# 软件名称：基于眼动控制的PDF阅读器

### 一、设计稿。

###### 1、PDF阅读器设计：

1）、

开发依赖：Python3，PyQt5，PyMuPDF。

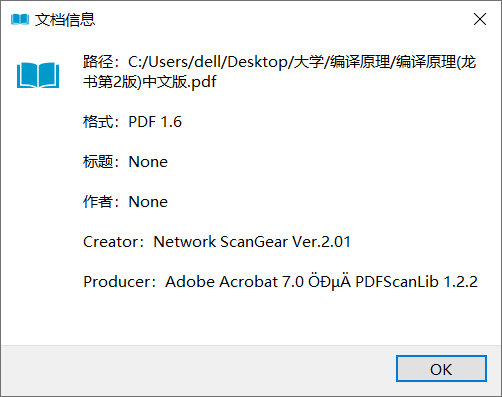
参考了开源代码，采用PyQt5作为前端，Python3作为开发环境，PyMuPDF 则用来解析 PDF ，来获取 PDF 文本信息。

首先在图书库中我们使用QtWidgets.QTableWidget来管理图书，同时收录了图书信息并在界面中显示图书的缩略图封面来使用户更简单的能找到自己想要的图书。



（主界面效果图）

同样的为了用户方便操作以及获取图书信息，我们运用QMenu设置右键菜单方便用户对图书进行操作。同时，为了保存用户上次导入的图书，我们使用sqlite3 来保存用户的数据，sqlite3 是轻量型本地数据库，在每次打开前阅读器都会从数据库中读取图书信息。

