# 项目名称：Expression

**项目成员：** 彭天祥 201730683314

谢晓民 20173068

申浩 20173068

吴岚锋 20173068

张利丹 20173068

**项目摘要：**

为实现将输入的人脸图片进行检测，并将其面部五官翻转，输出翻转后的图片

**项目设计：**

原理说明：

基本想法是要将人脸与周遭的像素差距凸显出来，因此需要提取人脸的特征，制成弱分类器，但是单个弱分类器的正确率较低（在实验样本数量下基本单个最低错误率在17~26%），因此将多个弱分类器同时使用，赋予其各自的权重系数，错误率越低，权重越高。

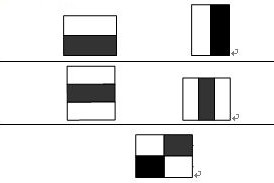
采用Haar分类器，总体组成是：5个Harr特征+积分图方法+AdaBoost算法。

仅引用了部分opencv的基本函数，如图片读入输出以及矩阵的基本功能,并未直接调用其实现好的人脸识别函数，而是（基本）全部自行实现。

代码分为训练版本以及使用版本，由FaceID.h中的宏控制

训练版本说明：

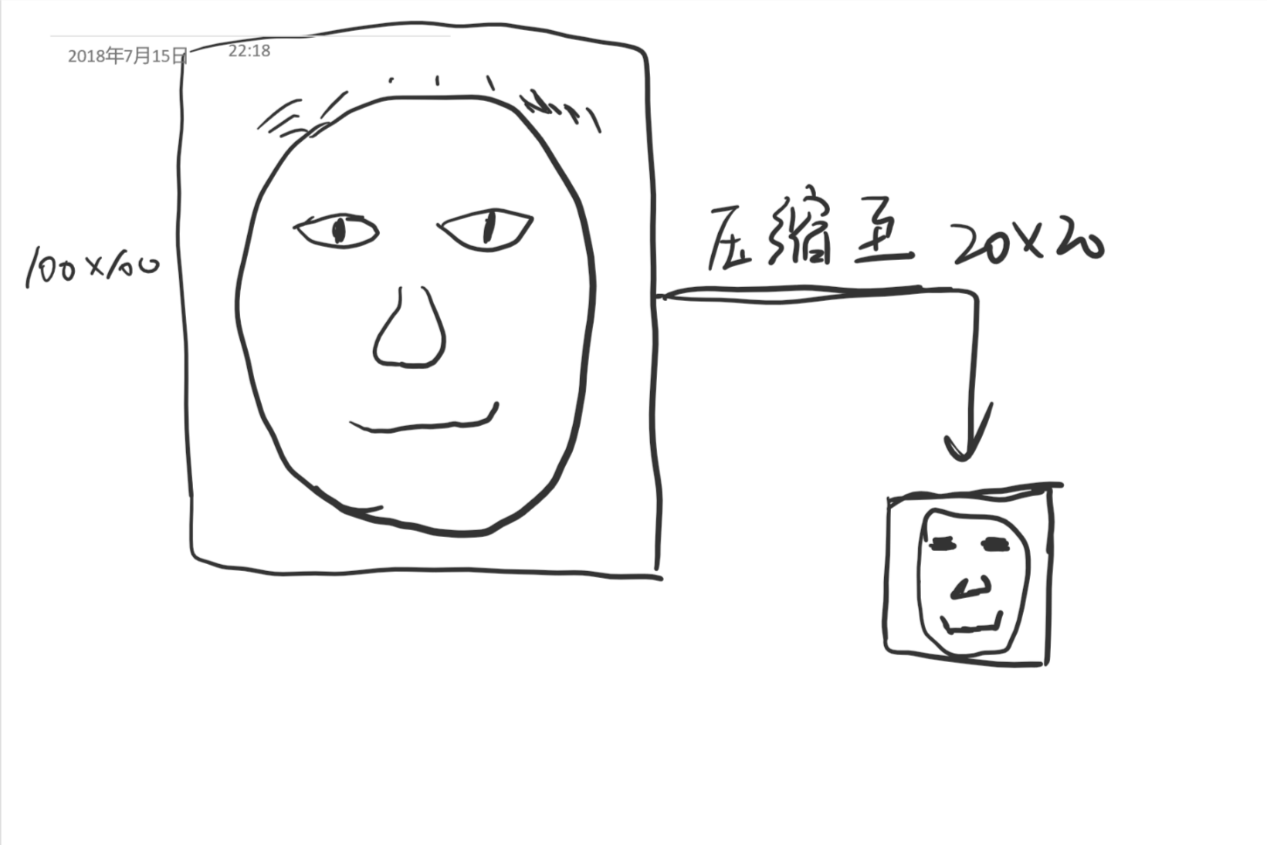
使用了五个Haar特征模型，分别如下



在FaceID.h文件中也有描述

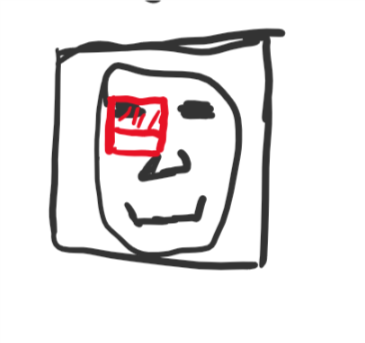
关于这些特征的作用，请看以下。

假如给定一张人脸，原图分辨率为100\*100



为了降低数据量的数量级，一律压缩至20\*20处理，在经过压缩后，图像的很多细节都被略去了，但是，由于眼睛等器官颜色较深，所以信息仍被保留，我们怎样获得这些信息呢？

如果能有一些框框住眼睛，如下：

就可以将特征提取出来，因此就有了Haar特征，我们选取了其中五个模型

由于不同图片眼睛鼻子大小，所以原则上这些框应该可大可小，而又因为不同图片脸的位置不同，这些框的位置又应该可变。

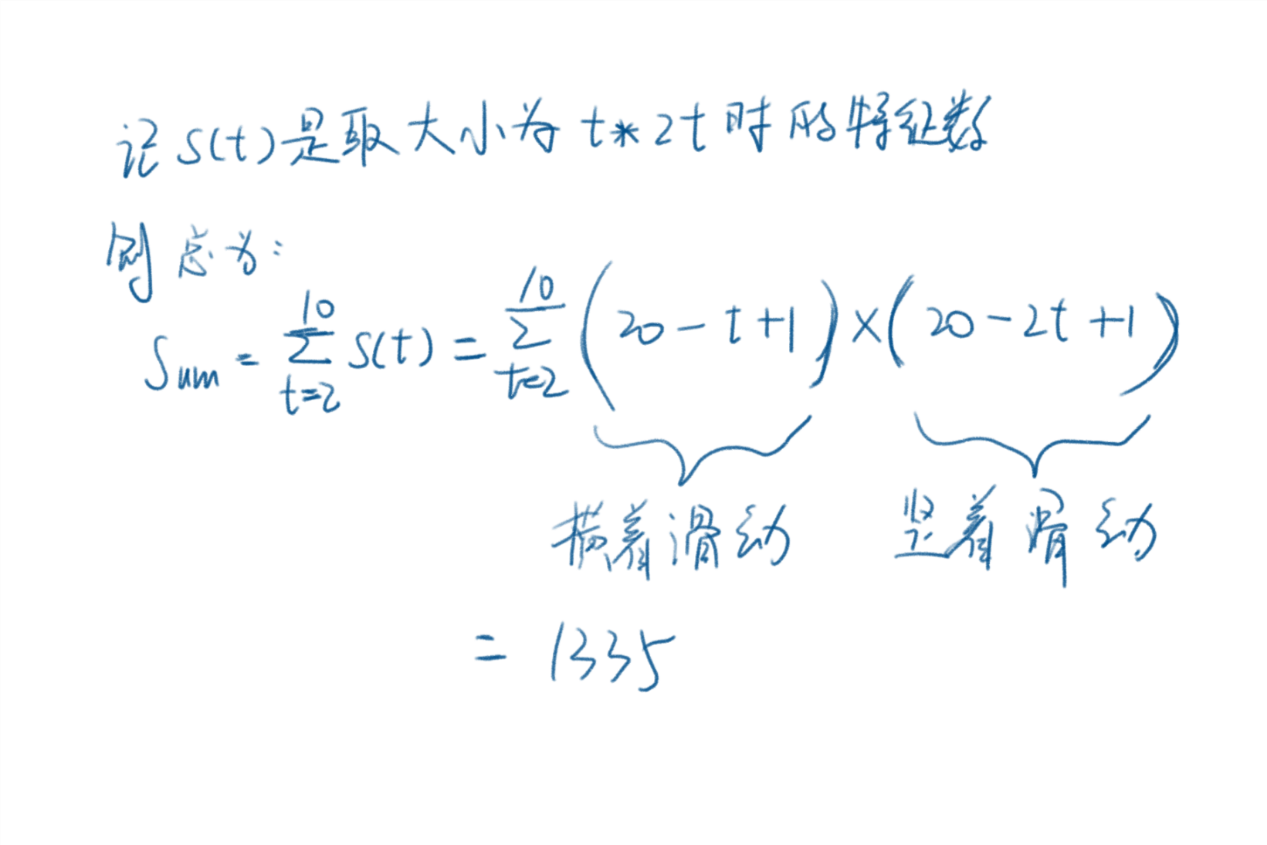
基于以上两个猜测，每个模型就会衍生出许多大小、位置不同，但相似的特征，这正是我们提取特征的依据。

