**CODE**

**IMPLEMENTATION OF THE ALGORITHM ON THE EXAMPLE GIVEN IN FIG.4.4**

**%initializing x array**

x =[155,156,155,158,157,156,158,159,160,158,158,160,159,156,157,155];

disp(x);

**%initializing the height(H) and width(W) of the image**

H = 4;

W = 4;

**%making d array**

d = zeros(1,H\*W);

index = 1;

d(1)= x(1);

d = [155,1,1,3,1,1,2,1,1,2,0,2,1,3,1,2];

disp(d);

**%finding peak element(peak) and its frequency(peakFreq)**

[peak, peakFreq] = mode(d(:));

disp(peak);

disp(peakFreq);

**%creating message**

message = [0,2,3,1,3,2,1,0];

message =0;

for i=2: peakFreq /2

message = [message , i-1]

end

for i= peakFreq /2 : peakFreq -1

message = [message,NoUse-i-1]

end

disp(message);

**%making y array according to the relation between di and P**

y = zeros(1, H\*W);

y(1) = x(1);

for i=2:H\*W

if d(i) < peak

y(i) = x(i);

elseif d(i) > peak

if x(i) >= x(i-1)

y(i) = x(i) + 3;

else

y(i) = x(i) - 3;

end

end

end

index=1;

for i = 1:H\*W

if d(i) == peak

if x(i) >= x(i-1)

y(i) = x(i) + message(index);

index= index+1;

else

y(i) = x(i) - message(index);

index = index+1 ;

end

end

end

disp(y);

**%construction of masked matrix from y array**

maskedImg = zeros(H, W);

index = 1;

for i=1:H

if mod(i,2)==0

for j=W:-1:1

maskedImg(i,j) = y(index);

index = index+1;

end

else

for j=1:W

maskedImg(i,j) = y(index);

index = index+1;

end

end

end

**%converting masked matrix to image**

embedImg = mat2gray(maskedImg);

**%display the embedded image that is image with message embedded in it**

figure

imshow(embedImg);

**%storing the embedded image as maskedImage.png**

imwrite(embedImg,'maskedImage.png')

**%Extracting message according to P (peak value)**

message='';

for i=2:H\*W

if abs(y(i) - x(i-1)) == peak

message = strcat(message, '0');

elseif abs(y(i) - x(i-1)) == peak+1

message = strcat(message, '1');

elseif abs(y(i) - x(i-1)) == peak+2

message = strcat(message, '2');

elseif abs(y(i) - x(i-1)) == peak+3

message = strcat(message, '3');

end

end

disp(message);

**%creating newx array that restores the original xi value of the host image**

newx = zeros(1,H\*W);

newx(1)=y(1);

for i=2:H\*W

if peak < abs(y(i) - x(i-1)) && abs(y(i)-x(i-1 ))<= (peak+3) && y(i) < x(i-1)

newx(i) = y(i) + (abs(y(i) - x(i-1)) - peak);

elseif peak < abs(y(i) - x(i-1)) && abs(y(i)-x(i-1)) <=(peak+3) && y(i) > x(i-1)

newx(i) = y(i) - (abs(y(i) - x(i-1)) - peak);

elseif abs(y(i) - x(i-1)) > peak+3 && y(i) < x(i-1)

newx(i) = y(i) + 3;

elseif abs(y(i) - x(i-1)) > peak+3 && y(i) > x(i-1)

newx(i) = y(i) - 3;

else

newx(i) = y(i);

end

end

disp(newx);

**%checking whether xi array equal to newxi array**

tf=isequal(x,newx);

disp(tf);

**%constructing the non-masked matrix from newx array**

nonmaskedImg = zeros(H, W);

index = 1;

for i=1:H

if mod(i,2)==0

for j=W:-1:1

nonmaskedImg(i,j) = newx(index);

index = index+1;

end

else

for j=1:W

nonmaskedImg(i,j) = newx(index);

index = index+1;

end

end

end

**%converting non-masked matrix to image**

nonembedImg = mat2gray(nonmaskedImg);

**%display the non-embedded image that is image with message extracted from it**

figure

imshow(nonembedImg);

**%storing the non-embedded image as NonMaskedImg.png'**

imwrite(nonembedImg, 'NonMaskedImg.png');

Screenshots of the results of the above code:



