clc

close all

I = imread('C:\Users\FaZoya\Desktop\OUR PROJECT\brain\_image.png');

I1 = addRVIN1(I,0.15,255);

[M,N] = size(I);

I\_2=cat(1,zeros(2,N),I);

I\_3=cat(1,I\_2,zeros(2,N));

I\_44=cat(2,zeros(M+4,2),I\_3);

orginal=cat(2,I\_44,zeros(M+4,2));

mask1=zeros(M,N);%%% my noisy mask

count1=0;

final=zeros(M,N);

I2=cat(1,I1(1:2,1:M),I1);

I3=cat(1,I2,I1(1:2,1:M));

I44=cat(2,I3(1:N+4,1:2),I3);

J=cat(2,I44,I3(1:N+4,1:2));

tl=0;

t=1;

for rep=1:2

if(rep>1)

mask1=zeros(M,N);

I2=cat(1,final(1:2,1:M),final);

I3=cat(1,I2,final(1:2,1:M));

I44=cat(2,I3(1:N+4,1:2),I3);

J=cat(2,I44,I3(1:N+4,1:2));

tl=2;

th=1;

end

ts=30;

ts2=20;

ts3=150;

ts4=300;

Th\_SMa=15;

Th\_FMa = 40;

Th\_FMb = 80;

Tsim=6;

TM=10;

for i=1:M

for j=1:N

sort1=zeros(1,9);

hh1=1;

for r=i+1:i+3

for c=j+1:j+3

sort1(hh1)=J(r,c);

hh1=hh1+1;

end

end

W=[J(i+1,j+1),J(i+1,j+2),J(i+1,j+3),J(i+2,j+1),J(i+2,j+2),J(i+2,j+3),J(i+3,j+1),J(i+3,j+2),J(i+3,j+3)];

MaxW=max(W);

MinW=min(W);

sort2=sort(sort1);

z4=abs(double(J(i+2,j+2))-double(sort2(4)));

z5=abs(double(J(i+2,j+2))-double(sort2(5)));

z6=abs(double(J(i+2,j+2))-double(sort2(6)));

h1=abs(double (sort2(4))-double(sort2(5)));

h2=abs(double (sort2(5))-double(sort2(6)));

min\_z4=sort2(4)-30;

max\_z6=sort2(6)+30;

G=(sort2(4)+sort2(5)+sort2(6))/3;

d1\_n=abs(double (J(i+2,j+2))-double(J(i+2,j+1)))+abs(double(J(i+2,j+2))-double(J(i+2,j+3)))+(1/2)\*abs(double(J(i+2,j+2))-double(J(i+2,j+4)))+(1/2)\*abs(double(J(i+2,j+2))-double(J(i+2,j)));

d2\_n=abs(double (J(i+2,j+2))-double(J(i+1,j+2)))+abs(double(J(i+2,j+2))-double(J(i+3,j+2)))+(1/2)\*abs(double(J(i+2,j+2))-double(J(i,j+2)))+(1/2)\*abs(double(J(i+2,j+2))-double(J(i+4,j+2)));

d3\_n=abs(double (J(i+2,j+2))-double(J(i+1,j+1)))+abs(double(J(i+2,j+2))-double(J(i+3,j+3)))+(1/2)\*abs(double(J(i+2,j+2))-double(J(i+4,j+4)))+(1/2)\*abs(double(J(i+2,j+2))-double(J(i,j)));

d4\_n=abs(double (J(i+2,j+2))-double(J(i+1,j+3)))+abs(double(J(i+2,j+2))-double(J(i+3,j+1)))+(1/2)\*abs(double(J(i+2,j+2))-double(J(i,j+4)))+(1/2)\*abs(double(J(i+2,j+2))-double(J(i+4,j)));

direct1=[J(i+1,j+3),J(i,j+4),J(i+3,j+1),J(i+4,j)];

direct2=[J(i+1,j+2),J(i,j+2),J(i+3,j+2),J(i+4,j+2)];

direct3=[J(i+1,j+1),J(i,j),J(i+3,j+3),J(i+4,j+4)];

direct4=[J(i+2,j+1),J(i+2,j),J(i+2,j+3),J(i+2,j+4)];

ave1=(J(i+1,j+3)+J(i,j+4)+J(i+3,j+1)+J(i+4,j))/4;

ave2=(J(i+1,j+2)+J(i,j+2)+J(i+3,j+2)+J(i+4,j+2))/4;

ave3=(J(i+1,j+1)+J(i,j)+J(i+3,j+3)+J(i+4,j+4))/4;

ave4=(J(i+2,j+1)+J(i+2,j)+J(i+2,j+3)+J(i+2,j+4))/4;

