به نام خدا

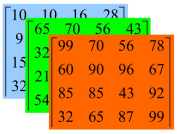
تکلیف دوم پردازش تصویر

ماهان فلاح پور

۸۱۰۸۰۲۰۸۳

Q1:

RGB ⟹ برای اینکه هر کانال RGB رو با رنگ خودش نشون بدیم
کافیه مقدار دیگر کانال ها را صفر کنیم



HSI Color model:

Hue: Dominant color ⟹ خود رنگ رو نشون میده به صورت زاویه θ تا 360 درجه تعریف میشه

Saturation: Relative purity (inversely proportional to amount of white light added) ⟹ غلظت یا خلوص رنگ رو نشون میده هرچی بیشتر باشه رنگ جیغ تره و هرچی کمتر باشه به سمت خاکستری میره

Intensity: Brightness ⟹ میزان روشنایی رو مستقل از رنگ نشون میده

$$I = \frac{1}{3}(r+g+b)$$

$$S = 1 - \frac{3}{r+g+b} \times \min(r,g,b)$$

$$\theta = \cos^{-1}\left(\frac{\frac{1}{2}((r-g)+(r-b))}{\sqrt{(r-g)^2+(r-b)(g-b)}}\right) \quad \begin{cases} H=\theta & b \le g \\ H=360^\circ-\theta & b>g \end{cases}$$

$$\frac{H}{2\pi} \Rightarrow [0,1]$$

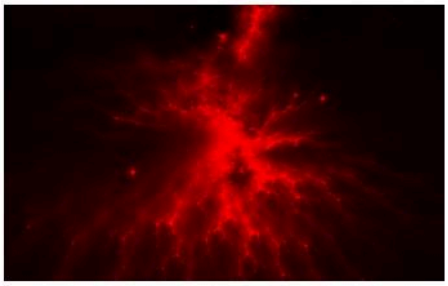
$$H = \begin{cases} \theta & \text{if } B \le G \\ 360-\theta & \text{if } B > G \end{cases}$$

$$\theta = \cos^{-1}\left\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{1/2}}\right\}$$

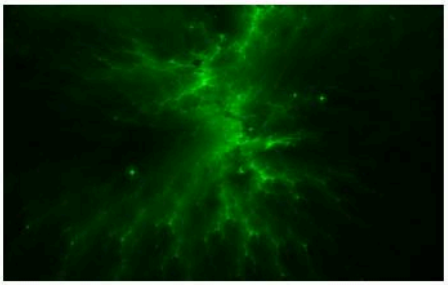
$$S = 1 - \frac{3}{R+G+B}$$

$$I = \frac{1}{3}(R+G+B)$$

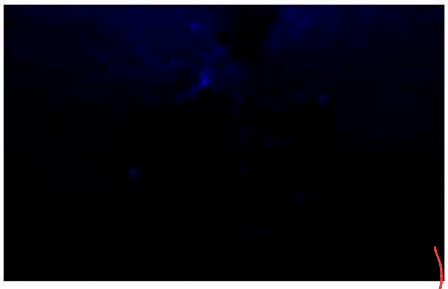
Red Channel



Green Channel

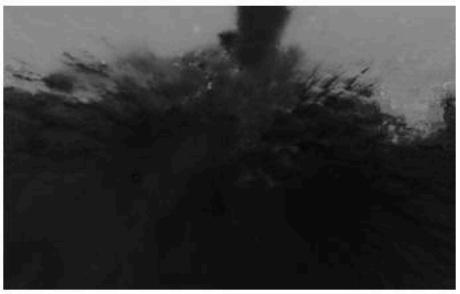


Blue Channel

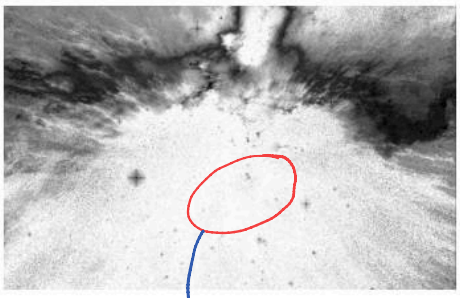


به دلیل کم بودن رنگ آبی در تصویر اصلی
این تصویر تاریک تره

Hue



Saturation



رنگ تو خالص ترین
حالتشه

Intensity

فقط همه جای عکس یقدر نور داره اصلاً کاری به رنگ
نداریم

*این سئوال از ۲ روش حل شده در ادامه به ترتیب هر ۲ روش را توضیح خواهم داد

Q2:

روش اول:

گام اول ⟸ ابتدا باید شدت روشنایی (معادل خاکستری) هر سه عکس مرجع را پیدا کنیم

$$I = 0.299R + 0.587G + 0.114B$$

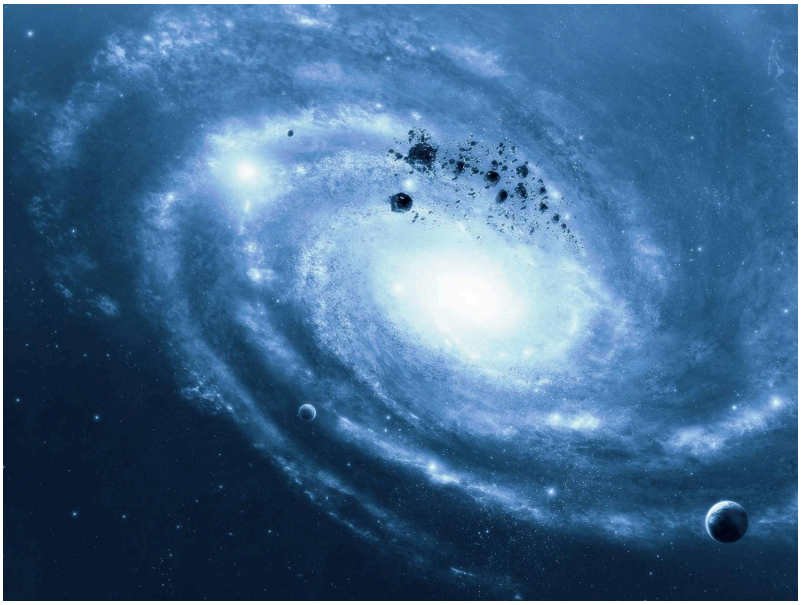
گام دوم ⟸ حالا باید به بدول بسازیم. برای هر سطح روشنایی i (از 0 تا 255) تمام پیکسل هایی از سه عکس مرجع که روشنایشون برابر i هست رو پیدا کنیم و میانگین مقادیر رنگی شون رو حساب می‌کنیم:

$$\underbrace{C(i)}_{\text{میانگین رنگ شامل R,G,B برای روشنایی } i} = \frac{1}{\underbrace{N_i}_{\text{تعداد پیکسل هایی که روشنایش } i \text{ بوده}}} \sum_{P \in Refs,\, I(P)=i} color(P)$$

گام سوم ⟸ ممکنه تو بعضی عکس های مرجع سطوح روشنایی اصلاً وجود نداشته باشد و $N_i=0$ بشه. در این حالت رنگ نزدیک‌ترین سطح روشنایی موجود رو بهش اختصاص می‌دیم تا بدولمون هیچ جای خالی‌ای نداشته باشد

گام آخر ⟸ درنهایت روی تک تک پیکسل های عکس خاکستری هدف (با شدت روشنایی I_t) راه می‌رویم و رنگ متناظرش رو از بدولی که ساختیم جایگذاری می‌کنیم:

$$output(x,y) = C(I_t(x,y))$$



روش دوم

در این روش من عکس سیاه و سفید رو به عنوان کانال روشنایی (I) در نظر گرفتم
بعد از روی عکس های رنگی مربع میدیم هر سطح معمولاً به رنگ (H) و به غلظتی (S) داره. و در نهایت
این 3 تا کانال (I و S و H) رو با هم ترکیب می‌کنیم و با استفاده از فرمول هایی که شرح میدم برش
می‌گردونم به فضای RGB تا عکس رنگی نهایی تولید بشه

* در کل 3 مرحله داریم:

الف) تبدیل مراجع از RGB به HSI

ب) نگاشت رنگ Mapping: برای هر مقدار از $I \in [0,255]$، میانگین H و S مربوطه رو از عکس های مربع پیدا می‌کنیم

$$H_{map}[I] = mean(H_{ref} \mid I_{ref} = I)$$

$$S_{map}[I] = mean(S_{ref} \mid I_{ref} = I)$$

2) تبدیل عکس هدف از HSI به RGB:

RG sector: $0 \le H < 120$

$$R = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right]$$

$$B = I(1-S)$$

$$G = 1-(R+B)$$

GB sector: $120 \le H < 240$

$$H = H - 120$$

$$R = I(1-S)$$

$$G = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right]$$

$$B = 1-(R+G)$$

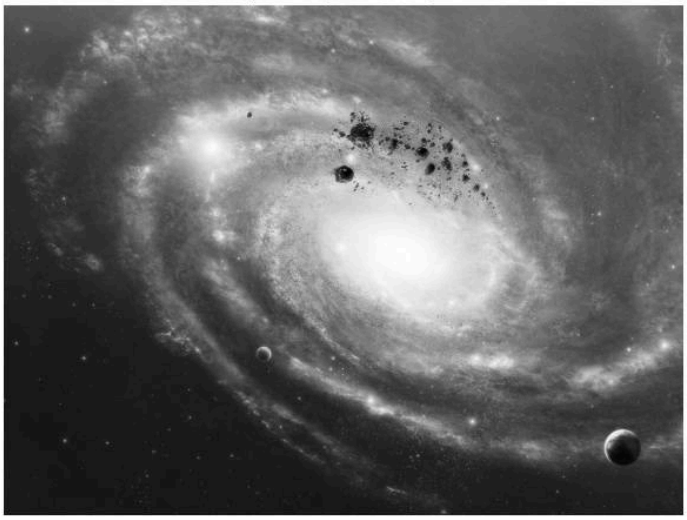
BR sector: $240 \le H \le 360$

$$H = H - 240$$

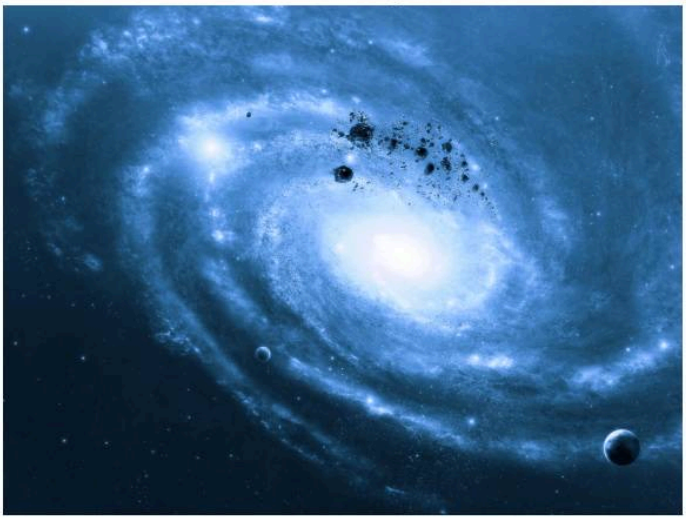
$$B = I\left[1 + \frac{S\cos H}{\cos(60^\circ - H)}\right]$$

$$G = I(1-S) \qquad R = 1-(G+B)$$

Input Grayscale Target



Colorized Output



Q3: معیار PSNR براساس خطای میانگین مربعات بین 2 تصویر محاسبه می‌شود. برای 2 تصویر RGB با ابعاد H×W×3، ابتدا تفاوت مقدار هر پیکسل در هر کانال محاسبه شده و به توان دو می‌رسد

$$error^2(i,j,c) = \left(I_{original}(i,j,c) - I_{distorted}(i,j,c)\right)^2$$

سپس، مقدار MSE برای تمام پیکسل‌ها و کانال‌ها ($N = H \times W \times 3$) به شکل زیر محاسبه می‌شه

$$MSE = \frac{1}{N}\sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{c=1}^{3} error^2(i,j,c)$$

درنهایت، برای تصاویر 8 بیتی که حداکثر مقدار پیکسل در آن $MAX_I = 255$ است مقدار PSNR برحسب dB این رابطه به دست می‌آید:

$$PSNR = 10\log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

$$PSNR = 10\log_{10}\left(\frac{L^2}{\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left|\hat{f}(x,y) - f(x,y)\right|^2}\right)$$

* هرچی مقدار PSNR بیشتر ← خطای بین 2 تصویر کمتر
* برای جلوگیری از سرریز در هنگام به توان 2 رساندن مقادیر، نوع داده تصویر باید به میز شناور تبدیل شود
* اگر 2 تصویر یکسان باشد، مقدار خطای MSE برابر صفر و PSNR تو تئوری به سمت بی‌نهایت میل می‌کند در کد به عنوان یک حالت استثنا با بازگرداندن مقدار بی‌نهایت مدیریت می‌شه

```
Manual PSNR: 18.5530 dB
CV2 PSNR:    18.5530 dB
```

⟹ محاسبه دستی بسیار دقیق بوده

Q4: برای حذف نویزهای نقطه ای از یک پنجره با ابعاد $K \times K$ استفاده می‌کنیم

که روی تمام پیکسل های تصویر حرکت می‌کند. اگر I تصویر نویزدار و W

پنجره همسایگی باشد، مقدار پیکسل جدید در تصویر بازسازی شده $\hat{I}$

از محاسبه میانه مقادیر پیکسل های درون پنجره به دست می‌آید

$$\hat{I}(x,y,c) = \text{median}\{I(x+i, y+j, c) \mid (i,j) \in W\}$$

نشان دهنده کانال رنگی RGB است. پس از اعمال این فیلتر و بازسازی تصویر، مقدار PSNR

با استفاده از همان فرمول سؤال قبلی ارزیابی میشه:

$$MSE = \frac{1}{H \times W \times 3} \sum_{x=1}^{H} \sum_{y=1}^{W} \sum_{c=1}^{3} \left(I_{original}(x,y,c) - \hat{I}(x,y,c)\right)^2$$

$$PSNR = 10 \log_{10}\left(\frac{255^2}{MSE}\right)$$

* تشخیص من از شکل 2.jpg به خاطر نقاط سفید ریز پراکنده نویز غالب نمکی بود

* من هر چه در عکس های قرار داده شده چسبوندم عکس بدون نویزی که میخوام مربع قرار بدم

