**天朗科技有限公司**

**Halcon培训教程**

Created by 朱为 许凡琪

2018.09.10

# 目录

[第一章 Halcon软件安装 1](#_Toc524437768)

[第二章 基本的界面介绍以及操作 5](#_Toc524437769)

[2.1 基本界面介绍 5](#_Toc524437770)

[2.2 熟悉自带例程 5](#_Toc524437771)

[第三章 图像相关基础知识： 8](#_Toc524437772)

[第四章 喷码项目实例 10](#_Toc524437773)

[4.1 罐底喷码识别一般流程 10](#_Toc524437774)

[4.2具体操作 10](#_Toc524437775)

[附录 喷码项目常用算子简介 22](#_Toc524437776)

**说明**

本教程仅限天朗科技有限公司内部学习交流使用，请勿上传至网络或发送给其他非本公司人员。

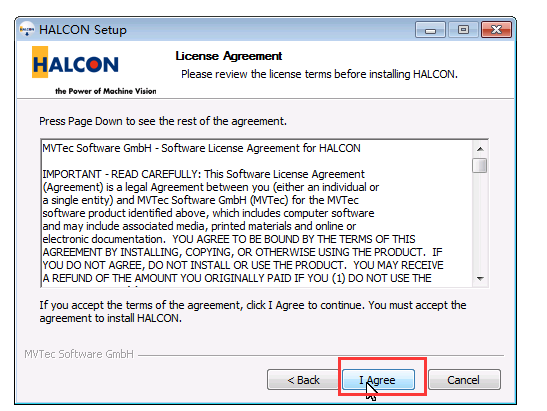
# 第一章 Halcon软件安装

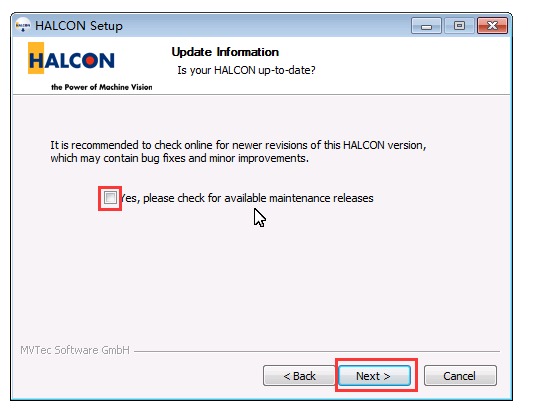
可以到官网([http://www.Halcon.com/Halcon/download/](http://www.halcon.com/halcon/download/))以及Halcon学习网([http://www.iHalcon.com/](http://www.ihalcon.com/))选择自己所需要的Halcon软件的版本。这里使用的是公司提供的Halcon10.0.

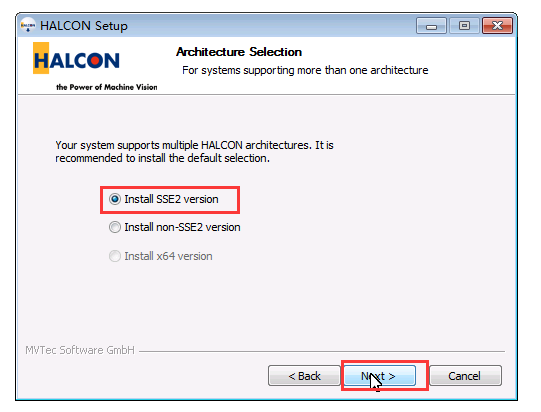
**安装过程如下：**

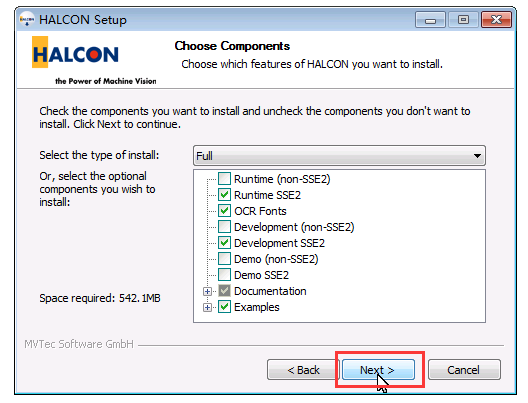
1、先安装Halcon-10.0-windows.exe。

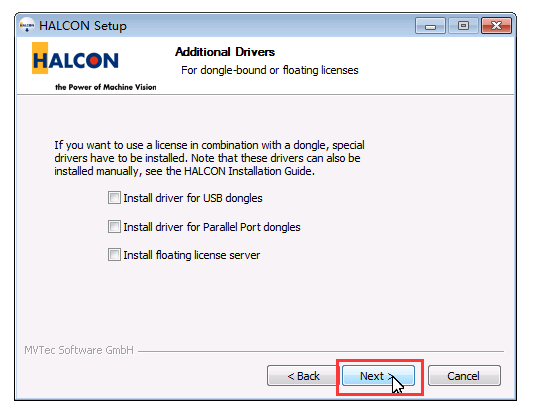


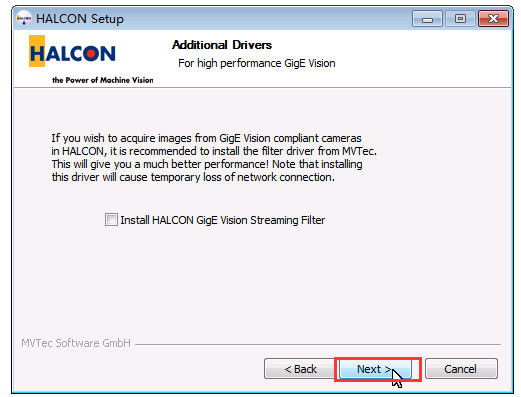


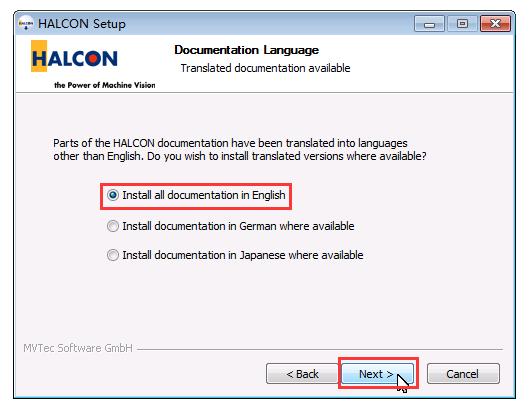


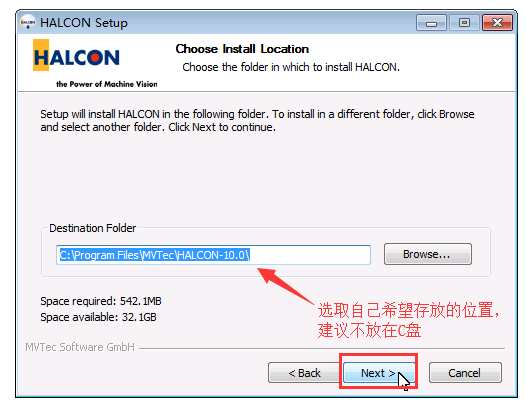


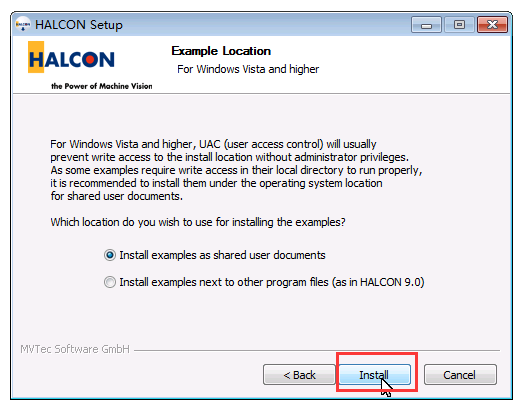












![C:\Users\Administrator\AppData\Roaming\Tencent\Users\1137789635\TIM\WinTemp\RichOle\7BS@S22Z}P512J](YBCEA2V.png](data:image/png;base64,)

2、安装Halcon-10.0-images-windows.exe，直接安装即可。

3、授权：

1. 把crack文件夹内的license文件夹内的license.dat文件复制到软件的安装目录下（默认路径C:\Program Files\MVTec\HALCON-10.0\license）

