



老照片修复系统

学院： 电子与信息工程学院

组员 1： 包广垠

组员 2： 苗春玲

完成日期： 2022 年 12 月 28 日

摘要

本项目为同济大学研究生课程《模式识别》期末大作业。

项目分工如下：

任务描述	负责人
划痕修复	包广垠
图像上色	苗春玲
UI 界面	苗春玲
报告文档	包广垠
PPT 制作	包广垠、苗春玲

关键词：显著目标检测，图像上色，照片修复

目 录

1 背景	1
2 需求分析	2
2.1 用户功能需求	2
2.2 系统功能需求	3
2.3 性能需求	3
3 总体设计	4
3.1 系统的功能子系统设计	4
3.3.1 文件管理子系统	4
3.3.2 划痕和破损修复子系统	4
3.3.3 图像上色子系统	4
3.3.4 系统界面子系统	4
3.2 系统的使用方法	5
3.3 开发环境	6
3.3.1 系统开发平台	6
3.3.2 开源工具和源代码	6
4 详细设计	8
4.1 划痕检测——基于残差的改进 Unet.....	8
4.2 划痕修复——快速行进算法	9
4.3 图像上色——卷积神经网络对图像颜色空间的预测.....	10
4.4 人机交互界面——图形界面的扁平化设计	11
5 模型训练与系统实现	13
5.1 划痕识别网络	13
5.2 上色网络	13
5.3 文件子系统	14
5.4 UI 界面	14
6 系统特色与系统展示	17
6.1 系统特色	17
6.2 系统展示	17
7 总结	19

1 背景

照片，它是一种用来承载记忆的物品，是定格思恋的工具，其存在意义极大。随着现代科技的高速发展，摄像设备更新换代极快，清晰度高、色彩丰富的照片占据着生活的一部分，在朋友圈分享照片已成人们的生活常态。

然而，并不是所有的照片都能做到如同现在的照片一样清晰度极高、色彩极为丰富。事实上，现实生活中存在着诸多的老照片。这些老照片或多或少都存在着一些裂缝和损坏，并且存在着褪色的现象，甚至原本的颜色就是黑白。这样的黑白老照片不足承担“承载思念”的任务。因此，如何修复照片上的划痕和破损，如何对褪色照片或者黑白照片进行上色，是一个十分具有现实意义的问题。

传统的照片修复需要使用 PhotoShop 进行人工修复，这需要操作人员对 PhotoShop 这一工具具有极高的熟练度，才能完成老照片修复的任务。然而，这样修复的照片虽然效果很好，但因为对人工的高要求而导致费用也很贵。根据我们的调研，一张照片的人工 PhotoShop 修复的费用可能高达百元之多。还有一些修复方式是使用现有的 AI 平台提供的照片修复技术，例如百度 AI 的照片修复。这些平台提供了较好的照片修复技术，但也存在如下问题：

- 知名度不高、用户范围小（几乎只有 AI 领域内的人员会去使用）
- 无法兼顾修复和上色两个任务
- 未对老照片的修复进行定向优化
- 单次仅能处理单张照片，批处理过程复杂（批处理需要编程调用 API）
- 用户等待时间过长
- 需要付费

基于以上问题，我们打算构建我们的基于深度学习的老照片修复系统。利用人工智能技术，开发出一款能解决以上问题的老照片修复系统。在给予老照片修复服务的同时，宣传相关的深度学习技术。本项目就在这样的背景和想法下展开的。

2 需求分析

2.1 用户功能需求

系统的用户功能需求主要分为三个模块：图像修复模块，图像上色模块。下图是一个待修复图片的示例，我们希望针对诸如此类的照片，得到没有破损的、彩色的照片。



(1) 图像修复模块

图像修复模块的功能是完成图像的划痕修复。应该包含两个部分，即划痕的检测和划痕的修复。

对于照片上划痕的种类，我们考虑的是照片上的白色细划痕，对于年代久远的照片，这样的白色细划痕是最常见的。

划痕的检测要求我们将划痕从照片中标注出来，划痕的修复要求我们将划痕填补。该模块作为老照片修复的第一个步骤，其输入是色彩失真的、带有划痕的照片，输出是色彩失真的照片。在这个过程中，我们需要解决输入照片分辨率是不固定的问题。

(2) 图像上色模块

图像上色模块的功能是完成图像的上色。

对于色彩失真的类型，我们主要考虑的是两种，一种是灰度照片，另外一种掉色的照片。对于年代久远的老照片，这两种情况是最为常见的。

图像的上色要求我们根据图像的内容（即灰度图像）给出图像的色彩方案，这样的上色并非真实的，但我们要做的是尽可能模拟其真实性。该模块作为老照片修复的第二个步骤，其输入是色彩失真的照片，输出是修复好的照片。照片分辨率的不固定仍然是该阶段的一个重要问题。此外，对于上色而言，我们需要对老照片进行定向优化。通常而言，待修复老照片一般是人像照片，包括了任务的脸和衣着。

