Multi-focus image fusion with the all convolutional neural network*

DU Chao-ben (杜超本)** **and GAO She-sheng** (高社生)
School of Automation, Northwestern Polytechnical University, Xi'an 710129, China

(Received 10 September 2017; Revised 24 September 2017)

A decision map contains complete and clear information about the image to be fused, which is crucial to various image fusion issues, especially multi-focus image fusion. However, in order to get a satisfactory image fusion effect, getting a decision map is very necessary and usually difficult to finish. In this letter, we address this problem with convolutional neural network (CNN), aiming to get a state-of-the-art decision map. The main idea is that the max-pooling of CNN is replaced by a convolution layer, the residuals are propagated backwards by gradient descent, and the training parameters of the individual layers of the CNN are updated layer by layer. Based on this, we propose a new all CNN (ACNN)-based multi-focus image fusion method in spatial domain. We demonstrate that the decision map obtained from the ACNN is reliable and can lead to high-quality fusion results. Experimental results clearly validate that the proposed algorithm can obtain state-of-the-art fusion performance in terms of both qualitative and quantitative evaluations.

Document code: A **Article ID:** 1673-1905(2018)01-0071-5
DOI https://doi.org/10.1007/s11801-018-7207-x

In recent years, various image fusion algorithms have been proposed. They can be divided into two categories[1]: spatial domain algorithms and transform domain algorithms. The latter ones usually convert the source image to another feature domain, where the source image can be effectively fused. The most popular transform domain fusion algorithms are based on multi-scale transform (MST) methods. Some representative examples include the Laplacian pyramid (LP)[2], the morphological pyramid (MP)[3], the discrete wavelet transform (DWT)[4], the dual-tree complex wavelet transform (DTCWT)[5] and the non-subsampled contourlet transform (NSCT)[6]. These methods share a common three-step framework, namely, decomposition, fusion and reconstruction[7]. Many studies have also been taken in this direction[8,9], where the input image is firstly transformed into a multi-resolution representation by multi-resolution, and then different spectral information is selected and combined to reconstruct the fused images.

A new transform domain fusion approach[10] has become a compelling branch of the field. Unlike the MST-based approach described above, it transforms the image into a single scale feature area through some advanced signal representation theories, such as sparse representation (SR) and independent component analysis (ICA). This approach usually uses the sliding window technique to pursue the approximate translation invariant fusion process. The key problem with it is to explore an effective feature domain to obtain the focus map.

Block-based fusion strategy decomposes the source images into blocks, and each pair of blocks is fused with a designed activity level measurement like sum-modified-Laplacian (SML)[11]. Obviously, the size of the block has a great impact on the quality of the fusion results[12-16]. The spatial domain methods[17,18] are based on image segmentation by sharing the similar idea of block-based methods, but the fusion quality relies heavily on the segmentation accuracy.

In recent years, the multi-focus image fusion algorithm based on spatial domain has been widely concerned. Several state-of-the-art pixel-based image fusion algorithms have been proposed[19-21]. However, these algorithms may lose some of the original image information due to inaccurate fusion decision maps.

In this letter, we address this problem with an all convolutional neural network (ACNN), in which the max-pooling of CNNs is replaced by a convolution layer, aiming to get a direct focus map. Based on this idea, we propose a new ACNN-based multi-focus image fusion method in spatial domain. We demonstrate that the feature map obtained from the ACNN can lead to high-quality fusion results.

CNN is a typical depth learning model that attempts to learn a hierarchical representation of the mechanism of an image at different levels of abstraction[22]. As shown in Fig.1, a typical CNN is mainly composed of input layer, convolution layer, max-pooling (subsampling), fully connected layer and output layer.

The input of the CNN is usually the original image X. In this letter, we use H_i to represent the feature map of the ith layer of CNN (H_0=X). Assuming that H_i is the

* This work has been supported by the National Natural Science Foundation of China (No.61174193).
** E-mail: dcbxjdaxue@163.com

convolution layer, the generation process of H_i can be described as

$$H_i=f(H_{i-1}\otimes W_i+b_i), \quad (1)$$

where W_i is the convolutional kernel, b_i is the bias, the symbol $\otimes$ indicates convolutional operation, and $f(\cdot)$ is non-linear ReLU activation function.

