Lab03 实验报告

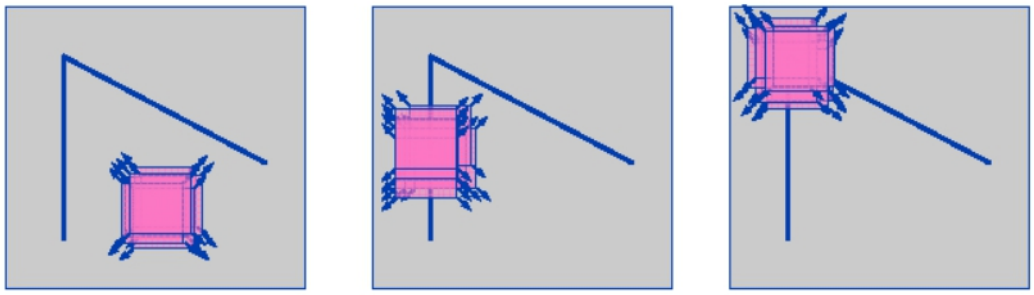
南开大学软件学院软件工程专业 2019 级 2 班 韦诗睿 1913184

Ex 4.1: Interest point detector Implement Forstner–Harris Hessian (try different formula variants given in (4.9–4.11))

$$\det(\boldsymbol{A}) - \alpha \operatorname{trace}(\boldsymbol{A})^2 = \lambda_0\lambda_1 - \alpha(\lambda_0 + \lambda_1)^2 \qquad (4.9)$$

I Introduction

Feature Detection 和 matching 一直是许多 cv 应用的重要组成部分。有时我们希望能够对齐两幅图像，使得它们可以拼接成一张组合图；有时我们希望构建点和点之间的密切关系，以此来构建 3D 模型。但图像点的 matching 并不是容易的事。首先我们需要从两张图像中提取特征点集，并使用向量的形式描述，如果对应特征向量之间的距离小于阈值 T，则认为两者 match。虽然我们希望能够找到对应的特征点，但我们事先并不知道是否存在；因此自然的，我们希望两图中所提取到的特征点尽量 stable/distinctive，不会因为一点扰动而消失失去匹配。所谓的 distinctive 在图像中定义为：若一个滑窗在某像素点的位置向任意方向滑动，都会带来显著的亮度变化，那么该点是 distinctive 的。而图像中的 Corner 点比起 flat\edge 更加符合这个要求[1]，因此 matching 任务最终细化到 Corner Detection。本实验主要探究 Corner Detection 中的 Harris corner detector 方法。



“flat” region: no change in all directions

“edge”: no change along the edge direction

“corner”: significant change in all directions

[1] 图像来源：https://www.cse.psu.edu/~rtc12/CSE486/lecture06.pdf

Ⅱ Strategy

由于 Corner 点的特征为亮度在所有方向上有显著变化，因此我们希望首先找到一个公式可以衡量窗口在图像上移动带来的亮度变化，则有 auto-correlation function：

$$E_{\mathrm{AC}}(\Delta \boldsymbol{u}) = \sum_{i} w(\boldsymbol{x}_i)[I_0(\boldsymbol{x}_i + \Delta \boldsymbol{u}) - I_0(\boldsymbol{x}_i)]^2 \qquad (4.2)$$

其中，w 为窗口函数，一般为 0/1 二值函数或高斯函数；I 为窗口滑动前后的亮度。这个函数可以帮助我们通过改变窗口微小的滑动 u 来观察亮度变化 E。但使用这个公式计算 E 的成本较高，我们希望能够对 E 进行近似操作。

