

بسم تعالی



آزمایشگاه پردازش سیگنال ها و تصاویر پزشکی

گزارش آزمایش ۵ : حذف نویز تصاویر پزشکی

امیرحسین زاهدی ۹۹۱۰۱۷۰۵

آرشام لولوهری ۹۹۱۰۲۱۵۶

پاییز ۱۴۰۲

مسئله: بهترین نسخه تصویر از روی نسخه عموماً آلوده است،
 چگونه آن را بهبود بخشیم؟

$$f = \arg \min_f \|g - f * h\|^2 \Rightarrow$$

به دنبال f ای هستیم که عبارت بالا را مینیمم کند
 پس نسبت به f مشتق میگیریم و f مرحله بعد یا f_{n+1} تفسیر این عبارت به علاوه f مرحله قبل
 است یعنی:

$$f_{n+1} - f_n = \frac{\partial}{\partial f} \|g - f * h\|^2 \Rightarrow f * h = Df \Rightarrow$$

$$f_{n+1} - f_n = \frac{\partial}{\partial f} \|g - Df\|^2 = -2D^T(g - Df) \Rightarrow$$

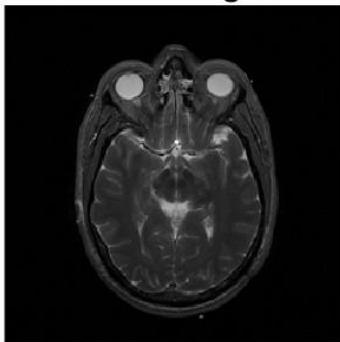
$$f_{n+1} = f_n - 2D^T(g - Df) \Rightarrow f_{n+1} = f_n + \beta D^T(g - Df)$$

سوال ۱:

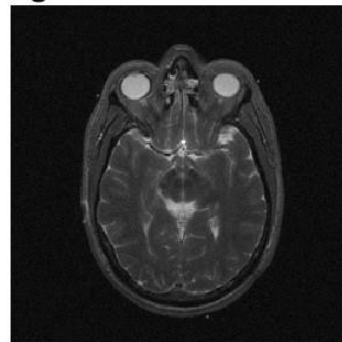
در ابتدا ماتریس ۲۵۶ در ۲۵۶ نویز گوسی با واریانس ۱۵ را تولید می کنیم و به تصویر t_2 اضافه می کنیم. سپس کرنلی با همان ابعاد تولید می کنیم که مربعی ۴ در ۴ در وسط آن ۱ باشد و بقیه ۰ باشند. (البته برای نرمالیزه کردن، مقادیر ۱/۱۶ را به جای ۱ در مربع ۴ در ۴ مرکزی می گذاریم). پس از تولید کرنل، با فوریه گرفتن از هر دوی تصویر و کرنل، ضرب کردن آن ها و سپس بازگشت به حوزه زمان، تصویری تار شده بدست می آید.

در بخش بعدی با استفاده از تابع `imgaussfilt` کانولوشن با یک کرنل گوسی با واریانس ۱ انجام شده و نتیجه نشان داده می شود.

