

数字图像处理课程设计

金属表面缺陷分 类与测量

作者:Faust.Cao

目录

1. 题目内容及要求	2
1.1 题目内容	2
1.2 题目要求	2
2. 题目分析	2
Step. 1: 对比度增强	3
Step. 2: 滤波处理	3
Step. 3: 阈值分割	4
Step. 4: 形态学处理	4
Step. 5: 特征提取	4
Step. 6: 分类训练	4
Step. 7: 缺陷位置大小计算	4
Step. 8: GUI 设计	4
3. 解决问题的方法及原理	5
3.1 对比度增强	5
3.2 Gabor 滤波	6
3.3 阈值分割	8
3.4 形态学处理	10
3.5 特征提取	10
3.5 监督学习与分类	13
3.6 缺陷位置大小计算	14
4. 采用的方案及结果	16
4.1 对比度增强	16
4.2 Gabor 滤波	17
4.3 阈值分割	19
4.4 形态学处理	19
4.5 特征提取	20
4.6 监督学习与分类	20
4.7 缺陷大小位置计算	20
4.8 GUI 设计	21
5. 结果讨论及总结	23
6. 论文评分表	24
参考文献	24
附录	25

金属表面缺陷识别与分类

160212205 第二组 曹恒镇

1. 题目内容及要求

1.1 题目内容

金属板广泛应用于工业生产与生产生活的各方面。由于金属板制造过程涉及到的设备、工艺等多因素的影响，金属板表面容易出现种类较多、形态各异的缺陷，这些缺陷对金属板的耐磨性、抗腐蚀性、电磁特性及美观性都会造成不同程度的影响，最终影响金属板的电磁特性和涂镀效果。因此对于生产金属板的企业来说，表面缺陷检测是必不可少的一个工序，一方面可以通过表面缺陷检测及时检测到缺陷产品，保证所产金属板的质量，维护企业的信誉，另一方面也可以通过分析检测结果及时发现生产过程中存在的问题，并及时解决^[1]。我们将依据网上提供的金属表面缺陷照片数据集为图片来源，构造相应的算法对金属表面的缺陷进行检测，识别，分类与大小测量。

1.2 题目要求

金属表面缺陷识别与分类有以下具体要求：

- (1) 依据金属表面缺陷图片的特性，对图片进行适当的灰度变换(对比度增强与滤波处理)；
- (2) 对金属表面缺陷图片进行全局优化阈值分割；
- (3) 提取二值图片区域边界坐标；
- (4) 对金属缺陷进行特征提取；
- (5) 对金属缺陷进行分类有监督训练；
- (6) 完成对金属缺陷类型的判断与位置大小的计算；
- (7) 完成金属缺陷检测的 GUI 设计。

2. 题目分析

我们查阅相关资料了解到常见的金属表面缺陷主要有如下几种，其分别是：细裂纹(crazing)，表面杂质(inclusion)，斑点(patchs)，麻点(pitted surface)，轧入氧化皮(rolled-in scale)与划痕(scratches)。其分别对应图(a)-图(b)。

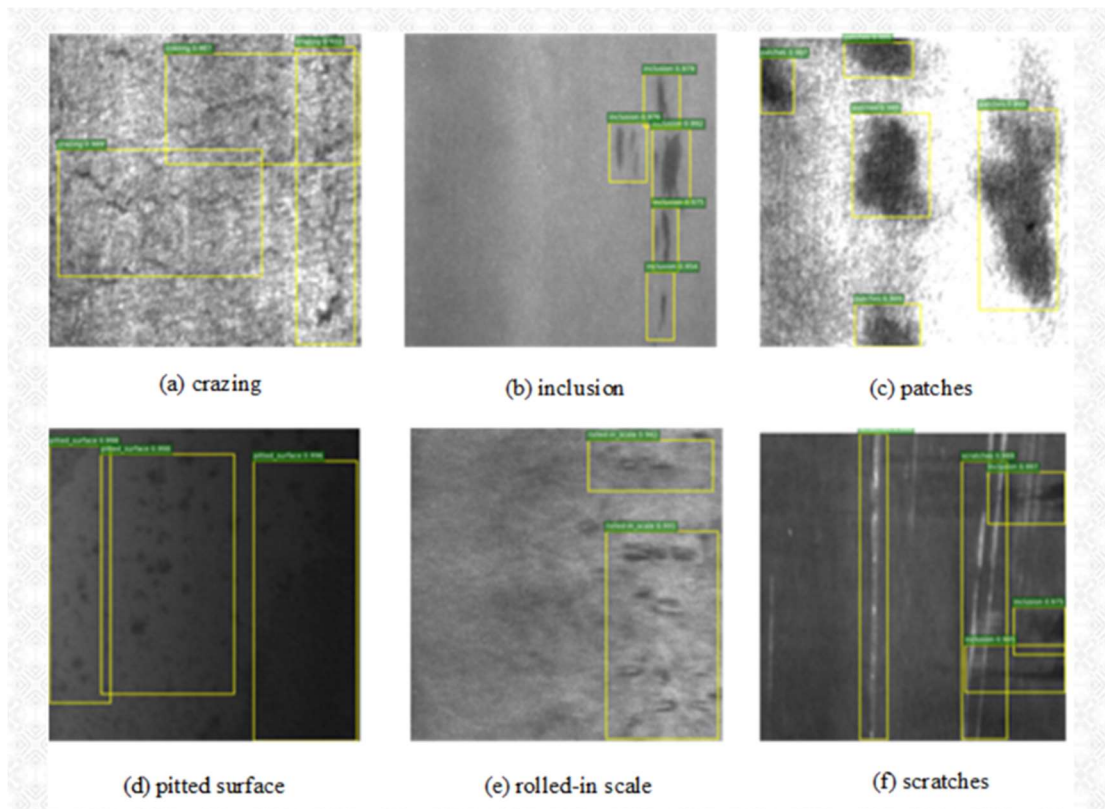


图 2-1 金属表面缺陷分类

我们只考虑其中的三种类型：表面斑点(图 b)，表面轧入氧化皮(图(e))与表面划痕(图(f))。

图像处理的主要步骤有:对比度增强，滤波处理，阈值分割，形态学处理，特征提取，分类训练，缺陷位置大小计算及 GUI 集成操作显示。

Step. 1:对比度增强

由于照明方式等缘故，我们发现原始图像中的金属表面缺陷和整个金属表面背景之间的灰度差较小，灰度范围较小，因而其对比度较低，金属缺陷并不明显，这不利于图像的后期处理，我们需要使用一些方法来增加图像的对比度。常见的对比度增强方法有对数变换，幂律变换，灰度级分层，灰度归一化，对比度拉伸，直方图均衡化，直方图规定化等。考虑到不同照片的背景光强并不相同，所以恒定参数的方法(如对数变换，幂律变换，灰度级分层，对比度拉伸)并不对所有图片适用，故我们考虑使用灰度归一化和直方图均衡化。

Step. 2:滤波处理

拍摄照片时，在采集过程将会不可避免的引入各种噪声，包括高斯噪声和椒盐噪声等。同时，金属表面本身就具有一定的纹理。噪声和纹理将一定程度上将金属表面缺陷掩盖，从而导致会提取中产生错误，我们需要先对图像进行滤波处理以减少后续误检测。滤波方法主要分为空域滤波与频域滤波，且二者的关系满足卷积定理。其中空域滤波主要有均值滤波，中值滤波，高斯滤波与双边滤波；频率滤波注意有傅里叶变换滤波，离散余弦变换滤波与小波

变换滤波。这些算法虽然可以减少噪声，但是也会滤除部分细节信息，导致某些缺陷细节的丢失。所有我们需要寻找新的滤波算法来实现。

Step. 3: 阈值分割

我们得到滤波图像后，还需要进行二值化阈值处理，从而恰当提取出缺陷信息。传统的阈值分割需要确定阈值，但对于不同复杂的金属缺陷图片，我们无法对所有的图片使用相同的阈值，我们拟使用全局优化阈值分割，自动确定最佳的阈值。

Step. 4: 形态学处理

受限于原图质量与图像阈值分割算法，二值图像中某些应该连通的区域可能被阈值分割了，例如划痕缺陷由于在某处灰度值较低而被错误分割，导致一条划痕可能会被分离为多个线段。故我们需要进行闭运算来连接临近物体。同时，图片中某些缺陷较大(例如划痕宽度较大)，在后续边缘检测后，一条直线的两个边缘相距较大，直线提取算法会误将同一根线作为两根相互平行划痕。故我们还需要对缺陷进行细化操作。

Step. 5: 特征提取

该部分可以使用两种方法，一种是基于通用的特征提取算法(例如 HOG, LBP, Haar 算法等);另一种是基于我们所要检测的特定金属缺陷进行特定的特征来直接对划痕进行分类。我将分别对这两种方法进行测试。

Step. 6: 分类训练

如果我们使用了通用的特征提取算法，我们将要对其进行分类监督训练，常用的监督学习算法包括:K 临近算法(KNN),朴素贝叶斯算法,决策树算法，支持向量机(svm)，逻辑回归等。由于支持向量机具有优秀的泛化能力，且在小样本训练集能够得到比其他算法好的多的效果^[6]，我们拟选择支持向量机算法。

Step. 7: 缺陷位置大小计算

我们提取到缺陷后，需要不同特征来表征缺陷的大小与位置。对与‘轧入氧化皮’，我们使用‘轧入氧化皮’个数来表征缺陷大小，用其质心表征其位置；对于斑点，我们使用斑点中的面积占整个图片的比例来表征缺陷大小。对于划痕，我们使用划痕的两个端点来表征缺陷位置，用划痕长度标准缺陷大小。

Step. 8: GUI 设计

为了方便用户使用，我们设计了 GUI 界面，能够选择电脑中的图片，且能在点击‘图像处理’后

显示提取到的缺陷信息(类型与位置大小),对于划痕长度,我们能够让用户输入相机的焦距与物距长度,从而根据几何光学知识计算出实际划痕大小,并将每个划痕的像面大小与实际大小显示出来。同时 GUI 能够让用户通过按键前往数据集的下载网站。

3. 解决问题的方法及原理

3.1 对比度增强

3.1.1 灰度归一化

灰度归一化可使使一幅图像像素的灰度值分布在 0~255 之间,避免图像对比度不足(图像像素亮度分布不平衡)从而对后续处理带来干扰。其基本原理如下所示。

$$I_{out} = \frac{(I_{in} - I_{min}) \times 255}{I_{max} - I_{min}} \quad (1)$$

式中, I_{in} ——原始图像的灰度值;

I_{min} ——原始图像的最小灰度值;

I_{max} ——原始图像的最大灰度值;

I_{out} ——灰度归一化后图像的灰度值。

3.1.2 直方图均衡化

直方图均衡化是图像处理领域中利用图像直方图对对比度进行调整的方法。这种方法通常用来增加许多图像的局部对比度,尤其是当图像的有用数据的对比度相当接近的时候。通过这种方法,亮度可以更好地在直方图上分布。这样就可以用于增强局部的对比度而不影响整体的对比度,直方图均衡化通过有效地扩展常用的亮度来实现这种功能。我们尝试使用普通的直方图均衡化算法,发现其效果较差。其与原图相比效果如下:

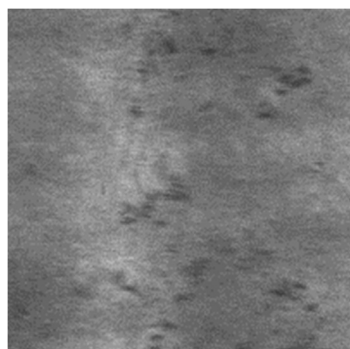


图 3-1-1 原始图片

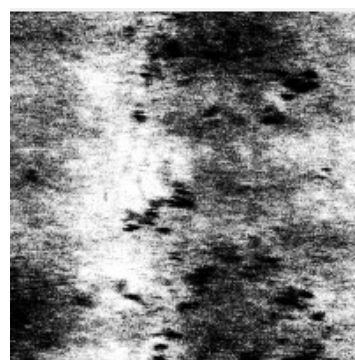


图 3-1-2 直方图均衡化后的图

可见，轧入氧化皮缺陷对比度提高的同时，其背景的对比度也被放大，导致背景中偏黑的部分变得与轧入氧化皮灰度相同，难以将其提取。传统直方图均衡化均衡整个图片的直方图分布，这种均衡化对于灰度对比度较小的部分效果较好，但对于灰度对比度较大的部分效果较差。我们发现上图中没有缺陷部分(低对比度)的对比度得以提升，但是对于含有缺陷的部分(高对比度)，其对比度增加并不明显。故我们考虑使用自适应直方图均衡化(AHE)，它通过计算图像每一个显著区域的直方图，来重新分布图像的亮度值，因此它更适合于用来改善图像的局部对比度，以及增强图像边缘信息，利于分割。但是，自适应直方图均衡化(AHE)有一个缺陷，就是它在增强对比度的同时也会增强图像同质（均匀）区域的噪声，以 AHE 为基础的对比度受限的自适应直方图均衡化(CLAHE)可以有效降低这种噪声的增强。CLAHE 中，每一个像素邻域都要进行对比度限制^[4]。其具体做法(见图 3-1-3)是对原始的直方图分布设定一个阈值，将大于阈值以上的部分剪切掉(钳位)，将剪切部分的先进行均衡化处理，再将剪切部分的均衡化后叠加到剩余的直方图分布上(即给直方图加了一个‘直流偏置’)，之后在对新的分布进行均衡化处理，这样就能减少噪声的影响。由于原始图像被分割成若干个不重叠的子图像，经过上面过程处理后的各个子图像分解处将会出现明显的灰度不连续现象（有明显的分块），为了消除这种不连续分块现象，我们只对于每个区域中央部分的像素进行均衡化处理，对于边缘像素，我们依据其四邻域的像素点灰度值对其进行双线性插值以消除这种“分块”效应^[5]。同时这一过程减少了均衡化函数的计算次数，所以也提高了计算速度。

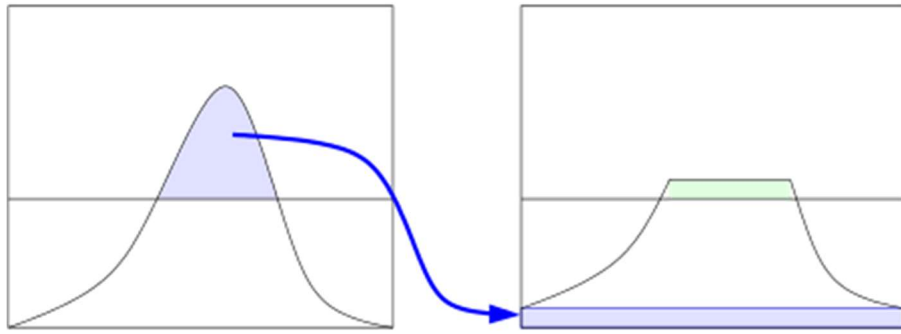


图 3-1-3 对比度受限自适应直方图均衡化过程(CLAHE)

3.2 Gabor 滤波

传统的滤波算法，例如均值滤波，高斯滤波等往往无法在特定的应用领域产生较好的效果，争对特定的应用领域，我们需要查询相关的文献资料了解合适的特殊滤波算法。我们通过查询相关资料了解到在纹理提取领域常用的滤波器是 Gabor 滤波器。Gabor 滤波主要基于小波分析理论，是由一个高斯函数与复指数函数相乘的母小波经过尺度变换与旋转变换后得到的。研究发现，Gabor 滤波器十分适合纹理表达和分离。在空间域中，一个二维 Gabor 滤波器是一个由正弦平面波调制的高斯核函数^[9]。其复数表达式为^[10]：

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right)$$

其可以分为实部与虚部，其中实部可以对图像进行平滑滤波，虚部可以用来边缘检测，故我们使用该滤波卷积核能够同时进行滤波平滑操作与纹理边缘锐化，可谓一举两得。其滤波后的图像为：

$$S(x,y)=\sqrt{(g_R * I(x,y))^2 + (g_i * I(x,y))^2}$$

其中， x,y 为像素坐标。

$$x' = x \cos \theta + y \sin \theta$$

$$y' = -x \sin \theta + y \cos \theta$$

Gabor 核中各参数意义如下：

1) λ 为调制正弦函数的波长，其决定了卷积核有效非零区域占整个核的比例，不同 λ 可以提取图像不同频率分量信息，一般 $\lambda > 2$ 。

下图显示相同卷积核不同 λ 对应的时域卷积核图像：

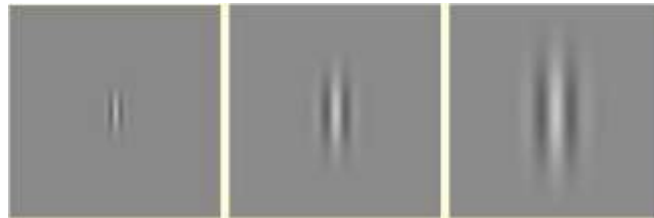
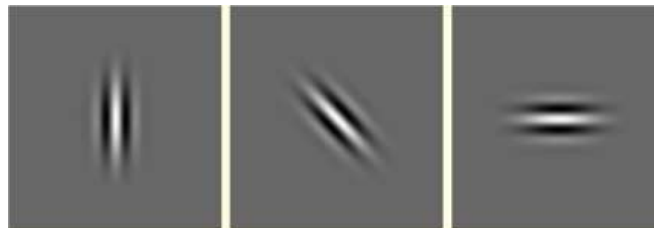


图 3-2-1 不同 $\lambda = 5, 10, 15$ 下的卷积核图像

2) θ 体现了 Gabor 滤波器平行线方向，该参数将决定 Gabor 滤波的滤波方向选择特性，即对不同方向上的锐化平滑效果并不相同。 $\theta \in [0, 360)$

下图显示相同卷积核不同 θ 对应的时域卷积核图像：



3-2-2 不同 $\theta = 0, 45, 90$ 下的卷积核图像

这两个参数将决定 Gabor 滤波器的方向与频率选择特性。

其他参数还有：

3) 相位偏移量 ϕ ，4) 空间方向比例因子 γ ，5) 频带宽度 $b(\sigma = b\lambda)$ 。

Gabor 滤波器有如下优点：

- (1) 对于光照变化不敏感,能够提供对光照变化良好的适应性。所以不同金属检测硬件系统由于光源差异而导致的背景光强不等时，它都能够有较好的滤波效果，具有较强的鲁棒性。
- (2) 与传统的傅立叶变换相比，Gabor 小波变换具有良好的时频局部化特性。即非常容易地调整 Gabor 滤波器的方向、基频带宽及中心频率从而能够最好的兼顾信号在时空域和频域中的分辨能力^[12]。
- (3) Gabor 滤波具有多分辨率特性即变焦能力。即采用多通道滤波技术，将一组具有不同时

频域特性的 Gabor 小波应用于图像变换，每个通道都能够得到输入图像的某种局部特性，这样可以根据需要在不同粗细粒度上分析图像^[17]。考虑到我们的不同的金属缺陷大小差异较大(较小的轧入氧化皮,较大的划痕),对应于图像中的不同的频率分量，我们能够通过改变调制正弦的波长来提取不同的缺陷。

- (4) Gabor 滤波器可以通过改变 θ 来对不同方向进行滤波，从而能够判断其纹理是否具有方向各向异性，从而能够据此在一定程度上判断缺陷的种类(划痕有方向各向异性，斑点与轧入氧化皮方向各向同性)。
- (5) Gabor 滤波可以在滤波除噪的同时，增强边缘以及峰、谷、脊轮廓等图像特征(虚部的锐化)，这相当于增强金属表面缺陷信息，它的小波特性说明了 Gabor 滤波结果是描述图像局部灰度分布的有力工具,因此,可以使用 Gabor 滤波来抽取图像的纹理信息。

综上所述，处理金属表面缺陷纹理，Gabor 滤波是一个很好的选择，它的滤波特性与表面缺陷特性相匹配。故我们选择 Gabor 算子进行空域滤波。

3.3 阈值分割

阈值化分割是一种传统的最常用的图像分割方法，因其实现简单、计算量小、性能较稳定而成为图像分割中最基本和应用最广泛的分割技术。为了能够更好的提取出表面缺陷，恰当的阈值分割算法至关重要，阈值分割常用的方式有全局阈值法，局部阈值法和多阈值分割法。考虑到我们要提取缺陷，我们只考虑单阈值分割算法^[8]。且我们的算法能争对不同图像有不同的阈值，所以我们只能使用动态变阈值算法。我们考虑全局阈值分割中的最大类间方差法（OTSU）和最大熵法，对于局部阈值分割，我们考虑 Niblack 算法。用缺陷图像实际测试三者的效果并比较，选择效果最好的一种。

3.3.1 最大类间方差法（OTSU）

最大类间方差法按图像的灰度特性,将图像分成背景和目标两部分。背景和目标之间的类间方差越大,说明构成图像的两部分的差别越大,当部分目标错分为背景或部分背景错分为目标都会导致两部分差别变小。因此,类间方差最大的分割意味着错分概率最小。其主要算法如下:将阈值从 0 开始一直逐次增大到 255(每次加 1)，按阈值将图像分为背景(0)与前景(1)。

对于每个阈值 $T(0 \leq T \leq 255)$ ，进行如下计算:

- 1) 计算背景和前景像素点各占总像素点的比例 w_0, w_1 ;
- 2) 计算背景和前景像素点的灰度均值 u_0, u_1 ;
- 3) 计算全部像素点的灰度均值 $u = w_0 u_0 + w_1 u_1$;
- 4) 计算此时二值图的类间方差: $\sigma^2 = w_1 (u - u_1)^2 + w_0 (u - u_0)^2$ 。

选择使类间方差最大的阈值 $T(0 \leq T \leq 255)$ ，此时的 T 即位所求最佳阈值。

我们通过不断迭代遍历 $T(0 \leq T \leq 255)$ ，以找到该值。

3.3.2 最大熵法

最大熵算法是另一种基于图像统计信息的全局阈值分割算法，其原理基于统计中的信息熵，信息论告诉我们一个系统不确定度(信息量越大)越大，其熵越大^[13]。故我们使用最大熵法保证图像类间熵最大来保证二值图包含尽量多的前景信息（缺陷）。我们考虑其中的 KSW 算法^[9]，其总体思路仍是遍历阈值：

