

---

**Mech-DLK**

**Mech-Mind**

2022 年 05 月 12 日

---

## Contents

---

<b>1</b>	<b>快速了解深度学习</b>	<b>4</b>
<b>2</b>	<b>深度学习方法介绍及应用流程</b>	<b>7</b>
<b>3</b>	<b>将深度学习用于典型项目</b>	<b>45</b>
<b>4</b>	<b>Mech-DLK 使用指南</b>	<b>50</b>
<b>5</b>	<b>FAQ</b>	<b>65</b>

梅卡曼德机器人已使用深度学习算法结合成熟的视觉产品为全球各地的汽车、3C、制造、物流等行业的客户提供了高效全面的解决方案，解决了多种关于 **智能识别** 的细分需求，如图像分类、缺陷检测、属性识别等。

## 入门

快速了解深度学习

## 深度学习方法介绍及应用流程



## 实例分割

识别图片中目标物体的轮廓、位置及类别



图像分类

识别目标物体所在图片的类别

将深度学习用于典型项目



纸箱拆/码垛项目应用

视觉引导纸箱拆/码垛项目流程



麻袋拆/码垛项目应用

视觉引导麻袋拆/码垛项目流程

Mech-DLK 软件使用指南 & 深度学习环境配置

*Mech-DLK* 使用指南

深度学习训练软件，用于构建深度学习模型

环境配置

深度学习环境配置及排错

---

常见问题

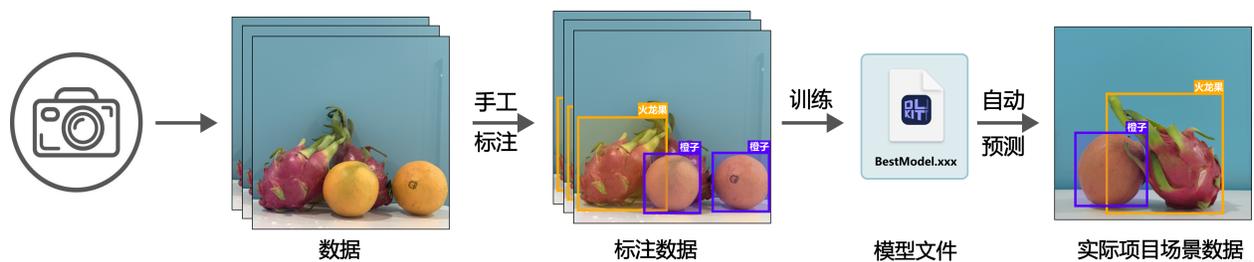
*FAQ*

解答项目常见问题

## 快速了解深度学习

深度学习，即是“通过学习数据的特征解决问题”。

把需要识别的物体拍成图像（得到的图像称之为“数据”），再在图像上标记出需要识别的特征（此过程称为“标注”），通过深度学习产品使用图像及标注信息学习并总结物体特征规律（这一过程称为“训练”），继而得出一个结果（以文件形式体现，称为“模型”），将结果应用到与“数据”内容相似的实际项目场景中（称为“预测”），用以解决不同的识别问题。



深度学习可以解决哪些问题？

针对不同问题，深度学习产品提供了不同的解决方案：

	实例分割	图像分类
典型应用	纸箱、麻袋、周转箱拆码垛工件上、下料货品拣选快递包裹供包…	工件上、下料区分种类、大小、朝向等装配、拣料区分物体是否正确摆放纸箱、麻袋拆码垛区分不同颜色类别的纸箱…
应用特点	定位分类	分类

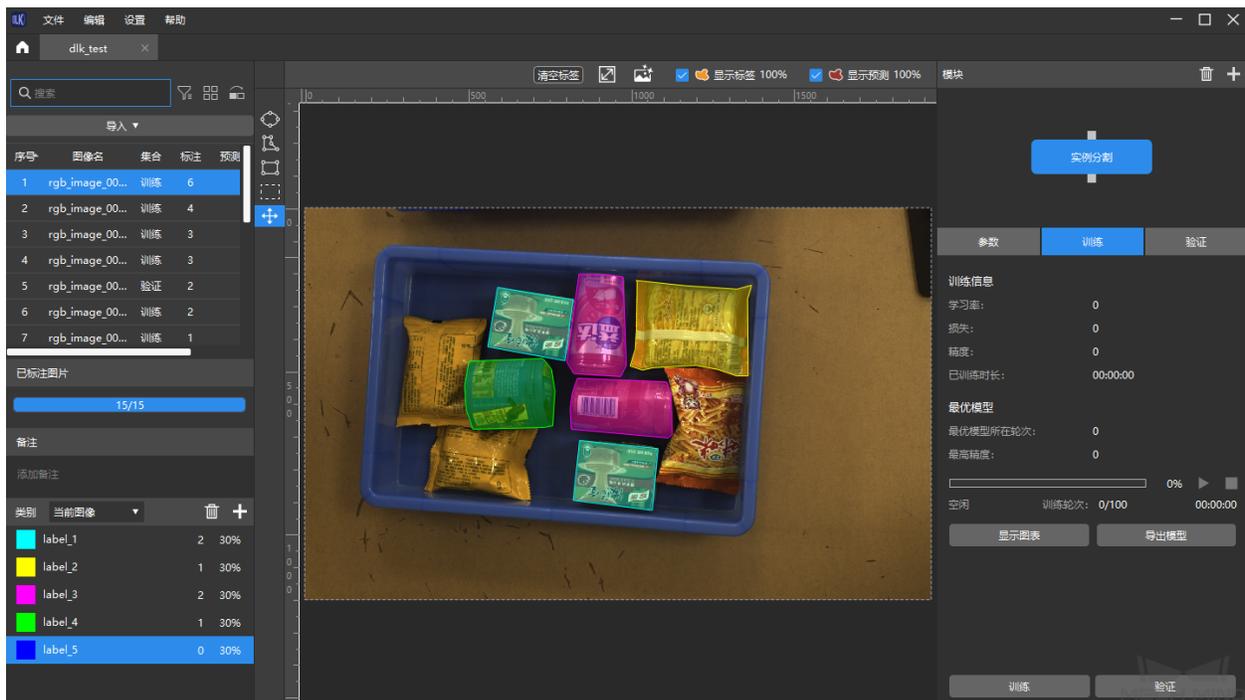
**注意：**深度学习不同的算法都有各自擅长的方向，这些方向并不互斥，是选择其中的一种还是混合使用取决于当前项目的应用场景及具体需求等。

深度学习流程一般包含以下五个步骤：

**Step1 安装环境：**参阅[环境配置](#)。

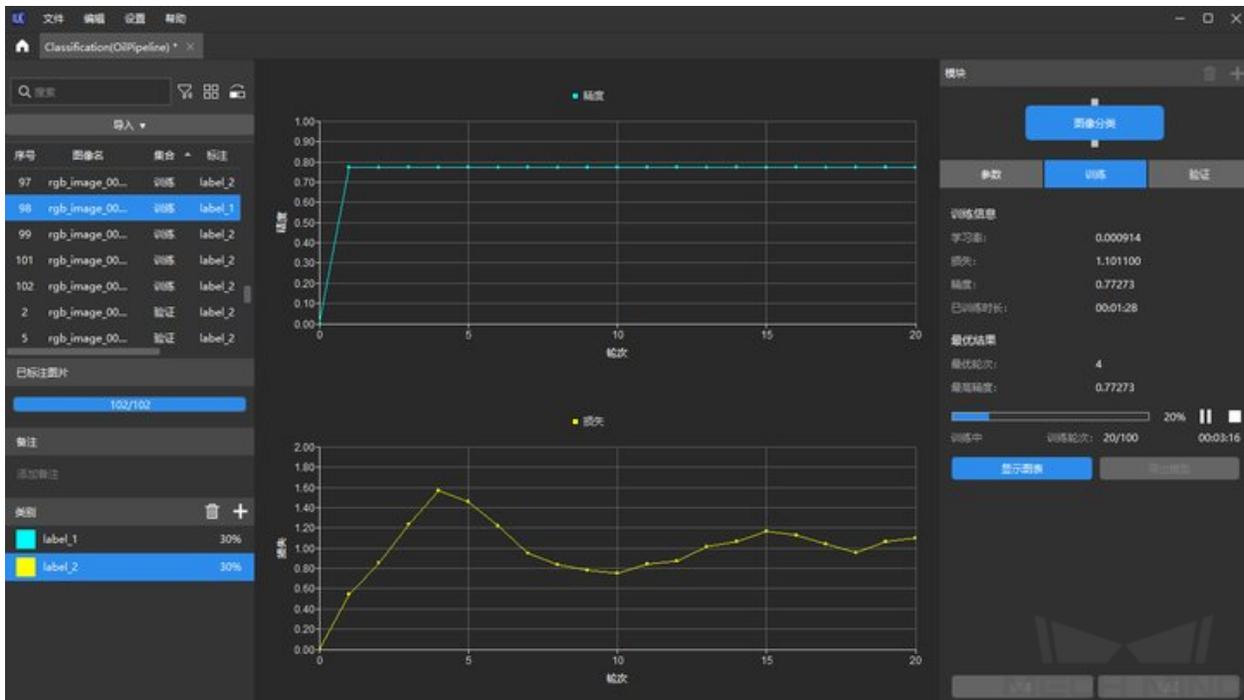
**Step2 准备数据：**采集、标注、检阅数据。

1. 确保与实际应用场景完全一致的情况下采集深度学习所需的图像数据。
2. 根据项目需求选择算法后按规则标注数据集。
3. 审阅是否有标记错误的的数据 (使用 Mech-DLK 可以高效地准备训练所需的数据集)。



Mech-DLK 标注数据

**Step3 训练模型：**用标注过后的数据在 Mech-DLK 训练所需的深度学习模型，详细步骤请参阅[快速入门](#)。



Mech-DLK 训练深度学习模型

**Step4 评估效果:** 用事先预留的测试集验证模型效果是否满足需求。

**Step5 预测:** 用户可将评估后的模型应用到项目的新图像中。

## 2.1 实例分割

### 2.1.1 整体介绍

#### 实例分割能做什么

实例分割要解决的的问题是“有什么，是什么以及在哪里”。  
即判断一张图中是否有目标物体，是什么种类，在图中什么位置，  
例如：

- 目标物体是纸箱，则判断图片中是否有纸箱（模型不会识别纸箱以外的物体），有则标出每个纸箱的轮廓，并输出“纸箱”标签代表类别（标签在对图片标记特征时已制定好），没有则不会给出任何结果。



- 目标物体是肥皂、牙刷以及洗发水等多类物品，则判断图片中是否有这些物品，有则标出每个物品的轮廓并给对应的标签进行分类，没有则不会给出任何结果。



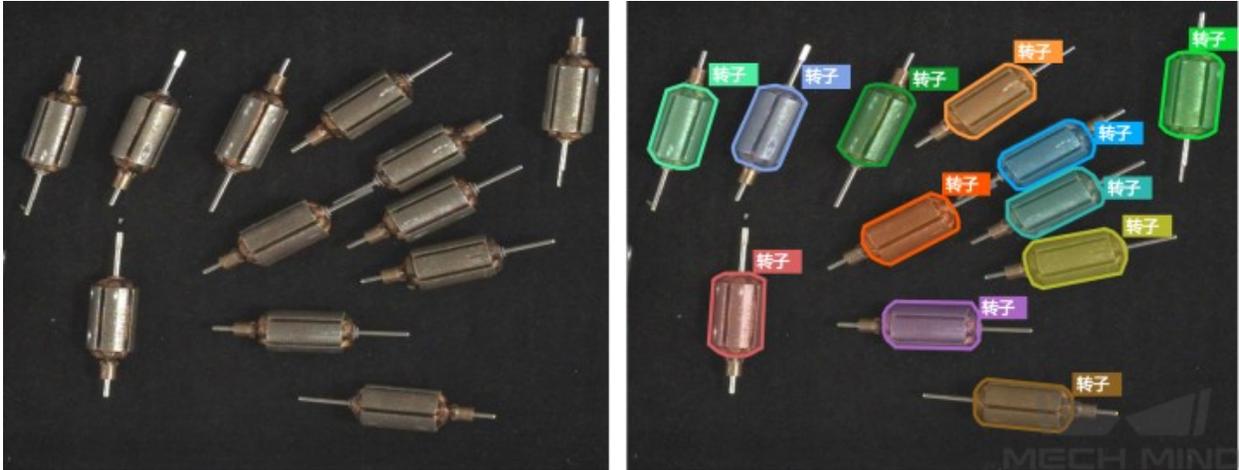
### 实例分割行业典型应用场景

- 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。



麻袋拆码垛项目中分割麻袋

- 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。



工件上下料项目中分割工件

- **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（如锅碗瓢盆）等。



货品拣选项目中分割货品

- **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。



快递包裹拣选项目中分割快递纸箱和包裹

## 实例分割的应用流程

基于深度学习的实例分割，只需用户提供足够多的实际应用场景中物体图片供实例分割模型学习，并标注出图片中每个物体的轮廓和类别，实例分割模型就能够自己学会如何分割。深度学习实例分割的应用流程如下：

- 采集训练所需数据：用相机拍摄足够多相关物体的图片。
- 标注数据：在每张采集的图片上标出每个物体的轮廓和类别。
- 训练模型：把前两个步骤得到的标注后的数据输入到实例分割模型学习。
- 使用模型进行预测：将训练好的模型用到项目中，发挥其功能。

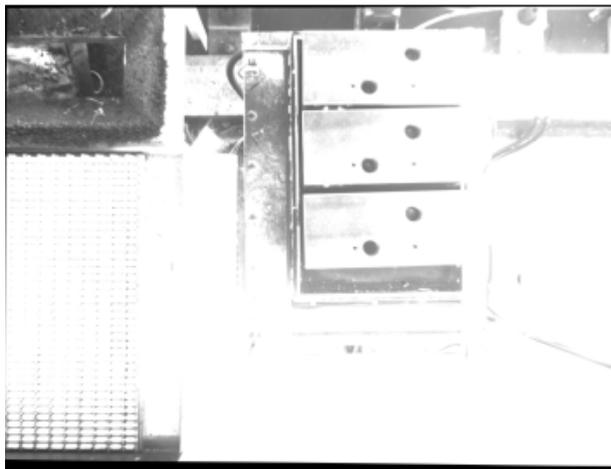
### 2.1.2 采集训练所需数据

**注意：**采集数据是深度学习项目最关键的部分。模型最终效果很大程度上取决于训练数据的质量，高质量的数据集往往能够提高模型训练的质量和预测的准确率。

#### 务必确保采集环境符合要求

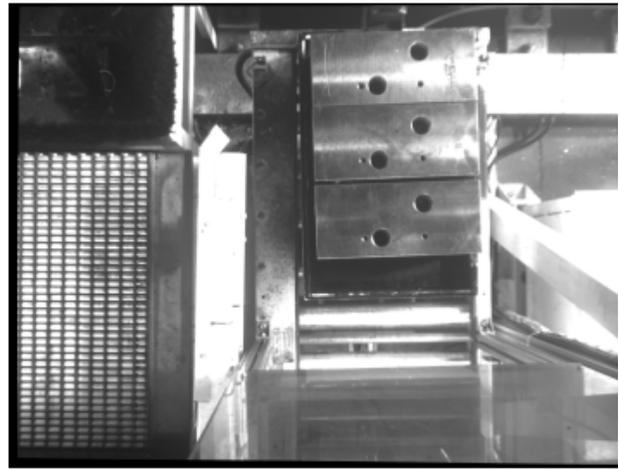
1. 避免 过曝、过暗、颜色失真、模糊、遮挡等。这些情况会导致深度学习模型所依赖的特征丢失，从而影响模型效果。

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

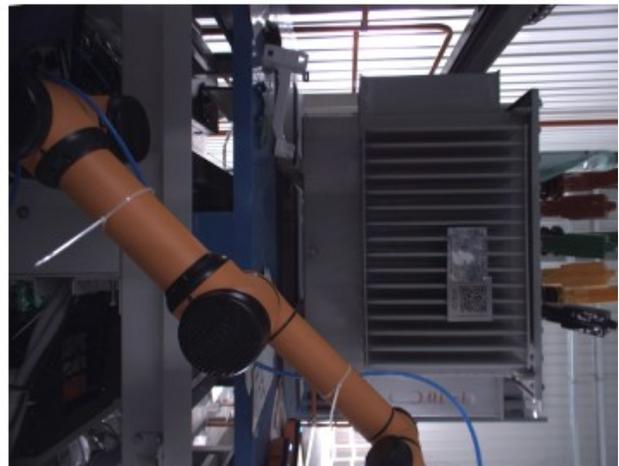


错误示例：过暗



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

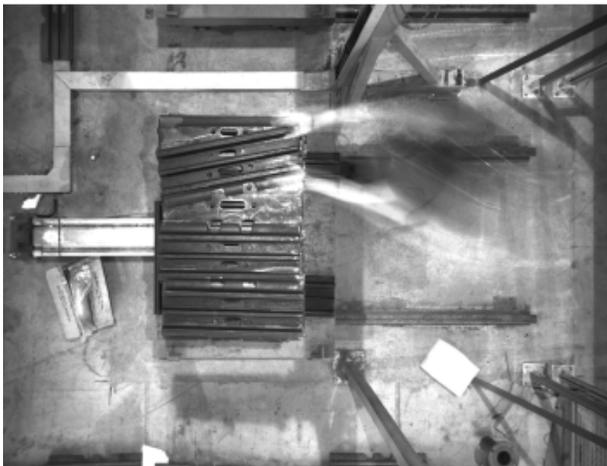


优化建议：可通过调整相机白平衡避免

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



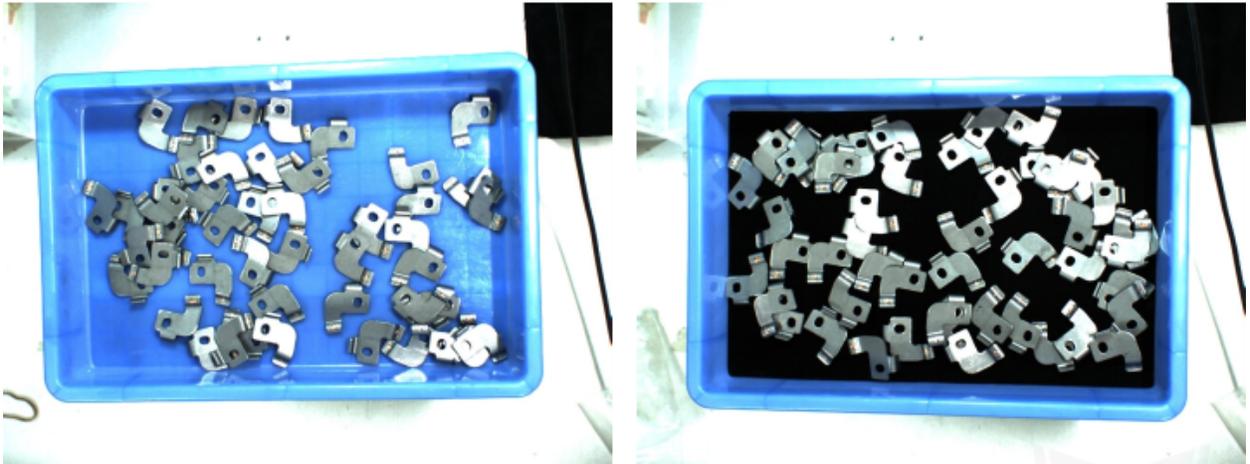
优化建议：确保机械臂和人离开拍照区

错误示例：人手遮挡



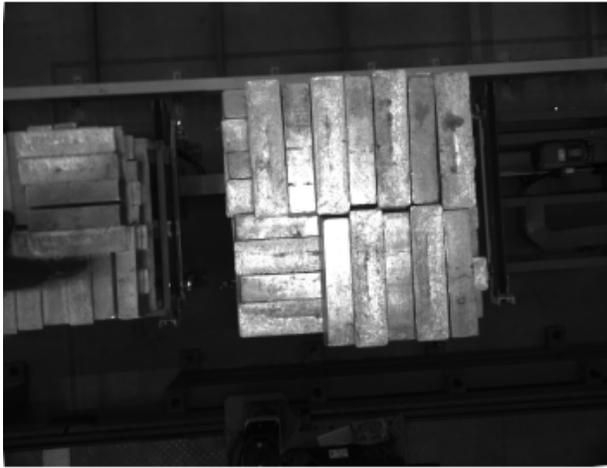
2. 确保采集数据的背景、视角、高度与实际应用一致。任何不一致都会降低深度学习在实际应用时的效果，严重情况下必须返工重新采集数据，请务必提前确认实际应用时的情况。

