



燕山大学课程实践项目报告

数据结构与算法课程实践 (二级项目)

ISBN 号识别系统的设计与开发

项目小组	软件 56 班第 1 组		指导教师	尤殿龙
小组成员	学号	姓名	项目成绩	所做工作
	201811040809	乔翱		字符识别部分代码和主函数的编写，报告、PPT 的编写和汇总，代码的汇总。
	201811040761	徐远新		字符识别部分代码的编写，报告以及 PPT 的编写。
	201811040802	贾浩然		字符切割部分代码的编写，报告以及 PPT 的编写。
	201811040794	王响		图片预处理部分代码的编写，报告以及 PPT 的编写。
开发日期	2020 年 2 月 24 日 - 2020 年 3 月 13 日			

摘要

[摘 要]

本文主要分为摘要、目录、前言、正文、结论五个部分，本文主要记录了本小组对于这次二级项目的设计方案以及心得体会等内容，重点阐述了本小组对于 ISBN 号识别整个项目的设计思路

数字识别目前已经广泛应用到日常生活中，而本项目是对 ISBN 号中的数字进行识别，典型的数字识别由读取图像、灰度化、二值化、字符分割、字符识别等几部分组成，这几步都存在密切的联系。本组经过讨论决定采用模板匹配法为主，划线法为辅来进行识别，并且为了达到一个比较好的识别效果，在预处理的二值化过程中对于每一张图计算出了全局阈值，使得预处理效果更好，在分割时做了优化避免了图像中黑点的干扰，提高了识别正确率和准确率。

[关键词] 字符识别；ISBN 号；数字识别；字符切割；模板匹配；划线。

目录

摘要.....	1
目录.....	2
前言.....	3
一、项目研究报告的目的.....	3
二、项目研究报告的范围.....	3
三、相关领域所做的工作以及相关领域研究的概况.....	3
四、研究报告的意图.....	4
五、预期的结果.....	4
六、项目组分工.....	4
正文.....	5
一、研究内容的基本原理.....	5
二、所采用的研究方法及相关工具.....	5
三、项目的方案设计.....	5
（一）、图片预处理.....	5
（二） 字符分割.....	8
（三）、字符识别.....	9
四、核心代码实现.....	12
（一）、预处理图片部分核心代码.....	12
（二）、字符切割部分核心代码.....	14
（三）、字符识别部分核心代码.....	17
（四）、主函数核心代码.....	19
五、项目测试.....	21
六、研究结果并讨论.....	22
结论.....	23
一、主要工作.....	23
二、主要结果.....	23
三、心得感受.....	23
四、下一步应当开展的主要工作.....	23
参考文献.....	25

前言

一、项目研究报告的目的

数字识别是目前计算机领域一个很重要的研究方向，在数字识别方面算法各种各样，各有各的优点和缺点，本小组在经过对本次 ISBN 识别项目的认真讨论并且思考分析后得出设计方案，最终决定采用模板匹配法进行识别，并且以划线法进行辅助。此报告展示了本小组对本次项目的研究成果，阐述了本组的设计思路，并且记录了本组在完成这次二级项目期间遇到的问题以及解决方案，最后对本次项目作出了一个总结以及记录了接下来的工作。

二、项目研究报告的范围

该项目的研究报告是基于我们小组对 ISBN 号的识别来进行撰写的，主要介绍了当前相关领域所作的工作和研究的概况，以及我们组成员对该项目的具体实现方法和具体的实现情况，以及我们组内的任务分配情况。

三、相关领域所做的工作以及相关领域研究的概况

数字识别在近几十年来一直是人们研究的热点和难点，国内外有很多算法实现数字识别，目前国内外对数字识别的主要研究都是提高识别的正确率和准确率以及提高识别的速度。国内外已研究比较多的匹配方法主要分为基于图像灰度的匹配方法和基于图像特征的匹配方法。

基于灰度相关的图像匹配方法，主要处理不存在旋转的匹配对象，且已经提出了各种各样针对灰度匹配的方法，例如：MAD 算法，序贯相似性检测法——SSDA，归一化积相关算法——NCC，这些方法在时间复杂度或匹配精度上均存在一定问题，不适用于实际的匹配应用。

