产品使用手册

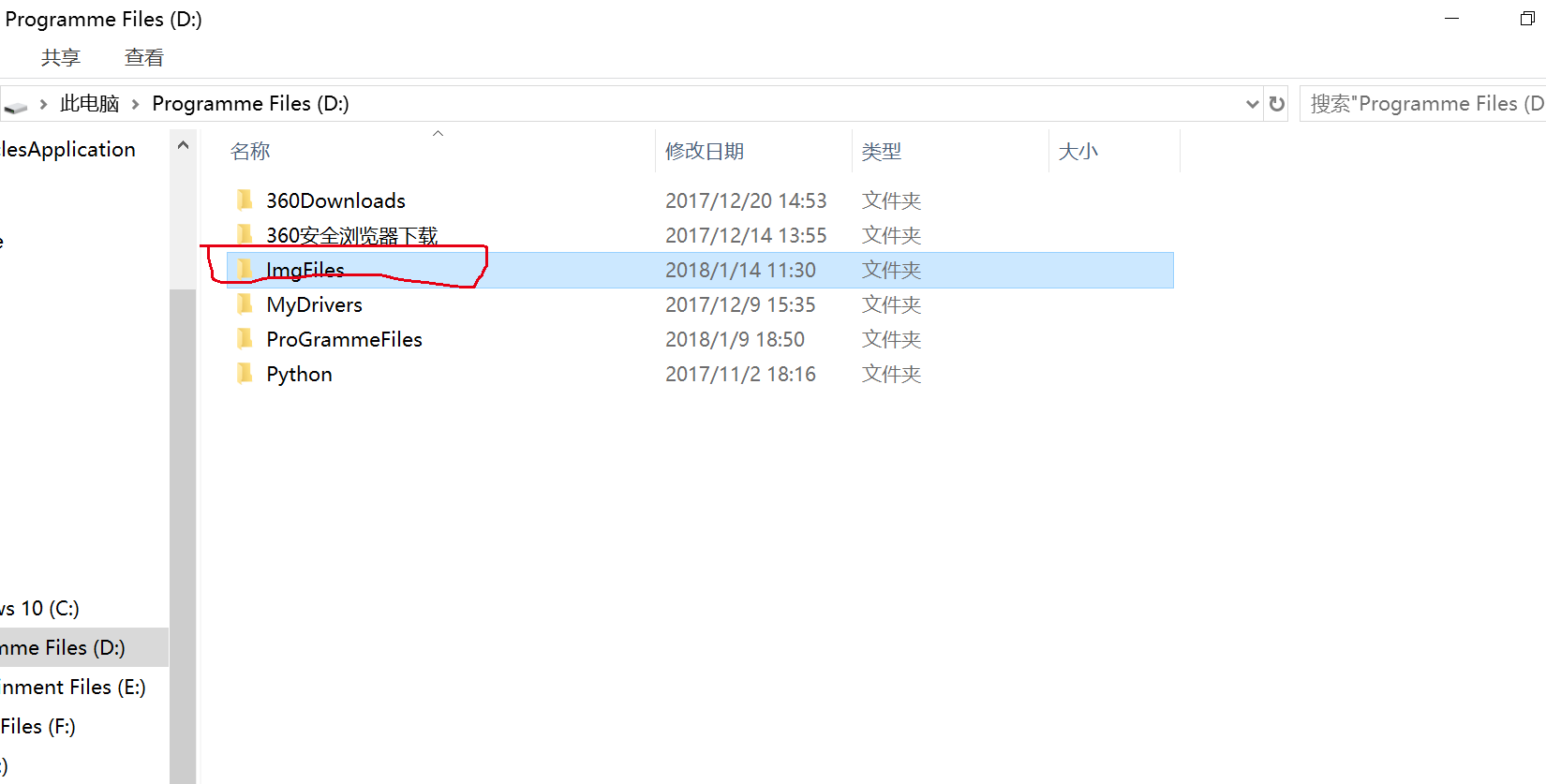
2018年1月14日星期日

齐勇

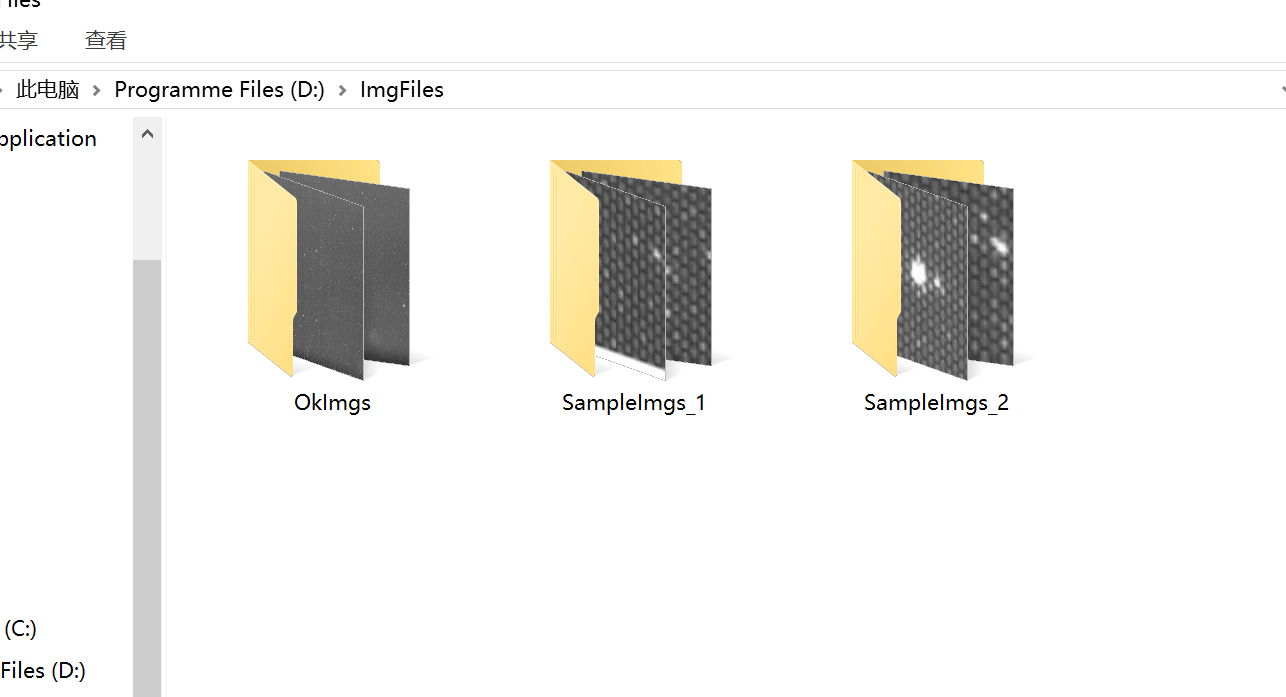
