

PaddleOCRSharp 使用说明文档

文档版本：V1.0.0

适用版本：v5.1.0

更新日期：2026年3月

目录

1.	概述	3
1.1.	简介	3
1.2.	应用场景	3
1.3.	依赖环境	4
	windows 系统	4
	linux 系统	4
	.NET 框架	5
2.	安装方式	6
2.1.	免费版	6
	在线方式	6
	离线方式	6
2.2.	付费版	7
3.	架构设计	8
3.1.	架构设计	8
4.	快速入门	9
4.1.	C#如何使用	9
4.2.	C++如何使用	10
4.3.	Python 如何使用	10
5.	PaddleOCRSharp 接口说明	11
5.1.	PaddleOCREngine 接口说明	11
	5.1.1. 构造函数	11
	5.1.2. 文本识别	11
	5.1.3. 资源释放	12
5.2.	类说明	13
	5.2.1. OCRModelConfig 模型配置对象	13
	5.2.2. OCRParameter 参数对象	13
	5.2.3. ModifyParameter 动态修改参数对象	14
	5.2.4. OCRResult 文字识别结果对象	14
	5.2.5. TextBlock 文本框对象	14
	5.2.6. OCRPoint 文本坐标对象	15
5.3.	内部逻辑机制	15
6.	PaddleOCR 接口说明	16
6.1.	接口说明	16
	6.1.1. 初始化与配置	16
	6.1.2. 文本识别	17
	6.1.3. 资源释放	18
7.	常见问题与解决方案	19
8.	示例项目	20
9.	其他	20

1. 概述

1.1. 简介

PaddleOCRSharp 是一个 .NET 版本 OCR 可离线使用类库。核心组件 PaddleOCR.dll 目前已经支持 C\C++、.NET、Python、Golang、Rust、java、labview、delphi 等众多开发语言的直接 API 接口调用，支持 windows、linux 跨平台。项目包含文本识别、文本检测、表格识别功能。本库做了大量优化，提高了识别率和推理性能。默认包含超轻量级中文 OCR，单模型支持中英文数字组合识别、竖排文本识别、长文本识别。同时支持中英文、纯英文以及多种语言文本检测识别。使用者可以自行修改其他模型或者自行训练的模型。

PaddleOCRSharp 封装极其简化，实际调用仅几行代码，极大的方便了中下游开发者的使用和降低了 PaddleOCR 的使用入门级别，几乎覆盖 .NET 所有框架使，方便各个行业应用开发与部署。Nuget 包即装即用，可以离线部署，不需要网络就可以识别的高精度中英文 OCR。

本库支持飞桨通用 OCR 模型，如：PPOCRV2、PPOCRV3、PPOCRV4、PP-OCRv4_server、PP-OCRv4_server_doc（1.5 万字符字典模型）。PP-OCRv5、PP-OCRv5_server、en_PP-OCRv5，当前版本默认使用中英文 V5 模型（PP-OCRv5，1.8+万字符字典模型）。

1.2. 应用场景

PaddleOCRSharp 主要应用场景：

文档数字化处理： 对于大量的纸质文档，PaddleOCRSharp 可以快速地将其转化为电子文档，方便存储、检索和编辑。这在企业级应用中尤为重要，如图书馆、档案馆、政府部门等需要对大量文档进行数字化处理的场景。

自动识别表单数据： 在需要自动化处理表单数据的场景中，PaddleOCRSharp 可以识别表单中的文字信息，并将其转化为结构化数据。这大大提高了数据录入的效率和准确性，特别适用于银行、保险、医疗等行业需要处理大量表单数据的场景。

车牌识别： PaddleOCRSharp 也支持车牌识别功能，可以准确地识别出车辆的车牌号码。这对于交通管理、停车场管理、安防监控等场景非常有用，可以帮助实现车辆的快速识别和追踪。

图像文字提取： 在需要从图像中提取文字信息的场景中，如从截图、图片或 PDF 文件中提取文字，PaddleOCRSharp 都能提供高精度的识别结果。这对于研究人员、学者、学生等需要处理大量图像文字信息的用户来说非常便捷。

