

به نام خدا

تمرین سری اول درس MIAP

محمد رضیئی فیجانی 98206223

## گیت هاب

پاسخ های این تمرین در لینک زیر نیز در گیت هاب قرار داده شده است که می توان تغییرات کد را در آن مشاهده نمود.

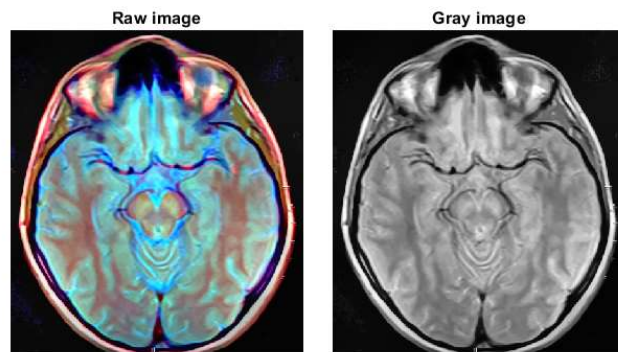
<https://github.com/MohammadRaziei/MIAP-Course/tree/master/HW01/>

## پاسخ سوالات

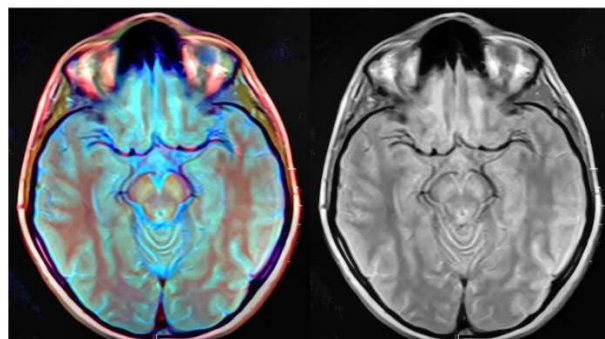
سوال ۱:

قسمت اول:

در این قسمت ابتدا با استفاده از تبدیل `rgb2gray` تصویر خام ورودی را به تصویر خاکستری تبدیل می کنیم. برای نمایش نیز این کار به دو صورت استفاده شده است. روش اول با استفاده از `imshow` می باشد که در شکل زیر خروجی آن آمده است.



و همچنین با استفاده از دستور `montage` که در شکل زیر نمایش داده شده است.



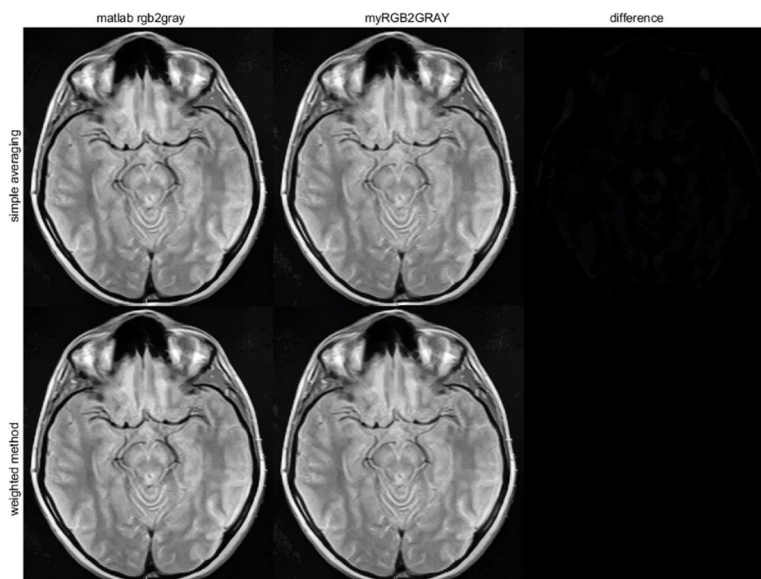
### قسمت دوم:

تصاویر `gray_scale` و `RGB` که در تعداد کانال ها بایکدیگر متفاوت هستند که به ترتیب ۱ و ۳ کانال دارند. `binary`, `uint8`, `uint16` میزان تعداد بیت های هر المان تصویر را نشان میدهد که به ترتیب 1 و 8 و 16 بیت می باشد. فرمت `double` هم که از استاندارد IEEE برای محاسبات اعشاری تبعیت می کند.

