Handcrafted Descriptors and Learned Descriptors

Chih-Hsiang Wang
Oregon State University
wangchih@oregonstate.edu

Jui-Hung Lu
Oregon State University
lujui@oregonstate.edu

Webster Cheng
Oregon State University
chengwe@oregonstate.edu

Abstract

We represent the comparison between handcrafted descriptors and learned descriptors. SIFT is selected to be our hand-crafted descriptor. ResNet, HardNet, and MatchNet are chosen to be our learned descriptors. We also built a new network based on ResNet and call it PockNet. The performance of PockNet is second to HardNet for the testing on Notredame-FPR95, but worse than HardNet and MatchNet during the one-to-one matching test. The final result shows that learned descriptors are slightly better under our assumption. However, we validate that learned descriptors are useful in image query by one-to-one matching.

1. Introduction

Image matching is an important part in computer vision. People want to find the corresponding images to apply on image retrieval [1], image alignment [2], texture classification [3], object recognition [4], motion tracking [5], 3D reconstruction [6], robot navigation [7], and much more. The basic principle for image matching is similar that it needs to get the keypoints descriptors of two images for matching.

Keypoints descriptors are the methods to compare keypoints that they summarize characteristics of keypoints in vector format. They are usually categorized as hand-crafted or learned descriptors. The hand-crafted descriptor is a traditional way of features extraction using algorithm predefined by experts. The hand-crafted features can be further divided into global and local that global features describe an image as a whole, while local descriptors describe significant patches around selected keypoints [3]. In contrast, learned descriptor relates to a way of extracting features by a neural network such as Convolutional Neural Network (CNN) [3]. People usually use a pre-trained neural network on a tremendous dataset and utilize the model as a feature extractor for a specific task.

In previous decades, hand-crafted descriptors are the only way to match images. The feature matching stage is crucial that finding the best local feature descriptors is the trend for improving the matching performance [8]. However, with the new invention and increasing ability of neural network and hardware devices, learning features directly from data gives people another way for image matching. There are two groups of people defending for hand-crafted descriptors or learned descriptors. Those who support hand-crafted descriptors demonstrate that SIFT and its variants (RootSIFT, DSP-SIFT) are clearly better on image matching, small-scale retrieval [9], and 3D-reconstruction [10]. In the opposite, the supporters of learned descriptors show that learned descriptors outperform hand-crafted descriptors. Some people indicated that the argument may derive from the comparison with more advanced hand-crafted descriptors and different datasets [10].

Based on the discussion of some papers, we try to compare the performances of hand-crafted and learned descriptors for image retrieval. The basic hand-crafted descriptors we use is SIFT, which is proved to be the most popular one for keypoints detection [11]. The RootSIFT as a more advanced hand-crafted descriptor is used for improving our models. The learned descriptors we applied are ResNet, HardNet, and MatchNet. Recently, ResNet is said to be good and popular on image matching [12] while HardNet was a relatively new proposed method [10]. MatchNet is not a new idea, but we want to use it as a comparison because it was proven useful for image matching [13]. In addition to using hand-crafted and learned descriptors separately, we also try combining them together to check the possibility of improvement. The testing dataset we use contains 35 query images and 140 matching images with ground truths that we can experiment which model performs better under our assumption.

2. Related work

Comparing descriptors is a noteworthy topic in deep learning. The following part is an overview of what people have done and researched to improve matching accuracy.

2.1. Benchmark and evaluation

Concerning about the ambiguities and inconsistencies in existing datasets and evaluation protocols, University of Oxford and Imperial College London introduced HPatches as a public benchmark for local descriptors [14]. Besides, they provided a new large dataset for training and testing modern descriptors because of the saturation in old datasets [14]. The evaluation protocols cover tasks including matching, retrieval, and classification [14]. Though we are not using this benchmark for our models, it is valuable to know the benchmark for future implementation and improvement.

2.2. Comparison between descriptors

Some evaluations of learned and advanced hand-crafted feature descriptors are proposed to find a better method. Based on the paper [8], learned descriptors have better matching ability than SIFT. However, in some conditions such as in Structure-from-Motion (SFM), advanced hand-crafted descriptors outperform state-of-the-art learned descriptors. The result shows that current learned descriptors have a high variance for different datasets. Furthermore, the authors think it is necessary for future learned descriptors to train on more data to achieve better performance.

