人工智能系统测试期末论文

学院：电子与信息工程学院

专业：计算机科学与技术

学号：2230771

姓名：包广垠

完成日期：2023 年 1 月 2 日

摘要

本报告为同济大学研究生课程《人工智能系统模型评估》的课程大作业。本文以一个简单的人工智能系统——老照片修复系统为例，阐述并说明了人工智能系统的生命周期。侧重于生命周期中的数据一致性判别、模型性能评估、模型调优、项目风险评估和系统测试方法。为人工智能系统生命周期的研究提供了一个较为完整的案例。

关键词： 人工智能系统，生命周期，数据一致性，风险评估，人工智能系统评估

装

订

线

目 录

装 订 线

装 订 线

1 老照片修复系统的概述

1.1 项目背景

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。

1.2 需求分析

1.2.1 用户功能需求

系统的用户功能需求主要分为两个模块：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，另外一种是掉色的照片。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，这样的上色并非是真实的，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。

1.2.2 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。

1.2.3 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 3GB 以内。

（2）系统响应时间需求

该系统将部署到边缘设备，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。

1.3 系统描述

基于深度学习的老照片修复系统是一个综合了图像划痕修复、图像上色、图像超分辨率的人工智能系统。在功能上，系统将对破损的、色彩失真的老照片进行三阶段序列化处理，得到清晰完整色彩丰富的高质量照片；在人机交互上，系统界面简单，便于用户的使用，响应及时，还支持批处理。

总而言之，这是一个要素齐全、功能完善、实用性强、交互性强的人工智能系统。

2 系统的人工智能组件

系统的人工智能组件主要包含两个：划痕检测网络和图像上色网络。

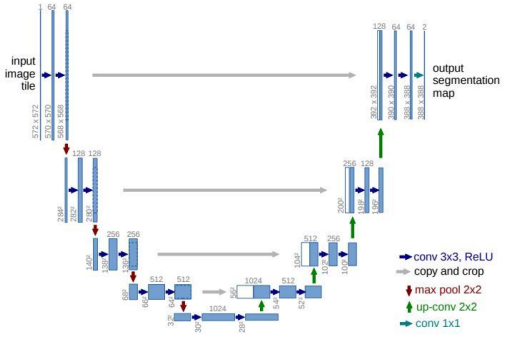
2.1 划痕检测模块

划痕检测任务本质上是一个显著性目标检测任务（salient object detection，SOD），即识别图像中最为显著的区域，将目标在像素级别分割出来。在划痕检测中，我们需要检测的显著性目标即为照片上的划痕。

下图展示了一个示例。

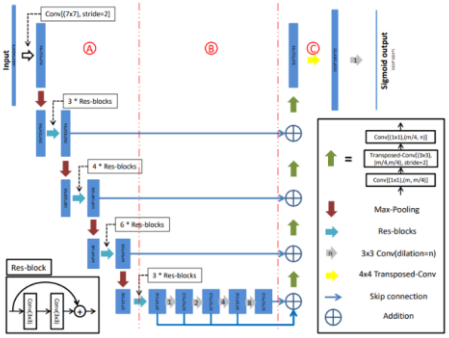
对于左图所示的带划痕的照片，我们希望能够将其划痕部分识别出来，并在像素级别进行标注。如右图所示，我们使用黑白二值图来标注图像中的划痕部分。

划痕识别可以使用显著性目标检测领域的经典网络结构 Unet，该网络可以实现端到端的像素级显著目标分割，其网络结构如下图所示：



Unet 因为其形状与字母“U”相似而得名。网络的输入端通过卷积与池化不断提取特征，并压缩特征图的维度；网络的输出端通过反卷积提高特征图的维度，做到与输入端逐一匹配。在特征图维度相同的输入端和输出端之间有跳越连接层，防止特征的消散。这样的网络具有提取图像中显著特征的能力。

在原始 Unet 的基础上，我们结合 ResNet 的思想，加入了跨层连接模块，这使得模型能够更加稳定的收敛和识别出更加准确的划痕。改进后的网络结构如下图所示：

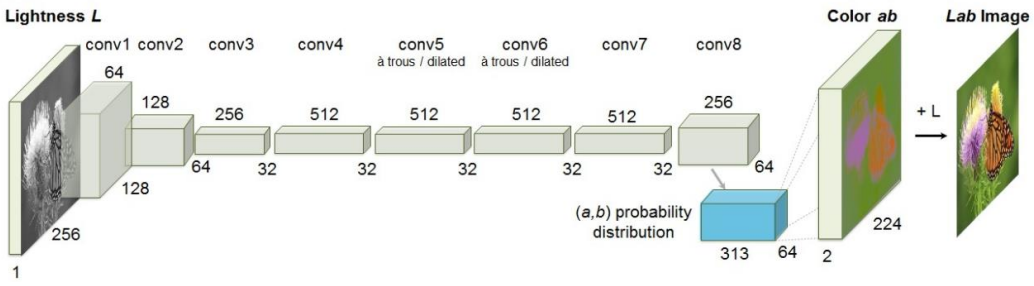


2.2 图像上色模块

图像上色模块可以看作是根据图像的内容预测图像的颜色。

在输入神经网络之前，需要将图片由 RGB 颜色空间转化为 LAB 颜色空间。在 LAB 颜色空间中，L 表示图像的亮度，即图像的灰度表示，A 和 B 是两个颜色层，分别表示由红色向绿色的转化度和由绿色向蓝色的转化度。在 LAB 颜色空间中，图像上色的任务即为根据图像的 L 层去预测图像的 A 层和 B 层。

图像上色的网络结构如下：



网络通过对图像的 L 层进行卷积操作，得到一个特征图。利用该特征图通过全连接层去表示图像的颜色层，从而实现对灰度照片的自动上色。

装 订 线

2.3 人工智能组件的评价指标

评价指标对于神经网络模型而言十分重要，下面介绍两个任务各自的评价指标。

2.3.1 语义分割

语义分割的常用评价指标如下：

（1）F-score

F-score 是统计学中用来衡量二分类模型精确度的一种指标，它同时兼顾了分类模型的准确率和召回率。F-score 是模型准确率和召回率的一种加权平均，其最大值是 1，最小值是 0，并且值越大意味着模型越好。

首先需要定义二分类的四种情况：

预测\观测	正样本	负样本
正样本	真阳性 TP	假阳性 FP
负样本	假阴性 FN	真阴性 TN

则 Precision 的定义如下：

$$precision = \frac{TP}{TP + FP}$$

Recall 的定义如下：

$$recall = \frac{TP}{TP + FN}$$

根据准确率与召回率，可以定义 F-score：

$$F_\beta = \frac{(1 + \beta^2) \times precision \times recall}{\beta^2 \times precision + recall}$$

（2）基于像素的 accuracy 准确率

假定有 k+1 个类别（k 个目标类和 1 个背景类），p_{ij}表示为第i类却预测为第j类。则可以定义如下准确率：

- Pixel Accuracy (PA)

  PA 表示分类正确的像素点数和所有的像素点数的比例，其计算公式为：

$$PA = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k} \sum_{j=0}^{k} p_{ij}}$$

- Mean Pixel Accuracy (MPA)

计算每一类分类正确的像素点数和该类的所有像素点数的比例然后求平均：

$$MPA = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}}$$

- Mean Intersection over Union (MIoU)

  计算每一类的 IoU 然后求平均：

$$MIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}+\sum_{j=0}^{k}p_{ji}-p_{ii}}$$

2.3.2 图像上色

图像上色任务主要有三个评价标准：

（1）视觉真实性(AMT)

视觉真实性类似于图灵测试。为了衡量图像真实程度，该测试需要随机给出一组图片（其中包含真实图片和神经网络着色后的图片），每次呈现来源不同的两张图片，让人类受试者从中选择他们认为包含不自然颜色的图片，最后计算神经网络图片上色模型的识别率。该测试需要消耗大量的人力资源。

（2）原始准确率(AuC)

