فیلترینگ مکانی

علی نصیری سروی

اطلاعات گزارش

تاریخ: 99/01/29

واژگان کلیدي:
فیلترینگ مکانی
ماسک
ماسکینگ غیرشارپ
فیلتر میانگین
فیلتر میانه
نویز نمک و فلفل
نویز گاوسین
شناسایی لبه

چکیده

در این تمرین در ابتدا به برخی فیلترهای خطی و غیرخطی پرداخته و از آن ها برای رفع نویز استفاده میکنیم.پس از آن تصویری دلخواه با کیفیت پایین را سعی میکنیم بهبود دهیم.

در بخشی دیگر به شناسایی لبه و روش های آن میپردازیم.

در نهایت روش های ماسکینگ غیرشارپ را معرفی کرده و به کار میبریم.

1-مقدمه

فیلترینگ مکانی در طیف گسترده ای از کاربردهای پردازش تصاویر به کار می‌روند و در نتیجه فهمیدن عمقی آنها بسیار مهم می‌باشد.

فیلترینگ مکانی یک تصویر را به این صورت تغییر میدهد که مقدار جدید هر پیکسل به کمک یک تابع از مقدار آن پیکسل و همسایه های آن بدست می آید.

اگر عملیاتی که بر روی پیکسل اجرا میشود خطی باشد، فیلتر را *فیلتر مکانی خطی* می‌نامیم.در غیر اینصورت آن را *فیلتر مکانی غیرخطی* گوییم.

2-شرح تکنیکال

فیلتر مکانی خطی

یک فیلتر مکانی خطی عملیات جمعِ ضرب ها را بین تصویر f و فیلتر w انجام میدهد.ضرایب فیلترمشخص کننده خواص فیلتر می‌باشد.به این فیلتر ، *ماسک* یا *پنجره* و یا *کرنل* نیز می‌گویند.

فیلتر مکانی غیرخطی

یک فیلتر مکانی غیرخطی عملیات غیرخطی را ب تصویر f و فیلتر w انجام میدهد.برای مثال فیلتر میانه، برای هر پیکسل،میانه خود و همسایه های درون پنجره اش را به عنوان مقدار پیکسل در نظر میگیرد.

3.1 فیلتر میانگین یکنواخت (فیلتر جعبه)

این فیلتر،یک فیلتر خطی است.اگر فیلتر جعبه $n \times n$ داشته باشیم،ضرایب فیلتر به صورت $\frac{1}{n^2}$ میباشد.

3.1.1 عیب فیلتر میانگین یکنواخت برای هموارسازی؟

یکی از مشکلات دیگر این فیلتر،حساس بودن آن به نویز میباشد.زیرا نویز معمولا با پیکسل های اطرافش اختلاف مقدار زیادی دارد و در میانگین گیری میتواند تاثیر منفی بسزایی داشته باشد.

3.1.2 با تکرار فیلتر چه اتفاقی برای ویژگی های منفی می افتد؟

ویژگی های منفی در ابتدا زیاد میشوند اما به مرور با تکرار فیلتر تصویر تغییری پیدا نمیکند.دلیل این اتفاق قضیه آماری centeral limit میباشد.با تکرار فیلتر سطح خاکستری پیکسل های تصویر به سمت توزیع نرمال همگرا میشود.

3.1.3 با تکرار فیلتر چه اتفاقی برای تصویر می افتد؟

با تکرار فیلتر برروی تصویر،تصویر blur تر میشود.دلیل آن این است که میانگین گیری پشت سر هم باعث میشود تصویر غیرشفاف شود.

3.2 فیلتر میانه

این فیلتر،یک فیلتر غیرخطی است.در یک پنجره مشخص برای یک پیکسل خاص،میانه مقادیر پیکسل های درون پنجره را به عنوان مقدار جدید پیکسل در نظر میگیریم.

