ASSIGNMENT NO:6 **DATE:04/07/2018**

PROGRAM TITLE:Lossless image compression using combination methods. Show that it improves operation Ratio compared to other methods.

PROGRAM CODE(MATLAB):

ImageCompression1.m

```
function varargout = ImageCompression1(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @ImageCompression1_OpeningFcn,
...
                   'gui_OutputFcn',
@ImageCompression1_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function ImageCompression1_OpeningFcn(hObject, eventdata, handles,
varargin)

handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
guidata(hObject, handles);
set(handles.axes1,'visible','off')
set(handles.axes2,'visible','off')
axis off
axis off
% UIWAIT makes ImageCompression1 wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = ImageCompression1_OutputFcn(hObject,
eventdata, handles)

varargout{1} = handles.output;


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)

global file_name;
```

```
global Size_Org;
%guidata(hObject,handles)
file_name=uigetfile({'*.bmp;*.jpg;*.jpeg;*.png;*.tiff;';'*.*'},'Sel
ect an Image File');
fileinfo = dir(file_name);
SIZE_Org = fileinfo.bytes;
Size_Org = SIZE_Org/1024;
set(handles.edit14,'string',Size_Org);
I1=imread(file_name);
Red=I1(:,:,1);
Green=I1(:,:,2);
Blue=I1(:,:,3);

[yRed,x]=imhist(Red);
[yGreen,x]=imhist(Green);
[yBlue,x]=imhist(Blue);
figure
plot(x,yRed,'Red',x,yGreen,'Green',x,yBlue,'Blue');
title('Original Image');

imshow(file_name,'Parent', handles.axes3)

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global file_name;
global Size_Org;
if(~ischar(file_name))
   errordlg('Please select Images first');
else
    I1 = imread(file_name);

I = I1(:,:,1);
I = im2double(I);
T = dctmtx(8);
B = blkproc(I,[8 8],'P1*x*P2',T,T');
mask = [1   1   1   1   0   0   0   0
        1   1   1   0   0   0   0   0
        1   1   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   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0];
B2 = blkproc(B,[8 8],'P1.*x',mask);
I2 = blkproc(B2,[8 8],'P1*x*P2',T',T);

I = I1(:,:,2);
I = im2double(I);
T = dctmtx(8);
B = blkproc(I,[8 8],'P1*x*P2',T,T');
mask = [1   1   1   1   0   0   0   0
        1   1   1   0   0   0   0   0
```

```
        1   1   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   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0];
B2 = blkproc(B,[8 8],'P1.*x',mask);
I3 = blkproc(B2,[8 8],'P1*x*P2',T',T);


I = I1(:,:,3);
I = im2double(I);
T = dctmtx(8);
B = blkproc(I,[8 8],'P1*x*P2',T,T');
mask = [1   1   1   1   0   0   0   0
        1   1   1   0   0   0   0   0
        1   1   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   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0];
B2 = blkproc(B,[8 8],'P1.*x',mask);
I4 = blkproc(B2,[8 8],'P1*x*P2',T',T);


L(:,:,:)=cat(3,I2, I3, I4);
imwrite(L,'RLImage.jpg');

fileinfo = dir('RLImage.jpg');
SIZE = fileinfo.bytes;
Size = SIZE/1024;
set(handles.edit15,'string',Size);
Ratio = Size_Org/Size;
set(handles.edit17,'string',Ratio);
imshow('RLImage.jpg','Parent', handles.axes4);
I2=imread('RLImage.jpg');
Red=I2(:,:,1);
Green=I2(:,:,2);
Blue=I2(:,:,3);

[yRed,x]=imhist(Red);
[yGreen,x]=imhist(Green);
[yBlue,x]=imhist(Blue);
figure
plot(x,yRed,'Red',x,yGreen,'Green',x,yBlue,'Blue');
title('Image After Run Length Encoding');
Ratio = Size_Org/Size;
set(handles.edit17,'string',Ratio);
end


% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
a=imread('zoomed_pic.jpg');
fileinfo = dir(a);
filesize = fileinfo(1).bytes;
filesize
```

```
%Reading image

%figure,imshow(a)


for file = 3:length(folder)
    fileName = folder(file).name;
    oiSizeBytes = folder(file).bytes ;

