A SELF-SUPERVISED KEYPOINT DETECTION NETWORK FOR MULTIMODAL REMOTE SENSING IMAGES

Liangzhi Li [a], Ling Han [b,*], Hongye Cao[a] and Ming Liu [b]

[a] College of Geological Engineering and Geomatics, Chang'an University, Xi'an 710064, China
[b]School of Land Engineering, Chang'an University, Xi'an 710064, China

Commission II, WG II/6

KEY WORDS: Keypoint detection, Self-supervision, Deep learning, SAR and optical image, DCN.

ABSTRACT:

Currently, multimodal remote sensing images have complex geometric and radiometric distortions, which are beyond the reach of classical hand-crafted feature-based matching. Although keypoint matching methods have been developed in recent decades, most manual and deep learning-based techniques cannot effectively extract highly repeatable keypoints. To address that, we design a Siamese network with self-supervised training to generate similar keypoint feature maps between multimodal images, and detect highly repeatable keypoints by computing local spatial- and channel-domain peaks of the feature maps. We exploit the confidence level of keypoints to enable the detection network to evaluate potential keypoints with end-to-end trainability. Unlike most trainable detectors, it does not require the generation of pseudo-ground truth points. In the experiments, the proposed method is evaluated using various SAR and optical images covering different scenes. The results prove its superior keypoint detection performance compared with current state-of-art matching methods based on keypoints.

1. INTRODUCTION

Advances in local feature detector research have led to significant improvements in areas such as remote sensing image matching, object recognition, photogrammetry, and 3D reconstruction. Therefore, various handcrafted keypoint detection methods have been developed, the most representative of which is the scale invariant feature transformation (SIFT) (Lowe, 2004), due to its keypoint feature being invariant under the translation, rotation and scale changes. Furthermore, it also includes these SIFT-like methods, such as SURF (Bay et al., 2008), Affine-SIFT (Morel and Yu, 2009).

However, customizing practical algorithms for processing remote sensing image keypoint detection remains a daunting task. Firstly, due to remote sensing images involve changes in ground features, radiation differences, and local distortions caused by the imaging viewpoints, this leads to complex spatial geometric relationships between image pairs. Therefore, the simple parametric models used in most existing handcrafted methods are no longer sufficient to produce repeatable keypoints. Additionally, there is no guarantee that the extracted features are repeatable in complex and variable remote sensing image.

In recent years, deep learning-based methods for keypoint detection have been hugely successful. The main reason is its completely data-driven scheme that tries to abstract the distribution structure from the input data. For example, SuperPoint (DeTone et al., 2018) learns keypoints by pixel supervision of artificial points. UnsuperPoint (Christiansen et al., 2019) uses a concatenated network to train keypoints end-to-end in an unsupervised manner, and adds non-maximal suppression to the model to make keypoints uniformly distributed. However, the lack of shape awareness of the feature points does not allow for stronger geometric invariance. R2D2 (Revaud et al., 2019) uses an expansive convolution strategy to maintain image size, increasing the computational burden, while the keypoints identified by the network's final detector are often at low levels of structure (corners, edges, etc.). LF-Net (Ono et al., 2018) extracts the features of the keypoints and transforms the intermediate features via a spatial converter, which requires multiple passes forward. This is only practically feasible for sparse shape parameter prediction. D2-Net (Dusmanu et al., 2019) produces a selection rule that derives keypoints from the same feature maps used to extract feature descriptors, avoiding the need to learn additional weights for the keypoint detector. However, these deep learning based keypoint detection methods give a promising direction in natural image processing. However, the working mechanism of these methods, which is to find the correspondence between pairs of images by processing them independently, would be difficult to apply to keypoint detection in multimodal remote sensing images with radiometric differences. Moreover, the detection of critical points is hampered by the fact that nonlinear radiometric differences between sar and optical images do not provide enough true correspondence.

