

# Contrast adjustment

فهمیم جعفری

اطلاعات گزارش	چکیده
تاریخ:	
واژگان کلیدی: Histogram Global histogram eq Local histogram eq Pdf :probability density function Cdf:cumulative density function	واژه مکانی به خود تصویر اطلاق میشود و روش های پردازش تصویر در این حوزه، مبتنی بر دستکاری مستقیم پیکسل ها در تصویر است. این موضوع عکس پردازش تصویر در حوزه تبدیل است، دو دسته اصلی از پردازش مکانی، تبدیلات شدت و فیلتر کردن مکانی است. تبدیلات شدت، بر روی هر یک از پیکسل های تصویر عمل میکنند تا دستکاری کنتراست و استانه گیری تصویر انجام گیرد. هیستوگرام یک تصویر دیجیتال با سطوح شدت در بازه $[0, L-1]$ یک تابع گسسته $h(r_k) = nk$ است که $r_k$ برابر با $k$ امین مقدار شدت و $nk$ برابر با تعداد پیکسلها در تصویری با شدت $r_k$ است. هیستوگرام ها مبنایی برای بسیاری از تکنیک های پردازش حوزه مکانی است. دستکاری هیستوگرام میتواند برای ارتقای تصویر به کار رود، که در این تمرین از pdf و cdf برای این دستکاری ها و بهبود تصویر استفاده شده است

## 1-مقدمه

## 2-شرح تکنیکال

### Histogram •

برای رسم هیستوگرام یک تصویر باید تعداد پیکسل های هر شدت از بازه ی (0,255) موجود در تصویر را اندازه گرفته و همراه با شدت آن ذخیره کرده و سپس این شدت ها را به ترتیب بزرگی شدت از 0 تا 255 بصورت نمودار هیستوگرام با کمک کتابخانه matplotlib مانند شکل مقابل نمایش میدهیم. برای اندازه گیری تعداد هر شدت یک حلقه بر روی تمام پیکسل های عکس میزنیم و به تعداد پیکسل های شدت مربوطه یکی اضافه میکنیم

$$h(r_k) = h(r_k) + 1$$

در زمینه پردازش تصویر، هیستوگرام یک تصویر به طور معمول به یک هیستوگرام از مقادیر شدت پیکسل اشاره دارد. این هیستوگرام گرافی است که تعداد پیکسل های یک تصویر را در هر مقدار شدت متفاوت در آن تصویر نشان می دهد. برای تصویر مقیاس خاکستری 8 بیتی 256 شدت ممکن مختلف وجود دارد، بنابراین هیستوگرام به صورت گرافیکی 256 عدد را نشان می دهد که توزیع پیکسل را در بین آن مقادیر خاکستری نشان می دهد. همچنین می توان از هیستوگرام از تصاویر رنگی استفاده کرد که در این بخش قصد داریم به طور عملی آن را مشاهده کنیم و از روی آن تابع تبدیلی برای تغییر شدت پیکسلهای عکس بدست بیاوریم که منجر به بهبود کیفیت تصویر شود.

\* پست الکترونیک نویسنده مسئول: @... ..

1. استادیار، دانشکده مهندسی عمران، دانشگاه سمنان

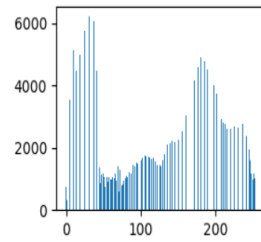
2. استادیار، دانشکده مهندسی عمران، دانشگاه سمنان

3. استادیار، دانشکده مهندسی عمران، دانشگاه سمنان

$$s_0 = T(r_0) = (8-1) \sum_{j=0}^0 p_r(r_j) = 7p_r(r_0) = 7(0.19) = 1.33$$

$$s_1 = T(r_1) = (8-1) \sum_{j=0}^1 p_r(r_j) = 7p_r(r_0) + 7p_r(r_1) = 7(0.19) + 7(0.25) = 1.33 + 1.75 = 3.08$$

$$s_2 = T(r_2) = (8-1) \sum_{j=0}^2 p_r(r_j) = 7(0.19) + 7(0.25) + 7(0.21) = 1.33 + 1.75 + 1.47 = 4.55$$



