درس سیگنال و سیستم ها
پروژه-فاز اول

سما نادعلی – ۴۰۰۱۰۲۱۳۶

سوال۱:

روش اول:

کد زده شده برای این قسمت به صورت زیر است:

```
%Q1
%Part 1:
clc;
close all;
clear;
image=imread('D:\term\term 4\signal\project\phase 1\SS-Project-Phase1\q1.jpg');
k=1.3;

image1=im2double(image(:,:,1));
image1_fft=fft2(image1);
image1_fft_real=real(image1_fft(1:size(image1,1),1:size(image1,2)));
Hhp=1-1./(1+(10^6)./((abs(linspace(-1,1,size(image1,1))).^2)));
image1_fft_filtered=Hhp.*image1_fft;
image1_fft_sharpened=k.*image1_fft_filtered+1;
image1_sharpened=ifft2(image1_fft_sharpened);
image1_sharpened=real(image1_sharpened(1:size(image1,1),1:size(image1,2)));

image2=im2double(image(:,:,2));
image2_fft=fft2(image2);
image2_fft_real=real(image2_fft(1:size(image2,1),1:size(image2,2)));
Hhp=1-1./(1+(10^6)./((abs(linspace(-1,1,size(image2,1))).^2)));
image2_fft_filtered=Hhp.*image2_fft;
image2_fft_sharpened=k.*image2_fft_filtered+1;
image2_sharpened=ifft2(image2_fft_sharpened);
image2_sharpened=real(image2_sharpened(1:size(image2,1),1:size(image2,2)));
```

```
image3=im2double(image(:,:,3));
image3_fft=fft2(image3);
image3_fft_real=real(image3_fft(1:size(image3,1),1:size(image3,2)));
Hhp=1-1./(1+(10^6)./((abs(linspace(-1,1,size(image3,1))).^2)));
image3_fft_filtered=Hhp.*image3_fft;
image3_fft_sharpened=k.*image3_fft_filtered+1;
image3_sharpened=ifft2(image3_fft_sharpened);
image3_sharpened=real(image3_sharpened(1:size(image3,1),1:size(image3,2)));

image_sharpened=zeros(565,565,3);
image_sharpened(:,:,1)=image1_sharpened;
image_sharpened(:,:,2)=image2_sharpened;
image_sharpened(:,:,3)=image3_sharpened;

image_fft_real(:,:,1)=image1_fft_real;
image_fft_real(:,:,2)=image2_fft_real;
image_fft_real(:,:,3)=image3_fft_real;

imwrite(image_fft_real , 'q1-res1.jpg' , 'jpg');
imwrite(image_sharpened , 'q1-res2.jpg' , 'jpg');
```

در این قطعه کد، ابتدا تصویر مورد نظر را می خوانیم. این تصویر به صورت یک آرایه ۵۶۵ در ۵۶۵ در ۳ توصیف می شود که دو بعد اول آن(۵۶۵*۵۶۵) نشان دهنده موقعیت مکانی پیکسل مربوطه است و بعد سوم، نشان دهنده رنگ آن rgb است. بنابراین کاری که انجام می دهیم این است که ابتدا بخش مربوط به هر رنگ را جدا کرده، کار های مورد نظر را روی آن انجام می دهیم و در نهایت، با به هم متصل کردن ۳ آرایه حاصل، تصویر نهایی را ایجاد می کنیم.
برای آرایه مربوط به هر یک از رنگ های، ابتدا به کمک تابع fft2 از آن تبدیل فوریه دو بعدی می گیریم. از آنجایی که با تبدیل فوریه گرفتن، ممکن است در اطراف تصویر نیز عباراتی ایجاد شود، با کد خط ۱۱، تنها بخشی از تبدیل را که مربوط به عکس است را انتخاب کرده و نگه می داریم.

سپس یک فیلتر بالاگذر طراحی کرده و پاسخ فرکانسی آن را ایجاد می کنیم. مطابق آنچه که در شرح روش آمده است، این فیلتر را روی عکس اعمال کرده، در ضریب k ضرب کرده و در نهایت با ۱ جمع می کنیم. تا تبدیل فوریه تصویر خروجی ایجاد شود. حالا می توانیم تبدیل فوریه معکوس ifft گرفته و مجدد تنها بخشی از آن را که مورد نظرمان هست را انتخاب و ذخیره می کنیم.

به طور شهودی می توان دلیل عملکرد این فرایند را اینگونه توجیه کرد که ابتدا تصویر را به حوزه فرکانس برده، با اعمال فیلتر بالا گذر، تغییرات ناگهانی آن را تشخیص داده، با ضرب در یک ثابت، این تغییرات را تشدید کرده و با تبدیل فوریه خود عکس جمع می کنیم.از آنجایی که تبدیل فوریه خطی است می توان گفت مانند این است که ضریبی از تغییرات عکس را به خود عکس اضافه کرده باشیم. بنابراین عکس شارپ می شود.

بعد از انجام این عملیات روی هر سه رنگ، باید آرایه نهایی مربوط به عکس را ایجاد کنیم. پس هر رنگ را در بخشی از آرایه نهایی قرار می دهیم. حال تصویر نهایی آماده است و می توان آن را نمایش و یا ذخیره کرد.

در کنار این فایل، تصویر تبدیل فوریه و تصویر شارپ شده به فرمت بیان شده آمده است. همچنین اندازه و فاز تبدیل فوریه عکس نیز به صورت جداگانه قرار دارند.

