
Mech-DLK

Mech-Mind

2022 年 12 月 23 日

1	软件更新说明	6
2	安装说明	9
3	开始训练第一个模型	14
4	快速定位	20
5	缺陷分割	27
6	图像分类	44
7	目标检测	63
8	实例分割	81
9	如何进行批量数据集测试	108
10	如何提升标注质量和效率	113
11	如何标记和筛选图像	121
12	如何解决困难问题	123
13	快捷键	129
14	关于 Mech-DLK SDK	131
15	入门指南	133
16	API 参考手册	147
17	软件使用前提	152
18	软件许可获取及更新	165

19 兼容性说明	178
20 术语表	181
21 FAQ	182

Mech-DLK 是梅卡曼德机器人自主研发的机器视觉深度学习软件，内置多种业界领先的深度学习算法，能够解决很多传统机器视觉无法处理的问题，如高难度的分割、定位、分类问题等。

借助直观简易的交互界面，用户无需编程，也无需掌握专业的深度学习知识，即可通过 Mech-DLK 快速实现模型训练与验证。



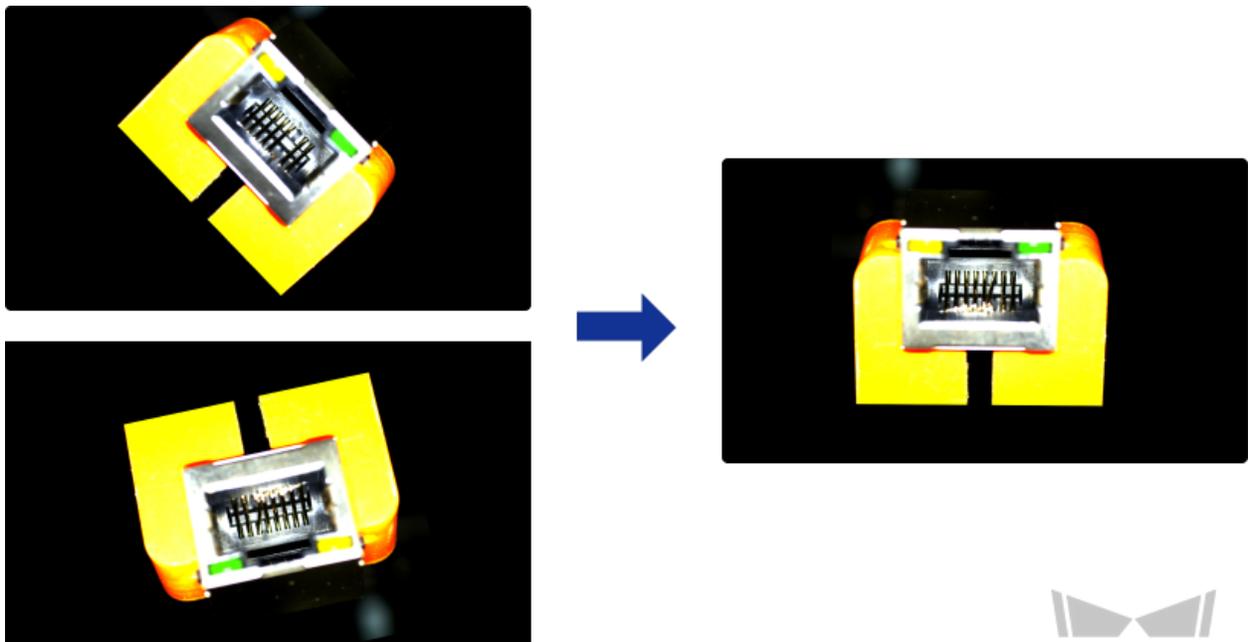
软件内包含“快速定位”、“缺陷分割”、“图像分类”、“目标检测”、“实例分割”五个算法模块。

快速定位

定位角度，根据定位结果转正图像。

用于修正物体姿态，速度快，通常作为其它算法模块的前置模块。

- 定位零部件角度并修正为统一的姿态。

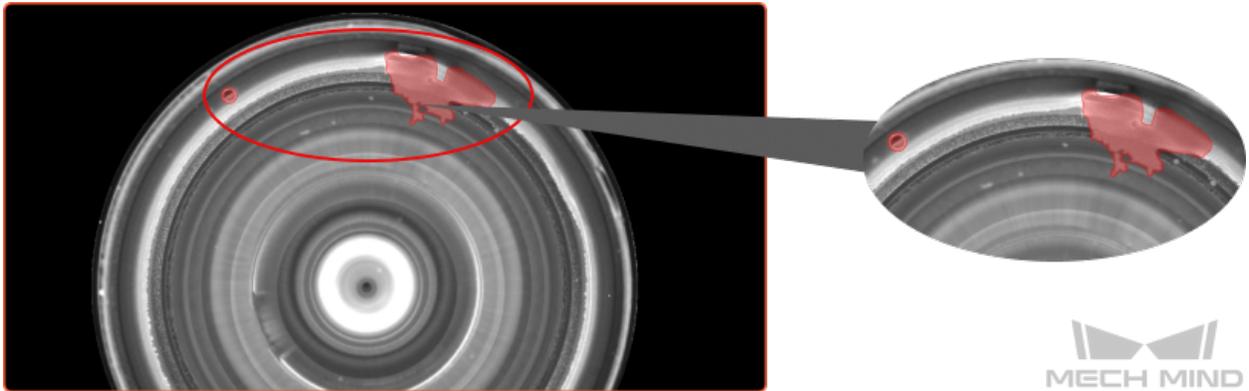


缺陷分割

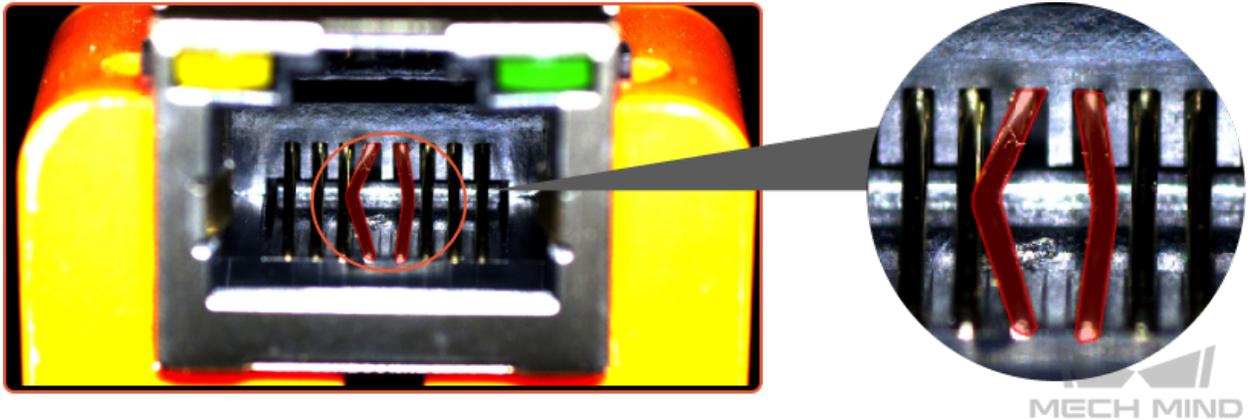
判断图像是 OK 还是 NG。若为 NG，则分割出缺陷区域。

用于检出各种类型的缺陷。如污渍、气泡、划痕等表面缺陷，折弯、异形、缺漏等位置缺陷。在缺陷微小、背景复杂、工件位置不固定等复杂情况下依然适用。

- 检出镜头表面的气泡和溢胶缺陷。



- 检出零部件的折弯缺陷。

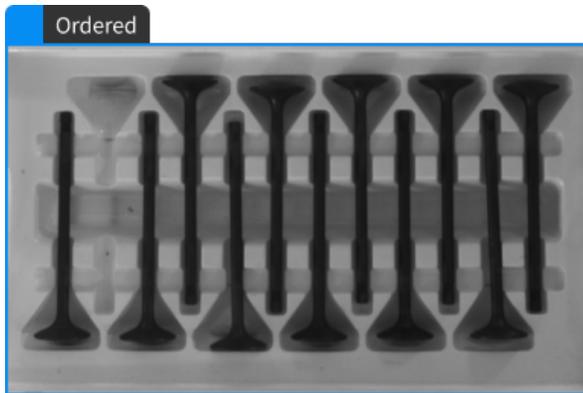


图像分类

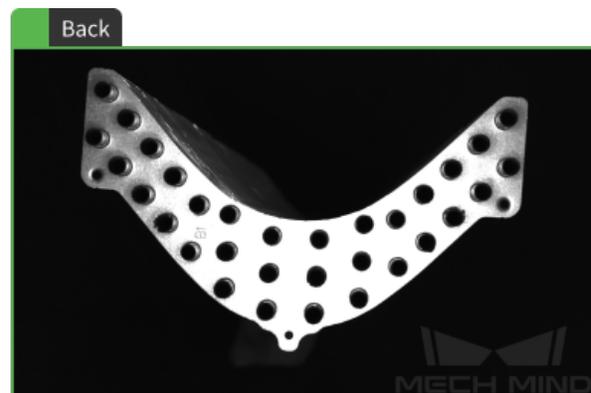
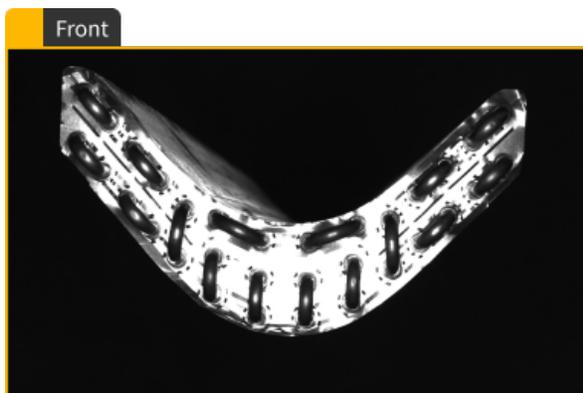
判断图像类别。

用于区分工件正反面、朝向，区分物体缺陷种类，判断物体是否缺失、排列是否整齐。

- 区分工件处于整齐摆放状态还是散乱状态。



- 区分工件正反面。

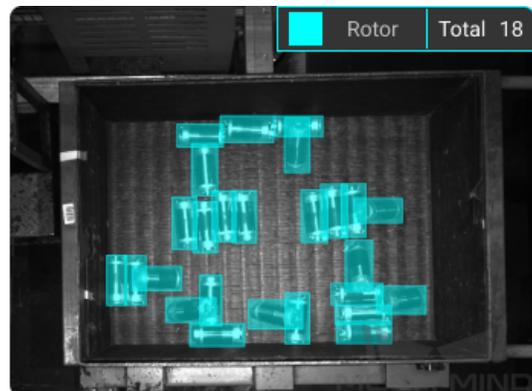
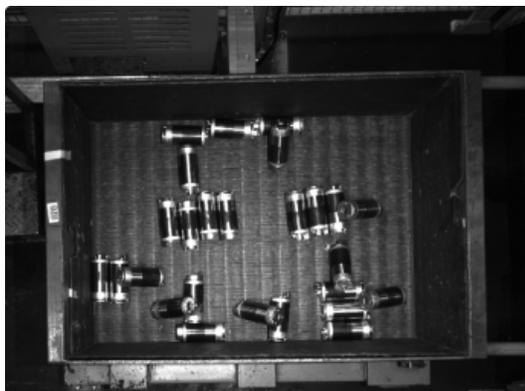


目标检测

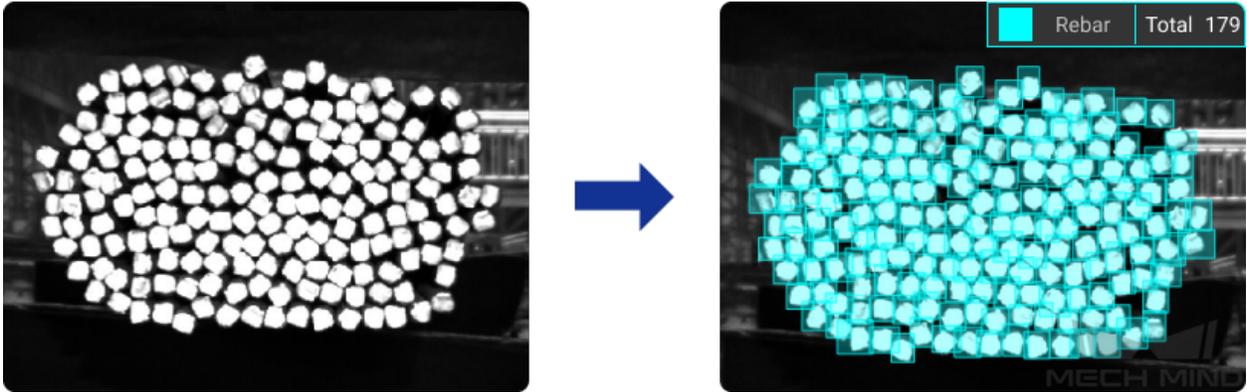
检测所有目标物体的位置，同时判断其类别。

用于检测固定位置工件的缺失，例如 PCB 中元器件缺失，也可用于物体计数。即使对成百上千个物体，也可以快速完成定位与计数。

- 检测互相压叠的转子。



- 对所有钢筋计数。

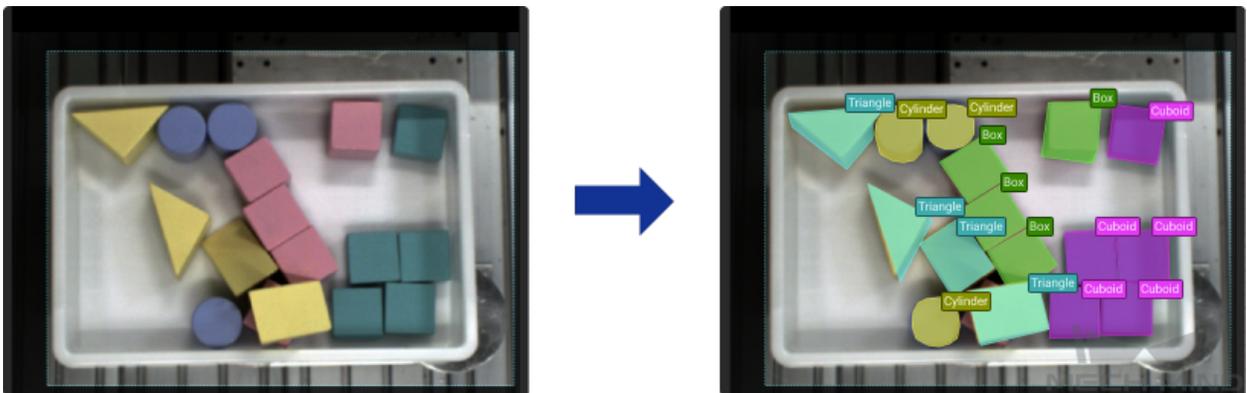


实例分割

分割出每个目标物体的轮廓，同时判断其类别。

比目标检测的分割结果更精细，可区分单类或多类的物体，并分割出相应的轮廓。用于拆码垛、上下料、货品拣选等场景，配合 Mech-Vision 和 Mech-Viz 完成物体抓取。

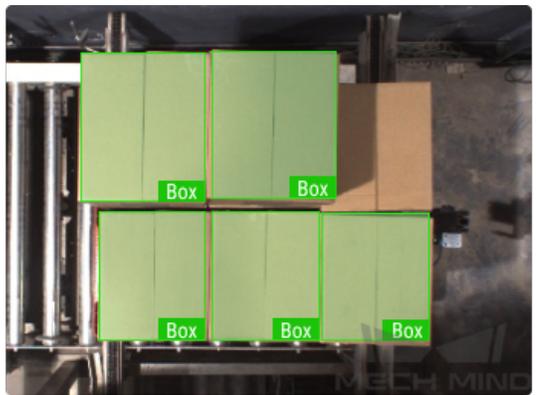
- 分割多种形状的积木。



- 分割散乱压叠的链轨节。



- 分割并排紧密码放的纸箱。



- *Mech-DLK 2.3.0* 更新说明
- *Mech-DLK 2.2.1* 更新说明

1.1 Mech-DLK 2.3.0 更新说明

1.1.1 更新显卡驱动版本

使用 Mech-DLK 2.3.0 软件前，显卡驱动版本需要更新至 472.50 以上

1.1.2 新增智能标注功能

在“缺陷分割”、“实例分割”、“目标检测”模块下，选择智能标注工具，点击待标注物体中心位置可以快速形成标注，右键删除多余标注区域，回车键完成标注。

1.1.3 新增多边形标注锚点增减功能

在“实例分割”和“目标检测”模块下，使用多边形标注工具标注完成后，若需修改标注结果，可在两个锚点之间的线上单击左键来增加锚点，选中锚点单击右键可以删除锚点。

1.1.4 新增标注印章工具

在“实例分割”和“目标检测”模块下，使用印章工具可以将已有的任意标注设置为模板，设置完成后可以直接点击进行标注。适用于图像中有多个同类别物体且摆放整齐的场景，可以提高标注效率。

1.1.5 新增缩放预览功能

支持预览单张图像和剪切后的小图，参考缩放预览功能。

1.1.6 优化网格剪切工具

优化网格剪切功能，剪切网格后，可以点击网格单元左上角的方框来选中单元图像，点击小图右上角按钮可以预览。

1.1.7 优化数据筛选机制

新增结果类型筛选项，可按“正确的结果”、“错误的结果”、“过检”、“漏检”标签来筛选数据。新增数据类型筛选项：“已标注为 OK”和“已标注为 NG”选项。

1.1.8 深度学习环境内置

深度学习环境内置到 Mech-DLK 软件里，不需要单独安装环境即可训练模型。

1.2 Mech-DLK 2.2.1 更新说明

1.2.1 新增“图像分类”模块新增类激活图显示功能

模型训练完成后，点击生成类激活图可以通过热力图的形式体现特征的权重，模型基于这些特征将该图像分为此类别。颜色越红的区域，在分为此类别的过程中所占的权重越大。

1.2.2 新增 CPU 版本模型的验证及导出功能

- 图像分类、目标检测：训练完成后，导出模型前可以将部署设备选为 CPU 或 GPU。
- 实例分割：训练模型前，设置训练参数。导出模型时，可以在 CPU/GPU 之间选择部署设备，具体如下：
 - *CPU* 轻量模型：训练模型前，将训练参数 模型类型选为 轻量（推荐使用 CPU 部署），导出模型部署时，可以将 部署设备选为 CPU 或 GPU。
 - *GPU* 标准模型：训练模型前，将训练参数 模型类型选为 标准（推荐使用 GPU 部署），导出模型部署时，推荐将 部署设备选为 GPU。

本章将介绍运行 Mech-DLK 软件必须满足的软硬件要求，以及如何安装 Mech-DLK 软件。

2.1 软硬件配置要求

	Mech-DLK Pro-Run	Mech-DLK Pro-Train/Standard
操作系统	Windows 10 及以上	
CPU	Intel® Core™ i5 及以上	Intel® Core™ i7 及以上
内存	8 GB 及以上	16 GB 及以上
显卡	GeForce GTX 1650 (4GB) 及以上	GeForce GTX 2070 (8GB) 及以上
显卡驱动	驱动版本 472.50 及以上	

2.1.1 显卡要求

- 计算机显卡算力至少等同于 Nvidia GeForce 6.1。
- [点击此处](#) 可以查看显卡的计算能力，如下图：

GeForce and TITAN Products

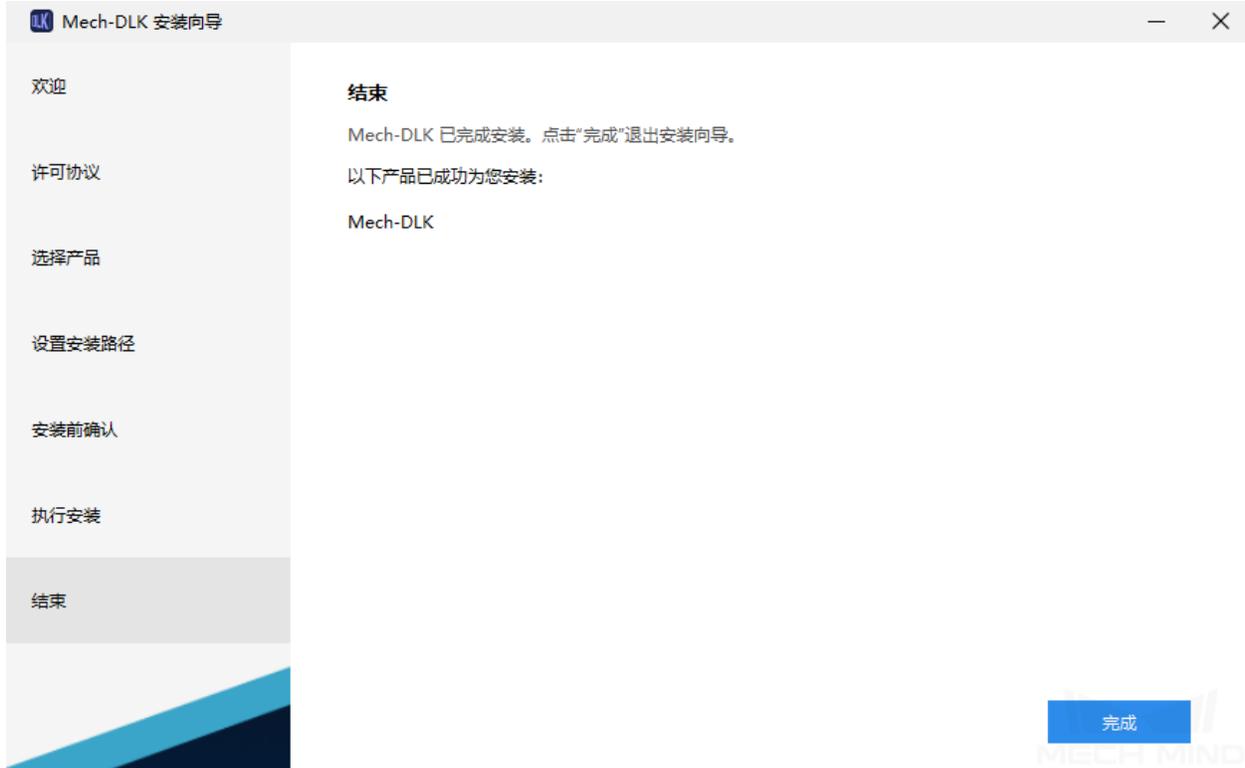
GPU	Compute Capability
GeForce RTX 3060 Ti	8.6
GeForce RTX 3060	8.6
GeForce RTX 3090	8.6
GeForce RTX 3080	8.6
GeForce RTX 3070	8.6
GeForce GTX 1650 Ti	7.5
NVIDIA TITAN RTX	7.5
GeForce RTX 2080 Ti	7.5
GeForce RTX 2080	7.5
GeForce RTX 2070	7.5
GeForce RTX 2060	7.5
NVIDIA TITAN V	7.0
NVIDIA TITAN Xp	6.1
NVIDIA TITAN X	6.1
GeForce GTX 1080 Ti	6.1

GeForce Notebook Products

GPU	Compute Capability
GeForce RTX 3080	8.6
GeForce RTX 3070	8.6
GeForce RTX 3060	8.6
GeForce RTX 3050 Ti	8.6
GeForce RTX 3050	8.6
GeForce RTX 2080	7.5
GeForce RTX 2070	7.5
GeForce RTX 2060	7.5
GeForce GTX 1080	6.1
GeForce GTX 1070	6.1
GeForce GTX 1060	6.1
GeForce GTX 980	5.2
GeForce GTX 980M	5.2
GeForce GTX 970M	5.2
GeForce GTX 965M	5.2

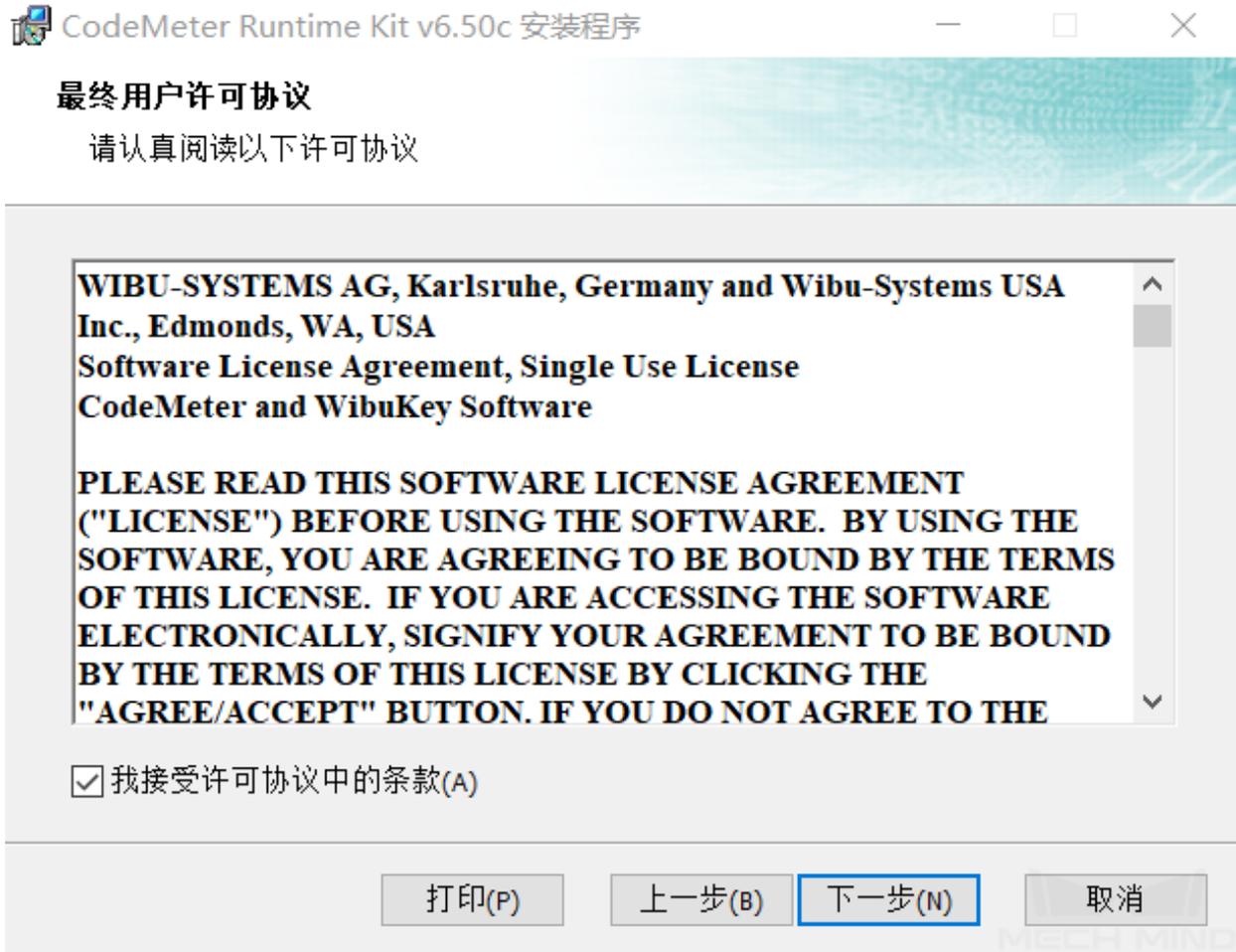
2.2 安装 Mech-DLK 软件

1. 下载 Mech-DLK 软件安装包。
2. 双击 Mech_DLK_installer_校验码.exe 文件，并按照提示安装 Mech-DLK 软件。若安装成功，则会出现如下窗口：



2.3 加密狗驱动安装

1. 安装加密狗驱动，勾选下图选项，在后续窗口中均点击 下一步 直至安装完成。



2. 若安装成功，则会出现如下窗口：



CodeMeter Runtime Kit v6.50c 安装向导 已完成

单击“完成”按钮退出安装向导。

上一步(B)

完成(F)

取消

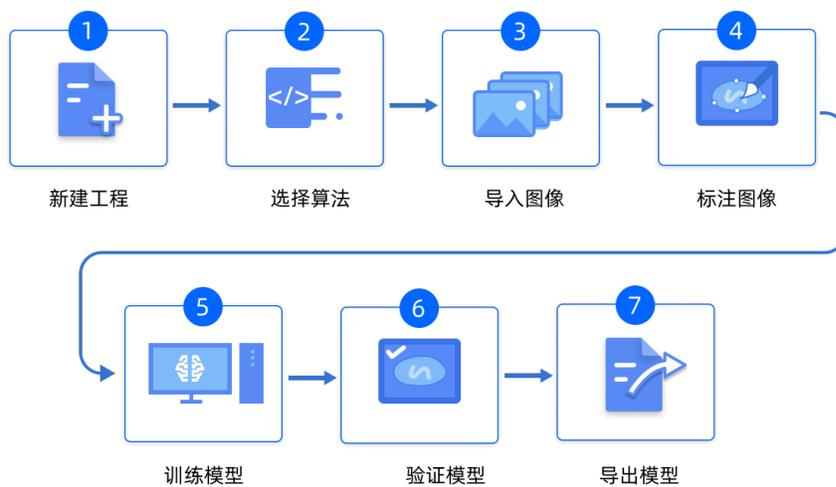
开始训练第一个模型

本章以“缺陷分割”模块为例，介绍如何快速训练深度学习模型。数据来自网口图像数据集。

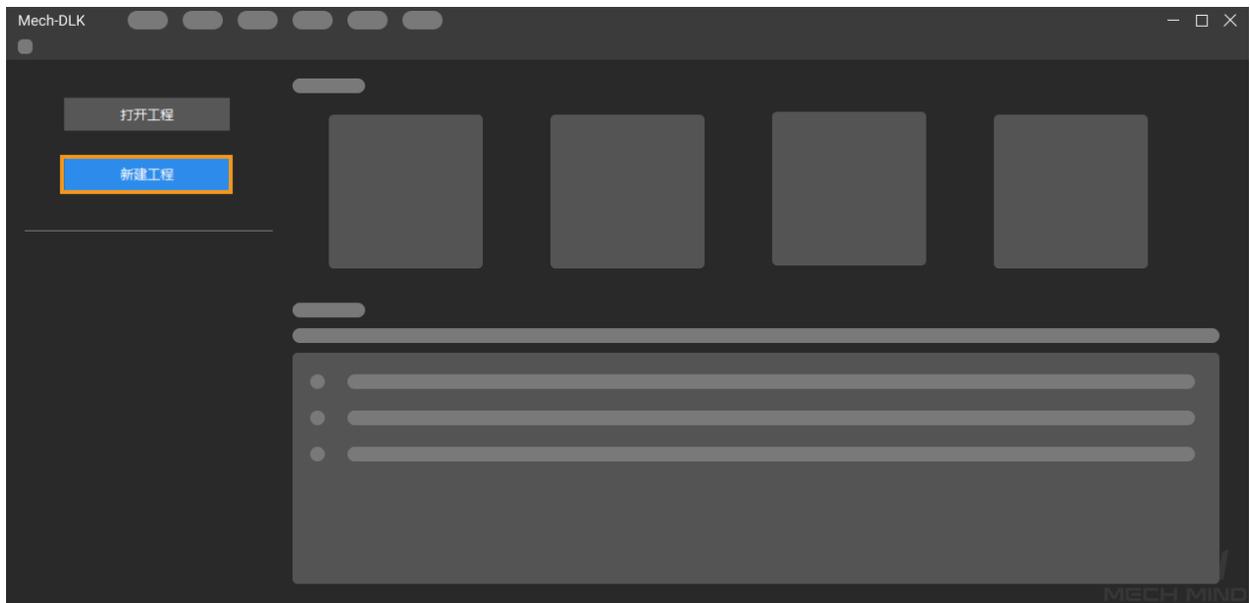
准备工作

- 下载网口图像数据集压缩包（点击下载）并解压缩。

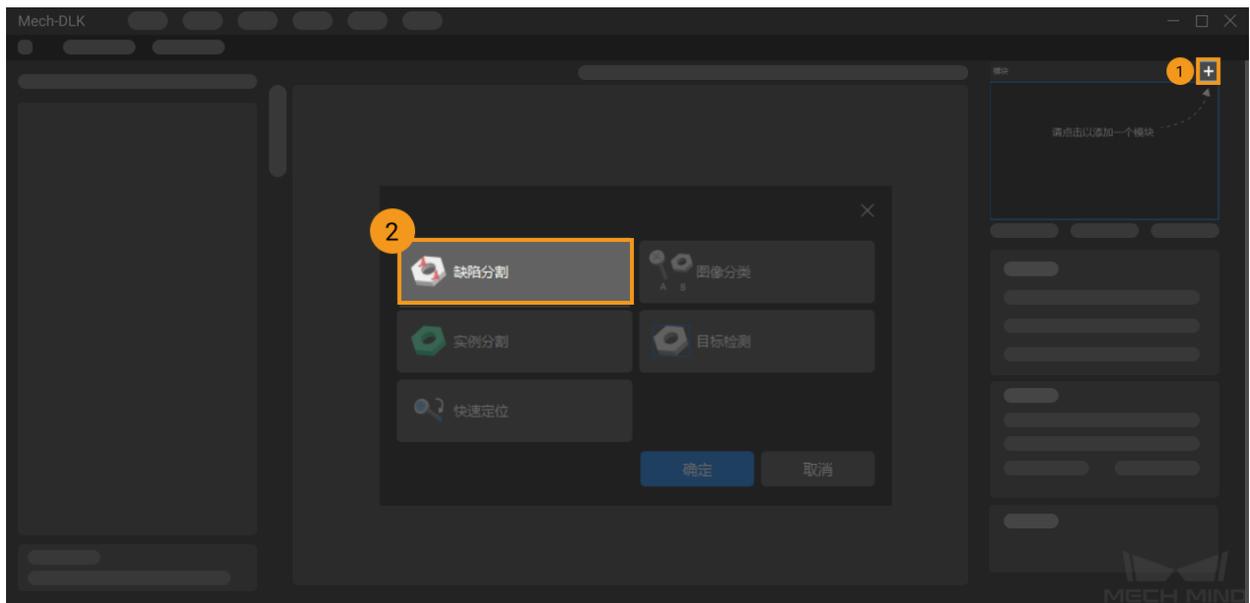
训练流程



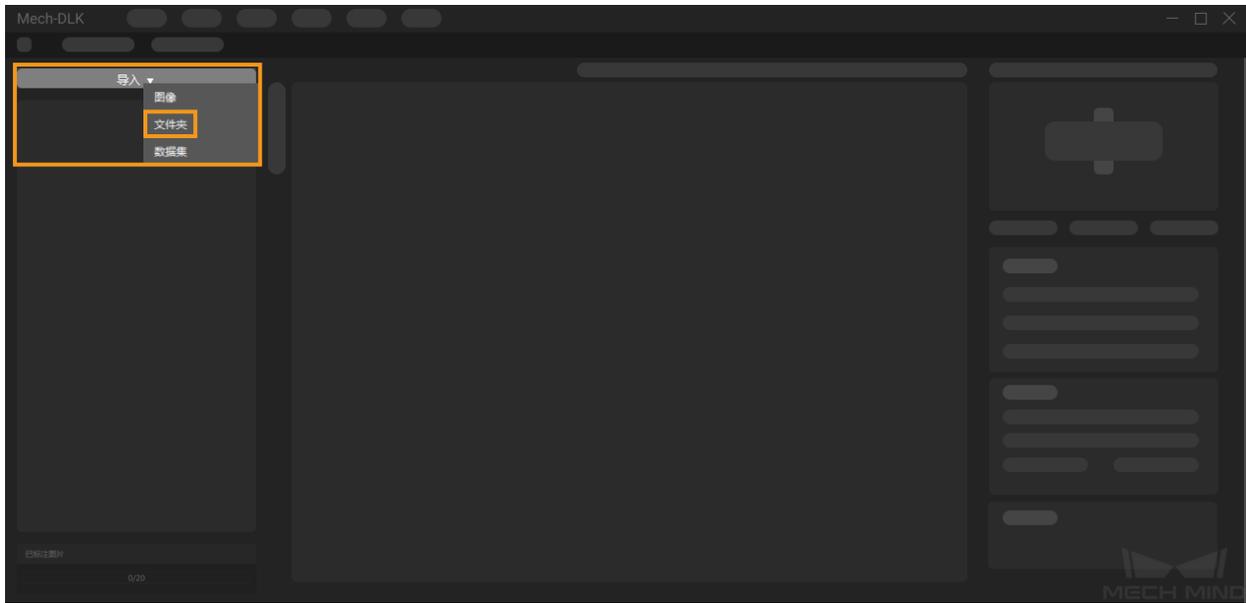
1. **新建工程**：点击初始页面上的 新建工程，选择工程路径并输入工程名以新建一个工程。



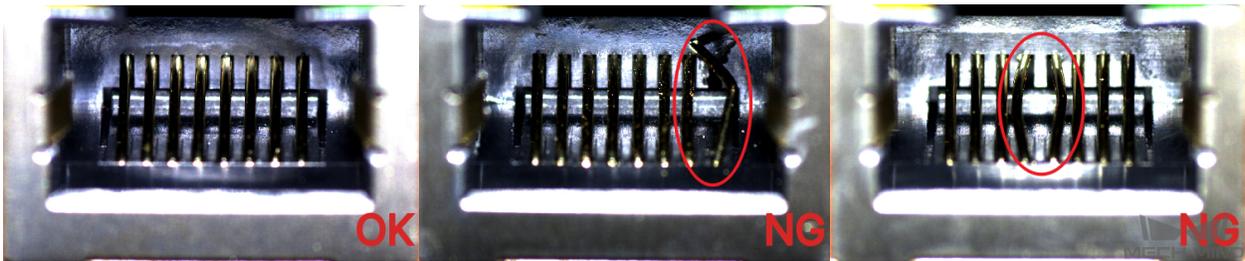
2. 添加“缺陷分割”模块：点击页面右侧模块栏中的  添加模块，选择 缺陷分割后点击确定。



3. 导入图像：点击左上方的 导入，选择 文件夹导入准备好的图像数据集。



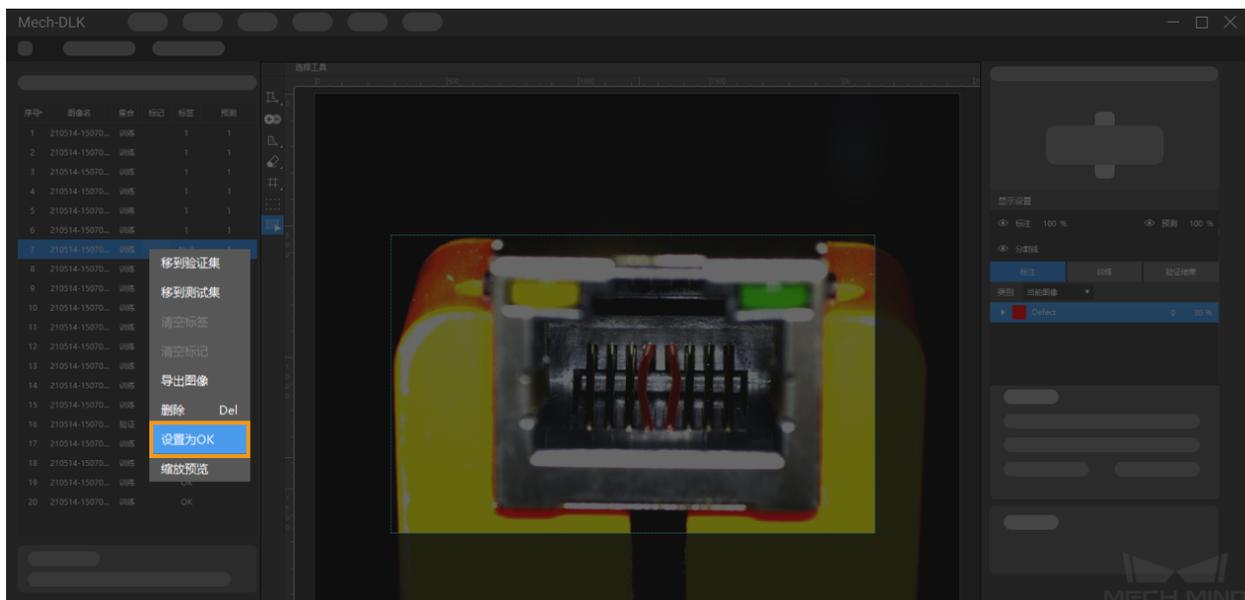
4. **标注图像**：本例中，需要用户标注数据集中的 OK 图像，以及含网口金手指弯折、断裂缺陷情况的 NG 图像。



对于 NG 图像，请左键长按或右键单击图像左侧工具栏 ，根据缺陷形状选择合适的画笔工具，标注出图像中所有缺陷的区域。点击  则进入橡皮擦模式，可擦掉已标注的缺陷区域。



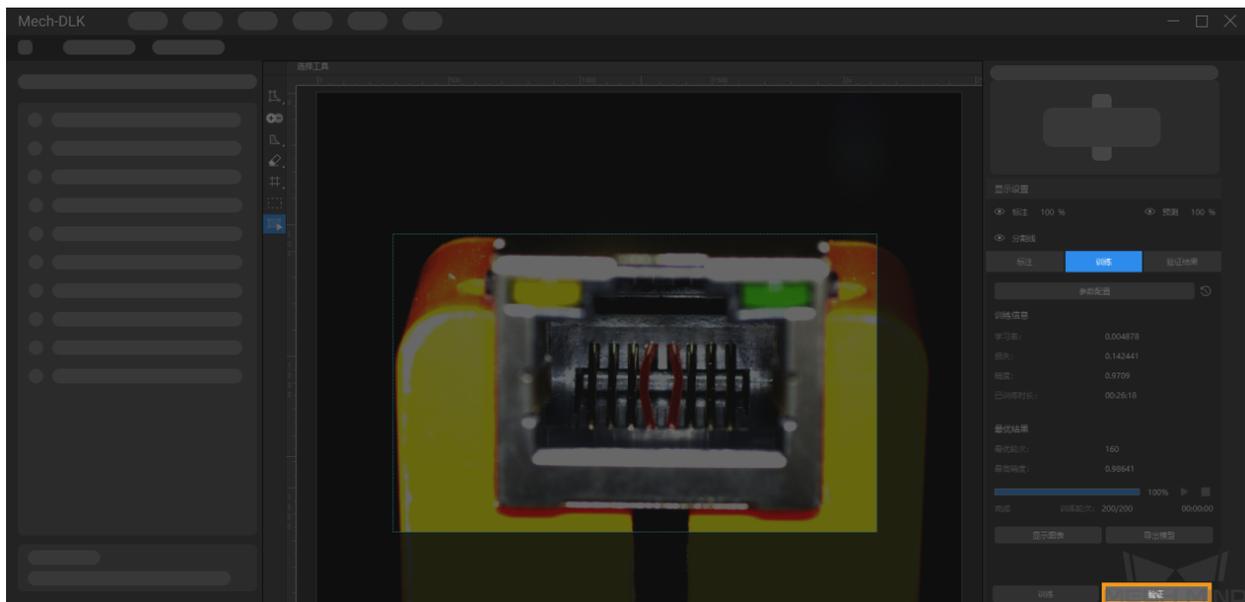
对于 OK 图像，请在左侧图像列表中选中无缺陷的图像，单击鼠标右键选择 设置为 OK。请确保训练集中至少有一张图像标注为 OK。



5. 训练模型：在界面右侧，切换至训练栏，点击右下方的 训练开始训练模型。



6. **验证模型**：模型训练完成后，点击 验证可以验证并查看模型效果。



7. **导出模型**：点击训练栏下方的 导出模型，选择文件路径后便可导出模型到指定文件夹，模型文件为 model.dlpack 格式。用户可以根据需求部署模型。

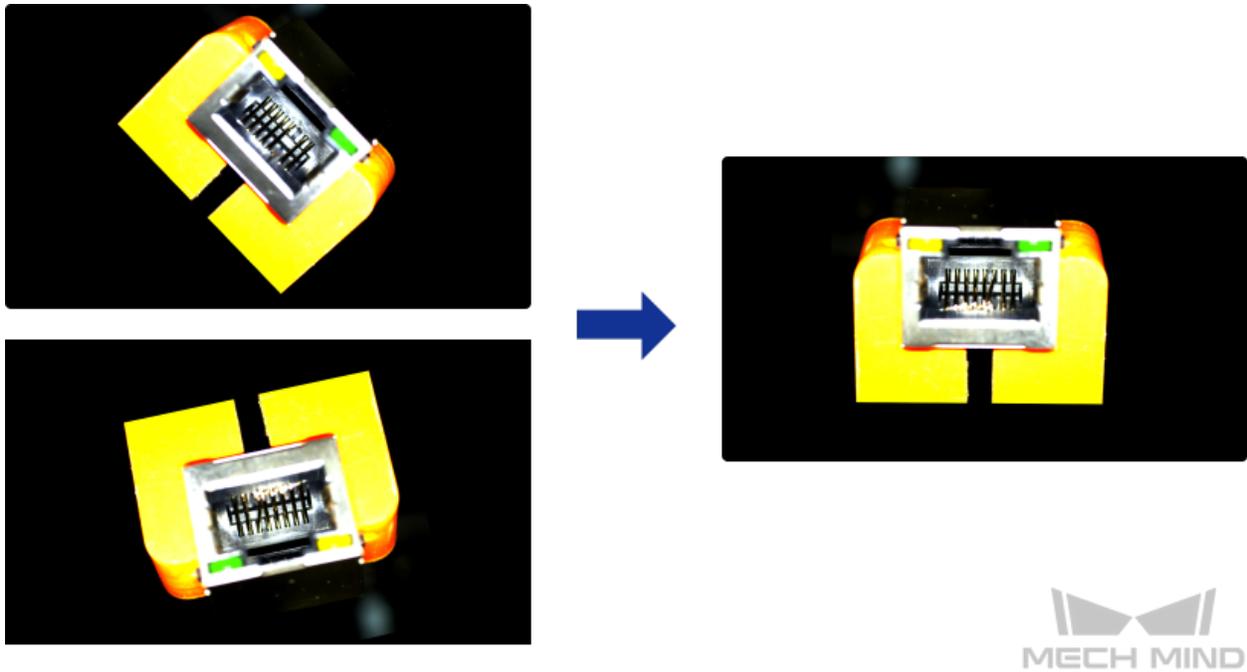


4.1 算法介绍

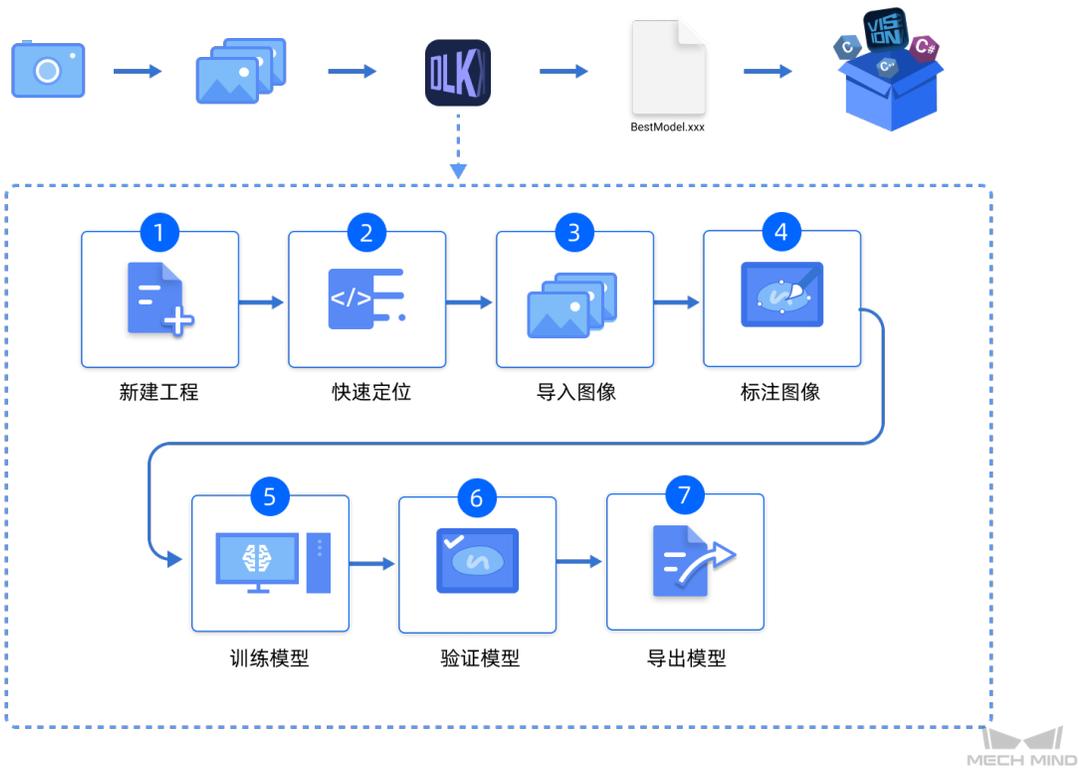
快速定位可以检测图像目标区域中的物体并旋转图像至指定朝向。

4.1.1 应用场景

电子制造业：适用于电子元器件散乱来料时，定位不同角度元器件并将其旋转为一致角度的场景。



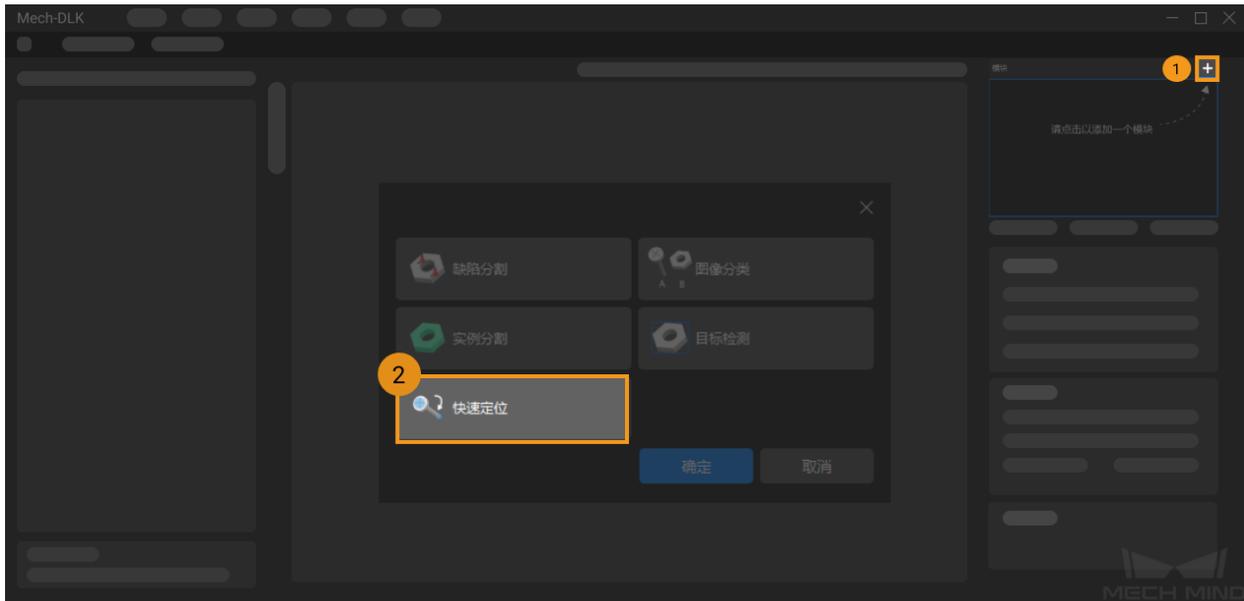
4.1.2 应用流程



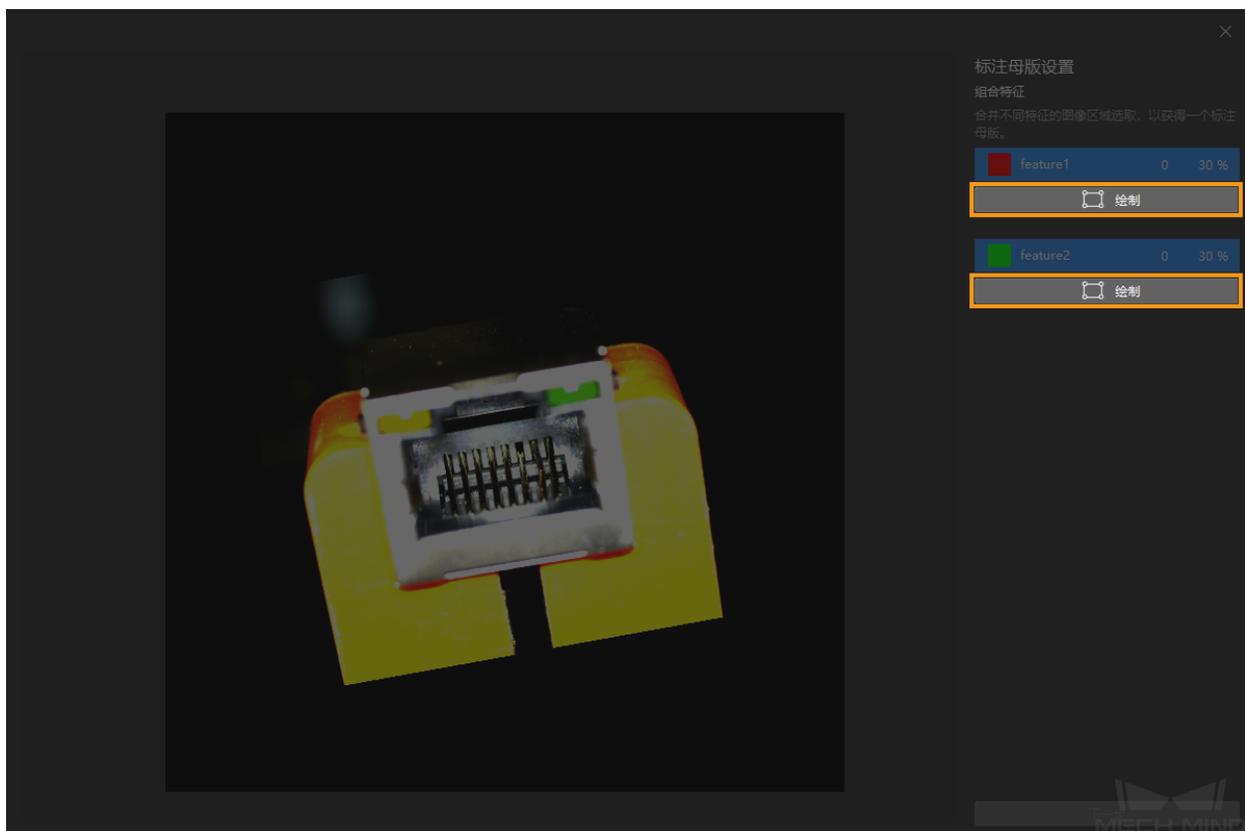
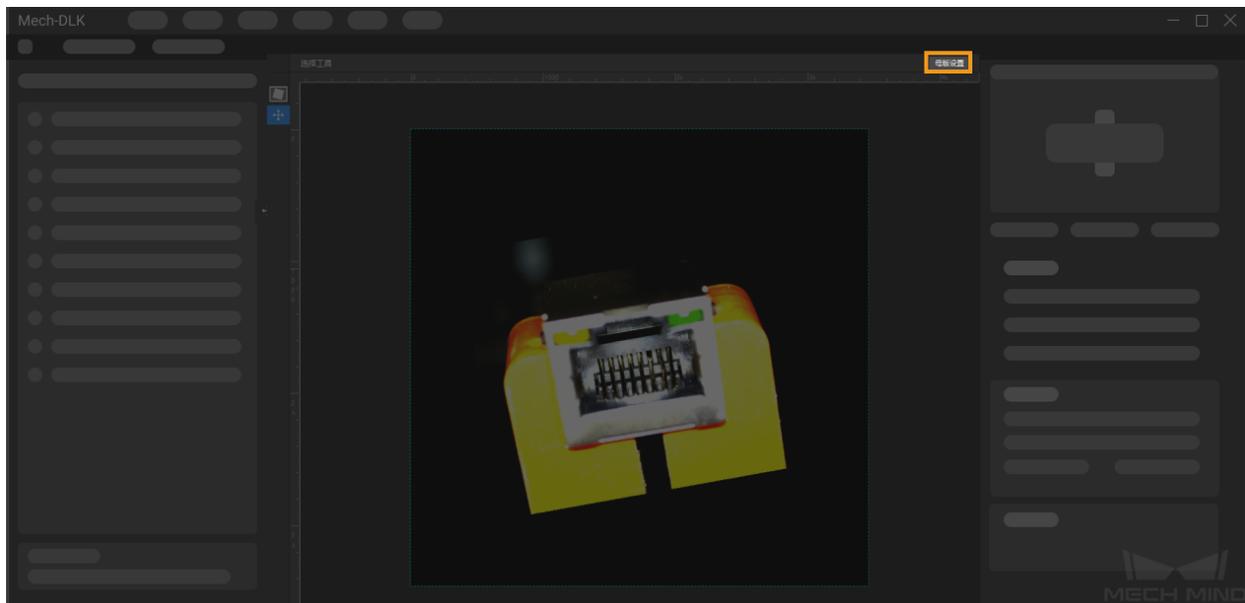
4.2 使用快速定位模块

本章提供网口数据集（点击下载），带领用户使用快速模块训练模型将图像中的物体旋转至指定朝向。

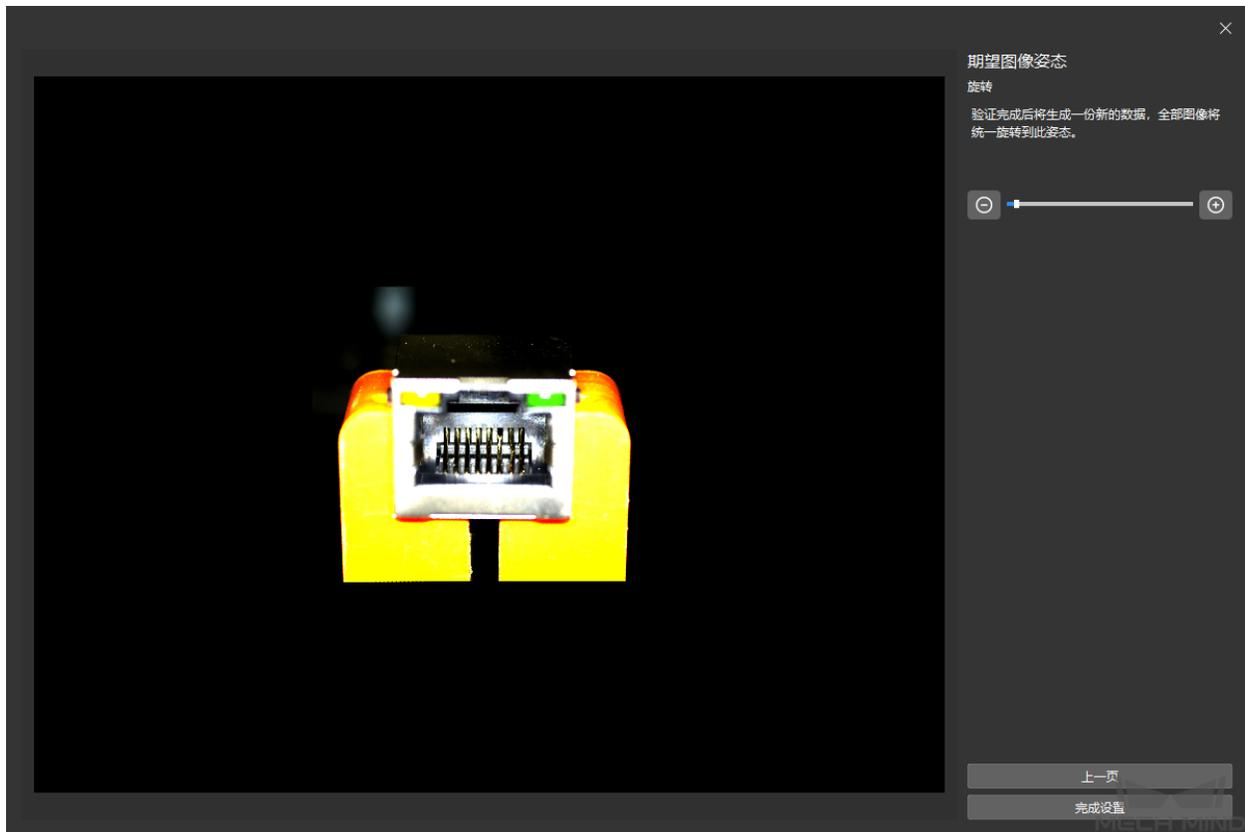
1. 新建工程并添加快速定位模块：点击初始显示页面上的 新建工程，选择工程路径并输入工程名以新建一个工程。再点击右上角的 ，选择快速定位模块。



