2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

A Comparison of CNN-Based and Hand-Crafted Keypoint Descriptors

Zhuang Dai, Xinghong Huang, Weinan Chen, Li He, and Hong Zhang

Abstract— **Keypoint matching is an important operation in computer vision and its applications such as visual simultaneous localization and mapping (SLAM) in robotics. This matching operation heavily depends on the descriptors of the keypoints, and it must be performed reliably when images undergo condition changes such as those in illumination and viewpoint. Previous research in keypoint description has pursued three classes of descriptors: hand-crafted, those from trained convolutional neural networks (CNN), and those from pre-trained CNNs. This paper provides a comparative study of the three classes of keypoint descriptors, in terms of their ability to handle conditional changes. The study is conducted on the latest benchmark datasets in computer vision with challenging conditional changes. Our study finds that (a) in general CNN-based descriptors outperform hand-crafted descriptors, (b) the trained CNN descriptors perform better than pre-trained CNN descriptors with respect to viewpoint changes, and (c) pre-trained CNN descriptors perform better than trained CNN descriptors with respect to illumination changes. These findings can serve as a basis for selecting appropriate keypoint descriptors for various applications.**

I. INTRODUCTION

Keypoint matching is an important step in many computer vision and robotics applications, such as structure-from-motion (SfM) [1], multi-view stereo (MVS) [2], image retrieval [3] and visual simultaneous localization and mapping (SLAM) [4]. The performance of these applications strongly depends on keypoint matching, which in turn depends on the quality of keypoint detectors and descriptors. This paper is concerned with the evaluation of keypoint descriptors in robotics applications. In general, the most important consideration in assessing the quality of a keypoint descriptor is its invariance properties with respect to viewing conditions. We are particularly interested in examining hand-crafted descriptors and learned descriptors with convolutional neural networks (CNN), in terms of their invariance with respect to illumination and viewpoint, two viewing conditions of dominant consideration in robotics applications.

The importance of keypoint description motivated its extensive research that has resulted in the development of many description techniques. At the same time, this also created the need to compare the performance of the developed keypoint descriptors. Early research in keypoint descriptors focused on hand-crafted solutions such as SIFT [5], SURF [6] and ORB [7], which still play an important role today. The seminal work by Mikolajczyk et al. [8] provided arguably the first and earliest comprehensive evaluation of hand-crafted descriptors, and their study included GLOH, SIFT, PCA-SIFT and Spin images, on a small dataset of only 48 images. More recently, Juan et al. [9] compared SURF with other hand-crafted descriptors, and drew the conclusion that SIFT performs the best in terms of stability, although SURF is the fastest. Another comparative study on SIFT and its variants [10], such as GSIFT, CSIFT, PCA-SIFT, and SURF, showed that each descriptor had its own advantage. In addition to invariance properties, the time complexity of keypoint matching is also important and was the focus of the comparative study in [11] that evaluated SIFT and SURF with several most recent binary descriptors including BRIEF, ORB and BRISK. Although SIFT performs the best in terms of matching accuracy, it is known to be computationally costly.

The recent rise of deep learning has created the opportunity to develop learning-based, data-driven techniques of keypoint description. In particular, the impressive success of CNN in tackling a variety of image classification problems motivated people to pursue learned keypoint descriptors that use either specially a trained CNN or the various layers of a pre-trained CNN. Among trained CNNs for keypoint description, the most notable models include DeepDesc [12], L2-Net [13], CS L2-Net [13] and HardNet [14], and they generate descriptors with a length of either 128 or 256 dimensions, similar to that of the hand-crafted descriptors. In parallel, a number of popular pre-trained CNN models exist, such as AlexNet [15], VGG16 [16], ResNet101 [17], DenseNet169 [18], and their various layers are candidates as keypoint descriptors as well. Any layer of these pre-trained CNNs would have a length that is typically two or three orders of magnitude larger than that of hand-crafted descriptors, and this length directly affects their time complexity for matching keypoints.

All studies on trained CNNs for keypoint description inevitably evaluated their performance against hand-crafted descriptors, with the common conclusion that they outperform the hand-crafted descriptors in terms of their invariance properties. Using a pre-trained CNN, Fischer et al. [19] conducted a performance evaluation of keypoint descriptors with one CNN model, the AlexNet [15], and the hand-crafted SIFT, on two small datasets, and drew the conclusion that the CNN-based descriptors outperform SIFT. Interestingly, there has not been a comprehensive comparison between pre-trained CNNs and hand-crafted descriptors, or between pre-trained CNNs and trained CNNs. This is the main motivation

All authors are with the Biomimetic and Intelligent Robotics Lab (BIRL), Guangdong University of Technology, Guangzhou, China, 510006. H. Zhang is the corresponding author (hzhang@ualberta.ca) and is also with the Department of Computing Science, University of Alberta, AB, Canada.