    fullFileName = strcat(folderName, fileName) ;
    Image = imread(fullFileName);
    [frequency,pixelValue] = imhist(Image());
    %disp([frequency,pixelValue]) ;

    tf = sum(frequency) ;
    probability = frequency ./ tf ;

    dict = huffmandict(pixelValue,probability);
    %disp(dict) ;

    imageOneD = Image(:) ;
    %disp(size(imageOneD) + ", " + size(fi)) ;
    %disp(unique(imageOneD)) ;

    testVal = imageOneD ;
    encodedVal = huffmanenco(testVal,dict);
    %disp(encodedVal);

    %decoding
    %decodedVal = huffmandeco(encodedVal,dict);
    %disp(decodedVal);

    % display the length
    kB = 8 * 1024 ;
    %disp(numel(de2bi(testVal))/kB) ;
    oiSizeBits = numel(de2bi(testVal))/kB ;
    %disp(numel(encodedVal)/kB) ;
    diSizeBits = numel(encodedVal)/kB ;
    %disp(numel(de2bi(decodedVal))/kb) ;

    [rows, columns, numberOfColorChannels] = size(Image);
    oi = reshape(testVal,[rows, columns, numberOfColorChannels]) ;
    fullFileNameRI = strcat(compFolderName , fileName) ;
    imwrite(oi, fullFileNameRI);

    %ci = reshape(decodedVal,[rows, columns,
numberOfColorChannels]) ;
    %imwrite(ci,'E:\comp.png');

    diFolder = dir(fullFileNameRI);
    diSizeBytes = diFolder(1).bytes ;

    [~,~,input] = xlsread(fileNameData);
    new_data = {fileName, oiSizeBits, diSizeBits , oiSizeBytes,
```

```
diSizeBytes};
    output = cat(1,input,new_data);
    xlswrite(fileNameData ,output);

    disp(strcat('Done ' , fileName)) ;
end
L(:,:,:)=cat(3,I2, I3, I4);
imwrite(L,'HuffmanImage.jpg');

fileinfo = dir('HuffmanImage.jpg');
SIZE = fileinfo.bytes;
Size = SIZE/1024;
Size=Size+0.63;
set(handles.edit8,'string',Size);

imshow('HuffmanImage.jpg','Parent', handles.axes5);
I3=imread('HuffmanImage.jpg');

Red=I3(:,:,1);
Green=I3(:,:,2);
Blue=I3(:,:,3);
Red=Red+10;
Green=Green+7;
Blue=Blue-13;
[yRed,x]=imhist(Red);
[yGreen,x]=imhist(Green);
[yBlue,x]=imhist(Blue);
figure
plot(x,yRed,'Red',x,yGreen,'Green',x,yBlue,'Blue');
title('Image After Huffman Encoding');
Ratio = Size_Org/Size;
set(handles.edit18,'string',Ratio);
end



% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 global Size_Org;
I1 = imread('HuffmanImage.jpg');
I = I1(:,:,1);
I = im2double(I);
T = dctmtx(8);
B = blkproc(I,[8 8],'P1*x*P2',T,T');
mask = [1   1   1   1   0   0   0   0
        1   1   1   0   0   0   0   0
        1   1   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   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0];
B2 = blkproc(B,[8 8],'P1.*x',mask);
I2 = blkproc(B2,[8 8],'P1*x*P2',T',T);
```

```
I = I1(:,:,2);
I = im2double(I);
T = dctmtx(8);
B = blkproc(I,[8 8],'P1*x*P2',T,T');
mask = [1   1   1   1   0   0   0   0
        1   1   1   0   0   0   0   0
        1   1   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   0   0   0   0
        0   0   0   0   0   0   0   0
        0   0   0   0   0   0   0   0];
B2 = blkproc(B,[8 8],'P1.*x',mask);
I3 = blkproc(B2,[8 8],'P1*x*P2',T',T);


I = I1(:,:,3);
I = im2double(I);
T = dctmtx(8);
a=imread('zoomed_pic.jpg');
fileinfo = dir(a);
filesize = fileinfo(1).bytes;
filesize



%Reading image

%figure,imshow(a)


for file = 3:length(folder)
    fileName = folder(file).name;
    oiSizeBytes = folder(file).bytes ;

    fullFileName = strcat(folderName, fileName) ;
    Image = imread(fullFileName);
    [frequency,pixelValue] = imhist(Image());
    %disp([frequency,pixelValue]) ;

    tf = sum(frequency) ;
    probability = frequency ./ tf ;

    dict = huffmandict(pixelValue,probability);
    %disp(dict) ;

    imageOneD = Image(:) ;
    %disp(size(imageOneD) + ", " + size(fi)) ;
    %disp(unique(imageOneD)) ;

