

به نام خدا

آزمون میانترم درس پردازش تصویر

استاد : دکتر آذرنوش

امیرحسین شریفی صدر

9733044

سوال 1 : در بخش ابتدایی سوال از ما خواسته شده ابعاد ، تعداد کانال ها و نوع داده های پیکسل را گزارش نماییم که در پایین مشاهده

میکنید :

```
image size is : (2912, 2912, 3)
number of channels is : 3
type of image is : uint8
```

در ابتدا چون تصویری را به شکل عادی و نه خاکستری خوانده ایم تعداد کانال های تصویر را 3 نشان میدهد اما اگر آن را به شکل خاکستری میخواندیم تعداد کانال ها 2 نمایش میداد .

*در ادامه حل سوال تصویر به شکل خاکستری خوانده شده

- بخش آ : از ما خواسته شده تا میزان حافظه برای ذخیره ی عکس را به دست بیاوریم که معادل استاد با مساحت تصویر (ضرب طول در عرض در تصویر) ضربدر میزان تعداد بیت تصویر و چون تصویر به

صورت 8 بیتی خوانده شده است میشود : $M * N * 8$ image storage is $M * N * 8 = 67837952$

- بخش ب : نمودار هیستوگرام یک تصویر : هر تصویر از تعدادی graylevel تشکیل شده است . برای مثال در یک تصویر 8 بیتی 2^8

بیت graylevel وجود دارد یعنی از 0 تا 255 . و این گری لول ها تصویر پخش هستند و تز هر کدام تعدادی وجود دارد . هیستوگرام یک تصویر به ما تعداد هر کدام از این graylevel ها را میگوید که در

توصیر چندتاست . برای مثال چند پیکسل با graylevel ، 200 در

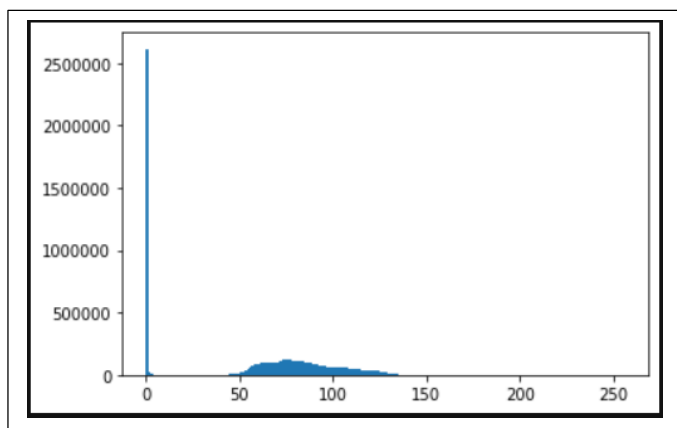
تصویر وجود دارد .

Histogram:	$h(r_k) = n_k$
Gray levels:	r_k
Repetition of each gray level:	n_k

و محور های نمودار هیستوگرام : محور افقی آن به ترتیب گری لول ها را از 0 تا 255 نشان میدهد و محور عمودی آن تعداد هر یک از گری لول ها را در تصویر به ما میدهد .

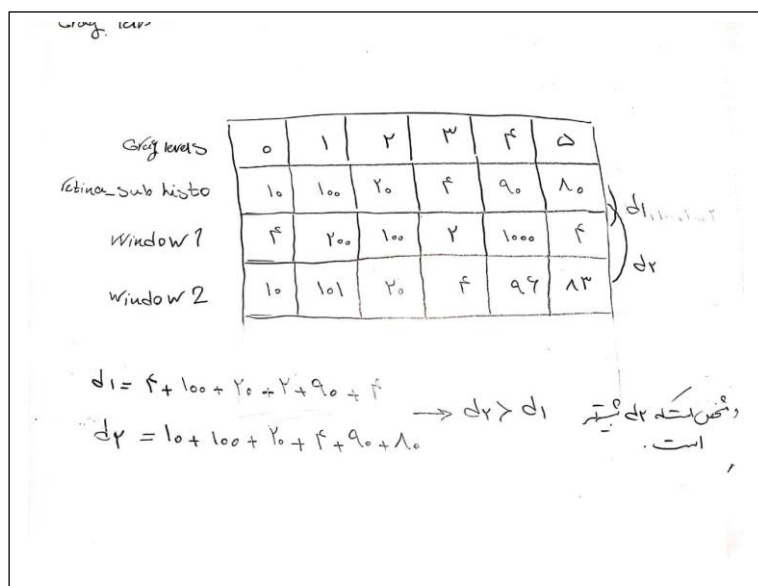
و این هیستوگرام برای تصویر ورودی ما نشان از تجمع بسیار زیادی graylevel با مقدار 0 به ما میدهد به خاطر حاشیه سیاه اطراف تصویر . و نشان میدهد که تصویر دایره ای شکل ما تجمع نسبتا یکسانی از گری لول های 50 تا حدود 150 را دارا است .