ضریب k در تصیر خروجی آمده، ۱/۳ گذاشته شده است.

روش دوم:

کد زده شده به صورت زیر است:

```
%%
%part2
clc;
close all;
clear;
image=imread('D:\term\term 4\signal\project\phase 1\SS-Project-Phase1\q1.jpg');
k=0.3;

image1=image(:,:,1);
image1_fft=fftshift(fft2(double(image1)));
%calculate the high pass filter using the given formula:
%size(image1);  size(image1)=565 by 565
u=-282:1:282;
v=u;
[u,v]=meshgrid(u,v);
H=4.*pi.*pi.*(u.^2+v.^2).*image1_fft;
image1_final=abs(ifft2(fftshift(H)));

image2=image(:,:,2);
image2_fft=fftshift(fft2(double(image2)));
H=4*pi*pi*(u.^2+v.^2).*image2_fft;
image2_final=abs(ifft2(fftshift(H)));

image3=image(:,:,3);
image3_fft=fftshift(fft2(double(image3)));
H=4*pi*pi*(u.^2+v.^2).*image3_fft;
image3_final=abs(ifft2(fftshift(H)));

final_image=zeros(565,565,3);
final_image(:,:,1)=image1_final;
final_image(:,:,2)=image2_final;
final_image(:,:,3)=image3_final;

max_image=max(final_image,[],"all");
min_image=min(final_image,[],"all");
final_image=(final_image-min_image.*ones(565,565,3)).*(255./(max_image-min_image));
final_image=k.*final_image+1.*double(image);

max_image=max(final_image,[],"all");
min_image=min(final_image,[],"all");
final_image=(final_image-min_image.*ones(565,565,3)).*(255./(max_image-min_image));
final_image=uint8(final_image);

imwrite(final_image , 'q1-res3.jpg' , 'jpg');
```

در این کد نیز مشابه روش قبل، تصویر را خوانده، رنگ های مختلف آن را جدا کرده، عملیات مورد نظر را روی آن انجام داده و مجددا رنگ ها را با هم ترکیب می کنیم تا تصویر مورد نظر ایجاد شود.

ابتدا تبدیل فوریه fft می گیریم و سپس آن را در فیلتری که گفته شده که در ادامه نیز می اورم ضرب می کنیم:

$f+kF^{-1}\{4\pi^2(u^2 + v^2)F(u, v)\}$

اما این عبارت را تا قبل ضرب در یک ثابت، در این مرحله انجام می دهیم.

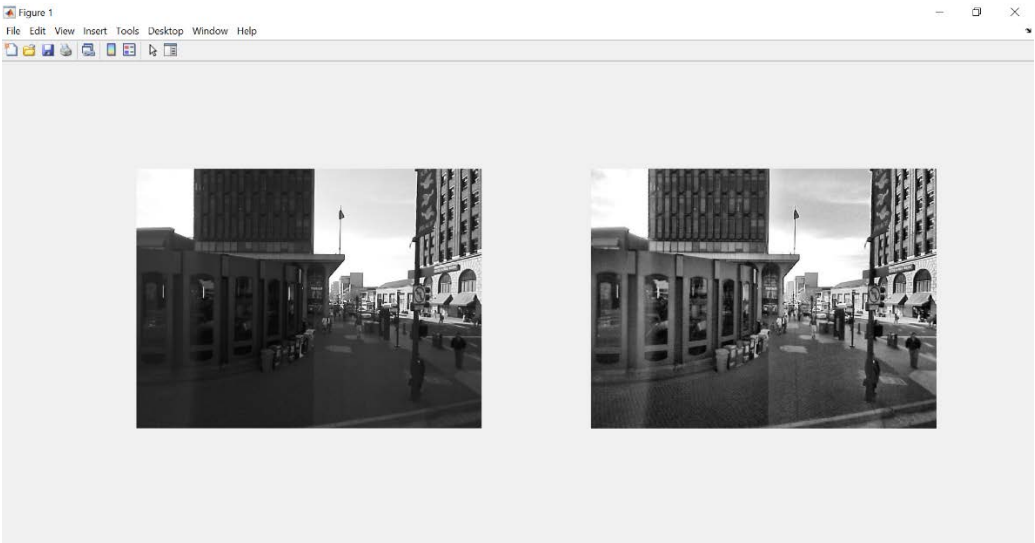
پس از اینکه این عملیات را برای رنگ های مختلف انجام داده و آرایه نهایی را که حاصل تجمع رنگ های مختلف هستند را تشکیل دادیم، مقادیر آرایه را یکبار نرمالایز می کنیم تا در محدوده ۰ تا ۲۵۵ که مربوط به تصویر است قرار گیرد، سپس بعد از ضرب در ضریب و جمع با تصویر اولیه، دوباره آن را نرمالایز می کنیم. حال تصویر نهایی آماده است و می توان آن را ذخیره کرد.
تصویری که کنار این فایل ارسال شده است، به ازای ثابت ۰/۳ است.
اگر این روش را هم بخواهیم توجیه کنیم، می توان این گونه گفت که پس از بردن عکس به حوزه فرکانس، یک فیلتر بالاگذر روی آن اعمال کرده تا تغییرات آشکار شوند سپس این تغییرات را به حوزه زمان برده، در ضریبی جهت تشدید ضرب کرده و با عکس اصلی جمع می کنیم. این گونه عکس را می توان شارپ تر کرد.

