

MOE-IR_30G 图像恢复算法设计文档

1. 概述

在实际应用场景中，复杂多变的环境、有限的设备性能等因素，导致图像质量问题频发。逆光下的细节丢失、低光照环境里的噪点丛生、强光引发的眩光干扰，ISO 设置错误导致的噪点，以及物体快速移动产生的运动模糊等，都对图像处理技术提出了更为严苛的挑战。

图像恢复（IR）指的是在去除诸如噪声、模糊、天气影响等图像降质因素的同时提升图像视觉质量的过程。传统的图像恢复方法通常针对特定类型的图像降质情况，这限制了它们在存在复杂失真的现实场景中的有效性。

为应对这一挑战，本文采用“一体化图像恢复”（All-in-one image restoration, AiOIR）范式开发轻量化深度学习模型，开发了一个统一的框架，能够地处理多种不同类型的图像退化问题。模型通过自适应地学习特定于降质类型的特征，同时利用不同损坏情况之间的共享知识，提高了便利性和通用性。

本文着重从模型结构，数据集和训练策略三个方面进行模型算法的开发。模型主体是一个非对称的 U 型结构，编解码器主要使用了头数较少的低参数量的 Transformer 模块，并在解码器集成了混合复杂度专家（MoCE）模块。数据集方面，收集了大量不同类型的公开数据集用于训练，总数量达到 43465 张不同的图片。尝试了不同的 loss 损失函数组合，还采用了渐进式的学习策略。

2. 问题定义

本任务要求设计一种基于深度学习的图像去噪与增强算法，满足以下核心需求：

输入：多场景、多分辨率、多种退化类型的噪声图像。

输出：去噪后的清晰图像，需保持原始内容的完整性，无信息损失或伪影引入。

● 算法限制：

计算复杂度限制：对于 720P 分辨率（ 1280×720 ）的输入图像，模型的乘加操作次数（MACs）须小于 30G。

3. 算法设计

如图 1(a) 所示,密集连接型一体化图像修复模型在处理多种图像退化类型时,参数分配效率较低。鉴于图像退化类型多样且需模型轻量化,本文采用混合专家架构 (MOE), 针对不同退化类型调用对应专家处理, 推理时仅激活部分专家参数, 以提升修复效果并降低计算量。

然而,混合专家架构存在表现不一致的问题: 部分专家意外具备多任务泛化能力, 另一些却难以正常发挥作用, 导致推理时无法跳过不相关专家, 难以发挥该架构的计算优势。此外, 如图 1(b) 所示, 僵化的路由机制在均匀分配输入时, 未考虑退化情况的内在联系。为此, 本文使用混合复杂度专家框架 (图 1(c)), 各专家配备不同规模计算单元的自适应处理模块。该框架引入基于弹簧原理的力机制, 动态分配处理能力, 优先引导路由选择更简单的专家, 力的大小与图像退化复杂度正相关。此方法不仅提升了计算效率, 还形成具备任务判别能力的学习框架, 可将不同退化情况精准分配给适配专家。

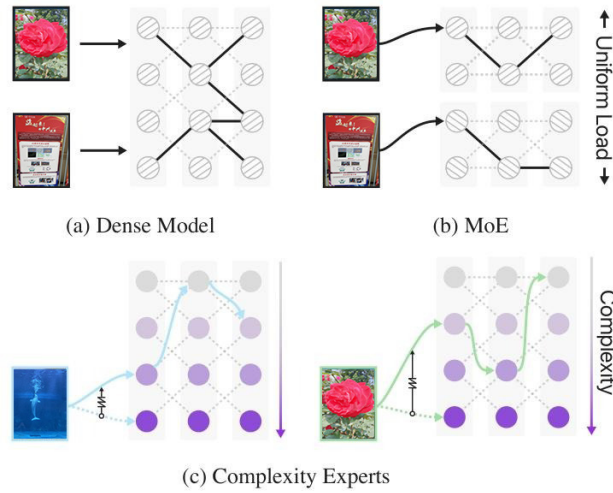


图 1 三种模型架构数据流示意图

3.1 模型总体设计

如图 2(a)所示, 本文所用的模型采用了 U 形架构[1], 具有非对称的编码器-解码器设计。首先用一个 3×3 卷积从退化的输入中提取浅层特征, 然后这些特征会经过四个层级的编码和解码阶段。这两个阶段都使用了 Transformer 模块[2], 并将一个新颖的混合复杂度专家 (MoCE) 层集成到了解码器模块中 (图 2(b))。为了进一步增强解码器的特征丰富化能力, 本文用一个尺寸为 3×3 , 中心值为 8, 其余位置全为 -1 的 Sobel 滤波器提取全局特征向量, 从而引入高频引导, 以提升门控函

