

تصاویر رنگی

علی نصیری سروی

اطلاعات گزارش	چکیده
تاریخ: 99/02/26	
واژگان کلیدی: تصاویر رنگی چندی سازی کاهش رنگ خوشه بندی	در این تمرین در ابتدا به معرفی رنگ پرداخته سپس فضاهاى رنگى مختلف را معرفی میکنیم و تبدیلات آنها به همدیگر را مورد بررسی قرار میدهیم. در مرحله بعد چندی سازی در فضای رنگی RGB را مورد بررسی قرار میدهیم. در نهایت تلاش میکنیم تعداد رنگ های تصویر را به کمک خوشه بندی کاهش دهیم.

1-مقدمه

رنگ یک توصیفگر بسیار قوی است که تشخیص یک شی و استخراج آن از صحنه را ساده میکند. چشم انسان به طور میانگین حدود 24 سطح خاکستری متمایز را می تواند تشخیص دهد. در حالی که در فضای رنگی هزاران رنگ قابل تمایز برای چشم وجود دارد و همین باعث میشود تصاویر رنگی برای چشم انسان کیفیت بهتری داشته باشد.

2-شرح تکنیکال

تصاویر رنگی

در سال 1666 میلادی اسحاق نیوتون آزمایش معروف منشور خود را انجام داد. در این آزمایش وی نور مرئی را به منشوری تابانده و تجزیه این نور به طیف های رنگی مختلف را مشاهده نمود. نور مرئی طول موجی بین 400 تا 700 نانومتر دارد. این بازه ی موج بین موج های فرابنفش و فروسرخ قرار دارد.

چشم انسان دو نوع حسگر استوانه ای و مخروطی برای دیدن دارد. حسگرهای مخروطی کیفیت بیشتری دارند و میتوانند رنگ ها را تشخیص دهند. البته این حسگرها برای فعال شدن نیاز به نور دارند (به همین دلیل در تاریکی رنگ ها قابل تشخیص نیستند).

Saturation: میزان ترکیب نور سفید با یک فام(میزان

خلوص رنگ)

Intensity: شدت رنگ.

برای تبدیل رنگ از فضای RGB به HSI از فرمول های

زیر استفاده میکنیم:

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

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

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

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

5.1.2 معرفی سه فضای رنگی دیگر

فضای رنگی YUV:

این فضای رنگی یک بازنمایی جدید از فضای RGB برای انتقال بهینه می باشد.

این فضا از luminance(Y) و دو بخش رنگی (U,V) تشکیل شده است.

مزیت اصلی مدل YUV در پردازش تصویر جدا کردن اطلاعات رنگی و luminance است. دلیل آن این است که جز luminance یک تصویر بدون اثرگذاری بر جز رنگی میتواند مورد پردازش قرار گیرد.

تبدیل از فضای RGB به YUV به صورت زیر است:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.14713 & -0.28886 & 0.436 \\ 0.615 & -0.51499 & -0.10001 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

برای تبدیل از YUV به RGB داریم:

به طور کلی سه رنگ قرمز آبی و سبز را رنگ های اصلی می نامند و دلیل آن سلول های مخروطی می باشد.

حدودا 2 درصد سلول های مخروطی را سلولهای آبی، 33 درصد را سلولهای سبز و 65 درصد را سلول های قرمز تشکیل می دهند.

مشخصاتی که به طور عمده برای شناسایی یک رنگ از دیگری استفاده می شوند عبارت اند از:

Brightness, Saturation, Hue.

Hue یا فام: نشان دهنده ی طول موج رنگ غالب که توسط بیننده دریافت می شود.

Saturation: میزان ترکیب نور سفید با یک فام(میزان خلوص رنگ)
Brightness: شدت روشنایی.

ترکیب hue و saturation با هم کروماتیسیته chromaticity می گویند.

چشم انسان در نهایت آن رنگی که دریافت میکند به صورت RGB است اما چیزی که توسط مغز ما درک میشود با ویژگی های کروماتیسیته و شدت روشنایی بهتر قابل بیان است.

5.1 فضاهای رنگی

فضاهای رنگی، یک مدل ریاضیاتی انتزاعی می باشند که رنج رنگ ها که معمولا 3 یا 4 تا می باشند، را توصیف می کند.

5.1.1 فضای رنگی HSI

این فضا بر اساس نحوه درک رنگ توسط بینایی انسان ایجاد شده است. فضای رنگی HSI هر رنگ را با سه جز نشان می دهد:

Hue یا فام: نشان دهنده ی طول موج رنگ غالب که توسط بیننده دریافت می شود.

مرکزی، شمالی و ژاپن مورد استفاده است.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.13983 \\ 1 & -0.39465 & -0.5806 \\ 1 & 2.03211 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix}$$

در این فضا Y اطلاعات luminance بوده و دو جز I و Q اطلاعات chrominance تصویر را دارا می باشند. این مدل رنگی سعی میکند از نحوه بازخورد چشم انسان نسبت به رنگ استفاده کند.

تبدیل از فضای RGB به YIQ به صورت زیر است:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.5959 & -0.2746 & -0.3213 \\ 0.2115 & -0.5227 & 0.3112 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

برای تبدیل از YIQ به RGB داریم:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0.956 & 0.619 \\ 1 & -0.272 & -0.647 \\ 1 & -1.106 & 1.703 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

فضای رنگی YUV معمولا برای پردازش ویدیو/تصاویر رنگی و انتشار داده ها در تلویزیون های آنالوگ به کار می رود.

فضای رنگی HCL

مشکلی که فضاها ی رنگی HSV و HSI دارند، این است که بر اساس saturation متغیر عمل می کنند و این آن روشی نیست که چشم ما تصاویر را درک میکند. این مشکل به کمک فضای HCL حل می شود. در این فضا کنترل رنگ (hue)، میزان رنگی بودن (chroma) و روشنایی (luminance) توسط ما کنترل می شود.

برای تبدیل از RGB به HCL از فرمول های زیر استفاده می شود:

$$Q = e^{\alpha\gamma}$$

$$\alpha = \frac{1}{100} \times \frac{\min(R, G, B)}{\max(R, G, B)} ; \gamma = 3.$$

Q پارامتری است که شدت luminosity بین یک رنگ کامل اشباع شده و رنگ سفید را مشخص می کند. حال برای سه جز رنگی داریم :

$$H = \text{atan2}(G - B, R - G)$$

$$C = Q \times \frac{|R - G| + |G - B| + |B - R|}{3}$$

$$L = \frac{Q \times \max(R, G, B) + (1 - Q) \times \min(R, G, B)}{2}$$

فضای رنگی YIQ

YIQ یک فضای رنگی مورد استفاده توسط تلویزیون های با سیستم رنگی NTSC می باشد و معمولا در آمریکای

5.2 چندی سازی

فرآیند چندی سازی بر روی تصاویر رنگی باعث کاهش تعداد رنگ های متمایز در یک تصویر را کاهش می دهد. هدف این است که کاهش تعداد رنگ ها با کمترین کاهش کیفیت تصویر همراه باشد.

این فرآیند برای نمایش تصاویر با تعداد زیادی رنگ بر روی دستگاه هایی با تعداد رنگ محدود لازم است.

برای چندی سازی از یک quantizer استفاده می کنیم: اگر تابع f را داشته باشیم، نتیجه اعمال quantizer بر آن Q(f) می باشد.

اگر داشته باشیم:

$$\text{Decision Levels} = \{t_k, k = 0, \dots, L\}$$

$$\text{Reconstruction Levels} = \{r_k, k = 0, \dots, L - 1\}$$

$$\text{If } f \in [t_k, t_{k+1})$$

$$\text{Then } Q(f) = r_k$$

این که این سطح ها به چه صورت باشند، انواع چندی سازی های مختلف را تشکیل می دهد. همچنین اگر L سطح داشته باشیم، $\lceil \log_2 L \rceil$ بیت برای ذخیره سازی نیاز داریم.

5.2.1 چندی سازی یکنواخت و تعداد سطوح ثابت

فاصله یکنواختی بین سطح های تصمیم همجوار با سطح های بازسازی همجوار وجود دارد.

$$t_i - t_{i-1} = r_i - t_{i-1} = q$$

پارامترهایی که چندی سازی یکنواخت دارد به صورت زیر است:

L = تعداد سطوح

$$B = f_{\max} - f_{\min}$$

q = فاصله چندی سازی

$$Q = \frac{B}{L}$$

تابع چندی سازی یکنواخت به صورت زیر است:

$$Q(f) = \left\lfloor \frac{f - f_{\min}}{q} \right\rfloor \times q + \frac{q}{2} + f_{\min}$$

در تصاویر رنگی RGB برای این که کل تصویر را چندی سازی کنیم هر مولفه را جداگانه را به تنهایی چندی سازی می کنیم. در صورتی که بخواهیم رنگ تصویر عوض نشد باید در هر سه مولفه، از تعداد سطوح یکسان استفاده می کنیم.

5.2.2 چندی سازی یکنواخت با تعداد سطوح متفاوت

روش مانند بخش قبل می باشد با این تفاوت که مولفه آبی 4 سطح خاکستری (2 بیت) داشته و مولفه های قرمز و سبز 8 سطح خاکستری (3 بیت) دارند. در شرح نتایج به تاثیر چندی سازی با تعداد سطوح متفاوت می پردازیم.

5.2.3 کاهش تعداد رنگ

مسئله ای که با آن در هنگام چاپ، رنگ آمیزی و... مواجه هستیم محدود بودن تعداد رنگ ها است و اگر بخواهیم تصویری را چاپ کنیم باید تعداد رنگ های آن را کاهش دهیم.

برای این کار از الگوریتم kmeans برای خوشه بندی استفاده میکنیم. ورودی این الگوریتم داده ها و پارامتر k که تعداد خوشه ها است، می باشد. این الگوریتم در نهایت k مرکز خوشه پیدا کرده و هر داده را به خوشه ای که به آن نزدیک تر است دسته بندی میکند. الگوریتم kmeans:

1. ابتدا k میانگین که نماینده خوشه ها هستند را به صورت تصادفی مقدار دهی میکنیم.
2. سپس، این دو مرحله پایین را به تناوب چندین بار اجرا می کنیم تا میانگین ها به یک ثبات کافی برسند و یا مجموع واریانس های خوشه ها تغییر چندانی نکنند:
3. هر داده را به نزدیک ترین خوشه مربوطه اختصاص بده
4. مرکز جدید هر خوشه را میانگین داده های درون آن قرار بده.
5. در نهایت هر داده را به خوشه های نهایی اختصاص بده.

داده های مان رنگ های هر پیکسل می باشند. با اعمال kmeans به تعداد k خوشه میرسیم. که هر خوشه سعی میکند رنگ های نزدیک خودش را دسته بندی کند. با این کار رنگ های نزدیک به هم به یک خوشه اختصاص می یابند.

در نهایت مقدار رنگ هر پیکسل را با مقدار رنگ مرکز خوشه آن عوض میکنیم. تا تصویر کاهش یافته بدست آید.

3-شرح نتایج

5.1.1 فضای رنگی HSI

تصویر تست این بخش تصویر فلفل میباشد:



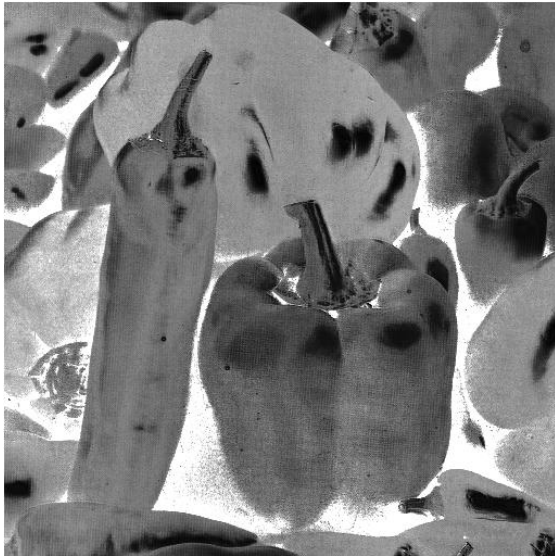
بخش hue آن به صورت زیر است:



بخش های سبز رنگ همانطور که مورد انتظار است نه کامل سیاه هستند و نه کامل سفید چون مقدار آنها نزدیک 0 یا 360 نیست. اما در مورد رنگ های قرمز اینطور نیست. برخی از آنها کامل سفید و برخی کامل سیاه هستند. همانطور که میدانیم در فضای hsi 0 درجه با 360 درجه از نظر بصری رنگ یکسانی خواهند داشت. یعنی رنگ قرمز میتواند مقدار نزدیک به صفر یا نزدیک به 360 داشته باشد که این یکی از مشکلات hsi در مورد

رنگ قرمز در نظر گرفته می شود.

بخش saturation آن به صورت زیر است:



همانطور که مورد انتظار است بخش های روشن تر (دارای مخلوط رنگ سفید بیشتر) در تصویر اصلی، مقدارشان در مولفه S کمتر بوده و سیاه تر می باشند. بخش های تاریک که رنگ سفید کمتری دارند، در مولفه S مقدار بیشتری داشته و سفید می باشند.

بخش intensity آن به صورت زیر است:

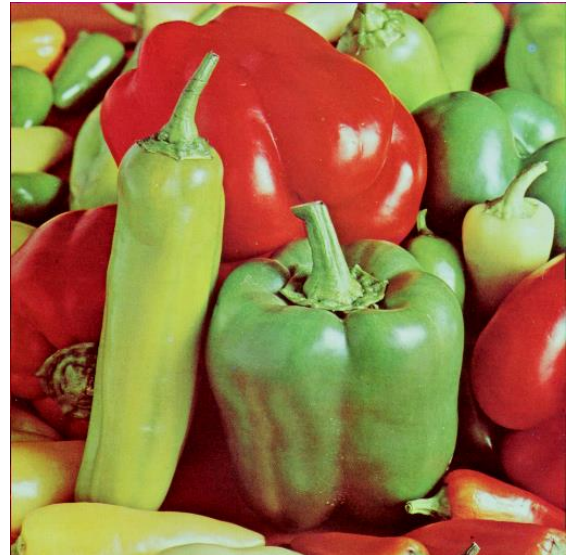


همانطور که مورد انتظار می بود، بخش های سفید تر، شدت بیشتر داشته و در مولفه I روشن تر بوده و بخش

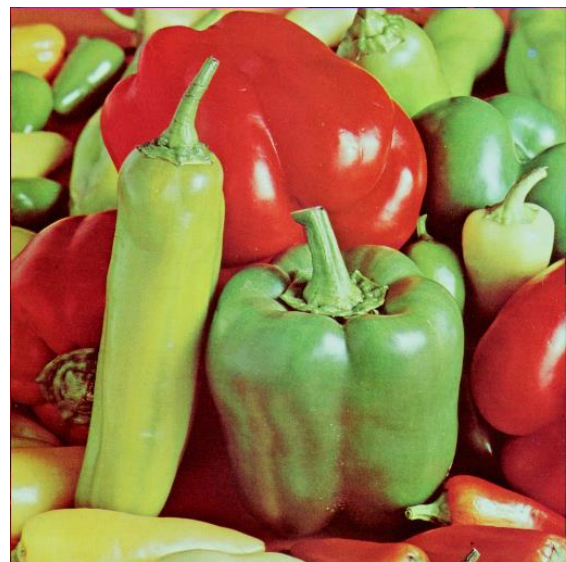
های سیاه تر شدت کمتری داشته و در مولفه I تیره تر می باشند.

5.2.1 چندی سازی یکنواخت و تعداد سطوح ثابت

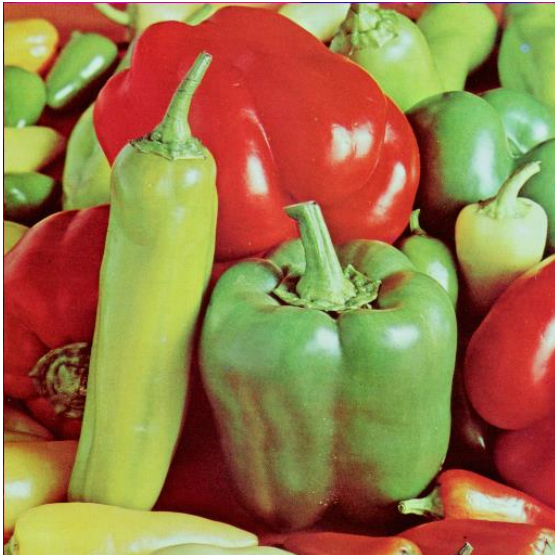
تصویر تست این بخش تصویر فلفل می باشد:



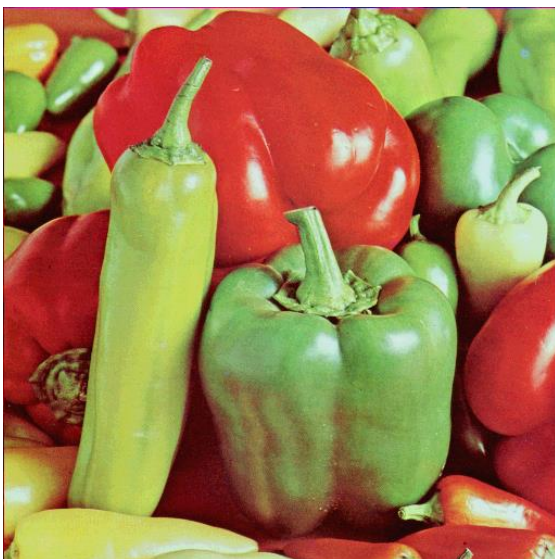
تصاویر به اعمال چندی سازی:



تصویر با اعمال چندی سازی و 64 سطح



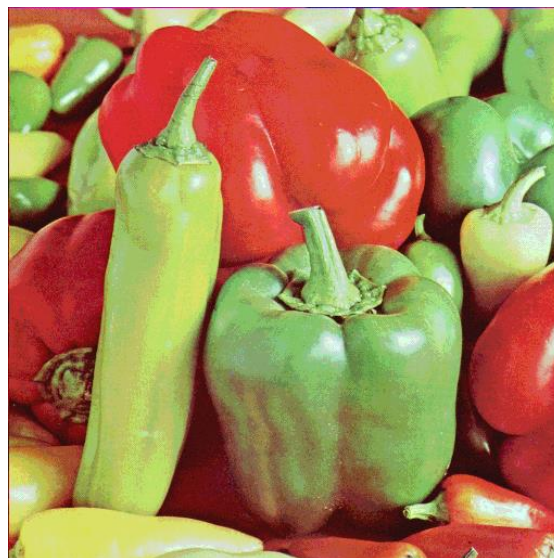
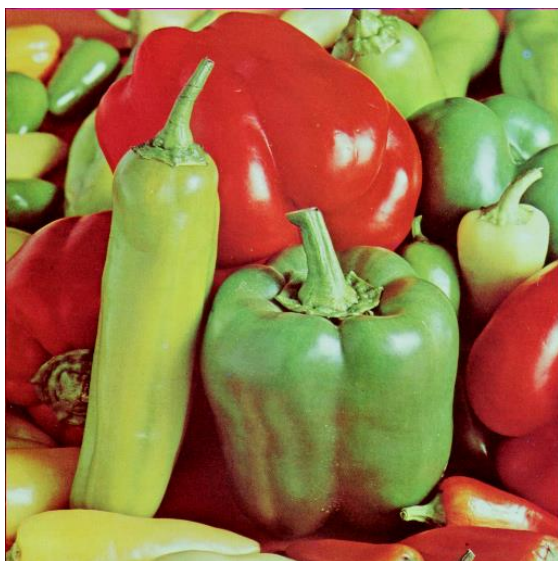
تصویر با اعمال چندی سازی و 32 سطح



تصویر با اعمال چندی سازی و 16 سطح

5.2.2 چندی سازی یکنواخت با تعداد سطوح متفاوت

تصویر تست این بخش تصویر فلفل می باشد:



تصویر با اعمال چندی سازی و 8 سطح

با چندی سازی با در نظر گرفتن 3 سطح برای مولفه های R,G و 2 سطح برای مولفه B به تصویر زیر خواهیم رسید:



تعداد سطوح	psnr	mse
64	39.48	7.32
32	35.16	19.81
16	29.37	75.09
8	23.43	294.92

نتایج بدست آمده

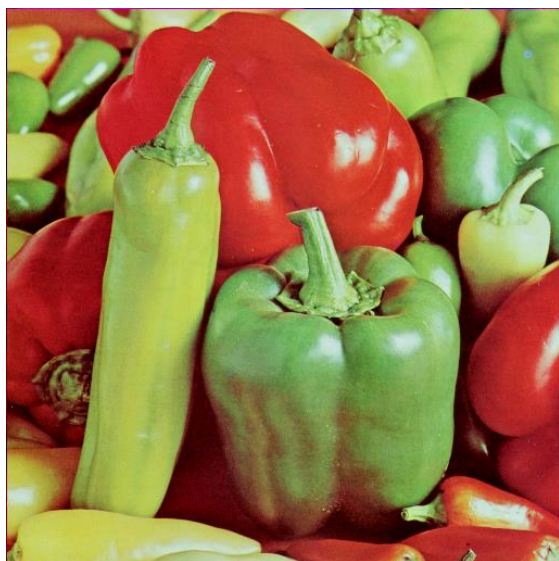
همانطور که قابل پیش بینی بود با کاهش تعداد سطوح mse افزایش یافته و نرخ سیگنال به نویز نیز افزایش می یابد.

دلیل آن هم کم تر بودن data loss زمانی که تعداد سطوح بیشتری داریم، می باشد.

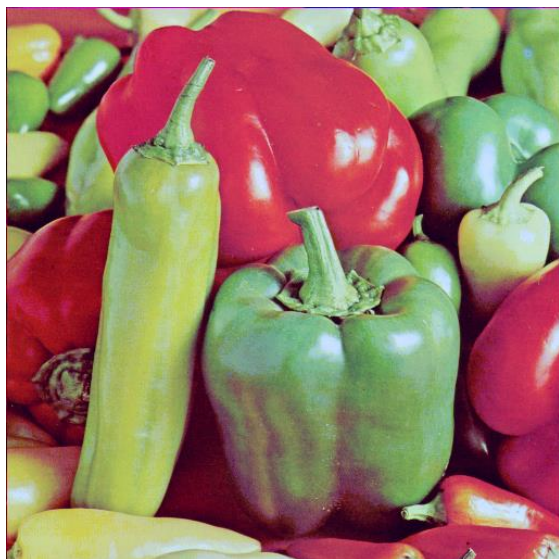
در حالت کلی کاهش تعداد بیت ها از 8 به 3 و 2 باعث افت کیفیت تصویر خواهد شد اما نکته مد نظر ما کاهش مولفه B به 2 بیت است. اتفاقی که در این حالت رخ می دهد این است که تاثیر مولفه B بر رنگ نهایی کم رنگ تر می شود. این کار تعداد رنگ هایی که میتوانیم داشته باشیم نصف می کند اما چون تصویر اصلی بیشتر شامل رنگ های قرمز و سبز هستند، اثر منفی زیادی نخواهد

گذاشت.

برای مثال زمانی که هر کدام از بیت های قرمز و سبز را 2 بیت قرار دهیم و دو مولفه دیگر 3 بیت داشته باشند به تصاویر زیر خواهیم رسید:



تعداد بیت های مولفه های R,G را 8 تا گرفته و 2 بیت برای مولفه B در نظر میگیریم. نتیجه به صورت زیر خواهد بود:



کیفیت رنگی تصویر به خوبی حفظ شده است و دلیل آن عدم وجود رنگ های دارای رنگ آبی در تصویر، میباشد.



2 بیت برای مولفه G



2 بیت برای مولفه R

همانطور که کاملاً مشخص است تاثیر بسیار زیادی بر نتیجه داشت.

5.2.3 کاهش تعداد رنگ

تصویر تست این بخش تصویر دختر می باشد:



با اعمال چندی سازی به کمک الگوریتم kmeans خواهیم داشت:



چندی سازی با 32 رنگ

چندی سازی با 16 رنگ



چندی سازی با 8 رنگ

تعداد خوشه ها	psnr	mse
32	28.71	87.33
16	26.18	156.40
8	23.35	300.24

همانطور که مورد انتظار هست هر چه تعداد رنگ ها کمتر باشد، data loss بیشتری خواهیم داشت.

4- کدها

5.2.1 چندی سازی یکنواخت و تعداد سطوح ثابت

```
img =  
imread('Homeworks\Images\5  
\Pepper.bmp');
```

```
L = 64;  
R =  
uniform_quantizer(img(:,:,  
1),L);  
G =  
uniform_quantizer(img(:,:,  
2),L);  
B =  
uniform_quantizer(img(:,:,  
3),L);  
x = img*0;  
x(:,:,1) = R;  
x(:,:,2) = G;  
x(:,:,3) = B;  
mse_error = immse(x,img);  
psnr_error = psnr(x,img);
```

```
imshow(img);  
figure  
imshow(x);  
imwrite(x,'x.png');
```

```
function output =  
uniform_quantizer(gray_img  
,L)
```

```
    %dynamic range  
    fmax =  
max(max(gray_img));  
    fmin =  
min(min(gray_img));  
    B = fmax - fmin;
```

```
    %quantization interval  
    q = B/L;
```

```
    [M,N] =size(gray_img);
```

5.1.1 فضای رنگی HSI

```
img =  
imread('Homeworks/Images/5  
/Pepper.bmp');
```

```
R = double(img (:,:,1));  
G = double(img (:,:,2));  
B = double(img (:,:,3));
```

```
tetha = acosd((0.5*((R-  
G)+(R-B)))/((R-G).^2+(R-  
B).*(G-B)).^0.5);
```

```
%H component
```

```
X = B-G;  
t = X<=0;  
h1 = tetha .*t;  
t = X>0;  
h2 = (360-tetha) .*t;  
H = h1+h2;  
H = H./360;
```

```
%S component
```

```
S = 1 -  
(3.*min(min(R,G),B)./(R+G+  
B));
```

```
%I component
```

```
I = (R+G+B)/3;
```

```
imshow(img);  
figure  
x= H;  
imshow(x);
```

```
imwrite(x,'x.png');
```

```

    fmax =
max(max(gray_img));
    fmin =
min(min(gray_img));
    B = fmax - fmin;

    %quantization interval
    q = B/L;

    [M,N] =size(gray_img);

output=zeros(M,N,'uint8');
    for r=1:M
        for c=1:N
            f =
gray_img(r,c);
            Qf = floor((f-
fmin)/q)*q+q/2+fmin;

output(r,c)=Qf;
        end
    end
end

```

5.2.3 کاهش تعداد رنگ

```

img =
imread('Homeworks\Images\5
\Girl.bmp');

imshow(img);
figure
L = 8;
out =
kmeans_quantizer(img,L);
imshow(out);
imwrite(out,'x.png');

mse_error =
immse(out,img);
psnr_error =
psnr(out,img);
function output =
kmeans_quantizer(img,clusters)

[M,N,dim] = size(img);

```

```

output=zeros(M,N,'uint8');
    for r=1:M
        for c=1:N
            f =
gray_img(r,c);
            Qf = floor((f-
fmin)/q)*q+q/2+fmin;
            output(r,c)=Qf;
        end
    end
end

```

5.2.2 چندی سازی یکنواخت با تعداد سطوح متفاوت

```

img =
imread('Homeworks\Images\5
\Pepper.bmp');

```

```

L = 8;
R =
uniform_quantizer(img(:,:,
1),L);
G =
uniform_quantizer(img(:,:,
2),L);
B =
uniform_quantizer(img(:,:,
3),L/2);
x = img*0;
x(:,:,1) = R;
x(:,:,2) = G;
x(:,:,3) = B;
mse_error = immse(x,img);
psnr_error = psnr(x,img);

```

```

imshow(img);
figure
imshow(x);
imwrite(x,'x.png');

```

```

function output =
uniform_quantizer(gray_img
,L)

```

%dynamic range

```

        samples =
zeros([M*N,dim], 'double');
        count = 1;
        for c=1:N
            for r=1:M

samples(count,:)=reshape(i
mg(r,c,:),[1,3]);
                count = count
+ 1;
            end
        end

        [idx,centers] =
kmeans(samples,clusters,'M
axIter',1000);
        centers =
uint8(centers);
        result =
centers(idx,:);
        result =
reshape(result,[M,N,dim]);
        output = result;
end

```