

2022- 2023年计算机科学A项目。面部识别

项目大纲

在这个项目中，你将使用已经提供的卡兹米地标算法创建你自己版本的人脸识别。你将使用3张照片来创建一个

"基础脸"，并与其他脸进行比较。我们的目标是创造一种算法，能够唯一地识别你的脸，同时拒绝你的同学和其他人的脸。在如何实现你的人脸识别版本方面，你将有很大的自由度，但你需要使用面部矢量与面部比例相结合，因为你缺乏一个深度摄像头。

基于摄像头的人脸解锁与深度投影的人脸解锁

基于摄像头的人脸解锁仅使用摄像头来唯一识别一个人，而深度投影系统则使用红外（Infrared）点投影仪来创建一个人脸的深度图，以确定它是谁的脸。这个项目类似于一些手机实现面部解锁的方式，如谷歌Pixel

7，它只是使用摄像头来识别用户。谷歌使用先进的机器学习模型进行面部识别，但即使如此，它通常也被认为不是很安全，因为基于摄像头的面部识别系统很容易被愚弄并出错--正如你将看到的。

用于面部识别的最常见的机器学习算法类型是深度学习卷积神经网络或CNN。CNN是一种人工神经网络，非常适用于图像分类任务。CNN学习从图像中提取特征，并能识别复杂的面部特征，如下巴或眼睛的形状。

要做到苹果公司的Face

ID，你需要一个红外投影仪，在一个人的脸上投射一个红外点的网格。通过建立一个人脸的深度图，你可以准确地测量一个人的眼睛和嘴巴之间的距离，以及他们头部的宽度，出错的机会要小得多。根据苹果公司的说法，别人解锁你的iPhone的机会只有百万分之一。由于我们缺乏深度感应相机系统，为了简单起见，我们将采用更容易出错和更简单的基于相机的系统。

