火眼臻睛
车牌识别系统

使用和编程接口说明

2013 年 10 月

一、火眼臻睛车牌识别系统软件介绍

火眼臻睛车牌识别系统是软件形式的汽车牌照识别产品，采用动态连接库(DLL)，可嵌入到用户应用程序中实现车牌识别功能。

火眼臻睛车牌识别系统，通过自主研发的车牌识别算法，能对车牌的大小，污损，边框，和倾斜度变化等有极强的适应能力。 达到对小车牌（宽度>45 个像素）、污损车牌、对比度低车牌、各式边框车牌、多角度拍摄的车牌保持高准确度和识别率，识别技术和效果处于国际先进水平。

车牌识别 Sdk 识别效果展示

核心功能：

- 1. SDK 开发，算法稳定可靠，场景适应能力强，系统集成极其简单。
- 2. 整牌识别率高，识别率达 98.5%以上。
- 3. 识别的最小车牌，达到 45 个像素宽度。

运行环境：

- Windows、linux、arm-linux 等。

1.1 图片识别

（1）图片识别包括以下内容：

1）车牌号码；

2）车牌颜色；

3）车牌类型；

4）车牌宽度；

5）车牌识别可信度；

（2）支持单张图片识别多车牌。

（3）支持高清图片。

1.2 识别图片种类

1）普通蓝牌；

2）普通黑牌；

3）普通黄牌；

4）双层黄牌；

5）教练车牌；

6）警车车牌；

7）新式单层武警车牌；

8）新式双层武警车牌；

9）新式单层军牌；

10）使馆车牌；

1.3 相关文件

1） LPKernelEx.dll 识别核心库；

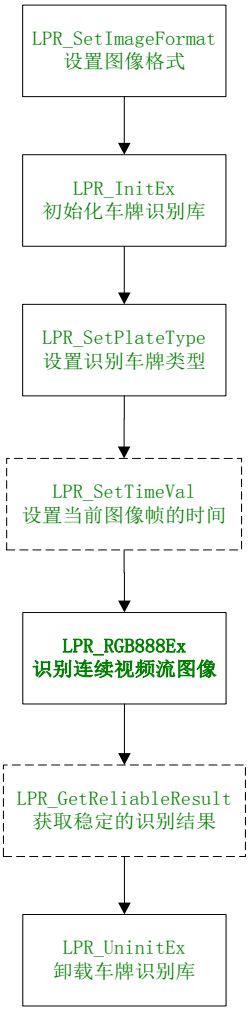
2） LPKernelEx.lib 动态 lib 文件；

3） LPKernelEx.h 调用接口说明；

二. 识别软件函数调用过程

1.1 流程

视频流识别调用流程：



图片识别调用流程：

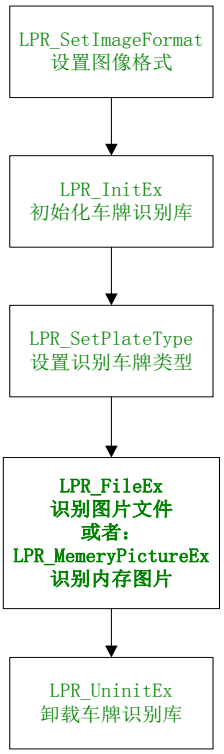


图 1.视频流和图片识别调用流程（实现框是必要过程，虚线框表示可选过程）

主要函数说明:
LPR_SetImageFormat：设置图像格式。必须在调用 LPR_InitEx 之前设置。失败，返回 0。
LPR_InitEx：初始化车牌识别库。失败，返回 0。
LPR_SetPlateType：设置识别车牌类型。必须在调用 LPR_InitEx 之后设置。失败，返回 0。
LPR_RGB888Ex：**识别连续视频流**。失败，返回 0。
LPR_FileEx：**识别图片文件**。失败，返回 0。
LPR_MemeryPictureEx：识别内存图片。失败，返回 0。
LPR_UninitEx：卸载车牌识别库，退出 SDK 时调用。失败，返回 0。
LPR_SetTimeVal：视频流识别时，设置当前图像帧的时间。
LPR_GetReliableResult：视频识别后获取稳定的识别结果。可以作为车辆抓拍使用。

