Learning Local Feature Descriptors with Quadruplet Ranking Loss

Dalong Zhang, Lei Zhao(✉), Duanqing Xu, and Dongming Lu

College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China
{dalongz,cszhl,xdq,ldm}@zju.edu.cn

Abstract. In this work, we propose a novel deep convolutional neural network (CNN) with quadruplet ranking loss to learn local feature descriptors. The proposed model receives quadruplets of two corresponding patches and two non-corresponding patches, and then outputs features measured by L_2 norm with dimension (256d) close to that of SIFT, which thus can be easily applied to practical vision tasks by plug-in replacing SIFT-like features. Moreover, the proposed model mitigates the problem that the margin separating corresponding and non-corresponding pairs varies with samples caused by commonly used triplet loss, and improves the capacity of utilizing the limited training data. Experiments show that our model outperforms some state-of-the-art methods like TN-TG and PN-Net.

Keywords: Image feature · Patch matching · Feature extraction Convolutional neural network

1 Introduction

Feature representation for local image patches is the problem of mapping raw image patches to descriptors in certain feature spaces to discriminate the correspondence of patches, which is one of major research topics in various computer vision tasks, such as structure from motion [17], multi-view stereo reconstruction [18], image stitching and alignment [21], etc. Recently, the field has witnessed the trend that traditional hand-crafted image features [2,14], are evolving into current data-driven and learning-based image feature descriptors [3,20,22], especially CNN-based features [6,19,26].

In this work, we aim to exploit the deep CNNs [4,7,8,12] to learn good local feature descriptors which can be used as plug-in replacement for popular descriptors like SIFT [14]. To that end, the metric measurement and feature dimension are restricted: the feature dimension should be close to that of SIFT while the distance is measured by L_2 norm. Many efforts has been made to move in this direction, but Siamese based methods [6,19,26,27] may not make full use of the relation between patch pairs while commonly used triplet loss [1,13] may lead to the problem that the margin separating corresponding and non-corresponding

J. Yang et al. (Eds.): CCCV 2017, Part I, CCIS 771, pp. 206–217, 2017.
https://doi.org/10.1007/978-981-10-7299-4_17

pairs varies with samples [23]. Learning a robust and discriminative descriptor that is competent for various patch-based computer vision tasks is still an open problem.

Inspired by the success of recent works [1,13,19,26], we propose a novel quadruplet CNN model to learn local feature descriptors. First, we construct our network with four branches sharing tied weights, receiving two corresponding patches and two non-corresponding patches simultaneously. Then, we minimize a quadruplet ranking loss to ensure that the distance between corresponding patches should be smaller than that between non-corresponding patches. At last, our model outputs a 256d feature descriptor measured by L_2 norm which can be easily used in a plug-in way to replace traditional descriptors. The proposed model enforces an interval between corresponding and non-corresponding patch pairs, which is helpful to distinguish the correspondence of them. Also we abrogate the 'anchor' used in triplet loss and design a online sampler to ensure any combination of positive pairs and negative pairs could contribute gradients in optimization process, which, as a result, pushes non-corresponding pairs far away from corresponding pairs without considering whether they refer to the same patch.

Overall, our main contributions are as follows: (1), we propose a novel quadruplet model to learn local feature descriptors and achieve better embedding results compared with some state-of-the-art methods; (2), we propose a way to improve the capacity of utilizing the limited data by online sampling and abrogating the 'anchor' when using quadruplet loss; (3), we apply our model to image matching task and it shows good generalization properties.

2 Related Work

Pretty a lot of works have been done to propose local feature descriptors and reached a practically applicable level of performance. Benefit from the growing availability of labeled data and increasing computational resources, data-driven methods for local descriptors have been a trend and already outperformed classic hand-crafted descriptors to some degree. Various techniques have been borrowed in carrying out such descriptors, including Linear Discriminant Analysis [3], boosting [22], convex optimization [20], and deep learning [10]. In this section, we will explore works based on deep CNNs, mainly organized by their architectures.