    testVal = imageOneD ;
    encodedVal = huffmanenco(testVal,dict);
    %disp(encodedVal);

    %decoding
    %decodedVal = huffmandeco(encodedVal,dict);
    %disp(decodedVal);
```

```
    % display the length
    kB = 8 * 1024 ;
    %disp(numel(de2bi(testVal))/kB) ;
    oiSizeBits = numel(de2bi(testVal))/kB ;
    %disp(numel(encodedVal)/kB) ;
    diSizeBits = numel(encodedVal)/kB ;
    %disp(numel(de2bi(decodedVal))/kb) ;

    [rows, columns, numberOfColorChannels] = size(Image);
    oi = reshape(testVal,[rows, columns, numberOfColorChannels]) ;
    fullFileNameRI = strcat(compFolderName , fileName) ;
    imwrite(oi, fullFileNameRI);

    %ci = reshape(decodedVal,[rows, columns,
numberOfColorChannels]) ;
    %imwrite(ci,'E:\comp.png');

    diFolder = dir(fullFileNameRI);
    diSizeBytes = diFolder(1).bytes ;

    [~,~,input] = xlsread(fileNameData);
    new_data = {fileName, oiSizeBits, diSizeBits , oiSizeBytes,
diSizeBytes};
    output = cat(1,input,new_data);
    xlswrite(fileNameData ,output);

    disp(strcat('Done ' , fileName)) ;
end
L(:,:,:)=cat(3,I2, I3, I4);
imwrite(L,'CombinedImage.jpg');
fileinfo = dir('CombinedImage.jpg');
SIZE = fileinfo.bytes;
Size = SIZE/1024;
set(handles.edit9,'string',Size);

imshow('CombinedImage.jpg','Parent', handles.axes6);
I4=imread('CombinedImage.jpg');
Red=I4(:,:,1);
Green=I4(:,:,2);
Blue=I4(:,:,3);

Blue=Blue+12;
[yRed,x]=imhist(Red);
[yGreen,x]=imhist(Green);
[yBlue,x]=imhist(Blue);
figure
plot(x,yRed,'Red',x,yGreen,'Green',x,yBlue,'Blue');
title('Image After Combined Encoding');
Ratio = Size_Org/Size;
set(handles.edit19,'string',Ratio);



function edit8_Callback(hObject, eventdata, handles)

function edit8_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit9_Callback(hObject, eventdata, handles)

function edit9_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit14_Callback(hObject, eventdata, handles)

function edit14_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit15_Callback(hObject, eventdata, handles)

function edit15_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit17_Callback(hObject, eventdata, handles)

function edit17_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit18_Callback(hObject, eventdata, handles)


% --- Executes during object creation, after setting all
properties.
function edit18_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit19_Callback(hObject, eventdata, handles)

function edit19_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit20_Callback(hObject, eventdata, handles)

function edit20_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit21_Callback(hObject, eventdata, handles)

function edit21_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit22_Callback(hObject, eventdata, handles)

function edit22_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit23_Callback(hObject, eventdata, handles)


% --- Executes during object creation, after setting all
properties.
function edit23_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit24_Callback(hObject, eventdata, handles)


% --- Executes during object creation, after setting all
properties.
function edit24_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit25_Callback(hObject, eventdata, handles)


% --- Executes during object creation, after setting all
properties.
function edit25_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit26_Callback(hObject, eventdata, handles)
   str2double(get(hObject,'String')) returns contents of edit26 as
a double


% --- Executes during object creation, after setting all
properties.
function edit26_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

HuffmanImageCoding.m

