



## AI 应用第一课：支付宝刷脸登录

### 1. 项目应用场景

VIP 学员小林入职的第一个项目：



支付宝刷脸支付：





## 2. 项目演示

演示奇牛支付刷脸登录。

## 3. 环境搭建

直接领取开发模板，直接开发。上完课后，再按照以下步骤自己搭建。

### 1. 配置人脸识别 SDK 开发环境

注意，配置环境前，先把 VS 的平台调整为 64 位系统（也可以使用 32 位，但是人脸识别的 SDK 也要改位 32 位版本的）

补充：

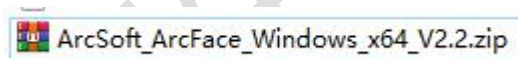
商业开发，人脸识别相关的项目，千万不要自己开发人脸识别  
应该选择合适的人脸识别 SDK

选择人脸识别 SDK：百度，腾讯，阿里，虹软

所有的商业 SDK，都有两个版本：（研发版本【免费的】，发行版【付费的】）

比如： UE4

### 下载虹软 SDK（64 位）



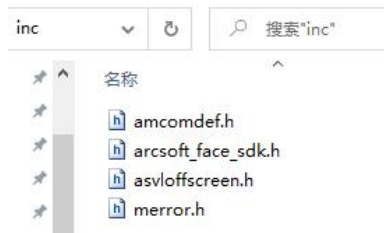
### 配置头文件目录和库文件目录

在项目目录下，创建 face 目录，再把人脸识别 SDK 的头文件和库文件复制到项目目录下的 face 目录下

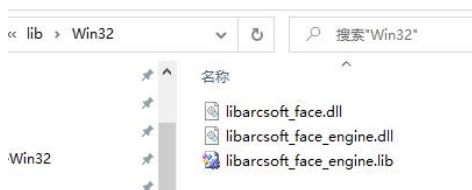




头文件



库文件



配置 SDK 的头文件目录

配置 SDK 的库文件目录

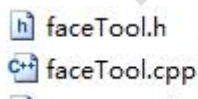
## 配置附加依赖项

libarcsoft\_face\_engine.lib

## 复制 DLL 库文件到可执行文件目录



## 导入 Rock 自定义的接口文件



## 2. 配置 opencv 开发环境

opencv 是图像处理，应用最广的免费的开源框架

已经有人研发了，使用 opencv 配合激光炮打蚊子的系统（秒变军用，斩首行动）

opencv 使用 C/C++ 开发（其中有 500 个常用接口使用 C 语言，其他使用 C++），跨平台



Qt 是跨平台的应用框架

opencv 是跨平台的图像处理框架

Qt + opencv

安装 opencv

头文件包含目录:

D:\opencv\build\include

D:\opencv\build\include\opencv

D:\opencv\build\include\opencv2

库文件包含目录:

D:\opencv\build\x64\vc12\lib

使用的库 (附加依赖性)

(以下是最常用的库文件, 还有很多其他的库文件)

opencv\_core249d.lib

opencv\_highgui249d.lib

opencv\_imgproc249d.lib

动态库

opencv\_core249d.dll

opencv\_highgui249d.dll

opencv\_imgproc249d.dll

### 3. 导入第 3 方类

CV\_Img

补充: 高薪就业最常见的面试问题:

你使用过哪些第三方框架? 使用过哪些第三方的类、文件、库、工具?

