

به نام خدا

پردازش تصویر

تمرین شماره ۲

تبدیلات شدت روشنایی و فیلترگذاری مکانی

امین سخایی

۹۷۳۳۰۳۶

استاد درس

دکتر حامد آذرنوش

## سوال شماره ۱:

ورودی برنامه تصویر brains.png است که به صورت خاکستری ۸ بیتی آنرا فراخوانی می کنیم.

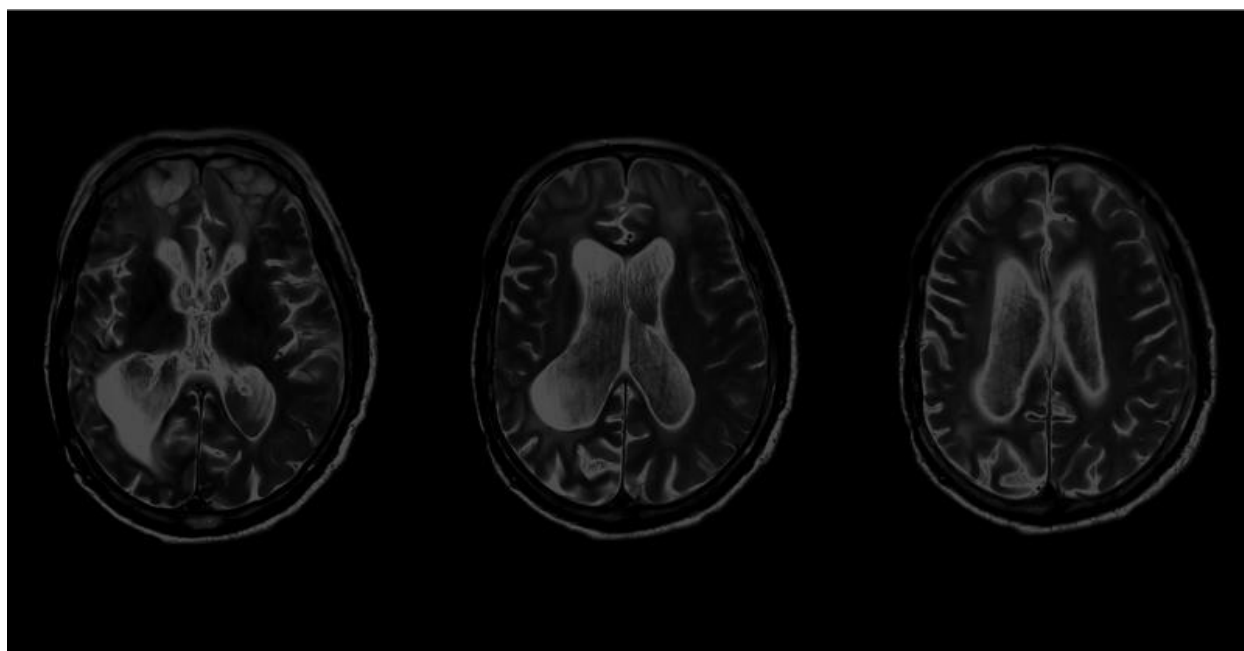
بخش ب-ز) برای تنظیم کنتراست تصویر با استفاده از تبدیل توانی از فرمول زیر و تکرار آنرا برای هر پیکسل استفاده میکنیم:

$$s = cr^\gamma$$

$$c = (l - 1)^\gamma$$

که  $s$  مقدار شدت خروجی و  $r$  شدت پیکسل ورودی است.

