

计算机视觉 实验

钱币定位系统

本实验设计了一个钱币定位系统，其不仅能够检测出输入图像中各个钱币的边缘，同时，还能给出各个钱币的圆心坐标与半径。

一 算法流程

1 Canny 算法实现

本实验首先使用 python 自行实现了 Canny 边缘检测算法，将输入图片边缘提取。通过适当调整算法的输入参数，可以将提取的边缘调整在适当的强度内，使得硬币内的花纹不明显、硬币外围的圆形弧线明显。将生成图作为下一流程的输入。

Canny 算法的常规实现步骤如下：

- (1) 对图像进行高斯模糊处理：高斯平滑，滤除高频分量。
- (2) 求出图像的梯度图像及梯度方向矩阵。
- (3) 非极大化抑制：使宽边变为细边。
- (4) 滞后门限：用强边延伸弱边，解决伪边问题。

2 Hough 算法实现

本实验在 Canny 算法的基础上，通过 python 实现了 Hough 算法实现圆形检测，并返回圆心坐标和半径，同时实现可视化。相比较最小二乘法、鲁棒估计等方法，霍夫变换的优点在于可以进行多目标的拟合。

Hough 算法的常规实现步骤如下：

- (1) 建立霍夫参数三维空间，并对空间内各个单元进行投票。
- (2) 设置阈值从投票结果中筛选合适的圆，并做非极大化抑制。

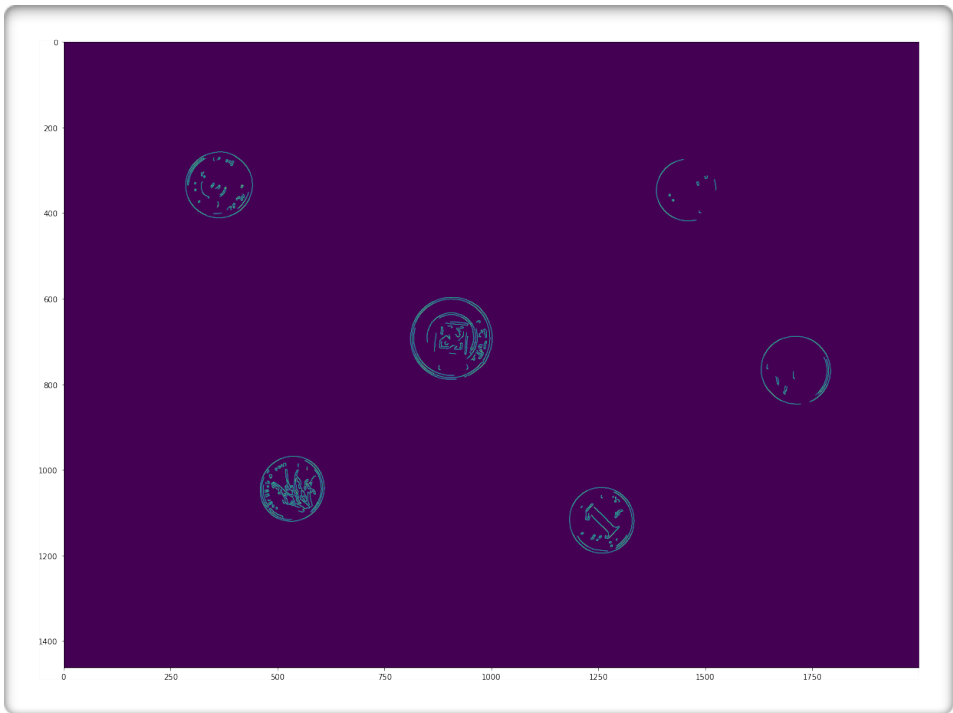
二 函数说明

	参数	参数意义	返回值	函数意义
Canny gaussian_kernel	size	高斯核大小	二维高斯核	对图像应用高斯核，实现高斯模糊处理
	Sigma	标准差，默认为1		
Canny sobel_filters	Img	灰度图像	边缘梯度和方向，二者互相垂直	利用 sobel 算子求图像梯度
Canny non_max_suppression	Img	sobel 生成图像	图像矩阵 Z	非极大化抑制，细化边缘。在每个像素梯度方向检查极大值，如果是则保留（非特定角度则采用软像素）。
	D	角度（方向）矩阵		
Canny threshold_hysteresis	Img	图像矩阵 Z	图像矩阵 Z	滞后门限，去除伪边。原理是先用高阈值删选出一定是边缘的像素，再用低阈值进行边的延伸。遍历周围 8 个像素点，如果有大于高阈值的像素点存在，那么该像素点就视为延伸出来的边，将其像素拉高，并将此改变返回给图像，以此类推。没有延伸到的像素将其像素值置为 0 即可。
	highThreshold	梯度最大值边界，大于是真边缘		
	lowThreshold	梯度最小值边界，小于与真边缘相连		
	Weakpixel	强像素		
	Strongpixel	弱像素		
	Img	图像矩阵 Z		
Hough HoughCircle	Img	Canny 边缘处理后的图像	有标记的输出图像圆心和半径	Hough 检测圆形，先建立参数空间投票，再筛选结果并做非极大化抑制。
	rad	梯度方向矩阵		

	参数	参数意义	返回值	函数意义
	filter_size	三维滤波器		注意到本例可以加入圆心距离判定器以避免单个硬币多个检测点的情况，不过本例通过调参将结果收敛到 6 个
	min_rad	最小半径		
	max_rad	最大半径		
Utils rgb2gray	rgb	图片	灰度图片	将 rgb 图像转化为灰度图像
Utils load_data	img	文件	图片	从指定路径读入 img 图片
Utils visualize	img	图片		图片可视化
Utils neighbor_strong	Z	图片矩阵	图片矩阵 Z	判断邻居像素是否强像素
	M			
	N			
	weak	弱像素值		
	strong	强像素值		

三 实验结果

(1) Canny 算法实现边缘提取结果。



(2) Hough 算法实现圆检测并标记。



(3) Hough 算法返回圆心坐标和半径，共六个。

```
Out[28]: [(362, 334),  
          75,  
          (1459, 346),  
          71,  
          (905, 692),  
          91,  
          (1707, 765),  
          75,  
          (534, 1045),  
          76,  
          (1256, 1118),  
          76]
```