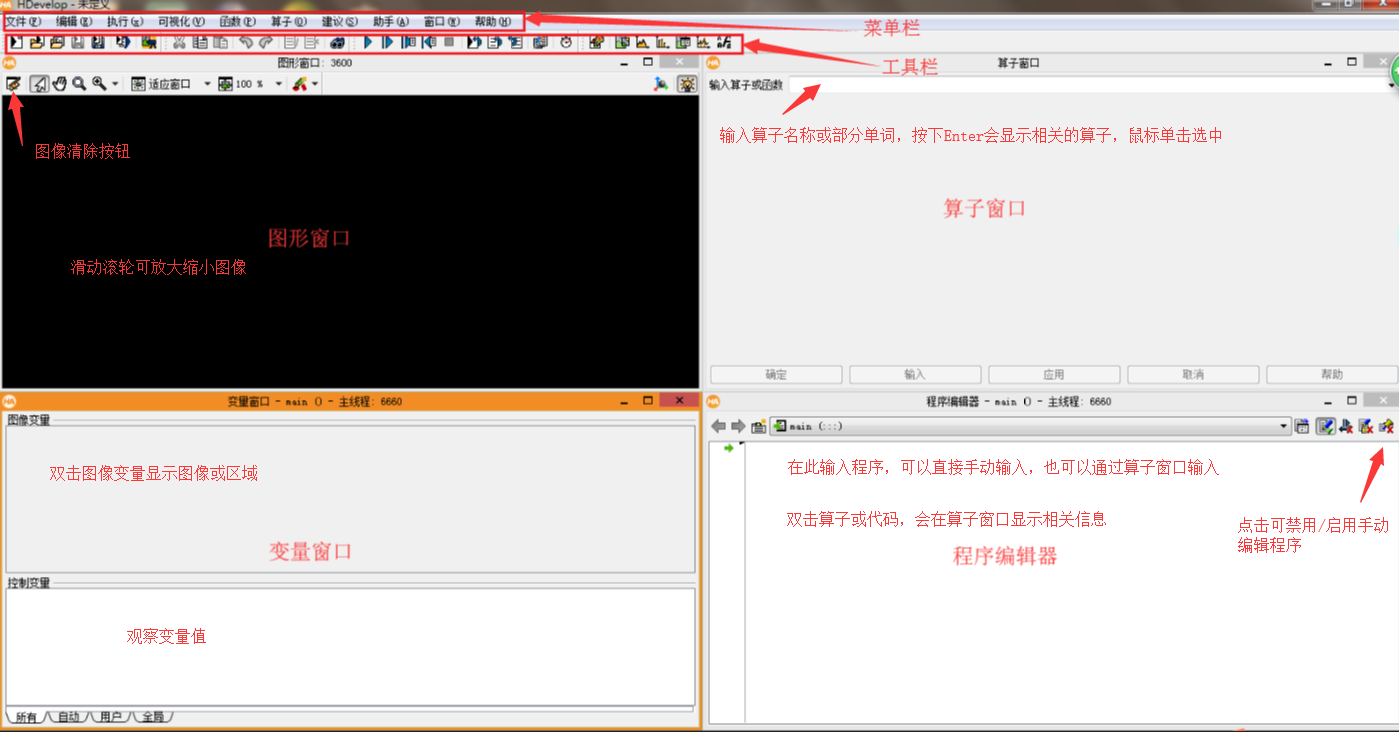
（如果刚开始是选择安装的HALCON\_10\_x86sse2-win32，请遵守第2步）

1. 把HALCON\_10\_x86sse2-win32\_Patcher.exe复制C:\Program Files\MVTec\HALCON-10.0\bin\x86sse2-win32目录下运行它即可（运行就是打开它之后，右下角有个Next点击，点击后，然后Finish就授权完成了，如果刚开始是选择安装的HALCON\_10\_x86-win32，请遵守第3步）
2. 把HALCON\_10\_x86-win32\_Patcher.exe复制C:\Program Files\MVTec\HALCON-10.0\bin\x86-win32目录下运行它即可

