

实训二：Python 图像处理基本操作

作者：杨仕龙

1 实验目的

1. 学会对图像进行基本的代数操作（图像的相加、相减、相乘）。
2. 掌握图像的空间域变换操作（图像的插值，缩放，旋转与剪切）。

2 实验内容

1. 图像的代数操作

- (a) 选取 Baboon.bmp 和 lena.bmp 两幅图像,完成对图像的相加运算;选取 lena.bmp 和 elaine.512.tiff 完成相减运算。完成之后的效果如图1所示。【思考：图像的加减与矩阵的加减有何联系?】
- (b) 使用乘法运算，使用 elaine.512.tiff 和完成掩膜的功能。完成之后的结果如图1(Image Multiply)所示。

2. 图像的空间域变换操作（图像的插值，缩放，旋转与剪切）

- (a) 对一幅图像进行放大操作 【cv2.resize(InputArray src, OutputArray dst, Size, fx, fy, interpolation)】，并使用最邻近插值法，双线性插值法和双三次插值法（INTER_NEAREST, INTER_LINEAR, INTER_CUBIC）对放大的效果进行比较。最后，你应该粘贴保存的结果至实验报告，如图2所示。
 - (b) 对图像 want_want.jpg 进行旋转插值操作 【cv2.getRotationMatrix2D 和 cv2.warpAffine】
 - (c) 对图像 want_want.jpg 进行剪切操作 【利用分片】。(b) 和 (c) 的结果如图3所示。
3. 图像灰度直方图绘制和直方图均衡化。选取 boat.512.tiff 作为原图，绘制原图和直方图均衡化之后的图像，并分别显示二者的灰度直方图。如果一切正常，你将看到如图4所示的结果。

