# 2024数字图像处理实验报告

**学号：09022107**

**姓名：梁耀欣**

1. 实验题目（实验题目具体内容）

对以下四张图片分别复现书上以下结果：

* + 霍夫变换：直线检测
  + 阈值分割
    - 基本的全局阈值处理
    - Otsu阈值分割
    - 可变阈值处理









1. 实验环境（实验所使用的平台和相关软件）

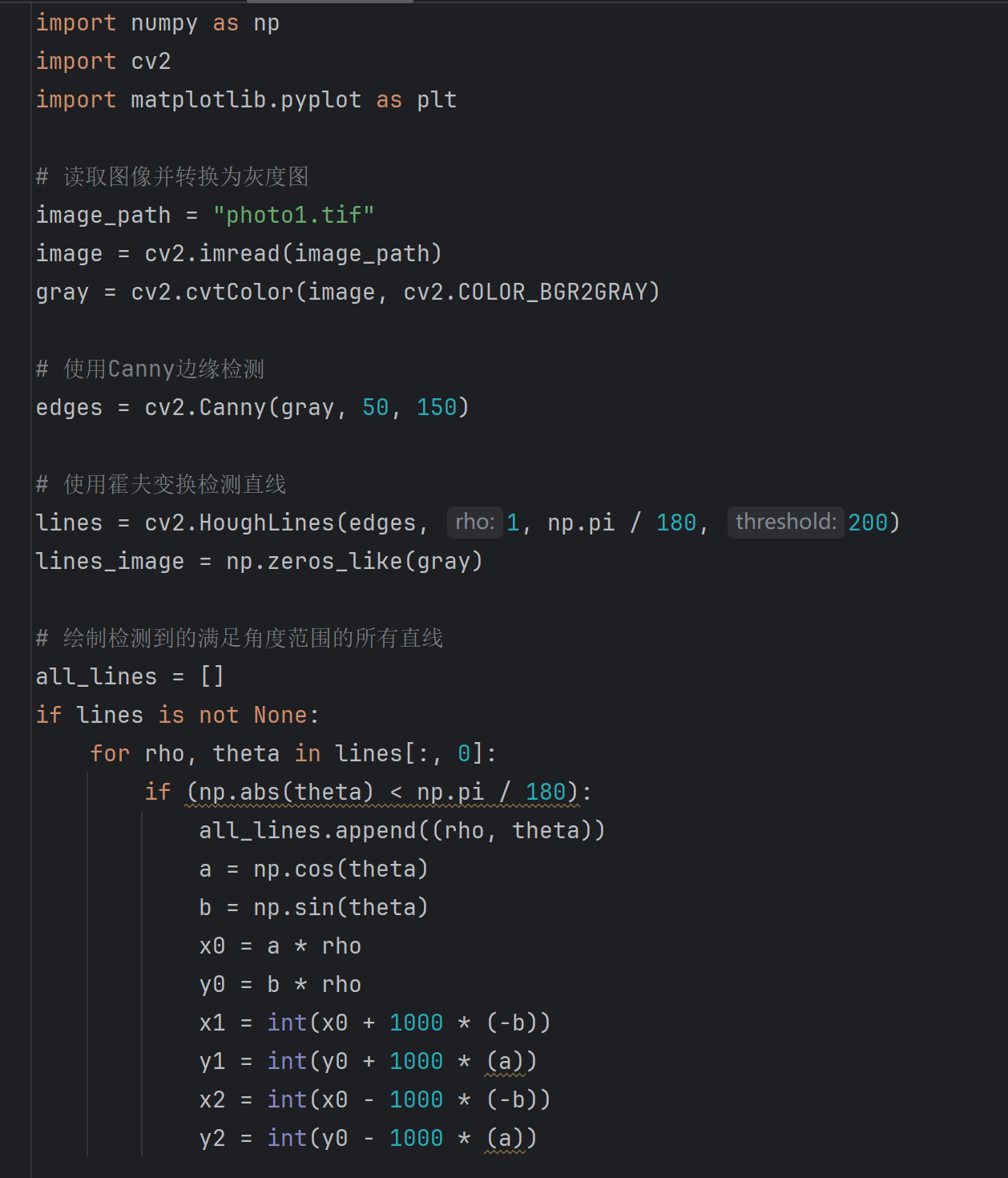
Pycharm 2024.2.0.1

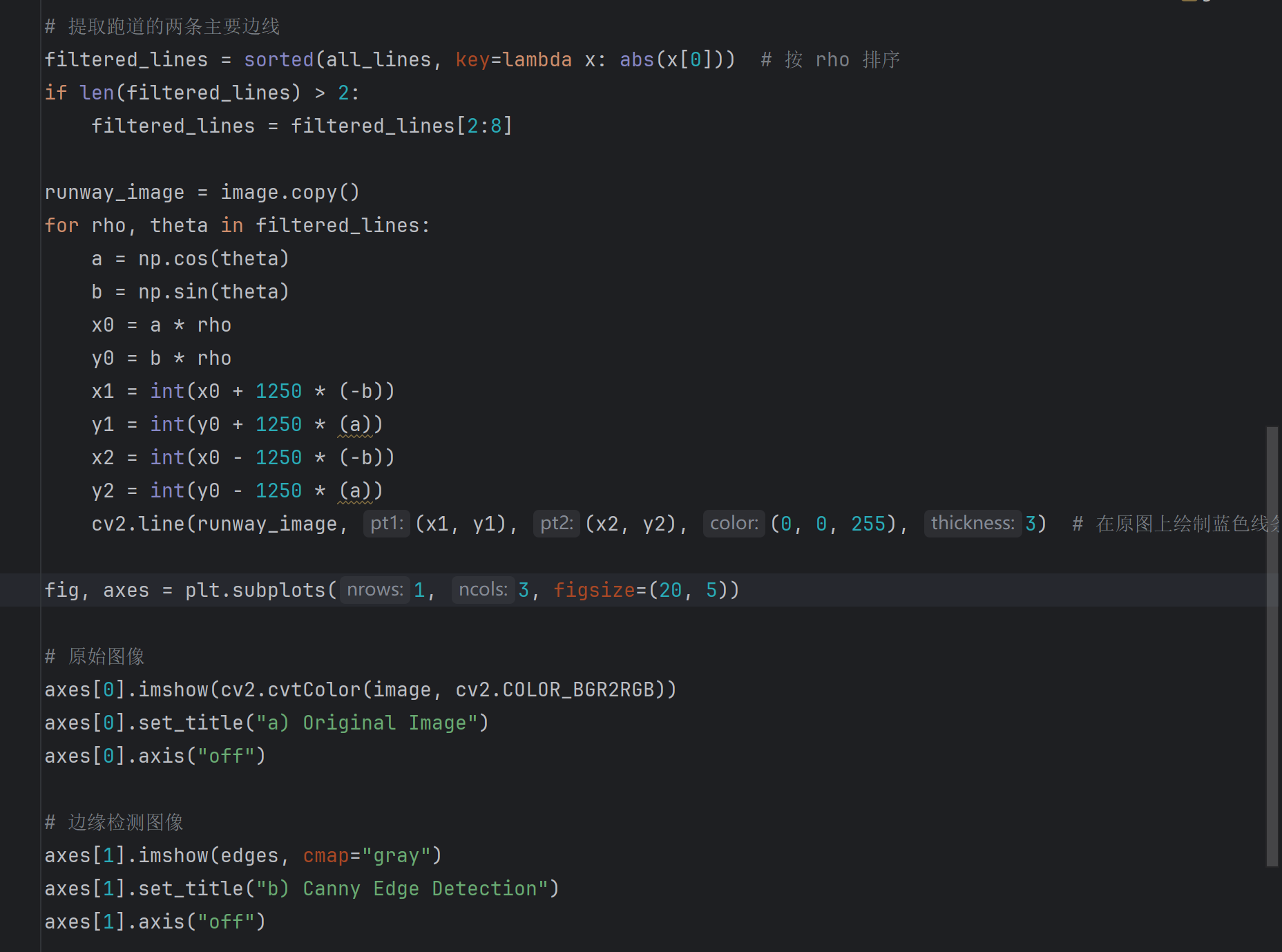
Python 3.9

1. 实验内容与方法（实验算法实现代码等）
2. 霍夫变换直线检测：

首先通过 cv2.imread 方法加载图像 photo1.tif，并将其转化为灰度图像，以简化处理和边缘检测。先用滤波器对图像进行降噪处理，然后使用 Canny 边缘检测算法，通过寻找强边缘梯度来提取图像中的边缘信息。用参数 50 和 150确定边缘强度的低阈值和高阈值。使用 的Sobel 算子窗口大小为 3x3，然后通过 cv2.HoughLines 方法应用霍夫变换来检测直线，利用直线参数 (ρ, θ) 来计算直线在图像平面的起点和终点，并在原图上绘制这些直线。