Siamese is one of the most frequently used architectures in learning local feature descriptors. Networks with Siamese architecture often has two branches sharing tied weights guided by pairwise loss. Works like [6,19,26] exploit this architecture to learn local feature representation. Simo-Serra et al. [19] trained a network outputting features measured by L_2 norm while [6,26] added a decision network to learn descriptors and metric simultaneously. Zagoruyko and Komodakis [26] also tried several variants of Siamese architecture like 2-channel model and pesudo-siam model and so on. Works with such architecture achieve significant improvement compared with their predecessor works [20,22]. But

some works like [19] mainly focus on classifying corresponding pairs and non-corresponding pairs without ensuring reasonable variation between them, which may lead to sub-optimal problem [1] when used for different tasks.

Triplet is born with the advantage of ensuring an interval between corresponding pairs and non-corresponding pairs, and once applied to this field, it becomes one of the most successful architectures. Works like [1,13] achieve the state-of-the-art performance on the Multi-view Stereo Correspondence dataset (MVS) [3], proving the effectiveness of this architecture. Kumar et al. [13] tried to combine the Siamese and Triplet architecture with global loss, while [1] trained a quite simple network but achieved epic performance and extraction efficiency. However, triplets of training set is formed based on the 'anchor', which may vary according to samples. Unfortunately, the margin separating corresponding pairs and non-corresponding pairs is depending on the position of 'anchor' and thus also varies with samples. The margin varying problem prevents the triplet architecture from enlarging variation between corresponding pairs and non-corresponding pairs and increases the risk of overfitting.

Inspired by these works, we propose a novel quadruplet model to learn feature descriptors. The model increases the capacity of utilizing the limited training data and also mitigates the margin varying problem. Experiments show that our method leads to significant improvements in performance.

3 The Proposed Approach

The overall methodology of the proposed approach is utilizing the power of deep convolution neural network to extract features from image patches to discriminate the correspondence of patches. But to get good features, we not only need a powerful network, but also a good loss function to guide the network. In this section, we will first explain our loss function and then detail the network structure.

3.1 Loss Function

Our loss is mainly inspired by triplet loss [1,13], which is successful exploited in learning feature descriptors. We will analysis the triplet loss first and then introduce our quadruplet ranking loss.

Triplet Loss. Triplet loss is based on triplet (a, p, n), where a is the 'anchor' (reference patch) while p and n represent positive and negative patch refer to the anchor respectively. Such loss enforces that the distance between positive pairs should be smaller than that between negative pairs:

$$D(f(a), f(p)) + I < D(f(a), f(n)) \tag{1}$$

where $f(\cdot)$ is the neural network mapping the raw image to certain feature space and I is used to ensure an interval between positive pairs and negative pairs to

Left: Triplet **Right:** Quadruplet

Fig. 1. Separating the positive and negative pairs with triplet and quadruplet loss. **Left:** Triplet loss enforces an interval between positive and negative pairs, but the margin varies with samples; **Right:** With quadruplet loss, the interval is ensured between any positive and negative pairs, thus even the minimum distance between them should be larger than the given interval, which works like support vectors in SVM [16].

separate them. $D(\cdot, \cdot)$ means a distance metric, and is often set to L_2 distance in this field. To satisfy the Eq. 1, triplet loss is formed as:

$$L(a, p, n) = max(0, I + \|f(a) - f(p)\|_2 - \|f(a) - f(n)\|_2) \tag{2}$$

The loss is efficient to push negative pairs away from positive pairs as the optimization process won't stop until the distance I between positive and negative pairs is confirmed. The interval I increases the average distance between positive and negative 'class' and thus makes them more different from each other, which is helpful to discriminate them.

However, according to Eqs. 1 and 2, the loss will confirm a margin between positive and negative pairs with reference to the 'anchor', which means the margin is depending on the position of 'anchor'. During optimization process, although the training samples can be content to Eq. 1 and the interval between positive and negative pairs can be confirmed, the position of the interval (separating hyperplane or say margin) may be distorted with different samples (shown in Fig. 1). The varying margin increases the risk that the distance between some positive pairs is larger than that between negative pairs, which will damage the performance when discriminating the correspondence of patch pairs. Moreover, the margin varying problem prevents the triplet loss from enlarging variation between corresponding pairs and non-corresponding pairs, and thus becomes a bottleneck to improve performance. What's worse, if the margin is highly nonlinear, it may increase the risk of overfitting. Hard negative mining can be helpful, but it's unclear to define the hard negative samples and it's quite time-consuming to find the proper hard negative samples.