## ImAdjust&HistEq

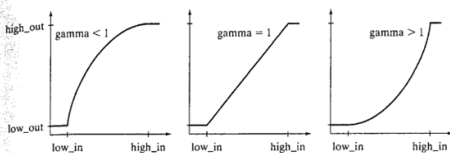
Imadjust:

با کمک این دستور می توان بازه شدت تصویر را تغییر داد ، مثلا فرض کنید مقادیر شدت یک تصویر خاکستری بین 100 تا 250 است ، یعنی حداقل مقدار شدت موجود در تصویر 100 و حداکثر آن 250 است. حالا اگر بخواهیم مقدار شدت تمام پیکسل ها را با نسبت یکسان به گونه ای تغییر دهیم تا حداقل آن صفر و حداکثر آن 255 شود ، این دستور به راحتی کار را برای ما انجام می دهد

`g=imadjust(f,[low_in  
high_in],[low_out  
high_out],gamma)`

مقادیر low,high بین صفر و یک میباشد

گاما نسبت تغییرات را مشخص می کند که اگر یک باشد به صورت خطی تغییر می کند و اگر مقداری غیر از یک باشد مطابق شکل زیر است :



histEq:

این دستور شدت ها را در هستوگرام یکسان میکند همانطور که در بخش قبل توضیح داده شده است

## Local histogram equalization

متعادل کردن هستوگرام که قبلا محاسبه کردیم هستوگرام کل تصویر را در نظر می گرفت و یکسان سازی انجام میداد ولی در متعادل کردن هستوگرام محلی تصویر را به کاشی هایی تقسیم کرده و بر روی کاشی عمل تعادل سازی

## Histogram Equalization

متعادل کردن هستوگرام عمل گسترده هستوگرام تصویر اصلی برای بهبود کنتراست و جزئیات سطوح خاکستری تصاویر است

$$s = T(r) \quad 0 \leq r \leq 1$$

$s, r$  مقادیر نرمالیز شده شدت پیکسل ها هستند  
 شروط تبدیل (1):  $T(r)$  در بازه  $0 \leq r \leq 1$  صعودی باشد. شیب آن باید همیشه غیر منفی باشد یعنی میتواند صفر باشد اما در این صورت اطلاعات از بین میرود (2) برای  $0 \leq r \leq 1$  باید  $0 \leq T(r) \leq 1$  باشد

فرض کنیم که میخواهیم هستوگرام ما به بهترین حالت درآمده است یعنی

$$\begin{aligned} p_s(s) &= \frac{1}{L-1} \\ \rho_{s(s)} &= \rho_r(r) \left| \frac{dr}{ds} \right| \Rightarrow \\ s = T(r) &= (L-1) \int_0^r p_r(w) dw \\ \Rightarrow \frac{ds}{dr} &= \frac{dT(r)}{dr} \Rightarrow \frac{ds}{dr} = \frac{dT(r)}{dr} \\ &= (L-1) \frac{d}{dr} \left[ \int_0^r \rho_r(w) dw \right] \\ &= (L-1) \rho_r(r) \\ \rho_s(s) &= \rho_r(r) \left| \frac{1}{(L-1)\rho_r(r)} \right| \\ &= \frac{1}{L-1} \end{aligned}$$

پس اگر از cdf هستوگرام نرمال شده به عنوان تابع تبدیل استفاده کنیم هستوگرام جدید تقریبا تعدیل شده میباشد علت تقریبی به خاطر گسسته بودن هستوگرام میباشد البته چون هستوگرام نرمال شده نیز میباشد باید در (-L) (1) نیز ضرب کنیم یک نمونه از بدست آوردن تابع تبدیل

1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10

تصویر ورودی نهایی که آخرین پنجره زده شده

1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10

برای حاشیه ها از نزدیک ترین پنجره متعادل شده قرمز همسایه استفاده شده است.

