

# Filters

## زهرا نیازی

اطلاعات گزارش	چکیده
تاریخ: 09/09/1401	مفاهیم معرفی شده در این تمرین شامل فیلتر کردن با ماسک های spatial است که از جمله اساسی ترین ابزارهای مورد استفاده در پردازش تصویر می باشند. این ابزارها در زمینه بهبود تصویر معرفی شدند و درک نحوه عملکرد آنها بر روی تصاویر دیجیتال به پردازش تصاویر کمک می کند.
واژگان کلیدی:	
مقاله	
شیوه نامه تدوین	
نویسنده	
چاپ	
شکل	
جدول	
فرمول	
نتایج	

### 1-مقدمه

به فیلتر، مجموع حاصلضرب ضرایب کرنل و پیکسل هایی از تصویر هستند که توسط کرنل احاطه شده اند.

### 2-شرح تکنیکال

#### Box Filter 1-2

فیلترهای فضایی هموارساز (میانگین گیر) برای کاهش تفاوت های شدید در شدت روشنایی تصویر استفاده می شود. از آنجایی که نویز تصادفی معمولاً شامل تغییرات شدید در شدت روشنایی است، یک کاربرد فیلتر هموارساز کاهش نویز است. فیلتر هموارساز برای کاهش جزئیات نامربوط در یک تصویر استفاده می شود، جایی که "نامرتب" به مناطق پیکسلی اشاره دارد که نسبت به اندازه هسته فیلتر کوچک هستند. کاربرد دیگر برای صاف کردن خطوط کاذب ناشی از استفاده از تعداد ناکافی سطوح شدت در یک تصویر است.

فیلتر spatial یک تصویر را با جایگزین کردن مقدار هر پیکسل با تابعی از مقادیر همان پیکسل و همسایگان آن تغییر می دهد. اگر عملیات انجام شده بر روی پیکسل های تصویر خطی باشد، فیلتر را فیلتر خطی می نامند. یک فیلتر خطی یک عملیات مجموع حاصلضربها را بین یک تصویر  $f$  و یک فیلتر کرنل  $w$  انجام می دهد. کرنل آرایه ای است که اندازه آن بازه عملیات را مشخص می کند و ضرایب آن ماهیت فیلتر را تعیین می کند. سایر عبارات مورد استفاده برای اشاره به فیلتر خطی spatial ماسک، قالب و پنجره هستند. در هر نقطه  $(x, y)$  در تصویر، پاسخ  $g(x, y)$

استفاده عمده از فیلترهای میانگین در کاهش جزئیات بی ربط در یک تصویر است. منظور از بی ربط مناطق پیکسلی است که نسبت به اندازه ماسک کوچک هستند.

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{4.8976} \times \begin{bmatrix} 0.3679 & 0.6065 & 0.3679 \\ 0.6065 & 1.0000 & 0.6065 \\ 0.3679 & 0.6065 & 0.3679 \end{bmatrix}$$

شکل 2-1-2 box filter and gaussian filter

شکل 2-1-2 دو فیلتر هموارساز  $3 \times 3$  را نشان می دهد. استفاده از فیلتر اول میانگین پیکسل های زیر ماسک را به عنوان خروجی می دهد. این را می توان با در نظر گرفتن ضرایب 1 به ازای هر پیکسل در فرمول 1 نظر گرفت:

$$R = \frac{1}{9} \sum_{i=1}^9 z_i \quad (2)$$

که میانگین سطوح خاکستری پیکسل ها در همسایگی  $3 \times 3$  تعریف شده توسط ماسک است. باید توجه کرد که ضرایب فیلتر همه 1 هستند. در پایان فرآیند فیلتر کردن، کل تصویر بر 9 تقسیم می شود. یک ماسک  $m \times n$  دارای یک ضریب نرمال کننده برابر با  $1/m \times n$  خواهد بود.

فیلتر میانگین گیر مکانی که در آن همه ضرایب برابر با 1 هستند Box Filter نامیده می شود.

فیلتر دوم نشان داده شده در شکل 2-1-2 میانگین وزن دار را به دست می آورد، که در واقع پیکسل ها در ضرایب مختلف ضرب می شوند. بنابراین اهمیت (وزن) بیشتری به برخی از پیکسل ها در مقابل بقیه پیکسل ها می دهد. در فیلتر مذکور، پیکسل در مرکز فیلتر در مقداری بالاتر از هر پیکسل دیگری ضرب شده است، بنابراین به این پیکسل اهمیت بیشتری در محاسبه میانگین داده می شود. سایر پیکسل ها بر حسب فاصله آنها از مرکز ماسک وزن معکوس دارند. استراتژی دادن بیشترین وزن به نقطه مرکزی و سپس کاهش مقدار ضرایب متناسب با افزایش فاصله از مرکز، به این دلیل است که بتوان تاری را در فرآیند هموارسازی کاهش داد.

## 1-1-2

به طور کلی نقاط ضعف این فیلتر ها شامل موارد زیر میشود:

- مشکل اصلی فیلترهای میانگین گیر این است که باعث میشود لبه های تصویر تضعیف شوند و کیفیت

$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

شکل 2-1-1 ماسک  $3 \times 3$

به عنوان مثال اگر فیلتر  $3 \times 3$  به شکل بالا داشته باشیم که  $w$  وزن های این فیلتر باشد،  $Z$  مقادیر شدت روشنایی ناحیه  $3 \times 3$  از تصویر است که توسط کرنل احاطه شده است و توسط فرمول زیر محاسبه می شود:

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9 = \sum_{k=1}^9 w_k z_k = w^T z \quad (1)$$

خروجی یک فیلتر هموارساز به سادگی میانگین پیکسل های موجود در همسایگی ماسک است. این فیلترها فیلترهای میانگین گیر نیز نامیده می شوند.

ایده فیلترهای هموارساز ساده است. با جایگزینی مقدار هر پیکسل در یک تصویر با میانگین سطوح خاکستری در همسایگی آن پیکسل که توسط ماسک فیلتر تعریف شده اند. این فرآیند منجر به تصویری با کاهش تفاوت های ناگهانی در سطوح خاکستری آن می شود. از آنجایی که نویز تصادفی معمولاً از تفاوت ناگهانی و زیاد در سطوح خاکستری به وجود می آید، واضح ترین کاربرد فیلترهای هموارساز کاهش نویز است. با این حال، لبه ها (که تقریباً همیشه ویژگی های مطلوب یک تصویر هستند و وجودشان باعث افزایش کیفیت تصویر می شود) نیز با تفاوت های واضح در سطوح خاکستری مشخص می شوند. بنابراین فیلترهای میانگین گیر، عوارض جانبی نامطلوبی دارند که لبه ها را محو می کنند.

یکی دیگر از کاربردهای این فیلترها صاف کردن خطوط نادرست است که به دلیل عدم استفاده از تعداد کافی سطوح خاکستری ایجاد می شود.

تصویر پایین بیاید. در نتیجه هر چه سایز فیلتر ها را بزرگ تر کنیم لبه ها بیشتر suppress میشوند. به خاطر همین مشکل که تاکید میشود که بر تصویر خروجی باید حتما یک فیلتر لبه یاب اعمال شود تا لبه ها تقویت شوند.

- مشکل دیگری که این فیلتر دارد این است که مقدار غیرطبیعی یک پیکسل که خیلی با پیکسل های اطرافش متفاوت است تاثیر زیادی بر پیکسل های همسایه خود میگذارد به این شکل که فرضا اگر 3 پیکسل با مقدار های 10 و 254 و 10 داشته باشیم میانگین این مقادیر برابر 91 میشود که همینطور که مشخص است این مقدار تاثیر زیادی بر روی مقادیر اطراف گذاشته است.
- نقطه ضعف دیگری که این فیلتر ها دارند این است که برای تصاویر که نویز نمک و فلفل و نویزهای مشابه روی آنها اعمال شده است، نتیجه مناسبی را نمیدهد.
- نکته اخر هم این است که این فیلتر در تصویر خروجی خاصیت بلاک بلاکی ایجاد میکند و لبه های بین پیکسل ها که از آن منتج میشوند شارپ میباشند.

## 2-1-2

## 3-1-2

با چند بار اعمال فیلتر میانگین گیر به روی تصویر، مشاهده میکنیم که مقدار سطوح خاکستری پیکسل ها به طور کلی بیشتر به سمت عدد 128 رفته است. در واقع با اعمال فیلتر میانگین گیر به مراتب زیاد میتواند منجر به این شود که تصویر خروجی کاملاً خاکستری شود.

## 2-1-4 تاثیر سایز فیلتر بر روی تار کردن و حذف

### نویز

اثرات هموارسازی متناسب با اندازه فیلتر در این تمرین بررسی می شود. در این تمرین تصویر elaine با استفاده از فیلترهای میانگین گیر مربعی با اندازه های  $n = 3, 5, 9, 15, 35$  پیکسل به ترتیب هموارسازی شده است. در بخش نتایج به تفصیل درباره تاثیر سایز فیلتر روی میزان تاری و حذف نویز تصویر توضیح داده شده است.

به طور خلاصه همانطور که قبلاً هم ذکر شد، یک کاربرد مهم فیلترهای میانگین گیر، تار کردن یک تصویر به منظور رسیدن به نمایی کلی از اشیاء مورد نظر درون تصویر است. به طوری که شدت روشنایی اشیاء کوچکتر با پس زمینه ترکیب می شوند و تشخیص اجسام بزرگتر آسان می شود. *اندازه فیلتر، اندازه نسبی اشیایی را که با پس زمینه ترکیب می شوند را تعیین می کند.*

فیلترهایی با اندازه کوچک برای رفع نویز های درون تصویر مناسب می باشند. از آنجایی که اگر اندازه فیلتر را حدوداً اندازه نویز درون تصویر بگیریم، آنگاه برای محاسبه حاصل کانولوشن این ماسک و تصویر اصلی، پیکسل های اطراف نویز نیز در میانگین گیری اثر خواهند داشت و باعث حذف شدن نویز خواهد شد.

اما اگر اندازه فیلتر را بزرگ بگیریم، آنگاه تصویر خروجی تارتر میشود. و می بینیم که تعدادی از اشیاء یا با پس زمینه ترکیب شده اند یا از شدت آنها به میزان قابل توجهی کاسته شده است. زیرا پیکسل های بیشتر و دورتری در میانگین گیری برای یک پیکسل خاص اثر داشته اند. در این نوع فیلتر که تصویر خروجی آن تارتر است، میتوان در کاربرد های segmentation استفاده کرد. زیرا به دلیل تاری زیاد تصویر خروجی، مرزهای اشیاء اصلی تصویر مشخص میشوند که برای تمیز دادن اشیاء کاربردی می باشد.

## 2-1-6 اعمال فیلتر لاپلاسیان

## Median Filter 2-2

فیلتر میانه، از فیلترهای فضایی غیرخطی می باشد که پاسخ آن بر اساس مرتب کردن پیکسل های موجود در ناحیه تصویری است که ماسک در بر می گیرد. و سپس مقدار پیکسل مرکزی را با مقدار تعیین شده توسط نتیجه رتبه بندی جایگزین می کند. همانطور که از نام فیلتر میانه پیداست، مقدار یک پیکسل را با میانه سطوح خاکستری در همسایگی آن پیکسل جایگزین می کند. این در حالیست که مقدار اولیه پیکسل در انتخاب میانه در نظر گرفته شده است.

فیلترهای میانه بسیار کاربردی هستند زیرا، برای انواع خاصی از نویزهای رندوم، قابلیت کاهش نویز بسیار خوبی را ارائه می دهند، و تاری کمتری نسبت به فیلترهای هموارساز خطی با

اندازه مشابه در نتیجه نشان میدهند. فیلترهای میانه به ویژه در حضور نویز نمک و فلف موثر می باشد.

میانه از مجموعه ای از مقادیر به گونه ای انتخاب میشود که نیمی از مقادیر مجموعه کمتر یا مساوی آن و نیمی بزرگتر یا مساوی آن هستند. برای اعمال فیلتر میانه در یک نقطه از یک تصویر، ابتدا مقادیر پیکسل مورد نظر و همسایگان آن را مرتب سازی می کنیم. سپس میانه آنها را تعیین کرده و این مقدار را به پیکسل مورد نظر اختصاص میدهم.

وقتی چندین مقدار در یک محله یکسان هستند، همه مقادیر مساوی گروه بندی می شوند. برای مثال، فرض کنید که یک ناحیه  $3 \times 3$  از تصویر دارای مقادیر 10, 20, 20, 20, 15, 20, 20, 25, 100 است. این مقادیر به صورت 10, 25, 100, 20, 20, 20, 20, 20, 25 مرتب می شوند که میانه آنها عدد 20 می شود.

بنابراین، عملکرد اصلی فیلترهای میانه به این شکل است که نقاطی با سطوح خاکستری مشخص را وادار کنند تا بیشتر شبیه همسایگان خود شوند. در واقع، گروه هایی از پیکسل ها که نسبت به همسایه های شان روشنتر یا تاریکتر هستند و مساحت آن ها کمتر از  $n^2/2$  (نصف سایز فیلتر) است، توسط فیلتر میانه  $n \times n$  حذف می شوند. در واقع حذف به معنای این است که مجبور به تغییر شدت روشنایی خود به شدت روشنایی متوسط همسایگان می شود. گروه های بزرگتر به میزان قابل توجهی کمتر تحت تأثیر این پدیده قرار می گیرند.