错误示例：训练数据背景（图左）与实际应用背景（图右）不一致



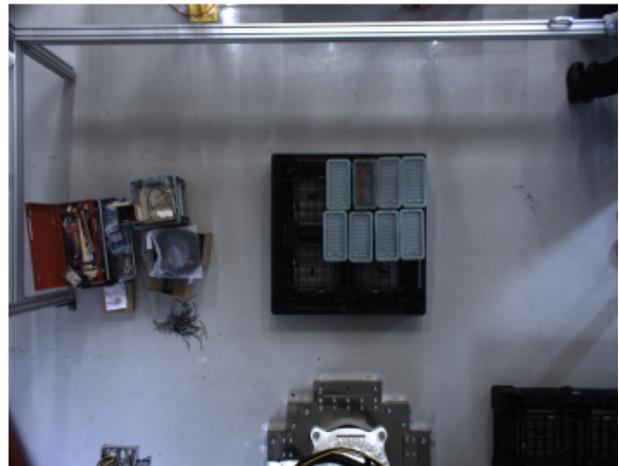
优化建议：保持训练数据背景与实际应用一致

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

### 采集多少张数据

- 如仅有单类物体，采集约 50 张。
- 如有多类物体，每类物体采集约 30 张，采集总量 =  $30 * \text{类别数量}$ 。
- 以上为通用采集量，典型行业应用有更细化的标准，可参考典型项目数据采集案例。

**注意：**训练数据集太少，模型将没有足够多的样本，在此基础上拟合的数据会导致深度学习模型无法有效训练，测试错误率很高；数据太多会导致训练速度不必要地降低，请务必合理分配。

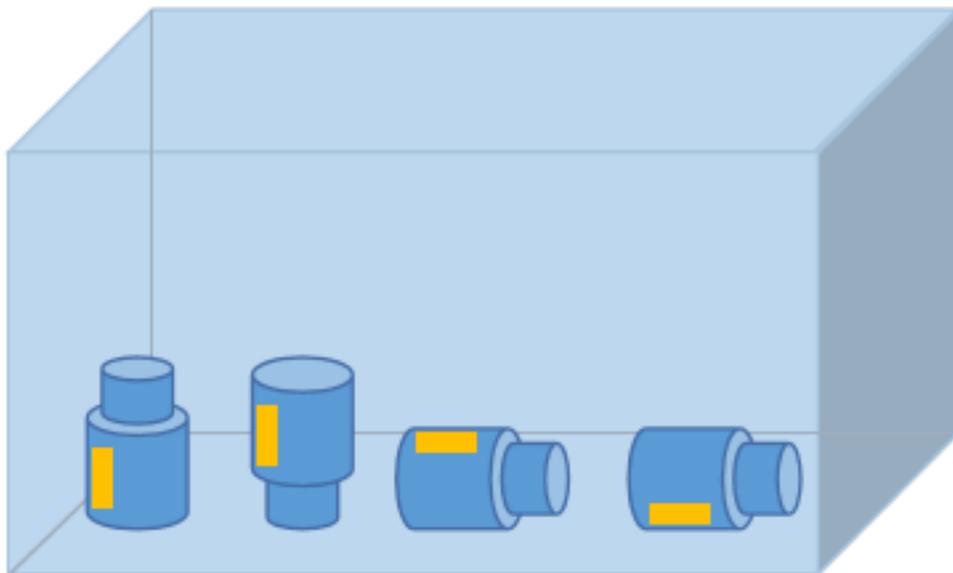
## 采集时的物体摆放

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和竖向，但只采集了横向来料的数据进行训练，那么无法保证竖向来料的识别效果。又如实际生产时来料有可能互相堆叠遮挡，但只采集了单独来料的数据，那么无法保证堆叠遮挡来料的识别效果。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**。

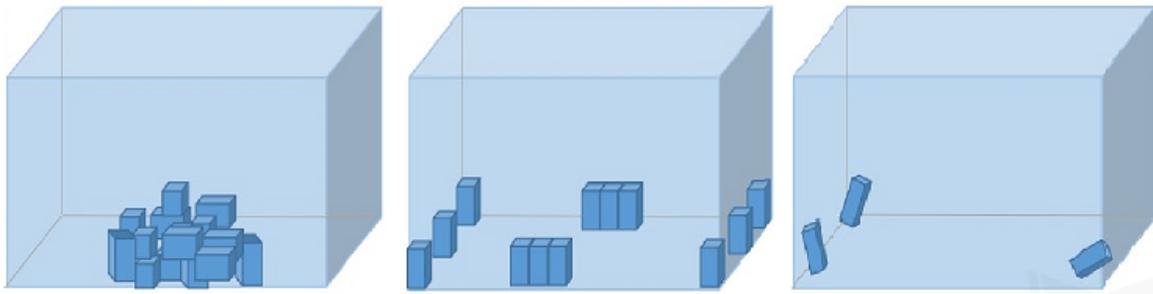
**注意：**如果漏采集了某种情况，会导致深度学习模型缺少对于该情况的学习，从而在该情况下无法有效识别，这意味着需要根据情况增加数据样本，降低误差。

### 1. 物体朝向

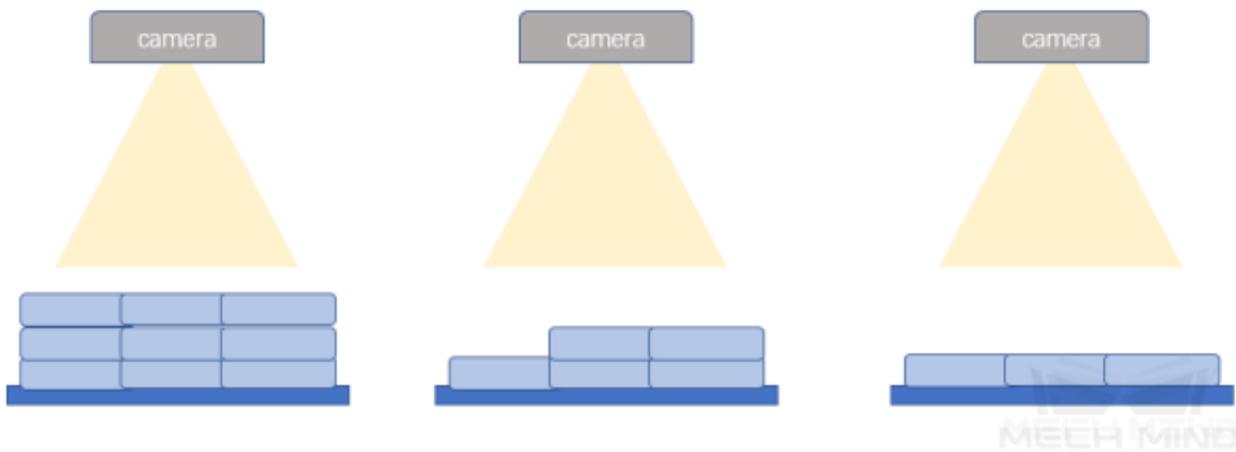


物体不同面朝上

### 2. 物体位置

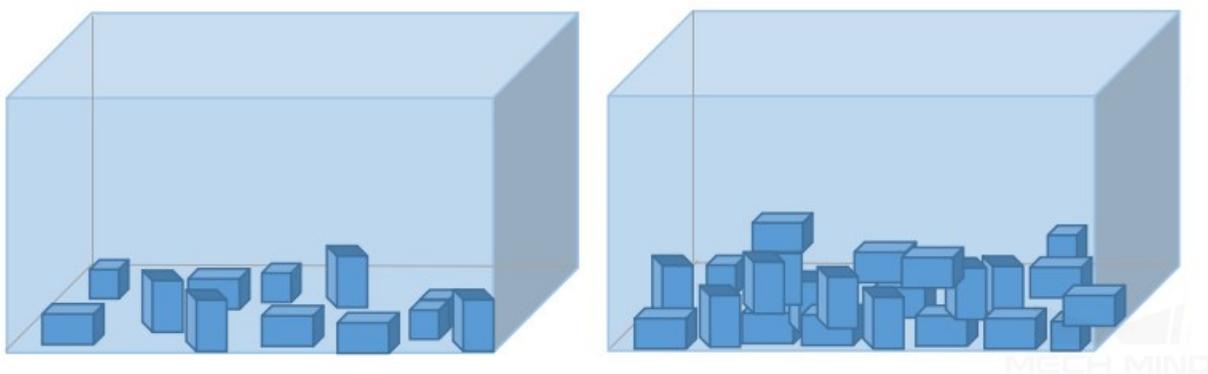


物体位于料筐中央、料筐四周、料筐边角

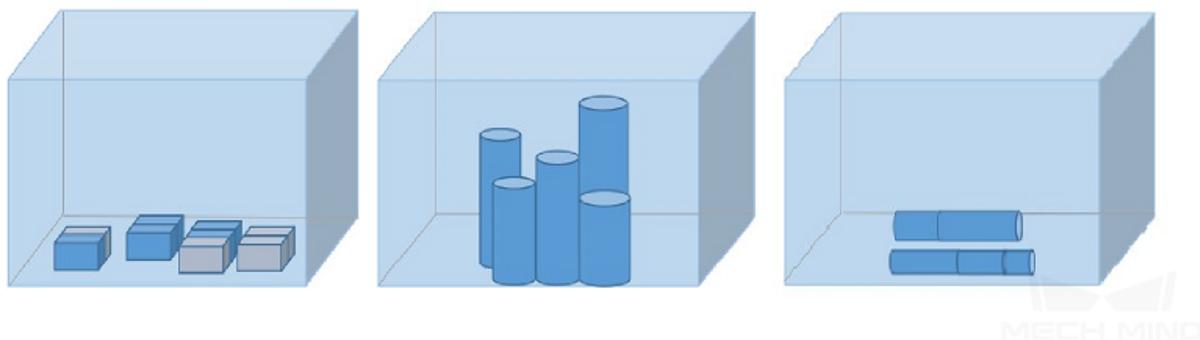


物体位于不同层高

3. 物体间关系



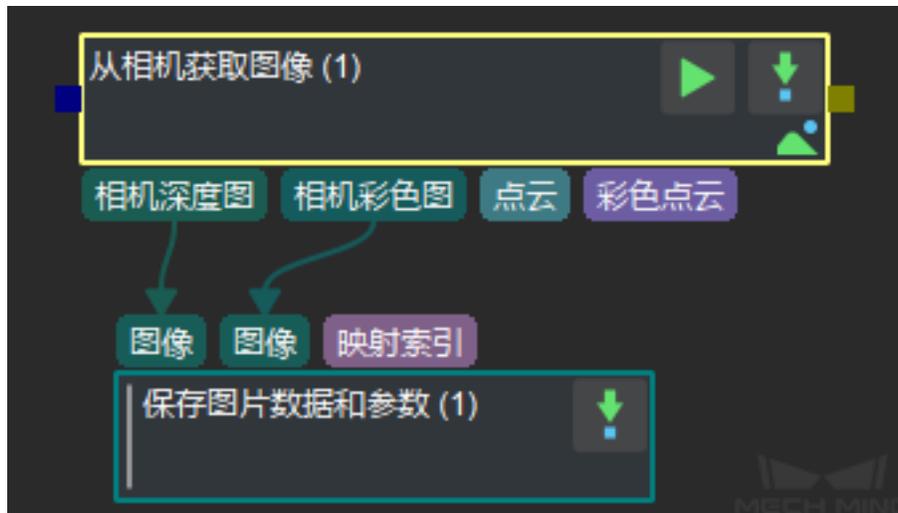
物体间互相平铺、堆叠



物体间紧密贴合

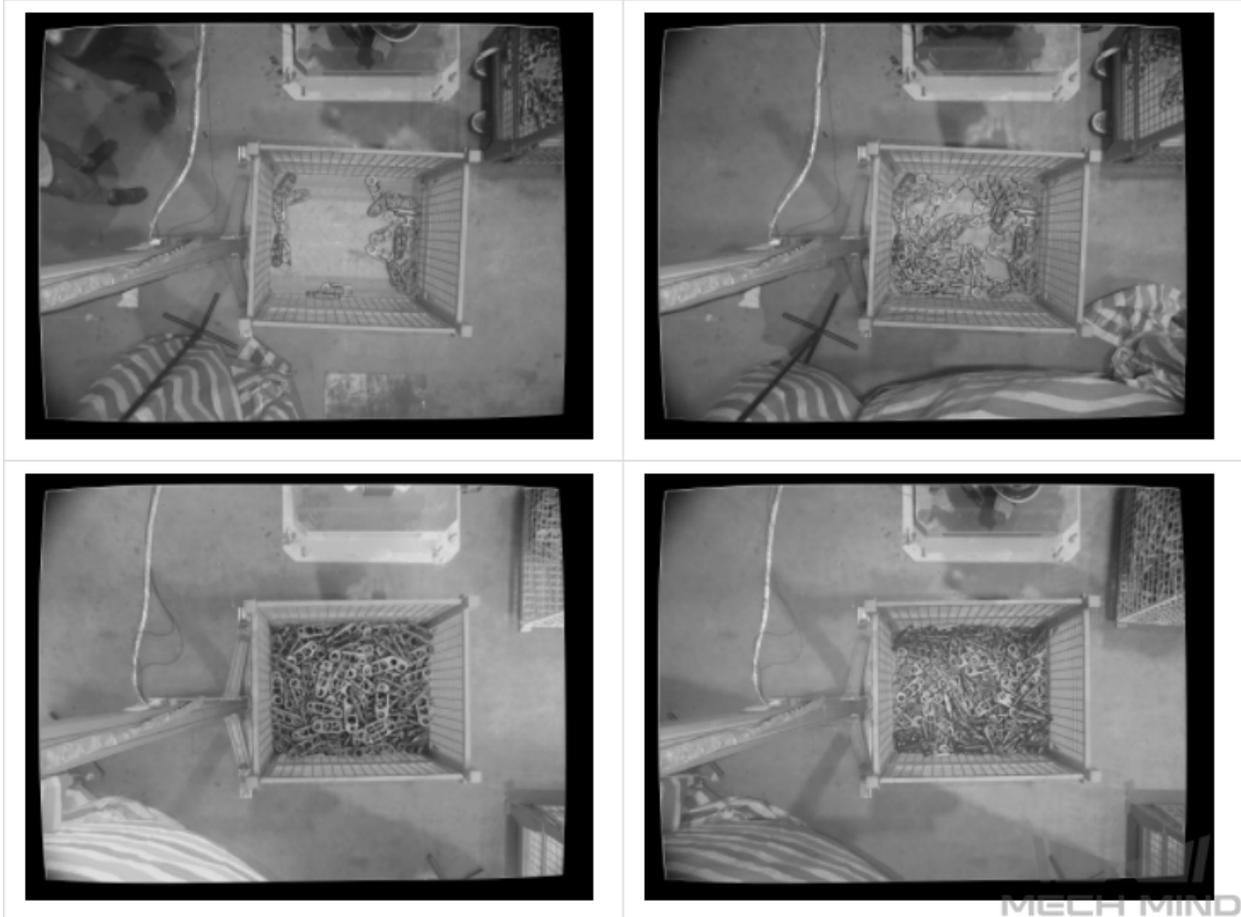
### 使用 Mech-Vision 采集数据

上述采集环境、采集数量、采集时摆放方式均确认后，可使用 Mech-Vision 搭建如下图所示的步骤组合来采集数据。步骤使用详解参阅 `capture_images_from_camera`。



