**google地图总结报告**

**一：软件技术框架**：

a：地图下载：

地图下载采用C/S模式，本机或任意一台机器作为服务器，主要作用是响应客户端发送来的请求，将客户端所需数据源源不断的发送过去，客户端主要是向服务端请求地图下载的区域，经纬度范围（起始经纬度，终止经纬度）以及是否已经被其他客户端请求过的状态。

地图下载时，客户端使用一个主程序，使用一个固定大小的线程池进行下载，保证线程池中每个线程都处于活动状态。另外有一个线程，我称为辅助线程，他的作用是在主程序进行下载时，对一些出现异常，未下载完成的图片，获取此图片的经纬度范围，进行二次下载；除此之外，还有另外一个线程，我称为降级下载线程，他的作用是，如果辅助线程并未将这张图片下载下来，那么就对图片进行降级下载（16级）。这样处理之后，加强了区域图片下载的完整性，但是仍会有一小部分图片未被完整下载下来，或者是图片降级到16级，尚未想到方法，将16级图片放大至20级之后，如何找到相适应的图片。

地图下载存储时，为防止存储混乱，采用的方法是，先根据区域下载的经纬度范围，求出行数，列数，文件以：省——市——区（县）——地图级数+行数——列数+纬度+经度存储，这样存储就不会混乱，同时，为后面在界面上加载图片提供方便。如：安徽省——黄山市——屯溪区——20-4——2-29.78612311484704-118.19380925364271.png。



b：软件安装：

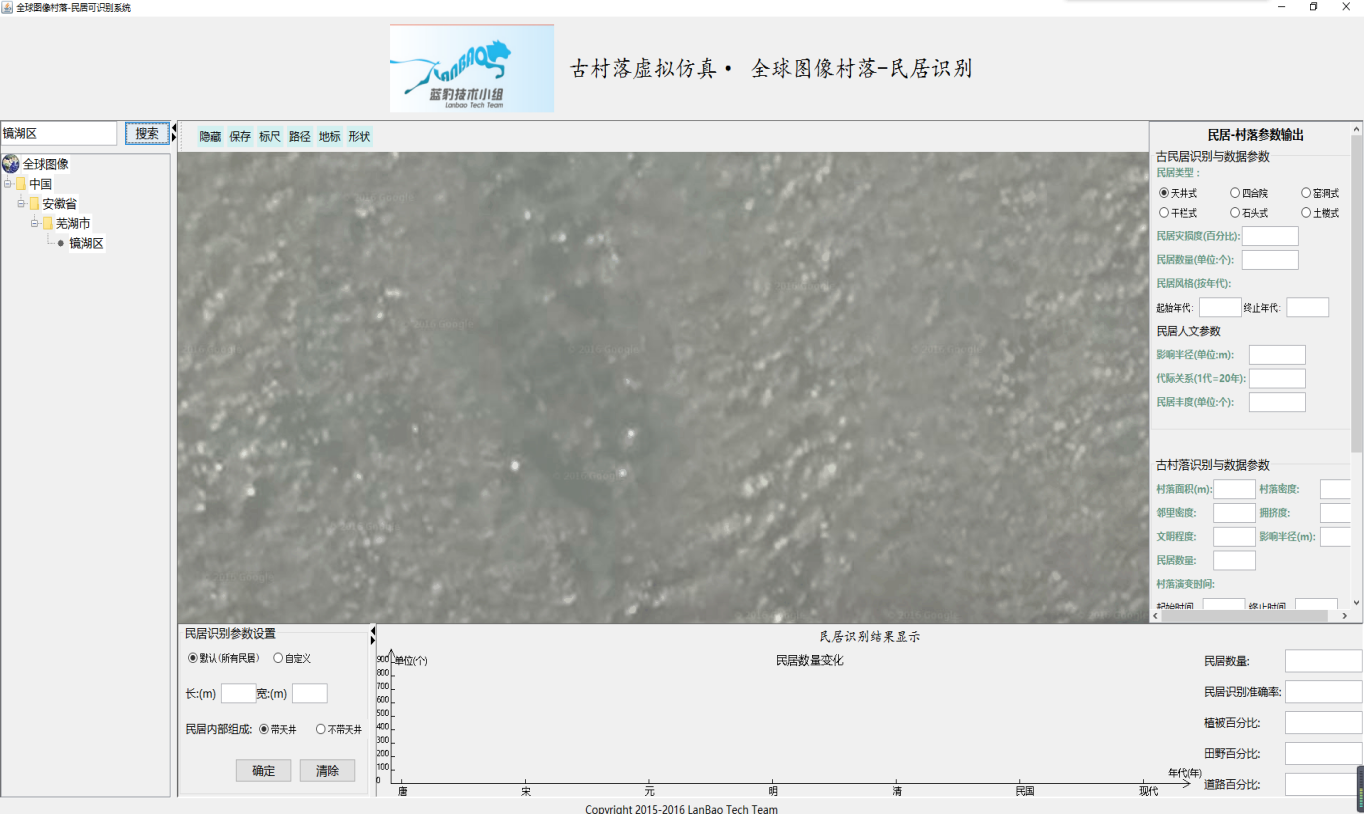
机器及系统要求：要求机器有JDK环境，内置hosts文件需重新替换，网络良好，可不间断上网。地图下载以及界面程序，点击运行即可。

