

# 基于深度学习的医院患者面部信息录入与识别系统设计 与制作



班 级： 16 计科 C2  
姓 名： 宋远迪  
学 号： 20161112726

# 目录

1. 设计总览
2. 深度学习的相关介绍
3. 设计思想
4. 开发与使用环境
5. 功能实现
6. 设计体会

## 1. 设计总览

基于深度学习的医院患者面部信息录入与识别系统是以患者个人面部为特征，适用于医院的各个身份验证环节。本系统可使人摆脱繁琐、低效的身份验证模式。识别工作不受环境影响，操作方便快捷。结合经过深度学习训练的人脸识别模型，在用户第一次入院时登记面部信息，之后只需在摄像头前直接利用个人面部即可获得个人在院中的信息。

软件共分为三个界面，分别为主界面，信息录入界面，信息识别页面。

## 2. 深度学习的相关介绍

**深度学习 (deep learning)** 是机器学习的分支，是一种试图使用包含复杂结构或由多重非线性变换构成的多个处理层对数据进行高层抽象的算法。深度学习是机器学习中一种基于对数据进行表征学习的算法。观测值（例如一幅图像）可以使用多种方式来表示，如每个像素强度值的向量，或者更抽象地表示成一系列边、特定形状的区域等。而使用某些特定的表示方法更容易从实例中学习任务（例如，人脸识别或面部表情识别）。深度学习的好处是用非监督式或半监督式的特征学习和分层特征提取高效算法来替代手工获取特征

**Face\_recognition** 是一个强大、简单、易上手的人脸识别开源项目，并且配备了完整的开发文档和应用案例，特别是兼容树莓派系统。本项目的人脸识别是基于业内领先的 C++ 开源库 dlib 中的深度学习模型，用 Labeled Faces in the Wild 人脸数据集进行测试，有高达 99.38% 的准确率。但对小孩和亚洲人脸的识别准确率尚待提升。

[Labeled Faces in the Wild](#) 是美国麻省大学安姆斯特分校 (University of Massachusetts Amherst) 制作的人脸数据集，该数据集包含了从网络收集的 13,000 多张面部图像。

（本项目内容翻译 by：同济大学开源软件协会 [子豪兄 Tommy](#)）

### 3. 设计思想

医院这一场景，短时间内要面对大量患者，而患者又由在医院中的就诊卡号唯一标识，患者与各个诊室间交互都需要用到就诊卡号这一信息。但传统就诊卡具有易丢失，难以复用等多种问题。而面部信息这一表征是各个患者专属的标识，将其直接与用户就诊卡对应，即可实现短时间内，不需要借助除摄像头外其他设备，的用户身份识别。

对于使用场景而言，分为录入与识别两种使用场景。

录入：用户第一次就诊时，将面部信息录入医院数据库（本软件因为并未实际介入医院系统，故在使用中为本地文件夹）。

识别：用户进入各个诊室时，只需调用摄像头识别面部信息，与数据库验证，即可获取用户姓名及卡号，方便进行接下来的用户行为。姓名卡号在本软件中也单独存放，可以打开接口直接与其他设备/行为进行交互使用。

## 4. 开发与使用环境

本项目基于 python3.7 开发:

<https://www.python.org/downloads/release/python-370/>

使用 pyqt5 完成图形界面开发:

<https://riverbankcomputing.com/software/pyqt/intro>

使用 opencv3.4.5(cv2) 完成图像获取及处理部分开发:

<https://opencv.org/>

使用 face\_recognition 完成面部识别部分开发:

[https://github.com/ageitgey/face\\_recognition](https://github.com/ageitgey/face_recognition)

