

# 海康威视车牌识别智能 DVS

## SDK 编程指南

### Version1.0

### 2008-9

车牌牌识别智能DVS介绍.....	3
SDK版本更新.....	3
错误代码及说明.....	3
函数调用顺序.....	4
数据类型及结构体定义.....	5
1.1 截图参数数据类型及结构体.....	5
1.1.1_VCA_PICTURE_FORMAT_截图的图片格式.....	5
1.1.2_VCA_PICTURE_MODE_截图的图片模式，决定图片分辨率 .....	5
1.1.3_tagNET_VCA_PICTURE_PARAM截图参数设置结构体 .....	5
1.3 矩形框结构体tagNET_VCA_RECT.....	6
1.4 车牌识别配置参数.....	7
1.4.1_VCA_RECOGNIZE_SCENE_识别场景（低速和高速） .....	7
1.4.2 可动态修改的车牌识别配置参数tagNET_VCA_PLATE_PARAM .....	7
1.4.3 车牌识别信息 子结构tagNET_VCA_PLATEINFO.....	8
1.4.4 车牌识别配置参数tagNET_VCA_PLATECFG .....	8
1.5 车牌识别检测结果参数.....	9
1.5.1_VCA_RECOGNIZE_RESULT_识别结果标志 .....	9
1.5.2_VCA_PLATE_COLOR_车牌颜色.....	10
1.5.3_VCA_PLATE_TYPE_车牌类型 .....	10
1.5.4 返回的车牌信息结构体.....	11
1.5.5 车牌识别检测结果.....	12
函数说明.....	13
2.1 命令码.....	13
2.2 回调及回调设置函数.....	13
2.3 获取设备对应配置的参数.....	14
2.4 设置设备对应配置的参数.....	14
2.5 进行告警上报的布防/撤防.....	15
Func:NET_DVR_SetupAlarmChan.....	15
Func:NET_DVR_CloseAlarmChan.....	15

## 车牌识别智能 DVS 介绍

海康威视车牌识别智能 DVS 是一款先进的智能视频分析产品,基于图像处理、机器视觉、模式识别等理论基础而实现,可输出车牌号码、车牌颜色、车牌类型、车牌位置等车牌信息参数,实现对交通监控场景的智能车牌识别处理。

车牌识别功能以SDK动态链接库形式提供给应用软件开发,并同时附有演示程序及其源码,能有效地缩短应用软件开发周期。

车牌识别SDK包含:

库文件HCNetSDK.dll, playm4.dll, DsSdk.dll;

引入库文件HCNetSDK.lib, playm4.lib, DsSdk.lib;

头文件HCNetSDK.h, plaympeg4.h, DecodeCardSdk.h, DataType.h。

除提供车牌识别功能外,SDK保留针对普通DVS的所有固有功能。实施车牌识别功能前,需对SDK进行初始化等基本处理,具体函数定义请参阅《客户端SDK使用手册V2.20》及demo源码。

### 参数取值归一化说明:

车牌识别 SDK 中涉及长度、宽度等度量参数都需对取值进行归一化处理,标注数据在图像中的百分比大小,数值精度取小数点后三位,视频图像的原始尺寸为 PAL 制 704\*576, NTSC 制 704\*480,数值归一化后图像左上角坐标为(0.000, 0.000),右下角坐标为(1.000, 1.000)。

## SDK 版本更新

版本编号	简要说明 (修改内容)	日期
V1.0	<ul style="list-style-type: none"> <li>支持多种车辆制式,可应用于民用车辆、军车、警车、武警车的车牌识别</li> <li>支持车牌汉字、字母、数字的识别</li> <li>支持车牌颜色的识别</li> </ul>	2008-7-14

## 错误代码及说明

ErrorNumber	Description
NET_DVR_PARAMETER_ERROR	17 参数错误
NET_DVR_NOSUPPORT	23 服务器不支持
NET_DVR_DVROPRATEFAILED	29 DVR 操作失败

## 函数调用顺序

A.

