

2019年秋季 图像处理与分析 第九讲作业

学院：人工智能学院 学号：201928014629008 姓名：牛李金梁

编程语言：Python3.7

必要依赖：numpy libopencv matplotlib

运行要求：在命令行中输入 `python 9.py`，所弹出的窗口展示了 2 种骨架提取的结果。

模块1 基本操作

程序使用 `opencv` 中的 `imread()` 读入下面的指纹灰度图像。



图 1 指纹原图

二值化方法采用的是简单的阈值分割，观察图像，将阈值设为 150，并将所有大于阈值的像素置为 0，小于阈值的像素置为 1（之后的二值图像均按此标准，即黑 0 白 1），得到如下的结果。



图 2 二值化图像

首先编写腐蚀函数 `erode()` 和膨胀函数 `dilate()`。

在腐蚀操作中，结构元反射后再卷积，相当于直接进行相关操作，因此可以直接调用 `opencv` 的 `filter2D()` 对图像和结构元来进行这一操作，将操作后的每个像素的值与结构元中 1 的个数相比较，相等的为 1，即完成了腐蚀。

在膨胀操作中，结构元对图像进行卷积，所以要旋转 180° 后调用 `opencv` 的 `filter2D()`，将操作后每个像素的值只要不为 0 就置 1，就完成了膨胀操作。

通过与 ppt 中提供的例子进行测试，验证了上述 2 个函数的编写没有问题。

之后编写击中击不中变换函数 `hit_miss()`。

击中击不中变换采用了 2 个不相交的卷积核，这 2 个卷积核可以存在 1 个模板中。在本程序中，用 1 和 0 指示这 2 个卷积核，模板的其他元素则设置为 -1。首先在图像外围填充上全 0 的边界。再利用 1 的模板腐蚀原图，再将图像取补，用 0 的模板腐蚀该图，得到的结果求交去除填充边界，便完成了击中击不中变换。

模块2 形态学骨架提取

该部分采用的是不断细化的方法。

首先是细化函数 `thin()`。先进行的是击中击不中变换，不过只将不破坏连通性的像素置 0。具体的实现是用卷积计算每个像素邻居中 1 的个数，并以此判断每个像素的连通性，从而与击中击不中变换的结果求交集便得到了需要被置 0 的像素。然后原图减去这些像素便

完成了1次细化。

在正式的骨架提取中，首先要定义8个模板，模板中的1和0指示了击中击不中变换中需要的2种模板。然后，循环使用这些模板进行细化操作，直到细化不再改变结果时停止迭代，便完成了骨架提取，结果如下图。



图3 形态学骨架提取

模块3 基于距离变换的骨架提取

首先要编写函数 `edge_extract()` 提取边界。使用的是全 1 的 3x3 模板来提取单像素宽度边界，具体就是原图减去腐蚀后的结果。效果如下图。



图4 边界提取

之后将边界提取后的结果送入距离变换函数 `distance_transform()`，该函数将 `matlab` 程序进行了修改，通过填充使其不再丢弃首尾两列。

然后将距离变换的函数送入求取局部极大值的函数 `get_local_max_img()` 中。该函数首先进行填充，然后以 7x7 邻域的极大作为局部极大值点。

这样得到的结果便是基于距离变换的骨架，如下图所示。



图4 基于距离变换的骨架提取

可以看出，和形态学相比，效果不太好，主要在于对一些连通分量的破坏，但对于如何在破坏连通的情况下删除非骨架点同时保持骨架的平滑，没有想到好的解决办法。

模块4 裁剪

裁剪完全按照 `ppt` 上的步骤所进行的，并将腐蚀次数设为 3，可以裁剪掉大于 3 个像素的毛刺。效果如下图所示。

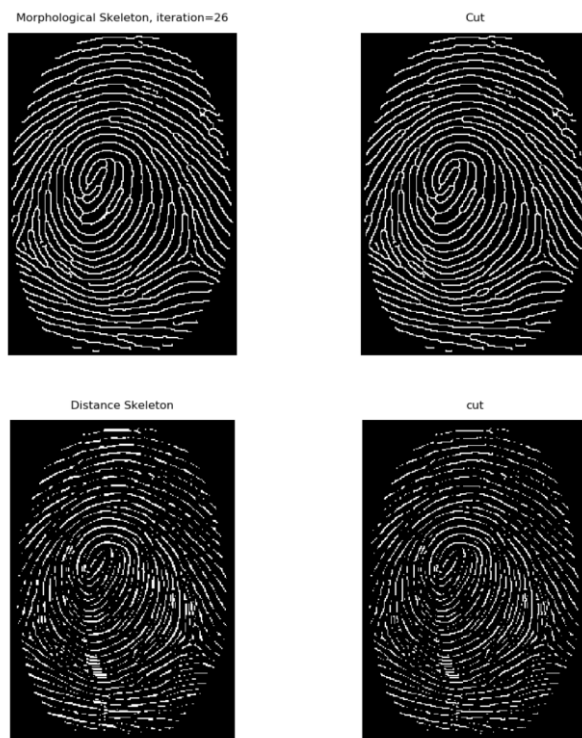


图 4 裁剪效果（上为形态学骨架，下为距离变换骨架）

通过裁剪前后进行减法可得，裁剪能删去数十个像素，具有一定的效果。从肉眼来看，裁剪对于骨架中某些过粗部分的毛刺去掉。