به طور کلی یعنی تصویر contrast بالایی ندارد چون همه گری لول ها در آن وجود ندارد .

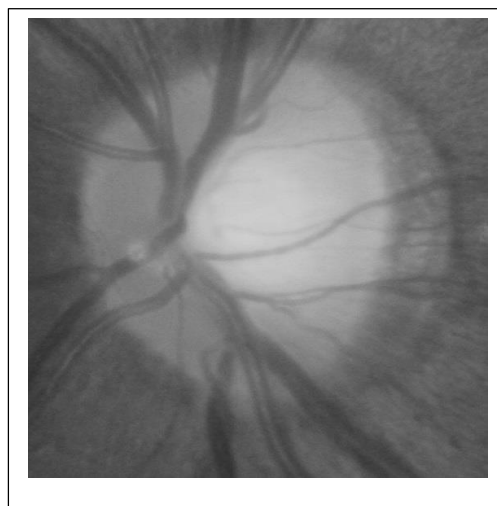


- بخش ج : در بخش ج از ما خواسته شده تا مشابه ترین تصویر به تصویر retina_sub را در درون تصویر اصلی خود ینی retina پیدا کنیم . برای اینکار س.ال پیشنهاد داده است که تصویر retina_sub را روی تصویر retina لغزاند و در هر مرحله هیستوگرام (تعداد گری لول های تصویر) تصویر retina_sub را با پنجره از تصویر که با همان سایز retina_sub جدا کرده ایم مقایسه کنیم و معیار مقایسه مان اینگونه است که هیستوگرام هر گری لول را از دو تصویر با هم مقایسه کرده و min (مینیموم) شان را در با متغیری مانند d با مقدار اولیه 0 جمع میکند و نتیجه را برابر با یک max قرار میدهد سپس d را صفر کرده و دوباره این کار را برای پنجره بعدی و تصویر retina_sub انجام داده و با max مقایسه میکند و اگر بزرگتر یا مساوی آن بود آن را جایگزین max میکند . و این کار را مدام انجام میدهیم تا تصویر retina_sub به طور کامل روی تصویر ما بلغزد . در نهایت مختصات آن بخش از تصویر که هیستوگرامش در مقایسه با هیستوگرام retina_sub بیشترین d را داشته میشود شبیه ترین تصویر به تصویر retina_sub و این موضوع نیز مشخص است زیرا اگر دو تصویر بسیار به هم شبیه باشند پس مقداری

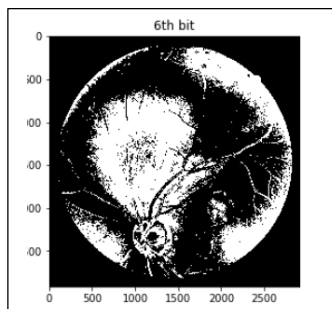
کمتر از هیستوگرام هر گری لول retina_sub با d جمع
نمیشود و اینگونه بیشترین d را داریم .



در تصویر بالا سعی شده به با در نظر گرفتن 6 گری لول و دادن
مقدار برای آن ها در 3 تصویر و مقایسه به شیوه استفاده شده در
کد نشان داد که به چه علت این شیوه پاسخ گو است .
تصویر سمت راست تصویر retina_sub و تصویر سمت چپ
مشابه ترین تصویر به آن در تصویر retina است .



بخش د : در این سوال هر شدت هر پیکسل از تصویر را به شکل باینری با 8 بیت درآورده و سپس بیت مورد نظر از شکل را برای تمام پیکسل ها جدا کرده و در یک تصویر نشان میدهیم و اینگونه تصویر را لایه

