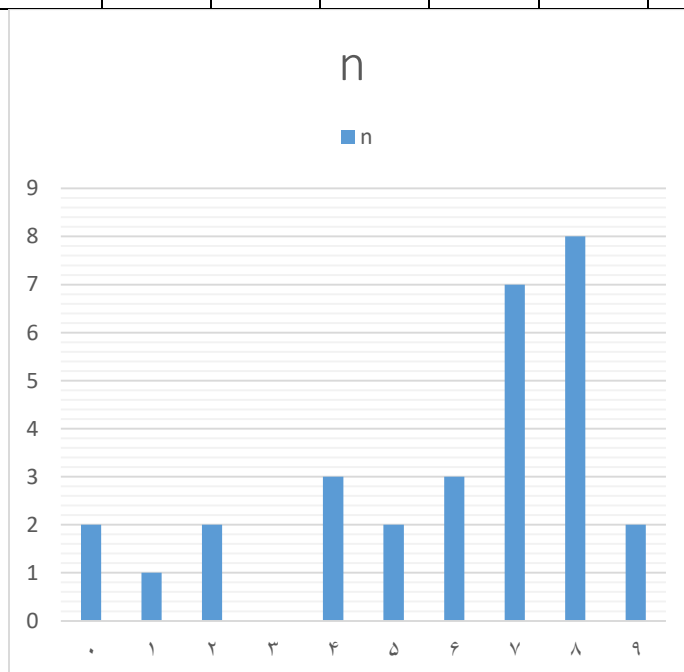


سوال اول) ۱- هیستوگرام تصویر ۲- برش ۱۰ درصدی هیستوگرام و اعمال کشش هیستوگرام ۳- برش ۱۰ درصدی هیستوگرام و اعمال متعادل سازی هیستوگرام:

۷	۷	۸	۸	۸	۸
۲	۱	۴	۴	۴	۸
۷	۰	۵	۵	۲	۸
۸	۰	۶	۹	۹	۷
۸	۷	۶	۶	۷	۷

(۱-۱)

k	0	1	2	3	4	5	6	7	8	9
n_k	2	1	2	0	3	2	3	7	8	2



(۱-۲)

تعداد کل نقاط:

$$۵ \times ۶ = ۳۰$$

۱۰ درصد نقاط:

$$۰.۱ \times ۳۰ = ۳$$

باید ۳ نقطه از ابتدا و ۳ نقطه از انتها حذف شود:

۳ نقطه ی ابتدایی به ترتیب دارای شدت روشنایی ۰، ۰، ۱ هستند.

۳ نقطه ی انتهایی به ترتیب دارای شدت روشنایی ۹، ۹، ۸ هستند.

فرمول کشش هیستوگرام:

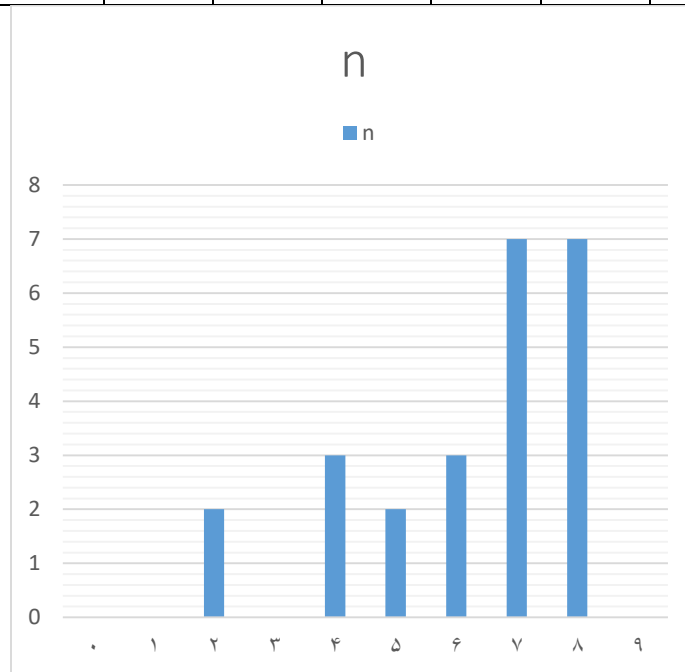
$$g(x, y) = stretch[f(x, y)] = \left(\frac{f(x, y) - f_{min}}{f_{max} - f_{min}} \right) (\max - \min) + \min$$

فرمول کشش هیستوگرام بعد از برش ۱۰ درصد:

$$g(x, y) = clip[f(x, y)] = \left(\frac{f(x, y) - f_{10}}{f_{90} - f_{10}} \right) (\max - \min) + \min$$