### salt&pepper noise 1-2-2

نویز impulse یا نمک و فلف وقوع پراکنده حداکثر 255 و حداقل 0 مقادیر برای یک پیکسل در تصویر است. این را می توان به عنوان وجود پیکسل های سیاه در مناطق روشن و پیکسل های سفید در مناطق تیره مشاهده کرد. این نوع نویز به دلیل اختلالات شدید و ناگهانی در مقادیر سطوح خاکستری تصویر ایجاد می شود. این نویز عمدتاً در اثر خطا در تبدیل آنالوگ به دیجیتال یا انتقال بیت در مانیتورهای قدیمی ایجاد می شود.

برای ایجاد نویز نمک و فلف، ابتدا یک توزیع یکنواخت ایجاد می کنیم که کران های پایین و بالایی آن حداقل و حداکثر مقادیر پیکسل (به ترتیب 0 و 255) در طول ابعاد تصویر هستند. سپس آستانه باینری را اعمال میکنیم تا جدولی از

پیکسل های سیاه و سفید ایجاد شود. شدت نویز را می توان به راحتی با تغییر مقدار آستانه تغییر داد.

با توجه به مقادیری که در جدول 1-2-3 مشاهده میکنیم، هنگامی که واریانس نویز تصویر برابر با 0.05 باشد، فیلتری با اندازه  $3 \times 3$  برای حذف این نویز موثر خواهد بود. این قضیه برای نویز با واریانس 0.1 نیز صدق میکند. اما هنگامی که واریانس نویز تصویر برابر با 0.2 باشد فیلتری با اندازه  $5 \times 5$  و هنگامی که واریانس نویز برابر با 0.5 باشد فیلتری با اندازه  $7 \times 7$  برای حذف این نویز مناسب می باشد.

### gaussian noise 2-2-2

با بررسی جدول 2-2-3 می یابیم که در تمام واریانس های مختلف برای نویزدار کردن تصویر، فیلتر میانه نسبت به فیلتر میانگین گیر عملکرد بهتری داشته است. هنگامی که واریانس نویز تصویر برابر با 0.1 باشد فیلتری با اندازه  $5 \times 5$  برای حذف این نویز مناسب می باشد. در بقیه حالات کوچک بودن اندازه فیلتر به حذف نویز کمک شایانی میکند.

## Sharpening, Blurring, 3-2 and Noise Removal

هدف اصلی شارپنینگ برجسته کردن جزئیات در یک تصویر و یا تقویت جزئیاتی است که تار شده است. موارد استفاده از شارپنینگ تصویر بسیار متنوع است و شامل کاربردهای مختلف از چاپ الکترونیکی و تصویربرداری پزشکی گرفته تا بازرسی صنعتی و هدایت مستقل در سیستم های نظامی می شود.

در بخش 2-2، دیدیم که تار کردن تصویر را می توان در حوزه spatial با میانگین گیری پیکسلها در یک همسایگی انجام داد. از آنجایی که میانگین گیری مشابه انتگرال گرفتن است، میتوان نتیجه گیری کرد که شارپنینگ می تواند با مشتق گرفتن انجام شود. در واقع قدرت پاسخ یک عملگر مشتق با درجه ناپیوستگی تصویر در نقطه ای که عملگر در آن اعمال می شود، متناسب است. بنابراین، مشتق گرفتن از تصویر لبه ها و سایر ناپیوستگی ها (مانند نویز) را افزایش می دهد و نواحی که مقادیر سطح خاکستری به آرامی تغییر می کنند را کم اثرتر میکند.

0	1	0
1	-4	1
0	1	0

شکل 1-3-2 Laplacian Kernel

از آنجایی که لاپلاسیان یک عملگر مشتق است، استفاده از آن ناپیوستگی‌های سطوح خاکستری در تصویر را برجسته می‌کند و بر نواحی که سطوح خاکستری به آرامی تغییر میکند تأکید نمی‌کند. این باعث ایجاد تصاویری با خطوط لبه مایل به خاکستری و سایر ناپیوستگی‌ها می‌شود که همگی روی پس‌زمینه‌ای تاریک قرار گرفته‌اند.

## Edge Detection 4-2

### 1-4-2

در این تمرین از ما خواسته شده است که سه تا از فیلترهای لبه یاب را به تصویر ورودی اعمال کنیم. فیلترهای لبه یاب معرفی شده در این سوال فیلترهایی هستند که اختلاف افقی را محاسبه میکنند در نتیجه لبه های عمودی را در تصویر خروجی ایجاد میکنند. برای اعمال این فیلترها از تابعی که از پیش تعریف شده استفاده میکنیم.

### 2-4-2

در این سوال از ما خواسته شده است که فیلتر لبه یاب روبرت را بر روی تصویر خاکستری الین اعمال کنیم. دو فیلتر روبرت مشخص شده است که فیلتر a فیلتری است که لبه های 45 درجه را به عنوان خروجی میدهد و فیلتر b فیلتری است که لبه های 135 درجه را به عنوان خروجی میدهد. در نتیجه پس از ترکیب کردن نتایج مربوط به هر یک از این دو فیلتر، خروجی تصویری است که لبه ها را در هر دو جهت 45 درجه و 135 میدهد و نکته ای که وجود دارد این است که فیلتر روبرت باعث تقویت بعضی از نویزها نیز میشود که در

می توان نشان داد که ساده ترین عملگر ایزوتروپیک مشتق، لاپلاسیان است که برای یک تابع (تصویر)  $f(x, y)$  از دو متغیر، به صورت زیر تعریف می شود:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (3)$$

از آنجا که مشتقات هر مرتبه ای عملیات خطی هستند، لاپلاسیان یک عملگر خطی است.

برای اینکه برای پردازش تصویر دیجیتال مفید باشد، این معادله باید به صورت گسسته بیان شود. راه های مختلفی برای تعریف لاپلاسی دیجیتال با استفاده از همسایگی ها وجود دارد. با این حال، هر تعریفی که باشد، باید ویژگی‌های مشتق دوم را برآورده کند. با توجه به اینکه اکنون دو متغیر داریم، از نماد زیر برای مشتق مرتبه دوم جزئی در جهت X استفاده می کنیم:

$$\frac{\partial^2 f}{\partial^2 x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y) \quad (4)$$

و به طور مشابه در جهت Y:

$$\frac{\partial^2 f}{\partial^2 y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y) \quad (5)$$

اجرای دیجیتال لاپلاسیان دو بعدی با جمع این دو مؤلفه به دست می آید:

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y) \quad (6)$$

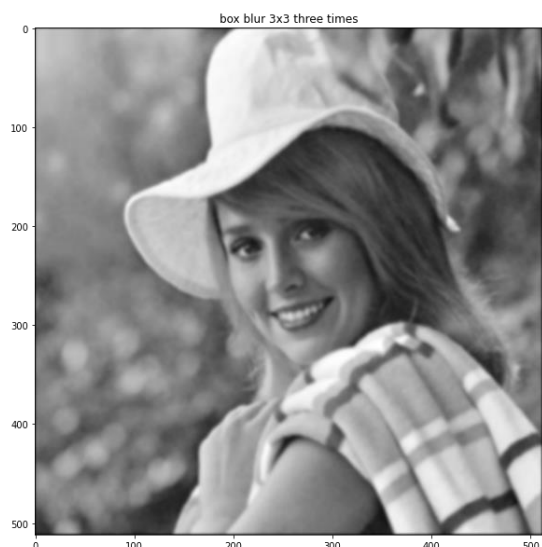
این معادله را می توان با استفاده از ماسک نشان داده شده در شکل 1-3-2 پیاده سازی کرد. مکانیک پیاده سازی در معادله آورده شده است.

خروجی هم مشخص است. در مقایسه با فیلتر های سوال قبلی نکته ای که وجود دارد این است که فیلتر روبرت به طور کلی نسبت به فیلتر سوبل نویز ها را بیشتر تقویت میکند.

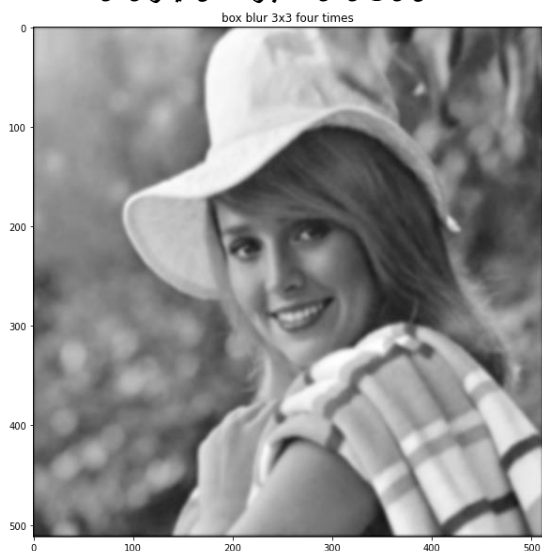
### 3-نتایج

#### Box Filter 1-3

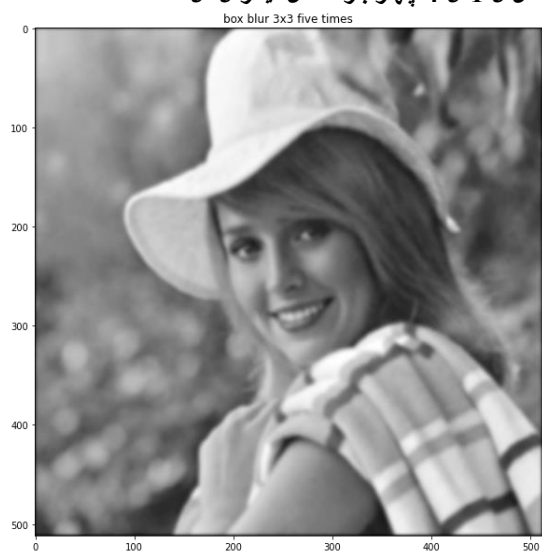
##### 3-1-3



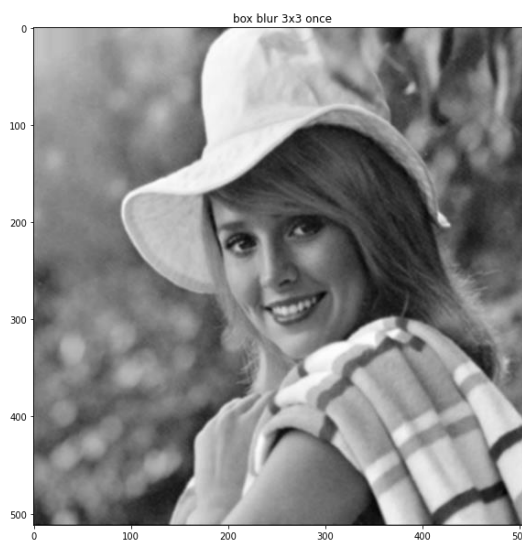
شکل 3-1-3 سه بار اعمال فیلتر  $3 \times 3$



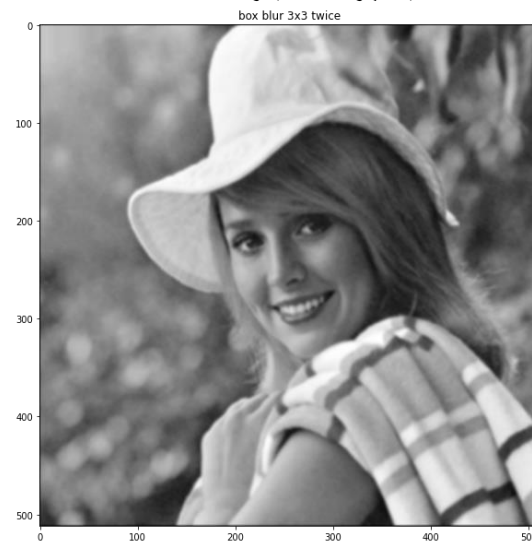
شکل 3-1-3 چهار بار اعمال فیلتر  $3 \times 3$



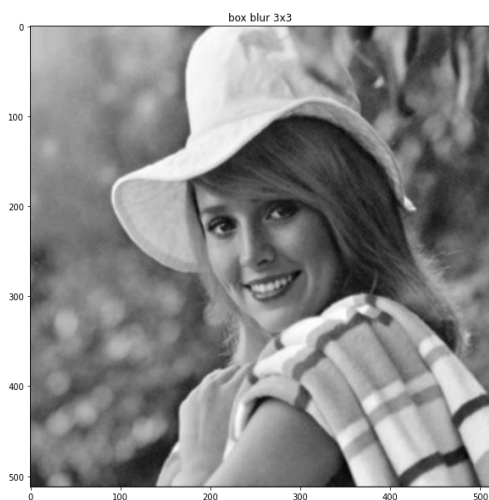
شکل 3-1-3 پنج بار اعمال فیلتر  $3 \times 3$



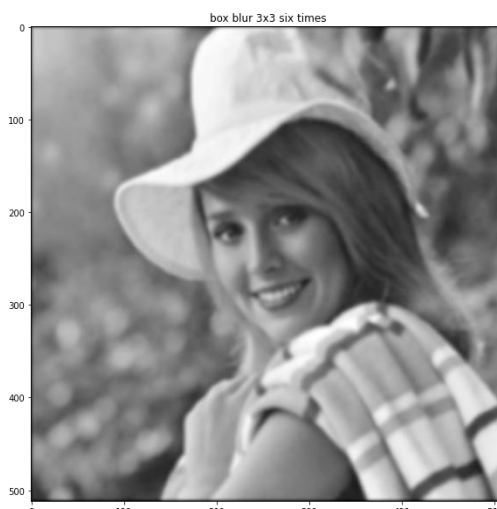
شکل 3-1-3 یک بار اعمال فیلتر  $3 \times 3$