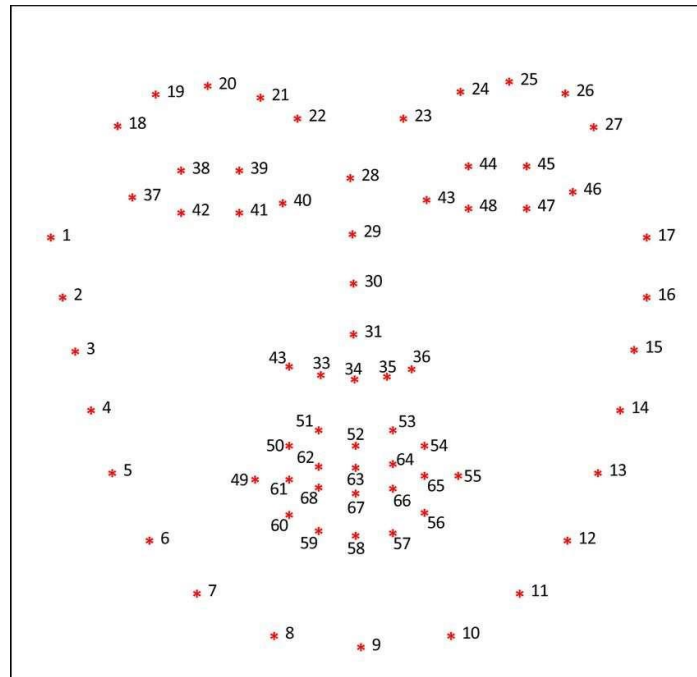


卡兹米地标算法

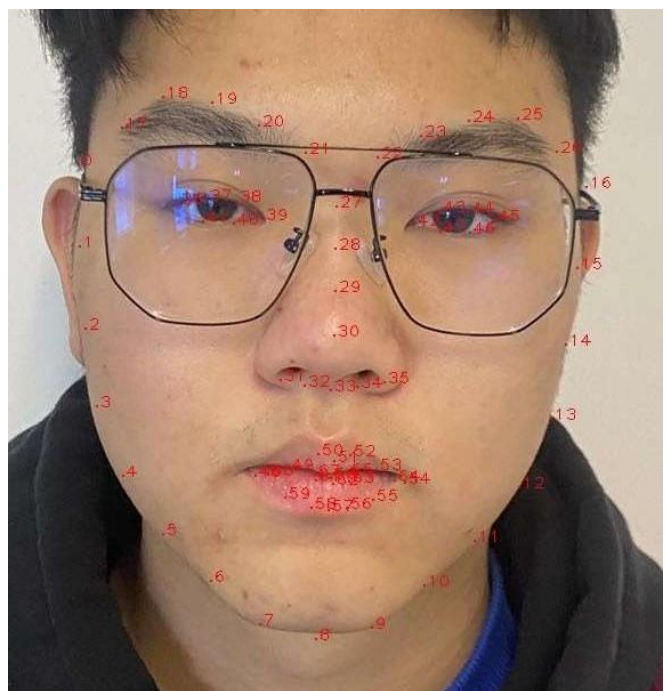
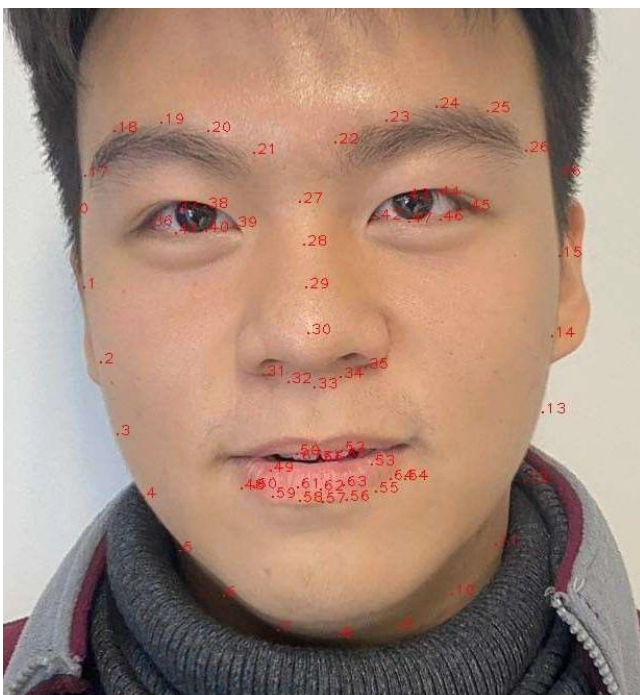
卡兹米地标算法是瑞典一所大学的算法，可以快速识别面部地标（如下图），你将在你的Face ID项目中使用它。`Facemark.java`类包含`getFacemarks(String PhotoNameAndLocation, String CreateMarkers)`方法，当给定一张照片时，这将返回一个int类型的二维数组，包含脸部标记的X和Y坐标。返回的二维数组的长度为68，代表下面图片中68个面部标志中的每一个。这68个数组中的每一个都只包含一个X和Y坐标，因此每个数组的长度只有2。