To solve the above problem, we propose a self-supervised keypoint detection network for multimodal remote sensing images. The method does not require the use of ground truth keypoint locations as labels. Instead, we propose to learn the confidence level of keypoints by computing the local spatial- and channel-domain peaks of the output depth feature map, which is actually an estimate of the keypoint likelihood. Due to the nonlinear radiometric differences between multimodal images, the corresponding confidence values will be significantly different, which will reduce the repeatability of keypoints screened by confidence thresholds. Therefore, to improve the reproducibility of keypoints, we propose a Siamese detection network with self-supervised learning for training keypoints with the same confidence level. Also, the proposed detection network introduces a multi-scale feature extraction approach of deformable convolutional networks (DCN) (Dai et al., 2017) to improve the localization accuracy of keypoints and make it flexible to re-

* Corresponding author(E-mail address:hanling@chd.edu.cn)

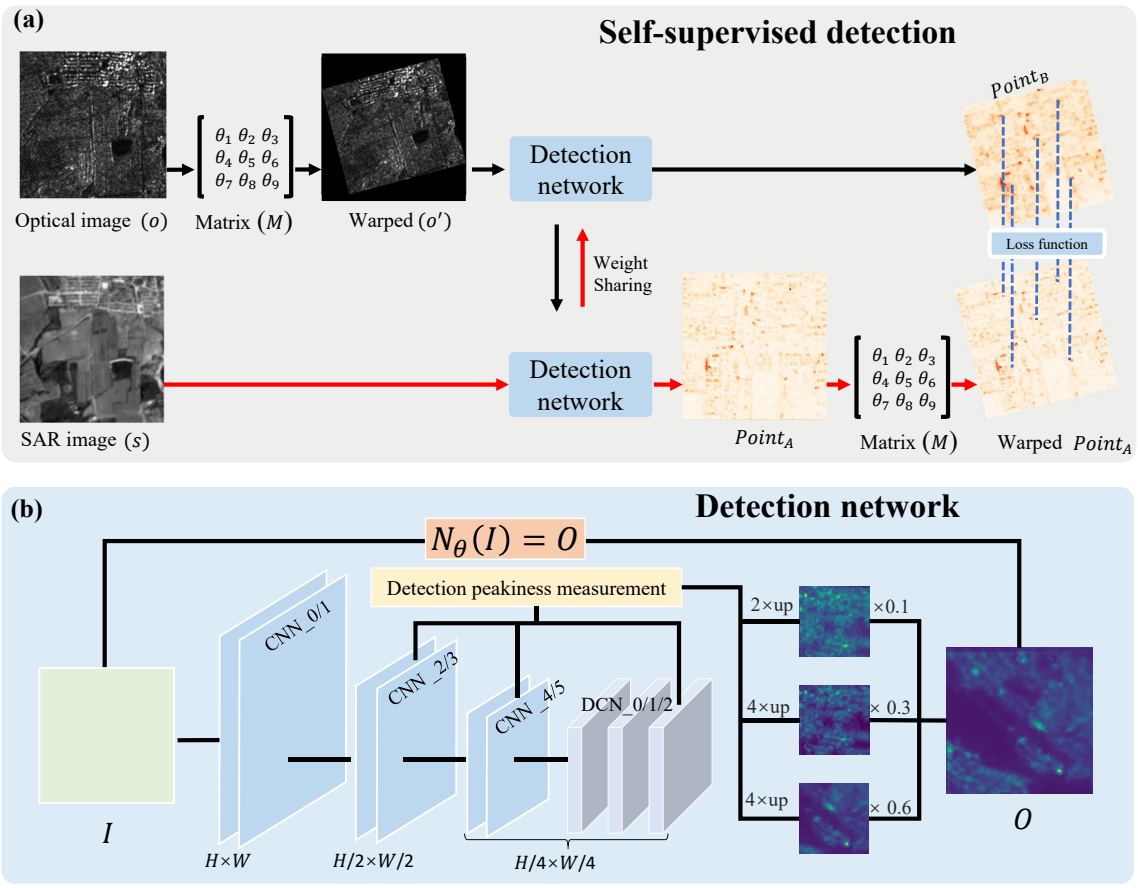


Figure 1. (a)Self-supervised detection. First, a randomly selected image from a strictly pixel-aligned remote sensing image is subjected to a projection transformation. Then, the warped image and the other image are simultaneously fed into the detection network with shared weights for feature point extraction. Finally, the losses of corresponding keypoints are calculated by the projection transformation relationship. (b)An overview of detection network. $N_\theta(I) = O$ represent the parameterization of detection network , and θ denotes parameters that the network needs to be trained. The peakiness measurement of keypoints are calculated from the local spatial(α_{ij}^c) and channel domains(β_{ij}^c)

solve geometric distortion between images

In short, the main contribution of this work is as follows. We proposed a self-supervised keypoint detection network. This network evaluates the confidence of keypoints by computing local spatial- and channel-domain peaks of feature maps. We design a Siamese detection network with self-supervised learning to optimize the confidence of keypoints simultaneously on multimodal remote sensing image pairs. Moreover, we introduce multi-scale and DCN operations for feature extraction. Therefore, the keypoint detection network is robust to geometric and radiometric differences between multimodal images, which can detect keypoints with high repeatability.

