



دانشگاه صنعتی شریف

دانشکده مهندسی برق

پردازش تصاویر پزشکی

تمرین سری دوم

دانشجو

سید ابوالفضل مرتضوی

۴۰۲۲۰۰۱۹۱

فروردین ۱۴۰۳

فهرست مطالب

بخش تئوری.....	۱
۱-۱- سوال اول.....	۱
۲-۱- سوال دوم.....	۳
۳-۱- سوال سوم.....	۴
بخش عملی.....	۴
۱-۲- سوال اول.....	۴
۲-۲- سوال دوم.....	۷
۱-۲-۲- الف.....	۷
۲-۲-۲- ب.....	۷
۳-۲-۲- ج.....	۸
۳-۲- سوال سوم.....	۹
۱-۳-۲- فایل Q3.ipynb.....	۹
۴-۲- سوال چهارم.....	۱۸
۱-۴-۲- الف.....	۱۸
۲-۴-۲- ب.....	۱۸
۳-۴-۲- ج.....	۱۹
۵-۲- سوال پنجم.....	۱۹

فهرست اشکال

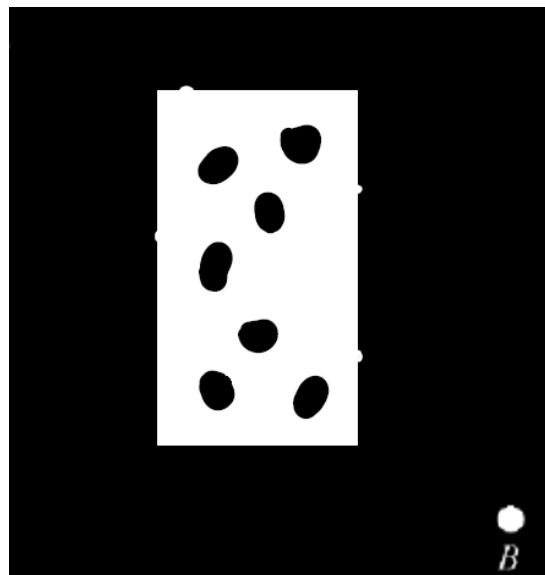
شکل ۱-۱	erosion عمل خروجی	۱
شکل ۱-۲	dilation عمل خروجی	۲
شکل ۱-۳	dilation عمل خروجی	۲
شکل ۱-۴	erosion عمل خروجی	۳
شکل ۲-۱	ناحیه‌ی مربوط به لباس فرد	۴
شکل ۲-۲	closing عمل خروجی	۵
شکل ۲-۳	DFS الگوریتم	۵
شکل ۲-۴	خروجی ماسک نهایی	۶
شکل ۲-۵	خروجی نهایی	۶
شکل ۲-۶	خروجی تصویر نویزی و grayscale شده	۷
شکل ۲-۷	تصویر فیلتر شده با فیلتر classical regression filtering و نویزی	۸
شکل ۲-۸	تصویر فیلتر شده با فیلتر bilateral و تصویر نویزی	۹
شکل ۲-۹	تصویر اصلی، تصویر با نویز نمک و فلفل و تصویر با نویز گوسی	۱۰
شکل ۲-۱۰	تصویر حذف نویز شده با استفاده از فیلتر NLM با پارامترهای $W=3, W_{sim}=3$	۱۰
شکل ۲-۱۱	تصویر حذف نویز شده با استفاده از فیلتر NLM با پارامترهای $W=5, W_{sim}=3$	۱۱
شکل ۲-۱۲	نمودار PSNR برای تصویر با نویز نمک و فلفل	۱۱
شکل ۲-۱۳	نمودار PSNR برای تصویر با نویز گوسی	۱۲
شکل ۲-۱۴	تصویر حذف نویز شده با استفاده از فیلتر NLM با پارامترهای $W=3, W_{sim}=3$	۱۳
شکل ۲-۱۵	تصویر حذف نویز شده با استفاده از فیلتر NLM با پارامترهای $W=5, W_{sim}=3$	۱۳
شکل ۲-۱۶	نمودار PSNR برای تصویر با نویز نمک و فلفل	۱۴
شکل ۲-۱۷	نمودار PSNR برای تصویر با نویز گوسی	۱۴
شکل ۲-۱۸	تصویر حذف نویز شده با فیلتر گوسی	۱۵
شکل ۲-۱۹	نمودار PSNR	۱۵
شکل ۲-۲۰	تصویر حذف نویز شده با فیلتر گوسی	۱۶
شکل ۲-۲۱	نمودارهای PSNR	۱۶
شکل ۲-۲۲	مقایسه معیار PSNR برای فیلترهای مختلف در تصویر ۱	۱۷
شکل ۲-۲۳	مقایسه معیار PSNR برای فیلترهای مختلف در تصویر ۲	۱۷
شکل ۲-۲۴	فانتوم بدون نویز و فانتوم نویزی	۱۸
شکل ۲-۲۵	تصویر بدون نویز، تصویر نویزی و تصویر حذف نویز شده	۱۸
شکل ۲-۲۶	تصویر بدون نویز، تصویر نویزی و تصویر حذف نویز شده	۱۹

شکل ۲۷-۲ تصویر اصلی، تصویر نویزی، تصویر حذف نویز شده با تابع TV_GPCL و تصویر حذف نویز شده با تابع TV_Chambolle و همچنین نمودارهای انرژی و شکاف دوگان مربوط به تصاویر حذف نویز شده ۲۱

بخش تئوری

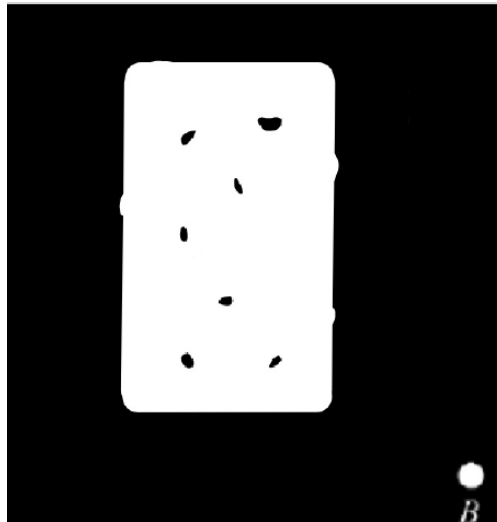
۱-۱- سوال اول

$C = A \ominus B$ در این بخش، در انجام عمل erosion تمامی بخش‌های سفید خارج از مستطیل سفید، چون کوچک‌تر از B هستند حذف می‌شوند، همچنین مستطیل سفید از هر طرف به اندازه‌ی شعاع B کوچک‌تر می‌شود. حفره‌های سیاه داخل مستطیل نیز از هر طرف به اندازه‌ی شعاع B گسترش می‌یابند. شکل تقریبی خروجی در شکل ۱-۱ نشان داده شده است.



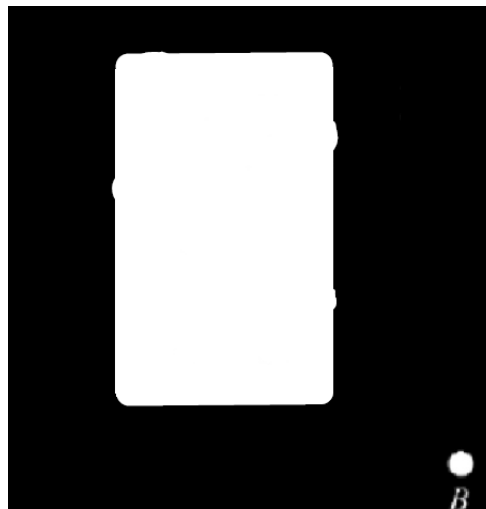
شکل ۱-۱ خروجی عمل erosion

$D = C \oplus B$ با توجه به حذف شدن بخش‌های سفید خارج مستطیل، در انجام عمل dilation مستطیل از هر طرف به اندازه‌ی شعاع B گسترش می‌یابد و همچنین بخش‌های سیاه داخل مستطیل نیز به اندازه‌ی شعاع B کوچک‌تر می‌شوند. خروجی این بخش در شکل ۱-۲ نشان داده شده است.



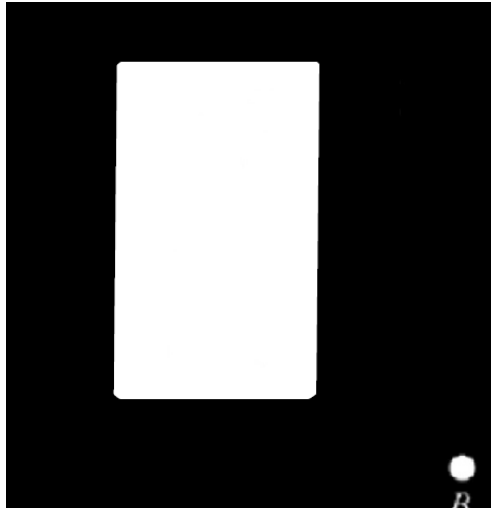
شکل ۱-۲ خروجی عمل *dilation*

$E = D \oplus B$, با انجام دوباره‌ی عمل *dilation* مستطیل دوباره به اندازه‌ی شعاع B از هر طرف گسترش می‌یابد و بخش‌های سیاه داخل مستطیل نیز از بین می‌روند. خروجی این بخش در شکل ۱-۳ نشان داده شده است.



شکل ۱-۳ خروجی عمل *dilation*

$F = E \ominus B$ با انجام این بخش مستطیل به اندازه‌ی شعاع B از هر طرف کوچک‌تر می‌شود.



