

风格迁移算法在 AI 绘画设计中的应用研究

李艳路¹, 孙玲玲²

(1. 安徽扬子职业技术学院 艺术学院, 安徽 芜湖 241000;

2. 安徽工程大学 设计学院, 安徽 芜湖 241000)

摘要: 为提高绘画后图像的信噪比和结构相似度, 利用人工智能设计了一种风格迁移算法, 并将其应用于 AI 绘画设计工作中. 该算法首先利用图像处理功能将连续图像序列逐帧分解, 并对单帧图像进行分类整理, 以便于后续处理图像; 然后, 通过特征映射层将图像特征映射到目标风格域, 并在特征重建层中逐步恢复图像的高层特征, 以实现风格的迁移; 最后, 对图像进行特征一致性处理后, 构建生成对抗网络, 并通过引入风格损失函数和内容损失函数优化输出, 由此得到最终的 AI 绘画结果. 对比实验表明, 该方法生成的 AI 图像的峰值信噪比为 73.18 dB, 与原始图像的结构相似度大于 90%, 且绘画效果显著优于传统方法; 因此, 该方法在 AI 绘画设计中具有良好的应用前景.

关键词: 风格迁移算法; AI 绘画; AI 图像; 原始图像

中图分类号: TP399

文献标志码: A

DOI: 10.16379/j.cnki.issn.1004-4353.2025.03.018

Research on the application of style transfer algorithm in AI painting design

LI Yanlu¹, SUN Lingling²

(1. School of Arts, Anhui Yangtze Vocational and Technical College, Wuhu 241000, China;

2. School of Design, Anhui University of Engineering, Wuhu 241000, China)

Abstract: In order to improve the signal-to-noise ratio and structural similarity of the painted image, a style transfer algorithm was designed by artificial intelligence and applied to the AI painting design work. Firstly, in the MATLAB environment, the image processing function is used to decompose the continuous image sequence frame by frame, and the single-frame images are classified and sorted out to facilitate the subsequent processing of the images. Then, the image features were mapped to the target style domain through the feature mapping layer, and the high-level features of the image were gradually restored in the feature reconstruction layer to realize the style transfer. Finally, after the feature consistency processing of the image, a generative adversarial network is constructed, and the output is optimized by introducing the style loss function and the content loss function, thereby to obtain the final AI painting result. Comparative experiments show that the average peak signal-to-noise ratio of the AI image generated by this method reaches 73.18 dB, and the structural similarity with the original image is more than 90%, which is significantly better than that of the traditional method. Therefore, this method exhibit a good application prospect in AI painting design.

Keywords: style transfer algorithm; AI painting; AI images; original image

收稿日期: 2025-03-07

基金项目: 安徽省社会科学重大课题(2022AH040360).

作者简介: 李艳路, 讲师, 硕士, 研究方向为 AI 绘画、艺术学.

0 引言

随着人工智能(artificial intelligence, AI)技术的飞速发展, AI 绘画技术逐步成为艺术创作与设计领域的重要手段。为了进一步提升 AI 绘画的质量与效果, 挖掘其更多的应用潜力, 目前已有学者对 AI 绘画技术进行了深入研究。例如: 张泽宇等^[1]对图像生成模型进行了分类总结, 详细介绍并讨论了各代表模型的原理、结构与应用优缺点, 并对 AI 绘画面临的问题进行了深入讨论; 马诗洁等^[2]提出了一种基于大模型微调范式的绘画风格模拟方法; 王文凤等^[3]提出了一种基于智能技术的交互式绘画设计方法; 吴秀敏等^[4]提出了一种基于图像纹理渲染的智能绘画技术。然而, 在实际应用场景中, AI 绘画技术仍面临诸多挑战, 尤其是在处理复杂背景和模拟光影效果方面存在明显的技术瓶颈, 进而使得生成的图像质量难以满足实际需求。为此, 本文利用人工智能设计了一种风格迁移算法, 并通过实际应用验证了该方法的有效性。

1 基于风格迁移算法的 AI 绘画设计

1.1 原始图像的预处理

为了保证 AI 绘画的设计质量以及风格迁移算法的计算精度, 本文对待迁移的图像进行了预处理。预处理的方法为: 首先, 利用 MATLAB 的图像处理功能将连续的图像序列逐帧分解, 以此确保每一幅图像都能作为独立单元, 以便于后续的精确定义; 然后, 根据图像特征的变化对单帧图像进行分类整理, 并以 JPG 格式进行存储, 这样既能优化存储效率, 又能便于后续快速操作与访问。