比如: 网络开发, libevent Qt 开发中的表格图像处理: xxx



## 导入 Rock 私有工具库



## 4. 项目实施

### 1. 设计系统界面



始于颜值，新手开发，从软件界面开始。

#### IT 开发常识

界面的设计，由美工说了算！美工的设计由产品经理说了算，产品经理的决策，由甲方爸爸说了算！

美工出图：（同时提供素材）

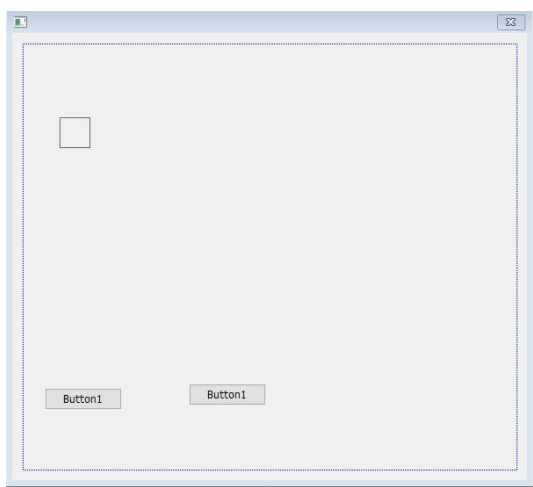


程序员实现：

一个图片控件，两个按钮控件。

一分钟搞定。

## 拖放控件



一个图片控件，2 个按钮控件

此时的界面，颜值丑陋，欲爱不能，不堪使用！

## 界面修饰

把欢迎界面的素材图片，加载到项目资源中。

- 1) 拷贝素材图片到项目目录的 res 子目录
- 2) 把图片添加到项目资源中





3) 配置图片控件

类型，改为：Bitmap

图像，设置为：IDB\_BITMAP1

4) 调整界面布局

问题：界面调不动了！

技巧：因为图片控件的图片太大了，导致对话框不能操作

把图片控件的类型，再设置为 **frame**，然后再把窗口拖放到更大，再把图片的类型调回到 **Bitmap**

最后再把窗口缩小到合适程度。

5) 修改按钮的文字

分别设置为“注册”、“登录”

设计效果：



## 为界面注入灵魂

此时的界面是一个没有灵魂的肉体。



1. 给图片控件修改 ID: IDC\_STATIC\_IMG



设置一个“成员变量”，变量名为： m\_img

2. 给两个按钮修改 ID

IDC\_BUTTON\_REGIST

IDC\_BUTTON\_LOGIN

3. ”添加“单击事件处理程序”

双击按钮，自动生成！

## 界面的灵魂测试

随意找两张图片，mm1.bmp, mm2.bmp

在两个按钮的事件处理程序中，分别添加：

```
HBITMAP pic = (HBITMAP)LoadImage(NULL, L"mm1.bmp", IMAGE_BITMAP, 0, 0, LR_LOADFROMFILE);  
m_img.SetBitmap(pic);
```

```
HBITMAP pic = (HBITMAP)LoadImage(NULL, L"mm2.bmp", IMAGE_BITMAP, 0, 0, LR_LOADFROMFILE);  
m_img.SetBitmap(pic);
```

测试效果。

## 2. 使用摄像头

要使用刷脸认证，必须使用摄像头。

原理：

使用定时器，定时的捕获摄像头的图像数据，把捕获的图像帧，显示到图片控件即可。

1. 添加工具库

tools.h tools.cpp CvImage.h CvImage.cpp

（tools.h 、tools.cpp 是 Rock 写的，专门用于摄像头处理的函数，可作为自己的私有工具使用）

把这两个文件复制到项目目录下，并添加到项目中。

2. 定时摄像头变量

```
#include <opencv2/opencv.hpp>  
using namespace cv;
```





```
// 定义一个摄像头变量
VideoCapture cap;
```

3. 在注册按钮的事件处理程序中，打开摄像头，并创建定时器

```
// 1. 打开摄像头
cap.open(0); //打开默认的摄像头

// 2. 创建定时器
SetTimer(
    1,          //定时器的编号
    100,        //每隔 100ms 提醒一次
    NULL);     //时间到了之后，执行哪个函数，NULL 表示执行默认的定时处理函数
```

4. 为对话框创建定时消息

先选择对话框，然后在对话框的属性窗口中的“消息”选项卡，添加“WM\_TIMER”消息



5. 在定时处理函数中，实现定时捕获

```
#include "tools.h"

void COpenCVTmpDlg::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值
    if (nIDEvent == 1) {
        // 把摄像头 cap 捕获的图像帧，放到当前窗口 m_hWnd 的 IDC_STATIC_IMG 控件中显示
        refreshCamera(&cap, m_hWnd, IDC_STATIC_IMG);
    }

    CDialogEx::OnTimer(nIDEvent);
}
```

## 3. 实现注册

方法：使用摄像头倒计时拍照，然后保存照片即可。

```
#include <mmsystem.h>
#pragma comment(lib, "winmm.lib")
```



```
void COpenCVTmpDlg::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值

    static int count1 = 0;

    if (nIDEvent == 1) {
        // 把摄像头 cap 捕获的图像帧，放到当前窗口 m_hWnd 的 IDC_STATIC_IMG 控件中显示
        refreshCamera(&cap, m_hWnd, IDC_STATIC_IMG);

        count1++;

        if (count1 == 1) {
            mciSendString(_T("play res/zhuche.mp3"), 0, 0, 0); // 播放 3 秒倒计时的功能
        } else if (count1 == 30) {
            paiZhao(&cap, "Rock.jpg");

            count1 = 0;
            KillTimer(1); // 关闭定时器
            cap.release(); // 关闭摄像头

            MessageBox(_T("注册成功!"));

            // 把画面切换到最开始的欢迎画面
            HBITMAP hbitmap = (HBITMAP)LoadImage(NULL, "res/start.bmp", IMAGE_BITMAP, 0, 0, LR_LOADFROMFILE);
            m_img.SetBitmap(hbitmap);
        }
    }

    CDialogEx::OnTimer(nIDEvent);
}
```