# 第二章 基本的界面介绍以及操作

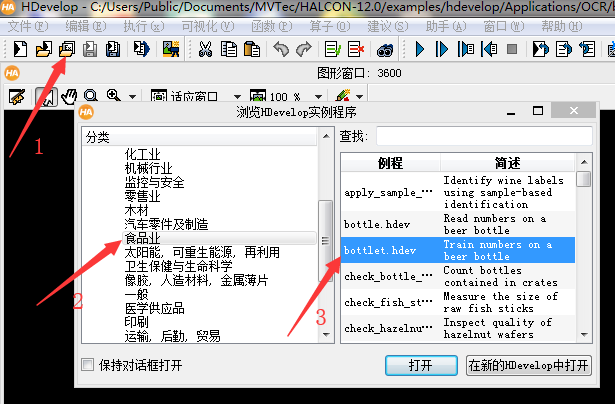
## 2.1 基本界面介绍

打开应用程序，可以看到主界面包含四个操作界面。

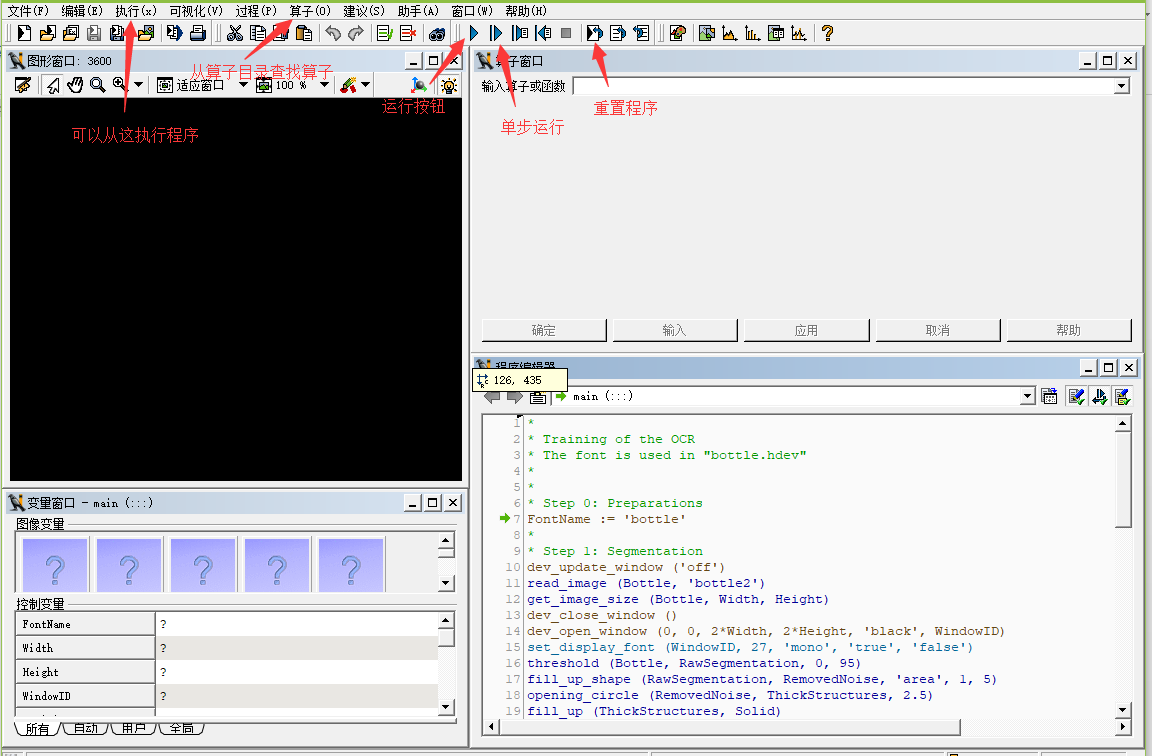


## 2.2 熟悉自带例程

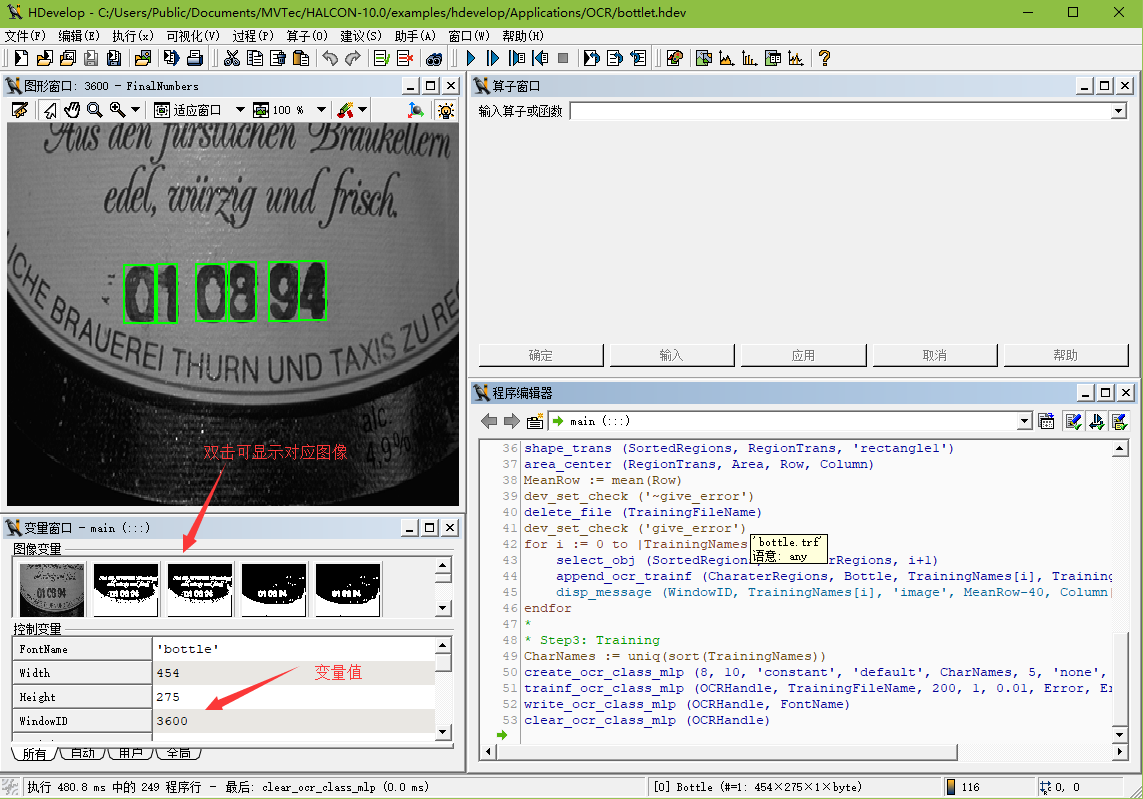
通过打开软件自带的某一例程熟悉一下环境。

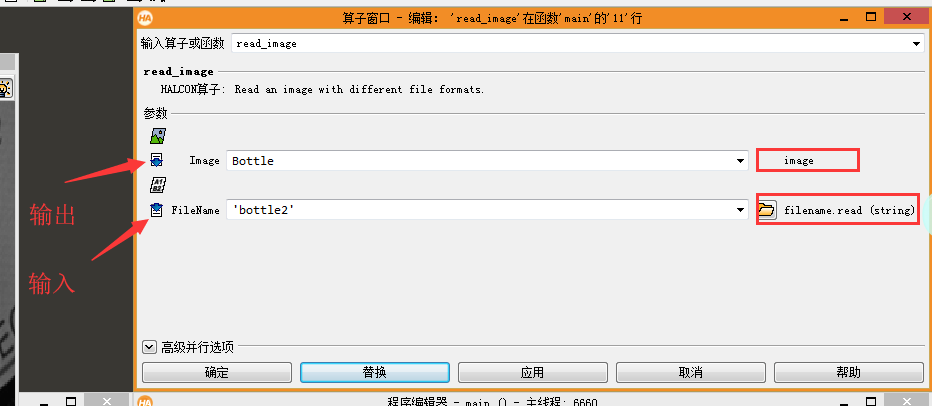


点击工作栏中的运行按键或者单步运行按键，程序将运行。

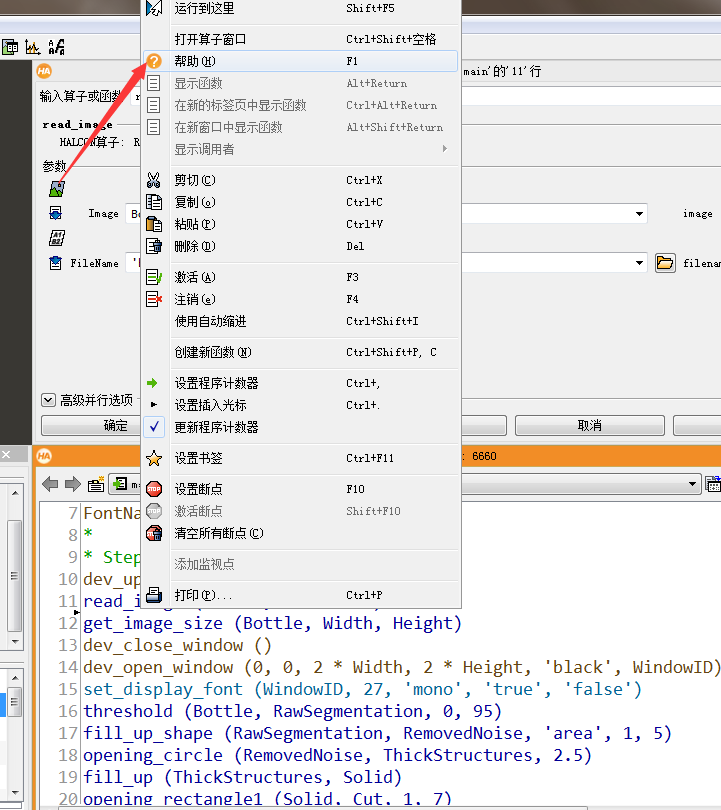


程序运行完之后的结果如下。我们能够在图像变量窗口和控制变量窗口上看到对应的变量值都显示了出来。



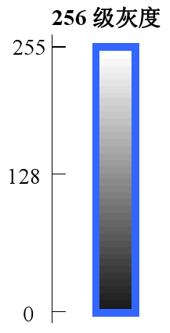
在理解该例程程序的过程中，可以双击某一算子，在算子窗口中将得到该算子的变量参数的信息描述。

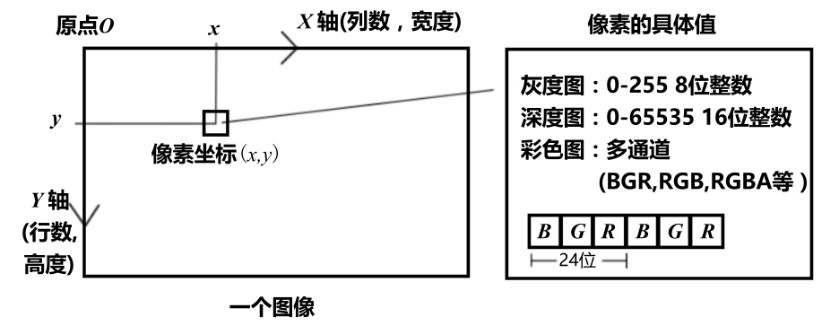
也可以选中该算子右键单击“帮助”，得到该算子更为细致的信息。



# 第三章 图像相关基础知识

摄像机拍摄物体并将图片传送到计算机。图像在计算机内按像素矩阵的形式存储，如下图所示。



**像素：**显示在显示器上光的单元，用来计算点阵图像的一种单位（即一个小方格），每个像素都被分配一个特定位置和颜色值。

每一幅图像都以左上顶点为坐标原点，水平向右表示X轴，即图像的宽或列数；垂直向下表示Y轴，即图像的高或行数。

每一个像素都有唯一的颜色值。当每个像素值用一个0-255的数表示时，这样的图像称为灰度图，0表示最黑，255表示最白；当每一个像素用三个0-255的数表示时，这样的图称为彩色图，它有三个通道。例如我们常见的彩色图是由红绿蓝（RGB）三种颜色混合而成。

**分辨率：**在一幅图像中，每单位长度能显示的像素数目，称为图像的分辨率（dpi:像素数/每英寸）。更一般的，我们常说的分辨率指的是图像的宽和高。例如，手机屏幕的分辨率为1920\*1080，意义是手机屏幕全屏显示时的图像的高为1920个像素，宽为1080个像素。

