IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 23, 2020, accepted November 4, 2020, date of publication November 10, 2020,
date of current version November 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3037154

D-MSCD: Mean-Standard Deviation Curve Descriptor Based on Deep Learning

ZHANQIANG HUO, GUANGXING DU, FEN LUO, YINGXU QIAO, AND JUNWEI LUO
School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454003, China

Corresponding author: Junwei Luo (luojunwei@hpu.edu.cn)

This work was supported in part by the Henan University Scientific and Technological Innovation Team Support Program under Grant 19IRTSTHN012, and in part by the Foundation of Henan Scientific and Technological Project under Grant 192102210281.

ABSTRACT Curve feature description is an important issue in the field of image matching. In the past years, this problem has been studied mainly based on handcrafted methods. To conquer the disadvantages of low discrimination and weak robustness of curve feature description under complex conditions, a Mean-Standard Deviation Curve Descriptor based on Deep learning (D-MSCD) is proposed in this paper. Firstly, a large-scale curve feature dataset with 210,000 labeled curve image patches is constructed for training and testing. After longitudinally compressing the support areas of the curve in each image into the support areas of points, the mean and standard deviation image patches of each curve are obtained, then the curve image patch is uniquely represented. Secondly, a modified L2-Net(DSM) which is a network architecture with dilated convolution is constructed to improve the performance of curve descriptors, and the experimental results on the Brown dataset show the mean FPR95 value is reduced by 17.48%. Finally, the modified L2-Net(DSM) is trained on the large-scale curve feature dataset and the model of D-MSCD is obtained, it achieves the best matching performance in every image change, and the average matching performance on the Oxford dataset is improved by 13.09%. Experimental results demonstrate the proposed D-MSCD has better effectiveness than the traditional handcrafted curve descriptors.

INDEX TERMS Computer vision, deep learning, curve descriptor, a large-scale dataset.

I. INTRODUCTION

Feature description is a key technology in the fields of computer vision, which has considerable applications in image retrieval [1], scene recognition [2], [3], and 3D reconstruction [4], [5], etc. Curve feature description is an important process of image feature matching, and the performance of descriptors has a direct impact on feature matching. Therefore, the study of robust feature description methods has received lots of researchers' attention.

The traditional curve feature description methods are handcrafted, which are based on the experience accumulation and design inspiration of the researchers, and the desired structural features of the image to be detected are formalized through appropriate mathematical tools to obtain corresponding feature description. Various handcrafted methods have been proposed for curve matching in recent years [6]–[10], but these methods have the disadvantages of low distinguishability and weak robustness under complex conditions. Therefore, it's necessary to use a new method for curve feature description.

With the continuous success of deep learning in image recognition tasks, the current research on image feature matching has entered a new data-driven era. In recent years, attempts on using deep learning for image feature description and matching have also shown great opportunities [11]–[18]. However, there are no reports on the method using deep learning to describe curve features at present. One reason is that deep learning requires a large amount of annotated data, another reason is that there is no neural network architecture suitable for curve feature training. Therefore, the key to using deep learning for curve feature description is to transform the problem of curve feature description into a deep learning problem.

This paper attempts to use the convolutional neural network to learn the curve characteristics. Specifically, we convert the curve feature description problem into the mean and standard deviation of point feature description and propose

The associate editor coordinating the review of this manuscript and approving it for publication was Emre Koyuncu.

a method for describing the curve feature based on deep learning to achieve reliable curve feature matching. Compared with traditional methods, the contributions of this paper are as follows: 1) A large-scale image curve dataset labeled with matching information is constructed for network training and testing. 2) Improved the latest L2-Net (DSM) [17] and used it to train the dataset. 3) Proposed a feasible curve feature descriptor based on deep learning, and the experimental results demonstrate that the proposed D-MSCD has better matching performance compared with the traditional handcrafted descriptors.

The remainder of this article is structured as follows: Section II describes the related work. Section III elaborates on the proposed method in detail. Experimental results are demonstrated in section IV, while the conclusion is in Section V.

II. RELATED WORK

A. TRADITIONAL CURVE FEATURE DESCRIPTOR

Curve feature description plays an important role in image feature description and it has attracted considerable attention from scholars. The most classical feature descriptor is the SIFT [6], which is proposed by Lowe. It is based on the gradient distribution in the detected regions and is invariant to scale rotation and viewpoint change. Wang *et al.* [7] proposed the Mean-Standard Deviation Line Descriptor (MSLD) based on the idea of neighborhood location division of SIFT and extended it to the curve description to obtain the Mean-Standard Deviation Curve Descriptor (MSCD). The MSCD successfully solved the problem of a unified description of lines of different lengths. However, when the viewing angle changes, the image deformation will distort the shape of the region, which can lead to the decline of the matching ability of the descriptor. Liu *et al.* [8] divided sub-region according to the overall intensity order and the local intensity order mapping, and proposed Intensity Order Curve Descriptor (IOCD), which performs robustly on image rotation, viewpoint change, illumination change, blur change, noise change, and JPEG Compression change. However, when the image has shadows and partial occlusions, there will be wrong sub-region divisions in the brightness sequence division, resulting in incorrect matching. Wang *et al.* [9] combined the idea of intensity order division with MSCD and proposed the Intensity Order Based Mean-Standard Deviation Descriptor (IOMSD), the principle of this algorithm is simple and stable, but the description performance is not high and it is not suitable for weakly textured images. Liu *et al.* [10] proposed the Gradient Order Curve Descriptor (GOCD), which is constructed based on a global gradient magnitude order for sub-region division and local gradient order feature. However, since the radius of the pixel support region and the stride for computing the gradient magnitude around a feature point are fixed, it is not invariant to large scale change. These traditional methods can be widely used for image registration, optical image matching with different geometric and photometric transformations such as scale, rotation, blur, illumination, and JPEG compression, and textured scenes images. However, these methods have some limitations due to the intensity distribution and varying illumination which are caused by noise.

