

林格曼烟雾黑度检测 APP

《Android软件开发》课程项目

吉林大学 计算机科学与技术学院

答辩人：21191427 潘恺璠

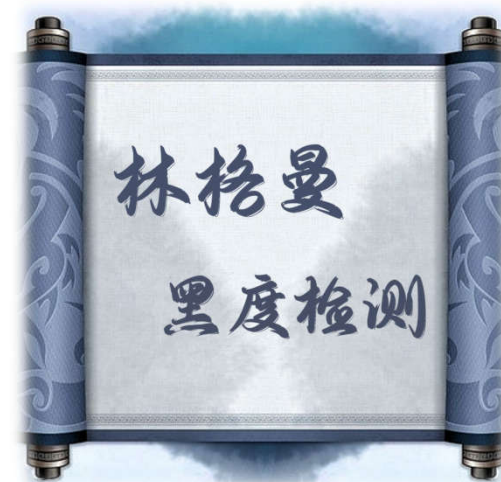
组号：43



目录

CONTENTS

- 01 功能简介
- 02 实现方法
- 03 分工比例
- 04 真机演示





功能简介



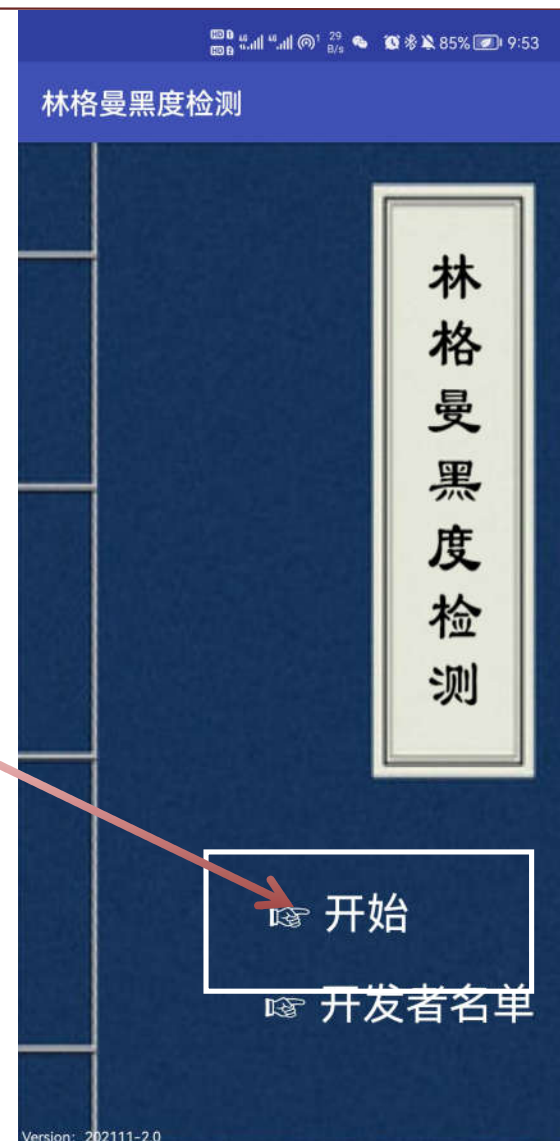
Part 01



功能简介

1. 开始按钮:

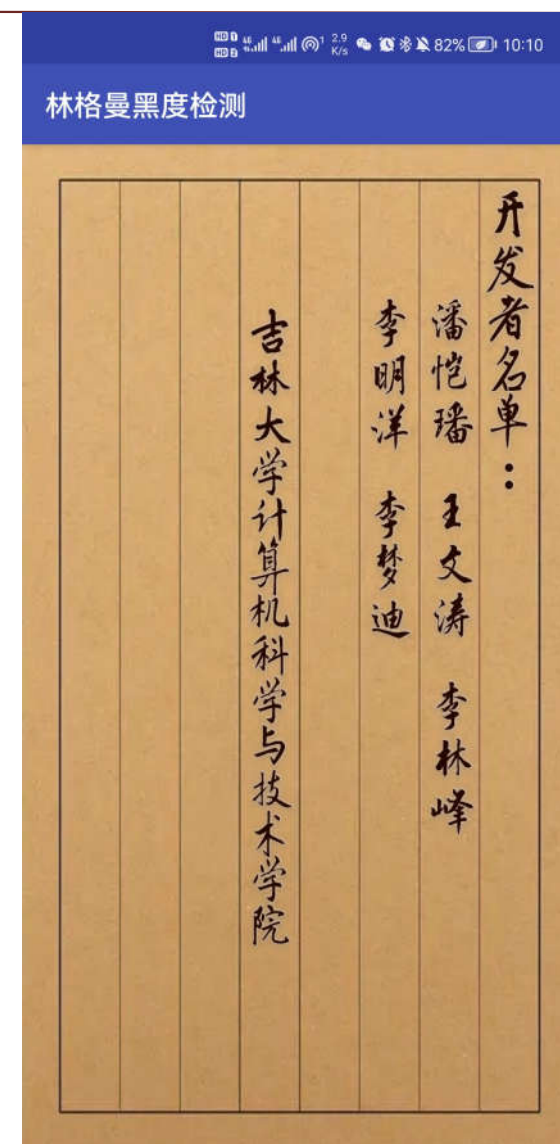
点击“开始”按钮进入主界面使用



功能简介

2. 开发者名单按钮:

点击“开发者名单”
进入开发者详情页



功能简介

3. 手动裁剪功能标签:

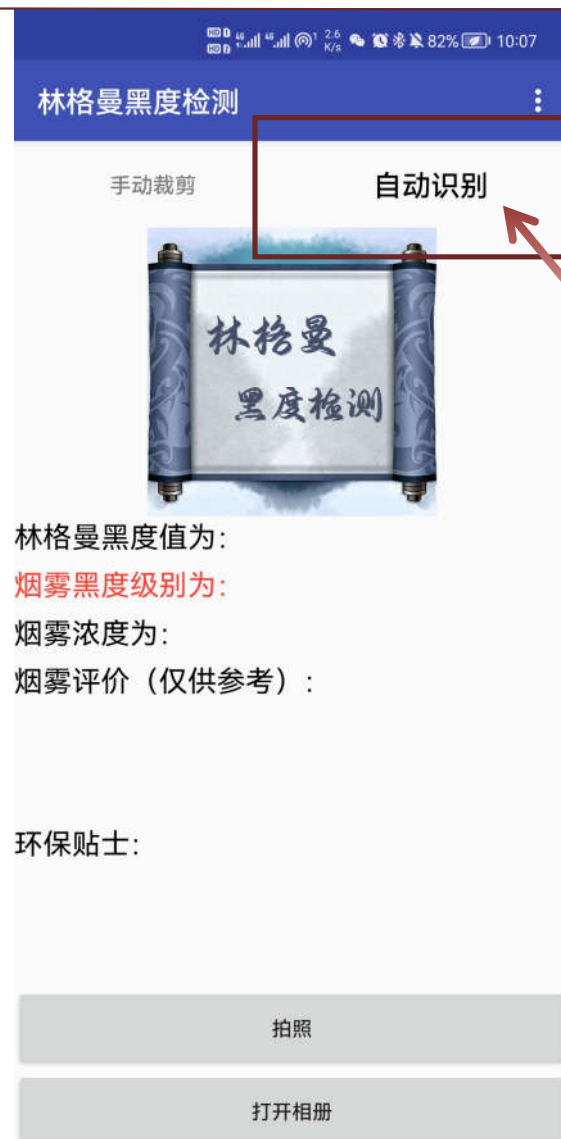
用户在当前标签下可以选择
“拍照”或“打开相册”
两种不同的图片来源，
之后进入到手动裁剪模式



功能简介

4. 自动识别功能标签:

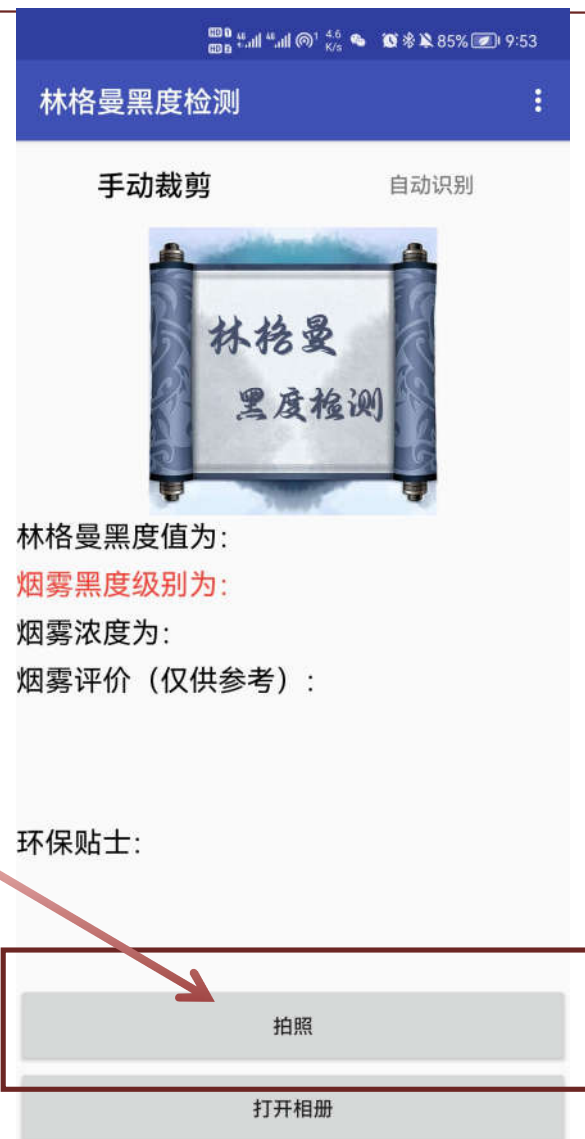
用户在当前标签下可以选择
“拍照”或“打开相册”
两种不同的图片来源，
之后无需裁剪图片，
人工智能会自动识别烟雾位置
并给出检测结果



