时光修补坊，组员：苏珈磊，薛皓林

1、任务概述

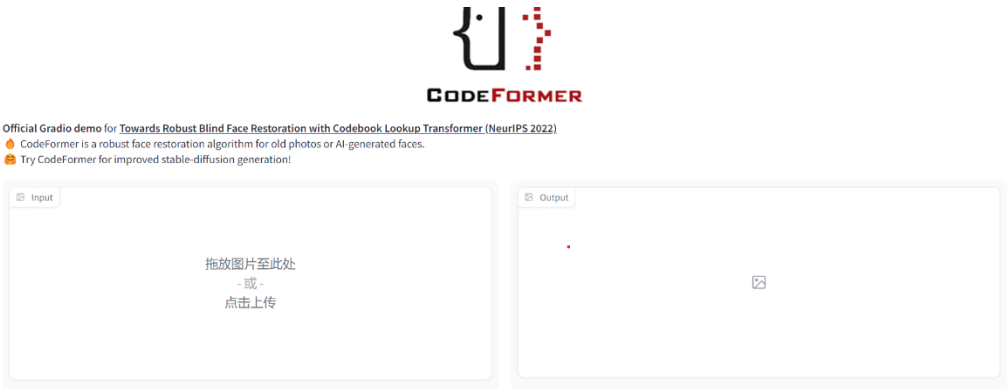
1.1、目标

(1)背景

人类，作为群居动物，不仅在物质生活上相互依赖，更在精神层面彼此支撑。照片，这一凝固时光的艺术形式，承载着我们的记忆，成为精神生活中不可或缺的一部分。它们记录着我们的欢笑与泪水，见证着我们的成长与变迁，让我们的精神世界因此变得更加丰富多彩。但过去的摄影技术尚未成熟，所拍摄的照片像素低、模糊不清，甚至仅能以黑白呈现。岁月流转，往昔的回忆也在悄然褪色。更因照片本身的不清晰，那些欢乐的时光愈发难以在脑海中重现。因此，我们迫切需要一种能够修复老照片的方法，让逐渐模糊的回忆重新焕发生机，变得清晰如初。

(2)与其他相关软件的关系

目前市面上的老照片修复网站，其功能相对单一，仅仅局限于老照片的修复，界面设计也相对简单，如 CodeFormer 和 RestorePhotos.io 这两个网站便是其中的代表。它们主要聚焦于功能的实现，界面单调，功能单一，而在提升用户体验方面却显得力不从心。



因此，我们计划打造一个更加强大的网站，它不仅利用超分模型进行老照片修复，更在用户体验上进行了深度优化，并突破了单一功能的限制。具体来说，

我们将在原有修复功能的基础上，通过改进现有的代码模型，引入一项创新功能——让照片动起来。想象一下，那些尘封已久的照片中的人物能够微笑眨眼，仿佛穿越了时空，回到了我们身边。这不仅能使照片变得更加生动有趣，更能唤起用户内心深处的美好回忆。

我们致力于为用户提供更为卓越的用户体验，让每一个用户都能在这个平台上找到属于自己的那份珍贵记忆，重温那份久违的感动。

(3)该项软件开发的应用目标将会包括但不局限于：

老照片像素优化：使以前的照片更加清晰

黑白照片上色：使黑白照片回复正常色彩

照片动起来：使上传的照片可以实现眨眼，微笑等简单动作(创新功能)

用户 ID 记录：记录下用户的 ID，保存用户的记录(创新功能)

1.2、系统（或用户）的特点

(1)软件技术特点：

利用云服务器实现图片的超分加强处理

利用 vue 和 Javascript 搭建用户交互网站，python 语言编写程序后端修复程序.

