**人工智能训练师（三级）操作技能考核**

**试题单**

准考证号：

试题代码：

试题名称：手写数字识别系统交互流程设计

考核时间：20min

**1.**场地设备要求

（1）人工智能训练师主机 1 台；   
（2）Python 编译环境；   
（3）Pytorch框架。

**2.**工作任务

手写数字识别系统是在数字化转型和自动化需求日益增长的社会背景下应运而生的。随着信息时代的到来，大量手写文档需要进行电子化处理，以提高数据存储、检索和分析的效率。传统的光学字符识别(OCR)技术在处理手写体时面临诸多挑战，如书写风格的多样性、笔迹的连笔和重叠等，导致识别率不高。然而，深度学习的兴起，尤其是卷积神经网络(CNN)的发展，为手写数字的精准识别提供了强大的工具。  
AI模型说明：提供的模型“mnist.onnx”是基于卷积神经网络训练得到的，专门用于进行手写数字的识别应用。该模型的使用交互流程为：  
1)加载模型“mnist.onnx”;  
2)加载一张本地手写数字图片“img\_test.png”，并进行预处理图像以符合模型输入要求；  
3)使用mnist模型对手写数字图片进行识别；  
4)输出识别后的数字。  
你作为一名人工智能训练师，请完成以下工作任务：  
（1）补全该模型的使用交互流程对应的Python代码（3.2.2.ipynb），实现本地测试图片“img\_test.png”的识别，将其识别结果截图保存为jpg格式文件，命名为3.2.2-1.jpg。  
（2）在上面的使用交互流程基础上，给出在手写数字识别系统中使用“mnist.onnx”模型的一种人机交互的最优流程，将其保存为docx文件，命名为3.2.2.docx。  
所有结果文件储存在桌面新建的考生文件夹中，文件夹命名为“准考证号+身份证号后六位”。

**3.**技能要求

（1）能确保模型在单一场景下稳定运行；   
（2）能通过分析，设计单一场景下人工和智能交互的最优流程。

**4.**质量指标

（1）模型运行稳定，使用正常；   
（2）单一场景下人工和智能交互的最优流程切实可行。

**人工智能训练师（三级）操作技能考核**

**试题评分表**

准考证号：

试题代码：

试题名称：手写数字识别系统交互流程设计

考核时间：20min

测量分评分表

| 细则编号 | 配分 | 评分细则描述 | 规定或  标称值 | 结果或  实际值 | 得分 |
| --- | --- | --- | --- | --- | --- |
| M1 | 2 | 模型加载代码正确得2分； | 根据数据 |  |  |
| M2 | 2 | 加载图像正确得2分； | 根据数据 |  |  |
| M3 | 2 | 调整大小为MNIST模型的输入尺寸代码正确得2分； | 根据数据 |  |  |
| M4 | 2 | 转为numpy数组代码正确得2分； | 根据数据 |  |  |
| M5 | 2 | 添加batch维度代码正确得2分； | 根据数据 |  |  |
| M6 | 2 | 添加通道维度代码正确得2分； | 根据数据 |  |  |
| M7 | 2 | 返回模型输入列表代码正确得2分； |  |  |  |
| M8 | 2 | 执行预测代码正确得2分 |  |  |  |
| M9 | 2 | 获取预测结果代码正确得2分； | 根据数据 |  |  |
| M10 | 1 | 截图3.2.2-1.jpg中预测类别结果正确：得1分； | 根据数据 |  |  |
| M11 | 1 | 人机交互最优流程（不少于3条）正确得1分； | 根据数据 |  |  |
| 合计配分 | 20 | 合计得分 | |  |  |

**参考答案**：

源代码（3.2.2.ipynb）：  
import onnxruntime

import numpy as np

from PIL import Image

# 加载ONNX模型 2分

ort\_session = onnxruntime.InferenceSession("mnist.onnx")

# 加载图像 2分

image = Image.open("img\_test.png").convert('L') # 转为灰度图

#图像预处理 4分

image = image.resize((28, 28)) # 调整大小为MNIST模型的输入尺寸1分

image\_array = np.array(image, dtype=np.float32) # 转为numpy数组1分

image\_array = np.expand\_dims(image\_array, axis=0) # 添加batch维度1分

image\_array = np.expand\_dims(image\_array, axis=0) # 添加通道维度1分

#使用模型对图片进行识别 2分

ort\_inputs = {ort\_session.get\_inputs()[0].name: image\_array}

# 执行预测 2分

ort\_outs = ort\_session.run(None, ort\_inputs)

# 获取预测结果 2分

predicted\_class = np.argmax(ort\_outs[0])

# 输出预测结果

print(f"Predicted class: {predicted\_class}")

3.2.2-1.jpg



3.2.2.docx  
人机交互最优方式为（仅需回答4个最优方式）：

1.用户界面设计，提供画布供用户手写输入或上传图片，按钮清晰标注“识别”和“清除”。

2.模型预加载，系统启动时自动加载“mnist.onnx”模型，初始化预处理参数，确保首次识别零延迟。

3.实时交互与预处理，用户手写或上传图片后，自动裁剪、缩放至28×28像素，灰度化并归一化像素值。

4.即时识别与展示，后台调用模型推理，0.5秒内返回识别结果，显示预测数字及置信度（如“识别结果：7，置信度：98%”）。

5.动态反馈机制，用户可点击“纠错”提交错误案例，触发模型增量学习优化。

6.轻量级帮助系统，侧边栏嵌入“使用指南”图标，一键查看示例图片格式要求及常见问题解答。

7.性能与安全优化，采用Web Workers异步处理任务，响应速度<1秒；用户数据本地处理，不上传服务器。

8.无障碍支持，界面支持高对比度模式，键盘快捷键操作（如空格键清空画布）。