

تمرین ۲

هادی تمیمی

| اطلاعات گزارش | چکیده |
|------------------|---------------------------------------------------------|
| تاریخ: ۱۴۰۰/۱/۲۸ | در این تمرین به روش های همسان سازی هیستوگرام میپردازیم. |
| واژگان کلیدی: | |
| هیستوگرام | |
| همسان سازی | |
| همسان سازی محلی | |
| تبدیلات نمایی | |
| تبدیلات لگاریتمی | |

۱-مقدمه

هیستوگرام یک تصویر، تابعی گسسته است که اطلاعات تصویر دو بعدی را به برداری یک بعدی تبدیل میکند. در صورت تبدیل تصویر به کمک هیستوگرام دیگر از بردار بدست آمده نمیتوان تصویر را تولید کرد اما این بردار اطلاعات مفیدی مانند توزیع سطوح خاکستری را میدهد.

۲-شرح تکنیکال

۲.۱.۱

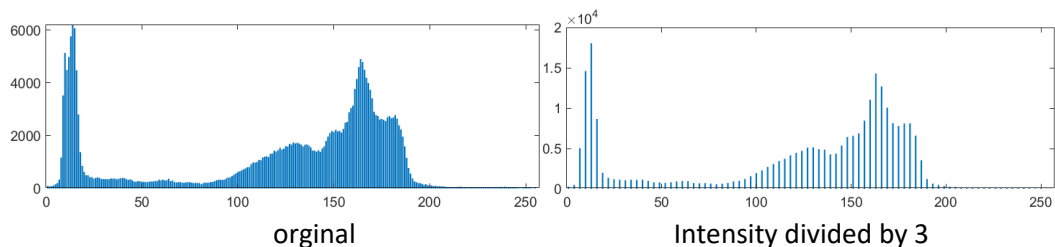
برای محاسبه هیستوگرام با پیمایش روی تمام پیکسل ها و شمارش تعداد پیکسل در هر سطح خاکستری، تعداد پیکسل های ۲۵۶ سطح خاکستری بدست می آوریم.

۲.۱.۱.۱

با توجه به مقدار intensity dividing (۳) بین هایی به طول این مقدار پیکسل ها را تقسیم میکنیم و برای هر بین مقدار floor بین را قرار میدهیم.

۲.۱.۱.۲

هیستوگرام شکل کلی خود را حفظ میکند اما فراوان سطوح خاکستری بین binها صفر شده و مقدار آن ها برای سطح اول bin مربوط به خود اضافه شده.



۲.۱.۱.۳

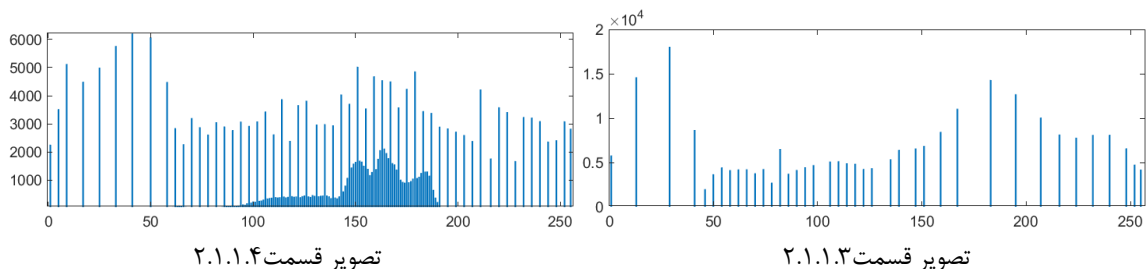
همسان سازی هیستوگرام را روی تصویر خروجی قسمت ۲.۱.۱.۲ قبل انجام میدهیم.

۲.۱.۱.۴

با انتخاب قسمتی از تصویر خروجی قسمت ۲.۱.۱.۲ و انجام همسان سازی هیستوگرام رو آن قسمت local histogram equalization را انجام میدهیم.