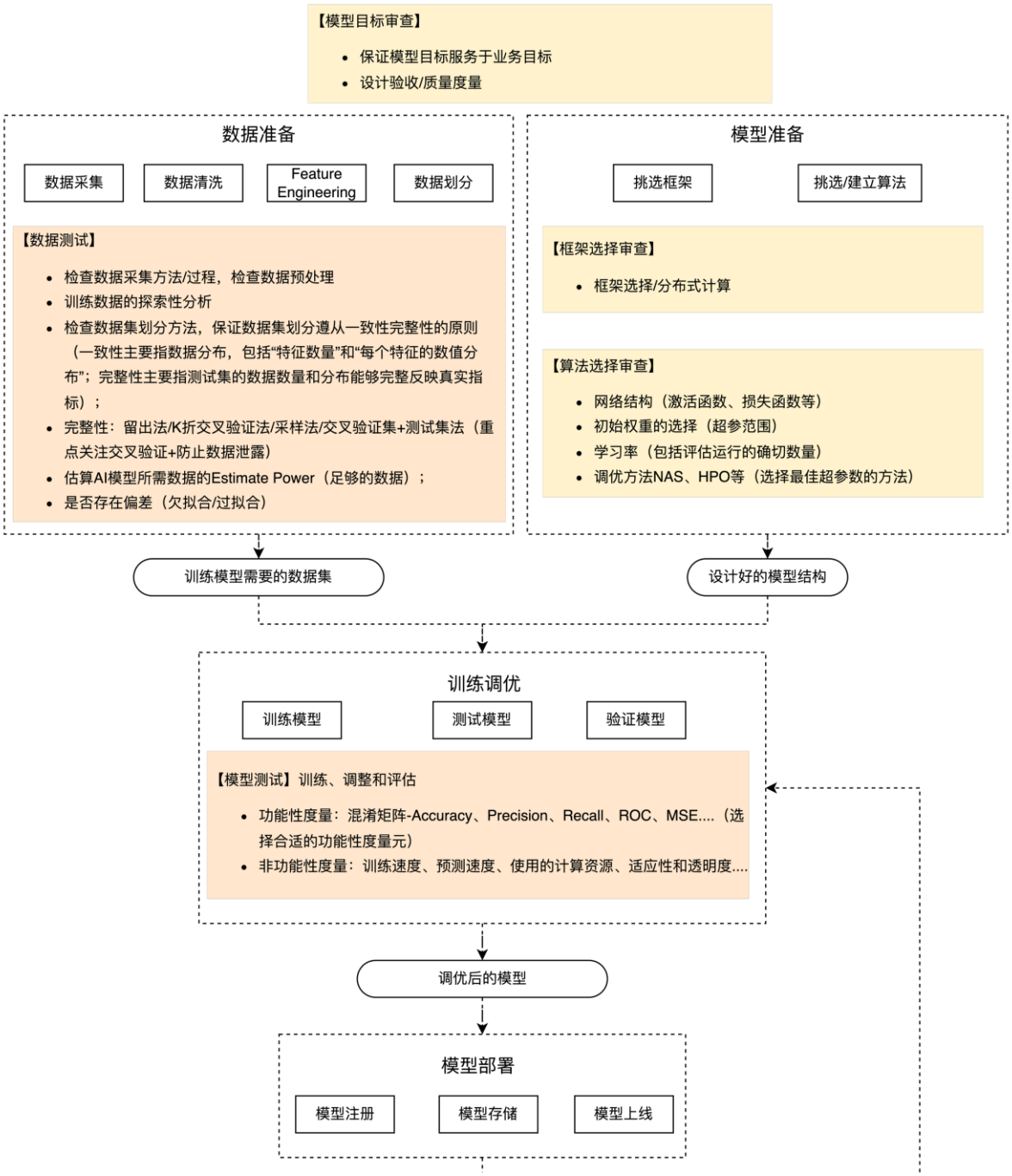
原始准确率即使用了真值图像和上色图像之间的差异来衡量上色网络的准确率。具体而言，使用图像 ab 通道的 L2 距离来衡量。

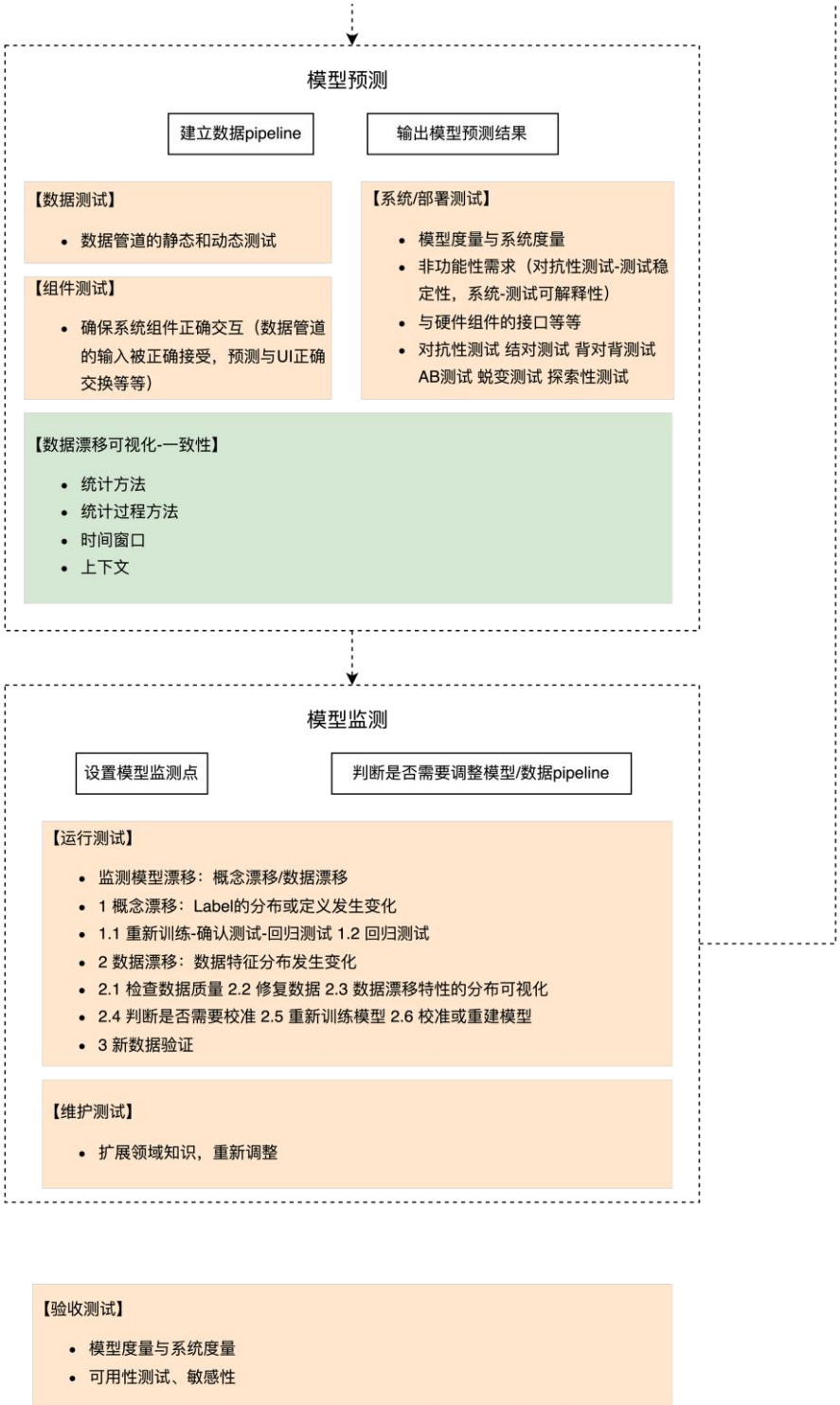
（3）语义解释性(VGG 分类)

用 VGG 网络来分类上色图片，根据分类效果和原始图片分类效果的好坏，从而来判断上色的逼真性。

3 系统的生命周期描述

3.1 人工智能系统的一般化生命周期




模型预测
建立数据pipeline
输出模型预测结果
【数据测试】
• 数据管道的静态和动态测试
【组件测试】
• 确保系统组件正确交互（数据管道的输入被正确接受，预测与UI正确交换等等）
【系统/部署测试】
• 模型度量与系统度量
• 非功能性需求（对抗性测试-测试稳定性，系统-测试可解释性）
• 与硬件组件的接口等等
• 对抗性测试 结对测试 背对背测试 AB测试 蜕变测试 探索性测试
【数据漂移可视化-一致性】
• 统计方法
• 统计过程方法
• 时间窗口
• 上下文
模型监测
设置模型监测点
判断是否需要调整模型/数据pipeline
【运行测试】
• 监测模型漂移：概念漂移/数据漂移
• 1 概念漂移：Label的分布或定义发生变化
• 1.1 重新训练-确认测试-回归测试 1.2 回归测试
• 2 数据漂移：数据特征分布发生变化
• 2.1 检查数据质量 2.2 修复数据 2.3 数据漂移特性的分布可视化
• 2.4 判断是否需要校准 2.5 重新训练模型 2.6 校准或重建模型
• 3 新数据验证
【维护测试】
• 扩展领域知识，重新调整
【验收测试】
• 模型度量与系统度量
• 可用性测试、敏感性

