

编号：

密级：公开

智能识别软件和安检机 交互接口规范 (数据部分)

V1.5

湖南苏科智能科技有限公司

2025 年 1 月

名称	智能识别软件和安检机数据交互接口规范 (数据部分)		编号	
			版本	V1.5
编制		质量		
审核		批准		
更改记录	序号	主要内容	修改人	修改时间
	1	编辑初稿	邹一振	2024-08-01
	2	修改章节内容	邹一振	2024-08-26
	3	修改接口请求参数	邹一振	2024-09-14
	4	修改 5.2.1 识别接口（响应结果增加危险等级、违禁品颜色）	邹一振	2024-11-08
	5	修改 5.2.1 识别接口（响应结果增加扩展属性）	邹一振	2024-11-25
	6	修改 5.2.1 识别接口（请求参数增加包裹图在安检机通道中的上下边距）	邹一振	2025-01-21
	7	修改 5.2.1 识别接口（请求和响应参数都增加扩展数据）	伍成乾	2025-02-11
分发范围				
备注				

目 录

1	文档说明.....	2
2	制定目标.....	2
3	交互说明.....	2
4	接口组成.....	2
5	接口设计.....	3
5.1	设备接口.....	3
5.1.1	首次握手（Hello）	3
5.1.2	心跳（Ping）	4
5.2	检测.....	5
5.2.1	识别（Detect）	5
5.3	基础数据.....	10
5.3.1	危险等级（RiskLevels）	10
5.3.2	违禁品类别（RiskObjects）	12

1 文档说明

此文档制定 X 光安检机与智能识别软件之间数据交互接口规范，后续软件实现须严格按照此文档规定要求执行。

2 制定目标

本规范目标旨在通过设计一套数据交互接口协议及数据结构，形成规范化、标准化接口标准，满足各种安检 X 光智能识别数据对接要求，简化对接复杂性，提高一致性和扩展性。

3 交互说明

如下：

- 安检机请求数据，按 API 接口协议规范以 form-data 形式请求智能识别软件；
- 智能识别软件接收到安检机的请求数据，按 API 接口协议规范解析为载荷数据；
- 智能识别软件对载荷数据进行检测分析，得到处理结果；
- 智能识别软件将检测处理结果数据标准化，封装形成响应载荷，并添加状态，最终以 json 形式返回给安检机；
- 安检机接收到结果载荷数据和状态后，完成后续处理工作；
- 接口 URL 上的 version 指 Restful 规范中 API 接口版本号（如：V1）；
- 报文编码采用 utf-8 标准进行编码。

4 接口组成

智能识别软件的 API 包含设备、AI 检测和标准数据等三类接口。

表 1 接口列表

接口名称	API 名字	作用	备注
1.设备			
1.1 握手	hello	设备初始握手交互，安检机发送设备信息，智能识别软件返回 Token 等	
1.2 心跳	ping	获取智能识别软件心跳及设备状态	
2.AI 检测			
2.1 识别	detect	完成 AI 识别检测，返回检测结果	
3.标准数据			

接口名称	API 名字	作用	备注
3.1 危险等级	risk_levels	获取危险等级标准代码表	
3.2 违禁品类别	risk_objects	获取违禁品类别标准代码表	

5 接口设计

5.1 设备接口

5.1.1 首次握手（Hello）

- (1) 作用：安检机与智能识别软件初始交互接口（可理解为登录接口）。
- (2) 访问地址：~/ski/device/<version>/hello
- (3) 请求方式：Post
- (4) 请求载荷：

表 2 Hello 请求载荷

属性	数据类型	可空	描述	备注
DeviceID	string	否	安检机设备号	
Vender	string	否	厂商唯一标识	参考表 3 厂商 ID 进行填写
Model	string	否	安检机型号代码	
Software	string	否	软件代号	统一规定，如 SKScan。
Version	string	否	软件版本号	如 V1.0-20221012
At	string	是	标记安检机在所在位置	具体实施约定
IsBigEnd	bool	否	是否使用大端传输	默认：否(X86)

表 3 厂商列表

厂商	厂商 ID	备注
苏科智能	Ski	

以苏科 6550D 安检机为例，载荷参考如下：

```
{
  "DeviceID": "D007758",
  "Vender": "Ski",
  "Model": "6550D",
  "Software": "SKScan",
  "Version": "V2.5.13SKGD-20220618",
  "At": "16 号线|体育公园|一号口",
  "IsBigEnd": false
}
```