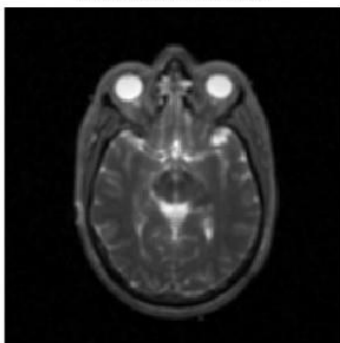
Main image



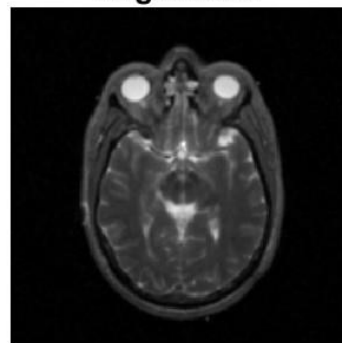
Signal + Gaussian noise



Kernel1 filter



imgaussfilt

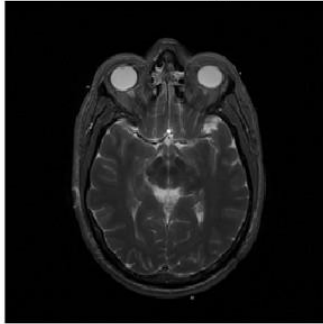


نویز اعمال شده تاثیر کمی داشته و به سختی دیده می شود اما تصویر حاصل از فیلتر کردن کرنل نشان می دهد که همان نویز گوسی اعمال شده نیز دیده نمی شود. البته تصویر تار شده است.

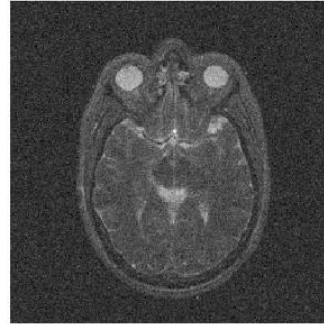
تابع اعمال شده نیز تقریباً نتیجه مشابهی داشته است و نویز را با وجود تار کردن تصویر حذف کرده است.

حال برای اینکه این نویز بیشتر نشان داده شود، واریانس را تا حدود ۳۰۰ بالا می بریم تا تاثیر را مشاهده کنیم.

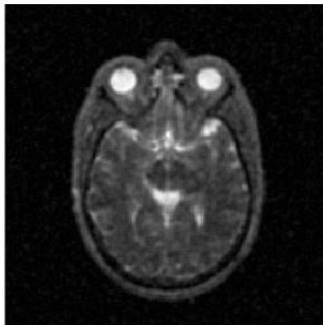
Main image



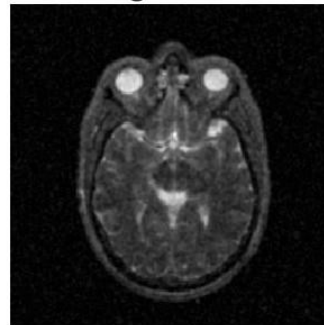
Signal + Gaussian noise



Kernel1 filter



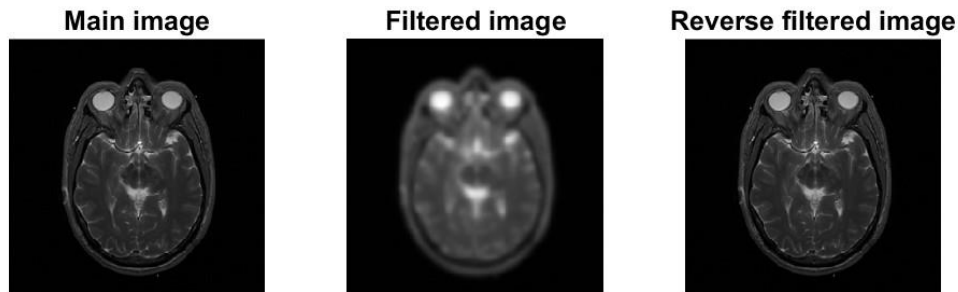
imgaussfilt



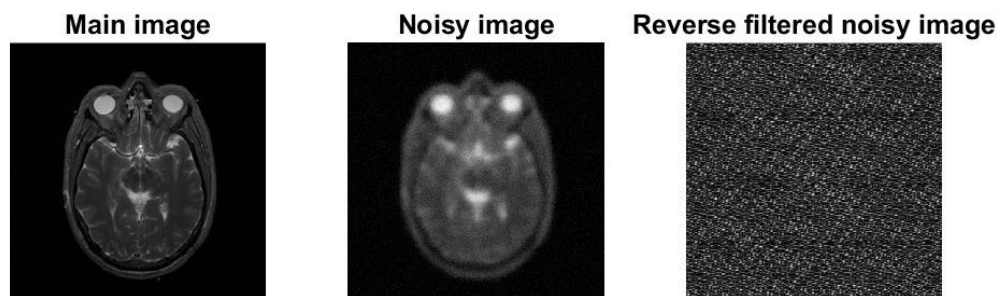
تاثیر نویز این بار به خوبی دیده می شود و همچنین تاثیر فیلترهای اعمال شده در هر دو بخش نیز مشهود تر از قبلا است. هر دو تا حد خوبی توانسته اند رفع نویز را انجام دهند. البته همچنان مشکل تار شدن تصویر وجود دارد. این فیلترهای اعمال شده به نوعی با نرم تر کردن تغییرات ناحیه ای، توانسته اند تغییرات فرکانس بالای ناشی از نویز گوسی را فیلتر و یا نرم کنند.