3.2.1 اعمال فیلتر میانه بر روی تصویر با نویز نمک و فلفل

چون در نویز فلفل و نمک،پیکسل های خراب،مقدار 0 یا 255 دارند و با احتمال p میانه پنجره نخواهند بود که در صورت انتخاب پنجره با سایز مناسب میتوان این احتمال را بالا برد.در این صورت میتوان پیکسل های خراب شده را به کمک همسایه هایشان بهبود داد.

در نتیجه فیلتر میانه بر روی این نوع نویز عملکرد خوبی دارد.

برای انتخاب سایز پنجره مناسب بر اساس میزان density نویز میتوان تصمیم گرفت.اگر نویز در تصویر چگالی زیادی داشته باشد،با سایز پنجره کم ممکن است تصویر بطور کامل خراب شود.

حالتی را فرض کنید چگالی نویز فلفل 0.44 باشد.یعنی از هر 9 پیکسل به طور میانگین 4 پیکسل کامل تیره است. حال فرض کنید سایز پنجره 3×3 باشد.حال اگر بدشانس باشیم،به جای 4 پیکسل در این پنجره 9 تایی 5 پیکسل فلفل داریم. چون بیش از نصف اعداد درون پنجره مقدار 0 دارند مطمئنیم فیلتر به اشتباه مقدار 0 که نویز میباشد را انتخاب میکند.

حال همین شرایط را در نظر بگیرید با این تفاوت که سایز پنجره 7×7 شده باشد.در 49 پیکسل با چگالی نویز 0.44 میتوان گفت به طور میانگین 22 پیکسل خراب میباشد.برای اینکه میانه عدد نویز دار باشد باید 3 پیکسل بیشتر خراب شود که احتمال آن از خراب شدن یک پیکسل بیشتر،کمتر میباشد.

این مثال نشان میدهد که در صورتی که چگالی نویز زیادتر شود با افزایش سایز پنجره میتوان نتیجه بهتری را بدست آورد.

اما باید به این نکته توجه کرد که در صورت افزایش سایز پنجره،تصویر blur خواهد شد.

3.2.2 اعمال فیلتر میانگین و میانه بر روی تصویر با نویز گاوسین

همانطور که گفته شد فیلتر میانگین بر روی تصاویر با نویز عملکرد ضعیفی دارد. هر چقدر سایز فیلتر بزرگتر باشد،این ضعف عمیق تر میشود.فرض کنید که نقاط سفید نویز در تصویر شما وجود دارد.با توجه به اینکه در فیلتر میانگین تفاوت های زیاد تاثیر زیادی میگذارد،کل تصویر شما رو به سفیدی خواهد رفت.هرچقدر فیلتر میانگین بزرگتری داشته باشید،این شدت تغییرات بیشتر خواهد بود.

فیلتر میانه بر روی نویز گاوسین عملکرد قابل قبولی دارد اما قابل قیاس با عملکرد آن بر روی نویز فلفل و نمک نیست.دلیل آن عملکرد عالی فیلتر میانه بر روی نویز فلفل و نمک میباشد.

3.3 بهبود کیفیت تصویر

برای بهبود کیفیت تصویر از روش highboost استفاده کرده ایم.این روش به این صورت است که ابتدا یک تصویر غیرشارپ به کمک فیلتر گاوسی با پنجره 5×5 محاسبه میکنیم.سپس یک ماسک را از تفاضل تصویر اصلی با تصویرغیرشارپ حساب میکنیم:

$$g_{mask}(x,y) = f\ (x,y) - \bar{f}(x,y)$$

در نهایت این ماسک را به ضریب مناسب به تصویر اصلی اضافه میکنیم:

$$g(x,y) = f\ (x,y) + k.gmask(x,y)$$

که در روش تقویت بالا این ضریب بزرگتر از 1 میباشد.

سپس بعد از اعمال فیلتر تقویت بالا از فیلتر میانه با سایز 3×3 استفاده میکنیم.