سوال ۲:

کد زده شده برای این سوال به صورت زیر است:

```
93    %%
94    %Q2:
95    clc;
96    close all;
97    clear;
98    image=imread('D:\term\term 4\signal\project\phase 1\SS-Project-Phase1\pic.jpg');
99    image_adjusted=adapthisteq(image);
100   subplot(1,2,1);
101   imshow(image);
102   subplot(1,2,2);
103   imshow(image_adjusted);
104   imwrite(image_adjusted , 'q2_output.jpg' , 'jpg');
105
```

نتیجه حاصل نیز به صورت زیر خواهد بود که در آن تصویر سمت چپ عکس اصلی و تصویر سمت راست تصویر تغییر داده شده است:

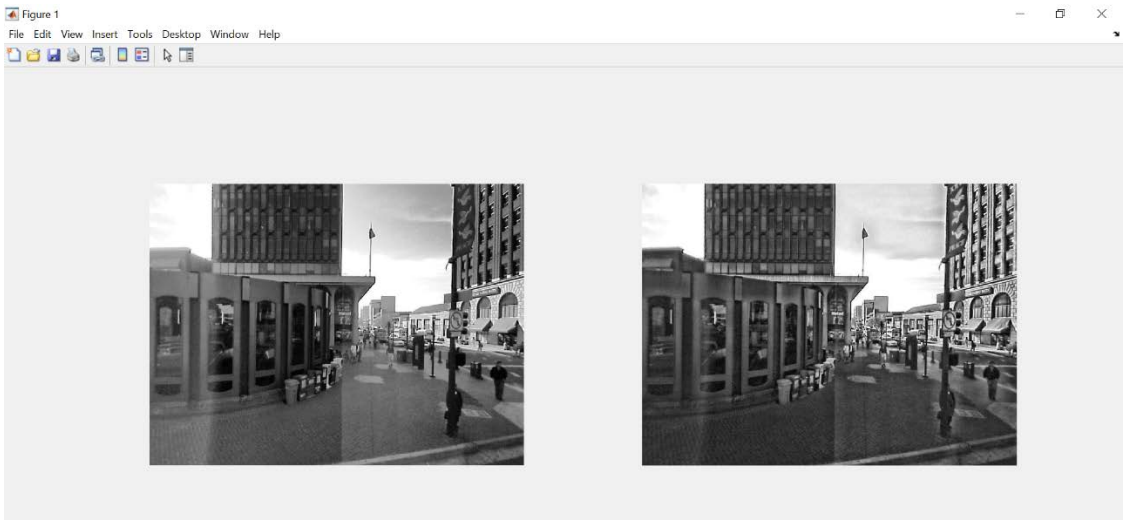


در کد بالا، از دستور adapthisteq استفاده شده است که عملکرد آن به صورت زیر است:

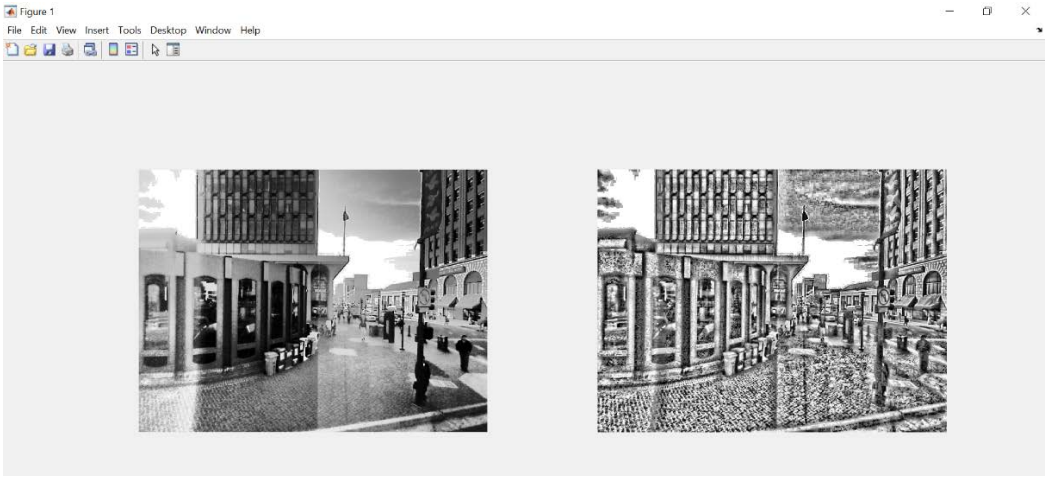
روش اصلی آن استفاده از contrast-limited adaptive histogram equalization است.
ابتدا تصویر ورودی آن باید به صورت gray_scale باشد. این تصویر، به کاشی(tile) هایی بدون هم‌پوشانی تقسیم بندی می شود که سایز این کاشی ها را می توان به کمک پارامتر numtiles مشخص کرد. برای مثال اگر کد زیر را اجرا کنیم خواهیم داشت:

```
%%
%Q2:
clc;
close all;
clear;
image=imread('D:\term\term 4\signal\project\phase 1\SS-Project-Phase1\pic.jpg');
image_adjusted=adapthisteq(image,"NumTiles",[2 2]);
subplot(1,2,1);
imshow(image_adjusted);
subplot(1,2,2);
image_adjusted=adapthisteq(image,"NumTiles",[32 32]);
imshow(image_adjusted);
%imwrite(image_adjusted , 'q2_output.jpg' , 'jpg');
```

خروجی به صورت زیر خواهد شد که در آن، تصویر سمت راست به ازای ابعاد کاشی ۳۲ در ۳۲ بوده و تصویر سمت چپ مربوط به کاشی ۲ در ۲:



اگر cliplimit را که در ادامه به آن می پردازیم افزایش داده و از حالت دیفالت که ۰/۰۱ است به ۰/۰۳ برسانیم این اختلاف بیشتر مشاهده می شود و خواهیم داشت:

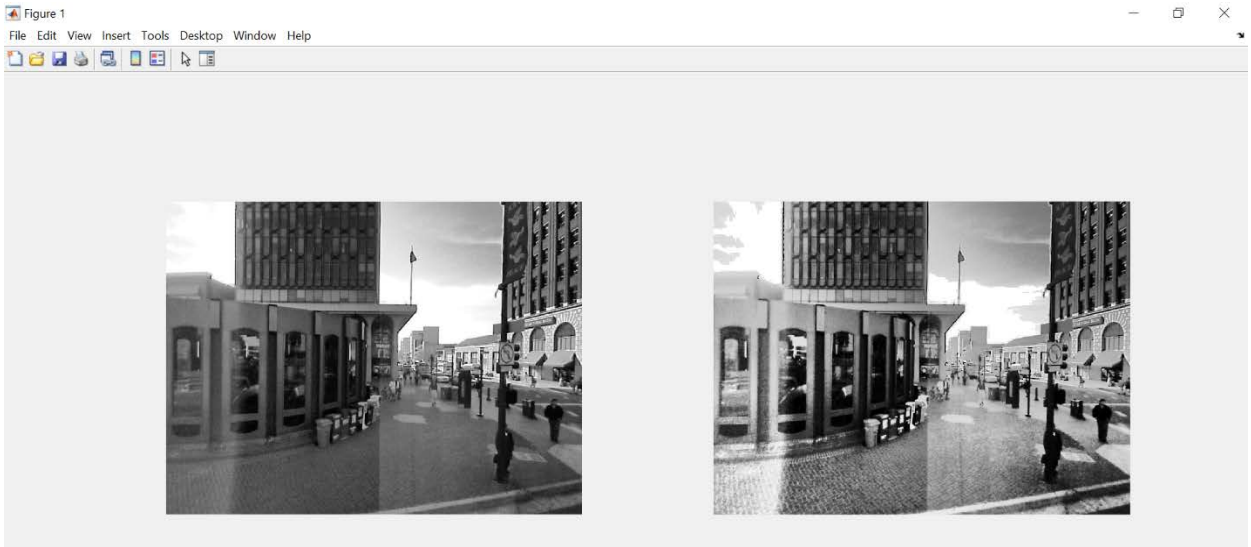


حالت دیفالت این پارامتر ۸ در ۸ است و ما ۲ در ۲ ست می کنیم تا کیفیت تصویر نهایی بیشتر شود.

سپس برای مقادیر پیکسل ها در هر کاشی، cdf(تابع توزیع تجمعی) تشکیل داده(با استفاده از نمودار هیستوگرام) و با توجه به cdf به دست آمده، مقدار شدت هر پیکسل بازسازی می شود. اما از آنجایی که ممکن است پیکسل های یک کاشی تماما تاریک یا روشن بوده و cdf مربوطه باعث شود که مقادیر پیکسل ها در یکی از دو سر هیستوگرام تجمع کنند و در نتیجه خطر فشرده شدن یا جایگزینی هیستوگرام وجود داشته باشد، از پارامتر cliplimit استفاده می شود تا اندازه باکس های هیستوگرام محدود شده و پیکسل هایی که بیشتر از محدوده مجاز باشند، متناسب با باکس های مجاور، توزیع می شوند.

مقدار این پارامتر باید عددی بین ۰ و ۱ باشد و مقدار دیفالت آن نیز ۰/۰۱ است. اگر کد زیر را زده و برای دو مقدار مختلف برای cliplimit تصویر را بازسازی کنیم، حاصل به صورت زیر شده که تصویر سمت چپ مربوط به پارامتر کمتر و تصویر سمت راست مربوط به پارامتر بیشتر است:

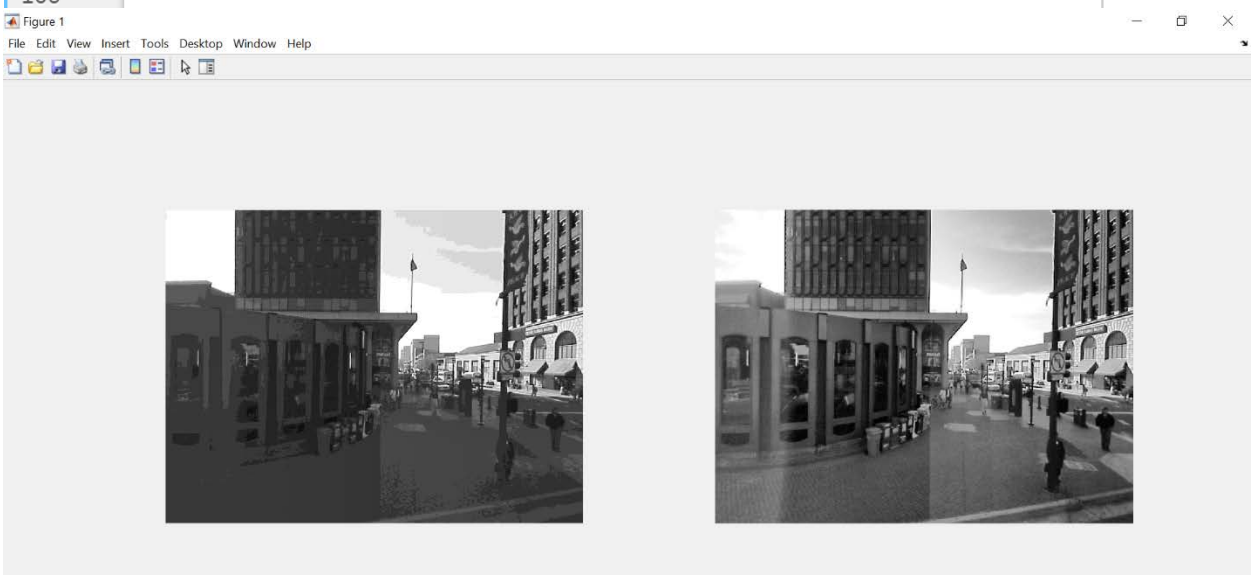
```
93    %%
94    %Q2:
95    clc;
96    close all;
97    clear;
98    image=imread('D:\term\term 4\signal\project\phase 1\SS-Project-Phase1\pic.jpg');
99    image_adjusted=adapthisteq(image,"NumTiles",[2 2],"ClipLimit",0.01);
100   subplot(1,2,1);
101   imshow(image_adjusted);
102   subplot(1,2,2);
103   image_adjusted=adapthisteq(image,"NumTiles",[2 2],"ClipLimit",0.3);
104   imshow(image_adjusted);
105   %imwrite(image_adjusted , 'q2_output.jpg' , 'jpg');
106
```



با سعی و خطا، به نظر آمد که مقدار ۰/۰۱۲ برای این پارامتر مناسب تر باشد.
همچنین برای جلوگیری از قابل مشاهده بودن مرز کاشی ها که به واسطه استفاده از هیستوگرام محلی به وجود می‌آید، یک پروسه smoothing نیز روی تصویر صورت می‌گیرد. در نهایت نیز تمام پیکسل های بازسازی شده کنار هم قرار گرفته و تصویر نهایی ایجاد می‌شود.
پارامتر های دیگری که می‌توان تنظیم کرد به صورت زیر است:

NBins: به کمک این پارامتر، میتوان تعداد باکس های نمودارم هیستوگرام را مشخص کرد. افزایش این پارامتر، می‌تواند باعث افزایش دقت این فرایند شود اما از طرفی دیگر، باعث افزایش حساسیت به نویز های تصویر می‌شود. در کد و نمونه خروجی زیر، تاثیر این پارامتر روی تصویر بررسی شده است:

```
93    %%
94    %Q2:
95    clc;
96    close all;
97    clear;
98    image=imread('D:\term\term 4\signal\project\phase 1\SS-Project-Phase1\pic.jpg');
99    image_adjusted=adapthisteq(image,"NumTiles",[2 2],"ClipLimit",0.012,"NBins",15);
100   subplot(1,2,1);
101   imshow(image_adjusted);
102   subplot(1,2,2);
103   image_adjusted=adapthisteq(image,"NumTiles",[2 2],"ClipLimit",0.012,"NBins",40000);
104   imshow(image_adjusted);
105   %imwrite(image_adjusted , 'q2_output.jpg' , 'jpg');
106
```

