HAL
open science

Robust Angular Local Descriptor Learning

Yanwu Xu, Mingming Gong, Tongliang Liu, Kayhan Batmanghelich, Chaohui Wang

To cite this version:

Yanwu Xu, Mingming Gong, Tongliang Liu, Kayhan Batmanghelich, Chaohui Wang. Robust Angular Local Descriptor Learning. Asian Conference of Computer Vision 2018, Dec 2018, Perth, Australia. pp.420-435, 10.1007/978-3-030-20873-8_27 . hal-03133246

HAL Id: hal-03133246

https://hal.science/hal-03133246v1

Submitted on 5 Feb 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Robust Angular Local Descriptor Learning*

Yanwu Xu[1,3], Mingming Gong[1], Tongliang Liu[2], Kayhan Batmanghelich[1], and Chaohui Wang[3]

[1] University of Pittsburgh, 4200 Fifth Avenue Pittsburgh, PA 15260, USA
{yanwuxu,mig73,kayhan}@pitt.edu
[2] The University of Sydney, Camperdown NSW 2006, Australia
tongliang.liu@sydney.edu.au
[3] Université Paris-Est, LIGM (UMR 8049), CNRS, ENPC, ESIEE Paris, UPEM, Marne-la-Vallée, France
chaohui.wang@u-pem.fr

Abstract. In recent years, the learned local descriptors have outperformed handcrafted ones by a large margin, due to the powerful deep convolutional neural network architectures such as L2-Net [1] and triplet based metric learning [2]. However, there are two problems in the current methods, which hinders the overall performance. Firstly, the widely-used margin loss is sensitive to incorrect correspondences, which are prevalent in the existing local descriptor learning datasets. Second, the L2 distance ignores the fact that the feature vectors have been normalized to unit norm. To tackle these two problems and further boost the performance, we propose a robust angular loss which 1) uses cosine similarity instead of L2 distance to compare descriptors and 2) relies on a robust loss function that gives smaller penalty to triplets with negative relative similarity. The resulting descriptor shows robustness on different datasets, reaching the state-of-the-art result on Brown dataset , as well as demonstrating excellent generalization ability on the Hpatches dataset and a Wide Baseline Stereo dataset.

Keywords: Local descriptor · CNNs · Robust loss.

1 Introduction

Finding correspondences between local patches across images is an important component in many computer vision tasks, such as image matching [3], image retrieval [4] and object recognition [5]. Since the seminal paper introducing SIFT [6], local patches have been encoded into representative vectors, called descriptors, which are designed to be invariant/robust to various geometric and photometric changes such as scale change, viewpoint change, and illumination change.

Given the success of deep learning, hand-crafted descriptors such as SIFT have been outperformed by learned ones [7–9]. Different from the hand-crafted

* Supported by grant Pfizer and organization by SAP SE and CNRS INS2IJCJC-INVISANA.

descriptors which extract low-level features such as gradients, the learned descriptors learn a convolutional neural network (CNN) from raw patches with ground-truth correspondences. These descriptor learningnetworks are trained by metric learning losses and can be divided into two cat-egories by whether there are learnable distance comparison layers in the network. The networks with distance comparison layers output distances directly without explicit descriptors [9–11]. This type of networks showed promising performance in patch verification but cannot be combined with nearest neighbor search. Recently, networks without similarity comparison layers achieved better performances due to more advanced network architectures such as L2Net [1] and training techniques such as triplet loss with hard negative mining [2]. These networks output descriptors which can be compared using simple L2 distance and be matched using fast approximate nearest neighbor search algorithms like Kd-tree [12].

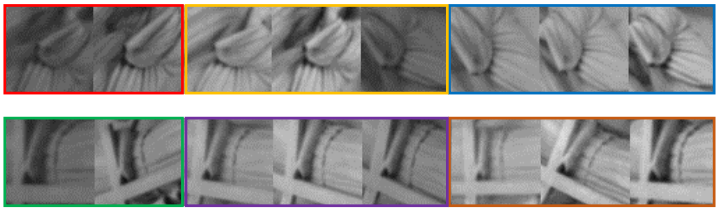


Fig. 1. Examples of false labeled patches, the patches sharing same label of 3D view point are marked by same color box,and different color boxes come from different 3D view point.

In the descriptor learning networks, the metric learning loss function and the distance/similarity measure between descriptors are two essential components. State-of-the-art methods usually adopt margin-based losses such as the hinge loss [2] to train the descriptor learning networks. Because the number of negative pairs is huge, batch hard negative mining is usually applied to stabilize the training process as well as reduce the computational load [2, 13]. However, the current triplet losses are not robust to the incorrect correspondences (outliers) in the training data, as shown in Fig 1. The patches at different locations (negative pairs) can exhibit strong similarities and the patches at the same location (positive pairs) can be very different due to local distortion or corruptions. Additionally, since the local descriptors are normalized to unit norm before comparison, L2 distance is no longer an appropriate distance measure to compare descriptors.

