تمرین شماره 2

درس پردازش و تحلیل تصاویر پزشکی

استاد درس : دکتر فاطمی‌زاده

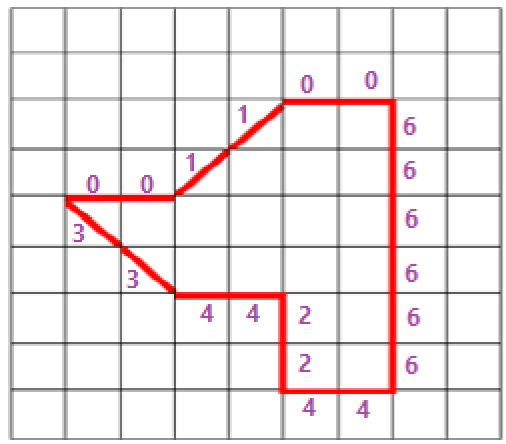
نام : محمد سینا حسن نیا

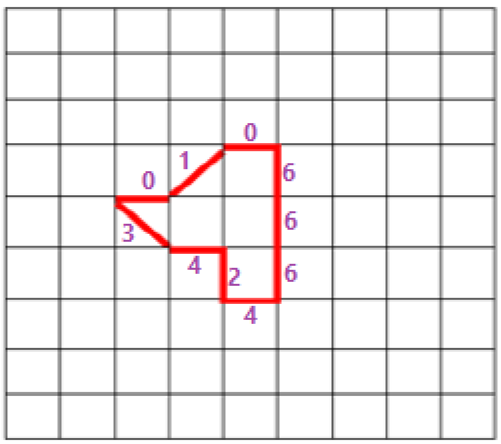
شماره دانشجویی : 96108515

بخش تئوری

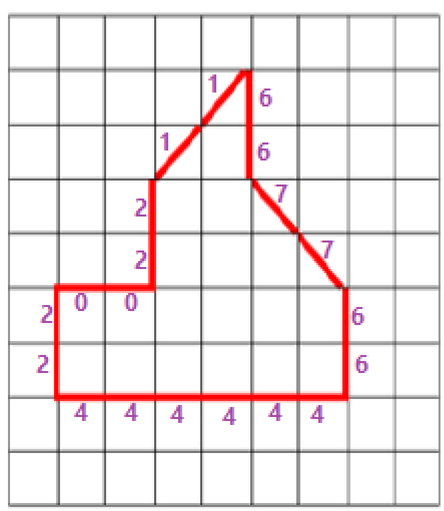
سوال 1

الف) برای این سوال داریم :





در مورد rotation باید گفت اگر تصویر را 90 درجه در سمت چپ rotate کنیم تصویر به دست آمده به صورت زیر می شود که همانطور که دیده می شود مقدار h(7) با یکدیگر تفاوت دارد و در شکل پایین برابر 2 و در بالا صفر است بنابراین عدد جدید در هیستوگرام ظاهر می شود.



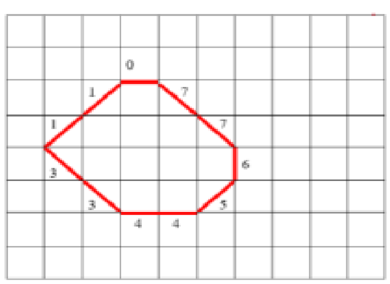
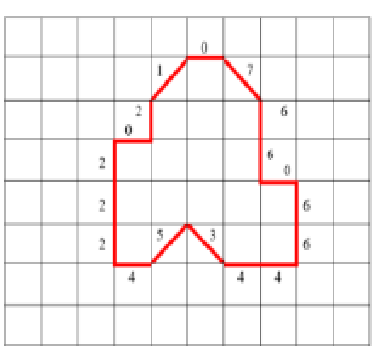
برای scaling اثر آن به این صورت است که هر کدام از binها در scale ضرب می شوند. بنابرین داریم :

$$h_s(i) = h(i) \times scale$$

برای starting point، تغییر ، در کد گذاری تاثیری ندارد زیرا اعداد مشخص شده اگر نقطه شروع تغییر کند عوض نمی شوند و کاری که ما انجام می دهیم شمارش تکرار هر کدام از اعداد است.

1-ب: در این روش scale به نقطه اولیه وابسته نیست و مستقل است بنابراین بهتر است. به این نکته باید توجه کنیم که با در نظر داشتن نقطه اولیه در chain code ساده باید محاسباتی برای min integer code انجام شود.

ب) برای این قسمت داریم :

ردیف	شکل	کد
1	 convex	07765443311
2	 Non_convex	0766066443542220210

همانطور که می بینیم برای convex اگر رقم اول از سمت چپ را کنار بگذاریم نزولی می شود در حالی که برای non-convex این گونه نیست.

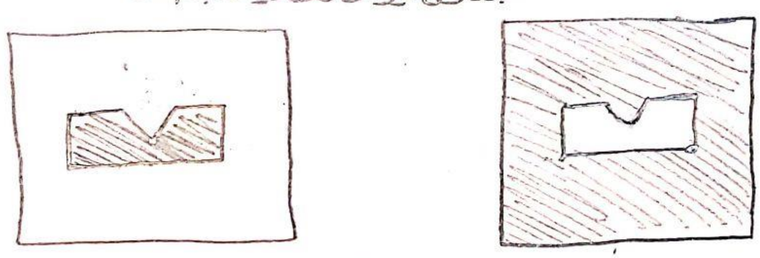
سوال 2

با استفاده از مفهوم erosion و dilation می دانیم:

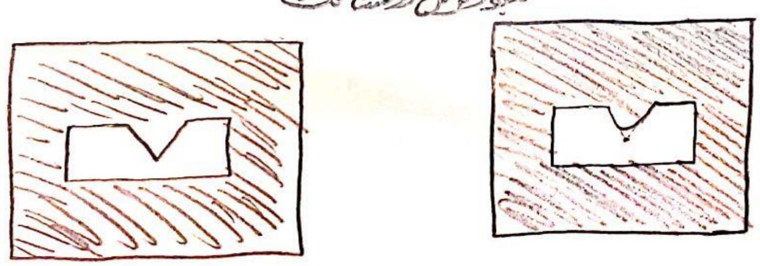
$$\begin{cases} (A \oplus B)^C = A^C \ominus \hat{B} \\ (A \ominus B)^C = A^C \oplus \hat{B} \end{cases}$$

$$(A \circ B)^C = \left((A \oplus B) \ominus B \right)^C = (A \oplus B)^C \oplus \hat{B} = \left(A^C \ominus \hat{B} \right) \oplus \hat{B} = A^C \circ \hat{B} \overset{\text{تقارن}}{=\!=} A^C \circ B$$

