

# Multi-Level Feature Aggregation-Based Joint Keypoint Detection and Description

Jun Li<sup>1</sup>, Xiang Li<sup>1</sup>, Yifei Wei<sup>1,\*</sup>, Mei Song<sup>1</sup> and Xiaojun Wang<sup>2</sup>

<sup>1</sup>Beijing Key Laboratory of Work Safety Intelligent Monitoring, School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876, China

<sup>2</sup>Dublin City University, Dublin, 9, Ireland

\*Corresponding Author: Yifei Wei. Email: weiyifei@bupt.edu.cn

Received: 06 March 2022; Accepted: 26 April 2022

**Abstract:** Image keypoint detection and description is a popular method to find pixel-level connections between images, which is a basic and critical step in many computer vision tasks. The existing methods are far from optimal in terms of keypoint positioning accuracy and generation of robust and discriminative descriptors. This paper proposes a new end-to-end self-supervised training deep learning network. The network uses a backbone feature encoder to extract multi-level feature maps, then performs joint image keypoint detection and description in a forward pass. On the one hand, in order to enhance the localization accuracy of keypoints and restore the local shape structure, the detector detects keypoints on feature maps of the same resolution as the original image. On the other hand, in order to enhance the ability to percept local shape details, the network utilizes multi-level features to generate robust feature descriptors with rich local shape information. A detailed comparison with traditional feature-based methods Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF) and deep learning methods on HPatches proves the effectiveness and robustness of the method proposed in this paper.

**Keywords:** Multi-scale information; keypoint detection and description; artificial intelligence

## 1 Introduction

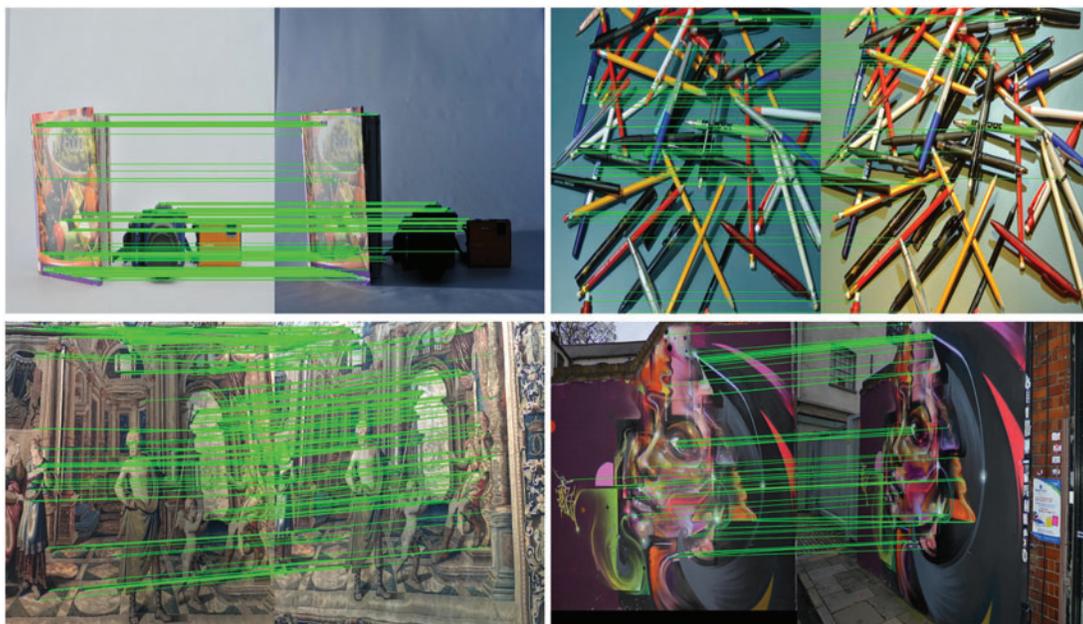
Finding the pixel-level connection between images is a basic step in many computer vision tasks such as Simultaneous Localization and Mapping (SLAM) [1], Structure from Motion (SfM) [2] and image matching [3,4]. Image keypoint detection and description is a popular method to achieve this connection [5–7]. As these tasks are faced with increasingly challenging scenes, such as indoor/outdoor scenes, illumination and weather changes. Thus, the positioning accuracy of keypoint and generation of robust and discriminative descriptors are very important.