在实际应用当中，一般以不规则匹配对象居多，且环境的光照变化也是非常不稳定因素，因此算法设计过程中必须考虑这几个因素。典型的还有在搜索策略上进行改进的一系列方法。由于基于灰度相关的图像匹配方法，不能解决匹配对象存在旋转情况下的匹配问题，所以继续研究基于特征的图像匹配方法，主要是针对匹配对象存在旋转情况下的匹配方法。常使用的特征包括边缘、区域、曲率、面积、线交叉点、质心等，其中以边缘和区域边界最常用，它们可由边缘检测方法和图像分割方法得到。

从上述分析可以看出，图像识别问题是一个研究的热点也是一个难点，并且这项技术还存在很多技术难题有待解决。识别的算法各种各样，但是每种算法都有自己的优点和缺点，还没有找到一劳永逸、通用的识别率高的识别算法。

四、研究报告的意图

项目研究报告的意图是阐述我们组对于本项目的分工，具体实现情况以及该领域目前研究的深度和广度和研究具体方向，以及未来关于相关领域的发展方向。

项目报告记录了本小组的研发过程，记录了设计方案以及对此项目的心得体会。

五、预期的结果

关于 ISBN 码识别这个项目，本组预期的结果是可以实现对于一个 ISBN 号的准确识别，首先对图片进行预处理，完后进一步对图片进行字符切割，最后实现字符识别，并且通过大量测试样本的测试，不断完善该系统，希望最后可以达到一个较高的识别正确率和准确率。最终实现的界面预期是读取一个文件夹中所有的图片进行识别，给出每一张 ISBN 图片识别的结果，并且最后统计总的识别正确率和准确率。

六、项目组分工

乔翱：字符识别部分代码的编写，报告以及 PPT 的编写。所有代码的汇总，报告的汇总以及 PPT 的汇总。

徐远新：字符识别部分代码的编写，报告以及 PPT 的编写。

贾浩然：字符分割部分代码的编写，报告以及 PPT 的编写。

王响：图片预处理部分代码的编写，报告以及 PPT 的编写。

正文

一、研究内容的基本原理

项目的工作是识别 ISBN 号，实际上考察的是对图像的处理，包括图像的裁剪，对图像像素的处理等一系列工作来达到实现对数字的成功识别。

数字识别的原理在于对其轮廓的分析和比较，我们在研究项目的时候选择的像素法实际就是对图片轮廓的研究，经过处理让它达到和模板尽量重合到达可以识别的效果。

主要先对图片进行预处理，预处理包括灰度化，去噪以及二值化，完后进行切割，最后识别，而最重要的是识别工作，我们主要采取模板匹配为主，划线法为辅的方法进行了识别。

二、所采用的研究方法及相关工具

本次二级项目采用小组分工、合作的方式进行，由于特殊原因，本次二级项目通过小组间线上的讨论、互帮互助来保证项目的顺利完成。

小组成员通过线上查阅资料以及借助教师推荐书籍制定制作方案，运用 VS、Opencv 工具完成项目开发。

三、项目的方案设计

经过小组成员的讨论，本组把此次项目大致分为三个部分：图片预处理，字符分割，字符识别。

（一）、图片预处理

图片预处理主要分为四步，分别是读取图像、图像灰度化、去噪处理以及图像二值化。

1、读取图像

把 24 位真彩色图像读入到程序中的 Mat 对象中，利用了 opencv 中的 `imread()` 方法。



图 3-1-1 24 位彩色真彩图

2、图像灰度化处理

为了后期方便对图像处理，以防颜色对系统的干扰，需要把 24 位真彩色图像转换为灰度图，利用 opencv 封装好的函数 `cvtColor()` 进行处理，把图像转为 8 位灰度图像。



图 3-1-2 8 位灰度图

3、去噪处理

由于图像中会有部分干扰识别的地方需要对图像进行去噪处理，去噪处理的方法有很多，例如：中值滤波、高斯滤波、均值滤波等等，经过分析比较效果，最终本小组采用了中值滤波进行去噪处理，这一步主要使用了 opencv 中的 `medianBlur` 函数。



