



دانشگاه صنعتی شریف

دانشکده مهندسی برق

پردازش تصاویر پزشکی

تمرین سری سوم

دانشجو

سید ابوالفضل مرتضوی

۴۰۲۲۰۰۱۹۱

اردیبهشت ۱۴۰۳

فهرست مطالب

۱	سوالات تئوری
۱	۱-۱- سوال اول
۲	۲-۱- سوال دوم
۳	۳-۱- سوال سوم
۳	۳-۱-۳- شبکه‌های عصبی کانولوشنی
۳	۲-۳-۱- Autoencoder
۴	۳-۳-۱- شبکه‌های مولد یا GAN
۵	سوالات عملی
۵	۱-۲- سوال اول
۱۲	۲-۲- سوال دوم
۱۲	۱-۲-۲- الف
۱۳	۲-۲-۲- ب
۱۴	۳-۲-۲- ج
۱۴	۴-۲-۲- کد
۲۲	۳-۲- سوال سوم
۲۲	۱-۳-۲- الف
۲۳	۲-۳-۲- ب
۲۴	۳-۳-۲- ج
۲۴	۴-۳-۲- د
۲۷	۴-۲- سوال چهارم
۲۷	۵-۲- الف
۲۸	۱-۵-۲- ب
۲۹	۲-۵-۲- ج
۳۰	۳-۵-۲- د

فهرست اشکال

..... ۶	شکل ۲-۱ آپدیت ضرایب C
..... ۶	شکل ۲-۲ توابع مربوط به آپدیت
..... ۷	شکل ۲-۳ تصویر اصلی و تصویر نویزی
..... ۹	شکل ۲-۴ نمودارهای مربوط به تغییرات معیارهای PSNR و SSIM برای مقادیر مختلف lam و conEff
..... ۹	شکل ۲-۵ تصویر اصلی، تصویر حذف نویز شده با بهترین معیار PSNR و تصویر حذف نویز شده با بهترین معیار SSIM
..... ۱۱	شکل ۲-۶ نمودار تغییرات PSNR و SSIM
..... ۱۲	شکل ۲-۷ تصویر اصلی، تصویر حذف نویز شده با بهترین معیار PSNR و تصویر حذف نویز شده با بهترین معیار SSIM
..... ۱۸	شکل ۲-۸ نمودارهای مقادیر SSIM و NIQE
..... ۲۱	شکل ۲-۹ نمودارهای مقادیر NIQE و SSIM
..... ۲۲	شکل ۲-۱۰ تصویر اصلی، تصویر نویزی و تصاویر حذف نویز شده
..... ۲۳	شکل ۲-۱۱ تصویر اصلی، تصویر نویزی با مقدار ۱۰٪ و تصویر حذف نویز شده
..... ۲۳	شکل ۲-۱۲ تصویر اصلی، تصویر نویزی با مقدار ۹۰٪ و تصویر حذف نویز شده
..... ۲۴	شکل ۲-۱۳ تصویر اصلی، تصویر نویزی با مقدار ۱۰٪ و تصویر حذف نویز شده
..... ۲۵	شکل ۲-۱۴ تصویر اصلی، تصویر نویزی با مقدار ۹۰٪ و تصویر حذف نویز شده
..... ۲۵	شکل ۲-۱۵ معیار MSE
..... ۲۶	شکل ۲-۱۶ معیار PSNR
..... ۲۶	شکل ۲-۱۷ معیار SNR
..... ۳۱	شکل ۲-۱۸ خروجی به دست آمده

سوالات تئوری

۱-۱ - سوال اول

Subject: _____
 Year: _____ Month: _____ Date: _____

 Sa Su Mo Tu We Th Fr

$J = \lambda \|X - Y\|_F^2 + \sum_{i,j} \|D\alpha_{ij} - R_{ij}X\|_F^2$ □

$J = \lambda (X - Y)^T (X - Y) + \sum_{i,j} (D\alpha_{ij} - R_{ij}X)^T (D\alpha_{ij} - R_{ij}X)$

$\frac{\partial J}{\partial X} = \lambda (X - Y)^T + \sum (D\alpha - R_X)^T (-R_{ij}) = 0$ 5

$\langle X^T - Y^T, D\alpha^T \rangle - \sum \langle X^T, D^T R \rangle + \sum \langle X^T, R^T R \rangle = 0 \rightarrow$

$X^T (\lambda I + \sum R^T R) = (\lambda Y^T + \sum \alpha^T D^T R) \xrightarrow{10}$

$(\lambda I + \sum R^T R) X = (\lambda Y + \sum \alpha^T D^T R) \rightarrow$

$X = (\lambda I + \sum_{i,j} R_{ij} R_{ij}^T)^{-1} (\lambda Y + \sum_{i,j} R_{ij}^T D\alpha_{ij})$ 15

20

۱- سوال دوم

Subject:

Year:

Month:

Date:

Sa	Su	Mo	Tu	We	Th	Fr
----	----	----	----	----	----	----

$$J = \frac{1}{n} \sum \sum u_{ij} \|x_i - w_j\|^2 + \sum \lambda_i (\sum u_{ij} - 1) \rightarrow \text{Eq. 1}$$

$$\frac{\partial J}{\partial w_j} = 0 \rightarrow \frac{1}{n} \sum \sum u_{ij} (x_i - w_j)^T = 0 \rightarrow$$

$$\frac{1}{n} \sum \sum u_{ij} x_i = \frac{1}{n} \sum \sum u_{ij} x_i^T \rightarrow w_j = \frac{\sum_{i=1}^n u_{ij} x_i}{\sum_{i=1}^n u_{ij}} \quad \text{Eq. 2}$$

$$\frac{\partial J}{\partial u_{ij}} = 0 \rightarrow \frac{1}{n} u_{ij} \|x_i - w_j\|^2 + \lambda_i \geq 0 \quad \text{Eq. 3}$$

$$\frac{\partial J}{\partial \lambda_i} > 0 \rightarrow \sum_{j=1}^K u_{ij} - 1 = 0$$

$$\frac{1}{n} u_{ij} \geq \frac{-\lambda_i}{\|x_i - w_j\|^2} \quad \text{Eq. 4}$$

$$\frac{1}{n} \sum_{L=1}^K u_{iL} = -\lambda_i \left\{ \frac{1}{\|x_i - w_L\|^2} \right\} \rightarrow \frac{1}{n} = -\lambda_i \sum_{L=1}^K \frac{1}{\|x_i - w_L\|^2} \quad \text{Eq. 5}$$

$$\lambda_i = \frac{1/n}{\sum_{L=1}^K \frac{1}{\|x_i - w_L\|^2}} \quad \text{Eq. 6}$$

$$\text{Eq. 4} \wedge \text{Eq. 6} \rightarrow \frac{1}{n} u_{ij} = \frac{1}{\sum_{L=1}^K \frac{1}{\|x_i - w_L\|^2}} \times (\|x_i - w_j\|^2)$$

$$= \frac{\frac{1}{\|x_i - w_j\|^2}}{\sum_{L=1}^K \frac{1}{\|x_i - w_L\|^2}} \quad \text{DOLPHIN}$$

۱-۳- سوال سوم

۱-۳-۱- شبکه‌های عصبی کانولوشنی

شبکه‌های عصبی کانولوشنی (CNN) برای وظایف مختلف بینایی ماشین، از جمله حذف نویز تصاویر، به طور گسترده‌ای استفاده می‌شوند. ایده اصلی در این رویکرد، آموزش یک شبکه عصبی عمیق به منظور یادگیری تطبیق میان تصاویر نویزی و نسخه‌های پاک آن‌ها است. معماری شبکه معمولاً شامل چندین لایه کانولوشنی است که ویژگی‌های محلی تصویر را به دست می‌آورند و به آن‌ها تابع فعال سازی غیرخطی مانند ReLU اعمال می‌کنند. این شبکه ممکن است شامل لایه‌های پولینگ برای کاهش نمونه برداری و لایه‌های افزایش اندازه برای بزرگنمایی نمونه‌ها نیز باشد. لایه نهایی شبکه معمولاً عمل جستجوی مقدار پیکسل به پیکسل را برای تولید تصویر پاک سازی شده انجام می‌دهد. برای آموزش یک CNN برای حذف نویز، لازم است یک مجموعه داده از تصاویر نویزی و پاک استفاده شود. شبکه آموزش داده می‌شود تا تفاوت بین تصاویر پاک پیش‌بینی شده و تصاویر پاک واقعی را کمینه کند. تابع خطای معمولاً استفاده شده در وظایف حذف نویز، خطای میانگین مربعات (MSE) یا خطای دریافتی است که اختلاف دریافتی بین تصاویر پاک سازی شده و تصاویر پاک را اندازه‌گیری می‌کند. شبکه با استفاده از الگوریتم‌های پس انتشار خطا backpropagation و گرادیان کاهشی descent gradient بهینه می‌شود تا وزن‌ها و بایاس‌ها به روز شده و به شبکه اجازه دهد که به طور موثر تصاویر را پاک سازی کند.

۱-۳-۲- Autoencoder

توانکودرها (Autoencoders) معماری شبکه عصبی‌ای هستند که شامل یک کدگذار و یک کدگشا می‌شوند. کدگذار تصویر ورودی را به یک نمایش کم بعد (فضای کدگذاری یا لاتنت latent) فشرده می‌کند و کدگشا تصویر را از این نمایش بازسازی می‌کند. در حوزه حذف نویز، یک اتوانکودر آموزش داده می‌شود تا تصاویر نویزی را به فضای لاتنت تبدیل کند و سپس آن‌ها را به تصاویر پاک بازسازی کند. در طول آموزش، از مجموعه داده‌ای شامل تصاویر جفت نویزی و پاک استفاده می‌شود. اتوانکودر آموزش داده می‌شود تا خطای بازسازی بین تصاویر نویزی و متناظر پاک خود را به حداقل برساند. این باعث می‌شود اتوانکودر نمایشی قوی از تصاویر پاک را یاد بگیرد که به طور موثر از نویز جدا می‌شود. تابع خطای استفاده شده برای آموزش می‌تواند خطای میانگین مربعات (MSE) یا سایر توابع خطای دریافتی باشد. پس از آموزش، اتوانکودر می‌تواند برای حذف نویز تصاویر نویزی جدید استفاده شود. تصویر نویزی از طریق کدگذار به فضای لاتنت منتقل شده و سپس توسط کدگشا بازسازی می‌شود.

۱-۳-۳- شبکه‌های مولد یا GAN

شبکه های مولد تخصصی(GAN) شامل یک مولد generator و یک تمییزدهنده discriminator هستند. مولد با توجه به تصاویر نویزی ورودی به تولید تصاویر پاک شده، می پردازد، در حالی که تمییزدهنده سعی در تمییز بین تصاویر پاک واقعی و تصاویر تولید شده (پاک شده) دارد. معمولاً مولد، یک مدل مبتنی بر CNN است که تصویر نویزی را به عنوان ورودی دریافت کرده و یک تصویر پاک شده را تولید می کند. تمییزدهنده نیز یک CNN است که هم تصاویر پاک واقعی و هم تصاویر پاک شده تولید شده را به عنوان ورودی دریافت کرده و سعی می کند بین آن ها تمایز بیندازد. مولد و تمییزدهنده به طور همزمان با هم آموزش داده می شوند. مولد سعی می کند تصاویری را تولید کند که تمییزدهنده نتواند آن ها را از تصاویر پاک واقعی تشخیص دهد و تمییزدهنده سعی می کند عملکرد تمایزدهی خود را بهبود بخشد. فرآیند آموزش شامل به روزرسانی مکرر وزن های مولد و تمییزدهنده است. وزن های مولد به روز می شوند تا توانایی تمییزدهنده در تمییز بین تصاویر واقعی و تصاویر تولید شده را به حداقل برساند، در حالی که وزن های تمییزدهنده به روز می شوند تا دقت تمییزدهی خود را بهبود بخشد. این آموزش مقابله ای باعث می شود مولد تصاویر پاک شده با کیفیت بالا تولید کند که نزدیک به تصاویر پاک واقعی باشد. پس از آموزش، مولد می تواند برای حذف نویز تصاویر نویزی جدید استفاده شود.

حال به بررسی دو روش FFDNet و DnCNN می پردازیم:

:DnCNN (Denoising Convolutional Neural Network)

یک مدل یادگیری عمیق است که به طور خاص برای وظیفه حذف نویز تصاویر طراحی شده است. DnCNN به دلیل عملکرد برترش، بسیاری از کاربردهای حذف نویز را دربر می گیرد. ایده اصلی پشت DnCNN آموزش یک شبکه عصبی عمیق به منظور یادگیری نگاشتی است بین تصاویر نویزی و نسخه های پاک آن ها. با یادگیری این نگاشت، DnCNN به خوبی نویز را از تصاویر حذف می کند. معماری DnCNN معمولاً شامل چندین لایه کانولوشنی است. این لایه ها مسئول استخراج ویژگی های محلی تصویر هستند و از توابع فعال سازی غیرخطی مانند ReLU برای بهبود عملکرد حذف نویز استفاده می کنند. شبکه بر روی مجموعه داده ای از جفت تصاویر نویزی و پاک آموزش می بیند و هدف آن کمینه کردن تفاوت بین تصاویر پاک پیش بینی شده و تصاویر پاک واقعی است DnCNN. عملکرد حذف نویز قابل تحسینی داشته و می تواند با سطوح و انواع مختلف نویز مقابله کند.