سوال ۲:

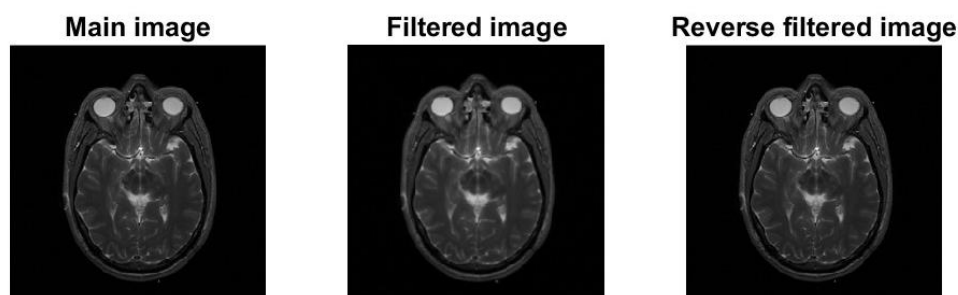
در بخش اول تصویر را با فیلتر گوسی با واریانس ۲.۷ فیلتر می کنیم زیرا که بالاتر از آن خروجی معکوس خوبی نمی دهد. سپس با استفاده از $G = F/H$ سیگنال را مجدد شفاف می کنیم.



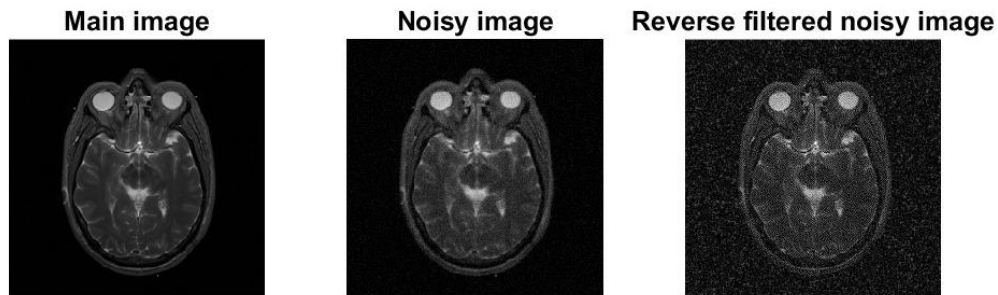
مشاهده می کنیم که پس از تار کردن تصویر، مجدداً توانسته ایم آن را شفاف کنیم که بسیار شبیه تصویر اول است. سپس در ادامه با اضافه کردن نویز گوسی با واریانس ۰.۰۰۱ به سیگنال تار شده، مجدداً $G = F/H$ را بدست می آوریم.



مشاهده می کنیم که نتوانسته است تصویر را بازسازی کند. به همین دلیل واریانس فیلتر استفاده شده در مرحله اول را تا ۰.۶ پایین می آوریم تا بتوانیم بازسازی کنیم.



اینبار تصویر کم تر کدر شده است و بازسازی شفاف شده آن نیز شبیه تصویر اصلی است. حال نویز را همانند قبل به تصویر کدر شده اضافه می کنیم و سپس آن را برمی گردانیم.