利用 GFPGAN 模型实现照片的超分处理，利用 paddleGAN 模型实现照片动起来

MySQL 数据库编程及储存用户信息

(2)网站应用特点：

方便快捷：使用手机电脑都行，随时随地

照片灵动：使照片中的人物动起来，更加生动，功能更加有趣

账户信息储存：用户有自己的账号密码，储存在云数据库里，以及用户的适用记录

网站界面：更加美观，交互界面更好。

(3)用户特点：

1. **怀旧情感强烈**：使用老照片修复系统的用户往往对过去有着深厚的情感。他们可能希望将那些因时间流逝而损坏或模糊的老照片进行修复，以重新唤醒那些美好的记忆。这些照片对他们来说可能具有极高的情感价值，代

表着家庭、友情或某个特定时期的记忆。

2. **中老年人群为主**：由于老照片往往属于较早的年代，因此修复老照片系统的用户群体以中老年人为主。这部分人群可能不太熟悉现代科技，但对修复老照片有着强烈的需求。因此，系统需要设计得简单易懂，便于他们操作。但系统界面也应该年轻化一点，这样会有更多年轻人来修复或者帮助父母修复老照片。
3. **对品质有要求**：用户在使用老照片修复系统时，通常对修复效果有较高的期望。他们希望修复后的照片能够尽可能还原原貌，无论是色彩、清晰度还是细节都应当尽可能达到最佳效果。
4. **追求个性化体验**：除了基本的修复功能外，用户还可能希望系统能够提供更多的个性化选项，如选择不同的修复风格、调整修复程度等。这样可以满足不同用户对于修复效果的个性化需求。

(4)使用频度：本产品的主要功能是修复老照片，用户使用时间并不大，可能耗费的时间就是在上传照片和在处理过程的等待时间。预计普通用户不会耗费过多的时间停留在该网站，修复老照片的需求可能是一次性或者“突发的事件”，对于专业修复照片的人员，使用的频度可能会变多。在大型的档案修复工程或者遗产保存功能等等重大功能中，使用频度都会很高。

1.3、假定和约束

（列出进行本软件开发工作的假定和约束，例如开发期限等。）

前期（2 周）：对所选题目做充分构思与准备工作

中期（4 周）：搭建出修复老照片系统外包网站的雏形，完成修复老照片系统的基本功能，对老照片进行修复，网站开发等。

后期（3 周）：进行优化与调试工作，使该系统鲁棒性更高，功能更加完善且成熟，同时完成创新功能，比如让老照片中的人物实现微笑等动作，用户 ID 登录，使用记录存储(历史记录)等一系列功能，同时要保证好用户数据(照片)的隐私性和安全性，不能泄露用户的信息。

2、需求规定

2.1、软件功能说明

（逐项定量和定性地从业务角度叙述所提出的功能要求，说明输入什么量、经怎样的处理、得到什么输出，说明产品的容量，包括系统应支持的终端数和应支持的并行操作的用户数等指标。）

(1)**账户注册**

定性：用户 ID 和密码受到长度限制，要求 ID 长度小于等于 10 个字符，密码长度小于等于 16 个字符

定量：用户 ID 不得含有违禁词且已存在的用户 ID 不得再次注册

说明：用户输入的 ID 和密码在被判定符合以上定性、定量要求的前提下可以注册新账户，其 ID 和密码将被传输并保存在数据库中

(2)**账户登录**

定性：要求用户输入 ID 与密码已经存在于数据库中且对应正确才能登录

定量：无

说明：用户输入的 ID 和密码在被判定正确后可以登录到个人账户

(3)**密码找回**

定性：输入密保信息正确

定量：新密码同样要求符合上述密码长度限制

说明：用户点击找回密码通过密保验证后可以重新设置账户密码，新密码将替换数据库中该用户 ID 对应的原密码

(4)**照片修复**

定性：上传用户本地的图片

定量：一次修复一张图片

说明：上传图片后，点击开始修复，等待几分钟后会在相应地方显示已经修复好的图片

(5)**照片人物动起来**

定性：上传需要动起来的照片

定量：一次一张

说明：上传图片后，点击开始，等待几分钟后会在相应地方显示已经更改好的图片

2.2、对功能的一般性规定

（本处仅列出对开发产品的所有功能（或一部分）的共同要求，如要求界面格式统一，统一的错误声音提示，要求有在线帮助等。）

为提高系统的鲁棒性，系统会对用户的异常操作抛出提示信息

字符编码格式：UTF-8

规定 Administrator 为管理员 ID，用户可向管理员咨询以获取在线帮助

2.3、 用户界面

将软件功能和用户体验相互匹配，以确保用户能高效，便捷快速地使用网站进行老照片的修复。

（1）直观的设计理念

网页应该简洁，直观，避免不必要的复杂性，图标和菜单应该标准化，确保即使是用户第一次使用也能快速明确如何使用该网页。

（2）易于导航

提供一个明显的导航系统，例如导航条或者菜单等，方便用户访问不同的功能区域，例如上传图片 编辑格式 上传和分享等。同时网页应该支持回退网页的操作，以促进无误的多步操作。