因此有方法提出了使用二次曲面局部近似 E，经过泰勒展开等一系列过程推导，最终得到了 E 的近似：

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

其中：u,v 为 x 和 y 方向上窗口的位移。而 M 为：

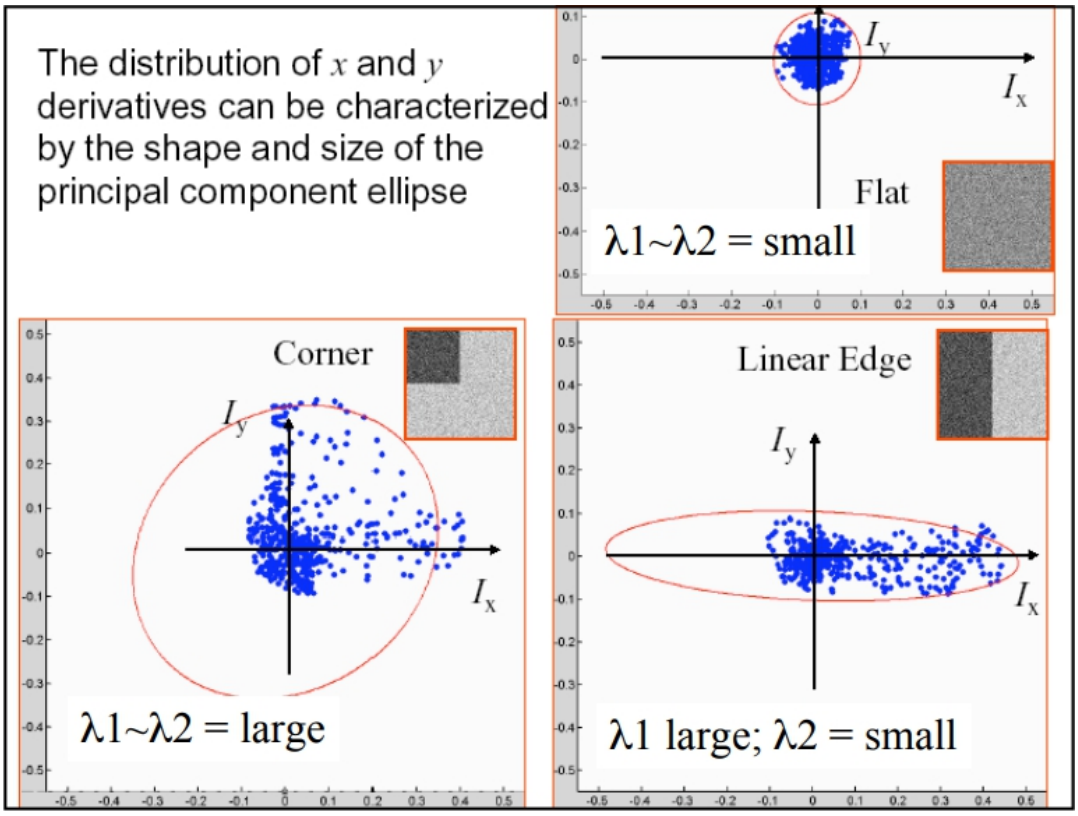
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

可以看到 M 为半正定的对称矩阵，将其化为正交矩阵后有：

$$M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

综合这三个公式，我们可以得到：当 E 随着 u,v 的改变而不断变化时，会形成一个切片为椭圆的二次曲面，其中每一个切片椭圆的朝向将由 M 的 R 来决定，椭圆的长短轴即变化最快和最慢的方向将由 M 的 λ 决定。通过 λ 指标变化，我们可以衡量不同点与周围点的亮度差异如图所示[2]：

[2] 图像来源：https://www.cse.psu.edu/~rtc12/CSE486/lecture06.pdf



我们希望能够使用一个公式综合两个 λ 的数值变化规律，就有了本次实验主要研究的公式：

$$C = \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

其中，alpha 取值在 0.04 和 0.06 之间，C 的数值可以衡量某点是否为 Corner，取名为 Cornerness，当 C>0 时证明该点为 Corner，C=0 时该点为 flat region，C<0 时该点为 edge。

但在计算的过程中我们发现，求 M 的特征值的过程较为繁琐，而由于一个矩阵的行列式和迹为：

$$\det(A) = \prod_{i=1}^{n} \lambda_i = \lambda_1\lambda_2\cdots\lambda_n .$$
$$\operatorname{tr}(A) = \sum_i \lambda_i .$$