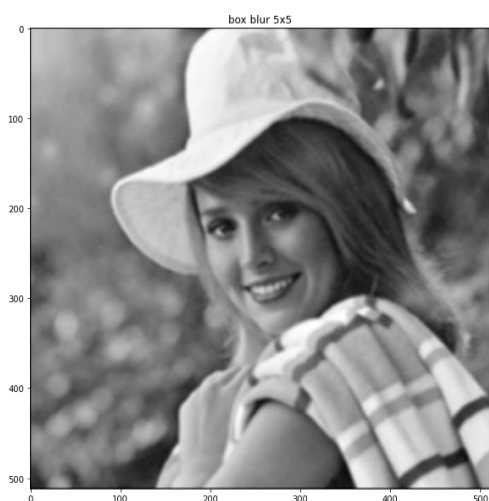
شکل 3-1-3 دو بار اعمال فیلتر  $3 \times 3$



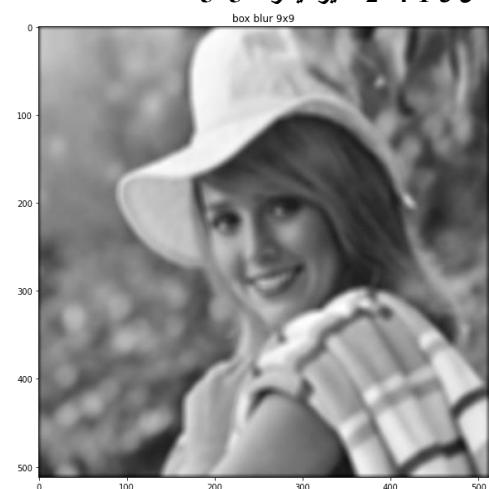
شکل 3-1-4-1 سایز فیلتر =  $3 \times 3$



شکل 3-1-3 شش بار اعمال فیلتر  $3 \times 3$



شکل 3-1-4-2 سایز فیلتر =  $5 \times 5$



شکل 3-1-4-3 سایز فیلتر =  $9 \times 9$

### 4-1-3 تاثیر سایز فیلتر روی تارکردن و حذف

#### نویز

ویژگی‌های اصلی این نتایج به شرح زیر است:

برای  $n=3$ ، ما یک تار شدن جزئی کلی در کل تصویر مشاهده می‌کنیم، اما همانطور که انتظار می‌رود، جزئیاتی که تقریباً به اندازه ماسک فیلتر هستند به طور قابل توجهی بیشتر تحت تأثیر قرار می‌گیرند. به عنوان مثال، بافت موها و جزئیات صورت در مقایسه با بقیه تصویر، تار قابل توجهی را نشان می‌دهند. یک نتیجه مثبت این است که نویز کمتر مشخص می‌شود. مورد توجه است که نقطه سیاهی که روی گونه قرار داشت حذف شده است و همچنین نویز یکنواختی که روی تصویر وجود داشت هم به طرز دلپذیری صاف شده است.

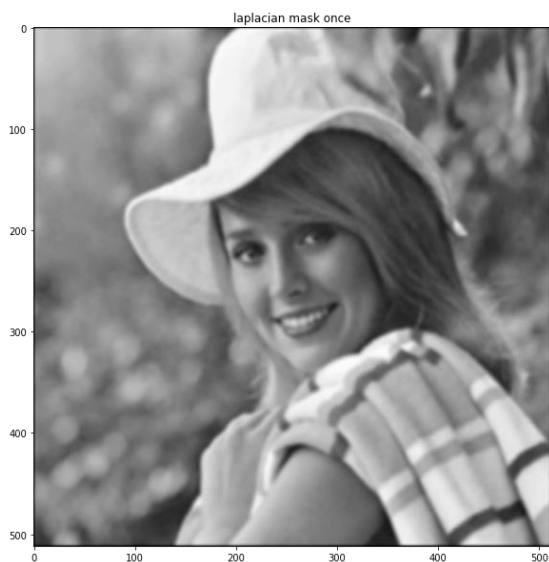
نتیجه برای  $n=5$  تا حدودی مشابه است، با افزایش جزئی بیشتری در تار. برای  $n=9$  ما تار شدن قابل توجهی بیشتری می‌بینیم، و بخش‌های مختلف دست و صورت از هم قابل تشخیص نیستند. این نشان دهنده تأثیری است که تار روی بخش‌هایی از تصویر که محتوای سطح خاکستری آنها نزدیک به سطح خاکستری پیکسل‌های همسایه آن می‌باشد، دارد.

نتیجه ای که از این بخش گرفته می شود این است که تا زمانی که اندازه فیلتر به اندازه نویزهای موجود در تصویر باشند آنگاه این نوع فیلترها نقش خوبی در حذف نویز ایفا میکنند. اما هرگاه سائز فیلتر از این مقدار بیشتر بشود و به اندازه اجزاء اصلی درون تصویر باشد، آنگاه کاری که این فیلتر انجام می دهد بیشتر از جنس segmentation خواهد بود. از آنجایی که در تسک image segmentation، دنبال کلیت تصویر هستیم و میخواهیم فقط مرزهای اصلی اشیاء حاضر در تصویر را پیدا کنیم، فیلتر میانگین گیر قوی فیلتر مناسبی می باشد. چون هنگامی که اندازه فیلتر بزرگتر باشد، پیکسل های بیشتر و دورتری در میانگین گیری برای یک پیکسل اثر میگذارند و در نتیجه، تصویر خروجی تارتر میشود.

### 5-1-3

اگر تسک مدنظر از نوع segmentation باشد آنگاه اندازه های بالا برای یافتن کلیت تصویر ثمربخش میباشد. اما اگر تسک حذف نویزهای موجود در تصویر باشد آنگاه بهتر است که فیلتر را در اندازه های کوچک در نظر بگیریم.

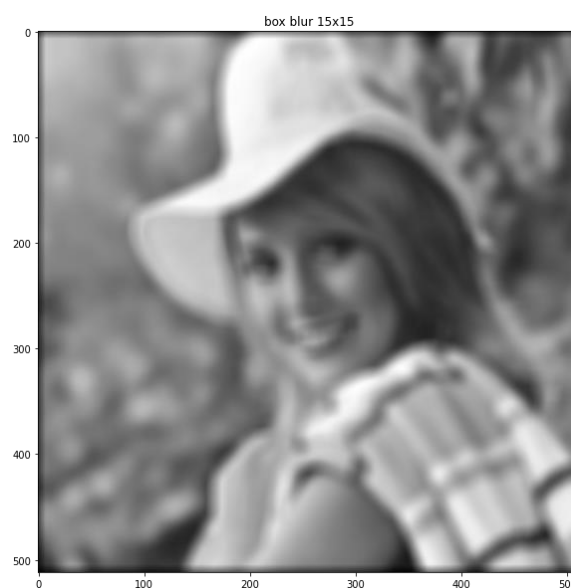
### 6-1-3 اعمال فیلتر لاپلاسی



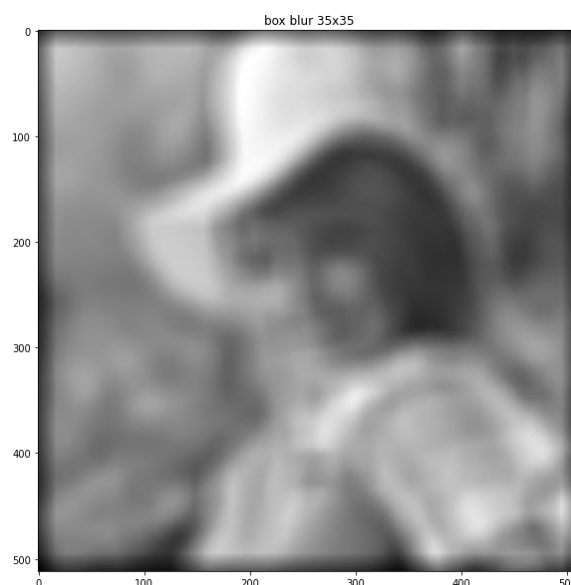
شکل 3-1-6 اعمال فیلتر لاپلاسی یک بار

نتایج برای  $n=15, 35$  نسبت به اندازه اشیاء در تصویر بسیار بزرگ انتخاب شده است. این نوع تارری بیش از حد به طور کلی برای حذف اشیاء کوچک از یک تصویر استفاده می شود. به عنوان مثال کل جزئیات تنه از سر و صورت جدا شده است و قابل تمیز می باشند و بیشتر مناطق پر سر و صدا در پس زمینه تصویر نیز ترکیب شده اند.

نکته قابل توجه دیگر در این شکل حاشیه سیاه می باشد. این نتیجه پر کردن حاشیه تصویر اصلی با 0 (سیاه) و سپس برش دادن ناحیه پد شده است. مقداری از رنگ سیاه در همه تصاویر فیلتر شده مشاهده می شود، اما برای تصاویر هموارسازی شده با فیلترهایی با سائز بزرگتر کاملاً فاحش است.



شکل 3-1-4 سائز فیلتر = 15x15



شکل 3-1-5 سائز فیلتر = 35x35

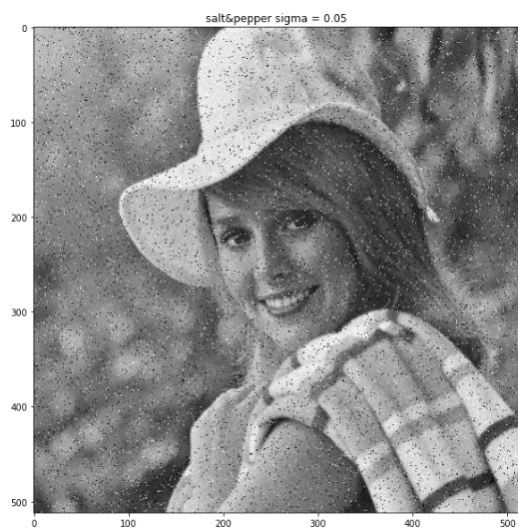


مشاهده میکنیم که به دلیل محو شدن اکثر لبه ها فیلتر لاپلاسین در تصویر 3-1-6-4 قادر به تشخیص لبه های اصلی نمی باشد.

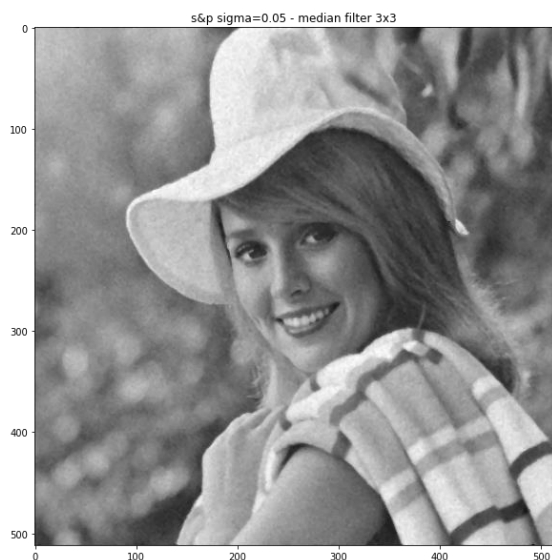
## Median Filter 2-3

### salt&pepper noise 1-2-3

در این تمرین باید تصویر Elaine را با چهار واریانس متفاوت نویز دار کنیم و سپس با فیلترهای میانه با سایزهای مختلف تلاش کنیم تا نویز ایجاد شده رفع شود.

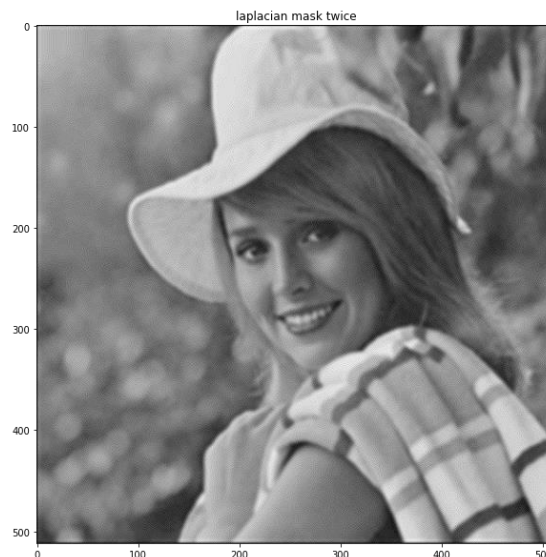


شکل 3-1-2-1 نویز نمک و فلفل  $\sigma = 0.05$

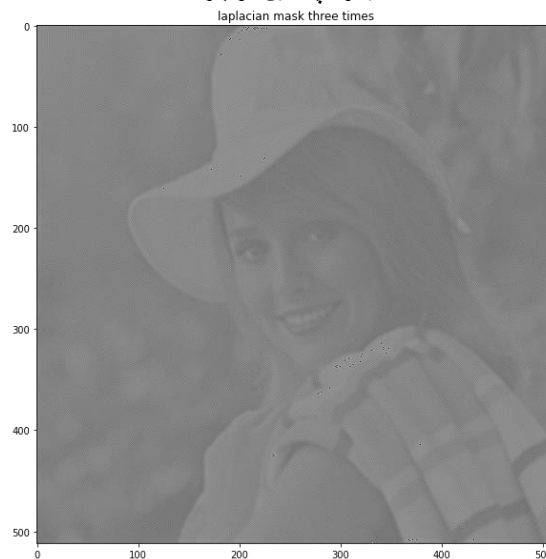


شکل 3-1-2-2  $\sigma=0.05$  - median filter 2-1-2-3 (3x3)

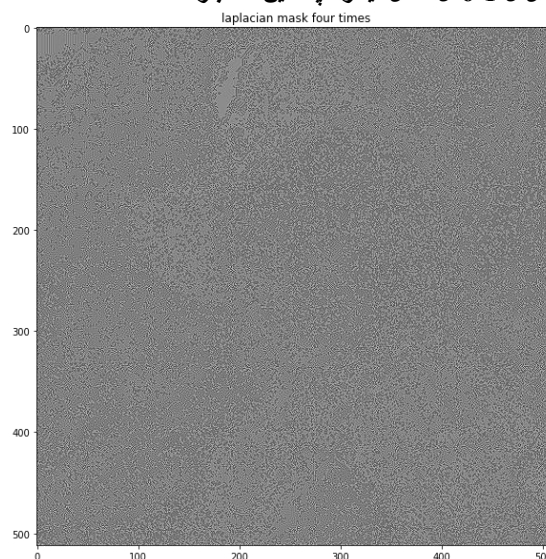
همانطور که مشاهده میکنیم نویز به طور کامل از بین رفته است. به طور کلی، فیلتر میانه برای حذف نویز نمک و فلفل بسیار مناسب تر از فیلتر میانگین گیر است.



