



# 2026

金砖国家职业技能大赛（金砖国家未来技能和技术挑战赛）

## 人工智能技术应用

BRICS-FS-56

样题（国际总决赛）

2026 年 06 月



目录

1 赛项简介.....2

    1.1 赛项名称.....2

    1.2 参赛形式.....2

    1.3 赛项描述.....2

2 参赛对象及竞赛模块.....2

3 竞赛内容和时间要求.....4

    3.1 竞赛模块和时间要求.....4

    3.2 任务内容.....4

# 1 赛项简介

## 1.1 赛项名称

2026 金砖国家职业技能大赛（金砖国家未来技能和技术挑战赛）人工智能技术应用赛项，赛项编号 BRICS-FS-56。

## 1.2 参赛形式

本赛项为双人赛，二位选手自行分配工作任务，完成竞赛考核。

## 1.3 赛项描述

2026 年金砖国家职业技能大赛人工智能技术应用赛项聚焦“人工智能+行业”应用场景与技术创新实践，考察人工智能技术在智能制造、智慧医疗、工业质检、舆情分析、低空无人机技术等场景的应用，同时注重选手在人工智能基础理论、技术应用和实操能力展开，旨在考察学生的数据处理、算法设计、编程实现、模型训练等能力。本赛项主要面向新一代信息技术相关专业开展，考察模块具体包括：模块 A：智能图像处理应用开发；模块 B：智能制造与工业质检；模块 C：智能客服与舆情分析；模块 D：AI 低空创新应用四个模块。实现培养国际化、高技能、未来技术技能型人才的目标。竞赛由专业的人工智能技能竞赛平台提供竞赛环境和考核系统，选手通过线上线下方式完成任务考核。本赛项为线上线下双人赛。

# 2 参赛对象及竞赛模块

本赛项的参赛对象为技师学院、中职、高职、应用型本科的在籍学生或教师，

采用 Python 语言，基于开源的 OpenCV、TensorFlow、keras 等国内外主流 AI 框架，使用机器学习经典算法、开源计算机视觉算法、卷积神经网络 CNN、循环神经网络 RNN、长短时记忆网络 LSTM 等技术，完成涉及 OpenCV 图像处理、机器学习、深度学习、自然语言处理技术的算法应用及应用开发。考核内容如下：

#### 模块 A：智能图像处理应用开发

使用 OpenCV 开源计算机视觉库，完成图像的基础操作、图像处理和图像特征提取与分析。包括但不限于读取与显示、图像格式转换、图像基本属性获取、图像保存、图像滤波、颜色空间转换、边缘检测等，重点结合工业质检场景实现表面缺陷分割、尺寸测量等应用。

#### 模块 B：智能制造与工业质检

聚焦智能制造场景中的质量检测需求，运用 YOLOv5/U-Net 等深度学习模型实现工业零件缺陷检测、装配验证等任务。考核内容包括但不限于数据集构建（如公开工业缺陷数据集）、模型训练优化、检测结果可视化及工业质检报告生成，支持 Jupyter Notebook 全流程验证。

#### 模块 C：智能客服与舆情分析

基于自然语言处理技术实现智能客服对话系统与舆情分析平台。考核内容包括但不限于文本预处理、BERT 情感分析、Seq2Seq 对话生成、舆情热点聚类（如 K-Means）及可视化看板开发，支持从社交媒体评论中提取用户反馈并生成舆情报告。

#### 模块 D：AI 低空创新应用

围绕低空经济场景，基于公开无人机数据集（如 DOTA Dataset）实现目标检测、语义分割及聚类分析。考核内容包括但不限于无人机航拍图像目标识别（如车辆/人员追踪）、交通流量优化算法应用、路径规划算法实现及 Jupyter Notebook 中的动态可视化演示，无需硬件支持。

## 3 竞赛内容和时间要求

### 3.1 竞赛模块和时间要求

人工智能技术应用赛项由 4 个模块组成，包括：模块 A：智能图像处理应用开发；模块 B：智能制造与工业质检；模块 C：智能客服与舆情分析；模块 D：AI 低空创新应用，综合考查参赛选手的人工智能基础理论、实践技能和创新能力的提升，进而提升选手职业素养和就业能力。本赛项国际总决赛要求选手在 6 个小时内完成答题，各个模块的答题时间由参赛选手自行分配。

### 3.2 任务内容

任务考核内容为根据给定的项目，严格按照题干要求，在指定位置（\*\*\*位置）编写代码，特别注意：已有代码请勿修改，编写代码时将\*\*\*删掉，并将指定区域（虚线区域内）的代码答案复制到竞赛平台上对应的答题框内。

#### 模块 A：智能图像处理应用开发

案例说明：该项目基于新版 OpenCV 库，围绕经典的 butterfly 图像展开一系列基础图像处理操作，涵盖图像的读取与显示、颜色空间转换（如灰度化）、图像滤波（均值、高斯、中值滤波）、边缘检测、阈值分割等核心功能，旨在通过具体案例展示 OpenCV 在图像处理中的基本应用方法与算法效果，帮助初学者理解并掌握图像处理的基础知识与技术流程。

第 1 题：将彩色图像 `image` 转换为灰度图像 `gray_image`。1 分

第 2 题：使用高斯模糊对灰度图像 `gray_image` 进行平滑处理，生成模糊后的图像 `blurred`。高斯核大小设置为  $5 \times 5$ ，自动计算高斯核在 X 方向的标准差。

