

郑州大学毕业设计（论文）

题 目： 基于 VR 和传感器的青光眼视野检测系统

指导教师： 王超 职称： 讲师

学生姓名： 李成玉 学号： 20177720521

专 业： 软件工程

院（系）： 软件学院

完成时间： 2021 年 6 月 2 日

2021 年 6 月 2 日

摘要

青光眼的危害极大，如果不及时处理，可能造成不可逆转的视野损失，视野检测在青光眼的诊断中不可或缺，但检测设备难以获取，为此开发了一款便携式的快速检测系统。只需要将本系统安装到 Android 设备上，配带 VR 眼镜，通过姿态传感器与检测程序进行交互，就能实现用户自主进行视野范围的检测，并将结果处理成图片分享给他人，还能将检测结果存放于数据库中，方便随时进行查询。本程序分别以 24-2，10-2 作为青光眼前后期的检测范围，对 SITA 快速程序策略加以实现和改进，将其移植到 Android 设备上。快速检测程序的开发使得够患者即使在不去医院的情况下也能进行视野检查，便能够及早发现并及时治疗，挽救患者视力，极大的提高了青光眼检测的高效性、便捷性。

关键词：青光眼；图像处理；VR；传感器

Abstract

Glaucoma is very harmful. If it is not handled in time, it may cause irreversible loss of visual field. Visual field detection is indispensable in the diagnosis of glaucoma, but the detection equipment is difficult to obtain. Therefore, a portable rapid detection system is developed. The system only needs to be installed on the Android device, equipped with VR glasses, and interacted with the detection program through the attitude sensor, which can realize the user's independent detection of the field of vision, and process the results into pictures to share with others, and also store the detection results in the database, which is convenient for query at any time. In this program, 24-2 and 10-2 are used as the detection range of glaucoma. SITA fast program strategy is implemented and improved, and transplanted to Android devices. The development of rapid detection program enables patients to have visual field examination even if they do not go to the hospital, so that they can find and treat as soon as possible, save the vision of patients, and greatly improve the efficiency and convenience of glaucoma detection.

Keywords: glaucoma; image processing; VR; sensors;

目录

摘要.....	I
Abstract.....	II
1 绪论.....	1
1.1 系统概述.....	1
1.1.1 研究背景.....	1
1.1.2 研究意义.....	1
1.1.3 研究现状.....	1
1.1.4 研究内容.....	1
1.2 系统可行性研究.....	2
1.2.1 技术可行性.....	2
1.2.2 经济可行性.....	2
2 系统总体描述.....	3
2.1 关键技术.....	3
2.1.1 开发语言.....	3
2.1.2 开源库.....	3
2.1.3 加速度传感器与地磁传感器.....	3
2.1.4 反畸变.....	3
2.1.5 FastAPI 和 SQLite.....	4
2.2 系统简介.....	4
2.3 功能性需求.....	4
2.4 非功能性需求.....	4
2.5 设计约束.....	4
2.6 本章小结.....	5
3 系统设计.....	6
3.1 总体设计.....	6
3.2 接口设计.....	8
3.3 数据库设计.....	8
3.4 本章小结.....	9
4 系统实现与测试.....	10
4.1 视图模块.....	10
4.1.1 功能描述.....	10
4.1.2 关键流程.....	10
4.2 业务逻辑模块.....	14
4.2.1 功能描述.....	14
4.2.2 关键流程.....	14
4.3 数据模块.....	19
4.3.1 功能描述.....	19
4.3.2 关键流程.....	19
4.4 主要问题及解决方法.....	19
4.5 测试与验证.....	19
4.5.1 测试原则.....	19

4.5.2 测试目的.....	20
4.5.3 功能测试.....	20
4.6 本章小结.....	20
5 总结.....	21
参考文献.....	22
致谢.....	23

1 绪论

1.1 系统概述

1.1.1 研究背景

青光眼的危害极大，据估计全世界该患病者超过六千五百万人，而且还在持续增加，很多青光眼患者在患病初期并没有明显症状，甚至其中一半的人对他们所患的疾病毫无意识，有些人可能只是感觉到眼睛发胀甚至头痛，然而青光眼一直发展下去，很容易造成不可逆转的视力下降，严重的会引起失明，因此青光眼的早期诊断成为保存视力非常重要的一环。

视野检测是现如今诊断和处理青光眼的基本手段，定期进行视野检测有助于发现和追踪病情的进展，以便医生定出最好的治疗方案，视盘图像分析或者视神经纤维缺损分析虽然也非常重要，但他们都不能代替视野检测在青光眼诊断和追踪中的作用^[1]。但是由于现今的检测机器价格普遍比较昂贵，并非普通民众所能承担的起的，故只能在有条件的医院才能进行视野检测。

1.1.2 研究意义

基于以上两点，虽然青光眼危害性大，但如果能够开发一款便携式的快速检测设备，能够使患者即使在不去医院的情况下也能进行视野检查，便能够及早发现并及时治疗，挽救患者视力。于是开发了一款使用 Android 手机与 VR 眼镜结合检测的快速视野检测系统。一方面 VR 眼镜排除了外界因素的干扰，另一方面将检测程序嵌入到手机当中也方便携带，且无需增加额外的花费，即使在没有条件的偏远地区也能进行视野检测^[2]。

1.1.3 研究现状

随着 VR 技术的发展，相关硬件再性能上已经能够更快更精确计算更复杂的模型。VR 设备的成本也大大降低，20 世纪 90 年代的 V8 头盔价格约为 10 万元，20 世纪初期的 xSight 工业级头盔价格 40 余万^[3]，如今的 VR 眼镜，价格在几百到上千元不等，小米的智能 VR 一体机价格在 2000 元左右。相关软件和技术也已经走向成熟稳定，且广泛应用于游戏、影视等领域^[4-5]。

随着计算机自动视野计算的不断发展，视野检测程序和策略也在不断更新。从最早的全阈值策略到瑞典交互式阈值算法 (SITA) 策略，不断为临床诊疗提供更为快捷和有效的检测方法^[6-7]。SITA 标准程序可用 Humphrey 全阈值策略的一半时间来完成一次 30-2 检测，而不会出现重复性或视野丢失敏感度下降。SITA 快速程序只用 FastPac 一半的时间，却能提供相同的结果^[8]。

1.1.4 研究内容

本文基于青光眼的检测问题，在 VR 技术和计算机自动视野计算发展的推动下^[9-10]，提出了一种基于 VR 技术的青光眼视野检测系统。本文的主要研究内容如下：

- (1) 研究青光眼检测的流程，实现青光眼检测算法。
- (2) 研究检测过程中传感器回收的时机和有效响应时间的判定。
- (3) 研究检测过程中图像位置和矩阵越界问题。
- (4) 研究青光眼检测结果灰度图和概率偏差图的绘制和存储问题。
- (5) 研究系统的工作流程和 Android 开发。

1.2 系统可行性研究

1.2.1 技术可行性

Google VR SDK 有着良好的解决方案，我们只需要将其引入无需主动调用，便可实现 VR 效果，解决了 VR 框架搭建的问题。

Unity 有着直观的用户可视界面和工具，能够快速启动工作原型，并规避在低级编程上花费的时间，比如画图，使得模型的建立更为简单，我们仅需要处理逻辑即可。

FastAPI 以其轻量的特性，可以快速搭建一个服务器，我们需要做的就是接收与处理请求，无需把大量的时间花费在搭建服务器上。

SQLite 也以其轻量的特性，可以快速建立数据库连接，我们只需实现我们所要的增删改查操作。

对于开发所用的 Java, Python, C#, Shell, 都是已经相对成熟的语言，且熟于使用，可为开发提供有力的支撑。

1.2.2 经济可行性

系统开发所使用的工具 IDEA、Unity、SQLite、JDK 和开发框架 FastAPI 均为免费版且易于下载获取，第三方库 Google VR SDK、Android API、canvas 和所涉及的开发语言 Java, Python, C#, Shell 都是开源免费的，且系统部署在手机上，无需购买专用设备，且减轻了工作量，节省了大量开发成本。因此，系统的开发在经济方面是可行的。

2 系统总体描述

本章讲述了系统实现所使用到的相关技术，为系统的实现提供了有效保障，紧接着结合需求分析对系统进行了简单介绍，最后对开发系统的相关环境进行限制，最大程度的减小了因环境问题造成系统的异常情况。

