

多尺度交叉模块的图像去模糊算法研究

谢壹珍¹

XIE Yizhen

摘要

近年来, 图像去模糊技术作为计算机视觉领域的一个重要研究方向, 吸引了大量研究者的关注。针对传统图像去模糊方法在处理复杂运动模糊时存在的局限性, 提出一种基于多尺度交叉模块的图像去模糊算法。通过融合多尺度信息, 并结合注意力机制, 实现了对模糊图像的高效复原。首先构建了一个多尺度特征提取网络, 用于捕获图像在不同尺度下的模糊特征; 随后, 设计了一个基于注意力机制的多尺度交叉模块, 通过空间注意力推断, 将较小尺度下的细节信息有效地补充到较大尺度的模糊特征中, 从而实现特征的精细化重建; 最后, 利用融合后的多尺度特征重建出清晰图像。实验结果表明, 所提出的多尺度交叉模块图像去模糊算法在多个公开数据集上均取得了显著优于传统方法的去模糊效果, 验证了算法的有效性和优越性。

关键词

多尺度特征提取; 交叉模块; 注意力机制; 图像去模糊; 深度学习

doi: 10.3969/j.issn.1672-9528.2024.10.004

0 引言

在数字图像处理领域, 图像去模糊作为恢复图像清晰度的关键技术, 其重要性不言而喻。随着智能手机、监控摄像头等设备的普及, 模糊图像问题愈发常见, 严重影响了图像信息的有效提取与应用, 如安全监控、医学影像分析、卫星遥感等领域均面临这一挑战。然而, 传统去模糊方法往往受限于计算复杂度高、对噪声敏感以及难以处理复杂模糊场景等局限性, 难以满足实际应用需求。近年来, 深度学习技术的兴起为图像去模糊领域带来了革命性的变化, 尤其是卷积神经网络(CNN)的广泛应用, 显著提升了去模糊效果^[1-2]。同时, 多尺度分析与交叉模块作为图像处理中的新兴技术, 通过在不同尺度上提取和融合特征, 展现出强大的信息表征能力, 为图像去模糊提供了新的思路。国内外学者已在这一领域取得了一系列研究成果, 不仅推动了图像去模糊技术的历史发展, 还探索了多尺度分析与交叉模块在其他图像处理任务中的广泛应用案例^[3]。本文旨在深入研究多尺度交叉模块的图像去模糊算法, 设计一种高效、鲁棒的去模糊算法, 以应对复杂多变的模糊场景。

1 图像去模糊基础理论

模糊图像的形成通常可以视为原始清晰图像经过某种退化过程(即模糊核或点扩散函数)和可能的噪声干扰后得到

的结果^[4]。这一过程可以通过公式(1)来描述。

$$g(x, y) = f(x, y) \otimes h(x, y) + n(x, y) \quad (1)$$

式中: $g(x, y)$ 是模糊图像, $f(x, y)$ 是原始清晰图像, $h(x, y)$ 是模糊核, $\otimes$ 表示二维卷积操作, $n(x, y)$ 是加性噪声。此模型描述了模糊图像如何由清晰图像经过模糊核与噪声的干扰而形成。去模糊问题的目标是从观测到的模糊图像 $g(x, y)$ 中恢复出原始的清晰图像 $f(x, y)$ 。由于模糊核 $h(x, y)$ 和噪声 $n(x, y)$ 的存在, 这一过程往往是一个病态的逆问题, 即存在多个可能的清晰图像能够产生相同的模糊图像。因此, 去模糊算法通常需要引入一些先验知识或正则化项来约束解空间, 使得恢复出的图像既符合观测数据, 又满足某些先验假设(如图像平滑性、边缘锐利性等)。

数学上, 去模糊问题可以表述为一个优化问题, 目标函数通常包含两部分: 数据保真项和正则化项。数据保真项确保恢复出的图像与模糊图像在某种度量下尽可能接近, 而正则化项则用于引入先验知识, 约束解空间。例如, 一个典型的去模糊优化问题可以表示为:

$$[\min_f \{ \frac{1}{2} \sum_{(x,y) \in \Omega} [g(x, y) - (f * h)(x, y)]^2 + \lambda \Phi(f) \}] \quad (2)$$