2.2 系统功能需求

系统的功能需求主要包含两方面：文件管理模块和图形化用户界面模块。

(1) 文件管理模块

文件管理模块主要是用于处理图像数据的输入输出，同时管理系统内部的深度学习模型。

对于图像的输入输出，需要做到能够识别用户计算机中的文件目录，并且读取文件目录。打开方式需要有打开单个图像文件和打开一个图像的文件夹这两种模式。并且，系统要求文件管理模块能够支持多种图片格式，包括但不限于 png 格式、jpg 格式。最后，需要支持对于图像的批处理功能。

对于深度学习模型的存放，需要合理放置各个深度学习模型。

(2) 图形化用户界面模块

图形化用户界面模块主要用于人机交互。

该模块需要设计清晰简便的人机交互界面，使得用户能够轻松地调用系统的全部功能。除了将系统的全部功能展示在图形界面上，图形化用户界面还需要将照片修复的结果展示出来，并且与修复之前的结果形成对比。最后，图形界面还需要包含一个欢迎页和开发者页。

2.3 性能需求

性能需求主要包括对系统所占存储空间的需求和对系统响应时间的需求。

(1) 存储容量需求

由于存在深度学习模块，该老照片修复系统将主要部署到云端服务器，三个部分的模型所占存储空间的大小应该限制在 5GB 以内。

(2) 系统响应时间需求

该系统将部署到云端服务器，在网速为 10M 的情况下，要求系统处理单张照片的响应时间满足以下要求：

- 单张图像的划痕修复：响应时间小于 0.5s
- 单张图像的图像上色：响应时间小于 0.5s
- 单张图像的序列化处理：响应时间小于 1.0s

对于批处理，需要系统给出等待提示，并且预估等待时间。

3 总体设计

3.1 系统的功能子系统设计

3.3.1 文件管理子系统

该子系统用于管理使用系统时产生的文件，当使用本系统进行图像处理时，会产生处理后的照片，并被该文件管理子系统进行保存留底，方便一段时间内用户进行回溯。同时，该文件管理系统负责日志管理，开发者或管理员可以根据日志查看该系统的运行情况。

3.3.2 划痕和破损修复子系统

该子系统总共分两个子模块。

数据处理模块，该模块用于将用户上传的照片或是文件管理系统中的文件处理成能够被该子系统处理的格式，即将照片处理成一个数据生成器，对每一张照片进行升维，转化为相应的张量形式。同时，对于神经网络模块的输出，该模块将其处理成可以被查看的图像格式，作为文件进行输出。

图像修复模块，该模块输入为经过数据处理后得到的数据生成器，经过神经网络的处理，输出一张新的二值图片，该结果用于获得原图中的划痕和破损的位置。而后，基于 opencv，根据原图和获得的划痕信息进行图像修复操作。本模块详细技术原理将在第三部分详细设计进行阐述。

3.3.3 图像上色子系统

该子系统总共分两个子模块。

数据处理模块，该模块用于将用户上传的照片或是文件管理系统中的文件处理成能够被该子系统处理的格式，即将照片处理成一个数据生成器，对每一张照片进行升维，转化为相应的张量形式。同时，对于神经网络模块的输出，该模块将其处理成可以被查看的图像格式，作为文件进行输出。

图像修复模块，该模块输入为经过数据处理后得到的数据生成器，经过神经网络的处理，输出一张新的彩色图片，本模块详细技术原理将在第三部分详细设计进行阐述。

3.3.4 系统界面子系统

系统界面子系统分为四个子模块。

“首页”界面，该界面大致介绍本应用的关键功能和使用方法，用户在打开应用时，将首先看到该界面内容，并根据该模块内容进行进一步的操作。

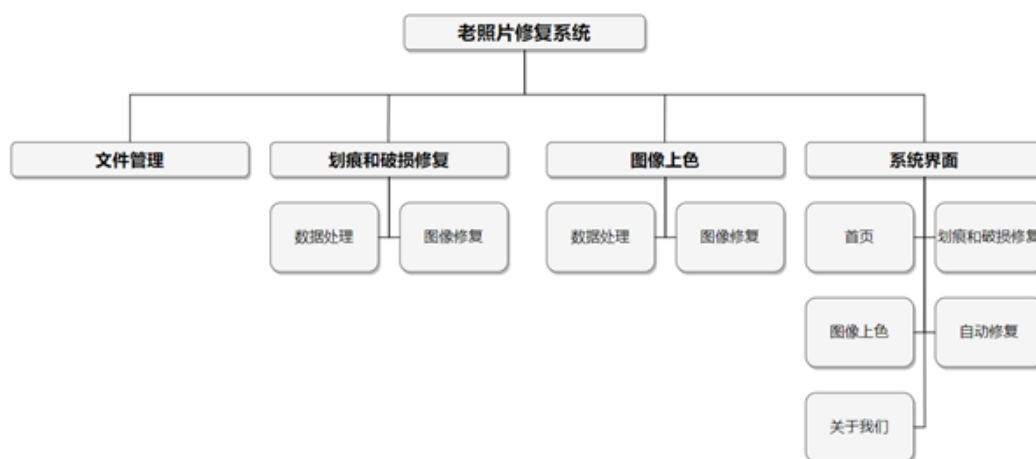
“划痕和破损修复”界面，该界面是用户和对应的子系统的接口，在该模块，用户可以选择需要处理的图片，界面上会显示用户选择的图片，用户可以点击修复按钮，即可调用修复子系统进行处理，完成后，对应的图片将会输出到界面上供用户参考，如果用户需要下载该图片，点击保存图片按钮即可选择保存路径。