To target the these two problems, we propose a robust angular loss to train the descriptor learning networks which is called RAL-Net. Instead of using the hinge loss as done in [2], we propose a robust loss function which gives bounded penalty to the triplets with incorrect correspondences. In addition, we propose to utilize cosine similarity to compare two descriptors, which is more appropriate to compare unit-norm vectors. We train our descriptor on the Brown dataset [14] and obtain state-of-the-art results using the same training strategy as [2].

Moreover, our descriptor performs much better than [2] when the sample size and batch size is small, which further veri

fies the effectiveness of the proposed method. Our codes is released in github[4]

2 Related works

Recent work on local descriptor designing has gone through a huge change from conventional hand-crafted descriptors to learning-based approaches, which ranges from SIFT [6] and DAISY [15] to latest methods such as DeepCompare, MatchNet, and HardNet [7–9, 2]. As for deep learning-based descriptors, there are two study trends including CNN structure designing and negative sampling for embedding learning.

Before CNN models being broadly applied, descriptors learning methods were limited to specific machine learning descriptors. Therefore, there were various kinds of methods inspired by different aspects. Principal Components Analysis (PCA) based SIFT (PCA-SIFT) [16] leads to normalized gradient patch compared to SIFT histograms of gradients. [14] proposed a filter with learned pooling and dimension reduction. Simonyan et al. [11] studied convex sparse learning to learn pooling fields for descriptors. Aside from these descriptors, [17] raised an online search method from a subset of tests which can increase inter-class variance and decrease intra-class variance. One thing these methods have in common is that they all rely on shallow learning architectures.

In the past few years, models based on CNN try to get better performance by designing various convolutional neural network architectures, e.g. [9, 10]. [9] choose a two-branched network, a typical Siamese structure for feature extraction and three full connected layers for deep metric learning. [10] explored further on Siamese network with two branches sharing no parameters and proposed a two-channel input structure which is stacked by center cropped patches and plain patches.

Recently methods focused more on loss function design because improving network structure can not give birth to significant improvement of descriptors as before. These works on learning embedding can be summarized as classification loss, contrastive loss, and triplet loss. [18, 19] proved the validity classification loss for face recognition ans scene recognition. As the most common pairwise loss, contrastive loss [20, 21] aims at increase all of the similarity of positive pairs and push away the negative pairs until bigger than a variant margin. [22] proposed a restricting two sides margin for contrastive learning and this method not only requires distance between positive pairs above the margin but also limits distance between positive pairs under the second margin. Compared to contrastive loss, triplet loss cares about relative similarity between positive pairs and negative pairs rather than absolute value which consists of anchor positive (a, p) and anchor negative (a, n) with a shared point anchor a. This method can meet with most of tasks involving large scale embedding learning [23]. However, there is still

[4] https://github.com/xuyanwu/RAL-Net

a great challenge in choosing hinge margin for triplet loss, as well as searching proper negative pairs fed into triplet loss. Distinguished from this embedding learning methods, Histogram loss [24] uses a quadruplet based sampling strategy by estimating distribution of similarity between positive pairs and negative pairs.

From previous studies, the loss functions have their advantages and disadvantages in different learning tasks. A indispensable component of these methods is the negative sampling strategy. In this paper, our main aim is to improve the loss function to further improve the performance of local descriptor learning. Our method can possibly be applied to other related tasks such as face recognition and person re-identification.

3 Proposed method

In this section, we will discuss the form of our robust triplet loss which similar to [25] and simply introduction to network structure which is based on [1]. In order to explain our loss function, we first review the general forms of triplet loss and contrastive loss then present our difference.

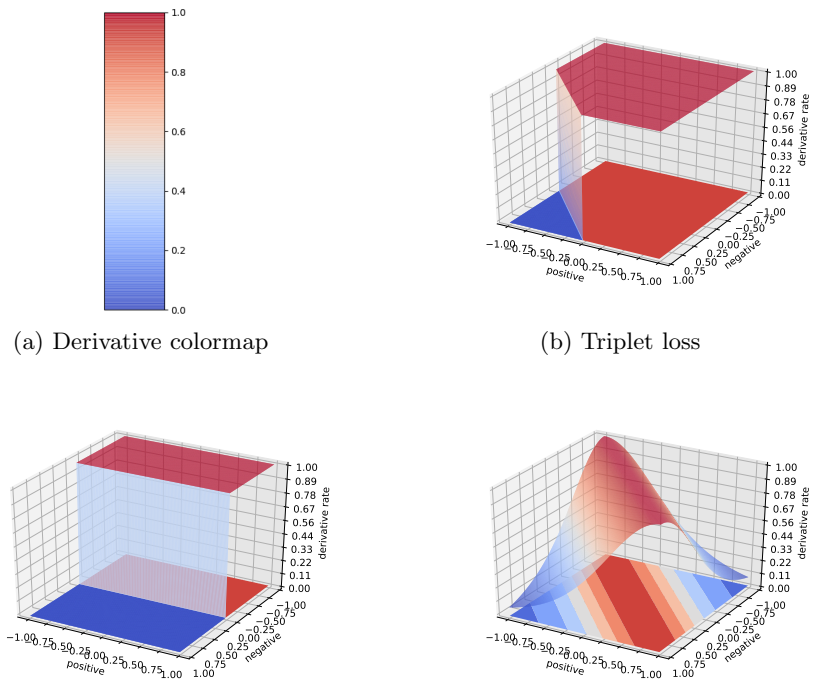
3.1 loss function

3.2 The triplet loss

Triplet loss has been successfully applied to many tasks, such as image matching, image retrieval, and face identification. The idea is to make positive pairs closer and keep relative negative pairs away from the positive. The very common expression of triplet loss is formulated as follows:

$$L_{triplet} = \frac{1}{N} \sum_{i,j,k} [s(a_i, n_k) - s(a_i, p_j) + m]_+ \tag{1}$$

where a, p and n represent anchor, positive and, negative of triplet tuple and operator $[l]_+$ means $max(l, 0)$ and function $s(x, y)$ represent similarity score between two featrues. Due to the large amounts of combination among a, p and n, the back propagation of loss is very time-consuming. Thus, an indispensable component is to sample hard negatives for both performance improvement and computation reduction. In the context of descriptor learning, the recent HarNet [2] method searches for the most difficult negative pair with reference to each achor positive pair. However, the hard negative sampling strategy in [2] is unable to fully explore the negative pairs because only negative pairs that share an element with the anchor positive pairs are considered. Due to the margin m, if the similarity between positive pairs and negative pairs is bigger than margin and then the derivation of triplet item will be 0. This will cause information loss, but triplet can help learn a better distribution of descriptors.



(a) Derivative colormap

(b) Triplet loss

(c) Contrastive loss

(d) Robust loss

Fig. 2. This figure depicts the derivation of three different loss function, (b)triplet loss,(c)contrastive loss,(d)robust loss. the cosine similarity of positive pairs and negative pairs represent x axis and y axis,where z axis denotes the absolute derivative values with respect to the change of x and y. The derivative value becomes larger when approaching red, vice versa when approaching blue.

3.3 The contrastive loss

The difference between the contrastive loss and the triplet loss is that triplet loss aims at comparing relative similarity between positive pairs and negative pairs, while contrastive loss only compares negative pairs with margin and pull positive pairs as close as possible. The general form of contrastive loss is formed as follows:

$$L_{contrastive} = \frac{1}{N} \sum_{i,j,k,l} ([m + s(a_k, n_l)]_+ - s(a_i, p_j)) \tag{2}$$

In contrastive loss, the number of training data pairs grows quadratic with respect to training sample size. Therefore, it is much easier to sample the data pairs than triplet loss, and random sampling is often employed for contrastive loss-based learning. However, as shown in previous works, contrastive loss showed

inferior performance to triplet loss in certain tasks. But [26] argues that the inferior performance of contrastive learning is due to the inappropriateness of the random sampling strategy. When combined with the proposed negative sampling strategy, contrastive loss performs as well as triplet loss in the descriptor learning task. The common difficulty for triplet and contrastive loss is that margin cause a great impact in result.

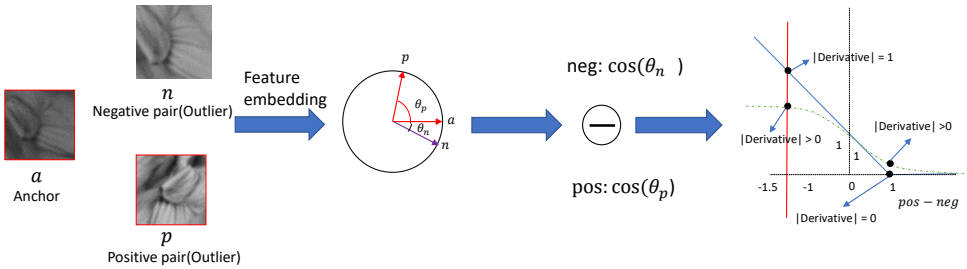


Fig. 3. Proposed descriptor learning model, the negative mining strategy is same as [2]. The features extracted from patches are located in embedding space and cosine similarity between positive pairs is $\cos\theta_p$ and negative pairs are $\cos\theta_{n_1}$ and $\cos\theta_{n_2}$.The curves in the right side represent triplet margin function(solid line),contrastive margin function(dash line) and our robust function(dash-doted line).

As for triplet loss or contrastive loss, these methods all consist of positive pairs, negative pairs and a discriminative margin. The keypoint is the negative search strategy and margin choosing, the later one of which is tricky for metric learning. Thus we propose a robust loss without margin confusion which can keep more relative embedding information as well as applying cosine similarity for our metric learning which is similar to cosine face [27],cause the angle distace is closer to original embedding distribution as a hypersohere. As shown in Fig.2, assuming margin values for triplet loss and contrastive loss are 1 and 0 separately which are common margin choice. Due to the margin strategy, there is always a part of selected triplet or contrastive items with no derivation, also the selected negative pairs are too sparse, resulting in a large selection bias in the sampling process. Explained in [25], robust regression and classification problem often requires non-convex loss function which can prevent scalable and global training where a natural approach to implement it is cutting out loss vales that exceed threshold, which is similar to triplet and contrastive loss. However, a relaxation of this kind of 'clipping loss' can improve robustness. As for our embedding metric learning problem, we can adopt this intuition.

