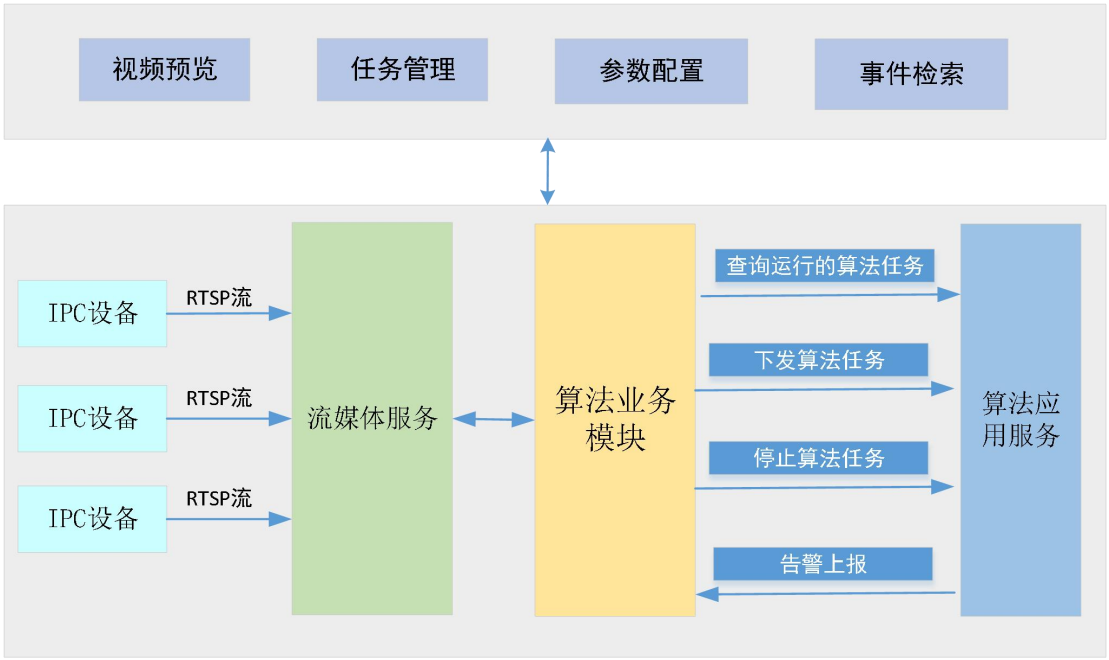
LITEOS 算法业务模块

1. LITEOS 算法业务介绍

LITEOS 算法业务模块作为可插拔插件部署于 LITEOS 基础服务之上，提供了 IPC 设备接入、视频实时预览、算法任务管理、算法参数配置以及算法告警事件检索功能，下面是 LITEOS 算法业务模块架构图。



LITEOS 算法业务模块架构图

模块介绍

流媒体服务：提供了 IPC 设备接入功能以及实时预览功能，管理 IPC 设备。需单独部署，可自由选择部署机器，输入流媒体服务 IP 和端口即可接入算法业务。

算法业务模块：作为模块化插件部署于 LITEOS 基础服务之上，对接算法应用服务，提供 AI 算法任务管理、AI 告警检索等功能。用户可根据自己的算法应用服务，在算法业务开源框架基础上，开发相应功能。为算法一体机软件服务的搭建、开发，提供便捷的能力支撑。

算法应用服务：由算法厂商提供，部署于算力设备上的算法应用服务，对接

LITEOS 的算法业务模块，对视频、图片进行 AI 算法分析。

本软件已经开源至 GitHub 平台，代码全部由 go 语言实现，用户可自由选择开发模式

2. 根据 LITEOS 开放 API 开发算法应用

LITEOS 提供了标准的算法对接 API，用户只需将根据下面的 API，开发算法应用，即可接入 LITEOS 算法业务平台，实现简单的算法一体机应用软件。

2.1 查询算法任务

接口 URL：*http://172.28.8.1:35001/dynamic/api/v1/find*

Content-Type：*application/json*

请求方式：*get*

请求头参数说明:

参数名	示例值	参数类型	是否必填	参数描述
Content-Type	application/json;charset=UTF-8	String	是	body 类型

返回参数说明：(200)成功

参数名	示例值	参数类型	参数描述
data		Array	
data.camerald	test1	String	算法任务的视频源名称

返回示例：(200)成功

```
{
    "data": [
        {
```

```
            "cameraId": "test1"
        },
        {
            "cameraId": "test2"
        },
        {
            "cameraId": "test3"
        }
    ]
}
```

2.2 停止运行任务

接口 URL：*http://172.28.8.1:35001/dynamic/api/v1/cancel*

Content-Type：*application/json*

请求方式：*post*

请求头参数说明：

参数名	示例值	参数类型	是否必填	参数描述
Content-Type	application/json;charset=UTF-8	String	是	暂无描述

请求体参数说明：

参数名	示例值	参数类型	是否必填	参数描述
cameraId	test3	String	是	算法任务的视频源名称

请求示例：

```
{
```

```
    "cameraId": "test3"
}
```

返回参数说明：(200)成功

参数名	示例值	参数类型	参数描述
message	success	String	成功
status	200	Integer	暂无描述

返回示例：(200)成功

```
{
    "message": "success",
    "status": 200
}
```

返回参数说明：(404)失败

参数名	示例值	参数类型	参数描述
code	1	Integer	暂无描述
msg	失败原因	String	暂无描述

返回示例：(404)失败

```
{
    "code": 1,
    "msg": "失败原因"
}
```

3.3 启动任务

接口 URL：*http://172.28.8.1:35001/dynamic/api/v1/setup/*

Content-Type：*application/json*

请求方式：*post*

请求头参数说明:

参数名	示例值	参数类型	是否必填	参数描述
Content-Type	application/json;charset=UTF-8	String	是	暂无描述

请求体参数说明:

参数名	示例值	参数类型	是否必填	参数描述
cameraId	test2	String	是	视频源名称
url	rtsp://172.28.8.119:26644/live/123/kaola_test_Smoking.mp4	String	是	rtsp 地址
notifyUrl	http://172.28.8.6:8081/algorithm/upload	String	是	告警上报地址
abilities		Array	是	算法能力
abilities.name	Smoking	String	是	算法名称
abilities.value		Object	是	算法能力参数配置
abilities.value.alarmInterval	10	Integer	是	告警间隔
abilities.value.minBox		Object	是	最小检测范围
abilities.value.minBox.width	50	Integer	是	最小检测范围宽度，单位 px
abilities.value.minBox.height	50	Integer	是	最小检测范围高度，单位 px
abilities.value.threshold	0.4	Number	是	告警阈值，取值 0-1

请求示例:

```
{
    "cameraId": "test2",
```

```
    "url":
"rtsp://172.28.8.119:26644/live/123/kaola_test_Smoking.mp4",

    "notifyUrl": "http://172.28.8.6:8081/algorithm/upload",
    "abilities": [
        {
            "name": "Smoking",
            "value": {
                "interval": 0.2,
                "alarmInterval": 10,
                "minBox": {
                    "width": 50,
                    "height": 50
                },
                "threshold": 0.4
            }
        },
        {
            "name": "WithoutHelmetOnSite",
            "value": {
            "alarmInterval": 5,
            "threshold": 0.4,
```

```
                "minBox": {
                    "width": 50,
                    "height": 50
                }
            }
        }
    ]
}
```


返回参数说明：(200)成功

参数名	示例值	参数类型	参数描述
message	success	String	成功
status	200	Integer	暂无描述

返回示例：(200)成功


```
{
    "message": "success",
    "status": 200
}
```


返回参数说明：失败

参数名	示例值	参数类型	参数描述
code	1	Integer	暂无描述
msg	失败原因	String	暂无描述

返回示例：失败

```
{
    "code": 1,
    "msg": "失败原因"
}
```

2.4 告警信息上报接口

接口 URL：*http://192.168.0.120:8080/algorithm/upload*

Content-Type：*application/json*

请求方式：*post*

请求头参数说明:

参数名	示例值	参数类型	是否必填	参数描述
Content-Type	application/json;charset=UTF-8	String	是	暂无描述

请求体参数说明:

参数名	示例值	参数类型	是否必填	参数描述
alarmType	MotorVehicleBreakIn	String	是	告警类型
boxes		Array	是	暂无描述
boxes.height	179	Integer	是	目标区域的高度
boxes.width	249	Integer	是	目标区域的宽度
boxes.x	1272	Integer	是	目标区域的 x 坐标起点
boxes.y	205	Integer	是	目标区域的 y 坐标起点
cameraId	测试图片	String	是	视频通道名称
extra		Object	是	目标检测的额外信息
extra.itemsInBox		Array	是	
extra.itemsInBox.con	0.9644582867622	Number	是	置信度（0-1）

fidence	375			
extra.itemsInBox.content		Object	是	保留字段
extra.itemsInBox.content.notSatisfy	0	Integer	是	保留字段
extra.itemsInBox.faceId		String	是	
ts	1695364805595	Integer	是	时间戳
scene		String	是	告警图片，base64 格式

请求示例：

```
{
    "alarmType": "MotorVehicleBreakIn",
    "boxes": [
        {
            "height": 179,
            "width": 249,
            "x": 1272,
            "y": 205
        },
        {
            "height": 210,
            "width": 287,
            "x": 903,
            "y": 272
```

```
        }
    ],
    "cameraId": "测试图片",
    "eventID": "dd86e988-5912-11ee-85df-0242ac110005",
    "extra": {
        "itemsInBox": [
            {
                "confidence": 0.9644582867622375,
                "type": "car",
                "content": {
                    "notSatisfy": 0
                },
                "faceId": ""
            },
            {
                "confidence": 0.9627598524093628,
                "type": "car",
                "content": {
                    "notSatisfy": 0
                },
                "faceId": ""
            }
```

```
        ]
    },
    "ts": 1695364805595,
    "scene":  "图片 base64 编码格式"
```

返回参数说明：(200)成功

参数名	示例值	参数类型	参数描述
message	success	String	成功
status	200	Integer	暂无描述

返回示例：(200)成功

```
{
    "message": "success",
    "status": 200
}
```

返回参数说明：(404)失败

参数名	示例值	参数类型	参数描述
code	1	Integer	暂无描述
msg	失败原因	String	暂无描述

返回示例：(404)失败

```
{
    "code": 1,
    "msg": "失败原因"
}
```

3. 开发 LITEOS 算法模块源码，适配算法应用

3.1 项目结构

目录结构

- api: 接口业务逻辑处理
- build: 编译打包脚本文件
- client: http client 请求
- config: 配置文件和配置读取驱动
- database: 数据库文件
- global: 全局变量
- initialization: 服务启动初始化
- logger: 日志管理
- middware: 中间件
- mvc: 结构体配置
- release: 安装包发布
- routes: 路由定义
- scrip: 部署脚本

3.2 算法业务功能详解

3.2.1 添加任务

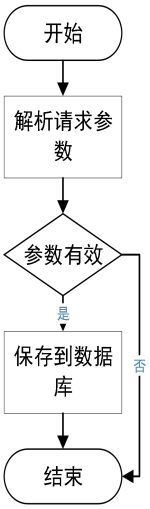
在前端填入信息，点击确认按钮后，向后端发送添加任务请求

请求 body 如下：

```
{
   "taskName": "10001",
   "abilities": [
      "PeopleWithoutMask"
   ],
```

```
    "deviceName": "face",
    "deviceId": "32028101001310000012",
    "url": "rtsp://172.28.8.34:26644/live/123/face.mp4"
}
```


后端通过 api/v1/algorithm 目录中 task.go 中 func (b *TaskApi) AddTask(c *gin.Context) 函数进行处理，处理流程如下：



