

## 实验 4-A.边缘检测和角点检测

参考钉钉群 `edge_demo.ipynb`，完成如下内容。

1. 打开 `Flower.png`，用 `cv2.cvtColor()` 将其转换成灰度图像，以 `Canny()` 函数对图像提取边缘，显示边缘图像。结合实验过程，理解各个参数的含义。

2. 在 1 的基础上，以 `Canny()` 函数返回的结果图像为输入，进一步使用 `findContours()` 函数寻找图像中的轮廓，且以 `drawContours()` 绘制轮廓。上述两个函数的用法可参见 [https://docs.opencv.org/4.x/d4/d73/tutorial\\_py\\_contours\\_begin.html](https://docs.opencv.org/4.x/d4/d73/tutorial_py_contours_begin.html)。

## 实验 4-B.直线检测

参考钉钉群 `Hough_Line.ipynb`，完成如下内容。

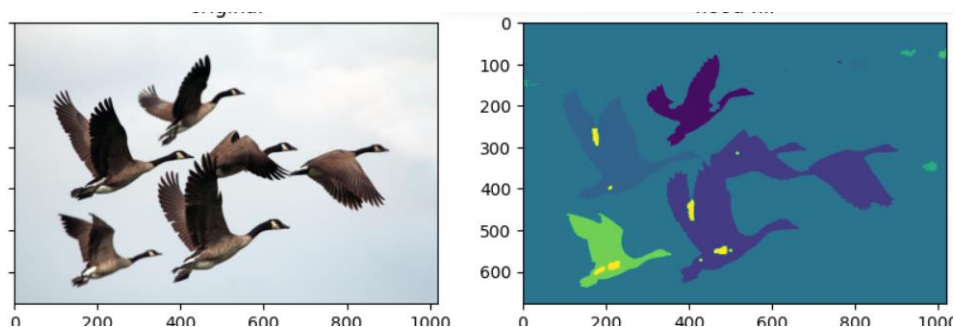
1. 打开 `TestFigures` 目录中的 `building.png`，将其转换成灰度图像，以 `Canny()` 函数获取二值化的边缘图像，以 `cv2.HoughLinesP()` 提取并画出其中较长的 5 条直线。

2. 分别以 `kernel_size=3, 5, 7, 9` 对 1 中打开的图像作中值滤波处理 (`cv2.medianBlur()`)，重复 1 过程观察检测得到的结果

## 实验 3-C.基于区域的图像分割

参考钉钉群 `region_segment.ipynb`，完成如下内容。

1. 打开 `TestFigures` 目录中的 `BirdFlying.png`，综合运用 `adaptiveThreshold()` 和 `floodFill()`，为分割所得的每只鸟的区域着不同的颜色，结果可参见下图所示。



2. 打开 `TestFigures` 目录中的 `region_bright.jpg`，根据裂纹区域具有高亮的特性，通过直方图分析确定种子点，以 `cv2.floodFill()` 对裂纹区域进行漫水填充分割，根据分割结果确定以像素计的裂纹宽度和高度。