3.4 شناسایی لبه

3.4.1 فیلترهای شناسایی لبه عمودی

سه فیلتر زیر در این بخش بررسی شده اند:

$$f_1 = \frac{1}{2}\begin{bmatrix}1 & 0 & -1\end{bmatrix}$$

$$f_2 = \frac{1}{6}\begin{bmatrix}1 & 0 & -1\\1 & 0 & -1\\1 & 0 & -1\end{bmatrix}$$

$$f_3 = \frac{1}{8}\begin{bmatrix}1 & 0 & -1\\2 & 0 & -2\\1 & 0 & -1\end{bmatrix}$$

این سه فیلتر لبه های عمودی تصویر را شناسایی میکنند. فیلتر دوم بسیار شبیه به فیلتر سوم میباشد با این تفاوت که اهمیت کمتری به پیکسل های کناری پیکسل مرکزی میدهد.در نتیجه شدت لبه کمتر میباشد.

فیلتر اول با دو فیلتر دیگر تفاوت دارد.در واقع این فیلتر لبه های بیشتری را شناسایی میکند حتی اگر در واقعیت لبه خاصی در جایی که این فیلتر شناسایی کرده وجود نداشته باشد.زیرا این فیلتر برای فعال شدن فقط دو پیکسل کناری را در نظر میگیرد در حالی که دو فیلتر دیگر 4 پیکسل مورب را نیز در نظر میگیرند و زمانی که لبه ای را پیدا کنند با اطمینان خاطر بیشتری میتوان گفت که آن واقعا یک لبه است.

فیلتر اول به دو فیلتر دیگر از نظر بار محاسباتی تفاوت دارد و استفاده از فیلتر اول بهینه تر میباشد.دلیل آنرا در بخش بعد ذکر میکنیم.

3.4.1 فیلترهای Robert

دو فیلتر روبرت به فرم زیر داریم:

$$f_x = \begin{bmatrix}-1 & 0\\0 & 1\end{bmatrix}$$

$$f_y = \begin{bmatrix}0 & -1\\1 & 0\end{bmatrix}$$

فیلتر اول تخمین مشتق افقی را محاسبه کرده و لبه های افقی را پیدا میکند.

فیلتر دوم تخمین مشتق عمودی را محاسبه کرده و لبه های عمودی را پیدا میکند.

تفاوت این فیلترها با فیلترهای بخش قبل در این است که فیلترهای روبرت لبه های جزئی بیشتری را شناسایی

میکنند که اگر صرفا برایمان کلیت لبه ها مهم باشد برای ما ایجاد نویز میکنند.ولی فیلترهای 2 و 3 بخش قبل جزئیات کلی لبه ها را بدست می آورند.ولی اگر جزئیات ریز برایمان مهم باشد میتوان از فیلترهای روبرت استفاده نمود.

در مورد حجم محاسبات فیلترهای روبرت بهتر(همانطور که فیلتر f_1 از f_2, f_3 بهینه تر است) میباشند.

زیرا برای هر پیکسل 4 ضرب و 3 جمع خواهند داشت.اما فیلترهایی مانند سوبل 9 ضرب و 8 جمع برای هر پیکسل خواهند داشت. به طور کلی هرچقدر سایز فیلتر کوچکتر باشد،از نظر محاسبه تصویر بار محاسباتی کمتری خواهد داشت.

تصویر به کمک فیلتر sobel (فیلتر دوم بخش قبل)

تصویر به کمک فیلتر روبرت

همانطور که در تصاویر مشخص است فیلتر روبرت جزئیات بیشتری را محاسبه میکند.