c：软件模拟使用过程：

软件界面上，若想查看某一区域的地图，只需搜索即可（磁盘上地图已经被下载），但是关于民居以及村落的参数，如民居灾损度，民居丰度，民居代际关系，村落密度。拥挤度，文明程度等参数，还需后期提供。

d：软件界面说明：

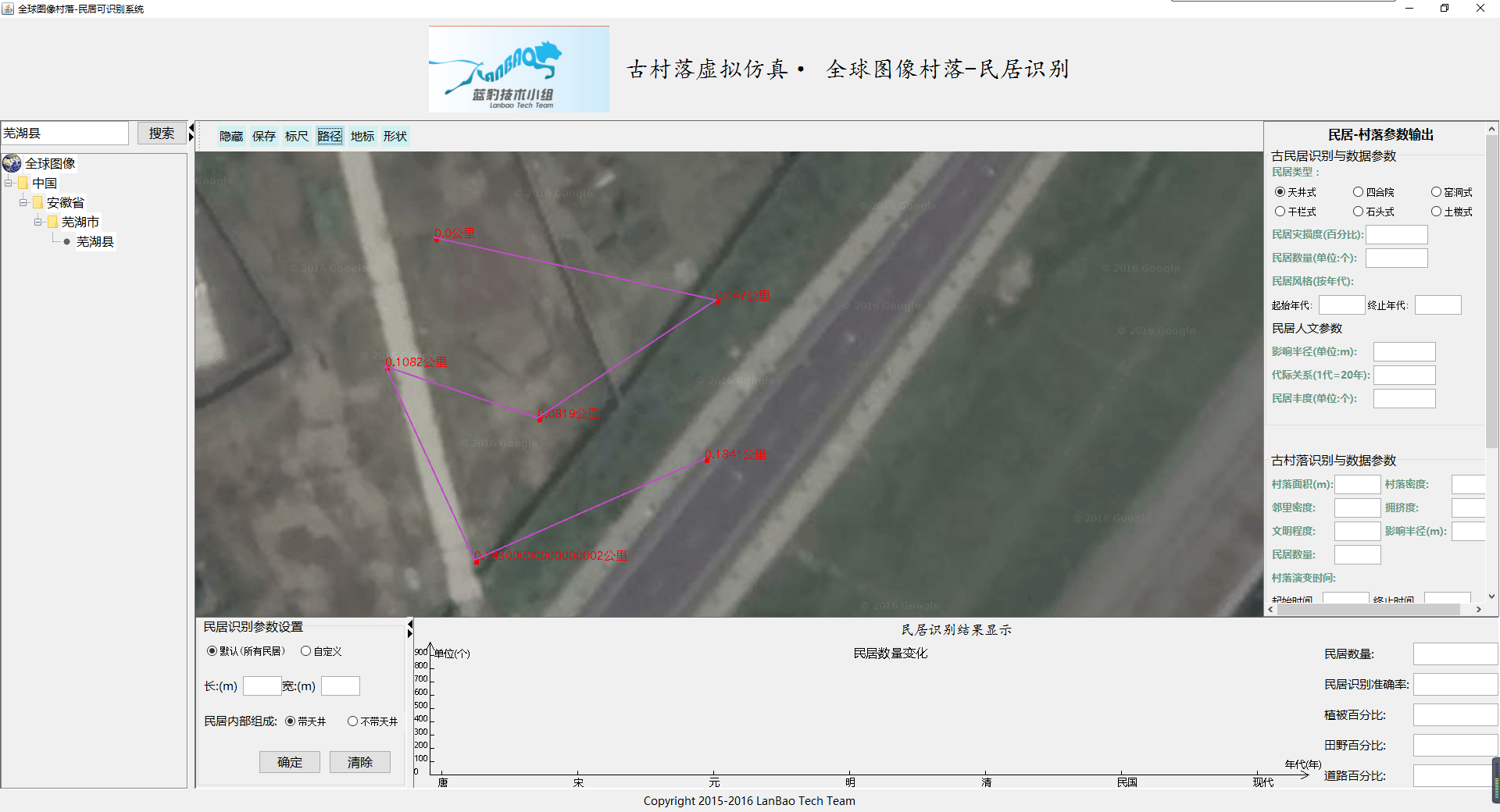
（1）：地图搜索，在搜索栏搜索任意区域，都可显示该区域地图（精确到区县），由于地图过大，采用动态加载的方式，关于镇、村也可搜索，采用的方式是调用google API，将区域的地理编码以及精确经纬度求出，将地理编码，按照中国行政区划分列表，以左侧树的形式展示出，其中树是自动生成，界面如下：