装 订 线

3.2 老照片修复系统的生命周期

老照片修复系统包含了上图所示的全部生命周期——数据准备、模型准备、模型调优、模型部署、模型预测和模型检测。

（1）老照片修复系统的数据准备

与上述的人工智能模型的一般生命周期不同之处在于，老照片修复系统（尤其是图像上色模块）使用了迁移学习的方法。具体而言，图像上色网络使用了多个不同的数据集，因此对每个被使用的数据集我们都需要进行合理的预处理与划分，需要考虑数据的一致性和数据的完整性问题，需要估算模型的 Estimate Power，需要考虑模型对于数据的过拟合问题和欠拟合问题。

（2）老照片修复系统的模型准备

由于老照片修复系统包含了两个人工智能模块，两个模块之间各自独立，对输入照片需要进行序列化的处理。在模型准备时，需要分别为每个人工智能组件准备合适的训练参数和损失函数。

（3）老作品修复系统的模型部署

与人工智能系统的一般生命周期相同，老照片修复系统也需要部署到服务器上为用户提供服务。除此之外，老照片修复系统也需要部署到个人计算机和智能手机等边缘设备中。

（4）老照片修复系统的模型预测

与人工智能系统的一般生命周期相同，老照片修复系统一经部署之后需要进行检验，进行数据测试、组件测试和部署测试，还要进行数据漂移测试。

（5）老照片修复系统的模型监控

与人工智能系统的一般生命周期相同，在老作品修复系统运行过程中，我们需要时刻检测系统的输出结果。保持对数据漂移的检测，一旦出现数据漂移现象和跨领域知识，我们需要重新训练我们老照片修复系统中的人工智能组件。

4 系统的准备阶段

4.1 数据的准备和获取

数据的来源包含两方面。一方面是开源数据集，包括用于图像上色模型预训练的 ImageNet 数据集和用于图像上色模型微调的 LFW（Labeled Faces in the Wild）数据集；另一方面是自行准备的数据集，即用于训练划痕识别网络的老照片划痕识别数据集。

4.1.1 ImageNet 数据集

ImageNet 数据集包含 14197122 个带标签的图像，是一个计算机视觉数据集，由斯坦福大学的李飞飞教授带领创建。数据集中的图片涵盖了大部分生活中会看到的图片类别，在促进计算机图像识别技术的发展方面该数据集作出很大的贡献。自 2010 年以来，该数据集被用于大规模视觉识别挑战赛（ILSVRC），成为了图像分类和目标检测的基准。数据集包含了训练集和测试集，并且训练集和测试集都带有标签注释。

数据集的一些信息如下：

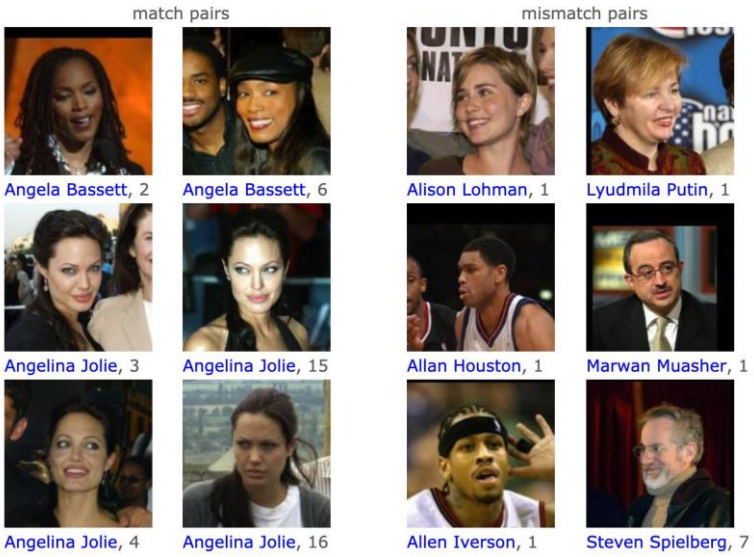
- 图像总数：14197122

- 图像的类别总数：21841
- 带有边界框注释的图像数量：1034908
- 具有 SIFT 特征的图像类别数：1000
- 具有 SIFT 特征的图像总数：1200000

4.1.2 LFW 数据集

LFW 是野外标记面孔数据集，该数据集主要用于无约束环境中的人脸识别问题。该数据集由 Gary 等人提出，包含从网络收集的 13233 张人脸图像。该数据集包含了 5749 个身份，是由 1680 个人组成的数据集，其中每个人都有有两张或更多图像。

数据集的示例如下：



4.1.3 老照片划痕识别数据集

老照片划痕识别数据集是我自己收集并标注的数据集。数据集包含 100 张带有划痕的老照片，并且均来源于网络。对于每张照片，数据集配备了一个将照片上的划痕标注出来的 mask 文件。该数据集主要用于训练老照片划痕识别网络，是一个用于二值语义分割任务的数据集。

数据集的示例如下：

4.2 数据分布的一致性评估（数据测试）

4.2.1 数据分布一致性的概念

数据分布的一致性的判断是旨在判断两个不同出处的数据（集）是否有着相同的来源。对于深度学习而言，数据分布一致性尤其重要，它直接关系到深度学习模型的性能好坏。

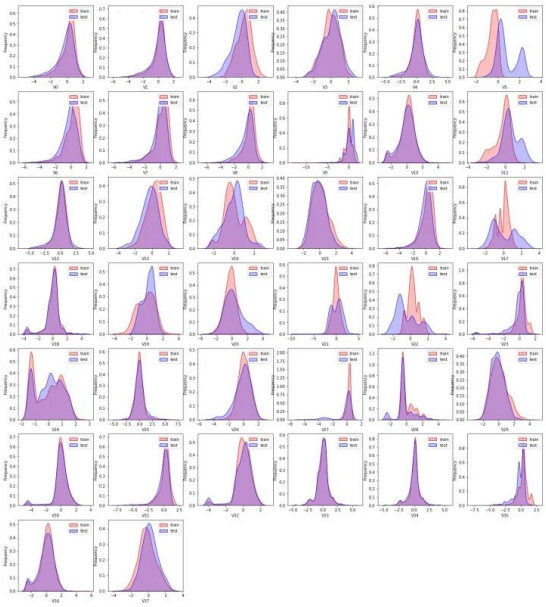
数据一致性的判断可以发生在多个阶段：在数据准备阶段，如果人工智能系统需要使用到多个不同的数据集，则对多个不同的数据集进行数据一致性判断是有必要的，因为它关系到模型具体的训练策略；在模型训练阶段，数据集往往需要划分为训练集、验证集和测试集来使用，对训练集、验证集和测试集之间数据一致性的判断也是有必要的，它直接关系到模型训练的结果；在部署后的模型监测阶段，训练时模型未见到的数据往往对模型会有较大影响，此时需要进行数据一致性的检验来验证模型是否发生了数据漂移/概念漂移现象。

4.2.2 数据分布一致性的评估方法

数据分布一致性的评估方法通常有三种，即基于核概率密度估计分布图的方法、基于 K-S 检验的方法和基于对抗验证的方法。

（1）核概率密度估计分布图

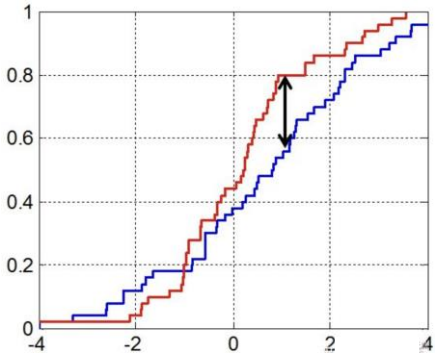
核概率密度估计（Kernel Density Estimation）分布图核概率密度估计可以理解为是每个数据特征的分布直方图进行加窗平滑后得到的结果。通过 KDE 分布图，可以查看并对训练数据集和测试数据集中特征变量的分布情况，如下图所示：



从 KDE 分布图可以直观得看出不同特征的分布一致性是不同的。KDE 分布图重合率较低的特征的分布不一致，重合度较高的特征的分布一致。

（2）K-S 检验

K-S 检验是一种非参数检验方法，用于检验两个数据分布是否一致。其利用两个数据分布的分布函数来进行计算。



其计算公式为：

$$D_{m,n} = sup_x \left| F_{1,n}(x), F_{2,m}(x) \right|$$

其中，F(x)是经验分布函数分布，n 和 m 分别是两个样本集合的数据量。H0:两个数据来自同一分布 H1:来自不同分布。若满足：

$$D_{m,n} \geq \sqrt{-\frac{1}{2}\ln\frac{\alpha}{2}} \times \sqrt{\frac{m+n}{mn}}$$

