目录

[1. 概述](#_Toc16197_WPSOffice_Level1) [1](#_Toc16197_WPSOffice_Level1)

[1.1. 概述](#_Toc6981_WPSOffice_Level2) [1](#_Toc6981_WPSOffice_Level2)

[1.2. 参考域说明](#_Toc1526_WPSOffice_Level2) [1](#_Toc1526_WPSOffice_Level2)

[1.2.1. 参考域说明](#_Toc6981_WPSOffice_Level3) [1](#_Toc6981_WPSOffice_Level3)

[1.2.2. 数据类型参考域](#_Toc1526_WPSOffice_Level3) [2](#_Toc1526_WPSOffice_Level3)

[2. API参考](#_Toc6981_WPSOffice_Level1) [2](#_Toc6981_WPSOffice_Level1)

[2.1. init](#_Toc28274_WPSOffice_Level2) [2](#_Toc28274_WPSOffice_Level2)

[2.2. detec](#_Toc771_WPSOffice_Level2) [3](#_Toc771_WPSOffice_Level2)

[2.3. uninit](#_Toc17458_WPSOffice_Level2) [3](#_Toc17458_WPSOffice_Level2)

[2.4. init\_no\_gpu](#_Toc2358_WPSOffice_Level2) [4](#_Toc2358_WPSOffice_Level2)

[2.5. detec\_no\_gpu](#_Toc20708_WPSOffice_Level2) [4](#_Toc20708_WPSOffice_Level2)

[2.6. uninit\_no\_gpu](#_Toc25687_WPSOffice_Level2) [5](#_Toc25687_WPSOffice_Level2)

[3. 结构体说明](#_Toc1526_WPSOffice_Level1) [5](#_Toc1526_WPSOffice_Level1)

[3.1. 图像格式](#_Toc7800_WPSOffice_Level2) [5](#_Toc7800_WPSOffice_Level2)

[3.2. 输出结果的结构体](#_Toc14035_WPSOffice_Level2) [6](#_Toc14035_WPSOffice_Level2)

[3.2.1. 单个结果的结构体](#_Toc28274_WPSOffice_Level3) [6](#_Toc28274_WPSOffice_Level3)

[3.2.2. 区域的结构体](#_Toc771_WPSOffice_Level3) [7](#_Toc771_WPSOffice_Level3)

# 概述

## 概述

此为yolov3版本算法库和示例代码。

## 参考域说明

### 参考域说明

本说明文档选用了8个描述API的相关信息，它们的作用如下表1-1所示。

表1-1 API参考域说明

|  |  |
| --- | --- |
| 参考域 | 含义 |
| 1 目的 | 描述相应 API 的主要功能 |
| 2 语法 | 列出调用 API 需要包括的头文件以及 API 的原形声明 |
| 3 参数 | 列出了 API 中调用的参数，参数类型以及参数属性说明 |
| 4 描述 | 简要的描述 API 的工作过程 |
| 5 返回值 | 描述了 API 返回的值及其含义 |
| 6 需求 | 描述 API 包含的头文件以及其需要连接的库 |
| 7 注意 | 使用 API 时需要注意的点 |
| 8 举例 | 描述如何使用 API |

### 数据类型参考域

本说明文档采用了 5 个参考域描述数据类型的相关信息，它们的作用如下表 1-2 所示。

表1-2 参考域描述数据类型

|  |  |
| --- | --- |
| 参考域 | 含义 |
| 1 说明 | 描述该数据类型的主要功能 |
| 2 含义 | 列出数据类型的定义 |
| 3 成员 | 列出该数据类型包含的成员 |
| 4 注意事项 | 列出了使用该数据类型的注意点 |
| 5 相关数据类型及接口 | 列出了与该数据类型相关的数据类型以及接口 |

# API参考

## init

|  |  |  |  |
| --- | --- | --- | --- |
| 参考域 | 含义 | | |
| 1 目的 | 初始化句柄，加载模型到内存中 | | |
| 2 语法 | 需包含头文件"yolo\_dll.h"  调用语法 init(void \*\*handle, char \*cfgfile, char \*weightfile,float thresh,int gpu\_index) | | |
| 3参数 | void \*\*handle | | 模型空间句柄 |
| char \*cfgfile | | 模型网络结构文件 |
| char \*weightfile | | 模型网络权重文件 |
| float thresh | | 模型阈值 |
| int gpu\_index | | 调用的GPU显卡序号 |
| 4 描述 | 初始化句柄，加载模型到内存中 | | |
| 5 返回值 | 0 | 接口运行正常 | |
| 非0 | 接口运行异常 | |
| 6 需求 | 调用需要包含头文件"yolo\_dll.h"  需要包含库文件“darknet.dll”“darknet.lib”“yolo\_cpp\_dll.dll”“yolo\_cpp\_dll.lib” | | |
| 7 注意 |  | | |
| 8 举例 | void \*handle;  char \*cfgfile;  char \*weightfile;  cfgfile = "yolov3-tiny-voc.cfg";  weightfile = "yolov3-tiny-voc\_18700.weights";  init(&handle, cfgfile, weightfile, 0.24, 0) | | |