**二值（化）图像：**图像中所有的像素只有2个值，例如黑白图像，黑色部分的像素都是0，白色部分的像素都是255。二值图像一般是分割后的图像，图像中只有背景和前景。

**分割：**图像分割是将图像中有意义或者我们感兴趣的内容提取或者凸显出来，方便后续处理。例如把图像变换成二值化图像，把我们感兴趣的部分变成白色，其余背景变成黑色，这样就将感兴趣的区域分割出来了。

**噪声：**图像中凡是对我们感兴趣的目标产生干扰的部分都可以称之为噪声。在图像预处理阶段一般需要使用某些方法来滤除噪声，以更好凸显我们感兴趣的目标。

**滤波：**当我们使用某些方法除去图像中噪声的过程就称为滤波。不同的噪声一般有对应的滤波方法。

**形态学处理：**包括腐蚀、膨胀、开运算和闭运算。详细内容请自行百度。。。

**连通域：**连通区域（Connected Component）一般是指图像中具有相同像素值且位置相邻的前景像素点组成的整体区域。连通区域分析是指将图像中的各个连通区域找出来并标记。

**Ocr：**OCR （Optical Character Recognition，光学字符识别）是指电子设备（例如扫描仪或数码相机）检查纸上打印的字符，通过检测暗、亮的模式确定其形状，然后用字符识别方法将形状翻译成计算机文字的过程。更一般地，指计算机识别图片中的文字。

**Mlp：**即多层感知机，是一种前向结构的人工神经网络，映射一组输入向量到一组输出向量。例如输入一张图片，输出一串（个）字符。神经网络需要大量样本用于训练以确定神经元的参数。常用的训练方法为反向传播（BP）算法。

# 第四章 喷码项目实例

## 4.1 罐底喷码识别一般流程

1. 读取图像。
2. 图像滤波（可选）。通过滤波除去图像中的噪点、平滑图像以及增强文字区域等。
3. 阈值分割。通过设定阈值将图像变为二值图像，即感兴趣区域（文字区域）为白色，背景为黑色。
4. 形态学处理与连通域分析。将整个文字区域连成一个整体，以便定位文字区域。
5. 文字区域定位。通过面积、高、宽等特征将文字区域筛选出来。
6. 减小区域。将文字裁剪出来，方便后续处理。
7. 形态学处理与连通域分析。将单个字符连接起来。
8. 字符排序。将单个字符按行或列排序。
9. 识别。利用Halcon的ocr模块识别单个字符。

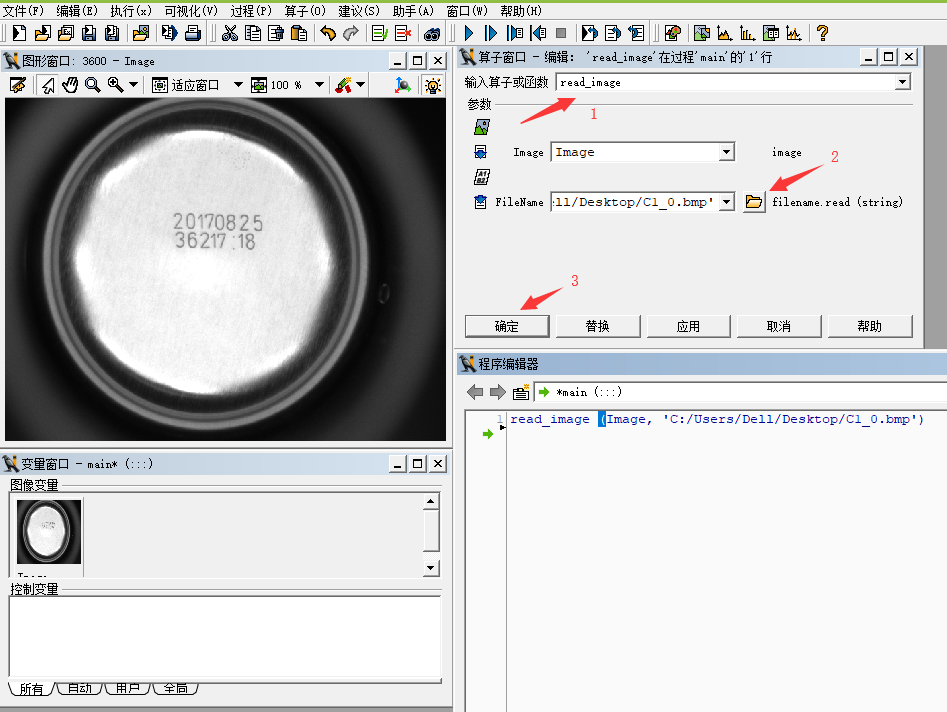
## 4.2具体操作

***1 读取图像***

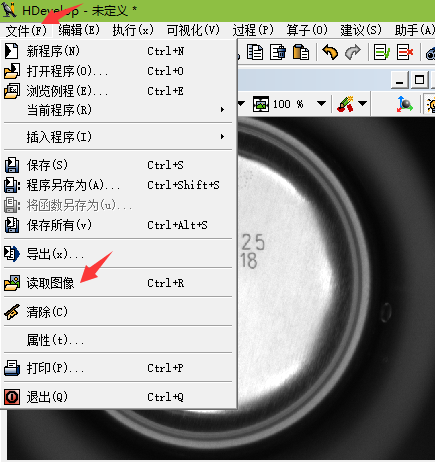
要读取的图像如下图所示。（可右键另存为bmp格式图片）



读取图像有两种方法。1）使用Halcon算子。在算子窗口输入 read\_image并按下 Enter，即可调出图像读取函数。然后打开文件对话框，选择相应的图片，确定就可以读取到图像（点击应用可以预览图像，但不会在程序编辑界面输入代码）。操作如下：

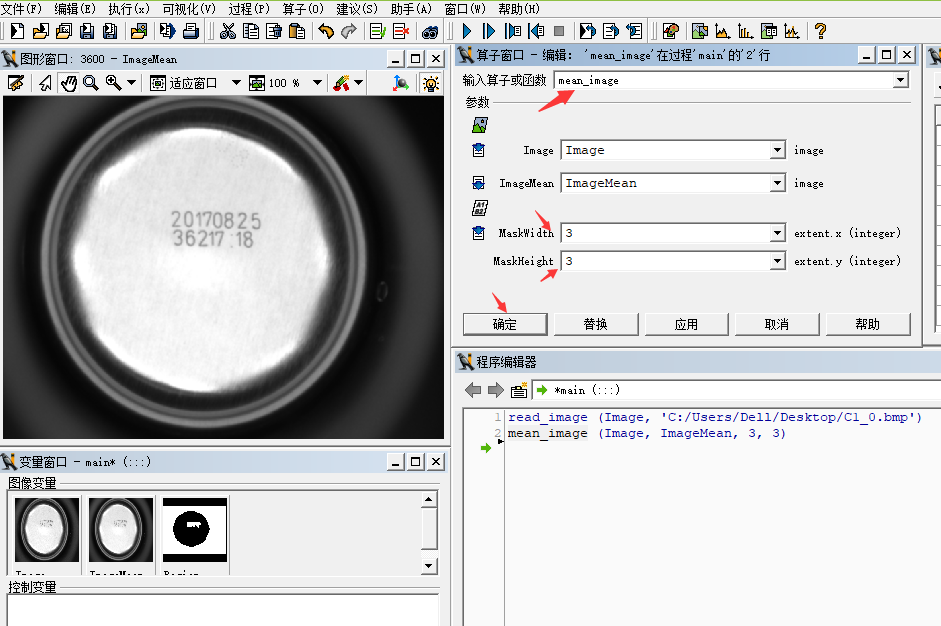


2）使用菜单栏读取图像。依次单击文件🡪读取图像，在打开的对话框中选择相应的图片，变量名称随意，一般设为Image，点击确定即可读取到图像。操作如下所示：



***2 图像滤波***

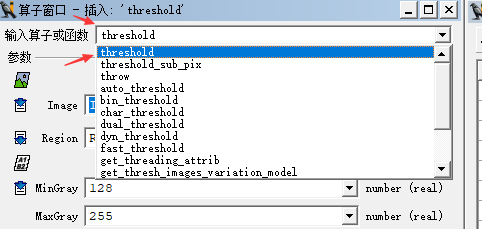
Halcon提供了多种滤波方法，常见的如中值滤波、均值滤波和二项式滤波等。这里我们选择均值滤波。在算子窗口中输入 mean\_image并按下 Enter，在算子界面修改相应的参数，变量名称保持默认值即可，点击确定。操作如下：

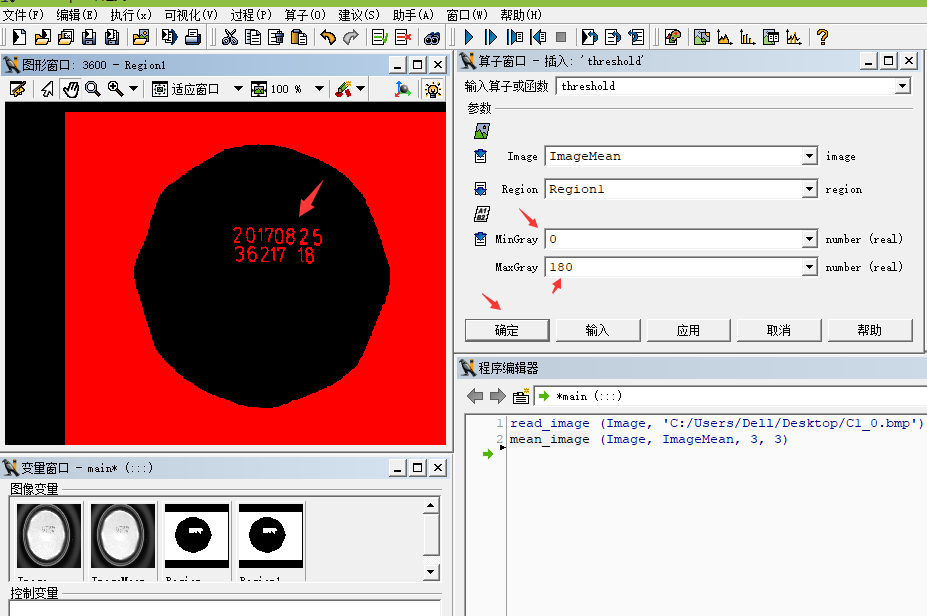


**温馨提示：点击图像窗口左上角的清除按钮，可以清除当前图像，这样就不会影响后面图像的显示。**

***3 阈值分割***

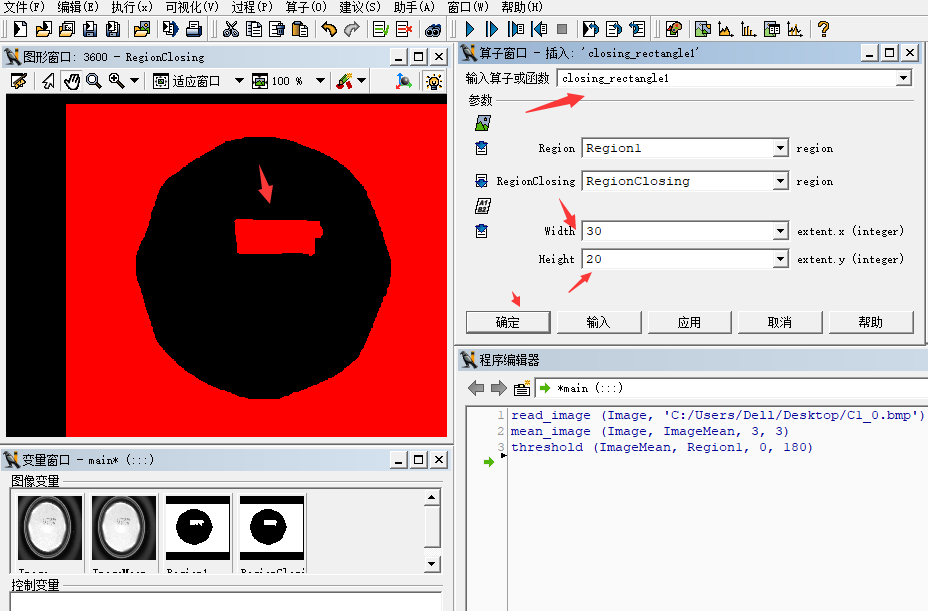
通过设定阈值，把前景和背景分开。计算阈值的方法一般有固定阈值、动态阈值、局部阈值等，根据实际情况进行选择。这里我们选择固定阈值。在算子窗口输入 thr，并按下键盘的enter键，会出现多个与阈值有关的算子，选择 threshold算子。在参数界面，设定最小灰度值和最大灰度值。图像中灰度值位于最小值与最大值之间的像素会被分割成前景，一般以红色显示。图像中字符偏黑，背景偏亮，字符所在阈值范围是0—n（n为非零数），所以我们把最小值设置为0，最大值设定为非0阈值，这里设定为180，这样就能把字符变成前景，也就是所谓的感兴趣区域。操作如下所示：



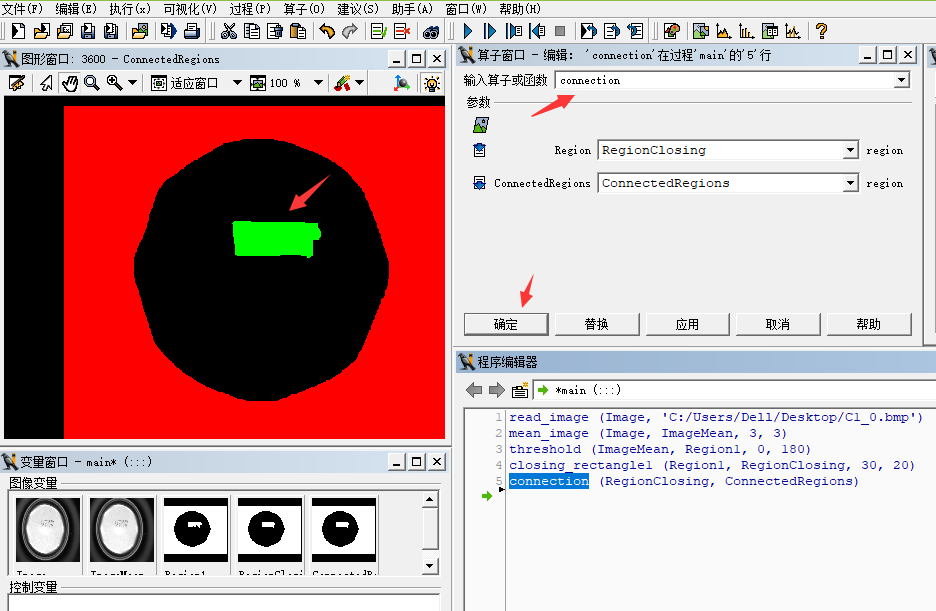


***4 形态学处理和连通域分析***

现在字符和一部分背景已经被分割成感兴趣区域，我们需要把字符区域单独提取出来（许多情况下图片会有很多噪声，无法做到只把字符分割出来，所以先定位整个字符区域）。我们利用形态学的方法将字符连成一个整体，这里使用闭矩形算子。在算子窗口输入 closing并按下Enter，选择 closing\_rectangle1算子，在参数界面修改参数。修改参数时可以点击应用注意观察图像的变化，把字符区域连成一个整体，而且不能与其他非字符区域粘连。操作如下所示：