图 3-1-3 去噪后的灰度图

4、图像二值化

Opencv 中自带了二值化函数 `threshold()`，也自带了，通过大津算法寻找局部阈值的方法，但是在识别过程中，本组发现有些图片的处理效果不好，于是又加了一个全局阈值的查找，利用全局阈值来对图像进行二值化。

将图像每一个位置的灰度值存入一个直方图数组，当遍历完整张图像后，不同灰度值的像素数都将记录在该数组中。在遍历过程中找到该图像的最大灰度值与最小灰度值，此时将最大灰度值与最小灰度值的二分之一作为一个初始阈值。此为全局阈值查找的第一步。第二步为进行精细的阈值查找。令“查找到的阈值与当前阈值之差小于 0.5”作为迭代器结束的条件。先计算小于当前阈值部分的平均灰度值，再计算大于当前阈值部分的平均灰度值，二者取平均值作为查找到的阈值，将其与当前阈值进行比较，看是否满足结束条件，若不满足，继续寻找，若满足，则该阈值为整张图像的全局阈值。



图 3-1-4 二值化后的图

(二) 字符分割

当图片进行预处理后，我们要把得到的二值化的图片进行切割，以便于后期对图片中的字符进行分割。

该过程包含水平投影和垂直投影两个部分。水平投影的目的在于将整张图片在水平方向上投影过去，由于经过二值化的图片只剩黑（0）和白（255）两种颜色（字符颜色默认为黑色），此时每一水平线上的黑色像素个数可知，以此绘制直方图，便可清晰的得到该图片在水平方向上的分块情况。选取最上方的黑色部分，此部分即为所要识别的 ISBN 号部分。同理，在垂直方向上进行投影，可得到每一竖直线上的黑色像素分布情况。经过垂直投影可以观察到黑色有明显的分块情况，每一个黑色块即代表了一个字符。经过两次投影，可以将所要求取的 ISBN 号的字符分割开来。

由于图像上方可能有黑点或者黑边会影响图片的切割，所有做了一个优化，判断水平投影最上面的长度，如果长度很小说明是黑点或者黑边就不进行切割。



图 3-2-1 水平投影示意图

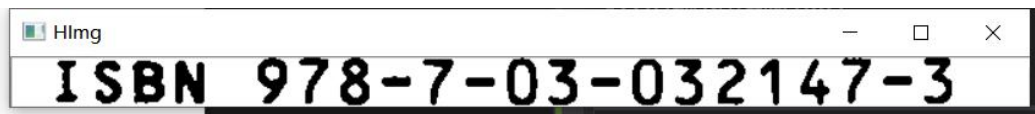


图 3-2-2 水平投影切割后的结果

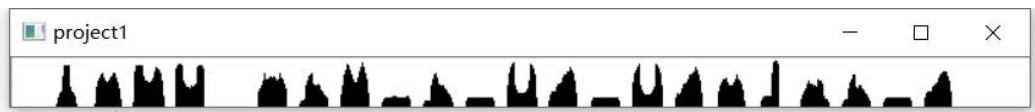


图 3-2-3 垂直投影示意图



图 3-2-4 垂直投影切割下的数字示意图

（三）、字符识别

1、模板制作

由于我们主要采取的是模板匹配的方法，所有字符识别的第一步就是制作模板。

在项目进行初期，最开始的想法是将字符切割出来的数字直接作为模板来使用，然后使用像素差法进行对比，但是在提取一套数字模板出来之后，识别其他图片的数字却正确率很低，不同图片截取出来的数字不同会造成误差，也就导致了这种方法被放弃的结果。