“图像上色”界面，该模块是用户和对应的子系统的接口，在该模块，用户可以选择需要处理的图片，界面上会显示用户选择的图片，用户可以点击上色按钮，即可调用修复子系统进行修复，完成后，对应的图片将会输出到界面上供用户参考，如果用户需要下载该图片，点击保存图片按钮即可选择保存路径。

“自动修复”界面，该模块是用户和对应的子系统的接口，在该模块，用户可以选择需要处理的图片，界面上会显示用户选择的图片，用户可以点击自动处理按钮，即可调用修复子系统进行修复，完成后，对应的图片将会输出到界面上供用户参考，如果用户需要下载该图片，点击保存图片按钮即可选择保存路径。

“关于我们”界面，本模块给出该应用的版本信息，开发和运维人员，联系方式等信息，方便用户和开发者联系交流。

整个系统结构如下图所示：



3.2 系统的使用方法

对于用户而言，可以看到四个功能。

“首页”界面，该界面大致介绍本应用的关键功能和使用方法，用户在打开应用时，将首先看到该界面内容，并根据该模块内容进行进一步的操作。

“划痕和破损修复”界面，该界面是用户和对应的子系统的接口，在该模块，用户可以选择需要处理的图片，界面上会显示用户选择的图片，用户可以点击修复按钮，即可调用修复子系统进行修复，完成后，对应的图片将会输出到界面上供用户参考，如果用户需要下载该图片，点击保存图片按钮即可选择保存路径。

“图像上色”界面，该模块是用户和对应的子系统的接口，在该模块，用户可以选择需要处理的图片，界面上会显示用户选择的图片，用户可以点击上色按钮，即可调用修复子系统修复，完成后，对应的图片将会输出到界面上供用户参考，如果用户需要下载该图片，点击保存图片按钮即可选择保存路径。

“自动修复”界面，该模块是用户和对应的子系统的接口，在该模块，用户可以选择需要处理的图片，界面上会显示用户选择的图片，用户可以点击自动处理按钮，即可调用修复子系统修复，完成后，对应的图片将会输出到界面上供用户参考，如果用户需要下载该图片，点击保存图片按钮即可选择保存路径。

“关于我们”界面，本模块给出该应用的版本信息，开发和运维人员，联系方式等信息，方便用户和开发者联系交流。

3.3 开发环境

3.3.1 系统开发平台

- 操作系统：Windows 10 专业版，版本号：20H2 19042.867。
- 开源平台：Anaconda3，python。
- 硬件设备：CPU: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz-1.99 GHz; RAM: 8.00 GB; GPU: NVIDIA GeForce MX 150。

3.3.2 开源工具和源代码

本项目调用了开源 python 库：tensorflow, keras, pytorch, opencv, skimage, pyqt。

- TensorFlow: TensorFlow 是一个开源的、基于 Python 的机器学习框架，它由 Google 开发，并在图形分类、音频处理、推荐系统和自然语言处理等场景下有着丰富的应用，是