B. DEEP LEARNING-BASED LOCAL FEATURE DESCRIPTOR

With the booming of handcrafted descriptors in the past decades, more and more deep learning-based descriptors have appeared. Han *et al.* [12] proposed MatchNet, which consists of a featured network for extracting feature representation, a bottleneck layer for reducing feature dimension, and a metric network for measuring the similarity of features pairs. It shows great potential for deep learning in local feature descriptions. Balntas *et al.* [13] proposed to use the distance relationship between a pair of negative samples and positive samples that are more difficult to distinguish in the triplet for CNN network training. And a good feature matching result is achieved with only 2 convolutional layers. Tian *et al.* [14] proposed L2-Net, which used a fully convolutional network structure for feature descriptor learning, and compared the distance of positive samples with the distance of all negative samples during training. This greatly outperformed the performance of previous methods. Mishchuk *et al.* [15] proposed HardNet, which only considered the relative distance between positive samples and the most difficult negative samples in a batch of training data when training. This further improved the matching performance of L2-Net. Zhang and Rusinkiewicz [17] proposed a new triplet loss based on HardNet, which replaced the hard margin with dynamic soft margin, and got a better matching performance. Tian *et al.* [18] proposed the Second Order Similarity Regularization (SOSR) and incorporated second order similarities into the learning of local descriptors. The matching performance of learning descriptors is significantly improved. These methods of using deep learning for image local feature description show us the possibility of using deep learning for curve feature description.

III. PROPOSED METHODOLOGY

Different from traditional handcrafted descriptors which are mostly driven by intuition or researcher's expertise, deep learning-based methods are driven by data. So we constructed a large-scale image curve dataset labeled with matching information for network training and testing.

A. THE CURVE FEATURE DATASET

1) IMAGE PAIRS

We created about 1,700 sets of image pairs with a size of 640 × 480 pixels through Internet downloading and mobile phone shooting, most of them are buildings and cultural relics that contain more curves. These images include seven types of changes, namely, **Scale, Illumination, Blur, Viewpoint, Rotation, Compression, and Noise**.

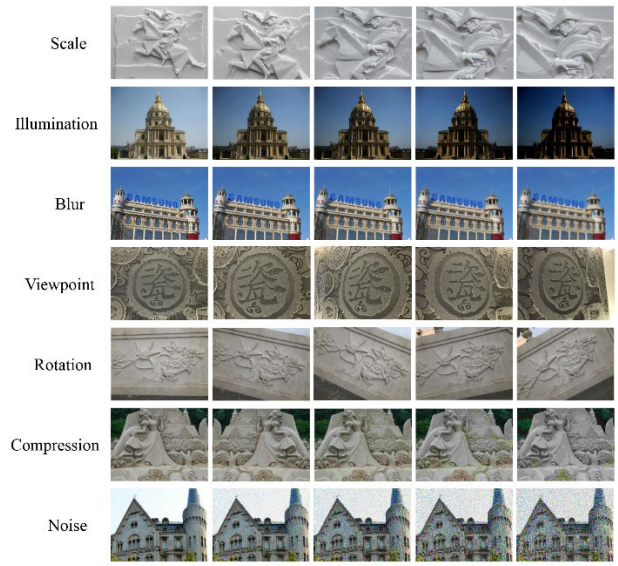

FIGURE 1. The example of image pairs with different transformations.

As can be seen in Fig. 1, the first image in each row is the reference image, and the rest are the target images of different transformation degrees, each target image and reference image form an image pair.

To produce the image pairs under different transformations, the MatlabR2015b and Photoshop CC2018 are used, and the following operations are performed:

a: SCALE

The reference image of each group is downloaded from the Internet, the target images are obtained by using Photoshop to crop the reference image in different degrees and then enlarge to the size of the reference image.

b: ILLUMINATION

The reference image of each group is downloaded from the Internet, the target images are obtained by adjusting the illumination of the reference image with the curve tool in Photoshop.

c: BLUR

The reference image of each group is downloaded from the Internet, the Matlab function 'fspecial' is used to generate the blurred target images on the reference image, the types of the function are set to 'average', 'gaussian' and 'disk'.

d: VIEWPOINT

The reference image and the target images in each group are all taken with a mobile phone from different perspectives.

e: ROTATION

The reference image and the target images in each group are all taken with a mobile phone from different perspectives.

f: COMPRESSION

The reference image of each group is downloaded from the Internet, the target images are obtained by compressing the reference image through a program, in which the compression ratios are set to 75%, 85%, 90%, and 95% respectively.

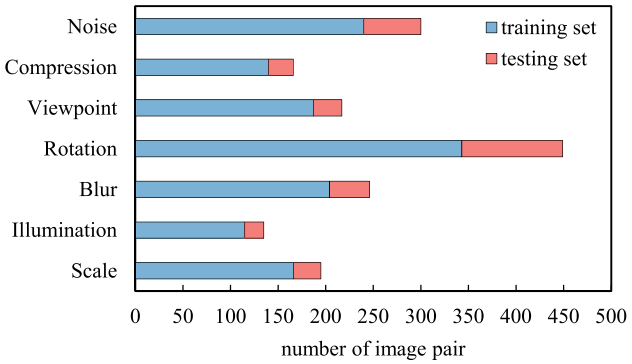

FIGURE 2. The number of image pairs under different transformations in the training set and testing set.

g: NOISE

The reference image of each group is downloaded from the Internet, the Matlab function 'imnoise' is used to generate the target images on the reference image, the types of the function are set to 'salt & pepper' and 'gaussian'.

