



2020 一带一路暨金砖国家技能发 展与技术创新大赛

【首届人工智能训练与应用技能大赛】

竞赛样题任务书（中文版）

金砖国家工商理事会（中方）技能发展工作组
一带一路暨金砖国家技能发展与技术创新大赛组委会
竞赛技术委员会专家组制定

2020 年 11 月 1 日

目录

选手须知:	1
竞赛设备描述:	2
竞赛任务背景:	2
竞赛任务目标:	2
任务一: 人工智能数据集制作.....	3
任务二: 人工智能开发框架应用.....	5
任务三: 人工智能系统装调运维.....	6

2020 一带一路暨金砖国家技能发展与技术创新大赛

人工智能训练与应用技能大赛

样题任务书

选手须知：

1. 任务书共 10 页，如出现任务书缺页、字迹不清等问题，请及时向裁判申请更换任务书。
2. 每个赛位配有人工智能一体机、数据集制作平台、模型训练平台、端侧应用开发平台、待识别商品等。
3. 参赛团队应在 **4 小时 00 分钟** 内完成任务书规定内容；选手在竞赛过程中创建的电子文件（1. 原始采集数据、2. 标注数据集、3. 训练模型）必须存储到“**home/参赛队编号**”文件夹下，文件夹名分别命名为 **data**、**dataset**、**model**，端侧开发板上的文件需存储到端侧开发板上的 **home/pi** 下，并命名文件名为 **参赛队伍编号**。未存储到指定位置的运行记录或程序文件均不予给分，选手应及时存储相关程序文件，由于断电或其他原因引起的程序丢失将由选手自行负责，断电时间可根据情况给予补时。
4. 比赛结束后，选手需将参赛队编号文件夹下的资料**拷贝到指定的 U 盘中**（U 盘需插在一体机上的 USB3.0 口进行拷贝），并在 **U 盘上写上参赛队编号**。拷贝完成后，U 盘上交给裁判进行保存。
5. 选手提交的试卷不得出现学校、姓名等与身份有关的信息，否则成绩无效。
6. 数据标注平台软件登录账号与密码不允许修改（标注账号见竞赛任务材料）。
7. 所有设备在比赛开始后由于错误接线、操作不当等原因引起计算机、端侧开发板、显示器、摄像头等设备的损坏，将依据扣分表进行处理。
8. 选手全程操作一体机的系统选项为_____密码为：**hdkj**

竞赛设备描述：

人工智能训练与应用技能大赛在“人工智能综合实训平台 AITE-S1”上进行，该设备由人工智能一体机、图像采集平台、数据集制作实验平台、模型训练平台、端侧应用开发平台及其他软件系统组成。如下图所示。



图 人工智能综合实训平台

竞赛任务背景：

智能零售作为人工智能技术在零售业的典型应用场景，因其具备无人化、低成本、智能化的应用优势在日常生活中深受欢迎。本次赛项结合智能零售应用场景下的数据集处理、深度学习、机器学习、计算机视觉、端侧人工智能开发等技术技能点，围绕快消品领域的商品识别项目制定竞赛内容，参赛选手将通过完成三个阶段的竞赛任务，最终模拟实现智能零售场景下商品的准确识别。

竞赛任务目标：

- (1) 人工智能数据集制作
- (2) 人工智能开发框架应用
- (3) 人工智能系统装调运维

任务一：人工智能数据集制作

参赛选手需仔细阅读竞赛任务背景与第一阶段“智能零售商品数据集制作任务说明”，并完成以下任务：

1. 根据给定的采集平台地址与账号采集图片，图片采集数量为 70 张，具体采集类别如下：

①单一物品种类采集 10 张，即 3*10 张

②两种物品种类采集 10 张，即 3*10 张

③三种物品种类采集 10 张，即 1*10 张

推流地址：默认 数据采集平台地址：_____

账号：_____ 密码：_____

2. 通过给定的零售商品制定填写商品数据采集方案 1 份、商品数据标注方案 1 份。

3. 参赛选手根据已制定的采集方案与标注方案利用实训平台数据标注软件完成商品数据的采集-清洗-标注任务。

标注平台管理员地址为：_____

账号：_____ 密码：_____

标注平台标注员地址为：_____

账号：_____ 密码：_____

标注平台审核员地址为：_____

账号：_____ 密码：_____

4. 根据给定的标定平台地址与账号完成数据集的制作，选手在标注平台中创建的数据集名称为队伍编号，创建的任务名称命名为零售商品识别。零售商品对应的标签名称参考附件一。