该界面处理的图片均为.jpg格式

## 一 关于界面的使用

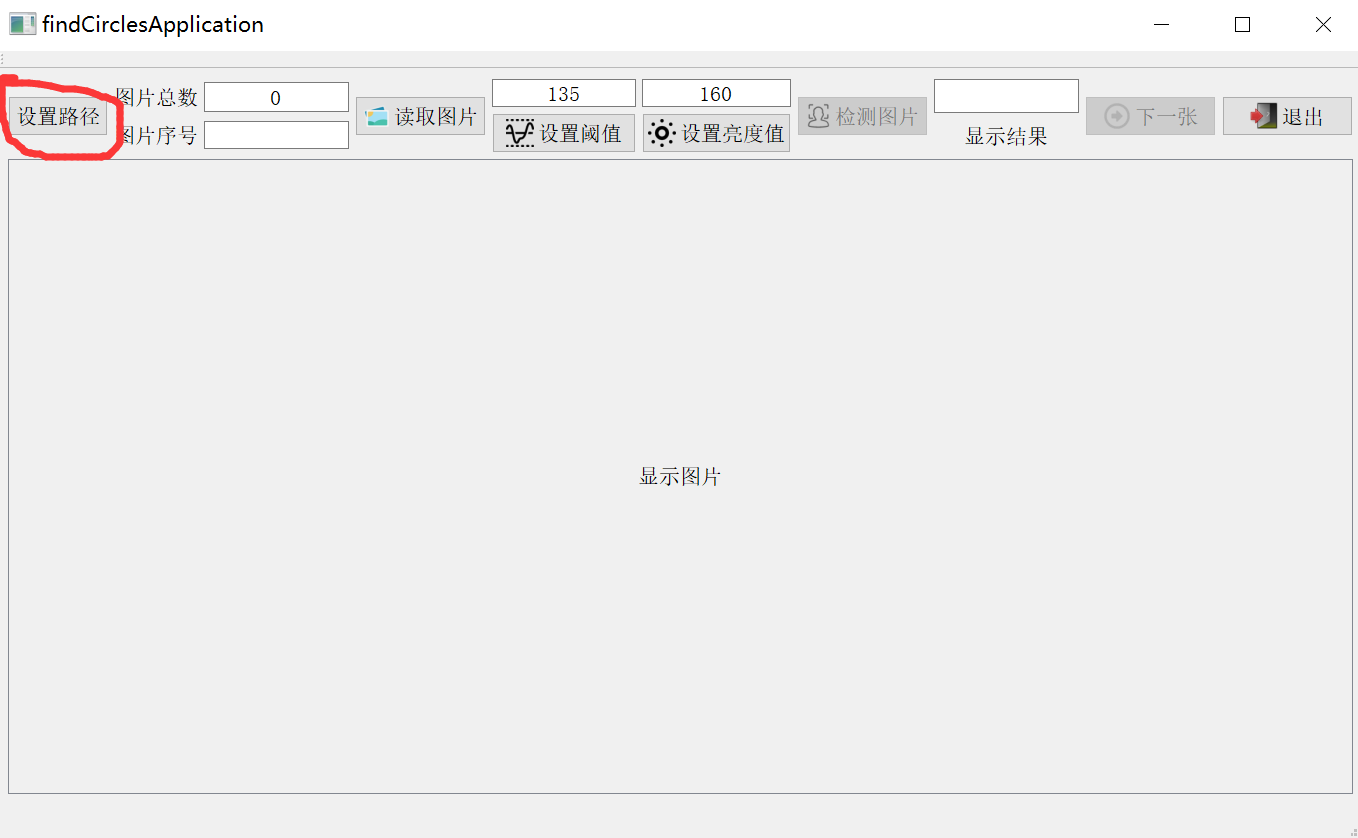
1. **在测试之前，先在D盘下新建一个文件夹:”ImgFiles” 这一步不可缺少，如下图：**