The rest of the paper is organized as follows. Section 2 introduces the related work of the keypoint detection with different methods. Section 3 presents our methods. Section 4 details the effectiveness of the detection network. We conclude in Section 5.

2. RELATED WORKS

In this section, we review the above four types of registration methods: intensity-based, feature-based, supervised learning-based, unsupervised learning methods.

2.1 Handcrafted Detectors

Traditional keypoint detection methods use handcrafted features to locate geometric structures, such as Harris (Derpanis, 2004) and Hessian detectors (Beaudet, 1978) use first- and second-order image derivatives to find corners or round points in an image. Extended Harris to handle multiscale and affine transformations, making the acquisition of keypoints invariant to scaling, rotation, and translation, and robust to illumination changes and limited viewpoint changes. SURF accelerates the process of detection by using integral images and approximations of the Hessian matrix. A multiscale improvement, called KAZE (Alcantarilla et al., 2012), is proposed in which the Hessian detector is applied to a nonlinear diffusion scale space. Affine-SIFT proved the affine invariance of the feature descriptions that obtained by varying the two camera axis orientation parameters (i.e., latitude and longitude angles) left by the SIFT method.

2.2 Learned Detectors

Data-driven learning-based methods have had a deeper impact on keypoint detection. We introduce a learning-based approach to detect reproducible keypoints under drastic imaging changes in weather and lighting conditions, through multiple binned linear regression models. (DeTone et al., 2017) proposed a point

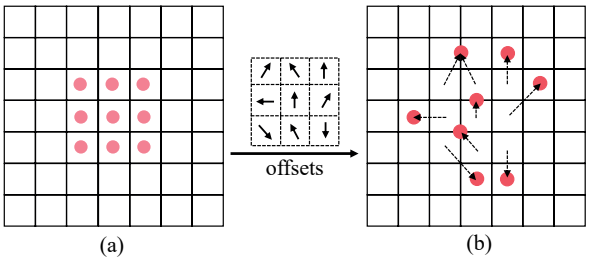


Figure 2. Deformable convolutional network. (a) is the common convolution with the 3x3 kernel, and (b) is the deformable convolutional network after the 3x3 convolution kernel with the offsets.

tracking system driven by two deep convolutional neural networks (MagicPoint and MagicWarp), where MagicPoint operated on a single image and extracted significant 2D points and MagicWarp operated on pairs of point images to estimate the associated single-response. MagicPoint was extended to SuperPoint, which includes a salient detector and descriptor. (Revaud et al., 2019) proposed a joint learning of keypoint detection and description and local descriptor discriminative predictors while outputting sparse, repeatable and reliable keypoints. LIFT(Yi et al., 2016) implemented an end-to-end feature detection and description pipeline including direction estimation for each feature. (Savinov et al., 2017) proposes to train a neural network in a transform invariant manner and rank points, and then extract points of interest from the top/bottom fractions of that ranking. LF-Net (Ono et al., 2018) estimated the location, scale, and orientation of features by jointly optimizing the detector and descriptor.

3. METHODS

3.1 Self-supervised detection

A detailed description of proposed remote sensing image keypoint detection method is shown in Figure 1. Figure. 1(a) shows the self-supervised detection process of the detection network, where the first image is randomly selected from the strictly pixel-aligned remote sensing images for projection transformation (the matrix M is randomly generated). The projection transformation is implemented on the optical image to obtain o'. Let o' and s be simultaneously input into a Siamese network (two detection networks, where their weights are shared) to predict the keypoint feature maps ($Point_A$, $Point_B$) using self-supervised training. Finally, $Point_B$ is warped using M to make it spatially consistent with $Point_A$. Specifically, the aim is to make them close in spatial distances, generating keypoints with the same location. We can determine the true transformation relationships of the above image (o', s) pairs. Moreover, we use the Huber loss function to compute the loss between their corresponding points. The Huber loss for $Point_A$ and $Point_B$ is detailed as follows.

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & for|y - f(x)| \le \delta \\ \delta|y - f(x)| - \frac{1}{2}\delta^2 & otherwise \end{cases} \tag{1}$$

where δ is the optional hyperparameter, $f(x)$ the predicted value, and y is the ground truth value.