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the past decades, many excellent algorithms have been proposed to complete joint local keypoint detection and description [8], such as SIFT [9], ORB [4], SURF [10] and other manually designed feature algorithms. With the rapid development of deep learning [11–14], more and more image matching methods based on deep learning have been proposed, such as D2-Net [15], SuperPoint [16] and ASLFeat [17]. However, few methods are robust in complex scenarios. What's more, most of the existing method based on deep learning tend to use the deepest features, which are shrunk several times (usually 4 or 8 times) compared to the original image for keypoint detection and description. This is yet to be optimized, because locating the coordinates of the keypoints in the deep feature maps will inevitably reduce the detection accuracy of the keypoints. At the same time, in the deep feature maps, the keypoints of different local structures may share the same high-level semantic information, which makes it difficult to extract distinguishing keypoint descriptors that retain rich local structural information.

The contributions of this paper are as follows: in order to solve the above problems, a new self-supervised deep learning method is proposed. This paper proposes two lightweight but effective modules: the keypoint detector and the keypoint descriptor. As the backbone feature encoder shares most of the calculations of these two modules, the network improves the positioning accuracy of keypoints and the discrimination of descriptors with a small computation cost. A detailed comparison with traditional feature-based methods SIFT, SURF and deep learning methods on HPatches [18] proves the effectiveness and robustness of the method proposed in this paper. The keypoint detector reorganizes the multi-level low-resolution feature maps into the original resolution of input images by bilinear up-sampling. So, keypoints are detected on the feature maps of the original image size, thus the positioning accuracy of the keypoints is improved. The keypoint descriptor aggregates multiple layers of features together to generate descriptors with rich local structure information. The example results of visual keypoints matching of the proposed network are shown in Fig. 1.



**Figure 1:** Visualization samples of matching keypoints on HPatches [18]. The proposed method can successfully find image correspondences even under big illumination or view point changes

## 2 Related Research

Joint local feature detection and description has been studied for a long time. SIFT, Oriented Fast and Rotated Brief (ORB), SURF and other hand-designed feature algorithms have achieved good results. Among these methods, SIFT occupies the mainstream position. SURF is the fast version of SIFT, while ORB is more efficient than these two algorithms and is suitable for tasks requiring high real-time requirements.

With the rise of deep learning, more and more methods have been proposed. Early methods follow the strategy of traditional feature extraction algorithms to detect firstly and then describe, and improve performance by replacing a certain step of keypoint detection or description with a learning method. For example, L2-Net [19] uses a hand-designed algorithm to detect keypoints and a convolutional neural network to generate descriptors. After that, Lf-Net [20] uses two sub-neural networks for keypoint detection and description respectively. Recently, the joint learning of detectors and descriptors has attracted more and more attention. The latest methods usually build a unified feature extraction network to extract deep features, then combining the two parts up, so as to achieve the result of sharing most of the calculation amount of the two tasks and real-time keypoints detection and description. D2-Net [15] uses dense feature description to detect keypoints at first, then closely combines detection loss and description loss to obtain a more integrated architecture. On the basis of D2-Net, ASLFeat [17] utilizes peak detection to detect keypoint on multi-level feature maps, greatly improving the positioning accuracy of keypoints. Since it is very difficult to obtain the ground truth labels for keypoint detection, SuperPoint [16] uses Homographic Adaptation to generate the keypoint labels, then trains the keypoint detector and descriptor together in a self-supervised way. However, most methods detect and describe keypoints on the deepest feature maps that are several times smaller than the original images, which needs to be optimized. In order to improve the network's perception ability and positioning accuracy of local shapes, two effective modules are proposed in this paper. The feature detector reorganizes the low-resolution feature maps of multiple layers to the original image resolution in a non-learning way, so as to improve the accuracy of keypoint location. The descriptor aggregates the multi-layer features together to generate descriptors with rich local structure information.

