

Contrast Adjustment

زهرا نیازی

اطلاعات گزارش	چکیده
تاریخ: 07/09/1401	هیستوگرام یکی از ساده ترین، مفیدترین، و کم حجم ترین روش های نمایش اطلاعات عکس می باشد. با پردازش هیستوگرام عکس می توان به اطلاعاتی همانند توزیع سطوح خاکستری دست پیدا کرد و علاوه آن هیستوگرام ابزار مناسبی برای افزایش کنتراست عکس می باشد. در این گزارش با روش های مختلف افزایش کنتراست تصویر، همانند همسان سازی هیستوگرام سراسری و محلی آشنا می شویم.
واژگان کلیدی:	
مقاله	
شیوه نامه تدوین	
نویسنده	
چاپ	
شکل	
جدول	
فرمول	
نتایج	

روشی برای به دست آوردن هیستوگرام یک تصویر ارائه کنیم و حالت های مختلف بهبود کیفیت تصویر توسط local/global histogram equalization را بررسی کنیم.

2-شرح تکنیکال

1-2 متعادل سازی هیستوگرام

1-1-2 هیستوگرام

هیستوگرام نمایشی از توزیع داده های عددی است.

هیستوگرام تصویر نیز، نوعی هیستوگرام است که به عنوان نمایش گرافیکی توزیع سطح رنگی در یک تصویر دیجیتال عمل می کند. هیستوگرام، تعداد پیکسل ها را برای هر مقدار سطح رنگی رسم می کند. با مشاهده هیستوگرام

1-مقدمه

تصاویر داده های حجیمی هستند و کار کردن با آنها نیازمند پردازنده های قوی و حافظه کافی می باشد. اما برای برخی کاربردها، نیازی نیست که آرایه ی دو بعدی تصویر را داشته باشیم و میتوان با استفاده از هیستوگرام عملیات مورد نظر را انجام داد. یکی از معروف ترین عملیات، افزایش کنتراست عکس است که معمولاً باعث خوانایی عکس می شود.

هیستوگرام حاوی اطلاعاتی درباره توزیع سطوح خاکستری است. اما درباره موقعیت مکانی پیکسل ها اطلاعاتی ندارد. هیستوگرام از نظر اندازه بسیار از عکس اصلی کوچکتر است و پردازش و انجام عملیات بر روی آن خیلی سریع است و نیاز به ابزارهای پیشرفته ای ندارد، به علاوه محاسبه آن از روی عکس اصلی ساده است، به همین دلیل به عنوان مشخصه ای محبوب از عکس بخش جدانشدنی در پردازش تصویر است. در این تمرین از ما خواسته شده است که

برای یک تصویر خاص، یک بیننده قادر است در یک نگاه، درباره ی کل توزیع تصویر را نظر بدهد.

فرمول محاسبه هیستوگرام را این گونه بیان میکنیم:
 r_k برای $k = 0, 1, \dots, L-1$ شدت رنگ پیکسل k ام تصویر در نظر می گیریم. هیستوگرام غیرنرمال به شکل زیر تعریف می شود که n_k تعداد پیکسل هایی است که شدت روشنایی r_k دارند:

$$h(r_k) = n_k \quad (1)$$

به صورت مشابه هیستوگرام نرمال شده به صورت زیر تعریف میشود که M و N طول و عرض تصویر است:

$$p(r_k) = \frac{h(r_k)}{MN} \quad (2)$$

تمرین 1-2 از ما محاسبه هیستوگرام تصویر Camera Man و نمایش آن به شکل stem plot را میخواهد.

Brightness Decrease 1-1-1-2

در این قسمت از ما خواسته شده تا روشنایی تصویر را کاهش دهیم. این کار را با تقسیم کردن مقدار هر پیکسل تصویر بر 3 انجام می دهیم و تصویر خروجی را D می نامیم.

2-1-1-2

در این بخش از ما خواسته شده تا هیستوگرام تصویر اصلی و تصویری که روشنایی آن کاهش داده شده را مقایسه کنیم. تابع محاسبه هیستوگرام طبق فرمولی که پیش از این ارائه شده است پیاده سازی شده است.

Histogram Equalization 3-1-1-2

در این مرحله پیاده سازی و اعمال متعادل سازی هیستوگرام روی تصویر D از ما خواسته شده است.

متعادل کردن هیستوگرام روشی پرکاربرد در پردازش تصویر است که با استفاده هیستوگرام کنتراست را تنظیم می کند. این روش عموماً کنتراست کلی تصویر را افزایش می دهد، به خصوص اگر در عکس از سطوح خاکستری نزدیک به هم استفاده شده باشد. با استفاده از این روش، سطوح خاکستری توزیع بهتری در هیستوگرام خواهند داشت. این روش اجازه می دهد که

مکان هایی که کنتراست کمتر دارند دارای کنتراست بیشتری شوند.

این متد برای عکس هایی که پیش زمینه عکس دارای سطوح خاکستری یکسانی با جزییات عکس است، بسیار کاربردی است. از نکات کلیدی این روش این است علاوه بر ساده بودن، قابل برگشت است. به این معنی که میتوان از روی هیستوگرام متعادل شده، هیستوگرام اصلی را بدست آورد. نقطه ضعف این روش این است که می تواند باعث افزایش نویز در پیش زمینه شود، در حالیکه اطلاعات مفید را از بین می برد.

متعادل کردن باعث می شود که عکس غیر واقعی به نظر برسد و برای کاربردهای علمی مانند عکس برداری پزشکی، عکس برداری ماهواره ای بسیار ابزار قوی است. برای انجام همسان سازی هیستوگرام، به شکل زیر عمل می کنیم:

۱. هیستوگرام نرمال شده را به دست می آوریم.

برای این کار در واقع PDF را به دست می آوریم که بیانگر احتمال رخداد یک سطح خاکستری می باشد.

۲. CDF را به دست می آوریم. این کار برای هر

سطح خاکستری با جمع تک تک pdf های کمتر از آن به دست می آید. این تابع فراوانی تجمعی برای یک پیکسل را محاسبه میکند.

۳. حال همه ی مقادیر بدست آمده از مرحله

قبل را در بیشترین سطح خاکستری (که در

اینجا ۲۵۵ است) ضرب می کنیم. آرایه به

دست آمده در واقع تابعی هست که شماره

اندیس آن ورودی و خروجی سطح خاکستری

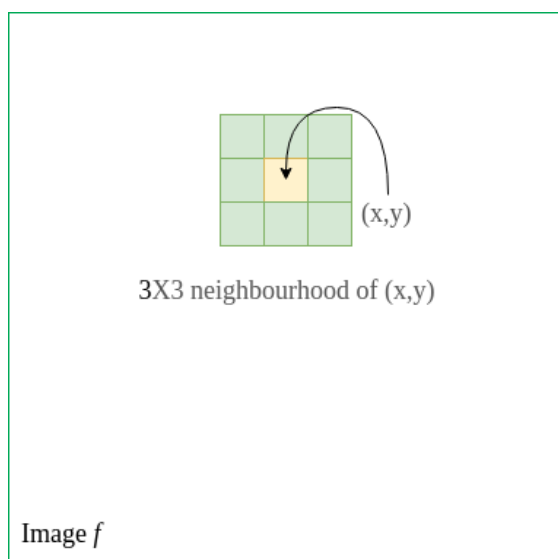
در تصویر متعادل شده است.

با استفاده از الگوریتمی که در بالا توضیح داده شد، می توان همسان شده هیستوگرام را بدست آورد. علاوه بر آن با توجه به آنچه توضیح داده شد، تابعی داریم که به ازای هر سطح خاکستری، مقدار آن را در تصویر همسان شده به ما میدهد، پس تبدیل عکس نیز ممکن و ساده است.

الگوریتم ارائه شده در بالا سرعت پایینی دارد و برای عکس های بزرگ و یا سایز پنجره های بزرگ، سرعت محدودکننده ای دارد.

Intensity Transformations 6-1-1-2

تغییر شدت (Intensity Transformation) بر روی تصاویر برای دستکاری کنتراست یا آستانه گذاری تصویر اعمال میشود. این تبدیلات در حوزه spatial هستند، یعنی مستقیماً روی پیکسل های تصویر اعمال میشوند، برخلاف اینکه بر روی تبدیل فوریه تصویر و در حوزه فرکانس انجام بشوند.



