**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：图像识别评估系统交互流程设计

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

图像识别评估系统是在深度学习技术日益成熟的背景下发展起来的，旨在解决传统图像识别方法在面对复杂场景和大规模数据集时的局限性。随着互联网和物联网技术的飞速发展，图像数据量呈指数级增长，对图像内容的自动理解和智能分析提出了更高的要求。ResNet作为一种深度卷积神经网络架构，凭借其深度残差连接机制，能够有效缓解梯度消失问题，实现更深层次的网络结构，从而捕获更加丰富和抽象的图像特征，极大地提高了图像识别的准确性和效率，推动了人工智能技术在现实世界中的广泛应用和商业化进程。  
AI模型说明：“resnet.onnx”模型是使用 Pytorch 框架和基于深度卷积神经网络网络训练得到的，专门用于进行图像识别。对应的标签文件为“labels.txt”。该模型的使用交互流程为：  
1)加载“resnet.onnx”模型和“labels.txt”类别标签；  
2)加载本地测试图片“img\_test.jpg”，并进行预处理图像以符合模型输入要求；  
3)使用“resnet.onnx”模型对加载的图片进行识别；  
4)输出加载图片的识别结果（输出概率值最大的5组类别和对应概率值）  
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（3.2.1.ipynb），实现本地测试图片“img\_test.jpg”的识别，将其识别结果截图保存为jpg格式文件，命名为3.2.1-1.jpg。  
（2）在上面的使用交互流程基础上，给出在图像识别评估系统中使用“resnet.onnx”模型的一种人机交互的最优方式，将其保存为docx文件，命名为3.2.1.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，找到单一场景下人工和智能交互的最优方式。

**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方式切实可行。

**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：图像识别评估系统交互流程设计

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 | 从指定路径加载一张图片代码正确得2分； | 根据数据 |  |  |
| M3 | 2 | 预处理图片代码正确得2分； | 根据数据 |  |  |
| M4 | 2 | 使用模型对图片进行识别代码得2分； | 根据数据 |  |  |
| M5 | 2 | 获取识别分类后的概率代码正确得2分； | 根据数据 |  |  |
| M6 | 3 | 获取最高的5个概率和对应的类别索引代码正确得3分； | 根据数据 |  |  |
| M7 | 1 | 本地保存识别后的截图3.2.1-1.jpg：得1分； | 根据数据 |  |  |
| M8 | 5 | 截图3.2.1-1.jpg中识别的前五个类别和概率正确：每个类别和概率正确得1分，总计得5分； | 根据数据 |  |  |
| M9 | 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（3.2.1.ipynb）：  
import onnxruntime as ort

import numpy as np

import scipy.special

from PIL import Image

# 预处理图像

def preprocess\_image(image, resize\_size=256, crop\_size=224, mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]):

image = image.resize((resize\_size, resize\_size), Image.BILINEAR)

w, h = image.size

left = (w - crop\_size) / 2

top = (h - crop\_size) / 2

image = image.crop((left, top, left + crop\_size, top + crop\_size))

image = np.array(image).astype(np.float32)

image = image / 255.0

image = (image - mean) / std

image = np.transpose(image, (2, 0, 1))

image = image.reshape((1,) + image.shape)

return image

# 模型加载 2分

session = ort.InferenceSession('resnet.onnx')

# 加载类别标签

labels\_path = 'labels.txt'

with open(labels\_path) as f:

labels = [line.strip() for line in f.readlines()]

# 获取模型输入和输出的名称

input\_name = session.get\_inputs()[0].name

output\_name = session.get\_outputs()[0].name

# 加载图片 2分

image = Image.open('img\_test.jpg').convert('RGB')

# 预处理图片 2分

processed\_image = preprocess\_image(image)

# 确保输入数据是 float32 类型

processed\_image = processed\_image.astype(np.float32)

# 进行图片识别 2分

output = session.run([output\_name], {input\_name: processed\_image})[0]

# 应用 softmax 函数获取概率 2分

probabilities = scipy.special.softmax(output, axis=-1)

# 获取最高的5个概率和对应的类别索引 2分

top5\_idx = np.argsort(probabilities[0])[-5:][::-1]

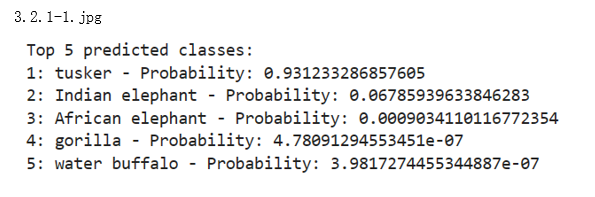
top5\_prob = probabilities[0][top5\_idx]

# 打印结果

print("Top 5 predicted classes:")

for i in range(5):

print(f"{i+1}: {labels[top5\_idx[i]]} - Probability: {top5\_prob[i]}")



3.2.1.docx  
人机交互最优方式为（仅需回答2个最优方式）：

1. 用户界面设计：提供一个简洁直观的用户界面，方便用户上传图片和查看识别结果。
2. 模型加载：系统启动时预加载模型和标签，减少用户等待时间。
3. 图像上传与预处理：允许用户上传图片，并自动进行必要的预处理。
4. 图像识别：后台使用“resnet.onnx”模型处理预处理后的图片。
5. 结果展示：清晰展示识别结果，包括最可能的类别和概率。
6. 交互反馈：提供用户反馈选项，以改进模型性能。
7. 帮助与支持：内置帮助文档和客服支持，提升用户体验。
8. 性能优化：优化系统响应速度和模型处理能力。
9. 安全性考虑：确保用户数据安全和隐私保护。
10. 多语言支持：界面支持多语言，满足全球用户需求。

可访问性：确保UI无障碍，方便所有用户操作。