## 3 Proposed Method

### 3.1 Problem Statement

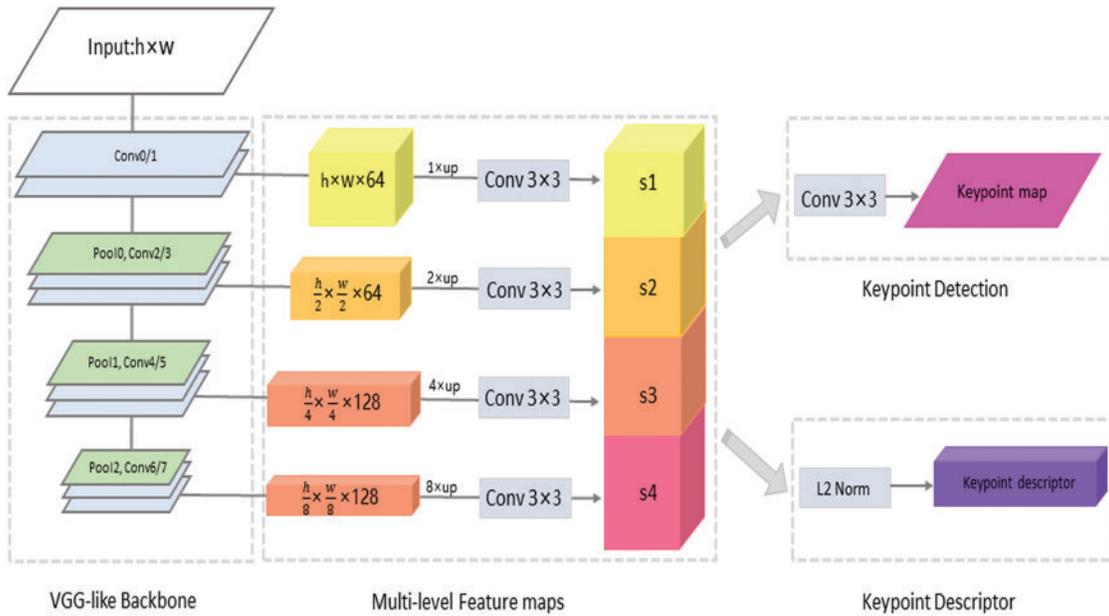
Geometric registration (or alignment) is a critical task in computer vision. The purpose of image registration is to find the optimal rigid perspective transformation matrix  $H$  to put the two images into the same coordinate. Assuming two images source image  $I_p$  and target image  $I_q$ , the coordinate transformation relationship between them is:

$$I_p = \text{warp}(H_{pq}, I_q) \quad (1)$$

Image keypoint detection and description is a key step for finding this connection. However, due to the changes in the environment when the images are captured and the large baseline case, it is not easy to find keypoints stably and accurately, and to generate robust keypoint descriptors.

### 3.2 Network Architecture

The model proposed in this paper takes an image as input and outputs the keypoint map and corresponding descriptors. As shown in Fig. 2.



**Figure 2:** Proposed network architecture, including VGG-like backbone feature encoder, keypoint detection module and keypoint description module. The layers of different colors in the backbone network represent the pooling layers and the convolutional layers, the spatial down-sampling is performed through three pooling layers. The proposed network takes a single image as input, and obtains multi-scale feature maps in a forward pass. These feature maps are restored to the original image resolution by bilinear up-sampling, and then the aliasing effect from up-sampling is reduced by  $3 \times 3$  convolution layers on each level [21]. The obtained features on different levels are aggregated by channel-wise concatenation. Then the network performs keypoint detection and description jointly

This model consists of three parts: VGG-like backbone feature encoder, keypoint detection module and keypoint description module. The proposed network up-samples the deep feature maps to the resolution of the input image, then combines the four layers of feature maps in the feature channel and feeds them into the keypoint detection and description module, thus sharing most of the computation of these two modules for efficient joint feature detection and description.

### 3.2.1 Backbone Feature Encoder

The VGG-like backbone feature encoder operates on full-size images. It contains 8 convolutional layers and 3 pooling layers, which executes three times space down-sampling. The steps of all convolutional layers are set to 1. The proposed multi-scale feature aggregation is performed in one forward pass. The input is  $I \in \mathbb{R}^{h \times w}$ , so the output features of Conv1, Conv3, Conv5 and Conv7 layers are  $P_i \in \mathbb{R}^{h_i \times w_i \times C_i}$ . Where  $C \in \{64, 64, 128, 128\}$ ,  $i \in \{1, 3, 5, 7\}$ ,  $h_i \in \left\{ H, \frac{H}{2}, \frac{H}{4}, \frac{H}{8} \right\}$  (the same for  $w_i$ ). Then, the feature maps of the four layers are all up-sampled by bilinear interpolation to the original image resolution, that is, the feature maps of the last three layers ( $i \in \{3, 5, 7\}$ ) are upsampled by 2 times, 4 times and 8 times respectively, and finally the feature maps of the four layers are obtained:  $h \times w \times 64, h \times w \times 64, h \times w \times 128$  and  $h \times w \times 128$ .

The feature extraction process can be formulated as:

$$P_i = \text{Enconer}(I), i \in \{1, 3, 5, 7\} \quad (2)$$

Considering that the shallow features are easily affected by noise, for these four feature maps from shallow to deep, they are multiplied by the corresponding weight: 0.1, 0.2, 0.3 and 0.4 respectively.

$$F_i = W_i P_i, W_i \in \{0.1, 0.2, 0.3, 0.4\} \quad (3)$$

### 3.2.2 Keypoint Detector

For keypoint detection, the feature maps of these four levels are respectively passed through a convolutional layer, to obtain four feature maps with a size of  $h \times w \times l$  pixels, and then added together to get the keypoint detection map  $M \in \mathbb{R}^{h \times w}$  and keypoint set  $A$ . And  $F$  is the multi-level feature maps.

$$F = \sum F_i, i \in \{1, 3, 5, 7\} \quad (4)$$

$$M, A = \text{Detector}(F), i \in \{1, 3, 5, 7\} \quad (5)$$

The condition for a point  $(i, j)$  on the detection map to be determined as a keypoint is that the value at the coordinates of the point is greater than the detection threshold  $\alpha = 0.5$ . As too dense keypoints are detrimental to the performance of the network, a non-maximum suppression (NMS) of size 3 is used to remove keypoints that are too close in space. The origin of the coordinates is the upper left corner of the keypoint map.

