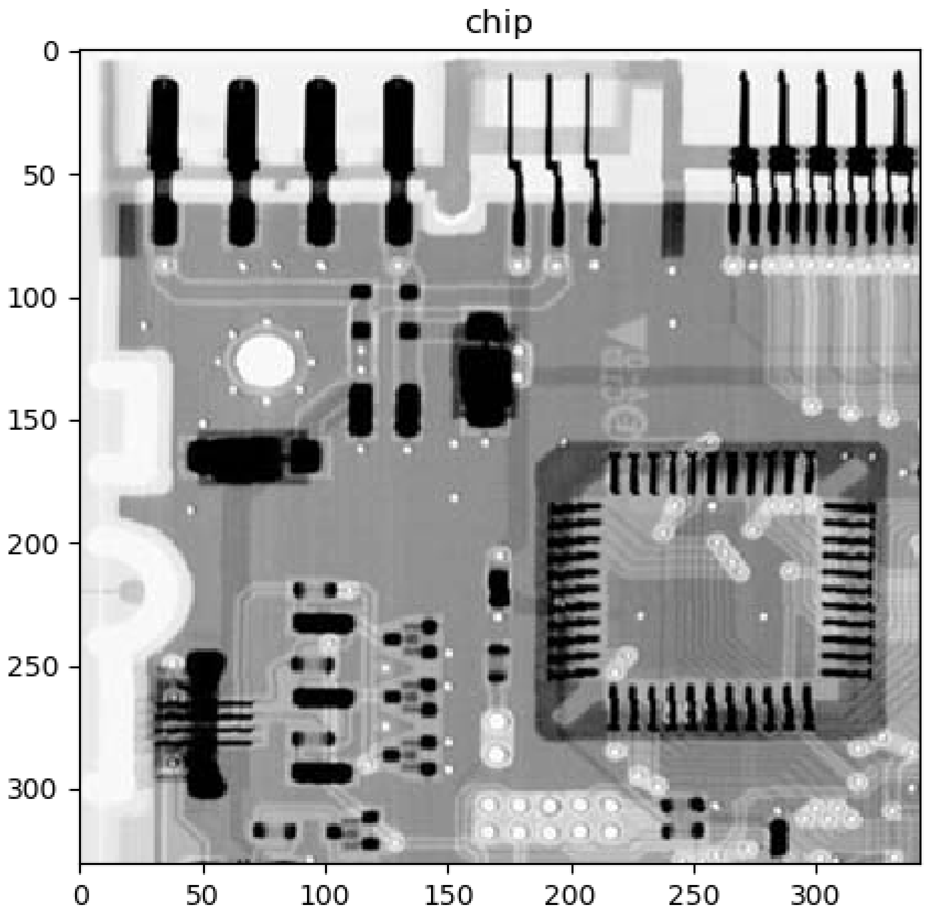
数字图像处理作业报告五

学号：71194506019 姓名：姜志刚 专业：计算机技术

题目

对一副图像加噪，进行几何均值，算术均值，谐波，逆谐波处理

待处理图像：

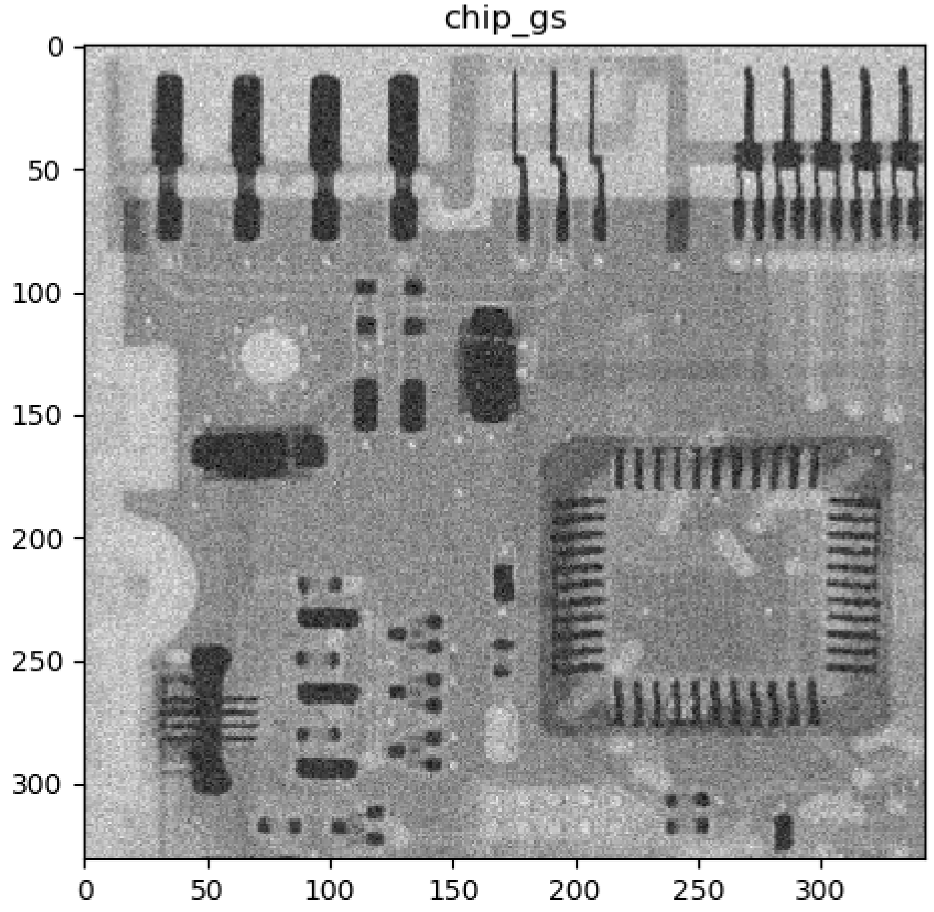


加噪声函数：

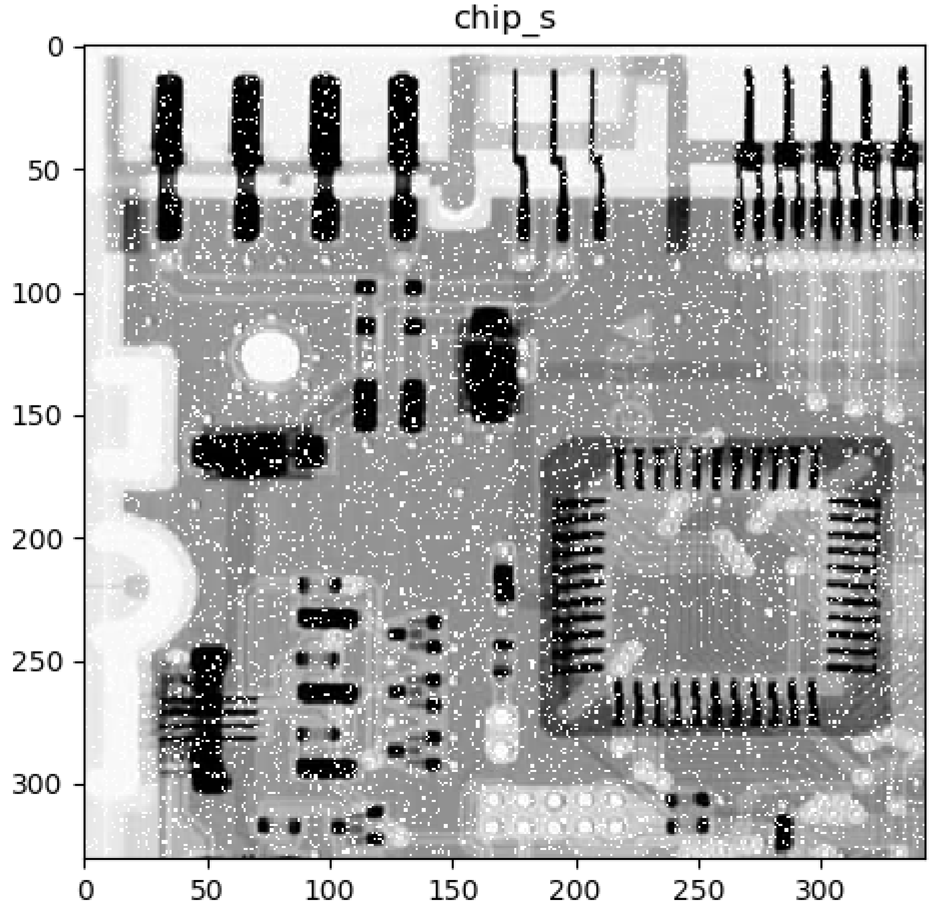
```
def add_gaussian_noise(image_in, noise_sigma=25):
    temp_image = np.float64(np.copy(image_in))
    h = temp_image.shape[0]
    w = temp_image.shape[1]
    noise = np.random.randn(h, w) * noise_sigma

    noisy_image = np.zeros(temp_image.shape, np.float64)
    if len(temp_image.shape) == 2:
        noisy_image = temp_image + noise
    else:
        noisy_image[:, :, 0] = temp_image[:, :, 0] + noise
        noisy_image[:, :, 1] = temp_image[:, :, 1] + noise
        noisy_image[:, :, 2] = temp_image[:, :, 2] + noise
    """
    print('min,max = ', np.min(noisy_image), np.max(noisy_image))
    print('type = ', type(noisy_image[0][0][0]))
    """
    return noisy_image

def sp_noisy(image, s_vs_p=0.5, amount=0.08):
    out = np.copy(image)
    num_salt = np.ceil(amount * image.size * s_vs_p)
    coords = [np.random.randint(0, i - 1, int(num_salt)) for i in image.shape]
    out[tuple(coords)] = 255
    num_pepper = np.ceil(amount * image.size * (1. - s_vs_p))
    coords = [np.random.randint(0, i - 1, int(num_pepper)) for i in image.shape]
    out[tuple(coords)] = 0
    return out
```