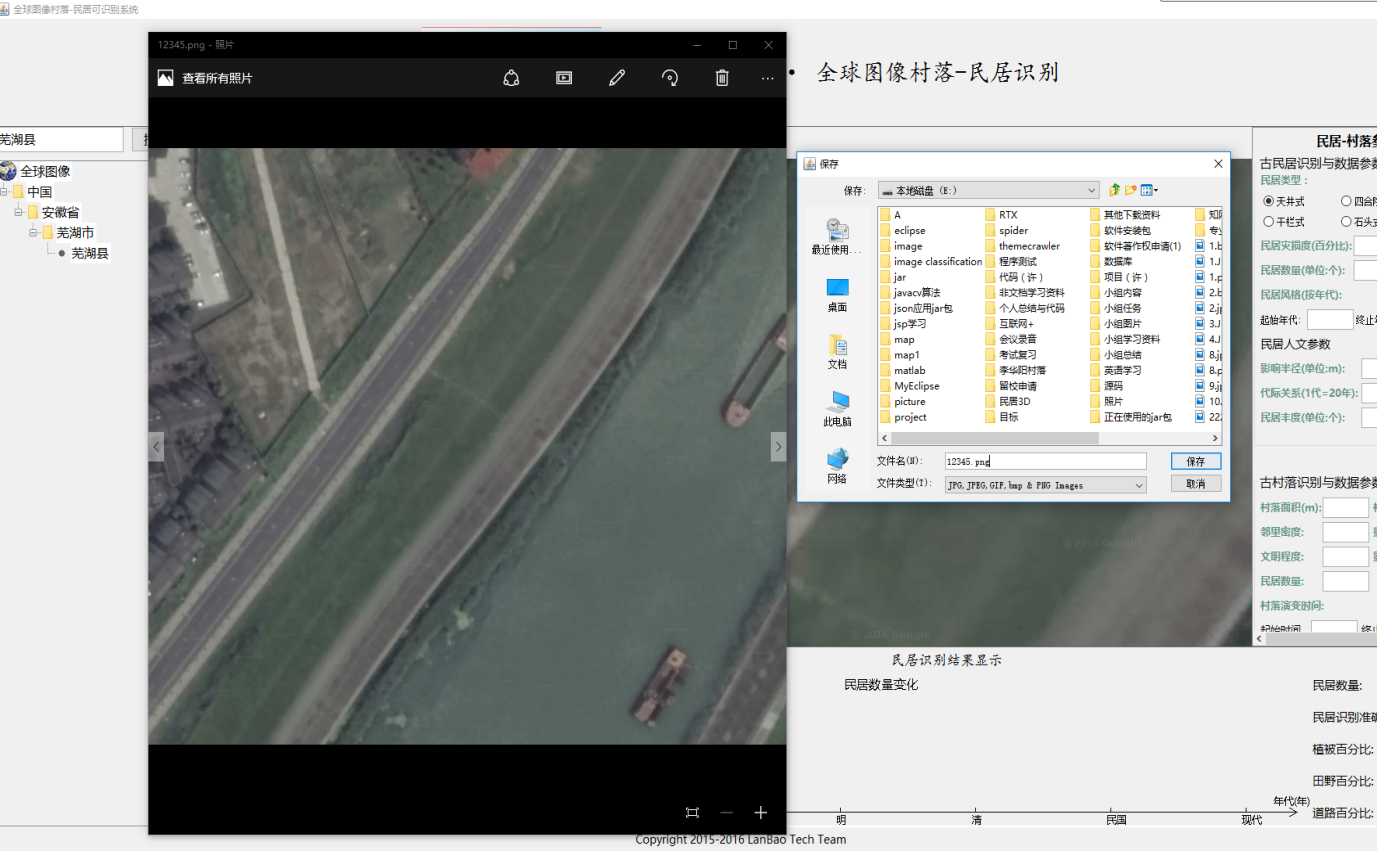
（2）：地图上显示点击地图，可根据经纬度查询区域名，方法还是调用google API，将经纬度转换成地理编码，转换成区域精确地址或模糊地址，再将识别的区域以及经纬度入库，下次识别时，先与库中数据进行对比，没有的话，再调用google API重新识别，此目的是，google API每天调用的数量是有限制的，防止调用次数过多，google封掉IP，（注意，此识别是有误差的，调用的google API与google map上经纬度表示的区域是有差别的）界面如下：