那么应该怎样计算特征值呢？根据Haar特征的文献，是取阴影区域的像素减去空白区域的像素，我们也正是这么做的。

那么这样应该会有多少个特征呢？我们可以简单地计算一下第一种模型在20\*20的图片大小下的特征数

模型的横纵比为1：2，那么特征的大小可以取 2\*4 ,3\*6…….10\*20

此处不取1\*2是为了避免过拟合，对其他模型也是如此，那么如下：



对第一行中的两个模型设置最小面积为3，第二行为4，第三行为5

实际上这样取得的特征数大约有5200多个

值得注意的是，我们也尝试使用100\*100的图像来提取特征，希望能提高精度，但是那样提取出的特征数高达78万个，训练时间过长，而且由于样本数只有约2000个，得到的结果过拟合十分严重，综合考虑，仍然使用20\*20的大

基本思想是AdaBoost的迭代算法，进行50次迭代，每次迭代从所有特征中选取错误率最低的弱分类器，将其选出，并调高被该分类器分错样本的权重，进行下一次迭代，通过50次迭代，选出50个最优的弱分类器（特征），根据它们的错误率分配权重，组成一个强分类器，供使用版本侦测人脸。

代码逻辑如下：

1. 获取样本，正样本1000个，负样本1000个，由FaceID.h中的宏控制
2. 计算每个样本的积分图
3. 用20\*20的尺寸根据5个模型生成特征（约5200+）
4. 初始化一部分变量
5. 训练