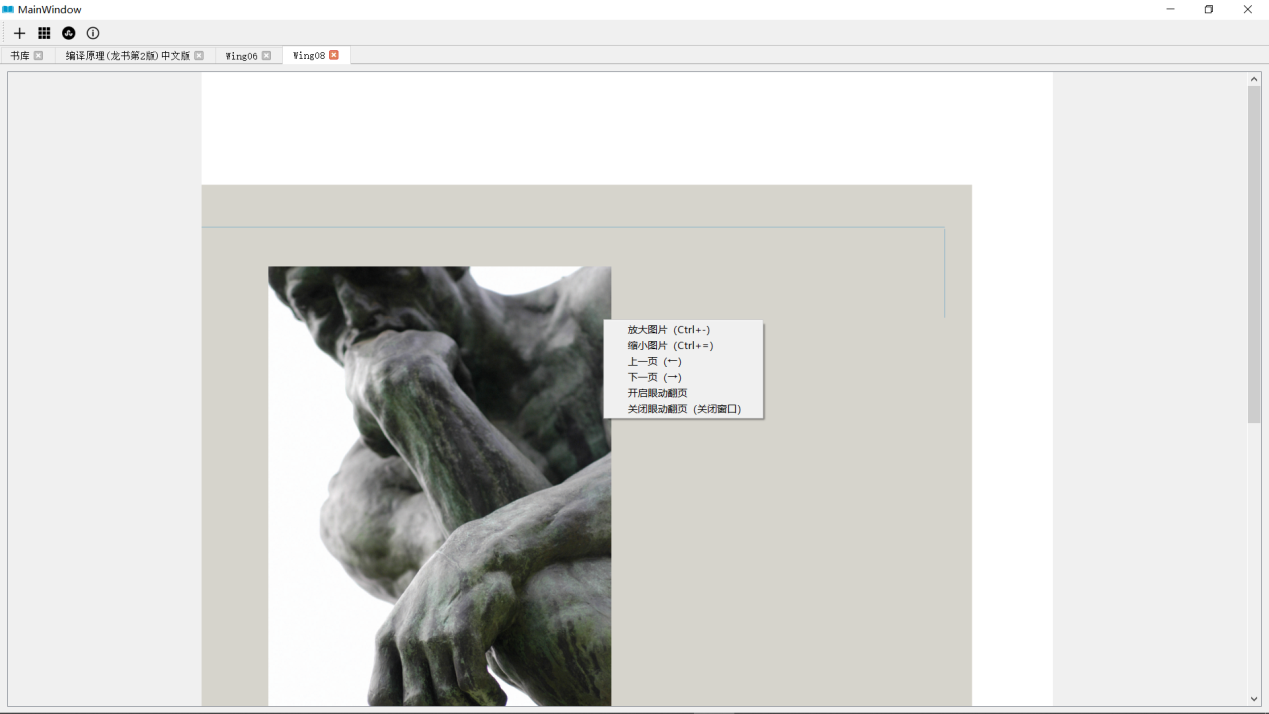


（图书信息样例）

由于QTabWidget 可以允许我们在一个窗口显示多个页面，我们即可以同时打开多个图书同时阅读，每次打开一本图书，我们只要新建一个QTableWidget的选项卡即可，无需打开新的窗口。

我们使用QScrollArea打开pdf图书页面，界面支持滚轮滑动并可以放大缩小。同样的我们运用QShortcut绑定按键事件，增加了放大，缩小，翻页的快捷键。同样，我们增加鼠标的按键事件，只要鼠标在pdf界面的左三分之一位置左键，即可向前翻页，向后同理。