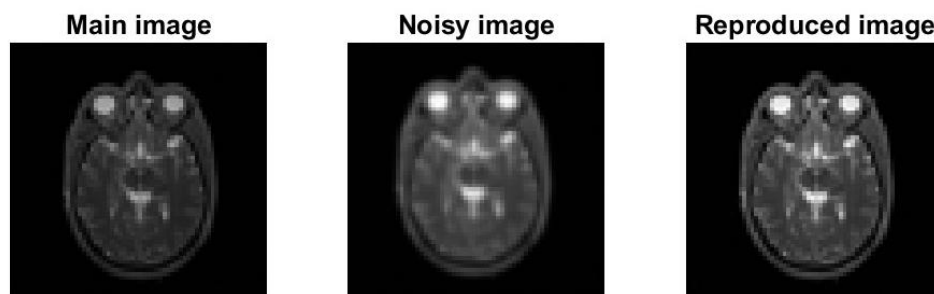
پس از برگرداندن مشاهده می کنیم که تصویر بازسازی شده نویزی است و حتی از تصویر نویزی نیز نویز بیشتری دارد. در قسمت اول تصویر نویزی نداشت و معمولی بود به همین دلیل خوب برگشت و شفاف شد اما تصویر دوم حاوی نویز بود که در برگشت حتی نویزش افزوده شد که نشان از مناسب نبودن این روش در رفع نویز است.

سوال ۳:

ابتدا با استفاده از `imresize` سایز تصویر را از 256×256 به 64×64 تبدیل می کنیم. سپس ماتری 64×64 K را که ۰ است و در گوشه سمت چپ آن h را دارد می سازیم. پس از آن به ازای هر پیکسل تصویر، K را شیف می دهیم و وکتور می کنیم تا هر سطر D مخصوص یک پیکسل باشد.

وقتی که تصویری را در فیلتری کانوالو می کنیم انگار که برای ساخت پیکسل های خروجی، فیلتر را بر روی تصویر ورودی می لغزانیم و در هر جابه جای ضرب و جمع انجام می دهیم. به جای اینکه بیاییم و این ضرب و جمع ها را با لغزش فیلتر چند مرحله تکرار کنیم و کانولوشن انجام دهیم، یک راست ضرایب پیکسل های ورودی برای بدست آوردن خروجی را حساب می کنیم و صرفاً یک بار ضرب انجام می دهیم. ساخت ماتریس D باعث تغییر عمل کانولوشن به سمت ضرب می شود. البته باید حواسمان باشد که در هر مرحله ابعاد را تغییر دهیم و مناسب ضرب و سپس برگشت تصویر به حالت اولیه کنیم.

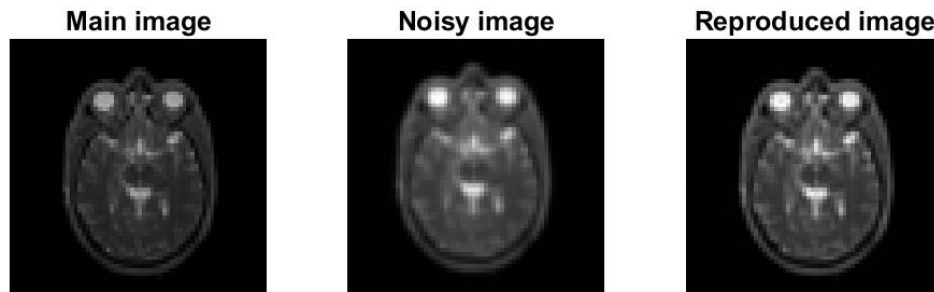
تصویر g را با ضرب D در f می سازیم و نویز را گوسی را به آن اضافه می کنیم. پس از آن تصویر اولیه را با استفاده از ماتریس تعمیم یافته D بازسازی می کنیم و نمایش می دهیم:



مشاهده می کنیم که تصویر محو شده و نویزی به خوبی تولید شده و در نهایت تا حد بسیار مناسبی محو شدگی و نویز از بین رفته است. حتی می توان گفت وضوح تصویر باز تولید شده از تصویر اول بیشتر است!

سوال ۴:

در این سوال با استفاده از g نویزی بدست آمده و الگوریتم Gradient descent همانطور که در برنامه نوشته شده مشاهده می شود، f بدست می آید و نمایش داده می شود.



