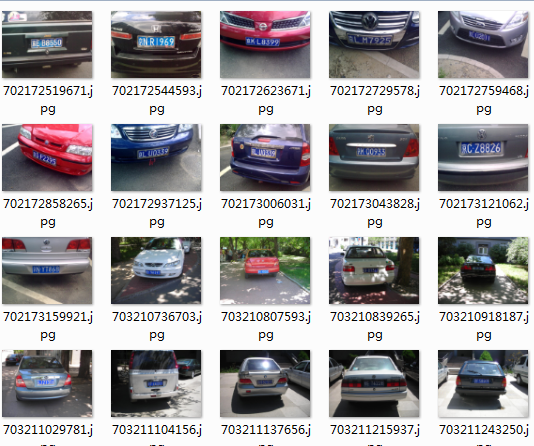
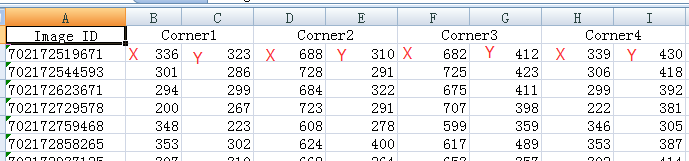
实验二：车牌定位和识别

实验材料：

1.150张包含车牌的图片



2.用于校验程序正确性的车牌所在位置坐标表格

****

**实验的目的：**

**1.学会使用常见的图像预处理方法，对图像进行处理。提取出需要的部分**

**实验过程：**

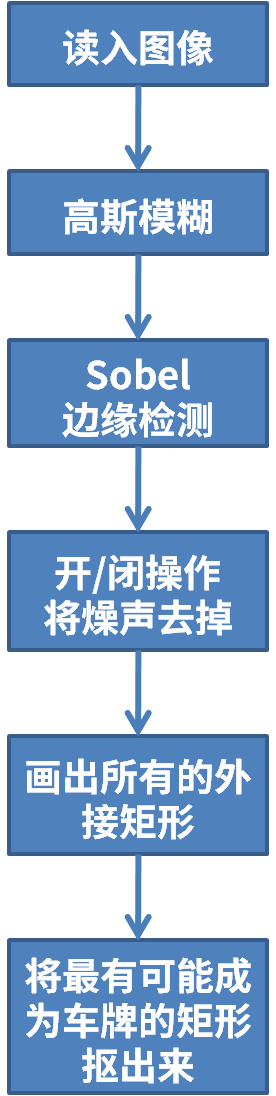
**老师在课堂上大致说了几个步骤：二值化→边缘检测→根据长宽比将车牌从图片中抠出来。**

**但是实际上无数多坑。前前后后坑了一个星期才弄通。**

**开坑还是感谢“计算机的潜意识”写的《EasyPR--一个开源的中文车牌识别系统》。http://www.cnblogs.com/subconscious/p/3979988.html**

**由于本次实验做到中段就转方向，中间泄及的技术细节和原理就不在本文提及。详细将在《ofo车牌定位》一文中概述。**

**流程如下图：**

****

**由于我是挑这张 京E.B8550来做，当然是符合预想。**

****

****

|  |
| --- |
| **carIdentify.m** |
| % function carIdentify ()  clear all;  clc    [Image\_ID] = csvread ('Plate\_Index.csv',0,0,[0,0,0,0]);    for i =1:length (Image\_ID)  filename=[int2str(Image\_ID(i)),'.jpg'];  filename  img = imread (['Plate\_Image\',filename]);    %高斯  a = imgaussfilt (img, 3);    %灰度化  a = rgb2gray (a);    %sobel边缘检测  a = edge (a, 'Sobel');  % imshow (a);    %开闭操作  % 开运算：去除较小的明亮区域  % 闭运算：消除低亮度值的孤立点  se = strel ('rectangle',[7 27]);  a = imclose (a,se);    a = imopen (a,se);    se = strel ('rectangle',[7 95]);  a = imclose (a,se);    se = strel ('rectangle',[11 3]);  a = imopen (a,se);    imwrite (a,'imout.jpg');    %画出所有的外接矩型(代码来自MATLAB中文论坛)  [l,m] = bwlabel (a,8);  status = regionprops (l,'BoundingBox');  figure(10);  imshow (img);  hold on;  for j = 1:m  rectangle ('position', status(j).BoundingBox, 'edgecolor', 'r');  end  hold off;  frame = getframe;  rec = frame2im(frame);  imwrite(rec,['S3\_Rectangle\_Image\',filename])    %查找最接近的图形  for k = 1:m  %左边距的x坐标<200的，丢弃  if status(k).BoundingBox(1) < 200  status(k).BoundingBox(1)  continue  %横向尺寸少于200的，丢弃  elseif status(k).BoundingBox(3) < 200  status(k).BoundingBox(3)  continue  elseif status(k).BoundingBox(4) < 50  status(k).BoundingBox(4)  continue  else  i2 = imcrop (img,status(k).BoundingBox);  imwrite (i2,['S4\_Crop\_Image\',int2str(Image\_ID(i)),'\_',int2str(k),'.jpg']);  end  end  end |