بعد از حذف ۱۰ درصد ابتدایی و انتهایی دوباره داده ها و نمودار را ثبت میکنیم:

k	0	1	2	3	4	5	6	7	8	9
n_k	0	0	2	0	3	2	3	7	7	0



با توجه به نمودار و جدول میتوان فهمید $f_{min} = 2$ زیرا در نقاط با شدت روشنایی کمتر از آن یعنی صفر و یک تعداد نقاط با این شدت روشنایی صفر است. (یادآوری f تابع شدت روشنایی نقاط است و f_{min} کمترین مقدار شدت روشنایی) همچنین $f_{max} = 8$ میباشد. Max و min هم تغییری نکردند و همان ۰ و ۹ هستند. با قرار دادن این مقادیر در فرمول کشش هیستوگرام، فرمول $g(x, y)$ بدست می آید:

$$g(x, y) = clip[f(x, y)] = \left(\frac{f(x, y) - 2}{8 - 2} \right) (9 - 0) + 0 = 1.5 \times (f(x, y) - 2) = 1.5f(x, y) - 3$$

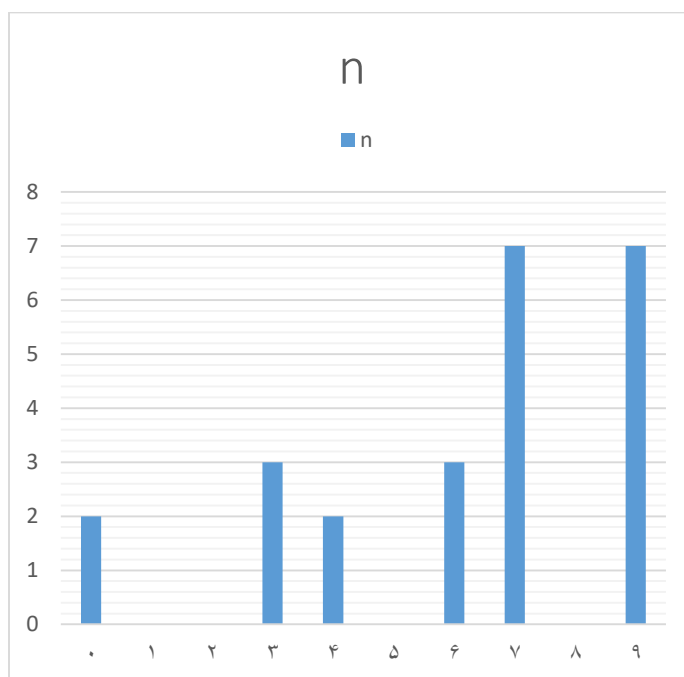
با توجه به رابطه ی بدست آمده در بعضی از نقاط مقدار جدید شدت روشنایی پس از تبدیل اعشاری خواهد شد پس باید یک مرحله هم جواب را تقریب بزنیم. زیرا باید مقادیر پیکسل ها در دامنه وجود داشته باشند. در این سوال دامنه اعداد گسسته ی ۰ تا ۹ هستند پس اعداد منفی را صفر و اعداد بیش از ۹ را ۹ و اعداد اعشاری را با حذف اعشار تقریب میزنیم.

$f(x, y)$	0	1	2	3	4	5	6	7	8	9
$g(x, y) = 1.5f - 3$	-3	-1.5	0	1.5	3	4.5	6	7.5	9	10.5
round	0	0	0	1	3	4	6	7	9	9
n	0	0	2	0	3	2	3	7	7	0

با توجه به اینکه بعضی مقادیر به مقدار یکسانی نظیر شدند، مثلاً ۰ و ۱ و ۲ هر سه در تصویر جدید به ۰ تبدیل شدند باید جدول جدیدی برای کشیدن نمودار هیستوگرام رسم کنیم تا قابل فهم تر باشد.

k	0	1	2	3	4	5	6	7	8	9
n_k	2	0	0	3	2	0	3	7	0	7

هیستوگرام تصویر جدید ارتقایافته:



همانطور که با مقایسه ی این نمودار با قبلی مشخص است انگار همان نمودار به صورت افقی از دو طرف کسیده شده است.

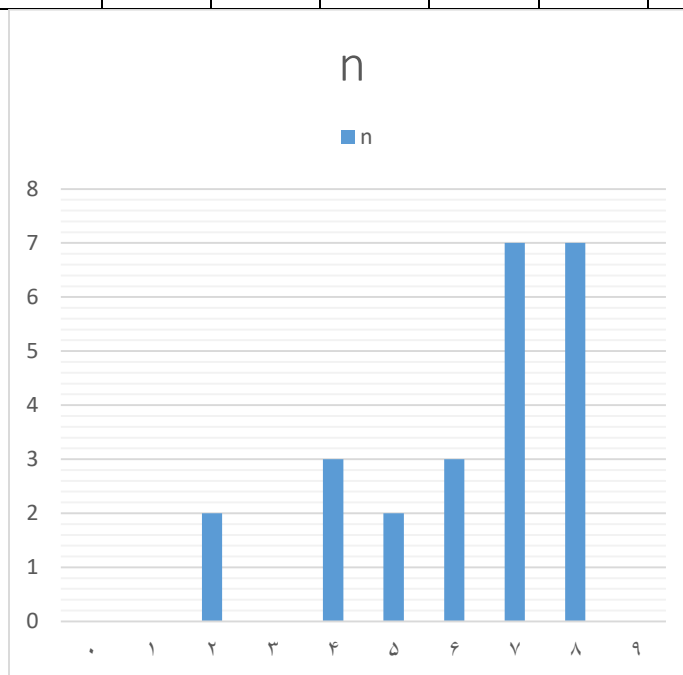
تصویر جدید:

7	7	9	9	9	9
0	0	3	3	3	9
7	0	4	4	0	9
9	0	6	9	9	7
9	7	6	6	7	7

(۱-۳)

باز هم مثل مورد قبل ابتدا ۱۰ درصد نقاط را از دو طرف حذف میکنیم. ۳۰ نقطه داریم که ۱۰ درصد میشود ۳ نقطه.

k	0	1	2	3	4	5	6	7	8	9
n_k	0	0	2	0	3	2	3	7	7	0



به همان نمودار و جدول ابتدای قسمت قبل میرسیم. سپس متعادل سازی را اعمال میکنیم.

در متعادل سازی به جای استفاده از فرمول های خطی برای تبدیل از فرمول هایی استفاده میکنیم که هیستوگرام تصویر را مسطح کند و ناحیه ی تیره را جدا و ناحیه ی روشن را جدا گسترش میدهد تا حد امکان چیزی حذف نشود. اساس این تبدیل تئوری احتمالات است که در آن هیستوگرام به عنوان تابع توزیع احتمال سطوح روشنایی تصویر در نظر گرفته شود. در حقیقت تابعی هست که این توزیع احتمال را به توزیع احتمال یکنواخت تبدیل کند.

$$p_r(r_k) = \frac{n_k}{n}$$

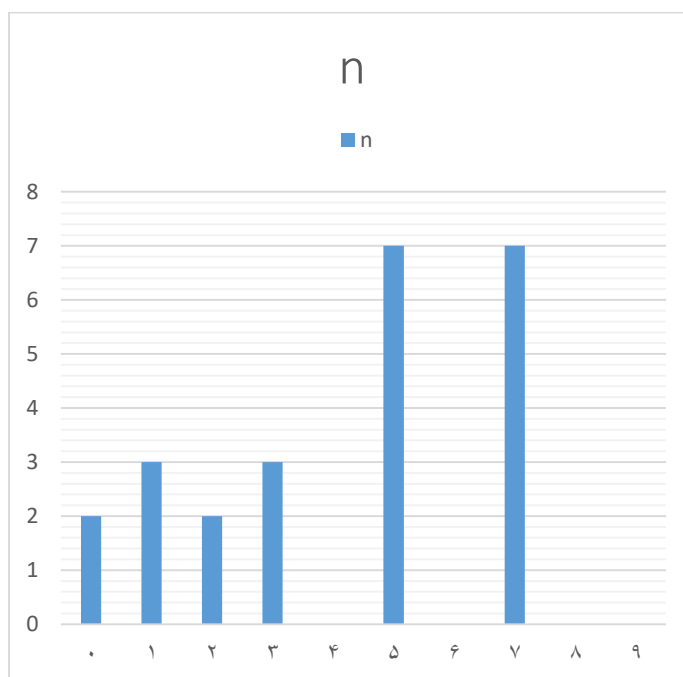
$$S_k = T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_j) = \frac{L - 1}{n} \sum_{j=0}^k n_j$$

k	0	1	2	3	4	5	6	7	8	9
n_k	0	0	2	0	3	2	3	7	7	0
$\sum_{j=0}^k n_j$	0	0	2	2	5	7	10	17	24	24
$\sum_{j=0}^k \frac{n_j}{n}$	0	0	$\frac{2}{30}$	$\frac{2}{30}$	$\frac{5}{30}$	$\frac{7}{30}$	$\frac{10}{30}$	$\frac{17}{30}$	$\frac{24}{30}$	$\frac{24}{30}$
$\frac{L-1}{n} \sum_{j=0}^k n_j$	0	0	0.6	0.6	1.5	2.1	3	5.1	7.2	7.2
round	0	0	0	0	1	2	3	5	7	7

تصویر جدید:

5	5	7	7	7	7
0	0	1	1	1	7
5	0	2	2	0	7
7	0	3	7	7	5
7	5	3	3	5	5

هیستوگرام تصویر جدید:



منبع: صحبت های استاد در جلسه ی سوم کلاس

سوال دوم:

(الف)