3.2.2 修改任务

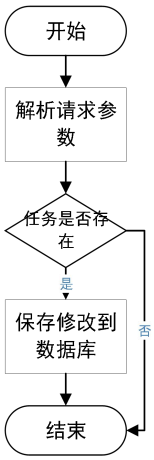
由于任务编号是任务的唯一编号，所以任务编号不支持修改。

在前端页面提交修改任务，点击确认

请求 body 如下：


```
{
    "taskName": "10001",
    "deviceId": "32028101001310000012",
    "abilities": [
        "PeopleWithoutMask",
        "UnexpectedEvents"
    ]
}
```


后端通过 api/v1/algorithm 目录中 task.go 中 func (b *TaskApi) ModTask(c *gin.Context) 函数进行处理，处理流程如下：



3.2.3 删除任务

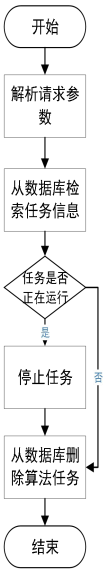
在前端点击删除按钮

序号	任务编号	视频源	算法配置信息	任务状态	操作
1	test3	test3	机动车识别（不带车牌）	已停止	
2	10001	face	电动车检测	运行中	

请求 body 为：

```
{
   "taskName": "10001"
}
```

后端通过 api/v1/algorithm 目录中 task.go 中 func (b *TaskApi) DeleteTask(c *gin.Context) 函数进行处理，处理流程如下：

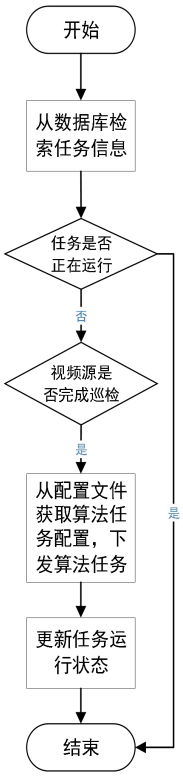


3.2.4 启动算法任务

添加算法任务后，可以通过页面启动按钮运行算法任务，请求 body 为：

```
{
   "taskName": "10001"
}
```

后端通过 api/v1/algorithm 目录中 task.go 中 func (b *TaskApi) StartTask(c *gin.Context) 函数进行处理，处理流程如下：



下发算法任务流程示例：

首先通过 getTaskString("task.base")函数获取基本结构体：

```
{"cameraId":"{{taskId}}","url":"{{streamUrl}}","imageOut":"base64","inputType":"RTSP","notifyUrl":"{{uploadUrl}}","skipFrame":25,"roi":[],"areaBoxes":[],"abilities":[{{abilities}}]}
```

然后根据算法任务的信息，替换`{{taskId}}、{{streamUrl}}、{{uploadUrl}}`。

根据算法能力列表，从配置文件中获取 abilities 字段的值

```
{"name":"Smoking","value":{"interval":0.2,"alarmInterval":10,"minBox":{"width":50,"height":50},"threshold":0.4,"areaBoxes":"{{hotRegion}}","roi":[]}}
```

其中，`alarmInterval、threshold、minBox`是通过参数配置页面进行设置，最终构造出的启动任务请求 body 为：

{"cameraId":"face","url":"rtsp://172.28.8.34:26644/live/123/face.mp4","imageOut":"base64","inputType":"RTSP","notifyUrl":"http://172.28.8.6:8081/algorithm/upload","skipFrame":25,"roi":[],"areaBoxes":[],"abilities":[{"name":"ElectricCarEntersElevator","value":{"interval":1,"alarmInterval": 20,"minBox":{"width":50,"height":50},"threshold":0.40,"arealsReverse":false,"objMinCount": 1,"areaBoxes":[],"roi":[]}}]}

3.2.5 停止算法任务

点击停用按钮，停止算法任务，发送 body 为：

```
{
   "taskName": "10001"
}
```

向算法应用下发结束任务请求。