3.5 ماسکینگ غیرشارپ

3.5.1 یک فیلتر ماسکینگ غیرشارپ ساده به فرم زیر میباشد:

$$(1 - \alpha)\mathrm{I} + \alpha\mathrm{I}' = \mathrm{I} + \alpha(\mathrm{I}' - \mathrm{I})$$

که در آن I تصویر اصلی،I' تصویر هموارسازی شده بوده و α یک پارامتر میباشد که آن را به صورت دستی انتخاب میکنیم.در صورتی که مقدار α زیاد باشد،تاثیر تصویر هموارشده بر روی نتیجه بیشتر میشود و هرچه مقدار آن کمتر باشد،تصویر اصلی بیشترین تاثیر را بر روی نتیجه خواهد گذاشت. در واقع به این معنا است که اگر مقدار α کم باشد تصویر اصلی تغییرات کمی خواهد داشت و اگر زیاد باشد،تغییرات آن زیاد خواهد بود. برای بدست آوردن مقدار بهینه آلفا میتوان مقادیر مختلف آلفا را به صورت یکنواخت(مثلا[1 0.8 0.6 0.4 0.2]) تست کرد و بهترین آنها را انتخاب کرد.سپس حول عدد بدست آمده ضرایب مختلف را امتحان کرد تا نتیجه مطلوب بدست آید.

3-شرح نتایج

3.1.3 نتیجه تکرار فیلتر میانگین

تصویر اصلی لنا

تصویر با 5 تکرار

تصویر با 1 تکرار

تصویر با 10 تکرار

تصویر با 20 تکرار

تصویر با 50 تکرار

همانطور که مشاهده میشود با تکرار تصویر،تصویر blur میشود.

3.2.1 نتیجه اعمال فیلتر میانه بر روی تصویر با نویز نمک و فلفل

ابتدا به تصویر اصلی نویز مورد نظر را اضافه میکنیم:

تصویر با نویز نمک و فلفل $\rho = 0.05$

تصویر با نویز نمک و فلفل $\rho = 0.1$

تصویر با نویز نمک و فلفل $\rho = 0.2$

سپس فیلتر میانه با سایز های مختلف را به آن اعمال میکنیم:

تصویر با نویز $\rho = 0.05$ و فیلتر با سایز 3×3

تصویر با نویز $\rho = 0.05$ و فیلتر با سایز 7×7

تصویر با نویز $\rho = 0.05$ و فیلتر با سایز 5×5

تصویر با نویز $\rho = 0.05$ و فیلتر با سایز 9×9

تصویر با نویز $\rho = 0.1$ و فیلتر با سایز 3×3

تصویر با نویز $\rho = 0.1$ و فیلتر با سایز 7×7

تصویر با نویز $\rho = 0.1$ و فیلتر با سایز 5×5

تصویر با نویز $\rho = 0.1$ و فیلتر با سایز 9×9

تصویر با نویز $\rho = 0.2$ و فیلتر با سایز 3×3

تصویر با نویز $\rho = 0.2$ و فیلتر با سایز 7×7

تصویر با نویز $\rho = 0.2$ و فیلتر با سایز 5×5

تصویر با نویز $\rho = 0.2$ و فیلتر با سایز 9×9

	3×3	5×5	7×7	9×9
$\rho = 0.05$	28.4963	59.1198	90.2961	120.0391
$\rho = 0.1$	39.4810	65.6223	100.5938	129.4713
$\rho = 0.2$	94.4900	93.6139	135.3949	185.8383

نتایج بدست آمده

همانطور که مشاهده میشود برای نویز $\rho = 0.2$ پنجره 5×5 بهتر از پنجره 3×3 عمل می کند.

3.2.2 نتیجه اعمال فیلتر گاوسین بر روی تصویر با نویز گاوسین و نویز نمک و فلفل

ابتدا نویز را به تصاویر اضافه میکنیم.

