Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8927822, 13 pages
https://doi.org/10.1155/2021/8927822

Research Article

A Robust Invariant Local Feature Matching Method for Changing Scenes

Di Wang, Hongying Zhang, and Yanhua Shao

School of Information Engineering, Southwest University of Science and Technology, 621010 Mianyang, China

Correspondence should be addressed to Hongying Zhang; zhywyd@163.com

Received 14 October 2021; Accepted 6 December 2021; Published 28 December 2021

Academic Editor: Ming Yan

The precise evaluation of camera position and orientation is a momentous procedure of most machine vision tasks, especially visual localization. Aiming at the shortcomings of local features of dealing with changing scenes and the problem of realizing a robust end-to-end network that worked from feature detection to matching, an invariant local feature matching method for changing scene image pairs is proposed, which is a network that integrates feature detection, descriptor constitution, and feature matching. In the feature point detection and descriptor construction stage, joint training is carried out based on a neural network. In the feature point extraction and descriptor construction stage, joint training is carried out based on a neural network. To obtain local features with solid robustness to viewpoint and illumination changes, the Vector of Locally Aggregated Descriptors based on Neural Network (NetVLAD) module is introduced to compute the degree of correlation of description vectors from one image to another counterpart. Then, to enhance the relationship between relevant local features of image pairs, the attentional graph neural network (AGNN) is introduced, and the Sinkhorn algorithm is used to match them; finally, the local feature matching results between image pairs are output. The experimental results show that, compared with the existed algorithms, the proposed method enhances the robustness of local features of varying sights, performs better in terms of homography estimation, matching precision, and recall, and when meeting the requirements of the visual localization system to the environment, the end-to-end network tasks can be realized.

1. Introduction

The excellent matching performance between local features of changing scene images can ensure the stability of adjacent frame matching in visual localization. Visual localization is also the key technology for the robot to realize autonomous movement. Moreover, visual localization is an essential technology in route planning. Route planning has been applied in many fields [1]. The determination of camera pose has always been the focus in the field of machine vision. It can ensure the better operation of follow-up work, such as tracking and loop closure detection. Feature detection, descriptor constitution, and local feature matching are necessary supporting means to deal with most machine vision problems. At present, standard traditional and improved feature extraction methods [2–5] have been widely used in the visual localization system, the Markov model [6] has also been widely used in the traditional route planning issues. Scale-invariant feature transform (SIFT) [2] constructs the Gaussian pyramid of the input image, queries the extreme points, and determines the feature points' position. Finally, it calculates the domain gradient with the feature points as the circle's center and gives the SIFT feature point direction to obtain the descriptor. Speed-Up Robust Features (SURF) [4] improves the SIFT algorithm, converts the input image into an integral image, transforms the image with Hessian matrix, and finally extracts local feature points through nonmaximum suppression to solve the problems of high computational complexity and time-costs of SIFT. As a feature detection algorithm, the document [3] as well as the descriptor construction algorithm proposed in document [5] is usually used together. Compared with SIFT and SURF, it has a faster speed. Oriented Fast and Rotated Brief (ORB) [7] improves FAST and BRIEF algorithms. Compared with FAST, this algorithm only uses fewer pixel values in feature point extraction and solves its disadvantage of

nondirectionality. Compared with BRIEF, ORB uses a statistical learning method to select the set of feature points, which strengthens the discrimination ability but reduces the computational efficiency. The research on feature point detection and descriptor construction mainly focuses on finding a unique location that can reliably estimate its scale, rotation, and illumination change. Feature points are usually the premise of processing machine vision tasks. However, even if conventional feature detection ways, such as SIFT, SURF, and other algorithms, have been widely used in the field of machine vision, they cannot reflect semantic features and lack strong robustness to scene viewpoint, scale, and weather illumination changes. The introduction of the convolutional neural network can effectively solve different computer tasks [8]. Computers benefit from the introduction of deep learning and can obtain robust models against changes in the natural environment under the training of large amounts of data.

It is very important to describe the attitude transformation between adjacent frames. DeTone et al. [9] applied depth learning to feature point detection and transformation estimation and proposed a paper on using depth neural network to obtain homography matrix. The Learned Invariant Feature Transform (LIFT) algorithm proposed by Trulls et al. [10] uses the spatial deformation combiner to combine the feature extraction algorithm [11], feature direction algorithm [12], and descriptor construction algorithm [13] based on neural network. The LIFT leads a fresh deep network structure into the algorithm, but it suffers the defect of carrying out an end-to-end learning network. DeTone et al. [14] put forward the Superpoint, an algorithm advanced on [15], which uses the Siamese neural network to train the joint model of local features, to complete the feature task by a single neural network. The descriptor constitution phase of the united training network uses the network architecture proposed in [16], to promote the performance of the feature point detector, and Superpoint also puts forward homographic adaptation, which is mainly responsible for operating on a single image and extracting two-dimensional feature points. Compared with the effect of the traditional feature point detector, it has better detection performance in the case of image noise. To improve the generalization and robustness of the extracted feature points, TGD is combined with a depth neural network to detect the confidence score of each extracted feature point to screen the wrongly extracted feature points through the confidence score. The descriptors built by Superpoint cannot adaptively adjust the invariance for changing scenes, to strengthen the robustness of local features to natural change scenes, and Pautrat et al. [17] proposed to use NetVLAD module [18] to integrate and compute the output of the local feature construction stage. NetVLAD is suitable for weakly supervised CNN architecture of location recognition. Its introduction can make the descriptor construction stage have strong compatibility combined with most feature point extraction algorithms. A good feature matching algorithm can reduce much computational burden for the follow-up work of machine vision. To strengthen the linkage of local correlative features from an image pair to another counterpart, Sarlin et al. [19] put forward a matching method being founded on Attention Graph Neural Network (AGNN) [20, 21].

The above algorithms are separately used in three single stages: feature detection, descriptor constitution, and local feature matching. However, for visual localization, it is essential to propose an end-to-end network that integrates phases of local feature detection and matching to facilitate the later realization of good machine vision work, such as visual odometry and loop closure detection. Being founded on this, in this work, a robust invariant local feature matching method for changing scenes is put forward, which unites the invariant local feature matching method of AGNN and the Sinkhorn algorithm [22]. Furthermore, this paper proposes a united network of feature detection and descriptor constitution. Due to the lack of datasets of unique work, self-supervised learning can effectively solve this problem [23]; therefore, this paper uses the method of self-supervised training network in the feature location extraction stage. The NetVLAD module is employed to compute and fuse the correlation of multiple descriptors after homography transformation, which makes the local features output in the later stage have robust illumination invariance and viewpoint invariance. AGNN plays a role in strengthening the relevance of local features in the matching stage, which is enhanced by iterative updating, and the added local features are the output of the above joint training network. Finally, the local features linkage strengthened by AGNN is solved by the Sinkhorn algorithm. Meanwhile, to ensure the robustness of the model to natural scenes viewpoint and illumination varieties, a completed end-to-end network is applied to construct and go together local features.