$\min_f$ 表示对函数 f 进行最小化, 即寻找使目标函数达到最小值的 f 。 $\frac{1}{2} \sum_{(x,y) \in \Omega} [g(x, y) - (f * h)(x, y)]^2$ 是数据保真项, 其中 $(x, y) \in \Omega$ 表示对图像域 Ω 内的所有像素点 (x, y) 进行求和。 $g(x, y)$ 是观测到的模糊图像在像素点 (x, y) 上的值。

$(f * h)(x, y)$ 表示原始清晰图像 f 与模糊核 h 进行二维卷

1. 闽南理工学院信息工程学院 福建石狮 362700

[基金项目] 闽南理工学院校级科研基金项目资助 (23KJX020)

积操作后在像素点 (x, y) 上的结果。 $\lambda\Phi(f)$ 是正则化项, λ 是正则化参数, 用于平衡数据保真项和正则化项之间的权重。 $\Phi(f)$ 是一个关于 f 的正则化函数, 用于引入先验知识或约束。这个表达式明确地定义了图像去模糊问题的数学框架, 其中目标是通过最小化目标函数来恢复出原始的清晰图像 f 。

2 交叉模块去模糊处理

在图像处理领域, 去模糊处理是一个重要的任务, 特别是在处理复杂场景或多模态图像时。交叉模块的设计旨在通过融合不同尺度的特征信息, 以及利用注意力机制增强关键信息的处理, 来实现更高效的去模糊效果^[5-7]。

交叉模块的设计基于多尺度特征提取与融合的思想, 结合注意力机制, 具体结构如下。

(1) 多尺度特征提取

使用不同尺度的卷积层或 Transformer 编码器从输入的模糊图像中提取多层次的特征图。每个尺度的特征图代表了图像在不同分辨率下的信息。通过池化层或步长大于 1 的卷积层来降低特征图的分辨率, 从而获取不同尺度的特征^[8-9]。

(2) 交叉注意力融合层

在多尺度特征提取之后, 引入交叉注意力机制来融合不同尺度的特征图。交叉注意力层允许每个尺度的特征图与其他尺度的特征图进行交互, 从而捕获更全面的上下文信息。

交叉注意力层采用类似于 Transformer 中的多头注意力结构, 其中查询 (query)、键 (key) 和值 (value) 可以来自不同尺度的特征图。

(3) 特征图更新机制

在交叉注意力融合之后, 更新每个尺度的特征图。通过加权求和或特征融合操作实现, 其中权重通过学习得到, 以反映不同尺度特征的重要性。

引入残差连接, 将原始特征图与融合后的特征图相加, 以保持信息的完整性, 并加速训练过程。

3 网络模型设计

本网络模型旨在处理图像去模糊任务, 通过多尺度特征提取、交叉模块去模糊处理以及去模糊结果重建等步骤, 从模糊的输入图像中恢复出清晰的图像。

(1) 输入层

输入: 接收一张模糊的图像作为输入。

(2) 多尺度特征提取模块

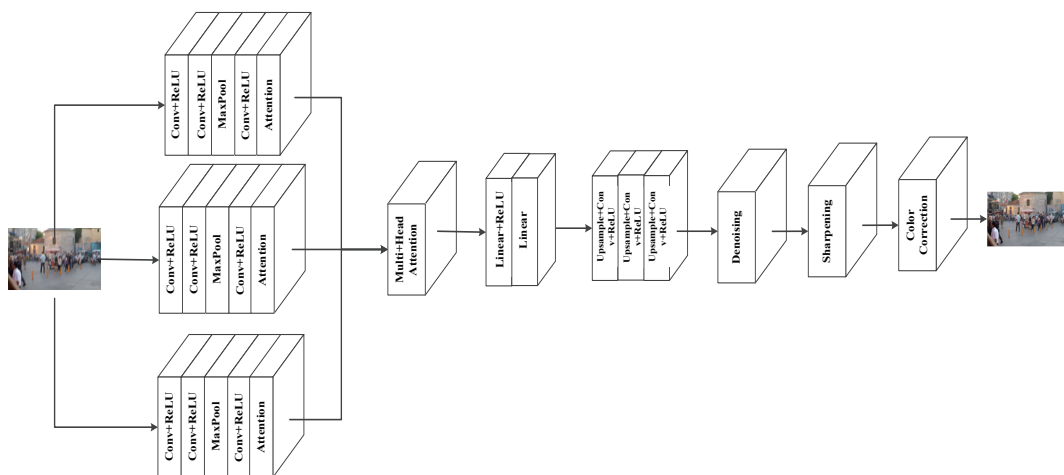


图 1 网络模型图

该模块旨在从不同尺度上提取输入图像的特征。通过多个并行的分支结构来实现, 每个分支对应于一个不同的尺度。在每个分支中, 首先通过初始的卷积层将输入图像转换为特征图。然后, 通过堆叠的卷积层进一步提取特征, 每个尺度的分支根据其对应的尺度调整卷积核的大小和步长, 以及是否包含池化层, 可以在某些分支的末尾引入注意力机制层, 以增强特征表达能力。