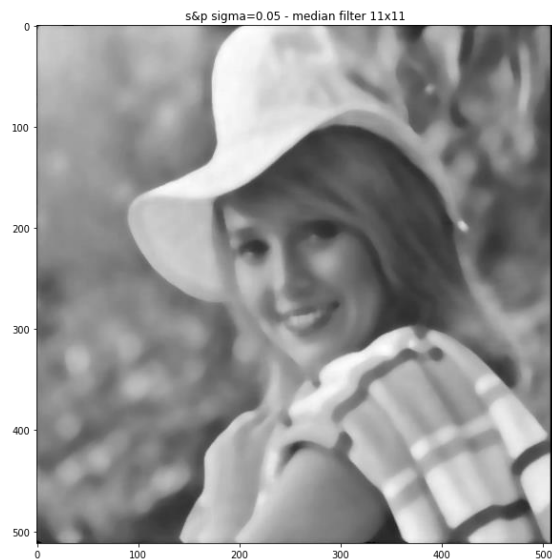
شکل 3-1-6-2 اعمال فیلتر لاپلاسین دو بار



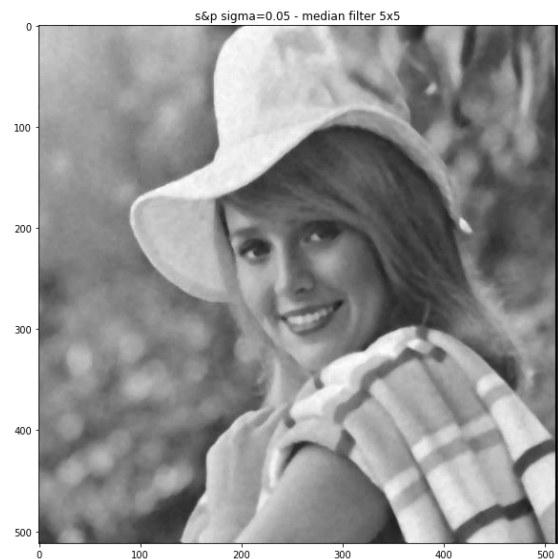
شکل 3-1-6-3 اعمال فیلتر لاپلاسین سه بار



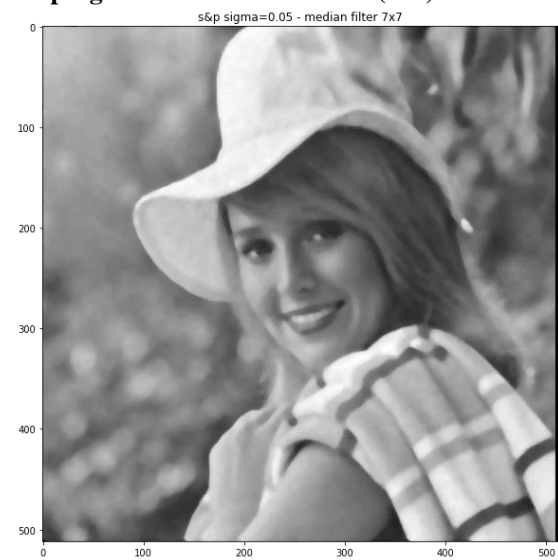
شکل 3-1-6-4 اعمال فیلتر لاپلاسین چهار بار



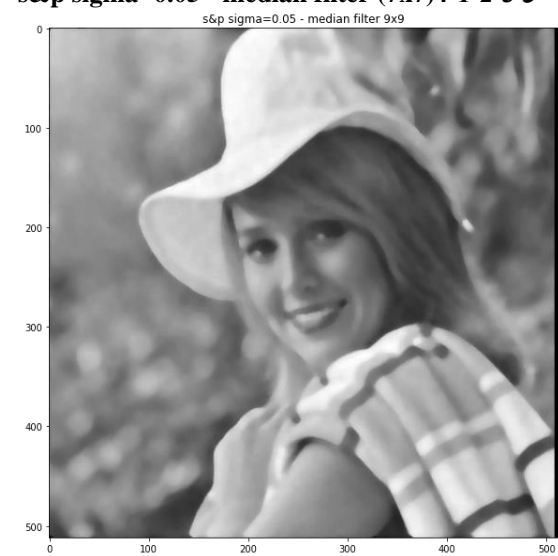
s&p sigma=0.05 - median filter 6-1-2-3 شكل  
(11x11)



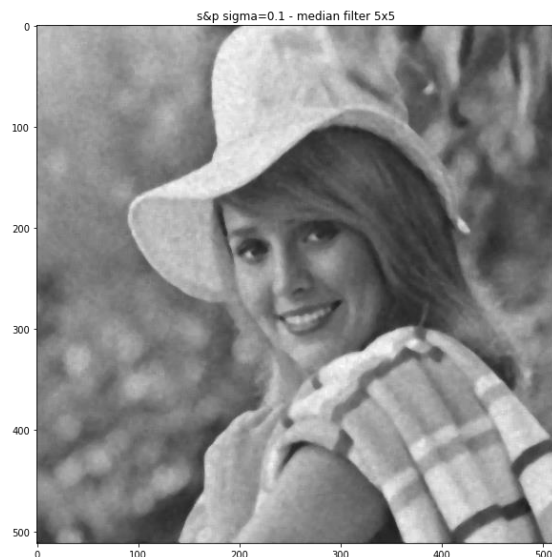
s&p sigma=0.05 - median filter (5x5)3-1-2-3 شكل



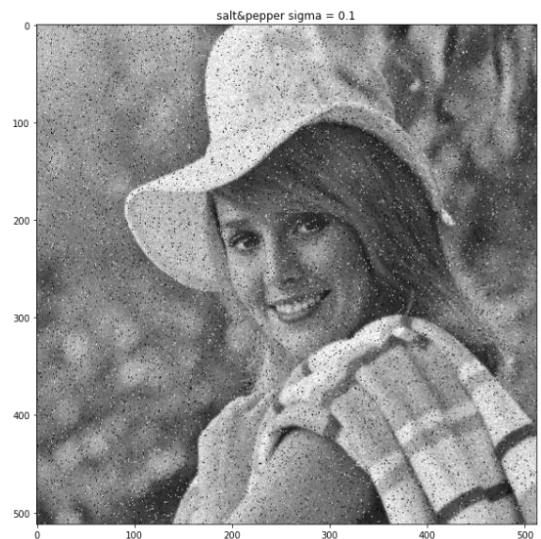
s&p sigma=0.05 - median filter (7x7)4-1-2-3 شكل



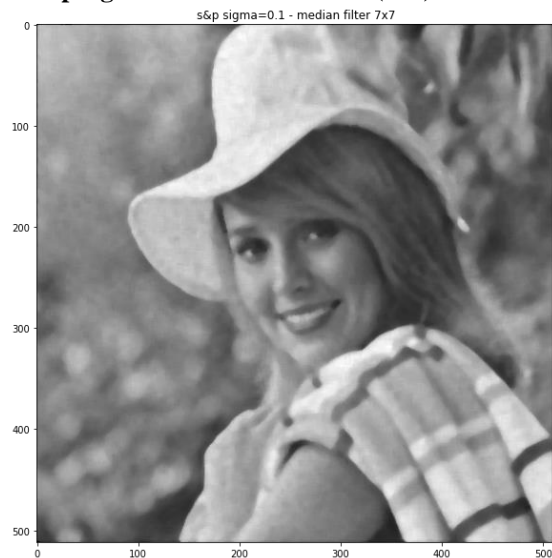
s&p sigma=0.05 - median filter (9x9)5-1-2-3 شكل



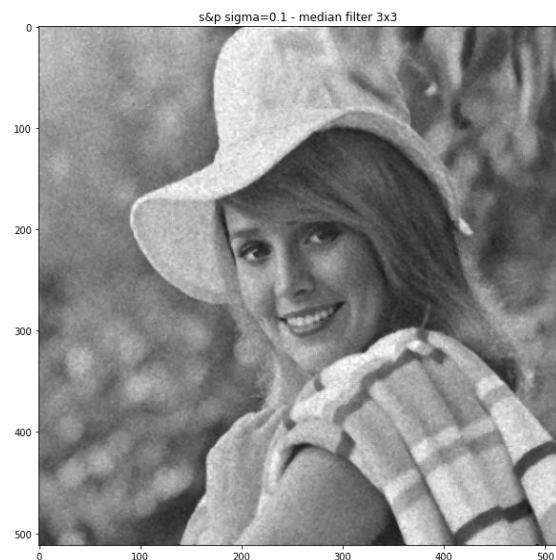
شکل 3-2-1-7 نویز نمک و فلفل  $\sigma = 0.1$  - median filter (5x5)



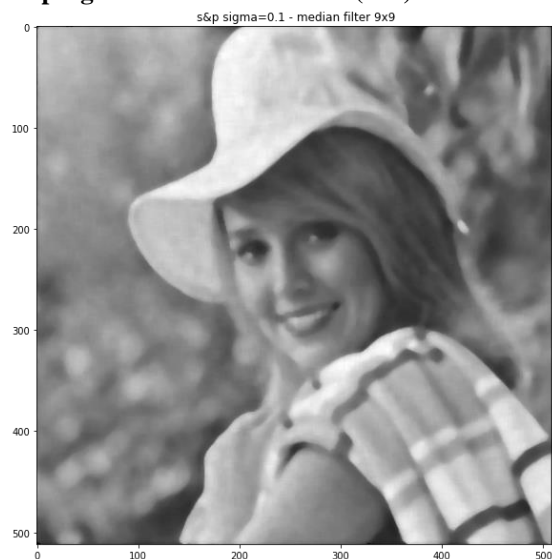
شکل 3-2-1-7 نویز نمک و فلفل  $\sigma = 0.1$



شکل 3-2-1-10 - median filter (7x7)  $\sigma = 0.1$

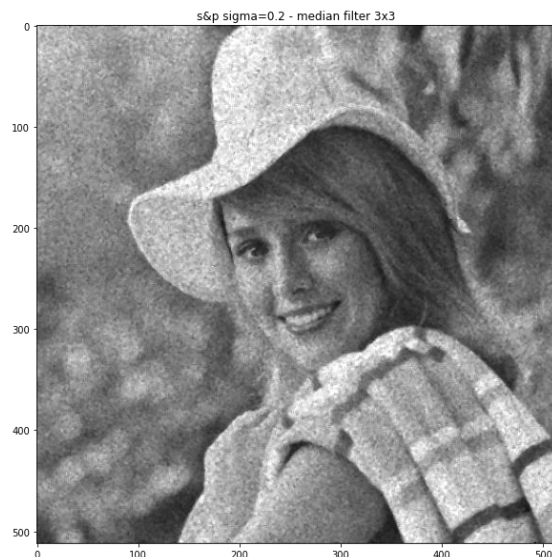


شکل 3-2-1-8 - median filter (3x3)  $\sigma = 0.1$

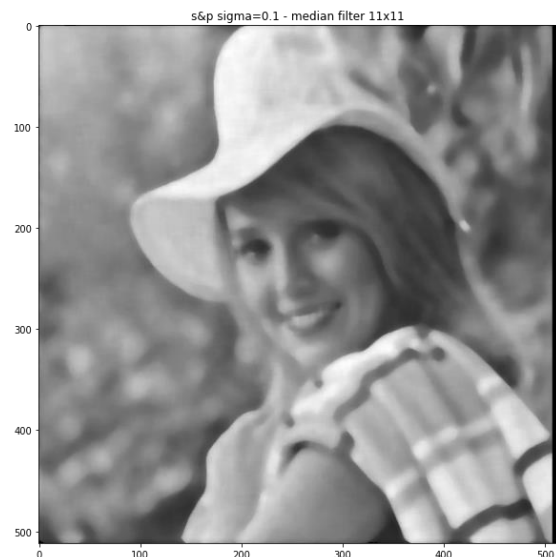


شکل 3-2-1-11 - median filter (9x9)  $\sigma = 0.1$

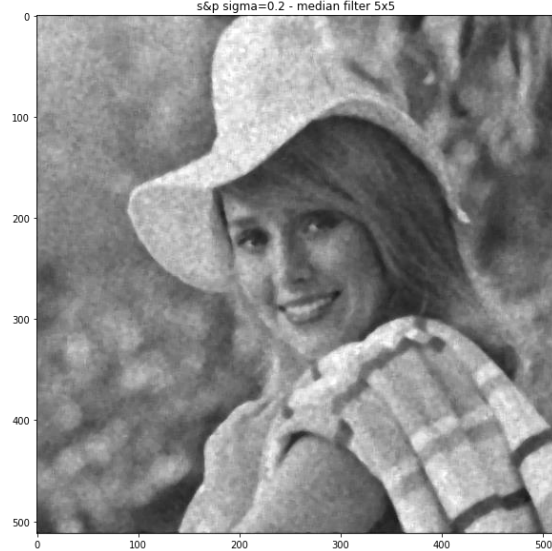
با توجه به افزایش مقدار نویز در این تصویر، این اندازه برای فیلتر نمیتواند نویز موجود را به خوبی از بین ببرد. اما با افزایش سایز فیلتر در تصویر 3-2-1-9 مشاهده میشود که نویز به طور کامل از بین رفته است. اما مشاهده میکنیم که به دلیل اینکه سایز فیلتر بزرگ است، مقداری از کیفیت تصویر کاسته شده است و دلیل این پدیده این است که فیلتر میانه هم مقداری باعث تضعیف و suppress شدن لبه ها در تصویر میشود.