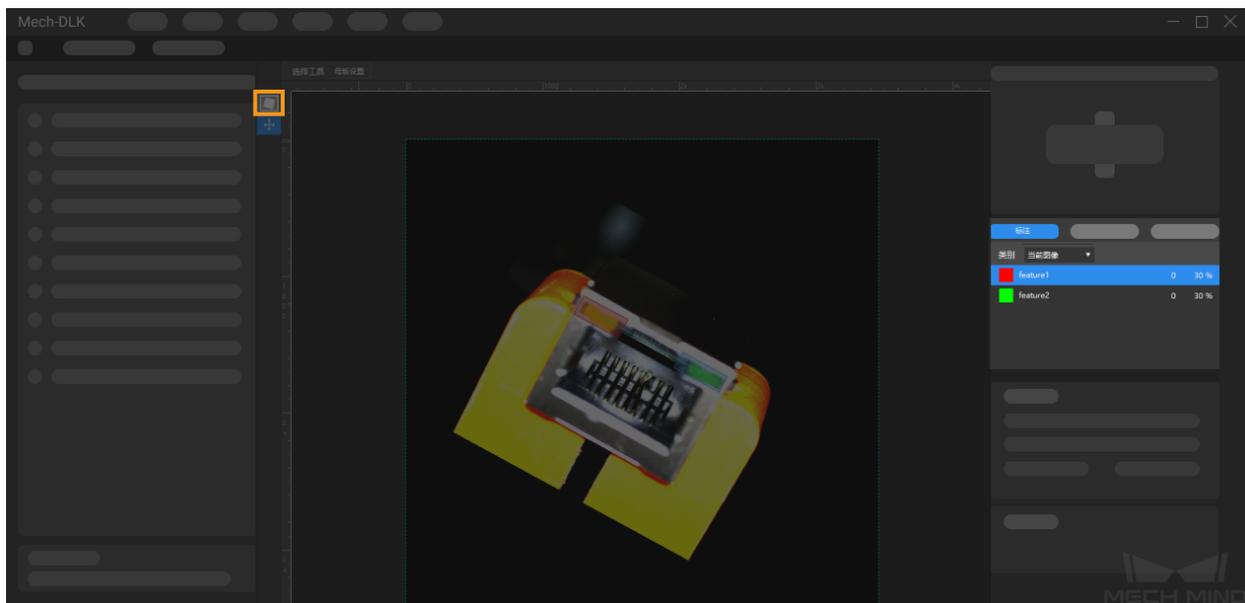
2. 母版设置：点击 母版设置，选取两处特征区域。点击 **feature1** 下的  框选第一处特征，再点击 **feature2** 下的  框选第二处特征，选中特征框移动旋转按钮可以调整角度。



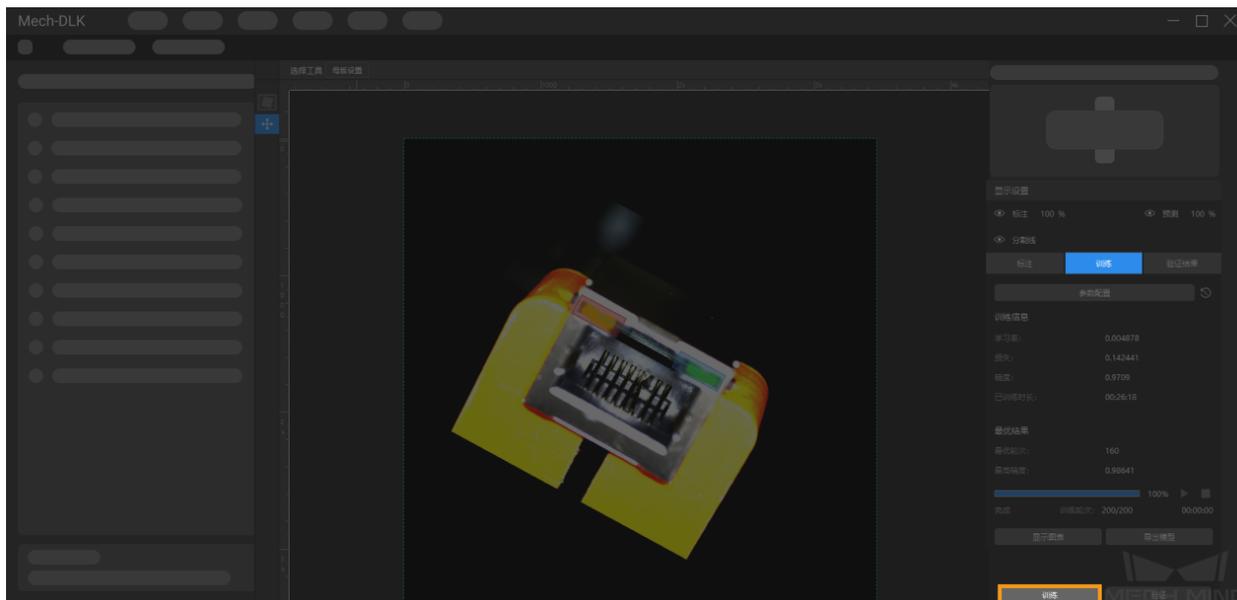
拖动滑块或点击  和  调整角度到期望图像朝向后点击 **完成设置**。



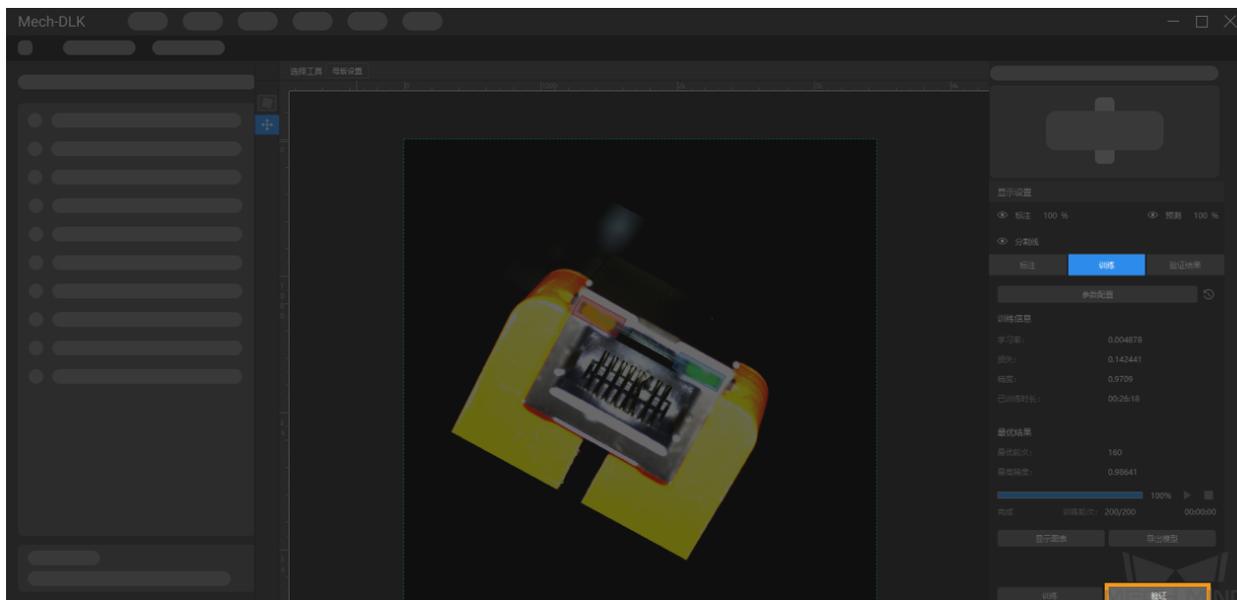
3. 修改标注：点击  后再点击标注界面，会出现标注信息，选中矩形框拖拽或旋转按钮调整角度到相应的位置角度。



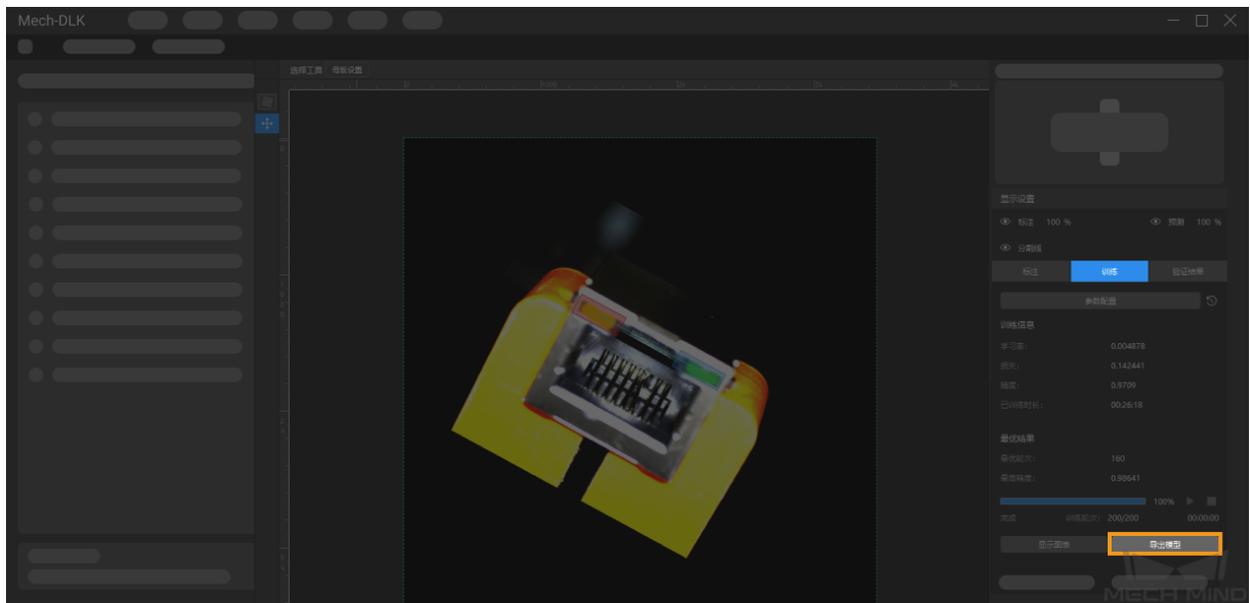
4. 训练模型：点击 训练开始训练模型。



5. 验证模型：训练结束后，点击 验证可以验证并查看模型识别效果。



6. 导出模型：点击 导出模型并选择存放路径即可导出模型到工程文件夹，用户可以根据需求部署模型。

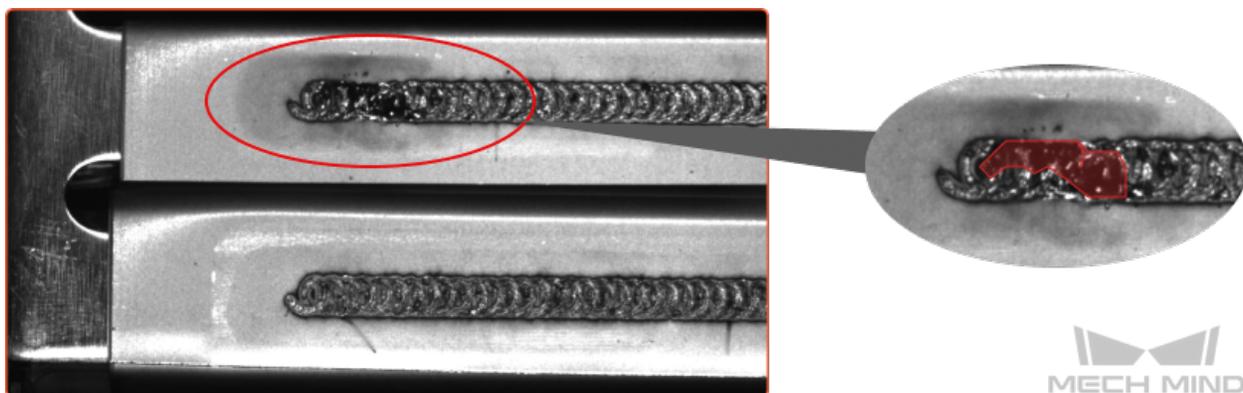


5.1 算法介绍

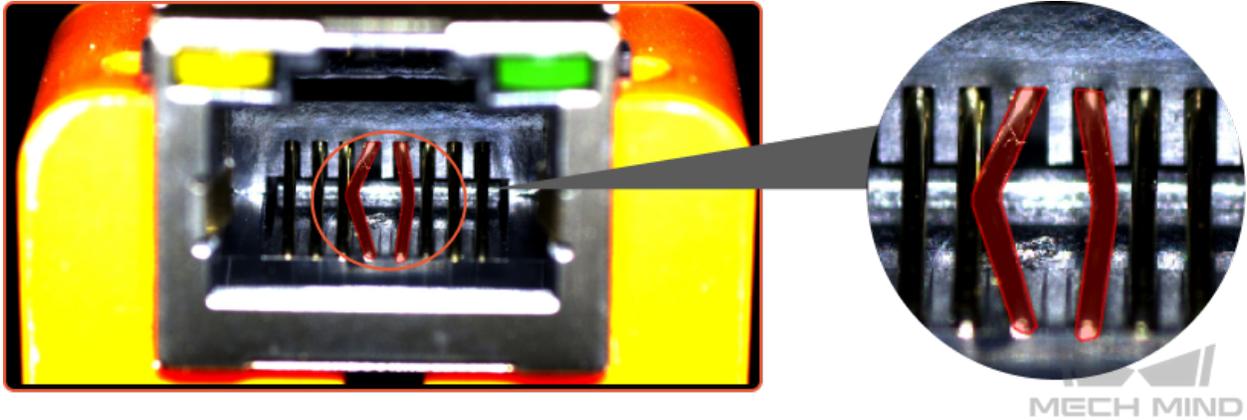
缺陷分割可以判断并分割出图像中的缺陷区域。

5.1.1 应用场景

1. **新能源行业：**用于检出各种类型的缺陷。在缺陷微小、背景复杂、工件位置不固定等复杂情况下依然适用（如锂电池的焊缝检测及外观检测等）。



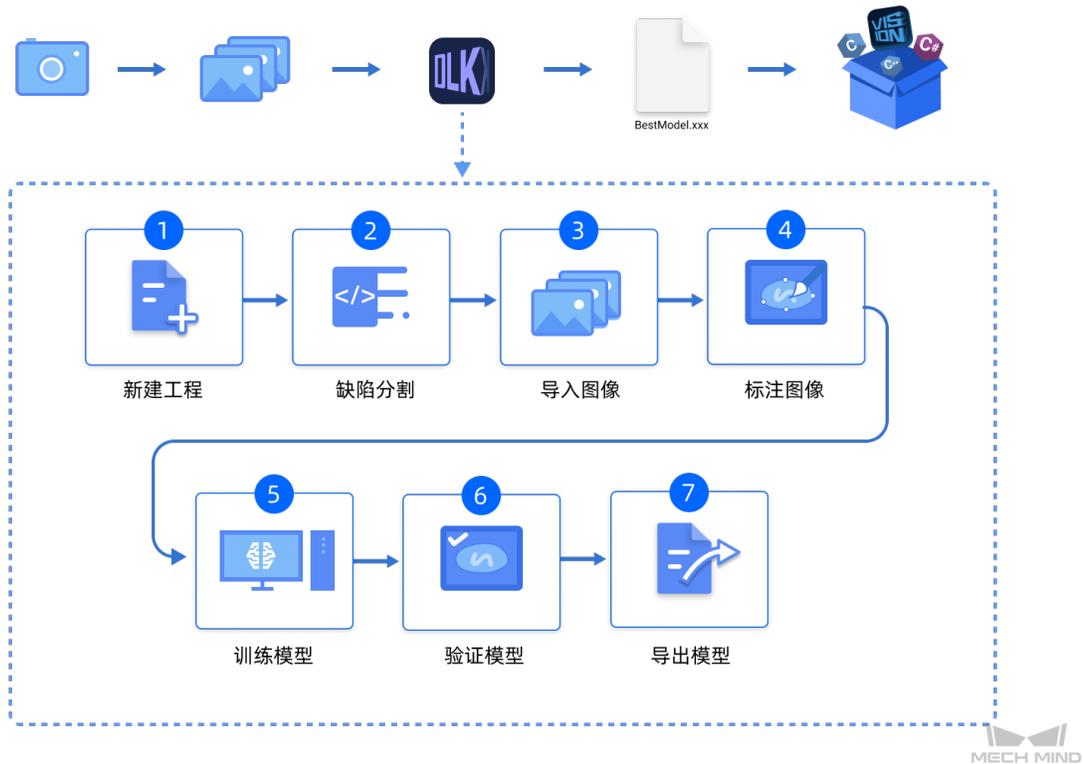
2. **电子制造业：**适用于功能模组和电子元器件的缺陷检出，如污渍、气泡、划痕等表面缺陷。



3. **PCB、印刷、日用品制造等行业：**适用于 PCB 基板、接插件、印刷产品、日用品等物品表面的缺陷检测、异形检测等，如划伤、异物等缺陷。



5.1.2 应用流程



5.1.3 应用要点

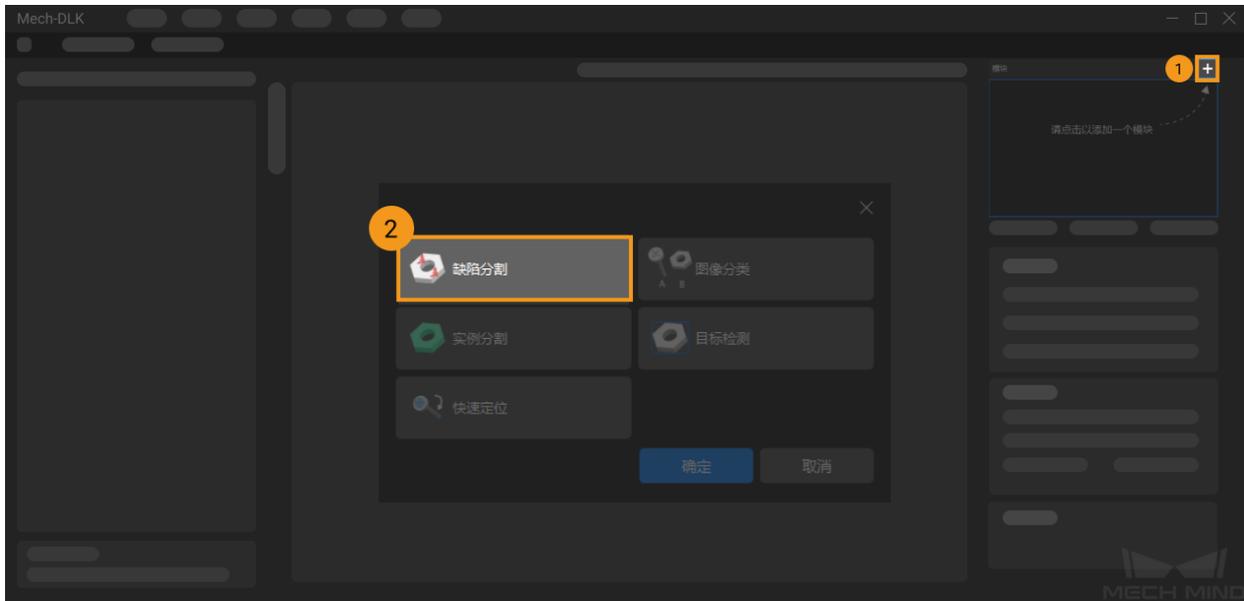
以下几点有助于提高模型质量：

1. 确保标注质量 [前往章节 >](#)。
2. 设置合理的感兴趣区域（ROI） [前往章节 >](#)。
3. 选择合适的训练集 [前往章节 >](#)。

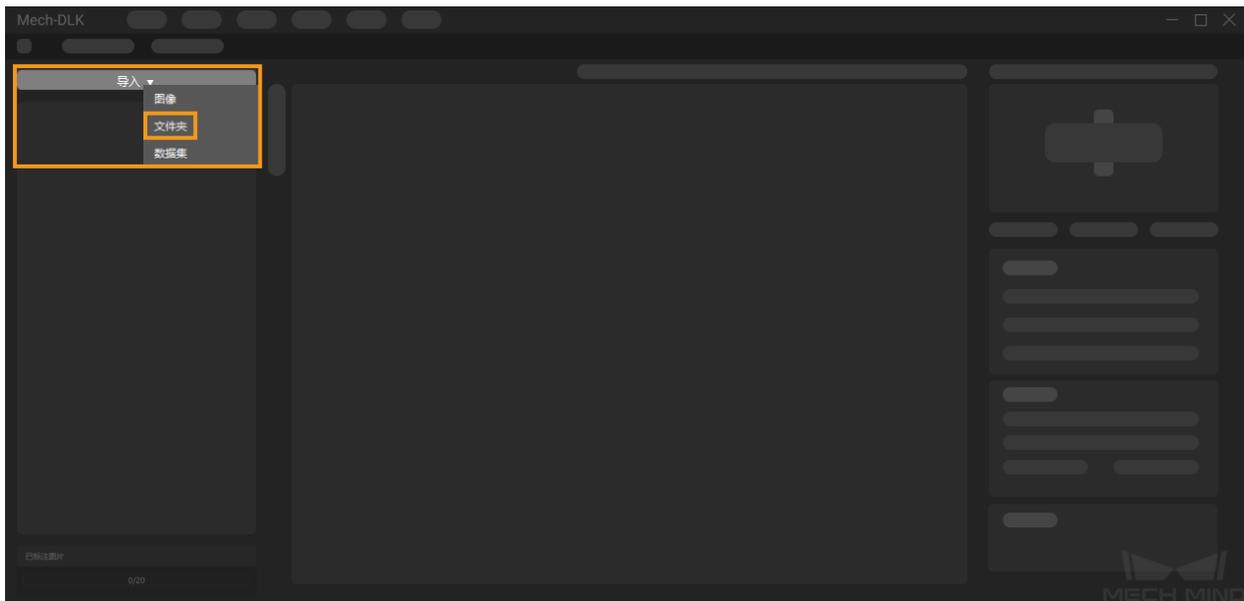
5.2 使用缺陷分割模块

本章提供 Mech-DLK 示例工程中的网口金手指图像数据集（[点击下载](#)），带领用户使用“缺陷分割”模块训练模型。区别于开始训练第一个模型的内容，本章将从标注缺陷、选取训练验证集以及调整模型验证标准等方面，说明如何操作以达到示例工程中检出口金手指的弯折、断裂缺陷的效果。

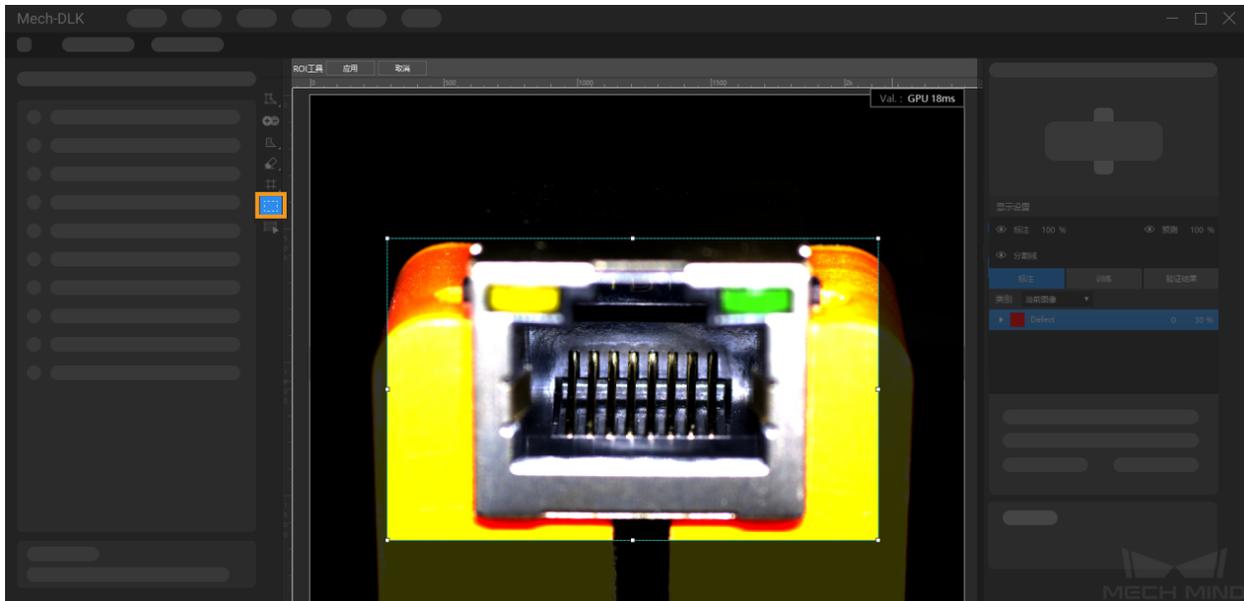
1. 新建工程并添加“缺陷分割”模块：点击初始页面上的 新建工程，选择工程路径并输入工程名以新建一个工程。再点击右上角的 ，选择“缺陷分割”模块。



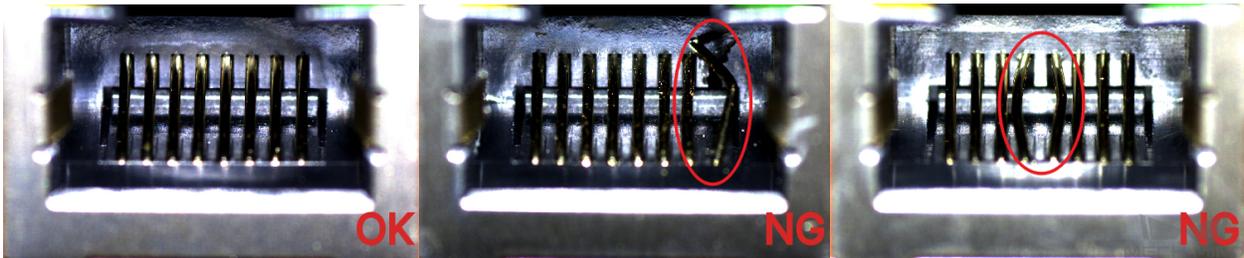
2. 导入网口金手指图像数据集：解压缩下载的数据集压缩包，点击左上方的 导入，选择 文件夹 导入下载的图像数据集。金手指图像数据集中包含金手指弯折、断裂以及完好的情况。



3. 截取 ROI：点击设置 ROI 选项  来从图像中框选网口上的金手指作为感兴趣区域，并点击左上方的 应用 确认使用。截取 ROI 的目的是减少无关背景信息的干扰。



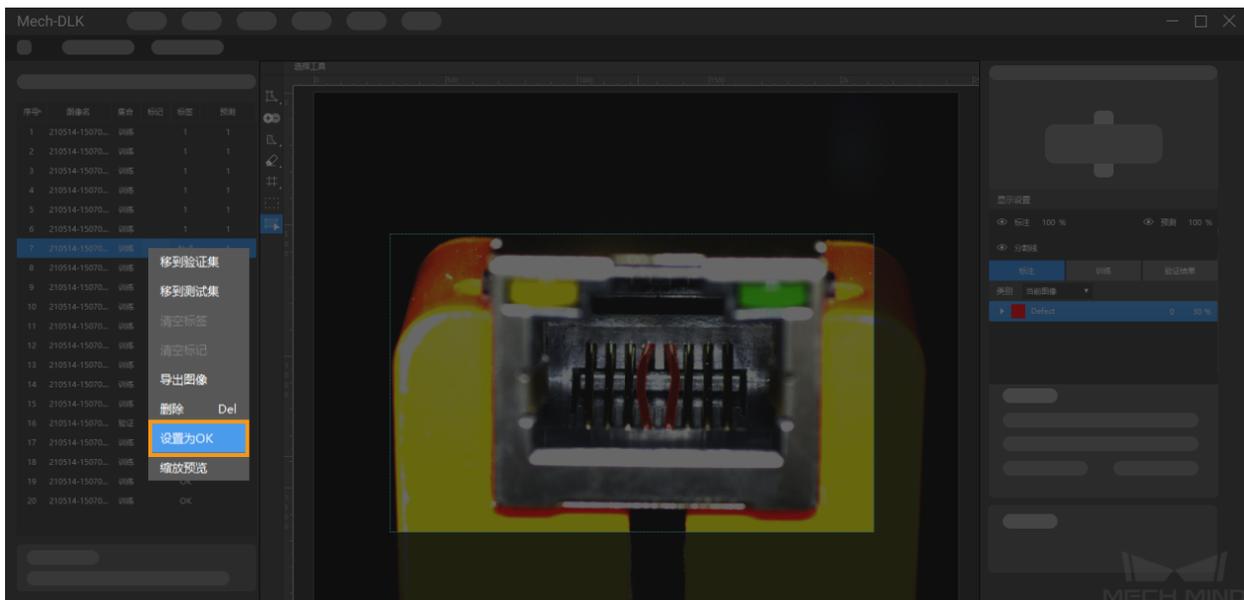
4. 标注图像：本例中需要标注数据集中的 OK 图像，以及包含网口金手指弯折、断裂缺陷的 NG 图像。



- 对于 NG 图像，请使用标注工具标出缺陷区域。标注时应注意画笔紧贴缺陷边缘标注，请避免包含大量非缺陷区域的情况。标注质量是影响模型效果的最主要因素，更多标注规则参考确保标注质量。



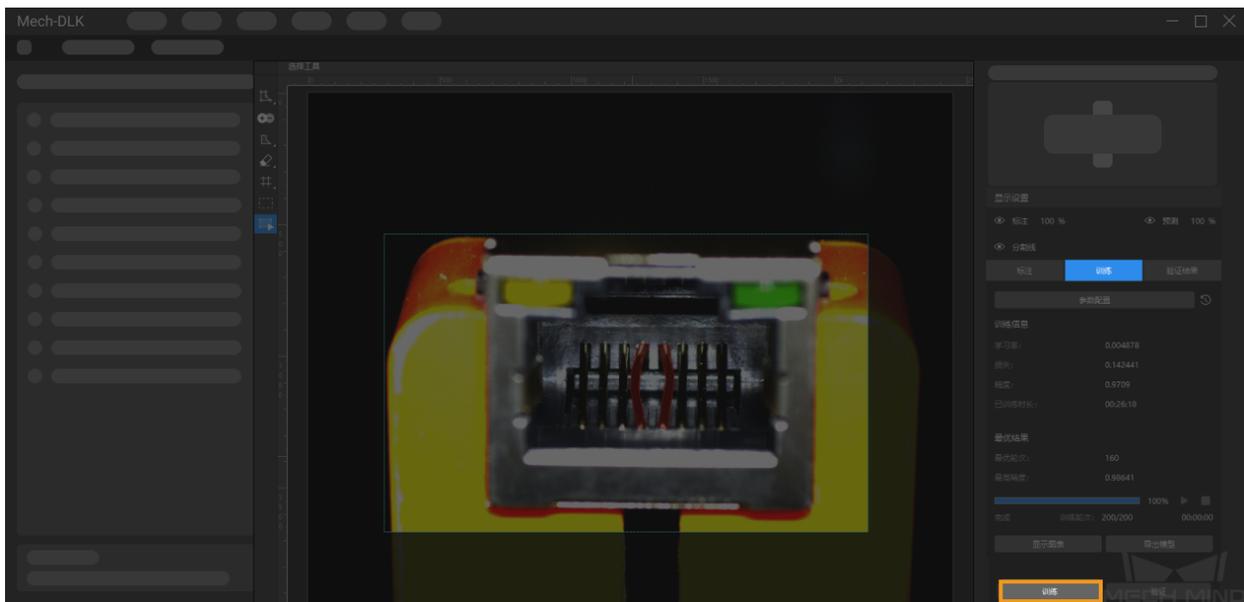
- 对于 OK 图像，请在左侧图像列表中选中图像，单击鼠标右键，选择 设置为 OK。



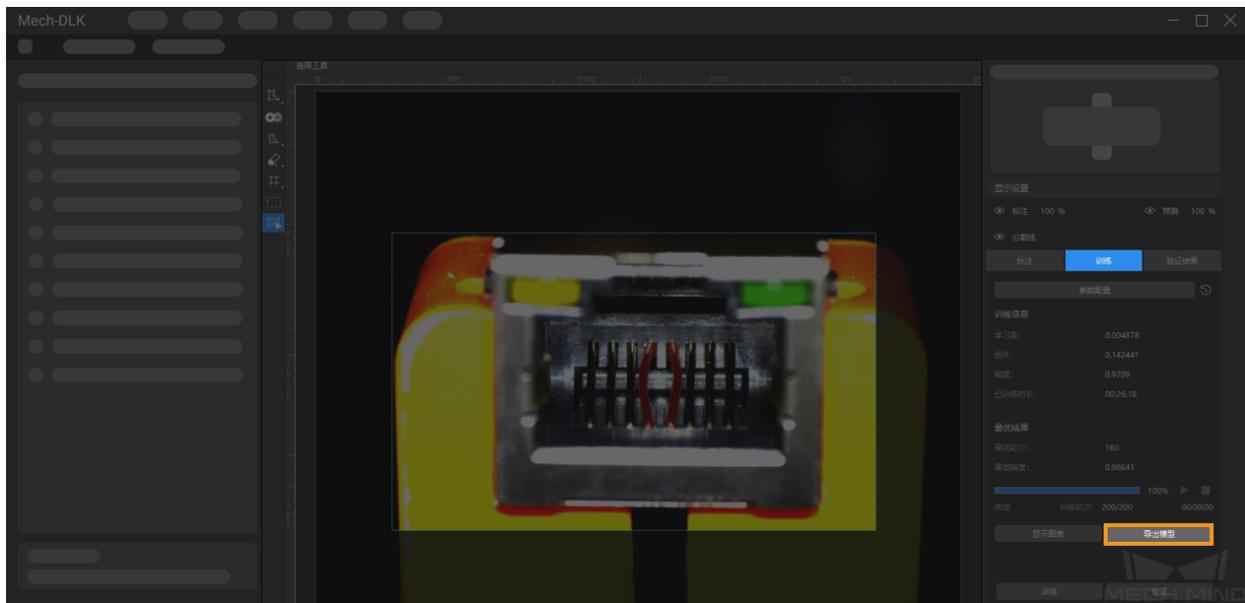
5. 划分训练集与验证集：需要确保训练集和验证集分别涵盖所有的缺陷种类，且至少包含 1 张 OK 图像。如果训练集或验证集中缺少某种缺陷样本，“缺陷分割”模块将无法有效学习该缺陷特征，导致训练出的模型难以识别该缺陷。此时应右键单击图像名称以切换图像归属于训练集还是验证集。



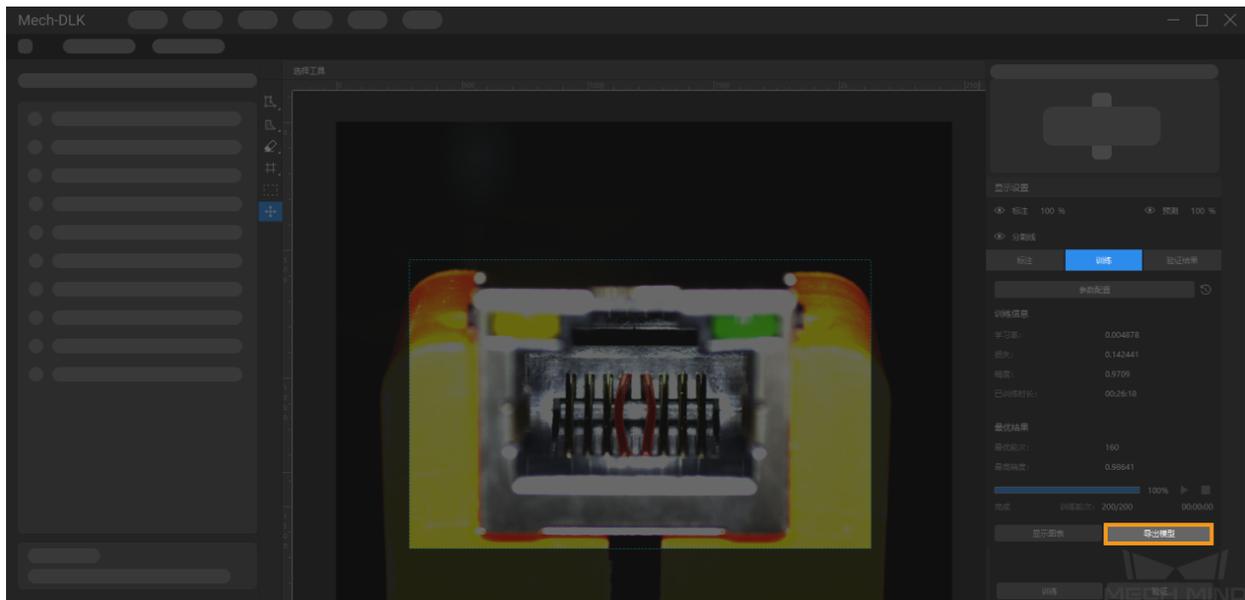
6. 训练模型：使用默认参数设定，点击 训练开始训练模型。



7. 验证模型：训练结束后，点击 验证可以验证并查看模型识别效果。用户还可使用缺陷筛选工具 调整缺陷判定规则。



8. 导出模型：点击 导出模型选择存放路径即可导出模型到工程文件夹，用户可以根据需求部署模型。



5.3 如何训练高质量的模型

工业质检通常对漏检率和过检率都有严格的标准，因此缺陷分割模型的质量非常重要。本章将介绍最影响模型质量的几个因素，以及如何训练高质量缺陷分割模型。

- 确保标注质量
- 设置合理的感兴趣区域 (ROI)
- 选取合适的训练集

5.3.1 确保标注质量

标注质量是影响模型效果的最大因素。实际项目中，因标注质量导致模型效果差的情况占比高达 **90%** 以上。因此，如果模型效果不佳，应优先排查标注质量问题。

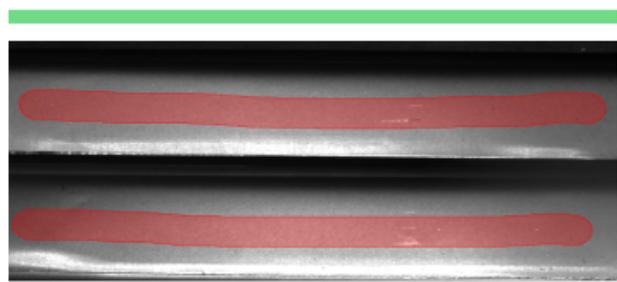
标注质量应从一致性、完备性、精确性、确定性几个方面考虑：

1. **一致性**：确保缺陷标注方式的一致性，禁止出现同一类型缺陷使用不同方式标注的情况。

错误示例：对同一种缺陷使用两种标注方式标注。



正确示例：用一种标注方式标注同一种缺陷。

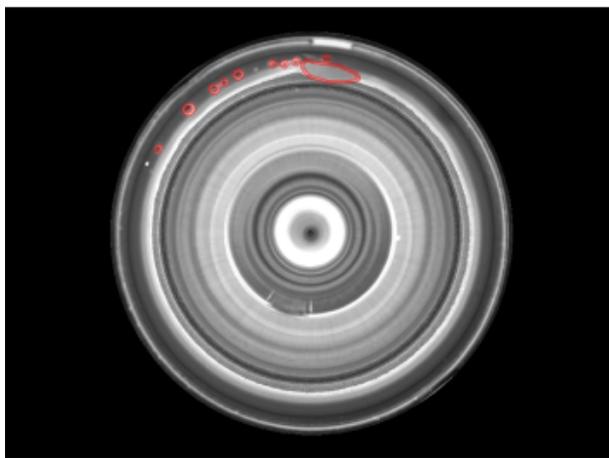


标注提示：需要标注焊接缺失区域，虽然两种标注方式都正确，但标注时只能选择其中一种标注。

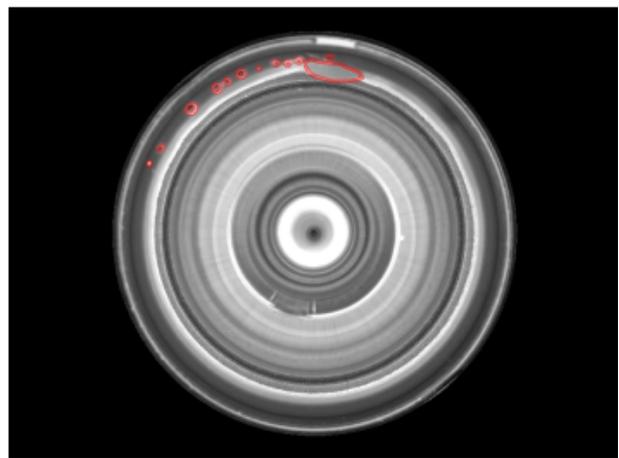


2. **完备性**：确保对所有符合缺陷评判标准的区域都进行了标注，禁止出现遗漏标注的情况。

错误示例：漏标注



正确示例：正确标注



标注提示：需要标注所有缺陷的区域，左图漏标注了较小的气泡缺陷。



3. **精确性**：确保标注区域贴合缺陷边缘，精细标注缺陷区域，禁止粗放使用大面积标注覆盖缺陷区域。

错误示例：大面积覆盖



正确示例：正确标注



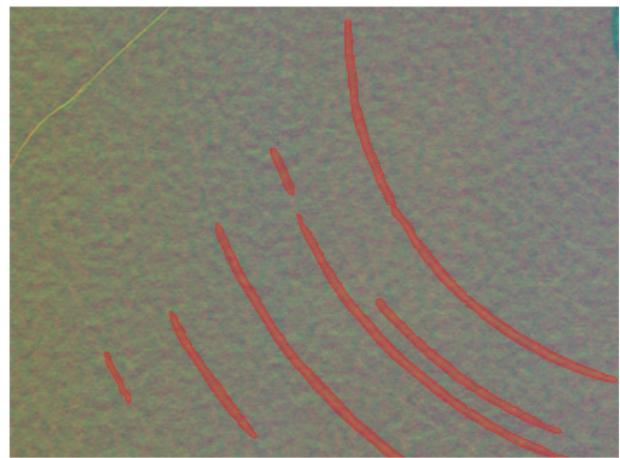
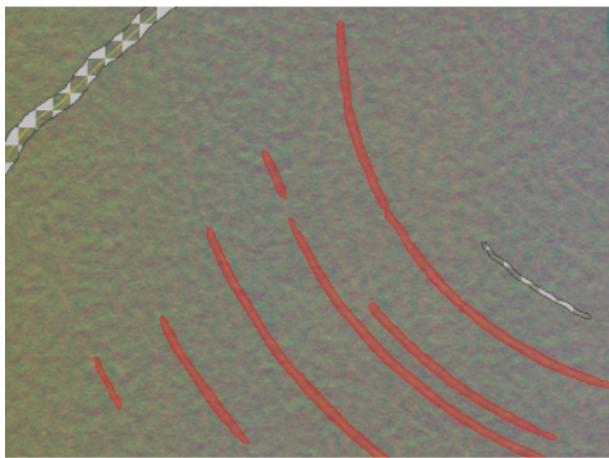
标注提示：标注缺陷区域应贴合缺陷边缘，禁止大面积覆盖。



4. **确定性**：对于模棱两可的缺陷，当无法判定是否符合缺陷判定标准时，可以使用掩膜多边形工具遮盖缺陷区域。

正确示例：掩膜多边形工具遮盖缺陷区域

正确示例：正确标注

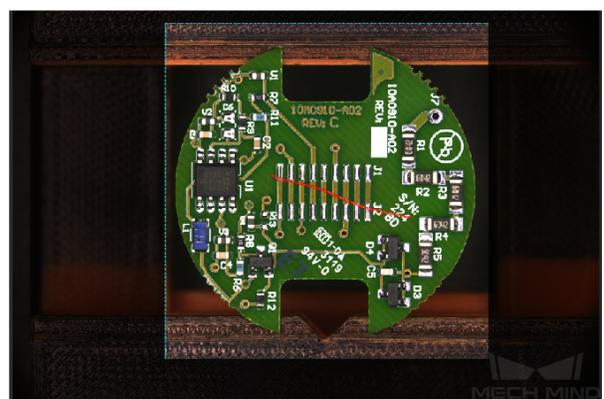
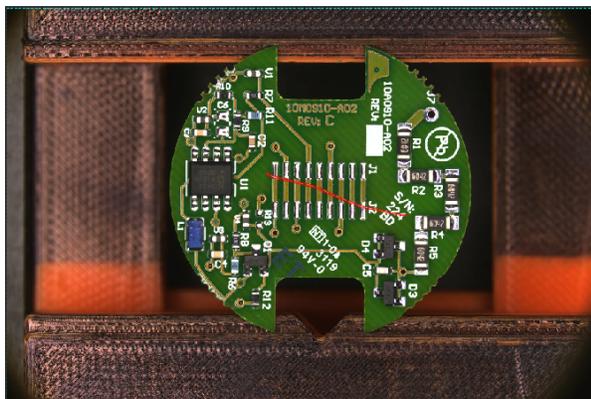


标注提示：无法判断是否属于缺陷，可以使用掩膜多边形工具遮盖缺陷区域。

注意：图像中有多处缺陷时，如果无法判断是否符合缺陷判定标准，可以删除当前图像，避免影响模型训练效果。

5.3.2 设置合理的感兴趣区域 (ROI)

设置 ROI 可以有效排除背景的干扰，ROI 边界应尽量贴近物体外轮廓。



提示：同一 ROI 设定将应用于所有图片，因此需要保证所有图片中的物体都位于 ROI 内，尤其是物体位置/尺寸不固定的场景。

5.3.3 选取合适的训练集

- 训练集数量需可控

对于“缺陷分割”模块初次建模，建议选择 20~30 张图片作为训练集（根据缺陷类别和差异程度可适当调整数量）。数据并不是越多越好，前期大量无效的数据集不利于后期模型迭代，且会增加训练时间。

- 数据具有代表性

训练集一定要涵盖所有需要检测缺陷类型的 NG 图像，包含所有形态、背景、颜色、尺寸等信息。OK 图像的特征差异较小时，可以选择少量的 OK 图。

- 数据占比均衡

训练集中各种缺陷图像数量比例需均衡，禁止一种缺陷图像有 20 张，另一种缺陷图像仅有 5 张的情况。

- 数据集与终端场景保持一致

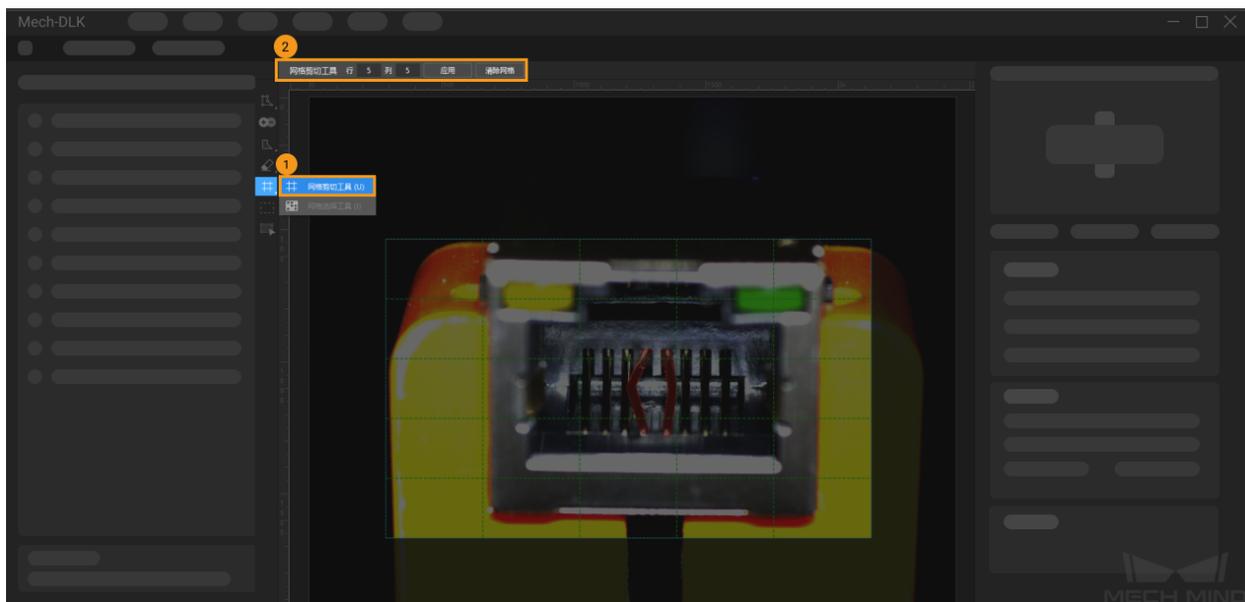
数据集的图像要与最终模型使用的终端场景保持一致，包括光照条件、工件特征、检测背景、视野大小等。

5.4 如何解决困难问题

5.4.1 网格剪切工具

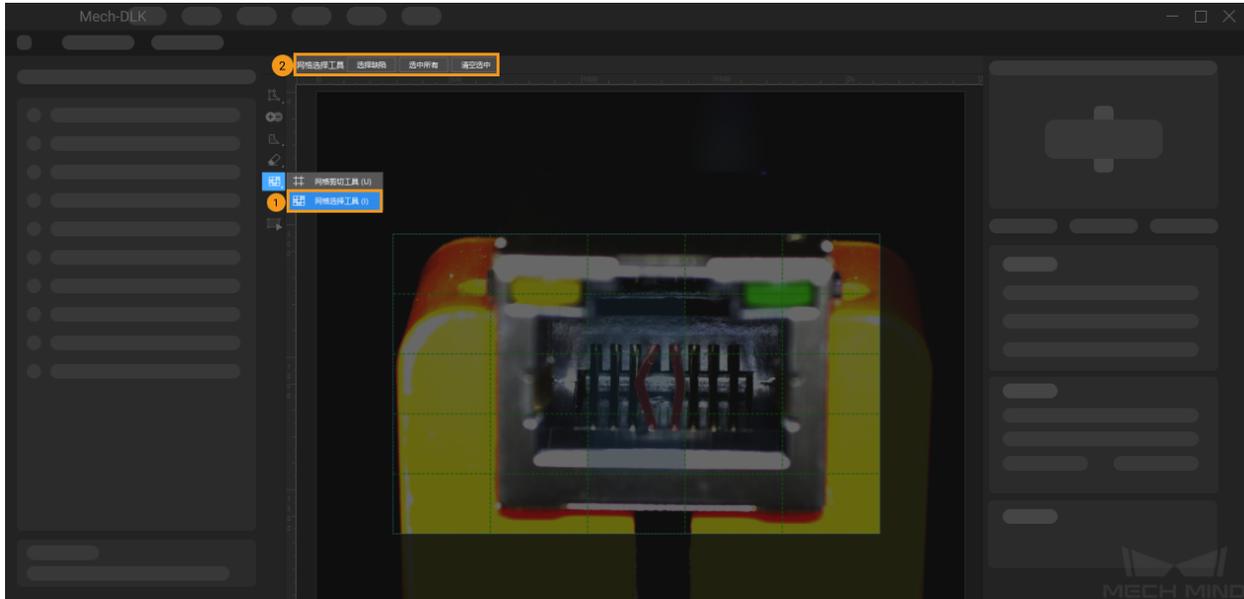
在工业检测场景中，如果相机采集的图像尺寸较大，那么图像中较小的缺陷会不明显。用户可以使用“缺陷分割”模块中的网格剪切工具，将尺寸较大的图像按照设置的比例剪切成大小相同的小图，便于检出较小的缺陷。该功能包含以下两个工具：

1. **网格剪切工具**：将较大的图像网格化处理，用户根据实际情况设置网格行列数后，点击 **应用**。



注意：行数和列数不宜设置过大，否则裁切后的小图数量较多，会导致推理时间变慢。

2. **网格选择工具：**网格单元左上角为勾选按钮，可根据需求选择有缺陷的和无缺陷的小图加到训练集/验证集中。如需预览剪切后的小图，可参考裁切后小图缩放预览。

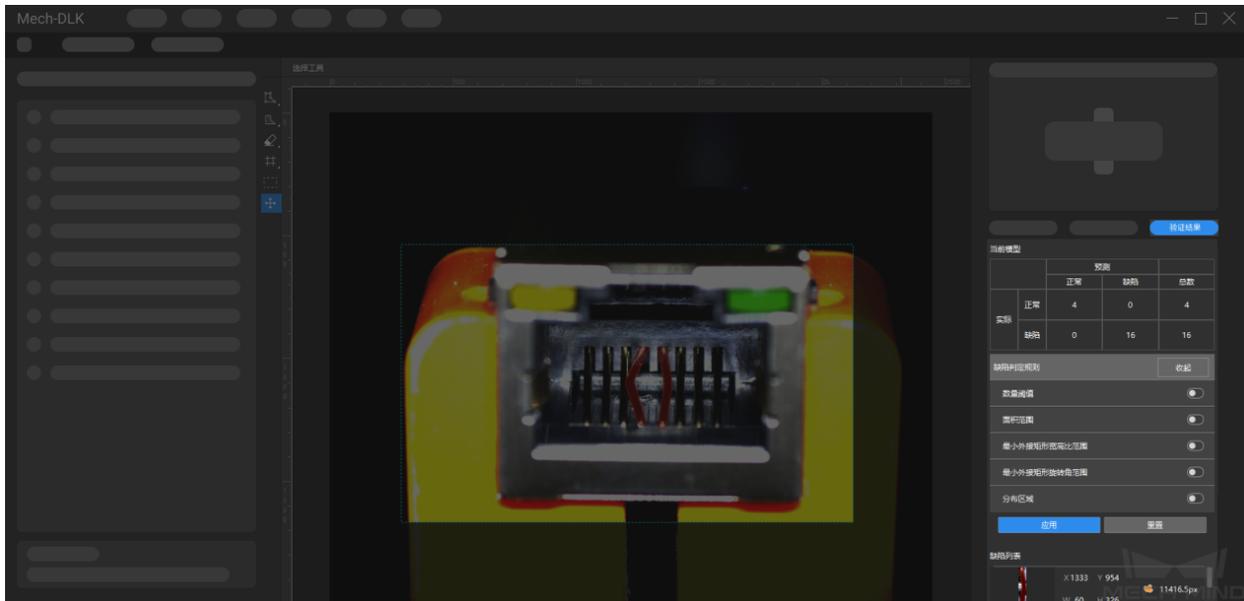


- 选择缺陷，将剪切后所有包含缺陷的小图全部选入到训练集/验证集中。
- 选中所有，将剪切后的全部小图都选入训练集/验证集中，并且将没有缺陷的小图设置为 OK。
- 清空选中，清空当前图像选择。

注意：如果用户剪切图像后没有使用网格选择工具，会默认把 NG 图像中所有包含缺陷的小图和 OK 图像剪切的所有小图选为训练集/验证集，可能导致选入训练集/验证集的相似的 OK 图像小图过多，影响训练及优化。

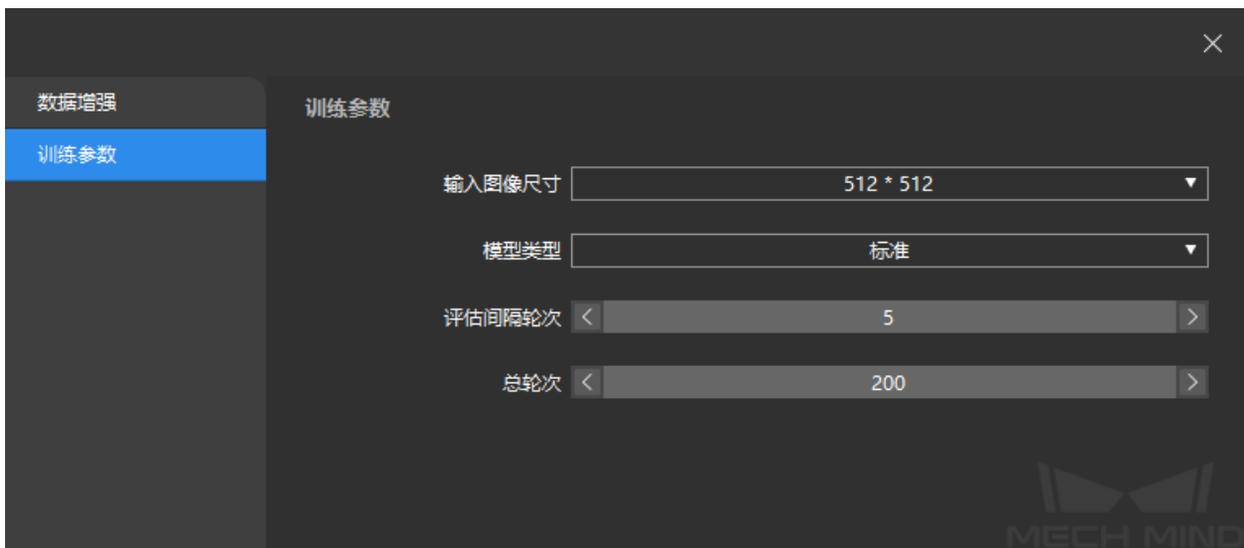
5.4.2 缺陷筛选工具

在缺陷分割场景中，如需参照用户定义的缺陷定义标准进一步判定缺陷，或希望根据良品率重新调整缺陷占比，可以通过调整数量阈值、面积范围、最小外接矩形旋转角范围、分布区域等参数进行筛选。



5.4.3 调整训练参数

使用模型过程中，如果有需要加速推理或提高模型精度等需求时，可以通过调整训练参数重新训练模型来优化效果。



输入图像尺寸 训练时输入神经网络的图像宽高（单位：像素）。建议使用默认设定；若图像中的物体或标注缺陷区域较小时，需要适当调大输入图像尺寸。图像尺寸越大，模型精度越高，但是训练速度越低。

模型类型

- **标准**：一般情况下建议使用标准模式。
- **增强**：模型效果不好或精度要求较高时可以选择增强模式，该模式会导致训练速度变慢。

评估间隔轮次 模型训练时，每评估一次间隔的轮次数量。建议使用默认设定，调大**间隔轮次**参数可以提升训练速度，参数值越大训练越快；参数值越小训练越慢，但有助于挑选最优模型。

总轮次 模型训练的总轮次。建议使用默认设定，若待识别图像特征较复杂时，需要适当增加训练轮次，提升模型效果，但会导致训练时间变长。

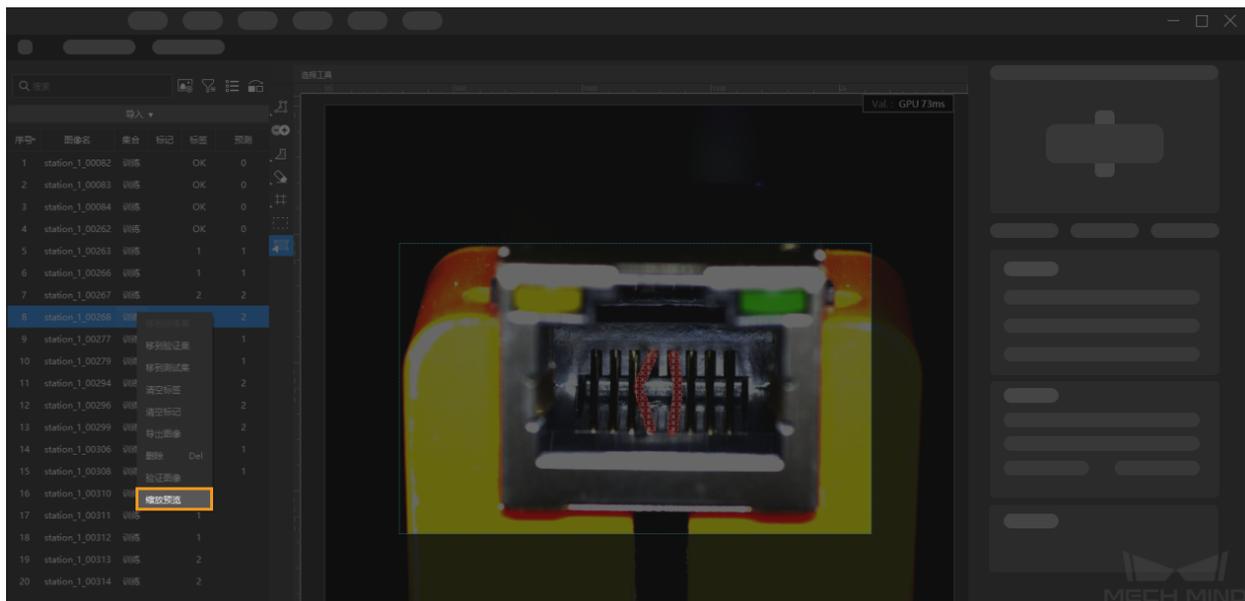
注意：训练轮次并非越大越好。总轮次设置较大时，模型准确率稳定后会持续训练，这种情况会导致训练时间变长，且有过拟合的风险。

