1 引言

1.1 编写目的

题目“猜猜我是谁”要求设计一个移动 APP，模拟实现无密码登录的功能。而当前人脸识别、指纹识别及声纹识别等身份识别技术对智能手机处理器性能要求较高，且这些身份识别技术也日趋成熟。鉴于此，我们提出了另外一种身份识别方式，即通过提取、分析用户的行为特征来判断当前用户是本人还是陌生人。目的是使用户在浏览有内容的 App 时（论坛 App，购物 App，新闻 App）能够不知不觉完成登录而无需记住和输入复杂的密码，高效准确的完成登录。

1.2 可行性研究

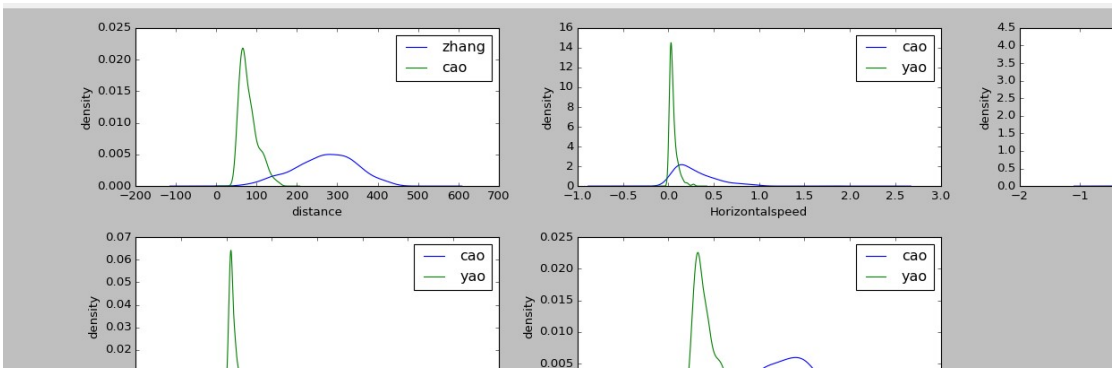

图 1 特征分布图

图 1 是距离，水平速度，垂直速度，水平距离，垂直距离的分布图，这些都是人浏览新闻时所产生的行为特征，纵坐标是密度，横坐标是值，从图中可以看出，不同人的距离，水平速度等特征量分布完全不同，且满足正太分布，特征的组合完全可以作为识别的依据。

2 总体架构

如图 2 所示，用户在注册时，将会被要求浏览 10 分钟的新闻，App 会在这 10 分钟内收集用户的行为特征数据，并在一系列的数据清洗和特征处理后存入数据库。用户在登陆时，将会被要求浏览 1 分钟的新闻，系统会在浏览时保存用户数据，但不存入数据库。1 分钟结束时，系统会根据注册时的数据训练模型，用以对登陆时产生的未标注数据进行分类，分类结果产生后，由一系列算法进行判别处理，最后返回判断结果。