在返回的二维数组中，位于

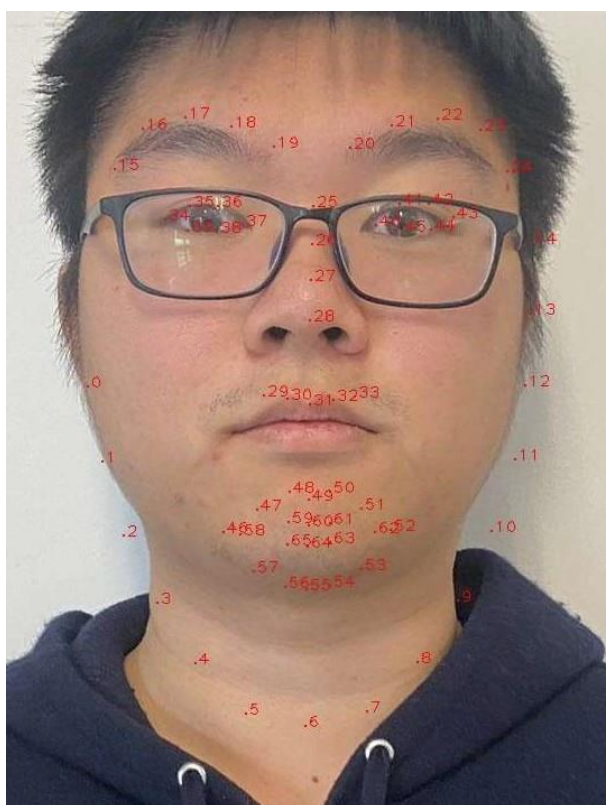
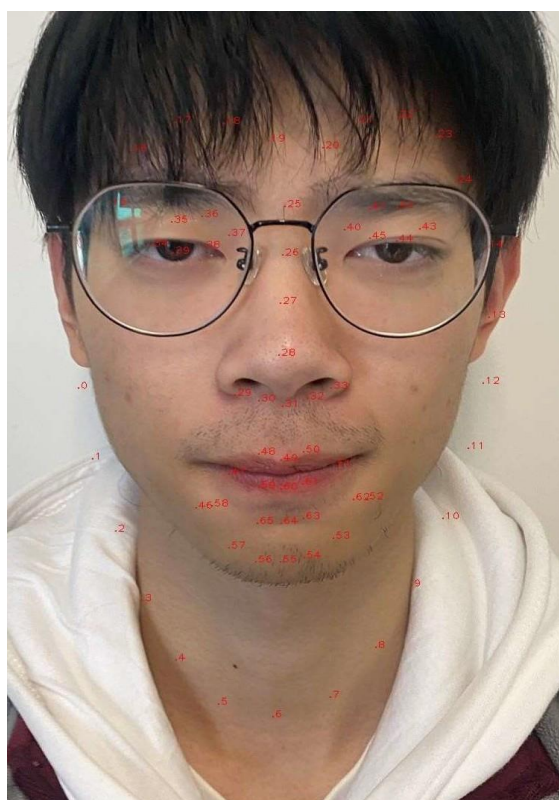
在下图中，[O]将包含点1的[X,Y]坐标（在位置[1]你会发现点2的[X,Y]坐标，以此类推）。



当你运行`getFacemarks()`并保存照片时，你应该得到一个类似这样的输出。



然而，卡兹米面部地标算法并不完美，你可能会得到奇怪的结果，将面部地标放在错误的地方，就像这样。



通常情况下，"长脸

"问题是在地标算法无法明确识别下巴的位置时出现的，在创建你的算法时，我建议更多地强调眼睛和眉毛的比例，而不是嘴巴和鼻子的位置。我发现的另一个问题是，该算法可以把不同的标记放在完全相同的照片上，在撰写本文时，我仍然不确定为什么会发生这种情况，但要注意这一点。