## detec

|  |  |  |  |
| --- | --- | --- | --- |
| 参考域 | 含义 | | |
| 1 目的 | 检测输入图像中的目标 | | |
| 2 语法 | 需包含头文件"yolo\_dll.h"  调用语法 detec(void \*handle, YOLOV3\_PIMAGES img, HYYOLOV3RESULT\_PLIST pResultList); | | |
| 3参数 | void \*handle | | 模型空间句柄 |
| YOLOV3\_PIMAGES img | | 输入待测图片 |
| HYYOLOV3RESULT\_PLIST pResultList | | 输出结果 |
| 4 描述 | 检测输入图像中的目标 | | |
| 5 返回值 | 0 | 接口运行正常 | |
| 非0 | 接口运行异常 | |
| 6 需求 | 调用需要包含头文件"yolo\_dll.h"  需要包含库文件“darknet.dll”“darknet.lib”“yolo\_cpp\_dll.dll”“yolo\_cpp\_dll.lib” | | |
| 7 注意 |  | | |
| 8 举例 | detec(handle, &img, &resultlist); | | |

## uninit

|  |  |  |  |
| --- | --- | --- | --- |
| 参考域 | 含义 | | |
| 1 目的 | 释放内存 | | |
| 2 语法 | 调用需要包含头文件"yolo\_dll.h"  调用语法uninit(void \*handle) | | |
| 3参数 | void \*handle | | 模型空间句柄 |
| 4 描述 | 检测输入图像中的目标 | | |
| 5 返回值 | 0 | 接口运行正常 | |
| 非0 | 接口运行异常 | |
| 6 需求 | 调用需要包含头文件"yolo\_dll.h"  需要包含库文件“darknet.dll”“darknet.lib”“yolo\_cpp\_dll.dll”“yolo\_cpp\_dll.lib” | | |
| 7 注意 |  | | |
| 8 举例 | uninit(handle); | | |

## init\_no\_gpu

|  |  |  |  |
| --- | --- | --- | --- |
| 参考域 | 含义 | | |
| 1 目的 | 初始化句柄，加载模型到内存中 | | |
| 2 语法 | 需包含头文件"yolo\_dll.h"  调用语法 init\_no\_gpu(void \*\*handle, char \*cfgfile, char \*weightfile, float thresh) | | |
| 3参数 | void \*\*handle | | 模型空间句柄 |
| char \*cfgfile | | 模型网络结构文件 |
| char \*weightfile | | 模型网络权重文件 |
| float thresh | | 模型阈值 |
| 4 描述 | 初始化句柄，加载模型到内存中 | | |
| 5 返回值 | 0 | 接口运行正常 | |
| 非0 | 接口运行异常 | |
| 6 需求 | 调用需要包含头文件"yolo\_dll.h"  需要包含库文件“darknet\_no\_gpu.dll”“darknet\_no\_gpu.lib”“yolo\_cpp\_dll\_no\_gpu.dll”“yolo\_cpp\_dll\_no\_gpu.lib” | | |
| 7 注意 |  | | |
| 8 举例 | void \*handle;  char \*cfgfile;  char \*weightfile;  cfgfile = "yolov3-tiny-voc.cfg";  weightfile = "yolov3-tiny-voc\_18700.weights";  init\_no\_gpu(&handle, cfgfile, weightfile, 0.24) | | |

## detec\_no\_gpu

|  |  |  |  |
| --- | --- | --- | --- |
| 参考域 | 含义 | | |
| 1 目的 | 检测输入图像中的目标 | | |
| 2 语法 | 需包含头文件"yolo\_dll.h"  调用语法 detec\_no\_gpu(void \*handle, YOLOV3\_PIMAGES img, HYYOLOV3RESULT\_PLIST pResultList); | | |
| 3参数 | void \*handle | | 模型空间句柄 |
| YOLOV3\_PIMAGES img | | 输入待测图片 |
| HYYOLOV3RESULT\_PLIST pResultList | | 输出结果 |
| 4 描述 | 检测输入图像中的目标 | | |
| 5 返回值 | 0 | 接口运行正常 | |
| 非0 | 接口运行异常 | |
| 6 需求 | 调用需要包含头文件"yolo\_dll.h"  需要包含库文件“darknet\_no\_gpu.dll”“darknet\_no\_gpu.lib”“yolo\_cpp\_dll\_no\_gpu.dll”“yolo\_cpp\_dll\_no\_gpu.lib” | | |
| 7 注意 |  | | |
| 8 举例 | detec\_no\_gpu(handle, &img, &resultlist) | | |

