

# 基于模糊算子的红外图像去模糊研究

张锦航<sup>1</sup>, 孙立辉<sup>1</sup>, 姜军强<sup>2</sup>

(1. 河北经贸大学信息技术学院, 河北 石家庄 050061; 2. 西安科技大学机械工程系, 陕西 西安 710054)

**摘要:** 针对镜头抖动, 目标移动等因素引起的图像运动模糊问题, 本文提出了一种基于模糊算子的红外图像去模糊算法, 使用深度自编码网络对数据集中的模糊算子进行编码, 通过编码后的模糊算子去逼近一个未知的模糊算子并搜索对应的清晰图像, 从而实现真实场景下红外图像去模糊, 弥补了现有基于深度学习的图像去模糊模型在跨域应用时对真实场景下运动模糊图像去模糊效果较差的不足。在红外图像上的实验结果表明, 相比于其他去模糊算法, 本文提出的去模糊算法取得了更高的性能指标, 恢复出的图像有着清晰的边缘轮廓和局部细节, 显著提升了红外图像的清晰度。

**关键词:** 红外图像; 深度学习; 模糊算子; 图像去模糊; 自编码

**中图分类号:** TP391.41 **文献标识码:** A **DOI:** 10.3969/j.issn.1001-5078.2023.01.020

## Research on infrared image deblurring based on blur operator

ZHANG Jin-hang<sup>1</sup>, SUN Li-hui<sup>1</sup>, JIANG Jun-qiang<sup>2</sup>

(1. Hebei University of Economics and Trade, College of Information Technology, Shijiazhuang 050061, China;

2. Department of Mechanical Engineering, Xi'an University of Science and Technology, Xi'an 710054, China)

**Abstract:** Aiming at the image motion blur caused by lens jitter, target movement and other factors, an infrared image deblurring algorithm based on fuzzy operator is proposed in this paper. The deep self-coding network is used to encode the fuzzy operator in the dataset, the encoded fuzzy operator is used to approximate unknown fuzzy operator and search for the corresponding clear image, thus the infrared image deblurring in the real scene is realized, which makes up for the deficiencies of the existing image deblurring model based on deep learning in the cross-domain application of the motion blurred image in the real scene. The experimental results on infrared blurred images show that compared with other deblurring algorithms, the proposed deblurring algorithm achieves higher performance, and the recovered images have clear edges and local details, significantly improving the clarity of infrared images.

**Keywords:** infrared image; deep learning; blur operator; image deblurring; self-coding

### 1 引言

随着红外成像技术的迅速发展, 红外成像技术开始应用于各个领域, 并占有重要的地位。红外成像通过不同对象的红外热辐射强度不同来形成图像, 无需使用外部可见光也可以成像, 相较于可见光成像, 红外成像可以实现夜视成像, 成像结果直观,

抗干扰能力强, 适用于复杂特殊环境的同时还具有可成像距离远、穿透能力强等诸多优势。红外成像设备发展到现在, 已经实现了便携式的设计, 方便携带, 操作简单。红外成像技术已经广泛应用于工业, 军事<sup>[1]</sup>等众多领域, 然而拍摄期间镜头的抖动或目标的快速移动会导致红外成像时产生运动模糊, 降

**基金项目:** 河北省重点研发计划项目 (No. 20350801D) 资助。

**作者简介:** 张锦航 (1998 -), 男, 硕士研究生, 研究方向为图像处理。E-mail: zjh7187666@163.com

**收稿日期:** 2022-02-22

低了红外图像的清晰度,对红外图像在这些领域的应用造成了严重影响。在相机曝光的时间段里镜头发生了移动,导致采集的像素点发生了变化,所以在得到的图像中场景和物体会看起来模糊或晃动,这被称为运动模糊。图像去模糊技术被广泛用于工业制造,医学影像等领域。红外图像去模糊的主要目的是去除模糊伪影以提高图像质量,在运动模糊的图像中提取有用的信息,对红外图像去模糊的研究具有重要价值。