Thus, we propose a robust version of triplet loss which can offset bias problem to some extent. We consider that selected positive pairs $(a_i, p_i$ and negative pairs (a_i, n_i) should be more important when when their similarity are high and vice versa given less weight rather than set their derivation to be 0. However, we

observed that a considerable amount false negative labels exiting which should be positive pairs but marked as negative which is shown in Fig.3, therefore, we put less weight to triplet items when the cosine distance between negative pairs (a_i, n_i) is much more bigger than positive pairs $(a_i, p_i$, the derivation of which should be similar to symmetrical arch bridges as shown in Fig.2 (d). Our method performs obviously better when training data by small batch where the effect of bias influences more and reach the best for bigger batch size, which will be discussed later.

We apply the same positive pairs and negative pairs sampling strategy as [2] and the features generated by networks are 128-D and euclidean normalized with length 1. We define a search procedure S_i starting with anchors (a_i and we search for all of the positive and negative items related to a_i. With regard to a batch which consists of N pairs of matched patches a and p from different given 3D point of view of descriptors,the size of descriptors matrices A and P are $N \times 128$, $N \times N$ cosine cosine similarity matrix $\boldsymbol{D} = A \times P^T$, so there is only one positive item $pos_i = d(a_i, p_i)$ in the diagonal of $\boldsymbol{D}$ for each search S_i. The goal is to find the closest negative item with respect to a_i and p_i.As we know, the bigger the gap between two features, the smaller the cosine similarity is. Please refer to [2] for more sampling details, the formula is organized as follow:

$$\begin{aligned}
&\boldsymbol{D}(i,j) = \cos\theta_{i,j} = a_i \cdot p_j, \\
&pos_i = \boldsymbol{D}(i,i), \\
&neg_i = \boldsymbol{D}(r_i, c_i), \\
&(r_i, c_i) = \mathrm{argmax}_{k,l}\boldsymbol{D}(k,l), \\
&s.t. \quad k, l = 1...n, \\
&\qquad \{k = i \vee l = i\} = True, \\
&\qquad k \neq l.
\end{aligned} \tag{3}$$

Finally, the triplet items are fed into loss function formed as follow,and our goal is to minimize this loss for each batch. Also, the derivation of L_i with respect to $(pos_i - neg_i)$ is even function as explained above in Fig.2, and as for better description,Fig.2 demonstrate the absolute value of derivation.

$$\begin{aligned}
&L_{robust} = \frac{1}{N}\sum_{i=1}^{N}(1 - \tanh(pos_i - neg_i)), \\
&L_i = 1 - \tanh(pos_i - neg_i), \\
&\frac{\partial L_i}{\partial(pos_i - neg_i)} = \tanh^2((pos_i - neg_i)) - 1.
\end{aligned} \tag{4}$$

3.4 Network structure

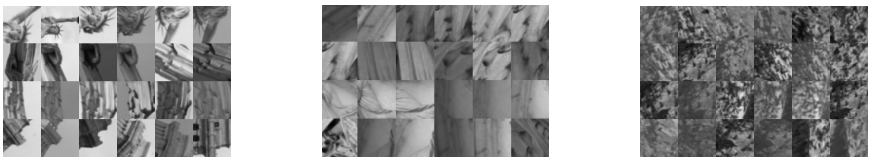
Following [2], we adopt the L2Net [1] architecture as our main network. The network consists of two parts, the main feature extraction network and the linear

decision layer which reduces the feature dimension. For fair comparison, we also make slight modifications by adding 0.3 dropout layer above the bottom layer of network as [2]. For an 32×32 normalized single-channel patch, the output is a L2 normalized 128-dimensional feature vector. It is worth noting that the whole feature extraction network is built by full convolutional layers, downsampling by two-stride convolutional net. Also, and there is a BN layer and a Relu activation layer in every layer except the last layer, except the bottom layer with no Relu layer. And the whole network is trained with the proposed negative sampling procedure and corresponding loss functions.

4 EXPERIMENTAL RESULTS

We train and test our RAL-Net descriptor on the Brown dataset and test the patch verification performance on Brown dataset. In addition, we use the models trained on the Brown dataset to test its generalization abilities in patch verification, patch matching, and patch retreival on the Hpatches dataset. Finally, we apply our RAL-Net descriptor on the Wide Baseline Stereo dataset to test its invariance properties.