$$\gamma = 1.5$$



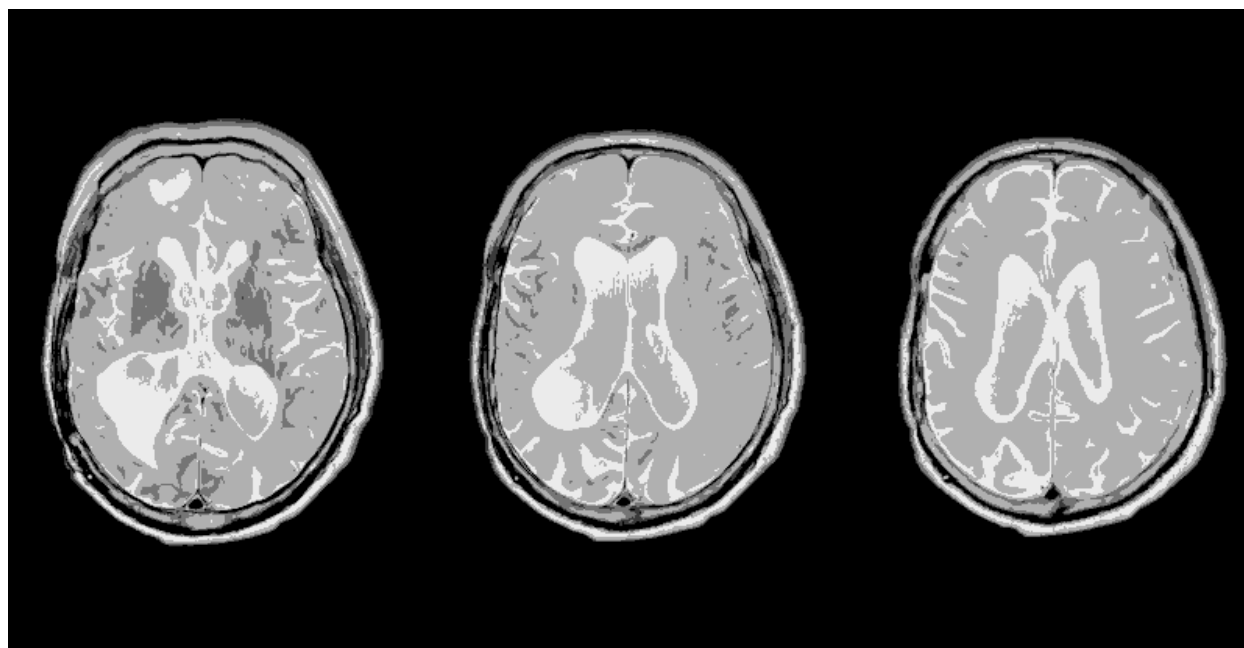
برای تنظیم کنتراست تصویر با استفاده از تبدیل لگاریتمی از فرمول زیر و تکرار آنرا برای هر پیکسل استفاده میکنیم:

$$s = c \log_k(1 + r)$$

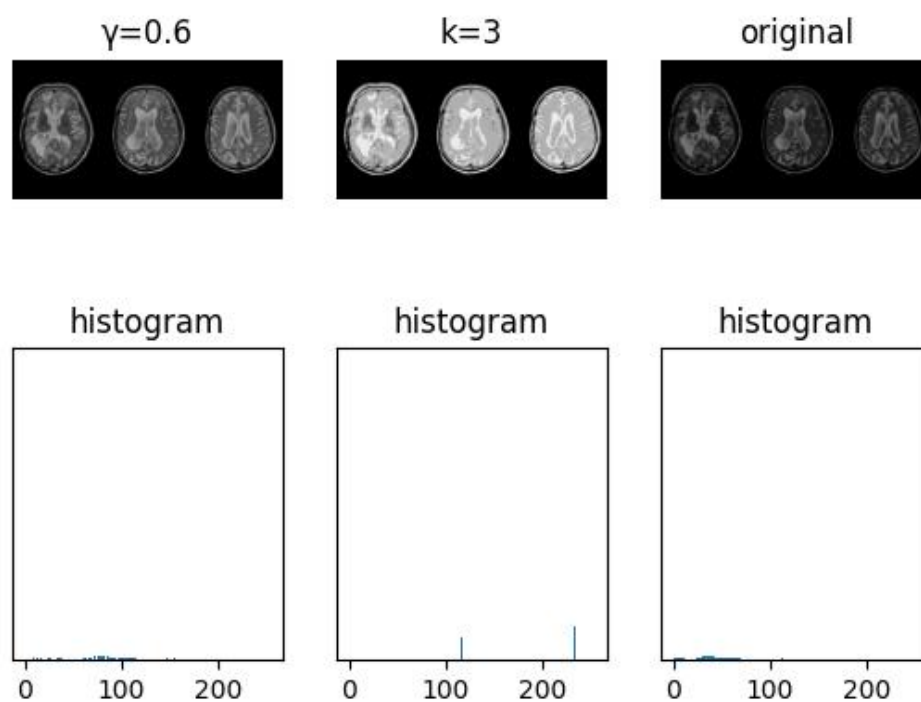
$$c = \frac{l - 1}{\log_k(l)}$$

که  $s$  مقدار شدت خروجی و  $r$  شدت پیکسل ورودی است.