:FFDNet (Fast and Flexible Denoising Network)

یک مدل پرطرفدار دیگر برای حذف نویز است که بر روی حالت های حذف نویز به طور لحظه ای و قابلیت انعطاف پذیری تمرکز دارد. این مدل برای مقابله با سطوح متنوع نویز و سازگاری با خصوصیات مختلف نویز طراحی شده است FFDNet. از یک معماری پی در پی تشکیل شده است که شامل مرحله

تخمین نویز و مرحله حذف نویز است. در مرحله تخمین نویز FFDNet ، مدل سطح نویز موجود در تصویر نویزی ورودی را تخمین می زند. این تخمین برای تعیین استراتژی حذف نویز بهینه حائز اهمیت است. در مرحله حذف نویز FFDNet با استفاده از سطح نویز تخمین زده شده، حذف نویز موثر انجام می دهد. این مدل از ویژگی های فیلترینگ غیر محلی (means local-non) استفاده می کند و از فیلترهای مشترک برای دستیابی به عملکرد حذف نویز سریع استفاده می کند. تفاوت اصلی بین FFDNet و DnCNN در طراحی معماری و استراتژی های حذف نویز آن ها قرار دارد DnCNN. سعی در پیدا کردن نگاشت بین تصاویر نویزی و پاک با استفاده از شبکه های عمیق کانولوشنی دارد. این مدل بر روی ویژگی های محلی تصویر عمل می کند و از توابع فعال سازی غیرخطی برای بهبود عملکرد حذف نویز استفاده می کند. از سوی دیگر، FFDNet از یک معماری پی در پی با مرحله تخمین نویز و مرحله حذف نویز استفاده می کند. استراتژی حذف نویز FFDNet بر اساس سطح نویز تخمین زده شده تنظیم می شود و از فیلترینگ غیر محلی (means local-)non با فیلترهای مشترک استفاده می کند. هر دو مدل قدرتمندی هستند و تصاویر پاک با کیفیت بالا تولید می کنند DnCNN. به عنوان یکی از بهترین مدل ها در حذف نویز شناخته شده است و با سطوح و انواع مختلف نویز مقابله می کند FFDNet. نیز به دلیل انعطاف پذیری و سرعت بالایش در حالت های حذف نویز لحظه ای مورد توجه قرار می گیرد.

سوالات عملی

۱-۲ - سوال اول

در پیادهسازی این سوال از مقاله‌ی زیر استفاده شده است.

<http://image.diku.dk/imagecanon/material/PeronaMalik1990.pdf>

طبق مقاله‌ی مرجع، دو راه برای آپدیت ضرایب C وجود دارد. در ادامه به بیان هر روش و توضیح آن پرداخته می شود.

$$\begin{aligned} c'_{N_{i,j}} &= g(|\nabla_N I'_{i,j}|) \\ c'_{S_{i,j}} &= g(|\nabla_S I'_{i,j}|) \\ c'_{E_{i,j}} &= g(|\nabla_E I'_{i,j}|) \\ c'_{W_{i,j}} &= g(|\nabla_W I'_{i,j}|) \end{aligned}$$

شکل ۲-۱ آپدیت ضرایب C

$$g(\nabla I) = e^{(-(\|\nabla I\|/K)^2)}$$

$$g(\nabla I) = \frac{1}{1 + \left(\frac{\|\nabla I\|}{K}\right)^2}$$

شکل ۲-۲ توابع مربوط به آپدیت

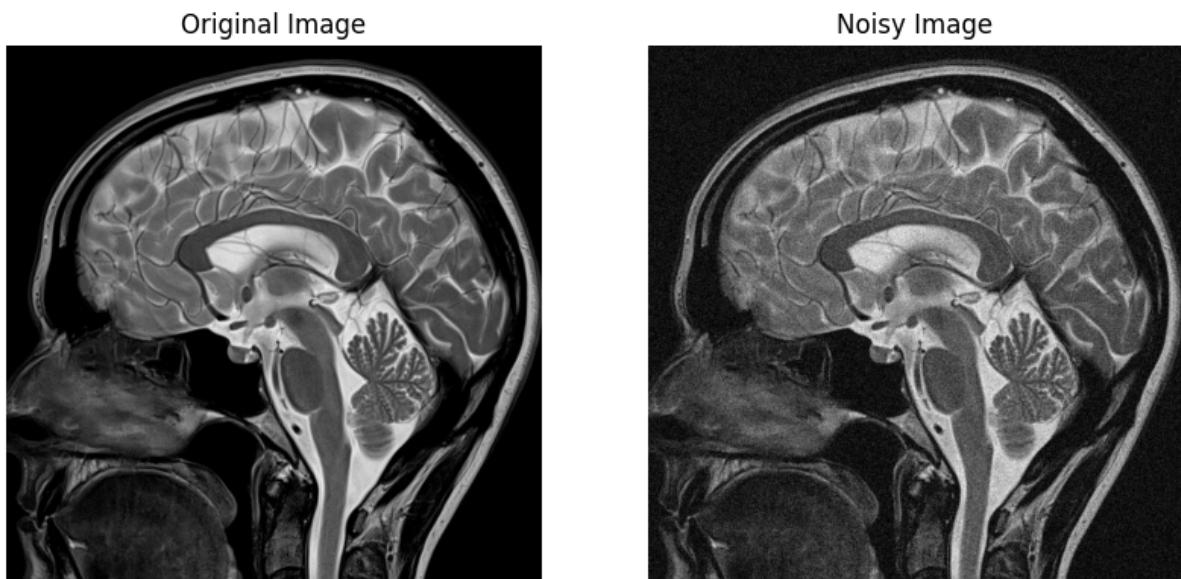
مقادیر ضرایب C در لحظه (یا در هر تکرار) طبق رابطه بالا محاسبه و بروزرسانی می‌شوند. برای انتخاب تابع g نیز در مقاله دو گزینه زیر را استفاده کردند که تابع اولی لبه‌های با کنتراست بالا را نسبت به لبه‌های کم کنتراست ترجیح میدهد، تابع دومی مناطق وسیع را نسبت به کوچکترها ترجیح میدهد.. مقدار K (یا کاپا) نیز میزان انتشار را به عنوان تابعی از گرادیان کنترل می‌کند. در حقیقت شکل این دو تابع شبیه کرنل یا فیلتر گوسی است و K نیز همانند انحراف معیار کرنل گوسی رفتار می‌کند. اگر مقدار K خیلی کم باشد اصلا هموارسازی اتفاق نمی‌افتد و تصویر نویزی باقی می‌ماند. اگر مقدار K خیلی زیاد باشد، رفتار این کرنل یا فیلترینگ همانند اعمال فیلتر گوسی (یا فیلتر isotropic) به تصویر نویزی می‌باشد و هموارسازی بدون توجه به لبه‌ها به صورت یکنواخت و همه جهات انجام می‌شود.

در این سوال تابعی پیاده شده که ورودی‌های آن به صورت زیر هستند.

```
def anisotropicFilter(img1, numIter, conEff, Lam,option):
```

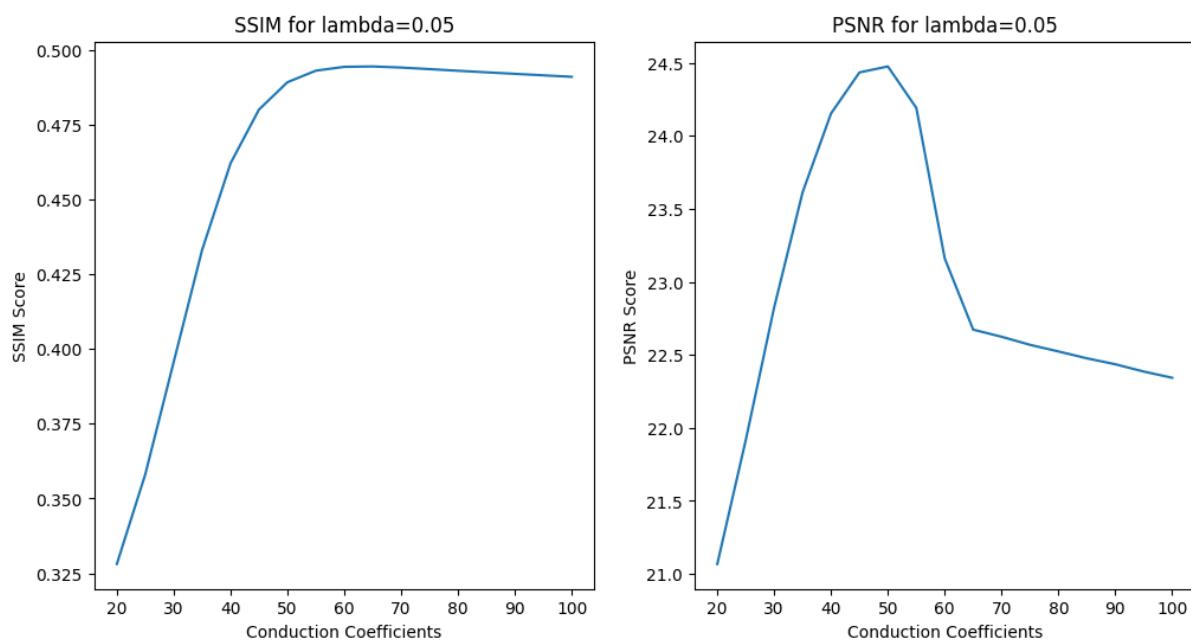
ورودی $img1$ تصویر نویزی است. ورودی $numIter$ تعداد تکرار برای الگوریتم است که در این سوال مقدار آن 20 در نظر گرفته شد. پارامتر $conEff$ همان ضریب K است. ورودی Lam مقداری است که حد تغییر تفاوت بین تصویر تمیز و نویزی را تعیین می‌کند. ورودی $option$ نیز برای تعیین تابع g برای محاسبه ضرایب C استفاده می‌شود.

در ادامه تصویر مربوط به این تمرین بارگذاری شده و نویز گوسی به آن اضافه شد.

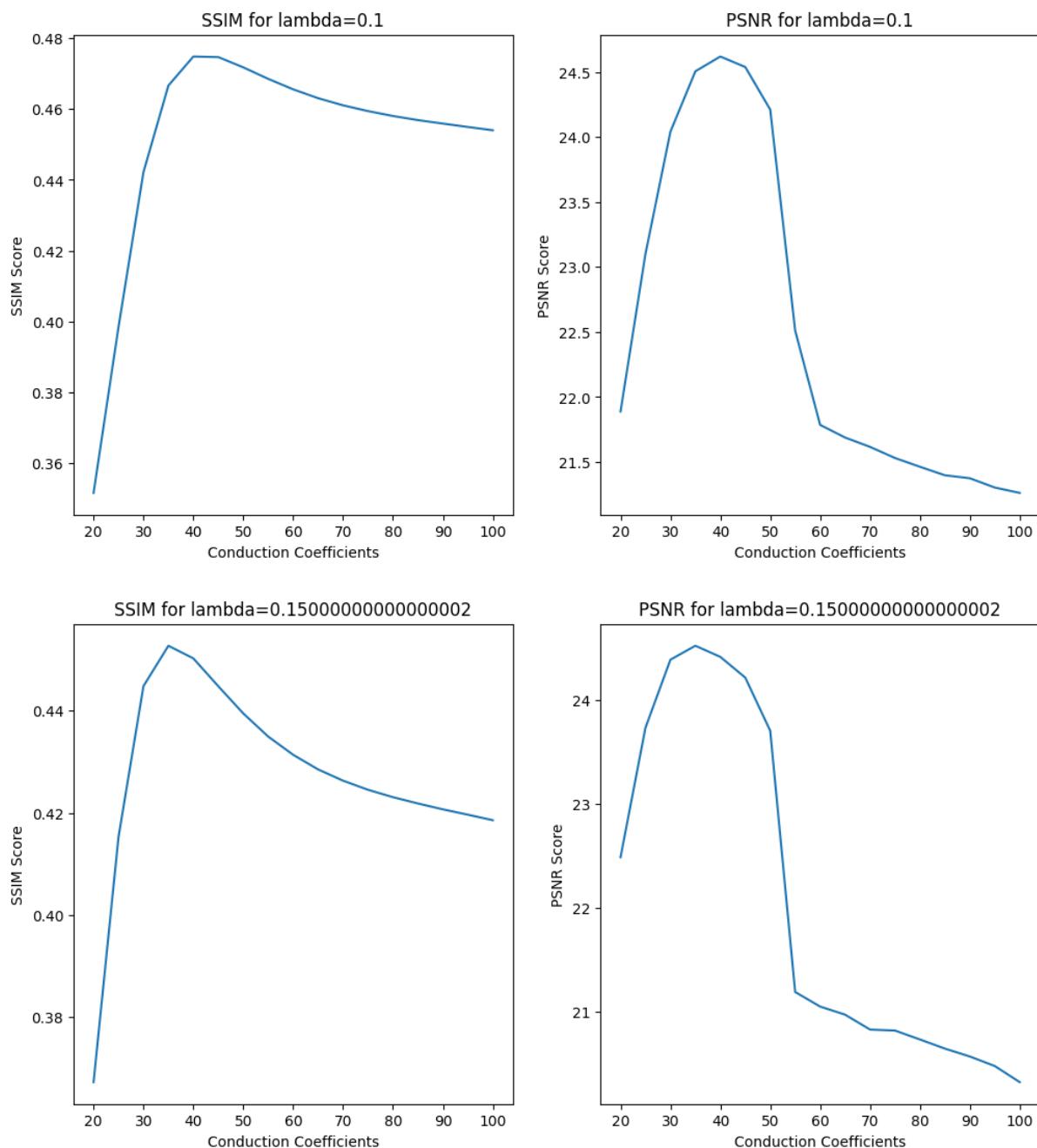


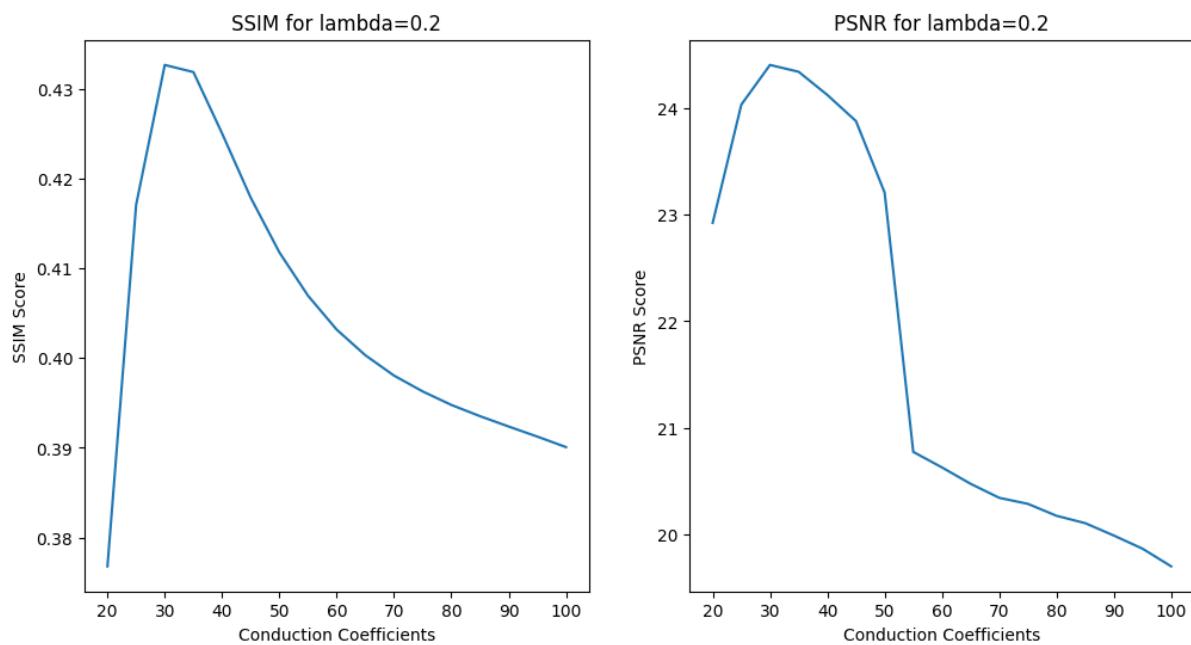
شکل ۳-۲ تصویر اصلی و تصویر نویزی

در ادامه برای به دست آوردن بهترین پارامترها از حلقه برای مقادیر conEff و lam به ازای هر استفاده شد و در هر مقدار، معیارهای PSNR و SSIM برای تصویر حذف نویز شده محاسبه شدند. برای خروجی به صورت زیر است.