برای تبدیل یک تصویر رنگی به تصویر سیاه سفید سه راه معمولاً استفاده میشود.

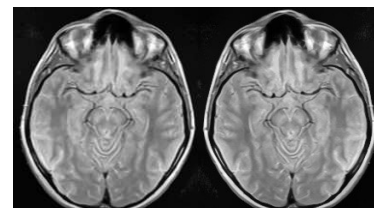
- 1) استفاده از صرفاً یک کانال که معمولاً کانال سبز را برای آن انتخاب میکنند.
- 2) میانگین گیری روی تمامی کانال ها
- 3) میانگین گیری وزن دار روی کانال ها که در متلب به صورت زیر محاسبه می شود.
$$0.2989 * R + 0.5870 * G + 0.1140 * B$$

تابع `myRGB2GRAY` روش دوم و سوم را پیاده سازی نموده است. که شکل زیر هر دو پیاده سازی را در مقایسه با `rgb2gray` خود نمایش داده است. که نشان می دهد متلب از همان روش وزن دار استفاده می کند.



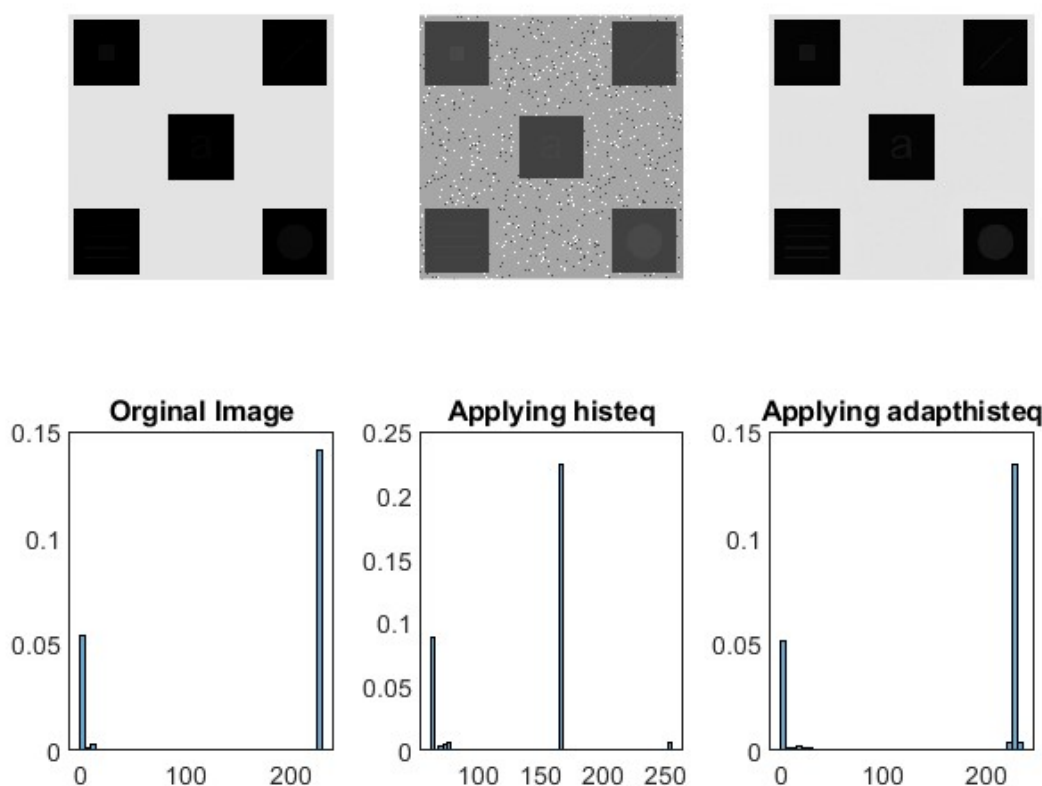
### قسمت سوم:

با استفاده از دستور `imwrite` تصاویر زیر که از دو طریق مختلف به تصویرهای خاکستری تبدیل شده اند را در فرمت `png` ذخیره شده اند. (البته تمام تصاویر `plot` شده و `figure` های این تمرین نیز با استفاده از همان دستور در همان فرمت ذخیره شده است.)



## سوال دوم:

خروجی این سوال در شکل زیر آمده است که مشاهده می‌شود که استفاده از histeq به صورت گلوبال باعث شده است تا الگوهایی مانند نویز فلکل نمکی در تصویر ظاهر شود اما استفاده از adapthisteq باعث بهبود تصویر خروجی و واضح تر شدن الگوهای مخفی شده در مربع های سیاه رنگ شده است.



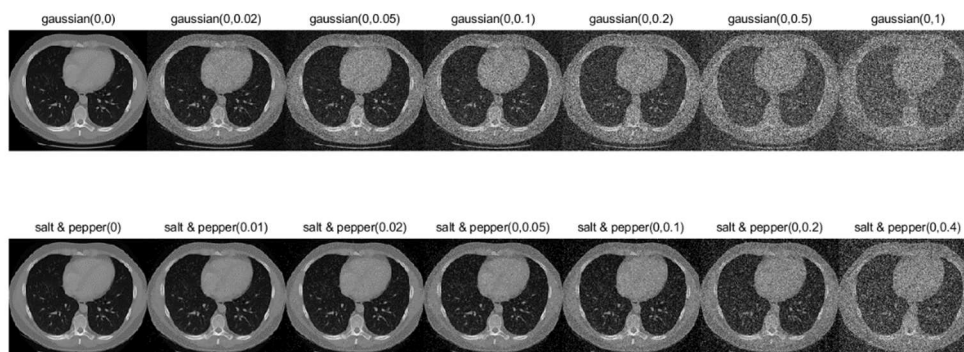
توضیح و تفاوت این روش را به خوبی در درس و لینک زیر توضیح داده شده اند.

<https://stackoverflow.com/questions/26818568/whats-the-difference-between-histeq-and-adapthisteq>

اما اگر بخواهیم مختصری در مورد آن توضیح دهیم میتوان گفت که در histeq ابتدا با استفاده از histogram تصویر تخمینی از pdf آن بدست می‌آورند. یک فرضی که در این روش وجود دارد آن است که عموماً تصاویری با توزیع یکنواخت، تنوع بیشتری از سطوح رنگی را استفاده میکند و همین امر باعث بهبود تصویر برای انسان می‌شود. در این روش پس از آن که cdf را از روی تخمین pdf (با یک سامیشتن) بدست آوردند، نقاط را از آن تابع می‌گذرانند و میتوان ثابت کرد که هر متغیر تصادفی ای که از تابع توزیع تجمعی خودش بگذرد، توزیع یکواخت پیدا میکند. اما این عملیات روی کل تصویر انجام میشود. در عملیات adapthisteq بجای کل تصویر در همسایگی های کوچکی این عملیات انجام میشود. در روش histeq امکان تقویت نویز وجود دارد(همانطوری که در این سوال نیز اتفاق افتاد) اما در روش adapthisteq کنتراست بسته به نویز محدود میشود.