三. 编程示例

下面是进行单路图片识别的参考代码。

```
#include <stdio.h>
#include <windows.h>
#include "LPKernelEx.h"
int main(int argc, char* argv[])
{
    int b;
    if(argc<=1)
    {
        printf("\n输入图片路径\n");
        return -1;
    }else
    {
        printf("\n图片路径：%s\n",argv[1]);
    }
    char* image_path = argv[1];
    // 通道号
    int nchannel = 1;
    // 初始化车牌识别
    b =
LPR_SetImageFormat(FALSE,FALSE,ImageFormatBGR,FALSE,80,400,TRUE,FALSE,TRUE,nchannel);
    if(!b) return FALSE;
    if(LPR_InitEx(nchannel)==FALSE)
        return FALSE;
    b = LPR_SetPlateType(FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,nchannel);
    if(!b) return FALSE;
```

```
    TH_PlateResult iresult[100]={0};
    int nRecoNum = 0;
    // 设置车牌识别区域，默认是全图识别
    TH_RECT rcRange = {0,0,0,0}
    // 识别车牌
    b = LPR_FileEx(image_path, NULL, iresult, nRecoNum, &rcRange, nchannel);
    for(int i = 0; i < nRecoNum; i++)
        printf("*** 识别结果:%s\n",iresult[i].license);
    LPR_UninitEx();
}
```

说明：视频流识别，识别函数是 LPR_RGB888Ex，调用流程和图片识别基本相同，需要注意的是，视频流内存格式，需要正确设置。比如，如果是 YUV420 的视频数据，那么在 LPR_SetImageFormat 函数设置视频格式为 ImageFormatYUV420。

更详细的例子，请参见[调用例程]文件夹下的例程。

四、函数说明

4.1 LPR_SetImageFormat 设置图像格式

```
BOOL WINAPI LPR_SetImageFormat(BOOL bMovingImage,
        BOOL bFlipVertical,
        int nColorOrder,
        BOOL bVertCompress,
        int nMinPlateWidth,
        int nMaxPlateWidth,
        BOOL bDwordAligned,
        BOOL bInputHalfHeightImage,
        BOOL bOutputSingleFrame,
        int nChannel=1);
```

参数：	bMovingImage[in]	识别运动或静止图像
	bFlipVertical[in]	是否上下颠倒图像后识别
	nColorOrder[in]	图像格式
	bVertCompress[in]	是否垂直方向压缩一倍识别
	nMinPlateWidth[in]	最小车牌宽度
	nMaxPlateWidth[in]	最大车牌宽度
	bDwordAligned[in]	是否四字节对齐
	bInputHalfHeightImage[in]	是否输入场图像
	bOutputSingleFrame[in]	是否只输出一个识别结果
	nChannel[in]	通道号

此函数在调用 LPR_InitEx 之前进行设置，函数调用成功返回 TRUE，否则返回 FALSE。

4.2 LPR_InitEx 初始化识别库

```
BOOL WINAPI LPR_InitEx(int  nChannel=1);
```

参数：nChannel[in] 通道号

4.3 LPR_UninitEx 初始化识别库

```
BOOL WINAPI LPR_InitEx(int  nChannel=1);
```

参数：nChannel[in] 通道号

函数调用成功返回 TRUE，否则返回 FALSE。

4.4 LPR_FileEx 识别图片文件

```
BOOL WINAPI LPR_FileEx(char* lpszFileName, char *lpszPlateFile, TH_PlateResult* pResult, int &nRecogNum, TH_RECT *prcRange,int nChannel=1);
```

参数：lpszFileName[in]　　图像的路径
　　lpszPlateFile[in]　　车牌的保存路径，如果不保存车牌图片此参数传NULL
　　pResult[in]　　识别结果结构体
　　RecogNum[out]　　实际识别到的车牌个数
　　prcRange[in]　　车牌识别的范围；设为(0,0,0,0)整张图片都识别，以像素为单位
　　nChannel[in]　　通道号