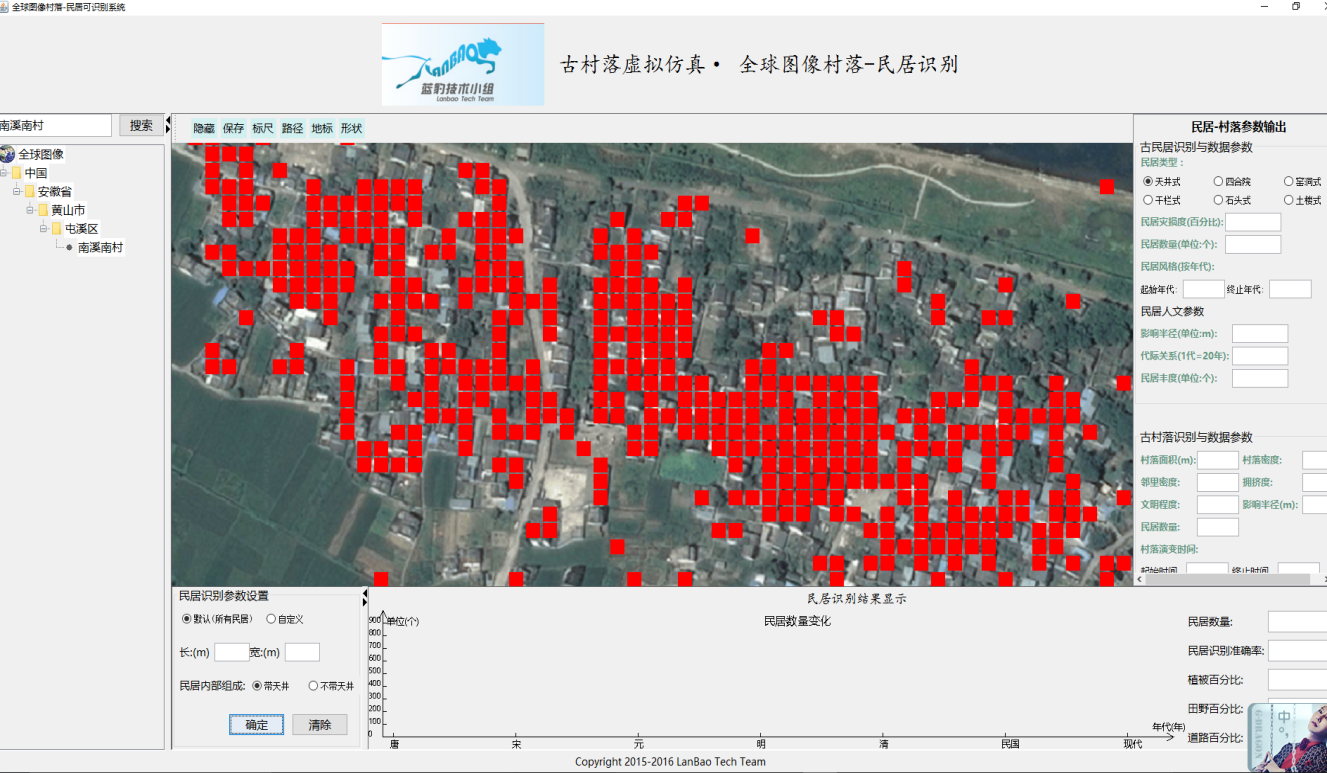
（3）：地图上显示测距，根据加载的图片最后一张的经纬度，根据像素换算，求出地图上点击任意一点的经纬度，再根据两点经纬度的范围，求出两点的范围。（此项仍有误差，通过赤道的半径，加上经纬度的范围，通过换算得出的距离，与实际距离有20~50m的差距）点击“路径”按钮，在地图上任意点击两点，开始测距，界面如下：