代码：

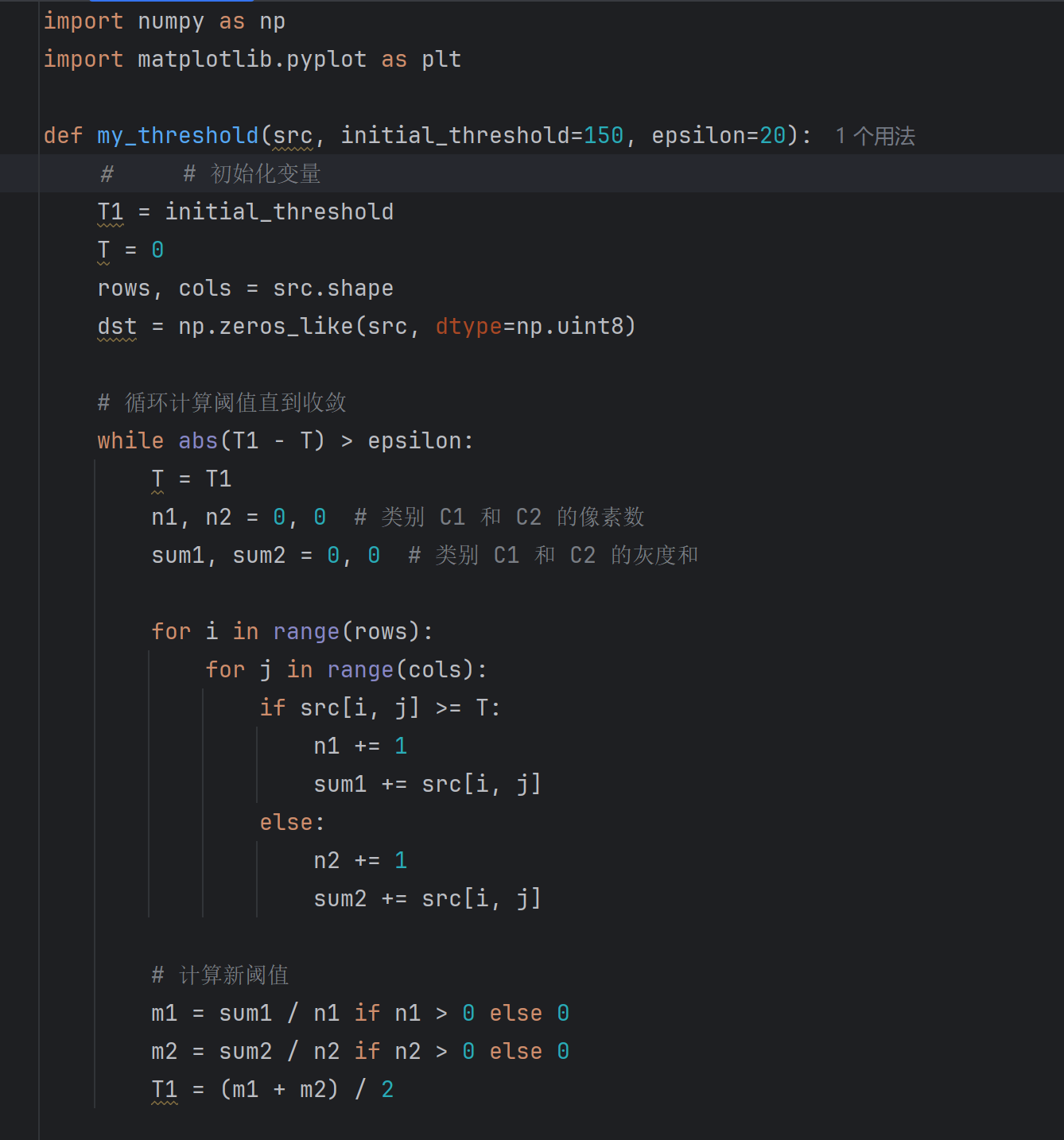


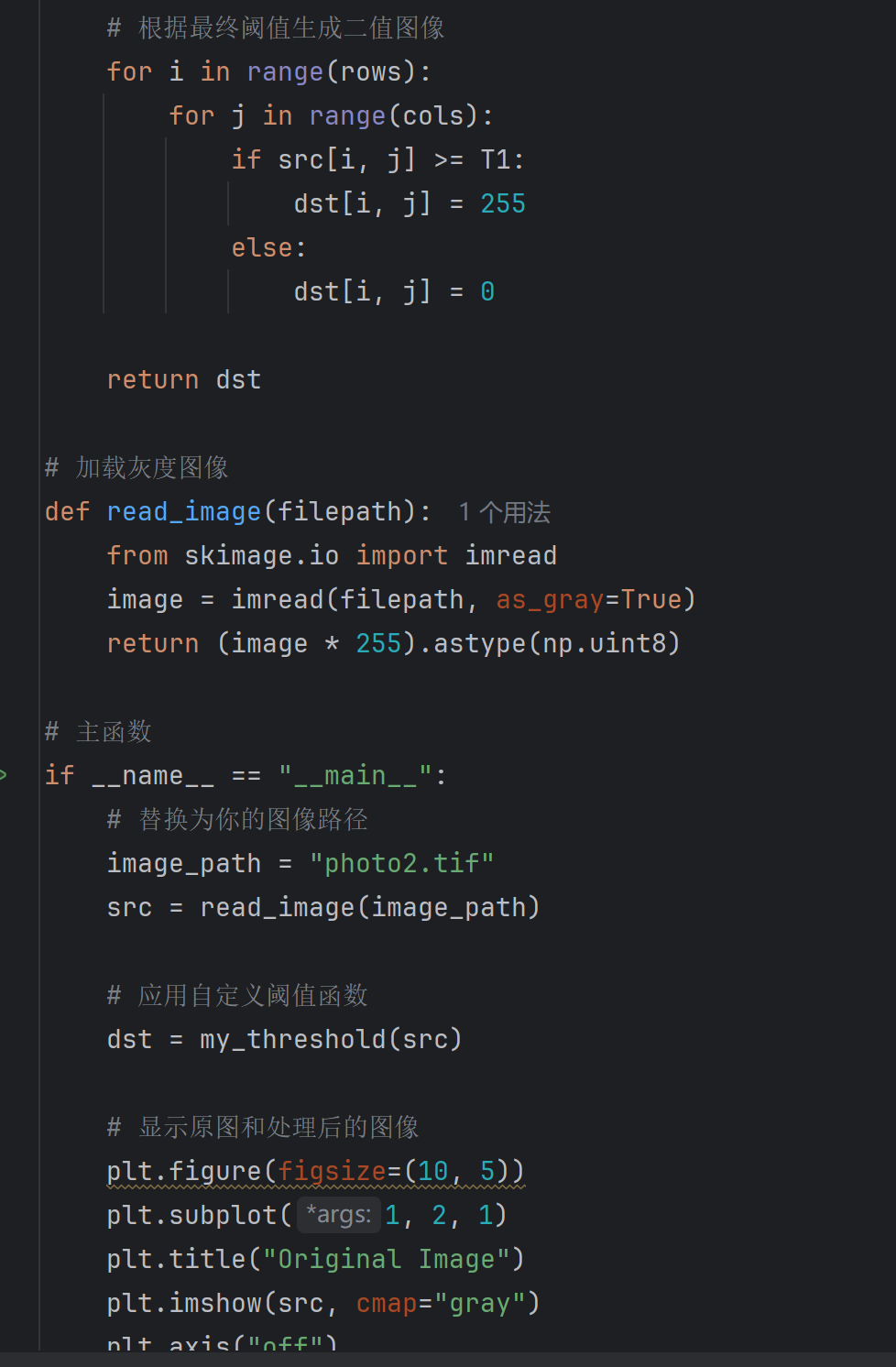


2．基本的全局阈值处理

使用 read\_image 函数加载灰度图像 photo2.tif，将其归一化为 0 到 255 之间的像素值。初始化阈值 T1 设为 150，并设置最小变化 epsilon 为 20，循环计算类间方差，直至阈值的变化小于 epsilon，达到收敛条件。对每个像素点计算是否归为两类背景或前景，每次迭代基于类间方差更新阈值，直到收敛，最终生成一个二值化图像。

