. For hyperparameter search, the Adam optimizer (Kingma and Ba 2015) is used with a constant learning rate of 0.1. The magnitude range of *scaling x/y*, *translation x/y*, *shear x/y* and *rotation* are $[0.5, 1.5]$, $[-0.5, 0.5]$, $[-0.5, 0.5]$ and $[-180°, 180°]$. The initial hyperparameter of each operation is set to 0.01 of its max magnitude. The manually set hyperparameters have the same values as that in (Fan et al. 2021), except that the shear operation is used to replace tilt and out-plane rotation operations for simplicity. The cluster number C, weighting factor λ and distance ratio ρ are set to 100K, 0.02 and 0.8 respectively. Training is done with PyTorch (Paszke et al. 2019) on an NVIDIA RTX 2080 GPU. The differentiable geometric transformations are implemented by the Kornia Library (Riba et al. 2020), while clustering is implemented by the Faiss library (Johnson, Douze, and Jégou 2019).

Evaluation

In this section, we compare *HybridDesc* with several existing approaches including unsupervised deep local descriptors DeepBit (Lin et al. 2019), GraphBit (Duan et al. 2018), D-GraphBit (Wang et al. 2022), TBLD (Miao et al. 2021), BLCD (Fan et al. 2021), RDRL (Yu et al. 2019), supervised deep local descriptors DeepDesc (Simo-Serra et al. 2015), DC (Zagoruyko and Komodakis 2015), TFeat (Balntas et al. 2016), L2Net (Tian, Fan, and Wu 2017), HardNet (Mishchuk et al. 2017), DOAP (He, Lu, and Sclaroff 2018), SOSNet (Tian et al. 2019), Dynamic Soft Margin (DSM) (Zhang and Rusinkiewicz 2019), DualHard (Wang et al. 2020a), HyNet (Tian et al. 2020) as well as handcrafted descriptors SIFT (Lowe 2004), BRIEF (Calonder et al. 2010), ORB (Rublee et al. 2011). The evaluation is conducted on the UBC Phototour (Brown, Hua, and Winder 2011), HPatches (Balntas et al. 2017), Heinly (Heinly, Dunn, and Frahm 2012) and W1BS (Mishkin et al. 2015) datasets.

UBC Phototour Dataset

As the most widely used dataset for local descriptor learning, the UBC Phototour dataset consists of three subsets, namely $Liberty$ (LIB), $Yosemite$ (YOS) and $Notredame$ (ND). For evaluation, the dataset is split into six training-test combinations, in which one subset is used for training while the other two are used for testing. The false positive rate at 95% true positive rate (FPR95) is reported following the standard evaluation protocol (Mishchuk et al. 2017). As can be seen from the results listed in Table 1, *HybridDesc* outperforms both hand-crafted descriptors and unsupervised local descriptors on all training/testing splits. To be specific, the previous best unsupervised real-valued descriptor RDRL and binary descriptor BLCD report a mean FPR95 of 15.66% and 8.53% respectively, while *HybridDesc* achieves much lower mean FPR95 values of 2.57% and 4.76%, which verifies the effectiveness of our approach. Moreover, for both real-valued and binary descriptors, the performance of *HybridDesc* is very close to that of the supervised approach L2Net which adopts the same network architecture, showing the great potential of *HybridDesc* to be applied in scenarios where patch labels are not available.