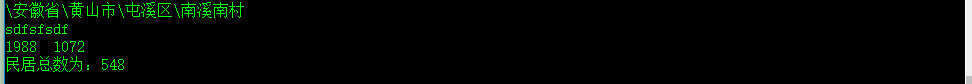


（4）：点击保存按钮，可将地图加载的图片（8\*8）保存起来，此功能是仿照google earth添加上去的，界面如下：

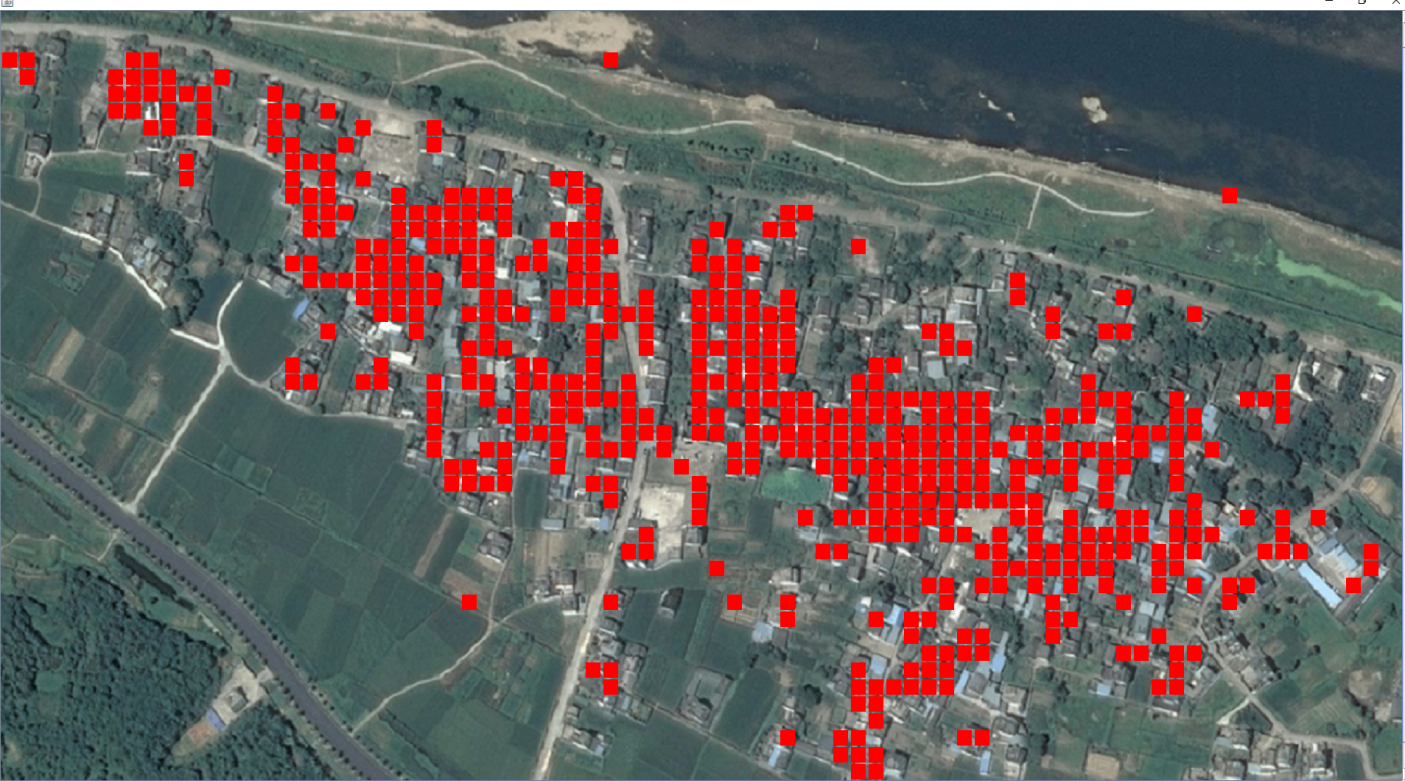


（5）：民居识别，民居识别主要是识别南溪南村的民居，最后识别总数是548栋民居，之后从网上查询得知，宏村、呈坎、婺源等地区是徽派民居的代表，尝试识别了宏村、呈坎，宏村识别的不准确，呈坎的较为准确。南溪南村识别如下：