The work in this paper is supported by the Natural Science Foundation of China (Grant No. 61673125, 61703115), and the Frontier and Key Technology Innovation Funds of Guangdong Province (Grant No. 2016B090910003).

978-1-5386-6026-3/19/$31.00 ©2019 IEEE

behind our comparative study described in this paper.

Another limitation of previous comparative studies of keypoint descriptors is the limited size of the datasets used. In order to overcome the problem of insufficient data, a new benchmark called Hpatches [20] was recently constructed. Hpatches contains a large number of multi-image sequences of different scenes under real and varying viewing conditions, and provides homography between image pairs as the ground truth. [20] subsequently evaluated several representative hand-crafted and trained CNN descriptors in the application of patch classification, image matching and patch retrieval.

Exploiting the new Hpatches dataset, this paper provides a comparative study of three classes of keypoint descriptors: hand-crafted (two representative methods), those from trained CNN (four representative models), and those from pre-trained CNNs (four representative models), in terms of their ability to handle viewing condition changes and runtime. The main contribution of our work is the comprehensive consideration of pre-trained CNNs in the comparative study, with both hand-crafted and trained CNN. We focus on two viewing condition changes, illumination and viewpoint, arguably the two most important changes to study in robotics. The most interesting finding of our study is that pre-trained CNN outperforms trained CNN in handling illumination change. Table I summarizes our study with respect to the previous studies in keypoint description evaluation.

TABLE I: Existing comparative studies, in terms of the size of the dataset used, where [8] includes 48 images with 8 sequences, [19] includes 416 images with 16 sequences and [14], [20] and ours include 696 images with 116 sequences, and the classes of keypoint descriptors involved, where HC refers to hand-crafted, T-CNN to trained CNN, and PT-CNN to pre-trained CNN.

Ref/Year	Dataset	Descriptor Classes
[8]/2004	small	Hand-Crafted (HC)
[19]/2014	medium	HC vs. Pre-Trained-CNN
[20]/2017	large	HC vs. Trained-CNN
[14]/2018	large	HC vs. Trained-CNN
Ours/2019	large	HC vs. T-CNN vs. PT-CNN

The remainder of this paper is organized as follows. In Section II, we will describe the details of our experimental methods in conducting the comparative study, including the dataset and performance measure. We present the experimental results of the study in Section III. In Section IV, we conclude the study, and outline the future work.

II. EXPERIMENTAL METHODS

In our comparative study, we evaluate different keypoint descriptors in terms of their performance with respect to illumination and viewpoint changes. Illumination and viewpoint changes are challenging problems in keypoint matching, especially in robotics applications, and we leave other types of changes such as blur and compression as future work. We consider three classes of keypoint descriptors: hand-crafted, those from trained CNNs, and those from pre-trained CNNs, as are shown in Fig. 1. The same keypoint detector based on difference of Gaussians (DoG) is used for detecting the keypoints before the descriptors are computed. We choose DoG as the detector because the training dataset of trained CNNs uses it [21] and because of its popularity in practice. We use mean average precision (mAP) of matched keypoints as the performance metric. Details of the descriptor methods and the datasets used in the comparative study are provided in the sections below.

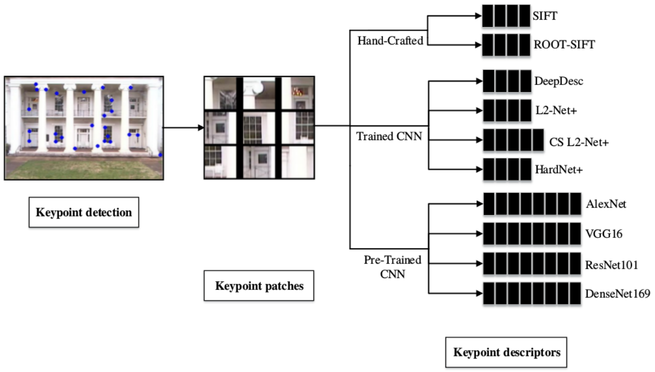


Fig. 1: The process of competing different keypoint descriptors in our experiments, to be compared in terms of mean average precision (mAP) of the matching result. Keypoint matching is performed via bidirectional nearest neighbor search. In the experiments, we detect 500 keypoints per image by difference of Gaussian (DoG) detector. For hand-crafted descriptors, we use SIFT and ROOT-SIFT as the representatives. We extract 64×64 image patches around the keypoints, for all descriptors, and then compute descriptors from these patches. For pre-trained CNN descriptors, we have to resize the 64×64 patches to 224×224 due to the input layer of the pre-trained CNN models.

