**2025 年河北省职业院校技能大赛**

**多信息智能识别控制的技术与应用（高职组）**

**竞赛样题**

第一部分 智能识别应用

**任务一：OpenCV 噪声去除**

**新建 Python 项目，通过编写代码运行实现以下内容**

**任务描述：**

1. 加载竞赛提供的“竞赛资料\第一部分\任务一”中提供的图片，使用 OpenCV 颜色转换算法将图像转换为灰度图像。使用自适应阈值法处理算法将图像转换为二值图像，将处理后的图像保存到“提交资料\第一部分\任务二\1”目录下。
2. 检测出图像中的所有直线。将直线绘制到原始图像上，并将结果保存到“提交资料\第一部分\任务一\2”目录下。
3. 使用礼帽操作进行去噪处理，并将处理后的图片保存到 “提交资料\第一部分\任务一\3” 目录下。请在这个目录中提供处理前和处理后的图片。
4. 尝试调整礼帽操作的核大小并比较不同参数下的处理结果。将最终的核大小和输出结果保存到 “提交资料\第一部分\任务一\4” 目录下。请在文件夹中提供至少两种不同核大小的处理结果。

**完成后，保存本工程所有代码到竞赛U盘“提交资料\第一部分\任务一”目录下。**

**任务二：OpenCV 图像处理**

**新建 Python 项目，编写代码调用 OpenCV 相关接口运行实现对“竞赛资料\第一部分\任务二”中提供的视频文件的以下操作：**

1. 加载竞赛提供的“竞赛资料\第一部分\任务二”中提供的图片mohu.jpg，使用高斯算法进行去噪处理，并将处理后的图片保存到 “提交资料\第一部分\任务二\3”目录下。请在这个目录中提供处理前和处理后的图片。
2. 对去噪后的图片mohu.jpg采用不同的边缘检测算法（两种）检测其边缘，比较检测效果，将处理后的图像以算法名称保存到“提交资料\第一部分\任务二\2”目录下。
3. 加载竞赛提供的“竞赛资料\第一部分\任务二”中提供的图片map.jpg，进行轮廓检测操作，使用红色边框在原图上标记地图轮廓。将绘制出轮廓的图片同名保存到“提交资料\第一部分\任务二\3”目录下。
4. 加载竞赛提供的“竞赛资料\第一部分\任务二”中提供的图片map.jpg，使用跟踪栏（Trackbar）选择通选择并显示图像的BGR通道，跟踪栏的值从0到3，分别对应于BGR三个通道和原图，将运行结果截图分别以“B”、“G”、“R”、“Original”保存到“提交资料\第一部分\任务二\4”目录下。

**完成后，保存本工程所有文件及代码到竞赛U盘“提交资料\第一部分\任务二”目录下。**

第二部分 智能数据处理

**任务二：OpenCV 视频处理**

**新建 Python 项目，编写代码调用 OpenCV 相关接口运行实现对“竞赛资料\第一部分\任务二”中提供的视频文件的以下操作：**

1. 加载竞赛提供的“竞赛资料\第二部分\任务二”中提供的视频文件“car．mp4”，使用MOG背景减除算法检测视频中运动的小车。并将结果保存到竞赛U盘“提交资料＼第二部分＼任务二＼1”目录下。
2. 对背景减除的结果使用开运算，去除检测结果中的噪声，将结果保存到竞赛U盘“提交资料＼第二部分＼任务二＼2”目录下。
3. 对去噪后的结果进行轮廓查找，使用蓝色矩形框对运动的小车进行标注，将处理后的视频保存到竞赛U盘“提交资料＼第二部分＼任务二＼3”目录下。

**完成后，保存本工程所有代码到竞赛 U 盘“提交资料\第二部分\任务二”目录中。**

**任务二：图像特征匹配与分析**

**新建 Python 项目，通过使用 OpenCV 对“竞赛资料\第二部分\任务二”中提供的图片进行以下操作：**

**任务描述：**

1. 对“竞赛资料\第二部分\任务二”中提供的2张图像进行二值化操作，将二值化处理后的图片保存，并提交到“提交资料\第二部分\任务二\1”目录下。
2. 对“竞赛资料\第二部分\任务二”中提供的2张图像使用ORB进行特征点检测，使用暴力匹配器进行特征点的1对1特征匹配操作，将处理后的2张图片的匹配特征点使用绿色直线连线保存，并提交到“提交资料\第二部分\任务二\2”目录下。
3. 对上述匹配结果的数量进行判断，若匹配数量大于10，则查询图1在图2中的位置，并在图2中使用白色矩形框绘制图1的范围，将处理后的结果提交到“提交资料\第二部分\任务二\3”目录下。