数的频域感知能力。最后，一个全局残差连接将浅层特征与细化模块的输出相连，在生成修复图像之前对关键的高频细节进行细化。

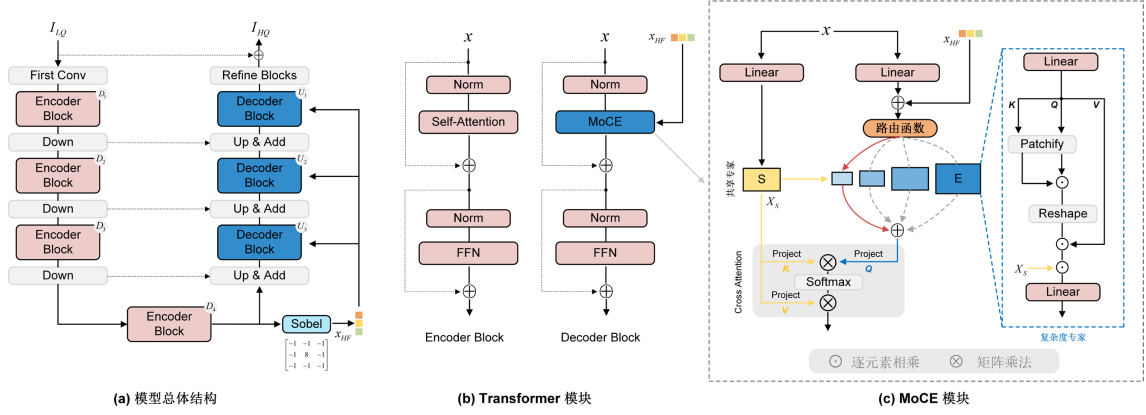


图 2 MOE-IR_30G 模型架构示意图

3.2 混合复杂度专家模块（MoCE）

混合复杂度专家是一种高效的输入自适应模型，用于一体化图像恢复。该模型能够动态调整处理能力，以匹配任务需求。该框架的核心是一个专门的混合复杂度专家（MoCE）层，它由 n 个具有不同复杂度的专家 E 和一个共享专家 S 组成。如图 2(c) 所示，与传统混合专家设计[3]的统一架构不同，本文的专家被设计为具有不断增加的计算复杂度和逐渐增大的感受野。这些组件之间的交互是通过一个两级门控机制来协调的：首先将经过路由的 tokens 投影到特定于专家的嵌入中，然后通过调制将专门的特征和通用特征相结合，最后通过交叉注意力(Cross Attention)将输出合并。这使得本文的模型既能捕捉特定于图像退化的特征，又能捕捉不同退化情况之间的关系，同时保持对计算效率高的处理路径的偏好。

如图 2c 所示，每个复杂度专家模块遵循两项设计原则：提高计算效率并捕获分层空间特征，这对恢复细节和处理多种退化模式至关重要。为控制计算成本，本文采用嵌套式结构，逐步降低后续专家的通道维度 r ，同时增大窗口划分大小 w ，以自适应平衡局部与全局处理。因此，复杂度专家通过窗口大小和通道维度两个轴向缩放构建。

每个专家 E 将其输入 tokens 投影到嵌入 $x_E \in \mathbb{R}^{H \times W \times r}$ 中，其中 $r = C/2^i$ ， $i \in \{1, \dots, n\}$ ，这样做是为了在通道维度上突出最相关的信息。在此之后，使用基于窗口的自注意力（Window-based Self-Attention, WSA）机制有效地捕捉空间信息。最后，专家模块的输出 \hat{x}_E^i 通过与来自共享专家 S 的整体特征进行逐元素相乘来进

行调制。作为共享专家，在通道维度上采用转置自注意力（T-SA）模块[2]，而复杂度专家则在空间上进行操作，经过调制的专家特征 y_E^i 如下式所示。

$$y_E^i = \hat{x}_E^i \odot S(x) \quad (1)$$

$$\text{with } \hat{x}_E^i = E^i(x_E^i) = \text{WSA}_{w_i}(W_{C \rightarrow r_i}^i x_E^i) \quad (2)$$

$$\text{and } S(x) = \text{T-SA}(x) \quad (3)$$

其中，用于投影的线性层用 W 表示， \odot 表示逐元素相乘， i 是当前复杂度专家 E^i 的索引，该专家的窗口大小为 w_i ，嵌入维度为 r_i 。例如，最轻量级的专家的特点是具有最小的嵌入维度 r_1 和窗口大小 w_1 ，而其他专家的维度和窗口划分可以线性增加。在本文的整个网络中，共享专家的能力保持不变， $r = C$ 。