则接受 H0，否则拒绝 H0。

（3）对抗验证

对抗验证通过一下步骤完成：

- 构建一个样本的分类器，该二分类器的任务用于区分样本来源于训练集，还是测试集。
- 将新的训练数据集进行划分，保留部分样本作为该样本分类任务的测试集 T， 利用分类算法（XGBoost, LightGBM）等对数据集进行训练，AUC 作为模型指标；
- 在测试集 T 中进行验证，如果模型效果 AUC 在 0.5 左右，说明该样本分类模型无法区分样本来源训练集，还是测试集，说明原始数据中训练集，测试集分布是一致的；
- 如果 AUC 较大，如 0.9, 说明样本分类器很容易区分样本，间接说明训练集与测试集存在很大差异；根据第 3 步的结论，对于分布一致的，正常对目标任务训练即可。对于分布不一致的，可以继续进行样本挑选的尝试。

利用上述样本分类器模型，对原始的训练集进行打分预测，并将样本按照模型得分从大到小排序，模型得分越大，说明与测试集越接近，那么取训练集中的 TOP-N 的样本作为目标任务的验证集，这样即可将原始的样本进行拆分得到训练集，验证集，测试集。那么线上模型验证时，在这样的数据上调参等得到模型，如果在验证集上效果稳定，那么应用在测试集上，大概率结果是一致的。

4.2.3 图像数据集的分布一致性的评估细则

图像的每一个像素都可以视为数据的一个特征来使用 4.2.2 中提到的三个方法。但由于图像像素通道过多，直接在每个像素通道上进行一致性的判断所需计算量巨大，因此需要对图像进行关键特征提取和特征降维。此处拟采用三种特征提取和降维的方法，分别是 PCA 降维、LBPH 特征提取和 HOG 特征提取。

（1）直接使用像素特征

直接采用图片所有通道的像素构建 KDE 分布图，观察判断并统计 KDE 分布图中高重合度特征所占比例，若高重合度特征所占比例大于μ（其中μ是阈值，可以取 90%），则认为数据集分布是一致的，反之则认为不一致。

此外，同理可以使用图片所有通道的像素在显著性水平α下进行 K-S 检验，统计所有特征中接受一致性假设的特征所占比例，若接受一致性假设的特征所占比例大于μ（其中μ是阈值，可以

取 90%），则认为数据集分布是一致的，反之则认为不一致。

（2）在 PCA 特征降维的基础上进行一致性检验

主成分分析是一种特征降维的方法，该方法将特征进行线性组合，并选取方差最大的方向作为主成分方向。由于图片数据的像素较多，导致特征的维度过高，使用主成分分析的方法可以降低一致性评估所需要的计算量。我们对图像数据进行主成分分析，选取 95%-99%的主成分作为降维后的主成分。

采用图片的主成分构建 KDE 分布图，观察判断并统计 KDE 分布图中高重合度特征所占比例，若高重合度特征所占比例大于μ（其中μ是阈值，可以取 90%），则认为数据集分布是一致的，反之则认为不一致。

此外，同理可以使用图片的主成分在显著性水平α下进行 K-S 检验，统计所有特征中接受一致性假设的特征所占比例，若接受一致性假设的特征所占比例大于μ（其中μ是阈值，可以取 90%），则认为数据集分布是一致的，反之则认为不一致。

（3）在 LBPH 特征提取的基础上进行一致性检验

LBPH 是局部二值模式直方图，该方法通过计算每一个像素值处的局部二值模式，并且将区域的内的二值模式进行统计得到直方图。局部二值模式具有旋转不变性和等价模式性，使得 LBPH 是比原始像素维度更低的特征。

在此基础上，采用图片的局部二值模式直方图构建 KDE 分布图，观察判断并统计 KDE 分布图中高重合度特征所占比例，若高重合度特征所占比例大于μ（其中μ是阈值，可以取 90%），则认为数据集分布是一致的，反之则认为不一致。

此外，同理可以使用图片的局部二值模式直方图在显著性水平α下进行 K-S 检验，统计所有特征中接受一致性假设的特征所占比例，若接受一致性假设的特征所占比例大于μ（其中μ是阈值，可以取 90%），则认为数据集分布是一致的，反之则认为不一致。

（4）在 HOG 特征提取的基础上进行一致性检验

HOG 是方向梯度直方图，该方法通过计算每一个点处的方向梯度，并且将区域内的方向进行统计得到直方图。由于方向梯度总共划分九个方向，这使得 HOG 是比原始像素维度更低的特征。

在此基础上，采用图片的方向梯度直方图构建 KDE 分布图，观察判断并统计 KDE 分布图中高重合度特征所占比例，若高重合度特征所占比例大于μ（其中μ是阈值，可以取 90%），则认为数据集分布是一致的，反之则认为不一致。

此外，同理可以使用图片的方向梯度直方图在显著性水平α下进行 K-S 检验，统计所有特征中接受一致性假设的特征所占比例，若接受一致性假设的特征所占比例大于μ（其中μ是阈值，可以取 90%），则认为数据集分布是一致的，反之则认为不一致。

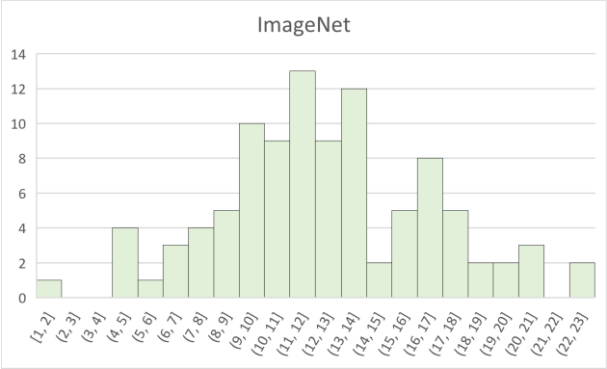
4.2.4 数据一致性的检验结果

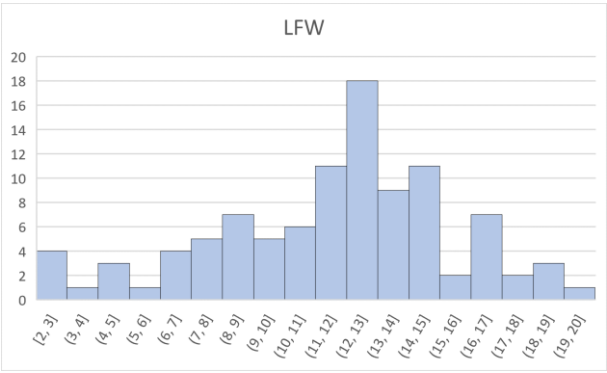
（1）实验内容

由于时间有限，仅进行部分实验，具体实验过程为：随机选取 LFW 数据集中的 100 张图像，提取 LBPH 特征，统计 LBPH 特征中每一个特征的数值分布；选取 ImageNet 数据集中的 100 张图像，提取 LBPH 特征，统计 LBPH 特征中每一个特征的数值分布；绘制特征数值分布统计图，进行一致性判别。其中，LBPH 的取样半径为 2，取样点个数为 16，采用旋转不变统一模式 LBP。

（2）实验结果

以第一个特征绘制直方图，结果如下：





从图中可以看到，数据分布存在一定的差异，因此 ImageNet 和 LFW 数据集之间的数据没有一致性。

装 订 线

4.3 数据在训练时的划分

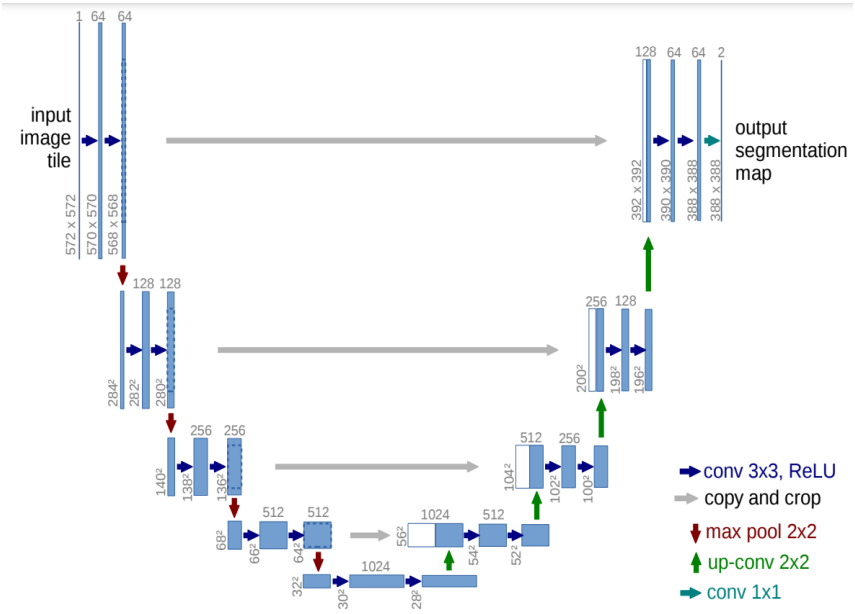
数据在模型的训练阶段需要划分为训练集和验证集作为训练的依据，同时需要测试集作为测试的依据。

