片层自动处理程序说明

1. 目的

为了克服人工检测热障涂层微观图像的一系列缺陷，提出利用计算机图像处理技术代替人工作业进行片层图像处理，构建基于机器视觉的热障涂层中单个片层的形貌特征参数的识别和计算系统。

1. 工程包含文件
2. gui\_detect.py：程序的Python源代码
3. gui\_detect.exe：系统的可执行文件
4. test1.tif，test2.tif：两个测试图片
5. 程序运行环境

操作系统：win10

运行语言：Python 3.7.3

第三方库：numpy 1.16.4，opencv-python 3.4.2，PyQt 5.9.2

1. 方法和算法

本系统是基于计算机图像处理技术开发的一款软件，具体研究内容有一下几个方面：

（1） 热障涂层图像中单个片层轮廓提取；

（2） 片层铺展率的计算；

（3） 系统图形界面的构建；

（4） 系统的打包发布。

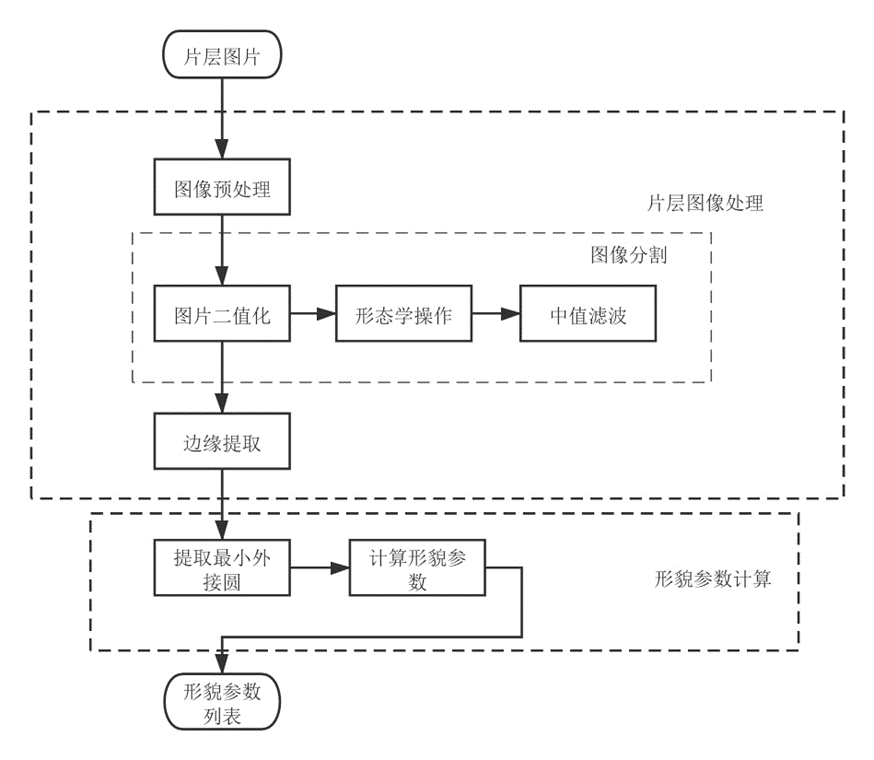
这个系统的核心是对片层进行分割和轮廓提取，所用到的主要是机器视觉中的图像分割和边缘提取两种技术。

图像分割是图像处理的一个重要分支，是实现图像处理到图像分析的一个关键节点，是图像理解的关键性步骤。图像分割通过一系列操作将一张图片的前景物体与背景分割开来：前景物体像素的灰度值通常与背景有显著差别。图像分割的目的就是在一张图片中提取出需要的具有相似性质的特征区域，以便后续针对每个目的区域进行对应处理。本文在片层图像分割时所选用的阈值分割方法是最大类间方差法（Otsu），是一种无参数无监督的自适应的阈值分割方法。后文会对此方法进行具体介绍。

边缘可以定义为一张图片中灰度值急剧变化的像素点的连线，而此变化的分界值即为上文所介绍的阈值，因此阈值分割与边缘提取是两种紧密相关的技术。边缘是图像的最基本特征，是在图像识别中需要重点关注的重要图像特征，只有提取出边缘才能真正地将前景目标与背景分割开来。

因此若要提取出片层的轮廓，首先需要使用图像分割算法对图像进行二值化处理；其次使用数学形态学算法对图像进行一些列处理使得前景与背景的特征差异变得明显，以使片层的轮廓更易提取出来；最后使用边缘提取算法提取出片层的轮廓。具体使用的算法如下：

热障涂层中单个片层的形貌特征参数的识别和计算软件用最大类间方差法求取二值分割阈值，运用均值滤波和形态学操作为片层图片去噪并保证单个片层连通性，通过OpenCV框架中的轮廓提取函数来获得片层边缘信息，最后根据所提取出的轮廓计算片层的实度等重要参数。流程图如下图所示。



热障涂层中单个片层的形貌特征参数的识别和计算软件主要组成

接下来对于上文中所提到的计算机图像处理算法作一些具体的介绍。

1. **最大类间方差法**

最大类间方差法是一种常用的自适应二值化阈值确定方法，简称OTSU。它是建立在一幅图像的灰度直方图基础上的，依据使类间距离极大为准则来确定区域分割阈值。OSTU阈值的具体计算方法为：首先计算影像的直方图；将0~255依次当作阈值，把直方图强度大于阈值的像素分成一组，把小于阈值的像素分成另一组；接下来分別计算这两组的组内方差，并把两个组内方差相加；最后比较每个阈值之下的组内方差和，总和最大的就是结果阈值。最终以此阈值为标准进行图像二值化操作，将灰度值大于此阈值的像素点赋为255，将灰度值小于此阈值的像素点赋为0。