- (5) 响应载荷：

表 4 Hello 响应载荷

属性	数据类型	可空	描述	备注
Code	int	否	状态码（成功：200，失败：-1）	请求结果状态
Msg	string	是	返回信息	
IsSuccess	bool	否	请求是否成功	
Data	object	否	请求结果载荷	
Data 的结构体如下：				
ID	string	否	智能识别软件标识 ID	
Version	string	否	智能识别软件版本号	
Date	date	否	智能识别软件软件日期	格式：yyyy-MM-dd
Token	string	否	token	用于后续请求

响应载荷参考如下：

```
{
  "Code": 200,
  "Msg": "success",
  "IsSuccess": true,
  "Data": {
    "ID": "SKI46ec31673132",
    "Version": "2.0",
    "Date": "2024-07-30",
    "Token": "SKI46ec31673132403c97afe65ef5b71136"
  }
}
```

注意事项：

- （1）安检机首次调用智能识别软件接口时需请求 Hello 接口获取 Token 用于后续接口请求。

5.1.2 心跳（Ping）

- （1）作用：获取智能识别软件心跳及设备状态。
- （2）访问地址：~/ski/device/<version>/ping
- （3）请求方式：Post
- （4）请求载荷：

表 5 Ping 请求载荷

属性	数据类 型	可空	描述	备注
DeviceID	string	否	安检机设备号	
Token	string	否	token	用于后续请求

请求载荷参考如下：

```
{
  "DeviceID": "dev0011",

```

```
"Token": "SKI46ec31673132403c97afe65ef5b71136"
}
```

(5) 响应载荷:

表 6 Ping 响应载荷

属性	数据类型	可空	描述	备注
Code	int	否	状态码（成功：200，失败：-1）	请求结果状态
Msg	string	是	返回信息	
IsSuccess	bool	否	请求是否成功	
Data	object	否	请求结果载荷	
Data 的结构体如下：				
Status	int	否	智能识别软件状态码（故障：-1，正常：1）	
TimeStamp	DateTime	否	时间	格式：yyyy-MM-dd HH:mm:ss

响应载荷参考如下：

```
{
  "Code": 200,
  "Msg": "success",
  "IsSuccess": true,
  "Data": {
    "Status": 1,
    "TimeStamp": "2024-07-14 14:06:20"
  }
}
```

5.2 检测

5.2.1 识别（Detect）

- (1) 作用：完成 X 光机伪彩图、原子序数、高低能数据的 AI 识别，返回识别结果。
- (2) 访问地址：~/ski/detect/<version>/detect
- (3) 请求方式：Post
- (4) 请求载荷：

AI 识别请求数据按 Web “form-data” 方式传输, 数据结构如下表所示：

表 7 识别请求数据结构

属性	数据类型	可空	描述	备注
DeviceID	string	否	设备 ID	全局唯一
PackageID	string	否	包裹 ID	全局唯一
Token	string	否	令牌	会话唯一

属性	数据类型	可空	描述	备注
ImageM	file	否	主视角图片	二进制数据流 (格式详见“数据格式要求”)。 相同视角的原子序数、高/低能尺寸要与图片保持一致，像素对齐。 高/低能单像素点数据精度要求为 0~65535
ImageS	file	是	侧视角图片	
ImageT	file	是	三视角图片	
AtomM	file	否	主视角原子序数	
AtomS	file	是	侧视角原子序数	
AtomT	file	是	三视角原子序数	
PowerMH	file	否	主视角高能	
PowerML	file	否	主视角低能	
PowerSH	file	是	侧视角高能	
PowerSL	file	是	侧视角低能	
PowerTH	file	是	三视角高能	
PowerTL	file	是	三视角低能	
Size	string	否	图片、原子序数和高低能宽高	Json 字符串
Extend	string	否	扩展信息	Json 字符串

Size 字段传输图片、原子序数和高低能尺寸大小等信息，字符串以 Json 格式承载。

表 8 请求数据结构中 Size 字段属性

属性	数据类型	可空	描述	备注
Main	object	否	主视角数据宽高	结构参见表 9
Sub	object	是	侧视角数据宽高	
Third	object	是	三视角数据宽高	

表 9 请求数据结构中宽高属性

属性	数据类型	可空	描述	备注
Image	array	否	图像宽高[Width, Height]数组	
Atom	array	是	原子序数宽高[Width, Height]数组	
Power	array	是	高低能宽高[Width, Height]数组	

