Multi-focus Image Fusion with PCA Filters of PCANet

Xu Song and Xiao-Jun Wu(✉)

Jiangsu Provincial Engineering Laborator of Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi 214122, China
xiaojun_wu_jnu@163.com

Abstract. As is well known to all, the training of deep learning model is time consuming and complex. Therefore, in this paper, a very simple deep learning model called PCANet is used to extract image features from multi-focus images. First, we train the two-stage PCANet using ImageNet to get PCA filters which will be used to extract image features. Using the feature maps of the first stage of PCANet, we generate activity level maps of source images by using nuclear norm. Then, the decision map is obtained through a series of post-processing operations on the activity level maps. Finally, the fused image is achieved by utilizing a weighted fusion rule. The experimental results demonstrate that the proposed method can achieve state-of-the-art fusion performance in terms of both objective assessment and visual quality.

Keywords: Multi-focus image fusion · PCA filters · Nuclear norm

1 Introduction

Image fusion is an information fusion of images. It combines different images obtained by different sensors for the same target or scene, or different images obtained with the same sensor in different imaging modes or at different imaging times. The multi-focus image fusion is a branch of image fusion. The fused image can reflect the information of multiple original images to achieve a comprehensive description of the target and the scene, making it more suitable for visual perception or computer processing. Multi-focus image fusion has become a representative topic since many algorithms have been developed in many fields, such as remote sensing applications, medical imaging applications and surveillance applications [14]. Conventionally, the multi-focus image fusion algorithms can be divided into transform domain algorithms and spatial domain algorithms [15]. Since there are many new algorithms that have been proposed recently, we would like to divide the existing fusion algorithms into three categories: multi-scale transform methods, sparse representation (SR) and low-rank representation based fusion methods, and deep learning based fusion methods.

The multi-scale transform (MST) methods are the most commonly used methods, such as discrete wavelet transform (DWT) [9], contourlet transform

F. Schwenker and S. Scherer (Eds.): MPRSS 2018, LNAI 11377, pp. 1–17, 2019.
https://doi.org/10.1007/978-3-030-20984-1_1

(CT) [25], shift-invariant shearlet transform [24] and curvelet transform (CVT) [5] etc. The basic idea is to perform image transformation on the source images to get the coefficient representation. Then fuse the coefficients according to a certain fusion rule to obtain fused coefficients, and finally obtain the fused image through inverse transformation. All these methods share a "decomposition-fusion-reconstruction" framework. These methods are good representation of their structural information, but can only extract limited direction information and cannot accurately extract the complete contours [26].

In recent years, methods based on sparse representation and low rank representation also have significant performance in image fusion. Yin et al. [27] proposed a novel multi-focus image fusion approach. The key point of this approach is that a maximum weighted multi-norm fusion rule is used to reconstruct fused image from sparse coefficients and the joint dictionary. And the method based on saliency detection in sparse domain [16] also has a remarkable result. Yang el al. [26] combined robust sparse representation with adaptive PCNN is also an effective method. Liu et al. [20] combined multi-scale transform with sparse representation for image fusion which overcomes the inherent defects of both the MST- and SR-based fusion methods. Besides the above methods, Li et al. [10] proposed a novel multi-focus image fusion method based on dictionary learning and low-rank representation which gets a better performance in both global and local structure. Li et al. also achieved significant results from the perspective of noisy image fusion using the low-rank representation [12].

With the development of deep learning, deep features are used as saliency features to fuse images. Liu et al. [19] suggested a convolutional sparse representation (CSR)-based image fusion. The CSR model was introduced by Zeiler et al. [28] in their deconvolutional networks for feature learning. Thus, although CSR is different from deep learning methods, the features extracted by CSR are still deep features. Liu et al. [18] also applied CNN model to image fusion, which can be used to generate the activity level measurement and fusion rule. Li et al. [13] proposed an effective image fusion method using the fixed VGG-19 [23] to generate a single image which contains all the features from infrared and visible images. But we all know that the training of deep model is very time consuming and complicated. And the requirements for hardware conditions are very high.

In this paper, we propose a novel and effective multifocus fusion method based on PCA filters of PCANet [4] which is a very simple deep learning model. The main contribution of this paper is using PCANet to extract image features and using nuclear norm to construct an effective feature space for image fusion. In particular, the training time of PCANet is shorter than that of other CNN-based network, and the extracted features can play the same role. The experimental results demonstrate that the proposed method can obtain state-of-the-art fusion performance in terms of both objective assessment and visual quality.

The structure of the rest paper is organized as follows. In Sect. 2, we give a brief introduction to related work. In Sect. 3, the proposed multi-focus image fusion method is presented in detail. Section 4 presents the experimental results. Finally, Sect. 5 concludes the paper and puts forward the future work.

2 Related Work

In [4], PCANet is a very simple deep learning network which contains three parts: cascaded principal component analysis (two-stage), binary hashing and block-wise histograms (output layer). In this architecture, PCA is employed to learn multistage filter banks. Therefore, this network can be designed and learned extremely easy and efficient. In this paper, we just use the PCA filters to extract image features, binary hashing and block-wise histograms are not used. In two-stage of PCANet [4], the number of filters in the first stage L1 is set as 8, the number of filters in the second stage L2 is the same. Therefore, we can get 8 image features in the first stage and 64 image features in the second stage. Considering the time efficiency and the non redundancy of the data, we just utilize image features of the first stage to fuse images. Therefore, the explanation of the first stage of PCANet is introduced as follows.