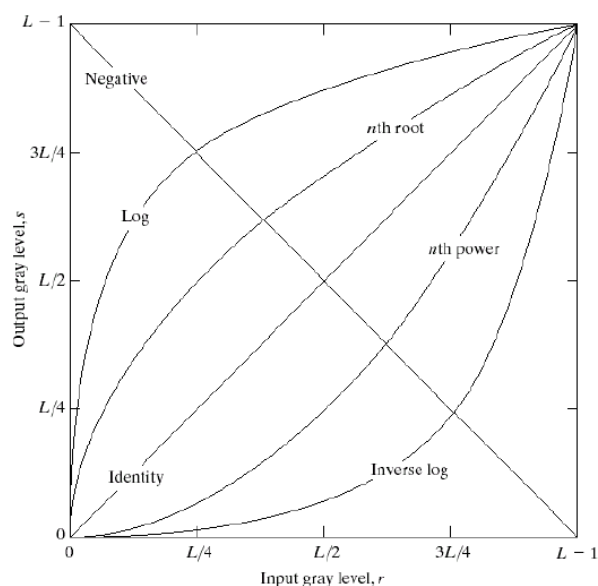
$$k = 10$$



نمایش تصاویر خروجی و نمودار فراوانی آنها:



## بخش ج)



تبدیل **power law** یک محدوده ی باریک از مقادیر تاریک ورودی را روی طیف وسیع تری نگاشت می کند. اگر  $0 < \gamma < 1$  باشد مقادیر تیره، تیره تر و مقادیر روشن، به سمت تیرگی می روند. اگر  $\gamma > 1$  باشد مقادیر تیره، روشن و مقادیر روشن، روشن تر می شوند.

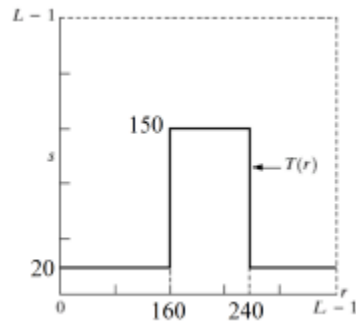
تبدیل **log** یک محدوده ی باریک از مقادیر با شدت خاکستری پایین را به سطح وسیع تر از خاکستری نگاشت می کند و برای گسترش مقادیر تاریک در یک تصویر به کار می رود.

تبدیل **log** زمانی مفید است که تصویر ورودی دارای مقادیر خاکستری با شدت بالا باشد.

در تصویر **brains.png** تبدیل **power law** نتیجه بهتری دارد زیرا فراوانی شدت پیکسل ها در کل تصویر پخش می شود و خروجی بهتری ایجاد می کند.

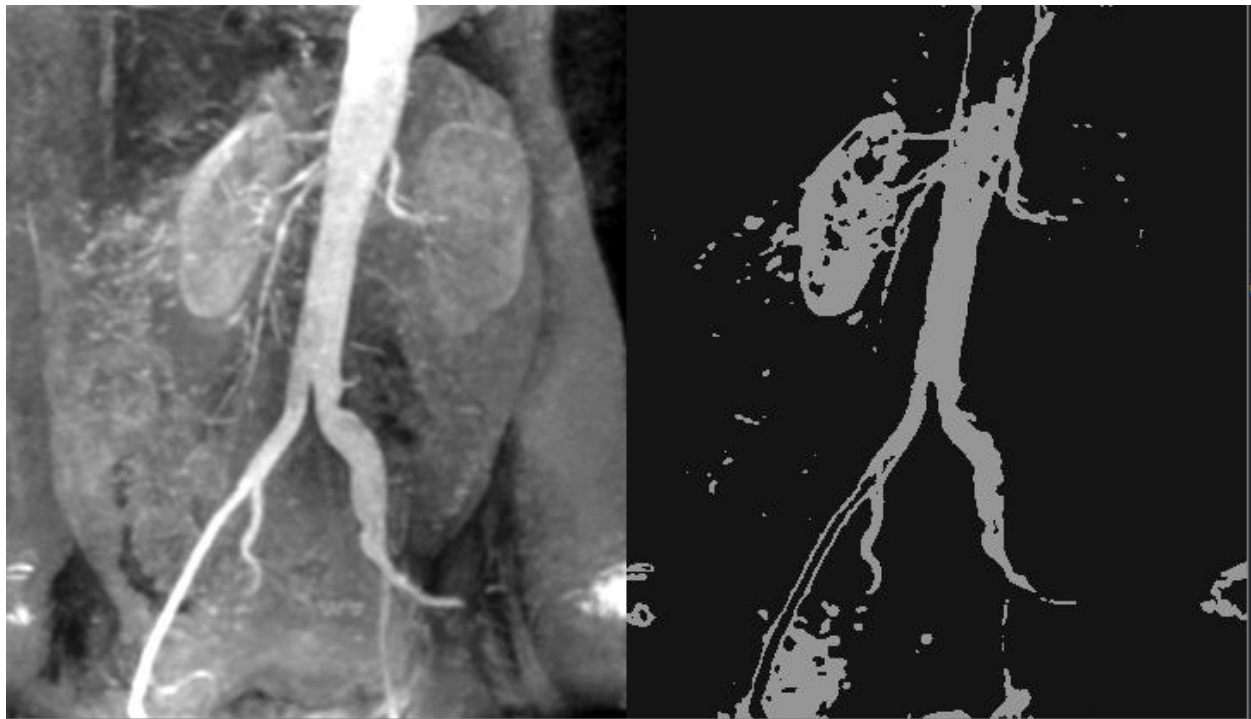
## سوال شماره ی ۲:

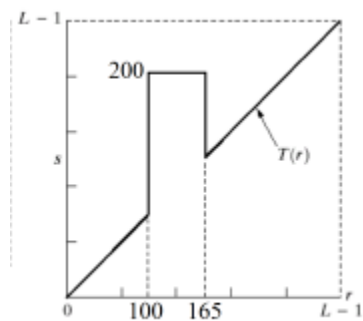
ورودی برنامه تصویر kidney.tif است که آنرا فراخوانی می کنیم.



شدت های بین ۱۶۰ تا ۲۴۰ به مقدار ۱۵۰ و بقیه شدت ها به مقدار ۲۰ می رسند و این عمل را به وسیله `vectorize` کردن و دستور شرطی انجام می دهیم. در نهایت مقادیر بین ۱۶۰ تا ۲۴۰ هایلایت می شوند و بقیه تاریک می شوند.

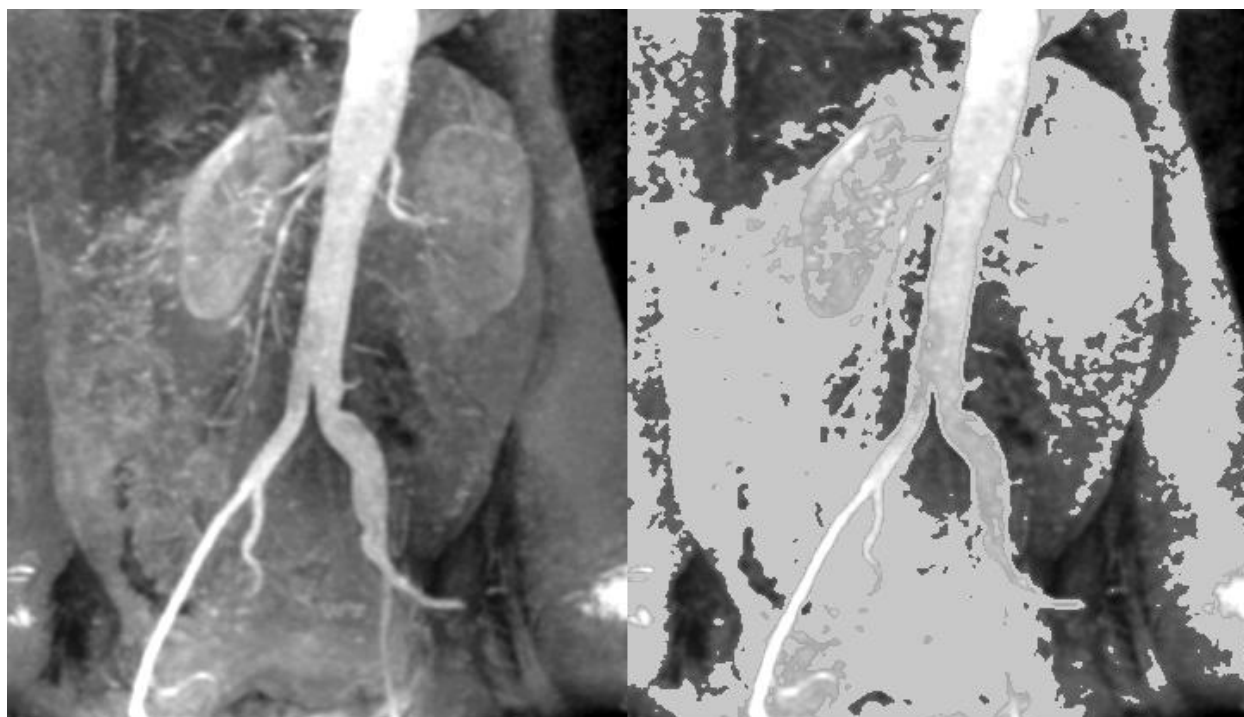
خروجی :





مقادیر بین ۱۰۰ تا ۱۶۵ به ۲۰۰ تبدیل می شوند و بقیه بدون تغییر می مانند. اجرای این روش مانند حالت قبل است. محدوده بین ۱۰۰ تا ۱۶۵ هایلایت می شود.