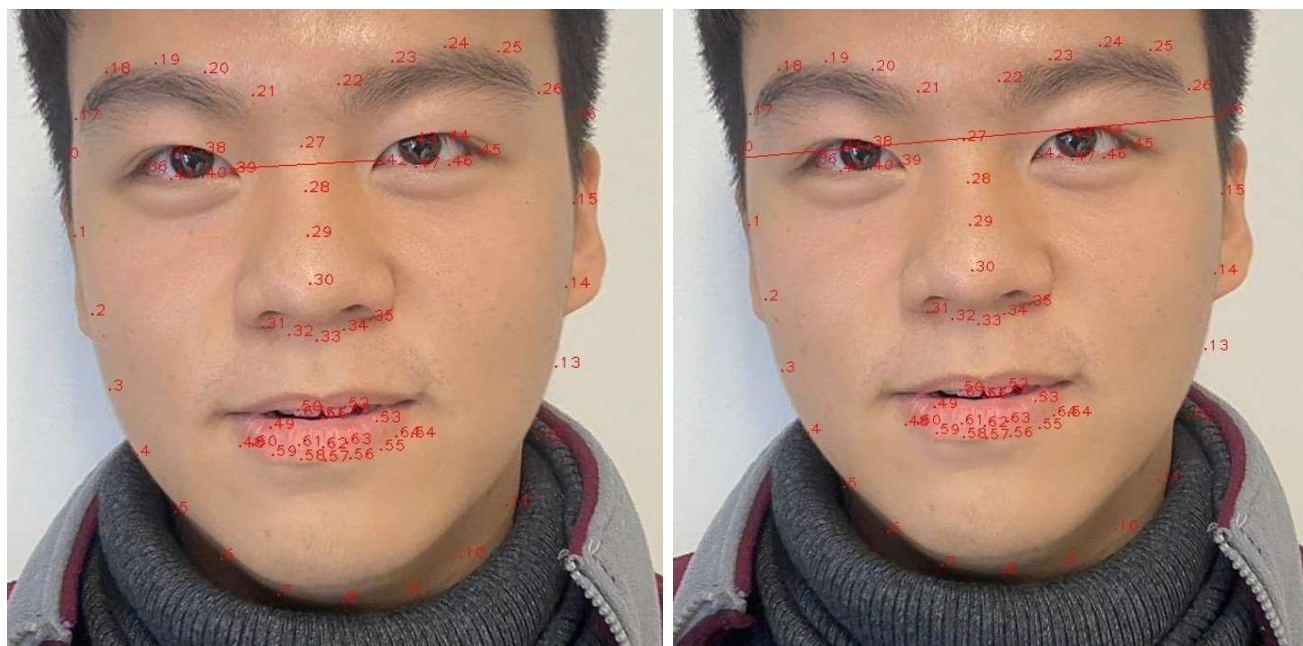
创建你自己版本的Face ID

如上所述，当使用`getFacemarks()`方法时，你会得到一个二维坐标数组，为了用它来唯一地识别一个人脸，你需要计算两点之间的距离，这可以用以下公式来完成。

$$\sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]}$$

你应该把这个公式放到一个方法中，因为你需要反复使用它。然而，请注意，给出的距离是以像素为单位的，这本身并不是特别有用，因为我们不知道相机和脸部之间的距离，也不知道脸部的宽度。你想用这个距离与其他距离进行比较，例如，脸的宽度与眼睛的宽度相比，或者鼻子的长度与脸的长度相比。

例如，你可以计算1-16点之间的距离，这代表头部的宽度，然后计算39-42点之间的距离，这代表眼睛之间的最短距离。利用这两个距离，你就可以计算出%的距离。然后你需要对几个测量值进行这样的计算，如眉毛的长度或鼻子的长度，你可以选择你认为最好的方法。在下一页，我用我们的好朋友亚伦举了一个例子。



在上图中，0号点的X和Y位置是346，667，16号点的X和Y位置是813，626，用上面的公式得出的距离是468.796。记住你从[getFacemarks\(\)](#)方法中得到X和Y坐标，在这个例子中，位置[0]的数组包含[346,667]，位置[16]的数组包含[468,796]。你可以看到下面的计算结果。

距离。	$\sqrt{(813 - 346)^2 + (626 - 667)^2}$
	$\sqrt{(467)^2 + (-41)^2}$
	$\sqrt{219770}$
	468.79633104366

你不需要成为一个数学天才，但如果你在Java中努力实现这个公式，不用担心，谷歌或百度上有很多例子。继续，39号点的X和Y坐标是497，670，42号点的X和Y坐标是638，662。再次使用我们奇妙的二维距离公式，距离是141.226，你可以在下面看到。

距离。	$\sqrt{(638 - 497)^2 + (662 - 670)^2}$
	$\sqrt{(141)^2 + (-8)^2}$
	$\sqrt{19945}$
	141.22676800097

