Contrast Adjustment

نسیم فانی

اطلاعات گزارش

تاریخ: 99/8/19

واژگان کلیدی:
هیستوگرام
Equalization
equalization محلی

چکیده

در بخش اول این تمرین، ابتدا به پیاده سازی الگوریتمی برای پیدا کردن هیستوگرام یک تصویر grayscale می‌پردازیم. سپس، تابعی برای equalize کردن هیستوگرام می‌نویسیم و در نهایت دو تابع hiseq و imadjust را با یکدیگر مقایسه می‌کنیم.

در بخش دوم، به بررسی histogram equalization محلی با اندازه‌ی پنجره‌های مختلف می‌پردازیم.

1-مقدمه

نوشتار حاضر، به بررسی روش پیاده‌سازی هیستوگرام، متعادل کردن آن به دو صورت سراسری و محلی و بررسی نتایج حاصل از استفاده از این روش‌ها در بهبود وضوح تصویر می‌پردازد.

2-توضیحات تکنیکال

2.1.1 برای رسم هیستوگرام یک تصویر، می‌بایست تعداد پیکسل‌های موجود از هر سطح را در کل تصویر بشماریم :

$$h(r_k) = n_k$$

برای این منظور به ازای هر سطح، کل تصویر را بررسی کرده و تعداد هر سطح را در یک ارایه نگه می‌داریم و در نهایت مقادیر بدست آمده را چاپ می‌کنیم.

2.1.2 برای متعادل‌سازی هیستوگرام مراحل زیر را طی می‌کنیم:

1. محاسبه‌ی فراوانی هر gray-level (این کار در مرحله محاسبه‌ی هیستوگرام تصویر توضیح داده شد.)
2. محاسبه‌ی PDF هر gray-level :

$$PDF(k) = \frac{n_k}{image\ total\ size}$$

3. محاسبه‌ی CDF هر gray-level :

$$CDF_x(k) = \sum_{j=0}^{k} PDF_x(x = k)$$

4. ضرب CDFها در بالاترین عدد gray-level و round کردن اعداد حاصل

2.1.3 تفاوت histeq با imadjust :

ازاین توابع برای تنظیم شدت و وضوح استفاده می‌شود. (contrast adjustment)

imadjust با نگاشت مقادیر input intensity تصویر به مقادیر جدید ، کنتراست تصویر را افزایش می دهد به گونه ای که به طور پیش فرض ، 1٪ از داده ها با شدت کم و زیاد داده های ورودی اشباع می شوند.

با استفاده ازاین تابع می توان دامنه تغییرات روشنایی یک تصویر را تغییر داد. شکل کلی این تابع بصورت زیر است:

J=imadjust(I,[low,high],[bottom, top])

آرگومان دوم برداری دو عنصری است که بیانگر دامنه حاوی روشنایی هایی از تصویر است که عملیات تنظیم شدت بر روی آنها باید اعمال گردد. آرگومان سوم، دامنه تغییرات جدید روشنایی برای نقاط فوق است.

histeq یکسان سازی هیستوگرام را انجام می دهد. با تبدیل مقادیر در یک تصویر شدت ، کنتراست تصاویر را افزایش می دهد تا هیستوگرام تصویر خروجی تقریباً با یک هیستوگرام مشخص مطابقت داشته باشد (توزیع یکنواخت به طور پیش فرض).

Imadjust به صورت خطی تصویر را scale می‌کند، درحالی‌که histeq این‌گونه نیست.

2.2.1 روش‌های مختلفی برای محاسبه هستوگرام متعدل محلی وجود دارد. به طور کلی این روش ها به دو دسته پوشا و غیرپوشا تقسیم می‌شوند. در پنجا به بررسی یکی از این روش‌ها می‌پردازیم.

برای محاسبه‌ی histogram equalization به صورت محلی، ابتدا padding zero به تمام sideهای ماتریس تصویر اضافه می‌کنیم.

