

T/BAX

团体标准

T/BAX XXXX.2—2023

视频图像感知智能应用适配技术要求 第2部分：算法适配技术要求

Technical requirements for intelligent application adaptation of video and image sensing—Part 2: Algorithm adaptation technical requirements

(征求意见稿)

XXXX—XX—XX 发布

XXXX—XX—XX 实施

北京安全防范行业协会 发布

目 次

前 言	III
引 言	IV
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 基本要求	1
4.1 算法形态要求	1
4.2 算法封装要求	1
5 功能要求	3
5.1 算法包功能要求	3
5.2 算法引擎功能要求	3
6 授权要求	4
7 接口要求	5
7.1 算法包接口	5
7.2 算法引擎接口	9
附 录 A （规范性） 算法描述文件	13
A.1 算法技术描述信息文件 - YAML 样例	13
A.2 算法管理扩展信息文件 - YAML 样例	15
附 录 B （资料性） 算法引擎对象属性	16
B.1 表中选项字符说明	17
B.2 基础数据结构	17
B.3 算法引擎状态	18
B.4 解析任务参数	18
B.5 设备参数配置	19
B.6 解析规则参数	21
B.7 解析任务状态	22
B.8 解析任务列表	23
B.9 解析任务操作结果	23
B.10 目标结构化类 - 解析输出结果	24
B.11 事件检测类 - 解析输出结果	25
B.12 态势统计类 - 解析输出结果	25
B.13 其他类 - 解析输出结果	26
附 录 C （资料性） 算法引擎对象样例数据	28
C.1 算法引擎状态	28
C.2 解析任务	28
C.3 设备配置	28

C.4 解析规则参数	29
C.5 解析任务状态	30
C.6 解析任务列表	30
C.7 解析任务操作结果	30
附录 D (资料性) 算法包接口 C 语言描述	31
D.1 函数命名规则	31
D.2 数据类型定义	31
D.3 数据格式定义	33
D.4 函数返回值定义	34
D.5 设置算法日志函数	34
D.6 设置算法授权信息函数	34
D.7 检测识别算法初始化函数	35
D.8 创建分析通道函数	35
D.9 调用分析算法函数	35
D.10 关闭分析通道函数	35
D.11 释放算法函数	36
D.12 分析结果回调函数	36
D.13 获取系统信息回调函数	36
D.14 获取目标 ID 回调函数	36
D.15 单帧同步分析函数	36
D.16 设置检测识别回调接口函数	37
附录 E (规范性) 算法输入输出参数 JSON Schema	38
E.1 算法输入参数 - JSON Schema 样例	38
E.2 事件检测类算法自定义输出参数 AlarmEvent - JSON Schema 样例	52
E.3 态势统计类算法自定义输出参数 StatusInfo - JSON Schema 样例	55
E.4 其他类算法自定义输出参数 UncategorizedInfo - JSON Schema 样例	56

前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

本文件是T/BAX XXXX—2023《视频图像感知智能应用适配技术要求》的第2部分。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由北京安全防范行业协会提出并归口。

本文件起草单位：北京市公安局、北京市经济和信息化局、视频图像信息智能分析与共享应用技术国家工程实验室、北京旷视科技有限公司、北京百度网讯科技有限公司、富盛科技股份有限公司、北京密码云芯科技有限公司、北京瑞莱智慧科技有限公司。

本文件主要起草人：陶山、步飞、崔云红、赵惠芳、王建勇、张伟、李标、张晨光、麻思、钟永强、孙马秋、张浩天。

引 言

视频图像感知智能应用是国家数字化进程和智慧城市建设中的重要内容，在公共安全管理、社会治理、社区管理等应用场景中，发挥关键作用。

T/BAX XXXX《视频图像感知智能应用适配技术要求》通过建立统一的适配技术规范，指导视频图像解析算法与设备及平台的适配，解决算法与设备及平台的紧耦合问题，实现算法随业务需求持续快速更新迭代，并通过云边端算力动态调度，逐步实现算法快速部署应用。

T/BAX XXXX拟由五部分构成。

- 第1部分：总体技术要求。目的在于确定视频图像感知智能化应用适配的总体结构，以及适配需要遵循的总体要求和具体要求等技术要求。
- 第2部分：算法适配技术要求。目的在于规范视频图像感知智能化应用中算法适配的基本要求、功能要求、算法授权要求、接口要求等技术要求。
- 第3部分：感知前端和边缘解析设备适配技术要求。目的在于规范视频图像感知智能化应用中感知前端和边缘解析设备的基本要求、功能要求、接口要求等技术要求。
- 第4部分：中心算力适配技术要求。目的在于规范视频图像感知智能化应用中中心算力的总体框架、功能要求、接口要求等技术要求。
- 第5部分：算法算力服务管理平台技术要求。目的在于规范视频图像感知智能化应用中算法接入、算法管理、算力接入、算力管理、视频图像接入、数据共享等技术要求。

视频图像感知智能应用适配技术要求

第 2 部分：算法适配技术要求

1 范围

本文件规定了视频图像感知智能应用适配技术要求中的算法适配基本要求、功能要求、授权要求和接口要求。

本文件适用于视频图像感知智能应用适配中算法适配的规划设计、软件开发、检测和验收。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GA/T 1399.1—2017 公安视频图像分析系统 第1部分：通用技术要求

GA/T 1400.1—2017 公安视频图像信息应用系统 第1部分：通用技术要求

GA/T 1400.4—2017 公安视频图像信息应用系统 第4部分：接口协议要求

GB/T 28181-2022 公共安全视频监控联网系统信息传输、交换、控制技术要求

T/CSPIPA 005-2021 公共安全视频图像分析多算法应用技术要求

T/BAX XXXX.1—202X 视频图像感知智能应用适配技术要求 第1部分：总体技术要求

T/BAX XXXX.5—202X 视频图像感知智能应用适配技术要求 第5部分：算法算力服务管理平台技术要求

3 术语和定义

GA/T 1399.1—2017、GA/T 1400.1—2017、视频感知智能应用适配技术要求 第1部分：总体技术要求界定的以及下列术语和定义适用于本文件。

3.1

视频图像智能解析算法封装文件 video and image analysis intelligent algorithm perception file

算法封装文件是由算法描述文件和算法包或算法引擎结合在一起的一种算法成果物实体。并将作为算法的实体进行交付、分发、部署。

4 基本要求

4.1 算法形态要求

视频图像智能解析算法应支持算法包或算法引擎的形态，其中感知前端、边缘解析设备建议支持算法包的形态。中心解析设备、中心计算平台、中心解析平台建议支持算法引擎形态。

4.2 算法封装要求

算法应按照统一的文件目录结构进行封装，算法封装文件内容示意图参考图1。

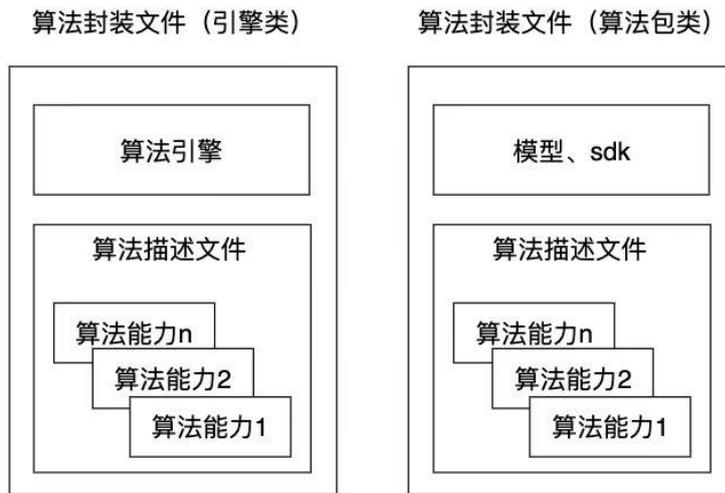


图 1 算法封装文件示意图

算法描述文件应至少包含以下两类文件：

- a) 算法技术描述信息文件，其中应包含算法名称、版本、运行设备、运行参数等信息；
- b) 算法管理拓展信息文件，其中应包含厂商管理信息（厂商名称、编码、联系方式、售后服务地址等）和算法管理信息（功能介绍、分类、适用场景、算法封面等）；

4.2.1 命名要求

算法包名称命名规则：算法封装文件标识_版本号.zip，详细命名规则应符合附录A.1的规定。

4.2.2 目录结构要求

算法封装文件目录结构应符合表1的规定。

表 1 算法封装文件目录结构

序号	目录	要求	文件	要求	说明
1	spec	必选	algo.yaml	必选	算法技术描述信息文件，包含算法名称、版本、运行设备、运行参数等，参考附录A.1算法技术描述信息文件样例。 注意：此文件会被下发至前端、边缘设备和中心计算平台中，作为算法运行的配置文件。
			managementinfo.yaml	必选	算法管理拓展信息文件，包含有算法的语义化描述：算法功能介绍、分类、适用场景、算法封面等，主要用于算法信息的门户展示，参考附录A.2算法管理描述信息文件样例
			assets	可选	存放各类媒体文件和算法自定义参数描述json。包括算法展示图片、算法封面图片、算法分析结果样例图片、算法演示视频等。在算法管理拓展信息文件中，会有字段指向此目录中的具体图片和视频文件。 注意：此目录下的多媒体文件，不宜被下发至感知前端、中心解析设备、中心计算平台、中心解析平台中。
2	bin	必选		-	存放算法包或算法引擎文件及相关配置文件： 算法包文件，常见包括服务器和边端设备的C函数头文件及相关二进制链接库文件； 算法引擎文件，常见包括服务器 docker 容器镜像文件，边端设备嵌入式容器镜像文件，边端设备可执行文件
3	data	可选		-	存放算法运行时需要挂载的算法模型。如果算法模型已经包含在算法包或者算法引擎镜像文件中，此处可以不单独存在。在算法包形态下，算法模型宜

序号	目录	要求	文件	要求	说明
					单独存在。在算法引擎形态下，算法模型宜包含在镜像文件内。
4	extra	可选		-	存放算法运行时需要挂载相关视图数据。根据需要，此处可以存放视图数据，用于算法效果的测试校验或增量训练等进阶需求。 注意：此目录下的文件，不宜被下发至感知前端、中心解析设备、中心计算平台、中心解析平台中。

5 功能要求

5.1 算法包功能要求

5.1.1 算法配置

算法配置满足以下要求：

- 应支持配置算法包的日志回调函数，算法包通过日志回调函数打印日志信息；
- 应支持配置算法包的初始化信息，包括算法包路径、处理结果回调函数、初始化参数、运行环境信息等。

5.1.2 算法分析

算法分析满足以下要求：

- 应支持算法任务创建、删除、详情查看、列表查看功能；
- 应支持分析结果回调、获取目标 ID 回调；
- 应支持单帧同步分析、异步分析；
- 宜支持设置定制回调，用于为算法包的特定分析通道单独设置定制的分析结果回调函数。

5.1.3 运行监测

运行监测满足以下要求：

- 应支持对算法包的运行状态（如加载，初始化，释放等状态）进行监测；
- 宜支持对算法包的运行中的分析任务的状态进行监测；
- 宜支持对算法包的系统资源占用情况进行监测。

5.1.4 算法授权

算法授权满足以下要求：

- 应支持在线授权，通过配置算法包申请授权的服务地址（如授权管理服务器的 IP 地址和端口等），算法包通过与 服务地址通信申请授权或释放授权；
- 应支持离线授权，离线授权文件导入设备，从而为该设备上的算法包运行提供本地授权验证能力；
- 当使用在线授权途径完成授权时，如遇到算法包无法通过服务地址通信申请授权时，应支持通过调用算法包回调函数返回的错误信息监测到算法授权异常；
- 当通过离线授权途径完成授权时，应支持查询算法包的授权信息。

5.2 算法引擎功能要求

5.2.1 算法配置

算法配置应支持通过环境变量方式配置算法引擎的相关配置，包括：日志参数配置、算法引擎路径、配置文件路径等。

5.2.2 算法分析

算法分析满足以下要求：

- a) 应支持算法任务创建、删除、详情查看、列表查看等功能；
- b) 应支持通过符合 GB/T 28181 规定的信令与公共安全视频监控联网系统进行交互，获取视频流进行分析处理；
- c) 应支持通过符合 GA/T 1400.4—2017 规定的信令与公共安全视频图像信息数据库进行交互，获取图片进行分析处理。

5.2.3 运行监测

运行监测满足以下要求：

- a) 应支持对算法引擎的运行状态（如启动，停止，异常等状态）进行监测；
- b) 应支持对算法引擎的运行中的分析任务的状态进行监测；
- c) 宜支持对算法引擎的系统资源占用情况进行监测。