第 3 题：使用 Canny 算法提取图像 `blurred` 边缘，并赋值给变量 `edges`。低阈值设置为 100，高阈值设置为 200。

第 4 题：采用固定阈值的方法将图像 `gray_image` 转换为二值图像（黑白图像），返回可忽略的固定阈值和二值化图像 `thresh`。阈值大小设置为 127，最大值设置为 255。

第 5 题：将处理后的图像分别命名为 `gray_image.jpg`, `blurred.jpg`, `edges.jpg`, `thresh.jpg` 保存在 `data` 文件夹中。

### 模块 B：智能制造与工业质检

案例说明：本项目基于 TensorFlow 框架（Keras API），使用预训练的 MobileNetV2 作为特征提取 backbone，在 NEU-DET 钢材表面缺陷数据集上实现六分类缺陷检测（crazing、inclusion、patches、pitted、rolled、scratches）。项目完整覆盖数据加载与预处理、迁移学习（先冻结卷积基训练分类头，再解冻部分层微调）、指数衰减学习率调度、训练与验证循环、训练曲线可视化、模型保存与加载，以及单张图片推理预测的全流程。

第 1 题：使用 `tf.keras.Sequential` 构建完整模型，结构为：`base_model`→`GlobalAveragePooling2D()`→`Flatten`→`Dense(256, activation='relu')`→`Dropout(0.5)`→`Dense(6, activation='softmax')`。

第 2 题：定义分类交叉熵损失函数 `criterion`，Adam 优化器（学习率 0.001）`optimizer`，以及指数衰减学习率调度器（衰减系数 0.9）`scheduler`。

第 3 题：解冻 `base_model` 的所有参数，然后冻结前 50 层的参数（将对应层的 `trainable` 属性设为 `False`）。

第 4 题：创建学习率为 0.0001 的新优化器 `optimizer_fine`，编译模型时仅针对需要训练的参数进行优化。

第 5 题：将训练好的模型参数保存到文件 `'neu_model.h5'` 中。

### 模块 C：智能客服与舆情分析

案例说明：该项目聚焦于利用预训练语言模型，在中文电商评论情感分析数

数据集（涵盖正向与负向二分类维度）上开展迁移学习研究，通过微调轻量级中文 TinyBERT 的序列分类头以适配数据集的二分类任务，并采用带 Warmup 的线性学习率调度与梯度裁剪策略优化模型训练过程，同时针对训练过程中的模型泛化与过拟合问题，提出基于验证集准确率的最优模型参数保存机制以提升推理稳定性，最终通过准确率及预测置信度等指标评估模型效果。

第 1 题：配置 AdamW 优化器，传入模型参数，设置学习率为  $2e-5$ ，epsilon 参数为  $1e-8$ 。

第 2 题：将预热比例 WARMUP\_RATIO 与总训练步数 total\_steps 相乘，并对乘积取整，将结果赋值给变量 warmup\_steps。

第 3 题：从 batch 中提取标签数据，将其转移至 DEVICE 指定的计算设备上，并把结果赋给变量 labels。

第 4 题：将预测结果张量转移至 CPU 并转换为列表，然后将该列表中的元素扩展添加到列表 predictions 中。

第 5 题：使用 load\_weights 方法将最优模型参数载入到 model 中，以恢复模型的权重。

#### 模块 D：AI 低空创新应用

案例说明：该项目聚焦于无人机航拍视角下的小目标行人检测任务，在无人机航拍行人数据集（涵盖大视场、极小尺度特征的单类别检测场景）上开展迁移学习研究，通过引入轻量级 YOLOv8n 作为基础特征提取与检测架构，并加载 COCO 预训练权重以继承通用视觉先验，构建了“冻结主干网络+微调检测头”的微调范式，针对少量训练条件下模型极易欠拟合且特征空间易受干扰的痛点，提出采用高初始学习率配合余弦退火调度策略以加速检测头对无人机视角的适配，同时强制关闭 Mosaic 等强数据增强以防止少样本迭代下的特征对齐混乱，最终通过 mAP@0.5、mAP@0.5:0.95 以及精确率与召回率等核心指标评估模型在无人

机小目标场景下的综合检测效能。

第 1 题：使用 `psutil` 库获取计算机的物理 CPU 核心数量（不包括超线程），并将结果存储在变量 `cpu_count` 中。

第 2 题：调用环境检查函数来验证系统配置是否符合要求。

第 3 题：使用 `random` 模块的 `sample` 函数，从训练图像列表 `train_imgs` 中随机抽取图像，抽取数量取 200 与 `train_imgs` 长度的最小值，并将抽取结果赋值给变量 `sample_imgs`。

第 4 题：使用 OpenCV 的 `imread` 函数读取路径 `img_path` 下的图像，并使用 `cvtColor` 函数结合 `COLOR_BGR2RGB` 参数将其颜色通道从 BGR 转换为 RGB，将最终结果赋值给变量 `img`。

第 5 题：调用可视化真值函数 `visualize_ground_truth`，通过传入关键字参数指定单次可视化的样本数量为 2。

第 6 题：根据配置对象 `cfg` 中的 `MODEL_NAME` 属性实例化 YOLO 类，并将生成的模型对象赋值给变量 `model`。