The main work of this paper includes the following aspects: (1) an end-to-end matching network that integrates local feature constitution and matching is put forward. (2) In the descriptor constitution phase, the NetVLAD module is introduced to enhance the descriptor performance. (3) In the matching stage, AGNN and Sinkhorn algorithms are used to gain the optimal matching matrix. In addition, the algorithm proposed in this article has a better performance than the existing algorithm in the public dataset test.

2. Invariant Local Features Matching Network Architecture

2.1. Network Framework. This work takes the integral channel image pair as input and the direct matching result as the output, which describes a single end-to-end local feature matching method based on a deep neural network. The whole system integrates feature detection, descriptor constitution, and local feature matching.

The fundamental network architecture is indicated in Figure 1, which contains a total of three phases. The initial stage is calibrating the natural scene dataset, and the second stage is the united training network of local features. To begin with, an elementary feature point extractor is achieved, which is trained by a labeled dataset; this dataset is obtained by the joint action of simple shape dataset, real scene dataset, and pretraining feature extractor. Secondly, the united

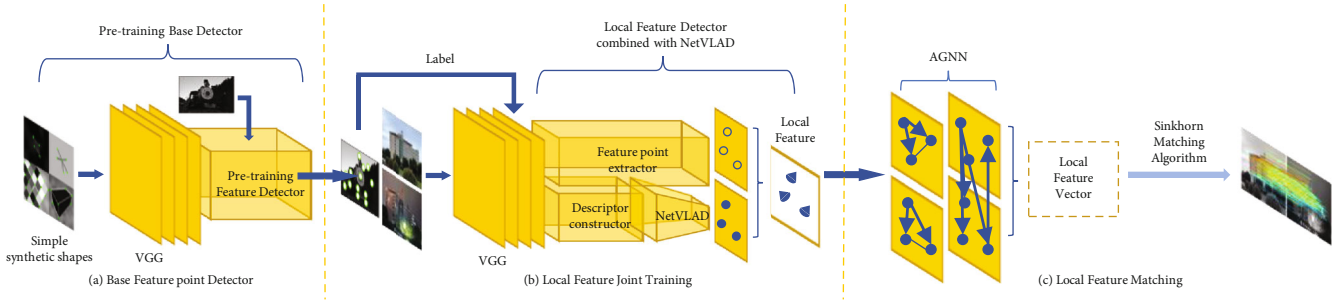


Figure 1: The fundamental network frame of local feature matching. It consists of three parts.

training network outputs the local feature vector composed of the feature position and descriptor vector, and the descriptor vector is integrated and computed by the NetVLAD module in the descriptor constitution stage. Going together of local feature vectors is the third stage of this work. The AGNN layer iteratively updates the local features that output in the second phase and then a fresh descriptor vector with a strong connection is output; eventually, the newly obtained descriptor vector is matched by the Sinkhorn algorithm, which outputs the final matching result of the image pair. As shown in Table 1, compared with the former, the algorithm proposed in this paper uses the deep neural network framework to construct strong, robust local features, enhances the relationship between local features by AGNN, and constructs the process of input to matching with complete channel image pairs. The following will be divided into multiple sections to describe in detail the three parts of the mainframe and the loss function of the network.

2.2. Base Feature Detector. Acquiring tagged datasets is one of the problematic points of leading depth neural networks into machine vision localization tasks. This paper uses a basic feature detector to obtain abundant natural scene image feature point labels. The network structure diagram of this part corresponds to Figure 1(a), and the specific details are shown in Figure 2. The central network framework of this part of the network uses VGG [24], and the convolution kernel of every layer in the network is 3×3. The last layer outputs the feature map of 65 channels that the dimension size is 15×20. An 8×8 area in a simple input image is used as a feature represented by each channel and then the multichannel feature map through the reconstruction upsampling layer that used 1×1 convolution kernel, which makes the network obtain the feature point images that accompany the identical output and input dimension. At the start of the training, firstly, the ordinary shape synthesis generator [15] of the Python file is used to generate a 120×160-dimensional dataset that is the data to be input, which contains ordinary synthetic forms such as polygons, quadrilaterals, stars, and chessboards. After obtaining the base feature point detector, the self-supervised learning means produces the feature point label for the unlabeled natural sight image.

To strengthen the adaptability of the model to the natural scene data, firstly, the input real sight data is handled by multiple homography changes like rotation and scaling, and then the simple graph data marked with corner features are used as the pretext for self-supervised learning. After the above two steps, the model has homography invariance for the input data. Finally, the image data with feature points output in this stage is used as the label data in the next stage. The labeling process of natural scene data is shown in Figure 3. In this stage:

$$x = H^{-1} f_\theta(H(I)). \tag{1}$$

x is the elementary corner vector, $f_\theta(.)$ is the fundamental feature extractor model, H is the homography transmutation matrix, and I indicates the input image. Therefore, the following formula can be used to represent the feature detection model:

$$\widehat{F}(I; \mathrm{f}_\theta) = \frac{1}{N_h} \sum_{i=1}^{N_h} x. \tag{2}$$

The frequency of use of homography transformations is represented by N_h. The joint training network of local features construction is described in detail below.

The original image undergoes a variety of homography matrix changes to obtain various images (N_h in Eq. (2) represents the number of transformations). Then, perform feature point extraction on different images (the extraction process is represented by $f_\theta(.)$ in formula (1)). Finally, the different homography images are superimposed and calculated (the superimposition process is expressed by formula (2)).

2.3. Local Feature United Training Network. Although the descriptors put forward in some prior researchs have invariant characteristics, their invariant characteristics remain in the fixed descriptors, making the descriptors unable to adapt to various change scenarios and causing redundancy of descriptors invariant characteristics. For example, if the descriptor only has the characteristic of illumination invariance, its matching effect on the image pair with changing viewpoint is not as good as the matching effect on the image pair with changing illumination. Similarly, the matching result of descriptors with single viewpoint invariance on a couple of illumination converted images is not as fine as that of a couple of viewpoints converted images. To deal with this problem effectively, this work strengthens the robustness of

Table 1: Related work comparison.