The max-pooling (subsampling) layer usually follows the convolution layer, then max-pooling (subsampling) the feature graph according to a certain max-pooling (subsampling) rule. Through the alternation of multiple convolution and max-pooling (subsampling) layers, the CNN relies on a fully connected network to classify the extracted features, and the probability distribution Y based on the input is obtained.

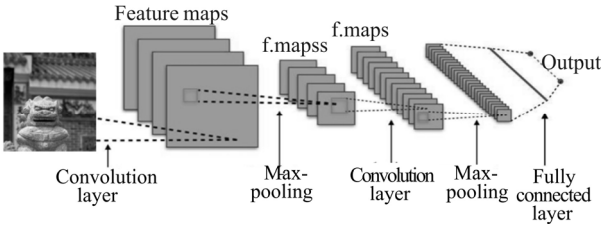


Fig.1 Typical structure of convolution neural network

The vast majority of modern CNN used for object recognition is built using the same principles: They use alternating convolution and max-pooling (subsampling) layers followed by a small number of fully connected layers, as shown in Fig.1. Within each layer, piece-wise-linear activation functions are used. The networks are typically parameterized to be large and regularized during training using dropout. A considerable amount of research has focused on improving the performance of this basic pipeline over the last years. Among these, two major directions can be identified. First, lots of extensions[23-26] were recently proposed to enhance networks which follow this basic scheme. Since all of these extensions and different architectures come with their own parameters and training procedures, which components of CNN are actually necessary for achieving state of the art performance on current object recognition datasets is a question. By studying the most simple architecture, we could conceive that a homogeneous network solely consists of convolutional layers, with occasional dimensionality reduction by using a stride of 2. Springenberg[27] found that this basic architecture reaches state-of-the-art performance with no need of complicated activation functions, any response normalization or max-pooling. Our results also confirm the effectiveness of small convolutional layers as recently proposed by Simonyan & Zisserman[28] and give rise to interesting new questions about the necessity of pooling in CNN.

The models we use in our experiments differ from standard CNNs in a key aspect. We replace the pooling layers, which are present in practically all modern CNNs used for object recognition, with standard convolutional layers with stride of 2.

Let ψ denote a feature map produced by some layer of a CNN. It can be described as a 3-dimensional array with size of $W\times H\times N$, where W and H are the width and height, and N is the number of channels (in case ψ is the output of a convolutional layer, N is the number of filters in this layer). Then p-norm pooling (or subsampling) with pooling size of k (or half-length k=2) and stride of r applied to the feature map ψ is a 3-dimensional array $s(\psi)$ with the following entries:

$$s_{i,j,u}(\psi)=\Big(\sum_{h=-\lfloor k/2\rfloor}^{\lfloor k/2\rfloor}\sum_{w=-\lfloor k/2\rfloor}^{\lfloor k/2\rfloor}\left|\psi_{g(h,w,i,g,u)}\right|^{p}\Big)^{1/p}, \quad (2)$$

where $g(h, w, i, g, u)=(r\cdot i+h, r\cdot j+w, u)$ is the function mapping from positions in s to positions in ψ respecting the stride, and p is the order of the p-norm (for $p\to\infty$, it becomes the commonly used max pooling). If $r>k$, pooling regions do not overlap. However, current CNN architectures typically include overlapping pooling with k=3 and r=2. Let us now compare the pooling operation defined by Eq.(2) with the standard definition of a convolutional layer c applied to feature map ψ given as

$$c_{i,j,o}(\psi)=f\Big(\sum_{h=-\lfloor k/2\rfloor}^{\lfloor k/2\rfloor}\sum_{w=-\lfloor k/2\rfloor}^{\lfloor k/2\rfloor}\sum_{u=1}^{N}\theta_{h,w,u,o}\cdot\psi_{g(h,w,i,g,u)}\Big), \quad (3)$$

where θ is the convolutional weight (or the kernel weight, or filters). $f(\cdot)$ is the activation function, typically a rectified activation ReLU $f(x)=\max(x, 0)$, and $o\in[1, M]$ is the number of output features (or channels) of the convolutional layer. It is clear that both operations depend on the same elements of the previous layer's feature map. The pooling layer can be seen as performing a feature-wise convolution ($\theta_{h,w,u,o}$=1 if $u=o$, and $\theta_{h,w,u,o}$=0 otherwise) in which the activation function is replaced by the p-norm. One main function of the pooling layer is that the spatial dimensionality reduction performed by pooling makes covering larger parts of the input in higher layers possible. Therefore, pooling can be removed from a network without abandoning the spatial dimensionality reduction by two means[27]: We can remove each pooling layer and increase the stride of the convolutional layer before it accordingly; We can replace the pooling layer by a normal convolution with stride larger than 1 (i.e., for a pooling layer with k=3 and r=2, we can replace it with a convolution layer with corresponding stride and kernel size, and the number of output channels is equal to the number of input channels).