同样，在阅读界面右键也可以打开QMenu菜单进行上述操作。



（阅读界面效果图）

###### 2、眼动识别模型设计：

1）、摄像头调用

借用了opencv库，调用电脑摄像头捕捉图像，并裁剪了图像提高运算速率和定位准确率。

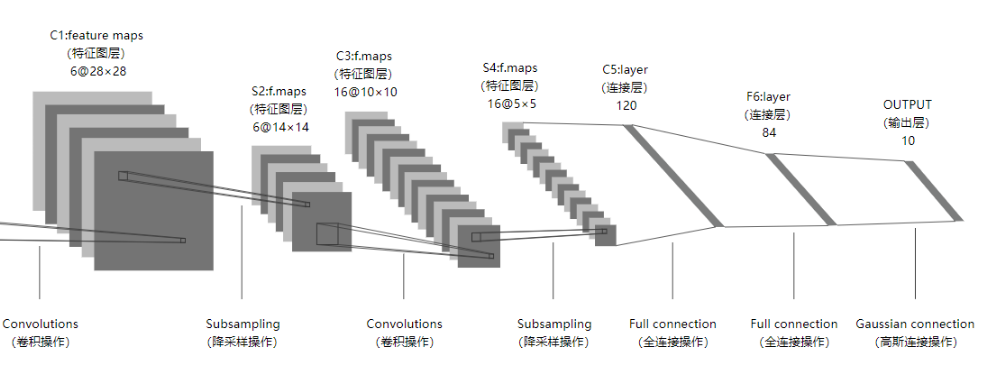
2）、模型组成：

a、脸部定位模型：采用了dlib的人脸定位函数get\_frontal\_face\_detector()，捕捉到人脸，再进行下一步定位；

b、眼部定位模型：采用了dlib自带的人脸关键点定位模型，取出其中有关人眼关键点，提取人眼矩形框，进入到下一步状态判断。

c、人眼状态判断：使用tensorflow库，参考LeNet-5卷积神经模型，搭建了一个卷积神经网络模型，模型如图所示：

为了一定程度上防止过拟合，在第二层卷积层之后加了一层Dropout层。



结果输出one-hot编码的四维向量，分别代表闭眼，睁眼，左看。右看。

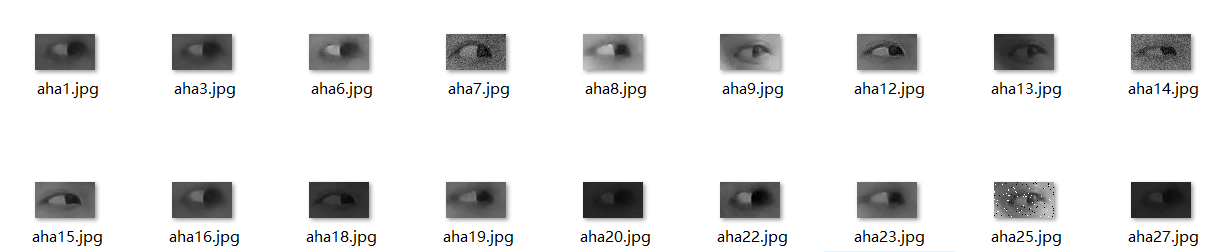
###### 3、眼动识别模型数据集获取

第一阶段：

图片使用了来自浙江大学的ZJU睁眼闭眼开源数据集和LFW的人脸图像集的CEW睁闭眼数据集，以及队员自采数据集组成，经过处理筛选，共30000张左右。训练出来的模型缺乏了一定的泛化能力，因此对数据集进行了改造。

第二阶段：

对第一阶段的图片进行了图片增强，在原有图片的基础上加入了高斯噪声，椒盐噪声，亮度增减变化，对比度增减变化，锐度增减变化，最终处理筛选，得到69192张图像。其中睁眼数据集23166，闭眼数据集18910张，左看数据集16489张，右看数据集17204张。数据集展示如图：



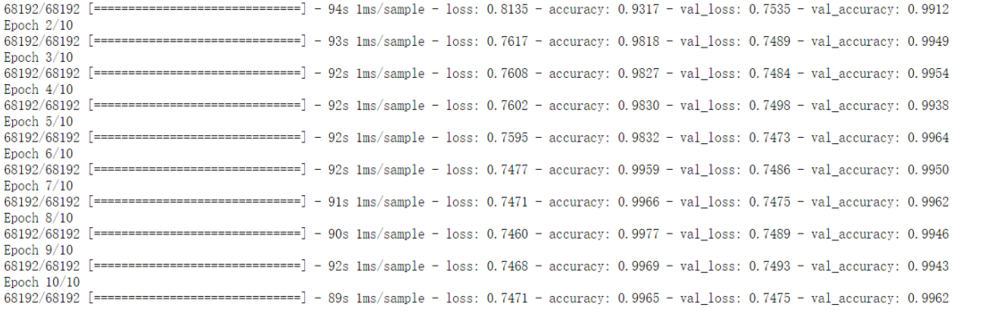
###### 4、使用数据集进行模型训练

1）、训练集处理：

得到训练数据，为防止数据集产生规律分布，影响训练集和验证集的分布，对数据集进行随机重命名打乱。

2）、训练集训练：

训练时从数据集取出10%作为验证集，并迁移了之前的初步训练模型，经过30轮迭代，训练集准确率和验证集准确率均能达到99%以上。



最终模型保存为h5格式供调用。

演示视频如下：

###### 5、PDF阅读器和眼动模型交接

1）、模型输出结果处理：

由dlib得到的左右眼图像输进模型，获得两眼的判断结果。先由两眼结果输出一个总的判断结果，然后判断：

a、翻页：

当一个总的判断结果连续3帧被判定为左看或者右看，则调用QT接口，实现左翻页和右翻页。同时对于结果进行了重复判断，如果连续判定为一个动作，则只下达一次指令，防止连续的翻页。

b、上锁：

当一个总的判断结果连续5帧被判定为闭眼，则执行上锁功能，此时用户眼动翻页功能将被锁定。同理，执行相同的功能解锁。

2）、摄像头调用

摄像头由cv2. VideoCapture创建摄像头对象并打开摄像头截取图像，并对图像进行逐帧识别。由于Python是多线程的，所以直接用函数调用可能会出现，函数被同时多次调用导致摄像头无法二次打开以至于程序崩溃。固添加函数锁，若摄像头无法打开则不执行函数。

3）、右键菜单绑定