5.2.4 算法授权

算法授权满足以下要求：

- a) 应支持在线授权，通过配置算法引擎申请授权的服务地址（如授权管理服务器的 IP 地址和端口等），算法引擎通过与授权服务地址通信申请授权或释放授权；
- b) 应支持离线授权，离线授权文件导入设备，从而为该设备上的算法引擎运行提供本地授权验证能力；
- c) 当通过在线授权途径完成授权时，算法引擎无法通过服务地址通信申请授权时，应支持通过算法引擎返回的错误信息监测到算法授权异常；
- d) 当通过离线授权途径完成授权时，应支持查询算法引擎的授权信息。

6 授权要求

6.1.1 基本要求

算法包或算法引擎的算法授权应支持在线授权或离线授权中的至少一种，从而合理约束算法服务的使用。具体需满足如下要求：

- a) 在线授权交付物应能满足在线授权交付物要求；
- b) 在线授权接口和服务流程应能符合 T/BAX XXXX. 5—202X（《视频图像感知智能应用适配技术要求 第 5 部分：算法算力服务管理平台技术要求》）中附录 D 的要求；
- c) 离线授权交付物应能满足离线授权交付物要求；
- d) 离线授权服务流程应能符合 T/BAX XXXX. 5—202X（《视频图像感知智能应用适配技术要求 第 5 部分：算法算力服务管理平台技术要求》）中附录 D 的要求。

6.1.2 在线授权交付物要求

当采用在线授权方案的情况下，算法厂商的交付物应符合表2的规定。

表 2 在线授权交付物要求

交付物名称	交付物形态	交付物功能
算法授权验证服务	部署在中心的服务	提供基于网络授权验证能力，通过与算法服务保持心跳通信，为相关服务提供运行许可。
算法授权验证服务-中心环境信息生成工具	运行在中心的脚本	用于收集授权验证服务所在机器的环境信息，生成环境文件，用于手动或者自动向算法厂商申请相关授权许可更新文件。
gen_cloud_env.sh	用于收集授权验证服务所在机器的环境信息，生成环境文件，用于手动或者自动向算法厂商申请相关授权许可更新文件。	用于将算法厂商提供的授权许可更新文件，导入中心算法授权的验证服务，更新其授权许可信息，从而为该系统中的所有边缘设备提供网络授权验证能力。

6.1.3 离线授权交付物要求

离线授权方案的情况下，算法厂商的交付物应符合表3的规定。

注：如果是试用场景，算法厂商可以选择采用“不绑定设备的算法试用离线授权文件”（`trial_license.key`）简化整体服务流程。此文件，可以跳过生成设备指纹的流程，直接被设备加载并试用，方便客户体验算法效果。

表 3 离线授权交付物要求

交付物名称	交付物形态	交付物功能
算法授权签发服务	部署在中心的服务	基于边端计算设备的指纹，以及需要运行的算法包，生成离线授权文件。
设备指纹生成工具	运行在边端计算设备的脚本 <code>gen_device_fingerprint.sh</code>	用于收集边端计算设备的指纹，并生成指纹文件。后续，可以基于此指纹文件，向算法厂商的授权签发服务发出请求，生成授权文件
设备离线授权文件导入工具	运行在边端计算设备的脚本 <code>load_device_license.sh</code>	用于将离线授权文件导入设备，从而为该设备上的算法服务运行提供本地授权验证能力。

7 接口要求

7.1 算法包接口

7.1.1 接口调用

7.1.1.1 接口调用流程

感知前端或边缘解析设备调用算法包接口的调用流程应符合图 2 的规定，主要包括配置算法包、初始化算法包，创建算法分析通道、创建分析任务、返回或回调分析结果、关闭算法分析通道、释放算法包等。

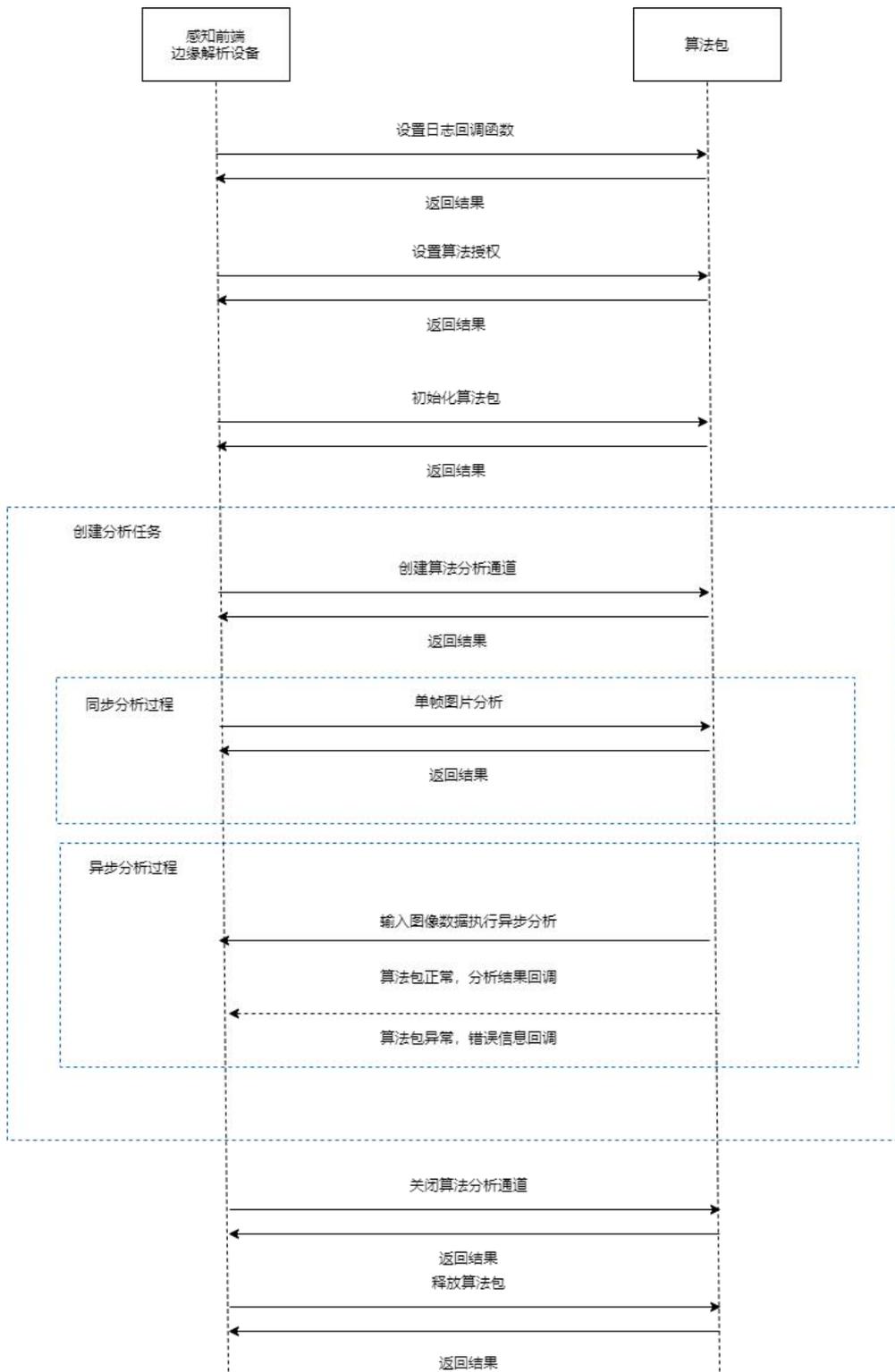


图 2 算法包接口调用流程

7.1.1.2 调用算法包分析视频流要求

调用算法包分析视频流，每一路采集设备的视频流使用一个分析通道进行分析。感知前端或边缘解析设备应具备视频对接的基础运行环境能力如：取流、解码、相关视频平台 API 调用、结果处理等并调用算法包接口完成算法分析。

7.1.1.3 调用算法包分析图片流要求

调用算法包分析图片流，每一路采集设备的图片流宜使用一个分析通道串行分析，或使用多个分析通道并行分析。多路采集设备的图片流也可以使用同一个通道进行分析。感知前端或边缘解析设备应具备图片对接的基础运行环境能力如：取图、相关图片数据平台 API 调用、结果处理等并调用算包接口完成算法分析。

7.1.2 配置接口

7.1.2.1 设置算法包授权信息函数

设置算法包授权信息函数传入授权管理服务器的访问地址，由算法包自行与授权管理服务器进行通信完成授权申请。在调用算法包进行分析之前，先调用设置算法包授权信息函数，调用参数应符合表4的规定，函数返回值参见附录D中D.4定义。

表 4 设置算法包授权信息函数参数列表

参数	参数类型	参数说明
授权信息	输入参数	授权信息所在的服务器访问地址（服务器的IP和端口或服务器域名）
信息长度	输入参数	授权信息长度

7.1.2.2 设置算法包日志函数

设置算法包日志函数用于设置算法包内部记录日志的回调函数，算法包通过调用日志回调函数将日志信息打印出来。在调用算法包进行分析之前，先调用设置算法包日志函数，调用参数应符合表5的规定，函数返回值见附录D中D.4定义。

表 5 设置算法包日志函数参数列表

参数	参数类型	参数说明
日志级别	输入参数	表示只会输出高于该日志级别的日志，数值越小级别越高
日志记录回调函数	输入参数	定义算法包的日志记录回调函数，算法包调用该函数记录日志

7.1.3 检测识别接口

7.1.3.1 检测识别算法初始化函数

算法包初始化函数传入算法包的配置信息XML文件、设置回调函数、设置算法包路径等，初始化算法包运行环境。在调用算法包进行目标检测识别之前，先调用设置分析算法包初始化函数，调用参数应符合表6的规定，函数返回值见附录D中D.4定义。

表 6 检测识别算法初始化函数参数列表

参数	参数类型	参数说明
配置文件	输入参数	算法包初始化需要用到的XML
内容说明	输入参数	设置算法包分析结果回调函数，获取当前算法包目录，获取目标ID回调函数，XPU 卡号等

7.1.3.2 创建分析通道函数

创建分析通道函数用于传入创建分析通道的XML配置，并为算法包创建分析任务。在调用算法包进行目标检测识别之前，先调用创建分析通道函数，调用参数应符合表7的规定，函数返回分析通道编号（大于等于0）。

表 7 创建分析通道函数参数列表

参数	参数类型	参数说明
----	------	------

参数	参数类型	参数说明
通道配置信息	输入参数	分析通道创建的XML

7.1.3.3 调用分析算法函数

调用分析算法函数用于将视频流/图片流解码后的图像信息发送给算法包并进行分析。调用分析算法函数的调用参数应符合表8的规定，函数返回值见附录D中D.4定义。

表8 调用分析算法函数参数列表

参数	参数类型	参数说明
通道号	输入参数	分析通道创建时的通道号
图像信息	输入参数	包括视频帧率、帧号、图片格式、尺寸、时间戳、地址类型等
私有数据	输入参数	系统私有数据，算法内部无需处理，回调分析结果数据时，透传给上层应用

7.1.3.4 关闭分析通道函数

关闭分析通道函数的调用参数应符合表9的规定，函数返回值见附录D中D.4定义。

表9 关闭分析通道函数参数列表

参数	参数类型	参数说明
通道号	输入参数	分析通道创建时的通道号

7.1.3.5 释放算法函数

释放算法函数无参数调用，无返回值。

7.1.3.6 分析结果回调函数

分析结果回调函数用于算法包分析完成之后，将分析结果返回，该函数与8.3.3配合使用，实现完整的检测识别任务。分析结果回调函数的调用参数应符合表10的规定，函数返回值见附录D中D.4定义。

表10 分析结果回调函数参数列表

参数	参数类型	参数说明
通道号	输入参数	分析通道创建时的通道号
目标号	输入参数	目标ID
分析结果元数据	输入参数	目标的属性信息和特征向量数据
数据长度	输入参数	分析结果元数据长度
关键帧图片信息	输入参数	场景图、目标小图的图像信息。包括图片格式、尺寸、时间戳、地址类型、目标个数等
轨迹信息	输入参数	包括轨迹长度，轨迹方框等。参数格式说明见附录C中C.3
私有数据	输入参数	多算法应用系统私有数据，算法内部无需处理，保持透传

7.1.3.7 获取系统信息回调函数

获取系统信息回调函数用于算法包获取多算法应用系统的系统信息。获取系统信息回调函数的调用参数应符合表11的规定，函数返回值见附录D中D.4定义。

表11 获取系统信息回调函数参数列表

参数	参数类型	参数说明
信息类型	输入参数	系统信息类型
信息缓存	输出参数	系统信息内容buffer
缓存长度	输出参数	系统信息buffer长度

7.1.3.8 获取目标 ID 回调函数

获取目标ID回调函数用于生成一个多算法应用系统内全局唯一的目標ID。算法包调用获取目标ID回调函数，为每个被算法检测到的目标生成目标ID。获取目标ID回调函数的调用参数应符合表12的规定，函数返回值见附录D中D.4定义。