### 3-شرح نتایج

#### Histogram •

تصویری که بر روی آن هیستوگرام گرفته شده است همراه با هیستوگرام را نشان می‌دهد از آنجا که بیشتر شدت پیکسل ها نزدیک 118 میباشد پس تصویر نه روشن میباشد نه تاریک. این عدد 118 با میانگیری از کل پیکسل ها محاسبه شده است

هیستوگرام را انجام می‌دهیم سپس مرکز کاشی را در عکس جدید قرار داده و کاشی را به پیکسل بعد عمودی یا افقی جا به جا کرده و دوباره عمل تعادل سازی بر روی کاشی را انجام می‌دهیم و این کار را برای کل تصویر انجام می‌دهیم این تعادل سازی هیستوگرام محلی با همپوشانی میباشد روش دیگر تعادل سازی هیستوگرام محلی که برای کاهش محاسبات است به جای اینکه کاشی را پیکسل به پیکسل جابه‌جا کنیم چند پیکسل چند پیکسل جابه‌جا کرده و در خروجی می‌گذاریم ولی مشکل این روش این است که تصویر خروجی به صورتی بلوکی بلوکی در می‌آید که برای رفع این مشکل میتوان از پنجره گوسی استفاده کرد تا به مرکز پنجره توجه بیشتر شده و لبه های پنجره اسموتر تر شود یا در آخر یک فیلتر گوسی

بزنیم

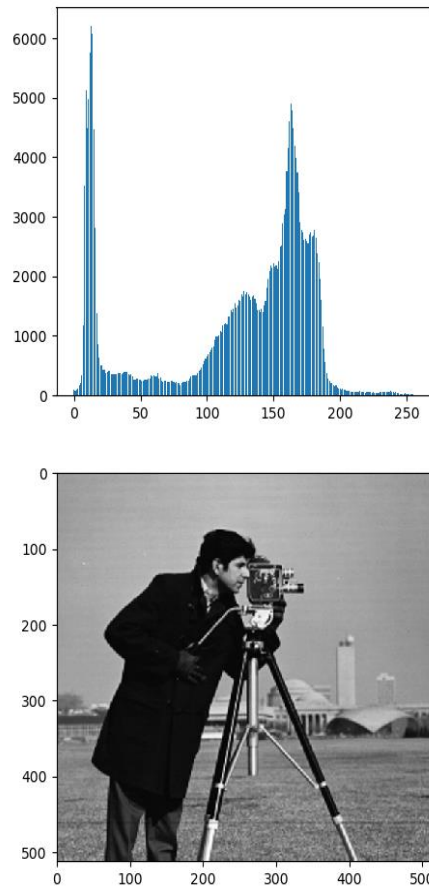
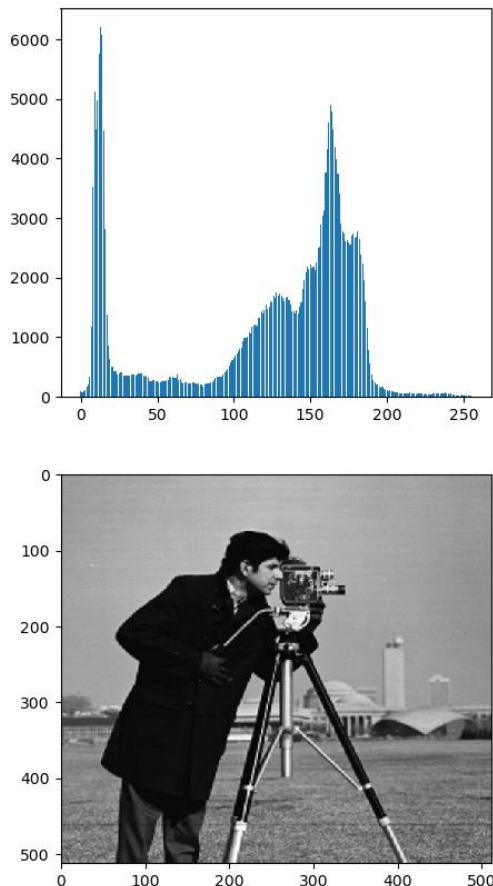
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10

بر روی پنجره ای histeq زده شده و پنجره قرمز از پنجره ای بر روی خروجی اعمال شده و به اندازه پنجره قرمز در راستای افقی جابه جا شده

1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10

پنجره دوم

به رنج هایی که دارای تجمع کمتر شدت مثلا 55 انتقال یافته است و هیستوگرام هموارتر شده است و همینطور کیفیت تصویر بهبود یافته است برای اطمینان درستی کارم متعادل سازی هیستوگرام cv2 را نیز در زیر نمایش داده ام .