Another paper [3] strengthens the viewpoint by experiment on color texture classification. Their result shows that learned descriptors generally outperform hand-crafted descriptors but fail in some datasets. Such as the case of Outex 14 [3] that learned descriptors may not deal with the simultaneous change in the direction and temperature of the light. With these previous researches, we want to validate the result under the constraint of our specific testing database and matching method, then further find out new ways for improvement.

2.3. Novel architecture

Learning from the conclusion of the paper [8] that current local patches datasets are not large and diverse enough for good descriptors for general cases, a new method called HardNet was proposed [10]. The result reveals that this method can increase the matching accuracy with the same datasets which other state-of-the-art models used. In the paper, HardNet outperforms both hand-crafted and learned descriptors that we also test the performance of HardNet on our dataset.

3. Technical approach

We use hand-crafted methods to extract features for the comparison in the next step. At the same time, by using the training dataset, we train different neural network models with triplet loss to get various learned descriptors.

3.1. Hand-crafted descriptors

To learn the features of images, we use SIFT to extract key points. SIFT will produce 128-dimension descriptors for each key point. This extracting method is used widely due to its scale invariant property [11]. Moreover, we use RootSIFT to generate key points and descriptors for higher accuracy. RootSIFT is an extension of SIFT which allows SIFT descriptors to be compared using Hellinger kernel [15]. The algorithm steps of RootSIFT is computing SIFT descriptors, L1-normalizing each SIFT vector, and taking the square root of each element in the SIFT vector [15].

3.2. Learned descriptors

3.2.1 Training dataset.

To learn local image descriptors data, we use Photo Tourism dataset collected by Winder et al [16]. The dataset consists of 1024 x 1024 bitmap (.bmp) images, each containing a 16 x 16 array of image patches. Each patch is sampled as a 64 x 64 grayscale image [16]. All the patches in the dataset are extracted around real interest points from several photo collections published on International Journal of Computer Vision [17]. There are three subsets in the dataset: Liberty, Notredame, and Yosemite. Each has 100k, 200k, and 500k pre-generated labeled pairs and all come with 50% matching.

3.2.2 Loss and sampling

In our project, the goal is to increase the accuracy of deciding whether two images are referring to the same object. Instead of using the softmax cross entropy loss, it is better to apply the triplet loss which enables a variable number of classes [18]. In order to get balanced training samples, the triplet pairs are randomly generated. Each pair has an anchor, a positive of the same class as the anchor, and a negative of a different class.

3.2.3 Network models

There are three models we select for descriptors learning: ResNet [12], HardNet [10] and MatchNet [13]. The parameters are unified to make sure a fair comparison. Each model is trained for 50 epochs and we use the best result in these epochs. We set the optimizer as stochastic gradient descent (SGD) with learning rate 10, momentum 0.9 and weight decay 0.0001. To be noticed, zero padding is applied to all convolutional layers of all models.

We follow the structure of HardNet to get the desired result. However, we found it difficult to apply original MatchNet and ResNet due to the limitation of the GPU memory in the OSU server. To solve the problem, we modify the structure of MatchNet and ResNet. For MatchNet, we remove the last two layers to make output become 128 dimensions. Because we found max pooling layers dramatically decrease the performance of the

descriptors, we substitute them with convolution layers of stride 2. For ResNet, we reduce the number of layers from four to three. We also remove the last fully connected layer. To fit the model to our training dataset, we change the input image from RGB to grayscale. Finally, we modify the output channels from [64, 128, 256] to [32, 64, 128]. We call the modified model "PockNet", and the structure is shown in Table 1. We try three kinds of models with different layers and choose 44-layer one as our final model.

Layer name	Output size	14-layer	18-layer	44-layer
conv1	16X16	7X7, 32 stride 2		
con2_x	16X16	[3X3, 32] * 2 3X3, 32	[3X3, 32] * 2 3X3, 32	1X1, 32 [3X3, 32] * 3 1X1, 124
con3_x	8X8	[3X3, 64] * 2 3X3, 64	[3X3, 64] * 4 3X3, 64	1X1, 64 [3X3, 64] * 8 1X1, 256
con4_x	4X4	[3X3, 128] * 2 3X3, 128	[3X3, 128] * 2 3X3, 128	1X1, 128 [3X3, 128] * 3 1X1, 512
FC	1X1	average pool		

Table 1: Structure of PockNet

4. Experiment

To further validate the matching ability of each set of descriptors, we apply the one-to-one matching method to pair query images and test images. The result can be easily examined by precision-recall curves.

4.1. Environment

All experimental processes were conducted on Oregon State University (OSU) server with NVIDIA GeForce GTX 1080 under CUDA 10.0. The coding environment is Pytorch with the auxiliary of PIL, Numpy, Pandas, Matplotlib, and OpenCV. During the training process, the feature net without bottleneck runs at 30 it/sec.