For each input image of size of $m \times n$, we take $k_1 \times k_2$ patches, and combine these patches (overlapping) together; i.e., $x_{i,1}, x_{i,2}, \cdots, x_{i,mn} \in R^{k_1 k_2}$, where $x_{i,j}$ denotes the $j-th$ vectorized patch in image I_i. Subtracting patch mean from each patch, we obtain $X_i = [\bar{x}_{i,1}, \bar{x}_{i,2}, \cdots, \bar{x}_{i,mn}]$, where $\bar{x}_{i,j}$ is a mean-centralized patch. Taking the same action for all input images $\{I_i\}_{i=1}^{N}$ (N is the number of the input images) and putting the results together, we get

$$X = [\bar{X}_1, \bar{X}_2, \cdots, \bar{X}_N] \in R^{K_1 K_2 \times N_{mn}} \tag{1}$$

Assuming that the number of filters in layer i is L_i, therefore, L_1, is the number of filters in layer 1. PCA minimizes the reconstruction error, i.e.,

$$\min_{V \in R^{K_1 K_2 \times L_1}} ||X - VV^T X||_F^2, s.t. V^T V = I_{L_1} \tag{2}$$

where I_{L_1} is identity matrix of size $L_1 \times L_1$ and V is a matrix composed of eigenvectors corresponding to the first L_1 eigenvalues of X. Therefore, PCA filters is expressed as

$$W_l^1 = map_{k_1,k_2}(q_l(XX^T)) \in R^{K_1 K_2}, l = 1, 2, \cdots, L_1 \tag{3}$$

where $map_{k_1,k_2}(v)$ is a function that maps v to a matrix $W \in R^{K_1 K_2}$, and $q_l(XX^T)$ is the $l-th$ principal eigenvector of XX^T. The leading principal eigenvectors capture the main variations of all the mean-centralized training patches. Therefore we can use PCA filters to extract image features.

3 The Proposed Fusion Method

3.1 Image Features

Considering the generalization performance of the filters, we train the two-stage PCANet using ImageNet [22] which contains 1000 categories to get PCA filters. We randomly select 5 images for these 1000 categories, therefore, 5000 images

in total are used to train PCA filters and all of them are resized to 256×256 and color images are converted to gray ones. Training PCA filters is implemented in Matlab R2016b on 3.60 GHz Inter(R) Core(TM) CPU with 64 GB RAM. We use the filters of the first stage to extract image features. For example, we use the fixed PCA filters to extract features from the source image, as shown in Fig. 1.

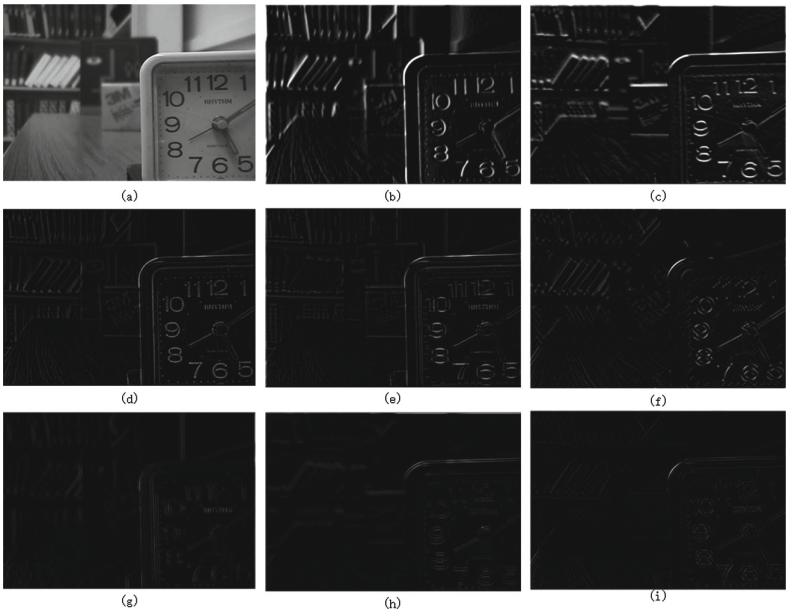

Fig. 1. (a) is an original image and (b)–(i) are image features obtained by the PCA filters of the first stage of the PCANet.

In Fig. 1, (a) is a near focused source image. (b)–(i) are image features extracted by PCA filters, such as texture features, vertical textures, horizontal textures, point textures, diagonal textures, etc. Obviously, the first few images extract features from a global perspective, and the subsequent features become more specific to local significance. Image fusion is the fusion of images with different contents of the same scene into an image. In other words, the saliency features of different images are extracted and combined into an image. In [3], to get better performance, the authors remove the first three principal components in the Eigenface method. Therefore, the feature extraction before fusion can reflect the significance of the source image as much as possible. Because of that, we do an experiment in which there are four cases including all the 8 image features or discarding the first image feature or discarding second one or both of them are abandoned.

Experimental results show that it is better to discard the first two feature maps. Therefore we just utilize the last six feature maps to make the activity level map. This experiment will be introduced in detail in Sect. 4.2.

3.2 Proposed Fusion Method

In this section we will introduce the proposed fusion method in detail. The framework of the proposed fusion method is shown in Fig. 2.

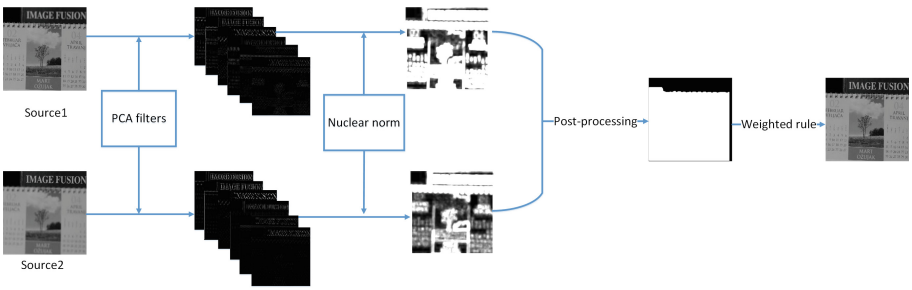

Fig. 2. The framework of the proposed method.

As shown in Fig. 2, the input images are denoted by $Source_1$ and $Source_2$ respectively the feature extraction of the $Source_1$ and $Source_2$ is carried out through the pre-trained PCA filters. We just take the last six image features to calculate the activity level map.

In the [11], authors apply the nuclear norm to the image features. In this paper, we use nuclear norm [17] which is the sum of the singular values of matrix to extract the image features to get the activity level maps. This procedure is shown in Fig. 3.

As shown in Fig. 3, the edge of the feature map is filled with zero, the six feature maps are concatenated as a 6-channel image feature, and the multi-channel image feature is processed by nuclear norm. Taking each pixel as the center, multi-channel block is taken from the same position, and the multi-channel block is transformed into a two-dimensional matrix. The sum of the singular values of the matrix is calculated, and the sum value is used to replace the original pixel point. Therefore, activity level maps are composed of the sum of singular values.

$$M_k(x,y) = ||R(F_k^{1:c}[(x-t):(x+t),(y-t):(y+t)])||_*, \quad c = 6, k \in \{1,2\}, t = 2 \tag{4}$$

where $R(*)$ is reshape operation, (x, y) is the position of the pixel, c is channel number, k is the number of preregistered source image, $(2t+1) \times (2t+1)$ is the block size, $M_k(*)$ is the activity level map and $F_k^{1:c}(*)$ is c feature maps of the $k-th$ source image.

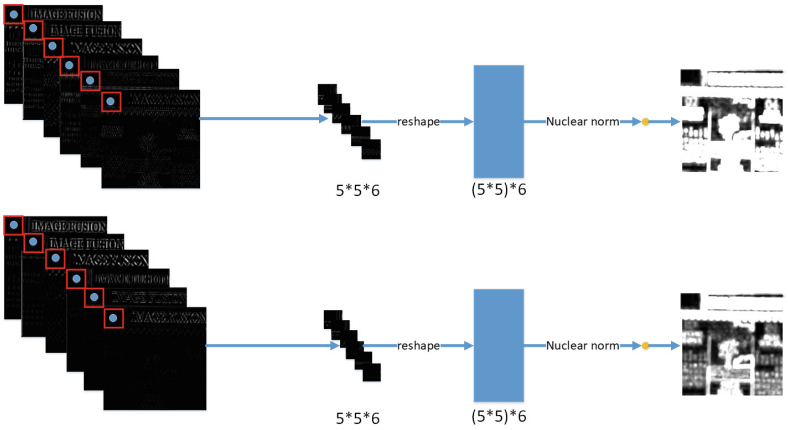

Fig. 3. The procedure of the processing strategy for image features.

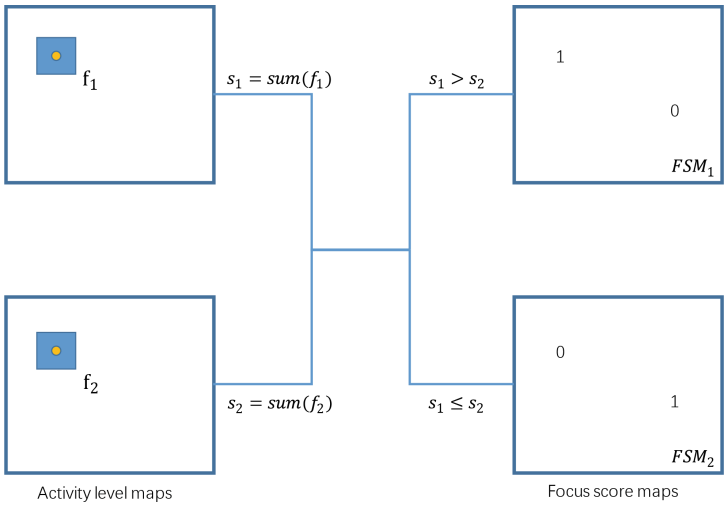

Fig. 4. The construction of focus score maps through the silding window technique.

Next, the activity level map is processed to obtain the decision map. According to the reference [21], we process the activity level maps as shown in Fig. 4.

In Fig. 4, for each corresponding pair of blocks which take each pixel as the center f_1 and f_2, we calculate the sum of all the coefficients in each of them, denoted as s_1 and s_2. If $s_1 > s_2$, the corresponding pixel is set as 1, otherwise, the pixel is set as 0. Finally, we can obtain two complementary focus score maps, denoted as FSM_1 and FSM_2. The steps are shown in Eqs. 5 and 6.

$$FSM_k(x,y) = \begin{cases} 1 & \text{if } s_k > s_n \\ 0 & \text{if } s_k \leq s_n \end{cases}, k \in \{1,2\}, n \in \{1,2\}, k \neq n \tag{5}$$

$$s_i = sum(f_i), i \in \{1, 2\} \tag{6}$$

where $sum(*)$ is the sum of all coefficients in f_i.

As focus score maps usually contain some small holes surrounded by the focused regions, we apply a simple post-processing approach to remove these regions. We apply a small region removal strategy [18] that the area threshold is universally set to $0.1 \times H \times W$, where H and W are the height and width of each source image. Subsequently, we apply morphological closing and opening operation to the focus score maps. Finally, according to [21], we combine the two focus score maps into one decision map, that is, for the pixels where the two focus score maps are complementary, take the value of the first focus score map; otherwise, the value is 0.5. The final decision map denoted as D_{final} is evaluated as shown in formula 7.

$$D_{final} = \begin{cases} FSM_1(x, y) & FSM_1(x, y) \neq FSM_2(x, y) \\ 0.5 & FSM_1(x, y) = FSM_2(x, y) \end{cases} \tag{7}$$

Finally, we obtain the D_{final} shown in Fig. 5 (a).

Fig. 5. (a) is D_{final} and (b) is fused image.

3.3 Fusion Method

With the final decision map D_{final}, the fused image F is calculated by

$$F = D_{final}Source_1 + (1 - D_{final})Source_2. \tag{8}$$

The fused image is shown in Fig. 5(b).
The algorithm is described in Table 1.