多语言识别： PaddleOCRSharp 支持多种语言的识别，包括中文、英文等常用语言。这使得它可

以在国际化的应用场景中发挥重要作用，如跨境电商、国际交流等领域。

定制化需求： 对于有特殊需求的用户，PaddleOCRSharp 提供了丰富的接口和模型库，可以根据具体需求进行定制化和扩展。这为用户提供了更多的灵活性和可能性，可以满足不同场景下的特定需求。PaddleOCRSharp 凭借其强大的 OCR 功能和广泛的应用场景，成为了 .NET 开发者在处理 OCR 任务时的有力工具。无论是企业级应用还是个人开发者，都可以通过 PaddleOCRSharp 快速实现 OCR 功能，提高工作效率和数据处理的准确性。

OCR 文字识别场景应用覆盖通用，制造、金融、交通等行业的主要 OCR 垂类应用。例如：工业激光喷码识别、点阵文字识别、数码管识别、液晶屏读数识别、电表识别、表单、增值税发票、印章检测与识别、通用卡证识别、银行电子回单、身份证识别、通用表格识别、车牌识别、驾驶证/行驶证识别、快递单识别等。

1.3. 依赖环境

windows 系统

操作系统支持 win7SP1_x64、win10_x64 及以上、win11、win server2012R2_x64 及以上。免费版仅支持在 x86_64 的 CPU 上使用，CPU 指令集需要包含 AVX2 指令集。付费版 x86 架构的 CPU 指令集支持 SSE4.2 指令集。

win7SP1 需要额外安装 VC++2017 运行时，win server2012R2 需要额外补丁文件。可以在公司官网资料中心下载。下载地址：www.yingtianit.com/datacenter。

linux 系统

x86_64 架构支持：

操作系统：统信 UOS、Kylin_V10\SP1\SP3\V11、ubuntu22.04、CentOS8，

CPU 支持：如 Intel、AMD、国产的海光，兆芯 CPU。CPU 指令集支持 SSE4.2 指令集。

ARM64 架构支持：

操作系统：统信 UOS、Kylin_V10\SP1\SP3\V11、ubuntu22.04。

CPU 支持：如鲲鹏、飞腾、瑞芯微。

Loongarch64 架构支持：

操作系统：统信 UOS-20、KylinV10SP1、Loongnix

CPU 支持：龙芯 3A5000、龙芯 3A6000。

.NET 框架

Windows 系统：PaddleOCRSharp 支持 .NET Framework 4.0~4.8，.NET 6.0 及以上框架。

linux 系统：支持 .NET 6.0 及以上框架。

2. 安装方式

2.1. 免费版

在线方式

Nuget 包管理界面安装 PaddleOCRSharp 包或者直接引用 PaddleOCRSharp.dll 库文件。PaddleOCRSharp 依赖 Paddle.Runtime.win_x64 组件，如果是 .NET Framework 框架额外依赖 Newtonsoft.Json 组件，如果是 .NET Core 框架额外依赖 System.Drawing.Common 组件。使用包管理安装方式，依赖自动添加。其中 PaddleOCRSharpV5.1.0 版本依赖 Paddle.Runtime.win_x64 版本为 3.1.0.1。项目安装 PaddleOCRSharp 包同时需要安装 Paddle.Runtime.win_x64 包。



离线方式

PaddleOCRSharp 包可以在线下载解压使用，下载地址：

<https://www.nuget.org/packages/PaddleOCRSharp>,

依赖运行时 Paddle.Runtime.win_x64 包，下载地址：

https://www.nuget.org/packages/Paddle.Runtime.win_x64

解压有关包文件，复制到程序目录下。PaddleOCRSharp 包解压后，提取 build\PaddleOCRLib 文件夹下的所有文件，复制到项目输出目录下：

 inference	2025-11-21 13:05	文件夹	
 PaddleOCR.dll	2025-7-1 10:51	应用程序扩展	886 KB

PaddleOCRSharp.5.1.0\lib 目录包含所有 .net 框架对应的 PaddleOCRSharp.dll 文件，根据项目使用 .net 框架的版本不同而使用。