噪声图

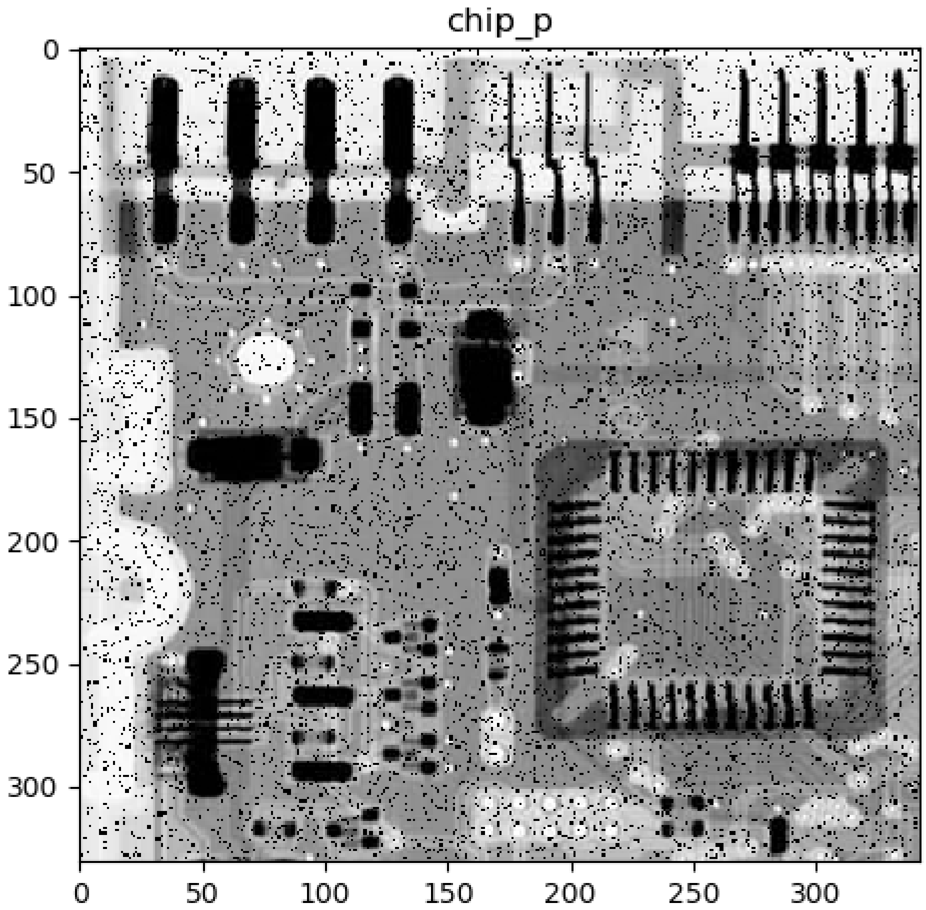
高斯噪声：


chip_gs
0
50
100
150
200
250
300
0
50
100
150
200
250
300

盐噪声：


chip_s
0
50
100
150
200
250
300
0
50
100
150
200
250
300

胡椒噪声：



滤波函数

```
def filter(image, op):
    new_image = np.zeros(image.shape)
    image = cv2.copyMakeBorder(image, 1, 1, 1, 1, cv2.BORDER_DEFAULT)
    for i in range(1, image.shape[0] - 1):
        for j in range(1, image.shape[1] - 1):
            new_image[i - 1, j - 1] = op(image[i - 1:i + 2, j - 1:j + 2])
    new_image = (new_image - np.min(image)) * (255 / np.max(image))
    return new_image.astype(np.uint8)
```

