

# 2023 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 图片视频识别

新建 Python 项目，通过编写代码运行实现以下内容

1. 根据 “竞赛资料\第一部分\任务一\1” 中提供的视频文件，使用已有的 OpenCV 行人检测分类器，对视频中的车辆进行检测。使用红色矩形框出检测到的车辆，并在框的中心点使用红色圆圈进行标注。将标注处理后的视频保存到 “提交资料\第一部分\任务一\4” 目录下。
2. 根据 “竞赛资料\第一部分\任务一\2” 中提供的图片数据集，使用已有的 OpenCV 缺陷检测分类器，对图像中的电子元器件进行检测。使用绿色矩形框出检测到的元器件，并在框的右上角使用白色文本标注缺陷类型。将标注处理后的图片集保存到 “提交资料\第一部分\任务一\5” 目录下。
3. 根据 “竞赛资料\第一部分\任务一\3” 中提供的视频文件，使用已有的 OpenCV 动作检测分类器，对视频中的足球运动员进行检测。

使用蓝色椭圆框出检测到的运动员，并在框的中心点使用黄色点进行标注。将标注处理后的视频保存到 “提交资料\第一部分\任务一\3” 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\任务一” 目录下。

## 任务二：OpenCV 图像识别

新建 Python 项目，通过编写代码运行实现以下内容

1. 加载竞赛提供的“竞赛资料\第一部分\任务二”中提供的 2 张图片，使用 OpenCV 颜色转换算法将图像转换为灰度图像。使用二值化算法将图像转换为二值图像，并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\任务二\1” 目录下。
2. 给定一个有噪声的黑白图像，将图像读取到 Python 中，并将其转换为灰度图像，将处理后的图像保存到 “提交资料\第一部分\任务二\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\任务二\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\任务二\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\任务二”目录下。

## 第二部分 智能数据处理

### 任务一：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\第二部分\任务一”中提供的数据集的以下操作：

1. 将数据集中所有图片都缩放到的长宽为 480\*270 像素，并同名保存到“提交资料\第二部分\任务一\1”目录下。
  2. 任选 2 张缩放后猫狗图片进行图像融合操作，要求 dog 占比 0.8，cat 占比 0.2，保存处理后的图片到“提交资料\第二部分\任务一\2”目录下。
  3. 将数据集的所有图片进行水平镜像翻转，并同名保存到“提交资料\第二部分\任务一\3”目录下。
  4. 对 mohu.jpg 进行中值滤波操作，并同名保存到“提交资料\第二部分\任务一\4”目录下。
  5. 将数据集的所有图片进行的颜色空间转换，要求将 RGB 图像转换为 HSV 图像，并同名保存到“提交资料\第二部分\任务一\5”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\任务一”目录下。

## 任务二：OpenCV 文本处理识别

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

1. 对“竞赛资料\第二部分\任务二”中提供的文本图像，将图像读取到 Python 中，并使用 OpenCV 进行灰度处理和二值化，优化图像质量，将处理后的图像保存，并提交到“提交资料\第二部分\任务二\2”目录下。
2. 使用 OpenCV 进行文本识别来识别图像中的文字，在图像上将识别到的文字区域用红色框标出，并保存标注后的图片到“提交资料\第二部分\任务二\2”目录下。
3. 使用 OpenCV 进行文本识别来识别图像中的文字，在图像上将识别到的文字打印展示在对应区域的下方，并保存处理后的图片到“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”目录中。

## 第三部分 智能模型构建

## 任务一：Fashion\_mnist 分类

在“竞赛资料\第三部分\任务一”提供的 TensorFlow 模型构建代码中找到并修改设定以下参数：

1. 读取 `keras.datasets` 内置数据集 `Fashion_mnist`，使用 `matplotlib` 以子图的形式查看 10 张 `Fashion_mnist` 图片，将查看的图片截图保存到答题卡指定位置。
2. 使用 `Tensorflow` 构建神经网络，要求至少包含 4 个卷积层 (`Conv`)，并使用 `model.summary()` 将模型结构打印输出，将输出的模型结构截图保存到答题卡指定位置。
3. 定义损失函数和优化器，以及对数据进行数据增强操作，开始模型训练，将训练过程截图保存到答题卡指定位置。
4. 完成训练后使用 `matplotlib` 绘制 `accuracy` 变化图和 `loss` 变化图并输出最高的 `accuracy` 值，要求不低于 (0.8)，将 `accuracy` 图以及输出的 `accuracy` 值截图保存到答题卡指定位置。