2.1 关键技术

系统的不同模块使用了不同的技术，客户端使用了 Java，JDK 版本为 1.8,由于直接使用真机调试，Android API 版本为 30，使用了 canvas 进行结果的复现，使用了 base64 对传输内容进行了加密，使用了 加速度传感器与地磁传感器 监听人体的点头，检测算法使用了 C#，版本 8.8.3，使用了反畸变处理图像,服务端使用了 Python，版本为 Python3.9，使用了 FastAPI 框架搭建服务器，SQLite 作为数据库。

2.1.1 开发语言

Java: 面向对象设计语言; C#: 面向对象设计语言; Python: 面向对象设计语言

2.1.2 开源库

- (1) Android API: 开源，免费的代码库，由于其开放性的原因，很多厂商对 Android 加以改造，出现了三星，小米等独具特色的 Android 系统。
- (2) canvas: canvas 类暴露了很多绘图方法，比如 drawText 可以绘制文字，drawPoint 可以绘制点等等，canvas 绘图有三个基本要素：画布，坐标，画笔，我们通过 canvas 的各种绘制方法将图形绘制到画布上，传入的坐标决定了我们绘制的形状，而画笔决定了绘制的样式，比如颜色，线条宽度等。
- (3) base64: base64 其实是一个编码再解码的过程，是将每三个字节转化成二进制然后再转为四个字符。

2.1.3 加速度传感器与地磁传感器

角度变化使用方向传感器即可满足要求，但由于方向传感器在高版本的 Android API 中已经过时，故使用加速度传感器与地磁传感器作为替代的解决方案，要获取坐标中 x,y,z 的值，坐标系如图 2-1 所示，就要使用 getOrientation，而 getOrientation 需要获取一个旋转矩阵，这时就要用到加速度传感器与地磁传感器提供数据源。

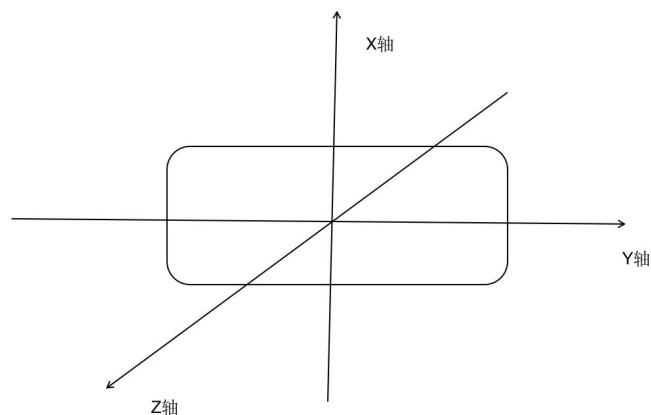


图 2-1 设备坐标系

2.1.4 反畸变

由于 VR 镜头中心与边缘处放大系数不同的原因，会使图像产生畸变，这时我们就需要对图片做反畸变，达到抵消畸变的目的，Carboard 中将图形畸变封装成了两个类，一个是 Distortion，用于计算畸变因子和做畸变与反畸变，一个是 DistortionRenderer，用于做图形的渲染。

2.1.5 FastAPI 和 SQLite

- (1) FastAPI: 一个用于快速构建服务端的 web 框架。
- (2) SQLite: 一个无需配置的轻量数据库, 支持 SQL92 标准的大多数查询功能, 适配 UNIX, Windows 常用平台。

2.2 系统简介

本系统通过与用户进行交互，不断计算用户眼睛各处的视野状况，完成视野范围的检测，并将检测结果转换为图像显示在界面上，同时还能将检测结果存放于远端数据库之中，方便跨设备查看，也可生成报告分享给他人。

2.3 功能性需求

本系统的功能大致分为以下几块：

- (1) 视野检测：通过选择对应的检测程序以及眼睛进行视野检测。
- (2) 登陆：输入账号密码进行登陆。
- (3) 注册：创建一个账号供保存个人信息
- (4) 匿名检测：不输入账号密码匿名检测。
- (5) 修改密码：修改登陆账号所使用的密码。
- (6) 查看结果：查看检测的结果。
- (7) 分享：将检测的结果分享给他人。
- (8) 数据可视化：将检测的结果以图像的方式展示在界面上。
- (9) 数据同步：保持客户端与服务端数据的一致性。
- (10) 生成检测报告：将检测结果合并。
- (11) 控制器：控制检测的正常进行。
- (12) 数据库操作：对数据库进行相应的操作，比如创建表以及增删改查。

2.4 非功能性需求

为了能给用户更好的体验，一些非功能性需求也是必要的：

- (1) 性能需求：页面跳转时间<3s, 由于需要对数据库进行全表查询求得检测着的检测结果在所有人中所占的比例，故随着数据库中表的增大，搜索返回结果的时间也会越来越长。
- (2) 安全性：传输数据皆使用 base64 加密，用户只有权修改数据库中属于自己的数据，如用户密码等，无权修改其他人的数据。
- (3) 可维护性：每个功能都单独拆分成模块，且使用日志，遵从代码开发规范，易于拓展和维护。
- (4) 易用性：app 中给予和许多提示性的文字用于帮助用户尽快适应。

2.5 设计约束

- (1) 需要运行在 Android 4.4 ‘KitKat’ 或者更高版本的安卓设备上^[1]。
- (2) 需要 LTS release 2018.4 或者更高版本的 Unity。

- (3) 需要在 Unity 上安装支持安卓构建的组件(Android Bulid Support(*)).
- (4) 需要在 Unity 中的用户配置 (Player Setting) 下的 XR Setting 中设置支持 VR, 即 Virtual Reality Support=true.
- (5) 需要 JDK>1.8.
- (6) 需要 Python>3.

2.6 本章小结

本章阐述了系统开发所用的关键技术, 对系统进行了简单介绍。接着介绍了系统的功能性需求, 从性能、安全性、可维护性和易用性四个方面描述了系统的非功能性需求, 最后介绍了系统开发所需的约束条件。

3 系统设计

本章主要讲述系统的总体设计，从模块、流程、用例三个方面讲述了系统是如何进行运作的，为系统的实现建立了坚实的基础，同时规定了系统中所使用到的接口，是系统的开发更为简洁，并对使用到的数据库进行了详细设计，降低了数据管理的难度和成本。

3.1 总体设计

3.1.1 模块设计

该检测程序通过不断在设备屏幕的不同位置上显示视标，受检测者则使用控制器对是否可见视标做出响应，从而得出系列检测结果，并将结果转换成图像以展示给用户，从功能上划分，主要分为三大部分，如图 3-1，用于展示界面的视图模块，用于图像处理，数据处理，以及与服务其之间通信的业务逻辑模块，用于与客户端通信以及操作数据库进行增删改查的数据模块。

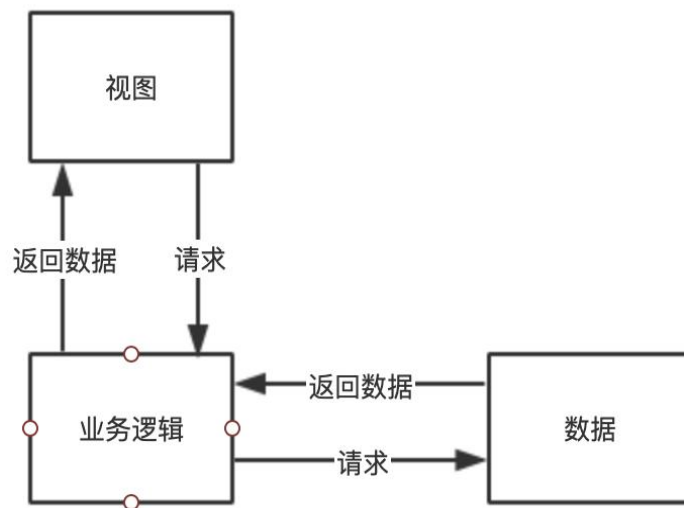


图 3-1 系统模块

每次展示结果之前，视图层都会向业务逻辑层请求更新数据，以确保数据结果是最新的。此时业务逻辑层会向服务器提出查询请求，如果可以从服务端拿到数据，则视图层会根据这个结果重新生成结果图，如果没有拿到数据，则视图层会展示无数据的提示。

每当完成一次检测时，如果是使用 id 密码进行登陆，则业务逻辑层会向服务器提出插入请求，服务端会首先判断用户存不存在，如果存在，则进行刷新操作，如果不存在，则进行插入操作，如果是匿名检测，则结果只会保存到程序退出之前。

当进行登陆操作时，业务逻辑层会向服务器发送查询请求进行验证，此时服务器会对发送过来的用户 id 和密码进行匹配，客户端根据服务器端的返回值来控制是否允许用户进行登陆。