4.1 Brown dataset



(a) Liberty (b) Notredame (b) Yosemite

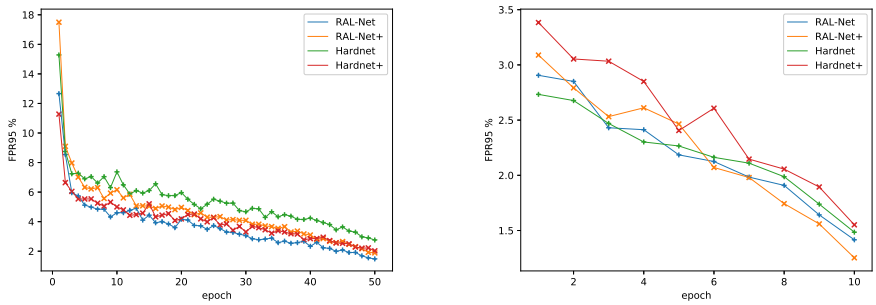
Fig. 4. Subsets of Brown

The Brown dataset is the most popular local descriptor learning dataset, which contains three subsets of images taken from different places, including Liberty, Notredame and Yosemite. Keypoints are firstly detected by Difference of Gaussians (DOG) [6] and then vefified with ground truth 3-D view. The patches are extracted around the keypoint locations and are normalized by scale and orientation calculated during keypoint detection. There are about 400k classes of patch pairs with 64×64 size, extracted from different sequences. In practice, the size of 64×64 is unnecessary, and we resized the patches to 32×32 by linear cubic interpolation.

Training setting On the three Brown subsets, we trained our RAL-Net descriptor in different setting with different training sample size and batch size. In the first

setting, we only extracted 200K pairs in total for each subsect repectively. In this setting, we compared the performance of our descriptor with the state-of-the-art HardNet trained with a small batchsize 128. The performance with a small batch size can desmonstrate the effectiveness of the negative sampling strategies. We applied the training strategy different from [2] and [1], which trains data for 50 epochs with learning rate linearly decreasing to 0 in the end. We choose Stochastic Gradient Descent (SGD) as our optimizer and we set the initial learning rate to be 10, and the rest momentum to be 0.9 and weight decay to be 0.0001. In addition, we have tried Adam optimizer and it converge faster than SGD, however SGD can achieve a better result with well chosen training parameters.

In the second setting, which is the standard setting, we extracted 5000K pairs and trained descriptor with batch size 512. As for HardNet, we trained it using batch sizes 512 and 1024, as the performance of HardNet is more sensitive to batch size. In this setting, due to big amount of data, the training is done within 10 epoch which is much less than strategy one but the other training aspects are the same as strategy one. In order to compare our RAL-Net descriptors, we also apply the same training strategy on HardNet and cite several result of recent works. Following previous works, we also applied data augmentation by random filpping and 90° rotation in both training settings.



(a) Training strategy 1 (b) Training strategy 2

Fig. 5. The curves describe the result of FPR95 of Hardnet and our RAL-Net tested in Brown dataset.X axis is training epochs and Y is the percent value of FPR95.

Overall evaluation The descriptors are trained on one subset and tested on the rest two subsets. As for evaluation, tested subset contains 100k pairs of patches for each subset with 50K matched and 50K unmatched labels. We follow the evaluation protocol [14] and give the results of false positive rate FPR at the recall of 95% true positive rate TPR (FPR95).The training precision alone training epoch is demonstrated in Fig.5 and the results are shown in Table.1, the best results are shown in bold.

Table 1. Descriptor performance on Brown dataset for patch verification. False positive rate at 95% true positive rate is displayed. Results of the best are in bold and "+" suffix represent training implemented by data augmentation of random flipping and 90° rotating.

Training	Notredame	Yosemite	Liberty	Yosemite	Liberty	Notredame	Mean
Test	Liberty		Notredame		Yosemite		FPR
SIFT[6]	29.84		22.53		27.29		26.55
MatchNet [9]	7.04	11.47	3.82	5.65	11.6	8.7	8.05
L2Net[1]	3.64	5.29	1.15	1.62	4.43	3.30	3.23
L2Net+[1]	2.36	4.70	0.72	1.29	2.57	1.71	2.22
CS L2Net[1]	2.55	4.24	0.87	1.39	3.81	2.84	2.61
CS L2Net+[1]	1.71	3.87	0.56	1.09	2.07	1.3	1.76
HardNetNIPS [2]	3.06	4.27	0.96	1.4	3.04	2.53	2.54
HardNet+NIPS [2]	2.28	3.25	0.57	0.96	2.13	2.22	1.9
Traing strategy 1: 200K training pairs for each subset, batch size 128							
HardNet_{128}	2.07	3.70	0.77	1.22	3.79	3.33	2.48
HardNet_{128}+	2.46	3.55	0.73	1.67	3.54	3.40	2.56
RAL-Net_{128}(ours)	**1.46**	**2.63**	**0.51**	**0.91**	**1.95**	**1.40**	**1.48**
RAL-Net_{128}+(ours)	1.81	3.80	0.55	1.01	1.96	2.18	1.89
Traing strategy 2: 5000k training pairs for each subset, batch size 512							
HardNet_{512}	1.54	2.56	0.63	0.92	2.65	2.05	1.73
HardNet_{512}+	2.53	2.69	0.54	0.83	2.49	1.70	1.80
HardNet_{1024}	1.47	2.67	0.62	0.88	2.14	1.65	1.57
HardNet_{1024}+	1.49	2.51	0.53	0.78	1.96	1.84	1.51
RAL-Net_{512}(ours)	1.44	2.60	0.48	0.77	1.77	1.43	1.42
RAL-Net_{512}+(ours)	**1.30**	**2.39**	**0.37**	**0.67**	**1.52**	**1.31**	**1.26**

Obviously, our RAL-Net generate the overall best results among all of the representative descriptors as well as the best among the testing subsets. Deep model-based descriptors have surpassed far more than hand-crafted ones, and focus on comparison between our descriptor and the HardNet descriptor.