将阈值从 0 开始一直逐次增大到 255(每次加 1)，按阈值将图像分为背景(0)与前景(1)。

对于每个阈值 $T(0 \leq T \leq 255)$ ，进行如下计算：

1) 计算背景和前景中各灰度值在背景和前景中概率分布 p_i ；

2) 计算背景和前景中的 $\frac{p_i}{p_n}(0 \leq i \leq T), \frac{p_j}{1-p_n}(T+1 \leq i \leq 255)$;

其中 $p_n = \sum_{i=0}^T p_i$ 为背景中的累计分布。；

3) 计算背景和前景的熵 $H_1(T) = -\sum_{i=0}^T \frac{p_i}{p_n} \ln \frac{p_i}{p_n}, H_2(T) = -\sum_{j=T+1}^{255} \frac{p_j}{1-p_n} \ln \frac{p_j}{1-p_n}$

4) 计算该二值图像的类间熵 $H(T) = H_1(T) + H_2(T)$ 。

选择使类间熵最大的阈值 $T(0 \leq T \leq 255)$ ，此时的 T 即位所求最佳阈值。

我们通过不断迭代遍历 $T(0 \leq T \leq 255)$ ，以找到该值。

3.3.3 Niblack 法

NiBlack 法是一种简单有效的局部动态阈值算法，这种算法的基本思想是，对图像中的每一个点，在它的 $r \times r$ 邻域内，计算邻域里像素点的均值和方差，然后用下式阈值进行二值化^[7]：

$$T(x, y) = m(x, y) + k * s(x, y)$$

其中，对于每一个像素点 (x, y) ， $T(x, y)$ 为该点的阈值。 $m(x, y)$ 为该点 $r \times r$ 邻域内像素点的均值。 $s(x, y)$ 为该点的 $r \times r$ 邻域内像素点的标准方差； k 为修正系数。此方法的优点是能够很好的针对于单个像素进行处理，缺点是处理速度慢，而且没有考虑到边界问题，且需要预设调节参数 k 。

3.4 形态学处理

3.4.1 闭运算

对于二值化后金属表面缺陷，我们发现一些本应该连通的区域（例如因划痕在某些地方较浅）在二值化过后存在一些间隙^[7]，我们需要进行闭运算(即先进行膨胀，再进行腐蚀操作)来弥合小狭缝并保证总体形状不变，以避免同一个划痕因为中间有部分断裂而导致被检测为多个分开的独立的划痕。

3.4.2 细化操作

由于有的金属表面划痕较宽，导致经过闭运算后的划痕还有一定的宽度，这将导致之后对缺陷进行边缘提取时，划痕的左右(上下)边缘之间的距离较大，在后续进行检测时会被误检测为两个独立的划痕，即导致左右边缘距离和两个划痕间距相近，导致我们无法判断划痕总共有2条粗划痕还是有3个划痕(1个粗划痕，2个细划痕),还是4个划痕(4个细划痕)。同样可能把两个独立的细划痕判为一个粗划痕。如下图划痕所示：

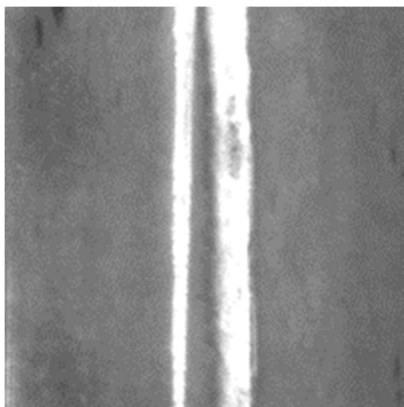


图 3-4-1 可能将粗划痕判为几个细划痕

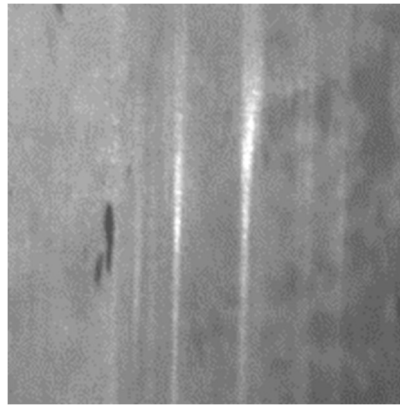


图 3-4-2 可能将两个细划痕判为一个粗划痕

故我们需要对划痕进行细化处理，将划痕变窄，同时，又要保证细化作用过度导致细划痕的丢失。

3.5 特征提取

在次，我们使用两种截然不同的思路。一种思路是基于通用的特征提取算法提取缺陷的特征，之后对这些缺陷特征进行监督学习训练，保存训练的结果，当输入测试图像时，计算其特征后放入已经训练完成的分类器中直接缺陷分类。该方案较为复杂，涉及一些高级算法。另一种是基于已给图片集，发现不同缺陷的显著可以用来区分的特征差异，直接根据此特征进行简单判断分类，该方法简单可行，且精度也不低。下面我们将分述这两种方法。

3.5.1 简单特征与识别

我们要测试的缺陷分为三种:轧入氧化皮，斑点与划痕。其三者直观比较可见图 2-1。我们通过观察总结发现三种的直观差异如下：

首先是轧入氧化皮与斑点和划痕不同。我们发现，单个轧入氧化皮是一个小斑点，其所占大

小都很小，而斑点和划痕的都较大。故我们可以提取二值化的缺陷的边界，计算每个边界集合的大小，统计其中边界最长的那个，看其是否达到阈值，若大于阈值，我们认为其属于斑点或划痕，若小于阈值则认为其为轧入氧化皮。这样，我们就可以识别出轧入氧化皮。

接着，是斑点与划痕的不同，我们发现，斑点相比于背景偏黑，其平均灰度值小于图像背景灰度均值;而划痕恰恰相反，其相比于背景偏亮，其平均灰度值大于图像背景灰度均值。故我们可以进行如下操作:

将二值图与原灰度图进行掩模操作，提取中其中非零像素的灰度均值，则该均值即为缺陷的均值;之后将二值图取反，再次重复上述掩模求均值过程即可得到背景的灰度均值，将二者进行大小比较即可确定缺陷是斑点还是划痕。

其流程图如下:

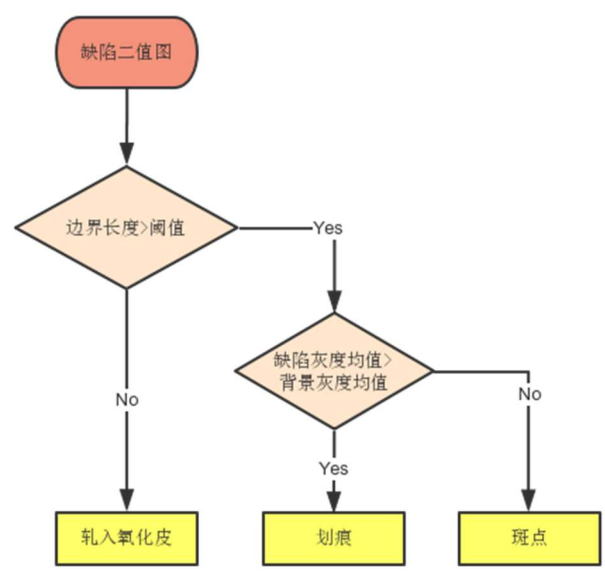


图 3-5-1 简单缺陷分类算法

3.5.2 通用特征提取算法

另一种思路是采用通用的特征提取算法，提取图像的深层信息。该方法能够更适用于一般的金属缺陷，但是算法较为复杂。常见的特征提取算法包括 HOG，LBP 与 Haar 等。我们通过查阅相关文献，了解它一种用来描述图像局部纹理特征的算子，且其改进的 LBP 算法具有较好的旋转不变性与灰度不变性。其中旋转不变性说明即使不同划痕或斑点的方向不同，其提取的特征仍是相近的(不受方向的影响),这可以提高后续的分类精度;而灰度不变性则可以保证不同光源硬件条件下其缺陷特征是相近的，因而其抗环境干扰性较强，应用范围更广。下面简述其原理:

传统的 LBP 算法是使用一个正方形区域，通过将邻域像素的灰度值与中间元素进行比较大小若大于中心灰度，则将该邻域像素置 1，若小于中心灰度，则将该邻域像素置 0。最后将这几个邻域元素从左上角顺时针排列构成一个二进制序列，再将该二进制转化为十进制，则该十进制数即为该区域类的特征，对整幅图像进行该操作，最后得到一个序列的数，则该序列即为该图像的 LBP 特征向量。

基本的 LBP 算子的最大缺陷在于它只覆盖了一个固定半径范围内的小区域，这显然不能满足不同尺寸和频率纹理的需要。为了适应不同尺度的纹理特征，并达到灰度和旋转不变性的要求，我们参考了改良的 LBP 算法^[15]。

改良的 LBP 算法使用圆域代替方形区域，且其半径可变，我们也可以设置在圆域边缘上的点

的个数，这些边缘点均匀分布在圆上，图 3-5-2 显示不同半径与边缘点个数的圆域：

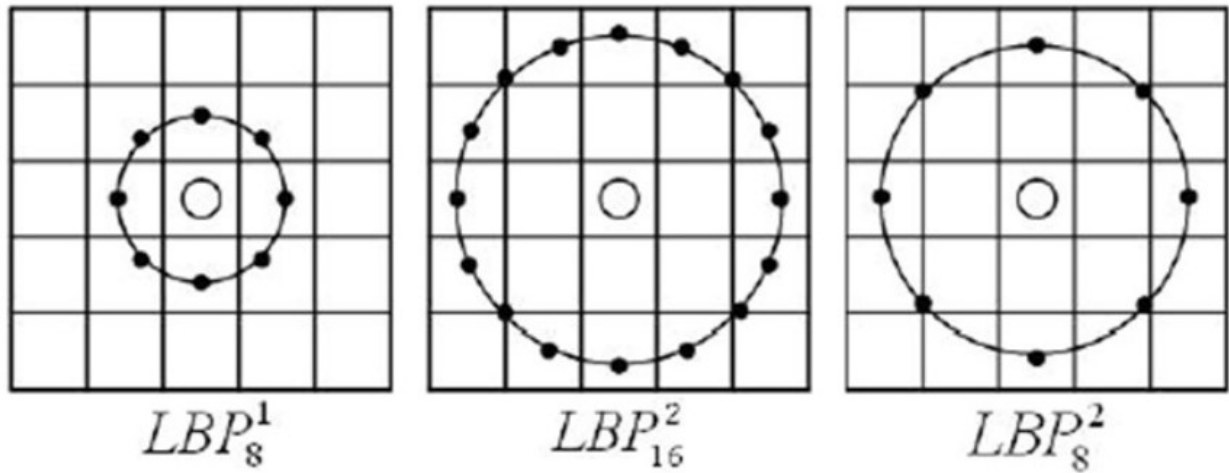


图 3-5-2 不同圆域与边缘点的 LBP 算子^[15]