之后，我们在官网上找到了标准的 ISBN 号的数字字体（见图 3-3-1），把他们截取出来经过一系列处理，然后作为新的模板来使用。

0 1 2 3 4 5 6 7 8 9 X

图 3-3-1 ISBN 标准数字

在提取出模板之后，识别效果好了很多，然后我们在测试的时候发现部分图片出现了不同版本的 ISBN 号字体，见图 3-3-2 和图 3-3-3 。



图 3-3-2 ISBN 号版本



图 3-3-3 ISBN 号不同版本

后面的版本更接近于阿拉伯数字，对此我们又针对阿拉伯数字又设计了一套模板，部分示例模板如下图（图 3-3-4）。



图 3-3-4 部分数字模板（包含阿拉伯数字和标准 ISBN 号版本对比）

2、外接矩形的提取

为了实现像素差法识别的方案，我们发现在模板和截取图像的图片上存在了白边的情况，这样会对识别结果产生较大的影响，对此我们决定进行去除图像边缘白色部分，这时候就要用到提取外接矩形的操作。

提取方法，对于一个已经截取好的数字图像，分别找到能够包含数字的在水平和垂直方向上的边界，然后使用矩形切割截取最小外接矩形。

核心函数：`Mat A=test(Range(xmin,xmax),Range(ymin,ymax))`；其中，A 为截取后的图像，test 作为待截取的图像，Range 中的四个参数分别代表提取边界。

以提取其中一个边界代码进行说明，

传入预处理的图像 src，其中 rows 和 cols 分别为图像的长度和宽度，从 (0, 0) 处开始遍历寻找黑色像素点。

正常情况下的像素提取是彩色图的提取，提取函数如下：

`at<Vec3b>(i,j)`；Vec3b 代表的是三个 unsigned char 合成的向量，我们在这里处理的是灰度图，只存在一个通道的情况下，提取像素的函数如下：`at<unsigned`

```
char>(i, j);
```

只要在一个方向上提取到最小的 i ，就可以赋值给 $xmin$ ，最后 `return xmin` 即可。

这样，找到四个边界之后，就可以将图像的白色边缘裁剪掉，结果见下图（图 3-3-5）。



图 3-3-5 对比图（左：处理前 右：处理后）

3、模板匹配法识别以及划线法识别

对于识别，我们采用了像素差法和划线法相结合，其中像素差法为主，划线法为辅。

像素差法：就是将测试图片和模板统一成相同尺寸之后，对于每一个像素进行作差并绝对值处理，然后将各像素点的差值求和，这样处理之后差值最小的图片就是最符合的结果。

首先将切割好的数字进行去边框处理，然后统一尺寸（这里我们将尺寸定义为 24×36 ），然后再次进行二值化处理，因为 `resize` 函数会改变图像的像素。

核心部分，将模板和测试图片传入计算函数，由于先统一了尺寸，直接使用其中一个图片的长度和宽度即可，然后计算差值，保证差值为整数，之后求和返回 `num` 即可。

定义一个 `node` 结构体，其中有两个元素，一个是 `cha` 代表像素差，一个是 `index` 代表编号，将测试数据与图片进行像素差计算之后存储在 `cha` 中，`index` 是输出字符的作用，计算完所有模板之后，进行排序，输出最符合的图像的 `index`；

划线法：在指定像素位置划线，统计线与图线的交点然后作为模板匹配，在我们这个系统里面是作为区分 3 和 8 的工具，由于裁剪和二值化都会产生一定的误差，于是我们选择划线法进行区分数字。

其中，我们统计的是高度 $2/3$ 处的像素交点个数，首先对图像进行去除边缘操作、统一尺寸和二值化处理，然后 `for` 循环统计相交的个数，返回统计值。

四、核心代码实现

（一）、预处理图片部分核心代码

预处理包括读取图片、灰度化，中值滤波去噪、二值化，核心代码如下：

```
Mat src = imread(files[rd]);
    Mat dst;
    //图片预处理，灰度化-去噪（中值滤波）-二值化（找局部阈值）
    cvtColor(src, src, COLOR_RGB2GRAY);
    medianBlur(src, src, 3);
    int zz = DetectThreshold(src); //计算全局阈值（部分图用全局阈值效果不好，部分图用局部阈值效果好）
    threshold(src, dst, 0, 255, CV_THRESH_OTSU);
```