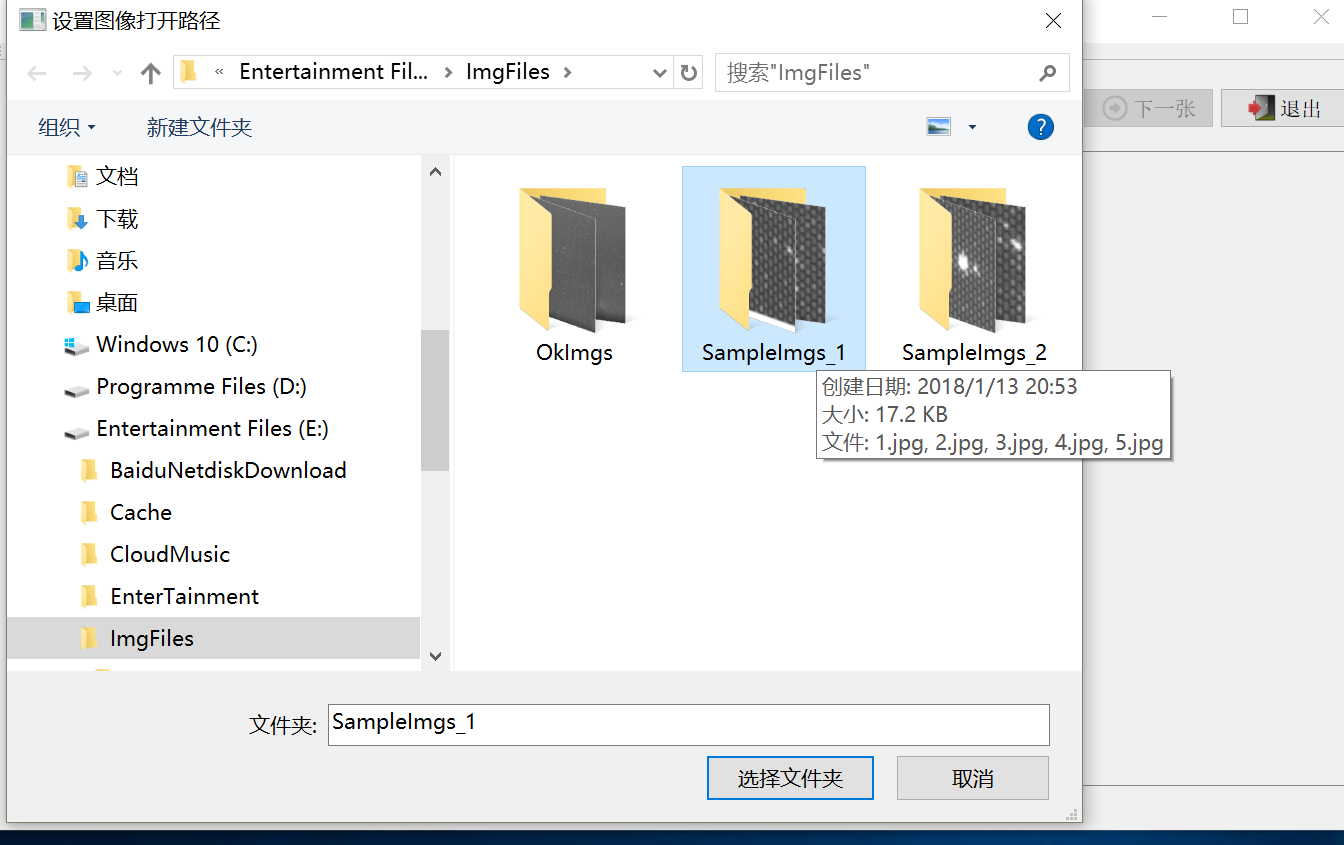


1. 之后可在“ImgFiles”下新建任意多个文件夹，里面存放待测图片，如下图：