3.3 复杂度感知路由

受稀疏混合专家（MoE）方法[3]的启发，本文在每个解码器模块中集成了线性层，以实现一种路由机制，将输入特征 $x \in \mathbb{R}^{H \times W \times C}$ 与其对应的特定复杂度专家 E 关联起来。为了实现对输入图像尺度不变的 token 化，确保在不同分辨率和尺度下保持一致性，本文选择了图像级路由策略，即为整个输入图像选择专家。

路由函数 $g(x)$ 根据所需的计算量，将输入样本分配给相应的复杂度专家 E^i ，其中 $i \in \{1, \dots, n\}$ ， n 表示专家的总数。在 $g(x)$ 中，模型将选择输入 token 与专家关联的 softmax 分布中的前 k 个元素，并将其他元素都置为零。在实际应用中，本文取 $k=1$ ，同时 ϵ 是独立采样的， $\epsilon \sim N(0, 1/n^2)$ （按元素采样），从而实现带噪声的 top-1 路由，公式表示如下。

$$g(x) = \text{top}_k(\text{Softmax}(Wx + \epsilon)) \quad (4)$$

在训练过程中，为了根据专家的计算复杂度来偏向某些专家，从而实现与专家能力更有效的平衡，本文使用了一个文献[3]中改进的辅助损失 L_{aux} 。该损失由负载损失和重要性损失两部分组成。

对于负载损失，理想情况下，本文希望能明确平衡分配任务的数量。这个数量是离散的；因此它不可微，需要依赖一个替代指标。为了简化，本文对文献[4]中的定义稍作修改。对于每个 token x ，定义一个分数阈值，高于该阈值的专家会被选中，这仅仅是第 k 大的分数：

$$\text{threshold}_k(x) = \max_{k\text{-th}}(Wx + \epsilon) \quad (5)$$

其中 ϵ 是在前向传播过程中最初采样得到的噪声向量。然后，对于每个专家 i ，本文只对其噪声进行重新采样，计算 i 高于该阈值的概率：

$$p_i(x) = P((Wx)_i + \epsilon_{\text{new}} \geq \text{threshold}_k(x)) = P(\epsilon_{\text{new}} \geq \text{threshold}_k(x) - (Wx)_i) \quad (6)$$

该概率是在 $\epsilon_{\text{new}} \sim N(0, \sigma^2)$ 上定义的，其中 $\sigma = 1/n$ 。在批次 X 上专家 i 的负载为：

$$\text{load}_i(X) = \prod_{x \in X} p_i(x) \quad (7)$$

本文对文献[43]中的重要性项进行了补充，从而实现与专家能力更有效的平衡。通过一个重要性损失来促进专家的均衡使用，其中对于一批图像而言，每个专家 E^i 的重要性被定义为在整个批次中与第 i 个专家相关的路由权重之和。此外，本文通过计算每个专家 E^i 中可学习参数的数量 p_i ，并根据专家中最大的参数数量 p_{\max} 对这些值进行归一化，从而引入了一个复杂度偏差 \mathbf{b} ：

$$\begin{aligned} \text{Imp}_i(x) &= \left(\sum_{x \in X} \text{Softmax}(Wx)_i \right) \times \mathbf{b} \\ \text{with } \mathbf{b} &= [p_1 / p_{\max}, p_2 / p_{\max}, \dots, p_n / p_{\max}] \end{aligned} \quad (8)$$

其中， W 是路由函数 $g(x)$ 的特定层权重。这种方法会给计算复杂度较低的专家分配逐渐降低的权重。为了便于直观理解，这种机制类似于一个机械弹簧系统，在这个系统中，恢复力与位移和弹性系数都成正比。在此处的情境中，专家的参数数量相当于位移，而归一化因子则充当材料特定的常数，它们共同决定了复杂度偏差的大小。如果没有这个机制，路由函数就缺乏选择特定专家的依据，从而导致随机路由，进而使模型的能力无法得到最优利用。最后，考虑复杂度的重要性 Imp 被用于计算总的辅助损失 L_{aux} ，其定义如下：

$$L_{\text{aux}}(x) = \frac{1}{2} \text{CV}(\text{Imp}(x))^2 + \frac{1}{2} \text{CV}(\text{Load}(x))^2 \quad (9)$$