شکل 1-1-3 spatial domain

فرآیندهای حوزه spatial را می توان با استفاده از معادله زیر توصیف کرد:

$$g(x, y) = T[f(x, y)] \quad (3)$$

که در آن $f(x, y)$ تصویر ورودی است، T یک عملگر روی f است که بر روی یک همسایگی از نقطه (x, y) تعریف می شود و $g(x, y)$ تصویر خروجی است.

Local Histogram Equalization 4-1-1-2

5-1-1-2

همسان سازی محلی هیستوگرام، یک تکنیک در پردازش تصویر است که باعث افزایش کنتراست تصویر می شود. این روش با همسان سازی هیستوگرام متفاوت است. همسان سازی هیستوگرام از یک تابع برای تمامی پیکسل ها استفاده می کند و برای تصاویری مناسب است که توزیع سطوح خاکستری در تمام پیکسل های آن تقریباً یکسان است. اما در بسیاری از عکس ها، توزیع سطوح خاکستری در قسمت های مختلف تصویر، بسیار متفاوت است و دارای بخش های خیلی تیره و یا خیلی روشن هستند و استفاده از یک تابع واحد، تاثیر دلخواهی رو این نواحی نمی گذارد.

در همسان سازی محلی، هیستوگرام های متعددی محاسبه می شود که هر کدام برای یک بخش مشخص از تصویر است. و از این هیستوگرام ها برای تغییر سطح خاکستری تصویر استفاده می کنیم. بر خلاف آنچه همسان سازی انجام می دهد، در همسان سازی محلی سطح خاکستری بر اساس همسایگی پیکسل انجام میشود و به همین علت این روش، روش مناسبی برای افزایش کنتراست محلی تصویر میشود و باعث تقویت لبه ها در هر بخش از عکس می شود. با این حال همسان سازی محلی باعث افزایش noise در تصویر می شود.

در ادامه چند خاصیت همسان سازی محلی را بررسی می کنیم:

در همسان سازی محلی، سایز همسایگی که از آن به عنوان پنجره هم یاد می شود از پارامترهای مهم و تاثیر گذار است. هنگامی که یک همسایگی از نظر شدت رنگ نسبتاً همگن باشد، هیستوگرام آن دارای قله بلندی می باشد و با همسان سازی آن سعی داریم سطح کوچکی از سطوح خاکستری را به سطح وسیعی تبدیل کنیم. این باعث می شود که یک مقدار کم نویز در تصویر اصلی، در تصویر همسان شده شدت بگیرد.

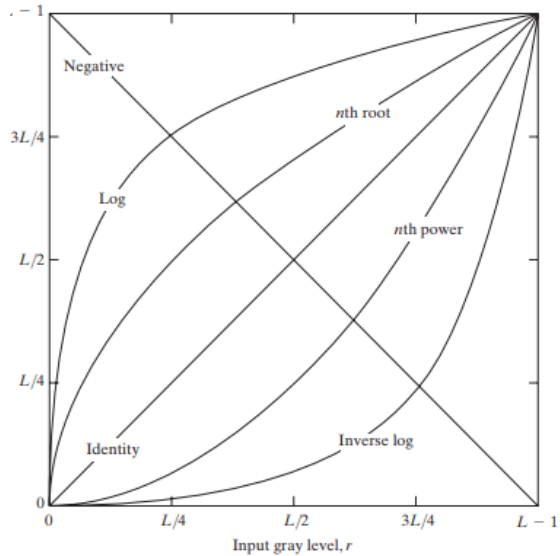
برای پیاده سازی همسان سازی محلی هیستوگرام روش های متعددی پیشنهاد شده است که ما به شکل زیر عمل میکنیم:

برای هر پیکسل:

1. همسایگی آن پیکسل را به دست می آوریم.
2. هیستوگرام نرمال شده آن را محاسبه میکنیم.
3. هیستوگرام همسان شده آن همسایگی را به دست می آوریم و فقط مقدار پیکسل مورد نظر را تغییر می دهیم.

$$s = cr^\gamma \quad (6)$$

تصحیح گاما برای نمایش صحیح تصاویر روی صفحه، برای جلوگیری از سفید شدن یا تیره شدن تصاویر هنگام نمایش تصویر از انواع مختلف مانیتور با تنظیمات نمایش متفاوت، کاربرد دارد. این کار به این دلیل انجام می شود که چشمان مانتصاویر را در یک منحنی گاما شکل درک می کنند، در حالی که دوربین ها تصاویر را به صورت خطی می گیرند. نکته قابل توجه در تبدیلات گاما این است که هنگامی که $\gamma < 1$ (که با منحنی "nth power" روی نمودار 1-1-3 نشان داده می شود)، شدت پیکسل ها کاهش می یابد، یعنی تصویر تیره تر می شود. از طرف دیگر، $\gamma > 1$ (که با منحنی مربوط به "ریشه n" در نمودار نشان داده می شود)، شدت افزایش می یابد، یعنی تصویر سبک تر می شود.



شکل 1-1-3 log transformations

از نظر ریاضی، تبدیل های log را می توان به صورت فرمول زیر بیان کرد.

$$s = c \log(1 + r) \quad (4)$$

در اینجا، s شدت روشنایی تصویر خروجی، $r \geq 0$ شدت روشنایی تصویر ورودی پیکسل، و c یک ثابت مقیاس است. c با فرمول زیر مقداردهی میشود:

$$c = 255 / (\log(1 + m)) \quad (5)$$

m حداکثر مقدار پیکسل در تصویر است. این کار برای اطمینان از اینکه مقدار پیکسل نهایی از $(L-1)$ یا 255 تجاوز نمی کند انجام می شود.

عملاً، تبدیل لگاریتمی، محدوده باریکی از مقادیر ورودی با شدت روشنایی پایین را به طیف وسیعی از مقادیر خروجی ترسیم می کند. منحنی لاگ محدوده باریکی از مقادیر با سطح خاکستری کم در تصویر ورودی را در محدوده وسیع تری از سطوح خروجی ترسیم می کند. این تبدیل برای گسترش مقادیر پیکسل های تیره در یک تصویر و در عین حال فشرده سازی مقادیر سطح بالاتر استفاده می شود. همچنین دامنه دینامیکی تصاویر را با تغییرات زیادی در مقادیر پیکسل فشرده می کند. توابع Log به ویژه زمانی مفید هستند که مقادیر سطح خاکستری ورودی ممکن است محدوده بسیار زیادی از مقادیر را داشته باشند.

تبدیل های Power Law (گاما) را می توان از نظر ریاضی به صورت فرمول زیر بیان کرد:

Histogram Equalization 2-1-2

در این بخش از ما خواسته شده که پس از اعمال همسان سازی هیستوگرام بر روی تصویر camera man، تصاویر و هیستوگرام های آنها را کنار هم رسم کنیم.

3-1-2

درباره همسان سازی سراسری هیستوگرام در قسمت قبل به تفصیل توضیح دادیم. از آنجایی که این روش، روشی بسیار پرکاربرد در پردازش تصاویر دیجیتال است و متلب نیز یکی از محبوبترین ابزارهای پردازش تصویر می باشد، این روش توسط خود متلب پیاده سازی شده است. این پیاده سازی، علاوه بر همسان سازی سراسری ویژگی های دیگری دارد که در ادامه توضیح می دهیم

- همسان سازی سراسری: همسان سازی سراسری را انجام می دهد و سعی میکند که هیستوگرام عکس داده شده را صاف کند که به معنی توزیع یکنواخت است.
- همسان سازی سراسری با گرفتن نوع عکس: در این قسمت تابع علاوه بر عکس یک آرایه عددی می گیرد که مشخص می کند مقادیر عکس ورودی در چه بازه ای هستند، مثلاً uint8 هستند و مقادیری بین ۰ تا ۲۵۵ دارند یا uint16 هستند و مقادیری بین ۰ تا ۶۵۵۳۵.