代码：

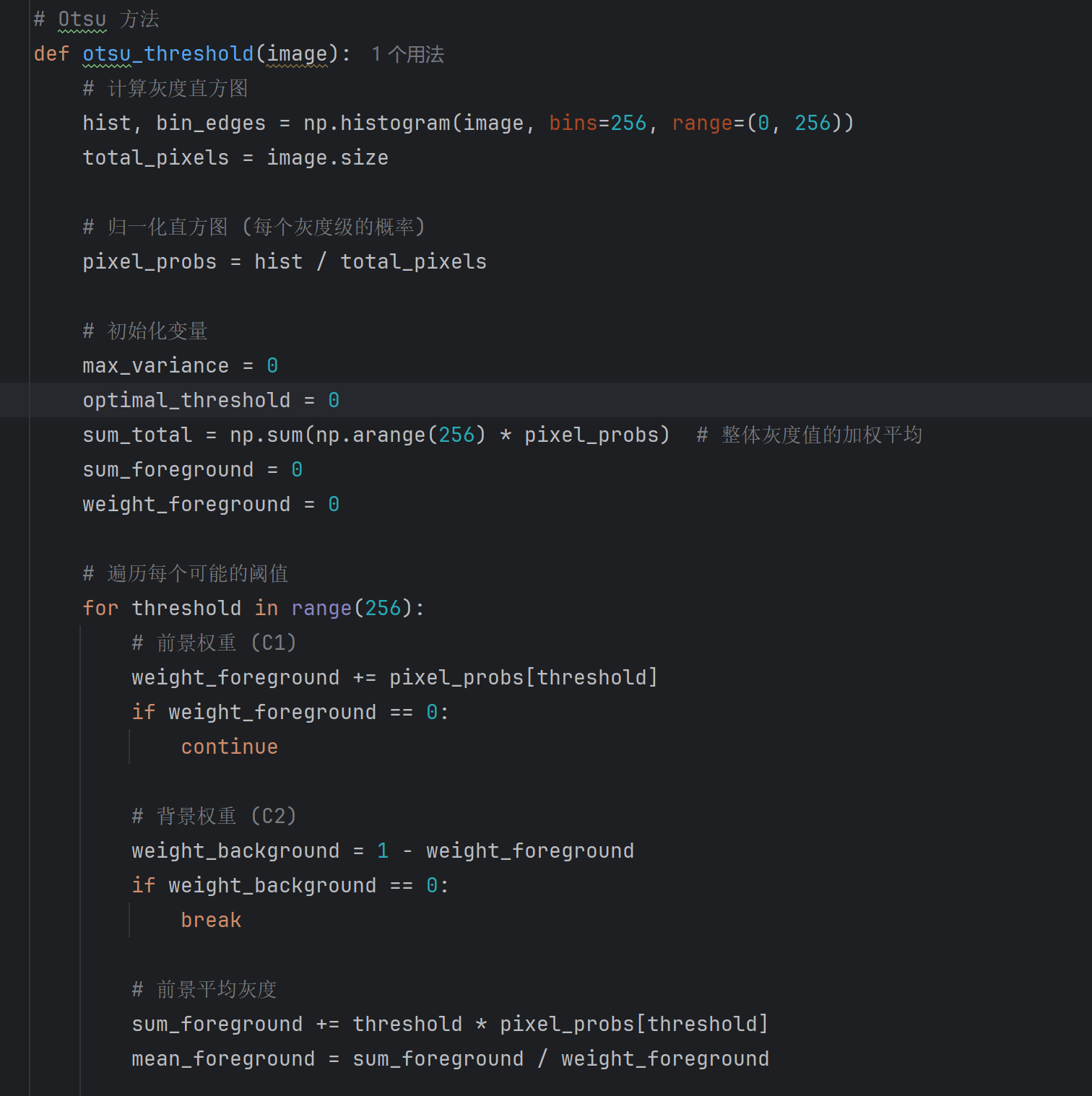


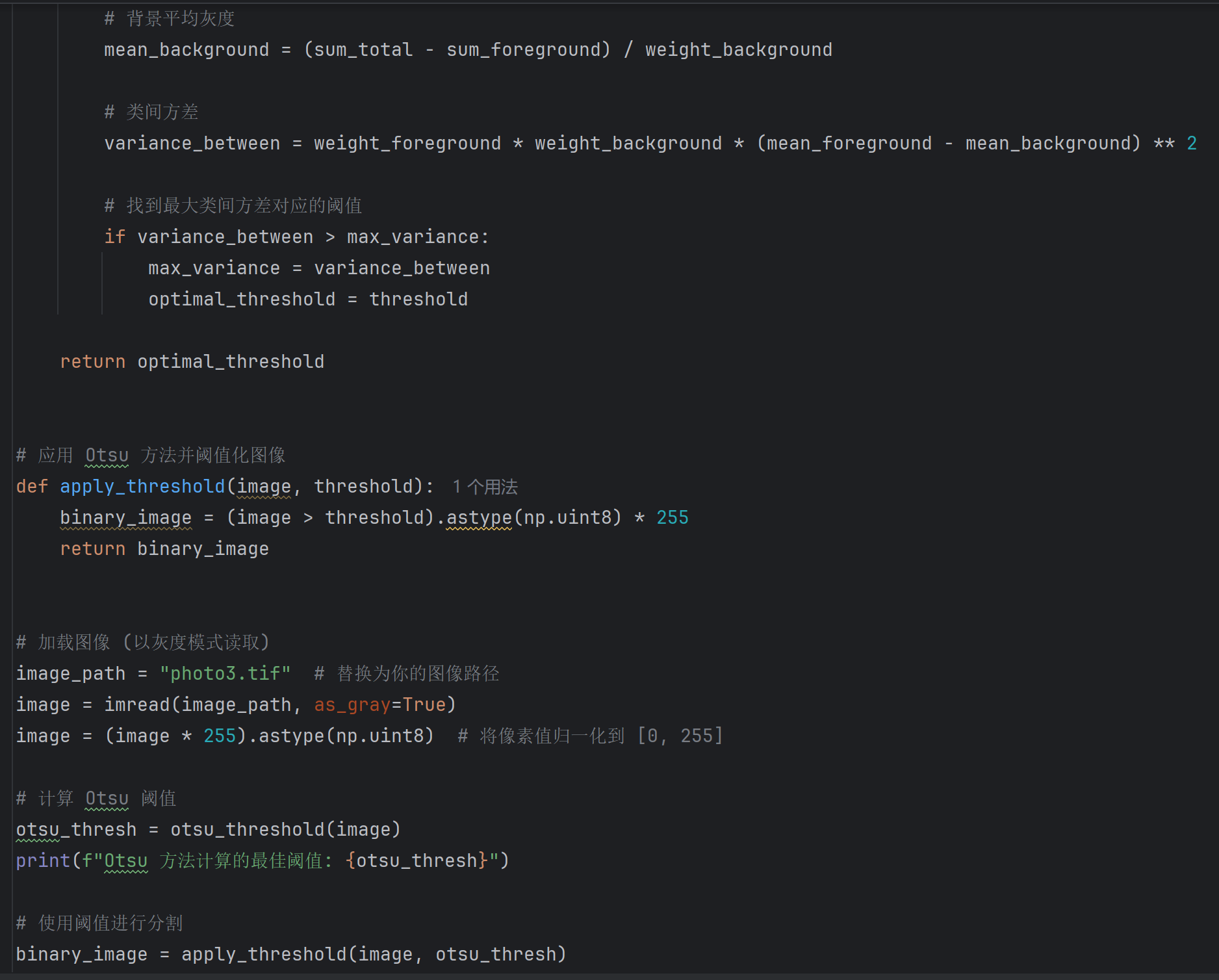


3．Otsu阈值分割

使用 skimage.io.imread 读取图像 photo3.tif，并转为灰度图像，将像素值归一化为 [0, 255] 范围。计算图像的直方图，并计算每个灰度级的概率分布，然后使用类间方差公式，对每个灰度级进行迭代，计算类间方差，寻找最大类间方差对应的阈值，即为 Otsu 方法的最终阈值。利用计算得到的 Otsu 阈值，将图像二值化，大于阈值的像素赋值为 255，其他像素赋值为 0。