当进行注册时，业务逻辑层会向服务器发送注册请求进行注册，此时服务器端会查询数据库中有没有相同的 id，如果存在,则返回 false，提示用户已存在，如果不存在，则返回 true，跳转页面。

3.1.2 流程设计

流程如图 3-2，打开 app 后可以选择是否登陆，无论是否登陆都会跳转到主界面，登陆与否的区别只是检测结果的有效时长，进入到主界面后可以查看结果，也可进行视野范围检测，检测完成后可再次进行检测或者查看结果，查看结果的同时页能将结果分享给他人。

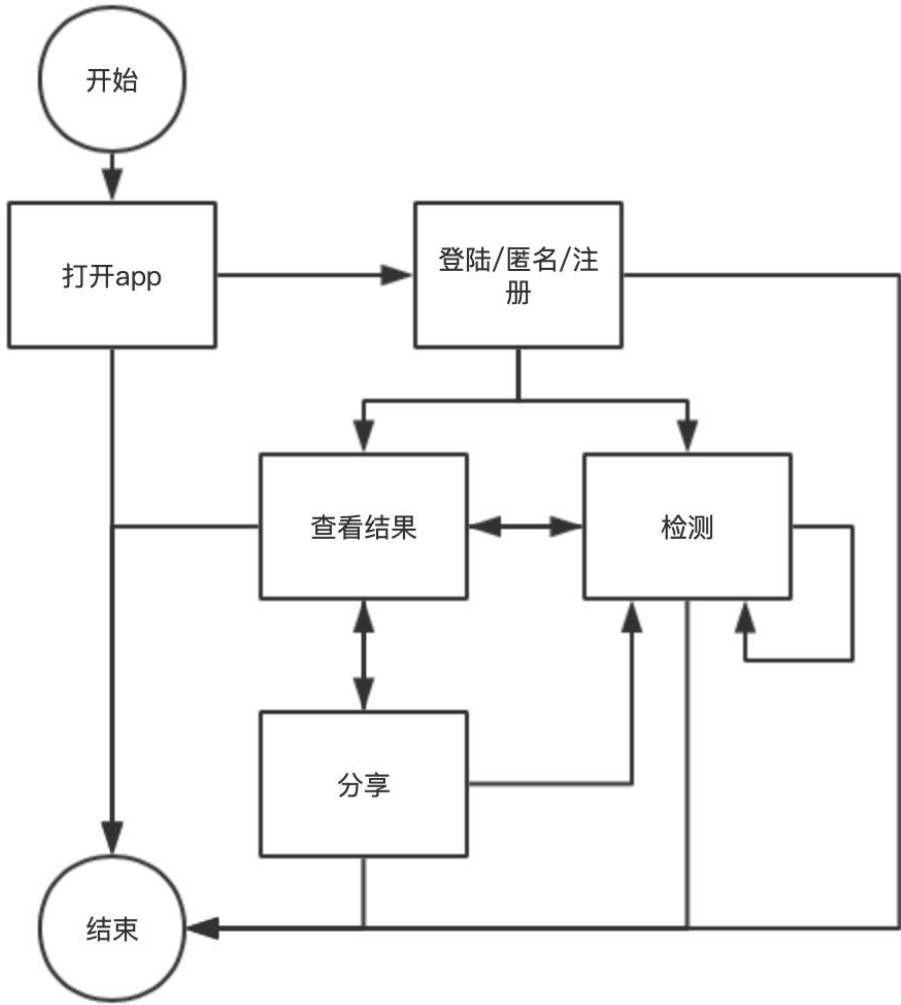


图 3-2 使用流程

3.1.3 用例设计

由于开发范围只设计客户端，故只对用户进行设计，用户可以进行视野检测，在检测的过程中，如果出现不适症状，还可进行暂停，待症状缓解后再行手动恢复，可查看检测结果，并将检测结果分享给他人，可进行登陆，登陆后可修改密码，也可进行注册，用例图如 3-3 所示。

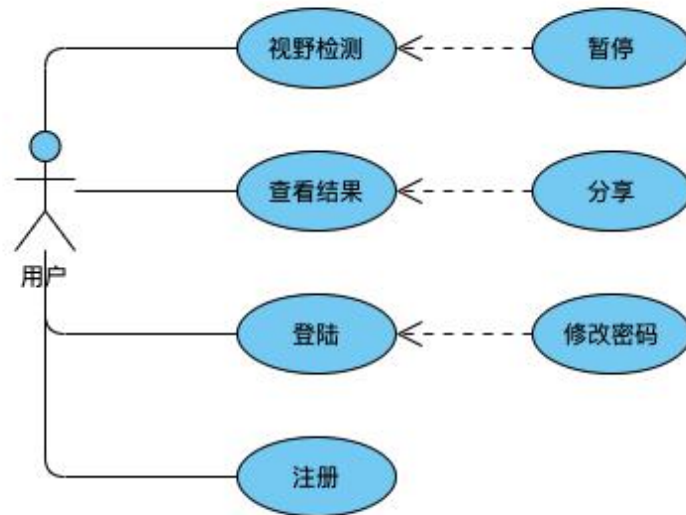


图 3-3 用例图

3.2 接口设计

- (1) onKeyDown: 用于处理控制器与检测程序之间的通信。
- (2) onClick: 用于处理各种点击事件。
- (3) onCreate: 用于初始化 view。
- (4) ifExistAir: 用于检测控制器是否连接。
- (5) DrawGrayImage: 用于控制生成灰度图。
- (6) DrawRatioImage: 用于生成概率偏差图。
- (7) HttpService: 用于向服务端发送各种请求，并接收返回结果，比如：注册、校验、查询等。
- (8) onPress: 用于分享生成的报告。
- (9) onActivityCreated: 用于初始化 view。
- (10) onResume: 用户刷新 view。
- (11) Start: 用于初始化数据。
- (12) Update: 用于更新数据。
- (13) SightingPostLocationDeal: 用于加载检测场景响应交互。
- (14) ThresholdCalculate: 用于控制检测程序进行结果的计算。
- (15) Service: 用于接收客户端的各种请求并加以响应，比如注册，校验，查询等。
- (16) TableTools: 用于操作数据库中表，比如增删改查。

3.3 数据库设计

本检测系统采用了 SQLite 数据库，SQLite 不依赖第三方软件，无需安装，他所有的数据库信息都存储在一个文件内，SQLite 无需配置复杂的连接驱动，连接数据库只需要 `sqlite3.connect("数据库地址")`，就像是打开了一个文件。SQLite 支持常规 sql 语句，只需要 `sqlite3.connect("数据库地址").execute("sql 语句")`，就能实现对表的增删改查。

为了保持数据的持久性，使用户登陆之后就能查询到自己之前最近一次的检测记录，故使用了 `early_date`，`late_date` 两张表来分别保存用户选择不同检测程序后的检测数据。

为了保证数据的唯一性，故又采用 `id` 与 `eye` 作为联合主键的方式来保证每个人的每只

眼睛都是唯一存在的, early_date 中的字段包括 id、eye、sightingLoseRatio (固视丢失率)、falseNegativeRatio(假阴性率)、falsePositiveRatio (假阳性率), location1-72(72 个位点的视野阈值情况), late_date 中的只是将描述视野阈值情况的位点增加到了 100 个, 其余字段与 early_date 相同。

为了数据的准确性, 确保每个人获得的测试数据都是自己的, 故使用了一个 user 表来进行 id 与 passWord 的校验, id 作为唯一主键, user 中的字段包括 id,passWord。每个 id 在 user 表中对应一条数据, 在两张 data 表中分别对应两条数据。

3.4 本章小结

本章阐述了从功能上阐述了系统的主要构成部分, 以及个模块之间是如何进行通信的, 实现了模块与模块之间数据流的打通, 并对系统中主要用到的接口和采用数据库的目的进行了阐述。

4 系统实现与测试

本章通过讲述模块工作流程的方式对系统各个模块具体实现进行详细阐述，并配以图标加以说明，之后讲述了开发中所遇到的较为突出的问题以及解决方案，增强了系统实现的可信度。

4.1 视图模块

4.1.1 功能描述

该模块主要用于视图展示，将业务逻辑层处理好的结果显示在手机屏幕上，在用户使用的过程中，给予相关提示性文字描述，辅助用户进行测试，以及展示检测过程中所用到的相关组件。