（3）响应式布局

布局需要优化以用来适应不同大小的屏幕，例如笔记本电脑，平板电脑，智能手机等等。用户应该能够在任意一台智能设备上使用该网页的完整功能。

（4）图像预览和处理

提供高清的图像预览，用户可以在修复的过程中，轻松的查看细节，界面可以支持简单的和高级的编辑工具，清楚的表示每个工具的功能。

（5）交互式教程和帮助

对于新用户，提供交互式教程和引导功能，帮助快速上手，也可以提供帮助文

档或者 FAQ。

（6）上传或者下载过程

确保上传和下载过程简单，便捷提供清晰的进度指示和完成通知。

（7）（拟定）多语言支持

根据市场目标，可能需要支持多种语言以支持和吸引更加广泛的用户群体。

（8）错误处理和反馈

再出现错误或者需要用户注意的情况下提供清楚及时的反馈，拟定对常见错误提供快速修复的建议。

（9）图标品牌和美术设计

网站外包的视觉元素应该现代化并且吸引人，提供愉悦的用户体验。

2.4、对性能的一般性规定

2.4.1、精度

用户 ID、密码、传上去的图片格式等都受到一定的限制。

2.4.2、时间特性要求

拟规定时间内正常的修复老照片，并且实现照片中的人物的动态效果。

2.4.3、输入输出要求

输入的图片格式支持 jpg 或者 png 模式，输出图片格式 jpg 或者 png 格式。照片动态修复的输出格式为.mp4 格式(视频).

2.5、数据管理能力要求

（1） 图片存储

需要为每个用户存储上传的原始图片和修复后的图片，每张图片的大小，可能会因为分辨率而异，根据目标市场和用户基础预计，每个用户可能会有平均上传 5 到 20 张图片进行修复。假设图片平均大小为 5MB，如果系统有 1 万名活跃用户，没人上传平均 10 张照片，则系统

需要处理大约 500GB 的图片数据。

（2） 文卷和记录

系统将保留所有用户的操作记录，包括上传编辑和下载的操作。可能还需要存储各种设置调整，使用的工具和滤镜等详细数据，对于每次操作，可以预计记录大小应该在几 KB 到几十 KB 不等。

（3） 数据库表

用户信息表应该包含用户的用户名，密码，联系方式等字段，每条记录的大小预计在 1KB 左右，图片信息表至少需要保存图片的 ID，用户 ID，上传时间，文件路径等相关信息，每条记录预计小于 1KB

（4） 增长预测

根据市场推广计划和用户增长趋势，可以对数据量的未来增长进行预测，假设每年用户增长率为 10%，预计数据增长将大致与此相同。

（5） 存储估算（拟定）

基于当前图片大小和用户操作记录，创建一个估算模型，并且考虑未来三到五年内的数据增长需要充分考虑数据备份和冗余存储的要求，确保数据的稳定性和系统的高可用性。

（6） 数据安全和隐私

数据管理还包括合规性考量，必须符合所有适用的数据保护法规，例如 GDPR 或类似法规。

（7） 性能要求

（8） 确保数据库设计支持快速查询和存储，以便于用户体验流畅，需根据预计的并发用户数，优化数据库性能。

2.6、故障处理要求

（列出可能的软件、硬件故障以及对各项性能而言所产生的后果和对故障处理的要求。）

正如项目管理老师所说，风险规划是项目推进过程中十分重要的一环，提前预测可能发生的软硬件故障并做出预案不仅能使开发过程更顺利还能在一定程度上降低用户遇到系统性故障的可能性进而提升用户体验。

1.软件故障可能包括:

（1） 应用程序崩溃-由于代码缺陷、资源泄漏或者第三方库的问题导致的服务中断。

后果	用户工作流程中断,数据丢失,用户满意度下降
处理要求	自动错误报告，快速错误响应和修复，(拟定)在线升级和补丁发布.