代码：





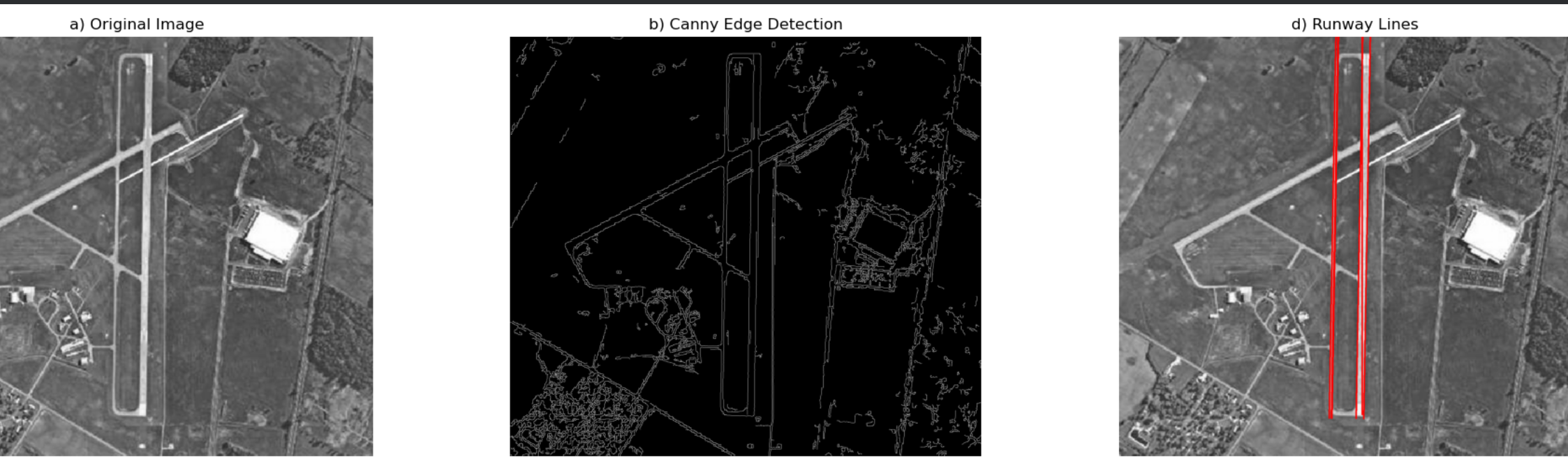
4.可变阈值处理

首先使用 OpenCV 的 cv2.imread 函数读取图像 photo4.tif 并将其转换为灰度图像。用 adaptive\_threshold 函数实现基于局部均值和标准差的自适应阈值分割。计算每个像素的局部阈值，将图像的每个像素与局部阈值进行比较：如果大于局部阈值，赋值为 255（前景）；否则，赋值为 0（背景）。