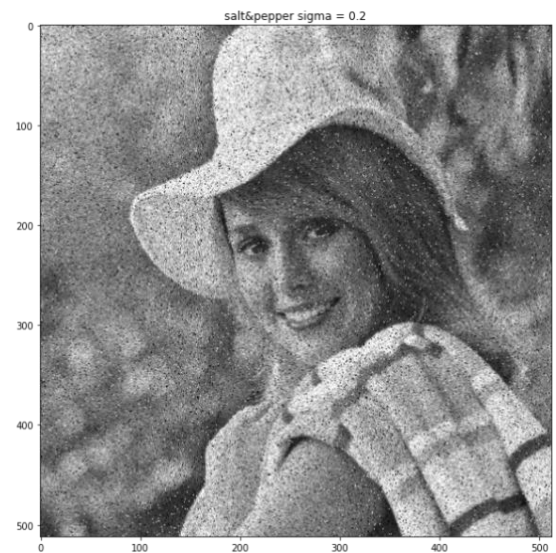
**شكل 14-1-2-3** s&p sigma=0.2 - median filter (3x3)



**شكل 12 -1-2-3** s&p sigma=0.1 - median filter (11x11)



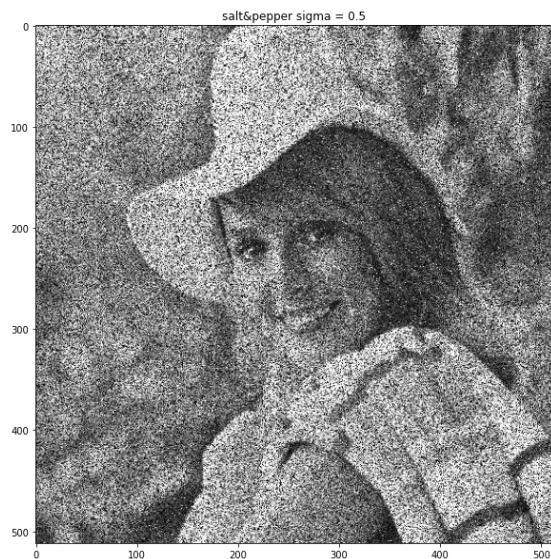
**شكل 15-1-2-3** s&p sigma=0.2 - median filter (5x5)



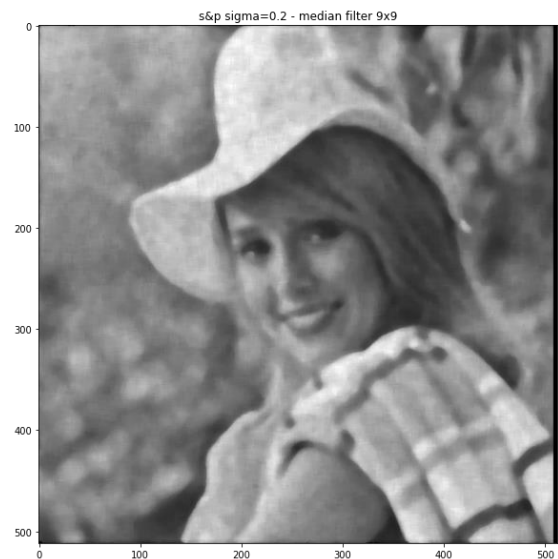
**شكل 13-1-2-3** sigma = 0.2 نويز نمک و فلفل



**شكل 16-1-2-3** s&p sigma=0.2 - median filter (7x7)



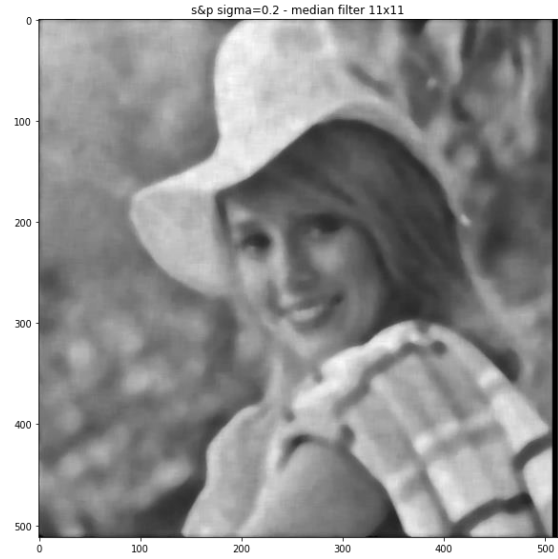
شکل 19-1-2-3 نویز نمک و فلفل sigma = 0.5



شکل 17-1-2-3 s&p sigma=0.2 - median filter (9x9)



شکل 18-1-2-3 s&p sigma=0.5 - median filter 20 (3x3)

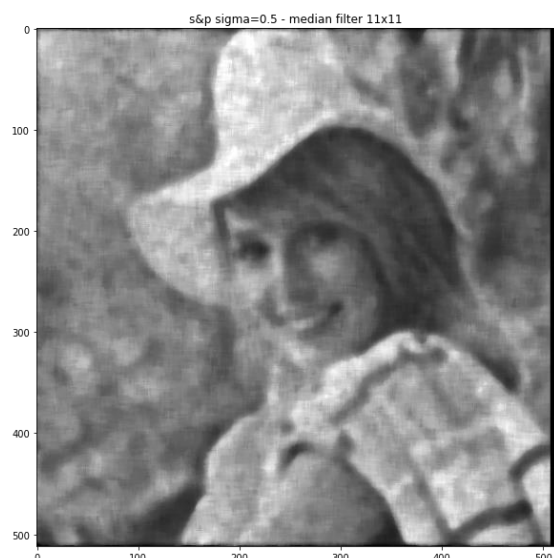


شکل 18-1-2-3 s&p sigma=0.2 - median filter 18 (11x11)



شکل 21-1-2-3 s&p sigma=0.5 - median filter (5x5)

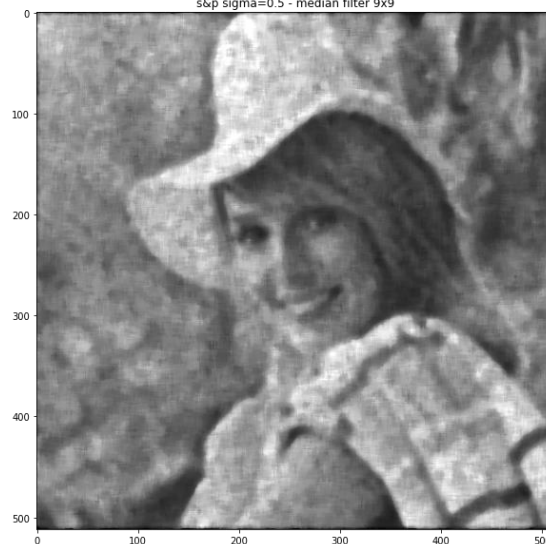




شکل 1-2-3 -1-2-3 median filter 24 -1-2-3  
sigma=0.5 (11x11)



شکل 1-2-3 -1-2-3 median filter (7x7)22 -1-2-3  
sigma=0.5



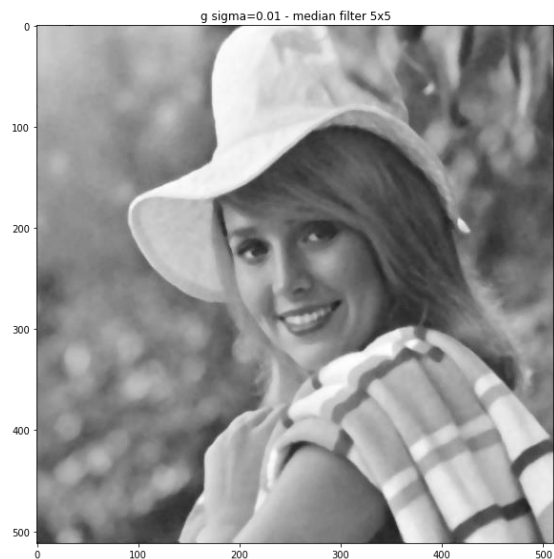
شکل 1-2-3 -1-2-3 median filter (9x9)23 -1-2-3  
sigma=0.5

	Report MSE 1-1-2-3 جدول				
	3x3	5x5	7x7	9x9	11x11
$\rho = 0.05$	813.885532	859.839096	899.951393	942.794144	988.814659
$\rho = 0.1$	833.660961	869.571842	907.882851	951.238842	998.175468
$\rho = 0.2$	926.647282	911.296482	938.130463	979.433563	1027.514576
$\rho = 0.5$	1585.927929	1196.418518	1126.808235	1142.120033	1184.815178

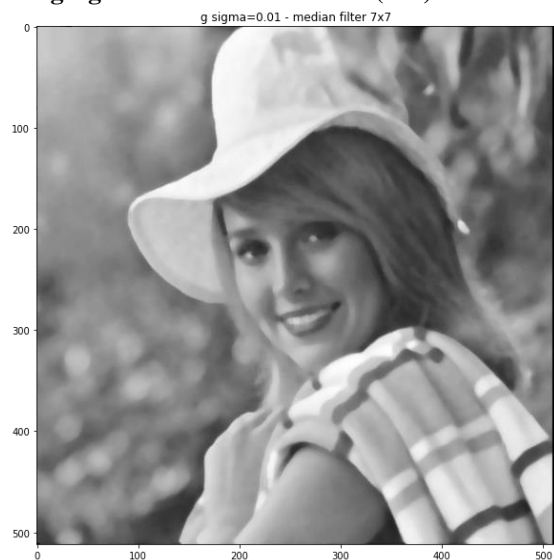
### gaussian noise 1-2-3

در این تمرین باید تصویر Elaine را با سه واریانس متفاوت نویز دار کنیم و سپس با فیلترهای میانه و میانگین با سایزهای مختلف تلاش کنیم تا نویز ایجاد شده رفع شود.

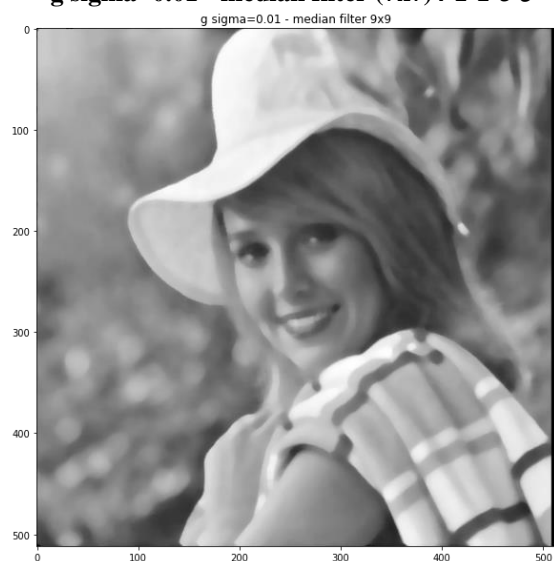
با توجه به مقادیری که در جدول 1-1-2-3 مشاهده میکنیم، هنگامی که واریانس نویز تصویر برابر با 0.05 باشد، فیلتری با اندازه 3x3 برای حذف این نویز موثر خواهد بود. این قضیه برای نویز با واریانس 0.1 نیز صدق میکند. اما هنگامی که واریانس نویز تصویر برابر با 0.2 باشد فیلتری با اندازه 5x5 و هنگامی که واریانس نویز برابر با 0.5 باشد فیلتری با اندازه 7x7 برای حذف این نویز مناسب می باشد.



