

武汉理工大学毕业设计（论文）

艺术风格迁移网站的设计与实现

学院（系）： 计算机科学与技术学院

专业班级： 计算机 1504 班

学生姓名： 李俊

指导教师： 程艳芬

学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包括任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名：

年 月 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保障、使用学位论文的规定，同意学校保留并向有关学位论文管理部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权省级优秀学士论文评选机构将本学位论文的全部或部分内容编入有关数据进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于 1、保密口，在 年解密后适用本授权书

2、不保密口。

（请在以上相应方框内打“√”）

作者签名： 年 月 日

导师签名： 年 月

摘 要

图像作为人类感知世界的基础，是人类获取信息、表达信息和传递信息的重要手段。从计算机出现开始，利用计算机进行数字图像处理成为了人们对图像的主要处理方法。而艺术风格，作为艺术创作中表现出来的一种综合性的总体特点，由于受到艺术家个人的世界观、价值观、文化教养、生活经历、审美趣味等因素的影响，难以用数学的方式表达。另外，传统的人工数字图像处理很难完成特定风格的提取和重建工作。近年来，由深度学习引领的人工智能技术浪潮开始越来越广泛地应用到社会的各个领域。这其中，人工智能与艺术的交叉碰撞，帮助人们找到了风格迁移和线稿自动上色的新途径。

基于深度学习的图像风格迁移算法可以完成以下工作：指定一幅输入图像作为内容图像，同时指定另一幅或多幅图像作为输出目标，算法在保证内容图像结构的同时，将图像风格进行转换，使得最终输出的合成图像呈现出输入内容和风格的完美结合。其中，图像的风格可以是某一艺术家的作品风格、也可以是由个人拍摄的图像所呈现出来的风格。另外，现如今漫画行业有着火爆的市场。作为图像风格迁移技术的衍生品，线稿自动上色技术可以帮助漫画爱好者和从业者大大减少漫画填色时间，提高经济效益。

市面上许多图像风格迁移应用不成熟，使用门槛较高，不能满足普通用户的图像处理需求。因此，做出一款简单易用、实时性高的艺术风格迁移在线网站对未来相关领域的研究和应用有着重要的指导意义和实用价值。

本文介绍了利用 Django 技术搭配 Apache 服务器开发艺术风格迁移网站的设计方案，采用 SQLite 数据库完成用户信息和图像文件的管理，结合实际需求，将艺术风格迁移分为“图像风格迁移”和“线稿自动上色”两个模块，并将它们部署至云服务器供用户在线使用。

关键词：卷积神经网络；图像风格迁移；线稿自动上色；Django

Abstract

Images, as a basic factor for human to perceive the world, are important media to receive, express and transmit information. Since the invention of modern computer, utilizing computer to conduct digital image processing has become the most popular method to manipulate images. Artistic styles, comprehensive general features shown in the creation of artworks, are influenced by factors like artists' personal world views, value views, cultural upbringing, life experiences, aesthetic taste etc. and as a result, too abstract to be expressed statistically. Moreover, traditional artificial digital image processing methods are unqualified for specific style extraction and reconstruction work. The overwhelming bloom of artificial intelligence led by deep learning is pushing its applications ubiquitous among various fields in current society. Specifically, the interaction of artificial intelligence and art has solved this image stylization problem.

Image style transfer algorithm based on deep learning can handle the following task: feeding one input image as content with one or several images as style, and the deep-learning-based algorithm will reconstruct syncretic style on the structure of content image to generate an image with content from one image with style from other images.

Additionally, the cartoon industry enjoys a hot market nowadays. As the derivative of image style transfer technology, sketch auto-colorization technology could help cartoon amateurs and practitioners save coloring time when drawing cartoons, which improves economic returns.

Many image stylization applications on the market are unmaturing and difficult to use, which cannot satisfy the needs of common people to manipulate images. Thus, developing a user-friendly and real-time art style transfer website has great guiding significance and practical value.

In this thesis, we introduce the design scheme using Django with Apache on Ubuntu to develop the artistic style transfer website. SQLite database was designed and deployed to manage user information and image files. Combined with actual requirements, the website was divided into 'image style transfer' and 'sketch auto-colorization' modules. And after offline development, the website was deployed on Aliyun to serve Internet users.

Key Words: Convolutional Neural Network; Image Style Transfer; Sketch Auto-colorization; Django

目 录

第 1 章 绪论	1
1.1 研究背景及意义	1
1.2 国内外研究现状	1
1.3 设计内容	2
1.4 论文组织结构	3
第 2 章 图像风格迁移关键技术	4
2.1 图像风格迁移方法分析	4
2.1.1 基于迭代优化的图像风格迁移方法分析	4
2.1.2 基于感知损失函数的实时图像风格迁移方法分析	5
2.2 图像风格迁移模块实现	7
2.2.1 运行环境配置	7
2.2.2 风格模型的训练	7
2.2.3 风格化图像的生成	8
2.3 本章小结	9
第 3 章 线稿自动上色关键技术	10
3.1 线稿自动上色方法分析	10
3.1.1 草稿绘制阶段	11
3.1.2 草稿细化阶段	11
3.2 线稿自动上色模块实现	11
3.2.1 运行环境配置	11
3.2.2 线稿上色模型的训练	12
3.2.3 结果图像生成	12
3.3 本章小结	13
第 4 章 网站设计与测试	14
4.1 网站设计	14
4.1.1 网站总体设计	14
4.1.2 网站详细设计	15
4.2 网站测试与运行结果	22
4.2.1 图像风格迁移模块测试及运行结果	22
4.2.2 线稿自动上色模块测试及运行结果	25
第 5 章 总结与展望	28
5.1 论文工作总结	28
5.1.1 已完成的主要工作	28

5.1.2 工作特点	28
5.2 未来展望	28
参考文献	30
致 谢	32

第 1 章 绪论

1.1 研究背景及意义

艺术风格是艺术家的创造个性与艺术作品的语言、情境交互作用所呈现出的相对稳定的整体性艺术特色，风格是艺术家创造个性成熟的标志，也是作品达到较高艺术水准的标志。例如，著名艺术家毕加索的画作中有着独具特色的立体主义风格，文艺复兴三杰之一的达·芬奇的绘画作品中体现出细腻的写实主义风格，抽象艺术先驱瓦西里·康定斯基以其独特的抽象风格闻名于世。然而，艺术家的作品数量有限，创作出的作品不会无限增加。因此，为了使艺术家的个人风格可以延续，将经典的艺术风格在现代作品中重现，对艺术作品风格的提取与再结合的研究有着重要意义。

上色是艺术创作中最重要但又耗时的步骤之一。创作富有表现力的画作需要良好的色彩构成以及纹理和阴影的准确使用，完成这个任务并非易事，因为它既需要美学感，又需要绘画经验。即便是专业人士也要花费大量的时间和精力来制作正确的色彩构图和精细的纹理阴影细节。因此，开发一个完善的自动或半自动的上色系统可以极大地便利这个行业的从业者。

深度学习（Deep Learning）的概念源于对人工神经网络的研究。深度学习通过组合低层特征形成更加抽象的高层特征来表示属性类别或特征，以发现数据的分布式特征表示。深度学习结构，例如深度置信网络、循环神经网络和卷积神经网络被广泛应用于计算机视觉、语音识别、自然语言处理、社交网络过滤和机器翻译等领域，能够得到媲美甚至超越人类专家的效果。

卷积神经网络（Convolutional Neural Network, CNN）是一类包含卷积计算且具有深度结构的前馈神经网络，是神经网络的代表结构之一，在视觉图像分析领域的应用最为广泛。卷积神经网络的灵感来自于生物过程，因为神经元之间的连接模式类似于动物视觉皮层的组织。单个的皮质神经元只对视野中一个被称为“接受区”的受限区域的刺激做出反应；不同神经元的接受区部分重叠，覆盖整个视野。因此，使用卷积神经网络完成艺术风格提取和再结合的任务最为合适。

另外，现代互联网用户更倾向于使用浏览器端的轻量级在线服务，一方面减少了繁琐的客户端软件安装过程，另一方面也降低了本地终端的计算负荷。本设计旨在通过使用卷积神经网络算法，结合网路编程和服务器部署技术，搭建实现在线图像风格迁移和线稿自动上色功能的网站，为广大艺术爱好者提供将经典风格融入任意图像、自动为线稿上色的平台。

1.2 国内外研究现状

得益于 GPU 等计算芯片的增强，深度学习近些年有着迅猛发展。在其众多应用领域

中，图像风格迁移任务成为了专家学者们的研究热点之一，国内外许多研究人员都对此做出了重要贡献。

一些学者将神经网络运用于图像风格化，用卷积的方法提取图像在神经网络各个层的风格和内容信息。Gatys 等利用卷积神经网络提取输入图像在其不同层级的特征，对图像的内容和风格进行分离，使用梯度下降调整输入响应，经过多次迭代获得具有特定风格和内容的图像^[1]。Li 等提出用感知损失函数替代逐像素损失函数，通过预训练的 VGG 模型简化损失计算，把计算负载从生成阶段转移至训练阶段，直接生成输出图像，大大加快了训练速度，使得实时风格迁移成为可能^[2]。Ulyanov 等提出了一种特殊的前馈卷积神经网络，可以产生大量任意尺寸的相同纹理，该方法引入一个用于风格转移的生成网络，大大减少迭代的次数，缩短图片生成的时间^[3]。Li 等和 Ulyanov 等的研究都围绕将求解全局最优解问题转换成用前馈网络逼近最优解问题^{[4][5]}。Dumoulin 等在文献[3]的基础上提出了条件实例正则化的方法来混合多种不同的画风，有助于创造新的风格^[6]。针对风格迁移后颜色变化和空间与实际不符的问题，Gatys 等提出了控制感知因素的方法，把 RGB 图像转换成 YIQ 图像，只在 Y 上进行风格转换，保持表示颜色信息的 I 和 Q 不变，另外，通过加入 R 个引导通道，对图像的不同部分分别转换，防止诸如把天空变成墙壁的错误发生^[7]。Luan 等提出了照片风格转换，抑制了画面的局部扭曲，把任务放在颜色域的仿射转换上，让照片不会因为局部扭曲产生“画”的感觉^[8]。