因此综合上式我们可以得到新的 Cornerness 计算公式：

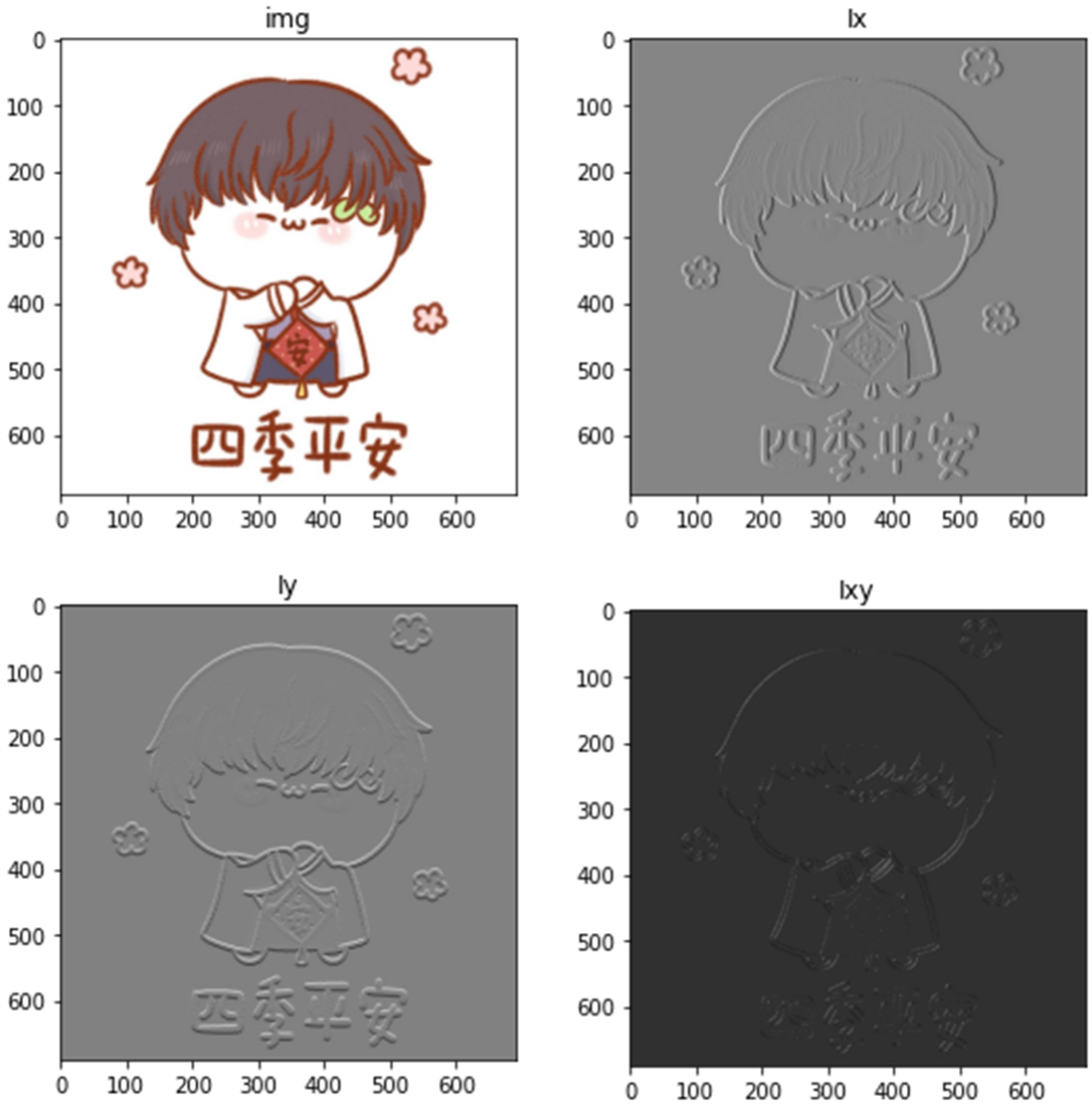
$$C = \det(M) - \alpha\,\operatorname{trace}(M)^2$$