(3) 交叉模块去模糊处理

该模块负责将多尺度特征提取模块输出的特征图进行融合和处理, 以去除模糊。首先, 对每个尺度的特征图应用多头注意力机制, 通过多头注意力层让每个头关注特征图的不同部分, 然后将不同头的输出进行拼接。接着, 通过前馈网络对拼接后的特征图进行进一步处理。最后, 将处理后的特征图与原始特征图进行融合, 采用加权求和或特征拼接的方式进行融合。

(4) 去模糊结果重建模块

该模块负责将交叉模块去模糊处理后的特征图重建为清晰的图像。从最小尺度的特征图开始, 通过上采样操作增加特征图的分辨率, 并在每个上采样步骤后引入额外的卷积层以细化图像细节。这个过程重复进行, 直到特征图的分辨率与输入图像的分辨率相同。在重建过程的最后, 包括可选的后处理步骤, 如去噪、锐化和色彩校正, 以进一步提升图像的清晰度和视觉效果。

(5) 损失函数

损失函数用于评估网络模型输出的清晰图像与真实清晰图像之间的差异, 并指导模型的训练过程。通常采用 L1 损失和 L2 损失的组合作为总损失函数。L1 损失有助于保持图像的边缘信息, 而 L2 损失则有助于平滑图像。通过调整 L1 损失和 L2 损失的权重, 可以在训练过程中平衡这两个目标。

综上所述, 该网络模型通过多尺度特征提取、交叉模块去模糊处理和去模糊结果重建等步骤, 实现了从模糊图像到清晰图像的转换。网络模型图如图 1 所示。

4 多尺度交叉模块的图像去模糊算法设计

4.1 总体算法流程

(1) 输入处理

输入：接收待处理的模糊图像 I_{blur} 。

输出：将 I_{blur} 送入多尺度特征提取模块。

(2) 多尺度特征提取模块

假设有 N 个尺度，每个尺度 i 的特征提取可以表示为 $F_i = F_i(I_{\text{blur}})$ ，其中 $F_i(I_{\text{blur}})$ 是第 i 个尺度的特征提取函数。

(3) 交叉模块去模糊处理

交叉模块通过注意力机制融合多尺度特征，可以表示为 $\hat{F} = A(F_1, F_2, \dots, F_N)$ ，其中 A 是注意力融合函数。

(4) 去模糊结果重建模块

重建过程包含多个上采样和卷积操作，可以表示为 $I_{\text{sharp}} = R(\hat{F})$ ，其中 R 是重建函数。

(5) 后处理与输出

后处理可能包括去噪、锐化等，可以表示为 $I_{\text{final}} = P(I_{\text{sharp}})$ ，其中 P 是后处理函数。

4.2 多尺度特征提取

对于每个尺度 i ，特征提取表示为一系列卷积、池化等操作的组合，其复合函数为：

$$F_i = F_i(I_{\text{blur}}) = A_i^{(m)} \circ \text{Pool}_i^{(m-1)} \circ \sigma_i^{(m-1)} \circ C_i^{(m-1)} \circ \dots \circ \text{Pool}_i^{(1)} \circ \sigma_i^{(1)} \circ C_i^{(1)}(I_{\text{blur}}) \quad (3)$$

式中： F_i 表示第 i 个尺度下的特征图； $F_i(I_{\text{blur}})$ 是第 i 个尺度的特征提取函数； I_{blur} 是输入的模糊图像； $C_i^{(j)}$ 表示第 i 个尺度下第 j 个卷积层； $\sigma_i^{(j)}$ 表示第 i 个尺度下第 j 个卷积层之后的激活函数（如 ReLU）； $\text{Pool}_i^{(j)}$ 表示第 i 个尺度下第 j 个池化层（如果存在的话）； $A_i^{(m)}$ 表示第 i 个尺度下可能存在的注意力机制，它可能作用于最后一个卷积层或池化层的输出上； m 表示在第 i 个尺度下卷积层的总数； $\circ$ 表示函数复合。