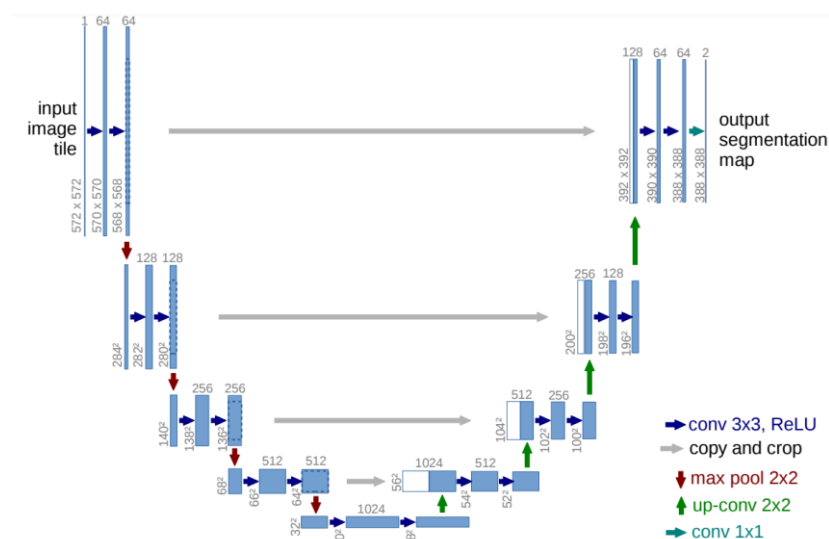
目前最热门的机器学习框架。本项目中使用该工具驱动 keras 库进行训练。

- **Keras:** Keras 是一个用 Python 编写的高级神经网络 API,它能够以 TensorFlow, CNTK, 或者 Theano 作为后端运行。Keras 的开发重点是支持快速的实验。本项目使用 Keras 完成神经网络的搭建和训练。
- **Pytorch:** PyTorch 是一个基于 Torch 的 Python 开源机器学习库,用于自然语言处理等应用程序。PyTorch 提供了两个高级功能: 1.具有强大的 GPU 加速的张量计算 2.包含自动求导系统的深度神经网络。除了 Facebook 之外, Twitter、GMU 和 Salesforce 等机构都采用了 PyTorch。
- **OpenCV:** OpenCV 是一个开源的跨平台计算机视觉和机器学习软件库,可以运行在 Linux、Windows、Android 和 MacOS 操作系统上。同时提供了 Python、Ruby、MATLAB 等语言的接口,实现了图像处理和计算机视觉方面的很多通用算法。本项目使用 OpenCV 进行图片的读取, 存储。并使用 OpenCV 自带的图像修复功能进行辅助修复。
- **Skimage:** Skimage 全称是 scikit-image SciKit, 它对 scipy.ndimage 进行了扩展, 提供了更多的图片处理功能。scikit-image 是基于 scipy 的一款图像处理包, 它将图片作为 numpy 数组进行处理, 正好与 matlab 一样。
- **Pyqt5:** PyQt5 是基于 Digia 公司强大的图形程式框架 Qt5 的 python 接口, 由一组 python 模块构成。PyQt5 本身拥有超过 620 个类和 6000 函数及方法。在可以运行于多个平台, 包括: Unix, Windows, and Mac OS。PyQt 是 Qt 最流行的 Python 绑定之一, 简单理解, 使用 Python 重新实现了一遍 Qt 的功能, 在实现的时候, 几乎保持了全部原有的 API。

4 详细设计

4.1 划痕检测——基于残差的改进 Unet

在划痕识别方面，我们使用了 Unet 网络的结构，如下图所示：



Unet 因为形状类似于 U 形而得名，具有编码器-解码器结构。左边是模型的编码器，负责特征提取，右边是模型的解码器，负责将对特征进行表示。与经典 Unet 网络不同之处在于，我们的模型对网络结构进行了改变，以适合划痕识别任务。模型的输入是 $256 \times 256 \times 3$ 的 RGB 图像。在所有的卷积层中，我们使用了 padding 方法让特征图的大小维持不变，这样在编码器和解码器之间的跳越连接时就不必进行裁剪。此外，在模型的输出层之后我们添加一个包含单个卷积核的 1×1 的卷积层，并且使用 sigmoid 激活函数，得到最终的 $256 \times 256 \times 1$ 的划痕二值图。

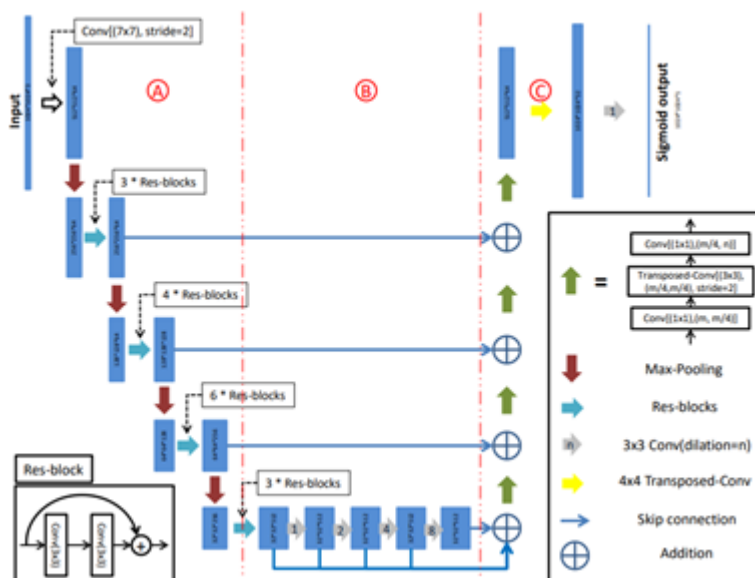
模型的输入和输出如下图所示：



其中白色部分即为图中的划痕。

传统的 Unet 在医学领域中主要用来分割肿瘤等占据图片中一片区域的内容，这与划痕仍然具有不同。具有残差结构的 Unet 网络被证明对眼球上血管的分割具有更好的效果。因此，在 Unet