A. Keypoint Descriptor Methods

We select SIFT provided in the VLFeat library and its variant ROOT-SIFT [22] as representatives of hand-crafted descriptors in our study. SIFT is found to be superior in terms of matching performance in many previous comparative studies [8], [9]. ROOT-SIFT is a simple algebraic extension of SIFT that measures descriptor distance in a different metric space with improved performance. For fair comparison, we use a 64×64 patch, rather than the default 16×16 patch, to generate a descriptor for both SIFT and ROOT-SIFT, as in [23].

From CNN-based descriptor methods, we select two classes: those computed by training a CNN, and those computed by various layers of a pre-trained CNN. From trained CNN models, we select DeepDesc [12], L2-Net [13], CS L2-Net [13] and HardNet [14]. These CNN description models have been trained on a large number of image patches, each generating a descriptor of either 128 or 256 dimensions. For pre-trained CNN models, we choose various layers of AlexNet [15], VGG16 [16], ResNet101 [17] and DenseNet169 [18], all of which have been trained on the ImageNet dataset for image classification applications. We use the parameters of the four pre-trained CNN models available in PyTorch. The specific layers used in the four pre-trained CNN models are shown in Table II, in order to identify layers that exhibit competitive performance as keypoint descriptors.

For each detected keypoint, we extract a 64 × 64 patch to compute all the competing descriptors. SIFT and ROOT-SIFT as well as trained CNN descriptor methods can readily work with this patch size. However, for the pre-trained CNN models, we must resize the image patch to 224 × 224 using OpenCV, a size that is expected by these models for them to generate the descriptors.

TABLE II: The different layers of popular CNN models we used to represent pre-trained CNN keypoint descriptors. We apply max-pooling to the extracted convolutional layers and denote the output of max-pooling as Pools obtained with pooling size of 3 and a step size of 1.

CNN Models	**Network Layers**
AlexNet	Conv2 to Conv5, Pool2 to Pool5
VGG16	Conv3_3 to Conv5_3, Pool3 to Pool5
ResNet101	Resdual Block1 to Block4, Pool1 to Pool4
DenseNet169	Dense Block1 to Block4, Pool1 to Pool4

B. Dataset Used in the Evaluation

For evaluation, we use the comprehensive Hpatches dataset [20], a collection of datasets from various sources including existing datasets designed to study keypoint descriptors. The Hpatches dataset includes a total of 57 sequences that involve mainly photometric changes and 59 other sequences that involve mainly viewpoint changes. In this paper, we refer to these two subsets as illumination dataset and viewpoint dataset, respectively. Each sequence includes a set of six images. The first image is used as the reference of the sequence and the others as target images. Some example sequences are shown in Fig. 2. Hpatches [20] provides the ground truth for keypoint matching in the form of homography matrices between the reference image and the target images.

With the homography matrices in Hpatches, the keypoint matches can be easily verified by following the method in [24]. Specifically, denote p_1 and p_2 as two matching keypoints of two images, and H as the given homography. Then $H \cdot p_1$ predicts its corresponding position on the other image. If the distance of $H \cdot p_1$ and p_2 is lower than a threshold, then p_1 and p_2 are matched correctly.