شکل 3-2-2-3 g sigma=0.01 - median filter (5x5)

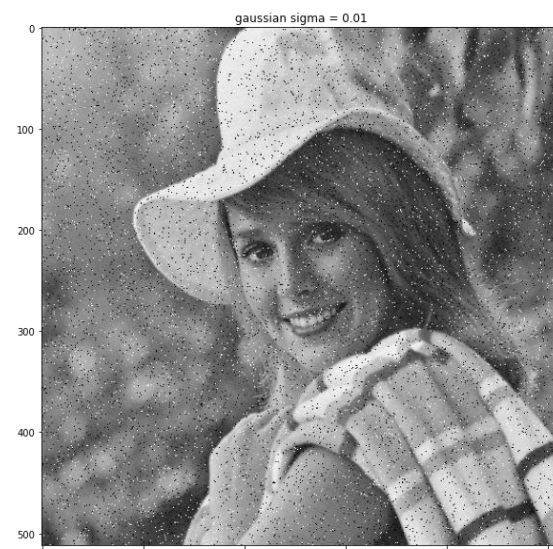


شکل 3-2-2-3 g sigma=0.01 - median filter (7x7)

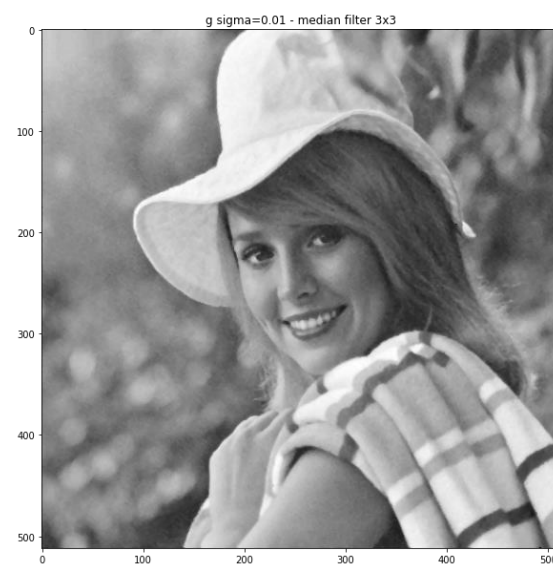


شکل 3-2-2-3 g sigma=0.01 - median filter (9x9)

در ابتدا نویز گوسین با واریانس 0.01 را به تصویر اعمال میکنیم. سپس با استفاده از فیلترهای میانه و میانگین گیر با سایزهای مختلف اقدام به حذف نویز ایجاد شده میکنیم.

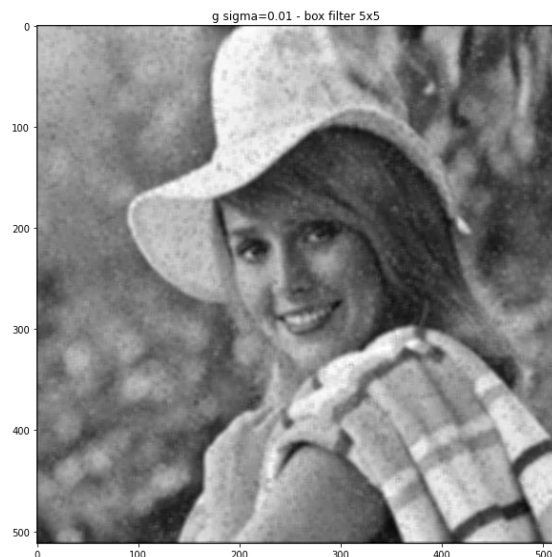


شکل 3-2-2-3 نویز گوسین sigma = 0.01

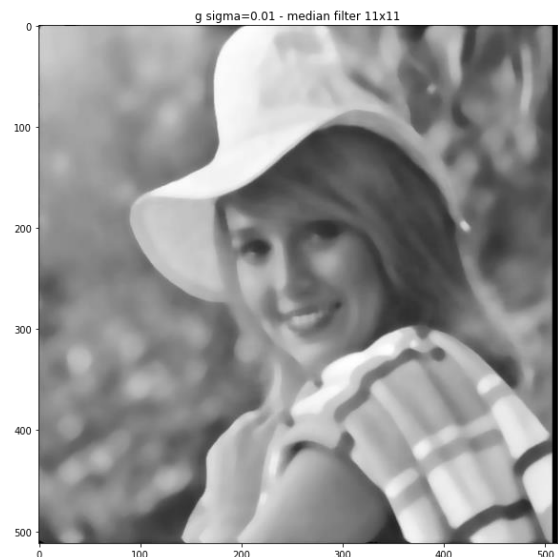


شکل 3-2-2-3 g sigma=0.01 - median filter (3x3)

همانطور که در تصویر مشاهده میکنیم نویز مقداری رفع شده است.

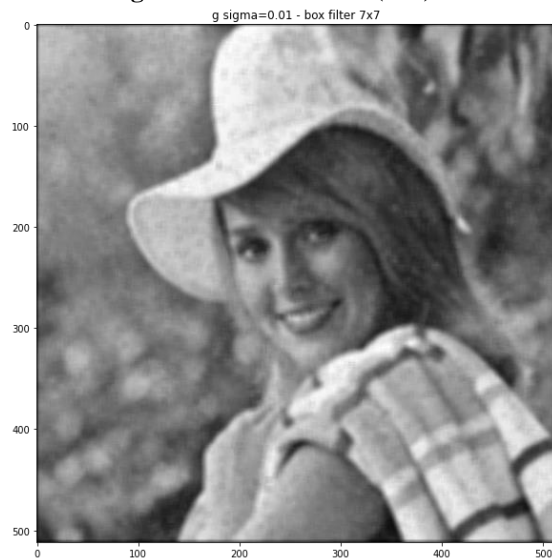


sigma=0.01 - box filter (5x5) شکل 8-2-2-3

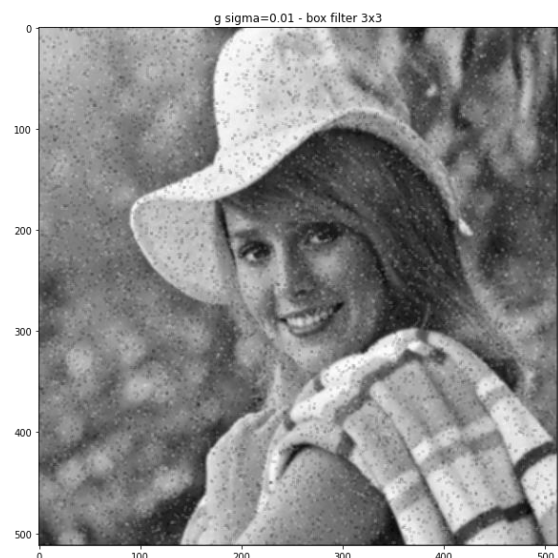


g sigma=0.01 - median filter (11x11) شکل 6-2-2-3

حال فیلتر میانگین گیر را با اندازه های مختلف جهت حذف نویز به تصویر اعمال میکنیم:

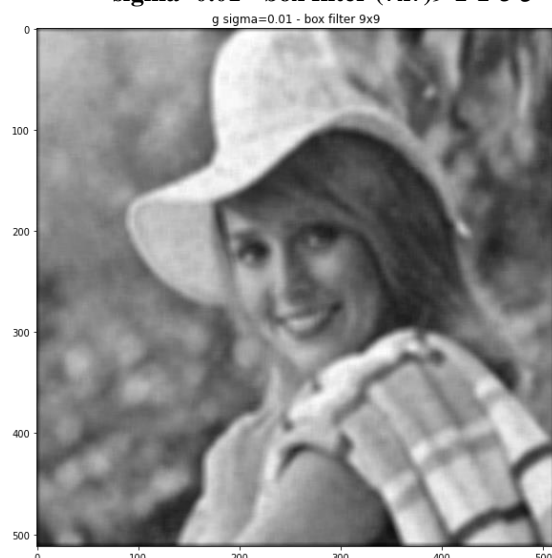


sigma=0.01 - box filter (7x7) شکل 9-2-2-3



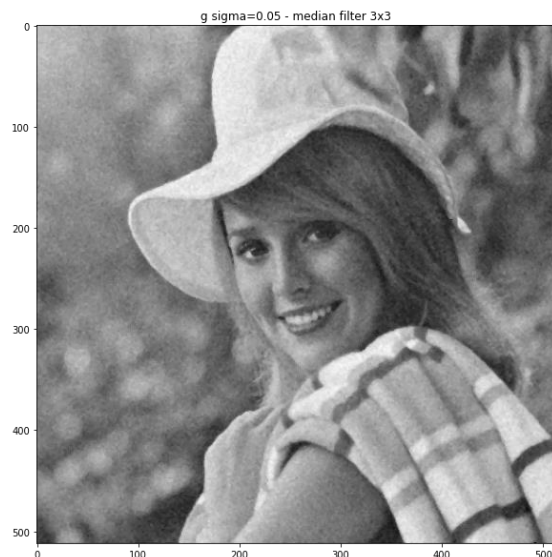
sigma=0.01 - box filter (3x3) شکل 7-2-2-3

مشاهده میکنیم که همانند نویز نمک و فلفل، نویز گوسی نیز توسط فیلتر میانگین گیر به خوبی قابل رفع نمی باشد. از آنجایی که برای نویزهای یکنواخت مانند دو نویز مذکور، فیلتر میانگین گیر باعث میشود نویزها بیشتر در تصویر پخش شوند. همینطور که مشاهده میکنیم تصویر 3-2-2-7 که یک فیلتر میانگین گیر 3x3 به آن اعمال شده است، کیفیت پایین تری از تصویر نویزدار اولیه دارد.

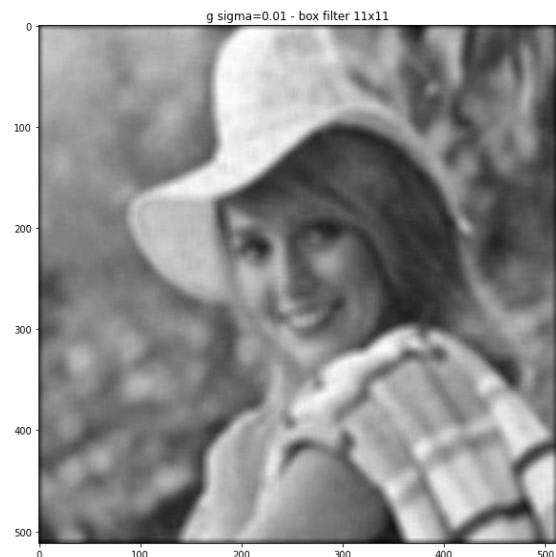


sigma=0.01 - box filter (9x9) شکل 10-2-2-3





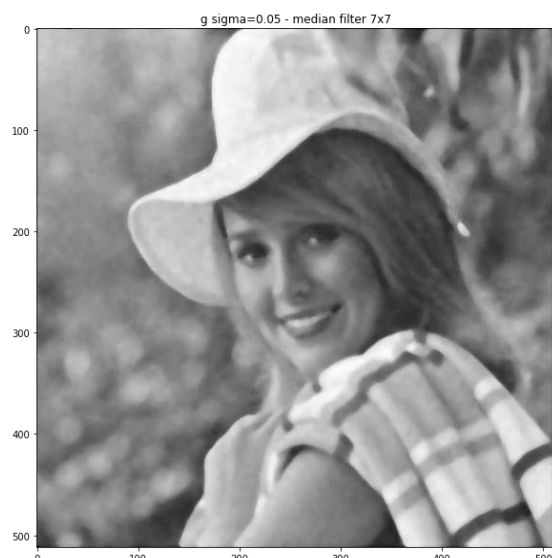
شکل 3-2-13 g sigma=0.05 - median filter (3x3)



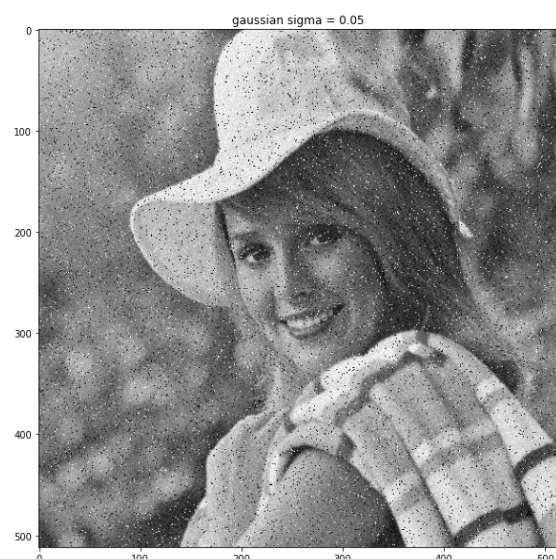
شکل 3-2-11 sigma=0.01 - box filter (11x11)

با توجه به افزایش مقدار نویز در تصویر فوق، این فیلتر با سایز فعلی نمیتواند نویز موجود را به خوبی از بین ببرد. اما با افزایش سایز فیلتر در تصویر 3-2-14 مشاهده میشود که نویز به طور قابل قبولی از بین رفته است اما به تبع آن و به دلیل افزایش سایز فیلتر، از کیفیت تصویر کاسته شده است.

با افزایش اندازه فیلتر میانگین گیر، تنها شاهد کاهش کیفیت تصویر و تقویت نویزها در آن می‌باشیم.