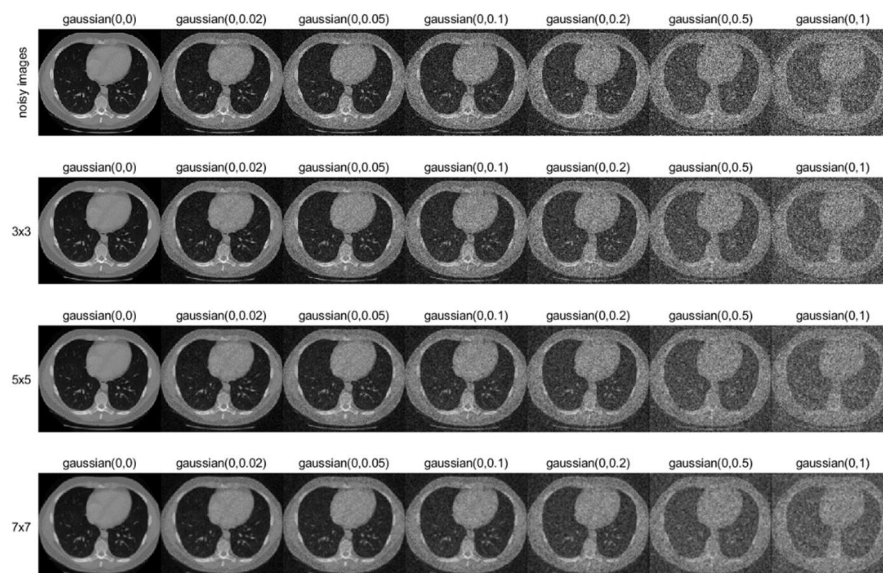
### سوال سوم:

در این سوال ابتدا با استفاده از دستور imnoise مقداری نویز به تصویر اضافه شده است. در نویز گوسی میانگین صفر و واریانس ها متنوع و در نویز فلقل-نمکی، احتمال رخ دادن نویز تغییر کرده است. در شکل زیر این موارد به تصویر کشیده شده اند. (اولین تصویرها در سمت چپ بدون نویز می باشند).



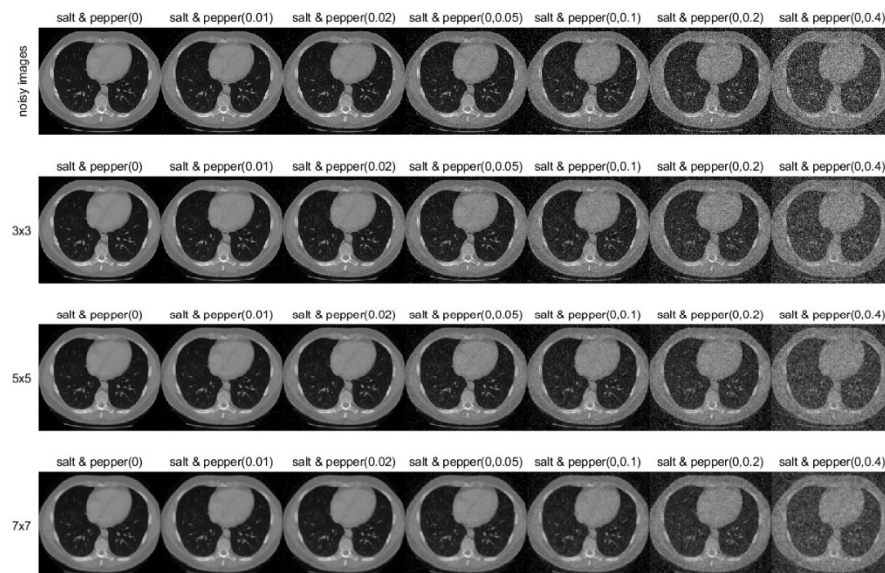
سپس با استفاده از یک فیلتر مربعی با درایه های یکسان و مجموع یک و با سایز های 3 و 5 و 7 بر روی تصاویر بالا اعمال شده است.

در تصاویری که نویز میانگین صفر گوسی داریم این فیلتر ها به شکل زیر عمل کرده اند.

