remote sensing

MDPI

Article

Infrared and Visible Image Fusion Method Based on a Principal Component Analysis Network and Image Pyramid

Shengshi Li [1], Yonghua Zou [1,2], Guanjun Wang [1,2,*] and Cong Lin [1]

[1] School of Information and Communication Engineering, Hainan University, Haikou 570228, China
[2] State Key Laboratory of Marine Resource Utilization in South China Sea, Hainan University, Haikou 570228, China
* Correspondence: wangguanjun@hainanu.edu.cn

Abstract: The aim of infrared (IR) and visible image fusion is to generate a more informative image for human observation or some other computer vision tasks. The activity-level measurement and weight assignment are two key parts in image fusion. In this paper, we propose a novel IR and visible fusion method based on the principal component analysis network (PCANet) and an image pyramid. Firstly, we use the lightweight deep learning network, a PCANet, to obtain the activity-level measurement and weight assignment of IR and visible images. The activity-level measurement obtained by the PCANet has a stronger representation ability for focusing on IR target perception and visible detail description. Secondly, the weights and the source images are decomposed into multiple scales by the image pyramid, and the weighted-average fusion rule is applied at each scale. Finally, the fused image is obtained by reconstruction. The effectiveness of the proposed algorithm was verified by two datasets with more than eighty pairs of test images in total. Compared with nineteen representative methods, the experimental results demonstrate that the proposed method can achieve the state-of-the-art results in both visual quality and objective evaluation metrics.

Keywords: image fusion; principal component analysis network; lightweight deep learning network; image pyramid; infrared image

Citation: Li, S.; Zou, Y.; Wang, G.; Lin, C. Infrared and Visible Image Fusion Method Based on a Principal Component Analysis Network and Image Pyramid. *Remote Sens.* **2023**, *15*, 685. https://doi.org/10.3390/rs15030685

Academic Editors: Riccardo Roncella and Mattia Previtali

Received: 13 December 2022
Revised: 15 January 2023
Accepted: 17 January 2023
Published: 24 January 2023

1. Introduction

An infrared (IR) sensor reflects the temperature or thermal radiation differences in a scene and captures thermal radiation objects in the dark or in smoke. However, IR images suffer from inconspicuous details, low contrast, and poor visibility. On the contrary, visible images can clearly show the detailed information of objects and have higher spatial resolution under great lighting conditions. For objects in poor lighting conditions or behind smoke, visible images barely capture useful information. Therefore, the purpose of IR and visible image fusion is to fuse the complementary features of two different modal images to generate an image with clear IR objects and a pleasing background, helping people understand the comprehensive information of the scene. The fusion of IR and visible images has many applications in military and civilian settings, such as video surveillance, object recognition, tracking, and remote sensing [1,2].

In recent years, the fusion of IR and visible images has become an active topic in the field of image processing. Various image-fusion methods have been proposed one after another, which are mainly divided into multi-scale transform (MST) methods, sparse representation (SR) methods, saliency methods, and deep-learning methods.

For MST methods, the source images are firstly decomposed in multiple scales and then fused by artificially designed fusion rules in different scales, and finally, the fused image is obtained via reconstruction. An MST fusion method can decompose the source images into different scales and extract more information to represent the source images. The disadvantage of an MST is that it often relies on artificially designed complex fusion

rules. The representative examples are the Laplacian pyramid (LP) [3], multi-resolution singular value decomposition (MSVD) [4], discrete wavelet transform (DWT) [5], dual-tree complex wavelet transform (DTCWT) [6], curvelet transform (CVT) [7], and target-enhanced multiscale transform decomposition (TE-MST) [8].

The SR method firstly learns an over-complete dictionary, then performs sparse coding on each sliding window block in the image to obtain sparse representation coefficients, and finally, reconstructs the image through the over-complete dictionary. The SR methods are robust to noise but usually have low computational efficiency. The representative examples are joint sparsity model (JSM) [9], joint sparse representation (JSR) [10], and joint sparse representation based on saliency detection (JSRSD) [11].

The saliency-based methods mainly perform fusion and reconstruction by extracting weights in salient regions of the image, such as weighted least squares (WLS) [12] and classification saliency-based rule for fusion (CSF) [13]. The advantage of saliency fusion methods is highlighting salient regions in the fused image, and the disadvantage is that saliency-based fusion rules are usually complicated.

In recent years, deep learning has been used for fusion tasks due to its powerful feature extraction capability. In [14], CNN was first used for multi-focus image fusion. Subsequently, in [15,16], a CNN was applied for IR and visible-image fusion, and for IR and medical-image fusion. For these two CNN-based fusion methods, the authors designed fusion rules based on three different situations. In addition, Li [17] et al. proposed a deep-learning method based on a pre-trained VGG-19, and adopted the fusion rules of the l_1-norm and weighted averages. In [18], Li et al. developed a fusion method based on a pre-trained ResNet and applied the fusion rules of zero-phase component analysis (ZCA) and the l_1-norm. Recently, more and more deep-learning fusion methods based on generative adversarial networks have been proposed. Ma et al. [19] proposed a fusion model, FusionGAN, based on a generative adversarial network, and applied a discriminator to continuously optimize the generator to generate the fusion result. The authors [20] presented a generative adversarial network with a dual-discriminator conditional, named DDcGAN, which aims to keep the thermal radiation in the IR image and the texture details in the visible image at the same time. Ma et al. [21] developed a generative adversarial network with multi-classification constraints (GANMcC) to transform the fusion problem into a multi-distribution estimation problem. Although these fusion algorithms have achieved good fusion results, they cannot effectively extract and combine the complementary features of IR and visible images.

Therefore, we propose a novel IR and visible fusion method based on a principal component analysis network (PCANet) [22] and an image pyramid [3,23]. The PCANet is trained to encode a direct mapping from source images to the weight maps. In this way, weight assignment can be obtained by performing activity-level measurement via the PCANet. Since the human visual system processes information in a multi-resolution way [24], a fusion method based on multi-resolution can produce fewer undesirable artifacts and make the fusion process more consistent with human visual perception [15]. Therefore, we used an image-pyramid-based framework to fuse IR and visible images. Compared with other MST methods, the running time of image-pyramid decomposition is short, which can improve the computational efficiency of the entire method [8].

The proposed algorithm has the following contributions:

- We propose a novel IR and visible image fusion method based on a PCANet and image pyramid, aiming to perform activity-level measurement and weight assignment through the lightweight deep learning model PCANet. The activity-level measurement obtained by the PCANet has a strong representation ability by focusing on IR target perception and visible-detail description.
- The effectiveness of the proposed algorithm was verified by 88 pairs of IR and visible images in total and 19 competitive methods, demonstrating that the proposed algorithm can achieve state-of-the-art performance in both visual quality and objective evaluation metrics.

The rest of the paper is arranged as follows: Section 2 briefly reviews PCANet, image pyramids, and guided filters. The proposed IR and visible image fusion method is depicted in Section 3. The experimental results and analyses are shown in Section 4. Finally, this article is concluded in Section 5.

2. Related Work

In this section, for a comprehensive review of some algorithms most relevant to this study, we focus on reviewing PCANet, the image pyramid, and the guided filter.

2.1. Principal Component Analysis Network (PCANet)

A principal component analysis network (PCANet) [22] is a lightweight, unsupervised deep learning neural network mainly used for extracting features in images, and it can also be considered as a simplified version of a CNN. In a PCANet, the critical task is the training of PCA filter, which will be specifically introduced in the next section. A PCANet consists of three components: cascaded two-stage PCA, binary hashing, and block histograms:

(1) Cascaded two-stage PCA: We assume that the filter bank W^1 in the first stage of the PCANet includes L_1 filters $W_1^1, W_2^1, \ldots, W_{L_1}^1$, and the filter bank W^2 in the second stage contains L_2 filters $W_1^2, W_2^2, \ldots, W_{L_2}^2$. Firstly, the input sample I is convolved with the l-th filter W_l^1 of the first stage:

$$I^l = I * W_l^1, l = 1, 2, \ldots, L_1 \tag{1}$$

where $*$ represents the convolution operation. Then, I^l is convolved with the r-th filter W_r^2 of the second stage:

$$O^q = I^l * W_r^2, l = 1, 2, \ldots, L_1, r = 1, 2, \ldots, L_2, q = 1, 2, \ldots, L_1 L_2 \tag{2}$$

where O^q represents the output of I and $L_1 L_2$ stands for the amount of output images.

(2) Binary hashing: Next, O^q will be binarized, and then these binary matrices are converted to decimal matrices as:

$$T^l = \sum_{r=1}^{L_2} 2^{r-1} H\left(I^l * W_r^2\right) \tag{3}$$

where T^l is the l-th decimal matrix for I, and $H(\cdot)$ is a Heaviside step function, whose value is one for positive entries and zero otherwise.

(3) Block histograms: In this part, each $T^l, l = 1, \ldots, L_1$ is split into B blocks, and we compute the histograms of the decimal values in each block and concatenate whole B histograms into one vector $\mathrm{Bhist}\left(T^l\right)$. Following this encoding process, the input image I is transformed into a set of block-wise histograms. We ultimately acquire the feature vectors as:

$$f = \left[\mathrm{Bhist}\left(T^1\right), \ldots, \mathrm{Bhist}\left(T^{L_1}\right)\right]^T \in \mathbb{R}^{\left(2^{L_2}\right) L_1 B} \tag{4}$$

where f is the network output.

The advantages of a PCANet are twofold:

- In the training stage, the PCANet obtains the convolution kernel through PCA auto-encoding does not need to iterate calculations of the convolution kernel like other deep learning methods.
- As a lightweight network, PCANet has only a few hyperparameters to be trained.

These two advantages make PCANet more efficient. PCANet has a wide range of applications in various fields, such as image recognition [22], object detection [25,26], image fusion [27], and signal classification [28,29].

2.2. Image Pyramids

An image pyramid [3,23] is a collection of images that consists of multiple sub-images of different resolutions of an image. In an image pyramid, the top-layer image has the lowest resolution, and the bottom-layer images have the highest resolution. Image pyramids include Gaussian pyramid and Laplacian pyramid [3].

In the Gaussian pyramid, we use I to represent the original image, that is, the 0-th layer Gaussian pyramid GP_0. We perform Gaussian filtering and interlaced subsampling on GP_0 to obtain the first layer of the Gaussian pyramid, GP_1. Repeat the above operations to obtain $GP_0, GP_1, \ldots, GP_h, \ldots, GP_N$ (where GP_h is the h-th layer of the Gaussian pyramid).

The Gaussian pyramid can be expressed as:

$$\begin{cases} GP_0 = I \\ GP_h = REDUCE(GP_{h-1}) \end{cases} \tag{5}$$

$$GP_h(i,j) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} GP_{h-1}(2i+m, 2j+n)s(m,n) \tag{6}$$

where (i,j) represents the coordinates in the image, $i \in [0, R_h - 1]$, $j \in [0, C_h - 1]$, and $h \in [1, N]$. N is the number of layers of Gaussian pyramid decomposition; R_h and C_h are the numbers of rows and columns of the h-th layer of the Gaussian pyramid, respectively; and $s(m,n)$ is a 2D separable 5×5 window function. By combining Equations (5) and (6), we can get the Gaussian pyramid image sequence $GP_0, GP_1, \ldots, GP_N$, and the upper layer is four times smaller than the lower layer.

