**一、背景描述**

**1、项目背景**：厨师口罩识别算法就是应用于厨房监控视频中通过机器视觉算法识别是否带了口罩，可对进入作业区域的人员进行自动识别：若检测到人员未佩戴口罩，可立即报警，报警信号同步推送至管理人员。该算法极大地提升了明厨亮灶项目更好的落地，也提高了消费者对厨房重地的信任。



**2、其他补充：无**

**二、目标描述**

**1、需求边界定义**：人员有无佩戴口罩（蓝色口罩、白色口罩、黑色口罩）

**2、算法报警的业务逻辑**：检测到有人未佩戴口罩，输出报警信息

**3、识别场景**：厨房

**4、识别对象**：人、口罩

**5、环境要求**：室外、室内，光照充足。

**三、算法内容**

1、算法**输入**：单个视频流，彩色画面

2、算法**输入设备**：摄像头3-5米

3、算法**输出**：callback

4、1080P算法**最小识别像素**：口罩最小40\*40像素

5、算法**实时性**：实时性高

**四、数据说明**

1、标签：

（1）front\_wear：标注正面人头，正面佩戴口罩包住鼻子

（2）front\_no\_wear：标注正面人头，正面未佩戴口罩

（3）front\_under\_nose\_wear：标注正面人头，正面佩戴口罩在鼻子下且在嘴巴上面

（4）front\_under\_mouth\_wear：标注正面人头，正面佩戴口罩在嘴巴下面

（5）mask\_front\_wear：标注正面口罩，口罩包住鼻子

（6）mask\_front\_under\_nose\_wear：标注正面口罩，口罩在鼻子下且在嘴巴上面

（7）mask\_front\_under\_mouth\_wear：标注正面口罩，口罩在嘴巴下面

-------------------------------------------------------------------------------------------

（8）side\_wear：标注侧面人头，侧面佩戴口罩包住鼻子【口罩在鼻子下面不明显情况标注该标签】

（9）side\_no\_wear：标注侧面人头，侧面未佩戴口罩

（10）side\_under\_nose\_wear：标注侧面人头，侧面佩戴口罩在鼻子下且在嘴巴上面【口罩在鼻子下面很明显情况标注该标签】

（11）side\_under\_mouth\_wear：标注侧面人头，侧面佩戴口罩在嘴巴下面

（12）mask\_side\_wear：标注侧面口罩，口罩包住鼻子

（13）mask\_side\_under\_nose\_wear：标注侧面口罩，口罩在鼻子下且在嘴巴上面

（14）mask\_side\_under\_mouth\_wear：标注侧面口罩，口罩在嘴巴下面

-------------------------------------------------------------------------------------------

（15）side\_back\_head\_wear：标注侧背面人头，如果能够看到口罩带子或者看不清楚口罩带子认为戴了口罩

（16）side\_back\_head\_no\_wear：标注侧背面人头，如果不能够看到口罩带子认为没戴

-------------------------------------------------------------------------------------------

（17）back\_head：标注背面人头

（18）strap：带子

（19）front\_unknown：标注正面人头，不能很明确知道正面是否佩戴口罩

（20）side\_unknown：标注侧面人头，不能很明确知道侧面是否佩戴口罩

**五、软硬件要求**

**1.软件要求**

运行在X86架构（酷睿i系列芯片和至强系列芯片，显卡支持RTX、GTX、Tesla）

需求为Linux SDK（C++实现）

算法部署：支持私有化部署

**2.硬件要求**

1. 英伟达显卡**（默认）**

**六、自动测试**

平台提供的自动测试服务，意在模拟算法真实场景下的落地过程，并且为大家提供方便的模型测试工具。因此，您需要按照平台的要求封装SDK，也就是按照平台封装模型推理时的输入输出。您可以选择python或者C++来封装，为了简便这里建议您使用python。

