**VideoCheck class接口调用说明**

**函数SetRectRangle**

把摄像头的初始化参数通过结构体VideoInitParam传递过去

|  |  |
| --- | --- |
| 参数 | 含义 |
| rect\_x | 限定区域的横坐标 |
| rect\_y | 限定区域的纵坐标 |
| rect\_weight | 限定区域的宽 |
| rect\_height | 限定区域的高 |
| i\_threshold | 二值化的初始值 |
| rect\_pa\_x | 托盘限定区域的横坐标 |
| rect\_pa\_y | 托盘限定区域的纵坐标 |
| rect\_pa\_weight | 托盘限定区域的宽 |
| rect\_pa\_height | 托盘限定区域的高 |
| animal\_mask\_image\_x | 匹配模板的宽 |
| animal\_mask\_image\_y | 匹配模板的高 |
| pallet\_threshold | 第二张图二值化的初始值 |

初始化的各个值

VideoInitParam videoparam =

{

rect\_x = 120;

rect\_y = 100;

rect\_weight = 400;

rect\_height = 300;

rect\_pa\_x = 190

rect\_pa\_y = 140

rect\_pa\_weight = 260

rect\_pa\_height = 230

animal\_mask\_image\_x = 34

animal\_mask\_image\_y = 9

i\_threshold = 230;

pallet\_threshold = 180

};

**函数 bool OpenVideo(int index);**

返回值是bool, true表示打开摄像头正确，false表示打开摄像头错误

填的参数是摄像头的ID值

**函数int StartRun();**

一个线程开启摄像头检测

返回值是-1，代表摄像头异常断开，返回值是0，代表摄像头正常关闭

返回-1，要再次开启摄像头，需要重新设置 OpenVideo

**函数 StartCheckCassette()**

开启检测卡座上芯片是否放置准确的功能

注意：与功能函数StartCheckPallet()有冲突，两者只能打开一个，

打开此功能后第一次调用GetResultq()前需要暂停100ms，以确保数据正确

**函数 StopCheckCassette();**

关闭检测卡座芯片位置的功能

注意：在调用StartCheckPallet()前需要调用一下

**函数GetResult(ResultParam \*result);**

把视觉处理计算的值赋值给结构体ResultParam

|  |  |
| --- | --- |
| 参数 | 意义 |
| float center\_x | 返回检测的芯片中心横坐标 |
| float center\_y | 返回检测的芯片中心纵坐标 |
| float angle | 返回检测的芯片角度 |

注意：只有打开StartCheckCassette()调用此函数返回值才有效，如果判断的返回值和预期的不一样，建议调用第二次再判断

**函数 void StartCheckPallet()**

开启检测托盘上的芯片功能

注意：与功能函数StartCheckCassette()有冲突，两者只能打开一个

打开此功能后第一次调用Point GetPalletPoint();需要暂停100ms，以确保数据正确

**函数void StopCheckPallet();**

关闭检测托盘芯片的功能

注意：在开启StartCheckCassette()前需要调用一下，确保功能不冲突

**函数Point GetPalletPoint();**

返回值是Point，即检测得到的芯片位置，得到的值需要外加判断是否正确，

注意：开启StartCheckPallet()功能后才能调用此函数

如果返回值判断有错，最好再调用一次

**函数void StopRun();**

停止检测