Kernelized Subspace Pooling for Deep Local Descriptors

Xing Wei, Yue Zhang, Yihong Gong, Nanning Zheng
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University

Abstract

Representing local image patches in an invariant and discriminative manner is an active research topic in computer vision. It has recently been demonstrated that local feature learning based on deep Convolutional Neural Network (CNN) can significantly improve the matching performance. Previous works on learning such descriptors have focused on developing various loss functions, regularizations and data mining strategies to learn discriminative CNN representations. Such methods, however, have little analysis on how to increase geometric invariance of their generated descriptors. In this paper, we propose a descriptor that has both highly invariant and discriminative power. The abilities come from a novel pooling method, dubbed Subspace Pooling *(SP) which is invariant to a range of geometric deformations. To further increase the discriminative power of our descriptor, we propose a simple distance kernel integrated to the marginal triplet loss that helps to focus on hard examples in CNN training. Finally, we show that by combining SP with the projection distance metric [13], the generated feature descriptor is equivalent to that of the Bilinear CNN model [22], but outperforms the latter with much lower memory and computation consumptions. The proposed method is simple, easy to understand and achieves good performance. Experimental results on several patch matching benchmarks show that our method outperforms the state-of-the-arts significantly.*

1. Introduction

Matching local patches across images has been one of the most extensively studied topics in computer vision. It is often used as an essential step in a wide range of vision tasks such as structure from motion [26], stereo matching [43], image stitching [7], image retrieval [27] and image classification [21], to name a few. In general, the patch matching accuracy strongly depends on the quality of the feature descriptors extracted from the local patches. Designing good feature descriptors is a very challenging problem. On one hand, the feature descriptor must be able to handle variations between two matching patches caused by viewpoint and illumination changes, occlusions, differences in camera settings, *etc*. On the other hand, it must be highly discriminative for non-matching patches with similar appearances.

In the past decades, a lot of research efforts have been made in the literature to seek for invariant and discriminative local descriptors. Many early works focused on developing hand-crafted feature descriptors such as SIFT, SURF, HOG, DAISY, ORB, LIOP, *etc*. [23, 5, 8, 35, 28, 39], while more recent works adopted machine learning methods to learn high-quality feature descriptors from large image datasets [6, 36, 33, 10, 31, 4]. Together with the great success of deep CNN in various vision related tasks, it has recently been demonstrated that patch matching using CNNs can significantly improve the matching performance accuracies [42, 14, 30, 3, 20, 41, 34, 9]. Machine learning methods for local patch matching can be mainly classified into two categories: (1) methods that treat the local patch matching problem as a binary classification problem, and output a matching score for each input patch pair; (2) methods that learn a feature extractor, and output a feature representation (descriptor) for each input local patch. An obvious drawback of the first approach is that the computational cost is expensive as it requires all combinations of patches to be tested against each other in a brute-force manner. In contrast, the second approach treats each image patch independently and produces feature descriptors that can be used in a broad range of vision tasks. Many research studies in this category have focused on developing new loss functions, regularization terms, and data mining strategies to learn discriminative feature descriptors. Such methods, however, have no theoretical guarantee to be invariant to geometric changes described above.

In this paper, we propose a novel CNN pooling method, namely *Subspace Pooling* (SP) to learn highly invariant and discriminative feature descriptors. The proposed pooling function is invariant to circular shift, flipping, in-plane rotation, and many kinds of in-plane deformations. More specifically, we take the output of the last convolution layer of a given CNN, and use it to form a matrix F where each row i represents the i'th feature map stacked to a 1D vector, and each column corresponds to a spatial location of the feature maps. We then compute the Singular Value Decomposition (SVD) of the matrix and use its principal singular vectors as the feature representation of the input patch. We show that the feature descriptor produced this way is invariant to all the geometric changes that can be expressed as the column permutations of the matrix F. The proposed SP method is differentiable and thus can be readily applied to a given CNN using the standard Back Propagation (BP) training method. Comprehensive experimental evaluations on three popular patch matching benchmarks show superior performance accuracies to the representative methods in the literature. Moreover, we propose a simple Gaussian kernel to further improve its discriminative power. The distance kernel together with the marginal triplet loss makes the training procedure focus more on hard examples, thus helps to push the envelope further.

In summary, our contributions include: (1) we propose a novel CNN subspace pooling method that can remarkably improve the patch matching accuracy; (2) we show that the SP method is invariant to all geometric changes that can be expressed as the column permutations of the matrix F; (3) to increase its discriminative power, we further proposed a distance kernel integrated to the marginal triplet loss which is helpful to focus on hard examples in CNN training; (4) we finally show that our SP method combined with the projection distance metric [13] is equivalent to the Bilinear CNN, but outperforms the latter with much lower memory and computation consumptions.

2. Related Work

2.1. Invariant and Discriminative Feature Learning

A general framework for building robust local descriptors is to first compute local statistics (*e.g.* gradients) and then pool them together. The first stage aims to extract discriminative and photometric invariant features and the second stage is helping to increase geometric invariance. Most conventional feature descriptors are hand-crafted and use a fixed configuration for region pooling. For example, SIFT [23] and its variants [5, 1] use rectangular regions organized in a grid, GLOH [24] uses a polar arrangement of summing regions, while DAISY [35] employs a set of multi-size circular regions grouped into rings. To improve performance, several learning based methods were proposed to select pooling regions in a principled way. Brown *et al.* [6] proposed a Powell optimisation method to find the best configuration of DAISY-like descriptors. Pooling region selection using boosting was explored in [36], achieving good performance. It is also shown in [31] that learning the pooling regions can be performed by optimising a sparsity-inducing L_1 regulariser, yielding a convex problem which has a global optimal solution. Though these methods achieve remarkable performance, they are much more sophisticated than our method and are difficult to be applied into CNNs.

Other works handle geometric invariance from different perspectives. Hassner *et al.* [15] proposed scale invariant SIFT, as an alternative to single-scale descriptors. They employed a linear subspace representation of multi-scale SIFT descriptors which is similar to our representation but in different domain. This idea was further extended in [40] to produce an affine invariant descriptor of a set of affine warped pathes. Our proposed method differs from them in that our method operates on spatial domain and can handle more general transforms such as the non-rigid deformation. Furthermore, these methods lack a learning procedure to learn descriptors from data, but using a pre-defined configuration to compute the linear subspace representation by generating helper images at multiple scales or using different affine transforms. This greatly limits the deformation types that can be handled by those methods. On the contrary, our method is an end-to-end framework, thus can improve invariance to such transforms by directly adding helper images into our training set as a data augmentation process. There are also descriptors to handle the non-rigid deformation. One representative is the DaLI method [29] (DaLI), which uses the heat diffusion theory to build a deformation and illumination invariant descriptor. Compared to DaLI, our method is simpler and performs better.

2.2. Local Patch Matching via CNNs

Motivated by tremendous successes of deep learning techniques in visual classification and recognition problems. Researchers have paid attention to using deep CNNs in local feature learning. Han *et al.* [14] (MatchNet) proposed to jointly learn a deep network for patch representation as well as a network for robust feature comparison. It significantly improves previous results, showing a great potential of CNNs in descriptor learning. Zagoruyko and Komodakis [42] (DC-S2S) explored different kinds of network architectures for patch matching. They found that a 2-channel model, which simply considers the two patches of an input pair as a 2-channel image, achieves the best result. This kind of methods is computational expensive since they require all combinations of patches to be tested against each other in a brute-force manner.

Another class of works tries to learn local descriptors using existing distance metric such as the L_2 distance. Simo-Serra *et al.* [30] (DeepDesc) trained a siamese network using a mining strategy to select hard pairs. Kumar *et al.* [20] (TNet-TGLoss) used triplet network and proposed a global loss function to minimize the overall classification error in the training set. Balntas *et al.* [3] (TF-M) also adopted a triplet network, together with an in-triplet mining method of hard negatives. Tian *et al.* [34] (L2-Net) proposed an efficient sampling strategy, and several regularizations on the intermediate feature maps and the output descriptor to improve performance. Mishchuk *et al.* [25] (HARDNET) proposed an effective mining strategy which mimics the feature matching procedure in a batch fashion, achieving the currently best performance.

This paper aims to develop an invariant and discriminative descriptor that can be measured with existing distance metric efficiently. The proposed method is distinctive in that it designs a novel pooling method which is invariant to many kinds of in-plane transforms and is robust to a wide range of deformations validated by various challenging benchmarks. Such properties are important and have long been pursued in local descriptor engineering.

3. Kernelized Subspace Pooling

3.1. Preliminaries

A typical CNN consists of mainly three building blocks: convolution, active function, and pooling fuction. The conventional convolution operation in CNN uses a fixed size kernel to extract features in a sliding window fashion. Each element of the kernel is used at every position (except perhaps some of the boundary pixels), thus the convolution can be considered as a spatially invariant filtering which has been intensively studied in the image processing community [11]. The spatially invariant filtering, however, dose not ensure that the output feature map is invariant to geometric transforms since the spatial locations of feature elements are maintained. Nevertheless, the learned kernels together with the non-linear active functions can be robust to photometric variations when trained on large data.

In order to handle geometric changes, pooling methods are necessary. A pooling function replaces the output of the network at a certain location with a summary statistic of its nearby outputs. For example, the max pooling computes the maximum output within a rectangular neighborhood. Other popular pooling functions include the average pooling, the L_2 norm of a rectangular neighborhood, or a weighted average based on the distance from the central pixel [12]. The fully-connected layer can also be treated as a pooling layer which consists of several weighted average operations. If the pooling area is the whole feature map, several popular pooling functions are invariant to in-plane transforms. For example, the max pooling and the average pooling both pool a feature map regardless of the location of each element. On the contrary, the fully-connected operation does not have such a property in general.

Though the max pooling and average pooling are invariant to a range of deformations, they both have low discriminative power. For example, if we decrease all the elements except for the maximum one in a max pooling layer, the output is unchanged. And for the average pooling, the input feature map can vary significantly even if the average value remains. Our aim is to develop a pooling function that is invariant to deformations like the max pooling and average pooling, but also equipped with high discriminative power.

3.2. Linear Subspace Pooling

We now present the proposed pooling method. The basic idea is to model the convolutional feature maps with the linear subspace spanned by its principal components. Formally, we write the CNN features of an input patch as a matrix $F \in \mathbb{R}^{m\times p}$. Where each row i represents the i'th feature map stacked to a 1D vector, and each column corresponds to a spatial location of the feature maps. The CNN features can then be represented by the linear subspace spanned by r $(r < p, m)$ principal components of F. Specifically, let $U\Sigma V^T$ be the SVD of F, thus the columns of U corresponding to the largest r singular values give the r principal orthonormal bases. The pooled CNN features obtained in this manner are r-dimensional linear subspaces of the m-dimensional Euclidean space, which lie on the (m, r) Grassmann manifold [13, 37], denoted by $\mathcal{G}^r_m$. Limiting r to be smaller than p and m has two reasons. One is helpful to decrease the effect of noise, shading, occlusion, and other fine variations which are not useful for recognition. Another one is to reduce feature dimension. A point on the $\mathcal{G}^r_m$ manifold is generally represented by a matrix $Y \in \mathbb{R}^{m\times r}$ whose columns store an orthonormal basis of the subspace. Previously, such linear subspace representation is often used to build a robust model of an image set or a video sequence [38] in computer vision.

We can easily justify that the linear subspace representation is independent to column permutation, leading up to the following proposition.

Proposition 1. *The proposed subspace pooling is invariant to all the geometric changes that can be expressed as the column permutations of the matrix F.*

Proof. Let P be a permutation matrix and $U\Sigma V^T$ be the SVD of F, we have

$$FP = U\Sigma V^T P = U\Sigma(P^T V)^T, \tag{1}$$

thus $U\Sigma(P^T V)^T$ is the SVD of FP with the left singular vectors unchanged and the right singular vectors row-permuted. □

The main computation of the proposed subspace pooling is based on the singular value decomposition. However, back-propagation in neural network for SVD is nontrivial. Previously, such kinds of matrix back-propagation is explored in [18]. Here we directly write the conclusion for completeness and refer interested readers to [17, 18] for proof.

Proposition 2. *(Back-propagation) Let $F = U\Sigma V^T$ be the SVD with $X \in \mathbb{R}^{m\times p}$ and $m \geq p$, and $\Sigma_p \in \mathbb{R}^{p\times p}$ be the* *top p rows of Σ. Let $\frac{\partial \ell}{\partial U}$ be the gradient of the loss function $\ell : \mathbb{R}^n \to \mathbb{R}$ w.r.t. U and consider the block decomposition $\frac{\partial \ell}{\partial U} = \left(\left(\frac{\partial \ell}{\partial U}\right)_1 \middle| \left(\frac{\partial \ell}{\partial U}\right)_2 \right)$ with $\left(\frac{\partial \ell}{\partial U}\right)_1 \in \mathbb{R}^{m \times p}$, $\left(\frac{\partial \ell}{\partial U}\right)_2 \in \mathbb{R}^{m \times m-p}$. Denote $A_{sym} = \frac{1}{2}\left(A + A^T\right)$, A_{diag} be A with all off-diagonal elements set to 0, and $\circ$ be the element-wise product. Then the gradient of the loss function ℓ w.r.t. F is*

$$\begin{aligned}\frac{\partial \ell}{\partial F} &= DV^T + U\left(-U^T D\right)_{diag} V^T \\ &\quad + 2U\Sigma\left(K^T \circ \left(-D^T U \Sigma\right)\right)_{sym} V^T,\end{aligned} \tag{2}$$

where

$$D = \left(\frac{\partial \ell}{\partial U}\right)_1 \Sigma_p^{-1} - U_2 \left(\frac{\partial \ell}{\partial U}\right)_2^T U_1 \Sigma_p^{-1}, \tag{3}$$

and

$$K_{ij} = \begin{cases} \left(\sigma_i^2 - \sigma_j^2\right)^{-1}, & i \neq j \\ 0, & i = j \end{cases}. \tag{4}$$

3.3. Kernelized Subspace Pooling

There exist various distance metrics in the Grassman manifold and a good choice is the projection distance [13]. Specifically, for two points Y_1 and Y_2 on the $\mathcal{G}_m^r$ manifold, the projection distance is defined as

$$d_P(Y_1, Y_2) = 2^{-1/2} \left\| Y_1 Y_1^T - Y_2 Y_2^T \right\|_F, \tag{5}$$

or more conveniently, using the squared form

$$\begin{aligned} d_P^2(Y_1, Y_2) &= 2^{-1} \left\| Y_1 Y_1^T - Y_2 Y_2^T \right\|_F^2 \\ &= r - \left\| Y_1^T Y_2 \right\|_F^2 , \end{aligned} \tag{6}$$

where the last equation holds for the fact that $Y_1^T Y_1 = Y_2^T Y_2 = I_r$ and I_r is the identity matrix. The first formula in Equation (5) requires calculating the Frobenius L_2 distance between $Y_1 Y_1^T$ and $Y_2 Y_2^T$, both of which are $m \times m$ matrixes. Since $m > r$, the direct computation of $\left\| Y_1 Y_1^T - Y_2 Y_2^T \right\|_F$ is inefficient. Equation (6) shows that it is sufficient to compute only the Frobenius norm of $Y_1^T Y_2$, an $r \times r$ matrix, to compare Y_1 and Y_2. Though inefficient, the first formula implies that we can map Y_1 and Y_2 on the $\mathcal{G}_m^r$ manifold to the points $Y_1 Y_1^T$ and $Y_2 Y_2^T$ in the Euclidean space, and compare them using the Frobenius L_2 distance directly. We will show later that the form $Y_1 Y_1^T$ is also related to the bilinear CNN model. Mapping a point to the Euclidean space may be more convenient for some algorithms that adopt Euclidean structures such as norm and inner product. Nevertheless, many machine learning algorithms can be applied in the original manifold directly for various tasks such as clustering and classification [19].

To further increase the discriminative power of SP, we propose a simple distance kernel integrated to the marginal triplet loss which is helpful to focus more on the hard examples during CNN training. Specifically, we define the projection Gaussian kernel on the Grassman manifold as

$$k_P : (\mathcal{G}_m^r \times \mathcal{G}_m^r) \to \mathbb{R} : k_P(Y_1, Y_2) := e^{\frac{d_P^2(Y_1, Y_2)}{\gamma}}, \tag{7}$$

where $d_P^2(Y_1, Y_2)$ is defined in Equation (6) and $\gamma > 0$. For a given triplet (a, p, n) represents the anchor, positive and negative example, we consider the marginal triplet loss combined with the projection Gaussian kernel,

$$J(a, p, n) = \max(0, \mu + k_P(a, p) - k_P(a, n)), \tag{8}$$

where μ is the margin parameter. The Gaussian kernel function ($\gamma > 0$) is monotonic increasing and grows faster at larger distances. Thus if $d_P^2(a, p)$ is large (*i.e.*, hard positives), Equation (8) gives an even larger penalty; whereas if $d_P^2(a, n)$ is large (easy negatives), the loss becomes more subdued. That is, the Gaussian kernel helps the triplet loss focus more on hard examples to push the envelope further. We denote the pooling method together with the projection Gaussian kernel by *Kernelized Subspace Pooling* (KSP).

3.4. Connection to the Bilinear CNN model

Recently, the bilinear CNN model [22] has yielded impressive performance on a range of visual tasks. Given two CNN feature matrixes $F_1 \in \mathbb{R}^{m \times p}$ and $F_2 \in \mathbb{R}^{n \times p}$ organized in the same manner as described in Sec. 3.2. The bilinear CNN model forms a descriptor as

$$b(F_1, F_2) = F_1 F_2^T . \tag{9}$$

It can be easily verified that the bilinear pooling function is equivalent to spatial reordering. Thus the resulted descriptor has the same geometric invariant properties as ours described in Proposition 1.

From Equation (5) and (9) we can see that, the proposed subspace pooling together with the projection distance is equivalent to the bilinear pooling function, but using a single CNN. The bilinear CNN model incorporates two feature extractors, which can increase its discriminative power. However, the model size, memory consumption, and computational cost is roughly twice as its single counterpart. Another drawback of the bilinear CNN is that the produced descriptor is of high dimensionality. Equation (9) shows that the dimension of a bilinear descriptor $F_1 F_2^T \in \mathbb{R}^{m \times n}$ is the product of m and n. In addition, for a typical CNN architecture, like VGG [32] or ResNet [16], the number of feature maps m or n is often much larger than the number of spatial locations p after all the convolution layers, thus $F_1 F_2^T$ is rank deficient. Also, as analysed in Sec. 3.2, the feature maps may be contaminated by fine variations such as noise, shading and occlusion that are not useful for recognition. This motivates us to find a robust low-rank model,

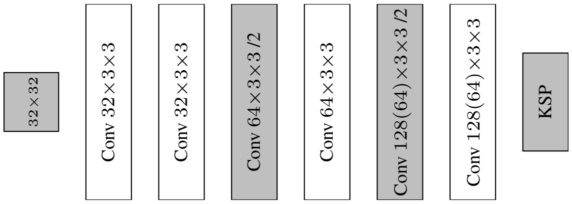


Figure 1: Network Architecture. Conv = Convolution + Batch Normalization + ReLU.

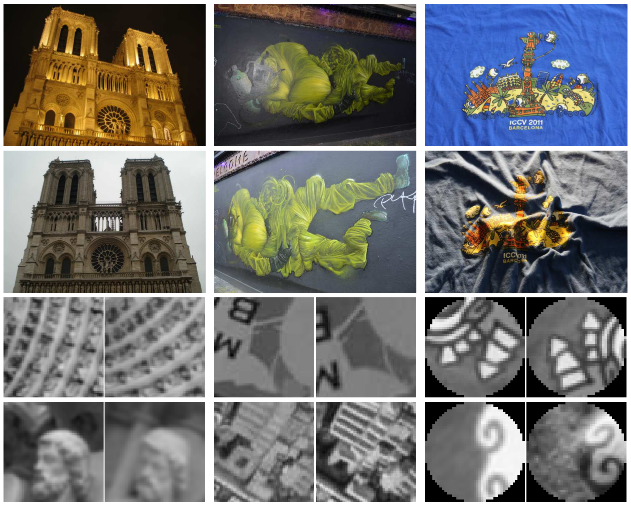
which comes up to the proposed subspace representation. We show that the performance of our method with a single CNN is comparable or superior to the bilinear CNN on several patch matching benchmarks, and a produced descriptor with much lower dimensionality.

4. Implementation

Network architecture. Figure 1 displays our network architecture. Our network consists of 6 convolution layers and a pooling layer. Batch normalization followed by ReLU non-linearity is added after each layer, except the last one. The spatial resolution is down-sampled twice, each of which is performed by the convolution with a stride of 2. This network is similar to the one adopted in the L2-NET and HARDNET, except for the last convolution layer which is replaced with the proposed subspace pooling layer. Our network takes a 32×32 image patch as input and outputs $128 \times 8 \times 8$ feature maps after all convolution layers. We then reshape the feature maps to a 128×64 matrix and extract 16 principal components for our subspace representation. Thus the dimension of the resulted descriptor is $128 \times 16 = 2048$. We also test an alternative configuration to reduce the feature dimension by decreasing the feature maps from 128 to 64 in the last two convolution layers and using 8 principal components in the pooling layer.

Loss function and data mining strategy. We use the marginal triplet loss with the projection Gaussian kernel defined in (8) to train our model. The projection distance $d_P^2(Y_1, Y_2)$ is first divided by r to be normalized in $[0, 1]$ and then modulated by the Gaussian kernel. We set the margin parameter μ and bandwidth of Gaussian kernel γ to 10 and 0.3 respectively for all our experiments except for specification. It is shown in previous works [30, 34, 25] that hard data mining is important to train the triplet objective. We employ the same mining strategy as HARDNET which is to find the hardest negative example for each patch in a mini-batch. This strategy enables the network to access the most useful training samples efficiently.

Training. We train our network from scratch using the PyTorch library. The network is optimized by SGD with a start learning rate at 0.1, momentum of 0.9 and weight decay of 0.0001. We train our model within 20 epochs and the learning rate is linearly decreased to zero. The data are augmented by random flipping and rotating 90^o online, which is the same setup as L2-NET and HARDNET.



(a) UBC [6] (b) HPatches [2] (c) DaLI [29]

Figure 2: Sample images and patches in the UBC, HPatches and DaLI datasets.

5. Empirical Evaluation

We conduct extensive experiments on widely used benchmarks, with particular emphasis on testing the invariant ability of descriptors to geometric and photometric changes. Sample images and patches in the benchmarks are displayed in Figure 2. We report the results of the proposed SP and KSP methods on all three benchmarks, together with their low-dimensionality counterparts which are denoted by SP* and KSP*, respectively. We also compare our methods to the original bilinear CNN model (BILINEAR). We use the same network architecture as ours for each of its two CNNs, and the same loss function, data mining strategy *etc*., to ensure a fair comparison.

5.1. UBC Dataset

Most of learning based descriptors report their results on the UBC PhotoTourism, also known as Brown's dataset [6]. It consists of three subsets: *Liberty*, *Notre-Dame*, and *Yosemite*. Keypoints are detected by the Difference of Gaussian (DoG) detector and correspondences are established by multi-view 3D reconstruction.

We use the standard evaluation protocol for the UBC dataset, which calculates the false positive rate (FPR) at the point of 95% true positive recall for the task of patch correspondence verification. Table 1 lists the results of proposed methods and state-of-the-arts using different subset as training data. Our KSP method outperforms state-of-the-arts in all categories. Our SP method achieves comparable

Training	Feature	*Notre-Dame*	*Yosemite*	*Liberty*	*Yosemite*	*Liberty*	*Notre-Dame*	Mean
Test	Dimension	*Liberty*		*Notre-Dame*		*Yosemite*		
SIFT [23]	128	29.84		22.53		27.29		26.55
MatchNet [14]	4096	6.90	10.77	3.87	5.67	10.88	8.39	7.74
DC-S2S [42]	512	6.45	11.51	3.05	5.29	9.02	10.44	7.63
D-Desc [30]	128	10.90		4.40		5.69		6.99
TNet-TGLoss [20]	256	9.91	13.45	3.91	5.43	10.65	9.47	8.80
TF-M [3]	128	7.22	9.79	3.12	3.85	7.82	7.08	6.47
L2-Net [34]	128	2.36	4.70	0.72	1.29	2.57	1.71	2.23
HardNet [25]	128	2.34	3.31	0.60	1.00	2.19	2.28	1.97
Bilinear	16384	1.39	2.06	0.38	0.59	1.68	1.53	1.27
SP	2048	1.56	2.10	0.39	0.62	1.70	1.92	1.38
KSP	2048	**0.82**	**1.36**	**0.29**	**0.47**	**0.60**	**0.51**	**0.68**
SP*	512	1.95	2.62	0.49	0.77	2.24	2.48	1.76
KSP*	512	1.63	2.11	0.48	0.62	1.47	1.38	1.28

Table 1: Performance on the UBC [6] dataset. Numbers are the false positive rate at 95% recall.

performance to the bilinear CNN but uses much lower dimensionality. It can also be seen that this dataset is nearly saturated due to the recent improvements on local descriptors learning. Therefore, we use more challenging datasets in the following sections for comprehensive comparison. In the rest of paper, we use our model trained on the *Liberty* sequence, which is a common practice [25] to allow a fair comparison.

5.2. HPatches Dataset

Recently, Balntas *et al.* [2] proposed a large dataset for local descriptor evaluation. It contains 116 sequences with 6 images for each. The dataset is split into two parts: viewpoint (Viewpt) - 59 sequences with significant viewpoint change and illumination (Illum) - 57 sequences with significant illumination change. Keypoints are detected by DoG, Hessian-Hessian and Harris-Laplace detectors in the reference image and projected to the rest of the images in each sequence, using the ground-truth homographies with 3 levels of geometric noise: Easy, Hard and Tough. The HPatches benchmark defines three tasks: patch correspondence verification, image matching, and patch retrieval.

Results are shown in Figure 3. Similar to UBC, HardNet was also the best performer on HPatches in previous literature, while Bilinear, the proposed SP and KSP all outperform it by a noticeable margin. Our KSP method achieves the best results on all three tasks. The advantage is larger on image matching and patch retrieval tasks where KSP outperforms HardNet 7 percentage points on both. SP and Bilinear achieve equal performance on the verification task while SP performs better on the image matching and patch retrieval task. This demonstrates that our SP descriptor is more compact and robust than that of Bilinear for the reason that SP only using the leading principal components. We also plot the patch retrieval accuracies when varying the number of distractors (non-matching patches) in Figure 4. The results of earlier CNN based methods DC-S2S, D-DESC and TF-M are remarkably better than handcrafted descriptors SIFT and RSIFT in the presence of low numbers of distractors. However, their accuracies degrade quickly as the size of the database grows. On the other hand, our SP and KSP outperform all the others and the differences are more significant in large numbers of distractors.

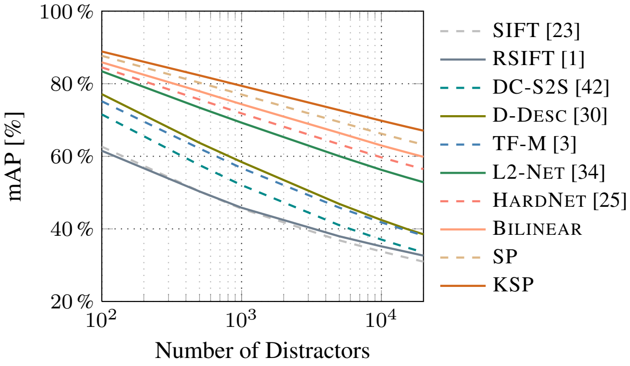


Figure 4: Performance (mAP) vs. the number of distractors, evaluated on the patch retrieval task of HPatches [2] dataset.

5.3. DaLI Dataset

In order to evaluate the non-rigid deformation and illumination invariant properties of local descriptors properly, Simo-Serra *et al.* [29] collected a dataset of deformable objects under varying illumination conditions. The dataset consists of 12 objects of different materials with four de-

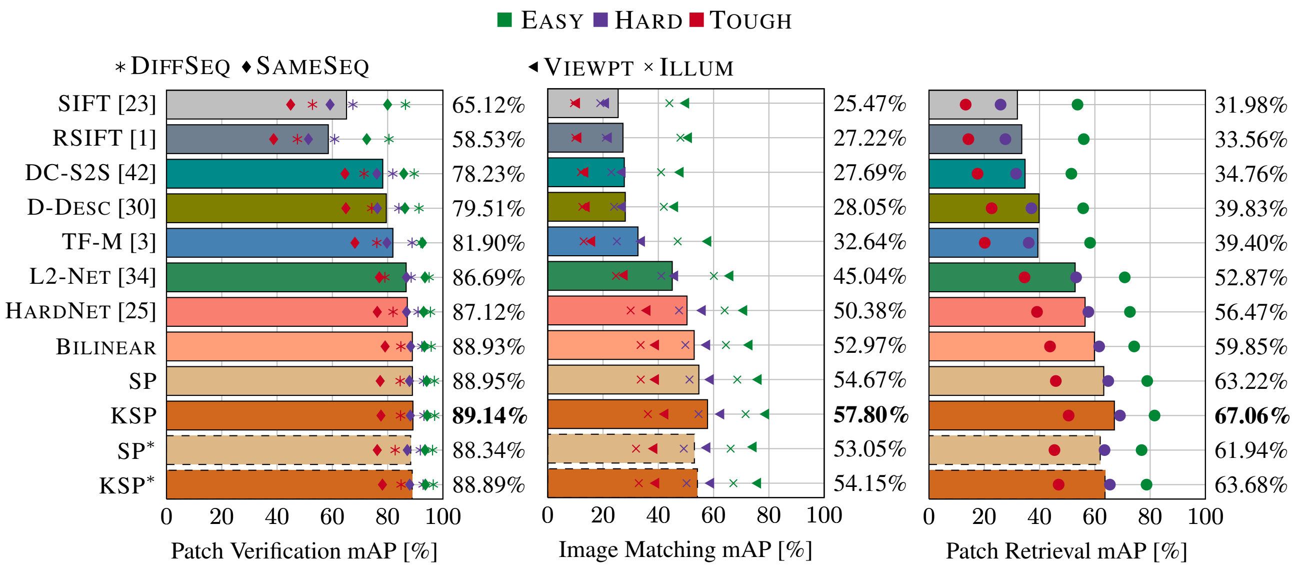


Figure 3: Verification, matching and retrieval results for HPatches [2] dataset. Colour of the marker indicates EASY, HARD, and TOUGH noise. The type of the marker corresponds to the variants of the experimental settings. The bar is the average of the variants of each task.

formation levels and four illumination conditions each, for a total of 192 images. Keypoints are detected by the DoG detector and correspondences are manually annotated.

We compare our methods to state-of-the-arts which have reported their performances on this dataset, including SIFT, DAISY, DALI and D-DESC. The standard evaluation protocol is spliting the dataset into deformation (*Def.*), illumination (*Ill.*) and deformation with illumination (*Def.+Ill.*) subsets. The results are shown in Table 2. Our SP and K-SP methods significantly improve on previous results in all three splits. Figure 5 displays the results for the deformation subset. As can be seen that our KSP performs best in all deformation levels and shows more advantage on higher-level deformations. What is worth to notice is that the image patches in this dataset are cropped to be circular, while our descriptors are trained on square patches. This shows that our methods have good generalization ability.

Descriptor	Training	*Def.*	*Ill.*	*Def.+Ill.*
SIFT [23]	-	55.82%	60.76%	53.43%
DAISY [35]	-	67.37%	75.40%	66.20%
DALI [29]	-	70.58%	89.90%	72.91%
D-DESC [30]	*Lib.+Yos.*	76.57%	88.43%	75.93%
BILINEAR	*Lib.*	81.12%	92.48%	81.29%
SP	*Lib.*	82.08%	95.07%	83.00%
KSP	*Lib.*	**83.60%**	**97.17%**	**84.81%**
SP*	*Lib.*	81.31%	92.84%	81.12%
KSP*	*Lib.*	81.61%	95.17%	82.55%

Table 2: Results on the DaLI [29] dataset. We show the mean accuracy of descriptor matches and highlight the top-performing descriptor for each of setting, in bold.

Descriptor	Verification	Matching	Retrieval
MP	84.63%	39.54%	48.90%
AP	86.08%	45.01%	54.61%
SP	88.95%	54.67%	63.22%

Table 3: Subspace pooling (SP) compared to max pooling (MP) and average pooling (AP).

5.4. Model Analysis

Compared to max pooling and average pooling. The max pooling (MP) and average pooling (AP) are also invariant to some geometric changes when used as global operations. Here we replace the proposed SP with the MP and AP and evaluate their performances on the HPatches dataset. As can be seen from Table 3 that the results of MP and AP are inferior to our SP due to their low discriminative abilities.

The influence of subspace dimension. As analysed in previous sections, one reason that we choose r principal components for our subspace representation rather than all the orthonormal bases is to reduce the effect of small useless variations in the CNN features. This section explores the performance of the SP descriptor w.r.t. the number of principal components. Figure 6 (a) shows the experimental

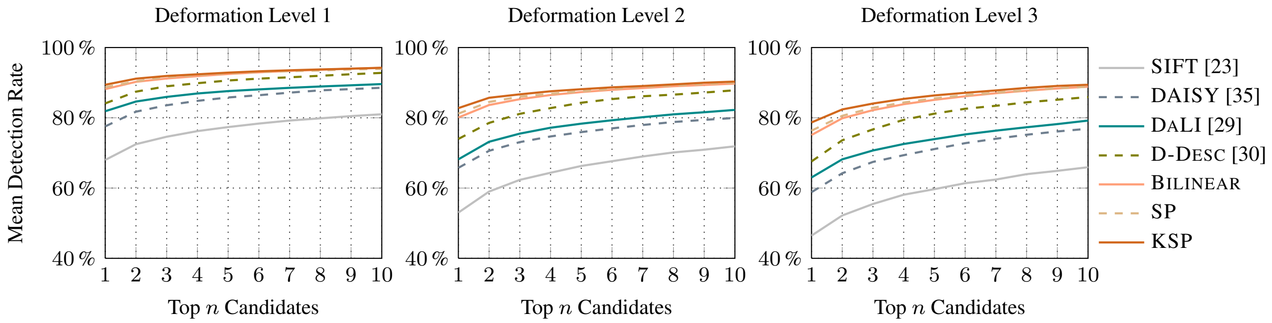


Figure 5: Results when increasing the deformation levels while keeping the illumination constant on the DaLI [29] dataset.

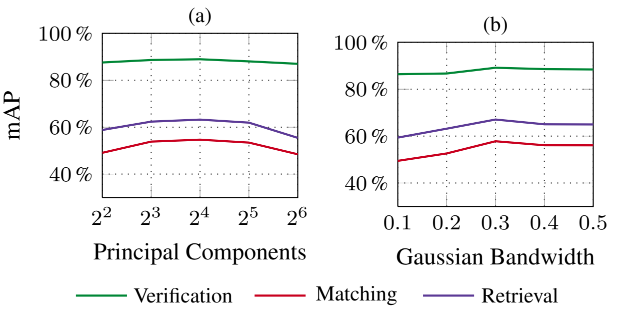


Figure 6: The effect of subspace dimension for SP and the bandwidth of the projection Gaussian kernel for KSP.

results. Using smaller or larger amount of principal components are both harmful to the SP descriptor. A good compromise on performance and feature size is achieved when using 16 principal components, while decreasing the number of principal components will lose valuable information and increasing it introduce trivial details.

The influence of Gaussian bandwidth. All experiments on the UBC, HPatches and DaLI datasets show that KSP consistently performs better than SP. The projection Gaussian kernel adopted in KSP contains one parameter γ that determines the bandwidth of the Gaussian distribution. In fact, γ can be learned from data, using the standard BP training method for CNN. However, our preliminary experiments show that the learned γ causes overfitting on several datasets. Thus this paper treats γ as a free parameter. Figure 6 (b) displays the performance of KSP w.r.t. several γs.

