H-Net: Neural Network for Cross-domain Image Patch Matching

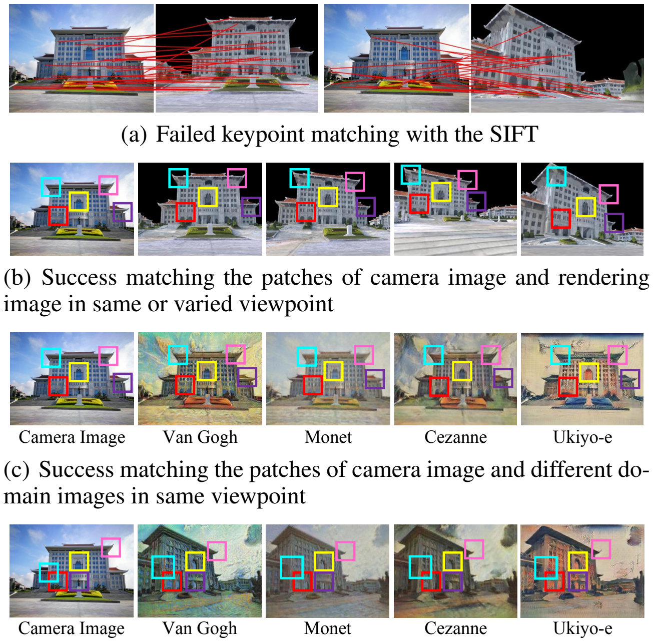
Weiquan Liu[1]*, Xuelun Shen[1]*, Cheng Wang[1]†, Zhihong Zhang[2], Chenglu Wen[1], Jonathan Li[1,3]
[1]Fujian Key Laboratory of Sensing and Computing for Smart City,
School of Information Science and Engineering, Xiamen University, Xiamen, China
[2]Software School, Xiamen University, Xiamen, China
[3]Department of Geography and Environmental Management, University of Waterloo, Waterloo, Canada
{wqliu1026, ziwuxuanxu}@163.com, {cwang, zhihong, clwen, junli}@xmu.edu.cn

Abstract

Describing the same scene with different imaging style or rendering image from its 3D model gives us different domain images. Different domain images tend to have a gap and different local appearances, which raise the main challenge on the cross-domain image patch matching. In this paper, we propose to incorporate AutoEncoder into Siamese network, named as H-Net, of which the structural shape resembles the letter H. The H-Net achieves state-of-the-art performance on the cross-domain image patch matching. Furthermore, we improved H-Net to H-Net++. The H-Net++ extracts invariant feature descriptors in cross-domain image patches and achieves state-of-the-art performance by feature retrieval in Euclidean space. As there is no benchmark dataset including cross-domain images, we made a cross-domain image dataset which consists of camera images, rendering images from UAV 3D model, and images generated by CycleGAN algorithm. Experiments show that the proposed H-Net and H-Net++ outperform the existing algorithms. Our code and cross-domain image dataset are available at https://github.com/Xylon-Sean/H-Net.

1 Introduction

Comparing local patches across images is a fundamental computer vision and image analysis problem, such as image retrieval [Simo-Serra *et al.*, 2015; Vo and Hays, 2016], image matching [Chen *et al.*, 2015; Hu and Lin, 2016], image alignment [Hsu *et al.*, 2015] and wide-baseline matching [Matas *et al.*, 2004]. Ideal feature descriptors should be invariant for matching patches and distinctive for non-matching patches. Currently, two main categories of feature description methods for image patch matching have been developed. One is based on hand-crafted features, like SIFT [Lowe, 2004], ORB [Rublee *et al.*, 2011], BRISK [Leutenegger *et al.*, 2011], the other one is neural-network-based method, which extracts feature descriptors [Simo-Serra *et al.*, 2015; Kumar *et al.*, 2016; Bailer *et al.*, 2017; Yang *et al.*, 2017]



(a) Failed keypoint matching with the SIFT

(b) Success matching the patches of camera image and rendering image in same or varied viewpoint

(c) Success matching the patches of camera image and different domain images in same viewpoint

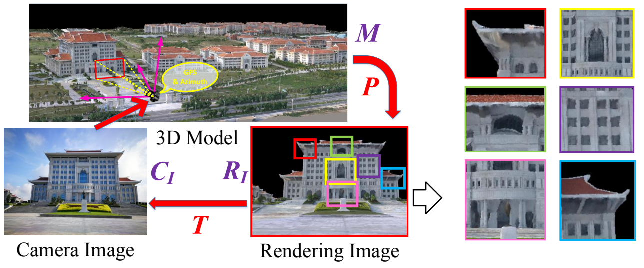
(d) Success matching the patches of camera image and different domain images in varied viewpoint

Figure 1: The results of cross-domain image patch matching. (a) is the failed result of camera image match rendering image from UAV 3D model by SIFT. (b), (c), and (d) are the results of cross-domain image patch matching by H-Net++. (b) is the camera image with the rendering images from UAV 3D model in the same and varied viewpoint, (c) and (d) are the camera image with the different domain images created by cycleGAN in the same or different viewpoint. The same color bounding box represents the corresponding pairs of patches.