现在我们可以很容易地计算出39-42点与1-16点的距离，即 $141.226/468.796=0.3012$ 。因此，眼桥的距离约占亚伦面部宽度的30%。现在，如果你要计算班上每个人的距离，你会发现有些人的距离会大于30%，而有些人的距离会更小。在创建你的版本的Face ID时，你要接受一些误差空间（例如接受+或-5%），同时拒绝那些比例相差很大的脸（例如+或-10%）。你需要对脸部的几个部分进行这样的处理，这样你就能确定你能选择正确的部分。请注意，为了创建上述图像，你需要改变[getFacemarks\(\)](#)，使其返回图像，而不是一个二维的int数组，为了在照片上画线，你需要使用方法[AddLineOnPhoto\(\)](#)。

项目要求

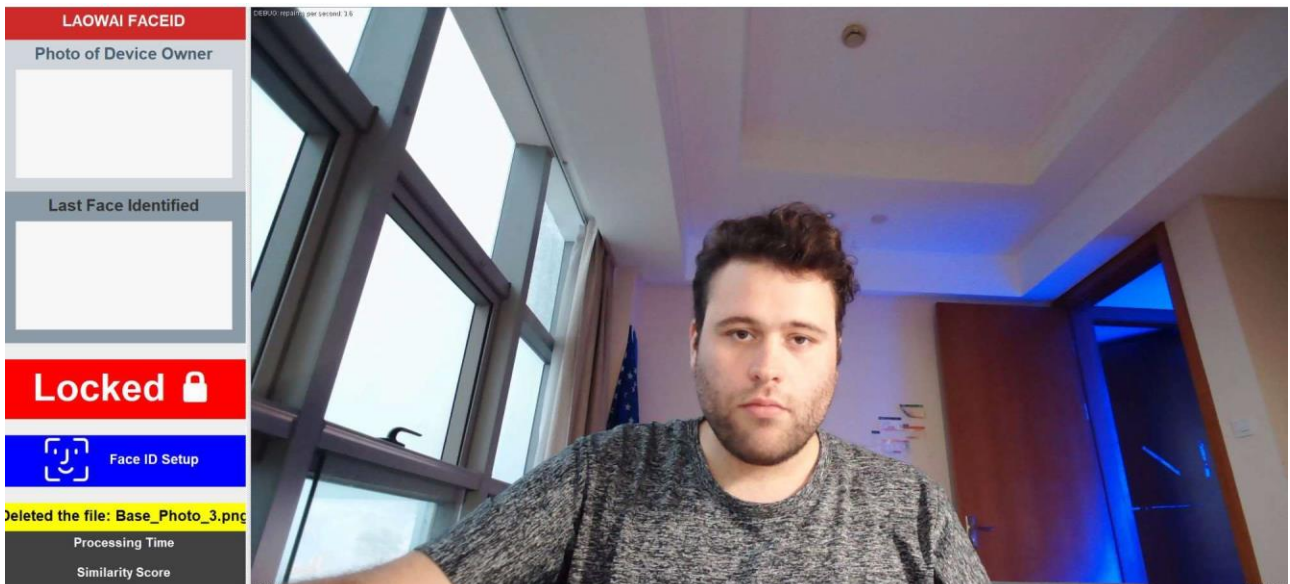
我将给你很大的自由度来进行这个项目，以便你能展示你的Java技能。我为你提供了2个类，[Facemark](#)和[Camera_GUI_System](#)，此外，我还为你提供了一个能在Mac上使用的新版OpenCV（耶！）。你的项目有两种评估方式，一种是通过向全班展示你的工作程序以及解释你的算法的PPT，另一种是类似于ANPR项目的测试，你需要对一个文件夹中的人脸图像进行分类，创建一个.csv文件，显示哪些照片通过了你的算法，哪些没有。你可以用任何方式做这个项目，唯一的要求是不要使用APIs，不要剽窃，而且你必须在你的程序中至少有一个部分包含类的继承。

1. GUI演示

你需要在课堂上使用[Camera_GUI_System](#)中提供的GUI来演示你的项目。这个GUI与ANPR项目中的GUI相似，你应该熟悉它，最大的变化是实现了"设置人脸识别"的按钮，你必须使用这个按钮以及一个"锁定"和"解锁"的面板，这个面板会根据你的算法是否成功识别你的脸而改变。