其中， CV 表示变异系数[3]。

3.4 损失函数

通过查阅大量文献资料，本文提出了一种加权组合损失函数，该函数由 $L1$ 损失、多尺度结构相似性指数度量（MS-SSIM）[5]、 $L2$ 损失、频域损失以及用于图像去噪的VGG感知损失[6]组成。在图像降噪和增强等任务中，仅使用 $L1$ 损失会生成过于锐利且视觉效果不佳的图像。为了克服 $L1$ 损失的这一缺点，本文提出了一种采用VGG-19感知损失、多尺度结构相似性指数度量（MS-SSIM）、 $L2$ 损失、频域损失以及 $L1$ 损失的加权组合损失函数。本文使用VGG-19感知损失来

有效地恢复图像中的颜色和整体结构等特征。多尺度结构相似性指数度量（MS-SSIM）损失有助于恢复局部和全局的结构信息， $L1$ 损失和 $L2$ 损失则用于最小化像素之间的广义误差。频域损失用于捕捉图像的全局结构（低频分量）和细节纹理（高频分量），适合图像降噪任务，故也组合进 $LOSS$ 函数。

3.4.1 VGG-19 感知损失

空间和上下文信息指的是诸如颜色、对比度、亮度以及局部和全局结构细节之类的特征。图像中的空间和上下文信息在图像去噪中起着主要作用。为此，本文考虑使用 VGG-19 感知损失来恢复空间和上下文信息。VGG-19 感知损失是一种基于内容的损失，它建立在预训练好的 19 层 VGG 网络的 ReLU 激活层之上，其表达式为：

$$L_{VGG} = \frac{1}{WH} \sum_{i=1}^W \sum_{j=1}^H |\phi_k(\hat{y})_{i,j} - \phi_k(x)_{i,j}|_1 \quad (10)$$

其中， $\phi_k(\cdot)$ 是 VGG-19 网络 ϕ 在处理图像 x 时第 k 层的激活值。 W 和 H 分别是第 k 层特征图的宽度和高度， $|\cdot|_1$ 表示 $L1$ 范数。

本文依据经验和实验结果，选取的 VGG-19 网络的层为第[3, 8, 13]层，名称为 `relu1_2`, `relu2_2` 和 `relu3_2`。`relu1_2`（第 3 层），是 VGG 网络中最浅层的特征图之一，对高频信号敏感，能有效引导模型恢复细节。`relu2_2`（第 8 层），位于第二个卷积 Block 末端，捕捉局部边缘和纹理，适合恢复细节但可能保留噪声。`relu3_2`（第 13 层），位于第三个卷积 Block 末端，提取中级结构和轻度语义，能抑制噪声但可能模糊细节。这三层的损失通过加权求和的方式组合在一起，作为最终的 VGG 损失，权重为[0.1, 0.6, 0.3]。

3.4.2 MS-SSIM 损失

VGG-19 感知损失更侧重于恢复上下文信息，这就要求改进结构信息。为此，本文考虑使用多尺度结构相似性指数度量（Multi-scale Structural Similarity Index Measure, MS-SSIM）作为损失函数，以学习全局和局部结构信息。MS-SSIM 如公式 2 所示，在多个不同的尺度上计算结构相似性指数（SSIM）。

$$MSSSIM = l_m(x, y)^{\alpha_m} \cdot \prod_{j=1}^m \left[c_j(x, y) \right]^{\beta_j} \left[s_{j(x, y)} \right]^{\gamma_j} \quad (11)$$

其中， l_m 是第 m 个尺度的亮度， c_j 是对比度参数， s_j 是结构参数。 β_j 和 γ_j 定义了亮度、对比度和结构参数之间的相对重要性。本文将 L_{MSSSIM} 定义为：

$$L_{MSSSIM} = 1 - MSSSIM \quad (12)$$

本文计算 5 个尺度的 SSIM，即从原始尺寸下采样 5 次，每次下采样至一半的大小。5 个不同尺度的 SSIM 加权求和，权重分别为[0.0448, 0.2856, 0.3001, 0.2363, 0.1333]。这组权重是通过人类视觉实验拟合得出的经验值，反映了 1、人类视觉系统（HVS）特性：人眼对中等尺度（尺度 2-4）的结构变化最敏感。2、信息重要性：中等尺度包含更多语义信息（如物体边缘），比高频噪声或全局亮度更重要[5]。

