AI Cube V1.3

用户指南

免责声明

您购买的产品、服务或特性等应受嘉楠公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，嘉楠公司对本文档内容不做任何明示或默示的声明或保证。由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

商标声明

Canaan 图标、嘉楠和嘉楠其他商标均为嘉楠捷思信息技术有限公司的商标，并归嘉楠股份有限公司所有。本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

版权所有©嘉楠股份有限公司

前言

文档目的

本文档主要介绍AI Cube通用计算平台的使用方式、工作流程，旨在帮助您缩短AI模型训练、部署开发周期，降低用户端侧AI功能开发门槛。

目标读者

本文档主要适用于以下工程师：

- AI产品软硬件开发工程师
- AI技术支持工程师
- AI测试工程师
- AI模型部署工程师

修订记录

修订版本	说明	修订日期
V 1.0.0	初次正式发版	20230928
V 1.1.0	AI Cube v1.1	20231212
V 1.2.0	AI Cube v1.2	20240108
V 1.3.0	AI Cube v1.3	20240329

1 AI Cube 软件概述及安装

1.1 概述

AI Cube 是由嘉楠科技开发的一款通用视觉 AI 计算平台，该平台能够帮助用户迅速训练 AI 模型，同时可以将训练完毕的模型轻松部署到嘉楠芯片中，如 Canaan k230。使用该平台能够以零代码成本实现项目管理、数据集拆分、数据集分布预览、模型训练评估及模型部署。

嘉楠科技成立于 2013 年，致力于人工智能芯片研发。2016 年实现了 16nm 芯片的量产，2018 年实现量产全球首款自主知识产权的 7nm 芯片，以及量产全球首款基于 RISC-V 架构自主知识产权商用边缘 AI 芯片。同时嘉楠科技已使用 AI 技术对多个下游行业进行了赋能，如 AI 医疗、新零售、智慧交通、AI 农业等。

嘉楠科技推出的 AI Cube 通用视觉检测平台以深度学习技术为依托，将人工智能技术与高性能计算技术应用于端侧设备的部署当中。从而缩短任务端开发周期，为产业下游的高速生产提供平台化支持。

AI Cube 由以下几部分组成，如图 1-1 所示。

1. 数据解析模块（支持不同格式数据集）
2. 模型训练（提供通用训练范式）
3. 多任务支持
4. 内置预训练网络
5. 芯片部署资源包
6. PC 模型验证及芯片推理支持

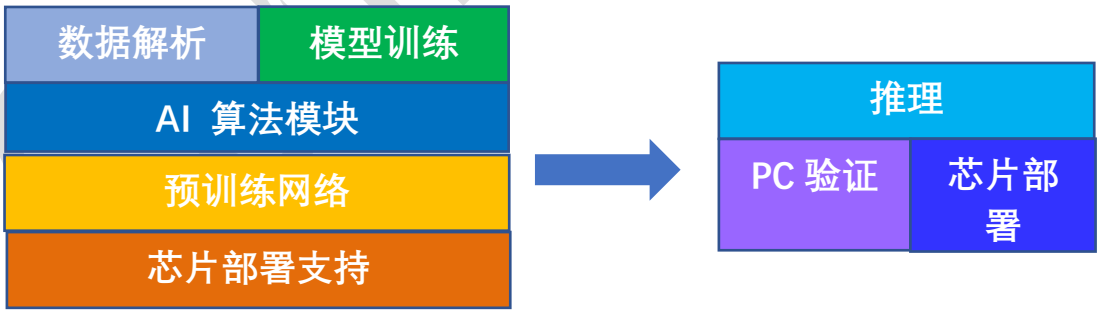

图 1-1 AI Cube 组成结构

1.2 软件安装

软件安装前准备：

1、ubuntu18.04 及以上或 win10 64 位系统以上

2、Nvidia 驱动版本：Nvidia driver version >= 465.89

硬件最低配置：

1、显存：4GB 以上

2、内存：8GB 以上

3、硬盘容量：空闲余量 20GB 以上

硬件推荐配置:

1、显存 8G 及以上

2、内存：16GB 及以上

3、硬盘容量：SSD 空闲余量 100GB 以上

1.2.1 ubuntu dotnet 安装

安装软件前需要在 ubuntu 系统中安装 dotnet-sdk-7.0，具体安装方式如下：

```
wget https://packages.microsoft.com/config/ubuntu/18.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb

sudo dpkg -i packages-microsoft-prod.deb

rm packages-microsoft-prod.deb

sudo apt-get update && \

sudo apt-get install -y dotnet-sdk-7.0
```

注意：不同的 ubuntu 版本安装 dotnet 方式略有不同，具体请参见 https://learn.microsoft.com/zh-cn/dotnet/core/install/linux-ubuntu

Ubuntu	支持的 .NET 版本	在 Ubuntu 源中可用	在 Microsoft 源中可用
23.04	7.0、6.0	7.0、6.0	7.0、6.0
22.10	7.0、6.0	7.0、6.0	7.0、6.0、3.1
22.04 (LTS)	7.0、6.0	6.0	7.0、6.0、3.1
20.04 (LTS)	7.0、6.0	无	7.0、 6.0、5.0、3.1、2.1
18.04 (LTS)	7.0、6.0	无	7.0、 6.0、5.0、3.1、2.2、2.1
16.04 (LTS)	6.0	无	6.0、5.0、3.1、3.0、2.2、2.1、2.0

表 1 ubuntu .NET 版本支持

1.2.2 windows dotnet 安装

如果使用的是 win10 及以上系统需要登录微软官网下载 dotnet 7.0 sdk 安装。下载地址：https://dotnet.microsoft.com/zh-cn/download/dotnet/7.0 如图 1-2 所示。

生成应用 - SDK ⓘ

SDK 7.0.407

OS	安装程序	二进制文件
Linux	包管理器说明	Arm32 | Arm32 Alpine | Arm64 | Arm64 Alpine | x64 | x64 Alpine
macOS	Arm64 | x64	Arm64 | x64
Windows	Arm64 x64 x86 | winget 指令	Arm64 | x64 | x86
全部	dotnet-install scripts	

图 1-2 windows .Net 7.0 SDK 下载

Dotnet 安装后需要配置 dotnet 系统环境变量，如下图所示：

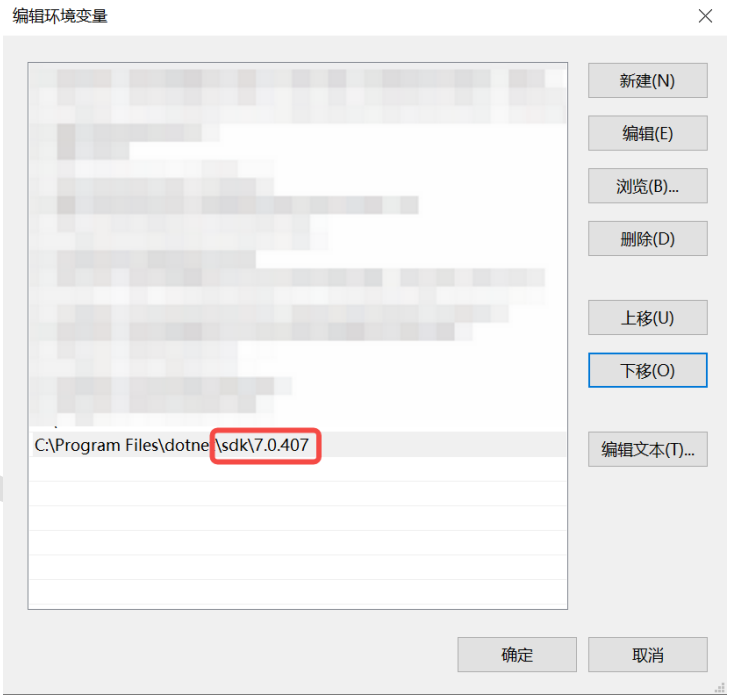

图 1-3 dotnet 环境变量配置

注意：环境变量要配置到 sdk\version 这级目录。如 C:\Program Files\dotnet\sdk\7.0.407

1.3 软件打开

安装好 dotnet 依赖后，如果是 ubuntu 系统，则需要在嘉楠科技官网开发者平台中下载 AI Cube for Linux.zip 压缩包，并在非中文路径下进行解压。

解压完成后，在 ubuntu 命令行终端执行./run.sh,如图 1-4 所示。

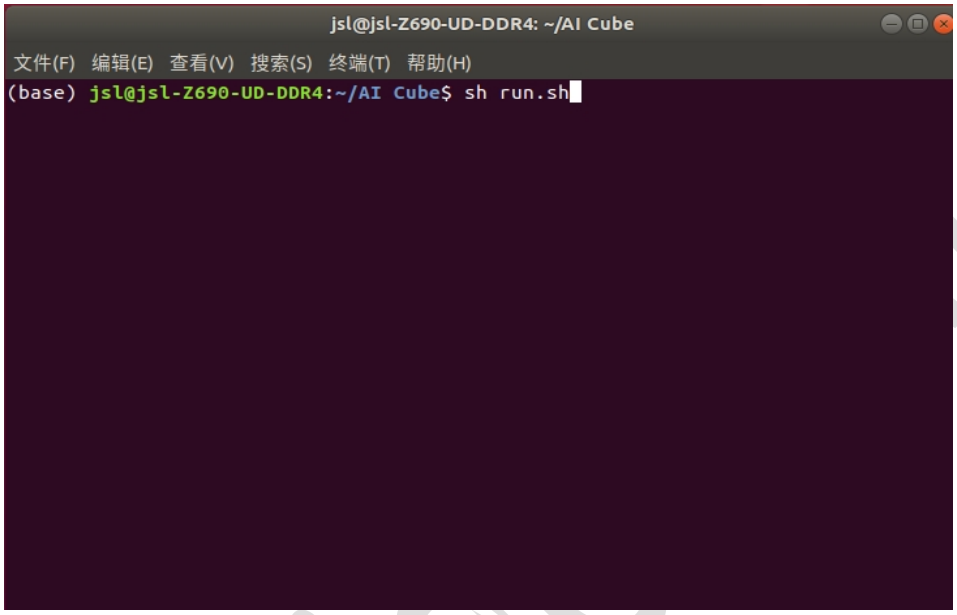

图 1-4 AI Cube 运行命令

执行 sh run.sh 命令后 AI Cube 将会正常打开，软件默认进入项目选项卡。如图 1-5 所示。

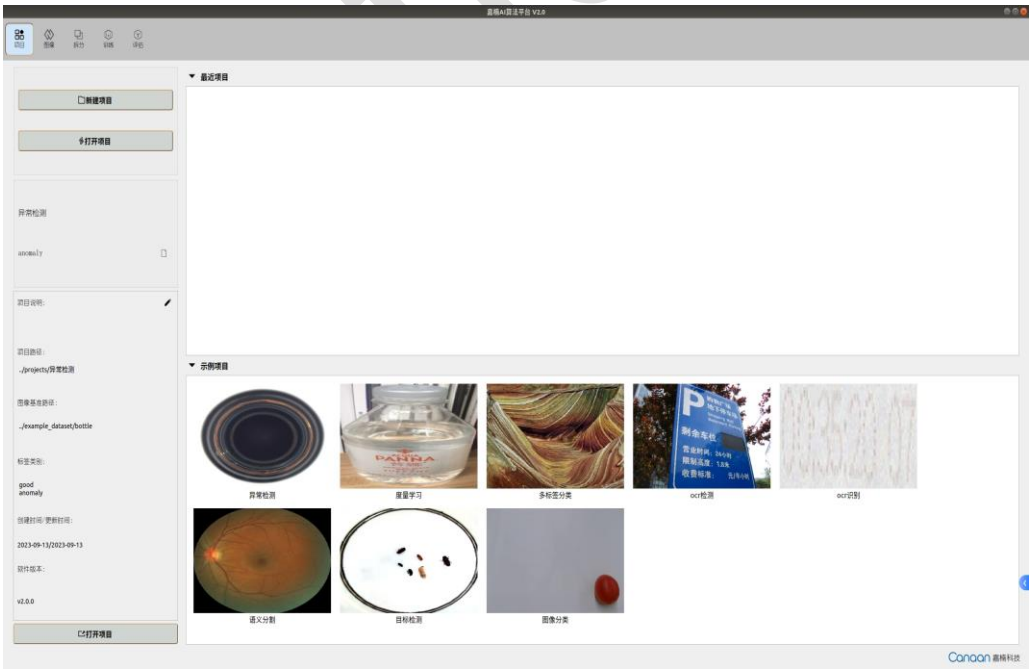

图 1-5 AI Cube 主界面

如果是 windows 系统则需要下载 AI Cube for Windows.zip 压缩包，并在非中文路径下解压。解压后能够看到如下内容，如图 1-6 所示。

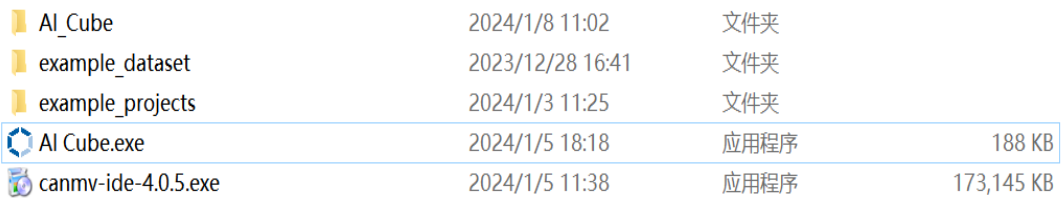

图 1-6 windows 版本 AI Cube

双击运行 AI Cube.exe 即可打开软件。注意：第一次打开软件所需时间较长，请耐心等待。

1.4 软件运行许可

AI Cube 运行时需要软件授权许可，软件运行许可以 AI_Matrix_license_日期.dat 命名，每个授权 license 有效时长为一个月，用户在使用 AI Cube 时，需将授权 license 拷贝至 AI Cube 软件目录，如图 1-7 所示。

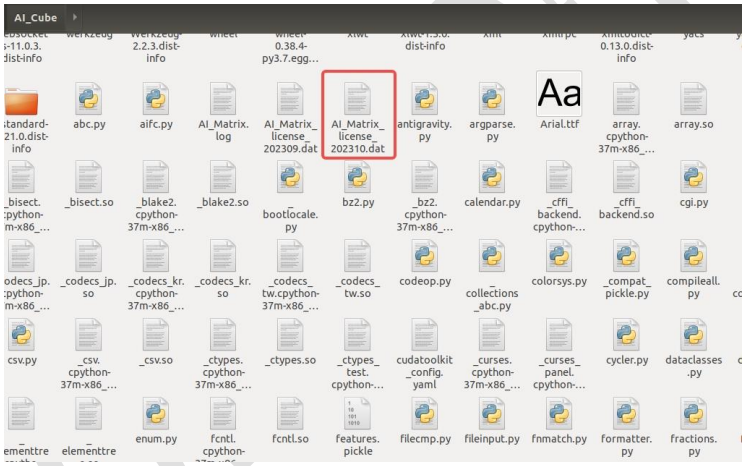

图 1-7 AI Cube 授权 license 所在目录

如果运行 AI Cube 时缺少授权 license 或者当月授权 license 过期，则会在软件启动前会弹出缺少 license 授权提示框，如图 1-8 所示。

图 1-8 运行时许可缺少提示

弹出该提示框后，用户可以向嘉楠科技开发者支持邮箱 Developersupport@canaan-creative.com 发送邮件，申请运行时许可证书。

2 AI Cube 项目选项卡

2.1 AI Cube 示例项目

打开 AI Cube 后，在项目选项卡页面的右侧可以看到最近的项目和示例项目两部分，由于是第一次打开，所以最近的项目一栏为空。AI Cube 为用户提供了 8 个示例项目分别是图像分类、目标检测、语义分割、OCR 检测、OCR 识别、度量学习、多标签分类以及异常检测。

在示例项目一栏中双击任意一张图片可以进入到该项目中，例如双击图像分类项目，就会打开图像分类项目，同时 AI Cube 会自动切换至图像选项卡页面，如图 2-1,2-2 所示。

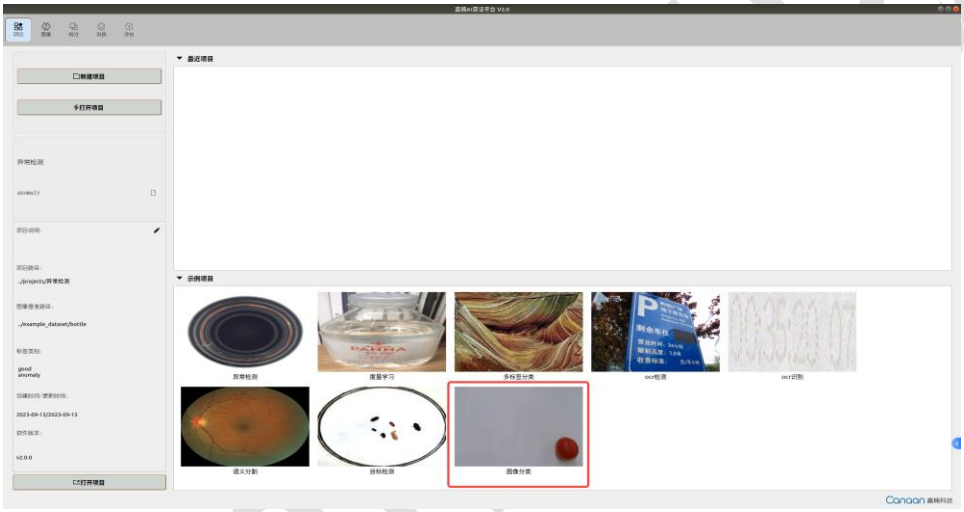

图 2-1 双击图像分类项目

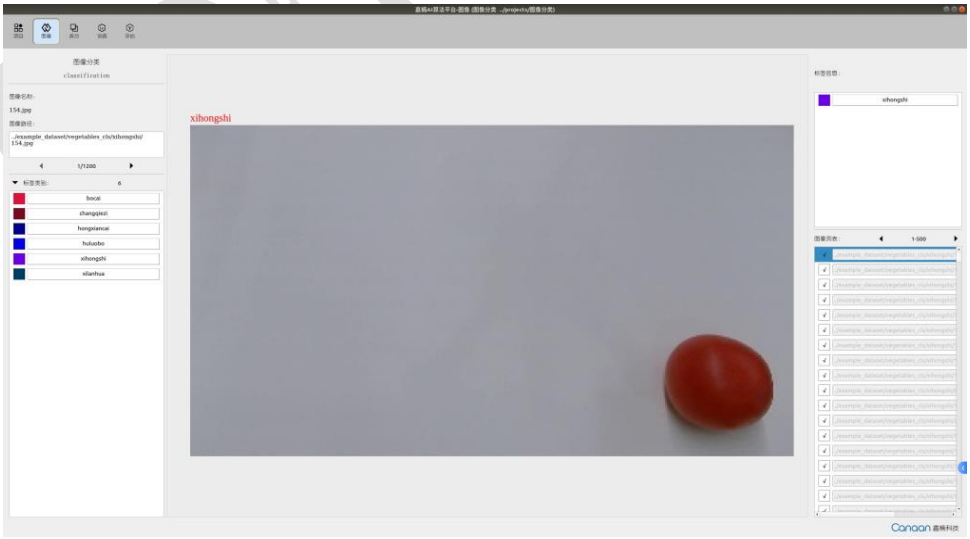

图 2-2 跳转至图像选项卡

用户可以双击任意的示例项目，进入到示例项目中后，图像、拆分、训练、评估选项卡将被激活，用户可以在不同选项卡中了解 AI Cube 各项功能。

2.2 AI Cube 最近项目

关闭软件后再打开软件，可以看到上方最近项目栏中会刷新出用户最近使用的项目，双击该栏中的项目图片可以进入到相应的工程，如图 2-3 所示。

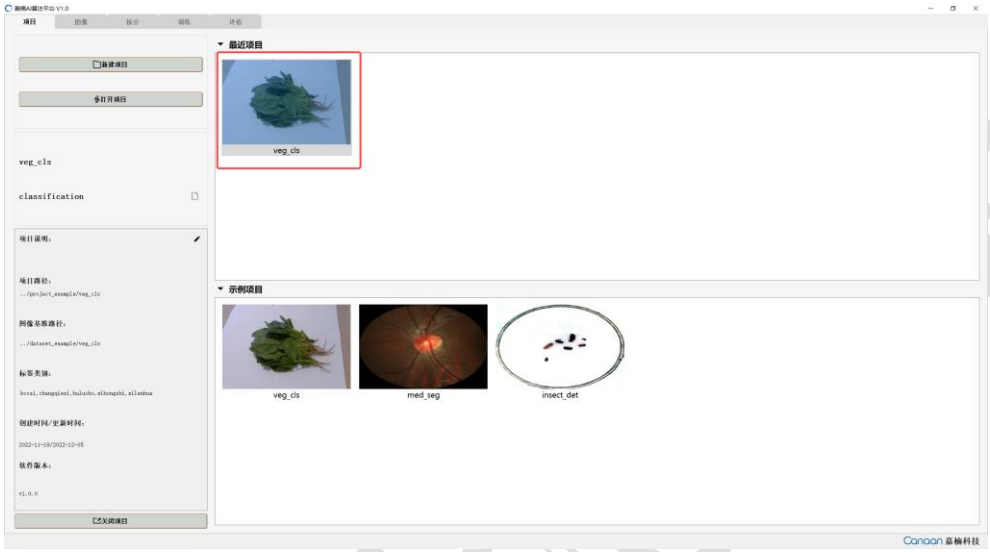

图 2-3 最近的项目

最近的项目中会保存上次用户关闭后所保存的拆分比例、训练参数等内容。方便用户快速打开最近工作项目内容。最近项目栏中最多支持 8 个用户最近使用过的项目。

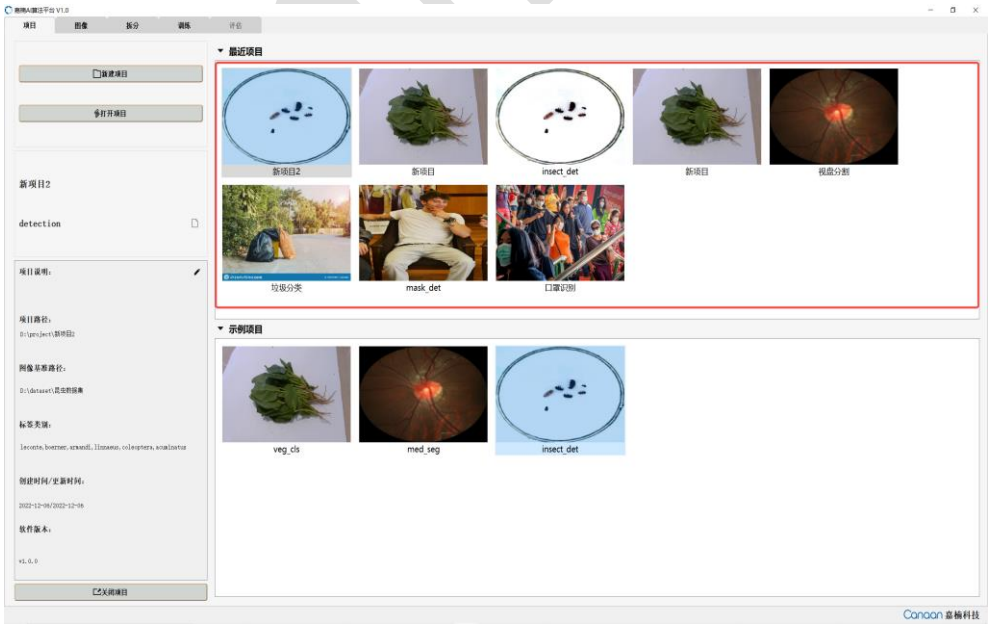

图 2-4 最多支持 8 个最近项目

2.3 创建项目

除了使用最近项目、示例项目入口进入项目工程外，AI Cube 还支持用户自主创建项目，单击新建项目按钮会弹出创建新项目对话框如图 2-5 所示。

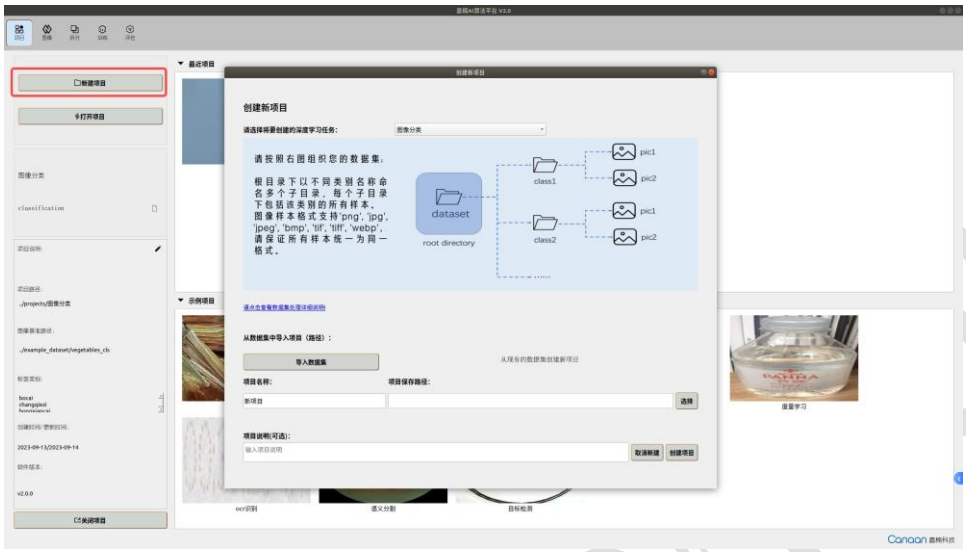

图 2-5 创建新项目对话框

在创建新项目对话框中，AI Cube 支持 8 种不同的任务类型，分别是图像分类、目标检测和语义分割、OCR 检测、OCR 识别、度量学习、多标签分类、异常检测如图 2-6 所示。

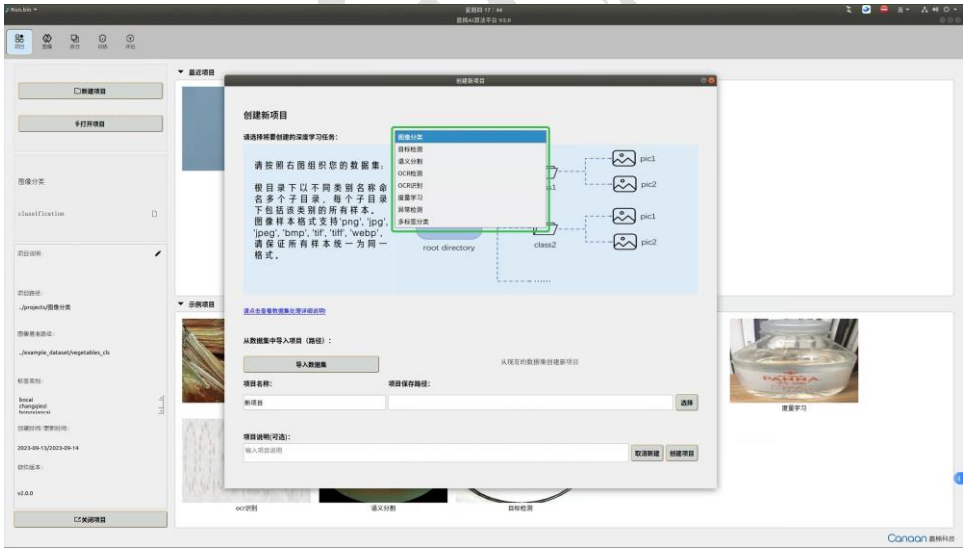

图 2-6 不同任务类型

选定不同的任务类型后，下方的数据集插图会相应的切换，指示用户在不同任务类型下导入不同类型的数据集。

2.3.1 图像分类数据集格式

对于图像分类任务，用户需要将分类数据集按照如下方式进行整理，如图 2-7 所示。

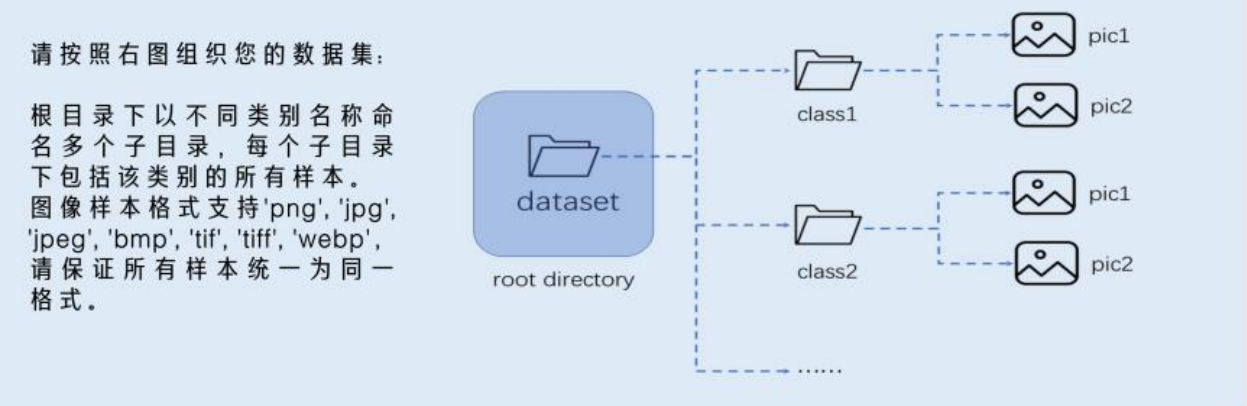

图 2-7 图像分类数据集整理格式

图像分类数据集中的不同类型图片需要放置在以类别命名的文件夹中，如 class1、class2 等，同时在导入数据集时应当选择 dataset root directory，以蔬菜数据集为例。

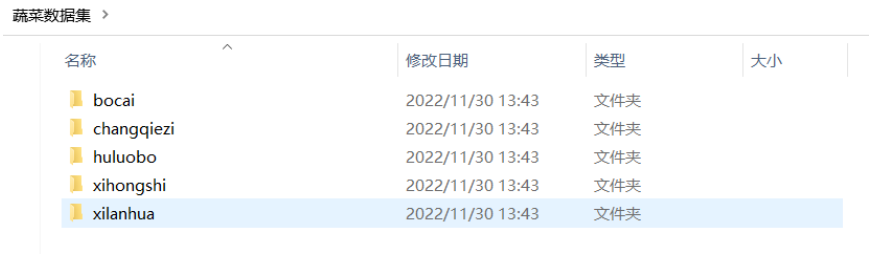

图 2-8 图像分类示例数据集

这里的 dataset root directory 为蔬菜数据集，用户在导入时要选择该级目录。

2.3.2 目标检测数据集格式

对于目标检测任务，用户需要将目标检测数据集按照如下方式进行整理，如图 2-9 所示。

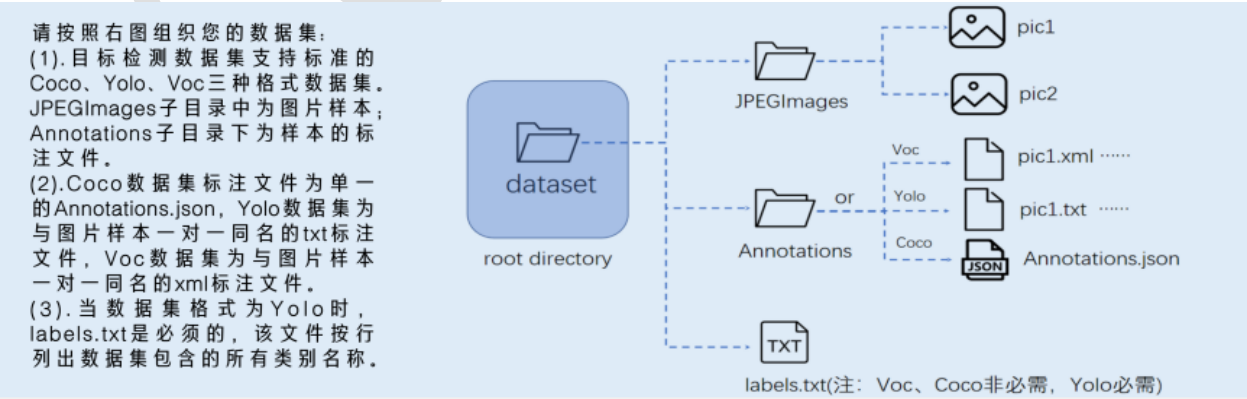

图 2-9 目标检测数据集整理格式

AI Cube 的目标检测任务依赖两个文件夹，分别是 JPEGImages 和 Annotations 文件夹，其中 JPEGImages 中存放的是原始图片，Annotations 中存放的是与之对应的标注文件。AI Cube 目标检测任务支持三种数据集格式，分别是 Annotations json、voc xml 和 yolo txt，用户在使用时将相应格式的标注文件（Annotations.json/pic.xml/pic.txt）放置在 Annotations 文件夹中。

如果是 coco 标注格式，json 文件应当被命名为 Annotations.json;如果是 voc 格式数据集，应当被命名为图像名.xml;如果是 yolo txt，应当被命名文图像名.txt，同时由于 yolo txt 中不含有类别信息，用户需要在 JPEGImages、Annotations 文件夹的同级目录下提供 labels.txt 类别信息文件。

2.3.3 语义分割数据集格式

对于语义分割任务，用户需要将语义分割数据集按照如下方式进行整理，如图 2-10 所示：

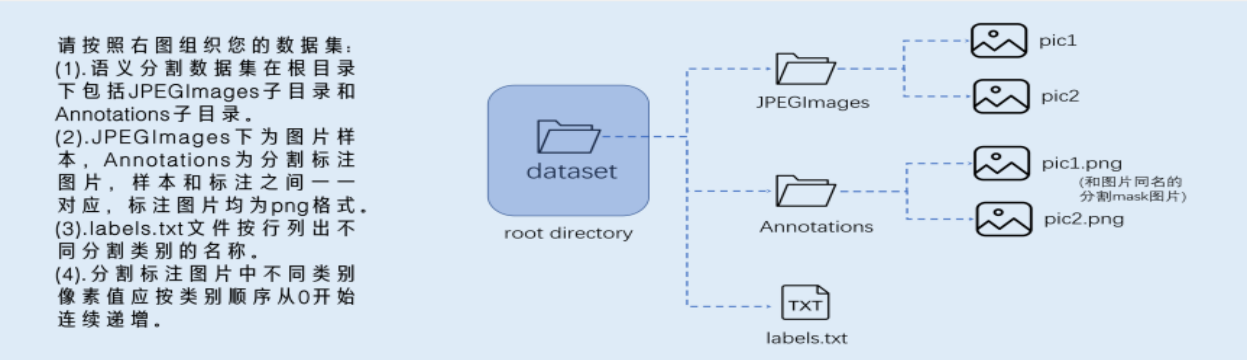

图 2-10 语义分割数据集整理格式

与目标检测任务类似，AI Cube 的语义分割任务依赖两个文件夹，分别是 JPEGImages 和 Annotations，其中 JPEGImages 中存放的原始图片，Annotations 中存放的是标注 mask，由于标注 mask 图片中没有类别信息，因此用户需要提供 labels.txt 作为标签信息文件。labels.txt 文件需要以 background 作为开头（与 mask 中灰度值为 0 的部分对应）。

2.3.4 OCR 检测数据集格式

对于 ocr 检测任务，用户需要将 ocr 检测数据集按照如下方式进行整理，如图 2-11 所示：

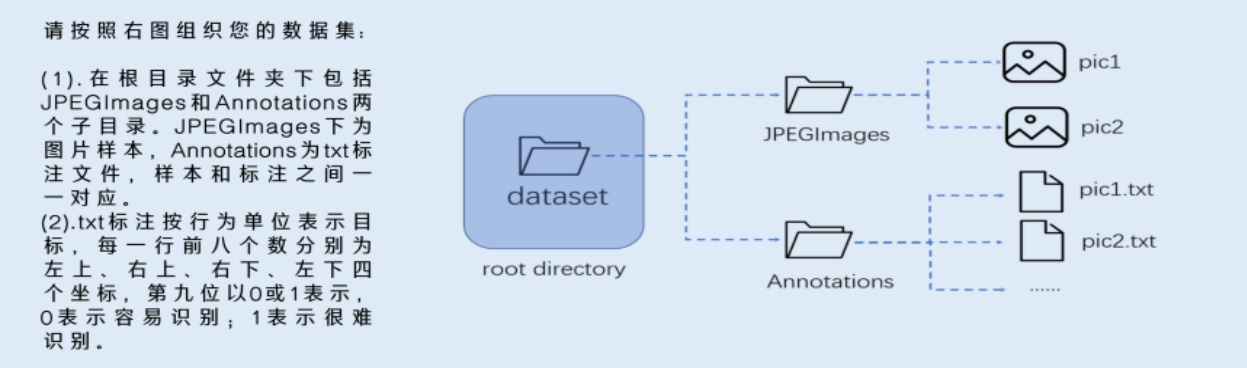

图 2-11 ocr 检测数据集整理格式

与目标检测任务类似，AI Cube 的 OCR 检测任务依赖两个文件夹，分别是 JPEGImages 和 Annotations，其中 JPEGImages 中存放的原始图片，Annotations 中存放的是标注 txt，每一个标注 txt 与图像 pic 之间名称应该一一对应。

标注 txt 文件中以行为单位来存放标注目标的标注信息，每行有 9 个值，分别代表目标的左上、右上、右下、左下四个点坐标；第9为用来描述目标的识别难易程度，其中0代表容易识别，1 代表难识别样本。如图 2-12 所示。

```
730,461,857,413,865,434,738,482,0
753,479,828,453,832,465,757,491,0
729,522,824,490,832,513,737,545,0
712,793,798,778,800,793,714,808,0
683,814,833,789,836,806,686,831,0
647,878,729,865,731,874,649,887,1
724,880,822,865,824,875,726,890,1
```

图 2-12 ocr 检测标注 txt

在该标注 txt 中共有 7 个标注目标，前五个为简单样本，后两个为困难样本。对应的标注图片由 2-13 所示。

图 2-13 OCR 检测标注样样本

从图 2-13 中可以看出按照从上到下的顺序，前 5 个为简单样本，后两个为困难细小样本。

2.3.5 OCR 识别数据集格式

对于 ocr 识别任务，用户需要将 ocr 识别数据集按照如下方式整理。如图 2-14 所示。

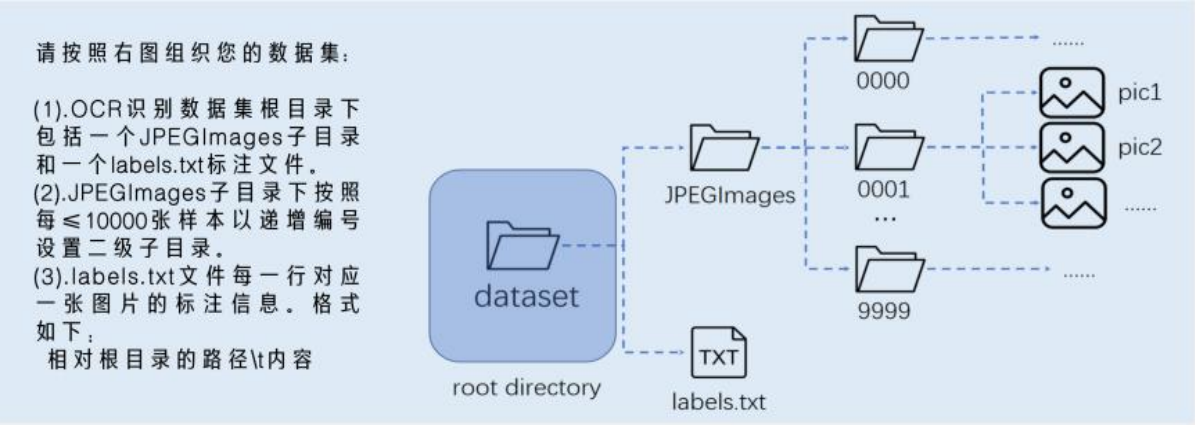

图 2-14 ocr 识别数据集整理格式

OCR 识别数据集中包含 JPEGImages 文件夹和 labels.txt 标注文件，JPEGImages 目录中包含多个子文件夹，每个子文件夹按照小于等于 10000 张样本数量来递增编号，从而形成子目录。

标注文件 labels.txt 文件的每一行对应一张图片的标注信息。如图 2-15 所示。

```
JPEGImages\mining_images\00000.jpg    2020.04.10
JPEGImages\mining_images\00001.jpg    2020.04.10
JPEGImages\mining_images\00002.jpg    2023.04.10
JPEGImages\mining_images\00003.jpg    2023.04.10
JPEGImages\mining_images\00004.jpg    2021.04.10
JPEGImages\mining_images\00005.jpg    2020.04.10
JPEGImages\mining_images\00006.jpg    2023.04.10
JPEGImages\mining_images\00007.jpg    2023.04.10
JPEGImages\mining_images\00008.jpg    2020.04.10
JPEGImages\mining_images\00009.jpg    2020.04.10
```

图 2-15 ocr 识别数据集标注文件

其中每行第一条信息为图像所在的相对路径，以 JPEGImages 文件夹起始；第二条信息为图像对应文本标注信息。

2.3.6 度量学习数据集格式

对于度量学习任务，用户需要将数据集格式按照如下方式进行整理。如图 2-16 所示。

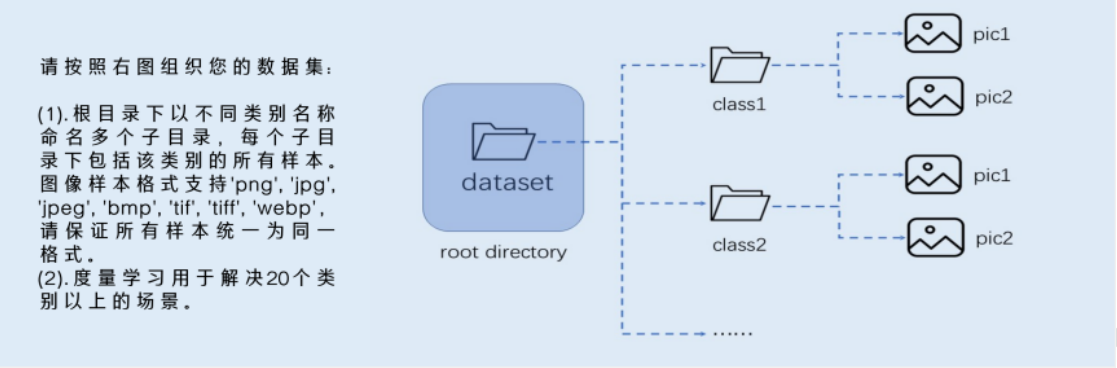

图 2-16 度量学习数据集格式整理

度量学习任务的数据集格式和图像分类任务的格式相同，需要注意的是度量学习任务数据集数量需要在 20 类以上。

2.3.7 异常检测数据集格式

对于异常检测数据集，用户需要将数据集格式按照如下方式进行整理。如图 2-17 所示。

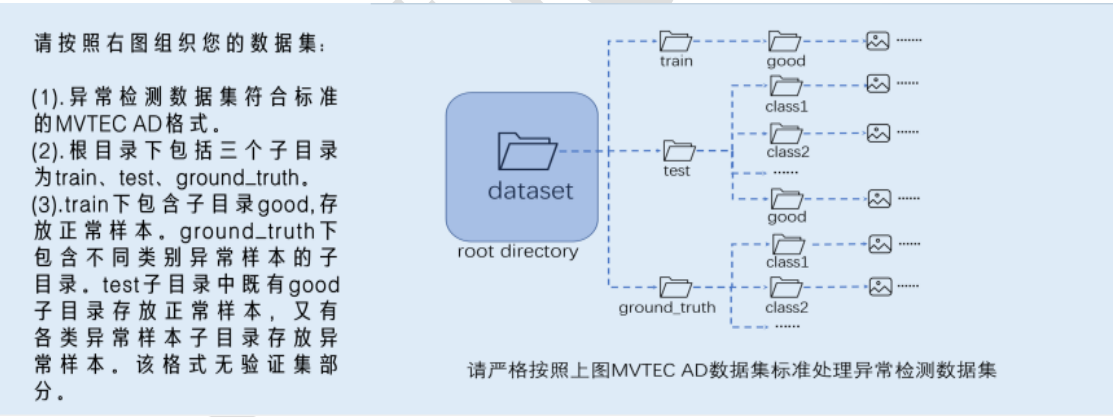

图 2-17 异常检测数据集格式

提供的异常检测数据集需要满足 MVTec AD 数据集格式，其中包括 train、test、ground_truth 目录。train 目录中包含 good 子目录，在该子目录中只包含无缺陷正样本图片；test 目录中包含多个测试类别目录，如 good、broken_large、broken_small、contamination 目录。ground_truth 目录中包含异常品类的二值标注 mask 数据。具体标注格式要求可参考 MVTEC AD。

2.3.8 多标签分类

对于多标签分类数据集，用户需要将数据集格式按照如下方式进行整理。如图 2-18 所示。

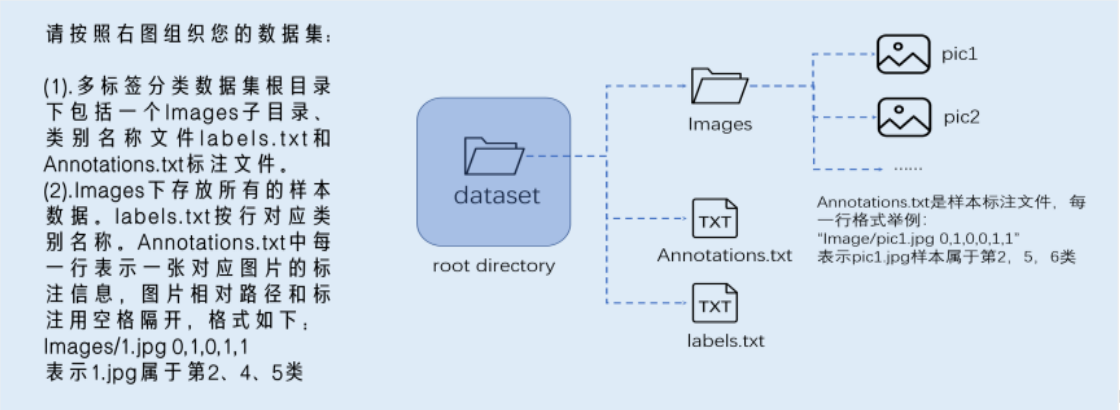

图 2-18 多标签分类数据集格式

多标签分类数据集中包含 Images 目录以及 Annotations.txt 和 labels.txt 两个文件，其中 Images 目录中包含多标签分类数据集中的所有图片，labels.txt 按行对应类别名称。Annotations.txt 中每一行表示一张对应图片的标注信息，图片路径和标注用空格隔开，格式形如：Images/1.jpg 0,1,0,1,1,表示 1.jpg 属于第 2,4,5 类.

用户在将自己的数据集按照提示整理完毕后，选择数据集根目录 dataset root directory 导入数据集，导入后选择项目保存路径，并对项目进行命名。如图 2-19 所示。

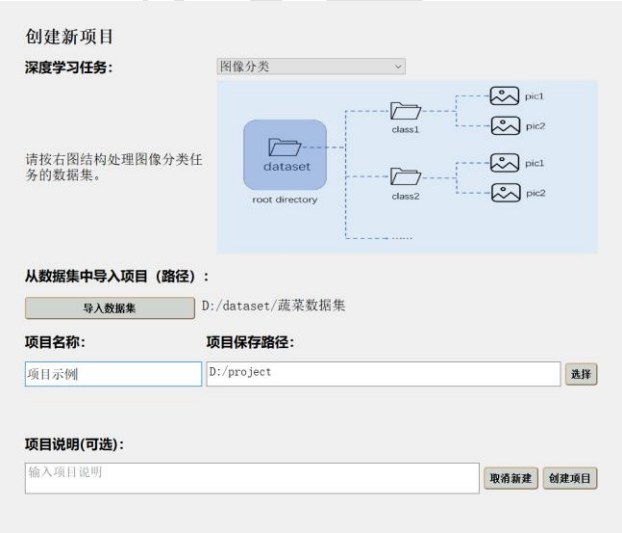

图 2-19 创建新项目

如图 2-19 选择好任务类型，导入数据集并选择工程路径后即可单击创建新项目按钮从而完成新项目的创建。**AI Cube 一个项目绑定一份数据集，为一对一关系。**

2.4 打开本地项目

用户还可以通过打开项目打开本地存在的项目，如图 2-20 所示。

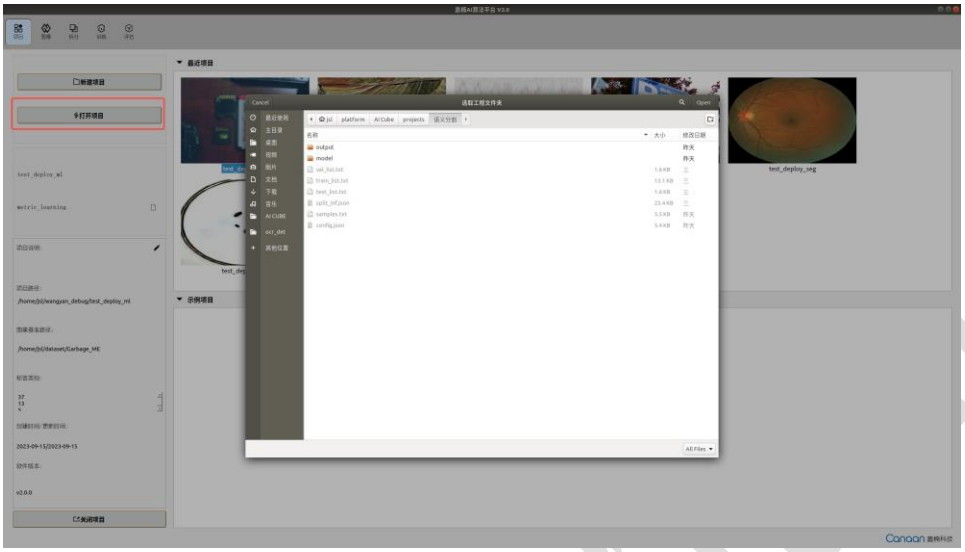

图 2-20 打开本地项目

弹出路径选择对话框后选择工程文件夹，即可打开本地工程。

3 AI Cube 图像选项卡

3.1 图像选项卡概览

AI Cube 图像选项卡页面共由项目图像解析栏、数据集标签解析栏、图像标注解析画布、标签信息解析栏、图像路径列表 5 部分组成，如图 3-1 所示。

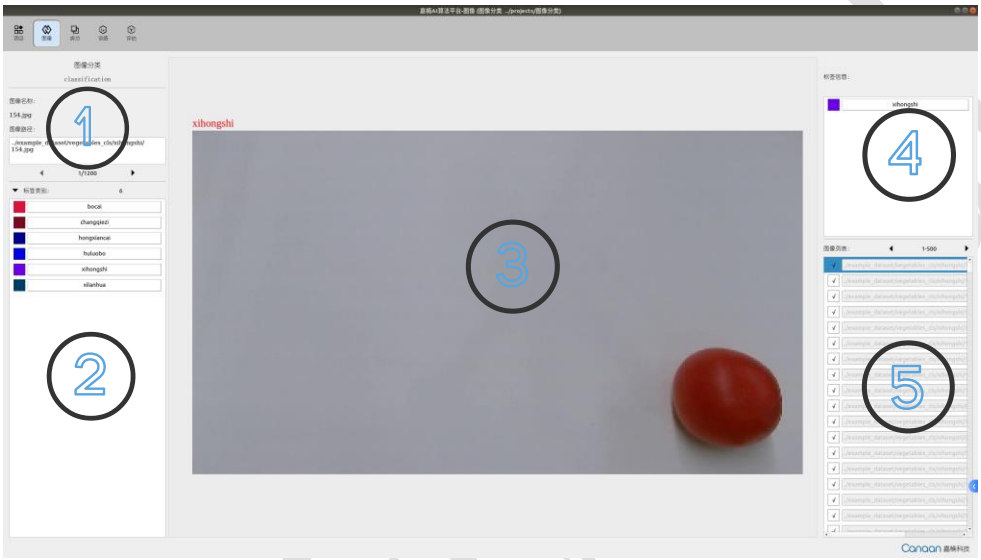

图 3-1 AI Cube 图像选项卡

1、项目图像解析栏：图 3-1 中的①号位置，该位置主要显示项目名称、项目类型、图像名称以及图像本地路径。

2、数据集标签解析栏：图 3-1 中的②号位置，该位置主要解析显示数据集中包含的所有类别（对于语义分割任务，不解析 background 背景类别）

3、图像标注解析画布：图 3-1 中的③号位置，该位置主要显示图片以及图片中的标注信息，对于不同的任务类型，画布中会有不同的解析结果，如图 3-2 所示。

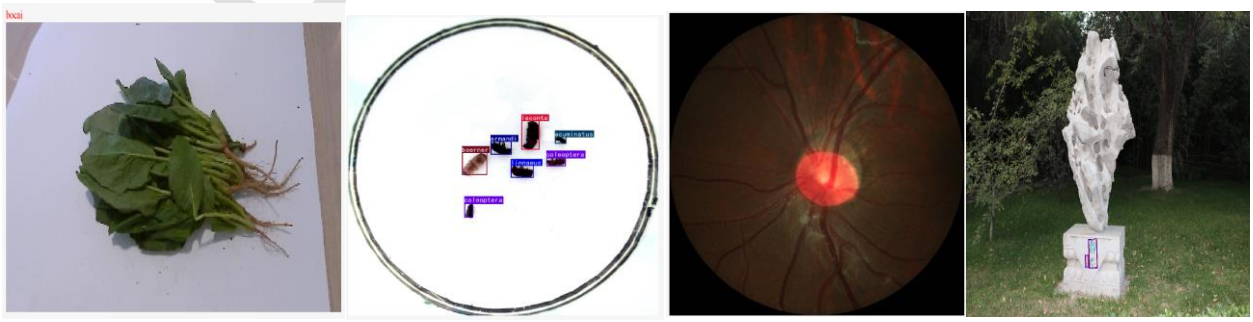

图 3-2 不同任务类型的画布解析

其中图像分类任务会显示图片类别，目标检测任务会显示目标框和目标类别，语义分割会显示区域蒙版 mask。

4、标签信息解析栏：图 3-1 中的④号位置，该位置主要解析图片中的标注信息，对于不同任务类型，该标签信息栏会有不同解析结果，如图 3-3 所示。

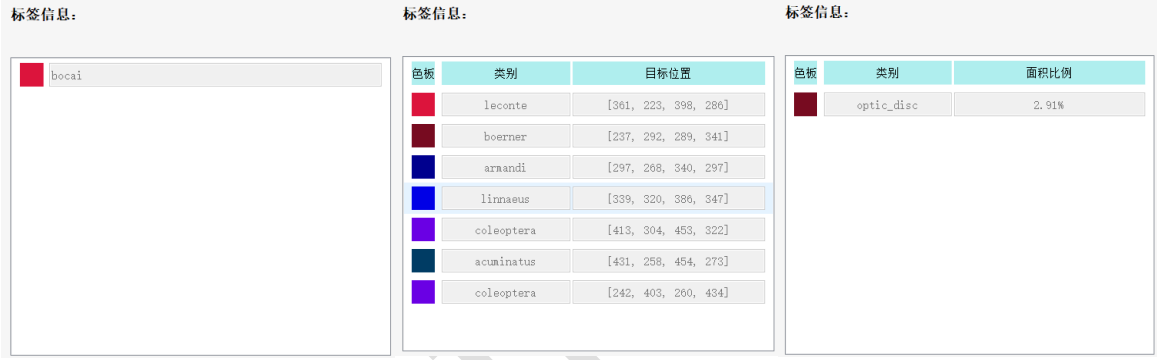

图 3-3 不同任务标注信息解析

如图 3-3 所示，图像分类任务会解析单张图片的分类类别；目标检测任务会解析图片中包围框的类别与位置坐标；语义分割任务会解析出区域类别与区域所占的面积比例；OCR 检测任务会解析出目标位置（八个点）；OCR 识别任务会解析出图片中包含的 OCR 内容（如生产日期）；度量学习会解析出当前图片所属类别（与分类任务相似）；异常检测检测任务会解析出当前图片所属类别（与分类任务相似）；多标签分类任务会解析出当前图片所属类别（与分类任务相似）。

5、图像路径列表：图 3-1 中的⑤号位置，该位置主要解析数据集中的图像路径。

3.2 图像选项卡动作支持

在图像选项卡中，用户可以鼠标单击图 3-1 中①号区域的方向键来浏览数据集中的图片，如图 3-4 所示。

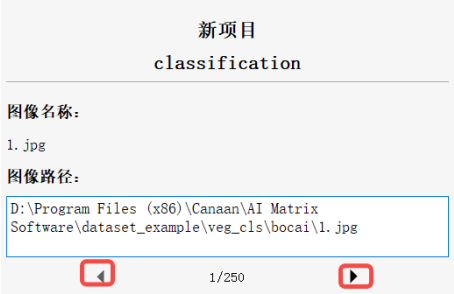

图 3-4 图像切换

或者使用键盘方向键<-,->来进行切换，同时用户还可以通过点击图像路径列表中的内容实现图片的跳跃浏览。

将鼠标放置在画布控件中，向上滑动滚轮，图片将会被放大，可以查看图片细节。向下滑动滚轮，图片将会缩小，可以浏览图片全貌。

4 AI Cube 拆分选项卡

4.1 拆分选项卡概览

AI Cube 的拆分选项卡中用户可以对输入数据集做训练集、验证集、测试集，三种不同集合的拆分，默认拆分比例为 80%、10%、10%。拆分选项卡页面由拆分比例控制栏、类别数量统计直方图、检查控制按钮组及预览相册四部分组成，如图 4-1 所示。

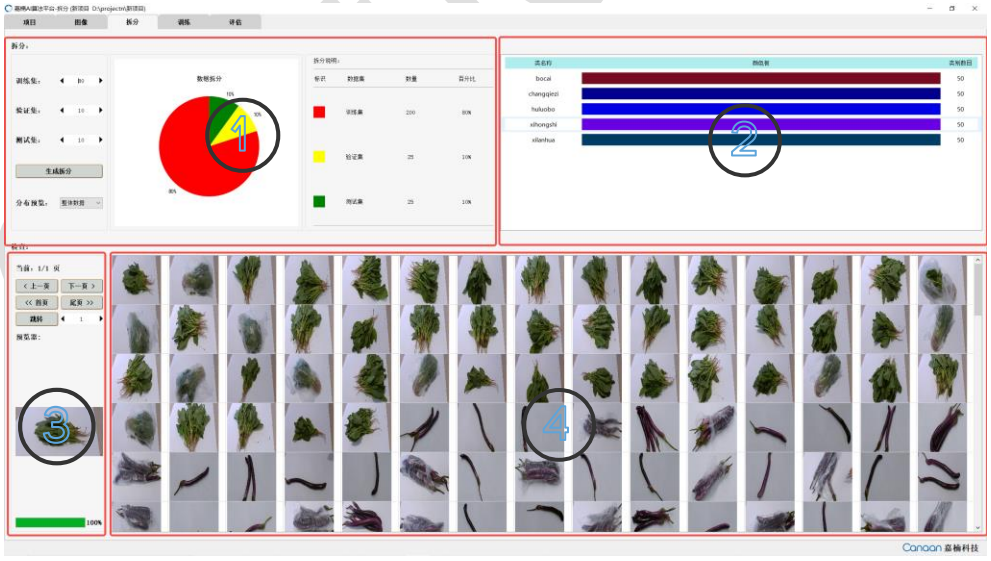

图 4-1 拆分选项卡页面

1、拆分比例控制栏：图 4-1 中的①号区域。用户可以在该栏中自定义拆分比例，其中训练集拆分比例下限位 60%，上限为 80%，设置拆分比例后单击生成拆分按钮，拆分饼图和各集合图像数量统

计会自动刷新。下方的分布预览按钮可以控制各集合的预览，可选项为整体数据集、训练集、验证集、测试集。

2、类别数量统计直方图：图 4-1 中的②号区域。在生成拆分后，可以预览不同分布集合中类别数量，用于查看数据集中各个类别是否分布均衡。

3、检查控制按钮组：图 4-1 中的③号区域。检查控制按钮组可以控制相册区域的翻页和跳转（每页共显示 120 张图片）

4、预览相册：图 4-1 中的④号区域。该区域用来显示指定集合的图像鸟瞰相册。

注意：**由于异常检测数据集格式为 MVTEC AD 数据集格式，因此不支持在该页面进行拆分。**

4.2 预览相册动作支持

在图 4-1 中的①号区域中指定分布预览集合后，相册部分会被动态刷新。相册部分的图像会按照 120 张每页进行显示，用户可以通过点击上一页下一页按钮来进行翻页，同时可以使用首页尾页进行跳转，或者使用跳转按钮，跳转到指定页。

鼠标单击相册部分图片，预览器中的内容会动态刷新，如果双击相册中的图片，则会自动跳转到图像选项卡来浏览该张图片的细节部分。

5 AI Cube 训练选项卡

5.1 训练选项卡概览

AI Cube 训练选项卡共分为模型配置、数据增强、训练参数、训练视图、训练仪表盘和图像评估画布六部分。如图 5-1 所示。其中需要用户配置的部分分别是模型配置、数据增强和训练参数这三部分。

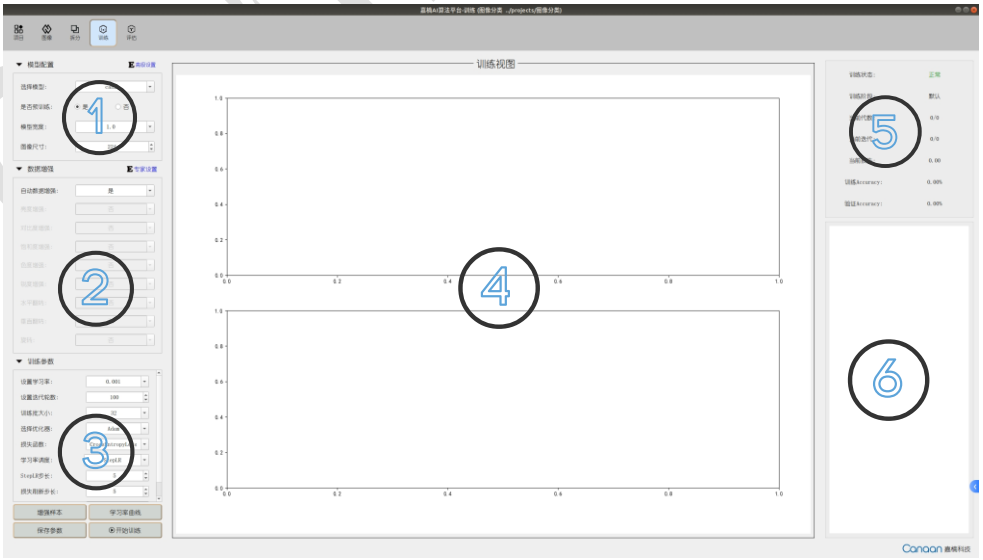

图 5-1 训练选项卡页面

1、模型配置栏：图 5-1 中的①号区域。用户可以在这里自定义的选择模型结构，模型大小，以及模型输入图像尺寸。

2、数据增强配置栏：图 5-1 中的②号区域。用户可以在这里根据所用数据集的特性配置数据增强方法。

3、训练参数配置栏：图 5-1 中的③号区域。用户可以在这里配置学习率、迭代轮数、训练批大小等参数。

4、训练视图：图 5-1 中的④号区域。在 AI Cube 进入训练状态后，AI Cube 计算出的指标数据会以曲线的形式刷新到该区域。

5、训练仪表盘：图 5-1 中的⑤号区域。在 AI Cube 进入到训练状态后，AI Cube 会动态的将训练状态、当前 Epoch 次数、迭代轮数等。在该区域用户可以自定义 loss 曲线的刷新步长。

6、图像评估画布：图 5-1 中的⑥号区域。在 AI Cube 进入到训练状态后，AI Cube 会动态的输出模型在验证集上的评估结果。

如果用户对深度学习模型参数不熟悉，可以直接使用默认参数进行训练，单击开始训练按钮，AI Cube 会进入到训练状态，开始训练按钮变为停止训练按钮，用户可通过操作该按钮来控制 AI Cube 的训练状态。如图 5-2 所示。

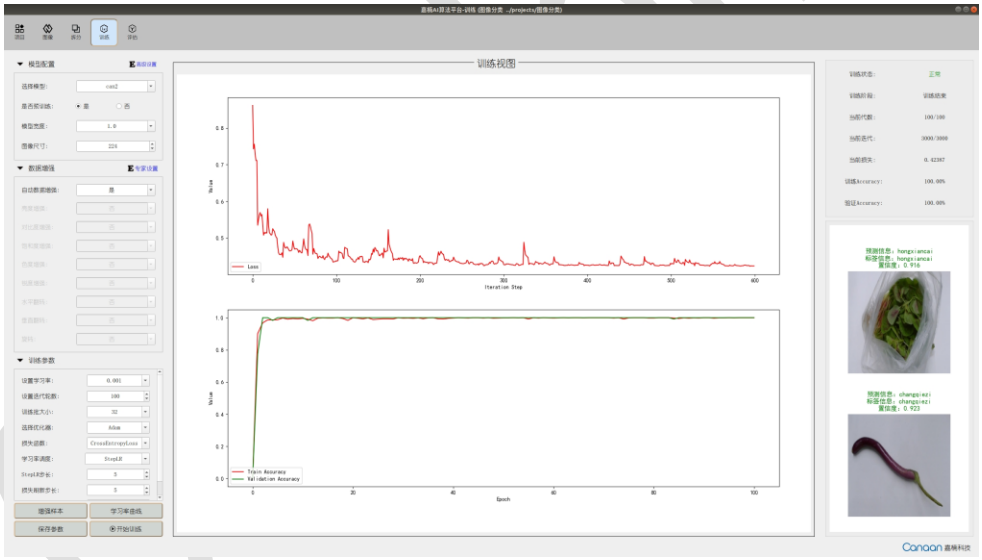

图 5-2 AI Cube 训练过程

在训练过程中，训练视图会实时刷新训练指标曲线，同时仪表盘与图像评估画布也会实时同步更新。

5.2 模型配置

AI Cube 将深度学习的训练模块抽象为模型配置、数据增强、训练参数三部分。在模型配置部分用户可以根据不同的任务类型、不同的数据集来进行配置。

1、对于分类模型用户可以配置模型结构、是否使用预训练、模型宽度和图像尺寸。模型结构提供了 can1、can2、can3、can5、can6、can7、can8、can9、can13、can14，共 10 种模型，用户可以选择任意的模型进行训练，如果在训练结束后想要断点续训，则选择 update 模型。如图 5-3 所示。

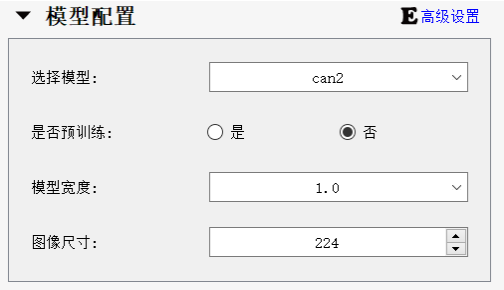

图 5-3 分类模型参数配置

2、对于目标检测模型，用户在配置模型的前提下还可以配置 backbone，这里的 backbone 部分复用了分类模型，目标检测任务 AI Cube 按照是否使用 anchor 提供了 AnchorBased 检测器和 AnchorFree 检测器。根据训练过程使用的模型辅助训练策略提供了 GFLDet 检测模型，同时 AI Cube 还提供了极简目标检测器 FreeDet。如图 5-4 所示。

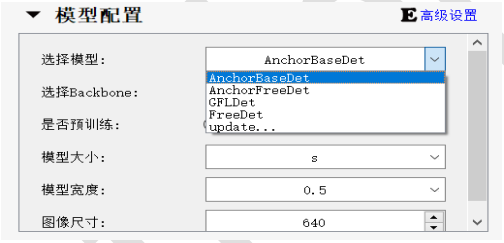

图 5-4 目标检测模型参数

目标检测模型还支持不同模型大小和模型宽度因子供用户选择，这个两个选择分别控制着模型的深度和模型宽度。用户可根据自己的需要进行选择。

3、对于语义分割任务，AI Cube 提供了 DeepNet 和 EDNet 两种模型，其中 DeepNet 参数量较大，分割精度较高；EDNet 模型参数量较少，分割速度较快，如图 5-5 所示：

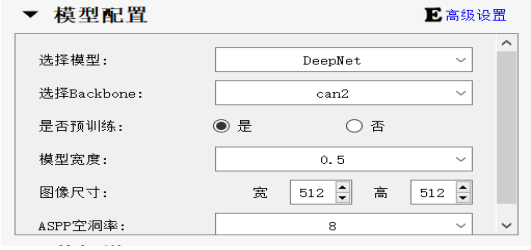

图 5-5 语义分割模型参数

4、对于 OCR 检测任务，AI Cube 提供了 OCR_DNet,该模型能够处理绝大多少场景下的 OCR 定位检测任务。

5、对于文字识别任务，AI Cube 提供了 RCNet 和 RLnet，其中 RCNet 泛用性较广适应多种场景。而 RLNet 适合长度固定的一些文字识别任务，如车牌识别，手机号识别，身份证号识别等。预训练模型中，English 适合训练纯英文识别任务，Chinese 适合训练中英文混合或者纯中文的识别任务，如果选用 Defualt 参数，则会使用 imagenet 预训练权重。使用的定长处理：将图片缩放到固定尺寸进行训练。不定长处理：将图片等比缩放到预设的高度，然后再填充至预设长度（注意如果等比缩放到预设高度，长度超过了预设长度，则会将多余部分去除，应该调整预设长度避免此情况发生）。

6、对于度量学习任务，AI Cube 提供了提供了 can1、can2、can3、can5、can6、can7、can8、can9、can13、can14，共 10 种模型，用户可以选择任意的模型进行训练，同时由于度量学习要对每张输入图片做编码输出，因此用户需要在模型配置栏中配置编码长度。（一般来说编码长度指定为 256 即可）

图 5-6 度量学习编码长度配置

7、对于多标签分类任务，AI Cube 提供的选项与分类任务相同

8、对于异常检测任务，AI Cube 提供了 can1、can2、can3、can5、can13、can14，共 6 中 backbone，backbone 的选择会对异常检测的结果影响较大。同时建议在异常检测任务中使用预训练。

注意：**如果用户对模型配置参数陌生，可以直接使用默认参数。**

5.3 数据增强

AI Cube 提供了以下数据增强方法，如表 5-1 所示

任务类型 / 增强方法	图像分类	目标检测	语义分割	ocr检测	ocr识别	多标签分类	度量学习	异常检测
自动数据增强	√	-	-	-	√	√	√	-
亮度增强	√	√	√	√	√	√	√	-
对比度增强	√	√	√	√	√	√	√	-
饱和度增强	√	√	√	√	√	√	√	-
色度增强	√	√	√	√	√	√	√	-
锐度增强	√	√	√	√	√	√	√	-
水平翻转	√	√	√	-	√	√	√	-
垂直翻转	√	√	√	-	√	√	√	-
旋转	√	-	-	-	√	√	√	-
随机缩放	-	-	√	-	-	-	-	-
随机裁剪	-	-	√	-	-	-	-	-

表 5-1 不同类型任务增强方法

对于分类任务，用户可以直接使用自动数据增强方式来进行数据增强，其他任务需要数据集特点来进行合理数据增强。**注意：数据增强方式并不是开启的越多越好，过度的数据增强有可能会降低检测器的检测性能。**

5.4 训练参数

AI Cube 对不同的任务提供了不同的训练参数，不同任务之间可选择的任务参数不同，

1、分类任务可调参数：学习率、迭代轮数、训练批大小、优化器、损失函数、学习率调度、StepLR 步长、GPU 索引。

2、检测任务可调参数：在分类任务可调参数的基础上增加是否使用 AutoAnchor、NMS 置信度阈值、NMS 交并比阈值、AMP 混合精度、移动平均 EMA、早停策略 ES、是否进行预热、多尺度训练等。如果在目标检测任务中使用 AnchorBased 检测器，可以使用 AutoAnchor 来自动对数据集进行 Anchor 计算，NMS 置信度阈值与交并比阈值会影响输出框的数量，以及重叠框之间的抑制程度。

3、分割任务可调参数：在分类任务可调参数的基础上增加验证批大小、验证缩放设置和验证翻转设置。

4、OCR 检测任务可调参数：置信度阈值、Box 阈值、学习率、迭代轮数、训练批大小、验证批大小、预热代数、损失刷新步长等。

5、OCR 识别任务可调参数：在 OCR 检测任务的基础上加入了学习率衰减节点以及学习率衰减因子。

6、度量学习任务可调参数：度量学习任务在分类任务的基础上加入了取样器、度量损失以及度量损失函数等。

7、多标签分类任务可调参数：多标签分类任务可看做分类任务的增强，参数与分类任务相同。

8、异常检测任务：异常检测任务只支持调整训练批大小及 GPU 序号。

注意：**AI Cube 目前只支持单卡训练，用户可以指定不同的显卡需要进行训练，默认的显卡序号为 0。**

5.5 训练前可视化

AI Cube 支持训练前数据增强可视化和学习率变化过程可视化。单击增强样本按钮，可以预览本次训练过程中样本图像的增强效果。如图 5-6 所示。

图 5-6 数据增强可视化

在数据增强预览页面中左侧为原始图片，右侧为增强效果图，用户可以通过选择图像列表中的图片查看增强效果。

点击学习率曲线按钮，将会弹出本次训练过程中学习率动态变化过程曲线，方便用户在训练前确定合适的学习率。如图 5-7 所示。

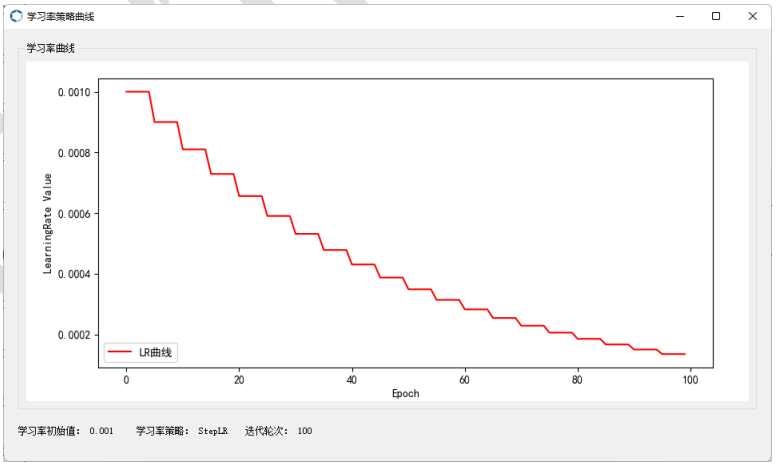

图 5-7 学习动态变化曲线

6 AI Cube 评估选项卡

6.1 评估选项卡概览

AI Cube 评估选型卡共分为测试参数配置栏、芯片部署配置栏、推理结果显示画布、测试数据列表、评估指标输出栏、指标说明栏六部分组成。如图 6-1 所示。

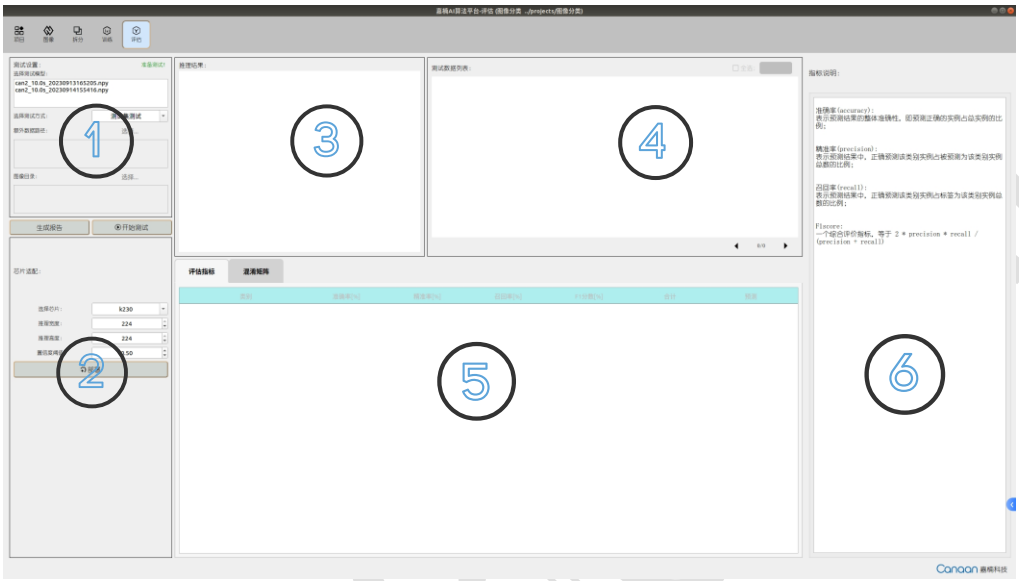

图 6-1 AI Cube 评估选项卡

在评估选项卡中用户可以在①号区域（测试参数配置栏）选择模型进行评估，如果在未选择模型的前提下点击评估测试按钮，则会弹出“请选择模型”提示，如图 6-2 所示。

图 6-2 模型选择提示

AI Cube 支持三种图像评估来源，分别是测试集测试、额外数据集测试和图像目录测试。

1、测试集测试：拆分选项卡中拆分的测试数据集。

2、额外测试集：非本项目使用数据集，但类别名称、类别数量、数据集格式与本工程所使用的数据集相同。切换测试方法到额外数据集测试后，需要配置额外数据集路径，如图 6-3 所示。

图 6-3 额外测试集测试

3、图像目录测试：选择**只含有图像的目录**进行测试。在该测试模式下推理，只会显示图像推理结果，不会输出测试指标。如图 6-4 所示。

图 6-4 图像目录测试

测试结束后，用户可以单击生成报告按钮，将测试的输出结果按照 pdf 与 xls 保存在工程目录下的 out 文件夹下。pdf 中保存项有任务类型、测试模型、测试模型、测试模型路径、测试数据

路径、标签映射、指标表单及混淆矩阵路径；xls 中保存着本次评估输出的混淆矩阵结果，如图 6-5 所示。

嘉楠AI算法平台测试报告

任务类型：图像分类

测试模式：测试集测试

测试模型：can2_10.0s_20230918144744.npy

测试模型路径:../projects/图像分类/model

测试数据路径:../example_dataset/vegetables_cls

标签映射：【1:bocai】,【2:changqiezi】,【3:hongxiancai】,【4:huluobo】,【5:xihongshi】,【6:xilanhua】

类别ID	准确率[%]	精确率[%]	召回率[%]	F1分数[%]	合计	预测
1	100.00	100.00	100.00	100.00	20	20
2	100.00	100.00	100.00	100.00	20	20
3	100.00	100.00	100.00	100.00	20	20
4	100.00	100.00	100.00	100.00	20	20
5	100.00	100.00	100.00	100.00	20	20
6	100.00	100.00	100.00	100.00	20	20
全类别	100.00	100.00	100.00	100.00	120	120

混淆矩阵保存路径:../projects/图像分类/output/can2_10.0s_20230918144744.xls

图 6-5 评估报告内容

6.2 模型评估显示

单击开始测试后，AI Cube 进入到测试状态，模型的推理结果会实时的刷新在推理结果显示画布上，同时被推理的图片路径会实时追加至测试数据列表中。如图 6-6 所示。

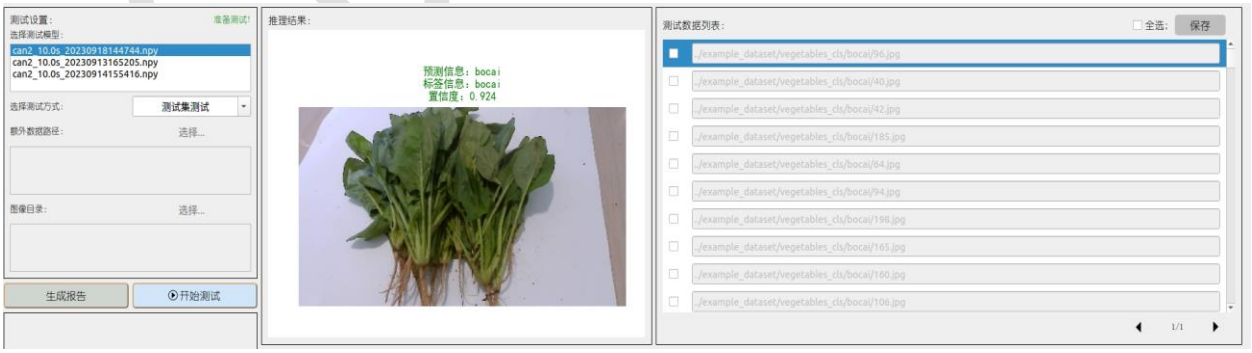

图 6-6 推理结果显示

推理结束后，用户可以单击测试列表中的选项对测试结果进行回溯查看。

在查看时用户可以双击图像条目进入到图片相册模式，在图片相册模式下用户可以放大图片细节进行查看。如图 6-7 所示。

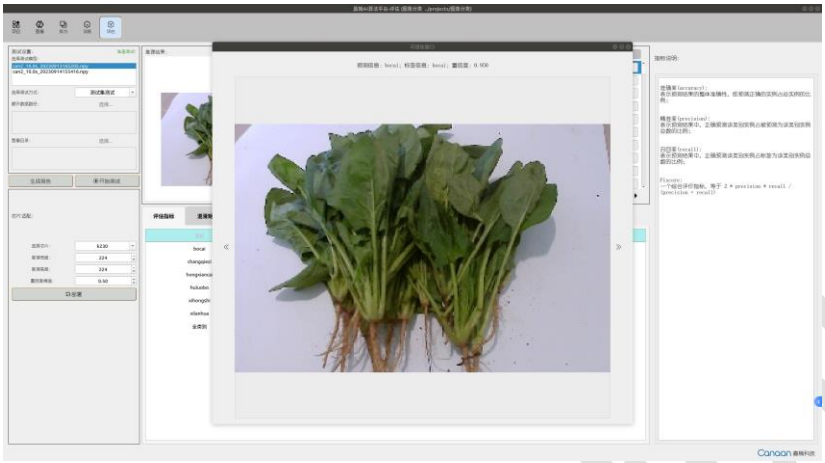

图 6-7 推理结果可视化窗口

如果用户是在测试集测试或额外测试测试模式下进行模型评估，评估结束后还会评估指标输出栏中输出评估指标。如图 6-8 所示。

评估指标 | 混淆矩阵

类别	准确率[%]	精准率[%]	召回率[%]	F1分数[%]	合计	预测
bocai	100.00	100.00	100.00	100.00	20	20
changqiezi	100.00	100.00	100.00	100.00	20	20
hongxiancai	100.00	100.00	100.00	100.00	20	20
huluobo	100.00	100.00	100.00	100.00	20	20
xihongshi	100.00	100.00	100.00	100.00	20	20
xilanhua	100.00	100.00	100.00	100.00	20	20
全类别	100.00	100.00	100.00	100.00	120	120

图 6-8 评估指标

在评估指标输出栏中单击混淆矩阵菜单按钮会显示本次模型在数据集上的评估混淆矩阵，如图 6-9 所示。

	bocai	chan...	hong...	hulu...	xihon...	xilan...
bocai	20	0	0	0	0	0
chan...	0	20	0	0	0	0
hong...	0	0	20	0	0	0
hulu...	0	0	0	20	0	0
xihon...	0	0	0	0	20	0

图 6-9 评估混淆矩阵

6.3 模型部署导出

测试完成后如果用户对模型的检测质量满意，可以在②号区域选择相应的芯片类型、推理宽度、推理高度、置信度阈值等进行导出。

AI Cube 支持训练和部署时不同的宽高设置，例如在训练时用户使用的图像宽高为 320x320，推理时可以选用 224x224。对于不同的任务，导出模型部分会有不同的参数选择，如图 6-10 所示。

图 6-10 模型部署栏

模型部署参数设置完毕后，单击部署按钮，AI Cube 会进入到模型适配状态，同时在 AI Cube 界面中弹出等待窗口。如图 6-11 所示。

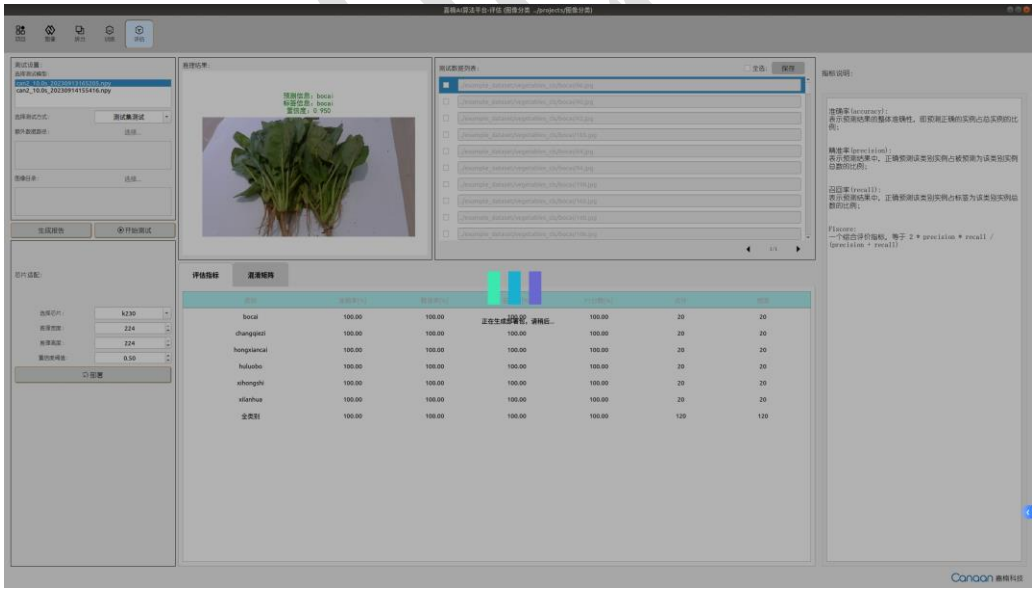

图 6-11 AI Cube 模型部署状态

模型部署结束后会在本地工程目录中生成部署资源包 deploy_source.zip 和 mp_deployment_source.zip 压缩包。其中 deploy_source.zip 包含 evb、canmv 开发板部署资源；mp_deployment_source.zip 包含 micropython 示例部署资源。

7 AI Cube 多版本部署支持

由于 K230 项目迭代速度较快，在开发过程中用户会看到多个 K230 镜像版本，这些镜像版本对应了不同的 nncase 版本和不同语言（C++、Micropython）的部署资源包。其中不同的镜像版本会有不同的部署方式，例如 canmv 版本镜像对应的 K230 可执行程序为*.elf；micropython 版本镜像对应的可执行程序为 xxx.py 文件。为了方便用户能够在 AI Cube 中自由的切换部署环境，AI Cube V1.3 提供了 Patch_Tool 工具，该工具能够支持用户自由刷入不同版本的部署环境。

用户在嘉楠科技官网 Tools->AI Cube 目录下载的 AI Cube V1.3 软件应用程序默认支持的部署镜像为 micropython_v0.5_sdk_v1.4_nncase_2.8.0,如图 7-1 所示。

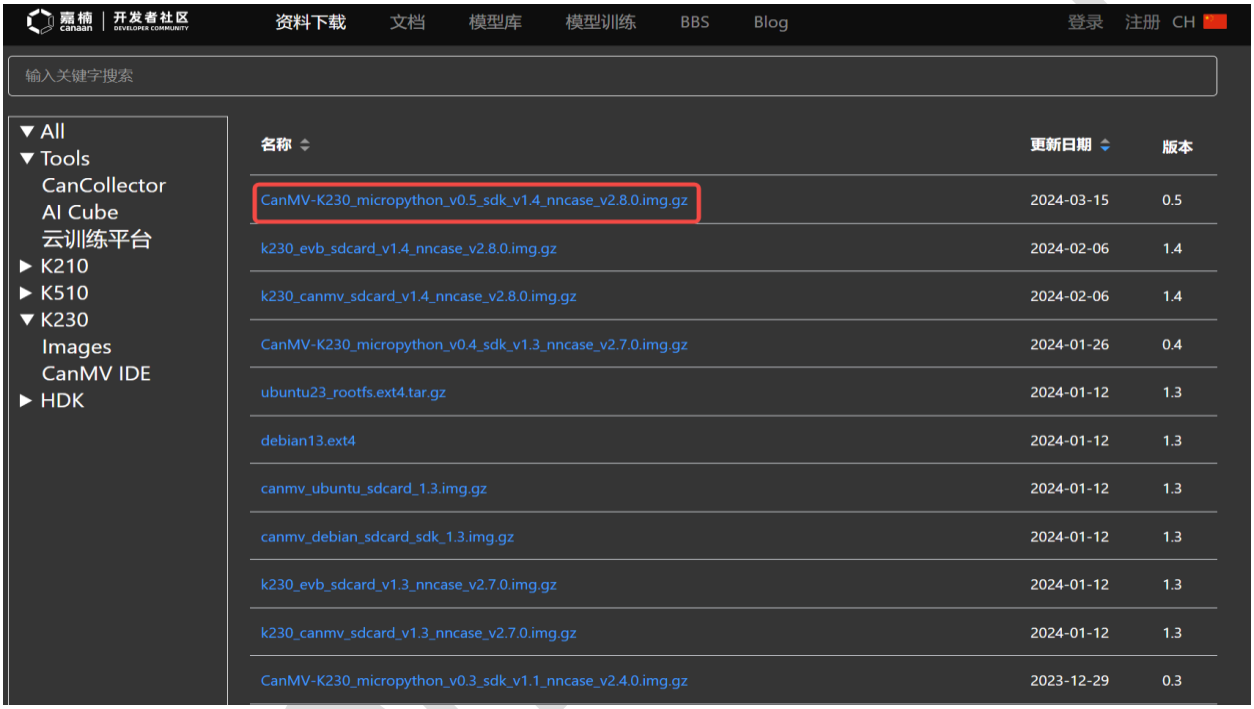

图 7-1 默认部署 sdk 版本支持

AI Cube V1.3 支持使用 Patch_Tool 动态刷入不同的 SDK 版本。目前支持的版本有：

1）k230_canmv_sdcard_v1.3_nncase_v2.7.0
2）k230_canmv_sdcard_v1.4_nncase_v2.8.0
3）CanMV-K230_micropython_v0.4_sdk_v1.3_nncase_v2.7.0
4）CanMV-K230_micropython_v0.5_sdk_v1.4_nncase_v2.8.0

AI Cube 使用 Patch_Tool 刷入不同版本的部署环境后，用户在部署过程可以得到对应的部署资源。

7.1 AI Cube Patch_Tool

AI Cube Patch_Tool 是 AI Cube V1.3 的补丁程序，使用该程序用户可以自由刷入不同的 sdk 部署环境。Windows 用户需要下载 Patch_Tool_for_Windows 压索包解压使用，Linux 用户需要下载 Patch_Tool_for_Linux 压索包解压使用。

软件下载后，对软件进行解压可以得到对应的 Patch 工具包，如下图所示，其中图 7-2 为 Windows 端 Patch 工具包，图 7-3 为 Linux 端 Patch 工具包。

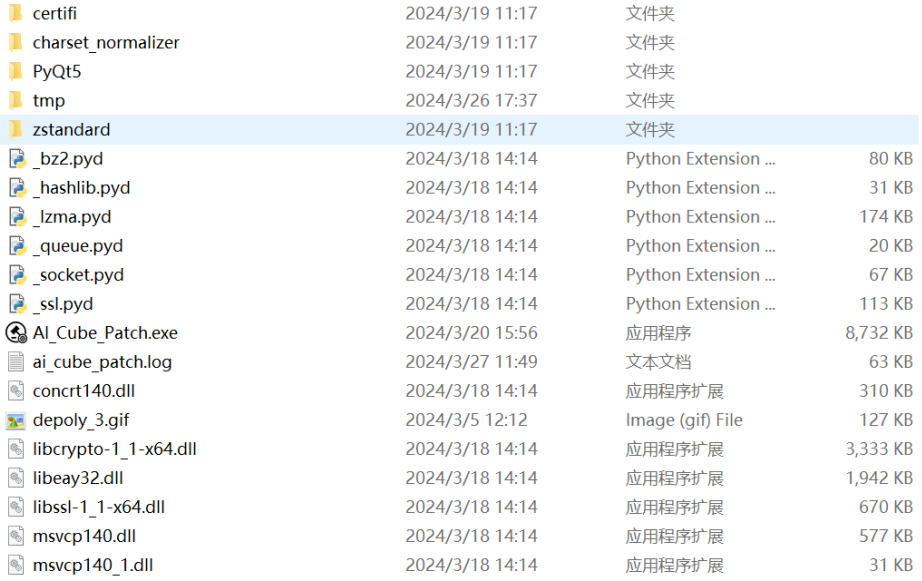

图 7-2 Windows 端 Patch 工具包内容

图 7-3 Linux 端 Patch 工具包

对于 windows 平台，直接运行 AI_Cube_Patch.exe 即可打开 Patch_Tool,如图 7-4 所示。

图 7-4 windows Patch_Tool 软件

对于 Ubuntu 系统则需要在控制台中运行./Run_Patch.sh 命令进行打开，如图 7-5 所示。

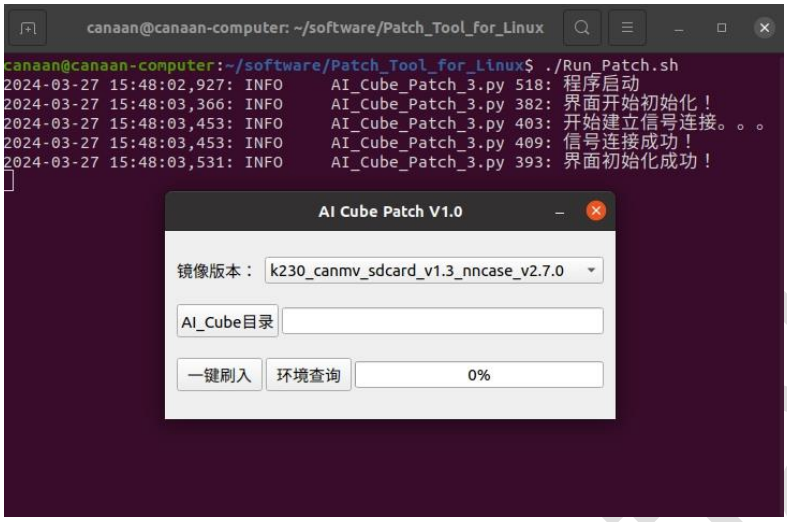

图 7-5 Linux Patch_Tool 软件

用户可以在镜像版本栏中选择需要的镜像以及 AI_Cube 软件目录进行刷入，如图 7-6、7-7 所示。

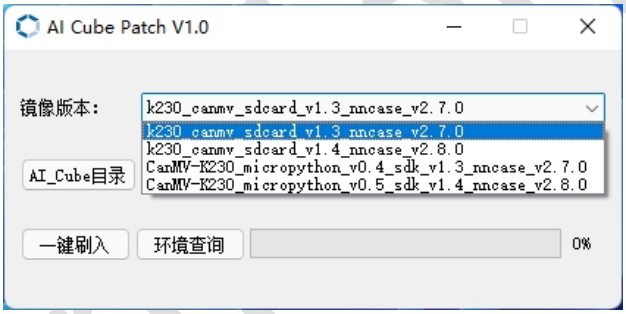

图 7-6 选择镜像版本

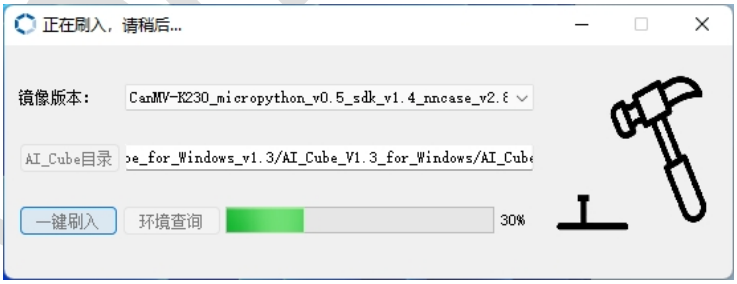

图 7-7 选择 AI Cube 目录

注意：1）AI Cube 目录要选择到 AI_Cube 软件目录，否则会提示刷入错误。

2）刷入时请保持网络通畅

刷入成功后会有提示窗口弹出。

同样的，Patch_Tool 工具还支持对当前的 AI Cube 部署环境进行查询，选择 AI_Cube 软件目录，单击环境查询按钮即可查询当前 AI_Cube 中所包含的部署环境。如图 7-8 所示。

图 7-8 查询 AI_Cube 部署环境

目前 Patch_Tool 支持的镜像 SDK 名称、部署包名称、部署语言 、onboard 内容如表 7-1 所示。

镜像 SDK 名称	部署包名称	部署语言	Onboard 内容
canmv_v1.3_nncase_v2.7.0	deployment_source.zip	C++	镜像压缩包+可执行 elf
canmv_v1.4_nncase_v2.8.0	deployment_source.zip	C++	镜像压缩包+可执行 elf
micropython_v0.4_sdk_v1.3_nncase_v2.7.0	mp_deployment_source.zip	mp	镜像压缩包
micropython_v0.5_sdk_v1.4_nncase_v2.8.0	mp_deployment_source.zip	mp	镜像压缩包

表 7-1 镜像 SDK 关系列表

7.2 部署资源包

使用 AI_Cube 完成部署后，打开本地工程路径，可以在工程路径下可以看到 deploy_source.zip 压缩包（canmv c++部署资源包）或 mp_deployment_source.zip 压缩包（micropython 部署资源包）。

对于 deploy_source.zip，其中包括 canmv 编译源码文件夹、kmodel 文件、deploy_config.json 以及 README.md，如图 7-9 所示。

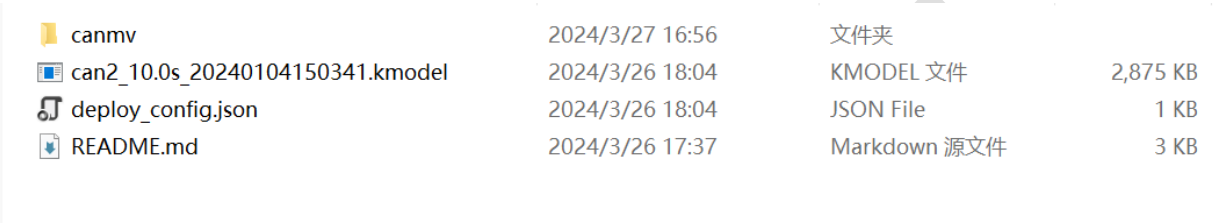

图 7-9 deploy_source 部署资源包内容

其中 canmv 编译源码文件夹中保存了 canmv 芯片端部署编译需要的 code 资源；kmodel 文件为用户本次工程训练指定导出的 kmodel；deploy_config.json 中写有部署推理时的配置参数；README.md 中写有部署包在芯片端部署的使用说明，用户可以按照 README.md 内容在芯片端使用 deployment_source 资源。

对于 mp_deployment_source.zip，其中包括 kmodel 文件、deploy_config.json、README.md、xxx_video.py 和 xxx_image.py。如图 7-10 所示。

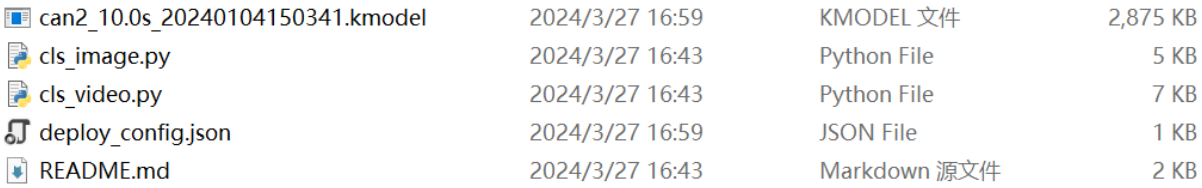

图 7-10 mp_deployment_source 部署资源包内容

其中 deploy_config.json、README.md、kmodel 文件与 7.9 类似。xxx_image.py 为 micropython 图像推理脚本；xxx_video.py 为视频流推理脚本（xxx 为本次部署时的任务类型）。

7.3 工程部署配置文件

图 7-9、7-10 中的 project 目录下保存着以工程名命名的文件夹，该文件夹中包含导出模型及配置参数。在芯片端使用的模型都会以 kmodel 作为文件结尾，打开模型配置参数文件 deploy_config.json，可以看到如下字段，如图 7-11 所示。

```
{
    "nncase_version": "2.8.0",
    "chip_type": "k230",
    "inference_width": 224,
    "inference_height": 224,
    "confidence_threshold": 0.5,
    "export_kmodel_name": "can2_10.0s_20240104150341.npy",
    "model_type": "can2",
    "img_size": [
        224,
        224
    ],
    "mean": [
        0.485,
        0.456,
        0.406
    ],
    "std": [
        0.229,
        0.224,
        0.225
    ],
    "categories": [
        "bocai",
        "changqiezi",
        "hongxiancai",
        "huluobo",
        "xihongshi",
        "xilanhua"
    ],
    "kmodel_path": "can2_10.0s_20240104150341.kmodel",
    "num_classes": 6
}
```


图 7-11 模型参数配置文件

该文件中保存的参数与用户在 AI Cube 中设置的参数一致。配置文件中各 key 值含义见表 7-2。

Key 值	Key 值含义
nncase_version	nncase 版本
chip_type	芯片型号
inference_width	推理宽度
inference_height	推理高度
confidence_threshold	置信度阈值
export_kmodel_name	被导出模型名称
model_type	模型结构
img_size	训练时图像尺寸
mean	训练时均值
std	训练时标准差
categories	数据集类别
samples_txt_path	校正集路径
kmodel_path	导出后模型名称
num_classes	数据集类别数量

表 7-2 deploy_config.json key 值含义

用户在使用 AI Cube 导出模型后，在芯片端使用该文件时可以按照实际需求更改文件内容，一般不推荐用户更改该文件。

7.4 芯片端可执行 demo

模型导出后，用户可以直接在 K230 芯片端推理模型。安装目录下的 on_board 文件夹中存放着 SDK 镜像，如果是 C++部署开发 on_board 中还会有对应的 elf 部署文件。如图 7-12、7-13 所示。

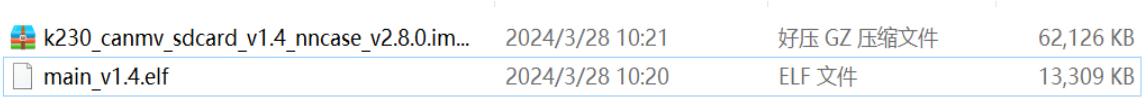

图 7-12 C++部署开发 on_board 文件夹内容

图 7-13 micropython 部署开发 on_board 文件夹内容

其中 elf 为 C++版本部署 code 编译出来的可执行程序，用户可以在芯片端直接使用该程序推理模型。Micropython 代码需要用户自行导出。

images 目录中存放着 canmv、evb、micropython 镜像。对应的 sdk 版本为 1.1，nncase 版本为 2.4。

对 elf 可执行程序，用户在使用该 demo 文件前需要先将 project 目录下 deploy_source.zip 压缩包中的 kmodel 文件、模型参数配置 json 文件以及 on_board 中的 main_xxx.elf 拷贝到 k230 芯片上，并保持同级目录。(板上运行时要保证 elf 版本与 sdk 版本对应)。

1、在使用时，如果是芯片端静态图推理，则在芯片大核控制台运行如下命令：

```
./main_xxx.elf deploy_config.json test.jpg 0
或：./main_xxx.elf deploy_config.json test.jpg 0
```

2、如果使用芯片摄像头推理，则在芯片控制台运行如下命令：

```
./main_xxx.elf deploy_config.json None 0
或：./main_xxx.elf deploy_config.json None 0
```

其中，参数 deploy_config.json 为部署配置文件；test.jpg 代表使用图片推理，None 代表使用摄像头推理；标号 0、1、2 分别表示不调试、简单调试、详细调试。具体使用方法可以参见部署资源包中的 README 文件。

对于 micropython 用户，用户需要本地解压 AI Cube 输出的 mp_deployment_source.zip，然后在 micropython Canmv IDE 下使用该脚本。如图 7-14 所示。

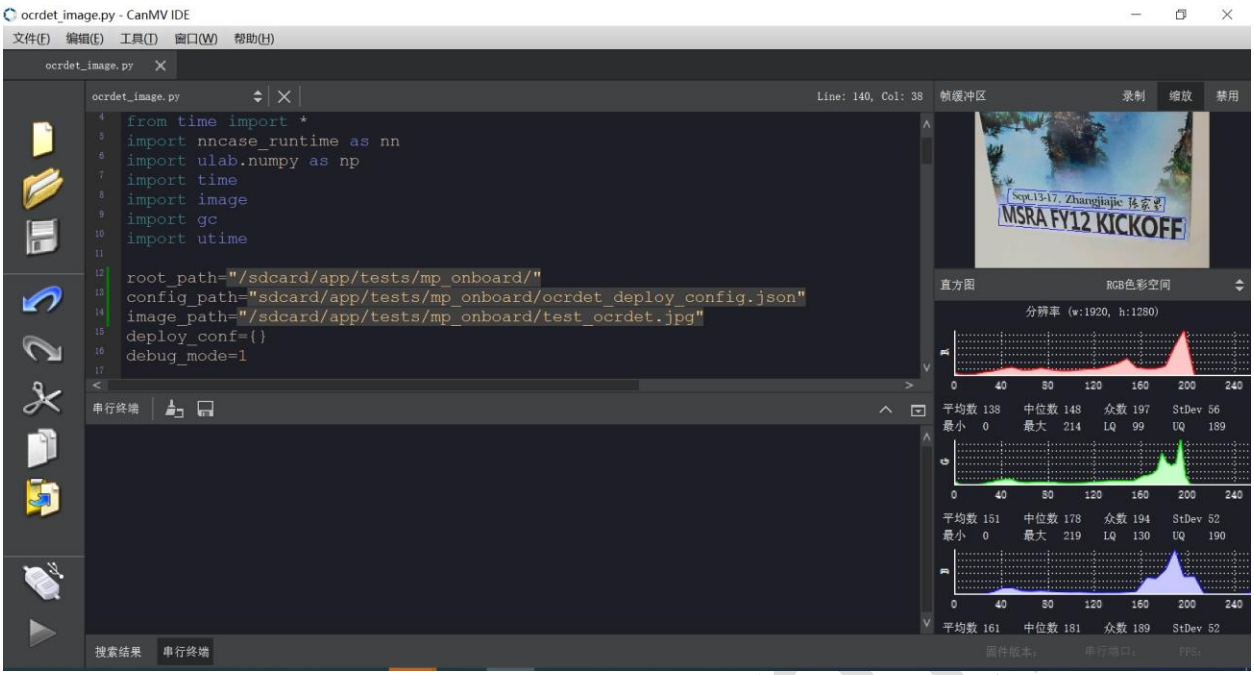

图 7-14 micropython 示例代码使用方法