Digital Image Processing
Project Report

Following are all the features included in the application and a small description for each one:

1. **Load Image:**
   Opens a window from which the user can choose whichever image they want to work on. The image is then displayed in the input axes.

2. **Save Image**
   Once the user is done editing the image, the output can be saved by clicking on this push button. A pop-up window opens up in which the user can choose the path and write the name of the image to be saved.

3. **Increase Brightness**
   This feature increases the brightness of the image by adding a value to each pixel. The output is then displayed in the output axes.

4. **Decrease Brightness**
   This feature decreases the brightness of the image by subtracting a value from each pixel.

5. **Increase Contrast**
   This feature increases the contrast of the image by multiplying a value with each pixel.

6. **Decrease Contrast**
   This feature decreases the contrast of the image by dividing each pixel by a value.

7. **Grayscale**
   This feature turns a three-channel image to a one-channel grayscale image.

8. **Binarize**
   This feature turns a three-channel image to a one-channel grayscale image. The grayscale image is then converted to a black-and-white image.

9. **Negative**
   With this feature, the input image is converted to its negative form and then displayed in the output axes.

10. **Webcam Image Capture**
    With this feature, the application uses the built-in webcam to capture an image of the user

which is then displayed in the input axes and used as input for other features.

11. **Equalize**
    This feature equalizes the input image. The output is then displayed in the output axes.

12. **Contrast Stretching**
    This feature applies contrast stretching on the input image.

13. **Histogram**
    This feature makes a histogram for the input image which is then displayed in the output axes.

14. **Blur**
    This feature adds blur to the input image by using convolution filter.

15. **Sharpen**
    This feature sharpens the image using unsharp masking which basically subtracts a blurred version of the image from the original input image.

16. **Green Screen Effect**
    This feature applies the green screen effect on 2 input images. First input will have a green screen background. The second image will be the replacement background image. This second input can be selected through the 'select second image’ feature. The feature then displays the output in the output axes.

17. **Blue Screen Effect**
    This feature applies the blue screen effect on 2 input images. First input will have a blue screen background. The second image will be the replacement background image. This second input can be selected through the ‘select second image’ feature.

18. **Color Isolator**
    This feature highlights the blue color in a three-channel image and all other colors are converted to grayscale. The output is then displayed in the output axes.

19. **Image Addition**
    This feature takes two input images, converts them both to the same size and then adds corresponding pixel values.

20. **Image Subtraction**
    This feature takes two input images, converts them both to the same size and then subtracts second image’s pixel values from corresponding pixel values of first image.

21. **Image Multiplication**
    This feature takes two input images, converts them both to the same size and then multiplies

first image's pixel values with corresponding pixel values of second image.

22. **Image Division**
    This feature takes two input images, converts them both to the same size and then divides first image's pixel values by corresponding pixel values of second image.

23. **Increase Saturation**
    This functions converts the RGB-format image to HSV-format and then increases the saturation of the image by 90% before converting it back to RGB-format and displaying it in the output axes.

24. **Gamma Encoding**
    This function applies gamma encoding on the input image by raising each pixel value to 2.2.

25. **Connected Components**
    This function converts the image to grayscale and then binarizes it. It then finds all the connected components in the image and draws a red rectangle around each connected component. It also displays total number of components in the 'Results' text box.

26. **Steganography (Embedding)**
    In this feature, user selects two images and the second image is hidden inside the first image using the steganography technique. The output can then be saved by the user.

27. **Steganography (Retrieval)**
    In this function, the function applies the steganography retrieval function on the input image to reveal the hidden image.

28. **Collage**
    This function takes two images as input, converts them to the same size and then builds a collage, which can then be saved by the user.

29. **Noise Removal**
    This function uses 2D median filtering to remove noise from images.

30. **Flip Image**
    This function flips the image horizontally.

31. **Laplacian of Gaussian**
    The LoG function highlight those areas in the image where there is significant change in contrast and the remaining areas are darkened out.