Paddle.Runtime.win_x64 包解压提取\build\win_x64 文件夹下的文件，制到项目输出目录下。

 common.dll	2025-7-20 3:40	应用程序扩展	676 KB
 con crt140.dll	2023-3-12 13:48	应用程序扩展	302 KB
 CPU指令集检测工具.exe	2025-6-22 2:29	应用程序	307 KB
 libiomp5md.dll	2025-7-1 2:12	应用程序扩展	1,687 KB
 mkl dnn.dll	2025-7-1 2:16	应用程序扩展	26,876 KB
 mkl ml.dll	2025-7-1 2:12	应用程序扩展	90,478 KB
 msvcp140.dll	2023-3-12 13:48	应用程序扩展	551 KB
 msvcp140_1.dll	2023-3-12 13:48	应用程序扩展	24 KB
 msvcp140_2.dll	2023-3-12 13:48	应用程序扩展	182 KB
 opencv_world470.dll	2022-12-28 15:59	应用程序扩展	62,112 KB
 paddle_inference.dll	2025-7-20 3:40	应用程序扩展	124,847 KB
 tbb12.dll	2024-5-24 3:04	应用程序扩展	183 KB
 tbbmalloc.dll	2024-5-24 3:04	应用程序扩展	106 KB
 tbbmalloc_proxy.dll	2024-5-24 3:04	应用程序扩展	41 KB
 vcamp140.dll	2023-3-12 13:48	应用程序扩展	397 KB
 vccorlib140.dll	2023-3-12 13:48	应用程序扩展	324 KB
 vcomp140.dll	2023-3-12 13:48	应用程序扩展	179 KB
 vc runtime140.dll	2023-3-12 13:48	应用程序扩展	94 KB
 vc runtime140_1.dll	2023-3-12 13:48	应用程序扩展	36 KB
 yaml-cpp.dll	2022-12-3 6:04	应用程序扩展	279 KB

2.2. 付费版

付费版仅能通过购买的方式获取有关付费版完整的文件。可以联系我公司 QQ：277784829 购买或者前往公司官网添加微信购买。公司网址：www.yingtianit.com

付费版介绍：www.yingtianit.com/product

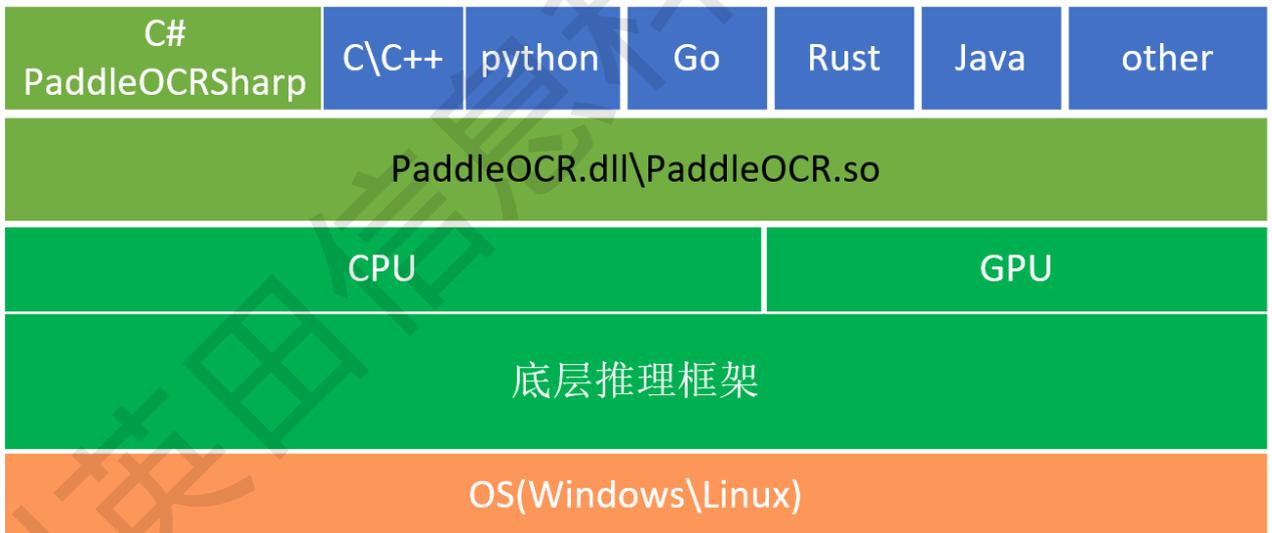
3. 架构设计

3.1. 架构设计

PaddleOCRSharp 库设计采用 P/Invoke 方式调用核心库 PaddleOCR.dll\PaddleOCR.so。采用在初始化 OCR 引擎时动态搜索 PaddleOCR.dll 所在路径，并添加进程环境变量。也可以在初始化前指定 PaddleOCR.dll 所在路径。其他依赖必须与 PaddleOCR.dll 放在同级目录。否则可能出现无法加载 PaddleOCR.dll 的错误。