### 3.2.3 Keypoint Descriptor

For the feature description, the four layers of shallow-to-deep feature maps obtained by the backbone multi-scale feature extraction network are stacked together according to the channel dimension, and the size is  $h \times w \times 384$ . Finally, L2 regularization is performed on the description feature map in the channel dimension to construct a unit-length descriptor, and the final descriptor is composed of several 384-dimensional vectors located at the keypoint coordinates in the description feature map. For a keypoint  $(i, j)$  in  $M$ , its keypoint descriptor  $d_{i,j}$  consists of the value of the point in  $D$  located at  $(i, j)$  on the feature map of each layer.

$$D = \text{Stack}(F_i), i \in \{1, 3, 5, 7\} \quad (6)$$

$$d_{i,j} = \text{Descriptor}(D), d_{i,j} \in \mathbb{R}^{384} \quad (7)$$

## 3.3 Training and Loss

For each image  $I$ , this work generates a projection matrix  $H$ , and then uses this matrix to perform random viewing changes on  $I$ , and adds random photometric distortion to obtain  $I'$ . We use the method provided in [16], firstly train the keypoint detector on the synthetic dataset, and then use the detector to generate the keypoint coordinate labels of the training dataset. Let the pseudo-ground truth labels of  $I$  and  $I'$  are  $L$  and  $L'$ . Saving  $I, I', L, L'$  and  $H$  as a sequence. Inputting  $I$ , the keypoint detector outs the keypoint map  $M$  and the keypoint set is  $A$ . Then we calculate the keypoint detection loss  $L_{DI}$  for  $I$ .

$$L_{DI} = \frac{1}{h * w} \sum_{i,j}^{h,w} l_D(A_{ij}, L_{ij}, \lambda) \quad (8)$$

$$l_D = -\lambda L_{ij} \log(A_{ij}) - (1 - L_{ij}) \log(1 - A_{ij}) \quad (9)$$

where  $\lambda$  is to deal with the case where positive samples are far fewer than negative samples.

Then, inputting its corresponding transformed image  $I'$ ,  $L'$  is the keypoint set of  $I'$ , and calculate its keypoint detection loss.

$$L_{D'I'} = \frac{1}{h * w} \sum_{i,j}^{h,w} l_D(A'_{ij}, L'_{ij}, \lambda) \quad (10)$$

The detection loss  $L_d$  is the sum of  $L_{DI}$  and  $L_{D'I'}$ .

$$L_d = L_{DI} + L_{D'I'} \quad (11)$$

The descriptors represent the image content near the keypoints, so the descriptor of the keypoint located at the corresponding position in the two images should be the most similar, that is, the distance in the vector space between their descriptors is the closest, and vice versa is the farthest. We follow this idea to calculate the description loss. For two keypoints  $a$  in  $I$  and  $a'$  in  $I'$ , the corresponding descriptors are  $d$  and  $d'$ . The condition for these two points to be paired is:

$$\|warp(k, H) - k'\| \leq \varepsilon \quad (12)$$

Then the descriptor loss of  $a$ , which is  $l_{fa}$  can be described by the ternary loss between  $d$  and  $d'$ ,  $d'^*$ :

$$l_{fa} = \max(0, p(d, d') - n(d, d'^*)) \quad (13)$$

$p(d, d')$  and  $n(d, d'^*)$  represent the positive sample distance between paired  $d$  and  $d'$ , the negative sample distance between unpaired  $d$  and  $d'^*$  respectively. The description loss is calculated separately for all matching keypoint pairs in the two pictures. Firstly, all matching points are determined according to Eq. (12) (there can only be one matching point for each key point). Then we compute the description loss for each paired point in the two images. The total description loss is as follows:

$$L_f = \frac{1}{2N} \sum_{i=1}^N (l_{fa}) \quad (14)$$

The total loss of the network is the sum of  $L_d$  and  $L_f$ :

$$L_{all} = L_d + L_f \quad (15)$$

#### 4 Datasets and Implementation Details

This work uses the COCO2014 [22] dataset for training. The COCO dataset is collected from a variety of real-world scenarios. Training on this can enhance the generalization ability of the model.