Table 1. Algorithm flow chart

(Input): Two registered source images
(Output): The fused image
(1) Training the two stages of PCANet on ImageNet to get PCA filters which are used to extract image features
(2) Applying the nuclear norm to image features, the activity level maps are obtained by Eq. 4
(3) According to Eqs. 5 and 6, the focus score maps are obtained by the activity level maps. Then, applying a simple post-processing approach and Eq. 7 to get the final decision map
(4) Finally, the fused image is obtained by Eq. 8

4 Experiments

In this section, we introduce the source images and experimental environment. There is also a detailed description of Sect. 3.1 and a subjective and objective comparison between the proposed method and the existing methods.

4.1 Experimental Settings

As introduced in Sect. 3.1, our images sets are denoted as SET1 and SET2 coming from two references [29] and [10]. There are 15 pairs of source images in SET1 and 20 pairs in SET2. Part of them are shown in Figs. 6 and 7.

Fig. 6. Four pairs of source images from SET1.

Secondly, we compare the proposed method with several existing fusion methods, including: convolutional sparse representation fusion method (CSR) [19],

Fig. 7. Four pairs of source images from SET2.

multi-focus image fusion with dense SIFT (DSIFT) [21], multi-focus image fusion with a deep convolutional neural network (CNN) [18], infrared and visible image fusion using a deep learning framework (VGG) [13], discrete cosine harmonic wavelet transform fusion method (DCHWT) [7] and cross bilateral filter fusion method (CBF) [8].

In order to evaluate our proposed method and existing methods from an objective perspective, we choose several image quality metrics. These are: Average Gradient (AG), entropy(EN), Mutual Information (MI) [1], FMI_gradient [6] and the sum of the correlations of differences (SCD) [2].

In our experiment, the sliding window size is 5×5 in nuclear norm used for feature processing and the step is one pixel. The sliding window size is 3×3 in the construction of focus score maps and the step is one pixel as well.

The fusion algorithm is implemented in Matlab R2016a on 3.00 GHz Inter(R) Core(TM) CPU with 4 GB RAM.

4.2 Feature Selection Experiment

In this section, we will introduce the experiment which is mentioned in Sect. 3.1. In this two images sets (SET1 and SET2), according to the number of image features, our method is divided into four cases: (1) all features are used, (2) discarding the first image feature, (3) discarding the second image feature and (4) discarding the first two image features. For all the cases, we compare each other and adopt multiple evaluation indexes as reference, and take the average value of experimental results of each image as the final, as shown in Tables 2 and 3.

In Tables 2 and 3, the best results are bloded. It can be seen from the two tables that the distribution of the best results is the same in the four cases of each data set, but the value of the fourth case is better than or equal to the first three cases. Therefore, only the last six image features are selected, and the first two are abandoned.

Table 2. The AG, EN, MI and FMI_gradient average values of the compared methods and the proposed method for SET1.

SET1	(1)	(2)	(3)	(4)
AG	0.0995	0.0995	0.0999	**0.1001**
EN	7.2779	7.2766	7.2782	**7.2787**
MI	14.5558	14.5531	14.5565	**14.5575**
FMI_gradient	0.6634	0.6656	0.6625	**0.6646**

Table 3. The AG, EN, MI and SCD average values of the compared methods and the proposed method for SET2.

SET2	(1)	(2)	(3)	(4)
AG	0.1083	0.1083	**0.1083**	**0.1083**
EN	7.4127	7.4128	**7.4132**	**7.4132**
MI	14.8254	14.8256	**14.8263**	**14.8263**
FMI_gradient	0.4726	0.4727	**0.4728**	**0.4728**

4.3 Image Fusion Results

We use fifteen pairs of source images (SET1) to test comparison methods and the proposed method. The fused results are shown in Fig. 8, we choose one pair of source images as an example. And the values of AG, EN, MI and FMI_gradient for fifteen fused images are shown in Tables 4 and 5.

As shown in Fig. 8, we can see, the proposed method has almost the same fusion performance compared with other classical and novel fusion methods in human visual system. Therefore we mainly discus the fusion performance with quality metrics, as shown in Tables 4 and 5.

In Tables 4 and 5, the best results are bloded, the second-best results are marked in red. We can see, in most cases, the proposed method has good indicators.

We also make the same comparison on SET2 which contains 20 pairs of source images. The fused results are shown in Fig. 9, we choose one pair of source images as an example as well. And the values of AG, EN, MI and SCD for twenty fused images are shown in Tables 6 and 7.

As shown in Fig. 9, we can see, from human visual perspective, there is almost no significant difference in the fusion results between these methods. Therefore we evaluate the fusion results objectively, as shown in Tables 6 and 7.

In Tables 6 and 7, the best results are bloded, the second-best results are marked in red. We can see, in most cases, the proposed method has good indicators as well.

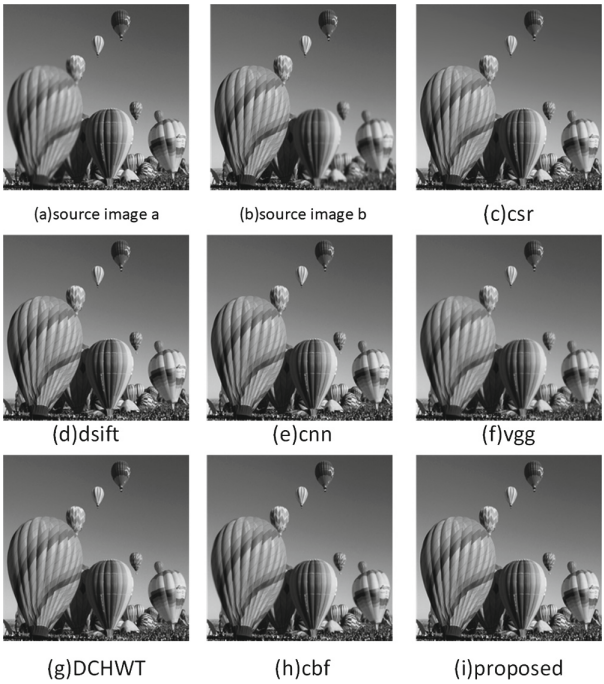

Fig. 8. The examples of fused results. (a) Source image a; (b) Source image b; (c) csr; (d) dsift; (e) cnn; (f) vgg; (g) DCHWT; (h) cbf; (i) The proposed method.