核心组件 PaddleOCR.dll 有导出标准 C 函数，可以被 C\C++、.NET、Python、Golang、Rust、java、labview、delphi 等众多开发语言的直接加载调用。直接切换 PaddleOCR.dll 及相关依赖文件，即可完成 CPU/GPU 切换或者免费版与付费版的切换。

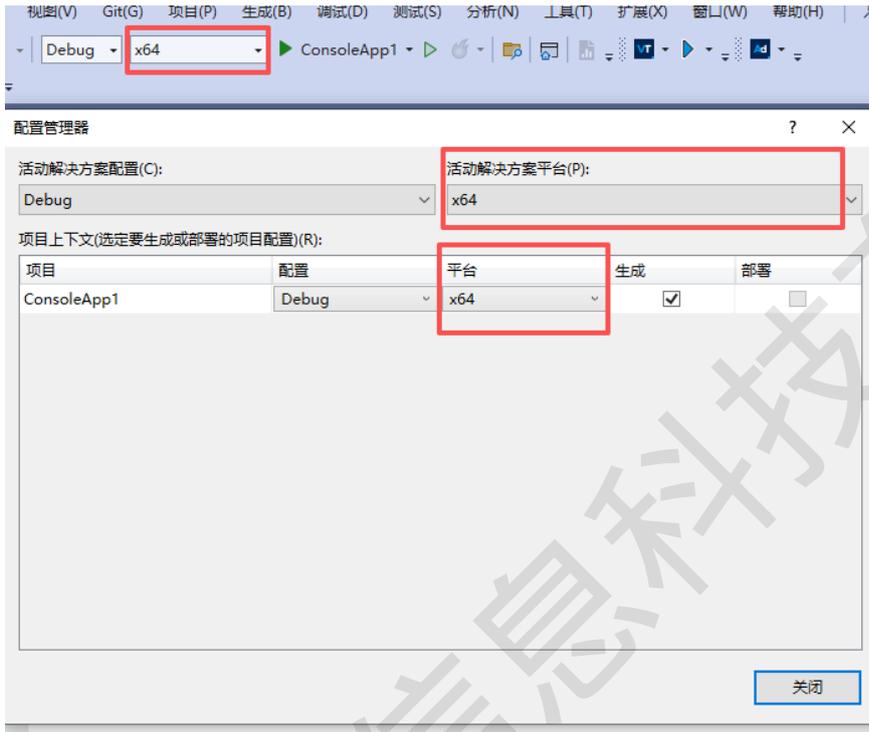
PaddleOCRSharp 架构示意图



4. 快速入门

4.1. C#如何使用

打开 Microsoft Visual Studio，新建一个 C#控制台项目，在配置管理界面，把解决方案和项目目标平台，都设置成 x64。



项目名称右键点击选择【管理 nuget 程序包】，在浏览界面，搜索“PaddleOCRSharp”选择版本 V5.1.0，点击【安装】，搜索 Paddle.Runtime.win_x64 选择版本 V3.1.0.1 点击【安装】。



安装完成或者引用 d11 后，引入命名空间，

第一步初始化 OCR 引擎实例，采用默认参数构造参数。

第二步调用 DetectText 方法，进行文字识别，参数传入一张图片文件的完整路径。

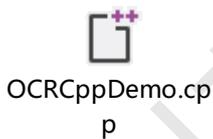
第三步获取识别结果。

第四步释放资源。

```
using PaddleOCRSharp;
PaddleOCREngine engine = new PaddleOCREngine();
var result = engine.DetectText("图片文件");
//显示识别结果
Console.WriteLine(result.Text);
//获取每个文本块内容
foreach (var item in result.TextBlocks)
{
    Console.WriteLine(item.Text);
}
//当很久不用或者需要关闭程序，可以调用 Dispose 方法释放对象。
engine.Dispose();
```

