**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：面部表情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

面部表情识别系统是一种先进的计算机视觉技术，它能够分析人脸的微表情，识别出诸如快乐、悲伤、惊讶等基本情绪。通过捕捉和解读面部特征，如眼睛、眉毛和嘴部的动作，这类系统能在实时或预录的视频中判断人的情感状态，广泛应用于人机交互、市场调研、医疗健康监测、安全监控及教育科技等多个领域，为提升用户体验、增进情感智能和优化社会服务提供了有力工具。  
AI模型说明：提供的已训练的模型“emotion-ferplus.onnx”，其专门用于进行面部表情识别。定义情感类别与数字标签的映射表为{'neutral':0, 'happiness':1, 'surprise':2, 'sadness':3, 'anger':4, 'disgust':5, 'fear':6, 'contempt':7}。  
该模型的使用交互流程为：  
1)加载模型“emotion-ferplus.onnx”和加载情感类别与数字标签的映射表；  
2)加载一张本地图片“img\_test.png”，并预处理图像；  
3)使用已训练的模型对图片面部表情识别；  
4)输出识别后的表情标签。  
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（3.2.3.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3.2.3-1.jpg。  
（2）在上面的使用交互流程基础上，给出在面部表情识别系统中使用“emotion-ferplus.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3.2.3.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：面部表情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 | 3 | 定义情感类别与数字标签的映射表代码正确得3分； | 根据数据 |  |  |
| M2 | 3 | 模型加载代码正确得3分； | 根据数据 |  |  |
| M3 | 3 | 加载本地图片并进行预处理代码正确得3分 ； | 根据数据 |  |  |
| M4 | 3 | 运行模型，进行预测代码正确得3分； | 根据数据 |  |  |
| M5 | 3 | 解码模型输出，找到预测概率最高的情感类别代码正确得3分； | 根据数据 |  |  |
| M6 | 3 | 根据预测的标签找到对应的情感名称代码正确得3分； | 根据数据 |  |  |
| M7 | 1 | 截图3.2.3-1.jpg中识别的面部情绪正确：得1分； | 根据数据 |  |  |
| M8 | 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（3.2.3.ipynb）：  
import onnx

import numpy as np

from PIL import Image

import onnxruntime as ort

import matplotlib.pyplot as plt

# 定义预处理函数

def preprocess(image\_path):

input\_shape = (1, 1, 64, 64)

img = Image.open(image\_path).convert('L') # 补全：Image.open

img = img.resize((64, 64), Image.ANTIALIAS)

img\_data = np.array(img, dtype=np.float32)

img\_data = np.expand\_dims(img\_data, axis=0) # 添加 batch 维度

img\_data = np.expand\_dims(img\_data, axis=1) # 添加 channel 维度

assert img\_data.shape == input\_shape, f"Expected shape {input\_shape}, but got {img\_data.shape}"

return img\_data

# 定义情感类别与数字标签的映射表

emotion\_table = {'neutral':0, 'happiness':1, 'surprise':2, 'sadness':3, 'anger':4, 'disgust':5, 'fear':6, 'contempt':7}

# 加载模型

ort\_session = ort.InferenceSession('emotion-ferplus.onnx') # 使用onnxruntime创建一个会话，用于加载并运行模型

# 加载并预处理图片

input\_data = preprocess('img\_test.png')

# 准备输入数据，确保其符合模型输入的要求

ort\_inputs = {ort\_session.get\_inputs()[0].name: input\_data} # ort\_session.get\_inputs()[0].name 是获取模型的第一个输入的名字

# 运行模型预测

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

# 解码模型输出，找到预测概率最高的情感类别 3分

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

# 根据预测的标签找到对应的情感名称 3分

predicted\_emotion = list(emotion\_table.keys())[predicted\_label]

# 输出预测的情感

print(f"Predicted emotion: {predicted\_emotion}")

3.2.3.docx  
人机交互最优方式为（仅需回答6个最优方式）：

1.用户界面设计

提供实时摄像头预览+上传图片双入口，按钮标注“拍照识别”和“上传图片”，界面突出情感标签（如表情符号+文字）。

2.模型预加载

系统启动时加载“emotion-ferplus.onnx”模型及情感标签表，初始化预处理参数，首帧识别延迟<0.3秒。

3.智能预处理

自动检测人脸区域，裁剪对齐后缩放至64×64灰度图，动态直方图均衡化增强对比度。

4.实时识别与反馈

后台异步推理，返回Top-3情感标签及概率（如“Happiness: 92%”），支持动态刷新（每秒10帧）。

5.交互优化

用户点击“结果纠错”可标记错误案例，触发模型微调；支持语音播报结果（如“检测到开心情绪”）。

6.性能与安全

本地化处理数据，浏览器端WebAssembly加速，识别耗时<200ms；禁用云端存储，隐私数据零上传。

7.无障碍设计

键盘快捷键控制（Enter拍照/Esc清空），高对比度模式适配视障用户，多语言界面一键切换。