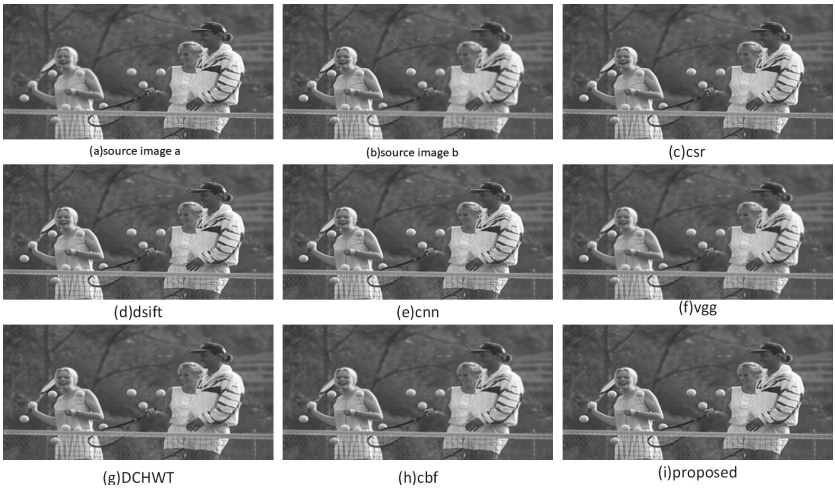

Fig. 9. The examples of fused results. (a) Source image a; (b) Source image b; (c) csr; (d) dsift; (e) cnn; (f) vgg; (g) DCHWT; (h) cbf; (i) The proposed method.

Table 4. The AG, EN, MI and FMI_gradient values of the compared methods and the proposed method for 10 pairs source images from SET1.

SET1		CSR	DSIFT	CNN	VGG	DCHWT	CBF	OURS
image1	AG	0.0957	**0.0985**	**0.0985**	0.0805	0.0934	0.0914	0.0982
	EN	7.4689	**7.4712**	7.4711	7.4447	7.4585	7.4583	7.4709
	MI	14.9377	**14.9424**	14.9422	14.8895	14.9170	14.9167	14.9419
	FMI_gradient	0.4700	**0.8761**	0.8757	0.8430	0.8001	0.8488	0.8738
image2	AG	0.0302	0.0308	0.0308	0.0216	0.0277	0.0275	**0.0309**
	EN	6.4619	6.4814	6.4788	6.4307	6.4734	6.4687	**6.4826**
	MI	12.9238	12.9628	12.9575	12.8615	12.9468	12.9374	**12.9652**
	FMI_gradient	0.2548	0.5554	0.5358	0.3678	0.3464	0.3835	**0.5683**
image3	AG	0.1030	0.1043	0.1059	0.0830	0.0994	0.0957	**0.1063**
	EN	7.2904	7.2950	**7.3017**	7.2823	7.2933	7.2774	7.3012
	MI	14.5808	14.5899	14.6035	14.5647	14.5866	14.5548	**14.6025**
	FMI_gradient	0.5254	0.7039	**0.7068**	0.6359	0.5960	0.6355	0.7065
image4	AG	0.0816	**0.0847**	0.0830	0.0538	0.0804	0.0780	0.0846
	EN	6.6060	6.6587	6.6534	6.5660	**6.6780**	6.6504	6.6571
	MI	13.2120	13.3174	13.3068	13.1321	**13.3560**	13.3009	13.3142
	FMI_gradient	0.4966	**0.6379**	0.6348	0.5315	0.4561	0.5624	0.6350
image5	AG	0.0638	0.0656	0.0644	0.0442	0.0589	0.0614	**0.0674**
	EN	7.3596	7.3394	7.3430	7.3093	**7.4217**	7.3300	7.3324
	MI	14.7192	14.6787	14.6860	14.6185	**14.8434**	14.6600	14.6648
	FMI_gradient	0.4603	0.6355	0.6294	0.5278	0.4359	0.5324	**0.6424**
image6	AG	0.0841	**0.0865**	**0.0865**	0.0616	0.0806	0.0766	**0.0865**
	EN	7.4277	7.4376	7.4329	7.3518	7.3876	7.3755	**7.4398**
	MI	14.8554	14.8753	14.8658	14.7036	14.7751	14.7509	**14.8796**
	FMI_gradient	0.4674	0.6220	0.6130	0.5232	0.4817	0.5214	**0.6288**
image7	AG	0.0863	0.0886	0.0881	0.0584	0.0828	0.0810	**0.0888**
	EN	7.2580	7.2547	7.2560	7.1865	**7.2582**	7.2496	7.2538
	MI	14.5160	14.5095	14.5120	14.3731	**14.5165**	14.4992	14.5077
	FMI_gradient	0.4735	**0.6185**	0.6168	0.5181	0.4543	0.5143	0.6183
image8	AG	0.1001	0.1034	0.1025	0.0823	0.0952	0.0922	**0.1036**
	EN	7.1272	**7.1448**	7.1421	7.1041	7.1199	7.1133	7.1445
	MI	14.2544	**14.2895**	14.2842	14.2082	14.2397	14.2266	14.2891
	FMI_gradient	0.6411	0.7085	0.7120	0.6778	0.6451	0.6702	**0.7123**
image9	AG	0.1066	0.1066	0.1102	0.0860	0.1018	0.0966	**0.1114**
	EN	7.8323	7.8322	7.8376	7.8287	7.8366	7.8351	**7.8402**
	MI	15.6645	15.6643	15.6751	15.6574	15.6733	15.6701	**15.6803**
	FMI_gradient	0.6611	0.6925	0.6955	0.6615	0.6406	0.6622	**0.6941**
image10	AG	0.1107	0.1155	0.1133	0.0718	0.1058	0.1041	**0.1160**
	EN	7.7827	7.7933	7.7870	7.7251	7.7747	7.7747	**7.7946**
	MI	15.5655	15.5866	15.5741	15.4502	15.5494	15.5494	**15.5893**
	FMI_gradient	0.5423	0.5972	0.5965	0.4940	0.4871	0.5248	**0.5983**