var1=abs(double(ave1)-(double(J(i+1,j+3))))+ abs(double(ave1)-double(J(i,j+4)))+abs(double(ave1)-double(J(i+3,j+1)))+abs(double(ave1)-double(J(i+4,j)));

var2=abs(double(ave2)-(double(J(i+1,j+2))))+ abs(double(ave2)-double(J(i,j+2)))+abs(double(ave2)-double(J(i+3,j+2)))+abs(double(ave2)-double(J(i+4,j+2)));

var3=abs(double(ave3)-(double(J(i+1,j+1))))+ abs(double(ave3)-double(J(i,j)))+abs(double(ave3)-double(J(i+3,j+3)))+abs(double(ave3)-double(J(i+4,j+4)));

var4=abs(double(ave4)-(double(J(i+2,j+1))))+ abs(double(ave4)-double(J(i+2,j)))+abs(double(ave4)-double(J(i+2,j+3)))+abs(double(ave4)-double(J(i+2,j+4)));

var\_met=[var1,var2,var3,var4];

varmin=min(var\_met);

%%%%%%%%%%%%%%%%%%%%%%% Edge-Preserved Filtering A

J=double(J);

D1\_1=100000000000000;

D2\_1=100000000000000;

D4\_1=100000000000000;

D5\_1=100000000000000;

D6 =100000000000000;

D7 =100000000000000;

D8 =100000000000000;

D3 =100000000000000;

a=J(i+1,j+1);

b=J(i+1,j+2);

c=J(i+1,j+3);

d=J(i+2,j+1);

e=J(i+2,j+3);

f=J(i+3,j+1);

g=J(i+3,j+2);

h=J(i+3,j+3);

if((h < max\_z6 && e < max\_z6) && (e > min\_z4 && h > min\_z4))

D1\_1=abs(a-e)+abs(d-h);

end

if((h < max\_z6 && g < max\_z6) && (g > min\_z4 && h > min\_z4))

D2\_1=abs(a-g)+abs(b-h);

end

if( (g < max\_z6) && (g > min\_z4 ))

D3=abs(b-g)\*2;

end

if((f < max\_z6 && g < max\_z6) && (g > min\_z4 && f > min\_z4))

D4\_1=abs(c-g)+abs(b-f);

end

if((e < max\_z6 && f < max\_z6) && (e > min\_z4 && f > min\_z4))

D5\_1=abs(c-d)+abs(e-f);

end

if( (e < max\_z6) && (e > min\_z4 ))

D6=abs(d-e)\*2;

end

if( (h < max\_z6) && (h > min\_z4 ))

D7=abs(a-h)\*2;

end

if( (f < max\_z6) && (f > min\_z4 ))

D8=abs(c-f)\*2;

end

zw1=[D1\_1,D2\_1,D3,D4\_1,D5\_1,D6,D7,D8];

minz=min(zw1);

if(minz==100000000000000)

count1=count1+1;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% start

if((h1>ts2 ) )

direct=[d1\_n,d2\_n,d3\_n,d4\_n];

mindirect=min(direct);

if(mindirect<ts3 ) %%%%%%%%%%%%%%%%% Edge-Detection A

if(rep==2)

sim=0;

for s1=i+1:i+3

for s2=j+1:j+3

if(abs(double (J(s1,s2))- double(J(i+2,j+2)))>TM)

sim=sim+1;

end

end

end

if(sim > Tsim) %%%%%%%%%%%%%%%%%%% Similarity Checking

final(i,j)=G;

mask1(i,j)=1;

else

final(i,j)=J(i+2,j+2);

end

else

final(i,j)=J(i+2,j+2);

end

else

if(rep==2)

final(i,j)=G;

mask1(i,j)=1;

else

%%%%%%%%%%%% Edge-Preserved Filtering B

if(varmin==var1)

final(i,j)= median(direct1);

mask1(i,j)=1;

elseif(varmin==var2)

final(i,j)= median(direct2);

mask1(i,j)=1;

elseif(varmin==var3)

final(i,j)= median(direct3);

mask1(i,j)=1;

elseif(varmin==var4)

final(i,j)= median(direct4);

mask1(i,j)=1;

end

end

end

elseif((h2>ts2 )) %%%%%%%%%%%%%%%% Edge-Detection A

direct=[d1\_n,d2\_n,d3\_n,d4\_n];

mindirect=min(direct);

if(mindirect<ts3 )

if(rep==2)

sim=0;

for s1=i+1:i+3

for s2=j+1:j+3

if(abs(double (J(s1,s2))- double(J(i+2,j+2)))>TM)

sim=sim+1;