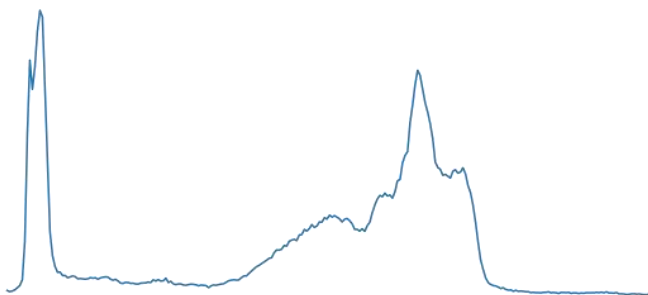
کنیم. در شرح توضیح داده شد که این روش به شدت کند است و با افزایش سایز پنجره، زمان اجرا افزایش می یابد. به همین دلیل هنگامی که از پنجره های کوچکتری استفاده بشود خروجی ها کیفیت قابل قبولی ندارند.

3- نتایج

1-1-3

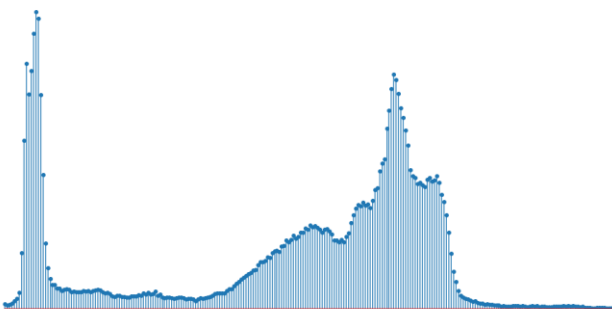


hist



شکل 31-1-3 تصویر اصلی و هیستوگرام آن

hist as stem plot



شکل 4 1-1-3 هیستوگرام به شکل یک stem plot

- همسان سازی سراسری با گرفتن تعداد bin : در این روش تابع تلاش می کند هیستوگرام را صاف کند، و هر چه n عدد کمتری از تعداد سطوح خاکستری عکس اصلی باشد، هیستوگرام خروجی صاف تر خواهد بود.
- این تابع پیاده سازی های خاصی نیز برای عکس های رنگی دارد.

تابع دیگری که باید بررسی شود تابع `imadjust` است. این تابع سطوح خاکستری و کنتراست تصویر را تغییر می دهد. به شکلی که مقادیر جدید سطوح خاکستری، در یک درصد اولیه و نهایی (تیره ترین و روشن ترین) باشد. به عبارت دیگر با دستکاری سطوح خاکستری، تجمع در هیستوگرام را در یک درصد ابتدایی و انتهایی هیستوگرام را افزایش می دهد و این کار باعث افزایش کنتراست خواهد شد.

همانند بخش قبل، این تابع نیز می تواند ورودی های دیگری بگیرد و عملیات پیچیده تری را انجام دهد.

- این تابع تصویر را به عنوان ورودی گرفته و سطوح خاکستری را طوری تغییر می دهد که تجمع در یک درصد تاریک و روشن به شدت افزایش یابد.
- می توان علاوه بر عکس، یک بازه در ورودی داد و این تابع اعداد این بازه را به بازه $[0, 1]$ تبدیل می کند. در واقع فقط سطوح خاکستری بین این بازه را تبدیل می کند و به بقیه سطوح کار ندارد.
- میتوان علاوه بر تعیین بازه ورودی، بازه خروجی را نیز تعیین کرد. مثلاً میتوان گفت از $[l, r]$ را به $[l_0, r_0]$ تصویر کن.
- با گرفتن یک تابع به عنوان ورودی خروجی را چنان تولید می کند که رابطه بین پیکسل های عکس ورودی و خروجی از آن تابع پیروی کند.
- این تابع همانند تابع قبلی، تنظیمات ویژه ای مخصوص عکسهای سیاه و سفید دارد.

2-2 همسان سازی محلی هیستوگرام

همانطور که در شرح توضیح داده شد، برای اجرای همسان سازی محلی اندازه پنجره های متفاوتی میتوان انتخاب کرد. به دلیل اینکه بتوان مقایسه خوبی انجام داد، نتیجه اجرای الگوریتم های مختلف همسان سازی را بر هر تصویر پشت سر هم بررسی می کنیم.

عملکرد الگوریتم توضیح داده شده در قسمت های قبل، یعنی مقدار دادن پیکسل مرکزی در هر مرحله را بررسی می

3-1-1-3



شکل 7 1-1-3 H (D Histeq)

1-1-1-3

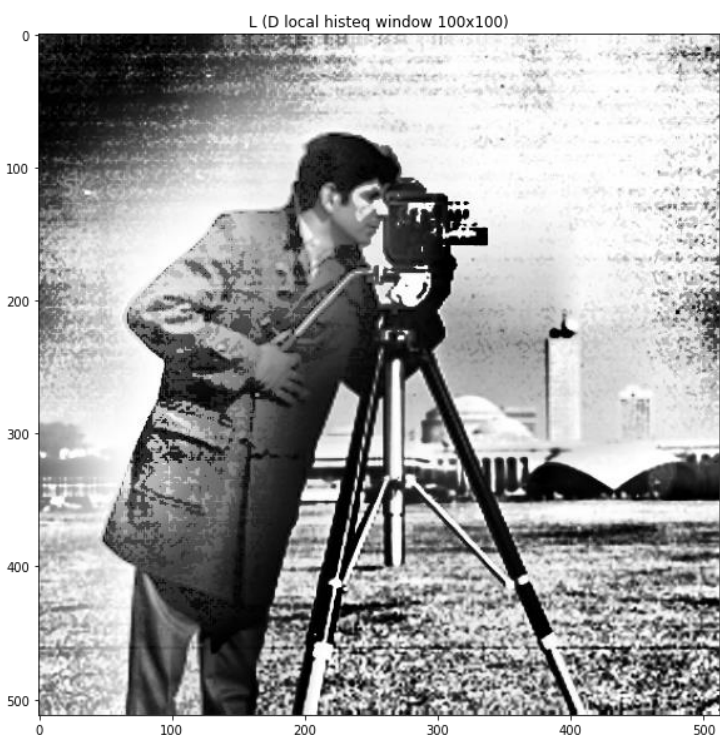


شکل 5 1-1-3 D = (camera_man / 3)

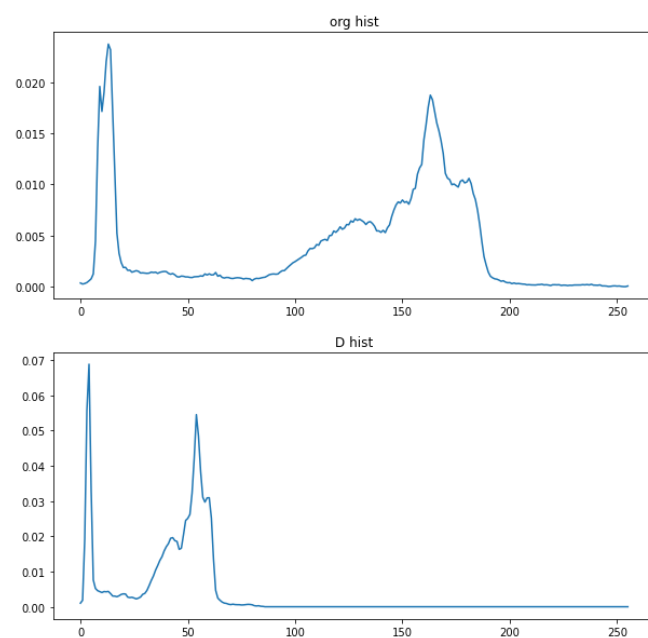
2-1-1-3

همانطور که در تصاویر مشاهده می شود، مقادیر سطح خاکستری همه به سمت چپ شیفت پیدا کرده و متراکم تر شده اند.