### 历史项目数据采集案例

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：



少量散乱（左上）& 集中散乱（右上）& 堆叠（左下）& 集中散乱（右下）



平铺 & 侧立 & 堆叠 & 并排

- 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：



料筐边角（左上）紧密拼接（右上）紧密贴合（左下）少量散乱（右下）

混合摆放：



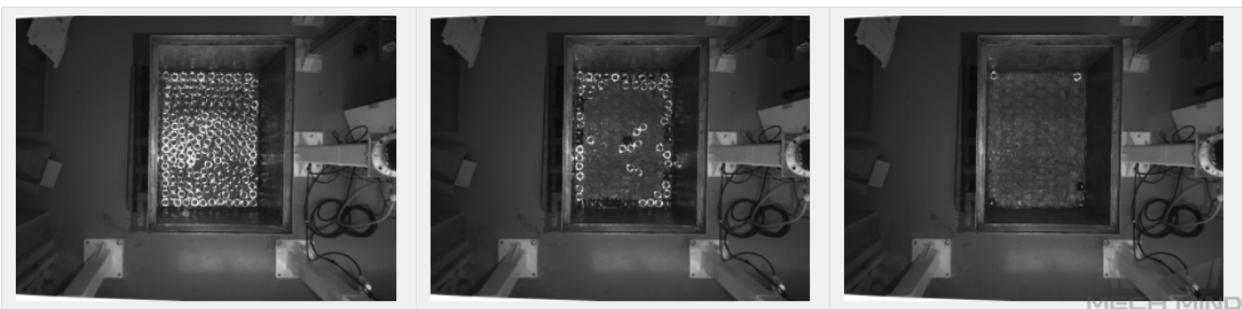
紧密贴合 &amp; 料筐边角 &amp; 散乱堆叠

3. 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，需重点关注紧密贴合的情况。实际采集图片如下：



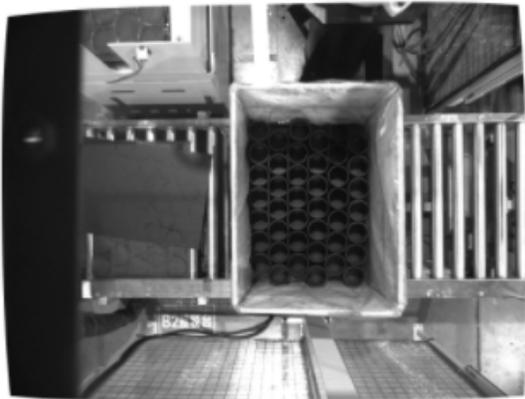
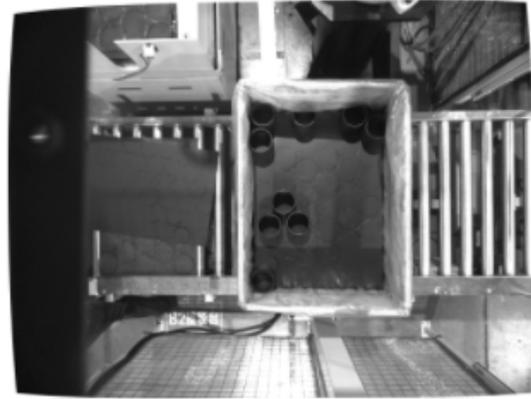
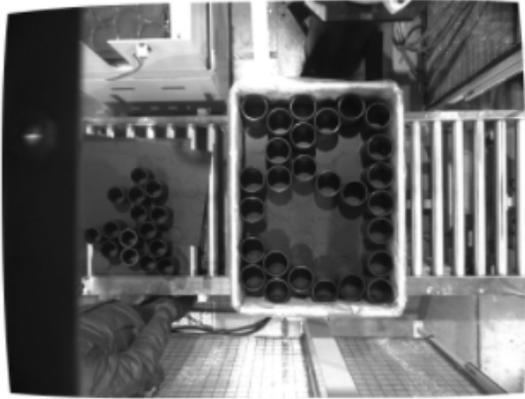
高、中、低层不同高度视野

4. 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：



满层摆放 &amp; 料筐四周 &amp; 料筐边角

5. 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：



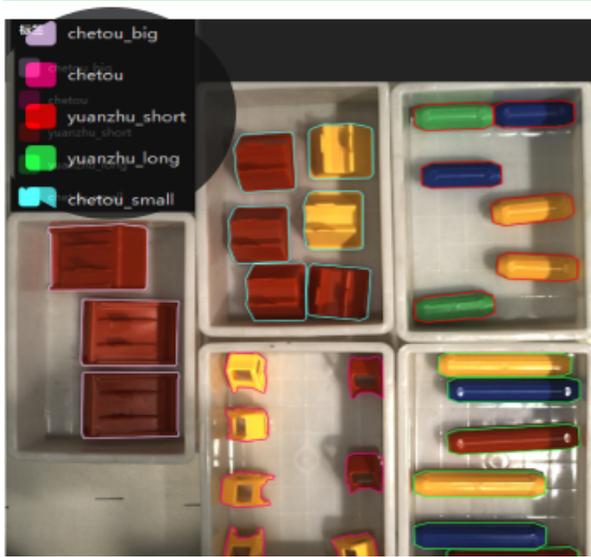
高层(左上)& 高层少量(右上)& 低层全满(左下)& 低层料筐四周(右下)