On the other hand, we apply interpolation to enlarge the h-th layer Gaussian pyramid GP_h to obtain the image GP_h^*:

$$GP_h^* = EXPAND(GP_h) \tag{7}$$

where the size of GP_h^* is the same as that of GP_{h-1}. GP_h^* can be denoted as:

$$GP_h^*(i,j) = 4 \sum_{m=-2}^{2} \sum_{n=-2}^{2} GP_h\left(\frac{i+m}{2}, \frac{j+n}{2}\right) s(m,n) \tag{8}$$

where $h \in [1, N]$, $i \in [0, R_h - 1]$, $j \in [0, C_h - 1]$. When $(i+m)/2$ and $(j+n)/2$ are non-integers:

$$GP_h\left(\frac{i+m}{2}, \frac{j+n}{2}\right) = 0. \tag{9}$$

The expansion sequence $GP_1^*, GP_2^*, \ldots, GP_N^*$ can be obtained by Equations (7)–(9).

The Laplacian pyramid can be expressed as:

$$LP_h = GP_h - EXPAND(GP_{h+1}) \tag{10}$$

$$\begin{cases} LP_h = GP_h - G_{h+1}^*, h \in [0, N-1] \\ LP_N = GP_N, h = N \end{cases} \tag{11}$$

where $LP_0, LP_1, \ldots, LP_N$ represent Laplacian pyramid images, and LP_N is the top layer.

The inverse Laplacian pyramid transform (reconstruction) process can be obtained as follows:

$$\begin{cases} GP_N = LP_N \\ GP_h = LP_h + EXPAND(GP_{h+1}), h \in [0, N-1] \\ I = GP_0 \end{cases} \tag{12}$$

where I is the reconstructed image.

2.3. Guided Filter

A guided filter [30] is an edge filter based on a local linear model which does not need to perform convolution directly like most other filtering methods, and has simplicity, fast speed, and great edge-preservation. We define that filter output q is a linear transform of guidance image GI in a window ω_k centered on pixel k:

$$q_i = a_k GI_i + b_k, \forall i \in \omega_k \tag{13}$$

where a_k and b_k are the linear coefficients in ω_k.

To determine a_k and b_k, we minimize the difference between the filter output q and the filter input p, i.e., the cost function:

$$E(a_k, b_k) = \sum_{i \in \omega_k} \left((a_k GI_i + b_k - p_i)^2 + \epsilon a_k^2 \right) \tag{14}$$

where ϵ is a regularization parameter that serves to prevent a_k from being too large. With the above equation, we can enable the local linear model maximally similar to the input image p in ω_k.

a_k and b_k can be obtained by the following:

$$a_k = \frac{\frac{1}{|\omega|} \sum_{i \in \omega_k} GI_i p_i - \mu_k \overline{p}_k}{\sigma_k^2 + \epsilon} \tag{15}$$

$$b_k = \overline{p}_k - a_k \mu_k. \tag{16}$$

In the above equations, μ_k and σ_k^2 are the mean and variance of GI in ω_k, $|\omega|$ is the number of pixels in ω_k, and $\bar{p}_k = \frac{1}{|\omega|} \sum_{i \in \omega_k} p_i$ is the mean of p in ω_k.

We employ this linear model to all the local windows of the input image, but these windows are overlapped, and their centers are located in ω_k. Thus, the filter output is averaged over all possible q_i values by:

$$q_i = \overline{a}_i GI_i + \overline{b}_i \tag{17}$$

where $\bar{a}_i = \frac{1}{|\omega|} \sum_{k \in \omega_i} a_k$ and $\bar{b}_i = \frac{1}{|\omega|} \sum_{k \in \omega_i} b_k$ are the mean coefficients acquired from whole the overlapped windows, including pixel i.

Guided filter is performed by combining Equations (13)–(17), which can be simply denoted by:

$$q = GuidedFilter(GI, p) \tag{18}$$

where p indicates the input image, GI denotes the guidance image, and q represents the filter output.

3. The Proposed Method

We propose a novel IR and visible fusion method based on PCANet and an image pyramid. The activity-level measurement and weight assignment are two key parts of image fusion. We used PCANet to perform activity-level measurement and weight assignment because PCANet has stronger representation ability by focusing on IR target perception and visible detail description. Due to the human visual system processing information in a multi-resolution way [24], we apply an image pyramid to decompose and merge the images at multiple scales in order to make the fused image details appear more suitable for human visual perception.

3.1. Overview

The proposed algorithm is exhibited in Figure 1. Our method consists of four steps: PCANet initial weight map generation, spatial consistency, image-pyramid decomposition and fusion, and reconstruction. In the first step, we feed the two source images into PCANet and get the initial weight maps. In the second step, we take advantage of the spatial consistency to improve the quality of initial weight maps. The third step is image-pyramid decomposition and fusion. On the one hand, the source images are multi-scale transformed through the Laplacian pyramids. On the other hand, the initial weight maps are decomposed into multiple scales through Gaussian pyramids, and the softmax operation is performed on each scale to obtain the weight maps of each layer. Then, the fused image of each scale is obtained through a weighted-average strategy. In the last step, the final fusion image is obtained by reconstructing the Laplacian pyramid.

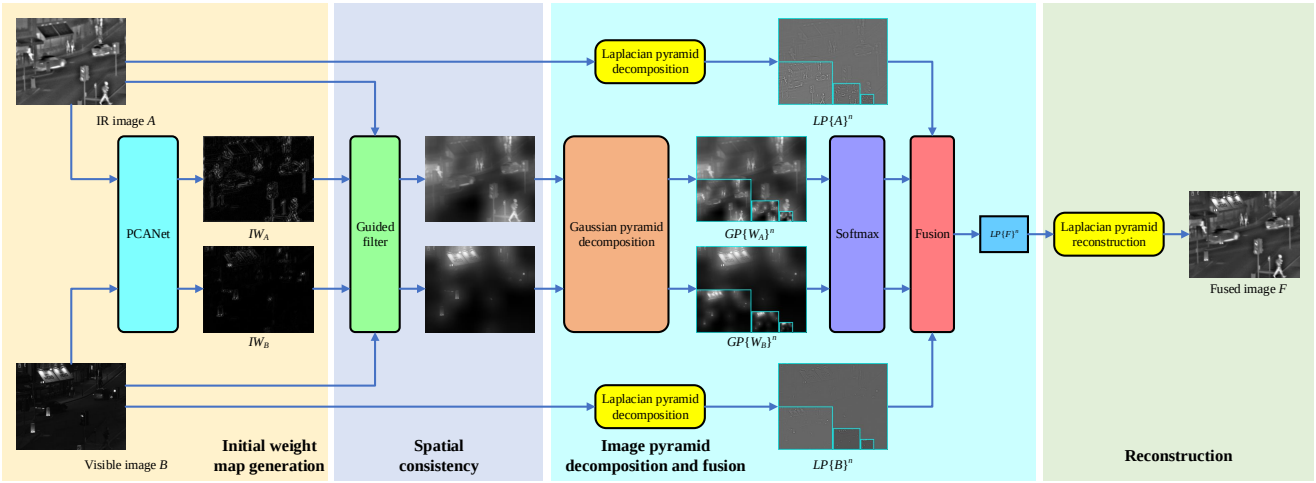


Figure 1. Schematic diagram of the proposed method.

3.2. PCANet Design

In our study, IR and visible image fusion is treated as a two-class classification problem. For each pixel from the same position of the source images, a scalar from 0 to 1 is output through PCANet to represent the probability of coming from different source images in the fused image. Standard PCANet contains cascaded two-stage PCA, binary hashing, and block histograms, where the role of the latter two components is to extract sparse features of images. If the network includes binary hashing and block histograms, the output sparse features have only two values of zero and one, and the size of the features are inconsistent with the source images. In our fusion task, in order to obtain more accurate probability values of the same position pixel from two source images and perform the fusion task faster, we only use cascaded two-stage PCA. The network design of PCANet is shown in Figure 2. In PCANet, the most important component is the PCA filter. In the next section, we describe the training process of the PCA filter in detail. In the PCANet framework, firstly, the input image is convolved with the first-stage PCA filter bank to obtain a series of feature maps. Then, these feature maps are convolved with the second-stage PCA filter bank to obtain more feature maps. These feature maps represent the details of the input image on different objects. Particularly, the second-stage filters can extract more advanced features. Two-stage PCA is usually sufficient to obtain a great effect, and a deeper architecture does not necessarily lead to further improvements [22], so we selected cascaded two-stage PCA for our experiments.

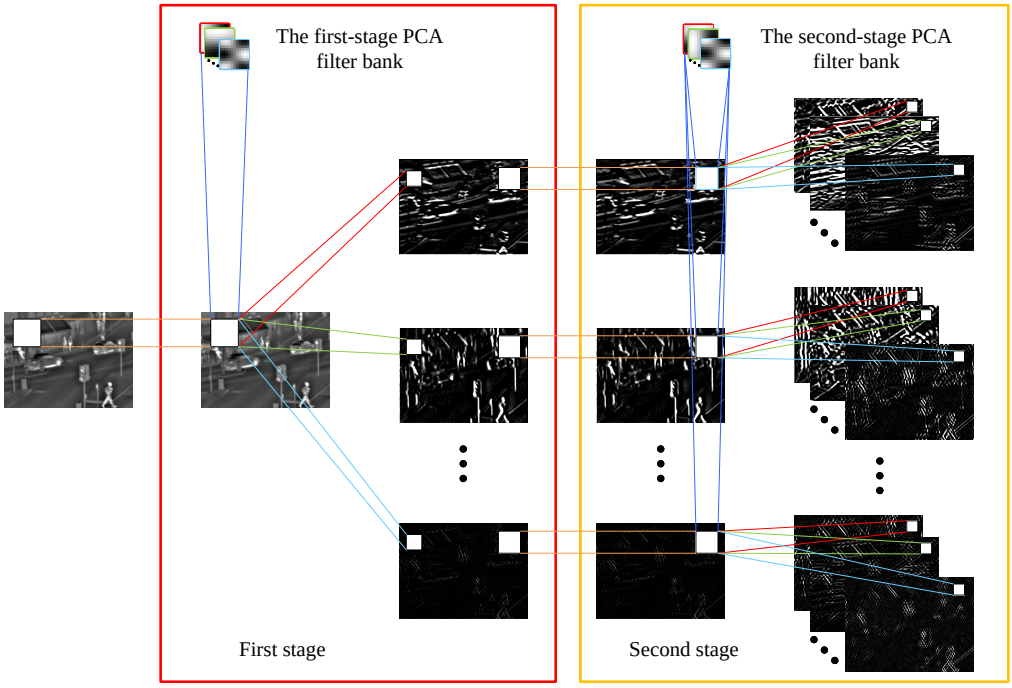


Figure 2. The PCANet model used in the proposed fusion method.

3.3. Training

The training of PCANet is essentially computing the PCA filter. PCA can be viewed as the simplest class of an auto-encoder, which minimizes reconstruction error [22]. We selected N images in the MS-COCO [31] database for training. In our experiments, we set N to 40,000. Considering that the size of each image in the MS-COCO database is different, each training image was converted into a 256×256 grayscale image. The training process of PCANet consisted of calculating two-stage PCA filter banks, and we assume that each filter size was $k_1 \times k_2$ in both stages. In the following, we describe the training process of each stage in detail.

