基于多域多尺度扩散模型的低光图像增强算法 (MDMS) 总结与分析

摘要

低光图像增强（Low-Light Image Enhancement, LLIE）是计算机视觉中的经典低层视觉任务，其目标是在提升图像亮度的同时尽量恢复颜色、纹理与结构信息。本文选择 AAAI 2024 论文《Multi-Domain Multi-Scale Diffusion Model for Low-Light Image Enhancement》中提出的 MDMS（Multi-Domain Multi-Scale Diffusion Model）作为分析对象 [1]，从扩散模型基础原理、MDMS 网络设计、算法执行流程、关键代码实现、实验结果与应用场景等方面进行总结。MDMS 在条件扩散模型的基础上引入多域学习和多尺度采样策略：一方面同时利用空间域与频域特征增强亮度恢复和纹理重建能力；另一方面通过多尺度重叠 patch 采样缓解单尺度重建中的棋盘伪影。实验结果表明，该方法在曝光恢复、结构保持和感知质量上均有较好表现，但同时也存在采样链路较长、推理成本较高等问题。

关键词：低光图像增强；扩散模型；频域建模；多尺度采样；MDMS

1 引言

低光环境会导致图像出现亮度不足、噪声增强、对比度下降以及颜色偏移等问题。这不仅影响图像的主观观感，也会对目标检测、分类、分割等下游任务带来明显干扰。传统方法如直方图均衡化、Retinex 等能够在一定程度上改善亮度，但面对复杂场景时往往容易出现过增强、颜色失真或细节丢失。近年来，卷积神经网络、生成对抗网络与扩散模型被广泛用于图像增强，其中扩散模型由于生成质量稳定、细节恢复能力强，逐渐成为 LLIE 的重要研究方向 [2, 3]。

本文选取 MDMS 作为分析对象，重点讨论以下问题：第一，扩散模型为什么适合用于图像恢复；第二，MDMS 如何把条件扩散、多域特征学习与多尺度采样结合起来；第三，该算法在本机评估时表现如何；第四，这类方法在实际应用中有哪些优势与局限。

2 扩散模型基础

2.1 扩散模型的提出动机

扩散模型属于生成模型的一类，其核心目标是学习训练数据的分布，并据此生成与训练样本同分布的新样本。与 VAE、GAN 等前代生成模型相比，扩散模型在生成高质量图像时具有两个显著优势：一是训练过程更稳定，不需要像 GAN 那样维持生成器与判别器之间的动态平衡；二是生成过程通过逐步去噪完成，因此对复杂分布中的细节建模能力更强 [2]。这也是扩散模型能够迅速成为高质量图像生成与图像恢复主流方法的重要原因。

2.2 前向扩散过程

扩散模型的前向过程，就是在原始样本上逐步加入高斯噪声，直到样本分布接近标准正态分布。设原始图像为 x_0，第 t 步的随机变量为 x_t，则其马尔可夫形式可写为：

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)\mathbf{I}), \tag{1}$$

其中 $\alpha_t = 1 - \beta_t$，β_t 为预先设定的噪声调度系数。β_t 通常随着时间步增大而逐渐增大，这意味着在前期加噪较少，在后期加噪更多。

若把上述递推关系不断展开，可以得到一个非常重要的闭式表达式：

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)\mathbf{I}), \tag{2}$$

其中 $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$。因此，任意时间步的带噪样本都可以直接由 x_0 一步采样得到：

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{3}$$

这个公式非常关键，因为训练时无需真的做 t 次递推，只需要随机采样一个时间步 t，就可以直接构造出对应的 x_t。

2.3 逆向去噪过程

扩散模型真正完成生成的，是逆向过程。直观上看，若前向过程是“不断把图像扰乱”，那么逆向过程就是“不断把图像还原”。理想情况下，我们希望学习到：

$$q(x_{t-1}|x_t), \tag{4}$$

也就是已知当前噪声状态 x_t，求上一时刻较干净的状态 x_{t-1}。从贝叶斯角度出发，可以证明在给定 x_0 的情况下，后验分布 $q(x_{t-1}|x_t, x_0)$ 仍然是高斯分布，其均值为：

$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)}{1 - \bar{\alpha}_t}x_0. \tag{5}$$

问题在于，真实采样时 x_0 并不知道，因此无法直接使用这个均值公式。DDPM 的关键思想就是让神经网络去预测加入的噪声 ϵ，再用预测噪声间接恢复 x_0。这也是为什么现代扩散模型通常不直接预测图像本身，而是预测噪声。

2.4 DDPM 训练目标

DDPM 采用的训练目标可以写成：

$$L_{\text{DDPM}} = \mathbb{E}_{x_0,t,\epsilon} \left\| \epsilon - \epsilon_\theta(x_t, t) \right\|_2^2. \tag{6}$$

训练过程可以理解为：随机取一张图像 x_0，随机取一个时间步 t，随机采样一个高斯噪声 ϵ，构造带噪图像 x_t，再让网络根据 (x_t, t) 去回归真实噪声 ϵ。由于这个目标是均方误差，训练过程相对稳定，且容易实现。

2.5 DDIM 采样思想

标准 DDPM 采样需要在每个时间步都注入随机噪声，因此步数较多、速度较慢。DDIM[3] 在不改变训练方式的前提下，对采样过程做了确定性近似，使采样可以在更少步数内完成。其更新形式可写为：

$$\hat{x}_0 = \frac{x_t - \sqrt{1 - \bar{\alpha}_t}\epsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}}, \tag{7}$$

$$x_{t'} = \sqrt{\bar{\alpha}_{t'}}\hat{x}_0 + c_1 z + c_2 \epsilon_\theta(x_t, t), \tag{8}$$

其中 $t' < t$，c_1 和 c_2 为由噪声调度系数决定的常量。MDMS 的采样阶段本质上就是在 DDIM 逆向去噪框架中，引入了多尺度 patch 聚合。

2.6 为什么扩散模型通常使用 U-Net

在扩散模型中，最常见的噪声预测网络是 U-Net[4]。原因在于：编码器能够提取大感受野上下文信息，解码器能够恢复空间分辨率，而跳跃连接又能把高分辨率细节直接传递给后续层。对于“输入是一张带噪图像，输出是同尺寸噪声图”的任务，U-Net 的结构非常自然，因此几乎成为扩散模型中的标准骨干网络。

3 MDMS 算法原理与主要特点

3.1 任务定义与条件输入

MDMS 的目标是学习从低光图像 y 到正常曝光图像 x_0 的映射关系。与一般图像生成任务不同，LLIE 不是“从无到有”地合成全新内容，而是在保持原场景结构的前提下恢复亮度、颜色与纹理。因此，MDMS 采用条件扩散建模：网络在带噪目标图像 x_t 上预测噪声，同时使用低光图像 y 和辅助先验 y_p 提供约束。

该方法额外引入了亮通道先验（Bright Channel Prior, BCP）。根据实现方式，可将每个像素位置处的先验写成：

$$T(i,j) = \max_{c\in\{R,G,B\}} y_c(i,j), \tag{9}$$

$$y_p^c(i,j) = \frac{y_c(i,j)}{T(i,j)+0.1}. \tag{10}$$

这里 $T(i,j)$ 表示当前位置 RGB 三个通道中的最大值，y_p 则把原图像按亮通道做归一化。这个先验能够显式提供亮度参考，使模型在恢复曝光和颜色时更稳定。

3.2 MDMS 的条件扩散目标

在 MDMS 中，训练阶段先把正常光图像 x_0 加噪得到 x_t，再让网络根据时间步 t、低光输入图像 y 和先验图像 y_p 预测噪声。其训练目标为：

$$L_{\text{diff}} = \mathbb{E}_{x_0,t,\epsilon\sim\mathcal{N}(0,I)} \left\|\epsilon - \epsilon_\theta(x_t, y, y_p, t)\right\|_2^2. \tag{11}$$

与普通无条件扩散模型相比，MDMS 的重点不在于重新定义扩散方程，而在于如何利用更有效的条件信息和更合理的采样策略来服务低光图像增强任务。

3.3 多域学习：同时利用空间域与频域

现有许多扩散式 LLIE 方法主要在空间域内学习噪声分布，而忽略了频域中的关键信息 [1]。实际上，图像恢复通常遵循“先恢复低频，再补充高频细节”的规律，因此频域信息对颜色、纹理和边缘恢复非常关键。为此，MDMS 在 U 形主干网络内部引入了 Multi-Domain Learning（MDL）模块。

该模块的核心思想可以概括为两条并行路径：

$$f_{\text{spa}} = \phi_{\text{proj}}\left[\phi_{\text{local}}(x_h) \parallel \phi_{\text{global}}(x_l)\right], \tag{12}$$

$$f_{\text{freq}} = \mathcal{F}^{-1}\left(\phi_A(|\mathcal{F}(x)|),\ \phi_P(\angle\mathcal{F}(x))\right), \tag{13}$$

$$f_{\text{MDL}} = x + \phi_{\text{out}}\left[f_{\text{spa}} \parallel f_{\text{freq}}\right]. \tag{14}$$

其中 $\mathcal{F}$ 表示 FFT，$\mathcal{F}^{-1}$ 表示 IFFT。空间分支负责提取局部和全局上下文信息，频域分支负责显式建模幅度和相位特征，最后两条路径在通道维融合，并通过残差连接回到主干网络，使模型既能提升整体亮度，又能恢复高频细节。

3.4 参数嵌入与位置感知

MDMS 并不是只把时间步 t 送入网络。为了让模型知道“当前 patch 在哪里、当前 patch 多大”，它还把 patch 左上角位置 (i, j) 与 patch 尺度 $osize$ 一并编码。设正弦嵌入函数为 $E(\cdot)$，则可写成：

$$e = [E(t) \parallel E(i) \parallel E(j) \parallel E(osize)], \tag{15}$$

$$\tilde{e} = W_2\sigma(W_1 e). \tag{16}$$

这样一来，模型就同时具备时间感知、位置感知和尺度感知能力。对于 patch-based 推理而言，这一点尤其重要，因为相同大小的 patch 处于图像中心还是边缘，其上下文意义是完全不同的。

3.5 多尺度采样：缓解棋盘伪影

许多 patch-based 恢复方法在推理时会把图像切成固定大小的 patch，再对重叠区域做拼接。若只使用单一 patch 尺度，不同位置的重叠次数往往不均匀，容易在边界处产生棋盘伪影。MDMS 使用 64×64、96×96 和 128×128 三种 patch 尺度，在每个扩散采样步中分别预测噪声并对重叠区域做归一化加权平均，再融合成最终结果 [1]。这样既保留了小 patch 的局部精细恢复能力，也兼顾了大 patch 的上下文一致性。

3.6 算法特点总结

综合来看，MDMS 有以下几个显著特点：

1. 以条件扩散模型为核心，具有较强的图像生成与细节恢复能力；
2. 同时建模空间域与频域特征，能够兼顾整体亮度、局部结构与纹理细节；
3. 在采样阶段引入多尺度 patch 融合，显著减轻单尺度 patch 带来的边界伪影；
4. 结合亮通道先验，增强了模型对低照度场景的恢复稳定性。

4 算法执行过程

为了更准确地说明该算法是如何工作的，本节按照“输入构造、网络前向、训练更新、推理采样”四个层面，逐步分析 MDMS 的完整执行过程。图 1 给出了详细流程图，表 1 总结了主要符号。

表 1: MDMS 中的主要符号及含义

符号	含义
y	低光输入图像
y_p	由亮通道先验构造的辅助条件图像
x_0	正常曝光目标图像
x_t	在时间步 t 上加噪后的目标图像
ϵ	前向扩散过程中加入的高斯噪声
ϵ_θ	条件扩散网络预测的噪声
i, j	patch 左上角的归一化位置坐标
$osize$	patch 的原始尺度大小
W	重叠 patch 的覆盖计数矩阵
$\bar{\alpha}_t$	时间步 t 对应的累计噪声系数

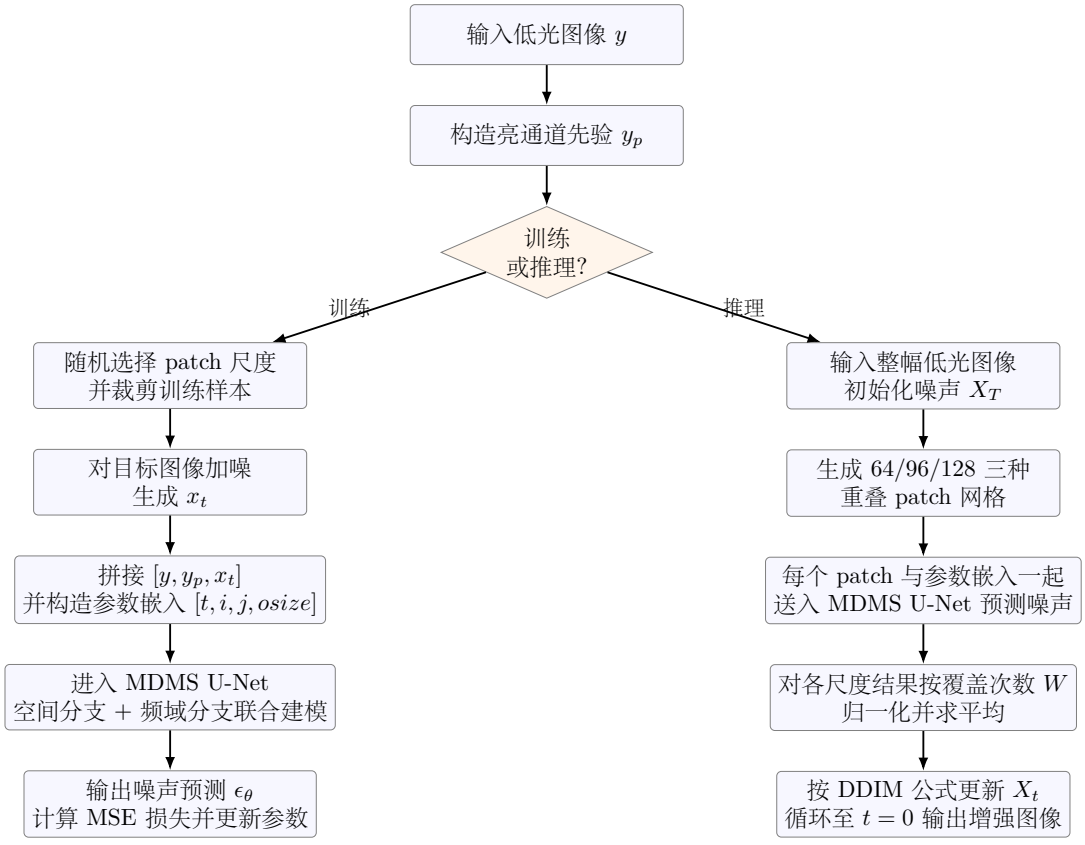


图 1: MDMS 的训练与推理详细流程图

4.1 训练阶段的逐步执行过程

训练阶段的目标，是让网络学会在条件输入 y 和 y_p 的帮助下，准确预测当前时间步上的噪声。其完整流程可分为以下七步：

1. **读入配对样本。**每个训练样本由低光图像 y 和正常光图像 x_0 组成，二者在语义上对齐。
2. **构造亮通道先验。**对低光图像按像素取 RGB 最大值，得到亮度参考，再形成辅助条件图像 y_p。
3. **随机选择 patch 尺度。**训练中会随机裁剪 64、128、256 三种尺度的 patch，再统一缩放到 64×64。这一步的本质是“用不同感受野采样上下文，但在统一分辨率下训练骨干网络”。
4. **随机采样时间步并加噪。**在 $1 \sim T$ 内均匀采样一个时间步 t，再把高斯噪声加到目标图像 x_0 上，得到带噪图像 x_t。
5. **构造网络输入。**把 y、y_p 和 x_t 在通道维拼接，形成 9 通道输入；同时把 t、i、j 和 $osize$ 一起编码为参数嵌入。
6. **经过 U 形扩散网络前向传播。**编码器逐步提取多尺度特征，中间层和解码器继续利用 MDL 模块做多域特征融合，最终输出与噪声同形状的预测结果 ϵ_θ。
7. **计算损失并更新参数。**使用预测噪声与真实噪声之间的均方误差作为训练目标。误差越小，说明网络越能准确理解在不同时间步、不同位置、不同尺度下应该去掉什么噪声。

4.2 网络前向传播中的关键计算

相比一般的 U-Net，MDMS 在一次前向传播中额外做了三件关键工作。

第一，构造条件输入。网络真正看到的不是单独的低光图像，而是 $[y, y_p, x_t]$ 的拼接结果。其中 y 提供原始场景结构，y_p 提供亮度先验，x_t 提供当前时间步上的带噪目标状态。

第二，构造参数嵌入。时间步 t 决定当前去噪阶段，位置 (i, j) 说明 patch 在原图中的相对坐标，$osize$ 表示 patch 原始尺度。网络把这四个量分别做正弦嵌入，再通过线性层映射为统一的高维向量，随后在每个残差块内注入，从而让模型具备时间感知、位置感知和尺度感知能力。

第三，执行多域融合。在 MDL 模块内部，空间域路径会同时提取局部信息和全局信息。局部分支通过卷积建模邻域纹理，全局分支通过池化后的注意力机制建立更大范围的关系；频域路径则先做 FFT，再对幅度和相位分别变换，最后用 IFFT 返回空间域。两条路径的输出在通道维拼接后再经卷积融合，并通过残差连接回到主干特征。这样做的好处是：空间域善于保留结构，频域善于恢复纹理和亮度变化，两者结合后更适合低光图像增强。

4.3 推理阶段的逐步执行过程

推理阶段不再使用真实正常光图像，而是从噪声开始反向采样，逐步恢复出增强结果。该过程可分为以下八步：

1. **输入整幅低光图像。**首先对输入图像生成亮通道先验，并把图像尺寸调整到便于 patch 划分和网络计算的大小。
2. **初始化高斯噪声。**从标准正态分布采样 $X_T \sim \mathcal{N}(0, I)$，作为反向采样的起点。
3. **建立多尺度 patch 网格。**在当前图像上分别生成 64×64、96×96 和 128×128 的重叠 patch 集合。不同尺度覆盖的上下文范围不同，因此会提供互补信息。

4. **逐个时间步逆向去噪。**从较大的时间步逐步迭代到较小的时间步。在每个时间步上，对三种尺度下的所有 patch 分别送入网络，得到对应的噪声预测。

5. **按位置回填 patch 输出。**每个 patch 的预测结果会被放回原图对应的位置。由于 patch 存在重叠，同一个像素可能收到多个预测值。

6. **用覆盖矩阵做归一化。**记 W 为覆盖计数矩阵，表示每个像素被多少 patch 命中。则每个尺度下的聚合结果都需要除以 W，以避免重叠区域数值偏大。

7. **融合多尺度结果。**将 64、96、128 三个尺度对应的噪声预测求平均，得到当前时间步的整体噪声估计 $\hat{\epsilon}_t$。

8. **使用 DDIM 公式更新状态。**先估计当前的无噪图像，再根据 DDIM 更新到下一时间步。重复上述过程，直到 $t = 0$，就得到最终增强图像。

4.4 为什么多尺度采样能够缓解棋盘伪影

单尺度 patch 重建最大的风险，是某些像素处于 patch 中心、某些像素处于 patch 边缘，不同位置接收到的上下文质量不同，导致最后拼接时出现周期性边界痕迹。MDMS 用两种方式减轻这一问题：其一，通过覆盖矩阵 W 对重叠区域做归一化，使每个像素的输出变成所有覆盖 patch 的平均值；其二，通过三种 patch 尺度同时采样，让一个像素既能接受小尺度的局部细节信息，也能接受大尺度的全局上下文信息。因此，边界位置不再只由单一 patch 决定，而是由多个尺度、多个重叠 patch 共同决定，最终结果会更加平滑自然。

5 关键代码实现

代码块 1 展示了 MDMS 在训练阶段的两个关键步骤：一是构造亮通道先验，二是在条件扩散框架下预测噪声并计算均方误差损失。这里真正被加噪的是目标正常光图像，而低光图像与亮通道先验始终作为条件输入保留下来。