三个数据集中的 LFW 和老照片划痕检测数据集在使用的过程中均按照 K 折交叉验证的思想来进行随机划分，K 取 10。模型在验证集上的准确度以 10 折交叉验证的平均值为标准。而 ImageNet 由于进行了预训练，故不需要再由我们来进行训练前的划分。

4.4 训练参数的确定与调优

4.4.1 划痕识别网络

在划痕识别上，我们使用了医学领域常用的二值语义分割网络 Unet。Unet 的结构如下图所示：

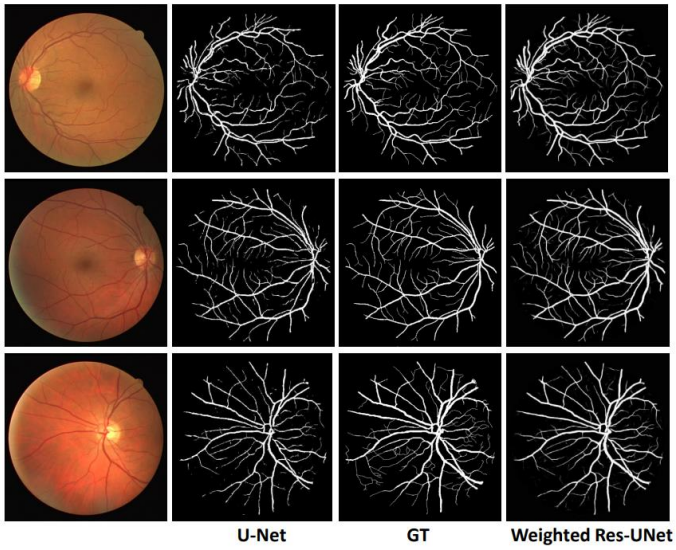


该网络结构由于类似于字母 U 而得名。Unet 包含了编码器-解码器结构，网络的左侧是特征提取结构，网络的的右侧是特征表示结构。特征提取结构通过四次卷积和最大池化的操作得到输入图像的特征图；特征表示结构通过四次上采样和卷积操作得到输出图像；此外，在编码器和解码器之间，Unet 通过裁剪和跳越拼接直接传递不同尺度的特征图，从而做到了特征的保持。

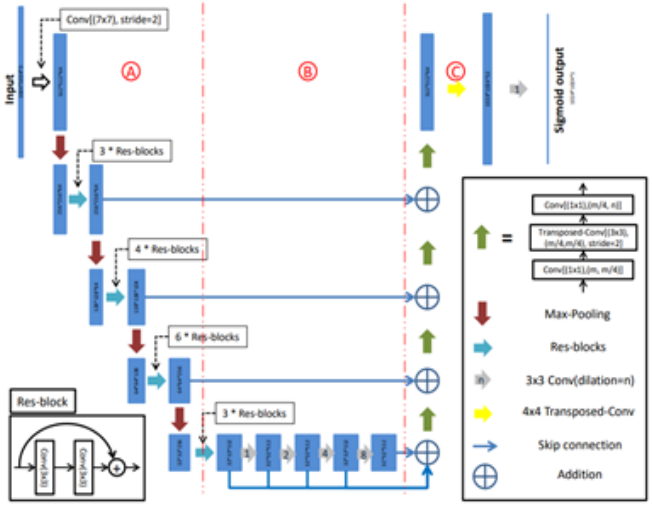
为了将 Unet 用于划痕识别网络，我们将 Unet 做三处修改：

- 将网络的输入修改为 224*224*3，以适应输入图像；
- 在网络中使用 padding，这样跨越连接时的特征图尺寸就是相同的，不再需要进行裁剪就可以直接拼接；
- 在模型的最后加上只有 1 个 1*1 卷积核的卷积层，并且使用 sigmoid 激活函数，从而网络可以输出得到 224*224*1 的二值分割图。

考虑到传统 Unet 主要用于区域的分割，与老照片上的划痕存在一定区别。相关研究表明，在 Unet 上引入残差结构有利于网络对眼部毛细血管的分割效果更加，会使得分割的边缘更加“圆滑”，如下图所示：



因此，我们同样在 Unet 中加入残差结构：



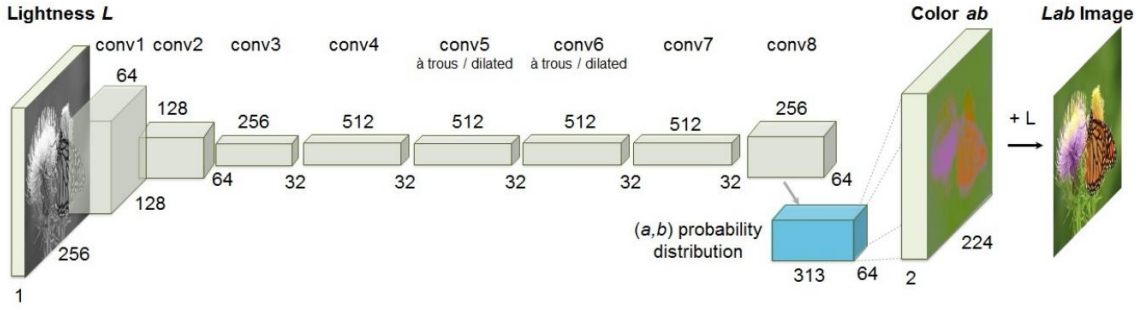
至此，我们用于划痕识别的网络模型搭建完毕。

装 订 线

装 订 线

4.4.2 图像上色网络

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*256*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。

5 系统的训练阶段

5.1 超参数搜索

使用网格搜索的方法进行超参数搜索。

（1）划痕识别网络

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-5, 2e-5, 5e-5, 8e-5,1e-4, 2e-4, 5e-4] 训练 [20, 25, 30, 35, 40] 个 epoch。

（2）图像上色网络

在 ImageNet 数据集上的预训练采用现有的模型参数。

在 LFW 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-7, 2e-7, 5e-7, 8e-7,1e-6, 2e-6, 5e-6]，训练[12, 15, 18, 21, 24, 27, 30]个 epoch。

5.2 模型评估

模型评估建立在交叉验证的基础上。数据集被划分为 10 折，每次取其中 9 折作为训练集，然后 1 折作为验证集进行模型的开发评估。因此每次实验将会得到 10 个开发集准确率，我们使用其平均值作为此次实验的准确率。

在评价标准方面，对于划痕识别的任务，采用二值图像素分类准确率作为评估标准；对于图像上色任务，由于上色具有多种合理性，因此使用视觉真实性这一指标进行测试。

5.3 划痕修复网络的模型评估

（1）普通 Unet

设定随机数种子为 42，得到实验结果如下：

学习率	epoch	准确率 PA
1e-5	20	88.4%
2e-5	20	92.5%
5e-5	20	93.7%

8e-5	20	90.8%
1e-4	20	93.3%
2e-4	20	94.3%
5e-4	20	93.9%
1e-5	25	94.6%
2e-5	25	88.3%
5e-5	25	94.3%
8e-5	25	92.4%
1e-4	25	89.2%
2e-4	25	91.5%
5e-4	25	91.6%
1e-5	30	90.2%
2e-5	30	90.9%
5e-5	**30**	**94.8%**
8e-5	30	88.3%
1e-4	30	91.7%
2e-4	30	91.6%
5e-4	30	94.1%
1e-5	35	89.7%
2e-5	35	89.4%
5e-5	35	90.1%
8e-5	35	94.4%
1e-4	35	92.0%
2e-4	35	93.3%
5e-4	35	88.1%
1e-5	40	94.5%
2e-5	40	89.9%
5e-5	40	92.5%
8e-5	40	92.6%
1e-4	40	90.0%
2e-4	40	88.4%
5e-4	40	91.9%
平均值		**91.6%**