Figure 1(b) depicts the pipeline for keypoint detection network. Let I be the remote sensing images. We focus our work on the number of bands as 1, but emphasize that the network structure does not depend on the number of bands for the input image. We introduce DCN operations to enhance the geometric robustness of keypoints in the decoding stage. $N_\theta(I) = O$ is used to describe the mapping relationship from the image to output results, where θ is the network parameter that needs to be trained on, and O is the output. The peakiness measurement of keypoints are calculated from the local spatial(α_{ij}^c) and channel domains(β_{ij}^c) respectively, additionally using $Softplus$ to activate the peaks to positive values. The detailed calculation procedure is described in this subsection 3.2.

3.2 Detection network architecture

The detection network is designed to generate feature maps, and extract the keypoints. The model is composed of different neural networks, including CNN and DCN.

Network architecture. The detection network performs the keypoint detection through the output feature maps. The architecture of the detection network is shown in Figure 1. In our experiments, we use images with a size of 240 $\times$ 240 pixels as input. The convolutional kernels of the CNN are all of size 3×3 pixels, and each convolutional layer is followed by ReLU (Agarap, 2018) and BN (Ioffe and Szegedy, 2015) layer. The input images are first passed through CNN-0/1, then through CNN-2/3, CNN-4/5 to reduce the size of the feature map, where the number of their feature maps are 32/64, 64/64, and 128/128, respectively. Finally, the output of CNN-5 is input to DCN-0/1/2 for generating feature maps with geometric invariance.

Deformable convolutional network. Figure 2 depicts the structurt of the DCN. The capability of feature extraction is enhanced by inserting offset (deformable convolution) in the convolution layer (Dai et al., 2017), which serves to enable the network to learn the dynamic sensory domain when extracting features to adapt to model geometric changes and can better adapt to the deformation of regional objects. In the conventional CNN operation, the input feature map is sampled using a regular grid R. For each position p_0 on the output feature map can be calculated by Equation 1.

$$y\left(p_0\right) = \sum_{p_n \in R} w\left(p_n\right) \cdot x\left(p_0 + p_n\right) \tag{2}$$

where p_n is an enumeration of the positions listed in R. In the deformable convolution operation, the regular grid R is expanded by adding an offset, and the same position p_0 becomes:

$$y\left(p_0\right) = \sum_{p_n \in R} w\left(p_n\right) \cdot x\left(p_0 + p_n + \Delta p_n\right) \tag{3}$$

Since the offsets Δp_n are usually fractional, they are implemented by bilinear interpolation, and the number of channels we use in experiments is 128. The rest of the parameter settings are implemented in reference (Dai et al., 2017).

Keypoint detection. To obtain keypoints that are robust to scaling changes, we use three stages of feature maps in the network CNN-3/5 and DCN-2 for keypoint detection. Subsequently, the outputs in the three stages are input to the upsampling network,

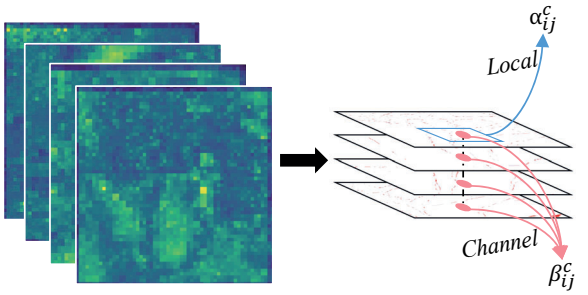


Figure 3. Keypoint detection. The keypoints are determined from the peaks in the local spatial- and channel domains. For each position (i, j) and channel $(c = 1, 2, ..., c)$ in the feature map output by the detection network, the local spatial(α_{ij}^c) and channel (β_{ij}^c) scores are calculated to generate the keypoint confidence.

which are used to recover the original size and assign the corresponding weights. Keypoints are determined from the peaks in the local spatial- and channel domains, as shown in Figure 3. Specifically, for each position (i, j) and channel $(c = 1, 2, ..., c)$ in the feature map output by the detection network, the local spatial(α_{ij}^c) and channel (β_{ij}^c) scores are calculated by:

$$\beta_{ij}^c = \text{softplus}\left(y_{ij}^c - \frac{1}{c}\sum_t y_{ij}^t\right) \tag{4}$$

where c is the feature map in the channel domain, y_{ij}^c is a value on the feature map (i, j). The activation function (Softplus) (Zheng et al., 2015) serves to activate the keypoint feature map to a positive value.

$$\alpha_{ij}^c = \text{softplus}\left(\mathbf{y}_{ij}^c - \frac{1}{|\mathcal{N}(i,j)|}\sum_{(i',j')\in\mathcal{N}(i,j)} y_{i'j'}^c\right) \tag{5}$$

where $N(i, j)$ is the set of 9 neighbors of pixel (i, j) and also its own pixel value.