We divided the image pairs into training set and testing set, the training set contains 1395 pairs of images and the testing set contains 313 pairs of images. Fig. 2 shows the number of image pairs in the training set and testing set under different transformations.

2) CURVE FEATURE DATASET

Canny edge detection operator [19] is used to extract the curve of the image pair, as well as filter the points with curvature greater than 0.8 and eliminate the curves with length less than 20 pixels. For each image pair, the IOCD [8] is used to obtain the curve matching result. To improve the accuracy of the matching result, artificial culling is used to delete the wrong matching to obtain the correct matching curve pair in the image pair.

We use the local image patches around the curve to characterize the curve, the neighborhood of the curve is transformed into a square image patch independent of the curve length. For any curve C composed of $Num\,(C)$ points, the pixel on C is denoted as P_k, $k = 1, 2, \ldots, Num\,(C)$. The image patch $I\,(P_k)$ along the gradient direction with the length and width of 64 pixels centered at P_k is extracted as its local neighborhood. Then, the mean matrix $M\,(C)$ and standard deviation matrix $S\,(C)$ of the local neighborhoods of all the pixels are calculated to obtain two patches of the same size with the neighborhood of the pixel, the $M\,(C)$ and $S\,(C)$ can be calculated as:

$$M\,(C) = Mean\left(I\,(P_1), I\,(P_2), \ldots, I\left(P_{Num(C)}\right)\right) \quad (1)$$

$$S(C) = Std\left(I\,(P_1), I\,(P_2), \ldots, I\left(P_{Num(C)}\right)\right) \quad (2)$$

where *Mean* means the mean value of the matrices and *Std* means the standard deviation of the matrices. Finally,

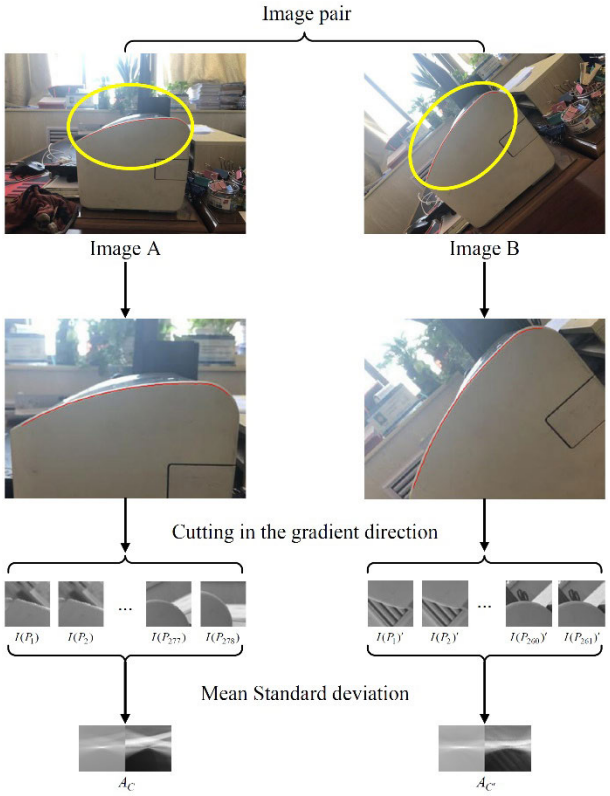

FIGURE 3. The process of curve patches construction.

the mean matrix and standard deviation matrix are concentrated to obtain the curve patch $A(C)$ to represent the curve C uniquely:

$$A(C) = [M(C), S(C)] \tag{3}$$

Fig. 3 describes the process of curve patches construction, which takes a matching curve as an example. As can be seen that for two matched curves C and C' with the length of 278 and 261 pixels respectively, the matrix A_C and $A_{C'}$ of a fixed size can represent the curve uniquely.

In this way, a large-scale curve feature dataset with the patch size of 64 × 128 pixels is constructed, it has 214,296 curve patches labeled with matching information. The number of different changes is shown in Fig. 6, each category has over 30,000 curve patches, of which the training set and the testing set are about 25,000 and 5,000 respectively.

B. NETWORK ARCHITECTURE

The basic architecture of our network-a, shown in Fig. 4, is adopted from L2-Net(DSM) [17], which is built by a seven-layer full convolution structure. Compared with L2-Net(DSM), our network has two more convolutional layers, which are the fourth and the seventh convolutional layers. The fourth convolutional layer contains 32 kernels with size 3 × 3 and the seventh convolutional layer contains 64 kernels with size 3 × 3. Dilated convolution can expand the receptive field without pooling the loss of information and make each convolution output contain a larger range of information [20], we use it in the fourth and seventh layers to learn more features. Batch normalization and ReLU are performed after each convolutional layer except the last layer. There are no pooling layers, and dropout regularization is used before the last layer. Padding with zeros is applied to all convolutional layers except the final one, the size of the convolutional kernel is 3 except for the last layer. Each curve patch with the size of 128 × 64 pixels in our curve feature dataset is divided into two curve patches with the size of 64 × 64 pixels, as the input of the network. The output of the network is L2 normalized to produce a 128-D descriptor with unit length.

Besides, we studied the improved L2-Net(DSM) in network-b, shown in Fig. 5, which produces a 256-D descriptor with unit length. Compared with network-a, the number of convolutional kernels in each layer is doubled. In the first to third convolutional layers, the number of convolutional kernels is changed from 32 to 64. In the fourth to sixth convolutional layers, the number of convolutional kernels is changed from 64 to 128. In the seventh to eighth convolutional layers, the number of convolutional kernels is changed from 128 to 256. The other network parameters are the same as the network-a shown in Fig. 4.