	Full image input	Feature points	Descriptors	Matching	Single network
Ours	√	√	√	√	√
SuperGlue [19]				√	
Superpoint [14]	√	√	√		√
UCN [16]	√		√		√
LIFT [10]		√	√		
TILDE [11]		√			√
DeepDesc [13]			√		√
BRIEF [5]			√		
FAST [3]		√			
SURF [4]		√	√		
SIFT [2]		√	√		
ORB [7]		√	√		

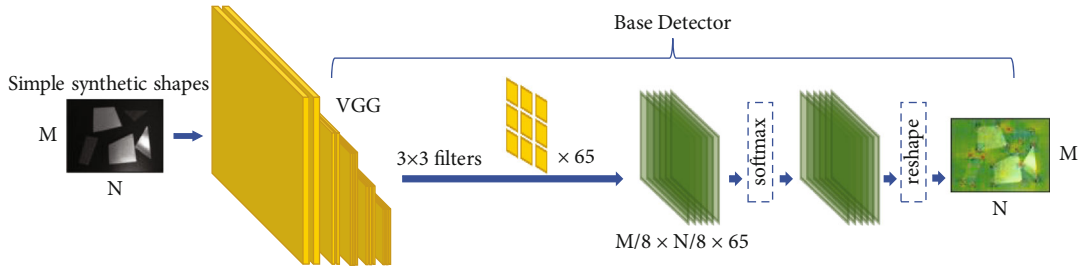


Figure 2: Elementary feature point detection phase.

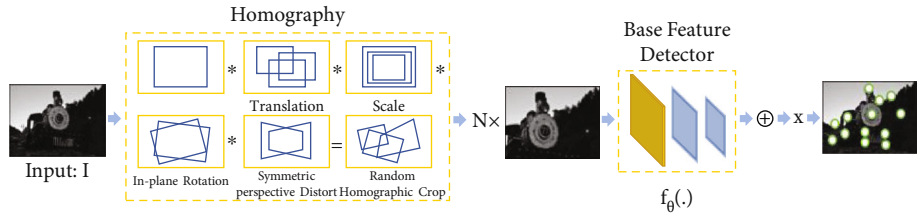


Figure 3: Natural scene data annotation.

descriptors to natural changing scenes. By using the NetVLAD module in the descriptor constitution stage, the diagram is shown in Figure 1(b).

As illustrated in Section 2.1, the natural scene data marked with feature points is used as the label data of the local feature united training network. As indicated in Figure 4, the 8-layer encoder based on VGG is used as the shared network of feature point extractor and descriptor builder to reduce the dimension of input image data. The number of convolution cores in each layer of the shared network is 64-64-64-64-128-128-128-128, and the dimension of convolution cores is 3×3. Every two layers are linked to the max-pooling layer, and the network uses ReLU nonlinear activation function. After the shared network reduces the input data dimension, the input data outputs the position vector p of feature point through the feature point detection model.

In the descriptor construction stage, the NetVLAD module, as indicated in Figure 5, is introduced. The introduction of this module can confirm the descriptor invariant traits for various changing scenes. Its advantages are shown in the test results of the experiment in Section 4.1. Supposing a collection of $H \times W \times D$ feature pictures as the input, which through the preconvolution network trained by multiscene transformation data, it outputs four groups of $H/8 \times W/8 \times 128$-dimensional local characteristic graphs that constitute illumination and viewpoint variations. The four semidense descriptors are, respectively, used as the input x of a NetVLAD module, and finally, the four descriptor vectors output by the NetVLAD module are integrated to obtain the final descriptor vector with strong robustness. Set the number of local descriptor inputs of NetVLAD module to 8, and then cluster center $D \times K$ obtains the elements of image description matrix v through clustering algorithm. Eventually, normalizing the description matrix v is implemented to obtain the descriptor vector represented by d.

2.4. Matching Layer Integrated with AGNN and Sinkhorn Algorithm. A large number of ambient intervention elements

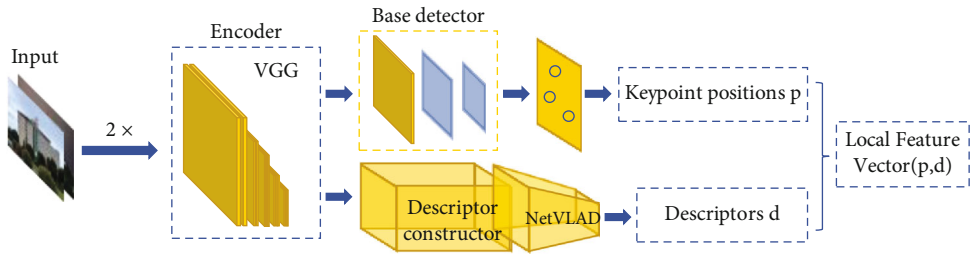


Figure 4: United training network with the insertion of NetVLAD.

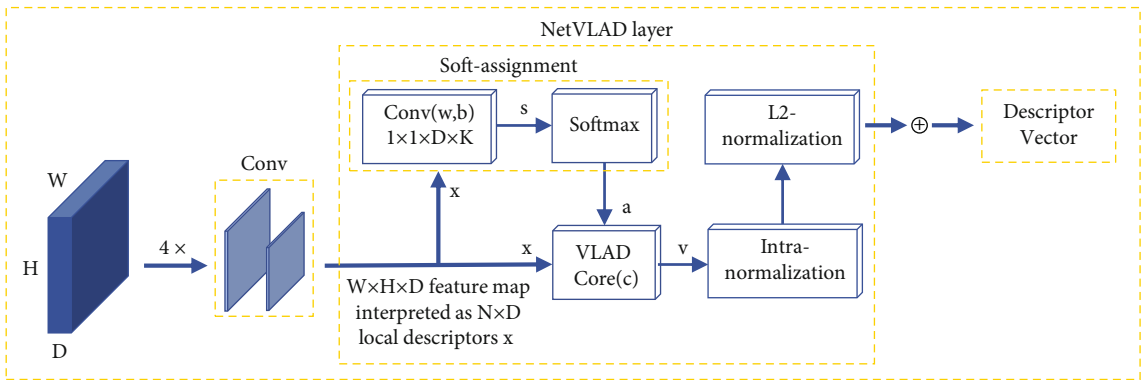


Figure 5: NetVLAD module.

in the real sight will lose the matching information between local features of the image pair, which will affect the subsequent machine vision work; as shown in Figure 1(c), to improve the above challenge, this paper infuses AGNN to achieve the target that strengthen the local feature information of the image itself and image pairs by multiple iterations. Iteration aims to strengthen the connection between local features of an image pair to obtain a better matching effect. As shown in Figure 6, the input of AGNN includes the position vector p and descriptor vector d of local features, and inputs p_i and d_i are mapped into a single vector through the feature point encoder. A five-layer multilayer perceptron encrypts p_i. Then, the local feature vector outputs 256-dimensional data under the action of attention, which is expressed as descriptor vector d, which is integrated to obtain the local feature vector x_i.