支持BMP、JPG、TIF图像格式，函数调用成功返回TRUE，否则返回FALSE。

4.5 LPR_RGB888Ex 识别视频流图像

```
BOOL WINAPI LPR_RGB888Ex(unsigned char *pImg, int nWidth, int  nHeight, TH_PlateResult* pResult, int &nRecogNum, TH_RECT *prcRange, int  nChannel=1);
```

参数：pImg[in]　　指向内存中图像的指针，格式为RGB888,YUV420,YUV422，格式在LPR_SetImageFormat函数中指定
　　nWidth[in]　　图像的宽度，以像素为单位
　　nHeight[in]　　图像的高度，以像素为单位
　　pResult[in]　　识别结果结构体
　　nRecogNum[out]　实际识别到的车牌个数
　　prcRange[in]　　车牌识别的范围；设为(0,0,0,0)整张图片都识别，以像素为单位
　　nChannel[in]　　通道号

识别连续视频流内存图像。函数调用成功返回TRUE，否则返回FALSE。

4.6 LPR_MemeryPictureEx 识别内存图片。

```
BOOL WINAPI LPR_MemeryPictureEx(unsigned char *pImg, int nWidth, int  nHeight, TH_PlateResult*
```

```
pResult, int &nRecogNum, TH_RECT *prcRange, int  nChannel=1);
```

参数：pImg[in]　　指向单幅图像内存中的指针，格式为RGB888,YUV420,YUV422，格式在LPR_SetImageFormat函数中指定

nWidth[in]　　图像的宽度，以像素为单位

nHeight[in]　　图像的高度，以像素为单位

pResult[in]　　识别结果结构体

nRecogNum[out] 实际识别到的车牌个数

prcRange[in]　　车牌识别的范围；设为(0,0,0,0)整张图片都识别，以像素为单位

nChannel[in]　　通道号

识别内存图片。函数调用成功返回TRUE，否则返回FALSE。

4.7 LPR_GetTotalChannelNum 获取加密狗支持的通道数。

```
int WINAPI LPR_GetTotalChannelNum();
```

参数：返回当前加密狗支持的通道数

获取加密狗支持的通道数。

4.8 LPR_SetTimeVal 视频流识别时，设置当前图像帧的时间。

```
BOOL WINAPI LPR_SetTimeVal(TH_TimeVal& tv_time, int nChannel=1);
```

参数：tv_time[in]　　调用视频流识别时时间

便于视频流识别时，返回最佳识别结果时对应的图像帧的时间。用户可以通过该时间，找到对应的图像帧。

4.9 LPR_GetReliableResult 获取视频流稳定的识别结果。

```
BOOL WINAPI LPR_GetReliableResult(TH_PlateResultImage* pResult, int& nRecogNum, int nChannel);
```

参数：pResult[in]　　识别结果结构体

nRecogNum[out] 实际识别到的车牌个数

nChannel[in]　　通道号

LPR_RGB888Ex接口内部会根据同一辆车多帧的识别情况，给出一个最佳的识别结果，并返回一张最清晰的抓拍图片以及抓拍时刻。

这个最佳的识别结果，通过本接口来获取。

本接口可以作为车辆抓拍使用。

使用方法:此函数需要在LPR_RGB888Ex函数之后调用

五、数据结构说明

5.1 TH_RECT 车牌区域结构体

```
typedef struct TH_RECT
{
  int left;
  int top;
```

```
    int right;
    int bottom;
}TH_RECT;
```

5.2 TH_PlateResult 识别结果结构体

```
typedef struct TH_PlateResult
{
    char license[16];       // 车牌号码
    char color[8];           // 车牌颜色
    int nColor;             // 车牌颜色序号
    int nType;              // 车牌类型
    int nConfidence;        // 车牌可信度
    int nBright;            // 亮度评价
    int nDirection;         // 运动方向，unknown, 1 left, 2 right, 3 up , 4 down
    TH_RECT rcLocation;     //车牌位置
    int nTime;               //识别所用时间
    unsigned char nCarBright;        //车的亮度
    unsigned char nCarColor;         //车的颜色
    char reserved[100];              // 保留
}TH_PlateResult;
```