Train / Test	Dim /Bits	ND LIB	YOS LIB	LIB ND	YOS ND	LIB YOS	ND YOS	Mean
Real-valued Descriptors								
DeepDesc	128	8.82	8.82	4.54	4.54	16.19	16.19	9.85
TFeat	128	7.22	9.79	3.12	3.85	7.82	7.08	6.47
L2Net	128	2.36	4.7	0.72	1.29	2.57	1.71	2.22
DOAP	128	1.54	2.62	0.43	0.87	2.00	1.21	1.45
HardNet	128	1.49	2.51	0.53	0.78	1.96	1.84	1.51
DSM	128	1.21	2.01	0.39	0.68	1.51	1.29	1.18
SOSNet	128	1.08	2.12	0.35	0.67	1.03	0.95	1.03
DualHard	128	1.24	2.23	0.41	0.67	1.59	1.28	1.24
HyNet	128	0.89	1.37	0.34	0.61	0.88	0.96	0.84
DC	256	13.24	17.25	6.01	8.38	19.91	15.89	13.45
SIFT	128	30.76	30.76	25.17	25.17	27.77	27.77	27.90
RDRL	128	19.65	18.92	12.56	11.70	16.22	14.92	15.66
Ours	128	2.97	4.15	1.06	1.14	3.29	2.83	2.57
	256	**2.70**	**3.63**	**0.81**	**1.00**	**3.17**	**2.67**	**2.33**
Binary Descriptors								
L2Net	128	7.44	10.29	3.81	4.31	8.81	7.45	7.01
DSM	128	2.70	4.01	0.93	1.44	3.69	2.98	2.63
RDRL	128	27.59	29.25	20.96	20.79	23.20	23.23	24.17
ORB	256	56.26	56.26	48.03	48.03	54.13	54.13	52.81
BRIEF	256	59.15	59.15	54.57	54.57	54.96	54.96	56.23
DeepBit	256	33.83	34.64	20.66	28.49	56.69	54.63	38.15
GraphBit	256	21.18	24.72	15.25	17.78	49.64	49.94	29.75
D-GraphBit	256	15.66	9.56	10.63	17.78	41.47	40.07	22.05
TBLD	256	20.45	21.95	14.47	16.53	36.88	35.09	18.25
BLCD	256	10.07	11.90	4.90	5.26	9.02	10.03	8.53
Ours	128	9.36	12.13	4.77	4.83	10.32	8.72	8.35
	256	**5.62**	**6.76**	**2.38**	**2.22**	**6.15**	**5.41**	**4.76**

Table 1: FPR95 (%) of different methods on the UBC Phototour dataset. Best results of unsupervised local descriptors are in bold.

To better understand the contribution of each module in *HybridDesc*, we also conduct an ablation study on the UBC Phototour dataset with the results summarized in Table 2. All models are trained for 20 epochs. For hybrid approaches, the models are trained with the rule-based approach in the first 10 epochs and the clustering-based approach in the other

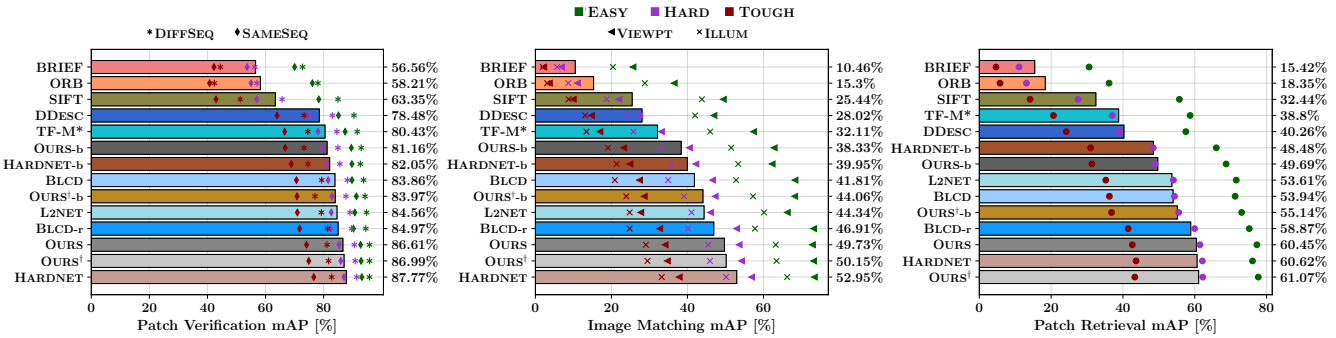


Figure 5: Verification, matching and retrieval results on the HPatches dataset. Marker color and type represent the level of geometric noises and experimental setup, respectively. DIFFSEQ and SAMESEQ indicate the source of negative samples for the patch verification task while VIEWPT and ILLUM indicate the splits type for the image matching task.