5.4.4 缩放预览

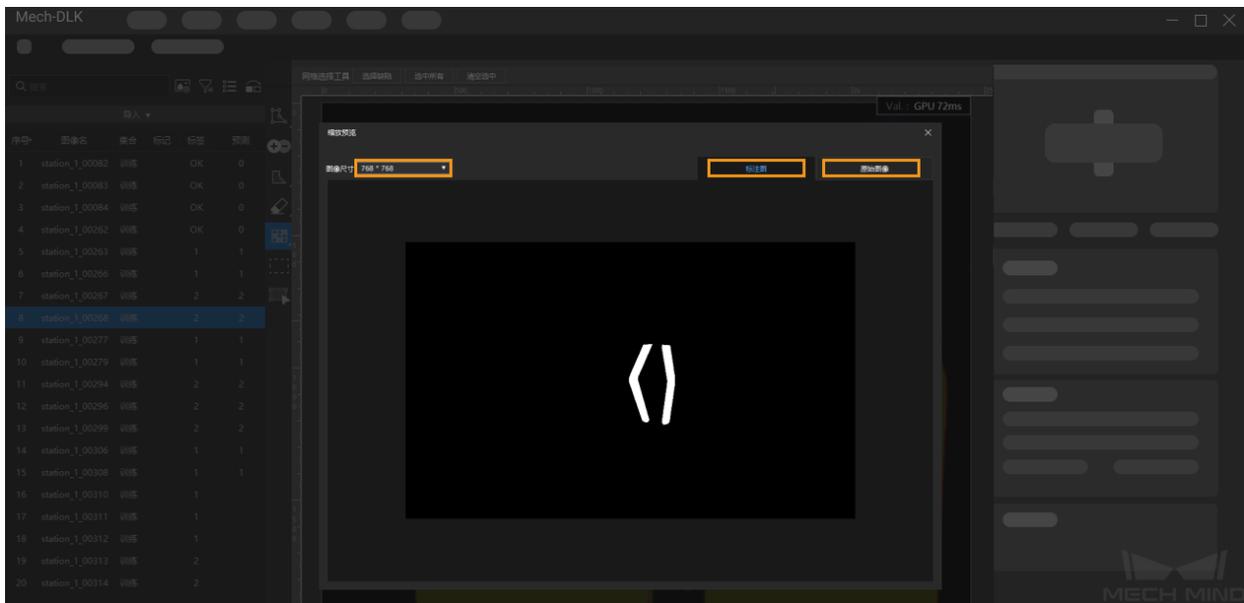
使用“缺陷分割”模块训练模型时，若图像缺陷较小，可能会影响训练效果，需要选择合适的图像尺寸。

单张图像

在数据栏右键点击需要预览的图像，点击**缩放预览**，会弹窗显示缩放预览后的图像。



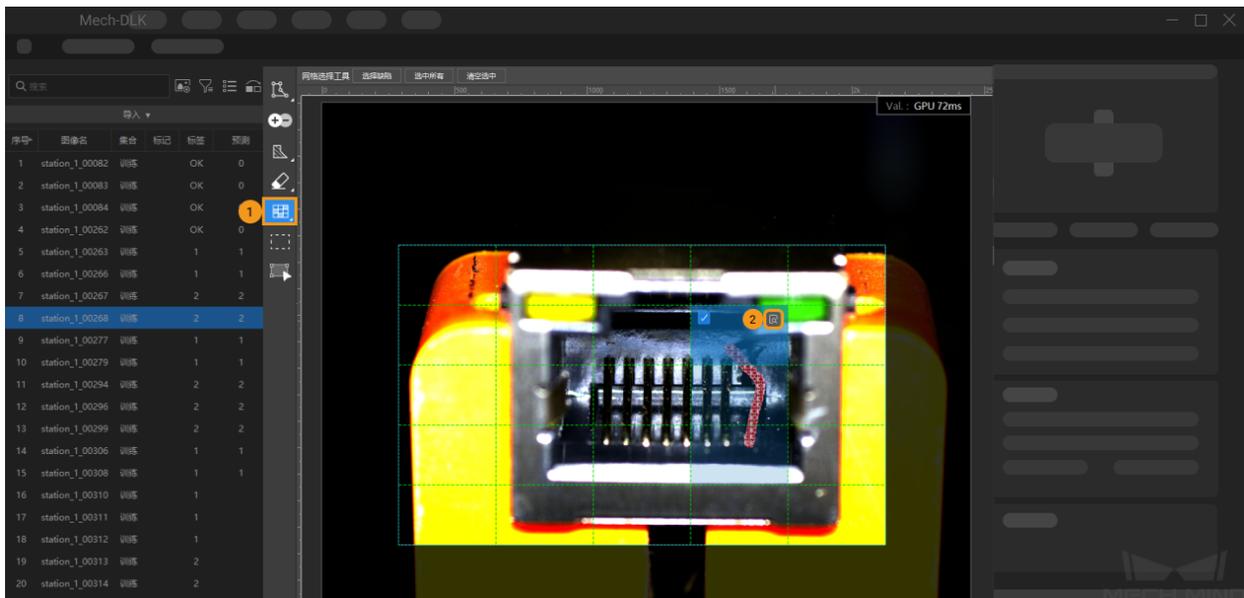
弹窗中可调整图像尺寸。用户可以点击**标注图**和**原图**查看效果。



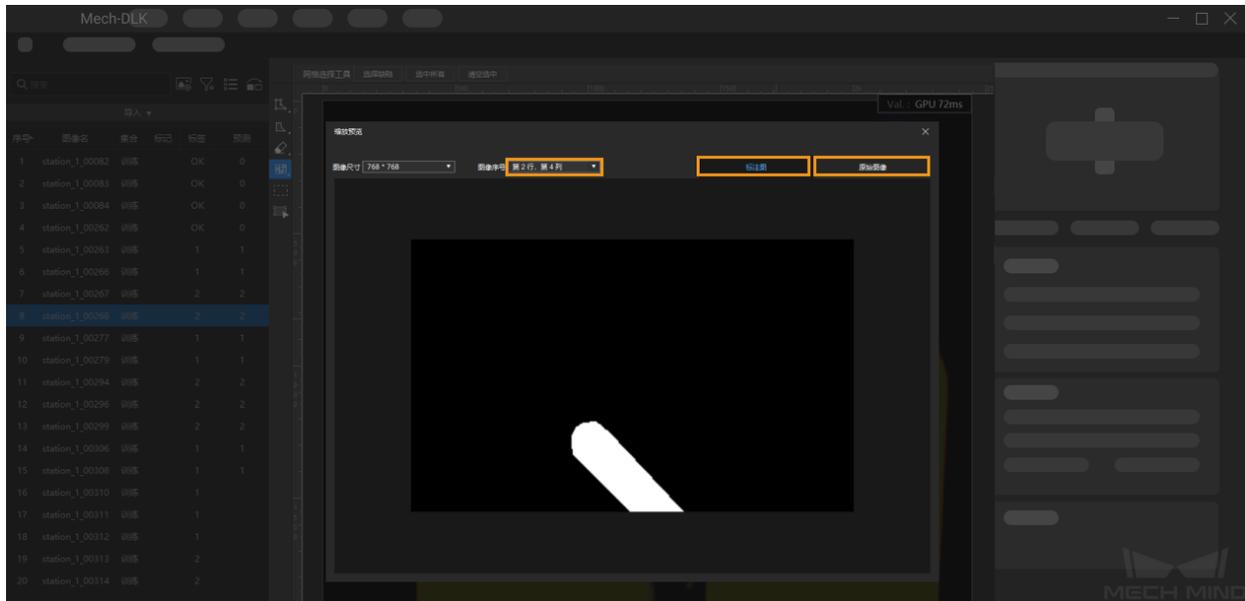
注意： 在缩放预览界面设置的图像尺寸不生效，实际用于训练的图像尺寸需在训练参数中设置。

图像裁切后缩放预览

点击 网格选择工具，鼠标放到需要预览的小图上，点击小图右上角的预览，会弹窗显示预览后的效果图。



调整 图像序号可以查看其他小图的预览效果；用户可以点击 标注图和 原图查看效果。



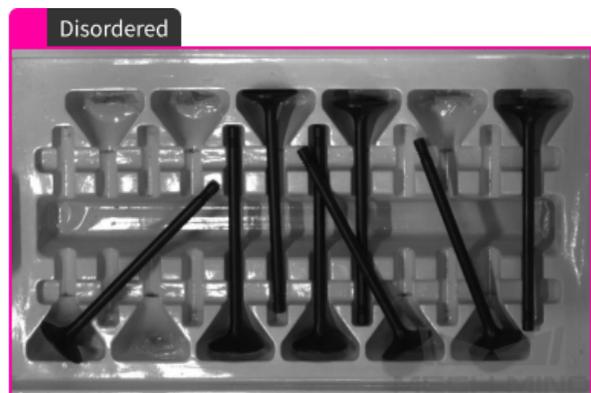
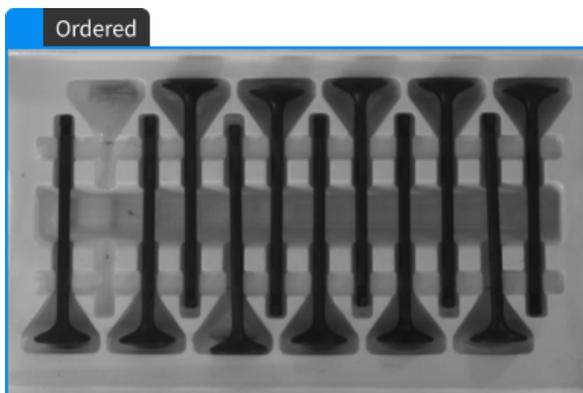
小技巧: 默认可以预览网格剪切后选中的图像，如需预览其它图像，需要选择 **选中所有** 后再点击小图右上角进行预览。

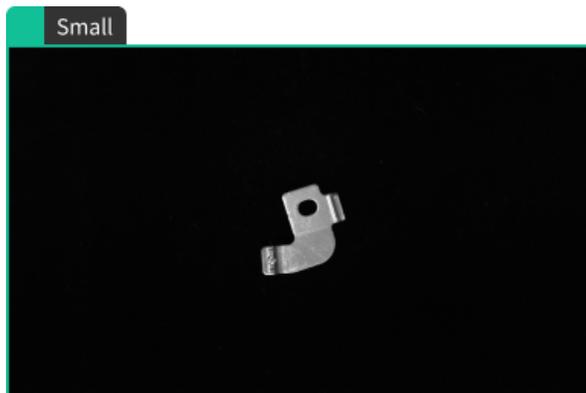
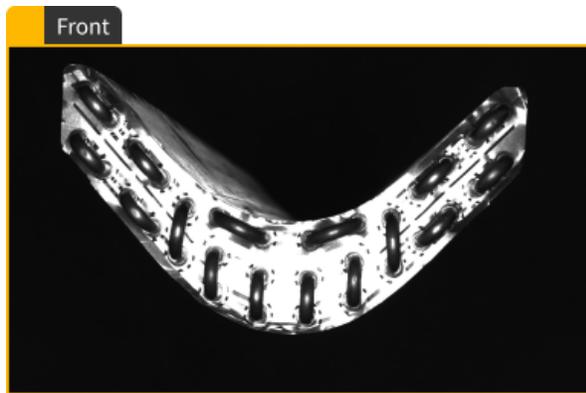
6.1 算法介绍

图像分类可以判断图像类别。

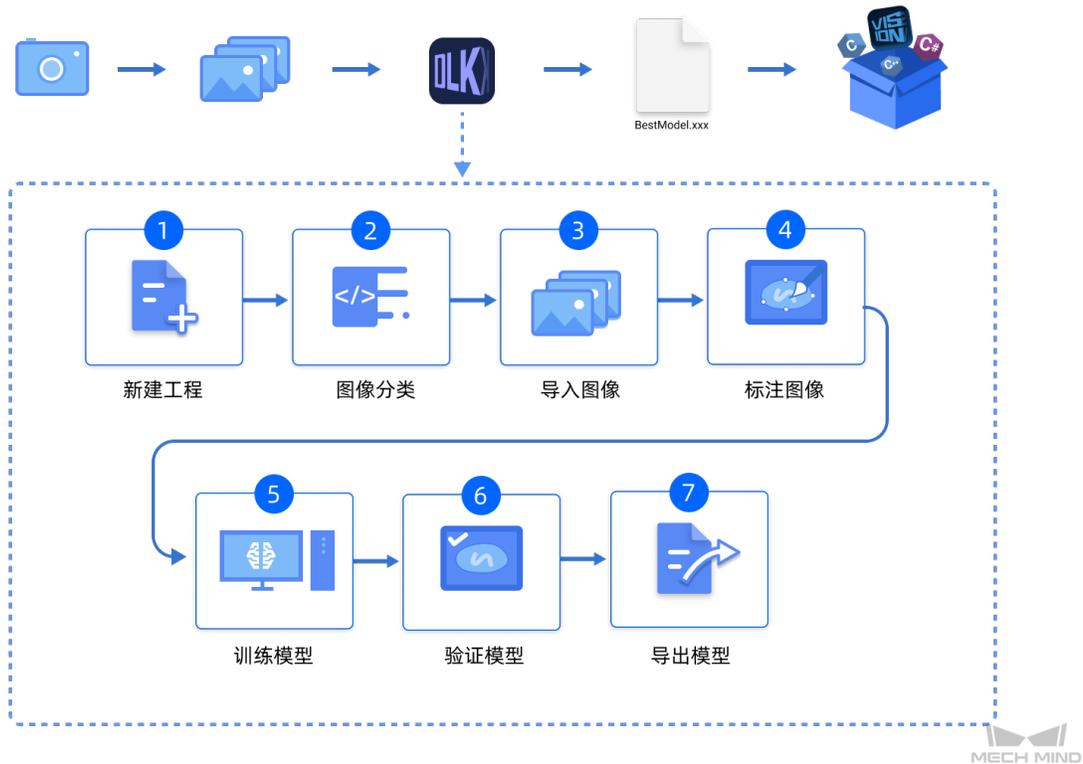
6.1.1 应用场景

工件上下料：适用于钢铁、机械等行业中，区分各类零部件正反面、朝向、种类等信息。





6.1.2 应用流程



6.1.3 应用要点

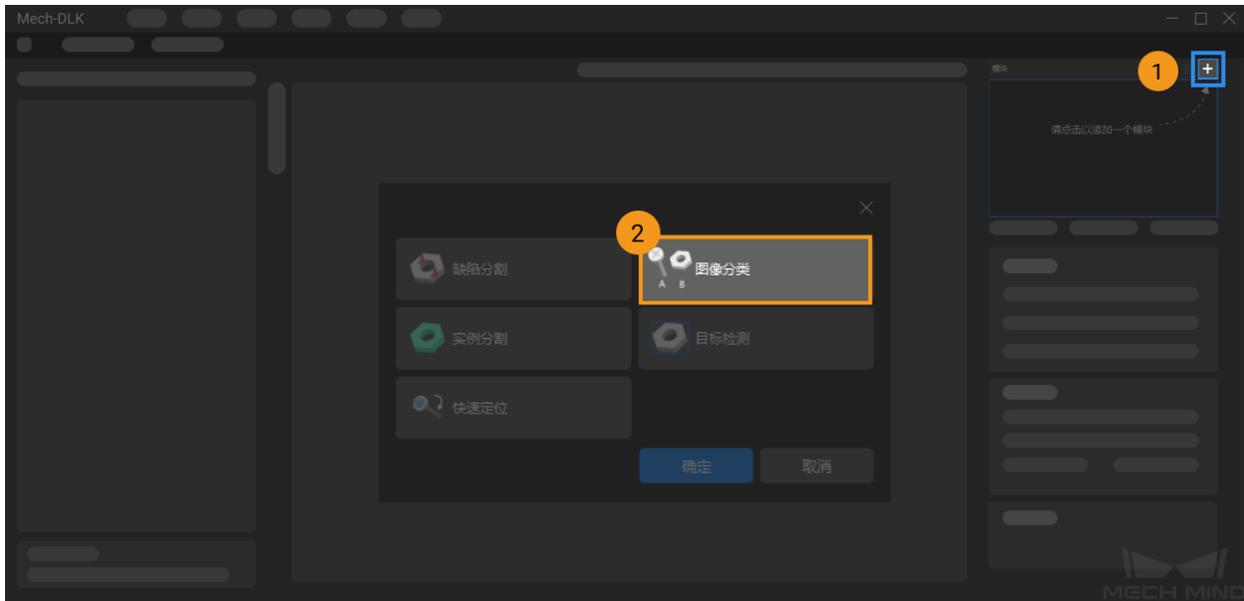
以下几点有助于提高模型质量：

1. 确保图像质量[前往章节 >](#)。
2. 确保数据集质量[前往章节 >](#)。
3. 确保标注质量[前往章节 >](#)。

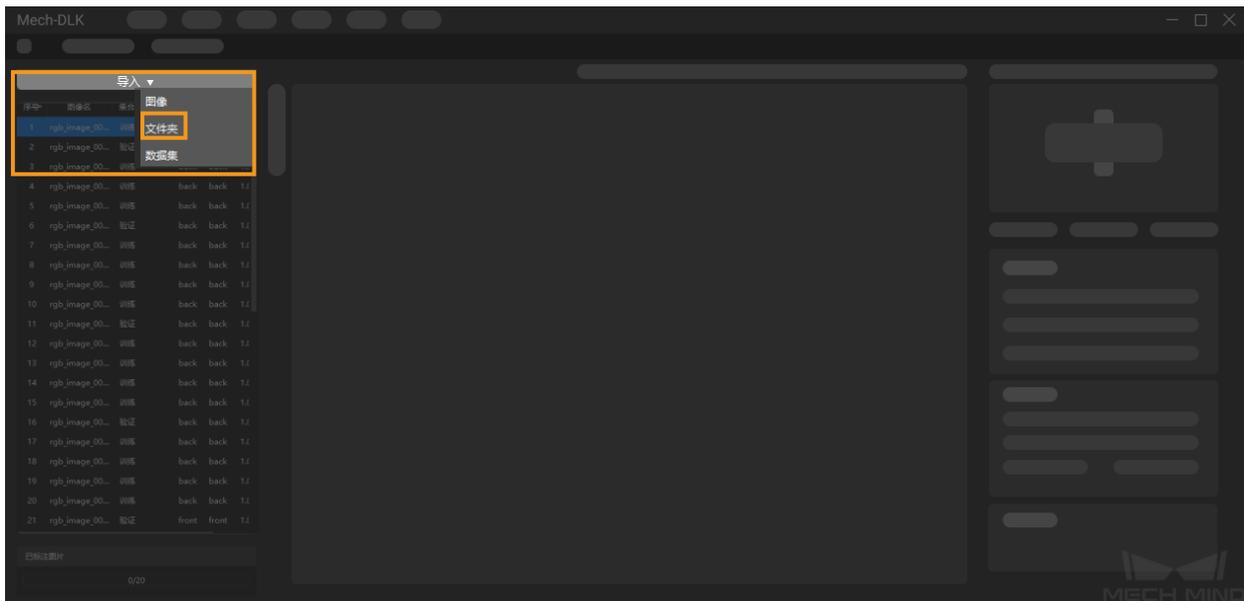
6.2 使用图像分类模块

本章提供 Mech-DLK 示例工程中的冷凝管数据集（[点击下载](#)），带领用户使用“图像分类”模块训练模型，达到示例工程中区分工件的正反面的效果。

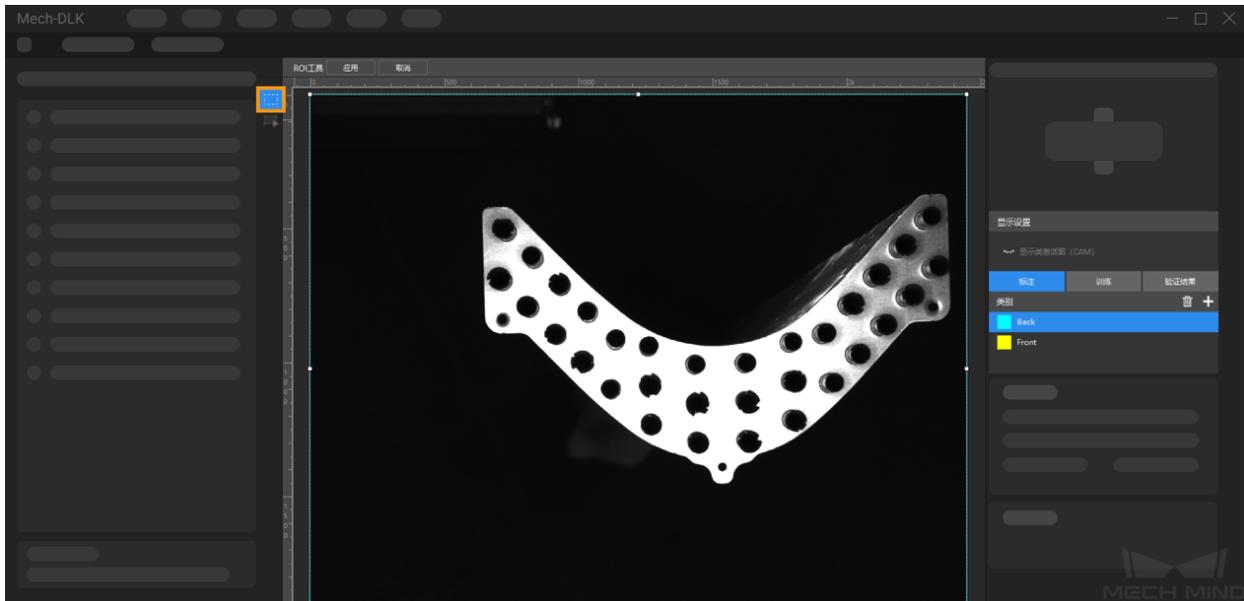
1. 新建工程并添加“图像分类”模块：点击初始页面上的 新建工程，选择工程路径并输入工程名以新建一个工程。再点击右上角的 ，选择“图像分类”模块。



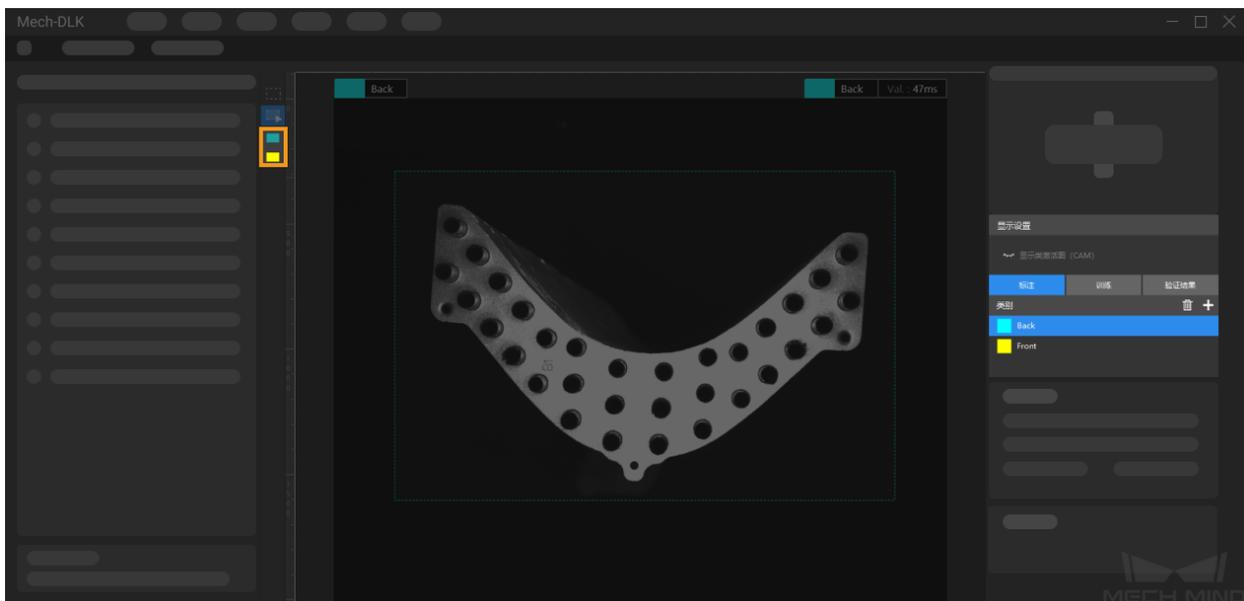
2. 导入工件正反面图像数据集：解压缩下载的数据集压缩包，点击左上方的 导入，选择 文件夹导入下载好的图像数据集。



3. 截取 ROI：点击窗口左侧的 ROI 框选选项从图像中框选整个冷凝管作为感兴趣区域，并点击左上方的应用确认使用。截取 ROI 的目的是减少无关背景信息的干扰。

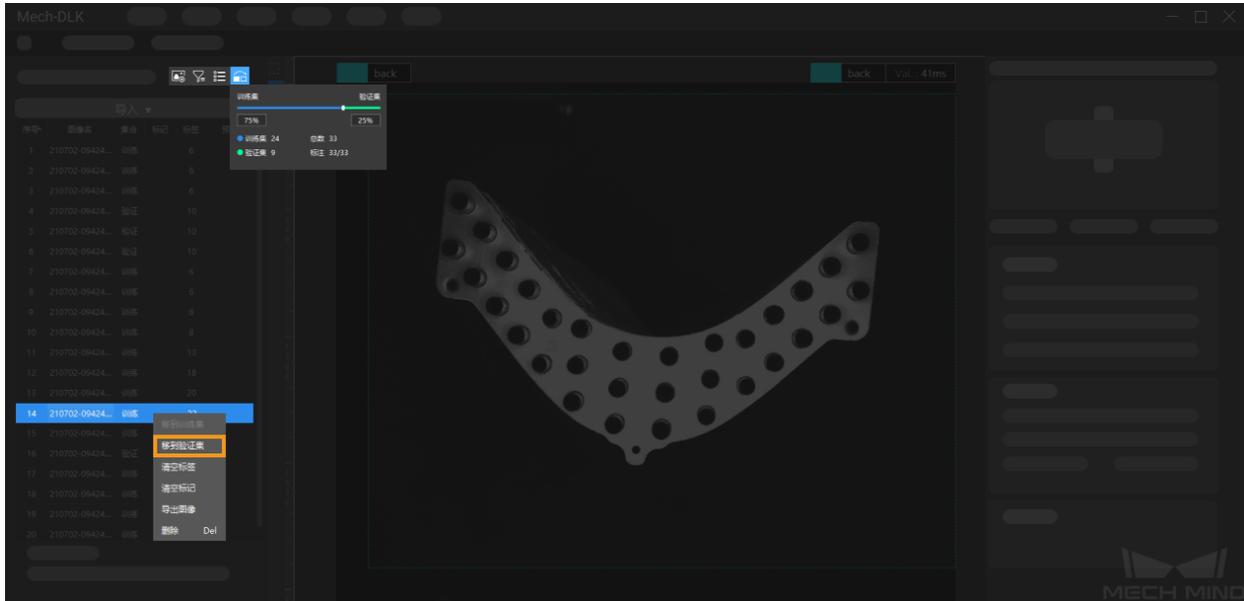


4. **创建标签：**根据物体名称或特征创建对应的标签，此处需要区分工件正反面，所以直接以正反命名。
5. **标注图像：**图像左侧标注工具栏选择对应的标签进行标注，可以同时选中多张图像进行标注。务必确保标签的一致性，禁止出现将图像反面标注为正面的情况。

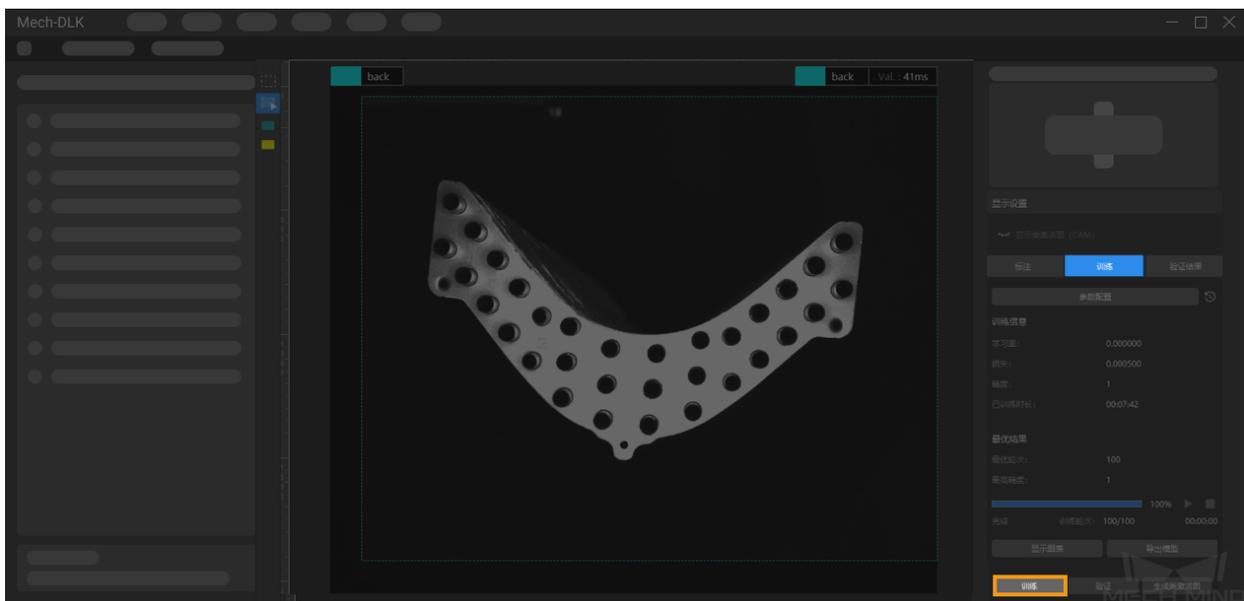


小技巧：“图像分类”模块支持同时选中多张图像进行批量标注。

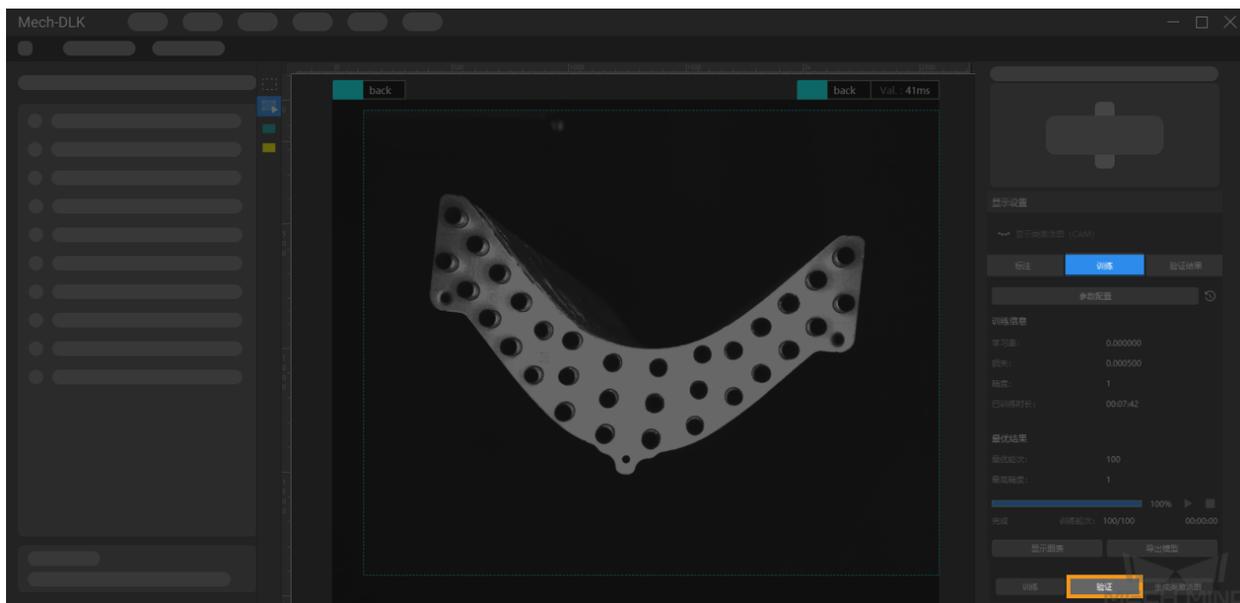
6. **划分训练集与验证集**：软件默认将数据集的 80% 划分为训练集，20% 为验证集。需要确保训练集和验证集中都包含 **所有种类** 的图像，目的是让算法模块在训练过程中可以学习所有种类图像的特征，并对所有种类的图像进行验证。如果默认划分的数据集不满足这一条件，可以点击  拖动滑块手动调整。



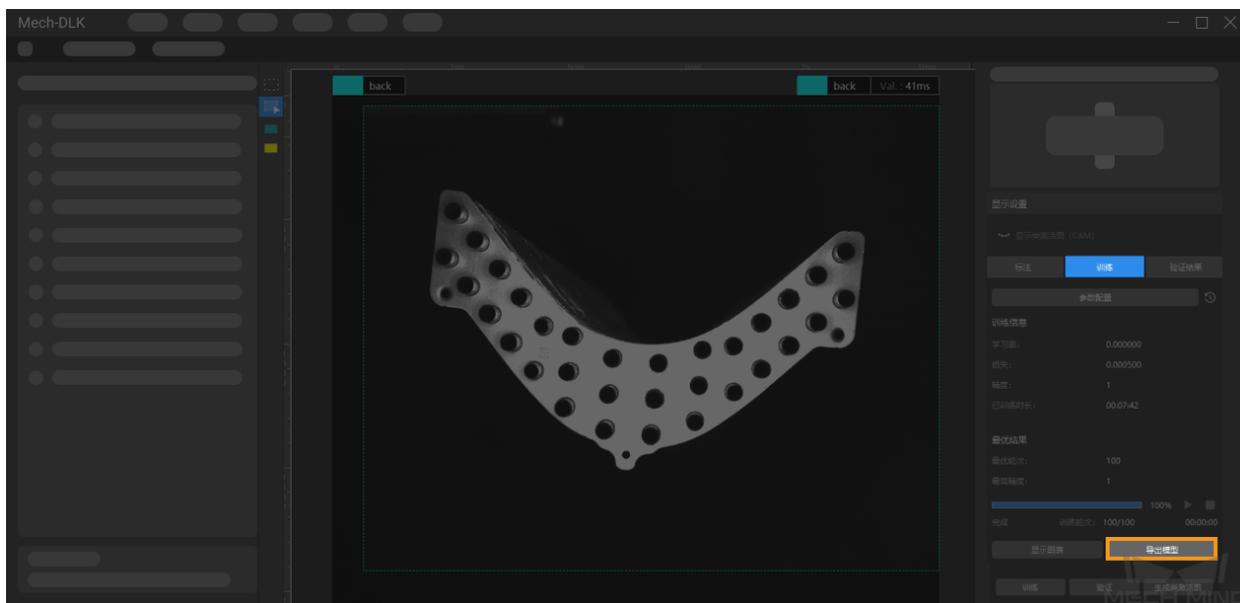
7. **训练模型**：使用默认参数设定，点击 **训练** 开始训练模型。



8. 验证模型：训练结束后，点击 验证可以验证并查看模型识别效果。



9. 导出模型：点击 导出模型，选择存放路径即可导出模型 model.dlcpack 到指定文件夹，用户可以根据需求部署模型。



6.3 如何训练高质量模型

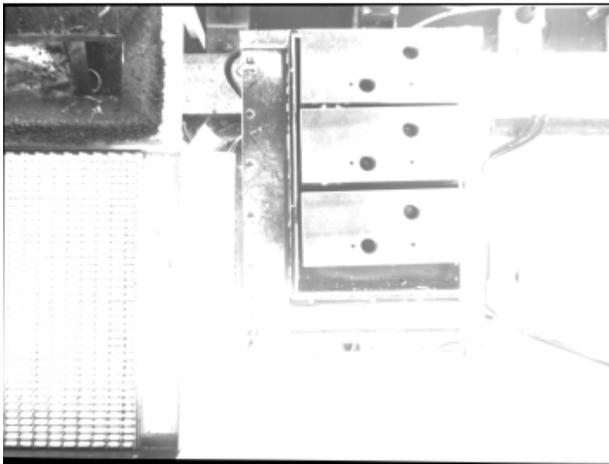
本章将介绍最影响模型质量的几个因素，以及如何训练出高质量图像分类模型。

- 确保图像质量
- 确保数据集质量
- 确保标注质量
- 类激活图显示

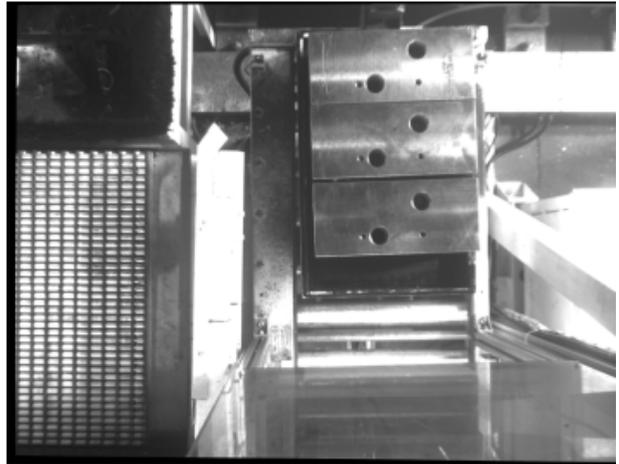
6.3.1 确保图像质量

1. 避免 过曝、过暗、颜色失真、模糊、遮挡等情况。这些情况会导致深度学习模型所依赖的特征丢失，影响模型训练效果。

错误示例：过曝



正确示例：正常



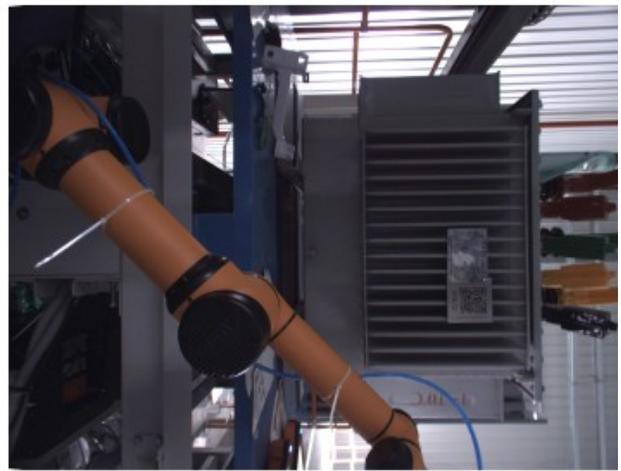
优化建议：可通过遮光等方式避免

错误示例：过暗



优化建议：可通过补光等方式避免

正确示例：正常



错误示例：颜色失真

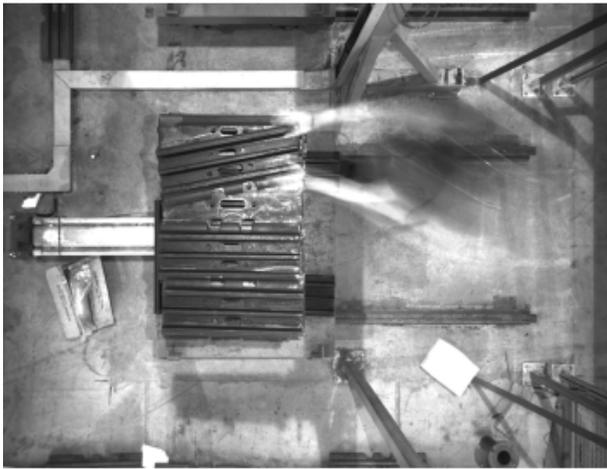


优化建议：可通过调整相机白平衡避免

正确示例：正常



错误示例：模糊



优化建议：避免相机或物体仍在运动时拍照

正确示例：正常



错误示例：机械臂遮挡



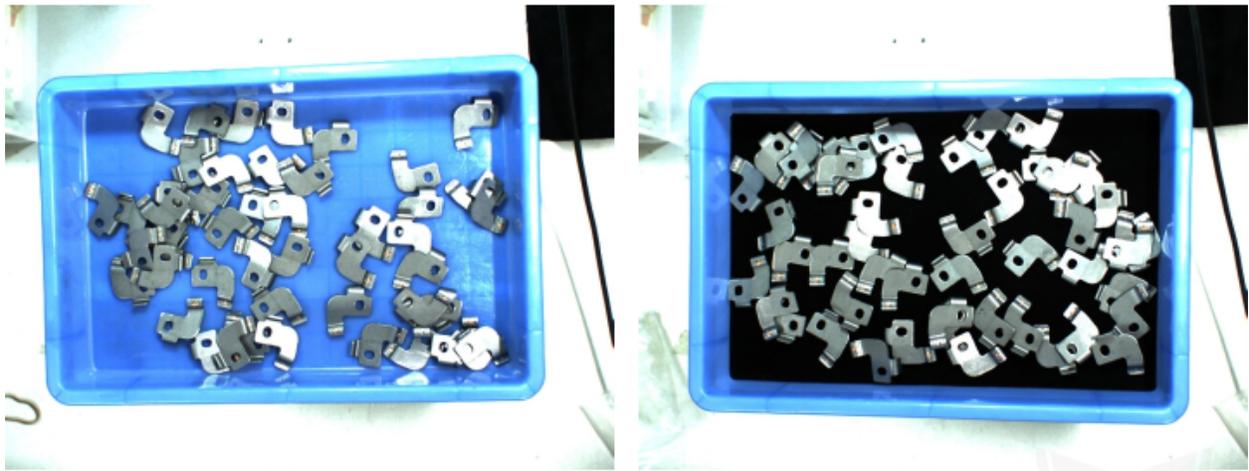
优化建议：确保机械臂和人离开拍照区

错误示例：人手遮挡



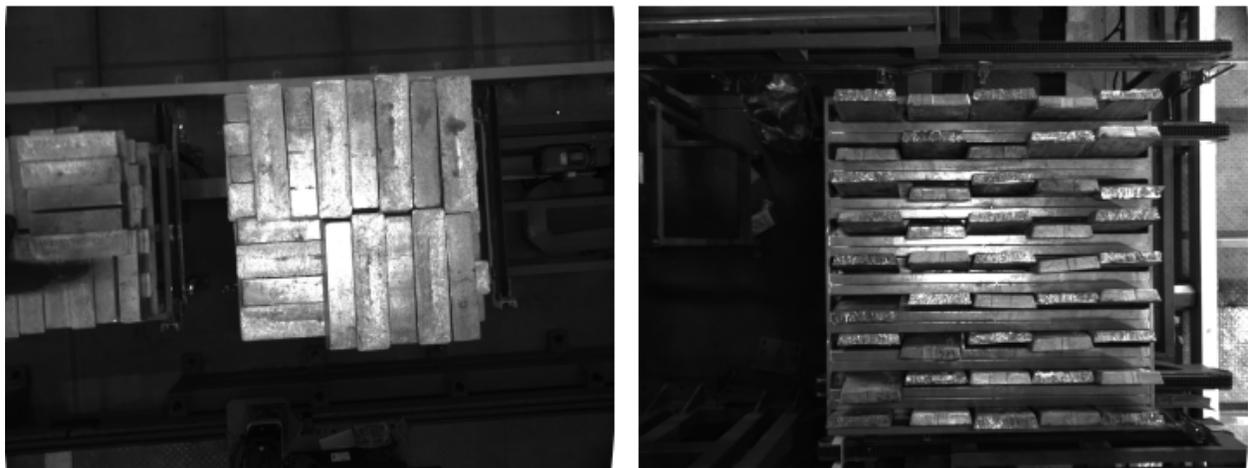
2. 确保采集数据的背景、视角、高度与实际应用一致。任何不一致都会降低深度学习在实际应用时的效果，严重情况下必须返工重新采集数据，请务必提前确认实际应用时的情况。

错误示例：训练数据背景（图左）与实际应用背景（图右）不一致



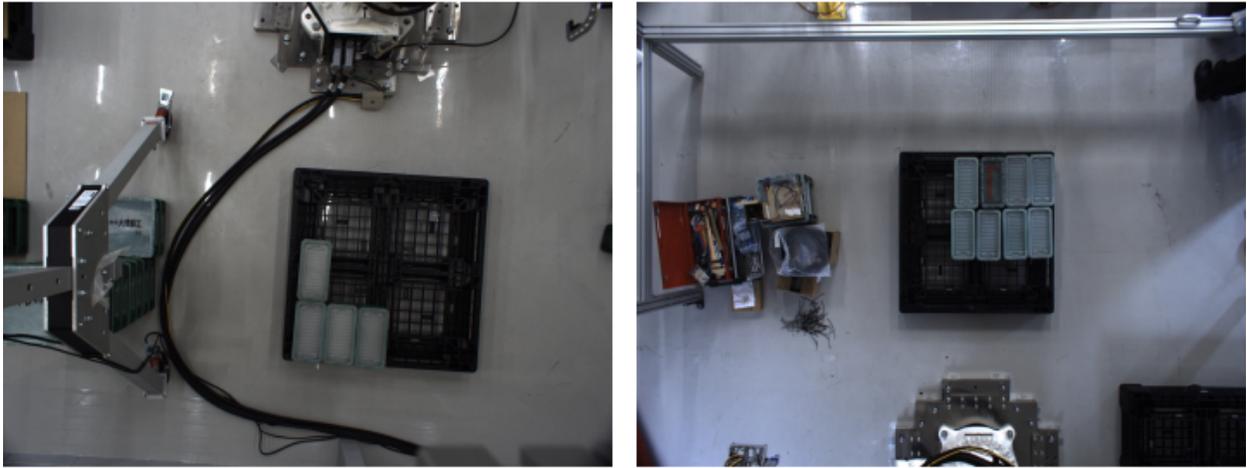
优化建议：保持训练数据背景与实际应用一致

错误示例：训练数据视野（图左）与实际应用视野（图右）不一致



优化建议：保持训练数据采集视野与实际应用一致

错误示例：训练数据高度（图左）与实际应用高度（图右）不一致



优化建议：保持训练数据采集高度与实际应用一致

注意：图像分类对光照比较敏感，采集时光照条件需保持一致；如若现场早晚光线不一致，需针对情况分别采集。

6.3.2 确保数据集质量

“图像分类”模块通过学习已有图像的特征得出模型并应用到实际场景，因此采集和选取的数据集必须与实际场景情况一致才能训练出高质量模型。

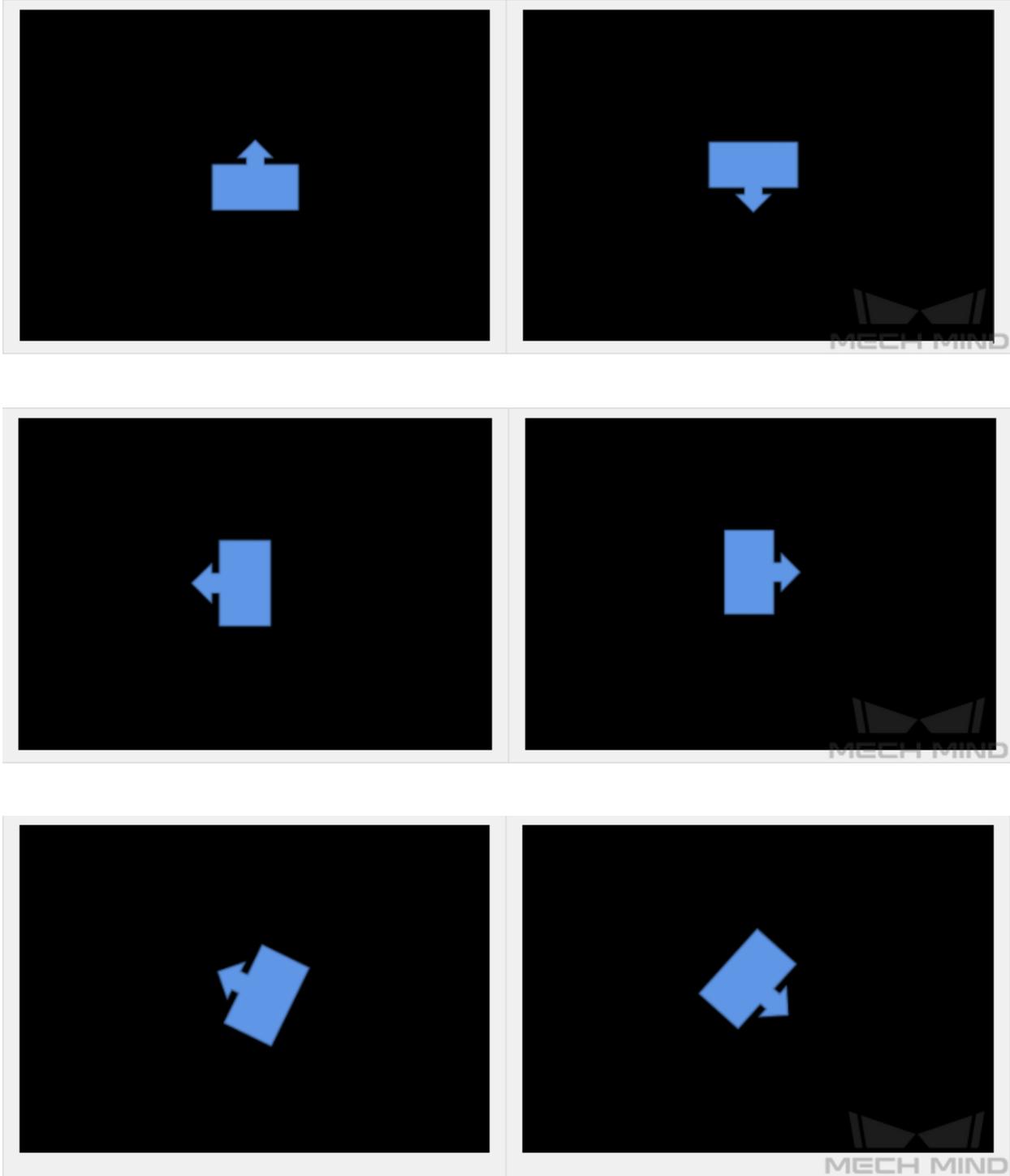
1. 采集数据集
2. 选取数据集

采集数据集

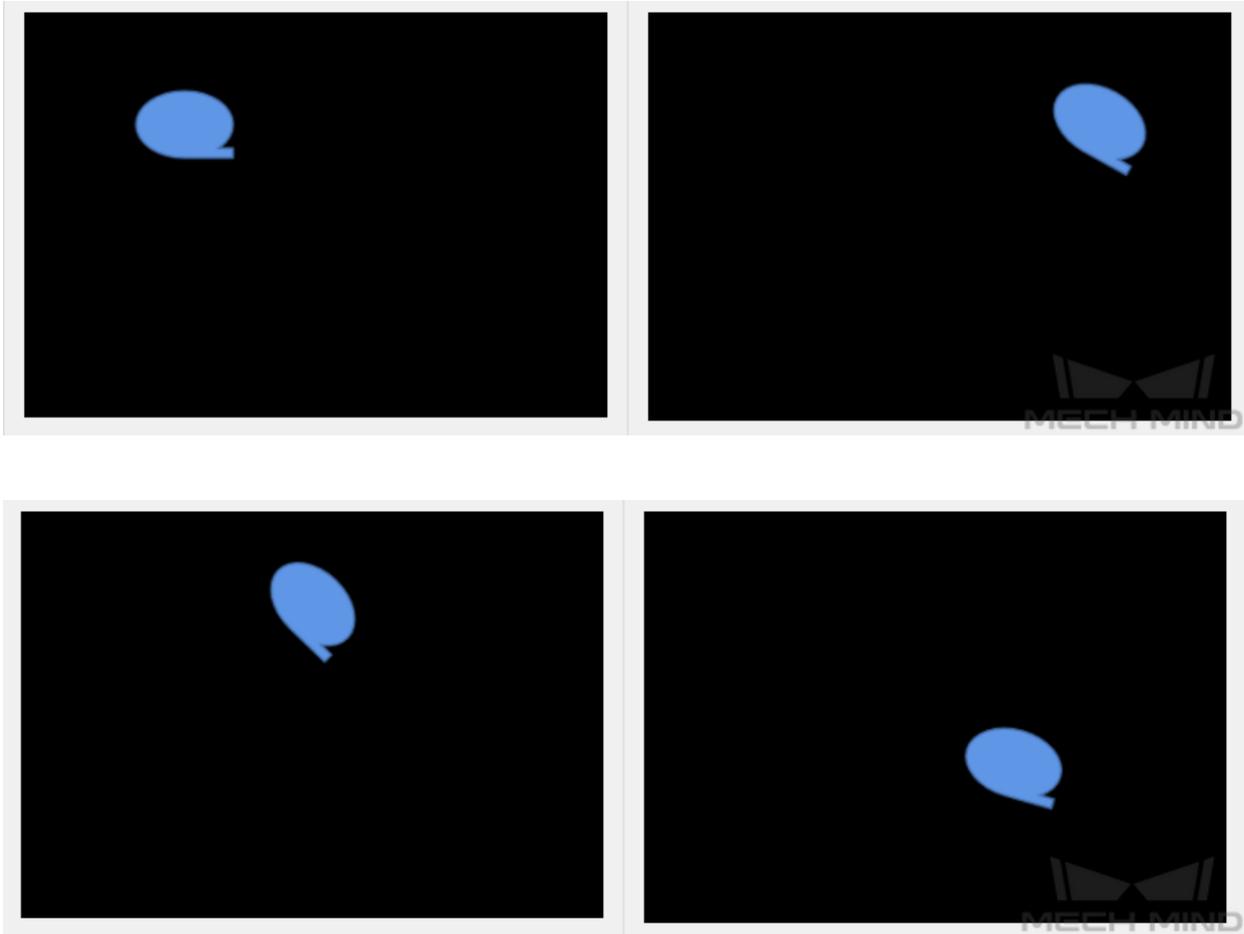
需合理分配各种摆放情况。例如实际生产时来料有横向和竖向，但只采集横向来料的数据进行训练，则无法保证竖向来料的分类效果。因此，采集数据时需要能包含实际生产的各种场景，具体包括：

- 实际应用可能出现的待分类物体 **不同角度**的特征。
- 实际应用可能出现的待分类物体 **不同位置**的特征。

1. 各个角度

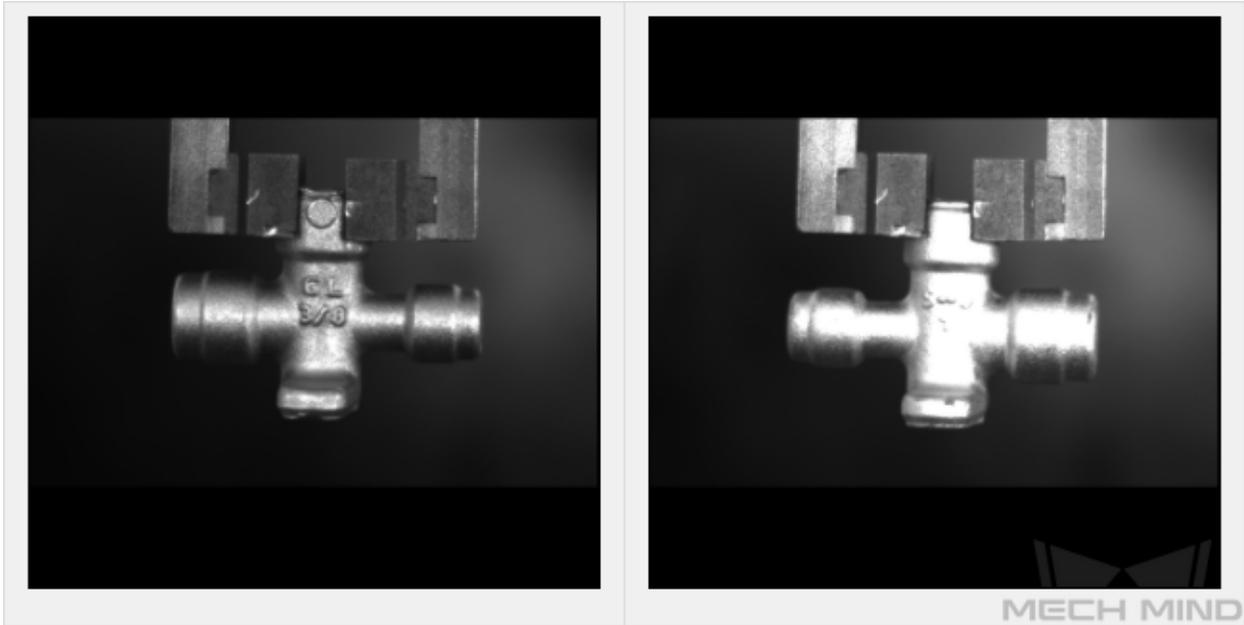


2. 不同位置

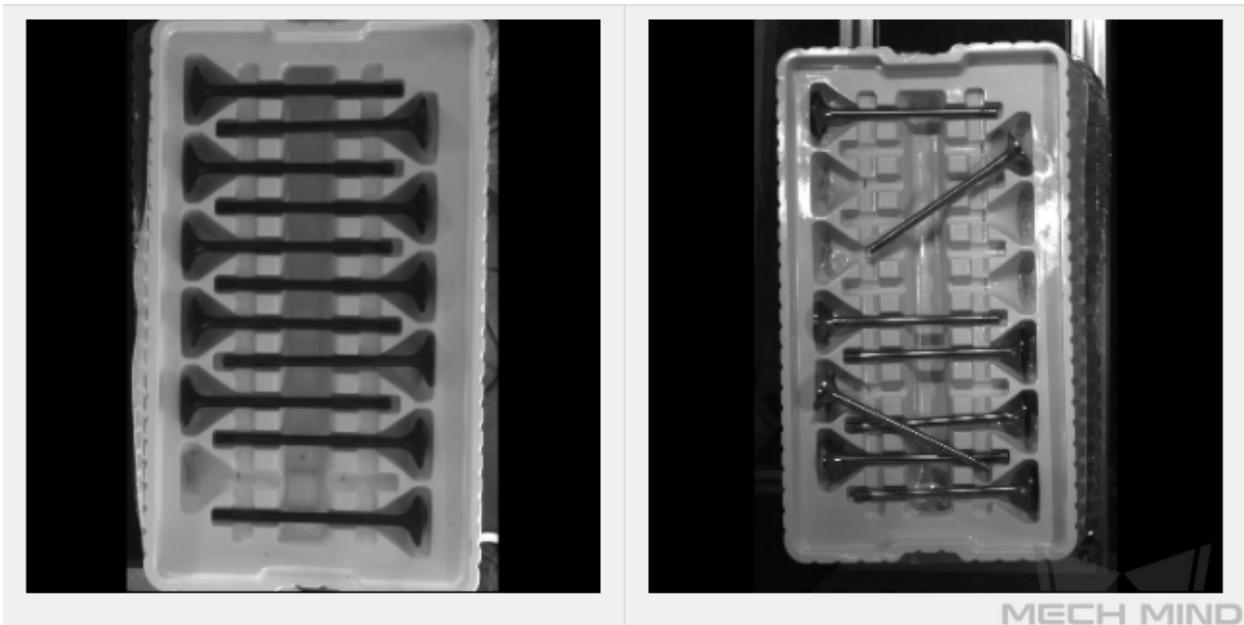


采集数据案例

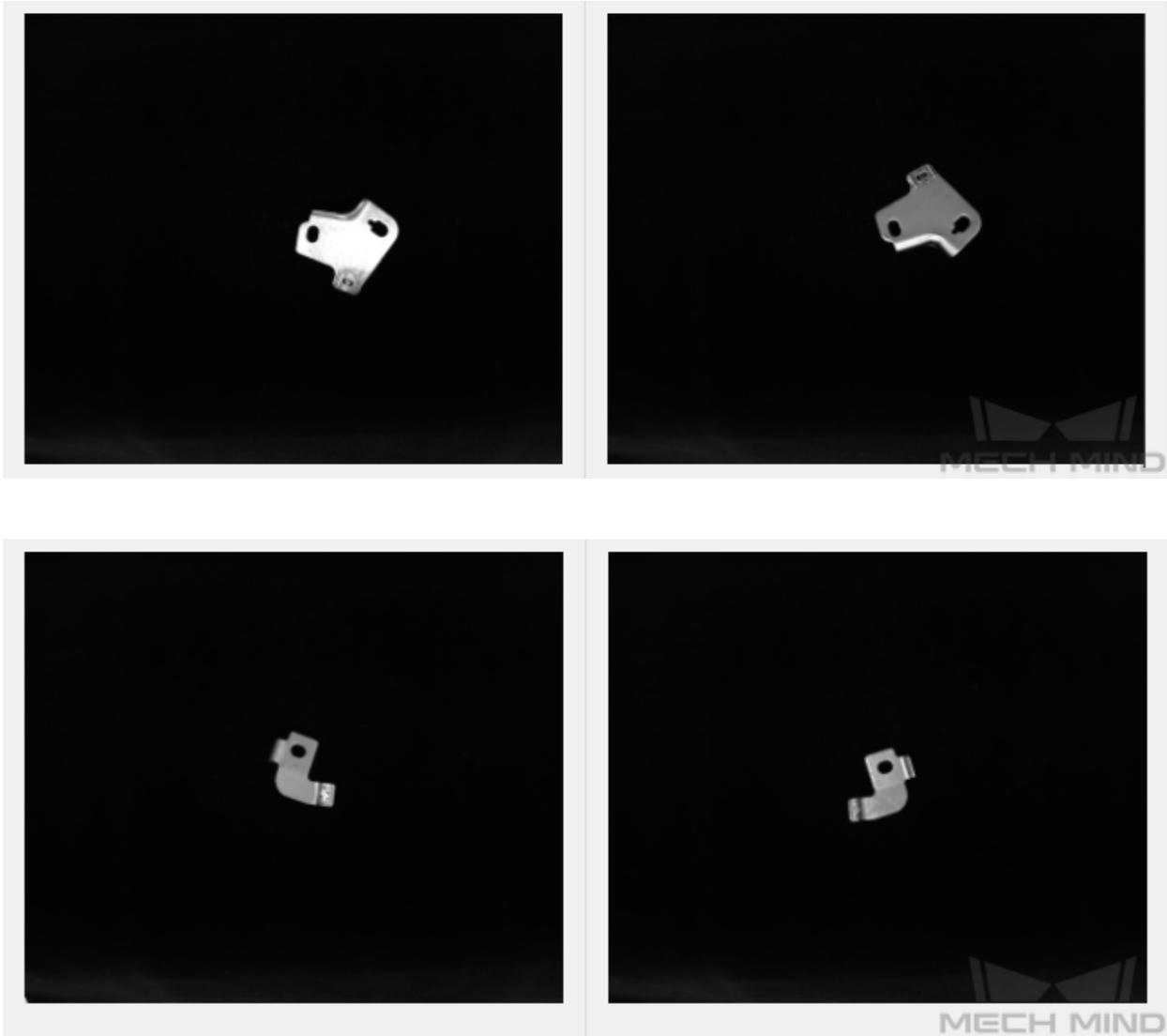
1. 某阀管项目，单类物体，需要区分阀管正反面，位置移动幅度较小，正反面各采集 15 张。