3.4.3 $L1$ 损失、 $L2$ 损失和频域损失

$L1$ 损失（也称平均绝对误差，MAE），计算恢复图像 \hat{y} 与 GT 图像 y 的像素级绝对差值的均值，公式如下。

$$L_1 = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (13)$$

优点是对异常值（如噪声）鲁棒性强，损失函数梯度稳定，适合生成任务（如图像恢复、超分辨率）。缺点是在平滑区域可能产生模糊效果。

$L2$ 损失（也称均方误差，MSE），计算恢复图像 \hat{y} 与 GT 图像 y 的像素级平方差的均值，公式如下。

$$L_2 = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (14)$$

相较于 $L1$ 损失， $L2$ 的收敛性较差，且容易过于平滑，因此本文将其作为辅助作用设置了一个较小的权重。

频域损失（Frequency Domain Loss），在频域（快速傅里叶变换）中计算 $L1$ 范数，公式如下。

$$L_{freq} = \frac{1}{N} \sum_{k=1}^N \| \mathcal{F}(y)_k - \mathcal{F}(\hat{y})_k \|_1 \quad (15)$$

其中 \mathcal{F} 表示快速傅里叶变换， k 为频域分量。

最终，本文们得到损失函数 \mathcal{L}_{MOEIR} 如下所示。

$$\mathcal{L}_{MOEIR} = (\alpha \times L_1) + (\beta \times L_{VGG}) + (\gamma \times L_{MSSSIM}) + (\lambda \times L_2) + (\theta \times L_{freq}) + (\eta \times L_{aux}) \quad (16)$$

其中， α 、 β 、 γ 、 λ 和 θ 是权重。本文通过实验将权重设置为 $\alpha = 1.5$ ， $\beta = 0.5$ ， $\gamma = 0.7$ ， $\lambda = 0.1$ ， $\theta = 0.1$ ， $\eta = 0.05$ 。

4. 数据集

训练模型纳入了以下数据集：在合成图像去噪方面，本文将 BSD400 和 WED 数据集合并，通过添加噪声水平 $\sigma \in [15, 25, 50]$ 的高斯噪声以及 Scale=100 的泊松噪声来生成含噪图像。在真实图像去噪方面，本文使用了 SIDD、NIND 和 PolyU 数据集。对于 SIDD 数据集，本文将其中的每张图片裁切至了 512x512 的大小以便于处理和扩充数目，裁切后的 SIDD 数据集包含 30608 个图像块。对于 NIND 数据集，将每个图片组中最小的 ISO 值的图片作为 GT，其余不同大小的 ISO 值的图片作为与其对应噪声图片，一共有 710 对图像对。

对于图像去模糊任务，本文使用了 GoPro 、REDS 数据集，其中 REDS 数据集选取了其中的模糊退化(REDS_blur)和模糊混合 JPEG 退化(REDS_blur_jpeg)两种退化类型的数据。

将以上所有数据集合并，得到训练集，其中为了平衡各个数据集的数量，对样本量小的数据集进行了重复，各个数据集的信息如下表所示。

表 1 数据集信息

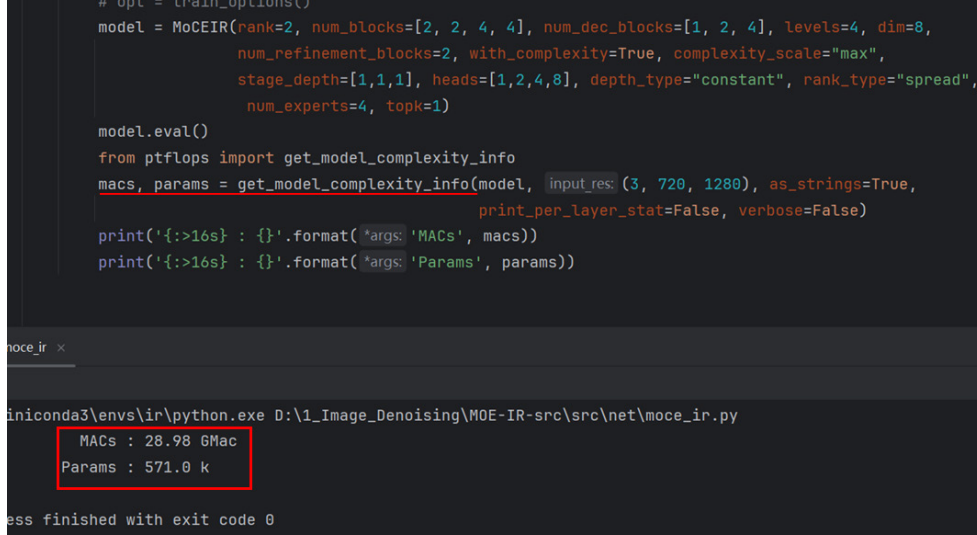
数据集名称	数据集类型	重复次数	不同图片的数量
BSD400	合成图像去噪	4	400
WED	合成图像去噪	4	4744
SIDD	真实图像去噪	1	30608
NIND	真实图像去噪	8	710
PolyU	真实图像去噪	40	100
GoPro	去模糊	3	2103
REDS_blur	去模糊	1	24000
REDS_blur_jpeg	去模糊+jpeg 伪影	1	24000