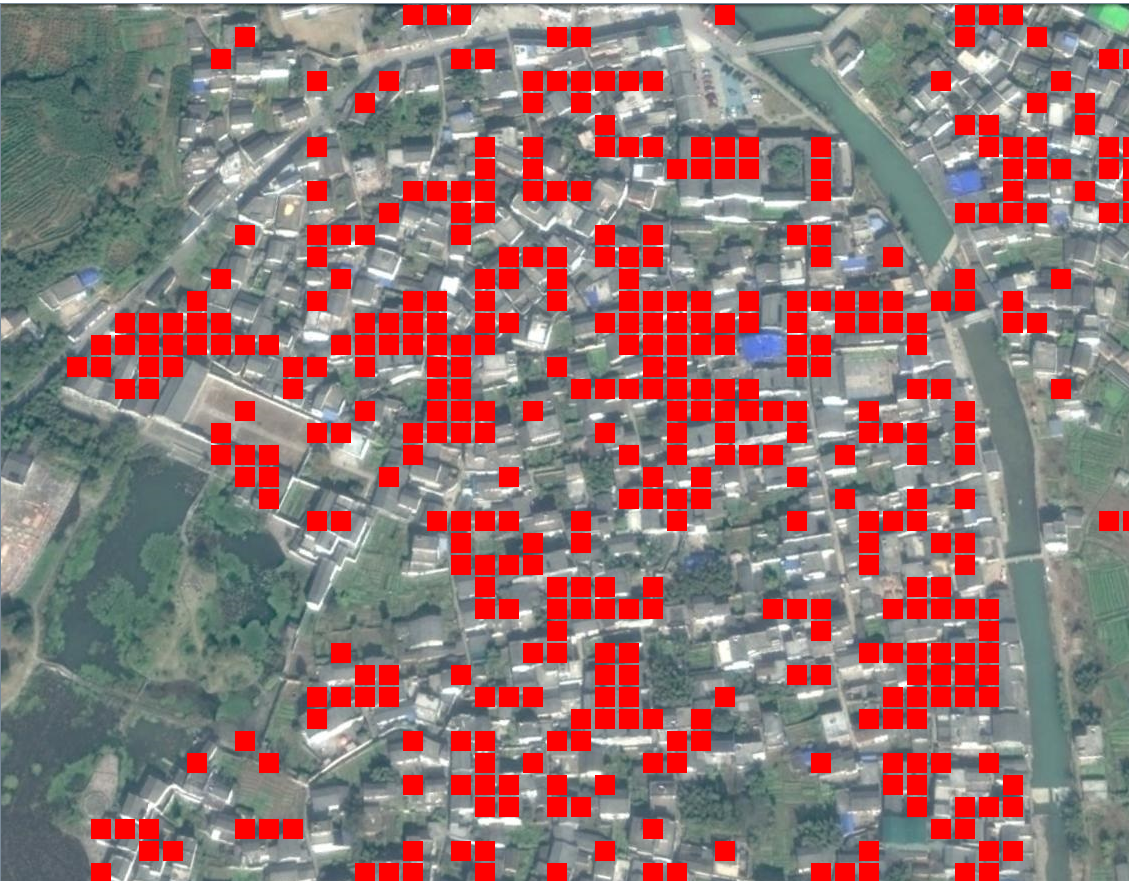




精确识别图像如下：



呈坎识别结果如下：



**二：数据库建立：**

数据库表库结构如下：



说明：

数据共有两个库，一个是地图下载库downloadmap，有4张表，nation\_division，break\_place，area\_recognition，mapFingerprint，其中除mapFingerprint表之外，其余表中皆有数据，图片的指纹不知道如何提取才更加准确。另一个库是图像参数库，有4张表，houseParameter，vilageParameter，results，historyRecord，其中数据大部分由模拟构成，准确数据需后期提供。

**三：未完善：**

a：现在正在思考老师开会的时候说的图像切割算法，服务端采用算法将下载好的地图切割成若干块，分发下去，客户端利用算法进行民居识别，目前正在思考，要将一张多大的图片分配下去是最恰当的，现在还没有太大进展（现在思考出现的一个问题是，地图分区域下载，下载的是方形形状，可以将某个区域包括进去，但难免有区域被重复下载，这样将会导致民居重复识别，这是后期需要解决的问题，目前先将算法以及服务器的设计考虑好）。

