

# 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 分类器识别应用

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”中提供的图片数据集，调用 opencv 中的分类器，对给定图片数据集中的人脸进行检测，要求使用绿色框将人脸标出，红色框将眼睛标出，并将标注后图片集保存到“提交资料\第一部分\任务一\1”目录下。
- 2、根据“竞赛资料\第一部分\任务一\2”中提供的视频数据集，调用 opencv 中的分类器，对视频中的面部微笑进行检测，要求检测到微笑时使用蓝色方框框出，并标注字符‘smile’打印展示在视频中，并将识别标注后的视频保存到“提交资料\第一部分\任务一\2”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，调用 OpenCV 分类器检测视频中的行人并用绿色框进行标注，将框的中心点用红色点进行标注，将识别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 关键点识别

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 `hstack` 方法将其水平拼接，拼接后的图片保存到“提交资料\第一部分\任务二\1”目录下。
  2. 在使用 `hstack` 方法将图片水平拼接的基础之上，使用 Python-OpenCV 中 `detectAndCompute` 函数检测图像的关键特征点，并使用 `cv2.drawKeypoints()` 函数将关键特征点部位绘制一个小圆圈，保存检测后的图片，提交到“提交资料\第一部分\任务二\2”目录下。
  3. 在使用 `hstack` 方法将图片水平拼接的基础之上，使用 KNN 检测识别来自左右图像的 SIFT 特征并对其进行匹配，将两张图像连接起来并使用线段将匹配点之间的连线绘制出来，匹配完成后的图片结果图保存，提交到“提交资料\第一部分\任务二\3”目录下。
  4. 在使用 `hstack` 方法将图片水平拼接的基础之上，计算视角变换矩阵  $H$ ，用  $H$  对右图进行变换并返回全景拼接图像，并将全景拼接后的图片结果进行保存，提交到“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：迷宫寻路处理

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”中提供的迷宫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”中提供的迷宫图像进行轮廓标注操作，将轮廓标注后的图片保存，并提交到“提交资料\第二部分\任务二\2”目录下。
3. 对“竞赛资料\第二部分\任务二”中提供的迷宫图像进行扩张操作，将扩展操作后的图片保存，并提交到“提交资料\第二部分\任务二\3”目录下。
4. 对“竞赛资料\第二部分\任务二”中提供的迷宫图像进行侵蚀操作，将侵蚀处理后的图片保存，并提交到“提交资料\第二部分\任务二\4”目录下。
5. 对“竞赛资料\第二部分\任务二”中提供的迷宫图像分别进行上述操作后，要求找到迷宫的入口到出口的唯一路径并使用红色线条标注在原图之上，将标注出入路径后的图片保存，并提交到“提交资料\第二部分\任务二\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\任务二”目录中。

## 第三部分 智能模型构建

### 任务一：模型参数设定

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. 设定填写 `input_shape` 的值, 查看自己构建模型后的 `Total parmas`, 值。`Total parmas` 代表模型的训练参数个数, `input_shape` 代表图片的输入尺寸（要求最低为 224）。
2. 设定填写 `epochs` 的值。`epochs` 代表训练轮数, 表示完成模型训练过程需要遍历整个训练数据集的次数。
3. 设定填写 `batch_size` 的值。`batch_size` 表示模型每次更新参数所需要的样本数量。
4. 设置训练的损失函数为交叉熵损失函数。
5. 将模型中的所有激活函数设置为“Rectified Linear Unit（修正线性单元）”。
6. 设置模型优化器为自适应学习率的优化器 Adaptive Moment Estimation。

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：猫狗识别模型训练与优化

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. 对提供的猫狗训练集图片进行随机翻转、平移、随机裁剪、随机缩放等数据增强操作，并截图对应代码块保存到竞赛 U 盘中答题卡的指定位置上。
2. 合理调整及优化代码（参考：网络结构、输入图片大小、优化器、batch\_size 等），使得测试集上的得分(test\_score)不低于 85%，并截图保存到竞赛 U 盘中答题卡的指定位置上。
3. 在模型训练中加入回调函数使得能在 tensorboard 中对训练过程进行可视化，运行 tensorboard，查看训练过程中的准确率和损失值的变化图像，并截图保存到竞赛 U 盘中答题卡的指定位置上。
4. 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\4”目录下。
5. 对提供的猫狗数据集重新进行迁移学习训练，模型要求使用 vgg19 网络，将 vgg19 网络结构图及训练过程结果对应截图保存到竞赛 U 盘中答题卡的指定位置上。
6. 舍弃 vgg19 原来的全连接层，自己添加新的全连接层训练，将对应的截图保存到竞赛 U 盘中答题卡的指定位置上。
7. 在 vgg19 网络训练中加入回调函数，要求在模型 ACC 达到 85%时停止训练并保存模型，将对应的截图保存到竞赛 U 盘中答题卡的指定位置。

置上，模型保存位置为竞赛 U 盘“提交资料\第三部分\任务二\7”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 模型或训练好的 vgg19 网络模型。
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-300)进行逐一进行预测并将预测结果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\第四部分\任务一”中提供的文件，按照下述步骤在竞赛设备上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，设定无线网络的名称为“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：车牌识别系统终端配置

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“车牌识别系统”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：要求终端可以播报“京”、“闽”、“粤”、“苏”、“沪”、“浙”等地的车牌，当节点底板上按键，按下奇数次时，语音播报“京 A • 12345”；按下偶数次时，语音播报“闽 B • 54321”模拟车牌播报；

2. 在竞赛平台上找到点阵显示模块，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：要求终端可以显示“京”、“闽”、“粤”、“苏”、“沪”、“浙”等地的车牌，当节点底板上按键，按下奇数次时，点阵显示“京 A • 12345”；按下偶数次时，点阵显示“闽 B • 54321”模拟车牌显示；

完成上述功能后将程序代码烧写到对应的终端节点上，实现车牌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：车牌识别系统部署

根据任务一中提供的示例代码，在竞赛平台上完成“车牌识别系统”的功能实现及项目部署，实现对监控中车牌的识别检测并使用语音播报模块播报车牌、使用点阵显示模块显示车牌。

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 pb 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提供的已打印车辆图片的车牌检测（允许一定识别误差）；在摄像头画面上方用黄色字体显示“车牌识别系统”，在摄像头画面右上角，用红色字体显示车牌号，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 平台车牌检测代码实现以下功能的增加：当 AI 实训平台程序检测到车牌号时，使用语音播报模块播报当前识别到的车牌号，车牌未播报完成之前不重新播报，以保证车牌播报的完整性。

3. 在竞赛平台上找到点阵显示模块，在任务二的基础上，修改终端节点代码及 AI 平台车牌检测代码实现以下功能的增加：当 AI 实训平台程序检测到车牌号时，使用点阵显示模块显示当前识别到的车牌号，车牌未显示完成之前不重新显示，以保证车牌显示的完整性。

完成后，请保持车牌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“车牌识别系统”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“车牌识别系统使用说明.docx”，提交保存至竞赛 U 盘“提交资料\第五部分”的指定目录中。