4.2. Design

For testing the design of models, we made two kinds of structures that one learns the descriptors from the whole image, while another learns the descriptors from the smaller patches of the image detected by SIFT. We will use the names of first structure and the second structure to mention these two designs in the following parts.

The testing dataset consists of 175 colored images with ground truths while 35 are query images and 140 are for matching. Each image displays an object taken at random angles and distance from the camera center. The background of each image is white, however, chromatism and luminance may be slightly different. Query images represent 35 different objects, each query image maps to four images with the same object in the matching set.

4.2.1 Get features

For the first structure which learns the descriptors from the whole image, we derive 128 features by forward passing the images to each model. The second structure learns the descriptors from patches. By using SIFT which is built in the OpenCV library to detect keypoints, we can get a set of 30 x 128 features for 30 patches detected in each image. In contrast, there are more procedures for obtaining learned descriptors. First, we attain the weights for different neural network models by training on the Photo Tourism dataset, then forward passing 30 patches of each image to the network to retrieve 30 x 128 features.

4.2.2 Build cost matrix

With the features from either hand-crafted descriptors or learned descriptors, we compute the cost matrix of matching feature pairs by Euclidean norm. In other words, there is a 30 x 30 cost matrix for each query and matching image pair in the second structure. Every element in the cost matrix means the distance between two patches. In total, the tensor size of all pairs of cost matrixes is 35 x 140 x 30 x 30 for the second structure, and the tensor size of the cost matrixes in the first structure is 35 x 140.

4.2.3 Build similarity matrix

In order to calculate the similarity between the query image and the matching image, we apply the Hungarian algorithm to achieve one-to-one matching [19]. Every patch in the query image matches to one patch in the matching image. As a result, in the second structure, all 30 patches in the query image will have a distinct matchup in the matching image. The similarity of patches can be derived from the formula: $\exp(-d(f, f'))$, where $d(f, f')$ is the distance between two matched patches. Finally, by adding up 30 similarity results, we get the similarity between two images. For every query image, there are 140 similarity results corresponding to 140 matching images. We select the largest similarity results as the matching pairs that they are categorized as the same object as the query object by our algorithm. To be noticed, there is no need to do one-to-one matching on the first structure but calculate the similarity form the cost matrixes directly.

4.3. Result

There are two kinds of results in the project, one is the testing accuracy on Notredame-FPR95 dataset during the training for each epoch, another is the matching accuracy on our testing dataset with 175 images.

4.3.1 Testing on Notredame-FPR95

The accuracy refers to the error rate of matching. We stop ResNet18 and ResNet50 early because their performances cannot meet our anticipation. In Figure 1,

HardNet has the lowest loss while MatcheNet results in a much higher loss. PockNet, which is our modification from ResNet, performs better than MatchNet but fails to surpass HardNet. The lowest losses of three useful models for matching are listed in Table 2.

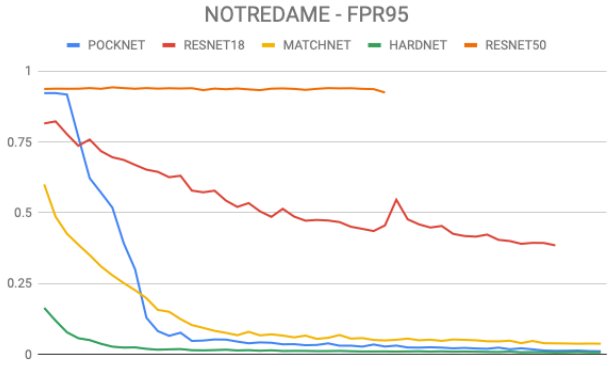

Figure 1: Accuracy vs. training epochs

train	Liberty	
Test (FPR (%))	Notredame	Yosemite
HardNet	0.67	2.6
MatchNet	3.7	9.88
PockNet	1.036	4.394

Table 2: Loss of models

4.3.2 One-to-one matching

Using a precision-recall curve (PRC) can summarize the trade-off between the true positive rate and the positive predictive value. Besides, PRC is more suitable than the ROC curve when facing imbalanced datasets [20]. In addition to selecting the largest four similarity results, we also select the largest one, two, and three similarity results separately to compare different models. The matching result can be easily checked by the PRC in Figure 2.