训练的基本过程如下：

样本数为m

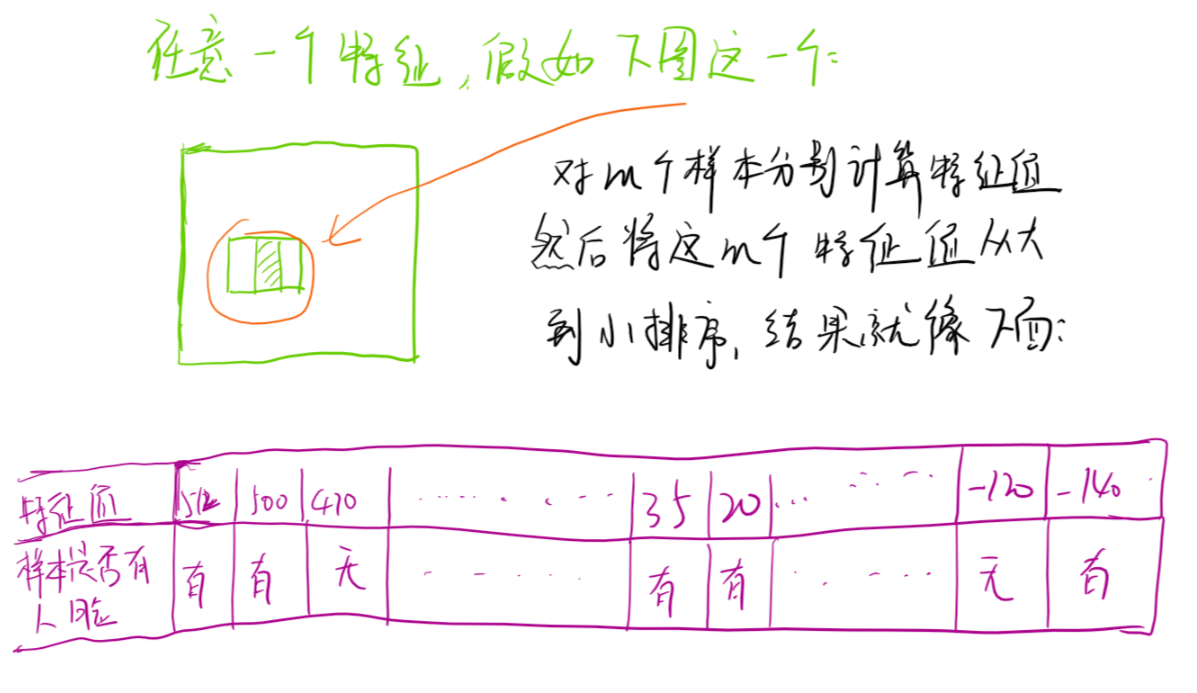
初始化所有样本的权重为1/m（由宏控制）（即平权）

进行k(设为50)次迭代，每次迭代进行如下步骤

1. 计算每个特征的最低错误率，并记录该值

（此处说明最低错误率的计算方式：

1. 计算每个样本在该特征下的特征值
2. 对这m个样本的特征值根据从高到低排序



1. 初始化minWrong=1，对这些根据特征值排好序的样本从第一个到第m个进行一次遍历

遍历过程中进行如下计算，对第i个样本

1. 更新在该样本之前所有有人脸样本（正样本）的权重和\_\_SP以及所有非人脸样本（负样本）的权重和\_\_SN
2. 更新在该样本之前所有正样本的权重和\_\_SN

计算如下值

curWrong=Min(\_\_SN+\_\_TP-\_\_SP, \_\_SP+\_\_TN-\_\_SN)，如果curWrong小于minWrong就更新minWrong并记录这个位置

