

# 答题卡项目技术总结

(王野，周东卓然，向陈铭)

## 目录

- 一、项目背景.....2
  - 1、项目来源.....2
  - 2、技术要求.....2
    - 1) 四角定位、扭曲矫正.....2
- 二、技术方案.....3
  - 1、开发环境.....3
  - 2、解决思路.....3
    - 1) 四角定位.....3
    - 2) 扭曲矫正.....3
    - 3) 阴影去除、图像增强.....3
- 三、项目成果.....5
  - 1、实验结果.....5
  - 2、指标完成.....5
  - 3、结果展示.....5
- 四、总结.....7
- 五、相关文档、及链接.....8

# 一、项目背景

## 1、项目来源

这个项目是来源一家在线教育公司的实际需求。该项目已经上线了。作为本次图像处理班的阶段性练手。

## 2、技术要求

这个需求实际是给一张答题纸，把答题纸的答题区域提取出来（图中红框），效果增强（去除阴影）。

### 1) 四角定位、扭曲矫正

- 程序拍摄纸张页面
- 由于拍摄角度的原因，页面可能会扭曲，所以需要程序识别、四角定位（黑框内部区域 --- 下图中浅绿色显示区域），然后矫正还原为长方形。

即：矫正后四个角为 90 度，并且长宽比与实际纸张对应区域一致。  
（见下图）



### 2) 效果增强:

包括两个方面：1、色彩增强；2、阴影去除，示例如下：

	原图效果	效果增强后效果
示例 1（色彩增强）		
示例 2（阴影去除）		

## 二、技术方案

### 1、开发环境

#### 1) 实验环境

平台: Anaconda(64bit),

语言: Python 3.6.3

所需的支持模块: **opencv 3.4.2**, **matplotlib 2.1.0**,

**imutils 0.5.1**, **numpy 1.15.2**

#### 2) 实现环境

平台: Visual Studio 2013

语言: C++

所需的库: **opencv 3.1.0**

### 2、解决思路

#### 1) 四角定位

- a) 用 **resize** 把图片像素缩小一半, 转为灰度图。
- b) 通过高斯滤波, 自适应阈值处理, 形态学操作来进一步去除噪点。
- c) **canny** 算子检测边缘, 多边形逼近 **approxPolyDP** 来过滤未闭合轮廓。
- d) 绘制所有闭合轮廓并加判断条件(相对于图像的面积及周长比例)得到四角点轮廓。
- e) 将四个角点排序。

#### 2) 扭曲矫正

- a) 利用排序好的四个角点来计算变换矩阵 **m\_transform**。
- b) 透视变换进行校正得到 **img\_undistort**。

#### 3) 阴影去除、图像增强

- a) **Range** 来截取增强区域
- b) 白平衡均衡色调
- c) 对数增强来提高亮度, 去除阴影
- d) 通过 **CMY** 色彩空间和 **HSV** 色彩空间来进行色彩增强