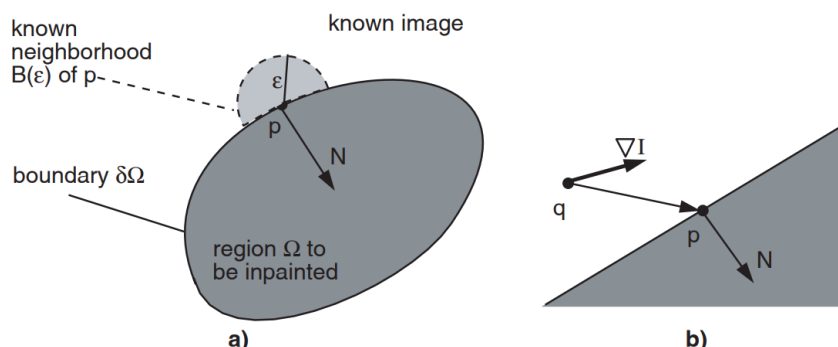
的基础上，我们也打算使用残差改进的 Unet，其结构如下：



残差改进的 Unet 所识别的划痕边缘更“圆滑”。

4.2 划痕修复——快速行进算法

在识别划痕的基础上，我们采用的数值方法来进行图像修复。修复算法采用的是快速行进算法。算法思想是：先处理待修复区域边缘上的像素点，然后层层向内推进，直到修复完所有的像素点。对于彩色图像，分别用同样的方法处理各个通道即可。



如上图所示，阴影区域是待修复的区域，要修复该区域中的像素，就需要计算出新的像素值来代替原值。现在假设 p 点是要修复的像素。以 p 为中心选取一个小邻域 B ，选取该邻域中已知的像素点来计算 p 点的像素：

$$I(p) = \frac{\sum_{q \in B_\varepsilon(p)} w(p, q) [I(q) + \nabla I(q)(p - q)]}{\sum_{q \in B_\varepsilon(p)} w(p, q)}$$

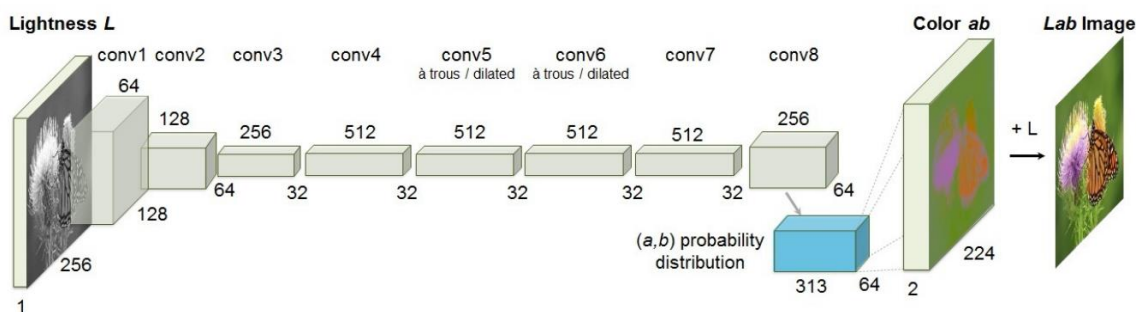
其中, w 是权值函数, 表示使用的像素 q 的权重。 $w(p, q) = dir(p, q) \times dst(p, q) \times lev(p, q)$, 表达式中三个式子计算方式如下:

$$\begin{aligned} dir(p, q) &= \frac{p - q}{\|p - q\|} \cdot N(p) \\ dst(p, q) &= \frac{d_0^2}{\|p - q\|^2} \\ lev(p, q) &= \frac{T_0}{1 + |T(p) - T(q)|} \end{aligned}$$

利用该算法, 从外向内把所有待修复区域全部修复即可。该算法实现起来非常简单, 快速, 和较慢的已知方法产生的结果几乎相同。

4.3 图像上色——卷积神经网络对图像颜色层的预测

照片上色任务的神经网络结构如下:



首先需要将照片转换为 LAB 格式。LAB 是由国际照明委员会 (CIE) 于 1976 年公布的一种色彩模式。是一个理论上包括了正常视力的人眼可以看见的所有色彩的色彩模式。Lab 颜色模式由三个通道组成, 它的一个通道是明度, 即 L; 另外两个是色彩通道, 用 A 和 B 来表示。A 通道包括的颜色是从深绿色 (低亮度值) 到灰色 (中亮度值) 再到亮粉红色 (高亮度值); B 通道则是从深蓝色 (低亮度值) 到灰色 (中亮度值) 再到黄色 (高亮度值)。因此, 这种色彩混合后将产生明亮的色彩。