$$x_i = d_i + MLP_{\text{enc}}(p_i). \tag{3}$$

In principle, the upper equation links the descriptor and the corresponding feature location, which enables the following attention methodology of encrypting proceeding to ruminate the resemblances between local features entirely. The local feature vector received by the prework transforms to a mapping pattern that contains information about the query, keys, and values, where query and key are calculated by softmax, and then output the weight between adjacent local features, which is multiplied by another element. The upper operation consequences output the refreshed value delta after MLP. Then, to strengthen the linkage between local feature vectors, renewing local features iteratively by using delta.

$$\text{delta} = \text{MLP}\left(\left[{}^{(l)}x_i^{A_{\text{or}}B} \| m_{\varepsilon\longrightarrow i}\right]\right). \tag{4}$$

The refreshed local feature vector can be indicated as

$${}^{(l+1)}x_i^{A_{\text{or}}B} = {}^{(l)}x_i^{A_{\text{or}}B} + \text{delta}. \tag{5}$$

The node info in AGNN is presented by the $m_{\varepsilon\longrightarrow i}$ in the delta treated as the polymerized data. Particular local features are used by AGNN as nodes to accumulate rest nodes surrounding it, indicated as

$$m_{\varepsilon\longrightarrow i} = \sum_{j:(i,j)\in\varepsilon} \alpha_{ij} v_j. \tag{6}$$

The correlation of associated feature points is indicated by attention weight α_{ij}, and the mapping value from local features to graph nodes is indicated by v_j. In Figure 7, to strengthen the matching behavior between local feature vectors of different images, the correlation is weighted by cyclic iteration. To refresh the local features, using AGNN with self and crossattention mechanism is carried on, and the number of iterations is set to 9 in this paper. If A and B are input image pairs, their comparative graphs are Q and S, respectively, and W indicates the similitude among key and query. The association among the local features of the inputs and the node through the AGNN layer is

$$q_i = W^{(L)} x_i^{Q_{\text{or}}S} + b, \tag{7}$$

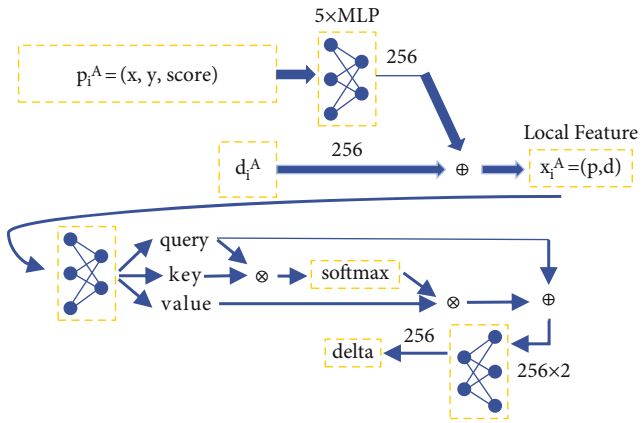


Figure 6: Local feature renewing integrated with attention mechanism.

which obtains nodes through local features handled by the AGNN layer. The ultimate matching descriptor vector of A and B can be indicated as

$$f_i^A = W\bullet^{(L)}x_i^A + b, \forall i \in A, \tag{8}$$

$$f_i^B = W\bullet^{(L)}x_i^B + b, \forall i \in B. \tag{9}$$

In the second stage of Figure 7(b), to achieve the local feature matching degree numerical matrix, it is necessary to calculate the inner product between the local feature vectors of various images. Then, this paper uses the Sinkhorn algorithm to indicate the optimal local feature matching matrix:

$$S_{i,j} = <f_i^A, f_i^B>, \forall (i,j) \in A \times B. \tag{10}$$

To percolate the inaccurate matching points between different image pairs, a channel in the matching score matrix is broadened to reserve undiscovered matching constituents. The broadened matching score matrix is $(M+1)\times(N+1)$ dimensions, which can be indicated as

$$\bar{s}_{i,N+1} = \bar{s}_{j,M+1} = \bar{s}_{M+1,N+1} = z \in \mathbb{R}. \tag{11}$$

To achieve the final image pair matching matrix $P = \bar{P}_{1:M,1:N}$, the matching score matrix $\bar{s}_{M+1,N+1}$ performs $T = 100$ times of alternating iterative calculation including row and column normalization through the Sinkhorn matching algorithm. At the same time, to ensure that the local feature vectors have a better precise matching matrix, the Sinkhorn algorithm draws into the entropic regularization function to legitimately distribute the loss weight in AGNN. The local features in the image pair are all in the form of a matrix, and the Sinkhorn algorithm needs to solve the Wasserstein distance deformation between the two matrices. Finally, outputting the image pair matching result has been realized.

2.5. Loss Function. To facilitate the optimization of network parameter settings, it is necessary to obtain the training degree, performance, and advantages and disadvantages of the network model. The joint loss function indicated in equation (12) is the loss function employed in the whole workflow. The loss function composes a feature detection module section L_{position}, a descriptor construction section $L_{\text{descriptor}}$, and a matching section L_{matching}.

$$\text{LOSS} = L_{\text{position}}(X, Y) + L_{\text{descriptor}}(D) + L_{\text{matching}}. \tag{12}$$

The feature detection section of the loss function is revealed below:

$$L_{\text{position}}(X, Y) = \frac{1}{(H_c W_c)} \sum_{\substack{h=1 \\ w=1}}^{H_c W_c} l_p(x_{hw} : y_{hw}). \tag{13}$$

Crossentropy loss [25] is applied in this function, where X and Y are the detected feature and the tagged feature locations after the homography matrix computation, $H_c W_c$ is the dimension of the inputs following the dimension decrease of the pooling layer, and $l_p(x_{hw} : y_{hw})$ is the logarithmic odds diffusion function.

In our work, the invariant descriptor loss function put forward by paurat et al. [17] is used as the loss function in the descriptor construction stage. Assuming there is a descriptor vector D and, the inputs I^a and I^b, the loss function of local feature descriptor is indicated as

$$L_{\text{descriptor}}(D) = \frac{1}{|D|} \sum_{d \in D} l\left(I_d^a, I_d^b, \text{dist}\right). \tag{14}$$

dist is the two-norm Euclidean distance between the features of the inputs I^a and I^b, and $l(I_d^a, I_d^b, dist)$ is the loss function comprising the local feature descriptor and the Euclidean distance of the image pair. The loss function's establishment helps the model more accurately extract the local descriptors suitable for matching in the next stage of the dense descriptor mapping.

In the process of model iteration, the loss function of the matching section requires refreshing and optimization to ensure the effectiveness and optimality of the algorithm. If the matching network is trained by the natural scene dataset M, the labeled homography transformation is used to estimate the matching of the natural scene image pair. The loss function is the difference of the negative logarithmic minimization of the training image corresponding to the matching matrix P, which is indicated below:

$$L_{\text{matching}} = -\sum_{(i,j)\in M} \log P_{i,j} - \sum_{(i,j)\in A,B} \log P_{i,j}. \tag{15}$$

The real sight dataset is revealed by M, and the input image pair is revealed by (A, B).