2. 某发动机气门装配项目，单类物体，需要区分工件是否正确摆放在卡槽内，由于摆放在卡槽外可能出现的姿态较多，需考虑不同位置 and 不同角度摆放的情况，采集 20 张左右；摆放在卡槽内的只需考虑不同位置的情况，采集 10 张左右。



3. 某金属薄片项目，两类物体，需要区分工件大小，可能出现不同位置不同角度的情况，正反面各采集 20 张。



选取合适的数据集

1. 训练集数量可控

对于“图像分类”模块初次建模，建议使用 30 张图像。数据量并不是越大越好，前期加入大量无效的数据集不利于后期的模型改进，同时还会延长训练时间。

2. 数据具有代表性

数据集图像一定要涵盖待检测目标的所有光照、颜色、尺寸等信息。

- 光照：实际存在光照变化，数据集应该包含不用光照情况下的图像。
- 颜色：工件存在不同颜色，数据集应该包含所有颜色的图像。

- 尺寸：工件存在不同尺寸，数据集应该包含所有不同大小尺寸的图像。

注意：若实际现场工件会出现旋转、缩放、或其他情况，无法采集相应图像数据集时，可以通过调整数据增强训练参数的方式来补充数据集，以确保现场所有的情况都包含在训练集内。

3. 数据占比均衡

训练集中不同种类的图像比例要均衡，否则会影响模型效果。禁止出现一种物体 20 张，另一种物体仅有 3 张的情况。

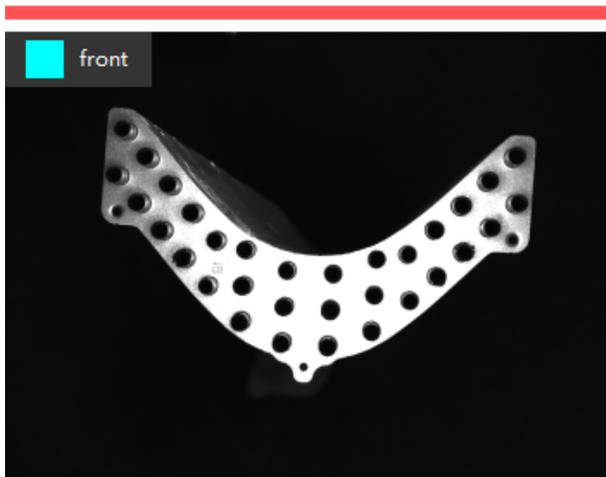
4. 数据集与终端场景保持一致

数据集的图像要与最终模型使用的终端场景保持一致，包括光照条件、工件特征、检测背景、视野大小等。

6.3.3 确保标注质量

- 确保一致性，即图像与类别标签相对应。

错误示例：标签错误



正确示例：正确标注

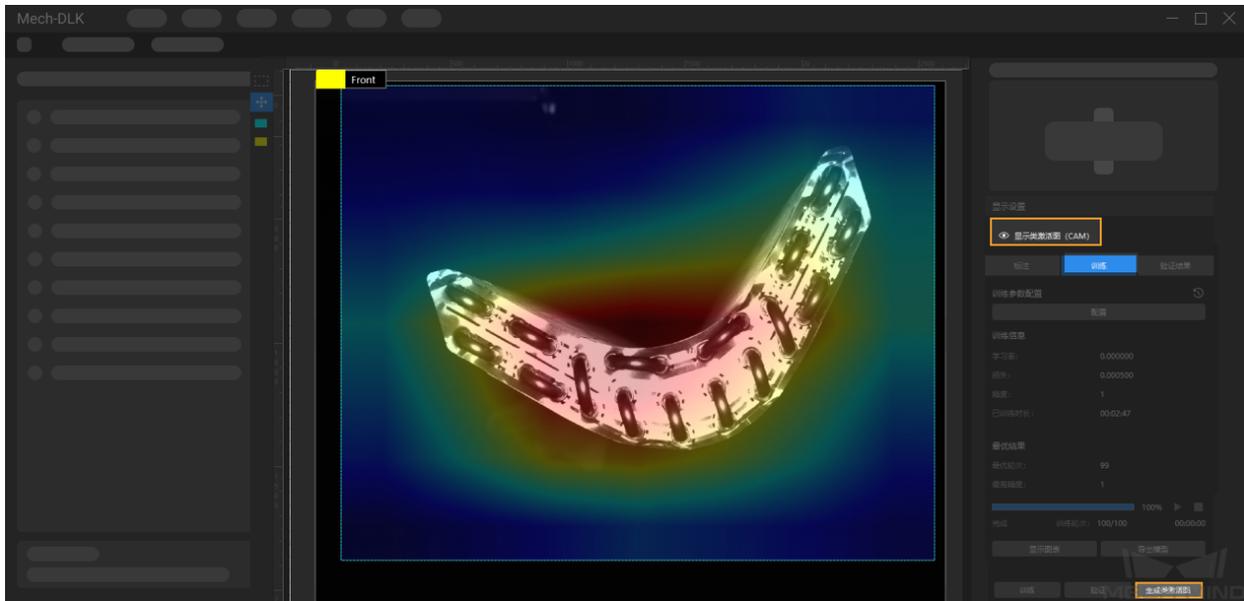


标注提示：标签与图像不一致，右图为工件正面(front)，左图为工件反面(back)。

6.3.4 类激活图显示

图像分类模型训练完成后，点击生成类激活图生成类激活图，点击显示类激活图即可查看。

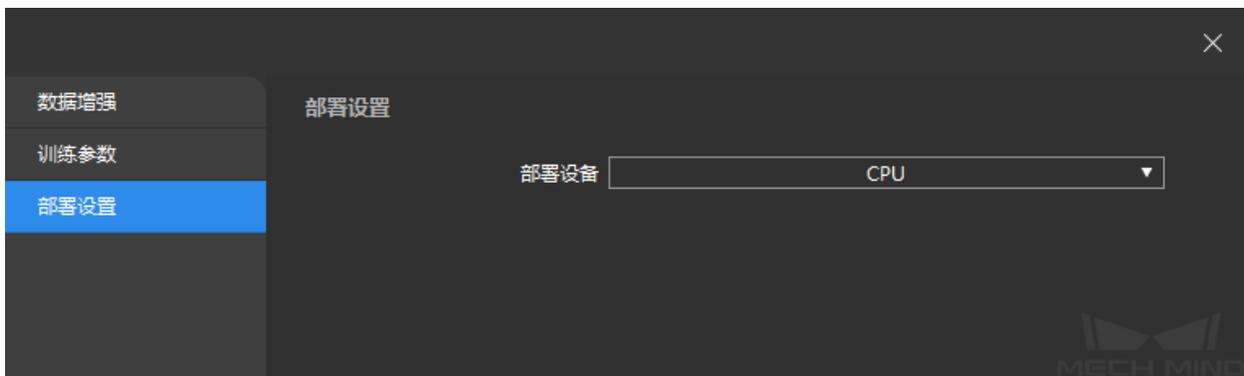
类激活图可展示模型训练时，当前图像中被关注的特征区域，从而帮助检查分类情况，为模型优化提供参考。



6.4 CPU 和 GPU 版本模型部署

6.4.1 CPU 版本模型部署

模型部署在 CPU 设备时，将 部署设备 选择为 CPU。



6.4.2 GPU 版本模型部署

模型部署在 GPU 设备时，将 部署设备 选择为 GPU。

- **dlkmo**: 模型需要优化大约 5 分钟，适用于训练模型与部署模型使用的设备 GPU 显卡型号不同的情况。
- **dlkmt**: 模型无需优化可以直接导出使用，仅支持训练模型与部署模型使用的设备 GPU 显卡型号相同的情况。

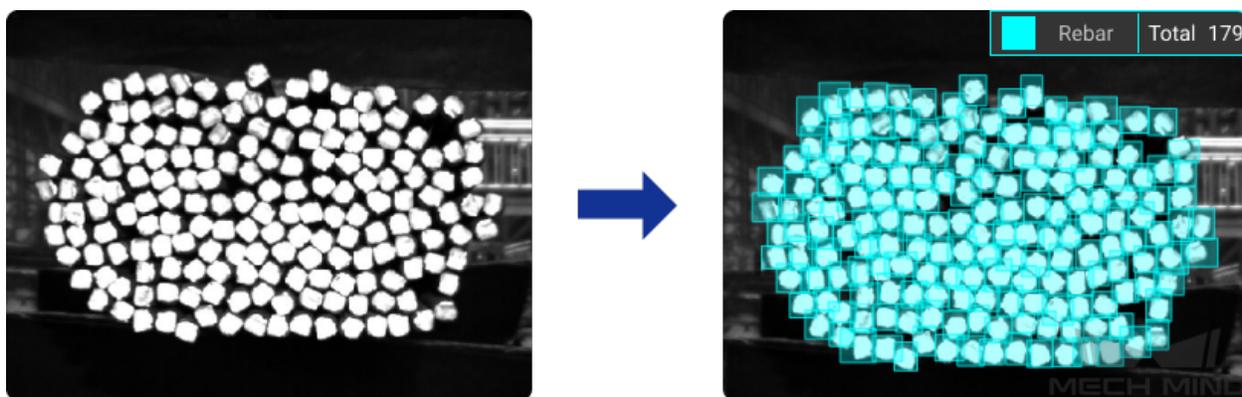


7.1 算法介绍

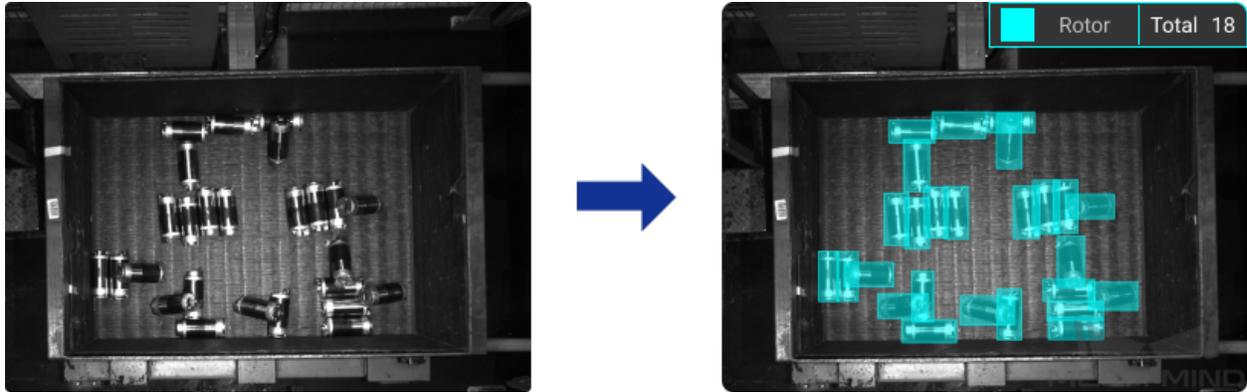
目标检测可以检测所有目标物体的位置，同时判断其类别。

7.1.1 应用场景

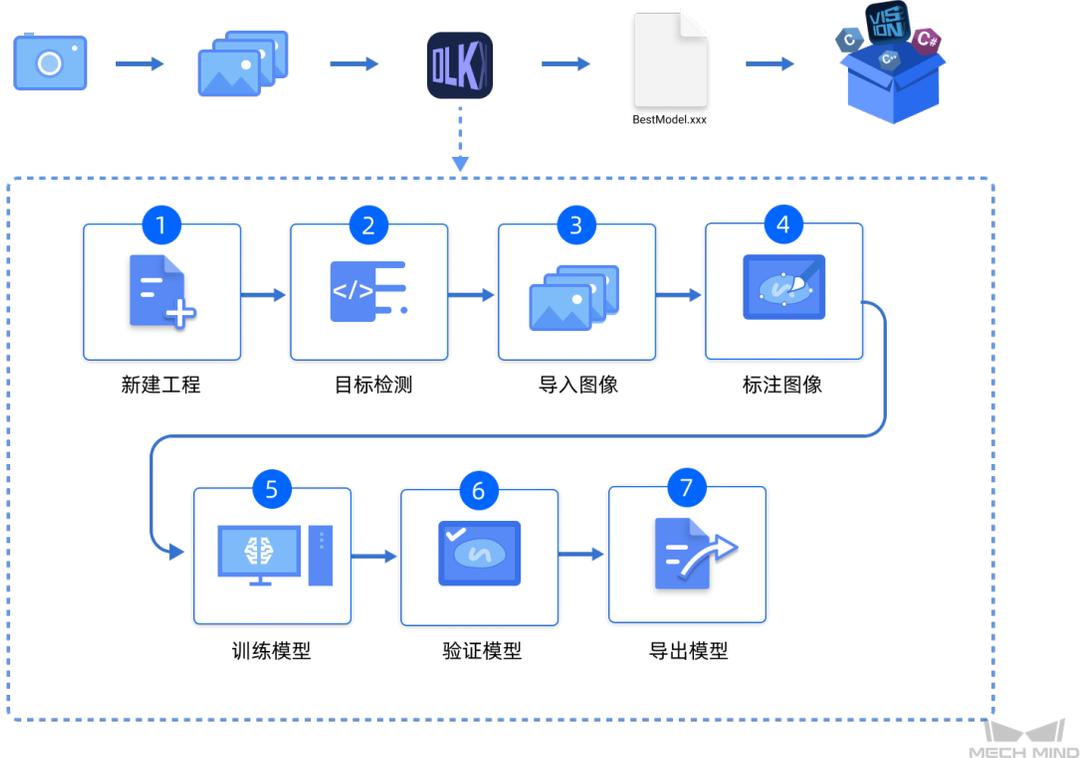
工件计数：适用于工厂中成捆钢筋、散乱金属件，以及细小的零部件等的计数场景。



检测工件位置：适用于工厂中或车间流水线上检测并提取零件所在区域的场景。



7.1.2 应用流程



7.1.3 应用要点

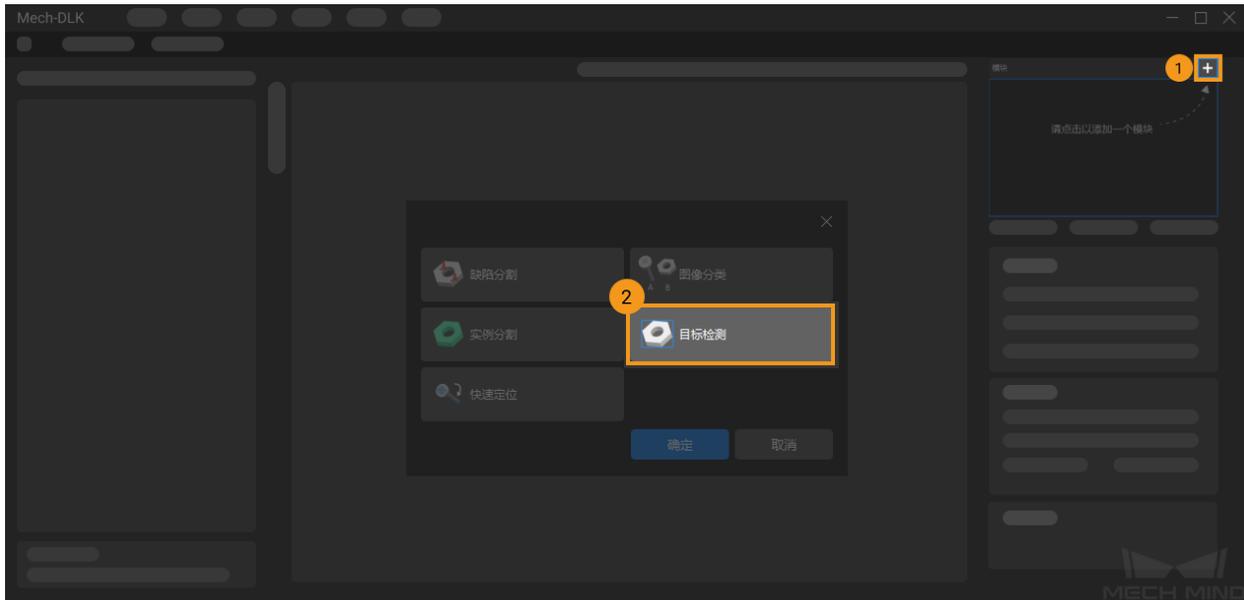
以下几点有助于提高模型质量：

1. 确保图像质量[前往章节 >](#)。
2. 确保数据集质量[前往章节 >](#)。
3. 确保标注质量[前往章节 >](#)。

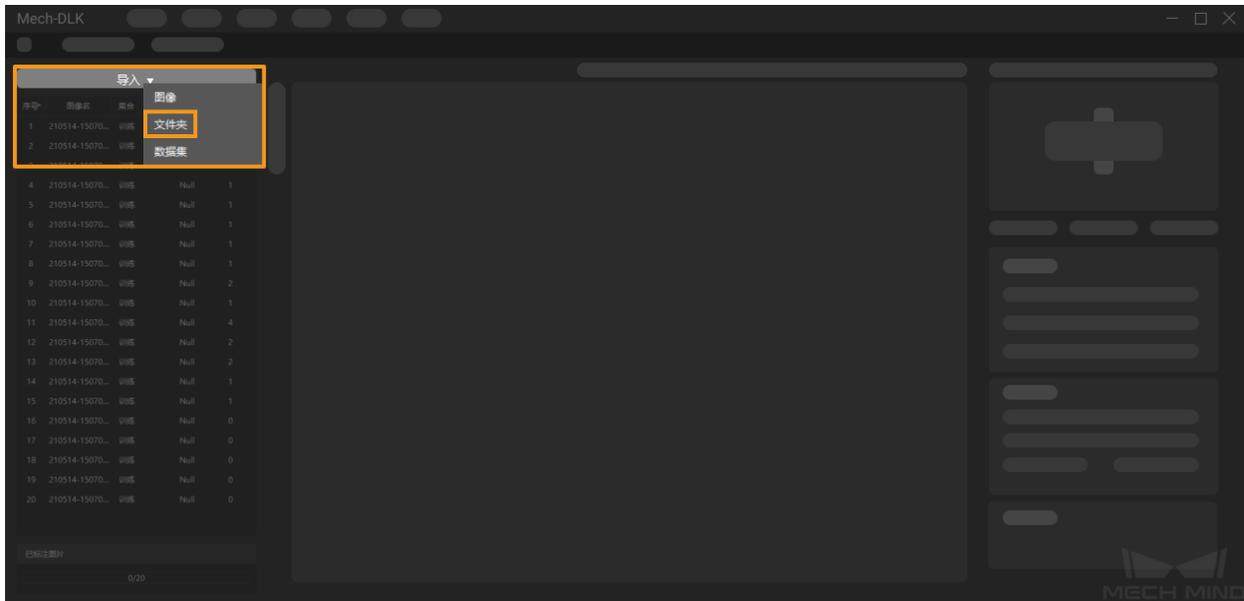
7.2 使用目标检测模块

本章将为用户提供转子数据集（点击下载），带领用户使用“目标检测”模块训练模型，达到示例工程中检测所有转子的位置并输出数量的效果。

1. 新建工程并添加“目标检测”模块：点击初始页面上的 新建工程，选择工程路径并输入工程名以新建一个工程。再点击右上角的 ，选择“目标检测”模块。



2. 导入转子图像数据集：解压缩下载的数据集压缩包，点击左上方的 导入，选择 文件夹导入下载好的图像数据集。

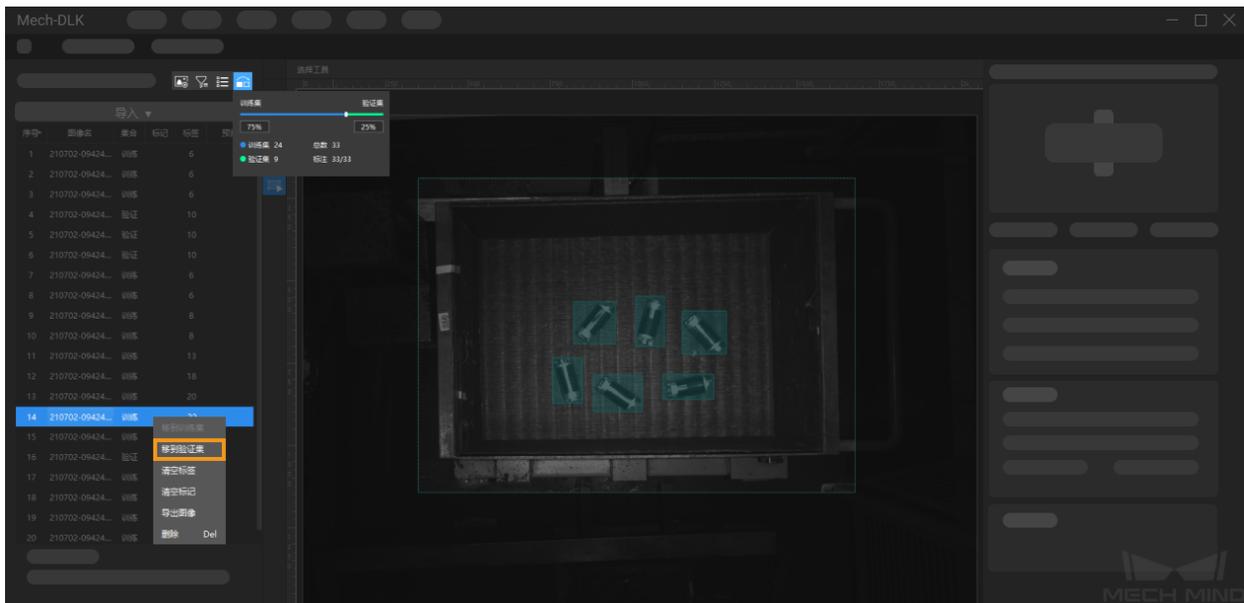


3. **截取 ROI:** 点击选项  从图像中框选盛放转子的料筐作为感兴趣区域，并点击左上方的应用确认使用。截取 ROI 的目的是减少无关背景信息的干扰。



4. **划分训练集与验证集:** 软件默认将数据集的 80% 划分为训练集，20% 为验证集，可以点击  拖动滑块调整图像占比。需要确保划分后的训练集和验证集中涵盖所有需要检测的物体种类，如果不满足这

一条件，可以选中图像名单单击右键进行调整。



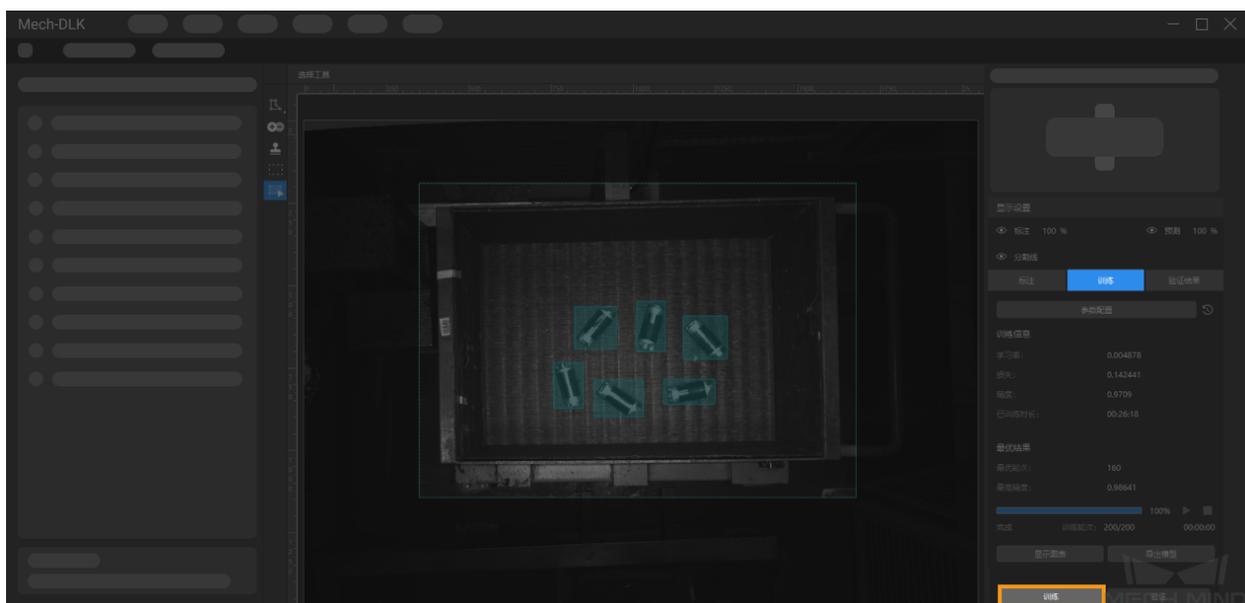
5. 创建标签：根据物体名称或特征创建对应的标签，此数据集以转子英文名称创建标签。



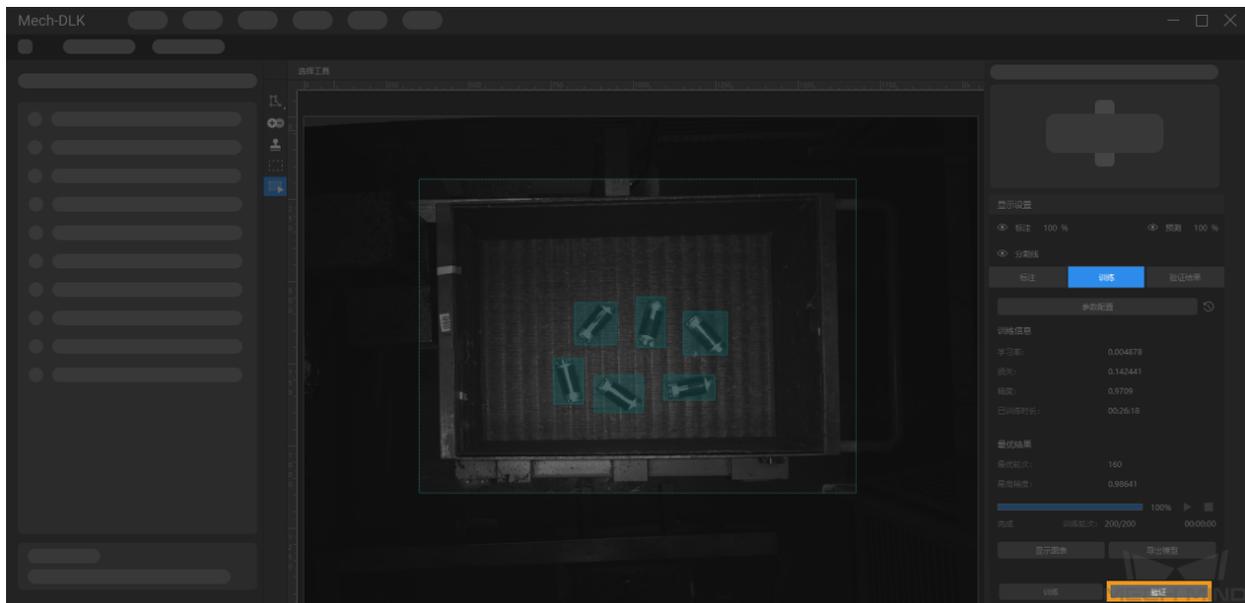
6. 标注图像：以矩形框标注出图像中所有转子。标注时尽量矩形框刚好包住转子边缘，避免过多标注转子边缘以外的区域，标注不准确会影响模型训练效果。



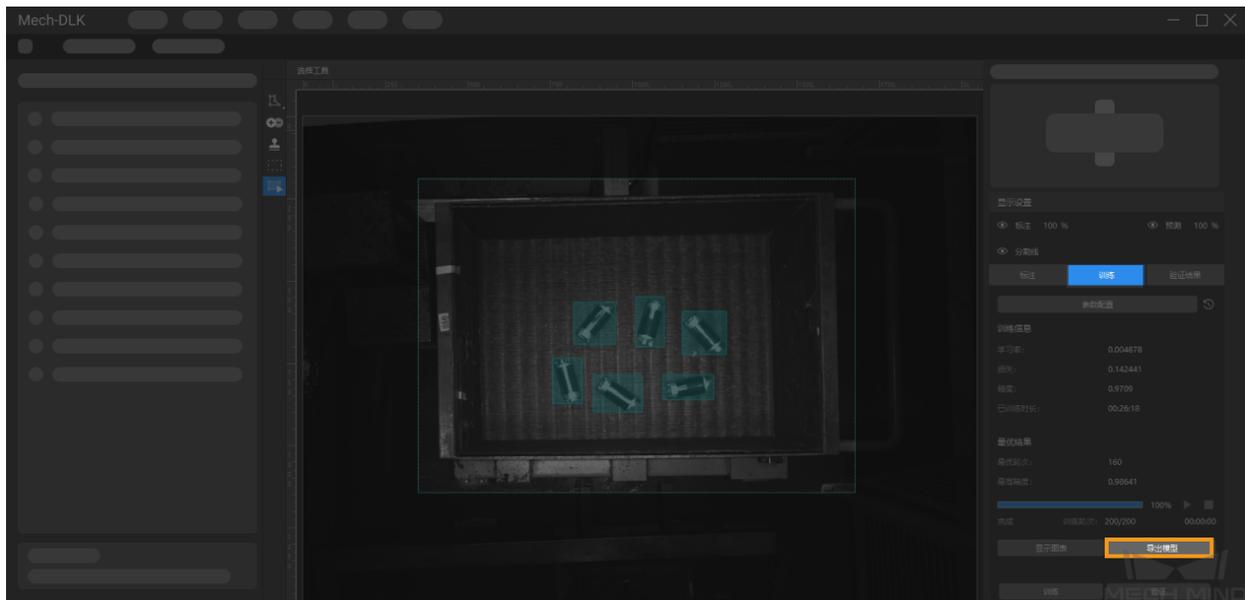
7. 训练模型：使用默认参数设定，点击 训练开始训练模型。



8. 验证模型：训练结束后，点击 验证可以验证并查看模型识别效果。



9. 导出模型：点击 导出模型 选择存放路径即可导出模型到工程文件夹，用户可以根据需求部署模型。



7.3 如何训练高质量模型

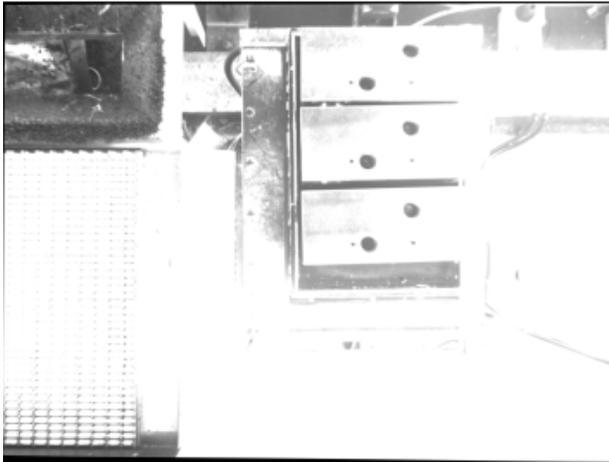
本章将介绍最影响模型质量的几个因素，以及如何训练出高质量目标检测模型。

- 确保图像质量
- 确保数据集质量
- 确保标注质量

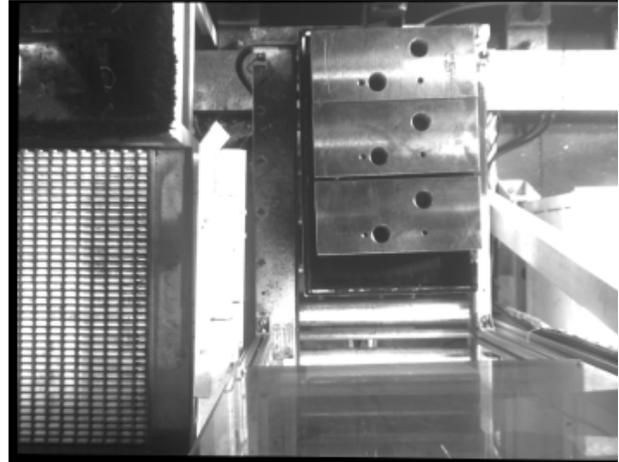
7.3.1 确保图像质量

1. 避免 过曝、过暗、颜色失真、模糊、遮挡等。这些情况会导致深度学习模型所依赖的特征丢失，影响模型训练效果。

错误示例：过曝

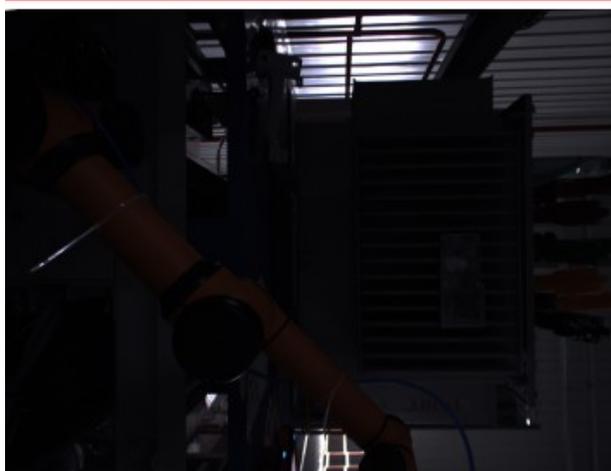


正确示例：正常



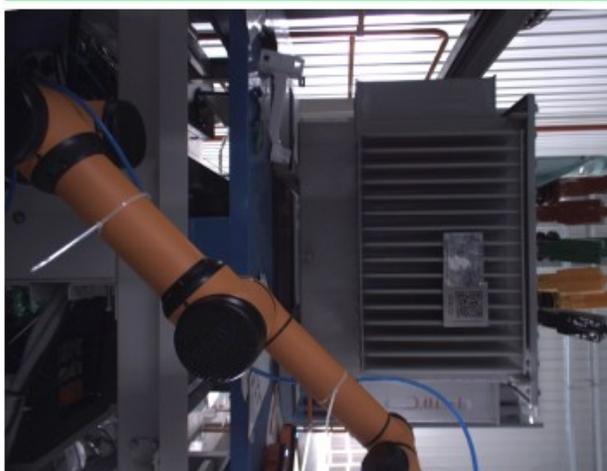
优化建议：可通过遮光等方式避免

错误示例：过暗



优化建议：可通过补光等方式避免

正确示例：正常

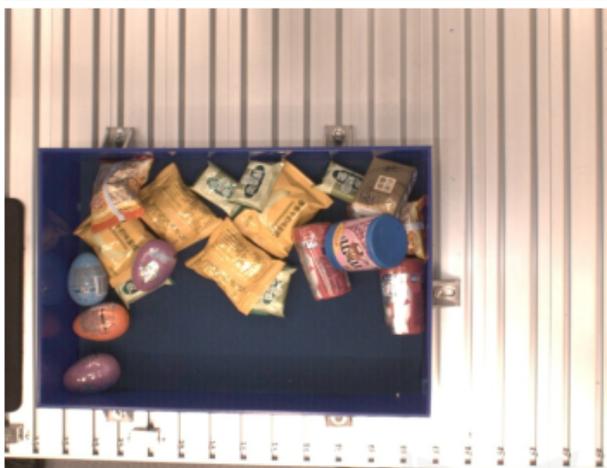


错误示例：颜色失真

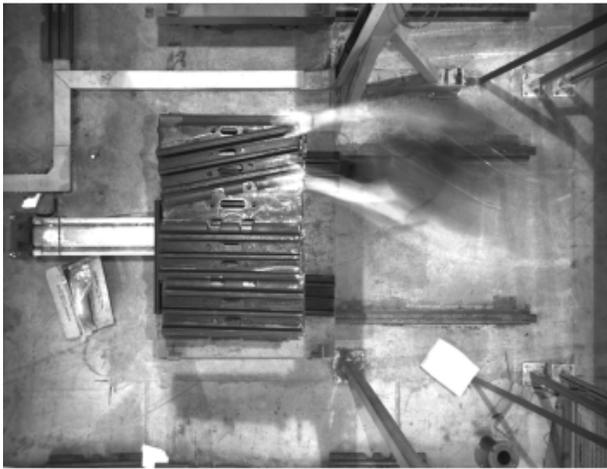


优化建议：可通过调整相机白平衡避免

正确示例：正常



错误示例：模糊



优化建议：避免相机或物体仍在运动时拍照

正确示例：正常



错误示例：机械臂遮挡



优化建议：确保机械臂和人离开拍照区

错误示例：人手遮挡



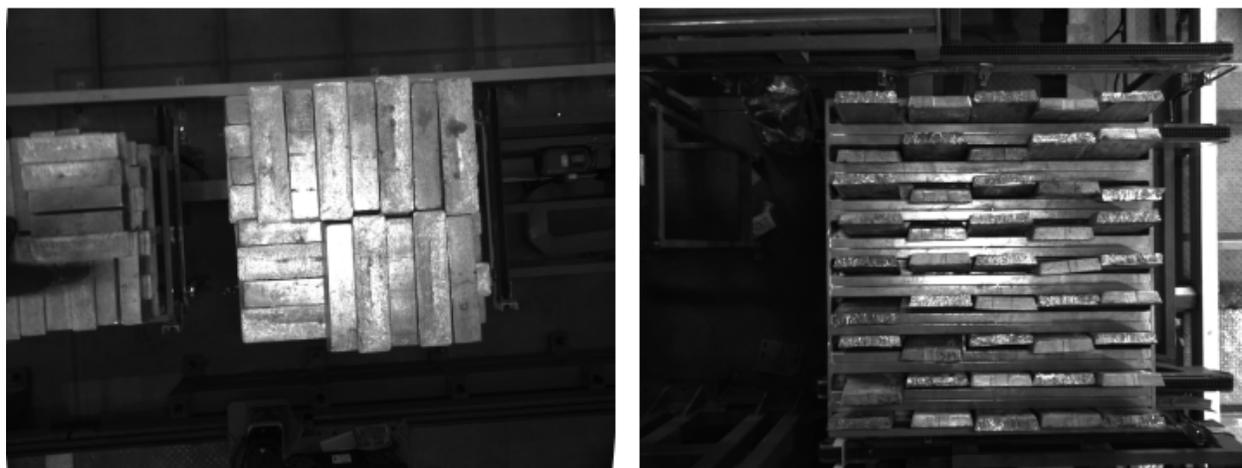
2. 确保采集数据的背景、视角、高度与实际应用一致。任何不一致都会降低深度学习在实际应用时的效果，严重情况下必须返工重新采集数据，请务必提前确认实际应用时的情况。

错误示例：训练数据背景（图左）与实际应用背景（图右）不一致



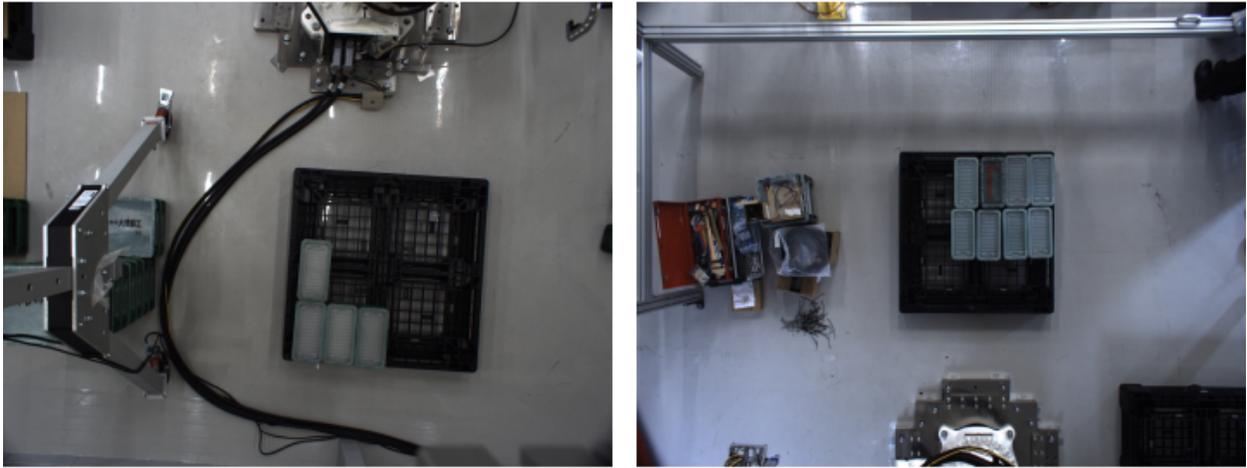
优化建议：保持训练数据背景与实际应用一致

错误示例：训练数据视野（图左）与实际应用视野（图右）不一致



优化建议：保持训练数据采集视野与实际应用一致

错误示例：训练数据高度（图左）与实际应用高度（图右）不一致



优化建议：保持训练数据采集高度与实际应用一致

7.3.2 确保数据集质量

“目标检测”模块通过学习图像中的物体特征得出模型并应用到实际场景，因此采集和选取的数据集必须与实际场景情况一致才能训练出高质量模型。

1. 采集数据集
2. 选取数据集

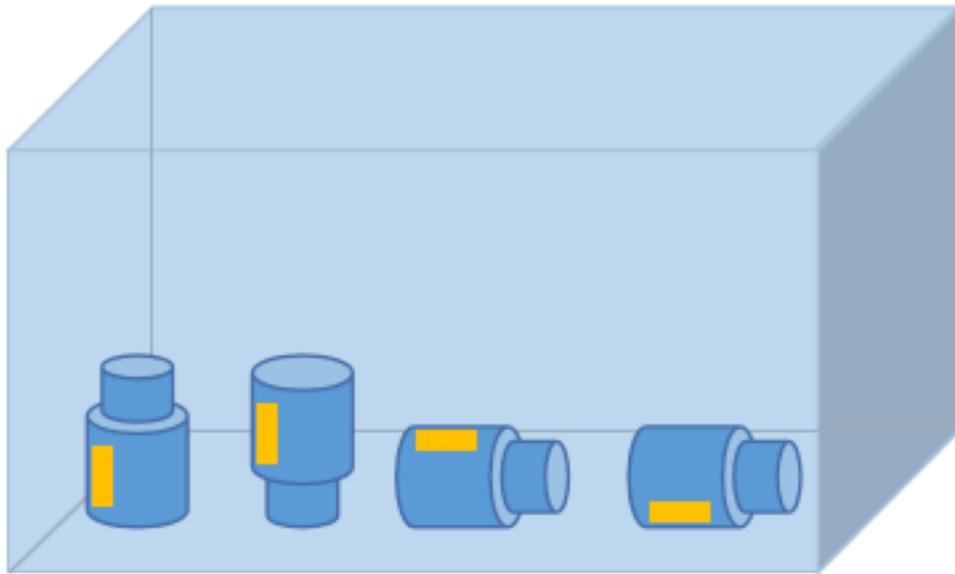
采集数据集

需合理分配各种摆放情况。例如实际生产时来料有横向、竖向和散乱堆叠的情况，但只采集横向和竖向来料的图像数据集进行训练，则无法保证散乱堆叠的识别效果。因此，采集数据时需要能包含实际生产的各种场景，具体包括：

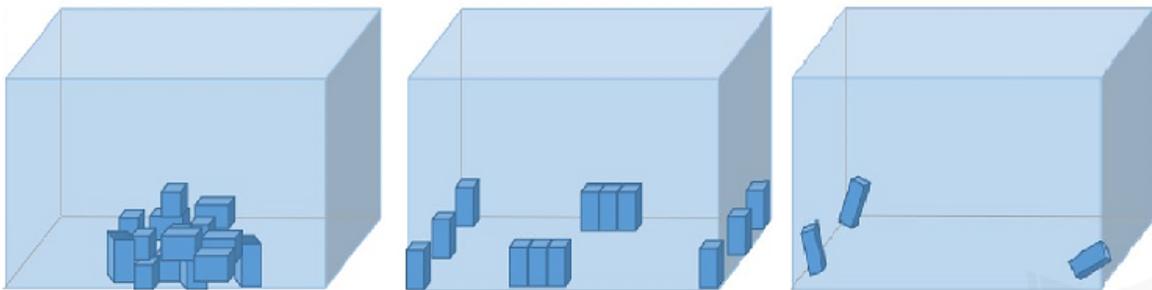
- 实际应用所有可能出现的 物体朝向。
- 实际应用所有可能出现的 物体位置。
- 实际应用所有可能出现的 物体间关系。

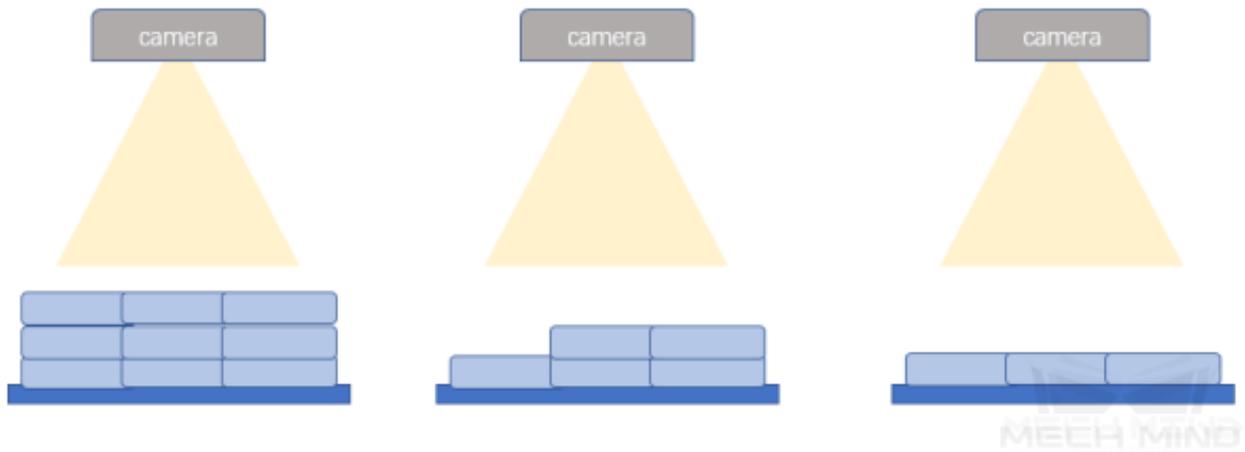
注意：如果少采集了某种情况的数据，算法模块会缺少该情况下数据的学习，将导致模型在该情况下识别效果不佳，因此必须根据情况增加数据样本，降低误差。

物体朝向

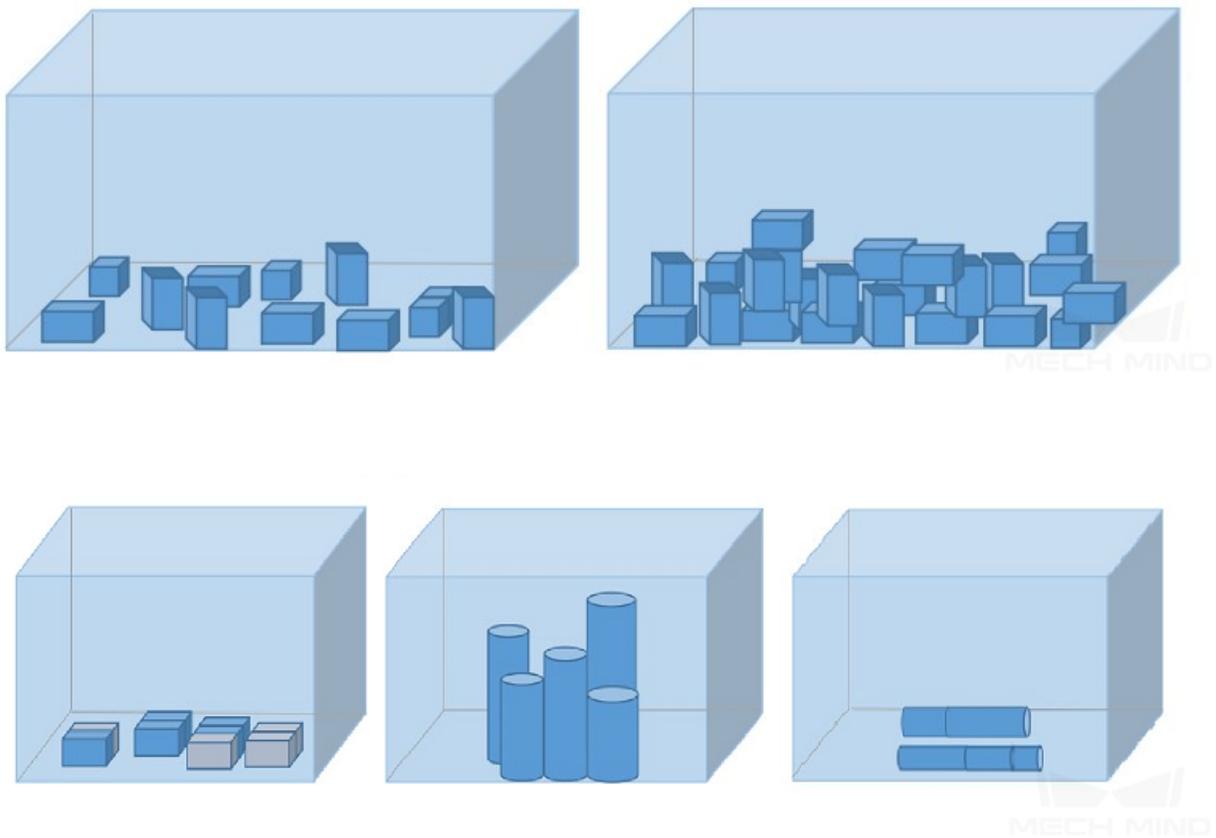


物体位置



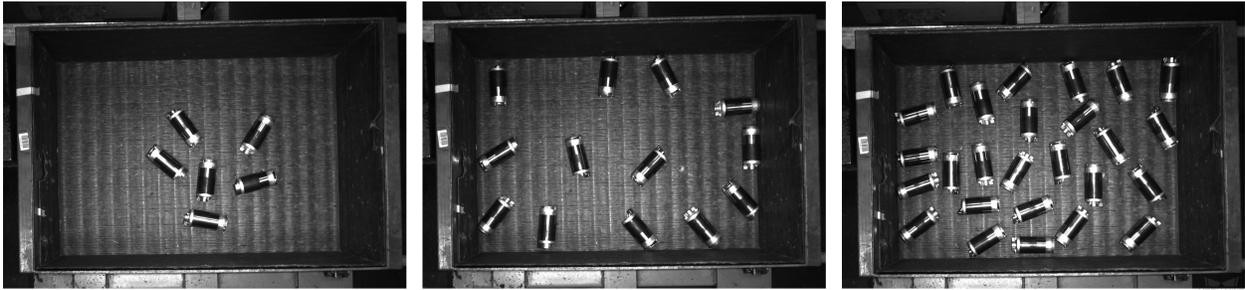


物体间关系



采集数据案例

1. 某工件检测项目，来料为无规则散乱摆放的转子，要求精准检测所有的转子位置。一共采集 30 张。
 - 物体位置方面，需考虑采集实际生产时转子可能位于料筐的所有位置；同时需要考虑抓取时转子数量由多到少的情况。
 - 物体间关系方面，需考虑转子散乱、整齐和压叠摆放的情况。



1. 某钢筋计数项目，来料为成捆的钢筋，要求精准输出钢筋数量。一共采集 20 张。
 - 钢筋特征相对单一，因此只需考虑 物体位置 的多样性，采集实际生产时钢筋可能位于相机视野中的所有位置。



选取合适的数据集

1. 训练集数量可控

“目标检测”模块初次训练建议准备 20 张图像，不需要过多数据。前期加入大量类似数据训练会延长不必要的训练时间，同时不利于后期改进模型。

2. 数据具有代表性

数据集图像一定要涵盖待检测目标的所有光照、颜色、尺寸等信息。

- 光照：实际场景存在光照变化，数据集应该包含所有光照情况下的图像。
- 颜色：工件存在不同颜色，数据集应该包含所有颜色的图像。
- 尺寸：工件存在不同尺寸，数据集应该包含所有尺寸工件的图像。

注意：若实际现场工件会出现旋转、缩放、或其他情况，无法采集相应图像数据集时，可以通过调整数据增强训练参数的方式来补充数据集，以确保现场所有的情况都包含在训练集内。

3. 数据占比均衡

训练集中不同种类或不同摆放方式的图像比例要均衡，否则会影响模型效果。禁止出现一种物体 20 张，另一种物体仅有 3 张；或整齐摆放 40 张，散乱摆放只有 5 张的情况。

4. 数据集与终端场景保持一致

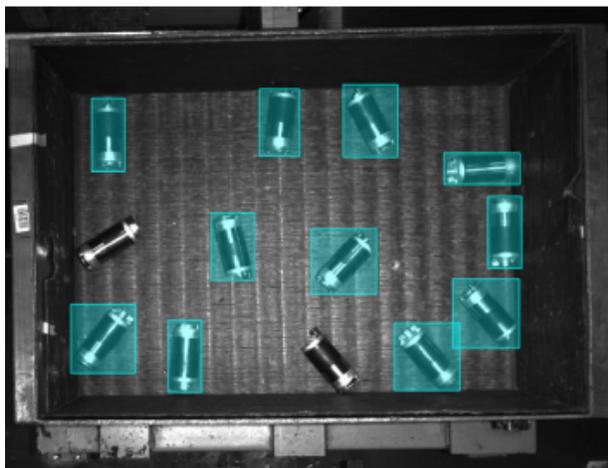
数据集的图像要与最终模型使用的终端场景保持一致，包括光照条件、工件特征、检测背景、视野大小等。

7.3.3 确保标注质量

标注质量应从完备性、精确性两个方面考虑：

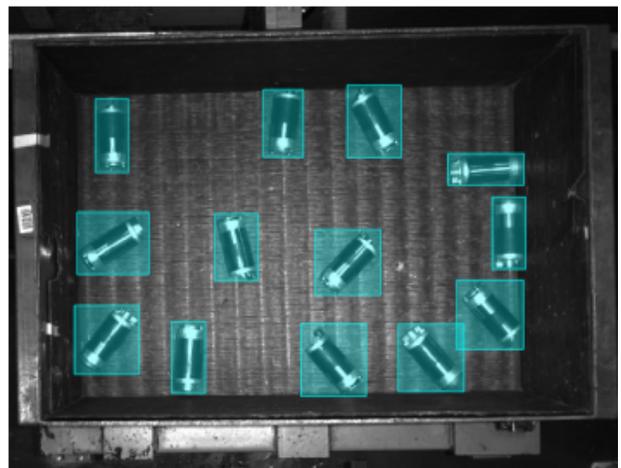
1. **完备性：**标注所有物体，禁止出现遗漏标注的情况。

错误示例：漏标注



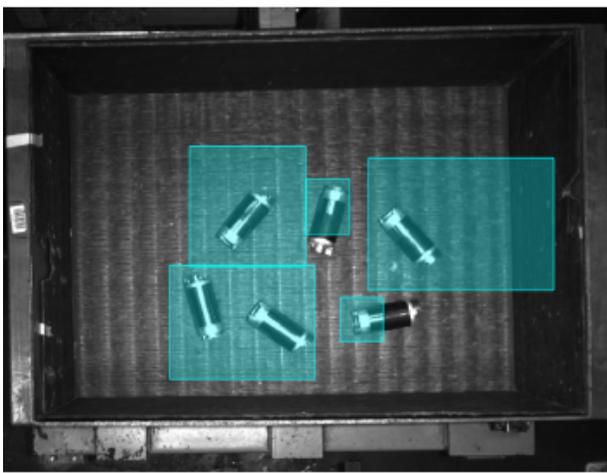
标注提示：要求标注所有转子时，不应遗漏某一个

正确示例：正确标注

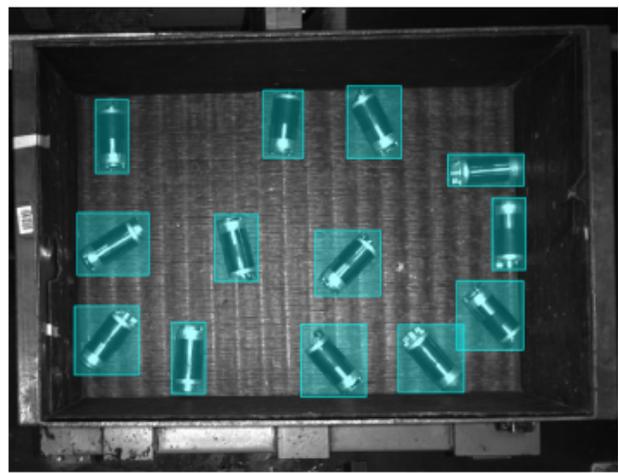


2. **精确性：**标注的矩形框应包住物体，禁止出现少标注物体某部分，或大量标注物体轮廓外区域的情况。

错误示例：少标注、多标注



正确示例：正确标注



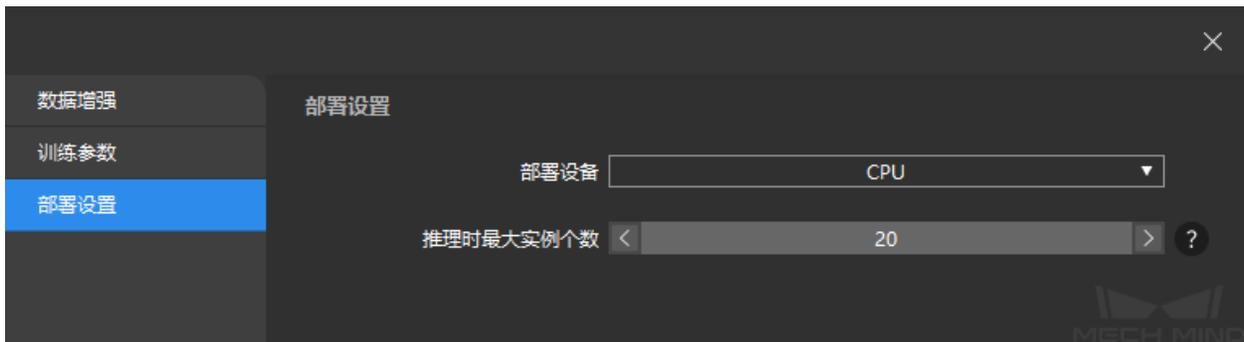
标注提示：要求完整标注所有转子时，不应遗漏转子的某部分或多标注无关区域

MECH MIND

7.4 CPU 和 GPU 版本模型部署

7.4.1 CPU 版本模型部署

模型部署在 CPU 设备时，将部署设备选择为 CPU。



7.4.2 GPU 版本模型部署

模型部署在 GPU 设备时，将 部署设备选择为 GPU。

- **dlkmo**: 模型需要优化大约 5 分钟，适用于训练模型与部署模型使用的设备 GPU 显卡型号不同的情况。
- **dlkmt**: 模型无需优化可以直接导出使用，仅支持训练模型与部署模型使用的设备 GPU 显卡型号相同的情况。



