

# CODE

## DCT in Mat Lab and Guide

```
function varargout = ImageCompression1(varargin)

% IMAGECOMPRESSION1 MATLAB code for ImageCompression1.fig

%   IMAGECOMPRESSION1, by itself, creates a
new IMAGECOMPRESSION1 or raises the existing

%   singleton*.

%

%   H = IMAGECOMPRESSION1 returns the handle to a
new IMAGECOMPRESSION1 or the handle to

%   the existing singleton*.

%

%   IMAGECOMPRESSION1('CALLBACK',hObject,eventData,handles,...)
calls the local

%   function named CALLBACK in IMAGECOMPRESSION1.M with the
given input arguments.

%

%   IMAGECOMPRESSION1('Property','Value',...) creates a
new IMAGECOMPRESSION1 or raises the

%   existing singleton*. Starting from the left, property value pairs are

%   applied to the GUI before ImageCompression1_OpeningFcn gets called.
An

%   unrecognized property name or invalid value makes property application
```

```
%    stop. All inputs are passed to ImageCompression1_OpeningFcn via  
varargin.
```

```
%
```

```
%    *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one  
%    instance to run (singleton)".
```

```
%
```

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help ImageCompression1

% Last Modified by GUIDE v2.5 15-Oct-2014 22:20:56

% Begin initialization code - DO NOT EDIT

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 nargin

    [varargout{1:nargout}] = gui\_mainfcn(gui\_State,  
varargin{:}); else

```
gui_mainfcn(gui_State, varargin{:});
```

```
end
```

```
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before ImageCompression1 is made visible.
```

```
function ImageCompression1_OpeningFcn(hObject, eventdata, handles,  
varargin) % This function has no output args, see OutputFcn. % hObject handle  
to figure
```

```
% eventdata reserved - to be defined in a future version of
```

MATLAB 60

```

% handles      structure with handles and user data (see GUIDATA)

% varargin    command line arguments to ImageCompression1 (see VARARGIN)


% Choose default command line output for ImageCompression1

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 cell array for returning output args (see VARARGOUT);

% hObject    handle to figure

% eventdata  reserved - to be defined in a future version of MATLAB

```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
% Get default command line output from handles structure
```

```
varargout{ 1 } = handles.output;
```

```
% --- Executes on button press in pushbutton1.
```

```
function pushbutton1_Callback(hObject, eventdata, handles)
```

```
% hObject handle to pushbutton1 (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;
```

```
% guidata(hObject,handles)
```

```
file_name=uigetfile({'*.bmp;*.jpg;*.png;*.tiff;','*.*'},'Select an Image File');
```

```
fileinfo = dir(file_name);
```

```
SIZE = fileinfo.bytes;
```

```
Size = SIZE/1024;
```

```
set(handles.text7,'string',Size);
```

```
imshow(file_name, 'Parent', handles.axes1)
```

```
% --- 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;
```

```
if(~ischar(file_name))
```

```
    errordlg('Please select Images first');
```

```
else
```

```
    I1 = imread(file_name);
```

```
    I = I1(:, :, 1);
```

```
    I = im2double(I);
```

```
    T = dctmtx(8);
```

```
    alfa=I1(1:8,1:8,1);
```

```
    disp(alfa);
```

```
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, 'CompressedColourImage.jpg');
fileinfo = dir('CompressedColourImage.jpg');
SIZE = fileinfo.bytes;
Size = SIZE/1024;
set(handles.text8, 'string', Size);
imshow(L, 'Parent', handles.axes2)
end

```

## DWT in Mat Lab and Guide :

```
function varargout = alfa(varargin)
% ALFA MATLAB code for alfa.fig

%   ALFA, by itself, creates a new ALFA or raises the existing
%   singleton*.

%   H = ALFA returns the handle to a new ALFA or the handle to
%   the existing singleton*.

%   ALFA('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in ALFA.M with the given input
%   arguments.

