

# 实验报告

由于页数限制，只保留必要的流程和函数介绍

## 算法流程

霍夫圆变换是一种用于图像处理中检测圆形物体的技术，以下是算法的流程。

### 1. 应用高斯模糊

使用 `apply_gaussian_blur` 函数对图像应用高斯模糊。

### 2. 计算梯度

`compute_gradients` 函数被用于计算高斯模糊后图像的梯度幅值和方向。

### 3. 非极大值抑制

使用 `non_maximum_suppression` 函数对梯度幅值进行非极大值抑制处理。

### 4. 双阈值处理

`double_threshold` 函数应用双阈值方法进一步细化边缘。

### 5. 霍夫圆变换

使用 `hough_circle_transform` 函数在确定的边缘图像上执行霍夫圆变换，检测图像中的圆形。

## 函数功能说明

#### 1. `apply_gaussian_blur(image, kernel_size=9, sigma=2.0)`

**功能：**对图像应用高斯模糊，以平滑图像并减少噪点。

**参数：**`image`：原始图像。`kernel_size`：整数，高斯核的大小。默认值为9。`sigma`：浮点数，高斯核的标准差。默认值为2.0。

**返回：**返回一个新的Pillow图像对象，它是原始图像经过高斯模糊处理后的结果。

#### 2. `compute_gradients(image)`

**功能：**计算图像的梯度幅值和方向。

**参数：**`image`：灰度图像的NumPy数组。

**返回：**`gradient_magnitude`：一个NumPy数组，包含了图像每一点的梯度幅。

`gradient_direction`：一个NumPy数组，包含了图像每一点的梯度方向。

#### 3. `non_maximum_suppression(gradient_magnitude, gradient_direction)`

**功能：**应用非极大值抑制，以细化图像的边缘。

**参数：**`gradient_magnitude`：NumPy数组，包含图像的梯度幅值。`gradient_direction`：NumPy数组，包含图像的梯度方向。

**返回：** 返回一个NumPy数组，它是应用了非极大值抑制算法后的图像。

4. `double_threshold(nms_image, low_threshold, high_threshold)`

**功能：** 进一步细化由非极大值抑制(`non_maximum_suppression`)步骤产生的边缘。

**参数：** `nms_image`：经过非极大值抑制处理的图像，它是一个灰度图像，其中的像素值代表梯度幅值。`low_threshold`：低阈值，用于边缘检测。低于此阈值的边缘将被忽略，不认为是边缘。

`high_threshold`：高阈值，用于确定强边缘。高于此阈值的边缘被认为是真正的边缘。

5. `hough_circle_transform(edges, radius_range, radius_step=1, angle_step=1, threshold=50, quantity=100, merge_distance=30)`

**功能：** 该函数使用霍夫圆变换方法，在给定的边缘图像中检测圆形。

**参数：** `edges`：边缘检测后的图像矩阵，边缘应标记为255，非边缘为0或其他较低值。

`radius_range`：一个元组，指定搜索圆形的最小和最大半径范围 (`R_min`, `R_max`)。

`threshold`：累加器的阈值，只有当圆的累加器值大于或等于此值时，才被认为是有效的

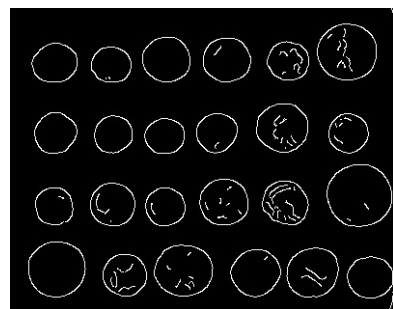
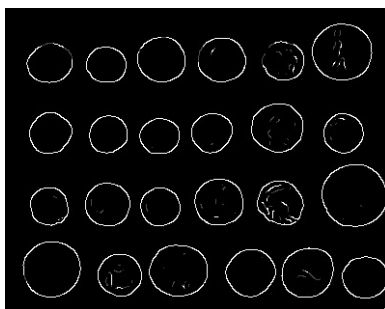
**返回：** 返回一个圆形列表。每个圆形由一个元组表示，格式为 `(y, x, r)`，其中 `y` 和 `x` 是圆心的行列坐标，`r` 是圆的半径。这些圆形是根据函数参数和边缘图像通过霍夫圆变换检测到的。