It is difficult to collect ground truth tags for keypoint detection. This work uses the method proposed by [16] to generate pseudo-ground truth labels for keypoints. In addition, the detector is trained separately and then jointly with the descriptor. Firstly, a synthetic dataset containing lines and various basic shapes is constructed as the initial dataset to complete the pre-training of the detector. Then, the pre-trained detector is used to generate the pseudo-ground truth keypoint labels of each image in the COCO2014 training set. The detector is then trained on the training set with the pseudo-ground truth key labels, and the pseudo-ground truth labels are regenerated. This work repeats this process several times, until convergence, so as to achieve self-supervised training of the detector.

Finally, the joint training of detector and descriptor is completed on the training dataset of several iterations. This work also adds random brightness, contrast, saturation, blur and Gaussian noise to the images in the training set to enhance the robustness of the network.

The proposed network is trained for 100 K iterations, using Adam optimizer, the learning rate is set to  $10 \times 10^{-3}$ . Batch size is set to 4. The implementation is based on TensorFlow. Each input image is resized to  $480 \times 640$ .

## 5 Experiments

### 5.1 Image Matching on Hpatches

#### 5.1.1 Dataset

This paper compares the proposed network with other methods on the Hpatches [18] dataset. It is a popular dataset for evaluating local features, containing a total of 116 sequences. Each sequence contains 6 pictures, an original image, 5 changed images based on the original image, and 5 corresponding homography transformed matrixes. For a fair comparison, this work follows the setting of ASLFeat [17] and removes several high-resolution sequences from the comparison experiment. At last, there are 52 sequences with progressively illumination changes without view point changes. The rest of 56 sequences with large view point changes and without illumination changes. For each sequence, this work matches the original images against all other changed images, resulting in 540 pairs. Detailed comparison experiments on this dataset show the performance of the proposed network in the task of homography transformation estimation.

#### 5.1.2 Evaluation Protocols

Considering a standard image matching process, inputting two associated images  $I_s$  and  $I_f$  into a network. Then, it extracts and matches keypoints, calculates the perspective transformation matrix  $H_{fs}$  between these two images, and outputs the transformed  $I_f$ , which is  $I_f (I_f = \text{warp} (H_{fs}, I_f))$ . Thus, the image  $I_f$  under the  $I_s$  coordinate is obtained, which means these two images matching is realized. The feature matching algorithm used here is nearest neighbor algorithm for all descriptors. The estimation of  $H$  matrix is realized by OpenCV (*findHomography* ()). In addition, RANSAC is used to enhance estimation accuracy. In order to measure the performance comprehensively, the quantitative evaluation is compared with mean matching accuracy [23,24], keypoint repetition rate and matching score. Specific definitions are as follows:

1. Mean matching accuracy ( $MMA\%$ ). The ratio of the number of correctly matching keypoint pairs to the number of pre-matched keypoint pairs in the two images. As shown in the formula:

$$MMA = \frac{1}{N} \sum_{I_f, I_s} \frac{N_{\text{correct matches}}}{N_{\text{pre-matches}}} \quad (16)$$

The condition that a pair of keypoints in two images are judged to be pre-matched is that their descriptors are the closest to each other. On this basis, a hypothetical match is judged to be correct if the distance between a pair of keypoints is not greater than the error threshold  $\varepsilon$  (pixel level). The  $MMA$  measures the comprehensive performance of feature detector and descriptor.

2. Matching score ( $M.S.\%$ ). The ratio of the number of correct matches keypoint pairs to the sum of the number of keypoints in the visible area in both images.

$$M.S. = \frac{1}{N} \sum_{shared(I_f, I_s)} \frac{N_{correct\ matches}}{N(shared(I_f, I_s))} \quad (17)$$

The calculation method of correct matches here is consistent with the calculation method of correct matching in *MMA*, except that *MMA* performed on the entire image. The *M.S.* measures the comprehensive performance of feature detectors and descriptors.