5.3 TH_PlateResultImage 识别结果结构体，含抓拍的图片

```
typedef struct TH_PlateResultImage
{
    char license[16];                // 车牌号码
    char color[8];                   // 车牌颜色
    int nColor;                      // 车牌颜色序号
    int nType;                       // 车牌类型
    int nConfidence;                 // 车牌可信度
    int nBright;                     // 亮度评价
    int nDirection;                  // 运动方向，unknown, 1 left, 2 right, 3 up , 4 down

    int nTime;                       // 识别所用时间
    unsigned char nCarBright;        // 车的亮度
    unsigned char nCarColor;         // 车的颜色

    unsigned char* pImageRGB24;      // 抓拍识别到车牌的图像,RGB24位格式
    int nImageWidth;                 // 抓拍识别到车牌的图像的宽度
    int nImageHeight;                // 抓拍识别到车牌的图像的高度

    TH_RECT       rcLocation;        // 抓拍车牌所在的位置
    TH_TimeVal  tv_time;             // 抓拍车牌的时间
}TH_PlateResultImage;
```

六、常量定义

6.1 车牌类型(数值)

```
#define LT_UNKNOWN  0   //未知车牌
#define LT_BLUE     1   //蓝牌小汽车
#define LT_BLACK    2   //黑牌小汽车
#define LT_YELLOW   3   //单排黄牌
#define LT_YELLOW2  4   //双排黄牌（大车尾牌，农用车）
#define LT_POLICE   5   //警车车牌
#define LT_ARMPOL   6   //武警车牌
#define LT_INDIVI   7   //个性化车牌
#define LT_ARMY     8   //单排军车牌
#define LT_ARMY2    9   //双排军车牌
#define LT_EMBASSY  10  //使馆车牌
#define LT_HONGKONG 11  //香港进出中国大陆车牌
#define LT_TRACTOR  12  //农用车牌
#define LT_COACH13  //教练车牌
#define LT_MACAO14  //澳门进出中国大陆车牌
#define LT_ARMPOL2   15 //双层武警车牌
```

6.2 车牌颜色(数值)

```
#define LC_UNKNOWN  0  //未知
#define LC_BLUE     1  //蓝色
#define LC_YELLOW   2  //黄色
#define LC_WHITE    3  //白色
#define LC_BLACK    4  //黑色
#define LC_GREEN    5  //绿色
```

6.3 运动方向(数值)

```
#define DIRECTION_LEFT    1  //左
#define DIRECTION_RIGHT   2  //右
#define DIRECTION_UP 3  //上
#define DIRECTION_DOWN    4  //下
```

6.4 图像格式(数值)

```
#define ImageFormatRGB          0       //RGBRGBRGB...
#define ImageFormatBGR          1       //BGRBGRBGR...
#define ImageFormatYUV422       2       //YYYY...UU...VV.. (YV16)
#define ImageFormatYUV420COMPASS 3      //YYYY...UV...     (NV12)
#define ImageFormatYUV420       4       //YYYY...U...V...  (YU12)
#define ImageFormatUYVY         5       //UYVYUYVYUYVY...  (UYVY)
#define ImageFormatNV21         6       //YYYY...VU...     (NV21)
#define ImageFormatYV12         7       //YYYY...VU...     (NV21)
```

```
#define ImageFormatYUYV        8        //YUYVYUYVYUYV...  (YUYV)
```

6.5 车辆颜色(数值)

```
#define LGRAY_DARK        0    //深
#define LGRAY_LIGHT       1    //浅
#define LCOLOUR_WHITE     0    //白
#define LCOLOUR_SILVER    1    //灰(银)
#define LCOLOUR_YELLOW    2    //黄
#define LCOLOUR_PINK      3    //粉
#define LCOLOUR_RED       4    //红
#define LCOLOUR_GREEN     5    //绿
#define LCOLOUR_BLUE      6    //蓝
#define LCOLOUR_BROWN     7    //棕
#define LCOLOUR_BLACK     8    //黑
```