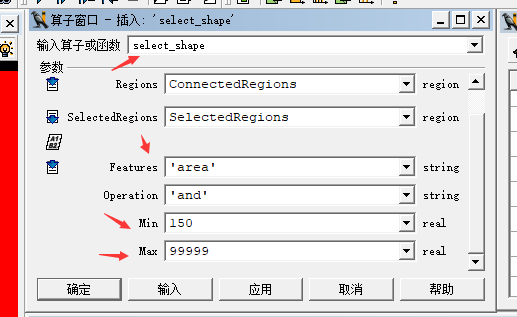


在算子窗口输入 connection，把相连像素区合并成一个元素。操作如下所示：

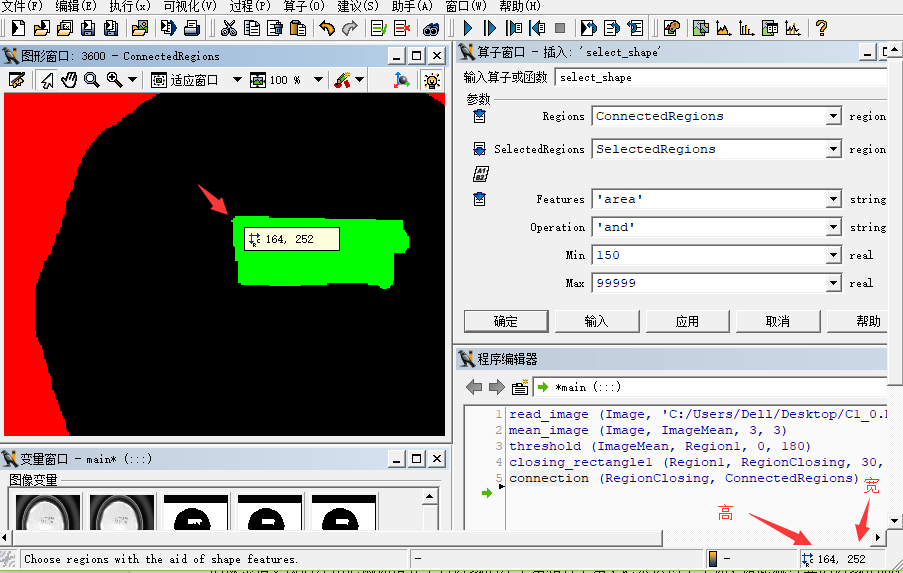


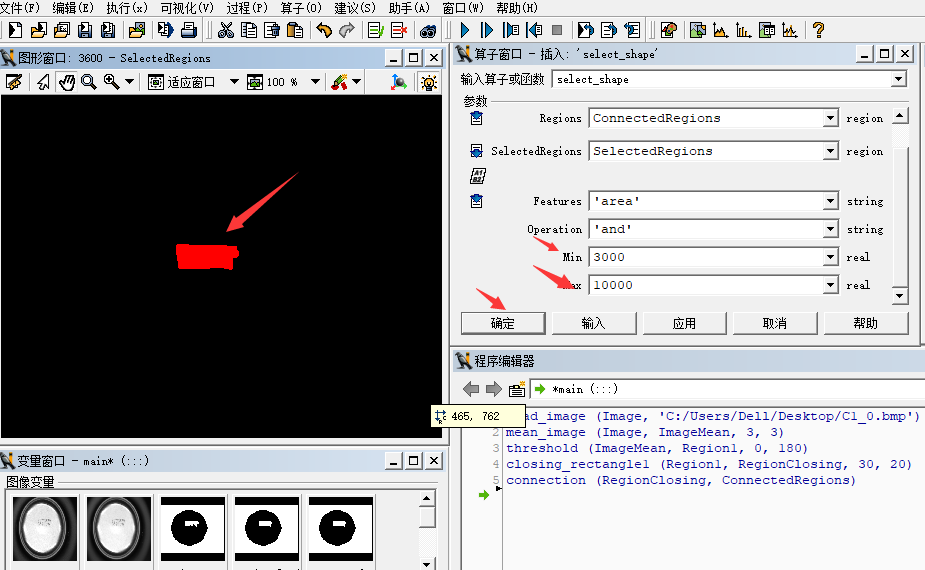
***5 文字区域定位***

由于字符区域面积基本不变，我们可以通过面积这个特征将文字区域筛选出来。在算子窗口输入 select，选择 select\_shape算子。在参数界面，特征默认是面积（area），操作默认是与，这两个参数可以不用修改，我们只需修改最大和最小值。



那么如何获取面积呢？当鼠标落在图像窗口时，在Halcon界面右下角会实时显示鼠标的坐标以及相应位置的像素值。我们把用鼠标滚轮把图像放大，分别把鼠标指在字符区域的左上角和右下角，记录这两个坐标，据此就可算得区域的面积。这样计算出的面积虽然不是特别准确，但足够我们把字符区域筛选出来。根据计算出的面积设定合适的范围即可把文字区域筛选出来。操作如下所示：

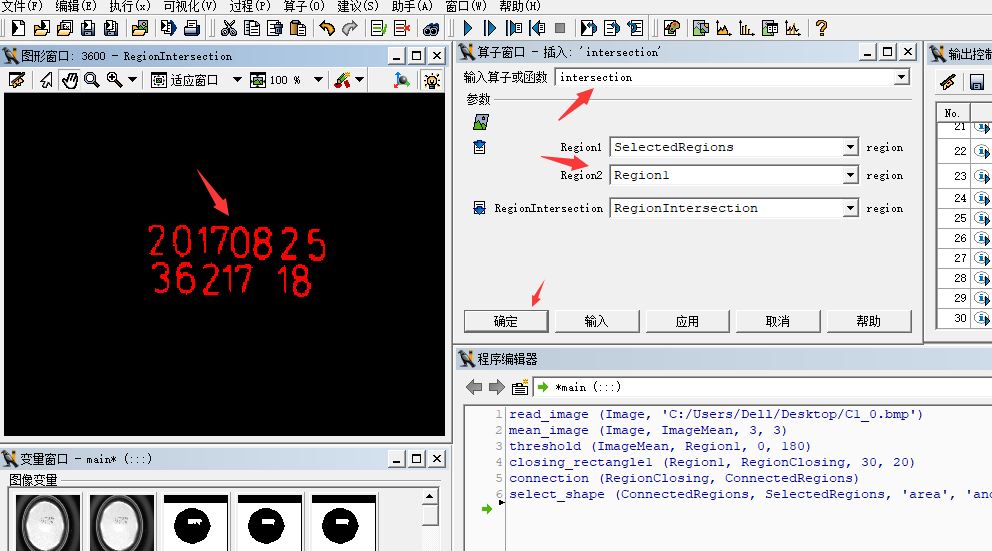




**点击确定后如果发现图像没有变化，可先点击图像窗口左上角的清除按钮，再在程序编辑器中双击 select\_shape算子，在算子窗口重新点击确定或应用即可。**

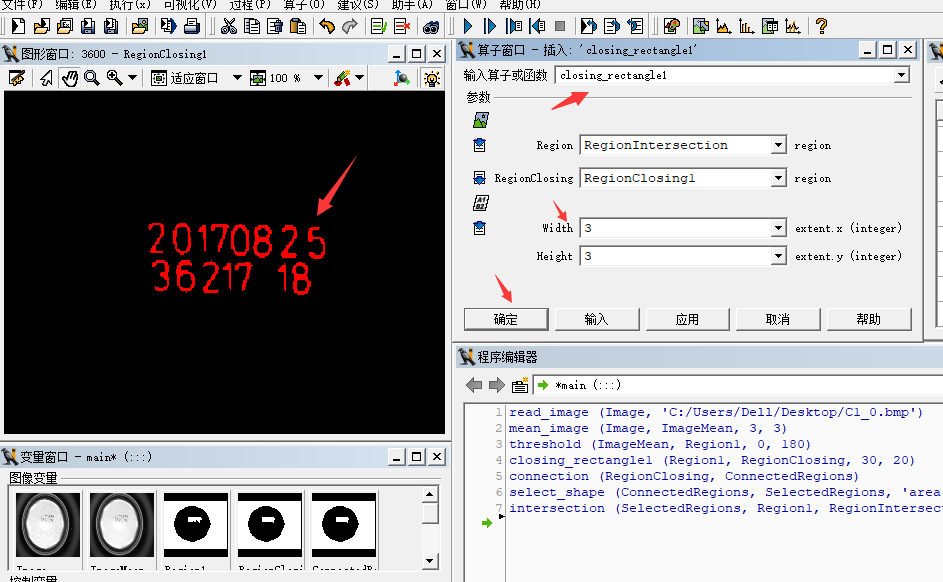
***6 缩小区域***

定位字符区域后，我们把字符区域单独截取出来，这样在后续处理中就避免了背景的干扰。我们将上一步得到的区域与第三步分割的图像做交集，就能得到只有字符的二值图像。在算子窗口输入 inter，选择 intersection算子。在参数界面，修改Region2参数，点击倒三角，选择Region1，即第三步得到的图像，点击确定就能得到分割后的字符。操作如下：

