توضیح سوال ۲ تمرین ۳

کامل کردن تکستچر

```
function F=texture();
   I=imread('/Users/user/Desktop/HW3/dog.jpg');
   %%%%%%%%%%%%%%%%%%%%%%%%%
   x=size(I,2);
   y=size(I,1);
   J=zeros(y,x);
```

تابع جی عکس باینری نقاط سیاه وسط است.

```
   J(I(:,:,1)==0)=1;
   J(I(:,:,2)==0)=1;
   J(I(:,:,3)==0)=1;
   imagesc(J);
   saveas(gcf,cat(2,'/Users/user/Desktop/HW3/16_mask.jpg'),'jpg');
   %%%%%%%%%%%%%%%%%%%%%%%%%
   I1=I;
```

متغیر اس تعداد نقاط پر نشده را نشان میدهد. اجرای برنامه تا آن زمان ادامه میابد که این تعداد به زیر ۱۰۰ برسد

```
   s=sum(sum(J));
   F=zeros(y,x,3);
   JF=J;
   while (s>100)
    JJJ=test7(I1,J);
    JF=JJJ(:,:,1);
    F=JJJ(:,:,2:4);
    s=sum(sum(JF));
    I1=F;
    imshow(F);
    saveas(gcf,cat(2,'/Users/user/Desktop/HW3/Layer',num2str(s),'.jpg'),'jpg');
   end
   imshow(F);
    saveas(gcf,cat(2,'/Users/user/Desktop/HW3/16_impained'.jpg'),'jpg');
end
```

تابع اصلی برنامه:

```
function JJJ=test7(I,J);
   %%%%%%%%%%%%%%%%%%%%%%%%%
   II=I;
   x=size(I,2);
```

```
y=size(I,1);
%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%
JJ=zeros(y,x);
n=25;
```

وسط عکس باینری را در ماتریس زیر میریزد

```
K=J([n+1:y-n],[n+1:x-n]);
```

این ماتریس را میچرخاند. به این ترتیب نقاط مرز بصورت خاکستری تر مشخص میشوند. نقاط خارج سیاه و نقاط وسط کاملا سفید میشوند

```
for i=-n:n
   for j=-n:n
      JJ([n+1+i:y-n+i],[n+1+j:x-n+j])=JJ([n+1+i:y-n+i],[n+1+j:x-n+j])+K;
   end
end
lim1=50;
```

این حد برای تفکیک مرز است.

```
JJ(JJ==(2*n+1)^2)=0;
JJ(JJ>lim1)=255;
JJ=JJ.*J;
JJ=uint8(JJ);
%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%
```

در ای قسمت برنامه برای نقاط مرز، یک ماتریس مربعی جدا میکند و به تابع تست۸ میفرستد که خروجی آن یک قسمت رندوم و شبیه به قسمت مرزی فرستاده شده است

```
Temp=zeros(2*n+1,2*n+1,3);
for i=n+1:n:y-n
   for j=n+1:n:x-n
      if (JJ(i,j)==255)
         Temp=test8(II,i,j);
         II([i-n:i+n],[j-n:j+n],:)=Temp;
      end
   end
end
%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%
```

در اینجا تصویر یک مرحله پر تر شده را به همراه باینری نقاط پر نشده به عنوان خروجی تحویل میدهد

```
JF=J;
JF(II(:,:,1)==0)=1;
JF(II(:,:,2)==0)=1;
JF(II(:,:,3)==0)=1;
JJJ=zeros(y,x,4);
JJJ(:,:,1)=JF;
JJJ(:,:,2)=II(:,:,1);
```

```
    JJJ(:,:,3)=II(:,:,2);
    JJJ(:,:,4)=II(:,:,3);
    %%%%%%%%%%%%%%%%%%%%
end
```

تابع تست۸، تشخیص شباهت و دادن قسمت رندوم مذکور

```
function Temp=test8(I,i,j);
    x=size(I,2);
    y=size(I,1);
    n=25;
    nn=5;
    J=zeros(y,x);
    J=J+200;
    G1=I([i-n:i+n],[j-n:j+n],:);
```

در اینجا روی کل عکس (با پرش به دلیل افزایش سرعت) سوییپ میکند و میزان شباهت دا در ماتریس جی میریزد. هر چی مقدار تست ۹ کم تر باشد، شباعت بیشتر است. مقدار اولیه این ماتریس ۲۰۰ داده شده است

```
    for ii=n+1:nn:y-n;
        for jj=n+1:nn:x-n
            G2=I([ii-n:ii+n],[jj-n:jj+n],:);
            J(ii,jj)=test9(G1,G2,n);
        end
    end
```

حد بزرگتر از ۲ را به عنوان برش شباهت در نظر میگیریم. ماتریس باینری جی جی نقاط شبیه را مشخص میکند که تعداد نقاط مورد نظر در کا ریخته میشود. یک عدد رندوم بین ۱ تا کا انتخاب شده و مربع متناظر با پیکسل مورد نظر (شبیه آر ام) به عنوان خروجی باز گردادنده میشود.

```
    JJ=J<2;
    k = sum(sum(JJ));
    r = randi(k,1,1);
    counter=0;
    quit=0;
    p1=0;
    p2=0;
    for ii=n+1:nn:y-n;
        for jj=n+1:nn:x-n
            if ( JJ(ii,jj) )
                counter=counter+1;
            end
```

```
            if ( counter== r )
                p1=ii;
                p2=jj;
                quit=1;
                break;
            end
        end
        if (quit)
            break;
        end
    end
    Temp=I([p1-n:p1+n],[p2-n:p2+n],:);
end
```

این تابع شباهت دو قطعه ارسال شده را بوسیله فیلتر اس-اس-دی گاوس مشخص میکند.

```
function kk=test9(G1,G2,n);
    G3=G1*0;
    G3(G1==0)=1.00;
    G1=G1+G2.*G3;
    GG=(G1-G2).*(G1-G2);
    sigma=5;
    GGG=zeros(2*n+1,2*n+1,3);
    X=[-n:n];
    Y=[-n:n];
    Gx=(exp(-(X.*X)/(2*sigma^2)));
    Gy=(exp(-(Y.*Y)/(2*sigma^2)));
    G=Gy'*Gx;
    G=G/(sum(sum(G)));


    GGG(:,:,1)=double(GG(:,:,1)).*G;
    GGG(:,:,2)=double(GG(:,:,2)).*G;
    GGG(:,:,3)=double(GG(:,:,3)).*G;
    k=sum(sum(GGG));
    kk=(k(1)+k(2)+k(3))/3;
end
```

در پایان، به دلیل حجم بالای عملیات های لازم، تنها خروجی لایه اول در فایل آورده شده است.