**说明**

demo演示了如何使用动态库来检测一个目录下所有的图片

**目录**

├─test

│ │ test.cpp

│ ├─include

│ │ ai\_cpp\_dll.lib

│ │ ai\_v2\_class.hpp

└─x64

└─Release

│ ai\_cpp\_dll.dll

│ ai\_cpp\_dll.lib

│ opencv\_world410.dll

│ pthreadGC2.dll

│ pthreadVC2.dll

│ test.exe

└─yunsheng

├─data

│ ai-voc\_last.weights #和power-ai.dll一定要在同一目录

│ config.data

│ power-ai.dll #和ai-voc\_last.weights一定要在同一目录

│ voc.names

├─result

└─testimg

**参数说明**

* thresh 检测的阈值
* names\_file 对应的缺陷分类对象名称文件(缺陷名可更改)
* cfg\_file 配置文件路径
* weights\_file 权重文件

float thresh = 0.2;//设置阈值

string names\_file = "C:/Users/Administrator/source/repos/test/x64/Release/yunsheng/data/voc.names";

string cfg\_file = "C:/Users/Administrator/source/repos/test/x64/Release/yunsheng/data/config.data";

string weights\_file = "C:/Users/Administrator/source/repos/test/x64/Release/yunsheng/data/ai-voc\_last.weights";

**demo中使用了opencv来处理图片和描框，这里不做配置说明，主要说明如何接入检测**

1. 引用ai\_cpp\_dll.lib，配置项目>属性>(选择Release)>链接库>附加目录库填写ai\_cpp\_dll.lib所在的目录

图片说明

1. 链接库

#pragma comment(lib, "ai\_cpp\_dll.lib")

1. 引入ai\_v2\_class.hpp头文件

#include "ai\_v2\_class.hpp"

1. 初始化检测器

Detector detector(cfg\_file, weights\_file);

1. 通过objects\_names\_from\_file方法获得分类对象名称

auto obj\_names = objects\_names\_from\_file(names\_file);

1. 读取文件夹下图片检测  
   检测的函数detector.detect(mat\_img, thresh)

cout << ">>>>>>>>>>>>>>>>>>>>>>>>>> \n" << files[i].c\_str()<<endl;

cv::Mat mat\_img = cv::imread(files[i].c\_str()); // opencv 读取图片

auto start = std::chrono::steady\_clock::now(); // 开始时间

std::vector<bbox\_t> result\_vec = detector.detect(mat\_img, thresh); // 检测函数

auto end = std::chrono::steady\_clock::now(); // 结束时间

std::chrono::duration<double> spent = end - start; // 检测时间

std::cout << " Time: " << spent.count() << " sec \n<<<<<<<<<<<<<<<<<<<<<<<<\n";

draw\_boxes(mat\_img, result\_vec, obj\_names); // 最后把缺陷框描绘在图像上

cv::imwrite("C:/Users/Administrator/source/repos/test/x64/Release/yunsheng/result/" + filesname[i], mat\_img); // 保存检测的结果图片

**小礼物走一走，来简书关注我**