

滤除噪声实验

实验目的

试滤除下图中的噪声信号。



实验环境

- OpenCV
- Matplotlib

实验原理

在数字图像处理中，噪声是不可避免的现象，它会对图像的质量产生负面影响。为了提高图像质量，去噪技术常用于减少噪声并保留图像细节。

噪声

噪声是图像信号中的随机干扰，主要影响图像的视觉效果。

高斯噪声

高斯噪声是一种广泛存在的随机噪声，其每个像素的噪声值独立地服从正态分布。高斯噪声的概率密度函数为：

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

其中， μ 为均值，通常为0， σ^2 为方差，代表噪声的强度。

泊松噪声

泊松噪声通常出现在光电效应驱动的成像过程中，如医学成像或低光条件下的摄影。它的特点是噪声与信号强度有关。泊松噪声服从泊松分布，其概率密度函数为：

$$p(k, \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

其中， λ 是期望值， k 是实际观测值。

滤波

滤波是用于去除图像噪声的常见方法。通过一个卷积核对图像进行卷积，滤波器能够减小噪声，同时保留图像的基本结构。

高斯滤波

高斯滤波是一种线性平滑滤波器，基于图像局部像素的高斯加权平均来减少高频噪声，特别是高斯噪声。其卷积核是一个二维的高斯函数：

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

其中， σ 控制滤波器的标准差，值越大，滤波器的平滑效果越强。高斯滤波适用于去除图像中的高斯噪声。

在Python中，使用OpenCV实现高斯滤波可以通过 `cv2.GaussianBlur()` 函数实现，其接受一张图片，一个二元组代表卷积核尺寸，以及一个数代表 σ ；如果后者留空为0，会自动计算。

双边滤波

双边滤波是一种非线性滤波器，它不仅考虑像素的空间距离，还考虑像素值的相似度，从而在去噪的同时保留图像边缘信息。双边滤波的卷积核可以表示为：

$$I_{filtered}(x) = \frac{1}{W(x)} \sum_{y \in \Omega} I(y) \cdot \exp\left(-\frac{|x - y|^2}{2\sigma_s^2}\right) \cdot \exp\left(-\frac{|I(x) - I(y)|^2}{2\sigma_r^2}\right)$$

其中， σ_s 控制空间域的权重， σ_r 控制像素值域的权重， $W(x)$ 是归一化常数。双边滤波器在平滑噪声的同时保留了边缘等显著结构。

在Python中，使用OpenCV可以通过 `cv2.bilateralFilter()` 实现双边滤波。第一个参数是输入图像，第二个参数是滤波邻域的直径，后两个参数分别是空间域标准差 σ_s 和像素值域标准差 σ_r 。 σ_s 控制空间距离的影响范围， σ_r 则决定了相似像素值之间的影响权重。

实验步骤

判断噪声类型

图像的噪点并没有太多出现255和0的极端值，所以判断不是椒盐噪声，而是高斯噪声或者泊松噪声。考虑使用高斯滤波和双边滤波去除。

尝试高斯滤波

对5, 7, 9大小的卷积核，分别尝试3, 6, 9, 12的方差。

```
In [ ]: import cv2
import matplotlib.pyplot as plt

%matplotlib inline
img = cv2.imread('lena.png')
fig, ax = plt.subplots(3, 4, figsize=(15, 10))

kernels = [(s, s) for s in range(5, 10, 2)]
sigmas = [3.0, 6.0, 9.0, 12.0]

for i, kernel in enumerate(kernels):
    for j, sigma in enumerate(sigmas):
        img_blur = cv2.GaussianBlur(img, kernel, sigma)
        ax[i, j].imshow(cv2.cvtColor(img_blur, cv2.COLOR_BGR2RGB))
        ax[i, j].set_title(f'Kernel: {kernel}, Sigma: {sigma}')
        ax[i, j].axis('off')

plt.tight_layout()
plt.show()
```



卷积核越大、方差越大，在去噪效果越好的同时，图像也更模糊了。目测7*7卷积核与方差为9取得了比较均衡的效果。

双边滤波

固定卷积核大小为9*9，尝试在75到125内两个方差的组合。

```
In [ ]: import cv2
import matplotlib.pyplot as plt

img = cv2.imread('lena.png')
fig, ax = plt.subplots(4, 4, figsize=(15, 15))

sigmas = [75, 100, 125, 150]

for i, sigma_s in enumerate(sigmas):
    for j, sigma_r in enumerate(sigmas):
        img_blur = cv2.bilateralFilter(img, 9, sigma_s, sigma_r)
        ax[i, j].imshow(cv2.cvtColor(img_blur, cv2.COLOR_BGR2RGB))
        ax[i, j].set_title(f"Space: {sigma_s}, Color: {sigma_r}")
        ax[i, j].axis('off')

plt.tight_layout()
plt.show()
```




至少直观看来，双边滤波更好地保留了图像的清晰度。随着 σ_s 变大，图像变模糊；随着 σ_r 变大，先是图像的边缘变得锐利，之后反而图像产生了一些斑点。(125, 125)的组合效果似乎还是不错的。

实验总结

在这次实验中，使用了高斯滤波和双边滤波来去除图像中的噪声。

- 高斯滤波：随着卷积核和方差的增大，图像变得更加平滑，但也失去了部分细节。最终 **7×7卷积核** 和 **9的方差** 达到了较好的去噪效果与保留细节的平衡。
- 双边滤波：相比高斯滤波，双边滤波在去噪的同时更好地保留了图像边缘。实验结果显示 **(125, 125)** 的参数组合效果最佳。

总体来说，双边滤波在本实验中的表现更优，去噪效果好且能保持图像清晰度。