In the first training setting, it is interesting that RAL-Net descriptor achieves better results than the HardNet descriptor when both of them are trained with 200K pairs with batch size of 128. Furthermore, our descriptor trained with less data achieves comparable results as Hardnet trained on 5000K data pairs with 512 batch size. We can also notice that data augmentation shows no better effect for small training sample size and even slightly worsen the performance due to the increasing difficulties of negative sampling and less training data with bigger bias. However, the results on the small training dataset and small batchsize verifies the effectiveness of our proposed robust loss training.

In the second setting, we obtain the best results on 5000K data pairs with a batch size of 512. Even in the training without data augmentation, RAL-Net

exceeds Hardnet with batch size 512 as well as 1024. It is worth noting that Hardnet descriptor gets improved when enlarging batch size from 512 to 1024, while increasing batchsize from 512 to 1024 leads to almost no enhancement for our descriptor. Thus, we only report the results of our Ral-Net trained with batch size of 512. The experimental results confirm the efficiency and validity of our RAL-Net descriptor. For the rest of experiments, we test HardNet and our descriptor on other datasets by training them on 5000K data pairs with 512 batch size on the Liberty subset.

Ablation studies Proving the effectiveness of angular and robust form separately with respect to our RAL loss. We set four variants based on our training strategy 1 without augmentation due to limited time, which are HardNet with angular embedding, HardNet with robust form, original HardNet and our RAL-Net respectively. The result is shown in Table.2. The results indicate that both the angular distance and the robust loss function contribute to the overall performance and the combination of them achieves state-of-the-art performance.

Table 2. Comparision between different combinations.

Training	Notredame	Yosemite	Liberty	Yosemite	Liberty	Notredame	Mean
Test	Liberty		Notredame		Yosemite		FPR
Traing strategy 1: 200K training pairs for each subset							
RAL-Net	**1.46**	**2.63**	**0.51**	**0.91**	**1.95**	**1.40**	**1.48**
HardNet/ angular embedding	1.63	3.26	0.56	1.24	2.87	2.02	1.93
HardNet / robust form	1.67	3.27	0.57	1.06	2.51	2.15	1.87
HardNet	2.07	3.70	0.77	1.22	3.79	3.33	2.48
L2Net/contrastive	3.52	7.83	1.63	2.75	7.36	6.68	4.96

4.2 Descriptor generalization ability on Hpatches dataset

Recently, Hpatches, a new local descriptor evaluation benchmark, provides a huge dataset and an evaluation criterion for modern descriptors. This dataset consists of 65×65 pixel size of patches extracted from 116 sequences which originate from 6 images. Different from the widely used Brown dataset, Hpatches contains more diversity and noisy changes. The keypoints of this dataset are detected by DOG, Hessian, and Harris detectors from reference images which are then applied to reproject the three different geometric noisy image sequences of easy, hard and tough. A small fraction of the dataset is shown in Fig.6.

To comprehensively test the generalization abilities of descriptors, Hpatches also propose three different tasks, including patch verification, image matching, and patch retrieval. First, patch verification is used to verify whether two patches match or not by confidence scores. As for a patches pair set $P = \{(x_i, x_i'), y_i), i = 1, ..., N)\}$, consisting of positive pairs and negative pairs (x_i, x_i') with labels

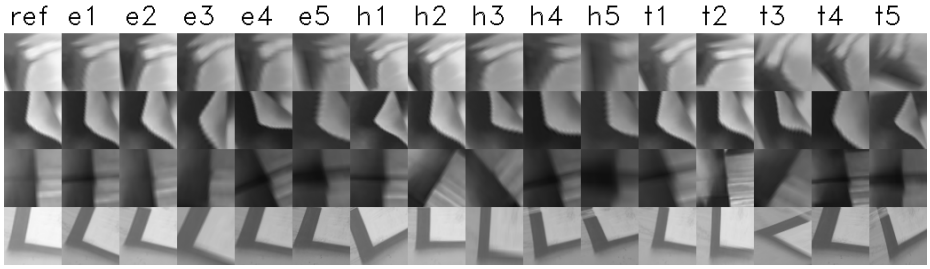


Fig. 6. Hpatches patches image. For each reference patch, there are 5 random geometric changing patches for three different changing range, which can be classified to e(easy), h(hard) and t(tough).