The first option is equivalent to a pooling operation in which only the top-left feature response is considered and can result in less accurate recognition. The second option does not suffer from this problem, since all existing convolutional layers stay unchanged, but it results in an increase of overall network parameters. It is worth noting that replacing pooling by convolution adds inter-feature dependencies unless the weight matrix θ is constrained. We emphasize that this replacement can also be seen as learning the pooling operation rather than fixing it, which has been considered using different parameterizations in Refs.[29-31]. In the multi-focus image fusion process in this letter, we adopt the second one. It should be noted that the idea of removing pooling is not entirely

unprecedented. First, the nomenclature in early work on CNNs[29] (referring to pooling layers as subsampling layers already) suggested the usage of different operations for subsampling. Second, although only considering small networks, the experiments on using only convolution layers (with occasional subsampling) in an architecture similar to traditional CNNs already appeared[32].

As described above, the generation of focus map in the image fusion process can be viewed as a feature detection problem or a classification problem[33]. In particular, the fusion rule is viewed as a two-class classification problem. For a pair of image patches $\{p_A, p_B\}$ of the same scene, our goal is to learn an ACNN whose output is a scalar ranging from 0 to 1. Specifically, when p_A is focused while p_B is the defocused region, the output value should be close to 1, and when p_B is focused while p_A is the defocused region, the output value should be close to 0.That is to say, the output represents the focus property of the patch pair. Therefore, the use of the ACNN to fuse the image is feasible in theory. In the ACNN, the input is an image, and the output is a label vector that represents the probability of each category. Between these two ends, the network consists of several convolutional layers and fully-connected layers. The convolution layers are generally considered to be feature extraction parts in the system, and the fully-connected layers present at the output are considered as classification section[34].

The schematic diagram of our algorithm is shown in Fig.2. It can be seen that our method consists of four steps: focus detection, initial segmentation, morphological operation and fusion. In the first step, the two source images are fed to a pre-trained ACNN model to output a feature map, which contains the focus information of source images. Particularly, each coefficient in the feature map indicates the focus property of a pair of corresponding patches from two source images. Then, a focus map with the same size of source images is obtained from the feature map by averaging the overlapping patches. In the second step, the feature map is segmented into a binary map with a threshold of 0.9. In the third step, we refine the binary segmented map with morphological operation, to generate the final decision map. In the last step, the fused image is obtained with the final decision map using the pixel-wise weighted-average strategy.

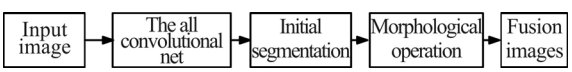


Fig.2 Schematic diagram of the proposed multi-focus image fusion algorithm

Assume that A and B represent two original images to be fused respectively. In this letter, if the image to be fused is a color image, we first transform it into the gray space. Through the method proposed in this letter, we first get the feature map $\boldsymbol{S}$, which ranges from 0 to 1. It can be seen from the feature map of Fig.2 that the focus information is accurately detected. Intuitively, the value of the area with rich details seems to be close to 1 (white) or 0 (black), while the plain area tends to have its own value close to 0.5 (gray).