4-1-1-3



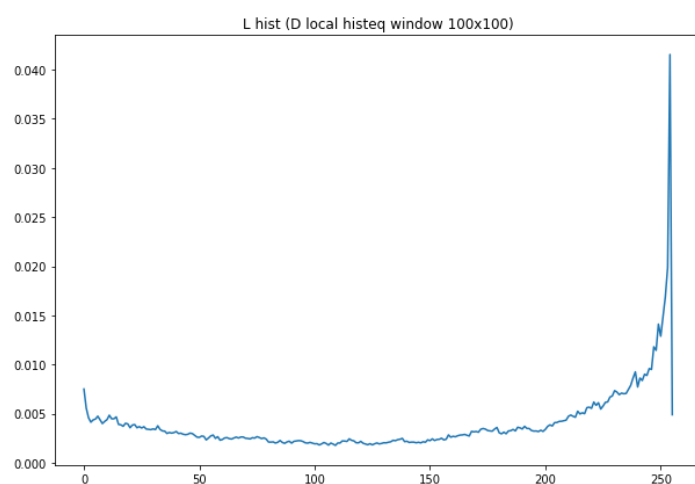
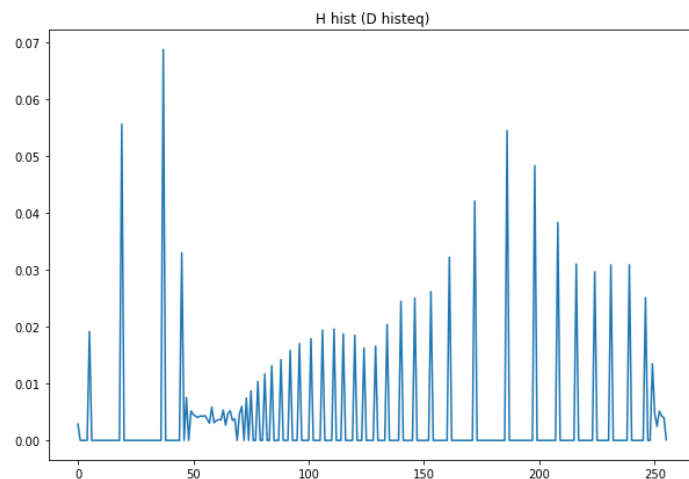
شکل 8 1-1-3 L (D local histeq window=100x100)



شکل 61-1-3 هیستوگرام تصویر اصلی و D

5-1-1-3

همانطور که در تصویر مشاهده می شود همسان سازی محلی باعث هموارتر شدن هیستوگرام تصویر شده است که این مطلوب ماست. چون همسان سازی بر اساس همسایگی انجام شده است، کنتراست محلی تصویر افزایش پیدا کرده است و باعث تقویت لبه ها در بخش های مختلف تصویر شده است.



شکل 9 1-1-3 هیستوگرام H و L

6-1-1-3

از آنجایی که اکثر پیکسل های تصویر D، سطوح خاکستری بسیار تیره ای بودند، با توجه به تابع لگاریتم میتوان توقع داشت که پیکسل هایی در محدوده (0 و $L/4$) از تصویر ورودی به محدوده ($0, 3L/4$) تصویر خروجی مپ شود و 75٪ باقی مانده روشن تصویر در 25٪ روشن تصویر خروجی قرار گیرد.

با بررسی تصاویر میتوان مشاهده کرد که پس از اعمال تبدیل لگاریتمی، تصویری که نواحی تیره بسیار زیادی داشت، روشن

تر شده است. در این حالت جزئیاتی که در نواحی تیره تصویر وجود داشتند و قابل تشخیص نبودند حال واضح تر شده اند. مانند جزئیات روی کت و دستکش شخص درون عکس. عکس تبدیل لگاریتم دقیقاً طبق نامش عکس لگاریتم عمل میکند. این تبدیل از فرمول زیر به دست می آید:

$$s = 10^{\frac{r}{c}} - 1$$

این تبدیل محدوده ای وسیع از مقادیر با سطح خاکستری کم در تصویر ورودی را در محدوده باریک تری از سطوح خروجی ترسیم می کند. به همین دلیل ما تیره تر شدن تصویر را مشاهده میکنیم. اعمال این تبدیل باعث میشود نواحی بسیار روشن تصویر D مانند آسمان روشنایی بسیار زیادی نداشته باشد. در انتها تابع تبدیل گاما با گامای برابر با 0.6 را به تصویر خروجی اعمال میکنیم. چون تصویری که به عنوان ورودی به گاما دادیم تصویری تیره است، گاما را کوچکتر از 1 میگیریم تا کیفیت آن را بهبود بخشد. مشاهده میکنیم که نسبت به تصویر اصلی (D) تفاوت های فاحشی رخ داده است. مانند روشن تر شدن نواحی لباس فرد ایستاده در تصویر و نمایش جزئیات بیشتر آن. و روشن تر شدن آسمان، صورت فرد و ساختمان های پس زمینه.



شکل 1 6-1-1-3 original image



شکل 6-1-1-3 gamma transform 4

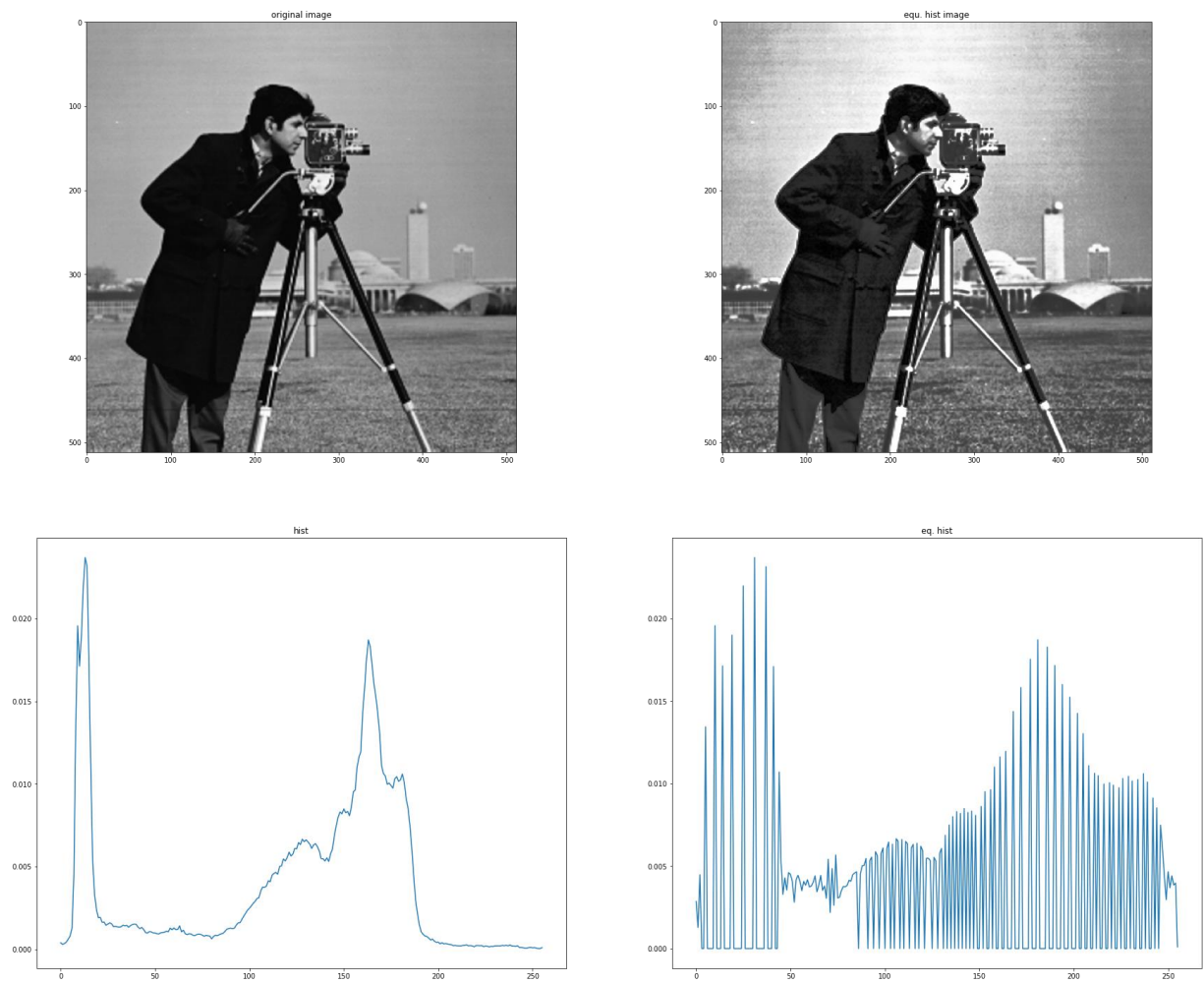


شکل 6-1-1-3 log transform 2



شکل 6-1-1-3 inverse log transform 3

1-2-3



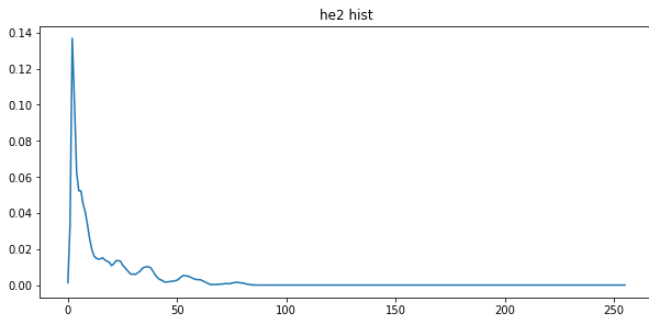
شکل 1-2-3 تصویر اصلی و همسان سازی شده

3-3 Local Histogram Equalization

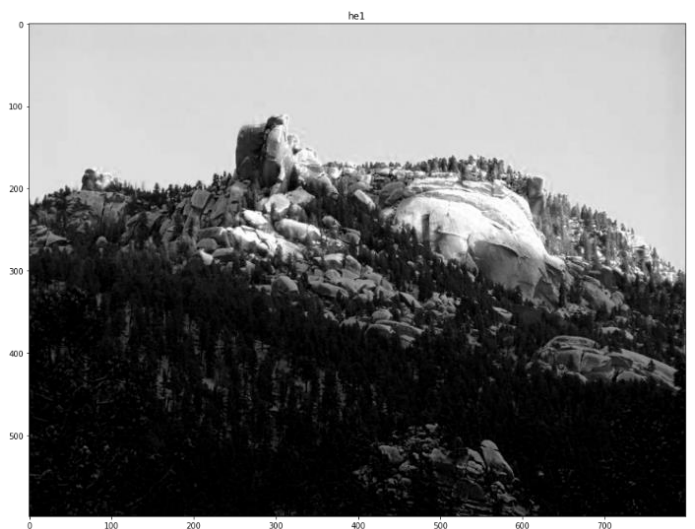
در ابتدا هیستوگرام 4 تصویر را رسم میکنیم. همانطور که در شکل 1-3-3 تا 4-3-3 مشاهده میشود، تمامی این تصاویر کیفیت پایینی دارند. در برخی تجمع تمام پیکسل ها در مقدار سطوح خاکستری بالا و در برخی تجمع در مقدار سطوح خاکستری پایین می باشد و این در هیستوگرام ها به وضوح دیده می شود. علاوه بر این در تصویر he4 جزئیات زیادی وجود دارد که باعث میشود به طور کلی پیکسل های زیادی در مقدار سطوح خاکستری بالا قرار داشته باشند.



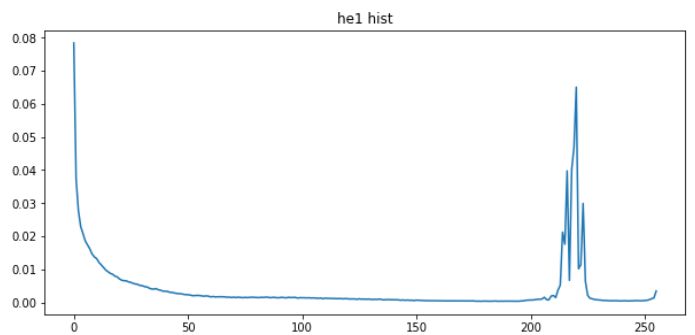
شکل 3-3 he2



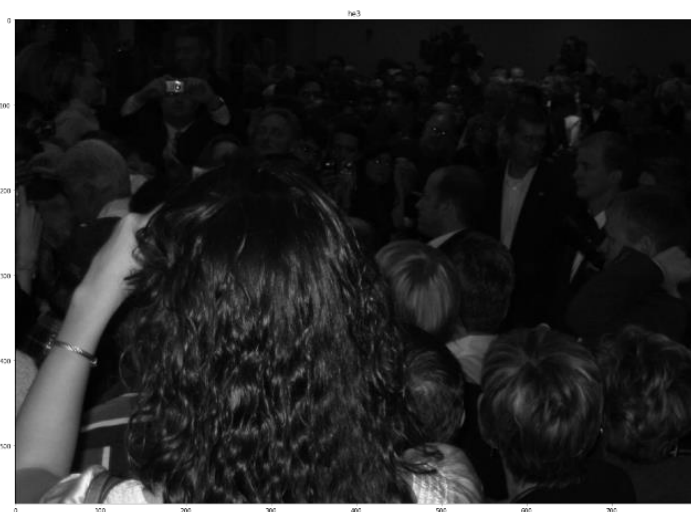
شکل 3-3 هیستوگرام he2



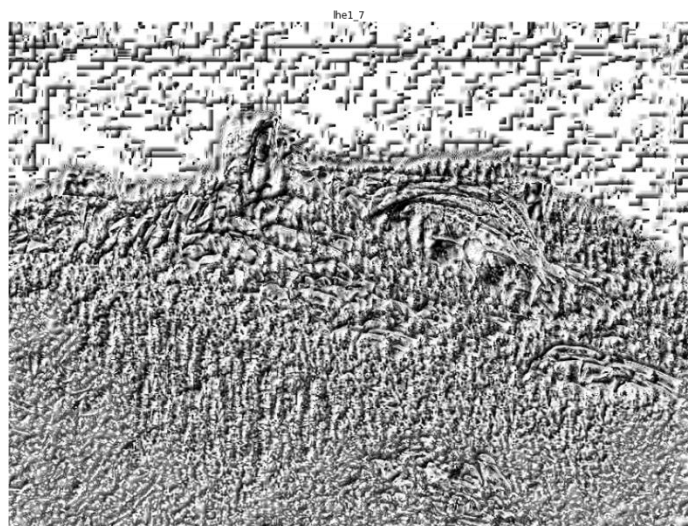
شکل 3-3 he1



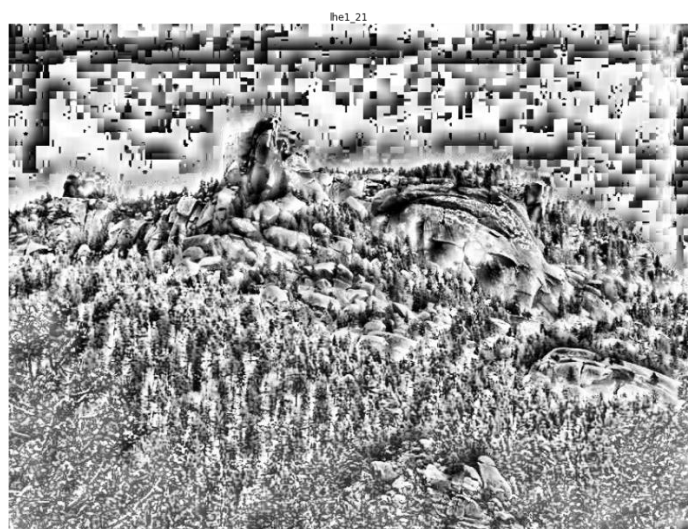
شکل 3-3 هیستوگرام he1



شکل 3-3 he3



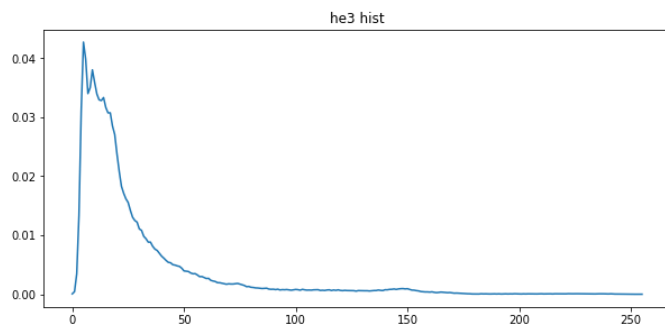
شکل 3-3 LHE3 window size (7x7)9



شکل 3-3 LHE1 window size (21x21)10



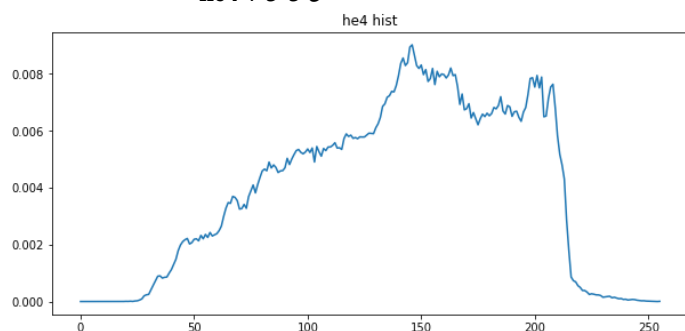
شکل 3-3 LHE1 window size (50x50)11



شکل 3-3 هیستوگرام he3



شکل 3-3 he4 7



شکل 3-3 هیستوگرام he4

حال تابع LHE را با اندازه پنجره های مختلف به تصاویر he1 و he3 و he4 اعمال کرده و نتایج را مشاهده میکنیم.