表 12 获取目标 ID 回调函数参数列表

参数	参数类型	参数说明
通道号	输入参数	分析通道创建时的通道号
时间	输入参数	目标生成时的时间戳
类型	输入参数	目标类型： 01-人员； 02-机动车； 03-非机动车； 04-物品； 05-场景； 06-人脸； 07-视频图像标签； 99-其他

7.1.3.9 单帧同步分析函数

单帧同步分析函数用于算法包获取单帧图片进行分析并同步返回分析结果。单帧同步分析函数的调用参数应符合表13的规定，函数返回值见附录D中D.4定义。

表 13 单帧同步分析函数参数列表

参数	参数类型	参数说明
通道号	输入参数	分析通道创建时的通道号
图像信息	输入参数	包括图片格式、尺寸、时间戳、地址类型等
私有数据	输入参数	多算法应用系统私有数据，算法内部无需处理，保持透传
目标号	输出参数	目标ID
分析结果元数据	输出参数	protobuf结构
数据长度	输出参数	元数据信息长度

7.1.3.10 设置定制回调函数

设置定制回调函数用于为算法包的特定分析通道单独设置定制的分析结果回调函数。设置定制回调函数的调用参数应符合表14的规定，函数返回值见附录D中D.4定义。

表 14 设置定制回调函数参数列表

参数	参数类型	参数说明
通道号	输入参数	分析通道创建时返回的通道号
回调函数类型	输入参数	标识要设置的回调函数类型，取值参见附录C.4定义的VICA_CALLBACK_TYPE E
回调函数指针	输入参数	回调函数指针
私有数据	输入参数	多算法应用系统私有数据，算法内部无需处理，保持透传

7.2 算法引擎接口

7.2.1 接口调用

7.2.1.1 接口调用说明

中心解析设备、中心计算平台、中心解析平台调用算法引擎接口的调用流程应符合图3的规定，主要包括配置启动参数和相关环境变量配置、启动算法引擎、创建解析任务、返回解析结果、删除解析任

务、停止算法引擎等。

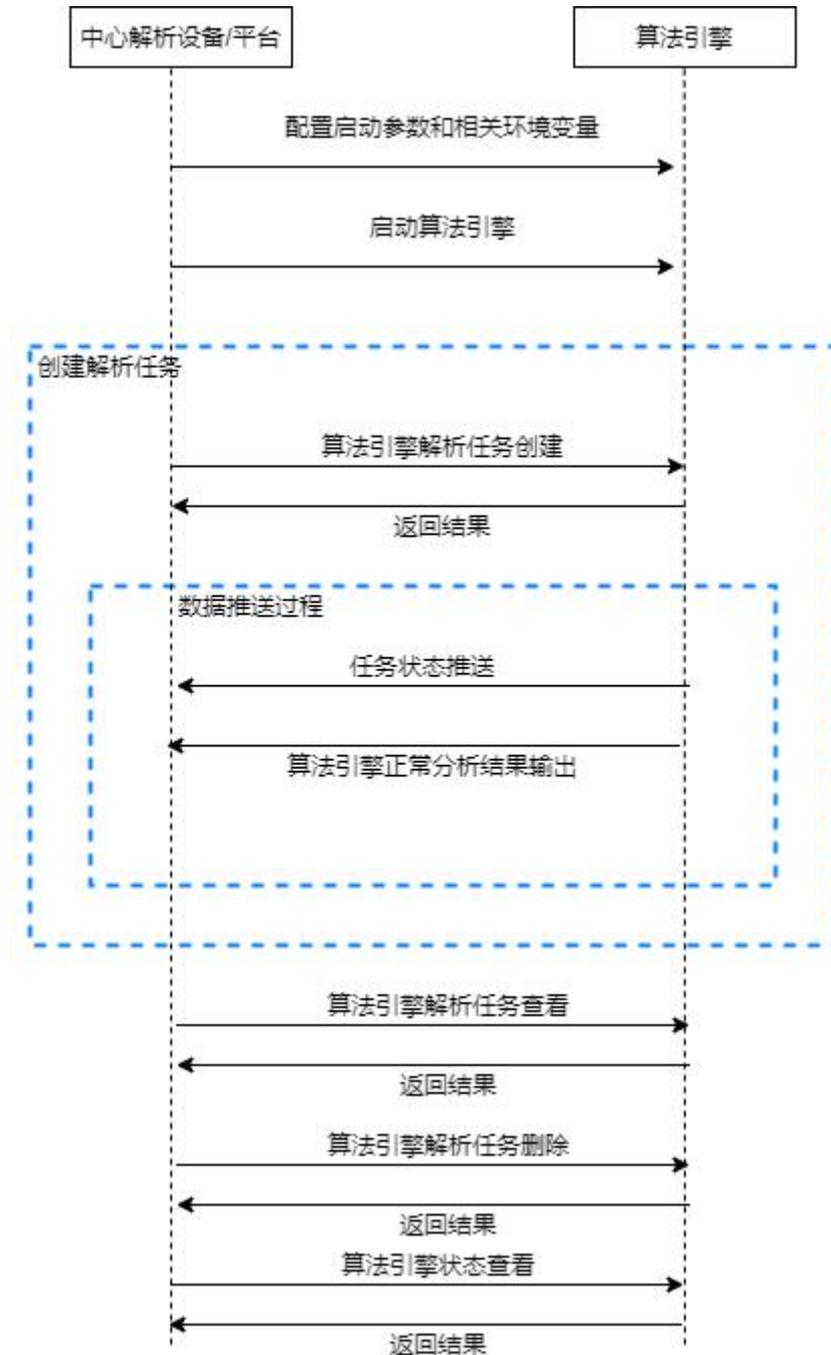


图 3 算法引擎接口调用流程

7.2.1.2 调用算法引擎分析视频流说明

调用算法引擎分析视频流，每一路采集设备的视频流使用一个分析通道进行分析。算法引擎应具备视频对接的基础能力如：取流、解码、相关视频平台 API 调用、结果处理等，以及算法分析能力并开放算法引擎接口供中心解析设备、中心计算平台、中心解析平台调用完成算法分析。

7.2.1.3 调用算法引擎分析图片流说明

调用算法引擎分析图片流，每一路采集设备的图片流宜使用一个分析通道串行分析，或使用多个分析通道并行分析。多路采集设备的图片流也可以使用同一个通道进行分析。算法引擎应具备视频对接的基础

能力如：相关视频平台API调用、取图、结果处理等，以及算法分析能力并开放算法引擎接口供中心解析设备、中心计算平台、中心解析平台调用完成算法分析。

7.2.2 算法引擎启动配置环境变量参数

作为中心解析设备、中心计算平台、中心解析平台启动算法引擎的配置选项，并在算法技术描述信息文件中声明，应符合附录A.1的规定。

算法引擎的相关配置环境变量参数应符合表15的规定。

表 15 算法引擎启动配置环境变量参数列表

参数名	参数类型	参数说明
LOG_LEVEL	环境变量	算法引擎日志级别，默认可配置级别从低到高可选值如下：
AUTH_SERVER_ADDR	环境变量	算法引擎授权验证地址，具体取值按照实际情况可能为以下三种：

7.2.3 接口协议结构

算法引擎接口协议结构应符合GA/T 1400.4—2017中4.2的要求。在接口消息的URI中使用“VIAE”标识“算法引擎”。

7.2.4 算法引擎接口列表

支持中心解析设备、中心计算平台、中心解析平台对算法引擎的调度管理和任务配置。算法引擎的API接口应符合表16-21的规定。

表 16 算法引擎接口列表

URI	方法	功能说明
/VIAE/Status	GET	返回算法引擎的当前状态
/VIAE/Tasks	GET	返回算法引擎上在运行的所有解析任务
/VIAE/Tasks	POST	在算法引擎上创建解析任务
/VIAE/Tasks/{id}	GET	获取算法引擎上指定 id 的解析任务
/VIAE/Tasks/{id}	DELETE	删除算法引擎上指定 id 的解析任务

详细对象属性描述，参见附录B

表 17 算法引擎状态查询

URI	/VIAE/Status		
功能	查询算法引擎状态信息		
方法	查询字符串	消息体	查询字符串
GET	无	无	无
注释	正常情况返回的是EngineStatusResponse对象，定义应符合B.3的规定；出现异常返回ErrorStatusResponse对象，定义应符合B.3的规定，示例见C.1		

表 18 算法引擎解析任务列表查询

URI	/VIAE/Tasks		
功能	查询算法引擎上在运行的所有解析任务信息		
方法	查询字符串	消息体	查询字符串
GET	无	无	无
注释	正常情况返回的是ListTaskResponse对象，定义应符合B.8的规定；出现异常返回ErrorStatusResponse对象，定义应符合B.3的规定，示例见C.6		

表 19 算法引擎解析任务创建

URI	/VIAE/Tasks
-----	-------------

URI	/VIAE/Tasks		
功能	创建一个运行在算法引擎上的解析任务		
方法	查询字符串	消息体	查询字符串
POST	无	TaskInfoRequest	无
注释	TaskInfoRequest对象，定义应该符合A. 3的规定 正常情况返回的是TaskInfoResponse对象，定义应符合B. 4的规定； 出现异常返回ErrorStatusResponse对象，定义应符合B. 3的规定，示例见C. 2。		

表 20 算法引擎单个解析任务查询

URI	/VIAE/Tasks/{uniqueId}		
功能	查询算法引擎上在运行的一个解析任务信息		
方法	查询字符串	消息体	查询字符串
GET	uniqueId	无	uniqueId
注释	正常情况返回的是TaskInfoResponse对象，定义应符合B. 4的规定； 出现异常返回ErrorStatusResponse对象，定义应符合B. 3的规定，示例见C. 2。		

表 21 算法引擎解析任务删除

URI	/VIAE/Tasks/{uniqueId}		
功能	删除算法引擎上在运行的一个解析任务		
方法	查询字符串	消息体	查询字符串
DELETE	uniqueId	无	uniqueId
注释	正常情况返回的是DeleteTaskResponse对象，定义应符合B. 9的规定 出现异常返回ErrorStatusResponse对象，定义应符合B. 3的规定，示例见C. 7。		

附录 A (规范性) 算法描述文件

A.1 算法技术描述信息文件 - YAML 样例

```

# 算法封装文件标识信息
# algoId: uuid 格式定义：算法算力服务平台编码（20位）+ 年月日时分秒（14位）+ 厂商（4
位，算法算力服务平台定义厂商字典）+增量序号（2位自增）共40位
# [required] 算法名称
name: example
# [required] 算法版本, 必须遵守 Semantic Version
version: 2.0.1

# 算法通用基本信息
# [required] 算法提供者, 公司名称或者个人姓名
provider: test
# [optional] 算法类型: cloud(服务器算法), edge(嵌入式算法)
type: cloud
# [required] 算法描述: 算法的发版说明, 文档链接等
description: |
  videoEngine is the most efficient analysis service.
# [optional] 算法标识: 算法标识信息, 可任意添加有含义的字段。
tags:
  - commit=5613def

# 可执行文件描述, 作为 image_name 的上位替代品
bin:
  # 可执行文件的类型:
  # 一种是常用于算法引擎集成模式, 例如服务器算法下发的容器镜像 (image);
  # 一种是常用于算法包集成模式, 例如嵌入式算法下发的包含可执行文件的打包文件(artifact)
tar 包:
  type: image | artifact
  # 可执行文件的本地 uri, 被打包工具用于指向可执行文件的包内位置
  local_url: bin/image.tar

# [required] 算法运行时信息
algoRuntime:
# [required]服务器算法镜像启动数
cmd: bash -c 'python main.py --foo=bar'
# [required]服务器算法环境变量
env:
  - name: LOG_LEVEL
    value: INFO
    description: '算法引擎日志级别, 默认可配置级别从低到高可选值如下: DEBUG: 调试级
别, INFO: 正常生产环境日志级别 (默认值), WARN: 警告级别, ERROR: 错误级别, FATAL: 致命级别'
  - name: AUTH_SERVER_ADDR
    value: localhost:8765