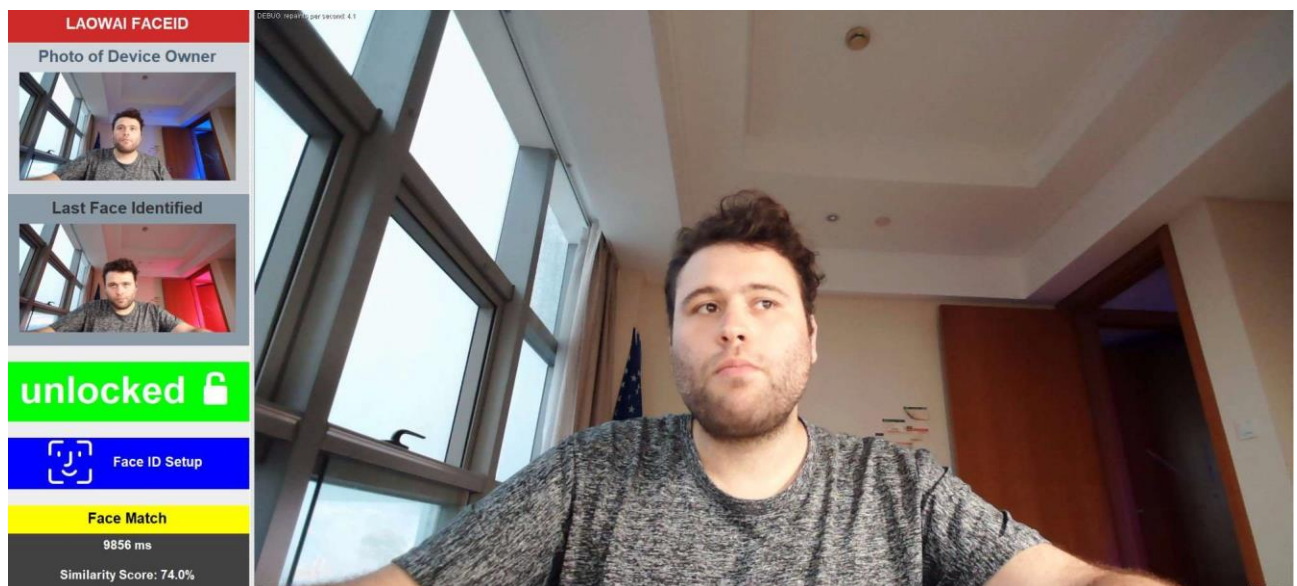
当点击'Face ID Setup'按钮时，它将删除文件夹\Photos\Base Photos中的3张照片，称为Base_Photo_1.png、Base_Photo_2.png和Base_Photo_3.png。然后它将使用网络摄像头拍摄3张照片，分别称为Base_Photo_1.png、Base_Photo_2.png和Base_Photo_3.png。你需要用这些照片来比较其他的面孔，你的算法会通过"与这3张照片匹配的面孔。你将需要创建你自己的类、对象和方法来做到这一点，例如，你可以从这3张照片中提取平均比例距离来创建你的"基础脸"，然后将其他脸与之进行比较。如果一个新的面孔与"基本面孔"相似，那么你就通过它，否则就拒绝它。