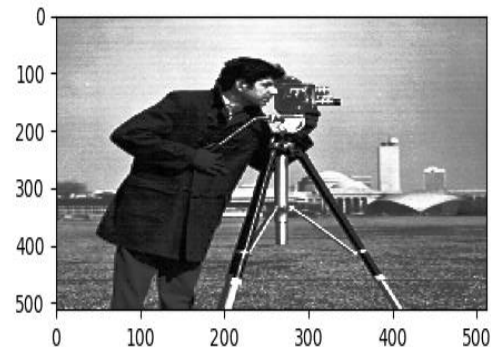
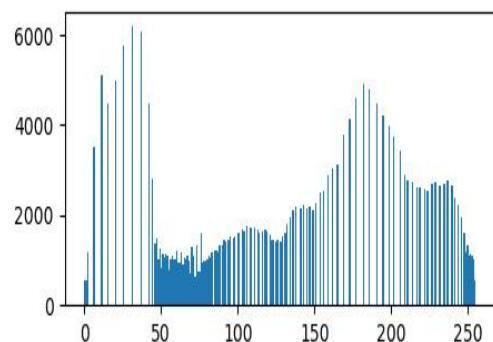
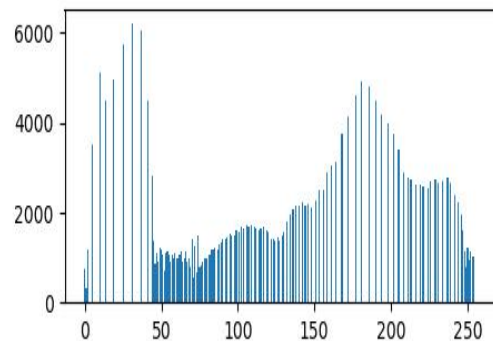
### • Histogram equalization

شکل زیر مربوط به تصویر Man.bmp میباشد که ردیف اول شکل هیستوگرام و تصویر اصلی میباشد و ردیف دوم هیستوگرام و تصویر متعادل شده ی تصویر اصلی میباشد که با تابع histeq خودم پیاده کرده بودم متعادل شده و ردیف سوم همانند ردیف 2 میباشد با این تفاوت که با استفاده از تابع histeq کتابخانه cv2 انجام شده است . میانگین شدت پیکسل ها برابر با 118 میباشد پس میتوان حدس زد که تصویر نه روشن و نه تاریک است ولی شدت ها در بعضی از رنج ها مانند قسمت تاریک (نزدیک 0) و نزدیک 160 جمع شده است و زیاد است پس واریانس میتواند بهبود یابد و کیفیت تصویر بهتر شود و همانطور که در شکل دیده میشود پس از اعمال histeq شدت های قسمت های تاریک و نزدیک 160 که تجمع شدت ها در آنجاست

چون تصویر اصلی (Man.bmp) کمترین شدت و بالاترین شدت آن به ترتیب 0 و 255 میباشد نمیشود از دستور `imadjust` برای بهبود تصویر استفاده کرد ولی میشود از آن بعنوان تابعی که به رنجی از شدت ها که مورد علاقه هستیم را نشان و هیستوگرام آن را به 0 تا 255 بسط دهد مثال تصویر زیر که به بازه ی 0 تا 1. نسبت به کل بازه ی 0 تا 1 علاقه داریم را بسط دادیم و که شی انسان و دوربین در آن مشاهده میشود



کاربرد دیگری که `imadjust` دارد مثل `histeq` هیستوگرام را دستکاری کرده و میتوان هیستوگرام را یکسان کرد البته باید با حدس زدن گامای مناسب مثلا اگر تصویر تاریک باشد با انتخاب گامای کوچکتر از یک این شدت ها را در کل بازه پخش کرده و اطلاعات پنهان در قسمت تاریک برای چشم انسان ظاهر میشود و تصویر روشن تر میشود مانند مثال زیر که گاما 0.2 انتخاب شده است





تصویر بالا با local hist eq تصویر اصلی با سایز پنجره  $256 \times 256$  و هم پوشانی بالا میباشد اگر این تصویر را با تصویر حالت global مقایسه کنیم مشاهده میشود که جزئیات محلی در عکس بالا دیده میشود که در حالت global دیده نمیشود مانند برجی که کاملاً به رنگ سیاه در حالت global دیده میشود این برج در عکس local hist eq دارای جزئیات بیشتر مانند تمایز بین طبقات بین برج دیده میشود .



تصویر بالا با local hist eq تصویر اصلی با سایز پنجره  $128 \times 128$  و غیر هم پوشان میباشد اولین چیزی که در تصویر مشخص است بلوکی بودن تصویر است و علت آن چیزی نیست جز غیر هم پوشان بودن پنجره ها چرا که در هر در پنجره تابع تبدیل متفاوتی برای متعادل سازی هیستوگرام انجام شده است و دومین نکته ای که در مورد تصویر وجود دارد این میباشد که در مقایسه با تصویر قبلی که سایز پنجره بزرگتر بود جزئیات بیشتری ظاهر شده است مثال همان برج سیاه قبلی، علت آن اینست که در هر پنجره شدت ها بین 0 تا 255 پخش و شده است هر چه این پنجره کوچکتر شود جزئیات بیشتر دیده شده ولی از یک سایزی

و تصویر زیر گامای 2 انتخاب شده است که تصویر تاریک شده است



برای مقایسه بین imadjust و histeq از histeq مرحله قبل استفاده شده است.

### • Local histogram equalization

بر روی تصاویر HE1, HE2, HE3, HE4 متعادل سازی هیستوگرام محلی با پنجره های با سایزهای مختلف و مقدار overlap ای که هر یک با پنجره مجاور دارد انجام شده است که هر یک در زیر تشریح کرده ام



این تصویر global histeq تصویر اصلی که روشن بوده است میباشد و بهبود یافته است

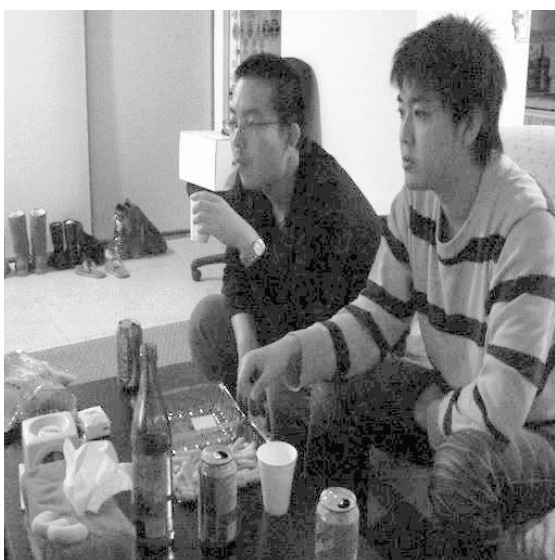
اطلاعات لوکال بیشتری ظاهر میشود و اگر هم پوشانی کمتر شود تصویر به حالت بلوکی در میاد مانند شکل پایین



کوچکتر در تصویر نویز (اطلاعات تصویر از بین میرود) ایجاد شده و عملاً نه دید کلی از تصویر خوب میباشد و نه جزییاتی دیده میشود



تصویر بالا global hist eq یک تصویر تاریک میباشد که الان روشن و بهبود یافته است



تصویر global histeq تصویر اصلی میباشد همانطور که در تصویر مشخص است نوشته ها و مارک های روی بطری ها ب علت اینکه سهم کوچکی در هیستوگرام داشتند و در تابع تبدیل تاثیر کم داشته است دیده نمیشود برای واضح کردن این نوشته ها باید از local hist eq استفاده کرد تا در هیستوگرام و تابع تبدیل مشارکت بیشتر داشته و در histeq تقویت شده و بهبود یابد و دیده شوند که نتیجه local hist eq با سایز پنجره  $256 \times 256$  و همپوشانی بالا میباشد