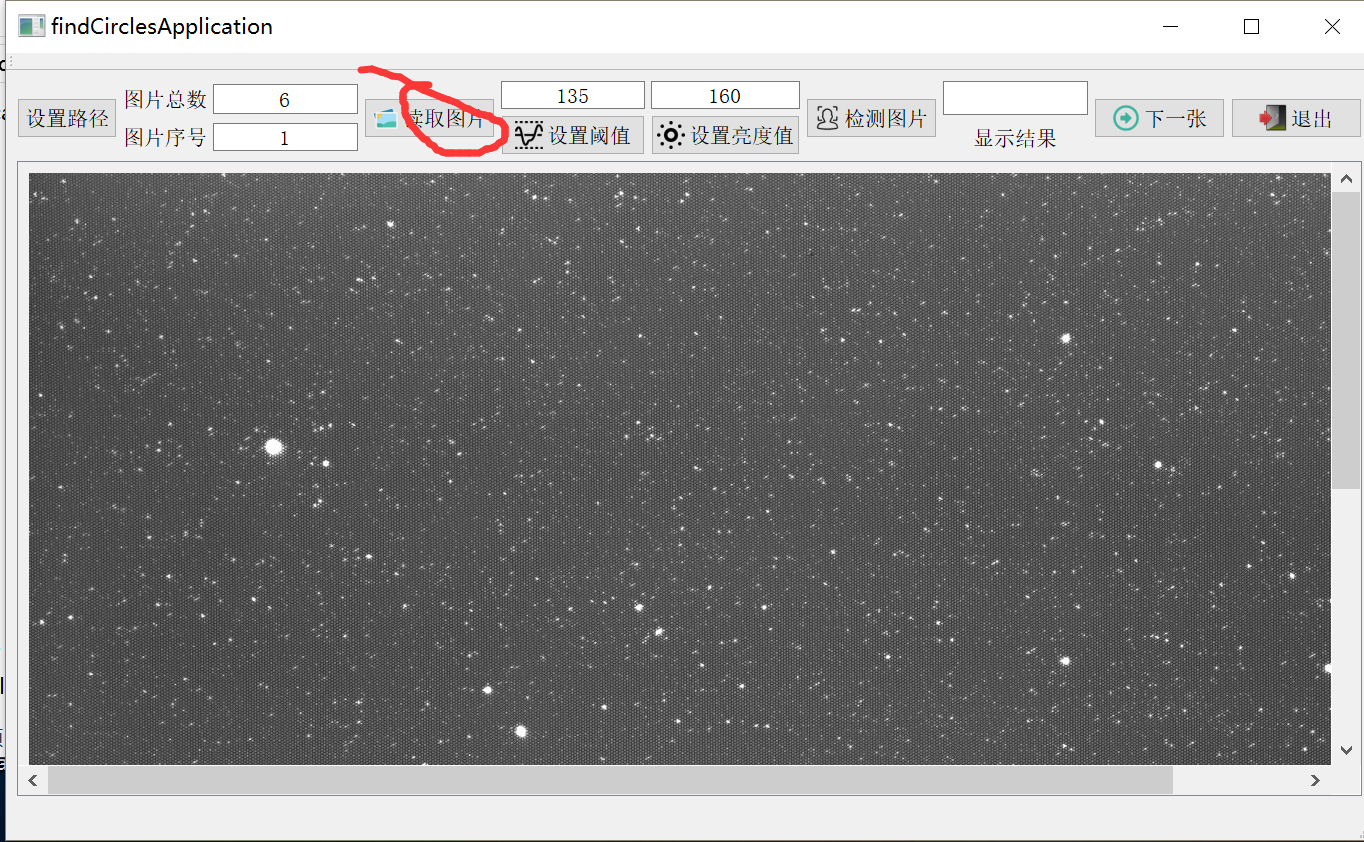
1. 点击界面中“设置路径”按钮，进行选择准备测试的图片文件





此时，界面中的“图片总数”一栏，会显示该文件夹下的所有图片数量；