شکل ۴-۱ خروجی عمل erosion

** تصاویر با موس رسم شده اند و دقت کافی ندارند. لبه‌های تصویر باید تیز باشند.

۱-۲- سوال دوم

۲) برای انجام این مورد باید داریم که:

$$g(x, y) = A e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad r^2 = x^2 + y^2 \rightarrow g(r) = A e^{-\frac{r^2}{2\sigma^2}} \quad (1)$$

همچنین می‌دانیم که:

$$F(e^{-ax^2}) = \sqrt{\frac{\pi}{a}} e^{-\frac{\omega^2}{4a}} = \sqrt{\frac{\pi}{a}} e^{-\frac{4\pi^2 K^2}{4a}} = \sqrt{\frac{\pi}{a}} e^{-\frac{\pi^2 K^2}{a}}$$

$$K^2 = u^2 + v^2 \rightarrow \sqrt{\frac{\pi}{a}} e^{-\frac{\pi^2 (u^2+v^2)}{a}} \quad (2)$$

حال باید به ۱ و ۲ نگاه داشته که:

$$F(A e^{-\frac{r^2}{2\sigma^2}}) = A \sqrt{\pi 2\sigma^2} e^{-\frac{\pi^2 \sigma^2}{2} (u^2+v^2)} \quad (3)$$

۳ می‌توانست به جای تبدیل به ۲ نیز تابع $e^{-\frac{x^2+y^2}{2\sigma^2}}$ را به صورت $e^{-\frac{x^2}{2\sigma^2}}$ و $e^{-\frac{y^2}{2\sigma^2}}$ بنویسیم و راه حل را ادامه دهیم.

۴ باید به ۳ و ۲ ضریب مورد نیاز low pass است چون با افزایش مقادیر u و v صفر می‌شود.

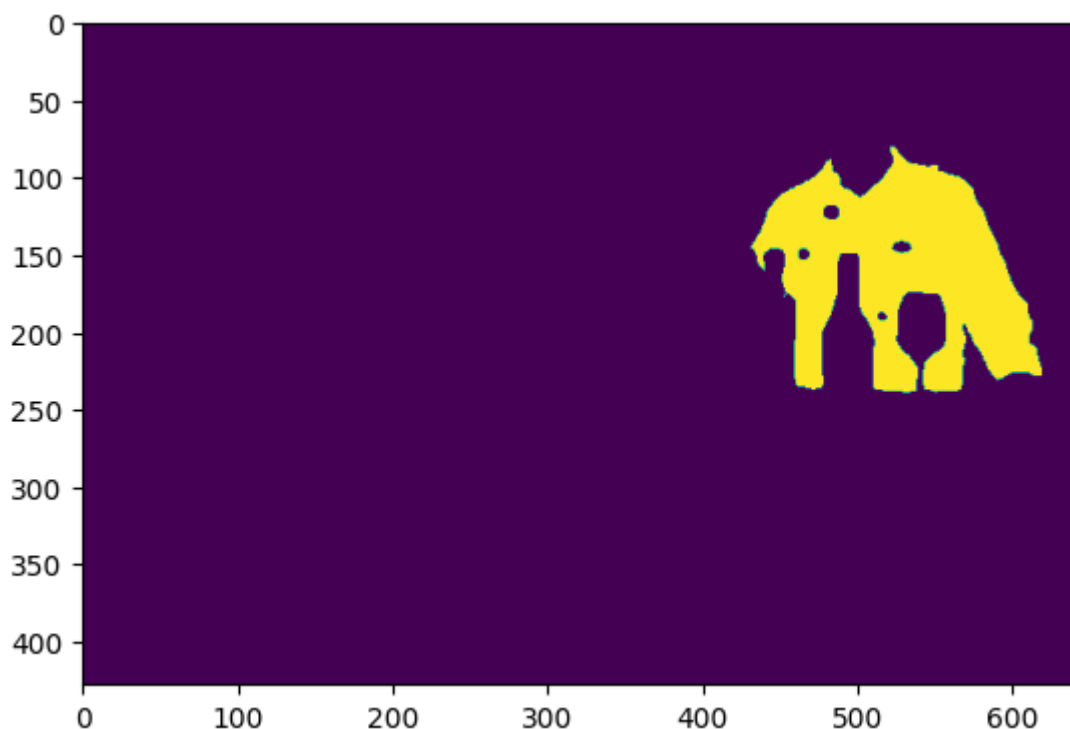
۱-۳- سوال سوم

از آن جایی که در رابطه داده شده صرفاً نویز فلغل نمکی داریم باید اگر پیکسل ۰ یا ۱ نویزی نشده باشد، u همان مقدار v را نگه دارد و یا گرادیان را کمینه کند. بنابراین نرم باید صفر انتخاب شود؛ چراکه فقط زمانی که u, v دقیقاً برابر باشد صفر می‌شود و در غیر این صورت این مقدار صفر نشده و باعث ایجاد ترم regularization می‌شود و به همین دلیل دیگر u مقدار v را دریافت نمی‌کند.

بخش عملی

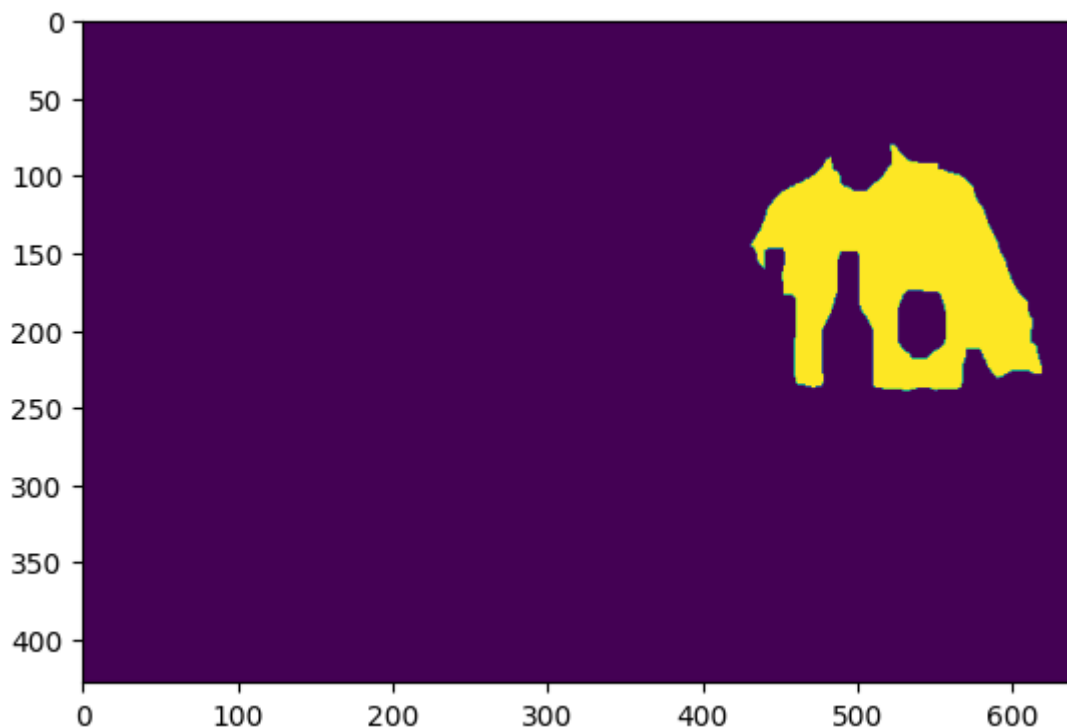
۲-۱- سوال اول

ابتدا پس از بارگذاری تصویر، با توجه به رنگ لباس با آستانه‌گذاری بر روی مقادیر سه کانال RGB ناحیه‌ی مربوط به لباس فرد تشخیص داده شد. خروجی این بخش در شکل ۱-۲ نشان داده شده است.



شکل ۱-۲ ناحیه‌ی مربوط به لباس فرد

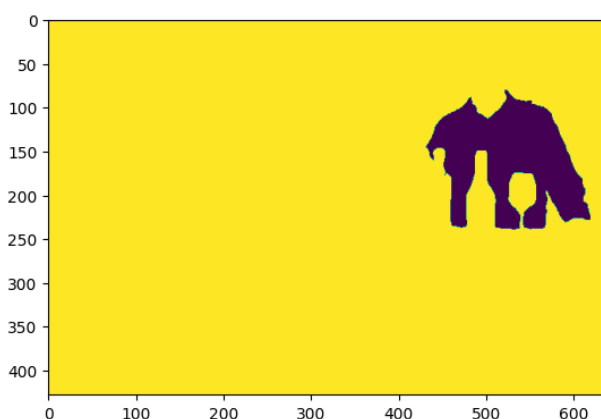
بعد از این مرحله با انجام عمل closing حفره‌های توخالی پر شدند. خروجی این بخش در شکل ۲-۲ نشان داده شده است.



شکل ۲-۲ خروجی عمل *closing*

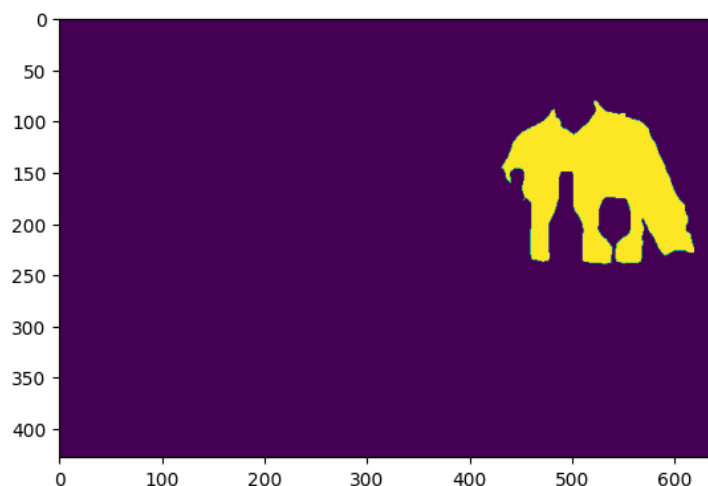
همانطور که دیده می‌شود، بعضی قسمت‌های باریک (مانند پایه‌ی لیوان) در تصویر حذف شده‌اند، برای بازگرداندن این موارد، با استفاده از لینک زیر، الگوریتم DFS بر روی ماسک به دست‌آمده از مرحله‌ی اول پیاده‌سازی شد تا قسمت پس‌زمینه‌ی تصویر بدست آید. خروجی این بخش در شکل ۲-۳ نشان داده شده است.