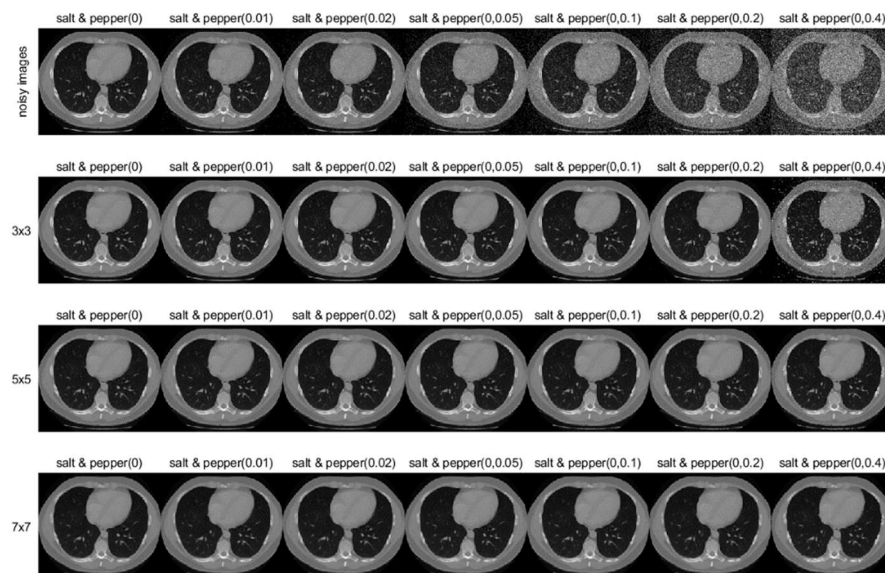


مشاهده میشود که هرچقدر طول پنجره فیلتر بزرگتر میشود قدرت دینویز کردن آن نیز افزایش یافته است. این مسئله به سادگی از دقت در تصاویر بالا به خصوص در توان های بالاتر نویز قابل مشاهده است. اما در عوض پدیده محو شدگی آن نیز با افزایش طول پنجره فیلتر، افزایش یافته است که مطلوب نیست و نوعی trade-off ایجاد شده است.

برای نویز های فلفل-نمکی نیز اعمال آن فیلتر ها به صورت زیر خواهد بود.



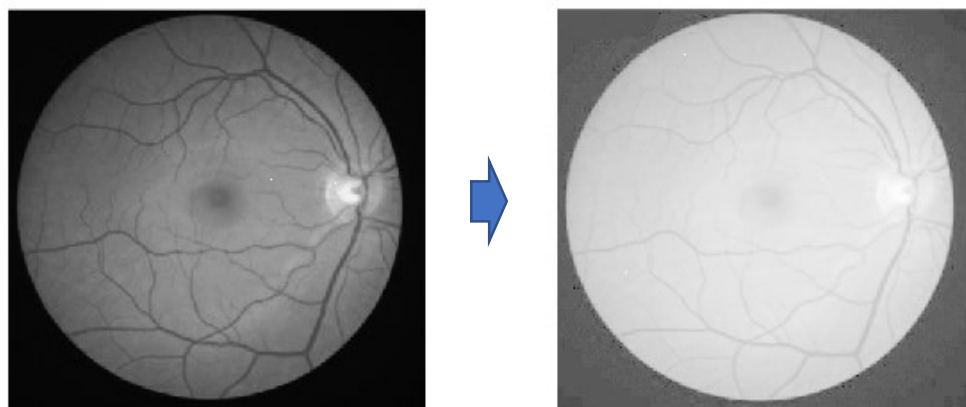
همانگونه که مشاهده میشود تا حدودی این فیلتر سعی در از بین بردن تاثیر نویز داشته است اما اعمال آن بر روی تصاویر با نویز گوسی که میانگین صفر دارند، موثرتر بوده است. همچنین پدیده محوشدگی نیز مانند حالت قبلی در این حالت اتفاق افتاده است. همچنین در این سوال تاثیر فیلتر median که مناسب این نویز است مورد بررسی قرار گرفته است.