一步一步地指导图形用户界面应该如何工作

阶梯	行动
1	启动程序，GUI加载
2	用户按下'Setup Face ID'按钮，程序会删除3张照片，然后拍摄3张照片，分别称为Base_Photo_1.png、Base_Photo_2.png和Base_Photo_3.png。你需要使用一个while循环来等待这些照片的拍摄，我在Camera_GUI_System类中加入了变量BaseFaceSet。使用一个while循环来等待，直到这个变量被设置为真。
3	当第二步完成后，你需要在你的算法中使用这3张照片，这样其他的人脸照片就可以和它进行比较。将每张照片发送到getFacemarks()方法，以获得脸部标记数据，然后例如，你可以使用保存脸部的平均比例或最大和最小比例。你可以创建一个类，为每张照片生成所有的脸部比例数据，然后创建一个"基本脸部"对象来存储这些数据--但这又是取决于你的想法
4	然后设置一个for循环，就像你在ANPR项目中看到的那样，使GUI在一定的时间内持续拍摄照片。对于每张照片，你要检查其中是否有一个人脸，如果有，并且你的算法确定照片中的人脸与基础人脸照片足够相似，那么使用Camera_GUI_System类中的方法updateLockedStatus (Boolean Status)将 "锁定/解锁" 面板设为"解锁"。否则就将面板设置为'锁定'。

你需要对它进行设置，使程序自动地只是不断地拍摄照片（使用do-while循环），对于拍摄的每张照片，都使用你的算法和getFacemarks()方法进行处理。在使用Camera_GUI_System类拍摄每张照片后，你需要将照片发送到getFacemarks()，如果返回空，则将面板设置为"锁定"，否则你需要使用你的算法来检查当有人按下"Face ID设置"按钮时，拍摄的照片是否与拍摄的3张基础脸部照片（Base_Photo_1.png等）足够相似。如果你的算法拒绝该脸部照片与基础脸部照片不够相似，那么将更新面板设置为"锁定"，否则你需要将更新面板设置为"解锁"，像这样。



你需要向全班同学做一个现场演示，展示你的算法的工作情况，不过不用担心会出错，因为面部识别不是一门完美的科学，容易出错。即使是苹果公司的Face ID也不是完美的!你的图形用户界面演示和PPT将由你的同学而不是我来评分。

2. 处理一个图像的文件夹

这将非常类似于ANPR项目，我将给你一个照片文件夹，你的算法必须决定哪些照片包括你的照片，哪些照片不包括。你必须使用之前提到的基本脸部照片来设置你的面部识别算法，所以你最多只允许有3张照片，所有其他脸部都要与之比较。

当运行你的算法时，你需要生成一个.CSV文件，用来标记你的工作。对于每张照片，你必须包括照片编号、相似度分数、合格分数、照片是否匹配以及处理时间。处理时间是指你的算法决定脸部是否匹配所需的时间--

但不包括拍摄照片或其他的时间。你可以通过使用Camera_GUI_System类中的addFaceIDData方法生成.CSV文件

就像在ANPR项目中一样。这次我没有给你提供代码，所以你需要制定方法来运行这个测试，但是我会检查你的代码，这样我就知道你是在诚实地做这件事--

如果有任何作弊的意图，我将给你整个项目打0分。输出应该类似于你看到的以下内容。

	A	B	C	D	E	F
1	Photo Name	Photo Number	Similarity Score	Similarity Passing Mark	Lock or Unlock	Processing Time
2	Aaron1.jpg	1	6	45 No Match		2756
3	Aaron2.jpg	1	36	45 No Match		2161
4	Aaron3.jpg	1	0	45 No Match		2338
5	Charlie1.jpg	1	36	45 No Match		1772
6	Charlie2.jpg	1	0	45 No Match		1867
7	Charlie3.jpg	1	27	45 No Match		2261
8	Charlie4.jpg	1	28	45 No Match		2001
9	Edwin1.jpg	1	0	45 No Match		2163
10	Edwin2.jpg	1	0	45 No Match		2210
11	Edwin3.jpg	1	12	45 No Match		2646
12	Edwin4.jpg	1	0	45 No Match		3083
13	Edwin5.jpg	1	3	45 No Match		2896
14	Evelyn2.jpg	1	39	45 No Match		2503
15	Evelyn3.jpg	1	36	45 No Match		2126
16	Evelyn4.jpg	1	55	45 Face Match		3164
17	HappyTim1.jpg	1	11	45 No Match		2761
18	HappyTim2.jpg	1	44	45 No Match		2749
19	HappyTim3.jpg	1	31.5	45 No Match		3047
20	Jony1.jpg	1	39.5	45 No Match		2497
21	Jony2.jpg	1	42	45 No Match		2363
22	Jony3.jpg	1	13	45 No Match		1930
23	Laoban1.jpg	1	0	45 No Match		2111
24	Laoban2.jpg	1	12	45 No Match		2609
25	Laoban3.jpg	1	10	45 No Match		2278
26	Leo1.jpg	1	3	45 No Match		2047
27	Leo2.jpg	1	11	45 No Match		2537
28	Leo3.jpg	1	24.5	45 No Match		2235
29	Leo4.jpg	1	5	45 No Match		1992
30	Zarc1.jpg	1	17	45 No Match		2014
31	Zarc2.jpg	1	13.5	45 No Match		2038
32	Zarc3.jpg	1	20.5	45 No Match		2215
33	Zarc4.jpg	1	44	45 No Match		2350
34	Zarc5.jpg	1	12	45 No Match		2267
35	Zarc6.jpg	1	39.5	45 No Match		2240
36	Zarc7.jpg	1	5	45 No Match		2112
37						