Config (dim=128)	**ND**	**YOS**	**LIB**	**YOS**	**LIB**	**ND**	**Mean**
	LIB		**ND**		**YOS**		
Manual	8.70	8.62	5.83	4.75	9.54	10.19	7.94
Clustering	4.76	5.67	1.32	1.26	5.49	4.14	3.77
Manual+Clustering	3.70	5.01	1.02	1.24	4.69	3.72	3.23
Ours (DHS)	8.51	8.18	3.44	4.10	7.28	7.74	6.54
Ours (DHS+Clustering)	3.64	4.69	0.99	1.18	3.99	3.40	2.98
Ours (DHS+ODC)	3.50	4.61	0.96	1.18	3.90	3.45	2.93

Table 2: FPR95 (%) on the UBC Phototour dataset achieved by models trained with different configurations.

Method	**LIB**	**ND**	**YOS**	**Average**
Clustering	45.82	47.44	68.95	54.07
ODC	20.03	21.56	34.39	25.33

Table 3: Comparison of the clustering time (%).

10 epochs. As can be seen from the table, using either the rule-based approach (Manual) or the clustering-based approach (Caron et al. 2018) only achieves inferior performance. Although directly combining these two approaches (Manual+Clustering) brings noticeable improvements, the benefit of combining both approaches is not fully unleashed due to the suboptimal hyperparameter setting of the transformation rules. For our approach, the DHS strategy enables to learn better local descriptor models than the rule-based approach with manually selected hyperparameters, thus verifying its advantage in finding the optimal setting. Moreover, the DHS strategy also helps boost the performance of the clustering-based approach in the second stage. Table 3 presents the ratio of time for clustering to time for model optimization. Since the time used for model optimization is the same for both methods, the results can directly reflect their clustering overhead. From the results in Table 2 and Table 3, it can be drawn that the proposed ODC method can consistently reduce the clustering overhead by more than 50% on all three subsets without degrading model performance.

HPatches Dataset

The HPatches dataset contains more than 1.5 million patches extracted from 116 sequences of viewpoint and illumination changing images. The extracted patches are divided into three groups including EASY, HARD and TOUGH based on the level of geometric noise. There are three evaluation tasks: patch verification, image matching and patch retrieval. Following previous studies (Mishchuk et al. 2017; Tian et al. 2019), all our models are trained on Liberty and the mean average precision (mAP) is used to measure the performance of compared approaches on the "a" split. The results are illustrated in Figure 5. As depicted by the figure, our approach surpasses the state-of-the-art unsupervised approach BLCD on all evaluation tasks in terms of both real-valued and binary local descriptors. In particular, our binary local descriptor OURS†-b with a dimension length of 256 improves the mAP by 0.11%, 2.25% and 1.2% on all three tasks compared with BLCD. In the meanwhile, its real-valued version OURS† yields even larger improvements over BLCD-r, which are 2.02%, 3.24% and 2.2% respectively. Surprisingly, our real-valued local descriptor OURS with a dimension length of 128 achieves better performance than L2Net and close performance to the seminal supervised approach HardNet.

Heinly Dataset

In real-world image matching scenarios, the diverse image distortions (e.g., geometric transformations, illumination changes, JPEG compression and etc.) pose great challenge for establishing correct correspondences. In order to verify whether our learned local descriptor models are capable of handling these distortions, we evaluate our approach on the Heinly dataset. The matching score is adopted to evaluate the image matching quality following the previous work (Heinly, Dunn, and Frahm 2012). To determine the correctness of a match, the ground-truth homographies is leveraged to warp the detected keypoints from the first image to all remaining images in the sequence. Only matching points that are within 2.5 pixels of each other are considered to be correct. The matching scores achieved by different approaches are presented in Figure 6. For binary local descriptors, our learned model OURS†-b surpasses the state-of-the-art unsupervised approach BLCD on all sequences. Besides, our real-valued local descriptor model not only outperforms the handcrafted descriptor SIFT and supervised local descriptors TFeat, L2Net, but also achieves a matching score close

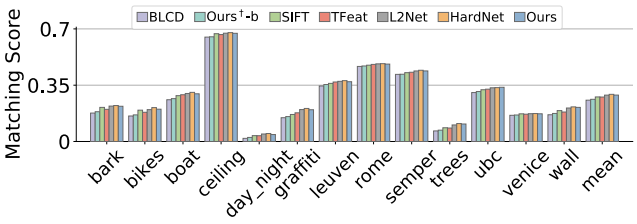


Figure 6: Image matching results on the Heinly dataset.