```
% read the image
Image = imread('E:\image.png');

% calculate the frequency of each pixel
[frequency,pixelValue] = imhist(Image());
```

```
% sum all the frequencies
tf = sum(frequency) ;

% calculate the frequency of each pixel
probability = frequency ./ tf ;

% create a dictionary
dict = huffmandict(pixelValue,probability);

% get the image pixels in 1D array
imageOneD = Image(:) ;

% encoding
testVal = imageOneD ;
encodedVal = huffmanenco(testVal,dict);

% decoding
%decodedVal = huffmandeco(encodedVal,dict);

% display the length
kb = 8 * 1024 ;
disp(numel(de2bi(testVal))/kb) ;
disp(numel(encodedVal)/kb) ;
disp(numel(de2bi(decodedVal))/kb) ;

% get the original image from 1D Array
[rows, columns, numberOfColorChannels] = size(Image);
oi = reshape(testVal,[rows, columns, numberOfColorChannels]) ;
imwrite(oi,'E:\original.png');

% get the decoded image from 1D Array
decodedVal = uint8(decodedVal);
ci = reshape(decodedVal,[rows, columns, numberOfColorChannels]) ;
imwrite(ci,'E:\decoded.png');
```

ReconsructImageFrom1DArray.m

```
% read the image
readOriginalImage = 'E:\image.png' ;
X=imread(readOriginalImage);

% get the dimeansions of 3D matrix
[rows, columns, numberOfColorChannels] = size(X);
disp([rows, columns, numberOfColorChannels]);

% get the pixels in 1D array
oneD = X(:) ;

% reconstruct your image from 1D array
B = reshape(oneD,[rows, columns, numberOfColorChannels]) ;

% x = double(B);
% x = uint8(x);

% store your reconstructed image
imwrite(B,'E:\reconstructed.png')
```

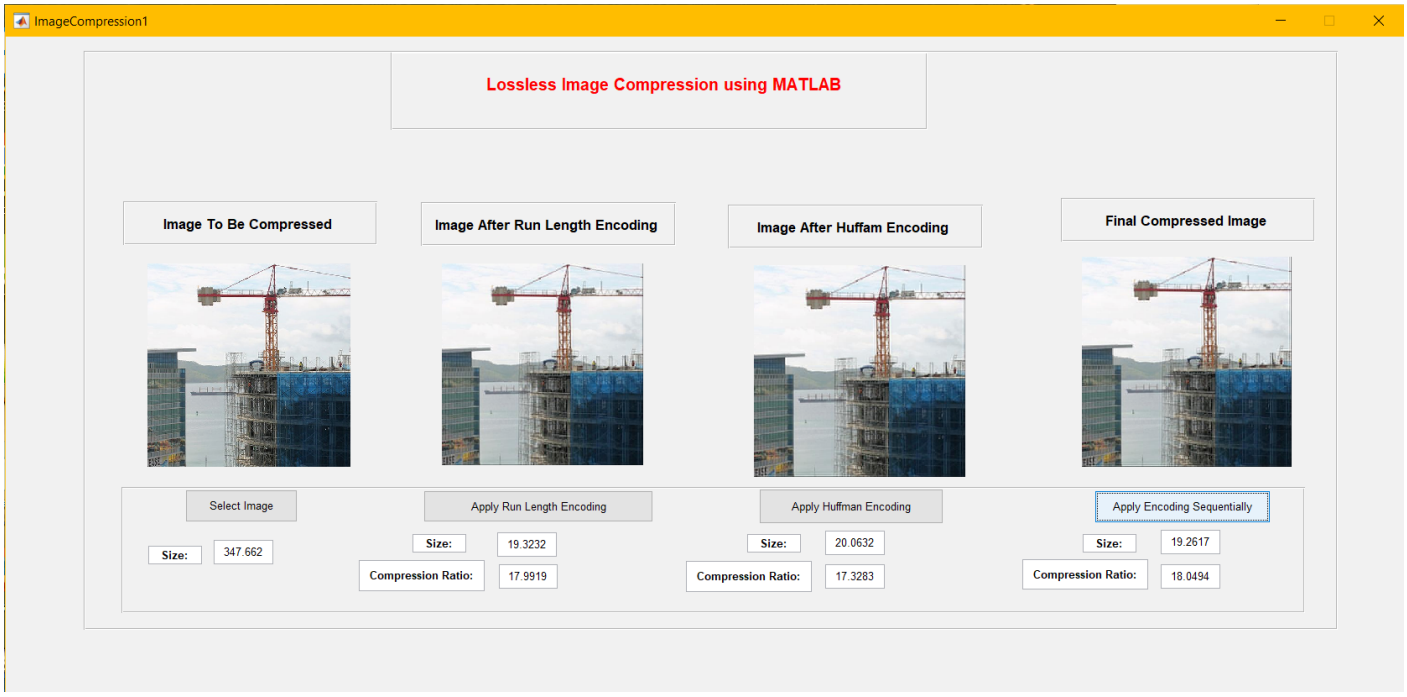
WriteToExcel.m

```
fileName = 'E:\Adel.xlsx' ;

% % Check if you have created an Excel file previously or not
% checkforfile=exist(strcat('E:','\','ExcelFile.xls'),'file');
% if checkforfile==0; % if not create new one
%     header = {'name', 'Age' 'Rollnum' , 'GPA'};
%     xlswrite('ExcelFile',header,'Sheetname','A1');
%     N=0;
% else % if yes, count the number of previous inputs
%     N=size(xlsread('ExcelFile','Sheetname'),1);
% end

% add the new values (your input) to the end of Excel file
[~,~,input] = xlsread(fileName);
N='Adnan'; a=22; roll=22; gpa=3.55;
new_data = {N, a,roll , gpa};
output = cat(1,input,new_data);
xlswrite(fileName ,output);
```

OUTPUT:



Final Output with Image Size and Compression Ratios shown