Table 5. The AG, EN, MI and FMI_gradient values of the compared methods and the proposed method for another 5 pairs source images from SET1.

SET1		CSR	DSIFT	CNN	VGG	DCHWT	CBF	OURS
image11	AG	0.0728	0.0750	0.0747	0.0511	0.0693	0.0694	**0.0753**
	EN	7.0322	7.0689	7.0222	6.9620	7.0523	**7.0690**	7.0245
	MI	14.0644	14.1377	14.0443	13.9240	14.1047	**14.1379**	14.0491
	FMI_gradient	0.4372	0.5947	**0.6411**	0.5459	0.4384	0.4996	0.6282
image12	AG	0.0876	0.0906	0.0887	0.0561	0.0818	0.0863	**0.0908**
	EN	7.3282	7.3379	7.3295	7.2406	7.3064	7.3195	**7.3404**
	MI	14.6565	14.6758	14.6589	14.4811	14.6127	14.6391	**14.6809**
	FMI_gradient	0.5249	0.6159	0.6122	0.5273	0.4943	0.5375	**0.6165**
image13	AG	0.0967	**0.1001**	0.0997	0.0762	0.0929	0.0954	0.1000
	EN	7.3415	7.3738	7.3740	7.3687	**7.3879**	7.3705	7.3741
	MI	14.6830	14.7475	14.7479	14.7375	**14.7758**	14.7411	14.7483
	FMI_gradient	0.6839	0.8539	**0.8547**	0.8322	0.7497	0.8369	0.8537
image14	AG	0.1780	0.1853	0.1830	0.1000	0.1647	0.1626	**0.1860**
	EN	7.2781	7.3062	7.2926	6.9421	7.1410	7.1515	**7.3102**
	MI	14.5561	14.6124	14.5852	13.8841	14.2819	14.3030	**14.6204**
	FMI_gradient	0.5753	**0.6085**	0.6000	0.4231	0.4283	0.4798	0.5961
image15	AG	0.1514	**0.1557**	0.1540	0.1014	0.1364	0.1405	0.1554
	EN	7.5571	7.4203	7.4427	7.5480	**7.7075**	7.5827	7.4144
	MI	15.1142	14.8407	14.8854	15.0960	**15.4149**	15.1654	14.8289
	FMI_gradient	0.5039	0.5899	0.5847	0.4870	0.4180	0.5044	**0.5972**

Table 6. The AG, EN, MI and SCD values of the compared methods and the proposed method for 10 pairs source images from SET2.

SET2		CSR	DSIFT	CNN	VGG	DCHWT	CBF	OURS
image1	AG	0.0610	0.0620	0.0619	0.0511	0.0581	0.0566	**0.0629**
	EN	6.8647	6.8664	6.8666	6.8466	6.8634	6.8757	**6.8813**
	MI	13.7294	13.7329	13.7333	13.6932	13.7269	13.7515	**13.7625**
	SCD	0.3889	0.3904	0.4100	0.1841	0.3089	0.3125	**0.4547**
image2	AG	0.1027	0.1068	0.1059	0.0766	0.1004	0.0967	**0.1072**
	EN	7.5038	**7.5141**	7.5124	7.4793	7.5054	7.4957	7.5101
	MI	15.0075	**15.0281**	15.0248	14.9586	15.0108	14.9915	15.0202
	SCD	0.4366	**0.4905**	0.4874	0.2553	0.3523	0.3579	0.4678

Table 6. (*continued*)