C. LOSS FUNCTION

The Dynamic Soft Margin (DSM) loss function [17] is used in this paper to get the real-valued curve feature descriptor. The ''harder'' triplets in a mini-batch are more useful for training, so it's necessary to measure how hard a triplet is compared with other triplets in the same mini-batch, by computing its signed distance to the decision boundary ($d_{\text{pos}} - d_{\text{neg}}$) and the distribution of these distances. Given a mini-batch of size N, the Probability Distribution Function (PDF) of signed distances is discretized into a histogram, and the $d_{\text{pos}} - d_{\text{neg}}$ for each triplet is computed to make the aggregated histogram more accurate, then the $d_{\text{pos}} - d_{\text{neg}}$ is linearly allocated into two neighboring bins in the histogram, and the Cumulative Distribution Function (CDF) is obtained by integrating the histogram. The loss is defined as:

$$L = \frac{1}{N}\sum_i w_i \cdot \left(d^i_{pos} - d^i_{neg}\right) \tag{4}$$

The w_i for each triplet is weighted by the corresponding value from the CDF.

$$w_i = CDF\left(d^i_{pos} - d^i_{neg}\right) \tag{5}$$

IV. EXPERIMENTS

To evaluate the performance of the proposed D-MSCD, we use the evaluation metrics, FPR95 (false positive rate (FPR) at true positive rate (TPR) equal to 95%) and mAP (mean Average Precision) [25], for reference. Specifically, in the experiments for parameters selection, the FPR95 is computed when TPR = 0.95 according to the following equation:

$$FPR = \frac{FP}{FP + TN} \tag{6}$$

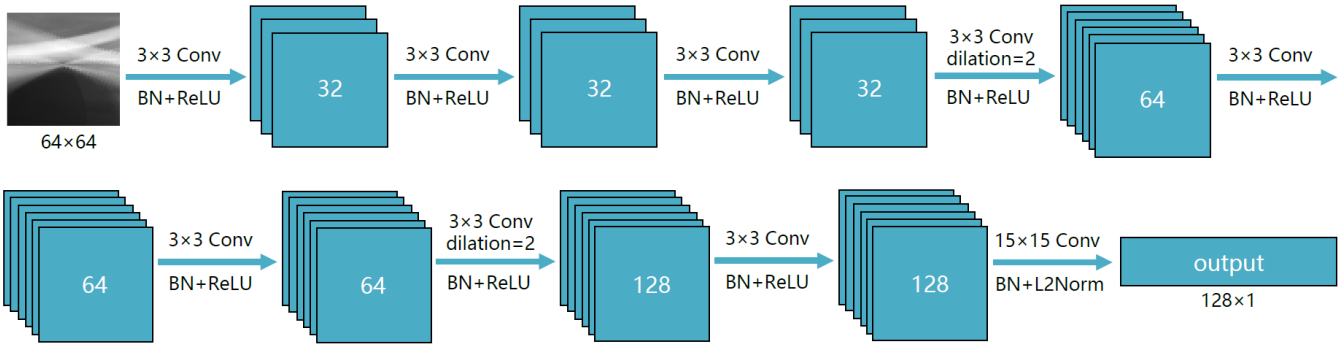

FIGURE 4. Network-a architecture.

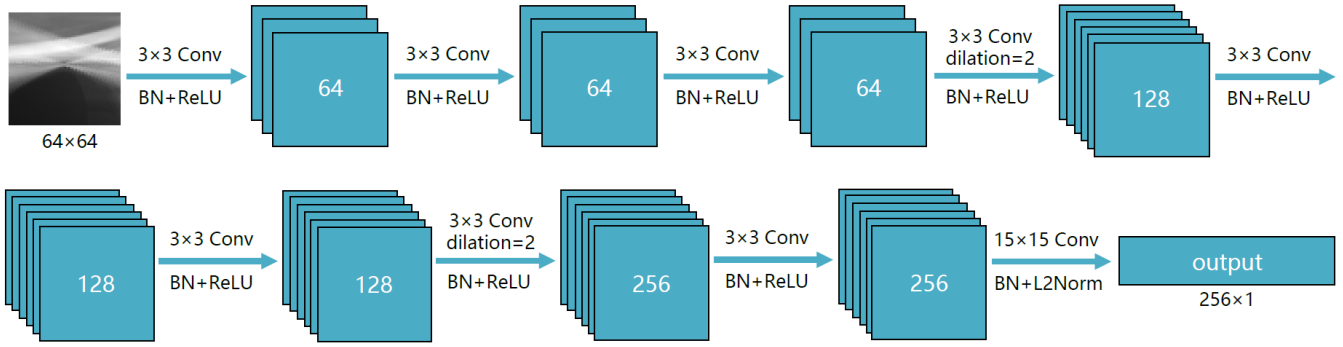

FIGURE 5. Network-b architecture.

where TN denotes true negative rate.

For image matching, the mAP is adopted as the performance indicator. The Average Precision (AP) score of the positive matching category on an image pair is firstly computed as:

$$AP = \frac{\sum_{j=1}^{n} Precision_j}{n} \tag{7}$$

where n represents the total number of retrieved positively matched line pairs and $Precision$ denotes the ratio of the number of the retrieved positive matched line pairs to the total number of retrieved line pairs. Then, the mAP can be calculated as:

$$mAP = \frac{\sum_{i=1}^{m} AP_i}{m} \tag{8}$$

where m is the total number of image pairs in the test set.

As the descriptor is learned from the mean and standard deviation curve patches by deep learning, we name the descriptor proposed in this paper as D-MSCD, and name the descriptor learned from Network-a as D-MSCD-a and the descriptor learned from Network-b as D-MSCD-b in the following experiments.