عبور کرنل دایره ای روی محیط و بیرون متمم



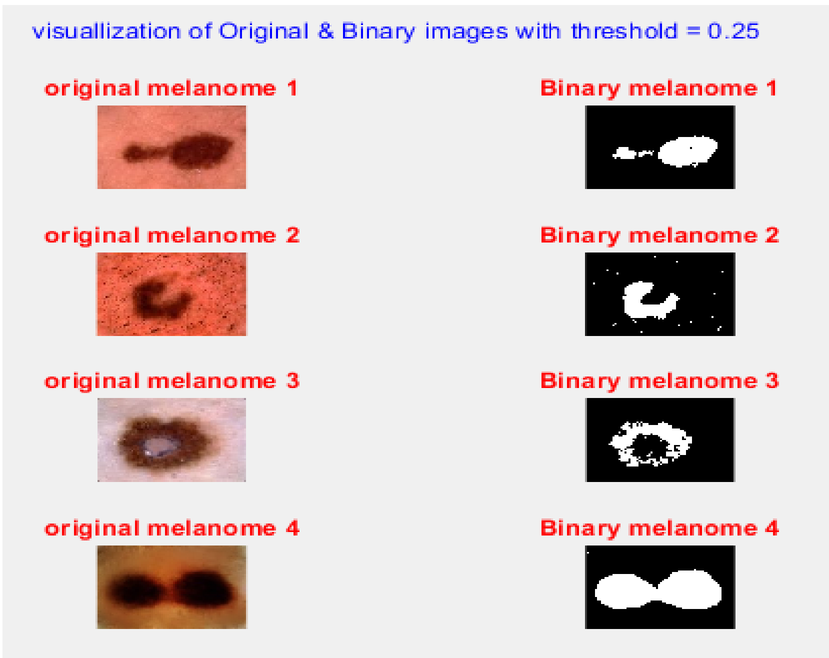
عبور کرنل در سطح متمم



بخش شبیه‌سازی

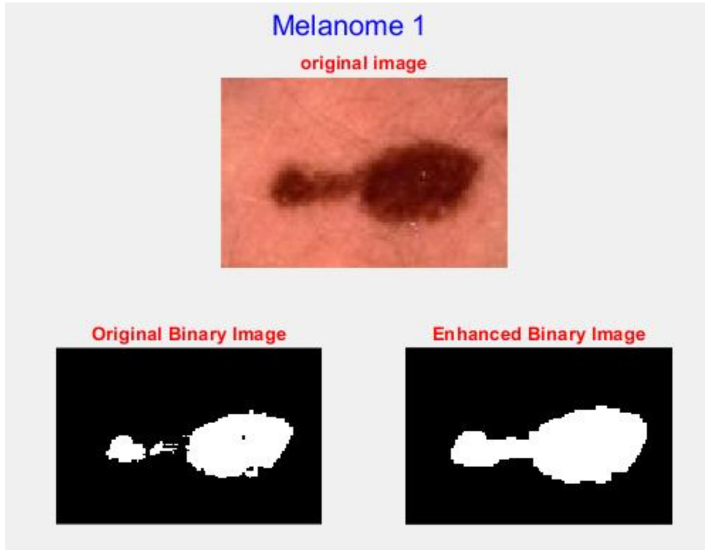
سوال 1

پاسخ: برای این قسمت هر **4** تصویر را می خوانیم و با ترشولد گذاری **0.25** به تصاویر باینری تبدیل می کنیم. داریم :

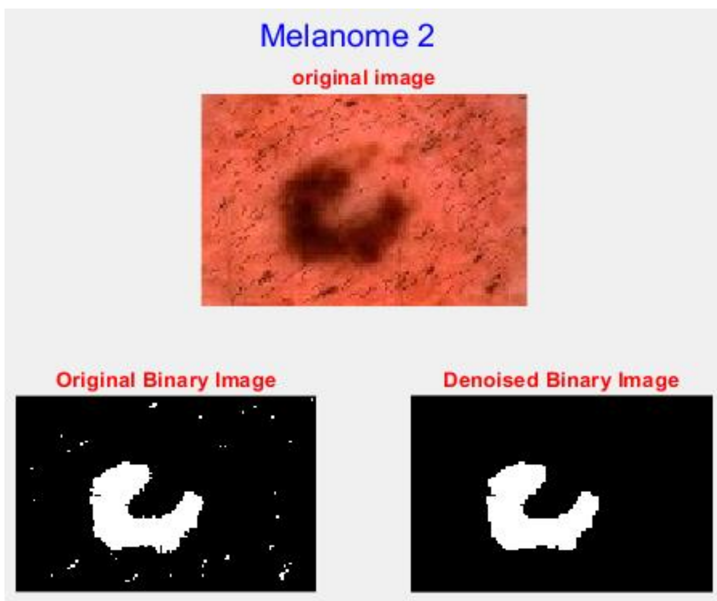


الف) برای این قسمت الگوریتم به این شکل است که که ابتدا یک dilation انجام می دهیم. در واقع با یک کرنل مناسب به نوعی شکل را بزرگ می کنیم تا این ناپیوستگی به هم پیوسته شود. با این کار ممکن است دور شکل به شکل بیش از حد بزرگ شود حال برای آن که این مشکل را حل کنیم با انتخاب یک کرنل مناسب از erosion می کنیم. (شکل را به نوعی باظرافت تر می کنیم (حالت سمباده زدن)) بنابراین تصویر به صورت زیر می شود. همه کرنل ها نیز مربعی می باشند که به صورت زیر می باشد:

Kernel	type	width
1	square	5
2	square	3

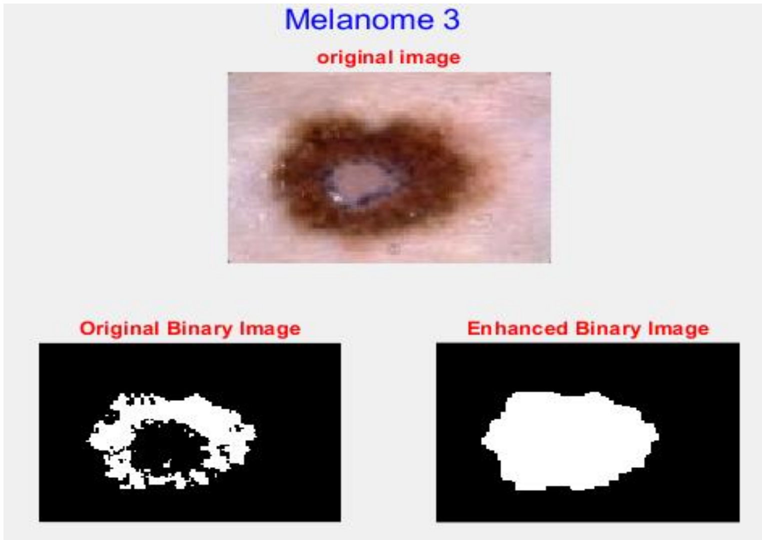


ب) برای این قسمت الگوریتم به این شکل است که با یک کرنل مناسب erosion انجام می دهیم تا این نقطه های سفید رنگ(نویزها) از بین بروند. برای جلوگیری از کوچک شدن شکل با استفاده از کرنل مناسب از dilation استفاده می کنیم. بنابراین تصویر به صورت زیر می شود: (همه کرنل ها نیز مربعی می باشند)



ج) برای این قسمت الگوریتم به این شکل است که ابتدا یک dilation اننجام می دهیم تا شکل به هم پیوسته شود چون شکل ما ناپیوستگی دارد.(شکل را بزرگ می کنیم تا این ناپیوستگی ها از بین برود) . حال با استفاده از closing تو رفتگی های بی‌جای شکل را رفع می کنیم. با استفاده از imfill می آییم و حفره ها را پر می کنیم. در انتها نیز با انتخاب یک کرنل مناسب عملیات erosion را انجام می دهیم. بنابراین تصویر به صورت زیر می

شود: (همه کرنل ها نیز مربعی می باشند). اگر از دستور آماده استفاده نمی کردیم باید مطابق تصویر زیر الگوریتمی را استفاده می کردیم که ابتدا یک نقطه را داخل حفره select می کردیم و الگوریتم را ادامه می دادیم تا همگرا شویم.