从表格中可以知道：普通 Unet 在学习率为 5e-5 时训练 30 个 epoch 得到最高的准确率 94.8%；而该模型的平均准确率为 91.6%。

（2）残差改进 Unet

设定随机数种子为 42，得到实验结果如下：

学习率	epoch	准确率 PA
1e-5	20	95.6%
2e-5	20	87.2%
5e-5	20	87.5%
8e-5	20	88.9%
1e-4	20	92.1%
2e-4	20	88.1%
5e-4	20	88.8%
1e-5	25	90.9%
2e-5	25	89.1%
5e-5	25	92.1%
8e-5	25	93.2%
1e-4	25	88.8%
2e-4	25	90.4%
5e-4	25	93.8%
1e-5	30	89.9%
2e-5	30	92.0%
5e-5	30	87.2%
8e-5	**30**	**96.2%**
1e-4	30	88.6%
2e-4	30	89.6%
5e-4	30	87.7%
1e-5	35	87.4%
2e-5	35	90.5%
5e-5	35	94.8%
8e-5	35	87.2%
1e-4	35	94.2%
2e-4	35	88.0%

5e-4	35	91.1%
1e-5	40	94.9%
2e-5	40	93.0%
5e-5	40	93.3%
8e-5	40	94.7%
1e-4	40	93.7%
2e-4	40	92.0%
5e-4	40	90.7%
平均值		**91.0%**

从表格中可以知道：残差改进 Unet 在学习率为 8e-5 时训练 30 个 epoch 得到最高的准确率 96.2%；而该模型的平均准确率为 91.0%。因此，在最高准确率上，残差改进 Unet 有更好的效果，但也面临性能稳定性下降的问题。

5.4 图像上色网络的模型评估

依照视觉真实性进行人工评估，具体方法为：将由上色网络上色的图像与 GT 图像放在一起，人工判断“哪一张是真实色彩、哪一张是 AI 上色的结果”。AI 上色的识别率越高则上色效果越差，而上色识别率接近 50%表明 AI 上色可以以假乱真。

下图是上色的一个示例，第一列是模型输入的灰度照片，第二列到第四列是模型上色结果（在不同的网络参数下），第五列是 GT 图像。

可以从结果中观察出模型上色的结果与 GT 在一定程度上是可以区分的，因此模型性能还有待进一步提高。

由于基于视觉真实性的评估需要消耗大量人力，本次课程项目并未完成图像上色任务的评估。

6 系统的部署阶段

在实现系统的人工智能组件的基础上需要构建整个系统，这其中包含了系统的文件子系统、系统的 UI 界面和系统各部分之间的连接。

6.1 项目风险分析

首先需要指出项目/系统存在的潜在风险，并且给各个风险以相应的风险等级。

风险分类	具体风险	风险等级
技术风险	人工智能组件涉及的技术过于新颖	2
	人工智能组件涉及的技术复杂程度高	2
	技术版本过多，没有统一标准	3
	团队技术薄弱，技术储备不足	3
	代码冗余、混乱	2
	缺乏硬件 GPU 和深度学习平台	3
	技术文档编写不全	2
需求风险	产品需求多变或不明确	2
	产品需求不规范	2
	缺乏备用的需求解决方案	4
人员风险	技术人员经验缺乏	3
	人员分工配置不合理	2
	核心技术人员离职/缺失	4

市场风险	市场需求不断变动	2
	AI 商业化落地困难	1
	竞品的出现	4
政策风险	违背政府的相关政策	4
	违背开源协议	5

对项目风险的风险分析如下：

（1）人工智能组件涉及的技术过于新颖

本项目涉及计算机视觉的语义分割和图像上色任务，还涉及图像修复的数值方法，最终的产品形式是老照片修复的人工智能平台，它可以代替常规的人工进行老照片的自动化修复，实现照片划痕修复、照片上色和综合修复的功能。项目依靠较为复杂的算法做支撑，且都是目前比较火热的计算机视觉领域的深度学习技术。深度学习的框架技术更新迭代较快，但研究新模型需要经过采集数据、做实验、适配到业务这三个流程，三个环节会花费一定的时间和精力。与此同时项目进展的节奏比较快，如果一味花大量的时间研究更新、更复杂的算法模型而忽略项目功能的实现会对项目的进度会造成一定的影响。除此之外因为 Pytorch、Tensorflow 等深度学习框架更新比较快，每套深度学习框架会有很多不同的版本，更新的框架版本可能会不兼容老版本的代码，或者因为当前 GPU 的 cuda 版本较老而导致新的版本的代码不能够使用，需要算法人员反复地进行代码升级和版本改动，而改动版本模型会花费一定的时间，而项目的开发时间是已经有确定排期的，因此如果研究时间过长会导致项目的延期，并导致此项目核心功能模块不能按时完成。

（2）人工智能组件涉及的技术复杂程度高

系统的技术体系较为复杂，涉及计算机视觉领域的语义分割、图像修复和图像上色三个不同的技术领域。如果仅仅采用各自领域最传统的算法技术，，那么项目的整体效果就很难达到较佳的效果。现阶段随着数据爆炸式增长以及计算能力的显著发展，以深度学习为代表的算法开始在工业界大规模落地并在很多场景下发挥了作用。在语义分割领域，Unet 及其变形的相关网络应用广泛；在图像修复邻域，基于深度学习的 GAN、Diffusion 模型则具有较好的修复效果，而数值方法则具有较高效率；在图像上色领域，CNN 和 Transformer 模型都有较好的上色表现。复杂的算法技术问题会会给项目带来延期的风险。

（3）技术版本过多，没有统一标准

传统的 IT 项目从技术架构上来说后端只会有一套编程语言，比如前端会以 Javasript 为主，后端只会以 Java 或者 php 为主，从技术的选择上来说极少会出现因编程语言版本过多而导致项目出现相关的风险问题。对于老照片修复系统而言却存在着多种编程语言的问题。从流程上来说会有算法模型作为最底层的支撑，而算法模型本身可以由当下最火热的编程语言 python 训练而成，算法工程师因技术背景不同，在选择编程语言版本时有很大差异。年轻的算法工程师喜欢使用 Pytorch、Tensorflow、 Keras 这种基于 Python 开发的深度学习框架或者使用基于 Scala 开发的 Sparkmlib、Flink 这种新兴的分布式机器学习框架进行模型训练，而资历比较老的算法工程师则比较喜欢使 Weka、JavaMlib 这种以 Java 为主的机器学习库或者自己手动使用 C++写算法模型;擅长数据分析的算法工程师更喜欢使用 Hive、Spark 进行算法分析和模型训练。技术选择过于多样性导致了算法模型虽然可以各自在本地进行算法模型开发，但不能够统一在线上进行使用并做工程化服务，技术版本过于分散导致会导致项目在开发的过程中受到核心的影响。

（4）团队技术薄弱，技术储备不足

老照片修复系统项目主要由三位本科生完成项目的全部流程。项目团队缺乏实际的算法工程开发经验，只能做一些比较简单的功能开发；缺乏一定的技术储备和一定的工程能力，尤其在目前人工智能需要大面积的落地的情况下，没有一定的工程能力，算法就不能够发挥其重要的作用。基于以上的风险，可能会导致项目最终延期甚至失败。

（5）代码冗余、混乱

代码冗余问题在本项目的开发过程中代码冗余问题在各个团队都有出现，因工程师经验不足的问题导致了提交的代码不规范，并且存在着代码严重冗余的问题，同一个功能反复出现相同的代码，并且缺乏相关的注释。其次代码编写也存在着不规范的问题，包括类命名不规范、方法名命名不规范、并且提交格式也存在不规范等种种问题，代码可读性差导致接手的开发人员花费大量的时间在老代码的阅读上，这样也导致了新功能开发受到影响。

（6）缺乏硬件 GPU 和深度学习平台

老照片修复系统集成了计算机视觉领域的众多任务，因此整体项目技术架构比较复杂。而相应的技术路线中，主流技术都是使用深度学习技术，然而深度学习技术需要大规模的计算力作为基础支撑，也就是需要运算力比较强的分布式机器学习平台和以 GPU 集群搭建的计算平台来支撑相关项目的计算和使用。然而，由于是学生完成的项目，算力资源自然匮乏。

（7）技术文档编写不全

技术文档的作用体现在在从技术、管理和开发环境等几个方面，确定了一个软件可以完成。技术文档可以加强团队之间的配合和沟通，譬如开发人员可以根据技术文档非常清楚地理解相关