使用时需要 python3.7 环境并加载上述函数库

## 5. 功能实现

主要流程如图实现，所有数据除访问数据库外都只存在于当前窗口下

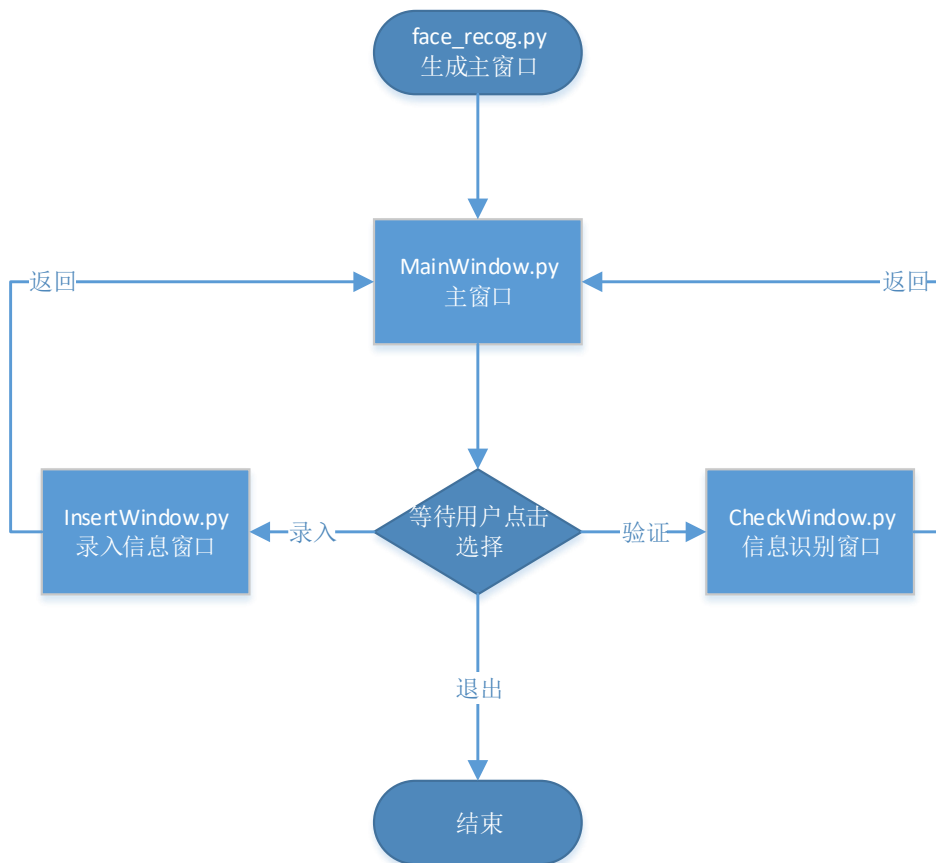


图 1

## 5.1 主窗口

窗口设计如图 2 所示，功能如流程图 1 中所示



图 2



## 5.2 录入窗口

窗口设计如图 3 所示，左侧灰色框体为摄像头图像显示窗口，右侧为信息键入部分。

工作逻辑如图 4 所示。

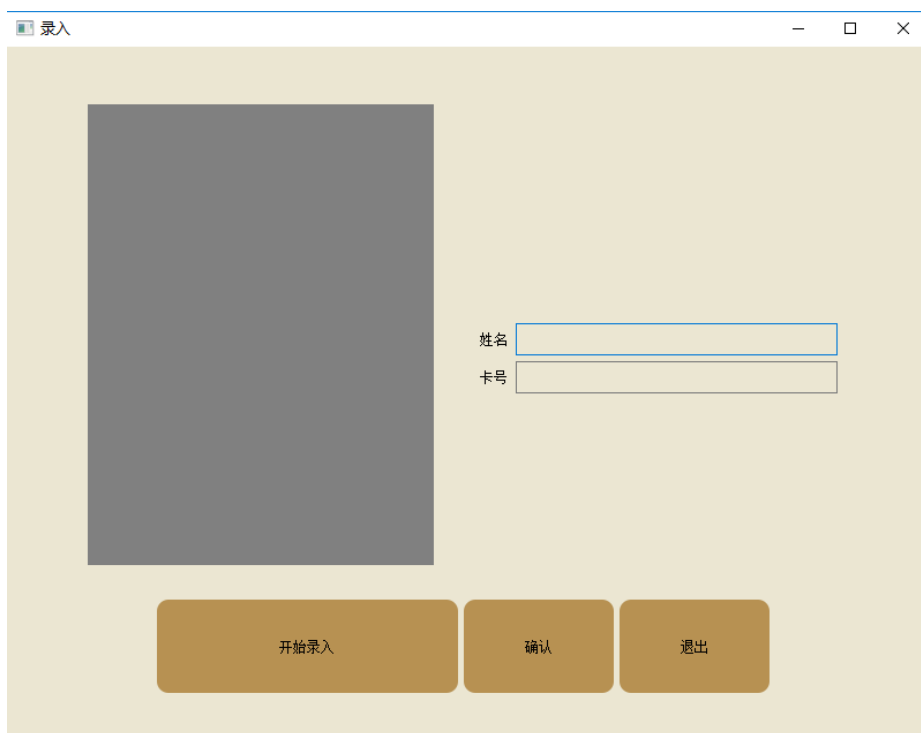


图 3

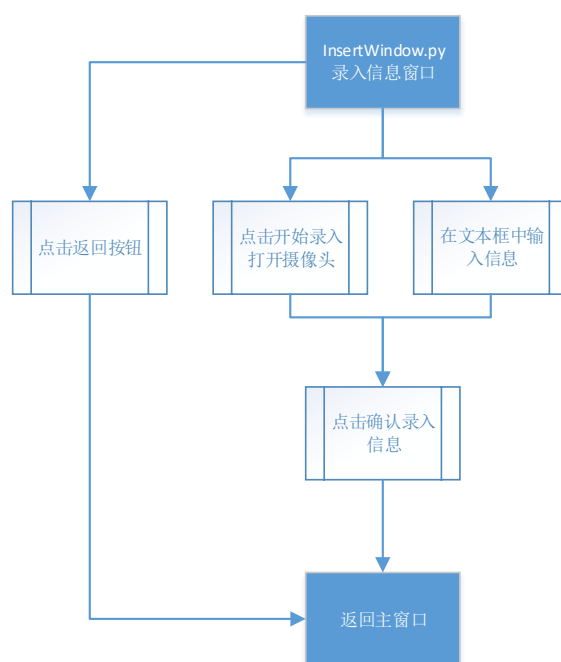