神经网络的输入是该格式中 L 层的内, 即灰度图片。而 A、B 两个颜色层则作为模型的监督

信号来训练神经网络。在此基础上，照片上色的任务即是根据照片的内容，即照片的 L 层，预测照片的颜色层，即 A、B 层。通过 L 层映射得到 AB 层的内容，从而组合成彩色照片。而这个映射方法在这里就是深度学习的方法，即采用深层卷积神经网络。

网络的输入是 LAB 格式照片的 L 层的内容，很显然这里存在一个严重的问题，那就是传入照片的分辨率是大小不固定的，而一个神经网络只能处理分辨率大小为某一给定值的照片。为了解决这一分辨率不固定的问题，算法采用了双线性插值的方法。首先将照片 L 层的输入大小统一到 256×256 ，输入神经网络。神经网络的输出是 $2 \times 256 \times 256$ 的 AB 层的内容，然后对大小为 256×256 的 A、B 层进行像素级别的插值操作，采用双线性插值模型，即可得到分辨率大小和原始图片一样的颜色层。最后再与 L 层进行组合，即可得到原始分辨率大小的彩色照片。这一操作可能会导致照片上的一些细小细节被抹去，但却可以解决分辨率问题。

在神经网络的结构方面，神经网络的结构十分简单，它使用卷积层和池化层来提取一些照片上的结构信息，并根据这些信息来推断 A、B 层。在经过 8 次的卷积、池化和非线性层后，算法得到了 L 层的特征图，特征图通过卷积和 Softmax 激活函数得到 A、B 层的概率分布，最后利用卷积和上采样操作得到 A、B 层的预测。

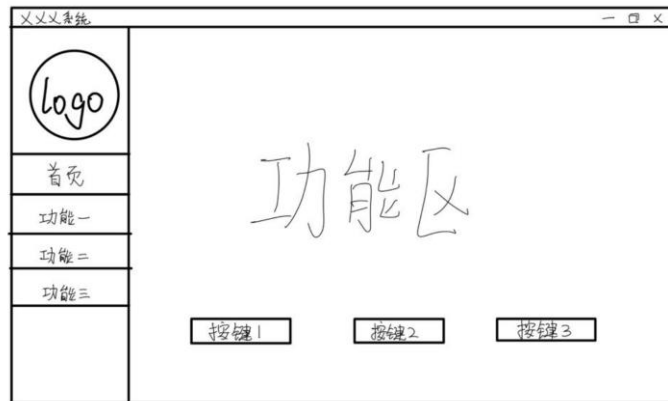
4.4 人机交互界面——图形界面的扁平化设计

在人机交互界面上，我们采用扁平化的设计理念。扁平化概念的核心意义是：去除冗余、厚重和繁杂的装饰效果。而具体表现在去掉了多余的透视、纹理、渐变以及能做出 3D 效果的元素，这样可以让“信息”本身重新作为核心被凸显出来。同时在设计元素上，则强调了抽象、极简和符号化。

扁平化软件设计直接体现在更少的按钮和选项，这样使得 UI 界面变得更加干净整齐，使用起来格外简洁，从而带给用户更加良好的操作体验。因为可以更加简单直接地将信息和事物的工作方式展示出来，所以可以有效减少认知障碍的产生。

结合我们系统的三个主要子功能，即照片修复、照片上色、综合修复，我们拟设计切换栏实现不同功能间的切换。在每个功能子页面上，需要展示出相应的功能子界面，配上简单、直观、清晰的功能按钮，从而满足用户的使用。

人机交互界面的设计图如下：



装

订

线

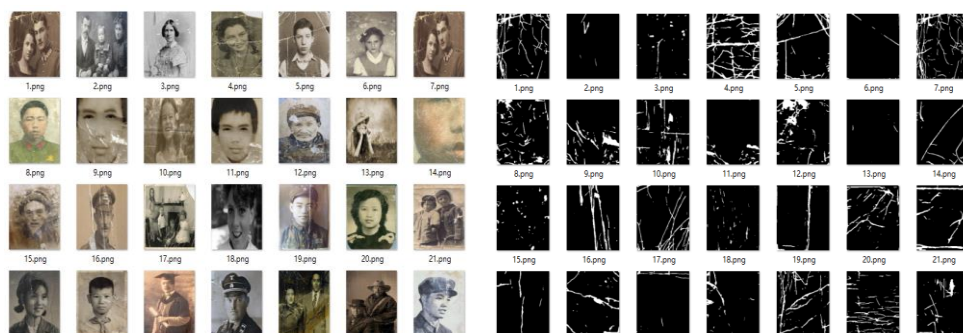
5 模型训练与系统实现

5.1 划痕识别网络

5.1.1 数据集

数据集方面,划痕识别网络使用了自制数据集。我们从网络上收集 100 张带有划痕的老照片,在简单处理之后,得到原始数据集。而对于数据集的标注,我们采用 OpenCV 中的库识别得到。

