广州航海学院

数字图像处理及应用实验 实验报告

|  |  |
| --- | --- |
| 成绩 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 专业班级 | 计科222 | 实验日期 | 2025.04.03 |
| 姓名 | 钟尹泽 | 学号 | 202215210229 |
| 实验名称 | 实验2.空域图像增强 | 指导老师 | 华国光 |

# 实验目的

1. 掌握灰度变换和点运算的实现方法；
2. 理解并掌握空域平滑和锐化滤波方法；
3. 理解点运算与邻域运算的差别与联系。 # 实验要求
4. 能够熟悉编写基于OpenCV和Numpy工具包的图像处理程序；
5. 程序的运行能够成功通过，无代码错误；
6. 训练模型，并验证程序的正确性。

# 实验环境

WINDOWS7及以上系统，Pycharm或VScode等开发工具，Python3.7及以上版本，OpenCV-Python 4.1.0.25及以上环境。

# 实验内容

1. 编程实现图像的灰度变换，包括图像变暗、图像变亮、降低对比度和直方图均衡化处理； img[i, j] = ((d-c) / (maxV - minV)) \* (img[i, j] - minV)+c
2. 编程实现gamma变换，图片效果参考教材附录-图像处理实验指导；
3. 编写程序，首先对图像添加椒盐噪声或高斯噪声，然后对加噪图像进行均值滤波、中值滤波和高斯滤波；cv2.filter2D(src, ddepth,kernal[,dst[,anchor,delta[,borderType]]])以上实验内容，代码参考教材附录-图像处理实验指导P220,222-223。
4. 编程实现拉普拉斯锐化和拉普拉斯增强，要求使用拉普拉斯锐化滤波核：laplace = np.array([[0, 1, 0], [1, -4, 1], [0, 1, 0]])，以及要求使用拉普拉斯锐化增强滤波核：laplace = np.array([[0, -1, 0], [-1, 5, -1], [0, -1, 0]])。
5. 编程实现拉普拉斯锐化和拉普拉斯强化

# 源代码及代码注释或解释

**1. 灰度变换及直方图均衡化**

import cv2  
import numpy as np  
import matplotlib.pyplot as plt  
  
# 读取图像  
img = cv2.imread('iris.jpg', 0)  
  
# 全局线性变换实现图像变亮 (例如乘以1.5)  
img\_bright = np.clip(img \* 1.5, 0, 255).astype(np.uint8)  
  
# 全局线性变换实现图像变暗 (例如乘以0.5)  
img\_dark = np.clip(img \* 0.5, 0, 255).astype(np.uint8)  
  
# 降低对比度 (例如线性变换压缩像素范围)  
img\_low\_contrast = np.clip((img - 128) \* 0.5 + 128, 0, 255).astype(np.uint8)  
  
# 直方图均衡化  
img\_equalized = cv2.equalizeHist(img)  
  
# 显示结果  
plt.figure(figsize=(12, 8))  
plt.subplot(231), plt.imshow(img, cmap='gray'), plt.title('Original Image')  
plt.subplot(232), plt.imshow(img\_bright, cmap='gray'), plt.title('Bright Image')  
plt.subplot(233), plt.imshow(img\_dark, cmap='gray'), plt.title('Dark Image')  
plt.subplot(234), plt.imshow(img\_low\_contrast, cmap='gray'), plt.title('Low Contrast Image')  
plt.subplot(235), plt.imshow(img\_equalized, cmap='gray'), plt.title('Equalized Image')  
  
# 绘制直方图  
plt.figure(figsize=(12, 8))  
plt.subplot(231), plt.hist(img.ravel(), 256, [0, 256]), plt.title('Original Histogram')  
plt.subplot(232), plt.hist(img\_bright.ravel(), 256, [0, 256]), plt.title('Bright Histogram')  
plt.subplot(233), plt.hist(img\_dark.ravel(), 256, [0, 256]), plt.title('Dark Histogram')  
plt.subplot(234), plt.hist(img\_low\_contrast.ravel(), 256, [0, 256]), plt.title('Low Contrast Histogram')  
plt.subplot(235), plt.hist(img\_equalized.ravel(), 256, [0, 256]), plt.title('Equalized Histogram')  
  
plt.show()