4.2. C++如何使用

打开 Microsoft Visual Studio，新建一个 C++控制台项目，在配置管理界面，把解决方案和项目目标平台，都设置成 x64。配置相关 PaddleOCR 的头文件和 lib 文件引用。代码参考附件。



4.3. Python 如何使用

代码参考附件



5. PaddleOCRSharp 接口说明

5.1. PaddleOCREngine 接口说明

5.1.1. 构造函数

PaddleOCREngine()

- **说明:** 使用默认参数初始化 OCR 引擎对象。
- **参数:** 无。
- **异常:** 若初始化失败, 抛出 `Exception`。

PaddleOCREngine(OCRModelConfig config)

- **说明:** 使用指定的模型配置对象初始化 OCR 引擎对象, 识别参数使用默认值。
- **参数:** `config`: 模型配置对象 (`OCRModelConfig`), 若为 `null` 则使用默认配置。
- **异常:** 若初始化失败, 抛出 `Exception`。

PaddleOCREngine(OCRModelConfig config, OCRParameter parameter)

- **说明:** 使用指定的模型配置和识别参数初始化 OCR 引擎对象。
- **参数:**
 - `config`: 模型配置对象 (`OCRModelConfig`), 若为 `null` 则使用默认配置。
 - `parameter`: 识别参数 (`OCRParameter`), 若为 `null` 则使用默认参数。
- **异常:** 若初始化失败, 抛出 `Exception`。

PaddleOCREngine(OCRModelConfig config, string parameterjson)

- **说明:** 使用指定的模型配置和 JSON 格式的识别参数字符串初始化 OCR 引擎对象。
- **参数:**
 - `config`: 模型配置对象 (`OCRModelConfig`), 若为 `null` 则使用默认配置。
 - `parameterjson`: 识别参数的 JSON 字符串。若为空或 `null`, 则尝试从默认配置文件 `"inference/PaddleOCR.config.json"` 读取。
- **异常:** 若初始化失败, 抛出 `Exception`。

5.1.2. 文本识别

以下方法均返回 `OCRResult` 对象, 包含识别出的文本块列表 (`TextBlocks`) 和原始 JSON 文本 (`JsonText`)。若识别过程出错, 将抛出 `Exception`。

OCRResult DetectText(IntPtr p_cvmat)

- **说明:** 对 OpenCV 的 `Mat` 对象指针进行文本识别。
- **参数:**
 - `p_cvmat`: 指向 `Mat` 对象的指针 (`IntPtr`)。
- **返回:** `OCRResult` 识别结果。
- **异常:** 若 `p_cvmat` 为 `IntPtr.Zero`, 抛出 `ArgumentNullException`。

OCRResult DetectText(string imagefile)

- **说明:** 对图像文件进行文本识别。
- **参数:**

- `imagefile`: 图像文件路径 (字符串)。
- **返回**: `OCRResult` 识别结果。
- **异常**: 若文件路径为空或文件不存在, 抛出 `ArgumentNullException` 或 `Exception`。

`OCRResult DetectText(Bitmap image)`

- **说明**: 对 `System.Drawing.Bitmap` 图像对象进行文本识别。
- **参数**:
 - `image`: `Bitmap` 图像对象。
- **返回**: `OCRResult` 识别结果。
- **异常**: 若 `image` 为 `null`, 抛出 `ArgumentNullException`。

`OCRResult DetectText(byte[] imagebyte)`

- **说明**: 对图像字节数组进行文本识别。
- **参数**:
 - `imagebyte`: 图像字节数组。
- **返回**: `OCRResult` 识别结果。
- **异常**: 若 `imagebyte` 为 `null`, 抛出 `ArgumentNullException`。

`OCRResult DetectTextBase64(string imagebase64)`

- **说明**: 对 Base64 编码的图像字符串进行文本识别。
- **参数**:
 - `imagebase64`: Base64 编码的图像字符串。
- **返回**: `OCRResult` 识别结果。
- **异常**: 若 `imagebase64` 为空或 `null`, 抛出 `ArgumentNullException`。