4.1.2 关键流程

打开检测程序首先会跳转到登陆界面。再登陆界面中首先会跳出弹窗，如图 4-1，弹窗的实现使用了 Android 提供的 AlertDialog，这里并没有直接使用 Android 自带的样式，其实 setMessage 同样可以简单设置文字描述，但是如果需要更复杂的样式，比如设置字体的颜色 setMessage 就解决不了，而且 AlertDialog 也没有提供其他设置字体的方法，这时我们就需要是自定义一个 view，这个 view 就是在 XML 文件中设置的样式，对于 XML 文件而言，我们可以根据需求设置各种样式，然后使用 setContentView 去加载这个 view，这样就可以对弹窗进行定制开发，就能显示出我们想要的效果。



图 4-1 提示弹窗

点击屏幕任意位置后会开始进行新手引导，在登陆界面的新手引导中包含了对此页面上主要组件的解释，如图 4-2，引导层的实现使用了开源框架 GuideView，与弹窗的用法相似，同样是自定义一个 view，不同的是，这里可以定义多个不同的 view，然后指定他们显示的顺序，这样就能在你点击遮罩层后跳出下一个遮罩层，达到新手指引的目的。引导完成后显示登陆页面，如图 4-3。

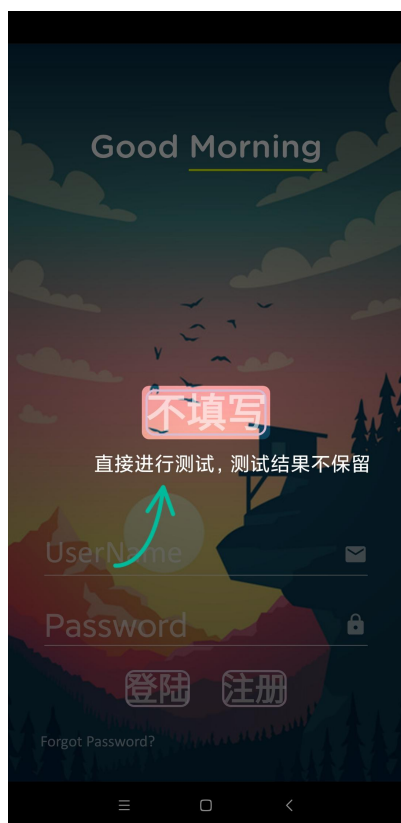


图 4-2 新手引导

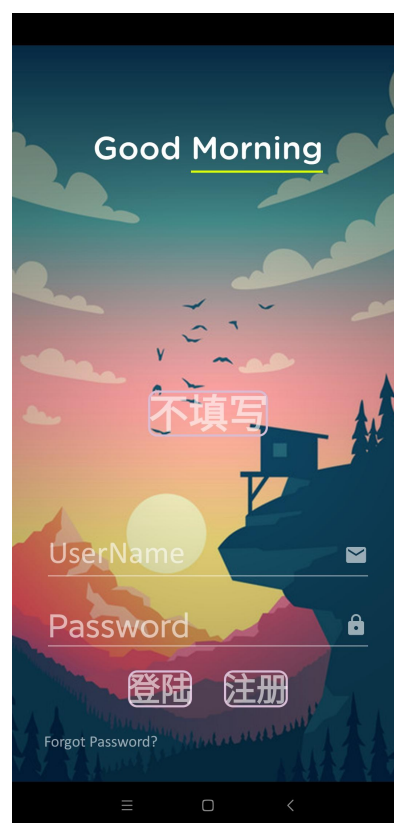


图 4-3 登陆界面

当在编辑框中填写好用户名和密码点击登陆按钮时，会对编辑框中输入的 id 进行校验，如果校验不通过，则会显示相应提示，如图 4-4，id 校验的实现使用了正则表达式进行匹配，只有当输入的 id 属于身份证号、电话号、邮箱中的一种时，才能匹配通过。



图 4-4 警告弹窗

通过校验后，会跳转到主界面。同样在进行完新手引导后显示主页面，当选中主界面

底部的导航栏时, 会根据选中的 tab 进入到不同的 Fragment 中, 当点击导航栏中的主页时, 会跳转到选择检测程序界面, 屏幕中会显示检测程序选择页面, 如图 4-5。

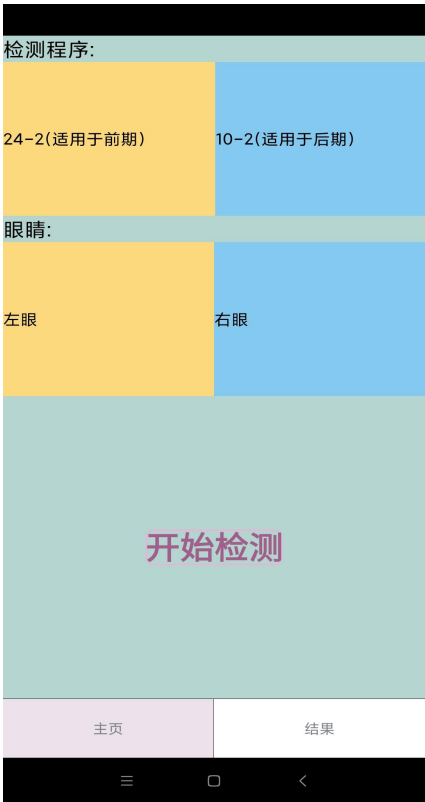


图 4-5 选择程序界面

此时可在本页面上选择相应的检测程序与检测眼睛, 点击开始检测按钮后, 会提示佩戴 VR 眼镜开始进行检测, 如图 4-6。、

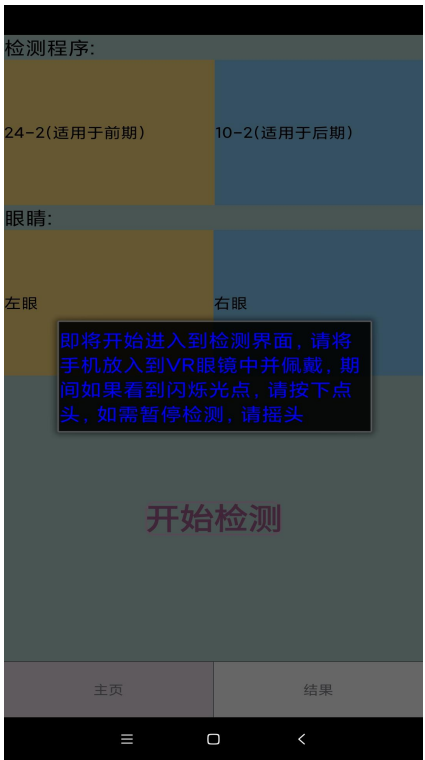


图 4-6 提示弹窗

为了使用户能够有足够的时间佩戴好设备, 这里使线程休眠了 10 秒, 10 秒钟后, 会

跳转检测界面并开启 VR 模式,此时屏幕上会显示检测界面。当用户确认开始检测后,会加载组件开始检测。在此期间屏幕上会随机出现不同亮度的视标,检测者只需在视标出现时完成对其响应,便可得到受检者的视野状况,检测完成后会重新返回到主界面,如图 4-7。

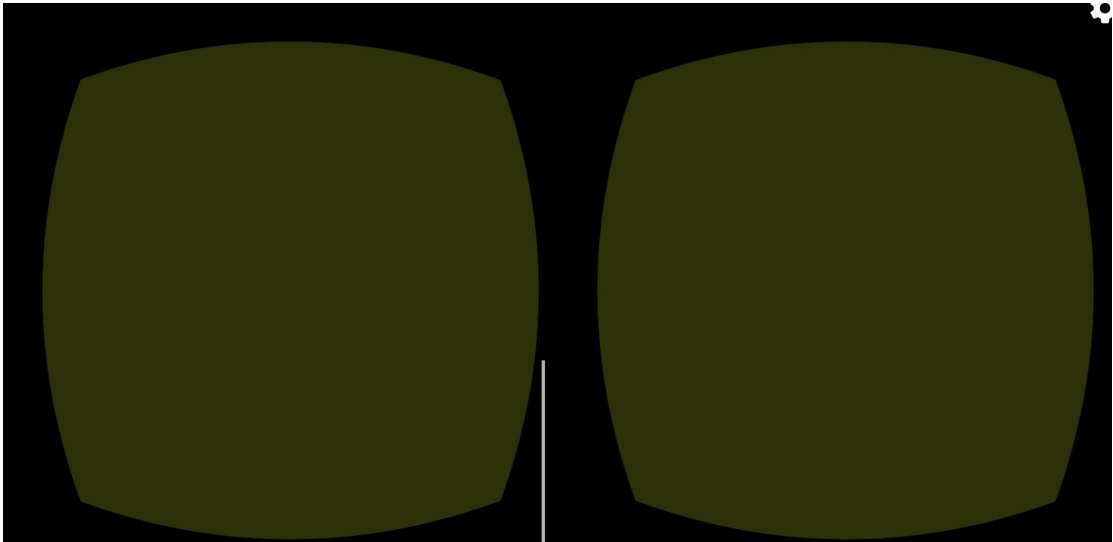


图 4-7 检测界面

当点击导航栏中的结果时,会跳转到结果界面,此时屏幕上会显示检测结果,如图 4-8,每次重新进入到结果界面中时,都会调用 onResume,用于保持结果最新,Fragment 的切换使用了 FragmentTransaction,当需要从一个 Fragment 切换到另一个 Fragment 时,调用 FragmentTransaction 中的 hide 将之隐藏,同时调用 show 将要跳转到的 Fragment 显示,如果需要显示的 Fragment 不存在,则调用 add 进行显示。