Size 字符串 Json 示例如下：

```
[
  {
    "Main": {
      "Image": [800, 600],
      "Atom": [800, 600],
      "Power": [800, 600]
    },
    "Sub": {
      "Image": [800, 600],
      "Atom": [800, 600],
      "Power": [800, 600]
    },
    "Third": {
      "Image": [800, 600],
      "Atom": [800, 600],
      "Power": [800, 600]
    }
  }
]
```

```

    }
}

```

Extend 字段传输包裹图片在安检机通道内的边距信息和分片信息等信息，字符串以 Json 格式承载。

表 10 请求数据结构中 Extend 字段属性

属性	数据类型	可空	描述	备注
Margin	object	是	包裹图片在安检机通道内的边距信息	结构参见表 11
Slice	object	是	分片信息	无此字段 或为 null 表示不分片，结构参见表 12

表 11 请求数据结构中 Margin 属性

属性	数据类型	可空	描述	备注
Top	int	是	包裹图片在安检机通道内的上边距	
Bottom	int	是	包裹图片在安检机通道内的下边距	

表 12 请求数据结构中 Slice 属性

属性	数据类型	可空	描述	备注
ID	short	是	分片 ID	-1 不分片 从 0 开始， 0x8FFF 表示分片结束

Extend 字符串 Json 示例如下：

```

{
  "Margin": {
    "Bottom": 50,
    "Top": 50
  },
  "Slice": {
    "ID": 0
  }
}

```

数据格式要求：

- 1) 图片：X 光伪彩图片为 Jpeg 图片，24 位深度。
- 2) 原子序数：原子序数矩阵（二维数组）数据采用列存储（探测板采集列存储）方式，单像素 1 字节。
- 3) 高/低能：高/低能矩阵（二维数组）数据采用列存储（探测板采集列存储）方式，单像素 2 字节无符号整数。字节数据传输方式（大小端）根据登录接口中 IsBigEnd 字段的值进行选择。
- (5) 响应载荷：

识别结果数据结构如下表所示：

表 13 识别结果数据结构

属性	数据类型	可空	描述	备注
ID	string	否	包裹 ID	请求的包裹 ID
Main	list	是	主视角检测结果	
Sub	list	是	侧视角检测结果	
Third	list	是	三视角检测结果	
Extend	Object	是	扩展信息	
TimeSpan	int	否	本次 AI 识别处理时间	单位为毫秒（ms）
TimeStamp	datetime	否	时间	格式：yyyy-MM-dd HH:mm:ss

其中 Main、Sub、Third 包含以下属性：

表 14 检测结果属性

属性	数据类型	可空	描述	备注
Object	object	是	检测违禁品	二者至少一个不为空
Material	object	是	物质识别结果	
Rect	string	否	位置“x,y,width,height”	左上角为坐标原点
Measure	object	是	尺寸预估结果	

表 13 Extend 数据结构

属性	数据类型	可空	描述	备注
AtSlice	int	是	检测结果起始分片 ID	

表 15 Object/Material 属性

属性	数据类型	可空	描述	备注
ID	string	否	识别物体标识码	参见标准规范
Code	string	否	物品代码	参见标准规范
Name	string	否	物品名称	参见标准规范
Label	string	是	标注西文标签	参见标准规范
Score	float	否	置信度	
RiskLevel	string	是	危险等级编码	参见危险等级定义表
Color	string	是	违禁品颜色	
ObjectExt	object	是	扩展属性	

表 16 Measure 属性

属性	数据类型	可空	描述	备注
Len	float	是	检测结果尺寸	单位：mm
Vol	float	是	违禁品体积、容积	单位：ml

表 17 ObjectExt 属性

属性	数据类型	可空	描述	备注
Code	string	是	第三方物品代码	
Name	string	是	第三方物品名称	
RiskLevel	string	是	第三方危险等级编码	
Color	string	是	第三方违禁品颜色	

