

Media Guard App V3.2

開發願景與目標:

CCTV安防行業技術提供商與升級AI革命應用提供解決方案.

D:\MediaGuardAppV31 | GIT : MediaGuardAppV32

雲平台連接或獨立脫機運行

API 請查閱 ./MediaGuard_DOC/DVR開發相關的說明VER3.1_2024-12.doc

安裝與部署 SETUP & DEPLOY

./SETUP_n_DEPLOY/SETUP_n_DEPLOY.MD

CMAKE項目的VS2022開發環境

請查閱 ./MediaGuard_DOC/README_組件安裝.md。

開發備註

2025-1-2

舊版 windows 運作正常, Linux版編譯通過,但是需要CUDA 顯卡硬件環境運行測試驗證,目標是: linux 穩定版。

linux 版編譯成功,運行需要支持CUDA顯卡的 Ubuntu 22.x 驅動才能運行CUDA硬件解碼。

開發環境詳細備註需要參考: MediaGuard_DOC/README_安裝與部署.md。

1.openssl linux和windows 編譯版本

2.curl 版本需要linux編譯 主要用於 函數MD5 (curl 是http協定指令)

3.ffmpeg 問題:

ffmpeg編譯的時候,需要選擇CUDA,如果機器沒有就不選。

HardAndSoftDecode_Ref.md 參考軟解和硬解碼。

代碼結構 SOFTWARE STRUCTURE

Linux 系统开发板安装 GCC 交叉编译器

參考LowLevelAnalsysic_MediaGuard.pdf 核心文件三个

1. RtspStreamHandle.cpp 解碼單元 對當前鏡頭單元解碼 MP4 FLV HLS JPEG RTMP等等
對解碼單元进行管理

2. StreamManager.cpp 創建鏡頭【解码单元】的日期文件夾等等 和當前鏡頭單元相關的。
整体控制
例如 StreamManager的構造函數

```
switch (nStreamType)
{
case kStreamTypeUsb:
    m_pHandle = std::make_shared<UsbStreamHandle>();    //沒用到 ignore
    break;
case kStreamTypeAudio:
    m_pHandle = std::make_shared<AudioStreamHandle>(); //沒用到 ignore
    break;
case kStreamTypeRtsp:
    m_pHandle = std::make_shared<RtspStreamHandle>();  //RTSP解碼錄像單元
    break;
default:
    m_pHandle = std::make_shared<RtspStreamHandle>();
    break;
}
```

StreamInfo 傳入參數結構體

StreamInfo 是傳入鏡頭對象/解碼單元(RtspStreamHandle.cpp)的參數對象結果(參數結構體)。

同時在運行過程中賦值像素長寬等等鏡頭解碼參數出去外面。

(\MediaGuard\StreamDefine.h)

1. 硬件解碼 (NVIDIA CUDA / INTEL INTEGRATE DISPLAY) 實例要修改對應參數 StreamInfo.nHdType
2. 如果要從雲端獲取則要從幾個API綜合獲取對應參數。
3. API: CameraMpeg::camera_list;
CameraMpeg::device_by_serial_no;CameraMpeg::setting_n_schedule_by_camera_id
4. 還有一個是邏輯轉換的api camera_list_trans_to_strean_info，主要是把雲端邏輯轉換為具體情況的,例如：nHdType合成本地的Device.json, Camera Info 來自API
5. 例如：bRtmp = ttrue是開啟RTMP的，但新規則改為 StreamDecodeType (StreamInfo.nStreamDecodeType)的類型，HLS/RTMP/進行切換。
6. 目前網上雲端沒有bRtmp對應的參數，對應的 CamSettingNSchedule。RtmpOutput和雲端對應的沒有UI改動的，只有默認FALSE.所有對RTMP比較混亂的，所以默認就是FALSE

通過共享指針實現獲取控制單元。

1. ManagerController.cpp 例如處理控制 保留全局文件夾Picture。
2. CameraMpeg.cpp 主要是API 業務邏輯的操作 記錄媒體文件開始結速時間等等的操作。

開發與場景功能描述

開發願景與目標: CCTV安防行業技術提供商與升級AI革命應用提供解決方案.

1. 目前實現[NVR 錄像設備]
2. 解碼單元提供各种识别业务与警报的圖片以供調用：

3. 人脸识别/警报
4. 二维码识别/警报
5. 车牌识别/警报
6. 鼠患识别/警报
7. 行为识别/警报
8. 入侵區域告警

功能

I:图片动态内存队列保存最近15分钟的图片，可配置保存硬盤選項,但內容隊列選項是固定的程序。

II:动态把Camera解码单元 (RtspStreamHandle.cpp) 的单元图片列表转移保存到全局列表 (内存) 以供识别业务处理。

III:实现MP4/FLV 持續保存

#全局设置

GlobalSetting 来自/conf/device.json 的全局配置。

1. picRemainMinutes;设置保存的图片在硬盘缓存时长,默認十五分；
2. PictInfo pictInfo; 保存圖片信息結構體,包括Base64格式存儲隊列(m_listFrame)中。
3. videoRemainMinutes = 15; //无论超出容量限制都要最少保留15分钟的video，预设必须考虑最少可以存儲15分钟的影片
4. HWDeviceType nHdType = kHWDeviceTypeNone; //硬件类型 默认是没有。

操作方法備註

主控台操作

1. 增加 Ctrl+c Ctrl+Break Enter 行為 (signal 監視) 。
2. 只能輸入exit退出。

WEB 頁面

```
HttpServer 測試 http://192.168.0.128:180/hello
部署測試      http://192.168.0.128:180/test      對應的實體文件：
/web/index_template.html
跨域與登錄測試 /web/playtext.html  需要文件形式在瀏覽器打開
```

上述IP要根據實際部署調整。

網絡穿透

實現雲平台與設備互訪, 目前技術問題, 不使用STUN協議實現,而是使用CURL命令獲取公網IP同步到雲端,LINUX版本需要安裝CURL組件。