**完成后，保存本工程所有代码到竞赛 U 盘“提交资料\第二部分\任务二”目录下。**

第三部分：智能模型构建

**任务一：构建手势（石头、剪刀、布）识别模型**

**任务说明：U盘\资料\第三部分\任务一（根据该目录给定资源完成以下任务）**

**任务描述：**

1. 输出数据集的大小和手势类别数量，并确保截图中包含这些信息。将代码以及运行结果截图保存到“U盘\提交资料\答题卡.docx的指定位置上”。
2. 加载给定数据集，使用代码将数据集划分为训练集.验证集和测试集，保证数据集划分的随机性和合理性。将代码截图保存到“U盘\提交资料\答题卡.docx的指定位置上”。
3. 调整输入图像的尺寸为统一的大小（224\*224），以确保所有图像都具有相同的尺寸，随机输出两张调整后的图像。将代码以及运行结果截图保存到“U盘\提交资料\答题卡.docx的指定位置上”。
4. 对调整后数据集进行必要的预处理操作至少包括归一化.随机翻转、平移、随机裁剪、随机缩放操作。将代码保存到“U盘\提交资料\答题卡.docx的指定位置上”。
5. 定义损失函数（Cross Entropy Loss）和优化器（Adam），以及评估指标（Accuracy），确保与手势识别问题相匹配。将代码以及运行结果截图保存到“U盘\提交资料\答题卡.docx的指定位置上”。
6. 输出训练过程中的每一轮的准确率值，并在截图中标注最高准确率值并确保模型在测试集上的准确率达到一定要求，不低于0.85。将代码以及运行结果截图保存到“U盘\提交资料\答题卡.docx的指定位置上”。
7. 将准确率最高的手势识别模型保存为model.h5文件，保存在当前文件夹下，确保文件名和保存位置正确。将代码以及运行结果截图保存到“U盘\提交资料\答题卡.docx的指定位置上”。
8. 将完整的项目文件结构截图保存，确保包含所有项目文件和目录。将截图保存到“U盘\提交资料\答题卡.docx的指定位置上”。

**完成后，将本任务中所以文件保存在“U盘\提交资料\第三部分\任务一”。**

**任务二：预测手势识别模型**

**任务说明：U盘\第三部分\任务二（根据该目录给定文件完成以下任务）**

**任务描述：**

1. 加载第三部分任务一中已保存的h5模型，将h5模型转换为TensorFlowlite模型，并将模型文件保存到当前文件夹下，将转换后的model.tflite模型的代码截图保存到“U盘\提交资料\答题卡.docx的指定位置上”。
2. 合理编写代码获取输入和输出张量和手势标签映射。将编写的代码截图保存到“U盘\提交资料\答题卡.docx的指定位置上”。
3. 加载‘test.jpg’并用OpenCV分类器检测器检测图片中的手势并以绿框进行标注。将编写的代码截图保存到“U盘\提交资料\答题卡.docx的指定位置上”。
4. 将预测后的手势图片，使用OpenCV将预测的标签和对应的概率添加到图片上，并将图片保存到当前目录的子文件夹finish文件夹中。并将编写的代码截图保存到“U盘\提交资料\答题卡.docx的指定位置上”。

**完成后，将完整的项目文件和代码保存到“U盘\提交资料\第三部分\任务二”中。**

第四部分：智能模型部署

**任务一：手势识别控制系统部署**

**任务说明：U盘\第四部分\任务一（根据该目录给定文件完成以下任务）**

**任务描述：**

1. 调用“资料\第四部分\任务一”中的手势识别模型，或使用“第三部分\任务二 ”中训练出的模型，控制RGB三色灯。
2. 使用摄像头对手势进行实时监测，实现：

* 检测到手势“拳头”时 RGB三色灯亮起红色。
* 检测到手势“剪刀”时，RGB三色灯亮起绿色。
* 检测到手势“布”时，RGB三色灯亮起蓝色后，开启风扇。

1. 将运行结果截图保存至“U盘\提交资料\答题卡.docx的指定位置上”。

**完成后，将代码文件保存到“U盘\提交资料\第四部分\任务一”中。**

**任务二：车牌识别播报系统部署**

**任务说明：U盘\第四部分\任务二（根据该目录给定文件完成以下任务）**

**任务描述：**

1. 调用“资料\第四部分\任务二”中的车牌识别模型，控制LCD1602显示器。
2. 开启摄像头对车牌进行检测，实现：

* 当检测到车牌时，识别出车牌的内容之后终端打印出识别的车牌
* 在LCD显示屏上显示出识别到的车牌信息。
* 对比车牌信息，是否为已知车牌，若为已知车牌亮起绿灯，打开门锁。若为未知车牌，则亮起红灯蜂鸣器响起。

1. 将运行结果以及代码截图保存在“U盘\提交资料\答题卡.docx的指定位置上”。

**完成后，将代码文件保存到“U盘\提交资料\第四部分\任务二”中。**

第五部分 智能文档编写

完成对第五部分“手势识别控制系统部署”和“车牌识别播报系统部署”的使用说明编写，内容需包括“系统运行要求”、“系统功能介绍”、“系统运行步骤”，文档格式要求统一美观，字数不低于100字，完成后将文档命名保存为“手势识别控制系统使用说明.docx”和“车牌识别系统使用说明.docx”，提交保存至“U盘\提交资料\第五部分”。