تصویر زیر مربوط به کد پیاده سازی تابع `hist_equ` می باشد که در آن ابتدا هیستوگرام تصویر ورودی با دو حلقه ی تو در تو روی پیکسل های تصویر و با توجه به رنگ ها محاسبه میشود. سپس `cdf` تصویر را حساب میکنیم. آن گاه پیکسل های تصویر خروجی که کپی از تصویر اولیه هستند با توجه به `cdf` تغییر میدهیم.

```

jupyter Q2 Last Checkpoint: a minute ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Python 3 (ipykernel)

In [35]: def hist_equ(image):
...
    input:
    image (ndarray): input image
    output:
    output_image (ndarray): enhanced image
    ...

    #####
    # Your code
    # Start
    output_image = image.copy()
    x,y = image.shape
    pixels_num = x*y
    color_num = 256
    image_histogram = np.zeros(color_num)
    ex_sum = np.zeros(color_num)
    image_cdf[None]*color_num
    for i in range(x):
        for j in range(y):
            image_histogram[image[i, j]] += 1

    for i in range(color_num):
        ex_sum[i] = image_histogram[i]
        if i > 0:
            ex_sum[i] += ex_sum[i - 1]
        sum = 0

    for i in range(0,color_num):
        image_cdf[i] = round((color_num - 1)*(ex_sum[i]/pixels_num))
    for i in range(x):
        for j in range(y):
            output_image[i,j]=image_cdf[image[i,j]]

    # End
    return output_image

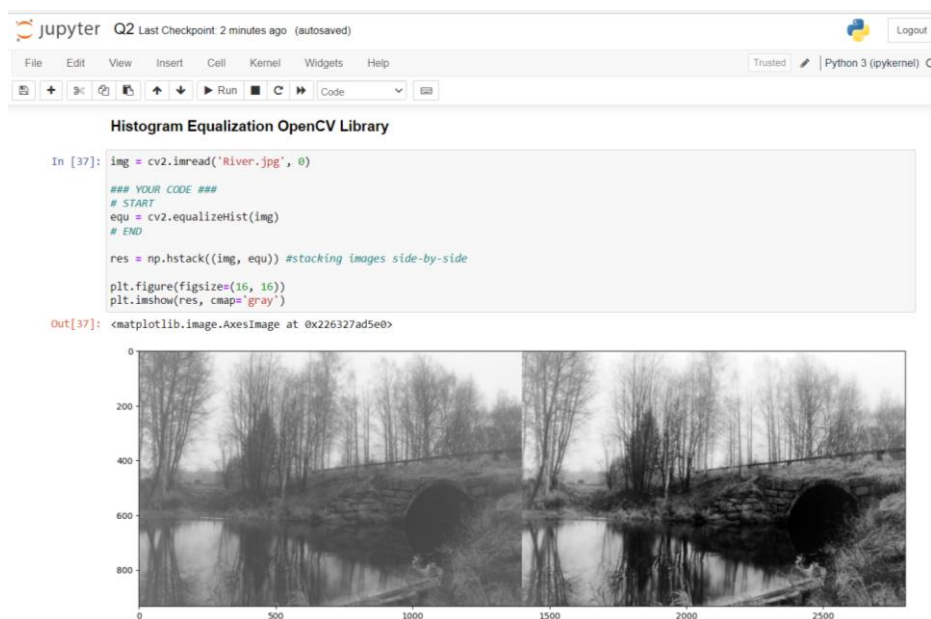
```

تصویر زیر نتیجه ی اعمال تابع بالا روی تصویر `river.jpg` می باشد.

همانطور که قابل مشاهده است کنتراست تصویر بالاتر رفته و رنگ ها نسبت به هم مشخص تر شده اند یعنی رنگ های تیره تر مشکی و رنگ های روشن تر به سفید تبدیل شدند و اجزای ریزتر مثل خطوط بین سنگ ها مشخص تر شده است و تصویر از تاری در آمده و طیف زنگی تصویر که قبلا فقط خاکستری بود گسترده تر شده و ماکزیمم و مینیمم آن تغییر کرده است.



تابع آماده ی `opencv` را هم روی همان تصویر اجرا کردیم و نتیجه به شکل زیر شد.



(ب)

لینک استفاده شده https://docs.opencv.org/4.x/d5/daf/tutorial_py_histogram_equalization.html

ابتدا باید تابع clahe را پیاده سازی کنیم.

Part 2

CLAHE OpenCV Library

```

In [38]: def CLAHE(image, gridSize, clip_limit):
        ...
        inputs:
        image (ndarray): input image
        gridSize (tuple): window size
        clip_limit (int): threshold for contrast limiting
        output:
        output_image (ndarray): improved image
        ...

        #####
        # Your code
        # Start
        clahe = cv2.createCLAHE(clipLimit=clip_limit, tileGridSize=gridSize)
        output_image = clahe.apply(img)
        # End

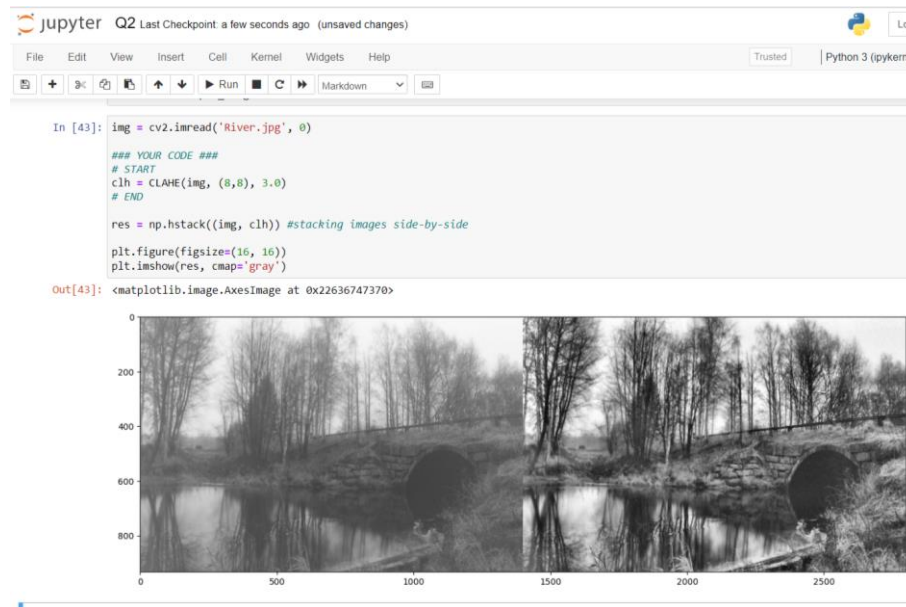
        return output_image