<https://www.javatpoint.com/dfs-in-python#:~:text=The%20DFS%20is%20an%20algorithm,the%20end%20of%20the%200branch.>



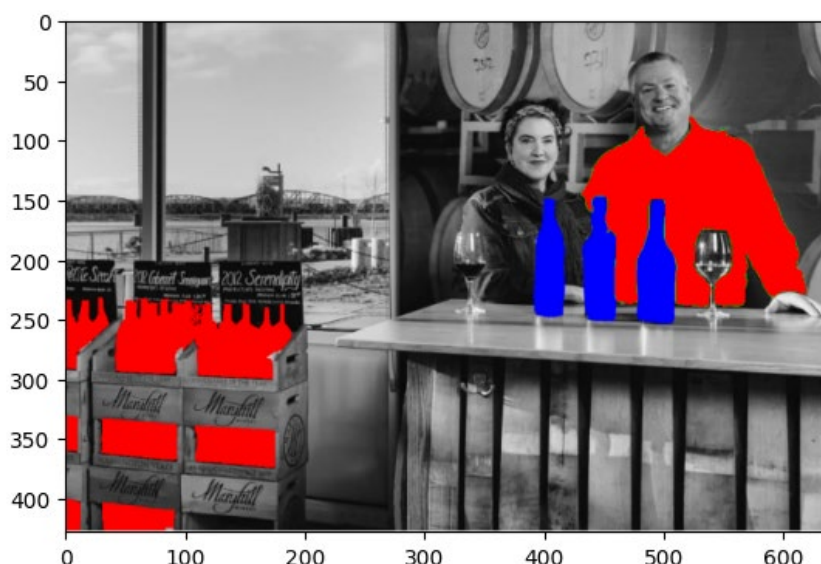
شکل ۲-۳ خروجی الگوریتم *DFS*

بعد از به دست آوردن پس‌زمینه‌ی تصویر، در ماسک شکل ۲-۲ مناطقی که در ماسک شکل ۲-۳ صفر بود را صفر کردیم تا بخش‌های باریک دوباره به دست آیند. خروجی نهایی در شکل ۲-۴ نشان داده شده است.



شکل ۲-۴ خروجی ماسک نهایی

در نهایت با استفاده از ماسک به دست آمده رنگ پیکسل‌های مربوط به لباس به قرمز تبدیل شد و خروجی نهایی به صورت شکل ۲-۵ به دست آمد.



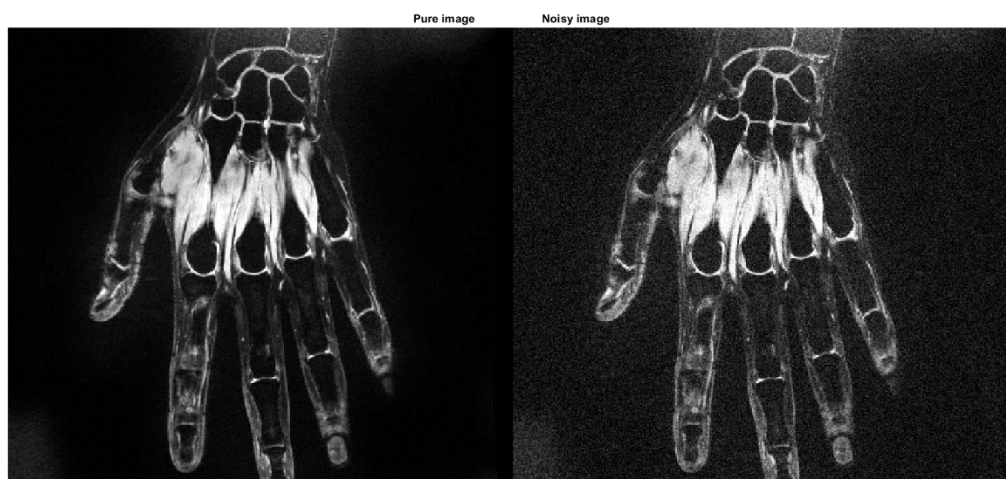
شکل ۲-۵ خروجی نهایی

۲-۲- سوال دوم

※ با توجه به خواسته‌ی سوال مبنی بر استفاده از تابع montage این سوال به جای پایتون با متلب پیاده‌سازی شد.

۲-۲-۱- الف

پس از بارگذاری تصویر و افزودن نویز به آن، خروجی به دست آمده به صورت شد.



شکل ۲-۶ خروجی تصویر نویزی و *grayscale* شده

همچنین SNR تصویر نویزی برابر با ۸.۶۸ دسی‌بل است.

۲-۲-۲- ب

این نوع فیلتر که از دسته‌ی point estimation method است به صورت متوسط‌گیر محلی عمل می‌کند به این صورت که هرچه Y دورتر از X باشد، مقدار وزن و تاثیر کمتری در محاسبه‌ی مقدار پیکسل X دارد. در اینجا ما شاهد linear filtering + normalization هستیم. بعد از پیاده‌سازی الگوریتم این فیلتر خروجی به دست آمده به صورت شکل ۲-۷ است. همچنین مقدار SNR تصویر فیلتر شده برابر با ۱۰.۷۷ دسی‌بل است.



شکل ۷-۲ تصویر فیلتر شده با فیلتر *classical regression filtering* و نویزی

۲-۲-۳ ج

در این روش بر خلاف روش قبل دو ترم گاوسی داریم. یکی از آن ها برای اعمال اثر فاصله و دیگری برای اعمال اثر روشنایی است. برای مثال اگر $g(y)$ نسبت به پیکسل مرکزی اختلافش زیاد بود کرنل ما کاهش پیدا می کند و بالعکس. در واقع با این کار نه اجازه می دهیم پیکسل در فاصله دور اثرگذار باشد و نه پیکسل با اختلاف زیاد در سطح روشنایی. در اینجا ما شاهد $non-linear filtering + computation cost$ هستیم. علت $computation cost$ آن است که برخلاف بخش قبلی که مخرج یکبار محاسبه می شد در این قسمت هر جا که برویم باید دوباره مخرج را محاسبه نماییم.

hx, hg دو پارامتر برای کنترل پنجره های وزنه های ضرب شده به صورت فیلترهای *bilateral* می باشد همانطور که از فرم آن واضح است هرچه مقدار آن بیشتر باشد باعث میشود وزن بیشتری گرفته و فیلتر به اصطلاح از ضربه بودن دور شود. اهمیت hx تاثیر گذاری در فیلتر مکانی دارد و باعث افزایش یا کاهش اثرگذاری فاصله ی دو پیکسل شود. در حالت نهایی به نوعی فیلتر میانگین گیر می شود پس در این مولفه قدرت میانگین گیری را مشخص میکنیم هرچه کمتر باشد میانگین گیری ضعیف تری انجام می شود. در حالتی که واریانس نویز خیلی بالا باشد زیاد کردن مقدار آن می تواند قدرت میانگین گیری را بیشتر کند به عبارت دیگری فرکانس قطع را پایین تر پایین تر بیاورد تا نویز را بهتر حذف کند.

عبارت hg برای بررسی و مقایسه اختلاف شدت می باشد و در بخش فیلتر ساختاری اعمال می شود، که تاثیر آن را چقدر تنظیم کند. هرچه مقدار آن کمتر باشد حساسیت آن به لبه ها بیشتر می شود و در خروجی مقدار کمتری میدهد تا کرنل لبه را حفظ کند. پس بر اساس اهمیت لبه و براساس اختلاف شدت باید مقدار آن را تنظیم کرد. پس براساس اینکه تصویر چقدر نویزی و شدت آن چقدر است که در تشخیص لبه بودن چه تاثیری دارد میتوان مقدار آن را تنظیم کرد، به طور مثال با زیاد کردن آن درواقع داریم مجبور میکنیم به ازای اختلافات شدید تر کرنل مقدار کمی بگیرد که در نهایت باعث شود لبه را حذف نکند در حالی که اگر مقدار

آن را کم کنیم لبه های ضعیف تر نیز در تصویر حفظ می شوند با تنظیم آن به مقدار مناسب میتوان تمایز بین قدرت نویز و لبه ها تعیین کرد. بنابراین به صورت خلاصه hg لبه ها را و hx همواری تصویر را کنترل می کنند. برای نویزدایی از تصویر ما در این بخش مقدار hg برابر با ۰.۸ و مقدار hx برابر با ۱.۲ در نظر گرفته شد. خروجی این بخش در شکل ۸-۲ نشان داده شده است.



شکل ۸-۲ تصویر فیلتر شده با فیلتر *bilateral* و تصویر نویزی

همچنین مقدار SNR به دست آمده برای تصویر فیلتر شده در این بخش برابر با ۱۰.۸۶ است.

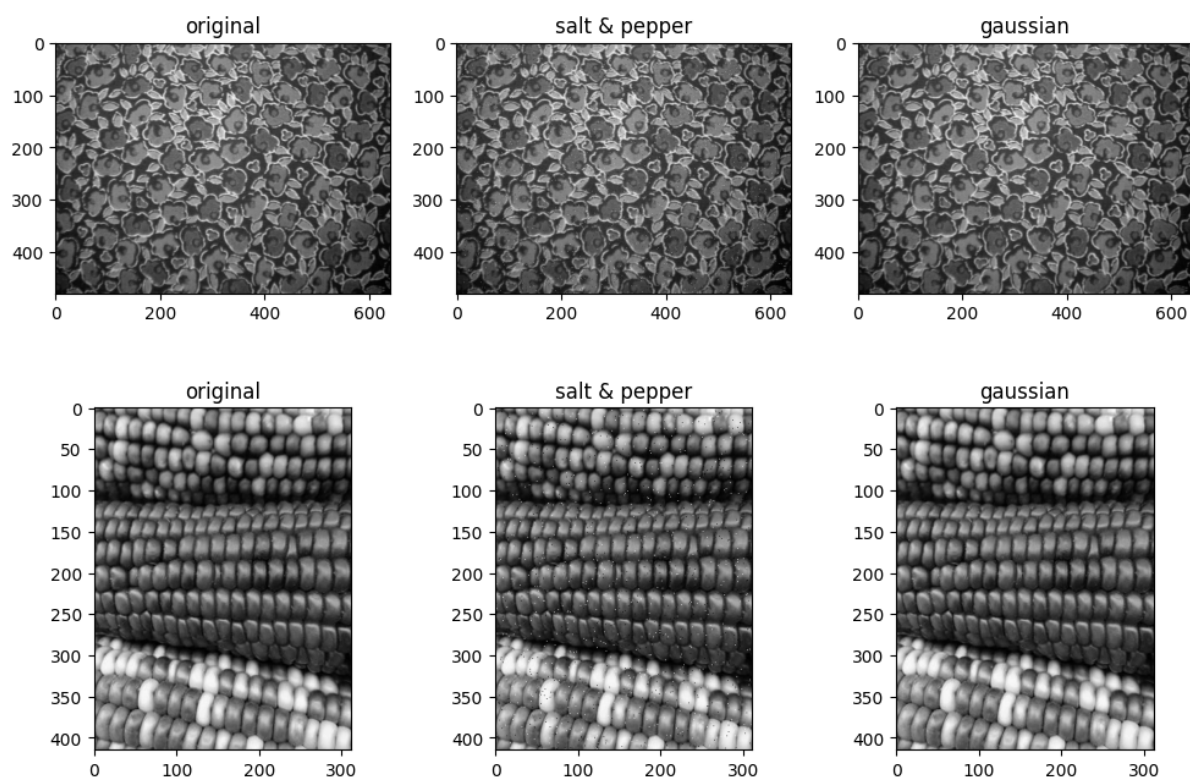
۲-۳- سوال سوم

** در این بخش دو کد پیاده سازی شده است. در Q3.ipynb نویز گوسی با میانگین ۰.۵ و واریانس ۰.۱ به تصویر اضافه شده است. در فایل Q3_2.ipynb نویز گوسی به تصویر نرمالیزه اضافه شده است.