通常还会再经过一个阈值筛选的过程，降低 corner 候选点的数量，使得 corner 同时是局部的最大值点。

Ⅲ Implementation

以上所述过程的代码实现如下：

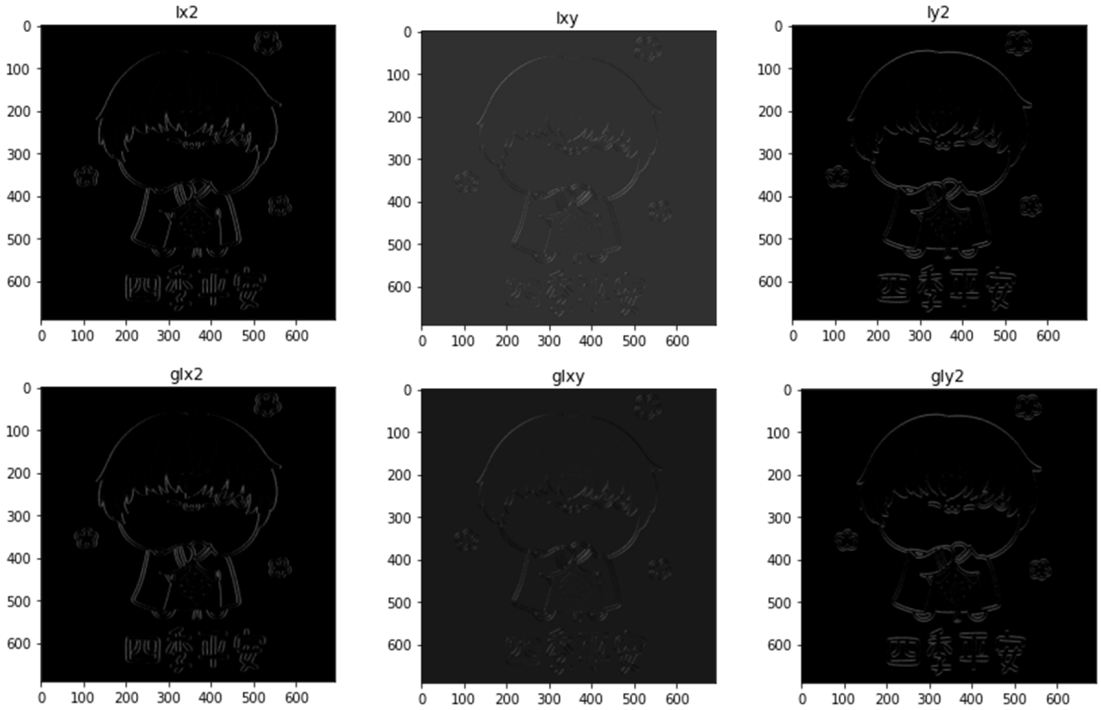
首先导入图片并计算 image derivatives：



易混淆的是对 x 求偏导凸显的是竖向的边；而对 y 求偏导凸显的是横向的边，可以从图中的“四季平安”字样看出两者差别。

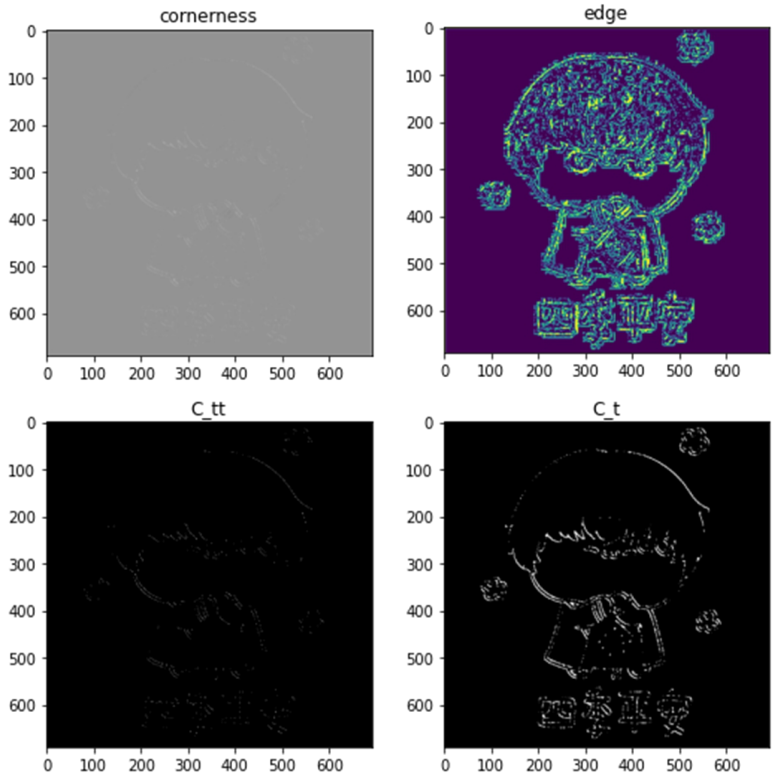