سپس یک پنجره (برای مثال با اندازه‌ی 3*3) را در نظر می‌گیریم و آن را به گونه قرار می‌دهیم که اول عنصر ماتریس که در موقعیت (1,1) در مرکز قرار بگیرد. برای مثال:

$$B=\begin{bmatrix}0&0&0&0&0\\0&2&1&0&0\\0&0&1&3&0\\0&4&1&4&0\\0&0&0&0&0\end{bmatrix}$$

در ادامه، احتمال وقوع هر مقدار را به دست می‌آوریم. در مثال بالا این مقادیر به صورت زیر است :

P(0/4) = 6/9

P(1/4) = 2/9

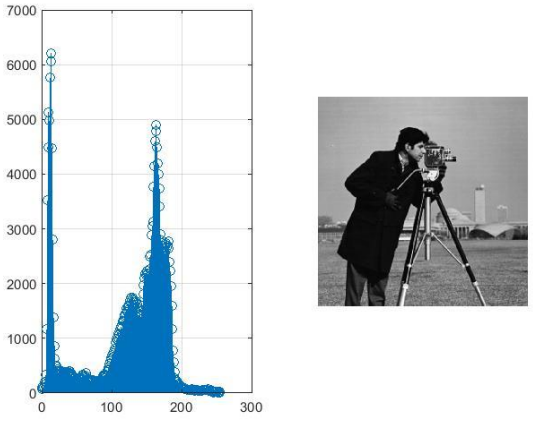
…

در نهایت با محاسبه‌ی CDF مقدار نهایی برای جایگزاری عدد 2 را به دست می‌آوریم.

به همین ترتیب می‌توانیم این مقادیر را به ازای تمامی عناصر ماتریس‌مان به‌دست آوریم.

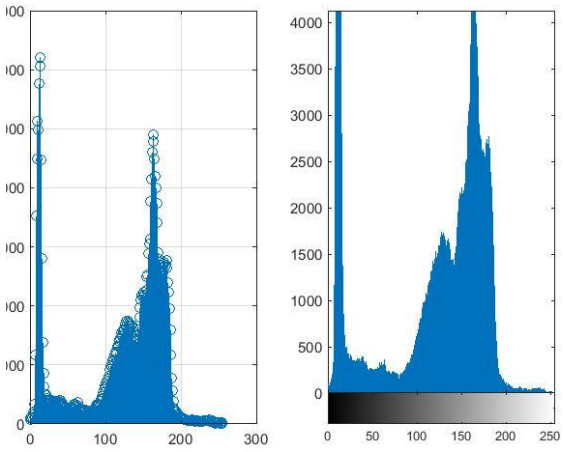
3-شکل‌ها، جدول‌ها و روابط (فرمول‌ها)

2.1.1 تابع رسم هیستوگرام خود را (تابع myHistogram) بر روی تصویر camera man اعمال می‌کنیم و با کمک تابع subplot آن‌ها را در کنار هم چاپ می‌کنیم:



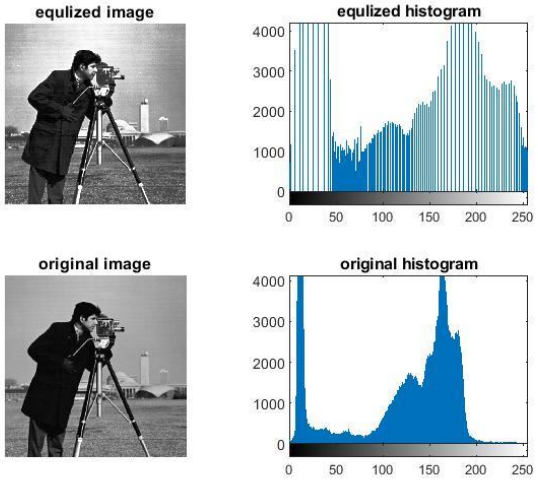
تصویر 1- تصویر camera man و هیستوگرام آن

برای اطمینان از صحت عملکرد تابع‌مان می‌توانیم هیستوگرام حاصل را با هیستوگرام رسم شده توسط تابع imhist مقایسه نماییم:



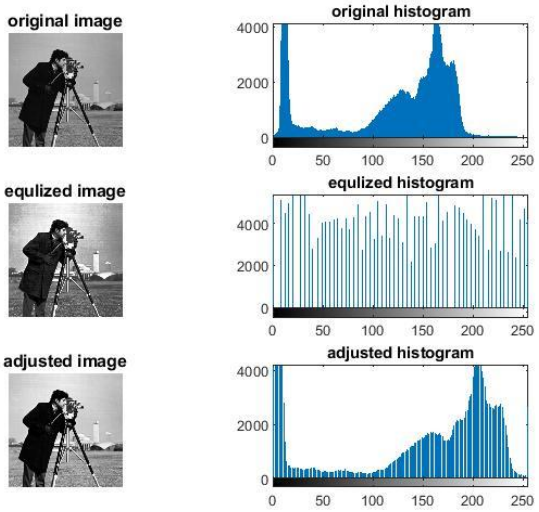
تصویر 2- سمت راست هیستوگرام حاصل از تابع imhist و سمت چپ هیستوگرام رسم شده توسط تابع myHistogram