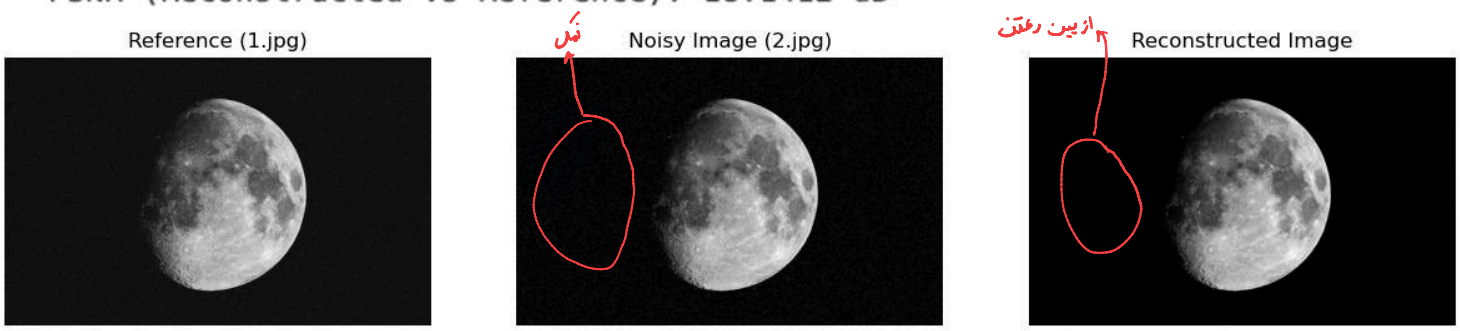
پیدا نکردم و مجبور شدم تصویر 1.jpg که نویز کمتری داشت را مربع قرار بدم

* به منظور افزایش سرعت برای تصاویر با رزولوشن بالا و جلوگیری از افت عملکرد پیاده سازی

فیلتر میانه به جای استفاده از حلقه های تودرتو (دفعه اول این کار را کردم فاجعه بود!!!)

با استفاده از تکنیک Vectorization و عملیات ماتریسی در کتابخانه numpy استفاده شد

```
PSNR (Noisy vs Reference): 15.2282 dB
PSNR (Reconstructed vs Reference): 18.1412 dB
```



Q5:

$$D = \begin{bmatrix} 195 & -23 & 12 & 2 \\ -19 & 11 & -4 & 0 \\ 8 & -2 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \Rightarrow D_q(i,j) = \text{round}\left(\frac{D(i,j)}{Q(i,j)}\right)$$

$$Q = \begin{bmatrix} 10 & 12 & 14 & 18 \\ 12 & 14 & 18 & 24 \\ 14 & 18 & 24 & 32 \\ 18 & 24 & 32 & 40 \end{bmatrix}$$

این ماتریس از طریق الگوی زیگ زاگ به یک دنباله یک بعدی تبدیل میشه

در نهایت با RLC به شکل جفت های فشرده می شود

پاسخ به سؤال بخش 4 :

1. در ماتریس D_q به دست آمده 10 درایه از مجموع 16 درایه برابر با صفر هستند این موضوع نشان می دهد که فرایند فشرده سازی JPEG با حذف جزئیات فرکانس بالا داده ها را کاهش می دهد و تنها اطلاعات اصلی را نگه می دارد

2. اگر تمام عناصر ماتریس کوانتیزاسیون Q دو برابر شوند :

* نسبت فشرده سازی افزایش پیدا می کند چون مخرج کسرها بزرگتر شده و مقادیر بیشتری از ماتریس D به سمت صفر گرد خواهند شد

* کیفیت تصویر کاهش می یابد زیرا با افزایش شدت کوانتیزاسیون خطای گرد کردن بیشتر شده و اطلاعات و جزئیات بیشتری از تصویر اصلی به طور غیر قابل بازگشت از بین می رود

```
Quantized Matrix D_q:
[[15 -2  1  0]
 [-2  1  0  0]
 [ 1  0  0  0]
 [ 0  0  0  0]]

Zig-Zag Sequence:
[15, -2, -2, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Run-Length Coding (RLC) Pairs:
[(0, 15), (0, -2), (0, -2), (0, 1), (0, 1), (0, 1)]
```