无人机巡检图像缺陷识别技术测试

算法模型接口规范

**一、前言**

本文仅针对“无人机巡检图像缺陷智能识别技术验证”进行说明。

**二、算法集成调用要点**

**2.1 总体要求**

1.算法通过**docker**与缺陷识别模块服务器进行集成。各厂家只有一份docker镜像，该镜像能接收并处理本厂家支持的各个缺陷类型（一家一镜像）。

2.各算法运行过程中不需要与宿主机或其他机器/容器进行网络通信（不需要任何网络）。

3.每个算法容器的生命周期仅对应一次缺陷识别任务，识别任务结束，容器即被销毁。在平台启动新任务时，通过镜像重新启动算法容器。

**2.2 调用步骤**

1.算法厂家提交的镜像名称(厂家单位简称拼音)和版本号，注意全部为小写。

**docker tag image\_id repository\_name:tag**

**如：docker tag a8cd90sd1 厂家名拼音:v1.0**

2.算法厂家通过save命令生成镜像文件：

**docker save -o xx.tar new\_name:new\_tag**

**2.3 算法运行环境**

基于Nvidia计算资源的环境如表2.1所示。

**表2.1 算法运行环境**

|  |  |
| --- | --- |
| 操作系统/软件环境 | 版本 |
| Centos | 7.6 |
| 内存 | 16GB |
| Nvidia Driver | RTX 2080Ti：430.50 显存11GB；CUDNN 7.6.5 |
| CUDA | 10.1 |
| docker | 19.03 |

运行参数如表2.2所示。

**表2.2 算法运行参数**

|  |  |  |
| --- | --- | --- |
| 变量名 | 取值示意 | 说明 |
| --gpu | device=1 | **Nvidia环境下**：指定该算法容器可用的某块gpu |
| -v | my\_host\_dir:/usr/input\_picture:ro | 1. 指定本次任务要识别的图像目录； 2. 约定容器内用/usr/input\_picture 目录来映射输入图像目录； 3. ro 指算法容器对图像目录只有可读权限。 |
| -v | my\_host\_dir:/usr/output\_dir | 1. 指定本次识别任务中，算法的日志和结果文件输出目录； 2. 约定容器内用/usr/output\_dir/log 文件来存储执行日志； 3. 约定容器内用/usr/ouput\_dir/result.csv文件存储算法的输出结果（csv格式具体见“算法运行业务规则”）。 |
| -e | MODEL\_TYPES=00,01 | 约定以环境变量MODEL\_TYPES 为本次任务选定的缺陷类型，多类型间以逗号隔开，详细见“算法运行业务规则”。 |
| -e | NMS | 约定以环境变量 NMS 为非极大值抑制参数，详细见“算法运行业务规则” |
| -e | SCORE | 约定以环境变量 SCORE 为置信度和概率参数，详细见“算法运行业务规则” |

平台和算法容器间交互如上面表格所示。输入的图像路径、运行过程中的日志监控和运行结果文件通过中间文件进行交互通信。各厂家容器启动后，应能自启动算法识别程序，自动读取上述表格中的参数进行缺陷识别。

平台终止/销毁容器，通过**docker rm -f container\_id**命令终止当前算法容器。

算法容器的终止有两种场景

1. 正常结束：监控到运行日志输出**---process complete---**信息后；
2. 管理员中断：算法长期运行且无输出的情况下，管理员可强行中断当前算法的运行。

**2.4 算法说明文件**

厂家测试人员上传算法镜像文件，同时填写算法描述，具体见表格2.3

**表2.3 算法模型描述信息**

|  |  |  |
| --- | --- | --- |
| 序号 | 填写项 | 备注 |
| 1 | 镜像的repository名称 | 厂家单位简称拼音（全部为小写） |
| 2 | 镜像的tag值 | 版本号（小写） |
| 3 | 支持的缺陷类别 | 00-07（按实际支持类型） |
| 4 | 支持的计算环境 | Nvidia系列 |

厂家应同时提供和**repositoryName**同名的.json文件，命名为“厂家名拼音.json”，JSON文件包含算法厂家名称（其中vendor字段以厂家单位简称拼音命名）、算法镜像名称、算法镜像版本号、支持算法的GPU型号类型以及算法支持的缺陷模型四个部分。其中缺陷模型model字段填写如下：