In this paper, we aim to match the cross-domain image and implement image retrieval based on patches, as the results shown in Figure 1(b-d). We built a cross-domain image dataset, which includes camera images (real images) of a scene, rendering images (synthetic images) generated from Unmanned Aerial Vehicle (UAV) 3D model of the same

*indicates equal contribution

†Corresponding author



(a) The schematic of rendering image (b) Enlargement of rendering image

Figure 2: The schematic of rendering image in (a). The detail enlargement of rendering image in (b), the colorful bounding box represents the details correspond to the Rendering Image in (a).

scene, and images generated by CycleGAN algorithm. The schematic of rendering image from UAV 3D model is shown in Figure 2(a). The motivation of our work is to find the mapping relationship between the camera images and 3D models, and to provide a potential solution for the virtual-reality registration of the Augmented Reality. As shown in Figure 2(a), we mark the 3D model as M, the rendering image as R_I, and the camera image as C_I. The camera images from mobile devices provide an important clue for location and orientation estimation of the user in the 3D environment. By using the positioning information from the camera images, 3D model can roughly render an image of a scene which is the same as the camera image scene. R_I is obtained by a 3D model through a projection matrix P, that is $M \cdot P \rightarrow R_I$, and each point in the 3D model corresponds to a pixel in the rendering image. Our goal is to match the camera image with the rendering image through a transformation matrix T, $R_I \cdot T \rightarrow C_I$. Therefore, the correspondence between the camera image and the 3D model is $M \cdot P \cdot T \rightarrow C_I$. However, the rendering images from UAV 3D model are generally of low quality, such as large distortion, blurred resolution, structural repetitiveness, and occlusions, as shown in Figure 2(b). These go beyond the reach of the hand-crafted features, for example, the SIFT [Lowe, 2004] which failed as shown in Figure 1(a).

To match camera images and rendering images from UAV 3D model, we use the Siamese networks, which can be divided into two categories depending on whether there are metric networks. Siamese networks with metric network can only judge patch matching by binary classification [Han *et al.*, 2015; Kumar *et al.*, 2016; Altwaijry *et al.*, 2016]. The main drawback of these models is that the extracted feature descriptors cannot be applied with nearest neighbor search (NNS). On the other hand, Siamese networks without metric network usually use Euclidean distance as loss function [Simo-Serra *et al.*, 2015; Melekhov *et al.*, 2016; Tian *et al.*, 2017], of which the output feature descriptors are retrieved by Euclidean distance. These feature descriptors outperformed the previous hand-crafted feature descriptors. However, Siamese networks only perform successfully on the same type of images or images with little distortion, but failed to perform on cross-domain image patch matching and retrieval.

Inspired by the Siamese networks framework and AutoEncoder [Hinton and Sala-khutdinov, 2006], we propose H-Net and H-Net++ for cross-domain image patch matching and retrieval respectively. The proposed H-Net is made up of two same AutoEncoders with different weights, and it feeds the feature maps from the Encoder network into metric network, which consists of two convolutional and four fully connected layers. H-Net is optimized by mean squared error (MSE) for AutoEncoder and hinge loss for metric network. Experiments show that H-Net achieves state-of-the-art performance in patch matching, but cannot apply to feature descriptors retrieval. To overcome this drawback, we upgrade H-Net to H-Net++. Compared with H-Net, H-Net++ removes the metric network and uses Euclidean distance as loss function for the feature descriptors extracted in the AutoEncoder. H-Net++ extracts invariant feature descriptors in cross-domain image patch pair and directly uses NNS to retrieve the cross-domain image patches in Euclidean space. Our goal is to extract robust, representative, and invariant feature descriptors in the cross-domain image patches. Furthermore, to validate the robustness and generalization of H-Net and H-Net++, we utilize CycleGAN [Zhu *et al.*, 2017] to generate another four kinds of cross-domain images as additional validation sets. We also compare our proposed networks with existing mainstream Siamese networks.

In summary, the cross-domain images have a gap and their distribution is inconsistent. Adding AutoEncoder into Siamese network strongly enhances the ability of the two branches in H-Net and H-Net++ to extract the feature descriptors in cross-domain images. And the feature descriptors extracted by AutoEncoder in H-Net and H-Net++ are more robust. Specifically, the intermediate feature maps in Encoder of H-Net have richer representative details. The feature descriptors extracted by H-Net++ are more invariant and the distribution tends to be consistent.

Our main contributions are as follows:

1) The proposed H-Net improves the robustness of the feature descriptors in the cross-domain images by incorporating AutoEncoder into the Siamese network. And it achieves state-of-the-art performance on the cross-domain image patch matching.

2) The proposed H-Net++ makes the feature descriptors of cross-domain image patches consistent. It maps the feature descriptors into the same space, which can be retrieved in Euclidean space based on NNS.

3) We create a cross-domain image patch dataset which consists of camera images, rendering images from UAV 3D model, and images generated by CycleGAN algorithm. We will publish them to encourage future research.