Method	A	G	L	S	M2P	Average
SIFT	0.1056	0.1097	0.1334	0.0089	0.0311	0.0777
TFeat	0.0702	0.0912	0.1130	0.0031	0.0076	0.0570
L2Net	0.1199	0.1259	0.1302	0.0131	0.0223	0.0823
HardNet	0.1268	0.1312	0.1324	0.0201	0.0239	0.0869
Ours	0.1260	0.1241	0.1335	0.0154	0.0271	0.0852
BLCD	0.0742	0.0852	0.1166	0.0045	0.0048	0.0571
Ours†-b	0.0962	0.1003	0.1192	0.0066	0.0149	0.0675

Table 4: Evaluation results on the W1BS dataset.

to HardNet, which demonstrates the superiority of our approach for handling diverse patch variations.

W1BS Dataset

To further validate the capability of our learned local descriptors to operate in extreme conditions, we evaluate them on the W1BS dataset, which contains 40 image pairs with each involving one specific nuisance factor such as Appearance (A), Geometry (G), Illumination (L), Sensor (S) and electronic map to satellite photos (M2P). In addition, local features in the W1BS dataset are detected with MSER (Matas et al. 2002), Hessian-Affine (Mikolajczyk and Schmid 2004) and FOCI (Zitnick and Ramnath 2011) detectors. These keypoint detectors fire on different locations from DoG (Lowe 2004) used to extract training patches, which can help verify the generalization ability of our learned local descriptors across different detectors. Following the previous study (Mishchuk et al. 2017), the mean Area Under Curve (mAUC) is employed to compare the performance of different local descriptors. Table 4 illustrates the evaluation results, in which our approach exhibits competitive performance for both real-valued and binary local descriptors. As shown in the table, our real-valued local descriptor achieves superior performance than SIFT, TFeat and L2Net on almost all sequences, and is only slightly inferior to HardNet. Moreover, the binary local descriptor OURS†-b consistently yields higher mAUC values than BLCD on all sequences. The above results indicates the strong generalization ability of local descriptors learned with our approach.

Sensitivity Analysis

In this part, we empirically investigate the impact of key parameters in our approach including the cluster number, clustering frequency, weighting factor λ as well as the distance ratio ρ on the performance of learned local descriptors. The local descriptor model is trained on *Liberty* with varying experimental settings and the mean FPR95 achieved on

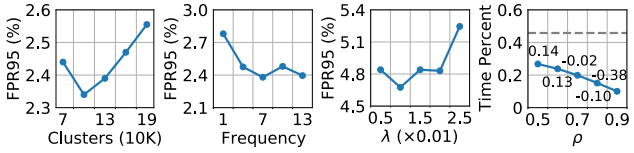


Figure 7: Impact of hyperparameters on model performance.

Yosemite and *Notredame* is used to measure model performance. The results of our approach under different hyperparameter settings are illustrated in Figure 7. Specifically, as can be seen from the figure that the optimal setting for cluster number is 100K, which is smaller than the actual cluster number of Liberty 161072. The reason is that some patches from different classes in Liberty have very similar appearances, thus the intrinsic data distribution can be well represented by fewer clusters. For the clustering frequency, a higher frequency yields a relatively steady trend of performance improvements, but at the cost of extra clustering overhead. The weighting factor λ has a significant impact on the discriminative ability and robustness of the learned local descriptors. As its value increases, the performance of the learned model first increases and then decreases when the value exceeds 0.01. In the meanwhile, it is observed that a wide range of λ values can achieve similar performance to that of the optimal setting. In the last figure, the dash line represents the ratio between the clustering time and the model optimization time when conducting clustering on all samples in the dataset (baseline). The values beside each marker denote the performance improvement or degradation with respect to the baseline. In general, as ρ increases, the clustering overhead is reduced, which however leads to the gradually degraded performance of the learned local descriptor model. In addition, when ρ is properly set, the clustering overhead can be greatly reduced without hurting model performance. We also note that the performance of the learned model outperforms the baseline when ρ is set to 0.5 or 0.6, we hypothesize this is because selectively clustering samples as done in the ODC method can help stabilize the training by alleviating the fluctuation of pseudo-labels.