该模块采用并行结构，每个分支对应一个尺度。每个分支由多个卷积层（或 Transformer 编码器层）组成，通过调整卷积核大小、步长或池化操作来实现不同尺度的特征提取。例如，第一个分支使用较小的卷积核和步长，保持较高的分辨率；而后续分支则通过逐步增大卷积核和步长或使用池化层来降低分辨率，从而提取更大尺度的特征。

4.3 交叉模块去模糊处理

交叉模块的设计巧妙地融合了多头注意力机制和特征融合技术，旨在显著提升深度学习模型的表示能力和性能。多头注意力层是这一机制的核心，它通过将输入特征分割成多个部分，每个部分由一个独立的“头”来处理。每个头能够专注于输入特征的不同方面，从而捕捉到更丰富的信息。随后，这些不同头的输出被拼接起来，形成一个综合的特征表示，这样做可以显著增强模型的表示能力和泛化能力。

多头注意力层：是深度学习模型中一个极为关键且强大的组件，其核心思想在于“分而治之”。具体而言，该层将输入的特征数据分割成多个部分，每个部分由一个独立的“头”来负责处理。每个头都会基于自身的权重和机制，专注于特征数据的不同方面或维度，从而捕捉到更加细致和丰富的信息。通过这样的方式，模型能够更全面地理解输入数据的内在结构和特性。在处理完各自负责的特征后，这些不同头的输出会被拼接起来，形成一个更加综合和全面的特征表示。这种机制不仅增强了模型的表示能力，还使其能够更好地适应和处理各种复杂的数据和任务。

前馈网络：在注意力层之后，它的主要任务是对注意力层输出的特征进行进一步的非线性处理和变换。前馈网络通常由两个线性层组成，这两个线性层之间会夹带 ReLU 激活函数，以增加网络的非线性能力。这样的设计使得模型能够提取出更深层次、更抽象的特征信息，从而进一步提高其性能。

特征融合层：将多头注意力层的输出与原始特征图进行融合的关键步骤。这一层的作用在于将注意力机制捕捉到的关键特征与原始特征图中的信息进行结合，以充分利用输入数据的所有可用信息。融合的方式通常有两种：一种是加权求和，即根据一定的权重将两部分特征进行线性组合；另一种是特征拼接，即将两部分特征直接拼接在一起，形成一个更加丰富的特征表示。通过这样的融合方式，模型能够更全面地理解 and 处理输入数据，从而在各种任务中表现出更加出色的性能。

将其公式简化为：

$$\hat{F} = A(F_1, F_2, \dots, F_N) = \text{Concat}(\text{MHA}_k(F_i, F_j))_{i,j,k} \quad (4)$$

式中： MHA_k 表示第 k 个头的多头注意力操作，作用于特征图 F_i 和 F_j ， i, j 的组合取决于注意力机制的设计； Concat 表示将不同头的输出拼接起来。此公式省略了详细的注意力机制和特征融合的具体实现。

4.4 去模糊结果重建模块

(1) 重建方法的选择与实现

该模块采用逐步上采样的方式重建清晰图像。首先，从最小尺度的特征图开始，通过反卷积层或转置卷积层进行上采样，并逐步增加分辨率。在每个上采样步骤后，可以引入额外的卷积层来细化图像细节。最终，通过一系列卷积层将特征图转换为与输入图像相同分辨率的清晰图像。

(2) 引入后处理步骤以提升图像质量

为了进一步提升图像质量，可以在重建模块之后引入后处理步骤。这些步骤可能包括去噪（使用图像去噪算法去除噪声）、锐化（增强图像的边缘和细节）以及色彩校正（调

整图像的色调、饱和度和亮度等)。

(3) 损失函数的设计与优化

损失函数是训练过程中指导网络学习的重要工具,它衡量了模型预测结果与真实数据之间的差异。对于去模糊任务,设计合理的损失函数对于提升模型性能至关重要。常用的损失函数包括像素级损失。

L1 损失 (mean absolute error, MAE): 计算预测图像与真实图像之间像素差的绝对值之和。L1 损失对异常值较为鲁棒,有助于保持图像的边缘和细节。