在实现的过程中将图片缩小至 0/1 区间，方便后续运算。

计算 M 的组成元素并经过高斯滤波：



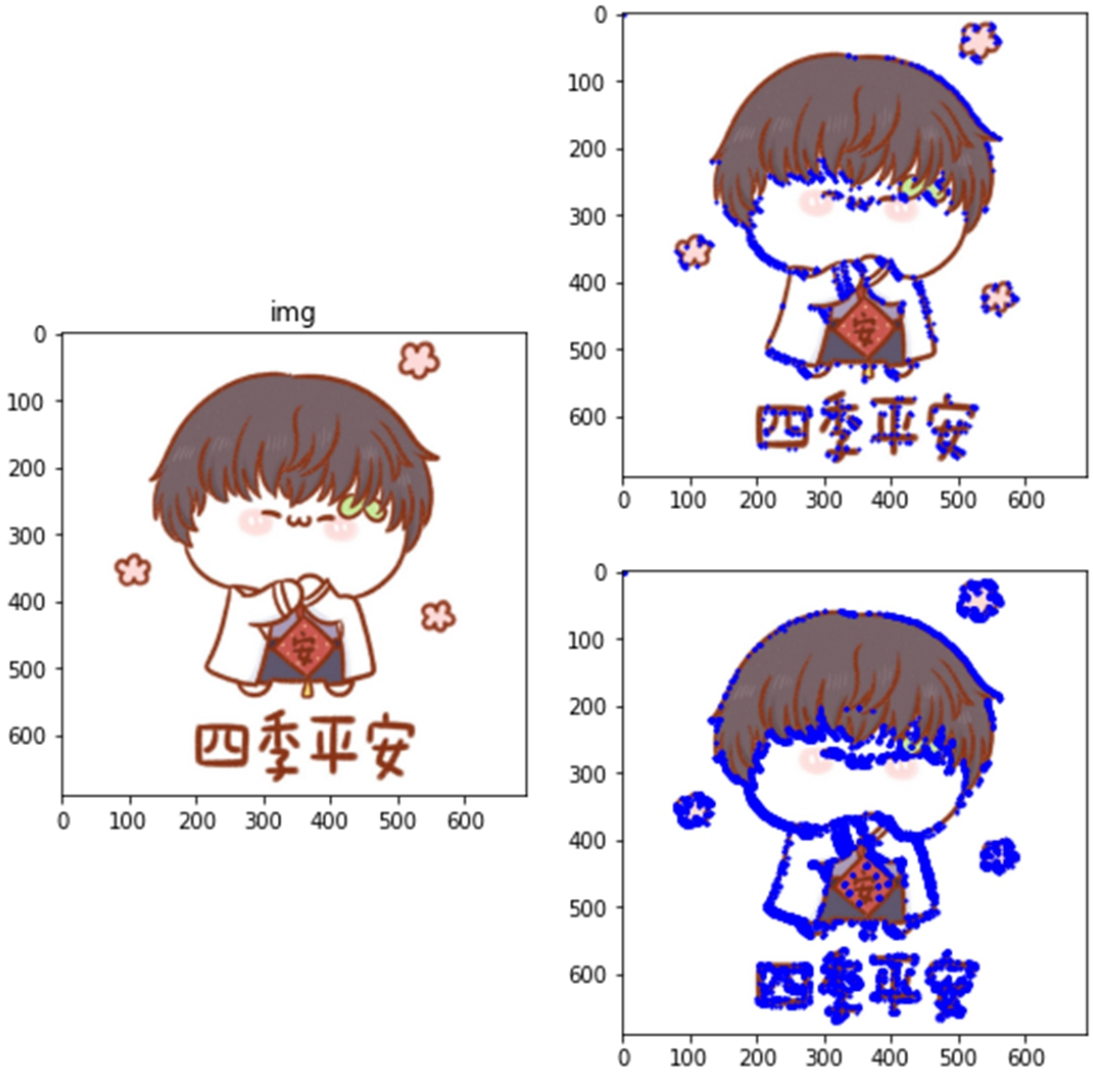
计算 Cornerness 并筛选得到 edge 点和 corner 点：

其中尝试了两种筛选方法，其中一种为根据每个 3*3 的窗口区域选出候选 Corner 点中的极大值点；另一种为根据图像数值的分布取 98%的分位点。两种方法比较，有前者筛选得到的 corner 点数量远小于后者。