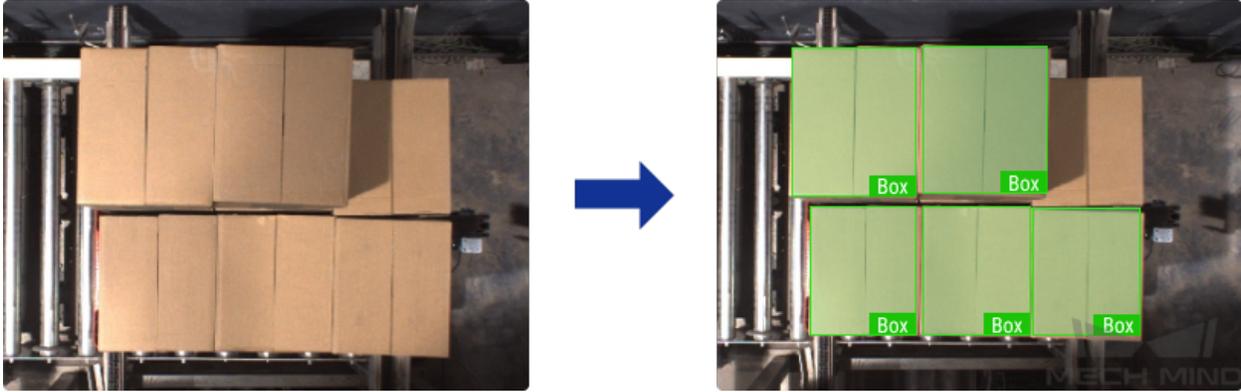
8.1 算法介绍

实例分割可以分割出每个目标物体的轮廓，同时输出类别标签。

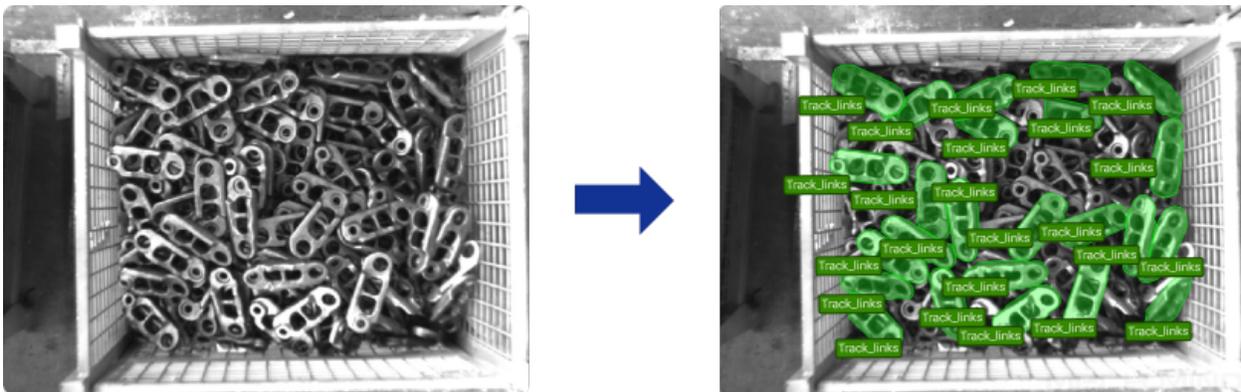
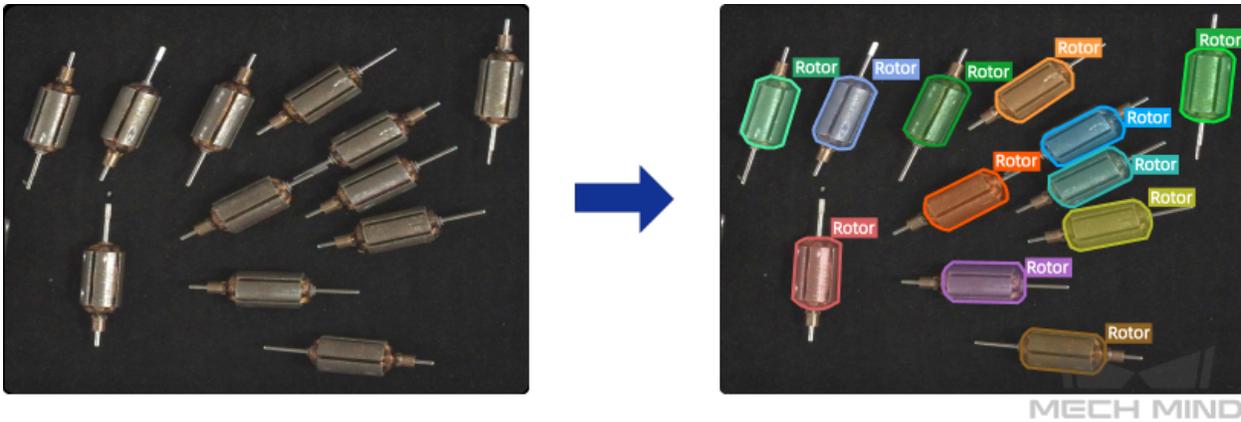
8.1.1 应用场景

拆码垛：适用于从托盘上按要求拆卸纸箱、周转箱、麻袋等物体，并放置于托盘或相关设备上（如破袋设备、传送带等）的场景。

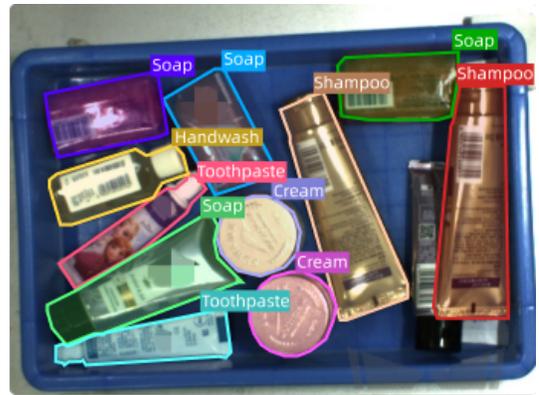




工件上下料：适用于汽车、钢铁、机械等行业中复杂工件、结构件、不规则零部件等物体的搬运抓取场景。



货品拣选：适用于播种、摘果等各种电商仓库中的常见拣货场景。支持各种货品，包括充气袋装、透明包装、瓶装铝罐、不规则的货品等。



MECH MIND

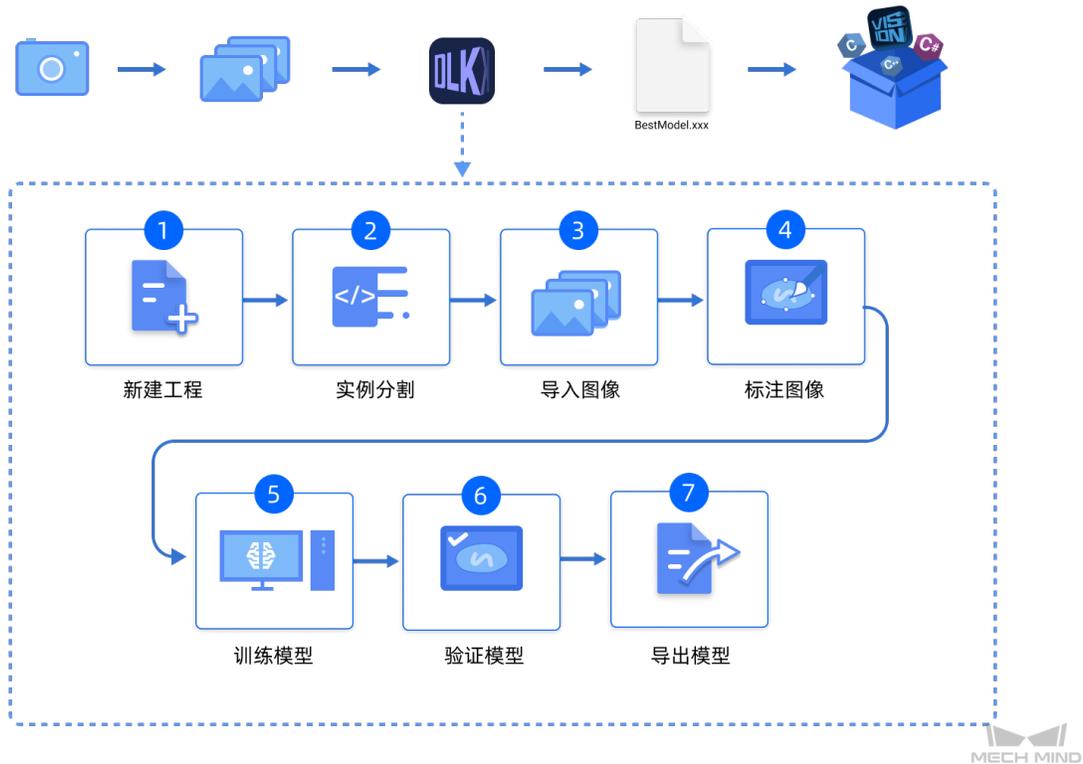


快递包裹供包：支持快递软包、邮政信封、快递纸箱、泡沫信封等各种常见包裹，同时可支持各种包裹异形件。



MECH MIND

8.1.2 应用流程



8.1.3 应用要点

以下几点有助于提高模型质量：

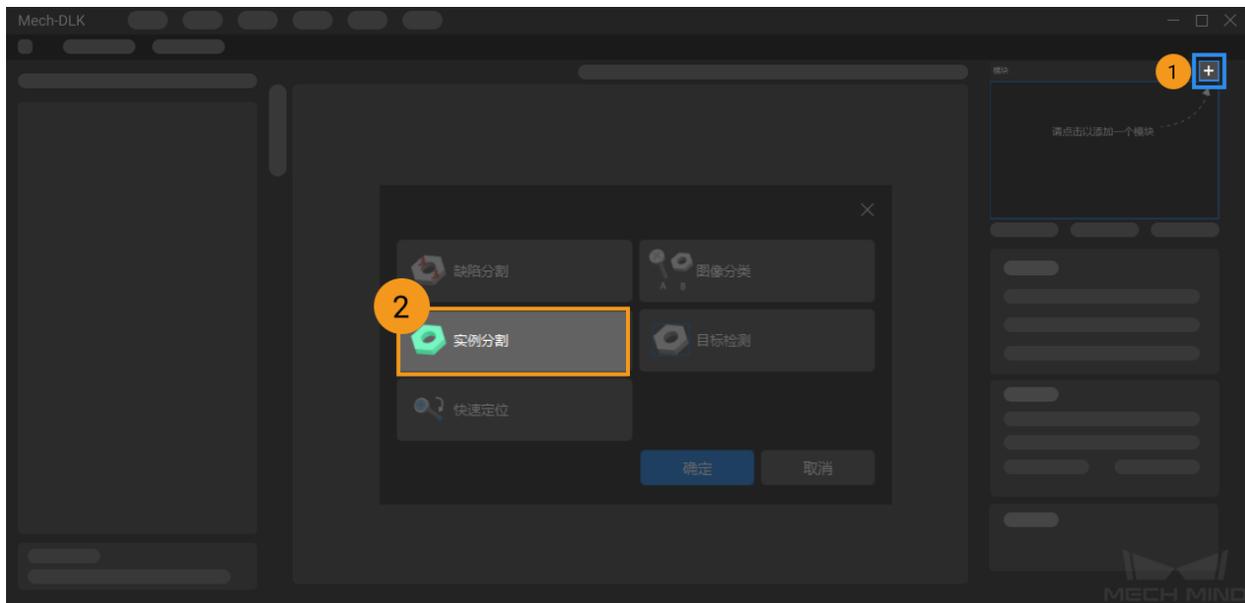
1. 确保图像质量[前往章节 >](#)。
2. 确保数据集质量[前往章节 >](#)。
3. 确保标注质量[前往章节 >](#)。

8.2 使用实例分割模块

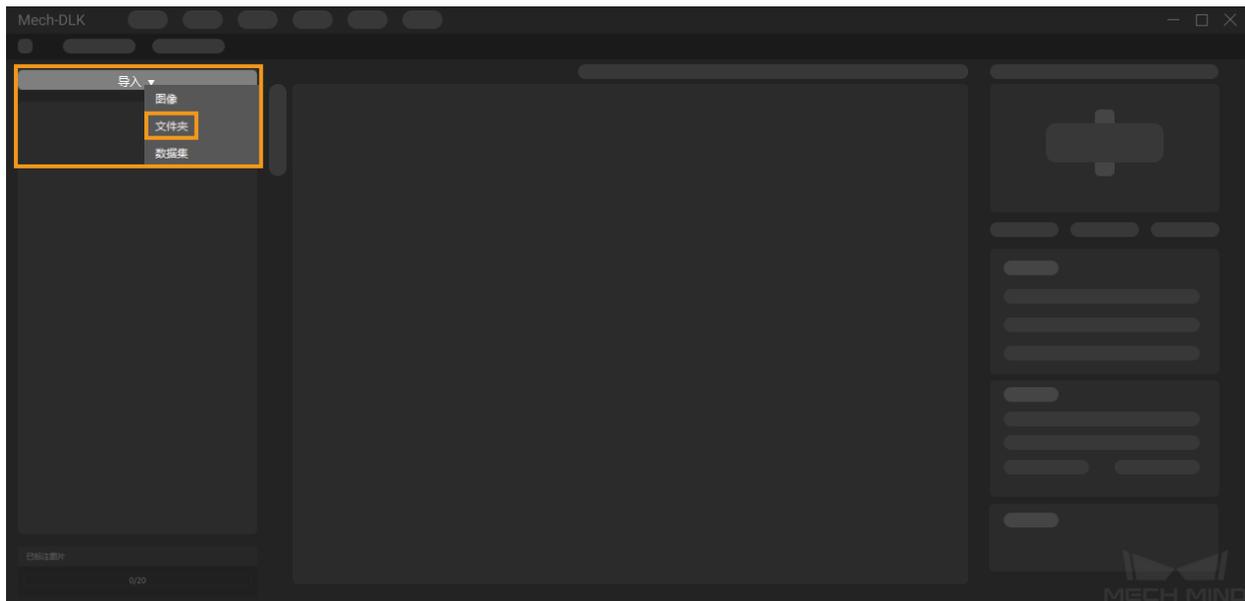
本章将为用户提供 Mech-DLK 示例工程中的积木数据集（[点击下载](#)），带领用户使用“实例分割”模块训练模型，达到示例工程中分割不同种类的积木并输出类别标签的效果。

1. 新建工程并添加“实例分割”模块：点击初始页面上的 新建工程，选择工程路径并输入工程名以新建

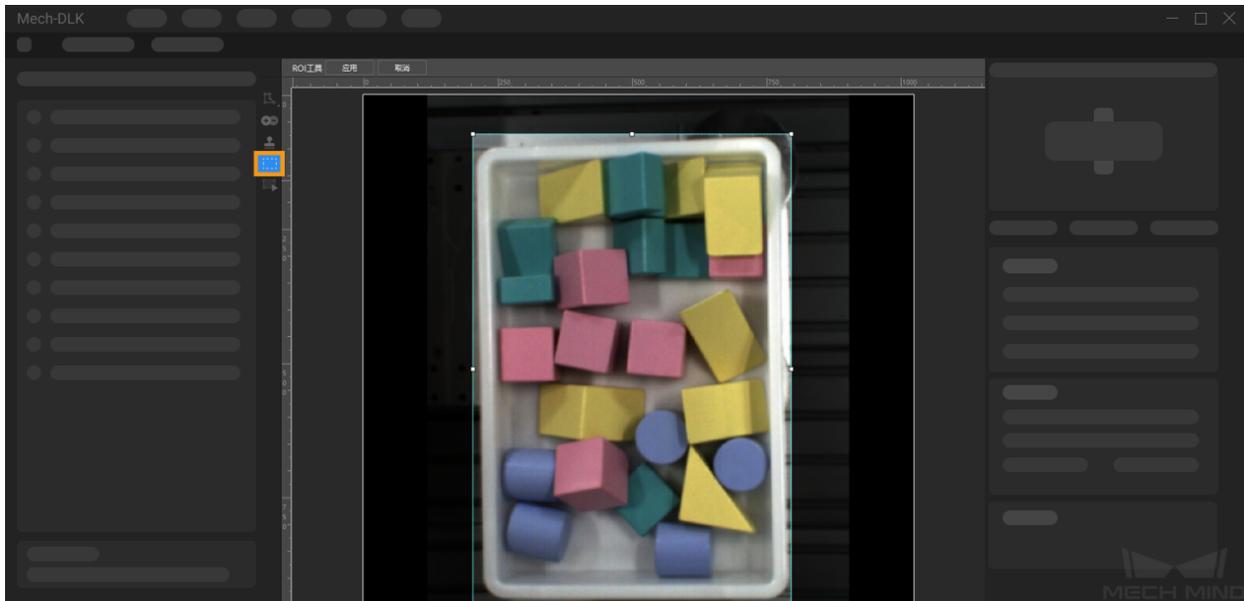
一个工程。再点击右上角的 ，选择“实例分割”模块。



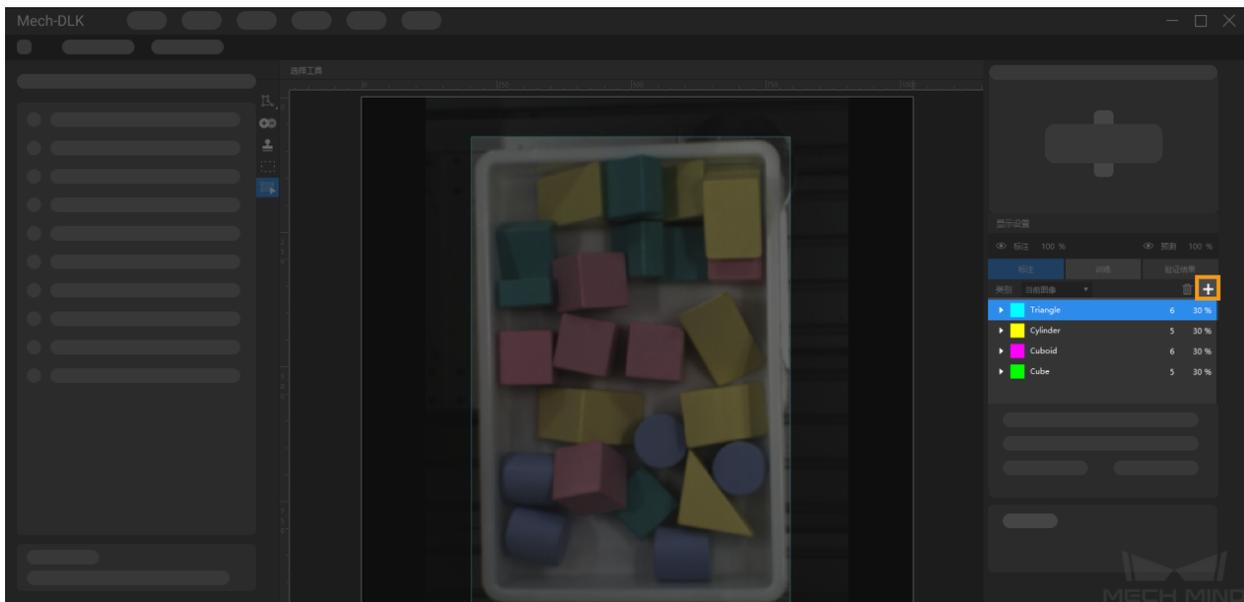
2. **导入积木图像数据集：**解压缩下载的数据集压缩包，点击左上方的 导入，选择 文件夹导入下载好的图像数据集。积木图像数据集包含了四种不同形状、颜色的积木。



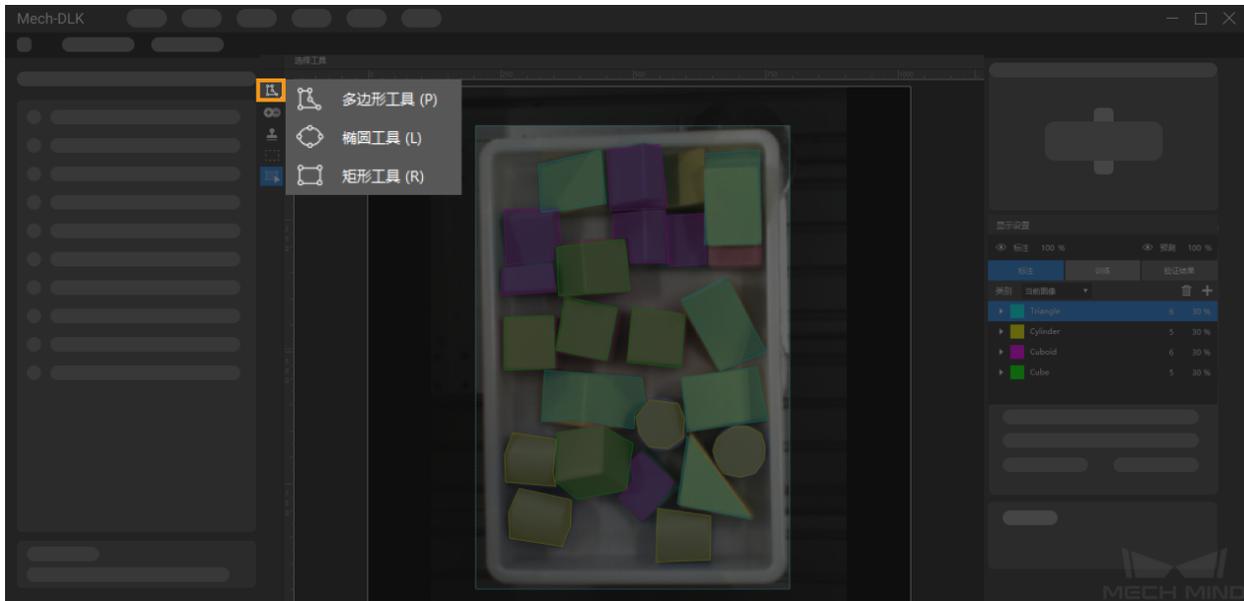
3. **截取 ROI：**点击选项  从图像中框选盛放积木的料筐作为感兴趣区域，并点击左上方的 应用 确认使用。截取 ROI 的目的是减少无关背景信息的干扰。



4. **创建标签**: 点击左侧标签栏的 ，根据物体名称或特征创建对应的标签。此处需要分割不同种类的积木，所以根据积木的不同形状来命名；用户也可以选择根据积木的不同颜色特征命名。

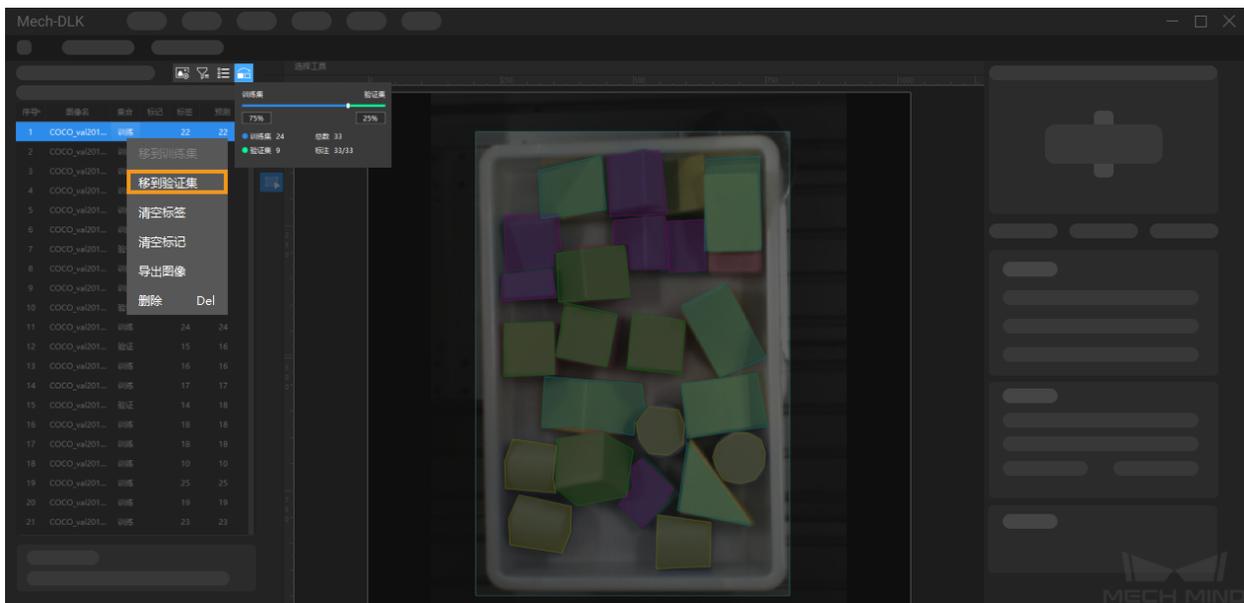


5. **标注图像**: 鼠标左键长按或右键单击图像左侧工具栏 ，选择适合的标注工具，标注出图像中所有需要分割积木的外轮廓，并确保标注的标签与积木的形状类别对应。

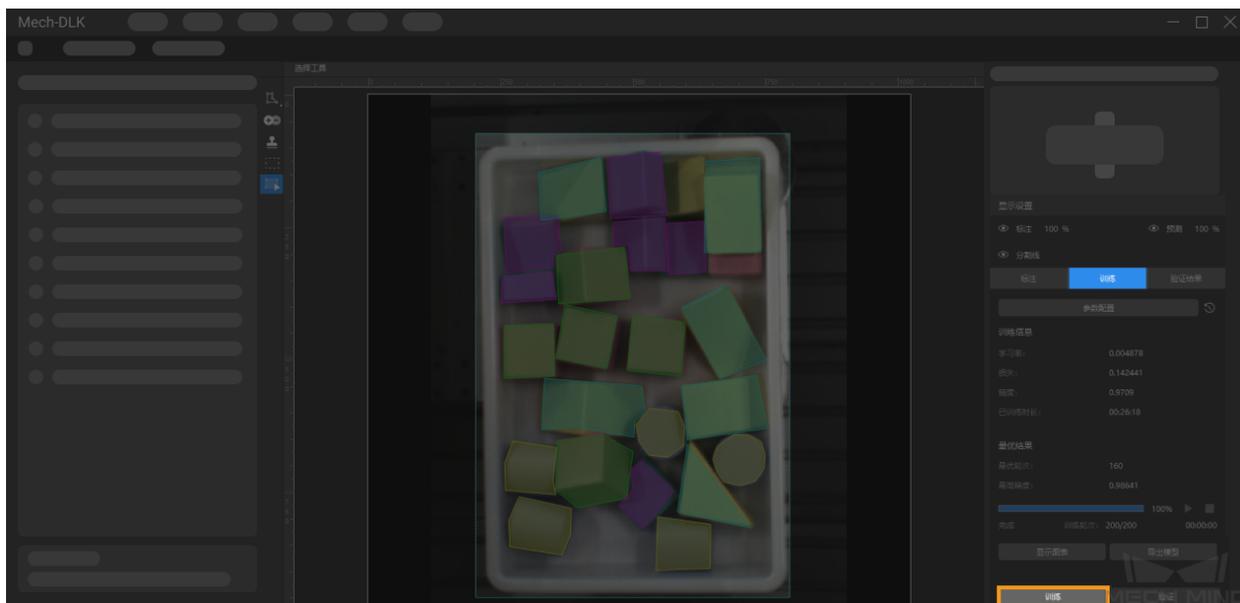


小技巧: 使用多边形工具进行标注，标注完成后，可以拖拽锚点对标注结果进行调整。

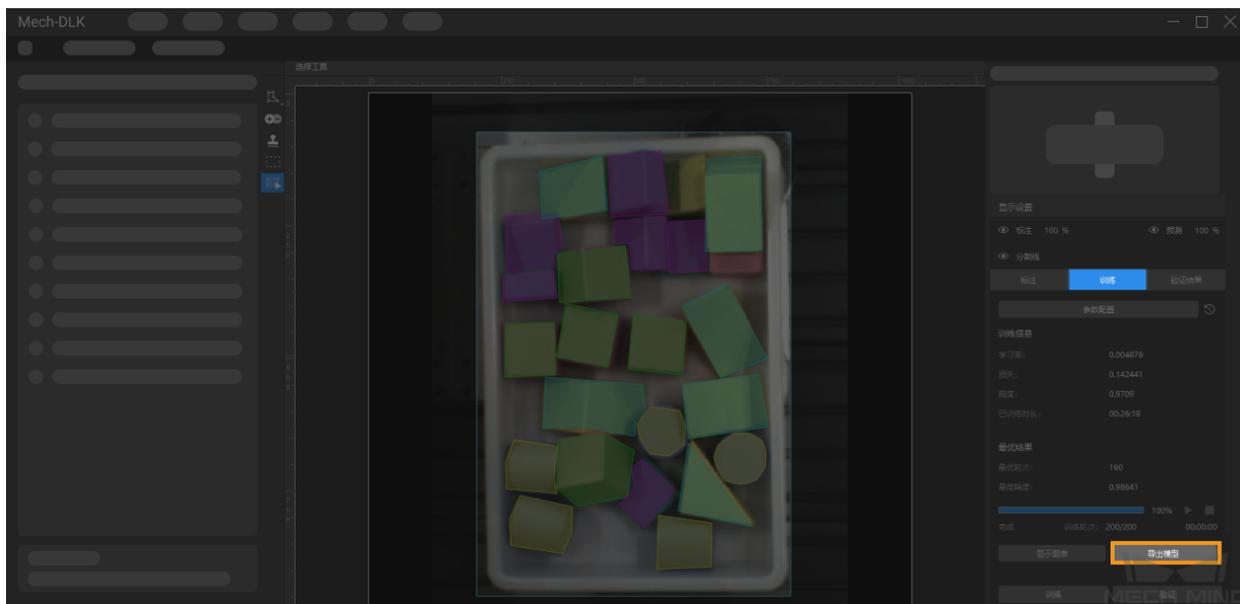
6. **划分训练集与验证集:** 软件默认将数据集的 80% 划分为训练集，20% 为验证集，可以点击  拖动滑块调整图像占比。需要确保划分后的训练集和验证集中涵盖所有需要分割的种类，如果不满足这一条件，可以选中图像名单击右键进行调整。



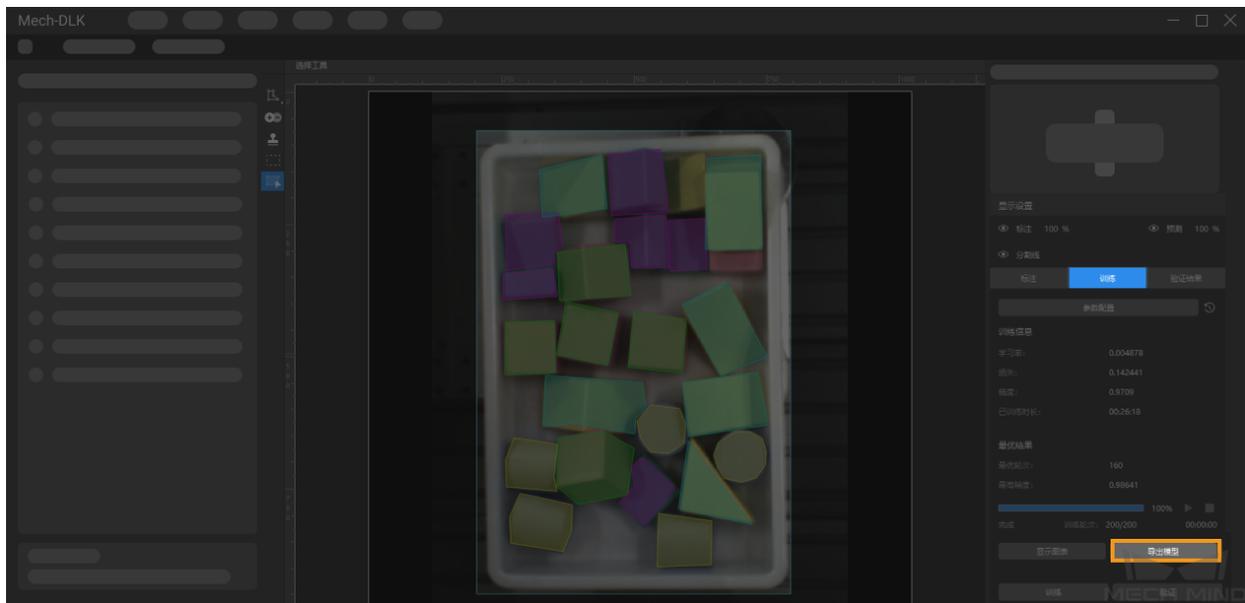
7. 训练模型：使用默认参数设定，点击 训练开始训练模型。



8. 验证模型：训练结束后，点击 验证可以验证并查看模型识别效果。



9. 导出模型：点击 导出模型选择存放路径即可导出模型到工程文件夹，用户可以根据需求部署模型。



8.3 如何训练高质量模型

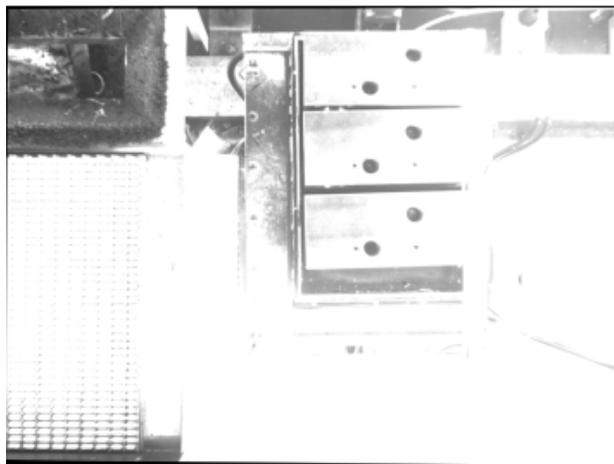
本章将介绍最影响模型质量的几个因素，以及如何训练出高质量实例分割模型。

- 确保图像质量
- 确保数据集质量
- 确保标注质量

8.3.1 确保图像质量

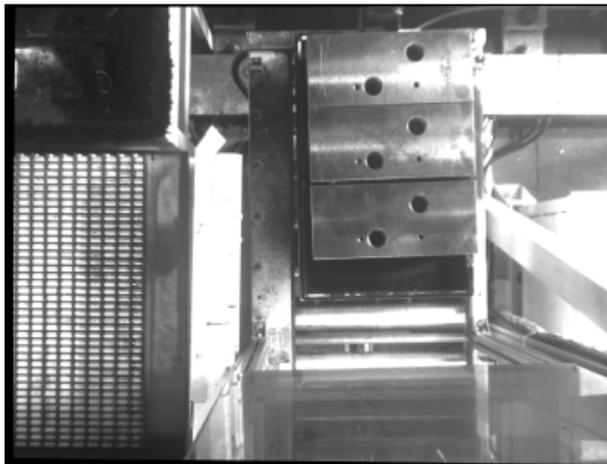
1. 避免 过曝、过暗、颜色失真、模糊、遮挡等。这些情况会导致深度学习模型所依赖的特征丢失，影响模型训练效果。

错误示例：过曝



优化建议：可通过遮光等方式避免

正确示例：正常

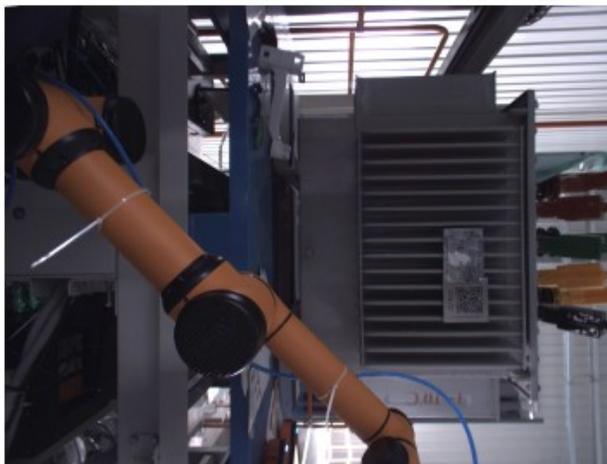


错误示例：过暗



优化建议：可通过补光等方式避免

正确示例：正常



错误示例：颜色失真

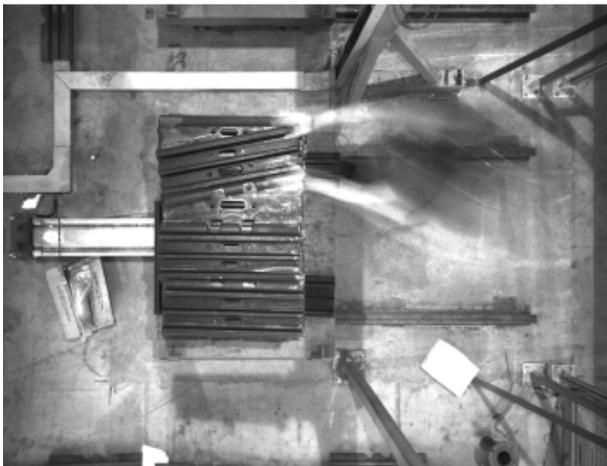


优化建议：可通过调整相机白平衡避免

正确示例：正常



错误示例：模糊



优化建议：避免相机或物体仍在运动时拍照

正确示例：正常



错误示例：机械臂遮挡



优化建议：确保机械臂和人离开拍照区

错误示例：人手遮挡



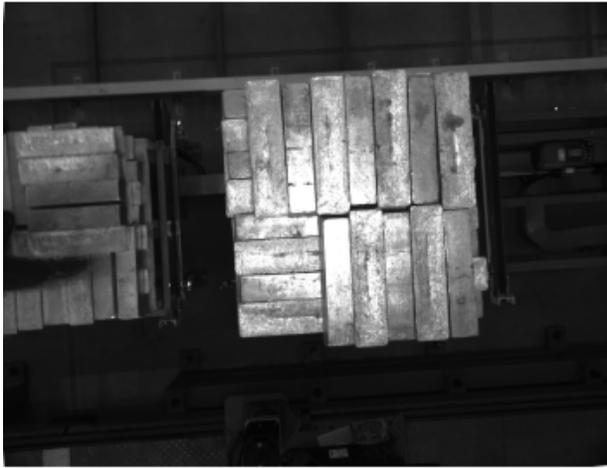
2. 确保采集数据的背景、视角、高度与实际应用一致。任何不一致都会降低深度学习在实际应用时的效果，严重情况下必须返工重新采集数据，请务必提前确认实际应用时的情况。

错误示例：训练数据背景（图左）与实际应用背景（图右）不一致



优化建议：保持训练数据背景与实际应用一致

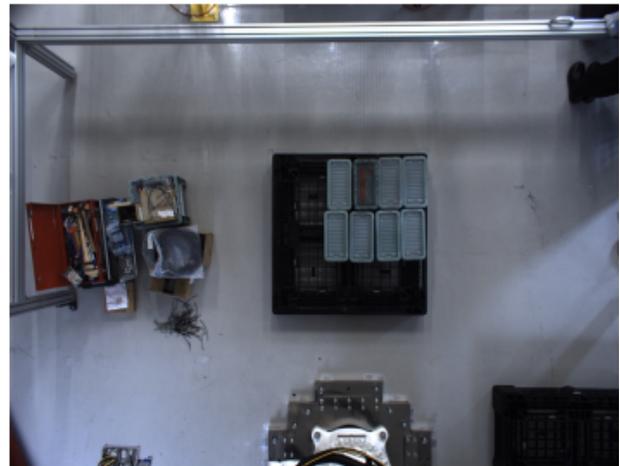
错误示例：训练数据视野（图左）与实际应用视野（图右）不一致



优化建议：保持训练数据采集视野与实际应用一致



错误示例：训练数据高度（图左）与实际应用高度（图右）不一致



优化建议：保持训练数据采集高度与实际应用一致



8.3.2 确保数据集质量

“实例分割”模块通过学习图像中物体的特征得出模型并应用到实际场景，因此采集和选取的数据集必须与实际场景情况一致才能训练出高质量模型。

1. 采集数据集
2. 选取合适的数据集

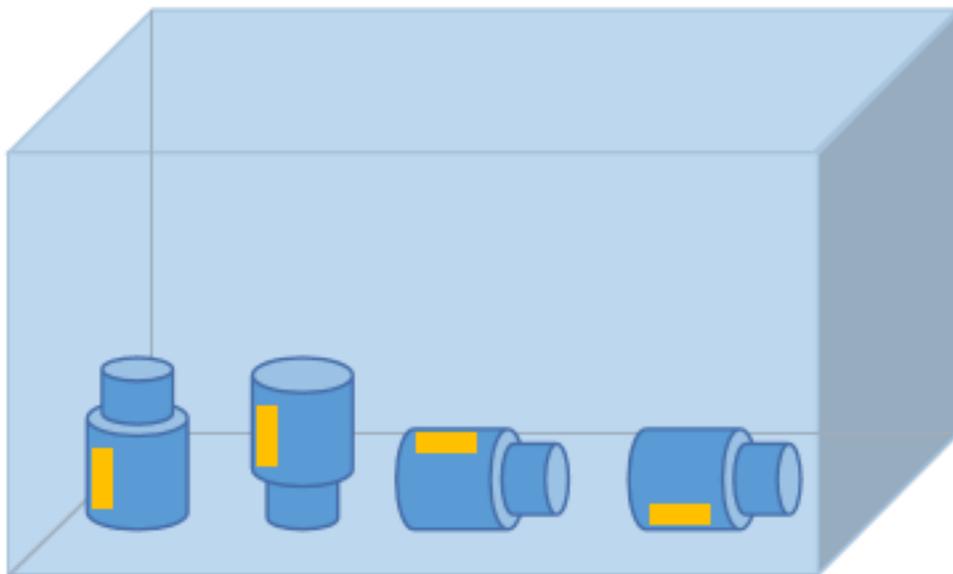
采集数据集

各种摆放情况均需按采集要求合理分配数量，例如实际生产时来料有横向、竖向和散乱堆叠的情况，但只采集了横向和竖向来料的图像数据集进行训练，那么无法保证散乱堆叠的识别效果。因此，采集数据时需要能包含实际生产的各种场景，具体包括：

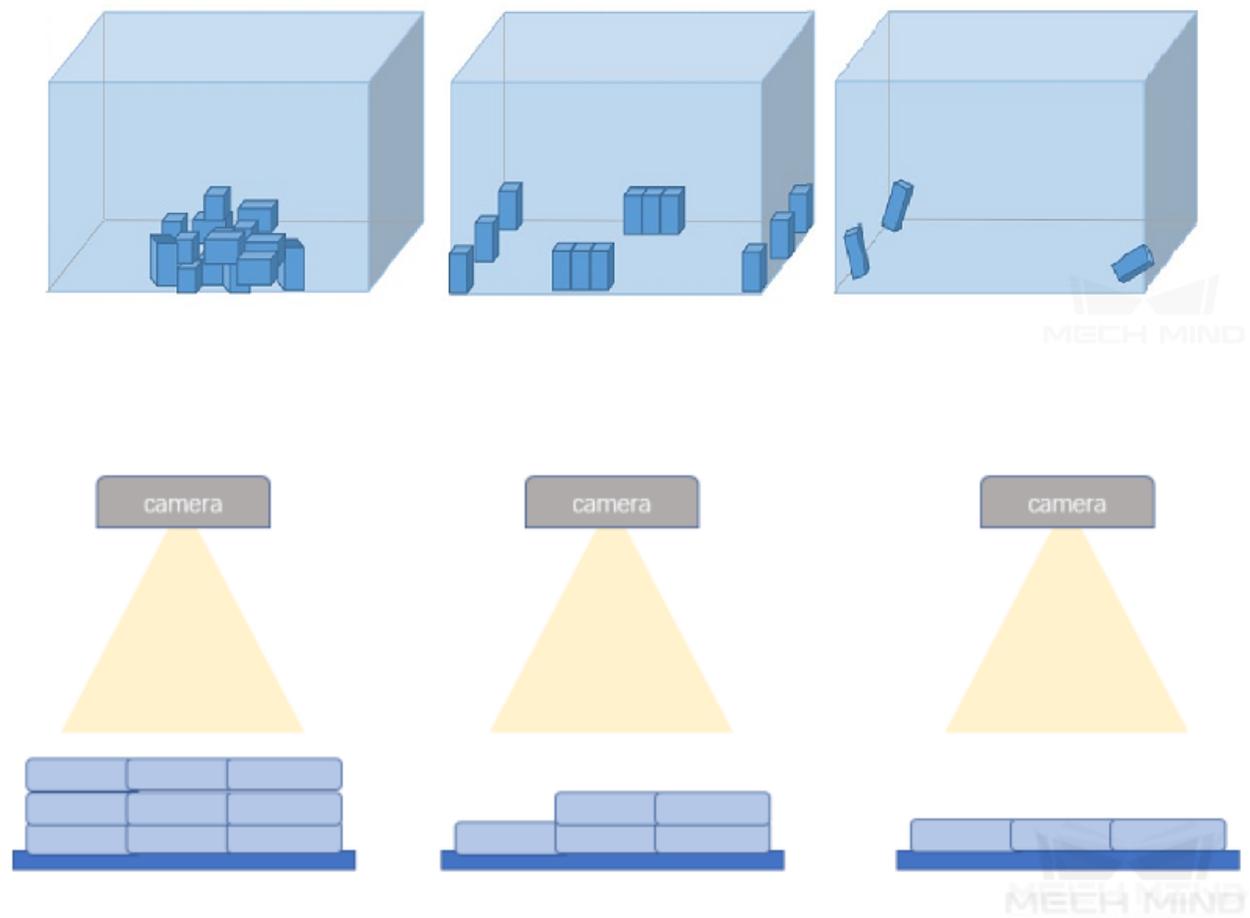
- 确保采集的数据中包含实际应用所有可能出现的 **物体朝向**。
- 确保采集的数据中包含实际应用所有可能出现的 **物体位置**。
- 确保采集的数据中包含实际应用所有可能出现的 **物体间关系**。

注意：如果少采集了某种情况，深度学习模型将会缺少对于该情况的学习，会导致模型在该情况下无法有效识别，必须根据情况增加数据样本，降低误差。

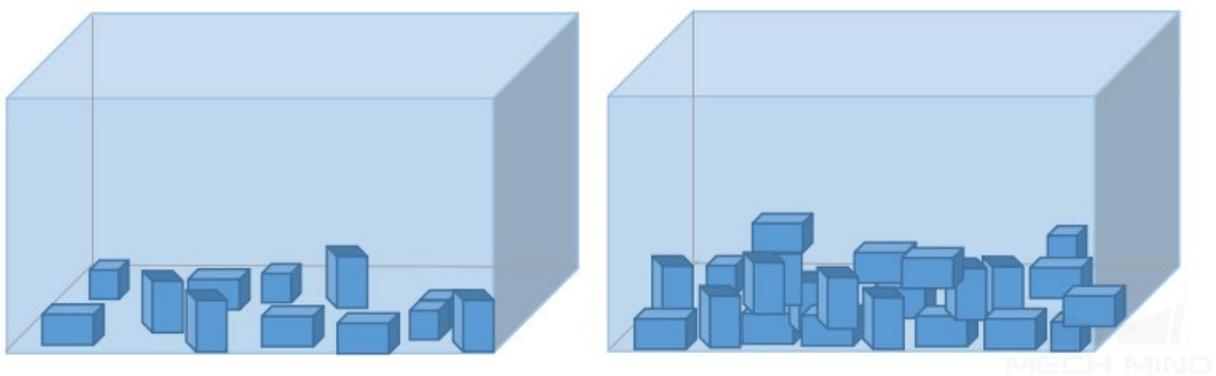
物体朝向

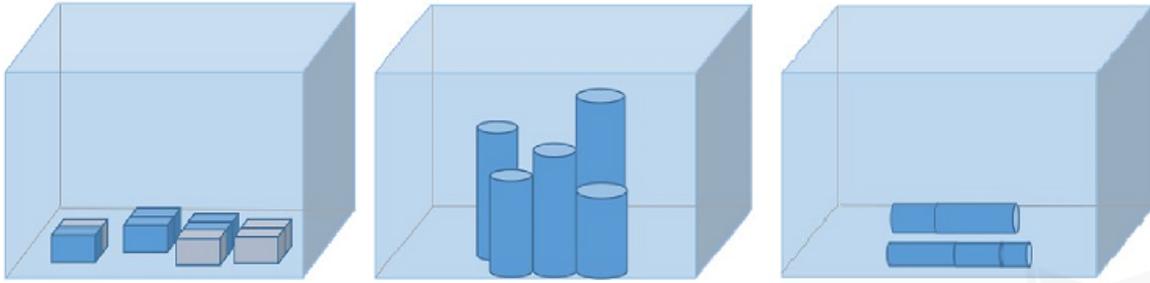


物体位置



物体间关系

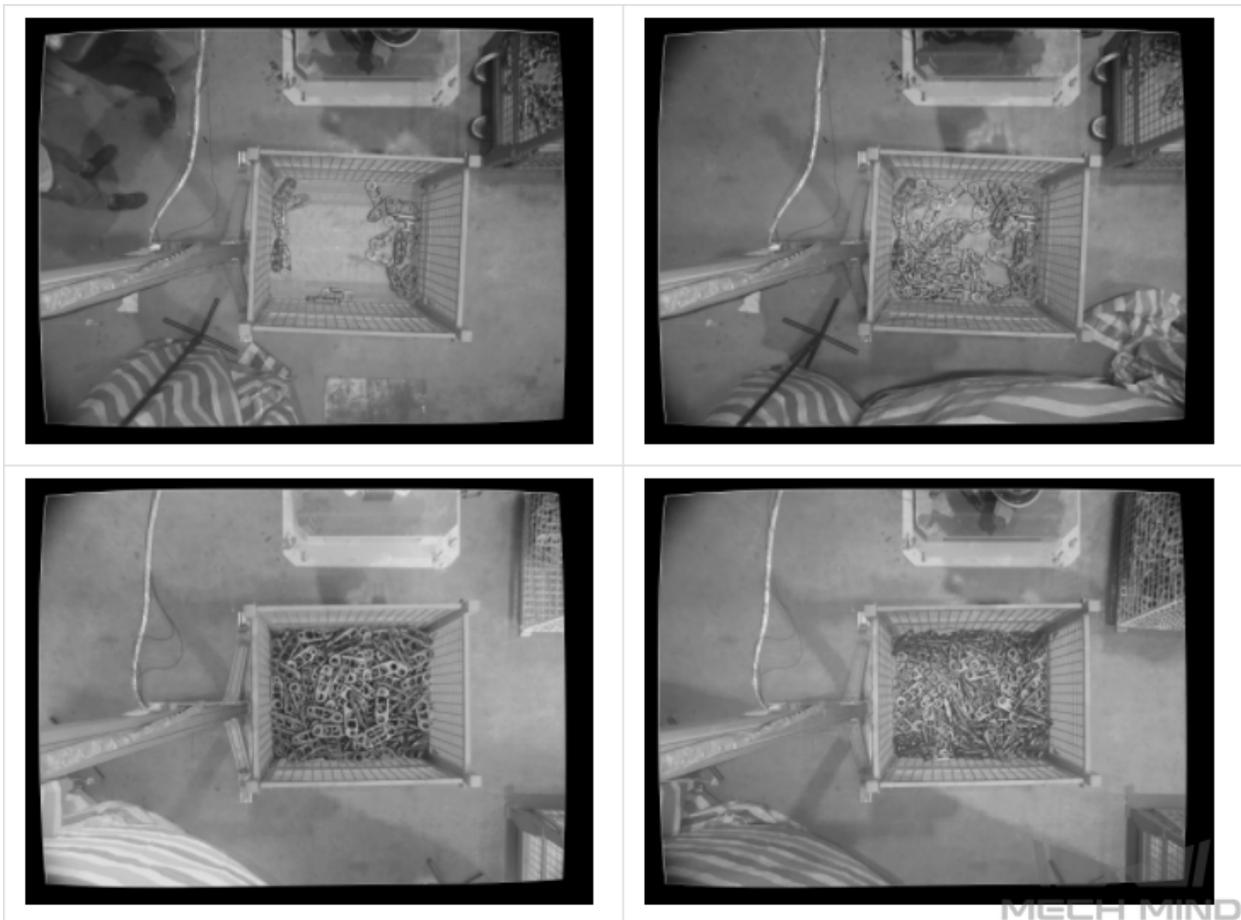




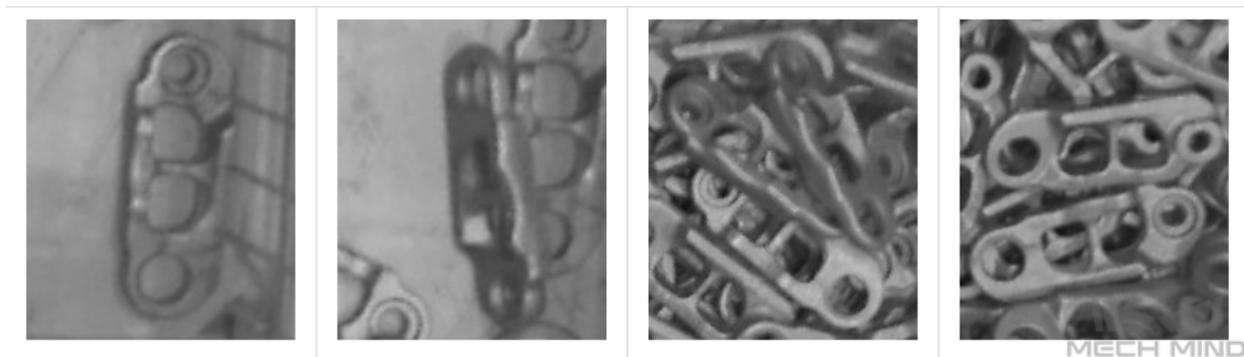
MECH MIND

数据集采集案例

1. 某金属件项目，单类物体，因此采集 50 张。物体朝向方面，可能平躺或侧立，采集时都需要考虑。物体位置方面，需要考虑位于料筐中央、四周、边角以及不同高度的情况。物体间关系方面，除堆叠外还需要考虑少量并排。实际采集的图片如下：



MECH MIND



2. 某日用品项目，7类物体混料，需要分类。采集时需考虑“单类物体多种朝向摆放”和“多类物体混合摆放”的情况，以全面地采集物体特征。单类物体采集数量 = $5 * \text{类别数量}$ ，多类物体混合摆放采集数量 = $20 * \text{类别数量}$ 。物体朝向方面，可能平躺、侧立或倾斜，采集时每个面都需要考虑。物体位置方面，需要考虑位于料筐中央、四周、边角。物体间关系方面，除堆叠外还需要考虑并排与紧密拼接情况。实际采集的图片如下：

单独摆放：



混合摆放：



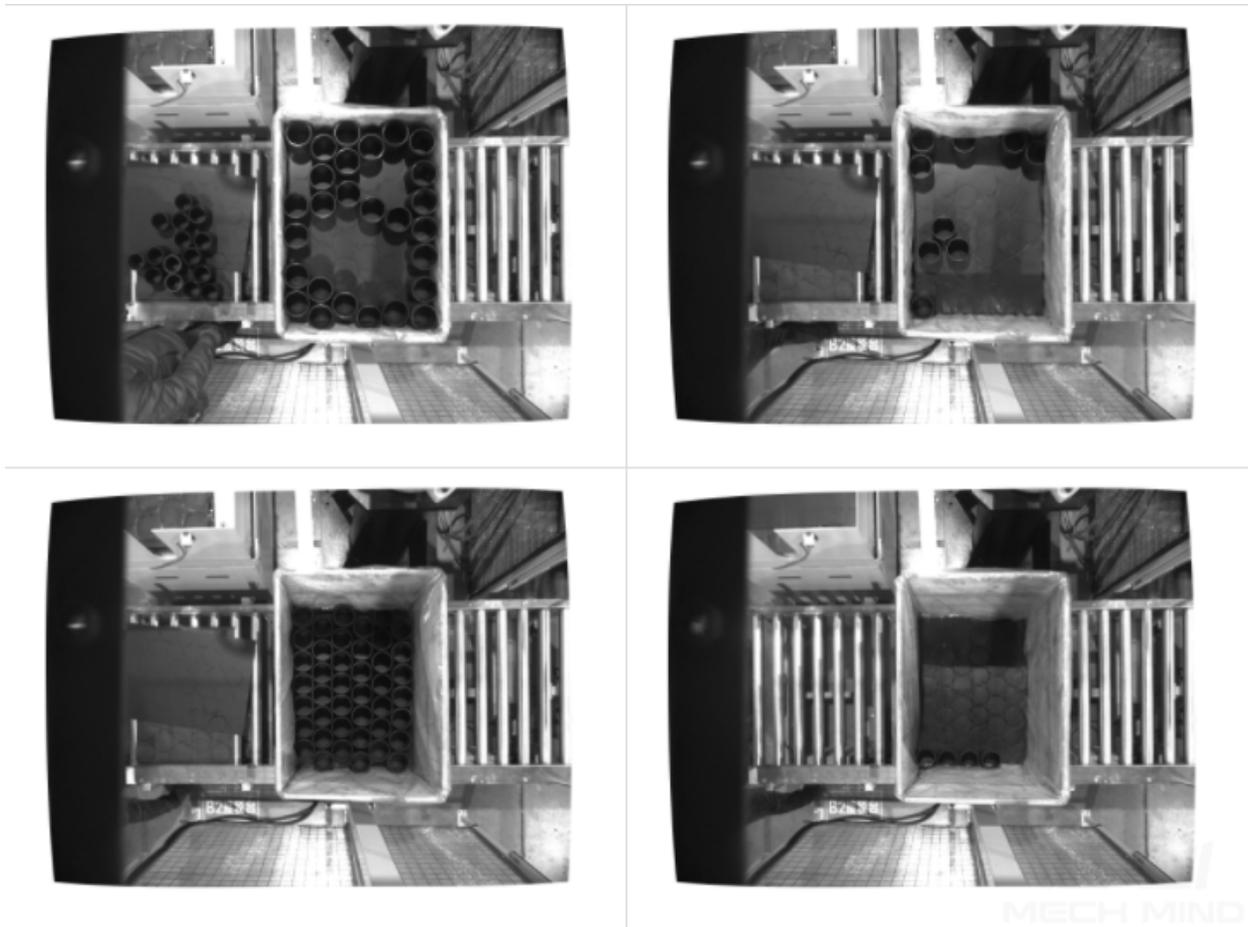
1. 某履带板项目，物体有多个型号，因此采集（30 * 型号数量）张。物体朝向方面，只需考虑正面向上摆放。物体位置方面，摆放方式单一，需包含考虑高、中、低层不同高度视野的数据。物体间关系方面，规则码放，需重点注意紧密贴合的情况。实际采集图片如下：



2. 某金属件项目，仅平铺一层，因此采集 50 张。物体朝向方面，仅平铺一层，只需考虑正面向上摆放。物体位置方面，需考虑到位于料筐中央、四周、边角。物体间关系方面，需考虑紧密贴合等，实际采集图片如下：



3. 某金属件项目，多层整齐码放，采集 30 张。物体朝向方面，只需考虑正面向上摆放。物体位置方面，需考虑到位于料筐中央、四周、边角以及高、中、低层不同高度的情况。物体间关系方面，需考虑紧密贴合等，实际采集图片如下：



选取合适的数据集

1. 训练集数量可控

对于“实例分割”模块初次建模，建议使用 30~50 张图像。数据量并不是越大越好，前期加入大量无效的数据集不利于后期的模型改进，同时还会延长训练时间。

2. 数据具有代表性

数据集图像一定要涵盖待检测目标的所有光照、颜色、尺寸等信息。

- 光照：实际存在光照变化，数据集应该包含不用光照情况下的图像。
- 颜色：工件存在不同颜色，数据集应该包含所有颜色的图像。
- 尺寸：工件存在不同尺寸，数据集应该包含所有不同大小尺寸的图像。

注意：若实际现场工件会出现旋转、缩放、或其他情况，无法采集相应图像数据集时，可以通过调整数据增强训练参数的方式来补充数据集，以确保现场所有的情况都包含在训练集内。

