

RealFill 的复现与研究

摘要

本文 [9] 提出了一种名为 RealFill 的新方法，用于解决真实图像修复问题。给定一组参考图像和一个目标图像，RealFill 能够生成与原始场景一致的内容，填补目标图像中的缺失区域。与现有的基于提示的图像修复方法不同，RealFill 通过微调预训练的扩散模型，使其能够利用参考图像中的场景信息，从而在目标图像中生成忠实于原始场景的内容。该方法能够处理参考图像与目标图像之间的显著差异，如视角、光照、相机光圈和图像风格等。实验表明，RealFill 在多样化的图像修复任务中表现优异，显著优于现有的基准方法。本文还提出了一个新的评估数据集 RealBench，用于定量评估真实图像修复任务的性能。RealFill 的创新性在于首次实现了多参考图像驱动的图像修复，能够在复杂场景中生成高质量且忠实于原始场景的图像内容。

关键词：真实图像修复；扩散模型；图像处理；计算机视觉；

1 引言

随着生成图像技术的快速发展，图像修复和扩展模型能够生成高质量、逼真的图像内容，但这些模型生成的内容往往是虚构的，缺乏对真实场景的了解。现有的基于提示的图像修复方法虽然能生成合理内容，但无法利用参考图像中的真实场景信息，导致生成内容与原始场景不符。为此，本文提出 RealFill，通过参考图像驱动的生成实现真实图像修复。RealFill 微调预训练的扩散模型，利用少量参考图像中的场景信息，在目标图像中生成与原始场景一致的内容，能够处理视角、光照、相机光圈和图像风格等差异，生成视觉上令人信服且忠实于原始场景的图像内容。RealFill 不仅填补了现有图像修复方法的不足，还为真实场景的图像修复提供了新的解决方案，具有重要的理论意义和实际应用价值，可广泛应用于修复老照片、损坏图像、扩展图像内容等场景，帮助用户更好地保存和分享珍贵记忆。

2 相关工作

2.1 预训练扩散模型的适配

近年来，扩散模型在文本到图像生成 (T2I) 任务中表现出色 [4,8]。许多研究通过微调预训练模型来适应特定任务。例如，个性化方法通过微调 T2I 模型，使其能够生成特定对象或风格的图像 [1]。其他方法则通过添加新的条件信号来实现图像编辑或更可控的生成 [3,7]。类似的方法也被应用于 3D 生成、视频生成等专门任务 [6]。本文的方法展示了如何通过微调预训练的 T2I 修复扩散模型，实现参考图像驱动的图像修复。

2.2 图像修复

图像修复是计算机视觉中的一个长期挑战，旨在用合理的内容填充图像的缺失部分。传统方法依赖于手工设计的启发式算法 [2]，而最近的深度学习方法则直接训练端到端的神经网络来完成图像修复 [5]。尽管基于生成模型的图像修复方法取得了显著进展 [8]，但它们通常依赖于文本提示，难以生成与真实场景一致的内容。本文提出的 RealFill 方法通过引入参考图像，解决了这一问题。

2.3 基于参考图像的图像修复

现有的基于参考图像的图像修复方法通常依赖于复杂的几何变换和图像融合管道 [11]，容易在复杂场景中出现错误累积。Paint-by-Example 方法提出了一个新颖的潜在扩散模型 [10] 其生成由参考图像和目标图像共同条件化。然而，其条件化基于单个参考图像的 CLIP 嵌入，因此只能捕捉参考对象的高层语义信息。相比之下，RealFill 首次实现了多参考图像驱动的图像修复，能够在参考图像与目标图像存在显著差异的情况下，生成忠实于原始场景的内容。

3 本文方法

3.1 本文方法概述

本文的目标是通过一组参考图像（最多五张）来补全目标图像中的缺失区域，生成的内容不仅要合理且逼真，还要与参考图像中的实际场景保持一致。与传统的图像补全方法不同，本文强调生成的内容应该是“应该在那里”的内容，而不是“可能在那里”的内容。为了增加问题的挑战性，本文允许参考图像和目标图像之间存在显著差异，例如不同的视角、光照条件、风格，甚至场景中的动态变化。RealFill 的流水线详见图 1 输入是一张需要补全的目标图像和几张同一场景的参考图像。本文首先在参考图像和目标图像上微调预训练的修复扩散模型的 LoRA 权重（在训练过程中随机掩码部分图像）。然后，本文使用微调后的模型来补全目标图像中的目标区域，生成高质量且与参考图像一致的结果。例如，尽管参考图像 1 中女孩的姿势不同，但目标图像中女孩的皇冠被成功恢复。

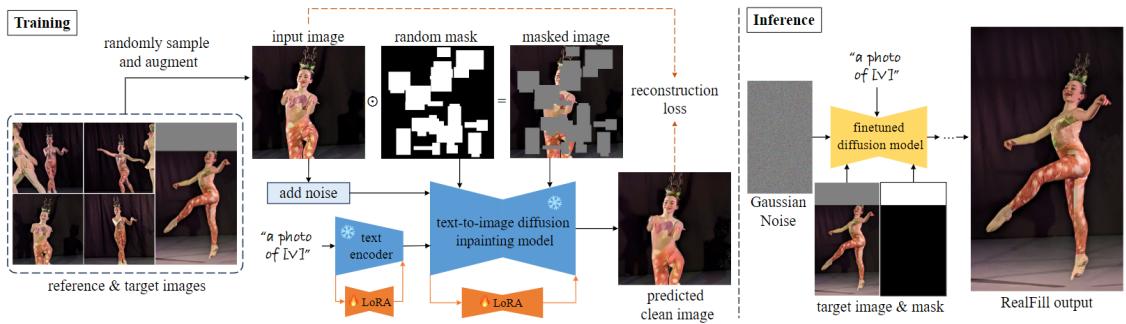


图 1. 方法示意图

3.2 特征提取模块

本文的特征提取功能主要通过扩散模型、LoRA 微调、LoFTR 以及 CLIP 和 DINO 等技术的结合隐式实现。首先，扩散模型（如 Stable Diffusion）在训练过程中通过逐步添加噪声并预测噪声，学习到了输入图像的深层特征表示，能够从参考图像和目标图像中提取场景的内容、光照和风格等特征。接着，本文使用 LoRA (Low-Rank Adaptation) 技术对预训练的扩散模型进行微调，通过向模型的权重矩阵注入低秩矩阵，使模型能够在不改变原始参数的情况下，高效地学习参考图像和目标图像中的特定场景特征，并将这些特征注入到目标图像的补全过程中。在推理阶段，本文使用 LoFTR (Detector-Free Local Feature Matching with Transformers) 提取生成图像与参考图像之间的局部特征对应点，通过计算对应点数量来评估生成结果的质量，并筛选出与参考图像最匹配的输出。此外，本文还利用 CLIP 和 DINO 等预训练模型提取图像的高层语义特征，用于评估生成图像与参考图像在语义上的一致性。通过这些技术的结合，本文实现了高效且高质量的特征提取与场景补全。

3.3 损失函数定义

本文的损失函数是基于扩散模型的噪声预测损失，其形式为：

$$\mathcal{L} = \mathbb{E}_{x,t,\epsilon,m} \|\epsilon_\theta(x_t, t, p, m, (1 - m) \odot x) - \epsilon\|_2^2$$

该损失函数的核心目标是通过最小化预测噪声与真实噪声之间的均方误差 (MSE)，使模型能够从带噪声的图像中恢复干净图像。具体来说，输入变量包括参考图像或目标图像 x 、时间步 t 、高斯噪声 ϵ 、随机生成的二值掩码 m 以及固定的语言提示 p 。掩码 m 用于指示图像中需要修复的区域，而 $(1 - m) \odot x$ 表示掩码后的干净图像，用于保留目标图像的现有内容。模型 ϵ_θ 通过预测噪声并计算其与真实噪声的误差来优化参数。对于目标图像 I_{tgt} ，损失仅在现有区域（即 $M_{tgt} = 0$ 的区域）计算，以确保生成结果与目标图像一致。通过结合随机掩码和条件输入，损失函数使模型能够在修复过程中保留目标图像的现有内容，并生成与参考图像一致的高质量补全结果。

4 复现细节

4.1 开源代码分析与对比

由于原始论文没有开源代码，在本次实验中，我基于一个非官方实现的代码进行了部署、训练、微调以及推理，原始代码已放入附件。其主要基于 hugging face 推出的 diffuser 框架实现，其网络架构等使用 diffuser 提供的已有网络框架搭建，并使用对应的 LoRA 实现了模型的微调，主要功能封装在 train.py 以及 infer.py 中，分别实现了训练以及推理。

训练部分主要包括初始化、网络模块载入与搭建、优化微调等数个模块，由于使用的是 diffuser 框架，所以可以便捷的进行网络模块的更换与编辑。在 train.py 的训练循环中，首先，模型被设置为训练模式，随后在每个训练周期中遍历数据批次。对于每个批次，输入图像和条件图像通过变分自编码器 (VAE) 编码到潜在空间，掩码和权重则被下采样以匹配潜在空间尺寸。接着，噪声被添加到潜在向量中，模拟扩散过程的前向噪声添加，并将噪声潜在向量、掩码和条件图像拼接为 UNet 的输入。文本提示通过文本编码器转换为语义向量，作为条