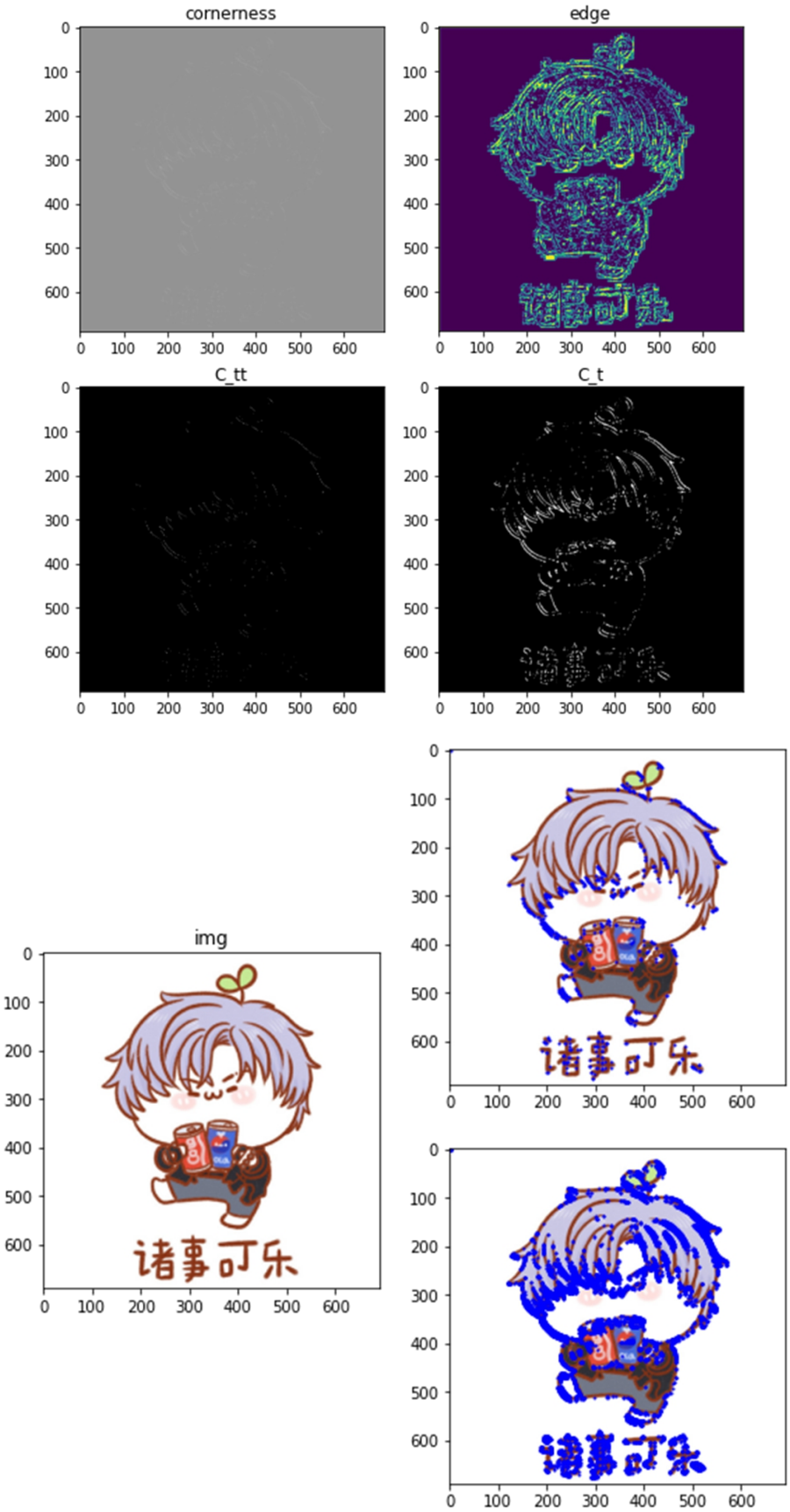
如上图，C_tt 是第一种筛选方法得到的角点；C_t 为第二种方法。

结果展示：



其中每种方法筛选出的角点在原图上以蓝色点进行标识，可以很明显地看到第二种方法筛选剩余的角点数量较多。

再尝试另一幅图像，有：



Ⅳ Conclusion

本次实验探究了图像处理中的 Corner Detection 问题，了解了 Edge 点以及 Corner 点的检测方法，实验结果也达到了预期。在接下来的学习中我将应用本节学到的相关知识更深入的学习 Feature Detection 和 Matching 的相关内容。

本次实验的代码可见于：*https://github.com/RuiNov1st/NKUComputerVision*

Ⅴ References

[1] Szeliski, R., Computer Vision: Algorithms and Applications. London: Springer, 2010.

[2] Robert Collins, CSE486, Lecture 06: Harris Corner Detector, https://www.cse.psu.edu/~rtc12/CSE486/lecture06.pdf

[3] Harris, Christopher G. and M. J. Stephens. “A Combined Corner and Edge Detector.” Alvey Vision Conference (1988).

[4] Alexei Efros, CS194: Convolution and Image Derivatives, https://inst.eecs.berkeley.edu/~cs194-26/fa21/Lectures/ConvolutionAndDerivatives.pdf