### 2.1.3 标注数据

#### 创建标签

确认项目需求是否需要分类，若需要分类就根据类别建立相应数量的标签；不需要分类则创建一个标签即可。

## 标签制定

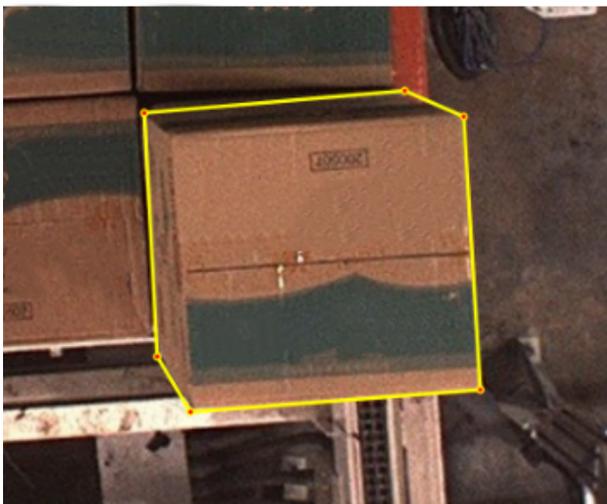


**注意：** 标签名称需要有辨识度，禁止使用 a、b、tmp 等无意义名称；标签名称应只包含英文字母或数字；尽量使用英文单词（如 box）而非汉语拼音（如 xiangzi）以避免歧义。

## 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



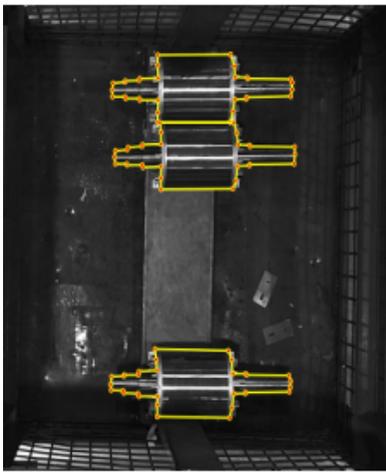
## 正确示例：正确标注



MECH MIND

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。

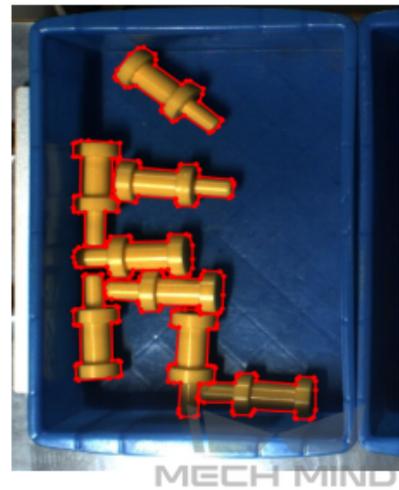
正确示例：标注外轮廓



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。

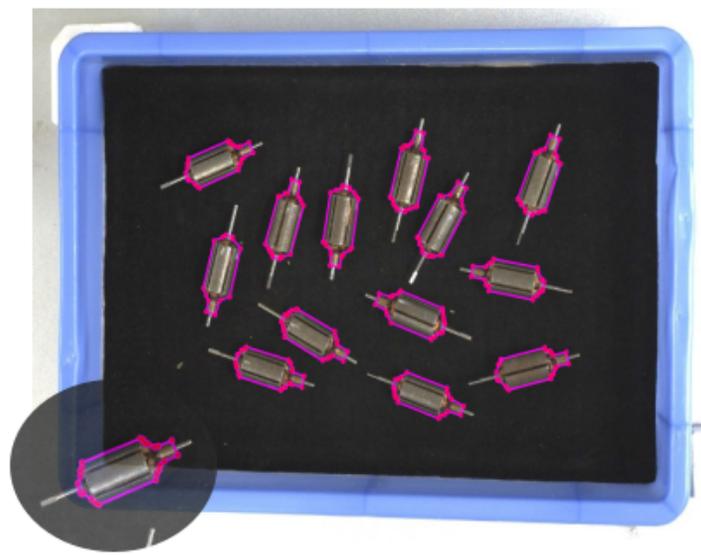
案例 1：需要保证吸盘与瓶口完全贴合（对精度要求很高），只需标注瓶口轮廓。

正确示例：标注瓶口



案例 2：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。

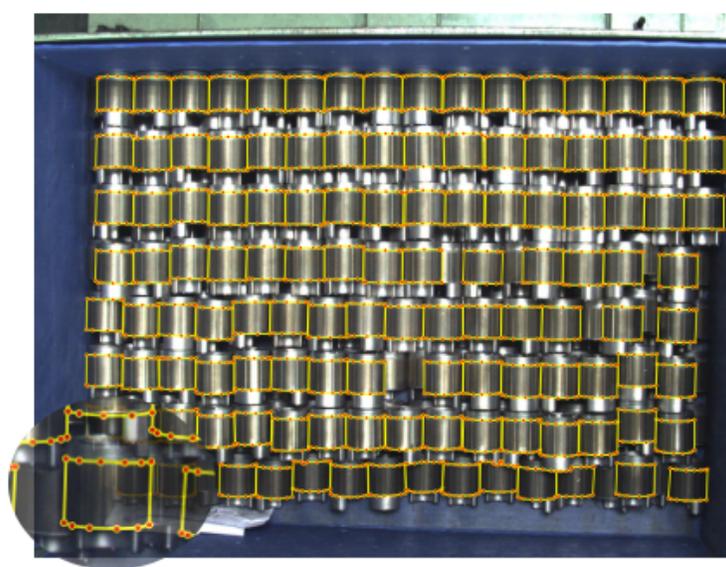
正确示例：标注转子中间位置



MECH MIND

案例 3：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

## 使用 Mech-DLK 标注数据

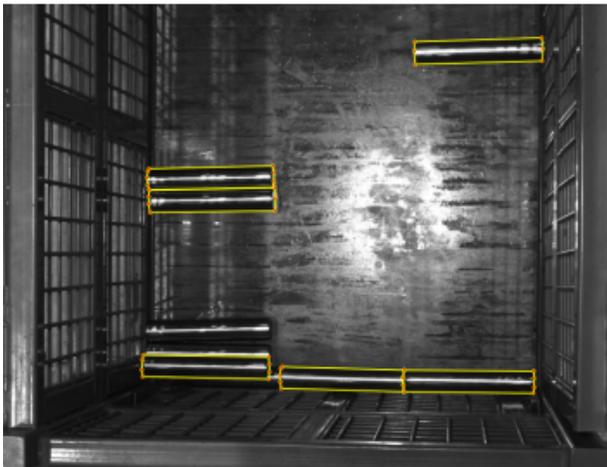
不同的算法标注方式不同，详情请参阅 Mech-DLK 数据标注方法。

**注意：**务必确保标注质量：标注完务必检查一下数据集的标注情况，并验证每个样本的标签是否一一对应，若标签分类错误，在数据集中使用反例会对学习过程产生不利影响，直接影响模型识别效果。

标注质量应从完备性、正确性、一致性、精确性几个方面考虑：

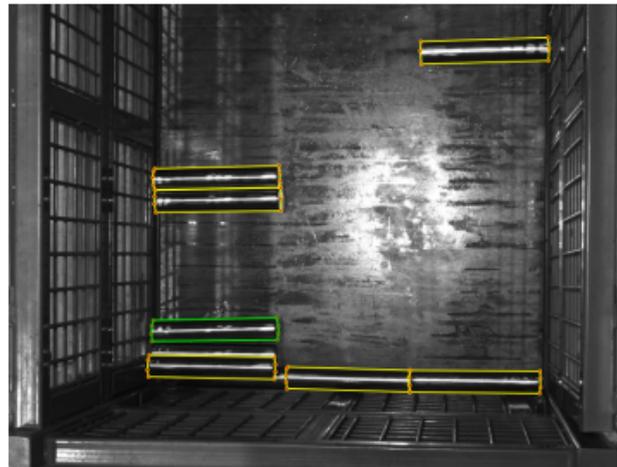
1. **完备性：**标注所有符合规则的物体，禁止出现遗漏标注的情况。

### 错误示例：漏标注



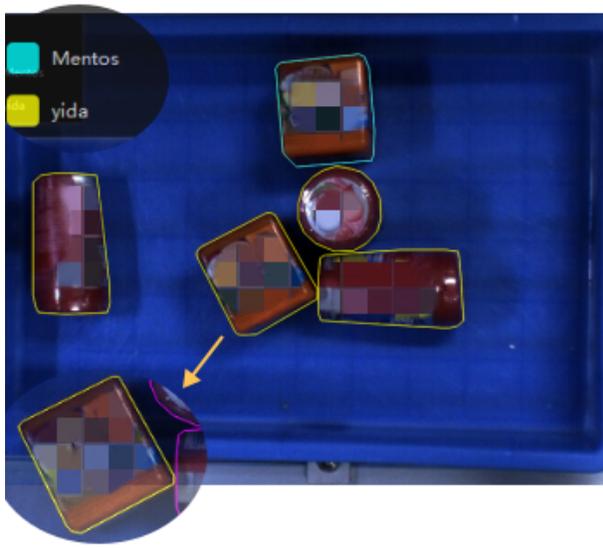
标注提示：要求标注所有完整的钢棒时，不应遗漏某一个

### 正确示例：正确标注

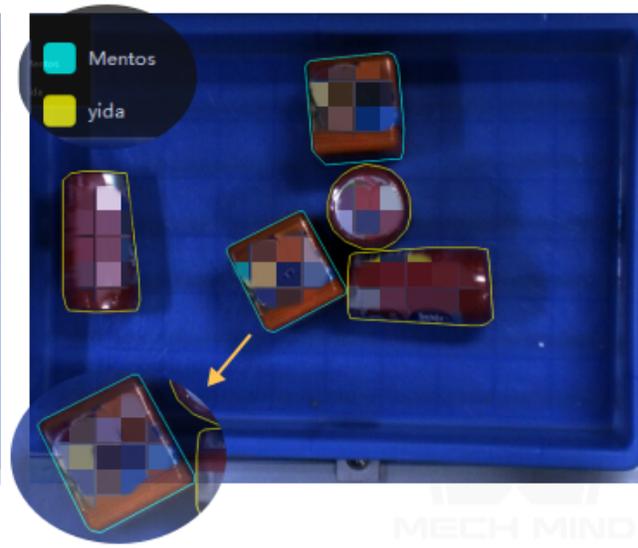


2. **正确性：**确保物体与所属的标签正确对应，禁止出现物体与标签不一致的情况。

错误示例：标签不对应, Mentos被标注为yida

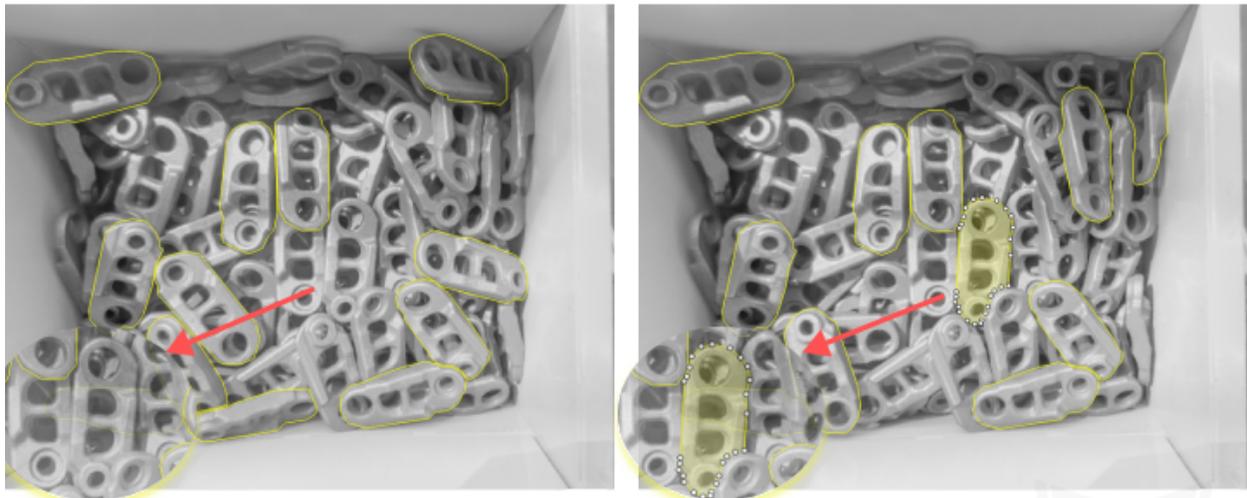


正确示例：正确标注



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。

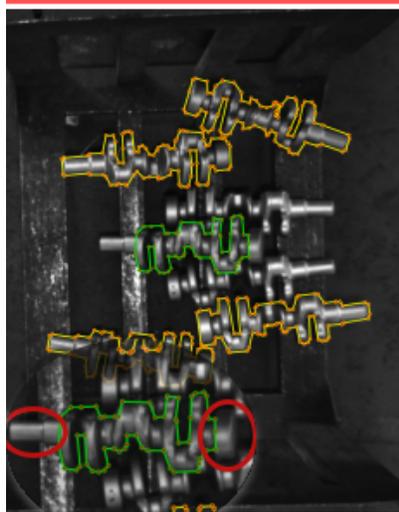
错误示例：相邻两张图片中相同姿态的金属件标注不一致



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。

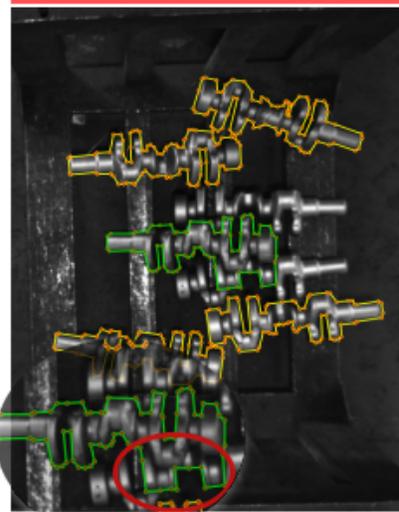
错误示例：少标注



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

MECH MIND

## 2.1.4 训练模型

### 使用 Mech-DLK 训练模型

参考快速入门。

### 可调整的训练参数

多数情况使用默认参数进行训练即可满足需求，以下参数仅在有特殊需求时按需调整。

参数      训练      验证

数据增强参数 ▼

亮度      -60 % ~ 60 %

对比度      -60 % ~ 60 %

平移      -15 % ~ 15 %

旋转      -135 ° ~ 135 °

缩放      -50 % ~ 50 %

翻转

训练参数 ▼

总轮次      < 200 >

学习率      < 0.0050 >

应用      重置

MECH MIND

### 亮度范围

现场光照变化较大，通过补光和遮光也无法弥补时，采集的训练数据光照差异也会比较大。这种数据用于训练需要适当调大亮度范围，差异较小则保持默认参数。

### 对比度范围

物体和背景的差异不明显时为了模型更好地学习物体特征，可适当调大范围，通常与亮度范围配合使用。由于此种情况极少发生，一般无需调整。

### 平移范围

若现场物料背景（料筐、托盘或其他）位置移动范围较大，需调大平移范围，反之则保持默认参数。

### 旋转范围

当物体在固定位置需区分不同方向时，需将范围调为 0-0，避免因旋转而影响模型学习方向特征，其他情况时保持默认参数。

### 缩放范围

当物体位置高度差异大或者同等高度时物体间体积差异较大时需调大缩放范围，其他情况时保持默认参数。

### 翻转

当物体在固定位置需区分不同方向时，需关闭翻转选项，避免因翻转而影响模型学习方向特征，通常与旋转范围配合使用，其他情况时保持默认参数。

### 通道随机混合

依靠颜色分类且形状差异不大时，需关掉此选项，避免因颜色混合变化而影响模型对物体颜色特征的学习，其他情况时保持默认参数。

### 总训练轮次

不分类、物体特征简单时，可设置在 600epoch 内取最优模型；物体姿态复杂，种类多或其他特征复杂时可设置在 1000epoch 内取最优模型。

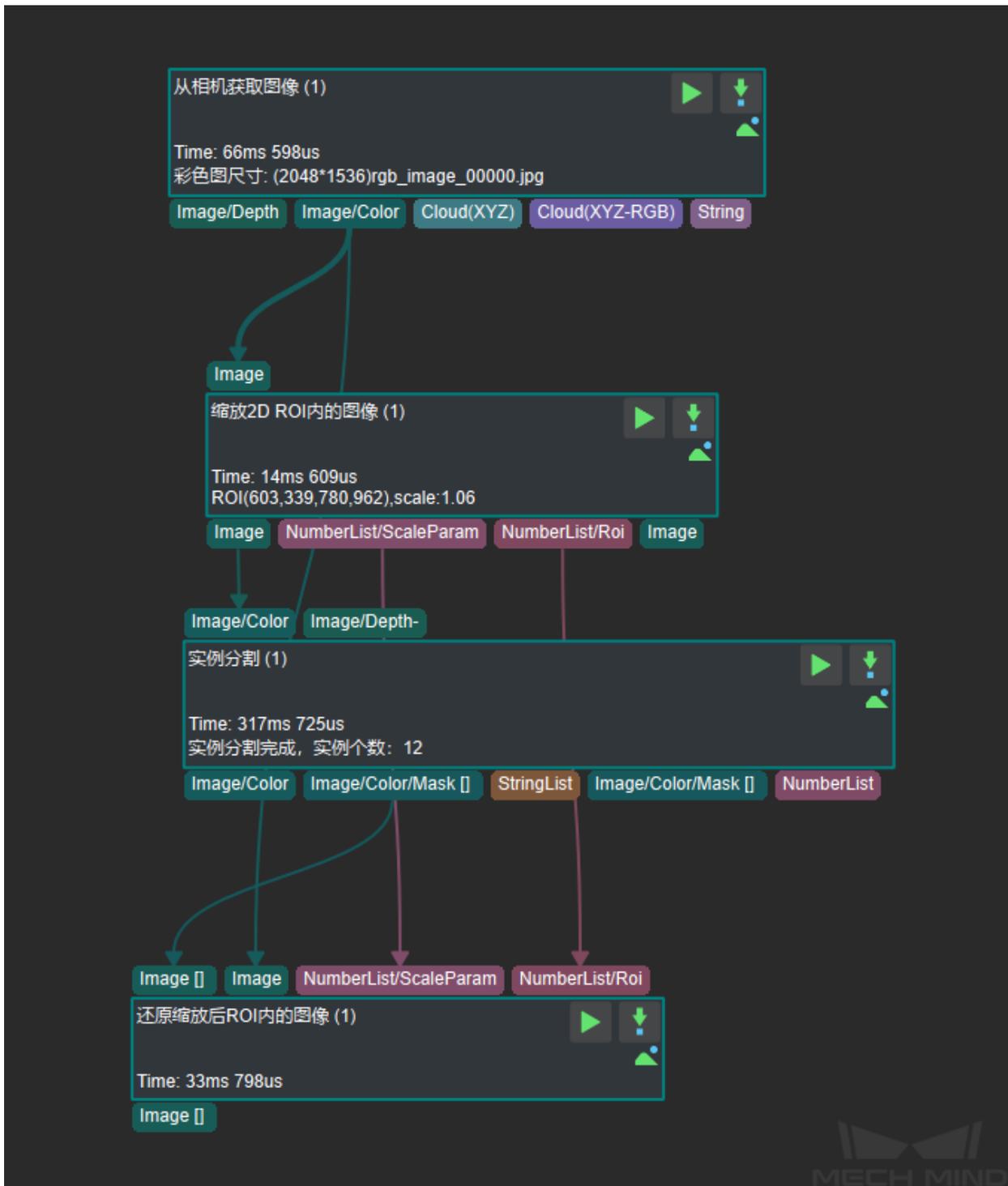
### 学习率

一般情况下无需调整，当准确率数值较低 ( $<0.8$ ) 或突然下降时，可调整为当前学习率数值的十分之一。

**注意：**如果调整为实际没有的范围，反而会影响模型效果；例如当光照稳定时，过多调整亮度范围，模型效果则不如默认参数值下稳定。

## 2.1.5 使用模型进行预测

在 Mech-Vision 中使用实例分割模型



第一步：从相机获取图像。

第二步：缩放 2D ROI 内的图像，使实际数据与训练数据的 ROI 保持一致。需要注意手动缩放 ROI 及自动缩放 ROI 的设置，一般情况下默认自动缩放即可。

▶ 运行标志	
图像输入端口个数 (1 ~ 8)	1
▼ 彩色图ROI设置	ROIByParam ★
Start X	150.0000
Start Y	160.0000
Width	1000.0000
Height	1000.0000
更新彩色图ROI区域的方法	Origin ▼
▼ 彩色图ROI缩放设置	
自动缩放	False
自定义缩放系数	0.6400
▼ 输出图像的理想长宽设置	
与输入彩色图ROI长宽相同 (不缩放)	False
横向长度	1024 pixel
纵向宽度	1024 pixel
▼ 填充图像至目标大小	

第三步：实例分割，输出深度学习分割结果。可根据需求判断分割结果是否符合要求，掩膜是否完整，是否有漏识别和误识别等情况（注意正确填写入模型文件.pth 和配置文件.py）。需注意以下几个参数：

▶ 运行标志	
▼ 服务端与模型设置	
服务端IP地址	127.0.0.1:50052
模型文件	C:/Users/admin/Desktop/Model/
配置文件	C:/Users/admin/Desktop/Model/
▼ 预加载设置	
打开工程时自动预加载模型	False
最多识别个数 (0 ~ 2000)	500
置信度阈值 (0 ~ 1.0)	0.7000
▼ 字体设置	
使用自定义大小	✓ True
字体大小 (0 ~ 10)	2.0
▼ 可视化设置	
在图上画出检测物体	✓ True
可视化不同物体的方式	Classes ▼

### 预加载-最多识别个数

当模型一次需识别几十上百个物体时，需注意调整最多识别个数，如果识别个数少且工程有节拍要求，调低此参数可优化节拍。

### 预加载-置信度阈值

一般情况下，置信度阈值使用默认参数 0.7 即可。当识别出的物体掩膜置信度低于默认参数时，默认该物体不符合抓取要求且置信度字体显示为红色。可根据项目精度需求适当调整此参数，例如置信度数值较低但物体掩膜比较完整符合抓取要求时，可以适当降低置信度参数以增加可抓取物体数量。

### 字体-使用自定义大小

常规情况使用默认值即可，当被识别物较小时，可适当调小字体便于查看识别效果。

### 可视化-在图上画出检测物体

勾选 **True** 选项时可显示出被识别物体的掩膜，便于观察模型识别效果。正式运行时可关掉此选项以提高节拍速度。

### 可视化-可视化不同物体的方式

选择 **classes** 选项，掩膜会根据标签种类显示出不同颜色用于区分不同种类；选择 **Instances** 选项，掩膜会随机根据每个独立物体显示出不同颜色以示物体之间的区别；选择 **Threshold** 选项，掩膜会根据是否低于置信度阈值显示出不同颜色以示置信度是否达标。

---

**小技巧：**若一个工程需要用到两个不同的实例分割模型，则使用两个实例分割步骤，分别配置对应的模型文件和配置文件；另外还需要将服务端 IP 地址分别设置为 127.0.0.1:50052 和 127.0.0.1:50053，否则会导致端口冲突。

---

第四步：还原缩放后 ROI 内的图像（将图像还原为原图大小）。

## 2.2 图像分类

### 2.2.1 整体介绍

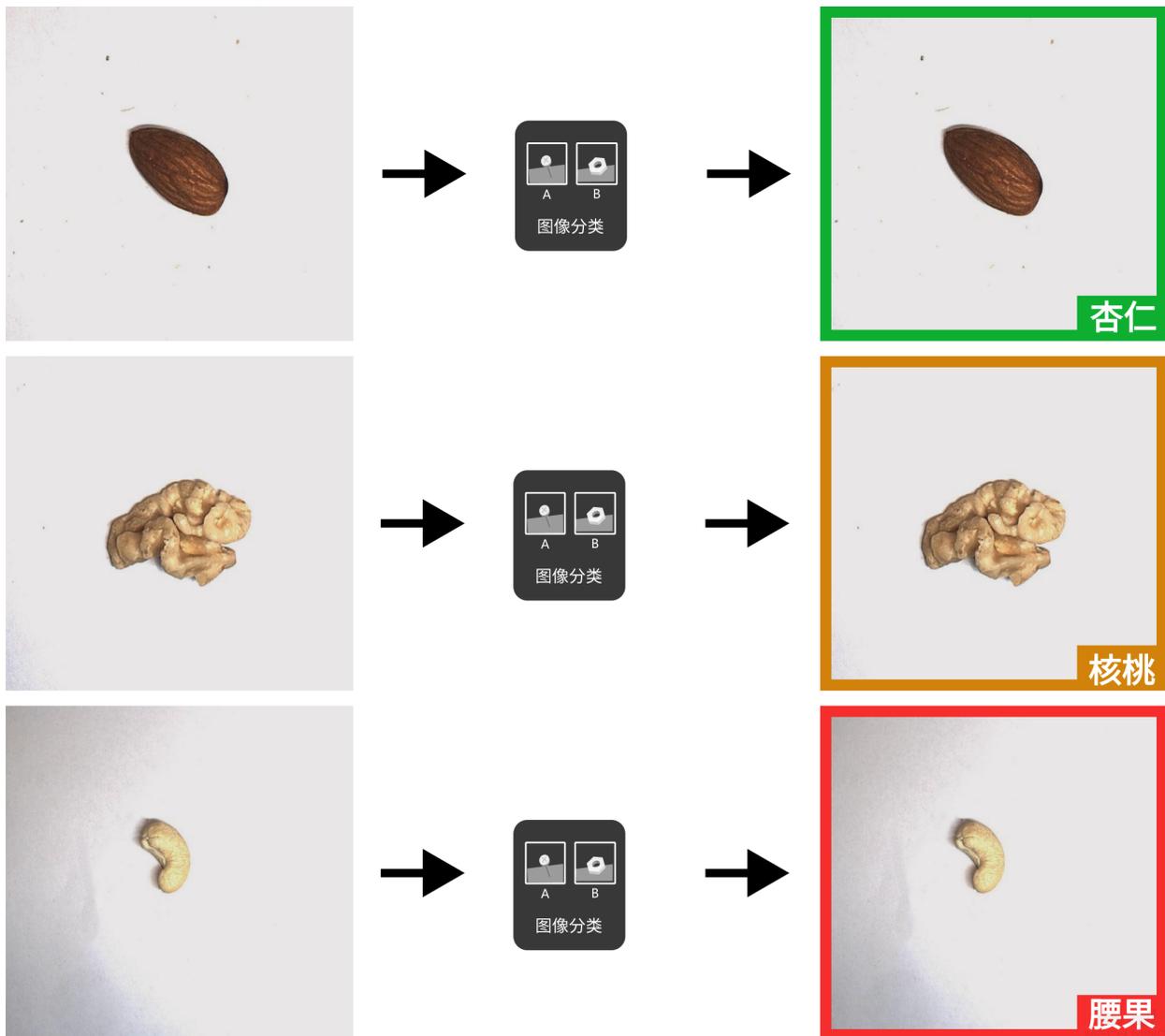
#### 图像分类能做什么

图像分类要解决的的问题是“是什么”。

即判断一张图是什么种类、工件大小型号、正面 & 反面、摆放正确 & 错误等。

例如：

目标物体是杏仁、核桃和腰果，输入一张图像，判断图像是其中的哪一种物体，并给出对应的类别标签。

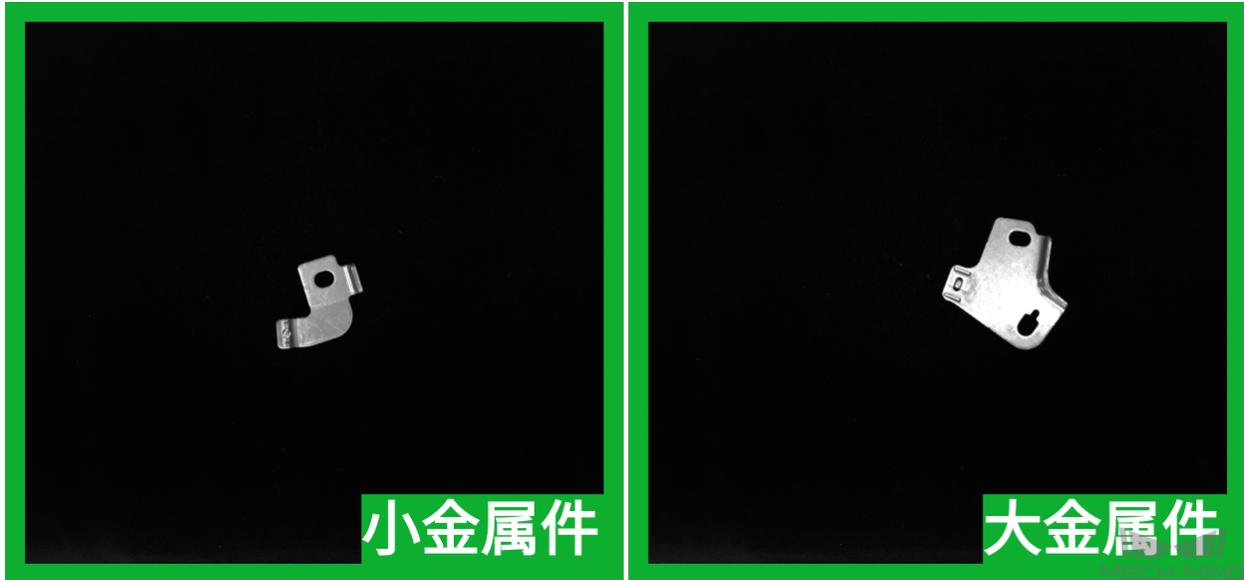


**注意：** 图像分类所给出的类别标签是针对整张图片的，如果图中有多个不同类别的物体，希望对每个物体进行分类，需要先将图片切割成小图（每张小图只包含一个物体），也可根据情况选择使用实例分割或目标检测。

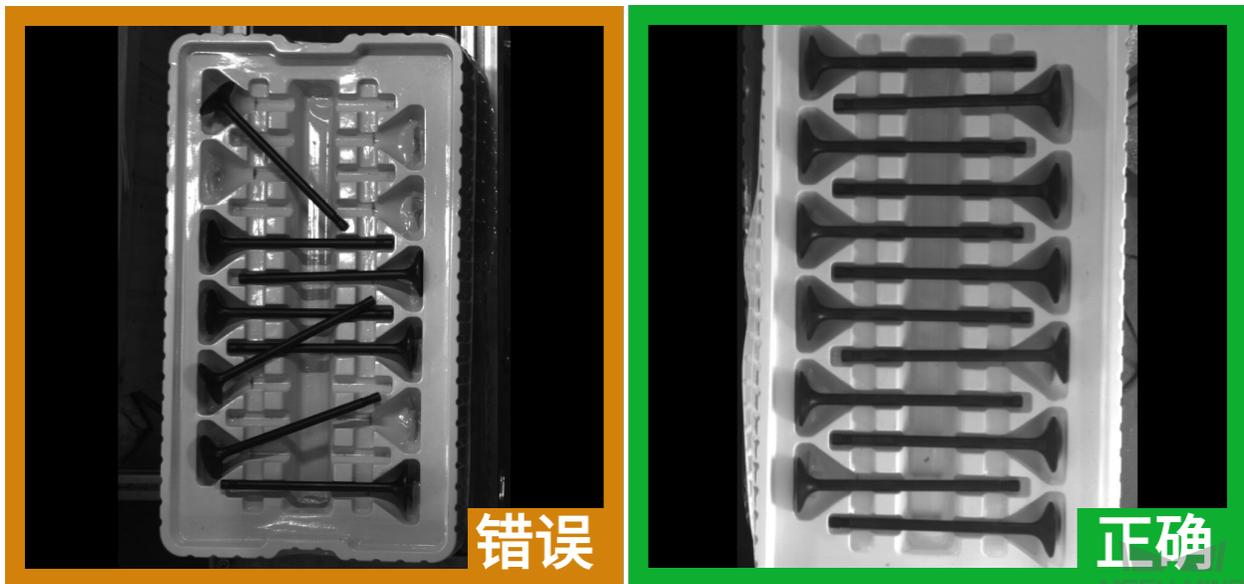
## 图像分类行业典型应用场景

项目需求若要对不同类别的图片进行区分，可以使用图像分类模型解决，下面列举了部分典型应用场景：

- 工件上下料项目中区分工件种类、朝向、正反面等。



- 装配/拣料项目中判断物体是否摆放正确等。



## 图像分类的应用流程

基于深度学习的图像分类，只需用户提供足够多的实际应用场景中物体图片供图像分类模型学习，并标注出待识别物体所属的标签类别，图像分类模型就能够自己学会区分。深度学习图像分类的应用流程如下：

- 采集训练所需数据：用相机拍摄足够多各类物体的图片。
- 标注数据：为每张采集的图片指定对应的类别标签。
- 训练模型：把前两个步骤得到的标注后的数据输入到图像分类模型学习。
- 使用模型进行预测：将训练好的模型用到项目中，发挥其功能。