2.1.2 تابع histogrameq را بر روی تصویر man camera اعمال میکنیم و به کمک نابع subplot تصویر و هیستوگرام متعادل شده را در کنار تصویر اصلی و هیستوگرامش نمایش می‌دهیم:



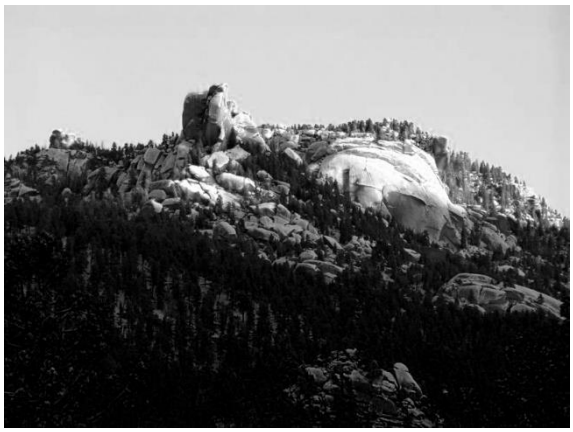
تصویر 3- متعادل کردن هیستوگرام

2.1.3 بر تصویر camera man دو تابع histeq و imadjust را اعمال می‌کنیم و هیستوگرام تصاویر حاصل را مقایسه می‌کنیم:

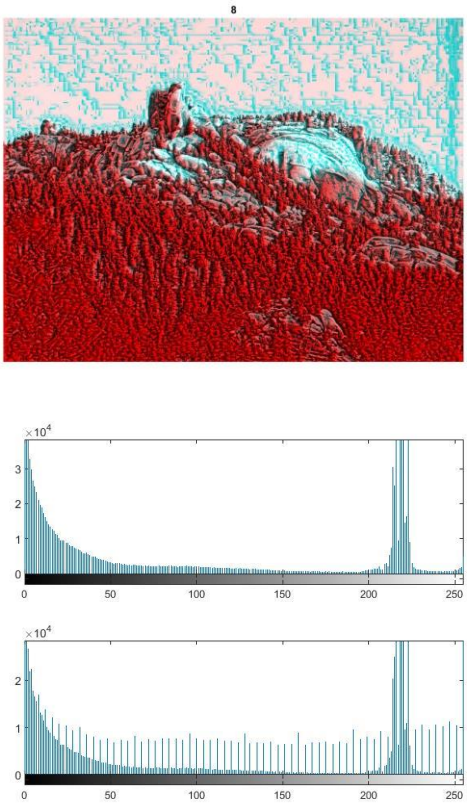


تصویر 4- مقایسه توابع histeq و imadjust

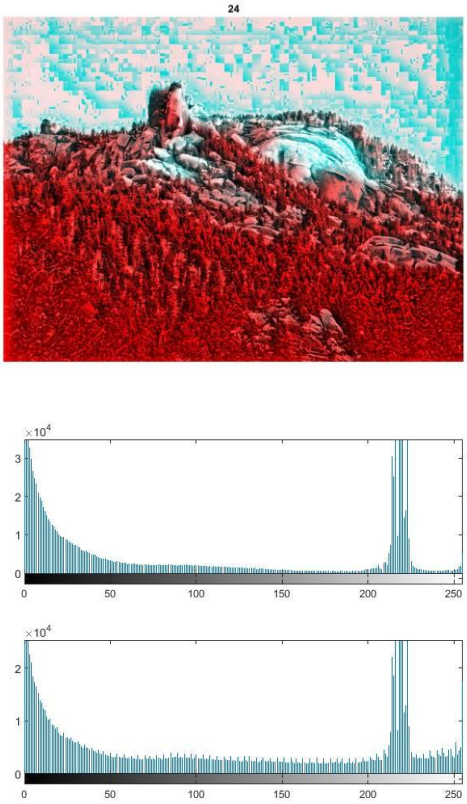
2.2.1 در این بخش چهار تصویر HE1,HE2,HE3 و HE4 را با پنجره هایی به سایز 8، 16 و 24 به صورت محلی متعادل می‌کنیم:



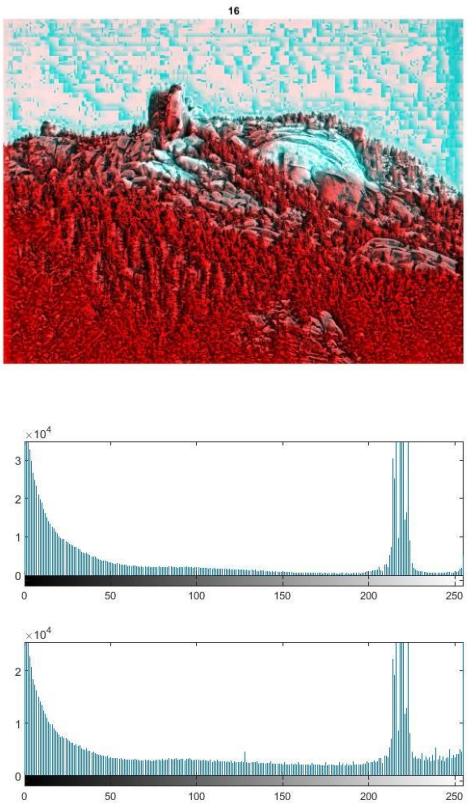
تصویر 5- تصویر HE1



تصویر 6- متعادل سازی تصویر HE1 با پنجره‌ی 8*8

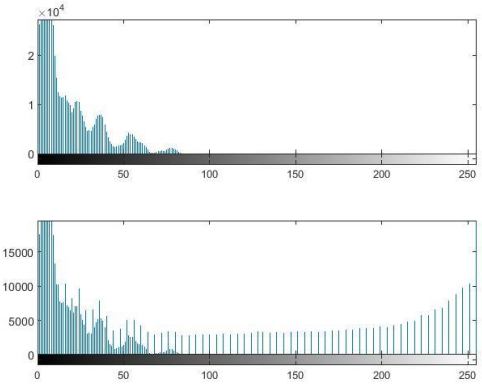


تصویر 8- متعادل سازی تصویر HE1 با پنجره‌ی 24*24

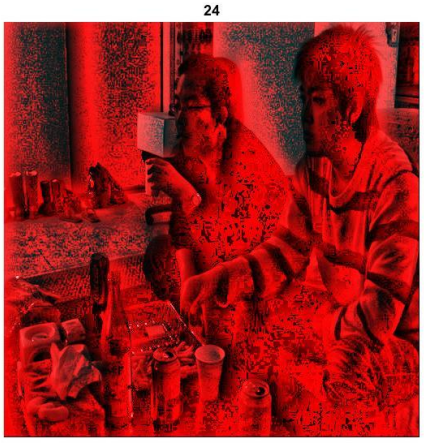


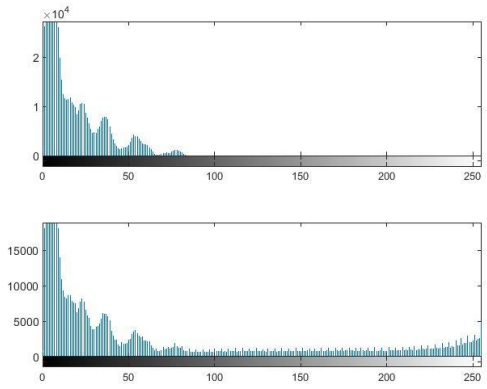
تصویر 7- متعادل سازی تصویر HE1 با پنجره‌ی 16*16

