

باسمه تعالی

گزارش تکلیف سری سوم سیستم های چند رسانه ای – استاد سماوی

عرفان بهرامی – ۹۶۲۴۵۱۳

توضیحات سوال ۱

قسمت الف:

با استفاده از تابع نوشته شده در تکلیف قبلی `cdf` را به دست می آوریم.

```
3 [CDF, Histogram] = hist_cdf(Image);
```

نقطه ی اول را به دست آورده و برش اول را ایجاد می کنیم: T_0

```
5 Cut = CDF(256, 1)/2;
6 T0 = 0;
7 for i=1: 1: 256
8     if CDF(i, 1) >= Cut
9         T0 = i;
10        break;
11    end
12 end
```

به طور مشابه نقطه وسط قسمت های چپ و راست نقطه اولیه را که بدست آوردیم ، پیدا می کنیم: T_{01}

و T_{02}

```
14 Cut = CDF(T0, 1)/2;
15 T01 = 0;
16 for i=1: 1: T0
17     if CDF(i, 1) >= Cut
18         T01 = i;
19         break;
20     end
21 end
22
23 Cut = ((CDF(256, 1) - CDF(T0, 1))/2) + CDF(T0);
24 T02 = 0;
25 for i=T01+1: 1: 256
26     if CDF(i, 1) >= Cut
27         T02 = i;
28         break;
29     end
30 end
```

در نهایت بر روی تمامی پیکسل های تصویر یک پیمایش انجام داده و بر اساس اینکه در کدام یک از چهار ناحیه ی ایجاد شده قرار می گیرد ، میانگین ابتدا و انتهای ناحیه به جای مقدار فعلی پیکسل قرار می گیرد تا کل عکس به چهار سطح روشنایی تبدیل شود. برای مثال اگر مقدار پیکسل از $T02$ کمتر باشد (بین $T0$ و $T02$) آنگاه میانگین $T0$ و $T02$ به جای مقدار فعلی پیکسل قرار می گیرد.

```

32     [Row, Col] = size(Image);
33     Out = uint8(zeros(Row, Col));
34
35     for i=1: 1: Row
36         for j=1: 1: Col
37             if Image(i, j) <= T0
38                 Out(i, j) = (T0+T01)/2;
39             elseif Image(i, j) <= T01
40                 Out(i, j) = T01/2;
41             elseif Image(i, j) <= T02
42                 Out(i, j) = (T02+T0)/2;
43             else
44                 Out(i, j) = (255+T02)/2;
45             end
46         end
47     end