下图展示了数据集的部分照片和标注:



5.1.2 模型训练

模型使用输出层的二值交叉熵损失作为损失函数,使用 Adam 优化器,学习率为 $1e-4$,训练 30 个 epoch。

5.2 上色网络

5.2.1 数据集

数据集包含两个: ImageNet 和 Flickr8k。

(1) ImageNet

ImageNet 数据集是一个计算机视觉数据集,是由斯坦福大学的李飞飞教授带领创建。该数据集包含 14197122 张图片和 21,841 个索引。图像上色网络在该数据集上进行预训练。

(2) Flickr8k

Flickr8k 是一个以人像为主、真实场景为辅的数据集,包含 8000 张图片。图像上色网络在该

数据集上进行迁移学习。

5.2.2 模型训练

在 ImageNet 数据集上的预训练采用现有的模型参数。

在 Flickr8k 数据集上的迁移学习，模型使用 A、B 层的 MSE 损失作为损失函数，使用 Adam 优化器，学习率为 $1e-6$ ，训练 20 个 epoch。

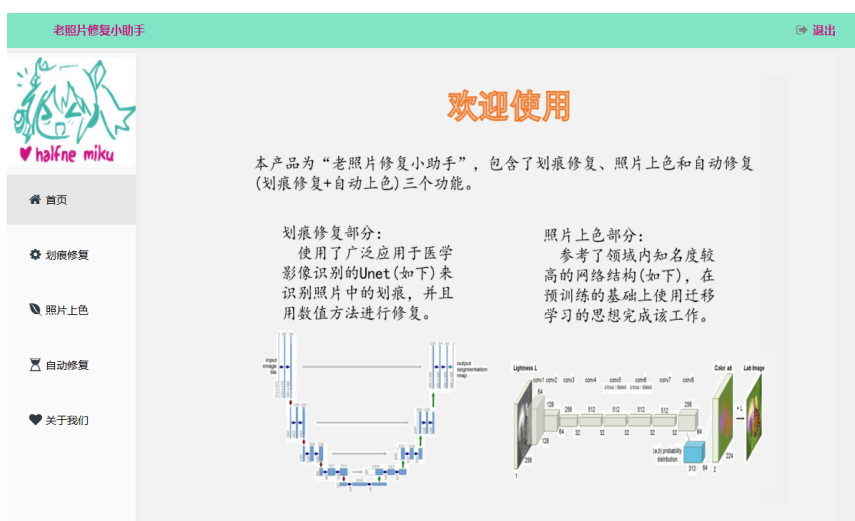
5.3 文件子系统

文件子系统需要做到读取计算机本地的文件和目录，打开图片文件，并且将生成的照片保存到本地。此处我们使用 PyQt5 的 file 库来实现该功能。

5.4 UI 界面

根据设计需求，我们使用 PyQt5 实现以下界面：

(1) 首页



首页界面介绍了我们程序所采用的主要技术。在划痕修复上，我们采用了在医学影像识别领域运用较广的 Unet，并且在次基础上对网络结构进行改进，使用 Resnet 的思想，从具有划痕的照片中识别出划痕，然后再使用数值方法进行划痕修复；在照片上色方面，我们参考了 CVPR2016

的一个网络结构，在预训练的基础上，使用迁移学习的思想，使得对于人像的上色结果会比原来更加好一些。

(2) 划痕修复



“划痕修复”界面如上，这一界面由两个图片显示框和三个操作按钮组成。界面的使用方法较为简易，只需经过三步即可“选择图片——>划痕修复——>保存照片”。“选择图片”按钮将打开对话框，从本地选择一张图片，打开之后图片将被放在左边的照片框；点击“划痕修复”按钮后，将执行划痕修复操作，根据被修复图片的大小，这一过程将会需要几秒，请稍加耐心等待这一过程，执行修复完成后，修复好的图片将被放在右边的图片显示框中；最后点击“保存图片”按钮，即可将修复好的图片保存在需要的目录下。

(3) 图像上色



“照片上色”界面如上，这一界面由两个图片显示框和三个操作按钮组成。界面的使用方法较为简易，只需经过三步即可“选择图片——>照片上色——>保存图片”。“选择图片”按钮将打开对话框，从本地选择一张图片，打开之后图片将被放在左边的照片框；点击“照片上色”按钮后，将执行照片上色操作，根据被上色图片的大小，这一过程将会需要几秒，请稍加耐心等待这一过程，执行上色完成后，上色完的图片将被放在右边的图片显示框中；最后点击“保存图片”按钮，即可将上色完的图片保存在需要的目录下。

