

基于图片的像素画生成系统项目报告

10194500123 成艾佳

摘要 近年来，随着游戏动漫行业的发展，人们对虚拟物体模型的精度要求愈来愈高，力求在虚拟世界中渲染绘制出与现实中相当的物品，但是对这些高质量素材的建模绘制往往需要消耗大量的时间与精力。而在传统的游戏行业有着像素游戏这样一类存在，通过少量的不同颜色的像素块便能表现出可以被理解与辨识的物体，同时也存在着独特的美感。本系统着眼于为尝试创造像素画却并不精通的用户提供简单的素材生成功能，为用户后续的完善与优化提供基础。本篇报告将说明该系统的主要功能以及核心算法的实现原理。

1. 系统应用场景介绍以及功能展示

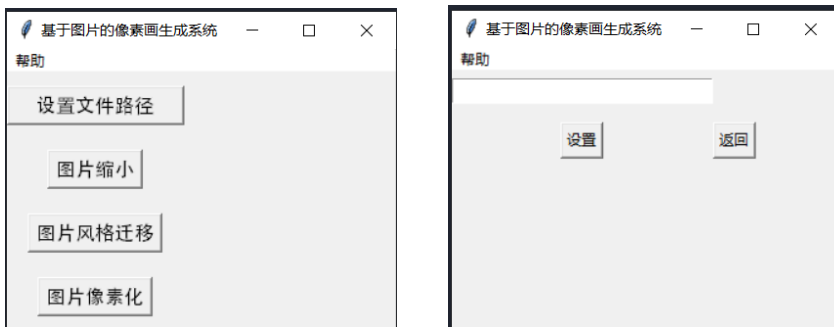
基于图片的像素画生成系统（以下简称为系统）致力于生成现实中的图片的**像素画**（将像素颗粒以肉眼可见的明显大小呈现在画布上，结合成一幅完整的图画）素材，从而作为艺术资源可以在日后使用——游戏布景等。

本系统功能主要分为三部分：像素画生成、图片缩小与图像风格迁移。其中**图片缩小**功能使用 Seam Carving 算法，当用户需要清除图像中不重要的留白部分而保留关键内容时调用，并且不会由于裁剪而使得图片有锯齿撕裂感，显得扭曲；同时**图片风格迁移**功能意在为用户基于同一张图片生成不同风格类型的图片，增加素材的丰富程度。

1.1 系统使用介绍

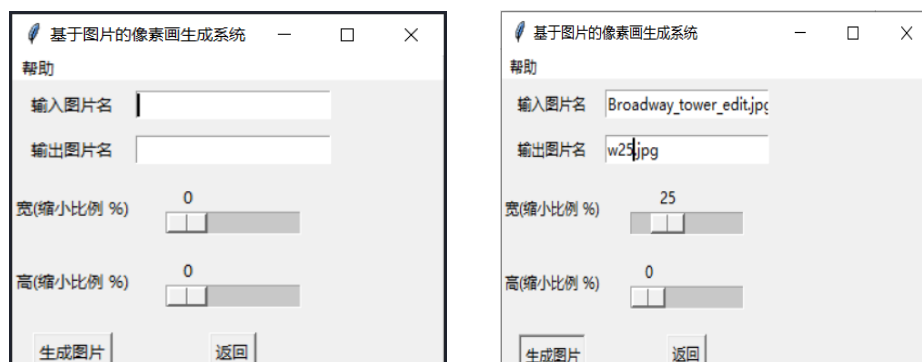
系统及各功能实现使用的计算机语言为 **Python**，用户使用本系统主要通过 Tkinter 编写的前端来进行，当然也可以直接运行各功能对应的.py 文件进行。

在运行 **SystemUI.py** 文件后便会弹出系统的前端界面，首先用户需要选择**设置文件路径**来决定接下来处理的图片所在的文件夹；然后用户便可以返回主界面选择对应的功能进行图像处理。



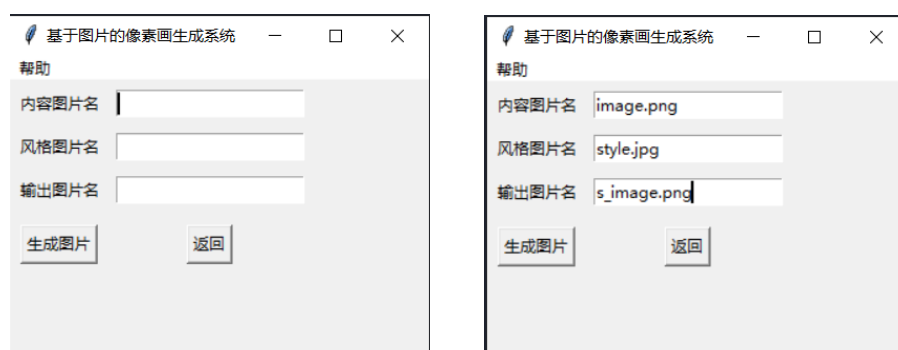
1.1.1 图片缩小

在选择**图片缩小**功能后，依次设置输入图片与输出图片的文件名，并通过滚动条设置图片的高度与宽度缩小的比率，随后点击**生成图片**便会在对应文件夹下生成对应的图片文件。



1.1.2 图片风格迁移

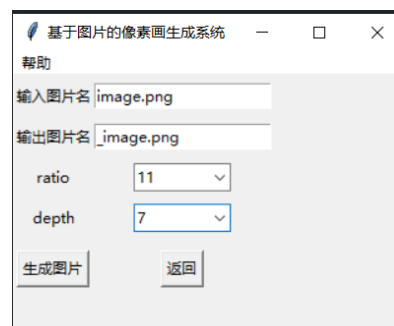
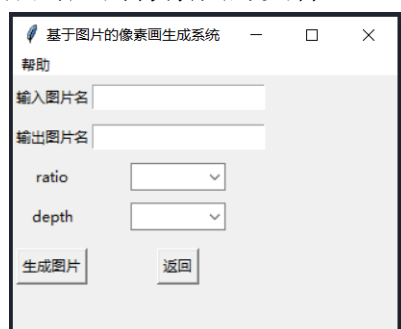
在选择**图片风格迁移**功能后，依次设置内容图片与风格图片以及输出图片的文件名，随后点击**生成图片**，便会在对应文件夹下生成对应的图片文件。



1.1.3 图片像素化

在选择**图片像素化**功能后，依次设置输入图片与输出图片的文件名，并选择 ratio（决定像素图片的精度）与 depth（决定像素图片的色彩丰富度），随后点击**生成图片**，便会在对应文件夹下生

成对应的像素图片文件。



2. 核心算法原理

2.1 图片缩小（代码见 seam.py）

对于图片缩小功能,本系统并没有简单的选择使用 cv2 中的 `resize` 函数或者是裁剪图像,前者会使图像扭曲(没有等比例缩放)或是图像内容失真,而后者则有较大概率使图像产生撕裂感。Seam Carving 算法是 Shai Avidan 提出的基于内容感知的图像缩放方法,意在移除那些与周围环境相融并不会让人“注意”的像素,在缩减后不会使图像显得“奇怪”^[1]。

在该算法中,首先需要计算图片各像素的梯度模(通过 Sobel 算子)从而得到这张图片的能量图;

$$e_1(\mathbf{I}) = \left| \frac{\partial}{\partial x} \mathbf{I} \right| + \left| \frac{\partial}{\partial y} \mathbf{I} \right|$$

然后基于能量图通过动态规划计算得出这张图片能量最小的接缝 Seam, 也即一条从上到下或从左到右的一条八连接像素路径;

最后将这条接缝删去, 如此重复 n 次以完成缩减 n 个像素宽或高的目的。

效果展示:



左为原图分辨率 411*279



右图宽度缩减 25% 为 309*279

2.2 图片风格迁移（代码见 neural_style.py）

A Neural Algorithm of Artistic Style^[2]这篇论文发现在卷积神经网络中图片的内容和风格的表达是可以分开来的。也就是说，通过独立地操作两种表达来产生新的图片——融合了两幅不同的原图片的内容和风格表现。

根据论文中的叙述，本系统使用 19 层的 VGG-Network 模型，根据论文中提供的算式分别让内容图片和风格图片与经过 VGG 模型预处理过的内容图片通过相同的卷积网络进行特征变换，并分别计算两者在每个层级特征变换后的差异，求和得出最后的总损失，并在迭代优化中求得最小的损失作为最优解。

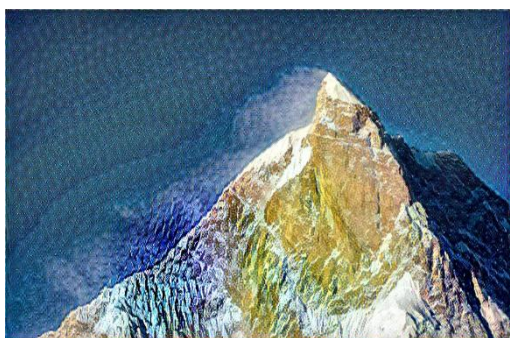
同时在计算总损失时，为内容损失与风格损失赋予不同的权重可以改变最后图像的效果，当内容损失与风格损失权重的比值越大，最终图片整体将更偏向内容图片；比值越小，则表示最终图片整体偏向风格图片。在本系统中，两者的比值为 10^{-1} ，使得图片的内容可以较好的保证。



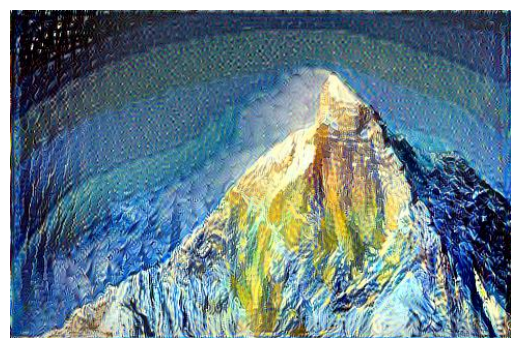
内容图



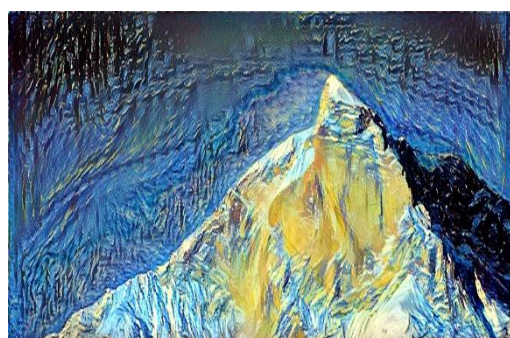
风格图



10^{-1}



10^{-2}



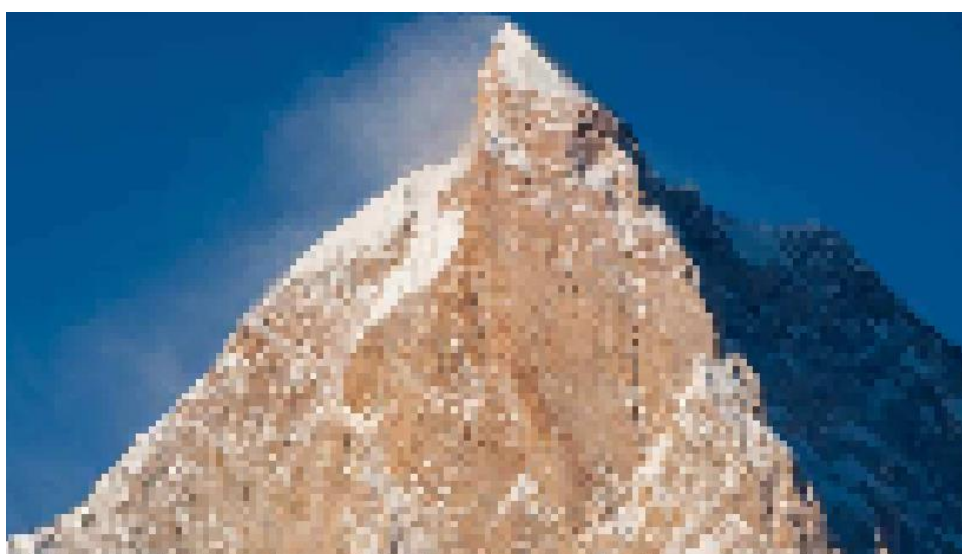
10^{-3}

相较于论文提供的算法，本系统并未过多修改，仅仅实现并作为一个功能写入整体。

2.3 图片像素化（代码见 `topixel.py`）

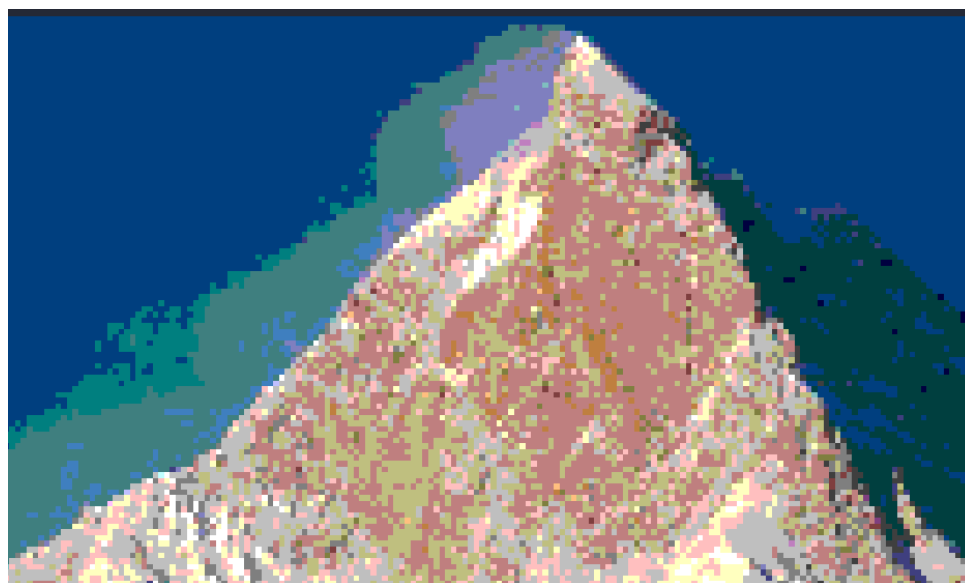
对于图片像素化的处理，简而言之就是降低图片的分辨率——对图片中的进行分块打包，通过计算每一个像素块中的各颜色像素出现的次数来决定这一像素块的新值，最后将所有的像素块依次排列形成图片。

对于颜色的处理，在完成像素块新颜色的选择之后，通过对生成图片的观察发现在保有像素图片与原图相同色彩选择时，由于原图分辨率较高而生成的像素图片往往需要直接将每一个像素都可辨识的，此时仍然保持原先的色彩变化将形成类似马赛克的效果，如下图：



上图每个通道 256 种取值， $ratio = 5$

然而，随着每个通道可取值的减少，再通过计算对应颜色间的绝对差值来选择新的像素值。



上图每个通道 5 种取值， $ratio = 5$

当然,可以看到的是随着可选颜色的减少,图像中的内容有所损失,为了得到满足用户的图片需要根据用户的主观的感受对参数进行调整。

3. 总结

本系统着重于生成像素画图片资源,同时将提供附加的图片处理能力来提升系统的功能性,然而,人们对于像素风图片或者说对于类似艺术的作品仍然有不同的看法,多数基于主观情感。由于作者并未受过专业的艺术类教育,所以对于系统所生成的像素画质量也许仅能得到少量认可,仍有较大的提升空间。

参考文献

- [1] Avidan, Shai; Shamir, Ariel (July 2007). "Seam carving for content-aware image resizing | ACM SIGGRAPH 2007 papers". Siggraph 2007: 10. doi:10.1145/1275808.1276390
- [2] Gatys, Leon A.; Ecker, Alexander S.; Bethge, Matthias (26 August 2015). "A Neural Algorithm of Artistic Style". arXiv:1508.06576

参考文档

- 1. <https://segmentfault.com/a/1190000009820053>