```

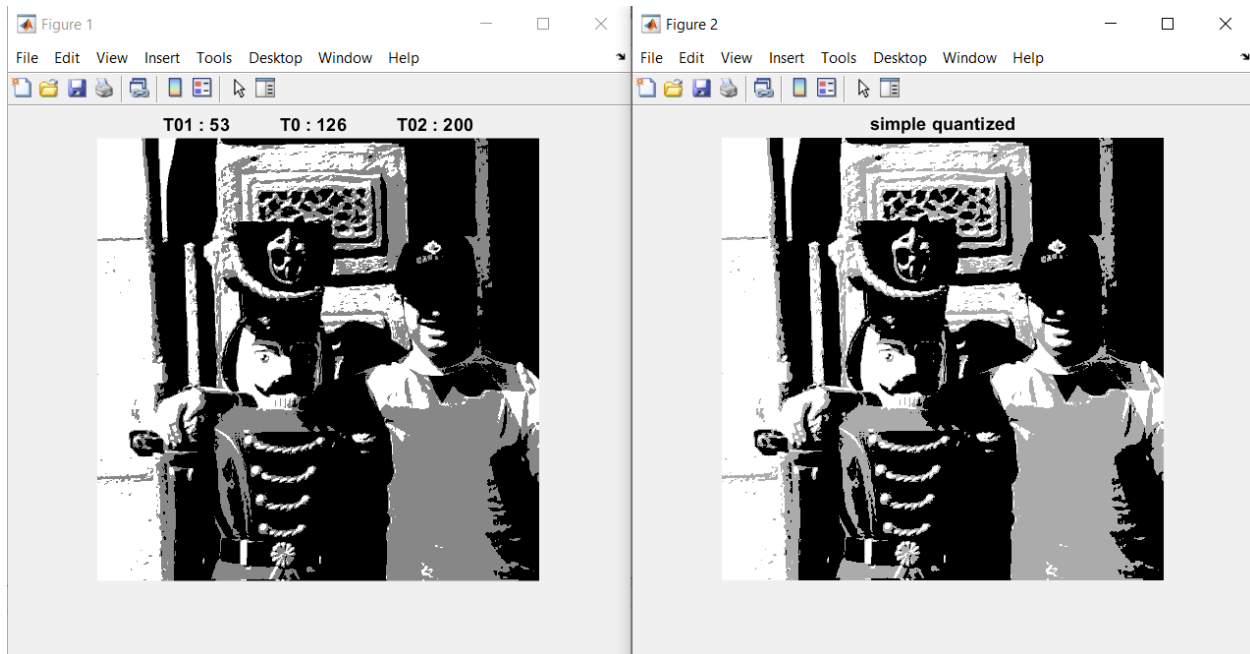
در این قسمت کد تصویر به صورت ساده کوآنتیزه شده و تمام پیکسل ها بر 64 تقسیم و در 64 ضرب شده و در نهایت در کنار خروجی کوآنتیزه نشده نمایش داده می شود.

```

49     SJ64=uint8(Out./64);
50     BJ64=SJ64.*64;
51
52     imshow(Out, []);
53     title(['T01 : ' num2str(T01) '          T0 : ' num2str(T0) '          T02 : ' num2str(T02) ']);
54     figure
55     imshow(BJ64, []);
56     title(['simple quantized']);
57

```

اسکرین شات 2.tif:



توضیحات سوال ۱

قسمت ب:

با استفاده از تابع نوشته شده در تکلیف قبلی `cdf` را به دست می آوریم.

```
3 [CDF, Histogram] = hist_cdf(Image);
```

در این بخش از کد `cdf` را به 2^n بخش تقسیم می کنیم (اندازه ی هر بخش $1/(2^n)$ می شود) و مقدار تصویر در مرزهای تقسیم بندی را بدست می آوریم.

```
4 for i=1:n
5     Cut = i*CDF(256, 1)/n;
6     for j=1: 1: 256
7         if CDF(j, 1) >= Cut
8             I(1,i) = j;
9             break;
10        end
11    end
12 end
```