تصویر با نویز گاوسین و $\sigma = 0.01$

تصویر با نویز گاوسین و $\sigma = 0.05$

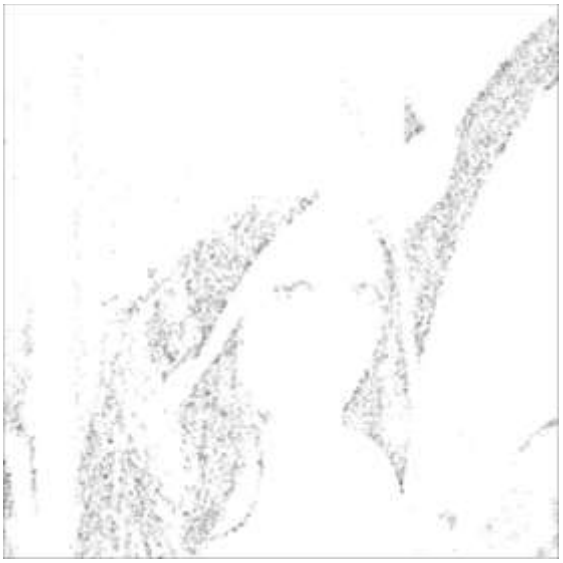
تصویر با نویز گاوسین و $\sigma = 0.1$

حال فیلترهای میانه و میانگین با سایز های مختلف را به تصاویر اعمال میکنیم:

تصویر با نویز $\sigma = 0.01$ و فیلتر میانگین با سایز 3×3

تصویر با نویز $\sigma = 0.01$ و فیلتر میانگین با سایز 7×7

تصویر با نویز $\sigma = 0.01$ و فیلتر میانگین با سایز 5×5



تصویر با نویز $\sigma = 0.01$ و فیلتر میانگین با سایز 9×9

تصویر با نویز $\sigma = 0.05$ و فیلتر میانگین با سایز 3×3

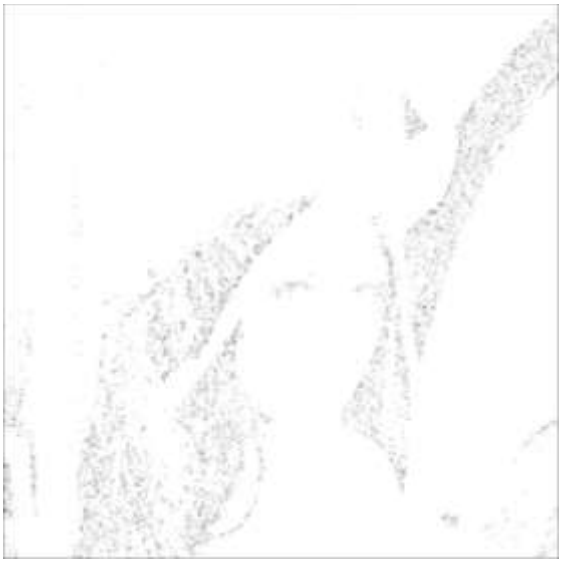
تصویر با نویز $\sigma = 0.05$ و فیلتر میانگین با سایز 7×7

تصویر با نویز $\sigma = 0.05$ و فیلتر میانگین با سایز 5×5

تصویر با نویز $\sigma = 0.05$ و فیلتر میانگین با سایز 9×9

تصویر با نویز $\sigma = 0.1$ و فیلتر میانگین با سایز 3×3

تصویر با نویز $\sigma = 0.1$ و فیلتر میانگین با سایز 7×7



تصویر با نویز $\sigma = 0.1$ و فیلتر میانگین با سایز 5×5

تصویر با نویز $\sigma = 0.1$ و فیلتر میانگین با سایز 9×9

همانطور که مشاهد میشود،این فیلتر عملکرد خوبی بر روی این نویز ندارد.همچنین با افزایش سایز فیلتر عملکرد این فیلتر مطابق انتظار بدتر میشود.