To consider these two criteria, we maximize the product of two scores in all feature maps c to obtain a single score map. The calculation formula is as follows:

$$\gamma_{ij} = \max_c\left(\alpha_{ij}^c\beta_{ij}^c\right) \tag{6}$$

CNN-3/5 and DCN-2 are rehabilitated to their original sizes by upsampling for obtaining undistorted features on-scale, called $\gamma_1, \gamma_2, \gamma_3$, respectively. For those multiple scale features, they are not assigned the same weight. This is because we have considered the abstraction of these features from low-level to high-level features. The final keypoint feature map O is computed as follows:

In CNN-3, CNN-5 and DCN-2 after keypoint detection generated $\gamma_1, \gamma_2, \gamma_3$, and then assigned different weights to $\gamma_1, \gamma_2, \gamma_3$. The final keypoint feature map O calculation process is as follows:

$$O = \Delta_1\gamma_1 + \Delta_2\gamma_2 + \Delta_3\gamma_3 \tag{7}$$

where Δ_1, Δ_2 and Δ_3 are weights, and $\Delta_1 + \Delta_2 + \Delta_3 = 1$.

3.3 Implementation Details

During the training process of the detection network, we use a self-supervised approach to map images to keypoint feature maps and globally optimize the network parameters using Huber loss function to obtain keypoints with stable locations. To obtain a more robust detection model, our training samples are not limited to one sensor image, Conversely, we obtain image training models with different remote sensing sensors for different scenes. During the training process, different weights are assigned to $\Delta_1, \Delta_2, \Delta_3$, where the weight combinations include 0, 0.1, 0.3, 0.6, and 1.0.

The network is trained to search for the optimal weight parameter combinations with the same other parameters. The larger value in the keypoint feature map indicates the higher the degree that the point is critical. The model was trained using Adam with an initial learning rate of 1×10^{-3} and a batch size of 16, on the RTX 2070 GPU.

4. EXPERIMENT

In this section, we first introduce the dataset used to train the networks in subsection 4.1. Then, we describe the details of evaluation metrics in subsection 4.2. In subsections 4.3 and 4.4, an ablation study is carried out to compare the performance between the network combination and scale weight parameters set in the detection network. In subsection 4.5, the overall performance of our proposed method in the multimodal image is evaluated.

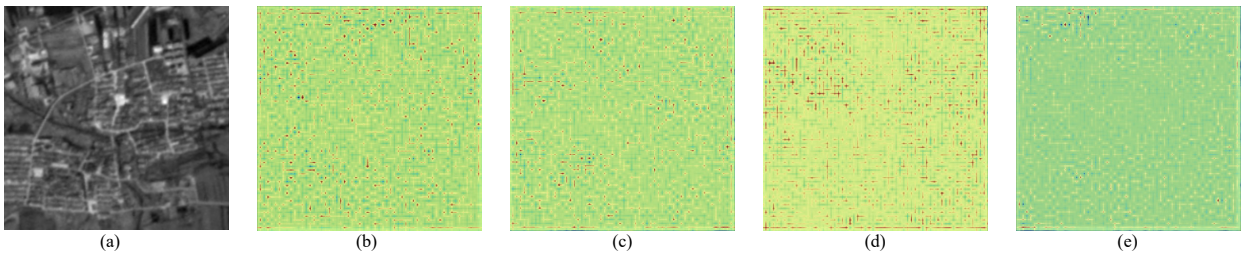
4.1 Dataset

To train the network model, 10 pairs of optical and SAR images from the world-wide region are acquired for the generation of the dataset, Sliding image crop size of 240 × 240, total 357,000 pairs of sar and optical images The optical satellite sensor is SkySat, and the RGB set of this data contains images with three sharpened 8-bit bands with a spatial resolution of 0.8 m. The SAR images were acquired by the Sentinel-1, and this data contains all GRD scenes. Each scene has three resolutions (10, 25 or 40 m) and four band combinations (corresponding to the scene polarization). For the experiments, we used a combination of $VV + VH$ polarizations. These images include urban, port, suburban, and rural scenes with a total coverage of about 3000 km^2.