建立告警接收通道 NET_DVR_SetupAlarmChan 设置告警回调函数 NET_DVR_SetDVRMessCallBack
--

B.

设置车牌识别参数，置“使能”，开始车牌识别 NET_DVR_SetDVRConfig(NET_DVR_SET_PLATECFG)
---

C.

获取车牌识别检测结果 车牌识别回调函数 fMessCallBack
--------------------------------------

D.

设置车牌识别参数，置“非使能”，停止车牌识别 NET_DVR_SetDVRConfig(NET_DVR_SET_PLATECFG)
--

E.

停止告警接收通道 NET_DVR_CloseAlarmChan
---------------------------------

## 数据类型及结构体定义

Macros		Description
MAX_CHAR_NUM	16	单个车牌最大的允许字符数
MAX_PLATE_NUM	3	单个视频帧中最大的允许识别车牌数

### 1.1 截图参数数据类型及结构体

#### 1.1.1 VCA\_PICTURE\_FORMAT\_截图的图片格式

```
typedef enum _VCA_PICTURE_FORMAT_
{
    PICTURE_NONE = 0,
    PICTURE_JPEG = 1,
    PICTURE_YUV420 = 2,
}VCA_PICTURE_FORMAT;
```

##### Constants

PICTURE\_NONE

不截图

PICTURE\_JPEG

截图格式为 JPEG

PICTURE\_YUV420

截图格式为 YUV420

#### 1.1.2 VCA\_PICTURE\_MODE\_截图的图片模式，决定图片分辨率

```
typedef enum _VCA_PICTURE_MODE_
{
    PICTURE_CIF = 0,
    PICTURE_QCIF = 1,
    PICTURE_4CIF = 2,
}VCA_PICTURE_MODE;
```

##### Constants

PICTURE\_CIF

截图尺寸为 CIF 大小

PICTURE\_QCIF

截图尺寸为 QCIF 大小

PICTURE\_4CIF

截图尺寸为 4CIF 大小

#### 1.1.3 tagNET\_VCA\_PICTURE\_PARAM 截图参数设置结构体

```
typedef struct tagNET_VCA_PICTURE_PARAM
{
    VCA_PICTURE_FORMAT    ePicFormat;
    DWORD    dwCompressQuality;
    VCA_PICTURE_MODE    ePicRes;
```

```
} NET_VCA_PLATE_PARAM, *LPNET_VCA_PLATE_PARAM;
```

### Members

ePicFormat

截图的图片格式

dwCompressQuality

如果截图格式选择为 JPEG，设置 JPEG 图像的质量，取值范围：<低>0~100<高>

ePicRes

截图的图片模式

### 1.2 返回的图片信息结构体

```
typedef struct tagNET_VCA_PICTURE
{
    DWORD dwWidth;
    DWORD dwHeight;
    VCA_PICTURE_FORMAT ePicFormat;
    DWORD dwRes[2];
    DWORD dwPicDataLen;
    BYTE *pImage;
} NET_VCA_PICTURE_INFO, *LPNET_VCA_PICTURE_INFO;
```

### Members

dwWidth

返回的图片宽度

dwHeight

返回的图片高度

ePicFormat

返回图片格式，目前支持 YUV 和 JPEG

dwRes [2]

保留参数

dwPicDataLen

返回的图片数据长度

pImage

指向图片的指针

### 1.3 矩形框结构体 tagNET\_VCA\_RECT

```
typedef struct tagNET_VCA_RECT
{
    float fX;
    float fY;
    float fWidth;
    float fHeight;
} NET_VCA_RECT, *LPNET_VCA_RECT;
```