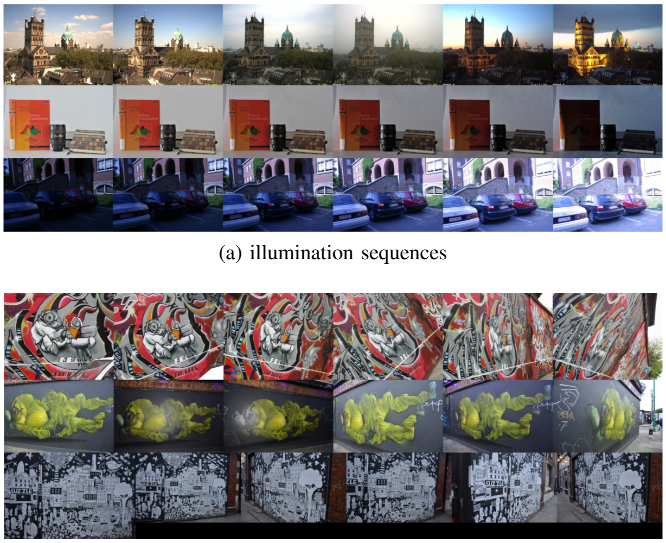
$$\|p_2, H \cdot p_1\| \leq threshold \tag{1}$$

In our study, we set the threshold to 3 pixels, which was found to provide the best results in our experiments.

C. Keypoints Matching and Performance Measure

Keypoints matching includes four main steps: keypoint detection, descriptor generation, and distance computation between descriptors, and nearest neighbor search. For keypoints detection, we detect 500 keypoints for one image by the DoG detector. For computing descriptor distance, we use L2 distance for descriptors with lower dimensions, and use cosine distance with higher dimensions. The reason for the difference in the choice of distance metric is that we found L2 to work well for low-dimensional descriptors and cosine distance to work better for high-dimensional descriptors. The final step of keypoint matching is performed by nearest neighbor search based on computed descriptor distances. We adopt the common bidirectional nearest neighbor criterion for a match to be accepted, i.e., for two descriptors to match, they must be mutually nearest neighbors of each other. The performance metric of keypoint matching we use is the mean average precision (mAP) which is the same as in Balntas et al. [20] designed.

In our experiments, we extract 500 keypoints patches around the 500 keypoints detected by the DoG detector. Some of the patches are discarded if they lie too close to an image border, for a 64 × 64 patch to exist.



