

# Report

- Name: Zhang Qi
- ID: 17343153
- Email: [zhangq295@mail2.sysu.edu.cn](mailto:zhangq295@mail2.sysu.edu.cn)

## 一、实验内容

1. 字符切割:完整切割出单个汉字和数字字符,思路可以使用作业5和作业6的思路,但是需要保证汉字字符的完整性.
2. 字符识别:可以选择简单和复杂两种之一完成(有能力的同学可选择完成复杂任务,最终得分会更高).
  - a)简单任务:只识别名片中的手机号码;即只需要训练数字的分类器,并完成识别名片中的手机号码识别.
  - b)复杂任务: 识别名片中所有信息(包括数字和汉字);即只需要训练数字的分类器,并完成识别名片中的手机号码识别.(可用Tesseract 库)

## 二、实验环境

- Ubuntu 18.04LST
- g++ 7.4.0
- OpenCV 3.2
- Tesseract

## 三、实验过程及结果

### 1. 字符切割

通过任务一我们可以得到像下面这样的工整的名片:



然后需要对其进行切割成一个个字符,整体思路是:

- RGB图转灰度图
- 灰度图自适应二值化得到黑白图
- 去除面积小的连通块(噪声)
- 膨胀

- 使用面积+纵横比的限制来找到“有意义”的字符（串）
- 使用垂直投影分割对上述字符串进行切割得到结果。

```
// 垂直投影分割
void save_single(Mat img, int num)
{
    vector<vector<Point>>cnts = dilate_contours(img, 5, 3);
    for(int i = 0 ; i < cnts.size(); ++i) {
        Mat sub_img;
        if(contourArea(cnts[i]) > 50 && contourArea(cnts[i]) < 2000){
            Rect temp = boundingRect(cnts[i]);
            if(temp.height / temp.width > 3 || temp.width / temp.height
> 3) continue;
            if(temp.x > 0) temp.x -= 1;
            if(temp.y > 0) temp.y -= 1;
            if(temp.x + temp.width < img.cols - 2) temp.width += 2;
            if(temp.y + temp.height < img.rows - 2) temp.height += 2;
            sub_img = img(temp);
            //
            imwrite("./ans_pic/part1/"+to_string(num)+"_"+to_string(i)+".jpg",sub_i
mg);
        }
        else
        {
            continue;
        }

        int* hist = new int[sub_img.cols];

        for(int j = 0; j < sub_img.cols; ++j) hist[j] = 0;

        for(int x = 0; x < sub_img.rows; ++x){
            for(int y = 0; y < sub_img.cols; ++y){
                if(sub_img.at<uchar>(x,y)>128){
                    hist[y]++;
                }
            }
        }

        int left = 0, right = 0;
        bool white = false;
        int count = 0;

        for(int j = 0; j < sub_img.cols; ++j) {
            if(hist[j]){
                white = true;
                if (j == sub_img.cols - 1){
                    right = j;
                    Rect r(left,0,right-left,sub_img.rows);
                    imwrite("./ans_pic/part1/"+to_string(num)+"_" +
to_string(i) + "_" +to_string(count)+".jpg",sub_img(r));
                    count++;
                }
            }
            else {
                if(white){

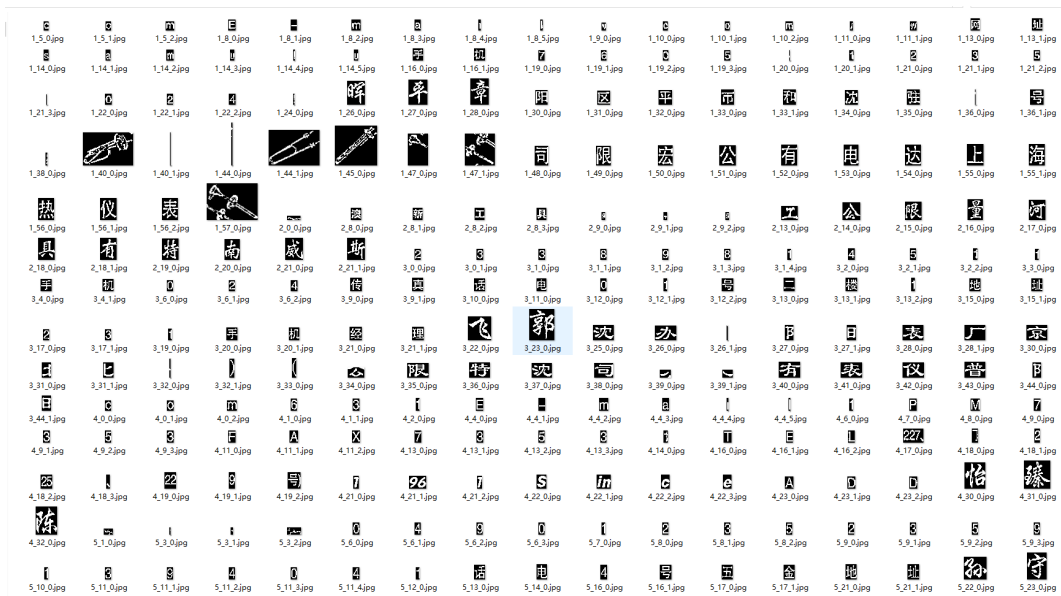
```

```

        right = j;
        Rect r(left,0,right-left,sub_img.rows);
        imwrite("./ans_pic/part1/"+to_string(num)+"_" +
to_string(i) + "_" +to_string(count)+".jpg",sub_img(r));
        count++;
    }
    left = j;
    white = false;
}
}
delete[] hist;
}
}