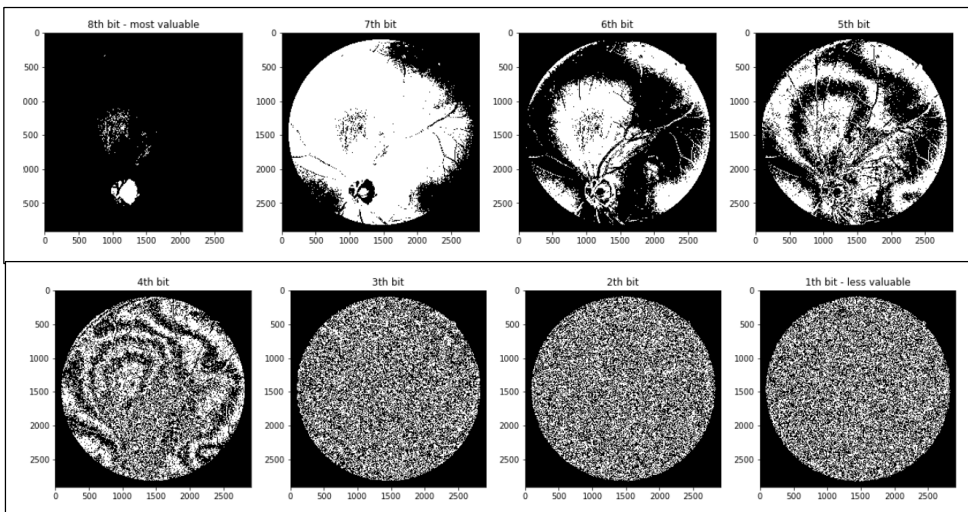


بندی میکنیم . برای مثال در این تصویر تمامی پیکسل ها نشان دهنده بیت ششم استخراج شده از باینری شده شدت هر

پیکسل تصویر اصلی هستند که یا 0 میتواند باشد یا 1 .

و این لایه بندی بدین شکل است که هر چه به بیت های پایین تری میرویم جزئیاتی نمایان میشود که تصویر بیت های بالاتر را کامل میکند .

پس کلیات تصویر در بیت های بالاتر که بیشترین در msb و جزئیات در بیت های پایین تر که کمترین lsb است .



سوال 2

بخش آ :

$$S(x) = (L-1) \sin(\alpha x) \quad [0, L-1] \rightarrow [0, L-1]$$

از آنجا که $S(0) = 0$ و $S(L-1) = L-1$

$$S(0) = (L-1) \alpha \sin(0) = 0 \quad \text{و} \quad S(L-1) = (L-1) \alpha \sin(\alpha(L-1)) = L-1$$

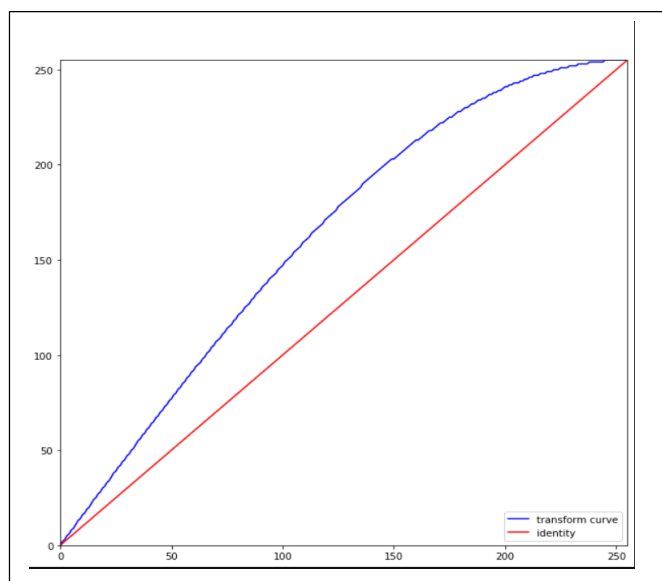
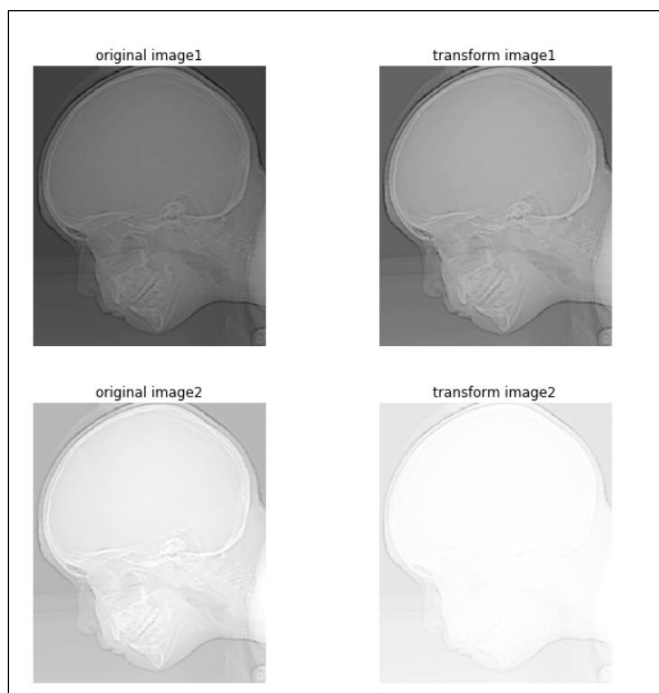
$$\rightarrow \sin(\alpha(L-1)) = 1$$