3. Training Particulars of Network

This part will introduce the specific details of network training, including the evaluation indexes of model performance

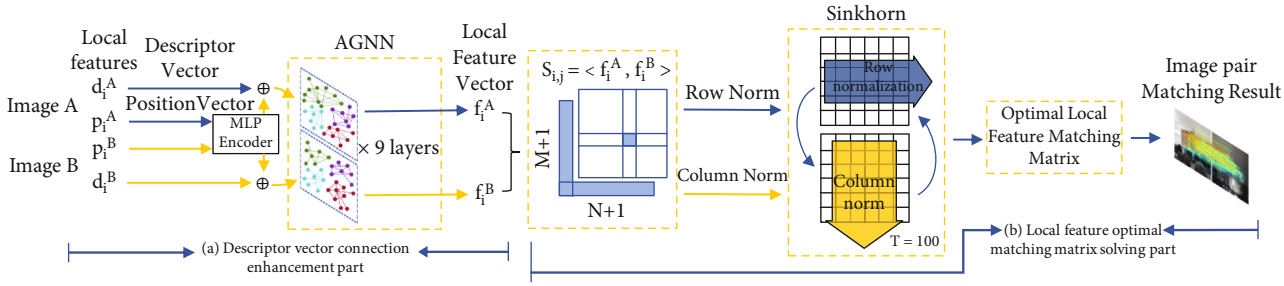


Figure 7: Local feature matching process, which contains two parts (a) and (b).

testing, the training data set used, and the specific parameter settings of the training network.

3.1. Evaluation Metrics

3.1.1. Homography Estimation. Homography estimation (HEstimation) is a critical reference to reflect the pros and cons of matching images and is widely used in machine vision fields such as SLAM. This paper refers to the local feature descriptor evaluation protocol proposed by Mikolajczyk et al. [14, 18, 26] to evaluate the matching performance of the model proposed in this paper and compare the work of the former. The homography matrix describes the mapping relationship between two planes, that is, the transformation relationship between two image feature points on a common plane. Supposing there is an image pair A and B with well-matched feature points p_1 and p_2, which fall on the plane P, and this plane satisfies the equation:

$$n^T P + d = 0. \tag{16}$$

Supposing K is the camera internal parameter matrix, t and R represent the camera motion between adjacent frames, where t is the translation vector, and R is the rotation matrix. The location of the feature point can be expressed as

$$p_2 \simeq K(RP + t) \simeq K\left(RP + t\bullet\left(-\frac{n^T P}{d}\right)\right) \simeq K\left(R - \frac{tn^T}{d}\right)K^{-1}p_1. \tag{17}$$

From the above formula, the transformation $p_2 \simeq Hp_1$ between the image coordinates p_1 and p_2 can be described, and H is the homography matrix.

Assuming that the pixel coordinates of the image pair feature points p_1 and p_2 are (u1, v1, 1) and (u2, v2, 1), H, as a 3×3 matrix, the transformation relationship is as follows:

$$\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} \simeq \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix}. \tag{18}$$

Conversion relationship $u_2 = h_{11}u_1 + h_{12}v_1 + h_{13}/h_{31}u_1 + h_{32}v_1 + h_{33}$, $v_2 = h_{21}u_1 + h_{22}v_1 + h_{23}/h_{31}u_1 + h_{32}v_1 + h_{33}$, can be obtained, and the matrix form can be obtained after sorting: $A\theta = 0$, where

$$A = \begin{bmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -u_2u_1 & -u_2v_1 & -u_2 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -v_2u_1 & -v_2v_1 & -v_2 \end{bmatrix}, \tag{19}$$

$$\theta = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{21} & h_{22} & h_{11} & h_{23} & h_{31} & h_{32} & h_{33} \end{bmatrix}. \tag{20}$$

The homography matrix has 8 degrees of freedom, and eight linear equations are needed to solve it. In summary, it can be concluded that two linear equations can be listed for a pair of matching feature points of a set of image pairs; that is, at least four pairs of matching feature points are required to solve the homography matrix H by the formula $A\theta = 0$.

This article estimates how the homography transformation transforms the four corner points of the first frame to the next frame of an image. Assuming that the four corner points of the first frame are $\hat{c}_{1\cdots4}$ and the four corner points of the second frame are $\hat{c}_{1\cdots4}{}'$, the homography matrix H is obtained by the homography transformation. Setting the threshold ϵ, the homography evaluation formula is expressed as follows:

$$\text{HEstimation} = \frac{1}{4}\sum_{i=1}^{4} \|\hat{c}_i - c_i\|_2 \le \epsilon. \tag{21}$$

3.1.2. Matching Evaluation Metrics. Precision is the average matching accuracy, referring to the percentage of correct matches in all predicted matches. Recall is the ratio of correctly predicted matches to the total number of labeled matches. Assuming that the correct match of the label is 1, the false match is 0; the predicted correct match is 1, and the false match is 0. The true and false positive graphs are shown in Table 2.

Among them, false positives are matching deviations, and false negatives are uncorrelated matches. To improve the reliability of the matching algorithm, the occurrence of FP and FN is usually reduced as much as possible, and the occurrence probability of TP and TN should be as high as possible, enabling the algorithm to screen out

Table 2: The classification of predicted and truth matching results.

	Predicted matching result 1	Predicted matching result 0
Truth matching result 1	True positive(TP)	False positive(FP)
Truth matching result 0	False negative(FN)	True negative(TN)

false matches more accurately. The precision and recall formulas are as follows.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{22}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{23}$$

3.2. Datasets. The training aggregation of the whole network applies the COCO14 dataset [27] that contains 80,000 images, and the input dimensionality of every images are 240×320, which is used to train the shared network; and the image dataset put forward in [28] includes diverse viewpoints and illuminations; between them, the training aggregation of feature detection, descriptor constitution, and local feature matching uses COCO14 dataset, which can guarantee that the ultimate model has adaptability in real scenes. The descriptor construction phase primarily uses the Multi-Illumination dataset put forward in [28], it contains 1000 natural scene images, and each image has 25 images under different illumination. The dataset has a total of 25000 images. The constantness of descriptors to environmental changes is ensured by the image features of various viewpoints and illumination in an identical site. Table 3 shows the instance of the dataset applied in this work.

3.3. Implementation. The experimental context of this work consists of hardware devices and system environment settings. The concrete type and version are NVIDIA RTX2080Ti GPU; Intel Core CPU I7-10700 K 3.8GHz eight-core; Ubuntu18.04, Spyder (Python3.6). Moreover, the realization of the deep neural network is carried out on the PyTorch framework.