"Number of repeats = ۴۲ : with thresh = ۱e-۱۰"

با حد مشخص شده، ۴۲ سیکل تکرار شده است تا همگرایی حاصل شود و تصویر بازسازی شده بدست آید.

همانطور که مشاهده می شود، تصویر تا حد خوبی شفاف تر شده است. البته به نظر می آید که روش استفاده شده در بخش قبل نتیجه بهتری داشته است ولی این نتیجه به واقعیت نزدیک تر است و از همه مهمتر، زمان و منابع مورد استفاده برای محاسبه بسیار کمتر از روش قبل است.

سوال ۵:

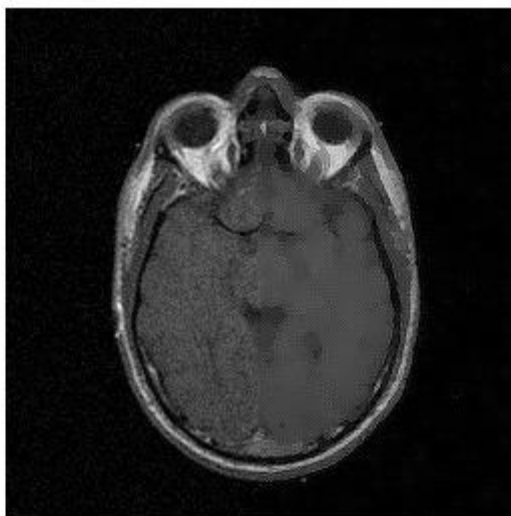
Anisotropic diffusion filtering یک روش پردازش تصویر برای رفع نویز است که در آن اطلاعات مهم تصویر مانند لبه ها نگه داشته می شوند اما اطلاعات دیگر تصویر که چندان مهم نیستند به عبارتی نرم و یکدست می شوند. این فیلتر بر مبنای انتشار کار می کند، به عبارتی با نرم تر کردن تغییرات، ویژگی هایی مانند میزان روشنایی را به همسایه ها انتقال می دهد. البته این انتشار در همه طرف به صورت یکسان پخش نمی شود و ویژگی های مهم تصویر سعی می شود که تا حد ممکن در این انتشار نباشند. اینگونه پس از چندین بار تکرار، ویژگی های مهم تصویر بارز تر می شوند و نویز ها و ... اسموث تر می شوند.

این فیلتر مشتق زمانی روشنایی را بر حسب دیورژانس ضریب انتشار در گرادیان روشنایی به ما می دهد. که ضریب انتشار خود بر حسب مکان و زمان است و ناهمسانی انتشار را تعیین می کند. خود این ضریب انتشار تابعی نمایی است که با تغییرات شدید شدت روشنایی و یا میزان حساسیت نسبت به تغییر می تواند کم شود تا این انتشار کند تر صورت بگیرد. به صورت کلی این سیکل می تواند چند بار ادامه یابد تا تصویر دلخواه بدست آید.

$$\frac{\partial I}{\partial t} = \text{div}(c(x, y, t) \nabla I) = \nabla c \cdot \nabla I + c(x, y, t) \Delta I \quad c(\|\nabla I\|) = e^{-(\|\nabla I\|/K)^2}$$

کد پنج را اجرا می کنیم و خروجی می گیریم.

time = 30 s



شاهد انجام الگوریتم بر روی نیمه راست تصویر مغز هستیم. همانطور که مشاهده می شود، تغییرات نویزی شکل داخل کورتکس در سمت چپ وجود دارند اما در نیمه راستی این تغییرات بسیار نرم تر و یک دست تر شده اند. همچنین ویژگی ها مانند خطوط مرزی و لبه ها تا حد خوبی باقی مانده اند.

روند اجرای کد نیز به این صورت است که ۳۰ بار الگوریتم را بر روی تصویر پیاده میکند و خروجی در نهایت تصویر نرم بالا برای نیم کره راست می شود.