تصویر با نویز $\sigma = 0.01$ و فیلتر میانه با سایز 3×3

تصویر با نویز $\sigma = 0.01$ و فیلتر میانه با سایز 7×7

تصویر با نویز $\sigma = 0.01$ و فیلتر میانه با سایز 5×5

تصویر با نویز $\sigma = 0.01$ و فیلتر میانه با سایز 9×9

تصویر با نویز $\sigma = 0.05$ و فیلتر میانه با سایز 3×3

تصویر با نویز $\sigma = 0.05$ و فیلتر میانه با سایز 7×7

تصویر با نویز $\sigma = 0.05$ و فیلتر میانه با سایز 5×5

تصویر با نویز $\sigma = 0.05$ و فیلتر میانه با سایز 9×9

تصویر با نویز $\sigma = 0.1$ و فیلتر میانه با سایز 3×3

تصویر با نویز $\sigma = 0.1$ و فیلتر میانه با سایز 7×7

تصویر با نویز $\sigma = 0.1$ و فیلتر میانه با سایز 5×5

تصویر با نویز $\sigma = 0.1$ و فیلتر میانه با سایز 9×9

	3×3	5×5	7×7	9×9
$\sigma = 0.01$	150.2643	117.7488	134.3739	161.6070
$\sigma = 0.05$	301.9585	265.4756	279.3113	302.7030
$\sigma = 0.1$	780.6088	738.6642	747.7729	768.0689

نتایج بدست آمده فیلتر میانه

نتایج بدست آمده ضعیف تر از نتایج اعمال این فیلتر بر روی نویز فلفل و نمک میباشد که مطابق انتظار است.

	3×3	5×5	7×7	9×9
$\sigma = 0.01$	12758	16488	17826	18437
$\sigma = 0.05$	14448	17869	18911	19334
$\sigma = 0.1$	16359	19115	19735	19955

نتایج بدست آمده فیلتر میانگین

نتایج بدست آمده بشدت بد می باشد که قابل پیش بینی می بود.

3.3.1 نتیجه بهبود کیفیت تصویر

تصویر اصلی

تصویر نهایی

3.4.1 نتیجه فیلترهای شناسایی لبه عمودی

تصویر با فیلتر f_1

تصویر با فیلتر f_2

فیلتر ها را به صورت زیر تعریف کرده ایم.

$$\boldsymbol{f_1} = \frac{\mathbf{1}}{\mathbf{2}}\begin{bmatrix}\mathbf{1} & \mathbf{0} & \mathbf{-1}\end{bmatrix}$$

$$\boldsymbol{f_2} = \frac{\mathbf{1}}{\mathbf{6}}\begin{bmatrix}\mathbf{1} & \mathbf{0} & \mathbf{-1}\\ \mathbf{1} & \mathbf{0} & \mathbf{-1}\\ \mathbf{1} & \mathbf{0} & \mathbf{-1}\end{bmatrix}$$

$$\boldsymbol{f_3} = \frac{\mathbf{1}}{\mathbf{8}}\begin{bmatrix}\mathbf{1} & \mathbf{0} & \mathbf{-1}\\ \mathbf{2} & \mathbf{0} & \mathbf{-2}\\ \mathbf{1} & \mathbf{0} & \mathbf{-1}\end{bmatrix}$$

تصویر با فیلتر f_3

تصویر با فیلتر f_y.

3.4.2 نتیجه فیلترهای Robert

دو فیلتر به فرم زیر تعریف کرده ایم:

$$f_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$f_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

دقت شود که نتیجه فیلتر f_y با بقیه فیلتر ها متفاوت است و دلیل آن این است که این فیلتر لبه های افقی را پیدا میکند در حالی که بقیه فیلتر ها لبه های عمودی را پیدا میکنند.

تصویر با فیلتر f_x.

3.5.1 ماسکینگ غیرشارپ

از مقادیر `alpha = [0.5,0.8,1]` استفاده کرده ایم.