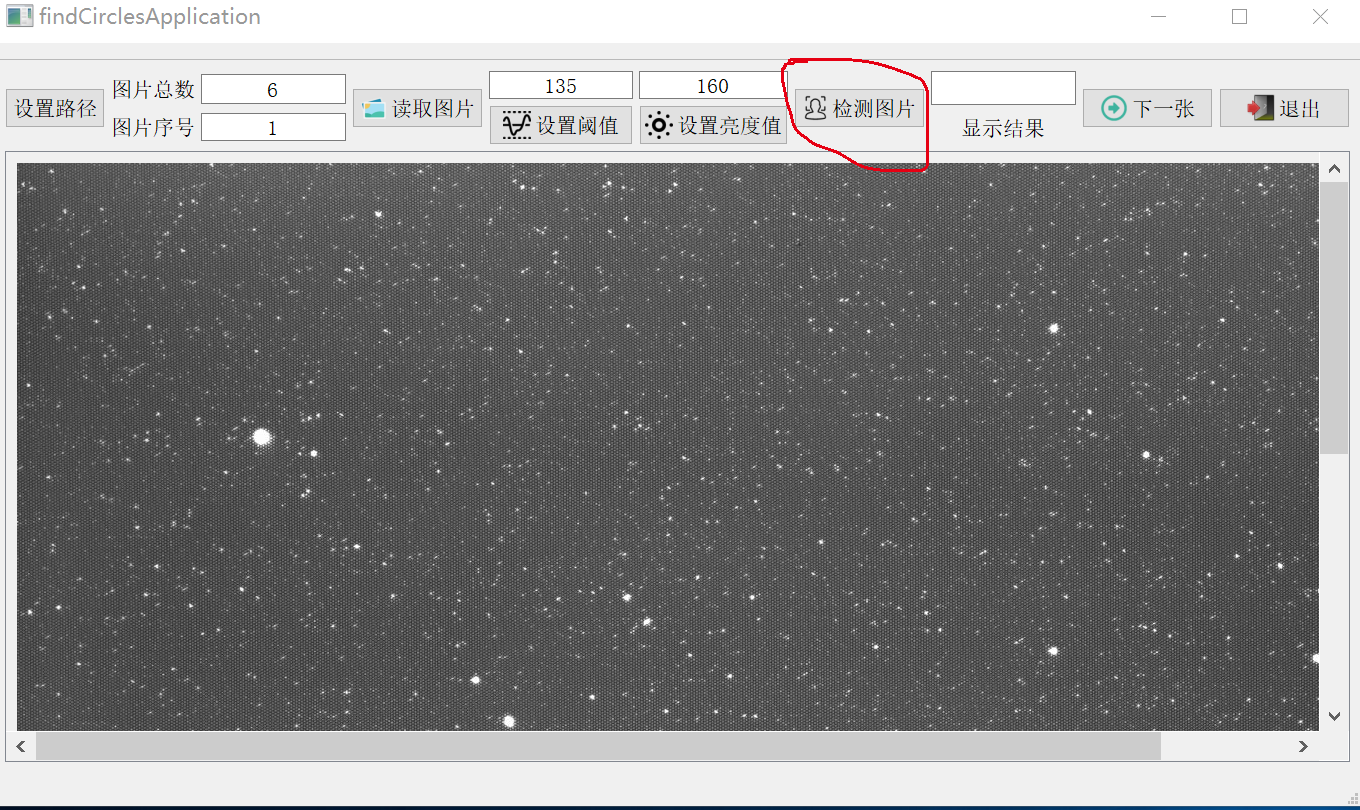
1. 点击界面中“读取图片”,此时会显示当前图片的序号；

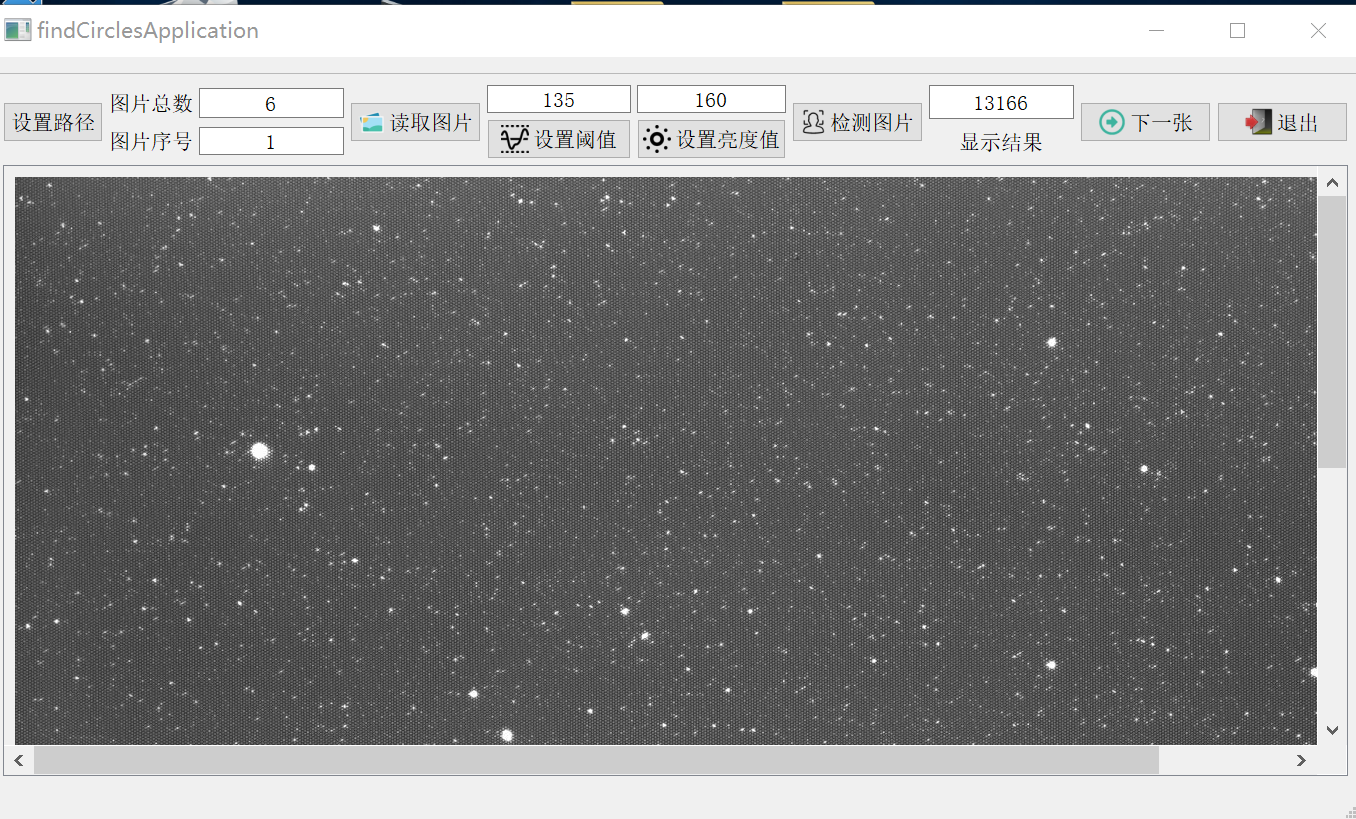


1. 界面中“设置阈值”与“设置亮度值”,已有默认值，如果需要重新设置，请在每次修改完数值之后，点击下面对应的按钮。比如：如果要修改“设置阈值”处对应的数值，在输入数字之后，请点击一下“设置阈值按钮” 切记。