```

得到的结果如下:



可以看到除了一些图片上的“图案”被误判成字符、个别汉字没有被整体识别以外，绝大部分还是能够正确切割的。

## 2. 字符识别

完成情况如下:

☒ 手机号码识别

因为名片的提供的信息不完全一致，而且各种信息的分布也不一致，所以我们只能够通过一种比较“朴素”的方法来找到手机号码。在本程序中我只实现了对于11位手机号码的识别。

首先使用21\*3的核对黑白图进行膨胀操作，得到水平方向上连成一块的有意义的信息。

然后将这一部分信息作为一个整体，加上纵横比的限制定位到名片中的手机号码的位置，将这个位置截取下来：



然后调用训练好的Tesseract库对这些图片进行识别。

```
void recoginze_phone(Mat img, int num)
{
    vector<vector<Point>> cnts = dilate_contours(img,21,3);
    Mat sub_img;
    for(int i = 0; i < cnts.size(); ++i){
        Rect temp = boundingRect(cnts[i]);
        if(temp.width / temp.height >=5 && temp.width / temp.height <= 8 &&
temp.height > 15){
            if(temp.x > 0) temp.x -= 1;
            if(temp.y > 0) temp.y -= 1;
            if(temp.x + temp.width < img.cols - 2) temp.width += 2;
            if(temp.y + temp.height < img.rows - 2) temp.height += 2;
            sub_img = img(temp);
            imwrite(to_string(num)+"_"+to_string(i)+".jpg",sub_img);

            tesseract::TessBaseAPI tess;
            tess.Init(NULL, "eng", tesseract::OEM_DEFAULT);
            tess.SetPageSegMode(tesseract::PSM_SINGLE_BLOCK);
            tess.SetImage((uchar*)sub_img.data, sub_img.cols, sub_img.rows,
1, sub_img.cols);

            char* out = tess.GetUTF8Text();
            cout << num<<","<<i<<"\t"<<out << endl;
        }
    }
}
```

因为姓名基本都是两个字或者三个字，所以也能够跟【手机号码检测】一样的思路来实现。

不过使用Tesseract库的中文字符识别的效果比较...



```
void recognize_name(Mat img, int num)
{
    vector<vector<Point>> cnts = dilate_contours(img,21,3);
    Mat sub_img;
    for(int i = 0; i < cnts.size(); ++i){
        Rect temp = boundingRect(cnts[i]);
        if(temp.width / temp.height >= 2 && temp.width / temp.height <= 4 &&
temp.height > 30){
            if(temp.x > 0) temp.x -= 1;
            if(temp.y > 0) temp.y -= 1;
            if(temp.x + temp.width < img.cols - 2) temp.width += 2;
            if(temp.y + temp.height < img.rows - 2) temp.height += 2;
            sub_img = img(temp);
            imwrite(to_string(num)+"_"+to_string(i)+".jpg",sub_img);

            tesseract::TessBaseAPI tess;
            tess.Init(NULL, "chi_sim", tesseract::OEM_DEFAULT);
            tess.SetPageSegMode(tesseract::PSM_SINGLE_BLOCK);
            tess.SetImage((uchar*)sub_img.data, sub_img.cols, sub_img.rows,
1, sub_img.cols);

            char* out = tess.GetUTF8Text();
            cout << num<<","<<i<<"\t"<<out << endl;
        }
    }
}
```

#### ☐ 公司识别

没有想到好的办法能够区分公司名、地址之类的其他信息，所以这一部分并未实现。

最终的识别结果如下：

	A	B	C	D	E
1	图片ID	姓名	手机		
2	1	章晖乎	13940363693		
3	2				
4	3		13898145123		
5	4	陈怡秦	13478828508		
6	5		13940412359		
7	6	文富宏			
8	7	春业务经理	18904396555		
9	8	常秀			
10	9	杨杰、郭建彬	13898172265		
11	10		13843967017		
12	11	牛志超	17649984089		
13	12	津达线缆			
14	13				
15	14				
16	15				
17	16	徐风	13897936374		
18	17	潘句房	18940277986		
19	18		13940045598		
20	19	胡余林	15998384841		
21	20		13134486869		
22	21	泽坤			

## 四、实验总结

感觉这个问题有一点麻烦，因为各个名片的提供的信息不完全一致，而且各种信息的分布也不一致，所以很难通过很普遍的方法来辨别出手机号、公司名、人名之类的信息在名片上的位置分布，所以首先需要对图片进行OCR，通过自然语言处理得到相关的信息然后才能定位...定位了之后就能够截取相应的信息进行OCR。恕我愚钝...我实在是想不到更好的解决办法，所以只能够使用最质朴的思想来找到相应的信息来进行无差别分析，找到有意义的信息。