```

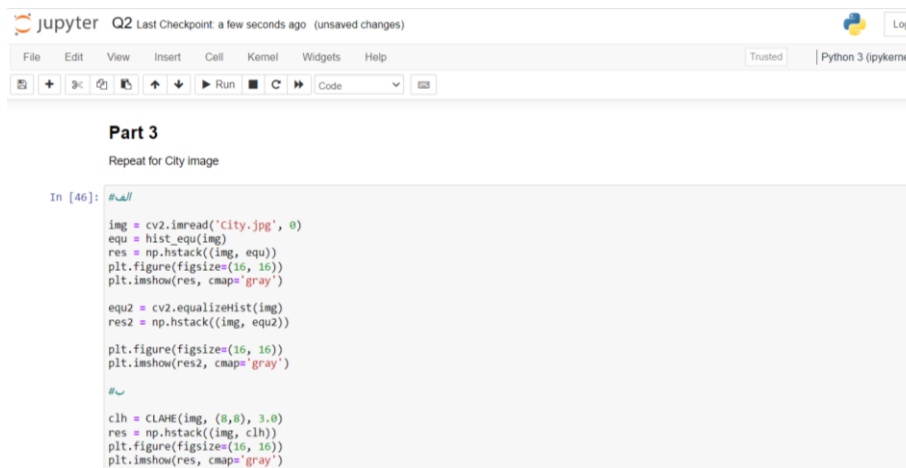
نتیجه:

در روش قبلی یعنی متعادل سازی هیستوگرام اطلاعات محلی لحاظ نشده بود یعنی X و Y پیکسل تاثیری نداشتند و در ارتقای تصویر میزان شدت روشنایی همسایه های یک پیکسل تاثیری روی بهبود پیکسل مورد نظر نداشت اما در روش کلاسه هم این نکات رعایت خواهد شد و هم میتوانیم میزان تقویت کنتراست را در نواحی مختلف کنترل کنیم.

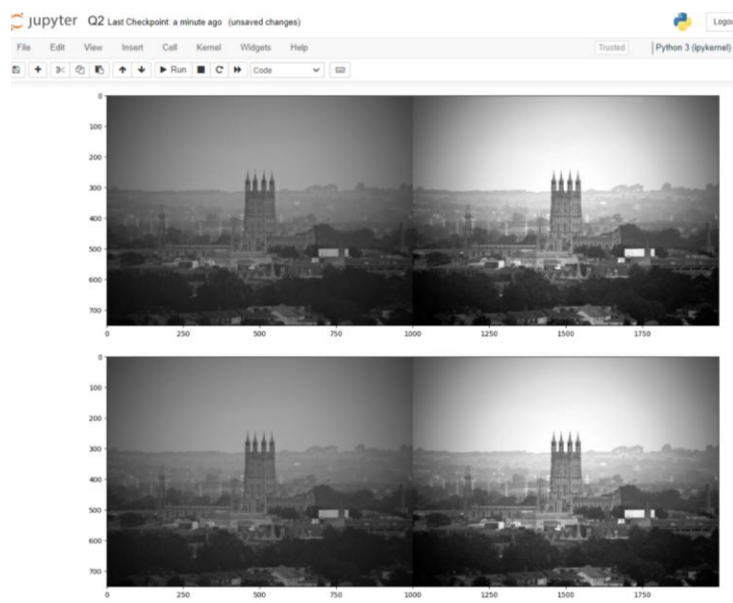
اگر تصویر حاصل از این بخش را با بخش قبلی مقایسه کنیم میبینیم در بخش های تیره تر در روش کلاسه ما تصویر را روشن تر و جزئیات را با دقت تر میبینیم نسبت به روش متعادل سازی که کاملاً در بخش تیره ی زیر پل رنگ های تیره شدت یافته اند و جزئیات پنهان شده اند.



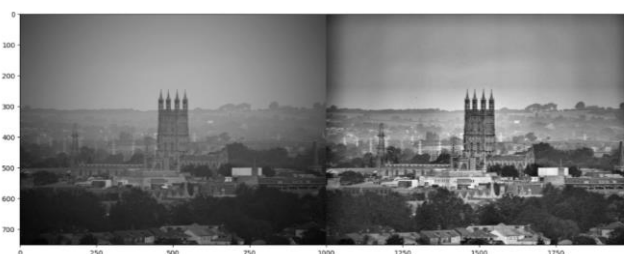
پ) تکرار قسمت الف و ب برای تصویر City.jpg



نتایج قسمت اول:



نتایج قسمت دوم:



(ت)

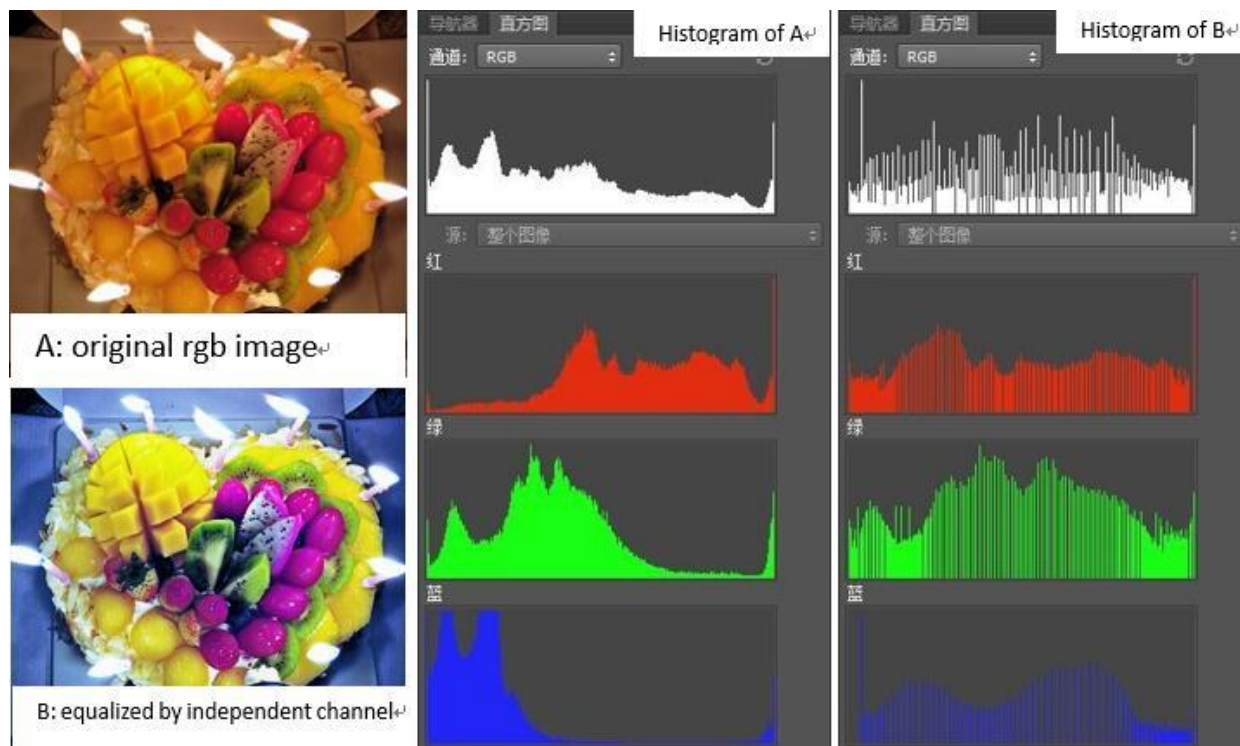
متعادل سازی هیستوگرام برای افزایش کنتراست تصاویر در مقیاس خاکستری میباشد که در تصاویر رنگی معمولاً در کانال های جداگانه اعمال میشود. در نظر نگرفتن ارتباط کانال R، G و B، اما فرآیند به ترتیب تصویر را مخدوش می کند. اعمال روش مشابه بر روی اجزای قرمز، سبز و آبی یک تصویر RGB ممکن است تغییرات چشمگیری در تعادل رنگ تصویر ایجاد کند زیرا توزیع نسبی کانال های رنگ در نتیجه اعمال الگوریتم تغییر می کند. البته این سریع ترین راه است و ممکن است برای بعضی جلوه های غیرواقعی مناسب باشد اما برای بهبود کنتراست تصاویر رنگی به طور کلی مناسب نیست.

نمونه از سایت: <https://hypjudy.github.io/2017/03/19/dip-histogram-equalization/>

همانطور که در نمونه ی زیر میبینید ابتدا هیستوگرام هررنگ بدست آمده و سپس متعادل شده است. در تصویر بدست آمده نقاط قرمز رنگ به بنفش تبدیل شدند چون تغییرات زیادی در هیستوگرام رنگ آبی بعد از متعادل سازی به وجود آمد و تعداد نقاط با شدت روشنایی بالای آبی بیشتر شد. این موضوع حتی در حاشیه ی تصویر هم قابل مشاهده است.

هیستوگرام کانال R، G و B کاملاً متفاوت است. وقتی به ترتیب تساوی مستقل را روی آنها اعمال می کنیم، B به دست می آید. شدت ها در هیستوگرام بهتر توزیع شده اند اما رنگ B از تعادل خارج شده است. این به دلیل تغییر توزیع نسبی کانال های رنگ است. به عنوان مثال،

اجزای سبز A بیشتر در سطوح پایین/ شدت های کوچک توزیع می شوند در حالی که اجزای سبز B در کل سطوح بسیار یکنواخت تر توزیع می شوند. این باعث می شود B آبی تر به نظر برسد.



اما راه حل مناسب تر:

(۱) متعادل سازی هیستوگرام بر اساس مقدار متوسط کانال رنگ:

آنچه ما می خواهیم یک هیستوگرام RGB با توزیع بهتر است اما نه همه ی هیستوگرام های شامل کانال های رنگی. بنابراین هنگام محاسبه هیستوگرام می توانیم سه کانال را به طور کلی در نظر بگیریم. تنها تفاوت بین این روش و روش قبلی این است که این روش از هیستوگرام میانگین هیستوگرام سه کانال استفاده می کند.

این روش ارتباط کانال های R، G و B را در نظر می گیرد و نتایج بهتری به دست می آورد. در اینجا یک مثال به صورت زیر داریم:

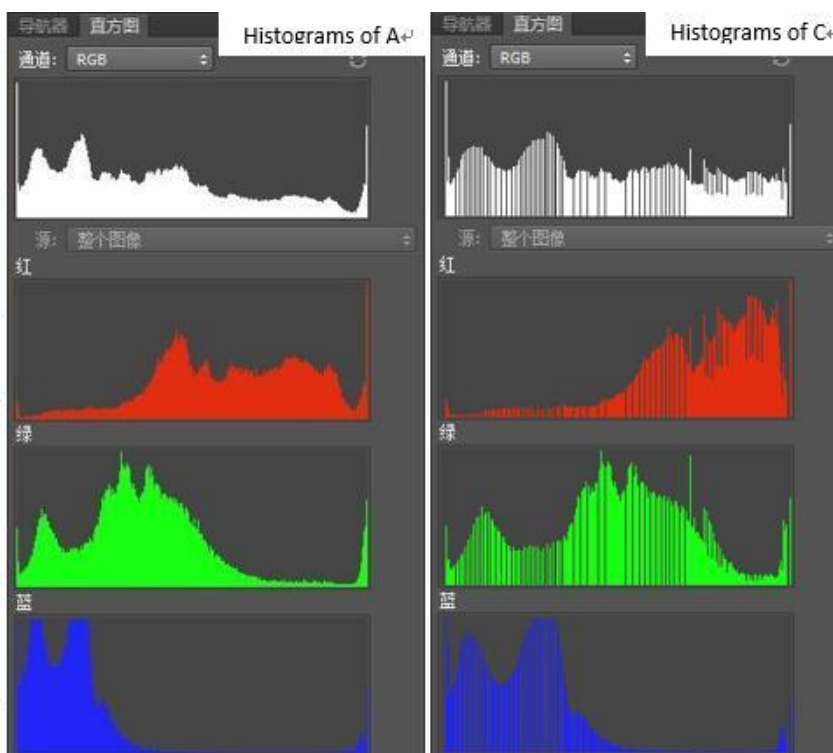
هیستوگرام RGB بهتر توزیع می شود و کنتراست افزایش می یابد. هیستوگرام کانال های R، G و B نیز گسترش یافته است اما همچنان توزیع نسبی اصلی را حفظ می کند.



A: original rgb image



C: equalized by average histogram



۲) متعادل سازی مولفه های شدت بر اساس فضای رنگی HSI:

تصویر ابتدا به فضای رنگی HSI تبدیل می شود، سپس روش مقیاس خاکستری را بدون تغییر در رنگ و اشباع تصویر روی درخشندگی اعمال می کند.

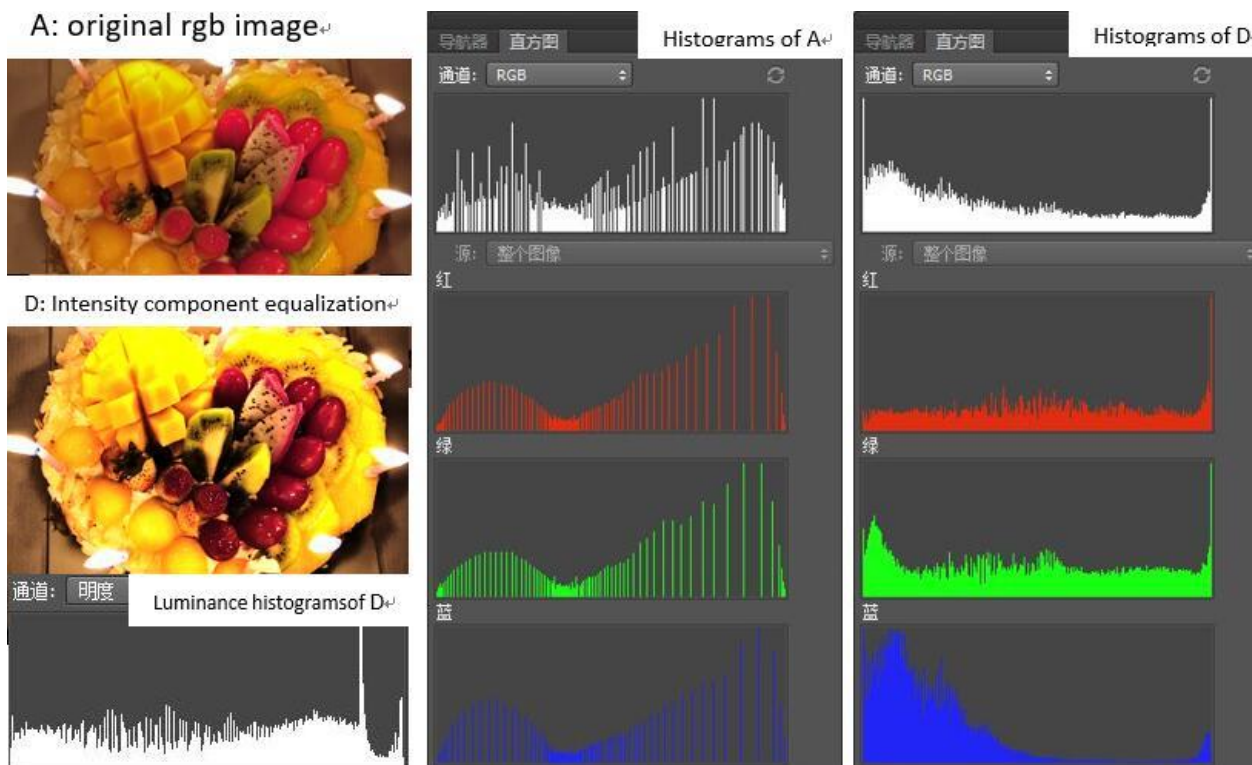
فرمولی استفاده شده از پردازش تصویر دیجیتال (ویرایش سوم): Rafael C. Gonzalez, Richard E. Woods, بخش 6.2.3 مدل رنگی HSI. همان فرمول را میتوان از PPT Yao Wang برای مرجع سریع پیدا کرد.

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases} \quad \text{with } \theta = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R-G) + (R-B)]}{\left[(R-G)^2 + (R-B)(G-B) \right]^{\frac{1}{2}}} \right\}$$

$$S = 1 - \frac{3}{(R+G+B)} [\min(R, G, B)]$$

$$I = \frac{1}{3} [R+G+B]$$

این روش اغلب نتایج خوبی دارد، اما گاهی اوقات ممکن است اینطور نباشد. در اینجا یک مثال وجود دارد:



در تصویر D، انبه ها خیلی روشن و توت فرنگی ها خیلی تیره هستند که جزئیات را پنهان می کند. این به دلیل توزیع نابرابر هیستوگرام RGB ایجاد می شود زیرا یکسان سازی در کانال روشنایی فضای رنگی HSI است.

سوال سوم:

الف) کد مورد نیاز برای این قسمت در زیر آمده است که در آن از تابع `match_histograms` استفاده شده که آرگومان دوم آن تصویر اولیه ایست که باید هیستوگرامش محاسبه شود و آرگومان اول آن تصویری است که باید هیستوگرام بدست آمده روی آن اعمال شود. و آرگومان سوم هم برای کانال تصویر است که در سوال خواسته بود رنگی باشد.

تصویر سمت چپ تصویر اصلی بدون تغییر است و تصویر سمت راست بعد از اعمال متعادل سازی هیستوگرام است.



(ب) استفاده از لینک: https://github.com/mapbox/rio-hist/blob/master/docs/notebooks/Example_LCH.ipynb

خودم تونستم کدی که برای تصویر خاکستری استفاده میشه رو پیاده سازی کنم اما برای تصویر با ۳ کانال از لینک بالا استفاده کردم. همانطور که قابل مشاهده هست تصویر حاصل با تصویر نتیجه ی تابع آماده متفاوت شده است. به نظر می آید تصویر حاصل در قسمت قبل یکنواخت تر است مثلا رنگ قهوه ای در برخی نقاط آن شدت یافته است اما این شدت یافتن باعث نشده اجزای رنگ قهوه ای جداگانه مشخص باشند اما در تصویر زیر رنگ بعضی قسمت ها به طور تجزیه شده بین قرمز و آبی شدت یافته است. مثلا بعضی نقاط آبی تر دیده میشود و بعضی نقاط قرمز تر نسبت به تصویر اصلی.



(پ)

تکرار قسمت الف:



تکرار قسمت ب)