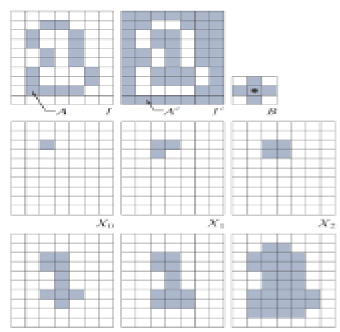


› Hole Filling Formulation:

$$X_k = (X_{k-1} \oplus B) \cap I^c$$

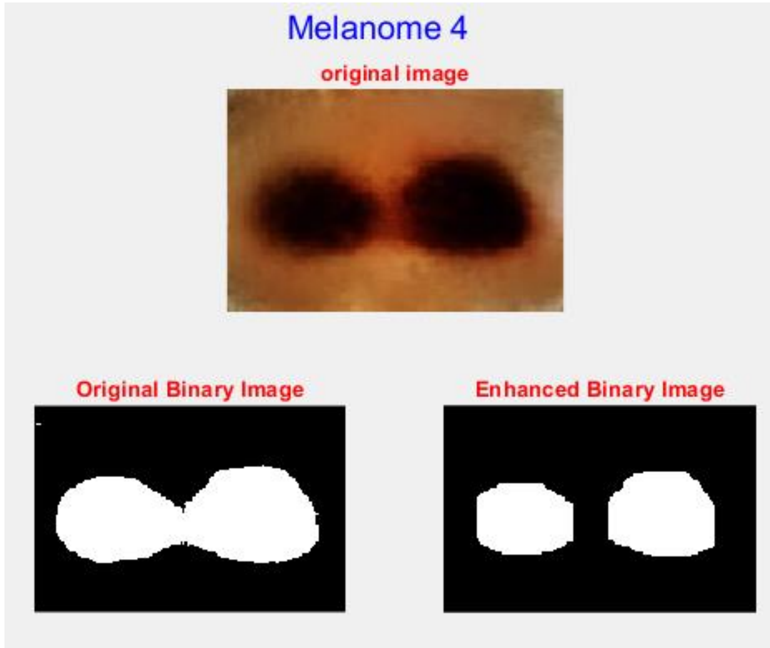
› Start inside the hole

› Repeat until convergence

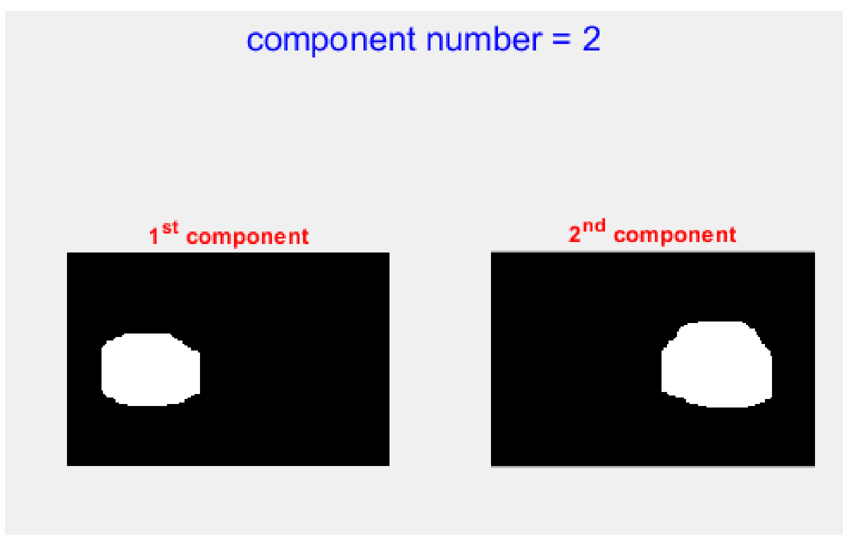


د) برای این قسمت الگوریتم به این شکل است که مطابق اسلاید ها با استفاده از یک کرنل سایز بزرگ سعی می کنیم تا دو ناحیه را از هم جدا کنیم. سپس با انتخاب کرنل‌های مناسب ابتدا از erosion و سپس از dilation استفاده می کنیم. در نهایت برای جبران بیرون زدگی‌های نابجای شکل از opening استفاده می کنیم.بنابراین تصویر به صورت زیر می شود. همه کرنل ها نیز مربعی می باشند:

Kernel	type	width
1	square	15
2	square	10
3	square	5
4	square	20

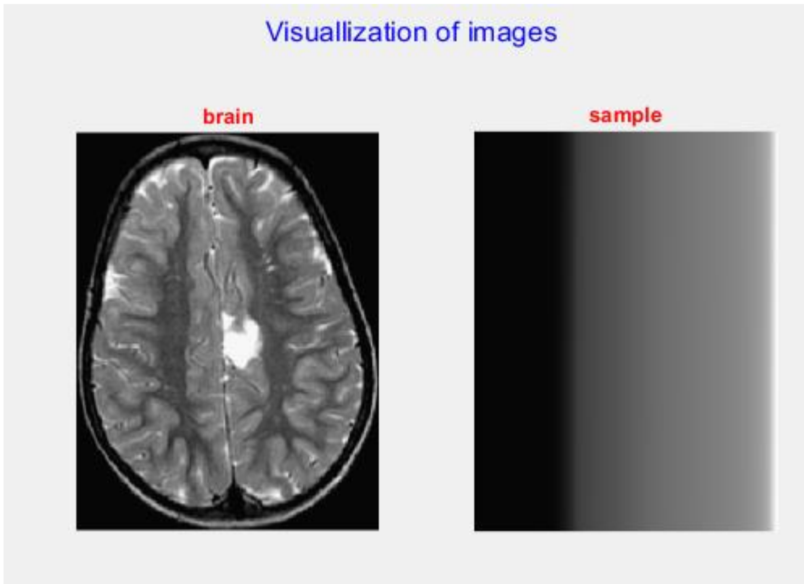


ه) این قسمت برای قسمت د احتمالا خواسته شده است چون قسمت د مربوط به 2 ضایعه است(در صورت سوال د این موضوع مطرح شده است) بنابراین داریم:

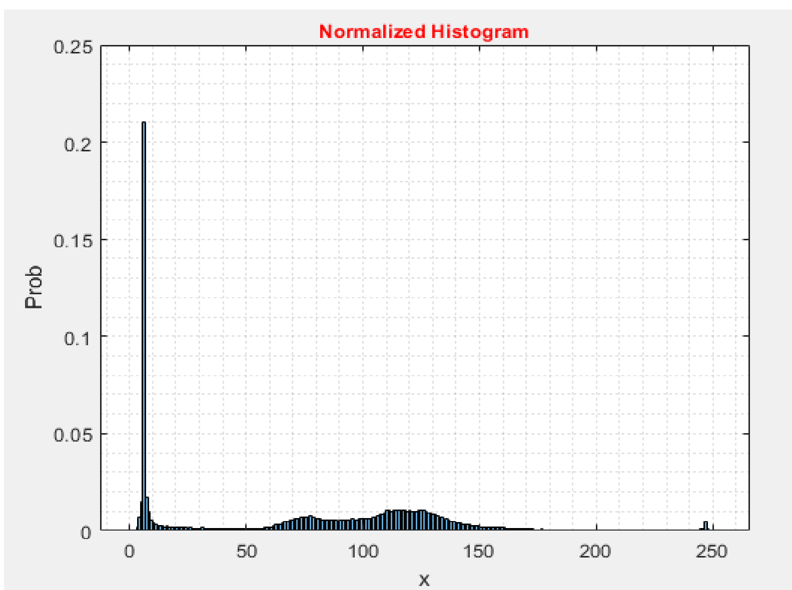


سوال 2

برای این قسمت تصاویر را لود می کنیم و نمایش می دهیم داریم :



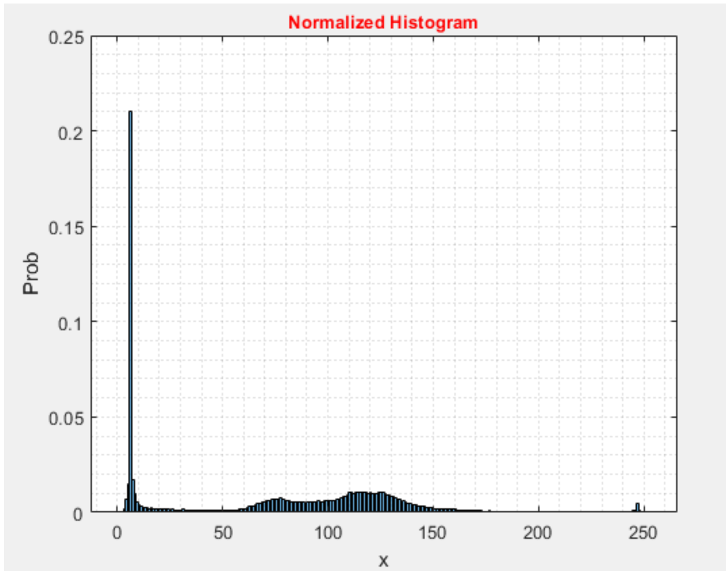
الف) برای این سوال داریم :



همچنین داریم : (برای محاسبه واریانس به جای رابطه اصلی از $Var(x) = E[x^2] - (E[x])^2$ استفاده شده است) :

Fig Name	mean	Variance	uniformity	Entropy
Brain	79.4222	3239.7252	0.049588	6.4437

ب) مشابه قسمت قبل داریم :



همچنین داریم : (برای محاسبه واریانس به جای رابطه اصلی از $Var(x) \ = \ E[x^2] - (E[x])^2$ استفاده شده است) :

Fig Name	mean	Variance	uniformity	Entropy
Brain	79.4222	3239.7252	0.049588	6.4437
sample	79.4222	3239.7252	0.049588	6.4437

همان گونه که دیده می شود در تصاویر Brain و sample میاگین، واریانس، uniformity و entropy با یکدیگر برابر هستند. همچنین علاوه بر این 2 بی شمار تصویر دیگر نیز وجود دارد که همین وضعیت را دارند. این به این دلیل است که ما از Pdf تصویر این مشخصات را استخراج می کنیم. مثلا اگر همین پیکسل ها را به نحو دیگری بچینیم تصویر متفاوتی خواهیم داشت در حالی که تمامی پارامتر ها با پارامتر های بالا یکسان خواهد بود. بنابراین بی شمار تصویر دیگر با این ویژگی وجود دارد.

ج) برای این قسمت از تابع graycomatrix استفاده می کنیم. همچنین برای آشنایی بیشتر با GLCM تابع آن را خودم نیز نوشته ام . (اگر امتیازی داشته باشد خیلی خوب می شود 😊) همانطور که تابعی که پیاده سازی کرده ام مشخص است ابعاد این ماتریس برابر تعداد سطوح روشنایی است یعنی در اینجا 256×256.

د) مشخصه های contrast، uniformity، homogeneity و entropy را از روی ماتریس GLCM برای دو تصویر به دست می آوریم. نتایج مطابق زیر می باشد:

Fig Name	contrast	uniformity	Homoginity	Entropy
Brain_f	114.3506	0.037457	0.43554	10.3673
sample_f	2.984	0.046276	0.7526	7.1545

همانگونه که در قسمت ب دیدیم اگر مشخصه ها را از روی هیستوگرام دو تصویر به دست آوریم ممکن است مشخصه ها برای دو تصویر کاملا متفاوت یکسان شود این به این دلیل است که ممکن است برای دو تصویر متفاوت هیستوگرام کاملا یکسانی داشته باشیم اما اگر از GLCM استفاده کنیم به احتمال بالایی GLCM برای دو تصویر متفاوت خواهد بود. بنابراین اگر مشخصه ها را بر این اساس محاسبه کنیم می بینیم که نسبت به یک دیگر متفاوت خواهند بود. در بالا نیز این دیده می شود که بر خلاف قسمت ب که مولفه ها یکسان شد در اینجا مولفه ها با یکدیگر متفاوت هستند.

ه) برای کاهش ابعاد ماتریس GLCM باید ابتدا سطوح روشنایی را کوآنتیزه کنیم بنابراین ابتدا سطوح را از 256 به 64 کوآنتیزه می‌کنیم و در نتیجه ابعاد ماتریس GLCM از 256×256 به 64×64 می رسد. محاسبه ماتریس GLCM از نظر complexity حجیم است بنابراین زمانی که با تصاویری با پیکسل بالا سر و کار داریم محاسبه GLCM بسیار سنگین و پیچیده خواهد بود. در این زمان مفید است که این راهکار را انجام دهیم. در نتیجه برای محاسبه سریع تر از ماتریس GLCM با سایز کوچک‌تر استفاده می کنیم. همان گونه که در زیر دیده می شود کنتراست نسبت به قبل کاهش ، uniformity افزایش ، Homoginity افزایش و entropy کاهش پیدا کرده است که همگی مطابق با کوآنتیزیشن و مطابق تئوری مسئله می باشد. بنابراین با نجام این کار می توان مقداری هزینه محاسبه را کاهش داد.