（此处进行说明，我们通过遍历到底想得到什么？

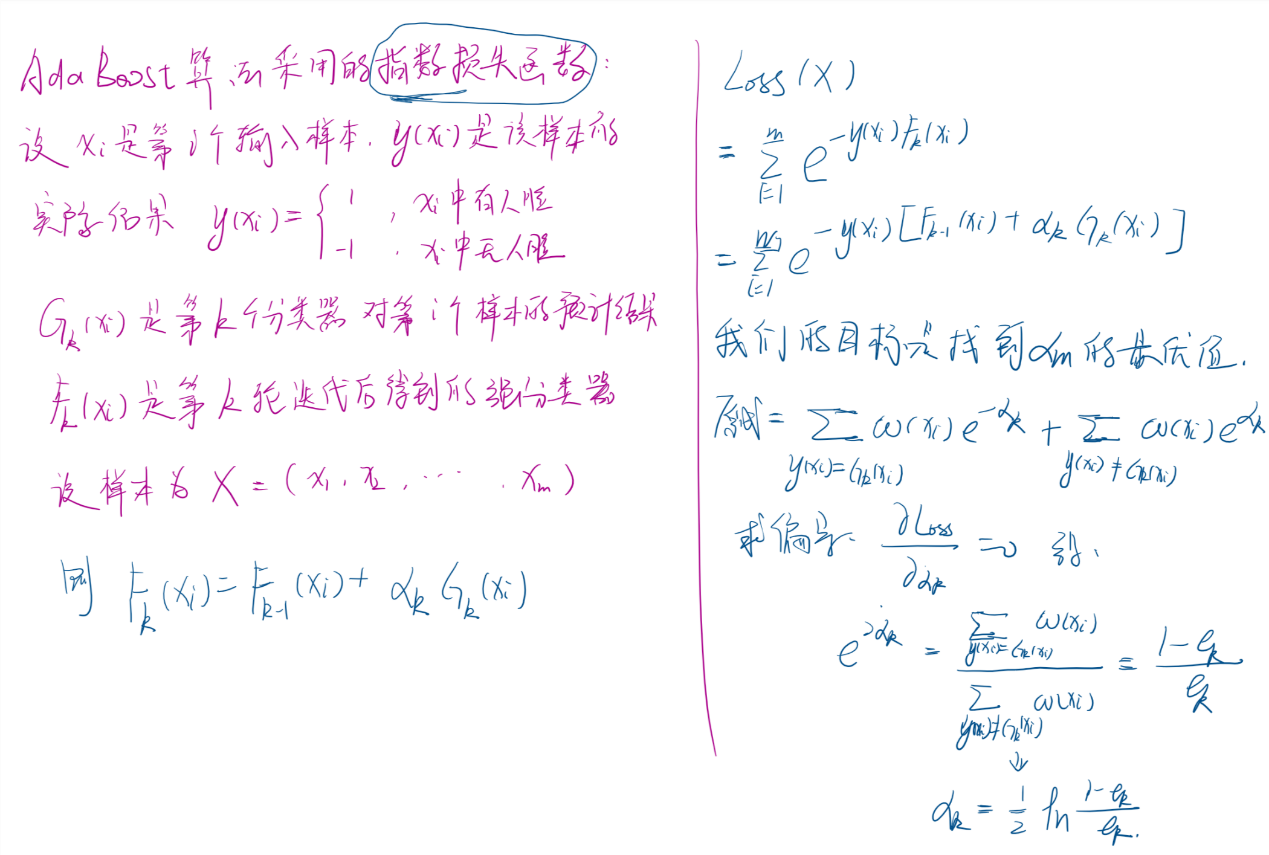
其实就是一个阈值，我们要找到这个特征的最佳阈值，我们可以说，特征值大于这个阈值的判定为人脸或者特征值小于这个阈值的是人脸。但不管怎么划，都会有一部分样本被划错，这也是为什么我们这个阶段选取的叫做弱分类器的原因了，因为它其实比较粗糙，只是根据数据拟合出了在这些数据样本下的一个最佳阈值点，拿它去判断一张新的图片。）而我们在遍历过程中计算的这个Min是以当前特征值为阈值时的错误率，我们遍历就是为了找到取得最小值的位置以及最小出错率