خروجی:



### سوال شماره ۳:

ورودی برنامه سه تصویر Bright.tif و Dark.tif و Lowcontrast.tif هستند که آنها را فراخوانی می کنیم.

**بخش الف)** برای تبدیل فراوانی نرمال شده به بردار تابعی طراحی میکنیم که به وسیله ی حلقه، تمامی شدت های متغیر ورودی را چک کند، در هر مرحله به مقدار متناظر آن شدت اضافه شود که خروجی آن یک آرایه با تعداد شدت های موجود در متغیر ورودی است.

**بخش ب)** رابطه نرمال سازی نمودار فراوانی به شکل زیر است:

$$\text{Histogram: } h(r_k) = n_k$$

$$\text{Gray levels: } r_k$$

$$\text{Repetition of each gray level: } n_k$$

$$\text{Normalized histogram: } p(r_k) = \frac{n_k}{MN}$$

ابتدا بوسیله ی تابع بخش قبل بردار فراوانی تصویر را بدست می آوریم. سپس با استفاده از تابع cdf تابع توزیع حجمی آنرا بدست می آوریم. حال بوسیله ی یک حلقه و فرمول بالا برای تک تک شدت ها آنرا تکرار می کنیم و پاسخ نهایی را به ۰ تا ۲۵۵ نگاشت می کنیم.

برای تبدیل بردار فراوانی نرمال شده به شکل زیر عمل می کنیم:

ابتدا در تابع قسمت الف یک دیکشنری تعریف میکنیم که مقادیر آن شدت پیکسل تصویر و فراوانی آن است.

در تابع cdf فراوانی به توزیع حجمی هر شدت تبدیل و در مرحله ی بعد با شدت جدید تصویر جایگزین می شود.

حال با استفاده از یک حلقه یک تصویر با شدت های به روز شده تولید می کنیم.

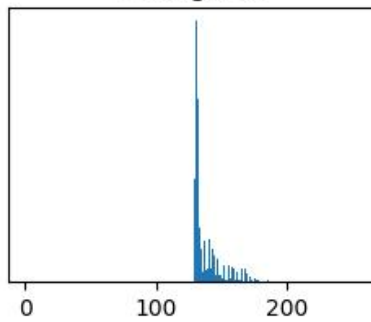
**بخش ج)**

خروجی :