($y_i = 1, -1$), we calculate the average precision by the ranked confidence scores. The mean average precision (mAP) for all of the rank is finally used as the evaluation criterion. Second, image matching is similar to patch verification, in which we are given patches collection $L_k = (x_{k,i}, i = 1, ..., N)$, where L_r is from the reference image and L_t is from the target image. With respect to $x_{r,i}$ from L_r, we aim to find the maximum matching $x_{t,j}$ from L_t and get the related index $\{\sigma_i, i = 1, ..., N\}$ of finding $x_{t,j}$. After finding all of the matching, we consider if the found $x_{t,j}$ corresponds to $x_{r,i}$ with a ground truth label, and we get the mathching set $M = \{y_i = 2[\sigma_i \stackrel{?}{=} i] - 1\}$ by whether the found patch is matched with the label. Similar to the first task, we calculate the mAP for AP of set M for all ranks. The final task is patch retrieval, which considers these retrieved patches from the the matched images of reference images with a large proportion of distraction, and returns AP of the collection of labels ranked by confidence scores. For more protocol details, please refer to [28]. The result is shown in Fig.7.

Hpatches evaluation protocol considers many different aspects of patches from different view points and different illumination as well as patches of intra-class and inter-class, which are implemented from three degree of difficult sequences separately. In order to make a clearer demonstration, we just give the average performance of all different factors. Actually, these descriptors which obtain better average result shown in Fig.7 also perform better on these different child factors respectively. In terms of the result, RAL-Net generates the best results on the image matching task and the patch retrieval task, and a little behind HardNet in patch verification task. We can also observe that for the hardest image matching task, our descriptor with data augmentation shows an obvious improvement over the competitors. Overall, our descriptor gives almost the same results as L2Net and HardNet. This might be due to the different distributions between the Hpatches data and the Brown data, which needs further consideration and new learning algorithms such as transfer learning.

(a) Patch Verification

(b) Patch Matching

(c) Patch Retrieval

Fig. 7. Descriptors performance on three tasks

4.3 WxBS testing

In order to test the performance of our RAL-Net descriptor in a hard environment with various changing factors, we apply our descriptor on Wide Baseline Stereo [29]. The dataset consists of three different tasks, which are Appearance (A) by environment change, Geometry (G) with different view point, scale variance, etc., Illumination (L) influenced by brightness or image intensity, and Sensor (S) consisting of different type of data. With local feature detected by maximally stable extremal regions (MSER), Hessian-Affine and FOCI, each local patch is matched perfectly with the reference image. In addition, the evaluation metric is the same as the image matching task as Hpatches. The image example is shown as Fig.8 and the result is shown in Fig.9.

RAL-Net performs the best on average and shows a distinguished performance on Appearance and Illumination distraction matching task. For the Geometry task, all of the task performs almost at the same level. It is also worth noticing that SIFT and RootSIFT do not fall behind these learning-based descriptors and even perform better on the task Geometry due to their scale-invariant characteristics. But,as we can observe in Fig.8, all of the descriptor reach a quite bad performance in S and Map2ph task due to this kind of data not exiting in Brown dataset.