主体函数如下：

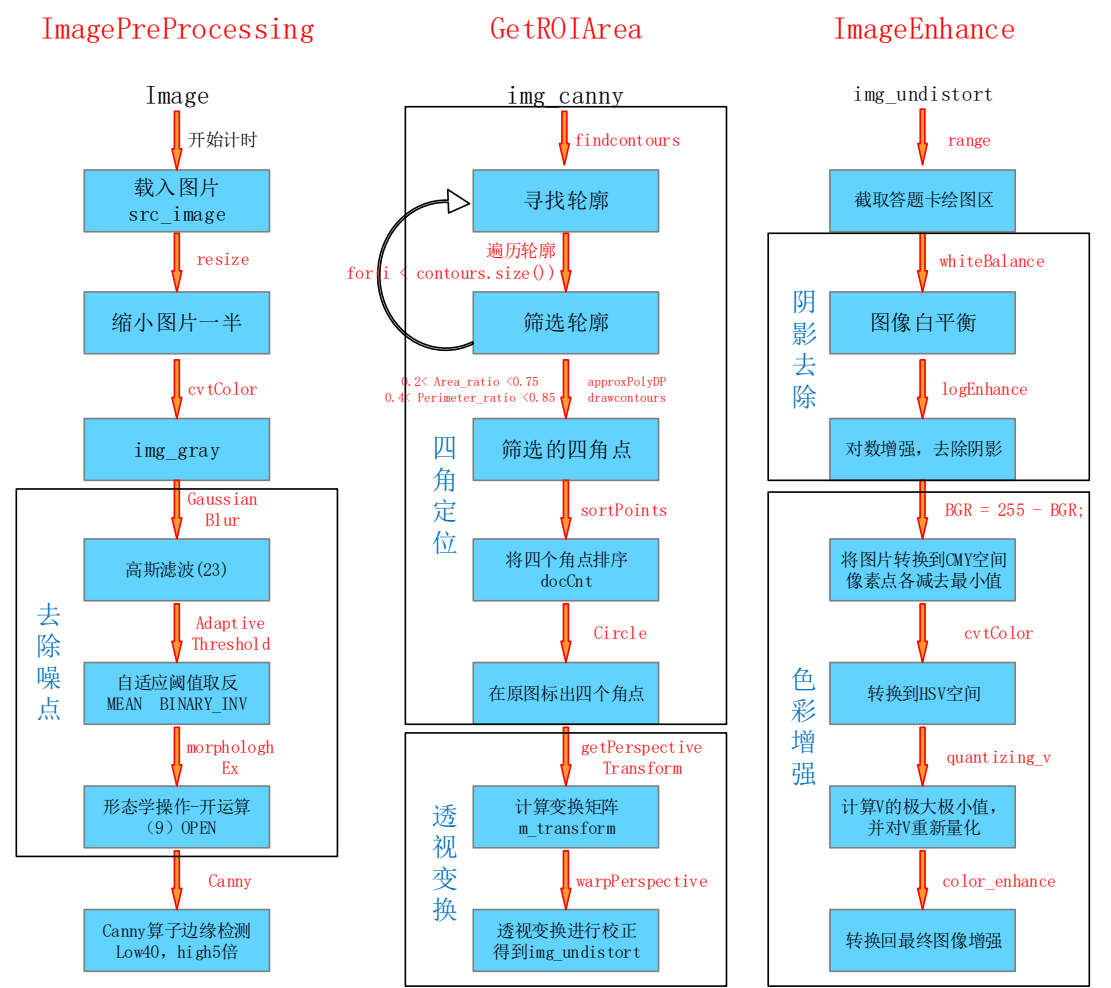
```
// Main Functions
Mat imagePreprocessing(Mat &img_src);           // Image Preprocessing
Mat getROIArea(Mat &src_image, Mat &img_canny); // Get ROI Area
Mat imageEnhance(Mat &img_undistort);           // Image Enhance
```

相关子函数如下：

```
// Sub Functions
vector<Point> sortPoints(vector<Point> &cnts); // Sort Points
Mat whiteBlance(Mat &img_src);                 // White Balance
Mat logEnhance(Mat &img_src);                   // Logarithm Enhance

// Color Enhance
void find_vhi_vlo(const Mat &img, double &vhi, double &vlo);
void quantizing_v(Mat &img, double vMax, double vMin);
void color_enhance(const Mat &img, Mat &dst, double vMax, double vMin);
```