2 Related Work

The research of designing image local feature descriptors has gradually shifted from hand-crafted features to CNN learning based features. The cross-domain images with huge distortion, blurred resolution, structural repetitiveness and occlusions go beyond the performance of hand-crafted feature descriptors. For hand-crafted feature descriptors, please refer to [Li and Allinson, 2008] for a review of the traditional

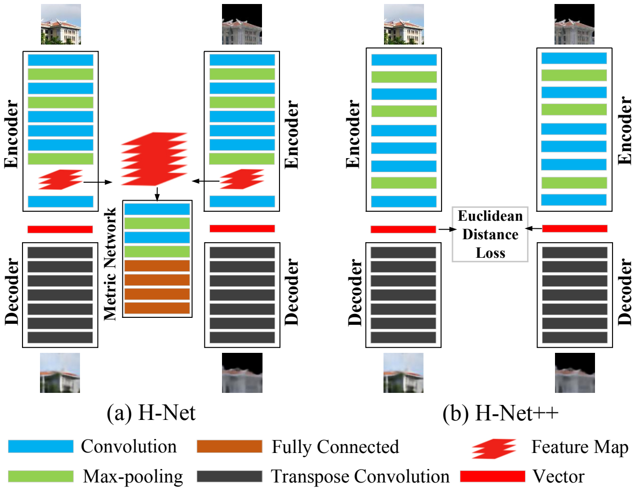


Figure 3: Network architecture of H-Net and H-Net++.

method, and [Zheng *et al.*, 2017] compared the classic feature descriptors with the CNN-based local feature descriptors. In this section, we will only review related Siamese networks.

2.1 Siamese Network with Metric network

More recently, in order to improve the performance of the feature descriptors extracted by the Siamese networks, the feature descriptors were usually exploited by concurrent work on joint feature descriptors and metric learning. Metric network allows the feature descriptors with pair of images to be considered jointly.

MatchNet [Han *et al.*, 2015] is one of the typical Siamese network, which constitutes two CNN branches sharing the parameters for feature encoding, followed by metric network to minimize the cross-entropy loss. CNN shows significant potential in feature descriptor learning. Based on Siamese network, [Zagoruyko and Komodakis, 2015] creatively presents 2-channel, 2-channel deep, 2-channel 2-stream and central-surround two-stream networks. The 2-channel and 2-channel deep network considers the two image patches as a 2-channel image, and directly feed them into single channel CNN. This model provides greater flexibility as it starts by processing the two image patches jointly. The 2-channel 2-stream network consists of two 2-channel networks. The 2-stream and central-surround two-stream network belong to a four-branch network which consists of two separate streams with central and surround, it takes multi-resolution as the inputs to improve the performance of image matching. These three models all outperform MatchNet. DeepCD 2-stream [Yang *et al.*, 2017] utilizes two completely different branches based on PN-Net [Balntas *et al.*, 2016] to learn two complementary feature descriptors, and jointly obtain a binary descriptor for patch matching. Binary CS L2-Net [Tian *et al.*, 2017] is state-of-the-art in the patch matching. The loss function of this network is constrained by the training data sample, and it is difficult to be applied to the common patch matching problem.

In general, although Siamese networks with metric network successfully work in image patch matching, their obvious drawback is that they cannot perform NNS for retrieval. And these methods often have an expensive cost on memory and computation.

2.2 Siamese Network without Metric network

As most matching tasks require NNS, many Siamese networks aim to learn high performance feature descriptors without metric network. These networks in [Simo-Serra *et al.*, 2015; Melekhov *et al.*, 2016; Lin *et al.*, 2015] are the traditional Siamese networks without metric network, which consist of two CNN branches and use Euclidean distance as the loss function, they extract the image feature descriptors by training deep convolutional models and achieving patch retrieval in Euclidean distance space. CS L2-Net [Tian *et al.*, 2017] is state-of-the-art in patch-based retrieval. The same as binary CS L2-Net [Tian *et al.*, 2017], the loss function of CS L2-Net is constrained by the training data sample, and CS L2-Net cannot be applied to the common patch-based retrieval problem.

3 Network Structure

In this section, we describe network architecture, loss function, training strategy of the proposed H-Net and H-Net++ in detail.

With huge distortion, blurred resolution, structural repetitiveness and occlusions in the rendering images, these rendering images look like camera images with a lot of noise. To overcome these challenges, we propose to incorporate the AutoEncoder into the Siamese network.

In detail,we denote the input patch in AutoEncoder as X and output patch as $X^{'}$, our goal is to learn the high performance feature descriptors V. We learn the Encoder as a mapping $F : F(X) \rightarrow V$, and learn the Decoder as an inverse mapping: $G : G(V) \rightarrow X^{'}$. We expect the distribution of X and $X^{'}$ as similar as possible to push $G(F(X)) \approx X$.

3.1 H-Net

The architecture of H-Net is depicted in Figure 3(a). The H-Net consists of two AutoEncoder modules and a metric network. The structure of the two AutoEncoder modules are identical, but the weights between them are not shared. The inputs of H-Net are a pair of cross-domain image patches with the size of $256 \times 256 \times 3$.

For the Encoder in AutoEncoder of H-Net, we use all convolutional layers with zero padding and max pooling layers without zero padding, the Batch normalization (BN) [Ioffe and Szegedy, 2015] is used after each convolution. Unlike previous Siamese networks [Simo-Serra *et al.*, 2015; Han *et al.*, 2015] using TanHyperbolic (Tanh) or Rectified Linear Units (ReLU) as the non-linear activate function, we use the Scaled Exponential Linear Units (SeLU) [Klambauer *et al.*, 2017]. The detail of the Encoder architecture is as follow: C(96,11,4)-BN-SeLU-P(3,2)-C(256,5,1)-BN-SeLU-P(3,2)-C(384,3,1)-BN-SeLU-C(384,3,1)-BN-SeLU-C(256,3,1)-BN-SeLU-P(3,2)-C(1024,7,1)-BN-SeLU. The shorthand notation of C(n, k, s) is that the convolution with n filters of kernel size $k \times k$ with the stride s, and P(k, s) is the max pooling of size $k \times k$ with the stride s. Finally, the size of the penultimate feature map in Encoder is $7 \times 7 \times 256$.