Conclusion

This paper proposes an unsupervised approach *HybridDesc* which improves local descriptor learning by combining the advantages of the rule-based approach and the clustering-based approach to construct accurate training tuples. To fully exploit the benefit of combining two approaches, we propose the DHS strategy to find the optimal hyperparameter setting of transformation rules to provide accurate prior for subsequent learning, and propose the ODC method to reduce the training overhead incurred by clustering without hurting model performance. Extensive experiments conducted on several datasets show that *HybridDesc* can achieve noticeable improvements over existing unsupervised approaches and close performance to the competitive supervised approach. The promising results suggest the potential of our approach for learning powerful local descriptors without patch labels.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 61936011, Grant 61922014 and Grant 62002020.

References

Balntas, V.; Edgar, R.; Daniel, P.; and Krystian, M. 2016. Learning Local Feature Descriptors with Triplets and Shallow Convolutional Neural Networks. In *Proceedings of the British Machine Vision Conference*, 1–11.

Balntas, V.; Lenc, K.; Vedaldi, A.; and Mikolajczyk, K. 2017. HPatches: A Benchmark and Evaluation of Handcrafted and Learned Local Descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3852–3861.

Brown, M.; Hua, G.; and Winder, S. 2011. Discriminative Learning of Local Image Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1): 43–57.

Calonder, M.; Lepetit, V.; Strecha, C.; and Fua, P. 2010. BRIEF: Binary Robust Independent Elementary Features. In *Proceedings of the European Conference on Computer Vision*, 778–792.

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep Clustering for Unsupervised Learning of Visual Features. In *Proceedings of the European Conference on Computer Vision*, 139–156.

Caron, M.; Bojanowski, P.; Mairal, J.; and Joulin, A. 2019. Unsupervised Pre-Training of Image Features on Non-Curated Data. In *Proceedings of the IEEE International Conference on Computer Vision*, 2959–2968.

Cubuk, E. D.; Zoph, B.; Mané, D.; Vasudevan, V.; and Le, Q. V. 2019. AutoAugment: Learning Augmentation Strategies from Data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 113–123.

Duan, Y.; Wang, Z.; Lu, J.; Lin, X.; and Zhou, J. 2018. GraphBit: Bitwise Interaction Mining via Deep Reinforcement Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8270–8279.

Fan, B.; Liu, H.; Zeng, H.; Zhang, J.; Liu, X.; and Han, J. 2021. Deep Unsupervised Binary Descriptor Learning Through Locality Consistency and Self Distinctiveness. *IEEE Transactions on Multimedia*, 23: 2770–2781.

Hataya, R.; Zdenek, J.; Yoshizoe, K.; and Nakayama, H. 2020. Faster AutoAugment: Learning Augmentation Strategies Using Backpropagation. In *Proceedings of the European Conference on Computer Vision*, 1–16.

He, K.; Lu, Y.; and Sclaroff, S. 2018. Local Descriptors Optimized for Average Precision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 596–605.

Heinly, J.; Dunn, E.; and Frahm, J.-M. 2012. Comparative Evaluation of Binary Features. In *Proceedings of the European Conference on Computer Vision*, 759–773.

Johnson, J.; Douze, M.; and Jégou, H. 2019. Billion-scale Similarity Search with GPUs. *IEEE Transactions on Big Data*, 7(3): 535–547.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations*, 1615–1630.

Lin, K.; Lu, J.; Chen, C.-S.; Zhou, J.; and Sun, M.-T. 2019. Unsupervised Deep Learning of Compact Binary Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(6): 1501–1514.

Lowe, D. G. 2004. Distinctive Image Features from Scale-Invariant Keys. *International Journal of Computer Vision*, 60(2): 91–110.

Matas, J.; Chum, O.; Urban, M.; and Pajdla, T. 2002. Robust Wide-baseline Stereo from Maximally Stable Extremal Regions. In *Proceedings of the British Machine Vision Conference*, 384–393.

Miao, Y.; Lin, Z.; Ma, X.; Ding, G.; and Han, J. 2021. Learning Transformation-Invariant Local Descriptors with Low-Coupling Binary Codes. *IEEE Transactions on Image Processing*, 30: 7554–7566.

Mikolajczyk, K.; and Schmid, C. 2004. Scale & Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, 60(1): 63–86.