In order to get a more accurate decision map, the feature map $\boldsymbol{S}$ needs to be further processed. In this algorithm, the most popular maximum strategy is used[35,36]. Correspondingly, a fixed threshold β=0.9 is used to segment $\boldsymbol{S}$ into binary map $\boldsymbol{T}$, which is given as follows

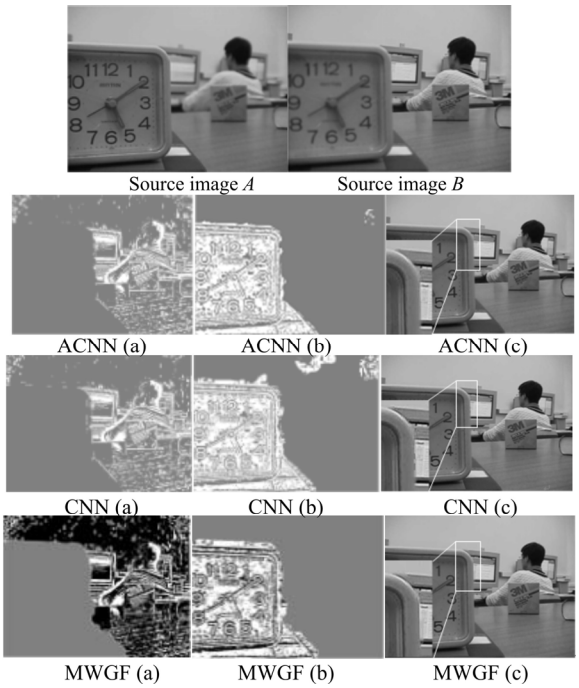
$$T(x,y)=\begin{cases}1, & S(x,y)>0.9\\ 0, & \text{otherwise}\end{cases}. \tag{4}$$

Combined with the final fusion decision map $\boldsymbol{D}$, the fused image F is obtained according to the pixel weighted average rule, as follows

$$F(x,y)=D(x,y)A(x,y)+[1-D(x,y)]B(x,y). \tag{5}$$

In order to verify the effectiveness of the proposed ACNN-based fusion method, two pairs of multi-focus images (including color images and gray-scale images) are used in our experiments. The proposed fusion method is compared with four state-of-the-art multi-focus image fusion methods, which are MWGF[21], SSDI[37], DCNN[35] and DSIFT[20].

We first compare the performance of different fusion methods in terms of subjectivity. For this purpose, we mainly provide the "Lab" source image pair as an example to show the difference between different methods. Fig.3(c) shows the fused images obtained with different fusion methods. In each of the fused images, the area around the boundary between the focused and defocused parts is magnified and displayed in the lower left corner. The fusion results of the CNN, MWGF and DSIFT -based methods contain some undesirable artifacts on the right border of the clock, especially for the CNN-based and MWGF-based methods. The fusion results based on the MWGF and SSDI are blurred in the upper right corner of the clock.



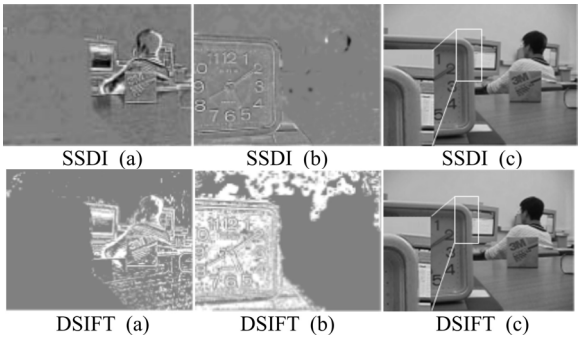


Fig.3 (a) (b) Different images obtained by subtracting source image *A* and source image *B* from each fused image; (c) Fused images

Fig.3(a) and (b) show the images obtained by subtracting source image *A* and source image *B* from each fused image, respectively, and the value of each difference image is normalized to the range from 0 to 1. The differential images CNN (b) and DSIFT (b) clearly show that the CNN and DSIFT-based methods have a partial residual in the upper right corner. The SSID-based approach is not sufficient in the integration of the head of the character. The difference image of SSDI (b) also reveals this point. We can see that the MWDF-based approach performs well in terms of extraction details, except for the border area. In general, the fusion image obtained by our method has the highest visual quality in all of the five methods, which can be further verified by the difference images shown in Fig.3(a) and (b).

Fig.4 shows the fused results of the 'Temple' image set. The results show that each algorithm can achieve the goal of image fusion. However, different algorithms can produce different quality fused images, depending on their different performance. The MWGF-based method produces a fusion image with blurring effects, such as the boundary between the focused and defocused parts around the stone lion. Compared with other algorithms, we can clearly see that the boundary area blurring obtained by the MWGF-based method is more serious (see Fig.4(c)). So, the MWGF-based algorithm often cannot achieve the ideal fusion image from the source image.