```

description: '算法引擎授权验证地址, 具体取值按照实际情况可能为以下三种ip:port, hostname:port, domain:port; 对于在线授权方案, 此地址为远端地址; 对于离线授权方案, 此地址为本地地址。'

```

- name: EXAMPLE_ENV
  value: foobar
  description: '厂商自定义扩展环境变量参数'
# [required]服务器算法暴露端口
port:
- 3223
# [required] 算法依赖的计算资源
resources:
- xpu:
  # [required] 算法运行时需要占用的XPU个数, 建议取值: 25/50/100/200/400等。
  limit:50 # 占用50%
  # [required] 加速卡家族, 支持常见卡品牌;
  vendor: NVIDIA | HW Ascend
  # [required] 加速卡类型, 填写所支持具体卡类型
  accelerator: T4 | Atlas-300V-Pro
  # [required] 算法能力说明, 对不同显卡类型可配置不
cpu:
  # cpu核数
  limit: 2000m
memory:
  # 内存占用, 单位MB
  limit: 6000M
capacities:
- specification: 200W # 1920*1080
  # [required] 支持的视频分析路数, 默认0
  stream_video_quota: 30
  # [required] 支持的图片分析路数, 默认0
  stream_capture_quota: 10
  # [required] 支持的图片静态分析QPS, 默认0
  analyze_quota: 100
- xpu:
  limit: 100
  vendor: NVIDIA | HW Ascend
  accelerator: T4| Atlas-300V-Pro
  capacities:
    streamVideoQuota: 40
    streamCaptureQuota: 20
cpu:
  limit: 4000m
memory:
  limit: 8000M
capacities:
- specification: 200W # 1920*1080
  # [required] 支持的视频分析路数, 默认0
  stream_video_quota: 30
  # [required] 支持的图片分析路数, 默认0

```

```

    stream_capture_quota: 10
    # [required] 支持的图片静态分析QPS, 默认0
    analyze_quota: 100
# [optional]可被动态挂载的文件包/模型包配置
modelSlots:T
# 配置的名称, 可自定义修改
- name: faceFeature
# 包的类型, 详见, 可以指定公网可以访问的 http perm link, 也可以指定算法包内的路径
type: http | local_path
# 包的描述信息, 可自由添加, 方便查询和筛选
tags:
  - device=nvidia
  - type=feature
# 文件包的定位符, 不同类型含义不同
resource: data/face_feature.tar
# 服务器算法对应容器中被挂载路径; 边端算法对应绝对物理路径
mount_path: /opt/models/face_feature

abilities:
  - abilityType: ObjectExtraction # 目标结构化类ObjectExtraction、事件检测类
EventExtraction、态势统计类StatusMonitor、其他类Uncategorized
  pipelineType: video # 视频流video、图片流capture、图片分析analyze
  eventCode: BAIDU-HUMAN_ATTR_ALERT-1.0.0 #算法封装包描述文件中应包含每个算法事件的
编码, 算法编码由用户自定义, 应唯一, 建议格式: provider-算法英文-version, 由大小写字母、下
划线、横线-、英文.组成, 长度小于128字符, 例如 BAIDU-HUMAN_ATTR_ALERT-1.0.0。
  spec:
    tag: face # 相同tag的eventCode可以被合并, 与eventMaxNum联动使用
    eventMaxNum: 3 # (相同tag下) 每个任务支持的最多事件类型数量
  input: # 此子算法支持的自定义输入参数
    path: "spec/assets/faceInput.json" # json schema, 描述中应指定默认值
  output: # 此子算法产出的消息的具体结构
    - subCode: PRE_ALERT
      path: "spec/assets/preOutput.json" # json schema
    - subCode: ALERT
      path: "spec/assets/output.json" # json schema

```

A.2 算法管理扩展信息文件 - YAML 样例

```

company:
  name: xxx
  code: 10001 # 填系统分配给公司的唯一编码, 第一次录入无需填写
  contactInfo: xxxx # 厂商联系方式
  afterSalesSupport: xxxx # 厂商售后服务网址
algoInfo:
  name:
    en: example-algo-engine # 可以跟 algo.yaml 中的 name 字段不同, 建议保持一致
    cn: 实例-目标结构化视频分析服务
  abilities:
    - name:
      en: face analyze

```

```
cn: 人脸检测
eventCode: face # 跟 algo.yaml 中 abilities 里的 event_type 一一对应
description:
  en: The face detection algorithm supports face detection, can quickly draw face
frames, and output face attributes.
  cn: 人脸检测算法支持人脸检测, 应能快速绘制人脸框, 并输出人脸属性。
categories: # 描述算法的分类, 便于分类筛选
- en: city safety
  cn: 安防
restrictions: # 算法适用范围约束说明文字, 比如适用场景、不适用场景等
- en: valid faces require min fazesize 20 px, pitch <= 90 degree; yaw <= 90 degree
  cn: 可检测目标范围约束为最小人脸为20像素, 俯仰角 pitch 不大于90度, 偏航角 yaw
不大于90度
scenes: # 算法适用的场所 (如城市交通、住宅建筑等) 说明文字
- en: city traffic
  cn: 城市交通
covers: # 封面轮播图, 此类文件应放到算法包中的 spec/assets 文件夹中
- spec/assets/face_1.png
- spec/assets/face_2.png
video: spec/assets/intro.mp4 # 介绍视频, 每个功能只支持一个
- name:
  en: pedestrian analyze
  cn: 人体检测
eventCode: ped # 跟 algo.yaml 中 abilities 里的 eventCode 一一对应
description:
  en: The ped detection algorithm supports ped detection, can quickly draw ped
frames, and output ped attributes.
  cn: 人体检测算法应能检测出人体, 并输出人体属性。
categories: # 描述算法的分类, 便于分类筛选
- en: city safety
  cn: 安防
restrictions: # 算法适用范围约束说明文字, 比如适用场景、不适用场景等
- en: valid person require min ped size 20 px, standing or squatting, not lying
down
  cn: 可检测目标范围约束为最小人体为20像素, 站立或者蹲坐状态, 非躺卧状态
scenes: # 算法适用的场所 (如城市交通、住宅建筑等)
- en: city traffic
  cn: 城市交通
covers: # 封面轮播图
- spec/assets/ped_1.png
- spec/assets/ped_2.png
video: spec/assets/intro.mp4 # 介绍视频, 每个功能只支持一个的规定
```

附录 B
(资料性)
算法引擎对象属性

B.1 表中选项字符说明

R=requirement 表示必选字段；0=Option 表示可选字段；R/0 表示特定条件下必选，其他可选。

B.2 基础数据结构

图像Point字段定义见表B.1。

表 B.1 图像 Point 字段定义

序号	名称	标志符	类型	长度	必选/可选	备注
1	X坐标	x	float32		R	X坐标数值
2	Y坐标	y	float32		R	Y坐标数值

图像关注区域Area字段定义见表B.2。

表 B.2 图像关注区域 Area 字段定义

序号	名称	标志符	类型	长度	必选/可选	备注
1	区域ID	areaId	string	20	R	区域唯一ID
2	区域类型	areaType	string	20	R	区域类型. <ul style="list-style-type: none"> ● POLYGON: 多边形 ● SINGLE_LINE: 单向单拌线 ● SINGLE_LINE_BOTH: 双向单拌线 ● MULTI_LINE: 单向多拌线 ● MULTI_LINE_BOTH: 双向多拌线
3	区域控制点	points	List<Point>	100	R	
4	触发方向	direction	int		0	触发方向: 0: 从左到右 1: 从右到左 2: 从上到下 3: 从下到上 4: 任意方向

图像遮蔽区域Mask字段定义见表B.3。

表 B.3 图像遮蔽区域 Mask 字段定义

序号	名称	标志符	类型	长度	必选/可选	备注
1	区域ID	maskId	string	20	R	区域唯一ID
2	区域类型	maskType	string	20	R	区域类型. <ul style="list-style-type: none"> ● POLYGON: 多边形 ● SINGLE_LINE: 单向单拌线 ● SINGLE_LINE_BOTH: 双向单拌线 ● MULTI_LINE: 单向多拌线 ● MULTI_LINE_BOTH: 双向多拌线
3	区域控制点	points	List<Point>	100	R	
4	触发方向	direction	int		0	触发方向: 0: 从左到右 1: 从右到左 2: 从上到下 3: 从下到上

序号	名称	标志符	类型	长度	必选/ 可选	备注
						4: 任意方向

B.3 算法引擎状态

算法引擎状态对象 EngineStatusResponse 字段定义见表B. 4。

表 B. 4 算法引擎状态对象 EngineStatusResponse 字段定义

序号	名称	标志符	类型	长度	必选/可选	备注
1	状态类型	status	string	20	R	<ul style="list-style-type: none"> ● INIT: 服务正在初始化过程中, 尚未可用 ● RUNNING: 服务正常运行, 可以承接解析任务 ● OVERLOAD: 服务过载, 无法被分配新的解析任务 ERROR: 服务异常, 无法承接解析任务
2	状态信息	message	string	100	0	更多详情描述信息, 比如错误原因等

备注: 算法引擎相关的配置信息, 通过独立的“算法技术描述信息文件”提供, 从而可以大幅简化算法状态查询接口的字段数量。

算法引擎错误信息对象 ErrorStatusResponse 字段定义见表B. 5。

表 B. 5 算法引擎错误信息对象 ErrorStatusResponse 字段定义

序号	名称	标志符	类型	长度	必选/ 可选	备注
1	状态码	code	int32	20	R	
2	状态信息	message	string	100	0	简要状态描述信息, 比如错误原因描述等
3	详细信息	details	List<string>	100	0	详细错误描述信息, 比如错误详情等

B.4 解析任务参数

创建解析任务对象 TaskInfoRequest 字段定义见表B. 6。

表 B. 6 创建解析任务对象 TaskInfoRequest 字段定义

序号	名称	标志符	类型	长度	必选/ 可选	备注
1	任务唯一ID	id	string	20	R	由平台自动生成, 用于唯一标识

序号	名称	标志符	类型	长度	必选/ 可选	备注
						识一个任务，算法服务在输出消息时应按照标准携带此字段的语义信息
2	任务名称	name	string	100	R	任务名称，自定义字符串
3	任务设备配置	streamOptions	StreamOptions		R	用于指定设备的类型和相关参数，支持图片流和视频流的设置。 详见附录B.5
4	任务规则配置	rulesParams	List<RuleParams>		R	任务配置参数，包含一个或这个多个事件类型对应规则的配置。 详见附录B.6

解析任务对象TaskInfoResponse字段定义见表B.7。

表 B.7 解析任务对象 TaskInfoResponse 字段定义

序号	名称	标志符	类型	长度	必选/ 可选	备注
1	任务唯一ID	id	string	20	R	由平台自动生成，用于唯一标识一个任务，算法服务在输出消息时应按照标准携带此字段的语义信息
2	任务状态	taskStatusStreamOptions	TaskStatus		R	任务运行时状态信息。

B.5 设备参数配置

设备参数配置 StreamOptions字段定义见表B.8。

表 B.8 设备参数配置 StreamOptions 字段定义

序号	名称	标志符	类型	长度	必选/ 可选	备注
1	设备唯一ID	deviceId	string	20	R	设备 ID
2	图片流参数配置	captureStreamOptions	captureStreamOptions		R/0	需处理的视频图像，任务创建时必选其一； - 图片流参数配置 - 视频流参数设置
3	视频流参数设置	videoStreamOptions	videoStreamOptions		视频流参数设置	

CaptureStreamOptions字段定义见表B.9。

表 B.9 CaptureStreamOptions 字段定义

序号	名称	标志符	类型	长度	必选/ 可选	备注
1	图片	subImageInfoList	array<string>		R	数据源地址，基于协议前缀适配类型
2	图片类型	imageType	string	20	0	指定合法的图片类型，不满足约束类型的图片将被自动丢弃 <ul style="list-style-type: none"> ● UNSPECIFIED - 未指定，可以接受各类图片类型 ● JPEG ● PNG ● RAW ● BMP
3	图片自小宽度	minWidth	int64		0	图片最小像素宽度约束，超出范围的图片将被自动丢弃
4	图片最大宽度	maxWidth	int64		0	图片最大像素宽度约束，超出范围的图片将被自动丢弃
5	图片最小高度	minHeight	int64		0	图片最小像素高度约束，超出范围的图片将被自动丢弃
6	图片最大高度	maxHeight	int64		0	图片最大像素高度约束，超出范围的图片将被自动丢弃

VideoStreamOptions字段定义见表B.10。

表 B.10 VideoStreamOptions 字段定义

序号	名称	标志符	类型	长度	必选/ 可选	备注
1	数据源地址	url	string	100	R	数据源地址，基于协议前缀适配类型 <ul style="list-style-type: none"> ● rtsp/rtmp 为视频流协议 ● http url 为离线视频文件协议
2	视频类型	playType	string	20	R	<ul style="list-style-type: none"> ● REALPLAY - 实时视频流 ● PLAYBACK - GB 历史录像流或离线文件 取流类型playback为历史流,realplay为实时流，默认是REALPLAY.