These data are aligned at the pixel level by manually selecting hundreds of matching points in each image pair on the basis of geographic matching. The training, validation and test set used to train the network are cropped from the above matched image pairs. In our experiments, we mix data from all scenes, ensuring that the ratio of data from each scene in the training, validation and test datasets are all $0.7/0.2/0.1$, fed into the same model for optimal training.

4.2 Evaluation metrics

We evaluate our method in the dataset mentioned above. This dataset contains different scenes, where the optical and SAR data are used as the reference and sensed images. The performance of the detection network and the cross-fusion matching network is evaluated using the following evaluation protocol.



(a) (b) (c) (d) (e)

Figure 4. Example of keypoint feature maps generated from different weight combinations. (a) is the original remote sensing image. (b), (c), (d), (e) are keypoint feature maps result with weight combinations C_3, C_4, C_7, C_9, respectively.

Table 1. Evaluation result of repeatability performance by different network combinations.

Network	CNN	Multiscale	DCN	Repeatability
NetBase	Y	N	N	0.425
NetMS	Y	Y	N	0.558
NetDCN	Y	Y	Y	0.642

Repeatability. The pixels with the same position between two images are recognized as a keypoint pair, indicating that the keypoint pairs at this position is reliable. We use the repeatability (n/N) to evaluate the performance of the detection network, where n, N are the number of repeatable and all keypoints obtained.

The keypoint is identified at the same location in same scene image, indicating that the keypoint at that location is repetitive. In addition, we use the number of keypoints to evaluate the performance of the detection network.

4.3 Ablation study

To aid in the design of the detection network described in Section 3, we performed an ablation study to compare the performance of networks with various architectures added. We tested variants of the two-stage network of the detection network, which were detailed in Table 1. The two components were replaced with the original convolution. We generated test datasets by cropping the data using the images presented above, where the pixel value size of both optical and SAR images was 240 × 240. We evaluated the performance of the keypoint repeatability.

From Table 1, the addition of DCN obtained a significant reduction in repeatability. Therefore, DCN was used as our selection framework for SAR and optical image matching, and all further experiments were conducted with this framework.

4.4 Combination of scale weighting parameters

Different combinations of scale weight parameters might produce the same final loss in the training process of the detection network. Nevertheless, the loss function was designed to optimize the accuracy of keypoint matching overall and does not limit the number of matching points, allowing the network to determine whether certain locations were keypoints based on the input image. More specifically any location in the input image could theoretically be a matching point. Therefore, the number of keypoints that can be matched might vary. In experiments, we analyzed the performance of the network in detecting keypoints by changing the parameter combinations of scale weights. The specific parameter combinations were shown in Table 2, with the CNN-3, CNN-5, and DCN-2 weights varying between 0 and 1 and the sum of the weights being 1. We conducted experiments using the training data mentioned in the previous section. The parameter of the epoch with the same are trained under each set of weight combinations while keeping the other parameters consistent. We set the value of the keypoint confidence greater than 0.6 as the criterion for judging the keypoints.

Table 2 listed all the weight combinations. The best experimental results were marked in bold with the highest number of keypoints at the combinations $(0.1, 0.3, 0.6)$. The reason for not obtaining a higher number of repeatability keypoints under a single scale minght be that increasing the scales at different levels enhances the network's ability to obtain global information with higher robustness. Under the combination of multiple scales, the higher weight is given to DCN-2, the higher the number of keypoints obtained. Table 2 showed the probability distribution of keypoint values under different combinations, which was a statistical analysis made on test data. The values of these keypoints were distributed between 0.0 and 1.0, where N', μ denoted the average number of keypoints, mean value of repeatability.

Figure. 5 provides additional insights about the keypoints generated with different weight combinations. To perform this evaluation, we select the keypoint feature maps generated by the combination of C_3, C_4, C_7, C_9 in Table 2 (number of keypoint matches $C_3 > C_4 > C_7 > C_9$) for comparing the keypoint feature maps under the same scene image. The distribution shows that keypoint feature maps with local maxima (local peaks) can be generated with all different weight combinations; some combinations generate better ones, while some the opposite. And from the comparison of C_3 and C_9, they show that the heat map peaks of C_3 combination are more discrete and each one of them has better local smoothness. On the other hand, the corresponding heat maps under the combination of C_4 and C_7 have irregular shapes and peaks that surround each other, lacking more discrete and concentrated keypoints compared to C_3.

