

# ShakelN 安卓项目文件相关说明

由于开发时并未注意编程文件的分包存放，因而文件可能较为散乱，主要文件分类说明以及应用接口介绍如下：

(1) MainActivity.java 为主界面文件主要涉及查看是否是第一次进入 APP，是的话转到应用介绍的滑动页面。如果还未设置密保口令则直接进入密保口令设置界面。设置过密保口令后进入主菜单页面。这些分支判断在 onCreate 函数中实现。

主菜单页面主要要注意 switch 按钮的设置。我们通过将 switch 按钮状态写入文件来保存用户的设置，通过读取文件的方法来确定锁屏应用是被用户启动还是关闭。

主要接口：

boolean isFirstRun() //第一次运行则返回 true，否则返回 false

void createFile() //在 SD 卡中创建 shakelN 文件夹

Boolean isLockingOpened() //若用户开启锁屏应用则返回 true，否则返回 false

void setLockingStatus() //设置 switch 按钮状态

(2) VibratorUtil.java 为实现手机震动工具类。

(3) GuidePagerAdapter.java 与 welcome.java 为初次登陆的介绍滑动页面。

(4) LockView.java,lockDialogView.java,lockDialog2View.java 分别为锁屏界面，锁屏界面对话框 1，锁屏界面对话框 2。（均为悬浮框，加载与取消在 lockService.java 中实现）

(5) Tuple.java 为震动数据的存储格式，以及赋值函数，相关计算的函数。

(6) Adapter.java 与 Size.java 为用户摇晃数据获取界面。通过 animate() 中 dashedCircularProgress.setValue()来修改进度条数据。界面动画效果可以通过修改 size.xml 来改变。

(7) svm\_train.java 为 svm 训练类。

官方接口：

svm\_train.main(String argv[]) //argv[] 中包含三个字符串分别为 options , input\_file\_name,model\_file\_name

由于官方接口都是以文件格式进行输入输出，在 android 中数据的存储与获取都很耗时，因而我自定义了一些接口：

my\_parse\_command\_line(string[] options) //解析训练参数

My\_read\_problem(ArrayList<Double[]> train\_x,ArrayList<Double> train\_y)//读入训练数据，标签与训练数据均是以多维数组的格式传入，不必按照 libsvm 标准格式

my\_run() //进行相关训练，并将训练后分类模型存储到 model\_file\_name

(8) svm\_predict.java 为 svm 测试类。

官方接口：

Svm\_predict.main(String argv[])//argv[] 中包含三个字符串分别为 options , model\_file\_name, out\_file\_name

同样由于官方接口都是以文件格式输入输出，不太方便，我们直接使用 svm 中 svm\_predict(model,test\_x)接口。

(9) Resampler.java 与 FilterKit.java 与 SampleBuffers.java 主要负责进行重采样。这部分我是直接重用了 github 上的代码，主要接口与用法 Resampler.java 中都有详细注释，我这里不再列出。

(10) CustomClipLoading.java 与 WaiActivity.java 为自定义等待页面

(11) setKeyActivity.java 与 reSetKeyActivity.java 分别为密保口令设置界面与修改界面，主要是 UI 界面的不同。

(12) sampleShakingActivity.java 负责摇晃数据收集以及摇晃数据重新收集。由于收集采样频率并不稳定，因此我们需要自己计算采样频率，在后面数据处理的地方需要使用。采样 flag 含义：flag=0 表明未开始摇晃，不记录数据；flag=1 表明开始收集数据；flag=2 表示摇晃结束，之后的数据不用收集。alertDialogForSubmit()函数是在一次摇晃结束后弹出的对话框，询问用户是否提交这次摇晃数据。选择放弃的话，重新摇晃。选择确定的话，将摇晃数据交给 Analysis.java 另开线程进行数据处理。如此收集摇晃数据直到达到指定摇晃次数。处理完成后通过 Handler 来讲结果传回，通过 Size.animate(msg.arg1)来修改进度条数据。

(13) Analysis.java 为数据处理类

主要接口：

void init(Context context,Handler msgHandler,double f) //初始化相关静态变量

void receiveData(ArrayList<Tuple[]> sensorData) //接收传递过来的摇晃数据，并另开线程进行处理。

void save(String input,String file)

void writeSDcard(String str,String txt)

void myStore(String train\_path,ArrayList<Double[]> train\_x\_store)//第一种为内存存储方法，后两种为 SD 卡存储方法，其中第三种更为高效（比第二种存储速度提高了接近 10 倍，比第一种速度提高了接近 5 倍）

void returnResult(int result,int arg) // 通过安卓异步处理机制，将结果传回给 sampleShakingActivity.java，以便于通过 Size.animate(msg.arg1)来修改进度条数据。

void dataProcessing(ArrayList<Tuple[]> list,double f) //数据处理

ArrayList<Double> MovingAverageFilter(ArrayList<Double>x,int windowSize) //滑动平均法来对数据进行平滑处理。

ArrayList<Integer> findZreo(ArrayList<Double> w) //找出过度点，返回过度点 index 数组

ArrayList<Double> Integral(ArrayList<Double> x,double t) //对加速度积分求速度，其中  $t=1/f$ 。

ArrayList<Double> CorrectedError(ArrayList<Double> x,ArrayList<Integer> zero) //消除积分产生的累计误差。

void normalization(Double[] v) 数据归一化处理

(14) Match.java 为解锁页面摇晃数据测试类，接口与 analysis 基本相似，不过在摇晃数据写入的时候是以最佳的方式写入的。

(15) LockService.java 为手机锁屏服务类。主要任务是监听手机屏幕点亮关闭广播，屏蔽手机系统锁屏界面，加载自己的锁屏界面。

其中：

addView1(), removeView1(), addView2(), removeView2(), addView3(), removeView3(), 分别是锁屏界面，锁屏界面对话框 1，锁屏界面对话框 2 的加载与取消页面。

(16) networkChangeReceiver.java 与 bootCompleteReceiver.java 分别监听网络变化广播与开机启动广播。同时启动锁屏服务或进行摇晃数据的再次训练(用户开启了锁屏服务的话)。

**仍存在的问题以及有待改进的地方：**

(1) 通过接收开机广播来实现开机启动的时候，因为从开机到接收到广播再到启动应用需要一定的时间（约 5s），在这段时间中，锁屏系统是完全起不了保护作用的。

(2) 不同型号的手机的解锁程序对安卓原程序都做了不同程度的修改的，由于我们是拿 Nexus 来做测试，基本可以实现取消手机自带锁屏系统。不过对于华为，小米之类的手机

效果各不相同。

- (3) 训练一类分类器，没有进行参数网格搜索优化。
- (4) 摇晃数据的采集数，以及一类分类正负样本比例设置参数，暂不确定最优解。
- (5) 摇晃数据的再训练放到了监听网络变化广播的函数里面，不是很合适。
- (6) 暂时对于用户通过清理软件将锁屏服务杀死的情况没有很好的解决办法。