我们发现选择圆域将会导致部分边缘点没有恰好位于像素中心处，则这些点的灰度值可以通过与其相接四个像素点进行双线性插值获得其灰度值。我们同时选择三种具有不同半径与边缘点个数的圆域，从而实现了不同尺度的特征提取。分别得到三组灰度值序列，之后我们对这三组序列进行一维离散快速傅里叶变换，得到各自的频谱图，这个频谱与原灰度序列互为傅里叶变换关系，我们考虑图像发生一定的偏转，这意味着灰度的序列发生了向左/右平移，由傅里叶变换的性质可知：

$$F[f(x-x_0)] = f(v)e^{-i2\pi x_0}$$

即灰度序列的移为操作并不会影响其对应的频谱分布(取复数的模)，应而具有了旋转方向不变性。图像中的噪声在图像的频谱中一般是高频分量，我们在提取图像特征时并不需要这些高频信息，故我们让频率序列通过一个低通滤波器滤去噪声。最后，我们按圆域半径大小按次序将三个频率序列拼接在一起即得到了该圆域的特征，我们对图像其他地方做同样的处理即可得到整个图像的特征信息^[14]。图 3-2-3 显示了上述过程。该方法并不完全是 LBP 算法，而是在其基础上的改进。

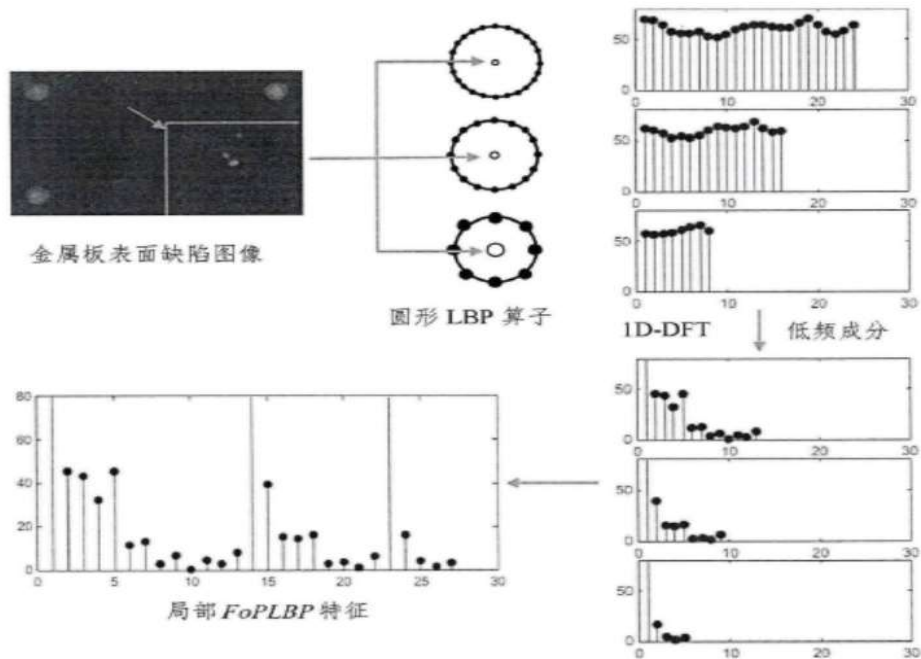


图 3-5-3 改进的 LBP 算法原理说明图^[14]

3.5 监督学习与分类

监督学习是在使用通用特征提取算法后进行了，我们将提取到的每一幅图像的特征即其标签放入分类器中进行训练，之后用新的图像输入测试其正确率用以验证其训练可靠性。我们使用支持向量机这个适合小样本进行训练的监督学习算法。支持向量机(SVM)为机器学习和多维机构风险最小化理论原则上建立的一种新型专门研究小样本分类的机器学习理论。该方法在解决非线性小样本的高维识别问题时，具有很强的适应性，并且能推广使用到其他数学模型中^[16]。

支持向量机((SVM)是在结构风险最小化理论与统计学习理论原则上提出的。支持向量机((SVM)在有限样本的基础上，寻找函数复杂程度和分类性能的最佳平衡，以求最佳的分类性能，其主要优点有：

- 1、支持向量机是针对有限样本统计设计的，在有限信息的情况下，具有很好的分类性能。
- 2、支持向量机算法引入核函数，能有效解决非线性分类中的维度问题。
- 3、支持向量机将分类问题转化为凸二次函数最值问题，所以，就理论上说，模型最终可得到全局最优解。

下面简述其原理：

一般 SVM 有下面三种：

- 1) 硬间隔支持向量机（线性可分支持向量机）：当训练数据线性可分时，可通过硬间隔最大化学得一个线性可分支持向量机。
- 2) 软间隔支持向量机：当训练数据近似线性可分时，可通过软间隔最大化学得一个线性支持向量机。
- 3) 非线性支持向量机：当训练数据线性不可分时，可通过核方法以及软间隔最大化学得一个非线性支持向量机。

给定训练样本集 $D=\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ $D=\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, $y_i \in \{+1, -1\}$,

i 表示第 i 个样本， n 表示样本容量， x_i 为第 i 张图的特征向量， y_i 是第 i 张图的标签。分类学

习最基本的想法就是基于训练集 D 在特征空间中找到一个最佳划分超平面将正负样本分开，而 SVM 算法解决的就是如何找到最佳超平面的问题。

超平面可通过如下的线性方程来描述：

$$\vec{w}^T \vec{x} + b = 0$$

我们要保证不同分类的两组数据被该超平面完全分开，且尽量保证数据尽量远离该超平面以保证分类器最大的鲁棒性，我们需要寻找参数 w, b 的最优值使得特征向量到超平面的距离尽量大，其因满足如下的约束条件：

$$y_i(\vec{w}^T \vec{x} + b) \geq 1$$

经过分析我们得到其最大值的条件是：求解 $\frac{2}{\|\vec{w}\|}$ 的最大值对应的 w, b 。

我们可以通过拉格朗日对偶将该问题简化为求解如下的约束最值问题：

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \vec{x}_i^T \vec{x}_j \\ \text{s.t.} \quad & \alpha_i \geq 0, i = 1, 2, \dots, n \\ & \sum_{i=1}^n \alpha_i y_i = 0 \end{aligned}$$

求解出 a 后即可求出对应的 w, b 。

之后使用 SVM 算法所满足的 Karush-Kuhn-Tucker (KKT) 条件将其转化为二次规划问题，在使用一些高效算法 (例如 Sequential Minimal Optimization (SMO) 算法) 进行最优化问题求解 [18]。

以上方法仅仅适用于第一种 SVM 算法，即硬间隔支持向量机 (线性可分支持向量机)。在现实任务中很难找到一个超平面将不同类别的样本完全划分开，即很难找到合适的核函数使得训练样本在特征空间中线性可分。退一步说，即使找到了一个可以使训练集在特征空间中完全分开的核函数，也很难确定这个线性可分的结果是不是由于过拟合导致的。解决该问题的办法是在一定程度上运行 SVM 在一些样本上出错，为此引入了“软间隔” SVM 算法与非线性 SVM，限于篇幅，我们不再介绍。

3.6 缺陷位置大小计算

3.6.1 轧入氧化皮位置大小计算

考虑到轧入氧化皮的大小较小且数量较多，其边界难以严格规定，我们不计算其大小，而是统计检测到的个数。我们提取二值化后的图像的边界信息，统计边界元胞数组的元胞个数来获得轧入氧化皮的个数，由于必然存在一些缺陷无法被检测到，我们用‘至少’来做保守估计。

对于轧入氧化皮的位置计算，我们对每个提取到的边界与原图像进行掩模后使用质心计算公式计算其对应的灰度质心中心作为其中心。公式如下：

$$g_i = \frac{\sum_{i=0}^M \sum_{j=0}^N ic(i, j)}{\sum_{i=0}^M \sum_{j=0}^N c(i, j)}, g_j = \frac{\sum_{i=0}^M \sum_{j=0}^N jc(i, j)}{\sum_{i=0}^M \sum_{j=0}^N c(i, j)}$$

3.6.2 斑点位置大小计算

对于斑点，我们只计算斑点的面积占整个金属表面面积的比值，我们对而主题中的为 1 的部分计算其总的像素面积处于图片的纵的像素值即为斑点所占比。

3.6.3 划痕位置大小计算

对于划痕，考虑到它是直线，我们需要使用直线检测算法提取其中的直线成分。我们使用传统的霍夫变换直线检测算法，其大体思路如下：

对于过 (x_0, y_0) 的直线系而言，其可以在 XOY 坐标系中表示为： $b = y_0 - ax_0$ ，如果其中过各点

的直线系中有一条参数 a, b 均相等的直线，则可以认为这几个点共线，即检测到了直线。但是使用直角坐标系的参数将可能出现斜率 a 无穷大的情况，因而无法检测竖直线。我们可以把直线放在极坐标系中考虑，则过 (x_0, y_0) 的直线系表达式为^[3]：

$$r = \cos \theta \cdot x_0 + \sin \theta \cdot y_0$$

如下图 3-6-1 所示:

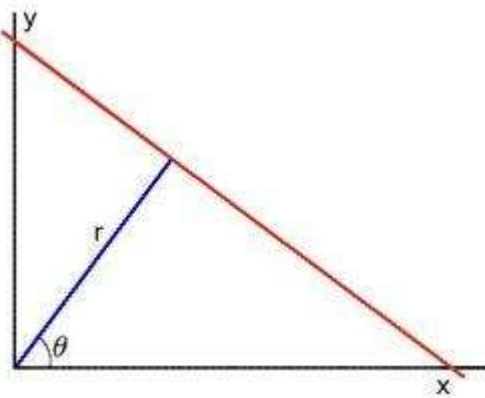


图 3-6-1 直线在极坐标中的表示

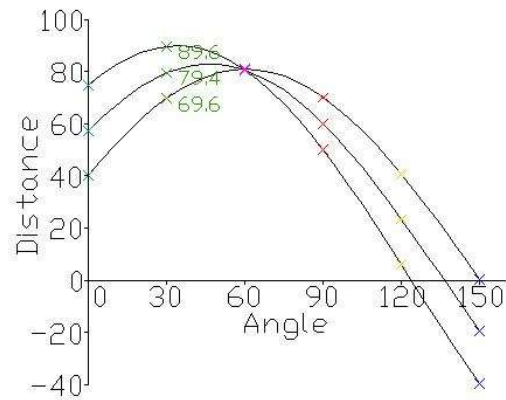


图 3-6-2, 过不同点的直线系的 $\theta-r$ 曲线

同样的, 如果其中过各点的直线系中有一条参数均相等的直线, 则可以认为这几个点共线, 即检测到了直线。使用极坐标系能对所有斜率的直线适用。我们画出 $\theta-r$ 坐标图, 如图 3-6-2 所示, 可以看出三个不同点对应的直线系曲线在某点处交于同一点, 即这三个点是共线的, 若直线系对应的 $\theta-r$ 曲线不相交, 则认为这些点不共线, 即不构成直线。我们对 $\theta-r$ 平面进行遍历, 找到所有曲线有交点的位置, 每一个交点是一条直线, 交点的坐标即为该直线的极坐标参数。

由于划痕并不是严格直线, 所以一跟直线将被霍夫直线检测分为若干个没有完全覆盖划痕的短线, 这些短线会有一定的重合与夹角, 如图 3-6-3 和图 3-6-4 所示:

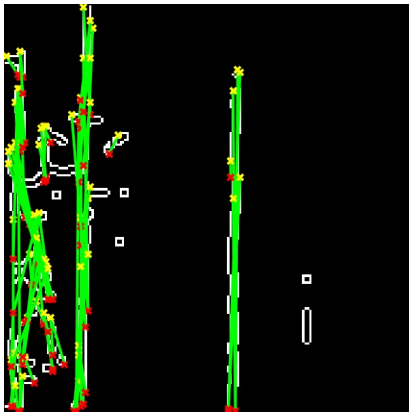


图 3-6-3 竖直线的 hough 直线检测效果

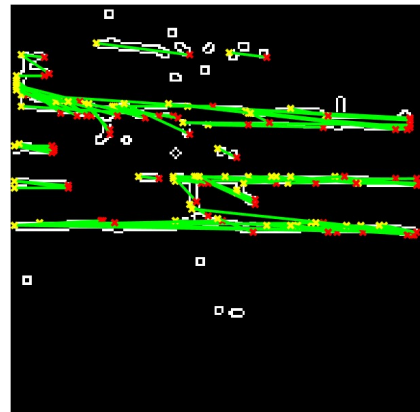


图 3-6-4 水平线的 hough 直线检测效果

我们需要后续处理来减少直线数, 并提取出当中的划痕。

我们的后续算法设计如下:

- 1) 首先我们计算每个检测出的直线长度, 剔除那些过短的线(小于阈值), 这些线往往是误检;
- 2) 计算所有检测直线与 y 轴的夹角, 若夹角小于某个阈值, 则认为是竖线, 否则是横线;
- 3) 计算每根线的中点的坐标;
- 4) 如果直线是竖线, 将直线按中点 x 坐标有小到大顺序排序。如果直线是横线, 将直线按中点 y 坐标有小到大顺序排序;
- 5) 将检测到的第一根线作为直线累计初始状态;
- 6) 对所有直线进行遍历, 对于竖线, 判断相邻两直线的 x 坐标之差是否小于某阈值, 若小于, 则认为该直线与之前的累计直线属于同一根划痕, 并继续进行累加; 若 x 之差大于阈值, 则认为该直线与下一个直线分属于不同的划痕, 则计算前面类间直线端点坐标的最大值与最小值作为划痕的端点, 求其长度并画图, 同时将累计的直线清 0, 重

复上述操作。对于横线，判断标准变为判断相邻两直线 Y 坐标之差是否小于阈值，其他操作同竖线；

7) 将剩余的线统一当作同一个划痕，按步骤 5) 进行。

其算法流程图入图 3-6-5 所示：

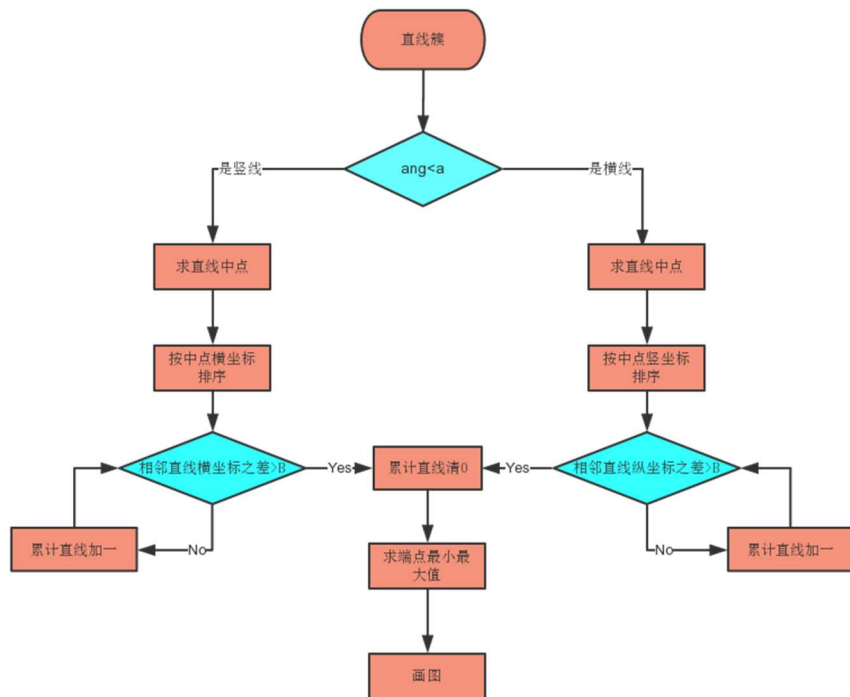


图 3-6-5 算法流程图

4. 采用的方案及结果

4.1 对比度增强

4.1.1 灰度归一化

其结果如下：

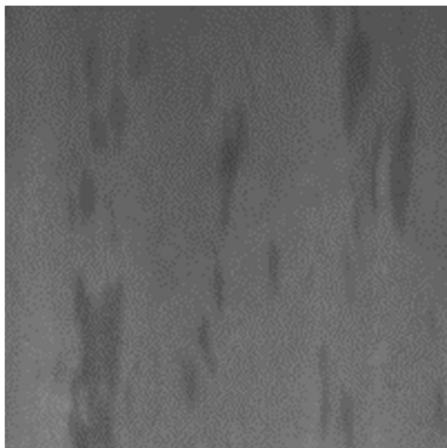


图 4-1-1 原始图像

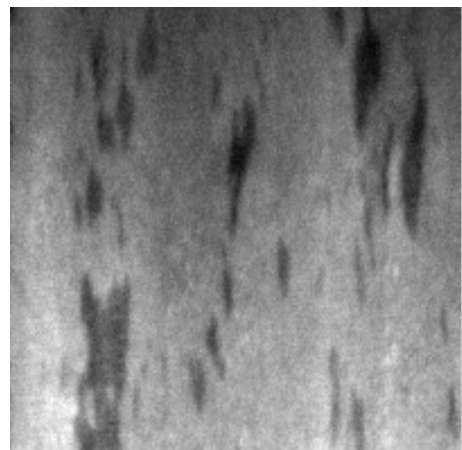


图 4-1-2 灰度归一化的图片

在两图对比中可以看出，原始图像的对比度较低。经过灰度拉伸后，图像的灰度级分布更广,更好地提高了手背与背景的对比度。

4.1.2 对比度受限的自适应直方图均衡化(CLAHE)

对于划痕缺陷，我们发现其即使经过了灰度归一化操作，其对比度仍然不是太高，故我们还需要争对划痕再进行一次对比度受限的自适应直方图均衡化(常规的直方图均衡化效果较差,见上文分析)。

其结果如下：

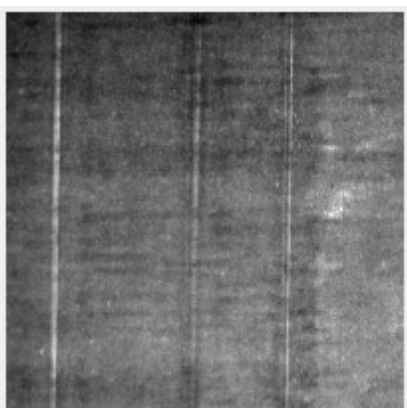


图 4-1-3 仅经过灰度归一化操作图像

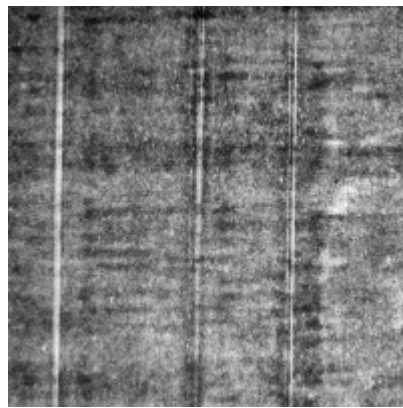


图 4-1-4 再经过 CLAHE 的图像

可见，在经过 CLAHE 后，其对比度更加突出，更易于后期阈值分割。

4.2 Gabor 滤波

我们分别使用 $\theta = 0, 45, 90, 135$ 对三种缺陷进行 Gabor 滤波，其他参数不变为：

$\lambda = 10, \varphi = 0, \gamma = 0.5, b = 0.5$ ，卷积核大小 25×25 。

其各自效果如下：

对于划痕图 4-2-5：

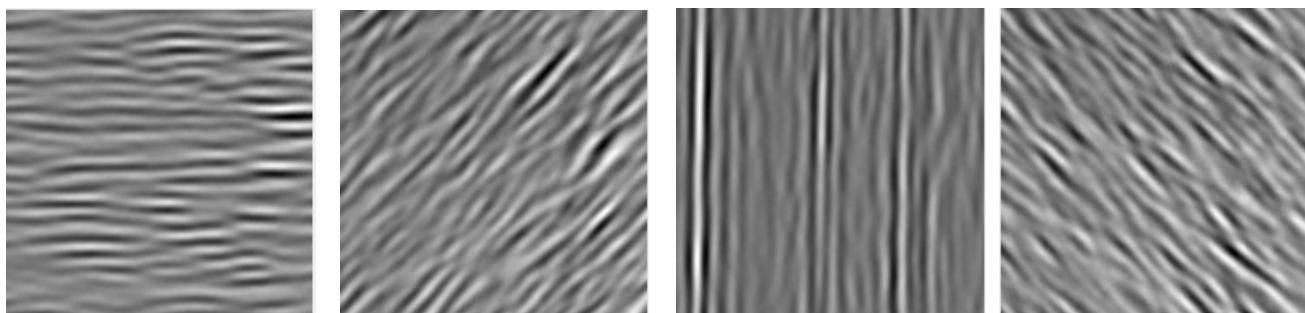


图 4-2-5 $\theta = 0, 45, 90, 135$ 的不同卷积核对划痕的滤波

可见只有 $\theta = 0$ 时滤波后才能显示出有划痕，可见 Gabor 滤波确实具有方向选择性。且当滤波方向与划痕方向一致时，会在划痕边界产生较大的波纹振荡，导致划痕似乎加宽了，故我

们对于划痕不进行 Gabor 滤波，仅对轧入氧化皮与斑点进行 Gabor 滤波。
对于对轧入氧化皮与斑点进行 Gabor 滤波，也会发现其背景处产生沿滤波器方向的较大振荡，为了减少振荡，我们用上述四个角度的滤波器分别滤波，将 $\theta = 90, 135$ 滤波器的相加与 $\theta = 0, 45$ 滤波器的相加做差，则可以抵消 Gabor 滤波器因方向选择性导致的纹波，同时滤除噪声并增强缺陷边界。其 Gabor 滤波器的空域三维图如图 4-2-6 所示：