32. **Low Light Enhancement**
    This function builds a complement/negative of the input image, reduces haze from the complemented image and then complements it again. This way, the low light areas appear

brighter in the final output.

33. **Smoothing**
    This function smoothens the input image by applying Gauss filter.

34. **Unique Expression**
    This unique feature resizes the image by making the width the new length of the image and the length the new width of the image. It then applies the following unique expression on it “(((1.5 – im2double(img)) * 1.3) ./ 2.1)” .

35. **Select Second Image**
    This feature is used to select the second input when using features such as image addition, collage, steganography, etc.

Program Code:

```
function varargout = untitled(varargin)
% UNTITLED MATLAB code for untitled.fig
%      UNTITLED, by itself, creates a new UNTITLED or raises the existing
%      singleton*.
%
%      H = UNTITLED returns the handle to a new UNTITLED or the handle to
%      the existing singleton*.
%
%      UNTITLED('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in UNTITLED.M with the given input arguments.
%
%      UNTITLED('Property','Value',...) creates a new UNTITLED or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before untitled_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to untitled_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 untitled

% Last Modified by GUIDE v2.5 08-May-2022 22:32:59

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @untitled_OpeningFcn, ...
                   'gui_OutputFcn',  @untitled_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 untitled is made visible.
function untitled_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no outt 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 untitled (see VARARGIN)

% Choose default command line outt for untitled
handles.outt = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes untitled wait for user response (see UIRESUME)
% uiwait(handles.DIP);


% --- Outputs from this function are returned to the command line.
function varargout = untitled_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning outt 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 outt from handles structure
varargout{1} = handles.outt;




% --- Executes on button press in b1.
function b1_Callback(hObject, eventdata, handles)
% hObject    handle to b1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
[fName, pName] = uigetfile('*jpg; *jpeg; *png; *tif;', 'Select an Image');
if pName ~= 0
    path  = [pName fName];
    img1 = imread(path);
    axes(handles.input);
    imshow(img1);
end


% --- Executes on button press in b2.
function b2_Callback(hObject, eventdata, handles)
% hObject    handle to b2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global out;
h = image(out);
imsave(h);


% --- Executes on button press in b3.
function b3_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to b3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;
out = img1 + 70;
axes(handles.axes11);
imshow(out);


% --- Executes on button press in b4.
function b4_Callback(hObject, eventdata, handles)
% hObject    handle to b4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;
out = img1 - 70;
axes(handles.axes11);
imshow(out);


% --- Executes on button press in b5.
function b5_Callback(hObject, eventdata, handles)
% hObject    handle to b5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;
out = 2 * img1;
axes(handles.axes11);
imshow(out);


% --- Executes on button press in b6.
function b6_Callback(hObject, eventdata, handles)
% hObject    handle to b6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;
out = 0.5 * img1;
axes(handles.axes11);
imshow(out);


% --- Executes on button press in b7.
function b7_Callback(hObject, eventdata, handles)
% hObject    handle to b7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;
out = rgb2gray(img1);
axes(handles.axes11);
```

```
imshow(out);


% --- Executes on button press in b8.
function b8_Callback(hObject, eventdata, handles)
% hObject    handle to b8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;
