2021 IEEE 7th International Conference on Cloud Computing and Intelligent Systems (CCIS) | 978-1-6654-4149-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/CCIS53392.2021.9754617

ATKey.Net: Keypoint Detection by Handcrafted and Learned CNN with Attention

Zhihong Wang , Jinshan Ma , Haiyang He , Zixuan Wu , Changying Wang , Li Cheng*

College of Computer and Information Science of Fujian Agriculture and Forestry University, Fuzhou 350002, China
li.cheng@fafu.edu.cn

Abstract: In image matching, it is essential to obtain more stable and effective feature points. This paper proposes Attention Key.net (ATKey.Net) for the keypoint detection task. Handcrafted and Learned CNN filters are used in a shallow multi-scale architecture with an attention module. Handcrafted filters provide anchor structures for learned filters, which localize, score, and rank repeatable features. Learned CNN filters improve the stability and convergence during backpropagation. Shallow multi-scale architecture has fewer parameters and less computational cost. The attention module gives channel importance. The model is trained on ImageNet and evaluated on the HPatches benchmark. The results show that the repeatability and matching performance is better than the experimental detector.
Keywords: Handcrafted and learned CNN; Keypoint detection; Scale spaces; Image matching

1 Introduction

Image matching is the core task of target recognition, image mosaic, 3D reconstruction, visual positioning, scene depth calculation, and so on. The commonly used matching method is to obtain image feature descriptors. The most widely used traditional method is SIFT [1], which gets feature descriptors by manually setting arithmetic functions. With the development of deep learning in recent years, Convolutional Neural Networks (CNNs) [2, 3, 4, 5] can significantly reduce matching errors in local descriptors [6]. However, the advantage of learning methods over handcrafted ones has not been demonstrated in keypoint detection[7]. This paper integrates the attention module into Key.Net [8] and studies the network structure of different scales and channels.

Bahdanau D [9] proposed the attention module and used it in the translation model firstly, using the seq2seq + attention to solve translation and alignment problems. Since the attention module has no limitation on the input length, the input can be appropriate for decoding. In addition, the attention module allows the input representation to be separated from the output, so a hybrid encoder/decoder can be introduced. The most popular approach is to choose CNN as the encoder and RNN [10] or LSTM [11] as the decoder. However, RNN is sequential processing, resulting in low computational efficiency. After that, Shen et al. used temporal convolution to encode position information and the Transformer self-attention module [12]. The Transformer is parallel processing, training time is shorter, and translation accuracy is improved without any loop components. It also facilitates the sequential processing of interrupt input. Raffel and Ellis proposed a feedforward attention model [13]. It uses the attention model to fold the time of the data, replaces RNN with FFN, and generates a fixed-length content vector from a variable-length input sequence, which improves processing effectively. Since SENet [14] and CBAM [15] are representative attention models, this paper uses them in image matching.

In image matching, the local feature detector is based on engineering filters. In Gaussians [16], Harris-Laplace [21] or Hessian [17] uses Gaussian kernel function to get frequency and time to calculate image feature maps. It is just like training CNN. In other words, the process of traditional detectors can be imitated by learning appropriate parameter values in the convolution filter. However, in terms of measuring repeatability, compared with the local image descriptors based on CNNs, the improvement of the proposed manual detector based on CNNs has some inaccuracy [2, 3, 4, 5, 18, 19]. One reason is that usual channels are not distinguished. The robustness of the scale variable is also a problem, which is solved by the controlling variable. Other parameters, such as the dominant direction, can be described by CNNs. This paper adds multi-scale information in key. Net and uses an attention module in the channel to improve the accuracy of detection. The robustness of gradient variables is enhanced by controlling variables, and the structure is shown in Figure 1.