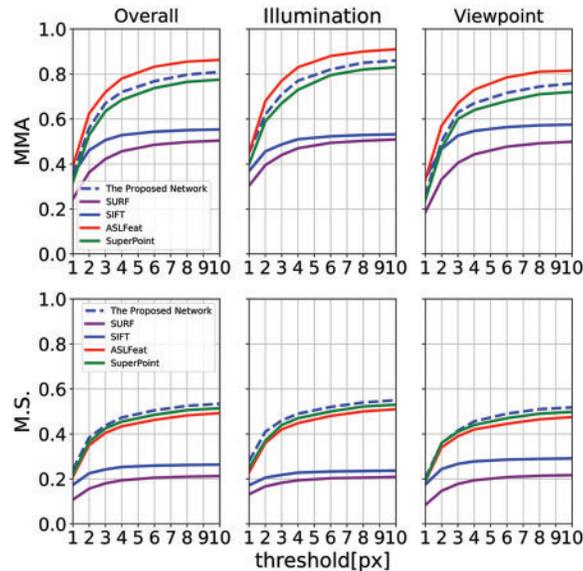
3. Keypoint repetition rate (*REP%*). The ratio of the number of possible matching keypoint pairs to the sum of the keypoints in the visible area in both pictures:

$$REP = \frac{1}{N} \sum_{shared(I_f, I_s)} \frac{N_{possible\ matches}}{N(shared(I_f, I_s))} \quad (18)$$

where, the  $N_{possible\ matches}$  is the number of keypoint pairs matched in  $I_s$  and  $I_f$ , and the condition for matching is that the distance between the point pairs is not greater than the error threshold  $\varepsilon$ . The *REP* represents the probability that a pair of feature points visible in both images is judged to be a match, thereby measuring the performance of the keypoint detector.

### 5.1.3 Results

The result is shown in Fig. 3 and Tab. 1. In Fig. 3, The abscissa is the pixel-level error threshold from 1 pixel to 10 pixels, and the two rows from top to bottom represent *MMA*, *M.S.* under different error thresholds.



**Figure 3:** The comparison of the proposed network with the classical methods SIFT, SURF and the learning-based networks ASLFeat and SuperPoint on mean matching accuracy (*MMA*) and matching score (*M.S.*) on HPatches. The second and the third column indicates the situation with illumination changes only and random perspective changes only respectively. And the first column indicates the average error value of these two columns. The proposed network has the best performance on *M.S.* and has comparable results with the latest learning-based methods on *MMA*

**Table 1:** Detector repeatability on HPatches

Method	Overall	Illumination	Viewpoint
SURF	49.7	47.6	51.8
SIFT	45.9	41.3	48.5
ASLFeat	57.8	57.1	58.6
SuperPoint	57.8	<b>65.2</b>	50.3
The Proposed	<b>59.1</b>	57.9	<b>60.2</b>

Because of the excellent network architecture that keypoint detector and descriptor reorganizes the low-resolution feature maps of multiple layers to the original image resolution, which improving the accuracy of keypoint location and generating descriptors with rich local structure information. The proposed network outperforms SuperPoint and traditional feature extraction algorithm on all metrics. ASLFeat has the best performance on the *MMA* metric. In addition, it is clear that learning-based methods are significantly better than traditional methods on *MMA* and *M.S*.

The Keypoint repetition rate (*REP%*) mainly measures the performance of the detector. To measure the detection ability of the proposed network for keypoints, this paper also compares with SIFT, SURF, SuperPoint and ASLFeat on *REP*. In order to ensure fairness, NMS is all set to 4, and the error threshold is 3 pixels. The results are shown in Tab. 1. The proposed network performs the best on the branch of viewpoint change, and the overall performance is also the best.

## 5.2 Homography Estimation on HPatches

The three metrics in Sec. 5.1 do not always reflect the geometric relationship between two images. For example, considering a case that a network detects only one keypoint with the same geometric content in each pair of images, and this pair of keypoints in the visible area in both pictures is judged to be a correct match. Then for this match, its *MMA*, *MS* and *REP* are all 1 (100%), which is actually meaningless, because the geometric relationship between these two pictures cannot be recovered by this match. For downstream tasks of image matching, such as SfM, it is usually necessary to find no less than 1000 pairs of precise matches between two images to robustly reconstruct 3D geometry. On the other hand, a relatively uniform distribution of keypoints is also important, but this is often difficult to measure directly [25,26]. Therefore, followings [16,27], this work uses H matrix estimation accuracy (*HA%*) to indirectly measure the accuracy and robustness of the network for finding connections between images.