Importantly, unlike other networks extracting the feature descriptors by flattening the last feature map into a one-dimensional vector before using fully connected layer, we perform a convolution operation on the penultimate feature map obtained by the Encoder. The size of the last convolution kernel in Encoder is 7×7, and the Encoder finally obtains a 1024-dimensional feature descriptor.

For the Decoder in the AutoEncoder of H-Net, we use transpose convolution to reconstruct the 1024-dimensional feature descriptor as similar as the input image patches which size is $256 \times 256 \times 3$. It should be noted that the 1024 dimensional feature descriptors are extracted from the Encoder and need to be resized to $2 \times 2 \times 256$ before performing Decoding. The detail of Decoder architecture is that: TC(128,5,2)-SeLU-TC(64,5,2)-SeLU-TC(32,5,2)-SeLU-TC(16,5,2)-SeLU-TC(8,5,2)-SeLU-TC(4,5,2)-SeLU-TC(3,5,2)-Sigmoid. TC(n, k, s) is a transposed convolution with n filters of size $k \times k$ applied with the stride s.

The framework of the metric network in H-Net (Figure 3(a)) contains two convolution layers with zero padding, two max pooling layers without zero padding and four fully connected layers. The output of the metric network is a scalar in $(-1, 1)$ as the activate function in the last fully connected layer is Tanh. BN is used after each convolution layer and fully connected layer, and the non-linear activate function in the convolution layers and fully connected layers is SeLU, except the last fully connected layer which uses Tanh. Unlike most commonly used Siamese networks, which ignore the importance of the intermediate feature maps, H-Net merges the penultimate feature map extracted by the two Encoder as the input of the metric network. Moreover, instead of flattening the merged feature map directly into a one-dimensional eigenvector and feeding it into the fully connected layers, we process the merged feature map with two convolution layers so that the cross-domain image patches can be considered jointly with more details. The detail of metric network is: C(1024,2,1)-BN-SeLU-P(2,2)-C(2048,2,1)-BN-SeLU-P(2,2)-FC(2048)-BN-SeLU-FC(256)-BN-SeLU-FC(128)-BN-SeLU-FC(64)-BN-SeLU-FC(1)-BN-Tanh. The FC(n) denote a fully connected layer with n output units.

Loss function in H-Net. To optimize the H-Net, we use three loss function terms, the first two terms are for AutoEncoder, the last one is for metric network.

In detail, for the two AutoEncoder branches in H-Net, we use the mean squared error (MSE):

$$L_{\text{MSE}}(C, C^{'}) = ||C - C^{'}||_2^2 \quad (1)$$

$$L_{\text{MSE}}(R, R^{'}) = ||R - R^{'}||_2^2 \quad (2)$$

where C represents one of the AutoEncoder input camera image patch, and $C^{'}$ is the output of this AutoEncoder. Similarly, R represents another AutoEncoder input rendering image patch, and $R^{'}$ is the output of this AutoEncoder.

We use the hinge-based loss for the metric network in H-Net, this term can be expressed as:

$$L_{hinge} = \sum_{i=1}^{N} max(0, 1 - y_i \cdot O_i) \quad (3)$$

where O_i is the output of the metric network for the i-th pair of cross-domain image patches, and $y_i \in \{-1, 1\}$ is the label of the training data. When the input pair of patches are matching, $y_i = 1$, otherwise, $y_i = -1$.

Training strategy for H-Net. Our goal is to minimize these three loss function terms in H-Net. However, to ensure the feature descriptors extracted by AutoEncoder are not influenced by the metric network, we add some constraints for H-Net when updating the parameters. Specifically, the parameters updating of H-Net is divided into two parts during training. First, only the parameters of the two AutoEncoder are updated at the minimum $L_{\text{MSE}}(C, C^{'}) + L_{\text{MSE}}(R, R^{'}))$. Second, only the parameters of the metric network are updated when the L_{hinge} is minimized. These two parts are done simultaneously and independently when training.

The detailed updating schedule is: (1) sampling a mini-batch *B* from training datasets; (2) fixing the parameters of metric network and train once the AutoEncoder by feeding into the training data *B*; (3) fixing the parameters of AutoEncoder and training once the metric network by feed in the same training data *B*; (4) repeating from (1) to (3) until loss convergence.

3.2 H-Net++

Since the feature descriptors of the cross-domain image patches extracted by H-Net cannot be used for retrieval, we improve H-Net to H-Net++, which can extract invariant and more robust, representative feature descriptors in cross-domain image patches.

The architecture of H-Net++ is shown in Figure 3(b). The network structure and details of the two AutoEncoders in H-Net++ are the same to the two AutoEncoders in H-Net. The main difference between H-Net++ and H-Net is that H-Net++ replaces the metric network of H-Net with Euclidean distance constraint.