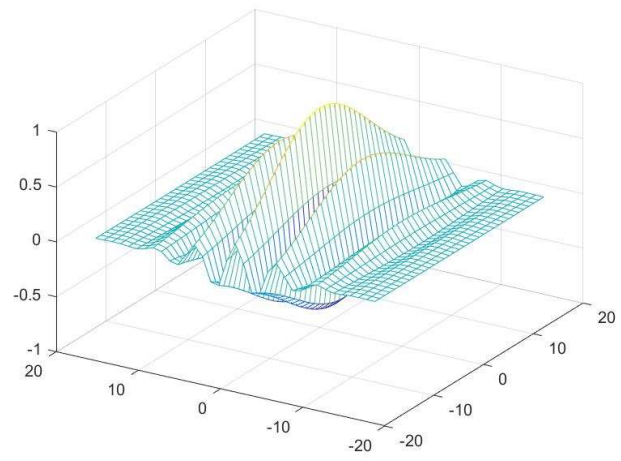


图 4-2-6 Gabor 滤波器的空域三维图

使用上述操作后轧入氧化皮与斑点与原图比较如图 4-2-7 与图 4-2-8 所示：

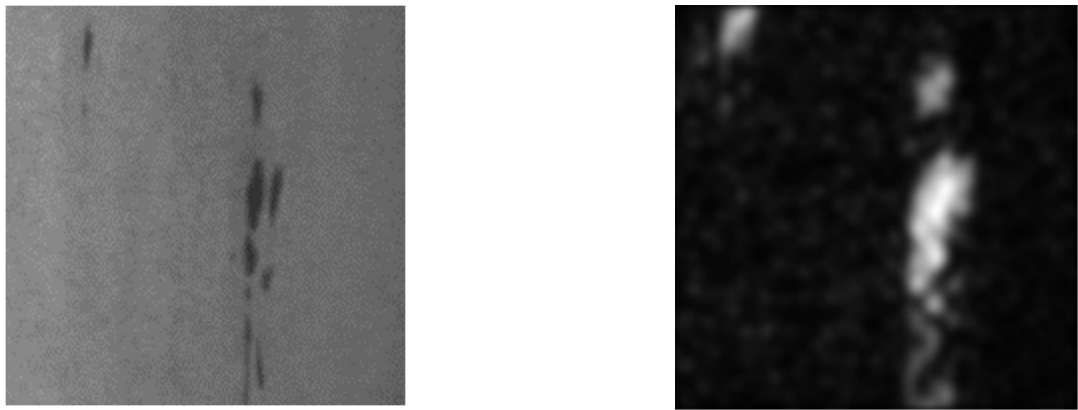


图 4-2-7 斑点原图及其 Gabor 滤波后的图像

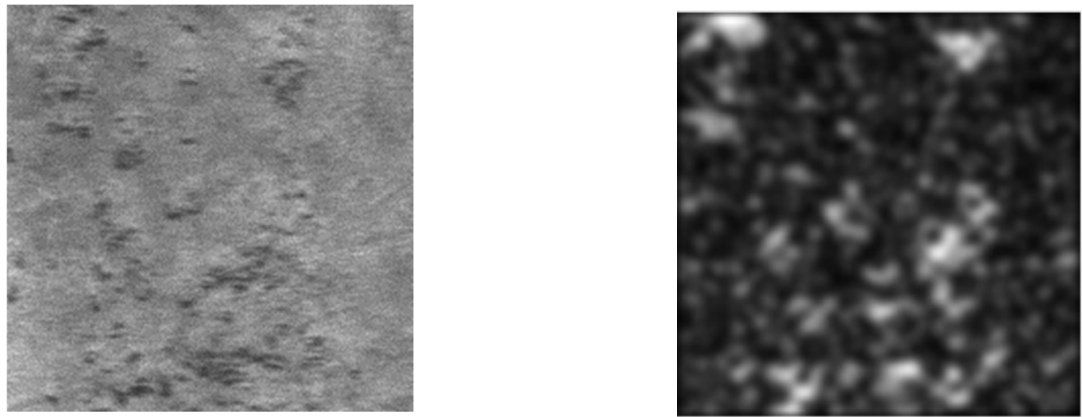


图 4-2-8 轧入氧化皮原图及其 Gabor 滤波后的图像

可见，Gabor 虽然让图片看上去更模糊了，但是却保留了增强边缘以及峰、谷、脊轮廓等图

像特征，同时增加了缺陷与背景的对比度与边缘轮廓信息。

4.3 阈值分割

我们对缺陷分别使用 OSTU(最大类间方差法), 最大熵法与 Niblack 法, 比较其差异: 图 4-3-1 显示了三种阈值分割算法的阈值分割效果: (原始图片见图 4-2-8)



图 4-3-1 三种阈值分割算法对同一张图像的效果

可以看到, OSTU(最大类间方差法)能更完整的显现一些缺陷, 但是也会引入一些图像的噪声, 最大熵法与 Niblack 法的效果相近, 但考虑到最大熵法作为全局阈值分割其计算量远小于 Niblack 法, 综上, 我们选择最大熵法。

4.4 形态学处理

4.4.1 闭运算

对于划痕缺陷, 经过阈值分割的划痕会在中间出现一些间隙, 这对于后期的直线检测是不利的, 我们对其先进行闭运算消除中间的小间隙, 我们选择 5×5 大小的 square 结构元素。其效果如图 4-4-1:



图 4-4-1 原始二值图与经过闭运算后的二值图

可见左边那条原先被分为三段的划痕经过闭运算后中间的狭缝被填补了, 划痕也变得连续了。

4.4.2 细化操作

前文已经阐述对于过粗的划痕，不使用细化所产生的不良影响，我们使用 MATLAB 自带的形态学处理函数'bwmorph'函数，参数选择为'thin'，对其进行 5 次重复操作。其效果如图 4-4-2 所示：



图 4-4-2 未细化划痕的二值图与细化后的划痕二值图

4.5 特征提取

若我们使用第一种方法，即提取缺陷的简单特征，我们首先提取二值图的轮廓，看轮廓大小是否大于阈值，若小于则认为是较小的轧入氧化皮缺陷，若大于则认为是斑点或划痕。之后我们将得到其掩模图像，从而计算其中的非零元素的均值，再对背景进行掩模，再次计算均值，将二者比较即可判断是斑点还是划痕。若我们使用第二种方法，即提取缺陷的深度特征信息，我们对检测到缺陷的位置求其最大外界矩形，将该矩形内部灰度图像构成新图像，求其该图像的特征并变为一个向量，最后我们得到一个大小为 600*18 的矩阵(600 张图片)。该矩阵将连通其标签放入分类器中进行训练。

4.6 监督学习与分类

我们使用支持向量机对 4.5 中得到的特征向量与标签进行分类训练，取其中 540 张图用于训练，将剩余的 60 张图用于测试，经过测试，我们发现其正确率为 93.3%，正确率较高。

4.7 缺陷大小位置计算

4.7.1 轧入氧化皮缺陷大小位置计算

对于轧入氧化皮，我们按前述方法对其进行位置定位并用蓝色星号标注其质心，其结果如图 4-7-1 所示：

共统计到 16 个轧入氧化皮缺陷

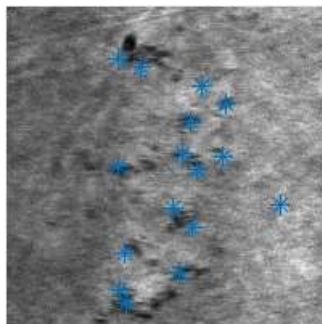


图 4-7-1 轧入氧化皮检测结果

4.7.2 斑点缺陷大小位置计算

对于斑点，我们按前述方法对其大小进行计算并用红圈标注其边界，其结果如图 4-7-2 所示：

斑点所占面积为 5.13%

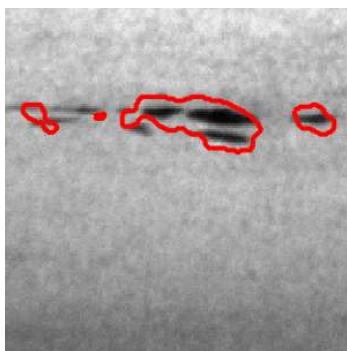


图 4-7-2 斑点检测结果

4.7.3 划痕缺陷大小位置计算

对于划痕，我们使用上述算法得到了各个划痕的起始点坐标与长度，并用叉号标出两端点，用线标出该直线，其结果如图 4-7-3 所示：（假设焦距 2cm,物距 50cm）

一共有四根划痕

第一条：

80 像素，实际 3.84cm

第二条：

172 像素，实际 8.26cm

第三条：

169 像素，实际 8.09cm

第四条：

173 像素，实际 8.31cm

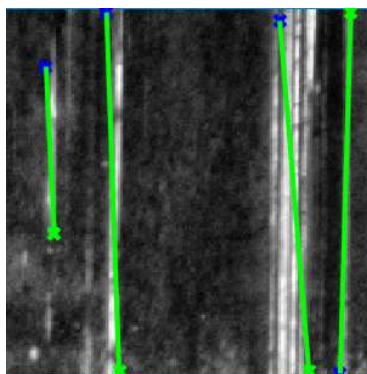


图 4-7-3 划痕检测结果

4.8 GUI 设计

为了方便操作与展示，我们开发了金属缺陷分类与测量页面，其包括两个图像显示窗口分别显示原始图线与标注检测到的缺陷位置大小的图片，2 个按键分别用于从文件中导入灰度图与进行图像处理，两个用户输入窗口分别让用户输入相机的焦距与物距以计算出实际划痕大小，其计算方法如下：

由几何光学高斯公式： $\frac{l}{l'} - \frac{1}{f} = \frac{1}{f}$ ，可以计算出相距 l' ，再通过垂轴放大率公式： $\beta = \frac{l'}{l} = \frac{y'}{y}$ 即可