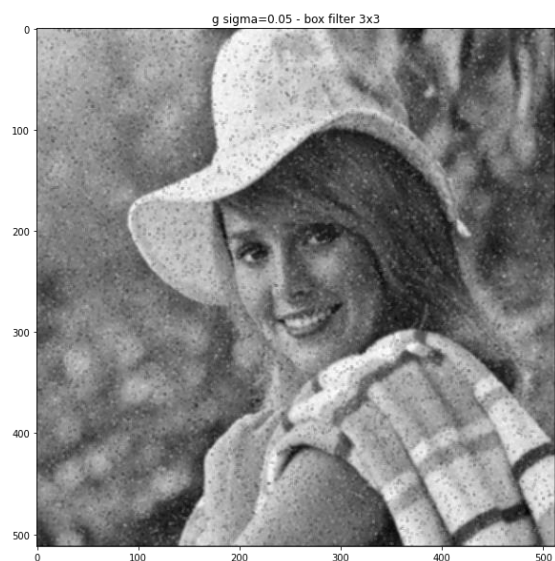


شکل 3-2-14 g sigma=0.05 - median filter (7x7)



شکل 3-2-12 sigma = 0.05 نویز گوسین

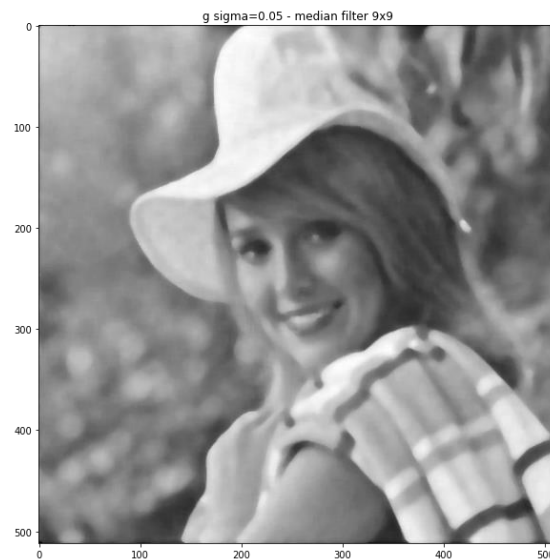
حال فیلتر میانگین گیر را با اندازه های مختلف جهت حذف نویز به تصویر اعمال میکنیم:



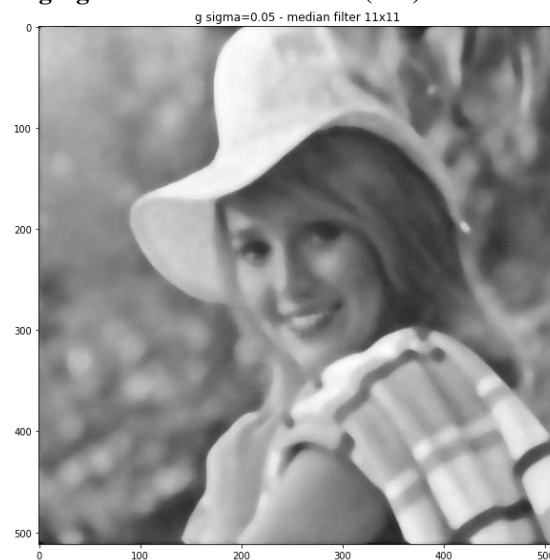
**sigma=0.05 - box filter (3x3) شکل 18-2-2-3**



**sigma=0.05 - box filter (5x5) شکل 19-2-2-3**



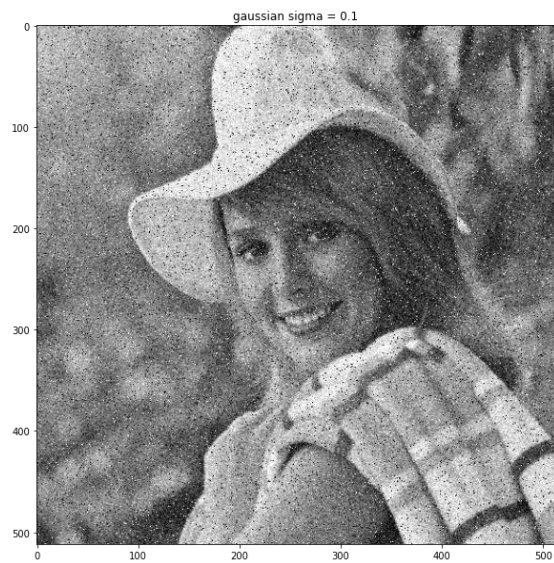
**g sigma=0.05 - median filter (9x9) شکل 15-2-2-3**



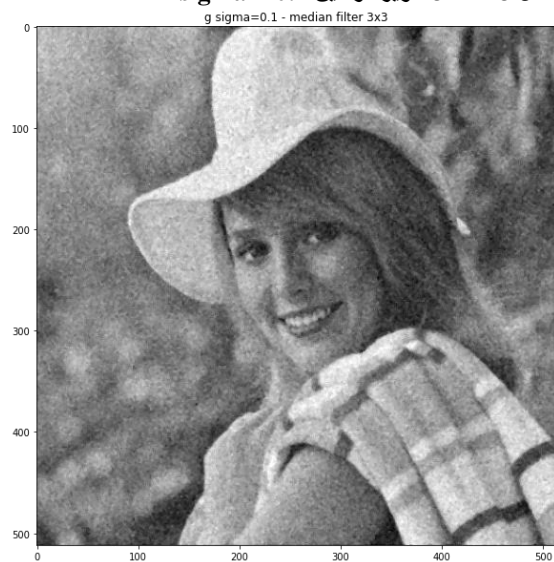
**g sigma=0.05 - median filter (11x11) شکل 16-2-2-3**



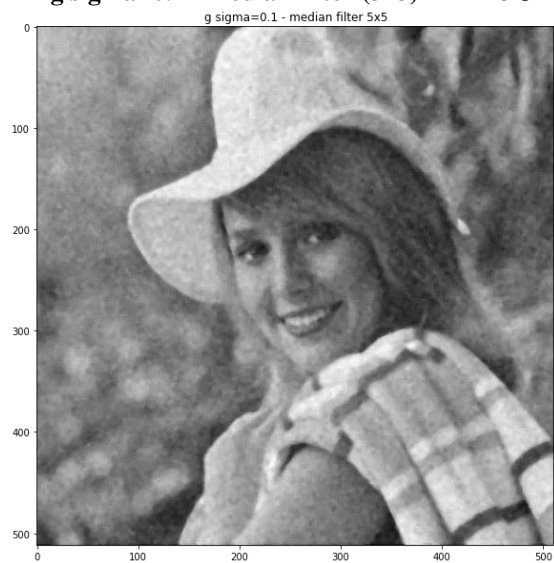
**g sigma=0.05 - median filter (5x5) شکل 17-2-2-3**



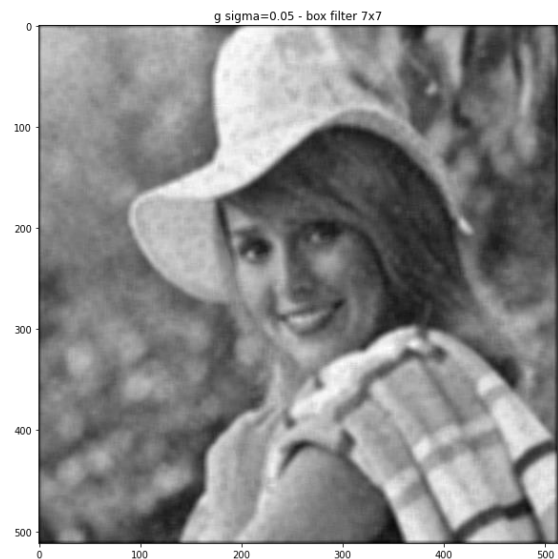
شکل 23-2-2-3 نویز گوسین  $\sigma = 0.1$



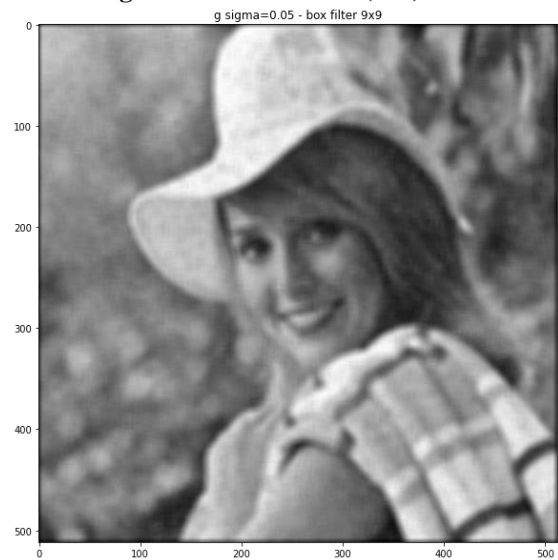
شکل 24-2-2-3 g sigma=0.1 - median filter (3x3)



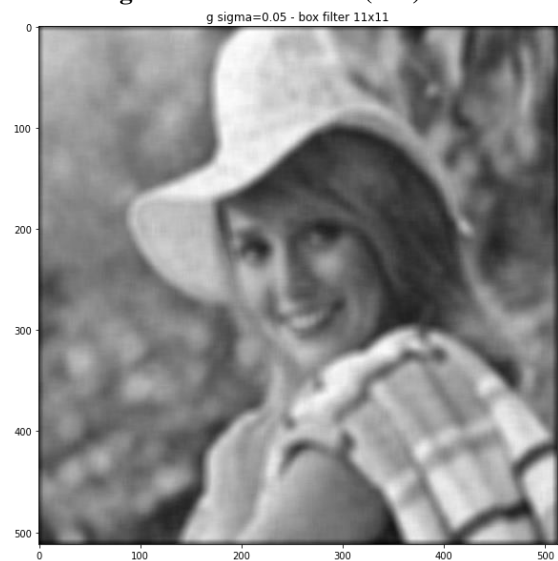
شکل 25-2-2-3 g sigma=0.1 - median filter (5x5)



شکل 20-2-2-3 sigma=0.05 - box filter (7x7)

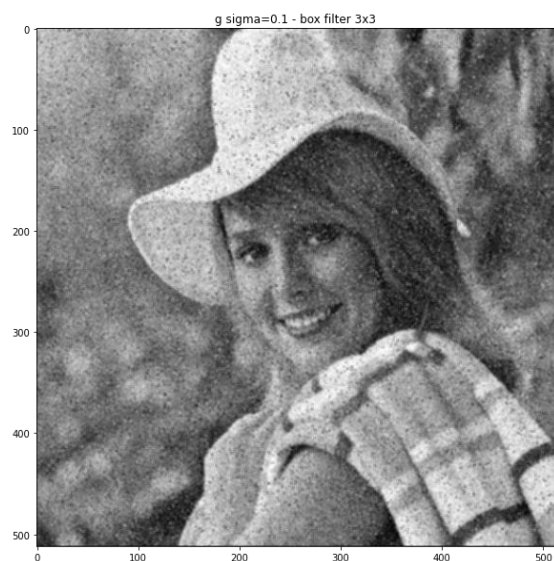


شکل 21-2-2-3 sigma=0.05 - box filter (9x9)



شکل 22-2-2-3 sigma=0.05 - box filter (11x11)

حال فیلتر میانگین گیر را با اندازه های مختلف جهت حذف نویز به تصویر اعمال میکنیم:



شکل 29-2-2-3 sigma=0.1 - box filter (3x3)



شکل 30-2-2-3 sigma=0.1 - box filter (5x5)



شکل 26-2-2-3 g sigma=0.1 - median filter (7x7)

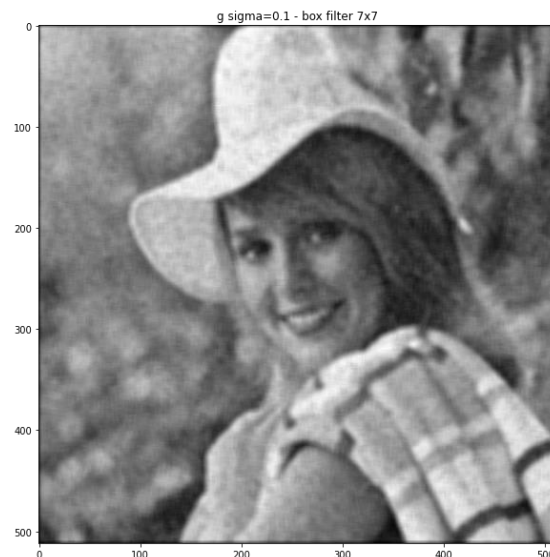


شکل 27-2-2-3 g sigma=0.1 - median filter (9x9)

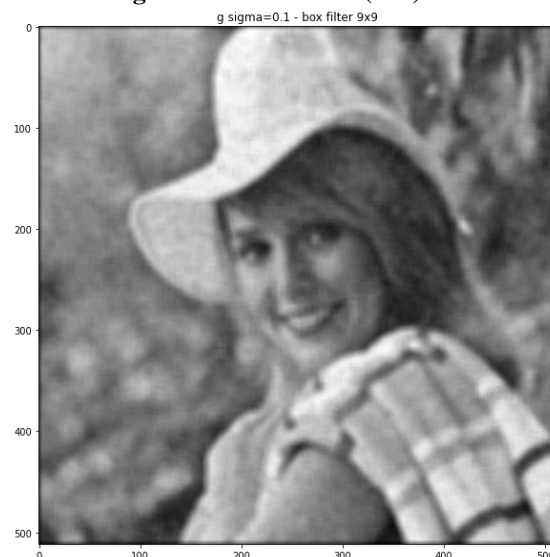


شکل 28-2-2-3 g sigma=0.1 - median filter (11x11)

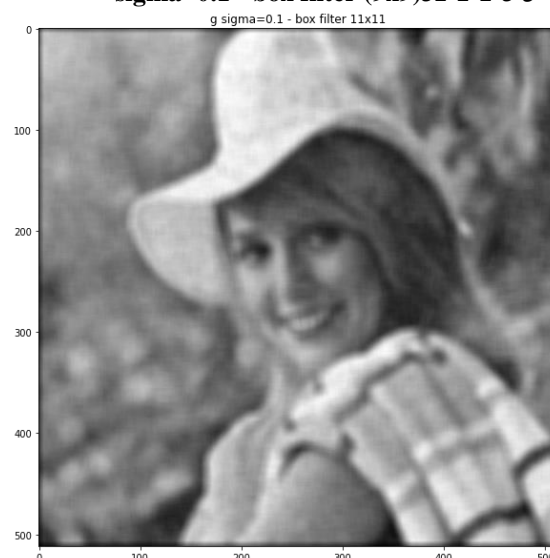
با بررسی جدول 3-2-2 می‌یابیم که در تمام واریانس‌های مختلف برای نویزدار کردن تصویر، فیلتر میانه نسبت به فیلتر میانگین گیر عملکرد بهتری داشته است. هنگامی که واریانس نویز تصویر برابر با 0.1 باشد فیلتری با اندازه 5×5 برای حذف این نویز مناسب می‌باشد. در بقیه حالات کوچک بودن اندازه فیلتر به حذف نویز کمک شایانی میکند.