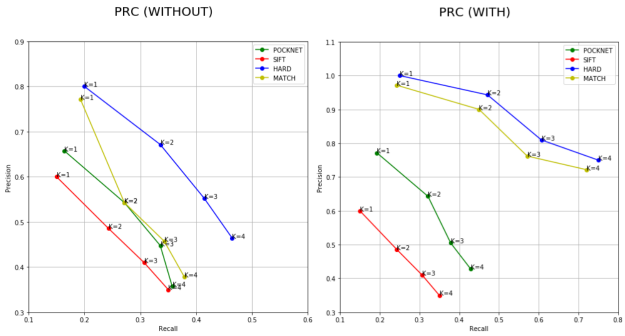

Figure 2: The left PRC is the result of the first structure without using SIFT for the inputs, which learns the descriptors from the whole image. The right PRC is the result of the second structure with SIFT for the inputs, which learns the descriptors from detected 30 patches.

5. Conclusion

In this paper, we present an experimental evaluation of handcrafted and learned descriptors to better understand their performance. Although we fail on improving our models by using RootSIFT that its performance is much lower than SIFT and being removed from our result, we still prove the practicality of hand-crafted descriptors that it could extract local features and improve models. The result shows that learned descriptors are better than hand-crafted descriptors. However, we are not sure of the outcome with advanced hand-crafted descriptors.

Though the performance of our PockNet does not exceed HardNet, we are confident in making a better model if we were not constraint by the power of GPU, which not only limit our design on parameters but batch size. We are not saying that HardNet is easy to beat because its performance might increase if there is no GPU constraint. Our point is there is still much for us to improve.

We discover that deeper network does not equal to a better network. If we increase the layers of a network, we need to modify parameters at the same time, or the outcome would be bad. Besides, the using of the OSU server makes it hard for us to train on complicated models for a long time. Not only because of the GPU limitation, but the server would disconnect automatically due to safety concern if we train for a long time.

6. Future work

The first thing we want to work on is fixing the advanced hand-crafted descriptors to verify others' research. To achieve a better model, we think about getting GPU with higher capacity for more complicated design and larger training epochs. We also consider using a benchmark such as HPatches [14] to make our experiment more thorough on the testing environment. Besides, we met some problems in our datasets while transforming images between RGB and grayscale that some outcomes become worse. It might improve our overall testing if we supersede the datasets with more suitable ones. The method for defining loss is another aspect to improve. We are thinking about substituting our random sampling method to online triplet mining. Moreover, we want to do experiments on more descriptors, especially for hand-crafted ones. There may be different detection conditions defined as easy, hard, tough detector noise [10]. We believe that with these improvements, our work can reach another level.

References

[1] W. Zhou, H. Li, and Q. Tian. Recent advance in content-based image retrieval: a literature survey. 2017.

[2] S. Berkiten and S. Rusinkiewicz. Alignment of images captured under different light directions. 2014.

[3] P. Napoletano. Hand-crafted vs Learned descriptors for color texture classification. 2017.

[4] M. Yang. Object recognition. 2009.

[5] Y. Wu, J. Lim, and M, Yang. Online object tracking: a benchmark. 2013.

[6] M. Zollhöfer, P. Stotko, A. Görlitz, C. Theobalt, M. Nießner, R. Klein, and A. Kolb. State of the art on 3D reconstruction with RGB-D cameras. 2018.

[7] G. Capi, S. Kaneko, and B. Hua. Neural network based guide robot navigation: an evolutionary approach. 2015.

[8] J. Schönberger, H. Hardmeier, T. Sattler, and M. Pollefeys. Comparative evaluation of hand-crafted and learned local features. 2017.

[9] M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. PAMI, 2011.

[10] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas. Working hard to know your neighbor's margins: local descriptor learning loss. 2017.

[11] D. Lowe. Distinctive image features from scale-invariant keypoints. 2004.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. 2015.

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

[14] V. Balntas, K. Lenc, A. Vedaldi. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. 2017.

[15] R. Arandjelovic, A. Zisserman. Three things everyone should know to improve object retrieval. 2012.

[16] S. Winder and M. Brown. 2007. http://phototour.cs.washington.edu/patches/default.html.

[17] N. Snavely, S. Seitz, and R. Szeliski. Modeling the world from internet photo collections. International Journal of Computer Vision, 80(2):189–210, 2008.

[18] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. 2015.

[19] H. Kuhn, "The Hungarian method for the assignment problem", Naval Research Logistics Quarterly, 2: 83–97, 1955.

[20] J. Davis, M. Goadrich. The Relationship Between Precision-Recall and ROC Curves. 2005.