4.5 Reproducibility of feature detection

In the feature detection process, the ability of the detection network to extract keypoints that appear at the same location on the image was critical for image matching. As a consequence, we evaluated the performance of detection networks to obtain repeatable keypoints using existing methods. For this purpose, we compared it with state-of-the-art techniques: SIFT, SURF, Affine-SIFT, and SuperPoint. Six pairs with different scene types were selected for testing, including: urban (I_1), suburban (I_2), industrial (I_3), pond (I_4), port (I_5), and mountain (I_6), as shown in Figure 6, where each pair of images was pixel-aligned, and with a size of 240 × 240. Nevertheless, to give

Table 2. Weight combination and result, where N, μ denote the average number of matching points and mean value of REME, respectively.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
	$(0,0,1)$	$(0,1,0)$	$(0.1,0.3,0.6)$	$(0.1,0.6,0.3)$	$(0.3,0.1,0.6)$	$(0.3,0.6,0.1)$	$(0.6,0.1,0.3)$	$(0.6,0.3,0.1)$	$(1,0,0)$
N	430	450	**498**	457	477	422	449	435	413
μ	0.559	0.497	**0.653**	0.624	0.571	0.611	0.607	0.593	0.515

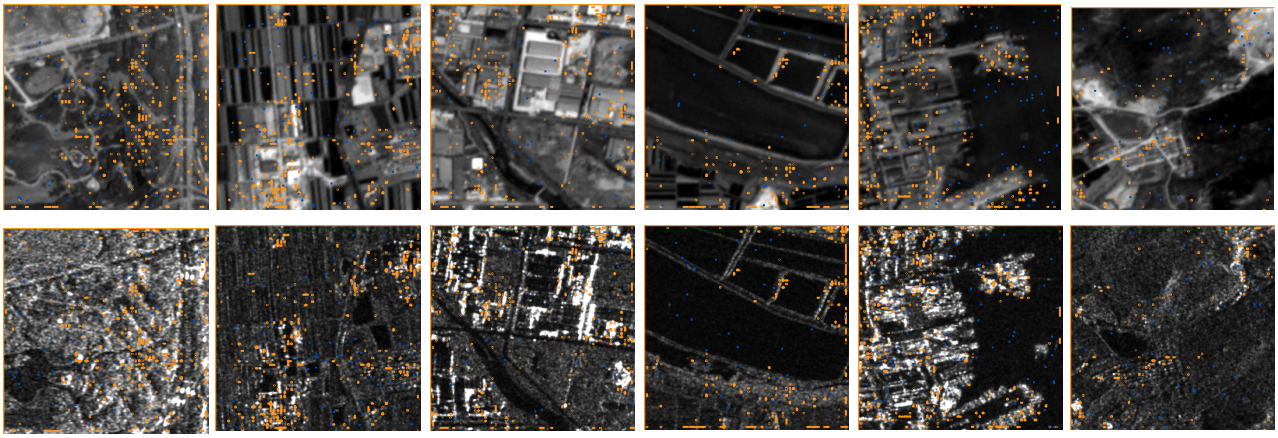


Figure 5. Keypoint detection results. (a), (b), (c), (d), (e) and (f) are six pairs of images covered with different scenes including: urban, suburban, industrial, pond, port, and mountain, where the first row are optical image and the second row are SAR images. Each pair of images is pixel-aligned and has a size of 240 × 240, where the yellow points indicate repeatable keypoints, and conversely blue points indicate non-repeatability.

Table 3. The result of keypoint repeatability. I_1, I_2, I_3, I_4, I_5, I_6 are urban, suburban, industrial, pond, port and mountain scenes respectively, where ASIFT is the abbreviation of Affine-SIFT.

Two pixels threshold					
I	**SIFT**	**SURF**	**ASIFT**	**SuperPoint**	**Proposed**
I_1	0.327	0.271	0.215	0.525	**0.566**
I_2	0.358	0.256	0.141	0.546	**0.547**
I_3	0.297	0.300	0.106	0.528	**0.608**
I_4	0.366	0.207	0.252	0.506	**0.511**
I_5	0.301	0.226	0.216	0.510	**0.610**
I_6	0.285	0.159	0.107	0.434	**0.556**

insight in the detection network' performance and especially to enable obtain repeatable keypoints at the same location in the same scene, we applied the number of repetitions of the detected matches to evaluate the network. In the experiments, the keypoint feature maps generated by our proposed method were arranged in the order from largest to smallest, and we selected the keypoints within the two pixel error thresholds as the final detection results.