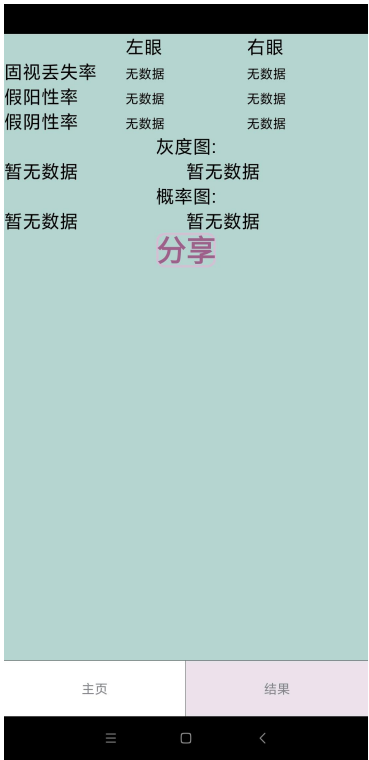


图 4-8 结果界面



图 4-9 分享界面

当点击结果界面中的分享按钮时,会根据选择的 app (如图 4-9) 跳转到其当前的 Activity 中,并将结果报告发送出去,分享的实现使用了 Android 中 Intent,只需要隐式指定其 Action 为 ACTION_SEND,因为我们并不清楚他要发送到哪个 Activity,然后设置需要发送的内

容和类型，之后当点击分享时，系统就会打开分享选择器。

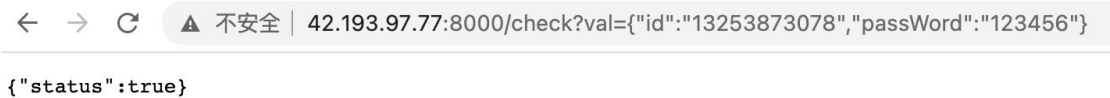
4.2 业务逻辑模块

4.2.1 功能描述

该模块主要用与响应视图模块的请求，以及向数据模块发送并接收数据模块返回的数据，同时还能控制检测的进行，主要包括视标位置，检测进度，检测结果的计算，程序的速度，以及图形的绘制。

4.2.2 关键流程

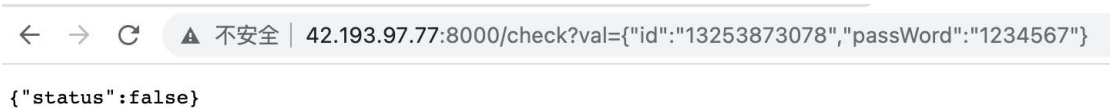
打开检测程序,当填写完用户 id 和密码后点击登陆按钮时，如果 id 合规，则会将用户 id 和密码包装成 json 数据，然后会向服务器发送一条校验请求，同样从服务器返回过来的内容也是经过加密的 json 数据，经过解密后如果得到的值是 true，如图 4-10，则校验通过，跳转到主界面，如果得到的值是 false，如图 4-11，则校验不通过，直接 return，如果 id 不合规，则直接 return。



```
← → ↻ ⚠ 不安全 | 42.193.97.77:8000/check?val={"id":"13253873078","passWord":"123456"}

{"status":true}
```

图 4-10 校验通过

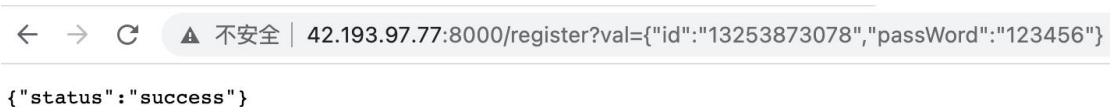


```
← → ↻ ⚠ 不安全 | 42.193.97.77:8000/check?val={"id":"13253873078","passWord":"123456"}

{"status":false}
```

图 4-11 校验不通过

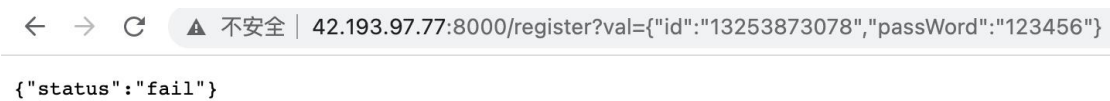
也可先进行注册，此时会将用户 id 和密码包装成 json 数据，向服务器发送一条注册请求，如果服务器返回 success，如图 4-12,则注册成功，跳转到主界面，如果服务器返回 fail，如图 4-13，则注册失败，直接 return。



```
← → ↻ ⚠ 不安全 | 42.193.97.77:8000/register?val={"id":"13253873078","passWord":"123456"}

{"status":"success"}
```

图 4-12 注册成功



```
← → ↻ ⚠ 不安全 | 42.193.97.77:8000/register?val={"id":"13253873078","passWord":"123456"}

{"status":"fail"}
```

图 4-13 注册失败

当每次选择结果界面后，会首先携带用户 id，以及选择的 eye，program 向服务器发送一条查询请求，如果获取到的数据不为空，如图 4-14，则更新全局变量中的值，并加载结果图，如果获取到的数据为空，如图 4-15，则提示无数据,更新的实现依靠这一个全局变量 GlobalVal，所有共享的资源都被存储在这个 GlobalVal 对象中，每当有值需要进行更新时，都会调用这个全局变量中对应的 set 方法去更新 GlobalVal 对象中的值。