传统的图像去模糊方法<sup>[2-4]</sup>通过使用约束条件,引入图像先验知识对图像模糊进行表征,但是现实中运动模糊情况多数是非均匀模糊,从而无法估计模糊核,因此这些传统图像去模糊方法在实际应用中的效果并不理想。近年来随着深度学习算法的发展,基于深度卷积神经网络(Convolutional Neural Network, CNN)的方法被大量用于图像去模糊领域。相比于传统的图像去模糊算法,深度学习算法可以做到更好的去模糊效果,过去的几年里已经提出了许多深度学习去模糊模型。Nah 等提出了一种端到端的多尺度卷积神经网络 DeepDeblur<sup>[5]</sup>用于图像去模糊,分别在三个尺度不同的层对图像进行去模糊,把低层输出的结果作为高层的输入,增加了收敛性。Tao 等提出了 SRN-Deblur<sup>[6]</sup>采用尺度循环结构进行图像去模糊。Kupyn 等提出了 DeblurGAN<sup>[7]</sup>首次将生成对抗网络(Generative Adversarial Network, GAN)用于图像去模糊,通过训练输入的模糊图像来生成高质量图像,随后他们又提出了 DeblurGANv2<sup>[8]</sup>,使用特征金字塔网络结构<sup>[9]</sup>提取图像特征并且结合了全局和局部两个尺度的判别器损失,运行速度快并且有一定的去模糊效果。还有许多关于多帧去模糊<sup>[10]</sup>和特定领域去模糊<sup>[11]</sup>的工作。然而基于深度学习的去模糊模型通常直接学习模糊图像和清晰图像之间的映射函数,在跨领域应用时泛化能力弱,鲁棒性差,并且将深度学习的方法用于红外图像去模糊的研究较为缺乏,大部分红外图像去模糊的研究使用的红外数据集也并非现实拍摄中的模糊核所生成的红外模糊图像,这就导致这些深度学习模型只能处理由简单模糊核产生的模拟红外模糊图像,无法处理真实场景中的模糊算子<sup>[12]</sup>,在面对真实场景下的红外模糊图像时没有效果。

针对上述问题本文提出了一种基于模糊算子的

红外图像去模糊方法,采用深度自编码网络对由清晰图像和模糊图像组成的数据集中的模糊算子进行编码,通过编码后的模糊算子去逼近一个真实未知的模糊算子并搜索对应的清晰图像,这样可以处理真实场景下红外图像的运动模糊,实现真实场景下红外图像去模糊,同时避免了使用复杂的模糊核手工先验。此外本文还设计了合成模糊的结构,实现了将现实中通过镜头运动等方式产生的模糊转移到另一个图像来生成真实模糊图像的功能,将现有数据集的运动模糊转移到红外图像上,制作红外图像的数据集对,解决了红外图像难以收集配对数据的问题,在完善模糊图像数据集并将技术推广至红外图像领域有着重要的意义。实验结果表明,该方法能有效提高红外图像的质量,去除运动模糊后可以得到较为清晰的红外图像。

## 2 基于模糊算子的红外图像去模糊算法

模糊算子是一种模糊关系合成运算,在模糊图像生成的过程中,模糊算子对模糊核和清晰图像进行合成运算从而生成模糊图像。由清晰图像  $x$  和模糊核  $k$  生成模糊图像  $y$  的模型通常被公式化表示为:

$$y = f(x, k) + \eta \approx f(x, k) \quad (1)$$

其中,  $f(x, k)$  是带有模糊核  $k$  的模糊算子;  $\eta$  是噪声。

若干模糊算子组成了模糊算子集合<sup>[12]</sup>,通过模糊核能够调用模糊算子集合中的模糊算子。本文设计了一种模糊核提取器用来提取清晰图像和对应模糊图像之间的模糊核,同时本文设定了包含模糊算子集合和模糊核提取器的显式表示的公式如下:

$$y = F(x, k) \quad (2)$$

$$k = G(x, y) \quad (3)$$

其中,  $F$  为模糊算子集合;  $G$  为模糊核提取器。模糊算子集合  $F$  中的每个特定的模糊算子  $f(x, k)$  都对应着一个特定的模糊核  $k$ ,当  $F$  的函数形式固定时,可以通过  $k$  引用  $F$  中的  $f(x, k)$ 。图像去模糊的目标是在给定模糊图像  $y$  的情况下找到清晰的图像  $x$ ,当训练好模糊算子集合  $F$  之后,通过模糊核  $k$  引用  $F$  中的模糊算子  $f(x, k)$ ,使  $f(x, k)$  不断逼近  $y$  上的真实模糊算子,从而找到满足上述公式的  $x$  和  $k$  来对模糊图像  $y$  进行去模糊,去模糊算法流程如表 1 所示。