再通过 y' 算出实际物长 y 。

还有一个按键用于进入数据集下载地址进行图像下载。显示包括两个文本框用以显示缺陷类型与大小(个数)，还有一个表格用以显示各个划痕的像的大小和实际大小。

对于轧入氧化皮，斑点和划痕的测试, 页面显示如图 4-8-1，4-8-2 与 4-8-3 所示：

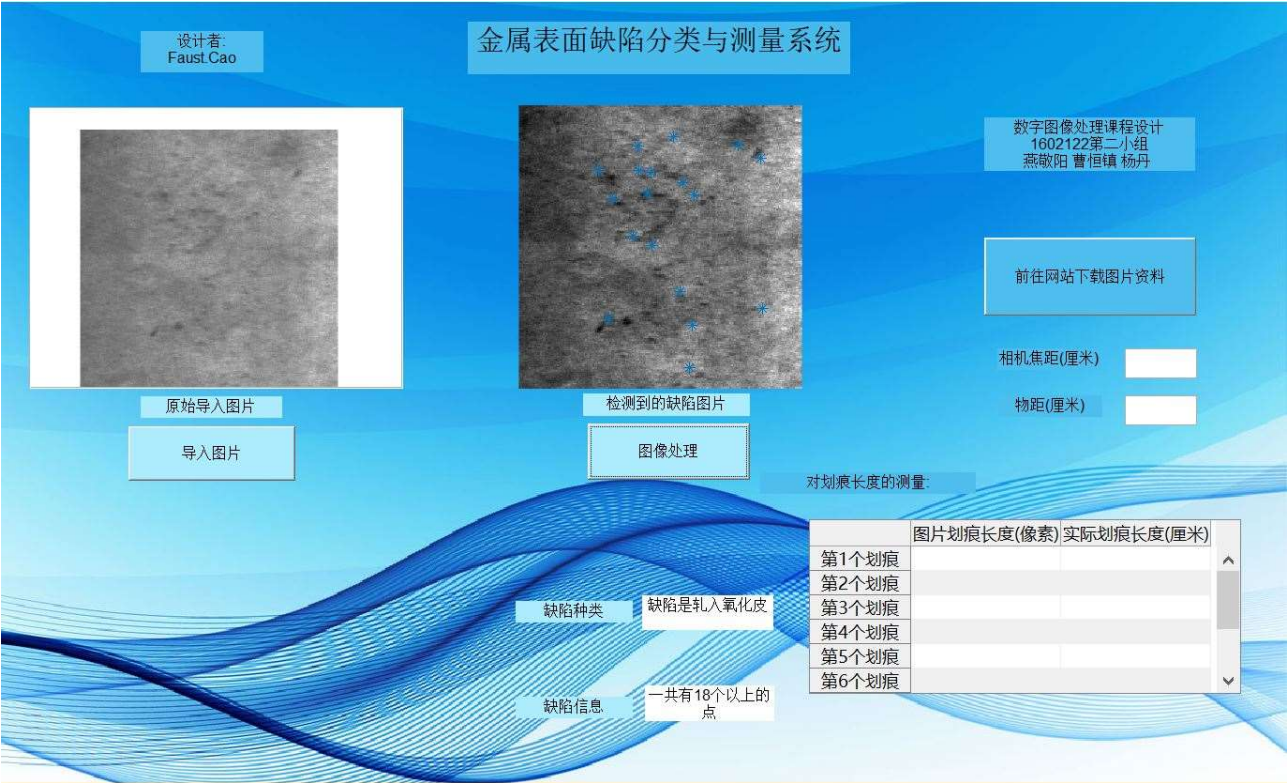


图 4-8-1 轧入氧化皮缺陷

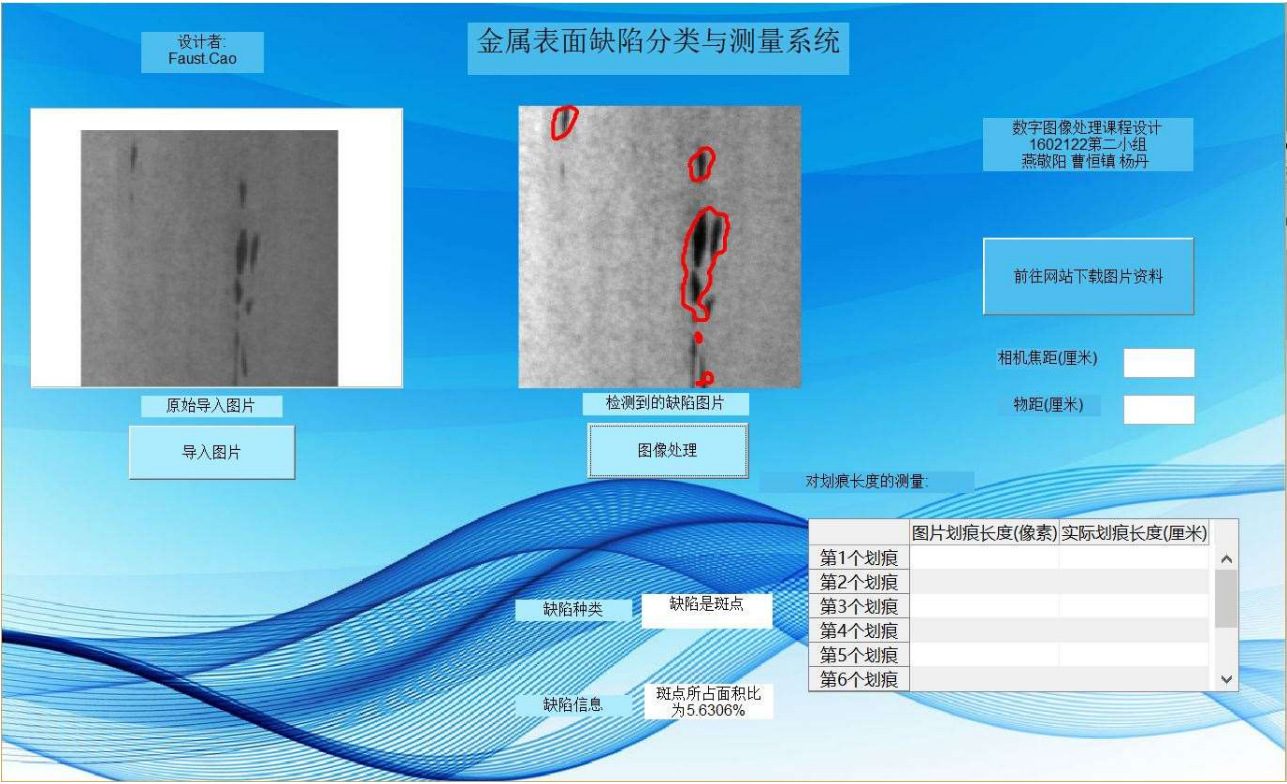


图 4-8-2 斑点缺陷

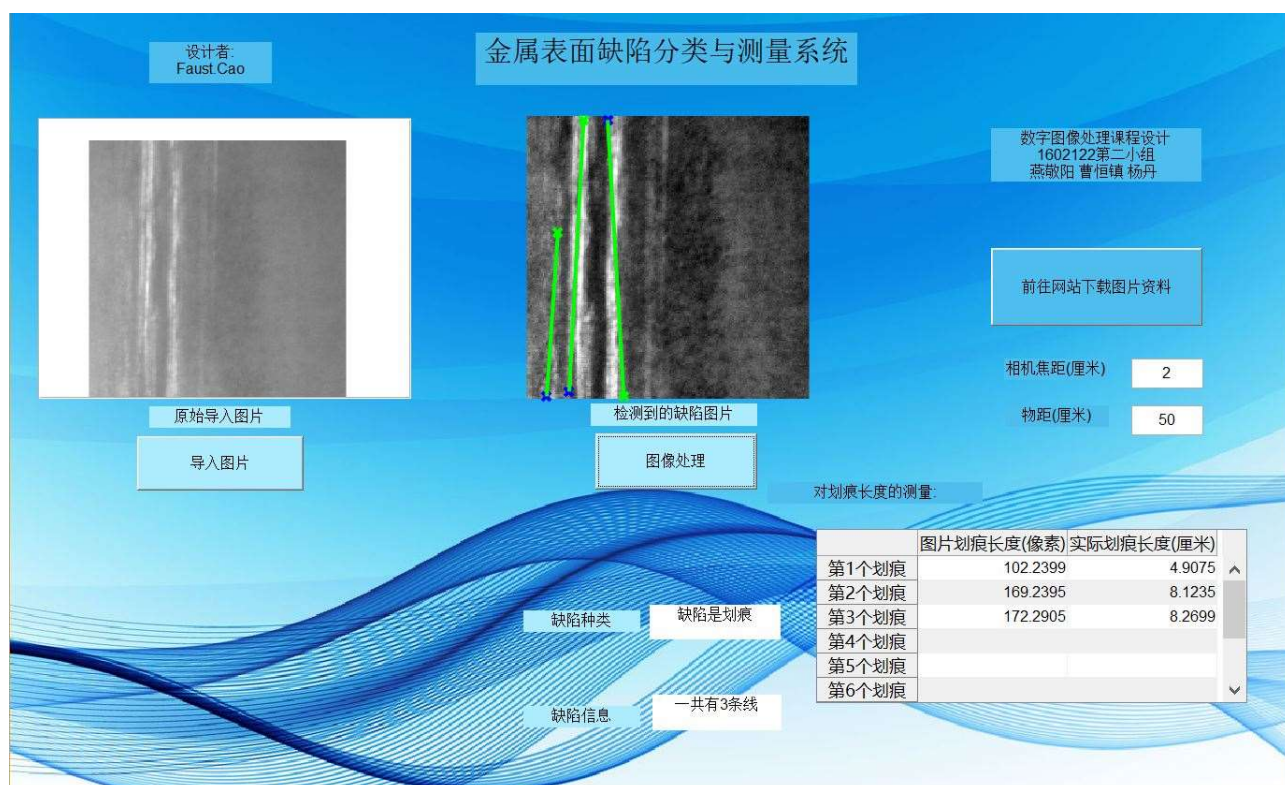


图 4-8-2 划痕缺陷

5. 结果讨论及总结

本方案能检测金属缺陷中比较典型的三种缺陷，对其进行分类，并争对不同缺陷进行位置与大小的计算与显示，使用了一些能够专门处理纹理的算法，例如 Gabor 滤波和 LBP 算法等，总体分类与测量准确性较高，同时开发出了 GUI 页面方便用户操作与展示。

但本文仍存在以下不足之处：

1. 对于划痕的检测，只能检测沿特定方向的划痕(横向与纵向的)。
2. 在一幅图片中，若其中含有多种缺陷，我们只能检测出其中一种缺陷，无法同时检测多种缺陷。
3. 经测试发现，分类时偶尔会出现错误，如下几种情况:将较大的轧入氧化皮检测为斑点，将较细划痕检测为轧入氧化皮。这是由于我们算法中包含一些自己设定的恒定参数，这些参数面对不同图像时往往会失效。
4. 存在一些缺陷的误检与漏检，即无法完全分离噪声与小的缺陷。

6. 论文评分表

考核内容及分值	解决方案 (20 分)	Ppt 及答辩 (20 分)	论文写作 (50 分)	代码编写 (10 分)	总分
得分					

参考文献

- [1] 张刘赞. 基于机器视觉的手机金属板表面缺陷检测技术研究[D]. 浙江大学, 2018
- [2] 范伟. 基于机器视觉的汽车车身零部件表面划痕检测的研究[D]. 南京理工大学, 2016.
- [3] 王旭. 直线提取算法研究[D]. 国防科学技术大学, 2013.
- [4] 宋迪. 基于机器视觉的手机隔板划痕检测研究[D]. 湘潭大学, 2014.
- [5] 乔爽. 复杂背景下光滑物体表面的缺陷检测技术研究[D]. 华中科技大学, 2016.
- [6] 李玉宝. 基于机器视觉的表面缺陷检测算法研究[D]. 中南大学, 2013.
- [7] 李克斌, 余厚云, 周申江. 基于形态学特征的机械零件表面划痕检测[J]. 光学学报, 2018, 38(08): 260-266.
- [8] 郭皓然, 邵伟, 周阿维, 杨宇祥, 刘凯斌. 全局阈值自适应的高亮金属表面缺陷识别新方法[J]. 仪器仪表学报, 2017, 38(11): 2797-2804.
- [9] 汤晓庆, 黄开兴, 秦元庆, 周纯杰. 基于 Gabor 滤波器和 HOG 特征的织物疵点检测[J]. 计算机测量与控制, 2018, 26(09): 39-42+47.
- [10] 徐建亮, 毛建辉, 方晓汾. 基于优化 Gabor 滤波器的铸坯表面缺陷检测应用研究[J]. 表面技术, 2016, 45(11): 202-209.
- [11] 李孟歆, 贾燕雯, 姜佳楠. 基于灰度-梯度二维最大熵阈值法的赤足迹轮廓提取[J]. 电子技术与软件工程, 2016(16): 96-97.
- [12] Weldon, Thomas & E. Higgins, William. (1999). Designing multiple Gabor filters for multitexture image segmentation. Optical Engineering - OPT ENG. 38. 1478-1489.
- [13] Zheng, L., Li, G., & Bao, Y. (2010). Improvement of grayscale image 2D maximum entropy threshold segmentation method. 2010 International Conference on Logistics Systems and Intelligent Management (ICLSIM).
- [14] Maani R, Kali-a S. & Yang Y. H. Noise Robust Rotation Invariant Features for Texture Classification 闭. Pattern Recognition, 2013, 46(8): 2103-2116
- [15] Ojala T., Pietikainen M. & Maenpaa T. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns[C]. European Conference on Computer Vision. 2000: 404-420
- [16] Han Tran, Mohammad Noori, Wael A. Altabey, Xi Wu. Fault diagnosis of rotating machinery using wavelet-based feature extraction and support vector machine classifier[J]. High Speed Machining, 2017, 3(1).
- [17] Kumar A, Pang G K H. Defect detection using Gabor filters[J]. Industry Applications, IEEE Transactions on, 2002, 38(2): 425-440.
- [18] Cristianini N, Shawe-Taylor J. An introduction to support vector machines and other kernel-based learning methods[M]. Cambridge university press, 2000.