**2. Gamma 变换**

import cv2  
import numpy as np  
import matplotlib.pyplot as plt  
  
def gamma\_trans(img, gamma):  
 # 具体做法是先归一化到1, 然后gamma作为指数值求出新的像素值再还原  
 gamma\_table = [np.power(x / 255.0, gamma) \* 255.0 for x in range(256)]  
 gamma\_table = np.round(np.array(gamma\_table)).astype(np.uint8)  
 # 实现这个映射用的是OpenCV的查表函数  
 return cv2.LUT(img, gamma\_table)  
  
# 读取图像  
img = cv2.imread('iris.jpg', 0)  
  
# gamma小于1的图像  
img\_gamma\_less = gamma\_trans(img, 0.5)  
  
# gamma大于1的图像  
img\_gamma\_more = gamma\_trans(img, 2)  
  
# 显示结果  
plt.figure(figsize=(12, 8))  
plt.subplot(131), plt.imshow(img, cmap='gray'), plt.title('Original Image')  
plt.subplot(132), plt.imshow(img\_gamma\_less, cmap='gray'), plt.title('Gamma < 1 Image')  
plt.subplot(133), plt.imshow(img\_gamma\_more, cmap='gray'), plt.title('Gamma > 1 Image')  
plt.show()

**3. 图像空域滤波**

import matplotlib.pyplot as plt  
import cv2  
import numpy as np  
  
# 读取图像  
img = cv2.imread('kennysmall.jpg', 0)  
cv2.imshow('original image', img)  
  
# 设置滤波核  
fil1 = 1 / 16 \* np.array([[1, 2, 1],  
 [2, 4, 2],  
 [1, 2, 1]])  
  
# 使用OpenCV的卷积函数  
ImgSmoothed = cv2.filter2D(img, -1, fil1, borderType=cv2.BORDER\_DEFAULT)  
  
cv2.imshow('Smoothed image', ImgSmoothed)  
cv2.waitKey(0)  
cv2.destroyAllWindows()

**4. 图像添加椒盐噪声**

import cv2  
import numpy as np  
import random  
  
def addSaltAndPepper(src, percentage):  
 NoiseImg = src.copy()  
 NoiseNum = int(percentage \* src.shape[0] \* src.shape[1])  
 for i in range(NoiseNum):  
 randX = random.randint(0, src.shape[0] - 1)  
 randY = random.randint(0, src.shape[1] - 1)  
 if random.randint(0, 1) == 0:  
 NoiseImg[randX, randY] = 0  
 else:  
 NoiseImg[randX, randY] = 255  
 return NoiseImg  
  
# 读取图像  
img = cv2.imread('kennysmall.jpg', 0)  
# 添加椒盐噪声，例如噪声比例为0.05  
noisy\_img = addSaltAndPepper(img, 0.05)  
  
# 均值滤波  
mean\_filtered = cv2.blur(noisy\_img, (3, 3))  
# 中值滤波  
median\_filtered = cv2.medianBlur(noisy\_img, 3)  
# 高斯滤波  
gaussian\_filtered = cv2.GaussianBlur(noisy\_img, (3, 3), 1)  
  
# 显示结果  
cv2.imshow('Original Image', img)  
cv2.imshow('Noisy Image', noisy\_img)  
cv2.imshow('Mean Filtered Image', mean\_filtered)  
cv2.imshow('Median Filtered Image', median\_filtered)  
cv2.imshow('Gaussian Filtered Image', gaussian\_filtered)  
cv2.waitKey(0)  
cv2.destroyAllWindows()

**5. 拉普拉斯锐化和拉普拉斯增强**

import cv2  
import numpy as np  
import matplotlib.pyplot as plt  
  
# 读取图像  
img = cv2.imread('moon.jpg', 0)  
  
# 拉普拉斯锐化滤波核  
laplacian\_kernel = np.array([[0, 1, 0],  
 [1, -4, 1],  
 [0, 1, 0]])  
# 拉普拉斯锐化增强滤波核  
laplacian\_enhance\_kernel = np.array([[0, -1, 0],  
 [-1, 5, -1],  
 [0, -1, 0]])  
  
# 拉普拉斯锐化  
laplacian\_img = cv2.filter2D(img, -1, laplacian\_kernel)  
# 拉普拉斯锐化增强  
laplacian\_enhance\_img = cv2.filter2D(img, -1, laplacian\_enhance\_kernel)  
  