图 4

## 5.3 识别窗口

窗口设计如图 5 所示，左侧灰色框体为摄像头图像显示窗口，右侧为信息显示部分。

工作逻辑如图 6 所示。

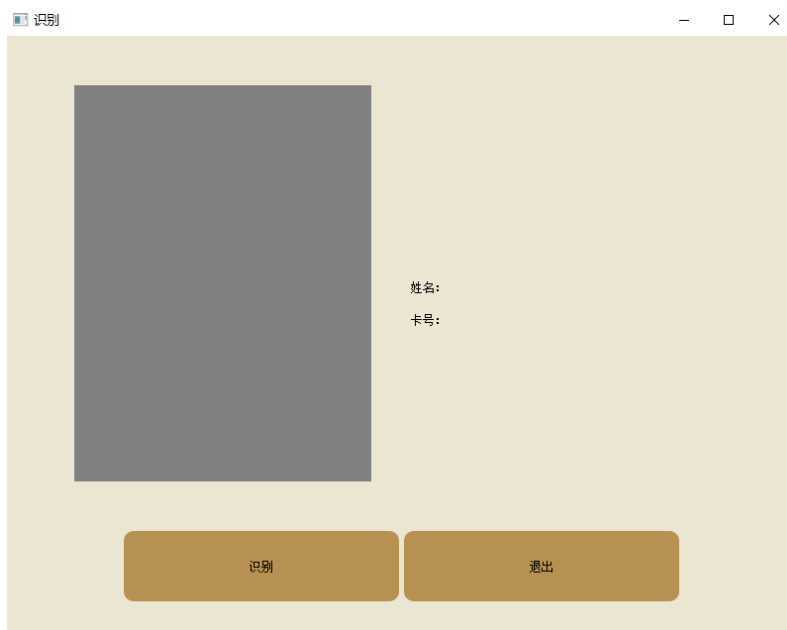


图 5

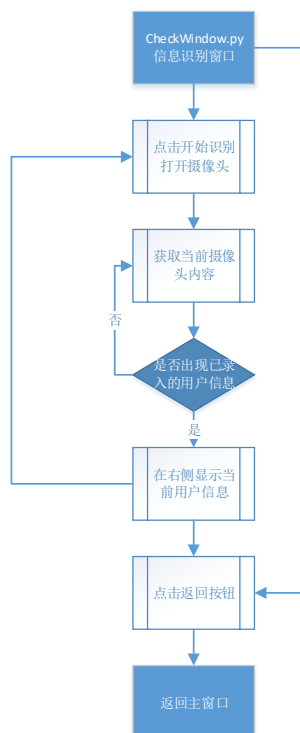


图 6

## 5.4 其他问题

因为录入和识别均需要单独调用摄像头，而一般电脑仅有一个摄像头，当被占用时将会无法重复调用。故录入和识别窗口同时间仅能使用其中一个（录入或识别）。程序中命名了摄像头选择的变量，若需用到多个摄像头，也可以增加获取可用摄像头及摄像头选择的方法。