序号	名称	标志符	类型	长度	必选/ 可选	备注
3	历史流开始时间	startTime	string		R/O	历史流开始时间, datetime 当 play_type 为 playback 时, 必填。 注意:这个时间也会用作抓拍时间的起点.
4	历史流结束时间	endTime	string		R/O	历史流结束时间, datetime 当 play_type 为 playback 时, 必填。 结束时间必须大于开始时间。 对于离线视频文件而言, endTime如果指定, 可以通过 endTime-startTime 计算出所需要解析的离线视频时长 duration; 如果没有指定 endTime, 就会把这个离线视频文件分析完。
5	历史流倍速	playbackSpeed	int64		0	历史流倍速, 默认值为 1, 表示正常速度。 倍速分析是否被支持, 和多方面有关, 例如算法服务是否支持, 数据源和网络带宽是否支持, 因此, 倍速分析的分析时间不一定是准确的。

B.6 解析规则参数

解析规则参数RuleParams字段定义见表B. 11。

表 B. 11 解析规则参数 RuleParams 字段定义

序号	名称	标志符	类型	长度	必选/ 可选	备注
1	事件类型	eventCode	string	100	R	代表算法支持的一种功能, 具体内容应该是算法包提交的 algo.yaml 中的 abilities 字段中 eventCode 的一种.
2	规则ID	ruleId	string		R	用于区分相同 eventCode 的不同规则, 也用于唯一标识一个任务中的一个规则.
3	规则名称	ruleCustomName	string		0	自定义规则名称, 相比 ruleID 更语义化

序号	名称	标志符	类型	长度	必选/ 可选	备注
4	自定义透传标签	labels	JSONObject		0	用于业务透传信息,需在结果回调时携带,其他不做约束
5	检测区域	areas	List<Area>		0	需要关注的检测区域,仅此区域内的事件会被关注。
6	遮蔽区域	masks	List<Mask>		0	需要遮蔽的区域,此区域内的事件会别屏蔽。如果遮蔽区域和检测区域有重叠,以遮蔽区域的屏蔽规则优先。
7	处理结果上传地址	resultReceiveUrl	List<string>		R	任务处理结果上传地址相关的IP地址和端口数组,数组item为单个接收结果地址信息,形如 http://192.168.1.1:8001
8	算法自定义输入参数字段	extendParams	JSONObject		0	扩展算法自定义输入参数字段, schema由算法技术描述信息文件定义,详见附录 A.1

B.7 解析任务状态

解析任务状态对象 TaskStatus字段定义见表B.12。

表 B.12 解析任务状态对象 TaskStatus 字段定义

序号	名称	标志符	类型	长度	必选/ 可选	备注
1	状态类型	status	string	20	R	<ul style="list-style-type: none"> ● STARTING - 准备中 ● RUNNING - 运行中 ● DELETING - 正在删除 ● FINISHED - 正常结束 ● ERROR - 异常结束 任务状态,表示该任务在算法服务侧的状态。其中 FINISHED 和 ERROR 状态表示任务结束,资源释放。对于支持同步删除和资源释放的算法服务,不需要支持 DELETING 状态,其有效状态转换如

序号	名称	标志符	类型	长度	必选/ 可选	备注
						下:RUNNING -> FINISHED, RUNNING -> ERROR。 对于异步删除, 释放资源的算法服务, 应支持 DELETING 状态, 增加状态转换如下: RUNNING -> DELETING
2	状态异常信息	errorMessage	TaskErrorMsg		0	任务错误信息, 对于重试策略为 ALWAYS 的任务, 该字段表示最近一次发生的错误信息, 对于重试策略为 NEVER 的任务, 该字段表示任务错误信息。
3	进度百分比	percent	float		0	只针对离线文件
4	当前解析进度时间点	currentTime	string		0	针对录像解析反馈当前解析绝对时间

TaskErrorMsg字段定义见表B. 13。

表 B. 13 TaskErrorMsg 字段定义

序号	名称	标志符	类型	长度	必选/ 可选	备注
1	异常信息描述	message	string	100	R	异常描述信息
2	错误调用栈	codeStack	List<string>	100	0	程序栈内的错误信息, 按照错误发生的顺序排列。
3	错误发生时间点	occurredTime	string	100	R	错误发生的时间, datetime。

B. 8 解析任务列表

解析任务列表对象 ListTaskResponse字段定义见表B. 14。

表 B. 14 解析任务列表对象 ListTaskResponse 字段定义

序号	名称	标志符	类型	长度	必选/ 可选	备注
1	任务列表	tasks	List<TaskInfoResponse>		R	详见附录B. 4

B. 9 解析任务操作结果

解析任务删除成功对象DeleteTaskResponse字段定义见表B. 15。

表 B. 15 解析任务删除成功对象 DeleteTaskResponse 字段定义

序号	名称	标志符	类型	长度	必选/ 可选	备注
1	状态码	code	int32	20	R	
2	任务唯一ID	id	string	20	R	由平台自动生成，用于唯一标识一个任务，算法服务在输出消息时应按照标准携带此字段的语义信息
3	简要信息	message	string	100	0	简要描述信息，比如删除结果描述等
4	详细信息	details	List<string>	100	0	详细描述信息，比如删除任务相关详情等

B. 10 目标结构化类 - 解析输出结果

目标结构化类分析结果ObjectExtractionResult字段定义见表B. 16。

表 B. 16 目标结构化类分析结果 ObjectExtractionResult 字段定义

序号	名称	标志符	类型	长度	必选/ 可选	备注
1	任务ID	taskID	string	33	R	对应任务对象的TaskID
2	设备ID	deviceId	string	20	R	结果来源的设备id
3	视频帧ID	frameId	string		0	结果来源的视频帧id
4	全景图Base64	backgroundImage	string		0	结果对应全景图的Base64 image file
5	全景图存储地址	backgroundImageUrl	string		0	结果对应全景图的存储地址
6	视频帧PTS	pts	string		0	视频帧对应的PTS，单位毫秒（ms），转为JSON后类型为string
7	接收到视频帧的时间戳	recvTs	string		0	算法服务收到视频帧的时间戳，单位毫秒（ms），转为JSON后类型为string
8	自定义透传标签	labels	JSONObject		0	用于业务透传信息，来源于B. 4的解析任务入参，需在结果回调时携带，其他不做约束

序号	名称	标志符	类型	长度	必选/ 可选	备注
9	抓拍目标分析 结果列表	captureObjects	List<CaptureObject>		R	CaptureObject 为算法自定义输出 参数字段， schema由算法技术 描述信息文件 定义，详见附录 A.1
10	事件编码	eventCode	string		R	事件编码

B.11 事件检测类 - 解析输出结果

事件检测类分析结果 EventExtractionResult字段定义见表B.17。

表 B.17 事件检测类分析结果 EventExtractionResult 字段定义

序号	名称	标志符	类型	长度	必选/ 可选	备注
1	任务ID	taskID	string	33	R	对应任务对象的TaskID
2	设备ID	deviceId	string	20	R	结果来源的设备 id
3	全景图 Base64	backgroundImage	string		0	结果对应全景图的Base64 image file
4	全景图存储 地址	backgroundImageUrl	string		0	结果对应全景图的存储地址
5	视频地址	backgroundVideoUrl	string		0	结果对应短视频的存储地址
6	事件开始绝 对时间	eventBeginTime	string		R/0	在线视频时必选 格式为ISO 8601 datetime，例 如 "2018-12-10T13:45:00.000Z"
7	事件开始相 对时间	eventBeginTimeRlt	float32		R/0	离线视频图像时必选 单位：秒
8	事件开始相 对帧号	eventBeginFrameNoRlt	int64		R/0	事件开始时的视频帧相对于起 始帧的序号，离线视频分析时 必选。
9	事件结束绝 对时间	eventEndTime	string		R/0	在线视频时必选 格式为ISO 8601 datetime，例 如 "2018-12-10T13:45:00.000Z"
10	事件结束相 对时间	eventEndTimeRlt	float32		R/0	离线视频图像时必选 单位：秒
11	事件结束相 对帧号	eventEndFrameNoRlt	int64		R/0	事件开始时的视频帧相对于起 始帧的序号，离线视频分析时 必选。
12	自定义透传 标签	labels	JSONObject		R	用于业务透传信息，来源于B.4 的解析任务入参，需在结果回 调时携带，其他不做约束
13	报警事件结 果列表	alarmEvents	List<AlarmEvent>		R	AlarmEvent为算法自定义输出 参数字段，schema由算法技术 描述信息文件定义，详见附录 A.1

B.12 态势统计类 - 解析输出结果

态势统计类分析结果 StatusMonitorResult字段定义见表B.18。

表 B.18 态势统计类分析结果 StatusMonitorResult 字段定义

序号	名称	标志符	类型	长度	必选/可选	备注
1	任务ID	taskID	string	33	R	对应任务对象的TaskID
2	设备ID	deviceId	string		R	结果来源的设备id
3	视频帧ID	frameId	string		0	结果来源的视频帧 id
4	全景图Base64	bgImage	string		0	结果对应全景图的Base64 image file
5	全景图存储地址	backgroundImageUrl	string		0	结果对应全景图的存储地址
6	视频帧PTS	pts	string		0	视频帧对应的 PTS, 单位毫秒(ms), 转为 JSON 后类型为 string
7	接收到视频帧的时间戳	recvTs	string		0	算法服务收到视频帧的时间戳, 单位毫秒(ms), 转为 JSON 后类型为 string
8	自定义透传标签	labels	JSONObject		R	用于业务透传信息, 来源于B.4的解析任务入参, 需在结果回调时携带, 其他不做约束
9	态势统计开始绝对时间	StartTimeAbs	string		R/0	对在线视频为绝对时间, 必选 格式为ISO 8601 datetime, 例如 "2018-12-10T13:45:00.000Z"
10	统计结束绝对时间	EndTimeAbs	string		R/0	对在线视频为绝对时间, 必选 格式为ISO 8601 datetime, 例如 "2018-12-10T13:45:00.000Z"
11	统计开始相对时间	StartTimeRlt	double		R/0	对离线视频是相对时间, 必选 单位: 秒
12	统计结束相对时间	EndTimeRlt	double		R/0	对离线视频是相对时间, 必选 单位: 秒
13	态势统计结果	statusInfo	StatusInfo		R	StatusInfo为算法自定义输出参数字段, schema由算法技术描述信息文件定义, 详见附录A.1

B.13 其他类 - 解析输出结果

其他类分析结果 UncategorizedResult 字段定义见表B.19。

表 B.19 其他类分析结果 UncategorizedResult 字段定义

序号	名称	标志符	类型	长度	必选/可选	备注
1	任务ID	taskID	string	33	R	对应任务对象的TaskID
2	设备ID	deviceId	string	22	R	结果来源的设备id
3	视频帧ID	frameId	string		0	结果来源的视频帧 id
4	视频帧PTS	pts	string		0	视频帧对应的 PTS, 单位毫秒(ms), 转为

序号	名称	标志符	类型	长度	必选/可选	备注
						JSON 后类型为 string
5	接收到视频帧的时间戳	recvTs	string		0	算法服务收到视频帧的时间戳，单位毫秒（ms），转为 JSON 后类型为 string
6	自定义透传标签	labels	JSONObject		R	用于业务透传信息，来源于B.4的解析任务入参，需在结果回调时携带，其他不做约束

附录 C (资料性) 算法引擎对象样例数据

C.1 算法引擎状态

基于附录B. 3, 算法引擎错误信息对象 `ErrorStatusResponse` 样例数据

```
{
  "code": 0,
  "message": "引擎异常",
  "details": [
    "授权服务通信异常",
    "参数异常, 启动参数不合法"
  ]
}
```

C.2 解析任务

基于附录B. 4, 解析任务对象请求 `TaskInfoRequest` 样例数据

```
{
  "id": "eckfw-343das",
  "name": "城市文明管理任务",
  "streamOptions": StreamOptions,
  "rulesParams": [ RuleParams ],
}
```

解析任务对象请求响应 `TaskInfoResponse` 样例数据

```
{
  "id": "eckfw-343das",
  "name": "城市文明管理任务",
  "streamOptions": StreamOptions,
  "rulesParams": [ RuleParams ],
  "taskStatus": TaskStatus,
}
```

C.3 设备配置

基于附录B. 5, 设备参数配置 `StreamOptions` 样例数据

```
{
  "captureStreamOptions": {
    "url": "iot://host:port",
    "imageType": "UNSPECIFIED",
    "minWidth": 50,
    "maxWidth": 50,
    "minHeight": 2000,
    "maxHeight": 3000,
  },
  "videoStreamOptions": {
    "url": "rtsp://host:port",
    "playTpe": "PLAYBACK",
  }
}
```

```

    "startTime": "2023-03-14T08:22:08.478Z",
    "endTime": "2023-03-14T012:22:08.478Z",
    "playbackSpeed": 1
  }
}