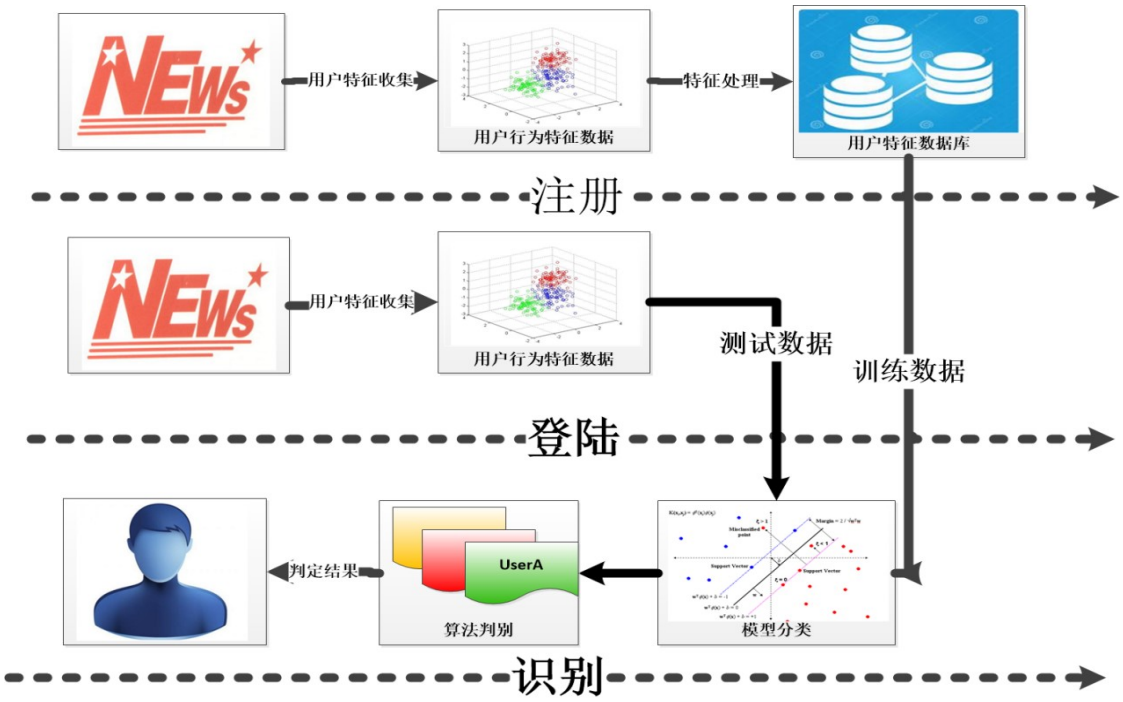

图 2 整体概要图

3.详细设计

本系统大概包括以下几个模块，新闻模块（基于开源新闻 App），滑动行为特征收集模块，滑动行为特征值处理模块，分类模型的构建以及分类决策。

3.1 滑动事件定义

为了有效的描述滑动事件，我们把一次滑动事件定义为 4 元组的形式，即

$$\text{Gesture}=<type,time,(x_1,y_1),(x_2,y_2)> \quad (1)$$

其中 type 为滑动的类型，有上滑、下滑、左滑、右滑、四种情况，time 为持续的时间，(x_1,y_1) 为滑动开始时手在屏幕的位置，(x_2,y_2) 为滑动结束时手在屏幕的位置。

3.2 滑动行为特征值及其计算方法

对于滑动的行为特征值，我们可以将它分为两大类，一种是可以原始数据，不需要做过多处理，比如滑动时间，滑动水平距离，滑动间隔时间等等，我们把这些特征叫做初级特征，一种是需要二种甚至多种原始数据的运算而成，我们把这些数据叫做高级特征，比如滑动速度，滑动角速度等等。

(A)初级特征

1. 滑动水平距离：

$$Horizontaldis\tan ce = x_2 - x_1 \quad (2)$$

其中 x_2 是滑动结束时手在屏幕的水平位置，x_1 是开始结束时手在屏幕的水这位置。

2. 滑动垂直距离：

$$Verticaldis\tan ce = y_2 - y_1 \quad (3)$$

其中 y_2 是滑动结束时手在屏幕的垂直位置，y_1 是开始结束时手在屏字的垂直位置.

3. 滑动持续时间：time
4. 滑动间隔时间：

$$Inte = gesturetime_{k+1} - gesturetime_k \quad (4)$$

其中 gesture$time_{k+1}$ 是后一次手势发生的时间，gesture$time_k$ 是前一次手势发生的时间

(B)高级特征

1. 滑动水平速度：

$$Horizontalspeed = Horizontaldis\tan ce / time \quad (5)$$

滑动水平速度用滑动水平距离除以持续时间得到。

2. 滑动垂直速度：

$$Verticalspeed = Verticaldis\tan ce / time \quad (6)$$

滑动垂直速度用滑动垂直距离除以持续时间得到。

3. 整体速度：

$$speed = \sqrt{\text{Verticalspeed}^2 + \text{Horizontalspeed}^2} \quad (7)$$

整体速度由垂直速度和水平速度的平方和开根号得到。

4. 整体距离：

$$\text{dis tan ce} = \sqrt{\text{Horizontaldistance}^2 + \text{Verticaldistance}^2} \quad (8)$$

整体距离由水平距离和垂直距离的平方和开根号得到

5. .加速度：

$$Acceleration = speed / time \quad (9)$$

加速度由速度除以时间得到。

6. 加加速度：

$$Jerk = Acceleration / time \quad (10)$$

加加速度由加速度除以时间得到

7. 角度：

$$Angle = \arctan(Verticaldis\tan ce / Horizontaldis\tan ce) / \pi * 180 \quad (11)$$