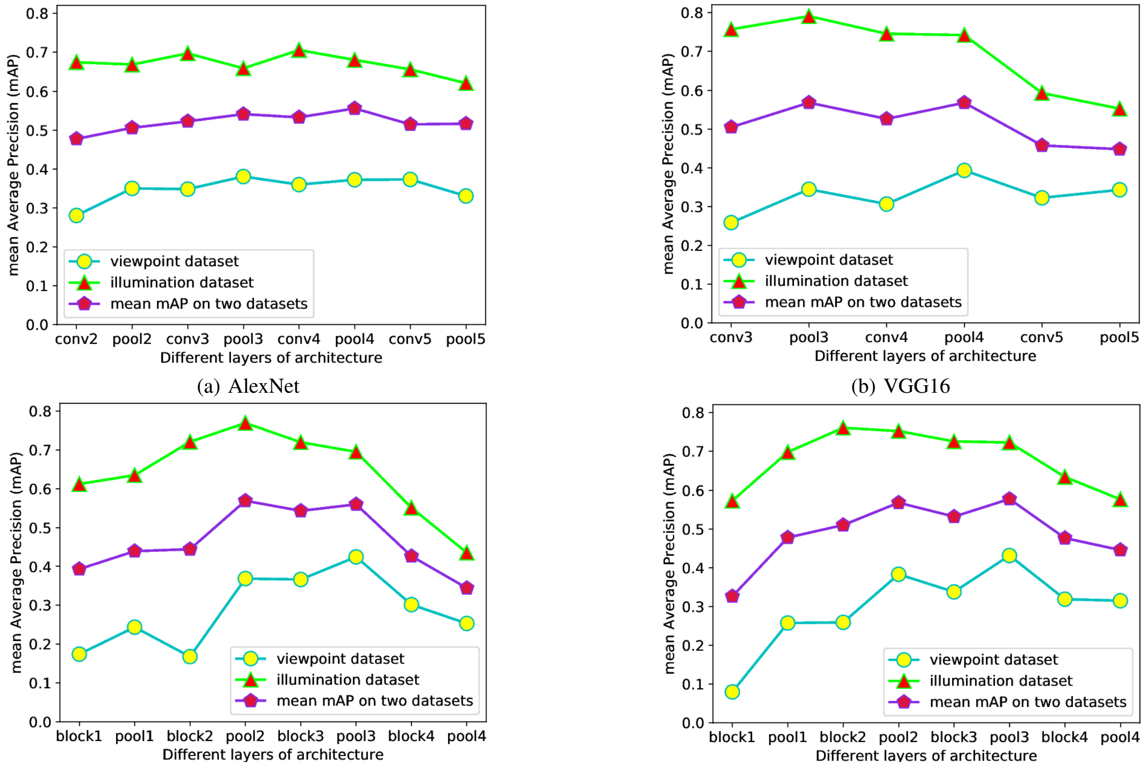
(a) illumination sequences

(b) viewpoint sequences

Fig. 2: Examples images in the Hpatches dataset of (a) illumination sequences and (b) viewpoint sequences where each row represents a sequence. The subsets of Hpatches we used include 57 illumination sequences and 59 viewpoint sequences. Each sequences includes six images. The first image is the reference and the others are target images with different illumination or viewpoint changes. For each sequence, we match the first image with other five, calculate the average precision (AP) for each pair of matching images, and then calculate the mean AP for the five pairs.

III. EXPERIMENTAL RESULTS

In this section, we describe the results of the comparative study on the performance of the three classes of keypoint



(a) AlexNet

(b) VGG16

(c) ResNet101

(d) DenseNet169

Fig. 3: Comparison the performance of different layers of four pre-trained CNN models in terms of the mean average precision (mAP) of matching keypoints on three subsets of the Hpatches dataset, called viewpoint dataset, illumination dataset, and combined dataset. (a) We extract the 2th to 5th convolutional layers of AlexNet and define them as conv2 to conv5. We then apply max-pooling on them. (b) We extract the 3rd, 4th and 5th convolutional layers of VGG16 and define them as conv3, conv4 and conv5, and perform max-pooling to obtain pool3 and pool5. (c) We use the 1st to the 4th residual block of ResNet101 and define them as block1 to block4, from where pool1 through pool4 are obtained with max-pooling. (d) We extract the 1st to the 4th dense block of DenseNet169 and define them as block1 to block4 from where pool1 through pool4 are obtained with max-pooling. The above figures show that, the pre-trained CNN descriptors perform better on illumination dataset than they do on viewpoint dataset. Using the average performance on two datasets as an overall performance. We select pool4 of AlexNet, pool4 of VGG16, pool3 of ResNet101 and pool3 of DenseNet169 as the representatives of pre-trained CNN descriptors.

descriptors on the Hpatches dataset.

The experimental comparison is run on a computer with 16 CPUs (2.1 GHz), 128 GB memory and 2 Nvidia TITAN Xp GPUs. We run hand-crafted descriptors and CNN-based descriptors on the CPU and GPU, respectively.

A. Selection of Pre-Trained CNN Descriptors

To manage the scope of the study, we first compare pre-trained CNN descriptors among themselves to identify the top performing representatives, which are then further compared with hand-crafted and trained CNN descriptors.

We use a total of 30 different layers from four different CNN models to generate descriptors in the selection of the best-performing pre-trained CNN descriptors. The performance of all the candidate pre-trained CNN descriptors is shown in Fig. 3, one figure for each CNN model. In each model, we plot the mAP of the descriptors on the illumination and viewpoint dataset, as well as the result on both datasets.

In general, all selected descriptors work reasonably well and, within each model, their performance does not fluctuate significantly. In addition, the illumination change is easier to handle than viewpoint change for all the models. Based on these mAP curves, on balance, from AlexNet we select Pool4, from VGG16 we select Pool4, from ResNet we select Pool3, and from DenseNet169 we select Pool3, as the four representative pre-trained CNN descriptors in the next phase of our comparative study.

B. Comparison of the Three Classes of Descriptors

Subsequently, the performance of 10 keypoint descriptors - four pre-trained CNN descriptors, two hand-crafted descriptors, and four trained CNN descriptors - is compared using the illumination and viewpoint datasets. Once again, we use mAP as the performance metric. The results of the comparison are shown in Table. III, which also includes the dimensions of the descriptors and the run time of performing keypoint matching for one pair of images.