In the training phase, a simple synthetic dataset is used as the input for the iterative training of the basic feature detector. After that, the basic feature detector is trained by the real scene dataset with a dimension of 240×320, and the output of this part is applied as the label data of the shared network in the local feature united training phase. In the descriptor constitution phase of the united training network, initially, a 128-channel descriptor is output, and then a 512 channel metadescriptor is output, which is composed of four 128 dimensional descriptors. Batch size is set to 32 in the entire training procedure, Adam gradient optimization algorithm is applied, and the initial learning rate is set to 0.001. The number of training epochs was 20. In this paper, the learning rate is set to change with the number of training times. If the model tends to be stable in 10 epochs, the frequency of training is reduced by half, and then the training is discontinued after 20 epochs. Through the above methods, the purpose of saving calculation cost and improving model training efficiency is achieved. After completing training, the input image pairs pass through the local feature constitution layer of the united network, which achieves the local features applied as input in the matching phase. The local features obtained above are iterated 9 times by AGNN, which outputs the updated local feature vector; subsequently, the new local feature vector is internally multiplied to achieve the matching numerical matrix. The matching numerical matrix is then computed by 100 times of alternating row and column normalization of the Sinkhorn algorithm. The ultimate matching consequence was obtained after achieving the optimized matching matrix.

4. Experiments and Discussion

This part will present the concrete experimental consequences. The experiment includes the descriptor and the matching algorithm tests. Compared with the existing algorithm, the experiment effectively verifies the performance of the local feature and matching algorithm.

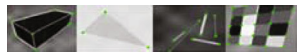
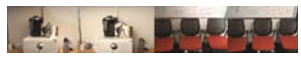
4.1. Descriptor Test Experiment. This work has tested many experiments on self-collected image pairs and open datasets to testify the availability of the local features put forward in this work.

4.1.1. Effect Exhibition of Feature Descriptor. To intuitively describe the performance effects of feature descriptors with invariant environment characteristics, this article first made a set of visual displays as shown in Table 4, comparing SIFT, Superpoint, and the algorithm proposed in this article, and collected two sets of the viewpoint and the rotation of the 1920×1200-dimensional image pair of the experimental building are set to collect up to 200 feature points. The nearest neighbor (NN) matching algorithm and the RANSAC mismatch optimization algorithm [29] are applied. The experimental results are shown in Table 4, and the method proposed in this paper (with invariance) has a better matching performance result than the feature descriptor without environment invariance.

4.1.2. Validation of the Invariance of Illumination and Viewpoint. The purpose of this experiment is to effectively verify that the descriptor constructed by the algorithm in this work has a better behavior consequence in realistic scenarios than the former, and this work applies the dataset HPatches [30], which is exclusively applied to assess descriptors and the model behavior. The 57 scene image data of this dataset all contain environmental transformation factors, containing the HPatches Illum dataset with illumination change and the HPatches View dataset with viewpoint change. Both datasets are applied to evaluate the homography, matching precision, and the recall of the model.

In the experiments displayed in Figures 8(a) and 8(b), the two groups were tested on the illumination and viewpoint change datasets of HPatches, respectively. The test results indicate the influence of the adaptive invariance

Table 3: Datasets examples.

	Dataset diagram
Sythetic Dataset [15]	
COCO2014 [27]	
Multi-Illumination Dataset [28]	

characteristics of descriptors on the matching results. In the experiment, 500 is set as the maximum number of feature points extracted from a single image, and the threshold is tune-up to compute the relevant matching precision. The threshold set in this experiment is the pixel correctness threshold, and its purpose is to compare the pixels contained in the feature points with the threshold to make a matching judgment. Different threshold settings will lead to different matching judgment decisions between the corresponding feature points of the image pair, and the comparison of matching precision under different thresholds can better illustrate the performance advantages of the algorithm.

It can be concluded from Figure 8(a) that descriptors with specified characteristics (Illum invar, rot var) have higher matching precision than other descriptors in experiments on the HPatches Illum dataset. Moreover, the experimental results of Figure 8(b) are just opposite to those of (a). In general, the above two experiments prove the availability of the environmental invariant characters applied in the descriptor.

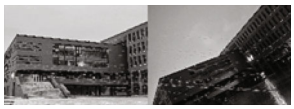
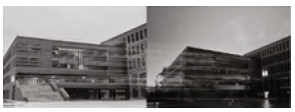
4.1.3. Feature Descriptor Algorithm Comparison. The experiment shown in Figure 8(c) compares our put forward method and other existing algorithms on the HPatches View dataset. It can be seen that our proposed method has higher matching precision. This experiment proves the effectiveness and the progressive nature of our descriptor method in natural scenes. In this experiment, the test dataset dimensionality is preprocessed, and 500 is set as the maximum quantity of extracted feature points of a single test image to guarantee the reliability of comparison between algorithms.

Meanwhile, this paper makes some evaluations on various descriptors on the HPatches dataset (HEstimation, matching precision, and matching recall) to better guarantee that our descriptor has superior performance in matching. The experiments in Tables 5 and 6 separately use HPatches Illum and HPatches View datasets. These two experiments set the same operation method: using the same threshold equal to 3, the nearest neighbor matching algorithm, and the RANSAC mismatching optimization algorithm [29]. As shown in Tables 5 and 6, in these two experiments, the descriptors with adaptability to changing environments put forward in this paper have superior performance in matching results in both HPatches Illum and View datasets. In the experiment in Table 5, the homography estimation is 0.883, the matching recall is 0.664, and the matching precision is 0.664. Besides, in the experiment in Table 6, the homography estimation in this work reaches 0.687, the matching recall is 0.494, and the matching precision is 0.625.

4.1.4. Matching Test Experiment. To make the local feature matching algorithm effectively deal with changing scenes, this paper makes use of the coordination of the AGNN+Sinkhorn matching algorithm and invariant feature descriptors. In this experiment, the combination of various local feature construction methods and various matching methods are used to test the matching outcome on the HPatches View dataset to prove the matching performance of our algorithm. The parameter selection in the local feature extraction phase is identical to the experiment shown in Tables 5 and 6, which sets 500 as the maximum quantity of local features collected for a single image. From Table 7, the various local feature construction algorithms using AGNN + Sinkhorn matching algorithm perform better than the conventional nearest neighbor matching approach in multiple indicators. The end-to-end algorithm integrating local feature formation and matching put forward in this paper has been dramatically increased in various evaluation indexes compared with the rest approaches. The homography estimation is raised by 0.097, with the matching precision is improved by 0.082, and the matching recall is rose by 0.038.