تمرین سری سوم پردازش و تحلیل تصاویر پزشکی





شکل ۲-۴ نمودارهای مربوط به تغییرات معیارهای $PSNR$ و $SSIM$ برای مقادیر مختلف $conEff$ و lam

بهترین مقادیر به دست آمده به صورت زیر هستند.

Maximum SSIM value: 0.49

Related lambda: 0.05

Related coefficient: 65

Maximum PSNR value: 24.62

Related lambda: 0.10

Related coefficient: 40

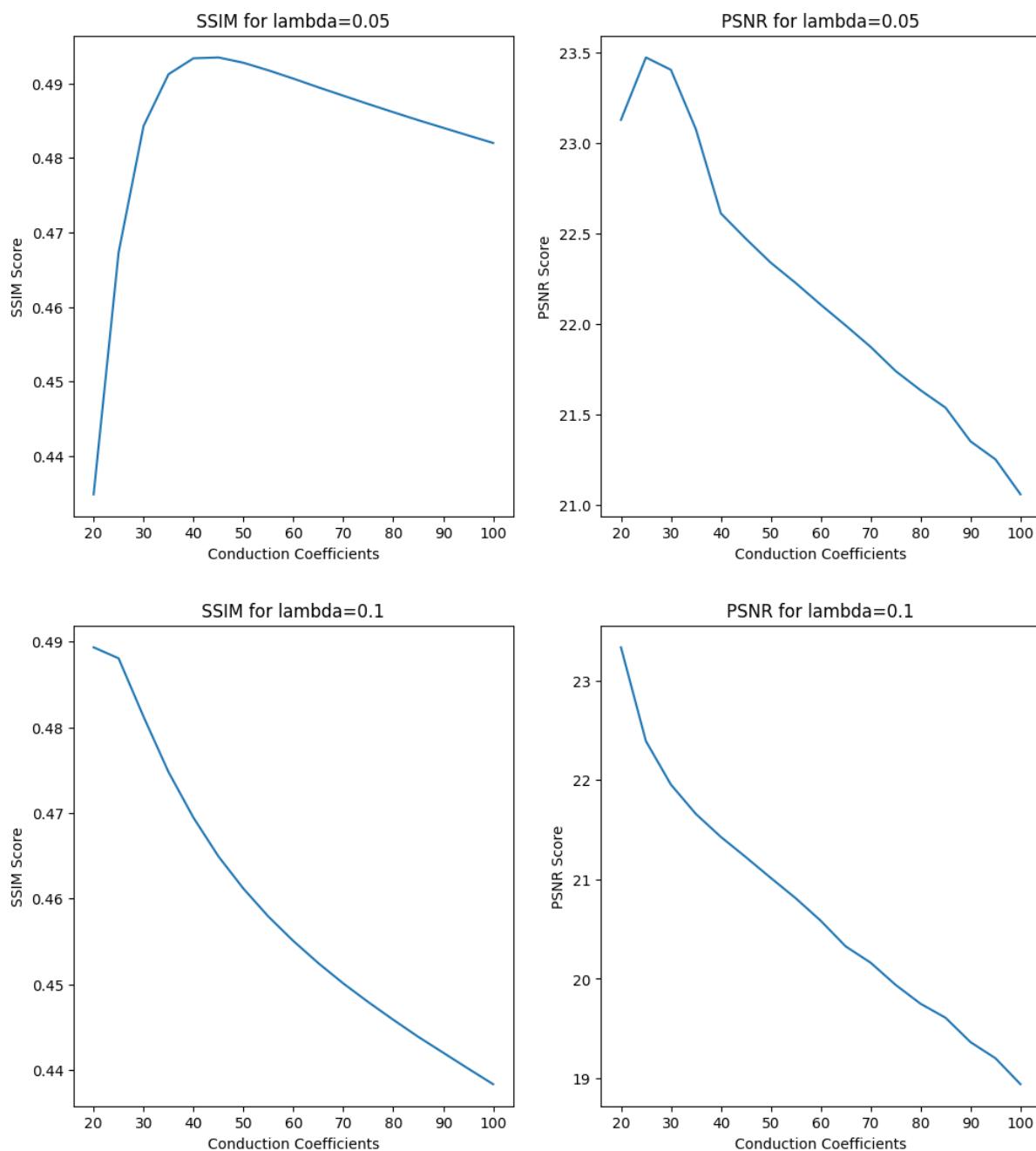
همانطور که مشاهده می شود پارامترهای مربوط به بهترین مقدار $PSNR$ و $SSIM$ با یکدیگر متفاوت هستند.
زیر تصویری که بهترین مقدار $PSNR$ و بهترین مقدار $SSIM$ را دارند نشان داده شده اند.

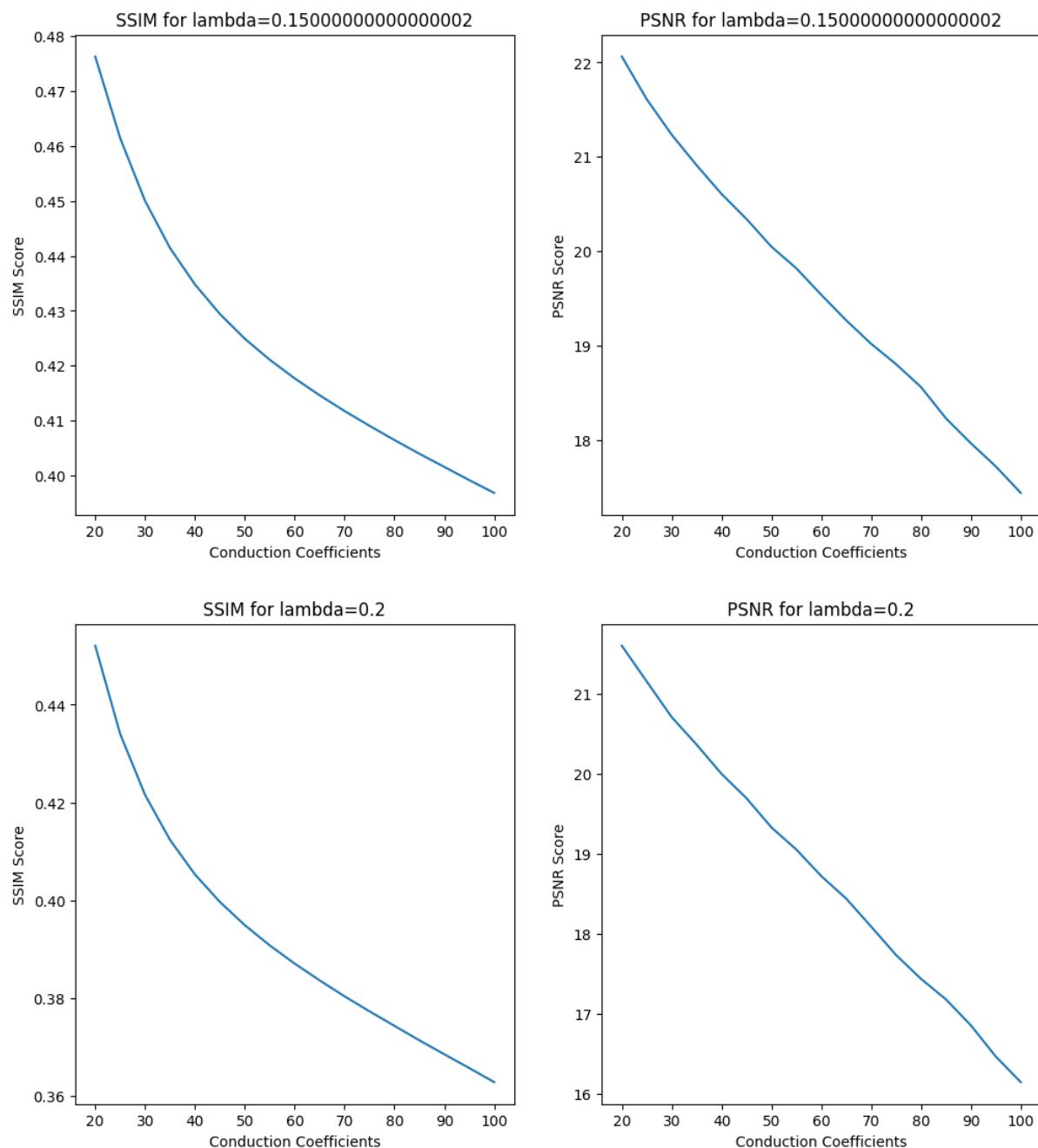


شکل ۲-۵ تصویر اصلی، تصویر حذف نویز شده با بهترین معیار $PSNR$ و تصویر حذف نویز شده با بهترین معیار $SSIM$

تمرین سری سوم پردازش و تحلیل تصاویر پزشکی

روند طی شده برای option=1 برای option=2 نیز تکرار شد و خروجی‌ها به صورت زیر هستند.





شکل ۲-۶ نمودار تغییرات SSIM و PSNR

خروجی بهترین مقدار معیار SSIM و PSNR

Maximum SSIM value: 0.49

Related lambda: 0.05

Related coefficient: 45

Maximum PSNR value: 23.47

Related lambda: 0.05

Related coefficient: 25



شکل ۷-۲- تصویر اصلی، تصویر حذف نویز شده با بهترین معیار PSNR و تصویر حذف نویز شده با بهترین معیار SSIM

در نهایت بهترین تصاویر حذف نویز شده با معیارهای گفته شده مربوط به `option=1` بودند که ذخیره شده و در فایل پیوست قرار دارند.

۲-۲ - سوال دوم

۱-۲-۱ - الف

توضیحات مربوط به کد و خروجی‌ها بعد از بخش ج آورده شده‌اند.

کد تابع به نام "anisodiff" برای رفع نویز تصویر استفاده می‌شود. در زیر توضیحی از عملکرد تک تک بخش‌های کد آمده است: ابتدا، تصویر ورودی "im" را به دوبل تبدیل کرده و سایز آن را در متغیرهای "rows" و "cols" ذخیره می‌کند. همچنین، تصویر نهایی را در متغیر "diff" تعریف می‌کند. سپس یک حلقه به تعداد "niter" اجرا می‌شود تا مراحل تکراری الگوریتم اجرا شود. در هر مرحله، تصویر "diff" را به اندازه یک پیکسل در هر جهت با صفرهای پر کنده شده در "diff1" کپی می‌کند. سپس، مشتق‌های شمال، جنوبی، غربی و شرقی تصویر "diff" را محاسبه می‌کند. در مرحله بعد، با توجه به مقدار "option"، ضریب‌های هادی را برای هر جهت محاسبه می‌کند. اگر "option" برابر با ۱ باشد، از تابع توزیع نرمال برای محاسبه ضریب‌ها استفاده می‌کند. اگر "option" برابر با ۲ باشد، از تابع توزیع رکورد برای محاسبه ضریب‌ها استفاده می‌کند. در نهایت، تصویر "diff" را با استفاده از ضریب‌ها و تفاوت‌ها به روزرسانی می‌کند. این عمل شامل جمع ضرب تفاوت‌ها در ضریب‌ها در هر جهت است. برای استفاده از این تابع، شما باید تصویر ورودی "im"، تعداد تکرار "niter"، ضریب "kappa"، پارامتر "lambda" و گزینه "option" را به عنوان ورودی به تابع ارسال کنید. خروجی این تابع، تصویر رفع نویز شده است. کد تابع "isodiff" نیز برای رفع نویز تصویر استفاده می‌شود. در مقایسه با تابع "anisodiff" که توضیح داده شد، تفاوت‌هایی در روش محاسبه ضریب

هادی و اعمال تفاوت‌ها وجود دارد. در این تابع، مانند قبل، تصویر ورودی "im" به نوع داده عددی دوبل تبدیل می‌شود و تصویر پایه مقداردهی می‌شود. سپس، مانند تابع قبل، مشتق‌های شمالی، جنوبی، شرقی و غربی هر پیکسل با مقادیر پیکسل مجاور آن محاسبه می‌شود. در اینجا، برای محاسبه ضرایب هادی در هر جهت، از یک ثابت به نام "constant" استفاده می‌شود که به عنوان ورودی به تابع ارسال می‌شود. تمام ضرایب هادی در این الگوریتم به همان مقدار ثابت برابر هستند. در نهایت، با استفاده از ضرایب هادی و تفاوت‌ها، تصویر پایه به روزرسانی می‌شود. این عمل شامل جمع وزن دار تفاوت‌ها در هر جهت با توجه به مقادیر ضرایب هادی است. تفاوت اصلی بین تابع "isodiff" و "anisodiff" در روش محاسبه ضرایب هادی و مقداردهی به آن‌ها است. در "isodiff"، تمام ضرایب هادی به یک مقدار ثابت تنظیم می‌شوند، در حالی که در "anisodiff"، محاسبه ضرایب هادی بر اساس شدت تغییرات روشنایی در هر جهت صورت می‌گیرد. این باعث می‌شود "isodiff" تصویر را با یک نرخ فیلترینگ یکنواخت تغییر دهد، در حالی که "anisodiff" نرخ فیلترینگ را در جهات مختلف تنظیم کرده و تغییرات روشنایی را بر اساس شدت آن‌ها انجام می‌دهد.