表 1 去模糊算法流程

Tab. 1 Algorithm flow of deblurring

算法: 基于模糊算子的红外图像去模糊算法
输入: 模糊图像 $y$
输出: 清晰图像 $x$
1. 设定标准正态随机向量 $Z_x, Z_k$
2. $Z_x$ 输入 $F, x \leftarrow F(Z_x)$ $Z_k$ 输入 $G, k \leftarrow G(Z_k)$
3. while $H(y, F(x, k))$ 未收敛: 随机初始化 $Z_x, x \leftarrow F(Z_x)$ $x_1 \leftarrow F(Z_x)$ while $H(y, F(x_1, k))$ 未收敛: 随机初始化 $Z_k, k \leftarrow G(Z_k)$ end while $k_1 \leftarrow G(Z_k)$ end while $H_{\min} \leftarrow H(y, F(x_1, k_1))$
4. $x \leftarrow F(Z_x)$

给定一个由  $n$  个清晰图像和模糊图像组成的训练集  $\{(x_i, y_i)\}_{i=1}^n$ , 通过这个数据集来训练模糊算子集合  $F$ ,  $F$  能够对清晰图像  $x_i$  和对应模糊图像  $y_i$  之间的模糊进行建模。每对图像都与一个潜在的模糊核  $k$  相关联, 以模糊核  $k_i$  和清晰图像  $x_i$  为模糊算子集合的参数得到模糊图像  $y_i$ , 即  $y_i = F(x_i, k_i)$ 。每个图像对都有一个特定的  $k_i$ , 因此可以假设  $k_i$  可以通过模糊核提取器  $G$  生成, 即  $k_i = G(x_i, y_i)$ 。通过合成模糊图像  $F(x_i, G(x_i, y_i))$  和真实模糊图像  $y_i$  之间的差异来训练模糊算子集合  $F$  和模糊核提取器  $G$ 。

### 3 基于模糊算子的红外图像去模糊算法实现

#### 3.1 总体结构

模糊算子集合  $F$  可用于在给定的模糊核  $k$  的情况下生成模糊图像, 模糊核提取器  $G$  可用于提取模糊核  $k$ , 本文通过两个神经网络来实现  $F$  和  $G$ , 使用一个带有跳跃连接<sup>[13]</sup> 的深度自编码网络实现  $F$ , 使用一个残差网络<sup>[14]</sup> 实现  $G$ 。首先从数据集中采样清晰图像和对应的模糊图像, 将这些图像拼接起来输入给模糊核提取器  $G$ , 生成相应的编码模糊核向量  $k$ 。然后将清晰图像  $x$  输入模糊算子集合  $F$ , 在编码器部分清晰图像  $x$  被编码为瓶颈结构嵌入向量, 在瓶颈结构部分将嵌入向量  $x$  与模糊核向量  $k$  拼接

成一个新的向量。最后在解码器部分对新的向量进行解码以获得合成的模糊图像, 通过使合成模糊图像  $F(x, k)$  和真实模糊图像  $y$  之间差距最小化的损失函数  $H(y, (x, k))$  来训练模糊算子集合  $F$ 。合成模糊模型结构如图 1 所示。