功能简介

5. 调用相机进行拍照：

点击“拍照”按钮
调用摄像头进行拍照
然后将拍下的照片保存
并保留到下一个裁剪页面

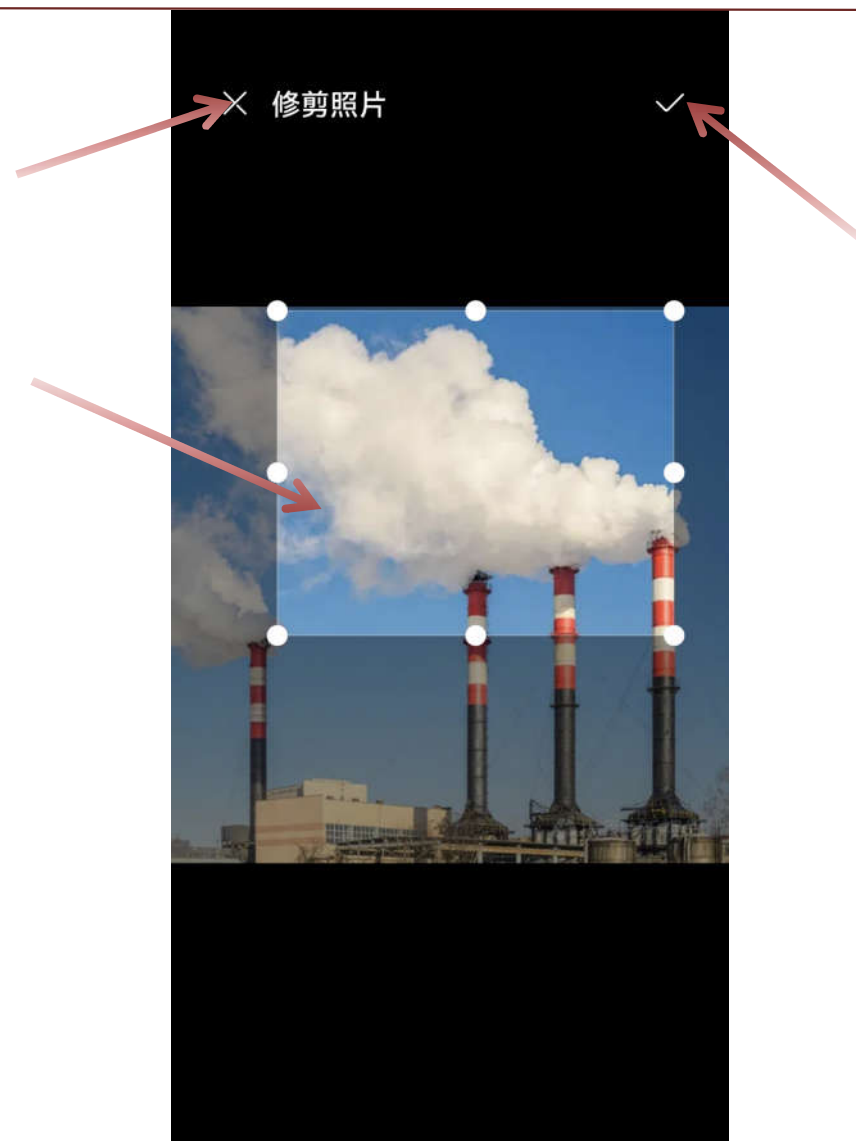


功能简介

6. 手动裁剪：

用户可以自己手动裁剪图片
裁剪完成后

点击“√”则生成检测结果报表
点击“×”则回到主界面



功能简介

7. 检测结果报表:

用户在识别烟雾后，系统会根据算法给出烟雾的林格曼黑度、黑度级别、烟雾浓度等报表，从环保角度给出建议

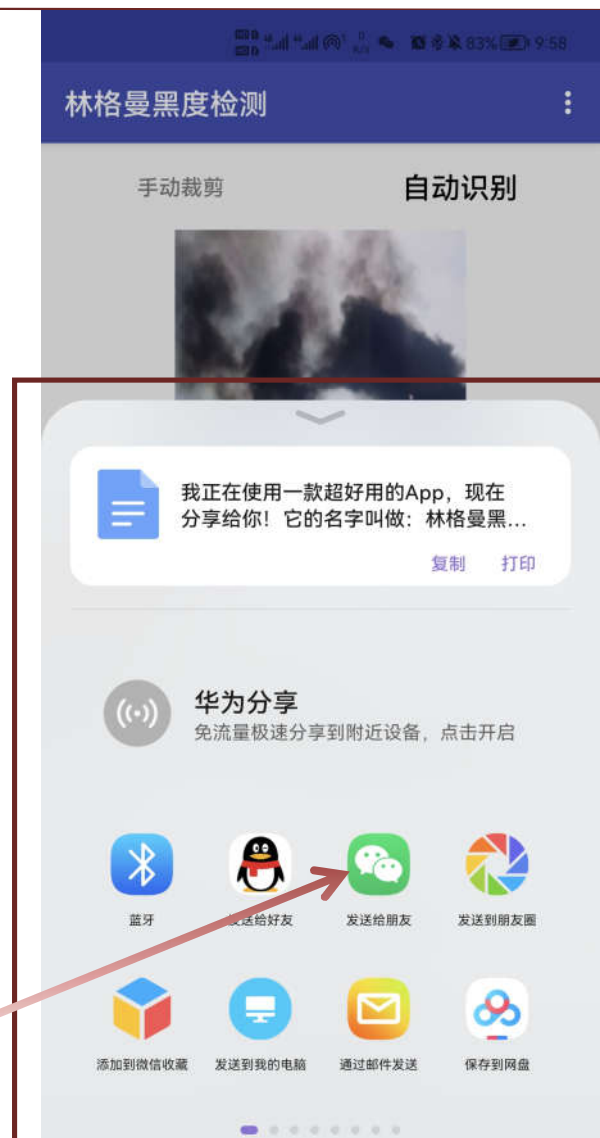


功能简介

8. 分享软件功能:

用户可以在获得良好的软件使用体验后，将此软件分享给朋友。

渠道多种多样，可以是：
短信、邮箱、QQ空间、
微信朋友圈等





实现方法



Part 02

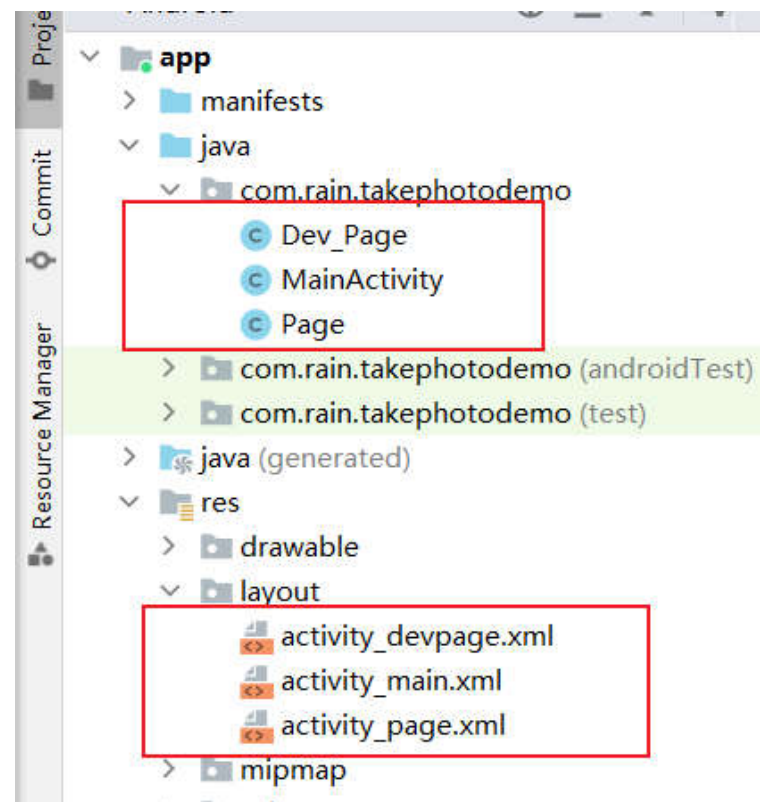


实现方法

1. 多界面视图

模块化页面布局文件与页面Activity

通过监听按钮来实现不同页面的切换



实现方法

1. 多界面视图

模块化页面布局文件与页面Activity

通过监听按钮来实现不同页面的切换

```
@Override
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    findViewById(R.id.btn_takephoto).setOnClickListener(this);
    findViewById(R.id.btn_open_photo_album).setOnClickListener(this);
    img = findViewById(R.id.iv);
    checkPermissions();
}
```

实现方法

2. UI

使用**Constraint布局**按钮和背景图片

字体采用各系统都自带的字体（黑体、楷体等），以防止出现兼容不一的情况

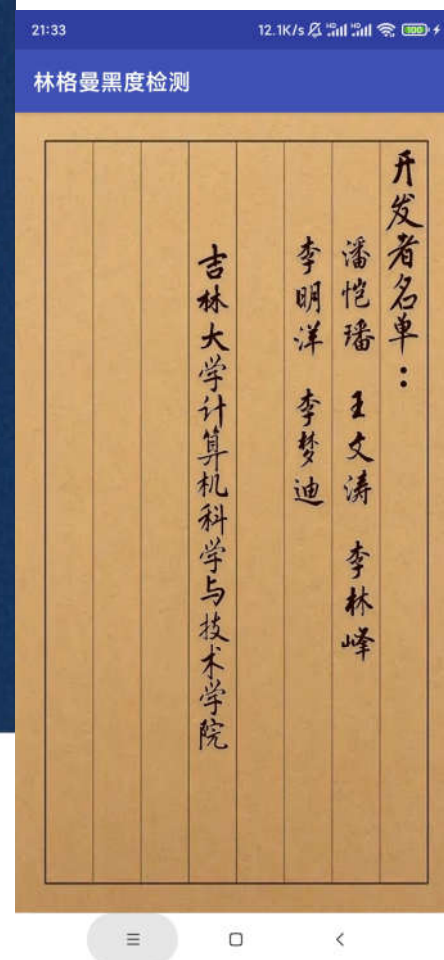


实现方法

2. UI

主界面、开发者详情页由技术精湛的美工设计，
可以给用户良好的体验

考虑到页面切换是从右往左，设计成书本比较
符合，美观大方



实现方法

3. 调用相机、裁剪

使用轮子并做适当修改以适合本项目

引用自github项目：

<https://github.com/YuPf1989/TakePhotoDemo>



实现方法

4. 分享模块

通过调用安卓软件自带分享功能，并查找该用户机上所有可用的软件接口进行分享

向**剪切板内写入**推荐语和软件下载链接，方便用户向好友分享此软件



新建短信

收信人: 爸



来检测烟雾的黑度并给出环保建议的，快快下载，和我一起做环保小卫士吧！下载链接：
<https://pan.baidu.com/s/1Z6Rrldmbo2yk-aSkruD4yw>，提取码：6666