تصویر 9- تصویر HE2

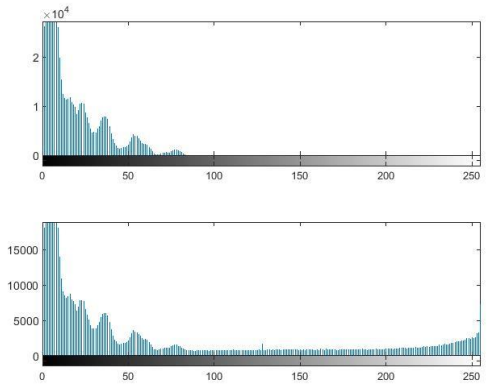


تصویر 10- متعادل سازی تصویر HE2 با پنجره‌ی 8*8



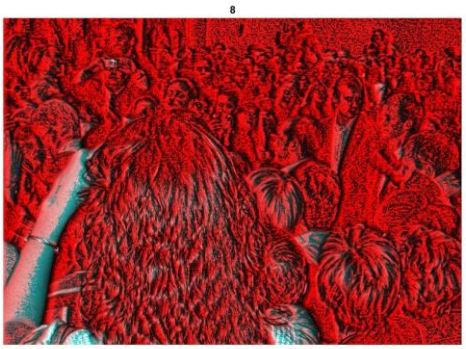


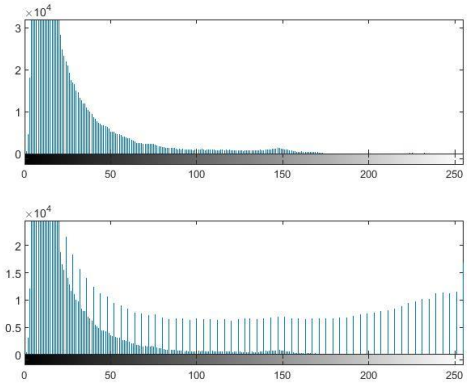
تصویر 12- متعادل سازی تصویر HE2 با پنجره‌ی 24*24



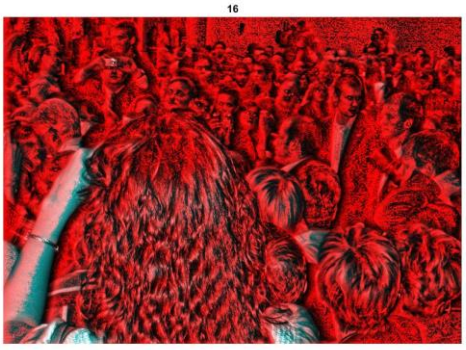
تصویر 11- متعادل سازی تصویر HE2 با پنجره‌ی 16*16