### Members

fX

边界框左上角点的 X 轴坐标

fY

边界框左上角点的 Y 轴坐标

fWidth

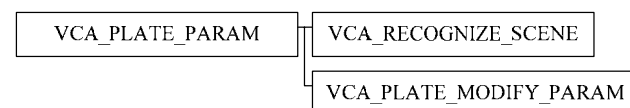
边界框的宽度

fHeight

边界框的高度

**Remarks**

矩形区域结构体主要用于设定搜索范围和描述车牌位置，参数取值归一化，取值精度为小数点后三位。

**1.4 车牌识别配置参数****1.4.1\_VCA\_RECOGNIZE\_SCENE\_识别场景（低速和高速）**

```
typedef enum _VCA_RECOGNIZE_SCENE_
{
    VCA_LOW_SPEED_SCENE = 0,
    VCA_HIGH_SPEED_SCENE = 1,
}VCA_RECOGNIZE_SCENE;
```

**Constants**

VCA\_LOW\_SPEED\_SCENE

低速行驶车辆场景（时速≤40 公里/小时），如收费站、小区工厂入口处、停车场等

VCA\_HIGH\_SPEED\_SCENE

高速行驶车辆场景（时速≥40 公里/小时），如治安卡口、高速公路、移动稽查等

**Remarks**

也可根据车牌在场景中出现的帧数来判断场景属于高速还是低速，当车牌在场景中的帧数大于 10 帧，作为低速场景处理；当车牌在场景中的帧数小于 10 帧，作为高速场景处理。

**1.4.2 可动态修改的车牌识别配置参数 tagNET\_VCA\_PLATE\_PARAM**

```
typedef struct tagNET_VCA_PLATE_PARAM
{
    NET_VCA_PICTURE_PARAM struPictureParam;
    NET_VCA_RECT          struSearchRect;
    NET_VCA_RECT          struInvalidateRect;
    DWORD dwRes[4];
}NET_VCA_PLATE_PARAM, *LPNET_VCA_PLATE_PARAM;
```

**Members**

struPictureParam

截图参数设置

struSearchRect

搜索区域，参数取值归一化到 0.000-1.000 之间的浮点数，精度为小数点后三位。设置搜索区域可以排除一些明显的干扰，如位于图像中的其他字符，可能会对识别形成干扰，在这些情况下，可以设置搜索区域提高识别准确度

#### **struInvalidateRect**

无效区域，参数取值归一化到 0.000-1.000 之间的浮点数，精度为小数点后三位。在具体的应用场景下，某些区域肯定不会出现车牌，则可以用这个区域来屏蔽。invalidateRect 与 searchRect 组合使用，可以组成一个不规则的车牌搜索区域

#### **dwRes [4]**

保留参数

### **1.4.3 车牌识别信息 子结构 tagNET\_VCA\_PLATEINFO**

```
typedef struct tagNET_VCA_PLATEINFO
{
    VCA_RECOGNIZE_SCENE eRecogniseScene;
    NET_VCA_PLATE_PARAM struModifyParam;
    DWORD dwPicProType;
    DWORD dwRes[3];
}NET_VCA_PLATEINFO, *LPNET_VCA_PLATEINFO;
```

#### **Members**

eRecogniseScene

识别场景

struModifyParam

可动态修改的车牌识别配置参数

dwPicProType

图片处理方式 0-不处理 1-上传

reserved[3]

保留参数

#### **Remarks**

与“可动态修改的车牌识别配置参数”相比，“车牌识别配置参数”多一个“识别场景”的配置参数，不支持动态修改，必须在开启车牌识别时进行设置。

### **1.4.4 车牌识别配置参数 tagNET\_VCA\_PLATECFG**

```
typedef struct tagNET_VCA_PLATECFG
{
    DWORD    dwSize;
    DWORD    dwEnable;
    NET_VCA_PLATEINFO struPlateInfo;
    NET_DVR_HANDLEEXCEPTION struHandleType;
    NET_DVR_SCHEDULETIME struAlarmTime[MAX_DAYS][MAX_TIMESEGMENT];
    BYTE byRelRecordChan[MAX_CHANNUM];
}NET_VCA_PLATECFG, *LPNET_VCA_PLATECFG;
```