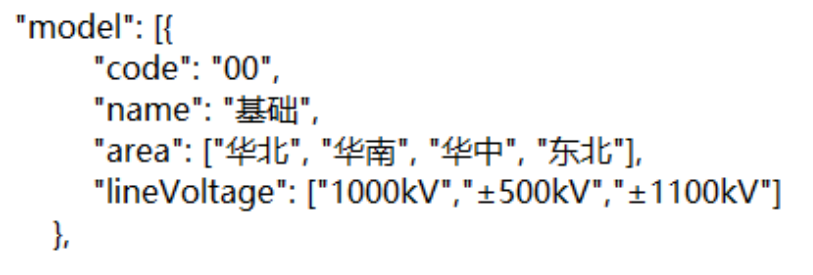
**“code”:** "00"： 填写缺陷字典第一类编码

**“name”:** "基础", 填写缺陷字典第一类名称

**“area”：**"[华北、华南]" 填写支持本类缺陷的地域（可按模板填）

**“lineVoltage”:**[”1000kV”,”00kV”] 填写支持本类的电压等级（可按模板填）

**“children”:**[070400021] 填写缺陷字典第一类下的缺陷小类，如果支持所有小类则可以不填，则没有”children”这项，示例如图2.1。



**图2.1 说明文件格式示例**

**三、算法运行业务规则**

**3.1算法执行进度日志**

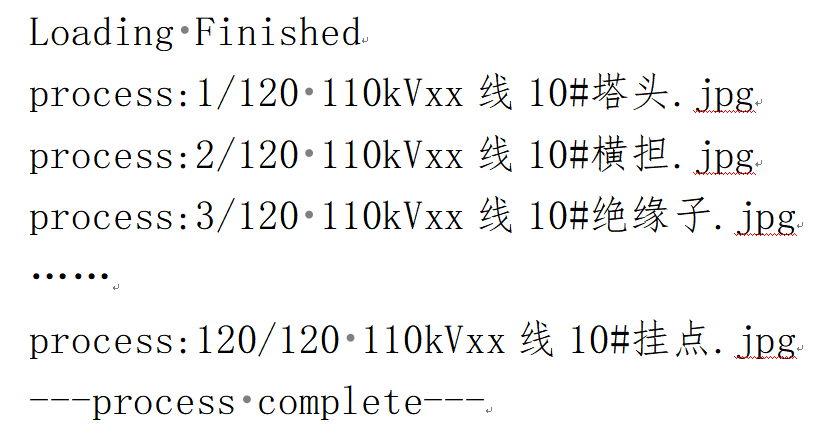
实时进度要写入容器内**/usr/output\_dir/log**文件中。

1.每张图片的进度打印信息，都以**“process:”**（没有空格）作为行首；

2.识别完成后输出**---process complete---**代表识别完成；

3.为方便获取总体进度，在对多大类缺陷进行识别时，必须保证每个图片的进度只被打印一次（目前发现有厂家对打印多轮图片进度，本版要合并成只打印一次）。

对一个文件夹下的120张图片执行检测，程序输出信息示例如图3.1。



**图3.1 执行进度格式示例**

**3.2算法运行状态日志**

厂家排查算法容器的异常状况见运行日志**program\_log**文件，路径约定为**/usr/output\_dir/program\_log**，将算法执行的各类执行细节日志输出至此文件中。**各厂家自行排查，Log文件格式自定，用于各自算法调试。平台仅需要按照规定格式的csv文件输出结果。**