**本榜得分采取准确率、算法性能绝对值综合得分的形式，其中IoU使用0.5，具体如下**

|  |  |  |
| --- | --- | --- |
| 评审指标指标说明权重分 |  |  |
| 算法精度 | F-score | 85 |
| 算法性能值 | 性能分=fps/90，如果fps>90则当满分100计算 | 15 |
| 总分 | Score=算法精度\*85+算法性能值\*15 |  |

**实战榜测试逻辑描述：**使用f1-score的测试algorithm\_data中的目标输出，包含以下几类

front\_no\_wear：正面未佩戴口罩

front\_under\_nose\_wear：正面佩戴口罩在鼻子下且在嘴巴上面

front\_under\_mouth\_wear：正面佩戴口罩在嘴巴下面

side\_no\_wear：侧面未佩戴口罩

side\_under\_nose\_wear：侧面佩戴口罩在鼻子下且在嘴巴上面

side\_under\_mouth\_wear：侧面佩戴口罩在嘴巴下面

**模型榜测试逻辑描述：**使用f1-score指标测试目标检测

**1、SDK封装输出规范**

**使用接口**

Python接口

用户需要按照要求实现如下函数接口，发起测试时，系统会调用文件/project/ev\_sdk/src/ji.py，并将测试图片逐次送入process\_image接口，需要实现的程序接口：

import argparse

import numpy as np

import cv2

import json

def init():

"""

Initialize model

Returns: model

"""

return {}

def process\_image(net, input\_image, args=None):

"""

Do inference to analysis input\_image and get output

Attributes:

handle: algorithm handle returned by init()

input\_image (numpy.ndarray): image to be process, format: (h, w, c), BGR

Returns: process result

"""

fake\_result = {}

fake\_result["algorithm\_data"] = {

"is\_alert": False,

"target\_count": 0,

"target\_info": []

}

fake\_result["modeldata"] = {}

fake\_result["modeldata"]["objects"] = [

{

"x": 716,

"y": 716,

"height": 646,

"width": 233,

"confidence": 0.999660,

"name": "front\_wear"

},

{

"x": 397,

"y": 397,

"height": 488,

"width": 215,

"confidence": 0.978979,

"name": "mask\_front\_wear"

},

{

"x": 397,

"y": 397,

"height": 488,

"width": 215,

"confidence": 0.978979,

"name": "strap"

}

]

return json.dumps(fake\_result, indent=4)

C++接口

请参考 /usr/local/ev\_sdk/include/ji.h

ev\_sdk 4.0接口如下

JiErrorCode ji\_init(int argc, char \*\*argv);

void\* ji\_create\_predictor(JiPredictorType pdtype);

JiErrorCode ji\_calc\_image(void\* predictor, const JiImageInfo\* pInFrames, const unsigned int nInCount, const char\* args, JiImageInfo \*\*pOutFrames, unsigned int & nOutCount, JiEvent &event);

接⼝的返回值 event.json 需要满⾜算法输出规范。

**2.口罩识别 json 输出协议**

不告警情形

{

"algorithm\_data": {

"is\_alert": false,

"target\_count": 0,

"target\_info": []

},

"model\_data": {

"objects": [

{

"x": 716,

"y": 716,

"height": 646,

"width": 233,

"confidence": 0.999660,

"name": "front\_wear"

},

{

"x": 397,

"y": 397,

"height": 488,

"width": 215,

"confidence": 0.978979,

"name": "mask\_front\_wear"

},

{

"x": 397,

"y": 397,

"height": 488,

"width": 215,

"confidence": 0.978979,

"name": "strap"

}

]

}

}

告警情形示例

{

"algorithm\_data": {

"is\_alert": true,

"target\_count": 1,

"target\_info": [

{

"x": 716,

"y": 716,

"height": 646,

"width": 233,

"confidence": 0.999660,

"name": "front\_no\_wear"

}

]

},

"model\_data": {

"objects": [

{

"x": 716,

"y": 716,

"height": 646,

"width": 233,

"confidence": 0.999660,

"name": "front\_no\_wear"

}

]