#### **Members**

dwSize

结构长度

dwEnable



是否启用视频车牌识别 0—否 1—是

struHandleType  
告警处理方式

struAlarmTime  
告警布防时间

byRelRecordChan  
告警触发的录象通道

Remarks

告警处理方式当中选中上传中，告警信息才会上报到 sdk。布防开启后只在布防时间段内才会有效。

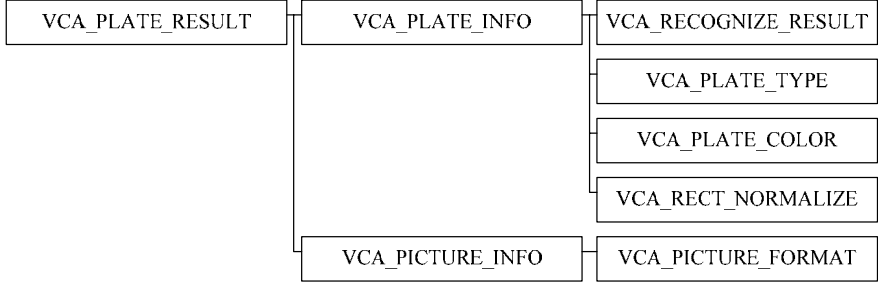
告警处理方式及告警布防时间结构如下（详细请参阅《客户端 SDK 使用手册 V2.20》）：

```
typedef struct
{
    DWORD dwHandleType;           /*处理方式,处理方式的"或"结果*/
                                   /*0x00: 无响应*/
                                   /*0x01: 监视器上警告*/
                                   /*0x02: 声音警告*/
                                   /*0x04: 上传中心*/
                                   /*0x08: 触发报警输出*/
                                   /*0x10: 触发抓图*/

    BYTE byRelAlarmOut[MAX_ALARMOUT]; //报警触发的输出通道,报警触发的输出,
    //为 1 表示触发该输出
}NET_DVR_HANDLEEXCEPTION;

typedef struct{
    //开始时间
    BYTE byStartHour;
    BYTE byStartMin;
    //结束时间
    BYTE byStopHour;
    BYTE byStopMin;
}NET_DVR_SCHEDTIME,*LPNET_DVR_SCHEDTIME;
```

1.5 车牌识别检测结果参数



1.5.1\_VCA\_RECOGNIZE\_RESULT\_识别结果标志

```
typedef enum _VCA_RECOGNIZE_RESULT_
```

```
{
    VCA_RECOGNIZE_FAILURE = 0,
    VCA_IMAGE_RECOGNIZE_SUCCESS,
    VCA_VIDEO_RECOGNIZE_SUCCESS_OF_BEST_LICENSE,
    VCA_VIDEO_RECOGNIZE_SUCCESS_OF_NEW_LICENSE
}VCA_RECOGNIZE_RESULT;
```

**Constants**

VCA\_RECOGNIZE\_FAILURE

识别失败，没有识别结果产生

VCA\_IMAGE\_RECOGNIZE\_SUCCESS

单帧图像识别模式：识别成功（1.0 版本不支持单帧图像模式）

VCA\_VIDEO\_RECOGNIZE\_SUCCESS\_OF\_BEST\_LICENSE

连续视频识别模式：识别到更优结果，对于同一个车牌，更优结果表示当前结果比前一结果的准确度或者置信度更高，建议输出车牌识别结果时选用更优结果

VCA\_VIDEO\_RECOGNIZE\_SUCCESS\_OF\_NEW\_LICENSE

连续视频模式：识别到新的车牌结果，表示视频中有新的车牌出现

**Remarks**

车牌识别分为单帧图像识别和连续视频识别两种模式。前者一次只处理一帧图像，前后处理的两帧图像没有相关性；后者指对连续视频进行识别统计，前后两帧图像识别是相关的，会在适当的时候输出识别结果。1.0 版本 SDK 不支持单帧图像模式。