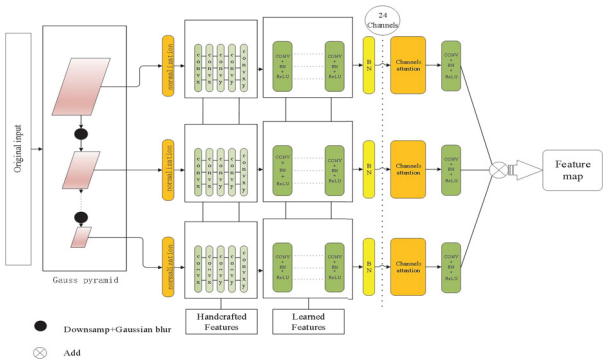

Figure 1 ATKey.Net network

ATKey.Net includes original image input, Gaussian

978-1-6654-4149-0/21/$31.00 ©2021 IEEE

pyramid [1], Handcrafted Features [20], Learned Features, and Feature map output. Gauss pyramid is responsible for the multi-scale transformation of the full-size input image. Down-sampling rate is 1.2, and down-sampling the image four times continuously to obtain five-scale images. Handheld features [4] filters the key points from height and width to complete the first detection. In the filter of Learned Features, each convolutional layer has n (set to 8 in the experiment) convolution kernels. The second detection of key points is completed by convolution of the whole image. The output is 24 channels. When training multi-scale and multi-channel CNNs, use average pooling and maximum pooling [15] to determine the channels that need special attention and extract more stable key points. Finally, five scales are accumulated to return the feature map containing the key point score of each pixel. HPatches [2] benchmark is used to test the model from different angles and lighting conditions. In an ideal state, the detector can extract the same features from images with various geometric or photometric transformations, and there will be errors in actual situations. Many previous works are studying how to reduce false matching. Atkey. Net focuses on how to make the detector extract more accurate key points.

The contributions of this paper are as follows: a) In CNN, a multi-channel attention module is combined with the manual method to design the image keypoint detector. b) Optimize the convolution kernel, improve the calculation efficiency, enhance the nonlinearity, and avoid over-fitting. c) Fusion of Gaussian pyramid and siamese network structure.

2 Related Work

2.1 Attention

The importance of each channel in CNN is different. Therefore, an attention module can be introduced to represent the importance of other channels. Senet (squeeze and exception networks) [14] is a widely used channel attention module model, in which the SE module is simple, easy to implement, and loads into the existing network framework. SENet learns the correlation between channels, and a slight increase in the amount of calculation for important channels will extract more accurate features. Then came the Convolutional Block Attention Module (CBAM) [15], a simple and effective forward convolutional attention module neural network. The Channel Attention Module (CAM) [15] of CBAM optimizes the SENet channel attention module. This method gives an intermediate feature map. This module infers the attention map in turn along two dimensions, namely channels, and then multiplies the attention map with the input feature for adaptive feature refinement mapping. Because CBAM is a lightweight, general-purpose module, it can be seamlessly integrated into any CNN with negligible overhead. It is used in this paper to distinguish channels with attention coefficient.

2.2 Descriptor

The descriptor is roughly divided into traditional and deep learning methods. Traditional methods include SUSAN [22], FAST [23], BRIEF [24], Laplacian of Gaussian (LoG) [25], Difference of Gaussian (DoG) [26], Harris [21], SIFT [1] and ORB [27] etc. Deep learning methods include SuperPoint [28], Deep Convolution Feature Point Descriptor (DeepDesc) [29], Learning-based Invariant Feature Transformation (LIFT) [30], HardNet [31] and SOSNet [32]. SUSAN uses a circular template to move in the image and judges the feature points according to the changes in the gray pixel value within the circle. The algorithm has a certain accuracy. FAST is similar to it. It judges whether the pixel value of the circle is the same as that of the center point. The algorithm is fast but susceptible to noise interference and does not have scale invariance. BRIEF is a feature descriptor, which describes the detected features. It is a binary code descriptor [24], which can be executed only with the feature point method. LoG and DoG transform image matching into a Gaussian problem and determine the inflection point by methods such as derivation. DoG saves a lot of computing power than LoG. The Harris corner detection judges by comparing the pixel changes before and after the sliding fixed window. SIFT uses Gaussian pyramids to obtain image features and uses a one-dimensional vector representation. The feature vector has features that are invariant to image scaling, translation, and rotation. It also has certain invariance to illumination, affine, and projection transformations, which is a perfect one. It is a very excellent local feature description algorithm. ORB (Oriented FAST and Rotated BRIEF) is a fast feature extraction and description algorithm. The fast feature point detection method can be combined with the BREF feature descriptor method. FAST is used to extract features in the Gaussian pyramid. BREF is used to change the coordinate origin position with scale invariance and direction invariance.