## 4. 人脸识别

人脸识别分析：

直接像素对比不合适，不能使用连连看辅助的方式。

解决方案：

使用人脸识别 SDK



## 1) 刷脸

和登录的方式类似.

```
void COpenCVTmpDlg::OnBnClickedButtonLogin()
{
    // TODO: 在此添加控件通知处理程序代码
    //HBITMAP pic = (HBITMAP)LoadImage(NULL, "mm2.bmp", IMAGE_BITMAP, 0, 0, LR_LOADFROMFILE);
    //m_img.SetBitmap(pic);

    cap.open(0);
    SetTimer(2, 100, NULL);
}

void COpenCVTmpDlg::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值
    static int count1 = 0;
    if (nIDEvent == 1) {
        .....
    } else if (nIDEvent == 2) {
        refreshCamera(&cap, m_hWnd, IDC_STATIC_IMG);
    }

    CDialogEx::OnTimer(nIDEvent);
}
```

## 2) 定义人脸识别接口

a. 导入人脸识别工具库 faceTool.h faceTool.cpp 到项目中。

b. 添加头文件

```
#include "faceTool.h"
```

c. 定义人脸识别模块变量

```
// 人脸识别的模块变量
MHandle handle;
```

d. 对模块变量进行初始化

在对话框窗口的初始化函数中, 添加:



```
faceInit(&handle);
```

## E. 定义人脸识别的接口 faceCheck

```
BOOL faceCheck(const char* name1, const char* name2) {  
    // 读取两个照片  
    // 数据类型的大小写是敏感的!  
    IplImage* img1 = cvLoadImage(name1);  
    IplImage* img2 = cvLoadImage(name2);  
  
    if (!img1 || !img2) {  
        return false; //FALSE  
    }  
  
    float ret = faceCompare(handle, img1, img2);  
  
    // return ret >= 0.95  
    if (ret < 0.95) {  
        return false;  
    }  
    else {  
        return true;  
    }  
}
```

## 3) 实现人脸识别登录

方法:

第 1 秒结束时, 自动拍照、并进行人脸识别,

第 2 秒结束时, 根据上一秒的识别结果刷新界面

代码实现

```
void COpenCVTmpDlg::OnTimer(UINT_PTR nIDEvent)  
{  
    // TODO: 在此添加消息处理程序代码和/或调用默认值  
    static int count1 = 0;  
    static int count2 = 0;  
    static bool success = false; //表示识别结果是否成功  
    if (nIDEvent == 1) {  
        .....  
    }  
    else if (nIDEvent == 2) {  
        refreshCamera(&cap, m_hWnd, IDC_STATIC_IMG);  
        count2++;  
    }  
}
```



```
        if (count2 == 10) {
            paiZhao(&cap, "Rock2.jpg"); //拍照

            // 人脸识别
            if (faceCheck("Rock.jpg", "Rock2.jpg")) {
                success = true;

                // 播放一个登录成功的声音，持续 1S
                mciSendString("play res/login.mp3", 0, 0, 0);
            }
        } else if (count2 == 20) {
            count2 = 0;
            KillTimer(2);
            cap.release();

            const char* fileName = success ? "res/bgInit.bmp" : "res/bgError.bmp";
            HBITMAP hbitmap = (HBITMAP)LoadImage(NULL, fileName, IMAGE_BITMAP, 0, 0, LR_LOADFROMFILE);
            m_img.SetBitmap(hbitmap);

            success = false;
        }
    }

    CDialogEx::OnTimer(nIDEvent);
}
```

## 5. 项目完善

### 1. 活体检测

张张嘴，向左摇头，向右摇头，点点头， 眨眨眼（金融相关的!）

### 2. 多用户账户登录

使用数据库

### VIP 课程第 3 卷

已就业学员在研发工作中，90%以上要使用数据库。

### 3. 跨平台

使用 Qt 移植到其他手机平台



## 6. 项目小结

1. 项目开发就是，从需求入门，逐步完善的过程。
2. 技术大佬，也都是从零基础开始，逐步升级，逐步变强的过程
  - 小翔-大二开始迷茫，刚开始学习时，只求能找到工作就足够了，后面入职百度服务器开发，年薪 30W
  - Marin, 32 岁开始学习，最开始只求能够快速转行，6 个月后入职 C++ 音视频服务器开发，月薪 15K，4 个月后，月薪 18K
  - 佳佳，大三实习，只求不要到学校指定的工厂实习，现在已经入职 C++ 开发一年多了