۲.۱.۱.۵

هیستوگرام تصویر پس از اعمال local histogram equalization در ناحیه انتخابی یکنواخت تر شده و واریانس آن افزایش پیدا کرده.



۲.۱.۱.۶

با پیمایش روی تمام پیکسل ها و اعمال فرمول های log transform , incverse log transform, power_law transform روی آنها تصویر نهایی برای فرایند های خواسته شده به دست می آوریم.

۲.۱.۲

همسان سازی هیستوگرام روشی است که در آن واریانس هیستوگرام یک تصویر افزایش میابد. این روش باعث میشود هیستوگرام یکنواخت تر بشود. تبدیل را انجام دهد، خواهیم داشت T : اگر تابع $s = T(r), 0 \leq r \leq 1$ سطح خاکستری نرمالیزه شده r که در آن سطح خاکستری نرمالیزه s قبل از تبدیل و شده بعد از تبدیل پیکسل ها میباشند. تابع، باید دو شرط را ارضا کند:

- در بازه $0 \leq r \leq 1$ صعودی باشد
- برای $0 \leq r \leq 1$ ، $0 \leq T(r) \leq 1$ باشد

برای به دست آوردن T از cumulative distribution استفاده میکنیم. با به دست آوردن CDF یک look up table برای نگاشت r به s میسازیم و هر مقدار سطح خاکستری با رابطه زیر به دست می آوریم.

$$s_i = \lfloor (L - 1) \times CDF(i) \rfloor$$

۲.۱.۳

Imadjust: با استفاده از این تابع می توان دامنه تغییرات روشنایی یک تصویر را تغییر داد. با نگاشت مقادیر شدت ورودی به مقادیر جدید، تضاد تصویر را افزایش می دهد

Histeq: همسان سازی هیستوگرام را با کشش بخش هایی از هیستوگرام تصویر انجام می دهد. کنتراست تصاویر را با تبدیل مقادیر در یک تصویر با تجمع بالا، افزایش می دهد. به طور پیش فرض، توزیع احتمال را به یک توزیع یکنواخت تبدیل می کند و معمولاً کیفیت روشنایی تصویر را به میزان زیادی بهبود می بخشد.

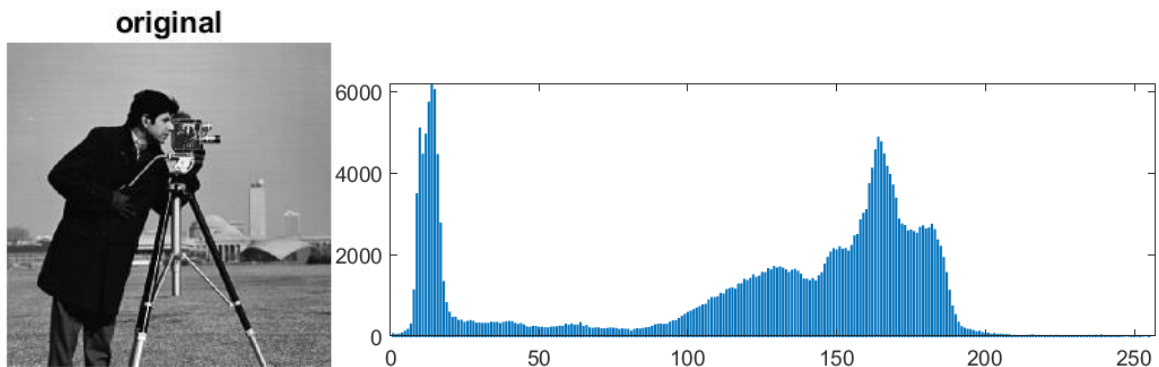
Imadjust به صورت خطی هیستوگرام را محاسبه می کند در حالی که **histeq** اینطور نیست. تصاویر با **imadjust** طبیعی تر به نظر می رسند.