## 实验结果

### 1.边缘检测（和cv2（右）对比）

分析两张图像可以看到，在参数一致的情况下，自定义的函数和cv2的canny边缘检测结果相似。

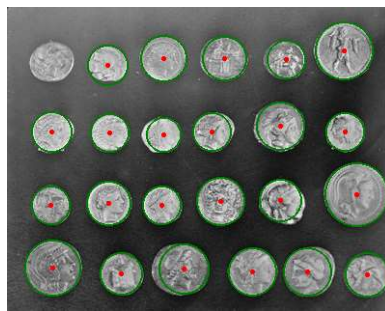
由于在自定义的函数中加入了双重阈值的处理，所以有些边的颜色比较淡，在后续圆形检测时低阈值部分将提供更少的votes（0.5），这样子更加容易找到圆而不会被内部或者外部的弱边缘影响。



### 2.圆形检测

可以看到除了左上角的硬币都能够成功检测到。（和cv2的结果一致）

在实验中，经过几次参数的调整，左上角的圆也能检测出来，但是在进行其他图像的实验室，该参数的泛用性太差，所以这里还是展示此结果。



### 3.不同参数对于结果的影响

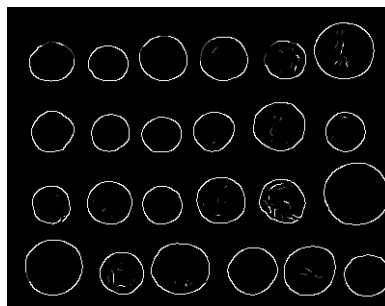
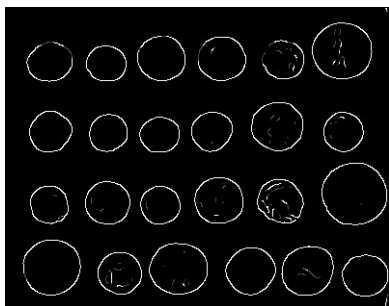
高斯模糊：

#### 1. 核大小 ( `kernel_size` )

**影响：**核大小决定了高斯滤波器的尺寸。较大的核会涵盖更多的邻域像素，导致更强的模糊效果。这有助于去除图像的噪声，但也可能导致边缘模糊，减少细节。但是实验中发现影响不大。

左图为9\*9 ( $\sigma=2.0$ )

左图为45\*45 ( $\sigma=2.0$ )

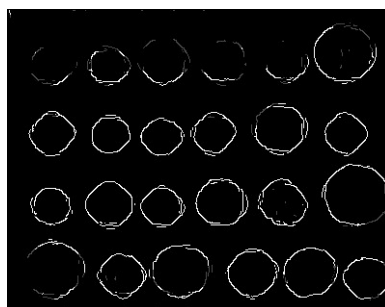
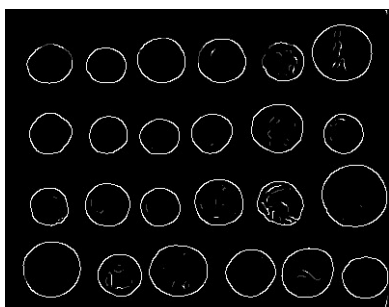


#### 2. 标准差 ( `sigma` )

**影响：** $\sigma$ 值决定了高斯函数的宽度，进而影响模糊的程度。较高的 $\sigma$ 值意味着更宽的高斯函数，导致更强的模糊效果，但是也很容易让边缘难以分辨。

左图为2.0 (`kernel_size=9`)

左图为6.0 (`kernel_size=9`)

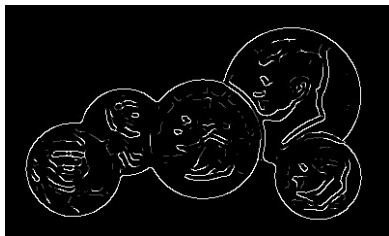


边缘检测：

## 1. 边缘阈值 ( `threshold` )

**影响：** 决定了判断是否为边缘的阈值，更大的阈值对非边缘的排除更好，更小的阈值对边缘的采样更充分。

左图为90



右图为70



**圆形检测：**

### 1. 半径范围 ( `radius_range` )

**影响：** 半径范围决定了算法搜索圆形的大小范围。如果半径范围设置得过窄，可能会错过一些圆形；如果设置得过宽，则会增加计算复杂度，且可能检测到错误的圆形。

左图为 (20, 100)



右图为 (10, 100)



### 2. 累加器阈值 ( `threshold` )

**影响：** 累加器阈值决定了一个位置被认定为圆心需要的最低票数。较高的阈值可以减少假阳性（即错误地检测到的圆形），但也可能错过一些真正的圆形。在实操中进一步在merge前先进行了圆形数量的限制（最多100个圆），导致这个阈值不是唯一决定的参数，对结果影响有限。

**实验结论：**

在对不同图片进行实验的过程中，发现参数必须根据数据的不同而进行调整，实际应用中，可能需要通过多次试验来调整这些阈值，以获得最佳的边缘检测效果。