```

C.4 解析规则参数

基于附录B.6，解析规则参数 RuleParams 样例数据

```

{
  "eventCode": "OVERFLOWED_GARBAGE",
  "ruleId": 0,
  "ruleCustomName": "",
  "labels": {
    "WOT-13088922786726912-0fM60Pzp": "{ \"address\": \" 垃圾 满 溢
\", \"algoId\": \"0f8deef9-366b-4caf-8d5d-2935d96f46f0\", \"crossNetwork\": \"false\" }"
  },
  "areas": [
    {
      "areaId": 0,
      "areaType": "POLYGON",
      "points": [
        {
          "x": 0,
          "y": 1
        },
        {
          "x": 1,
          "y": 1
        },
        {
          "x": 1,
          "y": 0
        },
        {
          "x": 0,
          "y": 0
        }
      ]
    }
  ],
  "masks": [],
  "resultReceiveUrl": [
    "http://10.172.198.67:23310",
    "http://192.168.1.1:8001"
  ],
  "extendParams": {
    "durationThreshold": 1,
    "repeatAlarmIntervalThreshold": 60,
    "sensitivity": "低"
  }
}

```

```

    }
  }

```

C.5 解析任务状态

基于附录B.7，解析任务状态对象 TaskStatus 样例数据

```

{
  "status": "ERROR",
  "errorMessage": {
    "message": "程序崩溃异常",
    "codeStack": [
      "*errors.errorString this function is supposed to crash
/0/go/src/github.com/go-errors/errors/crashy/crashy.go:8 (0x41280)",
      "    Crash: return errors.New(Crashed)",
      "/0/go/src/github.com/go-errors/errors/example/main.go:11 (0x2026)",
      "    main: err := crashy.Crash()"
    ],
    "occurredTime": "2023-01-28T03:26:27.747Z"
  }
}

```

C.6 解析任务列表

基于附录B.8，解析任务列表对象 ListTaskResponse 样例数据

```

{
  "tasks": [ TaskInfoResponse ]
}

```

C.7 解析任务操作结果

基于附录B.9，解析任务删除成功对象 DeleteTaskResponse 样例数据

```

{
  "code": 0,
  "id": "eckfw-343das",
  "message": "任务删除成功完成",
  "details": [
    "foo",
    "bar"
  ]
}

```

附录 D
(资料性)
算法包接口 C 语言描述

D.1 函数命名规则

算法包接口函数命名由智能分析业务英文缩写、函数类型和函数功能组成，用下划线连接，具体规则见表C.1，算法包接口函数C语言示例化命名见表C.2。

表 D.1 函数命名规则定义表

序号	字段	名称示例	说明
1	智能分析业务英文缩写	VICA	Video Image Content Analysis (视频图像内容分析) 缩写
		VIFC	Video Image Feature Comparison (视频图像特征比对) 缩写
2	函数类型	Plugin	算法包提供的插件函数(首字母大写)
		CALLBACK	平台提供的回调函数(全大写)
3	函数类型	Initialize	算法包提供的插件函数(首字母大写)
		SetLogFunction	函数功能英文描述,采用大驼峰命名格式,首字母大写,无连接符号

表 D.2 算法包接口函数命名表

序号	函数功能	C语言函数名	函数定义
1	设置算法日志函数	VICA_Plugin_SetLogFunction	参见C.5
2	设置算法授权信息函数	VICA_Plugin_SetLicense	参见C.6
3	检测识别算法初始化函数	VICA_Plugin_Initialize	参见C.7
4	创建分析通道函数	VICA_Plugin_Create	参见C.8
5	调用分析算法函数	VICA_Plugin_AnalyzeStream	参见C.9
6	关闭分析通道函数	VICA_Plugin_Close	参见C.10
7	释放算法函数	VICA_Plugin_Release	参见C.11
8	分析结果回调函数	VICA_CALLBACK_AnalyzeResult	参见C.12
9	获取系统信息回调函数	VICA_CALLBACK_GetSystemInfo	参见C.13
10	获取目标ID回调函数	VICA_CALLBACK_GetObjectID	参见C.14
11	单帧同步分析函数	VICA_Plugin_FrameSynchronizeAnalyze	参见C.15
12	设置检测识别回调接口函数	VICA_Plugin_SetCallbackFunction	参见C.16

D.2 数据类型定义

数据类型格式定义见表C.3。

表 D.3 数据类型定义表

序号	名称	格式定义	说明
1	VICA_VA_CONTEX_T	pfnAnalyzeCallback	算法分析结果回调函数,可选,默认对所有通道生效

序号	名称	格式定义	说明	
		pfnGetSystemInfoCallback	获取当前插件目录回调函数	
		pfnGetObjectIdCallback	获取目标ID回调函数	
		iXpuCardNo	XPU卡号, -1为CPU	
		reserved[256]	保留字节, 全部填写0	
2	VICA_IMAGE_INFO_S	uiImageFormat	图片格式, 参见 VICA_IMAGE_FORMAT_E	
		uiRowSize	图片每行的数据长度	
		uiWidth	图片宽度	
		uiHeight	图片高度	
		uiBufferLen	图片buffer长度	
		uiTimestamp	时间戳(ms)	
		uiPosNum	目标位置数量	
		uiFrameNo	视频帧序号	
		uiFrameRate	视频帧率	
		stObjectPos	uiType	位置类型, 参见 VICA_POS_TYPE_E
			uiLTX	左上角X坐标
			uiLTY	左上角Y坐标
uiHeight	高			
uiBufferAddr	图片地址类型, 参见 VICA_BUFFER_ADDR_E			
pBufferPtr	图片buffer地址			
3	VICA_KEY_IMAGE_INFO_S	uiImageFormat	图片格式, 参见 VICA_IMAGE_FORMAT_E	
		uiRowSize	图片每行的数据长度	
		uiWidth	图片宽度	
		uiHeight	图片高度	
		uiTimestamp	时间戳(ms)	
		uiBufferAddr	图片地址类型, 参见 VICA_BUFFER_ADDR_E	
		uiBufferLen	图片buffer长度	
		pBufferPtr	图片buffer地址	
		uiObjectParamNum	目标个数	
pstObjectParam	目标参数指针, 数据类型 VICA_OBJECT_PARAM_INFO			
4	VICA_TRACE_S	uiTraceLength	轨迹长度(下面 pstTrackList数组 的长度, 表示有多少个跟踪方框)	
		pstTrackList	ullTimestamp	时间戳
			uiType	位置类型, 参见 VICA_POS_TYPE_E:
			uiLTX	左上角X坐标
			uiLTY	左上角Y坐标
			uiWidth	宽
uiHeight	高			
5	VIFC_CONTEXT	pfnGetSystemInfoCallback	获取当前插件目录回调函数	
		pfnGetObjectIdCallback	获取目标ID回调接口回调函数	
		iXpuCardNo	XPU卡号, -1为CPU	
		reserved[512]	保留字节, 全部填写0	

序号	名称	格式定义		说明
6	VICA_OBJECT_PARAM_INFO	uiObjectType		目标类型： 1- 机动车； 2- 非机动车； 3- 人脸； 4- 人体； 5- 骑行者
		uiPosNum		位置个数
		stObjectPos	uiType	位置类型，参见 VICA_POS_TYPE_E
			uiLTX	左上角X坐标
			uiLTY	左上角Y坐标
			uiWidth	宽
uiHeight	高			
szReserve[180]		保留字段，请置空		

D.3 数据格式定义

数据格式定义见表C.4。

表 D.4 数据格式定义表

序号	数据格式	数据名称	数据取值
1	VICA_LOG_LEVEL_E	日志级别	0-FATAL（致命）； 1- ERROR（错误）； 2- WARN（告警）； 3- INFO（信息）； 4- DEBUG（调试信息）
2	VICA_IMAGE_FORMAT_E	图片格式	0- BGR24_PACKED； 1- RGB24_PACKED； 2- BGR24_PLANAR； 3- RGB24_PLANAR； 4- NV12； 5- JPEG； 6- H. 264 裸码流； 7- H. 265裸码流
3	VICA_POS_TYPE_E	位置类型	0-车身位置； 1- 车牌位置； 2- 目标整体位置； 3- 目标位置； 4- 其他
4	VICA_BUFFER_ADDR_E	地址类型	0-CPU 地址； 1-XPU地址；
5	VICA_SYS_INFO_TYPE_E	系统信息类型	0-算法路径； 1- License 信息
6	VICA_CALLBACK_TYPE_E	回调函数类型	0-分析结果回调函数； 1-分析状态回调函数；
7	VIFC_MATCH_TYPE_E	特征匹配类型	0-长特征 1- 短特征 2- 长特征之间的实际距离 3- 两个长特征类之间的相似度 4- 两个长特征类之间的实际距离 0xFFFFFFFF-非法枚举值

D.4 函数返回值定义

函数返回值定义见表C.5。

表 D.5 函数返回值定义表

序号	错误码取值	错误码含义
1	0	调用成功
2	-101	未知错误
3	-102	致命错误，应重启算法包应用进程以恢复整体环境
4	-103	分析的图像分辨率不支持
5	-104	License Server 连接异常
6	-105	超过License最大授权数量
7	-106	License 过期
8	-201	输入图像中未检测到目标
9	-202	算法提取目标特征向量失败

D.5 设置算法日志函数

函数原型：`int32_t VICA_Plugin_SetLogFunction(uint32_t uiLevel, VICA_CALLBACK_Log pfnCallback);`
 函数调用参数见表C.6。

表 D.6 VICA_Plugin_SetLogFunction 函数参数列表

参数	数据类型	参数类型	参数说明
uiLevel	uint32_t	输入参数	日志级别，表示只会输出高于该日志级别的日志，数值越小级别越高，取值见VICA_LOG_LEVEL E
pfnCallback	VICA_CALLBACK_Log	输入参数	日志记录回调函数，多算法应用系统定义日志记录回调函数，算法包调用该函数记录日志

日志记录回调函数原型：`typedef void(*VICA_CALLBACK_Log)(uint32_t uilevel, char *pstFormat, va_list argList)`
 函数调用参数见表C.7

表 D.7 VICA_CALLBACK_Log 函数参数列表

参数	数据类型	参数类型	参数说明
uiLevel	uint32_t	输入参数	日志级别，表示只会输出高于该日志级别的日志，数值越小级别越高，取值见VICA_LOG_LEVEL E
pfnCallback	VICA_CALLBACK_Log	输入参数	日志记录回调函数，多算法应用系统定义日志记录回调函数，算法包调用该函数记录日志

D.6 设置算法授权信息函数

函数原型：`int32_t VICA_Plugin_SetLicense(char *pLicenseInfoPtr, uint32_t uiLength);` 函数调用参数见表C.8。

表 D.8 VICA_Plugin_SetLicense 函数参数列表

参数	数据类型	参数类型	参数说明
----	------	------	------

参数	数据类型	参数类型	参数说明
pLicenseInfoPtr	char *	输入参数	授权服务器的IP和端口，或授权服务器域名，算法包需要授权时去该服务器申请
uiLength	uint32_t	输入参数	授权信息长度

D.7 检测识别算法初始化函数

函数原型：`int32_t VICA_Plugin_Initialize(char *pPluginConfPtr, VICA_VA_CONTEXT *pstContext);`
 函数调用参数见表D.9。

表 D.9 VICA_Plugin_Initialize 函数参数列表

参数	数据类型	参数类型	参数说明
pPluginConfPtr	char *	输入参数	算法插件初始化需要用到的xml
pstContext	VICA_VA_CONTEXT *	输入参数	设置算法分析结果回调函数，获取当前算法目录，获取目标ID回调函数，XPU卡号等。

D.8 创建分析通道函数

函数原型：`int32_t VICA_Plugin_Create(char *pChannConfPtr);`
 函数调用参数见表C.10。

表 D.10 VICA_Plugin_Create 函数参数列表

参数	数据类型	参数类型	参数说明
pChannelConfPtr	char *	输入参数	分析通道创建的xml

D.9 调用分析算法函数

函数原型：`int32_t VICA_Plugin_AnalyzeStream(int32_t uiChannel, VICA_IMAGE_INFO_S *pstImageInfo, void *pPrivateDataPtr)`

函数调用参数见表C.11。

表 D.11 VICA_Plugin_AnalyzeStream 函数参数列表

参数	数据类型	参数类型	参数说明
uiChannel	int32_t	输入参数	分析通道创建时的通道号
pstImageInfo	VICA_IMAGE_INFO_S *	输入参数	图片信息，包括视频帧率、帧号、图片格式、尺寸、时间戳、地址类型、目标个数等
pPrivateDataPtr	void *	输入参数	多算法应用系统私有数据，算法内部无需处理。

D.10 关闭分析通道函数

函数原型：`int32_t VICA_Plugin_Close(int32_t uiChannel);`
 函数调用参数见表C.12。

表 D.12 VICA_Plugin_Close 函数参数列表

参数	数据类型	参数类型	参数说明
uiChannel	int32_t	输入参数	分析通道创建时的通道号

D.11 释放算法函数

函数原型：void VICA_Plugin_Release(); 本函数无调用参数。

D.12 分析结果回调函数

函数原型：typedef void(*VICA_CALLBACK_AnalyzeResult)(uint32_t uiChannel, uint64_t uiObjectId, unsigned char *pDataPtr, uint32_t uiLength, VICA_KEY_IMAGE_INFO_S *pstImageBuf, VICA_TRACE_S *pstTrace, void *pPrivateDataPtr);