产品模块的实现要求，同时也可以阅读自己下游调用团队的接口定义，相关开发工具的配置以及公司内部各种配置平台和数据库申请平台的使用。老照片修复系统在开发的过程中面临着技术文档不全的问题，一方面产品文档需求不明确，有的需求甚至只是简单地描述，并且没有解释此功能的具体实现要求，同时开发人员也没有把自己工作范围内的文档写的很清楚清晰，包括接口定义不清晰、不明确、输入参数不明确、返回结果不明确等等技术的细节问题。上述的这种因为相关技术文档不全面的问题会导致项目因文档不全而拖延的风险，尤其是在相关的开发阶段没有技术文档的指引，技术人员很可能会因为做很多无用功，从而拖延项目的进展。

（8）产品需求多变、不明确或不规范

人工智能蓬勃发展也使具有人工智能特点产品出现在市场中，技术快速迭代带来的变革也让市场环境瞬息万变。在互联网高速发展的背景下，市场的环境竞争激烈。市场上的同类产品功能不断地推陈出新，因此对应的产品需求不断地进行变更，然而这些产品需求中有些是合理的，有些则是非常不合理的，这种无限度的需求变更给项目带来了很大影响,轻则导致研发人员经常加班进行功能开发，重则导致了项目成本增加，甚至严重影响了项目核心模块的开发进度。老照片修复系统的产品形态不同于普通 IT 项目的产品形态，过去产品经理根据用户明确的需求设计产品，产品研发出来的结果会和原型设计保持一致。但是人工智能产品经理则需要不同的思维模式，不应该再花大量的时间和资源来寻找确定的因果关系，而是应该通过大量的数据挖掘手段探索出相关性，并用数据指导产品设计。此外，一些功能需求的描述也比较摸棱两可，导致技术开发人员对需求的理解有很大的歧意，这种因为需求不明确的情况导致了项目在开发过程中进度受阻。

（9）缺乏备用的需求解决方案

在项目的开发的过程中，需求变动是难以避免的。需求的变化需要一些及时的备用解决方案来替代，从而保证项目可以正常地运转，而在老照片修复系统的开发过程中因开发模式的不当，导致需求如果发生了变化，我们则不能够及时提供相应的备用解决方案，与此需求相关的功能则受到阻滞，进而使项目的整体进度面临着延期的风险。

（10）技术人员经验缺乏

本项目的技术人员缺乏相关的项目经验，对此项目需要的技术不熟悉，这使项目本身的稳定性产生一定的影响，除此之外还会增加项目的成本，进而影响项目正常进度。

（11）人员分工配置不合理

本项目的人员分工是临时决定的，在分工之前仅按照任务量来进行分工，并没有按合理的方式进行分工，这将影响到任务的正常进行。

（12）核心技术人员离职/缺失

本项目由三位负责人全权负责并完成，每个人负责的部分相互独立。在这样的前提下，项目

核心人员的离职将对项目产生极大的影响，需要很长时间才能完成任务的交接工作，这必将严重影响到任务进度。

（13）市场需求不断变动

随着技术不断的发展，尤其是在人工智能领域，新的技术发展极快。在系统开发的过程中，在初步拟定技术路线后，很有可能新的技术也会出现，随之而来的是用户的新的需求。人工智能领域的高速发展现状决定了市场的需求会不断改变。

（14）AI 商业化落地困难

AI 商业化落地一直是一个困难的问题，人工智能行业目前缺乏统一标准，各个公司都是各自为主的发展，其表现之一就是没有统一的 AI 标准，也没有统一的项目系统，无法模仿现有方法进行开发。老照片修复系统也不例外，目前照片修复做得比较成熟的公司集中在各大互联网巨头中，对于我们这样的较小团队在真正地做完此项目后，对系统的运营维护必然有很大困难。

（15）违背政府的相关政策

人工智能相关的产品目前正处落地过程中，其本身受国家或者政府层面政策的监管，尤其近几年多个国家出台了多项关于人工智能应用监管的政策，政策提出人工智能产品在市场上的应用场景，以及人工智能产品本身的智能行为可能会与当地的政府政策等产生冲突。老照片修复系统同样面临着类似的风险。如果没有及时进行话术过滤和话术标准化，那么可能会导致违反相关政策。

（16）违背开源协议

老照片修复系统的整体技术栈是基于深度学习和机器学习作为算法基底，用于产品开发的技术类似于 Tensorflow 或者 Pytorch 属于开源的深度学习框架，系统内部集成的大部分计算机视觉技术都是开源的，然而仅用于学术交流的技术资源一般在商用方面仍需要仔细研究相关的开源协议，以免因违背开源协议造成不必要的损失。

6.2 系统测试

系统测试是人工智能系统生命周期里十分重要的一个环节。根据测试层次的不同，测试被分为多个不同的测试等级——单元测试、数据测试、模型测试、集成测试、确认测试、用户测试。对于人工智能组件的测试同样有多种方法，例如对抗测试、背对背测试、A/B 测试、蜕变测试、模型漂移测试等。

6.2.1 单元测试

单元测试集中对用源代码实现的每一个程序单元进行测试，检查各个程序模块是否正确地实现了规定的功能，这里主要指的是系统的非人工智能单元，人工智能单元具有不同的测试级别。

对于老照片修复系统，单元测试中需要测试的单元包括文件子系统单元和人机交互界面单元。

（1）文件子系统测试

文件子系统主要负责读取计算机本地的文件和目录并且将其显示，打开输入图片，保存输出结果。具体测试内容如下：

测试编号	测试内容	期望结果
T1.1	执行“浏览（打开）”功能	正确显示计算机本地文件和目录
T1.2	执行“打开图片”功能	图片被系统打开并显示
T1.3	重复打开相同图片	图片被系统打开并正确显示
T1.4	交替打开不同图片	图片被系统打开并正确显示
T1.5	执行“浏览”但不打开图片	系统正确执行
T1.6	打开非图片格式的文件	系统不打开并给予提示
T1.7	执行“浏览（保存）”功能	正确显示计算机本地文件和目录
T1.8	执行“保存图片”功能	图片被系统正确保存
T1.9	重复保存图片在同一目录	图片被系统正确保存
T1.10	重复保存图片在不同目录	图片被系统正确保存
T1.11	以非法文件名保存图片	系统给出提示且不保存

（2）人机交互界面测试

人机交互界面主要负责展示系统的功能，并且给予用户操作系统的空间。对人机交互界面的

具体测试内容如下：

测试编号	测试内容	期望结果
T2.1	功能界面间的切换	正确切换
T2.2	执行“打开图片”功能	系统正确给予提示并打开图片
T2.3	执行“修复”功能	系统正确完成相应的图片修复任务
T2.4	执行“保存图片”功能	系统正确给予提示并保存图片
T2.5	未打开图片时执行“修复”	系统给予提示并且忽略操作
T2.6	未打开图片时执行“保存图片”	系统给予提示并且忽略操作
T2.7	未修复时执行“保存图片”	系统给予提示并且忽略操作
T2.8	打开图片后切换界面再切换回来	系统正常，图片仍正确显示，且可以执行当前页面的全部操作
T2.9	修复后切换界面再切换回来	系统正常，图片仍正确显示，且可以执行当前页面的全部操作
T2.10	保存图片后切换界面再切换回来	系统正常，图片仍正确显示，且可以执行当前页面的全部操作
T2.11	多次点击同一按钮	系统正常执行相应操作
T2.12	点击“退出”按钮	系统退出
T2.13	点击“最小化”按钮	系统最小化

（3）接口测试

接口测试主要测试的是各个单元模块间的接口是否正确。主要测试内容如下：

测试编号	测试内容	期望结果

T3.1	调用“修复”函数	单独完成依次人工智能单元的推理过程，或者序列化完成划痕修复与图像上色的推理过程
T3.2	执行“打开图片”函数	调用输入图片函数，完成图片的打开
T3.3	执行“保存图片”函数	调用保存图片函数，完成图片的保存

装 订 线

6.2.2 数据测试

数据测试是针对人工智能模块的测试，此处测试的目的是确保系统用于预测数据具有相应的质量，并且测试数据漂移现象的发生。

（1）数据质量监测

在系统部署后，人工抽查数据流处的输入数据，人工核验输入图片在图片内容、图片清晰度等方面的质量因素。

（2）数据漂移检测

每当从数据流处收到一批数据（例如 20 张）后，使用 4.2.3 的方法检测这批数据与训练数据的分布一致性。若具有一致性，则模型正常运行；若数据的一致性无法满足，则表明发生了数据漂移现象，此时需要利用这批数据进行模型训练。