相似的，对于“设置亮度”，如欲修改其值，请在输入值之后，请点击“设置亮度值”按钮。

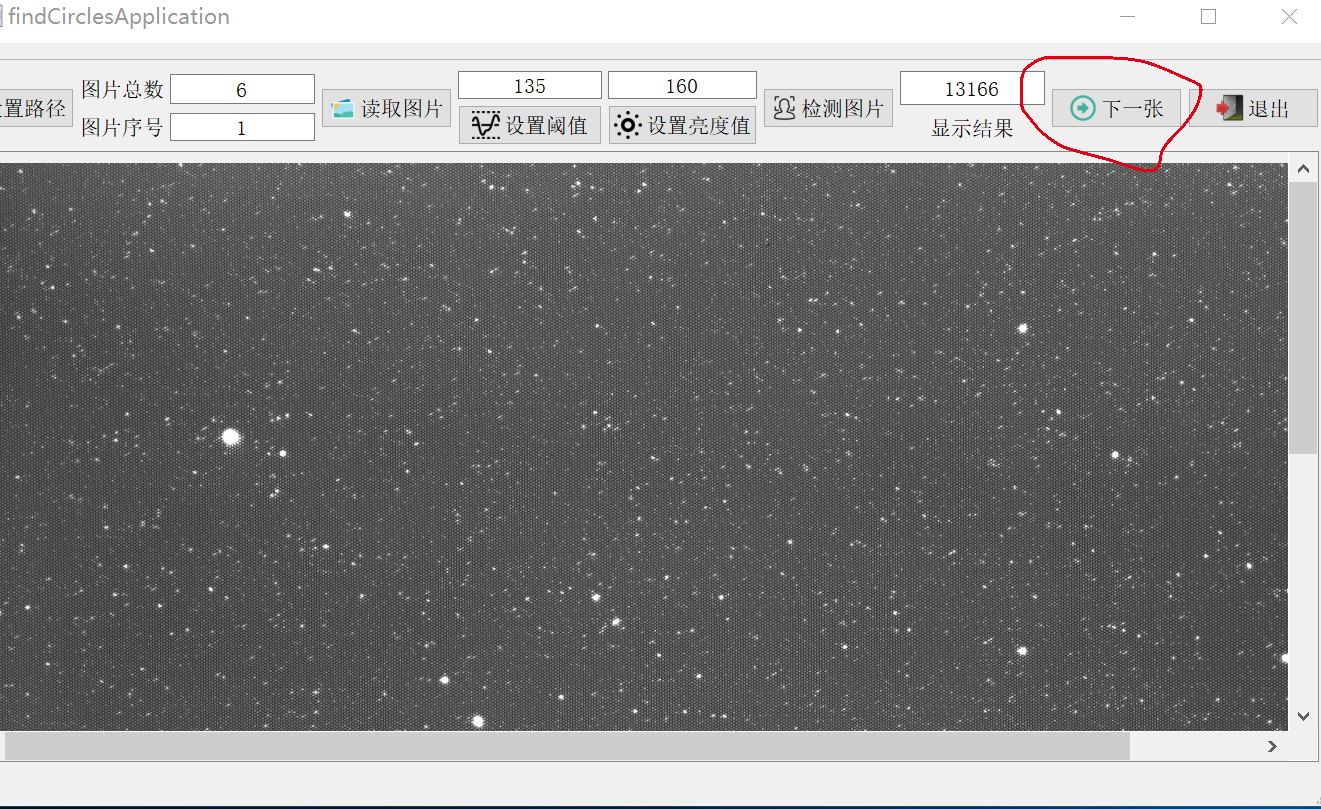
1. 在设置好阈值和亮度值后（如果是默认值，不用修改），点击“检测图片”





