

CV 第二次作业

姓名：黄沈一

学号：3180102081

文件结构和说明

```
hw_2
├── README.md
├── images
│   ├── circle.png
│   ├── circles.png
│   ├── hw-coin.JPG
│   ├── hw-highway.jpg
│   ├── hw-seal.jpg
│   ├── pentagon.PNG
│   └── test.jpg
├── instructions
│   └── Homework-Curve.pdf
├── result
│   ├── processedCoin.jpg
│   ├── processedHighway.jpg
│   └── processedSeal.jpg
└── src
    ├── circleDetection.py
    ├── lineDetection.py
    ├── main.py
    └── utils.py
```

4 directories, 16 files

其中 src 文件夹下为本次作业的代码，其中 main.py 是主文件。作业的剩余部分的作用通过文件夹命名表现了。

程序运行说明

无论是windows还是macOS还是linux环境下，只需要运行 `main.py` 即可：

```
python main.py
```

程序展示内容说明

程序运行后，需要一定时间的等待，随后出现图片顺序为：

1. 高速公路原图
2. 高速公路边缘检测后的图像
3. 高速公路直线拟合的图像
4. 印章原图
5. 印章边缘检测后的图像
6. 印章圆形拟合后的图像
7. 硬币原图
8. 硬币边缘检测后的图像
9. 硬币直线拟合后的图像

在所有的图像展示之后，该程序会将三张图片的最终结果输出在 `result` 文件夹中。

说明

由于目前的算法没有很好的泛化能力，为了提高测试图片的识别精准度，所以在实现拟合的过程中不停调试hyper parameter的数值（尽管这样会导致overfitting）。因此，在作业中直接将对应图片的检测过程打包成一个函数（为了包括测试好的hyper parameters）。

程序思路

转换为灰度图

直方图均衡化

高斯滤波

Canny边缘检测

投票过程

绘图过程

具体实现细节

直线投票

将边缘检测图中的点投射到(P, theta)空间

圆心投票

1. 通过Sobel算子计算出点梯度方向
2. 计算通过该点梯度方向上的直线
3. 直线上的点进行投票

半径投票

1. 找到一个圆心
 2. 从 (r_{\min}, r_{\max}) 不断增加 r ，计算半径为 r 的圆所经过的点的数量
 3. 将数量除以半径
-

一些问题

1. 圆心检测

对于印章的图片，左上角的印章无法检测到圆心，所以无法绘制。

2. 代码泛化能力

目前的代码，都是过拟合的，只能在某张图片上达到比较好的结果。今后会尝试泛化能力较高的算法。