پنجره گاوسی با سایز 3 و آلفا 0.5

پنجره گاوسی با سایز 5 و آلفا 0.5

پنجره گاوسی با سایز 3 و آلفا 0.8

پنجره گاوسی با سایز 7 و آلفا 0.5

پنجره گاوسی با سایز 5 و آلفا 0.8

پنجره گاوسی با سایز 9 و آلفا 0.5

پنجره گاوسی با سایز 7 و آلفا 0.8

پنجره گاوسی با سایز 9 و آلفا 0.8

پنجره گاوسی با سایز 7 و آلفا 1

پنجره گاوسی با سایز 3 و آلفا 1

پنجره گاوسی با سایز 9 و آلفا 1

پنجره گاوسی با سایز 5 و آلفا 1

همانطور که مشخص است با افزایش مقدار آلفا تاثیر تصویر smooth شده بر نتیجه نهایی بیشتر میشود.

4-کدها

3.1.3 کد تکرار فیلتر میانگین

```
image =
imread('Homeworks/Images/3
/Lena.bmp');
image = rgb2gray(image);


box_filter = [1 1 1; 1 1
1; 1 1 1];
box_filter =
uint8(box_filter);
imwrite(image,'im0.png');
output =
filtering(image,box_filter
);
for i=1:100

imwrite(output,"im"+i+".pn
g");
    output =
filtering(output,box_filte
r);
end
imwrite(output,"im101.png"
);

function
output=filtering(image,fil
ter)
    [R,C] = size(image);
    image =
padarray(image,[1,1]);
    output =
zeros(R,C,'uint8');
    for i=1:R
        for j=1:C
            part =
image(i:i+2,j:j+2);
            mult =
part.*filter;
            out =
sum(mult,'all')/9;
            output(i,j) =
out;
        end
    end
end
```

3.2.1 کد اعمال فیلتر میانه بر روی تصویر با نویز نمک و فلفل

```
image =
imread('Homeworks/Images/3
/Lena.bmp');
image = rgb2gray(image);

image1 =
imnoise(image,'salt &
pepper',0.05);
image2 =
imnoise(image,'salt &
pepper',0.1);
image3 =
imnoise(image,'salt &
pepper',0.2);

imgs{1} = image1;
imgs{2} = image2;
imgs{3} = image3;
filters = [3,5,7,9];

immse(image,median_filter(
image3,9))
for i=1:3
    for j=1:4
        img = imgs(i);
        img = img{1};
        f = filters(j);
        x =
median_filter(img,f);
        name =
"noise"+i+"_"+"filter"+j+"
.png";
        imwrite(x,name);
        mses(i,j) =
immse(image,x);
    end
end
```

```
for i=1:3
    for j=1:4
        img = imgs(i);
        img = img{1};
        f = filters(j);
        x =
box_filter(img,f);
        name =
"noise"+i+"_"+"box_filter"
+j+".png";
        imwrite(x,name);
    end
end
```

```
for i=1:3
    for j=1:4
        img = imgs(i);
        img = img{1};
        f = filters(j);
        x =
median_filter(img,f);
        name =
"noise"+i+"_"+"median_filt
er"+j+".png";
        imwrite(x,name);
    end
end
```

```
function
output=box_filter(image,fi
lter)
    [R,C] = size(image);
    image =
padarray(image,[1,1]);
    output =
zeros(R,C,'uint8');
    for i=1:R
        for j=1:C
            part =
image(i:i+2,j:j+2);
            mult =
part.*filter;
            out =
sum(mult,'all')/9;
            output(i,j) =
out;
        end
```

```
function
output=median_filter(image
,length)
    [R,C] = size(image);
    image =
padarray(image,[floor(leng
th/2),floor(length/2)]);
    output =
zeros(R,C,'uint8');
    for i=1:R
        for j=1:C
            part =
image(i:i+length-
1,j:j+length-1);

            out =
median(part,'all');
            output(i,j) =
out;
        end
    end
end
```