Loss function in H-Net++. For the input cross-domain image patch pair C and R, we utilize the MSE for the two AutoEncoder in H-Net++ by formula (1) and (2). For the Euclidean distance constraint of the two feature descriptors extracted by H-Net++, we use the margin-based contrastive loss as another loss function term

$$L_{contrastive} = \frac{1}{2} l D^2 + \frac{1}{2}(1 - l)\{max(0, m - D)\}^2 \quad (4)$$

where l is the label of the pair of cross-domain image patches whether matching, when the input pair of patches are matching, $l = 1$, otherwise, $l = 0$. $m > 0$ is the margin for dissimilar pair of patch. And $D = ||F(C)–F(R)||_2$ is the Euclidean distance between feature descriptors $F(C)$ and $F(R)$ of input cross-domain image patches C and R. This margin-based contrastive loss encourages matching pairs to be close, and non-matching pairs to have Euclidean distance of at least margin m. So, the loss function in H-Net++ can be expressed as:

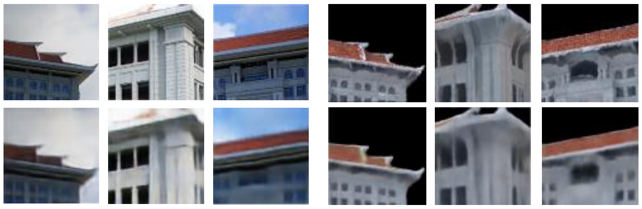
$$L_{\text{H-Net++}} = L_{\text{MSE}}(C, C^{'}) + L_{\text{MSE}}(R, R^{'}) + L_{contrastive} \quad (5)$$

	H-Net	L2-Net	CS L2-Net	DeepCD 2-stream	HybridNet	MatchNet	Pseudo- Siam	2ch- 2stream	2ch- deep	CS- 2stream
Accuracy	**0.9732**	0.8963	0.8811	0.8334	0.9116	0.6376	0.6493	0.6737	0.8856	0.7086
Recall	**0.9811**	0.9002	0.8857	0.8564	0.9123	0.6588	0.6677	0.6877	0.8901	0.7144
Precision	**0.9663**	0.8913	0.8755	0.8161	0.9094	0.6275	0.6395	0.6646	0.8801	0.7021

Table 1: The performance of patch matching by H-Net and comparative networks on the cross-domain image patch dataset.

Dimension	2048	1024	512	256	128
Accuracy	0.9688	**0.9732**	0.9637	0.9669	0.964
Recall	0.9811	**0.9811**	0.964	0.9782	0.9791
Precision	0.957	**0.9663**	0.9627	0.956	0.9498

Table 2: The performance of the cross-domain image patch matching in H-Net with different feature descriptor dimensions.



(a) Camera image patches (b) Rendering image patches

Figure 4: The visualization of the generated patches through the two AutoEncoder in H-Net with the 1024-dimensional feature descriptors. The top row is the input of patches, the bottom is the generated patches.

Training strategy for H-Net++. Different with the training strategy in H-Net, there is no metric network in H-Net++, we directly minimize $L_{\mathrm{H-Net++}}$ and all the parameters of H-Net++ are simultaneously updated together.

4 Experiments

In this section, we first describe the cross-domain image dataset used in our experiment. Then we provide some comparison among the proposed H-Net and the existing mainstream Siamese networks with metric network, and then we compare H-Net++ with several existing Siamese networks without metric network in retrieval performance. Finally, we discuss and analyze the proposed H-Net and H-Net++.

We implemented the proposed H-Net and H-Net++ with Tensorflow. All the experiments were performed on a NVIDIA Tesla P100. Our models were trained by using Adaptive Moment Estimation (Adam) Optimizer. The learning rate starts at 0.001 and decays 0.9 for each epoch.

4.1 Dataset

As there is no benchmark dataset for cross-domain images, we collected 10,000 pairs of cross-domain images which are camera images and their corresponding rendering images, the schematic of these cross-domain images is shown in Figure 2(a). Based on these corresponding pairs of cross-domain images, we collected 100,000 matching pairs of patches and 100,000 non-matching pairs of patches.

For the proposed networks and the comparative networks, we selected 160,000 pairs of cross-domain image patches as training data, in which 80,000 pairs of patches are matching and 80,000 pairs patches are non-matching. For testing, we used the remaining 40,000 pairs of cross-domain image patches as the testing data, of which, 20,000 pairs of patches are matching and 20,000 pairs of patches are non-matching.

The patch size in the camera images and rendering images is approximately at 512×512. We resize these cross-domain image patches into 256×256 as the input for the proposed networks. The camera images and rendering images are 3-channel color image, so the input of our proposed networks are patches with size $256 \times 256 \times 3$.

In addition, in order to enrich our cross-domain image dataset and demonstrate the robustness and generalization performance of the proposed networks, we utilize the cycle-GAN to simulate four kinds of cross-domain images, which render the camera images to the style of Van Gogh, Monet, Cezanne, Ukiyo-e, as shown in Figure 1(c) and (d).

4.2 Performance on H-net

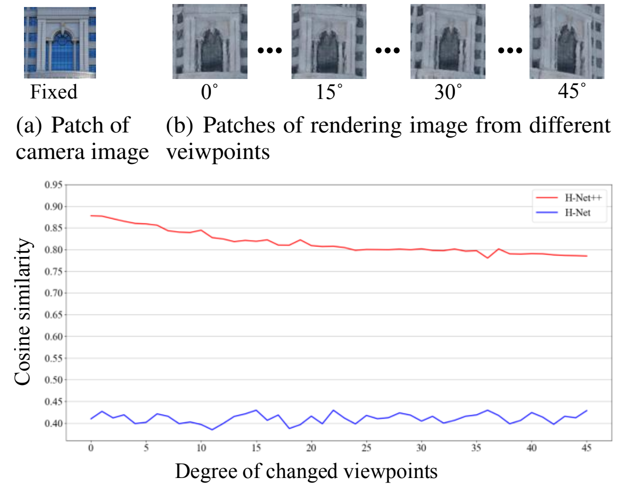
We compare the proposed H-Net with several existing mainstream structures of Siamese networks with metric network. As there is similar number of positive and negative in the training data and the testing data, we use accuracy, recall, and precision to measure the performance of the networks. The results are listed in Table 1.