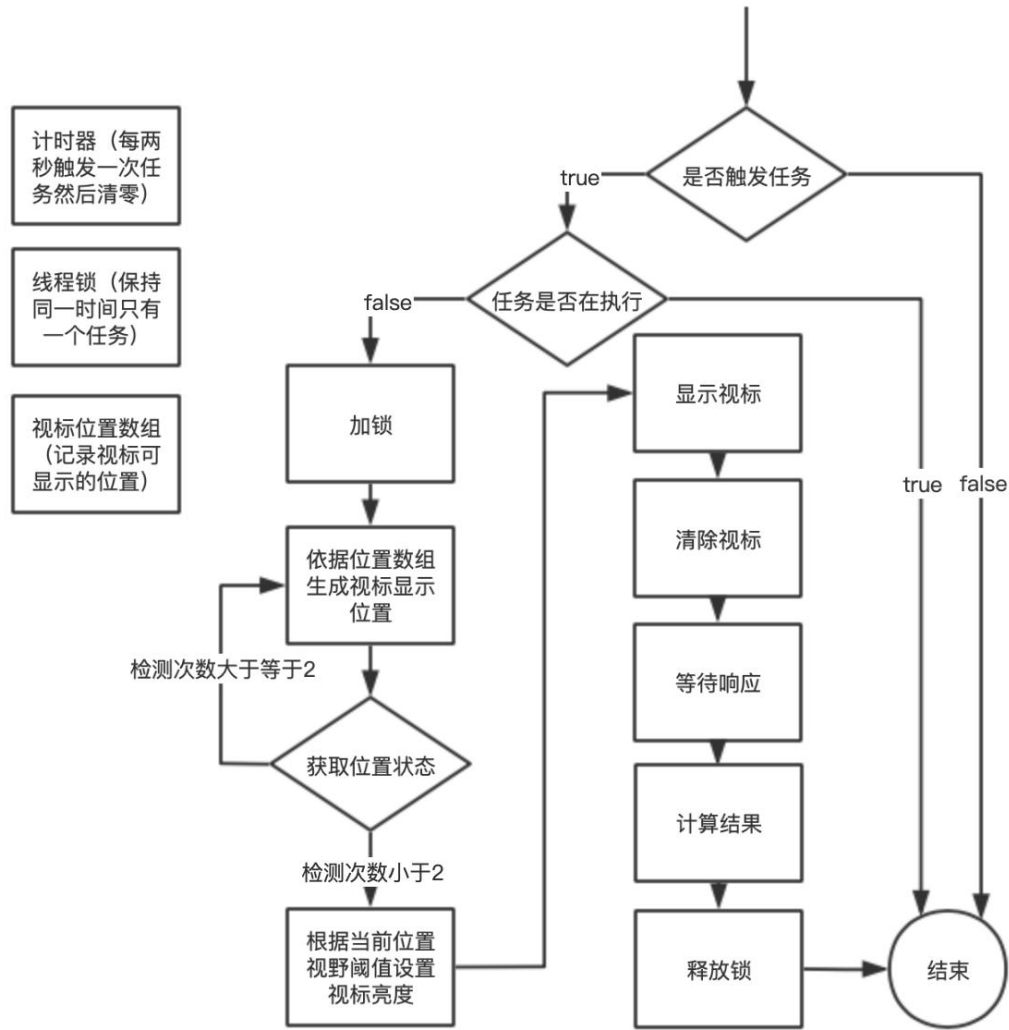


图 4-16 检测流程图

青光眼前后期检测算法的逻辑大体上相同，视标的位置由一个二维数组记录，同时以二维数组的大小建立两个随机数，使检测期间视标会在规定的范围内随机出现，位置的状态也是由一个一维数组记录着，那么为了使记录状态的一维数组与记录位置的二维数组能够对应，他们之前应存在这样一个关系， $b=ax+y$ ，其中 b 代表一维数组中的位置， a 代表二维数组中的一维长度， x,y 代表二维数组中的位置，当同一个位置被检测过两次以上后，则此位置视标不会再次出现，视野阈值同样由一维数组记录，与位置数组的对应关系同上，将阈值数组取出来的值作为分母，从而达到视标的亮度会随着视野阈值的降低而增加的目的。

从视标在某一位置出现开始到在另一位置上出现记为一次检测，在检测的过程中，视标的显示时间只有短短的 0.2 秒，如果仅在这 0.2 秒内同时完成发现，响应这两个动作是极为不合理的，所以在视标消失的空隙时间里，应有一部分时间被视为是有效的响应时间。

在这次检测过程中，如果之前进行了对视标的响应，而本次没有响应，则假阴性次数加一，如果在视标消失的过程中进行了对视标的响应，则假阳性次数加一，如果对非有效位置上显示的视标进行响应，则固视丢失次数加一，如果对有效位置的视标进行了响应，且视标是处于显示状态，则此位置上的检测次数加一，如果没有对有效位置的视标进行响应，且视标处于显示状态，则不仅此位置上的视野阈值会降低，与其相邻的周围八个位置（如果存在）的视野阈值也会降低，流程如图 4-17。

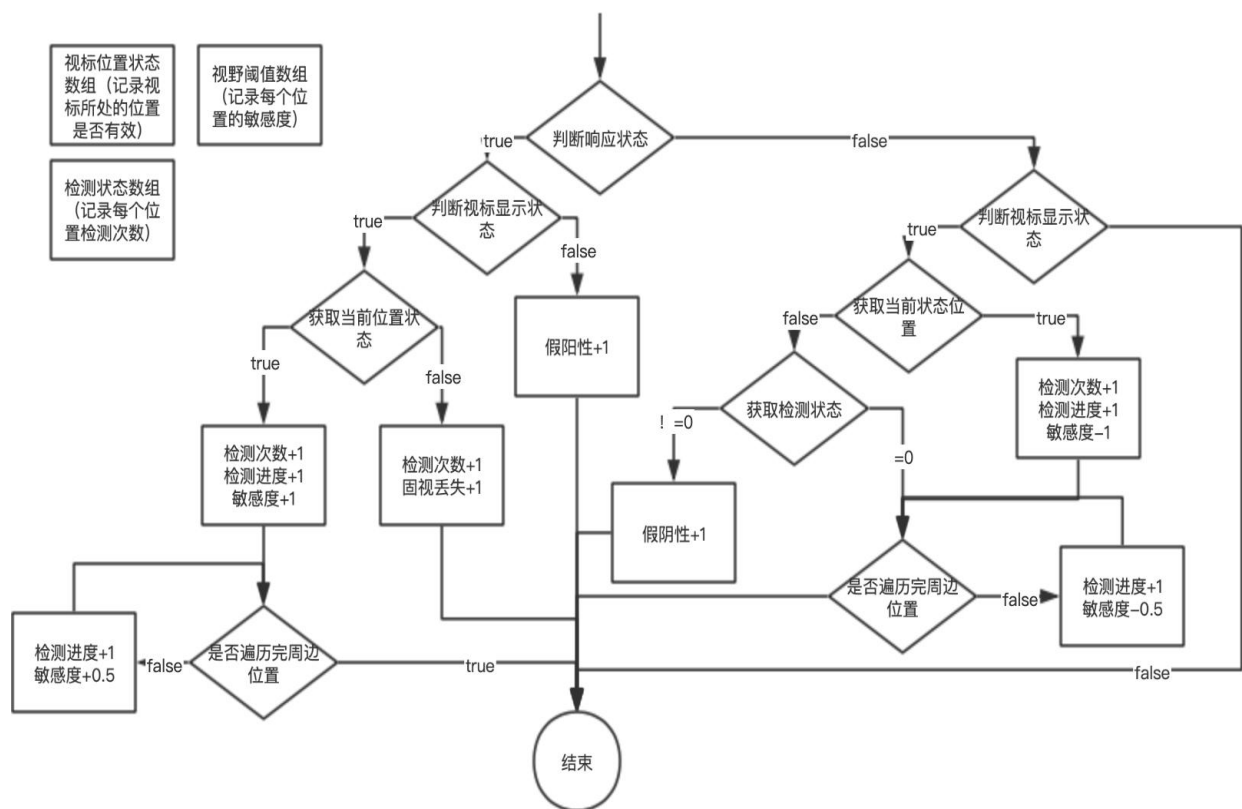


图 4-17 计算流程图

为了加快检测进度，在视标所处位置的视野阈值下降时，与其相邻的位置视野阈值也跟着下降，但如何确定有多少个位置与之相邻成为一个问题，例如中心点相邻的位置有八个，而边缘处相邻的位置小于八个，如果要分情况讨论，那么结果将是巨大的，为了解决这个问题，使用了八次 try-catch 操作，当发生越界时，捕获这个异常不进行任何操作，继续向下执行。不同的是视标的大小，检测位置以及检测的次数，前期的测试范围大致是后期的 3 倍，如图 4-18，4-19。

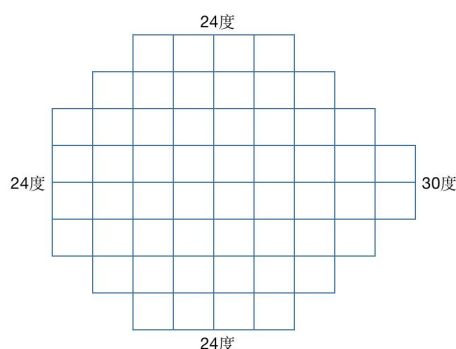


图 4-18 青光眼前期左眼检测范围

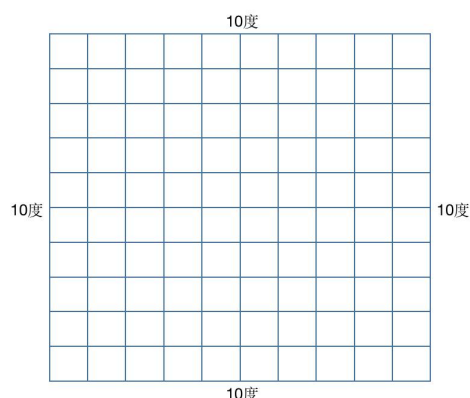


图 4-19 青光眼前期左眼检测范围

图像处理使用了 canvas 进行结果的绘制，灰度图的绘制范围是根据对应检测程序中视野的有效范围来进行绘制的，有效范围也是由一个一维数组进行记录的，如果进行绘制的位置在有效范围内，则使用 drawRect 绘制方块，如果不再，则跳过绘制，移动 canvas 的位置，绘制下一个方块，依据检测结果中阈值的大小为每个方块设置不同的颜色，每当一个方块绘制完成后，由于 canvas 的特性，之前绘制结果不会消失，最终绘制成灰度图，