录入的信息存放于 /当前目录/recorded 中，故用户也可以手动添加新的信息，也可以直接对内容进行操作。后续维护中可以选择为其设计一个管理方式，并将信息加密存储，以提高安全性与实用性。

## 6. 设计体会

这个项目其实拖了挺久的，第一次拿到需求的时候，直接从 face\_recognition 库作者的 GitHub 上拉下了利用摄像头实时进行人脸识别的 demo，当时认为没有什么难度也不好改进，就一直放着没管。

后来再重新开始做这个项目的时候学期已经过了大半了，当时有想过利用 opencv 来自己实现一个已存数据淘汰算法，在用户每次录入的时候，结合之前的结果来改进已存的数据。然而实现起来太过困难，编写淘汰算法的估价函数需要大量计算机视觉的相关知识，既是算法设计出来也需要借助机器学习来进行大量的模型训练，这在剩下的半学期内完成显然是不可能的。

最后选择了一个限定场景，更具有实用性的方向，整合各个功能将其做成一个可以直接使用的软件。

首先，这样的软件需要一个图形界面，因为 tkinter 部署方便，所以我一开始选择了用 tkinter 编写图形界面，在完成了大部分的界面设计之后，我发现 tkinter 纯代码编写界面实在是具有局限性。后来选择换为 pyqt5 来编写图形界面，直接在界面设计器中选择界面属性与位置将工作效率提升许多。

下一个问题是实时显示摄像头内容的问题，之前在使用 tkinter 的时候就在这个问题上花费了很多时间，因为 python 的弱类型性质，再加上 tkinter 的 document 完成度并不高，在控件引用内容的类型上迷茫了好长时间，最后换成 pyqt5 也有这方面的考虑，pyqt5 的文档相对完善，其他人使用的例子也更多一些。最后在这几个库文档的支持下实现了这个功能。

至此，基础问题已经解决完毕，软件已经可以使用了。但在测试阶段发现了一个导致程序直接崩溃的 bug，不加以限制保存图片直接在 face\_recognition 库函数调用过程中会直接产生数组溢出的问题，但由于是第三方库，调试信息也并没有写明，这个问题花了很久都没有能找到具体产生原因。最后能定位到问题是因为 GitHub 的讨论区中有人遇到了相关问题，回答区有人分析说是因为图片中可能并未出现可识别的人

脸，所以在将解析信息导入待解析数组的时候会产生溢出问题。定位之后这个问题就好解决了，只要确认找到能解析的脸之后再进行录入操作，就可以保证录入的图片可被解析。至此，所有问题全部解决。虽然仍存在一些新问题，比如 tab 无法快捷切换输入栏，并不能从操作逻辑上限制用户，保证不重复调用同一摄像头句柄等，但功能与正常工作在当前版本中已经可以保证。后续维护会修复这些问题。

其实在这个项目之前我对 python 也就是能写一两道算法题的理解程度，图形化编程的经验也只有上学期的 C#实践中积累的经验。python 的弱类型让人又爱又恨，因为弱类型，变量无法被定义，当时为了设置一个全局变量，就花了我很大功夫，但是弱类型也有好处，写的时候只要一股脑往下写就可以，只要编译器不报错一般就不会出问题。但无论如何，python 的稳定性实在是堪忧，我不太会、也不太想建议别人一直用 python 进行开发。尽管这个语言确实贴近自然语言，以至于我只花了一天来学习 python 和 py 图形化编程，又用了不到 24 小时的就完成了整个软件的开发与简单测试。

虽然最初的想法并没有得以实现，但是现在的样子似乎看起来更实用一点。整个过程中还是收获颇丰，丰富了很多 python、机器学习、计算机视觉相关的经验，但小组合作部分，也可能是我的问题，实在是太过缺乏沟通，都是自己做自己的，以后在面对相似的情形的时候，需要多加注意。