L2 损失 (mean squared error, MSE): 计算预测图像与真实图像之间像素差的平方和的平均值。L2 损失倾向于产生更平滑的结果,但在某些情况下可能导致图像过于模糊。

为了平衡边缘保持和平滑性,文中采用 L1 和 L2 损失的组合,或者根据训练阶段动态调整它们的权重。

重建过程通常包含一系列的上采样和卷积操作,可以表示为:

$$I_{\text{sharp}} = R(\hat{F}) = U^{(m)} \circ C^{(m)} \circ \dots \circ U^{(i)} \circ C^{(i)}(\hat{F}) \quad (5)$$

式中: $U^{(i)}$ 表示第 i 个上采样操作, $C^{(i)}$ 表示第 i 个卷积操作。

5 实验分析

GoPro 数据集是一个专门用于图像去模糊任务的公开数据集,由上海人工智能创新中心等机构提供。该数据集包含 3214 张大小为 1280×720 的图像,分为 2103 张训练图像和 1111 张测试图像。这些图像由高速摄像机拍摄,确保了数据的真实性和多样性。数据集由成对的模糊图像和对应的清晰图像组成,为训练图像去模糊模型提供了宝贵的资源。通过在该数据集上进行训练和测试,可以评估和优化图像去模糊算法的性能,推动图像处理技术的发展。

图 2 为算法在 GoPro 测试集上去模糊结果。从上到下分别为模糊图像、本算法处理后的图像的对比,可以看出本文的方法在去模糊方面表现出色,图片更加清晰。



图 2 算法在 GoPro 测试集上去模糊结果

6 结论

本文围绕图像去模糊问题,提出了一种基于多尺度交叉模块的图像去模糊算法。通过深入分析模糊图像在不同尺度下的特征表现,结合注意力机制,实现了对模糊图像的有效复原。论文构建了一个多尺度特征提取网络,能够捕获图像在不同尺度下的模糊特征。这种多尺度处理方式不仅有助于捕捉图像的局部细节,还能更好地描述图像的全局结构,为后续的去模糊处理提供丰富的信息源。针对传统去模糊方法在特征融合上的不足,设计了一个基于注意力机制的多尺度交叉模块。该模块通过空间注意力推断,将较小尺度下的细节信息精准地补充到较大尺度的模糊特征中,有效提升了特征的精细度和重建图像的清晰度。实验部分,在 GoPro 数据集上的实验结果表明,本文提出的多尺度交叉模块图像去模糊算法效果显著,算法在处理复杂运动模糊、提升图像质量方面的有效性和优越性明显,具有重要的理论意义和实际应用价值。

参考文献:

- [1] 胡张颖,周全,陈明举,等. 图像去模糊研究综述 [J]. 中国图象图形学报, 2024,29(4):841-861.
- [2] 王强,曾志男. 基于图像去模糊算法的缺陷检测信号增强方法 [J]. 中国激光, 2024,51(23):2304003.
- [3] 郑宇航,曹维清. 基于多尺度特征混合注意力的连续帧深度估计 [J]. 重庆工商大学学报(自然科学版), 2024, 41(4): 104-111.
- [4] 张艳珠,赵赫,刘义杰. 改进多尺度 U 型网络的红外图像去模糊方法 [J]. 沈阳理工大学学报, 2024,43(1):55-60.
- [5] 刘利平,孙建,高世妍. 单图像盲去模糊方法概述 [J]. 计算机科学与探索, 2022,16(3):552-564.
- [6] 王璐雪,王晓霞,李翔,等. 并行多尺度特征递归学习的低照度图像增强 [J/OL]. 计算机工程与应用, 1-8[2024-06-17]. <http://kns.cnki.net/kcms/detail/11.2127.TP.20240925.1212.028.html>.
- [7] 赵倩,周冬明,杨浩,等. 结合多尺度特征融合和多输入多输出编-解码器的去模糊算法 [J]. 红外与激光工程, 2022, 51(10): 353-365.
- [8] 虞志军,王国栋,张镔月. 基于增强多尺度特征网络的图像去模糊 [J]. 激光与光电子学进展, 2022,59(22):264-271.
- [9] 李旺鑫. 基于注意力机制的多尺度特征融合图像去模糊算法研究 [D]. 湘潭:湘潭大学, 2021.

【作者简介】

谢壹珍(1991—),女,福建泉州人,本科,助教,研究方向:数字媒体技术。

(收稿日期:2024-07-08)