### 2.2.2 采集训练所需数据

#### 务必确保采集环境符合要求

参阅实例分割：务必确保采集环境符合要求

**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。

#### 采集多少张数据

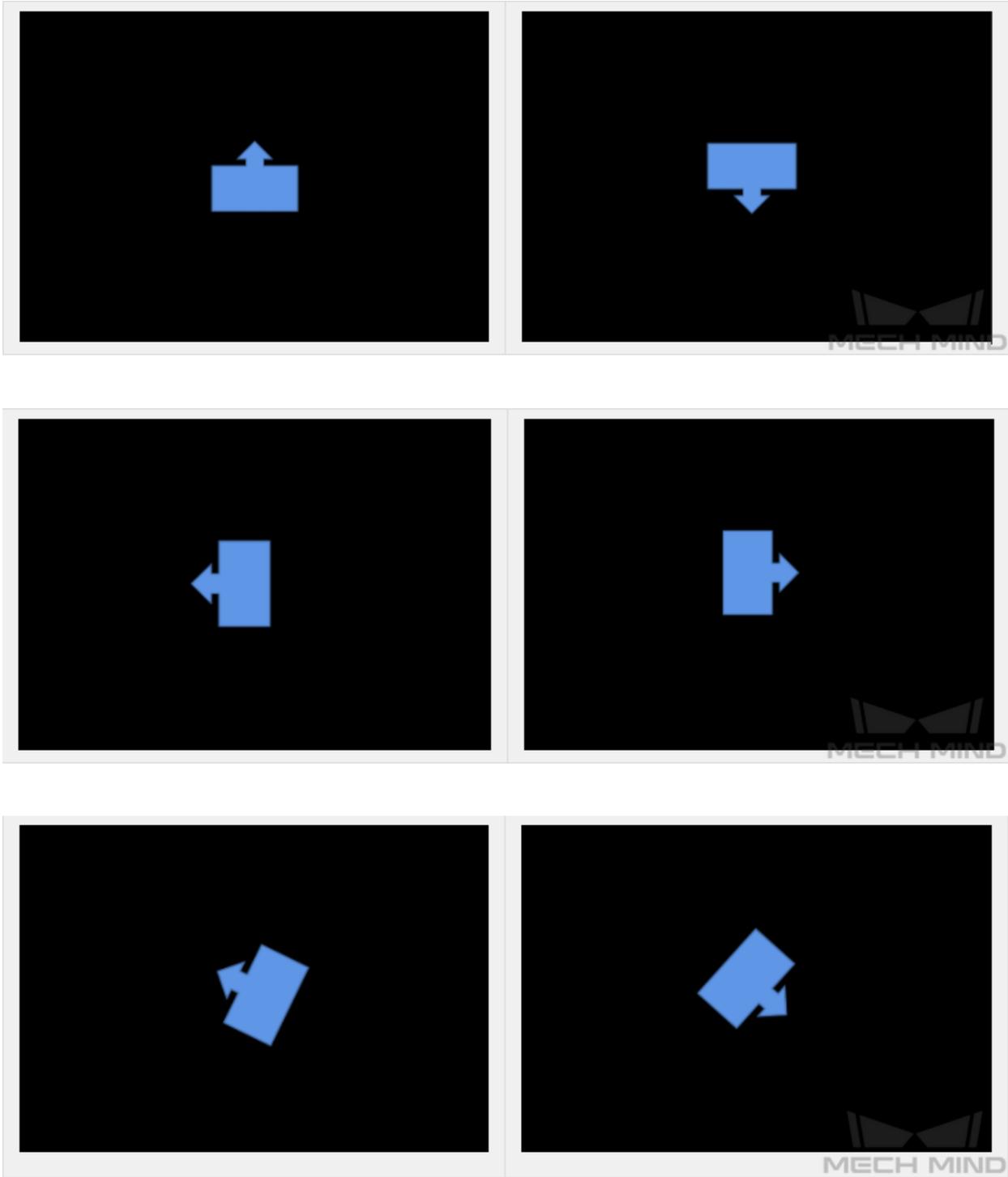
- 每类物体采集约 20 张。

#### 采集时的物体摆放

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和竖向，但只采集了横向来料的数据进行训练，那么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：

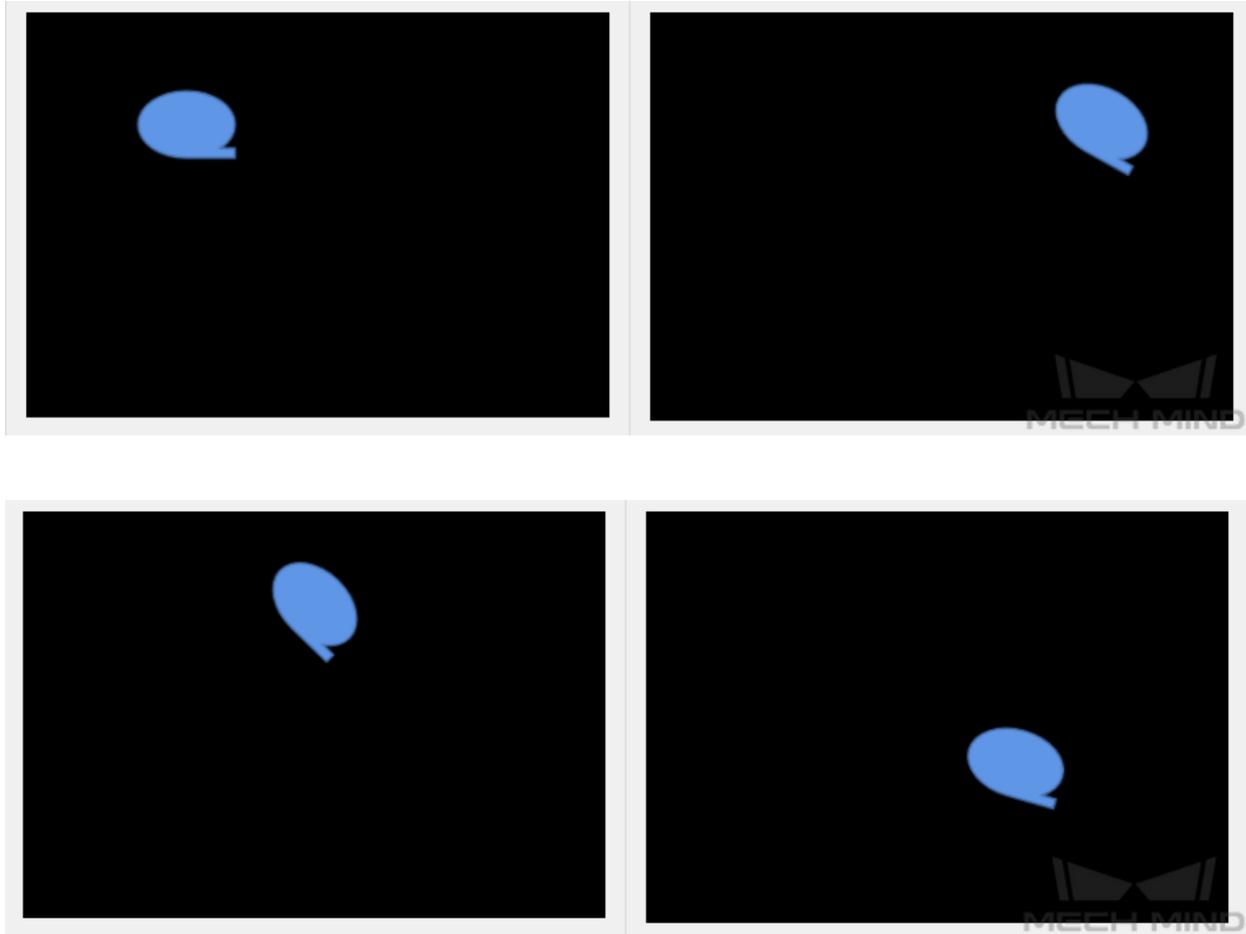
- 确保采集的数据中包含实际应用可能出现的待分类物体 **不同角度**的特征。
- 确保采集的数据中包含实际应用可能出现的待分类物体 **不同位置**的特征。

##### 1. 各个角度



不同角度

2. 不同位置



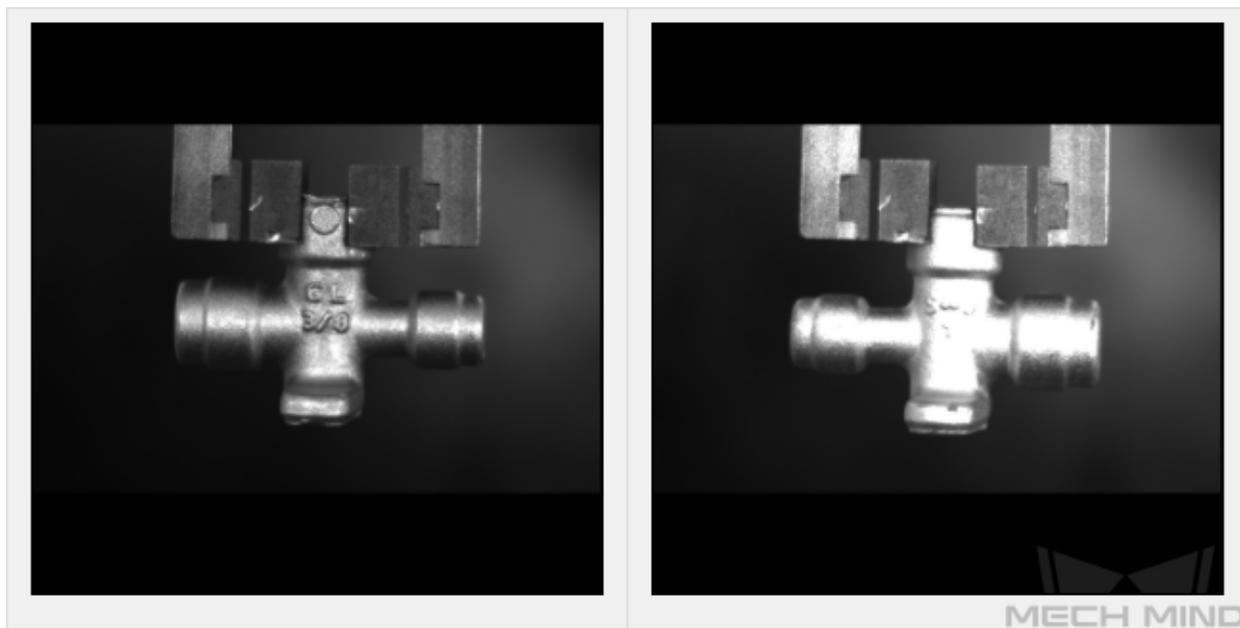
不同位置

### 使用 Mech-Vision 采集数据

上述采集环境、采集数量、采集时摆放方式均确认后，可参阅使用 *Mech-Vision* 采集数据。

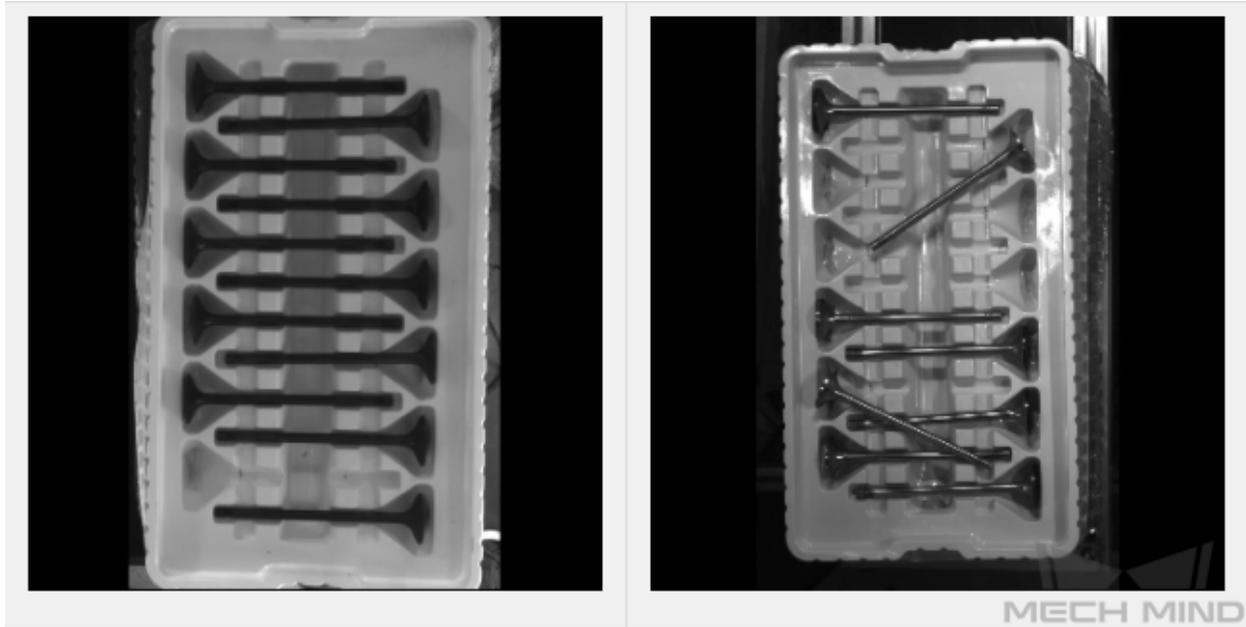
## 历史项目数据采集案例

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 10 张。



正面 & 反面

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，需考虑不同位置和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。



卡槽内 & 卡槽外

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





正面 &amp; 反面

## 2.2.3 标注数据

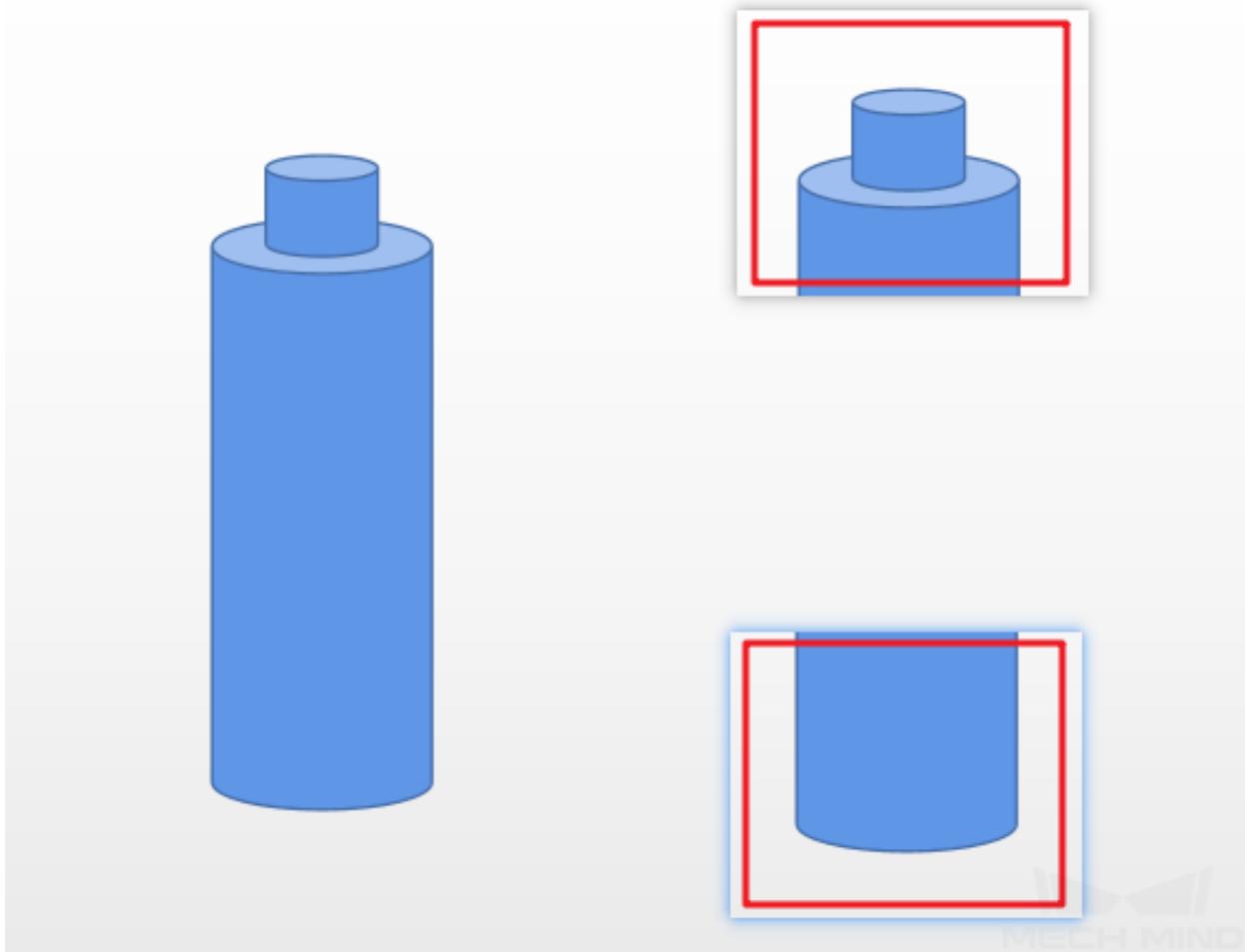
### 创建标签

依据项目需求创建对应的标签，例如项目需求是区分工件正反面，则创建标签 front 和 back。

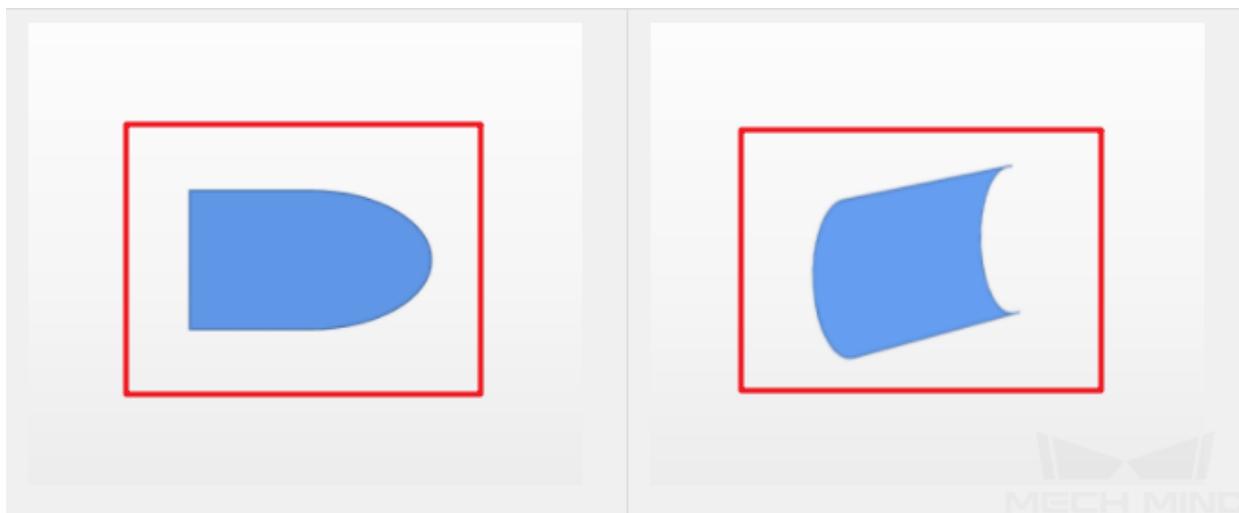
**注意：**标签名称需有辨识度，避免使用 a、b、tmp 等无意义名称；标签名称应仅包含英文字母与数字；应尽量使用英文单词（如 box）而非汉语拼音（如 xiangzi）以避免歧义。