正如你所看到的，我的算法拒绝了你所有的脸，除了Evelyn，你的目标是使算法能够拒绝尽可能多的脸，但不拒绝你自己的照片。在做项目的时候，如果你的算法拒绝自己的照片，你会受到惩罚，你要允许假阳性。我还会用其他人作为基本面孔来测试你的算法，比如名人或你的同学，所以你要做一个不只对你自己有效的算法，而是可以普遍应用的算法。

为了帮助你，我提供了大约10,000张人脸图片，你可以用来测试你的算法，使用其中一部分你认为对测试你的算法有用的图片，100-

200是个好数字，然后创建一个方法，可以计算出你的Face

ID算法通过和拒绝的照片的百分比。要找到能从一堆照片中准确选择你的脸的正确算法，需要很多时间。

这个项目将如何打分？

1. GUI演示

你需要在课堂上演示你的项目，展示你的算法对你的脸进行解锁，但对你的同学却没有解锁（不要担心错误！）。你还需要做一个PPT，解释你所选择的面部特征来设置你的Face ID版本，重点介绍你测试算法的经验。你的演示将由你的同学打分，平均分将成为你本节的成绩。这将占到你最终成绩的35%。

2. 正确率（处理一个文件夹的图像）

你将得到一个你以前没有见过的照片文件夹，这些照片将被用来测试你的算法，用不同的基本面孔测试几次。你将根据准确率获得这一部分的分数，如果你的程序拒绝了一张基本面孔的照片，将受到惩罚。同样，在这里你希望有假阳性，因为在现实中你不希望让一个人无法解锁面部识别系统。在Camera_GUI_System类中生成的.csv文件将被用来给你的作品打分，所以你必须正确实现它。这将是你的项目成绩的25%。

3. 处理时间（处理一个文件夹的图像）。

与第2项类似，只是在这里你将根据你的程序处理一张照片并决定它是否与基本面相匹配所需的时间来评分。如果你犯了错误，你会受到时间上的惩罚，如果你的程序拒绝了它应该接受的脸部照片，则会受到更大的惩罚。你将需要实施这个测试系统。你的处理时间将与你的同学进行排名比较，最快的人将获得100%的成绩。这将是你的项目成绩的25%。

4. 代码的质量

当项目完成后，你将被要求打印出你的代码，这将被标记为质量问题，例如，你是否对你的代码进行了注释，它是否包括不必要的函数或奇怪的变量名称。对于一个大型项目，你应该明白高质量的代码的重要性，否则事情会很快变得混乱。这应该是一个简单的15%。

	占最终项目成绩的百分比	小宝独立分级
GUI演示	35%	是
正确率百分比	25%	是
处理时间	25%	是
代码的质量	15%	是
整个项目	100%	是

记住：如果你不明白什么，不要担心！！。
请提出问题，我是来帮助你的！！。