۲-۳-۱- فایل Q3.ipynb

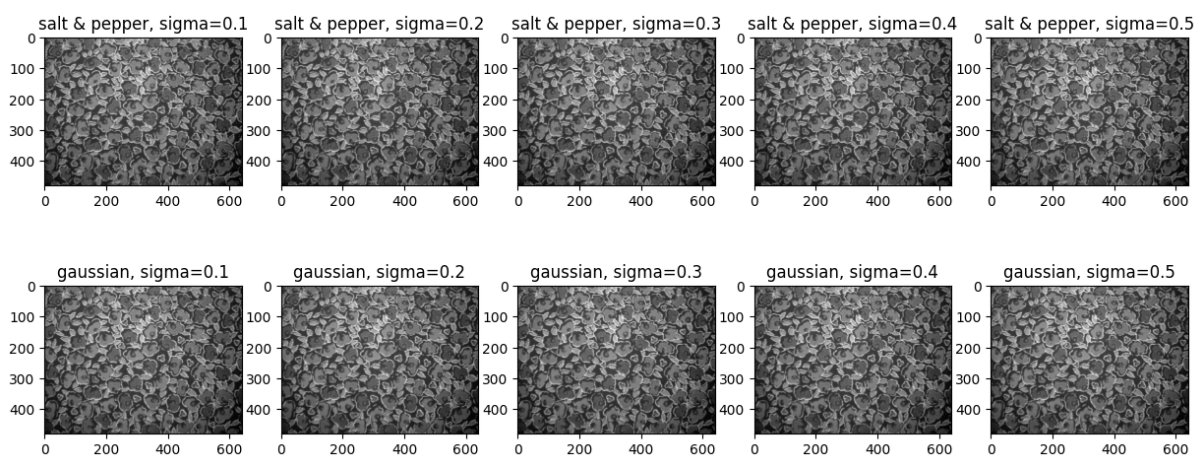
۲-۳-۱-۱- بخش الف

در این بخش ابتدا نویزها به تصاویر اضافه شدند. تصاویر به دست آمده در شکل ۹-۲ نشان داده شده است.

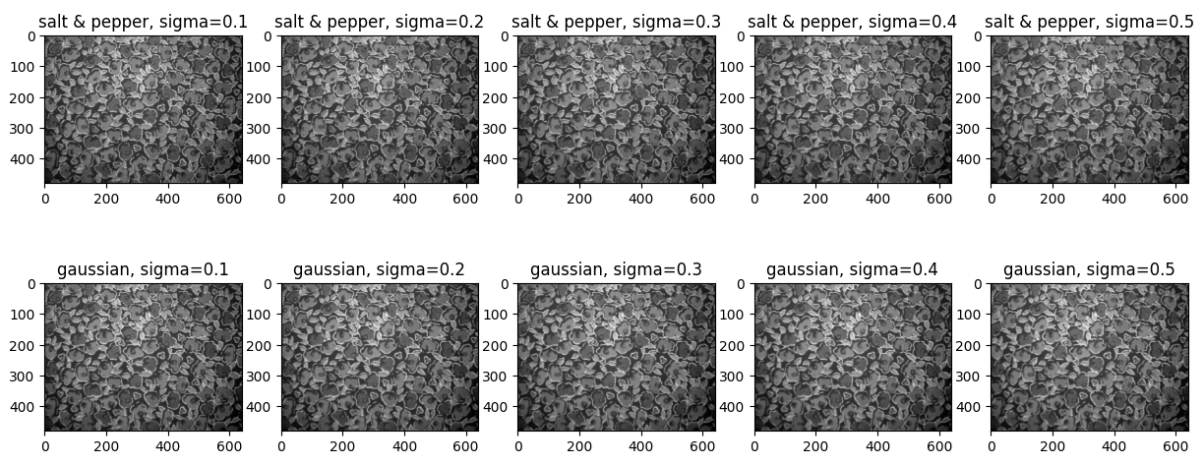


شکل ۹-۲ تصویر اصلی، تصویر با نویز نمک و فلفل و تصویر با نویز گوسی

پس از افزودن نویز به تصویر، الگوریتم فیلتر NLM برای حذف نویز پیاده‌سازی شد. خروجی این بخش به صورت شکل ۱۰-۲ و شکل ۱۱-۲ است.

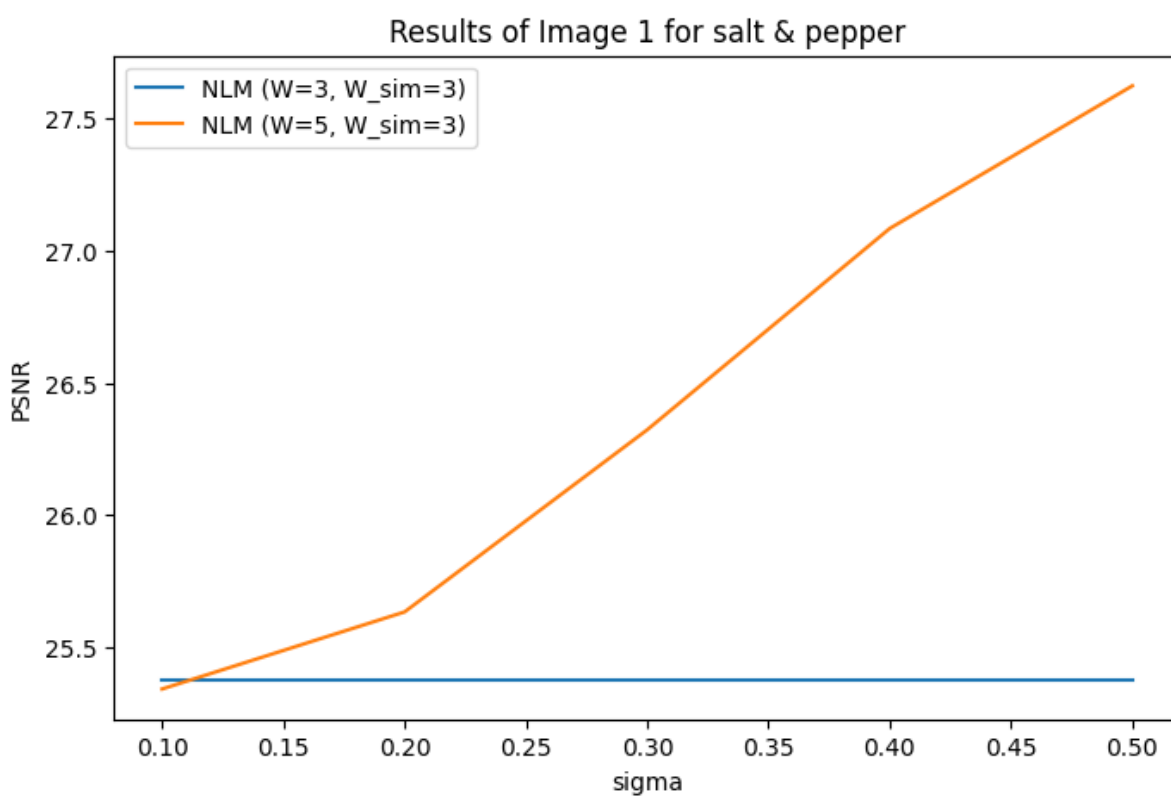


شکل ۱۰-۲ تصویر حذف نویز شده با استفاده از فیلتر NLM با پارامترهای $W=3, W_{sim}=3$