`OCRResult DetectText(IntPtr imgPtr, int nWidth, int nHeight, int nChannel)`

- **说明**: 对原始图像内存数据进行文本识别。
- **参数**:
 - `imgPtr`: 图像数据的内存地址指针 (`IntPtr`)。
 - `nWidth`: 图像宽度 (像素)。
 - `nHeight`: 图像高度 (像素)。
 - `nChannel`: 图像通道数 (通常为 3 或 1)。
- **返回**: `OCRResult` 识别结果。
- **异常**: 若 `imgPtr` 为 `IntPtr.Zero`, 抛出 `ArgumentNullException`。

5.1.3. 资源释放

1. `void Dispose()`

- **说明**: 释放引擎占用的非托管资源。调用后引擎实例将不可用。
- **参数**: 无。
- **注意**: 不能使用 `using` 自动释放。

5.2. 类说明

5.2.1. OCRModelConfig 模型配置对象

属性名称	类型	默认值	说明
det_infer	string		文本检测 det_infer 模型路径
cls_infer	string		文本方向检测 cls_infer 模型路径
rec_infer	string		文本识别 rec_infer 模型路径
keys	string		识别字典文件名全路径

5.2.2. OCRParameter 参数对象

属性名称	类型	默认值	说明
use_gpu	bool	false	是否使用 GPU, 免费版此属性无效。GPU 版本默认为 true
gpu_id	int	0	GPU id, 使用 GPU 时有效, windows 下只能是 0
gpu_mem	int	4000	申请的 GPU 内存
cpu_math_library_num_threads	int	10	CPU 预测时的线程数, 在机器核数充足的情况下, 该值越大, 预测速度越快
enable_mkldnn	bool	true	是否使用 mkldnn 库
det	bool	true	是否执行文字检测, 单行文本可以关闭该参数来提高速度
rec	bool	true	是否执行文字识别
cls	bool	false	是否执行文字方向分类
max_side_len	int	960	输入图像长宽大于 960 时, 等比例缩放图像, 使得图像最长边为 960。适当调小可以加快速度, 但影响精度
det_db_thresh	float	0.3	用于过滤 DB 预测的二值化图像, 设置为 0. -0.3 对结果影响不明显
det_db_box_thresh	float	0.5	DB 后处理过滤 box 的阈值, 如果检测存在漏框情况, 可酌情减小
det_db_unclip_ratio	float	1.6	表示文本框的紧致程度, 越小则文本框更靠近文本
use_dilation	bool	false	是否在输出映射上使用膨胀
det_db_score_mode	bool	true	true:使用多边形框计算 bbox score, false:使用矩形框计算。矩形框计算速度更快, 多边形框对弯曲文本区域计算更准确
visualize	bool	false	是否对结果进行可视化, 为 true 时, 预测结果会在当前目录下保存一个 ocr_vis.png 文件。默认 false
use_angle_cls	bool	false	是否使用方向分类器, 已弃用, 以 cls 为准
cls_thresh	float	0.9	方向分类器的得分阈值
cls_batch_num	int	1	方向分类器 batchsize
rec_batch_num	int	6	识别模型 batchsize, 适当调大, 可以加快速度
rec_img_h	int	48	识别模型输入图像高度, v2 模型需要设置为 32

rec_img_w	int	320	识别模型输入图像宽度
show_img_vis	bool	false	是否显示预测结果
use_tensorrt	bool	false	使用 GPU 预测时，是否启动 tensorrt，默认 false

5.2.3. ModifyParameter 动态修改参数对象

属性名称	类型	默认值	说明
m_det	bool	true	动态修改是否检测。在 OCRParameter.det=true 时有效
m_rec	bool	true	动态修改是否识别。在 OCRParameter.rec=true 时有效
m_max_side_len	int	960	输入图像长宽大于 960 时，等比例缩放图像，使得图像最长边为 960,；默认 960, 当 m_det=true 时有效
m_det_db_thresh	float	0.3	用于过滤 DB 预测的二值化图像，设置为 0.-0.3 对结果影响不明显，当 m_det=true 时有效
m_det_db_box_thresh	float	0.5	DB 后处理过滤 box 的阈值，如果检测存在漏框情况，可酌情减小，当 m_det=true 时有效
m_det_db_unclip_ratio	float	1.6	表示文本框的紧致程度，越小则文本框更靠近文本，当 m_det=true 时有效

5.2.4. OCRResult 文字识别结果对象

属性名称	类型	说明
TextBlocks	List	文本块 TextBlock 对象列表，由 JsonText 反序列化得到
Text	string	TextBlocks 所有文本块的字符串拼接字符串，没有换行符
JsonText	string	非托管代码返回的原始 json 格式字符串

