# EV\_SDK开发测试指南

目录

[EV\_SDK开发测试指南 1](#_Toc114042961)

[1. EV\_SDK与上层应用交互过程 2](#_Toc114042962)

[2. 上层应用如何获取算法的分析结果 2](#_Toc114042963)

[2.1 返回json说明 2](#_Toc114042964)

[2.2 返回图片说明 4](#_Toc114042965)

[3, 关于ROI的配置 5](#_Toc114042966)

[4. 关于动态配置的管理 6](#_Toc114042967)

[5. 关于EV\_SDK的编译 6](#_Toc114042968)

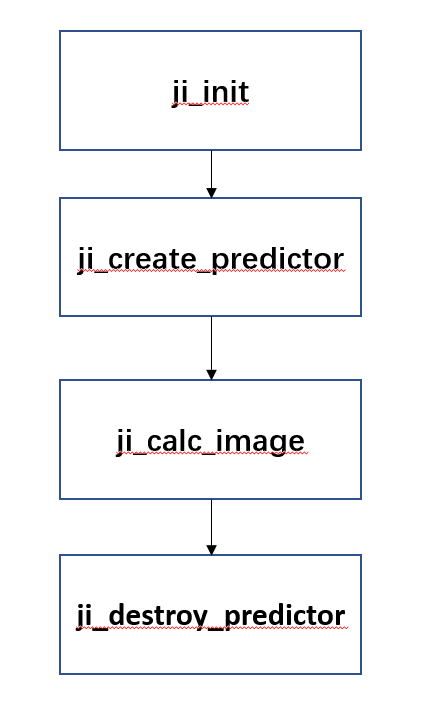
[6. 关与SDK demo的使用 7](#_Toc114042969)

[7. 算法自测要点 7](#_Toc114042970)

[8. 测试工具 8](#_Toc114042971)

## EV\_SDK与上层应用交互过程

上层应用通过ji\_create\_predictor接口创建算法实例，通过ji\_calc\_image接口传入图片进行算法分析，分析结束后调用ji\_destroy\_predictor接口释放算法实例。在分析过程中，上层应用也会通过ji\_update\_config接口动态更新算法实例对应的参数，并要求更新的参数能够实时生效。



## 2. 上层应用如何获取算法的分析结果

上层应用要求算法输出一个json字符串和一张图片来返回算法分析的结果。所以开发者在测试的过程中一定要注意输出的json和图片是否满足对应的算法要求。

### 2.1 返回json说明

每一个算法在发布的时候都会说明对应的json规范。开发者在实现EV\_SDK的时候，需要通过ji\_calc\_image接口中的JiEvent输出结构体的json字段返回算法的输出json，json字符串应该包含算法规范中的所有字段。

对于一般的检测类算法来说，返回json主要有两个部分，“model\_data”里面包含了所有的检测目标信息，这里是模型的原始检测结果，不包含任何算法处理的结果。“algorithm\_data”里面包含了是否报警的标志“is\_alert”，以及符合报警逻辑的检测目标信息。开发者需要注意“is\_alert”字段需要和JiEvent中的code字段对应，上层应用会校验这两个部分的值是否对应(“is\_alert”= true, code=0 /“is\_alert”= false, code=1)。一个典型的报警/未报警的json如下所示。

报警json 未报警json

一个使用jsoncpp生成json的典型列子如图所示，其中包含了对浮点数的精度设置。



### 2.2 返回图片说明

上层应用会利用算法返回图片进行可视化，所以算法返回的图片一定要进行正确的可视化,并能通过动态配置更改可视化的效果，一般的可视化内容包括ROI绘制/检测结果绘制/报警文字绘制等。Demo中可视化的代码如下所示。



## 3, 关于ROI的配置

极市的大多数算法都要求实现ROI的功能，即对图像中指定的多边形区域进行分析，例如对于行人闯入算法，只有当检测到的行人目标框的中心点位于多边形区域内时才算作一个违法目标。EV\_SDK中提供了判断点与多边形位置关系的函数供开发者使用，大家可以参考demo的实现，demo中直接判断所有的检测目标中心是否位于roi内部。