- The First Stage

In order to facilitate the convolution operation, each training image is preprocessed. Preprocessing contains two steps: (1) Each sliding $k_1 \times k_2$ patch in the i-th training image I_i was converted into a column of X_i, where $X_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,\tilde{m}\tilde{n}}] \in \mathbb{R}^{k_1 k_2 \times \tilde{m}\tilde{n}}$, $i = 1, 2, \ldots N$, $\tilde{m} = 256 - k_1 + 1$, $\tilde{n} = 256 - k_2 + 1$. (2) The patch mean is subtracted from each column in X_i to obtain $\bar{X}_i = [\bar{x}_{i,1}, \bar{x}_{i,2}, \ldots, \bar{x}_{i,\tilde{m}\tilde{n}}] \in \mathbb{R}^{k_1 k_2 \times \tilde{m}\tilde{n}}$.

After the above preprocessing, we perform the same operation on N training images to obtain $X = [\bar{X}_1, \bar{X}_2, \ldots, \bar{X}_N] \in \mathbb{R}^{k_1 k_2 \times N\tilde{m}\tilde{n}}$. Then, we compute the covariance matrix C_1 of X:

$$C_1 = \frac{XX^T}{N\tilde{m}\tilde{n}}. \tag{19}$$

Next, by calculating the eigenvalue Λ_1 and eigenvector Q_1 of the covariance matrix C_1, we can obtain:

$$C_1 = Q_1 \Lambda_1 Q_1^T \tag{20}$$

where Λ_1 is a diagonal matrix with $k_1 k_2$ eigenvalues on the diagonal. Each column in Q_1 indicates an eigenvector corresponding to the eigenvalue in Λ_1, that is, the PCA filter. Particularly, the larger the eigenvalue, the more important the corresponding principal component. Therefore, we select the eigenvectors corresponding to the top L_1 largest eigenvalues as the PCA filters. Accordingly, the l-th PCA filter can be expressed as $W_l^1, l = 1, 2, \ldots, L_1$. Clearly, the PCA filter bank of the first stage is denoted as:

$$\mathrm{W}^1 = \left[\mathrm{W}_1^1, \mathrm{W}_2^1, \ldots, \mathrm{W}_{L_1}^1\right]. \tag{21}$$

Actually, the role of the PCA filter bank is to capture the main changes in the input image [22]. Next, we zero-pad the height and width boundaries of the i-th image I_i with size $k_1 - 1$ and size $k_2 - 1$, respectively, so that the convolution outputs have the same size as the source image. Then, I_i is preprocessed to obtain $\bar{I}_i$. The $\bar{I}_i$ is convolved with the l-th PCA filter in the first stage to obtain:

$$T_i^l = \bar{I}_i * \mathrm{W}_l^1, i = 1, 2, \ldots, N, l = 1, 2, \ldots, L_1 \tag{22}$$

where $*$ represents the convolution operation and T_i^l indicates an input sample of the second stage.

- The Second Stage

Firstly, almost the same as the first stage, the input image T_i^l is preprocessed to obtain $\bar{Y}_i^l = [\bar{y}_{i,l,1}, \bar{y}_{i,l,2}, \ldots, \bar{y}_{i,l,\tilde{m}\tilde{n}}] \in \mathbb{R}^{k_1 k_2 \times \tilde{m}\tilde{n}}$, and then the i-th input image is represented as $Y_i = \left[\bar{Y}_i^1, \bar{Y}_i^2, \ldots, \bar{Y}_i^{L_1}\right] \in \mathbb{R}^{k_1 k_2 \times L_1 \tilde{m}\tilde{n}}$. Performing the same for all N input images, we obtain $Y = [Y_1, Y_2, \ldots, Y_N] \in \mathbb{R}^{k_1 k_2 \times N L_1 \tilde{m}\tilde{n}}$. Next, similar to the first stage, we compute the covariance matrix C_2 of Y:

$$C_2 = \frac{YY^T}{NL_1\tilde{m}\tilde{n}} \tag{23}$$

$$C_2 = Q_2 \Lambda_2 Q_2^T \tag{24}$$

where Λ_2 denotes the eigenvalues of the second stage, and Q_2 represents the eigenvectors of the second stage. We select the eigenvectors corresponding to the top L_2 largest eigenvalues as the filter bank of the second stage. Therefore, the r-th PCA filter in the second stage can be denoted as $\mathrm{W}_r^2, r = 1, 2, \ldots, L_2$. The PCA filter bank of the second stage is indicated as:

$$\mathrm{W}^2 = \left[\mathrm{W}_1^2, \mathrm{W}_2^2, \ldots, \mathrm{W}_{L_2}^2\right]. \tag{25}$$

Up till this point, the two-stage filter banks W^1 and W^2 of PCANet have been obtained. The difference between the filters of the two stages is that the second-stage filters can extract higher-level features than the first-stage.

3.4. Detailed Fusion Scheme

3.4.1. PCANet Initial Weight Map Generation

Let the input image A indicate an IR image and B represent a visible image, and they are pre-registered images with the same size. Assume that each PCA filter size is $k_1 \times k_2$ in both stages. Firstly, we zero-pad the height and width boundaries of A and B with size $k_1 - 1$ and size $k_2 - 1$, respectively, so that the convolution outputs have the same size as the source images. Next, the input image S, $S \in \{A, B\}$, takes advantage of the preprocessing to obtain $\bar{S} \in \mathbb{R}^{k_1 k_2 \times \tilde{m}\tilde{n}}$. The $\bar{S}$ is convolved with the l-th PCA filter in the first stage:

$$T_S^l = \bar{S} * \mathrm{W}_l^1, l = 1, 2, \ldots, L_1. \tag{26}$$

Through the first-stage filter bank W^1, the first-stage PCANet outputs a total of L_1 feature maps $T_S^1, T_S^2, \ldots, T_S^{L_1}$.

The second stage is similar to the first stage. Firstly, zero-padding is performed in each T_S^l, and then preprocessing is taken to obtain $\bar{U}_S^l \in \mathbb{R}^{k_1 k_2 \times \tilde{m}\tilde{n}}$. Next, $\bar{U}_S^l$ is convolved with the r-th PCA filter in the second stage:

$$O_S^q = \bar{U}_S^l * \mathrm{W}_r^2, l = 1, 2, \ldots, L_1, r = 1, 2, \ldots, L_2, q = 1, 2, \ldots, L_1 L_2. \tag{27}$$

The second-stage PCANet outputs a total of $L_1 L_2$ feature maps $O_S^1, O_S^2, \ldots, O_S^{L_1 L_2}$.

Next, we define the initial weight maps for IR image A and visible image B as IW_A and IW_B:

$$IW_A(x, y) = O_A^1(x, y) + O_A^2(x, y) + \ldots + O_A^{L_1 L_2}(x, y) \tag{28}$$

$$IW_B(x, y) = O_B^1(x, y) + O_B^2(x, y) + \ldots + O_B^{L_1 L_2}(x, y) \tag{29}$$

where x and y represent the coordinates of the pixels in the image. Particularly, IW_A and IW_B are the same size as the source images.

3.4.2. Spatial Consistency

Spatial consistency means that two adjacent pixels with similar brightness or color will have a greater probability of having similar weights [32]. The initial weight maps are in general noisy, which may create artifacts on the fused image. To improve the performance of fusion, the initial weight maps need to be further processed. Specifically, we utilize a guided filter [30] to improve the quality of the initial weight maps. The guided filter is a very effective edge-preserving filter which can transform the structural information of the guided image into the filtering result of the input image [30]. We adopt the source image S as the guidance image to guide the absolute value of the initial weight map for filtering:

$$IW_A = GuidedFilter(A, |IW_A|) \tag{30}$$

$$IW_B = GuidedFilter(B, |IW_B|) \tag{31}$$

where A and B represent guidance images. In guided filter, we experimentally set the local window radius to 50 and the regularization parameter to 0.1.

3.4.3. Image-Pyramid Decomposition and Fusion

We perform n-layer Gaussian pyramid decomposition [3,23] on IW_A and IW_B to obtain $GP\{W_A\}^n$ and $GP\{W_B\}^n$ according to Equations (5) and (6). Each pyramid decomposition layer is set to the value $\lfloor \log_2 \min(Hig, Wid) \rfloor$, where $Hig \times Wid$ is the spatial size of source images and $\lfloor \cdot \rfloor$ denotes the flooring operation. Then, $GP\{W_A\}^n$ and $GP\{W_B\}^n$ are fed into a 2-way softmax layer, which produces probability values for two classes, denoting the outcome of each weight assignment:

$$FW\{A(x, y)\}^n = \frac{e^{GP\{W_A(x,y)\}^n}}{e^{GP\{W_A(x,y)\}^n} + e^{GP\{W_B(x,y)\}^n}} \tag{32}$$

$$FW\{B(x, y)\}^n = \frac{e^{GP\{W_B(x,y)\}^n}}{e^{GP\{W_A(x,y)\}^n} + e^{GP\{W_B(x,y)\}^n}}. \tag{33}$$

The values of $FW\{A\}^n$ and $FW\{B\}^n$ are between zero and one, indicating the probabilities that A and B take values at the same position pixel point. After the above operations, the network can autonomously learn the features in the image and calculate the weight of each pixel, avoiding the complexity and subjectivity of manually designing the fusion rules.

In addition, we conduct n-layer Laplacian pyramid decomposition [3,23] on A and B to obtain $LP\{A\}^n$ and $LP\{B\}^n$ according to Equations (10) and (11). The number of the Laplacian pyramid's decomposition layers is the same as that of the Gaussian pyramid. It is noteworthy that $FW\{A\}^n$ and $FW\{B\}^n$ are the same sizes as $LP\{A\}^n$ and $LP\{B\}^n$. Then, the fused image $L\{F\}^n$ on each layer is obtained by the weighted-average rule:

$$L\{F(x,y)\}^n = FW\{A(x,y)\}^n \times LP\{A(x,y)\}^n + FW\{B(x,y)\}^n \times LP\{B(x,y)\}^n. \quad (34)$$

3.4.4. Reconstruction

Finally, we reconstruct the Laplacian pyramid $L\{F\}^n$ to obtain the fused image F according to Equation (12). The main steps of the proposed IR and visible image fusion method are summarized in Algorithm 1.

Algorithm 1 The proposed IR and visible image fusion algorithm.

Training phase

1. Initialize PCANet;
2. Calculate the first-stage PCA filter bank W^1 via Equations (19)–(22);
3. Calculate the second-stage PCA filter bank W^2 via Equations (23)–(25).