temp = rgb2gray(img1);
out = imbinarize(temp);
axes(handles.axes11);
imshow(out);


% --- Executes on button press in b9.
function b9_Callback(hObject, eventdata, handles)
% hObject    handle to b9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;
out = 1 - im2double(img1);
axes(handles.axes11);
imshow(out);


% --- Executes on button press in b10.
function b10_Callback(hObject, eventdata, handles)
% hObject    handle to b10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% global img1;
% cam = webcam;
% img1 = snapshot(cam);
% axes(handles.input);
% imshow(img1);


% --- Executes on button press in b11.
function b11_Callback(hObject, eventdata, handles)
% hObject    handle to b11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;
out = histeq(img1,256);
axes(handles.axes11);
imshow(out);


% --- Executes on button press in b12.
function b12_Callback(hObject, eventdata, handles)
% hObject    handle to b12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;
out = imadjust(img1, stretchlim(img1, [0.05, 0.95]),[]);
axes(handles.axes11);
imshow(out);


% --- Executes on button press in b13.
function b13_Callback(hObject, eventdata, handles)
% hObject    handle to b13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
axes(handles.axes11);
imhist(img1);


% --- Executes on button press in b14.
function b14_Callback(hObject, eventdata, handles)
% hObject    handle to b14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;
rgbImage = im2double(img1);
windowSize = 15;
avg3 = ones(windowSize) / windowSize^2;
out = imfilter(rgbImage, avg3, 'conv');
axes(handles.axes11);
imshow(out);


% --- Executes on button press in b15.
function b15_Callback(hObject, eventdata, handles)
% hObject    handle to b15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;
out = imsharpen(img1);
axes(handles.axes11);
imshow(out);


% --- Executes on button press in b16.
function b16_Callback(hObject, eventdata, handles)
% hObject    handle to b16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;                %green screen image
global img2;                %new background image
global out;
```

```
[img1_row, img1_column, ~] = size(img1);
img2 = imresize(img2,[img1_row img1_column]);

WB = img1(:,:,2) > 200;     %adds 1 where true, 0 where false
BW = img1(:,:,2) < 200;

WB = uint8(WB);
BW = uint8(BW);

out1 = BW .* img1;
out2 = WB .* img2;

out = out1 + out2;
axes(handles.axes11);
imshow(out);



% --- Executes on button press in b17.
function b17_Callback(hObject, eventdata, handles)
% hObject    handle to b17 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;                %green screen image
global img2;                %new background image
global out;

[img1_row, img1_column, ~] = size(img1);
img2 = imresize(img2,[img1_row img1_column]);

WB = img1(:,:,3) > 200;     %adds 1 where true, 0 where false
BW = img1(:,:,3) < 200;

WB = uint8(WB);
BW = uint8(BW);

out1 = BW .* img1;
out2 = WB .* img2;

out = out1 + out2;
axes(handles.axes11);
imshow(out);


% --- Executes on button press in b18.
function b18_Callback(hObject, eventdata, handles)
% hObject    handle to b18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;

[img1_row, img1_column,~] = size(img1);
```

```
r = [0 5];
g = [163 168];
b = [175 180];

for i = 1:img1_row
    for j = 1:img1_column
        pixel = img1(i,j,:);
        if (pixel(1) < r(1) || pixel(1) > r(2))  && (pixel(2) < g(1) ||
pixel(2) > g(2)) && (pixel(3) < b(1) || pixel(3) > b(2))
            avg = (pixel(1) + pixel(2) + pixel(3)) / 3;
            pixel(1) = avg;
            pixel(2) = avg;
            pixel (3) = avg;
            img1(i,j,:) = pixel(:,:,:);
        end
    end
end

out = img1;
axes(handles.axes11);
imshow(out);



% --- Executes on button press in b19.
function b19_Callback(hObject, eventdata, handles)
% hObject    handle to b19 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;                %first image
global img2;                %second image
global out;

[img1_row, img1_column, ~] = size(img1);
img2 = imresize(img2,[img1_row img1_column]);
out = img1 + img2;
axes(handles.axes11);
imshow(out);