بازه صعودی \sin مد نظر است

$$\rightarrow \alpha(L-1) = \frac{\pi}{2} \rightarrow \alpha = \frac{\pi}{2(L-1)}$$

بخش ب : تابع را بر دو تصویر خوانده شده اعمال میکنیم و نتیجه را

مشاهده میکنیم :



بخش ج : نمودار خواسته شده :

محور افقی این نمودار مقادیر شدت های ورودی ما و قسمت عمودی این نمودار برای نمودار آبی نشان دهنده مقادیر خروجی بعد از اعمال تابع تبدیل ما هستند .

بخش د : همانطور که از نمودار منحنی تبدیل و همچنین تصاویر به وجود آمده از اعمال تابع تبدیل بر آن ها مشخص است ، این تابع تبدیل برای تصاویر تاریکتر مناسب تر است . زیرا با توجه به نمودار بخش "ج" این تابع شدت های با مقدار بالاتر (یعنی روشن تر) را بیشتر از شدت های با مقدار پایینتر (یعنی تاریکتر)، بیشتر میکند و از چیزی که هستند نیز روشن تر میشوند و عملا اگر تصویر ما روشن باشد پس از اعمال این تابع تصویر بسیار روشنتر و ناواضح تر میشود . ولی برای تصاویر تاریک تر یعنی تصاویری که دارای شدت های پایینتری مثلا 0 تا 100 هستند مناسب تر است .

سوال 3 :

الف :

```
import numpy as np
import matplotlib.pyplot as plt
import cv2

# must be read in "gray" mode
img = cv2.imread('lung.png',0)

# the best filter for slat and paper noise is median and its
img_2 = cv2.medianBlur(img, 3)

# for vertical gradient
img_3 = cv2.Sobel(img_2, cv2.CV_64F, dx=0, dy=1)

plt.figure()
plt.suptitle('Problem 3 Figure')

plt.subplot(1, 3, 1)
plt.title('Original')
plt.imshow(img, cmap='gray')
plt.axis(False)

plt.subplot(1, 3, 2)
plt.title('Denoised')
plt.imshow(img_2, cmap='gray')
plt.axis(False)

plt.subplot(1, 3, 3)
plt.title('Gradient')
plt.imshow(img_3, cmap='gray')
plt.axis(False)

plt.show()
```

```
import numpy as np
import matplotlib.pyplot as plt
import cv2

img = cv2.imread('lung.png')

img_2 = cv2.blur(img, (3, 3))
img_3 = cv2.Sobel(img_2, cv2.CV_64F, dx=1, dy=0)

plt.figure()
plt.suptitle('Problem 3 Figure')

plt.subplot(1, 3, 1)
plt.title('Original')
plt.imshow(img, cmap='gray')
plt.axis(False)

plt.subplot(1, 3, 2)
plt.title('Denoised')
plt.imshow(img_2, cmap='gray')
plt.axis(False)

plt.subplot(1, 3, 3)
plt.title('Gradient')
plt.imshow(img_3, cmap='gray')
plt.axis(False)

plt.show()
```

اشتباه اول در کد این است که از ما خواسته شده تصویر به شکل

خاکستری خوانده شود ولی کد اولیه تصویر را عادی میخواند

اشتباه دوم این است که برای حذف نویز فلفل نمکی از ما خواسته شده

که با فیلتری به صورت غیرخطی و استاتیستیک آماری این کار انجام

شود ولی فیلتر استفاده شده در کد داده شده که فیلتر `cv.blur` است

فیلتر میانگین گیری خطی است و همچنین خیلی هم برای حذف نویز

فلفل نمکی مناسب نیست . مناسب ترین فیلتر برای حذف این نوع نوع نویز فیلتر مدین است . که هم غیرخطی است و هم از معیار آماری برای ماسک استفاده میکند که به این شکل است که پنجره به شکل مربع یا حتی مستطیل با تعدادی فردی خانه روی شکل ما میلغرد و پیکسل وسط را با میانه پیکسل های موجود در آن پنجره جایگزین میکند و به این ترتیب نویز حذف میشود .