角度由对垂直距离和水平距离的商取反三角函数除以 π 乘以 180 得到。

8. 角速度：

$$AngleSpeed = Angle / time \quad (12)$$

角速度由移动角度除以时间得到。

9. 角加速度：

$$AngleAcclerate = AngleSpeed / time \quad (13)$$

角加速度由角速度除以时间得到。

10. 角加加速度：

$$AngleJeck = AngleAcclerate / time \quad (14)$$

角加加速度由角加速度除以时间得到。

3.3 滑动行为特征汇总

滑动特征有四个种类，分别是上滑，下滑，左滑，和右滑，每一种都有上述13个特征，现在将所有特征汇总如下

表 1 特征汇总图

方向	滑屏过程中收集的特征												
上滑	时间	距离	速度	水平距离	垂直距离	水平速度	垂直速度	加速度	加加速度	角度	角速度	角加速度	角加加速度
下滑	时间	距离	速度	水平距离	垂直距离	水平速度	垂直速度	加速度	加加速度	角度	角速度	角加速度	角加加速度
左滑	时间	距离	速度	水平距离	垂直距离	水平速度	垂直速度	加速度	加加速度	角度	角速度	角加速度	角加加速度
右滑	时间	距离	速度	水平距离	垂直距离	水平速度	垂直速度	加速度	加加速度	角度	角速度	角加速度	角加加速度

3.4 滑动特征收集模块

3.4.1 注册时数据收集

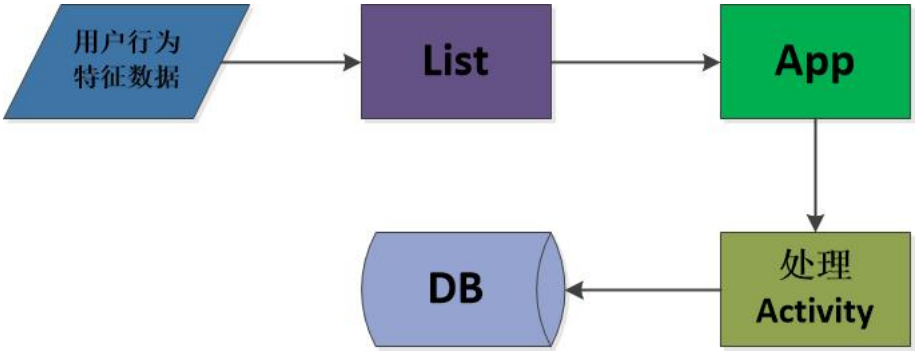

图 3 注册数据收集

由图 3 所示用户数据首先存储在临时的 List 里，然后存入安卓系统保存应用数据的 App 类内，最后由处理 Activity 从 App 内取出特征处理完毕后存入 Sqlite 数据库中。

3.4.2 登陆时数据收集

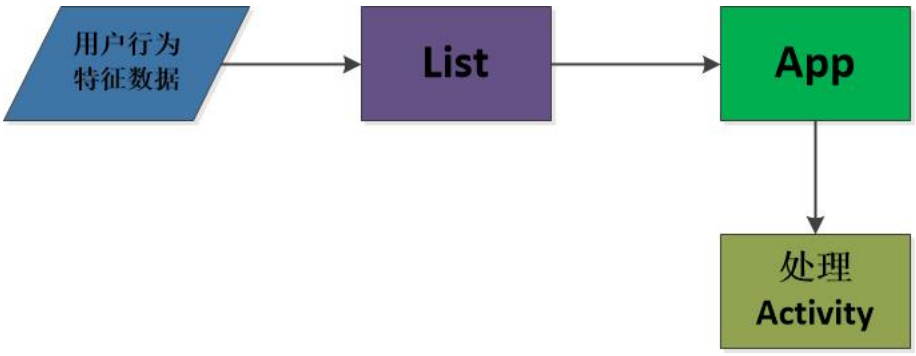

图 4 登陆数据收集

由图 4 所示用户数据首先存储在临时的 List 里，然后存入安卓系统保存应用数据的 App 类内，最后由处理 Activity 从 App 内取出特征处理。

3.5 滑动特征处理模块

首先对数据进行预处理，因为用户在浏览的过程中并不是所有时间都在全身灌注浏览新闻，有时会出现滑动行为间隔时间过长的问题。也有时候，由于屏幕有水或者其他情况，造成对滑动距离的测量不准确，这就需要对于数据选取一个合适的区间用来训练数据。我们在利用 java-ml 的 filter 对数据进行筛选。