%   ALFA('Property','Value',...) creates a new ALFA or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before alfa_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to alfa_OpeningFcn via varargin.

%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".

%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help alfa
% Last Modified by GUIDE v2.5 12-Mar-2019 22:02:56
% Begin initialization code - DO NOT EDIT

gui_Singleton = 1;
```

```

gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @alfa_OpeningFcn, ...
                  'gui_OutputFcn', @alfa_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

% End initialization code - DO NOT EDIT

% --- Executes just before alfa is made visible.

function alfa_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject    handle to figure

% eventdata  reserved - to be defined in a future version of MATLAB

% handles     structure with handles and user data (see GUIDATA)

% varargin    command line arguments to alfa (see VARARGIN)


% Choose default command line output for
alfa handles.output = hObject;

```

```
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes alfa wait for user response (see UIRESUME)

% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command
line. function varargout = alfa_OutputFcn(hObject, eventdata,
handles)

% varargout cell array for returning output args (see VARARGOUT);
% hObject    handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles
structure varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.

function pushbutton1_Callback(hObject, eventdata, handles)
global file_name;

%guidata(hObject,handles)

file_name=uigetfile({'*.bmp;*.jpg;*.png;*.tiff;'.*'},'Select an Image File');
fileinfo = dir(file_name);

SIZE = fileinfo.bytes;

zSize = SIZE/1024;

imshow(file_name,'Parent', handles.axes2)

set(handles.text18,'string',zSize);

set(handles.text2,'string','Image Loaded');

set(handles.text2,'string','Conversion In Progress');

f=msgbox('Compression Is In Process','Batch 14 ECE');

I=imread(file_name);

[m,n]=size(I);

%Wavelet transform

[cA1,cH1,cV1,cD1] = dwt2(I,'db2');

dec2d = [cA1,cH1;cV1,cD1];

%Inverse Wavelet transform
```

```

IA=idwt2(cA1,[],[],[],'db2');
fna=strcat('.',\Compressed\',file_name);
newf3=strcat(fna,'CompressDWT');
seem=randi(100);
disp(seem);
newf2=strcat(newf3,"");
newf=strcat(newf2,'.jpg');
imwrite(uint8(IA),newf);
set(handles.text2,'string',strcat('Image Written
to',newf )); IH=idwt2([],cH1,[],[],'db2');

IV=idwt2([],[],cV1,[],'db2');
ID=idwt2([],[],[],cD1,'db2');
%Compression ratio
I=double(I);
sumI=0;
sumIA=0;
sumIH=0;
sumIV=0;
sumID=0;
for i=1:m
    for j=1:n
        sumI=sumI+I(i,j);
        sumIA=sumIA+IA(i,j);
        sumIH=sumIH+IH(i,j);
        sumID=sumID+ID(i,j);
        sumIV=sumIV+IV(i,j);
    end
end
end

cr=(sumIA+sumIH+sumID+sumIV)/(sumI);

```

```

display('compression ratio is:');
disp(cr);
%relative data redundancy
red=(1)-(1/cr);
display('relative redundancy is:');
disp(red);
%Calculation of PSNR and compression ratio
squaredErrorImage = (double(I) - double(IA))
.^ 2; mse = sum(sum(squaredErrorImage)) /
(m*n); PSNR = 10 * log10( 255^2 / mse);
display('PSNR for LL band'); display(mse);

display(PSNR);
fileinfo = dir(newf);
OSIZE = fileinfo.bytes;
OSize = OSIZE/1024;
set(handles.text19,'string',OSize);

set(handles.text20,'string',cr);

set(handles.text21,'string',red);

set(handles.text22,'string',PSNR);

figure;

subplot(1,2,1),imshow(file_name),title('Original Image');

subplot(1,2,2),imshow(newf),title('Compressed Image');

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

f=msgbox('Compression Successful \n ....\n ....','Batch 14 ECE');

```

% --- Executes on button press in togglebutton1.

% hObject handle to togglebutton1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of togglebutton1