附录

数据集来源:

http://faculty.neu.edu.cn/yunhyan/NEU_surface_defect_database.html

美国东北大学金属表面缺陷图片数据集

主程序代码截图:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%主函数文件%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%请运行本程序%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%{
数字图像处理课程设计
1602122第二组
学生:燕敬阳 曹恒镇 杨丹
老师:李琳
-----
//作品概述:
本代码主要功能是实现金属表面三种缺陷的检测、分类及测量,并使用GUI进行封装以便于用户使用。
该GUI可以导入电脑中的灰度图片,之后进行处理,能确定缺陷的种类,将检测到的缺陷标注在原图上,
并能够测量不同缺陷的大小(不同缺陷衡量标准不同)。
-----
//重要函数简介:
#pushbutton1_Callback(hObject, eventdata, handles):
按键1(即'图像处理'按键)的回调函数,包含全部图像处理算法
#pushbutton2_Callback(hObject, eventdata, handles):
按键2(即'图片导入'按键)的回调函数,主要是图片的导入
#Ga(theta,num):图像的Garbor卷积核的初始化
#filter_image(a6,Gaborfilter):Garbor滤波算法
#thres(image):灰度图的最大熵阈值分割算法
#location(): '轧入氧化皮'缺陷的定位及显示
#houghtran():对于'划痕'缺陷的霍夫圆检测
#lbp(varargin):网上的'局部二值模式'特征提取算法
#svm.m:独立m文件,用于所有金属缺陷照片的LBP特征提取和SVM(支持向量机)监督学习算法的训练与测试
-----
author:Faust.Cao
time:2018.11.1
%}

%%
function varargout = app(varargin)
%GUI页面主函数, MATLAB提供, 无需修改

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @app_OpeningFcn, ...
                  'gui_OutputFcn',  @app_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
%%
```

```

function app_OpeningFcn(hObject, eventdata, handles, varargin)
% GUT初始化函数
%修改设置背景
%modified by:Faust.Cao

ha=axes('units','normalized','pos',[0 0 1 1]);
uistack(ha,'down');
%设置程序的背景图为back.jpg
ii=imread('D:\matlab\matlab工程项目\design\back.jpg');
image(ii);
colormap gray
set(ha,'handlevisibility','off','visible','off');

handles.output = hObject;
guidata(hObject, handles);
%%

function varargout = app_OutputFcn(hObject, eventdata, handles)
% GUT输出函数, MATLAB提供, 无需修改

varargout{1} = handles.output;
%%

function pushbutton1_Callback(hObject, eventdata, handles)
% 按键1(即'图像处理'按键)的回调函数, 包含全部图像处理算法

global a5;%全局变量
global svmModel;%全局变量, 从svm.m文件调用
axis off;%去除axes坐标, 完整显示图片
a6=a5;
a=str2double(get(handles.edit1,'string'));%读取用户输入的焦距
b=str2double(get(handles.edit2,'string'));%读取用户输入的物距

d=a/(b-a);%由高斯公式计算垂轴放大率

ma=max(max(a6));
mi=min(min(a6));
%对比度增强, 将灰度映射为[0,255]
for ii=1:200
    for jj=1:200
        a6(ii,jj)=255*double(a6(ii,jj)-mi)./double(ma-mi);
    end
end
%图像滤波
ab=filterimage(a6);
%图像最大熵阈值分割
ss=thres(ab);
ab1=im2bw(ab,ss/255);
[B,L]=bwboundaries(ab1);%二值图边缘坐标提取
[x,y]=size(B);
mm=0;
%寻找最大提取区域
for i=1:x
    ll=length(B{i});
    if ll>mm
        mm=ll;
    end
end
end

```

```

%判断是否是'轧入氧化皮'缺陷
if mm<120
    set(handles.text8,'string','缺陷是轧入氧化皮');%在text8文本框处输出缺陷类型
    [xc,yc]=findcenter(B,ab);%求各提取区域的质心位置
    axes(handles.axes2);%设置在axes2处显示
    imshow(a6);
    hold on;
    plot(yc+12,xc+12,'*');%加上滤波时失去的边界
    [dd,bb]=size(xc);
    tex=['一共有',num2str(dd),'个以上的点'];
    set(handles.text10,'string',tex);%在text10文本框处输出缺陷信息
%不是'轧入氧化皮'缺陷,继续判断
else
    %原图与二值图进行掩模处理,注意二者大小不同,故要先剔除原图边缘
    a6(1:12,:)=[];
    a6(177:188,:)=[];
    a6(:,1:12)=[];
    a6(:,177:188)=[];
    mu1=uint8(ab1).*a6;
    mu2=uint8(1-ab1).*a6;
    t0=mu1(mu1~=0);
    m1=mean(t0);
    t1=mu2(mu2~=0);
    m2=mean(t1);
    %判断缺陷是斑点还是划痕:斑点比背景暗,划痕比背景亮;将掩模与背景剩余均值比较
    %也可以先提取图像的LBP(局部二值模式)特征,再使用SVM训练的结果来进行分类
    if m1<m2+20
        set(handles.text8,'string','缺陷是斑点');

        all=bwarea(ab1);%计算斑点面积
        pro=100*all/176/176;%计算所占百分比
        tex=['斑点所占面积比为',num2str(pro),'%'];
        set(handles.text10,'string',tex);
        axes(handles.axes2);
        imshow(a6);
        hold on;
        %标注出斑点边界
        for k = 1:length(B)
            boundary = B{k};
            plot(boundary(:,2), boundary(:,1), 'r', 'LineWidth', 2)
        end

    %缺陷是划痕
    else
        a6=adapthisteq(a6);%自适应直方图均衡化再次加强对比度
        ss=thres(a6);
        ab1=im2bw(a6,ss/255);

        ab1=imclose(ab1,strel('square',5));%对二值图做闭运算,将断开部分连接
        ab1=bwmorph(ab1,'thin',5);%对二值图再进行细化
        ab1=edge(ab1,'Canny');%使用Canny算子边缘提取
        set(handles.text8,'string','缺陷是划痕');
        axes(handles.axes2);
        imshow(a6);
        hold on;
    end
end

```

```

%往下是划痕提取的后续处理与显示
[point,ang]=houghtran( ab1 );%对处理好的二值图进行霍夫直线检测

cen=zeros(length(ang),2);
po=zeros(length(ang),4);
an=zeros(length(ang),1);
thm=mean(abs(ang));

if abs(thm)<45
    thm=40;
else
    thm=60;
end

le=length(cen);
%求取每根直线的中点位置
for kk=1:le
    cen(kk,1)=(point(kk,1)+point(kk,3))/2;
    cen(kk,2)=(point(kk,2)+point(kk,4))/2;
    point(kk,5)=cen(kk,1);
    point(kk,6)=cen(kk,2);
end
%按中点对直线进行排序
if thm==40
    point=sortrows(point,5);
end
if thm==60
    point=sortrows(point,6);
end

anm=0;
count=0;
data=cell(10,2);
if thm==60
    for kk=2:le
        if abs(point(kk-1,6)-point(kk,6))<5
            po(kk,:)=point(kk,1:4);
            an(kk)=ang(kk);
            anm=mean(an(1:kk));
        else
            [max_x,min_x,max_y,min_y,count] = min_max(po,count);
            dis=plotfigure(max_x,min_x,max_y,min_y,anm);
            data{count,1}=dis;
            rea=dis/(d*500);
            data{count,2}=rea;
            po=zeros(length(ang),4);
            set(handles.uitable1,'DATA',data);
            po=zeros(length(ang),4);
        end
    end
end
[ max_x,min_x,max_y,min_y,count ] = min_max(po,count);
dis=plotfigure(max_x,min_x,max_y,min_y,anm);
tex=['一共有',num2str(count),'条线'];
set(handles.text10,'string',tex);
data{count,1}=dis;%直线所占像素长度
rea=dis/(d*500);%假设探测器像面1cm*1cm。像元20um。200*200像素
data{count,2}=rea;%几何光学转化后的实际长度
set(handles.uitable1,'DATA',data);

```

```

elseif thm==40
    for kk=2:le
        if abs(point(kk-1,5)-point(kk,5))<10
            po(kk,:)=point(kk,1:4);
            an(kk)=ang(kk);
            anm=mean(an(1:kk));
        else
            [max_x,min_x,max_y,min_y,count] = min_max(po,count);
            dis=plotfigure(max_x,min_x,max_y,min_y,anm);
            data{count,1}=dis;
            rea=dis/(d*500);
            data{count,2}=rea;
            po=zeros(length(ang),4);
        end
    end
    [max_x,min_x,max_y,min_y,count] = min_max(po,count);
    dis=plotfigure(max_x,min_x,max_y,min_y,anm);
    data{count,1}=dis;
    rea=dis/(500*d);
    data{count,2}=rea;
    set(handles.uitable1,'DATA',data);
    tex=['一共有',num2str(count),'条线'];
    set(handles.text10,'string',tex);
end
end
end
%%

function pushbutton2_Callback(hObject, eventdata, handles)
% 按键2(即'图片导入'按键)的回调函数,主要是图片的导入
data={};
set(handles.uitable1,'DATA',data);
hold off;
plot(0,0);
axis off;
global a5;%全局变量,函数间调用(能被按键1回调函数调用)
axes(handles.axes1);%设置在axes1处显示图片
%默认路径:D:\NEU surface defect database,用户可以在此处修改
[filename,pathname]=uigetfile({'*.bmp;*.jpg;*.png;*.jpeg;*.tif'},'Pick an image','D:\NEU
surface defect database');
str=[pathname filename];
if isequal(filename,0)||isequal(pathname,0)
    warndlg('请先选择一个金属表面缺陷的图片!','Warning');%未选择照片提示
    return;
else
    a5 = imread(str);
    imshow(a5);%在axes1处显示图片
end
end
%%

function pushbutton4_Callback(hObject, eventdata, handles)
% 按键4的回调函数,给用户提供一个下载金属缺陷数据集的网站超链接
web https://pan.baidu.com/s/1B_SLhZNRPeKznDc78GY2QA;%图片数据集下载地址
%%

```

```

function edit1_Callback(hObject, eventdata, handles)
% 文本框1的回调函数, 让用户输入所用相机的焦距来计算实际物体大小

a=str2num(get(hObject,'string'));%获得用户输入
if isempty(a)
    set(hObject,'string','0');
end
guidata(hObject,handles);
%%

function edit1_CreateFcn(hObject, eventdata, handles)
% 文本框1的初始化函数, MATLAB提供, 无需修改

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
%%

function edit2_Callback(hObject, eventdata, handles)
% 文本框2的回调函数, 让用户输入所用物距来计算实际物体大小

a=str2num(get(hObject,'string'));%获得用户输入
if isempty(a)
    set(hObject,'string','0');
end
guidata(hObject,handles);
%%

function edit2_CreateFcn(hObject, eventdata, handles)
% 文本框1的初始化函数, MATLAB提供, 无需修改

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%end of the
file%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

其他函数代码见文件夹附件。