۲.۲

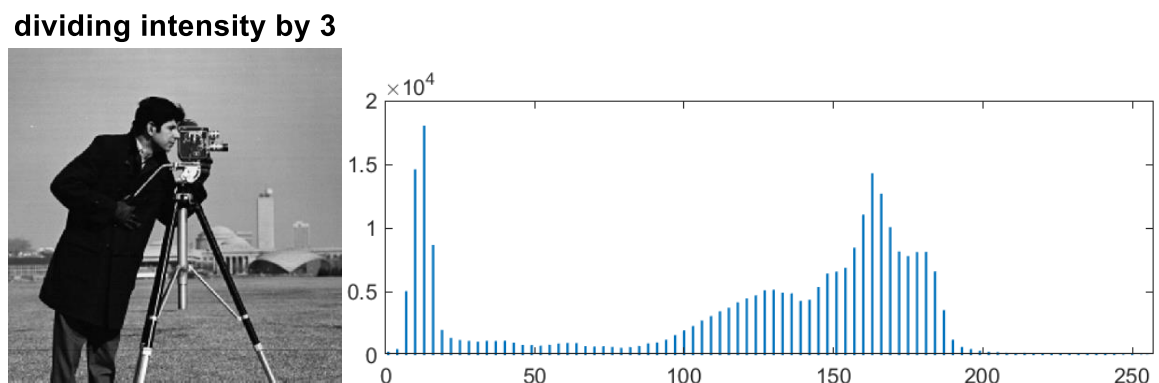
در روش قبلی اتفاقی که ممکن است رخ دهد این است که برخی نواحی ممکن است روشنایی بالایی داشته باشند که بر روی کل تصویر اثر بگذارند. یعنی تصویری که به طور کلی تیره است ولی یک ناحیه خیلی روشن دارد ممکن است در کل روشن تر شود که نتیجه مطلوب نیست. برای حل این مشکل از همسان سازی محلی هیستوگرام استفاده میشود. در این روش پنجره ای با ابعاد مشخص شده در نظر گرفته میشود و تصویر به قطعاتی با ابعاد این پنجره تقسیم میشود و همسان سازی هیستوگرام بر روی این قطعات صورت میگیرد.

۲-شرح نتایج

۲.۱.۱

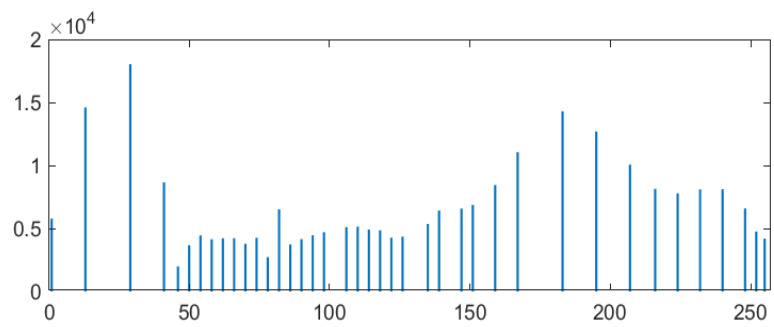


۲.۱.۱.۱



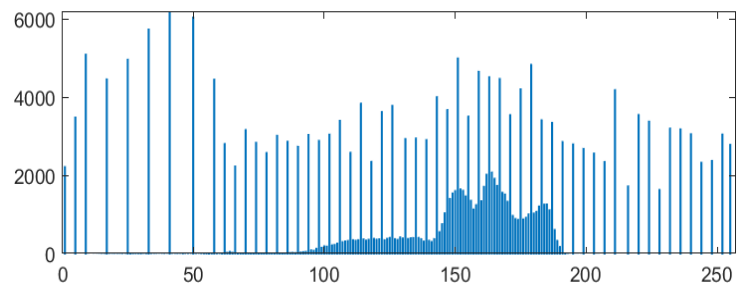
۲.۱.۱.۳

histogram equalization on D



۲.۱.۱.۴

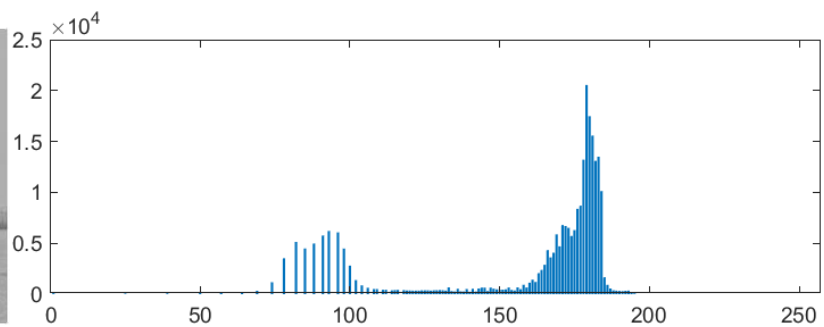
local histogram equalization on D



۲.۱.۱.۶

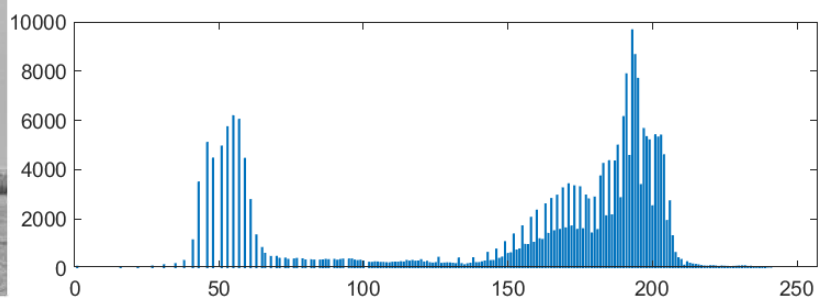
:Log transform

log transform



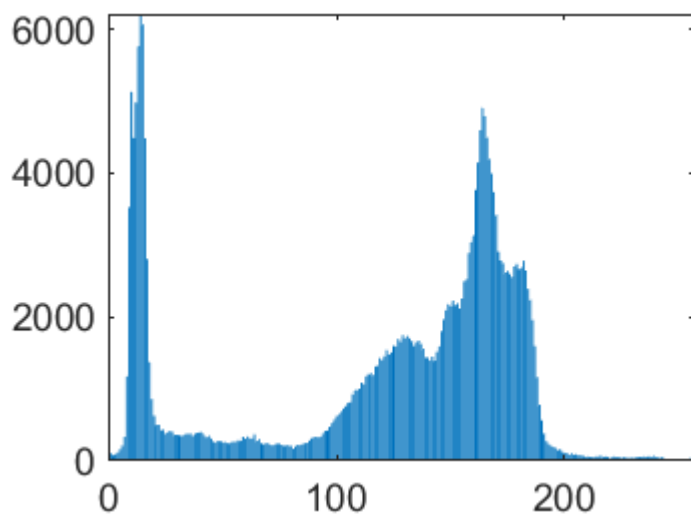
Power-law transform

power-law transform



۲.۱.۲

قبل همگانسازی:

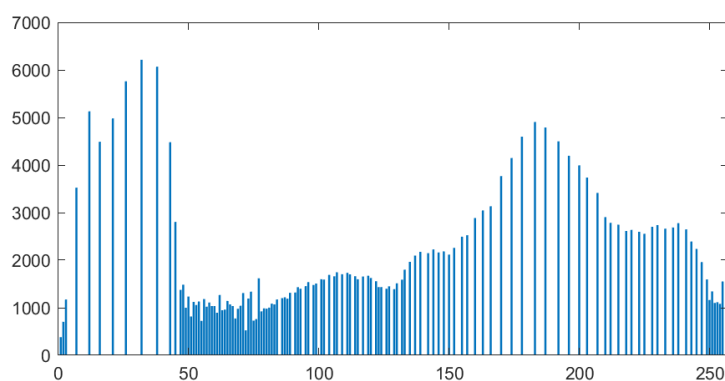


original



بعد از همگانسازی:

histeq



HE1 window size 130



HE1 window size 300



HE2 window size 130



HE2 window size 300



HE3 window size 130



HE3 window size 400



HE4 window size 130



HE4 window size 400



کد ها:

۲.۱.۱

لود کردن تصویر و فراخوانی توابع و نمایش خروجی:

```
image = imread('Images\2\Camera Man.bmp');  
hist =Img_Hist(image);  
figure
```

```

subplot(2,1,1);
imshow(image);
title('original');
subplot(2,1,2);
bar(hist)

```

تابع هیستوگرام:

```

function Hist = Img_Hist(img)
    [R,C]=size(img);
    Hist=zeros(256,1);
    for r = 1:R
        for c=1:C
            Hist(img(r,c)+1,1)=Hist(img(r,c)+1,1)+1;
        end
    end
end

```

۲.۱.۱.۱

لود کردن تصویر و فراخوانی توابع و نمایش خروجی:

```

image = imread('Images\2\Camera Man.bmp');
D = devide_func(image,3);
histD =Img_Hist(D);
figure
subplot(2,1,1);
imshow(D);
title('dividing intensity by 3');
subplot(2,1,2);
bar(histD);

```

تابع intensity dividing:

```

function out = devide_func(image, n)
    [M,N] = size(image);
    out=zeros(M,N, 'uint8');
    for i=1:M
        for j=1:N
            m=double(image(i,j));
            m=nearest_floor(m,n);
            out(i,j)=m;
        end
    end
end

```

تابع ایجاد Bin:

```

function out = nearest_floor(m,n)
    for r = 1:n

```

```

        if (mod(m,n)==0)
            out=m;
            break
        end
        m=m-1;
    end

end

```

۲.۱.۱.۳

فراخوانی توابع و نمایش خروجی:

```

H = histeq(D);
histH =Img_Hist(H);
figure
subplot(2,1,1);
imshow(H);
title('histogram equalization on D');
subplot(2,1,2);
bar(histH);

```

۲.۱.۱.۴

فراخوانی توابع و تعیین ناحیه و نمایش خروجی:

```

local = image(70:512, 42:450);
L = image;
localhisteq = histeq(local);
L(70:512, 42:450) = localhisteq;
histL =Img_Hist(L);
figure
subplot(2,1,1);
imshow(L);
title('local histogram equalization on D');
subplot(2,1,2);
bar(histL);

```

تابع histogram equalization:

```

function out = hist_eq(img)
    [r,c] = size(img);
    n = r * c;

    out = zeros(r,c,'uint8');

    cdf = zeros(256,1);
    count = zeros(256,1);
    out = zeros(256,1);
    for i=1:r
        for j=1:c

```

```

        value = img(i,j);
        count(value+1) = count(value+1)+1;
    end
end
pdf = count/n;
sum = 0; L =255;
for i=1:256
    sum = sum +pdf(i);
    cdf(i) = sum;
    out(i) = round(cdf(i)*L);
end
for i=1:r
    for j=1:c
        out(i,j) = out(img(i,j)+1);
    end
end
end
end

```

۲.۱.۱.۶

فراخوانی توابع و تعیین پارامترهای پارامترهای log و نمایش خروجی:

```

logImg = log_transform(image,35);
histLog =Img_Hist(logImg);
figure
subplot(2,1,1);
imshow(logImg);
title('log transform');
subplot(2,1,2);
bar(histLog);

```

تابع log transform:

```

function out = log_transform(image,c)
    [M,N] = size(image);
    out=zeros(M,N,'uint8');
    for i=1:M
        for j=1:N
            r=double(image(i,j));
            out(i,j)=c*log(1 + (r)); % Log Transform
        end
    end
end
end