3.2.2 کد اعمال فیلتر گاوسین بر روی تصویر با نویز گاوسین و نویز نمک و فلفل

```
image =
imread('Homeworks/Images/3
/Lena.bmp');
image = rgb2gray(image);

image1 =
imnoise(image,'gaussian',0
.01);
image2 =
imnoise(image,'gaussian',0
.05);
image3 =
imnoise(image,'gaussian',0
.1);

imgs{1} = image1;
imgs{2} = image2;
imgs{3} = image3;
filters = [3,5,7,9];
immse(image,box_filter(ima
ge3,9))
```

```
    end
end

function
output=median_filter(image
,length)
    [R,C] = size(image);
    image =
padarray(image,[floor(leng
th/2),floor(length/2)]);
    output =
zeros(R,C,'uint8');
    for i=1:R
        for j=1:C
            part =
image(i:i+length-
1,j:j+length-1);

            out =
median(part,'all');
            output(i,j) =
out;
        end
    end
end
```

3.3.1 کد بهبود کیفیت تصویر

```
image =
imread('HW3/own.jpg');
image = rgb2gray(image);

image_smooth =
imgaussfilt(image,5);
x = image+ 2*image_smooth;
x = median_filter(x,3);

imwrite(image,"image.png")
;
imwrite(x,"out.png");

function output=
median_filter(image,length
)
    [R,C] = size(image);
    image =
padarray(image,[floor(leng
th/2),floor(length/2)]);
    output =
zeros(R,C,'uint8');
    for i=1:R
        for j=1:C
            part =
image(i:i+length-
1,j:j+length-1);

            out =
median(part,'all');
            output(i,j) =
out;
        end
    end
end
```

3.4.1 کد فیلترهای شناسایی لبه عمودی

```
image =
imread('Homeworks/Images/3
/Lena.bmp');
image =
double(rgb2gray(image));

filter1 = [1 0 -1]/2;
filter2 = [1 0 -1;1 0 -1;1
0 -1]/6;
filter3 = [1 0 -1;2 0 -2;1
0 -1]/8;
filter4 = [1 0 0;0 -1 0;0
0 0 ];
filter5 = [0 1 0;-1 0 0;0
0 0];


im1 =
filtering(image,filter1);
im2 =
filtering(image,filter2);
im3 =
filtering(image,filter3);

imwrite(im1,'f1.png');
imwrite(im2,'f2.png');
```

```
    [R,C] = size(image);
    [x,y] = size(filter);
    image =
padarray(image,[1,1]);
    output =
zeros(R,C,'double');
    for i=1:R
        for j=1:C
            part =
image(i:i+x-1,j:j+y-1);
            mult =
part.*filter;
            out =
sum(mult,'all');
            output(i,j) =
out;
        end
    end
end
```

3.5.1 ماسکینگ غیرشارپ

```
image =
imread('Homeworks/Images/3
/Lena.bmp');
image = rgb2gray(image);

filters = [3,5,7,9];

image_smooth =
imgaussfilt(image,filters(
3));
alpha = [0.5 0.8 1];
for i=1:4
    f = filters(i);
    for j =1:3
        x = alpha(j);
        new_image = image
+ x*(image_smooth -
image);

imwrite(new_image,"alpha"+
x+"filter"+f+".png");
    end
end
```

```
imwrite(im3,'f3.png');

function
output=filtering(image,fil
ter)
    [R,C] = size(image);
    [x,y] = size(filter);
    image =
padarray(image,[1,1]);
    output =
zeros(R,C,'double');
    for i=1:R
        for j=1:C
            part =
image(i:i+x-1,j:j+y-1);
            mult =
part.*filter;
            out =
sum(mult,'all');
            output(i,j) =
out;
        end
    end
end
```

3.4.2 کد فیلترهای Robert

```
image =
imread('Homeworks/Images/3
/Lena.bmp');
image =
double(rgb2gray(image));

filter1 = [1 0;0 -1];
filter2 = [0 1;-1 0];

x1 =
filtering(image,filter1);
x2 =
filtering(image,filter2);


imwrite(x1,'x1.png');
imwrite(x2,'x2.png');

function
output=filtering(image,fil
ter)
```