DIGITAL IMAGE PROCESSING
ASSIGNMENT 2

BY
M V S SRAVAN
201531197

QUESTION 1:
A) GAUSSIAN AND LAPLACIAN PYRAMIDS

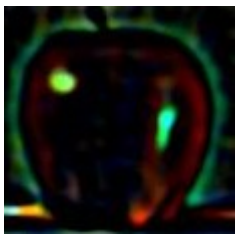
CODE

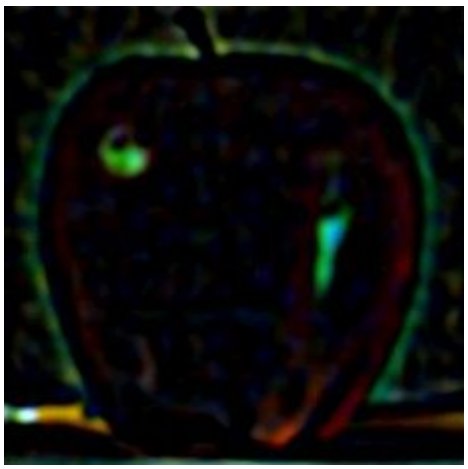
```
close all;
clear;
clc;

srcFiles =
dir('C:\Users\SRAVAN\Desktop\academics\dip\assignment2\Assign2_
imgs\Assign2_imgs\image_blending_with_laplacian_pyramid\example
1\*.jpg');
im1=strcat('C:\Users\SRAVAN\Desktop\academics\dip\assignment2\A
ssign2_imgs\Assign2_imgs\image_blending_with_laplacian_pyramid\
example1\',srcFiles(1).name);
%im2=strcat('C:\Users\SRAVAN\Desktop\academics\dip\assignment2\
Assign2_imgs\Assign2_imgs\image_blending_with_laplacian_pyramid
\example1\',srcFiles(2).name);

i1=imread(im1);
%i2=imread(im2);

i1=double(i1);
%i2=double(i2);
g1=getgaussian(i1,4);
l11=i1-g1;

i11=imresize(g1,0.75);

g2=getgaussian(i11,4);
l12=i11-g2;

i12=imresize(g2,0.5);

g3=getgaussian(i12,4);
l13=i12-g3;
i13=imresize(g3,0.75);

figure,imshow(uint8(l11)*10);
figure,imshow(uint8(l12)*10);
figure,imshow(uint8(l13)*10);
```

```
figure,imshow(uint8(i11));
figure,imshow(uint8(i12));
figure,imshow(uint8(i13));

function G=getgaussian(image,k)
I=image;
sigma = k;
sz = 5;
[x,y]=meshgrid(-sz:sz,-sz:sz);
M = size(x,1)-1;
N = size(y,1)-1;
Exp_comp = -(x.^2+y.^2)/(2*sigma*sigma);
Kernel= exp(Exp_comp)/(2*pi*sigma*sigma);
Output=zeros(size(I));
%I = padarray(I,[sz sz]);
G=imfilter(I,Kernel);
%G=G(3:size(image,1)-2,3:size(image,2)-2);
end
```

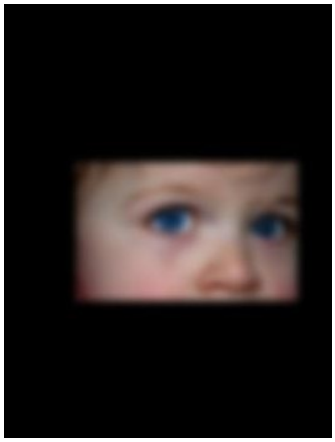
RESULTS:

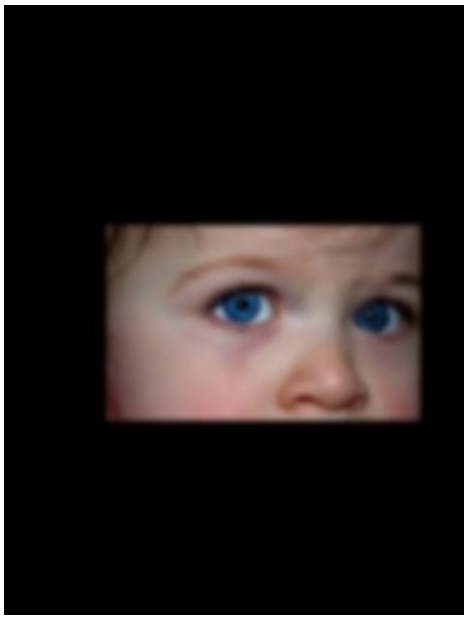
GAUSSIAN PYRAMID

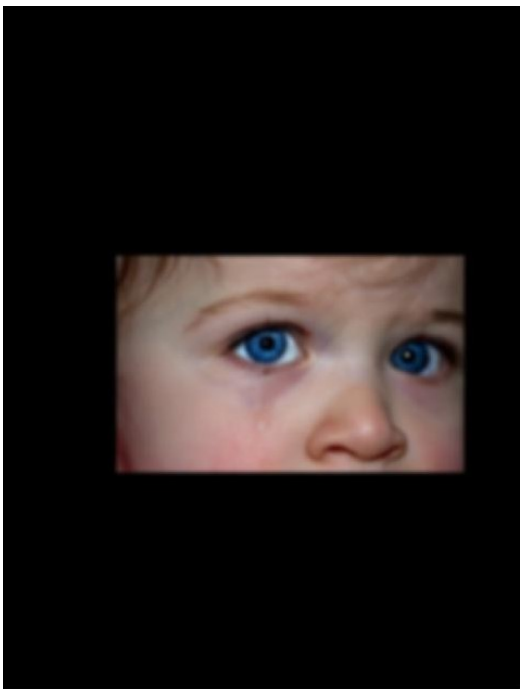




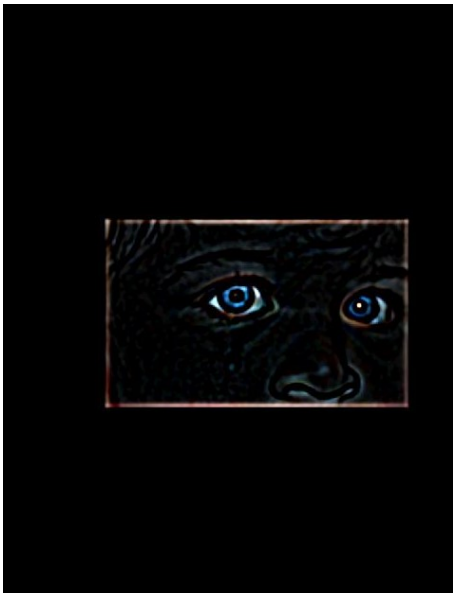
GAUSSIAN PYRAMID

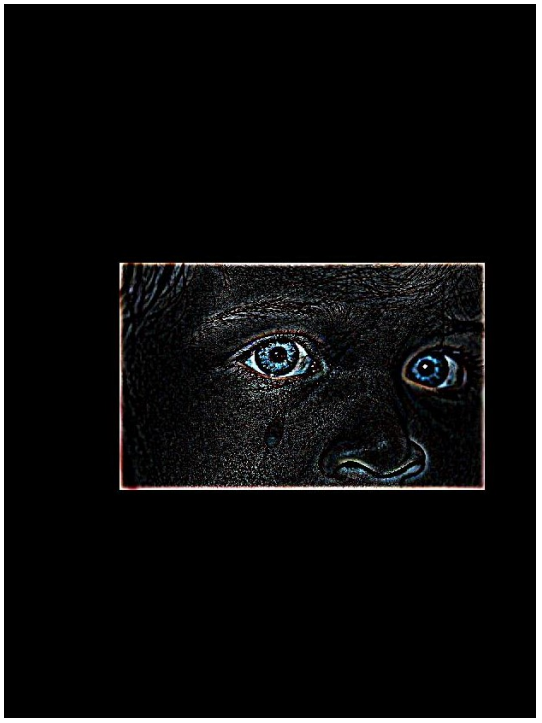






LAPLACIAN PYRAMIDS





B) IMAGE BLENDING

CODE

```
close all;
clear;
clc;

srcFiles =
dir('C:\Users\SRAVAN\Desktop\academics\dip\assignment2\Assign2_
imgs\Assign2_imgs\image_blending_with_laplacian_pyramid\example
1\*.jpg');
im1=strcat('C:\Users\SRAVAN\Desktop\academics\dip\assignment2\A
ssign2_imgs\Assign2_imgs\image_blending_with_laplacian_pyramid\
example1\',srcFiles(1).name);
im2=strcat('C:\Users\SRAVAN\Desktop\academics\dip\assignment2\A
ssign2_imgs\Assign2_imgs\image_blending_with_laplacian_pyramid\
example1\',srcFiles(2).name);
%srcFiles =
dir('C:\Users\SRAVAN\Desktop\academics\dip\assignment2\Assign2_
imgs\Assign2_imgs\image_blending_with_laplacian_pyramid\example
1\*.jpg');
im3=strcat('C:\Users\SRAVAN\Desktop\academics\dip\assignment2\A
ssign2_imgs\Assign2_imgs\image_blending_with_laplacian_pyramid\
example1\',srcFiles(3).name);

%figure,imshow(im1);
%figure,imshow(im2);
%figure,imshow(im3);
temp=30;
i1=imread(im1);
i2=imread(im2);
mask=imread(im3);

%figure,imshow(i1);
%figure,imshow(i2);
%figure,imshow((uint8(mask)));
mask=im2double(mask);
i1=double(i1);
i2=double(i2);
sigma=50;
mask=imgaussfilt(mask,sigma);
%figure,imshow(uint8(mask));
%G=fspecial('gaussian',[50,50],30);
%mask=imfilter(mask,G);
%figure,imshow(mask);
```

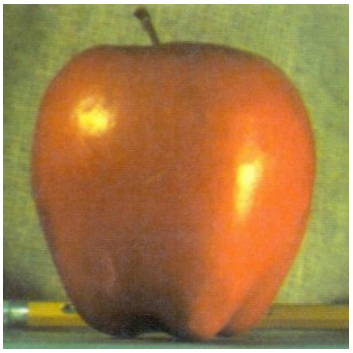
```
%figure,imshow(i1);
%figure,imshow(i2);
Itemp1=imresize(i1,0.25);
Itemp2=imresize(i2,0.25);
mask1=imresize(mask,0.25);

%subplot(1,2,1),
%imshow(uint8(mask1));
I1=Itemp1;
I2=Itemp2;
for i=1:size(I1,1)
for j=1:size(I1,2)
    for k=1:3
  I1(i,j,k)=Itemp1(i,j,k).* mask1(i,j);
    end
end
end
for i=1:size(I2,1)
for j=1:size(I2,2)
    for k=1:3
  I2(i,j,k)=Itemp2(i,j,k).* (1-mask1(i,j));
    end
end
end
f1=I1+I2;
figure,imshow(uint8(f1));

Itemp21=imresize(i1,0.5);
Itemp22=imresize(i2,0.5);
mask21=imresize(mask,0.5);
%subplot(1,2,2),
%imshow(mask21);
Itemp21=getlaplacian(Itemp21,temp);
Itemp22=getlaplacian(Itemp22,temp);
%figure,imshow(Itemp21);
%figure,imshow(Itemp22);

I21=Itemp21;
I22=Itemp22;
%size(I21)
for i=1:size(Itemp21,1)
for j=1:size(Itemp21,2)
    for k=1:3
  I21(i,j,k)=Itemp21(i,j,k).* mask21(i,j);
    end
end
```

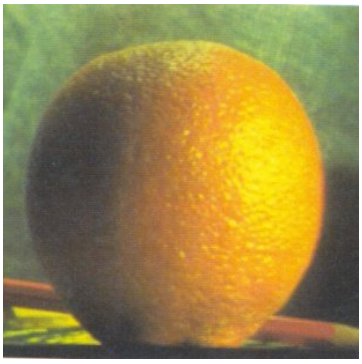
```
end
for i=1:size(Itemp22,1)
for j=1:size(Itemp22,2)
    for k=1:3
  I22(i,j,k)=Itemp22(i,j,k).* (1-mask21(i,j));
    end
end
end
f2=I21+I22;

f2=f2+(imresize(f1,[size(f2,1),size(f2,2)]));
figure,imshow(uint8(f2));

Itemp31=imresize(i1,0.75);
Itemp32=imresize(i2,0.75);
mask31=imresize(mask,0.75);
%figure,imshow(mask31);
Itemp31=getlaplacian(Itemp31,temp);
Itemp32=getlaplacian(Itemp32,temp);
I31=Itemp31;
I32=Itemp32;
for i=1:size(Itemp31,1)
for j=1:size(Itemp31,2)
    for k=1:3
  I31(i,j,k)=Itemp31(i,j,k).* mask31(i,j);
    end
end
end
for i=1:size(Itemp32,1)
for j=1:size(Itemp32,2)
    for k=1:3
  I32(i,j,k)=Itemp32(i,j,k).* (1-mask31(i,j));
    end
end
end
f3=I31+I32;
f3=f3+(imresize(f2,[size(f3,1),size(f3,2)]));
figure,imshow(uint8(f3));


Itemp41=imresize(i1,1);
Itemp42=imresize(i2,1);
mask41=imresize(mask,1);
%figure,imshow(mask41);
Itemp41=getlaplacian(Itemp41,temp);
Itemp42=getlaplacian(Itemp42,temp);
```

```
I41=Itemp41;
I42=Itemp42;
for i=1:size(Itemp41,1)
for j=1:size(Itemp41,2)
    for k=1:3
  I41(i,j,k)=Itemp41(i,j,k).* mask41(i,j);
    end
end
end
for i=1:size(Itemp42,1)
for j=1:size(Itemp42,2)
    for k=1:3
  I42(i,j,k)=Itemp42(i,j,k).* (1-mask41(i,j));
    end
end
end
f4=I41+I42;
f4=f4+(imresize(f3,[size(i1,1),size(i1,2)]));
figure,imshow(uint8(f4));

function final=getlaplacian(image,k)
G=getgaussian(image,k);
size(G);
final=image-G;
%figure,imshow(final);
end

function G=getgaussian(image,k)
I=image;
sigma = k;
sz = 5;
[x,y]=meshgrid(-sz:sz,-sz:sz);
M = size(x,1)-1;
N = size(y,1)-1;
Exp_comp = -(x.^2+y.^2)/(2*sigma*sigma);
Kernel= exp(Exp_comp)/(2*pi*sigma*sigma);
Output=zeros(size(I));
%I = padarray(I,[sz sz]);
G=imfilter(I,Kernel);
%G=G(3:size(image,1)-2,3:size(image,2)-2);
end
```