### 5.2.1 Evaluation Protocols

It is not appropriate to directly calculate the similarity between the  $3 \times 3$  true *H* and the estimated *H'*, because different numbers in the *H* matrix have different scales. Referring to the method in [16], considering the four corner points in  $I_f$ ,  $\{k_{f1}, k_{f2}, k_{f3}, k_{f4}\}$ , this paper uses the true *H* matrix to project it under the  $I_s$  coordinate and obtain  $\{k_{s1}, k_{s2}, k_{s3}, k_{s4}\}$ , then uses the estimated homography transformation matrix  $H'$  [28] to get  $\{k_{s1'}, k_{s2'}, k_{s3'}, k_{s4'}\}$ , finally calculates the distance between these

four projected point pairs. If the distance is less than  $\varepsilon$ , it is judged that the estimate is correct. As shown in the formula:

$$HA = \frac{1}{N} \sum_{i=1}^N \left( \left( \frac{1}{4} \sum_{j=1}^4 \|k_{ij} - k'_{ij}\| \right) \leq \varepsilon \right) \quad (19)$$

This work compares the method proposed with the traditional feature detection and description algorithms SIFT, SURT, these two methods are implemented using OpenCV, and the latest learning-based methods: SuperPoint and ASLFeat. In order to ensure fairness, all methods are performed on the image size of  $480 \times 640$ . The feature matching algorithm using here is nearest neighbor algorithm. The H-matrix estimation is realized by OpenCV (*findHomography* ()). RANSAC is also used to enhance estimation accuracy. In addition,  $NMS = 4$ .

### 5.2.2 Comparisons with Other Methods

This work follows the practice of patch2pix [29] and reports the percentage of errors below 1, 3 and 5 pixels, as shown in Tab. 2. This work chooses the default settings of SURF and SIFT to apply them. For SuperPoint and ASLFeat, this work uses the settings reported in their respective papers. Firstly, similar to the results in Sec. 5.1, deep learning-based methods have made great progress over traditional methods. Secondly, the proposed method has the best performance on the branch of Viewpoint change and comprehensive performance, and it performs the best when the thresholds are 1 pixel and 3 pixels in both Illumination and Viewpoint changes, which shows that the proposed method improves the localization accuracy of keypoints. In particular, the improvement is obvious compared to SuperPoint, which again demonstrates the effectiveness of the proposed structure.

**Table 2:** Homograph estimation on Hpatches

Method	Overall			Illumination			Viewpoint		
	$\leq 1$ px	$\leq 3$ px	$\leq 5$ px	$\leq 1$ px	$\leq 3$ px	$\leq 5$ px	$\leq 1$ px	$\leq 3$ px	$\leq 5$ px
SURF	35.5	65.1	76.4	46.5	77.7	85.2	24.2	53.5	67.3
SIFT	37.6	67.8	77.5	48.8	79.1	85.5	27.7	56.7	69.6
SuperPoint	36.2	71.3	83.2	50.2	<b>88.6</b>	97.2	23.2	54.2	70.6
ASLFeat	39.9	71.2	84.5	56.3	87.6	<b>98.1</b>	23.5	54.6	71.3
The proposed	<b>41.4</b>	<b>73.4</b>	<b>84.6</b>	<b>57.4</b>	87.2	97.3	<b>25.4</b>	<b>59.8</b>	<b>71.8</b>

## 6 Conclusion

The quality and accuracy of image matching play a vital role in many computer vision tasks. In order to alleviate the problems of current feature extraction methods, this work designs a new convolutional neural network to jointly and intensively extract keypoints. On the one hand, the proposed network uses multi-level features to restore low-level details, thereby improving the positioning accuracy of keypoints. On the other hand, the proposed network utilizes multiple hierarchical features to generate robust keypoint descriptors. Detailed experiments on Hpatches have confirmed the superiority of the proposed network against traditional methods and the learning-based approaches.

**Funding Statement:** This work was supported by the National Natural Science Foundation of China (61871046, SM, <http://www.nsf.gov.cn/>).