5. 选手需要根据要求将处理后的图片、数据集保存到指定的路径下。

6. 填写第一阶段数据集制作验收报告。

智能零售商品数据集制作任务说明			
任务类型	采集与标框标注	任务时间	60 分钟
图片数量	共 70 张	准确率要求	>90%
<p>任务需求概述：</p> <p>数据集为使用竞赛平台摄像头在竞赛平台上目标识别区域内采集到的不同的商品图片进行标注。标注要求是用矩形框将采集图片内的不同名称商品标注出来，同一名称商品应该对应相同的 id，矩形框的标签为商品的名称 id。</p>			
<p>验收方法及验收标准：</p> <ol style="list-style-type: none">1. 标注框要求尽量与标注目标最小外接矩形贴合，框的大小和位置与标注目标四边尽量贴合，误差超过 5%，则认为该标注框不合格；2. 目标对应的标签错误，或者出现漏标的情况，都认为该标注框不合格；3. 准确率的计算方法：在标注的所有图片中随机抽取 20%进行检查，将样本中不符合标注规范的标注框记为不合格标注框，标注准确率=样本中合格标注框数量/样本标注框总数量			

任务二：人工智能开发框架应用

在完成任务一中人工智能数据集制作的基础上(如果参赛队没有完成任务一，由裁判通知技术人员提供标准数据集，参赛队任务一不得分)，完成如下工作：

1. 参赛选手根据开发环境搭建过程资料安装 mmdetection 目标检测工具包，搭建资料见附件二。

2. 参赛选手根据第一阶段已完成数据集进行物体识别模型训练并进行代码参数修改与优化代码以提高训练精度。

3. 使用训练出来的模型对提供的测试图片进行预测，并得到预测值，得到预测值结果后，选手需向裁判举手示意，由裁判记录该结果。该测试图片存储于 home 路径下，根据抽取的零售商品编号对对应的测试图片进行测试。

3. 选手需要根据要求将训练出来的模型保存到指定的路径下。

4. 填写第三阶段基于深度学习框架的模型训练项目报告。

任务三：人工智能系统装调运维

在完成任务二中人工智能开发框架应用的基础上(如果参赛队没有完成任务二，由裁判通知技术人员提供标准模型，参赛队任务二不得分)，完成如下工作：

1 通过调用第二阶段已训练好的商品识别模型，完成系统软件调试。

2. 选手需要通过放置零售商品实物对 retail 文件代码进行编写，完成软硬件综合运维测试，实现任务要求。retail 文件存储在 home 下。

(1) 通过放置零售商品实物，完成物品的识别及相似度判别。

3. 选手需要根据要求将更改后的代码保存到指定的路径下。

4. 填写第三阶段智能零售商品识别系统装调运维报告。

5. 选手完成第三阶段的任务要求后，需向裁判举手示意，放置商品实物，由裁判记录相关成绩。

6. 总任务结束后，选手需打开标注平台管理员账号界面，选择任务中心中创建的任务，点击查看选项，并将每页显示条数改成 32 条选项，由裁判记录相关数据。

附件一

商品编号一

商品名称	对应标签	串口屏对应价格编号	商品对应指示灯	测试图片
可乐	KELE	1	红灯	test1.jpg
雪碧	XUEBI	2	绿灯	
芬达	FENDA	3	黄灯	

附件二

开发环境搭建过程资料

注：文档中所用的 `mmdetection` 工具包存放在 `home` 下，其余的安装包在 `home/package`

一、安装 NVIDIA 驱动

1.输入 `sudo gedit /etc/modprobe.d/blacklist.conf` 用编辑器打开 `blacklist.conf` 配置文件

2.在文件的最后一行加入下面的命令，屏蔽有影响的驱动包

```
blacklist nouveau
```

```
options nouveau modeset=0
```

加入后按 `ctrl+s` 保存，之后关闭

3.输入 `sudo update-initramfs -u` 更新 linux 内核

4.输入 `reboot` 重新启动

5.重启后，按 `Ctrl+Alt+F1` 进入命令提示符界面，输入对应的 `username` 和 `passwd` 进入命令行

6.输入指令：`sudo service lightdm stop` 关闭图形界面，再使用 `cd` 指令进入包目录，输入 `sudo chmod 755 NVIDIA-Linux-x86_64-440.82.run` 修改 `run` 文件权限