end

end

end

if(sim > Tsim) %%%%%%%%% Similarity Checking

final(i,j)=G;

mask1(i,j)=1;

else

final(i,j)=J(i+2,j+2);

end

else

final(i,j)=J(i+2,j+2);

end

else

if(rep==2)

final(i,j)=G;

mask1(i,j)=1;

else

%%%%%%%% Edge-Preserved Filtering B

if(varmin==var1)

final(i,j)= median(direct1);

mask1(i,j)=1;

elseif(varmin==var2)

final(i,j)= median(direct2);

mask1(i,j)=1;

elseif(varmin==var3)

final(i,j)= median(direct3);

mask1(i,j)=1;

elseif(varmin==var4)

final(i,j)= median(direct4);

mask1(i,j)=1;

end

end

end

%%%%%%%%%%%%%% Disorder Analysis

elseif(( z4>ts && z5>ts && z6>ts))

if(minz==100000000000000)

final(i,j)=(a+2\*b+c)/4; %%%%%%%% Edge-Preserved Filtering A

elseif(minz==D1\_1)

fi1=(a+d+e+h)/4;

xc=[J(i+1,j+2),J(i+2,j+1),J(i+2,j+3),J(i+3,j+2)];

lf1=[fi1,xc];

final(i,j)=median(lf1);

elseif(minz==D2\_1)

fi4=(a+b+g+h)/4;

lf4=[fi4,J(i+1,j+2),J(i+2,j+1),J(i+2,j+3),J(i+3,j+2)];

final(i,j)=median(lf4);

elseif(minz==D3)

fi7=(b+g)/2;

lf7=[fi7,J(i+1,j+2),J(i+2,j+1),J(i+2,j+3),J(i+3,j+2)];

final(i,j)=median(lf7);

elseif(minz==D4\_1)

fi8=(b+c+f+g)/4;

lf8=[fi8,J(i+1,j+2),J(i+2,j+1),J(i+2,j+3),J(i+3,j+2)];

final(i,j)=median(lf8);

elseif(minz==D5\_1)

fi11=(c+d+e+f)/4;

lf11=[fi11,J(i+1,j+2),J(i+2,j+1),J(i+2,j+3),J(i+3,j+2)];

final(i,j)=median(lf11);

elseif(minz==D6)

fi13=(d+e)/2;

lf13=[fi13,J(i+1,j+2),J(i+2,j+1),J(i+2,j+3),J(i+3,j+2)];

final(i,j)=median(lf13);

elseif(minz==D7)

fi14=(a+h)/2;

lf14=[fi14,J(i+1,j+2),J(i+2,j+1),J(i+2,j+3),J(i+3,j+2)];

final(i,j)=median(lf14);

elseif(minz==D8)

fi15=(c+f)/2;

lf15=[fi15,J(i+1,j+2),J(i+2,j+1),J(i+2,j+3),J(i+3,j+2)];

final(i,j)=median(lf15);

end

mask1(i,j)=1; %%%%%%%%%%%% Noisy Pixel Checking

elseif(abs(double(J(i+2,j+2))-double(MaxW) )<10|| abs(double(J(i+2,j+2))-double(MinW ))<10)

if(rep==2)

sim=0;

for s1=i+1:i+3

for s2=j+1:j+3

if(abs(double (J(s1,s2))- double(J(i+2,j+2)))>TM)

sim=sim+1;

end

end

end

if(sim > Tsim) %%%%%% Similarity Checking

final(i,j)=G;

mask1(i,j)=1;

else

final(i,j)=J(i+2,j+2);

end

else

final(i,j)=J(i+2,j+2);

end

else

final(i,j)=J(i+2,j+2);

end

end

end

end

err1=(sum (sum((double(I)-double(I1)).^2)))/((M)\*(N));

psnr1=10\*log10(255^2/err1);

display(psnr1);

err2=(sum (sum((double(I)-double(final)).^2)))/((M)\*(N));

psnr2=10\*log10(255^2/err2);

display(psnr2);

figure(1)

imshow(I1)

title('Noisy MRI image')

figure(2)

result = mat2gray(final,[0 255]);

imshow(result)

title('De-noised MRI image')

input = mat2gray(I,[0 255]);

n\_img = mat2gray(I1,[0 255]);

[ssimval,ssimmap] = ssim(n\_img,input);

figure(3)

imshow(ssimmap,[])

title(['Local SSIM Map with Global SSIM Value: ',num2str(ssimval)])

[ssimval,ssimmap] = ssim(result,input);

figure(4)

imshow(ssimmap,[])

title(['Local SSIM Map with Global SSIM Value: ',num2str(ssimval)])