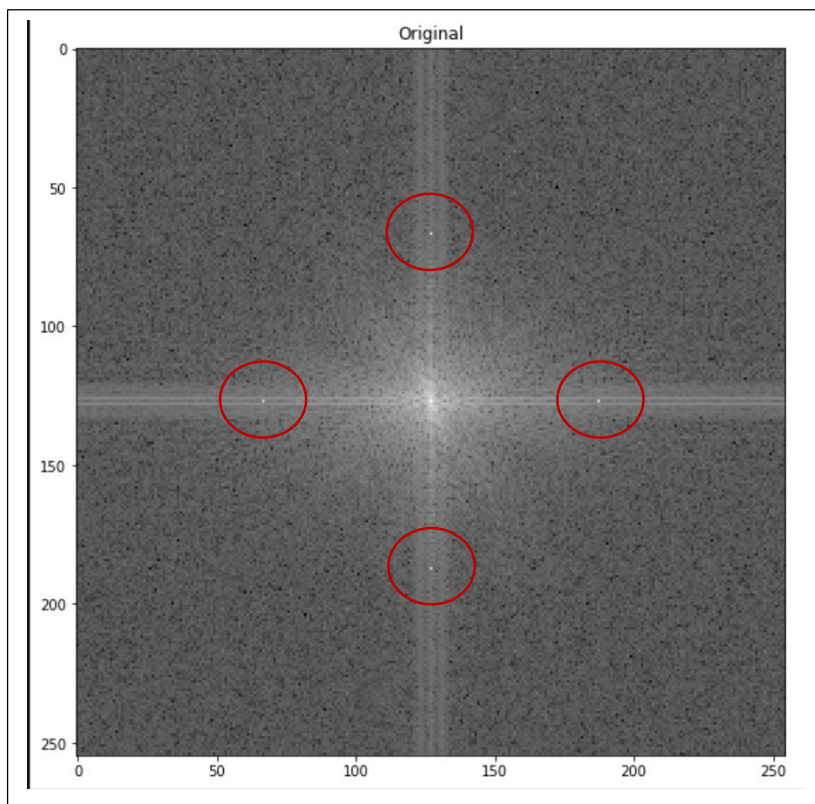
اشتباه سوم در فیلتر سوبل (`cv.sobel`) بر خلاف قرار داد درس محور عمودی y و محور افقی x است و برای گرفتن گرادیان عمودی نیاز است که از y مشتق گرفته شود پس $dy=1$ و $dx=0$ میشود و در مورد بالا گذر بودن نیز احتیاجی به تغییر نداریم .

بخش ب :

-1	-2	-1
0	0	0
1	2	1

برای اینکه هم لبه بالارونده داده شود و هم لبه پایین رونده بعد از اعمال ماسک مقادیر منفی را قدر مطلق گرفته و و به صورت مثبت نمایش میدهیم .

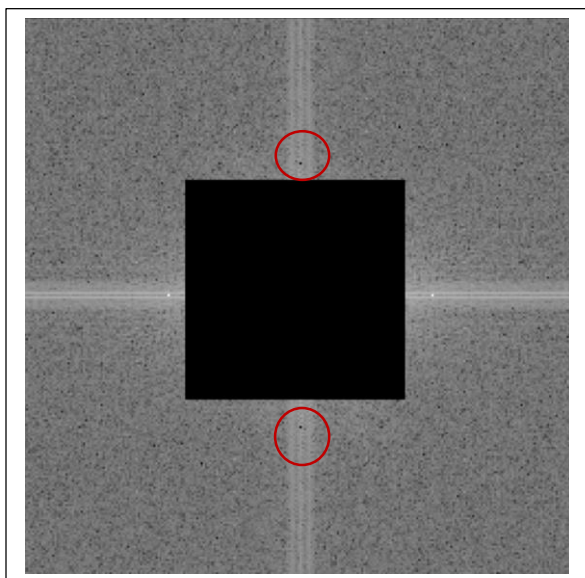
سوال 4 : در سوال 4 ابتدا باید پیکسل هایی که باعث اعوجاج در تصویر شده اند پیدا شوند . برای اینکه ابتدا تصویر را به شکل فرکانسی خوانده و تبدیل فوریه آن را به دست میاوریم سپس **magnituide** spectrum آن را استخراج میکنیم تا از روی آن بتوانیم بفهمیم نقاط اعوجاج چه در چه بازه ای قرار دارند .