6.2.3 模型测试

针对人工智能模型的性能测试已经在模型的训练阶段完成。在此基础上，可以使用蜕变测试的思想对模型进行进一步测试，测试系统的鲁棒性。

蜕变测试中测试用例的构建可以参考图像的数据增强方法，即通过图像的缩放、旋转、增强/减弱光照、增强/减弱对比度等操作生成新的测试用例，以完成蜕变测试的测试用例构建。

6.2.4 集成测试

集成测试是把已测试过的模块单元组装起来，主要对与设计相关的软件体系结构的构造进行测试。老照片修复系统将各个功能单元自底向上组装，集成测试主要通过定义系统的操作路径来

完成测试，要求操作路径覆盖系统所有可能出现的情况。

测试编号	测试内容	期望结果
T4.1	打开系统，切换到图像修复模块，打开照片，执行修复，将修复结果保存，关闭系统。	系统正常完成全部操作
T4.2	打开系统，切换到图像上色模块，打开照片，执行修复，将修复结果保存，关闭系统。	系统正常完成全部操作
T4.3	打开系统，切换到综合模块，打开照片，执行修复，将修复结果保存，关闭系统。	系统正常完成全部操作
T4.4	打开系统，切换到图像修复模块，打开照片，执行修复，将修复结果保存，切换到图像上色模块，打开照片，执行修复，将修复结果保存，切换到综合模块，打开照片，执行修复，将修复结果保存，关闭系统。	系统正常完成全部操作
T4.5	打开系统，依次切换所有的功能界面，最后会到首页，关闭系统。	系统正常完成全部操作
T4.6	打开系统，切换到图像修复模块，打开照片，执行修复，切换到图像上色模块，打开照片，执行修复，切换到综合模块，打开照片，执行修复，关闭系统。	系统正常完成全部操作
T4.7	打开系统，切换到图像修复模块，打开照片，执行修复，切换到图像上色模块，打开照片，执	系统正常完成全部操作

	行修复，切换到综合修复模块，打开照片，执行修复，保存结果，切换到图像上色模块，保存结果，切换到划痕修复模块，保存结果，切换到首页模块，关闭系统。	
T4.8	打开系统，切换到图像修复模块，打开照片，执行修复，将修复结果保存，打开图片，执行修复，将修复结果保存，打开图片，执行修复，将修复结果保存，关闭系统。	系统正常完成全部操作
T4.9	打开系统，切换到图像上色模块，打开照片，执行上色，将修复结果保存，打开图片，执行上色，将修复结果保存，打开图片，执行上色，将修复结果保存，关闭系统。	系统正常完成全部操作
T4.10	打开系统，切换到综合修复模块，打开照片，执行修复，将修复结果保存，打开图片，执行修复，将修复结果保存，打开图片，执行修复，将修复结果保存，关闭系统。	系统正常完成全部操作

以上测试用例基本可以涵盖用户对系统的所有可能操作。

6.2.5 确认测试

确认测试又称有效性测试，该测试的任务是验证软件的功能和性能及其它特性是否与用户的要求一致。老照片修复系统的功能需求已经在 6.2.4 的集成测试阶段完成测试，根据需求分析，确认测试需要检测系统的性能需求、环境需求和相关文档需求。经过确认测试的系统就可以交付

到用户手中使用。

（1）功能测试

即 6.2.4 的集成测试

（2）性能测试

性能测试即测试系统是否满足需求分析中的性能需求。

测试编号	测试内容	期望结果
T5.1	检查系统所占的储存空间的大小	小于 3GB
T5.2	在打开图片后，完成一次划痕修复，记录所需时间	用时小于 0.5 秒
T5.3	在打开图片后，完成一次图像上色，记录所需时间	用时小于 0.5 秒
T5.4	在打开图片后，完成一次综合修复，记录所需时间	用时小于 1.0 秒
T5.5	执行多次系统功能后，再完成一次划痕修复，记录所需时间	用时小于 0.5 秒
T5.6	执行多次系统功能后，再完成一次图像上色，记录所需时间	用时小于 0.5 秒
T5.7	执行多次系统功能后，再完成一次综合修复，记录所需时间	用时小于 1.0 秒

（3）环境需求测试

环境需求测试需要我们测试系统的可移植性，即在不同的个人计算机上安装老照片修复系统。测试内容为：在两台 windows10 的笔记本上安装本系统，在两台 windows11 的笔记本上安装本系统，在一台 MacOS 的笔记本上安装本系统。

（4）文档需求测试

检查老照片修复系统的所有文档资料是否准备完毕，是否存在遗漏。包括开发说明书、系统使用说明书，以及相关开源证件。

6.2.6 用户测试

用户测试即模拟用户使用该系统的真实环境进行测试，从中发现之前尚未发现的问题。

（1）α测试

将系统安装包和相关文档一并打包，交予 1 名系统开发人员，安装系统并操作系统，完成系统的使用过程。

（2）β测试

将系统安装包和相关文档一并打包，交予 3 名测试志愿者，安装系统并操作系统，完成系统的使用过程。

6.3 测试清单

将上述测试与需求相对应，测试清单和通过情况如下：

需求	测试编号	测试通过率
划痕修复功能模块	T4.1、T4.3、T4.4、T4.6、T4.7、T4.8、T4.9、T4.10	100%
图像上色功能模块	T4.2、T4.3、T4.4、T4.6、T4.7、T4.8、T4.9、T4.10	100%
文件子系统模块	T1.1、T1.2、T1.3、T1.4、T1.5、T1.6、T1.7、T1.8、T1.9、T1.10、T1.11、T3.1、T3.2、T3.3、T4.1、T4.2、T4.3、T4.4、T4.6、T4.7、T4.8、T4.9、T4.10	100%
人机交互界面模块	T2.1、T2.2、T2.3、T2.4、T2.5、T2.6、T2.7、T2.8、T2.9、T2.10 T2.11、T2.12 、T2.13、T3.1、T3.2、T3.3、T4.1、T4.2、T4.3、T4.4、T4.5、T4.6、T4.7、T4.8、T4.9、T4.10	100%

存储需求	T5.1	100%
响应需求	T5.2、T5.3、T5.4、T5.5、T5.6、T5.7	76.6%

其中，关于响应需求的每个测试使用不同大小的 10 张照片进行输入，得到结果。

6.4 系统监测

系统监测是在完成系统的部署之后，在线对系统的行为和表现进行监测。

（1）系统运行的监测

在系统部署后，时刻监测所有系统的运行状态，及时收集系统错误信息，并对系统进行改进。

（2）数据漂移的监测

在线监测系统所处理的数据，具体做法为：

每当从数据流处收到一批数据（例如 20 张）后，使用 4.2.3 的方法检测这批数据与训练数据的分布一致性。若具有一致性，则模型正常运行；若数据的一致性无法满足，则表明发生了数据漂移现象，此时需要利用这批数据进行模型训练，并且同步更新网络模型的参数。

装 订 线

7 总结

在本次《人工智能系统模型评估》课程的期末作业中，我以我曾经做过的一个人工智能项目老照片修复系统为例，详细阐述了人工智能系统的整个生命周期。与当时所撰写的“技术报告”不同的是，本次的课程报告是以测试为核心，详细阐述了人工智能系统的测试方法，包括值得关注的数据一致性问题、性能度量元和数据漂移问题等。

本次报告的核心在于人工智能系统生命周期的描述，报告从需求分析开始，介绍了系统的人工智能组件，介绍了系统的生命周期，介绍了数据准备阶段的数据收集、数据一致性判别、数据集的使用以及网络架构的搜索和调优，介绍了模型训练阶段超参数搜索和模型评估，并且介绍了模型在部署阶段的风险分析、测试方法以及系统监测方法。总的来说是对人工智能系统生命周期一个较好的描述。

当然，报告也存在一些不足之处：

- 叙述逻辑有待改进。报告在系统部署部分的叙述逻辑不够清晰，一方面针对项目的风险分析不应该仅在部署阶段考虑，另一方面对数据漂移的测试与监测也阐述不够清晰。
- 实验有待完善。本次报告仅完成了修复网络的性能评估，对上色网络的性能评估尚未完成，此外，部署阶段的测试和监测也仅仅给出了方法，并未实施。
- 示例存在局限性。本次报告所展示的老照片修复系统仅仅是人工智能系统的一个简单例子，和正式的、商用的人工智能系统之间还存在较大的差距。