其次，因为获得的数据都是连续值，必须对数值进行离散值处理，方便模型更快的对数据进行训练。我们用 java-ml 的 RecursiveMinimalEntropyPartitioning 方法对特征进行离散化处理。

最后，还要对特征进行归一化，加快模型分类的速度。我们用 java-ml 的 NormalizeMean 方法进行归一化。

具体情况如图 5 所示

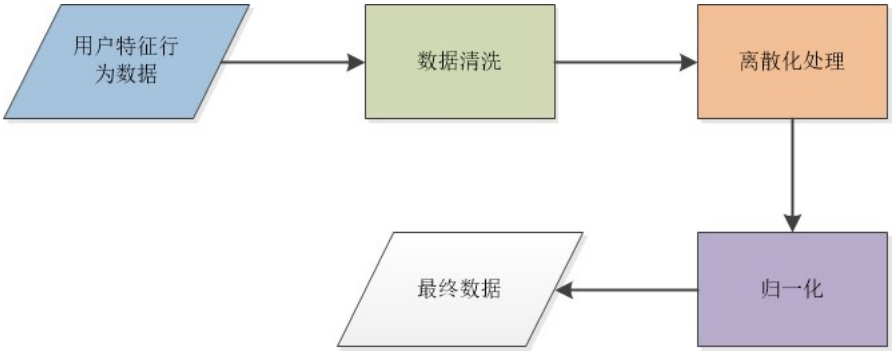

图 5 数据预处理

3.6 基于随机森林的分类器模块构造

本节利用基于随机森林的方法来构建分类器，进而达到对训练阶段收集到的用户滑动行为进行分类判断的目的。在本方案中，每一种滑动行为对应一个随机森林分类器，每一种类型的滑动行为数据通过矩阵的形式表现出来，矩阵的每列对应于这个鼠标行为的一个特征类型，矩阵的每一行对应一个用户该类型行为的实例即特征值向量，每一行数据通过用户 I D 进行标识。部分矩阵如下图所示

userId	Inte	speed	Horizontalspeed	Verticalspeed	Horizontaldistance	Verticaldista
cao	1314	1.829674766336054	0.7857142857142857	1.652380952380952	165	
cao	1503	2.002569569197053	0.9513513513513514	1.762162162162162	176	
cao	1784	1.511695873382868	0.6306306306306306	1.373873873873874	140	
cao	1235	1.808873262798515	0.8214285714285714	1.611607142857143	184	
cao	1875	1.247700662374635	0.4854771784232365	1.149377593360996	117	
cao	1242	2.186853188900914	1.005263157894737	1.942105263157895	191	
cao	753	1.778878275776833	0.8480392156862745	1.563725490196078	173	
cao	712	0.2907360715994519	0.07366071428571429	0.28125	66	
cao	867	0.3018081587386899	0.1717171717171717	0.2481962481962482	119	
cao	2819	0.2577871112146357	0.1060070671378092	0.234982332155477	60	
cao	5640	0.08048169016245672	0.0313588850174216	0.07412100095026924	99	
cao	2471	0.4072605938964657	0.160427807486631	0.374331550802139	90	
cao	1074	0.3575289386696328	0.1563421828908555	0.3215339233038348	106	

图6分类矩阵

如上图所示，userId 为用户的编号，Inte 为间隔时间等，每一行代表用户的每一次滑动。行数 n 代表滑动的总次数，列数 m 代表特征的个数，这些一起构成了一个 n*m 的特征矩阵。对于四种滑动，每一种都对应着一个随机森林分类器，获得的结果作为最后算法判别模块的输入。