5. 实验结果

5.1 超参数设置

本文的 MOE-IR_30G 框架是端到端的，该架构具有非对称的四级编码器-解码器设计。编码器由四级组成，每级包含数量不同的 Transformer 模块，从顶层到底层的排列为[2, 2, 4, 4]。解码器包括三级，分别包含[1, 2, 4]个 Transformer 模块。在每个 MoCE 层中，本文使用 $n = 4$ 个嵌套专家，其中专家嵌入的维度由 $R = C / 2^i$ 给出， $i \in \{1, \dots, n\}$ 。网络的初始嵌入维度设置为 $C = 8$ ，在后续的每一级中该维度翻倍，而空间分辨率则减半。

如图 3 所示(运行 src/net/moce_ir.py 脚本统计计算量),在该参数组合设置下,模型处理一张 720P ($3 \times 720 \times 1280$) 的 RGB 彩色图像的乘加操作次数 (MACs) 为 **28.98G**, 参数量为 **571K** (使用 python 包 ptfllops 的 get_model_complexity_info 函数统计)。



```
# opt = train_options()
model = MoCEIR(rank=2, num_blocks=[2, 2, 4, 4], num_dec_blocks=[1, 2, 4], levels=4, dim=8,
               num_refinement_blocks=2, with_complexity=True, complexity_scale="max",
               stage_depth=[1,1,1], heads=[1,2,4,8], depth_type="constant", rank_type="spread",
               num_experts=4, topk=1)

model.eval()
from ptfllops import get_model_complexity_info
macs, params = get_model_complexity_info(model, input_res=(3, 720, 1280), as_strings=True,
                                         print_per_layer_stat=False, verbose=False)
print('{:>16s} : {}'.format('args: MACs', macs))
print('{:>16s} : {}'.format('args: Params', params))

moce_ir x
miniconda3\envs\ir\python.exe D:\1_Image_Denoising\M0E-IR-src\src\net\moce_ir.py
MACs : 28.98 GMac
Params : 571.0 k
Press finished with exit code 0
```

图 3 模型计算量与参数量统计

训练策略方面,本文使用了渐进式的训练方式,第一阶段将 Batch Size 设置为 32 对模型进行 100 轮的训练,使用 AdamW 优化器 ($\beta_1 = 0.9$, $\beta_2 = 0.999$),初始学习率为 2×10^{-4} ,并采用余弦退火衰减策略。在训练过程中,本文使用大小为 128×128 的图像裁剪,并通过水平和垂直翻转来进行数据增强。第二阶段,保持其他设置不变,将裁剪大小增大为 192×192 ,将学习率调整为 2×10^{-5} ,Batch Size 调整为 16,继续训练 20 个轮次。第三阶段,同样保持其他设置不变,将裁剪大小增大为 256×256 ,将学习率调整为 1×10^{-6} ,Batch Size 调整为 8,最后精调 10 个轮次。

所有实验在同一台 PC 完成,操作系统为 Windows 11, CPU 为 AMD Ryzen 5 9600X 6-Core Processor, GPU 为一张 NVIDIA GeForce RTX 5070 Ti 16GB,内存 32GB。固态硬盘 1TB,完整训练一次模型需要 3 天左右。