# 显示结果  
plt.figure(figsize=(12, 8))  
plt.subplot(131), plt.imshow(img, cmap='gray'), plt.title('Original Image')  
plt.subplot(132), plt.imshow(laplacian\_img, cmap='gray'), plt.title('Laplacian Image')  
plt.subplot(133), plt.imshow(laplacian\_enhance\_img, cmap='gray'), plt.title('Laplacian Enhance Image')  
plt.show()

# 运行测试结果

**步骤1**

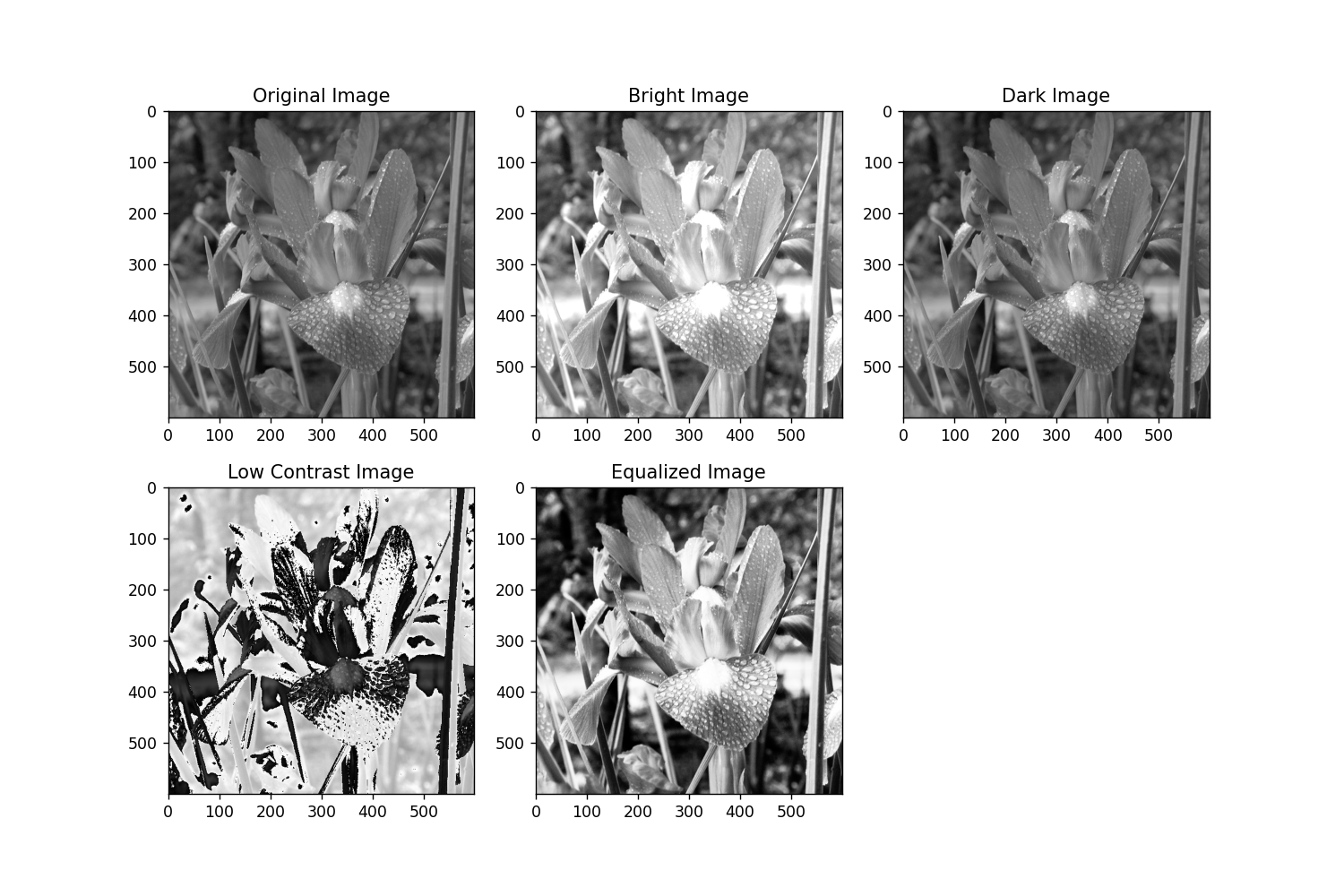


图1

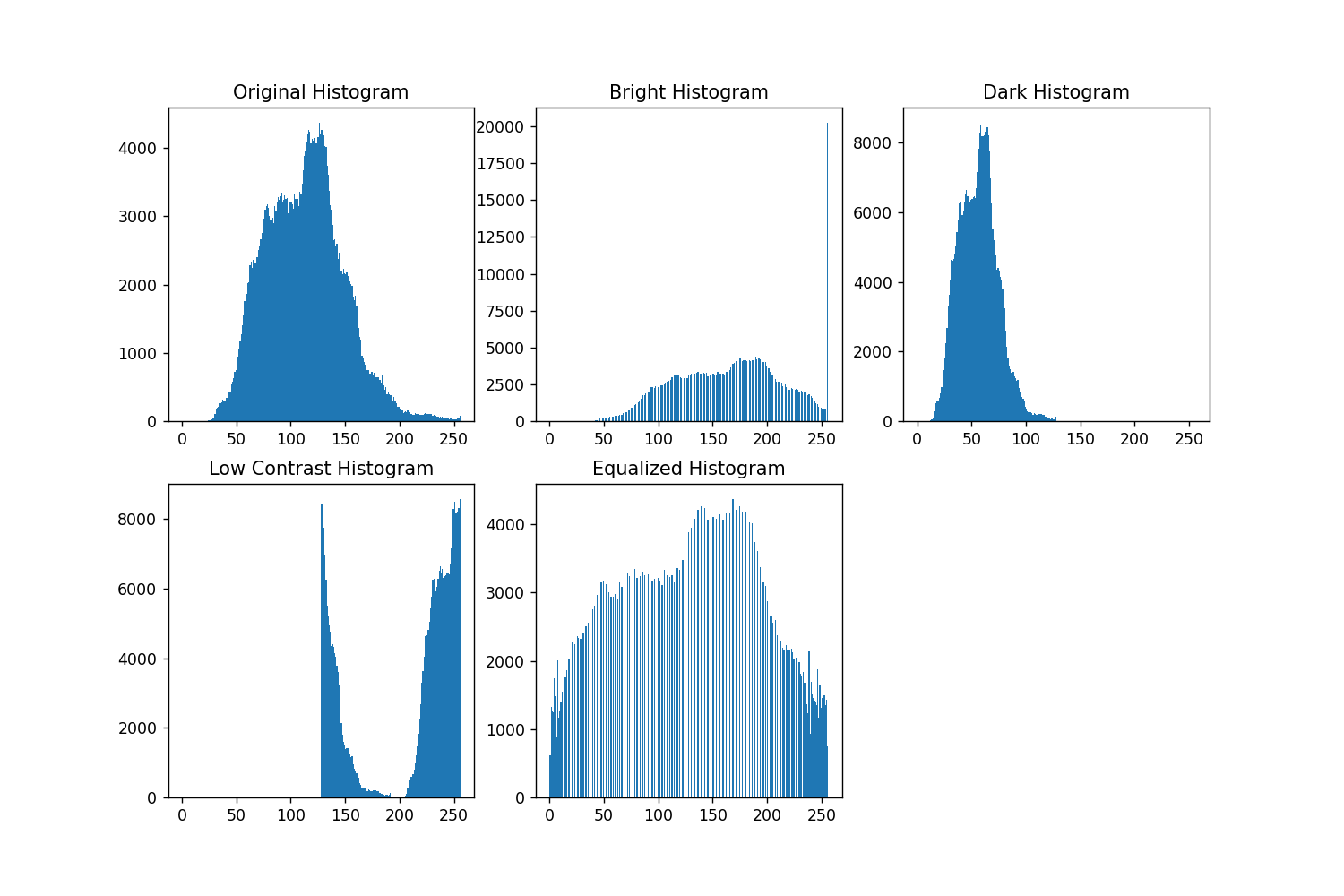