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2013.
- [2] M. J. Westoby, "Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications," *Geomorphology*, vol. 179, no. 66, pp. 300–314, 2012.
- [3] Y. Zhang, J. Wang, S. Xu, X. Liu and X. Zhang, "Multi-level information fusion based deep local features," in *Proc. of Asian Conf. on Computer Vision (ACCV 2020)*, Virtual Kyoto, JPN, online, 2020. <https://accv2020.github.io/>.
- [4] E. Rublee, "ORB: An efficient alternative to SIFT or SURF," in *Proc. of IEEE Int. Conf. on Computer Vision*, Barcelona, Catalonia, Spain, pp. 2564–2571, 2011.
- [5] E. Tola, V. Lepetit and P. Fua, "DAISY: An efficient dense descriptor applied to wide-baseline stereo," *IEEE PAMI*, vol. 32, no. 5, pp. 815–830, 2010.
- [6] L. Xiangchun, C. Zhan, S. Wei, L. Fenglei and Y. Yanxing, "Data matching of solar images super-resolution based on deep learning," *Computers Materials & Continua*, vol. 68, no. 3, pp. 4017–4029, 2021.
- [7] K. M. Yi, E. Trulls and V. Lepetit, "LIFT: Learned invariant feature transform," in *Proc. of European Conf. on Computer Vision (ECCV 2016)*, Amsterdam, North Holland, Netherlands, pp. 467–483, 2016.
- [8] P. Huang and C. S. Guo, "Overview of image registration methods based on deep learning," *Journal of Hangzhou University of Electronic Science and Technology*, vol. 40, no. 6, pp. 37–44, 2020.
- [9] D. G. Lowe, "Distinctive image features from scale invariant keypoints," *International Journal of Computer Vision*, vol. 34, no. 60, pp. 91–110, 2004.
- [10] B. Herbert, T. Tuytelaars and L. V. Gool, "SURF: Speeded up robust features," in *Proc. of the European Conf. on Computer Vision (ECCV 2006)*, Graz, Steiermark, Austrian, pp. 404–417, 2006.
- [11] Z. G. Qu, H. R. Sun and M. Zheng, "An efficient quantum image steganography protocol based on improved EMD algorithm," *Quantum Information Processing*, vol. 20, no. 53, pp. 1–29, 2021.
- [12] Y. F. Wei, F. Richard Yu, M. Song and Z. Han, "User scheduling and resource allocation in HetNets with hybrid energy supply: An actor-critic reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 680–692, 2018.
- [13] Z. G. Qu, Y. M. Huang and M. Zheng, "A novel coherence-based quantum steganalysis protocol," *Quantum Information Processing*, vol. 19, no. 362, pp. 1–19, 2020.
- [14] S. Sun, J. Zhou, J. Wen, Y. Wei and X. Wang, "A dqn-based cache strategy for mobile edge networks," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 3277–3291, 2022.
- [15] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic *et al.*, "D2-Net: A trainable CNN for joint detection and description of local features," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR 2019)*, Long Beach, CA, USA, pp. 8092–8101, 2019.
- [16] D. Detone, T. Malisiewicz and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR 2018)*, Salt Lake City, Utah, USA, pp. 224–236, 2018.
- [17] Z. Luo, L. Zhou, X. Bai, H. Chen, H. Zhang *et al.*, "ASLFeat: Learning local features of accurate shape and localization," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR 2020)*, Seattle, WA, USA, pp. 6589, 2020.

- [18] V. Balntas, K. Lenc, A. Vedaldi and K. Mikolajczyk, "HPatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR 2017)*, Honolulu, Hawaii, USA, pp. 5173–5182, 2017.
- [19] Y. Tian, B. Fan and F. Wu, "L2-net: Deep learning of discriminative patch descriptor in Euclidean space," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR 2017)*, Honolulu, Hawaii, USA, pp. 661–669, 2017.
- [20] Y. Ono, E. Trulls and P. Fua, Lf-net: Learning local features from images. in *Proc. of the 32nd Int. Conf. on Neural Information Processing Systems (NeurIPS 2018)*, Montreal, Quebec, CAN, pp. 6234–6244, 2018.
- [21] D. Liu, Y. Cui, L. Yan, C. Mousas, B. Yang *et al.*, "DenserNet: Weakly supervised visual localization using multi-scale feature aggregation," in *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 35, pp. 6101–6109, 2021.
- [22] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona *et al.*, "Microsoft coco: Common objects in context," in *Proc. of European Conf. on Computer Vision (ECCV 2014)*, Zurich, Switzerland, pp. 740–755, 2014.
- [23] 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.
- [24] S. Mutic, J. F. Dempsey and W. R. Bosch, "Multimodality image registration quality assurance for conformal three-dimensional treatment planning," *International Journal of Radiation Oncology, Biology, Physics*, vol. 51, no. 1, pp. 255–260, 2001.
- [25] Z. Zhang and S. Zhou, "Research on feature extraction method of social network text," *Journal of New Media*, vol. 3, no. 2, pp. 73–80, 2021.
- [26] W. Sun, X. Chen, X. R. Zhang, G. Z. Dai, P. S. Chang *et al.*, "A multi-feature learning model with enhanced local attention for vehicle re-identification," *Computers Materials & Continua*, vol. 69, no. 3, pp. 3549–3561, 2021.
- [27] D. DeTone, T. Malisiewicz and A. Rabinovich, "Deep image homography estimation," 2016. [Online]. Available: <https://arxiv.org/pdf/1606.03798.pdf>.
- [28] P. E. Sarlin, D. DeTone, T. Malisiewicz and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR 2020)*, Seattle, WA, USA, pp. 4938–4947, 2020.
- [29] Q. Zhou, T. Sattler and L. Leal, "Patch2pix: Epipolar-guided pixel-level correspondences," *arXiv preprint*, arXiv: 2012.01909, 2021. <https://cvpr2021.thecvf.com/>.