如图 4-20，4-21。



图 4-20 青光眼前期左眼灰度图

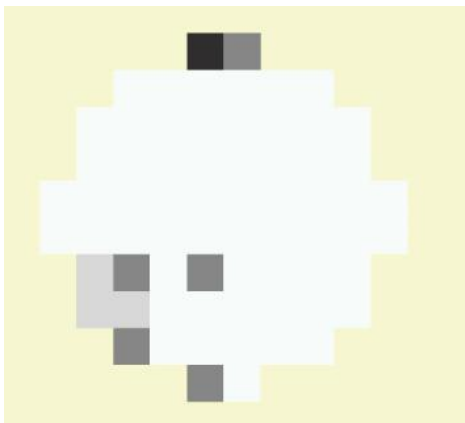


图 4-21 青光眼后期左眼灰度图

概率偏差图是根据自己的检测结果在数据库中的占比计算后得到的结果，绘制的方法大致与灰度图相同，绘制流程如图 4-22。

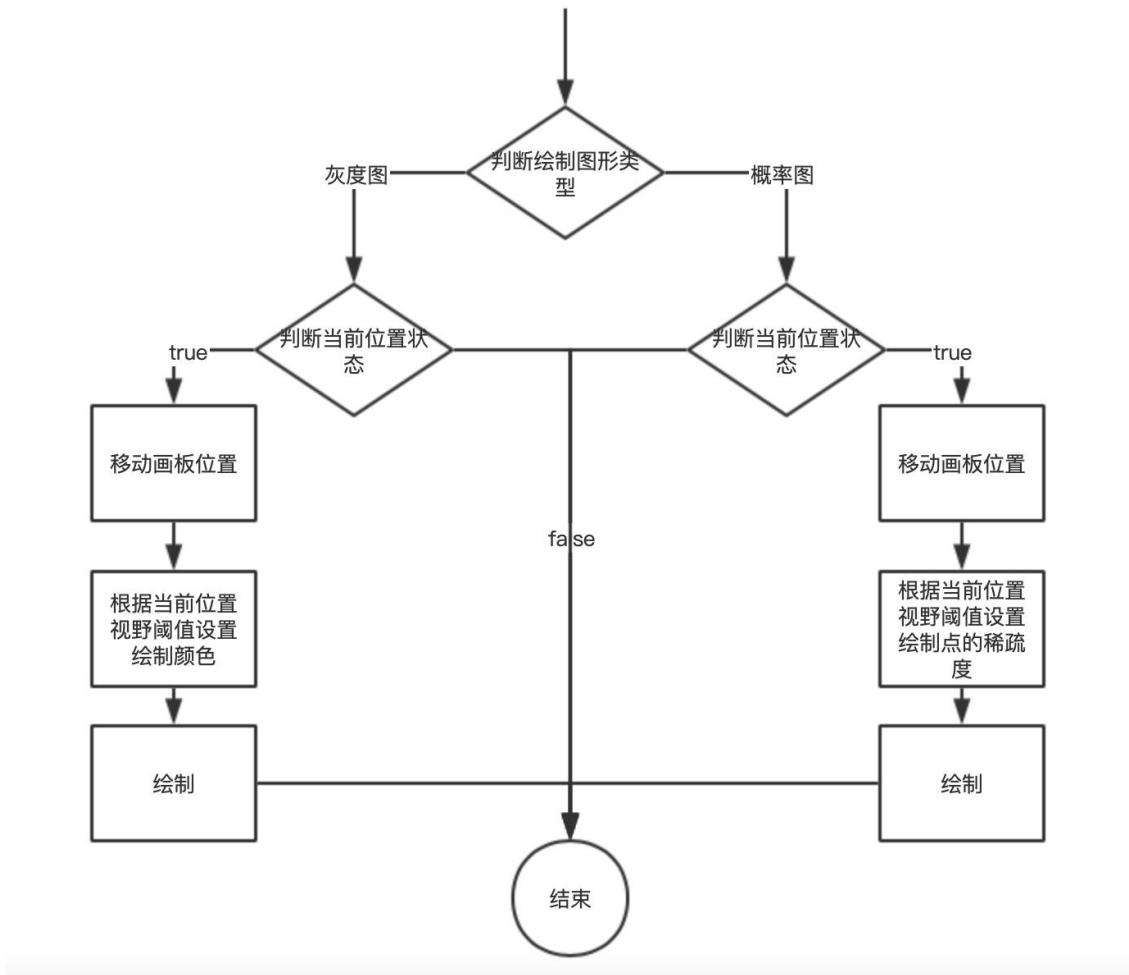


图 4-22 绘制流程图

不过概率偏差图使用的点的密集程度来表示结果值的大小，关于点状方块的绘制，其实是使用了 `drawPoints`，`drawPoints` 中需要传入一个一维数组，数组中每两个值代表一个位置，这时我们只需确定需要绘制点的位置，然后使用一个双层循环创建一个一维数组，每一次内循环，即代表写入了一列点，只需要控制循环中的步长，便可控制绘制点的密集程度，然后在重复灰度图的绘制操作，最终绘制成概率偏差图，如图 4-23，4-24。

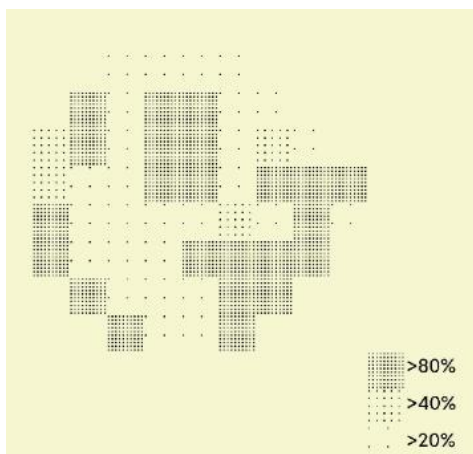


图 4-23 青光眼前期左眼概率偏差图

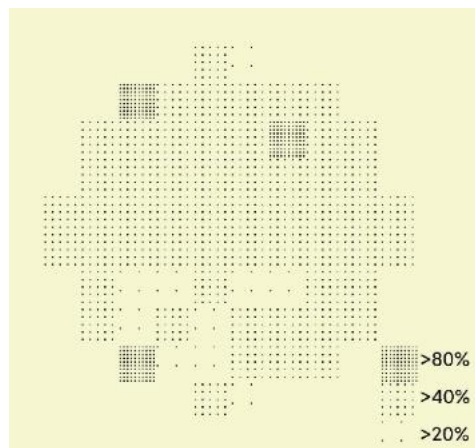


图 4-24 青光眼后期左眼概率偏差图

4.3 数据模块

4.3.1 功能描述

该模块主要用于处理客户端发送过来的请求，根据不同的请求生成并执行对应的 sql，根据操作返回结果和请求类型判断是否进行下一步处理，处理完成后会将结果包装成 json 并返回。

4.3.2 关键流程

当接收到客户端发送来的插入请求时，首先调用 TableTools 中的插入方法向对应 data 表中插入数据，如果插入失败则进行更新操作调用 TableTools 中的更新方法，插入成功则返回 True 否则返回 false。

当接收到客户端发送来的校验请求时，首先调用 TableTools 中的校验方法，向 user 表中查询请求过来的值与表中的值是否一致，如果 id 不存在，则返回 false，如果 id 存在，passWord 不一致，则返回 false，如果一致，则返回 true。

当接受到客户端发送来的注册请求时，首先调用 TableTools 中的注册方法向 user 表中插入数据，插入失败则会返回 fail，插入成功则返会 success。

当接收到客户端发送过来的查询请求时，首先调用 TableTools 中的查询方法向对应 data 表中查询数据，然后会以查询的结果中的视野阈值为依据，调用 TableTools 中的计算比例方法，再次向对应 data 表中查询每个值占总体的比例，同时根据相似度算法计算个体与总体的相似度，之后将结果返回。

4.4 主要问题及解决方法

由于 Unity 在 VR 模式下会主动开陀螺仪，故眼前的图形不会是固定不动的，这将对检测产生很大的影响，为了解决这个问题，将 Unity 所有的组件都绑定到了 Camera 上，这样，不管检测者怎样移动，图像始终是在眼前，但是随着测试者移动，组件的坐标确是在不断发生变化的，假如还是以固定的坐标作为视标的出现的位置，那么当测试者移动后，将看不到视标，故为了解决这个问题，将视标的位置改为相对于背景的位置，每次显示前，都回去刷新背景的位置。