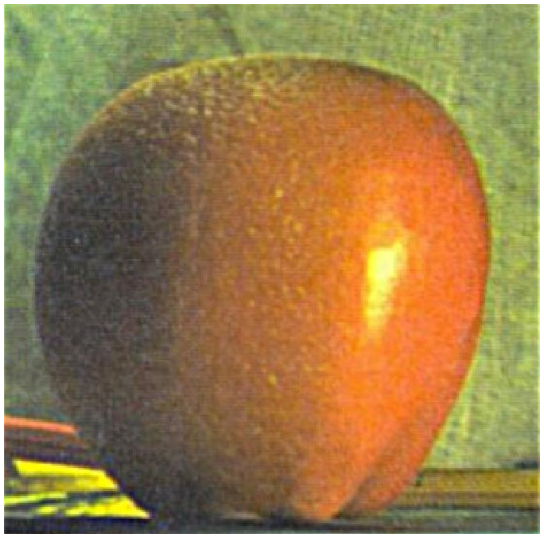
RESULTS

ORIGINAL IMAGE





OUTPUTS

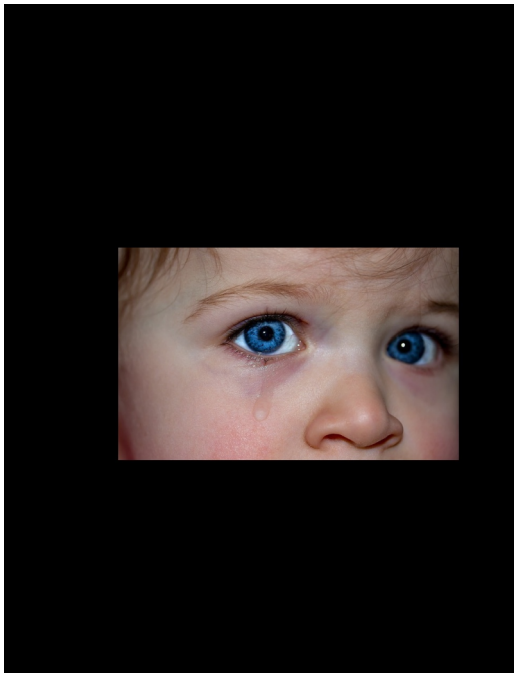


ORIGINAL IMAGES

LEVEL 1 IMAGE

FINAL IMAGE

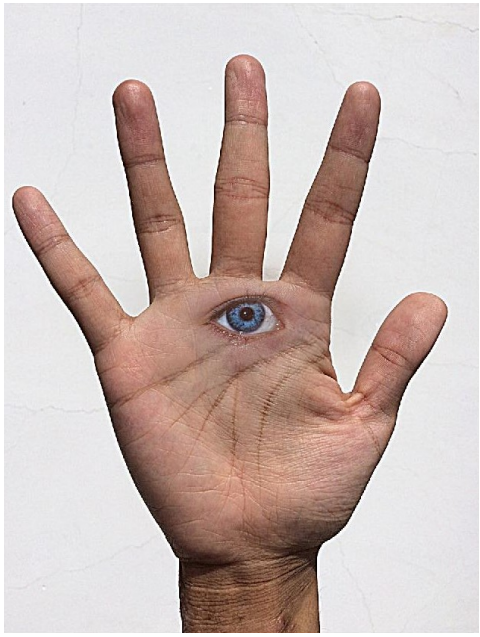
INPUT IMAGE



LEVEL 1

LEVEL 2

FINAL IMAGE



C) IMAGE UPSAMPLING

NEAREST NEIGHBOUR

CODE

```
img=imread('lighthouse.png');
img=double(img);
img2=zeros(2*size(img,1),2*size(img,2),3);
img2=double(img2);
figure,imshow(uint8(img));
for i=1:size(img2,1)
for j=1:size(img2,2)
    if(rem(i,2)==1 & rem(j,2)==1)
        for k=1:3
    img2(i,j,k)=img(floor(i/2)+1,floor(j/2)+1,k);
        end
        end
end
end
%imshow(uint8(img2));

for i=1:2:size(img2,1)-2
for j=1:2:size(img2,2)-2
```

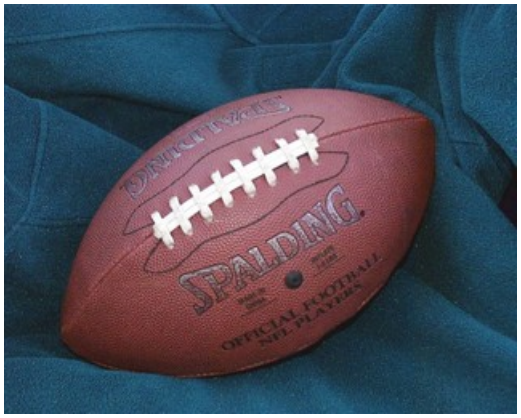
```
for k=1:3
    img2(i,j+1,k)=img2(i,j,k);
    img2(i+1,j,:)=img2(i,j,k);
    img2(i+1,j+1,k)=img2(i,j,k);
end
end
end


figure,imshow(uint8(img2));
```