شکل 3-2-2-31 sigma=0.1 - box filter (7x7)

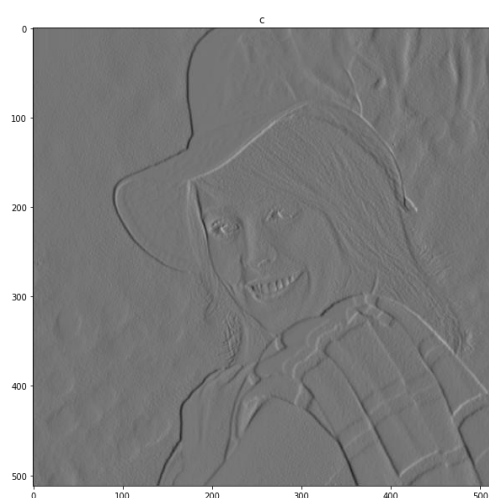


شکل 3-2-2-32 sigma=0.1 - box filter (9x9)



شکل 3-2-2-33 sigma=0.1 - box filter (11x11)

	جدول Report MSE 2-2-3									
	median 3x3	5x5	7x7	9x9	11x11	box filter 3x3	5x5	7x7	9x9	11x11
$\rho = 0.01$	808.815	857.21	897.753	940.64	986.15	719.79	806.985	853.229	897.595	940.66
$\rho = 0.05$	828.186	867.26	905.78	948.392	994.48	736.68	812.94	856.57	899.722	941.972
$\rho = 0.1$	899.124	897.718	926.888	968.066	1016.02	789.27	832.219	866.473	906.1612	947.02



شکل 3-4-3 فیلتر c

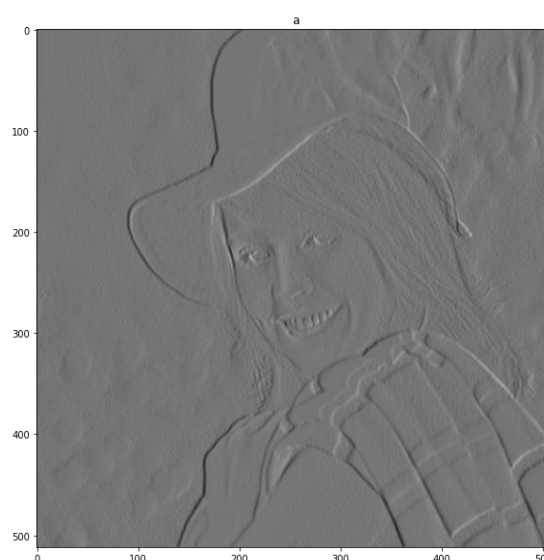
همانطور که قابل مشاهده است هر سه فیلتر لبه های عمودی را به عنوان خروجی به ما میدهند. ولی تفاوت هایی با یکدیگر دارند اولین تفاوتی که به چشم میخورد این است که فیلتر a لبه های نازک تری را به عنوان خروجی میدهد و فیلتر های b و c لبه های ضخیم تری را به عنوان خروجی میدهد.

### 2-4-3

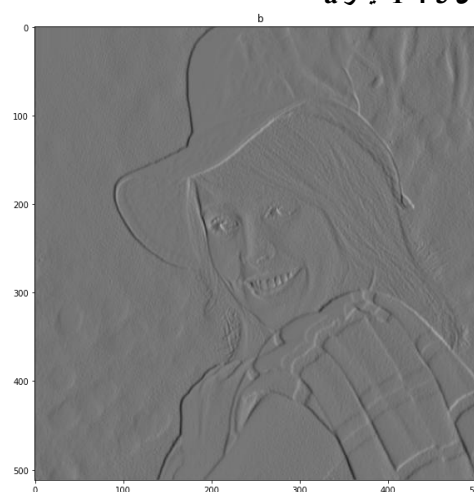
در این تمرین دو فیلتر روبرت را مشخص کرده است که فیلتر a فیلتری است که لبه های 45 درجه را به عنوان خروجی میدهد و فیلتر b فیلتری است که لبه های 135 درجه را به عنوان خروجی میدهد در نتیجه پس از ترکیب کردن نتایج مربوط به هر یک از این دو فیلتر خروجی تصویری است که لبه ها را در هر دو جهت 45 درجه و 135 میدهد و نکته ای که وجود دارد این است که فیلتر روبرت باعث تقویت بعضی از نویز ها نیز میشود که در خروجی هم مشخص است. در مقایسه با

## Edge Detection 4-3

### 1-4-3



شکل 1-4-3 فیلتر a



شکل 2-4-3 فیلتر b



```

                                ima
rr.shape[1]+2*pad), dtype=np.
uint8)
    padimarr[pad:padimarr.sh
ape[0]-pad,
                                pad:padimarr.sh
ape[1]-pad]=imarr
    return padimarr

```

```

def convolve(filter,image):
    m = image.shape[0]
    n = image.shape[1]

    filter_size = filter.shape
[0]
    a = b = filter_size//2
    image_padded = addpadding(
image, filter_size//2)

    image_conv = np.zeros(imag
e_padded.shape)

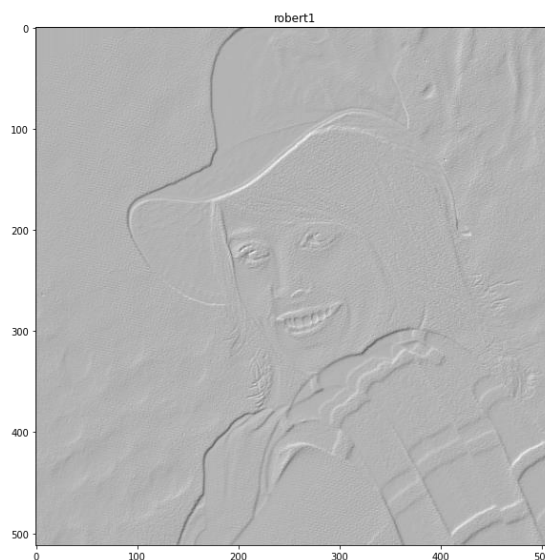
    for x in range(a, image_pa
dded.shape[0]-a):
        for y in range(b, image_
padded.shape[1]-b):
            arr = image_padded[x-
a : x-a+filter_size,
                                y-
b : y-b+filter_size]
            arr = arr.flatten()*fi
lter.flatten()
            image_conv[x][y] = arr
.sum()

    h_end = -a
    w_end = -b

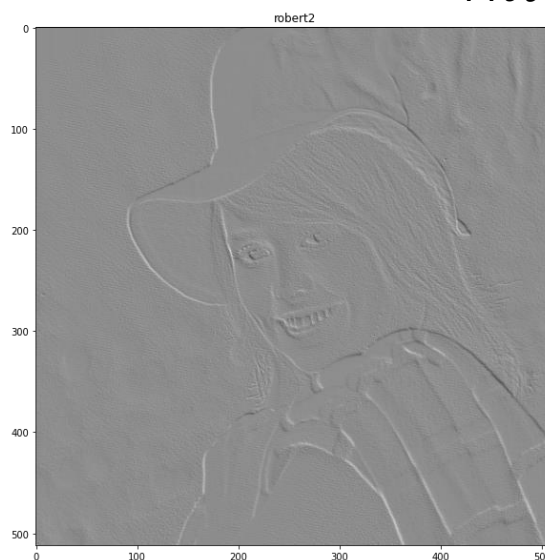
    if(a == 0):
        return image_conv[a:,b:w
end]

```

فیلتر های سوال قبلی نکته ای که وجود دارد این است که فیلتر روبرت به طور کلی نسبت به فیلتر سوبل نویز ها را بیشتر تقویت میکند در ادامه خروجی مربوط به فیلتر روبرت را خواهیم دید.



شکل 4-4-3



شکل 5-4-3

## Code 4

<https://colab.research.google.com/drive/1CBZaY01Y7a2p6sBp0HB4LLeCd33X9nAp?usp=sharing>

```

def addpadding(source,pad):
    imarr = np.array(source)
    padimarr = np.zeros((ima
rr.shape[0]+2*pad,

```

```

        box_filter=np.ones((n,n)
)*(1/n**2)
        return box_filter

def median_filter(image, filter_size):
    if filter_size%2==0:
        filter_size=filter_size+1
    elif filter_size<=2:
        filter_size=3
    temp = []
    indexer = filter_size // 2
    image_out = np.zeros((len(image),len(image[0])))
    for i in range(len(image)):
        for j in range(len(image[0])):
            for z in range(filter_size):
                if i + z - indexer < 0 or i + z - indexer > len(image) - 1:
                    for c in range(filter_size):
                        temp.append(0)
                    else:
                        if j + z - indexer < 0 or j + z - indexer > len(image) - 1:
                            temp.append(0)
                        else:
                            for k in range(filter_size):
                                temp.append(image[i + z - indexer][j + k - indexer])
                            temp.sort()

```

```

    if(b == 0):
        return image_conv[a:h_end, b:]
    return image_conv[a:h_end, b:w_end]

def create_gaussian_filter(sigma):
    n= 2*math.floor(3*sigma)+1
    if(n <= 2):
        n = 3
    elif(n%2 == 0):
        n = n+1
    gaussian_filter=np.zeros((n,n))
    rows, cols=gaussian_filter.shape
    for r in range(rows):
        for c in range(cols):
            gaussian_filter[r][c]=(1/(math.pi*sigma**2))*math.exp(-(r**2 + c**2)/(2*sigma**2))
    return gaussian_filter

def box_filter(n):
    if(n <= 2):
        n = 3
    elif(n%2 == 0):
        n = n+1

```



```

blur4 = convolve(box_f3,blur
3)
blur5 = convolve(box_f3,blur
4)
blur6 = convolve(box_f3,blur
5)

df = pd.DataFrame(
    [
        ('sigma = 0.05', MSE(elai
ne, sp1noise_median_f3), MS
E(elaine, sp1noise_median_f5
),
            MSE(elai
ne, sp1noise_median_f7), MSE
(elaine, sp1noise_median_f9)
, MSE(elaine, sp1noise_media
n_f11)
        ),
        ('sigma = 0.1', MSE(elai
ne, sp2noise_median_f3), MSE
(elaine, sp2noise_median_f5)
,
            MSE(elai
ne, sp2noise_median_f7), MSE
(elaine, sp2noise_median_f9)
, MSE(elaine, sp2noise_media
n_f11)
        ),
        ('sigma = 0.2', MSE(elai
ne, sp3noise_median_f3), MSE
(elaine, sp3noise_median_f5)
,
            MSE(elai
ne, sp3noise_median_f7), MSE
(elaine, sp3noise_median_f9)
, MSE(elaine, sp3noise_media
n_f11)
        ),
        ('sigma = 0.5', MSE(elai
ne, sp4noise_median_f3), MSE

```

```

        image_out[i][j]
= temp[len(temp) // 2]
        temp = []
        return image_out

sp1_noise_img = np.array(ran
dom_noise(elaine, mode='spec
kle',var=0.05**2)*255, dtype
= 'uint8')
sp2_noise_img = np.array(ran
dom_noise(elaine, mode='spec
kle',var=0.1**2)*255, dtype
= 'uint8')
sp3_noise_img = np.array(ran
dom_noise(elaine, mode='spec
kle',var=0.2**2)*255, dtype
= 'uint8')
sp4_noise_img = np.array(ran
dom_noise(elaine, mode='spec
kle',var=0.5**2)*255, dtype
= 'uint8')

laplacian_mask = np.array([[
0, -1, 0], [-1, 5, -
1], [0, -1, 0]])

box_f3 = box_filter(3)
blur1 = convolve(box_f3,elai
ne)
blur2 = convolve(box_f3,blur
1)
blur3 = convolve(box_f3,blur
2)

```

```

(elaine, sp4noise_median_f5)
',
                                MSE(elai
ne, sp4noise_median_f7), MSE
(elaine, sp4noise_median_f9)
, MSE(elaine, sp4noise_media
n_f11)
)
],
    columns=('', '3x3', '5x5
', '7x7', '9x9', '11x11')
)

```

```

g1_noise_img = np.array(rand
om_noise(elaine, mode='gauss
ian',var=0.01**2)*255, dtype
= 'uint8')
g2_noise_img = np.array(rand
om_noise(elaine, mode='gauss
ian',var=0.05**2)*255, dtype
= 'uint8')
g3_noise_img = np.array(rand
om_noise(elaine, mode='gauss
ian',var=0.1**2)*255, dtype
= 'uint8')

```