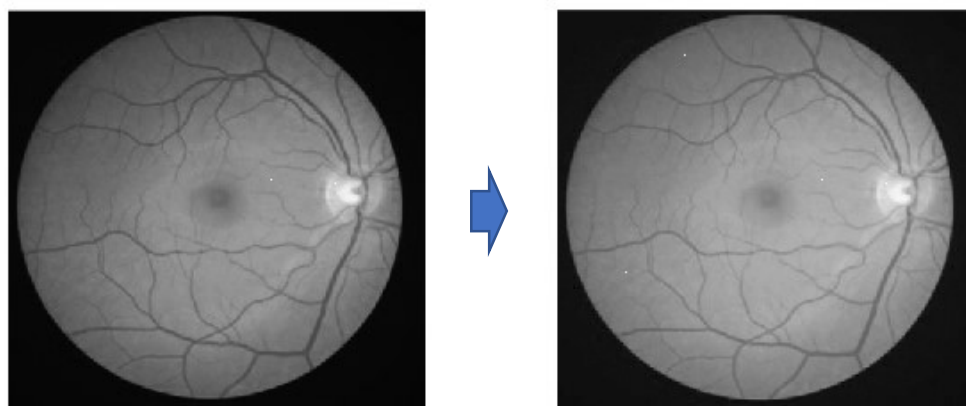
که در پنجره 7 تایی به شکل قابل ملاحظه ای موفق عمل کرده است.

#### سوال چهارم:

در این سوال ابتدا تبدیل لگاریتم اعمال میشود که باعث روشن شدن بسیار تصویر میشود.



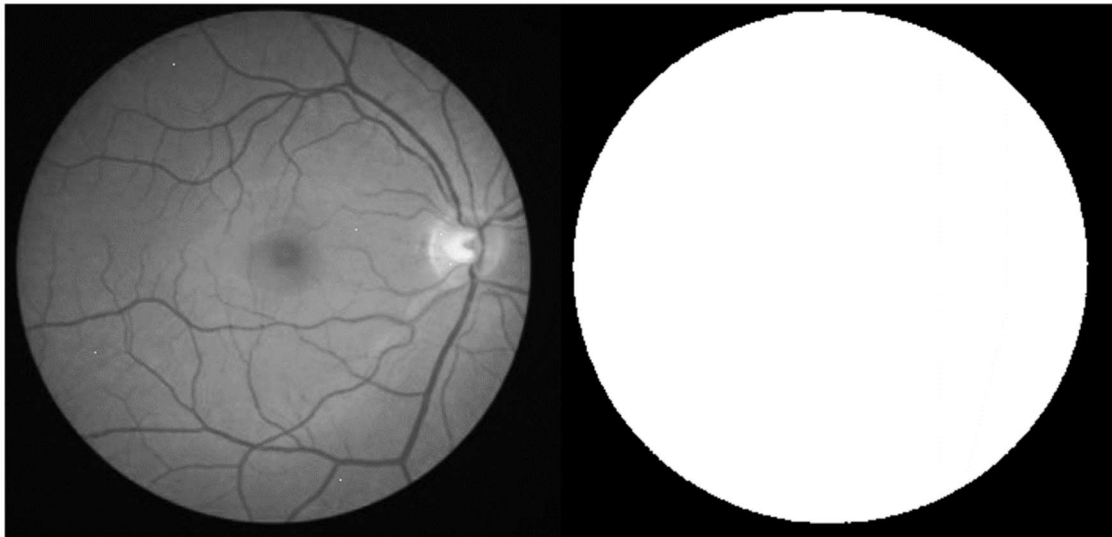
و پس از اعمال قانون power-law با عدد 0.75 به شکل زیر تبدیل میشود. از آنجایی که تصویر عموماً تیره است مقداری کوچکتر از یک برای آن انتخاب شده است.