3. 数据占比均衡

训练集中不同类型的图像比例要均衡，否则会影响模型效果。禁止出现一种物体 20 张，另一种物体仅有 3 张的情况。

4. 数据集与终端场景保持一致

数据集的图像要与最终模型使用的终端场景保持一致，包括光照条件、工件特征、检测背景、视野大小等。

8.3.3 确保标注质量

确认标注方式

1. **标注上表面轮廓：**适用于平放的规则物体，如纸箱、药盒、矩形工件等。通过上表面轮廓计算抓取点，用户只需要标注上表面矩形框。

错误示例：标注外轮廓

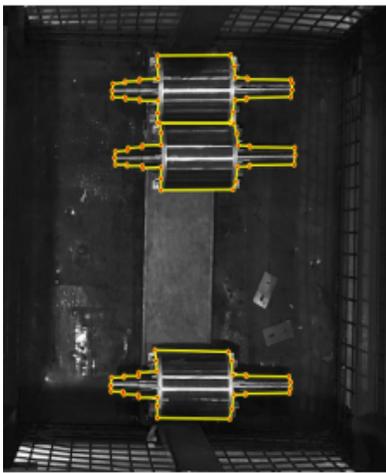


正确示例：正确标注



2. **标注完整外轮廓：**适用于麻袋、各类工件等，标注完整外轮廓是通用的标注方式。

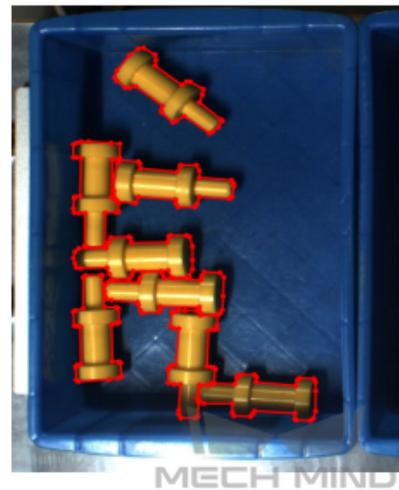
正确示例：标注外轮廓



正确示例：标注外轮廓



正确示例：标注外轮廓



3. 特殊情况：适用于需要配合夹具或抓取方式的特殊情况。

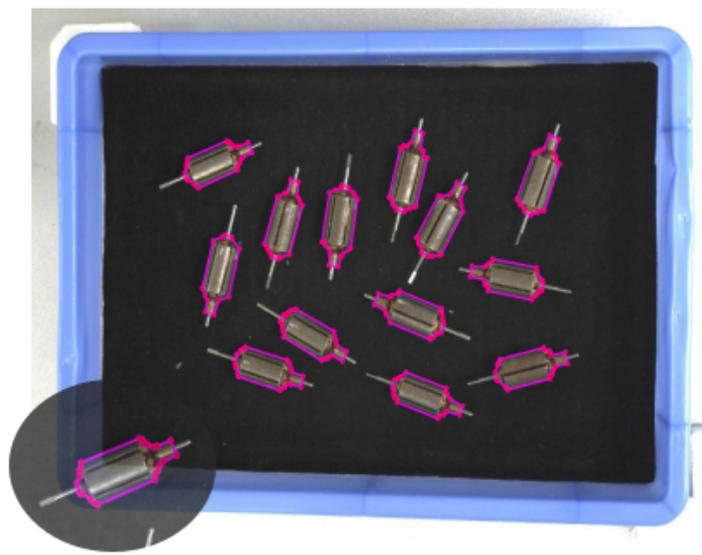
- 需要保证吸盘与瓶口完全贴合（对精度要求很高），只需标注瓶口轮廓。

正确示例：标注瓶口



- 转子抓取需要区分方向，只标注能明确区分方向的中间部分，不标注两端的细杆。

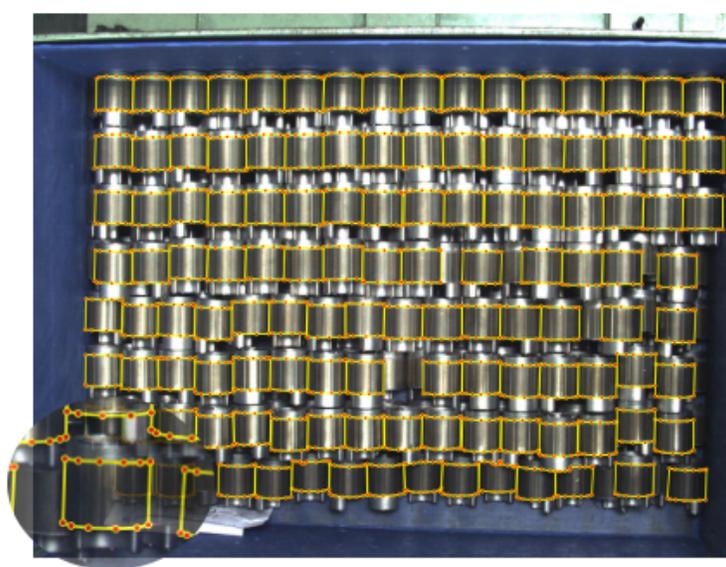
正确示例：标注转子中间位置



MECH MIND

- 需要保证吸取位置在金属件中间部分，所以只标注工件中间金属部分，不需要标注两端。

正确示例：标注中间位置



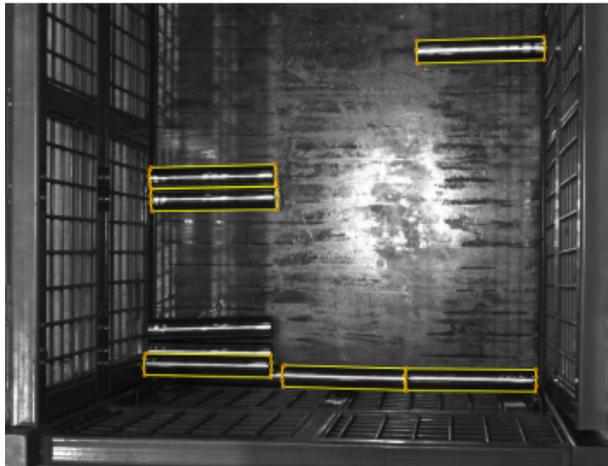
MECH MIND

检查标注质量

标注质量应从完备性、正确性、一致性、精确性几个方面考虑：

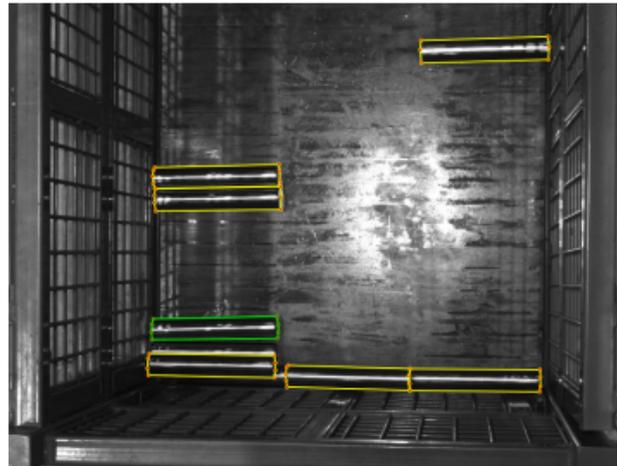
1. **完备性**：标注所有符合规则的物体，禁止出现遗漏标注的情况。

错误示例：漏标注



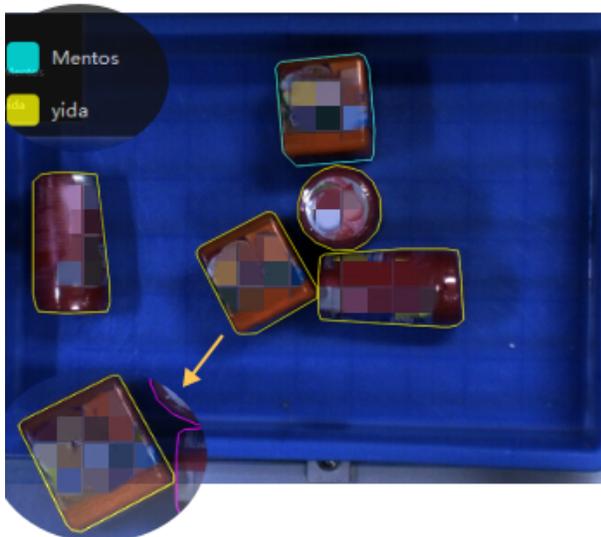
标注提示：要求标注所有完整的钢棒时，不应遗漏某一个

正确示例：正确标注

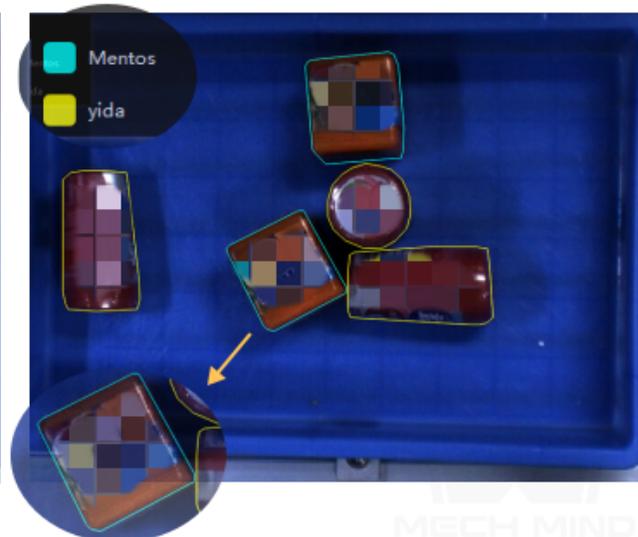


2. **正确性**：确保物体与所属的标签正确对应，禁止出现物体与标签不一致的情况。

错误示例：标签不对应, Mentos被标注为yida

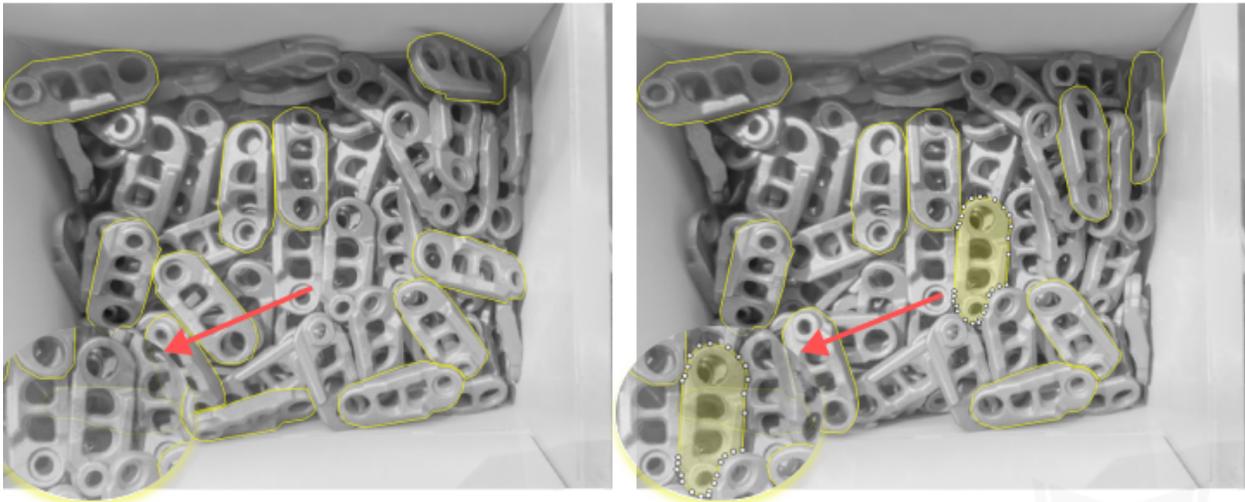


正确示例：正确标注



3. **一致性**：所有数据应遵循同一标注规则。例如标注规则规定只标注整体露出 85% 以上的物体，则应该标注所有符合该规则的物体，避免出现某一物体标注了但另一类似物体没有标注的情况。

错误示例：相邻两张图片中相同姿态的金属件标注不一致



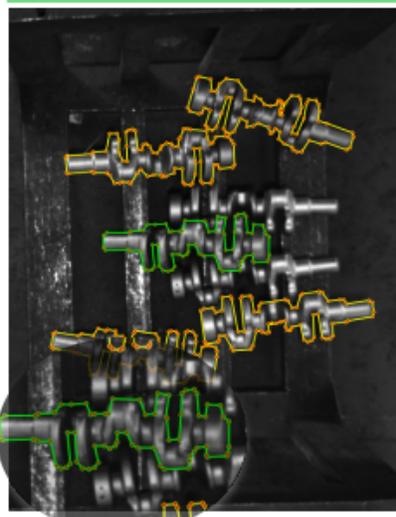
标注提示：标注不一致时，需检查不同处是否符合标注要求，符合则补标漏标物体，反之则删除多标注的标签

4. **精确性**：标注轮廓紧密贴合目标物体边缘，禁止出现少标注或多标注的情况。

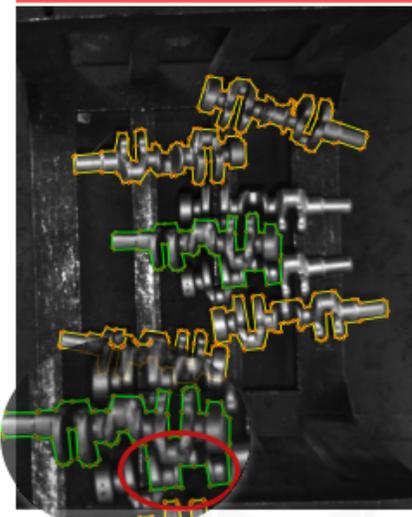
错误示例：少标注



正确示例：正确标注



错误示例：多标注



标注提示：要求标注所有完整的曲轴时，不应遗漏或多标注某个部位

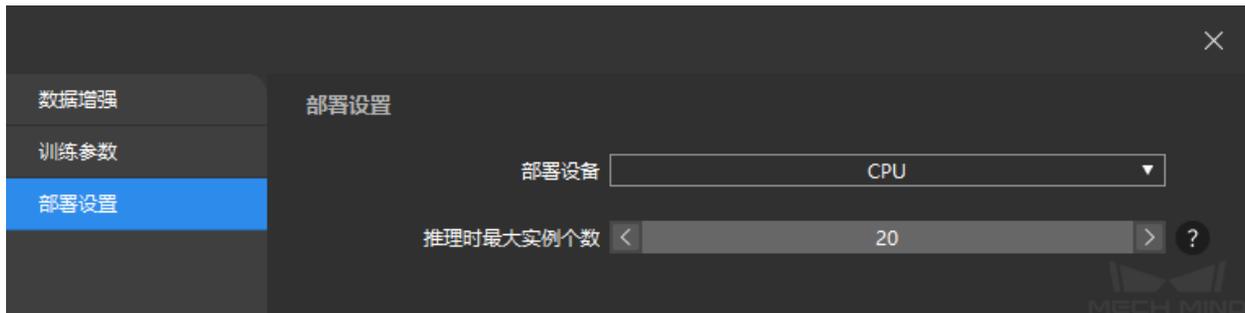
8.4 CPU 和 GPU 版本模型的训练及部署

8.4.1 CPU 版本模型训练与部署

模型部署在 CPU 设备时，训练参数中推荐将 模型类型 选择为 轻量（推荐 CPU 部署）。



将 部署设备 选择为 CPU。



8.4.2 GPU 版本模型训练与部署

模型部署在 GPU 设备时，训练参数中推荐将 模型类型 选择为 标准（推荐 GPU 部署）。



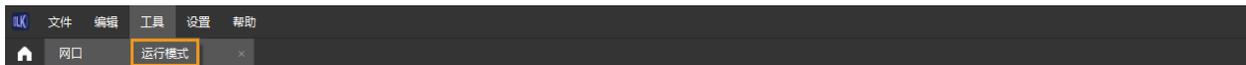
模型部署在 GPU 设备时，将 **部署设备** 选择为 **GPU**。

- **dlkmo**: 模型需要优化大约 5 分钟，适用于训练模型与部署模型使用的设备 GPU 显卡型号不同的情况。
- **dlkmt**: 模型无需优化可以直接导出使用，仅支持训练模型与部署模型使用的设备 GPU 显卡型号相同的情况。



如何进行批量数据集测试

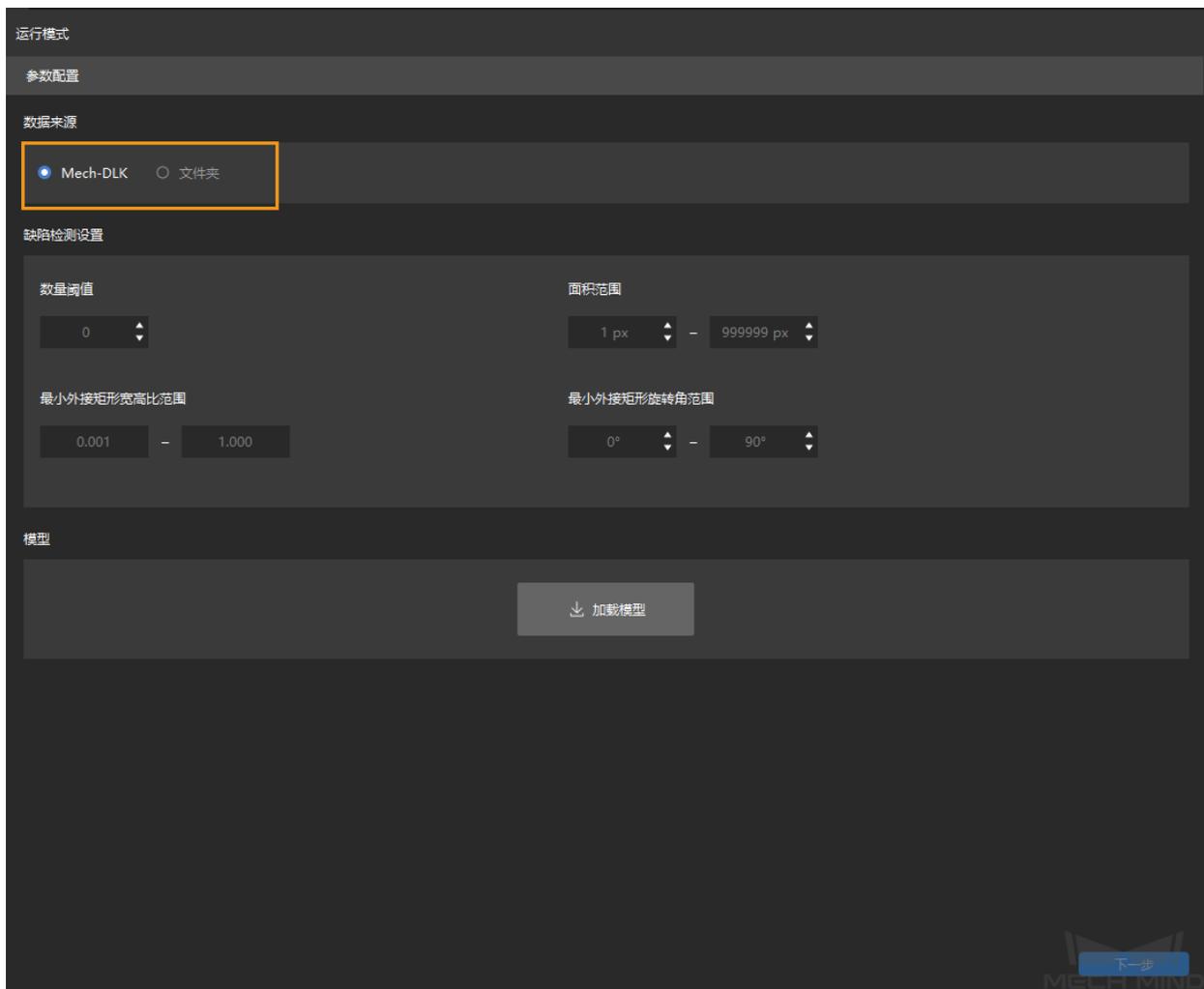
通过 **运行模式** 可批量测试数据。该功能只支持在“缺陷分割”和“图像分类”模块下使用。模型训练完成后，点击工具，打开 **运行模式** 可以导入大量新的数据集进行测试，并进行人工复判；验证结束后，可以输出报表，报表内包含准确率、过检率、漏检率等信息。



1. **选择数据来源：**若需要使用运行模式进行模型效果演示，数据来源选择 **Mech-DLK**；若需要读取大量新的数据集进行测试，数据来源选择 **文件夹**。

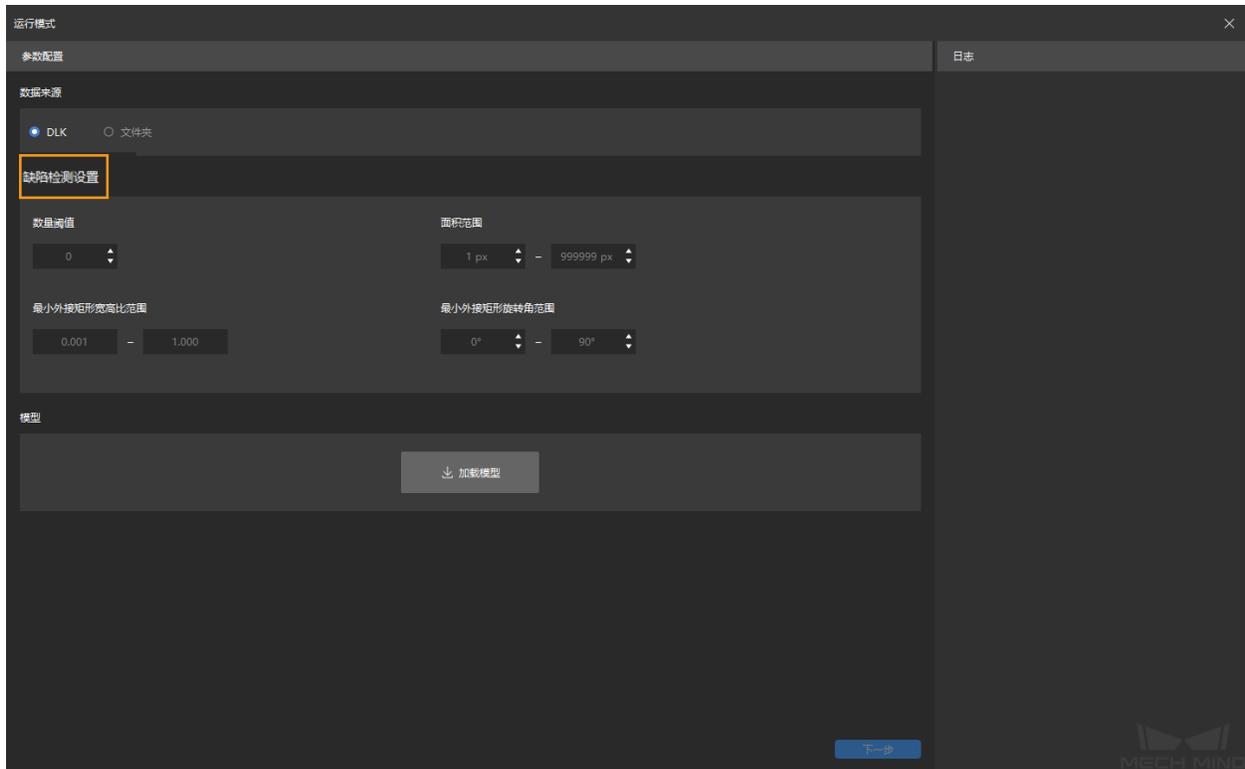
Mech-DLK：直接导入当前工程内的全部图像数据集。

文件夹：选择图像文件夹路径后，将文件夹中的所有图像作为一个新数据集导入。加载的新数据将独立于软件内的原始数据集。

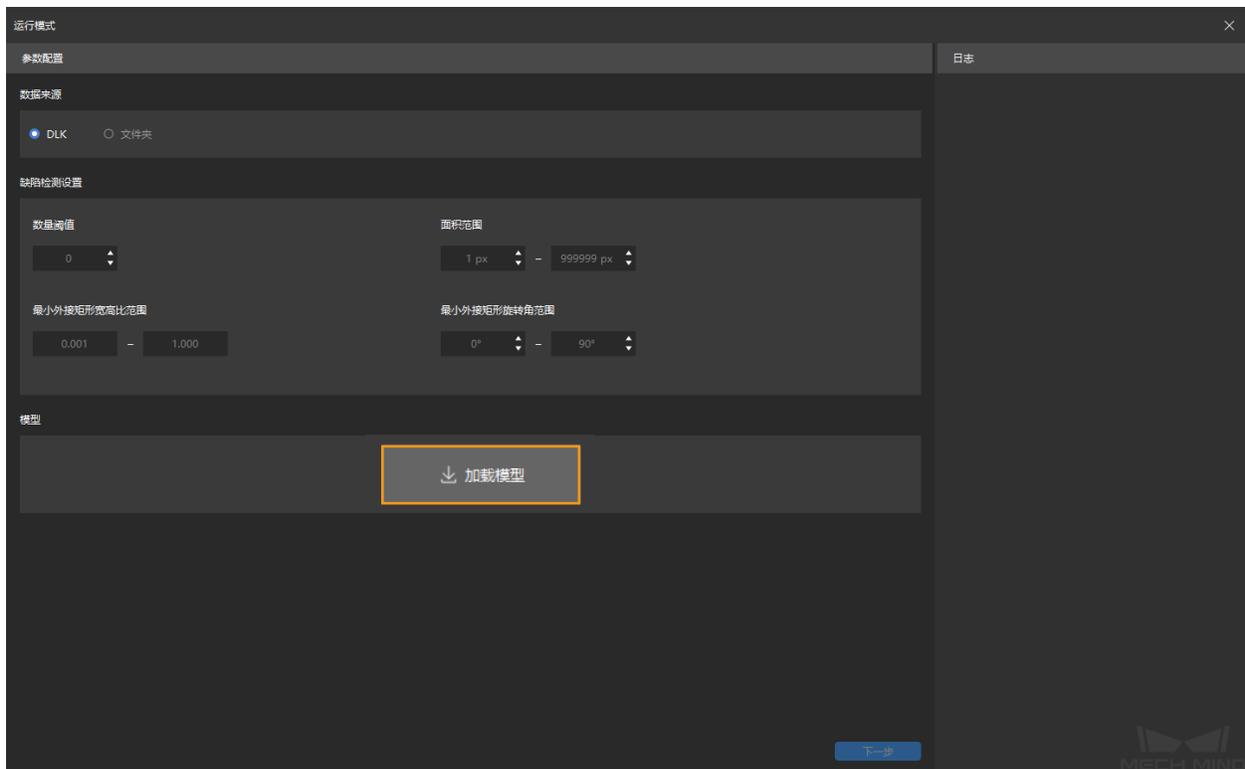


提示: 在使用运行模式之前, 如果除了当前工程的训练集和验证集, 还导入了新的图像数据到数据栏 (工程数据集), 那么当运行模式的数据来源为 *Mech-DLK* 时, 会一并读取新导入的图像。

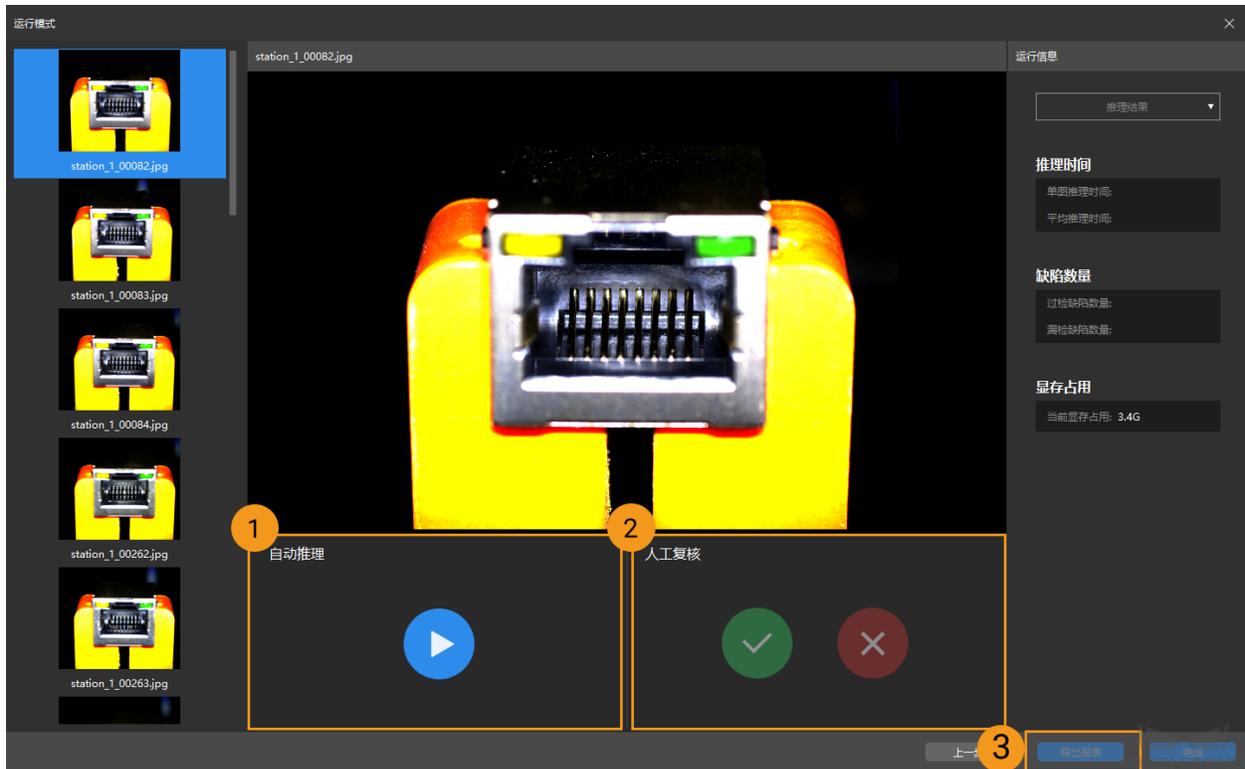
2. **缺陷检测设置:** 请根据缺陷判定标准设置各个参数, 软件中会同步开启缺陷过滤相关选项。缺陷检测参数与“缺陷分割”模块中的缺陷判定规则互相独立。



3. **加载模型**: 点击 **加载模型**, 加载成功后点击 **下一步** 进入推理界面。



4. **推理并导出报表:** 点击 **自动推理** 栏的 **启动** 按钮开始推理。自动推理完成后, 复核验证结果是否准确, 并在人工复核栏点击对应的按钮, 全部复核结束后点击 **导出报表** 查看准确率、过检率、漏检率等。



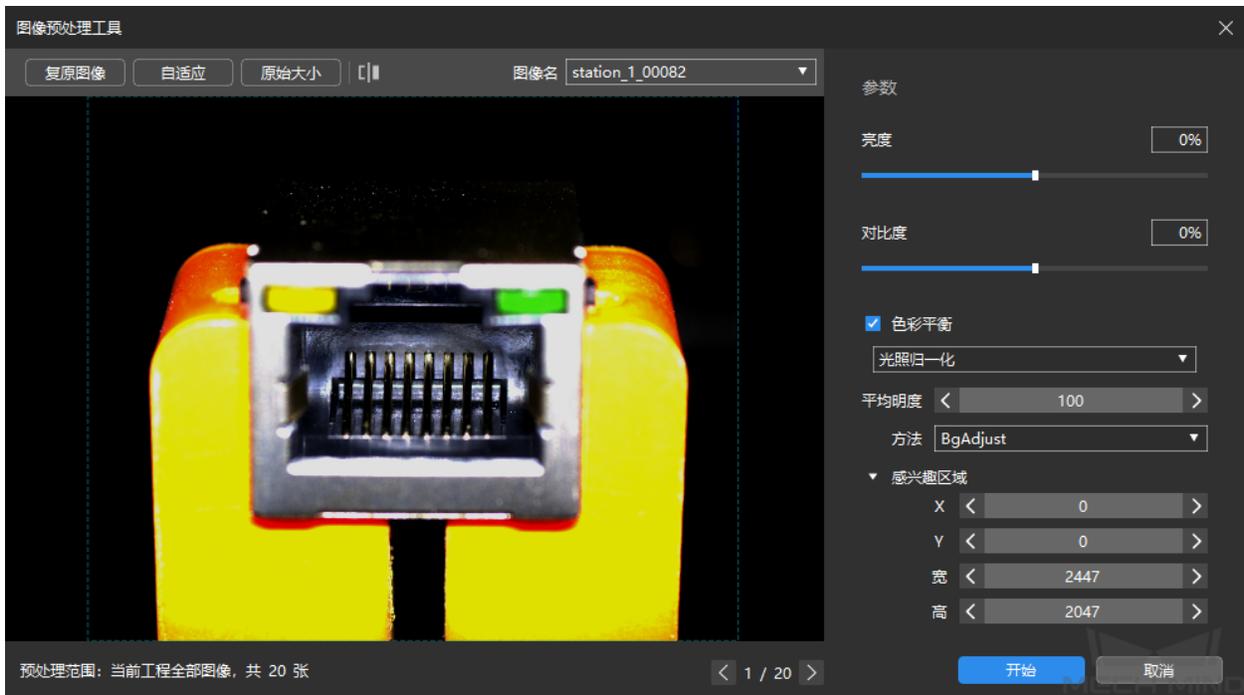
如何提升标注质量和效率

通过改善图像质量、选用合适的标注工具等方式可以提升标注质量和效率。

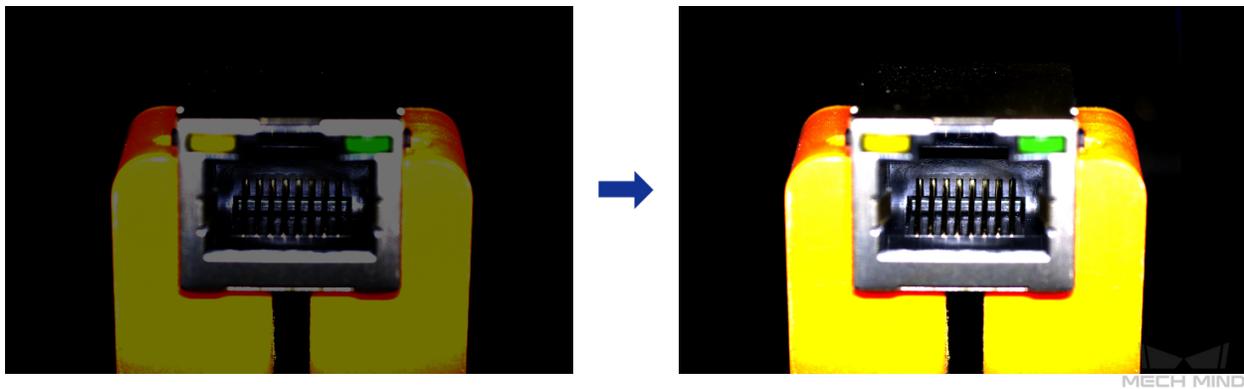
- 图像预处理
- 智能标注工具
- 印章工具
- 图像掩膜工具

10.1 图像预处理

相机拍摄的原始图像因光照或其他因素存在图像偏暗、物体特征不明显等问题时，为提高人工标注效率，可以使用  图像预处理工具。通过调整亮度、对比度或色彩平衡等参数，对图像进行预处理操作。



预处理前后的图像对比如下:



提示: 数据集经过图像预处理后会直接发生改变, 并作为训练集/验证集/测试集参与模型训练和优化。

10.2 智能标注工具

在“缺陷分割”、“实例分割”和“目标检测”模块下，图像中多个物体颜色差异较大且散乱放置时，可以使用智能标注工具对图像中物体进行智能化标注。

- **智能标注：** 点击  智能标注工具按钮，鼠标左键点击待标注的物体，会出现绿点并形成标注；标注轮廓较小时，可以多次点击左键进行优化。点击鼠标右键出现红点，可以删除多余的标注。标注结束后点击 应用 即可显示标注结果。



小技巧： 图像中物体间颜色差异较大且轮廓分明时，建议同时标注多个物体后再点击 应用；若物体间较贴合，建议每次标注单个物体。

- **调整边缘：** 标注轮廓不够精细时，可以使用选择工具 ，选中需要调整的标注框，通过拖拽或者增删边缘的锚点的方式来优化标注结果（左键增加、右键删除），保证标注区域的轮廓与物体轮廓尽可能贴合。

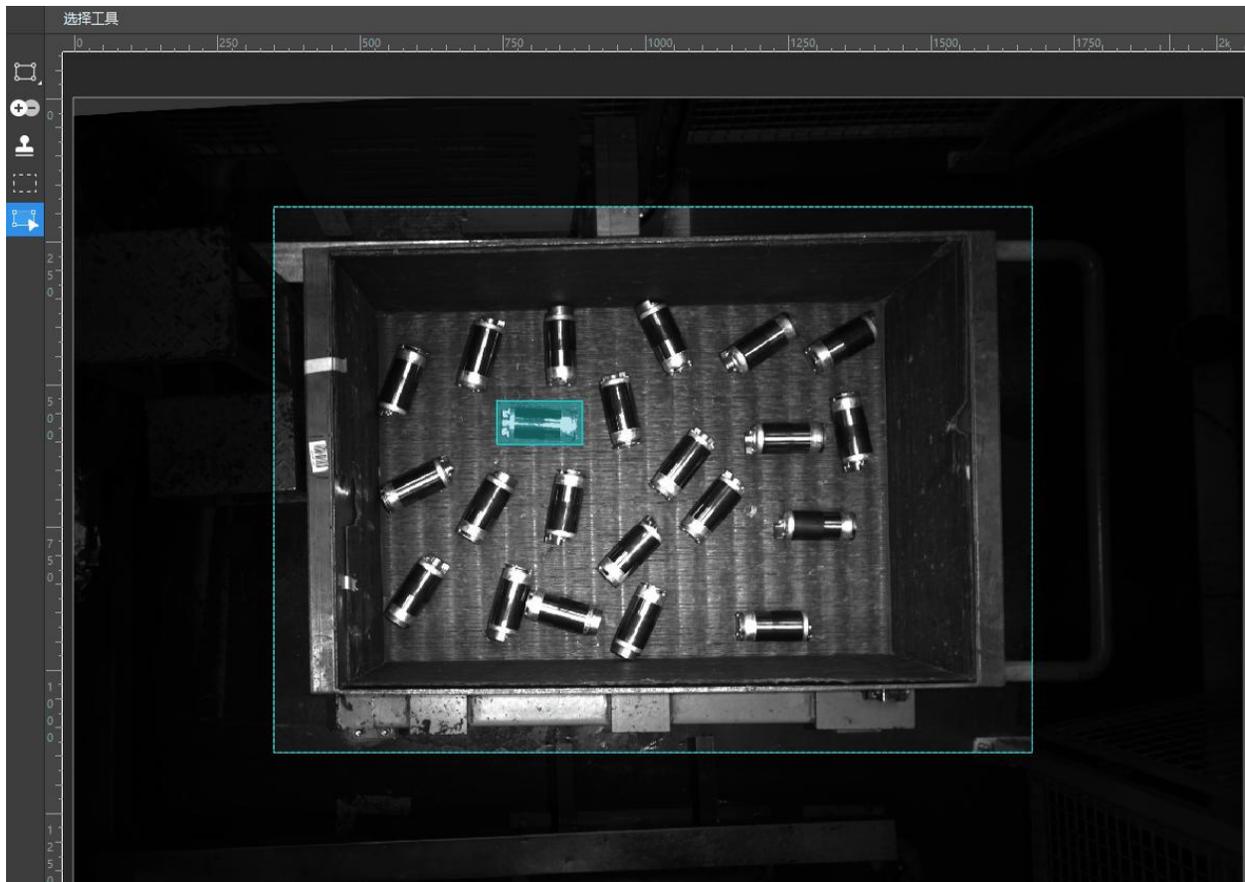


提示：图像中多个物体颜色相近且堆叠摆放、物体轮廓复杂或工件反光等困难场景下，建议选择其他合适的工具进行标注。

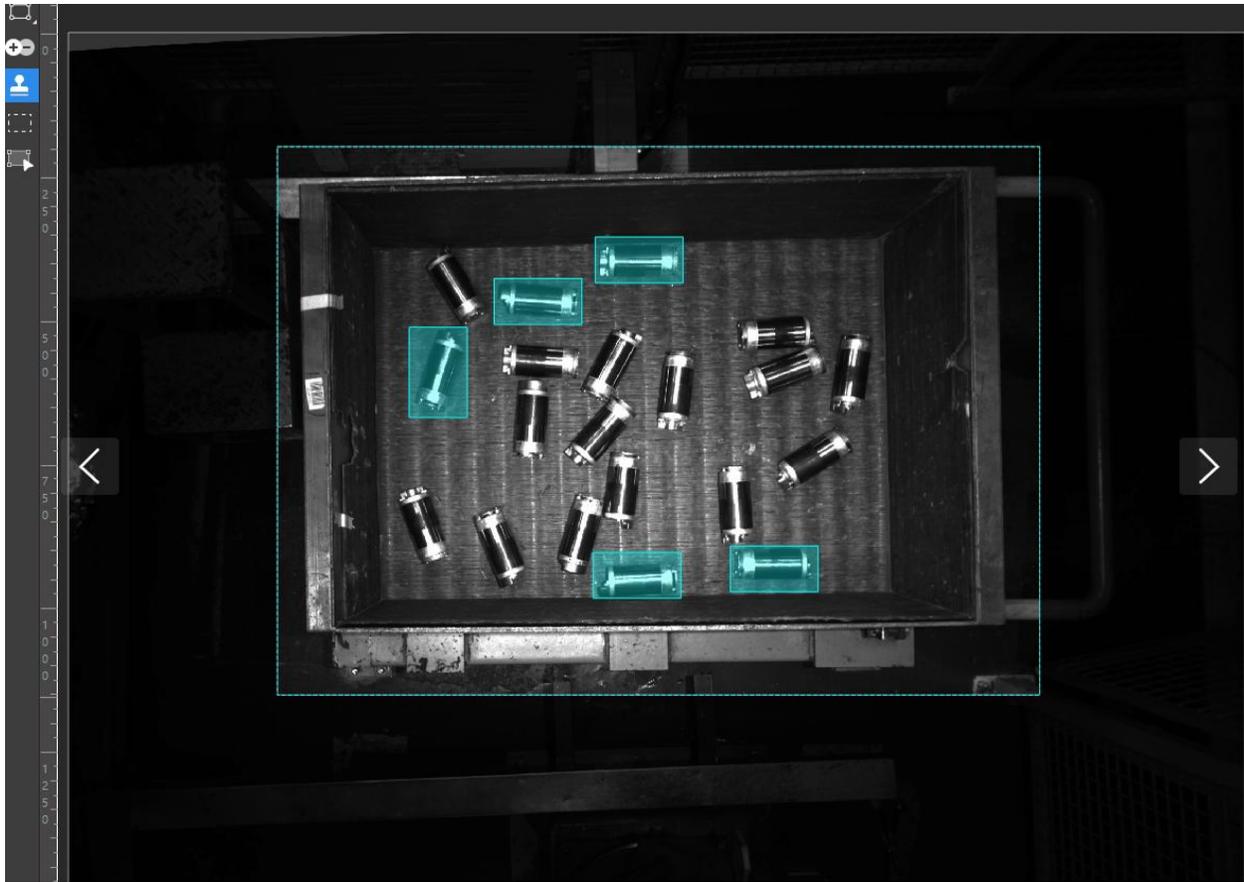
10.3 印章工具

在“实例分割”和“目标检测”模块下，可选择印章工具可以将已有标注设为模板。设置完成后，可以使用该模板快速标注轮廓及姿态相同的物体。该工具适用于图像中有多个摆放整齐的同类别物体的场景，可以提高标注效率。

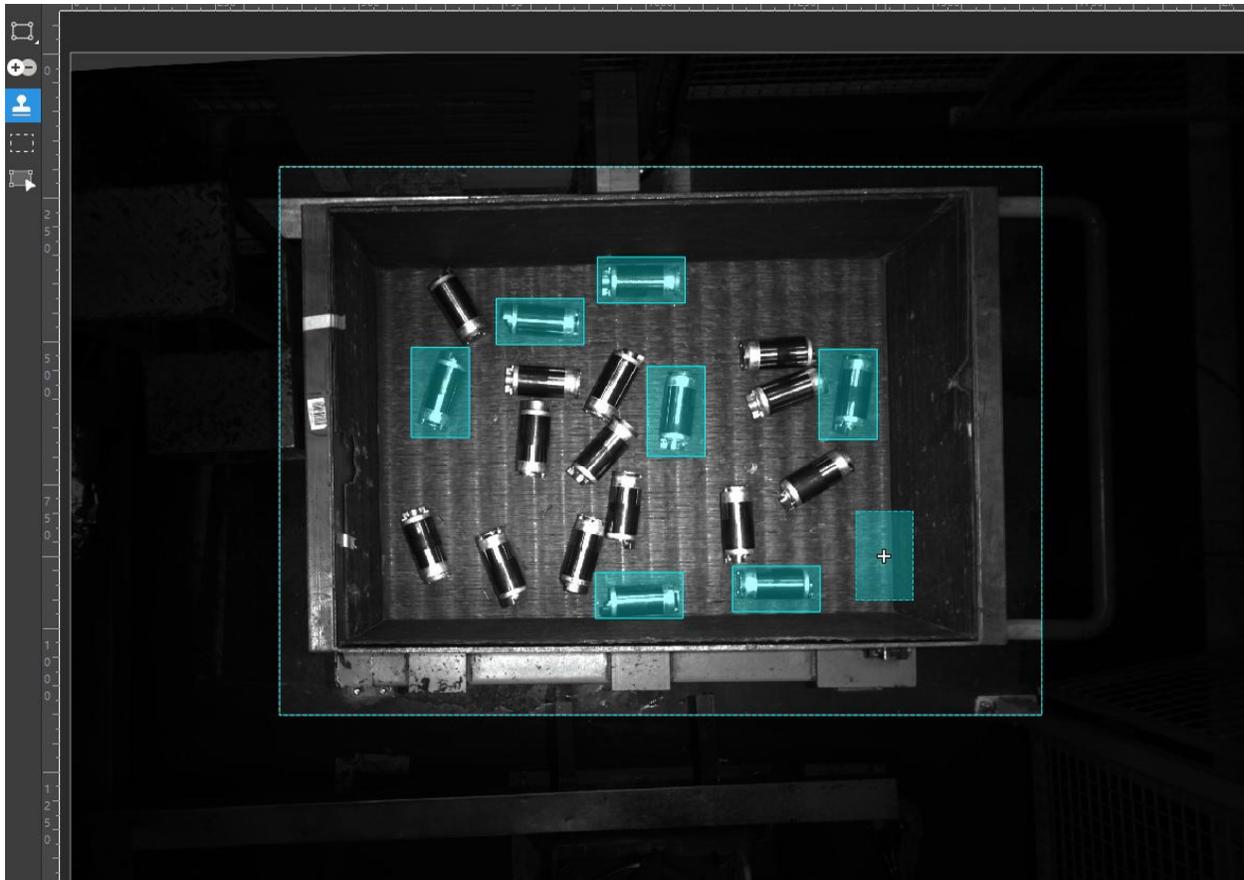
- **将标注设为模板：**点击工具栏的  印章工具按钮，右键点击需要设为印章的标注，选择 将当前标注设为印章，模板即设置成功。



- 更换模板样式：右键点击新的标注，选择 将当前标注设为印章。



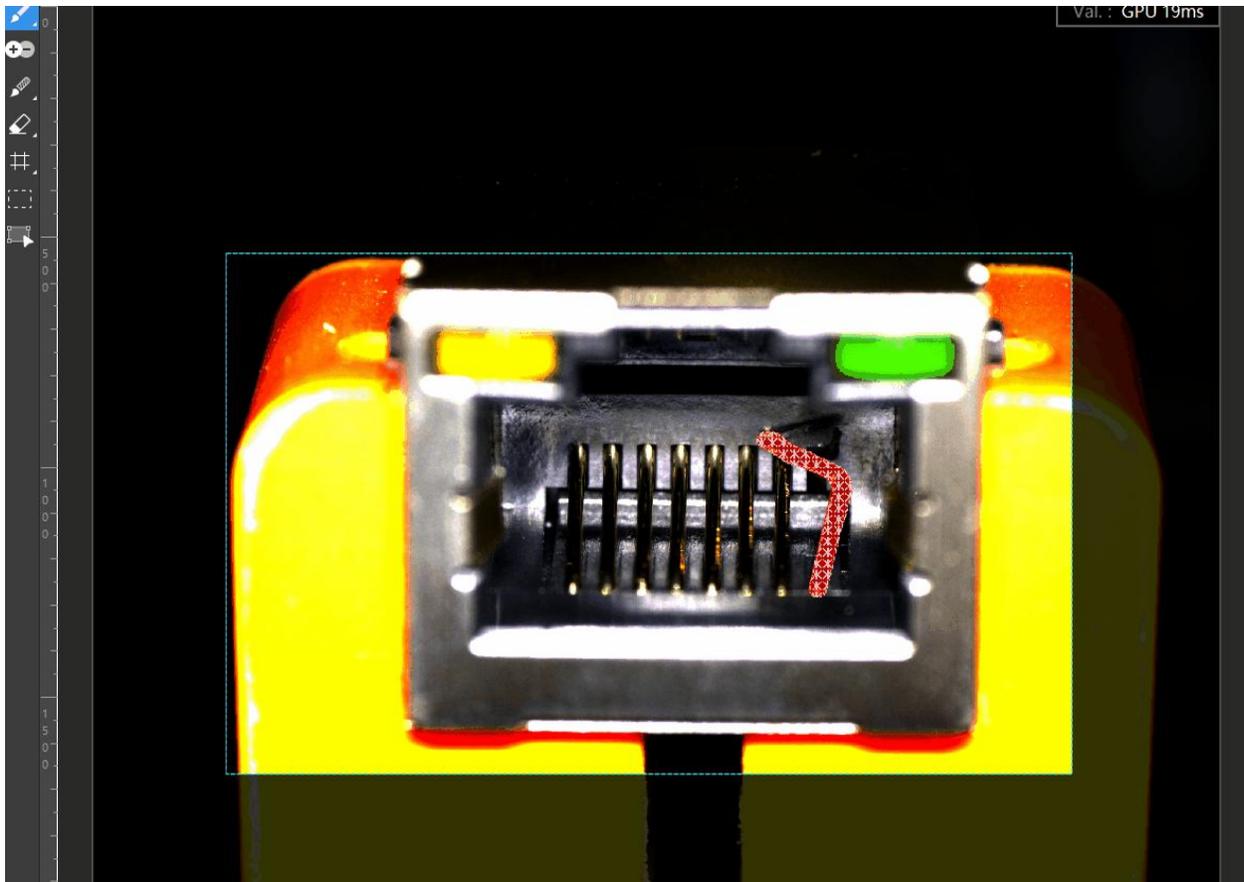
- **调整标注：**如果复制的标签模板无法完全符合需求，可以在标注结束后切换到  选择模式。选中需要调整的标签，拖动锚点调整轮廓。若需要调整角度，可以拖动标签上方的  旋转按钮进行调整。



提示：标注结果为多边形轮廓时，右键点击任意锚点可以删除该锚点。将光标置于两锚点之间，左键点击线段可新增锚点。

10.4 图像掩膜工具

若图像中存在一些不相关且干扰训练的部分，可用掩膜工具  遮盖。被遮盖的部分不参与模型训练。例如，图像中物体特征不符合用户缺陷判定标准，但和缺陷形态相似的，需要遮盖容易混淆的区域。

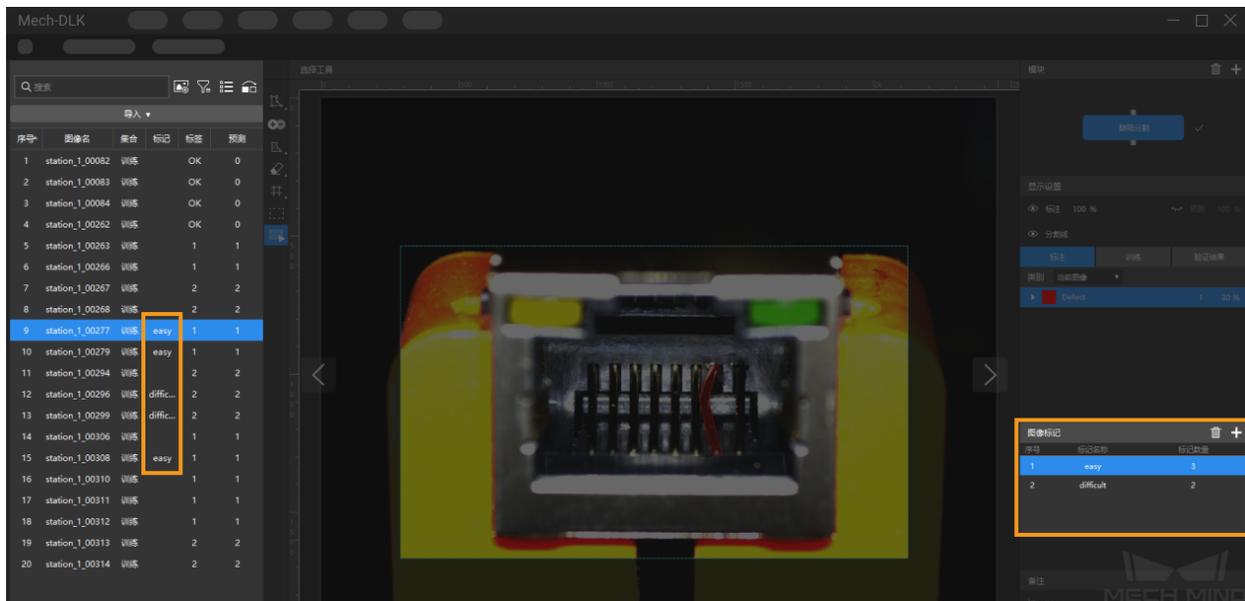


提示：用户可以根据实际情况选择合适的掩膜工具，使用掩膜工具遮盖的区域不会参与模型训练。

如何标记和筛选图像

11.1 图像标记

图像标记管理区域位于软件界面右下角。可以标记单张图像，或选中多张图像进行标记。从而根据图像的不同用途，集中管理大量的图像数据。



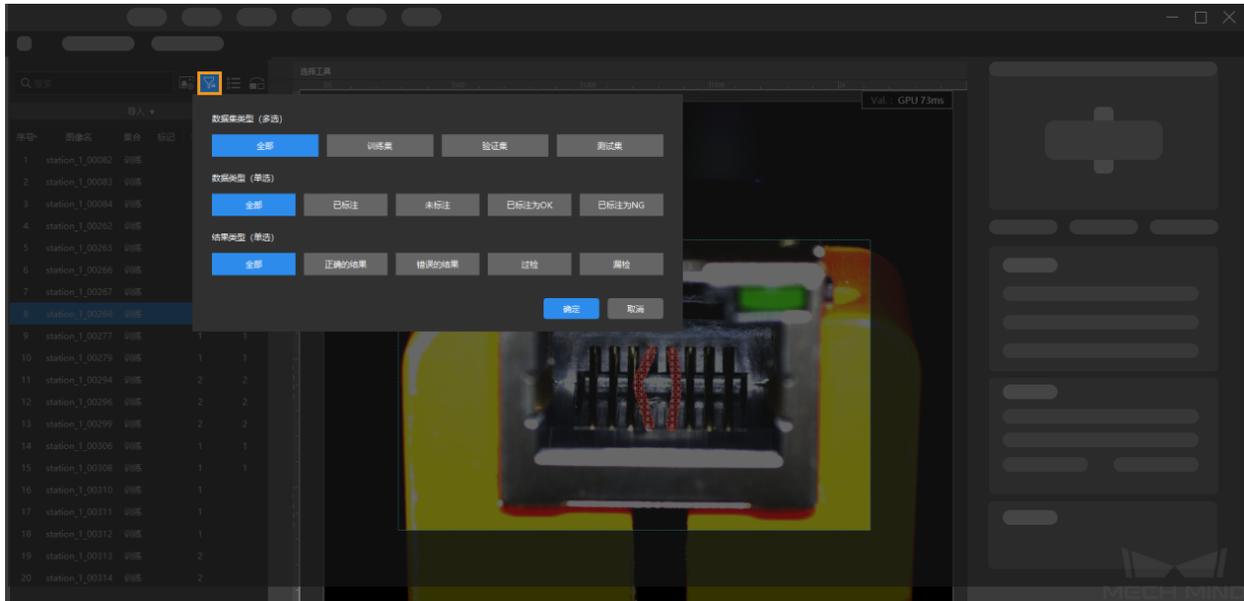
典型使用场景：

- 使用 缺陷分割模块时

- 如果图像中物体有多种类别的缺陷，需要根据缺陷特征进行划分，用户可以标记图像缺陷类别，便于分组查看图像。
 - 如果无法判定当前图像是否存在缺陷，用户可以先标记出该图像，便于后续对图像进行二次判定。
- 验证模型后，用户可以标记出验证结果不理想的图像，便于将标记后的图像批量导出用于优化模型。

11.2 图像筛选

用户可以根据需求选择筛选条件，快速定位数据集。下图以缺陷分割为例：



典型使用场景：

- 查看数据集划分结果。数据集划分完成后，用户可以通过筛选数据集类型，查看各个数据集内容，检验划分结果是否需要调整。
- 检查数据标注结果。数据标注过程中，用户可以通过筛选数据类型查看标注进度和标注结果。
- 检验模型预测结果。验证模型后，用户可以通过筛选结果类型查看验证结果。例如，当缺陷分割模型预测错误时，可以通过筛选 过检或 漏检图像集合，快速分析是否因标注错误导致模型预测错误。

如何解决困难问题

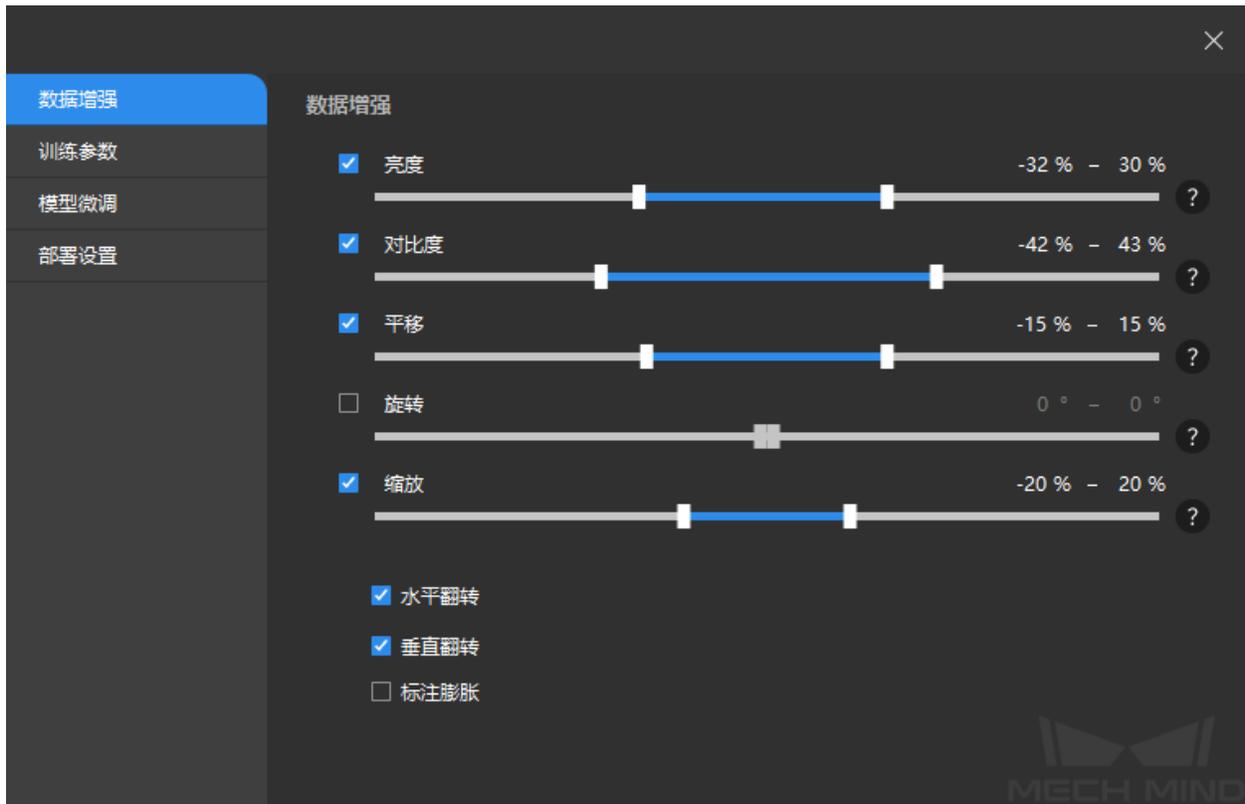
通过以下方法可以提升训练、部署的速度和模型的精度。

- 如何丰富数据集种类
- 如何提高模型训练速度和精度
- 如何进行模型微调
- 如何提高模型推理速度和精度

12.1 如何丰富数据集种类

训练模型的数据集需要尽量包含实际可能出现的所有情况，若现场不具备相应的采集条件，可以针对缺失的情况在训练前调整 **数据增强** 参数，在原有数据的基础上丰富训练数据集。

12.1.1 调整数据增强参数



注意：必须确保倍增的图像数据与现场实际情况一致。如果现场不会出现旋转的情况，那么不需要调整这一参数，否则可能会影响模型效果。

亮度 图像的明暗程度。现场光照变化较大时，可以通过调整 **亮度** 范围倍增不同光照状态下的图像数据集。

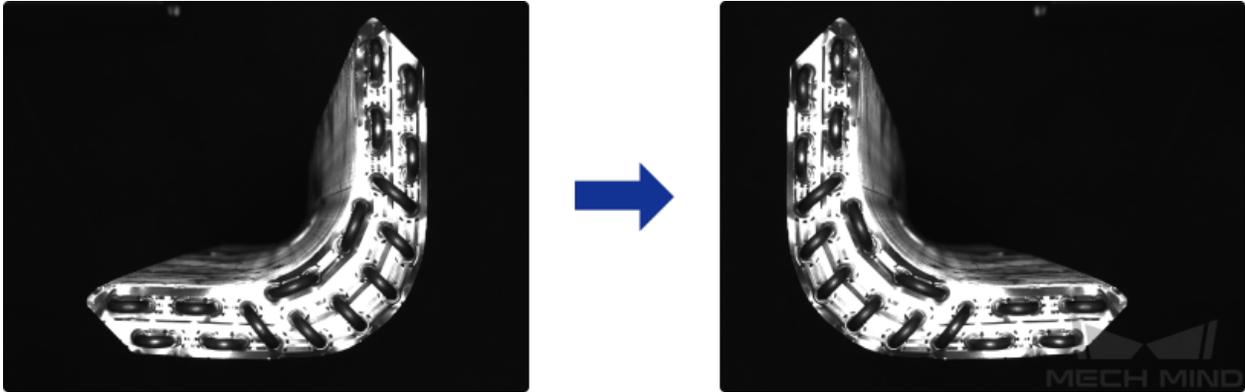
对比度 图像灰度反差的大小。目标物体和背景差异较小时，适当调整 **对比度** 范围，可以使目标物体特征更明显。