SET2		CSR	DSIFT	CNN	VGG	DCHWT	CBF	OURS
image3	AG	0.0935	0.0964	0.0956	0.0717	0.0910	0.0842	**0.0968**
	EN	7.5205	**7.5364**	7.5349	7.5020	7.5246	7.5151	7.5353
	MI	15.0410	**15.0727**	15.0698	15.0040	15.0493	15.0301	15.0706
	SCD	0.4293	**0.4772**	0.4724	0.2408	0.3511	0.3021	0.4690
image4	AG	0.1004	0.1047	0.1031	0.0749	0.0984	0.0912	**0.1052**
	EN	7.5335	7.5542	7.5519	7.4801	7.5406	7.5263	**7.5589**
	MI	15.0669	15.1083	15.1037	14.9602	15.0812	15.0527	**15.1178**
	SCD	0.4219	0.4560	0.4506	0.2456	0.3571	0.3238	**0.4549**
image5	AG	0.0901	0.0925	0.0920	0.0744	0.0862	0.0819	**0.0932**
	EN	7.6695	7.6756	7.6746	7.6517	7.6625	7.6558	**7.6786**
	MI	15.3390	15.3512	15.3491	15.3033	15.3251	15.3116	**15.3571**
	SCD	0.4387	0.4799	0.4751	0.2102	0.3346	0.2636	**0.5088**
image6	AG	0.0841	0.0859	0.0841	0.0625	0.0818	0.0746	**0.0866**
	EN	7.0884	7.1263	7.1241	7.0883	**7.1538**	7.1222	7.1340
	MI	14.1768	14.2525	14.2482	14.1766	**14.3077**	14.2444	14.2681
	SCD	0.2906	0.3302	0.3242	0.1949	0.2628	0.2270	**0.3384**
image7	AG	0.0746	0.0764	0.0762	0.0557	0.0729	0.0674	**0.0773**
	EN	7.2883	7.3089	7.3078	7.2659	7.3073	7.3024	**7.3154**
	MI	14.5767	14.6179	14.6156	14.5318	14.6146	14.6047	**14.6309**
	SCD	0.2584	**0.3168**	0.3158	0.1857	0.2257	0.1927	0.3129
image8	AG	0.1696	0.1745	0.1726	0.1140	0.1653	0.1560	**0.1746**
	EN	7.4343	7.4531	7.4498	7.3585	7.4415	7.4157	**7.4554**
	MI	14.8686	14.9063	14.8997	14.7171	14.8830	14.8314	**14.9109**
	SCD	0.6795	**0.6972**	0.6933	0.4368	0.6029	0.6116	**0.6972**
image9	AG	0.1495	0.1543	0.1530	0.1034	0.1459	0.1378	**0.1550**
	EN	7.6738	**7.6883**	7.6870	7.6179	7.6707	7.6535	7.6855
	MI	15.3475	**15.3767**	15.3739	15.2358	15.3415	15.3070	15.3710
	SCD	0.5786	**0.6140**	0.6088	0.3674	0.4962	0.4928	0.5813
image10	AG	0.1150	0.1183	0.1176	0.0875	0.1115	0.1030	**0.1185**
	EN	7.4619	7.4853	7.4830	7.4326	**7.4903**	7.4785	7.4884
	MI	14.9239	14.9705	14.9660	14.8652	**14.9806**	14.9570	14.9767
	SCD	0.4442	**0.4912**	0.4881	0.2396	0.3422	0.2873	0.4888

Table 7. The AG, EN, MI and SCD values of the compared methods and the proposed method for another 10 pairs source images from SET2.

SET2		CSR	DSIFT	CNN	VGG	DCHWT	CBF	OURS
image11	AG	0.0648	0.0659	0.0655	0.0514	0.0633	0.0564	**0.0668**
	EN	7.0320	7.0404	7.0403	7.0350	**7.0517**	7.0474	7.0494
	MI	14.0641	14.0808	14.0805	14.0700	14.1033	14.0949	**14.0988**
	SCD	0.2500	0.2893	0.2872	0.1565	0.2039	0.1383	**0.3036**
image12	AG	0.0974	0.1021	0.1013	0.0704	0.0961	0.0929	**0.1025**
	EN	7.0401	7.0711	7.0645	6.9295	7.0319	7.0393	**7.0748**
	MI	14.0802	14.1423	14.1290	13.8590	14.0637	14.0786	**14.1497**
	SCD	0.5151	0.5549	0.5428	0.3314	0.4740	0.5025	**0.5680**
image13	AG	0.0987	0.1004	0.0998	0.0732	0.0964	0.0877	**0.1014**
	EN	7.3394	7.3571	7.3556	7.2960	7.3530	7.3302	**7.3625**
	MI	14.6787	14.7143	14.7112	14.5920	14.7060	14.6604	**14.7249**
	SCD	0.3363	**0.3954**	0.3924	0.2185	0.2910	0.2449	0.3896
image14	AG	0.0759	0.0785	0.0783	0.0606	0.0750	0.0694	**0.0795**
	EN	7.5955	7.6102	7.6092	7.5695	7.5959	7.5907	**7.6135**
	MI	15.1910	15.2205	15.2185	15.1390	15.1919	15.1814	**15.2271**
	SCD	0.3024	**0.3691**	0.3654	0.1918	0.2647	0.2125	0.3563
image15	AG	0.1042	0.1055	0.1048	0.0809	0.1005	0.0935	**0.1069**
	EN	7.4169	7.4124	7.4136	7.4118	**7.4299**	7.4162	7.4172
	MI	14.8339	14.8247	14.8273	14.8237	**14.8598**	14.8325	14.8345
	SCD	0.5054	0.5398	0.5271	0.2756	0.3836	0.3345	**0.5414**
image16	AG	0.1145	**0.1184**	0.1167	0.0847	0.1119	0.1020	0.1182
	EN	7.1316	7.1596	7.1567	7.0791	7.1452	7.1368	**7.1655**
	MI	14.2631	14.3193	14.3135	14.1582	14.2903	14.2736	**14.3311**
	SCD	0.4453	0.4715	0.4653	0.2518	0.3731	0.3083	**0.4888**
image17	AG	0.1594	0.1647	0.1629	0.1113	0.1561	0.1437	**0.1649**
	EN	7.7790	**7.7817**	7.7819	7.7329	7.7720	7.7647	7.7796
	MI	15.5579	15.5634	15.5638	15.4657	15.5439	15.5293	**15.5593**
	SCD	0.5938	**0.6160**	0.6082	0.3571	0.4972	0.4140	0.5970
image18	AG	0.1432	0.1487	0.1475	0.0994	0.1408	0.1344	**0.1493**
	EN	7.4088	7.4331	7.4294	7.2854	7.3958	7.3940	**7.4385**
	MI	14.8176	14.8662	14.8588	14.5707	14.7915	14.7881	**14.8770**
	SCD	0.6146	**0.6398**	0.6353	0.3933	0.5504	0.5599	0.6375
image19	AG	0.1061	0.1123	0.1111	0.0717	0.1077	0.1030	**0.1125**
	EN	7.4622	7.4888	7.4841	7.3596	7.4674	7.4627	**7.4922**
	MI	14.9245	14.9775	14.9682	14.7193	14.9349	14.9253	**14.9844**
	SCD	0.4075	0.4311	0.4283	0.2667	0.3727	0.3592	**0.4323**
image20	AG	0.0840	0.0854	0.0852	0.0615	0.0810	0.0759	**0.0860**
	EN	7.6188	7.6243	7.6239	7.5925	7.6142	7.6128	**7.6271**
	MI	15.2377	15.2486	15.2478	15.1850	15.2284	15.2255	**15.2542**
	SCD	0.2762	0.3566	0.3338	0.2151	0.2503	0.2719	**0.3670**