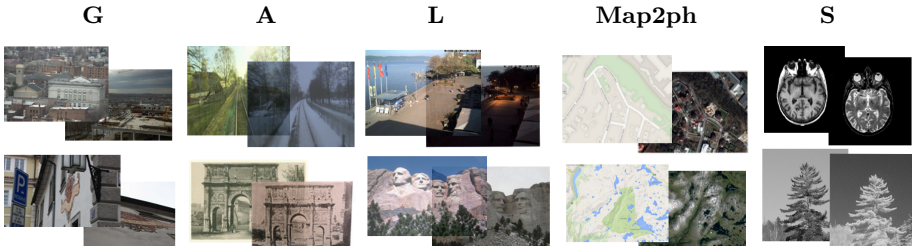


Fig. 8. Examples of WxBS dataset

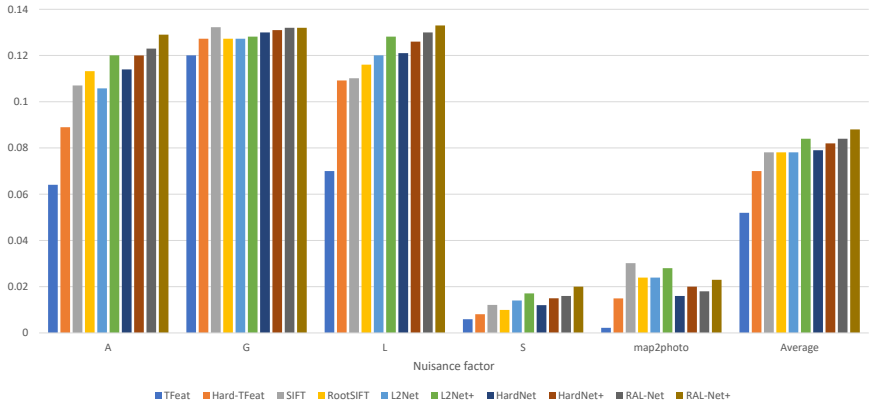


Fig. 9. Descriptors performance on three tasks

5 Conclusions

In this paper, we suggest a robust angular training loss named RAL-Net for deep embedding learning without sensitive parameters to further improve the performance of local descriptor learning,where the similarity between descriptors is defined as cosine distance, it is based on the idea of smooth the margin of triplet by giving different importance to triplet items with regard to the difference between the similarity of positive pairs and the similarity between chosen negative pairs. The loss can learn more information from limited data and performs better if with larger training data and relax the effect that false labels exits in training dataset. We test DigNet on typical Brown dataset, Hpathches dataset and W1BS dataset for diverse tasks verification and our RAL-Net have shown a superiority over existing local descriptors.

References

1. Tian, Y., Fan, B., Wu, F.: L2-net: Deep learning of discriminative patch descriptor in euclidean space. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017) 6128–6136

2. Mishchuk, A., Mishkin, D., Radenovic, F., Matas, J.: Working hard to know your neighbor's margins: Local descriptor learning loss. In Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., eds.: Advances in Neural Information Processing Systems 30. Curran Associates, Inc. (2017) 4829–4840
3. Choy, C.B., Gwak, J., Savarese, S., Chandraker, M.: Universal correspondence network. In Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R., eds.: Advances in Neural Information Processing Systems 29. Curran Associates, Inc. (2016) 2414–2422
4. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. (2007) 1–8
5. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision. Volume 2. (1999) 1150–1157 vol.2
6. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110
7. Fischer, P., Dosovitskiy, A., Brox, T.: Descriptor Matching with Convolutional Neural Networks: a Comparison to SIFT. ArXiv e-prints (2014)
8. Simo-Serra, E., Trulls, E., Ferraz, L., Kokkinos, I., Fua, P., Moreno-Noguer, F.: Discriminative learning of deep convolutional feature point descriptors. In: 2015 IEEE International Conference on Computer Vision (ICCV). (2015) 118–126
9. Han, X., Leung, T., Jia, Y., Sukthankar, R., Berg, A.C.: Matchnet: Unifying feature and metric learning for patch-based matching. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2015) 3279–3286
10. Zagoruyko, S., Komodakis, N.: Learning to Compare Image Patches via Convolutional Neural Networks. ArXiv e-prints (2015)
11. Simonyan, K., Vedaldi, A., Zisserman, A.: Learning local feature descriptors using convex optimisation. IEEE Transactions on Pattern Analysis and Machine Intelligence **36** (2014) 1573–1585
12. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Commun. ACM **18** (1975) 509–517
13. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2015) 815–823
14. Brown, M., Hua, G., Winder, S.: Discriminative learning of local image descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence **33** (2011) 43–57
15. Tola, E., Lepetit, V., Fua, P.: A fast local descriptor for dense matching. (2008)
16. Ke, Y., Sukthankar, R.: Pca-sift: A more distinctive representation for local image descriptors. (2004) 506–513
17. Balntas, V., Tang, L., Mikolajczyk, K.: Bold - binary online learned descriptor for efficient image matching. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2015) 2367–2375
18. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., eds.: Advances in Neural Information Processing Systems 27. Curran Associates, Inc. (2014) 487–495
19. Sun, Y., Wang, X., Tang, X.: Deep learning face representation from predicting 10,000 classes. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. (2014) 1891–1898

20. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). Volume 2. (2006) 1735–1742
21. Varior, R.R., Haloi, M., Wang, G.: Gated siamese convolutional neural network architecture for human re-identification. CoRR **abs/1607.08378** (2016)
22. Lin, J., Morère, O., Chandrasekhar, V., Veillard, A., Goh, H.: Deephash: Getting regularization, depth and fine-tuning right. CoRR **abs/1501.04711** (2015)
23. Chechik, G., Sharma, V., Shalit, U., Bengio, S.: Large scale online learning of image similarity through ranking. J. Mach. Learn. Res. **11** (2010) 1109–1135
24. Ustinova, E., Lempitsky, V.: Learning deep embeddings with histogram loss. In Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R., eds.: Advances in Neural Information Processing Systems 29. Curran Associates, Inc. (2016) 4170–4178
25. Yu, Y., Yang, M., Xu, L., White, M., Schuurmans, D.: Relaxed clipping: A global training method for robust regression and classification. In: NIPS. (2010)
26. Wu, C., Manmatha, R., Smola, A.J., Krähenbühl, P.: Sampling matters in deep embedding learning. CoRR **abs/1706.07567** (2017)
27. Wang, H., Wang, Y., Zhou, Z., Ji, X., Li, Z., Gong, D., Zhou, J., Liu, W.: Cosface: Large margin cosine loss for deep face recognition. (2018)
28. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. CoRR **abs/1704.05939** (2017)
29. Mishkin, D., Matas, J., Perdoch, M., Lenc, K.: Wxbs: Wide baseline stereo generalizations. CoRR **abs/1504.06603** (2015)