函数调用参数见表C.13。

表 D.13 VICA_CALLBACK_AnalyzeResult 函数参数列表

参数	数据类型	参数类型	参数说明
uiChannel	uint32_t	输入参数	分析通道创建时的通道号
uiObjectId	uint64_t	输入参数	目标ID
pDataPtr	unsigned char*	输入参数	分析结果元数据, protobuf结构
uiLength	uint32_t	输入参数	元数据信息长度
pstImageBuf	VICA_KEY_IMAGE_INFO_S *	输入参数	关键帧图片信息, 包括图片格式、尺寸、时间戳、地址类型、目标个数等。参数格式说明见表C.3
pstTrace	VICA_TRACE_S *	输入参数	轨迹信息, 包括轨迹长度, 轨迹方框等。参数格式说明见表C.3
pPrivateDataPtr	void *	输入参数	调用分析函数时传入的私有数据

D.13 获取系统信息回调函数

函数原型：typedef int32_t(*VICA_CALLBACK_GetSystemInfo)(uint32_t uiInfoType, unsigned char *pBufferPtr, uint32_t uiBufferLength);

函数调用参数见表C.14。

表 D.14 VICA_CALLBACK_GetSystemInfo 函数参数列表

参数	数据类型	参数类型	参数说明
uiInfoType	uint32_t	输入参数	系统信息类型。参数格式说明 VICA_SYS_INFO_TYPE E
pBufferPtr	unsigned char *	输出参数	系统信息内容buffer
uiBufferLength	uint32_t	输出参数	系统信息buffer长度

D.14 获取目标 ID 回调函数

函数原型：typedef uint64_t(*VICA_CALLBACK_GetObjectID)(uint32_t uiChannel, uint64_t uiTimestamp, uint32_t uiObjectType);

函数调用参数见表C.15。

表 D.15 VICA_CALLBACK_GetObjectID 函数参数列表

参数	数据类型	参数类型	参数说明
uiChannel	uint32_t	输入参数	分析通道创建时的通道号
uiTimestamp	uint64_t	输入参数	目标生成时的时间戳
uiObjectType	uint32_t	输入参数	目标类型

D.15 单帧同步分析函数

函数原型：`int32_t VICA_Plugin_FrameSyschronizeAnalyze(uint32_t uiChannel, VICA_IMAGE_INFO_S *pstImageInfo, void *pPrivateDataPtr, uint64_t uiObjectId, unsigned char *pDataPtr, uint32_t uiLength);`

函数调用参数见表C. 16。

表 D. 16 VICA_Plugin_FrameSyschronizeAnalyze 函数参数列表

参数	数据类型	参数类型	参数说明
uiChannel	uint32 t	输入参数	分析通道创建时的通道号
pstImageInfo	VICA_IMAGE_INFO_S *	输入参数	图片信息，包括图片格式、尺寸、时间戳、地址类型、目标个数等。
pPrivateDataPtr	void *	输入参数	多算法应用系统私有数据，算法内部无需处理。
uiObjectId	uint64 t	输入参数	目标ID
pDataPtr	unsigned char*	输入参数	分析结果元数据，protobuf结构
uiLength	uint32 t	输入参数	元数据信息长度

D. 16 设置检测识别回调接口函数

函数原型：`int32_t VICA_Plugin_SetCallbackFunction(uint32_t uiChannel, uint32_t uiFunctionType, void *pFunction, void *pPrivateDataPtr);`

函数调用参数见表C. 17。

表 D. 17 VICA_Plugin_SetCallbackFunction 函数参数列表

参数	数据类型	参数类型	参数说明
uiChannel	uint32 t	输入参数	分析通道创建时返回的通道号
uiFunctionType	uint32 t	输入参数	标识要设置的回调函数是何种功能的函数
pFunction	void *	输入参数	回调函数指针
pPrivateDataPtr	void *	输入参数	多算法应用系统私有数据，算法内部无需处理，保持透传。

附 录 E
(规范性)
算法输入输出参数 JSON Schema

E.1 算法输入参数 - JSON Schema 样例

```

{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "required": [
    "rulesParams",
    "streamOptions",
    "id",
    "name"
  ],
  "properties": {
    "rulesParams": {
      "items": {
        "required": [
          "eventType",
          "ruleId",
          "areas",
          "extendParams"
        ],
        "properties": {
          "eventType": {
            "type": "string",
            "description": "事件类型 代表算法支持的一种功能，具体内容应该是算法包提交的 algo.yaml 中的 abilities 字段中 eventType 的一种。",
            "title": "eventType",
            "enum": [
              "FIRE_EQUIPMENT"
            ],
            "default": "FIRE_EQUIPMENT"
          },
          "ruleId": {
            "type": "integer",
            "description": "规则 ID 用于区分相同 eventType 的不同规则，也用于唯一标识一个任务中的一个规格，常用于任务合并的计算，默认可以不填，由 AI-PaaS Gateway 生成。",
            "title": "ruleId"
          },
          "ruleCustomName": {
            "type": "string",
            "description": "自定义规则名称，相比 ruleID 更语义化",
            "title": "ruleCustomName"
          },
          "areas": {
            "items": {

```

```

"required": [
  "areaId",
  "areaType"
],
"properties": {
  "areaId": {
    "type": "integer",
    "title": "areaId"
  },
  "areaType": {
    "enum": [
      "POLYGON"
    ],
    "oneOf": [
      {
        "description": "多边形",
        "const": "POLYGON"
      },
      {
        "description": "单向单拌线",
        "const": "SINGLE_LINE"
      },
      {
        "description": "双向单拌线",
        "const": "SINGLE_LINE_BOTH"
      },
      {
        "description": "单向多拌线",
        "const": "MULTI_LINE"
      },
      {
        "description": "双向多拌线",
        "const": "MULTI_LINE_BOTH"
      }
    ],
    "title": "Area Type",
    "description": "区域类型.",
    "type": "string"
  },
  "points": {
    "items": {
      "required": [
        "x",
        "y"
      ],
      "properties": {
        "x": {
          "type": "number",
          "title": "x"
        }
      }
    }
  }
}

```

```

    },
    "y": {
      "type": "number",
      "title": "y"
    }
  },
  "additionalProperties": true,
  "type": "object",
  "title": "Point"
},
"type": "array",
"description": "**关于有向拌线方向及产品报警定义的说明：** 1. 有向拌
线可以为多段折线，按照起点-终点的顺序进行绘制 2. 拌线的箭头方向为第一段线（可以绘制折线）
逆时针旋转90度所指的方向 3. 如无特殊说明，所有功能的报警都是顺着箭头方向通过则报警 注意：
当 area_type 为 'POLYGON'/'SINGLE_LINE'/'SINGLE_LINE_BOTH' 时，使用 'points' 当 area_type 为
'MULTI_LINE'/'MULTI_LINE_BOTH' 时，需要使用 multi_points.",
"title": "points",
"minItems": 2,
"maxItems": 20
},
"multiPoints": {
  "items": {
    "required": [
      "points"
    ],
    "properties": {
      "points": {
        "items": {
          "required": [
            "x",
            "y"
          ],
          "properties": {
            "x": {
              "type": "number",
              "title": "x"
            },
            "y": {
              "type": "number",
              "title": "y"
            }
          }
        },
        "additionalProperties": true,
        "type": "object",
        "title": "Point"
      }
    },
    "type": "array",
    "title": "points",
    "minItems": 2,

```

```

        "maxItems": 20
      }
    },
    "additionalProperties": true,
    "type": "object",
    "title": "Points"
  },
  "type": "array",
  "title": "multiPoints",
  "minItems": 1,
  "maxItems": 10
}
},
"additionalProperties": true,
"type": "object",
"title": "Area",
"default": {
  "areaId": 1,
  "areaType": "POLYGON",
  "points": [
    {
      "x": 0,
      "y": 0
    },
    {
      "x": 1,
      "y": 0
    },
    {
      "x": 1,
      "y": 1
    },
    {
      "x": 0,
      "y": 1
    }
  ]
},
"if": {
  "properties": {
    "areaType": {
      "enum": [
        "POLYGON",
        "SINGLE_LINE",
        "SINGLE_LINE_BOTH"
      ]
    }
  }
}
},
},

```

```

    "then": {
      "allOf": [
        {
          "required": [
            "points"
          ]
        },
        {
          "not": {
            "required": [
              "multiPoints"
            ]
          }
        }
      ]
    },
    "else": {
      "allOf": [
        {
          "required": [
            "multiPoints"
          ]
        },
        {
          "not": {
            "required": [
              "points"
            ]
          }
        }
      ]
    }
  },
  "type": "array",
  "description": "检测区域",
  "title": "areas",
  "minItems": 1,
  "maxItems": 20
},
"masks": {
  "items": {
    "required": [
      "maskId",
      "maskType",
      "points"
    ],
    "properties": {
      "maskId": {
        "type": "integer",

```

```

    "title": "maskId"
  },
  "maskType": {
    "enum": [
      "POLYGON"
    ],
    "oneOf": [
      {
        "description": "多边形",
        "const": "POLYGON"
      }
    ],
    "title": "Mask Type",
    "description": "屏蔽区域类型。",
    "type": "string"
  },
  "points": {
    "items": {
      "required": [
        "x",
        "y"
      ],
      "properties": {
        "x": {
          "type": "number",
          "title": "x"
        },
        "y": {
          "type": "number",
          "title": "y"
        }
      }
    },
    "additionalProperties": true,
    "type": "object",
    "title": "Point"
  },
  "type": "array",
  "title": "points",
  "minItems": 3,
  "maxItems": 20
}
},
"additionalProperties": true,
"type": "object",
"title": "Mask"
},
"type": "array",
"description": "遮蔽区域",
"title": "masks",

```

```

    "minItems": 0,
    "maxItems": 20
  },
  "extendParams": {
    "additionalProperties": true,
    "type": "object",
    "title": "extendParams",
    "required": [
      "targetMin",
      "targetMax",
      "duration",
      "cooldownDuration",
      "targetTypes",
      "custom"
    ],
    "properties": {
      "targetMin": {
        "type": "number",
        "description": "目标大小过滤逻辑阈值, 0.00 <= target_min < target_max <=
1.00 目标大小过滤逻辑阈值, 目标边长占图片边长最小比例",
        "title": "targetMin",
        "default": 0,
        "minimum": 0,
        "maximum": 1
      },
      "targetMax": {
        "type": "number",
        "description": "目标大小过滤逻辑阈值, 目标边长占图片的最大比例",
        "title": "targetMax",
        "default": 1,
        "minimum": 0,
        "maximum": 1
      },
      "duration": {
        "type": "number",
        "description": "延迟报警(单位: 秒), 报警目标需至少维持报警状态该时长
后才报警",
        "title": "duration",
        "default": 1,
        "minimum": 0,
        "maximum": 1800
      },
      "cooldownDuration": {
        "type": "number",
        "description": "报警间隔时长(单位: 秒)",
        "title": "cooldownDuration",
        "default": 600,
        "minimum": 0,
        "maximum": 86400
      }
    }
  }
}

```

```

    },
    "targetTypes": {
      "items": {
        "type": "string",
        "enum": [
          "FIRE_EQUIPMENT"
        ]
      },
      "type": "array",
      "description": "list 类型，元素由 AlgoCabinObjType.ObjType 定义 目标
类型列表",
      "title": "targetTypes"
    },
    "level": {
      "enum": [
        "ALARM_LEVEL"
      ],
      "oneOf": [
        {
          "description": "报警层",
          "const": "ALARM_LEVEL"
        },
        {
          "description": "事件层",
          "const": "EVENT_LEVEL"
        },
        {
          "description": "模型层",
          "const": "MODEL_LEVEL"
        }
      ],
      "title": "Level",
      "description": "布控的报警层级。",
      "type": "string",
      "default": "ALARM_LEVEL"
    },
    "custom": {
      "additionalProperties": true,
      "type": "object",
      "description": "算法仓自定义布控参数，可用于指定
AlgoCabinExtendParams.custom 参数。",
      "title": "custom",
      "properties": {
        "fireEquipment": {
          "required": [
            "leastObjCount"
          ],
          "properties": {
            "leastObjCount": {

```

```

        "type": "integer",
        "description": "画面中至少出现的消防设施个数",
        "title": "leastObjCount"
    }
},
"additionalProperties": true,
"type": "object",
"title": "fireEquipment",
"description": "物品及物品属性报警（2XXX） 消防设施"
}
},
"oneOf": [
    {
        "required": [
            "fireEquipment"
        ]
    }
]
},
"threshold": {
    "additionalProperties": {
        "type": "number"
    },
    "type": "object",
    "description": "阈值设置的接口，用于传入阈值",
    "title": "threshold",
    "properties": {
        "FIRE_EQUIPMENT": {
            "type": "number",
            "title": "FIRE_EQUIPMENT",
            "default": 0.35,
            "minimum": 0,
            "maximum": 1
        }
    }
}
},
"description": "算法仓扩展参数，标准由 R 和业务制定."
},
"labels": {
    "type": "object",
    "description": "用于平台透传信息（比如gb id），算法仓应将此字段中的内容透传到输出消息的指定字段中",
    "title": "labels"
}
},
"additionalProperties": true,
"type": "object",
"title": "Rule Params",