b：前期设计界面时，有关于民居历史数量的显示（以柱状图的形式），在进行民居识别时，极难得知民居的朝代信息，所以我想，是否有各个村落关于民居历史数量的记载，直接入库，在界面显示即可。另外，软件设计初期关于识别部分，有识别的道路百分比，田野百分比等参数，这部分的识别，我尚未考虑，因为在识别之前需得知村落的面积等参数。所以，我想是否这些参数需要识别，并在界面上显示？

c：关于地图下载，在民居识别时，我发现18级的地图比较适合做民居识别，20级的地图太大，民居的屋顶也很大，一个村落的地图太大，会影响民居的识别速度。所以在下载时，是否要考虑18级图像的下载？

d：关于民居识别，民居识别的结果（以南溪南村为例），识别准确率约为75%~80%，有部分靠近道路的民居，或单独的民居未被识别到，有部分民居可能在计算时，被计算了2次。在民居识别时，后期尝试了一些较为复杂的算法，如边缘检测算法中的canny边缘检测算子，roberts边缘检测算子等，也了解了基于二阶导数的高斯-拉普拉斯算子，我发现没有单独一种算法是准确的，而且，基于一阶导数的边缘检测算法较好理解，高斯-拉普拉斯算法不太好理解，并且，这些算法我尚未找到切入点与其他算法结合在一起使用，也就无法再次提高民居识别的准确率。（还可再深入思考）