代码块 1　亮通道先验与条件扩散损失

```
def get_max(self, input):
    T, _ = torch.max(input, dim=0)
    T = T + 0.1
    input[0, :, :] = input[0, :, :] / T
    input[1, :, :] = input[1, :, :] / T
    input[2, :, :] = input[2, :, :] / T
    return input

def noise_estimation_loss(model, x0, t, e, b, i, j, osize):
    a = (1 - b).cumprod(dim=0).index_select(0, t).view(-1, 1, 1, 1)
    x = x0[:, 6:, :, :] * a.sqrt() + e * (1.0 - a).sqrt()
    output = model(torch.cat([x0[:, :6, :, :], x], dim=1),
                   t.float(), i, j, osize)
    return (e - output).square().sum(dim=(1, 2, 3)).mean(dim=0)
```

代码块 2 展示了采样阶段最具代表性的 Multi-Scale Sampling（MSS）策略。它会在不同尺度下分别裁剪 patch、预测噪声、把结果回填到整图对应位置，再用覆盖矩阵做归一化，最后使

表 2: 本机评估得到的平均指标结果

图像数量	PSNR / dB	SSIM	LPIPS
15	27.17	0.881	0.052

用 DDIM 公式更新当前时间步状态。这正是 MDMS 区别于普通单尺度 patch 扩散方法的关键所在。

代码块 2　多尺度重叠采样与 DDIM 更新

```
for i, j in zip(reversed(seq), reversed(seq_next)):
    t = (torch.ones(n) * i).to(x.device)
    next_t = (torch.ones(n) * j).to(x.device)
    at = compute_alpha(b, t.long())
    at_next = compute_alpha(b, next_t.long())
    xt = xs[-1].to(x.device)
    et_output = torch.zeros(x_cond.size(0), 3, x_cond.size(2),
                            x_cond.size(3), device=x.device)

    xt_patch = torch.cat([crop(xt, hi, wi, p_size, p_size)
                          for (hi, wi) in corners], dim=0)
    x_cond_patch = torch.cat(
        [data_transform(crop(x_cond, hi, wi, p_size, p_size))
         for (hi, wi) in corners], dim=0)
    outputs = model(x_input, t, ii_input, jj_input, osize_input)
    et_output[0, :, hi:hi + p_size, wi:wi + p_size] += outputs[idx]

    et0 = torch.div(et_output, x_grid_mask)
    et1 = torch.div(et_output1, x_grid_mask1)
    et2 = torch.div(et_output2, x_grid_mask2)
    et = (et0 + et1 + et2) / 3.0

    x0_t = (xt - et * (1 - at).sqrt()) / at.sqrt()
    xt_next = at_next.sqrt() * x0_t + c1 * torch.randn_like(x) + c2 * et
```

6　运行结果与效果分析

论文原始实验主要基于 LOL/LOLv2 等低光增强数据集展开 [1, 5]。本文在本机 CPU 环境下运行评估程序，对 15 张测试结果进行了定量评估。图 2 给出了终端输出截图，表 2 列出了平均指标结果。

从定量结果看，平均 PSNR 达到 27.17 dB，说明增强图像与目标图像在像素层面具有较好的接近程度；SSIM 为 0.881，说明模型在结构保持方面表现较好；LPIPS 为 0.052，说明在感知相似性上也取得了较好的结果。这三个指标共同说明，MDMS 不只是简单把图像“调亮”，而是在一定程度上同时恢复了结构、对比度和纹理。