% --- Executes on button press in b20.
function b20_Callback(hObject, eventdata, handles)
% hObject    handle to b20 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;                %first image
global img2;                %second image
global out;

[img1_row, img1_column, ~] = size(img1);
img2 = imresize(img2,[img1_row img1_column]);
out = img1 - img2;
axes(handles.axes11);
imshow(out);
```

```
% --- Executes on button press in b21.
function b21_Callback(hObject, eventdata, handles)
% hObject    handle to b21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;                %first image
global img2;                %second image
global out;

[img1_row, img1_column, ~] = size(img1);
img2 = imresize(img2,[img1_row img1_column]);
out = img1 .* img2;
axes(handles.axes11);
imshow(out);


% --- Executes on button press in b22.
function b22_Callback(hObject, eventdata, handles)
% hObject    handle to b22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;                %first image
global img2;                %second image
global out;

[img1_row, img1_column, ~] = size(img1);
img2 = imresize(img2,[img1_row img1_column]);
out = img1 ./ img2;
axes(handles.axes11);
imshow(out);


% --- Executes on button press in b23.
function b23_Callback(hObject, eventdata, handles)
% hObject    handle to b23 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;

HSV = rgb2hsv(img1);

% "90% more" saturation:
HSV(:, :, 2) = HSV(:, :, 2) * 1.9;
out = hsv2rgb(HSV);

axes(handles.axes11);
imshow(out);


% --- Executes on button press in b24.
function b24_Callback(hObject, eventdata, handles)
% hObject    handle to b24 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;

out = im2double(img1) .^ 2.2;
axes(handles.axes11);
imshow(out);


% --- Executes on button press in b25.
function b25_Callback(hObject, eventdata, handles)
% hObject    handle to b25 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;

BW = rgb2gray(img1);
out = imbinarize(BW,0.4);

axes(handles.axes11);
imshow(out);

[~, num] = bwlabel(out,4);

props = regionprops(out);


for i=1:size(props)
  position = props(i).BoundingBox;
  rectangle('Position', position, 'EdgeColor', 'r', 'LineWidth', 3,
'LineStyle', '-')
end

components = num2str(num);
a = " components";

final = strcat(components, a);

set(handles.textbox, 'String', final);



% --- Executes on button press in b26.
function b26_Callback(hObject, eventdata, handles)
% hObject    handle to b26 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;            %carrier image
global img2;            %secret message image
global out;

carrier_img = img1;
```

```
secret_img = img2;

[carrier_row,carrier_column,~] = size(carrier_img);
secret_img = imresize(secret_img,[carrier_row carrier_column]);

red_carrier = carrier_img(:,:,1);
green_carrier = carrier_img(:,:,2);
blue_carrier = carrier_img(:,:,3);

red_secret = secret_img(:,:,1);
green_secret = secret_img(:,:,2);
blue_secret = secret_img(:,:,3);


%Operation on Red Channel
for i = 1:carrier_row
    for j = 1:carrier_column
        new_carrier(i,j) = bitand(red_carrier(i,j),254); %254 = 11111110
        new_secret(i,j) = bitand(red_secret(i,j),128); %128 = 10000000
        divided_secret(i,j) = new_secret(i,j)/128; %It will make it from
10000000 to 00000001
        final_red(i,j) = new_carrier(i,j) + divided_secret(i,j);
    end
end

%Operation on Green Channel
for i = 1:carrier_row
    for j = 1:carrier_column
        new_carrier(i,j) = bitand(green_carrier(i,j),254); %254 = 11111110
        new_secret(i,j) = bitand(green_secret(i,j),128); %128 = 10000000
        divided_secret(i,j) = new_secret(i,j)/128; %It will make it from
10000000 to 00000001
        final_green(i,j) = new_carrier(i,j) + divided_secret(i,j);
    end
end

%Operation on Blue Channel
for i = 1:carrier_row
    for j = 1:carrier_column
        new_carrier(i,j) = bitand(blue_carrier(i,j),254); %254 = 11111110
        new_secret(i,j) = bitand(blue_secret(i,j),128); %128 = 10000000
        divided_secret(i,j) = new_secret(i,j)/128; %It will make it from
10000000 to 00000001
        final_blue(i,j) = new_carrier(i,j) + divided_secret(i,j);
    end
end