Meanwhile, this paper also gathers image pairs of the library, gymnasium, experimental building, and administrative building. These image pairs contain illumination, rotation, and viewpoint transformations. Before the test, the image pairs are preprocessed in dimension, and the pixel size is set to 1920×1200. The experiment aims to prove the effectiveness and superiority of the AGNN + Sinkhorn matching algorithm. Figure 9 indicates the comparison of visual matching results between the algorithm in this paper and some existing algorithms. In the experiment, 300 is set as the upper limit of feature point extraction of each algorithm on a single image. The first row is the original picture, the second row is the matching result of the ORB algorithm, the third row is the matching result of the SIFT algorithm, the fourth row is the matching result of the Superpoint algorithm, and the fifth row is for the matching results of the algorithm in this paper. In the last two columns of gymnasium and library image pairs with considerable viewpoints and illumination contrast, compared to other existing algorithms, the method put forward in this work can be used for local feature extraction and effective matching of image pairs in the presence of illumination, viewpoints, and abundant foliage occlusions and architectural interference. The color of the matching line in the figure indicates the matching confidence of the corresponding local features between the image pairs. The more the color is toward red, the higher the confidence of matching, and the more the color is toward blue, the lower the confidence of matching. To sum up, under the premise that the maximum extraction times of various local feature extraction algorithms are specified, the local features between image pairs can be more precisely and effectively matched under the algorithm proposed in this article.

Table 4: The performance of descriptor invariance in mutative environments.

	SIFT	SuperPoint	With invariance
With viewpoint, illumination, and rotation			
With viewpoint, illumination, and no rotation			

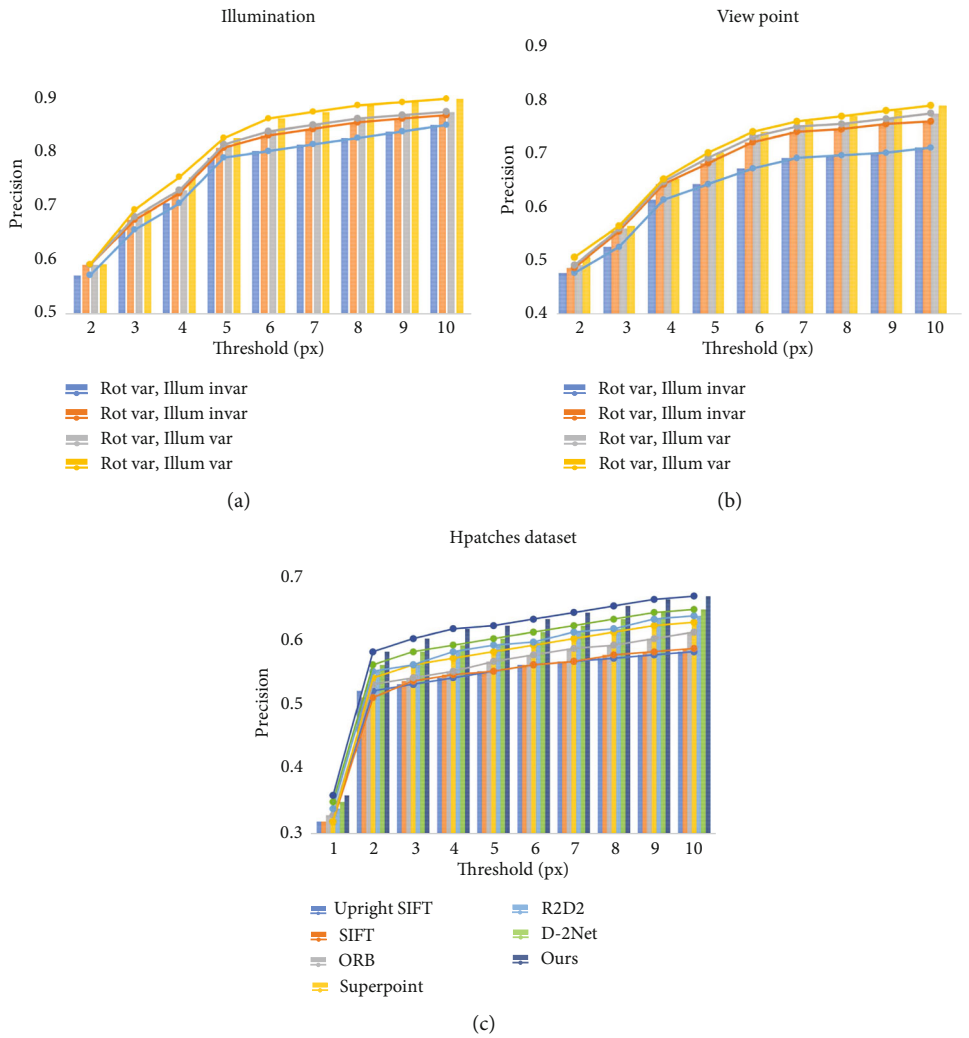


Figure 8: Descriptor experiment.

Table 5: The index comparison of HPatches Illum.

Dataset	Evaluation	SIFT	ORB	Superpoint	R2D2 [31]	D2-net [32]	Our method
HPatches Illum	Precision	0.553	0.564	0.629	0.656	0.651	0.664
	Recall	0.431	0.483	0.564	0.580	0.565	0.664
	HEstimation	0.897	0.884	0.876	0.816	0.818	0.883

Table 6: The index comparison of HPatches View.

Dataset	Evaluation	SIFT	ORB	Superpoint	R2D2 [31]	D2-net [32]	Our method
HPatches View	Precision	0.514	0.572	0.594	0.549	0.563	0.625
	Recall	0.349	0.421	0.445	0.371	0.381	0.494
	HEstimation	0.643	0.687	0.651	0.626	0.552	0.687

Table 7: The experimental contrast results.

Matching algorithm	Local feature	HEstimation	Precision	Recall
AGNN + Sinkhorn	SIFT	0.656	0.625	0.413
AGNN + Sinkhorn	ORB	0.692	0.677	0.579
Nearest neighbor	SIFT	0.643	0.514	0.349
Nearest neighbor	ORB	0.687	0.572	0.421
Nearest neighbor	Superpoint	0.651	0.594	0.445
Nearest neighbor	Des + NetVLAD	0.687	0.625	0.494
SuperGlue [19]		0.668	0.725	0.596
Our proposed method		0.784	0.807	0.634
Promotion		0.097	0.082	0.038