RESULTS

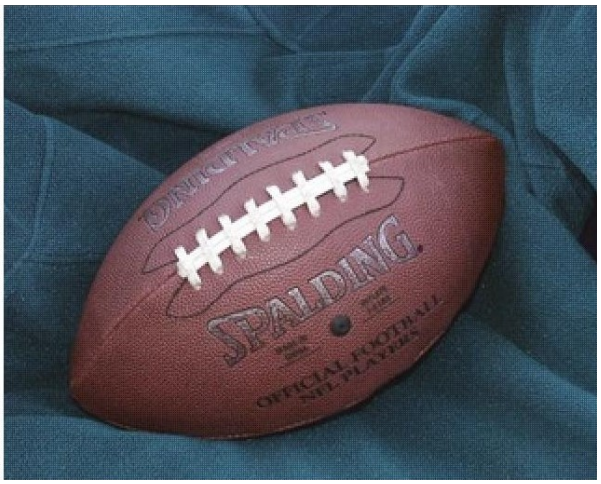
ORIGINAL IMAGE

2X IMAGE

ORIGINAL IMAGE



2X IMAGE



LINEAR INTERPOLATION

CODE

```
img=imread('lighthouse.png');
img=double(img);
img2=zeros(2*size(img,1),2*size(img,2),3);
img2=double(img2);
figure,imshow(uint8(img));
for i=1:size(img2,1)
for j=1:size(img2,2)
    if(rem(i,2)==1 & rem(j,2)==1)
        for k=1:3
    img2(i,j,k)=img(floor(i/2)+1,floor(j/2)+1,k);
        end
        end
end
end
%imshow(uint8(img2));

for i=1:2:size(img2,1)-1
for j=2:2:size(img2,2)
    if(j==size(img2,2))
        for k=1:3
        img2(i,j,k)=img2(i,j-1,k);
        end
    else
for k=1:3
    img2(i,j,k)=(img2(i,j+1,k)+img2(i,j-1,k))/2;
end
end
end
end

for i=2:2:size(img2,1)
for j=1:size(img2,2)
for k=1:3
    img2(i,j,k)=img2(i-1,j,k);
end
end
end
figure,imshow(uint8(img2));
```

RESULTS

ORIGINAL IMAGE

2X IMAGE

BILINEAR INTERPOLATION

CODE

```
clear;
close all;
clc;
im=imread('lighthouse.png');
figure,imshow(im);
i2=zeros(2*size(im,1),2*size(im,2),3);
im=double(im);
i2=double(i2);
for i=1:size(i2,1)
for j=1:size(i2,2)
    if(rem(i,2)==1 & rem(j,2)==1)
        for k=1:3
    i2(i,j,k)=im(floor(i/2)+1,floor(j/2)+1,k);
        end
        end
end
end
%imshow(uint8(i2));
for i=1:2:size(i2,1)
for j=2:2:size(i2,2)
    %if(rem(i,2)==1 & rem(j,2)==1)
       if(j==size(i2,2))
        for k=1:3
            i2(i,j,k)=i2(i,j-1,k);
        end
       else
            for k=1:3
          i2(i,j,k)=(i2(i,j-1,k)+i2(i,j+1,k))/2;
            end
       end
end
end
for i=2:2:size(i2,1)
for j=1:2:size(i2,2)
    %if(rem(i,2)==1 & rem(j,2)==1)
       if(i==size(i2,1))
        for k=1:3
    i2(i,j,k)=i2(i-1,j,k);
        end
       else
            for k=1:3
```

```
            alpha=floor( (i2(i-1,j,k)+i2(i+1,j,k) )/2);
            i2(i,j,k)=alpha;
              end
        end
end
end


for i=2:2:size(i2,1)-2
for j=2:2:size(i2,2)-2
    if(rem(i,2)==0 & rem(j,2)==0)
       %if(i==size(i2,1))
        for k=1:3
    i2(i,j,k)=floor((i2(i,j-1,k)+i2(i,j+1,k))/2);
        end
              end
end
end
figure,imshow(uint8(i2));
```

ORIGINAL IMAGE

2X IMAGE

BICUBIC INTERPOLATION

CODE

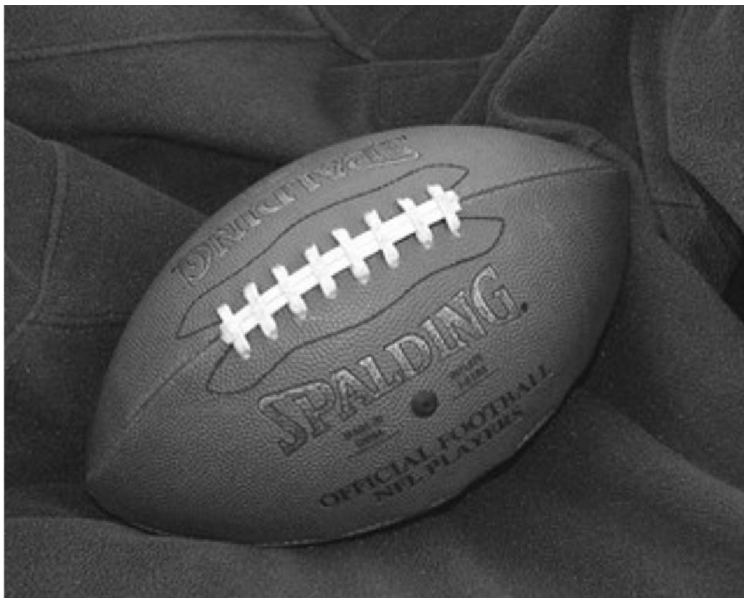
```
clear;
close all;
clc;
i=imread('football.jpg');
k=2;
final=bicubic(i,k*size(i,1),k*size(i,2));
figure,imshow(uint8(final));
k=3;
final=bicubic(i,k*size(i,1),k*size(i,2));
figure,imshow(uint8(final));


function output_image = bicubic( input_image,xdim,ydim)
I = input_image;
I = double(rgb2gray(I));
[j k] = size(I);
x_new = xdim;
y_new = ydim;

x_scale = x_new./(j-1);
y_scale = y_new./(k-1);
temp_image = zeros(x_new,y_new);
Ix = double(zeros(j,k));
Iy = double(zeros(j,k));
Ixy = double(zeros(j,k));

Ix=getxd(I,j,k);
Iy=getyd(I,j,k);
Ixy=getxyd(I,j,k);

for count1 = 0:x_new-1
 for count2 = 0:y_new-1
 W = -(((count1./x_scale)-floor(count1./x_scale))-1);
 H = -(((count2./y_scale)-floor(count2./y_scale))-1);
 I11_index =
[1+floor(count1./x_scale),1+floor(count2./y_scale)];
 I21_index =
[1+floor(count1./x_scale),1+ceil(count2./y_scale)];
 I12_index =
[1+ceil(count1./x_scale),1+floor(count2./y_scale)];
 I22_index = [1+ceil(count1./x_scale),1+ceil(count2./y_scale)];
 I11 = I(I11_index(1),I11_index(2));
 I21 = I(I21_index(1),I21_index(2));
```

```
 I12 = I(I12_index(1),I12_index(2));
 I22 = I(I22_index(1),I22_index(2));
 Ix11 = Ix(I11_index(1),I11_index(2));
 Ix21 = Ix(I21_index(1),I21_index(2));
 Ix12 = Ix(I12_index(1),I12_index(2));
 Ix22 = Ix(I22_index(1),I22_index(2));
 Iy11 = Iy(I11_index(1),I11_index(2));
 Iy21 = Iy(I21_index(1),I21_index(2));
 Iy12 = Iy(I12_index(1),I12_index(2));
 Iy22 = Iy(I22_index(1),I22_index(2));
 Ixy11 = Ixy(I11_index(1),I11_index(2));
 Ixy21 = Ixy(I21_index(1),I21_index(2));
 Ixy12 = Ixy(I12_index(1),I12_index(2));
 Ixy22 = Ixy(I22_index(1),I22_index(2));
 beta = [I11 I21 I12 I22 Ix11 Ix21 Ix12 Ix22 Iy11 Iy21 Iy12
Iy22 Ixy11 Ixy21 Ixy12 Ixy22];
 M_inv = [
 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
 0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0;
 -3,3,0,0,-2,-1,0,0,0,0,0,0,0,0,0,0;
 2,-2,0,0,1,1,0,0,0,0,0,0,0,0,0,0;
 0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0;
 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0;
 0,0,0,0,0,0,0,0,-3,3,0,0,-2,-1,0,0;
 0,0,0,0,0,0,0,0,2,-2,0,0,1,1,0,0;
 -3,0,3,0,0,0,0,0,-2,0,-1,0,0,0,0,0;
 0,0,0,0,-3,0,3,0,0,0,0,0,-2,0,-1,0;
 9,-9,-9,9,6,3,-6,-3,6,-6,3,-3,4,2,2,1;
 -6,6,6,-6,-3,-3,3,3,-4,4,-2,2,-2,-2,-1,-1;
 2,0,-2,0,0,0,0,0,1,0,1,0,0,0,0,0;
 0,0,0,0,2,0,-2,0,0,0,0,0,1,0,1,0;
 -6,6,6,-6,-4,-2,4,2,-3,3,-3,3,-2,-1,-2,-1;
 4,-4,-4,4,2,2,-2,-2,2,-2,2,-2,1,1,1,1
 ];
 alpha = M_inv*beta';
 temp_p=0;

 for counts = 1:16
 w_temp = floor((counts-1)/4);
 h_temp = mod(counts-1,4);
 temp_p = temp_p + alpha(counts).*((1-W)^(w_temp)).*((1-
H)^(h_temp));
 end
 temp_image(count1+1,count2+1)=temp_p;
 end
end
```