另外，神经网络也可以用于线稿自动上色。Sangkloy 等开发的 Scribbler 学习在草图上生成不同的场景，尤其是卧室，并提供用户提示^[17]。Furusawa 等建议用给定的不同调色板为漫画着色^[18]。Liu 等开发的 Autopainter 引入对抗式损失以实现更好的线稿着色^[19]。

目前，市面上也存在提供图像风格迁移功能和线稿自动上色的软件，如 Prisma，利用神经网络和人工智能技术，为普通照片加入艺术效果，以及 PaintsChainer，可以实现在线线稿上色。然而，这些软件的使用收费较高，在一定程度上影响了用户的使用。

1.3 设计内容

本设计主要是利用卷积神经网络算法，结合网络编程和服务器部署技术，在服务器端训练风格模板供用户使用，实现在线实时图像风格迁移和线稿自动上色功能。另外，设计简洁精美的网站页面，建立完善的服务器后台，实现用户信息和风格模板管理。研究内容主要包括：

（1）核心风格转换神经网络模型的设计和实现

以文献[1][2][5]为基础，利用卷积神经网络，结合感知损失函数，并预先训练 VGG 模型以加速损失计算，设计出满足实时性要求的风格迁移网络。

（2）神经网络模型的测试和调参

为神经网络模型输入若干风格图像，测试其模型训练速度。再输入内容图像，测试它们的生成速度，若生成图像的效果和速度不佳（影响用户体验），则对模型进行参数调整和

再测试。

（3）网站设计与实现

基于 Bootstrap 框架快速开发简洁精美的网站,并与 Django 实现的服务器端协同工作,提供在线风格迁移平台。用 Cocos Creator 开发线稿自动上色在线程序,打包成独立的服务运行在服务器端。

（4）服务器端的数据库管理和运行维护

使用 SQLite 管理用户信息和模板信息,并对其定期维护,保证正常工作。

1.4 论文组织结构

本文论述了基于 Django 框架的艺术风格迁移网站、后台神经网络模型训练以及评估程序的设计方案,围绕所使用的关键技术和实现方法展开,论文在组织结构上分为以下五章:

第 1 章:绪论。介绍了图像风格迁移网站开发的目的意义,分析了图像风格迁移相关领域的发展状况,最后陈述了本文的主要设计内容和论文的组织结构。

第 2 章:图像风格迁移关键技术。参考近年来图像风格迁移领域的几篇著名论文,分别对基于迭代优化的图像风格迁移方法和基于感知损失函数的实时图像风格迁移方法进行分析,并对本设计涉及的神经网络模型结构和实现算法进行详尽的介绍。

第 3 章:线稿自动上色关键技术。分析 Lvmin Z 等^[20]在 2018 年提出的的两级线稿上色方法,并对本设计涉及的神经网络模型结构和实现算法进行详尽的介绍。

第 4 章:网站设计与测试。在总体设计中,从各功能之间的逻辑联系和数据流动展开设计,搭建了项目总体构架,并以核心功能为中心介绍了网站的设计思路。在详细设计中,展示了网站的用户登录界面、主界面和功能界面,详细的介绍了数据库的设计和模块的设计,结合设计思路与具体的实现方法进行叙述,突出强调艺术风格迁移网站的主要特色。另外,对网站进行模块化测试,通过真实数据完成对网站功能的说明,并给出相应的结果。

第 5 章:总结与展望。本章中对所做的工作从已完成的和可改进的两个方面进行总结,分析了网站的特色和不足之处,并提出了后续的优化展望

第 2 章 图像风格迁移关键技术

图像风格迁移是通过识别图像的艺术风格和语义内容，使图像的内容能与不同的风格结合起来，展现出不同的魅力。本章主要研究基于生成模型的图像风格迁移方法，首先分析基于迭代优化的图像风格迁移方法，阐述其主要思想，解释其内容表示、风格表示和风格迁移的数学原理。然后，针对这个方法在计算量上的局限性，介绍基于感知损失函数的实时图像风格迁移方法，分析其独特的网络结构。最后，从环境配置、模型训练和图像生成三个方面介绍本设计中图像风格迁移模块的实现过程。

2.1 图像风格迁移方法分析

2.1.1 基于迭代优化的图像风格迁移方法分析

基于迭代优化的图像风格迁移方法，源于 Gayts 等在 2016 年的论文《Image Style Transfer Using Convolutional Neural Networks》^[1]，是研究人员首次发现卷积神经网络不同层的响应可以表示输入图片的内容和风格信息。

文中使用卷积神经网络的低层响应描述图像的风格，用高层响应描述图像的内容，通过优化风格和内容的联合损失函数调整输入响应（这里输入响应使用白噪声），通过多次迭代优化，将白噪声调整为具有特定内容和风格的图像，使之匹配内容图像的内容和风格图像的风格。下面将从内容表示、风格表示和风格迁移三个方面对基于迭代优化的图像风格迁移方法进行分析。

2.1.1.1 内容表示

如果将图片输入卷积神经网络中，其高层的响应就是内容信息。在卷积神经网络中，各层定义的非线性滤波器组的复杂度随着网络层数的加深而不断增加。一个层 l 的响应可以存储在矩阵 $F^l \in R^{N_l \times M_l}$ 中， N_l 表示这个层有 N_l 个滤波器， M_l 表示特征关系图的高乘宽。 F_{ij}^l 表示层 l 的位置 j 上的第 i 个滤波器的激活值（使用 ReLU 激活函数）。

为了将一张图像的内容信息可视化，可以在一个带有白噪声的图像上执行梯度下降算法重建图像内容，使其可以匹配原始图像特征响应。这里，令 \vec{p} 和 \vec{x} 分别为原始图像和生成的图像， P^l 和 F^l 分别是它们在层 l 中的特征表示。那么，则两个特征表示之间的方差损失为（即内容损失函数）：

$$\mathcal{L}_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2 \quad (2.1)$$

2.1.1.2 风格表示

为了获得输入图像的风格表示，可以在任一层上构建特征空间，以获得纹理信息。特征空间由不同滤波器响应结果的相关关系组成，这些特征关系用 Gram 矩阵 $G^l \in R^{N_l \times N_l}$ 表

示，其中 G_{ij}^l 是层 l 的向量化特征关系图 i 和 j 之间的内积：

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l. \quad (2.2)$$

通过包含多个网络层的特征相关关系，我们可以得到一个固定的、多尺度的输入图像的表示，其可以捕获纹理信息，而不会含有内容信息。和内容重建相似，风格重建同样可以通过在白噪声图像上使用梯度下降算法最小化原始图像的Gram矩阵和生成图像的Gram矩阵的平均平方损失来实现。

令 \vec{a} 和 \vec{x} 分别表示原始图像和生成图像， A^l 和 G^l 分别表示层 l 的风格表示。层 l 相对于总损失的贡献为：

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2 \quad (2.3)$$

那么总的风格损失函数为：

$$\mathcal{L}_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^L w_l E_l \quad (2.4)$$

2.1.1.3 风格迁移

为了将画作 \vec{a} 的风格转移到照片 \vec{p} 上，可以用白噪声图像同步匹配 \vec{p} 的内容表示和 \vec{a} 的风格表示。因此，我们从卷积神经网络一层的图像内容表示和若干层上定义的风格表示得到的白噪声图像，联立求解其内容和风格的特征表示最小化距离。最小化的损失函数为：

$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x}) \quad (2.5)$$

其中 α 和 β 分别是内容重建和风格重建的权重因子。这里没有采用常用的梯度下降方法，而是使用对L-BFGS优化算法，这个可以非常好的应用于图像合成任务中。

这里的联合损失函数是内容损失函数和风格损失函数用权重相乘后的线性组合，当权重 α/β 较小时，损失更偏向包含更多的风格，因而会减少生成图片中的内容信息，更多表现画作 \vec{a} 的风格；反之，当权重 α/β 较大时，损失更偏向包含更多的内容，因而会减少生成图片中的风格信息，更多表现照片 \vec{p} 的内容。

2.1.2 基于感知损失函数的实时图像风格迁移方法分析

Gayts 等^[1]提出的基于迭代优化的图像风格迁移方法虽然能够生成高质量的风格化图像，但是需要漫长的迭代优化过程，无法做到实时风格迁移。针对这一点，Johnson 等^[2]提出了基于感知损失函数的实时风格迁移方法。该研究中指出，Mahendran 等^[22]在有监督模式下训练一个前馈卷积神经网络，用逐像素差作损失函数来衡量输出图像和输入图像的差距，只需要一次前馈通过已训练好的网络即可完成测试，速度极快，但是逐像素差的损失函数无法准确定义图像间感知上的差距。而 Gayts 等^[1]通过建立感知损失函数，可以很好

地获取输入输出图像的感知差距，但是生成图像时需要 CN 进行反向传播优化，参数众多，计算量巨大，无法做到实时风格迁移。

结合二者的优势，Johnson 等^[2]提出采用感知损失函数代替逐像素差损失函数训练前馈网络完成图像风格迁移的任务，这个方法可以产生与基于迭代优化方法质量相当的结果，并做到三个数量级的提速^[2]。该方法在网络结构上参考 Radford 等^[21]的 DCGAN 的设计思路，其中参考 Mahendran 等^[22]的前馈神经网络作为图像转换网络，损失网络采用与 Gayts 等^[1]类似的感知损失函数。在测试时，只需要将内容图像输入图像转换网络，一次前馈计算后输出就是具有特定风格的生成图像。以下从网络结构和感知损失函数两方面分析基于感知损失函数的图像风格迁移方法的特点。

2.1.2.1 网络结构

如图 2.1 中，基于感知损失函数的实时图像风格迁移系统由两部分组成：一个图片转换网络 f_w 和一个损失网络 ϕ （用来定义一系列损失函数 L_1, L_2, L_3 ）。图片转换网络是一个深度残差网络，参数是权重 W ，它把输入的图片 x 通过映射 $y = f_w(x)$ 转换成输出图片 y ，每一个损失函数计算一个标量值 $l_i(y, \hat{y})$ ，衡量输出的 y 和目标图像 \hat{y} 之间的差距。图片转换网络使用随机最速下降法训练。