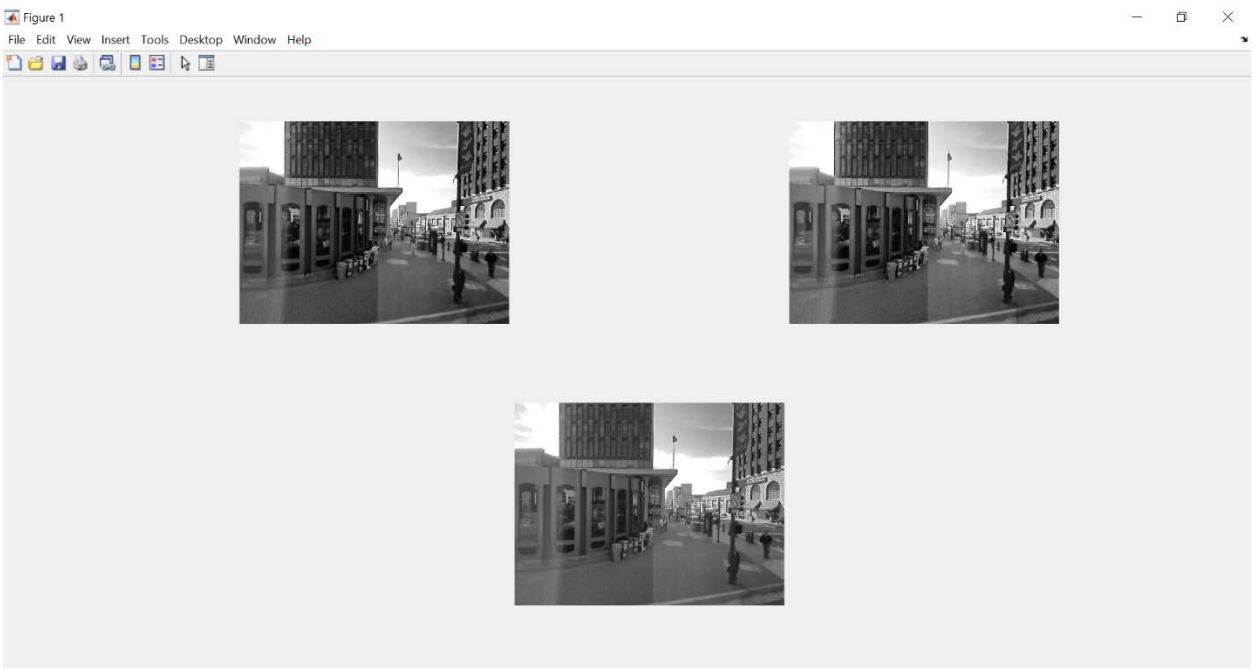


مقدار این پارامتر باید عددی مثبت انتخاب شود و دیفالت آن نیز ۲۵۶ است. که در اینجا عدد ۴۰۰۰۰ انتخاب شده است.

Range: این پارامتر، محدوده مقدار پیکسل های استفاده شده برای ایجاد هیستوگرام را مشخص می کند که می تواند برای جلوگیری از over-enhancement به کار آید.
برای تعیین این پارامتر، دو مقدار full و original داریم. اگر پارامتر را فول انتخاب کنیم، کل بازه خروجی انتخاب می‌شود(مثلا ۰ تا ۲۵۵ برای خروجی از نوع unit8) اما اگر original را انتخاب کنیم، با دادن بازه می‌توانیم مینیمم و ماکسیمم محدوده مورد نظر را مشخص کنیم.
دیفالت این پارامتر "full" است.

Distribution: این پارامتر برای تعیین شکل تابع تبدیل مقدار پیکسل ها بر اساس هیستوگرام است که دیفالت آن 'uniform' است اما آپشن های 'Rayleigh' و 'exponential' نیز داریم.

```
%%
%Q2:
clc;
close all;
clear;
image=imread('D:\term\term 4\signal\project\phase 1\SS-Project-Phase1\pic.jpg');
image_adjusted=adapthisteq(image,"NumTiles",[2 2],"ClipLimit",0.012,"NBins",40000,"Range","full","Distribution","exponential");
subplot(2,2,1);
imshow(image_adjusted);
subplot(2,2,2);
image_adjusted=adapthisteq(image,"NumTiles",[2 2],"ClipLimit",0.012,"NBins",40000,"Range","full","Distribution","uniform");
imshow(image_adjusted);
subplot(2,2,[3 4]);
image_adjusted=adapthisteq(image,"NumTiles",[2 2],"ClipLimit",0.012,"NBins",40000,"Range","full","Distribution","rayleigh");
imshow(image_adjusted);
%imwrite(image_adjusted , 'q2_output.jpg' , 'jpg');
```



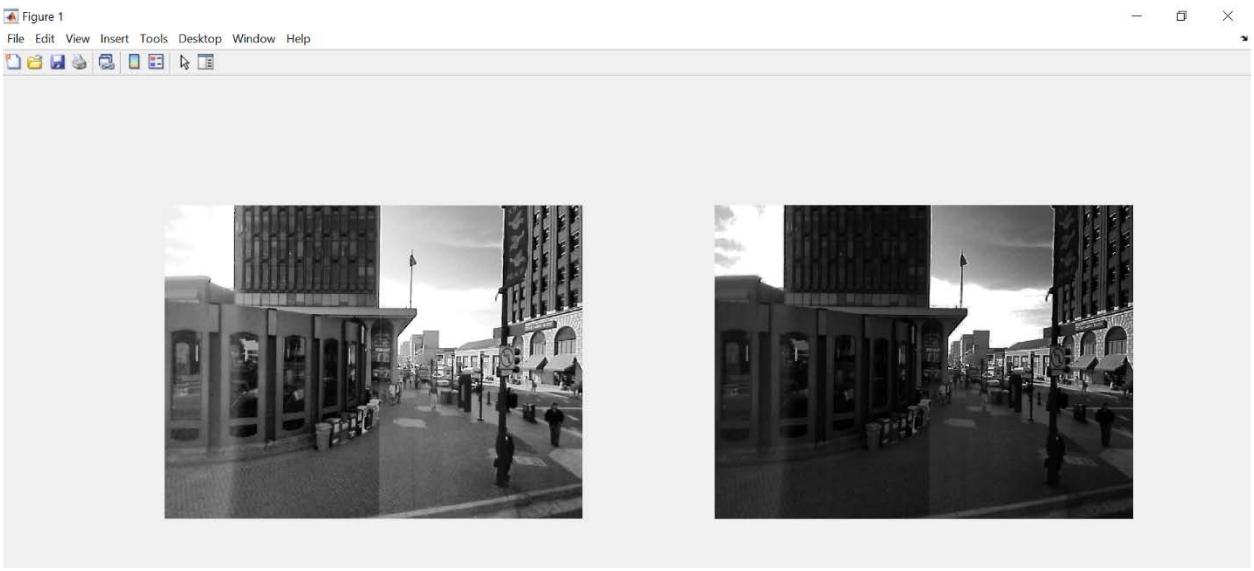
تصویر بالا سمت چپ مربوط به exponential، تصویر بالا سمت راست مربوط به uniform، و تصویر پایین مربوط به Rayleigh است.

حال اگر پارامتر را روی حالتی غیر از حالت دیفالت آن یعنی یونیفورم ست کنیم، پارامتر alpha را نیز می توانیم تغییر دهیم.