完成后，将每一步的代码截图保存至竞赛 U 盘中答题卡的指定位置上，参数要求合理以确保能在比赛时间内完成模型训练。

## 任务二：车牌识别模型训练

根据任务一修改完毕的代码，合理优化模型完成模型训练，训练要求如下：

1. 搭建模型训练，要求至少包含 6 个卷积层（Conv）和 4 个池化层，并使用 `model.summary()` 将模型结构打印输出，将输出的模型结构截图保存到竞赛 U 盘中答题卡的指定位置上。
  2. 开始模型训练，对于训练好的所有模型文件进行保存，要求指定模型格式为 hdf5，并保存到竞赛 U 盘“提交资料\第三部分\任务二\2”目录下。
  3. 使用训练好的模型进行车牌号识别，在图像上将识别到的文字区域用红色框标出，并保存标注后的图片到“提交资料\第二部分\任务二\3”目录下。
  3. 使用训练好的模型进行识车牌上的文字，在图像上将识别到的文字打印展示在对应区域的下方，并保存处理后的图片到“提交资料\第二部分\任务二\4”目录下。
- 完成后，保存本工程所有文件和代码到 U 盘“提交资料\第三部分\任务二”目录中。

### **任务三：模型调用与预测**

**新建 Python 项目，通过编写代码实现以下功能**

1. 调用任务二中训练保存的 hdf5 模型。
2. 选择准确率较高的模型对“竞赛资料\第三部分\任务三”中提供的图片数据集(待预测图片集)进行逐一进行预测并将预测车牌号用绿色字体打印保存在验证集图片上，将预测完毕的验证集保存到竞赛 U 盘“提交资料\第三部分\任务三”中。

完成后，保存本工程所有代码及模型到竞赛 U 盘“提交资料\第三部分\任务三”目录中。

## 第四部分 智能模型部署

### 任务一：表情灯控系统部署

根据“竞赛资料\第五部分\任务一”中提供的文件，按照下述步骤在竞赛设备上完成表情灯控系统的项目部署。

1. 在局域网内完成路由器网络的配置，设定无线网络名称为“JNDS-XX”，XX 为你的工位号（如 JNDS-01），八位密码自行设定；设定无线网络 DHCP 服务的开始地址为 192.168.x.100，结束地址为 192.168.x.199，x 为你的工位号（如 192.168.1.100–192.168.1.199）；完成后根据答题卡要求将 WiFi 名称、密码和设置截图填写在答题卡的指定位置上。
2. 根据“AI 实训平台使用教程”，查看 AI 实训平台 IP 地址并修改 `wz_biaoqing.py` 中的 HOST 的 IP 为 AI 实训平台 IP。
3. 根据“AI 实训平台使用教程”，使用指定软件将修改后的 AI 实训平台代码传输至 AI 实训平台，创建并保存在桌面的 test1 文件夹中。
4. 根据“AI 实训平台使用教程”，在 AI 实训平台上运行使用 `rknn_transfer.py` 将提供的 hdf5 模型转换为 rknn 模型，模型同样保存在桌面的 test1 文件夹中。

5. 根据“AI 实训平台使用教程”，在 AI 实训平台上运行修改好的 `wz_biaoqing.py` 调用转换好的 rknn 模型，在竞赛设备上实现摄像头检测人脸表情识别。
6. 修改终端节点代码中的 SERVER 及 WiFi 名称和密码，其中 SERVER 为 AI 实训平台 IP, WiFi 名称和密码为当前 WiFi 信息。
7. 根据“终端节点使用教程”，将修改后的代码下载到 RGB 三色灯模块对应的终端节点上并运行代码。

完成上述步骤后，实现表情灯控系统联动，将所有修改过的代码保存到 U 盘“提交资料\第四部分\任务一”目录中，分析代码并将系统的效果现象记录在答题卡的指定位置上。

## 任务二：红绿灯识别系统终端配置

根据任务一中提供的“终端节点使用教程”及基础示例代码，完成“红绿灯识别系统”的终端功能实现，具体功能如下。

1. 在竞赛平台上找到 RGB 三色灯模块，根据提供的基础示例代码，编写并修改终端节点代码实现以下功能：要求终端可以红绿黄三种颜色模拟红绿灯，当节点底板上按键，按下第一次时，RGB 三色灯亮绿灯；按下第二次时，RGB 三色灯亮黄灯；按下第三次时，RGB 三色灯亮红灯，依此循环。
2. 在竞赛平台上找到语音播报模块，根据提供的基础示例代码，编写并修改终端节点代码实现以下功能：要求终端能够播报当前红绿灯颜色，当节点底板上按键，按下第一次时，语音播报“绿灯，请通行”；



按下第二次时，语音播报“黄灯，请慢行”；按下第三次时，语音播报“红灯，禁止通行”，依此循环。

完成上述功能后将程序代码烧写到对应的终端节点上，实现红绿灯识别终端系统的配置，将所有编写修改的代码保存到竞赛 U 盘“提交资料\第四部分\任务二”目录中。

### 任务三：红绿灯识别系统部署

根据任务一中提供的示例代码，在竞赛平台上完成“红绿灯识别系统”的功能实现及项目部署，实现对监控中红绿灯状态的识别检测并使用 RGB 三色灯模拟灯光、使用语音播报模块提示识别结果。

1. 参考任务一中 AI 实训平台的代码及模型转换代码，修改代码实现将“竞赛资料\第四部分\任务二”中提供的 TensorFlow 模型转换为 rknn 模型；编写代码调用 rknn 模型进行检测，在竞赛设备上调用摄像头实现对赛场提供的红绿灯 A4 纸照片的识别检测播报（允许一定识别误差）；在摄像头画面上方用黄色字体显示“红绿灯识别系统”，在摄像头画面右上角，用红色字体显示识别结果，代码传输至 AI 实训平台的桌面自行创建“test3”目录下运行。

2. 在竞赛平台上找到 RGB 三色灯模块，在任务二的基础上，修改终端节点代码及 AI 平台红绿灯识别代码实现以下功能的增加：当 AI 实训平台程序检测到红灯时，使用 RGB 三色灯模块显示红色；当 AI 实训平台程序检测到绿灯时，使用 RGB 三色灯模块显示绿色；当 AI 实训平台程序检测到黄灯时，使用 RGB 三色灯模块显示黄色。

3. 在竞赛平台上找到语音播报模块，在任务二的基础上，修改终端节点代码及 AI 平台红绿灯识别代码实现以下功能的增加：当 AI 实训平台程序检测到红灯时，使用语音播报模块播报“红灯，禁止通行”；当 AI 实训平台程序检测到绿灯时，使用语音播报模块播报“绿灯，请通行”；当 AI 实训平台程序检测到黄灯时，使用语音播报模块播报“黄灯，请慢行”；播报未完成之前不重新播报，以保证播报的完整性。

完成后，请保持红绿灯识别系统程序的正常运行，以便竞赛结束后裁判进行评分，将所有编写修改的代码保存到竞赛 U 盘“提交资料\第四部分\任务三”的指定目录中。

## 第五部分 智能文档编辑

完成对第四部分“表情灯控系统”和“红绿灯识别系统”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“表情灯控系统使用说明.docx”和“红绿灯识别系统使用说明.docx”，提交保存至竞赛 U 盘“提交资料\第五部分”的指定目录中。