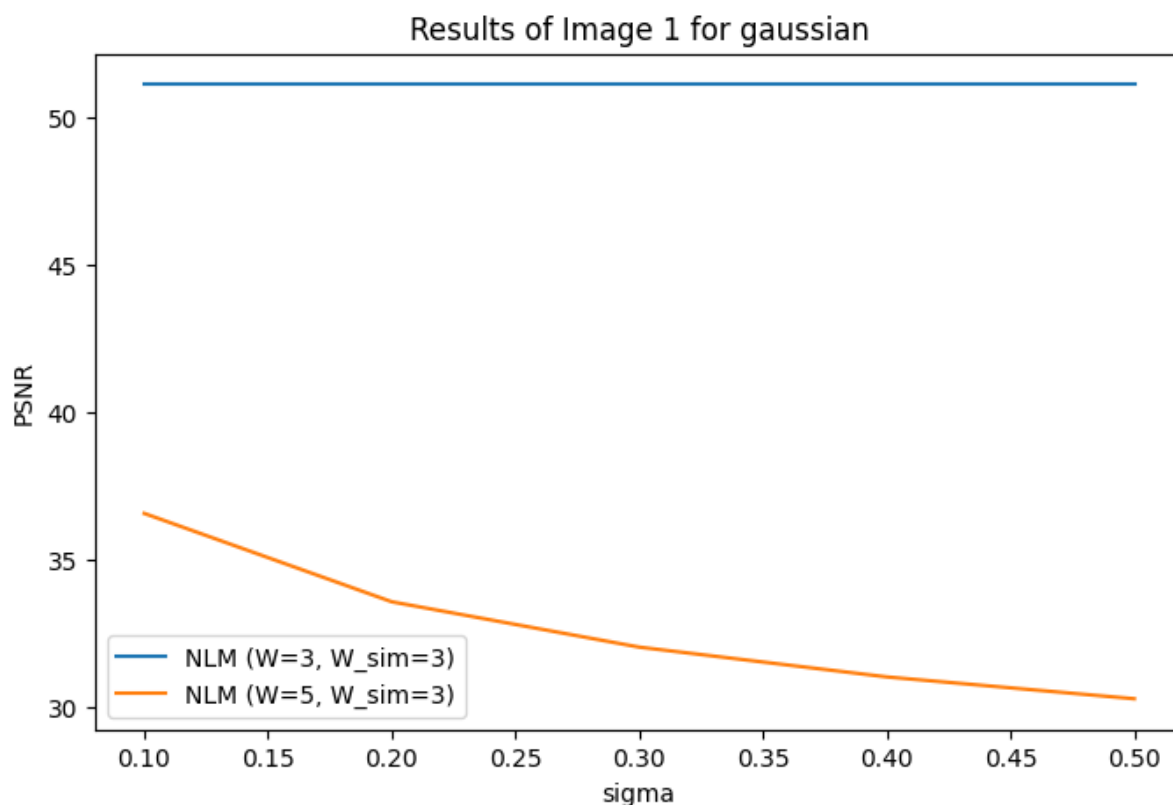


شکل ۲-۱۱ تصویر حذف نویز شده با استفاده از فیلتر NLM با پارامترهای $W=5, W_{sim}=3$

همچنین نمودارهای PSNR تصاویر به صورت شکل ۲-۱۲ و شکل ۲-۱۳ می‌باشند.



شکل ۲-۱۲ نمودار $PSNR$ برای تصویر با نویز نمک و فلفل



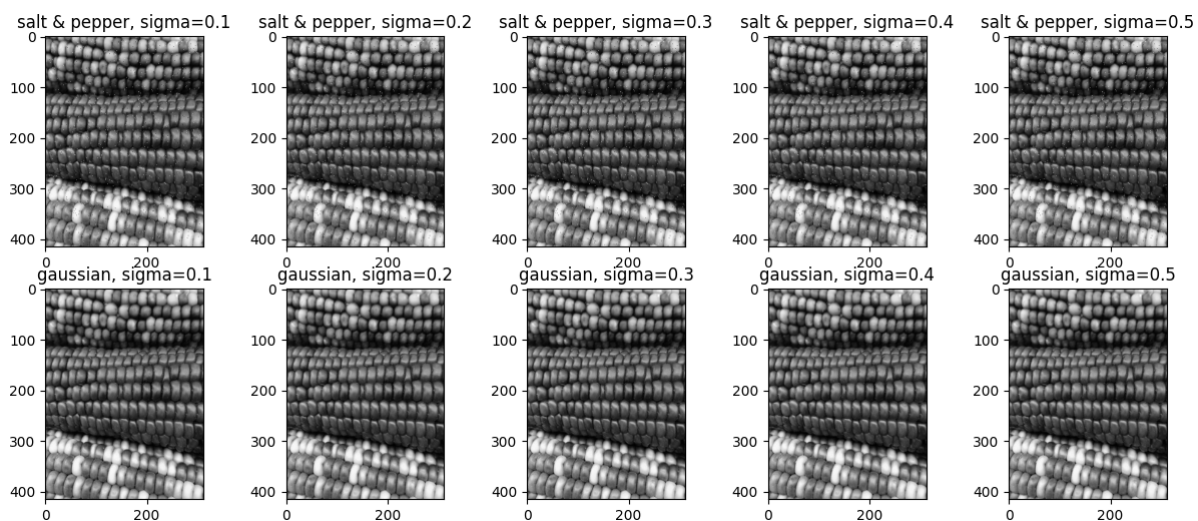
شکل ۱۳-۲ نمودار $PSNR$ برای تصویر با نویز گاوسی

با توجه به نمودارها نکات زیر به دست می‌آید

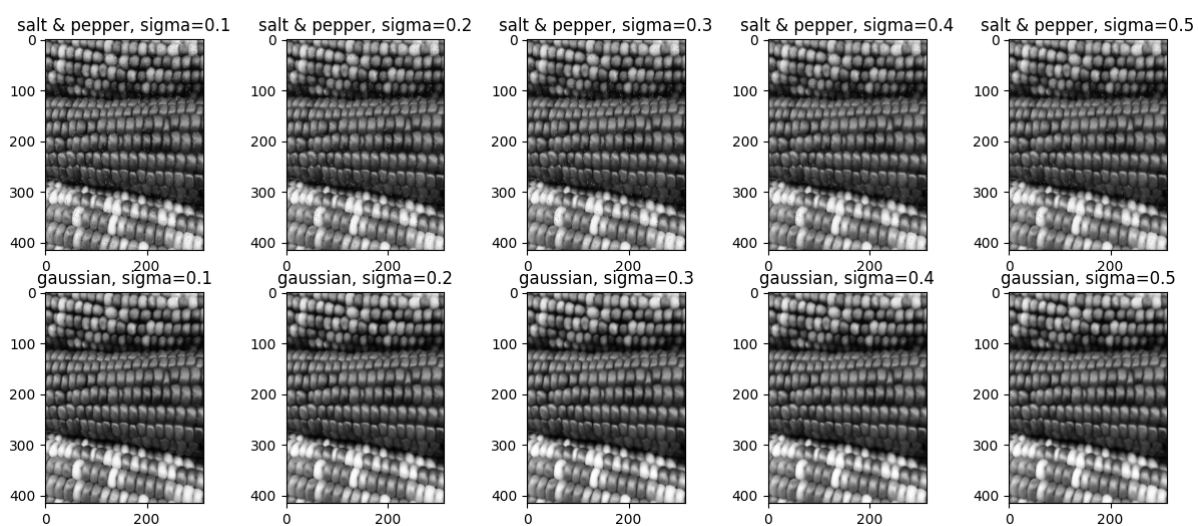
۱. در مقادیر یکسان برای W و W_{sim} این نوع فیلتر مانند فیلتر میان‌گیر عمل می‌کند و با افزایش σ ، مقدار $PSNR$ تغییر نمی‌کند،

۲. در نویز نمک و فلفل افزایش مقدار σ باعث بهبود $PSNR$ می‌شود ولی در مورد نویز گاوسی افزایش این مقدار باعث کاهش $PSNR$ می‌شود.

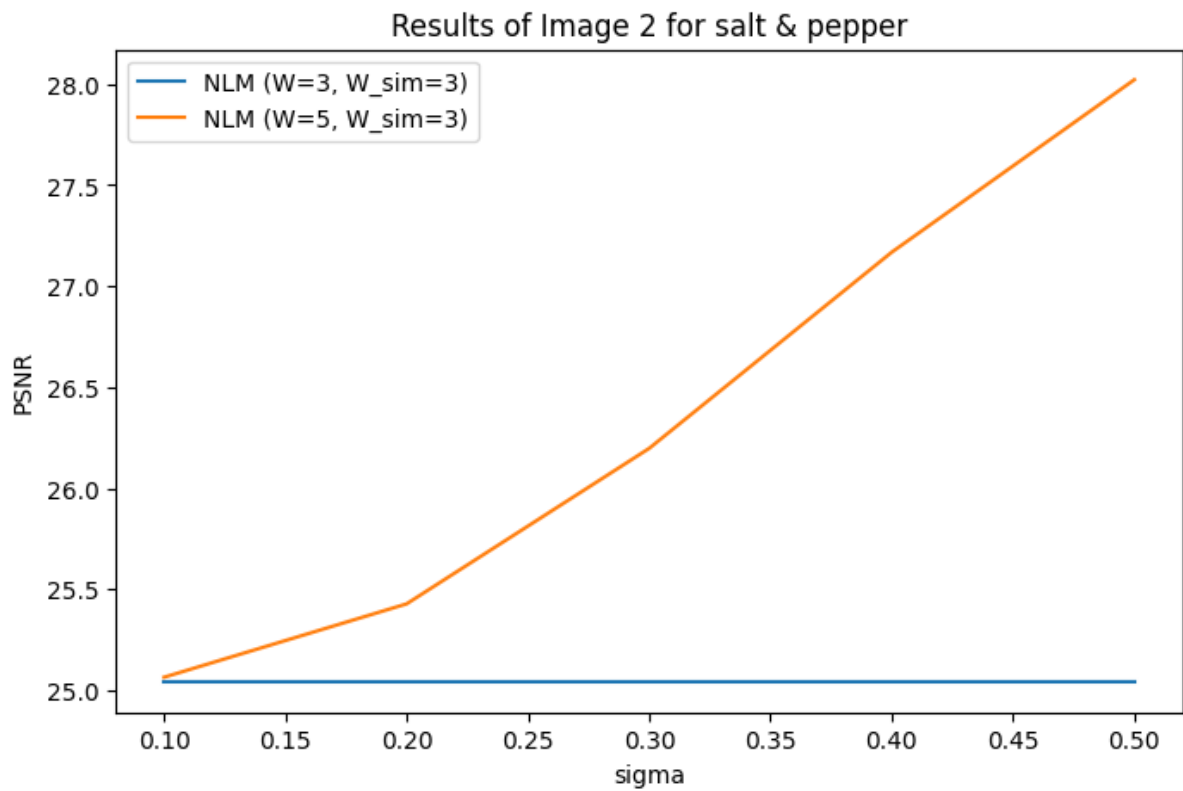
در ادامه روند طی شده برای تصویر ۱ بر روی تصویر ۲ طی می‌شود. خروجی‌های این بخش در شکل ۱۴-۲ تا شکل ۱۷-۲ نشان داده شده‌اند.



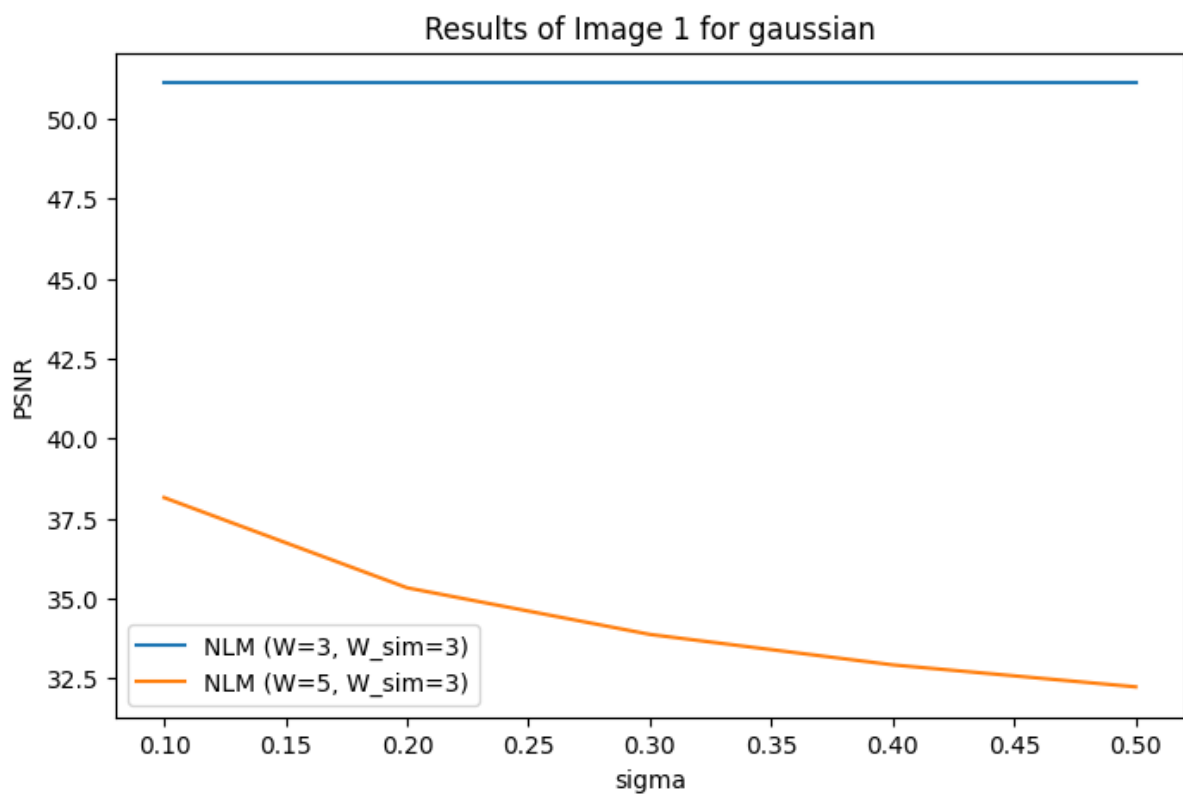
شکل ۱۴-۲ تصویر حذف نویز شده با استفاده از فیلتر NLM با پارامترهای $W=3, W_{sim}=3$



شکل ۱۵-۲ تصویر حذف نویز شده با استفاده از فیلتر NLM با پارامترهای $W=5, W_{sim}=3$



شکل ۱۶-۲ نمودار $PSNR$ برای تصویر با نویز نمک و فلفل

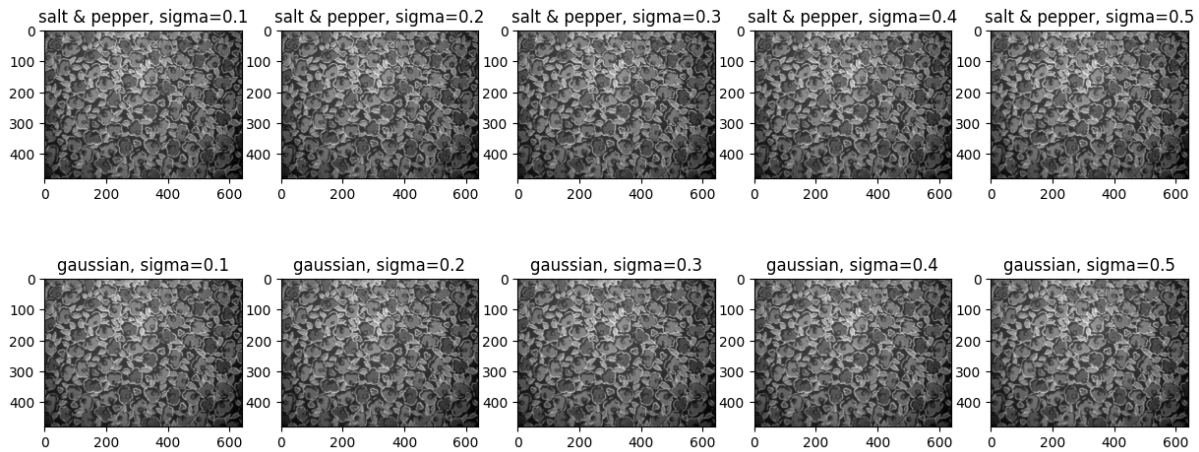


شکل ۱۷-۲ نمودار $PSNR$ برای تصویر با نویز گاوسی

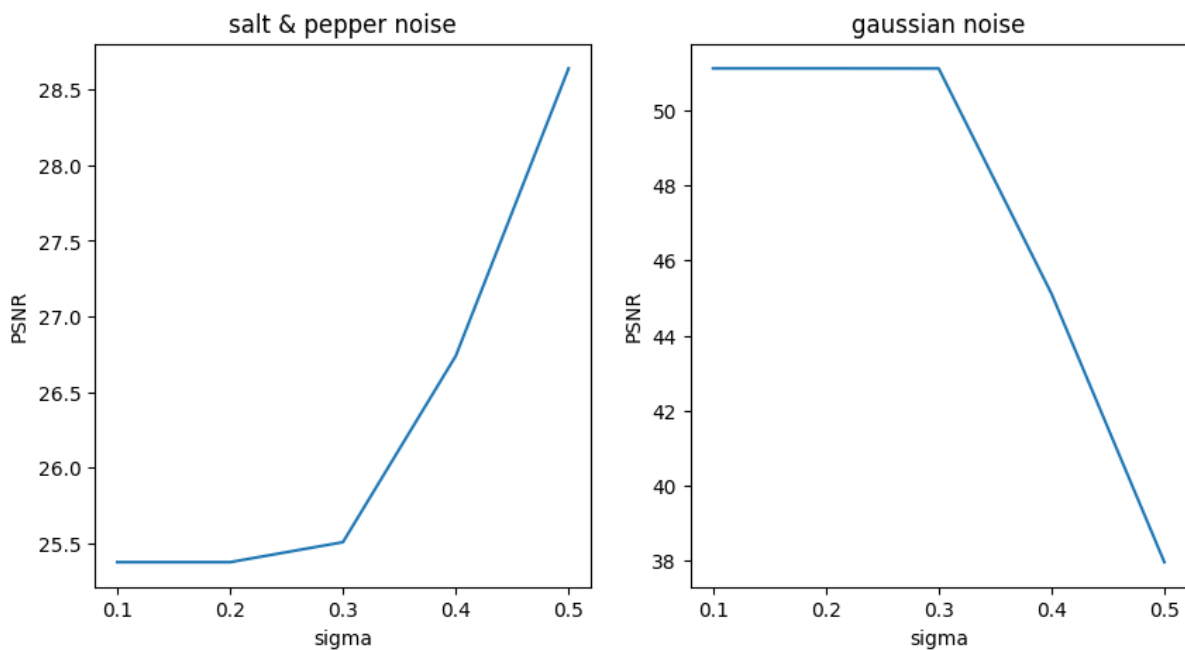
خروجی‌های تصویر ۲ نیز مانند تصویر ۱ می‌باشد.

۲-۱-۳-۲-ب

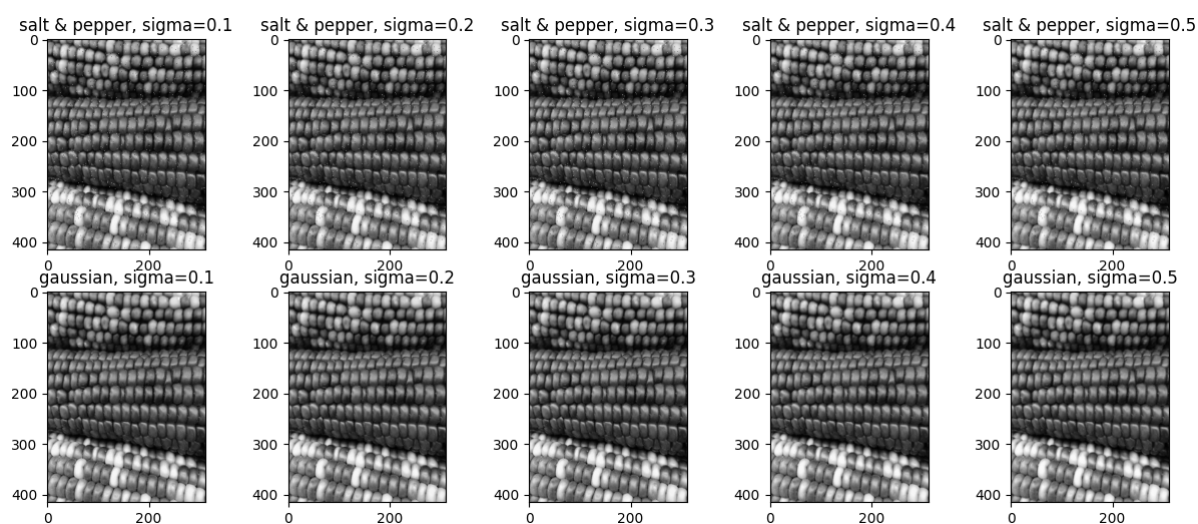
پس از پیاده‌سازی الگوریتم فیلتر گاوسی، تصاویر خروجی این بخش به صورت شکل ۲-۱۸ تا شکل ۲-۲۱ است.



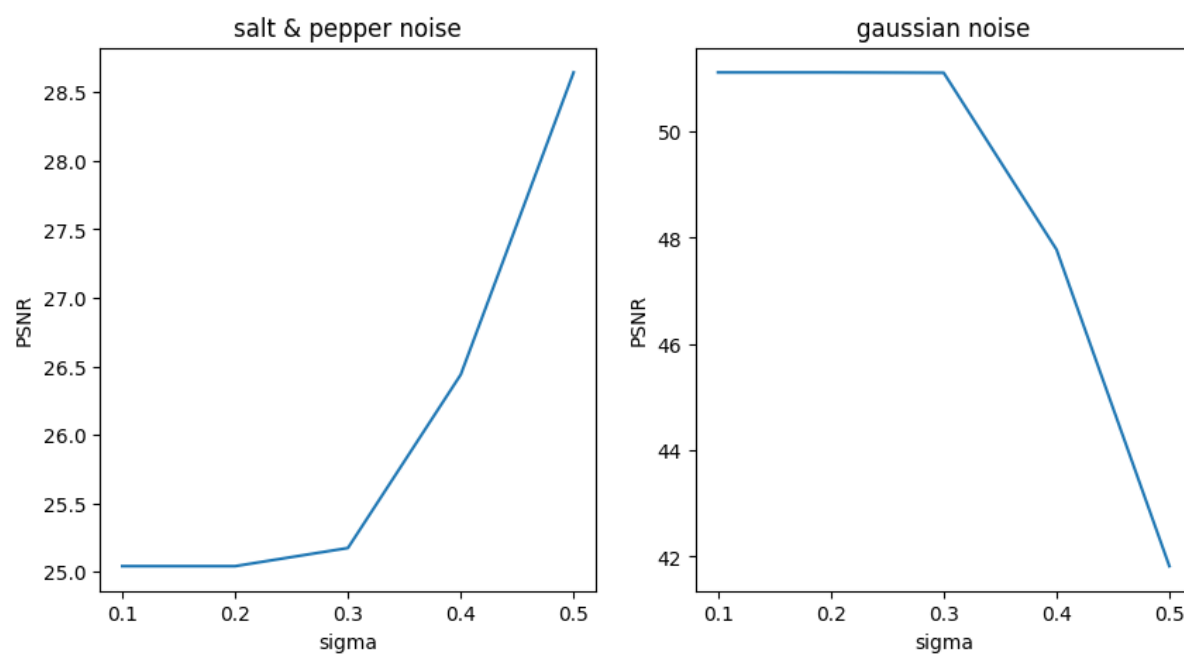
شکل ۲-۱۸ تصویر حذف نویز شده با فیلتر گاوسی



شکل ۲-۱۹ نمودار PSNR

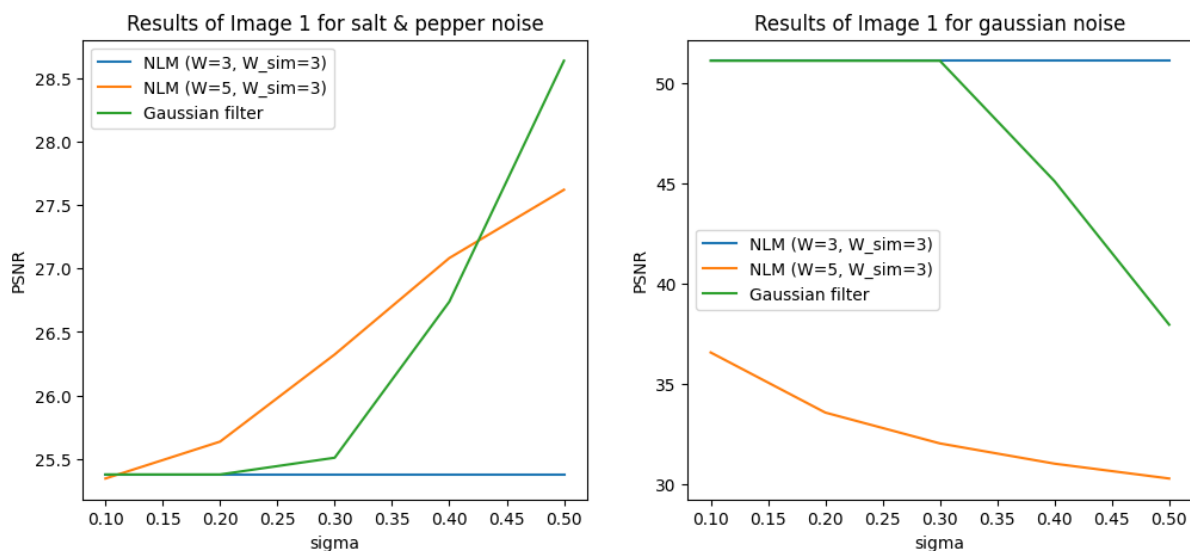


شکل ۲۰-۲ تصویر حذف نویز شده با فیلتر گوسی

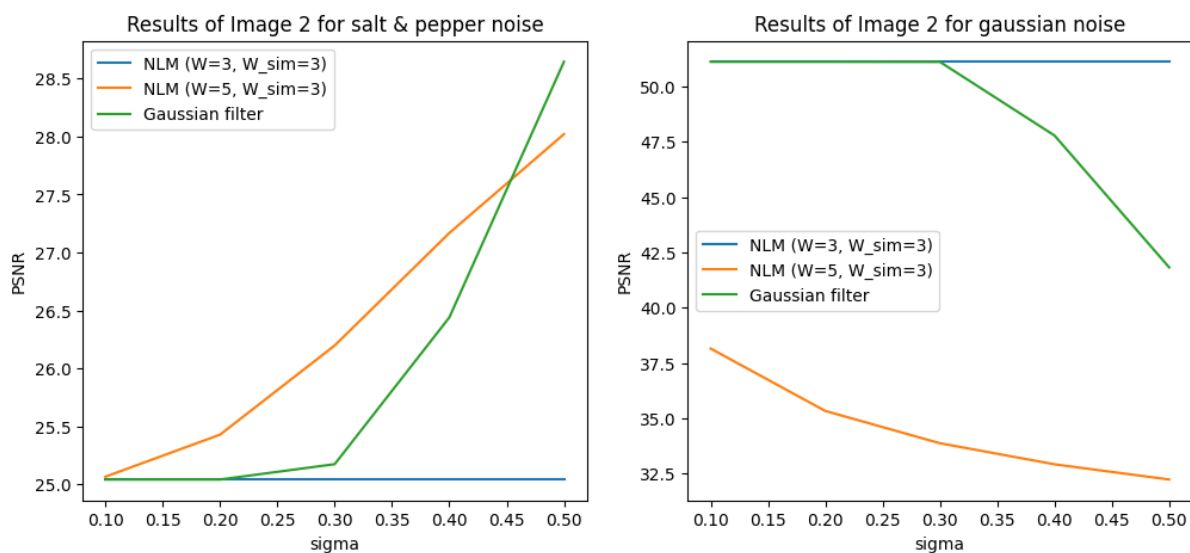


شکل ۲۱-۲ نمودارهای PSNR

در شکل ۲۲-۲ و شکل ۲۳-۲ مقایسه‌ی معیار PSNR برای دو نوع فیلتر، برای دو تصویر نشان داده شده است.



شکل ۲۲-۲ مقایسه معیار $PSNR$ برای فیلترهای مختلف در تصویر ۱



شکل ۲۳-۲ مقایسه معیار $PSNR$ برای فیلترهای مختلف در تصویر ۲

فیلتر NLM یک فیلتر غیرخطی است که از میانگین وزن داده شده‌ی پیکسل‌های همسایگی محلی استفاده می‌کند تا نویز در یک تصویر را کاهش دهد. این فیلتر با مقایسه شباهت بین دو پیکسل در یک تصویر عمل می‌کند و سپس آن‌ها را بر اساس شباهت‌شان میانگین می‌گیرد. فیلتر NLM از مقایسه‌ی مناطق به جای مقایسه پیکسل‌ها استفاده می‌کند. از سوی دیگر، فیلتر گوسی یک فیلتر خطی است که از یک تابع گوسی برای حذف نویز در یک تصویر استفاده می‌کند. این فیلتر با کانالو کردن یک تصویر با یک هسته گوسی، نسخه‌ای بهبود یافته از تصویر اصلی را تولید می‌کند. به طور خلاصه، فیلتر NLM موثرتر از فیلتر گوسی در کاهش نویز در یک تصویر است زیرا از شباهت بین مناطق به جای مقایسه پیکسل‌ها استفاده می‌کند. با این حال، هزینه‌ی محاسباتی بیشتری دارد.

*** برای جلوگیری از شلوغ شدن گزارش خروجی‌های فایل دوم در این جا ارائه نشدند، لطفا به نوت‌بوک مورد نظر مراجعه شود.

۲-۴- سوال چهارم

۲-۴-۱- الف

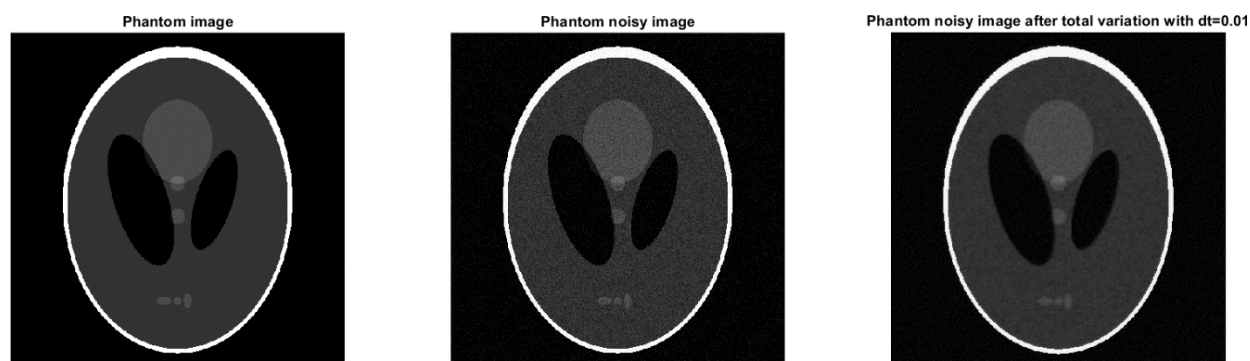
پس از ایجاد فانتوم و افزودن نویز به آن خروجی به صورت شکل ۲-۲۴ به دست آمد.



شکل ۲-۲۴ فانتوم بدون نویز و فانتوم نویزی

۲-۴-۲- ب

با پارامترهای گفته شده، تصویر خروجی به صورت شکل ۲-۲۵ به دست آمد.

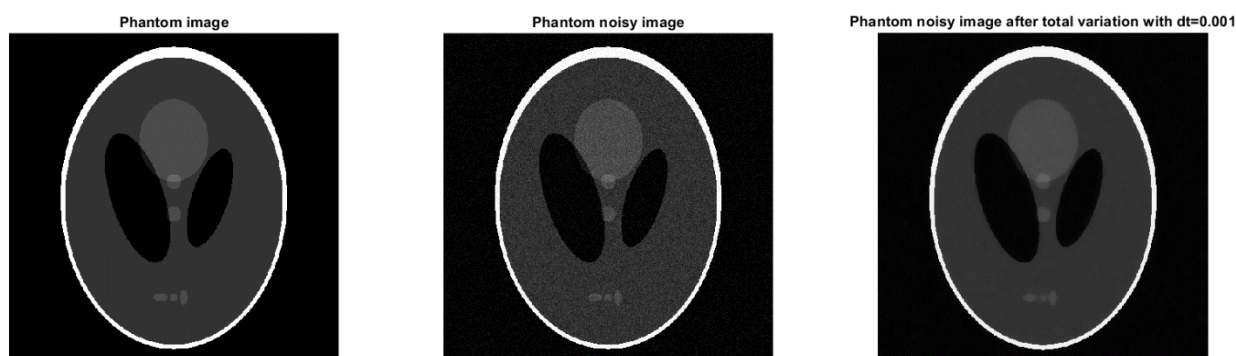


شکل ۲-۲۵ تصویر بدون نویز، تصویر نویزی و تصویر حذف نویز شده

خروجی به دست آمده با مقادیر گفته شده برای پارامترها، خیلی خوب نبود، برای بهبود خروجی از مقادیر زیر برای پارامترها استفاده شد.

$$dt=0.001 \quad \lambda=2$$

خروجی به دست آمده به صورت شکل ۲-۲۶ است.



شکل ۲-۲۶ تصویر بدون نویز، تصویر نویزی و تصویر حذف نویز شده

همانطور که ملاحظه می شود، خروجی به دست آمده بسیار بهتر شده است.

۲-۴-۳- ج

مقادیر SNR به صورت زیر است.

SNR for filtered phantom image: 15.9454

SNR for noisy phantom image: 15.4894

SNR for filtered phantom2 image: 21.12 (this the filtered image with $dt=0.001$ and $\lambda=2$)

با توجه به مقادیر SNR مشخص می شود که کیفیت تصویر با فیلتر دوم بسیار بیشتر بهبود یافته است.

۲-۵- سوال پنجم

کد ارائه شده در توابع TV_GPCL و TV_Chambolle دو الگوریتم متفاوت برای حذف نویز با روش Total Variation را پیاده سازی می کند. در ادامه یک شرح مرحله به مرحله از الگوریتم ها آمده است:

پارامترهای ورودی:

$W1, W2$ متغیرهای دوگان اولیه، f تصویر نویزی، lbd : ثابت پایداری، α طول گام، NIT حداکثر تعداد تکرار، $GapTol$ تلورانس همگرایی، $Verbose$ برای نمایش نتایج در حین انجام فیلترینگ.

کد، متغیرها را مقداردهی اولیه می کند و گرادیان های لازم را براساس داده های ورودی محاسبه می کند. انرژی اولیه سیستم براساس متغیرهای دوگان اولیه، به دست می آید. سپس این تابع متغیر اصلی u را بر اساس انرژی محاسبه شده و سایر متغیرها به دست می آورد. پس از آن کد وارد حلقه ای می شود که تکرارهای اصلی الگوریتم را انجام می دهد. درون حلقه متغیرهای دوگان به روز می شوند و پس از به روزرسانی نرمالیزه می شوند. سپس انرژی سیستم و همچنین متغیر اصلی u براساس متغیرهای دوگان به روز شده محاسبه می شود. برای همگرایی مقدار $dualGap$ نسبت به مقدار تلورانس مشخص شده مقایسه می شود، اگر معیار همگرایی برآورده شود، حلقه ای تکرار متوقف می شود و در غیر این صورت به تکرار بعدی منتقل می شود. پس از پایان حلقه و همگرایی، تابع مقادیر متغیر اصلی u ، متغیرهای دوگان، انرژی، شکاف دوگانی، هزینه زمانی و تعداد کل تکرارها را برمی گرداند.

به طور کلی، الگوریتم $TV_Chmbolle$ با به روزرسانی متغیرهای دوگان بر اساس گرادیان تابع هدف، نرمال سازی متغیرها و محاسبه ی متغیر اصلی و شکاف دوگانی پیش می رود. این مراحل را تا زمانی که معیارهای همگرایی برآورده شود یا تعداد حداکثر تکرارها به پایان برسد ادامه می دهد. نتیجه ای که به وسیله ی حذف نویز با این تابع به دست می آید یک تصویر تمیز شده ی u است. همچنین کد ارائه شده در تابع TV_GPCL روش $gradient\ projection$ را با طول گام ثابت برای حل فرمول بندی دوگان مدل بازیابی تصویر پیاده سازی می کند. تفاوت اصلی بین این دو تابع در روش بهینه سازی مورد استفاده شان برای حل فرمول دوگانه ی مدل بازیابی تصویر (TV) است.

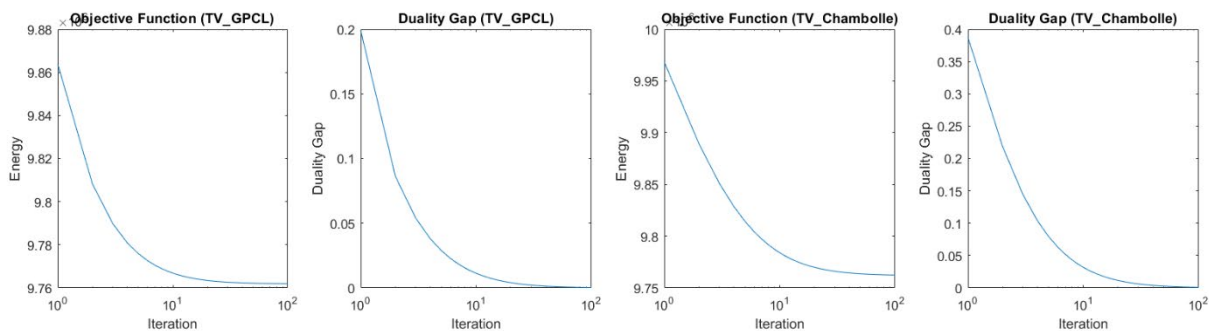
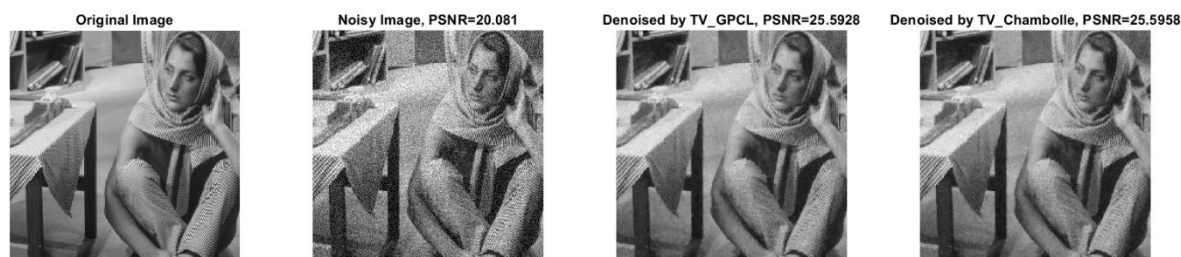
$TV_Chambolle$: این تابع روش $chambolle$ را که در سال ۲۰۰۴ پیشنهاد شد را پیاده سازی می کند. این الگوریتم از طول گام ثابت برای کاهش گرادیان استفاده می کند. این الگوریتم مقدار متغیرهای دوگان $w1, w2$ را به طور مکرر با استفاده از روش کاهش گرادیان نیمه ضمنی $chambolle$ به روز می کند. فرآیند گام به گام شامل محاسبات گرادیان، به روزرسانی متغیرهای دوگان و محاسبه ی انرژی و شکاف دوگان است. همگرایی با بررسی شکاف دوگانگی نسبی در برابر یک تلورانس مشخص تعیین می شود.

TV_GPCL : این تابع روش $gradient\ projection$ را با طول گام ثابت پیاده می کند. مشابه روش قبل در این روش نیز متغیرهای دوگان $w1, w2$ به طور مکرر به روز می شوند. با این تفاوت که به روزرسانی متغیرهای دوگان با استفاده از $gradient\ projection$ بازسازی مدل ROF است. همگرایی نیز در این الگوریتم با مقایسه ی شکاف دوگان با تلورانس مشخص شده، تعیین می شود.

به طور خلاصه، هدف هر دو تابع حل یک فرمول دوگانه مدل بازیابی تصویر است. اما هر تابع از روش متفاوتی برای بهینه سازی استفاده می کنند. $TV_Chambolle$ از گرادیان نزولی نیمه ضمنی و TV_GPCL از روش

اولیهی gradient projection استفاده می‌کند. انتخاب روش بهتر به عواملی مانند خواص همگرایی، کارایی محاسباتی و تسک موردنظر بستگی دارد.

در این بخش، در ابتدا با استفاده از حلقه‌ی for و سنجش خطای MSE مقدار بهینه‌ی پارامترهای α و λ به دست آمد. سپس با استفاده از مقادیر به دست آمده، تصویر مورد نظر حذف نویز شد. خروجی در شکل ۲۷-۲۸ نشان داده شده است.



شکل ۲۷-۲۸ تصویر اصلی، تصویر نویزی، تصویر حذف نویز شده با تابع TV_{GPCL} و تصویر حذف نویز شده با تابع $TV_{Chambolle}$ و همچنین نمودارهای انرژی و شکاف دوگان مربوط به تصاویر حذف نویز شده

همانطور که قابل مشاهده است با توجه به مقدار PSNR در این تصویر تابع $TV_{Chambolle}$ خروجی بهتری داشته است.