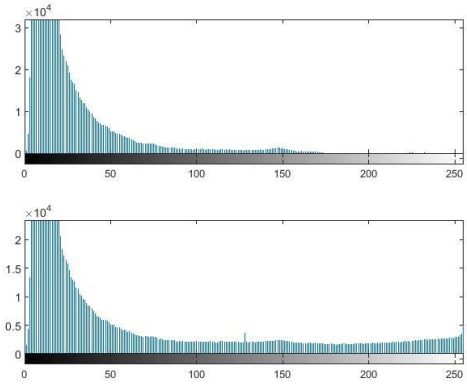
تصویر 13- تصویر HE3



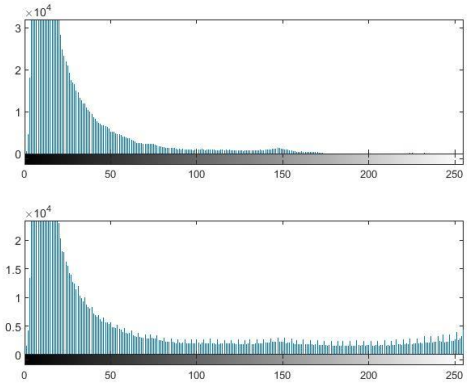


تصویر 14- متعادل سازی تصویر HE3 با پنجره‌ی 8*8



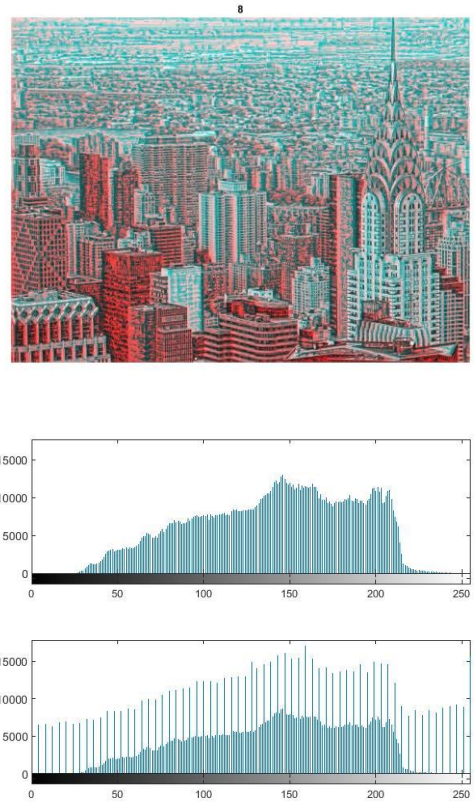


تصویر 15- متعادل سازی تصویر HE3 با پنجره‌ی 16*16

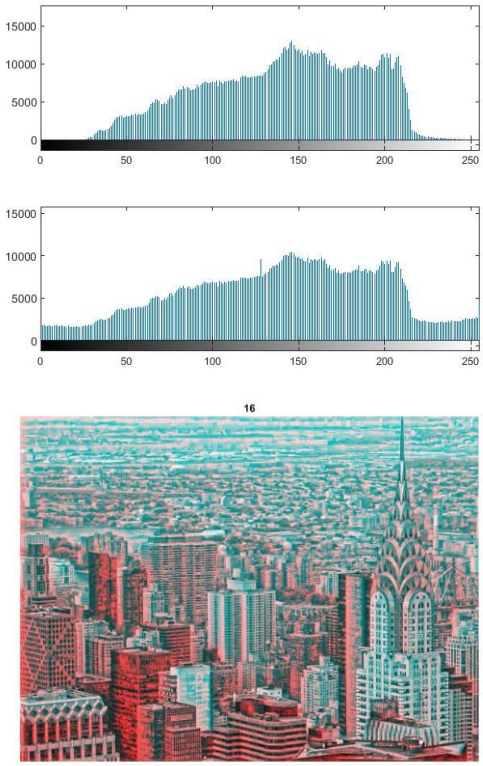


تصویر 16- متعادل سازی تصویر HE3 با پنجره‌ی 24*24

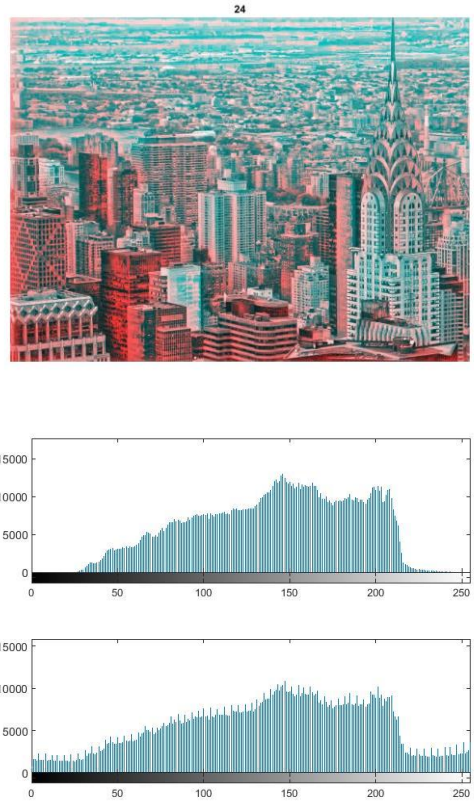
تصویر 17- تصویر HE4



تصویر 18- متعادل سازی تصویر HE4 با پنجره‌ی 8*8



تصویر 19- متعادل سازی تصویر HE4 با پنجره‌ی 16*16



تصویر 20- متعادل سازی تصویر HE4 با پنجره‌ی 24*24

4- نتایج

متعادل کردن هیستوگرام موجب بهبود کنتراست تصویر می‌شود اما ممکن است جزئیات تصویر را در نقاط اشباع (نقاط خیلی روشن) از بین ببرد.

طبق نتایج به دست آمده، در همسان سازی محلّی با افزایش اندازه‌ی پنجره شاهد بهبود کیفیت خروجی و کاهش نویز و بار محاسباتی در تصویر هستیم امّا در تصاویر خروجی با اندازه‌ی پنجرهٔ کوچک جزئیات بیشتری از یک پنجره قابل مشاهده بوده و کنتراست پیکسل های موجود در یک پنجره بیشتر است.