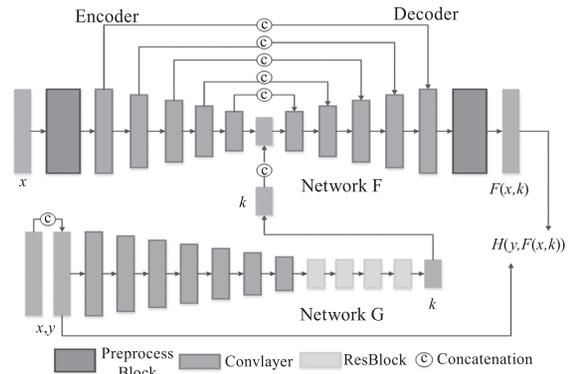


图 1 合成模糊模型结构

Fig. 1 Synthetic fuzzy model structure

模糊算子集合  $F$  固定之后, 可将其用于图像去模糊。给定模糊图像  $y$ , 在去模糊的过程中, 设定两个标准正态随机向量  $z_1$  和  $z_2$  分别作为模糊核提取器  $G$  和模糊算子集合  $F$  的输入, 模糊核提取器  $G$  输出随机模糊核向量  $k$ , 模糊算子集合  $F$  输出随机清晰图像向量  $x$ , 因为  $F$  的形式已经固定, 所以只需要找到清晰图像向量  $x$  和模糊核向量  $k$  使损失函数  $H$  最小化, 从而找到清晰图像  $x$ 。去模糊过程中使用迭代优化的方法来优化  $H$ , 交替进行以下两个步骤: (1) 固定模糊核  $k$  并优化潜在清晰图像  $x$ 。(2) 固定潜在清晰图像  $x$  并优化模糊核  $k$ 。去模糊的模型结构如图 2 所示。

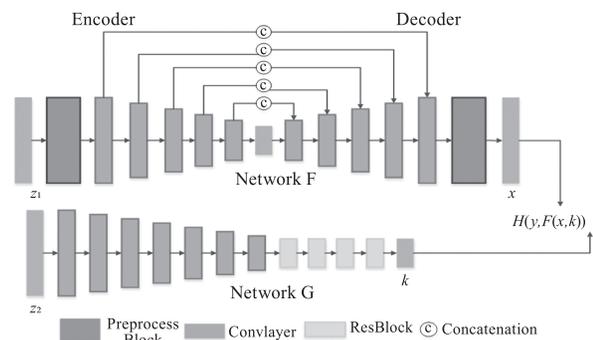


图 2 去模糊模型结构

Fig. 2 Deblurring model structure

#### 3.2 模糊算子集合设计

模糊算子集合  $F$  由预处理模块和深度自编码网络构成, 预处理模块由卷积层和十个残差块组成, 在网络头部通过预处理模块先对输入图像进行简单

特征提取,将输入图像转换为特征图,方便后续编码器部分进行更深层的特征提取。在网络尾部使用变换卷积层和残差块位置的预处理模块将特征图转换为输出图像。先使用预处理模块对输入图像进行两次下采样,输出将输入图像转换为 64 通道的特征图到编码器(Encoder)和解码器(Decoder)部分。编码器和解码器部分各包含五个卷积层,采用多个  $3 \times 3$  卷积,每个卷积后都添加 LReLU 激活函数,有助于增大网络稀疏性,增强了网络的泛化能力。在相同维度的卷积层之间增添跳跃连接,增强了特征信息在网络中的传递,确保了特征信息的可重用性,并且避免了梯度消失的问题。编码器对特征图进行多次下采样,每次下采样将通道数变为前一层通道数的一半,输出为  $512 \times 2 \times 2$  大小的向量,输出向量与模糊核向量拼接之后作为解码器的输入,解码器对拼接向量进行多次上采样重建输出特征图。最后在网络尾部使用预处理模块对解码器输出的特征图上采样,将 64 通道的特征图转换回输出图像。详细网络层次结构如表 2 所示。

表 2 自编码网络模型参数

Tab. 2 Model parameters for the autoencoder network

编码部分		
层数	特征图尺寸	步长
Conv	$64 \times 256 \times 256$	1
Conv	$64 \times 128 \times 128$	2
Conv	$64 \times 64 \times 64$	2
ResBlock10	$64 \times 64 \times 64$	1
Conv	$64 \times 32 \times 32$	2
Conv	$128 \times 16 \times 16$	2
Conv	$256 \times 8 \times 8$	2
Conv	$512 \times 4 \times 4$	2
Conv	$512 \times 2 \times 2$	2
解码部分		
层数	特征图尺寸	步长
Conv	$512 \times 4 \times 4$	2
Conv	$256 \times 8 \times 8$	2
Conv	$128 \times 16 \times 16$	2
Conv	$64 \times 32 \times 32$	2
Conv	$64 \times 64 \times 64$	2
ResBlock10	$64 \times 64 \times 64$	1
Conv	$64 \times 64 \times 64$	1
Conv	$256 \times 128 \times 128$	1
Conv	$64 \times 256 \times 256$	1

### 3.3 模糊核提取器设计

模糊核提取器  $G$  由卷积层和残差模块构成,残差模块包含了两个卷积层和一个 LRelu 激活函数,通过添加残差模块增加了网络深度,避免产生梯度消失和梯度爆炸的问题。该网络的输入是清晰图像和模糊图像拼接而成的向量,输出一个大小为  $512 \times 2 \times 2$  的模糊核向量,详细网络层次结构如表 3 所示。