代码：

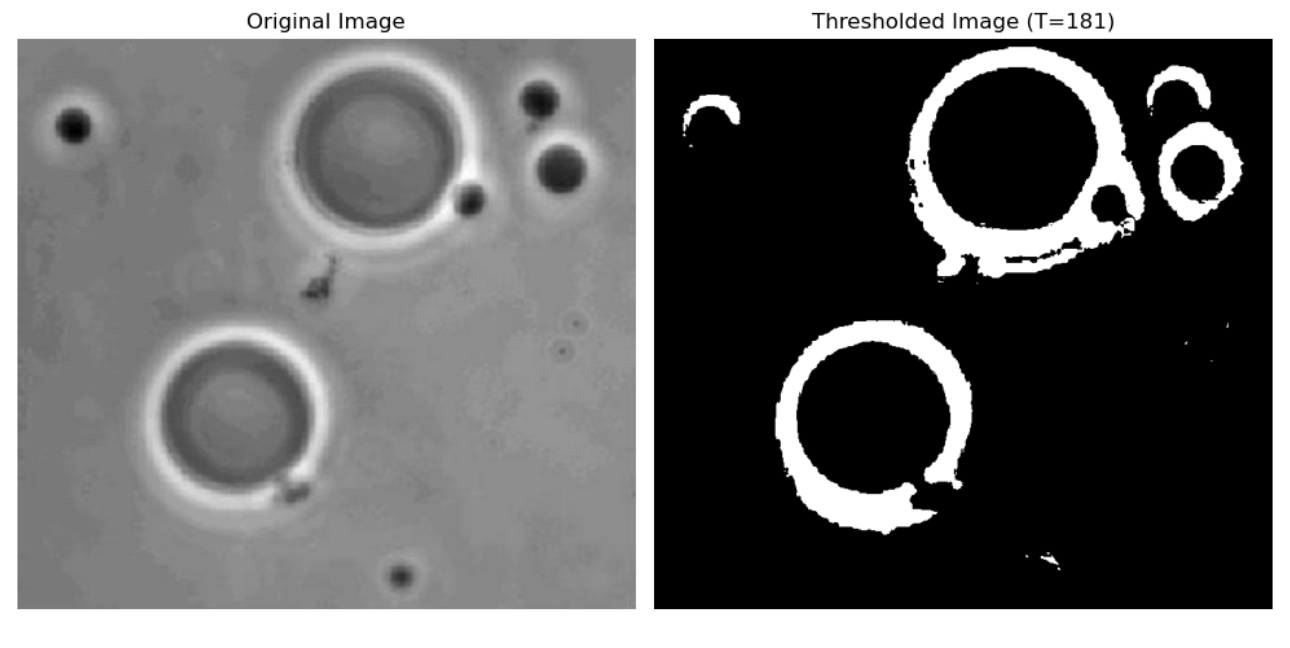


1. 实验结果与分析

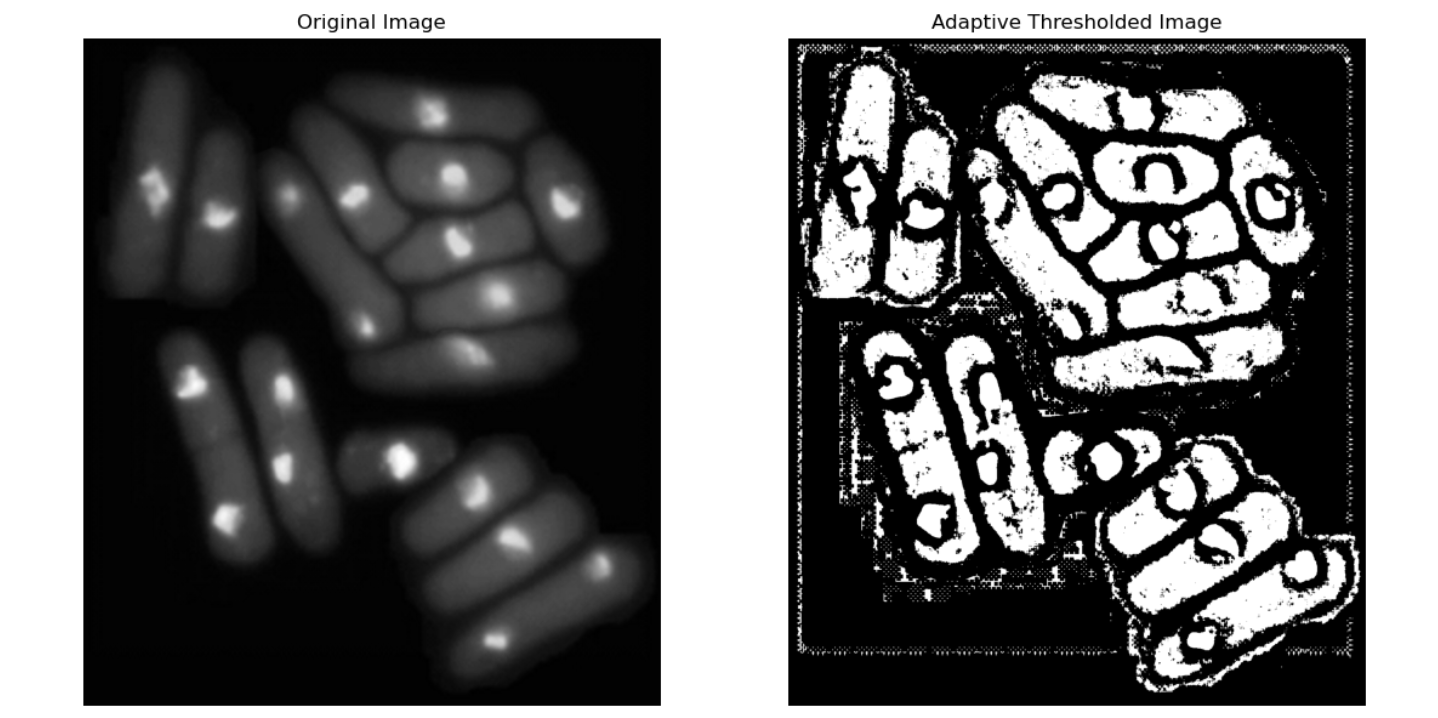


使用霍夫变换对原始图像进行直线检测处理后，得到了标记有红色线条的图像，能够准确地标记出机场跑的位置，表明直线检测算法在这种场景下工作良好，能够有效识别出图像中的线性特征。



对原始指纹图像进行了固定阈值处理，将图像转换为二值图像。阈值处理后，指纹的纹路被清晰地分离出来，背景变为白色，指纹纹路变为黑色。

使用ostu对原始图像进行处理后，得到了二值图像。阈值处理后，细胞的轮廓被突出显示，背景被去除。但也表明阈值的选择可能会影响结果的准确性，不同的阈值可能会导致不同的细胞细节被保留或丢失。



使用可变阈值处理对原始图像进行处理后，得到了二值图像。自适应阈值处理能够根据图像局部的亮度和对比度自动调整阈值，因此在处理光照不均匀的图像时效果较好。在这种情况下，自适应阈值处理能够较好地突出细胞的结构，而不受背景亮度变化的影响。

1. 总结与思考

在实验四里对于参数的调整一直把握不好，导致产生的效果不佳，通过反复实验调整出了最佳的参数值。本次实验通过对四张不同类型的图像进行数字图像处理操作，包括霍夫变换直线检测、基本的全局阈值处理、Otsu 阈值分割和可变阈值处理，成功复现了书上的相关结果。在霍夫变换直线检测中，利用 Canny 边缘检测和霍夫变换能够准确地检测出图像中的直线特征。基本的全局阈值处理通过迭代计算类间方差来确定阈值，实现了对指纹图像的二值化处理，有效分离出指纹纹路。Otsu 阈值分割基于最大类间方差原理，自动确定阈值，对细胞图像进行二值化处理，突出了细胞轮廓，但也显示了阈值选择对结果的影响。可变阈值处理则通过自适应阈值分割，根据图像局部的亮度和对比度自动调整阈值，在处理光照不均匀的细胞图像时表现出较好的效果。