5.2.5. TextBlock 文本框对象

属性名称	类型	说明
BoxPoints	List	文本块四周点坐标列表，左上、右上、右下、左下四个点
Text	string	文本块识别的文本
Score	float	文本识别置信度
cls_score	float	角度分类置信度，oCRParameter 参数 cls 和 use_angle_cls 同时为 true 有效，否则返回 0
cls_label	float	角度分类标签，oCRParameter 参数 cls 和 use_angle_cls 同时为 true 有效，否则返回-1

5.2.6. OCRPoint 文本坐标对象

属性名称	类型	说明
X	int	X 坐标, 单位像素
Y	int	Y 坐标, 单位像素

5.3. 内部逻辑机制

初始化: 构造函数通过调用非托管 DLL 函数 `Initialize` 或 `Initializejson` 创建推理引擎实例, 并将返回的句柄存储在 `enginePtr` 中。

识别过程: 所有 `DetectText` 重载方法最终调用相应的非托管函数 (如 `DetectMat`、`DetectByte` 等) 进行推理, 并将返回的指针结果通过 `ConvertResult` 方法解析为 `OCRResult` 对象。

结果解析: `ConvertResult` 方法负责将非托管函数返回的指针 (指向包含识别结果的字符串) 转换为托管对象。在 Windows 环境下使用 `Marshal.PtrToStringUni`, 在非 Windows 环境下使用 `Marshal.PtrToStringAnsi` 进行字符串转换, 以确保中文字符的正确解析。

表格识别原理: `DetectStructure` 方法在基础文本识别结果上, 通过分析文本块的坐标位置, 计算行和列的分割阈值, 从而将文本块组织成表格结构。

6. PaddleOCR 接口说明

6.1. 接口说明

以下是 PaddleOCR 库的 C++ 接口详细说明文档。该接口主要用于为其他编程语言（如.NET、Python、Go、Rust 等）提供底层调用支持。

6.1.1. 初始化与配置

```
void libaddLicense(char* licfile);
```

- **功能:** 指定许可授权文件（仅 OEM 版本使用）。
- **参数:**
 - `licfile`: 许可证文件路径。
- **返回值:** 无。

```
uintptr_t Initialize(char* det_infer, char* cls_infer, char* rec_infer, char* keys,
OCRParameter parameter);
```

- **功能:** 通过模型路径和 `OCRParameter` 对象初始化 OCR 引擎。
- **参数:**
 - `det_infer`: 文本检测模型路径。
 - `cls_infer`: 文本方向分类模型路径。
 - `rec_infer`: 文本识别模型路径。
 - `keys`: 识别模型对应的字典文件路径。
 - `parameter`: `OCRParameter` 对象，包含识别参数。
- **返回值:** 成功返回 OCR 引擎对象指针 (`uintptr_t`)，失败返回 0。失败时可通过 `GetError()` 获取错误信息。

```
uintptr_t Initializejson(char* modelPath_det_infer, char* modelPath_cls_infer, char*
modelPath_rec_infer, char* keys, char* parameterjson);
```

- **功能:** 通过模型路径和 JSON 格式的参数字符串初始化 OCR 引擎。
- **参数:**
 - `modelPath_det_infer`: 文本检测模型路径。
 - `modelPath_cls_infer`: 文本方向分类模型路径。
 - `modelPath_rec_infer`: 文本识别模型路径。
 - `keys`: 识别模型对应的字典文件路径。
 - `parameterjson`: `OCRParameter` 的 JSON 格式字符串。
- **返回值:** 成功返回 OCR 引擎对象指针 (`uintptr_t`)，失败返回 0。失败时可通过 `GetError()` 获取错误信息。

```
void EnableANSIResult(bool enable);
```

- **功能:** 设置是否使用 ANSI 编码返回结果。适用于 Python、Go、Linux 等环境。
- **参数:**
 - `enable`: `true` 表示使用 ANSI 编码格式返回结果，默认为 `false`。
- **返回值:** 无。

```
void EnableJsonResult(uintptr_t engine_p, bool enable);
```