Fig Name	contrast	uniformity	Homoginity	Entropy
Brain_f_q	7.2631	0.057888	0.65557	7.1907
sample_f_q	2.984	0.046276	0.7526	7.1545

پارامتر ها با کاهش ابعاد

ی) برای این قسمت ابتدا به سراغ GLDM می رویم. GLDM در واقع Gray level Difference Method Probability Density Functions را برای تصویر داده شده محاسبه می کند. در واقع اگر تصویر را به صورت یک ماتریس در نظر بگیریم می توان تابع توزیع روشنایی را به صورت $f(x, y)$ تعریف کرد. حال با در نظر

گرفتن d به عنوان ورودی چهار جابجایی (d,0) و (0,d) و (-d,-d) و (-d,d) را تحت عنوان D_i تعریف می کنیم و چهار $f_i(x,y)$ را به صورت زیر محاسبه می کنیم. داریم :

$$for\ D_i \rightarrow f_i(x,y) = |f_i(x,y) - f(x + D_i(1), y + D_i(2)| \qquad i = 1:4$$

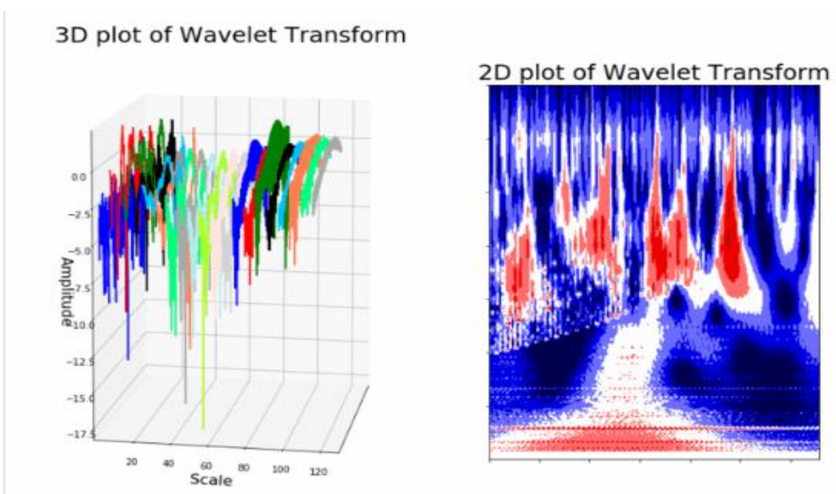
حال برای 4 المانی که به دست آمده اند Pdf ها را محاسبه و به خروجی می دهیم حال با استفاده از این PDf می توان ویژگی های مختلفی را استخراج کرد. فایل تابع آن نیز تحت عنوان GLDM.m در کد ها آورده شده است که 4 ورودی آن d و تصویر و خروجی آن Pdf1 , pdf2, pdf3, pdf4 می باشد.

برای GLRLM نیز می دانیم GLRLMمخفف gray level run length matrix است که از آن برای استخراج ویژگی استفاده می شود.(با استفاده از ماتریس run length) این ماتریس هم مثل ماتریس‌های هم رخداد درGLCM ، اگه تصویرمان n تا سطح خاکستری داشته باشه، n*n عنصر می‌تواند داشته باشد. در این روش اگر فاصله را 1 فرض کنیم سطح i در ماتریس در حال نشان دادن یک عدد خاکستری از ماتریس سمت تصویرمان می باشد و ستون ها (j) بیانگر تعداد مرتبه ای هست که اینها کنار هم قرار گرفتند**.** مثلا در ماتریس اگر در خانه ی (i=2,j=3) عدد یک نوشته شده باشد، به این معنی هست که در ماتریس تصویرمان عدد 2 تنها یک بار به صورت سه تایی در تصویر قرار گرفته است.و یا به عنوان مثال وقتی به ازای مقدار i=4 , j=2 در ماتریس مقدار 1 مشاهده می شود این به این معنی است که پیکسل دارای سطح خاکستری 4 تنها یک بار در ماتریس تصویر اصلی به صورت دوتایی در کنار هم قرار گرفته است. (در واقع یک gray level run به صورت یک خط از پیکسل ها با شدت روشنایی مشخص در یک جهت خاص تعریف می شوند.)

سوال 3

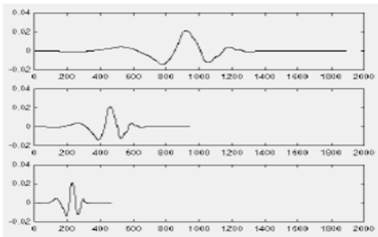
الف) می دانیم تبدیل فوریه برای آنالیز سیگنال از یک سری امواج سینوسی با فرکانس‌های مختلف استفاده می‌کند و بنابراین سیگنال را می توان صورت ترکیبی خطی از سیگنال‌های سینوسی نوشت. تبدیل موجک از تعدادی توابع به نام موجک(wavelet) استفاده می‌کند که هر کدام مقیاس متفاوتی دارند. همان طور که می‌دانیم معنی واژه wavelet، موج کوچک است . برای آن که با مفهوم آن آشنا شویم ابتدا باید تفاوت سیگنال سینوسی و wavelet را بدانیم. در واقع wavelet بر خلاف سیگنال سینوسی در یک لحظه خاص از زمان واقع شده است. یکی از ضعف های تبدیل فوریه این است که اطلاعات زمانی را به ما نمی دهد و تنها محتوای فرکانسی را به ما می دهد در حالی که چون wavelet در یک لحظه خاص از زمان واقع شده است در نتیجه علاوه بر اطلاعات فرکانسی اطلاعات temporal را هم به ما می‌دهد. چون موجک در زمان واقع شده است، در نتیجه می‌توان سیگنال اصلی را در لحظات مختلف از زمان در موجک ضرب کرد.

در گام نخست، با نقاط ابتدایی سیگنال شروع می‌کنیم و به تدریج موجک را به سمت انتهای سیگنال حرکت می‌دهیم. این عمل را کانولوشن (Convolution) می‌گویند. بعد از این که کانولوشن را با سیگنال موجک اصلی (موجک مادر) انجام دادیم، می‌توانیم آن را به نحوی مقیاس‌دهی کنیم که بزرگ‌تر شود و دوباره فرایند را تکرار کنیم. (در واقع می خواهیم بدانیم که چقدر از موجک در سیگنال موجود است)



همان طور که از تصویر بالا مشخص است، تبدیل موجک یک سیگنال تک بعدی، دارای دو بعد است. این خروجی دو بعدی مربوط به تبدیل موجک، نمایش سیگنال اصلی بر حسب مقیاس و زمان است که به طیف

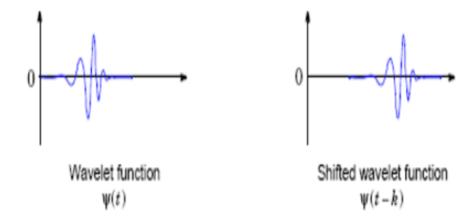
اسپکتروگرام (Spectrogram) یا اسکالوگرام (Scaleogram) معروف است. منظور از shift , scale در واقع به صورت زیر است:



$$f(t) = \psi(t) \quad ; \quad a = 1$$

$$f(t) = \psi(2t) \quad ; \quad a = \frac{1}{2}$$

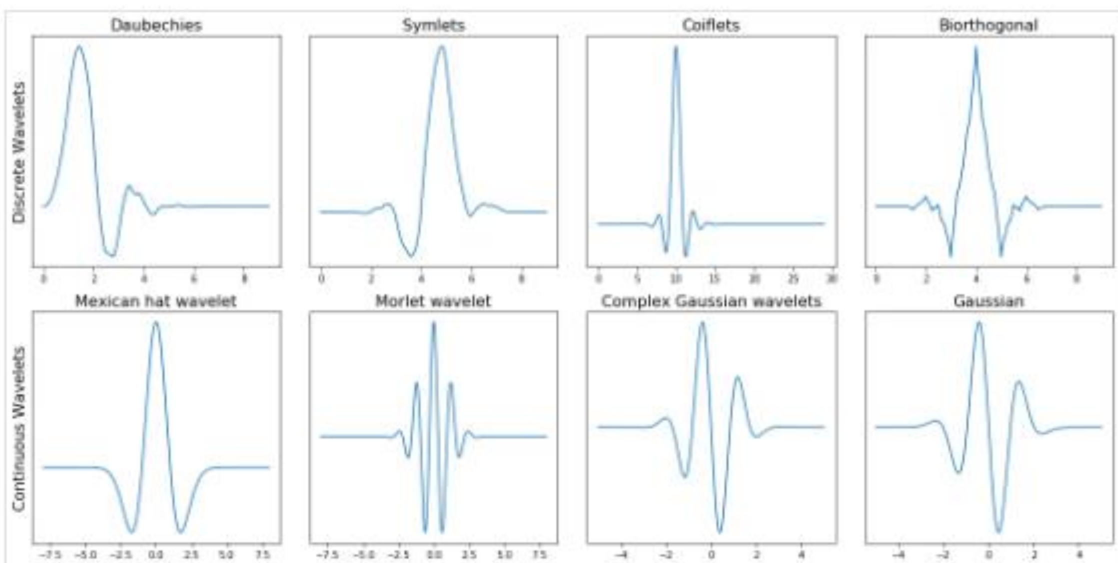
$$f(t) = \psi(4t) \quad ; \quad a = \frac{1}{4}$$



scale

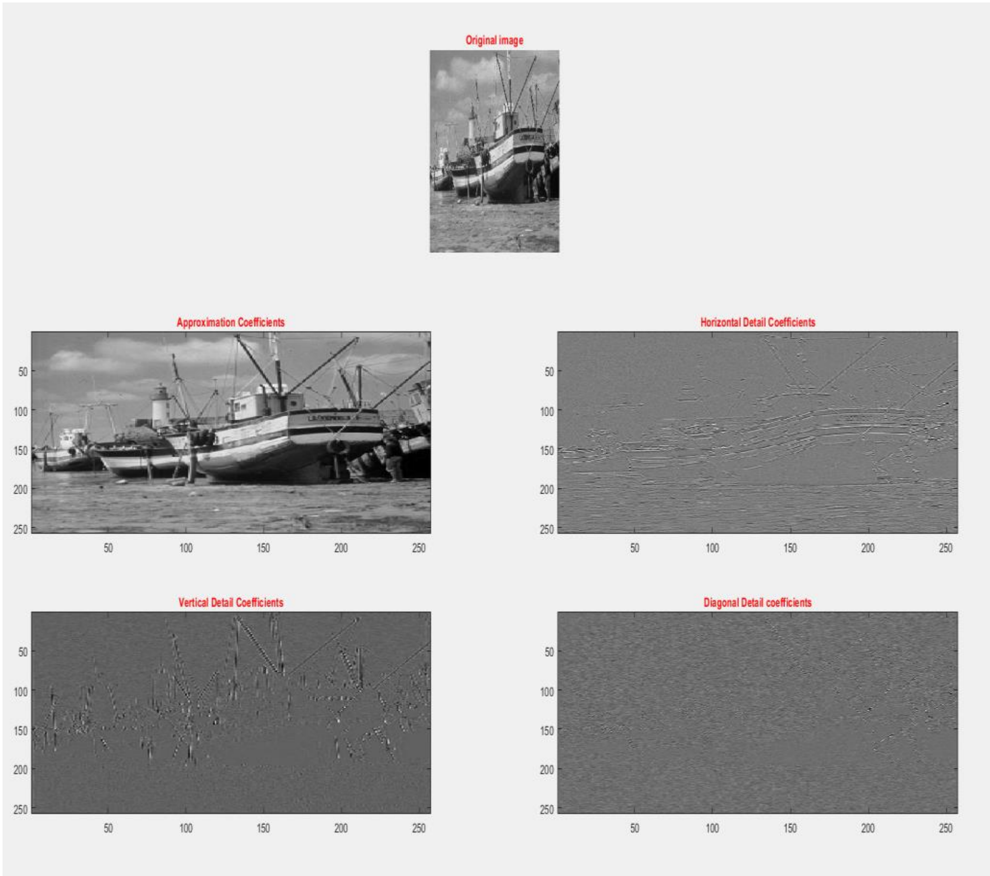
Shift

همچنین برای ویولت ها خانواده های متفاوتی وجود دارد که هر خانواده دارای شکل، فشردگی و همواری متفاوتی هستند و برای هدف متمایزی مورد استفاده قرار می گیرند. در زیر تعدادی از این موجک ها نشان داده شده اند:

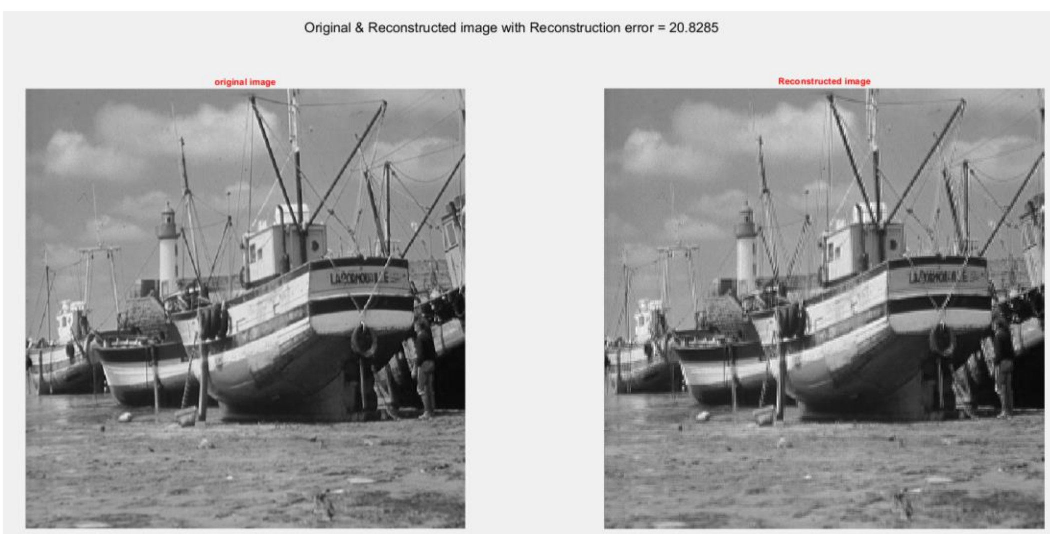


تصویری از چند خانواده موجک گسسته در ردیف بالا و چند خانواده موجک پیوسته در ردیف پایین