In the following **Hyper-Parameters** section and **Curve Matching** section, each curve patch in the training set with the size of 128 × 64 pixels is divided into a mean patch and a standard patch with the size of 64 × 64 pixels, then the mean descriptor and the standard descriptor are obtained by network training, finally, they are combined to generate the D-MSCD. The D-MSCD-a is 256-D and D-MSCD-b is 512-D respectively in the two sections.

A. HYPER-PARAMETERS SELECTION

The basic architecture of our modified network and the loss function are based on L2-Net(DSM), so we choose the same hyper-parameters as L2-Net(DSM). The Stochastic Gradient Descent (SGD) is used with momentum and weight decay equal to 0.9 and 0.0001, respectively, to optimize the network. Weights are initialized to orthogonally with gain equal to 0.6, biases set to 0.01, the learning rate is linearly decayed from 0.1 to 0 and the dilated rate is 2. Training is done with PyTorch library. Two TITAN RTX GPUs are employed to run the experiments.

Besides, we studied the influence of the batch size on network performance. We reported the results for batch sizes 64, 128, 256, 512, 1024. We trained the model on Network-b using the training set of the large-scale curve feature dataset constructed in section III. B2), and tested on the testing set. Fig. 7 shows the average FPR95 value over seven types of changes. The performance improves with increasing the mini-batch size but brings little benefit after 512 batch size, and the performance stabilizes after 14 epochs. To make full use of the GPUs, we set the batch size to 1024 and the training epochs to 20 in the following experiments. Table 1 shows the results of the testing set under different transformations.

TABLE 1. The FPR95 (%) values of the testing set under different transformations.

	Scale	Illumination	Blur	Viewpoint	Rotation	Compression	Noise	Mean
D-MSCD-a	0.0371	0.2097	0.0000	0.2077	0.1183	0.1266	0.0353	0.1049
D-MSCD-b	0.0288	0.1748	0.0000	0.1898	0.0788	0.1071	0.0000	0.0827

TABLE 2. Performance of networks on the Brown dataset. The numbers shown are FPR95 (%), the lower the better. "∗" and "+" denote training with anchor swapping and data augmentation.

Descriptor	Length	Train / Test	Notredame		Liberty		Yosemite		Mean
			Liberty	Yosemite	Notredame	Yosemite	Liberty	Notredame	
TFeat-M*[13]	128		7.39	7.24	3.06	8.06	10.31	3.80	6.64
TL+GOR* [22]	128		4.80	5.15	1.95	5.40	6.45	2.38	4.36
PCW [23]	128		7.44	6.56	3.48	5.02	9.84	3.54	5.98
L2-Net+ [14]	128		2.36	1.71	0.72	2.57	4.70	1.29	2.23
CS-L2-Net+ [14]	256		1.71	1.30	0.56	2.07	3.87	1.09	1.76
HardNet+ [15]	128		1.49	1.84	0.53	1.96	2.51	0.78	1.51
DOAP+ [24]	128		1.54	1.21	0.43	2.00	2.62	0.87	1.45
L2-Net(DSM)+ [16]	128		1.21	1.29	0.39	1.51	2.01	0.68	1.18
SOSNet+ [18]	128		1.08	0.95	0.35	1.03	2.12	0.67	1.03
Ours-a+	128		0.88	1.12	0.29	1.37	1.86	0.58	0.89
Ours-b+	256		0.69	0.86	0.20	1.38	1.45	0.53	0.85

FIGURE 6. The number of curve image patches under different transformations.

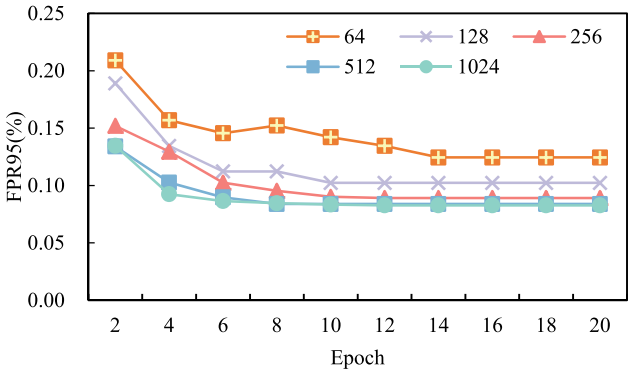

FIGURE 7. Influence of the batch size on network performance.

B. NETWORK PERFORMANCE

To verify the performance of our modified network, we conducted tests on the Brown dataset [26]. The Brown dataset consists of three subsets: *Liberty*, *Notredame*, and *Yosemite* with about 400k normalized 64×64 patches in each, and the dataset assigns each patch with its 3D point ID to identify the matching image patches. Each 3D point ID is associated with a list of patches that are assumed to be matching. Key points were detected by DoG detector and verified by 3D model. Data augmentation is achieved by random flipping and rotating the patch by 90, 180, or 270 degrees. The patch pair classification benchmark measures the ability of a descriptor to discriminate positive patch pairs from negative ones in the Brown dataset. We adopt the commonly used false positive rate at 95% true positive recall (FPR95) to evaluate how well the descriptor classifies the patch pairs. We train one model using each subset and test on the other two subsets, for example, we train one model using *Notredame* subset and test on *Liberty* and *Yosemite* subsets. The results are shown in Table 2. Our descriptors show the best performance compared to other descriptors under the same configuration, the mean FPR95 value is reduced by 17.48% and 27.94% compared to SOSNet and L2-Net(DSM) respectively. This can prove the superiority of our modified network.