**1.5.2\_VCA\_PLATE\_COLOR\_车牌颜色**

```
typedef enum _VCA_PLATE_COLOR_
{
    VCA_BLUE_PLATE= 0,
    VCA_YELLOW_PLATE,
    VCA_WHITE_PLATE ,
    VCA_BLACK_PLATE
}VCA_PLATE_COLOR;
```

**Constants**

VCA\_BLUE\_PLATE

蓝色车牌

VCA\_YELLOW\_PLATE

黄色车牌

VCA\_WHITE\_PLATE

白色车牌

VCA\_BLACK\_PLATE

黑色车牌

**1.5.3\_VCA\_PLATE\_TYPE\_车牌类型**

```
typedef enum _VCA_PLATE_TYPE_
{
    VCA_STANDARD92_PLATE= 0,
    VCA_STANDARD02_PLATE,
```

```
VCA_WJPOLICE_PLATE,
VCA_JINGCHE_PLATE,
VCA_STANDARD92_BACK_PLATE
}VCA_PLATE_TYPE;
```

### Constants

VCA\_STANDARD92\_PLATE  
标准民用车与军车

VCA\_STANDARD02\_PLATE  
02 式民用车牌

VCA\_WJPOLICE\_PLATE  
武警车

VCA\_JINGCHE\_PLATE  
警车

VCA\_STANDARD92\_BACK\_PLATE  
民用车双行尾牌

### 1.5.4 返回的车牌信息结构体

```
typedef struct tagNET_VCA_PLATE_INFO
{
    VCA_RECOGNIZE_RESULT eResultFlag;
    VCA_PLATE_TYPE ePlateType;
    VCA_PLATE_COLOR ePlateColor;
    NET_VCA_RECT struPlateRect;
    DWORD dwRes[4];
    DWORD dwLicenseLen;
    char *pLicense;
    char *pBelieve;
}NET_VCA_PLATE_INFO, *LPNET_VCA_PLATE_INFO;
```

### Members

**eResultFlag;**  
识别结果标志，  
单帧图像识别模式: 0-识别失败;1-图像识别成功（**1.0 版本不支持单帧图像识别模式**）  
连续视频识别模式: 0-识别失败;2-视频识别更优结果;3-视频识别新的车辆识别结果

**ePlateType**  
车牌类型

**ePlateColor**  
车牌颜色

**struPlateRect**  
车牌在视频图像中的位置，参数取值归一化，取值精度为小数点后三位

**dwRes [4]**  
保留

**dwLicenseLen**

车牌长度

**pLicense**

车牌号码

**pBelieve**

各个识别字符的置信度，如检测到车牌"浙 A12345",置信度为 10,20,30,40,50,60,70，则表示"浙"字正确的可能性只有 10%，"A"字的正确的可能性是 20%

### 1.5.5 车牌识别检测结果

```
typedef struct tagNET_VCA_PLATE_RESULT
{
    DWORD      dwSize;
    DWORD      dwPlateNum;
    LPNET_VCA_PLATE_INFO  pPlateInfo;
    DWORD dwRes[4];
    NET_VCA_PICTURE_INFO struPictureInfo;
}NET_VCA_PLATE_RESULT, *LPNET_VCA_PLATE_RESULT;
```

**Members****dwSize**

结构的大小

**dwPlateNum**

视频图像内的车牌个数

**pPlateInfo**

返回的车牌信息结构体

**dwRes [4]**

保留参数

**struPictureInfo**

返回的图片信息结构体

## 函数说明

### 2.1 命令码

配置参数命令

//车牌识别对应 NET\_VCA\_PLATE\_CFG

#define NET\_DVR\_SET\_PLATECFG 150 //设置车牌识别参数

#define NET\_DVR\_GET\_PLATECFG 151 //获取车牌识别参数

回调命令码

//对应 NET\_VCA\_PLATE\_RESULT

#define COMM\_ALARM\_PLATE 0x1101 //车牌识别报警信息

### 2.2 回调及回调设置函数

```
BOOL NET_DVR_SetDVRMessCallBack(  
    BOOL (CALLBACK *fMessCallBack)(  
        LONG lCommand,  
        char *sDVRIP,  
        char *pBuf,  
        DWORD dwBufLen)  
);
```