从图 3 可以看出，MDMS 对低光区域的亮度提升较为明显，同时能够较好地保留场景结构。以样例 23 为例，输出图相比输入图具有更高的可见度和更自然的整体曝光；而在样例 493 中，虽然增强后整体亮度明显改善，但复杂区域的颜色和对比度仍然与真实图像存在差距。这也说明 LLIE 任务本身仍然具有较高难度。

```
Code/NN/Project01/MDMS [ v3.11.15(.venv)]
❯ python evaluation.py -dirA ./datasets/scratch/LLIE/data/lowlight/test/gt \
  -dirB ./results/images/lowlight/lowlight_clean --use_gpu false
Comparing:
./datasets/scratch/LLIE/data/lowlight/test/gt
./results/images/lowlight/lowlight_clean
Setting up [LPIPS] perceptual loss: trunk [alex], v[0.1], spatial [off]
Loading model from: /Users/     /Code/NN/Project01/MDMS/.venv/lib/python3.11/site-packages/lpips/weights/v0.1/alex.pth
1.png, 1.png, 24.68, 0.910, 0.050, 0.1
22.png, 22.png, 27.86, 0.917, 0.043, 0.1
23.png, 23.png, 27.07, 0.887, 0.058, 0.1
55.png, 55.png, 26.32, 0.917, 0.051, 0.1
79.png, 79.png, 33.91, 0.941, 0.022, 0.1
111.png, 111.png, 28.74, 0.936, 0.035, 0.1
146.png, 146.png, 27.98, 0.919, 0.032, 0.1
179.png, 179.png, 22.26, 0.838, 0.088, 0.1
493.png, 493.png, 21.03, 0.677, 0.095, 0.1
547.png, 547.png, 26.35, 0.916, 0.051, 0.0
665.png, 665.png, 28.34, 0.842, 0.067, 0.1
669.png, 669.png, 28.95, 0.923, 0.051, 0.0
748.png, 748.png, 29.55, 0.921, 0.045, 0.0
778.png, 778.png, 25.36, 0.851, 0.057, 0.0
780.png, 780.png, 29.11, 0.819, 0.040, 0.0
Final Result: 27.17, 0.881, 0.052, 1.1s

Code/NN/Project01/MDMS [ v3.11.15(.venv)]  2s
❯
```

图 2: 评估程序在终端中的实际输出截图



图 3: MDMS 增强效果示例。第一行为样例 23，第二行为样例 493；每行从左到右依次为输入、输出和真实图像。

7 算法优缺点与应用场景

7.1 优点

1. **恢复质量较高**：扩散模型对细节与纹理的恢复能力较强，增强结果通常比传统 CNN 方法更加自然。
2. **多域信息更全面**：同时利用空间域和频域特征，有利于恢复颜色、边缘与高频纹理。
3. **边界更平滑**：多尺度重叠采样有效抑制了 patch 拼接造成的棋盘伪影。
4. **具有一定可解释性**：亮通道先验和多尺度策略都能对应到明确的图像恢复动机。

7.2 缺点

1. **推理代价较高**：扩散模型本身采样步数较多，再叠加多尺度 patch 推理，计算开销明显增加。
2. **训练成本较大**：扩散模型通常需要较长训练周期和较多算力支持。
3. **工程复杂度较高**：训练、采样、patch 融合与先验构造都增加了实现复杂度。
4. **极端场景仍有不足**：在噪声特别强、颜色严重失真的低光图像中，输出仍可能与真实图像存在偏差。

7.3 应用场景

MDMS 这类低光增强算法适用于多种实际场景：

1. 夜间监控、安防摄像与智慧交通；

2. 移动端摄影、夜景拍照和视频增强；

3. 自动驾驶、机器人视觉等低照度环境感知任务；

4. 医学成像、工业检测、水下或弱光环境目标观察。

8 总结

本文围绕 MDMS 低光图像增强算法，对扩散模型基础、主要结构设计、执行流程、关键实现与实验效果进行了总结。MDMS 的创新点主要体现在三个方面：一是把空间域与频域联合建模，引导扩散模型学习更完整的恢复特征；二是采用多尺度重叠采样缓解 patch 伪影；三是引入亮通道先验增强恢复稳定性。总体来看，MDMS 是一种兼具生成能力与工程技巧的低光增强方法，适合作为扩散模型在低层视觉任务中的代表性案例进行学习与分析。

未来若继续深入这一方向，可以从轻量化采样、跨数据集泛化、零样本适应以及与 Transformer 或更大视觉模型先验结合等角度进一步改进，以提升低光增强在真实场景中的部署价值。

参考文献

[1] Kai Shang, Mingwen Shao, Chao Wang, Yuanshuo Cheng, and Shuigen Wang. Multi-domain multi-scale diffusion model for low-light image enhancement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4722–4730, 2024.

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

[3] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015.

[5] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. In *British Machine Vision Conference*, 2018.