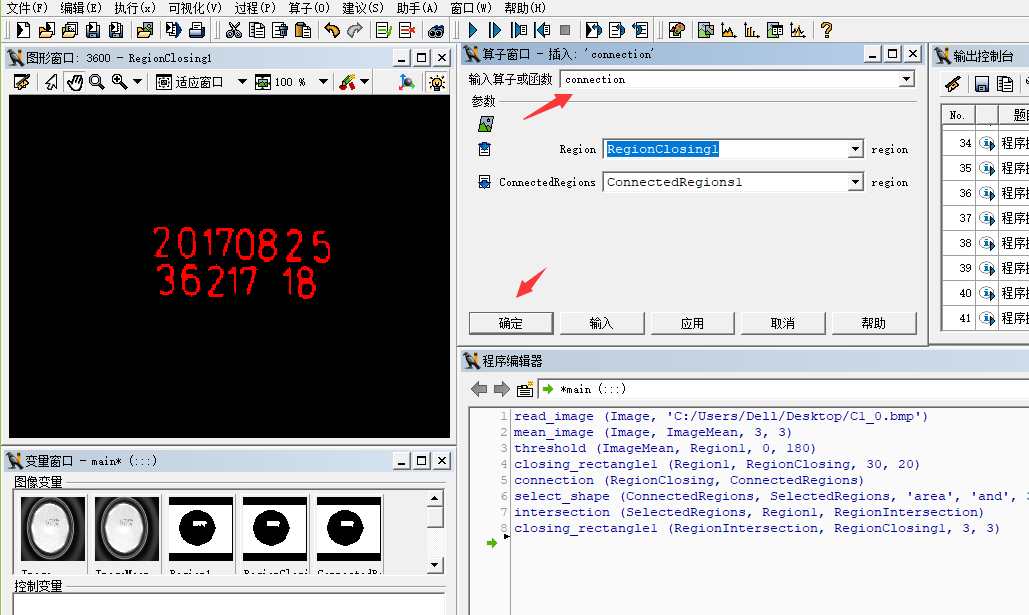


***7形态学处理与连通域分析***

此步骤与第四步不同。第四步是为了把所有字符连成一个区域，此步骤是为了把单个字符连接起来。同样，在算子窗口输入 closing，选择 closing\_rectangle1。由于此步骤是为了把单个字符连接起来，所以参数要设置得小。在这个例子中，阈值分割的效果较好，形态学处理也可以省略。形态学操作如下：

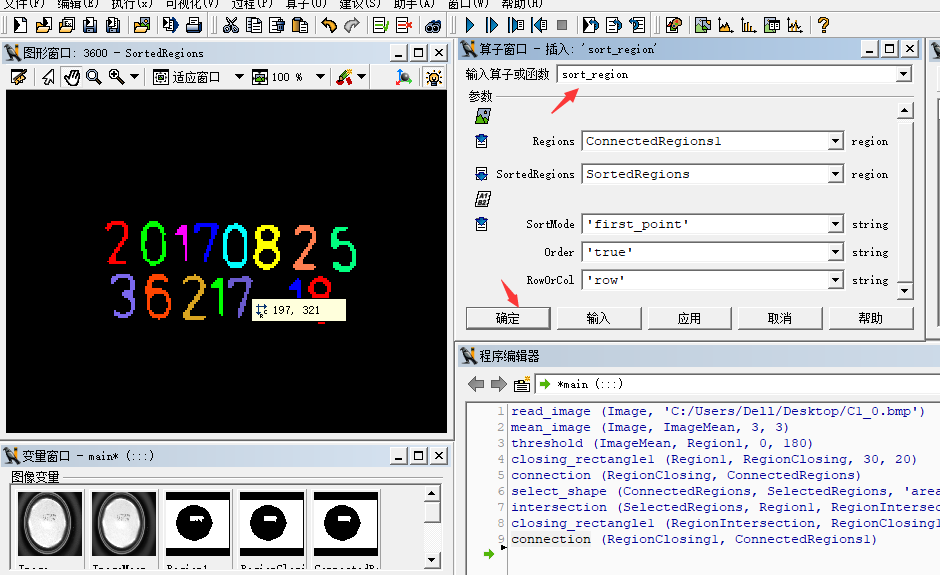


在算子窗口中输入 connection，并点击确定。操作如下：



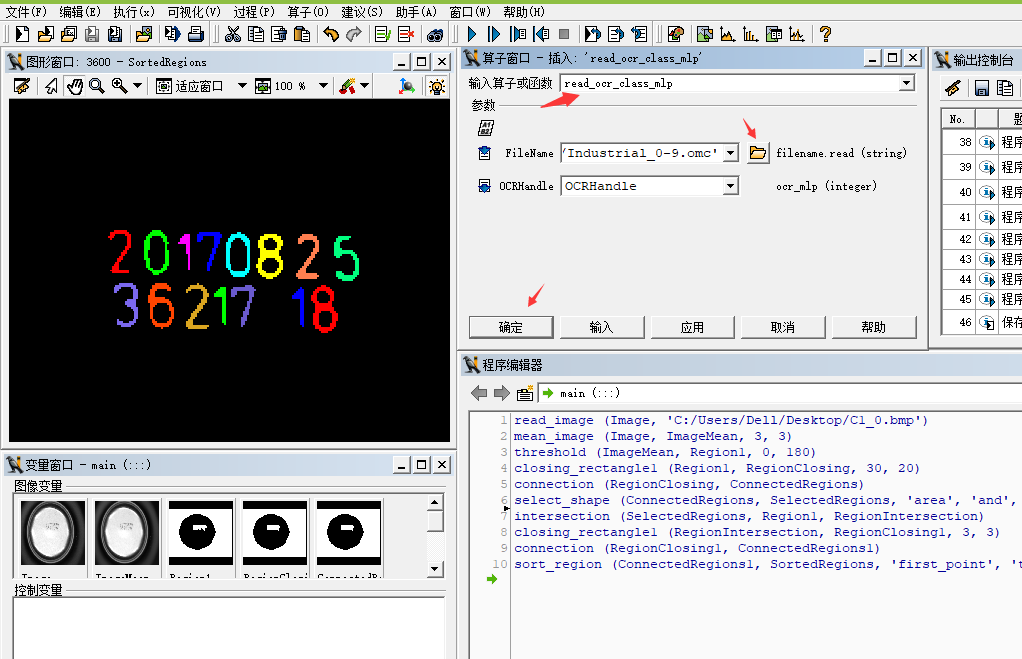
***8字符排序***

在算子窗口输入sort，选择 sort\_region，参数默认即可，点击确定。

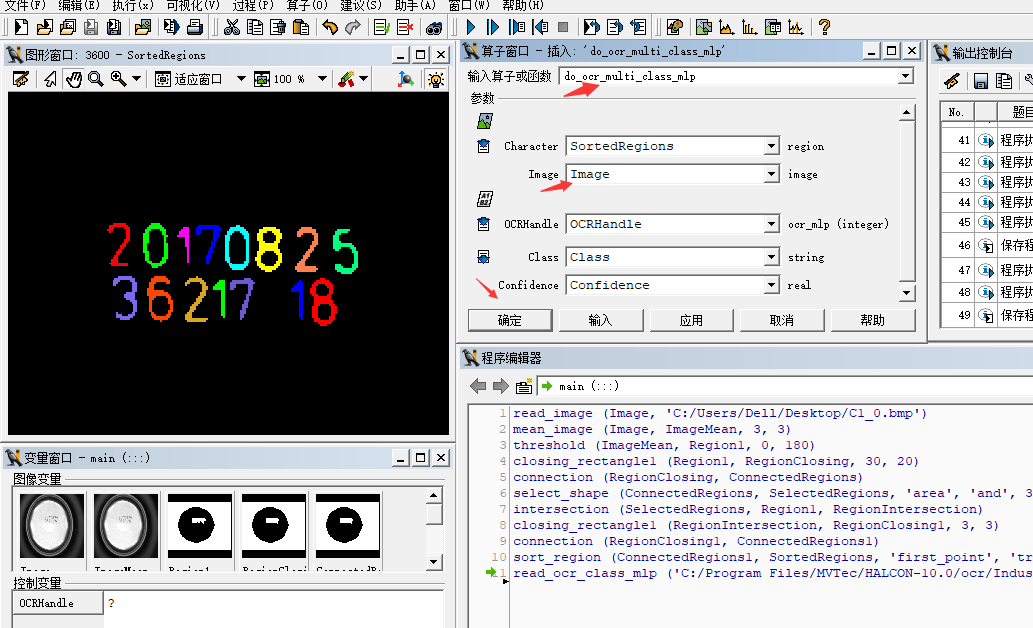


***9 字符识别***

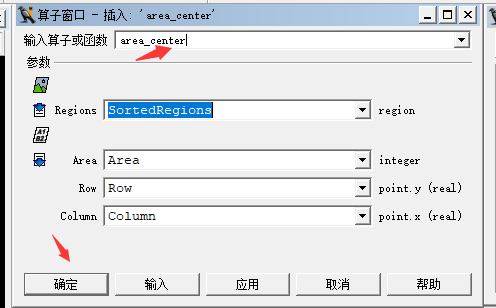
字符识别我们采用Halcon提供的mlp即多层感知机算法。在算子窗口输入 read，选择 read\_ocr\_class\_mlp算子。在参数界面，文件名选择Halcon提供的训练模型，该文件在Halcon安装目录下，默认为：C:\Program Files\MVTec\HALCON-10.0\ocr。模型选择Industrial\_0-9.omc。如下图所示：