5 Conclusion

In this paper, we propose a novel fusion method based on PCANet. First of all, we utilize the PCA filters to extract image features of source images, and then we apply the nuclear norm to process the image features in order to get activity level maps. Through a series of post-processing operations on activity level maps, the decision map is obtained. Finally, the fused image is obtained by utilizing a weighted fusion rule. The experimental results demonstrate that the proposed method can obtain state-of-the-art fusion performance in terms of both objective assessment and visual quality.

References

1. Mutual information. https://ww2.mathworks.cn/matlabcentral/fileexchange/28694-mutual-information
2. Aslantas, V., Bendes, E.: A new image quality metric for image fusion: the sum of the correlations of differences. AEU Int. J. Electron. Commun. **69**(12), 1890–1896 (2015)
3. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: recognition using class specific linear projection. Technical report, Yale University New Haven United States (1997)
4. Chan, T.-H., Jia, K., Gao, S., Jiwen, L., Zeng, Z., Ma, Y.: PCANet: a simple deep learning baseline for image classification? IEEE Trans. Image Process. **24**(12), 5017–5032 (2015)
5. Guo, L., Dai, M., Zhu, M.: Multifocus color image fusion based on quaternion curvelet transform. Opt. Express **20**(17), 18846–18860 (2012)
6. Haghighat, M., Razian,M.A.: Fast-FMI: non-reference image fusion metric. In: 2014 IEEE 8th International Conference on Application of Information and Communication Technologies (AICT), pp. 1–3. IEEE (2014)
7. Shreyamsha Kumar, B.K.: Multifocus and multispectral image fusion based on pixel significance using discrete cosine harmonic wavelet transform. Signal Image Video Process. **7**(6), 1125–1143 (2013)
8. Shreyamsha Kumar, B.K.: Image fusion based on pixel significance using cross bilateral filter. Signal Image Video Process. **9**(5), 1193–1204 (2015)
9. Li, H., Manjunath, B.S., Mitra, S.K.: Multisensor image fusion using the wavelet transform. Graph. Models Image Process. **57**(3), 235–245 (1995)
10. Li, H., Wu, X.-J.: Multi-focus Image fusion using dictionary learning and low-rank representation. In: Zhao, Y., Kong, X., Taubman, D. (eds.) ICIG 2017. LNCS, vol. 10666, pp. 675–686. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71607-7_59
11. Li, H., Wu,X.-J.: Infrared and visible image fusion with ResNet and zero-phase component analysis. arXiv preprint arXiv:1806.07119 (2018)
12. Li,H., Wu, X.-J.: Multi-focus noisy image fusion using low-rank representation. arXiv preprint arXiv:1804.09325 (2018)
13. Li, H., Wu, X.-J., Kittler,J.: Infrared and visible image fusion using a deep learning framework. arXiv preprint arXiv:1804.06992 (2018)
14. Li, S., Kang, X., Fang, L., Jianwen, H., Yin, H.: Pixel-level image fusion: a survey of the state of the art. Inf. Fusion **33**, 100–112 (2017)

15. Li, S., Kang, X., Jianwen, H., Yang, B.: Image matting for fusion of multi-focus images in dynamic scenes. Inf. Fusion **14**(2), 147–162 (2013)
16. Liu, C.H., Qi, Y., Ding, W.R.: Infrared and visible image fusion method based on saliency detection in sparse domain. Infrared Phys. Technol. **83**, 94–102 (2017)
17. Liu, G., Lin, Z., Yu, Y.: Robust subspace segmentation by low-rank representation. In: Proceedings of the 27th International Conference on Machine Learning (ICML-2010), pp. 663–670 (2010)
18. Liu, Y., Chen, X., Peng, H., Wang, Z.: Multi-focus image fusion with a deep convolutional neural network. Inf. Fusion **36**, 191–207 (2017)
19. Liu, Y., Chen, X., Ward, R.K., Wang, Z.J.: Image fusion with convolutional sparse representation. IEEE Signal Process. Lett. **23**(12), 1882–1886 (2016)
20. Liu, Y., Liu, S., Wang, Z.: A general framework for image fusion based on multi-scale transform and sparse representation. Inf. Fusion **24**, 147–164 (2015)
21. Liu, Y., Liu, S., Wang, Z.: Multi-focus image fusion with dense sift. Inf. Fusion **23**, 139–155 (2015)
22. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
24. Wang, L., Li, B., Tian, L.-F.: EGGDD: an explicit dependency model for multi-modal medical image fusion in shift-invariant shearlet transform domain. Inf. Fusion **19**, 29–37 (2014)
25. Yang, S., Wang, M., Jiao, L., Wu, R., Wang, Z.: Image fusion based on a new contourlet packet. Inf. Fusion **11**(2), 78–84 (2010)
26. Yang, Y., Yang, M., Huang, S., Ding, M., Sun, J.: Robust sparse representation combined with adaptive PCNN for multifocus image fusion. IEEE Access **6**, 20138–20151 (2018)
27. Yin, H., Li, Y., Chai, Y., Liu, Z., Zhu, Z.: A novel sparse-representation-based multi-focus image fusion approach. Neurocomputing **216**, 216–229 (2016)
28. Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: Deconvolutional networks (2010)
29. Zhang, Y., Bai, X., Wang, T.: Boundary finding based multi-focus image fusion through multi-scale morphological focus-measure. Inf. fusion **35**, 81–101 (2017)