out = cat(3, final_red,final_green,final_blue); %cat function concatenates
three dimensions of images.
axes(handles.axes11);
imshow(out);
```

```
% --- Executes on button press in b27.
function b27_Callback(hObject, eventdata, handles)
% hObject    handle to b27 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;

stegnographed_image = img1;

[carrier_row,carrier_column,~] = size(img1);

stegnographed_image_red = stegnographed_image(:,:,1);
stegnographed_image_green = stegnographed_image(:,:,2);
stegnographed_image_blue = stegnographed_image(:,:,3);

%Red Channel Decoding
for i = 1:carrier_row
    for j = 1:carrier_column
       new_carrier(i,j) = bitand(stegnographed_image_red(i,j),1);
       multiplied_carrier(i,j) = new_carrier(i,j)*128;
    end
end
decoded_red = multiplied_carrier;

%Green Channel Decoding
for i = 1:carrier_row
    for j = 1:carrier_column
       new_carrier(i,j) = bitand(stegnographed_image_green(i,j),1);
       multiplied_carrier(i,j) = new_carrier(i,j)*128;
    end
end
decoded_green = multiplied_carrier;

%Blue Channel Decoding
for i = 1:carrier_row
    for j = 1:carrier_column
       new_carrier(i,j) = bitand(stegnographed_image_blue(i,j),1);
       multiplied_carrier(i,j) = new_carrier(i,j)*128;
    end
end
decoded_blue = multiplied_carrier;

out = cat(3,decoded_red,decoded_green,decoded_blue);
axes(handles.axes11);
imshow(out);


function b28_Callback(hObject, eventdata, handles)
% hObject    handle to b28 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global img2;
global out;

[img1_row, img1_column,~] = size(img1);
img2 = imresize(img2,[img1_row img1_column]);

out = [img1, img2];
axes(handles.axes11);
imshow(out);


% --- Executes on button press in b29.
function b29_Callback(hObject, eventdata, handles)
% hObject    handle to b29 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;

temp = rgb2gray(img1);
out = medfilt2(temp);

axes(handles.axes11);
imshow(out);


% --- Executes on button press in b30.
function b30_Callback(hObject, eventdata, handles)
% hObject    handle to b30 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;

out = flip(img1,2);
axes(handles.axes11);
imshow(out);


% --- Executes on button press in b31.
function b31_Callback(hObject, eventdata, handles)
% hObject    handle to b31 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;
h = fspecial('log',7,0.4);
out = imfilter(img1,h);
axes(handles.axes11);
imshow(out);


% --- Executes on button press in b32.
function b32_Callback(hObject, eventdata, handles)
% hObject    handle to b32 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;

A = imcomplement(img1);
B = imreducehaze(A, 'Method','approx','ContrastEnhancement','boost');
out = imcomplement(B);
axes(handles.axes11);
imshow(out);



% --- Executes on button press in b33.
function b33_Callback(hObject, eventdata, handles)
% hObject    handle to b33 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;

out = imgaussfilt(img1,2);
axes(handles.axes11);
imshow(out);





% --- Executes on button press in b34.
function b34_Callback(hObject, eventdata, handles)
% hObject    handle to b34 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img1;
global out;

[img1_row, img1_column,~] = size(img1);
A = imresize(img1,[img1_column img1_row ]);

out = (((1.5 - im2double(A)) * 1.3) ./ 2.1);
axes(handles.axes11);
imshow(out);



% --- Executes on button press in b35.
function b35_Callback(hObject, eventdata, handles)
% hObject    handle to b35 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global img2;
[fName, pName] = uigetfile('*jpg; *jpeg; *png; *tif;', 'Select an Image');
if pName ~= 0
```

```
    path  = [pName fName];
    img2 = imread(path);
end



function textbox_Callback(hObject, eventdata, handles)
% hObject    handle to textbox (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of textbox as text
%        str2double(get(hObject,'String')) returns contents of textbox as a
double


% --- Executes during object creation, after setting all properties.
function textbox_CreateFcn(hObject, eventdata, handles)
% hObject    handle to textbox (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```