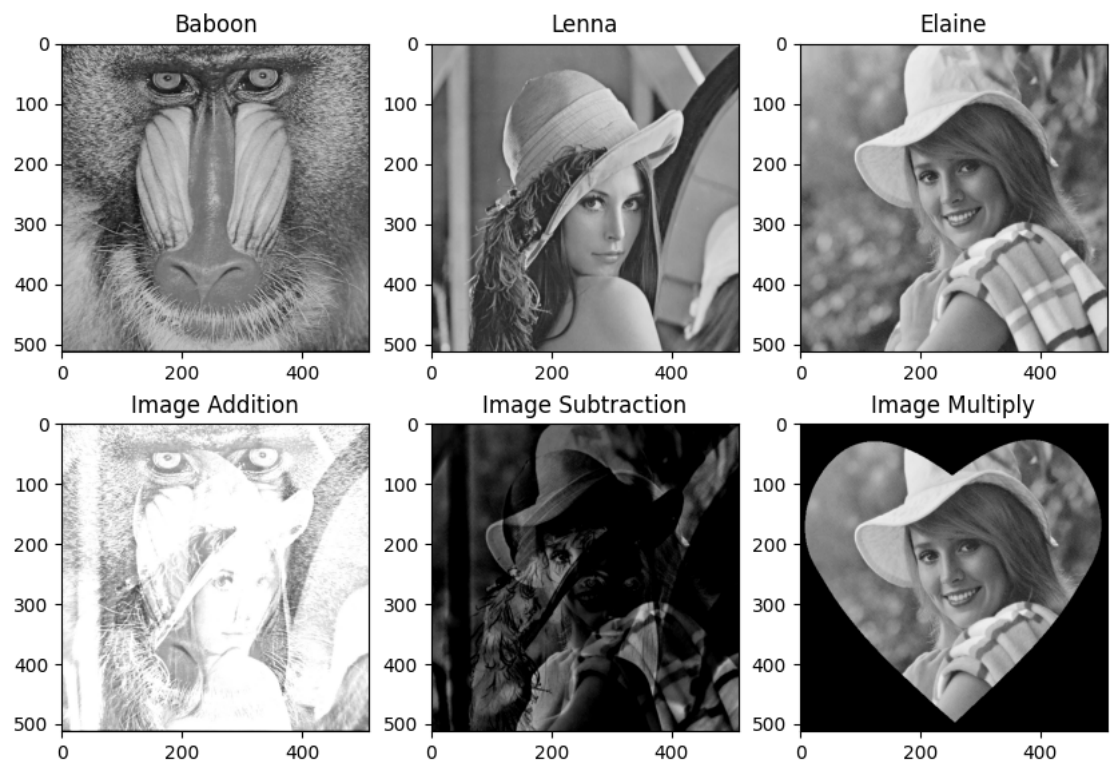


图 1: 图像四则运算

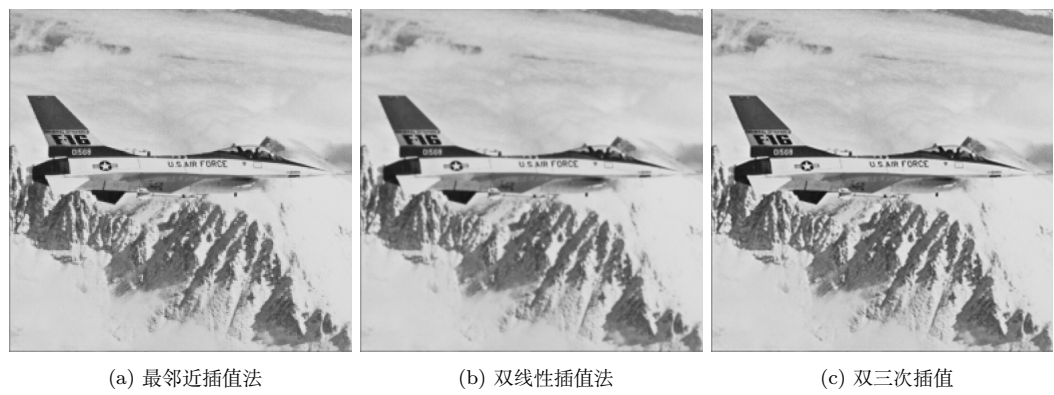


图 2: 图像放大插值

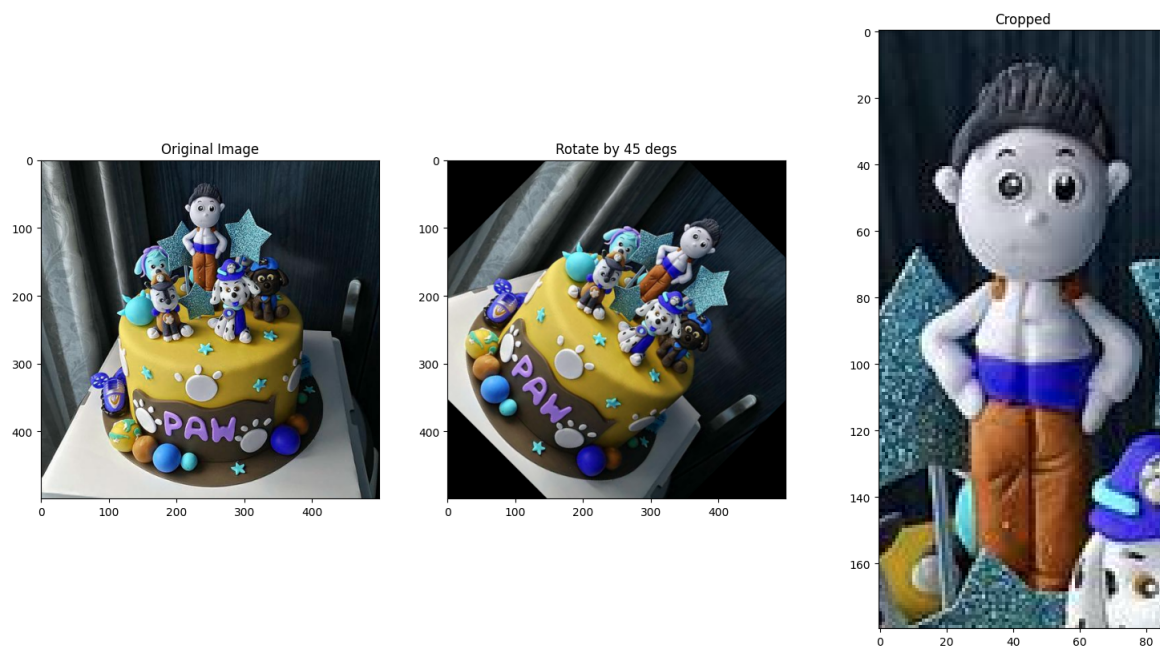


图 3: 图像旋转与裁减

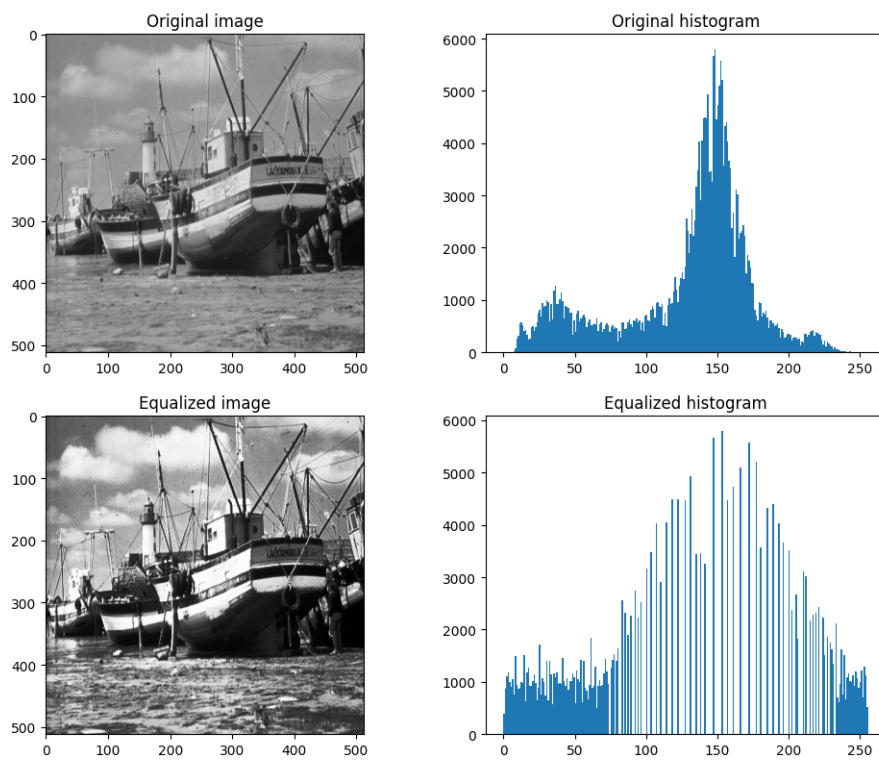


图 4: 图像灰度直方图绘制和直方图均衡化