4.5 测试与验证

4.5.1 测试原则

- (1) 该测试的目的是为了找出程序中存在的缺陷。
- (2) 不可能测试到每一个角落。
- (3) 尽早的开始测试有助于减少开发成本。
- (4) 不要只专注于某一点的测试。
- (5) 测试用例要经常性的进行更新。
- (6) 不能对不同的模块使用相同的测试方法。
- (7) 必须严格按照测试流程。

4.5.2 测试目的

系统测试的目的如下：

- (1) 为了以最低的成本，找出程序中存在的漏洞和缺陷，衡量系统的兼容性。
- (2) 测试系统运行时是否会出现系统崩溃的情况，衡量系统的稳定性。
- (3) 测试系统进行青光眼检测的响应时间和数据反馈时间，衡量系统的性能。
- (4) 测试系统进行一次青光眼检测的过程，衡量青光眼检测流程设计的合理性，完善程序提升用户体验。

4.5.3 功能测试

- (1) 测试对象：基于 VR 技术的青光眼视野检测系统。
- (2) 测试内容：程序是否正常运行，结果是否准确。
- (3) 测试方法：单元测试 集成测试。
- (4) 测试结果：程序运行正常，结果准确。
- (5) 兼容性测试：在华为上运行状况良好。
- (6) 测试结论：程序正常运行，结果准确，兼容性良好。

4.6 本章小结

本章节主要包含详细设计和系统测试两大块。详细设计从视图模块、业务逻辑模块、数据模块、开发过程中所遇到的问题及解决方法和系统测试四个方面进行详细阐述。视图模块主要用来进行视图渲染，展示了 app 的页面设计。业务逻辑模块讲述了检测系统中各模块之间是如何进行运作的，并对开发中主要用到的算法是如何实现的进行了详细介绍。数据模块主要是处理客户端请求，对数据库进行操作。至此完成了系统的详细设计工作。之后介绍了开发过程中所遇到的较为突出的几个问题以及相对应的解决方案。系统测试主要从测试原则、测试目的、测试环境和测试功能四个方面进行介绍。最后对系统的稳定性、兼容性、流程合理性和时间性能进行测试，给出测试结论。

5 总结

基于 VR 的青光眼检测系统以实现用户无需去医院便能进行青光眼的视野检查为核心，以及时发现、及早治疗、挽救患者视力为目标，解决了目前青光眼检测对医疗设备依赖问题，提高了青光眼的早期发现率，为保护患者视力做出了突出贡献，同时也有助于患者进行病情追踪，为医生提出合理有效的治疗方案提供了保障。便携式青光眼检测系统将会成为未来医学领域青光眼检测的发展趋势，VR 技术的发展加快了这一趋势的实现和推进，因此基于 VR 的青光眼检测系统的开发是应运而生的。本系统的优势和不足之处如下：

系统优势：

- (1) 对目前最新的视野检测策略 SITA 做了移动端的实现。
- (2) SITA 提高检查速度的原因是参考临近点的反应选择刺激光标的强度，对估计的阈值不断更新^[13-15]，基于这一点，设计了一个二维数组，每当数组中的某个位置变化时，其周围相邻的位置也跟着同样变化。
- (3) 使用 canvas 实现数据的可视化。
- (4) 使用了 base64 对客户端与服务器端的传输结果进行加密，增加了网络传输中的安全性。
- (5) 将结果存放于服务器端 SQLite 数据库中，使之每当更换设备进行登录时，也能获取到最近一次测试的结果。
- (6) 使用了余弦相似度算法对数据进行分析，参照数据选用人群的平均值，将自身数据与参照数据当作向量求其余弦值，越接近 1 则结果越与人群相似，反之则不相似。
- (7) 提供分享功能，可将结果分享给他人。

不足之处：

目前系统只能显示检测结果而没有病情的发展过程。

如何使系统优势扩大化，解决系统中存在的不足，将是我未来系统改进的方向，下面是我对系统的展望：

- (1) 对检测的历史数据进行机器学习，使之能根据之前的检测结果，推算出之后的病情发展。
- (2) 对程序进行精简与优化，减轻程序包的体积。

参考文献

- [1] 袁援生. Humphrey 视野检测分析原则[M]. 人民卫生出版社. 2005.
- [2] 刘雨辰,秦海旻.预防失明:用手机早期发现青光眼[J].中国高新科技, 2019(24):108-111.
- [3] 范秀敏.深度沉浸:VR 技术的发展[J].互联网经济,2016(11):32-35.
- [4] 任远,吴梦蝶,刘光丽,宋国强,王常青.移动医疗研究现状和趋势的文献计量学分析[J].医学信息,2021,34(03):12-15+19.
- [5] 余飞,赵晶,陈万里,苏涛永,范理宏.上海三级甲等医院移动医疗应用现状调查[J].中国医院管理,2017,37(02):69-71.
- [6] 钟华,钱朝旭.Humphrey 视野计最新程序和策略的临床解读[J].中华眼科杂志,2020(01):71-72-73-74-75-76.
- [7] 谭永琳,黄润楷,郭俏丽,林绮珊,林岳新.移动医疗 app 的应用效果及完善研究[J].继续医学教育,2019,33(06):64-66.
- [8] 龚滢欣,朱佩文,邵毅.移动互联网医疗在眼科的应用现状及分析[J].转化医学电子杂志,2018,5(05):39-43.
- [9] 杨金东.移动医疗 app 现状与展望[J].医学信息学杂志,2016,37(01):59-61+71.
- [10] 刘佩玉,刘倩,张小曼.可用性评估方法在移动医疗 app 中的应用现状[J].医学信息学杂志,2020,41(11):37-40.
- [11] 辛杰. 基于 Unity3D 的 XR 装配式开发关键技术研究[D].华中师范大学,2019.
- [12] 李津,徐红波.基于 Android 操作系统的指南针应用的研究与实现[J].中国新通信,2020,22(21):96-98.
- [13] 王兰,刘磊,林丁.视野计的原理及发展[J].国外医学.眼科学分册,1999(05):309-314.
- [14] Pham Alex T,Ramulu Pradeep Y,Boland Michael V,Yohannan Jithin. The Effect of Transitioning from SITA Standard to SITA Faster on Visual Field Performance.[J]. Ophthalmology,2021.
- [15] Anderson Scott,Battiston Adrian,Bong Donna,Carrell Nathan,Damji Karim F. Using the 24-2 Sita Fast Humphrey to Detect Visual Field Defects Noted in Patients with Neurological Lesions Impacting the Visual Field Normally Assessed by Octopus Visual Field Testing[J]. Journal of Clinical and Experimental Ophthalmology,2021,12(1).

致谢

转眼间，四年的大学生涯接近尾声，回想开学步入校园时的青涩与紧张，经过四年的知识积累和生活磨练，现在的我自信、冷静且镇定，在这个转变过程中，学校、老师、朋友、同学给了我很大的支持和帮助，借此感谢我在这感谢郑大所带给我的开心与成长，感谢在校园中所遇到的老师和朋友，感谢所经历的事。这是我大学生活中所拥有的体验与经历，他们将伴随着我，成为我生命中最珍贵的东西。

我要衷心地感谢我的毕设项目指导老师王超老师，王老师认真负责，对于我的毕业设计开题和研究方向都提出了许多有建设性的宝贵意见。对于我在毕设过程中所遇到的问题和开发困难都给予不厌其烦的为我分析，提供解决方案和措施，并且对我的错误十分包容。我的毕业设计的开发和论文的编写能够顺利完成，这其中离不开王老师的指导和帮助。

其次，还要特别的感谢我大学中所有的任课老师，是他们教给了我知识，给我的毕设开发奠定了理论基础，但是他们传授给我的不仅仅是专业知识，他们身上所具有的无私奉献的精神和高尚的品格，都在无形之中影响了我的成长和价值观。

最后，我还要感谢我的父母，他们在我迷茫的时候鼓励我，给我指引方向，他们永远是我的坚强的后盾，为我解决后顾之忧。我还要感谢我的朋友们，他们在我的学习生活中给予了极大的关怀和帮助，我的成长和进步离不开我身边的每一个人。

四年的大学生活将是我一生中最美好的时光，这四年为我的生命打上了底色，接下来我要用我的努力去为我的生命增添亮色。我的老师们，朋友们，感谢你们这四年的陪伴、关怀和帮助！

李成玉 2021年5月23日 于郑州大学