lhe3 15



شکل 3-3 (15x15) LHE3 window size

lhe3 21



شکل 3-3 (21x21) LHE3 window size

lhe3 50

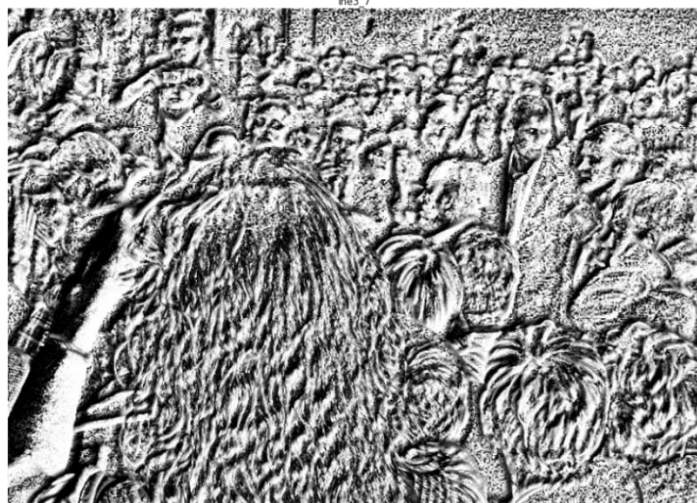


شکل 3-3 (50x50) LHE3 window size

همانطور که مشاهده میشود با افزایش سایز پنجره کیفیت تصویر خروجی بهتر شده و شاهد کاهش نویز هستیم. همانطور که شرح داده شده است، در همسان سازی محلی و در همسایگی هایی که مقدار همگنی دارند، نویز بسیار بسیار زیاد است. خروجی های بالا نیز گواهی بر این حرف است.

افزایش سایز پنجره، باعث کاهش همگان شدن همسایگی می شود زیرا ناحیه بزرگتری را بررسی می کنیم که امکان همگان بودن را پایین می آورد، پس احتمال تقویت نویز در پنجره های بزرگتر کمتر است.

lhe3 7



شکل 3-3 (7x7) LHE3 window size

در این تصویر، در عکس اصلی تقریباً هیچ اطلاعاتی از افرادی که عقب ایستاده اند وجود ندارد و فقط میتوان چند نفر جلو را به دقت دید. پس از انجام همسان سازی محلی، جزییات کاملاً قابل مشاهده است و بنابراین کنتراست خروجی افزایش یافته است. و درواقع افزایش سایز پنجره باعث بهبود کیفیت تصویر و کاهش نویز می شود.

lhe4_50



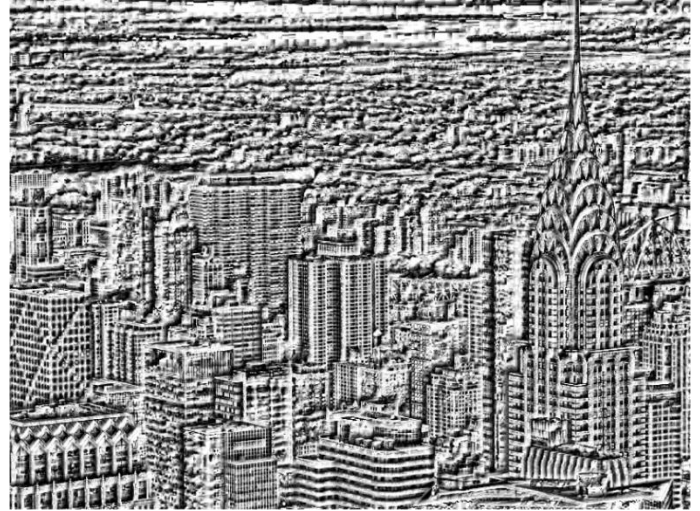
شکل 3-3 18 (50x50) LHE4 window size

می بینیم که همسان سازی محلی در عکس هایی با سطح خاکستری بالا نیز به خوبی عمل می کند و عکس از کیفیت خوبی برخوردار است.

مشاهده میکنیم در حالتی که توزیع سطوح خاکستری در قسمت های مختلف تصویر، بسیار متفاوت باشد و تصویر دارای بخش های خیلی تیره و یا خیلی روشن هستند و جزئیات بسیار زیادی در اعماق مختلف در تصویر داشته باشیم، آنگاه روش همسان سازی محلی، جزئیات بیشتری را نمایان میکند. به دلیل اینکه در همسان سازی محلی، سطح خاکستری بر اساس همسایگی پیکسل تعیین میشود و به همین علت افزایش کنتراست محلی تصویر را در نمونه های زیر مشاهده میکنیم. این افزایش کنتراست محلی همچنان باعث تقویت لبه ها در هر بخش از عکس می شود که برای تصاویر با جزئیات زیاد بسیار موثر است.

حال اندازه پنجره را خیلی زیاد در نظر گرفته و دوباره LHE را اجرا میکنیم.

lhe4_7



شکل 3-3 16 (7x7) LHE4 window size

lhe4_21



شکل 3-3 17 (21x21) LHE4 window size

در این عکس نیز به دلیل شباهت پیش زمینه و محتوای اصلی عکس، همسان سازی سراسری تاثیر قابل قبولی دارد. تاثیر افزایش سایز پنجره بر کیفیت در این سری از عکس ها کاملاً مشخص است.

عکس هایی که سایز پنجره در آنها کوچک است به شدت نویز دار بودند و با افزایش سایز پنجره کیفیت رفته رفته بهبود می یابد.

همانطور که این روش عکس های تیره را روشنتر می کند، به تیرگی عکس های روشن می افزاید. بر خلاف دو عکس قبلی این عکس، عکسی با سطح خاکستری بالاست و در پیش زمینه جزئیات واضح نیستند.

lhe3



شکل 3-3 21 LHE3 window size (100x400)

lhe1



شکل 3-3 19 LHE1 window size (100x400)

lhe4



شکل 3-3 22 LHE1 window size (100x400)

lhe2

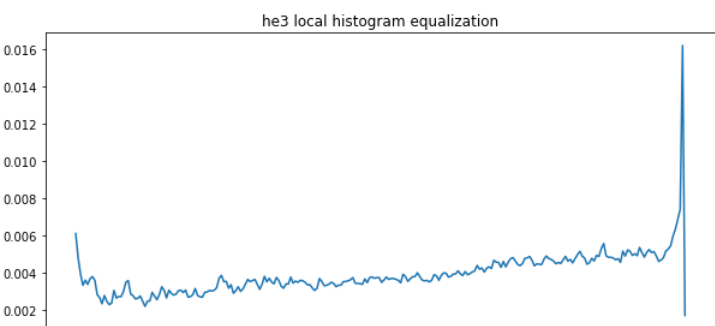
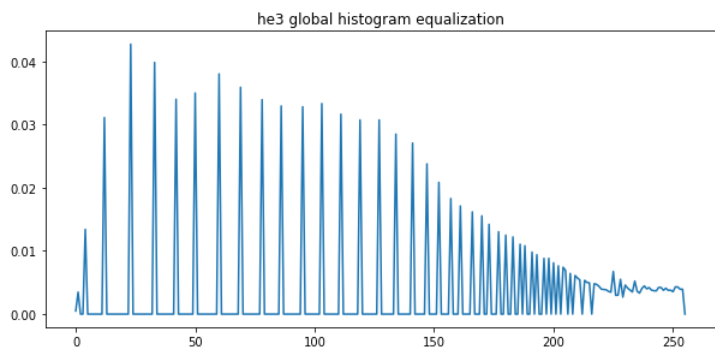


شکل 3-3 20 LHE2 window size (100x400)

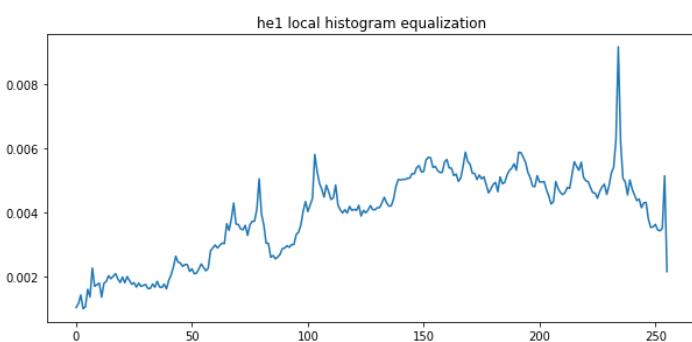
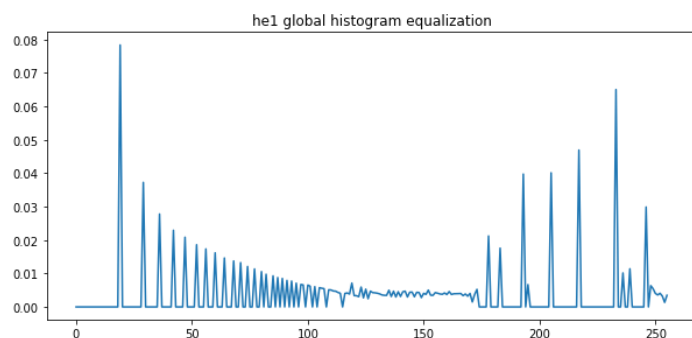
جزئیات انتهای تصویر در همسان سازی محلی بهتر نمایان شده و به طور کل کنتراست عکس پس از همسان سازی محلی افزایش چشم گیر تری داشته است.

در تصاویر زیر، مقایسه همسان سازی هیستوگرام global و local انجام شده است. روش همسان سازی سراسری برای عکس هایی مناسب است که توزیع سطوح خاکستری در زمینه و محتوای اصلی عکس تقریباً مشابه باشد. اما در این عکس چنین نیست، به همین علت خروجی همسان سازی محلی، جزئیات بیشتری را نمایان میکند، هر چند در روش اصلی به علت وجود نویز کیفیت تصویر قابل قبول نیست، اما در صورتی که نیاز به

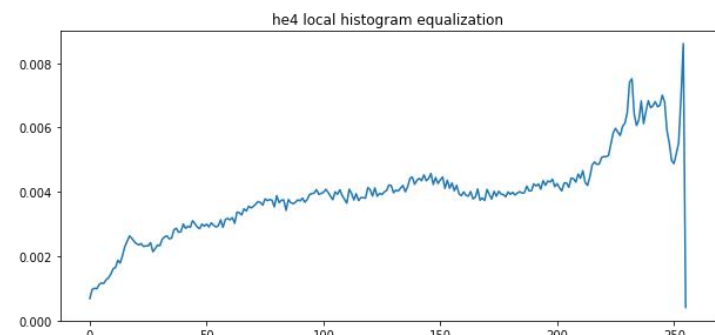
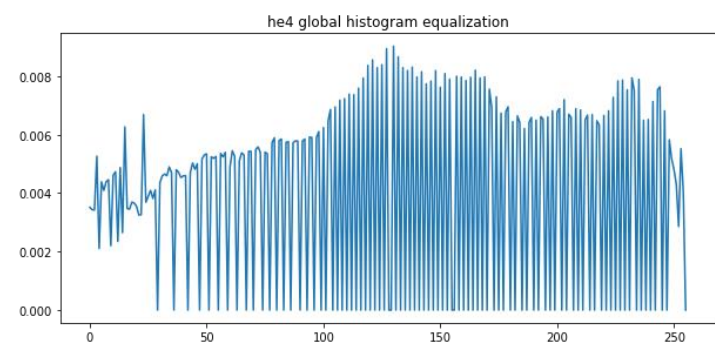
فهمیدن جزئیاتی در عکس باشد، بررسی خروجی همسان سازی محلی راه حل مناسبتری است.



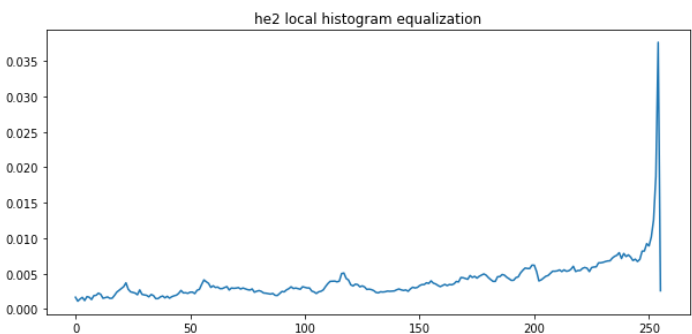
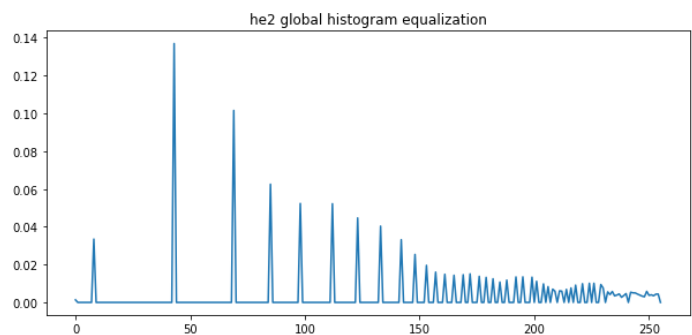
شکل 3-25 مقایسه هیستوگرام سراسری و محلی he3



شکل 3-23 مقایسه هیستوگرام سراسری و محلی he1



شکل 3-26 مقایسه هیستوگرام سراسری و محلی he4



شکل 3-24 مقایسه هیستوگرام سراسری و محلی he2


```

        return new_img , Hist

camera_man = cv2.imread('/content/sample_data/Camera Man
.bmp', cv2.IMREAD_GRAYSCALE)

##2.1. Histogram Equalization

###2.1.1 illustrate the histogram as a stem plot

plt.figure(figsize=(30,10))

plt.subplot(121)
plt.imshow(camera_man)
plt.title('original image')

orghist = hist(np.uint8(camera_man))

plt.subplot(122)
plt.plot(orghist)
plt.title('hist')
plt.set_cmap('gray')

plt.show()

plt.subplot(111)
plt.stem(orghist)
plt.subplots_adjust(right=2.5, wspace=0.1)
plt.show()

####2.1.1.1. Decrease the brightness of Camera Man by dividing the intensity values by 3 and named output as D.

D = camera_man//3

```

Code-4

```

import numpy as np
import matplotlib.pyplot as plt
import cv2
import math

```

1-1-4

```

def hist(image):
    m, n = image.shape
    hist = [0.0] * 256
    for i in range(m):
        for j in range(n):
            hist[image[i, j]]+=1
    return np.array(hist)/(m*n)

def cdf(hist):
    return [sum(hist[:i+1]) for i in range(len(hist))]

def histeq(image):
    h = hist(image)
    cumsum = np.array(cdf(h))
    sk = np.uint8(255 * cumsum)

    s1, s2 = image.shape
    new_img = np.zeros_like(image)
    # applying transferred values for each pixels
    for i in range(0, s1):
        for j in range(0, s2):
            new_img[i, j] = sk[image[i, j]]
    Hist = hist(new_img)
    #return transformed image, original and new histogram,

    # and transform function