- **功能:** 设置是否使用 JSON 格式返回识别结果。
 - **参数:**
 - `engine_p`: OCR 引擎对象指针, 由 `Initialize` 或 `Initializejson` 函数返回。
 - `enable`: `true` 表示返回 JSON 格式字符串, 默认为 `true`。
 - **返回值:** 无。
- void libEnableDetUseRect(uintptr_t engine_p, bool enable);**
- **功能:** 设置是否使用矩形检测框。单个文字容易检测成菱形, 此选项可将其处理成正矩形, 以提高水平文字的识别准确度。
 - **参数:**
 - `engine_p`: OCR 引擎对象指针。
 - `enable`: `true` 表示启用矩形检测框, 默认为 `false`。
 - **返回值:** 无。
- bool libModifyParameter(uintptr_t engine_p, ModifyParameter parameter);**
- **功能:** 在引擎实例化后动态修改识别参数。
 - **参数:**
 - `engine_p`: OCR 引擎对象指针。
 - `parameter`: `ModifyParameter` 对象, 包含可修改的参数。
 - **返回值:** 修改成功返回 `true`, 失败返回 `false`。失败时可通过 `GetError()` 获取错误信息。

6.1.2. 文本识别

以下所有识别函数均返回一个 `char*` 指针, 指向包含识别结果的字符串 (格式取决于 `EnableJsonResult` 和 `EnableANSIResult` 的设置)。调用者需负责释放该指针指向的内存 (通常由调用方语言的绑定库处理, 调用 **FreeEngine** 方法进行释放)。

char* DetectMat(uintptr_t engine, cv::Mat& cvmat);

- **功能:** 对 OpenCV 的 `Mat` 图像对象进行文本识别。
- **参数:**
 - `engine`: OCR 引擎对象指针。
 - `cvmat`: OpenCV 的 `Mat` 图像对象引用。
- **返回值:** 识别结果字符串指针。

char* Detect(uintptr_t engine, char* imagefile);

- **功能:** 对图像文件进行文本识别。
- **参数:**
 - `engine`: OCR 引擎对象指针。
 - `imagefile`: 图像文件路径字符串。
- **返回值:** 识别结果字符串指针。

char* DetectByte(uintptr_t engine, char* imagebytedata, size_t* size);

- **功能:** 对图像字节数组进行文本识别。
- **参数:**
 - `engine`: OCR 引擎对象指针。
 - `imagebytedata`: 图像字节数组的指针。
 - `size`: 指向图像字节数组大小的指针。
- **返回值:** 识别结果字符串指针。

char* DetectBase64(uintptr_t engine, char* imagebase64);

- **功能:** 对 Base64 编码的图像字符串进行文本识别。
- **参数:**
 - engine: OCR 引擎对象指针。
 - imagebase64: Base64 编码的图像字符串。
- **返回值:** 识别结果字符串指针。

char* DetectByteData(uintptr_t engine, char* img, int nWidth, int nHeight, int nChannel);

- **功能:** 对原始图像内存数据进行文本识别。
- **参数:**
 - engine: OCR 引擎对象指针。
 - img: 图像数据的内存地址指针。
 - nWidth: 图像宽度 (像素)。
 - nHeight: 图像高度 (像素)。
 - nChannel: 图像通道数 (通常为 3 或 1)。
- **返回值:** 识别结果字符串指针。

6.1.3. 资源释放

- **void FreeEngine(uintptr_t engine);**
 - **功能:** 释放指定的 OCR 引擎对象及其占用的资源。
 - **参数:**
 - engine: 要释放的 OCR 引擎对象指针。
 - **返回值:** 无。

7. 常见问题与解决方案

前往在线网址查看：[广州英田信息科技有限公司 常见问题与解决方案](#)

8. 示例项目

详见示例文件夹 sample。

9. 其他

公司官网：[广州英田信息科技有限公司](#)

技术博客：[英田科技-明月心 - 博客园](#)

B 站：[明月心技术学堂-哔哩哔哩视频](#)

PaddleOCRSharp 开源网址：

Gitee: <https://gitee.com/raoyutian/PaddleOCRSharp>

Github: <https://github.com/raoyutian/PaddleOCRSharp>

扫码关注公众号，可以了解更多的技术文章

扫码加入 QQ 群，可以了解更多的技术和资源文件。



群名称:明月心技术学堂
群 号:318860399