1. **数学形态学**

数学形态学是一门建立在集合代数基础之上，用集合论的方法定量描述几何结构的学科，是数学形态学图像处理的基本理论。形态学操作就是基于数学形态学的一系列图像处理操作。OpenCV为进行图像的形态学变换提供了快捷、方便的函数。最基本的形态学操作有两种：膨胀与腐蚀。用这两种基本的形态学操作组合起来可以实现多种图像形状和结构的分析及处理。

实现膨胀和腐蚀操作前首先需要定义结构元素。结构元素是形态学的基本算子，也称核。设有两幅图像A和B。若A是被处理的对象，而B是用来处理A的，则称B为结构元素。结构元素在每个像素位置上与图像的对应区域进行特定的逻辑运算，逻辑运算结果为输出图像的相应像素，所以结构元素的选取对图像处理的结果有重大影响。结构元素中心拥有一个单独定义出来的参考点，称为锚点。

腐蚀：用B来腐蚀A得到的集合是B完全包括在A中时B的参考点的集合，用公式表示为：



膨胀：把结构元素B平移x后得到Bx，若Bx击中A，记下这个x点。所有满足上述条件的x点组成的集合称为A被B膨胀的结果，用公式表示为：



1. **中值滤波**

中值滤波法是一种非线性平滑技术，它将每一像素点的灰度值设置为该点某邻域窗口内的所有像素点灰度值的中值，从而消除孤立的噪声值。对于恒定信号来说，中值滤波输出信号呈一致性，因在去除噪声的同时，还可以较为完整地保留图像的边缘细节。

中值滤波的原理如下：

设数列x1,x2,x3,…,xm（m为奇数）为升序排列的数列，则数列的中值记为med(x1,x2,x3,…,xm)。设定序列{}的标准中值滤波器为：

代表自然数集。

片层轮廓提取

使用cv2.findContours()函数提取片层的不规则轮廓。此函数有三个参数，第一个是输入图像，第二个是轮廓检索模式，第三个是轮廓近似方法。返回值有三个，第一个是图像，第二个是轮廓，第三个是轮廓的层析结构。第二个返回值（轮廓）是一个Python列表，其中储存这图像中所有轮廓。每一个轮廓都是一个Numpy数组，包含对象边界点的坐标。

在后续的铺展率计算中，需要圈定每个片层的外轮廓以获得面积、周长等信息，所以此处轮廓检索模式选用cv2.RETR\_EXTERNAL。使用此模式则函数只会返回所有的最外层轮廓，内轮廓都会被忽视。

cv2.findContours()会找到轮廓上的所有点，但并不是全部的点都需要才能确定完整的轮廓。例如，一条线段只需要存储两个端点即可。因此为了减少轮廓中存储的点的数量，选取cv2.CHAIN\_APPROX\_SIMPLE作为轮廓近似方法。此方法会将轮廓上的冗余点去掉，压缩轮廓，从而节省内存开支。

1. **片层实度**

片层实度即喷射粒子实占面积与其所占区域外接圆的比例，是影响热障涂层热导率的重要因素。计算公式表示为：

: 片层的实际面积，: 片层外轮廓的外接圆面积。

片层的实际面积（）即检测出的外轮廓线内片层的面积，计算方法如下：

提取第一个外轮廓，生成一个纯黑画布，将外轮廓画在其上并将其内部填充为白色。

将步骤1)所得图片与去除噪声后的图片与二值化后的图片逻辑与的结果进行与运算，以保留完整的片层内部结构。

计算步骤2)所得图片中白色像素点的数量，即片层的实际面积。

计算片层外轮廓的外接圆面积（）首先需要使用cv2.minEnclosingCircle(cnt)来提取所传入轮廓cnt的最小外接圆。

cv2.minEnclosingCircle(cnt)函数会找到传入轮廓所对应对象的外接圆，返回该外接圆的圆心坐标和半径长度。获得外接圆半径长度后，通过圆面积公式即可计算出外接圆面积。

1. 代码说明
2. EmittingStream类：将控制台输出转为输出在GUI控件内
3. myQGraphicsView类：重载QGraphicsView中的resizeEvent()函数，使图片可以跟随控件大小自动变化。
4. splatGUI类：带GUI界面的程序实现

类内函数：

1. **def** outputWritten(self, text)

功能：修改输出流。

1. **def** setupUi(self, Form)

功能：程序的GUI界面的实现。

1. **def** retranslateUi(self, Form)

功能：设置程序GUI界面中控件上显示的文字

1. **def** openImage(self)

功能：此函数与“选择图片”按钮关联，从本地文件夹选择图片，读入路径

1. **def** openImage(self)

功能：将路径中的图片显示到GUI界面左侧的图片展示框中

1. **def** detect(self)

功能：与“片层检测”按钮关联，检测待测的热障涂层图像，圈出其中的片层轮廓并绘制出最小外接圆，在GUI界面右侧的文本框中输出每个片层对应的实度计算结果。