```

```

    "description": "布控规则参数。",
  },
  "type": "array",
  "description": "任务配置参数, 包含一个或这个多个事件类型对应规则的配置。",
  "title": "rulesParams"
},
"streamOptions": {
  "properties": {
    "captureStreamOptions": {
      "required": [
        "url",
        "imageType",
        "minWidth",
        "maxWidth",
        "minHeight",
        "maxHeight",
        "deprecatedQps",
        "qps"
      ],
    },
    "properties": {
      "url": {
        "type": "string"
      },
    },
    "imageType": {
      "enum": [
        "UNSPECIFIED",
        "JPEG",
        "PNG",
        "RAW",
        "BMP"
      ],
      "type": "string",
      "title": "Image Type",
      "description": "图片类型在开放平台上需要准确定义"
    },
    "minWidth": {
      "type": "integer"
    },
    "maxWidth": {
      "type": "integer"
    },
    "minHeight": {
      "type": "integer"
    },
    "maxHeight": {
      "type": "integer"
    },
    "deprecatedQps": {
      "type": "integer"
    }
  }
}

```

```

    },
    "qps": {
      "type": "number",
      "description": "抓拍流对应的 qps, 最小值是 0.001 为了 ai-paas gateway 计
算任务所需 quota."
    }
  },
  "additionalProperties": true,
  "type": "object",
  "title": "Capture Stream Options",
  "description": "图片流参数设置."
},
"videoStreamOptions": {
  "required": [
    "url",
    "resolution",
    "playType",
    "startTime",
    "endTime",
    "playbackSpeed"
  ],
  "properties": {
    "url": {
      "type": "string",
      "description": "视频地址"
    },
    "resolution": {
      "enum": [
        "UNSPECIFIED",
        "PIXELS_200W",
        "PIXELS_400W",
        "PIXELS_800W"
      ],
      "type": "string",
      "title": "Resolution",
      "description": "视频流分辨率 用于 Quota 计算, 用户需要保证视频流的分辨率
在该范围内, 当未指定时, 默认为 PIXELS_200W, 注意: AI-PaaS 不会对该字段进行实际拉流校验, 当
视频流分辨率超出指定范围时, 可能会导致解析服务负载过高."
    },
    "playType": {
      "enum": [
        "REALPLAY",
        "PLAYBACK"
      ],
      "type": "string",
      "title": "Play Type",
      "description": "取流类型 playback 为历史流, realplay 为实时流 默认是
REALPLAY."
    }
  }
},

```

```

        "startTime": {
            "type": "string",
            "description": "历史流开始时间，当 play_type 为 playback 时，必填，注意：
这个时间也会用作抓拍时间的起点。",
            "format": "date-time"
        },
        "endTime": {
            "type": "string",
            "description": "历史流结束时间，当 play_type 为 playback 时，必填，结束
时间必须大于开始时间。",
            "format": "date-time"
        },
        "playbackSpeed": {
            "type": "integer"
        }
    },
    "additionalProperties": true,
    "type": "object",
    "title": "Video Stream Options",
    "description": "视频流参数设置."
}
},
"additionalProperties": true,
"type": "object",
"oneOf": [
    {
        "required": [
            "capture_stream_options"
        ]
    },
    {
        "required": [
            "video_stream_options"
        ]
    }
],
"title": "Stream Options",
"description": "用于指定流的类型，以及流的参数，支持图片流和视频流的设置."
},
"id": {
    "type": "string",
    "title": "ID",
    "description": "任务的UUID，一般由算法平台生成"
},
"name": {
    "type": "string",
    "title": "Name",
    "description": "任务的自定义别名"
}
}

```

```

    },
    "additionalProperties": true,
    "type": "object",
    "title": "Task Info",
    "description": "算法服务所需的具体任务信息."
} 目标结构化类算法自定义输出参数 CaptureObject - JSON Schema 样例
// 参考新国标 - A.47 目标检测/目标属性识别结果对象 (AttributeRecResult)
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "required": [
    "objectType",
    "objectScore",
    "leftTopX",
    "leftTopY",
    "rightBtmX",
    "rightBtmY",
    "objectAttrs",
    "objectFeatureInfo",
    "trackId",
    "objectId",
    "areaId"
  ],
  "properties": {
    "objectType": {
      "type": "string",
      "description": "目标类型: 1- 人脸, 2- 人员, 3- 机动车, 4- 非机动车, 5- 物品",
      "title": "objectType",
      "enum": [
        "FACE",
        "PERSON",
        "MOTOR_VEHICLE",
        "NON_MOTOR_VEHICLE"
      ]
    },
    "leftTopX": {
      "type": "number",
      "description": "目标检测框 - 左上角 X 坐标",
      "title": "leftTopX"
    },
    "leftTopY": {
      "type": "number",
      "description": "目标检测框 - 左上角 Y 坐标",
      "title": "leftTopY"
    },
    "rightBtmX": {
      "type": "number",
      "description": "目标检测框 - 右下角 X 坐标",
      "title": "rightBtmX"
    },
  },
}

```

```

"rightBtmY": {
  "type": "number",
  "description": "目标检测框 - 右下角 Y 坐标",
  "title": "rightBtmY"
},
"objectId": {
  "type": "string",
  "description": "目标 id",
  "title": "objectId"
},
"objectScore": {
  "type": "number",
  "description": "目标置信度",
  "title": "objectScore"
},
"objectAttrs": {
  "additionalProperties": true,
  "type": "object",
  "description": "目标属性（年龄、性别、帽子、眼镜、口罩等）",
  "title": "objectAttrs"
},
"objectImage": {
  "type": "string",
  "contentEncoding": "base64",
  "contentType": "image/jpeg",
  "description": "目标抓拍图片数据, base64 image file格式",
  "title": "objectImage"
},
"objectImageUrl": {
  "type": "string",
  "description": "目标抓拍图片数据, url格式",
  "title": "objectImageUrl"
},
"objectFeatureInfo": {
  "required": [
    "vendor",
    "algoVer",
    "algoType",
    "featureData"
  ],
  "properties": {
    "vendor": {
      "type": "string",
      "description": "厂商",
      "title": "vendor"
    },
    "algoVer": {
      "type": "string",
      "description": "算法版本",

```

```

        "title": "algoVer"
    },
    "algoType": {
        "type": "string",
        "description": "算法类型",
        "title": "algoType"
    },
    "featureData": {
        "type": "string",
        "contentEncoding": "base64",
        "contentType": "application/octet-stream",
        "description": "特征向量数据, base64 encoded string",
        "title": "featureData"
    }
},
"type": "object",
"description": "目标特征向量信息",
"title": "objectFeatureInfo"
},
"trackId": {
    "type": "string",
    "description": "目标的 track_id, 当没有 track_id 时可以为 -1, 表示没有",
    "title": "trackId"
},
"areaId": {
    "type": "integer",
    "description": "目标抓拍对应的 area_id",
    "title": "areaId"
}
},
"additionalProperties": true,
"type": "object",
"title": "Object Capture Info",
"description": "目标抓拍."
}

```

E.2 事件检测类算法自定义输出参数 AlarmEvent - JSON Schema 样例

```

// 参考新国标 - A.53 绊线检测结果对象 (TripwireDetResult)
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "required": [
        "ruleId",
        "eventCode",
        "areaIds",
        "targets",
        "level"
    ],
    "properties": {

```

```

"ruleId": {
  "type": "integer",
  "description": "报警配置 id, 与触发报警的 RuleParam.rule_id 一致",
  "title": "ruleId"
},
"eventCode": {
  "type": "string",
  "description": "报警类型, 与触发报警的 RuleParam.event_type 一致",
  "title": "eventCode",
  "enum": [
    "FIRE"
  ]
},
"subEventCode": {
  "type": "string",
  "description": "报警类型, 与触发报警的 RuleParam.event_type 一致",
  "title": "subEventCode",
  "enum": [
    "PreAlarmFIRE"
  ]
}
"areaIds": {
  "items": {
    "type": "integer"
  },
  "type": "array",
  "description": "报警区域 id 列表, 触发此报警事件的所有 area_id 集合 当触发报警的事件包含多个目标时 (如消防设施报警), 可能会包含多个 area_id.",
  "title": "areaIds"
},
"targets": {
  "items": {
    "required": [
      "points",
      "targetType",
      "targetId",
      "targetScore",
      "trackId",
      "areaId"
    ],
    "properties": {
      "points": {
        "items": {
          "required": [
            "x",
            "y"
          ],
          "properties": {
            "x": {

```

```

        "type": "number",
        "title": "x"
    },
    "y": {
        "type": "number",
        "title": "y"
    }
},
"additionalProperties": true,
"type": "object",
"title": "Point"
},
"type": "array",
"description": "目标坐标",
"title": "points"
},
"targetType": {
    "type": "string",
    "description": "目标类型， 由 AlgoCabinObjType.ObjType 定义",
    "title": "targetType",
    "enum": [
        "FIRE"
    ]
},
"targetId": {
    "type": "string",
    "description": "目标 id",
    "title": "targetId"
},
"targetScore": {
    "type": "number",
    "description": "目标置信度",
    "title": "targetScore"
},
"trackId": {
    "type": "string",
    "description": "目标的 track_id, 当没有 track_id 时可以为 -1, 表示没有",
    "title": "trackId"
},
"areaId": {
    "type": "integer",
    "description": "触发报警时此目标对应的 area_id",
    "title": "areaId"
},
"maskId": {
    "type": "integer",
    "title": "maskId"
},
"custom": {

```

```

        "additionalProperties": true,
        "type": "object",
        "description": "see AlgoCabinCustomTarget in algo_cabin_custom.proto 定
制化功能自定义参数",
        "title": "custom"
    }
},
    "additionalProperties": true,
    "type": "object",
    "title": "Algo Cabin Target",
    "description": "目标对象."
},
    "type": "array",
    "description": "报警目标列表",
    "title": "targets"
},
    "level": {
        "enum": [
            "ALARM_LEVEL"
        ],
        "type": "string",
        "title": "Level",
        "description": "布控的报警层级."
    }
},
    "additionalProperties": true,
    "type": "object",
    "title": "Algo Cabin Alarm Event",
    "description": "报警事件."
}
}

```

E.3 态势统计类算法自定义输出参数 StatusInfo - JSON Schema 样例

```

// 参考新国标 - 目标流量统计结果对象 (FlowStaResult)
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "required": [
        "ruleId",
        "eventCode",
        "personDensityAbs"
    ],
    "properties": {
        "ruleId": {
            "type": "integer",
            "description": "分析配置id, 与触发消息的 RuleParam.rule_id 一致",
            "title": "ruleId"
        },
        "eventCode": {
            "type": "string",

```

```

    "description": "消息事件类型，与算法入参配置一致",
    "title": "eventCode",
    "enum": [
      "PERSON_DENSITY"
    ]
  },
  "personDensityAbs": {
    "type": "integer",
    "description": "人群密度估计结果绝对值，例如 48",
    "title": "personDensityAbs"
  },
  "personDensityHeatmap": {
    "type": "string",
    "contentEncoding": "base64",
    "contentType": "image/png",
    "description": "人群密度估计，热力图",
    "title": "personDensityHeatmap"
  }
},
"additionalProperties": true,
"type": "object",
"title": "Algo Person Density",
"description": "人群密度估计算法输出."
}

```

E.4 其他类算法自定义输出参数 UncategorizedInfo - JSON Schema 样例

// 参考新国标 - 视频图像增强

```

{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "required": [
    "ruleId",
    "params",
    "outputImage"
  ],
  "properties": {
    "ruleId": {
      "type": "integer",
      "description": "分析配置id，与触发消息的 RuleParam.rule_id 一致",
      "title": "ruleId"
    },
    "params": {
      "additionalProperties": true,
      "type": "object",
      "description": "图像增强处理相关参数",
      "title": "params"
    },
    "outputImage": {

```

```
    "type": "string",
    "contentEncoding": "base64",
    "contentMediaType": "image/jpeg",
    "description": "图像增强处理之后的图片",
    "title": "outputImage"
  },
  "inputImage" {
    "type": "string",
    "contentEncoding": "base64",
    "contentMediaType": "image/jpeg",
    "description": "图像增强处理之前的图片",
    "title": "inputImage"
  },
},
"additionalProperties": true,
"type": "object",
"title": "Algo Image Enhancement",
"description": "图像增强算法算法输出."
}
```