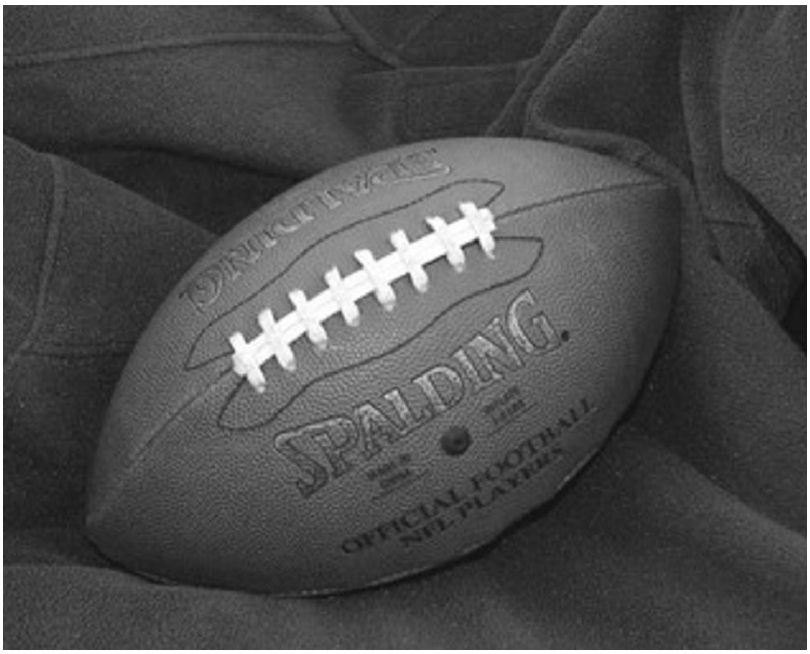
```
output_image = temp_image;
end

function Ix=getxd(I,j,k)

for row = 1:j
 for col = 1:k
 if( (col==1) || (col==k) )
 Ix(row,col)=0;
 else
 Ix(row,col)=(0.5).*(I(row,col+1)-I(row,col-1));
 end
 end
end

end
function Iy=getyd(I,j,k)
for row = 1:j
 for col = 1:k
 if( (row==1) || (row==j) )
 Iy(row,col)=0;
 else
 Iy(row,col)=(0.5).*(I(row+1,col)-I(row-1,col));
 end
 end
end
end

function Ixy=getxyd(I,j,k)

for row = 1:j
 for col = 1:k
 if( (row==1) || (row==j) || (col==1) || (col==k) )
 Ixy(row,col)=0;
 else
 Ixy(row,col)=(0.25).*((I(row+1,col+1)+I(row-1,col-1)) -
(I(row+1,col-1)+I(row-1,col+1)));
 end
 end
end
end
```

RESULTS

2X IMAGE



3X IMAGE



QUESTION 2
A) FFT

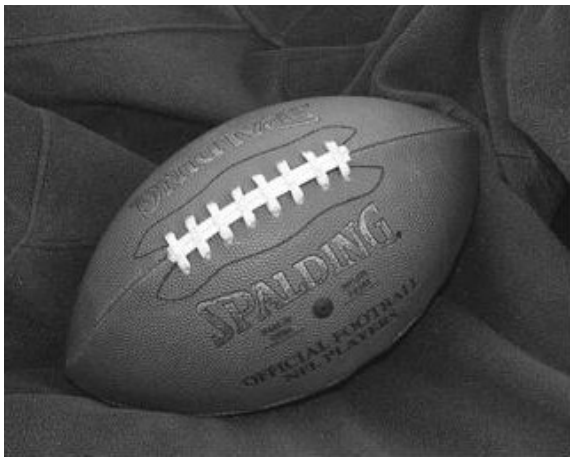
CODE

```
clc;
clear all;
close all;
Im = imread('football.jpg');
Im = rgb2gray(Im);
figure,
imshow(Im);

Mx = size(Im,1);
Ny = size(Im,2);

M1 = 2^(ceil(log2(Mx)));
N1 = 2^(ceil(log2(Ny)));

F = zeros(M1,N1);
F(1:Mx,1:Ny) = Im;
Im2 = zeros(M1,N1);
Im2 = transpose(F);

for i=1:M1
    A = myFFT(Im2(1:N1,i));
    C(i,1:N1) = transpose(A);
end
F1_new = C;
F3 = zeros(M1,N1);
for i=1:Mx
    for j=1:N1
        F3(i,j) = F1_new(i,j);
    end
end
F3_new = zeros(M1,N1);
for j=1:N1
    F3_new(:,j) = myFFT(F3(:,j));
end
figure;
imshow(F3_new);
J = ifft2(ifftshift(F3_new));
final = zeros(Mx,Ny);
for i=1:Mx
    for j=1:Ny
        final(i,j) = J(i,j);
```