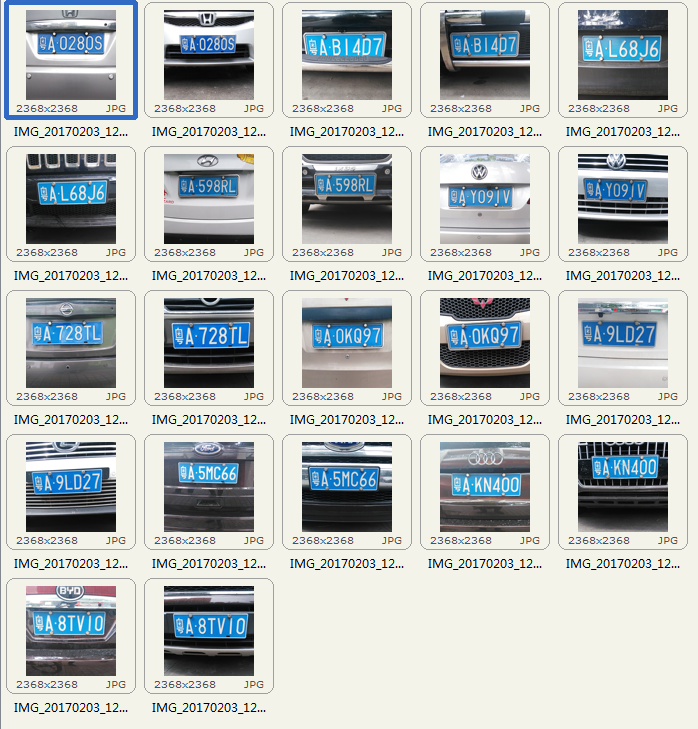
**但是一但将这个程序放到150张图片时，瞬间就雷倒了。**

****

**只有21张图片是符合预想情况可以进行下一步的操作，其它图片都相当难处理，各种夸张的情况都有。当然用来做高技术的研究是不错的。但是却缺乏实用性。**

**并且广州基本上是以“粤”字头为多，所以，干脆训练集自己重新做。**

**重新制作训练集的困惑，拍了22张车牌照片。忽然发觉一件事：别人很容易会误解我在干嘛，别人以为我是便衣警察去抓违章......汗！**

****

**突然间发觉ofo的APP写得特烂，连个二维码扫描都没有。ofo车牌上的二维码，仅仅是下载APP的链接（开始调研时只有6位ofo，后期7位ofo上面的二维码就直接带了数字）。**

**所以，ofo车牌识别，开工吧！**

首先先调查各大应用市场的车牌识别情况，车牌识别对于android来说显然是低频非刚需，而市场上也己经有人做了，识别的正确率还是挺高的。



再来看看目前火爆的共享单车二大巨头：ofo和摩拜，摩拜车上己有完整的二维码开锁系统，对ocr识别显然没有需求。但是ofo，显然没有为互联网和物联网做准备，APP中需要手工输入车牌号码，然后再开锁。(调研时只有6位ofo，没有7位ofo)



再查了一下专门针对ofo车牌进行识别的应用，目前依然没有。



并且去拍ofo车牌比较容易：

1.ofo是停在人行道上

2.拍ofo不会被人认为是抄牌之类的误解

3.ofo产量惊人，铺货是10万台、10万台那样上线，随便路上都可以找到

进一步分晰，ofo车牌由0~9数字组成(10种分类)，远比汽车车牌的容易得多(31个省份+24个英文字母+10个数字=65种分类)。



毕竟作为学习，而非做产品或项目，只需要明白原理就好。所以就选择ofo。

##开发平台、工具、语言的选择

而开发平台的选择上，ofo作为移动互联网时代下的产品，做成web或者pc端的产品也太不符合应用场景了。那就干脆做成微信公众号、小程序、APP之类。

一开始我是想做成微信公众号，毕竟这是低频非刚需：拍照发到公众号，公众号进行识别返回结果。但是微信公众号，显然与自己服务器上什么都可以放上去的不同，只能做web。

让微信公众号将照片先发到自己服务器上，处理后再返回结果。但是公网服务器需要购买云主机并且还要申请域名，对于仅仅是做学习研究来说，付费显然不太友好。

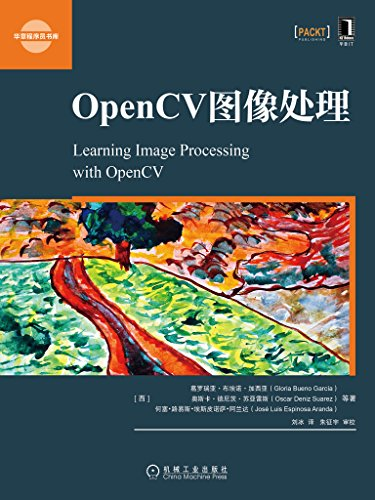
之后，再来看看微信小程序，微信小程序主推是线下导流，线下放置二维码，然后下载。并且体验了几个小程序，有点回到5年前时android应用的感觉。限制极多，并且也是要服务器。所以只得放弃。

最后只得在APP上开发。APP虽然推广成本极高，适配麻烦。但却有着强大的本地化资源可用的优势。并且OpenCV有Android版的API。而Matlab却没有Android版API。

再核查OpenCV，也是有腐蚀、膨胀、高斯模糊、外接矩形、svm算法。满足开发需要。

所以移动端就采用Android+OpenCV

而刚好去图书馆时发现《OpenCV图像处理》(ISBN: 9787111527473)一书，有书就比网上凌散的资料开坑爽多了。



所以，就先将windows版的Qt+OpenCV环境搭起来，调试没有问题，再通过JNI接口方式，移植到Android。

而在开发windows版时还是以6位ofo为主，但是ofo的产能真的太猛，一下子就上到100万号(太有钱了！)。所以前期做了个6位ofo车牌识别(在 "6位ofo识别" 目录中，主要以795116)。

下面主要讲解7位ofo车牌识别的过程。