C. CURVE MATCHING

To further evaluate the performance of the D-MSCD, we compared the matching performance with the traditional handcrafted descriptors IOCD, IOMSD, and GOCD on the Oxford dataset [25] and the Paper dataset (A dataset of image pairs used in papers including IOCD, IOMSD, and GOCD shown in Fig. 8, we named it the Paper dataset for convenience) by using different descriptors to match the curves in the reference image R and the target image T. First, we obtain the curve patches $M_R = \{M(C_i, i = 1, \ldots, N_1)\}$ in reference image R and $M_T = \left\{M\left(C'_j, j = 1, \ldots, N_2\right)\right\}$

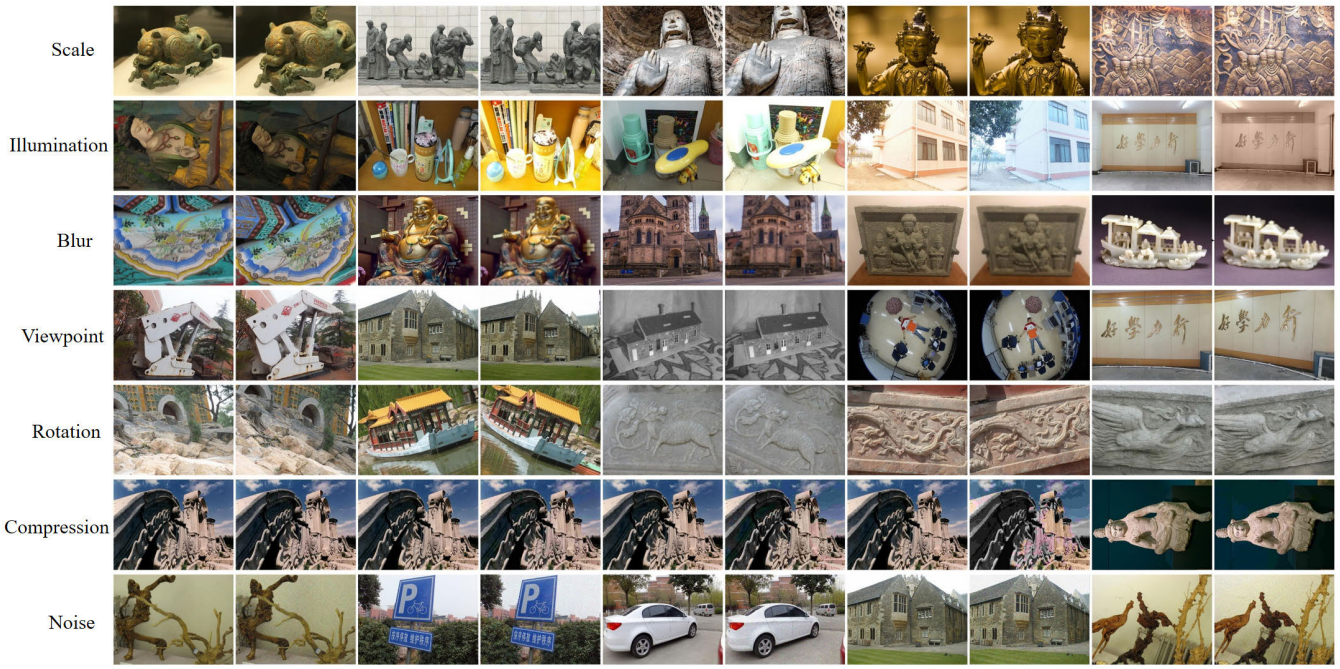

FIGURE 8. Matching test image pairs in the Paper dataset.

in target image T, where N_1 is the number of the detected curves in R and N_2 is that in T. The curve patches were obtained by using the same method as stated in section III. A2). Next, we train M_R and M_T in the network respectively and output the corresponding description matrices of $N_1 \times 256$ and $N_2 \times 256$ or $N_1 \times 512$ and $N_2 \times 512$. Then, the nearest neighbor to the next nearest neighbor distance ratio (NNDR) matching criterion is used to obtain the final matching results of the two images, and the threshold is 0.8. To obtain the true matching results, all the correctly matched curves contained in the image pairs in the two datasets are manually labeled. The large-scale curve feature dataset we constructed is used for training in the following experiment. The results reveal that the descriptors achieve state-of-the-art performance.

1) CURVE MATCHING ON OXFORD DATASET

The Oxford dataset is a standard benchmark library used to evaluate the performance of image feature algorithms. We evaluate our descriptors on five image sequences, namely, Boat (Rotation), Leuven (Illumination), Bikes (Blur), Graf (Viewpoint), and UBC (Compression). In each image sequence, there are six images sorted in an order of increasing degree of distortions with respect to the first image, so each image sequence constitutes five pairs of images. The mAP is used to measure the matching performance of the descriptors.

Fig. 9 (a) shows the matching performance of the proposed D-MSCD-a and D-MSCD-b with IOCD, IOMSD, and GOCD on the Oxford dataset. It can be easily observed that both the proposed D-MSCD-a and D-MSCD-b achieve the best performance on each image sequence compared with the traditional handcrafted descriptors, D-MSCD-b has a little advantage over D-MSCD-a, and the average matching performance is improved by 13.09%, 34.48%, and 31.32% compared with the IOCD, IOMSD, and GOCD respectively. The performance of our descriptors is greatly improved especially under the image transformation of blur change and viewpoint change. The performance improvement proves that the proposed D-MSCD is superior to handcrafted descriptors.

In addition to the matching accuracy, the number of correctly matched curves is also an important factor to measure the performance of the descriptor. Table 3 shows the total number of correctly matched curves obtained by different descriptors on the Oxford dataset. It can be seen that with the same number of detected curves, the D-MSCD has an obvious advantage in the total number of correctly matched curves.

2) CURVE MATCHING ON PAPER DATASET

To further verify the matching performance of our descriptor, we tested different descriptors on the Paper dataset (shown in Fig. 8) with the same method above. The dataset includes seven image sequences, namely, Scale change, Illumination change, Blur change, Viewpoint change, Rotation change, JPEG compression change, and Noise change. In each image sequence, there are five pairs of different images. Fig. 9 (b) shows the matching performance of the D-MSCD-a and D-MSCD-b with IOCD, IOMSD, and GOCD on the Paper dataset, as can be seen that the proposed D-MSCD shows the best performance on each image sequence. What's more, MSCD-b has a little advantage over MSCD-a, and the average matching performance is improved by 5.14%, 12.23%, and 16.94% compared with the IOCD, IOMSD, and GOCD respectively. It also has a great improvement under the image

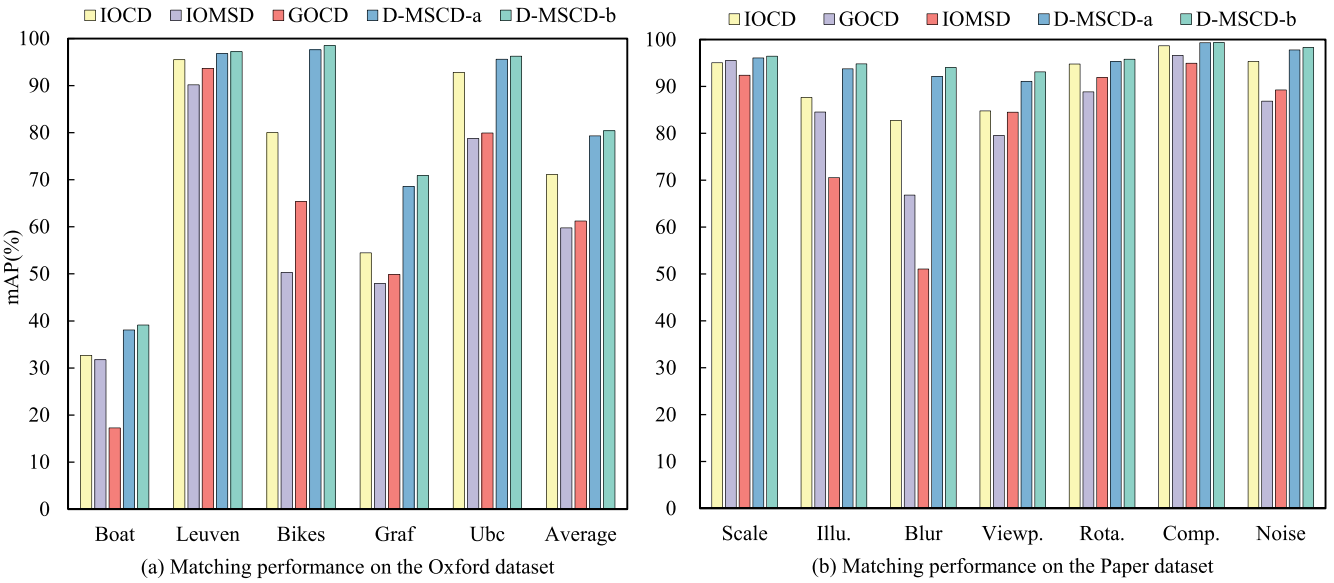

FIGURE 9. Matching performance on the Oxford dataset and the Paper dataset.

TABLE 3. Total correct number of matches for different descriptors on the Oxford dataset.

Descriptor	IOCD	IOMSD	GOCD	D-MSCD-a	D-MSCD-b	Total
Boat	294	292	184	349	353	540
Leuven	595	521	561	611	613	628
Bikes	778	410	572	935	940	998
Graf	405	389	348	466	469	976
Ubc	661	566	557	730	732	751
All	2733	2178	2222	3091	3107	3893

TABLE 4. Total correct number of matches for different descriptors on the Paper dataset.

Descriptor	IOCD	IOMSD	GOCD	D-MSCD-a	D-MSCD-b	Total
Scale	586	521	585	602	604	630
Illumination	232	199	253	300	308	329
Blur	435	201	277	505	509	537
Viewpoint	273	243	248	313	315	377
Rotation	615	573	532	617	618	687
Compression	691	637	639	713	715	735
Noise	400	338	316	413	416	435
All	3242	2712	2850	3452	3485	3730

transformation of blur change and viewpoint change, which is similar to the performance on the Oxford dataset.

Table 4 shows the total number of correctly matched curves for different descriptors on the Paper dataset with the same number of curves detected. It can be seen that the proposed D-MSCD in this paper shows the highest performance on each image sequence compared to the handcrafted descriptor IOCD, IOMSD, and GOCD.

V. CONCLUSION

Inspired by the great progress achieved by the description of feature points in deep learning, we convert the curve feature description problem into the mean value and standard deviation problem of point features, and then propose the curve feature description method based on deep learning. Specifically, we constructed a large-scale curve image dataset labeled with matching information and improved the L2-Net(DSM), the descriptor is obtained by training the network in the self-build dataset. Experimental results show that the obtained descriptor of D-MSCD is superior to the traditional handcrafted descriptors under different image transformations, which demonstrate the great potential of deep learning in curve feature description.

In the future, we will study the effects of different network architectures and loss functions on the learning of curve feature descriptors.

REFERENCES

[1] F. Radenović, G. Tolias, and O. Chum, "CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, 2016, pp. 3–20.

[2] W. Weihong and T. Jiaoyang, "Research on license plate recognition algorithms based on deep learning in complex environment," *IEEE Access*, vol. 8, pp. 91661–91675, May 2020, doi: 10.1109/ACCESS.2020.2994287.