It can be seen from Fig.4(d) that two black spots appear on the left side of the stone, indicating that the integration of the SSDI-based method is not sufficient. Similarly, it can be seen from Fig.4(f) that the fused image obtained by the DSIFT-based method has many jagged phenomena in the lower right corner of the stone lion. Fig.4(e) and (g) show that the fusion images of CNN and ACNN-based methods seem very good, and they are relatively smooth in the boundary between the focused and defocused regions with respect to other methods. Because of the superiority of the proposed method, ACNN could accurately find the multi-focus boundary between the focused and defocused parts, and it can achieve better visual quality.

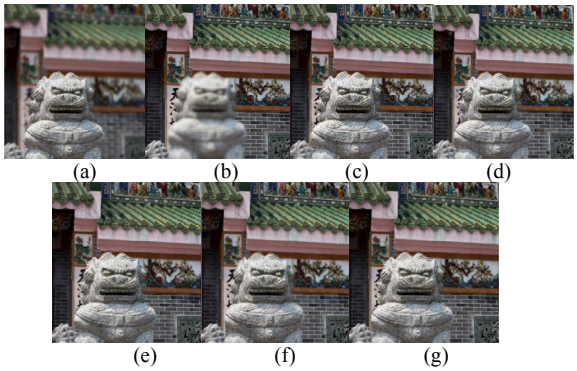


Fig.4 The fusion results of all algorithms on the 'Temple' image set: (a) (b) Source images; (c)—(g) Fusion results of MWGF, SSDI, CNN, DSIFT and ACNN algorithms, respectively

In order to prove the validity and practicability of the proposed algorithm, two indexes of mutual information (MI) and edge information retention ($Q^{AB/F}$) are used as the objective evaluation index of information fusion performance. Owing to the high focus detection accuracy of the ACNN model, we don't use any complicated post-processing technique in our fusion algorithm. Tab.1 lists the objective performance of the fused images using the five fusion methods. We can see that the ACNN-based method outperforms all the other fusion methods, which further verifies the effectiveness of ACNN for image fusion.

Tab.1 Comparison on objective criteria of different methods and multi-focus images

Image	Criteria	MWGF	SSDI	CNN	DSIFT	Proposed
Lab	MI	8.061 8	8.141 2	8.600 8	8.520 1	8.862 2
	$Q^{AB/F}$	0.714 7	0.752 8	0.757 3	0.758 5	0.757 4
Temple	MI	5.965 5	7.089 6	6.889 5	7.351 4	7.427 1
	$Q^{AB/F}$	0.750 1	0.763 4	0.759 0	0.764 3	0.759 7
Seascape	MI	7.140 4	7.482 4	7.628 5	7.948 7	8.027 6
	$Q^{AB/F}$	0.705 9	0.711 0	0.711 3	0.712 6	0.709 5
Book	MI	8.236 8	8.400 8	8.779 6	8.662 3	9.529 2
	$Q^{AB/F}$	0.724 0	0.726 0	0.727 7	0.713 4	0.760 1
Leopard	MI	9.947 4	10.888 7	10.879 2	10.922 6	10.947 4
	$Q^{AB/F}$	0.817 5	0.817 1	0.797 3	0.806 9	0.826 3
Children	MI	8.262 2	7.850 5	8.333 8	8.525 2	8.533 7
	$Q^{AB/F}$	0.674 1	0.679 9	0.740 8	0.739 4	0.736 4
Flower	MI	8.325 5	8.104 9	8.265 9	8.536 5	8.632 7
	$Q^{AB/F}$	0.691 3	0.649 0	0.718 3	0.715 9	0.715 0

In this letter, a new multi-focus image fusion method based on ACNN is presented, aiming to get a state-of-the-art decision map. The ACNN-based multi-focus image fusion method is developed in spatial domain. We demonstrate that the decision map obtained from the ACNN can lead to high-quality fusion results. Experimental results clearly validate that the proposed algorithm can obtain state-of-the-art fusion performance in terms of both qualitative and quantitative evaluations.

References

[1] Chaoben Du and Shesheng Gao, IEEE Access **5**, 15750 (2017).

[2] P. Burt and E. Adelson, IEEE Trans. Commun. **31**, 532 (1983).