مشهود است که تصاویر خروجی همسان سازی محلی در نمایش جزئیات تصویر در محدوده های با کنتراست پایین موفقتر عمل کرده‌اند، اگرچه همسان سازی سراسری شکل مطلوب تری دارد

همسان سازی محلی پوشا نتایج بهتری را نسبت به غیر پوشا ایجاد می‌کند.

```
function MyHistogram(grayImage)
[M,N]=size(grayImage);
t=1:256;
n=0:255;
count=0;
for z=1:256
   for i=1:M
      for j=1:N

         if grayImage(i,j)==z-1
            count=count+1;
         end
      end
   end
         t(z)=count;
         count=0;
end
figure;
subplot;(1,2,1)
stem(n,t) ;
grid on;
subplot;(1,2,2)
imhist(grayImage);
end
```

```
function finalResult =
histogrameq(orginal)
[rows,columns,~] = size(orginal);
finalResult =
uint8(zeros(rows,columns));
pixelNumber = rows*columns;
frequncy = zeros;(256,1)
pdf = zeros;(256,1)
cdf = zeros;(256,1)
cummlative = zeros;(256,1)
outpic = zeros;(256,1)
for i = 1:1:rows
   for j = 1:1:columns
      val = orginal(i,j);
      frequncy(val+1) =
frequncy(val+1)+1;
      pdf(val+1) =
frequncy(val+1)/pixelNumber;
   end
end
sum =0;
%we want the 256 - 1 that's why
we initailzed the intensityLevel
with 255
%instead of 256
intensityLevel = 255;

for i = 1:1:size(pdf)
   sum =sum +frequncy(i);
   cummlative(i) = sum;
   cdf(i) = cummlative(i)/
pixelNumber;
   outpic(i) = round(cdf(i) *
intensityLevel);
end


for i = 1:1:rows
   for j = 1:1:columns
      finalResult(i,j) =
outpic(orginal(i,j) + 1);
   end
end
subplot(2,2,1),imshow(finalResult
),title('equlized image');
subplot(2,2,2),imhist(finalResult),
title('equlized histogram');
subplot(2,2,3),imshow(orginal),titl
e('original image');
subplot(2,2,4),imhist(orginal),
title('original histogram');
```

```
function localEq(Img,windowSize)
 A= Img;
%WINDOW SIZE
M=windowSize;
N=M;

mid_val=round((M*N)/2);

%FIND THE NUMBER OF ROWS AND COLUMNS TO BE
PADDED WITH ZERO
in=0;
for i=1:M
   for j=1:N
      in=in+1;
      if(in==mid_val)
         PadM=i-1;
         PadN=j-1;
         break;
      end
   end
end
%PADDING THE IMAGE WITH ZERO ON ALL SIDES
B=padarray(A,[PadM,PadN]);

for i= 1:size(B,1)-((PadM*2)+1)

   for j=1:size(B,2)-((PadN*2)+1)
      cdf=zeros;(256,1)
      inc=1;
      for x=1:M
         for y=1:N
%  FIND THE MIDDLE ELEMENT IN THE WINDOW
            if(inc==mid_val)
               ele=B(i+x-1,j+y-1)+1;
            end
               pos=B(i+x-1,j+y-1)+1;
               cdf(pos)=cdf(pos)+1;
              inc=inc+1;
         end
      end
%    COMPUTE THE CDF FOR THE VALUES IN THE
WINDOW
      for l=2:256
         cdf(l)=cdf(l)+cdf(l-1);
      end
         Img(i,j)=round(cdf(ele)/(M*N)*255);
    end
end
figure,imshow(Img),title(windowSize);
figure,subplot(2,1,1);title('Before Local Histogram Equalization');
imhist(A);
subplot(2,1,2);title('After Local Histogram Equalization');
imhist(Img);
end
```

مراجع

- Digital Image Processing / Rafael C. Gonzalez, 4th Edition
- https://stackoverflow.com/questions/26818568/whats-the-difference-between-histeq-and-adapthisteq#:~:text=imadjust%20increases%20the%20contrast%20of,histeq%20performs%20histogram%20equalization.&text=adapthisteq%20performs%20contrast%2Dlimited%20adaptive%20histogram%20equalization
- https://uk.mathworks.com/matlabcentral/answers/267371-what-is-the-difference-between-imadjust-and-histeq
- https://www.imageeprocessing.com/2011/06/local-histogram-equalization.html

.