7. 输入 `sudo ./NVIDIA-Linux-x86_64-440.82.run --no-x-check --no-nouveau-check --no-opengl-files` 安装驱动

8.等待后，发现预安装错误询问是否继续，选择 `continue installation`

9.询问是否安装 DKMS，选择 `NO`

10. 警告兼容问题，选择 `OK` 继续

11.询问是否启用 `nvidia` 配置文件，选择 `yes`

12. 安装成功，选择 `OK`

13.输入 `sudo service lightdm start` 打开图形化界面

14.使用 `nvidia-smi` 查询是否成功挂载

二、安装 CUDA

1.使用 `cd` 命令进入目录后，输入 `sudo sh cuda_10.0.130_410.48_linux.run` 开始安装 CUDA

2.阅读安装须知后（使用 `enter` 键换行，使用 `space` 键翻页），选择合适的选项安装 CUDA。

3.安装后，输入 `cat /usr/local/cuda/version.txt` 查询 CUDA 是否安装成功

三、安装 cudnn

1.使用终端进入包目录后，输入 `tar -xzvf cudnn-10.0-linux-x64-v7.6.5.32.tgz`
解压 cudnn 包文件

2.依次输入

```
sudo cp cuda/include/cudnn.h /usr/local/cuda/include
```

```
sudo cp cuda/lib64/libcudnn* /usr/local/cuda/lib64
```

```
sudo chmod a+r /usr/local/cuda/include/cudnn.h /usr/local/cuda/lib64/libcudnn*
```

3.输入 `cat /usr/local/cuda/include/cudnn.h | grep CUDNN_MAJOR -A 2` 查看 cudnn 版本

四、安装 anaconda

1.进入包目录后输入 `bash Anaconda3-2020.02-Linux-x86_64.sh`

2.按下回车后开始阅读安装须知

3.阅读后询问是否接受条款，之后选择安装位置，默认位置安装即可

4.关闭终端后，再次打开终端命令行的前方出现（base）表示安装成功，由于此试验台不使用 Jupyter 命令，所以输入 `conda config --set auto_activate_base False` 关闭 base

5. 输入后再次打开终端，发现前面的 base 消失

五、安装 pytorch

1.输入 `conda create -n hdjk python=3.7`，创建虚拟环境

2.进入包目录后，输入 `conda activate hdjk`，激活虚拟环境，前方有（hdjk）表示激活成功

3.输入 `pip install torch-1.1.0-cp37-cp37m-linux_x86_64.whl` 安装 pytorch

5.安装成功后，输入 `pip install torchvision-0.3.0-cp37-cp37m-linux_x86_64.whl`
安装 torchvision

6.输入 `pip list` 查看是否安装成功

六、安装 mmec

1.进入包目录后，输入 `unzip mmcv-master.zip` 解压 mmcv 包

2.进入 mmcv-master 目录，输入“`pip install .`”安装 mmcv

3.输入 `pip list` 查看是否安装成功

七、安装 mmdetection

1.进入目录后，输入 `python setup.py develop`

2.输入 `pip list` 查看是否安装成功

八、安装其余所需要的库

1.使用 `pip install` 安装其他需要安装的库包括
`appdirs, chardet, click, cyclor, Cython, decorator, Flask, idna, imagecodecs, imageio, imgaug, intel-openmp, mkl, opencv-python-headless, packaging, pooch, pycocotools, setuptools, terminaltables`

之后输入 `pip uninstall opencv-python` 卸载默认安装的 opencv,再输入 `pip install opencv-python` 安装 opencv。

2.安装过后输入 `pip list` 查看环境已安装完毕

九、配置开发环境。

- 1.打开 `pycharm` 软件。出现 `pycharm` 主界面，选择 `Create New Project`。
- 2.在此处我们选择 `Existing interpreter`，之后点击右侧的“...”按钮
- 3.点击 `Conda Environment`，选择之前创建的环境 `hdkj`，之后点击“OK”
- 4.点击后返回 `New Project` 窗口，选择项目路径，之后点击 `Create`，创建后，之前创建的环境即可在 `pycharm` 中进行使用。