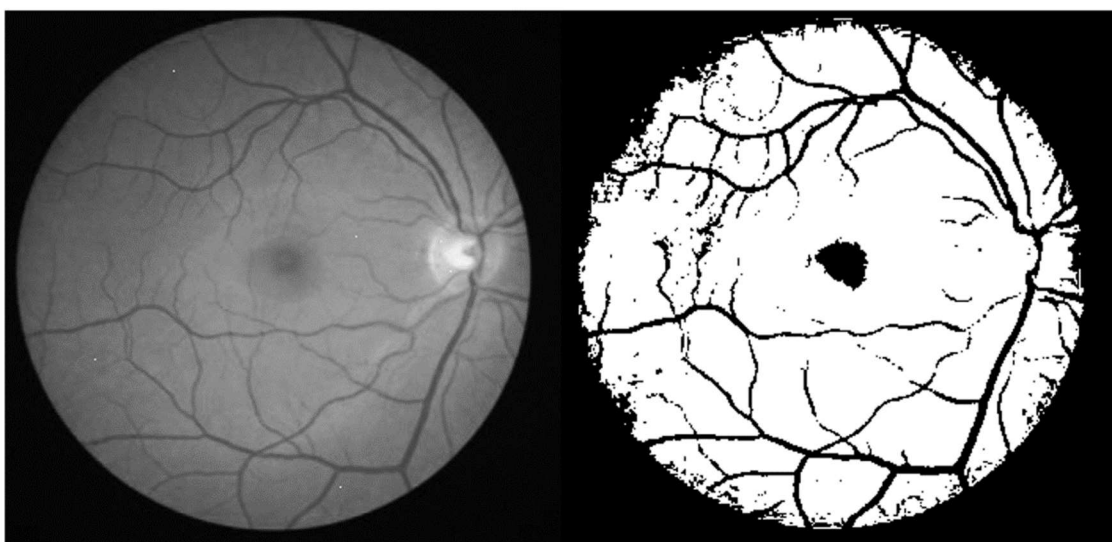
#### سوال پنجم:

در این سوال برای من ابهامی در صورت سوال وجود داشت برای همین هردو حالت آن را بررسی کردم. در حالت اول فرض من از سوال آن بود که دایره وسط را بدون هیچ گونه رگ یا رشته عصبی در بیاورد این کار با ترشولد 0.15 بدون مشکل انجام شد. که در شکل زیر آمده است.





اما حالت دوم این بود که در تصویر فوق رگ ها نیز مشخص باشد. برای این کار از چند لایه تبدیل استفاده شده است که در شکل زیر نتایج آن تبدیلات را مشخص میکند.



از آنجایی که میخواهیم ترشولد بزنیم اعمال یک تابع غیرخطی صعودی کمکی نمی‌کند زیرا که صرفاً مقدار ترشولد را جابجا میکند و تاثیری در ترتیب سطوح قسمت های مختلف تصویر ندارد. بنابراین الگوریتم های `histeq` و `power-law` و لگاریتم و ترکیبشان کمکی نخواهد کرد. اما استفاده از یک الگوریتم که به صورت ادپتیو و محلی تغییر میکند مانند `adapthisteq` کمک کننده خواهد بود. برای این کار ابتدا تصویر را از `adapthisteq` میگذارنیم و سپس یک تابع غیرخطی مانند سیگموئید بر روی آن اثر میدهیم و دوباره `adapthisteq`

میزنیم و همینجا میتوانیم که ترشولد بزنینم. که اتفاقا در همانجا نیز زده شد ولی بهتر بود قبلا از آن با تبدیل power-law روشنایی تصویر را کاهش میدادیم و سپس ترشولد مورد نظر را اعمال میکردیم که حساسیتش کمتر شده باشد.

پایان