TABLE III: Comparison the performance in terms of mean average precision (mAP) of the three classes of descriptors: hand-crafted descriptors, trained CNN descriptors, pre-trained CNN descriptors. For hand-crafted descriptors, we use SIFT and ROOT-SIFT. For trained CNN descriptors, DeepDesc is trained on Brown dataset [25] including three sub-datasets and the other three are trained on Liberty dataset which is a sub-dataset of Brown dataset. For pre-trained CNN-based descriptors, we extract the output of convolutional layers of CNN models trained on ImageNet dataset and then applying max-pooling to the extracted features. We find the trained CNN descriptors perform better than pre-trained CNN descriptors on viewpoint dataset, while pre-trained CNN descriptors perform better than trained CNN descriptors on illumination dataset. The hand-crafted descriptors underperforms substantially those from the other two classes of descriptors in all datasets.

descriptor classes	different descriptors	viewpoint dataset	illumination dataset	dimension	matching time (s)[1]
Hand-Crafted	SIFT [23]	32.08%	50.09%	128	1.8277
	ROOT-SIFT [22]	33.42%	54.14%	128	1.8362
Trained CNN	DeepDesc [12]	35.77%	49.06%	128	2.0395
	L2-Net+[2][13]	48.56%	65.15%	128	2.6299
	CS L2-Net+[2][13]	50.15%	66.97%	256	2.7655
	HardNet+[2][14]	**51.57%**	64.04%	128	1.7571
Pre-Trained CNN	AlexNet-pool4	43.11%	68.05%	30976	2.2288
	VGG16-pool4	39.35%	**74.22%**	346112	6.0847
	ResNet101-pool3	42.45%	69.52%	147456	4.5169
	DenseNet169-pool3	43.16%	72.30%	184320	4.7232

[1] The matching time includes generating CNN-based or hand-crafted descriptors for each keypoint of two matching images and computing their distance.

[2] the + means the model uses data argument.

A few interesting conclusions can now be readily drawn. First, hand-crafted descriptors are not competitive with respect to either class of CNN-based descriptors. Secondly, with respect to viewpoint changes, the trained CNN descriptors perform the best among all three classes of keypoint descriptors, with mAP that is 4.49% higher than pre-trained CNN descriptors and 13.26% higher than hand-crafted descriptors, on average. Thirdly, the pre-trained CNN descriptors perform the best with respect to illumination changes, with mAP that is 9.71% higher than the trained CNN descriptors and 18.9% higher than the hand-crafted descriptors, on average. One could speculate that because pre-trained models have been exposed to more varied lighting conditions in the training dataset than the trained models, they exhibit better illumination invariance, although this should be further analyzed. On balance, CS L2-Net+ among the trained CNN descriptors and DenseNet169-Pool3 among the pre-trained CNN descriptors are strong contenders for the overall winner.

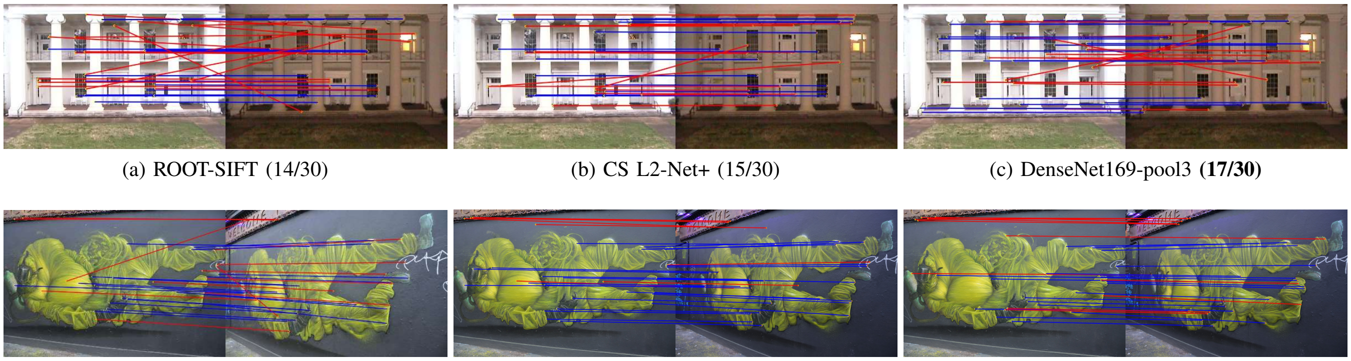
In Fig. 4 we show the qualitative results of keypoint matching using the top performing descriptors from each of three classes studied in this paper. For visibility, we only show, in each image pair, 30 of the matched keypoints with the highest similarity. In the example from the illumination dataset, DenseNet169-pool3 correctly matches 17 keypoints, CS L2-Net+ 15 keypoints, and ROOT-SIFT 14 keypoints. In the example from the viewpoint dataset, CS L2-Net+ correctly matches 23 keypoints, DenseNet169-pool3 19 keypoints, and ROOT-SIFT 18 keypoints.