整体流程框架图如下：



### 三、项目成果

#### 1、实验结果

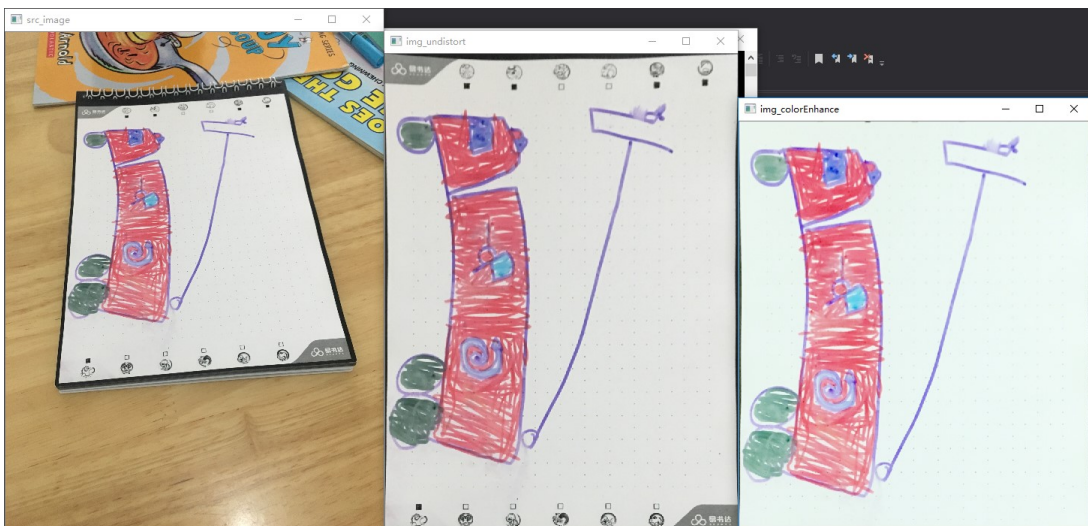
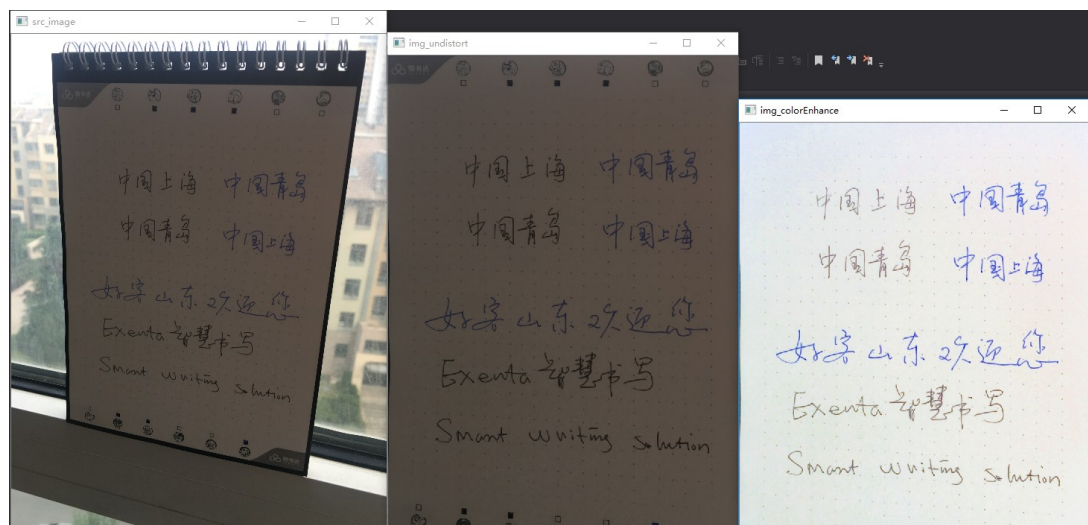
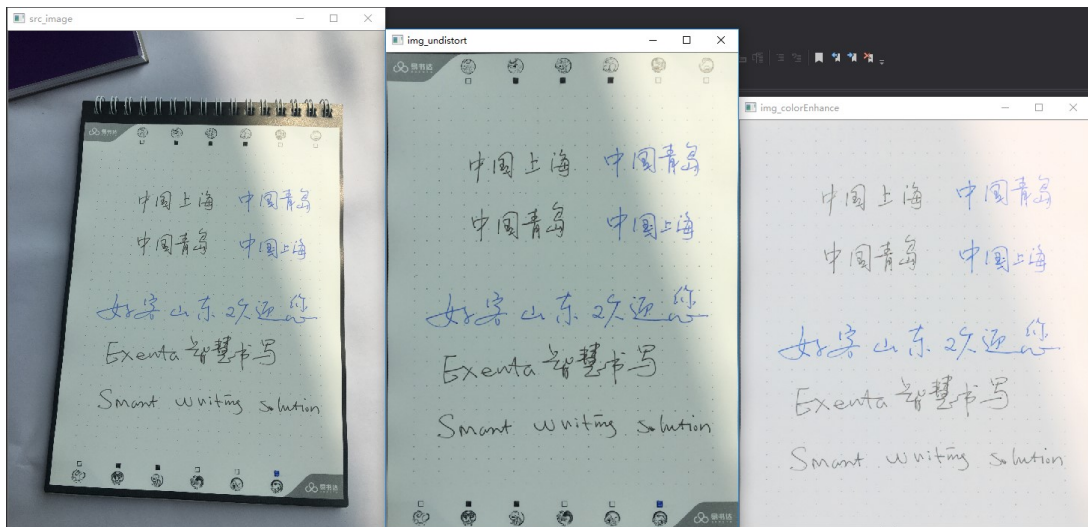
指标	效果
测试图像	184 张
正确识别	168 个
错误识别	16 个
准确率	91.3043%
总测试时长	5 分 35.5 秒（335.5 秒）
识别单张图片时长	1.8234 秒