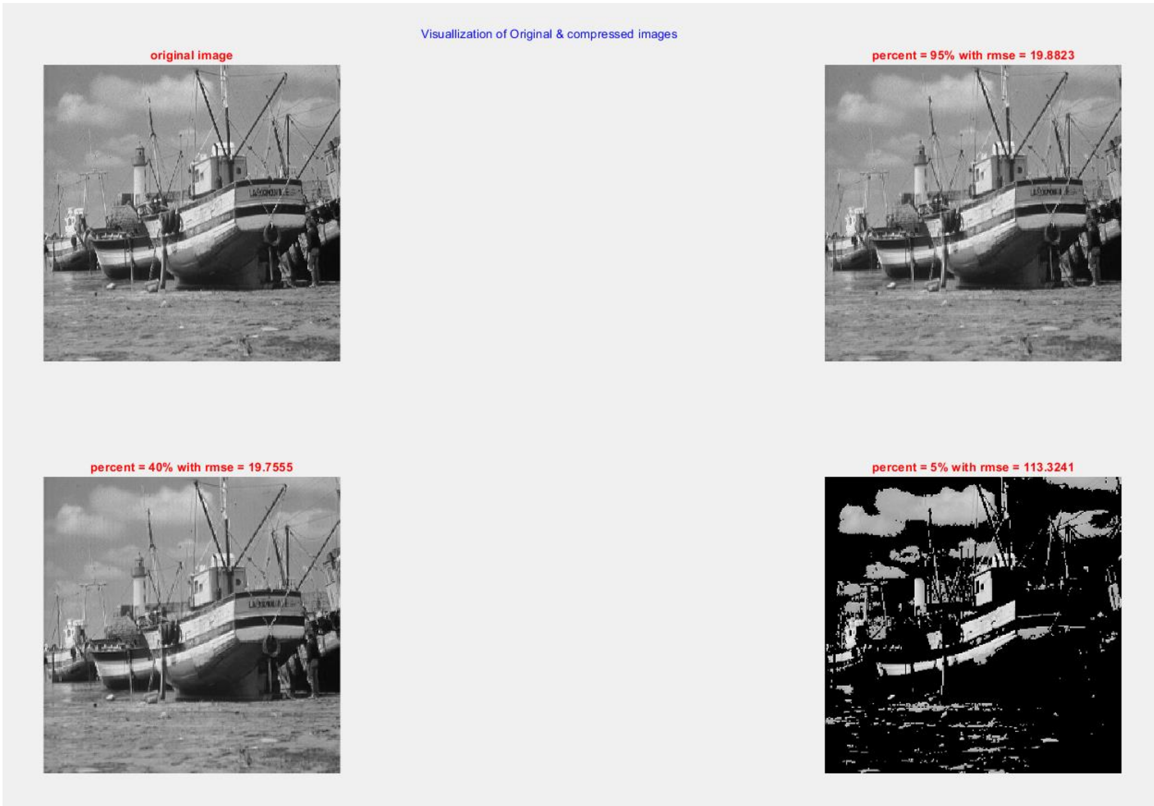
ب) برای این قست داریم :



ج) برای این قسمت داریم : (خطا در title تصویر آورده شده است) :



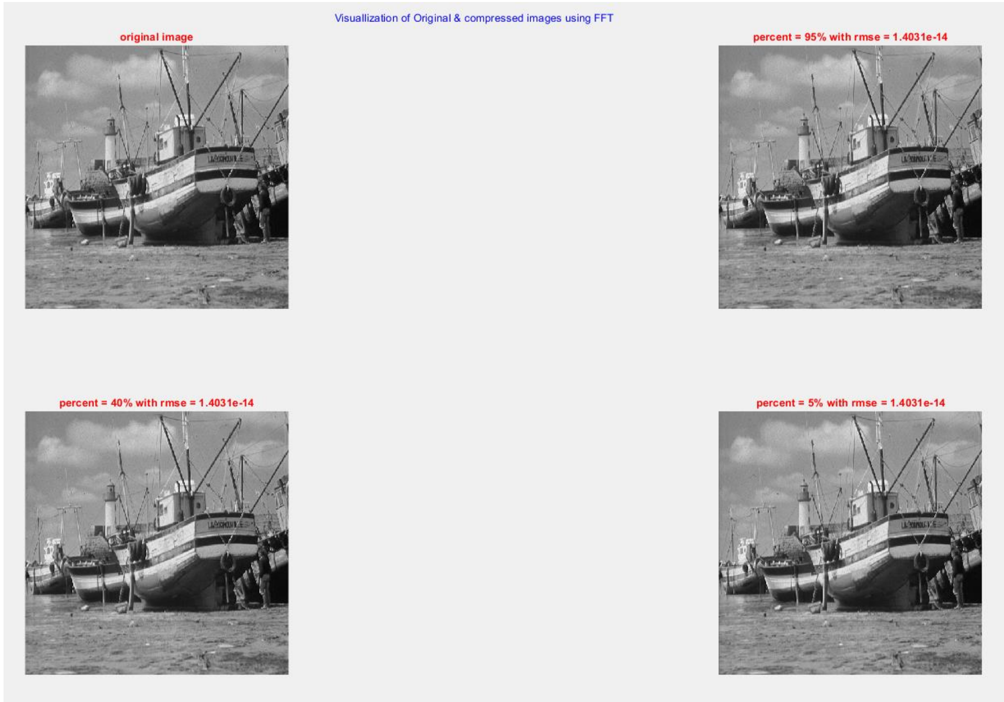
د) برای این قسمت داریم : (خطا در title تصویر آورده شده است) :



ه) برای این قسمت داریم :

Ratio	FFT	wavelet
95%	1.0657:1	1.0569:1
40%	1.0657:1	1.1292:1
05%	1.0657:1	3.0561:1

همچنین نتایج بازسازی FFT به صورت زیر می باشد :

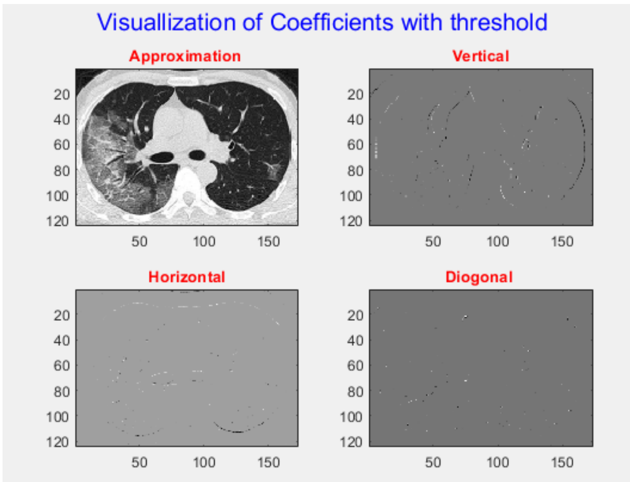


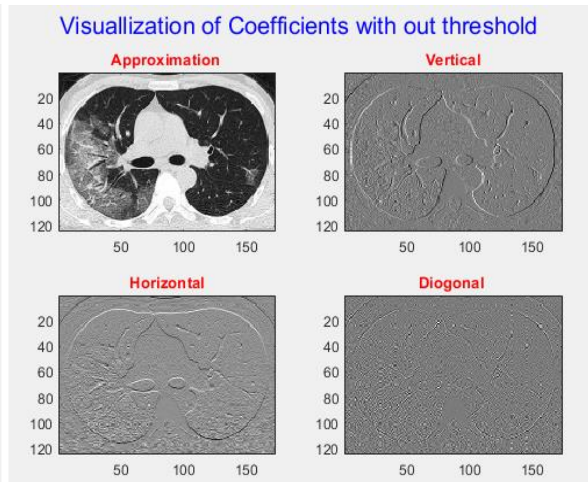
همانطور که میبینیم در اینجا DFT مثلا به ازای 5 درصد بهتر عمل کرده است به گونه ای که میبینیم تصویر بازسازی شده از FFT با درصد 5 درصد نسبت به تصویر بازسازی شده از DWT با درصد 5 درصد بهتر است.