平移 将图像所有的像素坐标分别加上指定的水平偏移量和垂直偏移量。现场物体（料筐、托盘等）位置移动范围较大时，可以通过调整 **平移** 参数，随机倍增横向、纵向平移的数据集。

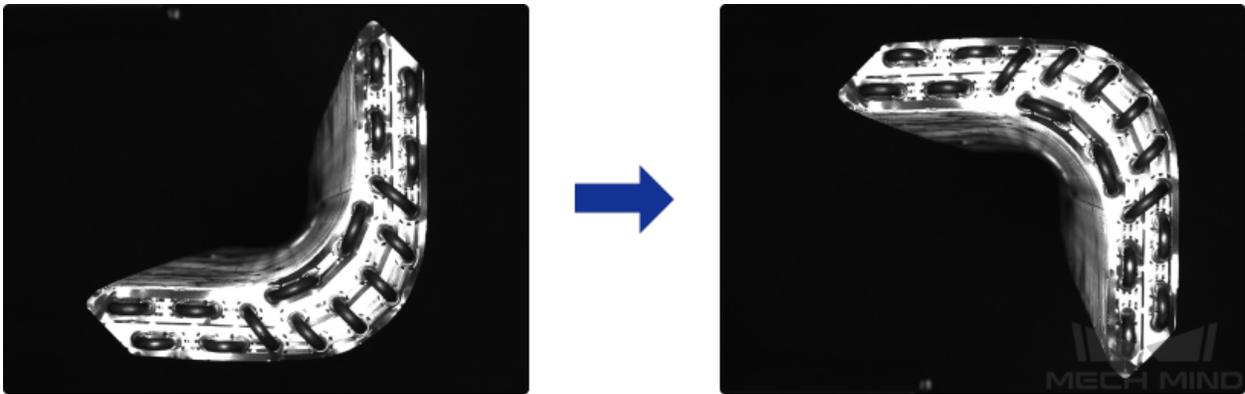
旋转 指图像以某一点为中心旋转一定的角度，形成一张新的图像。一般情况下保持默认参数设定即可满足需求。物体摆放角度变化较大时，通过调整 **旋转** 参数，可以随机倍增多种角度的图像数据集。

缩放 将图像按一定比例缩小或放大。物体摆放高度差异较大时，通过调整 **缩放** 参数，可以随机倍增不同缩放程度图像数据集。

水平翻转 图像左右翻转 180°。若待识别物体存在左右对称的情况，可以开启 **水平翻转**。



垂直翻转 图像上下翻转 180°。若待检测物体存在上下对称的情况，可以开启 **垂直翻转**。



标注膨胀 按一定比例扩大图像中标注的缺陷区域。只支持在“缺陷分割”模块使用，一般情况下不需要勾选，若图像标注的缺陷较小可以勾选 **标注膨胀**，避免因图像缩放导致缺陷太小而影响训练的情况。

12.2 如何提高模型训练速度和精度

默认参数设定可以满足大部分需求，如果需要提高模型训练速度或模型精度时，可以通过调整 **训练参数** 重新训练模型以满足需求。

12.2.1 调整训练参数



输入图像尺寸 训练时输入神经网络的图像宽高（单位：像素）。建议使用默认设定；若图像中的物体或标注缺陷区域较小时，需要适当调大 **输入图像尺寸**。图像尺寸越大，模型精度越高，但是训练速度越低。

批量大小 神经网络训练一次选取的样本数量。建议使用默认设定；若需要提高训练速度时，可以适当调大 **批量大小** 参数，参数设置太大会导致显存占用变高。

模型类型 “缺陷分割” 模块：

- **标准**：一般情况下建议使用 **标准** 模式。
- **增强**：模型效果不好或精度要求较高时可以选择 **增强** 模式，该模式会导致训练速度变慢。

“实例分割” 模块：

- **标准（推荐 GPU 部署）**：模型部署在 GPU 设备时选择此选项。
- **轻量（推荐 CPU 部署）**：模型部署在 CPU 设备时选择此选项。

评估间隔轮次 模型训练时，每评估一次间隔的轮次数量。建议使用默认设定，调大 **间隔轮次** 参数可以提升训练速度，参数值越大训练越快；参数值越小训练越慢，但有助于挑选最优模型。

总轮次 模型训练的总轮次。建议使用默认设定，若待识别图像特征较复杂时，需要适当增加训练轮次，提升模型效果，但会导致训练时间变长。

注意：训练轮次并非越大越好，总轮次设置较大时，模型准确率稳定后还会持续训练，会导致训练时间变长，且有过拟合的风险。

学习率 学习率决定神经网络训练过程中每一次迭代优化的步长。建议使用默认设定，图表损失曲线收敛较慢时，可以适当调大 **学习率**；若图表精度波动较大，可以适当调小 **学习率**。

GPU ID 用户训练设备的显卡信息。若用户的设备有多块 GPU，可以在指定 GPU 进行训练。

网络简化 该选项用于简化神经网络结构。默认不勾选。训练数据集相对简单时，勾选该选项可以提高训练和推理速度。

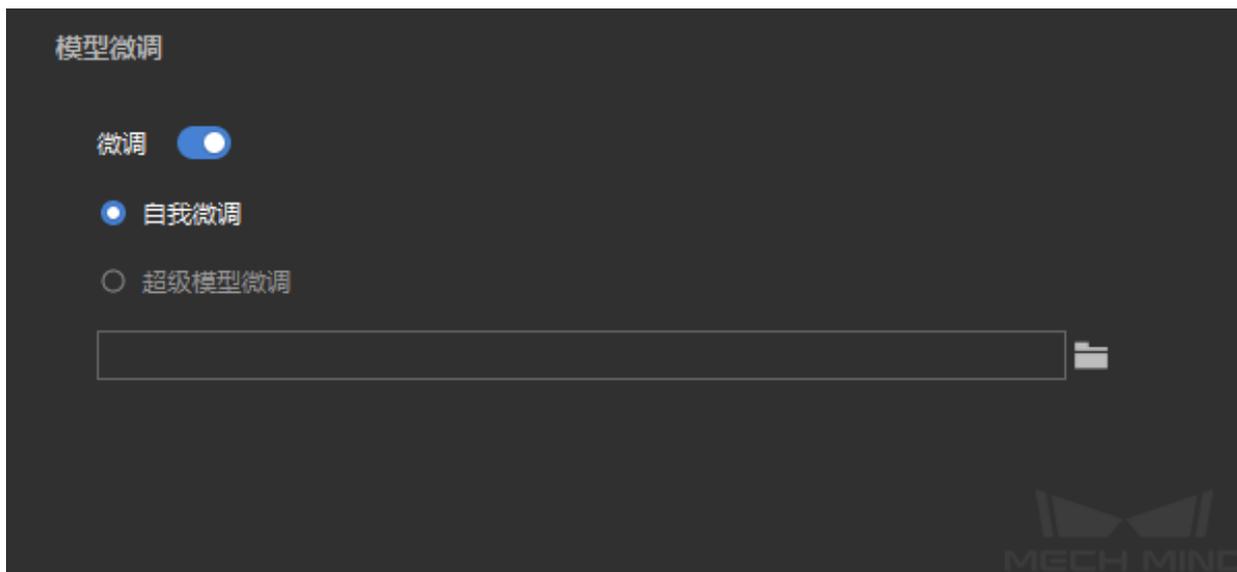
12.3 如何进行模型微调

自我微调 在上一次训练的最优模型的基础上进行训练，可以减少训练时间。

超级模型微调 在现有超级模型的基础上进行微调，需要选择已有的超模，并加入模型预测效果不好的图像进行训练，可以减少训练时间。

超级模型：纸箱超模 [点击此处下载](#)。

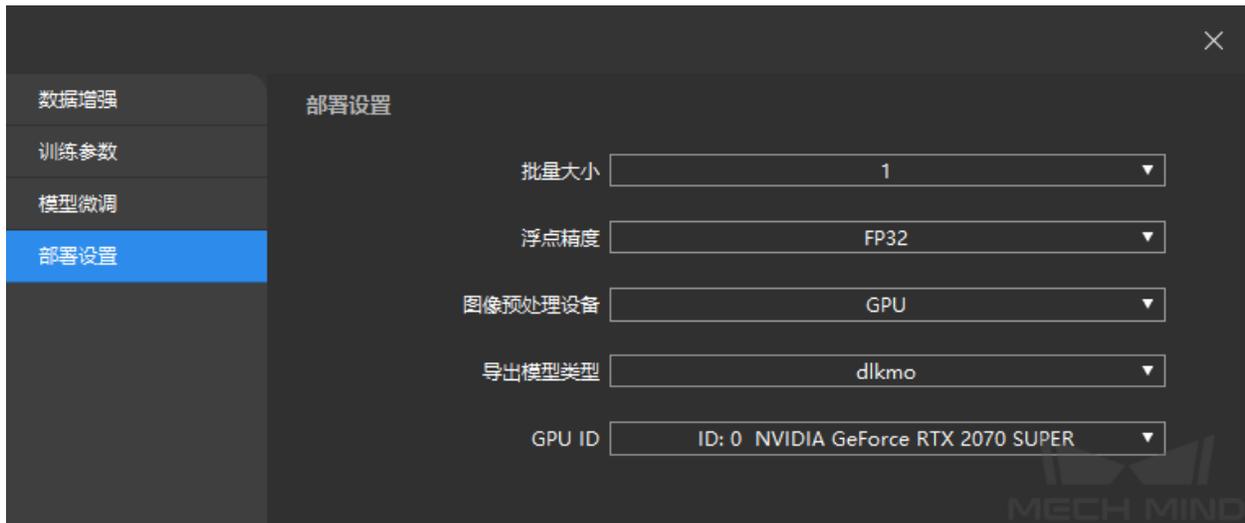
12.3.1 开启模型微调参数



12.4 如何提高模型推理速度和精度

默认参数设定可以满足大部分需求，若需要提高推理速度或精度，可以根据实际情况调整 **部署设置** 相关参数。

12.4.1 调整部署设置参数



批量大小 模型推理一次所选取的样本数量。建议使用默认设定；在“缺陷分割”模块下，若图像尺寸太大缺陷区域较小时，适当调大 **批量大小** 参数可以加快推理速度，但会导致显存占用变高。

浮点精度

- **FP32**: 模型精度高，推理速度较慢。
- **FP16**: 模型精度低，推理速度较快。

图像预处理设备

- **GPU**: 使用模型进行推理前，使用设备显卡处理图像数据。建议使用 **GPU** 参数。
- **CPU**: 若图像较大需要提高图像预处理速度，可以选择 **CPU** 参数。

导出模型类型

- **dlkmo**: 模型需要优化大约 5 分钟，适用于训练模型与部署模型使用的设备 **GPU** 显卡型号不同的情况。
- **dlkmt**: 模型无需优化可以直接导出使用，仅支持训练模型与部署模型使用的设备 **GPU** 显卡型号相同的情况。

GPU ID 用户部署设备的显卡信息。若用户的设备有多块 **GPU**，可以在指定 **GPU** 上部署。

CHAPTER 13

快捷键

编号	功能	快捷键	其他说明
1	新建工程	Ctrl + n/N	
2	保存工程	Ctrl + s/S	
3	打开工程	Ctrl + o/O	
4	撤销 (Undo) 标注	Ctrl + y/Y	
5	恢复 (Redo) 标注	Ctrl + z/Z	
6	复制标注	Ctrl + c/C	
7	粘贴标注	Ctrl + v/V	
8	全选标注	Ctrl + a/A	
9	删除标注	Delete	焦点在标注页内, 选中标注项
10	椭圆工具	l / L	
11	多边形工具	p / P	
12	多边形完成绘制	Enter	
13	矩形工具	r / R	
14	画笔工具	b / B	
15	自动填充套索工具	a / A	
16	标注橡皮工具	e / E	
17	自动标注工具	t / T	
18	掩膜多边形工具	Shift + p/P	
19	掩膜画笔工具	Shift + b/B	
20	掩膜套索工具	Shift + a/A	
21	掩膜橡皮工具	Shift + e/E	
22	网格裁剪工具	u / U	
23	网格选择工具	i / I	
24	特征组标注工具	f / F	
25	ROI 工具	o / O	

下页继续

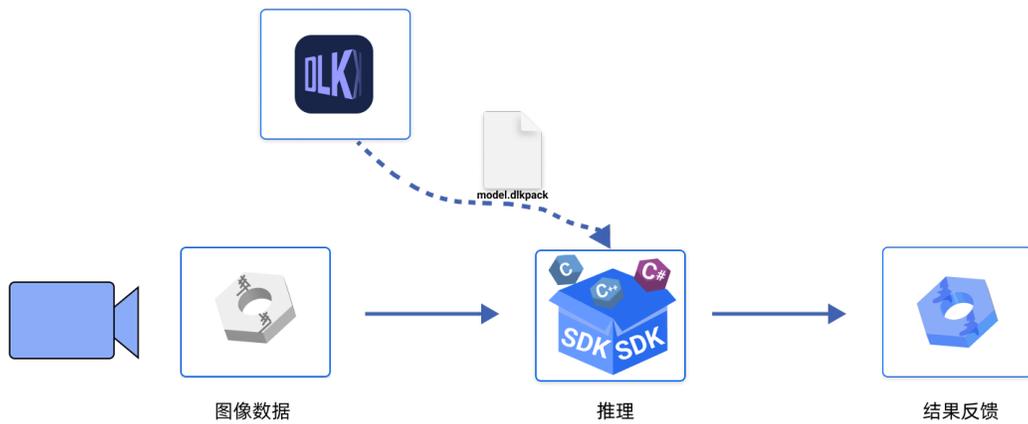
表 1 - 续上页

26	选择工具	s / S	
27	清空当前页面标注	Ctrl + d/D	
28	切换标签显示	Ctrl + l/L	
29	切换预测显示	Ctrl + p/P	
30	删除数据集图片	Delete	焦点在数据集窗口页，选中图像数据
31	印章工具	c / C	
32	印章工具选择模板	Ctrl + 鼠标左键	
33	切换相邻图片	↑、↓	
34	智能标注工具	m / M	
35	智能标注应用	Enter	
36	智能标注取消	q / Q	
37	切换到选择模式	Esc	
38	调整橡皮工具/画笔大小	←→	

小技巧: 将鼠标悬停在界面的工具图标上，可快速查看该工具对应的键盘符号。

14.1 什么是 SDK

SDK 作为 Mech-DLK 软件的开发套件，支持 C、C# 等多语言接口，并提供 DLL 库、样例等满足用户模型部署和二次开发的需求。



14.2 系统要求与授权

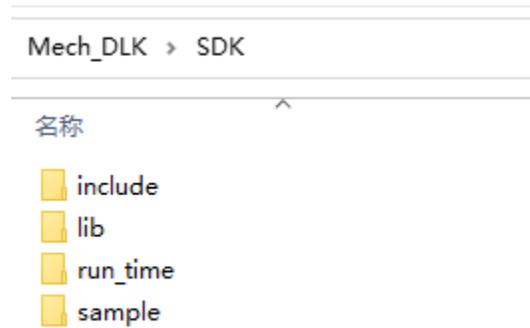
- 确保加密狗已授权；
- 确保计算机系统及软硬件满足 SDK 部署要求：

操作系统	Windows 10（推荐）/ Windows 11
CPU	Intel® Core™ i5 及以上
内存	8GB 及以上
显卡	GeForce GTX 1650 (4GB) 及以上
显卡驱动	驱动版本 471.68 及以上
.NET 环境	.net 版本 >= 4.61
C/C++	编译器支持 C99 及以上

注意：计算机系统和硬件的配置对 SDK 部署后的性能影响较大，因此需确保满足以上要求。

14.3 SDK 位置及结构说明

SDK 位置：



SDK 结构：

include: C API 的头文件，用于 C 工程。

lib: C API 的静态库，用于 C 工程。

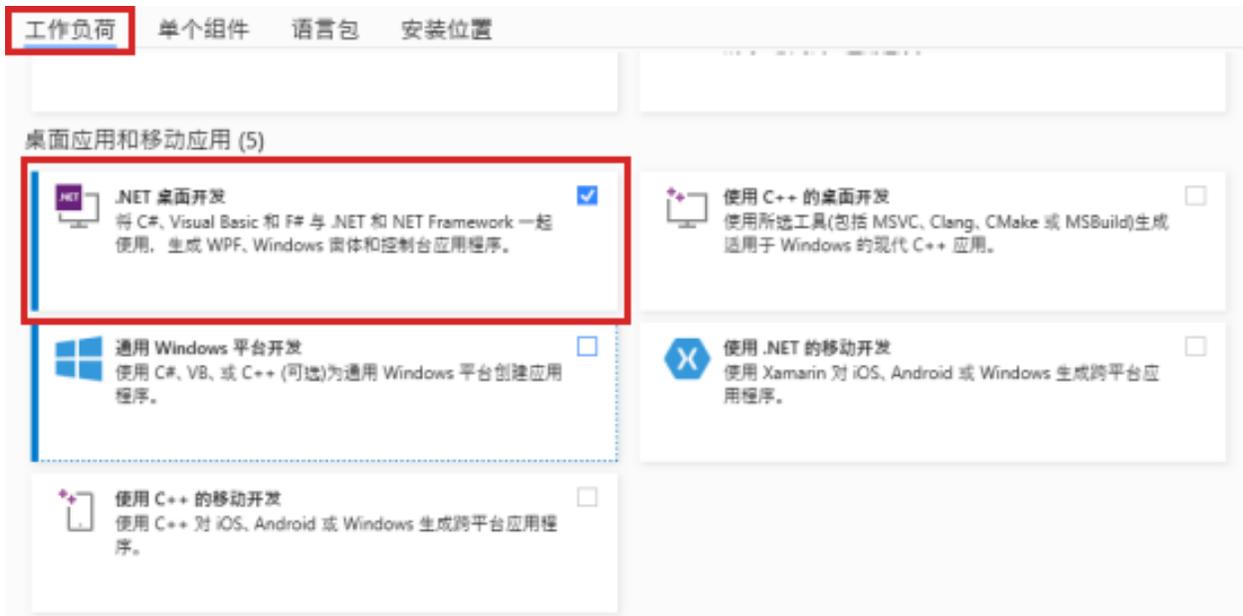
run_time: 动态运行时库，用于 C/C# 工程。

sample: SDK 示例工程，目前有 C# 示例。

本章用于引导初次使用 SDK 的用户熟悉相关操作流程，快速地创建并完成 Mech-DLK SDK 新工程。

15.1 准备工作

1. 下载 Visual Studio 2019
2. 安装 Visual Studio 相关组件：
工作负荷 ▶ .NET 桌面开发。



单个组件 ▶ .NET Framework 4.6.1 目标包。



3. 确保 Mech-DLK 文件夹下有 SDK 文件。

Mech_Mind > Mech_DLK

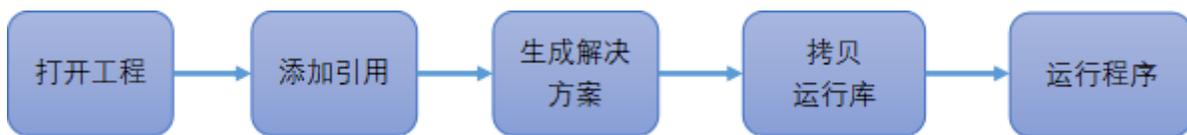
在 Mech_DLK 中搜索

名称	修改日期	类型
installerResources	2022/7/13 11:33	文件夹
mech_tools	2022/7/13 11:34	文件夹
osgPlugins-3.6.4	2022/7/13 11:34	文件夹
platforms	2022/7/13 11:34	文件夹
resource	2022/7/13 11:34	文件夹
SDK	2022/7/15 15:53	文件夹
styles	2022/7/13 11:35	文件夹

15.2 运行 SDK C# 示例工程

SDK_Example 工程提供了示例，展示如何使用 Mech-DLK SDK 来完成应用程序中所需的功能。此处以缺陷分割为例，并为用户提供了所需的模型和图像数据集，带领用户运行 SDK 工程：

- 点击下载图像数据集
- 点击下载模型



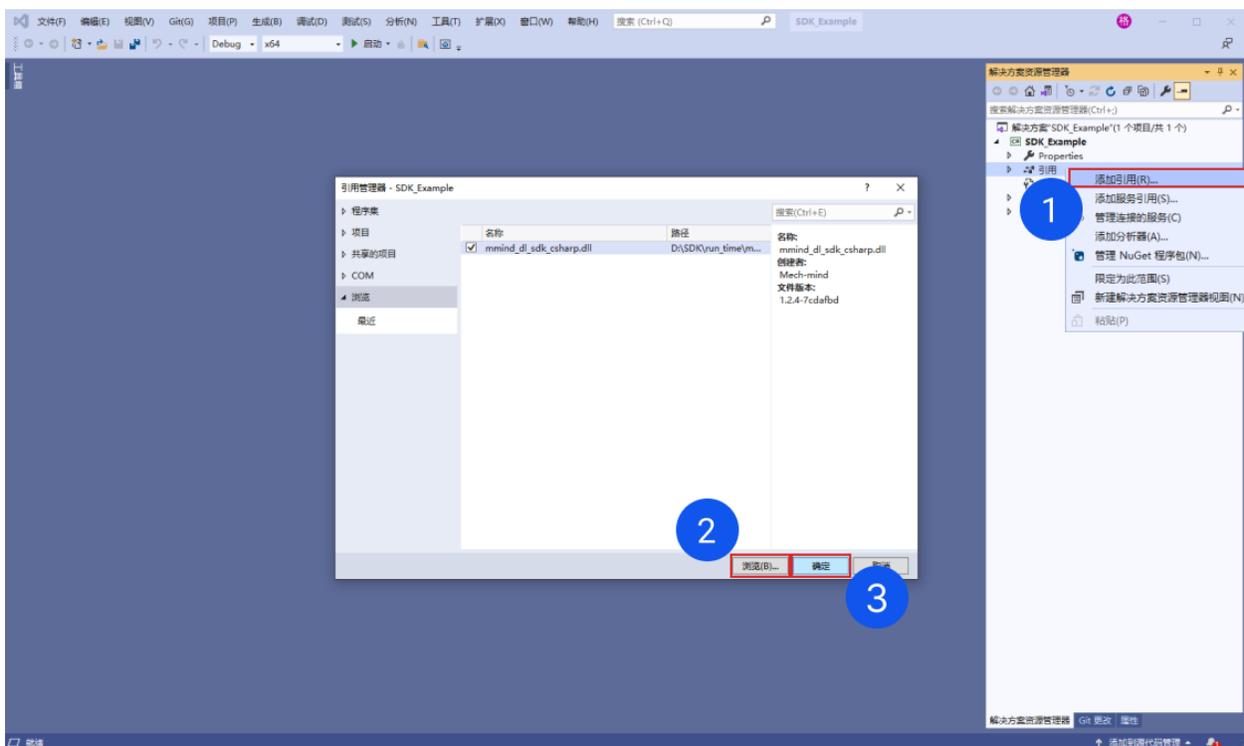
1. 打开示例工程：双击下图路径下的 SDK_Example.sln 文件，使用 Visual Studio 2019 应用打开。

« Mech_DLK » SDK » sample » c# » SDK_Example »

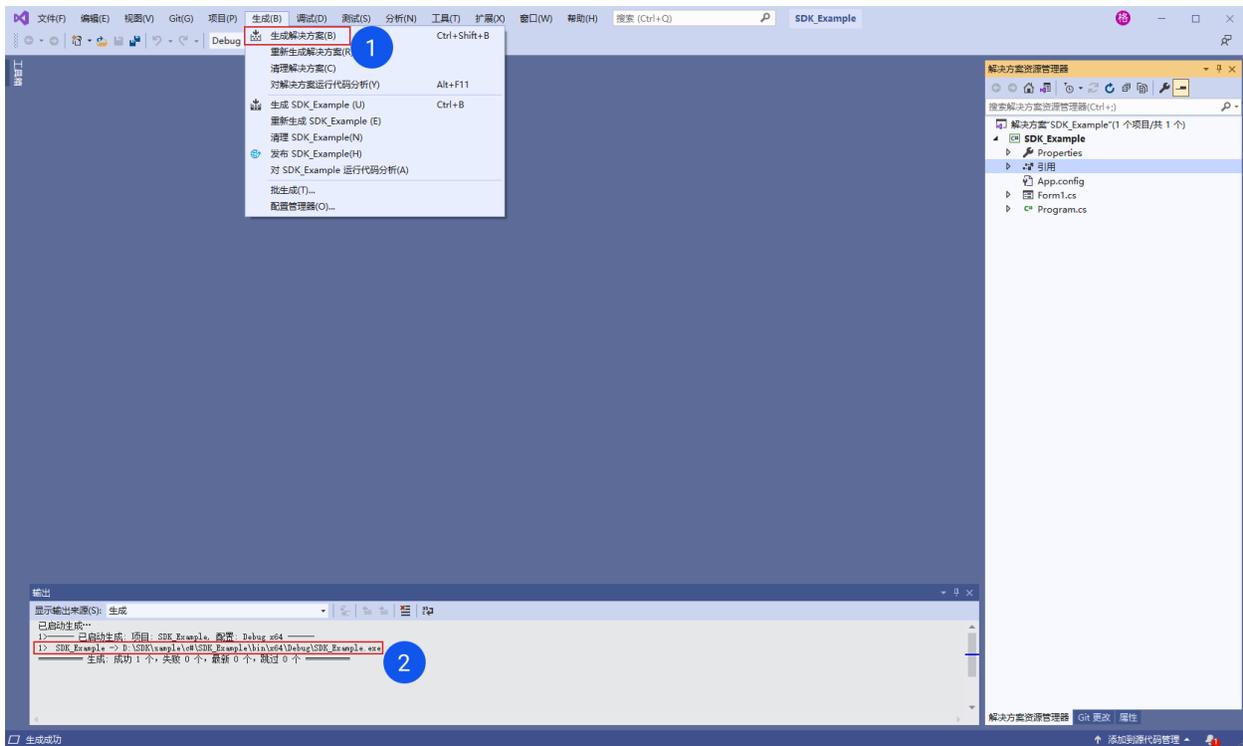
在 SDK_Exar

名称	修改日期	类型	大小
Properties	2022/7/15 15:53	文件夹	
App.config	2022/7/11 14:29	CONFIG 文件	1 KB
Form1.cs	2022/7/11 19:53	CS 文件	3 KB
Form1.Designer.cs	2022/7/11 17:32	CS 文件	8 KB
Form1.resx	2022/7/11 17:32	RESX 文件	6 KB
Program.cs	2022/7/11 17:04	CS 文件	1 KB
SDK_Example.csproj	2022/7/11 19:40	CSPROJ 文件	5 KB
SDK_Example.sln	2022/7/11 16:53	SLN 文件	2 KB

2. 引用 SDK: 点击右键工程栏中的 引用选择添加引用, 会弹出引用管理器窗口, 点击 浏览勾选 run time 文件夹下的 mmind_dl_sdk_csharp.dll 文件后点击 确定。



3. 生成解决方案：点击上方导航栏的生成选择生成解决方案 (B) 生成可执行文件。在界面下方的输出栏中查看可执行文件生成的路径。



4. 添加运行时库：将下图文件夹内所有的 dll 格式的文件拷贝到上一步生成的可执行文件 Release 目录下，若有重复文件，直接覆盖即可。

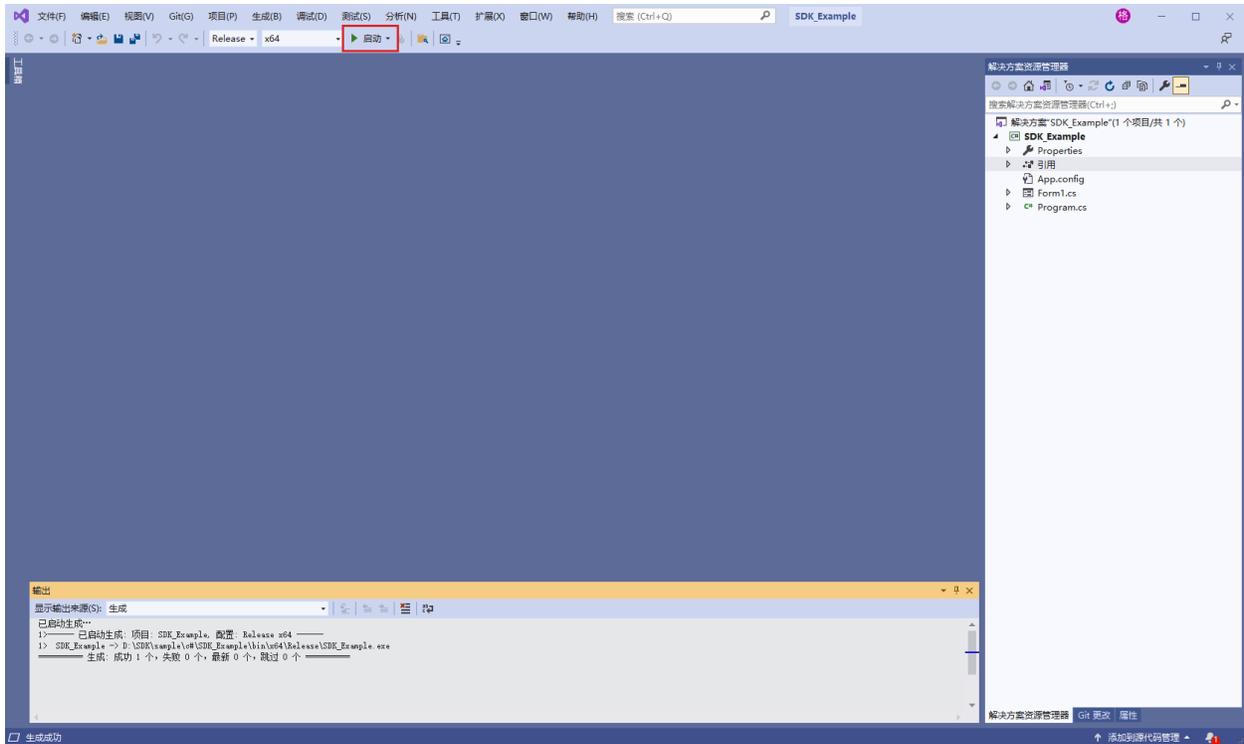
原文件位置：

« Mech_DLK » SDK » run_time

目标路径：

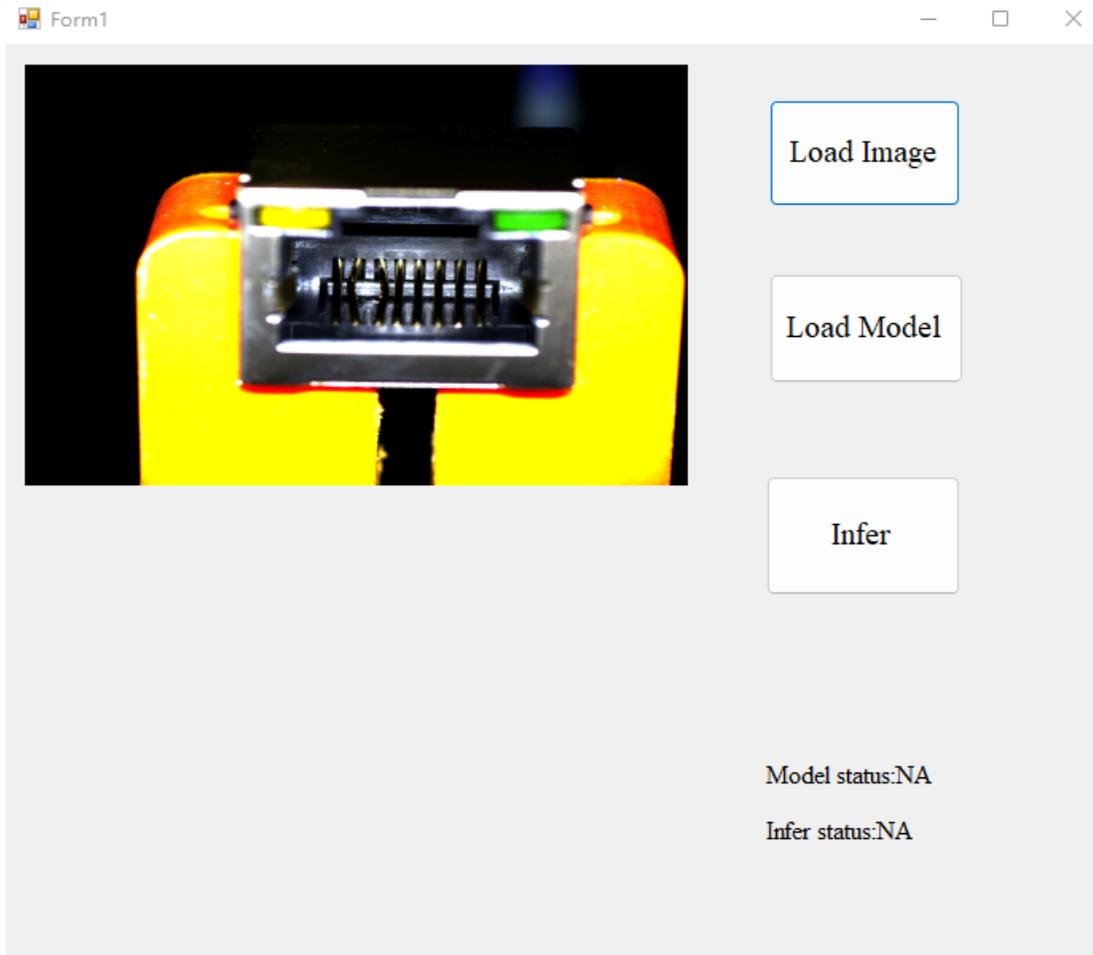
« Mech_DLK » SDK » sample » C# » SDK_Example » bin » x64 » Debug

5. 启动项目：点击启动运行程序。

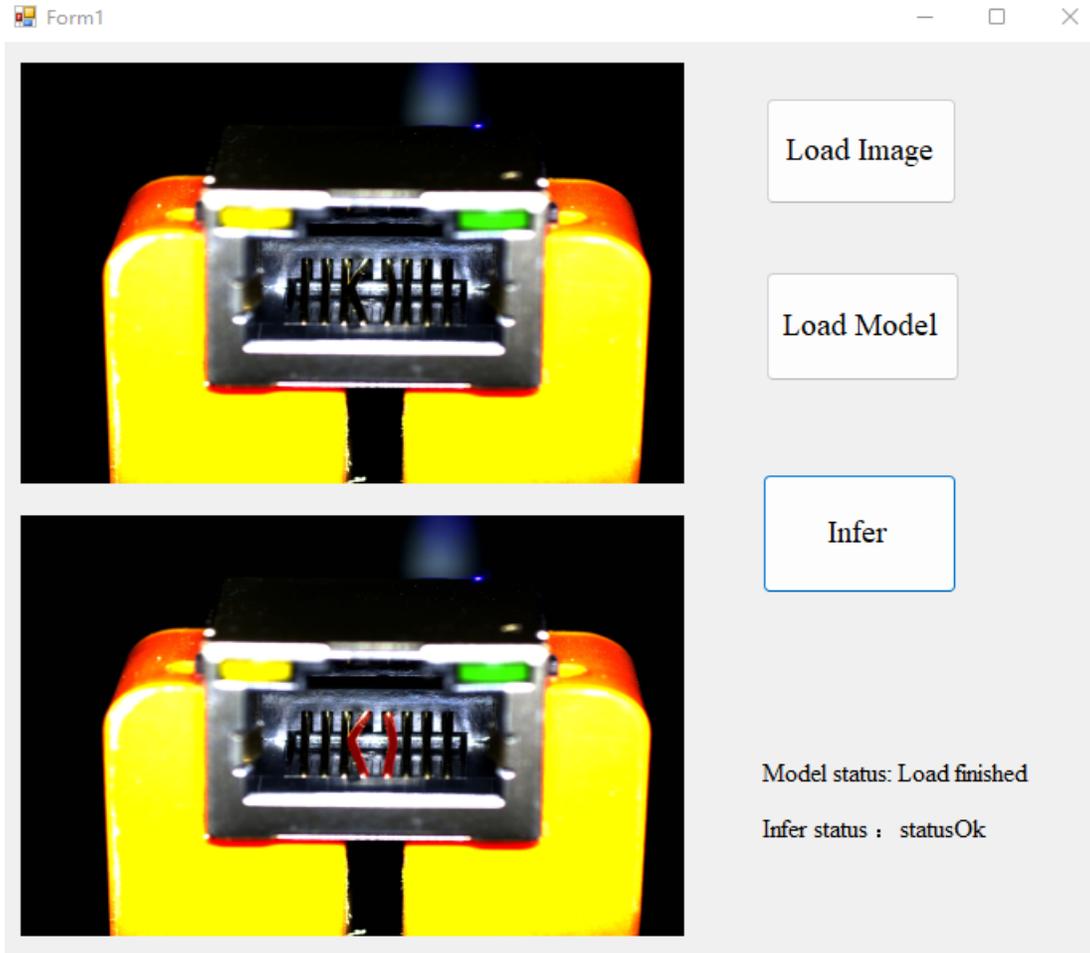


启动后程序界面:

- *Load Image* : 加载待推理图像。
- *Load Model* : 加载 Mech-DLK 导出的 dlkpack 格式的模型。
- *Infer* : 进行推理。



运行结果界面：

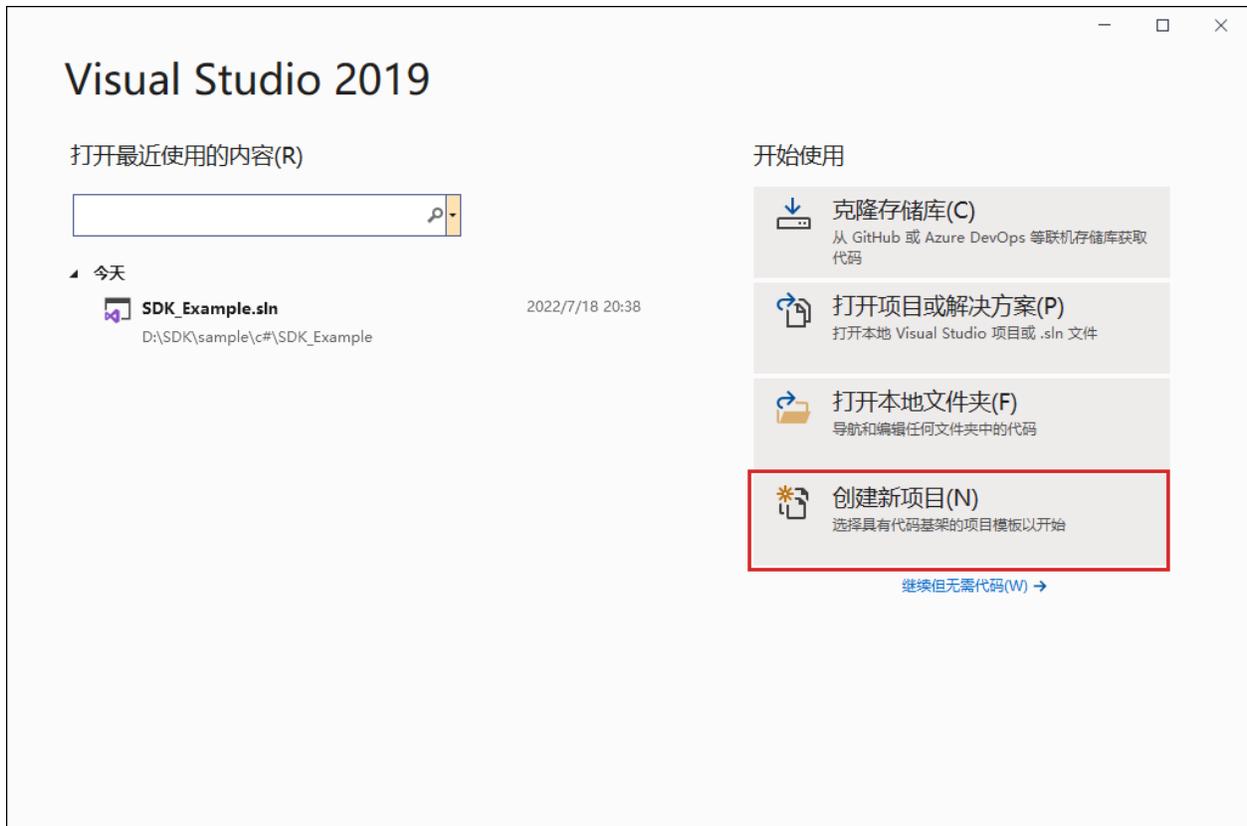


注意：首次加载模型时间略长，需要等待 5 分钟左右，建议用户使用多线程来调用相应接口。

15.3 创建一个新工程

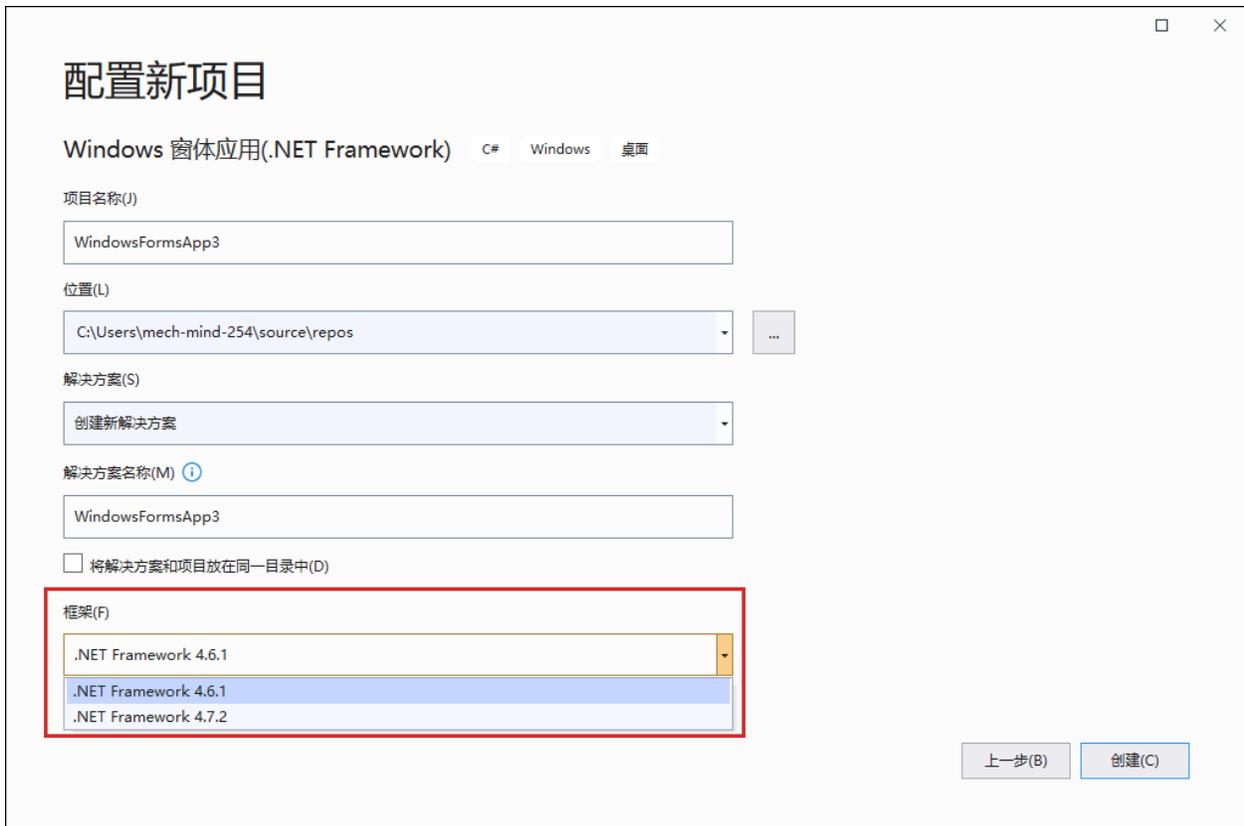
确保系统环境满足二次开发要求后，即可参考以下步骤使用 SDK 创建新工程，此处以 .NET (Visual Studio 2019) 为例：

1. **创建新项目：**打开 Visual Studio 2019，初次新建工程的用户点击从左上角 文件 ▶ 新建项目，在创建新项目页面点击 Windows 桌面向导创建一个 Windows 应用。

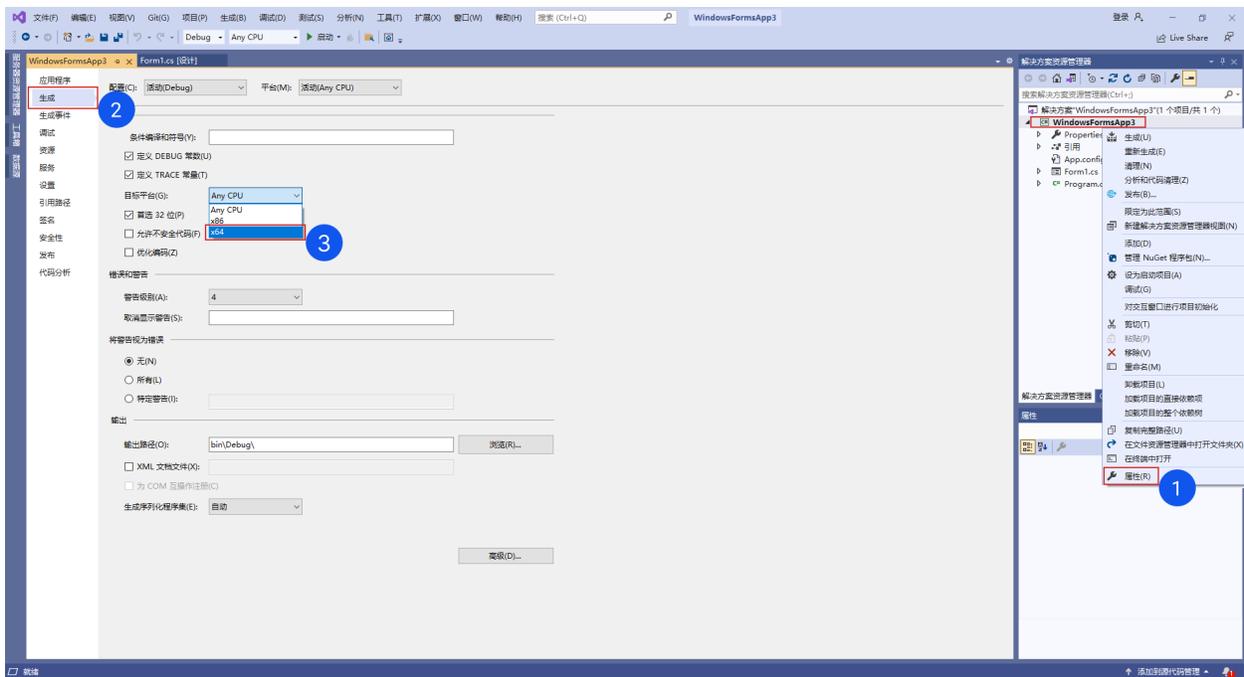




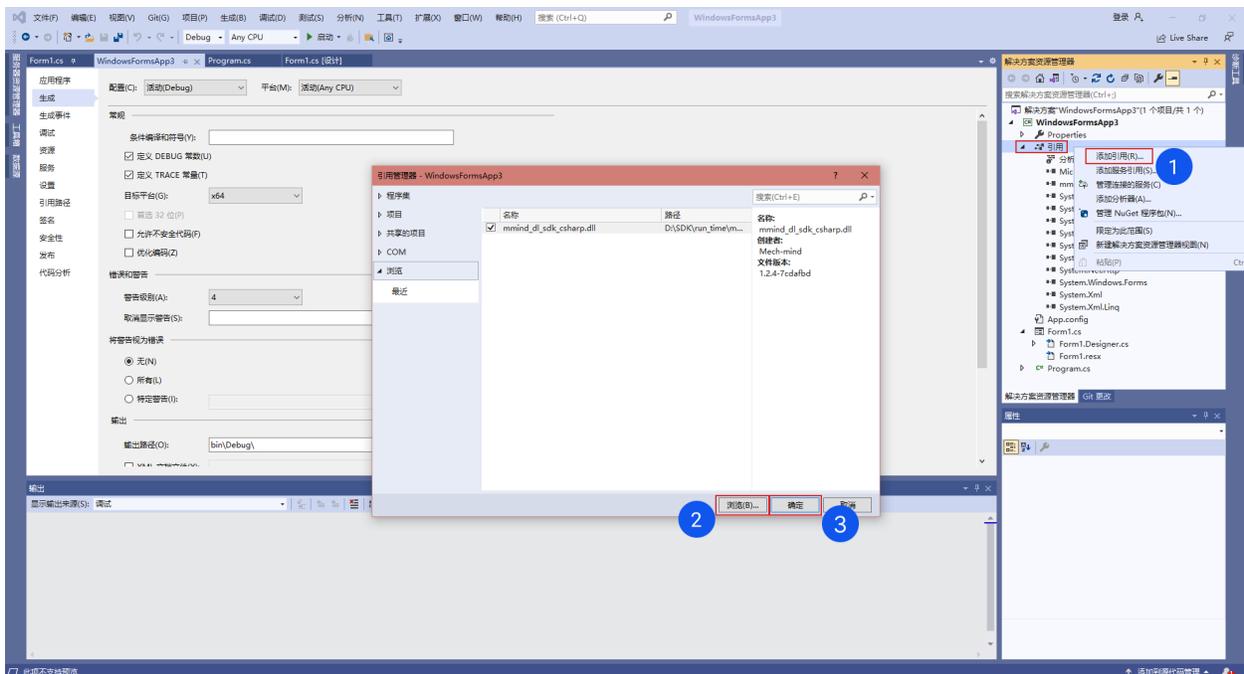
2. 配置新项目：选择.NET 框架.NET Framework 4.6.1(支持 >= 4.6.1)



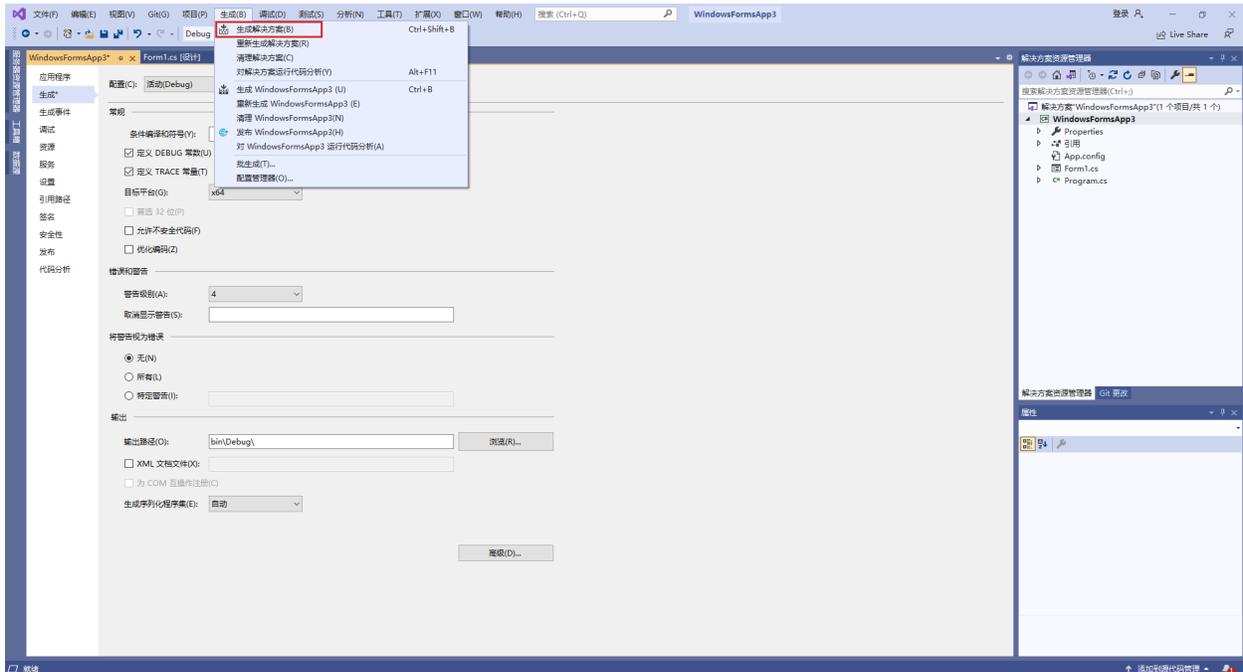
3. 配置工程属性：在右侧工程栏中右击新工程名称依次选择 属性 ▶ 生成 ▶ 目标平台选择 x64 模式，也可以通过配置管理器来新增 x64 活动解决方案平台。



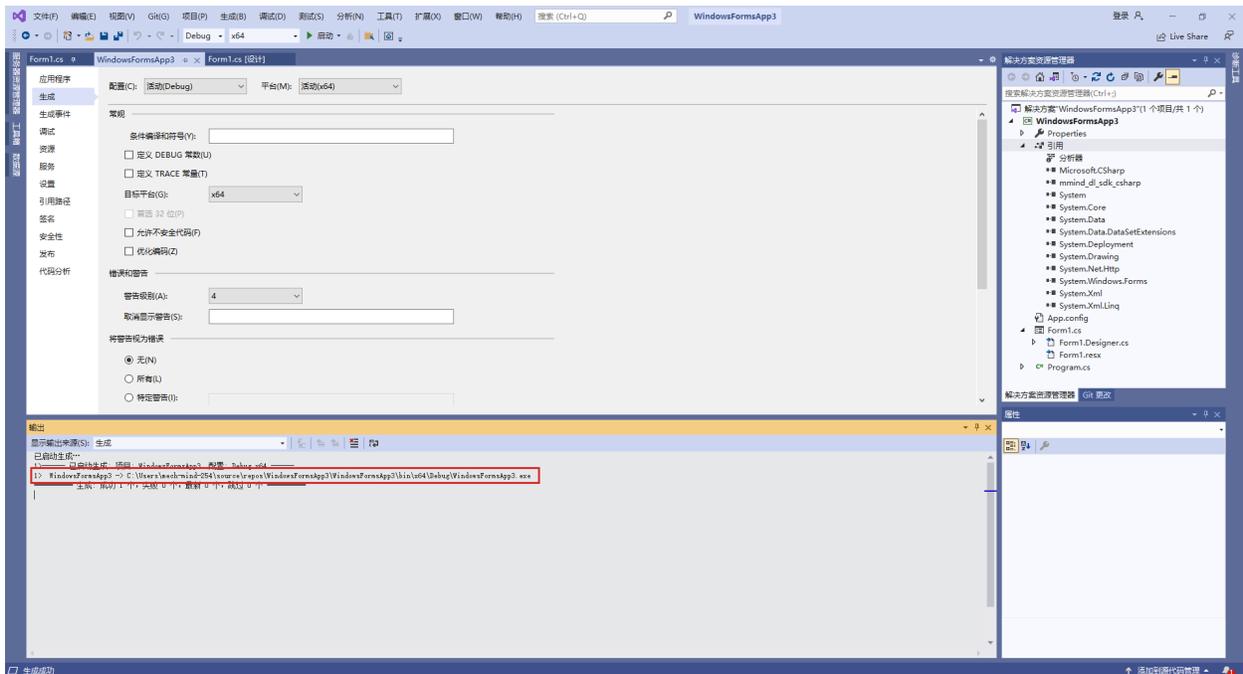
4. 添加引用：点击右键工程栏中的引用项选择添加引用，在弹窗中点击浏览勾选 run time 文件夹下的 mmind_dl_sdk_csharp.dll 文件后点击确定。如果窗口中已经有该文件，则直接勾选。



5. 生成解决方案：点击上方导航栏 生成 ▶ 生成解决方案 (B)，生成可执行文件。



将 SDK 文件夹下的 run_time 中所有 DLL 文件拷贝到 exe 目录当中，可在下图中红框位置找到 exe 目录路径。



6. 开始定制化开发

用户可参考示例工程的代码或者[API 参考手册](#) 完成后续的定制化开发。

16.1 C# API

16.1.1 接口函数

1. setVisible

```
void setVisible(Boolean flag);
```

设置结果可视化。若配置为 `true`，SDK 会自动将预测的结果与输入数据进行融合，结果在接口函数推理返回结果中，若性能要求较高请勿勾选。

输入参数： `flag`：可视化布尔量，默认为 `false`。

返回值： 空

2. setConvertInseg

```
void setConvertInseg(Boolean Convertflag);
```

设置对实例分割结果进行处理。若配置为 `true`，实例分割类型的推理结果将会有内外接矩形信息，结果在接口函数推理返回结果中，若性能要求较高请勿勾选。

输入参数： `Convertflag`：实例分割掩膜绘制内外接矩形布尔值，默认为 `false`。

返回值： 空

3. LoadModel

```
SDKStatus LoadModel(string packPath);
```

加载 Mech-DLK 导出的模型，该接口为同步接口，会阻塞，建议使用线程调用。

输入参数： packPath: dlkpack 文件路径。

返回值： SDKStatus: 1000 为正常，其他值为异常，详情见接口函数调用返回值。

4. Predict

```
InferPackResult Predict(Bitmap img);
```

执行推理。

输入参数： img: bitmap 类型的图像。

返回值： InferPackResult: 推理结果。详情见 接口函数推理返回结果。

16.1.2 接口函数调用返回值

返回状态码	枚举值	说明
STATUS_OK	1000	接口调用未出现异常
FILE_NOT_FOUND	1001	文件未找到
VALUE_OUTOF_LEFTRANGE	1002	参数值超出左限位。如参数值范围为 [0.0, 1.0], 当参数值小于 0.0 时产生
VALUE_OUTOF_RIGHTRANGE	1003	参数值超出右限位。如参数值范围为 [0.0, 1.0], 当参数值大于 1.0 时产生
MODELTYPE_ERROR	1004	模型类型异常
MODEL_REGISTEXCEEDLIMIT_ERROR	1005	注册模型的个数超过限制错误
MODEL_CREATE_ERROR	1006	模型创建时产生异常
MODEL_DESTROY_ERROR	1007	模型注销时产生异常
MODEL_LOAD_ERROR	1008	模型加载时产生异常
MODEL_INIT_ERROR	1009	模型初始化时产生异常
MODEL_INFER_ERROR	1010	模型推理时产生异常
GET_RESULT_ERROR	1011	获取模型结果时产生异常
IMGPATH_ERROR	1012	错误的图像路径
IMGNULL_ERROR	1013	图像为空
IMGCHANNEL_ERROR	1014	图像的通道数错误
IMGDEPTH_ERROR	1015	图像的位深度错误
IMGSIZE_ERROR	1016	图像的尺寸错误
INCONSISTENT_IMAGEDEPTH_ERROR	1017	两个图像的位深度不一致
INCONSISTENT_IMAGESIZE_ERROR	1018	两个图像的尺寸不一致
INCONSISTENT_IMAGETYPE_ERROR	1019	两个图像的类型不一致
IMGROI_ERROR	1020	错误的图像 ROI 参数
IMGROI_NULL_ERROR	1021	图像的 ROI 参数为空
IMG_CONVERTTYPE_ERROR	1022	图像色彩空间变换的转换参数错误
CONFIG_PARAM_ERROR	1023	配置参数错误
UNKNOWN_ERROR	1024	未知错误

16.1.3 接口函数推理返回结果

返回状态码	枚举值	说明
status	SDKStatus	1000 为正常，其他值为异常，详情见接口函数调用返回值
type	Type	模型类型，0 为未知模型，1 为缺陷分割，2 为实例分割，3 为目标检测，4 为图像分类
imageShow	Bitmap	可视化图像
masks	Bitmap[]	掩膜结果
labels	int[]	推理的标签结果
bboxes	float[]	包围框坐标序列，4 个一组（左上 X、左上 Y、右下 X、右下 Y）
external-Rect	float[]	实例分割最小外接矩形坐标序列，4 个一组（左上 X、左上 Y、右下 X、右下 Y）
internal-Rect	float[]	实例分割最大内接矩形坐标序列，4 个一组（左上 X、左上 Y、右下 X、右下 Y）
confidence	float[]	置信度结果

16.2 C API

16.2.1 接口函数

1. mmind_packInfer_create

```
int mmind_packInfer_create(Engine* engine, const char* packPath);
```