图2

**步骤2**

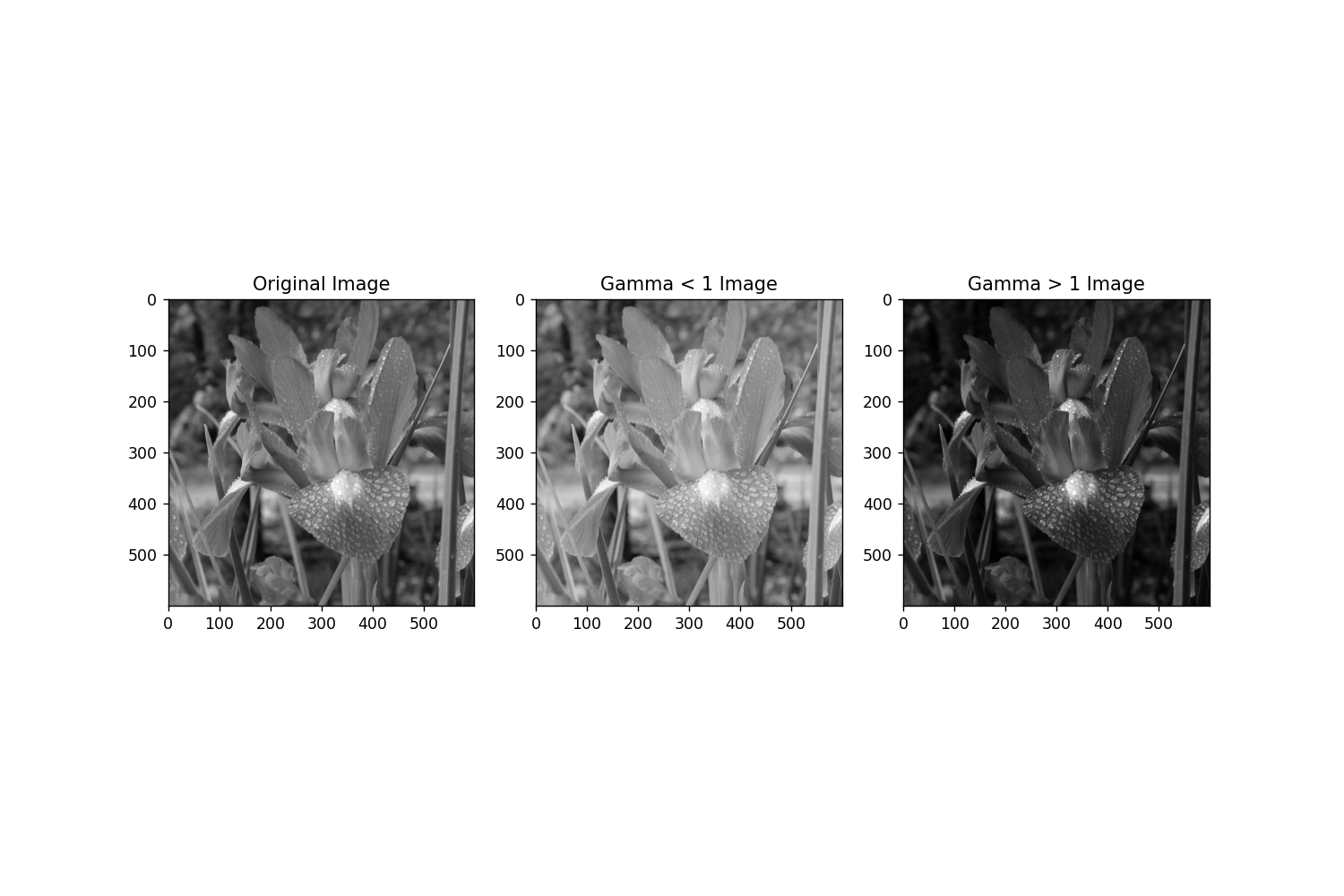


图3

**步骤3**

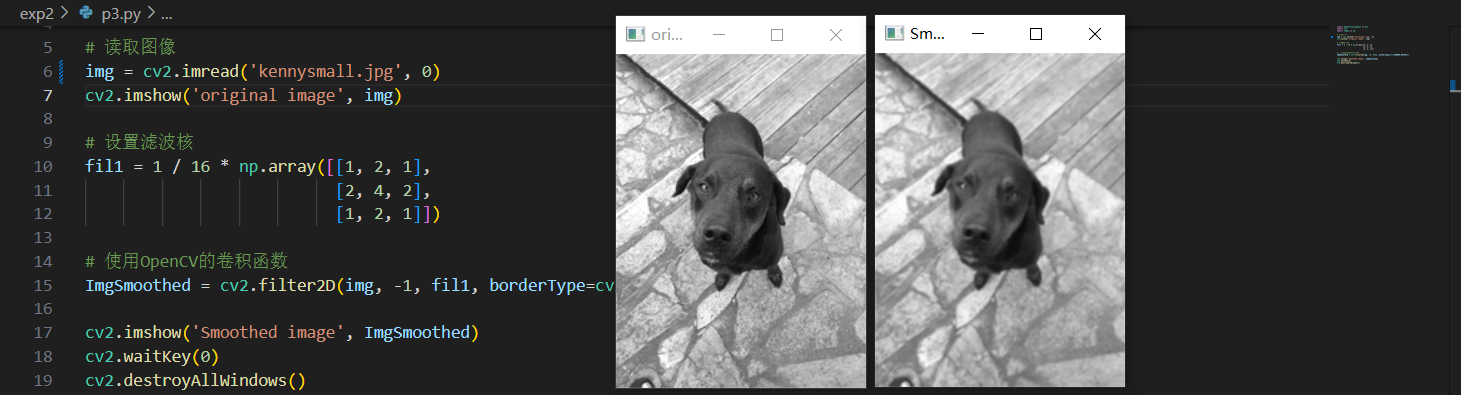


图4

