**CV实验二**

1. 添加噪声
2. 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

1. 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

1. 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)

1. 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

1. 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()

1. 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

图形用户界面, 网站, 日程表

AI 生成的内容可能不正确。