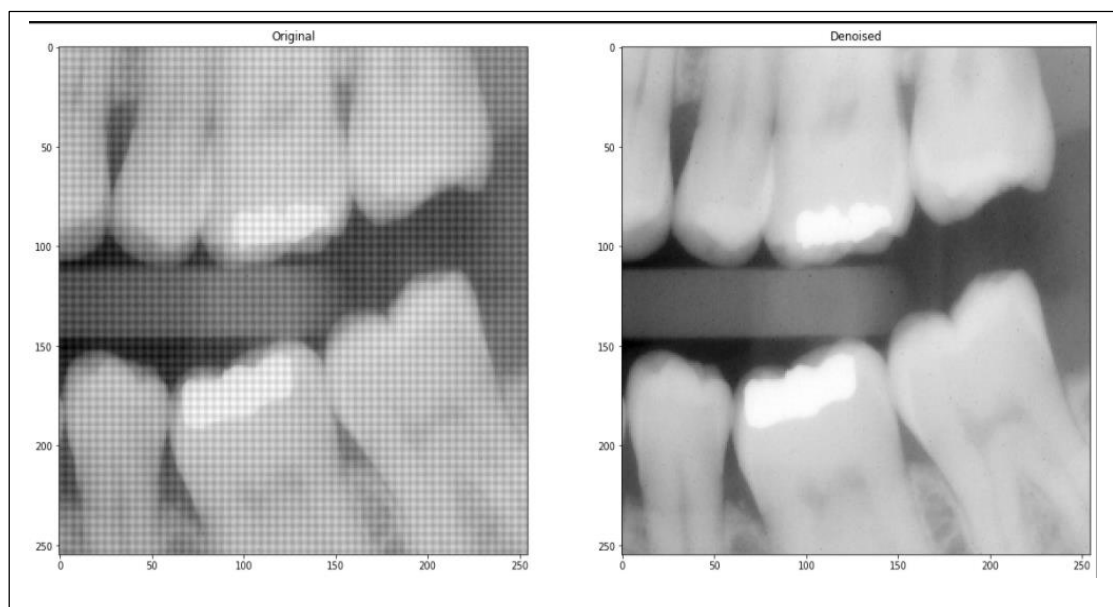
نقاط اعوجاج در شکل مقابل قابل مشاهده هستند .

متوجه میشویم که 4 نقطه اعوجاج در تصویر موجود است پس ما باید مختصات این نقاط را به دست آورده و آن ها را با میانگین 4 نقطه در همسایگیش جایگزین کنیم .

برای به دست آوردن محل این نقاط روشی که بنده استفاده کردم بدین شکل است که چون تا حدودی مشخص است این نقاط شدت بیشتری نسبت به نقاط اطراف خود دارند البته نسبت به نقاط مرکزی spectrum خیر چون آن ها بسیار روشن هستند . پس برای یافتن این نقاط ابتدا با یک تخمین مناسب با توجه به طول و عرض تصویر ناحیه ای مربع شکل از وسط تصویر را به شکل سیاه درمیاوریم تا مطمئن شویم دیگر در آن ناحیه پیکسلی با شدت ماکسیمم وجود ندارد . سپس حدس میزنیم که نقاط اعوجاج ما دارای شدت یکسانی هستند پس اگر از بقیه تصویر ماکسیمم بگیریم به آنها دست پیدا میکنیم . اما پس اعمال این روش میبینیم که تنها به نقاط اعوجاج بالا و پایین دست پیدا میکنیم و نقاط چپ و راست مشخص نمیشوند یعنی آنها دارای شدت برابر با نقاط بالا و پایین نیستند :



جایگزینی مد نظر را برای نقاط اعوجاج بالا و پایین در تصویر فوریه انجام می‌دهیم و سپس این نقاط را در تصویر spectrum سیاه می‌کنیم تا در مرحله بعد جزو ماکسیمم‌ها نباشند و ما به آن دو نقطه اعوجاج دیگر دست یابیم. آن دو نقطه را نیز مطابق با همین روش پیدا کرده و در تصویر فوریه با میانگین 4 خانه همسایه خود جایگزین می‌کنیم. و در نهایت از تبدیل فوریه مان، فوریه معکوس می‌گیریم تا به تصویر اصلاح شده برسیم:



روش‌های متفاوت و حتی ساده‌تری نیز ممکن است برای به دست آوردن نقاط اعوجاج وجود داشته باشد اما بنده با این ایده به پاسخ رسیدم.