```

فراخوانی توابع و تعیین پارامترهای پارامترهای power_law transform و نمایش خروجی:

```

powerImg = power_transform(image,15,0.5);
histPower =Img_Hist(powerImg);
figure

```

```

subplot(2,1,1);
imshow(powerImg);
title('power-law transform');
subplot(2,1,2);
bar(histPower);

```

تابع power-law transform:

```

function out = power_transform(image,c,omega)
    [M,N] = size(image);
    out=zeros(M,N,'uint8');
    for i=1:M
        for j=1:N
            r=double(image(i,j));
            out(i,j)=c*(r^omega); % Log Transform
        end
    end
end

```

۲.۱.۲

لود کردن تصویر و فراخوانی توابع و نمایش خروجی:

```

image = imread('Images\2\Camera Man.bmp');
new_image = hist_eq(image);
hist_orig = Img_Hist(image);
hist_eqimg = Img_Hist(new_image);
subplot(2,2,1);
imshow(image);
title('original');
subplot(2,2,2);
bar(hist_orig);
subplot(2,2,3);
imshow(new_image);
title('histeq');
subplot(2,2,4);
bar(hist_eqimg);

```

تابع histogram equalization:

```

function new_image = hist_eq(img)
    [r,c,z] = size(img);
    n = r * c;
    new_image = zeros(r,c,z,'uint8');
    cdf = zeros(256,1,'double');
    pdf = zeros(256,1,'double');
    count = zeros(256,1,'double');
    out = zeros(256,1,'double');
    for i=1:r
        for j=1:c

```

```

        value = img(i,j,:);
        count(value+1) = count(value+1)+1;
    end
end
pdf = count/n;
sum = 0; L =255;
for i=1:256
    sum = sum +pdf(i);
    cdf(i) = sum;
    out(i) = round(cdf(i)*L);
end
for i=1:r
    for j=1:c
        new_image(i,j,:) = out(img(i,j,:)+1);
    end
end
end
end

```

تابع histogram:

```

function Hist = Img_Hist(img)
    [R,C]=size(img);
    Hist=zeros(256,1);
    for r = 1:R
        for c=1:C
            Hist(img(r,c)+1,1)=Hist(img(r,c)+1,1)+1;
        end
    end
end
end

```

۲.۲.۱

لود کردن تصویر و فراخوانی توابع و نمایش خروجی:

```

%he = imread('Images/2/HE1.jpg');
%he = imread('Images/2/HE2.jpg');
he = imread('Images/2/HE3.jpg');
%he = imread('Images/2/HE4.jpg');
%imshow(he3);

im = rgb2gray(he);
x = local_hist(im,320);
imshow(x);

```

تابع local histogram equalization:

```

function L=local_hist(image,local_size)
    [r,c] = size(image);
    radias= local_size/2;
    if((radias*2)>min(r,c))
        radias = floor((min(r,c))/2)-1;
    end

```

```

end
s1=floor(((r/2)-radias));
s2=floor(((r/2)+radias));
s3=floor(((c/2)-radias));
s4=floor(((c/2)+radias));
local = image(s1:s2,s3:s4);
L = image;
localhisteq = hist_eq(local);
L(s1:s2,s3:s4) = localhisteq;
end

```

تابع histogram equalization :

```

function new_image = hist_eq(img)
[r,c,z] = size(img);
n = r * c;

new_image = zeros(r,c,z,'uint8');

cdf = zeros(256,1,'double');
pdf = zeros(256,1,'double');
count = zeros(256,1,'double');
out = zeros(256,1,'double');
for i=1:r
    for j=1:c
        value = img(i,j,:);
        count(value+1) = count(value+1)+1;
    end
end
pdf = count/n;
sum = 0; L =255;
for i=1:256
    sum = sum +pdf(i);
    cdf(i) = sum;
    out(i) = round(cdf(i)*L);
end
for i=1:r
    for j=1:c
        new_image(i,j,:) = out(img(i,j,:)+1);
    end
end
end

```