#### Parameters

fMessCallBack

[in]回调实现函数地址

lCommand

[in]回调命令码，车牌识别为 **COMM\_ALARM\_PLATE**

sDVRIP

[in]设备 ip 地址

pBuf

[in]存放信息的缓冲区，

dwBufLen

[in] 缓冲区的大小

#### Return Values

成功返回 TRUE

失败返回 FALSE，通过 NET\_DVR\_GetLastError 获取错误码，详细请参阅《客户端 SDK 使用手册 V2.20》。

#### Remarks

NET\_DVR\_SetDVRMessCallBack\_EX ， NET\_DVR\_SetDVRMessCallBack\_NEW ，  
NET\_DVR\_SetDVRMessageCallBack

也支持车牌告警信息回调。详细请参阅《客户端 SDK 使用手册 V2.20》。

## 2.3 获取设备对应配置参数

```
BOOL NET_DVR_GetDVRConfig(  
    LONG    IUserID,  
    DWORD   dwCommand,  
    LONG    IChannel,  
    LPVOID   lpOutBuffer,  
    DWORD   dwOutBufferSize,  
    LPDWORD lpBytesReturned  
);
```

### Parameters

IUserID

[in]NET\_DVR\_Login()返回值;

dwCommand

[in]参数类型, 车牌宏命令 NET\_DVR\_GET\_PLATECFG

IChannel

[in] 通道号

lpOutBuffer

[out] 返回参数结构的数据缓冲区, 这里为 NET\_VCA\_PLATECFG 结构

dwOutBufferSize

[out] 缓冲区的大小, 这里为 sizeof(NET\_VCA\_PLATECFG)

lpBytesReturned

[out] 实际返回的缓冲区大小

### Return Values

成功返回 TRUE

失败返回 FALSE, 通过 NET\_DVR\_GetLastError 获取错误码, 详细请参阅《客户端 SDK 使用手册 V2.20》。

## 2.4 设置设备对应配置参数

```
BOOL NET_DVR_SetDVRConfig(  
    LONG    IUserID,  
    DWORD   dwCommand,  
    LONG    IChannel,  
    LPVOID   lpInBuffer,  
    DWORD   dwInBufferSize  
);
```

### Parameters

IUserID

[in]NET\_DVR\_Login()返回值;

dwCommand

[in]参数类型, 车牌宏命令 NET\_DVR\_GET\_PLATECFG

IChannel

[in] 通道号

lpInBuffer

[in] 存放输入参数的缓冲区, 这里为 NET\_VCA\_PLATECFG 结构

dwInBufferSize

[in] 缓冲区的大小, 这里为 sizeof(NET\_VCA\_PLATECFG)

### Return Values

成功返回 TRUE

失败返回 FALSE, 通过 NET\_DVR\_GetLastError 获取错误码, 详细请参阅《客户端 SDK 使用手册 V2.20》。

## 2.5 进行告警上报的布防/撤防

/\*\*\*\*\*\*

### Func:NET\_DVR\_SetupAlarmChan

Description: 建立报警上传通道

Input: IUserID: NET\_DVR\_Login () 的返回值

Output:

Return: -1 表示失败, 其他值作为 NET\_DVR\_CloseAlarmChan() 等函数的参数

\*\*\*\*\*/

LONG NET\_DVR\_SetupAlarmChan(LONG IUserID); //布防

/\*\*\*\*\*\*

### Func:NET\_DVR\_CloseAlarmChan

Description: 断开报警上传通道

Input: lAlarmHandle: NET\_DVR\_SetupAlarmChan 的返回值

Output:

Return: TRUE/FALSE

\*\*\*\*\*/

BOOL NET\_DVR\_CloseAlarmChan(LONG lAlarmHandle); //撤防

以上两个接口详细请参阅《客户端 SDK 使用手册 V2.20》。