این پارامتر، قدرت کانترست ایجاد شده در تصویر را تنظیم می کند به گونه ای که هر چه مقدار آلفا بیشتر باشد، کانترست ایجاد شده نیز قوی تر می شود.

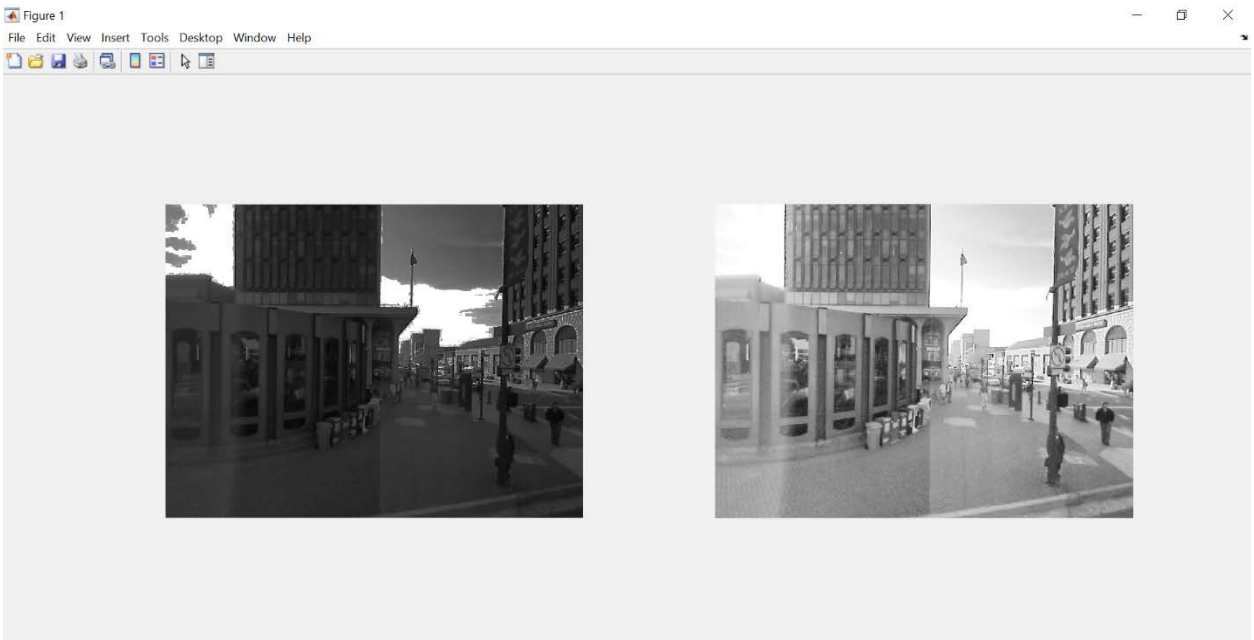
برای مثال با تغییر پارامتر آلفا برای حالت exponential خواهیم داشت:

```
%%
%Q2:
clc;
close all;
clear;
image=imread('D:\term\term 4\signal\project\phase 1\SS-Project-Phase1\pic.jpg');
image_adjusted=adapthisteq(image,"NumTiles",[2 2],"ClipLimit",0.012,"NBins",40000,"Range","full","Distribution","exponential","Alpha",0.1);
subplot(1,2,1);
imshow(image_adjusted);
subplot(1,2,2);
image_adjusted=adapthisteq(image,"NumTiles",[2 2],"ClipLimit",0.012,"NBins",40000,"Range","full","Distribution","exponential","Alpha",3);
imshow(image_adjusted);
% subplot(2,2,[3 4]);
% image_adjusted=adapthisteq(image,"NumTiles",[2 2],"ClipLimit",0.012,"NBins",40000,"Range","full","Distribution","rayleigh");
% imshow(image_adjusted);
%imwrite(image_adjusted , 'q2_output.jpg' , 'jpg');
```