考虑到原始图像通常为彩色图像, 且是以三维数组的形式存在, 因此可将其表示为:

$$I = R \times G \times B. \quad (1)$$

式中, I 表示待迁移的图像; R 表示图像的红色分量; G 表示图像的绿色分量; B 表示图像的蓝色分量。为了便于后续处理和对图像亮度信息进行分析, 本文将彩色图像转换为灰度图像, 即将每个像素点仅用一个亮度值表示, 而不再区分颜色通道。

在彩色图像向灰度图像转变的过程中, 考虑到人类视觉系统对不同色彩具有不同的敏感度, 本文对红色(G)、绿色(R)、蓝色(B) 3 个颜色通道执行了精细的加权平均处理, 以此来模拟人眼感知灰度的方式^[5]。该过程中, 绿色通道对亮度的贡献最大, 红色次之, 蓝色最小, 其用公式可表示为:

$$F = \omega_R R + \omega_G G + \omega_B B. \quad (2)$$

式中, F 表示灰度化后的待迁移图像, ω_R 、 ω_G 、 ω_B 分别表示待迁移图像中红色、绿色和蓝色分量的权重系数。利用公式(2)即可将一幅彩色的 RGB 图像转换为一幅仅包含亮度信息的灰度图像。

1.2 基于风格迁移算法的 AI 图像生成

基于上述得到的包含亮度信息的灰度图像, 本文利用风格迁移算法将图像的风格迁移到待生成的图像中, 以此得到最终的 AI 绘画结果。具体流程如下:

1) 数据增强与预处理。引入一个多样化的训练样本数据集, 并随机对其角度进行旋转和不规则裁剪, 同时动态调整图像的亮度与对比度。上述操作旨在模拟真实图像在自然环境中可能遭遇的各种变化, 从而增强风格迁移算法对复杂场景的适应能力和泛化性能。

2) 特征提取与重建。特征提取时, 首先将预处理的数据集统一调整为标准化尺寸(256 像素×256 像素), 并对其进行特征提取, 以此从输入图像中提取出高层次的抽象特征。然后, 将这些特征通过特征映射层映射到目标风格域, 从而使输入图像能够适应不同的艺术风格或视觉效果。在特征重建层中, 利用解码器结构逐步恢复图像的高层特征, 以此生成与目标风格高度一致的输出图像。对于任意的输入图像, 其经过特征映射处理后可得到包含丰富信息的中间特征图, 然后再对这些中间特征图进行重建即可

得到输出图像,其用公式可表示为:

$$X = \sum_{i=1} s_F \cdot F^{\eta\rho}. \quad (3)$$

式中, X 表示重建得到的AI图像, i 表示图像特征向量数量, s_F 表示待迁移图像特征值, η 表示输入的预训练样本数量, ρ 表示迁移系数。

3) 特征一致性处理. 为了确保生成的图像在内容上与原始图像保持一致,采用预训练的视觉几何组(visual geometry group, VGG)网络来分别提取原始图像与生成图像的深层特征,并计算二者特征之间的损失值. 通过最小化该损失函数,可使生成AI图像的特征尽可能地与原始图像相似,其用公式可表示为:

$$f = \min |X - F|. \quad (4)$$

式中, f 表示损失函数。

4) 风格迁移与AI绘画生成. 图像经特征一致性处理后,利用导向滤波技术去除纹理细节,并保留图像的主要结构和边缘信息,以此得到平滑的图像. 导向滤波技术是一种边缘保持的滤波方法,它能够在平滑图像的同时,有效地保留图像中的重要边缘和结构特征. 图像经过导向滤波处理后,会变得更加平滑,进而为后续的风格迁移提供良好的基础。

为了使生成的图像与原始图像在特征分布上尽可能接近,本文构建一个生成对抗网络. 生成对抗网络由生成器和判别器2个部分组成:

① 生成器采用编码器-解码器结构,它能够从输入的图像中提取特征,并基于这些特征生成新的图像. 因此,在风格迁移的过程中,生成器主要负责从预处理后的图像中提取特征,并将这些特征映射到目标风格域中. 在特征重建层中,利用解码器结构逐步恢复图像的高层特征,最终生成具有目标风格的图像。