Quadruplet Ranking Loss. To mitigate the problem mentioned above, we design a quadruplet ranking loss enforcing that the distance between arbitrary positive pairs should be smaller than that between negative pairs with given interval I:

$$\begin{aligned} &\|f(p_1) - f(p_2)\|_2 + I < \|f(n_1) - f(n_2)\|_2 \\ &\text{s.t.} \quad \forall(p_1, p_2) \in P, \quad \forall(n_1, n_2) \in N \end{aligned} \tag{3}$$

where (p_1, p_2) and (n_1, n_2) represent the positive pair and negative pair and they together form a quadruplet (p_1, p_2, n_1, n_2). P and N mean the whole set of positive pairs and negative pairs. The constraint Eq. 3 pushes any negative pairs away from positive pairs and implies that even the minimum distance between them should be larger than the given interval I:

$$\max_{(p_1,p_2)\in P} \|f(p_1) - f(p_2)\|_2 + I < \min_{(n_1,n_2)\in N} \|f(n_1) - f(n_2)\|_2 \tag{4}$$

which is illustrated in Fig. 1. According to Eqs. 3 and 4, we design a quadruplet ranking loss:

$$L(p_1, p_2, n_1, n_2) = \max(0, I + \|f(p_1) - f(p_2)\|_2 - \|f(n_1) - f(n_2)\|_2) \tag{5}$$

We use SGD to minimize the loss Eq. 5, and the gradient can be derived as:

$$\begin{aligned} \frac{\partial L}{\partial f(p_1)} &= 2(f(p_1) - f(p_2)) \\ \frac{\partial L}{\partial f(p_2)} &= -2(f(p_1) - f(p_2)) \\ \frac{\partial L}{\partial f(n_1)} &= -2(f(n_1) - f(n_2)) \\ \frac{\partial L}{\partial f(n_2)} &= 2(f(n_1) - f(n_2)) \end{aligned} \tag{6}$$

The proposed quadruplet loss holds some appealing properties: (1), the margin now depends on the position of $\max_{(p_1,p_2)\in P} \|f(p_1) - f(p_2)\|_2$ rather than the 'anchor', thus the margin varying problem is mitigated to some degree (shown in Fig. 1). (2), the loss is flexible to handle any combinations of positive and negative pairs as we abrogate the 'anchor' when forming quadruplets. The loss is degraded to triplet loss if $p_1 = n_1$; otherwise, it represents the added constraint on distance between positive and negative pair with different 'anchor'. As a result, we can make full use of any combinations of positive and negative pairs to optimize the target function, which improves the capacity to utilize the limited training data. We also propose a online sampling method to provide sufficient quadruplets, which will introduce in the next section.

3.2 Network Structure

We build a quadruplet deep convolution network subject to Eq. 5 to learn local feature descriptors.

The overall network consists of four branches with tied weights as shown in Fig. 2. In consistent with recently proposed methods, our network receives quadruplets of gray-scale patches with size 64×64 as input (one patch each

for the four branches). As the quadruplets is constructed with equal number of positive and negative patch pairs, we don't need to balance them to prevent the model from swing to one side during training. Moreover, each branch outputs a 256d vector measured by L_2 norm as learned feature descriptor, which is similar to the traditional local feature descriptors (such as SIFT). It's worth noting that, during inference, we just forward one branch to get the features as all these branches share the same weights.

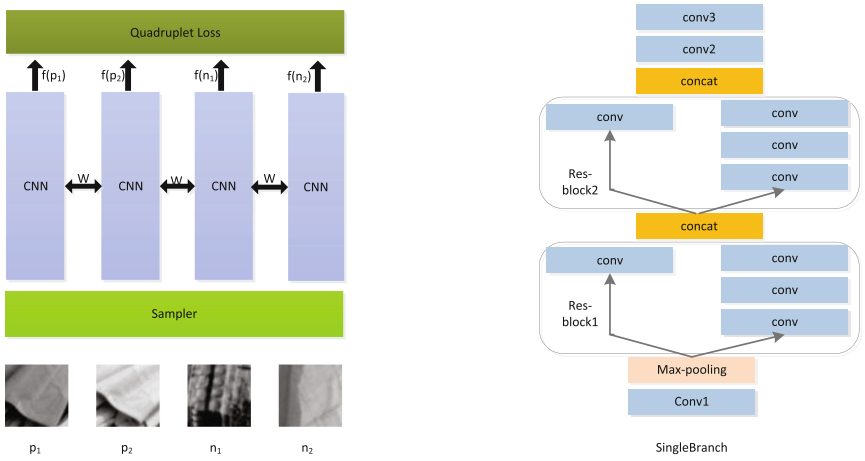


Fig. 2. Network Structure. The **left** shows the whole network with four branches sharing the same weights. The **right** part shows the structure of a single branch and the detail of layer parameters is reported in Table 1.

The single branch is designed referring to ResNet [7,8]. Each branch consists of two residual blocks along with other layers. We detail the structure of the single branch in Fig. 2 (the right part) and the parameters for each layer are listed in Table 1. Note that every convolution layer is followed with Batch Normalization layer [9] and ReLU activation [5] except for the last one, which are not shown in Fig. 2 and Table 1.

We also design a online sampler layer to enlarge training set. The sampler works as follows: (1), receive offline quadruplets $(p_{1,i}, p_{2,i}, n_{1,i}, n_{2,i})$ of training samples, where i implies the i-th quadruplet in current batch. (2), randomly select positive pair $(p_{1,m}, p_{2,m})$ and negative pair $(n_{1,k}, n_{2,k})$ from offline quadruplets to form a new quadruplet $(p_{1,m}, p_{2,m}, n_{1,k}, n_{2,k})$. (3), repeat the second step for certain times and add these new quadruplets to original quadruplets batch, then feed the expanded batch to the network. During training, we use the online sampler to double the offline quadruplets. The online sampler layer not only helps enlarge the training set, but also provides sufficient combinations of positive and negative pairs, which is helpful to discriminate the correspondence of patch pairs . Moreover, the sampling process is done within a batch, thus no extra patches from outside is needed.

We implement the work with caffe [11] and use SGD with learning rate 0.01 decreasing by a factor of 2 every 100,000 iterations, momentum 0.9 and weight decay 0.0001 to train our model. Margin I in Eq. 5 is set to 0.8 as $f(\cdot)$ is normalized. With NVIDIA GTX980Ti, it often takes about 2 hrs per epoch for training with $2M$ offline quadruplets (doubled by online sampling).

Table 1. The parameters for a single branch. Note that in this table Res-Block1 and Res-Block2 mean the two residual blocks in the single branch. The quadruple (f, s, p, o) means the parameters of the corresponding layer, mainly for convolution layer and pooling layer, representing filter size, stride size, padding size, and the num of the output feature maps, respectively. For pooling layer, there's no need to set the last parameter 'o', and we use '$-$' instead, like $(2, 2, 0, -)$.

Layer name	Output dimension	Detail	
Conv1	$96 \times 58 \times 58$	$(7, 1, 0, 96)$	
Max-Pooling	$96 \times 29 \times 29$	$(2, 2, 0, -)$	
Res-Block1	$192 \times 12 \times 12$	$(5, 1, 0, 192)$ $(3, 2, 0, 192)$ $(1, 1, 0, 192)$	$(1, 3, 0, 192)$
Res-Block2	$256 \times 6 \times 6$	$(5, 1, 0, 256)$ $(3, 1, 0, 256)$ $(1, 1, 0, 256)$	$(1, 2, 0, 256)$
Conv2	$128 \times 4 \times 4$	$(3, 1, 0, 128)$	
Conv3	$256 \times 1 \times 1$	$(4, 1, 0, 256)$	

4 Experiments

In this section, we will first introduce the dataset used for assessing our model, and then present the result with standard evaluation protocol. Also, we apply our model to image matching task.

4.1 Dataset and Evaluation Protocol

For training, we use the Multi-view Stereo Correspondence dataset (MVS) [3], which is collected by Winder et al. for learning descriptors. The dataset consists of three subsets, Liberty (LY), Notre Dame (ND), and Yosemite (YO), which were sampled from 3D recontructions of the statue of Liberty (New York), Notre Dame (Paris) and Half Dome (Yosemite). Patches were extracted around each interest point that generated by Difference of Gaussian detector (DoG) or Harris interest points detector. Each of the three subsets contains more than 450k grayscale image patches (64×64 pixels) with pre-generated label to form patch

pairs, all with equal number of match and dismatch patch pairs. In our work, we use the DoG set and make data augmentation by rotating the pair of patches by 90, 180, and 270 degrees, and flipping the images horizontally and vertically.

To evaluate the performance of our model, we follow the evaluation protocol of [3]. We train the model on one subset and test on the other two subsets. And then We report the false positive rate at 95% recall (FPR95) on each of the six combinations of training and testing sets, as well as the mean across all combinations. Following [13,20,26], our test is done on the 100,000 patch-pair Liberty, Notre Dame, and Yosemite sets.

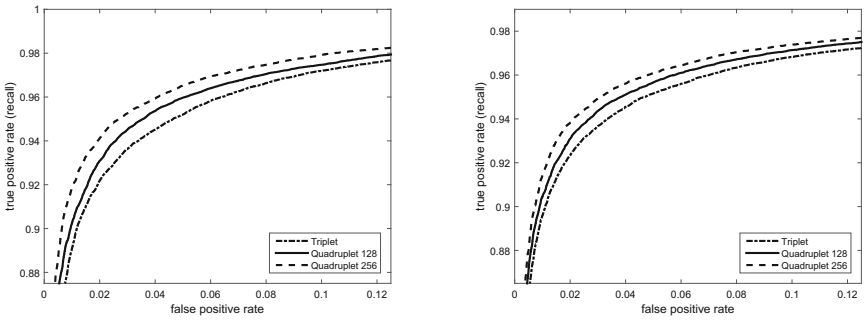
4.2 Results

Following the evaluation protocol mentioned above, we evaluate our model and some recently proposed methods with normalized SIFT as baseline. Meanwhile, we indicate the memory footprint for each of the descriptors by specifying their dimension, as summarized in Table 2. Moreover, we draw ROC curves to show the result of triplet and quadruplet with same network structure.

Table 2. The MVS matching result. Numbers are false positive rate at 95% recall, and the smaller the better. **Bold** numbers are the best results on the dataset.

Train	Dim.	ND	YO	LY	YO	LY	ND	Mean
Test		LY		ND		YO		
SIFT (no training) [14]	128d	29.84		22.53		27.29		26.55
Simonyan et al. (2014) [20] PR	<640d	16.56	17.23	9.88	9.49	11.89	11.11	12.69
Simonyan et al. (2014) [20] PR-proj	<80d	12.42	14.58	7.22	6.82	11.18	10.08	10.38
Zagoruyko et al. (2015) [26] siaml_2	192d	13.24	17.25	6.01	8.38	19.91	12.64	12.90
Zagoruyko et al. (2015) [26] 2streaml_2	256d	8.79	12.84	4.54	5.58	13.24	13.02	9.67
Simo-Serra et al. (2015) [19]	128d	8.50		4.59		14.59		9.22
Kumar et al. (2016) [13] TN-TG	256d	9.91	13.45	3.91	5.43	10.65	9.47	8.80
Balntas et al. (2016) [1] PN-Net	128d	8.27	9.76	3.81	4.45	9.55	7.74	7.26
Balntas et al. (2016) [1] PN-Net	256d	8.13	9.65	3.71	4.23	8.99	**7.21**	6.98
ours (Triplet $\approx$ 50 epoches)	128d	8.43	10.85	4.66	4.69	9.62	9.01	7.88
ours (Quadruplet $\approx$ 50 epoches)	128d	8.67	**9.58**	**3.61**	**3.87**	9.39	8.54	7.27
ours (Quadruplet)	256d	**6.87**	**8.63**	**2.77**	**3.16**	**8.30**	8.19	**6.32**

Our model show significant improvement in performance compared with the baseline and recently proposed works [1,13,19,20,26]. We reduce the average error rate from 26.5% (SIFT) to 6.32%, while win five out of all six combinations in terms of FPR95 compared with the state-of-the-art methods [1,13]. Our best model reduces the FPR95 by 2.48% and 0.66% compared with TN-TG [13] and PN-Net [1] separately. The comparison between our model and these previous works proves that it's powerful for our model to discriminate correspondence of patches.



Left: Trained on LY **Right:** Trained on YO

Fig. 3. ROC curves for model tested on ND subset. We draw the ROC curves of models with triplet (128d) and quadruplet (128d,256d). **Left:** Model trained on LY subset and tested on ND subset; **Right:** Model trained on YO subset and tested on ND subset.

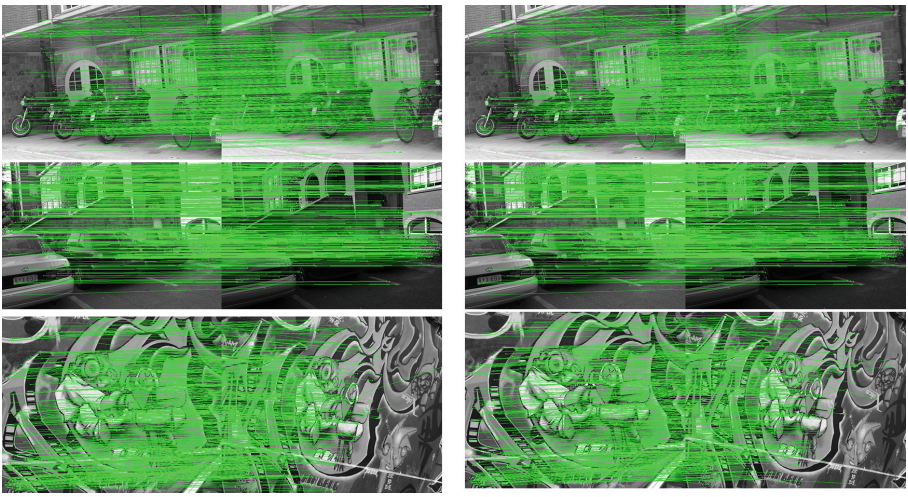
We also train the model with triplet and quadruplet loss with same single branch and training epoches ($\approx$ 50 epoches), and results show that quadruplet reduce the average error rate by 0.6% compared with triplet. We draw ROC curves of models to show the performance of the triplet model and quadruplet model, which is illustrated in Fig. 3. The comparison is done firstly with same output feature dimension (128d), and then we add our best model with 256d feature into competition. We present the result that tested on ND subset for brief, and the ROC curves clearly show that the quadruplet model outperforms the triplet one. The comparison between the 128d and 256d model also shows that it's helpful to improve performance by enlarging the feature dimension with our model. Moreover, while [19,26] is based on siamese architecture and [1,13] utilize triplet architecture to learn feature descriptors, our quadruplet model achieve best result, which implies that it's potential to use the proposed quadruplet architecture to learn feature descriptors.

4.3 Application Examples

We also apply our model to practical image matching task. We extract the features from raw image as follows: (1), detect the interest points with DoG; (2), extract the patches according to the position, scale, and orientation of the interest points; (3), resize the patches to 64×64 and feed the patches to the network to get the final feature descriptors. These steps are very similar to that of the most frequently used descriptors — SIFT. It's also worth mentioning that, as our feature is measured by L_2 distance and the dimension (256d) is close to that of SIFT, many well studied techniques (KNN search et al.) can be retained, thus it's quite easy to apply our descriptor to practical vision tasks by plug-in replacing SIFT descriptors.

When compared with SIFT in image matching task, our model shows its advantages. Following [15], we compute the number of false match points

(according to the fixed overlap error 50%) on a group of images with blur, light and viewpoint changes. As illustrated in Fig. 4, our descriptors are robust to changes and reduce the false match points by about 40% compared with SIFT. Note that, the images used in matching task is totally different from our training set, which proves the good generalization properties of our model.



Left: Our Model **Right:** SIFT

Fig. 4. Image matching examples. The left is our result while right SIFT's. **Top Row:** 'bikes' with blur. False matching points 58 vs. 112. **Middle Row:** 'leuven' with light change. False matching points 40 vs. 72. **Bottom Row:** 'graf' with viewpoint change. False matching points 59 vs. 82.