66(3)



实现方法

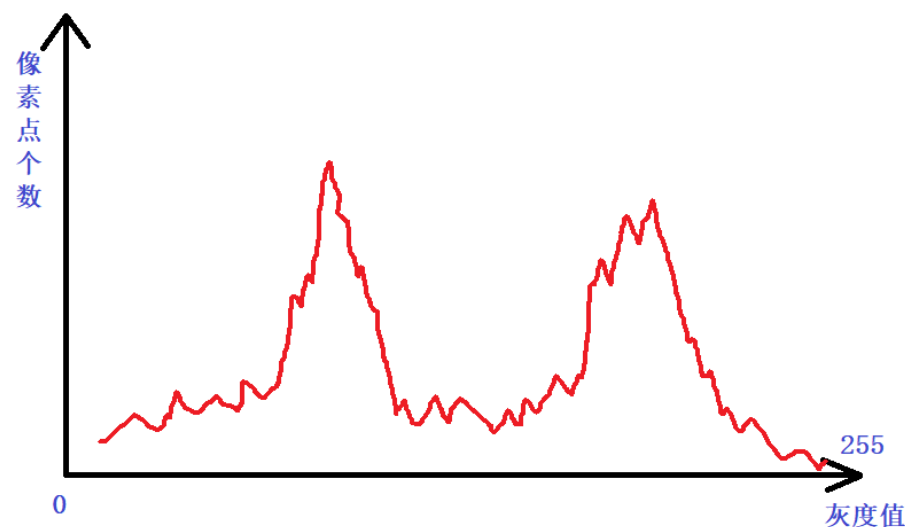
5. 林格曼黑度检测算法

算法思想来源：

桂柏林. 基于林格曼黑度的烟囱排放自动监测分析系统的研究与实现[D]. 湘潭大学, 2009.

复现算法代码 (Java)