② 判别器中含有多个PatchGAN结果,用于判断输入图像是真实图像还是由生成器生成的假图像. PatchGAN是一种基于图像块的判别器结构,它能够捕捉图像中的局部特征,并判断这些特征是否真实. 考虑到本文的数据集规模较小(数据量和多样性不足),难以支撑GAN模型生成器与判别器充分学习,进而使得重新训练的GAN模型难以达到理想效果,且计算资源消耗较大;因此,本文选择在预训练好的GAN模型基础上进行微调,使模型在当前有限的数据集条件下,通过微调即可适应本文特定的AI绘画风格迁移任务,且在能够保证生成图像质量的同时,尽可能减少计算资源的浪费. 在调整训练生成对抗网络时,生成器和判别器通过相互竞争来不断优化,即生成器试图“欺骗”判别器,使其将生成的图像误认为是真实图像,而判别器则“努力”提高自己的辨别能力,以更准确地区分真实图像和生成的图像. 在每一步的调整中,首先固定生成器,并训练判别器以更好地区分真实图像和生成图像;然后固定判别器,并训练生成器以生成更逼真的图像,用以“欺骗”判别器. 调整训练的目标是找到一个平衡点,使得生成器能够生成足以“欺骗”判别器的图像. 当达到上述平衡状态时,即表明风格迁移算法成功实现。

为了使生成的图像在风格迁移过程中既能呈现出目标风格,又能保留原始图像的关键内容信息,本文在风格迁移的过程中引入了风格损失和内容损失2个损失函数. 其中,风格损失用于衡量生成图像与目标风格图像之间的风格差异,内容损失用于衡量生成图像与原始图像之间的内容差异. 通过最小化这2个损失函数的加权和,既能约束生成的图像在风格上接近目标风格,又能在内容上保持与原始图像的一致. 生成图像的风格损失和内容损失可以表示为:

$$S_{Loss} = \|G(I_g) - G(I_m)\|_2^2. \quad (5)$$

式中: S_{Loss} 表示风格损失; $G(\cdot)$ 表示计算格拉姆矩阵的函数,用于捕捉图像的风格特征; I_g 表示生成器生成的图像; I_m 表示目标风格图像; $V(\cdot)$ 表示图像特征函数。

基于上述,风格迁移算法的输出可以表示为:

$$L = \arg \min_{I_g} (\omega_s S_{\text{Loss}} + \omega_c C_{\text{Loss}}). \quad (6)$$

式中, L 表示最终算法输出的迁移生成的 AI 绘画; ω_s 和 ω_c 分别表示风格损失和内容损失的权重系数, 用于平衡两者之间的重要性。

在此基础上, 为了进一步削弱颜色和亮度信息对生成图像的影响, 本文利用随机数生成器随机调整 L 的颜色通道比例, 以此生成具有随机颜色强度的 AI 绘画图像。

2 实验与分析

2.1 实验数据集及实验环境

为验证本文提出的基于风格迁移算法的 AI 绘画设计方法的有效性, 将其与文献[3]中的基于智能技术的交互式绘画设计方法(对比方法 1)和文献[4]中的基于图像纹理渲染的智能绘画效果生成技术(对比方法 2)进行了对比。实验数据集包含 FHAAFGL、IYFASGHH 2 个图像数据集, 其中 FHAAFGL 数据集包含 10 000 张人物图像, IYFASGHH 数据集包含 10 000 张风景图像。实验环境: 操作系统为 Windows HN2.62, 处理器为 Inter Coer i8, 硬盘容量为 8 GB, 服务器的内存为 64 GB, AI 绘画设计程序为 6.2.3 版本的 Python。

2.2 实验参数及指标

实验中, 风格迁移算法的参数设置如下: 学习率为 0.001, 训练批次大小为 1, 最大迭代次数为 100, 斜率为 1.53, 损失函数为 Gram 矩阵, 风格权重为 0.05, 图像大小为 4 像素 \times 4 像素。实验共设计可爱、潮流、少女等 10 余种人物风格的 AI 图像, 以及森林、草原、大海等 10 余种自然风格的 AI 图像。

性能对比指标选择峰值信噪比和结构相似度^[12]。其中: 峰值信噪比值越高, 表示图像质量越好; 噪声越少, 表示图像清晰度越高; 结构相似度值越高, 则表示 AI 绘画设计精度越高。