به صورت کلی در پردازش تصویر نمی توان گفت که کدام تبدیل بهتر است و هر کدام فواید خودش را دارد. مزایای استفاده از DWT نسبت به DFT در این است که DWT اجزای تصویر با جزئیات بالا را بر روی توابع پایه کوتاه‌تر نمایش می‌دهد(با رزولوشن بالاتر) در حالی که اجزای جزئیات کمتر بر روی توابع پایه بزرگتر پروجکت می شوند، که مربوط به زیر باندهای باریک تر است، که یک مبادله بین رزولوشن زمان و رزولوشن فرکانس ایجاد می کند.

سوال 4

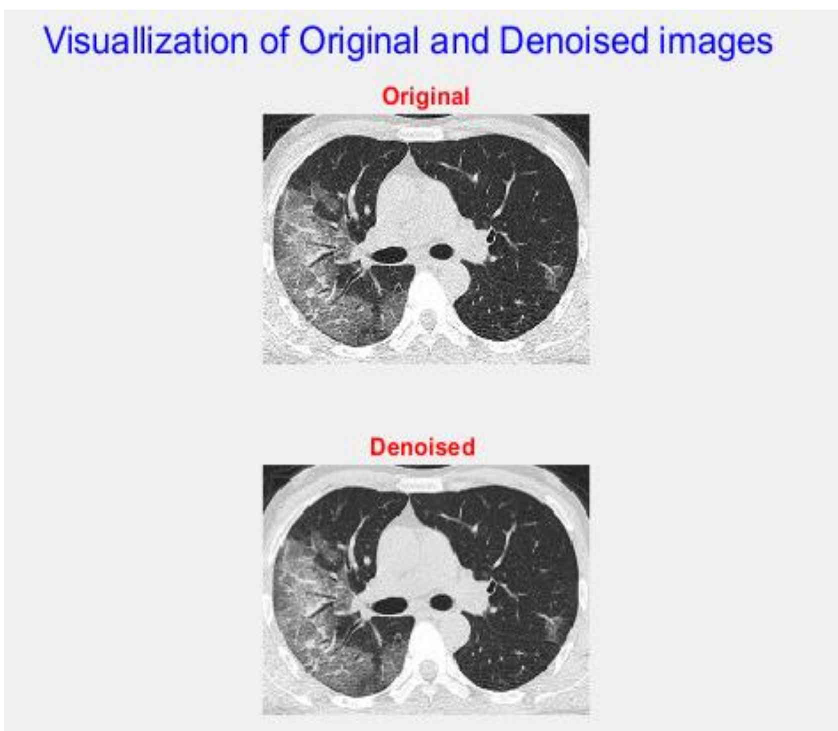
الف) برای این قسمت برای هر کدام از زیر باند ها به صورت جداگانه آستانه می گذاریم. بنابراین داریم :





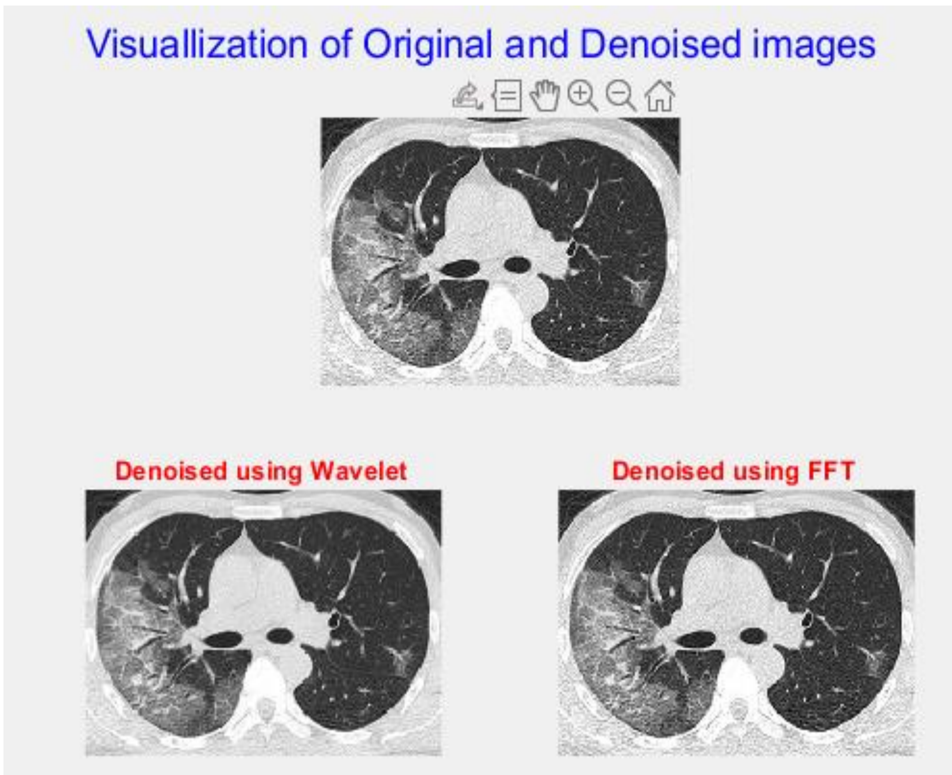
همانگونه که دیده می شود تصویر با اعمال ترشولد هموار تر شده است. این موضوع را به وضوح می توان در قسمت diagonal دید به طوری که تصویر نویزش کمتر و هموار تر شده است.

ب) برای این قسمت داریم :



همانگونه که دیده می شود تصویر نویزش کمتر شده است و به عبارتی دیگر دینویز شده است. این حذف نویز در قسمت های سفید رنگ به راحتی دیده می شود به طوری که نویز تصویر کمتر شده است و تصویر هموارتر شده است.

ج) برای این قسمت داریم :



همانگونه که در تصویر بالا دیده می شود می بینیم که wavelet عملکرد به مراتب بهتری نسبت به DFT دارد و با استفاده از ویولت تصویر به مراتب هموارتری خواهیم داشت. همانگونه که می دانیم و در قسمت 3 الف توضیح دادیم DFT با استفاده از توابع سینسی سیگنال را آنالیز می کند ولی ویولت با استفاده از موجک های محدود در زمان سیگنال را آنالیز می کند بنابراین ویولت می تواند با توجه به نوع انتخاب ویولت مادر به ما تنوع بیشتر و بهتری بدهد و همچنین اطلاعات زمانی را نیز می تواند به ما ارائه کند. برای دینویز کردن زمانی می توان از DFT استفاده کرد که در چگالی توان طیف فرکانس اطلاعات مربوط به تصویر سطحشان نسبت به نویز جمع شونده بالاتر

باشد و در این مواقع می توان با استفاده از DFT دینویز کرد و نتیجه مناسبی گرفت اما در مواقعی که نویز پیچیده تر است با توجه به آن که ویولت های متنوعی داریم می توان با استفاده از ویولت مناسب نتیجه بهتری نسبت به DFT گرفت.