



دانشکده مهندسی کامپیوتر

انتقال داده

پاییز ۱۴۰۰

پروژه اول

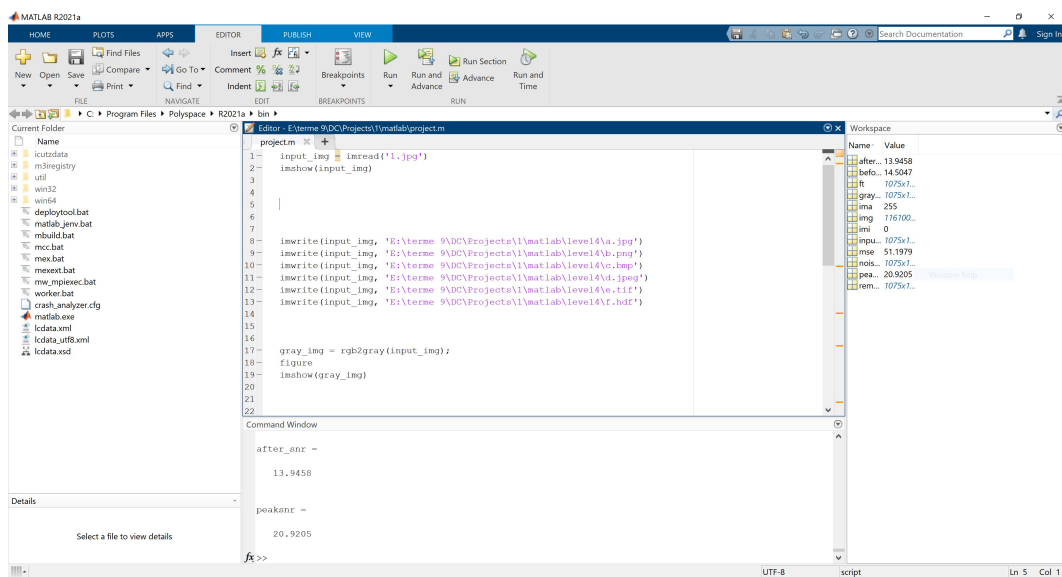
Noise

استاد درس دکتر ابوالفضل دیانت

تدوین ۹۶۴۶۲۱۰۴ - محمد یارمقدم

گام اول

در ابتدا به نصب نرم افزار Matlab پرداختم. برای این کار نسخه R2021a را نصب کردم و از طریق کرک موجود در آن به انجام مراحل فعالسازی پرداختم.



شکل ۱: عکس ورودی

گام دوم

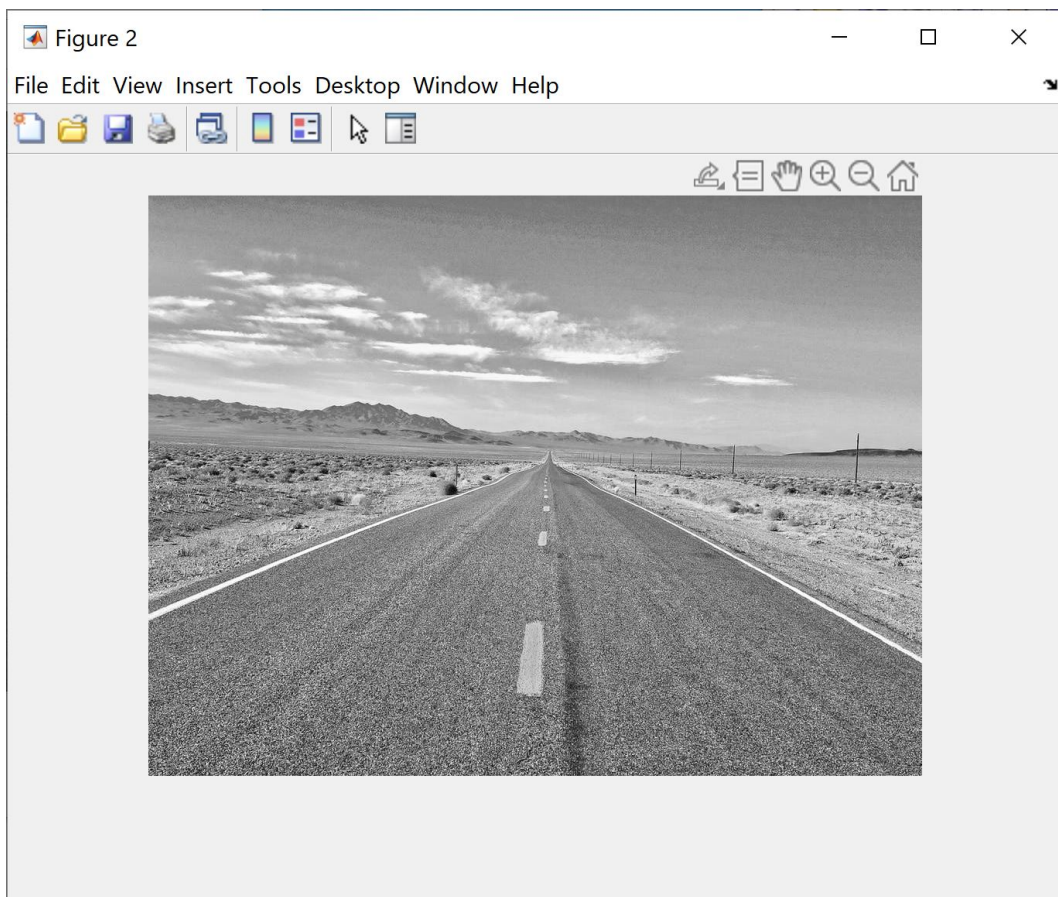
در این مرحله با استفاده از دستور `imread` عکس موردنظر را از کامپیوتر لوکال خوانده و وارد نرم افزار matlab کردم.



شکل ۲: محیط Matlab

گام سوم

در این مرحله فرمت اصلی عکس را که RGB است را به فرمت Gray scale که ۸ بیتی است تبدیل می کنیم. سایز عکس در این حالت کاهش می یابد. در واقع از سایز $1920 * 1080 * 3$ که برای عکس هاس رنگی است به سایز $1920 * 1080$ کاهش می یابد. در واقع ۳ کانال عکس حذف شده و هر پیکسل (که کوچک ترین واحد عکس محسوب می شود) با عددی بین ۰ تا ۲۵۵ نمایش داده می شود.



شکل ۳: تصویر تغییر یافته به فضای Gray Scale

مقایسه انواع تصاویر

۱. تصویر رنگی: تصاویر روزمره و کاربردی در این دسته از عکس ها قرار می گیرند. این تصاویر ۳ کانال دارند. در این حال برای هر یک از ۳ کانال موجود RGB ۸ بیت در نظر گرفته می شود.
۲. تصویر باینری (binary): این تصویر در ساده ترین سطح قرار دارد و بیس محسوب می شود. برای تشکیل این عکس از روش ب binary classification استفاده می شود. به این صورت که برای هر مقدار پیکسل اگر از ترش هولد کمتر باشد پیکسل سیاه خواهد بود و در غیر اینصورت پیکسل سفید خواهد بود. در این نوع هر پیکسل فقط می تواند سیاه یا سفید باشد.
۳. تصویر خاکستری (Gray Scale): این تصویر تک کانال است. در واقع فقط رنگ خاکستری دارد و مقدار هر پیکسل سطوح مختلف از این رنگ را نمایش خواهد داد. هر پیکسل مقداری بین ۰ تا ۲۵۵ دارد. یعنی هر بیت توسط ۸ بیت نمایش داده می شود.
۴. عکس های رنگی را می توان از نظر تعداد بیت نیز دسته بندی کرد. در این حال ۳ دسته ۸، ۱۶ و ۳۲ بیتی ایجاد می شوند.

گام چهارم

در این مرحله تصویر خاکستری شده را با دستور `imwrite` ذخیره می کنیم و با استفاده از دستور `imshow` نمایش می دهیم. برای ذخیره عکس بدون فشرده سازی و به صورت طبق آزمایش و مشاهده انجام شده باید از فرمت PNG استفاده کنیم. هم چنین طبق تحقیق انجام شده فرمت ذخیره سازی به سه حالت Lossy compression, Lossless compression, Uncompressed تقسیم می شود که PNG در دسته آخر قرار می گیرد.

گام پنجم

سیگنال از نوع انرژی است زیرا هیچ توانی در تصویر موجود نیست. برای محاسبه مقدار انرژی عکس از تابع `sum` استفاده می کنیم. با استفاده از این تابع آرایه های پیکسل عکس را جمع می کنیم و انرژی را محاسبه می کنیم. پس با استفاده از تابع `sum` و پاس دادن آرگومان `'all'` به آن انرژی عکس را محاسبه کردیم که این مقدار برابر 111667892 شد.

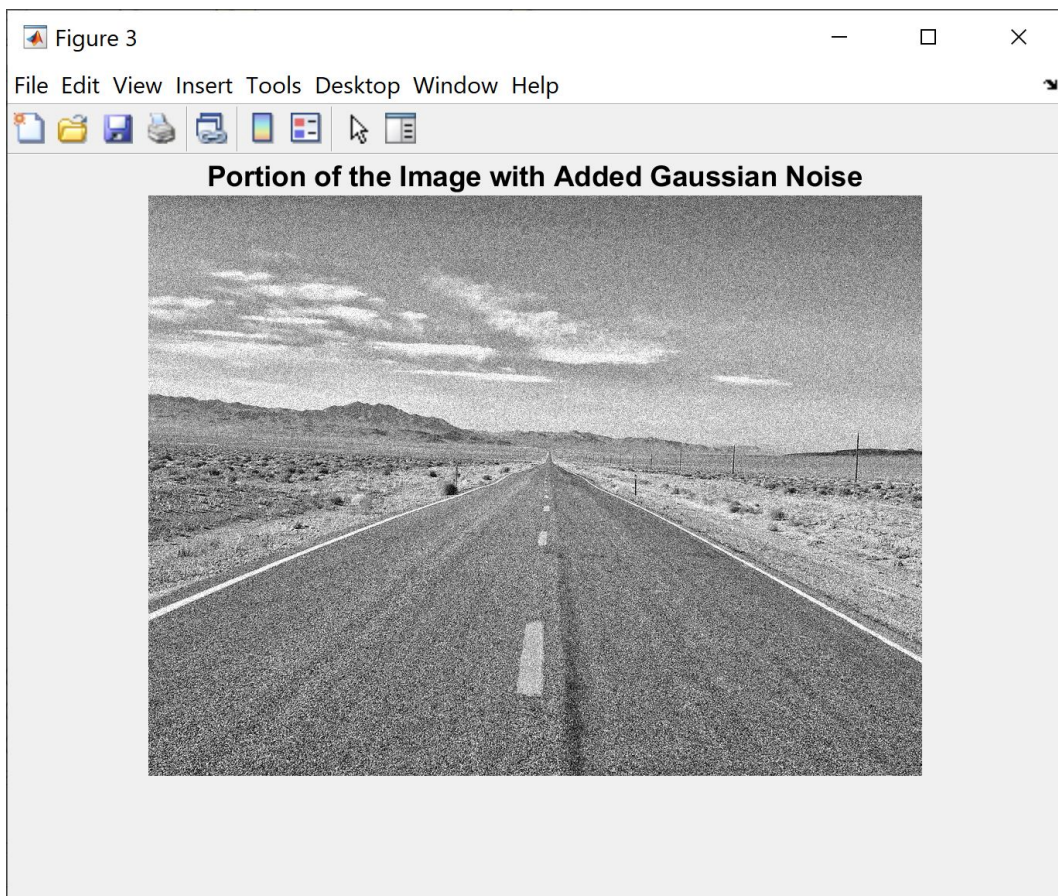
```
energy = sum(gray_img, 'all')
```

شکل ۴: تابع محاسبه انرژی عکس

چون نویز گوسین است و فرکانس ندارد پس نویز اضافه شده باعث می شود که تعداد نقاط سفیر بیشتر شود و فرکانس در وسط عکس پایین باشد و هر چه به سمت اطراف می رویم بیشتر شود. از لینک زیر بهره گرفته شد: <https://www.cs.unm.edu/brayer/vision/fourier.html>

گام ششم

در این مرحله با استفاده از تابع `imnoise` با میانگین ۰ و مقدار واریانس ۰.۱ به عکس ورودی ذخیره شده نویز اضافه کردیم. نتیجه را در عکس زیر می توانید مشاهده نمایید.



شکل ۵: تصویر موردنظر پس از اضافه شده نویز

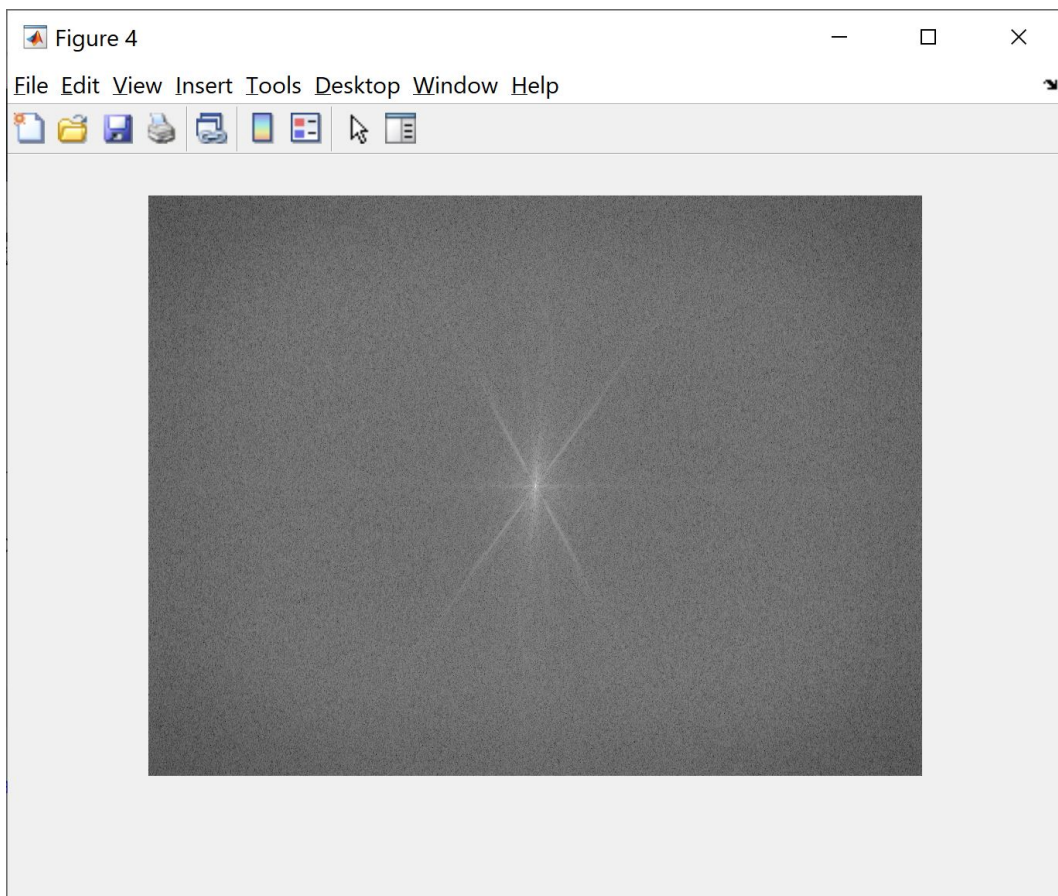
در مورد معیار SNR می توان گفت که این مورد در متلب با استفاده از تابع $\text{snr}(x, y)$ انجام می شود. با سرچی که من انجام دادم برای فهم بهتر به صورت موجود در عکس زیر این مقدار را برای قبل و بعد از اضافه شدن نویز محاسبه کردم.

```
img=input_img;  
img=double(img(:));  
ima=max(img(:));  
imi=min(img(:));  
mse=std(img(:));  
before_snr=20*log10((ima-iml)/mse)  
  
img=noisy_img;  
img=double(img(:));  
ima=max(img(:));  
imi=min(img(:));  
mse=std(img(:));  
after_snr=20*log10((ima-iml)/mse)
```

شکل ۶: رابطه استفاده شده برای محاسبه معیار SNR

گام هفتم

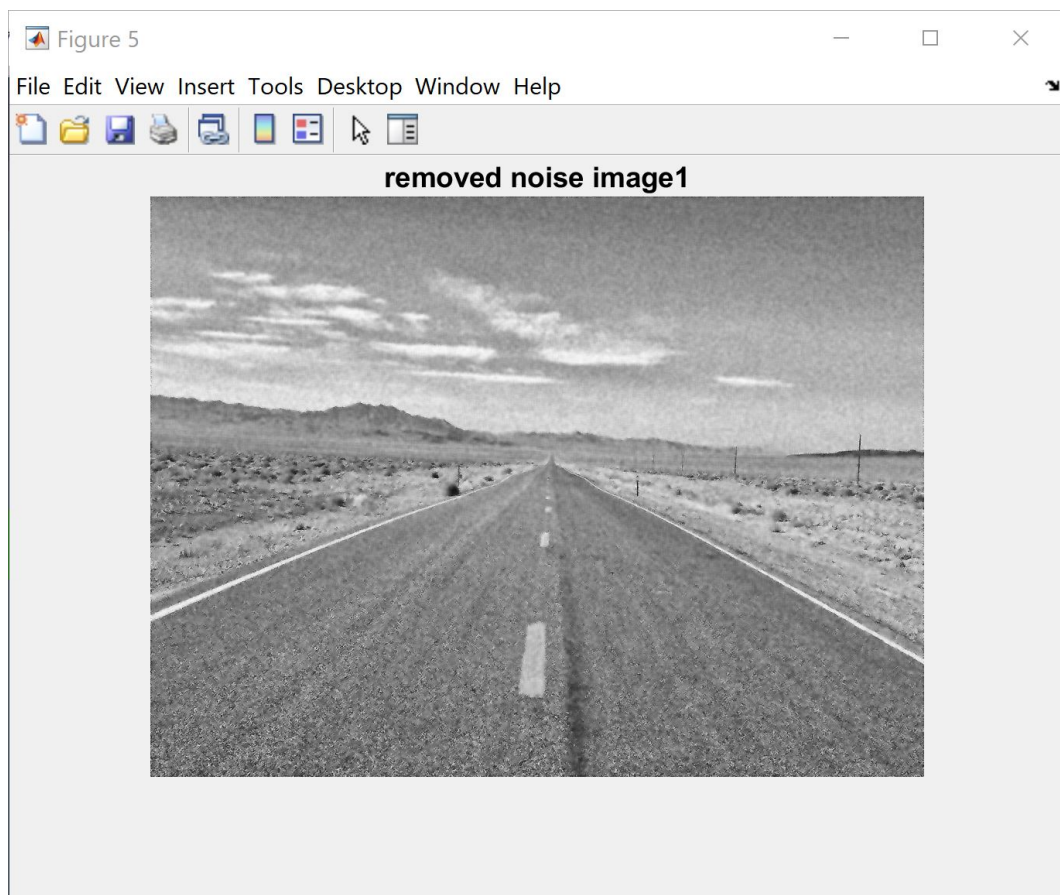
در این مرحله تصویر خاکستری را به حوزه فرکانس بردم. برای این کار از تابع `fftshift` انجام می شود. سپس تصویر انتقال یافته در حوزه فرکانس را مجدد با تابع `imshow` نمایش دادم.



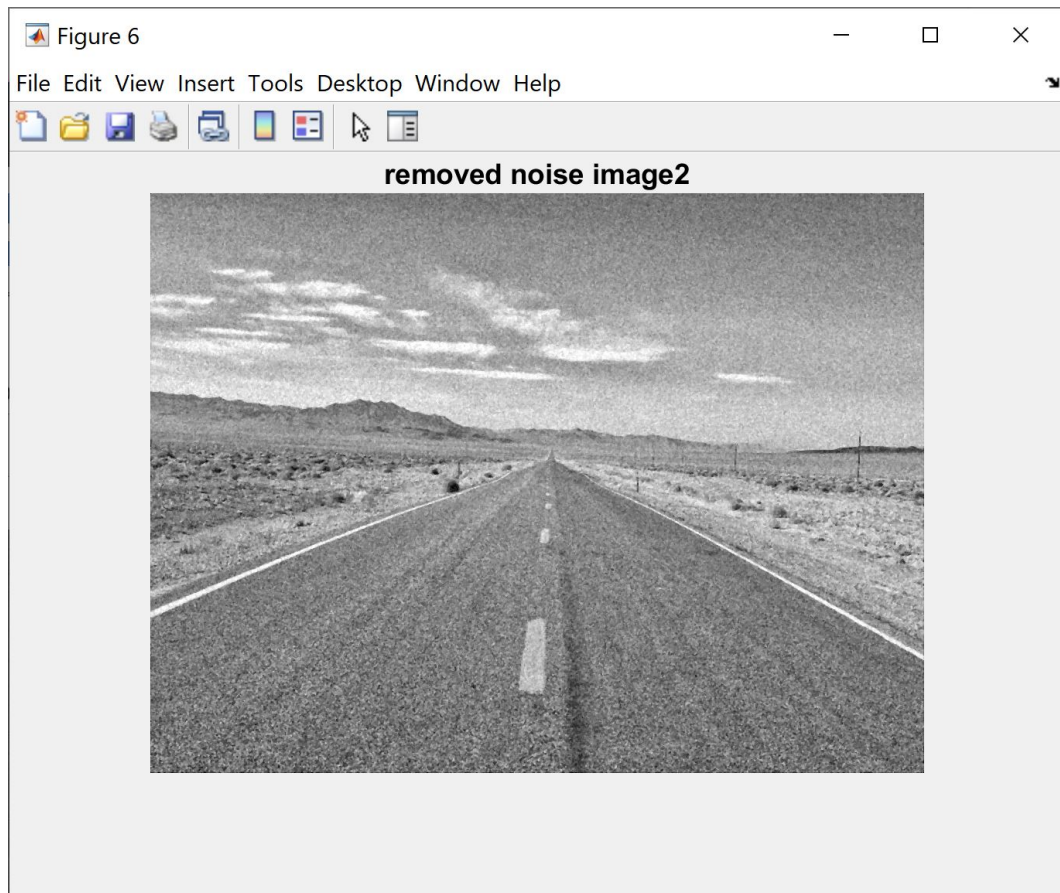
شکل ۷: تصویر انتقال یافته به حوزه فرکانس

گام هشتم

در این مرحله می توان از روش های متفاوتی برای رفع نویز عکس استفاده کرد. این موارد شامل روش wiener2 و یا median filter و یا average filter می شود. در اینجا من دو روش median filter و wiener2 استفاده کردم



شکل ۸: تصویر رفع نویز ده توسط wiener2



شکل ۹: تصویر رفع نویز ده توسط median filter

در ادامه برای اینکه بفهمیم چقدر خوب رفع نویز کردیم از معیار PSNR استفاده کردیم. این اسم مخفف Peak Signal to Noise Ratio است. این معیار بیانگر حداکثر میزان توان سیگنال و قدرت نویز موجود در آن است. برای محاسبه این معیار در متلب از تابع psnr استفاده کردم. این تابع تصویر رفع نویز شده و نویزی را دریافت می کند و این معیار را محاسبه می کند. این مقدار برای روش اول ۳۴.۱۹ و برای روش دوم ۷۸.۱۷ شد.