4. 取得最低错误率以及阈值，记录不等号方向，用p控制

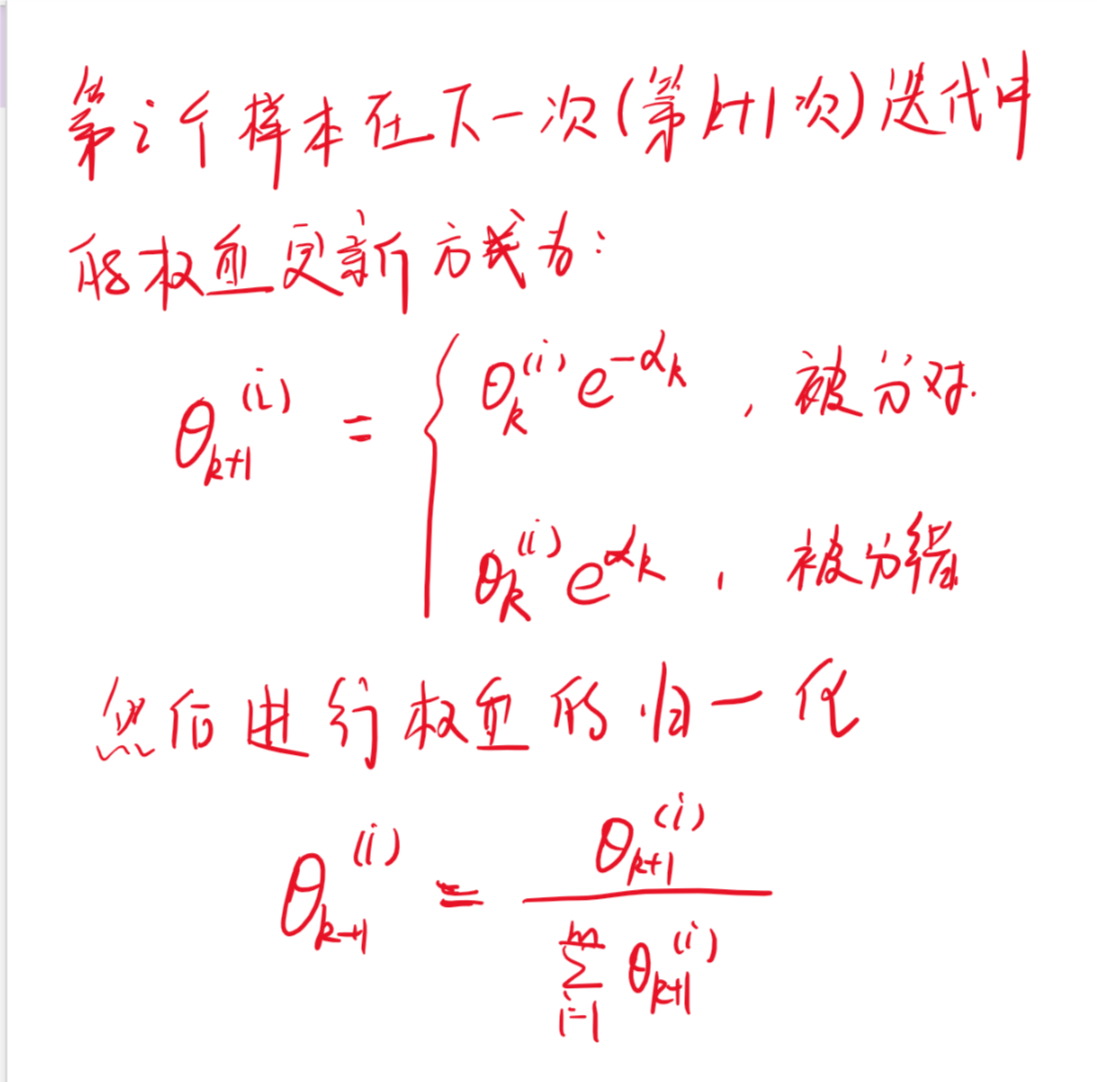
2. 对所有特征的错误率进行排序，找到最小的那个，就是这次迭代选择出来的最优弱分类器（当然，要先确认是不是已经选择过了）

然后，设这个弱分类器在最终的强分类器中的权重为

具体的推导如下：



而在下一次迭代开始之前，要进行一次权重的调控，具体的原则就是，这次没分对的下一次要着重针对，于是，这一次被分错的样本的权重要提升，被分对的样本的权重要降低，具体操作如下



这样，经过k次迭代，我们就得到了k个弱分类器组成的强分类器，这也是我们训练版本的主要任务。得到一个较可靠的强分类器。

使用版本说明：

在进行过训练后，训练版本会将得到的弱分类器全部存储在名为classifier.txt的文件中，使用版本会从中读取所有的弱分类器，classifier文件内容大致如下：

**项目功能列表：**

1. 识别一张图片中是否有人脸（以训练样本探测则基本可以识别，但输入其他人脸暂时并不可行，主要是因为训练样本较少，过拟合较严重）
2. 若输入的图片中有人脸，则将人脸中部分器官反转并输出（正确抓取率目前只有约15%）

**代码说明：**

**项目分工：**