之后，“显示结果”处的数值便为此张图中的小圆斑的个数。

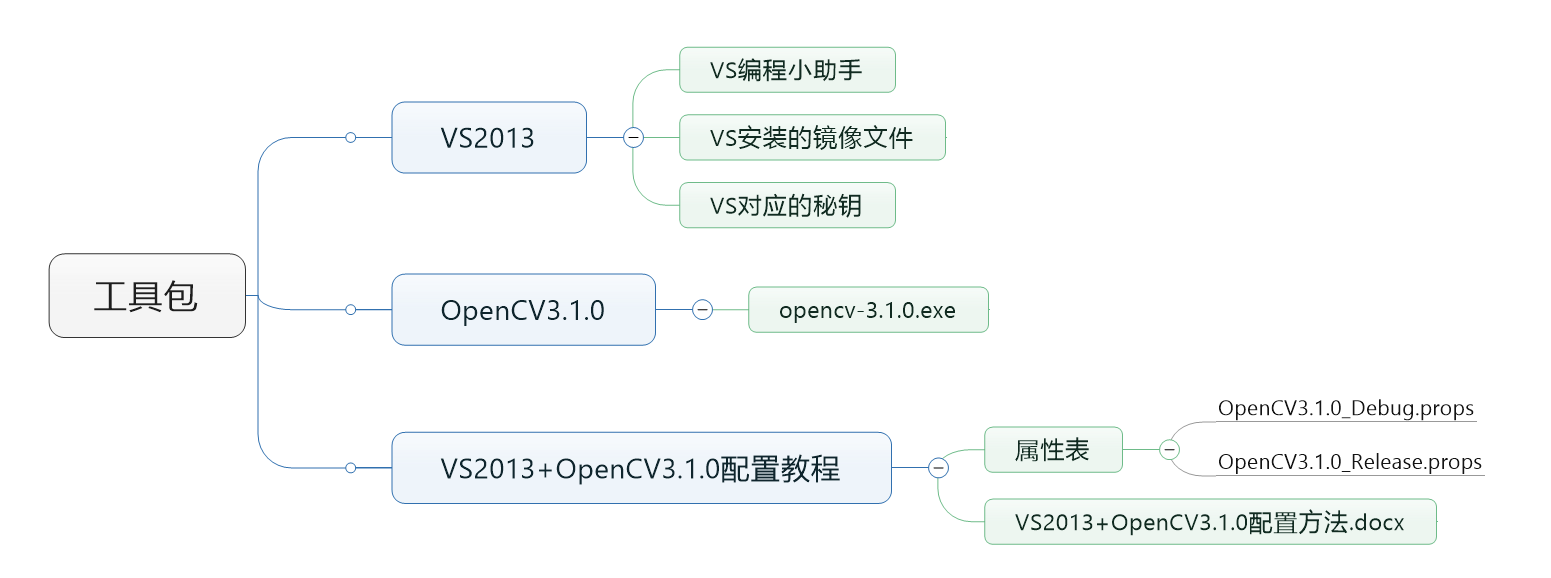
7)如果这张测试完成之后，点击“下一张”，便可以开始检测了。



## 二 关于工程文件的配置安装

### 2.1 工具包：VS2013+OpenCV3.1.0

见附件“工具包”



配置方法请参考文档“VS2013+OpenCV3.1.0配置方法.docx”。

## 三 算法思想

Step1: 将图像中像素值特别大的（大的圆斑）地方，像素值赋值为0；

Step2: 对图像进行二值化。

大于某个亮度值的地方，便将其赋值为255。 此处是将小圆斑的地方变亮，非小圆斑的地方全部变暗，像素值赋值为0;

Step3：进行边缘检测。

如果有亮的地方，必能检测到边缘。边缘的像素值个数应该在（1,5）之间。

Step4：由于每次图片中的大圆斑的个数不同，因而图片的亮度也不同，因而每次需要区别对待，也就是阈值的选择需要手动输入.

理论上：二值化阈值范围应该为（120,160）之间；

亮度阈值范围应该在（160,255之间）；

## 四 程序实现核心代码

void main()

{

//读取图片

Mat imgSrc = imread("./1.jpg", 0);

//将该图片做个备份

Mat img;

imgSrc.copyTo(img);

//将大圆斑的像素值全部设为0；

for (size\_t i = 0; i < img.rows; i++)

{

for (size\_t j = 0; j < img.cols; j++)

{

int value = img.at<uchar>(i, j);

if (value>160) //此处暂且将亮度阈值设为160；

{

img.at<uchar>(i, j) = 0;

}

}

}

//图像二值化 将小圆斑变亮，非小圆斑地方涂黑。

Mat img\_threshold;

threshold(img, img\_threshold, 135, 255, CV\_THRESH\_BINARY);

//边缘轮廓检测

vector<vector<cv::Point>> contoursOrigin;

vector<cv::Vec4i> hierarchy;

findContours(img\_threshold, contoursOrigin, hierarchy, CV\_RETR\_LIST, CV\_CHAIN\_APPROX\_NONE);

int nLittleCircle = 0; //申请变量统计小圆斑个数

if (!contoursOrigin.size())

{

return;

}

else

{

// 寻找小圆斑 边缘中的像素个数在（1,5）之间的，便算作是小圆斑。

for (int i = 0; i < contoursOrigin.size(); i++)

{

if (contoursOrigin[i].size() >= 1 && contoursOrigin[i].size() <= 5)

{

nLittleCircle++;

}

else

{

continue;

}

}

}

cout << "检测到的小圆斑个数：" << nLittleCircle << endl;

}