}

}

**说明**  
target\_info的值类型为列表，列表可以包括n个元素；  
每个元素必须包含：矩形目标框、目标类别；  
如果包含置信度，键名为confidence；  
目标类别名称使用name表示，且其值必须可以唯一区分当前目标所属的类别。

2.2字段详解

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 字段名称 | 类型 | 备注 |
| 1 | algorithm\_data | object | 业务相关的输出信息 |
| 2 | is\_alert | bool | 报警标识, true告警, false不告警 |
| 3 | target\_count | int | target\_info中对象数目 |
| 4 | target\_info | array | 报警信息 |
| 5 | model\_data | object | 模型相关的输出信息 |
| 6 | objects | array | 模型输出信息 |

* **target\_info字段详解:**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 字段名称 | 类型 | 备注 |
| 1 | x | int | 识别框左上角X坐标 |
| 2 | y | int | 识别框左上角Y坐标 |
| 3 | width | int | 识别框宽度 |
| 4 | height | int | 识别框高度 |
| 5 | confidence | double | 置信度 |
| 6 | name | string | 检测到告警时目标名称:  front\_no\_wear：正面未佩戴口罩  front\_under\_nose\_wear：正面佩戴口罩在鼻子下且在嘴巴上面  front\_under\_mouth\_wear：正面佩戴口罩在嘴巴下面  side\_no\_wear：侧面未佩戴口罩  side\_under\_nose\_wear：侧面佩戴口罩在鼻子下且在嘴巴上面  side\_under\_mouth\_wear：侧面佩戴口罩在嘴巴下面 |

* **object字段详解:**

|  |  |  |  |
| --- | --- | --- | --- |
| 序 号 | 字段名称 | 类型 | 备注 |
| 1 | x | int | 识别框左上角X坐标 |
| 2 | y | int | 识别框左上角Y坐标 |
| 3 | width | int | 识别框宽度 |
| 4 | height | int | 识别框高度 |
| 5 | confidence | double | 置信度 |
| 6 | name | string | 模型检测到的目标名称（与标注对应）:  （1）front\_wear：标注正面人头，正面佩戴口罩包住鼻子  （2）front\_no\_wear：标注正面人头，正面未佩戴口罩  （3）front\_under\_nose\_wear：标注正面人头，正面佩戴口罩在鼻子下且在嘴巴上面  （4）front\_under\_mouth\_wear：标注正面人头，正面佩戴口罩在嘴巴下面  （5）front\_unknown：标注正面人头，不能很明确知道正面是否佩戴口罩  （6）side\_wear：标注侧面人头，侧面佩戴口罩包住鼻子  （7）side\_no\_wear：标注侧面人头，侧面未佩戴口罩  （8）side\_under\_nose\_wear：标注侧面人头，侧面佩戴口罩在鼻子下且在嘴巴上面  （9）side\_under\_mouth\_wear：标注侧面人头，侧面佩戴口罩在嘴巴下面  （10）side\_unknown：标注侧面人头，不能很明确知道侧面是否佩戴口罩  （11）side\_back\_head\_wear：标注侧后方人头，带了口罩  （12）side\_back\_head\_no\_wear：标注侧后方人头，没有戴口罩  （13）back\_head：标注背面人头  （14）mask\_front\_wear：标注正面口罩，口罩包住鼻子  （15）mask\_front\_under\_nose\_wear：标注正面口罩，口罩在鼻子下且在嘴巴上面  （16）mask\_front\_under\_mouth\_wear：标注正面口罩，口罩在嘴巴下面  （17）mask\_side\_wear：标注侧面口罩，口罩包住鼻子  （18）mask\_side\_under\_nose\_wear：标注侧面口罩，口罩在鼻子下且在嘴巴上面  （19）mask\_side\_under\_mouth\_wear：标注侧面口罩，口罩在嘴巴下面  （20）**strap：口罩的带子** |

备注:  
1. 其中algorithm\_data字段对应的是业务数据，只有当检测到有至少1个告警目标时才会使得  
target\_info不为空集。  
2. model\_data字段对应的是模型数据，只要模型检测到目标，则objects不为空；否则为空[]