Performance on another network architecture. The proposed subspace pooling method is applicable to a given CNN architecture to increase its invariant ability. Here we test another network which is adopted in the TF-M [3] descriptor. We replace the fully-connected layer in TF-M with our subspace pooling layer and reserve other settings. As can be seen from Table 4 that our proposed methods remarkably improve the performance of TF-M.

Descriptor	Verification	Matching	Retrieval
TF-M [3]	81.90%	32.64%	39.40%
TF-M+SP	82.54%	36.07%	43.25%
TF-M+KSP	82.78%	38.42%	45.68%

Table 4: The performance of TF-M can be significantly improved when incorporating our subspace pooling layer.

6. Concluding Remarks

This paper proposes a pooling method to learn invariant and discriminative descriptors using CNNs. We first analyse that the convolution operation in CNNs is not invariant to various geometric changes. In order to achieve such invariance, pooling methods are necessary. However, the widely used max pooling and average pooling have low discriminative power when adopted as global operations. The proposed subspace pooling (SP), on the contrary, has both highly invariant and discriminative power. The proposed SP, as well as the max pooling, average pooling and bilinear CNN, do not use any spatial information of the CNN features which makes it convenient to handle complex transforms that do not have a parametric model, *e.g.*, the non-rigid deformation. In fact, this representation is more reasonable for a local descriptor rather than the global description of a large region. Especially for scenes obtained from certain situations such as vision-based navigation, where the relative positions of feature elements provide useful information. A potential solution to describe relationships of feature elements is to use graph embedding techniques. We will explore this direction in the future.

Acknowledgement

This work was supported by the National Basic Research Program of China (Grant No.2015CB351705), and the State Key Program of National Natural Science Foundation of China (Grant No.61332018).

References

[1] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012.

[2] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. H-patches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, 2017.

[3] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *BMVC*, 2016.

[4] V. Balntas, L. Tang, and K. Mikolajczyk. Bold-binary online learned descriptor for efficient image matching. In *CVPR*, 2015.

[5] H. Bay, T. Tuytelaars, and L. J. V. Gool. SURF: speeded up robust features. In *ECCV*, 2006.

[6] M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *IEEE PAMI*, 33(1):43–57, 2011.

[7] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *IJCV*, 74(1):59–73, 2007.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[9] Y. Duan, J. Lu, Z. Wang, J. Feng, and J. Zhou. Learning deep binary descriptor with multi-quantization. In *CVPR*, 2017.

[10] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE PAMI*, 35(12):2916–2929, 2013.

[11] R. C. González and R. E. Woods. *Digital image processing, 3rd Edition*. Pearson Education, 2008.

[12] I. J. Goodfellow, Y. Bengio, and A. C. Courville. *Deep Learning*. Adaptive computation and machine learning. MIT Press, 2016.

[13] J. Hamm and D. D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *ICML*, 2008.

[14] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *CVPR*, 2015.

[15] T. Hassner, V. Mayzels, and L. Zelnik-Manor. On sifts and their scales. In *CVPR*, 2012.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[17] C. Ionescu, O. Vantzos, and C. Sminchisescu. Matrix backpropagation for deep networks with structured layers. In *ICCV*, 2015.

[18] C. Ionescu, O. Vantzos, and C. Sminchisescu. Training deep networks with structured layers by matrix backpropagation. *CoRR*, abs/1509.07838, 2015.

[19] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi. Kernel methods on riemannian manifolds with gaussian rbf kernels. *IEEE PAMI*, 37(12):2464–2477, 2015.

[20] V. Kumar B G, G. Carneiro, and I. Reid. Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In *CVPR*, 2016.

[21] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[22] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *ICCV*, 2015.

[23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[24] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE PAMI*, 27(10):1615–1630, 2005.

[25] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *NIPS*, 2017.

[26] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robotics*, 31(5):1147–1163, 2015.

[27] J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Descriptor learning for efficient retrieval. In *ECCV*, 2010.

[28] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, 2011.

[29] E. Simo-Serra, C. Torras, and F. Moreno-Noguer. Dali: deformation and light invariant descriptor. *IJCV*, 115(2):136–154, 2015.

[30] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *ICCV*, 2015.

[31] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *IEEE PAMI*, 36(8):1573–1585, 2014.

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[33] C. Strecha, A. Bronstein, M. Bronstein, and P. Fua. Ldahash: Improved matching with smaller descriptors. *IEEE PAMI*, 34(1):66–78, 2012.

[34] Y. Tian, B. Fan, and F. Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *CVPR*, 2017.

[35] E. Tola, V. Lepetit, and P. Fua. A fast local descriptor for dense matching. In *CVPR*, 2008.

[36] T. Trzcinski, M. Christoudias, V. Lepetit, and P. Fua. Learning image descriptors with the boosting-trick. In *NIPS*, 2012.

[37] P. Turaga, A. Veeraraghavan, and R. Chellappa. Statistical analysis on stiefel and grassmann manifolds with applications in computer vision. In *CVPR*, 2008.

[38] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. *IEEE PAMI*, 33(11):2273–2286, 2011.

[39] Z. Wang, B. Fan, and F. Wu. Local intensity order pattern for feature description. In *ICCV*, 2011.

[40] Z. Wang, B. Fan, and F. Wu. Affine subspace representation for feature description. In *ECCV*, 2014.

[41] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. Lift: Learned invariant feature transform. In *ECCV*, 2016.

[42] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, 2015.

[43] J. Zbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. In *CVPR*, 2015.