Table 3 gave an overview results of detection for the keypoint repeatability. It showed that our proposed detection network obtained the highest keypoint repeatability compared to the benchmark method. SuperPoint and the proposed method were higher than the hand-designed method, indicating the effectiveness of the learning-based keypoint detection method. On I_1, I_2,I_3, I_4, I_3, I_4, the repeatability of our proposed method was higher than SuperPoint due to the fact that our proposed method considered both SAR and optical image differences in a self-supervised manner, which would facilitate cross-modal keypoint detection. Figure 6 (a), (b), (c), (d) (e) and (f) showed the keypoint detection maps of our proposed method. Overall, the keypoint locations were basically concentrated in regions with richer textures. Fewer keypoints were obtained on the homogeneous region in Figure 6 (e). This illustrated that our proposed method could suppress the generation of keypoints in homogeneous regions, which would improve the robustness of the keypoint description. However, some keypoints were generated in the edge regions of the image, which might affect the keypoint characterization. Therefore, in the next study, we would further investigate the removal of edge keypoints.

5. CONCLUSION

In this paper, we propose self-supervised keypoint detection networks for remote sensing image, which are shown to be less sensitive to radiometric differences across modalities while still providing repeatable keypoints. The trainable detection network is combined to form a Siamese network for self-supervised training to optimize the overall network parameters. We conducted a series of thorough experiments to obtain an optimal weight combination approach for keypoint detection.

We demonstrate that our trainable detection network is able to obtain a higher number of reproducible keypoints in images with nonlinear radiometric differences compared to existing keypoint detection methods. In addition, we test the detection network on multimodal images with different scenes, and our method obtains a high number of cross-modal keypoints for all scene images, which proves the effectiveness of our proposed method.

REFERENCES

Agarap, A. F., 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Alcantarilla, P. F., Bartoli, A., Davison, A. J., 2012. Kaze features. *European conference on computer vision*, Springer, 214–227.

Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., 2008. Speeded-up robust features (SURF). *Computer vision and image understanding*, 110(3), 346–359.

Beaudet, P. R., 1978. Rotationally invariant image operators. *Proc. 4th Int. Joint Conf. Pattern Recog, Tokyo, Japan, 1978*.

Christiansen, P. H., Kragh, M. F., Brodskiy, Y., Karstoft, H., 2019. Unsuperpoint: End-to-end unsupervised interest point detector and descriptor. *arXiv preprint arXiv:1907.04011*.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y., 2017. Deformable convolutional networks. *Proceedings of the IEEE international conference on computer vision*, 764–773.

Derpanis, K. G., 2004. The harris corner detector. *York University*, 2.

DeTone, D., Malisiewicz, T., Rabinovich, A., 2017. Toward geometric deep slam. *arXiv preprint arXiv:1707.07410*.

DeTone, D., Malisiewicz, T., Rabinovich, A., 2018. Superpoint: Self-supervised interest point detection and description. *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 224–236.

Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T., 2019. D2-net: A trainable cnn for joint description and detection of local features. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 8092–8101.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning*, PMLR, 448–456.

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91–110.

Morel, J.-M., Yu, G., 2009. ASIFT: A new framework for fully affine invariant image comparison. *SIAM journal on imaging sciences*, 2(2), 438–469.

Ono, Y., Trulls, E., Fua, P., Yi, K. M., 2018. LF-Net: Learning local features from images. *arXiv preprint arXiv:1805.09662*.

Revaud, J., Weinzaepfel, P., De Souza, C., Pion, N., Csurka, G., Cabon, Y., Humenberger, M., 2019. R2D2: repeatable and reliable detector and descriptor. *arXiv preprint arXiv:1906.06195*.

Savinov, N., Seki, A., Ladicky, L., Sattler, T., Pollefeys, M., 2017. Quad-networks: unsupervised learning to rank for interest point detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1822–1830.

Yi, K. M., Trulls, E., Lepetit, V., Fua, P., 2016. Lift: Learned invariant feature transform. *European conference on computer vision*, Springer, 467–483.

Yu, G., Morel, J.-M., 2011. ASIFT: An algorithm for fully affine invariant comparison. *Image Processing On Line*, 1, 11–38.

Zheng, H., Yang, Z., Liu, W., Liang, J., Li, Y., 2015. Improving deep neural networks using softplus units. *2015 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 1–4.