Mishchuk, A.; Mishkin, D.; Radenović, F.; and Matas, J. 2017. Working Hard to Know Your Neighbor's Margins: Local Descriptor Learning Loss. In *Proceedings of the International Conference on Neural Information Processing Systems*, 4829–4840.

Mishkin, D.; Matas, J.; Perdoch, M.; and Lenc, K. 2015. WxBS: Wide Baseline Stereo Generalizations. In *Proceedings of the British Machine Vision Conference*, 12.1–12.12.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proceedings of the International Conference on Neural Information Processing Systems*, 8024–8035.

Piao, J.-C.; and Kim, S.-D. 2019. Real-Time Visual–Inertial SLAM Based on Adaptive Keyframe Selection for Mobile AR Applications. *IEEE Transactions on Multimedia*, 21(11): 2827–2836.

Radenović, F.; Tolias, G.; and Chum, O. 2019. Fine-Tuning CNN Image Retrieval with No Human Annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7): 1655–1668.

Ren, H.; and Li, Z.-N. 2016. Object Detection Using Boosted Local Binaries. *Pattern Recognition*, 60(C): 793–801.

Riba, E.; Mishkin, D.; Ponsa, D.; Rublee, E.; and Bradski, G. 2020. Kornia: an Open Source Differentiable Computer Vision Library for PyTorch. In *IEEE Winter Conference on Applications of Computer Vision*, 3663–3672.

Rublee, E.; Rabaud, V.; Konolige, K.; and Bradski, G. 2011. ORB: An Efficient Alternative to SIFT or SURF. In *Proceedings of the IEEE International Conference on Computer Vision*, 2564–2571.

Simo-Serra, E.; Trulls, E.; Ferraz, L.; Kokkinos, I.; Fua, P.; and Moreno-Noguer, F. 2015. Discriminative Learning of Deep Convolutional Feature Point Descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, 118–126.

Tian, Y.; Barroso-Laguna, A.; Ng, T.; Balntas, V.; and Mikolajczyk, K. 2020. HyNet: Learning Local Descriptor with Hybrid Similarity Measure and Triplet Loss. In *Proceedings of the International Conference on Neural Information Processing Systems*, 7401–7412.

Tian, Y.; Fan, B.; and Wu, F. 2017. L2-Net: Deep Learning of Discriminative Patch Descriptor in Euclidean Space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6128–6136.

Tian, Y.; Yu, X.; Fan, B.; Wu, F.; Heijnen, H.; and Balntas, V. 2019. SOSNet: Second Order Similarity Regularization for Local Descriptor Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 11008–11017.

Ustinova, E.; and Lempitsky, V. 2016. Learning Deep Embeddings with Histogram Loss. In *Proceedings of the International Conference on Neural Information Processing Systems*, 4177–4185.

Wang, S.; Guo, X.; Tie, Y.; Qi, L.; and Guan, L. 2020a. Deep Local Feature Descriptor Learning with Dual Hard Batch Construction. *IEEE Transactions on Image Processing*, 29: 9572–9583.

Wang, X.; Zhang, H.; Huang, W.; and Scott, M. R. 2020b. Cross-Batch Memory for Embedding Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6387–6396.

Wang, Z.; Xiao, H.; Duan, Y.; Zhou, J.; and Lu, J. 2022. Learning Deep Binary Descriptors via Bitwise Interaction Mining. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1.

Xie, J.; Girshick, R.; and Farhadi, A. 2016. Unsupervised Deep Embedding for Clustering Analysis. In *Proceedings of the International Conference on Machine Leaning*, 478–487.

Yu, X.; Tian, Y.; Porikli, F.; Hartley, R.; Li, H.; Heijnen, H.; and Balntas, V. 2019. Unsupervised Extraction of Local Image Descriptors via Relative Distance Ranking Loss. In *ICCV Workshop*, 2893–2902.

Zagoruyko, S.; and Komodakis, N. 2015. Learning to Compare Image Patches via Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4353–4361.

Zhang, L.; and Rusinkiewicz, S. 2019. Learning Local Descriptors with a CDF-Based Dynamic Soft Margin. In *Proceedings of the IEEE International Conference on Computer Vision*, 2969–2978.

Zitnick, C. L.; and Ramnath, K. 2011. Edge Foci Interest Points. In *Proceedings of the IEEE International Conference on Computer Vision*, 359–366.