寻找全局阈值的核心代码如下：

```
int DetectThreshold(Mat& src)
{
    uchar iThreshold; //阈值
    try
    {
        int height = src.cols;
        int width = src.rows;
        int step0 = src.step / sizeof(uchar);
        uchar *data0 = (uchar*)src.data;
        //cout << step0;
        int iDiffRec = 0;
        int F[256] = { 0 }; //直方图数组
        int iTotalGray = 0; //灰度值和
        int iTotalPixel = 0; //像素数和
        uchar bt; //某点的像素值
        uchar iNewThreshold; //新阈值
        uchar iMaxGrayValue = 0, iMinGrayValue = 255; //原图像中的最大灰度值和最小灰度值
        uchar iMeanGrayValue1, iMeanGrayValue2;
        //获取(i,j)的值，存于直方图数组F
        for (int i = 0; i < width; i++)
```

```

{
    for (int j = 0; j < height; j++)
    {
        bt = data0[i*step0 + j];
        if (bt < iMinGrayValue)
            iMinGrayValue = bt;
        if (bt > iMaxGrayValue)
            iMaxGrayValue = bt;
        F[bt]++;
    }
}
iThreshold = 0;
iNewThreshold = (iMinGrayValue + iMaxGrayValue) / 2; //初
始阈值
iDiffRec = iMaxGrayValue - iMinGrayValue;
for (int a = 0; (abs(iThreshold - iNewThreshold) > 0.5); a++) //
迭代中止条件
{
    iThreshold = iNewThreshold;
    //小于当前阈值部分的平均灰度值
    for (int i = iMinGrayValue; i < iThreshold; i++)
    {
        iTotGray += F[i] * i; //F[] 存储图像信息
        iTotPixel += F[i];
    }
    iMeanGrayValue1 = (uchar)(iTotGray / iTotPixel);
    //大于当前阈值部分的平均灰度值
    iTotPixel = 0;
    iTotGray = 0;
    for (int j = iThreshold + 1; j < iMaxGrayValue; j++)
    {
        iTotGray += F[j] * j; //F[] 存储图像信息
        iTotPixel += F[j];
    }
    iMeanGrayValue2 = (uchar)(iTotGray / iTotPixel);

    iNewThreshold = (iMeanGrayValue2 + iMeanGrayValue1) / 2;
    //新阈值

```

```

        iDiffRec = abs(iMeanGrayValue2 - iMeanGrayValue1);
    }
}
catch (cv::Exception e)
{
}

return iThreshold;
}

```

(二)、字符切割部分核心代码

字符切割分两部分进行，首先是通过水平投影切割出上面那一块，水平投影核心代码如下：

```

int Horizontal_projection(const Mat& src, Mat& roiImg)
{
    //step1. 计算水平投影黑色点数量
    int w = src.cols; //图形宽
    int h = src.rows; //图形长
    vector<int> project_val_array; //记录每行黑色数量
    int per_pixel_value;
    for (int j = 0; j < h; j++) //行
    {

        Mat j_im = src.row(j); //取第 j 行作为一个图像
        int num = w - countNonZero(j_im); //每行黑色点数量

        project_val_array.push_back(num);
    }
    //显示
    if (1)
    {
        Mat hist_im(h, w, CV_8UC1, Scalar(255));
        for (int i = 0; i < h; i++)
        {
            for (int j = 0; j < project_val_array[i]; j++)

```

```

        {

            hist_im.ptr<unsigned char>(i)[h - 1 - j] = 0;

        }

    }

    //imshow("project", hist_im);
}

//step2. 字符分割
int startIndex = 0;
int endIndex = 0;
bool inBlock = false; //是否遍历到了字符区内
for (int i = 5; i < h; ++i)
{
    if (!inBlock && project_val_array[i] != 0 &&
project_val_array[i] > 10) //进入字符区了
    {
        inBlock = true;
        startIndex = i;
        //cout << "startIndex is " << startIndex << endl;
    }
    else if (project_val_array[i] < 10 && inBlock) //进入空白区
了
    {
        endIndex = i;
        inBlock = false;
    }
    If ((endIndex - startIndex) > 0.03 * h)
        {roiImg = src(Rect(0, startIndex, w, endIndex + 1 -
startIndex));
        //imshow("HImg", roiImg);
        break;
        }
}

return 0;
}