（4）自动修复



“自动修复”界面如上，这一界面同样由两个图片显示框和三个操作按钮组成。另外“自动修复”即为将修复与上色同时进行，前两个界面将修复与上色分离开是为了满足用户的特定需求。

界面的使用方法较为简易，只需经过三步即可“选择图片——>自动修复——>保存图片”。

“选择图片”按钮将打开对话框，从本地选择一张图片，打开之后图片将被放在左边的照片框；点击“自动修复”按钮后，将执行照片的自动修复操作，包括划痕修复与上色，这一过程将会需要几秒，请稍加耐心等待这一过程，执行自动修复完成后，图片将被放在右边的图片显示框中；最后点击“保存图片”按钮，即可将自动修复完的图片保存在需要的目录下。

6 系统特色与系统展示

6.1 系统特色

(1) 创意方面

本项目实现了利用人工智能对老照片进行处理的功能。一直以来，关于老照片的处理呈现出一种分裂的状态，一方面在实际生活中，老照片通常通过人工操作，使用 Photoshop 等软件进行修复；另一方面，理论上基于深度学习方法的照片处理技术日益强大，在理论上表现出了强大的能力，本作品就是一个将科研上获得的成果落地到实际应用的产物。这将极大方便非相关领域专业人士对图像处理的操作。

(2) 开发实现

本作品的同时实现了前端和后端的开发，两者建立连接，与目前主流实验室成果不同的是，用户可以不再直接通过代码指令对程序进行控制，简单点击用户界面的按钮，即可调用程序的功能，十分方便。同时，本应用成功开发出一款完整的应用软件层次的作品，与用于演示科研结果的简单可视化界面不同的是，该应用充分考虑到了软件的升级，运维，管理的相关事宜，是一款具有强可扩展性和可维护性的应用。

(3) 应用层面

目前实验室得到的图像处理结果可以得到比较好的结果，但并不考虑运行成本。通常，实验室的程序只能在配置非常高的环境下使用，也有的只能在特定数据上进行使用，扩展性和通用性并不理想。为了实现性能和配置之间的平衡，本应用在设计过程中充分考虑到了机器的配置，力求使用较少的资源达到较高的效果。在搭建神经网络的过程中，本应用将一个完整的图像处理任务分成两个子任务，分别使用两个独立的神经网络，大大降低了网络对配置的要求，在图像修复方面，介于当前基于深度学习的图像修复对配置要求极高，普通家用机基本不可能能够用这些程序修复大型图片，本应用采用“识别-修复”两步骤降低了算法对机器的要求。达到了满意的运行速度和效果。

6.2 系统展示

(1) 划痕修复



修复前



修复后

(2) 图像上色



上色前



上色后

(3) 综合修复



修复前



修复后

7 总结

7.1 总结

本项目利用两个结构简单神经网络和一种数值方法，完成了老照片修复这一任务。老照片的修复需要达成的效果自然是越逼真越好，“以假乱真”自然是最理想的结果。我们的照片修复算法在图片磨损程度小的情况下取得了令人满意的效果：在划痕修复方面，对于细划痕的修复和面积较小的破损效果十分好；在照片上色方面，对于风景图像的上色效果十分好，能够做到“以假乱真”的最终目标。

7.2 不足与改进

本项目存在一些不足：

1. 本项目主要追求的是算法的简便和在边缘设备上的推理效率，一定程度上忽略了修复效果和上色效果。在修复任务上，生成对抗网络（GAN）和扩散模型（Diffusion）在近些年取得了较大成功；在上色任务上，基于 Transformer 的上色网络也有更逼真的效果。
2. 本项目所使用的数据集具有较大的局限性。划痕识别的数据集仅包含 100 张图片，是一个规模很小的数据集；上色网络的两个数据集与老照片数据之间存在着明显的数据漂移现象。因此，为了让系统有更好的表现，需要引入模型学习少量数据、模型学习跨领域数据的方法。
3. 系统构建方面存在不足。项目的系统是使用 pyqt5 开发的本地项目，限制了其使用范围，可以考虑开发网页应用，并部署到服务器上。并且，该系统仅支持单次处理一张照片，不支持批处理功能，可以添加此功能，使得系统更加实用。此外，可以增加在线学习功能，对新来的、发生概念漂移/数据漂移的老照片数据加以学习和训练，从而不断提升模型的泛化能力。

8 参考文献

- [1] Ronneberger, O, P. Fischer , and T. Brox . "U-Net: Convolutional Networks for Biomedical Image Segmentation." International Conference on Medical Image Computing and Computer-Assisted Intervention Springer International Publishing, 2015.
- [2] Zhang, Richard, Phillip Isola, and Alexei A. Efros. "Colorful image colorization." European conference on computer vision. Springer, Cham, 2016.
- [3] Telea, Alexandru. "An image inpainting technique based on the fast marching method." Journal of graphics tools 9.1 (2004): 23-34.

装

订

线