MatchNet [Han *et al.*, 2015] and Hybrid Network [Altwaijry *et al.*, 2016] consist of two CNN branches that share weights, and use cross entropy as loss function after metric network. Pseudo-siam, 2ch-2stream, 2ch-deep, and CS-2stream [Zagoruyko and Komodakis, 2015] all use the hinge loss as the loss function. DeepCD 2-stream [Yang *et al.*, 2017] consists of two completely different CNN branches. L2-Net [Tian *et al.*, 2017] has state-of-the-art performance on patch matching. Although the loss function in L2-Net cannot be applied to the general patch matching problem, we compare its branch structure with our H-Net using our metric network and the hinge loss as loss function. The structure of center surround in CS L2-Net [Tian *et al.*, 2017] is different from the center surround in CS-2stream [Zagoruyko and Komodakis, 2015], we also compared the structure of CS L2-Net. We implemented the aforementioned networks strictly according to the description in the papers.

As shown in Table 1, the proposed H-Net performs the best in the cross-domain image patch matching. Compared with other mainstream Siamese network frameworks, H-Net has significant improvements. We also investigated the dimen-

	H-Net++	VGG16 H-Net++	ResNet H-Net++	DeepCD2-stream	Siam_l2	Simo
TOP 1	**0.7056**	0.6115	0.4426	0.5223	0.3652	0.4389
TOP 5	**0.8571**	0.7293	0.5397	0.6604	0.4421	0.5417

Table 3: The retrieval accuracy on TOP 1 and TOP 5 by H-Net++ and comparative networks on the cross-domain image patch dataset.



(a) Patch of camera image

(b) Patches of rendering image from different veiwpoints

(c) The cosine similarity of fixed patch of camera image and varied viewpoint patches of rendering images calculated by H-Net and H-Net++

Figure 5: The cosine similarity calculated by H-Net and H-Net++.

sions of the feature descriptors extracted by AutoEncoder in H-Net, and whether they affect the performance in the cross-domain image patch matching. As shown in Table 2, whatever the dimension of feature descriptors extracted by H-Net, their performance all outperform the comparative Siamese networks listed in Table 1, and the 1024-dimensional feature descriptors extracted by Auto-Encoder in H-Net have the best performance in cross-domain image patch matching. We also visualize the generated images from the two AutoEncoder in H-Net, respectively in Figure 4. Above results demonstrate that H-Net is robust.

4.3 Performance on H-Net++

We compared the proposed H-Net++ with several existing Siamese networks without metric network, which are DeepCD 2-stream [Yang *et al.*, 2017], Siam_l2 [Zagoruyko and Komodakis, 2015], Simo [Simo-Serra *et al.*, 2015], VGG16 H-Net++ and ResNet H-Net++. The VGG16 H-Net++ is a H-Net++ with the Encoder replaced by VGG16 [Simonyan and Zisserman, 2014]. Similarly, the ResNet H-Net++ is a H-Net++ with the Encoder replaced by ResNet [He *et al.*, 2016]. We randomly selected 5,000 pairs of matching cross-domain image patches as the retrieval benchmark dataset from the testing data. And we use the TOP1 and TOP5 retrieval accuracy in Euclidean space to measure the performance of the networks. The results are listed in Table 3.

As shown in Table 3, compared with the Siamese networks which use Euclidean distance as the loss function, the proposed H-Net++ performs the best in the cross-domain image patch retrieval. Furthermore, we conducted the experiments of similarity score with different viewpoints (Figure 5). Figure 5(c) contains two curves showing the cosine similarities between the feature descriptors extracted by H-Net (red curve) and H-Net++ (blue curve), regarding a fixed viewpoint camera image patch (Figure 5(a)) and rendering image patches with different viewpoints from 0 degree to 45 degree (Figure 5(b)). The X-axis represents the viewpoint degree and the Y-axis represents the similarity between two patches, respectively. The similarity curves for pairs of patches calculated by H-Net++ tend to decrease slowly and keeping a high degree of similarity (as the training data are only paired cross-domain image patches corresponding to the viewpoint, the decrease in similarity is reasonable with the changed of viewpoints). While the similarity of paired patches calculated by H-Net is very low, around 40% (this similarity relationship does not have a regular pattern with the changed viewpoints). Therefore, the feature descriptors extracted by H-Net++ are invariant, more meaningful and representative.

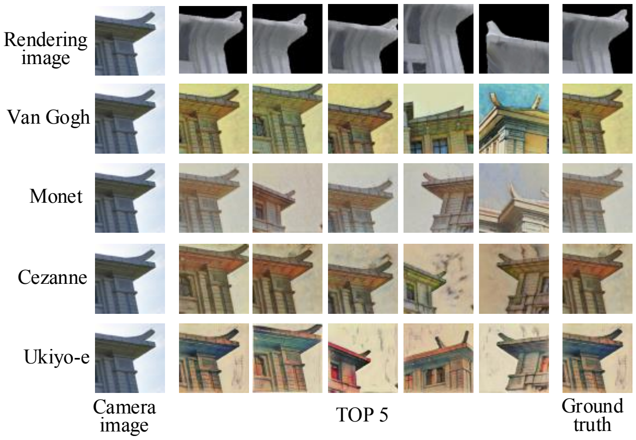


Figure 6: The retrieved TOP 5 with H-Net++ in five different domain image patches.

