

CV 实验二

1. 添加噪声

(1) 椒盐噪声

```
def add_salt_pepper(img, prob=0.05):
    """椒盐噪声（双极性）"""
    noisy = np.copy(img)
    # 生成盐噪声
    salt_mask = np.random.rand(*img.shape[:2]) < prob/2
    noisy[salt_mask] = 255
    # 生成胡椒噪声
    pepper_mask = np.random.rand(*img.shape[:2]) < prob/2
    noisy[pepper_mask] = 0
    return noisy
```

(2) 脉冲噪声

```
def add_impulse(img, prob=0.05):
    """脉冲噪声（单极性）"""
    noisy = np.copy(img)
    mask = np.random.rand(*img.shape[:2]) < prob
    noisy[mask] = 255 # 设置为白噪声
    return noisy
```

(3) 高斯噪声

```
def add_gaussian(img, mean=0, sigma=25):
    """高斯噪声"""
    row, col, ch = img.shape
    gaussian = np.random.normal(mean, sigma, (row, col, ch))
    noisy = img.astype(np.float32) + gaussian
    return np.clip(noisy, 0, 255).astype(np.uint8)
```

2. 滤波器

```
def apply_filters(img):
    """三种滤波器实现"""
    # 均值滤波
    mean = cv2.blur(img, (5,5))
    # 高斯滤波
    gaussian = cv2.GaussianBlur(img, (5,5), 0)
    # 中值滤波
```

```
median = cv2.medianBlur(img, 5)
return mean, gaussian, median
```

3. 主函数

```
if __name__ == "__main__":
    # 读取图像（替换为实际路径）
    original = read_image("D:\Samples\cvImage.jpg")

    # 步骤 2: 叠加三种噪声
    noisy = add_salt_pepper(original) # 椒盐
    noisy = add_impulse(noisy)        # 叠加脉冲
    noisy = add_gaussian(noisy)       # 叠加高斯

    # 步骤 3: 应用三种滤波器
    mean_filtered, gaussian_filtered, median_filtered =
    apply_filters(noisy)

    plt.figure(figsize=(12, 8)) # 调整画布尺寸

    # 第一行显示原始图像和噪声图像
    plt.subplot(2, 3, 1)        # 2 行 3 列的第 1 个位置
    plt.imshow(original)
    plt.title('Original Image')
    plt.axis('off')

    plt.subplot(2, 3, 2)        # 第 2 个位置
    plt.imshow(noisy)
    plt.title('Noisy Image\n(Salt-Pepper + Impulse + Gaussian)')
    plt.axis('off')

    # 第二行显示三种滤波结果
    plt.subplot(2, 3, 4)        # 第 4 个位置（第二行第一列）
    plt.imshow(mean_filtered)
    plt.title('Mean Filter')
    plt.axis('off')

    plt.subplot(2, 3, 5)        # 第 5 个位置
    plt.imshow(gaussian_filtered)
    plt.title('Gaussian Filter')
    plt.axis('off')

    plt.subplot(2, 3, 6)        # 第 6 个位置
    plt.imshow(median_filtered)
    plt.title('Median Filter')
```

```
plt.axis('off')

plt.tight_layout()

# 步骤 4: 保存结果
cv2.imwrite('noisy.jpg', cv2.cvtColor(noisy, cv2.COLOR_RGB2BGR))
cv2.imwrite('mean.jpg', cv2.cvtColor(mean_filtered,
cv2.COLOR_RGB2BGR))
    cv2.imwrite('gaussian.jpg', cv2.cvtColor(gaussian_filtered,
cv2.COLOR_RGB2BGR))
    cv2.imwrite('median.jpg', cv2.cvtColor(median_filtered,
cv2.COLOR_RGB2BGR))

plt.savefig('filter.jpg') # 保存对比图
plt.show()
```

4. 对比分析，输出结果

最后输出的对比分析结果为下图，可以看到在本图的噪声情况下，中值滤波器除噪的效果最好