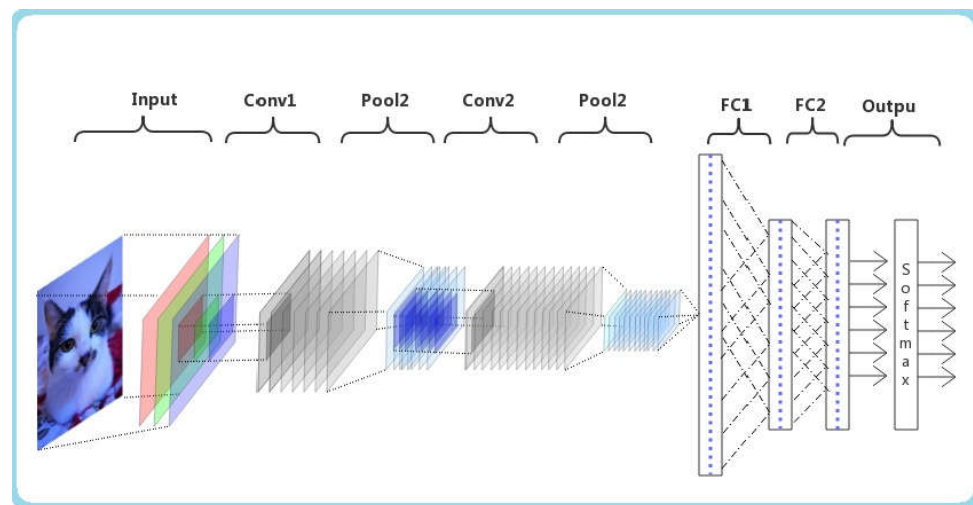
图像灰度化→迭代法分析→找到双波峰→区分烟雾与背景



实现方法

6. 烟雾位置自动识别

深度学习框架——YOLO v3



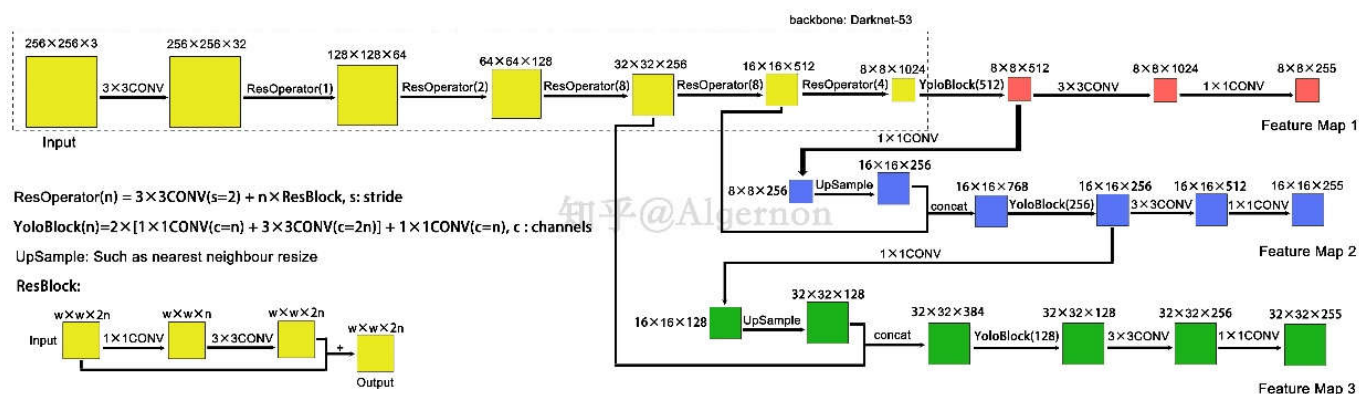
实现方法

6. 烟雾位置自动识别

特征图:

共输出3个特征图

第一个下采样8倍
第二个下采样16倍
第三个下采样32倍



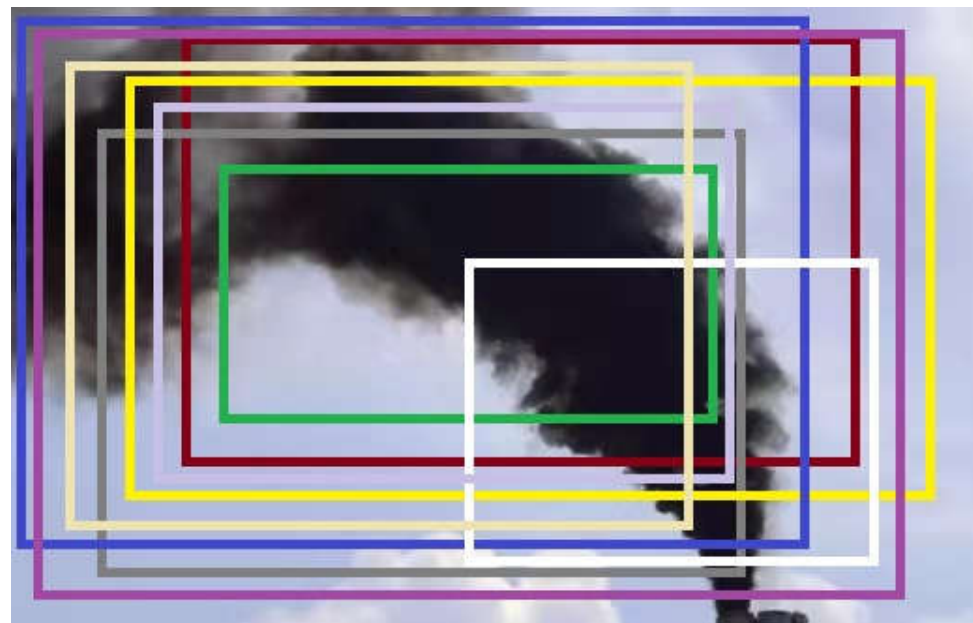
实现方法

6. 烟雾位置自动识别

先验框：

Yolov3沿用了Yolov2中回归检测框的宽、高基于先验框的变化的技巧

使用K-means对数据集中的标签框进行聚类，得到类别中心点的9个框作为先验框。



6. 烟雾位置自动识别

检测框解码:

根据特征图和先验框解码检测框的中心点 (x, y) 、先验框宽 w 与高 h ，进行计算

其中，

$\delta(t_x)$ 、 $\delta(t_y)$ 是基于矩形框中心点左上角格点坐标的偏移量

p_w 、 p_h 是先验框的宽、高

$$b_x = \sigma(t_x) + c_x$$

$$b_y = \sigma(t_y) + c_y$$

$$b_w = p_w e^{t_w}$$

$$b_h = p_h e^{t_h}$$

实现方法

7. 云端服务器与本地调用

采用forest轻量级HTTP客户端框架



实现方法

7. 云端服务器与本地调用

前端部分代码：

```
@Post(url = "http://47.95.148.117:8090/upload", timeout = 120000)
String upload(@DataFile("file") String filePath, OnProgress onProgress);

@Post(url = "http://47.95.148.117:8090/download", timeout = 120000)
@DownloadFile(dir = "${0}", filename = "${1}")
File downloadFile(String dir, String filename, OnProgress onProgress, @JSONBody("downloadPath") String path);
```

创建一个interface，在其中中创建相应接口，调用两个POST请求

第一个：上传待检测的图片，采用json格式返回服务端中识别后的图片路径等信息

第二个：从服务端中下载识别成功的图片（参数为待下载的图片路径）

实现方法

7. 云端服务器与本地调用

后端部分代码：

```
//设置服务端监听端口为8090，成功后回调在控制台上打印提示
let server = app.listen(8090,()=>{
  console.log('The server is listening on port : 8090')
})

1 source /root/anaconda3/etc/profile.d/conda.sh
2 conda activate pytorch
3 path="/root/Android-Smoke-app/smokePredict"
4 io_path="/root/Android-Smoke-app/img_out"
5 cd $path
6 command="python predict.py predict ${1} ${io_path}/img_out ${io_path}/smoke_out "
7 result=`$command`
8 echo $result
9 conda deactivate
```

服务器采用nodejs来搭建通讯，通讯端口为8090

通过调用listen在服务端一直监听8090端口是否有接受到请求，若收到请求则调用相应的shell脚本，执行林格曼黑度检测程序并返回结果



分工比例



Part 03



分工比例

潘恺璠：23%

林格曼黑度检测算法、代码优化、接口、PPT

李林峰：23%

烟雾识别算法、云端服务器与本地调用、接口

王文涛：18%

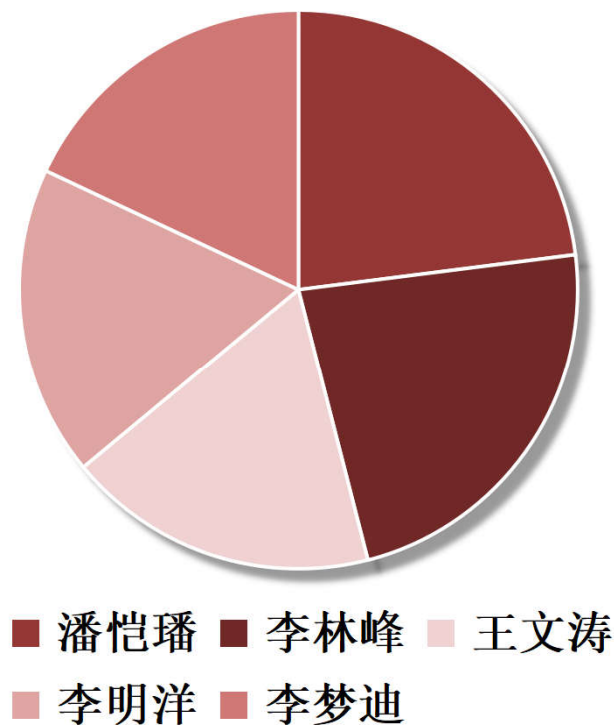
相机与裁剪功能、多界面调用、接口

李明洋：18%

界面布局与设计、分享功能、实验报告

李梦迪：18%

页面设计与优化、实验报告、PPT





真机演示



Part 04



下面请观看真机演示

演示环境：Android 11



感谢聆听
欢迎提出宝贵意见

THANK YOU!