2.3 实验结果与讨论

图 1 中, 第 1 排为 3 幅不同的风格、主题和场景的原始图像(取自 Flickr 创意共享图像库), 第 2 排为应用本文方法绘制的 3 幅图像。由绘制效果可以看出: 尽管图像风格发生了变化, 但所绘制的图像在内容上仍然保持了原始图像的基本特征, 且图像质量较好, 没有出现明显的模糊、失真或噪声等现象。这表明本文方法在处理图像时能够保持较高的精度和稳定性, 方法可行。



图 1 本文方法的绘图效果

利用 3 种方法生成 AI 图像的峰值信噪比如表 1 所示, 实验数据集为 FHAAFGL 数据集。利用 3 种方法生成 AI 图像的结构相似度如表 2 所示, 实验数据集为 IYFASGHH 数据集。因篇幅限制, 所以本文

仅随机抽取了 8 个样本. 由表 1 和表 2 中的数据可以看出,本文方法生成的 AI 图像的峰值信噪比(平均值为 73.18 dB)和结构相似度(均大于 90%)远高于其他 2 种方法,由此表明本文方法生成的 AI 图像质量相对更好,可以实现高质量的 AI 绘画设计.

表 1 利用 3 种方法进行 AI 绘画的峰值信噪比				dB	
图像编号和统计指标	本文方法	对比方法 1	对比方法 2		
YH-178	71.26	55.24	62.62		
YH-256	72.42	53.26	61.24		
YH-379	70.36	54.25	62.63		
YH-1084	74.15	55.62	62.58		
YH-2791	72.86	54.56	61.35		
YH-5632	73.61	55.96	61.42		
YH-7852	71.58	55.74	62.53		
YH-9811	72.53	55.25	62.25		
平均值	73.18	53.92	61.47		
标准差	1.00	2.10	1.60		

表 2 利用 3 种方法进行 AI 绘画的结构相似度				%	
图像编号和统计指标	本文方法	对比方法 1	对比方法 2		
YH-178	99.86	75.15	81.26		
YH-256	98.75	74.35	82.63		
YH-379	98.36	72.56	81.42		
YH-1084	94.56	72.95	82.25		
YH-2791	96.35	73.14	81.46		
YH-5632	98.75	72.56	81.56		
YH-7852	98.15	73.66	84.36		
YH-9811	96.36	73.14	82.52		
平均值	98.31	74.26	81.49		
标准差	0.01	0.03	0.02		

3 结论

对本文提出的风格迁移算法进行研究显示,该算法生成的 AI 图像峰值信噪比平均值为 73.18 dB,生成的图像与原始图像的结构相似度超过 90%,其绘画效果显著优于基于智能技术的交互式绘画设计方法和基于图像纹理渲染的智能绘画效果. 这表明,本文提出的风格迁移算法可有效提升 AI 绘画图像质量,具有良好的应用潜力. 但本文算法在处理复杂图像时,其性能表现一般,难以对复杂的场景、纹理和细节进行精准处理,导致生成的图像可能无法达到预期效果. 同时,算法的运算效率有待提高,在处理大规模图像数据或高分辨率图像时,可能会耗费较长时间,影响实际应用体验. 为此,在今后的研究中,我们将通过优化算法和提高风格迁移的准确性、稳定性来提高本文算法的计算效率和增强算法对不同风格和内容图像的适应性.

参考文献:

[1] 张泽宇,王铁君,郭晓然,等. AI 绘画研究综述[J]. 计算机科学与探索, 2024,18(6):1404-1420.

[2] 马诗洁,徐华艺,李聪聪,等. 基于大模型微调范式的绘画风格模拟方法[J]. 计算机应用, 2024,44(S1):268-272.

[3] 王文凤,张荣. 智能技术在交互式绘画设计中的应用及其创作机制研究[J]. 包装工程, 2022,43(S1):89-95.

[4] 吴秀敏,崔巍. 基于图像纹理渲染的智能绘画效果生成算法设计[J]. 电子设计工程, 2023,31(12):73-77.

[5] 杨盼,张敏情,葛虞,等. 基于风格迁移过程的彩色图像信息隐藏算法[J]. 计算机应用, 2023,43(6):1730-1735.