۲-۲-۲ - ب

یکی از معیارهای معروف برای ارزیابی کیفیت تصویر، شاخص شباهت ساختاری (Index Similarity) یا به اختصار SSIM (Structural Similarity) است. SSIM یک معیار ریاضی است که برای اندازه‌گیری شباهت بین تصاویر مبتنی بر ساختارهای آن‌ها استفاده می‌شود. این شاخص به مقایسه سه عامل مختلف بین تصویر مرجع (تصویر اصلی) و تصویر تحلیل شده (تصویر پردازش شده) می‌پردازد. این سه عامل عبارتند از: شباهت ساختاری (similarity structure luminance)، شباهت کنترast (similarity contrast) و شباهت سیگنال سیاه و سفید (similarity luminance). این معیار از ۰ تا ۱ محدوده مقادیر خود قرار می‌دهد که مقدار بالاتر به معنای شباهت بیشتر است و مقدار پایین‌تر به معنای شباهت کمتر یا تغییرات زیاد در تصویر است. فرمول محاسبه این معیار به صورت زیر است:

$$SSIM(x(i,j), y(i,j)) = \frac{(2\mu_{x(i,j)}\mu_{y(i,j)} + c_1)(2\sigma_{x(i,j)y(i,j)} + c_2)}{(\mu_{x(i,j)}^2 + \mu_{y(i,j)}^2 + c_1)(\sigma_{x(i,j)}^2 + \sigma_{y(i,j)}^2 + c_2)}$$

$\mu_{x(i,j)}$ and $\sigma_{x(i,j)}^2$ are the mean and variances of the window around pixel (i, j) , and $\sigma_{x(i,j)y(i,j)}$ is the covariance between windows around (i, j) in x and y .

یکی دیگر از معیارهای مورد استفاده برای ارزیابی کیفیت تصویر Naturalness Image Quality Evaluator است. این معیار یک مدل محاسباتی است که برای اندازه‌گیری کیفیت طبیعت تصویر استفاده می‌شود NIQE. با تحلیل ویژگی‌های مختلف تصویر مانند روشنایی، کانتراست، و جزئیات تصویر، به ارزیابی کیفیت طبیعت تصویر می‌پردازد. در معیار NIQE مقدار کمتر به معنای کیفیت بهتر و مقدار بیشتر به معنای کیفیت پایین‌تر است.

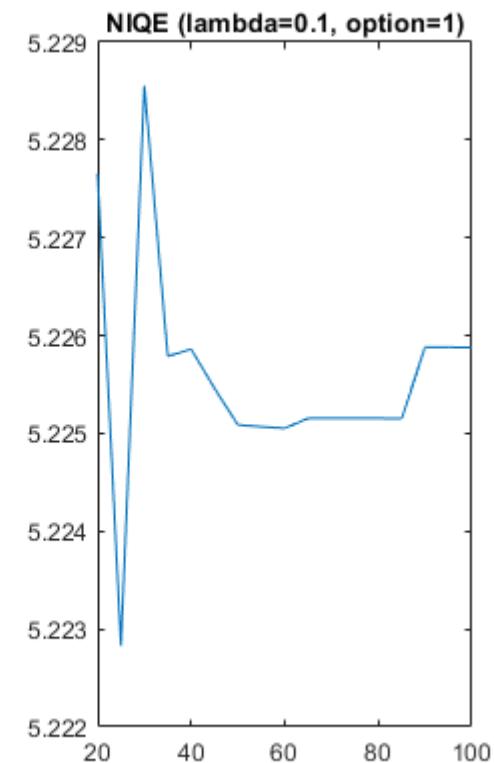
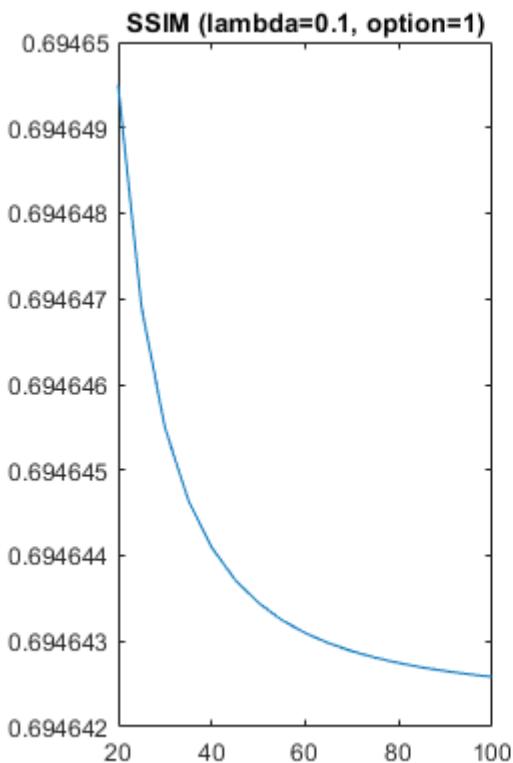
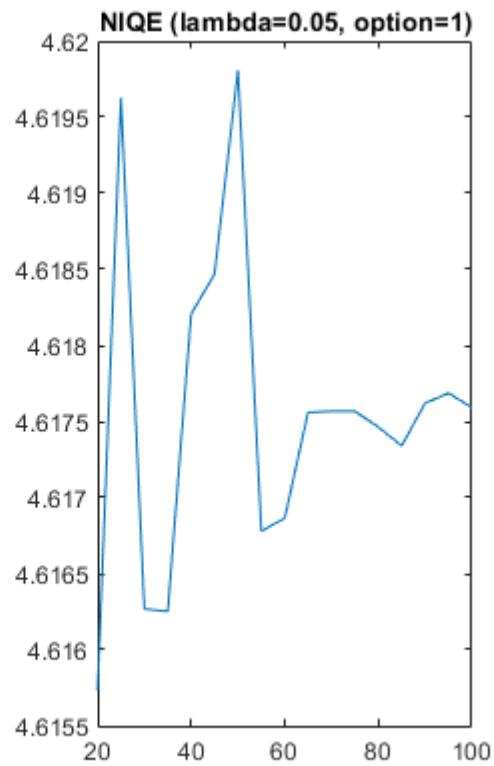
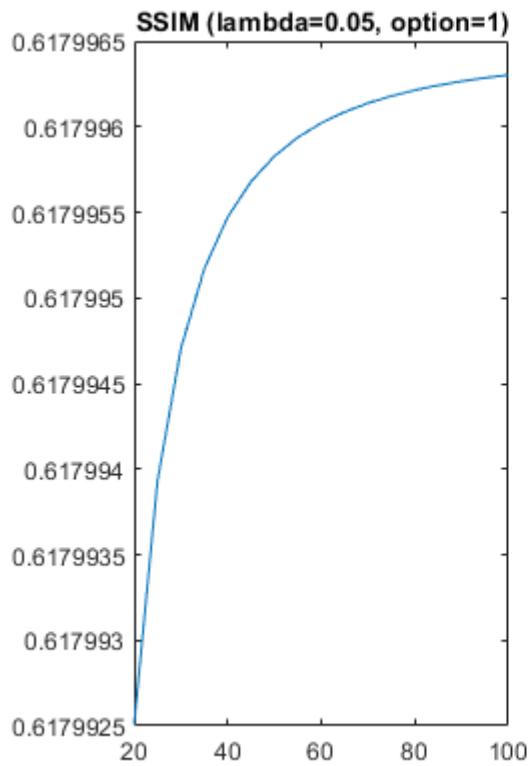
۴-۲-۳-ج

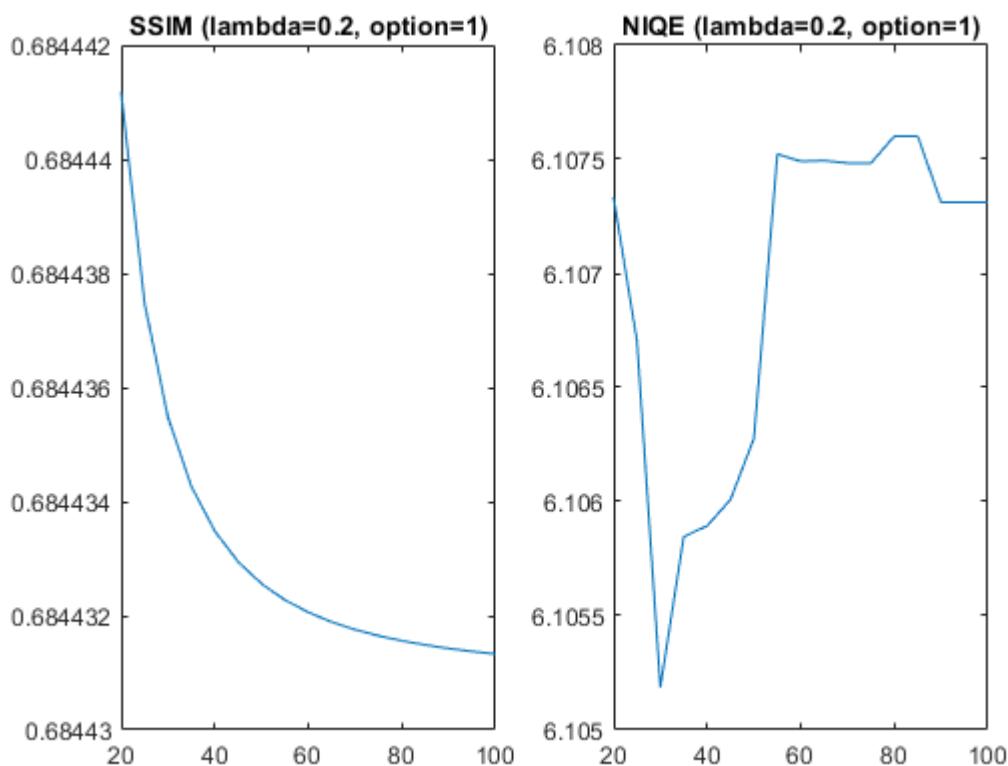
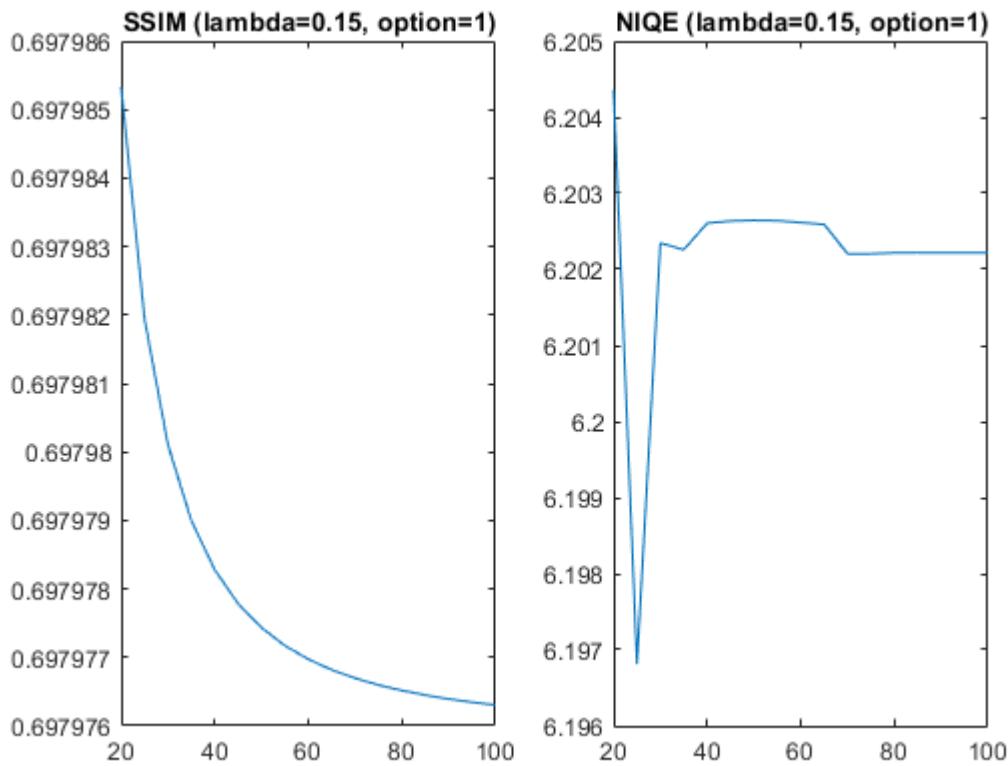
فیلترهای Anisotropic و Isotropic در پردازش تصویر برای کاهش نویز و بهبود جزئیات تصویر استفاده می‌شوند. در مورد فیلتر گوسی، این فیلتر معمولاً برای حذف نویز و تاری در تصاویر استفاده می‌شود. فیلتر Gaussian، یک فیلتر خطی است که برای حذف نویز و نرم کردن تصاویر استفاده می‌شود. فیلتر گوسی عموماً بر اساس توزیع گوسی ساخته می‌شود و با استفاده از یک ماسک (kernel) دو بعدی یا سه بعدی به تصویر اعمال می‌شود. فیلتر گوسی اثرات نرم کنندگی و انتشار را بر روی تصویر ایجاد می‌کند و جزئیات ریز تصویر را کاهش می‌دهد. میزان اثرات نرم کنندگی فیلتر گوسی به وابستگی نسبت مستطیل نمونه و پارامتر انحراف معیار توزیع گوسی است. در مقابل، فیلترهای Anisotropic (نمتران) توانایی حفظ لبه‌ها و جزئیات ریز در تصویر را دارند. این فیلترها برای حذف نویز و بهبود تصویر در مناطق باله‌های تند و جزئیات ریز مفید هستند. فیلترهای Anisotropic معمولاً بر اساس معیارهای متفاوتی مانند گرادیان تصویر، جهت لبه‌ها و نواحی همسایگی اعمال می‌شوند. این فیلترها به صورت هدفمند و به شکل محدودتری در نقاط مشخصی از تصویر به منظور حفظ جزئیات و جلوگیری از تغییرات غیرمطلوب در آن‌ها اعمال می‌شوند. فیلترهای Isotropic (نمتران) نیز همانند فیلترهای Anisotropic می‌توانند برای حذف نویز و بهبود تصویر استفاده شوند، اما از روش‌های متقارن و یکنواخت برای اعمال تغییرات به تصویر استفاده می‌کنند. این فیلترها به ترتیبی همچون فیلتر گوسی عمل می‌کنند و نسبت به جهت و محل تغییرات تصویر حساسیت کمتری دارند.

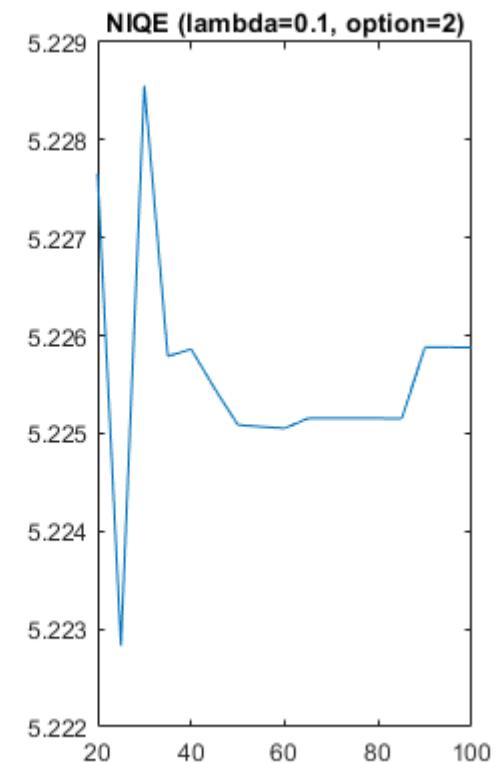
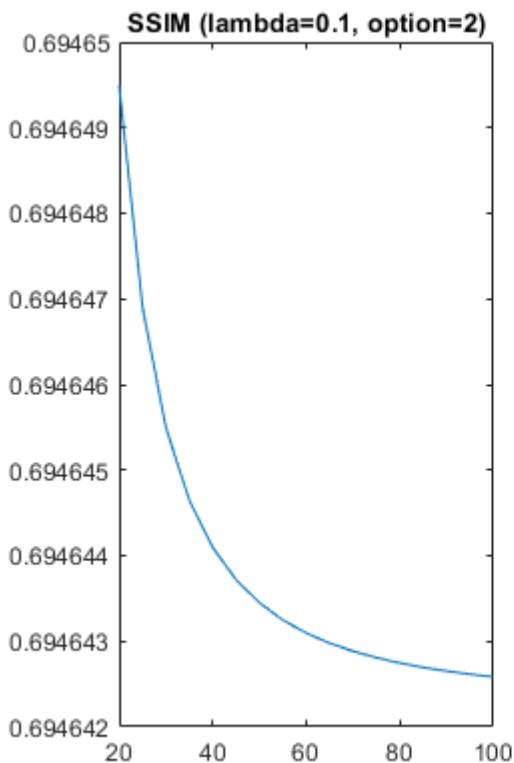
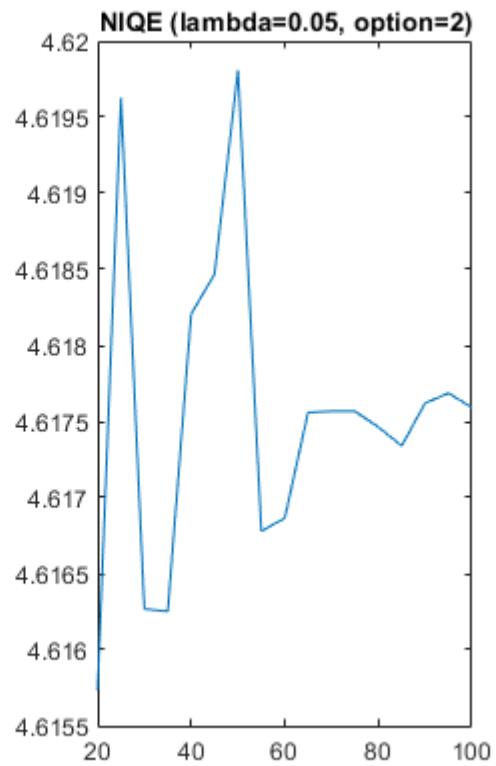
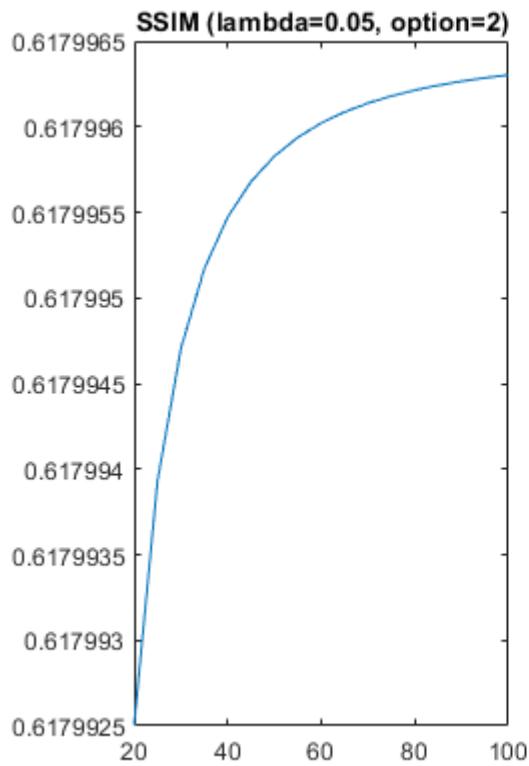
۴-۲-۴-کد

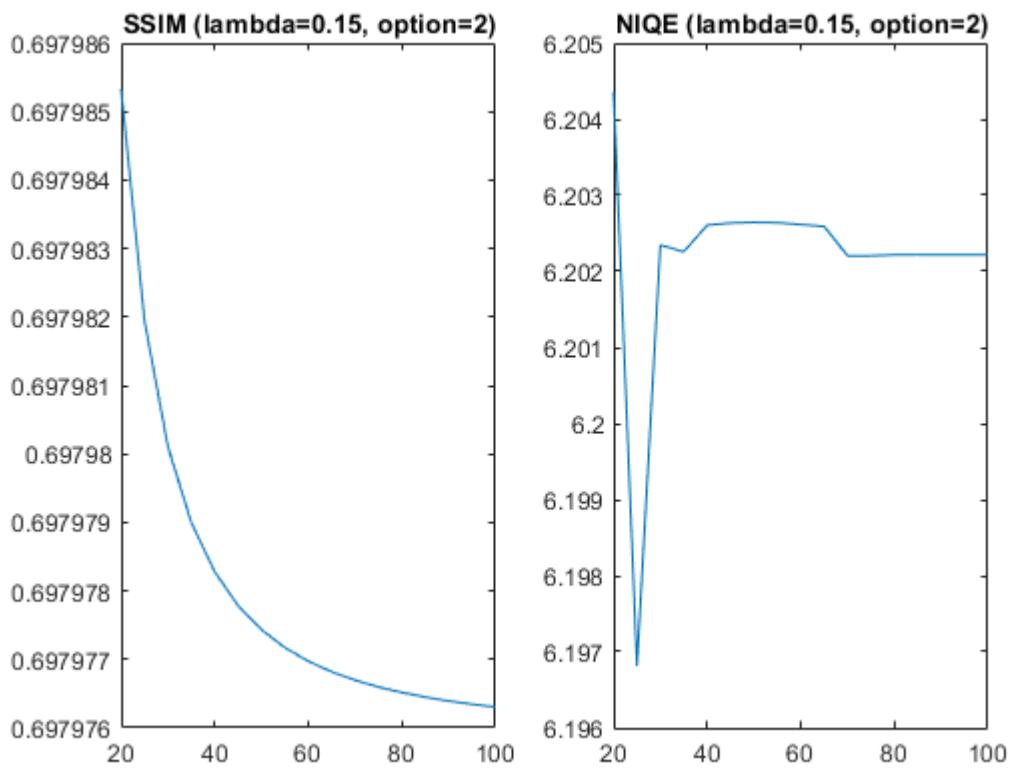
در این بخش ابتدا با استفاده از حلقه‌ها برروی پارامترهای مختلف توابع و همچنین سنجش معیارهای معرفی شده در بالا، برای تصاویر حذف نویز شده، بهترین پارامترها به دست آمدند.

در تصویر زیر نمودارهای تغییرات مقادیر SSIM و NIQE بر حسب پارامترهای مختلف در تابع anisodiff نشان داده شده‌اند.









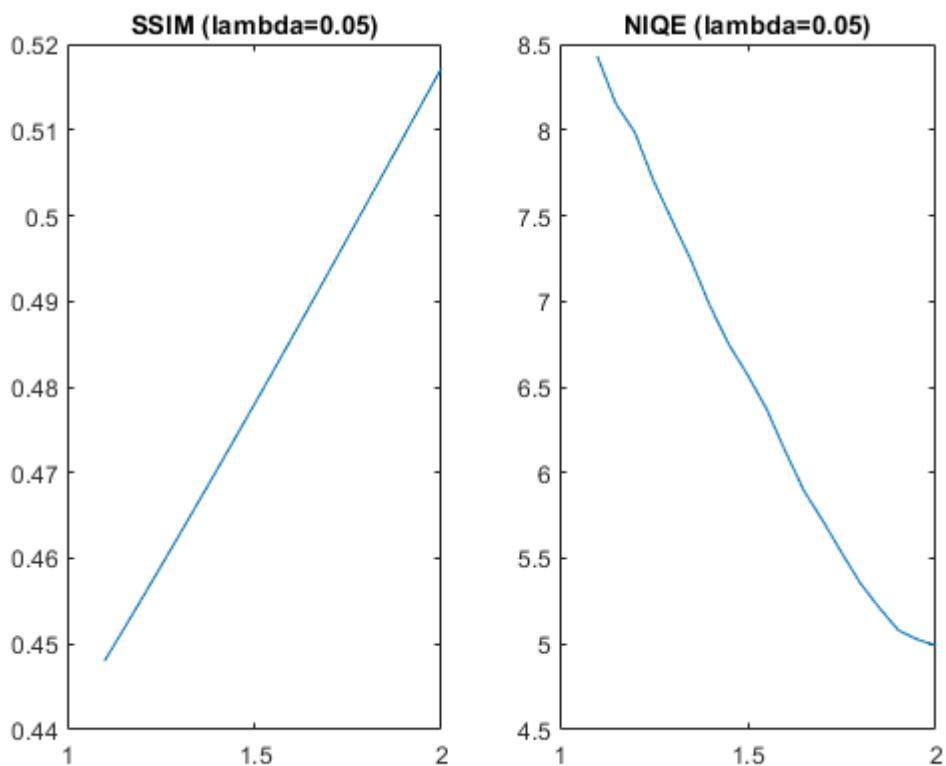
شکل ۲-۸ نمودارهای مقادیر SSIM و NIQE

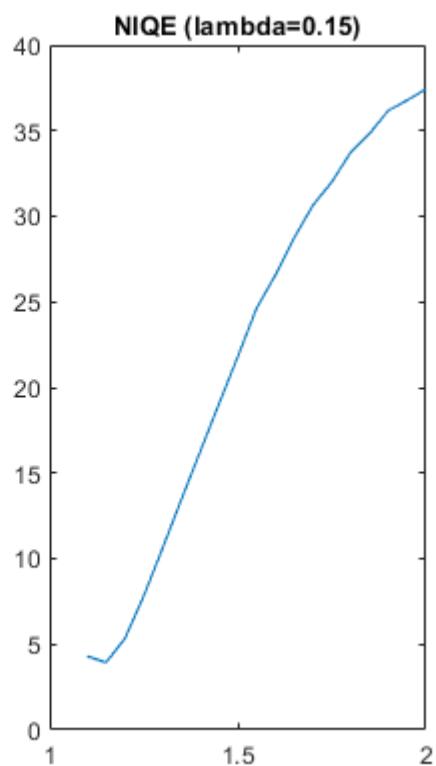
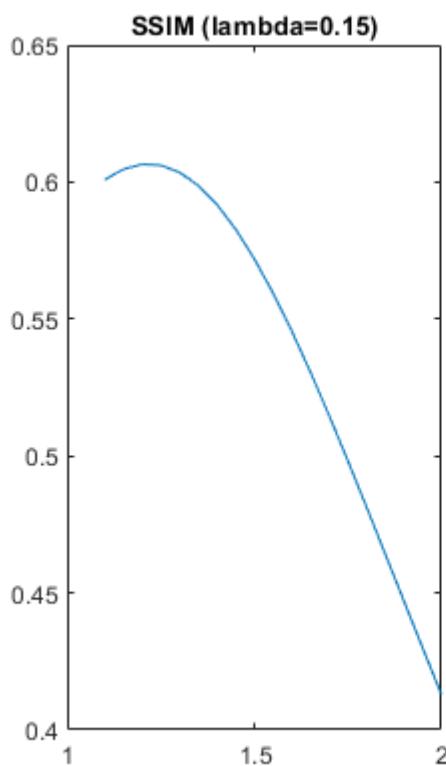
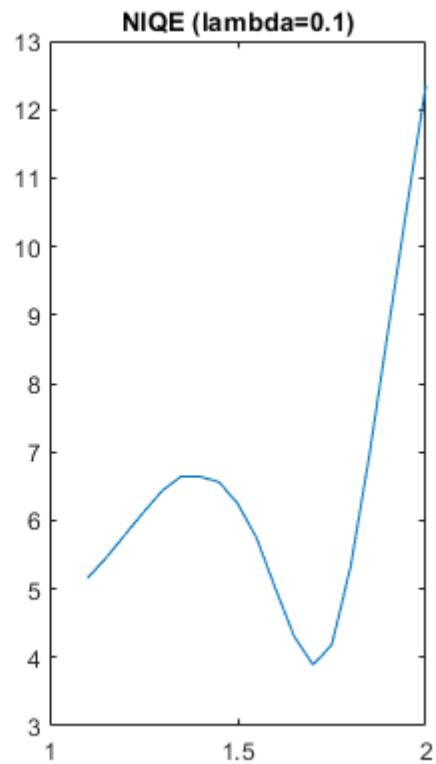
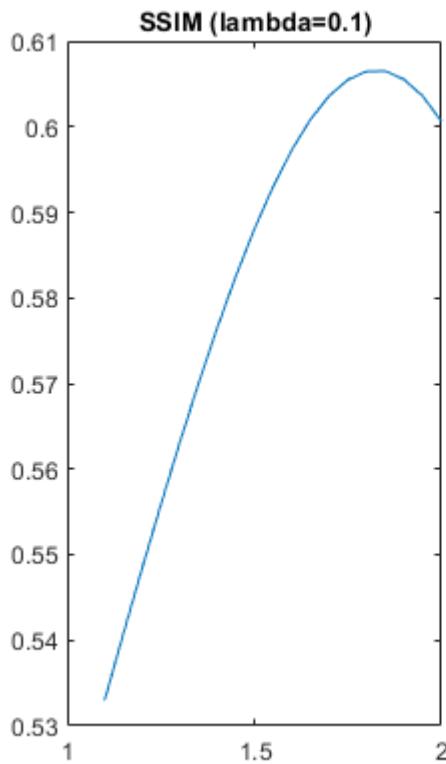
بهترین مقادیر SSIM و NIQE به صورت زیر هستند.

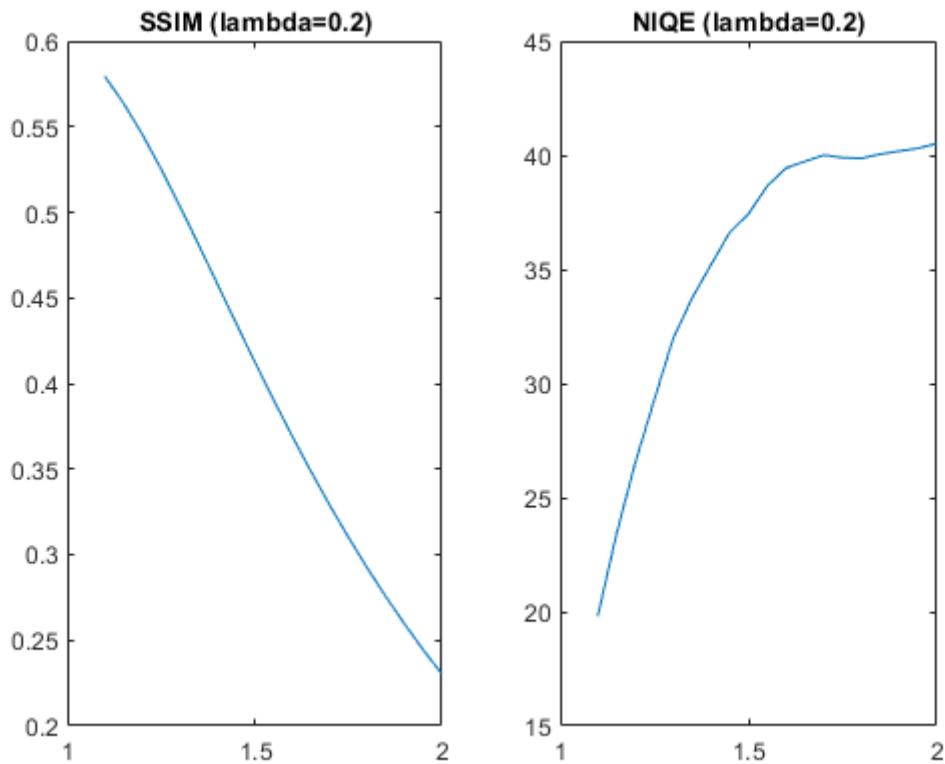
تمرین سری سوم پردازش و تحلیل تصاویر پزشکی

Best SSIM: 0.70, with lambda=0.15, kappa=20.00 and option:1.00
Best NIQE: 4.62, with lambda=0.15, kappa=20.00 and option:1.00

برای تابع isodiff داریم:







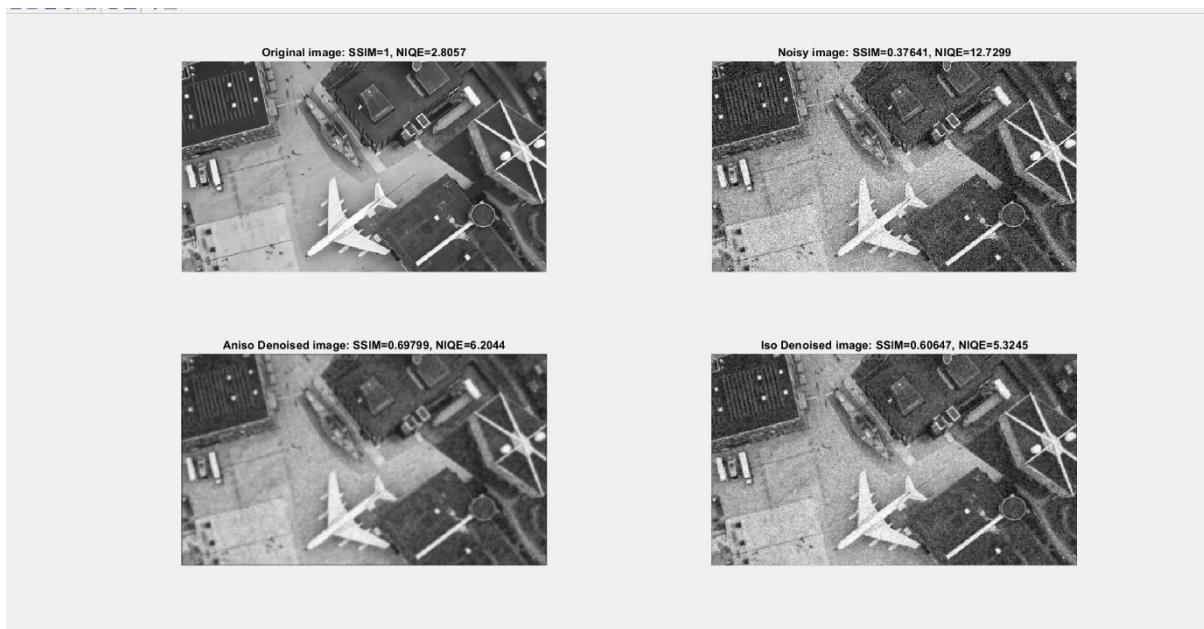
شکل ۲-۹ نمودارهای مقادیر SSIM و NIQE

: بهترین مقادیر SSIM و NIQE

Best SSIM: 0.61, with lambda=0.10 and constant=1.85

Best NIQE: 3.89, with lambda=0.10 and constant=1.85

در آخر در شکل زیر، تصویر اصلی، تصویر نویزی، بهترین تصویر حذف نویز شده با تابع anisodiff و بهترین تصویر حذف نویز شده با تابع isodiff، به همراه مقادیر SSIM و NIQE هر تصویر نشان داده شده‌اند.



شکل ۱۰-۲ تصویر اصلی، تصویر نویزی و تصاویر حذف نویز شده

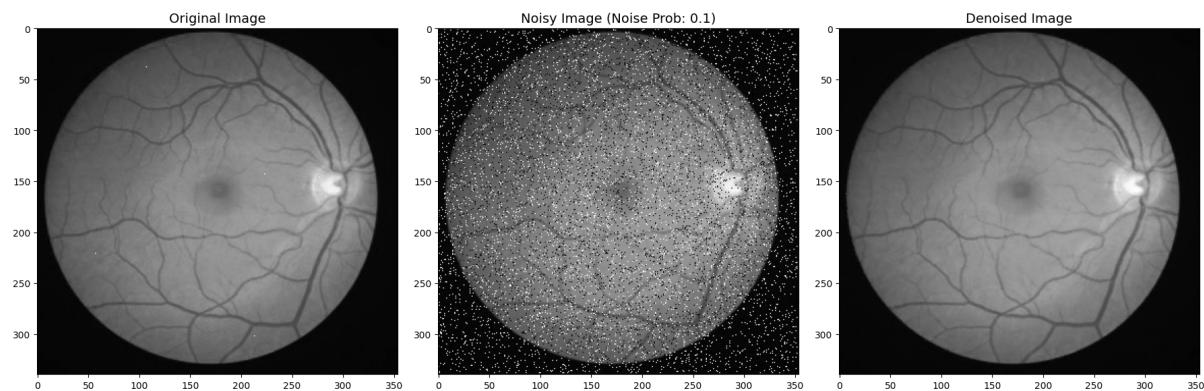
با توجه به شکل بالا، عملکرد الگوریتم Anisodiff بهتر از الگوریتم isodiff است چرا که مقادیر SSIM بیشتر و NIQE کمتری دارد.

۳-۲- سوال سوم

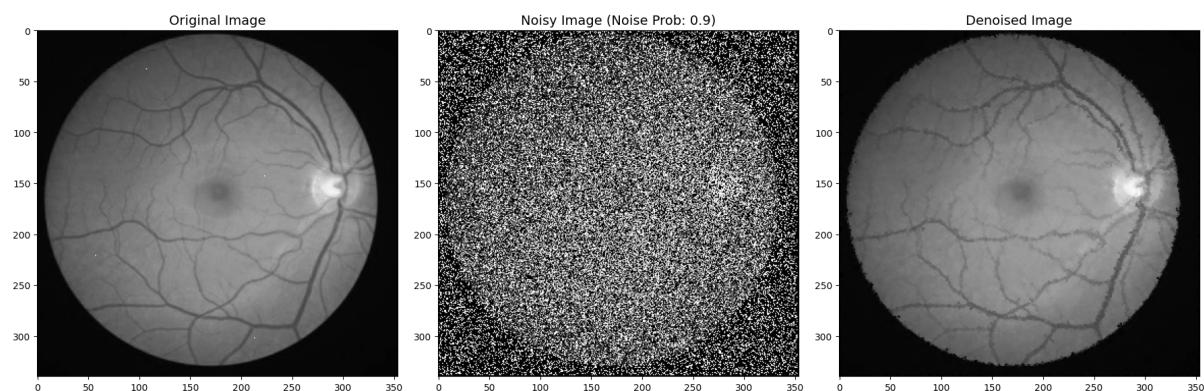
۱-۳-۲- الف

ابتدا تصویر مربوطه بارگذاری شد. پس از بارگذاری، به تصویر نویز نمک و فلفل با مقادیر مختلف اضافه شد. به طوری که یک بار ۱۰٪ پیکسل‌ها نویزی شدند و یک بار ۹۰٪ و این روند تا ۹۰٪ ادامه پیدا کرد. سپس با استفاده از الگوریتم گفته شده تصویر حذف نویز شد. در شکل زیر دو تصویر نمونه برای مقدار نویز ۱۰٪ و ۹۰٪ نشان داده شده است. بقیه‌ی تصاویر در فایل نوتبوک موجود هستند.

تمرین سری سوم پردازش و تحلیل تصاویر پزشکی



شکل ۲-۱۱ تصویر اصلی، تصویر نویزی با مقدار ۱۰٪ و تصویر حذف نویز شده



شکل ۲-۱۲ تصویر اصلی، تصویر نویزی با مقدار ۹۰٪ و تصویر حذف نویز شده

با توجه به مقادیر بالا، مشاهده می‌شود که الگوریتم به خوبی عمل کرده است.

نمودارهای معیارهای سنجش عملکرد پس از الگوریتم AWMF نشان داده شده‌اند.

۲-۳-۲ ب

0	68	255	0	0	70	255
0	255	255	255	255	255	0
0	255	68	67	67	255	0
255	0	255	66	78	255	70
255	0	255	255	255	255	255
0	255	0	255	0	0	0
0	78	0	0	255	255	255

$$S_{\min}=66, S_{\text{med}}=75, S_{\max}=255 \rightarrow Z_{ij}=78$$

در مناطق تصویر که مقادیر شدت نسبتاً یکسان هستند و نویز قابل توجهی وجود ندارد، فیلتر میانه تطبیقی ممکن است مقادیر پیکسل‌ها را به طور غیرضروری تغییر دهد و باعث blur شدن تصویر یا حذف جزئیات شود.

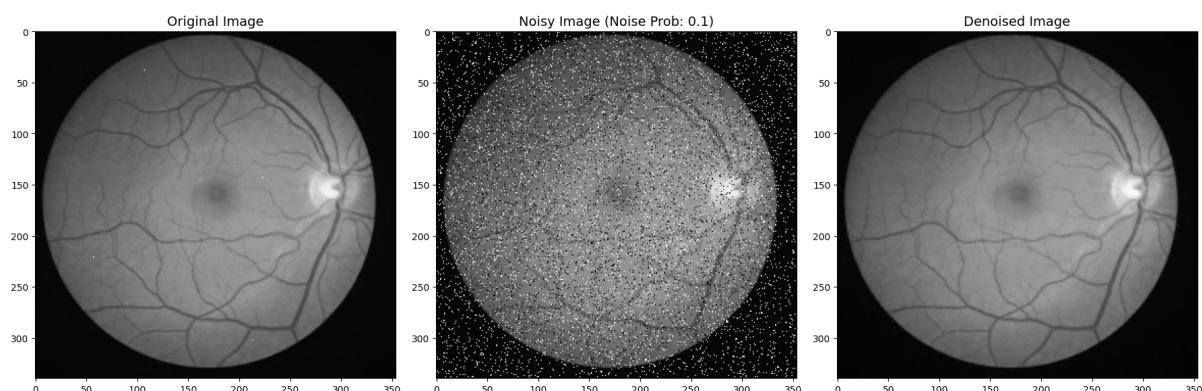
۳-۳-۲-ج

الگوریتم Adaptive Weighted Mean Filter (AWMF) با استفاده از وزن‌دهی به پیکسل‌ها به دقت مقادیر پیکسل‌ها را در مناطقی که مقادیر شدت نسبتاً یکسان هستند، حفظ می‌کند. وزن‌دهی این امکان را فراهم می‌کند که الگوریتم به طور دقیق‌تر تمايل به اعمال تغییر در مقادیر پیکسل‌ها داشته باشد. وزن‌دهی در AWMF بر اساس شباهت مقادیر پیکسل‌ها به پیکسل مرکزی انجام می‌شود. به عبارت دیگر، پیکسل‌هایی که مقدارشان با پیکسل مرکزی شبیه‌تر است، وزن بیشتری در فیلترینگ دارند. این وزن‌دهی باعث می‌شود که در مناطقی از تصویر که مقادیر شدت نسبتاً یکسان هستند، پیکسل‌های نزدیک به پیکسل مرکزی وزن بیشتری داشته باشند. این به این معنی است که تغییرات در این مناطق کمتر اعمال می‌شود و بنابراین جزئیات تصویر حفظ می‌شوند.

نحوه وزن‌دهی باعث می‌شود که اگر تصویر در منطقه‌ای همگن قرارداشت که به نویز آغشته نشده است، پیکسل مرکزی تغییری نکند، چرا که مقدار S^{Mean} برابر با ۱- شود که این مقدار بدون تغییر مقدار پیکسل مرکزی پنجره حرکت کند.

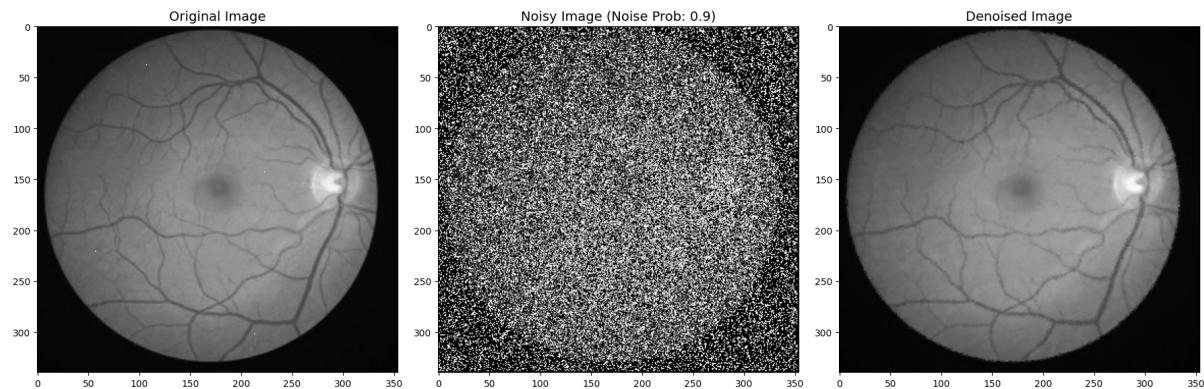
۴-۳-۲-د

در این بخش نیز روند مشابه روند بخش الف طی شد. در شکل زیر تصویر اصلی، نویزی و حذف نویز شده برای مقادیر نویز ۱۰٪ و ۹۰٪ نشان داده شده‌اند.



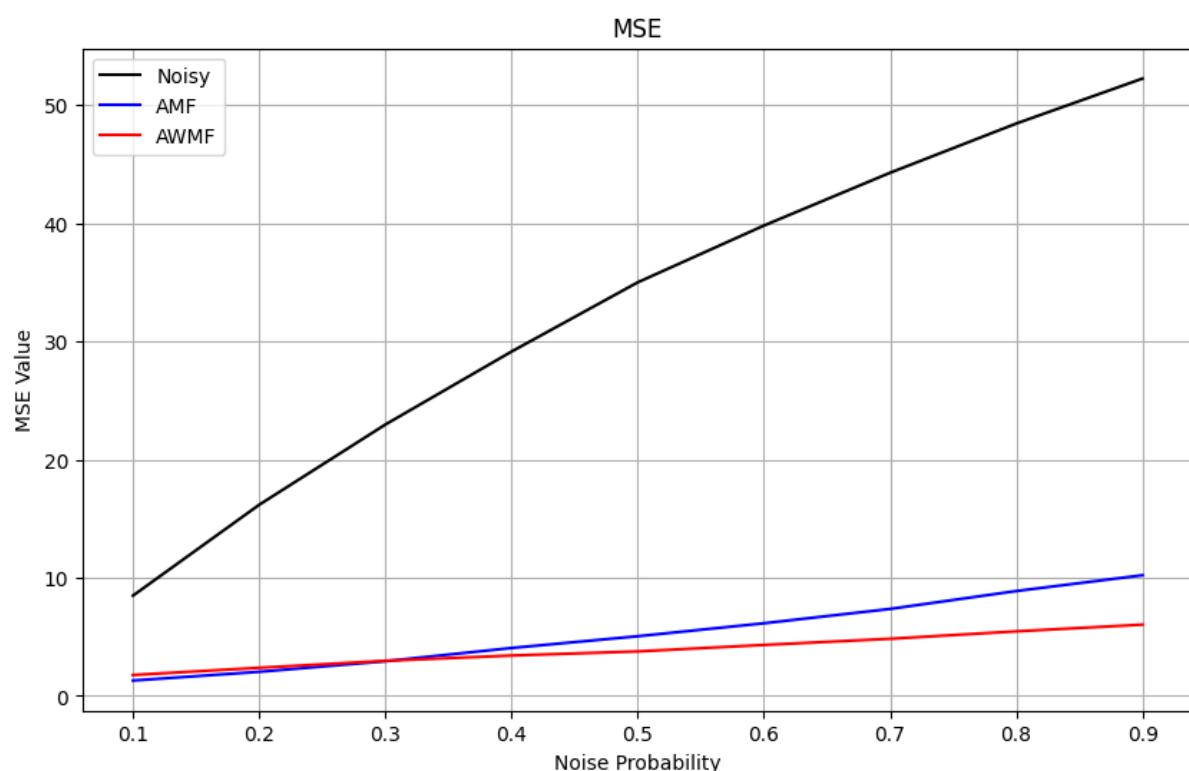
شکل ۱۳-۲-تصویر اصلی، تصویر نویزی با مقدار ۱۰٪ و تصویر حذف نویز شده

تمرین سری سوم پردازش و تحلیل تصاویر پزشکی

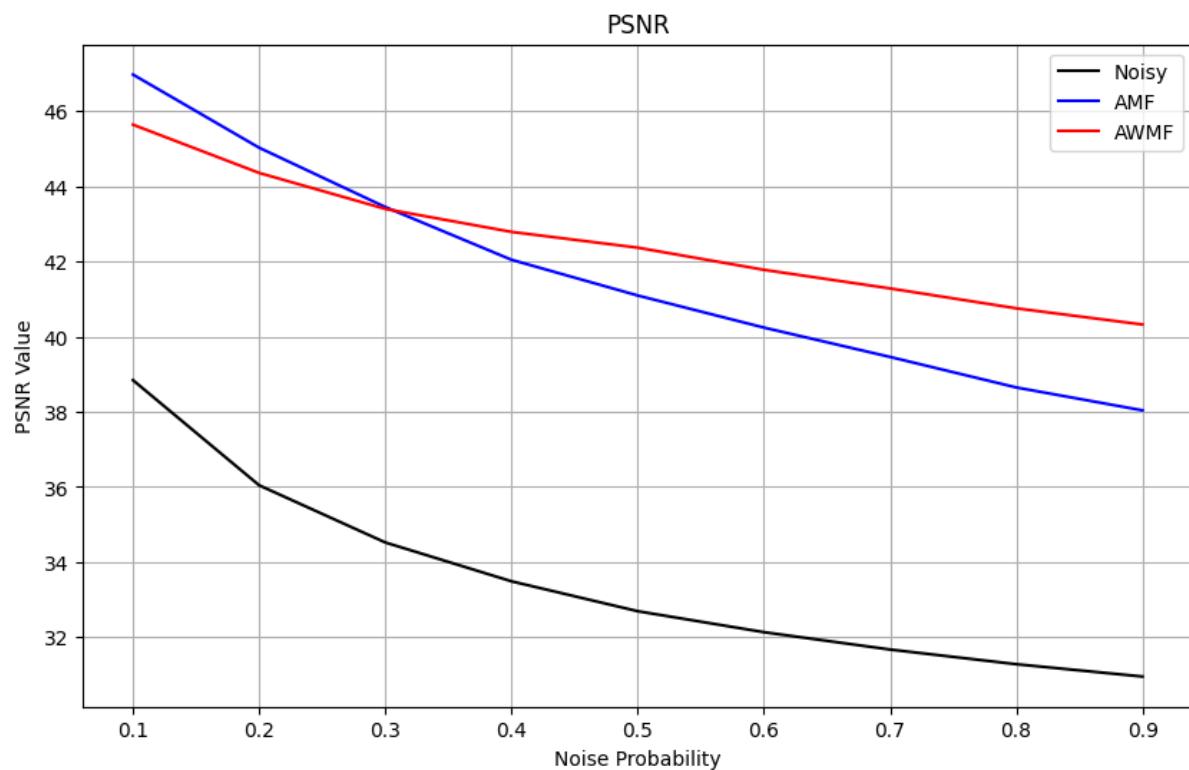


شکل ۱۴-۲ تصویر اصلی، تصویر نویزی با مقدار ۹۰٪ و تصویر حذف نویز شده

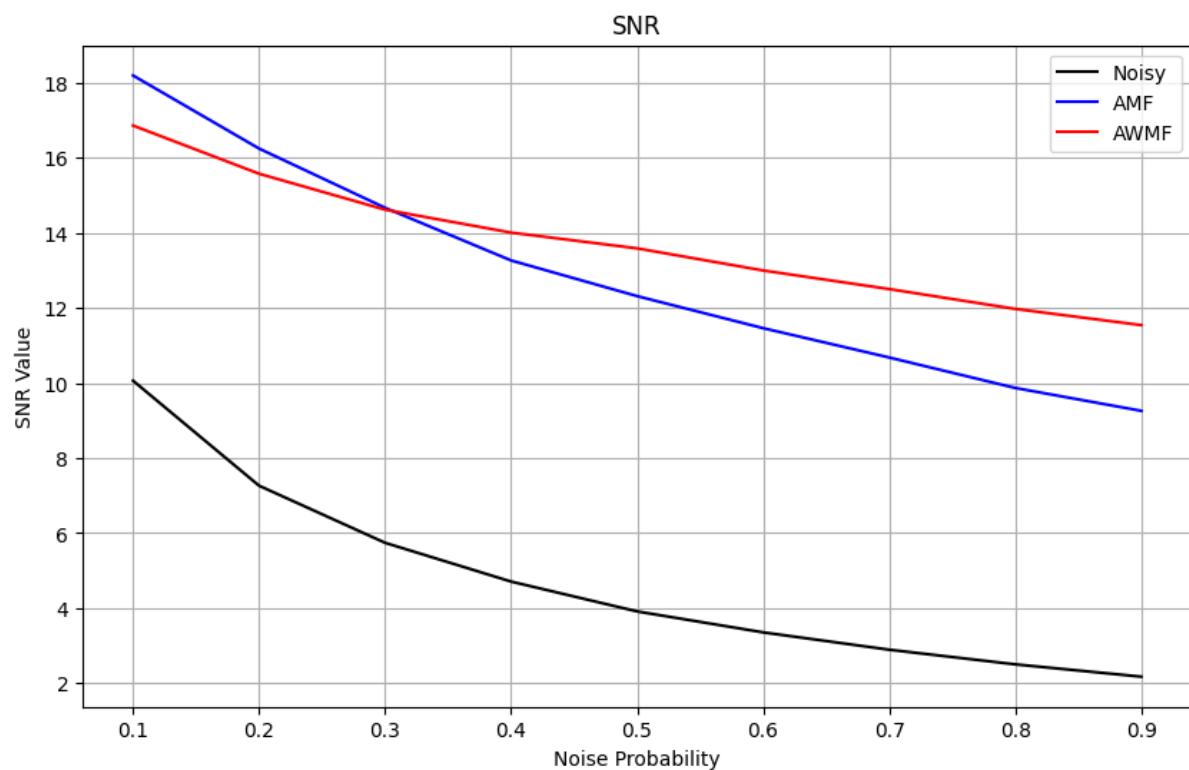
در ادامه نمودارهای مربوط به معیارهای سنجش کیفیت حذف نویز نشان داده شده است.



شکل ۱۵-۲ معیار MSE



شكل ۲-۱۶ معيار PSNR



شكل ۲-۱۷ معيار SNR

با توجه به نمودارهای بالا، با افزایش مقدار نویز، عملکرد فیلترها افت می‌کند که بدیهی است. مقایسه‌ی دو روش AMF و AWMF نشان می‌دهد که روش AWMF در مقادیر نویز بیشتر، بهتر عمل می‌کند.

۴-۲- سوال چهارم

۲-۵- الف

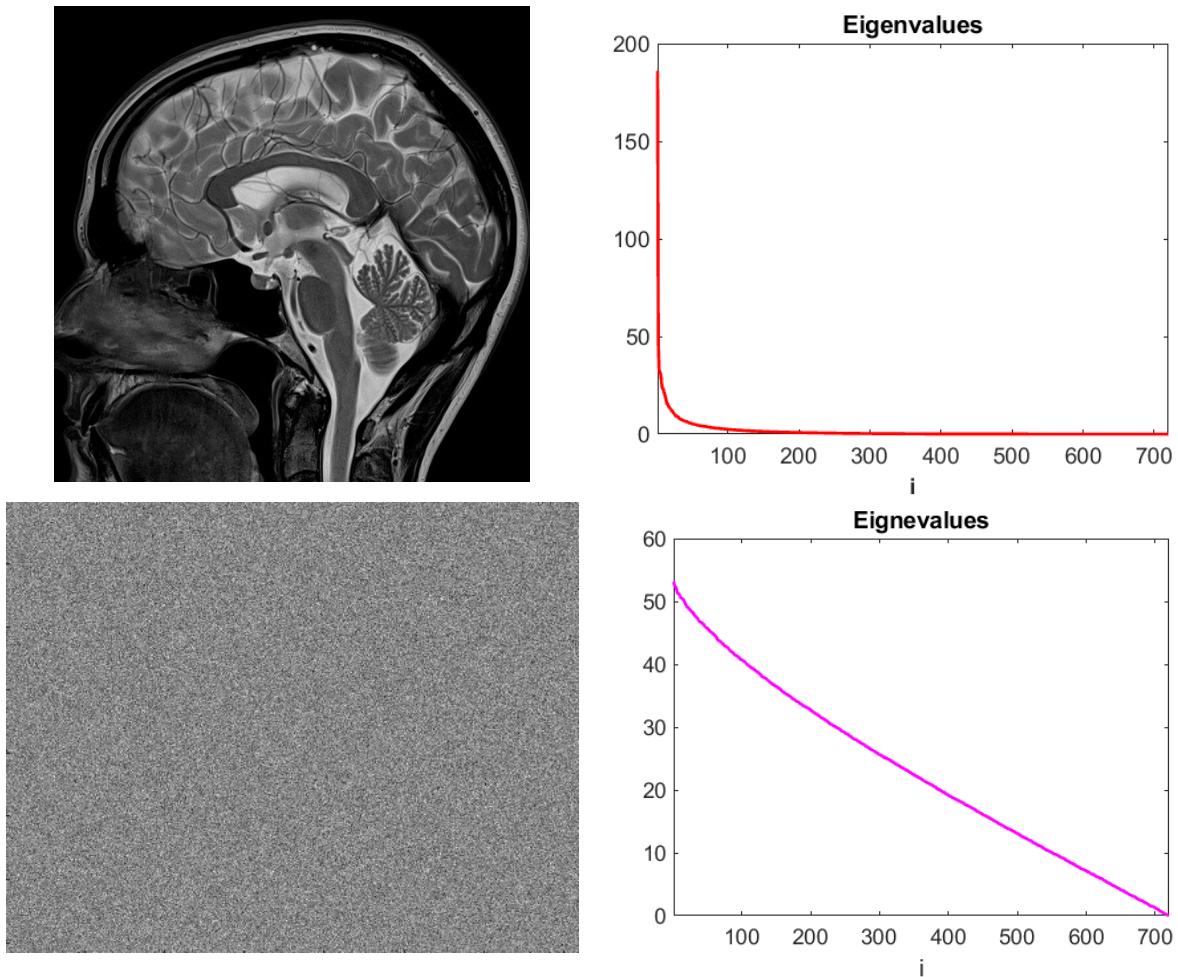
در مسائل بازیابی ماتریس با رتبه کم یا "low rank matrix recovery"، هدف اصلی این است که یک ماتریس از داده‌های مشاهداتی را با توجه به شرایط خاصی که رتبه آن ماتریس از قبل مشخص است، بازیابی کنیم. این مسئله در بسیاری از حوزه‌هایی مانند پردازش تصویر، پردازش سیگنال و یادگیری ماشین مورد استفاده قرار می‌گیرد. وقتی در تصاویر طبیعی با نویز روبرو هستیم، مقادیر ویژه‌های مربوط به تصاویر و نویز به صورت زیر ممکن است تغییر کنند:

۱. **مقادیر ویژه‌ی تصویر طبیعی:** در تصاویر طبیعی، ماتریس کوواریانس (ماتریس هیسنبرگ) معمولاً رتبه‌ی کمی دارد؛ زیرا تصویرها اغلب از تعداد محدودی از ساختارهای مهم تشکیل شده‌اند (مانند لبه‌ها و قطعات بافتی). این به این معنی است که مقادیر ویژه بزرگ در این ماتریس متعلق به ساختارهای اصلی تصویر هستند و مقادیر ویژه کوچک‌تر مربوط به جزئیات کم اهمیت‌تر هستند.

۲. **مقادیر ویژه‌ی نویز:** در تصاویر دارای نویز، مقادیر ویژه بیشتر از مقادیر بزرگ و کوچک متناوب هستند. این به این معنی است که نویز معمولاً باعث افزایش مقادیر ویژه کوچک و کاهش مقادیر ویژه بزرگ می‌شود. این ویژگی نشان‌دهنده‌ی این است که نویز عمدتاً اطلاعات با اهمیت کم را اضافه می‌کند و ساختار اصلی تصویر را از بین می‌برد.

با توجه به این توضیحات، در فرآیند بازیابی تصویر طبیعی از نویز، معمولاً سعی می‌شود که مقادیر ویژه‌های کوچک در ماتریس مشاهداتی کاهش یابند و مقادیر ویژه بزرگ بازسازی شوند تا ساختار اصلی تصویر حفظ شود.

توضیحات فوق در تصویر زیر قابل مشاهده هستند.



در تصویر فوق قابل ملاحظه است که مقادیر ویژه کوچک، اغلب متعلق به نویز هستند.

۱-۵-۲ - ب

آستانه‌گذاری سخت و نرم دو روش مهم در مسائل بهینه‌سازی و Low Rank Matrix Recovery می‌باشند. در اینجا، به مزایای آستانه‌گذاری سخت و نرم در این مسئله می‌پردازیم:

آستانه‌گذاری سخت:

۱. **سادگی پیاده‌سازی:** این روش اغلب به راحتی و سریع قابل پیاده‌سازی است، زیرا شامل استفاده از عملگرهای ساده مقایسه و اندازه‌گیری است.
۲. **پایداری عملکرد:** آستانه‌گذاری سخت معمولاً به صورت پایدار عمل می‌کند و به نتایج قابل اعتماد منجر می‌شود.

۳. عملکرد خوب در موارد با ابعاد بزرگ: این روش به خوبی در مسائلی که ابعاد ماتریس بسیار بزرگ هستند، عمل می‌کند و به صورت مقیاس‌پذیر به مسائلی با اندازه‌های مختلف قابل اعمال است.

آستانه‌گذاری نرم:

۱. ارزش اطلاعاتی بیشتر: آستانه‌گذاری نرم اطلاعات مربوط به ماتریس با رتبه کم را بهتر در نظر می‌گیرد و از این رو به دقت و کیفیت بیشتری در بازیابی منجر می‌شود.

۲. قابلیت انعطاف‌پذیری بیشتر: این روش قابلیت تطبیق با ساختارهای ماتریسی پیچیده‌تر را دارد و می‌تواند در مواردی که ماتریس‌ها از ساختارهای غیرخطی یا ناهمگن برخوردارند، عملکرد خوبی داشته باشد.

۳. کارایی در حضور نویز: آستانه‌گذاری نرم معمولاً در مواردی که داده‌ها دارای نویز باشند، بهبود عملکرد دارد، زیرا قابلیت اصلاح نویز را دارد و می‌تواند اطلاعات معتبر را از نویز جدا کند.

۴-۵-۲ ج

روش (WNNM) Weighted Nuclear Norm Minimization یکی از روش‌های مؤثر برای حذف نویز از تصاویر است. این روش بر پایه‌ی کاهش نرم هسته‌ای وزن دار است که نسخه‌ی تعمیم یافته‌ای از روش کاهش نرم هسته‌ای (NNM) Nuclear Norm Minimization است. در WNNM، به هر مقدار تکین در تجزیه ماتریس، یک وزن مخصوص داده می‌شود که این اجازه را می‌دهد تا کنترل بهتری بر میزان شدت حذف نویز داشته باشیم.

توضیح روش WNNM:

۱. تبدیل تصویر به ماتریس: ابتدا تصویر ورودی به ماتریسی از پچ‌های (قطعات کوچک تصویر) تبدیل می‌شود.
۲. اعمال WNNM: برای حذف نویز، نرم هسته‌ای وزن دار این ماتریس محاسبه می‌شود. این کار شامل محاسبه تجزیه مقادیر تکین (SVD) برای ماتریس و سپس اعمال یکتابع وزن به مقادیر تکین است تا نویز را کاهش دهد.
۳. بازسازی تصویر: ماتریس ویرایش شده به فضای تصویری اصلی بازگردانده می‌شود تا تصویر نهایی بدون نویز حاصل شود.

تفاوت با Low Rank Matrix Recovery (LRMR)

- تمرکز: روش LRMR بر روی بازیابی ماتریسی تمرکز دارد که حداقل رتبه را دارد و می‌تواند برای موارد مختلفی به جز حذف نویز مورد استفاده قرار گیرد، مانند تکمیل ماتریس یا تشخیص داده‌های پرت. اما WNNM به طور خاص برای حذف نویز طراحی شده است.

- فلسفه وزن‌دهی: در LRMR، هدف بازیابی ماتریس با کمترین رتبه ممکن است، بدون تمایز قائل شدن بین مقادیر تکین مختلف. در مقابل، WNNM به طور خاص وزن‌ها را برای مقادیر تکین اعمال می‌کند تا نویز را به شکل مؤثرتری حذف کند، بنابراین می‌تواند در مواردی که نویز گستردگی‌تر و پیچیده‌تر است، کارآمدتر باشد.

۵-۳-۲

** در این بخش برای اطمینان نتیجه‌ی به دست آمده از الگوریتم پیاده‌سازی شده، خروجی خود را با خروجی الگوریتم نوشته شده برای مقاله‌ی زیر مقایسه می‌کنم.

http://stanford.edu/class/ee367/Winter2022/report/kanavalau_report.pdf

لینک دسترسی به کد:

https://github.com/kanavalau/EE367_Project_WNNM_denoising

فایل استفاده از این کد با نام Stanfor.ipynb در پوشه‌ی سوال ۴ قرار دارد.

در این سوال ابتدا الگوریتم توضیح داده شده در اسلایدها، پیاده سازی شد.تابع تعریف شده ورودی‌های زیر را می‌گیرد.

```
def WNNM(img, patchSize, c, K, delta, sigma, nThreshold):
```

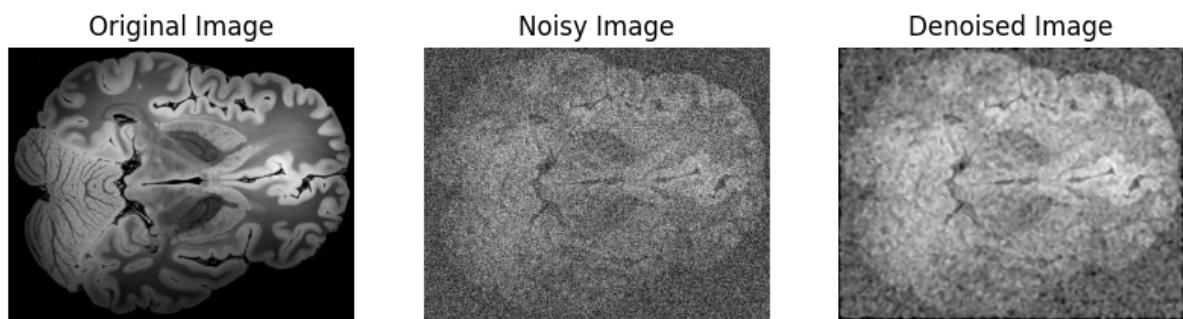
:تصویر،patchSize:نصف اندازه‌ی پچ‌ها،C ضریب برای تولید وزن‌ها،K تعداد تکرار الگوریتم،Sigma انحراف معیار نویز و nThreshold تعداد پچ‌های شبیه. همچنین سایز پنجره‌ی سرج برای یافتن پچ‌های شبیه، ۳ برابر اندازه‌ی پچ است.

پارامترهای ورودی به تابع برای حذف نویز به صورت زیر هستند:

```
patchSize = 3
delta = 0.1
c = 2.8
K = 3
sigma = 1
nThreshold = 35
```

مقدار K مقدار کوچکی در نظر گرفته شد چرا که با مقادیر بزرگ فرایند حذف نویز مدت زمان زیادی به طول می‌انجامید (با این مقدار نیز فرایند حدود ۲ ساعت طول کشید). بقیه‌ی مقادیر نیز از مقاله‌ی پیوست به دست آمدند.

پس از حذف نویز، خروجی به صورت زیر به دست آمد.



شکل ۲-۱۸ خروجی به دست آمده

همچنین مقدار PSNR و SSIM ها به صورت زیر هستند:

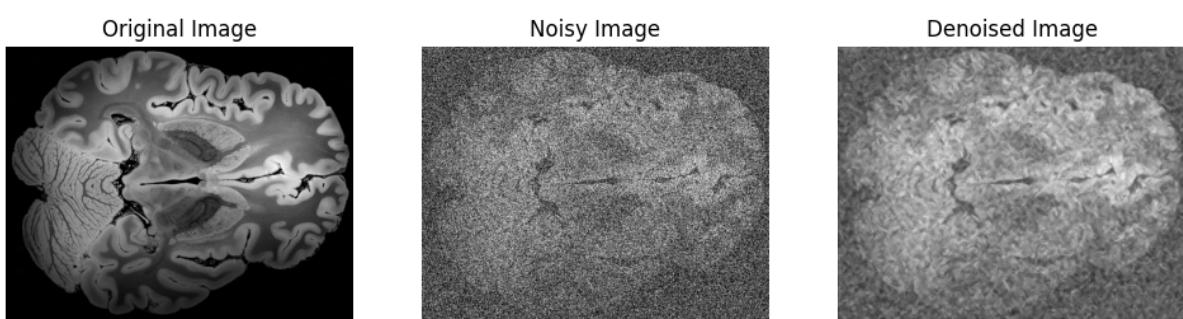
SSIM Denoised: 0.4178

PSNR Denoised: 14.58 dB

SSIM Noisy: 0.0243

PSNR Noisy: 6.69 dB

ملاحظه می‌شود که مقدار SSIM حدود ۲۰ برابر بهتر شده است همچنین مقدار PSNR نیز ۸ دسیبل افزایش یافته است. با توجه به مقدار بسیار بالای نویز اضافه شده، اعداد به دست آمده قابل قبول هستند. در تصویر زیر خروجی کد مربوط به مقاله‌ی Stanford نشان داده شده است.



SSIM Denoised: 0.3987

PSNR Denoised: 13.42 dB

SSIM Noisy: 0.0239

PSNR Noisy: 6.66 dB

تمرین سری سوم پردازش و تحلیل تصاویر پزشکی

همانطور که ملاحظه می‌شود، برای این مقدار نویز، اعداد الگوریتم نوشته شده با الگوریتم مربوط به مقاله‌ی تقریباً برابر هستند.

*** از الگوریتم مقاله صرفاً به عنوان مرجع برای سنجش خروجی‌ها استفاده شده است.