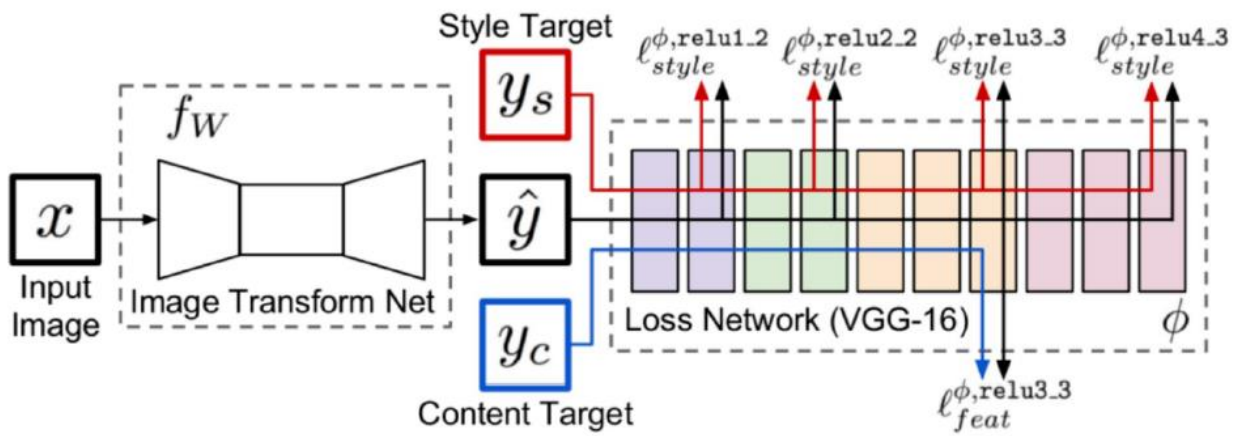


图 2.1 基于感知损失函数的实时图像风格迁移网络结构

2.1.2.2 感知损失函数

定义了两个感知损失函数，用来衡量两张图片之间的特征损失和风格损失。

(1) 特征（内容）损失：用 VGG 计算高级特征（内容）表示，这个取法和 Gayts 等^[1]使用 VGG-19 提取风格特征是一样的，函数如下：

$$l_{\text{feat}}^{\phi, j}(\hat{y}, y) = \frac{1}{C_j H_j W_j} \left\| \phi_j(\hat{y}) - \phi_j(y) \right\|_2^2 \quad (2.6)$$

(2) 风格损失：让 $\phi_{j(x)}$ 代表网络 ϕ 的第 j 层，输入为 x ，则特征关系图的形状为 $C_j \times H_j \times W_j$ ，定义矩阵 $G_{j(x)}$ 为 $C_j \times C_j$ 矩阵（特征矩阵）：

$$G_j^\varphi(x)_{c,c'} = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \varphi_j(x)_{h,w,c} \varphi_j(x)_{h,w,c'} \quad (2.7)$$

把图像转换网络的输出图像 y 和目标图像 \hat{y} 在损失网络的每一层都求出 Gram 矩阵，对应层之间欧氏距离之和即为风格损失，函数如下：

$$l_{style}^{\varphi,j}(\hat{y}, y) = \left\| G_j^\varphi(\hat{y}) - G_j^\varphi(y) \right\|_F^2 \quad (2.8)$$

2.2 图像风格迁移模块实现

如同其他利用深度学习模型完成特定任务的模式，图像风格迁移任务也分为模型的训练阶段和模型的测试（即图像生成）阶段。在本设计中，主要参考基于感知损失函数的图像风格迁移方法^[2]，并在其基础上做了少许修改。

2.2.1 运行环境配置

若要运行和测试图像风格迁移模块，需要配置以下环境：

- TensorFlow-gpu 1.12.0
- Python 3.6, Pillow 3.4.2, scipy 0.18.1, numpy 1.11.2
- NVIDIA GPU, CUDA 9, cuDNN 7

2.2.2 风格模型的训练

本设计的神经网络风格模型分为图像转换网络和损失网络。

（1）图像转换网络

该网络的输入是要风格转换的图像，输出是转换后的图像，用于生成风格迁移的只是这部分网络，具体的网络结构如下表 2.1。

图像转换网络总体属于一个残差网络，一共由 6 个卷积层、5 个残差块构成。这里没用采用池化操作，而在两个步长为 2 的卷积层做下采样，在两个步长为 1/2 的卷积层做上采样。5 个残差块都是使用相同个数的（128 个）卷积核，每个残差块中都有 2 个卷积层（ 3×3 核），这里的卷积层中没有进行标准的零填充（因为使用零填充会使生成出的图像的边界出现严重伪影）。为了保证输入输出图像大小不改变，在图像初始输入部分加入了反射填充。这里的残差块使用 Gross 等^[14]的残差网络，但是将批归一化用 Ulyanov 等^[15]的实例归一化代替。

表 2.1 图像转换网络结构

网络层	激活大小
输入	$3 \times 256 \times 256$
反射填充 (40×40)	$3 \times 256 \times 256$
$32 \times 9 \times 9$ 卷积, 步长 1	$32 \times 336 \times 336$
$64 \times 3 \times 3$ 卷积, 步长 2	$64 \times 168 \times 168$
$128 \times 3 \times 3$ 卷积, 步长 2	$128 \times 84 \times 84$
残差块, 128 个卷积核	$128 \times 80 \times 80$
残差块, 128 个卷积核	$128 \times 76 \times 76$
残差块, 128 个卷积核	$128 \times 72 \times 72$
残差块, 128 个卷积核	$128 \times 68 \times 68$
残差块, 128 个卷积核	$128 \times 64 \times 64$
$64 \times 3 \times 3$ 卷积, 步长 1/2	$64 \times 128 \times 128$
$32 \times 3 \times 3$ 卷积, 步长 1/2	$32 \times 256 \times 256$
$3 \times 9 \times 9$ 卷积, 步长 1	$3 \times 256 \times 256$

（2）损失网络

使用与 Gayts 等^[1]相同的损失网络 VGG19（一个用于图像分类任务的网络模型），是直接采用 ImageNet 预先训练好的，但在训练中权值不再更新。该网络的存在是用于定义感知损失函数来计算图像间内容和风格的感知偏差的。

（3）训练数据集和超参数设置

- COCO 数据集：共 8 万张训练图，所有图像调整至 256×256 大小
- 批大小：4
- 迭代次数：40000
- 轮数：2
- 优化函数：Adam
- 初始学习速率：0.001
- 没有使用权重衰减或 dropout

2.2.3 风格化图像的生成

在模型训练阶段，针对一种风格的前馈神经网络已经训练完成。将一张任意图像输入该前馈神经网络模型（不需要经过损失网络），即可输出有着内容图的内容信息和风格图的风格信息的图像，则图像风格化任务完成。这里，我们使用命令行调用评估脚本，命令行格式如下：

```
python evaluate.py --checkpoint path/to/style/model.ckpt \  
--in-path dir/of/test/imgs/ \  
--out-path dir/for/results/
```

2.3 本章小结

本章首先分别分析了基于迭代优化和基于感知损失函数的图像风格迁移方法，介绍了它们的数学原理和网络模型结构，然后详尽地陈述了本设计中针对图像风格迁移任务的运行环境配置、网络模型的具体实现结构和超参数设置，对本设计涉及的图像风格迁移模块的开发进行了详尽的叙述。

第3章 线稿自动上色关键技术

线稿自动上色是计算机视觉领域中的一个新兴的课题，本章主要研究两级线稿自动上色方法，首先从宏观上介绍其整体思路，然后分别从草稿绘制阶段和草稿细化阶段分析其数学原理，定义损失函数。最后针对两级线稿自动上色方法，从运行环境配置、网络模型训练和图像生成三个方面介绍本设计中线稿自动上色模块的实现。

3.1 线稿自动上色方法分析

两级线稿自动上色是一种基于颜色提示的用户交互图像上色方法，是仅针对线稿上色的半自动学习框架。这个框架包括两个阶段，在第一个绘制草稿阶段，该模型猜测线稿中的颜色区域并在草图上绘制丰富多样的颜色以获得颜色草稿；在第二个细化阶段，它会检测不自然的颜色，并尝试修复和细化结果。与其他方法相比，这种两级阶段设计有效地将复杂的着色任务划分为两个更简单的且目标更清晰的子任务。这样可以简化学习并提高着色质量。该模型可以完美解决诸如水色模糊、颜色失真和纹理暗淡之类的问题。

模型训练的框架如图 3.1 所示。草稿绘制阶段的任务是确定颜色组成并预测颜色草稿，以生动地绘画为目标。然后，草稿细化阶段识别不正确的颜色区域，并用一组额外的用户颜色提示对其进行细化。这两个阶段背后的模型是独立训练的。



图 3.1 两级线稿自动上色模型训练框架

模型测试（生成）框架如图 3.2 所示。在测试阶段，两个阶段连接在一起以生成最终输出。

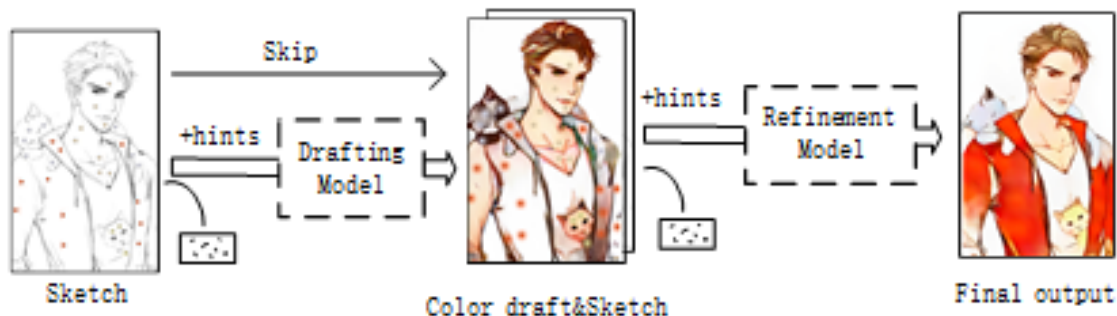


图 3.2 两级线稿自动上色模型测试框架

3.1.1 草稿绘制阶段

在此阶段，深层网络模型学习如何切丁输入草图的全局颜色组成并预测颜色草图。网络不需要产生高质量的结果，但它应遵循用户的颜色提示并积极地在草图上填充颜色，目的是丰富颜色变化。为实现此目的，需要一个起草网络 G 从草图 x 创建颜色草稿预测 \hat{y}_m 用户提示 u_i 。