表 3 提取器模型参数

Tab. 3 Model parameters for the extractor

层数	特征图尺寸	步长
Conv	$64 \times 512 \times 512$	1
Conv	$64 \times 256 \times 256$	2
Conv	$64 \times 128 \times 128$	2
Conv	$64 \times 128 \times 128$	1
Conv	$128 \times 64 \times 64$	2
Conv	$256 \times 32 \times 32$	2
Conv	$512 \times 16 \times 16$	2
Conv	$512 \times 8 \times 8$	2
Conv	$512 \times 4 \times 4$	2
ResBlock4	$512 \times 2 \times 2$	1

### 3.4 损失函数

损失函数用来指导模型训练,对训练效果有着不可忽视的影响。本文采用的损失函数  $h$  是 Charbonnier 损失<sup>[15]</sup>, Charbonnier 损失是一种较为稳定的改良版 L1 范数损失,收敛速度快,处理异常的能力强,性能提升大,目前在超分领域应用较多。Charbonnier 损失用于计算合成模糊图像  $F(x_i, G(x_i, y_i))$  和对应真实模糊图像  $y_i$  之间的差距。表示为:

$$\sum_{i=1}^n H(y_i, F(x_i, G(x_i, y_i))) \quad (4)$$

为了使优化过程更加稳定,得到更好的去模糊效果,在损失函数中添加两个正则化项,在模糊核  $k$  上添加一个正则化项使优化过程更稳定,在清晰图像  $x$  的图像梯度上使用超拉普拉斯先验<sup>[16]</sup>,加大梯度的稀疏性,减少噪声并获得更自然的图像  $x$ 。添加正则化之后的损失函数如下:

$$H(y, F(x, k)) + \mu \|k\|_2 + \beta (g_u^2(x) + g_v^2(x))^{\frac{\alpha}{2}} \quad (5)$$

其中,  $g_u$  和  $g_v$  分别是水平方向和垂直方向上的导数;  $\mu, \beta, \alpha$  是可调的超参数。

#### 4 实验结果及分析

##### 4.1 实验环境

本文设计了模糊合成与红外图像去模糊两部分实验来验证本文提出基于模糊算子的方法是否有效, 红外图像去模糊实验采用自制红外图像数据集, 模糊合成实验采用公开的 GOPRO 数据集<sup>[7]</sup>。在配有 RTX2080SGPU, 32GB 内存, Intel Xeon e5-2640CPU, 操作系统为 windows10 的联想 P900 工作站上进行实验, 使用基于 Pytorch 的深度学习框架进行训练和测试, 所有实验均在相同的环境下进行。

##### 4.2 实验数据集和参数设置

自制红外图像数据集是采用 Xcore Micro II 系列非制冷红外机芯组件在不同场景下拍摄而成的。其中训练集是由拍摄的清晰图像和使用本文模糊合成方法生成的模糊图像组成的, 包含了 5568 张清晰图像和模糊图像, 每张图像分辨率为  $640 \times 512$ 。测试集是通过镜头移动或晃动等方式拍摄出的 241 张真实红外模糊图像, 用于测试去模糊效果。

GOPRO 数据集中的图像是采用 GOPRO4 相机拍摄的每秒 240 帧的高速视频中的视频帧, 用来模拟运动模糊图像。该数据集包含 3214 对清晰图像和模糊图像, 其中训练集为 2103 对, 测试集为 1111 对, 每张图像分辨率为  $1280 \times 720$ 。

两次实验分别使用自制红外图像数据集和 GOPRO 数据集去训练模糊算子集合和模糊核提取器组成的模型, 在使用训练集前要将每对模糊图像和清晰图像在同样的位置裁剪出分辨率为  $256 \times 256$  的图像, 两次训练均使用相同的参数, 实验采用 Adam 优化器<sup>[17]</sup>, 参数设置为  $b_1 = 0.9, b_2 = 0.99$ , 设定 batchsize 为 4, 将模糊核上正则化的权重设为  $1 \times 10^{-4}$ , 将超拉普拉斯先验的权重设为  $1 \times 10^{-2}$ , 设定初始学习率为  $1 \times 10^{-4}$ , 并采用余弦退火算法<sup>[18]</sup>逐步衰减学习率。实验总共迭代  $6 \times 10^6$  次。实验结果选用峰值信噪比 (Peak Signal-to-Noise Ratio, PSNR) 和结构相似度 (Structural Similarity, SSIM)<sup>[19]</sup> 两项指标进行衡量。峰值信噪比数值高表示图像质量较好, 引入噪声较少, 失真较小。结构相似度的数