件输入引导 UNet 预测噪声残差。损失函数通过计算预测噪声与真实噪声之间的加权均方误差 (MSE) 来优化模型参数，并通过反向传播、梯度裁剪和参数更新完成训练。训练过程中，进度条和日志记录会定期更新，同时模型检查点会被保存，并在验证集上评估性能。如果达到最大训练步数，训练将提前终止。整个过程通过梯度累积和分布式训练技术高效地进行。

在 infer.py 的推理过程中，首先加载预训练模型、验证图像和掩码图像。接着对掩码图像进行腐蚀和模糊处理以优化修复效果。然后，使用文本提示、输入图像和掩码图像进行 16 次推理，生成修复后的图像，并将结果与原始图像合成后保存到指定目录。最后，清理模型并释放 GPU 显存。整个过程支持通过随机种子实现可重复的推理。

开源使用的由 StabilityAI 提供的 stable-diffusion-2-inpainting 模型。在部署以及代码修改中，我尝试修改了模型使用的主体框架，将其替换成了数个不同的模型框架（如 FLUX.1-dev ControlNet Inpainting - Beta, FLUX.1-dev ControlNet Inpainting - Alpha, stable-diffusion-inpainting 等），并分别修改测试了数个不同的训练参数（如 epoch 数目、学习率等），经过测试，发现模型修改后效果与 stable-diffusion-2-inpainting 模型类似，并无太多差异，甚至会降低一部分的补全效果。

4.2 实验环境搭建

实验使用 NVIDIA Quadro P4000 (显存 $\geq 8\text{GB}$)，支持 CUDA 11.8，操作系统为 Windows 10，Python 版本为 3.8。为避免依赖冲突，我使用了 Anaconda 虚拟环境来隔离实验环境。首先安装 PyTorch 2.0.1 及对应的 CUDA 支持，并通过简单的 Python 代码验证 PyTorch 和 CUDA 是否正确安装，确保 GPU 可用。接着，安装实验所需的核心 Python 库，主要包括 diffusers、transformers、peft 等，这些库提供了模型加载、训练、推理以及日志记录等功能。为使用 Hugging Face 的预训练模型，登录 Hugging Face Hub 并下载所需的 Stable Diffusion 模型，需要确保具有 Hugging Face 的访问权限并正确配置模型路径。通过一个简单的测试脚本验证环境是否正常工作，该脚本加载 Stable Diffusion 模型，生成一张示例图像并保存。如果图像成功生成，则表明环境搭建成功。通过以上步骤，即可成功搭建了一个支持 RealFill 的图像修复的实验环境。

4.3 创新点

RealFill 的主要创新点在于其基于参考图像的个性化生成模型和真实图像补全任务的定义。通过微调预训练的扩散模型，并结合基于对应关系的种子选择机制，RealFill 能够在复杂的场景中生成与真实场景高度一致的图像补全结果。此外，论文为了进行实验评估而提出的 RealBench 数据集为这一领域的研究提供了新的评估基准。

5 实验结果分析

复现工作可以较好的完成对应的任务，即基于三张图像作为训练数据，并进行对应的图像补全。训练数据见图像 2，结果图像见图像 3 与图像 4。



图 2. 训练数据集



图 3. 实验结果：左图为输入的图像；中下为 mask，其中白色部分表示被遮住的部分；中上为经过 mask 的输入数据；右图为补全结果；



图 4. 实验结果：左图为输入的图像；中下为 mask，其中白色部分表示被遮住的部分；中上为经过 mask 的输入数据；右图为补全结果；

可以看到，补全的结果是比较符合原始图像的语义以及内容的，同时补全的区域不管对于是四周还是中央都可以有一个比较优秀的结果。除此之外，RealFill 在训练过程中还会对图像的语义信息、结构信息以及颜色分布信息有一定的理解，具体可见图像5。其中，输入的图像不在训练数据集中，但是可以看到在补全图像中，上方图像的四周颜色属性能够较好的融入补全的图像，可以一定程度上说明 RealFill 对于边缘之间的颜色过渡有一定的理解，并能够基于这个进行对应的补全；下方图像任务的面部与手部能够处在一个较为正常的人类姿态的位置上，并且可以发现补全的部分并没有出现其他的手部，说明 RealFill 同样对于人类的姿态与人体结构有着一定的理解，并能够基于这个进行对应的补全。

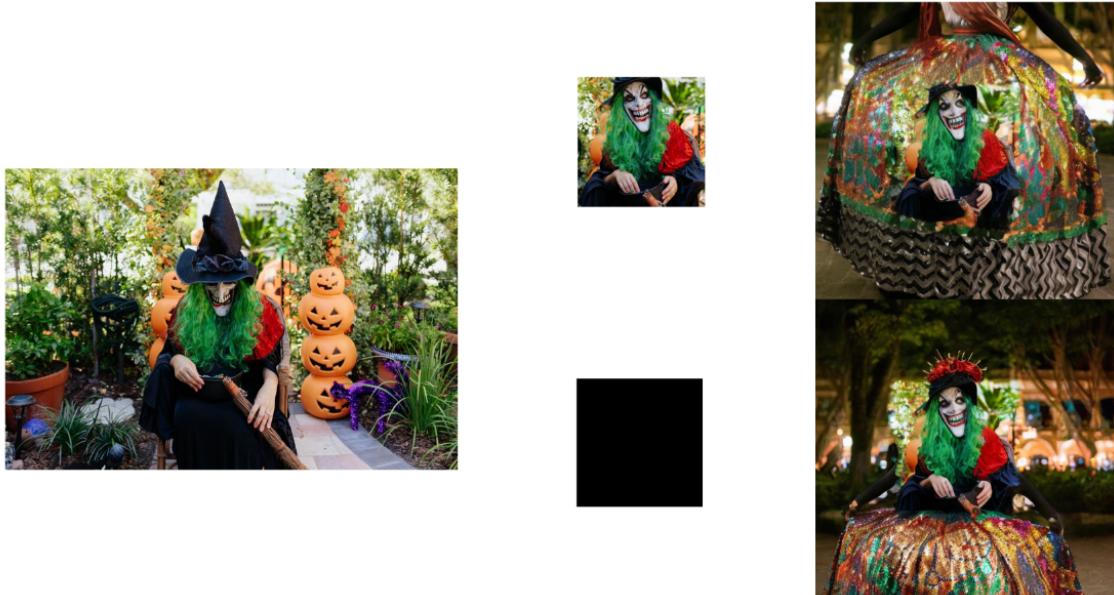


图 5. 实验结果：左图为输入的图像；中下为 mask，其中白色部分表示被遮住的部分；中上为经过 mask 的输入数据；右图为补全结果；

6 总结与展望

对于 RealFill 来说，其优势在于能够使用一个较小的数据集进行微调，就能够实现一个较为不错的图像补全效果，这对于图像修复、图像移除等工作提供了可用的工具与贡献；此外 RealFill 能够处理参考图像和目标图像之间的大幅差异，例如视角变化、光照变化、相机光圈变化、图像风格变化甚至动态物体的变化。这使得它在处理复杂场景时表现出色。但 RealFill 的微调过程较为漫长，使用上文实验环境下的设备进行实验时，三张图像数据的情况下，微调将会消耗大约两个小时，当需要处理大量图像时，其微调开销可能成为瓶颈。另外，其最多只能够接收 5 张以下的图像作为训练数据，但需要重建补全一个更加复杂，数据需求更大的场景时，其可能无法胜任工作。所以未来可能可以从效率以及数据处理的角度进行改进，同时可以深入探究 RealFill 的结构理解与色彩信息理解这两个原始论文中未曾提及的特性。

参考文献

- [1] Omri Avrahami, Kfir Aberman, Ohad Fried, Daniel Cohen-Or, and Dani Lischinski. Break-a-scene: Extracting multiple concepts from a single image. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–12, 2023.
- [2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch. *ACM Transactions on Graphics*, page 1–11, Jul 2009.
- [3] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. Nov 2022.
- [4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [5] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics*, page 1–14, Aug 2017.
- [6] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. Mar 2023.
- [7] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. Feb 2023.
- [8] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.
- [9] Luming Tang, Nataniel Ruiz, Qinghao Chu, Yuanzhen Li, Aleksander Holynski, David E. Jacobs, Bharath Hariharan, Yael Pritch, Neal Wadhwa, Kfir Aberman, and Michael Rubinstein. Realfill: Reference-driven generation for authentic image completion. Sep 2023.
- [10] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. Nov 2022.
- [11] Yunhan Zhao, Connelly Barnes, Yuqian Zhou, Eli Shechtman, Sohrab Amirghodsi, and Charless Fowlkes. Geofill: Reference-based image inpainting with better geometric understanding. Jan 2022.