几何均值滤波器

几何均值可由如下模板进行卷积求得，第三次作业已求过

```
k1 = np.array([
    [1, 1, 1],
    [1, 1, 1],
    [1, 1, 1]
], np.float32)/9
```

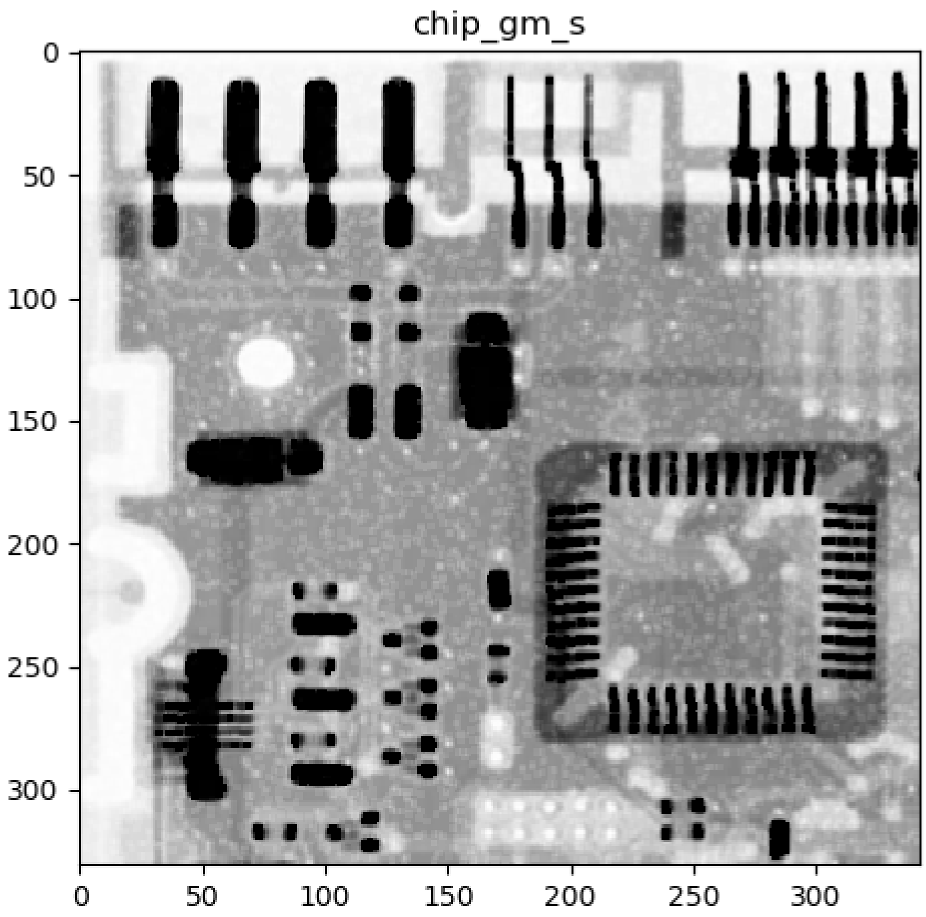
算术均值滤波器

操作函数:

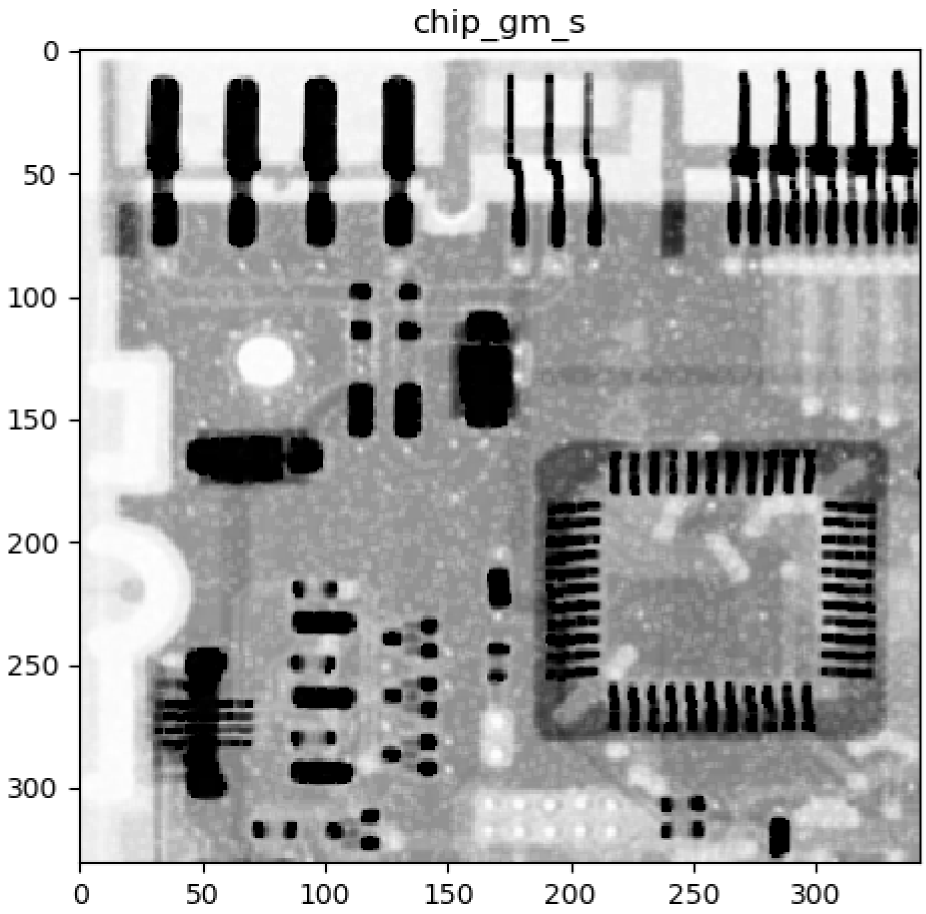
```
def GeometricMeanOperator(roi):
    roi = roi.astype(np.float64)
    p = np.prod(roi)
    re = p ** (1 / (roi.shape[0] * roi.shape[1]))
    if re < 0:
        re = 0
    if re > 255:
        re = 255
    return re
```

效果

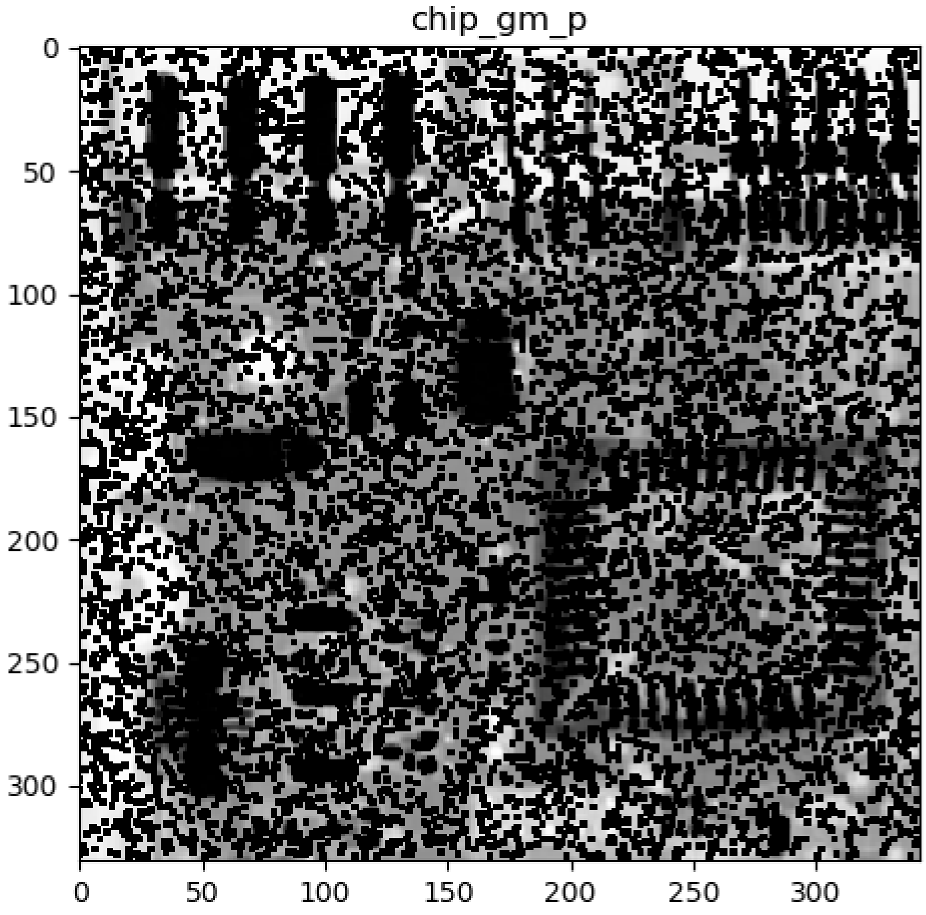
高斯:



盐噪声:



胡椒噪声



几何均值不适用于胡椒噪声

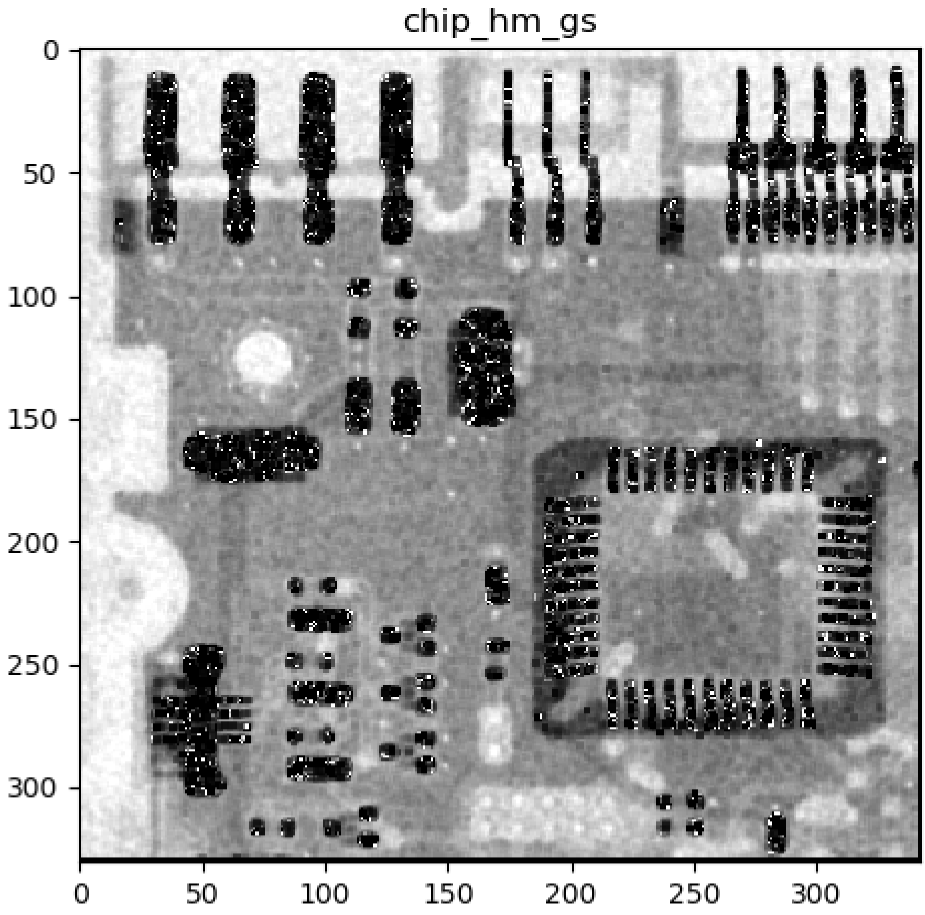
谐波均值滤波器

操作函数：

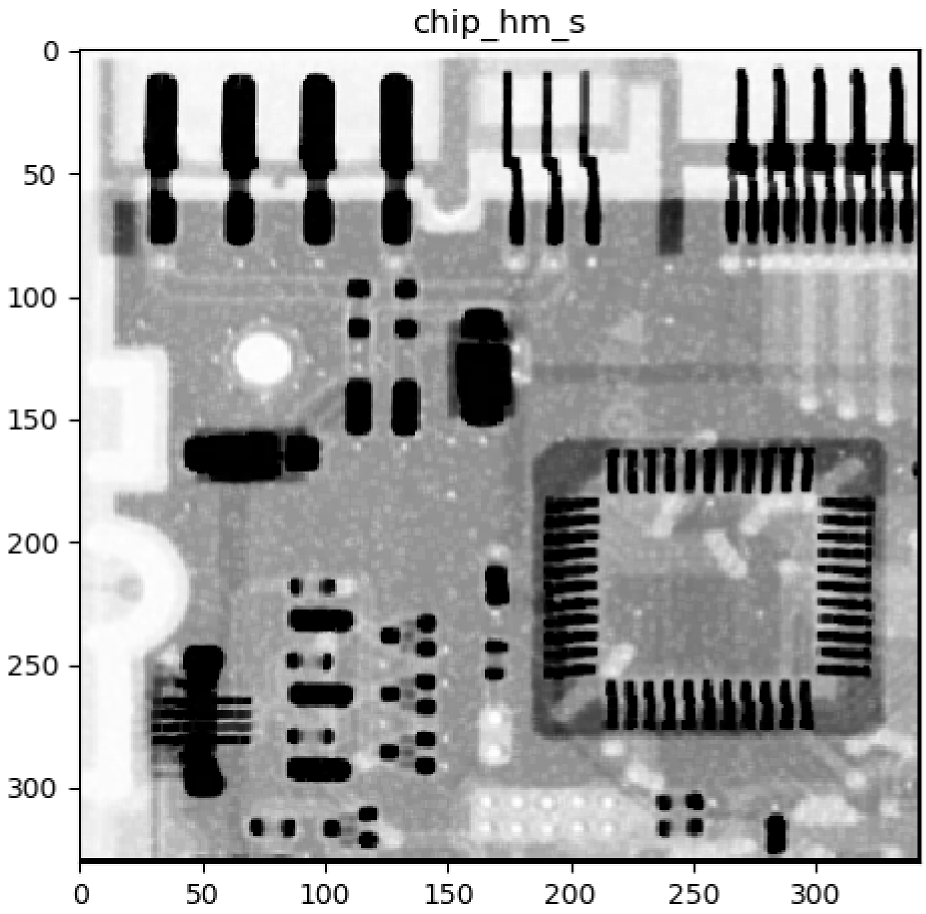
```
def HMeanOperator(roi):
    roi = roi.astype(np.float64)
    re = roi.shape[0] * roi.shape[1] / np.sum([1/(p+0.0001) for p in roi])
    if re < 0:
        re = 0
    if re > 255:
        re = 255
    return re
```