值越高表示生成的清晰图像越接近真实清晰图像, 两幅图像的结构越相似。

##### 4.3 实验结果

##### 4.3.1 模糊合成

模糊合成实验是为了验证使用本文提出的方法生成的模糊图像, 能否替代真实的模糊图像。在原数据集中选取清晰图像  $x$  和对应模糊图像  $y$ , 再随机选取清晰图像  $x_1$ , 使用从  $(x, y)$  中提取的模糊核在  $x_1$  上生成模糊图像  $y_1$ , 然后使用由  $(x_1, y_1)$  组成的合成数据集去训练深度学习去模糊模型, 最终比较使用原数据集训练的模型和使用合成数据集训练的模型之间的性能。

本文在 GOPRO 数据集上进行模糊合成实验。通过使用 GOPRO 数据集训练出的模型生成模糊图像, 原清晰图像和新生成的模糊图像组成了合成数据集之后, 使用原数据集和合成数据集分别训练去模糊模型 DeblurGANv2<sup>[8]</sup>, 评价指标值和实验效果图如表 4 和图 3 所示。从实验结果可以看到使用原数据集和合成数据集训练出的模型在 PSNR 和 SSIM 性能指标上几乎没有差异, 使用本文提出的方法生成的模糊图像可以替代真实模糊图像。

表 4 训练结果

Tab. 4 Training results

数据集	PSNR/dB	SSIM
GOPRO	28.03	0.9224
合成数据集	27.43	0.9193



图 3 合成模糊图像效果

Fig. 3 The effect of synthetic blurred images

##### 4.3.2 红外图像去模糊

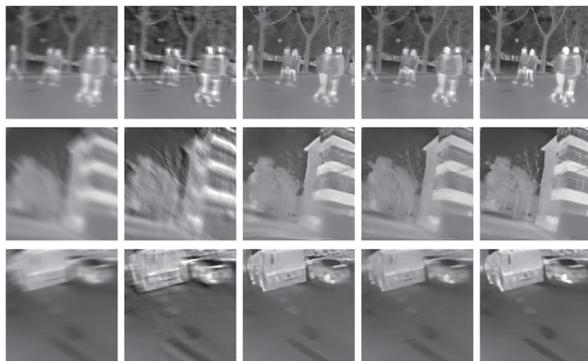
本文在自制红外图像数据集上进行去模糊对比

实验。采用 DeepDeblur<sup>[5]</sup>, SRN-Deblur<sup>[6]</sup>, DeblurGANv2<sup>[8]</sup> 三种常见的去模糊算法与本文算法进行对比,先使用自制红外图像数据集去训练模型,最后在测试集上测试去模糊效果,所有算法均采用相同的训练样本和测试样本,评价指标值和实验效果图如表 5 和图 4 所示。DeepDeblur 处理之后的模糊图像产生了畸变,存在严重失真的现象。SRN-Deblur 处理之后的模糊图像虽然清晰度有了较大提升,但是局部细节不够清晰。DeblurGANv2 处理之后的模糊图像产生了噪声,图像质量较为粗糙。本文算法处理之后的图像得到了清晰明了的边缘轮廓,丰富的局部细节信息,图像质量显著提高,在视觉上更接近清晰图像,有效提升了红外模糊图像的清晰度并且取得了高于其他方法的 PSNR 和 SSIM 性能指标。

表 5 红外图像去模糊评价指标值

Tab. 5 Evaluation index value of infrared image deblurring

算法	PSNR/dB	SSIM
DeepDeblur	17.52	0.6594
SRN-Deblur	28.18	0.8771
DeblurGANv2	23.06	0.8213
本文算法	30.98	0.9105