解析 dlkpack 文件并创建对应的推理引擎。

输入参数： engine：推理引擎信息（指针）。

packPath：dlkpack 文件路径。

返回值： int：1000 为正常，其他值为异常，详情见接口函数调用返回值。

2. mmind_packInfer_destroy

```
int mmind_packInfer_destroy(Engine engine);
```

注销推理引擎。

输入参数： engine：推理引擎信息。

返回值： int：1000 为正常，其他值为异常，详情见接口函数调用返回值。

3. mmind_packInfer_infer

```
int mmind_packInfer_infer(Engine engine, Image* img, Image* segMask, ↵
↵ Image* inSegMask, int& maskNumber, int* h, int* w, int& labelNumber, ↵
↵ int* labels, int& bboxNumber, float* bboxes, int& confidenceNumber, ↵
↵ float* confidences);
```

推理引擎执行推理。

输入参数： engine: 推理引擎信息。

img: 输入图像。

segMask: 缺陷分割的掩膜结果。

insegMask: 实例分割的掩膜结果。

maskNumber: 实例分割掩膜数量。

h: 实例分割的掩膜图像的高。

w: 实例分割的掩膜图像的宽。

labelNumber: 标签数量。

labels: 推理的标签结果。

bboxNumber: 包围框的数量。

bboxes: 包围框坐标序列，4 个一组（左上 X、左上 Y、右下 X、右下 Y）。

confidenceNumber: 置信度结果的数量。

confidences: 置信度结果。

返回值： int: 1000 为正常，其他值为异常，详情见[接口函数调用返回值](#)。

16.2.2 接口函数调用返回值

返回状态码	枚举值	说明
STATUS_OK	1000	接口调用未出现异常
FILE_NOT_FOUND	1001	文件未找到
VALUE_OUTOF_LEFTRANGE	1002	参数值超出左限位。如参数值范围为 [0.0, 1.0], 当参数值小于 0.0 时产生
VALUE_OUTOF_RIGHTRANGE	1003	参数值超出右限位。如参数值范围为 [0.0, 1.0], 当参数值大于 1.0 时产生
MODELTYPE_ERROR	1004	模型类型异常
MODEL_REGISTEXCEEDLIMIT_ERRORS	1005	注册模型的个数超过限制错误
MODEL_CREATE_ERROR	1006	模型创建时产生异常
MODEL_DESTROY_ERROR	1007	模型注销时产生异常
MODEL_LOAD_ERROR	1008	模型加载时产生异常
MODEL_INIT_ERROR	1009	模型初始化时产生异常
MODEL_INFER_ERROR	1010	模型推理时产生异常
GET_RESULT_ERROR	1011	获取模型结果时产生异常
IMGPATH_ERROR	1012	错误的图像路径
IMGNULL_ERROR	1013	图像为空
IMGCHANNEL_ERROR	1014	图像的通道数错误
IMGDEPTH_ERROR	1015	图像的位深度错误
IMGSIZE_ERROR	1016	图像的尺寸错误
INCONSIS-TENT_IMAGEDEPTH_ERROR	1017	两个图像的位深度不一致
INCONSIS-TENT_IMAGESIZE_ERROR	1018	两个图像的尺寸不一致
INCONSIS-TENT_IMAGETYPE_ERROR	1019	两个图像的类型不一致
IMGROI_ERROR	1020	错误的图像 ROI 参数
IMGROI_NULL_ERROR	1021	图像的 ROI 参数为空
IMG_CONVERTTYPE_ERROR	1022	图像色彩空间变换的转换参数错误
CONFIG_PARAM_ERROR	1023	配置参数错误
INVALID_ENGINE_ID	2000	推理引擎对象异常
UNKNOWN_ERROR	9000	未知错误

梅卡曼德视觉系列软件是梅卡曼德机器人自主研发的 Mech-DLK、Mech-Vision、Mech-Viz 和 Mech-Center 软件等的统称（以下简称“软件”）。

为了确保可以正常使用软件，需要按照下列内容进行操作：

- 检查接口和驱动程序
- 设置软件许可
- 关闭 *Windows Defender* 防火墙
- 阻止 *Windows* 更新

17.1 检查接口和驱动程序

用户需要在 Windows 操作系统的 **控制面板** 中检查以下内容：

1. **网络和 Internet**：确保用于连接其他设备的网络接口工作正常。
2. **设备管理器**：点击 **网络适配器** 和 **显示适配器**，确保安装了网络接口和 GPU 所需的驱动程序。

17.2 设置软件许可

梅卡曼德使用 Wibu-Systems 的 CodeMeter 作为软件的许可系统。从 1.6.0 版本开始，CodeMeter 安装程序包含在软件的安装包中。

1. 将“加密狗”插到工控机上。
2. 运行软件安装程序安装 CodeMeter。
3. 确认 CodeMeter 正在运行，即 CodeMeter 的图标  显示在系统托盘中。

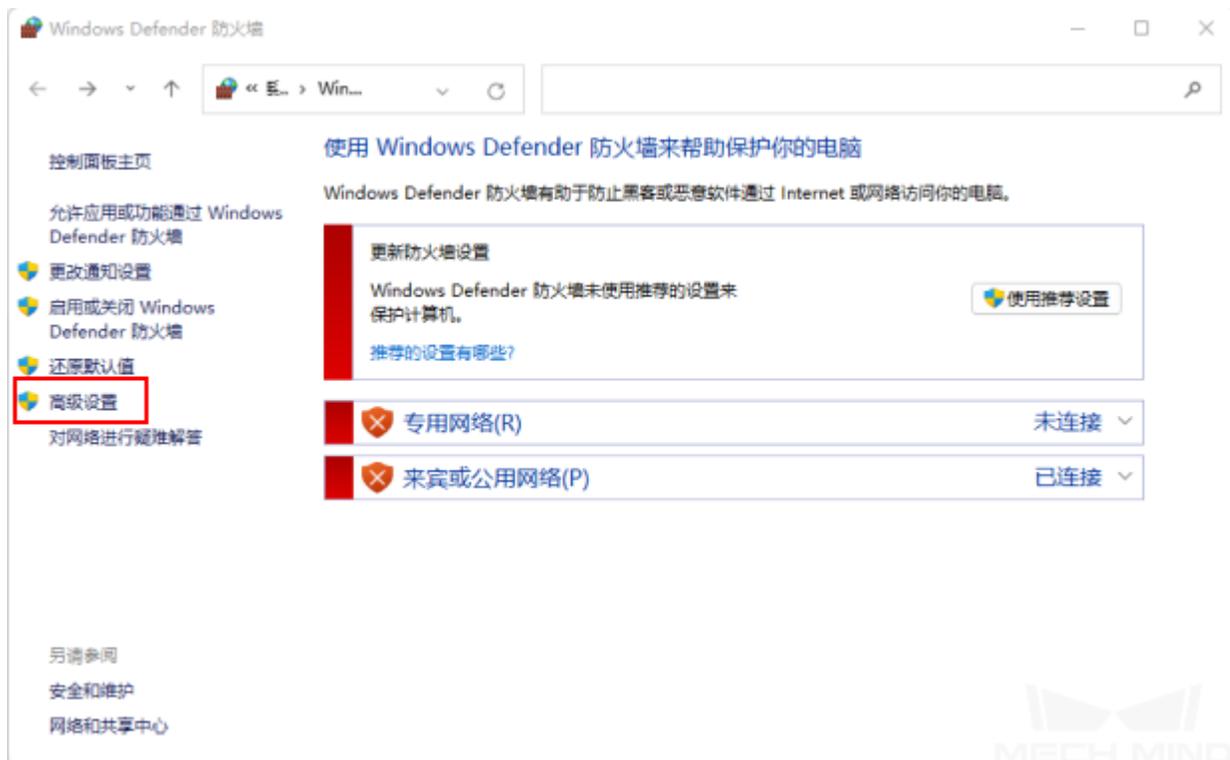
注解:

- 若用户需要获取试用许可（不需要加密狗），请参见获取试用许可。
- 若用户需要更新已有许可，请参见更新软件许可。

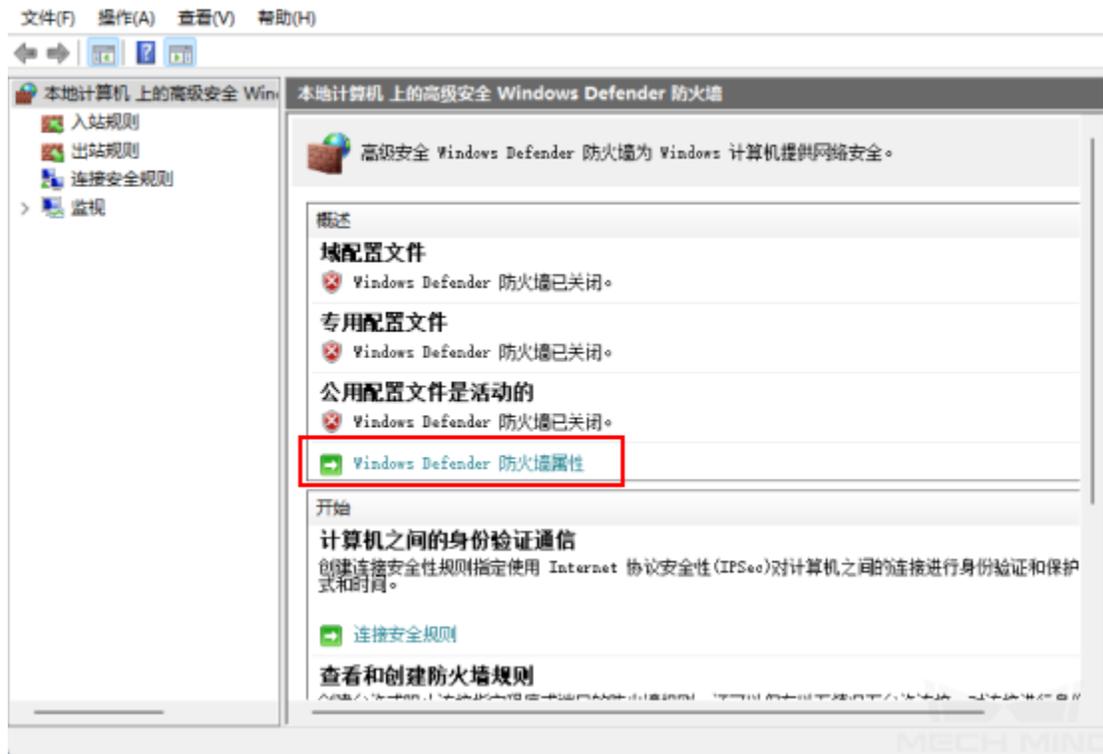
17.3 关闭 Windows Defender 防火墙

Windows Defender 防火墙可能会影响软件与外部设备通过网络接口进行通信，用户需要关闭 Windows Defender 防火墙对这些网络接口的防护，具体操作如下：

1. 打开控制面板中点击 系统和安全 ▶ Windows Defender 防火墙 ▶ 高级设置。



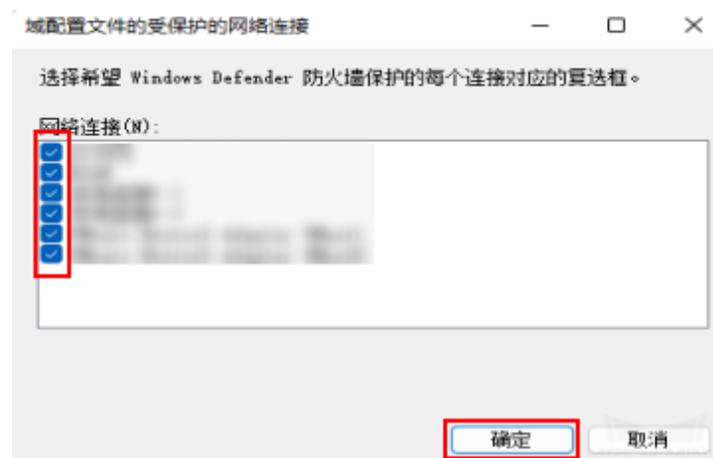
2. 点击弹窗中的 **Windows Defender** 防火墙属性。



3. 选中 **域配置文件** 选项卡，点击 **受保护的网络连接** 右侧的 **自定义**。



- 取消勾选弹出的窗口中所有与软件通信的设备的网络接口，点击 确定。



- 专用文件配置和 公用文件配置选项卡中重复步骤 3 与步骤 4。



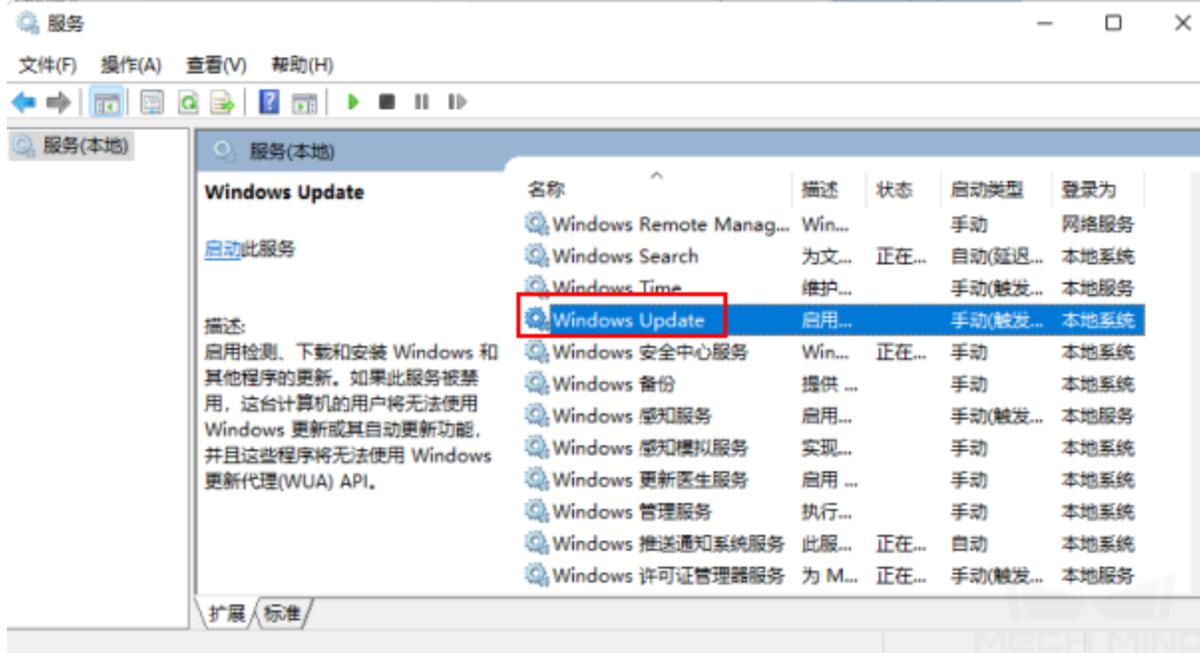
17.4 阻止 Windows 更新

Windows 系统更新时会强制工控机关机或重启，可能会影响现场正常生产，因此建议设置为阻止 Windows 更新来避免意外停机。

注解： 如果选择保持启用 Windows 更新，注意采取措施来确保工控机不在工作时间内关机或重启，例如：设置 Windows 更新的活跃时间。

17.4.1 禁用 Windows 更新

1. 点击任务栏的放大镜，搜索 **服务**。
2. 点击 **服务** 双击弹窗中的 **Windows Update** 并打开。



3. 选择 停止后点击确认。



4. 启动类型的下拉列表框中选择 禁用，并点击 应用。



5. 点击 恢复选项卡，将所有失败时的反应选择为 无操作，点击 确定。

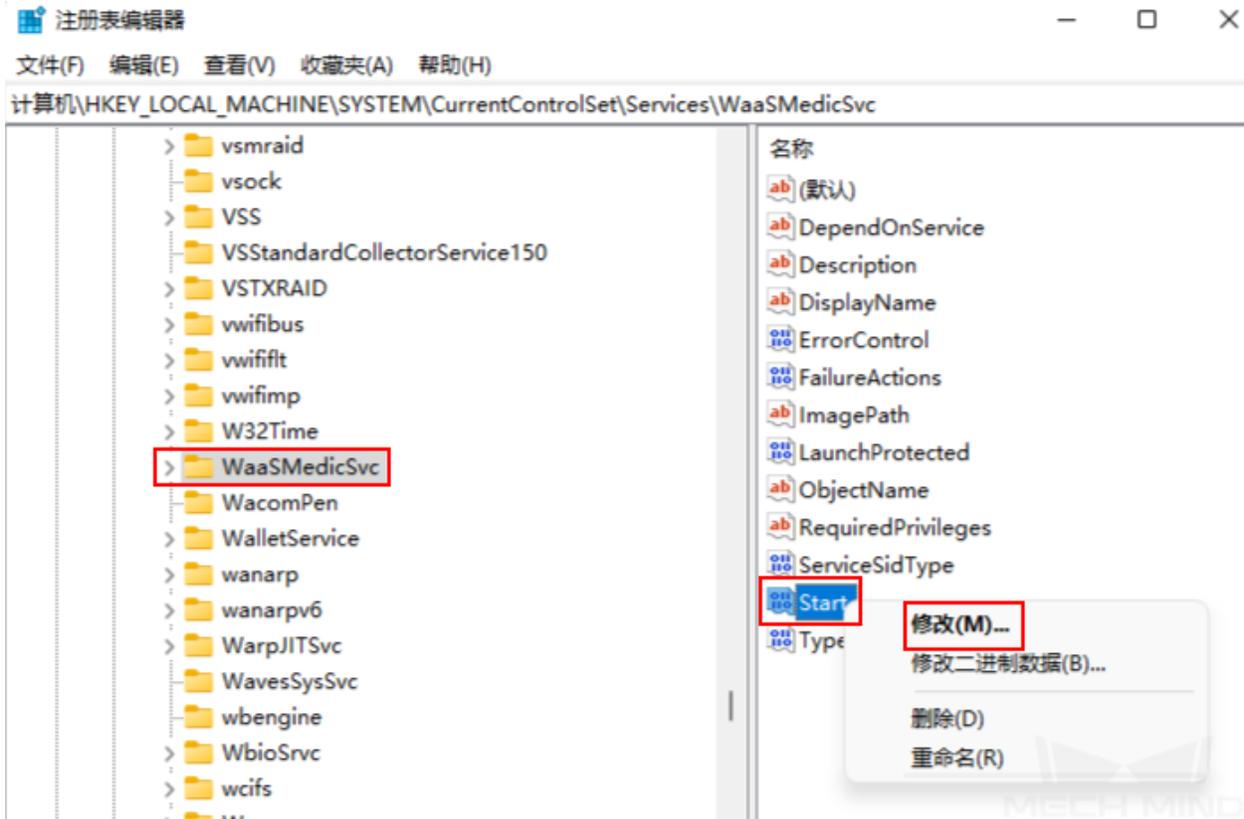


17.4.2 禁用 Windows 更新医生服务

打开 Windows 更新医生服务会使计算机持续接收更新。即使用户已经关闭 Windows 更新，Windows 更新医生服务依旧会重新启用 Windows 更新。用户只有关闭 Windows 更新医生服务才能确保彻底禁用 Windows 更新。

Windows 更新医生服务无法通过简单的按钮点击来关闭，需要用户在注册表编辑器中完成以下操作。

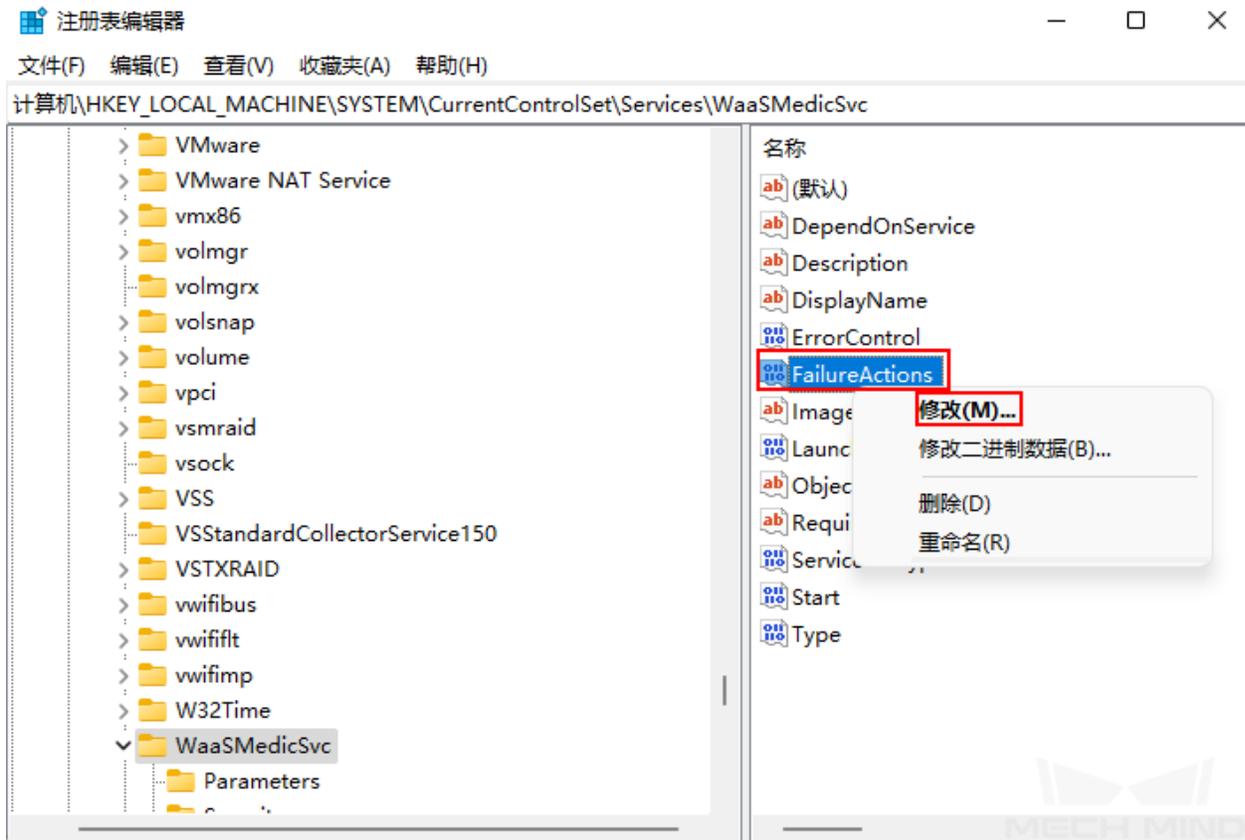
1. 点击任务栏的放大镜，搜索并点击 **注册表编辑器**。
2. 左侧面板导航到 **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WaaSMedicSvc** 路径下，点击右侧面板中的 **Start**，选择 **修改**。



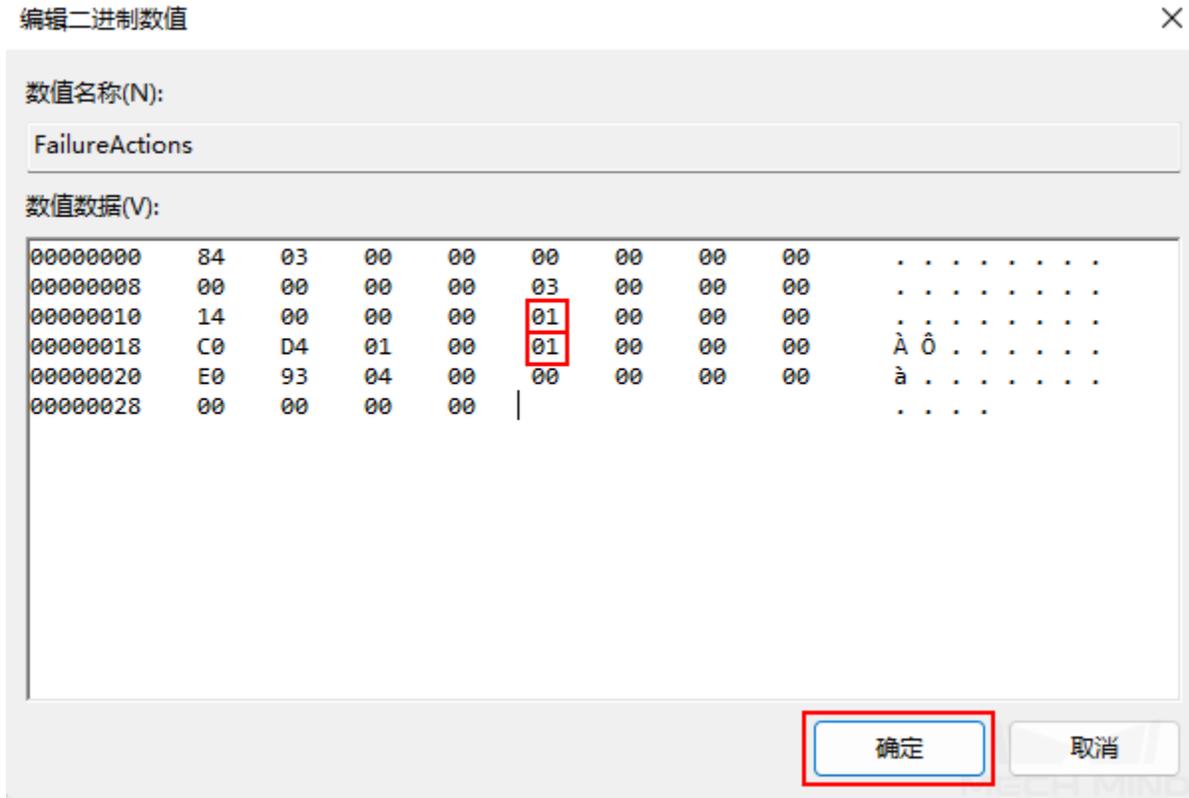
3. 将弹窗中的 数值数据修改为 4 后，点击 确定。



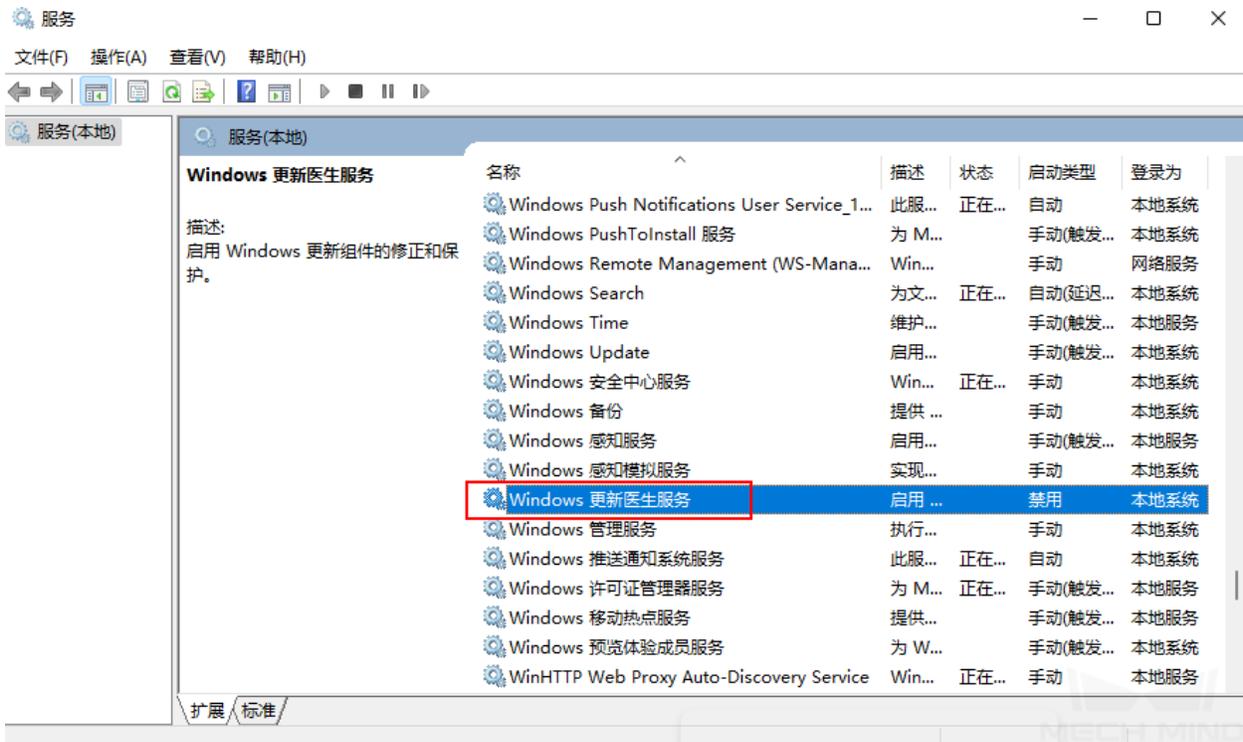
4. 右键点击 **FailureActions**，在弹出的快捷菜单中点击 **修改**。



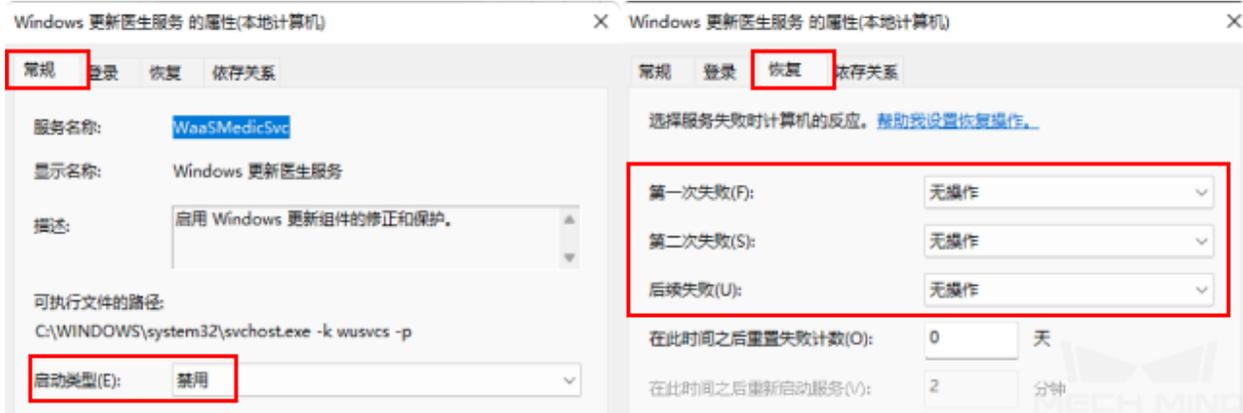
5. 将弹窗中框选的值修改为 **01** 后，点击 确定。



6. 返回 服务找到 Windows 更新医生服务双击并打开。



7. 确认修改成功：常规选项卡中的启动类型为禁用，恢复选项卡中所有失败时的反应为无操作。



本章内容介绍了如何获取及更新 Mech-DLK 软件许可。

18.1 获取试用许可

梅卡曼德使用 Wibu-Systems 的 CodeMeter 作为软件的许可系统，不需要“加密狗”即可获得试用许可。

18.1.1 获取票单

联系梅卡曼德销售获取 **票单**（票单是由数字、字母、连字符组成的 25 位字符串），详细信息将通过电子邮件发送给用户。

18.1.2 安装 CodeMeter

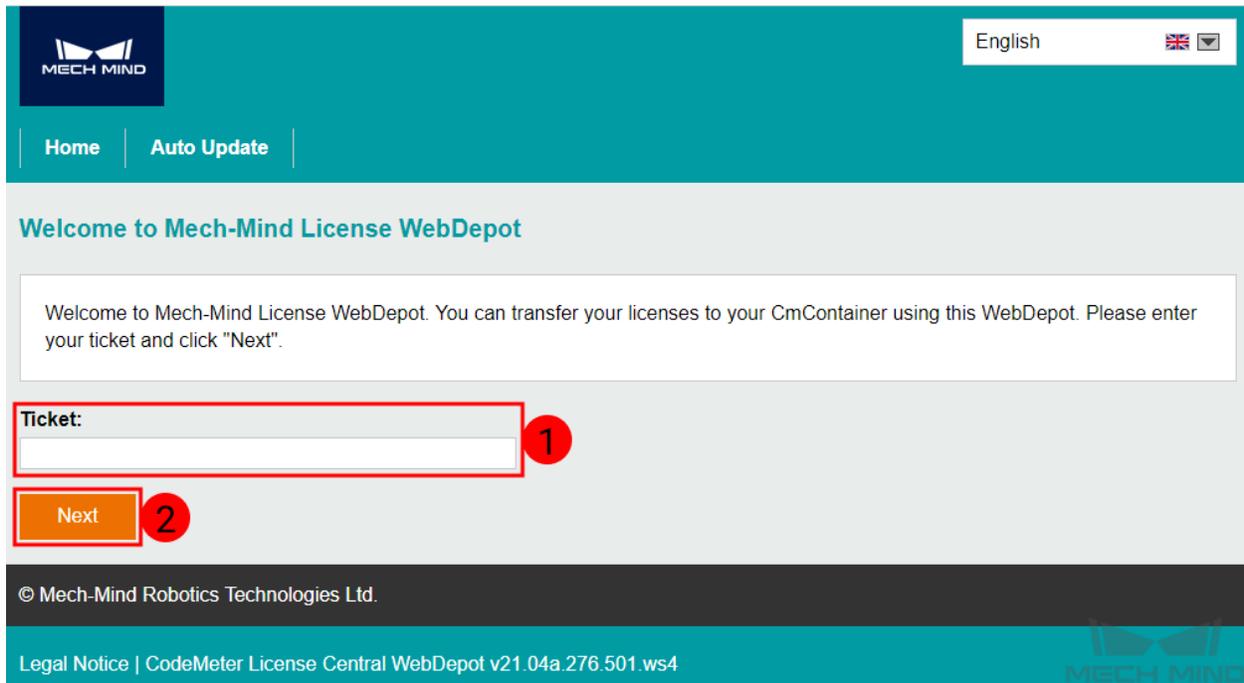
CodeMeter 安装程序包含在软件安装包里，运行软件安装包即可安装 CodeMeter。

18.1.3 激活许可

在 Mech-Mind License WebDepot 中进行激活操作。

激活流程如下：

1. 复制票单后将其粘贴到 **Ticket** 输入框中，然后单击 *Next*。



English  

[Home](#) | [Auto Update](#)

Welcome to Mech-Mind License WebDepot

Welcome to Mech-Mind License WebDepot. You can transfer your licenses to your CmContainer using this WebDepot. Please enter your ticket and click "Next".

Ticket:

[Next](#)

© Mech-Mind Robotics Technologies Ltd.

Legal Notice | CodeMeter License Central WebDepot v21.04a.276.501.ws4 

2. 选中许可后点击 *Activate Selected Licenses Now* 进行许可下载。

English

Home | My Licenses

Available Licenses

To activate your licenses:

1. Select the licenses you want to activate.
2. Select the locally connected CmContainer to which you want to transfer the licenses.
3. Click "Activate Selected Licenses Now".

✓ Name	Activated On	CmContainer	Status
<input checked="" type="checkbox"/>	-		Available

Select CmContainer

Get CmContainer automatically

Activate Selected Licenses Now File-based license transfer

[My Licenses](#)

3. 下载成功后将显示下图所示信息，点击 *OK* 。

Online License Transfer

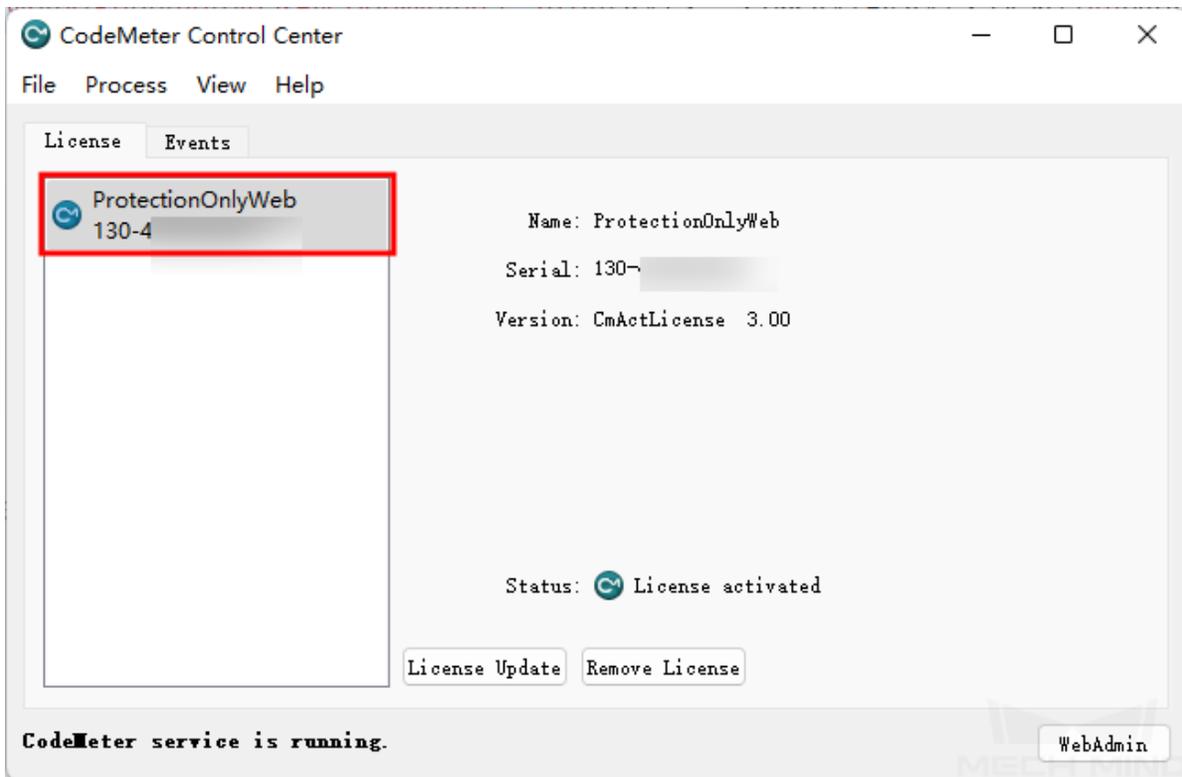
Starting license transfer.
Downloading license template.
Registering license template.
Creating license request.
Downloading license update.
Importing license update to CmContainer.
Creating receipt.
Uploading receipt.



License transfer completed successfully!

OK

4. 打开 CodeMeter 控制中心，确认许可信息。



18.2 更新软件许可

软件许可即将到期时 CodeMeter 会弹窗提示，用户可以联系梅卡曼德技术支持人员（以下简称“梅卡曼德”）来更新许可。

用户需要导出许可请求文件并发送给梅卡曼德，梅卡曼德会返回许可更新文件，用户使用许可更新文件，即可延长许可有效期。

18.2.1 导出许可请求文件

开始操作前，请确保：

- 计算机已安装 CodeMeter。
- 计算机已插上“加密狗”。

1. 点击位于系统托盘的  打开 CodeMeter 控制中心。
2. 选择需要更新的许可后，点击 许可更新。



3. 点击 下一步。

← CmFAS助手

欢迎使用CmFAS助手!

CodeMeter 激活服务 (CmFAS)助手能帮助你增加、改变和删除在CodeMeter许可管理系统中的许可。

CmFAS助手能创建许可请求文件，你能通过email把它发送给软件开发商。你也可以通过CmFAS助手把收到的许可更新文件导入到许可管理系统并且能发送回执给开发商。

下一步(N)

帮助(H)

4. 选择 创建许可请求，点击 下一步。

← CmFAS助手

请选择你希望的操作

创建许可请求

当你想要创建一个许可请求文件并把它发送给软件开发商时选择这个选项。

导入许可更新

如果你收到一个从软件开发商发来的许可更新文件并且想要导入该文件，请选择这个选项。

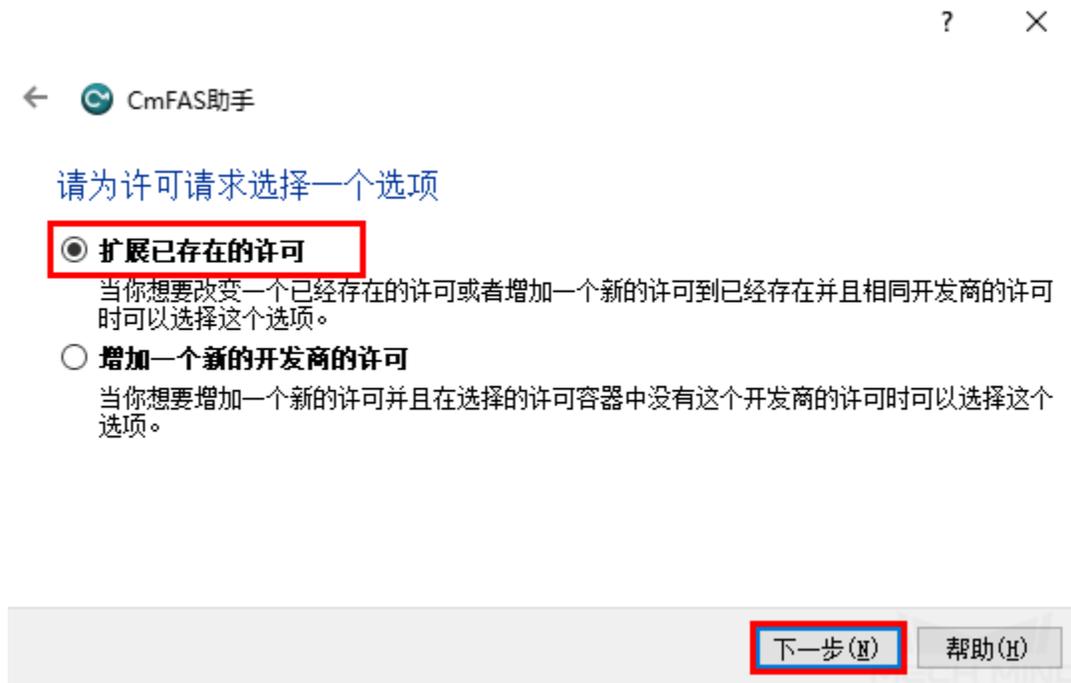
创建回执

当你想要确认是否成功导入从软件开发商那里获得的许可更新文件时选择这个选项。

下一步(N)

帮助(H)

5. 选择 扩展已存在的许可，点击 下一步。

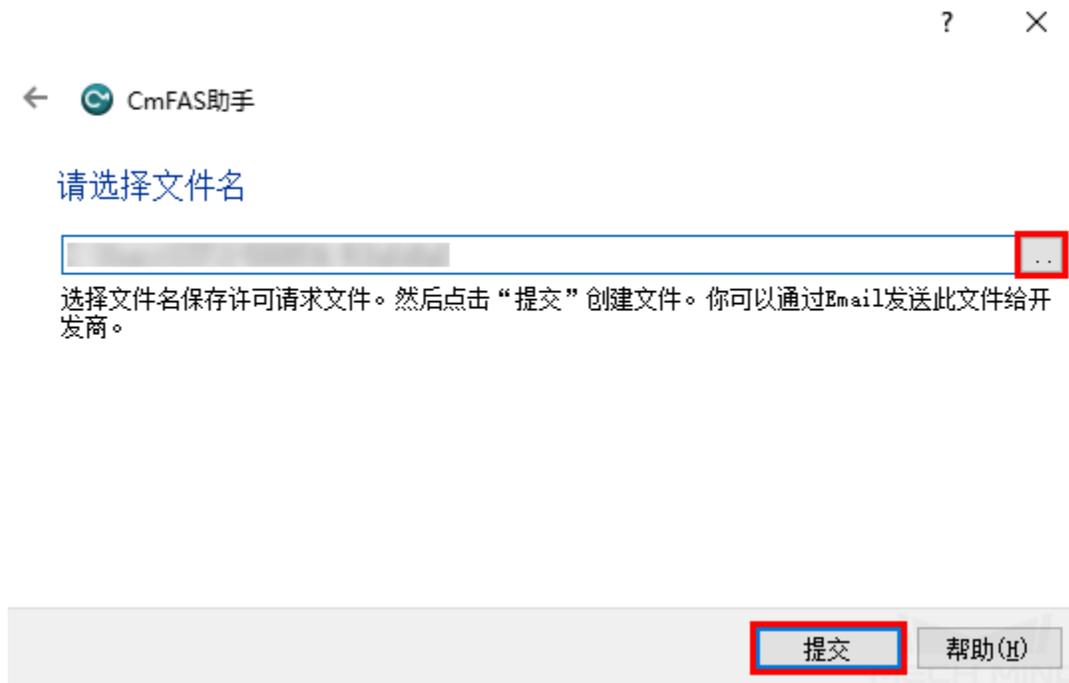


6. 勾选 Mech-Mind，点击 下一步。



注解： 用户实际操作时开发商名称中 **Mech-Mind** 后的公司编码可能与上图所示不一致。

7. 点击 .. 选择保存许可请求文件的位置后，点击 提交。



提示： 如果有多个“加密狗”的许可需要更新，请重复步骤 2 至 7，为每一个“加密狗”导出许可请求文件。

8. 将许可请求文件发给梅卡曼德。

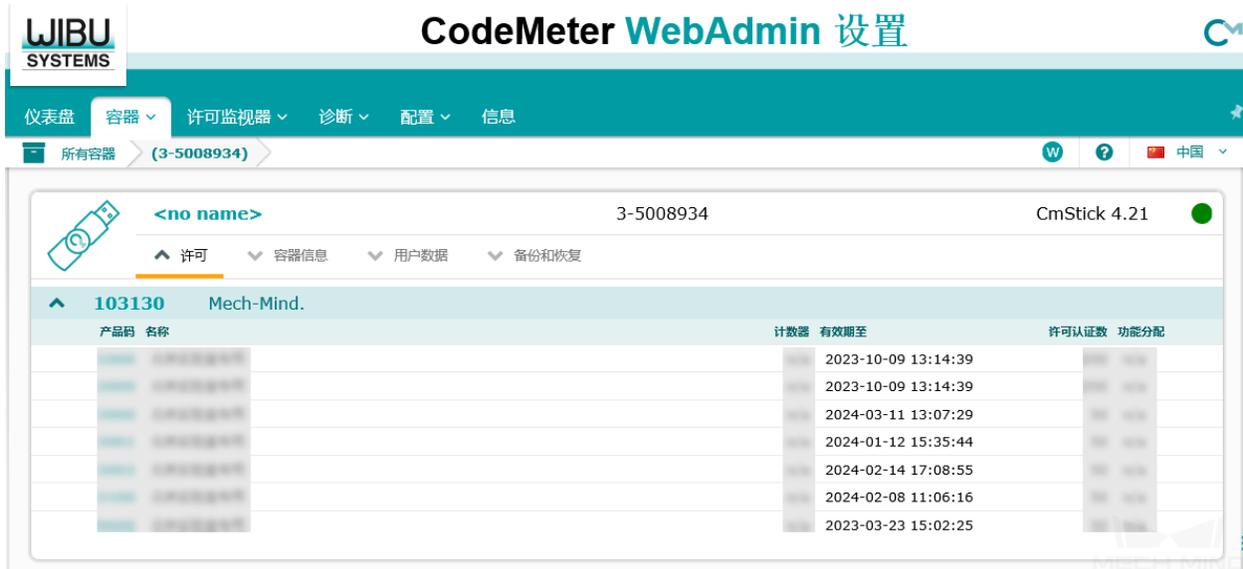
18.2.2 更新许可

用户发出许可请求文件后，梅卡曼德将返回一个 WIBUCMRAU 格式的许可更新文件。双击这个文件即可更新许可，参考以下步骤检查许可是否更新成功：

1. 单击系统托盘的  打开 CodeMeter 控制中心。
2. 选择需要检查的许可，点击右下角的 *Web* 管理界面。



3. 在 CodeMeter WebAdmin 页面中，可以通过查看 有效期时间，检查许可是否已更新。



注解:

- 永久许可的 **有效期至**日期数据是 **n/a**。
 - 点击页面上方的 **容器**可查看其他“加密狗”的信息。
-

若双击 WIBUCMRAU 格式文件无法更新许可，可以参考以下步骤手动导入许可更新文件。

1. 单击位于系统托盘的  打开 CodeMeter 控制中心。
2. 选择需要更新的许可，点击 **许可更新**。



3. 点击 **下一步**。



← CmFAS助手

欢迎使用CmFAS助手!

CodeMeter 激活服务 (CmFAS)助手能帮助你增加、改变和删除在CodeMeter许可管理系统中的许可。

CmFAS助手能创建许可请求文件，你能通过email把它发送给软件开发商。你也可以通过CmFAS助手把收到的许可更新文件导入到许可管理系统并且能发送回执给开发商。

下一步(N)

帮助(H)

4. 选择 导入许可更新，点击 下一步。



← CmFAS助手

请选择你希望的操作

创建许可请求

当你想要创建一个许可请求文件并把它发送给软件开发商时选择这个选项。

导入许可更新

如果你收到一个从软件开发商发来的许可更新文件并且想要导入该文件，请选择这个选项。

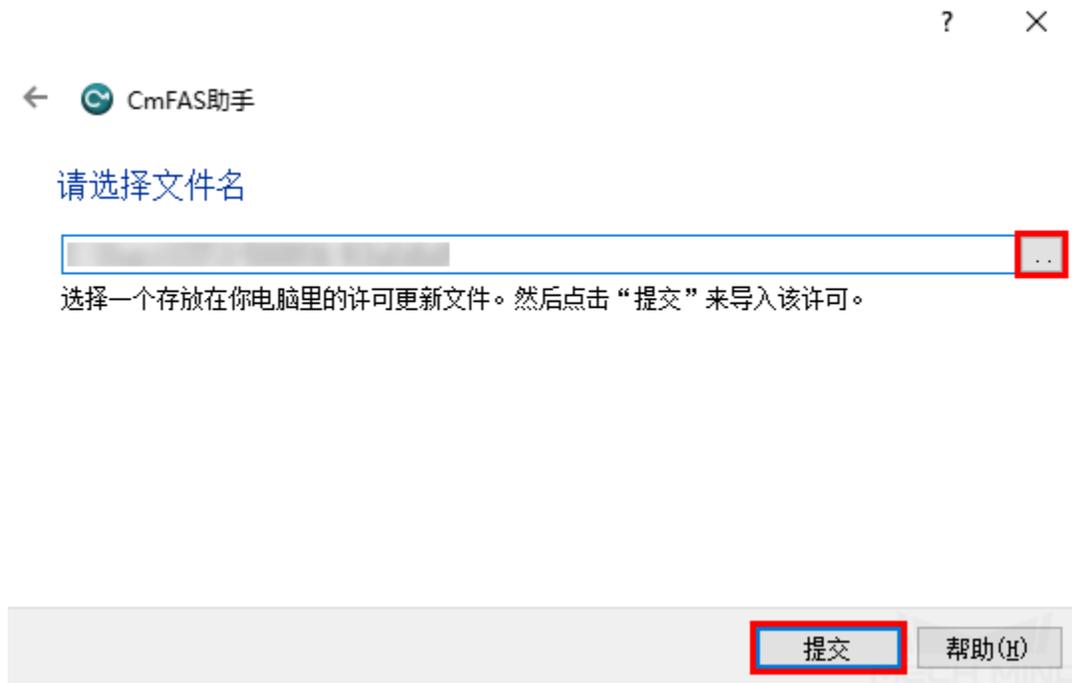
创建回执

当你想要确认是否成功导入从软件开发商那里获得的许可更新文件时选择这个选项。

下一步(N)

帮助(H)

5. 点击 .. 选择许可更新文件，然后点击 提交。



19.1 实例分割

Mech-Vision 版本	深度学习环境版本	Mech-Vision 步骤	模型对应的 Mech-DLK 版本	模型及配置文件后缀
1.4.0	1.4.0	实例分割（需要开启深度学习服务器）	1.4.0	.pth/.py
1.5.x	2.0.0/2.1.0	实例分割（需要开启深度学习服务器）	1.4.0	.pth/.py
			2.0.0/2.1.0	.dlkmp/.dlkcfg
1.6.0	2.0.0/2.1.0	实例分割（需要开启深度学习服务器）	1.4.0	.pth/.py
	2.0.0/2.1.0	实例分割（需要开启深度学习服务器）	2.0.0/2.1.0	.dlkmp/.dlkcfg
	无需安装环境	深度学习模型包推理（Mech-DLK2.2.0+）	2.2.0	.dlkpack
1.6.1	无需安装环境	深度学习模型包 CPU 推理/深度学习模型包推理（Mech-DLK2.2.0+）	2.2.1	.dlk-packC/.dlkpack
1.6.2	无需安装环境	深度学习模型包 CPU 推理/深度学习模型包推理（Mech-DLK2.2.0+）	2.2.1	.dlk-packC/.dlkpack

19.2 图像分类

Mech-Vision 版本	深度学习环境版本	Mech-Vision 步骤	模型对应的 Mech-DLK 版本	模型及配置文件后缀
1.4.0	1.4.0	图像分类（需要开启深度学习服务器）	1.4.0	.pth/.json
1.5.x	2.0.0/2.1.0	图像分类（需要开启深度学习服务器）	1.4.0	.pth/.json
	无需安装环境	深度学习推理	2.0.0/2.1.0	.dlkpack
1.6.0	2.0.0/2.1.0	图像分类（需要开启深度学习服务器）	1.4.0	.dlkpack
	无需安装环境	深度学习推理（Mech-DLK2.1.0/2.0.0）	2.0.0/2.1.0	
		深度学习模型包推理（Mech-DLK2.2.0+）	2.2.0	
1.6.1	无需安装环境	深度学习模型包 CPU 推理/深度学习模型包推理（Mech-DLK2.2.0+）	2.2.1	.dlk-packC/.dlkpack
1.6.2	无需安装环境	深度学习模型包 CPU 推理/深度学习模型包推理（Mech-DLK2.2.0+）	2.2.1	.dlk-packC/.dlkpack

19.3 目标检测

Mech-Vision 版本	深度学习环境版本	Mech-Vision 步骤	模型对应的 Mech-DLK 版本	模型及配置文件后缀
1.4.0	1.4.0	目标检测（需要开启深度学习服务器）	1.4.0	.pth/.py
1.5.x	2.0.0/2.1.0	目标检测（需要开启深度学习服务器）	1.4.0	.pth/.py
	无需安装环境	深度学习推理	2.0.0/2.1.0	.dlkpack
1.6.0	2.0.0/2.1.0	目标检测（需要开启深度学习服务器）	1.4.0	.dlkpack
	无需安装环境	深度学习推理（Mech-DLK2.1.0/2.0.0）	2.0.0/2.1.0	
		深度学习模型包推理（Mech-DLK2.2.0+）	2.2.0	
1.6.1	无需安装环境	深度学习模型包 CPU 推理/深度学习模型包推理（Mech-DLK2.2.0+）	2.2.1	.dlk-packC/.dlkpack
1.6.2	无需安装环境	深度学习模型包 CPU 推理/深度学习模型包推理（Mech-DLK2.2.0+）	2.2.1	.dlk-packC/.dlkpack

19.4 缺陷检测

Mech-Vision 版本	深度学习环境版本	Mech-Vision 步骤	模型对应的 Mech-DLK 版本	模型及配置文件后缀
1.4.0	1.4.0	缺陷检测（需要开启深度学习服务器）	1.4.0	.pth/.py
1.5.x	无需安装环境	深度学习推理	2.0.0/2.1.0	.dlkpack
1.6.0	无需安装环境	深度学习推理（Mech-DLK2.1.0/2.0.0）	2.0.0/2.1.0	.dlkpack
		深度学习模型包推理（Mech-DLK2.2.0+）	2.2.0	
1.6.1	无需安装环境	深度学习模型包推理（Mech-DLK2.2.0+）	2.2.1	.dlkpack
1.6.2	无需安装环境	深度学习模型包推理（Mech-DLK2.2.0+）	2.2.1	.dlkpack

19.5 快速定位

Mech-Vision 版本	深度学习环境版本	Mech-Vision 步骤	模型对应的 Mech-DLK 版本	模型及配置文件后缀
1.6.0	无需安装环境	深度学习模型包推理（Mech-DLK2.2.0+）	2.2.0	.dlkpack
1.6.1	无需安装环境	深度学习模型包推理（Mech-DLK2.2.0+）	2.2.1	.dlkpack
1.6.2	无需安装环境	深度学习模型包推理（Mech-DLK2.2.0+）	2.2.1	.dlkpack

ROI: 感兴趣区域 (ROI) 是从图像中选择一个图像区域，该区域是图像分析所关注的重点，截取 ROI 后可以减少处理时间，提高精度。

标注: 标注指用户使用工具手动选择图像中的区域，以标明特征或缺陷，并对所选区域添加标签，或直接对整张图像添加标签的过程。通过标签告诉工具，这是它需要学习的内容。

数据集: Mech-DLK 导出的格式为 dlkdb 的带有标注信息的文件。

未标注: 未经过标注的图像数据。

训练集: 人工手动标注过的图像数据集，用于模型训练。

验证集: 人工手动标注过的图像数据集，用于模型效果测试。

OK 图: 没有缺陷的图像。

NG 图: 有缺陷的图像。

训练: 用“训练集”图像训练深度学习模型的过程。

验证: 用已经训练好的模型对数据进行计算，并给出结果。

精度: 模型在预测验证集时，其正确预测的样本数占总样本数的比值。

损失: 估量模型预测验证集的结果与真实结果的不一致程度。

轮次: 深度学习算法在整个训练数据集中学习的次数。

过检 (FP): 实际不含缺陷的图被预测为含缺陷的图。

漏检 (FN): 实际含缺陷的图被预测为不含缺陷的图。

1. 缺陷分割模型效果不好，如何排查原因？

1. 检查标注是否有误。
2. 检查训练集中是否包含所有种类的缺陷。
3. 检查图像输入尺寸是否合理，若缺陷太小可能无法有效训练模型。

2. 通过主动调整相机曝光或者补光的方式模拟环境变化采集数据是否可行？

不可行。若现场早晚光照会发生变化，应该分别采集实际现场不同光照条件下的数据，人工制造的数据无法确定可参考性。

3. 拍照位置固定，来料位置会稍有变动，通过移动拍照位置来模拟工件位置变动情况是否可行？

不可行。相机位置在采集数据前应该固定好，移动相机会影响深度学习模型效果且会影响相机外参，此种情况可在训练时适当截大感兴趣区域。

4. 原相机数据质量不好，更换相机后是否需要加入原始数据迭代模型？

不需要。需要重新采集更换相机后的数据，再进行模型训练。

5. 更换背景是否会影响深度学习效果？

会影响。背景变化会导致模型出现误识别或漏识别的情况，所以前期确认背景后最好不要再更换。

6. 相机型号、高度不同的情况下，采集的数据是否可以放在一起训练模型？

可以，需注意 ROI 问题，不同高度 ROI 分别截取，减小差异。

7. 对于高反光金属件，采集数据时需要注意哪些情况？

需注意数据光照不可以过曝或过暗，对于不可避免的局部过曝的情况，需保证工件轮廓清晰。

8. 模型识别效果不好如何排查原因？

可以从训练数据的数量及质量、数据多样性、现场设置的 ROI 参数以及现场光照等方面排查。

1. 数量：训练数据量是否足够使模型达到良好的效果。
 2. 质量：数据质量是否达标，是否足够清晰且无过曝过暗的情况。
 3. 数据多样性：数据是否都涵盖了现场可能出现的所有情况。
 4. ROI 参数：现场设置的数值是否确保与训练时设置的一致。
 5. 光照：光照是否发生变化，是否与采集时保持一致。
9. 现场光照复杂，因为物体被阴影遮挡造成的模型识别效果不稳定应如何改善？
- 可根据现场实际情况采取遮光或补光的措施。
10. 为什么现场数据与训练数据 ROI 设置不一致会影响实例分割置信值的高低？
- 与训练数据 ROI 不一致会使物体不在模型的最佳识别大小范围内，因而影响置信值，所以实际应用时数据 ROI 设置应与训练数据 ROI 保持一致。
11. 纸箱超级模型是什么？
- 针对纸箱拆/码垛场景提供了“超级模型”（点击下载），可以直接应用到大多数现场，在无需采集图像训练的情况下就可以正确分割绝大多数纸箱。
12. 纸箱超级模型适用于哪些场景？
- 适用于颜色、花纹单一或多样的纸箱拆、码垛场景；需注意此模型只适用于纸箱在同一平层时水平摆放，没有倾斜放置的情况。
13. 纸箱超级模型该如何采集数据？
- 先用纸箱超级模型测试一遍，如果无法完全正确分割，则针对问题数据种类有针对性的采集 20 张左右。
14. 无 GPU 的情况下，图像分类模型是否能使用？
- 不能。
15. 用新版 Mech-DLK 打开旧工程验证时，验证结果出现偏移如何解决？
- 重新点击一下 验证即可。
16. 使用 Mech-DLK 训练模型时，软件报错显示 `ModuleNotFoundError: No module named 'onnxruntime'` 怎么解决？
- 进入 C 盘的“用户”文件夹，打开当前用户文件夹。
- 按如下路径检查用户 `site-packages` 目录下是否为空，若不为空，请手动删除该目录下所有内容。
- `AppData/Roaming/Python/Python36/site-packages`