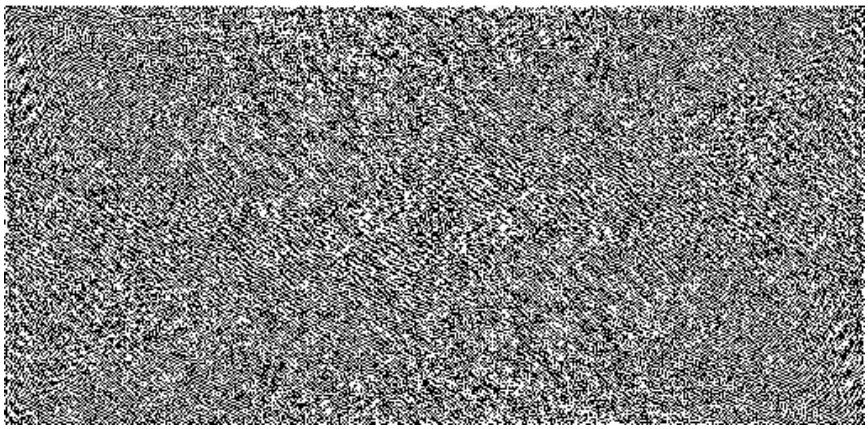
```
    end
end

figure,imshow(uint8(abs(final)));
title('final output');
```

RESULTS

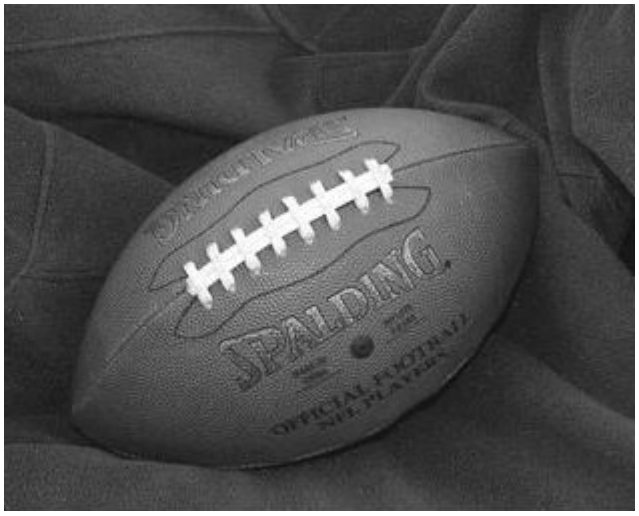
ORIGINAL IMAGE



FOURIER



OUTPUT IMAGE



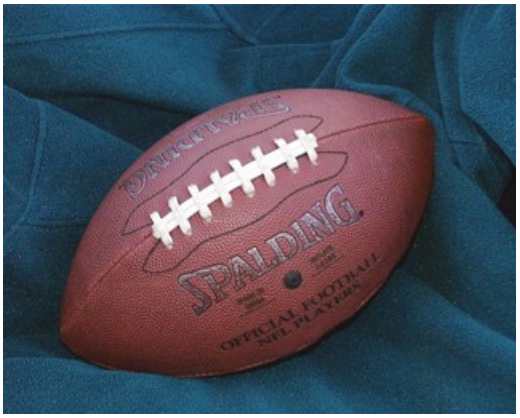
B)
1)IDEAL LOW PASS FILTER

CODE

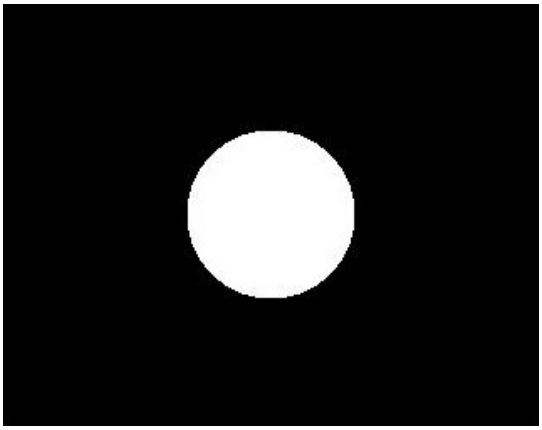
```
close all;
clear;
clc;
i=rgb2gray(imread('football.jpg'));
F= fft2(double(i));
H=zeros(size(F));
H=double(H);
D0=200;
for i=1:size(H,1)
    for j=1:size(H,2)
    D=sqrt((size(H,1)/2-i)^2 + (size(H,2)/2-j)^2);

    if(D<D0)
        H(i,j)=1;
    end
    end

end
%figure,imshow(uint8(H));
FINAL=F.*H;
final=ifft2(FINAL);
figure,imshow(uint8(final));
```