Testing (fusion) phase

Part 1: PCANet initial weight map generation

1. Feed IR image A and visible image B into PCANet to obtain the initial weight maps according to Equations (26)–(29);

Part 2: Spatial consistency

2. Perform guided filtering on the absolute values of IW_A and IW_B according to Equations (30) and (31);

Part 3: image-pyramid decomposition and fusion

3. Perform n-layer Gaussian pyramid decomposition on IW_A and IW_B to generate the results $GP\{W_A\}^n$ and $GP\{W_B\}^n$ according to Equations (5) and (6);
4. Perform softmax operation at each layer to obtain $FW\{A\}^n$ and $FW\{B\}^n$ according to Equations (32) and (33);
5. Perform n-layer Laplacian pyramid decomposition on A and B to obtain $LP\{A\}^n$ and $LP\{B\}^n$ according to Equations (10) and (11);
6. Apply the weighted-average rule on each layer to generate the result $L\{F\}^n$ according to Equation (34);

Part 4: Reconstruction

7. Reconstruct the Laplacian pyramid to obtain the fused image F according to Equation (12).

4. Experiments and Discussions

In this section, the two experimental datasets and thirteen objective quality metrics are introduced. Secondly, the effects of different sizes and various number of filters in our method are discussed. Thirdly, we verify the effectiveness of our algorithm through two ablation studies. Fourthly, the proposed algorithm is evaluated by using visual quality and objective evaluation metrics. We selected nineteen state-of-the-art fusion methods to compare with our algorithm. Finally, we show the computational efficiency of different algorithms. All our experiments were performed on Intel (R) Core (TM) i7-11700, 64 GB RAM, and MATLAB R2019a.

4.1. Datasets

In order to comprehensively verify the effectiveness of our algorithm, we selected two datasets of different scenes for experiments, namely, the TNO dataset [33] and the RoadScene [34] dataset. The TNO dataset consists of several hundred pairs of pre-registered IR and visible images, mainly including military-related scenes, such as camps, helicopters, fighter jets, and soldiers. We chose 44 pairs of images in TNO dataset as test images.

Figure 3 exhibits eight pairs of testing images of the TNO dataset, where the top row represents the IR images and the bottom row denotes the visible images.

Figure 3. Illustrations of 8 pairs of testing images of the TNO dataset.

Differently from the TNO dataset, the RoadScene dataset has 221 pairs of road-related IR and visible pre-registered images, mainly including scenes of rural roads, urban roads, and night roads. We selected 44 pairs of images in the RoadScene dataset as test images. Figure 4 exhibits eight pairs of testing images of the RoadScene dataset, where the top row indicates the IR images and the bottom row denotes the visible images.

Figure 4. Illustrations of 8 pairs of testing images of the RoadScene dataset.

4.2. Objective Image Fusion Quality Metrics

In order to verify the fusion effect of our algorithm, we selected 13 objective evaluation metrics to conduct experiments. In what follows, we precisely describe various evaluation metrics:

- Yang's metric Q_Y [35]: Q_Y is a fusion metric based on structural information, which aims to calculate the degree to which structural information is transferred from the source images into the fused image;
- Gradient-based metric Q_G [36]: Q_G provides a fusion metric of image gradient, which reflects the degree of edge information of the source images preserved in the fusion image;
- Structural similarity index measure $SSIM$ [37]: $SSIM$ is a fusion index based on structural similarity, which mainly calculates the structural similarity between the fusion result and the source images;
- FMI_w, FMI_{dct} and FMI_{pixel} [38] calculate wavelet features, discrete cosine, and feature mutual information (FMI), respectively;
- Modified fusion artifacts measure N_{abf} [39]: N_{abf} provides a fusion index that introduces noise or artifacts in the fused image, reflecting the proportion of noise or artifacts generated in the fused image;
- Piella's three metrics Q_S, Q_W, Q_E [40]: Piella's three measures are on the basis of the structural similarity between source images and the fused image;
- Phase-congruency-based metric Q_P [41]: Q_P calculates the degree to which salient features in the source images are transferred to the fused image, and it is based on the absolute measure of image features;
- Chen–Varshney metric Q_{CV} [42]: The metric Q_{CV} is based on the human vision system and can fit the results of human visual inspection well;
- Chen–Blum metric Q_{CB} [43]: Q_{CB} is a fusion metric based on human visual perception quality.

In the above metrics, except N_{abf} and Q_{CV}, the larger the values, the better the fusion performance. On the contrary, the smaller the values of N_{abf} and Q_{CV}, the better the fusion effect. Among all the metrics, $SSIM$, N_{abf}, and Q_{CB} are the most important.

4.3. Analysis of Free Parameters

PCANet is a lightweight network with only three free parameters: the number of first-stage filters L_1, the number of second-stage filters L_2, and the size of the filter. We set the filter sizes of the two stages to be the same. We used the 44 pairs of images in TNO dataset to perform parameter setting experiments. The fusion performance is calculated by the average values of 13 fusion metrics, and the best values are indicated in red.

4.3.1. The Effect of the Number of Filters

We discuss the effect of the number of filters on fusion performance. As shown in Table 1, we fixed the PCA filter size to 3×3, and then the number of first-stage filters L_1 and the number of second-stage filters L_2 were set to vary from 3 to 8. In PCANet, the number of L_1 and L_2 affects the feature extraction of input samples. A higher number of filters means that the model extracts more features. Table 1 shows the influences of different numbers of L_1 and L_2 on the fusion performance. When $L_1 = L_2 = 8$, the model obtains 10 best values. If the values of L_1 and L_2 are greater than eight, the model will take more time, and the value of $SSIM$ may be lower. We should keep the model as simple as possible, so we set $L_1 = L_2 = 8$.

Table 1. The effect of the number of filters. L_1 and L_2 denote the numbers of first-stage and second-stage filters, respectively.

L_1	L_2	Q_Y	Q_G	$SSIM$	FMI_w	FMI_{dct}	FMI_{pixel}	N_{abf}	Q_S	Q_W	Q_E	Q_P	Q_{CV}	Q_{CB}
3	3	0.6868	0.3662	0.7495	0.4168	0.3991	0.9079	0.0000	0.8019	0.7429	0.3432	0.3211	500.3300	0.4732
3	4	0.6874	0.3669	0.7495	0.4168	0.3991	0.9079	0.0000	0.8021	0.7432	0.3439	0.3212	497.5229	0.4733
4	4	0.6878	0.3676	0.7494	0.4169	0.3991	0.9080	0.0000	0.8024	0.7435	0.3447	0.3216	500.7296	0.4730
4	5	0.6886	0.3685	0.7494	0.4169	0.3991	0.9080	0.0000	0.8026	0.7439	0.3456	0.3219	500.0824	0.4734
5	5	0.6883	0.3681	0.7494	0.4169	0.3991	0.9080	0.0000	0.8026	0.7438	0.3454	0.3216	500.4060	0.4729
5	6	0.6885	0.3683	0.7494	0.4170	0.3992	0.9080	0.0000	0.8026	0.7437	0.3454	0.3216	500.6746	0.4731
6	6	0.6887	0.3685	0.7494	0.4170	0.3992	0.9080	0.0000	0.8028	0.7439	0.3456	0.3217	500.3191	0.4732
6	7	0.6888	0.3687	0.7494	0.4171	0.3993	0.9080	0.0000	0.8028	0.7439	0.3455	0.3217	500.0603	0.4735
7	7	0.6894	0.3692	0.7494	0.4171	0.3994	0.9080	0.0000	0.8030	0.7441	0.3463	0.3219	499.6389	0.4735
7	8	0.6897	0.3696	0.7494	0.4172	0.3994	0.9081	0.0000	0.8031	0.7442	0.3465	0.3218	499.4519	0.4733
8	8	0.6920	0.3726	0.7493	0.4175	0.3996	0.9081	0.0000	0.8042	0.7455	0.3489	0.3228	499.3465	0.4750

4.3.2. The Influence of Filter Size

In this experiment, we discuss the impact of filter size on fusion performance. In Table 2, we fixed $L_1 = L_2 = 8$, and then the sizes of PCA filters were set to 3×3, 5×5, 7×7, 9×9, and 11×11, respectively. In PCANet, different filter sizes affect receptive field and feature extraction. A larger filter size means that the model extracts more features. Table 2 exhibits the influence of different filter size on the fusion performance. One can see that the fusion performance is the best when the size of the PCA filter is 11×11.

Table 2. The effect of filter size.

Size	Q_Y	Q_G	$SSIM$	FMI_w	FMI_{dct}	FMI_{pixel}	N_{abf}	Q_S	Q_W	Q_E	Q_P	Q_{CV}	Q_{CB}
3×3	0.6920	0.3726	0.7493	0.4175	0.3996	0.9081	0.0000	0.8042	0.7455	0.3489	0.3228	499.3465	0.4750
5×5	0.7163	0.4020	0.7476	0.4195	0.3982	0.9108	0.0000	0.8131	0.7707	0.4152	0.3441	466.6789	0.4675
7×7	0.7511	0.4434	0.7432	0.4244	0.3932	0.9133	0.0000	0.8216	0.8008	0.5020	0.3776	449.1673	0.4755
9×9	0.7864	0.4786	0.7374	0.4306	0.3829	0.9150	0.0001	0.8251	0.8207	0.5659	0.4065	427.3843	0.4879
11×11	0.8238	0.5097	0.7299	0.4406	0.3719	0.9162	0.0002	0.8240	0.8277	0.5998	0.4333	407.4069	0.4959

Therefore, we set $L_1 = L_2 = 8$, and the PCA filter size was 11×11.

4.4. Ablation Study

In this part, we conducted two ablation studies to verify the effectiveness of the image pyramid and guided filter.

4.4.1. The Ablation Study of the Image Pyramid

Figure 5 shows the results of image pyramid ablation experiment. We compared the model with and without image pyramids regarding the fusion results. The first column represents the IR image, the second column denotes the visible image, the third column indicates the model without the image pyramid, and the fourth column represents the model with the image pyramid. Except the image pyramid, other parameters were the same. For the four examples, the fusion results for the model with the image pyramid are better than the fusion results for the model without the image pyramid. The fusion results for without the pyramid introduce some artifacts and noises, and the model with the pyramid almost eliminated these artifacts and the noise through multi-scale decomposition (see the red boxes in Figure 5).

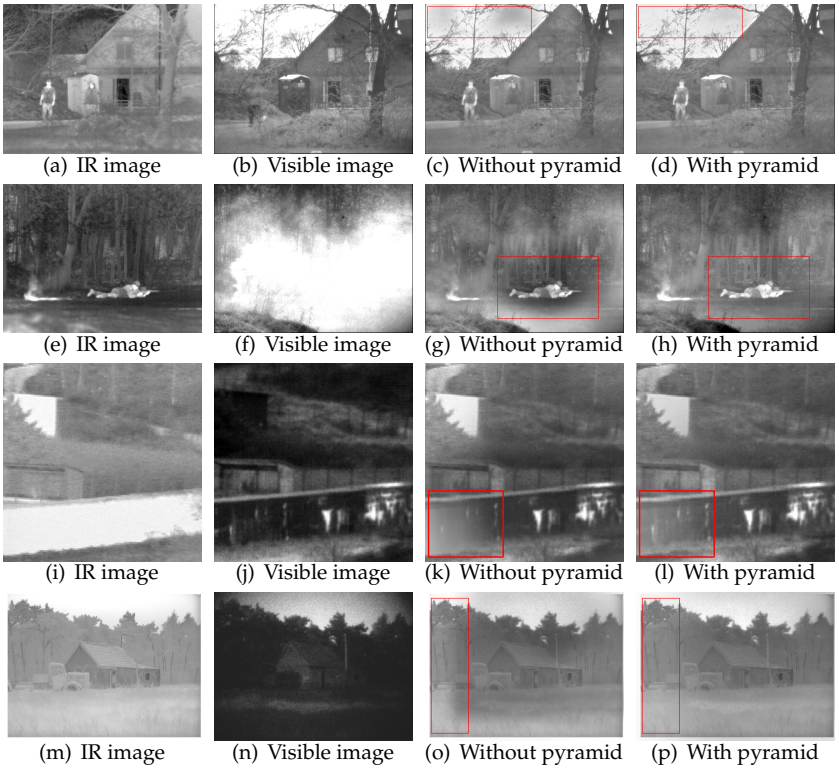


Figure 5. The ablation study of the image pyramid. The first column has IR images, the second column has visible images, the third column has images without the use of the image pyramid, and the fourth column has images with the use of the image pyramid.