对于任何草图 x 和用户提示 u_i ，将 $G(x, u_i)$ 的前馈预测作为 \hat{y}_m ，并通过优化损失函数 l_{draft} 来优化网络 ($\lambda = 0.01$):

$$l_{draft} = \arg \min_G \max_D E_{x, u_i, y_f \sim P_{data}(x, u_i, y_f)} [\|y_f - G(x, u_i)\|_1 + \alpha L(G(x, u_i)) - \lambda \log(D(y_f)) - \lambda \log(1 - D(G(x, u_i)))] \quad (3.1)$$

另外，积极调整损失 $L(x)$ 用来鼓励草图 $G(x, u_i)$ 颜色更加丰富:

$$L(x) = - \sum_{c=1}^3 \frac{1}{m} \sum_{i=1}^m \left(x_{c,i} - \frac{1}{m} \sum_{i=1}^m x_{c,i} \right)^2 \quad (3.2)$$

3.1.2 草稿细化阶段

绘制草稿阶段所生成的图像尚不能满足工业用途，因为它们可能包含颜色错误和伪影。为了完善绘画，我们必须确定有问题的区域并修复他们。为了实现这一目标，需要用到另一种深度神经模型。它可以学习检测和细化有问题的颜色区域。该模型读取一组新的用户提示，以便为草稿引入新颜色。同时，检测并移除草稿中的颜色错误和伪影。

现假设有数据集 P_{data} ，其中包含有错误的颜色草稿 y_m 和真实结果 \hat{y}_m ，草图 x 、用户颜色提示 u_{ii} 作为输入， \hat{y}_j 是输出，则损失函数 l_{refine} 可写为 ($\lambda = 0.01$):

$$l_{refine} = \arg \min_G \max_D E_{x, u_{ii}, y_m, y_f \sim P_{data}(x, u_{ii}, y_m, y_f)} [\|y_f - G(x, u_{ii}, y_m)\|_1 - \lambda \log(D(y_f)) - \lambda \log(1 - D(G(x, u_{ii}, y_m)))] \quad (3.3)$$

3.2 线稿自动上色模块实现

采用 3.1 节的两级线稿自动上色方法，绘制草稿阶段和草稿细化阶段分别进行训练。

3.2.1 运行环境配置

若要运行和测试线稿自动上色模块，需要配置以下环境:

- TensorFlow-gpu 1.12.0
- Keras 2.2.4
- bottle 0.12.16
- gevent 1.4.0

- h5py 2.9.0
- Paste 3.0.8
- opencv-python 4.0.0.21
- Scikit-image 0.14.2

3.2.2 线稿上色模型的训练

本设计的线稿自动上色模块的模型训练分为草稿绘制阶段训练和草稿细化阶段训练。网络结构如图 3.3 所示，(a) 为草稿绘制阶段网络模型，(b) 为草稿细化阶段网络模型，两个神经网络模型均为生成式对抗网络。

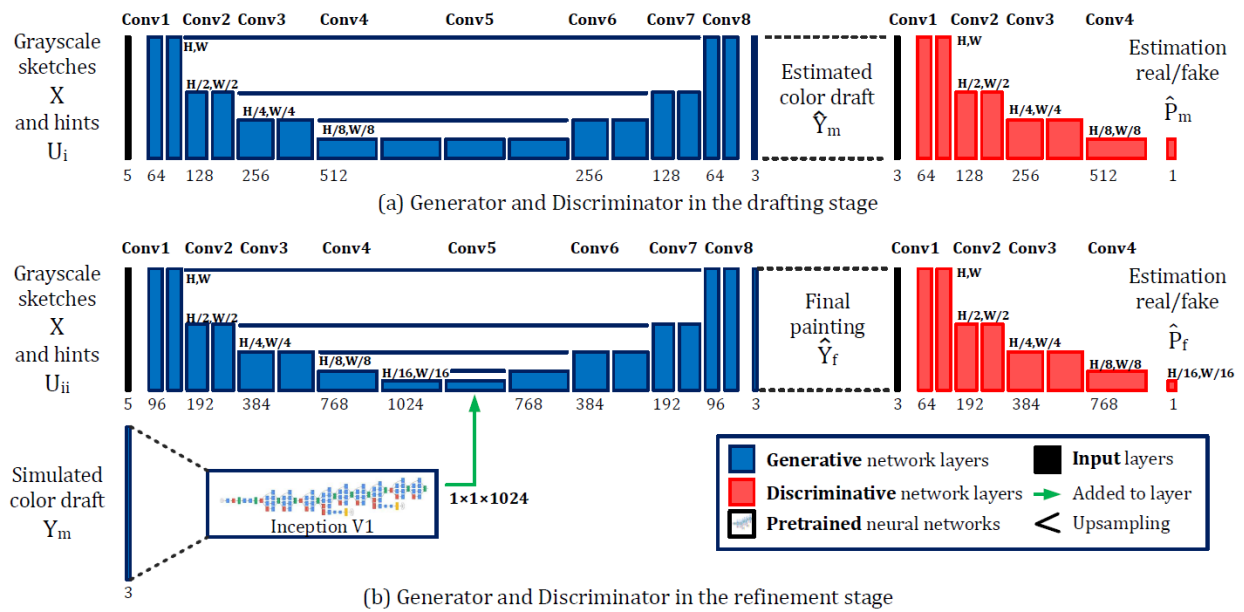


图 3.3 两级线稿自动上色网络结构

训练数据集和超参数设置如下：

- 数据集：Danboroo 数据集作为已上色的画稿，用 PaintsChainer 提取线稿作为对应线稿，用 Zhang 等^[18]的方法模拟用户输入色彩提示点
- 批大小：4
- 优化函数：Adam
- 迭代次数：直到损失函数收敛

3.2.3 结果图像生成

将训练好的草稿绘制模型和草稿细化模型连接起来，形成一个前馈神经网络。把任意线稿输入模型中，模型会先后生成草稿和已细化定稿，若需要修改区域颜色，可以加入用户色彩提示点，并重复以上操作即可。

3.3 本章小结

本章从两级线稿自动上色方法的两个阶段（草稿绘制阶段和草稿细化阶段）详尽地分析了两级线稿自动上色方法，介绍了核心的算法逻辑，然后针对线稿自动上色方法陈述了该模块的运行环境配置、网络模型结构和超参数设置，对本设计涉及的线稿自动上色模块的开发进行了详尽的叙述。

第 4 章 网站设计与测试

本章从软件工程的角度阐述艺术风格迁移网站的设计和测试情况。首先从结构设计和功能设计方面对网站进行总体设计，然后在网站详细设计中，展示主要页面的设计，为各功能设计关系型数据库，根据业务流程为两个主要模块设计详细功能。最后为图像风格迁移模块和线稿自动上色模块设计测试用例，并给出测试结果和对应截图。

4.1 网站设计

4.1.1 网站总体设计

4.1.1.1 网站结构设计

本网站主要由首页、用户管理、图像风格迁移和线稿自动上色四个部分组成。网站结构功能图如图 4.1 所示。

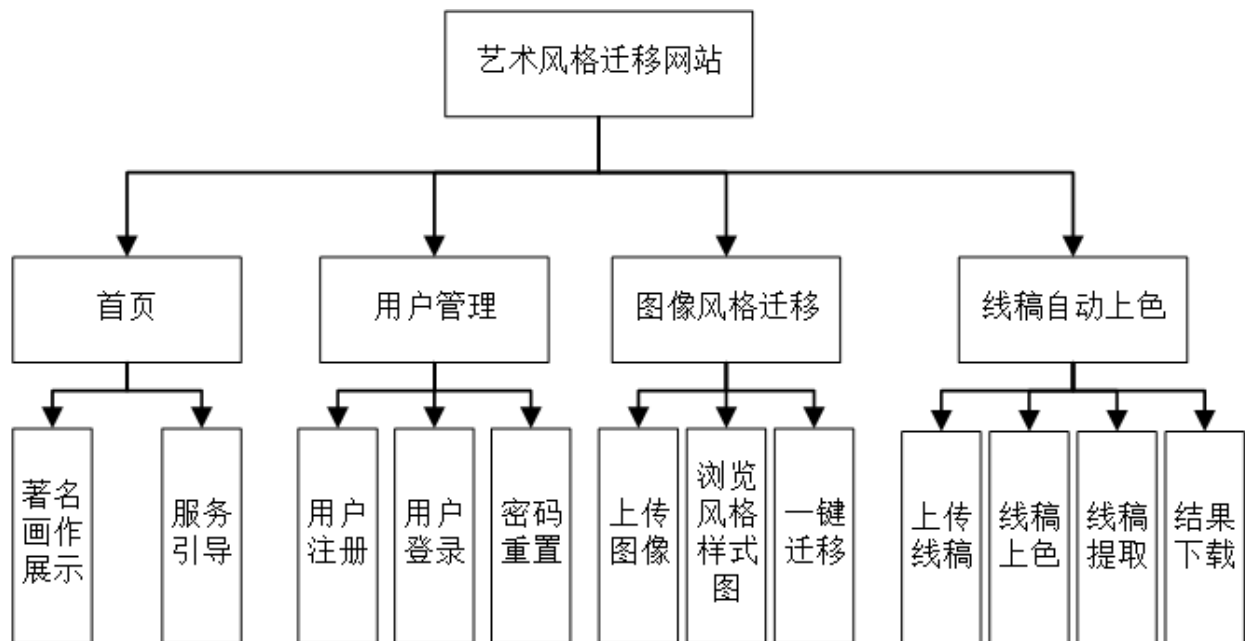


图 4.1 艺术风格迁移网站功能结构图

4.1.1.2 网站功能设计

（1）图像风格迁移模块

图像风格迁移指利用算法（本设计使用卷积神经网络）处理数字图像，使其具有另一幅图像的外观或视觉风格。该模块的实现能让用户实现实时在线的任意图像风格迁移。考虑到用户的普遍性和非专业性，将用户操作简化至三步：上传图像、选择风格、确认。

在上传图像步骤，提供图像上传功能，系统对图像大小进行判断，若小于等于 500KB，则直接上传；若大于 500KB，则将之压缩至 500KB，再进行上传处理。上传完成后，根据

图像的尺寸等比例放大/缩小，使之适应页面中图像展示框的大小，并提示用户上传成功。

在选择风格步骤，提供下拉框选择风格功能，选择风格的同时页面自动展示风格图像，为用户提供参考。

在确认步骤，点击确认按钮，若用户已上传图像，服务器调用脚本，将用户上传的图像和选择的风格名称作为输入，生成风格化后的图像，并将之展示在页面中，提示用户生成成功；若用户未上传图像，则提示用户应上传图像。

（2）线稿自动上色模块

线稿自动上色指对用户上传的任意线稿结合色彩提示点进行自动上色，使其具有与人工上色相同的效果。该模块的实现能让普通用户学习上色技巧，也能节省经验丰富的画家为线稿上色的时间，提高产量。

本模块主要提供线稿上传、色彩提示点绘制、一键自动上色、线稿提取的功能。

在线稿上传步骤，提供上传功能，服务器保存上传的图像，并对其进行大小调整、去噪声等优化操作。

在色彩提示点绘制阶段，提供选色及绘制功能，用户也可以一键清除或选择清除提示点。

在一键自动上色阶段，用户点击箭头，服务器接收色彩提示点信息，参照提示点信息调用脚本生成完整画稿，并提供下载功能。

若用户上传有色图片，可以选择线稿提取功能，服务器调用脚本对线稿进行提取，并显示在页面中。

4.1.2 网站详细设计

本节根据 4.1.1 节网站总体设计的内容，对艺术风格迁移网站进行详细设计。艺术风格迁移网站作为提供在线服务的网站，网站本身简洁清晰，数据主要用于训练神经网络，其他数据较少。按照总体设计中所确定的网站结构和功能，设计网站的界面、数据库、功能模块等内容，逐个模块完成程序的设计和测试。

4.1.2.1 网站界面设计

为了吸引用户使用，提升用户体验，良好的人机交互界面是必不可少的。在设计网站界面时应注意保持界面的间接性，不要过于复杂，既要突出网站的特色和设计风格，又要便于用户上手操作。

（1）主页设计



图 4.2 艺术风格迁移网站主页

主界面如图 4.2 所示，用户访问 www.styletransfer.cn 后，会进入网站主页，页面上方显示网站 logo 和“登录”、“注册”按钮，并有一行导航栏链接至所有页面，页面正中显示一个轮播图，循环展示世界著名画作，下方可以选择服务类型，可选“图像风格迁移”和“线稿自动上色”，用户点击后可跳转至相应界面。最下方为页面尾部，展示备案号、快速连接、联系地址和社交媒体链接。

（2）注册界面设计

注册界面如图 4.3 所示，页面的头部（包括 logo、导航栏等）和尾部（备案号、快速链接等）与主页相同。用户在页面主体中表单部分输入用户名、电子邮箱、密码和确认密码，然后点击提交，系统确认输入格式无误后会向数据库写入用户信息，并向用户的邮箱发送包含用户信息的邮件。



The image shows a user registration page with a dark blue header containing navigation links: 首页, 风格迁移, 线稿上色, 关于, F&Q, and 支持. The main content area has a light gray background. At the top, the word '注册' (Register) is displayed in a large, bold, dark blue font. Below it, a link '如果你已经注册, 请[登录](#)' (If you are already registered, please [login](#)) is shown. The registration form consists of four input fields with labels: '用户名' (Username), '电子邮箱' (Email), '密码' (Password), and '再次输入密码' (Re-enter password). Each field is a simple white rectangle with a thin gray border. At the bottom of the form is a dark blue button with the white text '提交' (Submit).

图 4.3 用户注册页面

（3）登录界面设计

登录界面如图 4.4 所示，页面的头部（包括 logo、导航栏等）和尾部（备案号、快速链接等）与主页相同。页面主体分为左右两部分：左边引导新用户注册，点击“创建账号”即可跳转至注册界面；右边用户输入电子邮箱地址、密码和验证码即可登录账号，点击“登录”验证成功后可跳转至主页，进行后续操作。点击“忘记密码”即跳转至“忘记密码”页面，输入电子邮箱后，系统向该地址发送密码重置链接，点击链接即可重置密码。



The image shows a user login page with a dark blue header containing navigation links: 首页, 风格迁移, 线稿上色, 关于, F&Q, and 支持. The main content area has a light gray background. On the left side, under the heading '新用户' (New User), there is a sub-heading '通过注册成为我们的会员, 你将可以免费享受个人模板创建, 云端存储等功能。' (By registering as our member, you can enjoy personal template creation, cloud storage, etc. for free). Below this is a dark blue button with the white text '创建账号' (Create Account). On the right side, under the heading '已注册的用户' (Already Registered User), there is a sub-heading '如果你已经拥有账号, 请登录。' (If you already have an account, please login). Below this are three input fields: '电子邮箱地址*' (Email Address*), '密码*' (Password*), and '验证码:' (Captcha:). The captcha image shows the characters 'GSO4'. At the bottom of the form are two buttons: a dark blue button with the white text '登录' (Login) and a light gray button with the dark blue text '忘记密码?' (Forgot Password?).

图 4.4 用户登录页面

（4）图像风格迁移界面设计

图像风格迁移界面如图 4.5 所示，页面的头部（包括 logo、导航栏等）和尾部（备案号、快速链接等）与主页相同。页面分为左右两个部分，左边是图像风格迁移的操作步骤。用户根据操作步骤操作：先选择图像文件，右边左侧的框内显示上传的图像；再选择风格，风格选择后下方会自动展示风格图片供参考；最后点击确认，后台将风格化后的图像生成完毕后在最右边的框内显示出来。

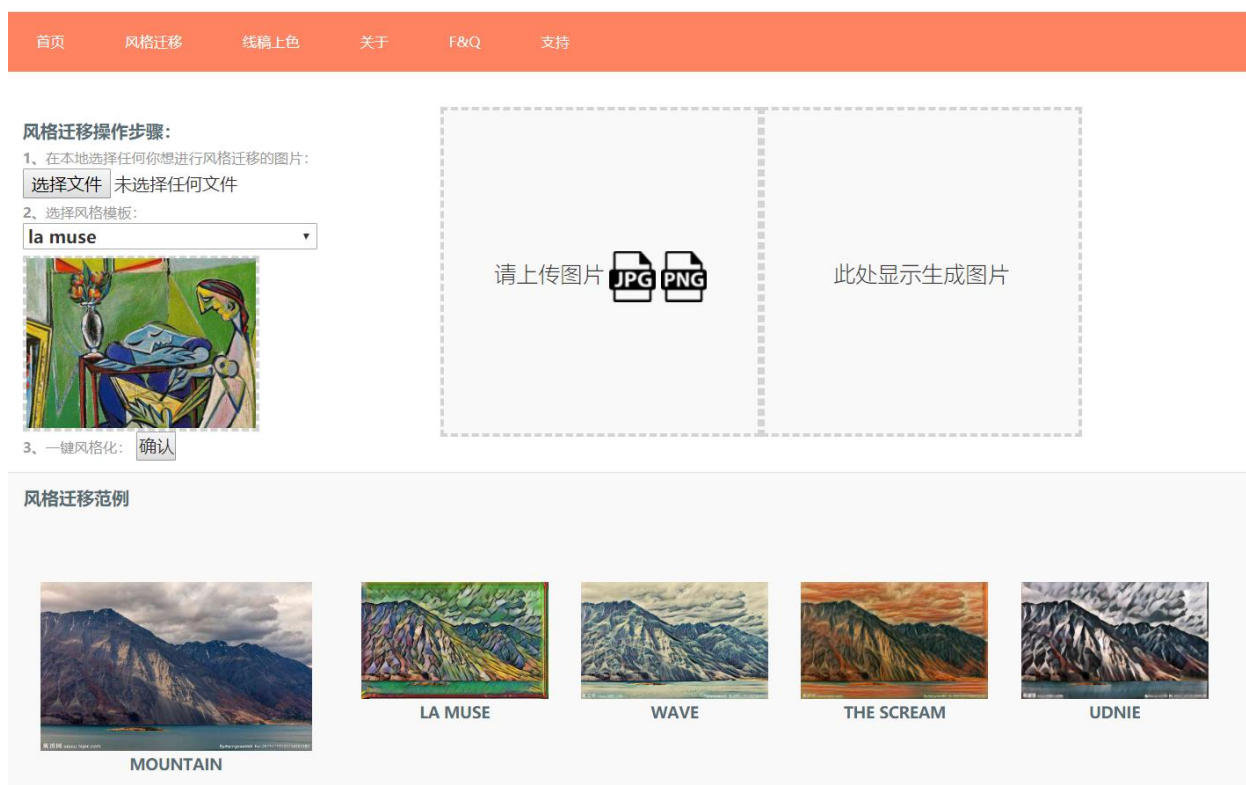


图 4.5 图像风格迁移页面

（5）线稿自动上色界面设计

线稿自动上色页面如图 4.6 所示。用户点击“开始使用”，即可跳转至线稿自动上色应用页面，如图 4.7 所示。该页面分为三个部分：画布池、工具条和选色器。画布池中包含三张画布，分别展示输入线稿、着色草稿和最终结果。用户上传线稿后，线稿显示在第一个画布；然后，用户可以在选色器中选择颜色，涂在画布上，作为神经网络生成结果图像的参考；点击箭头，后台生成着色草稿和最终结果图像，分别显示在第二张和第三张画布上；用户可以点击下载按钮来下载结果图像。

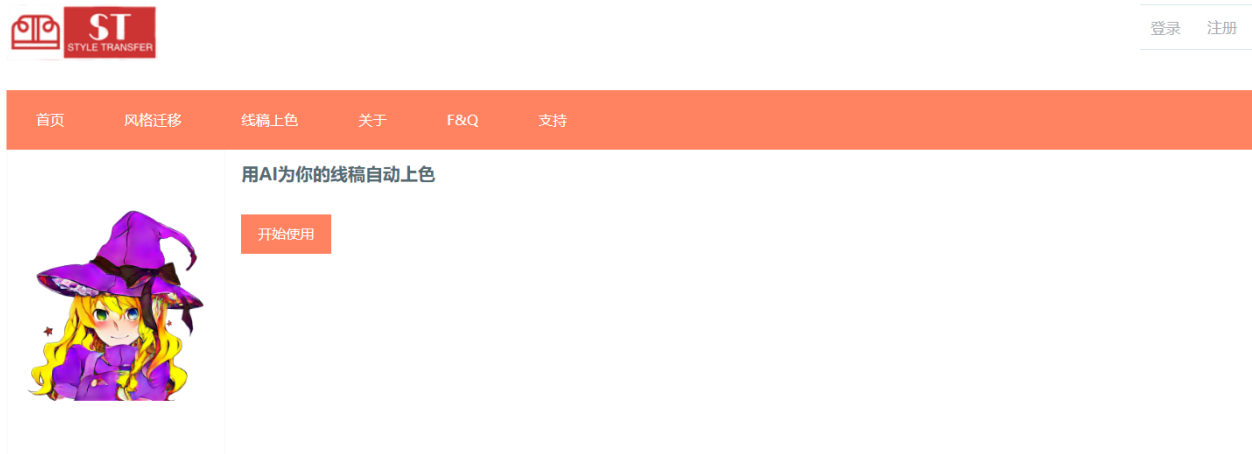


图 4.6 线稿自动上色引导页面



图 4.7 线稿自动上色应用页面

4.1.2.2 数据库设计

根据网站功能的需要，建立了数据库表，如表 4.1-4.4 所示。

表 4.1 用户登录信息表

字段名称	数据类型	是否可空	字段说明	备注
id	int	否	用户编号	自增
password	varchar(200)	否	用户登录密码	SHA256 算法加密
last_login	datetime	否	上次登录日期和时间	精确到微秒
is_superuser	Boolean	否	是否是超级用户	默认否
username	varchar(50)	否	用户名	
first_name	varchar(25)	是	用户的名	
last_name	varchar(25)	是	用户的姓	
email	varchar(50)	否	电子邮件地址	
is_staff	Boolean	否	是否是管理员	默认否
is_active	Boolean	否	是否活跃	默认是
date_joined	datetime	否	注册时间	

用户登录信息表记录网站用户登录的信息，主键为 id。

表 4.2 图片上传地址信息表

字段名称	数据类型	是否可空	字段说明	备注
id	int	否	上传图片编号	自增
pic	varchar(100)	否	图片保存地址	
ip	varchar(20)	否	用户的 ip	
uploaded_timestamp	Varchar(20)	否	上传的时间戳	

图片上传地址信息表记录用户上传图片的地址信息，主键为 id。

表 4.3 Django 会话表

字段名称	数据类型	是否可空	字段说明	备注
session_key	varchar(100)	否	会话编号	
session_data	varchar(100)	否	会话临时保存的数据	
expire_date	datetime	否	会话失效时间	

Django 会话表记录 Django 后台临时记录的数据，如图片 id 和路径等，主键为 session_key。

表 4.4 数据库迁移记录表

字段名称	数据类型	是否可空	字段说明	备注
id	int	否	记录编号	自增
app	varchar(50)	否	执行迁移的 app	
name	varchar(50)	否	迁移记录	
applied	datetime	否	执行日期时间	

数据库迁移记录表记录 SQLite 数据库结构修改的记录，如增加表字段或删除表。主键为 id。

4.2.1.3 功能模块设计

（1）图像风格迁移模块设计

图像风格迁移模块主要用来完成对一张任意图像的风格化操作。在这个过程中，用户需要按照步骤完成整个操作：上传图像、选择风格模板、确认。

上传图像：点击“选择文件”按钮，弹出本地文件选择窗口，这里文件格式限制为图片格式（BMP、PCX、PNG、JPEG、GIF、GIF、DXF、CGM、CDR、WMF、EPS、EMF、PICT），且文件名不能含有中文，否则弹出警告提示“请修改文件名称，不能含有汉字”。用户选择本地图片后，点击“打开”，JavaScript 脚本判断文件大小，若大于 500KB，则使用 canvas 将其压缩至 500KB 以内。然后使用 ajax 上传至服务器，上传成功后弹出提示“上传成功”，并将图片调整尺寸（使其适应展示框大小）后在前端页面中展示。

选择风格模板：在下拉框中的 6 个预先训练好的风格模板中选择一个，则下拉框下方的预览框展示该风格原始画作，如《The Shipwreck of the Minotaur》和《Rain Princess》。

确认：点击“确认”按钮，若此时还未上传图片，则弹出警告“请上传图片”；若已上传图片成功，且已经选择风格模板（默认选择 La Muse），则将图片路径和风格模板名传至服务器，服务器调用 python 生成脚本生成风格化后的图像，将之保存至特定文件夹，并在前端页面显示。此时可以右键点击图片，选择“图片另存为”即可下载至本地。

至此，图像风格迁移的所有操作结束。若用户想对已上传的图片风格化至其他风格，可以重新选择风格，点击“确认”即可再次生成对应风格的图像；若想对其他图片进行风格迁移操作，则需要从“上传图像”开始重新操作。

（2）线稿自动上色模块设计

线稿自动上色模块主要用来完成对一张未上色线稿的上色操作。在这个过程中，用户需要按照规定步骤完成整个操作：上传线稿、画色彩提示点、确认。另外，本模块还提供颜色渲染、线稿提取、边缘强调的辅助功能。

上传线稿：在这个阶段，可以选择从本地上传一张线稿，或者选择预先训练好的参照。若选择本地上传，则点击“上传线稿”，弹出本地文件选择窗口，这里文件格式同样限制为

图片格式（BMP、PCX、PNG、JPEG、GIF、GIF、DXF、CGM、CDR、WMF、EPS、EMF、PICT）。选择文件并点击“打开”后，图片文件保存至服务器的静态文件夹中，并调用 python 脚本对其进行优化（改为灰度图、调整尺寸、去噪），将优化后的线稿保存至静态文件夹中，并在前端显示；若选择范例线稿，则点击“范例线稿”，然后在页面中选择一张范例即可，这里范例都是预先训练好的，从服务器直接下载至前端页面显示，跳过了图像优化阶段。

画色彩提示点：在工具条中选中“色彩提示点”，可以从选色器和调色盘中选择颜色，画在线稿对应的位置，如果画错了，可以在工具条中选择“色彩橡皮擦”，擦除色彩提示点即可。

确认：如果已上传本地线稿（或选择范例线稿），且已经在合适的位置画上了色彩提示点，则可以点击右箭头，则线稿存储路径和色彩提示点信息上传至服务器，服务器调用 python 脚本进行线稿上色处理。处理完成后在前端页面中显示，并可以点击下载按钮保存至本地。

边缘强调：在工具条中选中“边缘强调”，再按照规定步骤操作，即可得到描边后的已上色画稿。这种方法生成的画稿边缘更清晰，总体颜色更深。

颜色渲染：在工具条中选中“颜色渲染”，再按照规定步骤操作，即可得到颜色渲染后的已上色画稿。这种方法生成的画稿不会出现水彩模糊的问题，颜色更清晰。

线稿提取：在工具条中选中“线稿提取”，如此时已经上传图像或选择范例线稿，则服务器将图像处理后在前端页面显示该图像的线稿，然后可以根据规定步骤进行线稿上色操作，相当于对已上色的线稿进行再次上色。

4.2 网站测试与运行结果

4.2.1 图像风格迁移模块测试及运行结果

图像风格迁移模块主要是完成图像上传、风格选择、图像风格化的操作，本节中选择一幅来源于网络的图像作为内容图像，对上述三个操作逐步进行测试。测试用例表如表 4.5 所示。

表 4.5 图像风格迁移模块测试用例

测试功能点	测试用例步骤说明	输入数据	预期结果	实际结果	测试状态
图像上传	1.点击图像风格迁移页面“选择文件”按钮 2.在文件选择框中选择图片文件 3.点击“打开”	选择文件“example02.jpg”	上传成功，内容图展示框显示图片	上传成功，内容图展示框显示图片	通过
风格选择	1.点击风格选择下拉框 2.选择期望的风格	选择“wave”风格	下拉框显示“wave”，下方预览框显示“wave”原画作	下拉框显示“wave”，下方预览框显示“wave”原画作	通过
图像风格化	1.点击“确认”	点击“确认”	生成图片展示框出现进度条，稍等后提示“风格化成功”，在框中显示有“wave”风格的“example02.jpg”	生成图片展示框出现进度条，稍等后提示“风格化成功”，在框中显示有“wave”风格的“example02.jpg”	通过

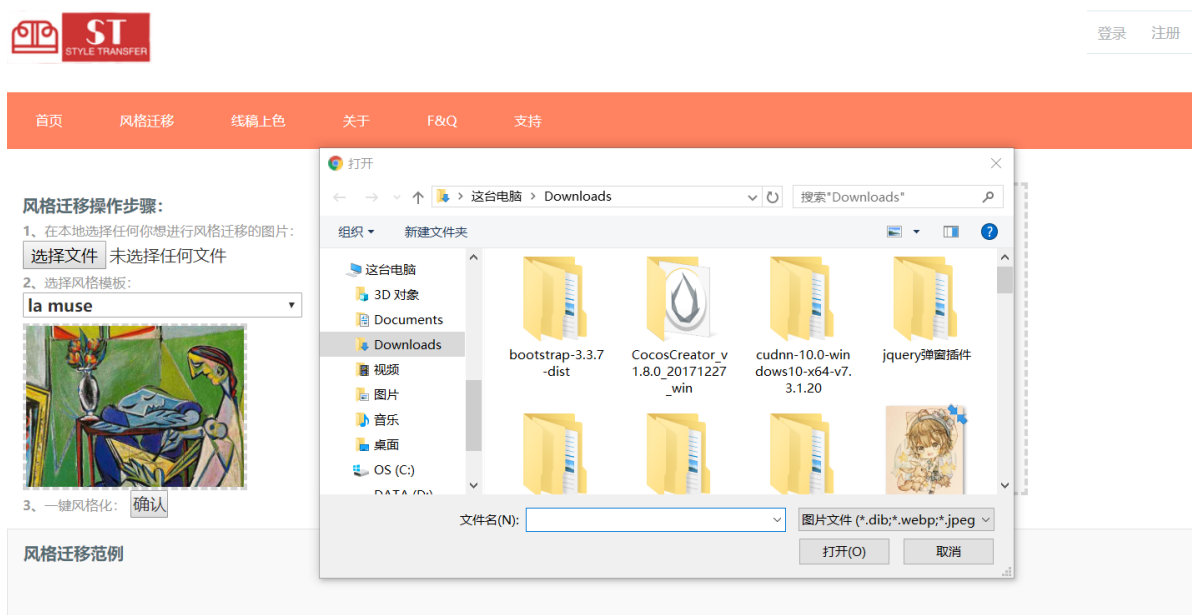


图 4.8 选择文件测试图

图像上传操作，点击图像风格迁移页面“选择文件”按钮，弹出本地文件选择框，如图 4.8。

图像上传操作，选择文件后点击“确认”，即可上传图像，上传成功时如图 4.9。



图 4.9 上传文件成功测试图

点击风格模板下拉框，可以选择期望的风格模板，这里选择“wave”，如图 4.10。



图 4.10 选择风格测试图

点击一键风格化后面的“确认”按钮，即可生成风格化图像，如图 4.11 和 4.12。

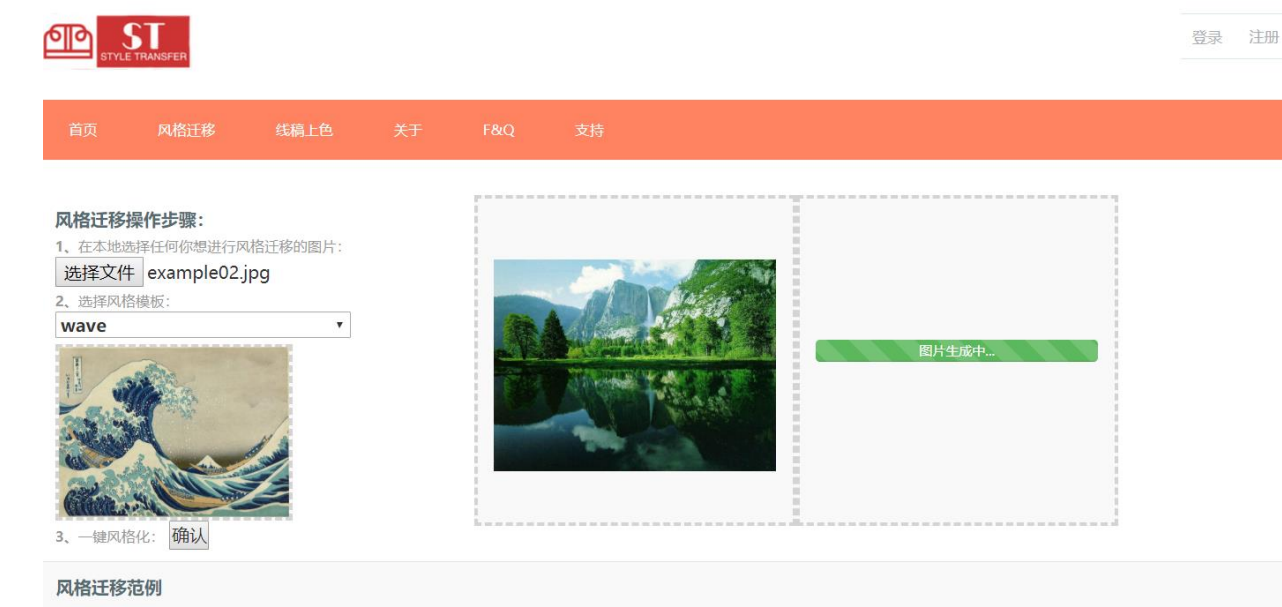


图 4.11 风格化图像生成中测试图

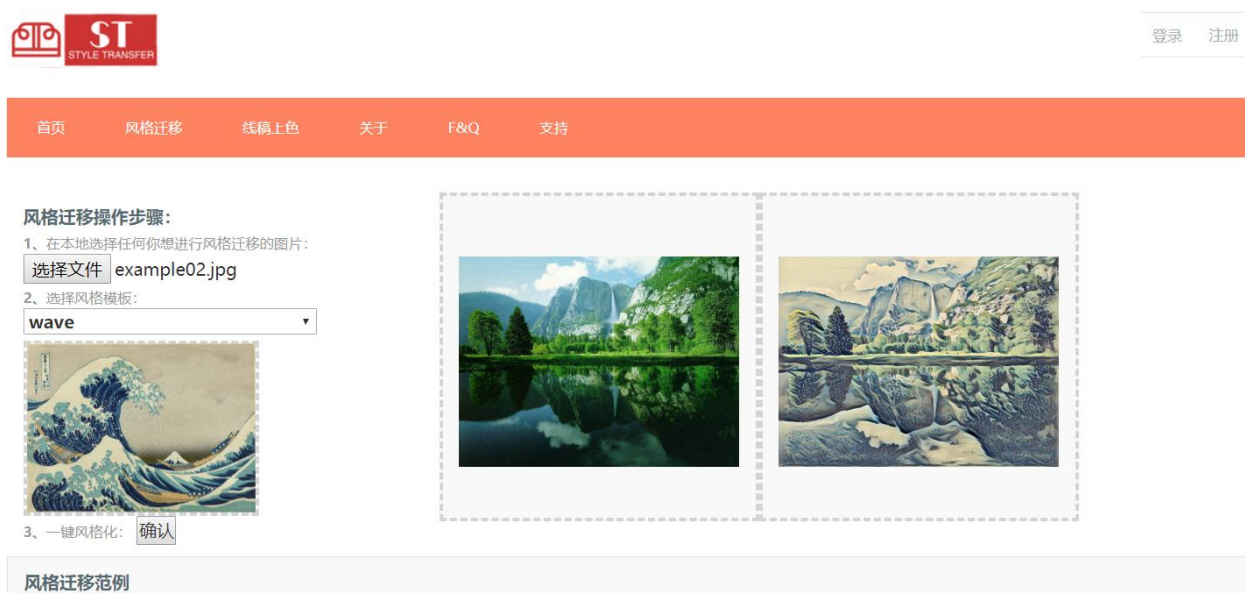


图 4.12 风格化图像生成成功测试图

4.2.2 线稿自动上色模块测试及运行结果

线稿自动上色模块主要是完成上传线稿、画色彩提示点、确认的操作，本节中选择一幅来源于网络的线稿，对上述三个操作逐步进行测试。测试用例表如表 4.6 所示。

表 4.6 线稿自动上色模块测试用例

测试功能点	测试用例步骤说明	输入数据	预期结果	实际结果	测试状态
上传线稿	1.点击“上传线稿” 2.选择线稿 3.点击“打开”	选 择 文 件 “ 20180405202737_vY2UT.jpg ”	三块画布均显示该图像，且进入图像预处理阶段	三块画布均显示该图像，且进入图像预处理阶段	通过
画色彩提示点	在下方选色器中选择合适的颜色画在合适的位置	色彩提示点	在画布上显示相应的色彩提示点	在画布上显示相应的色彩提示点	通过
确认	点击画布间的右箭头	点击右箭头	进入线稿上色生成阶段，稍等后显示已上色画稿	进入线稿上色生成阶段，稍等后显示已上色画稿	通过

上传线稿操作，点击线稿自动上色页面的“上传线稿”按钮，弹出本地文件选择框，选择文件并点击“打开”后，进入线稿预处理状态，如图 4.13。



图 4.13 线稿预处理测试图

在工具条中的色彩选择器中选择合适的颜色在线稿上绘制色彩提示点，如图 4.14。

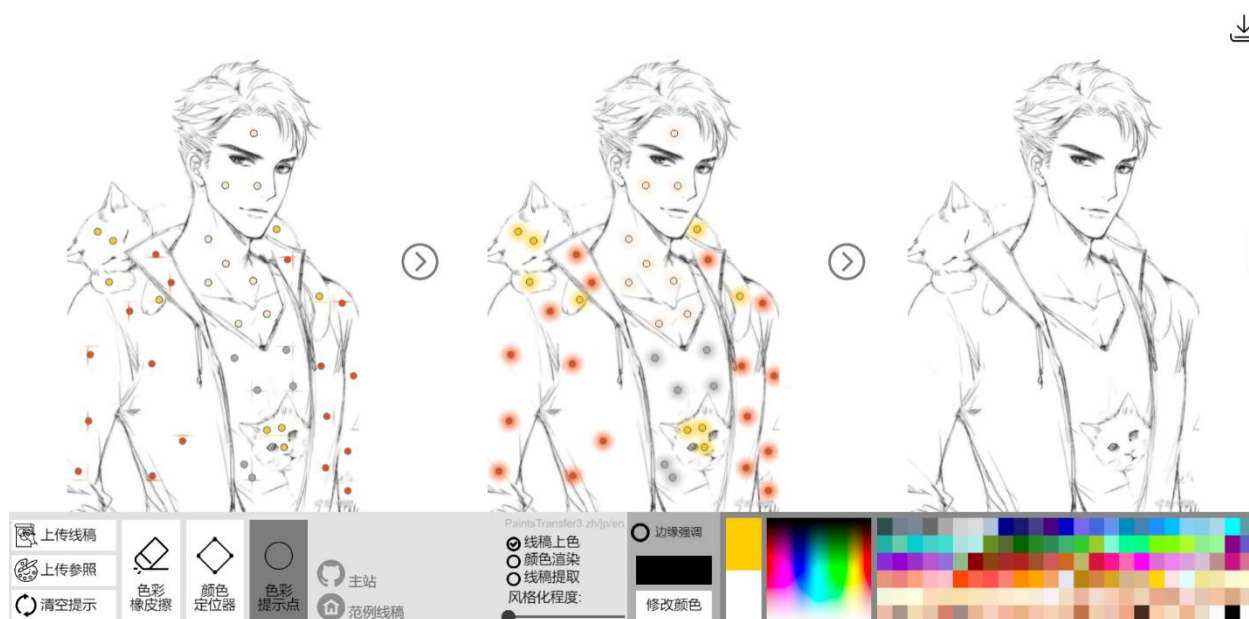


图 4.14 色彩提示点绘制测试图

点击画布间的右箭头，即可开始对线稿上色，完成后的效果如图 4.15。

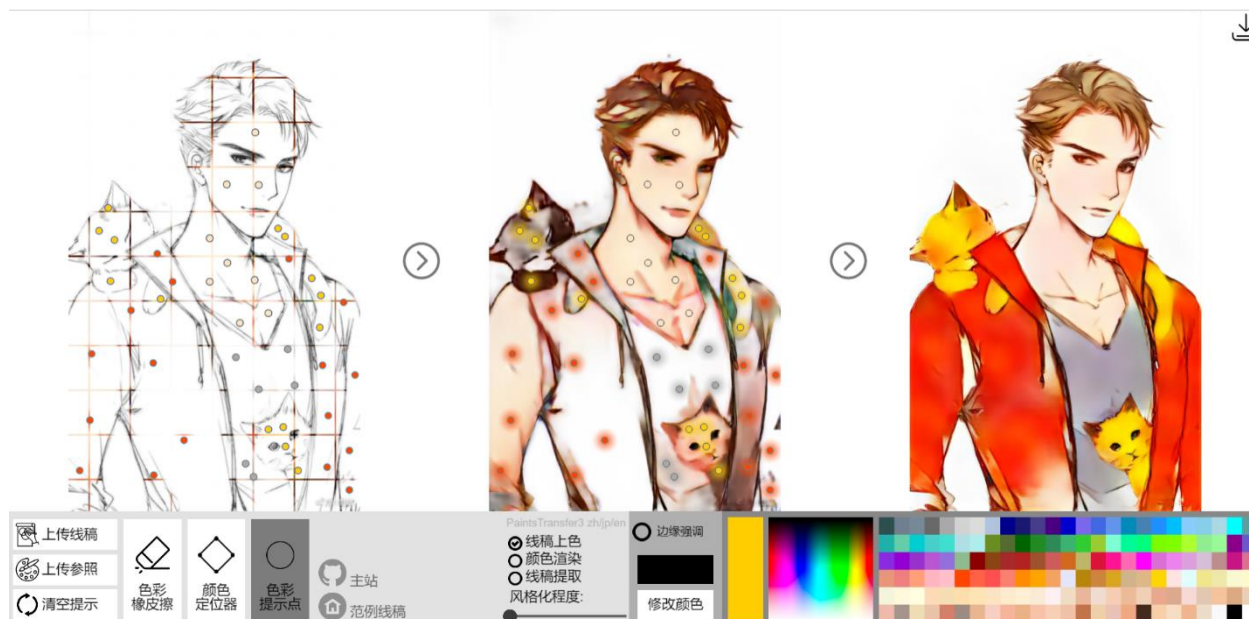


图 4.15 线稿自动上色完成测试图

完成后点击下载按钮即可下载上色后的画稿。

第 5 章 总结与展望

5.1 论文工作总结

5.1.1 已完成的主要工作

(1) 根据需求分析进行网站设计，围绕艺术风格迁移（尤其是图像风格迁移和线稿自动上色）的特点和实际用户的需要，设计了一个符合用户艺术审美需求、简单实用的在线艺术风格迁移平台，为广大艺术爱好者提供了便利。

(2) 研究了图像风格迁移和线稿自动上色两个深度学习任务的发展历程，阅读了若干相关文献，并从中选择合适的文献作为功能开发的参考。

(3) 完成了网站的整体开发。完成了图像风格迁移、线稿自动上色脚本的开发和模型的训练，利用 Django 框架完成了网站的前后端开发，完成了网站在云服务器的部署和域名解析。

(4) 完成了毕业论文的撰写。通过查阅相关书籍和文献，总结国内外的图像风格迁移和线稿自动上色的研究现状，分析当前社会在这一方面的需求，对艺术风格迁移网站的开发进行总结，以论文的形式展现出来。

5.1.2 工作特点

(1) 研究内容新颖。图像风格迁移和线稿自动上色这两个任务领域是近几年新兴起的研究热点。虽然国内外的专家学者在这两个领域获得了大量的研究成果，但是相应的商业化应用还不成熟，普通用户无法享受到科技革新带来的便利。开发艺术风格迁移网站是将理论结合实际的一次勇敢尝试，试图把最新的技术商业化，供普通用户使用。

(2) 工作量较大。在开发艺术风格迁移网站的过程中，需要为云服务器配置运行环境，管理网络端口，设置域名解析，搭建数据库连接。另外，学习并实践了全站开发的知识，包括 HTML+CSS+JavaScript 的前端开发技术，以及基于 Django 框架的 Python Web 技术。

(3) 用户体验良好。网站的每一个细节都经过仔细斟酌考虑，比如用户每次操作后的弹出提示、用户注册后收到个人登录信息邮件、密码重置操作等。另外，网站具有良好的人机交互界面，根据界面上的相关提示便可完成相应操作，使用户在使用时易上手。

5.2 未来展望

因为用于毕业设计的时间有限，虽然完成了艺术风格网站的基本功能，但是可以从以下几点在网站进行完善：

(1) 网站所使用的服务器规格较低，进行图像风格迁移和线稿自动上色操作时会使服务器满负荷运行。可以提高服务器配置规格，或使用相应的内存管控技术，限制单用户分配的内存上限。

（2）完善图像风格迁移的功能。这个功能虽然可行，但无法做到实时两张任意图像的风格转移，而必须使用已有的风格。在以后的研究中，可以优化这个功能。

（3）增加用户与功能间的联系。本网站虽然建设了用户登录注册模块，但是这个模块与所提供的服务间没有实质性的联系。在后期的版本迭代中，可以加强这方面的联系，如增加风格化图片的云存储和分享功能等。

参考文献

- [1] Gatys L A, Ecker A S, Bethge M. Image style transfer using convolutional neural networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2414-2423.
- [2] Johnson J., Alahi A., Fei-Fei Li. Perceptual Losses for Real-Time Style Transfer and Super-Resolution.[C]//European Conference on Computer Vision, 2016:694-711.
- [3] Gatys, Leon & Ecker, Alexander & Bethge, Matthias. [J]. A Neural Algorithm of Artistic Style. 2015, arXiv. 10.1167/16.12.326.
- [4] Gatys LA, Ecker AS, Bethge M. Texture synthesis using convolutional neural networks[J]. Advances in Neural Information Processing Systems. 2015: 262–270.
- [5] Ulyanov D., Lebedev V., Vedaldi A., et al. Texture Networks: Feed-forward Synthesis of Textures and Stylized Images[C]. //International Conference on Machine Learning, 2016:1349-1357.
- [6] Dumoulin V , Shlens J , Kudlur M . A Learned Representation For Artistic Style[J]. arXiv e-print, 2017, arXiv: 1610.07629.
- [7] Huang, Xun & Belongie, Serge. (2017). Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization. 1510-1519. 10.1109/ICCV.2017.167.
- [8] Gatys L A , Ecker A S , Bethge M , et al. Controlling Perceptual Factors in Neural Style Transfer[J]. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) .2017: 3985-3993.
- [9] Luan F , Paris S , Shechtman E , et al. Deep Photo Style Transfer[J]. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: 4990-4998.
- [10] Risser E , Wilmot P , Barnes C . Stable and Controllable Neural Texture Synthesis and Style Transfer Using Histogram Losses[J]. arXiv e-print, 2017, arXiv:1701.08893.
- [11] J. Zhu, T. Park, P. Isola and A. A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 2242-2251.
- [12] Jing Y , Yang Y , Feng Z , et al. Neural Style Transfer: A Review[J]. arXiv e-print, 2017, arXiv:1705.04058.
- [13] 邓盈盈, 唐帆, 董未名. 图像艺术风格化的研究现状[J]. 南京信息工程大学学报(自然科学版), 2017(06):31-36.
- [14] Gross, S., Wilber, M.: Training and investigating residual nets[OL]. <http://torch.ch/blog/2016/02/04/resnets.html> (2016)
- [15] Ulyanov D , Vedaldi A , Lempitsky V . Instance Normalization: The Missing Ingredient for Fast Stylization[J]. arXiv e-print, 2016, arXiv: 1607.08022.

- [16] Richard Zhang, Jun-Yan Zhu, Phillip Isola, et al.. Real-Time User-Guided Image Colorization with Learned Deep Priors[J]. arXiv e-print, 2017, arXiv:1705.02999
- [17] Sangkloy, Patsorn Lu, Jingwan Fang, et al. (2017). Scribbler: Controlling Deep Image Synthesis with Sketch and Color[J]. 6836-6845. 10.1109/CVPR.2017.723.
- [18] Furusawa, Chie Hiroshiba, Kazuyuki Ogaki, et al. (2017). Comicolorization: semi-automatic manga colorization[J]. 1-4. 10.1145/3145749.3149430.
- [19] Yifan L, Zengchang Q, Tao W, et al. Auto-painter: Cartoon image generation from sketch by using conditional Wasserstein generative adversarial networks[J]. Neurocomputing, 2018:S0925231218306209-.
- [20] Lvmin Z, Chengze L, Tien-Tsin W, et al. Two-stage Sketch Colorization[J]. ACM Transactions on Graphics, 2018, Vol 37, No.6: Article 261.
- [21] Radford, Alec & Metz, Luke & Chintala, Soumith. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks[J]. arXiv e-print, 2016, arXiv:1511.06434.
- [22] Mahendran A, Vedaldi A. Understanding deep image representations by inverting them[J]. arXiv e-print, 2014, arXiv:1412.0035.

致 谢

本科的生涯即将结束，在老师和同学们的帮助下，我对计算机领域有了新的认知。从最开始的 C++、Java，到离散数学、算法分析和设计，我从一个没有接触过编程的技术小白，成长为一个能够独立开发完整项目的大学毕业生。这期间有汗水，有欢乐，正是因为老师和同学们的陪伴我才能一路过关斩将，收获成功。在此谨向我的老师和同学们表示诚挚的感谢。

我的毕业设计是一个基于 Django 框架的网站，在研究和设计时，我参考了大量有关迁移和线稿自动上色的科技论文，收获颇丰，使我感受到了这个领域在未来的无限可能。在此，感谢那些致力于研究这个领域的专家学者们、无私分享开发经验的博主们和热情友好的开源社区的开发者们。是你们的辛勤工作，让我能够站在巨人们的肩膀上完成我的毕业设计。

另外，感谢程艳芬老师和李超学姐从我大三一直到毕业设计结束期间的悉心指导，是她们让我体会到深度学习这个领域的魅力，带我走进科学研究的大门。在以后的学习研究中，我会谨记她们的教诲，把自己的热情投注在科研事业。

大学四年的学习、生活如同白驹过隙，回首往昔，倍感充实。一路走来，感谢父母作为我坚强的后盾，让我在遇到困难时不会退缩，砥砺前行。

最后，感谢武汉理工大学为我提供良好的学习环境，感谢每一位曾经教过我的老师，祝福每一位同学前程似锦，祝福母校越来越好。

李俊

2019 年 5 月 27 日