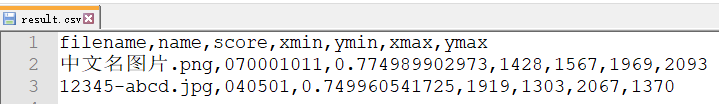
**3.3算法识别结果文件**

算法厂家的识别结果输出到**result.csv**文件中，原则上要求必须输出5层级别，且按json中支持的缺陷类型输出，不支持的可不输出。

同一位置仅允许输出一个缺陷框。目前发现有厂家针对同一位置输出了多个不同分类的算法框，这种情况会影响使用效果，需要纠正。若存在重复输出，多输出的全部判定为“错”。

识别结果文件共有7列，分别是：

**“filename,name,score,xmin,ymin,xmax,ymax”**，见图3.2。



**图3.2 输出结果格式示例**

其中**name**对应当前缺陷的描述级别，取值与训练集中的字典值相同，最短为两位，代表缺陷的第一层级，见表3.1。

**表3.1 缺陷name取值说明**

|  |  |
| --- | --- |
| **name取值** | **含义** |
| 00 | 基础 |
| 01 | 杆塔 |
| 02 | 导地线 |
| 03 | 绝缘子 |
| 04 | 金具 |
| 05 | 接地装置 |
| 06 | 通道环境 |
| 07 | 附属设施 |

金具类大、小尺寸金具算法输出的一级代码统一为04，小尺寸时输出增加二级代码即输出0405。

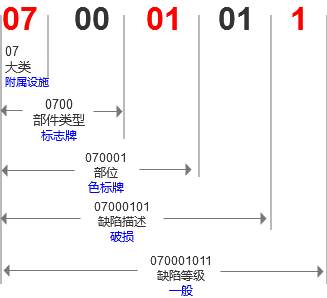
**name**最长为九位，代表缺陷的全五层级，表3.2示意了各长度对应的层级关系：

**表3.2 缺陷层级关系示例**

|  |  |  |
| --- | --- | --- |
| **name取值** | **缺陷识别精度** | **含义** |
| 04 | 1层 | 金具 |
| 0405 | 2层 | 金具-小金具 |
| 040501 | 3层 | 金具-小金具-螺母 |
| 04050102 | 4层 | 金具-小金具-螺母-缺螺母 |
| 040501023 | 5层 | 金具-小金具-螺母-缺螺母-危急 |

层级图示：

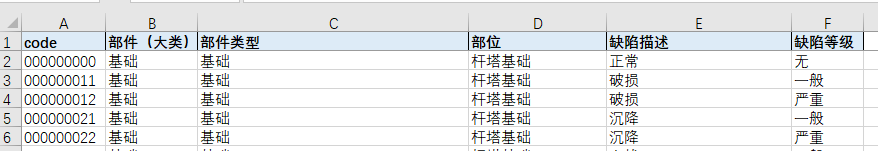
070001011“附属设施-标志牌-色标牌-破损-一般”的示意如图3.3。



**图3.3 缺陷层级示例**

**3.4识别目标编码**

图片中识别目标编码见链接中文件(双击可打开)，表头截图见图3.4。

****

**图3.4 缺陷目标字典表头截图**

**四、算法测试过程**

**4.1图片文件名称获取**

创建测试识别任务，在算法进行识别图片之前，平台将图片重命名后的文件名和原始文件名配对保存在json文件中，名为“**pictureName.json**”。

指定方式：**-v $jsonDir:/usr/input\_picture\_attach**

（$jsonDir 为宿主机存放json文件目录）

|  |
| --- |
| [  {  "fileName": "f4a3550d04484b11ac3cec9f8f5f7241.jpg",  "originFileName": "1000kV 曹湖Ⅰ线#84-塔全貌.jpg"  },  {  "fileName": "f4a3550d04484b11ac3cec9f8f5f7242.jpg",  "originFileName": "1000kV 曹湖Ⅰ线#84-塔头.jpg"  },  {  "fileName": "f4a3550d04484b11ac3cec9f8f5f7243.jpg",  "originFileName": "1000kV 曹湖Ⅰ线#84-左回下相导线端挂点.jpg"  },  {  "fileName": "f4a3550d04484b11ac3cec9f8f5f7244.jpg",  "originFileName": "1000kV 曹湖Ⅰ线#84-右回上相绝缘子.jpg"  },  {  "fileName": "f4a3550d04484b11ac3cec9f8f5f7245.jpg",  "originFileName": "1000kV 曹湖Ⅰ线#84-左回上相横担端挂点.jpg"  },  {  "fileName": "f4a3550d04484b11ac3cec9f8f5f7246.jpg",  "originFileName": "1000kV 曹湖Ⅰ线#84-大号侧通道"  }  ] |

备注：图片名中可能出现“空格”、“#“、“-“、“\_”等，文件后缀“jpg”、“JPG”大小写，算法应都支持。表中的文件名均为示例。

**4.2执行算法命令**

# 算法模型识别的样本库目录

inputDir=""