RESULTS

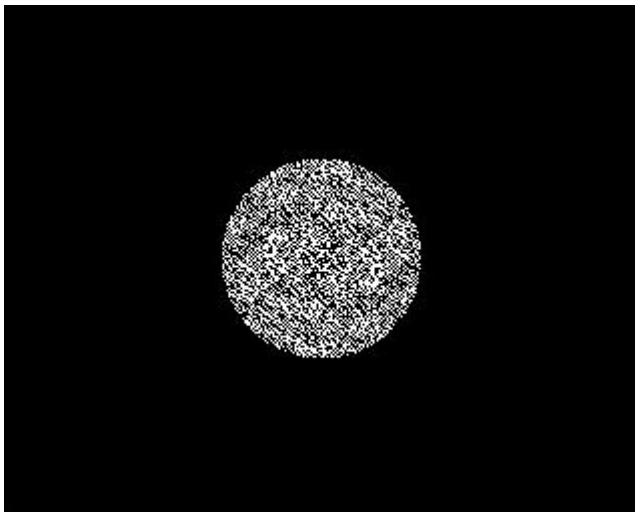
ORIGINAL IMAGE



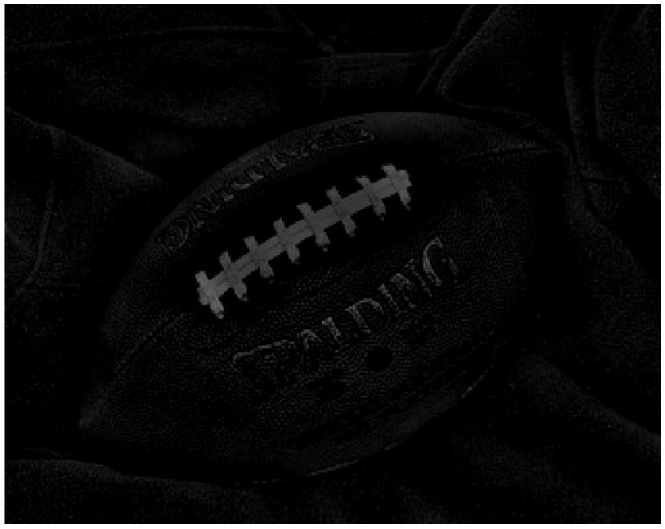
filter



FILTER ON IMAGE IN FREQUENCY DOMAIN



FINAL IMAGE RESULT



2)BUTTERWORTH FILTER

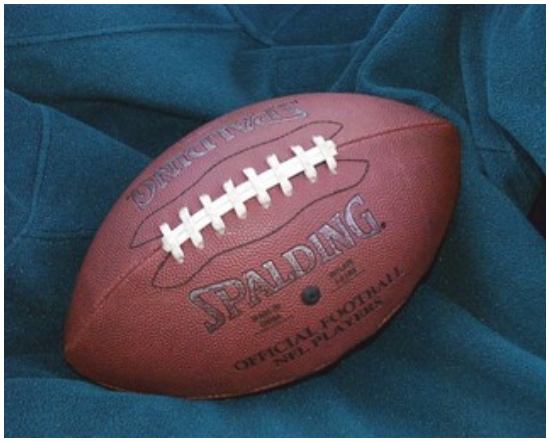
CODE

```
clc;
close all;
clear;
I = imread('football.jpg');
M = size(I,1);
N = size(I,2);
d0 = 100;
n = 2;
%figure;
%imshow(I);
FI = fft2(I);
FI = fftshift(FI);
%figure;
%imshow(FI);
q = zeros(M,N);
for i = 1:M
    for j = 1:N
        dis = (((i - M/2)^2) + (j - N/2)^2)^(1/2);

        q(i,j) = 1/(1+(dis/d0)^(2*n));
```

```
    end
end
mask = cat(3,q,q,q);
filtered = FI.*mask;
%figure;
%imshow(filtered);
filtered = ifftshift(filtered);
final = real(ifft2(filtered));
figure;
imshow(uint8(final));
```

RESULTS

ORIGINAL IMAGE



D0=100

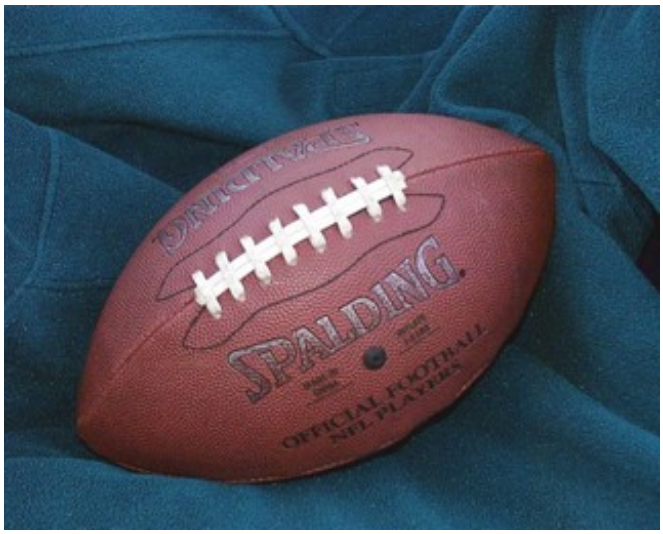
D0=60

3)GAUSSIAN FILTER

CODE

```
clc;
close all;
clear;
I = imread('football.jpg');
M = size(I,1);
N = size(I,2);
d0 = 10;
n = 1;
%figure;
%imshow(I);
FI = fft2(I);
FI = fftshift(FI);
%figure;
%imshow(FI);
q = zeros(M,N);
for i = 1:M
    for j = 1:N
        dis = (((i - M/2)^2) + (j - N/2)^2)^(1/2);

        q(i,j) = exp(-((dis^2)/(2*((d0)^2))));
    end
end
mask = cat(3,q,q,q);
filtered = FI.*mask;
%figure;
%imshow(filtered);
filtered = ifftshift(filtered);
final = real(ifft2(filtered));
figure;
imshow(uint8(final));
```