In particular, the feature descriptors extracted by H-Net++ have the characteristics of continuity. Whether the retrieved cross-domain images are from the same view or varied view, they can be retrieved (Figure 1(b)), and the red curve in Figure 5(c) also prove the feature descriptors extracted by H-Net++ are continuous. In order to demonstrate the generalization of H-Net++, we used the cross-domain images generated by cycleGAN to test the retrieval, as shown in Figure 1(c) and (d). They are directly applied to the H-Net++ which is trained with camera images and rendering images from UAV 3D model. Finally, in Figure 6, we show TOP5 retrieval results with a camera image patch retrieved in five different domain image patches. Above results demonstrate that the feature descriptors extracted by H-Net++ are robust.

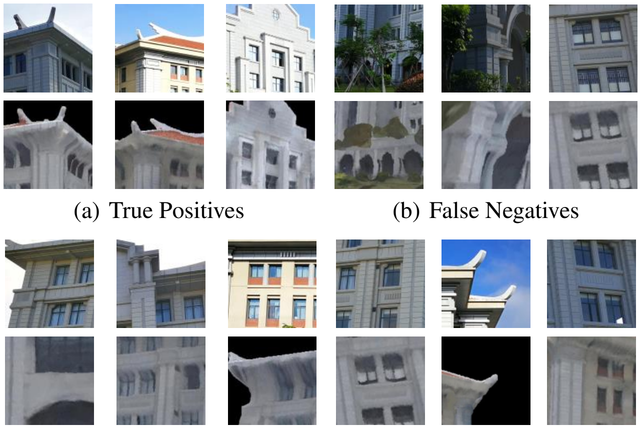


Figure 7: Top-ranking false and true matches by H-Net.

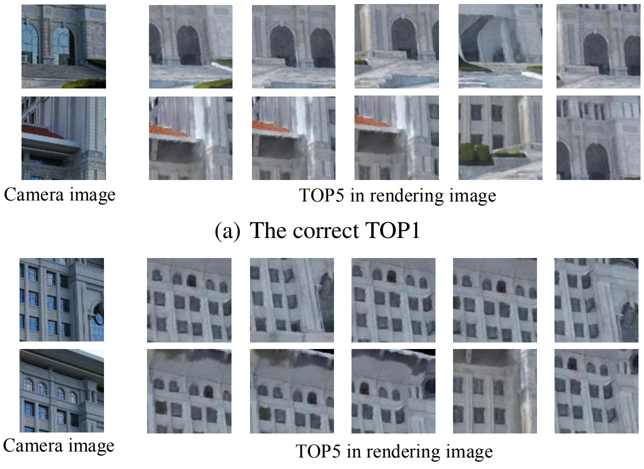


Figure 8: The retrieved TOP5 by H-Net++ for the first two corresponding cross-domain images in Figure 1(b).

4.4 Discussion and Analysis

Since the two different domain images tend to have a gap, they have different local appearances, and their distributions are inconsistent. Therefore, it is hard to apply the handcrafted feature descriptors or standard CNN-based descriptors for matching cross-domain image patches.

The backbone network of H-Net and H-Net++ is AutoEncoder. Compared with CNN-based Siamese network, AutoEncoder based network can more effectively capture the domain-specific information. The penultimate layer of CNN can extract the high-level (abstract) information, but greatly lose low level (close to pixel level) domain specific information. The encoder of AutoEncoder works similarly as CNN, but the decoder of AutoEncoder reconstructs the input image, which contains rich domain-specific information. Therefore, AutoEncoder is more effective for domain-specific information extraction.

For H-Net, the intermediate feature maps in Encoder have richer representative details due to the self-constraint by AutoEncoder. So, merging the intermediate feature maps from the two AutoEncoder into the metric network yields high performance in cross-domain image patch matching. The feature descriptors extracted by the two AutoEncoder in H-Net++ are constraint with Euclidean distance, which makes their distribution consistent and can be retrieved in Euclidean space.

In addition, the inputs of the metric network are the penultimate intermediate feature maps in Encoder, and metric network which consists of not only fully connected layers but also convolutional layers. These jointly consider the details in cross-domain images, and play a decisive role in the high performance of H-Net. Besides, the MSE loss function term and the updating strategy of network parameters also contribute to the high performance of H-Net and H-Net++.

The failed cases of the proposed H-Net and H-Net++ are shown in Figure 7 and Figure 8, respectively. In Figure 7, we show some top ranking correct and false matches by the H-Net. For the false negatives, the paired patches are with severe occlusion, huge distortion and the repeated structures. These pair of patches are also very challenging for human. For the false positives, the paired cross-domain image patches appear similar, which results in the misjudgment of H-Net. Figure 8 shows some TOP5 search results of patches by H-Net++ for the first two corresponding cross-domain images in Figure 1(b). We randomly selected 2,000 points in the two corresponding cross-domain images and obtained 2,000 patches respectively, then searched these 2,000 patches in the two images. The correct TOP1 retrieval results are shown in Figure 8(a), and the failed TOP1 retrieval are shown in Figure 8(b). We overserved from Figure 8(b) that large number of repeated structures (such as windows) have a great influence on the performance of H-Net++.