效果

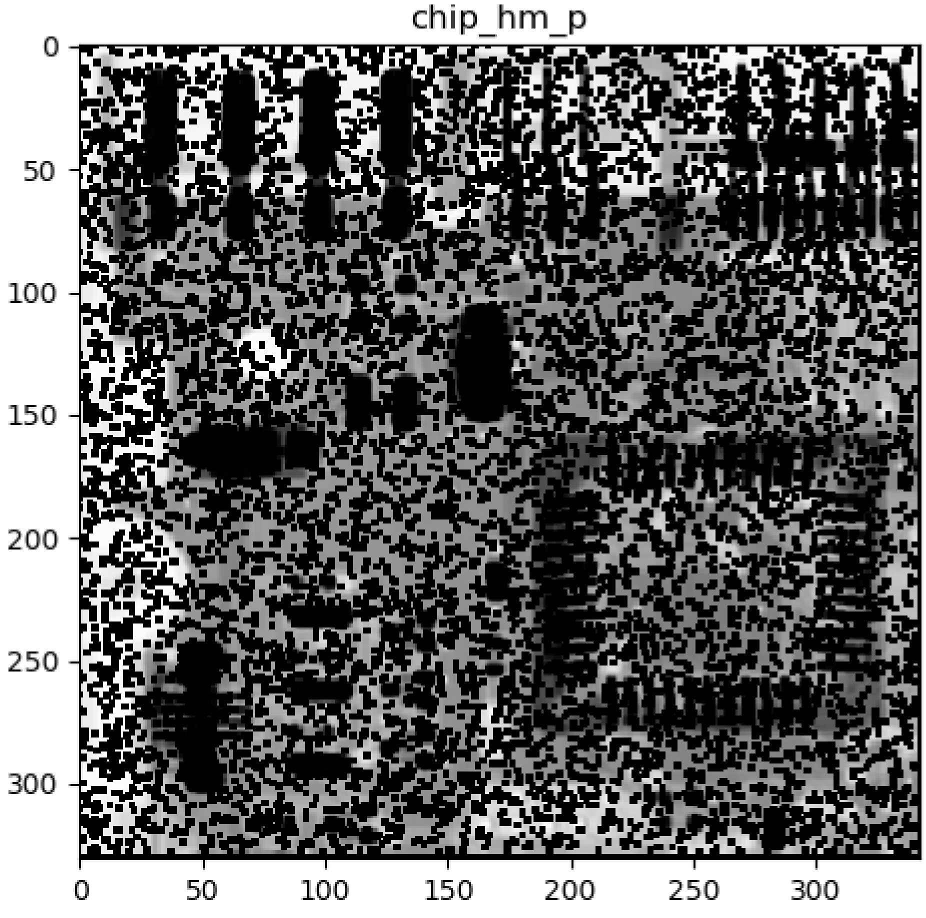
高斯：



盐噪声：



胡椒噪声：



谐波均值也不适用于胡椒噪声

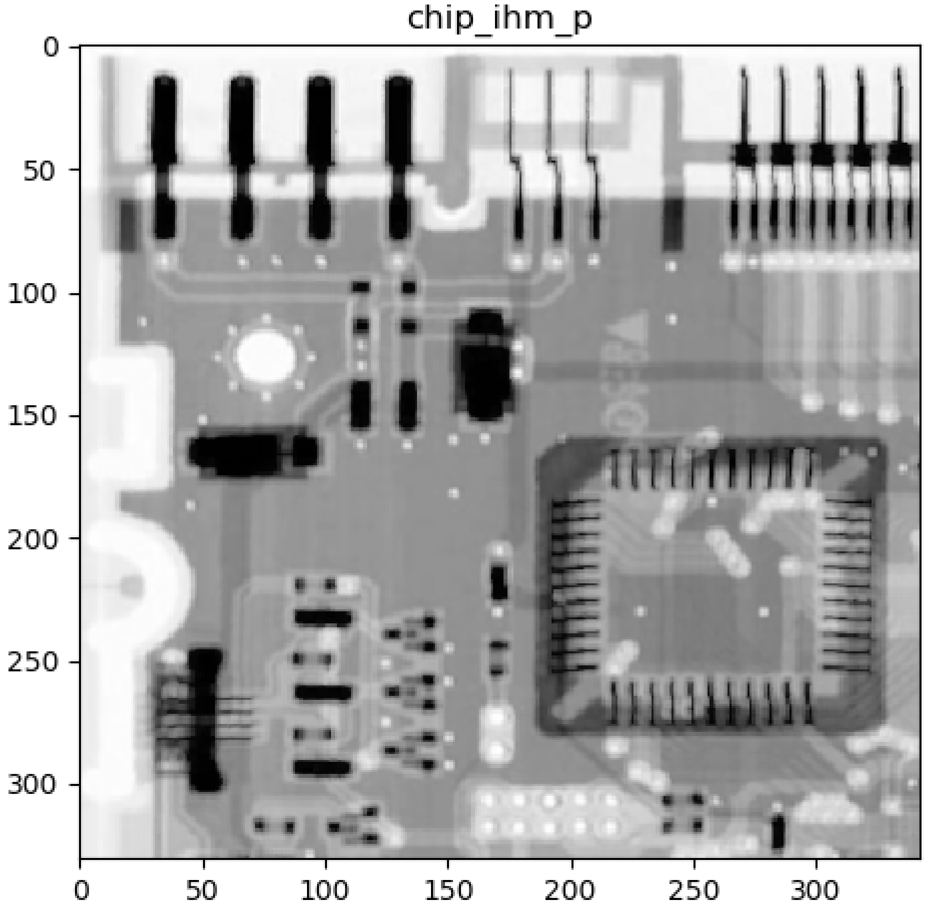
逆谐波均值滤波器

操作函数:

```
def IHMeanOperator(roi, q):
    roi = roi.astype(np.float64)
    return np.mean(roi ** (q + 1)) / np.mean(roi ** q)


def IHMeanAlogrithm(image, q):
    new_image = np.zeros(image.shape)
    image = cv2.copyMakeBorder(image, 1, 1, 1, 1, cv2.BORDER_DEFAULT)
    for i in range(1, image.shape[0] - 1):
        for j in range(1, image.shape[1] - 1):
            new_image[i - 1, j - 1] = IHMeanOperator(image[i - 1:i + 2, j - 1:j
    new_image = (new_image - np.min(image)) * (255 / np.max(image))
    return new_image.astype(np.uint8)
```

效果

q>0时消除胡椒噪声：



q<0时消除盐噪声：