识别结果示例如下：

```
{
  "ID": "100",
  "Main": [
    {
      "Object": {
        "ID": "92405",
        "Code": "SL.G.001",
        "Name": "塑料罐",
        "Label": "plasticTank",
        "Score": 0.25,
        "RiskLevel": "10",
        "Color": "#FF0000",
        "ObjectExt": {
          "Code": "123",
          "Name": "塑料罐",
          "RiskLevel": "10",
          "Color": "#FF0000"
        }
      },
      "Rect": "129,279,282,279",
      "Material": {
        "ID": "4270",
        "Code": "SL.G.001C",
        "Name": "塑料罐(易燃)",
        "Label": "plasticTank",
        "Score": 0.25,
        "RiskLevel": "10",
        "Color": "#FF0000",
        "ObjectExt": {
          "Code": "123",
          "Name": "塑料罐",
          "RiskLevel": "10",
          "Color": "#FF0000"
        }
      }
    },
    {
      "Object": {
        "ID": "3768",
        "Code": "HY.BZ.001",
        "Name": "爆竹",
        "Label": "firecrackers",
        "Score": 0.78,
        "RiskLevel": "10",
        "Color": "#FF0000",
        "ObjectExt": {
          "Code": "456",
```

```

        "Name": "爆竹",
        "RiskLevel": "10",
        "Color": "#FF0000"
    },
    "Rect": "129,279,282,279",
    "Material": null,
    "Measure": null
},
],
"Sub": [
    {
        "Object": {
            "ID": "3768",
            "Code": "HY.BZ.001",
            "Name": "爆竹",
            "Label": "firecrackers",
            "Score": 0.78,
            "RiskLevel": "10",
            "Color": "#FF0000",
            "ObjectExt": {
                "Code": "456",
                "Name": "爆竹",
                "RiskLevel": "10",
                "Color": "#FF0000"
            }
        },
        "Rect": "129,279,282,279",
        "Material": null,
        "Measure": null
    }
],
"Third": [],
"Extend": {
    "AtSlice": 10
},
"TimeSpan": 35,
"TimeStamp": "2024-07-14 14:06:20"
}

```

5.3 基础数据

5.3.1 危险等级（RiskLevels）

- (1) 作用：获取危险等级标准代码。
- (2) 访问地址：~/ski/code/<version>/risk_levels
- (3) 请求方式：Get
- (4) 危险等级定义表：

表 18 危险等级定义表

等级编码	等级名称	描述
00	零级	安全的
10	一级	可忽略
20	二级	临界
30	三级	危险
40	四级	破坏性

(5) 响应载荷:

表 19 RiskLevels 响应载荷属性 (数组)

属性	数据类型	可空	描述	备注
Code	string	否	危险等级代码	
Name	string	否	名称	
LevelName	string	否	危险等级名称	
Enable	bool	否	是否启用	
Remark	string	是	备注	

响应载荷参考如下:

```
[
  {
    "Code": "00",
    "Name": "安全",
    "LevelName": "零级",
    "Enable": true,
    "Remark": ""
  },
  {
    "Code": "10",
    "Name": "可忽略",
    "LevelName": "一级",
    "Enable": true,
    "Remark": ""
  },
  {
    "Code": "20",
    "Name": "临界",
    "LevelName": "二级",
    "Enable": true,
    "Remark": ""
  },
  {
    "Code": "30",
    "Name": "危险",
    "LevelName": "三级",
    "Enable": true,
    "Remark": ""
  },
  {
    "Code": "40",
    "Name": "破坏性",
    "LevelName": "四级",
    "Enable": true,
    "Remark": ""
  }
]
```

```
}  
]
```

5.3.2 违禁品类别（RiskObjects）

- (1) 作用：获取违禁品类别(含扩展类别)标准代码表。
- (2) 访问地址：~/ski/code/<version>/risk_objects
- (3) 请求方式：Get
- (4) 响应载荷：

表 20 RiskObjects 响应载荷（数组）

属性	数据类型	可空	描述	备注
Code	string	否	类别代码	
Name	string	否	名称	
Label	string	否	英文名	
Threshold	float	否	阈值	
RiskLevel	string	否	危险等级代码	
Enable	bool	否	是否启用	
Remark	string	是	备注	

响应载荷参考如下：

```
[  
  {  
    "Code": "JS.SQ.001",  
    "Name": "手枪",  
    "Label": "gun",  
    "Threshold": 80,  
    "RiskLevel": "10",  
    "Enable": true,  
    "Remark": ""  
  },  
  {  
    "Code": "SJ-00251",  
    "Name": "违禁品 251",  
    "Label": "Label001",  
    "Threshold": 80,  
    "RiskLevel": "10",  
    "Enable": true,  
    "Remark": ""  
  }  
]
```