lowcontrast



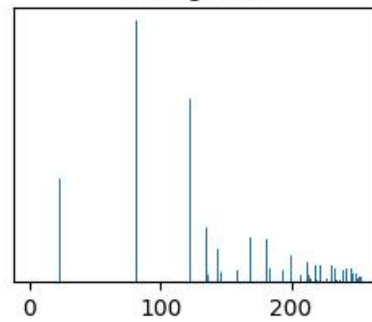
Histogram



lowcontrast improved



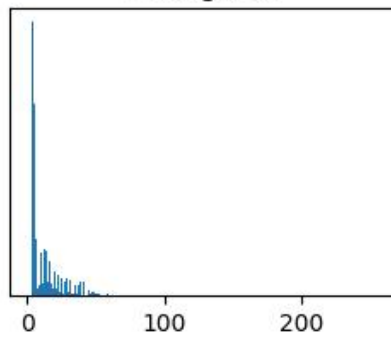
Histogram.i



Dark



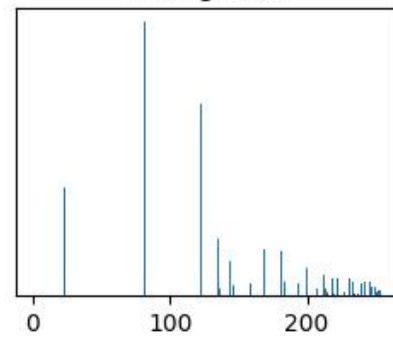
Histogram



Dark improved



Histogram.i





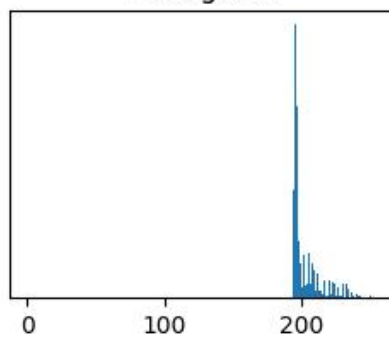
Bright



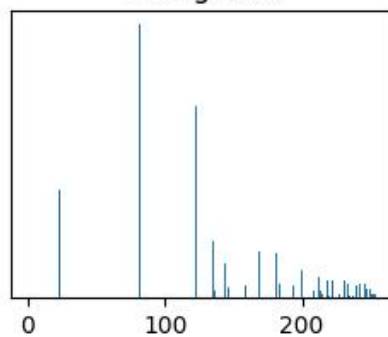
Bright improved



Histogram



Histogram.i



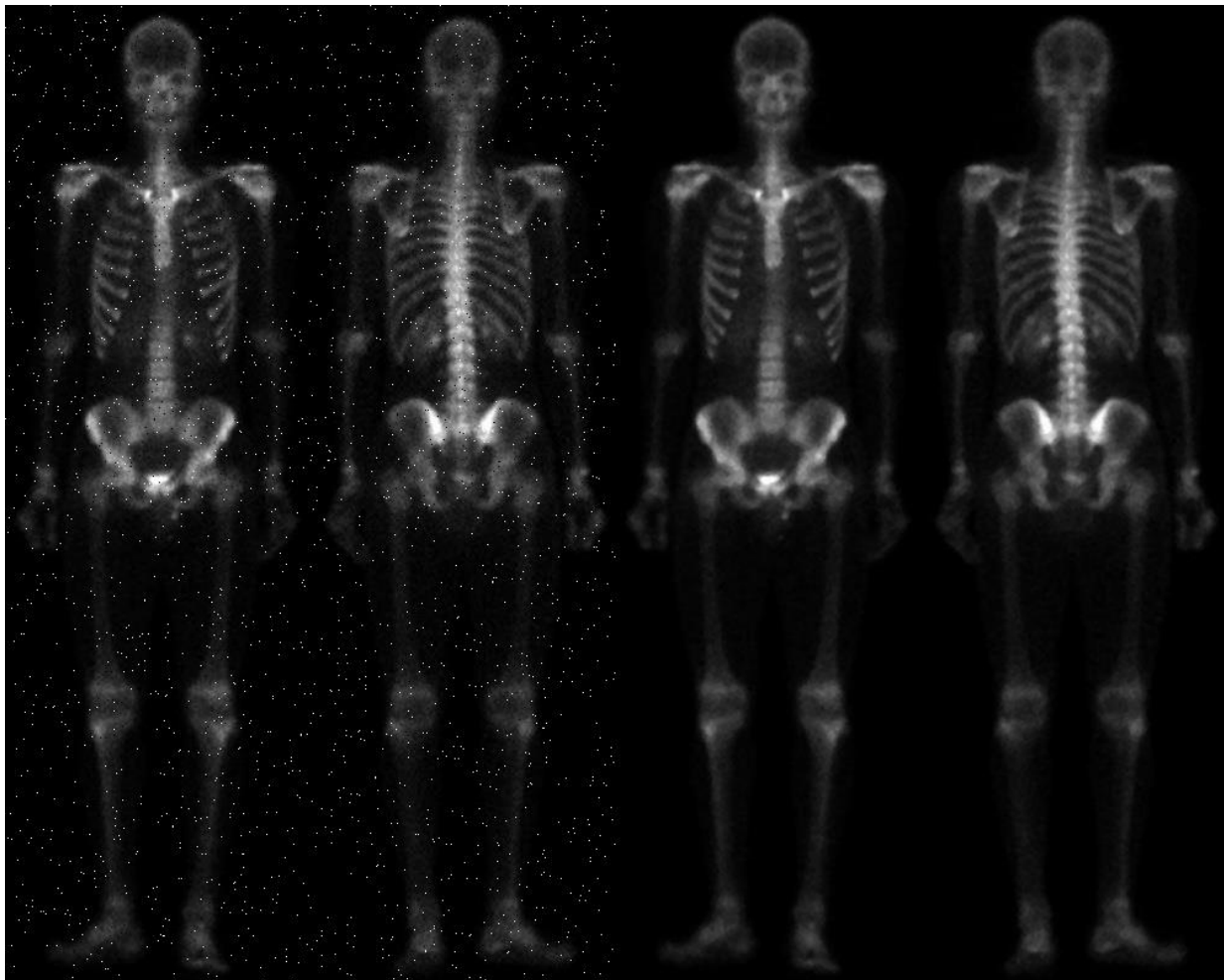
#### سوال شماره ۴:

ورودی برنامه تصویر bone-scan.png است که آنرا فراخوانی می کنیم.

بخش الف) با توجه به ویژگی های خواسته شده تابع filter را تعریف می کنیم.

بخش ب) با متغیرهای img و 'median' تابع را فراخوانی می کنیم. ابتدا به تصویر ورودی یک پیکسل حاشیه به شکل مرزی استفاده می کنیم. فیلتر median به این صورت عمل می کند که در هر مرحله ابتدا یک برش  $3 \times 3$  از تصویر جدا کرده و مقدار پیکسل در تصویر جدید برابر با درایه  $[2, 2]$  تصویر برش خورده است.

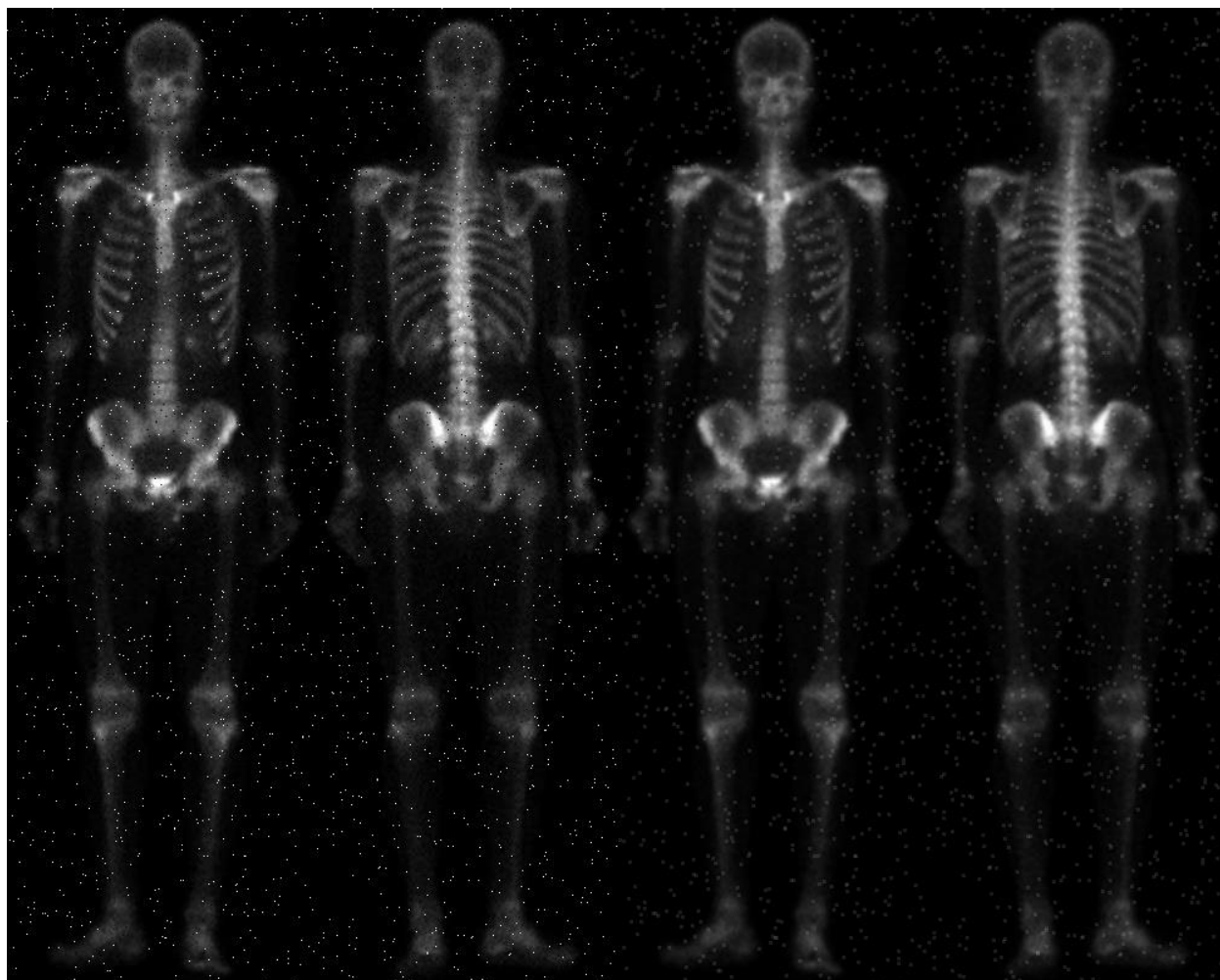
خروجی:



نویز نمک فلفلی به طور کامل حذف شده است.

بخش ج) با متغیرهای `img` و `x` تابع را فراخوانی می‌کنیم. ابتدا به تصویر ورودی یک پیکسل حاشیه به شکل مرزی استفاده می‌کنیم. فیلتر میانگین‌گیر به این صورت عمل می‌کند که در هر مرحله ابتدا یک برش  $3 \times 3$  از تصویر جدا کرده و مقدار پیکسل در تصویر جدید برابر میانگین ضرب دو ماتریس است.

خروجی:



از این فیلتر برای نرم کردن تصویر استفاده می‌شود.

**بخش د)** فیلتر میانگین گیر به این صورت عمل می کند که در هر مرحله ابتدا یک برش  $3 \times 3$  از تصویر جدا کرده و مقدار پیکسل در تصویر جدید برابر حاصل جمع ضرب درایه های دو ماتریس است. برای این منظور تابعی با مشخصات گفته شده تعریف می کنیم.

خروجی:

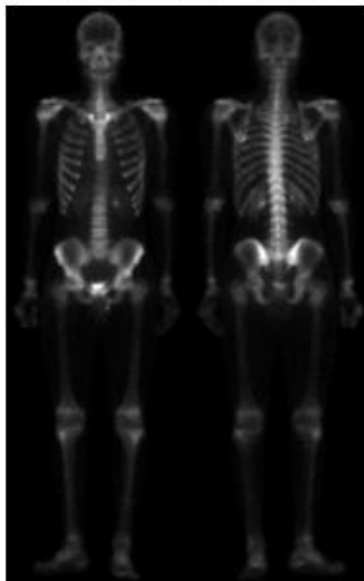
False

(اشتباه این قسمت رو متوجه نشدم)

**بخش ه)** نمودار پویای اثر برآیند ماسک لاپلاسیان بر تصویر اصلی با ضرایب متفاوت را ایجاد میکنیم.

خروجی:

Adding laplacian mask to original image with varying C



C  0

Reset

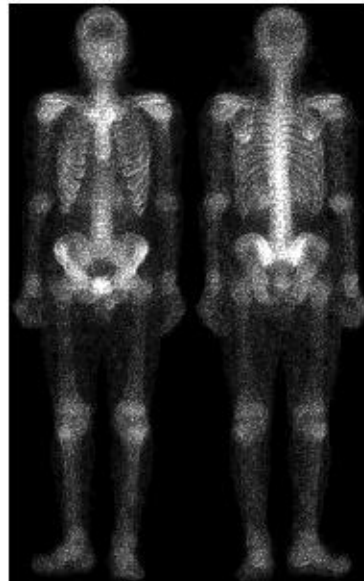
فرمول گفته شده در سوال با فرمول تولید تصویر شارپ برابر است.

$$new = img_b + c \times laplacian$$

برای تولید تصویر شارپ، پالس های موجود در تصویر را تیز تر میکنیم که این کار با افزایش شدت انجام می شود. بنابراین تصویر را با ضربی در لاپلاسین فیلتر کرده و با تصویر فیلتر شده (لاپلاسین) قبلی جمع می کنیم.

اگر  $C > 0$  باشد پالس های مخالف با هم جمع میشوند در نتیجه شدت های پایین بیشتر و شدت های بالا کمتر میشوند و مخالف با خواسته ی ماست در نتیجه تصویر مناسبی ایجاد نمیشود. حال هرچه مقدار  $C$  بیشتر شود وضوح تصویر افت می کند.

Adding laplacian mask to original image with varying C

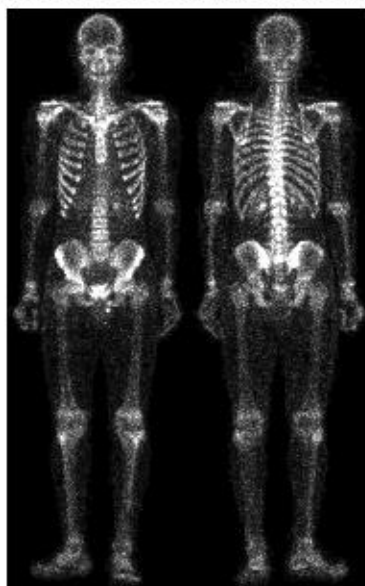


C  20

Reset

اما اگر  $C < 0$  باشد پالس های موافق با هم جمع می شوند و در نتیجه ی آن تصویر شارپ تولید می شود. حال هرچقدر مقدار  $C$  کوچکتر شود تصویر شارپ تری تولید خواهد شد.

Adding laplacian mask to original image with varying  $C$



$C$   -20

Reset