**Huffmann coding for compression**

%Reading image

a=imread('flowers.jpg');

%converting an image to grayscale

I=rgb2gray(a);

imwrite(a,"gray\_castle.jpg")

%size of the image

[m,n]=size(I);

Totalcount=m\*n;

%variables using to find the probability

cnt=1;

sigma=0;

%computing the cumulative probability.

for i=0:255

k=I==i;

count(cnt)=sum(k(:))

%pro array is having the probabilities

pro(cnt)=count(cnt)/Totalcount;

sigma=sigma+pro(cnt);

cumpro(cnt)=sigma;

cnt=cnt+1;

end;

%Symbols for an image

symbols = [0:255];

%Huffman code Dictionary

dict = huffmandict(symbols,pro);

%function which converts array to vector

vec\_size = 1;

for p = 1:m

for q = 1:n

newvec(vec\_size) = I(p,q);

vec\_size = vec\_size+1;

end

end

%Huffman Encodig

hcode = huffmanenco(newvec,dict);

%Huffman Decoding

dhsig1 = huffmandeco(hcode,dict);

%convertign dhsig1 double to dhsig uint8

dhsig = uint8(dhsig1);

%vector to array conversion

dec\_row=sqrt(length(dhsig));

dec\_col=dec\_row;

%variables using to convert vector 2 array

arr\_row = 1;

arr\_col = 1;

vec\_si = 1;

for x = 1:m

for y = 1:n

back(x,y)=dhsig(vec\_si);

arr\_col = arr\_col+1;

vec\_si = vec\_si + 1;

end

arr\_row = arr\_row+1;

end

%converting image from grayscale to rgb

[deco, map] = gray2ind(back,256);

RGB = ind2rgb(deco,map);

imwrite(RGB,'decoded.JPG');

b=imread("decoded.JPG");

subplot(1,2,1);

imshow(I);

title("Original")

subplot(1,2,2);

imshow(b);

title("Compressed");