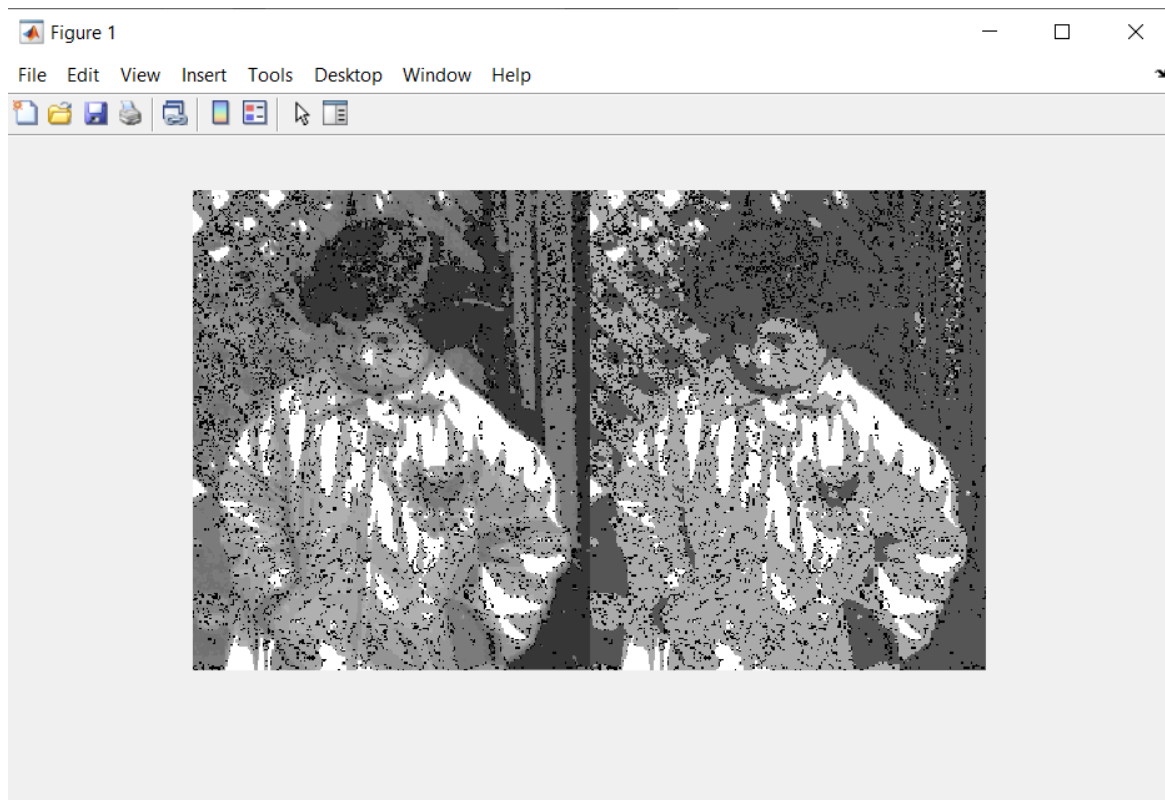
در این قسمت مقدار هر پیکسل با مقدار مرزهای به دست آمده در قسمت قبل مقایسه می شود و به جای مقدار فعلی ، میانگین ابتدا و انتهای ناحیه ای که پیکسل در آن است ، قرار داده می شود.

```
15 [Row, Col] = size(Image);
16 Image=Image;
17 Num=[0,T,255];
18 Out = uint8(zeros(Row, Col));
19 for i=1:Row
20     for j=1:Col
21         for k=1:n
22             if( (k==1) && ( Image(i,j)<T(1,1) ) )
23                 Out(i,j)=T(1,1)/2;
24             end
25             if((k==n) && (Image(i,j)>T(1,n-1)))
26                 Out(i,j)=(255+T(1,n-1))/2;
27             end
28             if( (k~=n) && (k~=1) )
29                 if ((Image(i,j)>T(1,k-1)) && ((Image(i,j)<T(1,k))))
30                     Out(i,j)=(T(1,k-1)+T(1,k))/2;
31                 end
32             end
33         end
34     end
35 end
```

در این قسمت کد تصویر به صورت ساده کوآنتیزه شده و تمام پیکسل ها بر 64 تقسیم و در 64 ضرب شده و در نهایت در کنار خروجی کوآنتیزه نشده نمایش داده می شود.

```
37 SJ64=uint8(Out./64);
38 BJ64=SJ64.*64;
39
40 figure;
41 imshowpair(Out,BJ64,'montage');%
```

اسکرین شات 1.tif:



اسکرین شات 2.tif:



توضیحات سوال ۲:

۱- ابتدا با توجه به سایز بلاک ورودی تابع ، تصویر را `resize` کرده تا در مورد بلاک بندی مشکلی پیش نیاید.

```
[row, col] = size(Image);

row = row + K - rem(row, K);
col = col + K - rem(col, K);

J = imresize(Image, [row, col]);    % Fit Size of the Image according to the K
temp = uint8(zeros(K, K));
Counter = 0;
```

۲- سپس بر روی عکس پیمایش کرده و بلاک های $K * K$ را جدا می کنیم.

```
a=uint16(1);
b=uint16(1);
for i=1: K: row
    for j=1: K: col
        a=1;
        b=1;
        for m=i: 1: i+K-1           % divide image to K*K images
            for p=j: 1: j+K-1
                temp(a, b) = J(m, p); % local parts are stored in K*K image named "temp"
                b=b+1;
            end
            a=a+1;
            b=1;
        end
    end
end
```

۳- به ازای هر بلاک ضرایب مربوطه را استخراج کرده و اگر از حد آستانه تعیین شده کوچکتر باشد ، آن ها را صفر می کنیم و در نهایت عکس تبدیل می گیریم.

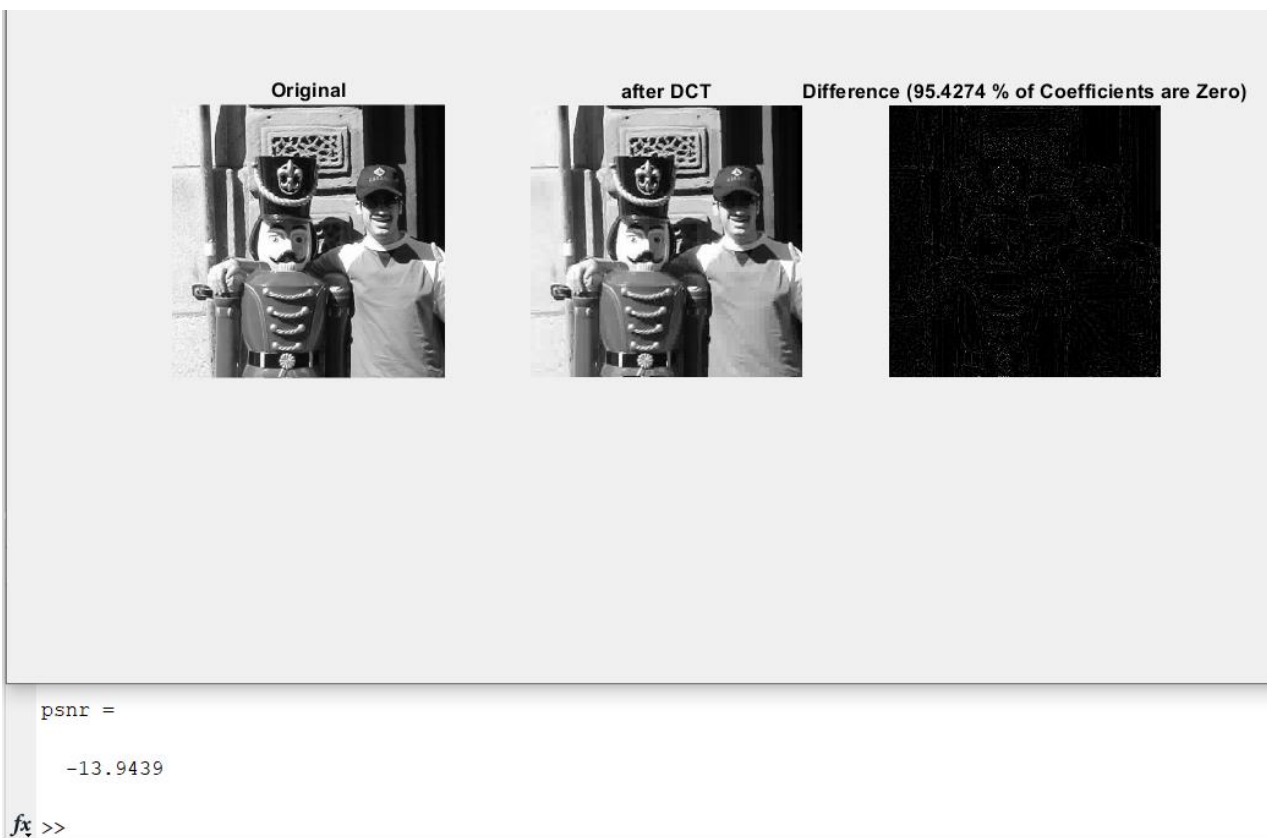
```
temp = dct2(temp);           % Calculate DCT Coefficients
for r=1: 1: K
    for c=1: 1: K
        if abs(temp(r, c)) < T % Set Coefficients with value less than T to 0
            temp(r,c) = 0;
            Counter = Counter + 1;
        end
    end
end
temp = uint8(idct2(temp));    % Calculate IDCT
```

۴- قسمت آخر کد مربوط به درصد ضرائب صفر شده و نمایش موارد خواسته شده در صورت سوال می باشد. همچنین از تابع PSNR استفاده شده است که تابع آن طبق تعریف آن به طور مشخص پیاده سازی شده است.

اسکرین شات 1.tif:



اسکرین شات 2.tif:



پایان

با تشکر از زحمات شما