[3] H. Liu, X. Tang, and S. Shen, "Depth-map completion for large indoor scene reconstruction," *Pattern Recognit.*, vol. 99, Mar. 2020, Art. no. 107112, doi: 10.1016/j.patcog.2019.107112.

[4] J. Zhang, L.-R. Gong, K. Yu, X. Qi, Z. Wen, Q. Hua, and S. H. Myint, "3D reconstruction for super-resolution CT images in the Internet of health things using deep learning," *IEEE Access*, vol. 8, pp. 121513–121525, Jul. 2020, doi: 10.1109/ACCESS.2020.3007024.

[5] H. Cui, S. Shen, W. Gao, H. Liu, and Z. Wang, "Efficient and robust large-scale structure-from-motion via track selection and camera prioritization," *ISPRS J. Photogramm. Remote Sens.*, vol. 156, pp. 202–214, Oct. 2019.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[7] Z. Wang, F. Wu, and Z. Hu, "MSLD: A robust descriptor for line matching," *Pattern Recognit.*, vol. 42, no. 5, pp. 941–953, May 2009.

[8] H. Liu, S. Zhi, and Z. Wang, "IOCD: Intensity order curve descriptor," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 27, no. 7, Nov. 2013, Art. no. 1355011.

[9] Z. Wang, S. Zhi, and H. Liu, "Intensity order based mean-standard deviation descriptor," *Pattern Recognit. Artif. Intell.*, vol. 26, no. 4, pp. 409–416, Apr. 2013.

[10] H. Liu, L. Chen, Z. Wang, and Z. Huo, "GOCD: Gradient order curve descriptor," *IEICE Trans. Inf. Syst.*, vol. 100, no. 12, pp. 2973–2983, 2017.

[11] C. Leng, H. Zhang, B. Li, G. Cai, Z. Pei, and L. He, "Local feature descriptor for image matching: A survey," *IEEE Access*, vol. 7, pp. 6424–6434, Dec. 2019, doi: 10.1109/ACCESS.2018.2888856.

[12] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "MatchNet: Unifying feature and metric learning for patch-based matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 3279–3286.

[13] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks," in *Proc. Brit. Mach. Vis. Conf.*, York, U.K., 2016, pp. 1–11.

[14] Y. Tian, B. Fan, and F. Wu, "L2-net: Deep learning of discriminative patch descriptor in Euclidean space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6128–6136, doi: 10.1109/CVPR.2017.649.

[15] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, 2017, pp. 4829–4840.

[16] B. Fan, H. Liu, H. Zeng, J. Zhang, X. Liu, and J. Han, "Deep unsupervised binary descriptor learning through locality consistency and self distinctiveness," *IEEE Trans. Multimedia*, early access, Aug. 17, 2020, doi: 10.1109/TMM.2020.3016122.

[17] L. Zhang and S. Rusinkiewicz, "Learning local descriptors with a CDF-based dynamic soft margin," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 2969–2978, doi: 10.1109/ICCV.2019.00306.

[18] Y. Tian, X. Yu, B. Fan, F. Wu, H. Heijnen, and V. Balntas, "SOSNet: Second order similarity regularization for local descriptor learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 11008–11017, doi: 10.1109/CVPR.2019.01127.

[19] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[20] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Juan, PR, USA, 2016, pp. 1–13.

[21] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 118–126, doi: 10.1109/ICCV.2015.22.

[22] X. Zhang, F. X. Yu, S. Kumar, and S.-F. Chang, "Learning spread-out local feature descriptors," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 4605–4613, doi: 10.1109/ICCV.2017.492.

[23] A. Mukundan, G. Tolias, and O. Chum, "Multiple-kernel local-patch descriptor," in *Proc. Brit. Mach. Vis. Conf.*, London, U.K., 2017, pp. 1–11.

[24] K. He, Y. Lu, and S. Sclaroff, "Local descriptors optimized for average precision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 596–605, doi: 10.1109/CVPR.2018.00069.

[25] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005, doi: 10.1109/TPAMI.2005.188.

[26] S. A. J. Winder and M. Brown, "Learning local image descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, USA, Jun. 2007, pp. 1–8, doi: 10.1109/CVPR.2007.382971.

ZHANQIANG HUO received the B.Sc. degree in mathematics and applied mathematics from the Hebei Normal University of Science and Technology, China, in 2003, the M.Sc. degree in computer software and theory, in 2006, and the Ph.D. degree in circuit and system from Yanshan University, China, in 2009. He is currently an Associate Professor with the College of Computer Science and Technology, Henan Polytechnic University, China. His research interests include computer vision and machine learning.

GUANGXING DU received the B.S. degree in software engineering from Henan Polytechnic University, Jiaozuo, China, in 2018, where he is currently pursuing the M.S. degree in software engineering with the School of Computer Science and Technology. His research interests include deep learning and image processing.

FEN LUO received the bachelor's degree in computer application from the Jiaozuo Institute of Technology, China, in 2002, and the master's degree in resource development and planning from the China University of Mining and Technology-Beijing, in 2010. She is currently an Internal Associate Professor with the College of Computer Science and Technology, Henan Polytechnic University, China. Her research interests include image processing and virtual reality technology.

YINGXU QIAO received the B.Sc. degree in computer science and technology and the M.Sc. degree in controlling theory and engineering from Henan Polytechnic University, China, in 2003 and 2007, respectively, where he is currently pursuing the Ph.D. degree in mining information engineering. His research interests include computer vision and machine learning.

JUNWEI LUO received the Ph.D. degree in computer science from Central South University, Changsha, China. He is currently an Associate Professor with Henan Polytechnic University, Jiaozuo, China. His current research interests include machine learning, bioinformatics, and data mining.