```

水平切割完后需要得到每一个数字就要垂直切割，垂直切割核心代码如下：


```

int vertical_projection(const Mat& src, vector<Mat>& roiList)
{
    //step1. 计算竖直投影黑色点数量
    int w = src.cols; //图形宽
    int h = src.rows; //图形长
    vector<int> project_val_array; //记录每列黑色数量
    int per_pixel_value;

    for (int j = 0; j < w; j++) //列
    {

        Mat j_im = src.col(j); //取第 j 列作为一个图像
        int num = h - countNonZero(j_im); //每列黑色点数量
        project_val_array.push_back(num);
    }
    //显示
    if (1)
    {
        Mat hist_im(h, w, CV_8UC1, Scalar(255));
        for (int i = 0; i < w; i++)
        {
            for (int j = 0; j < project_val_array[i]; j++)
            {
                hist_im.ptr<unsigned char>(h - 1 - j)[i] = 0;
            }
        }
        //imshow("project1", hist_im);
    }

    //step2. 字符分割

    int startIndex = 0;
    int endIndex = 0;
    bool inBlock = false; //是否遍历到了字符区内
    int num = 0;

```

```

for (int i = 20; i < w; ++i)

{

    if (!inBlock && project_val_array[i] != 0) //进入字符区了

    {

        inBlock = true;
        startIndex = i;
        //cout << "startIndex is " << startIndex << endl;

    }

    else if (project_val_array[i] == 0 && inBlock) //进入空白区了

    {

        endIndex = i;

        inBlock = false;
        if (num < 21)
        {
            Mat roiImg = src(Rect(startIndex, 0, endIndex + 1
- startIndex, h));

            roiList.push_back(roiImg);
            num++;

        }

    }

}

return 0;
}

```

（三）、字符识别部分核心代码

求像素差的核心代码如下：

```

int PixelDifference(Mat src1, Mat src2)
{
    int num = 0;
    for (int i = 0; i < src1.rows; i++)
    {
        for (int j = 0; j < src1.cols; j++)
        {
            if (src1.at<unsigned char>(i, j) >= src2.at<unsigned char>(i, j))
            {
                num += src1.at<unsigned char>(i, j) - src2.at<unsigned
char>(i, j);
            }
            else
            {
                num += src2.at<unsigned char>(i, j) - src1.at<unsigned
char>(i, j);
            }
        }
    }
    return num;
}

```

由于切割下来的字符可能边缘有空白会影响识别效果，所以要先去除切割字符的白色边缘也就是找最小外接矩形，核心代码如下：

```

int ok=0;
int xmin = getxmin(srctest);
int xmax = getxmax(srctest);
int ymin = getymin(srctest);
int ymax = getymax(srctest);
Mat A = srctest(Range(xmin, xmax + 1), Range(ymin, ymax)); //
获取外接矩形去除多余白边
int getxmin(Mat& src)
{
    int xmin = src.rows;
    for (int j = 0; j < src.cols; j++)
    {
        for (int i = 0; i < src.rows; i++)
        {

```

```

        if (src.at<unsigned char>(i, j) == 0)
        {
            if (i <= xmin)
            {
                xmin = i;
            }
        }
    }
    return xmin;
};

```

与模板数字进行匹配的核心代码如下：

```

Mat src0 = imread("0.jpg", 0);
if (src0.empty()) {
    printf("could not load image...\n");
    return -1;
}
resize(src0, src0, Size(24, 36));
threshold(src0, src0, 0, 255, CV_THRESH_BINARY |
CV_THRESH_OTSU);
node a_0;
a_0.cha = PixelDifference(A, src0);
a_0.index = '0';
difference.push_back(a_0);
//对匹配结果进行排序，返回像素差值最小的数字
sort(difference.begin(), difference.end(), com);
return difference[0].index;