As for the benefits of H-Net and H-Net++ analyzed above, they can be easily applied to many cross-domain image matching applications, such as cross-sensor images matching, like RGB images to infrared images or hyperspectral images, CT images to MRI images, etc. In addition, H-Net++ can be also extended to cross-modal retrievals, such as images to text, voice to images, voice to text, etc.

5 Conclusion

In this paper, we present H-Net and H-Net++ by integrating AutoEncoder into the Siamese network for cross-domain image patch matching and retrieval, respectively. By utilizing the two intermediate feature maps from Encoder as the inputs of the metric network, the H-Net achieves state-of-the-art performance on the cross-domain image patch matching. Based on H-Net, H-Net++ add Euclidean distance constraint for the features extracted by AutoEncoder, so that the features can be retrieved in Euclidean space. And H-Net++ also significantly achieves the best retrieval performance on the cross-domain image. Besides, the feature extracted by H-Net++ are more robust and invariant. The high performance of H-Net and H-Net++ is mainly attributed to the loss function in AutoEncoder. In the future work, we plan to extend our proposed networks to other domain images and cross-modal data. To improve the retrieval performance in cross-domain images, we plan to balance the extracted sufficient representation and suppress the independent influence on the cross-domain images.

Acknowledgments

This work is supported by National Natural Science Foundation of China (No.U1605254, 61771413, 41471379).

References

[Altwaijry *et al.*, 2016] Hani Altwaijry, Eduard Trulls, James Hays, Pascal Fua, and Serge Belongie. Learning to match aerial images with deep attentive architectures. In *CVPR*, pages 3539–3547, 2016.

[Bailer *et al.*, 2017] Christian Bailer, Kiran Varanasi, and Didier Stricker. Cnn-based patch matching for optical flow with thresholded hinge embedding loss. In *CVPR*, volume 2, page 5, 2017.

[Balntas *et al.*, 2016] Vassileios Balntas, Edward Johns, Lilian Tang, and Krystian Mikolajczyk. Pn-net: conjoined triple deep network for learning local image descriptors. *arXiv preprint arXiv:1601.05030*, 2016.

[Chen *et al.*, 2015] Hsin-Yi Chen, Yen-Yu Lin, and Bing-Yu Chen. Co-segmentation guided hough transform for robust feature matching. *IEEE TPAMI*, 37(12):2388–2401, 2015.

[Han *et al.*, 2015] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *CVPR*, pages 3279–3286, 2015.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Hinton and Sala-khutdinov, 2006] Geoffrey E Hinton and Ruslan R Sala-khutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[Hsu *et al.*, 2015] Kuang-Jui Hsu, Yen-Yu Lin, and Yung-Yu Chuang. Robust image alignment with multiple feature descriptors and matching-guided neighborhoods. In *CVPR*, pages 1921–1930, 2015.

[Hu and Lin, 2016] Yuan-Ting Hu and Yen-Yu Lin. Progressive feature matching with alternate descriptor selection and correspondence enrichment. In *CVPR*, pages 346–354, 2016.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.

[Klambauer *et al.*, 2017] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. *NIPS*, pages 972–981, 2017.

[Kumar *et al.*, 2016] BG Kumar, Gustavo Carneiro, Ian Reid, et al. Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In *CVPR*, pages 5385–5394, 2016.

[Leutenegger *et al.*, 2011] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *ICCV*, pages 2548–2555, 2011.

[Li and Allinson, 2008] Jing Li and Nigel M Allinson. A comprehensive review of current local features for computer vision. *Neurocomputing*, 71(10):1771–1787, 2008.

[Lin *et al.*, 2015] Tsung-Yi Lin, Yin Cui, Serge Belongie, and James Hays. Learning deep representations for ground-to-aerial geolocalization. In *CVPR*, pages 5007–5015, 2015.

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

[Matas *et al.*, 2004] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[Melekhov *et al.*, 2016] Iaroslav Melekhov, Juho Kannala, and Esa Rahtu. Image patch matching using convolutional descriptors with euclidean distance. In *ACCV*, pages 638–653, 2016.

[Rublee *et al.*, 2011] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, pages 2564–2571, 2011.

[Simo-Serra *et al.*, 2015] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *ICCV*, pages 118–126, 2015.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Tian *et al.*, 2017] Yurun Tian, Bin Fan, Fuchao Wu, et al. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *CVPR*, volume 2, 2017.

[Vo and Hays, 2016] Nam N Vo and James Hays. Localizing and orienting street views using overhead imagery. In *ECCV*, pages 494–509. Springer, 2016.

[Yang *et al.*, 2017] Tsun-Yi Yang, Jo-Han Hsu, Yen-Yu Lin, and Yung-Yu Chuang. Deepcd: Learning deep complementary descriptors for patch representations. In *CVPR*, pages 3314–3322, 2017.

[Zagoruyko and Komodakis, 2015] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, pages 4353–4361, 2015.

[Zheng *et al.*, 2017] Liang Zheng, Yi Yang, and Qi Tian. Sift meets cnn: A decade survey of instance retrieval. *IEEE TPAMI*, 2017.

[Zhu *et al.*, 2017] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017.