读取完模型后就可以对我们的字符进行识别。在算子窗口输入 class，选择do\_ocr\_multi\_class\_mlp算子。在参数界面，图像Image选择第一步中的输入图像，点击确定。



在算子窗口输入 area，选择 area\_center算子，获取每个字符的中心坐标。



在程序编辑器中输入如下代码，用以显示识别到的字符：

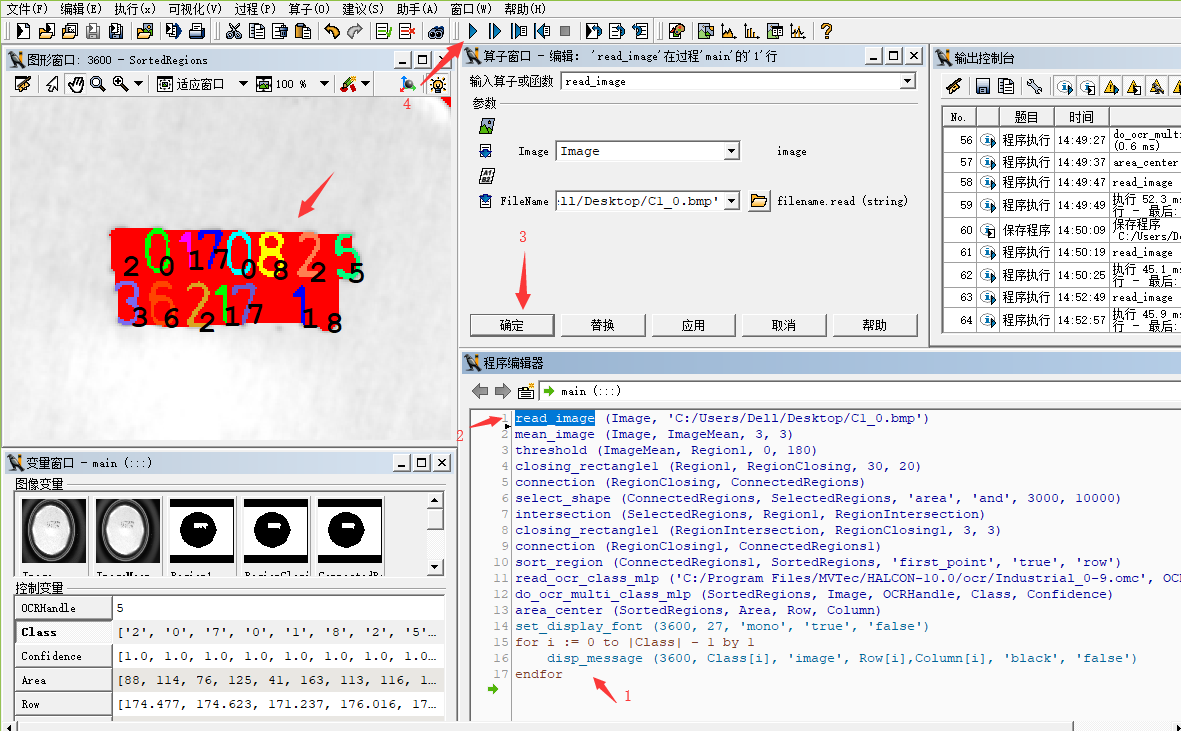
set\_display\_font (3600, 27, 'mono', 'true', 'false')

for i := 0 to |Class| - 1 by 1

disp\_message (3600, Class[i], 'image', Row[i],Column[i], 'black', 'false')

endfor

双击 read\_image算子，点击确定，将程序定位到开头。单机工具栏的运行按钮，在图像窗口中就会看到图像处理结果和识别的字符。如下图所示：



至此，第一个喷码识别项目完成。程序完整代码如下：

read\_image (Image, 'C:/Users/Dell/Desktop/C1\_0.bmp')

mean\_image (Image, ImageMean, 3, 3)

threshold (ImageMean, Region1, 0, 180)

closing\_rectangle1 (Region1, RegionClosing, 30, 20)

connection (RegionClosing, ConnectedRegions)

select\_shape (ConnectedRegions, SelectedRegions, 'area', 'and', 3000, 10000)

intersection (SelectedRegions, Region1, RegionIntersection)

closing\_rectangle1 (RegionIntersection, RegionClosing1, 3, 3)

connection (RegionClosing1, ConnectedRegions1)

sort\_region (ConnectedRegions1, SortedRegions, 'first\_point', 'true', 'row')

read\_ocr\_class\_mlp ('C:/Program Files/MVTec/HALCON-10.0/ocr/Industrial\_0-9.omc', OCRHandle)

do\_ocr\_multi\_class\_mlp (SortedRegions, Image, OCRHandle, Class, Confidence)

area\_center (SortedRegions, Area, Row, Column)

set\_display\_font (3600, 27, 'mono', 'true', 'false')

for i := 0 to |Class| - 1 by 1

disp\_message (3600, Class[i], 'image', Row[i],Column[i], 'black', 'false')

endfor

# 附录 喷码项目常用算子简介

|  |  |
| --- | --- |
| 算子名称 | 描述 |
| read\_image | 读取不同文件格式的图像 |
| dots\_image | 增强图像中的圆形点，常用于点阵喷码项目 |
| mean\_image | 均值滤波 |
| median\_image | 中值滤波（常用于去除孤立的噪点） |
| binomial\_filter | 二项式滤波（抑制小斑点，常与emphasize\_image组合） |
| threshold | 固定全局阈值分割 |
| dyn\_threshold | 动态局部阈值分割（需与mean\_image等滤波组合） |
| bin\_threshold | 根据自动产生的阈值分割图像 |
| closing\_circle | 用圆形结构封闭一个区域（常用来增大单字符或区域的轮廓） |
| closing\_rectangle1 | 用矩形结构封闭一个区域（常用来增大单字符或区域的轮廓） |
| opening\_circle | 对区域进行开操作，常用来去除孤立的噪点或断开细线 |
| opening\_rectangle1 | 对区域进行开操作，常用来去除孤立的噪点或断开细线 |
| connection | 计算连通域，把相邻像素连成一个整体 |
| select\_shape | 选择形状，根据不同的特征如面积、高、宽等进行选择 |
| select\_shape\_std | 进一步选择形状，常接于select\_shape之后 |
| intersection | 两个Region做交集，保留共有部分 |
| smallest\_rectangle2 | 寻找一个区域的最小包围矩形，这个矩形可以是有角度的 |
| tuple\_deg | 将角度单位从弧度化为度 |
| rotate\_image | 旋转图像 |
| count\_obj | 计算前景（感兴趣区域）数量 |
| gen\_rectangle1 | 生成水平矩形 |
| gen\_rectangle2 | 生成带角度的矩形 |
| reduce\_domain | 缩小图像操作域 |
| shape\_trans | 转换形状，常用来将单个字符区域转换成矩形区域 |
| partition\_rectangle | 根据给定的宽和高拆分大矩形 |
| sort\_region | 对区域进行排序 |
| area\_center | 计算区域面积和中心坐标 |
| read\_ocr\_class\_mlp | 读取mlp模型 |
| do\_ocr\_multi\_class\_mlp | 使用mlp模型识别多个字符 |
| set\_display\_font | 设置图像窗口显示文本的字体 |
| disp\_message | 显示文本信息 |
| dev\_clear\_window | 清除图像窗口 |
| dev\_set\_color | 设置窗口显示颜色 |
| dev\_display | 显示图像或区域 |
| list\_files | 列出文件夹下所有文件，常用于批量处理 |
| tuple\_regexp\_select | 选择给定格式的文件，常用于批量处理 |