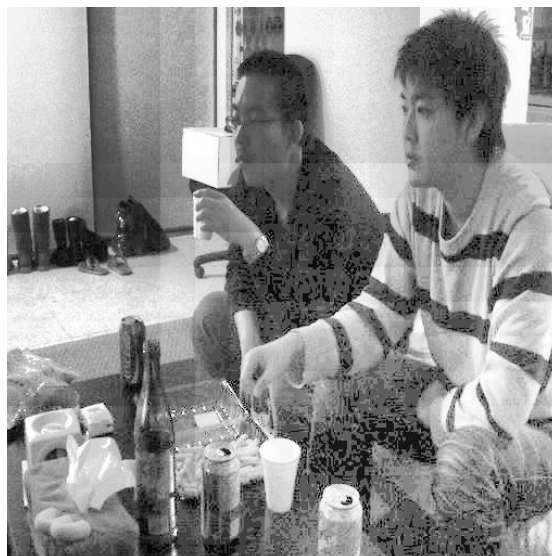
تصویر local histeq با سایز پنجره  $256 \times 256$  و همپوشانی بالا چون تصویر اصلی بیشتر تاریک بوده است قسمت هایی از عکس که روشن تر بوده اند در هیستوگرام نقش کمتری داشته اند و در نتیجه در تابع تبدیل global قسمت های تاریک تاثیر بیشتر داشته و تاثیر قسمت روشن در تابع تبدیل به چشم نیامده است به همین سبب در حالت global قسمت های روشن بیش از حد روشن شده است به طوری که در آن قسمت کنتراست پایین آمده است مثال النگوی خانمی که در حال عکاسی میباشد ولی در حالت local چون در هر پنجره هیستوگرام گرفته میشود آن قسمت ها در هیستوگرام تاثیر چشم گیر داشته و تابع تبدیل مطلوبی بدست آمده و همانطور که در تصویر نیز دیده میشود النگوی خانم در حال عکاسی بهتر و با کنتراست بهتر دیده میشود هر چه این سایز کوچکتر شود



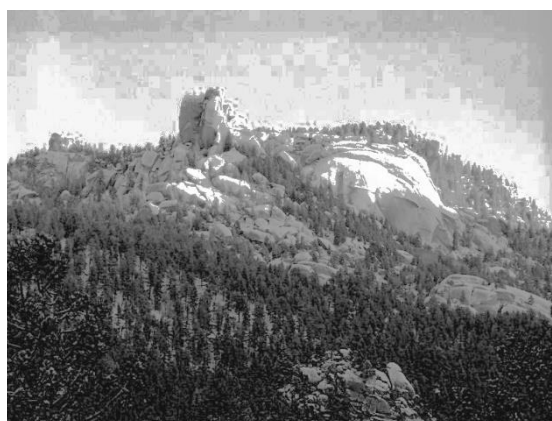
است با عمل متعادل سازی هیستوگرام به شدت های روشن تر انتقال یافته و همینطور شدت های بالا مربوط به رنگ صخره به همان نزدیکی که شدت های درختان روی صخره انتقال یافته بود در آمده است چون در قسمت شدت های وسط در هیستوگرام کم بودند لذا تابع تبدیل در این قسمت به یک خط تقریباً صاف افقی در آمده و این دو شدت (شدت درختان روی صخره و شدت صخره) به هم نزدیک شده و این تفاوت اندک بین آن ها با چشم انسان قابل دیدن نیست و بعضی از جزییات تصویر از دست رفته است برای بهبود این مشکل میتوان از  $local\ hist\ eq$  استفاده کرد تا فاصله بین آنها زیاد شده و تفاوت محسوس تر شود مثال تصویر پایین با سایز پنجره  $256 \times 256$  و همپوشانی بالا که روی تصویر اصلی اعمال شده است



تصاویر اصلی که بر روی آنها متعادل سازی بر روی آنها انجام شده است



اگر سایز هم پوشان را کم شوی شبیه مثال های قبل تصویر بلوکی در می آید



تصویر بالا  $global\ histeq$  تصویر اصلی میباشد مشکل اولی که در تصویر دیده میشود تصویر به حالت  $quantize$  ای دیده میشود در قسمت بالای تصویر واضح است علت آن این است که شدت ها در تصویر اصلی در دو نقطه جمع شده است قسمت تاریک و قسمت روشن و هنگامی هیستوگرام متعادل سازی میشود شدت های نزدیک به هم از هم فاصله گرفته کنتراست زیاد میشود که در تصویر هم پیکسل های مجاور که شدت نزدیک به هم دارند کنتراست بین آنها زیاد میشود و اما مشکل دوم این که درختان روی صخره به رنگ صخره نزدیک شده و تفاوت بین صخره و درختان به راحتی قابل تمایز نمیشود علت آن اینست که چون شدت های پایین که مربوط به درختان روی صخره



استفاده از تابع محاسبه هیستوگرام

```
img = cv2.imread('Man.bmp')
img = cv2.cvtColor(img,
cv2.COLOR_BGR2GRAY)
mean,histogram = compute_hist(img)
print(mean)
rng = []
for i in range(0,256):
    rng.append(i)
fig,axs =
plt.subplots(2,1,figsize=(5,10))
axs[0].bar(rng,histogram[0,:])
axs[1].imshow(img,cmap='gray',aspect='au
to')
fig.savefig('res.png')
```



