DSP LAB

LAB 3

|  |  |
| --- | --- |
| **Name** | **ID** |
| **Ali Khaled Mohamed** | **19017540** |
| **Seif Hisham El Didi** | **19017524** |
| **Ali Ehab Mohamed Sono** | **19016012** |
| **Ali Mohamed Ali El Ruby** | **19016023** |
| **Ali Mostafa El Sayed El Naklawy** | **19016024** |

**MATLAB Code:**

%Here we computed 8\*8 DCT Matrix

N=8;

C8=zeros(8,8);

for k=0:7

for r=0:7

if(k==0)

u=sqrt(1/N);

else

u=sqrt(2/N);

end

C8(k+1,r+1)=u\*cos((pi\*k\*(r+0.5))/N);

end

end

% Here we read the image and converted it to grey scale

arr1=imread('Leo.jpeg');

arr = im2double(rgb2gray(arr1));

%In this part we check if the image is not divisible by 8 and pad extra zeros to make it fit

[rows colms]=size(arr);

if mod(rows,8) ~= 0 %Here we check on the rows

needed\_rows=8-mod(rows,8);

add\_rows=zeros(needed\_rows,colms);

arr=[arr;add\_rows];

[rows colms]=size(arr);

end

if mod(colms,8) ~= 0 %Here we check on the columns

needed\_colms=8-mod(colms,8);

add\_colms=zeros(rows,needed\_colms);

arr=[arr add\_colms];

[rows colms]=size(arr);

end

% Here we divide the image into 8\*8 pixels

BlockRowsStep = rows / 8;

blockVectorR = [8 \* ones(1, BlockRowsStep)];

BlockColmsStep=colms/8;

blockVectorC=[8 \* ones(1,BlockColmsStep)];

ca = mat2cell(arr, blockVectorR, blockVectorC);

%Here is the quantization matrix

DCTQ1 =[16 11 10 16 24 40 51 61;

12 12 14 19 26 58 60 55;

14 13 16 24 40 57 69 56;

14 17 22 29 51 87 80 62;

18 22 37 56 68 109 103 77;

24 35 55 64 81 104 113 92;

49 64 78 87 103 121 120 101;

72 92 95 98 112 100 103 99];

%The user is asked to enter the scale factor

ra=input('Enter the scale factor you want : NB.(Range is from 2:100 as 1 will be Black image)\n');

r=1/ra;

%It was found that r needs to be smaller than 1 as any integer values will distroy the image

%as the DCTQ matrix will much larger than the image matrix hence, the resultant matrix will consist of zeros (Black Image)

DCTQ=r\*DCTQ1;

ca = mat2cell(arr, blockVectorR, blockVectorC);

[x y]=size(ca);

output1=zeros(rows,colms);

%DCT is done here

for i=1:x

for j=1:y

ca{i,j}=C8 \*ca{i,j} \*C8';

ca{i,j}= round(ca{i,j}./DCTQ);

end

end

%Inverse DCT is done here

for i=1:x

for j=1:y

ca{i,j}= ca{i,j}.\*DCTQ;

ca{i,j}=C8' \* ca{i,j} \*C8;

end

end

%The image is merged here

out=cell2mat(ca);

Result=cellfun(@(x)x\*C8,ca,'un',0) ;

figure;

imshow(out);

imsave;

**Original Image:**

**A picture containing text

Description automatically generated**

**Original Gray-Scaled Image:**

**A picture containing text, outdoor

Description automatically generated**

**Compressed Images with different scale factors:**

**N.B:** As the scale factor “r” increases, quality increases and compression decreases.

1. **At 10% scale factor:**

****

1. **At 30% scale factor:**

****

1. **At 50% scale factor:**

****

1. **At 70% scale factor:**

**A picture containing text, outdoor

Description automatically generated**

1. **At 100% scale factor:**

**A picture containing text

Description automatically generated**