（2） 数据库故障-数据库服务不可用，查询执行异常导致的数据访问失败。

后果	无法存储图片和用户信息，业务操作无法进行
处理要求	定期备份,故障转移策略,专业数据库恢复服务.

（3） 网络问题-通信中断或延迟影响的服务器响应时间。

后果	服务无响应,增加加载时间,数据同步问题.
处理要求	冗余网络连接,使用网络监控.

（4） 安全漏洞-黑客攻击或恶意软件感染系统。

后果	数据泄露,服务破坏,信誉损害.
处理要求	实时安全修复,紧急响应计划(拟定)

2.硬件故障可能包括:

（1）服务器故障-物理服务器组件（如硬盘、内存、电源）故障。

后果	数据损失,服务中断,长时间未响应
处理要求	冗余服务器，定期维护，替换旧件

（2）计算机存储设备等硬件故障-硬盘或 ssd 损坏导致的数据丢失。

后果	历史数据丢失，恢复时间长.
处理要求	定期备份，数据迁移.

（3）电力供应中断-电网故障或不稳定的电源导致的突然电力中断。

后果	硬件损坏，服务中断
处理要求	高级电源支持

关键故障处理策略:

监控和警告，自动化故障恢复，冗余设计，定期维护和测试，技术支持.

2.7、其他专门要求

（如用户对安全保密的要求,包括信息加密、信息认证方面的要求;对使用方便的要求,对可维护性、可补充性、易读性、可靠性、运行环境可转换性的特殊要求等。）

为确保用户信息安全性和规范性。首先，注册时就要求用户 ID 和密码符合规范，其次，登录时要求用户输入 ID 与密码已经存在于数据库中且对应正确才能登录到账户。

并且，我们人性化地设置了重置密码的功能，用户点击找回密码通过密保验证后可以重新设置账户密码。

目前可能面临的问题是用户收发信息的安全性。在老师的提醒下，我们注意到用户所收发的信息都会在服务器中留下记录,这相对用户确实是一个安全隐患。我们可能会采用 md5 加密/解密的方式更进一步地保障用户信息的安全性。

同时程序还要具有可维护性和扩展性的要求，因此我们采用了模块化架构，系统设计具有高度的模块化，方便未来功能的添加，修改和删除.同时程序代码中添加了注释文档，具有详尽的代码注释和开发文档，以便于我们技术人员进行维护。同时我们编写的代码具有易读性和可靠性，遵循了清晰的编码标准和相应的命名规定，以用来增加代码的可读性和维护性.

3、运行环境规定

3.1、设备

我们暂且计划用两台电脑进行模拟，一台电脑模拟技术处理的后端，另一台电脑模拟使用用户。(前期暂定使用本地服务端 localhost)

3.2、支撑软件

操作系统：Windows 客户端

编译程序： nodejs npm 开发环境

数据库管理系统：MySQL

测试支持软件：Pycharm vscode Webstorm

我们的电脑自身是 Windows 系统，同时编写代码时前端和后端都需要进行编

写。因此我们采用 Pycharm 来运行和调试后端的 python 代码，使用 vscode 和 Webstorm 来运行和调试前端的 Vue 代码.再利用 api 端口进行前后端连接，后期还可能会引入数据库来进行信息的相应存储.

3.3、接口

前端网页编写代码和后端逻辑处理代码之间存在接口，数据库调用上存在接口，文件上传和下载存在接口,

3.4、控制

用户从前端上传照片，传送到后端进行逻辑处理，数据库进行存储,用户然后从前端获取处理结束的图片或者视频。

4、尚需解决的问题

1. 不清楚如何对数据库关系表结构做优化以提高查找性能.
2. 对于用户数据的完全性不清楚如何加密保存，采用何种加密方式.
3. 由于硬件的限制，不知道不同设备和平台之间是否兼容.
4. 未经众多用户图片检测与测试，图像修复算法的高效性与准确性无法完全保证.
5. 数据库的调用和与前端后端代码的完全配合处理方式还有待实现.