We used the 44 pairs of images in the TNO dataset to verify the effect of the model with and without image pyramids. Table 3 shows the average value of each evaluation index and the fusion time for 44 pairs of images. The best values are indicated in red. The running times of the two models were almost the same, and the model with image pyramid obtained eight optimal values. Combined with visual quality and objective evaluation metrics, it is proved that the algorithm with the image pyramid is better.

Table 3. The average value of with and without image pyramids on the TNO dataset (unit: seconds).

Method	Q_Y	Q_G	$SSIM$	FMI_w	FMI_{dct}	FMI_{pixel}	N_{abf}	Q_S	Q_W	Q_E	Q_P	Q_{CV}	Q_{CB}	Time
With pyramid	0.8238	0.5097	0.7299	0.4406	0.3719	0.9162	0.0002	0.8240	0.8277	0.5998	0.4333	407.4069	0.4959	257.6713
Without pyramid	0.8218	0.4936	0.7308	0.4366	0.3649	0.9162	0.0012	0.8269	0.8314	0.5937	0.4326	360.0513	0.5008	251.6412

4.4.2. The Ablation Study of the Guided Filter

Figure 6 shows the results of the guided-filter ablation experiment. We compare the model with and without guided filtering. The first column has IR images, the second column has visible images, the third column has images produced without guided filtering, and the fourth column has images produced with guided filtering. All other parameter settings were the same. There are some obvious artifacts and noise in the red boxes in the third column of Figure 6. After guided filtering, these artifacts and the noise were eliminated. It can be seen in the figure that the fusion effect with guided filtering is better.

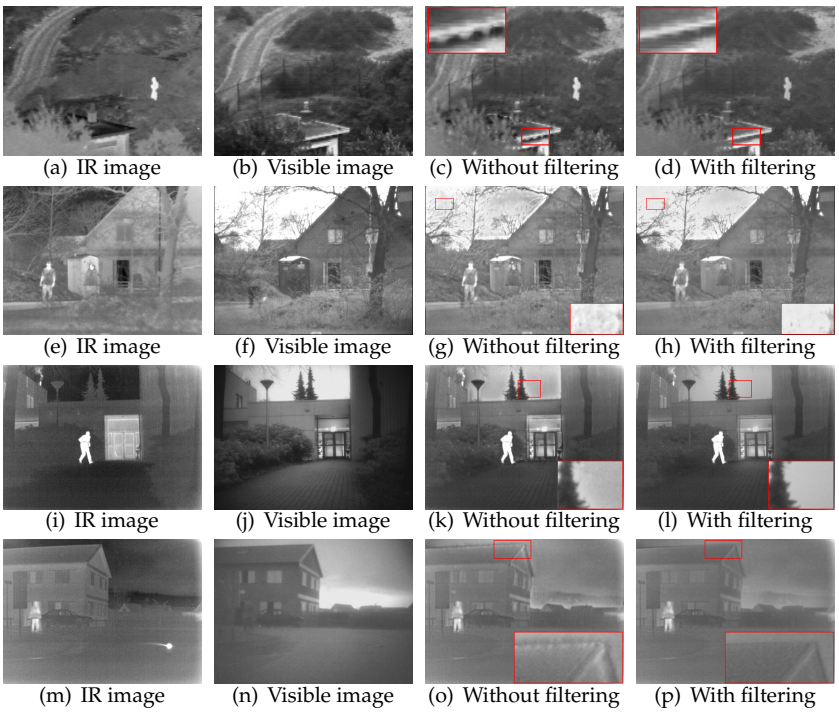


Figure 6. The ablation study of the guided filter. The first column has IR images, the second column has visible images, the third column has images produced without guided filtering, and the fourth column has images produced with guided filtering.

4.5. Experimental Results and Discussion

4.5.1. Comparison with State-of-the-Art Competitive Algorithms on the TNO Dataset

We used the TNO dataset to verify the performance of our algorithm. The competitive algorithms numbered 19: MST methods (MSVD [4], DWT [5], DTCWT [6], CVT [7], ML-GCF [44], and TE-MST [8]), SR methods (JSM [9], JSR [10], and JSRSD [11]), deep learning methods (FusionGAN [19], GANMcC [21], PMGI [45], RFN-Nest [46], CSF [13], DRF [47], FusionDN [34], and DDcGAN [20]), and other methods (GTF [48] and DRTV [49]). In particular, the comparative algorithms based on deep learning have been proposed in the last three years. The corresponding parameter settings in the comparison algorithms were set to the default values given by their authors.

In our approach, we set the filter size to 11×11 for both stages, and the number of filters to eight for both stages. The number of image-pyramid decomposition layers was n, $n = \lfloor \log_2 \min(Hig, Wid) \rfloor$, where $Hig \times Wid$ represents the size of the source images and $\lfloor \cdot \rfloor$ denotes the flooring operation. We set the radius of the guided filter to 50 and

the regularization parameter to 0.1. The fusion performance of the proposed method was evaluated by comparing the visual quality and objective evaluation metrics.

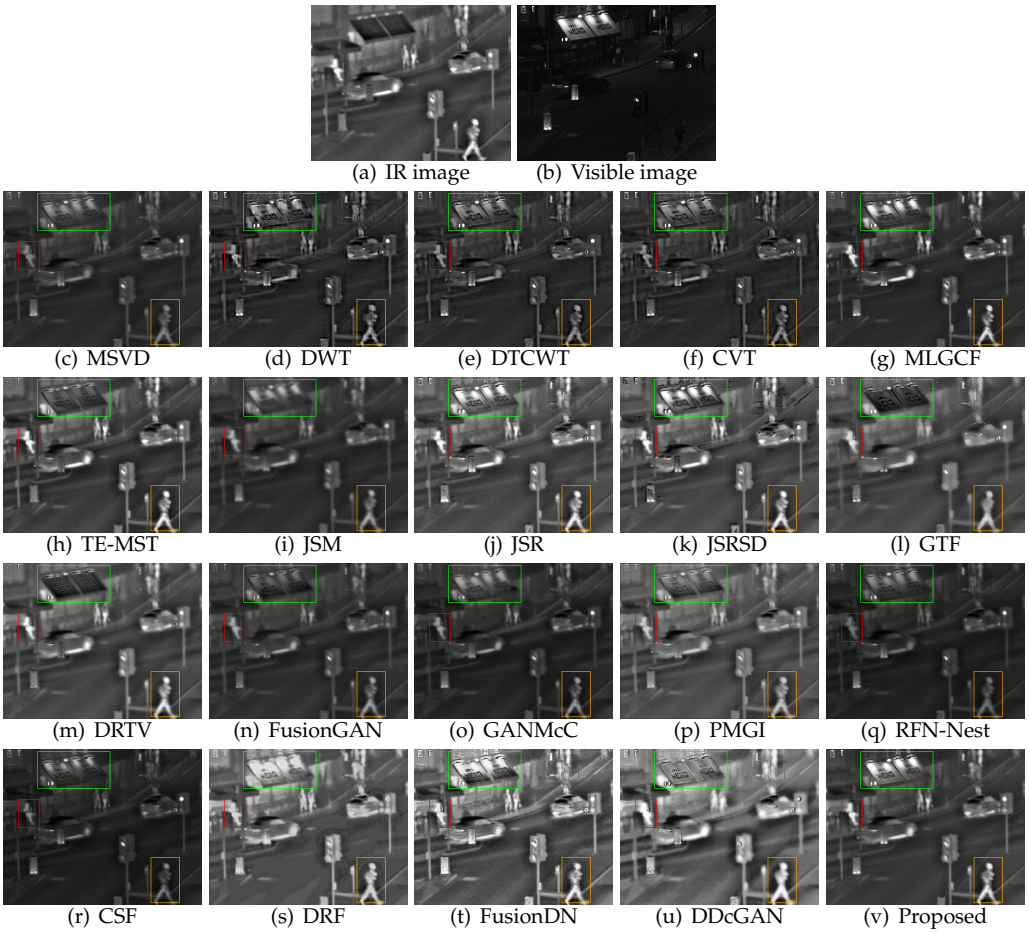
Figures 7 and 8 show two representative examples. For better comparison, some regions in the fused images are marked with rectangular boxes. Figure 7 shows the fusion results of "Queen Road" source images. The described nighttime scene includes rich content, containing pedestrians, cars, street lights, and shops. IR images exhibit thermal radiation information of pedestrians, vehicles, and street lights, while visible images provide clearer details, especially the details of the plate of storefront. The ideal fusion result of this example is to preserve the thermal radiation information in the IR image while extracting the details in the visible image. Pedestrians in the MSVD, DTCWT, and CVT methods suffer from low brightness and contrast (see red and orange boxes in Figure 7c,e,f). The DWT-based method introduces undesired small rectangular blocks (see three boxes in Figure 7d). Although the MLGCF algorithm can extract the thermal objects well, the whole image is too dark. The TE-MST technique has high fusion quality, but it introduces too much of the infrared spectrum to the plate of storefront, resulting in an unnatural visual experience (see the green box in Figure 7h). The plate of storefront in the JSM fusion result is clearly blurred (see the green box in Figure 7i). Although JSR and JSRSD schemes achieve a great fusion effect, their backgrounds lack some details. Both GTF and DRTV methods suffer from low fusion performance, especially the lack of details on the plate of storefront (see green boxes in Figure 7l,m). Among the deep-learning-based algorithms, the FusionGAN, GANMcC, PMGI, and RFN-Nest methods cannot extract the details of the plate of storefront well due to introducing too much of the infrared spectrum (see the green boxes in Figure 7n,o,p,q). The CSF technique cannot extract thermal radiation information well (see the red and orange boxes in Figure 7r). The DRF, FusionDN, and DDcGAN methods appear overexposed and introduce some undesired noise (see Figure 7s,t,u). Our algorithm can well extract thermal radiation objects in the IR image and details in the visible image with a more natural visual experience (see Figure 7v). Our algorithm has the stronger representation ability by focusing on IR target perception and visible detail description compared with other methods.

Figure 8 shows the fusion results of the "Kaptein" source images, which exhibit a person standing at a door. On the one hand, IR images mainly capture the thermal radiation information of person. On the other hand, the visible images clearly show the details of buildings, trees in the distance, and grass. The person after MSVD, DWT, DTCWT, and CVT methods suffers from low brightness and contrast. In particular, the DWT, DTCWT, and CVT algorithms produce some artifacts around the people. The MLGCF and TE-MST methods cannot well extract the details of the ground textures (see the orange boxes in Figure 8g,h). The JSM fusion result is blurry, and JSR and JSRSD schemes introduced some noise. The GTF and DRTV methods introduce artifacts around distant trees. Regarding the deep learning algorithms, the man after application of the FusionGAN and DDcGAN methods is blurry, and the person after the RFN-Nest and DRF methods has low brightness. These fusion results constitute an unnatural visual experience. In addition, the GANMcC, PMGI, and CSF methods cannot well capture the details of the sky and ground (see the orange and green boxes in Figure 8o,p,r). The FusionDN technique achieves high fusion performance. Compared to other methods, our method obtains better perceptual quality for the sky (see green box in Figure 8v), higher brightness of the thermal radiation objects (see red box in Figure 8v), and clearer ground textures (see orange box in Figure 8v).

Table 4 shows the averages of 13 objective evaluation metrics for the TNO dataset, and the best values are indicated in red. As can be seen in Table 4, except FMI_{dct} and Q_E, our algorithm obtained the best results for all metrics, indicating that our algorithm has excellent fusion performance.

Table 4. The average values of different methods on the TNO dataset.

Type	Method	Q_Y	Q_G	$SSIM$	FMI_w	FMI_{dct}	FMI_{pixel}	N_{abf}	Q_S	Q_W	Q_E	Q_P	Q_{CV}	Q_{CB}
MST	MSVD	0.6297	0.3274	0.7220	0.2683	0.2382	0.8986	0.0022	0.7735	0.7091	0.3107	0.2456	549.9197	0.4428
	DWT	0.7354	0.5042	0.6532	0.3678	0.2911	0.8970	0.0581	0.7632	0.7643	0.5499	0.2473	522.7137	0.4732
	DTCWT	0.7732	0.4847	0.6945	0.4127	0.3547	0.9122	0.0243	0.8019	0.8100	0.6361	0.3087	524.0247	0.4956
	CVT	0.7703	0.4644	0.6934	0.4226	0.4021	0.9095	0.0274	0.8017	0.8141	0.6365	0.2784	539.9093	0.4931
	MLGCF	0.7702	0.4863	0.7078	0.3717	0.3229	0.9009	0.0208	0.8063	0.8032	0.5694	0.2974	454.5477	0.4627
	TE-MST	0.7653	0.4503	0.7006	0.3749	0.3313	0.9075	0.0224	0.7775	0.7251	0.4518	0.2787	923.3319	0.4512
SR	JSM	0.2233	0.0830	0.6385	0.1404	0.1061	0.8928	0.0048	0.6076	0.3961	0.0057	0.0604	676.3967	0.3086
	JSR	0.6338	0.3392	0.6053	0.2208	0.1672	0.8839	0.0566	0.6858	0.7111	0.4051	0.2051	431.9517	0.4182
	JSRSD	0.5558	0.2981	0.5492	0.1981	0.1451	0.8632	0.1032	0.6322	0.6830	0.3389	0.1436	476.0037	0.4288
Other methods	GTF	0.6639	0.3977	0.6706	0.4301	0.4059	0.9045	0.0103	0.7168	0.6571	0.3439	0.1991	1161.7491	0.3984
	DRTV	0.5906	0.3012	0.6622	0.4104	0.4198	0.8888	0.0214	0.7098	0.6502	0.2111	0.1016	1348.3111	0.4202
Deep learning	FusionGAN	0.5263	0.2446	0.6430	0.3754	0.3565	0.8889	0.0131	0.6626	0.5842	0.1370	0.1076	963.9209	0.4115
	GANMcC	0.5976	0.3056	0.6824	0.3820	0.3512	0.8980	0.0099	0.7197	0.6771	0.2768	0.2506	674.4502	0.4369
	PMGI	0.7166	0.4040	0.6981	0.3948	0.3810	0.8996	0.0282	0.7771	0.7716	0.4566	0.2699	586.3804	0.4604
	RFN-Nest	0.6263	0.3453	0.6820	0.2976	0.2897	0.9032	0.0114	0.7345	0.7079	0.3010	0.2340	584.3049	0.4749
	CSF	0.6841	0.4136	0.6901	0.3007	0.2541	0.8826	0.0280	0.7578	0.7568	0.4753	0.2714	538.8530	0.4873
	DRF	0.4466	0.2024	0.6184	0.1694	0.1184	0.8866	0.0342	0.6400	0.5430	0.1025	0.0962	1004.4690	0.3941
	FusionDN	0.6856	0.3788	0.6230	0.3597	0.3097	0.8842	0.1356	0.7301	0.7467	0.4439	0.2678	633.9079	0.4935
	DDcGAN	0.6390	0.3364	0.5820	0.4114	0.3863	0.8760	0.1016	0.6530	0.5918	0.2060	0.1451	1017.1516	0.4360
	Proposed	0.8238	0.5097	0.7299	0.4406	0.3719	0.9162	0.0002	0.8240	0.8277	0.5998	0.4333	407.4069	0.4959



(a) IR image (b) Visible image (c) MSVD (d) DWT (e) DTCWT (f) CVT (g) MLGCF (h) TE-MST (i) JSM (j) JSR (k) JSRSD (l) GTF (m) DRTV (n) FusionGAN (o) GANMcC (p) PMGI (q) RFN-Nest (r) CSF (s) DRF (t) FusionDN (u) DDcGAN (v) Proposed

Figure 7. Fusion results of the "Queen Road" source images.

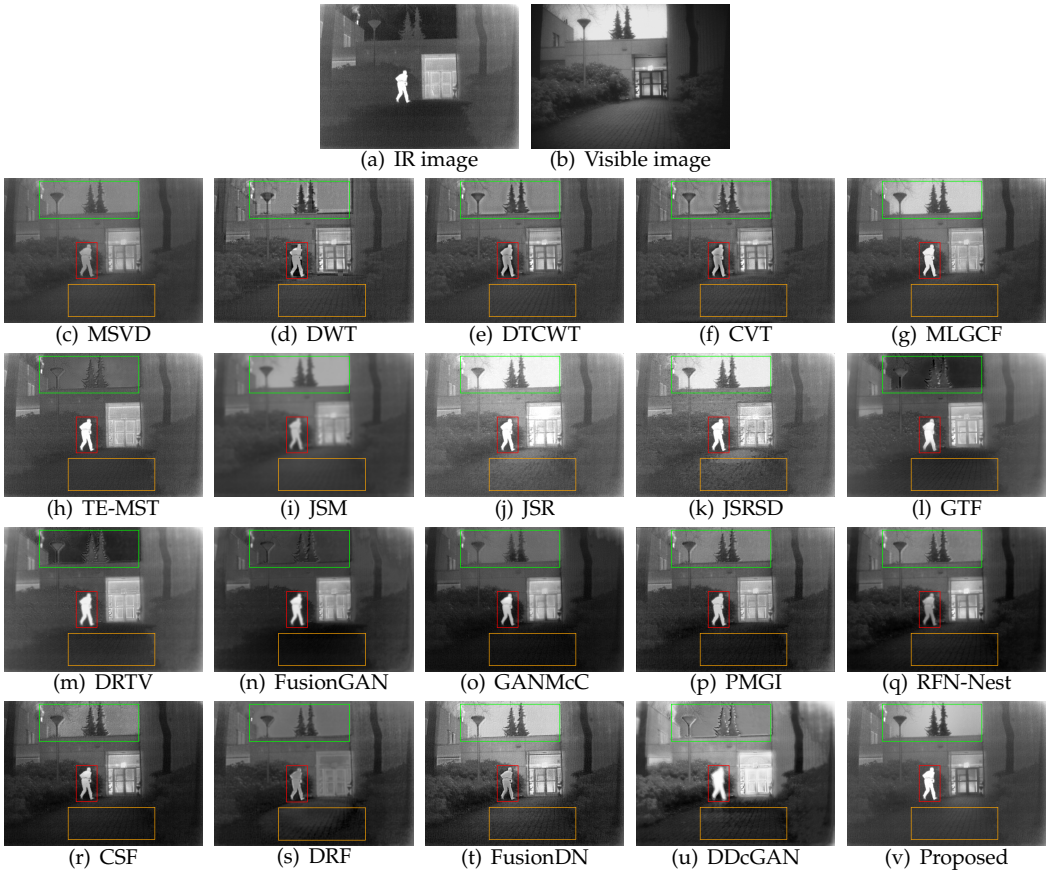


Figure 8. Fusion results of the "Kaptein" source images.

4.5.2. Further Comparison on the RoadScene Dataset

In order to verify the fusion performance in different scenes, we employed the RoadScene dataset for experiments. Figures 9 and 10 show two representative examples. Figure 9 exhibits the fusion results of "FLIR04602" source images. The scene shows a pedestrian standing on the side of the road and a car parked on the road during the daytime. The IR images mainly capture the thermal radiation information of pedestrian and car, and visible images show the details of buildings and trees. The pedestrian and car in the MSVD method lost brightness and contrast. The DWT method introduces undesired "small rectangles" (see car and buildings in Figure 9d). The trees in the DTCWT, CVT, MLGCF, and TE-MST methods introduce too many "small black spots" from the infrared spectrum, resulting in unnatural visual experience (see green boxes in Figure 9e–h). The fusion result of JSM method is noticeably blurry. The JSR and JSRSD results appear overexposed. In particular, the JSRSD method introduces a certain amount of noise. The pedestrian and car became blurry by the GTF and DRTV methods. Regarding deep-learning-based methods, the fusion results of FusionGAN, DDcGAN, and DRF appear blurry. Specifically, the pedestrian and car through FusionGAN and DDcGAN methods were blurred, and trees and buildings through the DRF method were blurred. Since this example is a daytime scene, most of the visible image details are required. Although GANMcC, PMGI, RFN-Nest, CSF, and FusionDN methods achieved a good fusion effect, too many "small black dots" from IR images were introduced into the trees, resulting in an unnatural visual experience (see green boxes in Figure 9o–r,t). Compared with other algorithms, our algorithm can extract the pedestrian and car in the IR images well, and the results look more natural.

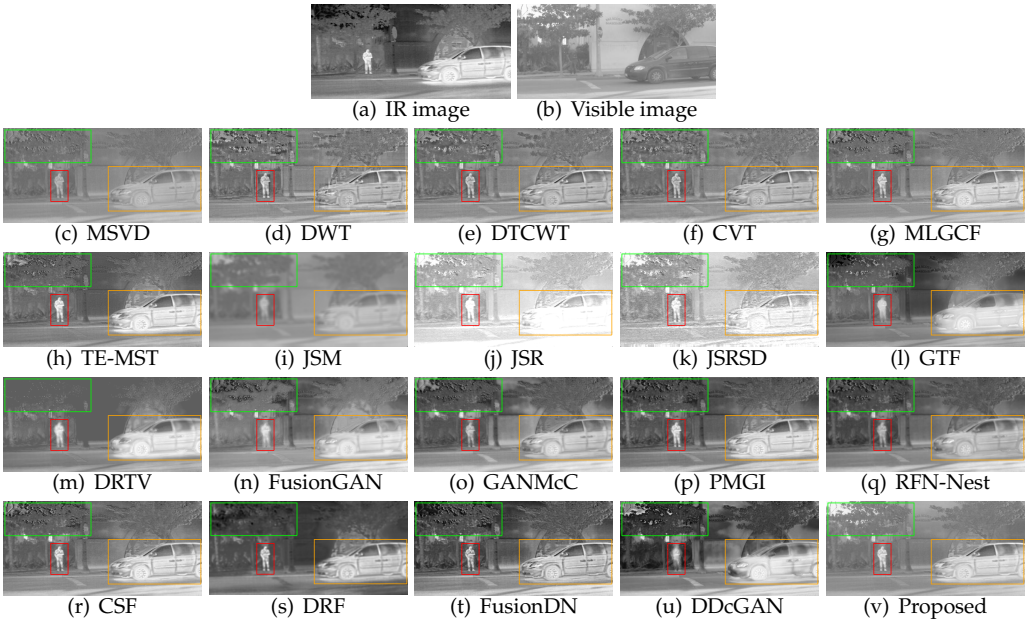


Figure 9. Fusion results of the "FLIR04602" source images.

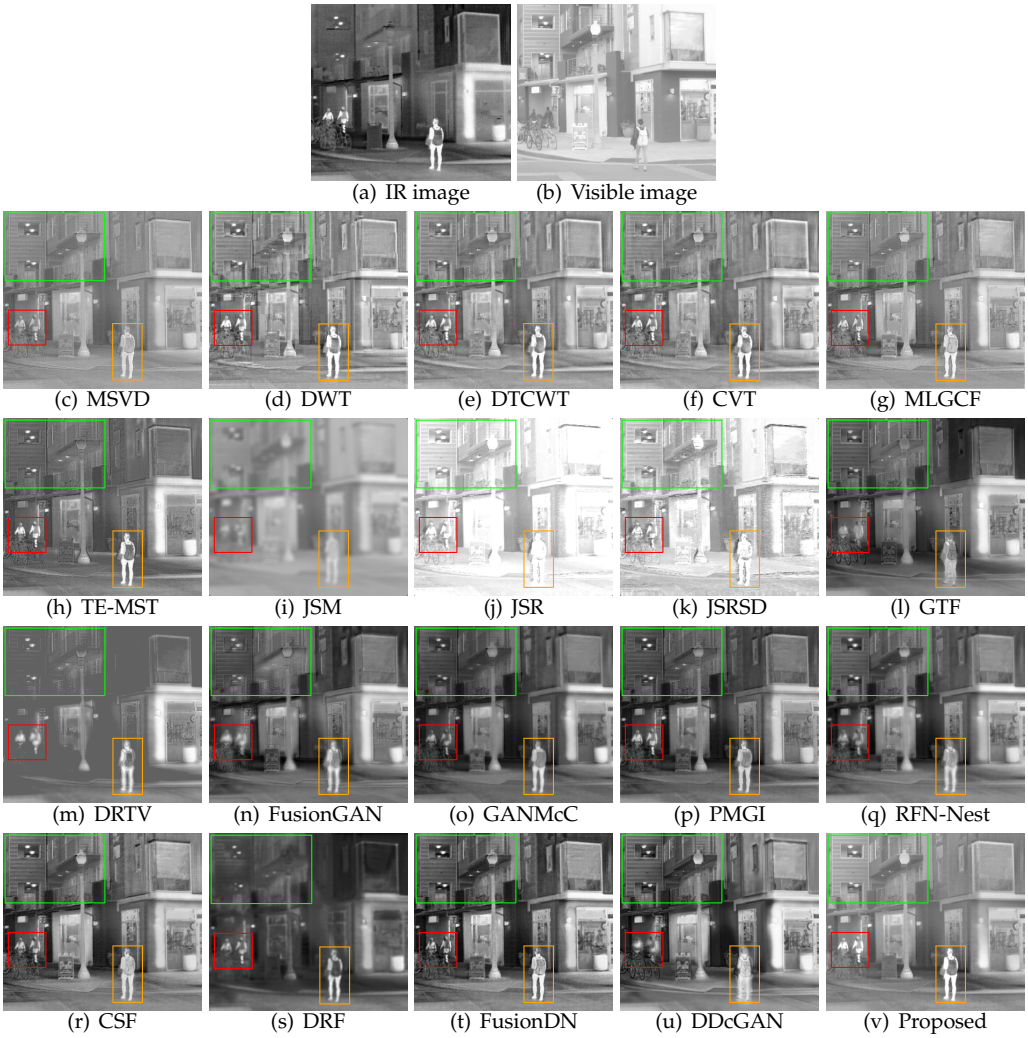


Figure 10. Fusion results of the "FLIR08835" source images.

Figure 10 shows the fusion results of "FLIR08835" source images. The described scene contains rich content, including pedestrians, a street, and buildings. On the one hand, the IR image mainly extracts the thermal radiation information of the pedestrians to better display the locations of pedestrians. On the other hand, visible image provides clearer background details. The MSVD algorithm cannot extract thermal radiation information well. The DWT, DTCWT, CVT, TE-MST, and MLGCF fusion results all introduce some noise. The JSM fusion result is blurry, and JSR and JSRSD methods appear overexposed. The GTF method achieved great fusion performance, and the background areas in the DRTV algorithm's images are obviously blurry (see the green box in Figure 10m). The pedestrians in the FusionGAN, DRF, RFN-Nest and DDcGAN algorithms' images are blurry (see red and orange boxes in Figure 10n,s,q,u). The CSF method introduced some noise into the background. The GANMcC, PMGI, and FusionDN schemes achieved high fusion performance. Based on the above observations, it is clear that our algorithm captures the thermal radiation information of pedestrians well and has a great fusion effect. It can be at least stated that our method achieves competitive performance with the GANMcC, PMGI, and FusionDN methods.

Table 5 shows the averages of 13 objective evaluation metrics for the RoadScene dataset and the best values are indicated in red. It can be seen in Table 5 that, except for Q_W, Q_E, Q_{CV}, and Q_{CB}, the proposed fusion method achieved the best results for all other metrics.

Overall, it was found that the 19 competitive algorithms all suffer from some defects. Considering the above comparisons in relation to visual quality and objective evaluation metrics together, our algorithm can generally outperform other methods, leading to state-of-the-art fusion performance.

Table 5. The average values of different methods on the RoadScene dataset.

Type	Method	Q_Y	Q_G	$SSIM$	FMI_w	FMI_{dct}	FMI_{pixel}	N_{abf}	Q_S	Q_W	Q_E	Q_P	Q_{CV}	Q_{CB}
MST	MSVD	0.6703	0.3694	0.7239	0.2724	0.2225	0.8571	0.0030	0.7723	0.6920	0.3006	0.3122	808.7879	0.4781
	DWT	0.7732	0.5673	0.6455	0.4015	0.2677	0.8623	0.0496	0.7721	0.7757	0.5732	0.3266	769.0781	0.4922
	DTCWT	0.7517	0.4625	0.6645	0.3584	0.2415	0.8589	0.0386	0.7752	0.7610	0.4769	0.3255	800.1602	0.4976
	CVT	0.7990	0.4975	0.6785	0.4353	0.3738	0.8738	0.0277	0.8057	0.8068	0.6127	0.3523	982.3925	0.5075
	MLGCF	0.8136	0.5395	0.7064	0.3604	0.2783	0.8600	0.0174	0.8252	0.7899	0.5449	0.3732	795.6147	0.4647
	TE-MST	0.8534	0.5855	0.6983	0.4091	0.3093	0.8751	0.0199	0.8210	0.7799	0.5416	0.4262	981.4404	0.5305
SR	JSM	0.2689	0.0983	0.6011	0.1538	0.1060	0.8426	0.0044	0.5105	0.2606	0.0008	0.0789	752.1129	0.2918
	JSR	0.4876	0.2678	0.5774	0.1955	0.1601	0.8292	0.0389	0.6192	0.6128	0.2610	0.2039	591.9430	0.3618
	JSRSD	0.4595	0.2499	0.4937	0.1777	0.1437	0.8196	0.0859	0.5540	0.6420	0.2871	0.1442	509.1361	0.4136
Other methods	GTF	0.6671	0.3007	0.6820	0.3755	0.3742	0.8721	0.0077	0.6782	0.5256	0.1842	0.2495	1595.9816	0.3950
	DRTV	0.5268	0.2310	0.6695	0.3379	0.3704	0.8478	0.0168	0.6883	0.5930	0.1187	0.1313	1672.9384	0.4308
Deep learning	FusionGAN	0.4997	0.2381	0.6025	0.3169	0.3312	0.8529	0.0151	0.6179	0.5254	0.1181	0.1387	1138.3050	0.4551
	GANMcC	0.6350	0.3511	0.6594	0.3693	0.3330	0.8561	0.0092	0.7094	0.6479	0.2718	0.3029	943.6773	0.4778
	PMGI	0.7566	0.4718	0.6736	0.3875	0.3597	0.8597	0.0140	0.7819	0.7388	0.4448	0.3740	967.0633	0.5222
	RFN-Nest	0.5928	0.2906	0.6562	0.2723	0.2691	0.8627	0.0079	0.6831	0.6091	0.1779	0.2648	981.0049	0.4833
	CSF	0.7525	0.4916	0.6837	0.3258	0.2507	0.8536	0.0220	0.7793	0.7570	0.4763	0.3727	772.7454	0.5250
	DRF	0.4226	0.2078	0.5590	0.1858	0.1137	0.8402	0.0222	0.5808	0.4117	0.0459	0.1138	1668.1819	0.4167
	FusionDN	0.7681	0.4825	0.6478	0.3665	0.2943	0.8524	0.0686	0.7797	0.7616	0.4975	0.3522	1223.1102	0.5510
	DDcGAN	0.5267	0.2668	0.5491	0.3499	0.3451	0.8548	0.0587	0.5329	0.4443	0.1147	0.1723	1004.4252	0.4566
	Proposed	0.8720	0.5903	0.7252	0.4681	0.4065	0.8820	0.0001	0.8315	0.7959	0.5609	0.5286	683.7624	0.5357

4.6. Computational Efficiency

To compare the computational efficiency, we ran all deep learning algorithms on the TNO dataset 10 times and took the average running time. It is worth noting that our experimental hardware environment was an Intel (R) Core (TM) i7-11700 with 64 GB RAM, but the experimental environments for various algorithms were different. The FusionGAN, GANMcC, PMGI, CSF, DRF, FusionDN, and DDcGAN methods used TensorFlow (CPU version). The RFN-Nest method used Pytorch (CPU version). Our algorithm was

implemented in Matlab. All parameters in the comparison algorithms were the default values given by their authors. Table 6 shows the average time of 10 operations, and the optimal value is shown in red font. The running time of our algorithm achieved fourth place, namely, 255.6642 s, behind PMGI, FusionGAN, and RFN-Nest methods. Although the running time of our algorithm obtained fourth place, our fusion effect is state of the art.

Table 6. The average running time of different methods for the TNO dataset (unit: seconds).

Method	FusionGAN	GANMcC	PMGI	RFN-Nest	CSF	DRF	FusionDN	DDcGAN	Proposed
Time	170.4436	338.3344	36.9569	193.7670	899.4110	350.3019	330.3895	304.0095	255.6642

5. Conclusions

In this paper, we propose a fusion method for IR and visible images based on PCANet and the image pyramid method. We use PCANet to obtain the activity-level measurement and weight assignment and apply an image pyramid to decompose and merge the images in multiple scales. The activity-level measurement obtained by PCANet has the stronger representation ability in focusing on IR target perception and visible detail description. We performed two ablation studies to verify the effectiveness of the image pyramid and the guided filter. Compared with nineteen representative methods, the experimental results demonstrated that the proposed method can achieve the state-of-the-art performance in both visual quality and objective evaluation metrics. However, we only used the results of the second stage of PCANet as image features, ignoring the useful information of the first stage. In the future research, we will explore combining features of multiple stages for fusion tasks.

Author Contributions: Conceptualization, S.L.; methodology, S.L.; software, S.L.; validation, S.L., G.W., Y.Z. and C.L.; formal analysis, S.L.; investigation, S.L.; data curation, S.L.; writing—original draft preparation, S.L.; writing—review and editing, S.L., G.W. and Y.Z.; project administration, G.W. and Y.Z.; funding acquisition, G.W. and Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the National Natural Science Foundation of China (62175054, 61865005, and 61762033), the Natural Science Foundation of Hainan Province (620RC554 and 617079), the Major Science and Technology Project of Haikou City (2021-002), the Open Project Program of Wuhan National Laboratory for Optoelectronics (2020WNLOKF001), the National Key Technology Support Program (2015BAH55F04 and 2015BAH55F01), the Major Science and Technology Project of Hainan Province (ZDKJ2016015), and the Scientific Research Staring Foundation of Hainan University (KYQD(ZR)1882).

Data Availability Statement: The data are not publicly available.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Qi, B.; Jin, L.; Li, G.; Zhang, Y.; Li, Q.; Bi, G.; Wang, W. Infrared and Visible Image Fusion Based on Co-Occurrence Analysis Shearlet Transform. *Remote Sens.* **2022**, *14*, 283. [CrossRef]
2. Gao, X.; Shi, Y.; Zhu, Q.; Fu, Q.; Wu, Y. Infrared and Visible Image Fusion with Deep Neural Network in Enhanced Flight Vision System. *Remote Sens.* **2022**, *14*, 2789. [CrossRef]
3. Burt, P.J.; Adelson, E.H. The Laplacian pyramid as a compact image code. In *Readings in Computer Vision*; Elsevier: Amsterdam, The Netherlands, 1987; pp. 671–679.
4. Naidu, V. Image fusion technique using multi-resolution singular value decomposition. *Defence Sci. J.* **2011**, *61*, 479. [CrossRef]
5. Li, H.; Manjunath, B.; Mitra, S.K. Multisensor image fusion using the wavelet transform. *Gr. Models Image Process.* **1995**, *57*, 235–245. [CrossRef]
6. Lewis, J.J.; O'Callaghan, R.J.; Nikolov, S.G.; Bull, D.R.; Canagarajah, N. Pixel-and region-based image fusion with complex wavelets. *Inf. Fusion* **2007**, *8*, 119–130. [CrossRef]

7. Nencini, F.; Garzelli, A.; Baronti, S.; Alparone, L. Remote sensing image fusion using the curvelet transform. *Inf. Fusion* **2007**, *8*, 143–156. [CrossRef]
8. Chen, J.; Li, X.; Luo, L.; Mei, X.; Ma, J. Infrared and visible image fusion based on target-enhanced multiscale transform decomposition. *Inf. Sci.* **2020**, *508*, 64–78. [CrossRef]
9. Gao, Z.; Zhang, C. Texture clear multi-modal image fusion with joint sparsity model. *Optik* **2017**, *130*, 255–265. [CrossRef]
10. Zhang, Q.; Fu, Y.; Li, H.; Zou, J. Dictionary learning method for joint sparse representation-based image fusion. *Opt. Eng.* **2013**, *52*, 057006. [CrossRef]
11. Liu, C.; Qi, Y.; Ding, W. Infrared and visible image fusion method based on saliency detection in sparse domain. *Infrared Phys. Technol.* **2017**, *83*, 94–102. [CrossRef]
12. Ma, J.; Zhou, Z.; Wang, B.; Zong, H. Infrared and visible image fusion based on visual saliency map and weighted least square optimization. *Infrared Phys. Technol.* **2017**, *82*, 8–17. [CrossRef]
13. Xu, H.; Zhang, H.; Ma, J. Classification saliency-based rule for visible and infrared image fusion. *IEEE Trans. Comput. Imaging* **2021**, *7*, 824–836. [CrossRef]
14. Liu, Y.; Chen, X.; Peng, H.; Wang, Z. Multi-focus image fusion with a deep convolutional neural network. *Inf. Fusion* **2017**, *36*, 191–207. [CrossRef]
15. Liu, Y.; Chen, X.; Cheng, J.; Peng, H.; Wang, Z. Infrared and visible image fusion with convolutional neural networks. *Int. J. Wavel. Multiresolut. Inf. Process.* **2018**, *16*, 1850018. [CrossRef]
16. Liu, Y.; Chen, X.; Cheng, J.; Peng, H. A medical image fusion method based on convolutional neural networks. In Proceedings of the 2017 20th International Conference on Information Fusion (Fusion), Xi'an, China, 10–13 July 2017; pp. 1–7.
17. Li, H.; Wu, X.J.; Kittler, J. Infrared and visible image fusion using a deep learning framework. In Proceedings of the 2018 24th international conference on pattern recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 2705–2710.
18. Li, H.; Wu, X.J.; Durrani, T.S. Infrared and visible image fusion with ResNet and zero-phase component analysis. *Infrared Phys. Technol.* **2019**, *102*, 103039. [CrossRef]
19. Ma, J.; Yu, W.; Liang, P.; Li, C.; Jiang, J. FusionGAN: A generative adversarial network for infrared and visible image fusion. *Inf. Fusion* **2019**, *48*, 11–26. [CrossRef]
20. Ma, J.; Xu, H.; Jiang, J.; Mei, X.; Zhang, X.P. DDcGAN: A dual-discriminator conditional generative adversarial network for multi-resolution image fusion. *IEEE Trans. Image Process.* **2020**, *29*, 4980–4995. [CrossRef] [PubMed]
21. Ma, J.; Zhang, H.; Shao, Z.; Liang, P.; Xu, H. GANMcC: A generative adversarial network with multiclassification constraints for infrared and visible image fusion. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 1–14. [CrossRef]
22. Chan, T.H.; Jia, K.; Gao, S.; Lu, J.; Zeng, Z.; Ma, Y. PCANet: A simple deep learning baseline for image classification? *IEEE Trans. Image Process.* **2015**, *24*, 5017–5032. [CrossRef]
23. Mertens, T.; Kautz, J.; Van Reeth, F. Exposure fusion. In Proceedings of the 15th Pacific Conference on Computer Graphics and Applications (PG'07), Seoul, Republic of Korea, 29 October–2 November 2007; pp. 382–390.
24. Piella, G. A general framework for multiresolution image fusion: From pixels to regions. *Inf. Fusion* **2003**, *4*, 259–280. [CrossRef]
25. Wang, S.; Chen, L.; Zhou, Z.; Sun, X.; Dong, J. Human fall detection in surveillance video based on PCANet. *Multimed. Tools Appl.* **2016**, *75*, 11603–11613. [CrossRef]
26. Gao, F.; Dong, J.; Li, B.; Xu, Q. Automatic change detection in synthetic aperture radar images based on PCANet. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 1792–1796. [CrossRef]
27. Song, X.; Wu, X.J. Multi-focus image fusion with PCA filters of PCANet. In Proceedings of the IAPR Workshop on Multimodal Pattern Recognition of Social Signals in Human–Computer Interaction, Beijing, China, 20 August 2018; pp. 1–17.
28. Yang, W.; Si, Y.; Wang, D.; Guo, B. Automatic recognition of arrhythmia based on principal component analysis network and linear support vector machine. *Comput. Biol. Med.* **2018**, *101*, 22–32. [CrossRef]
29. Zhang, G.; Si, Y.; Wang, D.; Yang, W.; Sun, Y. Automated detection of myocardial infarction using a gramian angular field and principal component analysis network. *IEEE Access* **2019**, *7*, 171570–171583. [CrossRef]
30. He, K.; Sun, J.; Tang, X. Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 1397–1409. [CrossRef] [PubMed]
31. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2014; pp. 740–755.
32. Li, S.; Kang, X.; Hu, J. Image fusion with guided filtering. *IEEE Trans. Image Process.* **2013**, *22*, 2864–2875.
33. Toet, A. TNO Image Fusion Dataset. 2014. Available online: https://figshare.com/articles/TN_Image_Fusion_Dataset/1008029 (accessed on 21 September 2022).
34. Xu, H.; Ma, J.; Le, Z.; Jiang, J.; Guo, X. Fusiondn: A unified densely connected network for image fusion. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12484–12491.
35. Yang, C.; Zhang, J.Q.; Wang, X.R.; Liu, X. A novel similarity based quality metric for image fusion. *Inf. Fusion* **2008**, *9*, 156–160. [CrossRef]
36. Xydeas, C.; Petrovic, V. Objective image fusion performance measure. *Electron. Lett.* **2000**, *36*, 308–309. [CrossRef]
37. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef]

38. Haghighat, M.; Razian, M.A. Fast-FMI: Non-reference image fusion metric. In Proceedings of the 2014 IEEE 8th International Conference on Application of Information and Communication Technologies (AICT), Astana, Kazakhstan, 15–17 October 2014; pp. 1–3.
39. Shreyamsha Kumar, B. Multifocus and multispectral image fusion based on pixel significance using discrete cosine harmonic wavelet transform. *Signal Image Video Process.* **2013**, *7*, 1125–1143. [CrossRef]
40. Piella, G.; Heijmans, H. A new quality metric for image fusion. In Proceedings of the 2003 International Conference on Image Processing (Cat. No. 03CH37429), Barcelona, Spain, 14–17 September 2003; Volume 3, p. 173.
41. Zhao, J.; Laganiere, R.; Liu, Z. Performance assessment of combinative pixel-level image fusion based on an absolute feature measurement. *Int. J. Innov. Comput. Inf. Control* **2007**, *3*, 1433–1447.
42. Chen, H.; Varshney, P.K. A human perception inspired quality metric for image fusion based on regional information. *Inf. Fusion* **2007**, *8*, 193–207. [CrossRef]
43. Chen, Y.; Blum, R.S. A new automated quality assessment algorithm for image fusion. *Image Vis. Comput.* **2009**, *27*, 1421–1432. [CrossRef]
44. Tan, W.; Zhou, H.; Song, J.; Li, H.; Yu, Y.; Du, J. Infrared and visible image perceptive fusion through multi-level Gaussian curvature filtering image decomposition. *Appl. Opt.* **2019**, *58*, 3064–3073. [CrossRef] [PubMed]
45. Zhang, H.; Xu, H.; Xiao, Y.; Guo, X.; Ma, J. Rethinking the image fusion: A fast unified image fusion network based on proportional maintenance of gradient and intensity. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12797–12804.
46. Li, H.; Wu, X.J.; Kittler, J. RFN-Nest: An end-to-end residual fusion network for infrared and visible images. *Inf. Fusion* **2021**, *73*, 72–86. [CrossRef]
47. Xu, H.; Wang, X.; Ma, J. DRF: Disentangled representation for visible and infrared image fusion. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–13. [CrossRef]
48. Ma, J.; Chen, C.; Li, C.; Huang, J. Infrared and visible image fusion via gradient transfer and total variation minimization. *Inf. Fusion* **2016**, *31*, 100–109. [CrossRef]
49. Du, Q.; Xu, H.; Ma, Y.; Huang, J.; Fan, F. Fusing infrared and visible images of different resolutions via total variation model. *Sensors* **2018**, *18*, 3827. [CrossRef]