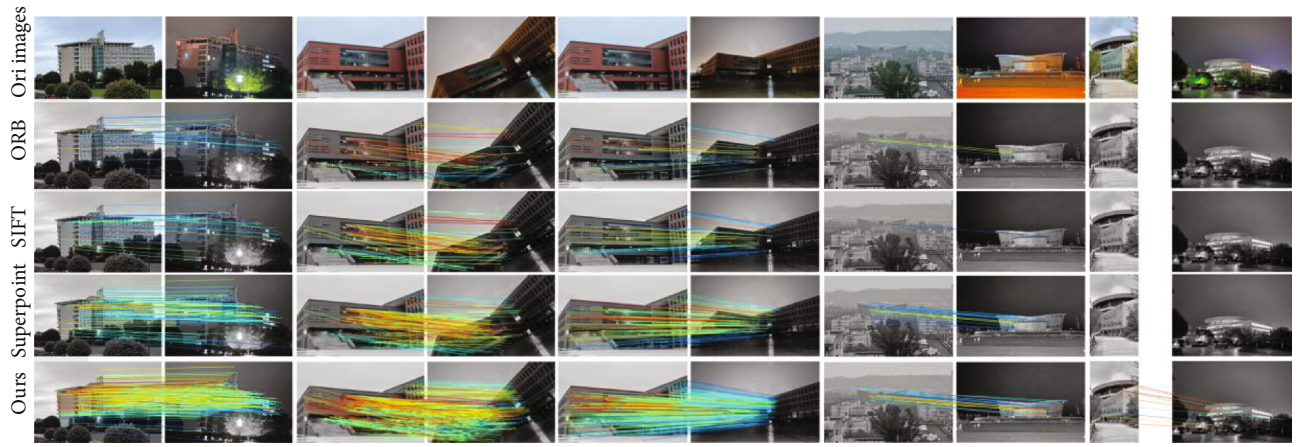


Figure 9: Matching self-collected changing pairs with ours and comparison algorithms.

5. Conclusions

This paper studies a local feature matching method in which the NetVLAD neural network module is implanted in the descriptor constitution phase, and the AGNN and Sinkhron matching algorithm is introduced in the matching stage, which effectively realizes the robustness of the algorithm against scene changes. Compared with typical algorithms, this article has achieved better behavior consequences in homography estimation, matching precision, and matching recall on the public dataset HPatches and self-collected image pairs with significant scene changes. Meanwhile, a single end-to-end network that can match local features more accurately and effectively is achieved. Owing to the strengthening of the invariance of descriptors, even though the matching problem between local features of natural scene images can be well done, the number and complexity of descriptors corresponding to feature points have also expanded compared with traditional algorithms resulting in calculation increase in cost. Finally, the research on the fusion of deep neural networks and local feature matching engineering has received increasing attention. The introduction of deep learning into machine vision-related issues such as visual odometry engineering has also received increasingly investment and research. However, the stability and effectiveness of the system after effective fusion are also nodes and difficulties to be considered, as well as issues that need to be studied in the follow-up.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declared that there are no potential conflicts of interest with any financial organizations regarding the material reported in this article.

Acknowledgments

The research of the paper is supported by the Project of National Funding, PRC ([2019], No. 1276).

References

[1] M. Yan, H. Yuan, J. Xu, Y. Yu, and L. Jin, "Task allocation and route planning of multiple UAVs in a marine environment based on an improved particle swarm optimization algorithm," *EURASIP Journal on Advances in Signal Processing*, vol. 2021, no. 1, 2021.

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

[3] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*, pp. 430–443, Berlin, Heidelberg, 2006.

[4] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: speeded up robust features," in *European conference on computer vision*, pp. 404–417, Berlin, Heidelberg, 2006.

[5] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: binary robust independent elementary features," in *European conference on computer vision*, pp. 778–792, Berlin, Heidelberg, 2010.

[6] M. Yan, S. Li, C. A. Chan, Y. Shen, and Y. Yu, "Mobility prediction using a weighted Markov model based on mobile user classification," *Sensors*, vol. 21, no. 5, p. 2021, 1740.

[7] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *2011 International conference on computer vision*, pp. 2564–2571, Barcelona, Spain, 2011.

[8] M. Yan, X. Lou, and Y. Wang, "Channel noise optimization of polar codes decoding based on a convolutional neural network," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 1434347, 10 pages, 2021.

[9] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Deep image homography estimation," 2016, https://arxiv.org/abs/1606.03798.

[10] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "Lift: learned invariant feature transform," in *European conference on computer vision*, pp. 467–483, Cham, Switzerland, 2016.

[11] Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit, "TILDE: a Temporally Invariant Learned DEtector," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5279–5288, Boston, MA, USA, 2015.

[12] K. M. Yi, Y. Verdie, P. Fua, and V. Lepetit, "Learning to assign orientations to feature points," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 107–116, Las Vegas, Nevada, 2016.

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

[14] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: self-supervised interest point detection and description," in *2018 Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 224–236, Salt Lake City, UT, USA, 2018.

[15] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Toward geometric deep slam," 2017, https://arxiv.org/abs/1707.07410.

[16] C. B. Choy, J. Y. Gwak, S. Savarese, and M. Chandraker, "Universal correspondence network," 2016, https://arxiv.org/abs/1606.03558.

[17] R. Pautrat, V. Larsson, M. R. Oswald, and M. Pollefeys, "Online invariance selection for local feature descriptors," in *European Conference on Computer Vision*, pp. 707–724, Cham, Switzerland, 2020.

[18] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, 2018.

[19] P. -E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: learning feature matching with graph neural networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4937–4946, Seattle, WA, USA, 2020.

[20] J. Zhou, G. Cui, S. Hu et al., "Graph neural networks: a review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[21] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc.*, pp. 6000–6010, Red Hook, NY, USA, 2017.

[22] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Advances in Neural Information Processing Systems*, vol. 26, pp. 2292–2300, 2013.

[23] S. Yang, JingWang, S. Arif, M. Jia, and S. Zhong, "SAL-net: self-supervised attribute learning for object recognition and segmentation," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 2891303, 13 pages, 2021.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[25] C. Jin, T. Wang, X. Li et al., "A transformer generative adversarial network for multi-track music generation," *CAAI Transactions on Intelligence Technology*, vol. 2021, pp. 1–12, 2021.

[26] 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.

[27] T. Y. Lin, M. Maire, S. Belongie et al., "Microsoft coco: common objects in context," in *European conference on computer vision*, pp. 740–755, Cham, Switzerland, 2014.

[28] L. Murmann, M. Gharbi, M. Aittala, and F. Durand, "A dataset of multi-illumination images in the wild," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4080–4089, Seoul, Republic of Korea, 2019.

[29] K. G. Derpanis, "Overview of the RANSAC algorithm," *Image Rochester NY*, vol. 4, no. 1, pp. 2-3, 2010.

[30] V. Balntas, K. Lenc, A. Vedaldi, T. Tuytelaars, J. Matas, and K. Mikolajczyk, "[Formula: see text]-Patches: a benchmark and evaluation of handcrafted and learned local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 11, pp. 2825–2841, 2020.

[31] R. Jerome, W. Philippe, D. S. Cesar, and H. Martin, *R2D2: Repeatable and Reliable Detector and Descriptor*, NeurIPS, 2019.

[32] M. Dusmanu, I. Rocco, T. Pajdla et al., "D2-Net: a trainable CNN for joint description and detection of local features," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8084–8093, Long Beach, CA, USA, 2019.