3.7 算法判别模块

由于 java-ml 的随机森林算法每一次分类返回的是用户名，所以为了方便论述，我们把随机森林返回的用户名次数和占总次数的比叫做分类比.比如数据库里已经注册的人数为 2，用户 ID 是 cao 和 zhang，返回 cao20 次，返回 zhang10 次，则 cao 的分类比就是 0.66。由于在实验中，当注册人数超过三人时，分类比很难稳定在一个数值附近，这样不太好做登陆判别，而注册人数为 2 人时，分类比为 0.8 时，识别结果很准确。这样当注册人数超过三人时，我们就采用两两组合的方法，当有一个分类器有大于 0.8 的分类比产生时，我们就把该用户存入 list，如果没有，则记为错误，记录错误次数。最后如果分类器错误次数大于等于总次数的 50%，则判定为未注册，如果返回的用户次数少于该用户比较次数减一次，则返回该用户，否则判定为未注册，具体算法为算法 1 所示。

算法 1：用户判别算法

输入：用户登陆数据 testUpInfo,testDowninfo,testLeftInfo, testRightInfo, 用户数目 n，

输出：判别结果

1:while(C_n^2)

2: trainUp = idm.queryTwoGestrue("UpInfo", C_n^2);

3: trainDown = idm.queryTwoGestrue("DownInfo", C_n^2);

4:trainLeft = idm.queryTwoGestrue("LeftInfo", C_n^2);

5: trainRight = idm.queryTwoGestrue("RightInfo", C_n^2);

6: resultUp = Classer(trainUp, testUp);

7: resultDown= Classer(trainDown, testDown);

8: resultLeft = Classer(trainLeft, testLeft);

9: resultRight = Classer(trainRight, testRight);

10:resultTotal = merge(resultUp, resultDown, resultleft,resultright);

11: String results=idc.identification(resultTotal);

12: List userhit;

13：if(results.equals("none"))errornum++;

14：else userhit.add(results);

15:endwhile

16：if(errornum/totalcount>=0.5)return "请注册";

17：else return findmaxtime(userhit);

其中 queryTwoGestrue 方法是从数据取出两个用户的滑动数据，identification 方法用来判别哪位用户(返回分类比为 0.8 以上的用户，否则返回 none)，findmaxtime 返回出现最多且出现次数多余该用户比较次数减 1 次的用户 ID，否则返回请注册。

5.实验

手机选用红米 3s 进行测试，实验对象为 10 人其中 5 人已注册，5 人未注册，实验结果如下表所示

表 2 实验结果

用户 ID	是否注册	成功次数	失败次数	成功率
Cao	是	5	0	100%
Yao	是	4	1	80%
Ya	是	5	0	100%
Chen	是	4	1	80%
Zhang	是	3	2	60%
无	否	4	1	80%
无	否	5	0	100%
无	否	5	0	100%
无	否	4	1	80%
无	否	4	1	80%

其中注册平均识别率达到 84%，未注册识别达到达到 88%，取得了不错的效果。

6.软件界面

本用户识别 APP 使用 Android 平台上一个开源的新闻快讯客户端，并在其中添加注册和登录模块，而注册和登录模块分别添加了我们所使用的相应算法。在 Android 手机上打开该 APP，会进入如图 6 所示的界面。

图 6 APP 首界面

下面对注册模块和登录模块分别做详细阐述。

（1） 注册模块

当用户 A 点击注册按钮后，APP 会进入新闻快讯的首页，如图 7 所示。

图 7 新闻快讯首页

（2）登录模块

登录的过程也即识别的过程，登录模块分两种情况。

情况一：已注册的用户 yao 把该 APP 交给陌生用户 I，以判断用户 I 是陌生人。当用户 I 点击登录按钮后，APP 同样会进入新闻快讯的首页。用户 I 根据自己的兴趣爱好浏览新闻，同样，APP 在此过程中会收集用户 I 的行为特征，然后根据登录模块中相应的算法，对用户 I 进行识别，由于用户 I 是陌生人，故识别结果是“请注册”，如图 8 所示。

图 8 陌生人识别结果

情况二：已注册的用户 yao 用该 APP 进行登录，以验证用户 yao 是本人。用户 yao 点击登录按钮并进入新闻快讯的首页后，其可以根据注册时的兴趣爱好对新闻进行浏览，同样，APP 在此过程中会收集用户 yao 的行为特征，然后根据登录模块中相应的算法，对用户 yao 进行识别，由于用户 yao 是本人，故识别结果是“亲爱的 yao，欢迎登陆”，如图 9 所示。

图 9 本人识别结果