5 Conclusions

In this work, we proposed a quadruplet deep CNN model to learn local feature descriptors from raw data. Together with quadruplet ranking loss, we also design a sampler layer to make full use of limited training data. Our model achieves best results compared with SIFT and some state-of-the-art approaches, and shows good generalization properties when applied to practical image matching tasks. Moreover, our descriptor is able to replace traditional descriptors like SIFT in a plug-in way, thus many well studied techniques (KNN search et al.) can be retained. The main drawback of our method is that the process of training and inferring is time-consuming.

In future, we will simplify the network to accelerate the speed of feature extracting [1]. Moreover, we will try integrating our method with keypoint detection methods [14,25] to build a unified end-to-end framework [24]. In this way, we can learn keypoint detection and feature representation at the same time, which is an interesting research direction to compete with current pipeline methods.

Acknowledgments. This work was financially supported by the Zhejiang Province Natural Science Foundation (Y16F020023).

References

1. Balntas, V., Johns, E., Tang, L., Mikolajczyk, K.: Pn-net: conjoined triple deep network for learning local image descriptors. arXiv preprint arXiv:1601.05030 (2016)
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). Comput. Vis. Image Underst. **110**(3), 346–359 (2008)
3. Brown, M., Hua, G., Winder, S.: Discriminative learning of local image descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **33**(1), 43–57 (2011)
4. Cun, Y.L., Boser, B., Denker, J.S., Howard, R.E., Habbard, W., Jackel, L.D., Henderson, D.: Handwritten digit recognition with a back-propagation network. In: Advances in Neural Information Processing Systems, pp. 396–404 (1990)
5. Dahl, G.E., Sainath, T.N., Hinton, G.E.: Improving deep neural networks for LVCSR using rectified linear units and dropout. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8609–8613. IEEE (2013)
6. Han, X., Leung, T., Jia, Y., Sukthankar, R., Berg, A.C.: Matchnet: unifying feature and metric learning for patch-based matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3279–3286 (2015)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
8. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 630–645. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_38
9. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456 (2015)
10. Jahrer, M., Grabner, M., Bischof, H.: Learned local descriptors for recognition and matching. In: Computer Vision Winter Workshop, vol. 2 (2008)
11. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J.: Caffe: convolutional architecture for fast feature embedding. In: Eprint Arxiv, pp. 675–678 (2014)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Neural Information Processing Systems, pp. 1097–1105 (2012)
13. Kumar, B., Carneiro, G., Reid, I., et al.: Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5385–5394 (2016)
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
15. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10), 1615 (2005)
16. Mohri, M., Rostamizadeh, A., Talwalkar, A.: Foundations of Machine Learning. MIT Press, Cambridge (2012)
17. Molton, N., Davison, A.J., Reid, I.: Locally planar patch features for real-time structure from motion. In: BMVC, pp. 1–10 (2004)

18. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: 2006 IEEE Computer Society Conference on Computer vision and pattern recognition, vol. 1, pp. 519–528. IEEE (2006)
19. Simo-Serra, E., Trulls, E., Ferraz, L., Kokkinos, I., Fua, P., Moreno-Noguer, F.: Discriminative learning of deep convolutional feature point descriptors. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 118–126 (2015)
20. Simonyan, K., Vedaldi, A., Zisserman, A.: Learning local feature descriptors using convex optimisation. IEEE Trans. Pattern Anal. Mach. Intell. **36**(8), 1573–1585 (2014)
21. Szeliski, R.: Image alignment and stitching: a tutorial. Found. Trends® Comput. Graph. Vis. **2**(1), 1–104 (2006)
22. Trzcinski, T., Christoudias, M., Fua, P., Lepetit, V.: Boosting binary keypoint descriptors. In: Computer Vision and Pattern Recognition, pp. 2874–2881 (2013)
23. Ustinova, E., Lempitsky, V.: Learning deep embeddings with histogram loss. In: Neural Information Processing Systems, pp. 4170–4178 (2016)
24. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: LIFT: learned invariant feature transform. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 467–483. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_28
25. Yi, K.M., Verdie, Y., Fua, P., Lepetit, V.: Learning to assign orientations to feature points. In: Computer Vision and Pattern Recognition, pp. 107–116 (2016)
26. Zagoruyko, S., Komodakis, N.: Learning to compare image patches via convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4353–4361 (2015)
27. Zbontar, J., LeCun, Y.: Computing the stereo matching cost with a convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1592–1599 (2015)