```

（四）、主函数核心代码

主函数首先是对文件夹中所有的图片以及图片的名字进行了一个读取，核心代码如下：

```

//读取所有测试数据（读取一个文件中所有的 jpg 文件）
string filePath = "C:\\Users\\qa\\Desktop\\二级项目\\2018 级课程实践--项目测试数据集"; //正样本路径
//string filePath = "C:\\Users\\qa\\Desktop\\二级项目

```

```

\\ceshi"; //正样本路径
vector<string> files;
string file_format = "jpg";
getFiles(filePath, file_format, files);

//读取所有的文件的名字
vector<String> tem;
string path = "C:/Users/qa/Desktop/二级项目/2018 级课程实践--项目测试数据集/";
//string path = "C:/Users/qa/Desktop/二级项目/ceshi/";
cv::glob(path + "*.jpg", tem);
int number = files.size();//文件数量

```

主函数中主要的调用识别函数的核心代码如下：

```

//遍历切割下来的数字
for (int i = 4; i < k1; i++)
{
    //判断是否该字符是不是-
    for (int y = 0; y < gang.size(); y++)
    {
        if (i-4==gang[y])
        {
            test += '-';
            i++;
            break;
        }
    }

    char c;
    try { c = recognize(char_im_vec[i]);
    if (c == '3') {
        int p = numberx(char_im_vec[i]);
        if (p <= 9) { c = '3'; }
        else { c = '8'; }
    }
    test += c;
}
catch (exception e) { i++; }

```

```
//因为 8 与 3 容易识别混淆，加一个划线区分 8 与 3
```

```
}
```

统计正确识别的 ISBN 号，以及统计正确识别的字符，核心代码如下：

```
//统计正确识别字符数
for (int z = 0; z < correct.size(); z++)
{
    if (z >= test.size())
        break;
    if (correct[z] == '-' || correct[z] == 'X')
        continue;
    if (correct[z] == test[z])
        num_2++;
}
for (int z = 0; z < correct.size(); z++)
{
    if (correct[z] != '-')
        num_1++;
}
printf("当前累计识别正确个数: %d", num);
```

计算正确率以及准确率核心代码如下：

```
//输出正确率以及准确率
double accuracy = (double)num / (double)tem.size();
double precision_rate = (double) num_2 / (double) num_1;
cout << "-----" << endl;
printf("总共%d 个测试样本，识别正确个数为%d\n", tem.size(), num);
printf("识别正确率为: %.3f\n", accuracy);
printf("识别准确率为: %.3f\n", precision_rate);
```

五、项目测试

对于本次项目测试，我们通过读取一个文件中的所有 jpg 图片来进行对我们制作的系统的测试，总共有八十张测试图片，我们对这八十张图片进行一个识别，统计了识别的正确率和准确率。测试运行截图如下：

```
识别正确 当前累计识别正确个数: 33
待识别样本: 978-7-115-27946-0 识别结果: 978-7-115-27946-0 ✓
识别正确 当前累计识别正确个数: 34
待识别样本: 978-7-115-32840-3 识别结果: 978-7-115-32840-3 ✓
识别正确 当前累计识别正确个数: 35
待识别样本: 978-7-115-36045-8 识别结果: 978-7-115-36047-8 ×
识别错误 当前累计识别正确个数: 35
待识别样本: 978-7-115-42951-3 识别结果: 978-7-115-42951-3 ✓
识别正确 当前累计识别正确个数: 36
待识别样本: 978-7-115-43978-9 识别结果: 978-7-115-43978-9 ✓
识别正确 当前累计识别正确个数: 37
待识别样本: 978-7-115-44154-6 识别结果: 978-7-115-44154-6 ✓
识别正确 当前累计识别正确个数: 38
```

图 5-1-1 代码运行示意图

```
识别正确 当前累计识别正确个数: 38
-----
总共80个测试样本，识别正确个数为77
识别正确率为: 0.963
识别准确率为: 0.995
```

图 5-1-2 运行界面识别准确率和正确率示意图

六、研究结果并讨论

对于本次二级项目 ISBN 号的识别，我们开发出的系统能以较高的正确率和准确率识别出来，通过对图像的预处理、字符切割、字符识别完成了这么一个 ISBN 识别系统，从中学会了对图像的一系列操作，对于大多数 ISBN 号我们可以准确识别，我们的系统有较高的识别正确率和准确率。