(a)清晰图像 (b)DeepDeblur (c)SRN-Deblur (d)DeblurGANv2 (e)本文算法

图 4 红外图像去模糊效果

Fig. 4 Deblurring effect of infrared image

## 5 总结

由于深度学习去模糊模型在跨领域应用时表现不佳,对真实场景下红外模糊图像没有效果,本文提出了一种基于模糊算子的红外图像去模糊方法。通过深度自编码网络和残差网络实现了模糊算子集合和模糊核提取器,该方法可以处理真实场景下红外图像的运动模糊,恢复出边缘结构更清晰,局部细节

更丰富的清晰红外图像。该模型还可以对清晰图像进行模糊合成,解决了红外图像难以收集配对数据的问题。实验结果表明,本文提出的方法取得了有说服力的图像去模糊效果,显著提升了模糊图像的清晰度。

## 参考文献:

- [1] Sun Wei, Wang Biao, Huang Jinke, et al. Research on jamming strategy of surface-type infrared decoy against infrared guided missile[J]. Laser & Infrared, 2019, 49(6): 706-713. (in Chinese)  
孙卫,王彪,黄金科,等. 面源红外诱饵对抗红外制导弹干扰策略研究[J]. 激光与红外, 2019, 49(6): 706-713.
- [2] Qi S, Jia J, Agarwala A. High-quality motion deblurring from a single image[J]. Acm Transactions on Graphics, 2008, 27(3): 1-10.
- [3] Whyte O, Sivic J, Zisserman A, et al. Non-uniform deblurring for shaken images[C]//Computer Vision & Pattern Recognition. IEEE, 2010.
- [4] Cho T S, Paris S, Horn B K P, et al. Blur kernel estimation using the radon transform[C]//Computer Vision & Pattern Recognition. IEEE, 2011.
- [5] Nah S, Kim T H, Lee K M. Deep multi-scale convolutional neural network for dynamic scene deblurring[C]//IEEE Computer Society, 2016.
- [6] Tao X, Gao H, Wang Y, et al. Scale-recurrent network for deep image deblurring[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2018.
- [7] Kupyn O, Budzan V, Mykhailych M, et al. DeblurGAN: blind motion deblurring using conditional adversarial networks[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR). IEEE, 2018.
- [8] Kupyn O, Martyniuk T, Wu J, et al. DeblurGAN-v2: deblurring (orders-of-magnitude) faster and better[C]//2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2019.
- [9] Lin T Y, Dollar P, Girshick R, et al. Feature pyramid networks for object detection[OL/J]. <http://arxiv.org/pdf/1612.03144.pdf>.
- [10] Su S, Delbracio M, Wang J, et al. Deep video deblurring for hand-held cameras[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR). IEEE, 2017.
- [11] M Hradiš. Convolutional neural networks for direct text de-

- blurring[C]//British Machine Vision Conference,2015.
- [12] Chen Wufan, Lu Xianqing, Chen Jianjun, et al. A new algorithm for color image edge detection – generalized fuzzy operator method [J]. *Scientia Sinica (Mathematica)*, 1995, (2):219 – 225. (in Chinese)
- 陈武凡, 鲁贤庆, 陈建军, 等. 彩色图象边缘检测的新算法——广义模糊算子法[J]. *中国科学(A辑数学物理学天文学技术科学)*, 1995, (2):219 – 225.
- [13] Ronneberger O, Fischer P, Brox T. U-Net: convolutional networks for biomedical image segmentation[M]. Springer International Publishing, 2015.
- [14] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[J]. *arXiv arXiv:1512.03385*, 2015.
- [15] Lai W S, Huang J B, Ahuja N, et al. Deep Laplacian pyramid networks for fast and accurate super-resolution[C]//IEEE Conference on Computer Vision & Pattern Recognition. IEEE Computer Society, 2017:5835 – 5843.
- [16] Krishnan D, Fergus R. Fast image deconvolution using Hyper-Laplacian Priors [C]//Advances in Neural Information Processing Systems 22:23rd Annual Conference on Neural Information Processing Systems, 2009.
- [17] Kingma D, Ba J. Adam: a method for stochastic optimization[J]. *Computer Science*, 2014.
- [18] Loshchilov I, Hutter F. SGDR: stochastic gradient descent with warm restarts[J]. *arXiv e-prints*, 2016.
- [19] Shivsubramani K, Soman K P. Implementation and comparative study of image fusion algorithms[J]. *International Journal of Computer Applications*, 2010, 9(2):10 – 20.