ORIGINAL IMAGE



RESULT

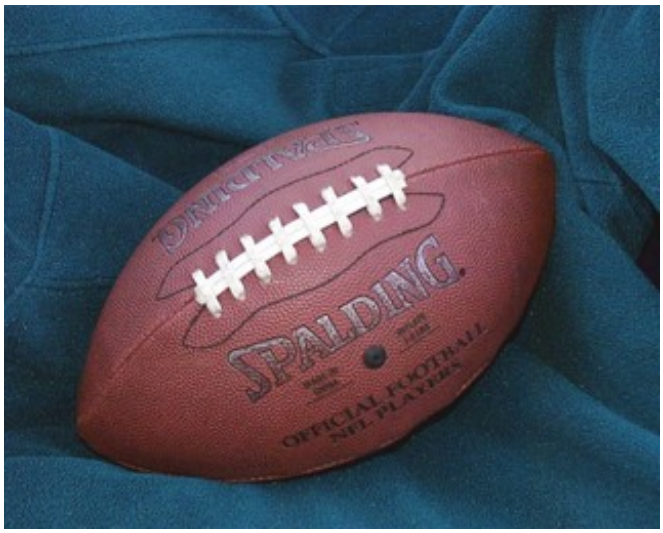
C)LAPLACIAN FILTER

CODE

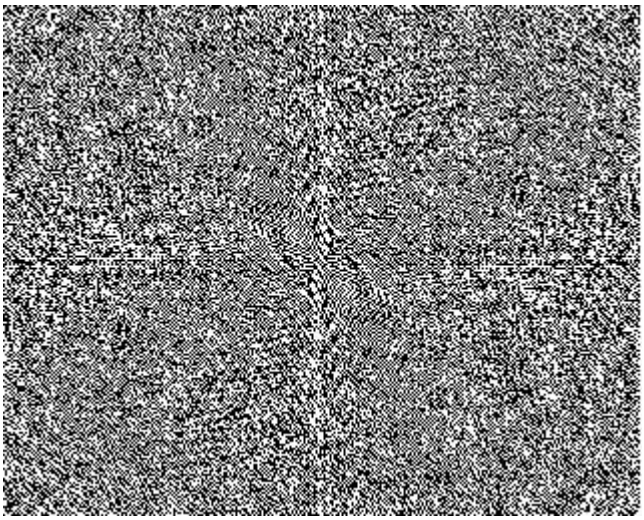
```
close all;
clear;
clc;
I = rgb2gray(imread('football.jpg'));
%figure,imshow(I);
[M,N,l] = size(I);
d0 = 100;
n = 1;
FI = fft2(I);
FI = fftshift(FI);
figure,imshow(real(FI));
lapl=[0,1,0;1,-4,1;0,1,0];
L=fft2(double(lapl),M,N);
L=fftshift(L);
figure,imshow(real(L));
%mask = cat(3,L,L,L);
filtered = FI.*L;
%filtered=filtered+(-1 * min(filtered(:)));
%figure;
%imshow(filtered);
```

```
filtered = ifftshift(filtered);
%figure,imshow(filtered);
final = (ifft2(filtered));
figure;
imshow(uint8(final));
```

ORIGINAL IMAGE

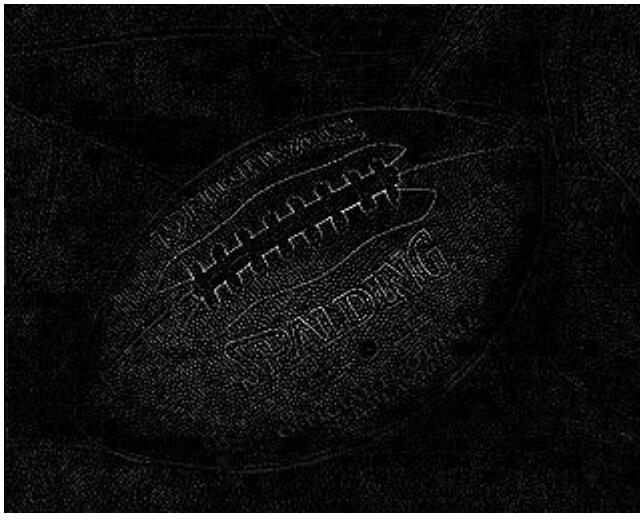


INPUT IMAGE IN FREQ DOMAIN



LAPLACIAN FILTER

RESULT



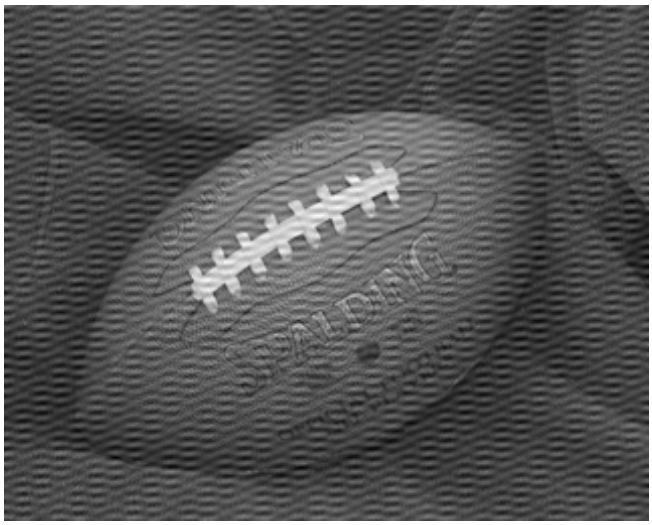
D)NOTCH REJECT

CODE

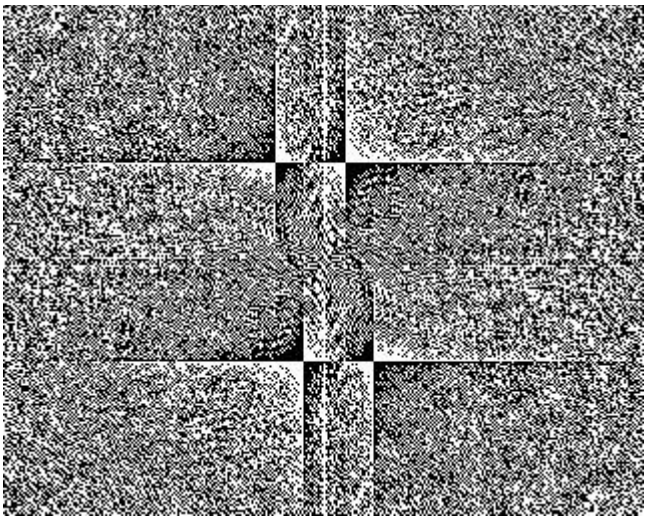
```
clc;
close all;
clear;

I = imread('notch1.png');
%I=uint8(I);
%figure,imshow(I);
[M,N] = size(I);
d0 = 100;
n = 1;
k=2;
FI = fft2(I);
FI = fftshift(FI);
figure,imshow(real(FI));
for i=1:M
    for j=1:N
        for k=-1:1
   if( (i==79+k & (j>115 & j<206))||(i==178+k & (j>110 &
j<220))|| (j==161 & (i<101 || i>145)) || (j==136+k & (i>45 &
i<115)) || (j==171+k & (i>45 & i<115))  ||  (j==186+k & (i>134
& i<224)) || (j==151+k & (i<224 & i>142)))
    FI(i,j)=0;
   end
   end
    end
end

figure,imshow(real(FI));

filtered = ifftshift(FI);
%figure,imshow(filtered);
final = real(ifft2(filtered));
final=uint8(final);
figure;
imshow((final));
```

RESULTS

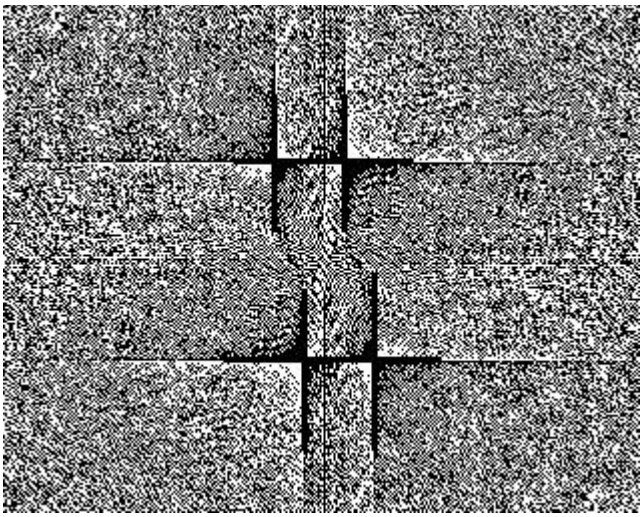
ORIGINAL IMAGE



GIVEN IMAGE IN FREQ DOMAIN



FINAL IMAGE (FREQ)



FINAL IMAGE