经过小组内的讨论，我们最终一致认为模板匹配法有较好的识别率，但是对于一些特殊的图片识别效果还是很差，此系统还有很大的提升空间，这是接下来我们需要改进的。

结论

一、主要工作

在大致了解了项目的背景之后，小组举行了第一次会议，组长进行了项目简要介绍，并完成了项目的分工。之后，小组成员按照项目要求进行各自的项目设计，并在设计期间相互沟通，提出问题并解决问题，之后在项目各部分大致完成的情况下，进行初步的项目结合，进行数据测试。对于发现的问题，小组成员积极思考讨论并对项目进行进一步优化以促成项目的完善，提高识别正确率与准确率。

二、主要结果

对于测试数据来说可以基本实现 ISBN 的识别，能够在不识别字母和特殊情况的前提下，一步步提高识别的正确率，并且能够输出识别结果，与测试数据的对比一目了然，最后输出正确率和准确率来检验识别的成果。

三、心得感受

此次二级项目，识别 ISBN 号中的数字，需要利用到 opencv，这对于我们来说是一个全新的没有接触过的领域，极大的考验我们的自学能力，在整个项目制作过程中，我们也遇到了各种各样的困难，但是小组成员没有放弃，遇到困难迎难而上，一点点去克服困难。

此次二级项目是小组合作完成的项目，这就需要每一个小组成员团结一致，沟通合作，虽然此次二级项目由于特殊原因都是在家完成的，但是整个过程小组成员也多次进行了线上会议，不断去沟通讨论项目的设计方案。

在项目进行的过程中，每个人都展现了对项目的责任感和设计项目的积极性。

最后完成这个项目后，每个人都有很大的自豪感，通过自己的努力完成一件事情，证明了自己的能力。

四、下一步应当开展的主要工作

对于此次二级项目下一步要开展的工作我们小组进行了一个讨论，最主要的工作是提高识别 ISBN 码的正确率和准确率，在做这个项目的过程中，我们也试验了很多方法，力争寻求识别正确率和准确率最高的方法，但是每一种方法都还是有一定的局限性的，在接下来我们还要对这套系统进行改进，提高识别的正确率和准确率。

接下来还需要对系统改善的是对于一些特殊图片的识别，例如倾斜的 ISBN 号或者是模糊的 ISBN 号，由于时间的原因，这部分我们的处理还不是特别完善，这部分也是下一步开展的工作中主要要完善的。

最后，我们小组也希望可以把这套系统的界面改善改善，由于控制台的界面限制，始终对于用户的使用不友好，我们计划可以用其他的界面设计软化设计一个界面，把我们的系统移植过去，整体封装成一个界面友好的软件。

下一步开展的工作还有很多，一个软件特别重要的一部分就是完善，全部工作设计完成后，我们计划去收集使用用户的反馈，完后再进一步不断完善我们的系统。

项目小组	软件 56 班第 1 组			
小组成员	学号	姓名	自评成绩	所做工作
	201811040809	乔翱	10 分	字符识别部分代码和主函数的编写，报告、PPT 的编写和汇总，代码的汇总。
	201811040761	徐远新	9 分	字符识别部分代码的编写，报告以及 PPT 的编写。
	201811040802	贾浩然	9 分	字符切割部分代码的编写，报告以及 PPT 的编写。
	201811040794	王响	8 分	图片预处理部分代码的编写，报告以及 PPT 的编写。

参考文献

- [1]程杰, 大话数据结构[M], 2011, 6。
- [2]Sartaj Sahni, 数据结构、算法与应用[M], 2019, 5。
- [3] 毛星云, OpenCV3 编程入门[M], 电子工业出版社
- [4] (德)斯蒂格(Steger, C.), (德)尤里奇(Ulrich)著, 杨少荣译, 机器视觉算法与应用, 清华大学出版社, 2016 年, P346-377, “光学字符识别”: 字符分割-特征提取-字符分类。

其中: M 表示书; J 表示期刊; C 表示会议; D 表示博士