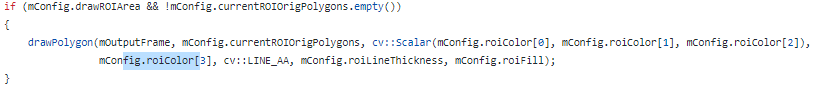
## 4. 关于动态配置的管理

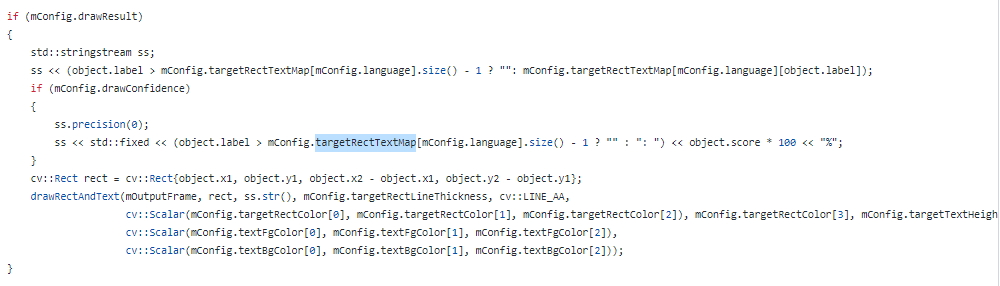
EV\_SDK 要求配置参数能够在算法运行的过程中进行动态的修改，上层应用主要是通过ji\_update\_config接口，以及ji\_calc\_image接口中的args参数来进行动态修改。所以开发者一定要保证动态配置能够生效。开发者在demo的基础上新增一个配置时需要在Configuration类中添加对应的成员变量，并在ParseAndUpdateArgs接口中添加更新规则。在算法中使用配置的时候一定使用Configuration的对象成员来进行配置获取，不要将配置值赋给其他变量再间接访问。

例如运行检测的时候，也是通过配置项来获取阈值



例如可视化的时候也是通过配置来获取颜色等配置参数





## 5. 关于EV\_SDK的编译

编译EV\_SDK主要分为编译算法库和测试工具两部分，开发者必须确保两部分都能成功地编译。编译EV\_SDK执行如下命令，确保要执行INSTALL。如果/usr/local/ev\_sdk/build目录不存在请先生成目录。

cd /usr/local/ev\_sdk/build && cmake -DCMAKE\_BUILD\_TYPE=Release ../ && make install -j4

编译测试工具执行以下命令，请确保不要对测试工具的代码和编译脚本做任何更改。如果/usr/local/ev\_sdk/test/build目录不存在请先生成目录。

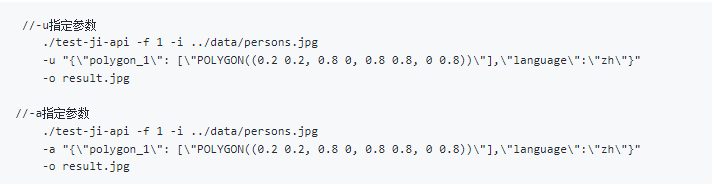
cd /usr/local/ev\_sdk/test/build && cmake -DCMAKE\_BUILD\_TYPE=Release ../ && make install -j4

## 6. 关与SDK demo的使用

SDK 使用的不同的模型以及推理引擎来实现模型推理，开发者在复用SDK demo的时候需要确定模型结构及推理引擎是否和demo中使用的一致.demo中已经实现了roi解析，动态配置管理，json封装等功能，开发者可以参考其中的实现。

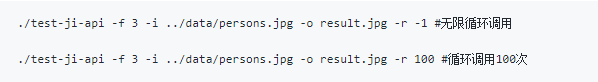
## 7. 算法自测要点

5.1 配置是否生效，通过test-ji-api的-u 和 -a两个参数都要测试，例如改变模型阈值，roi等观察算法的输出和可视化效果是否有对应的改变。



5.2 测试在报警和不报警两种情况下，算法的输出json和返回的JiEvent中的code 值是否符合算法对应的规范。

5.3 内存测试，运行 ./test-ji-api -f 3 -i ../data/person.jpg -r 1000,观测进程内存是否存在一直增长的情况。



## 8. 测试工具

7.1定位内存泄露，valgrind工具

7.2定位coredump, gdb工具