**步骤4**

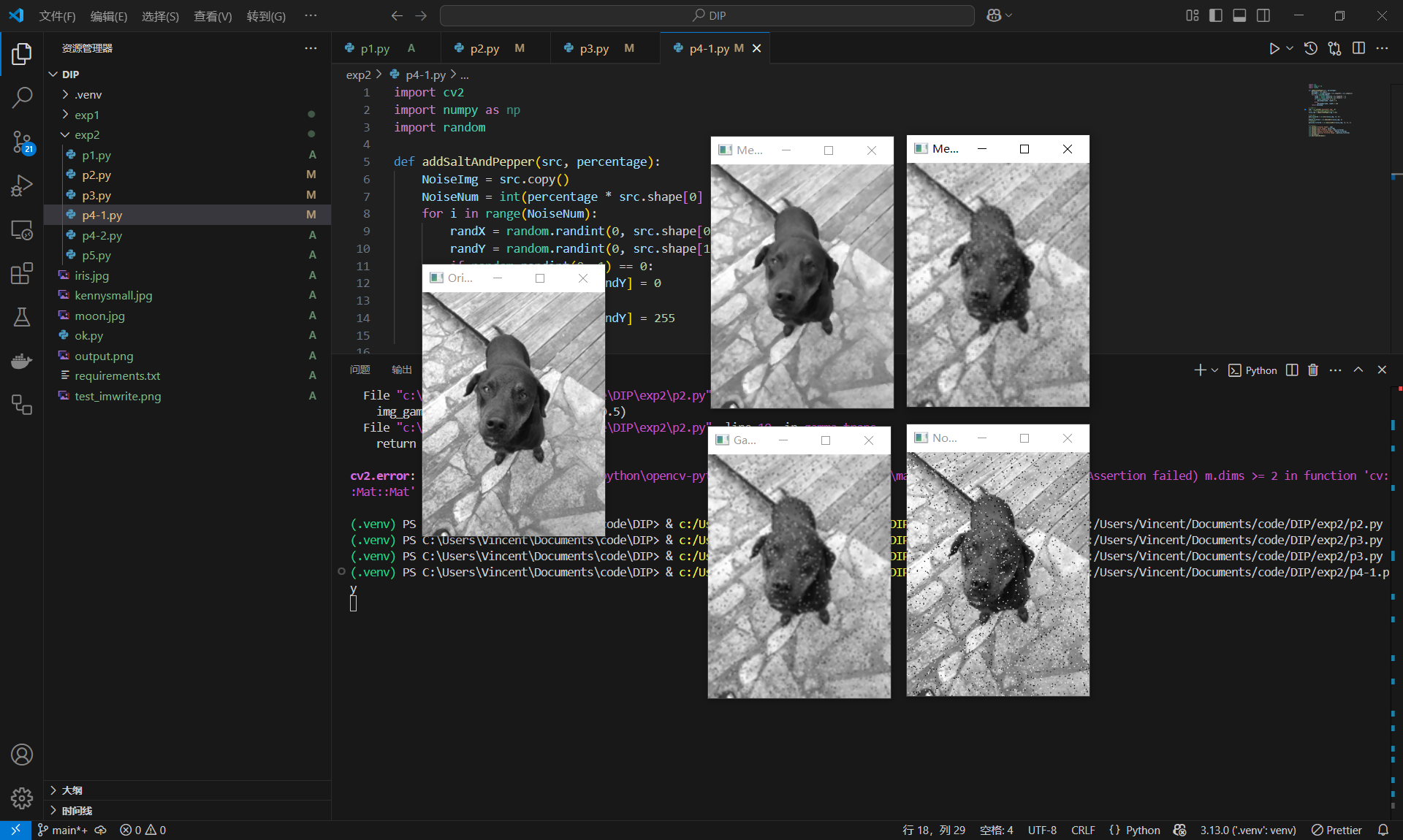


图5

**步骤5**

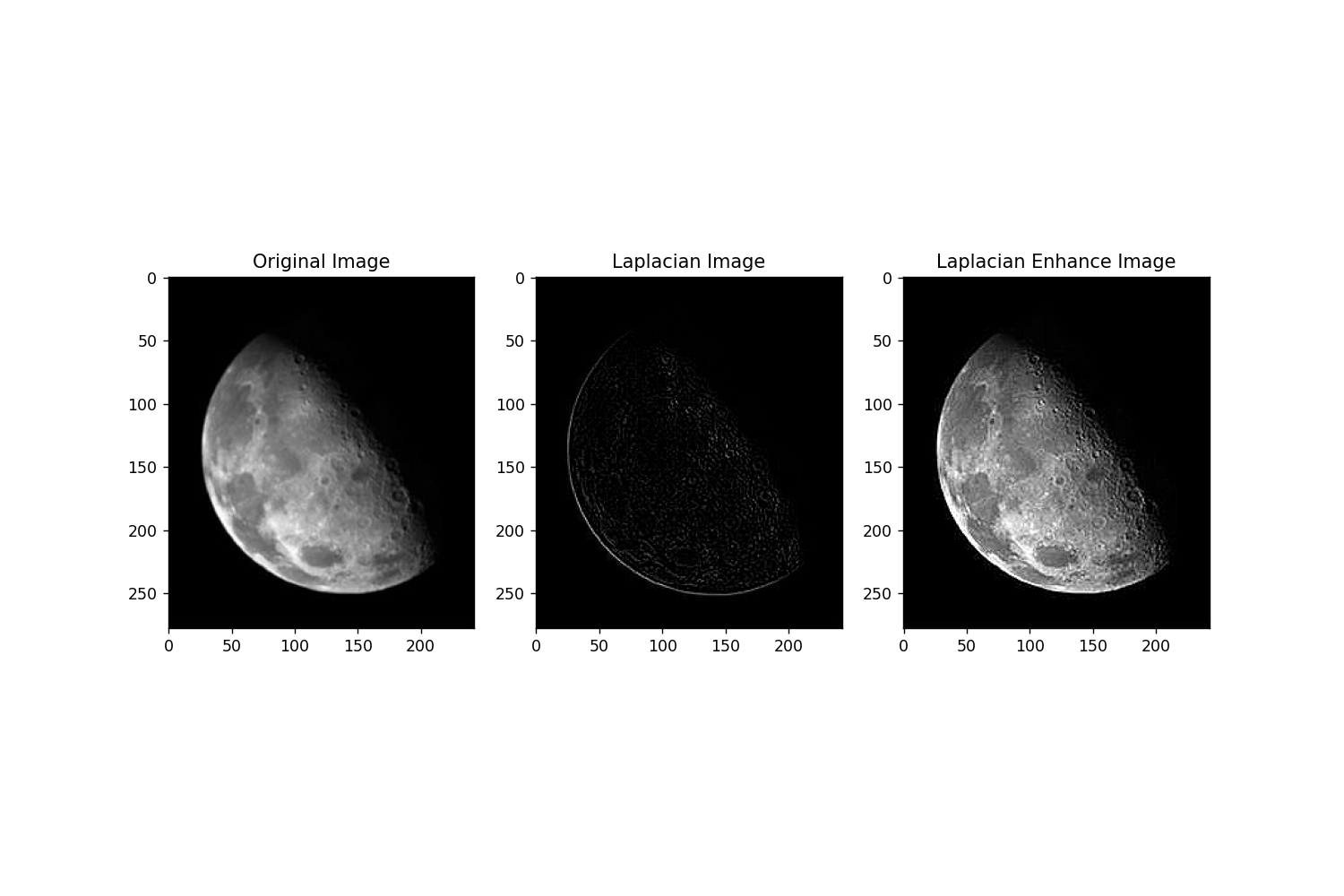


图6

# 实验总结分析

本次实验让我图像处理有了更深入的认识。