### 确认标注方式

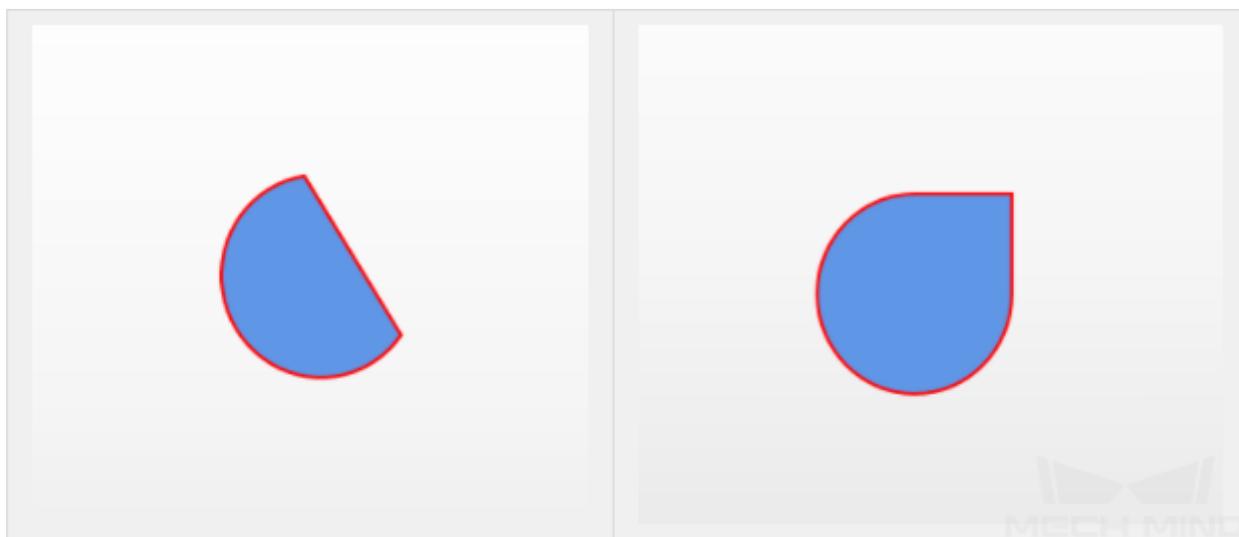
1. 若在同一物体上区分类别，需分别用矩形框标注明显特征位置，如下图：



2. 若要区分多个物体类别，需用矩形框标注整个物体，如下图：



3. 若在实际应用中输入图像分类模型的图像为去过背景后的图像，则在标注训练数据集时，需要在原图中标注整个物体轮廓，如下图：



**注意：**务必确保标注质量：图像分类的目的就是区分类别，需确保图像与标签正确对应。如果 10 张标签为“front”的数据中有一张被误标为“back”，会严重影响模型分类效果。

## 2.2.4 训练模型

### 使用 Mech-DLK 训练模型

参考快速入门。

## 2.2.5 使用模型进行预测

### 在 Mech-Vision 中使用图像分类模型

**注意：**模型文件为.pth 格式，配置文件为.json 格式。（图像分类输入的数据必须与训练数据保持一致）。



---

## 将深度学习用于典型项目

---

### 3.1 纸箱拆/码垛项目应用

- 纸箱拆/码垛项目中，通常需要使用实例分割来分割出图中的每个纸箱并给出位置信息。
- 本公司专门针对纸箱拆/码垛场景提供了**超级模型**，可以直接应用到 Mech-Vision 中，在无需训练的情况下正确分割绝大多数纸箱。

整体流程如下：

1. 使用超级模型进行预测，确认项目中用到的所有纸箱是否都能被正确分割，若能则直接使用超级模型即可，跳过后面所有步骤。
2. 若不能，采集所有无法被超级模型正确分割的纸箱数据用于训练。
3. 对数据进行去背景操作。
4. 标注数据。
5. 训练模型。
6. 使用新模型进行预测。
7. 对于纸箱种类较多的情况，项目初期无法拿到所有类型的纸箱来测试。可以先用模型测试现有种类的纸箱数据，若后续有新类型纸箱加入且出现分割错误的情况，则需要重复步骤 2~6 再次进行模型更新。

#### 1. 使用超级模型并确认是否满足需求

使用超级模型对项目数据进行预测，参见[实例分割：使用模型进行预测](#)

确认项目中用到的所有纸箱是否都能被正确分割，若能则直接使用超级模型即可，跳过后面所有步骤，若不能继续下面步骤。

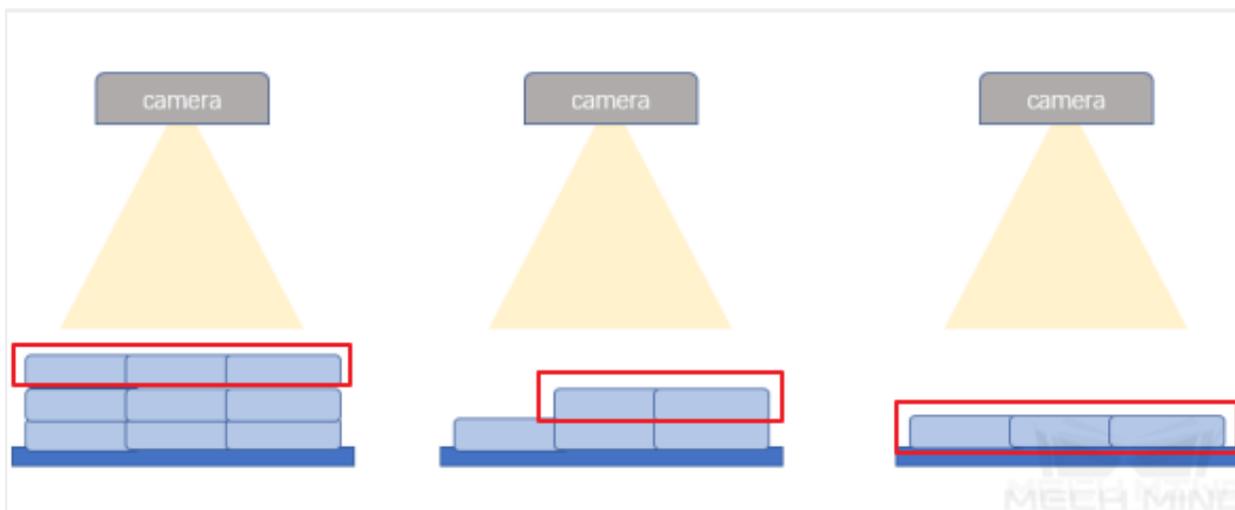
**注意：**无论超级模型效果如何，请务必保留所有测试数据，便于后期可能需要的测试。

## 2. 采集超级模型分割错误的纸箱数据用于训练

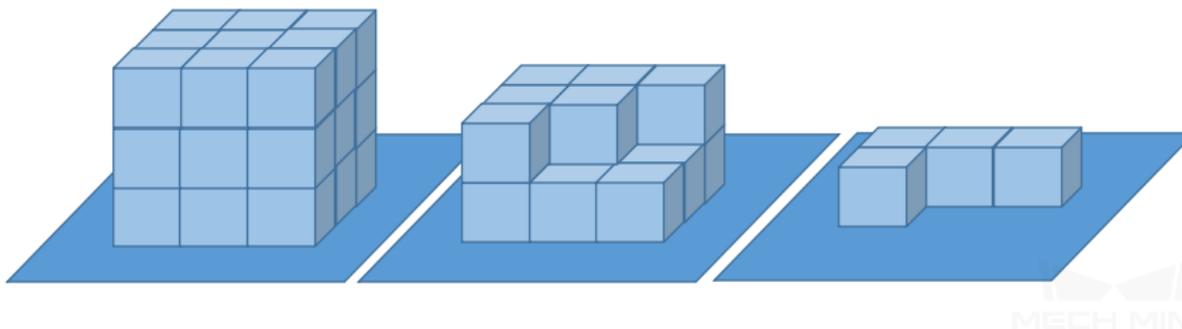
一般情况下，超级模型可以识别大部分纸箱。极端情况下，如纸箱紧密贴合摆放或表面花纹复杂时，可能出现分割错误或掩膜不完整的情况；用户只需针对识别有问题的纸箱采集数据迭代模型即可。

例如，20种纸箱中有18种可以正确分割，有2种不能，则只需要采集这2种的数据。又如，单独摆放的纸箱都可以被正确分割，但紧密贴合摆放的纸箱不能，则只需要采集紧密贴合摆放的数据。

- 采集数量：每种纸箱（或摆放方式）采集20张。
- 数据要求：①在不同垛型高度（高层、中间层、底层）采集紧密贴合的纸箱图像10张；②在不同垛型高度（高层、中间层、底层）采集纸箱满垛、半垛、少量的情况图像10张。



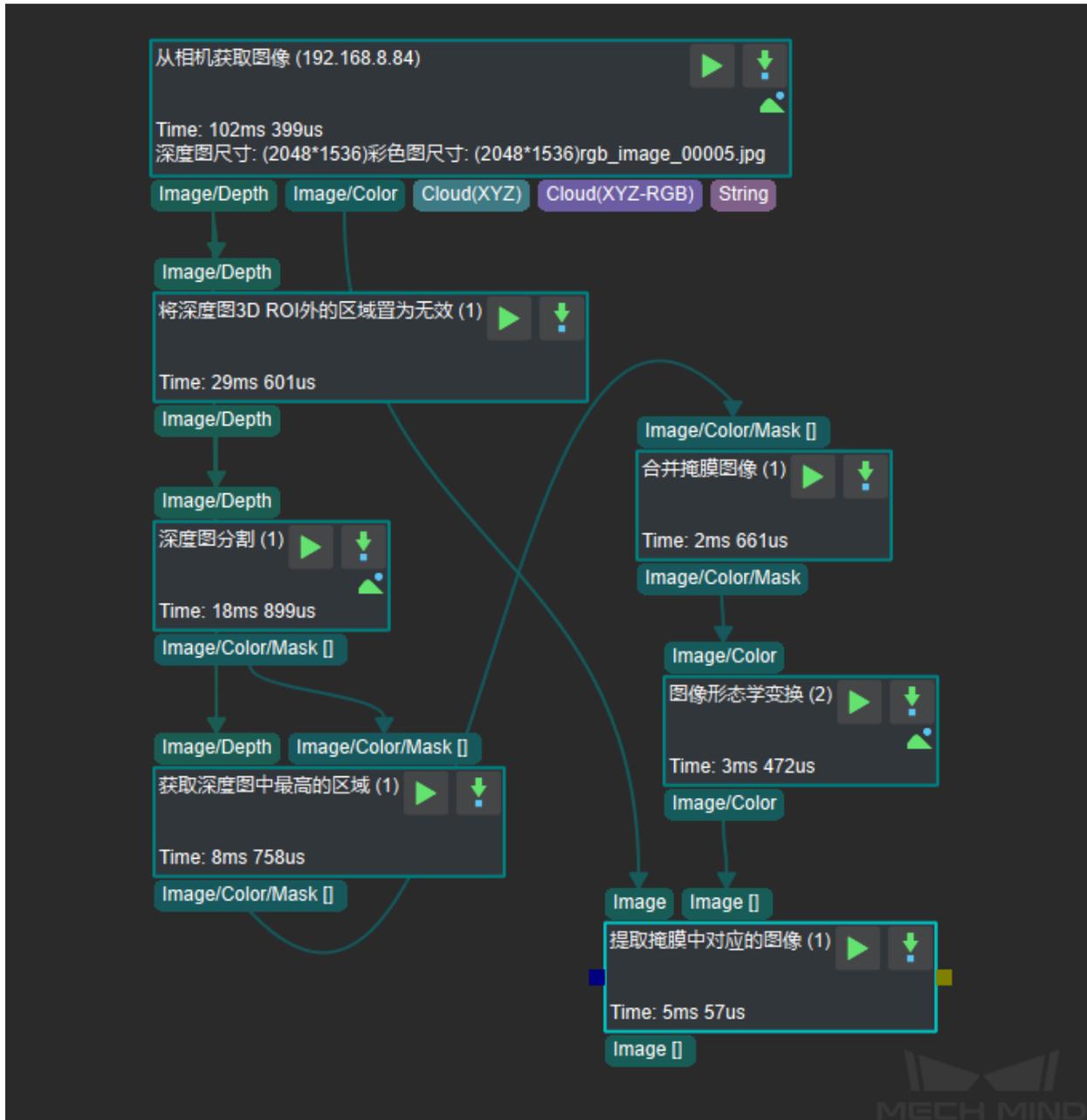
高层（图左）& 中间层（图中）& 底层（图右）



最高层满垛（图左）& 中间层半垛（图中）& 最底层少量（图右）

## 3. 对数据进行去背景操作

由于纸箱拆/码垛场景中纸箱都以成垛形式出现，进行去背景操作可以避免背景干扰，显著提高模型效果。在 Mech-Vision 中，通过去背景组合步骤处理得到去背景的数据。



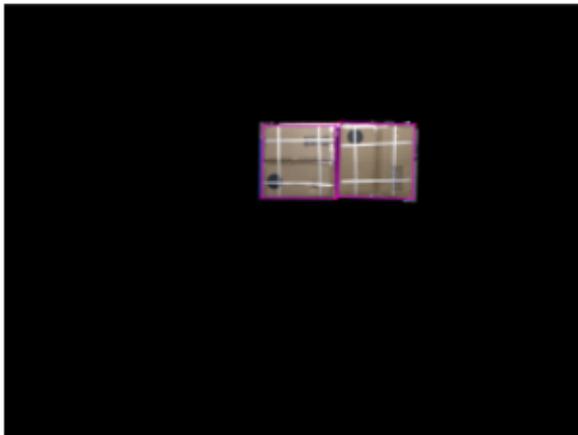


原图 &amp; 去背景图

#### 4. 标注数据

参见实例分割：标注数据

纸箱拆/码垛场景只标注纸箱的上表面不标注侧面，且只标注 **完整** 的上表面，被遮挡则不标注。



标注上表面轮廓

#### 5. 训练模型

参见实例分割：训练模型

训练参数中，总训练轮次（epoch）设置为 200，其余保持默认值。

## 6. 使用新模型进行预测

参见实例分割：使用模型进行预测

## 7. 视情况重复步骤 2-6 再次进行模型更新

对于纸箱种类较多的情况，项目初期无法拿到所有类型的纸箱来测试。可以先用模型测试现有种类的纸箱数据，若后续有新类型纸箱加入且出现分割错误的情况，则需要重复步骤 2-6 再次进行模型更新。

---

**注解：** 超级模型指基于海量数据训练的适用于某类物体（如纸箱、麻袋、快递包裹等）的通用深度学习模型。

---

## 3.2 麻袋拆/码垛项目应用

- 麻袋拆/码垛项目中，通常需要使用实例分割 来分割图中的每个麻袋。
- 本公司专门针对麻袋拆/码垛场景提供了 **超级模型**，可以在不经过训练的情况下正确分割绝大多数麻袋。

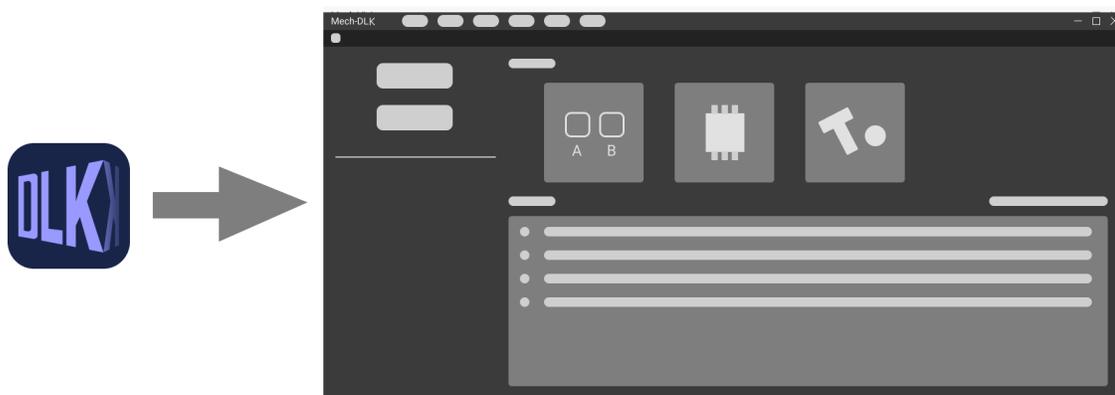
整体流程如下：

1. 使用超级模型进行预测，确认项目中用到的所有麻袋是否都能被正确分割，若能则直接使用超级模型即可，跳过后面所有步骤。
2. 若不能，采集所有无法被超级模型正确分割的麻袋数据用于训练。
3. 标注数据。
4. 训练模型。
5. 使用新模型进行预测。
6. 对于麻袋种类较多的情况，项目初期无法拿到所有类型的麻袋来测试。可以先用模型测试现有种类的麻袋数据，若后续有新类型麻袋加入且出现分割错误的情况，则需要重复步骤 2~6 再次进行模型更新。

### 4.1 软件概述

#### 4.1.1 功能简介

Mech-DLK 是梅卡曼德自主研发的一款平台化深度学习软件，内置多种业界领先的 AI 算法，借助直观简易的操作解决复杂的问题，如堆叠物体识别、高难度缺陷检测、产品等级分类等。它可以提升生产效率、产品良率、降低产线用工成本，适用于消费电子、新能源、汽车、家电、物流等行业。