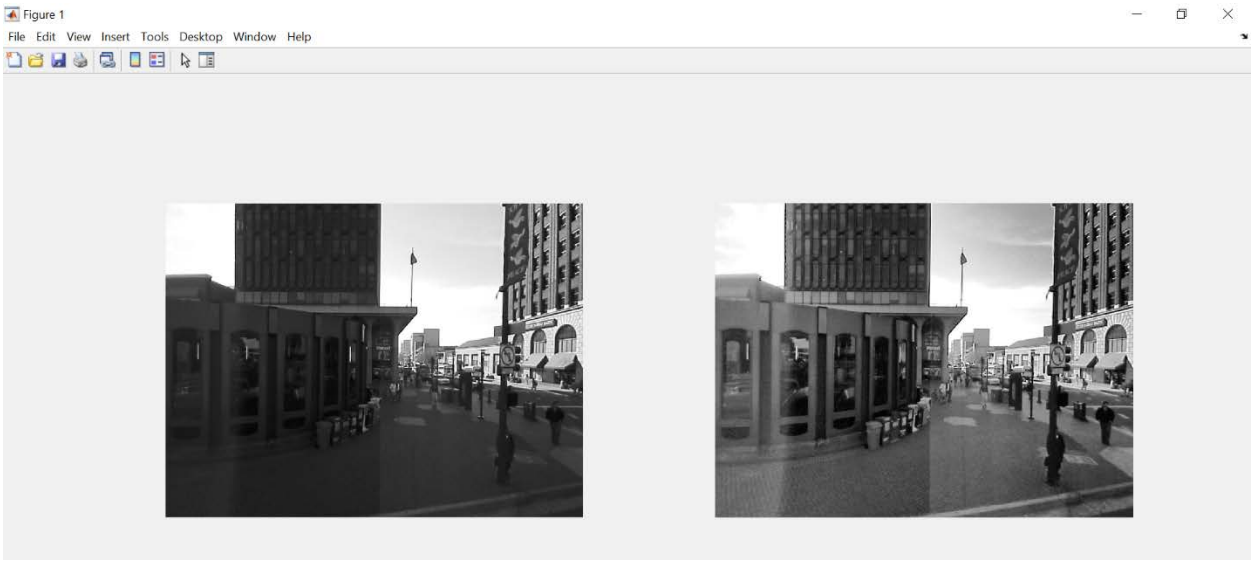
همچنین اگر این پارامتر را برای حالت Rayleigh تغییر دهیم خواهیم داشت:

```
93  %%
94  %Q2:
95  clc;
96  close all;
97  clear;
98  image=imread('D:\term\term 4\signal\project\phase 1\SS-Project-Phase1\pic.jpg');
99  image_adjusted=adapthisteq(image,"NumTiles",[2 2],"ClipLimit",0.012,"NBins",40000,"Range","full","Distribution","rayleigh","Alpha",0.2);
100 subplot(1,2,1);
101 imshow(image_adjusted);
102 subplot(1,2,2);
103 image_adjusted=adapthisteq(image,"NumTiles",[2 2],"ClipLimit",0.012,"NBins",40000,"Range","full","Distribution","rayleigh","Alpha",2);
104 imshow(image_adjusted);
105 % subplot(2,2,[3 4]);
106 % image_adjusted=adapthisteq(image,"NumTiles",[2 2],"ClipLimit",0.012,"NBins",40000,"Range","full","Distribution","rayleigh");
107 % imshow(image_adjusted);
108 %imwrite(image_adjusted , 'q2_output.jpg' , 'jpg');
```



در نهایت به نظر می رسد که بهترین تصویری که می توان با تغییر این پارامتر ها تولید کرد به صورت زیر باشد:

```
%%
%Q2:
clc;
close all;
clear;
image=imread('D:\term\term 4\signal\project\phase 1\SS-Project-Phase1\pic.jpg');
image_adjusted=adapthisteq(image,"NumTiles",[2 2],"ClipLimit",0.012,"NBins",40000,"Range","full","Distribution","exponential","Alpha",0.1);
subplot(1,2,2);
imshow(image_adjusted);
subplot(1,2,1);
imshow(image);
imwrite(image_adjusted , 'q2_output.jpg' , 'jpg');
```



سوال ۳:

کد زده شده برای این قسمت به صورت زیر است:

```
%%
%Q3:
clc;
close all;
clear;
einstein=imread('D:\term\term 4\signal\project\phase 1\SS-Project-Phase1\einstein.jpg');
marilyn=imread('D:\term\term 4\signal\project\phase 1\SS-Project-Phase1\marilyn.jpg');
size(einstein)
size(marilyn)
einstein=rgb2gray(einstein);
marilyn=marilyn(1:234,1:225);
einstein=imrotate(einstein, 10 , 'bilinear' , 'crop');
lp_filter=imgaussfilt(marilyn , 2);
hp_filter=1.3*einstein - imgaussfilt(einstein , 5);
result=1/2*lp_filter + 1/(2*1.3)*hp_filter;
subplot(2,3,1)
imshow(lp_filter);
subplot(2,3,2)
imshow(hp_filter);
subplot(2,3,4)
imshow(marilyn);
subplot(2,3,5)
imshow(einstein);
subplot(2,3,[3,6])
imshow(result);
```

در این قطعه کد، پس از خواندن دو عکس مورد نظر، اندازه آن ها را پرینت کردیم تا ببینیم که آیا یکسان هستند یا نه. که مشاهده می شود که عکس مرلین مونرو بزرگ تر است. پس تنها به اندازه ای از آن را نگه میداریم که هم اندازه با عکس انیشتین شود. سپس عکس انیشتین را می چرخانیم تا پس از انداختن دو تصویر روی هم، چشم دو عکس روی هم بیفتد. سپس فیلتر گوسی با سیگما ی ۲ را به عنوان فیلتر پایین گذر روی عکس مرلین اعمال می کنیم. برای اینکه روی عکس انیشتین فیلتر بالا گذر اعمال کنیم، ابتدا روی آن همان فیلتر گوسی پایین گذر را اعمال کرده و سپس حاصل را از عکس اصلی کم می کنیم تا تنها فرکانس های بالای آن باقی بماند. در اینجا برای اینکه عکس حاصل از فیلتر بالا گذر بهتر شود، فیلتر پایین گذر را از ضریبی از عکس اصلی کم کرده ایم.
در نهایت دو عکس حاصل را میانگین گیری کردیم که در این میانگین گیری، ضریبی که برای اعمال فیلتر استفاده کرده بودیم نیز دخیل شده است.
در نهایت نیز عکس اصلی مرلین، عکس پس از اعمال فیلتر پایین گذر، عکس اصلی انیشتین، عکس پس از اعمال فیلتر بالاگذر و در نهایت ترکیب دو عکس را نمایش دادیم. که خروجی به صورت زیر خواهد شد: