2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC) | 978-1-6654-7968-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/IMCEC55388.2022.10019834

Improved Feature Point and Descriptor Extraction Network for SuperPoint

Qihan Suo ,Changqing Yan* ,Chi Liu ,Chenyin Ma
College of Intelligent Equipment, Shandong University of Science and Technology, Tai'an, China
Suo97_July@outlook.com, yancq@sdust.edu.cn, 2383167531@qq.com, 295159830@qq.com
Corresponding Author: Changqing Yan Email: yancq@sdust.edu.cn

***Abstract*—Image feature point and descriptor extraction is the basis of SLAM, SFM and 3D reconstruction tasks. In this paper, we study the SuperPoint network, which has good robustness in extracting feature points and descriptors, and introduces the idea of group convolution, replaces the normal convolution with group convolution, and introduces the Mish activation function to replace the ReLU activation function to solve the problem that some data fall into negative intervals. The experimental results show that the accuracy of single-strain estimation only decreases by 0.01 when the tolerance distance difference is 3, and the repetition rate of feature point detection increases by 0.3%, which has good robustness. In this paper, the network achieves lightness without excessive loss of accuracy.**

Keywords*—Deep Learning; SuperPoint; Group Convolution; Mish function

I. INTRODUCTION

Feature point and descriptor extraction is one of the core problems in the field of computer vision. The purpose of feature point extraction is to show the key information of the image, and the purpose of descriptor extraction is to describe the information around the feature point and generate feature vectors to distinguish different regions. The extraction of image interest points and descriptors is widely used in image matching, 3D reconstruction [1] and SLAM [2].

There are two main categories of the feature point and descriptor extraction algorithms: traditional image feature extraction algorithms and algorithms that extract features by deep learning [3].

The traditional extraction algorithms are based on the image grayscale mean, texture and other information as the extraction target, and the image information is obtained entirely by hand-designed algorithms, such as SIFT and SURF algorithms. The SIFT algorithm proposed by Lowe [4] is scale-invariant, i.e., when the brightness or offset rotation angle of the sample changes, it can still perform feature extraction well. The algorithm is to first calculate the scale space by approximating the Gaussian fuzzy function with different parameters to form a Difference of Gaussian (DoG) scale space. After generating the DoG scale space, each sample point is scanned and compared with the surrounding 26 pixels to determine whether it is an extreme point. After determining the extreme points, the axes are rotated to the principal direction, and the magnitude and direction of the pixel gradient within a 16×16 window centered on the feature point are calculated and normalized to form a 128-dimensional feature vector as a descriptor. The SURF algorithm obtains the main direction of the interest points by counting the Harr wavelet features in the neighborhood of the interest points, and extracts the Harr features in the neighborhood along the main direction of the interest points to form a 64-dimensional feature vector as a descriptor.

Traditional feature point and descriptor extraction algorithms extract information by evolving and abstracting images through mathematical formulas, which have natural disadvantages in robustness and generalization compared with deep learning feature point and descriptor extraction algorithms driven by large-scale datasets. Since 2012, deep learning based feature point and descriptor extraction algorithms have developed rapidly in the fields of image classification, target detection and image segmentation with the advantages of richer extracted features, better robustness, higher accuracy and no need to design features manually. In recent years, the combination of deep learning with feature point and descriptor extraction is also a hot issue in this field.

The process of extracting feature points or descriptors by deep learning does not rely on hand-designed algorithms, but is obtained by convolutional neural network training and reasoning. The L2-Net [6] network proposed by Tian et al. has an image as input and a 128-dimensional vector as output. The network describes the distance between image features in L2 parametric terms, and the distance of descriptors is constrained by a loss function. Mishchuk et al [7] proposed HardNet with improved loss function based on L2-Net HardNet maximizes the distance between nearest neighbor positive and negative samples using loss function referring to SIFT algorithm.Barroso-Lagun et al proposed LF-Net [9] by Ono et al. and D2-Net [10] by Dusmanu et al. are both end-to-end feature learning approaches, with input images and output feature points and descriptors.LF-Net consists of two parts: a fully convolutional network generates feature points and D2-Net generates only descriptors, and then determines whether the current point is a feature or not based on whether the descriptor of the current point is a local maximum in the maximum response channel.

II. SUPERPOINT NETWORK BASED ON GROUP CONVOLUTION

A. General Structure

SuperPoint is an end-to-end feature point and descriptor extraction network based on self-supervised training, and the network structure is similar to the encoder-decoder structure of semantic segmentation. The implementation steps of the method are as follows: 1). A full convolutional neural network is used to train a point-of-interest base detector on a synthetic dataset consisting of simple geometric shapes such as triangles, quadrilaterals, lines, cubes, checkerboards, and stars, as well as noisy images. The detector trained with the synthetic dataset has good noise immunity. 2). Detect the unlabeled real scene images with the interest point basis detector trained above and apply the single-strain transformation to label the real scene images to generate the labeled dataset. 3). Use the labeled dataset generated in the second step to train the full convolutional God network, which can extract both interest points and descriptors from images. the SuperPoint framework consists of three parts, namely, a shared encoder, an interest point decoder, and a descriptor decoder. This is shown in Figure 1.

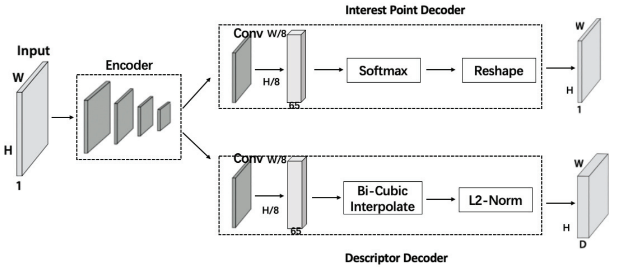

Fig. 1. SuperPoint overall framework

B. Improvement of shared encoder network structure

The shared encoder is a convolutional neural network. This convolutional neural network uses the VGG-Style [13] network model to act as an encoder, which can not only extract image features but also reduce the dimensionality of images. To address the problem of large number of parameters in the original network, in this paper, to reduce the number of parameters in the network and make the network more lightweight, the convolution method of the convolutional neural network with shared encoder is changed to group convolution, and then the activation function is replaced with the Mish function. The problem of dead neurons generated by data falling into negative intervals is solved while reducing the number of operations.

Improvement of convolution method. To make the convolutional neural network more lightweight, in this paper, the convolution method of the convolutional neural network in the shared encoder except the first convolutional layer is changed from normal convolution to grouped convolution. The idea of grouped convolution first appeared in AlexNet [12]. Compared with the ordinary convolutional network, the same convolutional operation of grouped convolution requires fewer parameters to be computed and is less prone to overfitting. Therefore, we introduce grouped convolution in the network instead of normal convolution. Grouped convolution first groups the input data and then performs the convolution operation on each group separately. Suppose the size of the input data is $W \times H \times C_1$, the size of the output data is $\mathrm{W} \times \mathrm{H} \times \mathrm{C}_2$, and the size of the convolution kernel is $K \times K$. The difference between normal convolution and grouped convolution is shown in Figure 2. The upper part of Figure 2 shows the normal convolution, and the lower part shows the grouped convolution (the number of groups in the figure is 2). For grouped convolution, if the number of groups is set to g, the input data size of each group is $W \times H \times (C_1/g)$, the output data size is $W \times H \times (C_2/g)$, the convolution kernel size is $K \times K \times (C_1/g)$, and the number is C_2/g. Each group of convolution kernel only convolves with the input data of the same group, but not with the input data of other groups, and finally all the group outputs together form the output data. In one convolution operation, the number of parameters of normal convolution is: $C_1 \times C_2 \times K \times K$, while the number of parameters of group convolution is: $(C_1/g) \times (C_2/g) \times K \times K \times g$. The parameters of group convolution are only $1/g$ of normal convolution.

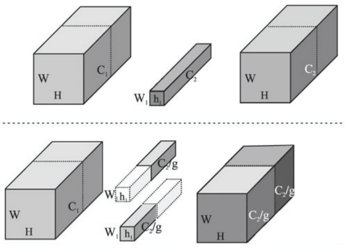

Fig. 2. Comparison of normal convolution and grouped convolution

Replace the activation function. ReLU solves the problem of gradient disappearance generated by the Sigmod function in the problem of the non-zero interval without gradient saturation, and can effectively propagate the gradient for updating. However, because there will be a part of data falling into negative intervals, these data will become 0, resulting in the corresponding weights cannot be updated and becoming the so-called dead neurons. Therefore, in this paper, the Mish [15] function is used to replace the ReLU function. The value of the Mish function in the positive direction is basically consistent with the coordinate axis, and the value in the negative direction allows smaller negative gradient values to ensure that the information will not be interrupted, and each point of the function is smoother, allowing better feature information to penetrate deeper into the convolutional network, which can achieve better accuracy as well as robustness. The function image is shown in Figure 3. the Mish function expression is shown in (1).

$$Mish = x \times \tanh(\ln(1 + e^x)) \tag{1}$$

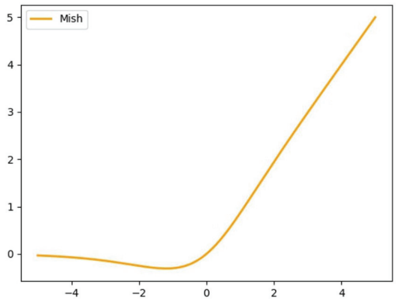

Fig. 3. Image of Mish function

The structure of the improved encoder network is shown in Figure 4.

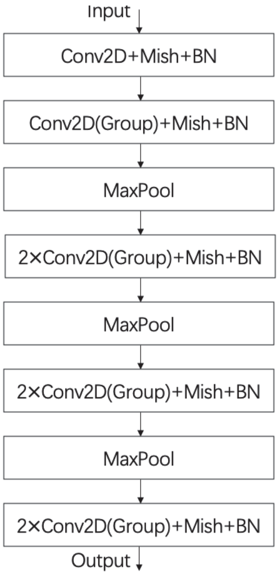

Fig. 4. Improved encoder network structure

C. Interest Point Decoder

For interest point detection, each pixel of the output corresponds to the probability that the pixel is a feature point in the input. The decoder reduces the low-dimensional output tensor to the same dimension as the input. The interest point decoder processes X $\in$ R$^{Hc\times Wc\times 65}$ and outputs the tensor as R$^{H\times W}$, which is the dimension of the image. Here 65 denotes the local area of the original image 8×8, plus a non-feature point Dustbin. then it is processed by Softmax function to remove this dimension of Dustbin. Finally, the Reshape function is executed to complete the process of R$^{Hc\times Wc\times 65}$ $\times F$ to R$^{H\times W}$. To make the model easy to train, the decoder uses non-learning upsampling.

D. Descriptor decoder

The descriptor decoder contains two convolutional layers with two convolutional kernels of size $3\times 3\times 256$ and $1\times 1\times 256$, respectively. the descriptor decoder processes D $\in$ R$^{Hc\times Wc\times D}$ and outputs a tensor of size $H\times W\times D$. Then, the decoder performs bi-trivial interpolation of the descriptors and finally performs L2 Normalization to output the descriptors.

E. Loss function

The loss function used in this paper is consistent with the loss function of the SuperPoint[11] network. The loss function consists of two components, the loss of interest points and the loss of descriptors, and is shown by equation (2).

$$L(X,X',D,D',Y,Y',S) = Lp(X,Y) + Lp(X',Y') + \lambda Ld(D,D',S) \quad (2)$$

In Eq. (2), X, D are the feature point feature map and the descriptor feature map respectively, Y is the label value of the feature point of the original image, X', D' and Y'correspond to the input images of the original image after the single-response transformation, and the rest of the meanings are the same as X, D and Y. S is illustrated by equation (5). Lp and Ld denote the feature point loss and descriptor loss, respectively, and the hyperparameter λ is used to balance the feature point detection loss and descriptor loss. the specific formula of Lp is shown in equation (3):

$$Lp(X,Y) = \frac{1}{HcWc}\sum_{h=1,w=1}^{Hc,Wc} l_p(x_{hw}; y_{hw}) \quad (3)$$

In Eq. (3), Hc, Wc denote the height and width of the feature map of interest points respectively. x_{hw}, y_{hw}denote the values of X, Y at (h,w) respectively. l_p is specified as shown in equation (4):

$$l_p(x_{hw}; y) = -\ln(\frac{\exp\,(x_{hwy})}{\sum_{k=1}^{65}\exp\,(x_{hwk})}) \quad (4)$$

In Eq. (4), x_{hwk} is denoted as the value of x_{hw} at the k channel. l_p makes x_{hw} as large as possible at the channel corresponding to the label value y.

$$L_d(D,D',S) = \frac{1}{(H_cW_c)^2}\sum_{\substack{h=1,\\ w=1}}^{Hc,Wc}\sum_{\substack{h'=1,\\ w'=1}}^{Hc,Wc} l_d(d_{hw}; d'_{h'w'}; s_{hwh'w'}) \quad (5)$$

In Eq. (5), $d_{hw}, d'_{h'w'}$ denote the values of D, D' at $(h,w), (h',w')$, respectively. Since the shared encoder is downsampled 8 times, the points in the output descriptor feature map correspond to an 8×8 pixel image cell in the input image. $s_{hwh'w'}$ is used to determine whether the center position of d_{hw} corresponding to the input image cell is within the neighborhood of the center position of $d'_{h'w'}$ corresponding to the input image cell after the single-response transformation consistent with the original image. $s_{hwh'w'}$=1 means the corresponding positions in the original image are similar. $s_{hwh\prime w\prime}$ and l_d are shown in equation (6):

$$s_{hwh\prime w\prime} = \begin{cases} 1, if ||Hp_{hw} - p_{h\prime w\prime}|| < 8 \\ 0, \; else \end{cases} \tag{6}$$

In Eq. (6), p_{hw}, $p_{h\prime w\prime}$ denote the position centers of the input image units corresponding to $d_{hw}, d'_{h\prime w\prime}$, respectively. Hp_{hw} is a single-strain transformation of p_{hw}, which is the same as the original image.

$$l_d(d; d'; s) = \lambda_d \cdot s \cdot \max\left(0, m_p - d^T d'\right) + (1 - s) \cdot \max\left(0, d^T d' - m_n\right) \tag{7}$$

In Eq. (7), The hyperparameter λ_d is used to balance the positive counterpart loss and negative counterpart loss values within the descriptor, the hyperparameter m_p is the positive counterpart threshold, and m_n is the negative counterpart threshold.

III. Experiment

The operating system used in this paper is Windows 11, the language is Python 3.6 , the IDE is Pycharm, the deep learning framework is PyTorch, the hardware environment CPU is Intel Core i5-11400F, the GPU is NVIDIA GeForce 3060 12G. The hyperparameters in this paper correspond to thresholds in the positive direction $m_p = 1$, negative threshold $m_n = 0.2$, and epoch of 8 during training.

The comparison of the number of parameters is shown in Table I, SuperPoint is the unchanged network. SuperPoint(Gruops+Mish) indicates that the normal convolution in the original VGG-Style neural network in Encoder is changed to group convolution and the activation function is replaced with Mish function. The number of parameters is shown in the data in Table I. After using the final optimization method in this paper (SuperPoint(Gruops+Mish)), the number of network parameters is reduced by nearly 70%.

TABLE. I. COMPARISON OF NUMBER OF PARTICIPANTS

Method	Number of participants (million)
SuperPoint	130
SuperPoint(Gruops+Mish)	40

In this paper, the evaluation is performed on the Hpatches [14] dataset, which is a feature point and descriptor evaluation dataset released in 2017, by referring to the evaluation in the published SuperPoint paper. 2.5 million image blocks in 116 scenes are available inside the Hpatches dataset, of which 57 scenes are of large illumination changes and 59 The Hpatches dataset contains 2.5 million image blocks in 116 scenes, of which 57 scenes have large illumination changes and 59 scenes have large perspective changes. Each image block has 6 images with known single response, and each image block is annotated with true values. In this paper, we compare the effect of SuperPoint and this method on feature point detection and feature point matching, and the hyperparameter settings of the network are the same.

In this paper, we use repetition rate to judge the effectiveness of feature point detection and matching. The repetition rate is: the ratio of feature points that appear simultaneously to the total feature points in two images with changes in lightness or darkness, etc.

TABLE. II. COMPARISON OF FEATURE POINT DETECTION EFFECT

Method	Repetition rate（%）
SuperPoint	67.8
SuperPoint(Gruops)	63.9
SuperPoint(Gruops+Mish)	68.1

As shown by the data in Table II, the improved network, although simpler, has a small improvement of 0.3% in the feature point repetition rate.

To achieve the feature point matching effect comparison, firstly, the single-response change matrix of the two input images is obtained through interest points and descriptors. The process of obtaining the single-response matrix is shown as follows: first picture 1 and picture 2 generated by single-response transformation of picture 1 are sent into the network to generate interest points and descriptors respectively, and descriptors are matched by nearest-neighbor matching in a violent way, and the paired interest points and descriptors call the findHomography() function in OpenCV, and the method selects the RANSAC algorithm to generate the estimated single-response transformation matrix between the two single-strain transformation matrices for the estimation between images. Table III The single-strain estimation accuracy is: the number of image boundary corner points of image 1 after the real single-strain transformation matrix and the number of image boundary corner points of the estimated single-strain transformation matrix generated by the process of Figure 2 under a certain tolerance distance difference e as a proportion of the total number, and the single-strain estimation accuracy can reflect the feature point matching effect between images. The results in Table 4 show that the SuperPoint(Gruops+Mish) method decreases 0.03 at e=1, 0.01 at e=3, and 0.01 at e=5 compared with the original SuperPoint, which is a smaller decrease and does not lead to a significant decrease in the feature point matching effect.

TABLE. III. HPATCHES HOMOGRAPHY ESTIMATION.

Method	e=1	e=3	e=5
SuperPoint	0.34	0.73	0.84
SuperPoint(Gruops)	0.30	0.69	0.81
SuperPoint(Gruops+Mish)	0.31	0.72	0.83

Figure 5 shows the comparison between the original SuperPoint network, the network using SuperPoint(Gruops) and the final network implemented in this paper in terms of feature point detection and matching on the same graph, the red circles are the detected interest points, and the green lines are the matching interest points. The difference between the detected and matched interest points of the three networks is small, which proves that the detected interest points of this network are more accurate based on the reduced amount of parameters, and the matched interest points have a small improvement of 0.3% compared with the previous one.

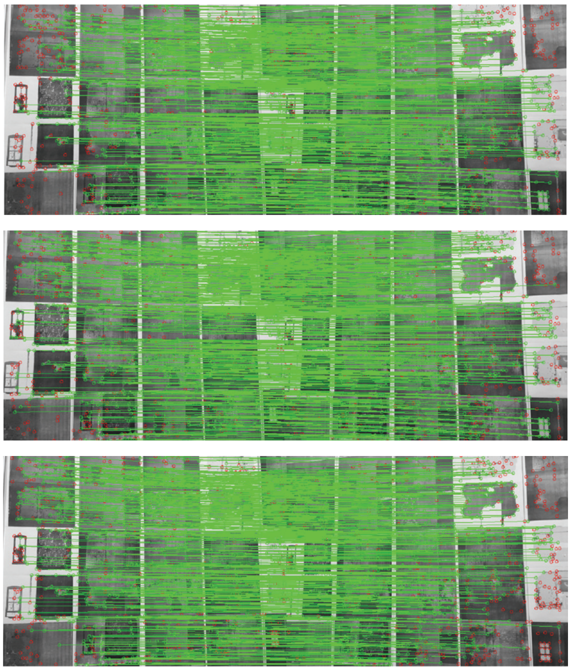

Fig. 5. Effect of feature point detection and feature point matching

IV. Conclusions

In this paper, we further optimize and simplify the network structure to address the problem that the SuperPoint network has a large number of parameters and operations, and does not perform well in real-time tasks. First, we apply the idea of grouped convolution to SuperPoint network, and replace the normal convolution in the shared encoder with grouped convolution to reduce the number of neural network parameters and make the network more lightweight. Then for the problem that the data fall into the negative interval becomes dead neurons leading to information interruption, the Mish function is used to replace the ReLU function to solve the problem. The experimental results show that the amount of network parameters is reduced by 70%, the accuracy of single-strain estimation only decreases by 0.01 when the tolerance distance difference is 3, and the repetition rate of feature point detection improves by 0.3%. The network in this paper is more lightweight than the previous network and achieves better results in feature detection and matching. The experiments prove that the network in this paper has better robustness.

The future work is to combine the network of this paper with the SLAM algorithm and replace the traditional feature point and descriptor extraction algorithm with the feature point and descriptor extraction network proposed in this paper to form a more robust SLAM algorithm.

References

[1] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," ACM Transactions on Graphics (TOG), vol. 25, no. 3, pp. págs. 835-846, 2006.

[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," IEEE transactions on robotics, vol. 31, no. 5, pp. 1147-1163, 2015.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature,* vol. 521, no. 7553, pp. 436-444, 2015.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision,* vol. 60, no. 2, pp. 91-110, 2004.

[5] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *European conference on computer vision*, 2006: Springer, pp. 404-417.

[6] Y. Tian, B. Fan, and F. Wu, "L2-net: Deep learning of discriminative patch descriptor in euclidean space," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 661-669.

[7] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," Advances in neural information processing systems, vol. 30, 2017.

[8] A. Barroso-Laguna, E. Riba, D. Ponsa, and K. Mikolajczyk, "Key. net: Keypoint detection by handcrafted and learned cnn filters," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5836-5844.

[9] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "LF-Net: Learning local features from images," *Advances in neural information processing systems,* vol. 31, 2018.

[10] M. Dusmanu *et al.*, "D2-net: A trainable cnn for joint description and detection of local features," in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2019, pp. 8092-8101.

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

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM,* vol. 60, no. 6, pp. 84-90, 2017.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[14] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "HPatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5173-5182.

[15] D. Misra, "Mish: A self regularized non-monotonic neural activation function," *arXiv preprint arXiv:1908.08681,* vol. 4, no. 2, p. 10.48550, 2019.