5.2 后处理滤波

实验中发现,对模型的推理结果使用**双边滤波**处理一下能进一步提升视觉效果和 PSNR 值。使用 opencv 提供的 API(cv2.bilateralFilter)进行双边滤波,其参数设置为 $d=3$, $\text{sigmaColor}=10$, $\text{sigmaSpace}=10$ 。

双边滤波的计算公式如下所示，

$$I_{\text{filtered}}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) \cdot f_r(\|I(x_i) - I(x)\|) \cdot g_s(\|x_i - x\|) \quad (17)$$

其中， Ω 是滤波窗口（邻域），大小为 $d \times d$ （这里 $d=3$ ，即 3×3 窗口）， f_r 是颜色域高斯核，由 ‘sigmaColor’ 控制， g_s 是空间域高斯核，由 ‘sigmaSpace’ 控制， W_p 是归一化因子。

对于每个像素 x ，计算其双边滤波的值时需要：

1. 遍历 3×3 邻域（9 个像素）。

2. 对每个邻域像素 x_i ，计算：

(1) 空间权重 $g_s(\|x_i - x\|)$ ：

计算空间距离 $\|x_i - x\|$ （2D 坐标差）。

计算高斯权重（通常用查表或近似计算）。

约 3 MACs（2 次减法 + 1 次高斯计算）。

(2) 颜色权重 $f_r(\|I(x_i) - I(x)\|)$ ：

计算像素值差 $\|I(x_i) - I(x)\|$ （1 次减法）。

计算高斯权重（1 次高斯计算）。

约 4 MACs（1 次减法 + 3 MACs 高斯计算）。

(3) 乘积 $I(x_i) \cdot f_r \cdot g_s$ ：

2 次乘法（ $I(x_i) \times f_r \times g_s$ ）。

(4) 累加分子和分母：

2 次累加（分子 $\sum I(x_i) \cdot f_r \cdot g_s$ ，分母 $\sum f_r \cdot g_s$ ）。

3. 归一化。

每邻域像素的 MACs 为，

MACs per neighbor = 3(space) + 4(color) + 2(product) + 2(accumulate) = 11 MACs

每像素的 MACs（ 3×3 邻域）为，

MACs per pixel = $11 \times 9 = 99$ MACs

综上，对 720P 输入图像进行一次双边滤波的 GMAC 估算如下式所示。

$$\text{GMAC} \approx \frac{3 \times 1280 \times 720 \times 99}{10^9} = 0.27 \text{ GMACs} \quad (18)$$

因此，本文算法对于 720P 输入图像的最终 GMACs 为 $28.98 + 0.27 = \mathbf{29.25}$ ，符合赛题小于 30G 的要求。另外，OpenCV 的实现优化了计算效率，实际的 GMAC 可能更低。

5.3 部分图像降噪结果展示

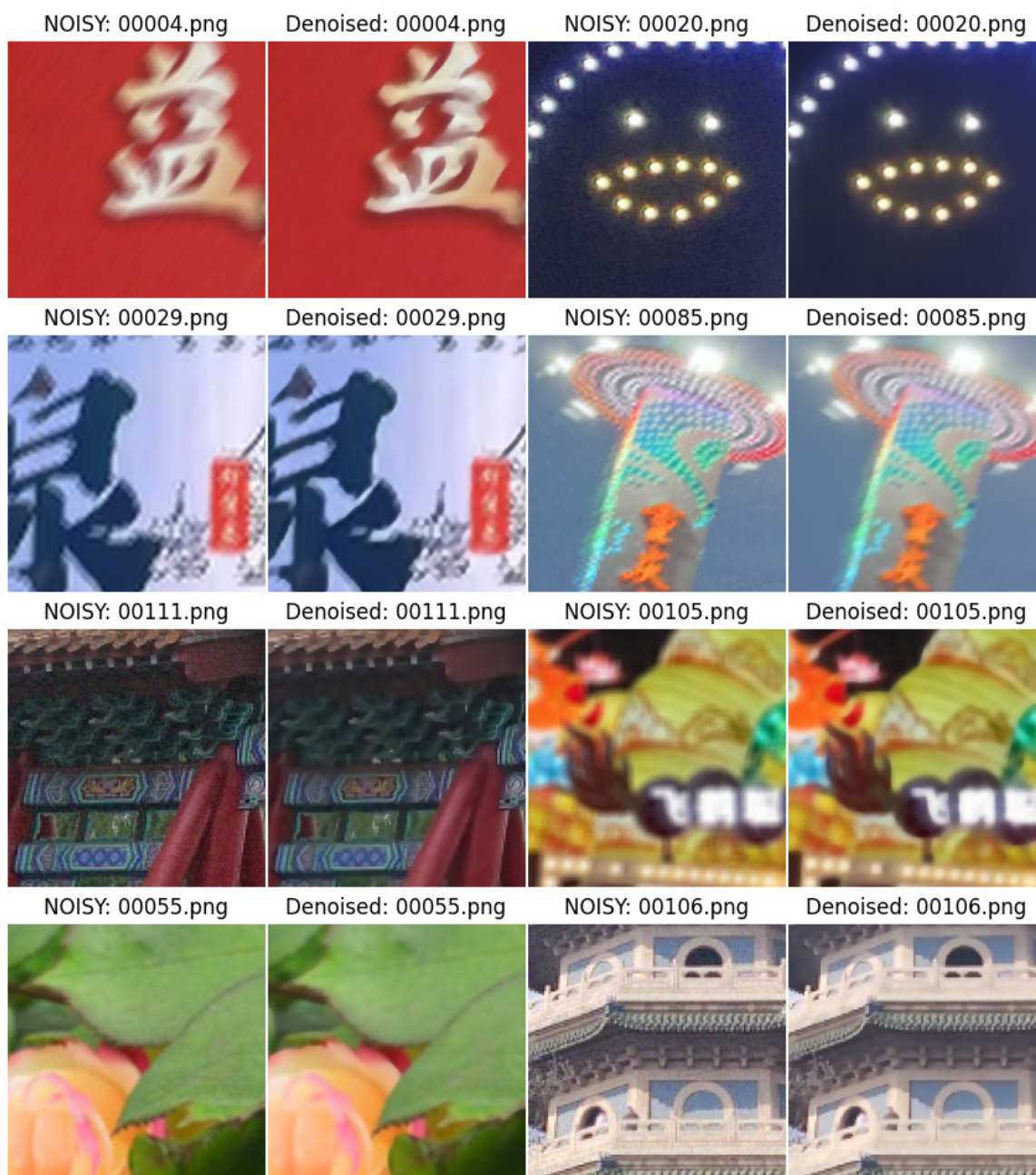


图 4 部分图像降噪结果展示

可以看到，本文的模型能有效降低图片中的各种噪声以及恢复运动模糊等退化。但是模型对有些未见过的退化类型的图片恢复能力较差，后续需要想办法扩充数据集以尽可能多的包含各种退化类型，或者改进模型结构使其泛化能力增强。

上传至网站的测试结果显示，150 张测试图片的 PSNR 值为 **22.63**，取得了不错的域外图像恢复效果。

6. 参考文献

- [1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In MICCAI, pages 234–241. Springer, 2015. 3
- [2] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In CVPR, 2022. 1, 2, 3, 4, 5
- [3] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. NeurIPS, 2021. 2, 3, 4, 7, 8
- [4] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In ICLR, 2017.
- [5] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, pages 1398 – 1402. Ieee, 2003. 4
- [6] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photorealistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4681 – 4690, 2017. 4
- [7] Joshi, Amogh, et al. "HNN: Hierarchical Noise-Deinterlace Net Towards Image Denoising." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024.
- [8] Jiang, Junjun, et al. "A survey on all-in-one image restoration: Taxonomy, evaluation and future trends." *arXiv preprint arXiv:2410.15067* (2024).
- [9] Jiang, Bo, et al. "Efficient image denoising using deep learning: A brief survey." *Information Fusion* (2025): 103013.
- [10] Zamir, Syed Waqas, et al. "Learning enriched features for fast image restoration and enhancement." *IEEE transactions on pattern analysis and machine intelligence* 45.2 (2022): 1934-1948.
- [11] Brummer, Benoit, and Christophe De Vleeschouwer. "Natural image noise dataset." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2019.
- [12] Vaishnav, P., et al. "Promptir: Prompting for all-in-one blind image restoration." *arXiv preprint arXiv:2306.13090* (2023).

- [13] Jiang, Bo, et al. "QFormer: an efficient quaternion transformer for image denoising." *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, Jeju, South Korea*. 2024.
- [14] Zamfir, Eduard, et al. "Complexity Experts are Task-Discriminative Learners for Any Image Restoration." *arXiv preprint arXiv:2411.18466* (2024).
- [15] Zhang, Kai, et al. "Practical blind image denoising via Swin-Conv-UNet and data synthesis." *Machine Intelligence Research* 20.6 (2023): 822-836.