#### 2、指标完成

- 1) 四角定位准确率超过 90%，基本达到要求；
- 2) 能较为有效的对阴影进行去除，对比度明显，达到要求；
- 3) 能对图像和文字进行色彩加强，基本达到要求。

#### 3、结果展示







## 四、总结

由于国庆来了还在忙工作面试的事，导致项目时间一开始较少。然后参考了其他同学组的思路（刘晟，王学艺，李俊凯）。最开始的预处理只有**默认内核值的高斯滤波**，**自适应阈值化处理**，然后使用了形态学操作的**闭运算**。然后第一遍遍历的时候，识别率大约在 **74%**，其中很多图片有误识别（识别到角点，但是有偏差，不是刚好答题卡四个顶点，或者误识别城堡的角点）。

后来开始考虑把高斯滤波转换为中值滤波，但是**中值滤波耗时比高斯滤波至少多一倍**，而且中值滤波对于光照条件影响很大的时候，容易把线条少的地方直接过滤到没有，在这个地方卡了大约 4 天左右。然后我仔细的分析了，**滤波**，**自适应阈值处理**，**形态学操作**三个步骤的顺序对预处理的影响。然后我通过实验，得到了先滤波，把图片模糊化处理，先处理细小的噪点；然后针对光线问题，用自适应阈值处理来获得答题卡边框轮廓；最后通过形态学操作的开运算把其他的稍微大一点的噪点去除。

阈值处理的时候，我发现答题卡区域的左上角和右下角的易书达标签对于边缘检测的影响很大，在轮廓提取的时候，很容易造成误识别。在这里面，我先将图片阈值化处理后**取反**，得到**黑底白色**的图片，然后开运算去除白色的小标签。我对比了下**黑底白色**的开运算，以及**白底黑色**的闭运算，发现**前者**的噪点去除效果更好。与此同时，在获得了四个角点之后，我发现储存的顺序不是固定的，会导致透视变换之后的坐标变换矩阵计算出现问题，于是重新写了个**四点排序**的子函数。在以上工作做完之后，再重新进行遍历检测的过程中，成功率就达到了 **85%**。然后针对滤波和形态学操作的卷积内核进行调参之后，成功率到了 **89%**，但是运行时间长的问题始终没有解决，一直都大于 4 秒多，而且最后光照影响很大的图片的识别始终效果不佳。

这时候我开始考虑先简单粗暴的把处理图片的尺寸减半（**resize**），然后发现识别率反而首次达到了 **90%**，而且运行时间也到了 2 秒左右。我仔细研究了识别出来的图片中，最后光照影响很大的图片的识别率反而上去了。我觉得可能是 **resize** 之后因为**插值**的原因，导致了原本因为光照有像素点丢失的反而能得到**恢复**。现在的高斯内核是 15 到 25 效果差别不大，同时开运算内核设定为 9（通过对比发现开运算内核过大时，对于边缘会有影响。），识别率稳定在 **91.3%**左右。

对于图像增强，一开始的阴影的原因，想着直接提高亮度，但是对于没有阴影的图片又会造成过度曝光。通过查找资料，我发现了对**数增强**的方法，这个方法对于对比度较低时，可以有很好的改善效果，同时对于对比度较高时，也不会过多的补偿，达到了**自适应光照补偿**的效果。色彩增强则是想到了转换颜色空间进行增强，找了相关的资料，因为时间的关系就没有过多的优化，引用成分较大。

这个月的项目实施中，感谢老师们对我们思路的各种启发；感谢队友卓然对于算法流程的思路，以及王野的后期和我各种商量具体识别率优化；感谢刘晟和王学艺两位同学跟我的耐心讲解和交流，让我获益良多。

## 五、 相关文档、及链接

对数增强

[https://blog.csdn.net/qq\\_29540745/article/details/74681853?locationNum=11&fps=1](https://blog.csdn.net/qq_29540745/article/details/74681853?locationNum=11&fps=1)

图像颜色增强算法，算法来源于 GIMP 中的 Color-enhance 插件，主要思路就是对 HSV 空间的 V 值进行重新量化。

<https://blog.csdn.net/hyqwmxsh/article/details/77980709>