### 4.1.2 模块功能简介

**图像分类：**用于对多种物体进行分类，只需提供少量多类别物体图像，便可通过学习区分不同种类的物体。

**目标检测：**用于在图像中快速定位目标，只需提供少量图像，便可通过学习定位目标物体的位置。

**语义分割：**用于识别图像中有缺陷的位置，只需提供一定量的 OK 样本和 NG 样本，便可通过学习识别出图像中是否有缺陷。

**实例分割：**用于定位图像中目标物体的位置并对其进行分类，只需提供少量多类别物体图像，便可通过学习定位到物体的具体位置并给出类别标签。

**注意：**上述几种模块均可独立训练出模型满足用户需求。如有多种功能综合的定制化需求，也可在 Mech-DLK 中通过串接不同模块组合训练来满足。

### 4.1.3 支持二次开发

二次开发配置要求	
操作系统	Windows7 及以上
开发平台	VS2013、VS2015、VS2017、VS2019(推荐)
开发语言	C、C++、C#

## 4.2 环境配置

### 4.2.1 硬件要求

	Mech-DLK Pro-Run	Mech-DLK Pro-Train/Standard
操作系统	Windows10	
CPU	i5 及以上	i7 及以上
内存	8GB 及以上	16GB 及以上
显卡	GeForce GTX 1650(4GB) 及以上	GeForce GTX 2070 (8GB) 及以上

显卡型号	台式机	笔记本
10 系列	NVIDIA GeForce GT 1030 NVIDIA GeForce GTX 1050 NVIDIA GeForce GTX 1050Ti NVIDIA GeForce GTX 1060 NVIDIA GeForce GTX 1070 NVIDIA GeForce GTX 1070Ti NVIDIA GeForce GTX 1080 NVIDIA GeForce GTX 1080Ti NVIDIA GeForce GTX 1650 NVIDIA GeForce GTX 1650 SUPER NVIDIA GeForce GTX 1660 NVIDIA GeForce GTX 1660Ti NVIDIA GeForce GTX 1660 SUPER	NVIDIA GeForce GTX 1050 NVIDIA GeForce GTX 1050Ti NVIDIA GeForce GTX 1060 NVIDIA GeForce GTX 1070 NVIDIA GeForce GTX 1080 NVIDIA GeForce GTX 1650 NVIDIA GeForce GTX 1650Ti NVIDIA GeForce GTX 1660Ti
20 系列	NVIDIA GeForce RTX 2060 NVIDIA GeForce RTX 2060 SUPER NVIDIA GeForce RTX 2070 NVIDIA GeForce RTX 2070 SUPER NVIDIA GeForce RTX 2080 NVIDIA GeForce RTX 2080Ti NVIDIA GeForce RTX 2080 SUPER	NVIDIA GeForce RTX 2060 NVIDIA GeForce RTX 2070 NVIDIA GeForce RTX 2080
30 系列	NVIDIA GeForce RTX 3050 NVIDIA GeForce RTX 3060 NVIDIA GeForce RTX 3060Ti NVIDIA GeForce RTX 3070 NVIDIA GeForce RTX 3070Ti NVIDIA GeForce RTX 3080 NVIDIA GeForce RTX 3080Ti NVIDIA GeForce RTX 3090	NVIDIA GeForce RTX 3050 Laptop GPU NVIDIA GeForce RTX 3060 Laptop GPU NVIDIA GeForce RTX 3070 Laptop GPU NVIDIA GeForce RTX 3080 Laptop GPU

## 4.2.2 环境配置

### 1. 直接安装 V2.1.0 环境

通常的选择，适用于新的电脑，此前从未安装过任何版本的深度学习环境。

- 将 Mech-Mind\_software\_environment\_installer(2.1.0).zip.001、Mech-Mind\_software\_environment\_installer(2.1.0).zip.002 和 Mech-Mind\_software\_environment\_installer(2.1.0).zip.003 下载至同一路径。
- 因为文件较大，请校验文件以确保下载正确，校验方式是右键点击文件，选择 CRC SHA -> CRC-32 查看校验码。上述三个文件的标准校验码分别是：

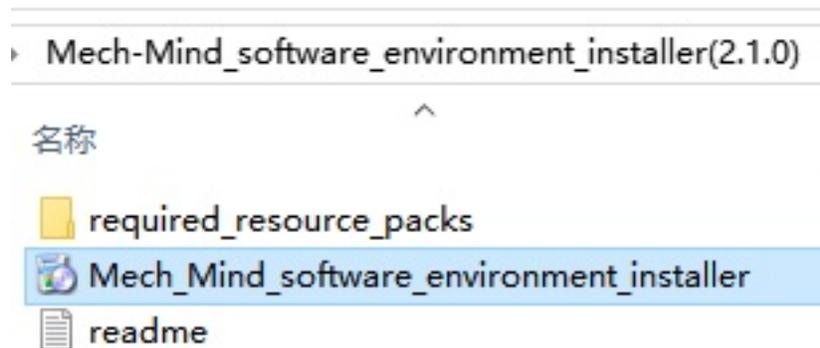
Name Mech-Mind\_software\_environment\_installer(2.1.0).zip.001  
Size 4290772992 Bytes (4092 MiB)  
CRC32 903E692A

Name Mech-Mind\_software\_environment\_installer(2.1.0).zip.002  
Size 4290772992 Bytes (4092 MiB)  
CRC32 C95B7D3F

Name Mech-Mind\_software\_environment\_installer(2.1.0).zip.003  
Size 918773062 Bytes (876 MiB)  
CRC32 C36F7F64



- 使用解压缩软件对上一步骤中的压缩包进行解压。
- 在解压后的文件中找到 Mech\_Mind\_software\_environment\_installer.exe，双击该程序并按照提示进行安装。



## 2. 从 V2.0.0 环境升级至 V2.1.0 环境

适用于当前环境版本为 V2.0.0 的电脑,即安装过 Mech-Mind\_software\_environment\_installer(V2.0.0) 或 Mech-Mind\_software\_environment\_installer(V2.0.0\_Lite) 环境安装包。

- 下载 Mech-Mind\_software\_environment\_installer(2.1.0\_upgrade) 文件夹。
- 下载完成后在文件中找到 install\_pips.bat，双击该脚本完成环境更新。

Mech-Mind_software_environment_installer(2.1.0_upgrade)	
名称	修改日期
 install_pips	2022/5/6 18:02
 mmind_mmcv_shell-1.1.0-py3-none-a...	2022/5/6 18:02
 mmind_mmdet_shell-1.1.0-py3-none-...	2022/5/6 18:02
 mmind_mmseg_shell-1.1.0-py3-none-...	2022/5/6 18:02
 Readme	2022/5/6 18:02

### 3. 从 V1.4.0 或更早的环境升级至 V2.1.0 环境

适用于当前环境版本为 **V1.4.0** 或更早的电脑，即安装过 **Mech-Mind\_software\_environment\_installer(V1.4.0)** 或更早的环境安装包，但并未安装过 **V2.0.0** 环境安装包。

- 若需要使用 Mech-DLK 训练模型请直接参照上文“1、直接安装 V2.1.0 环境”。
- 若不需要使用 Mech-DLK 训练模型
  - 下载 Mech-Mind\_software\_environment\_installer(2.1.0\_lite) 文件夹。
  - 下载完成后在文件中找到 Mech\_Mind\_software\_environment\_installer.exe，双击该程序并按照提示进行安装。

Mech-Mind_software_environment_installer(2.1.0_lite)	
名称	
 required_resource_packs	
 Mech_Mind_software_environment_installer	
 readme	

### 4. 安装指引

当前环境版本	推荐操作		需要下载的文件	下载后的操作	备注
未安装任何版本	安装完整的V2.1.0环境		Mech-Mind_software_environment_installer(2.1.0).zip.001 Mech-Mind_software_environment_installer(2.1.0).zip.002 Mech-Mind_software_environment_installer(2.1.0).zip.003	<ul style="list-style-type: none"> <li>使用压缩软件对上述文件进行解压</li> <li>运行解压后得到的exe格式安装程序</li> </ul>	<ul style="list-style-type: none"> <li>下载的三个压缩文件需放在同一路径</li> <li>下载后需要对文件进行校验</li> </ul>
V2.0.0	安装V2.1.0_Upgrade		Mech-Mind_software_environment_installer(2.1.0_upgrade)	<ul style="list-style-type: none"> <li>运行install_pips.bat</li> </ul>	
V1.4.0及以下	使用Mech-DLK训练模型	安装完整的V2.1.0环境	Mech-Mind_software_environment_installer(2.1.0).zip.001 Mech-Mind_software_environment_installer(2.1.0).zip.002 Mech-Mind_software_environment_installer(2.1.0).zip.003	<ul style="list-style-type: none"> <li>使用压缩软件对上述文件进行解压</li> <li>运行解压后得到的exe格式安装程序</li> </ul>	<ul style="list-style-type: none"> <li>下载的三个压缩文件需放在同一路径</li> <li>下载后需要对文件进行校验</li> </ul>
	不使用Mech-DLK训练模型	安装V2.1.0_Lite环境	Mech-Mind_software_environment_installer(2.1.0_lite)	<ul style="list-style-type: none"> <li>运行文件夹中的exe格式安装程序</li> </ul>	

如何查看当前环境的版本：

打开控制面板，在程序和功能中可查询到当前环境的版本号。



名称	发布者	安装日期	大小	版本
WPS Office (11.1.0.9192)	Kingsoft Corp.	2021/2/22	590 MB	11.1.0.9192
ChLake	Kitware	2022/3/19	107 MB	3.22.3
Netron 5.6.6	Lutz Roeder	2022/3/26	420 MB	5.6.6
Mech-Vision	Mech-Mind	2022/4/12	1.71 GB	1.5.1
Mech-Viz	Mech-Mind	2022/4/8	3.37 GB	1.5.1-dev-04-07
Mech-Center	Mech-Mind	2022/4/6	319 MB	1.5.1
Mech-Eye SDK	Mech-Mind	2022/5/5	824 MB	1.5.2
Mech-Viz	Mech-Mind	2022/4/12	3.37 GB	1.5.1-dev-04-11
Mech-DLK	Mech-Mind Robotics Technologies Ltd	2022/4/28	2.07 GB	2.0.0
Mech-Mind Software Environment	Mech-Mind Robotics Technologies Ltd	2022/4/28	2.00	2.0.0
Microsoft Visual Studio Installer	Microsoft Corporation	2021/6/29	5.12 GB	8931
Microsoft OneDrive	Microsoft Corporation	2022/5/5	238 MB	22.077.0410.0007
Microsoft Visual Studio Code (1.66.2)	Microsoft Corporation	2022/4/13	292 MB	1.66.2

## 4.3 快速入门

- 本章节将带领用户快速训练并导出一个模型，该模型可识别出物体缺陷，并且对缺陷种类进行分类。
- 通过 Mech-DLK 软件，用户可以用语义分割模块级联图像分类模块的方式实现该功能。
- 使用 Mech-DLK 前，需要先确认深度学习环境和 Mech-DLK 是否安装成功。

### 1. 新建工程

点击初始页面上的新建工程以新建一个工程，如图 1 所示。

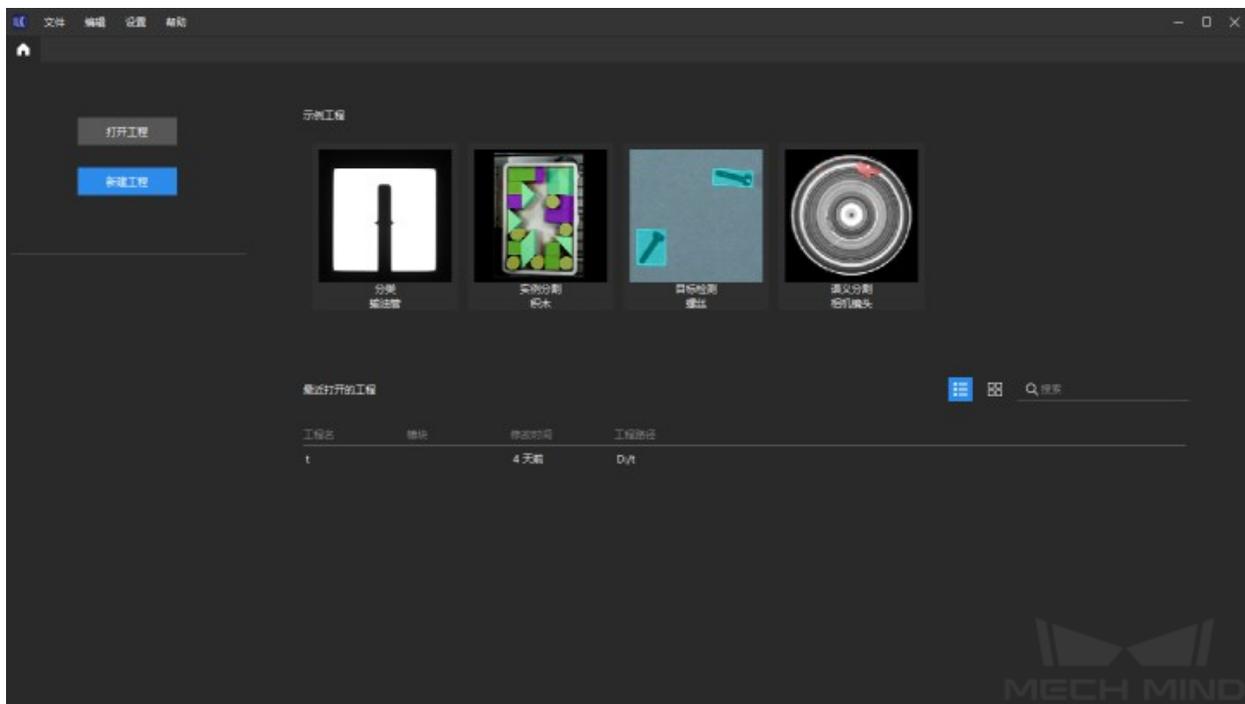


图 1 新建工程

## 2. 选择模块

添加模块：点击页面右侧模块栏中的  选择语义分割算法，点击确定。

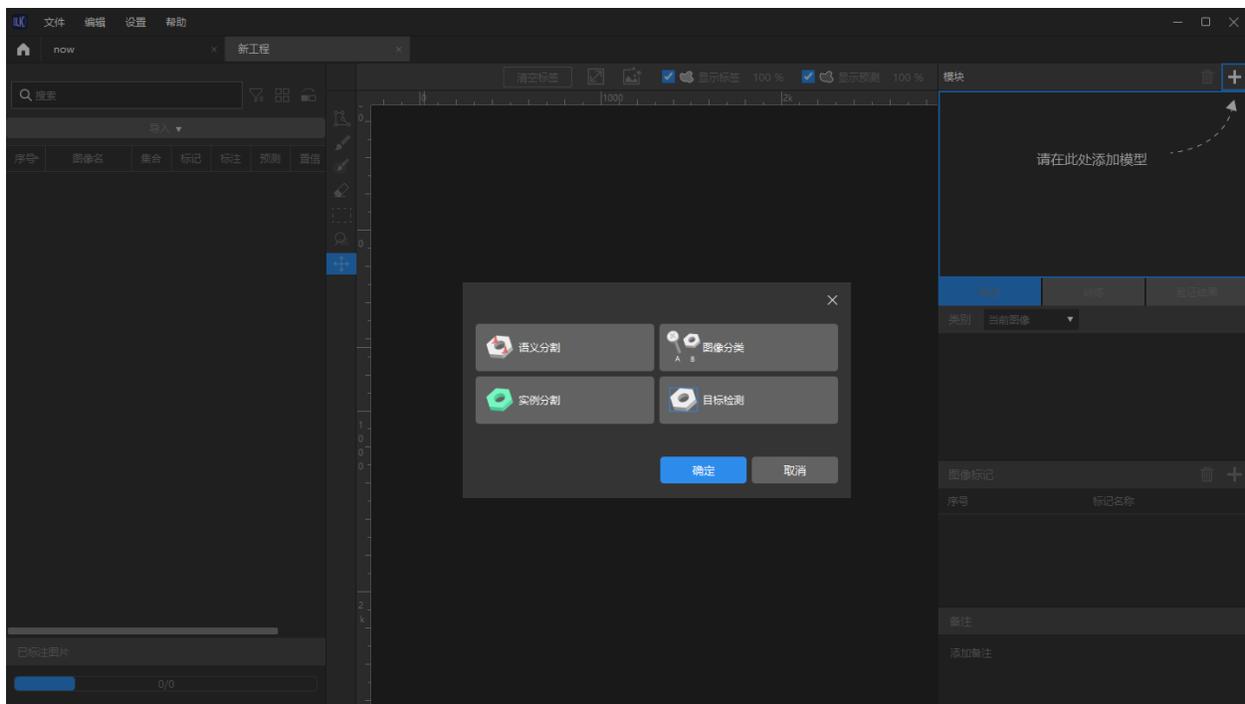


图 2 添加模块

### 3. 导入数据

点击左上侧的 **导入** 标签，选择导入本地的图像。

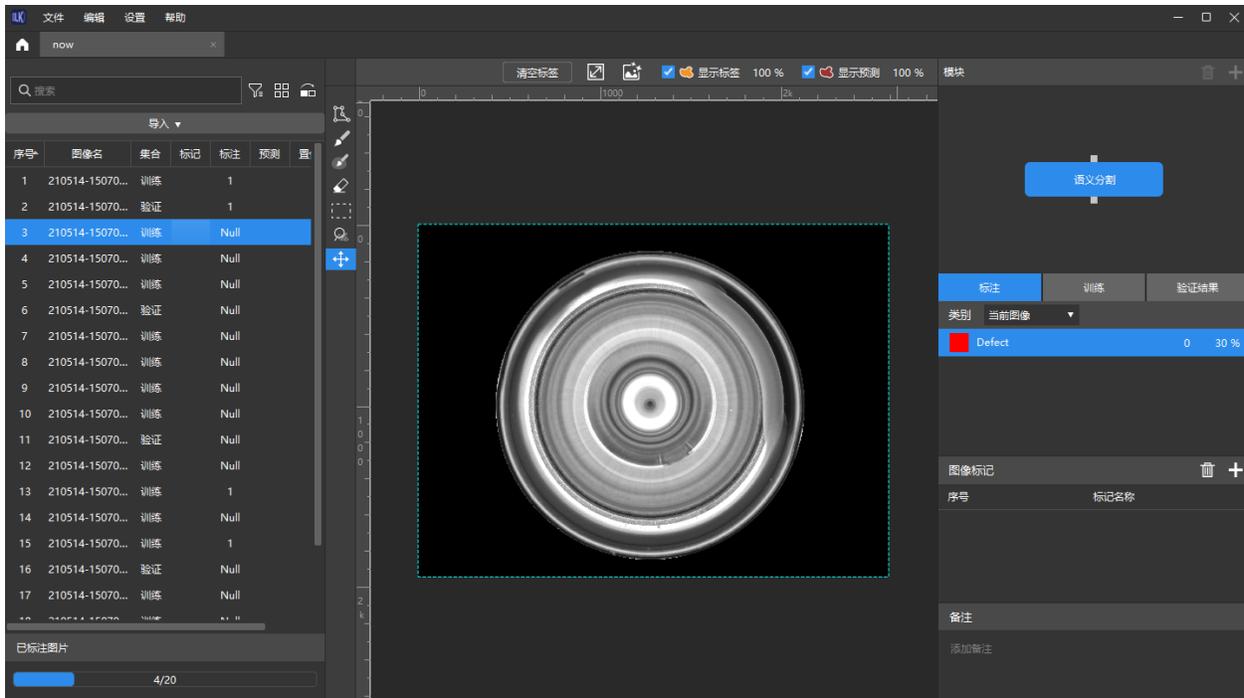


图 3 导入数据

**注意：**语义分割模块导入的图像中必须包含没有缺陷的图像，且验证集和训练集中都要有，否则会出现弹窗导致后续的训练无法正常进行。

#### 4. 标注数据

语义分割算法会自动生成一个名为 Defect 的标签，标注数据直接从左侧工具栏选取适当的标注工具标注即可。



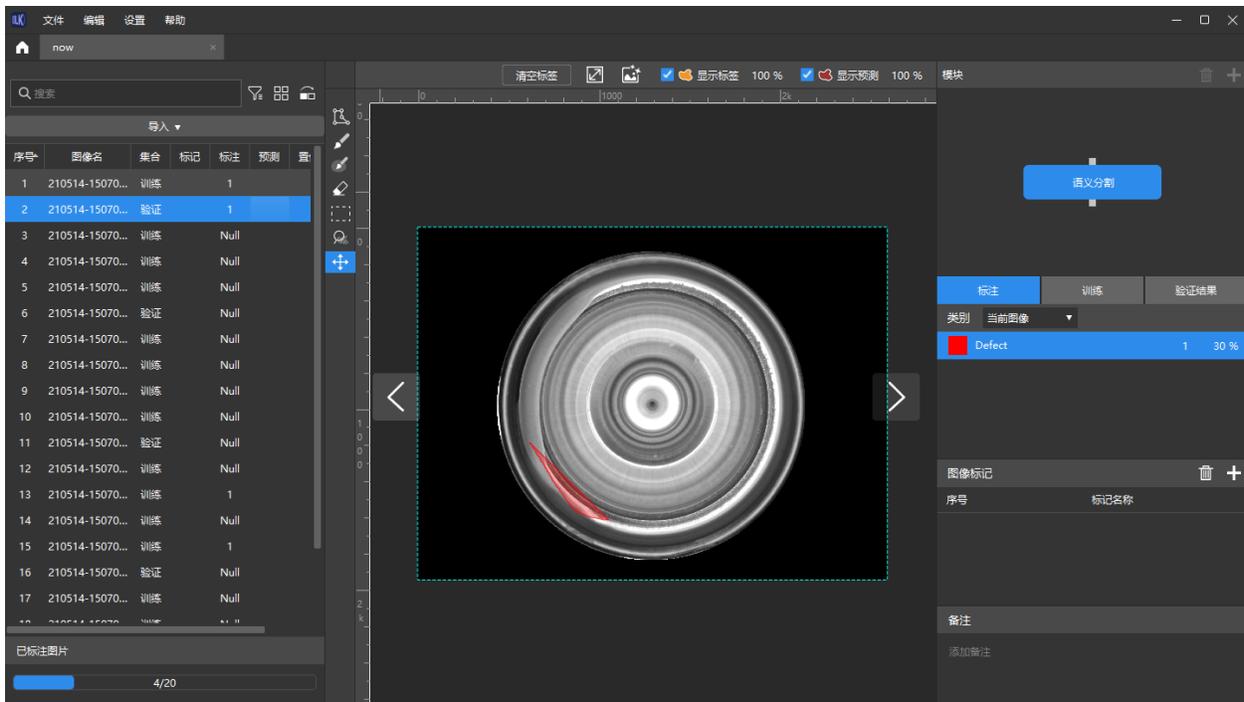


图 4 语义分割标注

## 5. 训练模型

点击右下方的 训练按钮开始训练即可。点击 显示图表可以查看训练的精度和损失。



图 5 训练图表

## 6. 验证模型

语义分割模型训练完后，点击 验证，进行结果验证，验证后会保存一个训练结果。

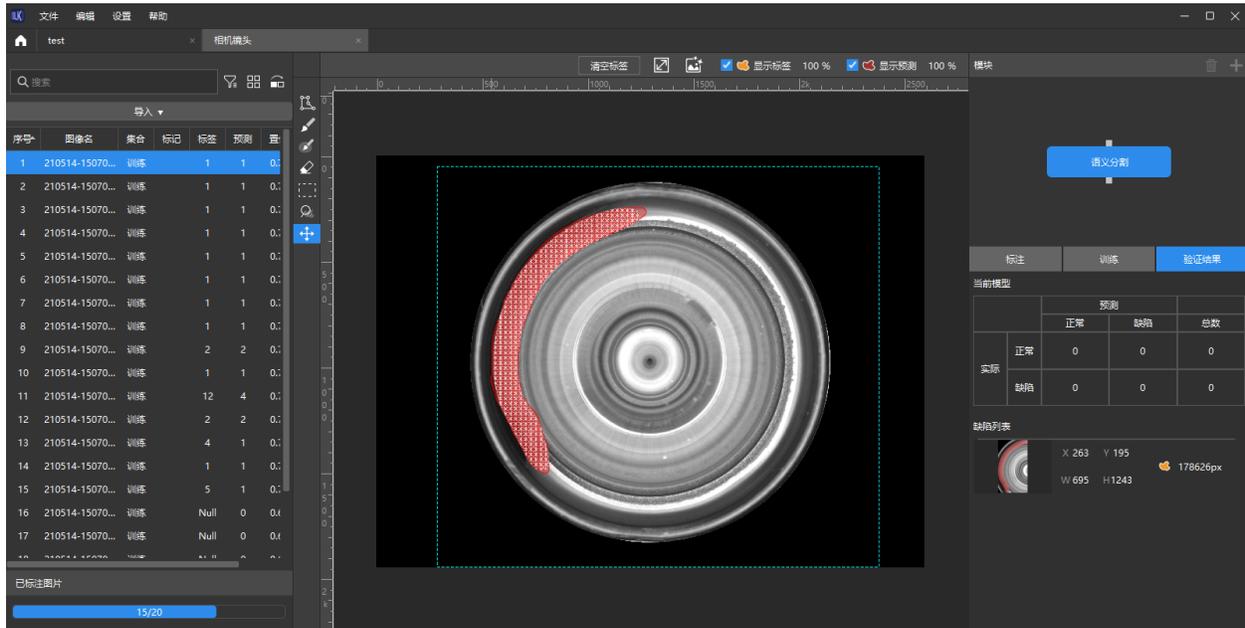


图 6 训练图表

## 7. 增加下一模块

模型效果确认良好后点击页面右上方模块栏中的  添加图像分类算法。

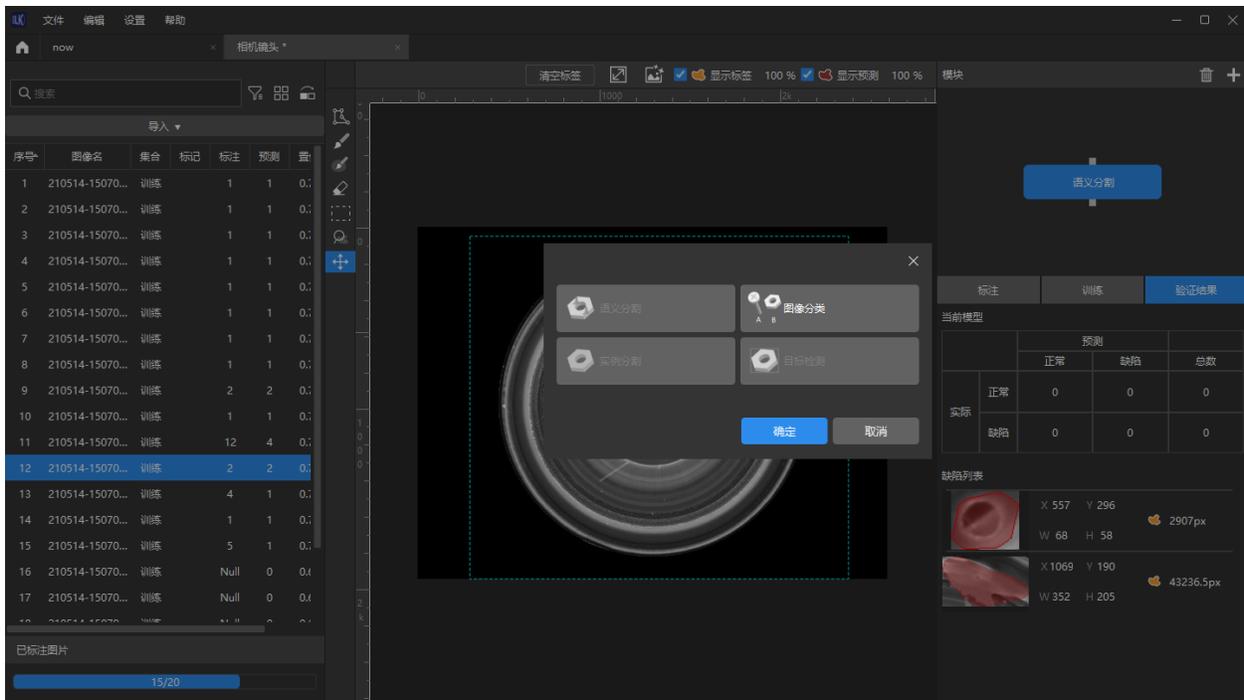


图 7 添加算法

#### 8. 将数据导入图像分类模块

上一模型训练结果会作为数据源导入图像分类算法，点击导入会直接弹框显示可选择数据，此时手动选择需要的数据即可。

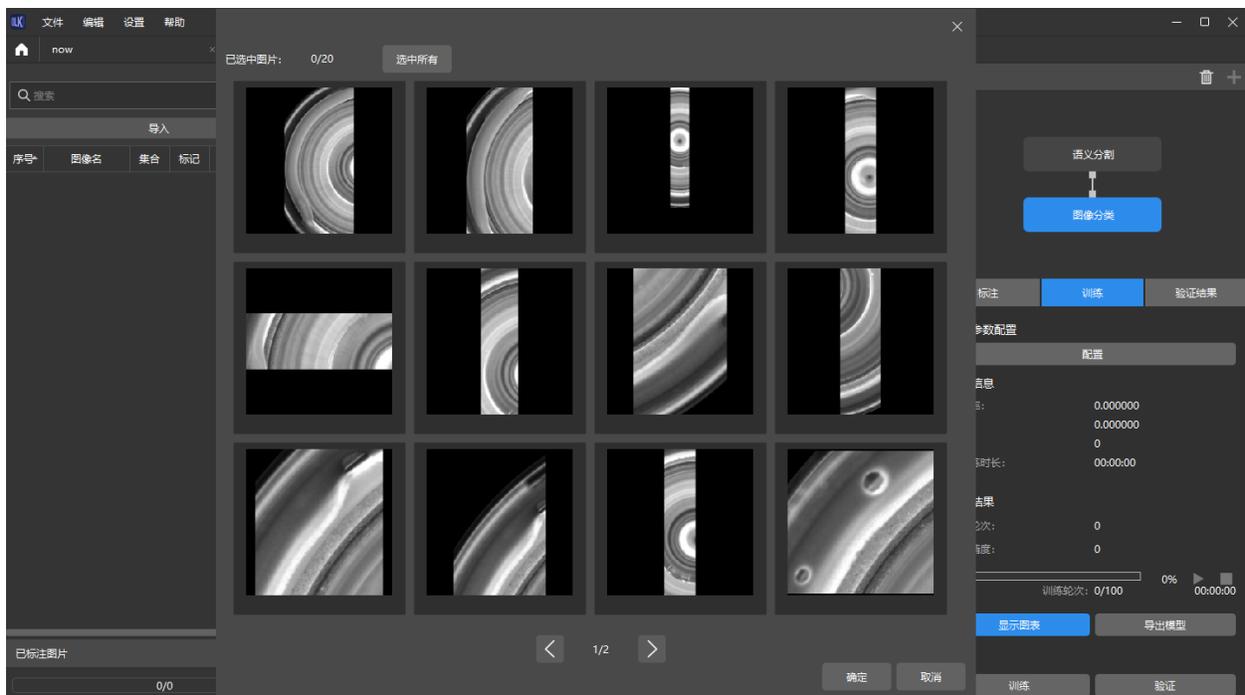


图 8 导入数据源

### 9. 建立图像分类标签并标注

图像分类在标注前需要先点击右侧标签栏的  创建针对不同种类物体的标签；之后点击标注栏左侧的  便可对图像进行标注。

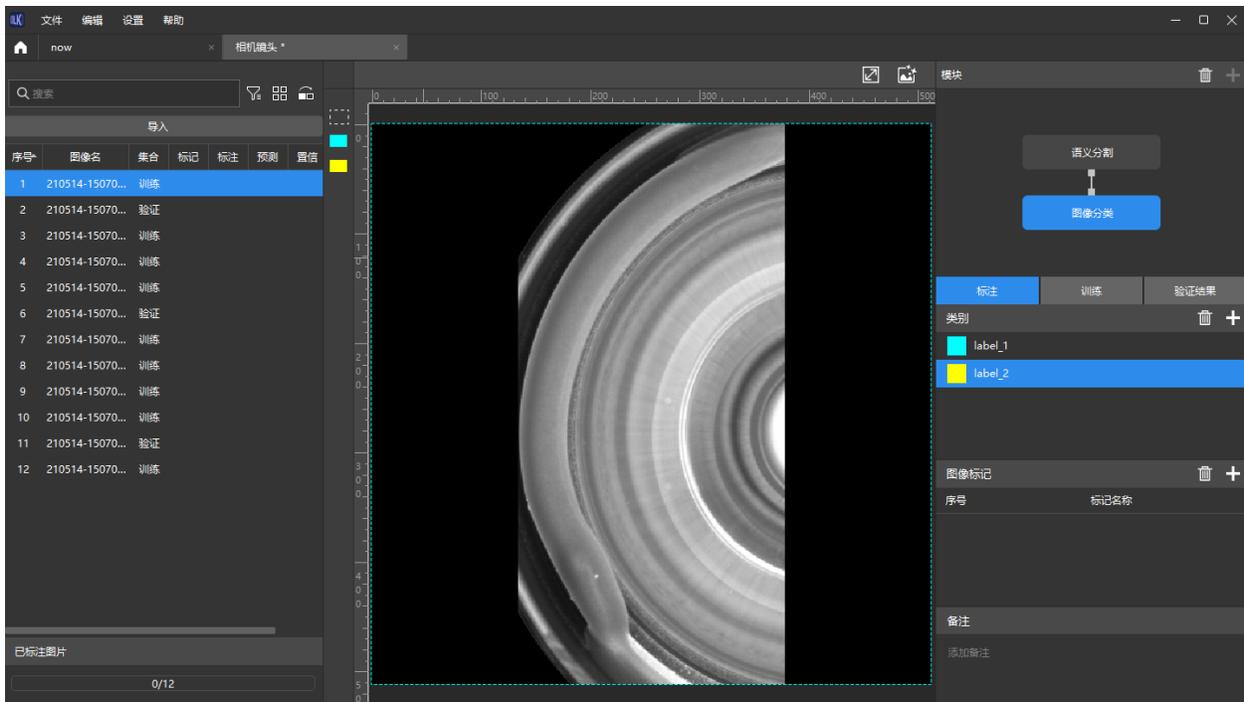


图 9 图像分类标注

#### 10. 训练模型

点击右下方的 **训练** 按钮开始训练即可。点击 **显示图表** 可查看训练的精度和损失。

#### 11. 导出最终模型

在模型训练完成后点击右侧 **导出模型**，选择保存路径后即可导出最佳模型到文件夹内。如图 10 所示。

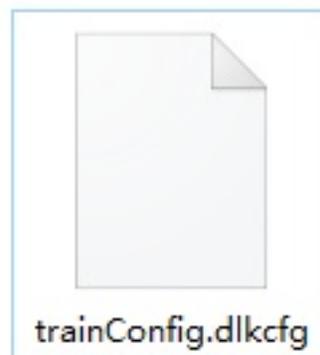


图 10 模型文件

## 4.4 术语概念

**标注：** 人工手动对图像上的目标物体进行框选。

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

**已标注：** 人工手动标注过的图像数据。

**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”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。

## 4.5 快捷键

编号	功能	快捷键	其他说明
1	新建工程	Ctrl + n	
2	存储工程	Ctrl + s	
3	打开工程	Ctrl + o	
4	撤销标注 (Undo)	Ctrl + y	
5	恢复标注 (Redo)	Ctrl + z	
6	复制标注	Ctrl + c	
7	粘贴标注	Ctrl + v	
8	全选标注	Ctrl + a	
9	删除标注	Delete	光标在标注页内，选中标注项
10	标注工具——椭圆	l	
11	标注工具——多边形	g	
12	标注工具——矩形	r	
13	标注工具——画笔	b	
14	标注工具——轮廓自动填充	a	
15	标注工具——橡皮擦	e	
16	标注工具——掩膜	m	
17	标注工具——选择	s	
18	清空当前页面标注	Ctrl + l	
19	删除数据集图片	Delete	光标在数据集窗口页
20	数据集等列表切换/滑动条拖拉	↑↓→←	光标在相应控件窗口页（默认快捷键）

1. 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。

2. 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。

3. 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。

4. 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。

5. 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。

6. 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。

7. 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。

- a. 数量：训练数据量是否足够使模型达到良好的效果。
- b. 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。
- c. 数据多样性：数据是否都涵盖了现场可能出现的所有情况。
- d. ROI 参数：现场设置的数值是否确保与训练时设置的一致。

- e. 光照：光照是否发生变化，是否与采集时保持一致。
8. 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。
9. 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。
10. 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。
11. 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，具体方法请参考[纸箱拆/码垛项目应用](#)。
12. 无 GPU 的情况下，图像分类模型是否能使用？  
不能。