[3] A. Toet, Pattern Recognit. Lett. **9**, 255 (1989).

[4] Li H, B. Manjunath and S. Mitra, Graphical Models Image Process. **57**, 235 (1995).

[5] J. Lewis, R. O. Callaghan, S. Nikolov, D. Bull and N. Canagarajah, Inf. Fusion **8**, 119 (2007).

[6] Zhang Q and Guo B, Signal Process. **89**, 1334 (2009).

[7] G. Piella, Inf. Fusion **4**, 259 (2003).

[8] Li X, Li H, Yu Z and Kong Y, Opt. Eng. **54**, 073 (2015).

[9] Liu Y, Liu S and Wang Z, Inf. Fusion **24**, 147 (2015).

[10] Liu Z, Chai Y, Yin H, Zhou J and Zhu Z, Inf. Fusion **35**, 102 (2017).

[11] W. Huang and Z. Jing, Pattern Recognit. Lett. **28**, 493 (2007).

[12] S. Li, J. Kwok and Y. Wang, Inf. Fusion **2**, 169 (2001).

[13] Li S, Kwok J and Wang Y, Pattern Recognit. Lett. **23**, 985 (2002).

[14] V. Aslantas and R. Kurban, Appl. **37**, 8861 (2010).

[15] De I and Chanda B, Inf. Fusion **14**, 136 (2013).

[16] Bai X, Zhang Y, Zhou F and Xue B, Inf. Fusion **22**, 105 (2015).

[17] Li M, Cai W and Tan Z, Pattern Recognit. Lett. **27**, 1948 (2006).

[18] Li S and Yang B, Image Vis. Comput. **26**, 971 (2008).

[19] Li S, Kang X and Hu J, IEEE Trans. **22**, 2864 (2015).

[20] Liu Y, Liu S and Wang Z, Inf. Fusion **23**, 139 (2013).

[21] Zhou Z, Li S and Wang B, Inf. Fusion **20**, 60 (2014).

[22] https://en.wikipedia.org/wiki/Deep _ learning

[23] Goodfellow Ian J, Wardefarley David, Mirza Mehdi, Courville Aaron and Bengio Yoshua, Maxout Networks, Computer Science,1319 (2013).

[24] Stollenga Marijn F, Masci Jonathan, Gomez Faustino and Schmidhuber Jurgen, Advances in Neural Information Processing Systems **4**, 3545 (2014).

[25] Zeiler M D and Fergus R, Stochastic Pooling for Regularization of Deep Convolutional Neural Networks, Eprint Arxiv, 2013.

[26] Lee Chen Y, Xie S, Gallagher Patrick, Zhang Z and Tu Z, Deeply Supervised Nets, In Deep Learning and Representation Learning Workshop, NIPS (2014).

[27] JT Springenberg, A Dosovitskiy, T Brox and M Riedmiller, Striving for Simplicity: The All Convolutional Net, Eprint Arxiv, 2014.

[28] Simonyan K and Zisserman A, Very Deep Convolutional Networks for Large-Scale Image Recognition, Computer Science, 2014.

[29] LeCun Y, Bottou L and Bengio Y, Proceedings of the IEEE **86**, 2278 (1998).

[30] Gulcehre C aglar, Cho KyungHyun, Pascanu Razvan and Bengio Yoshua, Learned-norm Pooling for Deep Feedforward and Recurrent Neural Networks, ECML (2014).

[31] Jia Y, Huang C and Darrell T, Beyond Spatial Pyramids: Receptive Field Learning for Pooled Image Features, Computer Vision and Pattern Recognition, IEEE, 3370 (2012).

[32] Behnke S, Hierarchical Neural Networks for Image Interpretation, Lecture Notes in Computer Science **2766**, 1345 (2003).

[33] Li S, Kwok J and Wang Y, Pattern Recognit. Lett. **23**, 985 (2002).

[34] Gao J and Xu L, Neural Process. Lett. **43**, 805 (2016).

[35] Liu Y, Chen X, Peng H and Wang Z, Inf. Fusion **36**, 191 (2017).

[36] Zhang Y, Bai X and Wang T, Inf. Fusion **35**, 81 (2017).

[37] Guo D, Yan J.W and Qu X, Opt. Commun. **338**, 138 (2015).