## uninit\_no\_gpu

|  |  |  |  |
| --- | --- | --- | --- |
| 参考域 | 含义 | | |
| 1 目的 | 释放内存 | | |
| 2 语法 | 调用需要包含头文件"yolo\_dll.h"  调用语法uninit\_no\_gpu(void \*handle) | | |
| 3参数 | void \*handle | | 模型空间句柄 |
| 4 描述 | 检测输入图像中的目标 | | |
| 5 返回值 | 0 | 接口运行正常 | |
| 非0 | 接口运行异常 | |
| 6 需求 | 调用需要包含头文件"yolo\_dll.h"  需要包含库文件“darknet\_no\_gpu.dll”“darknet\_no\_gpu.lib”“yolo\_cpp\_dll\_no\_gpu.dll”“yolo\_cpp\_dll\_no\_gpu.lib” | | |
| 7 注意 |  | | |
| 8 举例 | uninit\_no\_gpu(handle); | | |

# 结构体说明

## 图像格式

|  |  |  |
| --- | --- | --- |
| 参数名称 | YOLOV3\_IMAGES, \*YOLOV3\_PIMAGES | |
| 参数定义 | typedef struct {  int lWidth; // Off-screen width  int lHeight; // Off-screen height  int lPixelArrayFormat; // Format of pixel array  union  {  struct  {  int lLineBytes;  void \*pPixel;  } chunky;  struct  {  int lLinebytesArray[4];  void \*pPixelArray[4];  } planar;  } pixelArray;  } YOLOV3\_IMAGES, \*YOLOV3\_PIMAGES; | |
| 参数成员 | lWidth | 图像宽度 |
| lHeight | 图像高度 |
| lPixelArrayFormat | 图像格式，如RGB，YUV等 |
| pixelArray | 图像数据存储形式 |
| chunky | 图像按chunky格式存储，lLineBytes指图像的行长，\*pPixel为像素指针 |
| planar | 图像按planar格式存储， lLinebytesArray[4]为图像的行长， \*pPixelArray[4]为图像的像素指针 |
| 注意事项 | 需要将读取的图像转换为该图像格式，为接口函数调用，彩图默认通道顺序为‘BGR’ | |
| 相关接口 | detec\_no\_gpu,detec | |

## 输出结果的结构体

|  |  |  |
| --- | --- | --- |
| 参数名称 | HYYOLOV3RESULT\_LIST,\*HYYOLOV3RESULT\_PLIST | |
| 参数定义 | typedef struct  {  HYYOLOV3\_RESULT \*pResult;  int lResultNum; //程序得到结果数目  }HYYOLOV3RESULT\_LIST,\*HYYOLOV3RESULT\_PLIST; | |
| 参数成员 | pResult | 算法输出的目标结果 |
| lResultNum | 算法输出的目标结果数目 |
| 注意事项 |  | |
| 相关接口 | detec\_no\_gpu,detec | |

### 单个结果的结构体

|  |  |  |
| --- | --- | --- |
| 参数名称 | HYYOLOV3\_RESULT | |
| 参数定义 | typedef struct  {  int flag;//  YOLOV3RECT Target;///目标rectangle  float dVal; //目标value  float dConfidence; //目标confidence  }HYYOLOV3\_RESULT;///text read result | |
| 参数成员 | flag | （冗余变量） |
| dVal | 输出类别 |
| dConfidence | 输出目标置信度 |
| Target | 目标位置 |
| 注意事项 | 无 | |
| 相关接口 | 无 | |

### 区域的结构体

|  |  |  |
| --- | --- | --- |
| 参数名称 | YOLOV3RECT, \*YOLOV3PRECT | |
| 参数定义 | typedef struct  {  int left;  int top;  int right;  int bottom;  } YOLOV3RECT, \*YOLOV3PRECT; | |
| 参数成员 | left | 左上角横坐标 |
| top | 左上角纵坐标 |
| right | 右下角横坐标 |
| bottom | 右下角纵坐标 |
| 注意事项 | 无 | |
| 相关接口 | 无 | |