توابع مربوط به محاسبه global histeq

```
def histequal(img,histogram):
    num_total =
img.shape[0]*img.shape[1]
    cdf =
computingPdfCdf(histogram,num_total)
    eqimg = mapTable(img,cdf)
    mean,eqhist = compute_hist(eqimg)
    return eqimg,eqhist
```

```
def mapTable(img,table):
    newimg =
np.zeros(img.shape,dtype=int)
    for i in range(0,img.shape[0]):
        for j in range(0,img.shape[1]):
            newval = int(table[img[i,
j]]*255)
            newimg[i, j] = newval
    return newimg
```

```
def
computingPdfCdf(histogram,num_total):
    pdf = []
    for i in range(histogram.shape[1]):
pdf.append((histogram[0,i]/num total))
    cdf = [pdf[0]]
    for i in
range(1,histogram.shape[1]):
        cdf.append((pdf[i]+cdf[i-1]))
    return cdf
```



کد استفاده از تابع global histeq

```
HE = 1
img = cv2.imread('HE{}.jpg'.format(HE))
img = cv2.cvtColor(img,
cv2.COLOR_BGR2GRAY)
mean,histogram = compute_hist(img)
eqimg,eqhist = histequal(img,histogram)

cv2imgeq = cv2.equalizeHist(img)
rng = []
for i in range(0,256):
    rng.append(i)
fig,axs =
plt.subplots(6,1,figsize=(5,10))
axs[0].bar(rng,histogram[0,:])
axs[1].imshow(img,cmap='gray',aspect =
'auto')
```

4-کد برنامه

تابع مربوط به محاسبه هیستوگرام

```
def compute_hist(img):
    gray_scale_list =
np.zeros((1,SzeGraySclae))
    mean = 0
    for i in range(0,img.shape[0]):
        for j in range(0,img.shape[1]):
            value = img[i,j]
            gray_scale_list[0,value] +=
1
            mean += value
    mean =
mean/(img.shape[0]*img.shape[1])
    return mean,gray scale list
```

```

dwCornerRDj]
        elif i==toi-1 and j==toj-1:

localHistImg[cornerLUi:localHistImg.shape[0],cornerLUj:localHistImg.shape[1]] = \

eqimgWindow[innrWnwCornerLUi:sizeWindow,innrWnwCornerLUj:sizeWindow]
        elif i==0:

localHistImg[0:cornerRDi,cornerLUj:cornerRDj] = \

eqimgWindow[0:innrWnwCornerRDi,innrWnwCornerLUj:innrWnwCornerRDj]
        elif j==0:

localHistImg[cornerLUi:cornerRDi,0:cornerRDj] = \

eqimgWindow[innrWnwCornerLUi:innrWnwCornerRDi,0:innrWnwCornerRDj]
        elif j!=0 and i!=0:

localHistImg[cornerLUi:cornerRDi,cornerLUj:cornerRDj] = \

eqimgWindow[innrWnwCornerLUi:innrWnwCornerRDi,innrWnwCornerLUj:innrWnwCornerRDj]

        imgWindow = img[i:i+sizeWindow,-sizeWindow:img.shape[1]]
        mean,histWindow = compute_hist(imgWindow)
        eqimgWindow,eqhistWindow = histequal(imgWindow,histWindow)
        cornerLUj = toj-1 + innrWnwCornerLUj
        cornerRDj = cornerLUj + distWindow
        if i==0:

localHistImg[0:cornerRDi,cornerLUj:localHistImg.shape[1]] = \

eqimgWindow[0:innrWnwCornerRDi,innrWnwCornerLUj:sizeWindow]
        else:

localHistImg[cornerLUi:cornerRDi,cornerLUj:localHistImg.shape[1]] = \

eqimgWindow[innrWnwCornerLUi:innrWnwCornerRDi,innrWnwCornerLUj:sizeWindow]
        print(i)
        i = toi-1
        cornerLUi = cornerRDi
        for j in range(0,toj,distWindow):
            imgWindow =
img[i:i+sizeWindow,j:j+sizeWindow]
            mean,histWindow = compute_hist(imgWindow)
            eqimgWindow,eqhistWindow = histequal(imgWindow,histWindow)
            cornerLUj = j + innrWnwCornerLUj
            cornerRDj = cornerLUj + distWindow
            if j==0:

```

```

axs[2].bar(rng,eqhist[0,:])
axs[3].imshow(eqimg,cmap='gray',aspect='auto')
axs[4].hist(cv2imgeq.ravel(),bins=256,range=(0,255))
axs[5].imshow(cv2imgeq,cmap='gray',aspect='auto')

fig.savefig('../imageres/HE{}.png'.format(HE))
cv2.imwrite('../imageres/HE{}.jpg'.format(HE),eqimg)

```

کد مقایسه histeq و imadjust

```

A = imread('Man.bmp');
histogram(A(:,:,));
pause;
gamma = 2;
A = imadjust(A,[0 1],[0 1],gamma);
imshow(A(:,:,1));
imwrite(A,'res.jpg');

```

تابع محاسبه local histeq و استفاده از آن

```

def localHistEqWOL(img,histogram,sizeWindow,distWindow):
    localHistImg = np.zeros(img.shape,dtype=int)
    innrWnwCornerLUi = int((sizeWindow-distWindow)/2)
    innrWnwCornerLUj = int((sizeWindow-distWindow)/2)
    innrWnwCornerRDi = innrWnwCornerLUi + distWindow
    innrWnwCornerRDj = innrWnwCornerLUj + distWindow

    toi = img.shape[0]-sizeWindow+1
    toj = img.shape[1]-sizeWindow+1
    for i in range(0,toi,distWindow):
        cornerLUi = i + innrWnwCornerLUi
        cornerRDi = cornerLUi + distWindow
        for j in range(0,toj,distWindow):
            imgWindow = img[i:i+sizeWindow,j:j+sizeWindow]
            mean,histWindow = compute_hist(imgWindow)
            eqimgWindow,eqhistWindow = histequal(imgWindow,histWindow)
            cornerLUj = j + innrWnwCornerLUj
            cornerRDj = cornerLUj + distWindow

            if i==0 and j==0:

localHistImg[0:cornerRDi,0:cornerRDj] = \

eqimgWindow[0:innrWnwCornerRDi,0:innrWnwCornerRDj]

```

```

img = cv2.cvtColor(img,
cv2.COLOR_BGR2GRAY)
winsize = 512
movesize = 128
mean,histogram = compute_hist(img)
eqimg =
localHistEqWOL(img,histogram,winsize,mov
esize)
mean,eqhist = compute_hist(eqimg)

counter = 0
for i in range (0,len(eqhist[0])):
    counter += eqhist[0,i]
print("counter : ")
print(counter)
rng = []
for i in range(0,256):
    rng.append(i)
fig,axs = plt.subplots(2,2)
axs[0,0].bar(rng,histogram[0,:])
axs[0,1].imshow(img,cmap='gray')
axs[1,0].bar(rng,eqhist[0,:])
axs[1,1].imshow(eqimg,cmap='gray')
fig.savefig('../imageres/HE{ }({},{}) .png'
'.format(HE,winsize,movesize))
cv2.imwrite('../imageres/HE{ }({},{}) .jpg'
'.format(HE,winsize,movesize),eqimg)

```

```

localHistImg[cornerLUi:localHistImg.shap
e[0],0:cornerRDj] = \
    eqimgWindow[cornerLUi-
i:sizeWindow,0:innrWndwCornerRDj]
    else:

localHistImg[cornerLUi:localHistImg.shap
e[0],cornerLUj:cornerRDj] = \
    eqimgWindow[cornerLUi-
i:sizeWindow,innrWndwCornerLUj:innrWndwC
ornerRDj]
    j = toj -1
    imgWindow =
img[i:i+sizeWindow,j:j+sizeWindow]
    mean,histWindow =
compute_hist(imgWindow)
    eqimgWindow,eqhistWindow =
histequal(imgWindow,histWindow)
    cornerLUj = cornerRDj

localHistImg[cornerLUi:localHistImg.shap
e[0],cornerLUj:localHistImg.shape[1]] =
\
    eqimgWindow[cornerLUi-
i:sizeWindow,cornerLUj-j:sizeWindow]
    return localHistImg

HE = 1
img = cv2.imread('HE{ } .jpg'.format(HE))

```

مراجع