DeepDesc was the first to use neural networks to train learnable feature descriptions [29], but the division of labor for each CNNs part is unclear. The later LIFT has three parts of CNNs, and the spatial conversion layer is used to perform an affine transformation on the image. Modify the image block to achieve image feature point detection, direction estimation, and descriptor extraction. For the pre-training of feature points, it isn't easy to set the groundtruth with the artificial annotation in the traditional method, so a self-supervised method SuperPoint is proposed. If there are only line segments, triangles, rectangles, cubes, etc., in an image, then the feature points are at their endpoints or vertices. The image is trained by rendering to obtain the basic shape, and the essential shape elements are used to generate the training. Set and truth values can extract the location of feature points and descriptors at the same time [33]. This method has an excellent performance in both virtual and natural scenes, but there is no particular emphasis on changing the angle when changing the angle of view. SOSNet (Second Order Similarity network), a network model that uses second-order similarity as part of its

constraints, focuses on angle transformation. Generally, a pair of positive matching points have similar distances to other points in the embedding space [32]. Following this, use second-order similarity as a constraint to further optimize loss, and train the network model to learn more and better descriptors. Key.Net uses the Gaussian differential function to identify potential SIFT points of interest with invariant scale and rotation, convert them into a loss function as part of the constraints, and apply them to deep learning. Based on the basic structure of DenseNet and L2-net, the constraint is based on the square difference between the points extracted by the IP (index suggestion) layer and the actual maximum coordinates in the corresponding window. At the same time, the IP layer is extended to multiple scales. The multi-scale loss function is the average value of the covariant constraint loss for all scale levels. The improved model performs well under the MMA evaluation index. Because the light changes in space can be adjusted by manual intervention, but the viewing angle changes cannot be done manually, this article improves the Key.Net+HardNet [4] network. It uses SOSNet to replace HardNet to extract feature descriptors, which significantly enhances the MMA evaluation index.

3 Method

3.1 Gaussian Pyramid

The SIFT uses a Gaussian kernel to filter when constructing the scale-space. The original image has the most detailed features and simulates the feature representation in large-scale situations by reducing the detailed features after Gaussian filtering. The Gaussian kernel is the only scale-invariant kernel function, and the DoG kernel function can be approximated as an LoG function, making feature extraction easier. This paper uses a five-layer Gaussian pyramid [1], the five-layer input is set with the same weight coefficients, and the SIFT scale space is used. The downsampling factor is 1.2 for four times of downsampling and then output to siamese network [34,35]. It is shown in Figure 1.

3.2 Attention module

The channel attention module of CBAM is different from that of senet. The cam channel attention module in CBAM processes the feature map obtained by convolution, obtains a one-dimensional vector with the same number of channels as the evaluation score of each channel, and assigns the score to the corresponding channel, respectively. In the SENet structure, only global pooling is used, and too much information will be lost. This paper introduces the channel attention module and adopts the parallel method of maximum global pooling and global average pooling [14,15] to obtain more helpful information. The process of extracting features has been significantly improved in the MMA evaluation index. By replacing the large convolution kernel with multiple small convolution kernels, the training time on the ImageNet dataset was successfully shortened. Use SIFT, SURF, LIFT, HardNet, SOSNet feature extraction methods, and Resnet-50 to test the HSequences evaluation index.

In CNN, this experiment uses two-dimensional images. The same input passes through different convolution kernels to get various features. The CBAM channel attention module is used during fusion to emphasize the importance of features. The structure of the channel attention module is shown in Figure 2.

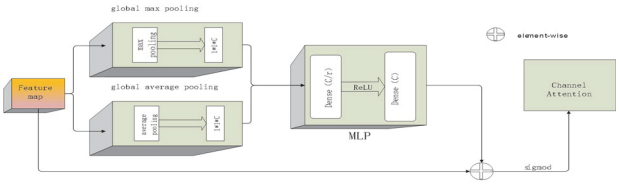

Figure 2 Channel attention

The input of Figure 2 is the Feature map (H × W × C), H is height, W is width, and C is the number of channels. The Feature map is subjected to global max pooling based on width and height, respectively. Global average pooling (global average pooling), get two 1 × 1 × C feature maps, send them to a two-layer neural network (MLP), the number of neurons in the first layer is C/r, and r is the reduction rate. The activation function is ReLU. The number of neurons in the second layer is C. The element-wise addition operation is performed on the output of the feature by the MLP, and the final Channel Attention Feature is generated after sigmoid activation, namely M_c [15]. The calculation of the channel attention module is as follows:

$$M_c(F) = \sigma(MLP(AugPool(F)) + MLP(MaxPool(F))) \quad (1)$$

$$= \sigma(W_1(W_0(F_{avg}^C)) + W_1(W_0(F_{\max}^C))) \quad (2)$$

Where σ denotes the sigmoid function, $W_0 \in R^{C/r*c}$, and $W_1 \in R^{C*C/r}$. Note that the MLP is weight W_0 and W_1 is shared for both inputs. The ReLU activation function is followed by W_0.

3.3 Convolution kernel and feature extraction method

The image search range in a complex environment has a large span. 5 × 5 and 7 × 7 convolution kernels are used in the context module to increase the receptive field proportional to the stride of the corresponding layer. After learning the VGG network structure, multiple 3 × 3 convolution kernels are used instead of 5 × 5 and 7 × 7 convolution kernels [36] to reduce the number of parameters and the computational cost. In the case of an even number of convolution kernels, a symmetrical increase of padding cannot guarantee the size of the input and output feature maps. An odd number of convolution kernels are used. The calculation amount of the input image is computed as:

$$f(x) = O_{time}(i) \quad (3)$$

O_{time} Indicates the time required to calculate i.

If 5 × 5 and 7 × 7 are regarded as large convolution kernels, and 3 × 3 is considered to be small convolution kernels, then the calculation amount proportion of large and small convolution kernels is as follows:

$$\mathrm{p} = \frac{O_{time}(a)}{O_{time}(\mathrm{b})} = \frac{F_a * F_a}{F_\mathrm{b} * F_\mathrm{b} * n} \quad (4)$$

a is a large convolution kernel, b is a small convolution kernel, F_a is the size of a large convolution kernel, F_b is the size of a small convolution kernel, and n is the number of convolution kernels.

The small convolution kernel can integrate more than one nonlinear activation layer to replace a single nonlinear activation layer, increase the nonlinear fitting ability, reduce network parameters and calculations, and control the rise and fall of the number of channels. When creating a network structure, multiple small convolution kernels in series can increase the flexibility of setting the numbers of channels. In short, set the size of the convolution kernel to 3.

After the input image passes through the ATKey.Net network, the sub-response map of the key point of each pixel is returned. The output needs to be further extracted and matched. SOSNet is used to extract the features and compare them with ResNet-50 and improved on MMA(Mean Matching Accuracy).

4 Experiment

This section presents implementation details, metrics, and the dataset used for evaluating the method.

4.1 Training data

The experiment uses a comprehensive training set ILSVRC 2012 in ImageNet. We apply a random geometric transformation to the image and extract the corresponding region pair as the training set. Set the transformation parameter scale to [0.5,3.5], tilt to [-0.8,0.8] and rotation to [-60°, 60°]. Since the untextured area is not discernible, we identify it by checking whether the response of any handmade filter is below the threshold. We modify the contrast, brightness, and hue value in HSV space to one of the images to improve the network's robustness against illumination changes. In addition, for each pair, we generate binary masks that indicate the common area between images. The data set has 12000 images, and the size of each image is 192*192, of which 9000 images are used as training data, and 3000 images are used as verification sets.

4.2 Evaluation

Follow the evaluation scheme proposed in the literature [37]. The repeatability score of a pair of images is the ratio between the number of corresponding key points and the number of lower key points detected in the two images. The number of extracted keypoints remains the same to compare different methods and allows each keypoint to be matched only once [38,39]. The intersection ratio IoU between the two candidate data regions is calculated to determine the corresponding key points. When IoU is less than 0.4, use the first 1000 points of interest that belong to the common area between the images. That is, the overlap between the corresponding areas exceeds 60%. The HPatches [20] data set is used for testing. HPatches contain 116 sequences divided between viewpoint and illumination transformation, respectively, 59 and 57 sequences. HPatches provides pre-defined image patches for evaluating descriptors. Unlike HPatches, a complete image is used to evaluate the keypoint detector.

4.3 Evaluation Metrics

Training is performed in a siamese pipeline, with four instances of ATKey.Net that share the weights and are updated simultaneously. Each convolutional layer has M = 8 filters of size 7 × 7,5 × 5 or 3 × 3, with weights initialization and L2 kernel regularizer. Set the batch size to 32, the learning rate of the Adam optimizer is 0.001, and the attenuation factor after 30 cycles is 0.7. On average, the architecture converges within 1500 epochs and runs on an Ubuntu18.04.1, Linux5.4.0 system, an i7-10700k CPU operating frequency of 3.8GHz, and an NVIDIA GeForce gtx2080ti machine.

4.4 Results and analysis

In the ATKey.Net structure, if the equivalent weighted evaluation of different channels is adopted, the obtained feature maps have deviation, and various features have different degrees of importance. Channels corresponding to essential features should be strengthened, and unimportant channels should be weakened. The number of channels is split to 24, and the attention module is added. The experimental results also prove effective.

Table I Repeatability for translation and scale invariant detectors on HPatches

	Viewpoint				Illumination			
	Repeatability(%)		IoU		Repeatability(%)		IoU	
	SL	L	SL	L	SL	L	SL	L
SIFT-SI[1]	43.4	59.1	0.18	0.12	47.7	61.1	0.18	0.12
SURF-SI[40]	46.7	60.3	0.18	0.18	53.0	64.0	0.15	0.11
LIFT-SI[30]	43.6	59.9	0.20	0.13	51.5	66.1	0.18	0.12

Key.Net-TI[4]	34.2	71.5	0.20	0.11	72.0	72.0	0.10	0.10
Key.Net-SI[4]	60.5	73.2	0.19	0.14	61.3	66.2	0.12	0.10
ATKey.Net-TI	35.1	74.3	0.20	0.11	74.7	74.7	0.11	0.11
ATKey.Net-SI	55.5	67.8	0.18	0.09	63.4	67.3	0.10	0.09

Table I: Repeatability results for translation (TI) and scale (SI) invariant detectors on HPatches. We also report average overlap error IoU. In SL, scales and locations are used to compute overlap error, meanwhile, in L, only locations are used, and scales are assumed to be correctly estimated. In the illustration, ATKey.Net is improved, and in viewpoint, Ti is also enhanced.

In this paper, rtx2080ti is used for experiments. The training time before the convolution kernel size is not changed is 18 hours, and after the change is 13 hours, which is 5 hours more than before the change (the time is rounded).

Table II Comparison of training time for different convolution sizes

Convolution size	Time(h)
3×3	13
5×5	18
7×7	18

In theory, the training time can be shortened by replacing the large convolution block with the small convolution block. When the 3×3 convolution kernel replaces the 7 × 7 convolution kernel, the size of the convolution kernel is reduced by more than half, but the running time is reduced by less than half. After analysis, the convolution kernel of 7×7 only needs one weighted sum to output 1×1. The convolution kernel of 3×3 is filtered into 5 × 5, then becomes 3 × 3 through the first 3 × 3 convolution kernel, and becomes 1 × 1 through the second 3×3 convolution kernel. The convolution kernel of 3×3 has nine parameters and convolutes three times. That is, there are 27 parameters in total. The convolution kernel of 7 × 7 has 49 parameters. Although the parameters become less after changing the size of the convolution kernel, the operation steps increase in the process of weighted summation. In computer operation, multiplication is obtained by addition so that the addition will be faster than multiplication and division.

The HardNet and SOSNet extraction feature comparison experiments are carried out to verify the improvement of the average matching accuracy of ATKey.Net compared to Key.Net.

Table III Average matching accuracy

MMA(%)	Viewpoint		Illumination	
	Ti	Si	Ti	Si
Ket.Net[4]+Resnet50[41]	3.1	6	4.11	6.5
Key.Net[4]+HardNet[31]	43.4	46.45	49.71	48.48
Key.Net[4]+SOSNet[32]	43.77	46.16	49.26	48.12
ATKey.Net+HardNet[31]	45.45	45.57	50.42	46.36
ATKey.Net+SOSNet[32]	45.77	45.1	49.95	45.8

Table III： Comparison of average matching accuracy. The experiment compares the average matching accuracy of Key.Net and ATKey.Net extracted features. After Key.Net, SOSNet performs best under TI in viewpoint, ATKey.Net+HardNet and ATKey.Net+SOSNet Outstanding performance in TI and SOSNet in view is also improved compared to HardNet. At the same time, comparing with Resnet50 proves that simply increasing the network depth cannot achieve good results.

In the HardNet network structure, a new loss [31] is proposed for feature metric learning. The proposed loss can maximize the distance between the nearest positive and negative samples in a training batch, affecting shallow and deep CNN networks. The outstanding contribution of SOSNet is to integrate the second-order similarity into the similarity judgment. Since the second-order similarity principle is the similarity measurement between nodes in the network graph [32], when the view is changed, the change of the pixel node will be more Highlight and accurately constrain so that the matching results will be improved in different view situations.

5 Conclusions

We designed a new method to detect the local features of the target and reduce the mismatch rate of the image. It combines the traditional way and the CNN filter with the attention module. In the Gaussian Pyramid and Handcrafted Features modules, multi-scale and human prior knowledge in conventional methods are added. The

channel attention module is added after the Learned Features module to increase the model's attention to important channels. The siamese network shares the weight coefficients. The results obtained from multiple scales are superimposed, and the response map containing the key point score of each pixel is output.

By optimizing the convolution kernel to shorten the training time and verifying the large benchmark HPatches, the repeatability score is improved compared with other feature point detection methods. Experiments have proved that the improved method also improves the matching accuracy. Using the returned response map of the key point score of each pixel and different descriptors for experiments, compared to the previous research, improves the MMA score.

However, there is still a part of this research that is worth continuing to study. In the experiment, it can be seen that the scores are higher under the evaluation index of changing the angle of view, and the scores will be lower under the conditions of different light intensities. The scores will be higher under the conditions of varying light intensities. Under the condition of changing perspectives, the score is lower. At present, the author has not found a method that is both in the lead. It is worthwhile for those interested in this field to continue research. You can start with the Handcrafted Features module and the Learned Features module. Traditional methods Human experience is not fully involved in the model, but instead of traditional methods, it may be better to add structural relationships to learnable modules.

Acknowledgments

The work was supported by the Research on the Information Construction of Classroom Teaching Management under Grant No. 322/111418018 (Key Project of Teaching Reform of Fujian Agriculture and Forestry University in 2018), "Research on the Deep Learning Model of Cross-category Fruit Appearance Evaluation" (Wuyi University Cognitive Computing and Intelligent Information Processing, Fujian Provincial Key Laboratory of Colleges and Universities Open Project, Project Number: KLCCIIP2020201), Fujian Province University, Fuzhou, Fujian Province 350002, PR China, and The Research on Pixel Coordinate Calibration Method for Video by Multi-Mobile Terminal Collaboration.

References

[1] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 2004, 60(2): 91-110.

[2] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, et al. Lift: Learned invariant feature transform. ECCV, 2016: 467-483.

[3] Karel Lenc, Andrea Vedaldi. Learning covariant feature detectors. ECCV, 2016: 100-117.

[4] Zhang Xu, Yu Felix X., Karaman Svebor et al. Learning discriminative and transformation covariant local feature detectors. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: 4923-4931.

[5] Yuki Ono, Eduard Trulls, Pascal Fua, et al. Yi.LF-Net: Learning Local Features from Images. NIPS, 2018: 6237-6247.

[6] Balntas Vassileios, Lenc, Karel, Vedaldi, et al. HPatches: A Benchmark and Evaluation of Handcrafted and Learned Local Descriptors. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: 3852-3861.

[7] Karel Lenc, Andrea Vedaldi. Large scale evaluation of local image feature detectors on homography datasets. BMVC, 2018: 1807.07939.

[8] Laguna A B, Riba E, Ponsa D, et al. Key.Net: Keypoint Detection by Handcrafted and Learned CNN Filters. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019: 5835-5843.

[9] Bahdanau Dzmitry, Cho Kyunghyun, Bengio, Yoshua. Neural Machine Translation by Jointly Learning to Align and Translate. Computer Science, 2014: 1409.0473.

[10] Cho K, Van Merrienboer B, Gulcehre C, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Computer Science, 2014: 10.3115/v1/D14-1179.

[11] Gers Felix A, Schmidhuber Jurgen, Cummins Fred. Learning to Forget: Continual Prediction with LSTM. Neural computation, 2000, 12(10): 2451-2471.

[12] Tao Shen, Tianyi Zhou, Guodong Long, et al. DiSAN: Directional Self-Attention Network for RNN/CNN-Free Language Understanding. Computer Science, 2017: 1709.04696.

[13] Colin Raffel, Daniel PW Ellis. Feed-Forward Networks with Attention Can Solve Some Long-Term Memory Problems. CoRR. 2015: 1512.08756.

[14] Hu Jie, Shen Li, Albanie Samuel, et al. Squeeze-and-Excitation Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(8): 2011-2023.

[15] Woo Sanghyun, Park Jongchan, Lee Joon-Yong, et al. CBAM: Convolutional Block Attention Module. Springer International Publishing. 2018: 3-19.

[16] Wong, M. W.. Gaussians. Birkhäuser Basel. 2002: 129-140.

[17] Krystian Mikolajczyk, Cordelia Schmid. Scale & affifine invariant interest point detectors. International Journal of Computer Vision, 2004, 60(1): 63-86.

[18] Roska T., Chua L.O.. The CNN universal machine: an analogic array computer. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 2015, 40(3): 163-173.

[19] Yi Kwang Moo, Verdie Yannick, Fua Pascal, et al. Learning to assign orientations to feature points.2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016: 107-116.

[20] Schönberger Johannes L., Hardmeier Hans, Sattler Torsten, et al. Comparative Evaluation of Hand-Crafted and Learned Local Features. IEEE Conference on Computer Vision & Pattern Recognition. 2017: 6959-6968.

[21] Chris Harris, Mike Stephens. A combined corner and edge detector. In Proceedings of the Alvey Vision Conference. UK. 2 August-2 September. 1988. 15: 593–600.

[22] Tissainayagam P, Suter D . Assessing the performance of corner detectors for point feature tracking applications. Image & Vision Computing, 2004, 22(8):663-679.

[23] Rosten Edward, Drummond Tom. Machine learning for

high-speed corner detection. European Conference on Computer Vision. 2006. 3951: 430–443.

[24] Michael Calonder, Vincent Lepetit, Christoph Strecha et al. BRIEF: Binary Robust Independent Elementary Features. European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2010. 6314: 778-792.

[25] LONG Peng, LU Huaxiang. Weighted guided image filtering algorithm using Laplacian-of-Gaussian edge detector. Journal of Computer Applications. 2015. 35(9): 2661-2665.

[26] Krystian Mikolajczyk, Cordelia Schmid. Indexing based on scale invariant interest points. ICCV, 2001. 1: 525-531.

[27] Ethan Rublee, Vincent Rabaud, Kurt Konolige et al. ORB: an efficient alternative to SIFT or SURF. IEEE International Conference on Computer Vision, Barcelona, Spain, 6-13 November, 2011: 2564-2571.

[28] Daniel DeTone, Tomasz Malisiewicz, Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 2018: 337-33712.

[29] Simo-Serra Edgar, Trulls Eduard, Ferraz Luis et al. Discriminative Learning of Deep Convolutional Feature Point Descriptors. IEEE International Conference on Computer Vision, 2015: 118-126.

[30] Zhang Minling, Wu Lei. Lift: Multi-Label Learning with Label-Specific Features. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2015. 37: 107-120.

[31] Anastasiya Mishchuk, Dmytro Mishkin, Filip Radenović et al. Working hard to know your neighbor's margins: Local descriptor learning loss. Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017:4829 – 4840.

[32] Yurun Tian, Xin Yu, Bin Fan et al. SOSNet: Second Order Similarity Regularization for Local Descriptor Learning. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, CA, USA. 2019: 11008-11017.

[33] Daniel DeTone, Tomasz Malisiewicz, Andrew Rabinovich. Deep Image Homography Estimation. Computer Vision and Pattern Recognition. 2016: 1606.03798.

[34] Sumit Chopra, Raia Hadsell, Yann LeCun. Learning a Similarity Metric Discriminatively, with Application to Face Verification. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2005.1: 539-546.

[35] Norouzi M, Fleet D, Salakhutdinov R, et al. Hamming Distance Metric Learning. Advances in Neural Information Processing Systems, 2012, 2: 1061-1069.

[36] Karen Simonyan, Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. Computer Science. 2014: 1409.1556.

[37] Krystian Mikolajczyk, Cordelia Schmid. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2005. 27(10): 1615-1630.

[38] Yannick Verdie, Kwang Moo Yi, Pascal Fua et al. Tilde: a temporally invariant learned detector. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015: 5279-5288.

[39] Edward Rosten, Reid Porter, Tom Drummond. Faster and better: A machine learning approach to corner detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010. 32(1): 105-119.

[40] Herbert Bay, Tinne Tuytelaars, Luc Van Gool. SURF: Speeded up robust features. Computer vision – ECCV2006. Graz, Austria. 2006: 404-417.

[41] Abbas Jafar, Myungho Lee. Hyperparameter Optimization for Deep Residual Learning in Image Classification. IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C). Washington, DC, USA. 2020: 24-29.