As a final remark, although pre-trained CNN descriptors provide competitive performance with respect to the other two classes, especially with respect to illumination changes under which they outperform the best, they have the obvious weakness of being memory-intensive. Their descriptor length is two or three orders of magnitude higher than the other two descriptor classes, and the improved keypoint matching accuracy comes at the expenses of memory and computation, which though acceptable are important in most applications.

IV. CONCLUSION

In this paper, we have provided a comparative study among three classes of keypoint descriptors: hand-crafted, those from pre-trained CNNs, and those from specifically trained CNNs. This study allows us to draw the following conclusions. First, CNN-based descriptors in general outperform the hand-crafted descriptors in terms of their matching accuracy with respect to illumination and viewpoint changes. This conclusion is consistent with that in [19]. Second, the trained-CNN descriptors perform better than the pre-trained CNN descriptors with respect to viewpoint changes. Third, pre-trained CNN descriptors perform better than the trained CNN descriptors with respect to illumination changes. While our comparison between hand-crafted and trained CNN descriptors confirms the previous studies, it was conducted on a large, comprehensive dataset. In addition, the comparison between trained and pre-trained CNN descriptors has not been investigated in the literature in the past and represents our major contribution.

Our future work will focus on two issues. First, the matching complexity of pre-trained CNN is higher than the other two classes of descriptors. To exploit its excellent



(a) ROOT-SIFT (14/30)

(b) CS L2-Net+ (15/30)

(c) DenseNet169-pool3 **(17/30)**

(d) ROOT-SIFT (18/30)

(e) CS L2-Net+ **(23/30)**

(f) DenseNet169-pool3 (19/30)

Fig. 4: Qualitative comparison of three descriptors, one from each competing class, for keypoints matching. The upper pairs of images are illumination changed and the lower pairs of images are viewpoint changed. For the convenience of viewing, we only show the 30 pairs of matching points with the highest similarity for each image. The red lines connect incorrect matches and the blue lines connect the correct matches. DenseNet169-pool3 is the best among three descriptors detecting 17 correct matches with illumination changes, while the CS L2-Net+ is the best detecting 23 correct matches with viewpoint changes.

performance in dealing with illumination change without the additional computational cost, we will investigate dimensionality reduction techniques to the pre-trained CNN descriptors. Second, we will examine ways of combining trained and pre-trained CNN descriptors to obtain novel descriptors that are robust with respect to both viewpoint and illumination changes, as well as other classes of condition changes commonly experienced in computer vision and robotics.

References

[1] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4104–4113.

[2] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision*. Springer, 2016, pp. 501–518.

[3] L. Zheng, Y. Yang, and Q. Tian, "Sift meets cnn: A decade survey of instance retrieval," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 5, pp. 1224–1244, 2018.

[4] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part ii: Matching, robustness, optimization, and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vision & Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[7] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *International Conference on Computer Vision*, 2012, pp. 2564–2571.

[8] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[9] L. Juan and O. Gwun, "A comparison of sift, pca-sift and surf," *International Journal of Image Processing (IJIP)*, vol. 3, no. 4, pp. 143–152, 2009.

[10] J. Wu, Z. Cui, V. S. Sheng, P. Zhao, D. Su, and S. Gong, "A comparative study of sift and its variants," *Measurement science review*, vol. 13, no. 3, pp. 122–131, 2013.

[11] J. Heinly, E. Dunn, and J.-M. Frahm, "Comparative evaluation of binary features," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 759–773.

[12] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 118–126.

[13] Y. Tian, B. Fan, F. Wu *et al.*, "L2-net: Deep learning of discriminative patch descriptor in euclidean space." in *Cvpr*, vol. 1, 2017, p. 6.

[14] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," in *Advances in Neural Information Processing Systems*, 2017, pp. 4826–4837.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks." in *CVPR*, vol. 1, no. 2, 2017, p. 3.

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

[20] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 4, no. 5, 2017, p. 6.

[21] M. Brown, G. Hua, and S. Winder, "Discriminative learning of local image descriptors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 43–57, 2011.

[22] R. Arandjelovic and A. Zisserman, "Three things everyone should know to improve object retrieval," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2911–2918.

[23] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "Matchnet: Unifying feature and metric learning for patch-based matching," in *Computer Vision and Pattern Recognition*, 2015, pp. 3279–3286.

[24] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *European conference on computer vision*. Springer, 2002, pp. 128–142.

[25] S. A. Winder and M. Brown, "Learning local image descriptors," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.