```

```

        for j in range(0, w)
:
            # Get the block
            blk = im[i: min(
i + tile_x, h), j: min(j + tile_y, w)]
            probs = get_distr(blk)
            out[i: min(i + tile_x, h), j: min(j + tile_y, w)] = CHE(blk, probs)
        return out

def CHE(im, probs):
    T = np.array(list(map(int, 255*np.cumsum(probs))))
    return T[im]

def get_distr(im):
    hist, _ = np.histogram(im.flatten(), 256, [0, 256])
    return hist / hist.sum()

L = AHE(D, 100, 100)
plt.figure(figsize=(10, 10))
plt.imshow(L)
plt.title('L (D local histeq)')

plt.show()

####2.1.1.5 H and L hists. *
What's the main difference between local and global histogram equalization?

plt.figure(figsize=(10, 25))

plt.subplot(121)
plt.title('D hist')

```

```

plt.figure(figsize=(10, 10))
plt.imshow(D)

####2.1.1.2. Plot the histograms of Input and D. *What can you observe from these two histograms?
plt.subplot(121)
plt.title('org hist')
plt.plot(orghist)

dhist = hist(np.uint8(D))

plt.subplot(122)
plt.plot(dhist)
plt.title('D hist')
plt.subplots_adjust(right=2.5, wspace=0.1)

plt.show()

####2.1.1.3. histeq on D
H, dhisteq = histeq(np.uint8(D))

plt.figure(figsize=(10, 10))
plt.imshow(H)
plt.title('H (D histeq)')

plt.show()

####2.1.1.4 local histeq on D
def AHE(im, tile_x=8, tile_y=8):
    h, w = im.shape
    out = np.zeros(im.shape)
    # Declare output variable

    for i in range(0, h):

```

```

        for j in range(0, im
g.shape[1]-1):
            f = img[i,j]
            pixel = round(lo
g_transform(c, f))
            s[i,j]=pixel
        return s

#####inverse log transform
def inverse_log_transform(c,
f):
    p = float(f / c)
    g = 10**p
    return float(g-1)

def inverse_log_transform_im
age(img, outputMax = 255, in
putMax=255):
    c = outputMax/math.log(i
nputMax+1,10);
    s = np.zeros(img.shape)
    for i in range(0, img.sh
ape[0]-1):
        for j in range(0, im
g.shape[1]-1):
            f = img[i,j]
            pixel = round(in
verse_log_transform(c, f))
            s[i,j]=pixel
        return s

#####gamma transform
def gamma_transform(c, f, ga
mma):
    g = float(f)**gamma
    return g*c

def gamma_transform_image(im
g, gamma, outputMax = 255, i
nputMax=255):
    c = outputMax/math.log(i
nputMax+1,10);

```

```

plt.plot(dhisteq)

Lhist = hist(np.uint8(L))

plt.subplot(122)
plt.plot(Lhist)
plt.title('L hist')
plt.subplots_adjust(right=2.
5, wspace=0.1)

plt.show()

####2.1.1.6 enhance D by per
forming log transform, inver
se log transform and power-
law transform. *adjust the
parameters to obtain the res
ults as best as you can. Sho
w the parameters, resultant
images and corresponding his
tograms. Provide some discus
sions on the results as well
.*

#####log transform

def log_transform(c, f):
    g = c * math.log(float(1
+ f),10)
    return g

def log_transform_image(img,
outputMax = 255, inputMax=2
55):
    c = outputMax/math.log(i
nputMax+1,10);
    s = np.zeros(img.shape)
    for i in range(0, img.sh
ape[0]-1):

```

```
plt.title('gamma_transformed_image')

plt.show()
```

2-1-4

```
###2.1.2. histeq
orghist = hist(np.uint8(camera_man))
equimg, ehist = histeq(camera_man)
plt.figure(figsize=(20,20))
plt.subplots_adjust(right=2,
    wspace=0.1)

plt.subplot(221)
plt.title('original image')
plt.imshow(camera_man)

plt.subplot(222)
plt.title('equ. hist image')
plt.imshow(equimg)

plt.subplot(223)
plt.title('hist')
plt.plot(orghist)

plt.subplot(224)
plt.title('eq. hist')
plt.plot(ehist)

plt.show()
```

```
s = np.zeros(img.shape)
for i in range(0, img.shape[0]-1):
    for j in range(0, img.shape[1]-1):
        f = img[i,j]
        pixel = round(gamma_transform(c, f, gamma))
        s[i,j]=pixel
    return s
```

```
plt.figure(figsize=(20,15))

log_transformed_image = log_transform_image(D)
inverse_log_transformed_image = inverse_log_transform_image(D)
gamma_transformed_image = gamma_transform_image(D,0.6)

plt.subplot(411)
plt.imshow(D)
plt.title('original image')

plt.subplot(412)
plt.imshow(log_transformed_image)
plt.title('log_transformed_image')
plt.set_cmap('gray')

plt.subplot(413)
plt.imshow(inverse_log_transformed_image)
plt.title('inverse_log_transformed_image')

plt.subplot(414)
plt.imshow(gamma_transformed_image)
```

```

plt.title('he2')
plt.imshow(he2)

plt.subplot(424)
plt.title('he2 hist')
plt.plot(he2hist)

plt.subplot(425)
plt.title('he3')
plt.imshow(he3)

plt.subplot(426)
plt.title('he3 hist')
plt.plot(he3hist)

plt.subplot(427)
plt.title('he4')
plt.imshow(he4)

plt.subplot(428)
plt.title('he4 hist')
plt.plot(he4hist)

plt.show()

lhe1_7=AHE(he1,7,7)
lhe1_21=AHE(he1,21,21)
lhe1_50=AHE(he1,50,50)

lhe3_7=AHE(he3,7,7)
lhe3_15=AHE(he3,15,15)
lhe3_21=AHE(he3,21,21)
lhe3_50=AHE(he3,50,50)

lhe4_7=AHE(he4,7,7)
lhe4_21=AHE(he4,21,21)
lhe4_50=AHE(he4,50,50)

```

```

##2.2 LHE
he1 = cv2.imread('/content/sample_data/HE1.jpg', cv2.IMR
EAD_GRAYSCALE)
he2 = cv2.imread('/content/sample_data/HE2.jpg', cv2.IMR
EAD_GRAYSCALE)
he3 = cv2.imread('/content/sample_data/HE3.jpg', cv2.IMR
EAD_GRAYSCALE)
he4 = cv2.imread('/content/sample_data/HE4.jpg', cv2.IMR
EAD_GRAYSCALE)

plt.figure(figsize=(25,30))
plt.subplots_adjust(right=1,
    wspace=0.1)

he1hist = hist(np.uint8(he1)
)
he2hist = hist(np.uint8(he2)
)
he3hist = hist(np.uint8(he3)
)
he4hist = hist(np.uint8(he4)
)

plt.subplot(421)
plt.title('he1')
plt.imshow(he1)

plt.subplot(422)
plt.title('he1 hist')
plt.plot(he1hist)

plt.subplot(423)

```

```

plt.subplot(222)
plt.title('he1 local histogram equalization')
plt.plot(lhehist1)

plt.figure(figsize=(10,10))

plt.subplot(211)
plt.title('he2 global histogram equalization')
plt.imshow(he2hist)

plt.subplot(222)
plt.title('he2 local histogram equalization')
plt.plot(lhehist2)

plt.figure(figsize=(10,10))

plt.subplot(211)
plt.title('he3 global histogram equalization')
plt.imshow(he3hist)

plt.subplot(222)
plt.title('he3 local histogram equalization')
plt.plot(lhehist3)

plt.figure(figsize=(10,10))

plt.subplot(211)
plt.title('he4 global histogram equalization')
plt.imshow(he4hist)

```

```

lhe1=AHE(he1,100,400)
lhe2=AHE(he2,100,400)
lhe3=AHE(he3,100,400)
lhe4=AHE(he4,100,400)

#hist of local eq
lhehist1 = hist(np.uint8(lhe1))
lhehist2 = hist(np.uint8(lhe2))
lhehist3 = hist(np.uint8(lhe3))
lhehist4 = hist(np.uint8(lhe4))

#global histeq
he1equimg, he1hist = histeq(he1)
he2equimg, he2hist = histeq(he2)
he3equimg, he3hist = histeq(he3)
he4equimg, he4hist = histeq(he4)

plt.figure(figsize=(10,10))

plt.subplot(211)
plt.title('he1 global histogram equalization')
plt.imshow(he1hist)

```

```
plt.subplot(222)
plt.title('he4 local histogram equalization')
plt.plot(lhehist4)
```