# 算法模型输出结果目录

outputDir=""

# 原始文件名存放目录

jsonDir=""

docker run --gpus device=0 #设置执行的gpu

-v /etc/localtime:/etc/localtime:ro #设置时区同步

-v $inputDir:/usr/input\_picture:ro #设置算法模型识别的样本库

-v $outputDir:/usr/output\_dir #设置算法模型输结果输出位置

-v $jsonDir:/usr/input\_picture\_attach #原始文件名存放目录

-e MODEL\_TYPES=00,01 #缺陷类型(约定规则)

-e NMS=0.6 #非极大值抑制参数(约定规则)

-e SCORE=0.3 #置信度和概率参数(约定规则)

-d 厂家名拼音:v1.7 #算法模型名称及版本号

**4.****3 算法执行流程和输出结果**

1.算法解析json文件，找到当前识别的图片原始名称，进行语义分析。算法读取文件中文名后，可自行灵活调用所支持的算法模型进行识别，从而有效提升发现率降低误报率（仅作为推荐策略，不强制要求）。例如，解析到当前图片名称为“杆塔”类时，若算法支持杆塔类模型的缺陷识别（对应docker命令中的MODEL\_TYPES参数），则可自动调用进行识别；若算法不支持杆塔类模型的缺陷识别（对应docker命令中的MODEL\_TYPES参数），则自动跳过此类型的识别。

2.厂家输出结果：只输出MODEL\_TYPES参数中的类型。

附件1

**Josn说明文件文件内容**

|  |
| --- |
| {  "vendor": "xxxxx",  "version": "v8.1.2",  "repositoryName": "xxxx",  "gpuTypeList": ["RTX 2080Ti"],  "model": [{  "code": "00",  "name": "基础",  "area": ["华北", "华南", "华中", "东北"],  "lineVoltage": ["1000kV","±500kV","±1100kV"],  "children": [  "000000051",  "000000061",  "000000151",  "000002011",  "000000091",  "000000081",  "000000111",  "000000021",  "000000101",  "000000011"  ]  },  {  "code": "01",  "name": "杆塔",  "area": ["华北", "华南", "华中", "东北"],  "lineVoltage": ["1000kV","±500kV","±1100kV"],  "children": [  "010000021",  "010000031",  "010002011"  ]  },  {  "code": "02",  "name": "导地线",  "area": ["华北", "华南", "华中", "东北"],  "lineVoltage": ["1000kV","±500kV","±1100kV"],  "children": [  "020100031",  "020000031",  "020001031",  "020000111",  "020001021",  "020001013"  ]  },  {  "code": "03",  "name": "绝缘子",  "area": ["华北", "华南", "华中", "东北"],  "lineVoltage": ["1000kV","±500kV","±1100kV"],  "children": [  "030200131",  "030000011",  "030100021",  "030200061",  "030000071"  ]  },  {  "code": "04",  "name": "金具",  "area": ["华北", "华南", "华中", "东北"],  "lineVoltage": ["1000kV","±500kV","±1100kV"],  "children": [  "040303031",  "040303011",  "040000071",  "040000011",  "040202011",  "040205011",  "040201011",  "040102031",  "040303041",  "040100011",  "040209011",  "040500013",  "040501031",  "040500021",  "040501021",  "040001041",  "040500031"  ]  },  {  "code": "05",  "name": "接地装置",  "area": ["华北", "华南", "华中", "东北"],  "lineVoltage": ["1000kV","±500kV","±1100kV"],  "children": [  "050000011",  ]  },  {  "code": "06",  "name": "通道环境",  "area": ["华北", "华南", "华中", "东北"],  "lineVoltage": ["1000kV","±500kV","±1100kV"],  "children": [  "060800031",  "060700021",  "060800021",  "060300021",  "060800011",  "060400041",  "060500021"  ]  },  {  "code": "07",  "name": "附属设施",  "area": ["华北", "华南", "华中", "东北"],  "lineVoltage": ["1000kV","±500kV","±1100kV"],  "children": [  "070400021",  "070002011",  "070002021",  "070000031",  "070002041"  ]  },  ]  } |